

**DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y
COMPUTACIÓN**

UNIVERSIDAD POLITÉCNICA DE VALENCIA

P.O. Box: 22012 E-46071 Valencia (SPAIN)



Informe Técnico / Technical Report

Ref. No: DSIC-II/11/06

Pages: 241

Title: USO DE UN MODELO ARQUITECTÓNICO DE
COMPONENTES Y ASPECTOS EN SISTEMAS DE
DIAGNÓSTICO

Author (s): María Eugenia Cabello Espinosa,
Isidro Ramos Salavert,
José Ángel Carí Cubel

Date: 20 de Julio de 2006

Key Words: arquitecturas software, reutilización del software,
lenguaje de definición de arquitecturas, elementos arquitectónicos,
aspectos, sistemas expertos, diagnóstico médico.

VºBº
Leader of Reasearch Group

Author (s):

Isidro Ramos Salavert

Mª Eugenia Cabello Espinosa

RESUMEN

Este trabajo de investigación presenta el análisis y diseño de un modelo arquitectónico basado en componentes y orientado a aspectos, considerando el caso de estudio de enfermedades infecciosas infantiles, dentro del dominio del diagnóstico médico. Se especifican los distintos elementos arquitectónicos que conforman el modelo, de un gránulo funcional más simple al más complejo, utilizando el LDA de PRISMA en sus dos niveles: el de tipos para definir los elementos arquitectónicos del modelo, y el nivel de configuración permitiendo definir las instancias y especificar la topología del modelo.

Palabras clave: arquitecturas software, reutilización del software, lenguaje de definición arquitecturas, elementos arquitectónicos, aspectos, sistemas expertos, diagnóstico médico.

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO I: MARCO CONCEPTUAL	
1.1 El modelo PRISMA	15
1.2 Los sistemas expertos en el diagnóstico médico	17
CAPÍTULO II: DESCRIPCIÓN DEL PROBLEMA PARA EL CASO DE ESTUDIO	
2.1 Descripción del problema	22
2.2 Escenarios, casos de uso y diagramas de secuencia	25
2.2.1 Casos de uso	27
2.2.2. Diagramas de secuencia de los casos de uso	32
CAPÍTULO III: MODELO ARQUITECTÓNICO DE UN SISTEMA PARA TAREAS DE DIAGNÓSTICO MÉDICO	
3.1. Identificación	35
3.1.1 Identificación de los elementos arquitectónicos	35
3.1.2. Identificación del modelo arquitectónico final	39
3.1.3 Identificación de los <i>concerns</i>	40
3.1.4 Identificación de los aspectos	40
3.1.4.1 Identificación de los aspectos del Componente Motor de Inferencia	40
3.1.4.2 Identificación de los aspectos de Componente Base de Conocimientos	41
3.1.4.3 Identificación de los aspectos del Conector Diagnóstico	41
3.1.4.4 Identificación de los aspectos del Componente Usuario Clínico	41
3.1.4.5 Identificación de los aspectos del Componente Usuario Laboratorio	42
3.1.4.6 Identificación de los aspectos del Conector Usuarios	42
3.1.5 Identificación de interfaces	43
3.2 Diagramas de secuencia correspondientes a las realizaciones de los casos de uso	44
3.3 Especificación	51
3.3.1 Especificación de interfaces, aspectos y elementos arquitectónicos.	
Configuración de los elementos arquitectónicos	51
3.3.1.1 Especificación de los aspectos del componente Motor de Inferencia	51

3.3.1.2	Especificación del componente Motor de Inferencia	65
3.3.1.3	Instanciación del componente Motor de Inferencia	69
3.3.1.4	Especificación de los aspectos del componente Base de Conocimientos	69
3.3.1.5	Especificación del componente Base de Conocimientos	94
3.3.1.6	Instanciación del componente Base de Conocimientos	96
3.3.1.7	Especificación de los aspectos del conector Diagnóstico	96
3.3.1.8	Especificación del conector Diagnóstico	117
3.3.1.9	Instanciación del conector Diagnóstico	119
3.3.1.10	Especificación del sistema Diagnosticador	119
3.3.1.11	Instanciación del componente del sistema Diagnosticador	122
3.3.1.12	Especificación de los aspectos del componente Usuario Clínico	122
3.3.1.13	Especificación del componente Usuario Clínico	139
3.3.1.14	Instanciación del componente Usuario Clínico	147
3.3.1.15	Especificación de los aspectos del componente Usuario Laboratorio	147
3.3.1.16	Especificación del componente Usuario Laboratorio	156
3.3.1.17	Instanciación del componente Usuario Laboratorio	163
3.3.1.18	Especificación de los aspectos del conector Usuarios	163
3.3.1.19	Especificación del conector Usuarios	181
3.3.1.20	Instanciación del componente del conector Usuarios	182
3.3.2.	Especificación del modelo arquitectónico y configuración final	183
3.3.2.1	Especificación del modelo arquitectónico final	183
3.3.2.2	Instanciación de la configuración final	185
3.4	Ventajas de realizar la especificación en el LDA de PRISMA	186
CONCLUSIONES Y TRABAJO FUTURO		
	Conclusiones	189
	Trabajo futuro	192
BIBLIOGRAFÍA Y REFERENCIAS		
		193

APÉNDICES

APÉNDICE A:	Ontología del diagnóstico médico	202
APÉNDICE B:	Elementos del dominio del diagnóstico médico:	
	Las enfermedades infecciosas infantiles	206
APÉNDICE C:	Reglas para inferir las entidades etiofisiopatológicas, nosológicas y terapéuticas del dominio de las enfermedades infecciosas infantiles	210
APÉNDICE D:	Código del sistema <i>DiagMed</i>	224

ÍNDICE DE FIGURAS

Fig. 1	Vistas de un elemento arquitectónico PRISMA	15
Fig. 2	Elemento arquitectónico PRISMA	16
Fig. 3	Vista del médico como usuario del sistema de diagnóstico médico	23
Fig. 4	Entradas y salidas del sistema de diagnóstico médico	24
Fig. 5	Casos de uso del sistema de diagnóstico médico	27
Fig. 6	Diagrama de secuencia del caso de uso Registrar paciente	32
Fig. 7	Diagrama de secuencia del caso de uso: Visualizar datos paciente.	32
Fig. 8	Diagrama de secuencia del caso de uso: Realizar diagnóstico clínico	33
Fig. 9	Diagrama de secuencia del caso de uso: Realizar diagnóstico laboratorio	34
Fig. 10	Diagrama de secuencia del caso de uso: Resultados del diagnóstico	34
Fig. 11	Componente Motor de Inferencia	35
Fig. 12	Componente Base de Conocimientos	36
Fig. 13	Conector Diagnóstico	36
Fig. 14	Sistema Diagnosticador	37
Fig. 15	Componente Usuario Clínico	38
Fig. 16	Componente Usuario Laboratorio	38
Fig. 17	Conector Usuarios	38
Fig. 18	Arquitectura del Sistema de Diagnóstico Médico	39
Fig. 19	Diagrama de secuencia de las realizaciones del caso de uso: Registrar paciente	44
Fig. 20	Diagrama de secuencia de las realizaciones del caso de uso: Visualizar datos paciente	45
Fig. 21	Diagrama de secuencia de las realizaciones del caso de uso: Realizar diagnóstico clínico	47
Fig. 22	Diagrama de secuencia de las realizaciones del caso de uso: Realizar diagnóstico de laboratorio	49
Fig. 23	Diagrama de secuencia de las realizaciones del caso de uso: Resultados del diagnóstico	50
Fig. 24	Esquema del funcionamiento de la Interfaz Inferencia	54
Fig. 25	Grafo del proceso de inferencia de las enfermedades	56
Fig. 26	Componente Motor de Inferencias	67
Fig. 27	Grafo del proceso de inferencia de las posibles enfermedades	70
Fig. 28	Grafo del proceso de inferencia de las sugerencias de laboratorio.	71
Fig. 29	Grafo del proceso de inferencia para adquirir los signos y síntomas de grano fino	72
Fig. 30	Grafo del proceso de inferencia para adquirir los estudios de laboratorio	75
Fig. 31	Grafo del proceso de inferencia de la terapia	79

Fig. 32	Esquema del funcionamiento de la Interfaz IDomain	79
Fig. 33	Razonamientos médicos para realizar un diagnóstico	81
Fig. 34	Grafo de inferencia de los razonamientos forward y backward para realizar el diagnóstico médico	82
Fig. 35	Componente Base de Conocimientos	95
Fig. 36	Esquema que muestra el flujo de los servicios a través de los puertos del conector Diagnóstico	96
Fig. 37	Grafo de transición del protocolo del conector Diagnóstico	97
Fig. 38	Puertos del conector Diagnóstico.	102
Fig. 39	Conector Diagnóstico	118
Fig. 40	Sistema Diagnosticador.	121
Fig. 41	Esquema del funcionamiento de la Interfaz IClinical	124
Fig. 42	Gráficos que puede utilizar el aspecto de presentación del Usuario Clínico	133
Fig. 43	Componente Usuario Clínico	144
Fig. 44	Esquema del funcionamiento de la Interfaz ILaboratory	148
Fig. 45	Gráficos que puede utilizar el aspecto de presentación del Usuario Laboratorio	153
Fig. 46	Componente Usuario Laboratorio	161
Fig. 47	Esquema que muestra el flujo de los servicios a través de los puertos del conector Usuarios	163
Fig. 48	Grafo de transición del protocolo del conector Usuarios	164
Fig. 49	Puertos del conector Usuarios	168
Fig. 50	Conector Usuarios	182
Fig. 51	Modelo Arquitectónico final del Sistema de Diagnóstico Médico <i>DiagMed</i>	184
Fig. 52	Grafo RDF del diagnóstico médico	202

INTRODUCCIÓN

Actualmente han emergido dos necesidades: la de diseñar arquitecturas para sistemas a un mayor nivel de abstracción, de forma que sea posible reutilizar los diseños para distintos tipos de arquitecturas y que sean fácilmente modificables; y la necesidad de la evolución de los diseños de forma correcta y preservando los requisitos de calidad del sistema, tanto en tiempo de compilación como en tiempo de ejecución.

Para solventar estas necesidades, ha surgido el modelo PRISMA (PlatafoRma oasIS para Modelos Arquitectónicos), un enfoque integrado y flexible para describir modelos arquitectónicos complejos, distribuidos, evolutivos y reutilizables, basados en componentes y orientados a aspectos. PRISMA presenta propiedades y ventajas en la construcción de modelos arquitectónicos complejos que ilustramos mediante la especificación de un caso de estudio. El desarrollo del caso mostrará como PRISMA potencia la reutilización, soporta la evolución de sus tipos y topologías mediante sus capacidades reflexivas.

Por otro lado, es notable el interés que han cobrado en los últimos años los sistemas que realizan tareas de diagnóstico (médico, electrónico, mecánico, de software, entre otros), con el fin de identificar las causas internas que provocan un problema o disfunción a partir de una serie de datos (variables observables) y que son consecuencia de las mismas. Al respecto, es conveniente contemplar que este tipo de aplicaciones ha desarrollado una gran importancia en nuestros días en el campo de la medicina.

Lo anteriormente expuesto ha provocado que para la realización del presente trabajo se haya elegido la arquitectura PRISMA como ámbito de desarrollo de los sistemas de diagnóstico. Con este enfoque, se elaboró la construcción de un modelo arquitectónico, mediante la especificación del caso de estudio de las enfermedades infecciosas infantiles.

Por ello se establece como objetivo del presente trabajo el proponer un modelo arquitectónico con componentes y aspectos para sistemas que realicen tareas de diagnóstico, considerando el caso de estudio de enfermedades infecciosas infantiles, dentro del dominio del diagnóstico médico.

Se identificaron y posteriormente se especificaron los distintos elementos arquitectónicos que conforman el modelo, de un gránulo funcional más simple al más complejo (ascendente), utilizando el Lenguaje de Descripción de Arquitecturas de PRISMA en sus dos niveles:

- el nivel de tipos, utilizando un lenguaje de definición de componentes para definir los tipos de la arquitectura (componentes, conectores y sistemas), potenciando la

reutilización que integra el Desarrollo de Software Basado en Componentes y el Desarrollo de Software Orientado a Aspectos, y

- el nivel de configuración para especificar el modelo arquitectónico, permitiendo definir las instancias y especificar la topología del modelo.

Cabe señalar que, el modelo PRISMA ha sido aplicado a otros dominios, como la arquitectura PRISMA del robot TeachMover, el cual posee una gran cantidad de componentes, conectores y attachments a diferentes niveles de abstracción [Per04], así como en sistemas de manejo de cuentas bancarias [Per03a], mostrando el mejoramiento del desarrollo y mantenimiento de dichos sistemas.

En un futuro, este trabajo será la base de una tesis, en la que se realizará una investigación que toca problemas fundamentales de las arquitecturas software y de la inteligencia artificial, al proponer un modelo arquitectónico orientado a aspectos de los sistemas expertos

Para la presentación de este trabajo de investigación, se consideró necesario desarrollar tres capítulos:

- En el capítulo I se describen los conceptos principales referentes al modelo Prisma y una visión de los sistemas expertos en el campo del diagnóstico médico.
 - Del modelo Prisma se presenta brevemente la descripción de dicho modelo y su perspectiva desde una visión interna, la orientada a aspectos, así como una visión externa, la basada en componentes. Así mismo se comenta sobre el lenguaje de definición de arquitecturas que utiliza PRISMA, a saber el lenguaje de definición de tipos y el lenguaje de configuración arquitectónica.
 - De los sistemas expertos se comenta sobre el impacto y uso de estos sistemas en el diagnóstico médico.
- En el capítulo II se presenta una descripción del problema con el caso de estudio de las enfermedades infecciosas infantiles en el dominio del diagnóstico médico, dado que PRISMA presenta sus propiedades en la construcción de modelos arquitectónicos mediante el desarrollo y especificación de un caso de estudio. Inicialmente se realiza una breve descripción del problema y a continuación se realizan los casos de uso que se corresponden con los escenarios (escenas funcionales del sistema), mismos que se describen con los diagramas de secuencias de cada caso de uso (vista de la interacción humano-máquina).

- El capítulo III representa la parte experimental de la tesis, donde se establece la propuesta de un modelo arquitectónico para tareas de diagnóstico médico en el marco del modelo PRISMA, mencionándose el funcionamiento y estructura de éste. Para ello, este capítulo se ha seccionado en cuatro partes. En el primer apartado se realiza la identificación de cada uno de los elementos arquitectónicos, de los *concerns* que atraviesan el sistema, de los aspectos de cada elemento arquitectónico y de las interfaces. En el segundo apartado se presentan los diagramas de secuencias de las realizaciones de los casos de uso (vista del funcionamiento del sistema). En el tercer apartado se presenta la especificación en el LDA de PRISMA de las interfaces, seguida por la de los aspectos y finalmente la de los elementos que conforman al modelo. Así como la especificación de la topología del modelo arquitectónico propuesto y se definen las instancias en la configuración final de dicho modelo. En el cuarto apartado se mencionan las ventajas de dicha especificación.
- Finalmente, se establecen las conclusiones que se desprenden del análisis del sistema propuesto, donde se incluyen los futuros trabajos relacionados con el tema de este trabajo de investigación.

Se ha intentado proporcionar al lector aquellos aspectos relacionados con el modelo arquitectónico PRISMA y los sistemas expertos basados en reglas, los cuales resultarán de gran utilidad para una mejor comprensión de lo discutido en los capítulos II y III de este trabajo. Este material ha sido escrito para personas relacionadas con los campos de la arquitectura de software, la inteligencia artificial y de la medicina. El trabajo presenta el caso de las enfermedades infecciosas infantiles (rubéola, sarampión, roséola, escarlatina, varicela, dengue, tifoidea, parotiditis viral o paperas, parotiditis bacteriana, crup espasmódico o laringitis estridulosa, bronquiolitis infecciosa aguda, neumonía, otitis viral y otitis bacteriana) las cuales están enmarcadas en el contexto del diagnóstico médico, como una aplicación en el dominio de la medicina, y cuya información contenida en los apéndices B y C de este documento, fueron consultados con un pediatra en funciones [Ros04].

Al final de este documento se han incorporado algunos apéndices para clarificar el caso de uso y para enriquecer el trabajo presentado:

- En el apéndice A se muestra una breve ontología del diagnóstico médico.
- El apéndice B presenta los elementos del dominio del diagnóstico médico.
- El apéndice C incluye las reglas que utiliza la base de conocimientos del sistema propuesto para inferir las entidades etiofisiopatológicas (o síndromes), nosológicas (o enfermedades) y terapéuticas (o terapia) del dominio.
- Y finalmente, con el fin de mostrar que las propiedades de PRISMA se satisfacen, en el apéndice D se presenta la implementación de dicho sistema, codificado en Turbo Prolog 1.1 (Borland International) como un sistema experto tradicional. Se mencionan

algunas de las ventajas que se tienen con la especificación en el LDA de PRISMA frente al PROLOG. A continuación se comentan algunas similitudes entre ambos.

ESTADO DEL ARTE

Arquitecturas software y la arquitectura PRISMA

En nuestros días existen una gran cantidad y variedad de trabajos relacionados con el desarrollo de sistemas software complejos: modelos arquitectónicos, reconfiguración dinámica de arquitecturas, componentes, aspectos, lenguajes de definición de arquitecturas. Todo ello persigue un consenso, aún no logrado, en los conceptos básicos de sistemas software complejos, su definición precisa y la metodología a aplicar para su desarrollo.

De los complejos requisitos software surge el modelo PRISMA (Plataforma OASIS para Modelos Arquitectónicos), su potencia expresiva, facilidad de uso y novedad. De esta manera se cubre el hueco existente en el modelado de sistemas software altamente reconfigurables y reutilizables dentro de un marco de calidad controlada.

PRISMA [Per02] presenta propiedades y ventajas en la construcción de modelos arquitectónicos complejos mediante la especificación de un caso de estudio, el cual, al desarrollarlo, mostrará como PRISMA potencia la reutilización, soporta la evolución de sus tipos y topologías mediante sus capacidades reflexivas.

PRISMA ha tenido en cuenta varios trabajos para su modelo, que incorporan las distintas áreas de interés, entre ellos:

El trabajo realizado por Garlan [Gar01] es un punto de referencia muy importante a la hora de establecer los elementos de un sistema software complejo, en el cual se introducen conceptos como componente, conector, sistema, puertos de entrada y de salida.

Se han considerado los trabajos de Andrade y Fiadeiro [And99] acerca del concepto de contrato, ya que se han definido los conectores como una extensión del contrato propuesto por su trabajo. Dicha extensión se produce debido a la incorporación del concepto de coreografía en el conector, concepto utilizado con una semántica distinta en los trabajos de [Mon02] y [Bra02].

Una línea cercana, se encuentra en The Model Driven Architecture (MDA), un enfoque propuesto por The Object Management Group (OMG) [Omg] para la integración e interoperabilidad de los sistemas. MDA es una iniciativa de la OMG que surge como una

de las estrategias más prometedoras para el diseño y desarrollo de aplicaciones software que propone la definición de modelos de alto nivel de abstracción, independientes de cualquier hardware, sistema operativo o plataforma que finalmente lo soporte (Modelo Independiente de Plataforma, PIM).

Existe una gran variedad de lenguajes de definición de arquitecturas (LDA), cada uno de ellos presenta ventajas y desventajas, tal y como presenta el estudio realizado en [Med00]. Uno de los lenguajes que más se aproxima al LDA de PRISMA es el propuesto por Loques y Leit [Loq00] en su modelo R-RIO, en el cual incorporan aspectos de reconfiguración dinámica a través del metanivel. Estos también se encuentran en enfoques orientados a un nivel de abstracción inferior, es decir, orientados a implementación como [McG02]. En Guaraná [Oli98] se presenta un metanivel complejo que hay que definir durante la compilación, siendo su estructura ajena al modelo.

Otro marco de referencia es el desarrollo de software orientado a aspectos en el que se encuentran trabajos como [Aos03]. Esta aproximación se aplica principalmente en la implementación y no se aplica para la especificación a un alto nivel de abstracción de sistemas arquitectónicos que integren componentes y aspectos. El DSOA está teniendo un gran auge en la comunidad informática debido a que su código modular consigue que los costes de desarrollo, mantenimiento y evolución del software se reduzcan. Por este motivo, existe una tendencia hacia la incorporación del concepto de aspecto desde las primeras fases del ciclo de vida, existiendo trabajos elaborados en el área de la especificación de requisitos [Ras02].

Los aspectos se definen en [Kic97] como propiedades de un sistema que tienden a estar presentes en varias componentes funcionales, dando lugar al término *crosscutting concerns*. En PRISMA se entiende a un aspecto como la definición completa de la estructura y el comportamiento de un tipo desde un determinado punto de vista o *concern*. Por ello, el *crosscutting* no sólo se realiza sobre propiedades funcionales sino sobre cualquier *concern*.

En PRISMA se define el concepto de aspecto sin tener en cuenta las peculiaridades sintácticas de los lenguajes de programación orientados a aspectos. Su definición se basa en las características comunes de un sistema susceptibles de ser reutilizadas. Atendiendo a esta definición, menos dependiente del lenguaje de desarrollo, la funcionalidad es susceptible de reutilización en distintos tipos e incluso dicha funcionalidad se puede subdividir en distintos aspectos. Los últimos trabajos de [Brio03] empiezan a apuntar hacia esta última idea.

La mayoría de los modelos arquitectónicos analizan cuáles son los tipos básicos para la especificación de sus arquitecturas y exponen su sintaxis y semántica. El modelo

PRISMA, además de definir sus tipos, también especifican los aspectos que cubren las necesidades de cada uno de ellos. Un tipo del modelo PRISMA puede ser visto como un prisma con tanta caras como aspectos considere, los cuales están definidos desde la perspectiva del problema y no de su solución, aumentando el nivel de abstracción y evitando el solapamiento de código (no monotonicidad) que puede producir la programación orientada a aspectos [Kic01].

Dentro de los distintos modelos de aspectos que se ha creado hasta el momento, se pueden observar dos claras tendencias que permiten clasificarlos en dos grandes grupos: los modelos estáticos y los dinámicos. La diferencia entre ambos se encuentra en el proceso de generación de código. Los modelos estáticos generan un solo componente en el que se encuentra mezclado el código funcional y el código de los aspectos, mientras que en los modelos dinámicos se generan distintas entidades para los distintos aspectos y el componente (modelo de disfraces) [Her03]. La ventaja de los modelos dinámicos es la facilidad que proporcionan para incluir o eliminar aspectos durante el proceso de ejecución.

Algunas de las aproximaciones que han integrado el Desarrollo de Software Basado en Componentes (DSBC) y el Desarrollo de Software Orientado a Aspectos (DSOA) a nivel de implementación han extendido los java beans utilizando aspectos para su descripción y añadiendo el concepto de conectores [Suv03]. Además, los aspectos pueden ser manipulados en tiempo de ejecución. Un trabajo que describe los asuntos que se tienen que tener en consideración para la integración de ambas aproximaciones es introducido en [Con00]. Los asuntos y requisitos descritos en este trabajo son contemplados en PRISMA, ya que PRISMA aglutina estas dos aproximaciones actuales del DSBC y DSOA

El metanivel del modelo PRISMA y sus propiedades reflexivas de los lenguajes diseñados, dan soporte a la evolución de los elementos arquitectónicos y a la reconfiguración dinámica de la topología. Esta característica permite definir un metanivel por aspecto que reifica las propiedades que deseen ser evolucionadas basándose en la reflexión de OASIS [Car99]. Con ello, PRISMA da soporte a la evolución de sus modelos reduciendo el esfuerzo de mantenimiento de sus productos; la evolución proporciona la dinámica del tipo, i.e. la capacidad de cambio de su estructura y comportamiento basándose en la evolución del software que plantea la reflexión de OASIS.

Los elementos arquitectónicos PRISMA se obtienen identificando escenas funcionales del sistema y asignando a cada elemento una escena funcional, dependiendo de si la escena es simple o compleja, el elemento será un componente (componente simple) o un sistema (componente complejo), respectivamente. El concepto de escena aparece en el trabajo de [Nor98], y es definido con un enfoque PRISMA por [Per03b] al considerar que "Una escena se caracteriza por las tareas que se desempeñan en la actividad siguiendo

un determinado protocolo, los actores que las realizan y el espacio virtual o físico donde se desarrolla”

Este enfoque para la detección de componentes, es compatible con The Architecture Based Design Method (ABD) [Bac00], ya que este método se basa en la descomposición funcional del problema, es decir del sistema. ABD es una metodología propuesta por The Software Engineering Institute of The Carnegie Mellon University para diseñar arquitecturas software de un dominio de aplicación.

Siguiendo esta metodología se van a identificar los distintos elementos arquitectónicos del Sistema de Diagnóstico Médico. La identificación funcional se puede realizar de un gránulo funcional mayor a más fino (descendente) o de nivel más simple al más complejo (ascendente). En este caso se va a realizar una identificación ascendente, ya que resulta más sencillo para el análisis del mismo.

Finalmente, es muy conveniente mencionar la investigación que se ha estado realizando en el marco del modelo PRISMA, en el grupo denominado Ingeniería de Software y Sistemas de Información (ISSI) del Departamento de Sistemas Informáticos y Computación (DSIC) de la Universidad Politécnica de Valencia, a través de un importante análisis y una especificación detallada, de los aspectos funcional, de distribución y de replicación, que aparecen en los trabajos de las doctorantes Jennifer Pérez y Nour Alí [Ali03a], [Ali03b], [Ali04a], [Ali04b], [Ali04c], [Ali04d], [Ali04e], [Ali04f], [Ali05], [Per02], [Per03a], [Per03b], [Per03c], [Per03d], [Per03e], [Per03f], [Per04], [Per05].

Sistemas expertos, la toma de decisiones y el diagnóstico

En el transcurso de la vida, el hombre tiene un problema cuando no sabe cómo continuar en algo. De acuerdo a Von der Becke, una vez que ha detectado un problema, se tiene que tomar una decisión (y aunque suene ilógico, inclusive la de no hacer nada). Se elige una alternativa que le parezca suficientemente racional, y que le permita incrementar o no, el valor esperado después de que se haya resuelto el problema. Posteriormente se emite un plan de control que servirá de guía en la toma de decisiones, inclusive en decisiones con respecto a modificar dicho plan.

El hombre se enfrenta a varios tipos de decisiones. Howard realiza una clasificación de esos tipos, que pueden ser consultados en la página de Internet: lafacu.com:

Existen diferentes tipos de Sistemas de Apoyo a las Decisiones:

- Sistema de Soporte a la Toma de Decisiones (DSS)
- Sistemas de Información para Ejecutivos (EIS)

- Sistemas para la Toma de Decisiones en Grupo (GDSS)
- Sistemas Expertos para la Toma de Decisiones (EDSS)

Los Sistemas de Soporte a la toma de Decisiones (del término inglés Decision Support Systems ó DSS) han ayudado a los ejecutivos a tomar decisiones dentro de las organizaciones desde los años 60`s. El DSS en sí, es un bloque de toma de decisiones sustentado en base de datos que quienes toman las decisiones puedan usar para apoyar el proceso de decidir. Little (1970) define los DSS como un modelo basado en un conjunto de procedimientos para procesar datos y juicios que ayudan a los administradores a tomar decisiones. Bonczek et al (1980) definen los DSS como un sistema basado en computadoras con tres componentes clave: lenguaje de sistema (un mecanismo para proveer comunicación entre el usuario y otros componentes del DSS), un sistema de conocimiento (repositorio de conocimiento sobre problemas, datos o procesos) y el sistema procesador de problemas (la unión entre los otros dos componentes). En nuestros días la toma de decisiones es uno de los procesos más valorados en cualquier empresa.

De los diferentes tipos de sistemas de apoyo a las decisiones, los sistemas expertos darán mayor soporte en el proceso de toma de decisiones, permitiendo tener el conocimiento del experto capturando en una base de conocimiento y utilizarlo cuando se requiera sin que esté él presente. Los sistemas expertos que están basados en reglas, contienen conocimientos predefinidos que se utilizan para tomar todas las decisiones.

Por ello, los sistemas expertos empiezan a tener cada vez mayor auge, hasta el punto de ir suponiendo un punto de referencia importante en la toma de decisiones. En realidad, incluso se podría decir que el límite de las aplicaciones, objeto de los sistemas expertos, está en la imaginación humana, siendo siempre de utilidad allí donde se necesite un experto.

Diferentes definiciones de sistemas expertos han sido propuestas hasta la fecha. Sin embargo un fundamento común puede ser extraído de todas ellas: un sistema experto es un sistema con pericia en la solución de problemas; esto es, un sistema que posee razonamientos, habilidades y conocimientos acerca de un dominio particular, para resolver los problemas de forma similar a la de un experto humano.

Los primeros éxitos en los sistemas expertos surgieron a mediados de las décadas de los 60`s. Joshua Lederberg, profesor de genética de la Universidad de Stanford, realizó un programa para enumerar todas las posibles configuraciones permitidas de un conjunto de átomos. Él llamó a su programa DENDRAL. Posteriormente a principios de los 80`s los sistemas expertos salen del ámbito de las universidades y de los laboratorios de investigación para infiltrarse con una tecnología comercial muy prometedora para

solucionar una cantidad y variedad de problemas. A principios de la década de los 90's los sistemas expertos se emplean con bastante éxito en problemas de diagnóstico de fallas, interpretación de datos, predicción de comportamientos, planeación de producción, monitoreo de sistemas, el diseño, la depuración, la reparación, la instrucción y control de procesos, entre otros.

Algunos de los campos de aplicación de los sistemas expertos son: medicina, finanzas y gestión, educación, administración, militar, transportes, aeronáutica, agricultura, arqueología, derecho, geología, industria electrónica, informática y telecomunicaciones.

Las metodologías y aplicaciones desarrolladas de los sistemas expertos tienen una amplia categoría de productos de investigación, ofreciendo sugerencias y soluciones a dichos sistemas en dominios específicos. Dichas metodologías y aplicaciones están diversificadas según los antecedentes de los autores, su experticia, sus intereses de investigación, sus habilidades en la metodología utilizada y el dominio del problema. Un estudio realizado por [Liao, 2005] examina y contempla las metodologías de los sistemas expertos, clasificándolos en 11 categorías.

Asimismo, los sistemas expertos han sido implementados en diversos paradigmas de programación, como la estructurada, la lógica y la orientada a objetos, entre otros, existiendo una tendencia en su desarrollo hacia lenguajes de cuarta generación y métodos de programación visual para dar un ambiente y una estructura amigable de comunicación con el usuario [Liao, 2003].

Sin embargo, conforme a lo consultado en la literatura, la arquitectura de estos sistemas se ha realizado con una estructura basada en componentes [Giarratano and Riley, 2004] sin considerar adicionalmente la orientación a aspectos, como lo contempla el modelo propuesto en este trabajo, que ha integrado estos dos enfoques.

La aplicación de los sistemas expertos ha sido extensa en muchos ámbitos de la vida humana, sin embargo se ha mostrado un notable interés en la ciencia biomédica y la práctica clínica. Por ello, la construcción de los sistemas expertos para resolver tareas de diagnóstico en el campo de la medicina es muy amplia, como puede observarse en [Liebewitz, 1998].

En lo referente al diagnóstico médico, existe una serie de aplicaciones extensa en número, pero quizá la más conocida, a la vez que la más antigua, podría ser MYCIN [Mycin]. El cual es el primer sistema experto que llegó a funcionar con la misma calidad que un experto humano, dando a su vez explicaciones a los usuarios sobre su razonamiento. Antes del desarrollo de MYCIN (mediados de los 70), se criticaba a la Inteligencia Artificial por resolver únicamente problemas "de juguete", sin embargo, el

éxito de MYCIN demostró que la tecnología de los Sistemas Expertos estaba suficientemente madura como para salir de los laboratorios y entrar en el mundo comercial. MYCIN es, en definitiva, un sistema de diagnóstico y prescripción en medicina, altamente especializado, diseñado para ayudar a los médicos a tratar con infecciones de meningitis.

Por otra parte, el campo del diagnóstico abarca otras aplicaciones además de las médicas (si bien pueden ser estas últimas las más conocidas). En este caso se trata de fallos, averías o anomalías que se producen generalmente en un ente.

En la actualidad existen una gran cantidad de sistemas expertos utilizados en diversos ámbitos. Algunos ejemplos de sistemas expertos que actualmente existen o que se encuentran en desarrollo son:

- DIAG: obtiene el diagnóstico de anomalías craneofaciales [Gue97],
- REDSIEX: una red cooperativa de sistemas expertos [Gon98],
- COOPERATIVE: un shell para el desarrollo de sistemas basados en el conocimiento utilizando la solución cooperativa de problemas distribuidos [Rod99],
- SETI: realiza el diagnóstico de un alumno para ayudar al tutor a llevar a cabo su tarea de tutoría individualizada [Cab00],
- YOTANPOX: realiza el diagnóstico de enfermedades cardiovasculares [Ros00],
- INTERMED: una red cooperativa de sistemas expertos para la consulta e interconsulta médica [Gon00],
- SEGBS: sistema experto de diagnóstico médico del síndrome de Guillian Bare [Car02],
- PARFAC: sistema experto para la realización del diagnóstico de parálisis facial con electromiografía: PARFAC [Rui04],
- DEPIAN: detecta y previene intrusiones de red basado en anomalías [Mar05],
- AUDI EXPERT: ayuda al auditor en las fases de elección de opinión y emisión de informes de auditoría [San95],
- ENDODIAG II, un sistema de apoyo a la enseñanza diagnóstica en Endodoncia [Agos05],
- ADICORP: con su sistema TESP para la diagnóstico de robots Puma, trabaja además en proyectos de Visión Artificial,
- TROPICAIID: selecciona un conjunto de posibles diagnósticos a partir del análisis del cuadro médico y propone un tratamiento óptimo para el caso concreto,
- PROSPECTOR: evalúa emplazamientos geológicos; es capaz de deducir una gran parte del depósito de pérfiro de molibdeno en el estado de Washington (yacimiento que los geólogos no habían sabido detectar hasta entonces, y que esta valorado en cien millones de dólares),

- LABEIN (del Laboratorio de Ensayos e Investigaciones Industriales, en España): diseña motores eléctricos mediante la aplicación de las tecnologías clásicas de Sistemas Expertos a los sistemas de CAD/CAE de diseño y análisis,
- DELTA: ayuda a los mecánicos en el diagnóstico y reparación de locomotoras diesel y eléctricas. Este sistema no solo da consejos expertos, sino que también presenta informaciones por medio de un reproductor de video,
- STEAMER: enseña a los oficiales de la armada los problemas de funcionamiento de una planta de propulsión a vapor, como las que impulsan a ciertos barcos,
- Eolo CN-235: ofrece un curso específico para pilotos y técnicos de mantenimiento, a todos los compradores del avión CN- 235. Es un sistema de enseñanza interactivo que integra gráficos, texto y video,
- GUIDON: utilizado por las Facultades de Medicina para formar a los médicos en la realización de consultas. GUIDON viene a ser una reorganización de MYCIN con intenciones educativas, por esto, tiene la ventaja adicional de disponer de toda la base de conocimientos de MYCIN además de la experiencia acumulada,
- PUFF: estudia la función pulmonar,
- HERSAY: intenta identificar la palabra hablada,
- CASHVALUE: evalúa proyectos de inversión,
- VATIA: asesora acerca del impuesto sobre el valor añadido o IVA,.
- COACH (Cognitive Adaptive Computer Help): permite crear ayuda personalizada al usuario. Es un observador de las acciones del usuario que está aprendiendo a operar un ambiente, y en base a ellas construye un modelo adaptativo del usuario,
- Palladian Operations Advisor (de Palladian Software Inc. en USA y diseñado específicamente para la dirección de la producción): analiza el estado de la combinación de productos para mantener la mayor eficacia y rentabilidad posible de las operaciones,
- Sistema Experto de Diagnóstico de fallas y ayuda a la decisión [Bre92],
- Sistema Experto para detección de fallas en máquinas de polietileno [Bre94],
- Varios sistemas expertos en el dominio de la auditoria (auditoria externa, auditoria interna y auditoria informática) [San95].
- Diseño y desarrollo de un SE auxiliar en el diagnóstico temprano de cáncer en niños [Rod01],
- Desarrollo de un sistema experto fuzzy para la predicción de “macroinvertebrate taxa” [Adr01],
- Sistema asistente de diagnóstico de enfermedades cardiovasculares a través de síntomas [Ovi03],
- Desarrollo de un sistema experto para diferenciar los dolores de cabeza de la migraña [Kop04],
- Sistema experto para la evaluación de campos visuales automáticos del proceso de descomprensión neuronal óptico [Fel04],

- Sistema de soporte a la decisión clínica en psiquiatría en la información de la edad [Kot04],
- Desarrollo de biomarcadores basados en estereotipos metabólicos dependientes de la dieta: producto resultante en el desarrollo del sistema experto basado en la clasificación de modelos en estudios metabólicos [Shi04],
- Desarrollo y validación de una lista de reglas de relación estructura-actividad para ser usada en sistemas expertos para la predicción de las propiedades sensitivas de químicos [Ger04],
- Manejo de la interacción en el razonamiento de reglas de producción fuzzy [Yeu04],
- Sistema de soporte a la decisión médica y el concepto de contexto [Kar04],
- Métodos para la multicategoría de diagnóstico de cáncer de datos de genes [Sta04],
- Sistema experto médico desarrollado en J. MD, un shell de sistema experto basado en Java: aplicación en laboratorios clínicos [Van04],
- Detección asistida por computadora de nódulos pulmonares [Mar05],
- Sistemas expertos para guiar una población de pacientes de cuidado primario para dejar de fumar, comer saludable, prevenir cáncer de piel y recibir mamografías regulares [Pro05],
- Soporte a la decisión asistido por computadora para el diagnóstico y tratamiento de enfermedades infecciosas en unidades de cuidado intensivo [Sch05],
- Inteligencia artificial- la base de conocimiento aplicada a la nefrología [San05],
- La enfermedad de Alzheimer es sustancialmente prevenible en los Estados Unidos: reseña de factores de riesgo, terapia y los prospectos para un sistema experto [Jan05],
- La validación del sistema experto EASE [Che05],
- Comparación de medidas con los datos por el sistema experto EASE [Hug05],
- Desarrollo del modelo EASE [Tick05],
- Valuación de los riesgos de la salud en gente adulta usando un sistema experto: un estudio piloto [Ili05],
- y muchos más que pueden ser consultados en:
<http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?CMD=search&DB=pubmed>.

Metodologías relacionadas con el trabajo de investigación

La metodología de este trabajo está relacionada con:

- La integración de las aproximaciones DSBC y DSOA, introducido por [Constantinides y Errad, 2000]. Los asuntos y requisitos descritos en este trabajo son contemplados en el modelo arquitectónico propuesto, ya que se aglutinan estas dos aproximaciones, obteniendo las ventajas de cada una de ellas, al definir los elementos arquitectónicos mediante sus aspectos.

- La definición de un modelo de alto nivel de abstracción, independiente de la tecnología, contemplado en la MDA [Model Driven Architectural], un enfoque propuesto por la OMG [Object Management Group] para la integración e interoperabilidad de los sistemas, que propone la definición de modelos de alto nivel de abstracción, independientes de cualquier hardware, sistema operativo o plataforma que finalmente lo soporte (Modelo Independiente de Plataforma o PIM). En este trabajo, se ha seguido esta línea con un enfoque a los sistemas expertos.
- La detección de componentes basada en la descomposición funcional del problema, es decir del sistema, compatible con The Architecture Based Design Method (ABD) [Bachman, 2000]. ABD es una metodología propuesta por The Software Engineering Institute of The Carnegie Mellon University para diseñar arquitecturas software de un dominio de aplicación. Esta metodología se ha aplicado para la construcción del modelo propuesto.
- El trabajo realizado por [Garlan, 2001], ya que es un punto de referencia muy importante para establecer los elementos de un sistema software complejo, en el cual se introducen conceptos como componente, conector, sistema, puertos de entrada y de salida. Dichos conceptos han sido contemplados en el modelo propuesto.
- Los trabajos de [Andrade y Fiadeiro, 1999] acerca del concepto de contrato, ya que se han definido los conectores del modelo arquitectónico "DiagMed" como una extensión del contrato propuesto en su trabajo. Dicha extensión se produce debido a la incorporación del concepto de coreografía en el conector, representado en este trabajo por el protocolo del aspecto de coordinación del conector.
- La existencia de la gran variedad de lenguajes de definición de arquitecturas (LDA), donde cada uno de ellos tiene ventajas y desventajas., tal y como presenta el estudio realizado en [Medvidovic y Taylor, 2000]. Uno de los lenguajes que más se aproxima al LDA de PRISMA es el propuesto por [Loques et al., 2000] en su modelo R-RIO, el cual tiene capacidades de reconfiguración al igual que PRISMA, sin embargo no incorpora la noción de aspecto.
- Los sistemas basados en reglas y los sistemas basados en el conocimiento, par de categorías de las metodologías de los sistemas expertos descritas por [Liao, 2005]. Este sistema utiliza conocimiento representado en forma de reglas (cláusulas de Horn).
- Las arquitecturas de los sistemas expertos conformadas por componentes, mencionadas en [Giarratano y Riley, 2004], ya que en el modelo propuesto se definen elementos arquitectónicos en el nivel de tipos, permitiendo que en el nivel de configuración se especifique la topología del modelo arquitectónico.
- El estudio realizado por [Liao, 2003] referente a los métodos tradicionales de desarrollo de los sistemas expertos, que han tratado el conocimiento y el procedimiento como componentes independientes. El modelo arquitectónico propuesto contempla esta separación al considerar un componente que contiene el

conocimiento del dominio y otro componente que realiza el proceso de inferencia del diagnóstico.

- Las aplicaciones de los sistemas expertos en un dominio específico. [Liao, 2005] señala que las metodologías de los sistemas expertos desarrollan aplicaciones con una orientación del dominio del problema. En este trabajo se consideró un caso de estudio dentro del dominio de la medicina.

CAPÍTULO I

MARCO CONCEPTUAL

1.1 El Modelo PRISMA

El modelo arquitectónico PRISMA [Per03] integra dos aproximaciones: el Desarrollo de Software Basado en Componentes (DSBC) y el Desarrollo de Software Orientado a Aspectos (DSOA). Esta integración se consigue definiendo los elementos arquitectónicos mediante aspectos [Ali05]. De esta forma, el modelo PRISMA, además de definir los elementos arquitectónicos básicos y especificar su sintaxis y semántica, también especifica los aspectos que cubren las propiedades necesarias de cada uno de ellos. Por ello un elemento arquitectónico de PRISMA puede ser analizado desde dos vistas diferentes: interna y externa. La vista interna muestra al elemento arquitectónico como un prisma, de forma que cada lado del prisma corresponde a un aspecto. Mientras que la vista externa encapsula la funcionalidad del elemento arquitectónico teniendo sólo la visibilidad de los servicios que éste publica a través de sus puertos.

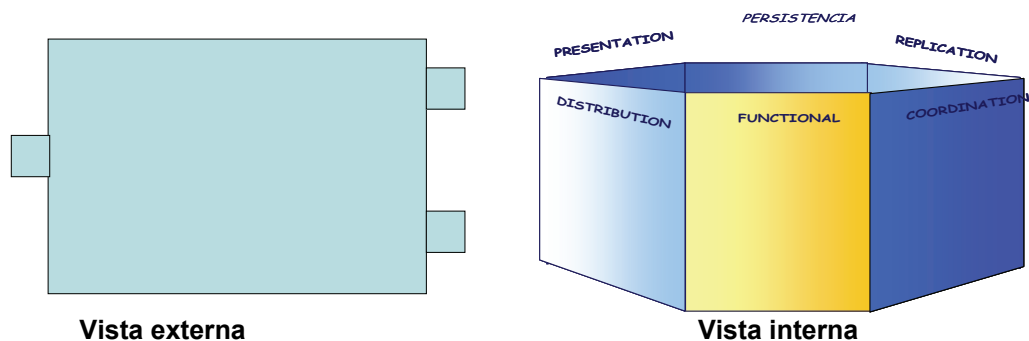


Fig. 1 Vistas de un elemento arquitectónico PRISMA

El modelo PRISMA consta de tres tipos de elementos arquitectónicos: componentes, conectores y sistemas. Un componente captura la funcionalidad del sistema, mientras que un conector actúa como coordinador entre otros elementos arquitectónicos. Un sistema es un elemento arquitectónico de mayor granularidad que permite encapsular un conjunto de componentes, conectores y otros sistemas, correctamente conectados entre sí.

Un elemento arquitectónico PRISMA está formado por un conjunto de aspectos de diferente tipo, las relaciones de entretejido entre estos aspectos (*weavings*) y uno o más puertos.

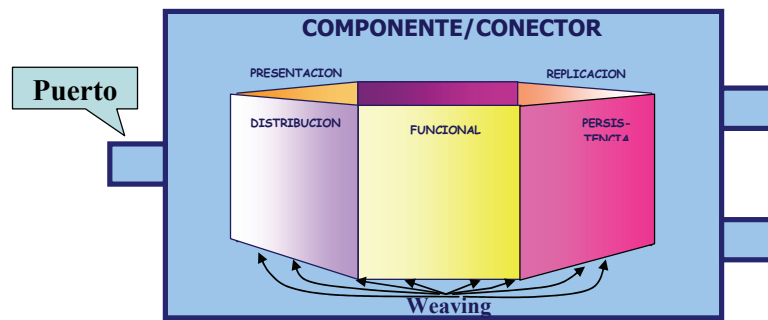


Fig. 2 Elemento arquitectónico PRISMA

Un aspecto PRISMA es una propiedad de interés del sistema (*concern*) que es común y compartido por el conjunto de tipos del sistema (*crosscutting concerns*). Los tipos de aspectos (funcional, coordinación, distribución, etc.) que forman un elemento arquitectónico varían dependiendo del sistema software.

El *weaving* establece las sincronizaciones necesarias entre los aspectos que conforman un elemento arquitectónico. El *weaving* define que la ejecución de un servicio de un aspecto puede generar la invocación de un servicio de otro aspecto.

Los puertos representan los puntos de interacción entre los elementos arquitectónicos.

Existen dos tipos de relaciones para interconectar los elementos arquitectónicos: *attachments* y *bindings*. Los *attachments* establecen el canal de comunicación entre el puerto de un componente y el puerto de un conector., mientras que los *bindings* establecen la comunicación entre un puerto de un sistema y un puerto de un elemento arquitectónico de los que encapsula. De este modo, los *bindings* permiten mantener un enlace entre los distintos niveles de granularidad de los elementos arquitectónicos que forman el modelo.

PRISMA especifica la arquitectura en dos niveles: el de definición de tipos y el de configuración. En el nivel de tipos se definen los tipos de la arquitectura: interfaces, aspectos, componentes, conectores y sistemas, los cuales son guardados en una librería para su reutilización. A nivel de configuración se especifican los modelos arquitectónicos de un determinado sistema software. Esta diferenciación proporciona importantes ventajas, ya que permite gestionar de forma independiente los tipos de elementos y las topologías específicas de cada sistema, obteniendo de este modo un mayor nivel de reutilización y un mejor mantenimiento de las librerías de tipos.

1.2 Los Sistemas Expertos en el Diagnóstico Médico

Nuestros días se caracterizan por un explosivo, colosal y omnipresente desarrollo de la técnica, y su aplicación cada vez más extensa a todos los ámbitos de la vida humana, en particular, la ciencia biomédica y la práctica clínica. Sin duda alguna, el más significativo desarrollo tecnológico durante el último siglo, ha sido la construcción de computadoras de finalidades generales, capaces de hacer cosas que en el hombre se consideran como comportamiento inteligente.

En la actualidad, la literatura médica recoge con alta frecuencia trabajos sobre el pensamiento médico en informática, debido a que la informática es la rama más relacionada con el pensamiento médico.

La integración de la medicina con la informática, ha permitido extender la aplicación de las computadoras a los servicios administrativos y de apoyo, la dirección, la investigación, el diagnóstico y el tratamiento, sin dejar de mencionar la educación.

Hay que tener en cuenta nuevas tecnologías surgidas dentro de la computación y que rápidamente han sido aplicadas a la Medicina. Si bien es cierto que la computadora tiene gran capacidad de cálculo, velocidad y exactitud, esta claro que una computadora no puede sustituir al médico. Sólo éste es capaz de razonar lógicamente y mezclar la razón con la intuición, la ética, lo afectivo y la experiencia, algo que una máquina no puede hacer. No puede mantener el aspecto más importante: la relación médico-paciente.

Vista la informática como una herramienta, no es despreciable la ayuda que puede ofrecer al médico. Cada día serán más fuertes los lazos de integración, pero sólo esa integración será beneficiosa si el médico ve a la computadora como un instrumento de apoyo a su trabajo, no como un posible afán de lucro, no como un instrumento de reafirmación de su prestigio en su entorno académico y hospitalario, no como un posible sustituto de su persona.

Los estudiantes y los profesionales de la salud deberán recibir una formación que los enseñe a usar la tecnología y no hacerse dependiente de ella.

Por su precisión, velocidad de operación y otras cualidades, las computadoras podrán ayudar al médico a realizar complejos y precisos procesos, incluso, intervenciones quirúrgicas, pero siempre dirigido y controlado por el hombre. No debe asustar el uso, sino el abuso sin control.

La influencia positiva en el desarrollo del pensamiento científico en medicina, es que se obliga al médico a buscar formas de descripción de los procesos biológicos, con la formalización matemática, mas objetiva y precisa que la intuición. Por otra parte, el uso

de la informática, herramienta del pensamiento médico, con la gran velocidad de desarrollo de los cálculos matemáticos, ofrece soluciones en el campo de probabilidades que hacen objetivizar realidades en los procesos biológicos, que por otros métodos quedarían inaparente, o sólo a la vista de los clínicos de mayor intuición o agudeza del llamado "ojo clínico".

De lo anterior, podemos deducir que existe una emulación entre el pensamiento clínico simple y el simulado por la informática, que al ser más rápido y objetivo hace que la intuición del pensamiento trate de encontrar formalizaciones del pensamiento, que logren encontrar soluciones mucho más objetivas y evidentes para la solución de los problemas médicos.

La informática es una simulación del pensamiento o juicio clínico, ésta lo hace de una forma más rápida, pero sin la creatividad y el profundo conocimiento e inteligencia del humano.

No obstante, el médico posee en su práctica una serie de datos imprecisos e incompletos y en esa incertidumbre discurre, teniendo como mayores armas, su inteligencia y experiencia. Es por ello que en ayuda de estas últimas crea nuevas armas, dentro de las cuales se destaca la informática.

Las características del pensamiento clínico están determinadas tanto por el arte como por ciencia; el arte está representado por la intuición y el instinto y éstas no pueden ser remplazadas por un conjunto de reglas o formalizaciones matemáticas; a esto se le llama juicio clínico o pensamiento médico, que no es más que los procesos en la toma de decisiones en ausencia de leyes o reglas explícitas.

El procedimiento clínico abarca desde la primera entrevista con el paciente hasta la decisión de un diagnóstico tentativo. Esta es, sin duda, la parte donde se necesita el volumen mayor de conocimiento médico, ya que estamos preparados para tratar con una o varias enfermedades de las miles de enfermedades o estados clínicos que a su vez tienen muchas manifestaciones que no necesariamente representan enfermedad.

La entrevista, el examen físico y los exámenes complementarios reducen el enorme conjunto de posibilidades a un pequeño grupo de diagnósticos diferenciales.

Como un lógico proceso de desarrollo, la Medicina ha ido asimilando la introducción de las computadoras para agilizar y mejorar los procesos de apoyo médico, teniendo una gran influencia, la que sigue aumentando más cada día con la introducción de la Inteligencia Artificial en la vigilancia del paciente con complejos equipos biomédicos, realización de procesamiento voluminoso de información para la toma de decisiones y

muchas otras aplicaciones. Podemos hablar entonces del surgimiento de la Informática Médica, que comprende una amplia gama de cuestiones de la organización y del uso de la información biomédica. El objetivo de la Informática Médica es reforzar y mejorar la toma de decisiones médicas y la atención al paciente.

La informática como primera ayuda ofreció los sistemas estadísticos y de probabilidades computarizados para los diagnósticos, o sea, se calculaba la probabilidad de cada signo en el diagnóstico y se determinaba el más indicado.

Si tenemos en cuenta que la informática depende del suministro de datos o informaciones previamente elaboradas y la limitación temporal del uso de las computadoras, obtendremos como resultado que, el impulso de la informática, depende del número de científicos que se interesen por ésta, así como por la calidad de los científicos integrados a ella.

La característica principal de la información o de los lotes de información que se suministran a la computadora para los procesos de la informática, es la exactitud en la descripción de la realidad objetiva. Esto requiere un esfuerzo de los científicos para desarrollar el pensamiento médico en aras de una mejor descripción de la realidad objetiva, donde se trata de sustituir la intuición y las inferencias por procesos lógicos o matemáticos.

La cantidad y la calidad de los lotes de información suministrados a la computadora por los científicos, que llevan el pensamiento médico, el juicio clínico y los procesos diagnósticos, determinan la calidad y el desarrollo del producto informático, ya sea un programa de diagnósticos o pronósticos o un sistema experto. En los sistemas expertos es permitida la interacción del médico con la computadora.

Un sistema experto es un programa de computadoras que permite simular el comportamiento de un especialista humano frente a un problema de su competencia en un determinado campo o materia. Intentan codificar los conocimientos y reglas de decisión de los especialistas, de manera que los clientes puedan aprovechar estas pericias al tomar sus propias decisiones. Estos sistemas están orientados esencialmente a ciertos tipos de trabajos limitados conceptualmente a una serie de acciones o decisiones.

El sistema experto es la forma más demostrativa de la informática; por concepto es un programa de computación basado en el conocimiento, que emula con un experto humano en la resolución de un problema significativo en un dominio específico. Un sistema experto puede ser considerado como un programa inteligente.

El sistema experto generalmente se utiliza en soluciones complejas a problemas específicos, las características de los procesos que realizan simulan al del juicio clínico y utilizan la esquematización matemática del pensamiento médico. Estos sistemas se utilizan en apoyo a las decisiones médicas, de tipo diagnósticas o terapéuticas y no deben ser asumidos como sustitutos del médico.

La necesidad del uso de estos sistemas se debe al aumento constante de la cantidad de conocimientos y datos a considerar en una decisión, debido al gran avance tecnológico y a la intensa investigación clínica. El éxito de estos sistemas expertos se debe a la capacidad de las computadoras de manejar una gran cantidad de conocimientos y a la posibilidad de que en él participen un grupo de expertos en un tema, pero enfocados por especialidades diversas.

Probablemente en pocos años, el uso de los sistemas expertos se haya extendido a todas las actividades humanas complejas en las que interviene gran cantidad de datos y variables. Analizarán y comprimirán datos para nosotros, tomarán decisiones de poca importancia y servirán como medio de apoyo para decisiones complejas o de gran trascendencia.

En la estructura de un sistema experto se pueden distinguir dos partes fundamentales: la base de conocimientos y el mecanismo de inferencias. La base de conocimientos está formada por los conocimientos que se usan para resolver los problemas que se plantean. El mecanismo de inferencia se refiere a la estrategia que usa el sistema para el manejo de los conocimientos que le permitan llegar a resultados satisfactorios. Los médicos, que son los expertos que lo construyen, deben tener bien claro la definición de las variables y las reglas utilizadas en la simulación del juicio clínico. Dentro de ellas, una de las más utilizadas es la de “si-entonces”. La mejor forma de entenderlo es con un ejemplo:

SI Hay disnea, molestia al respirar, se presenta con los esfuerzos, es progresiva
ENTONCES Hay hipótesis del aumento de volumen de las venas pulmonares.

El conocimiento médico se expresa como un conjunto de reglas en las cuales y en la parte del “si”, están expresadas las premisas de alguna situación, en tanto que en la parte de los “entonces” se encuentra la conclusión.

La estructuración del conocimiento como sistemas de si-entonces, permite ordenar el conocimiento como árboles virtuales, en los que la base se encuentra formada por las conclusiones terminales y las premisas son hojas de diferentes ramas. La complejidad del árbol resulta de la interconexión dinámica de las diferentes ramas y del número de conclusiones posibles. Los árboles de decisión, son una secuencia de pasos en los cuales el médico selecciona un camino a través de una cadena de eventos y acciones.

Los algoritmos clínicos en los cuales un árbol lógico es empleado por el médico para el diagnóstico o cuidado de un paciente de acuerdo con una estrategia predeterminada, también son ejemplos de la formalización matemática del pensamiento médico.

El diagnóstico médico es entendido como el proceso encaminado a la identificación o reconocimiento de una enfermedad sobre la base de los signos y síntomas presentes, con el apoyo de los estudios de laboratorio y gabinete. El diagnóstico médico indica todo el proceso de investigación del paciente, a partir de las observaciones y razonamientos del médico para determinar la enfermedad.

En particular, las personas que realizan diagnósticos médicos, por lo general se basan en ciertos tipos de razonamientos para llegar a un diagnóstico. Entre los más comunes se encuentran: el razonamiento progresivo o deductivo (donde se parte de la premisa de los signos y síntomas para concluir la enfermedad), el razonamiento regresivo o inductivo (que parte del hecho de que un paciente tiene cierta enfermedad para inferir los signos y síntomas), y el razonamiento diferencial (para discernir entre dos o más enfermedades parecidas para diferenciar de forma más clara cuál es la enfermedad diagnosticada).

El diagnóstico diferencial está íntimamente relacionado con la precisión diagnóstica y por lo tanto, con la capacidad para poder agrupar enfermedades relacionadas, como entidades nosológicas o síndromes. Éste se incluye en el curso natural para lograr un diagnóstico final lo más depurado posible. La capacidad diagnóstica de un buen médico está relacionada directamente con su capacidad para poder ver todas las posibilidades diagnósticas para un cierto grupo de manifestaciones clínicas (signos, síntomas, pruebas de laboratorio y gabinete) y su capacidad para discernir, sobre la base de su experiencia, cuál de todos estos padecimientos es con mayor certeza la causa del proceso mórbido. La meta del diagnóstico diferencial es “arribar” a un diagnóstico específico, sobre la base de la exclusión razonada de otras enfermedades que tienen en común gran parte de los síntomas.

Es el diagnóstico, quizás, el más controvertido de los sectores de aplicación de las computadoras en la medicina, por las implicaciones éticas que puede traer. La introducción de sistemas expertos para el diagnóstico ha planteado la interrogante: ¿Nos sustituirá la computadora algún día?. Uno de los problemas de estos sistemas expertos en el diagnóstico es que no toman en cuenta que una persona puede tener más de una enfermedad o que el paciente puede estar fingiendo.

CAPÍTULO II

DESCRIPCIÓN DEL PROBLEMA

PARA EL CASO DE ESTUDIO

2.1 Descripción del problema

Dado el auge de la automatización en los últimos tiempos, surge la necesidad de diseñar arquitecturas para sistemas a un mayor nivel de abstracción, de forma que sea posible reutilizar dichos diseños para otro tipo de arquitecturas y que sean capaces de evolucionar tanto en tiempo de compilación como en tiempo de ejecución de forma correcta y preservando los requisitos de calidad del sistema.

Uno de los campos de aplicación del enfoque presentado es el de los sistemas que realizan tareas de diagnóstico en entornos médicos. Las peculiaridades inherentes para la toma de decisiones en este tipo de sistemas hacen adecuado abordar de manera innovadora soluciones para la incorporación de aspectos dinámicos en las arquitecturas software construidas para dichos sistemas.

Se hará uso del enfoque anterior para establecer un marco arquitectónico único que recoja los requisitos comunes de los distintos sistemas y permita la adaptación dinámica a cada uno de ellos.

Para conseguir estos objetivos será necesaria la identificación de aquellos componentes, aspectos e interfaces del sistema de diagnóstico que puedan ser susceptibles de variación y por tanto impliquen una modificación en el modelo arquitectónico empleado.

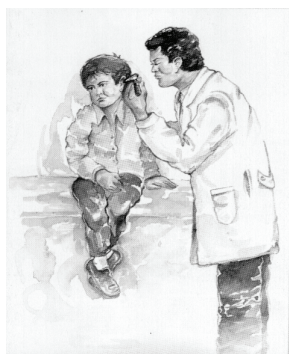
Una vez identificados los componentes, aspectos e interfaces (en ese orden), se procederá a la especificación formal haciendo uso del lenguaje de definición de arquitecturas (LDA) de PRISMA de los distintas interfaces, aspectos y elementos arquitectónicos (en dicho orden) que forman al modelo. Posteriormente, utilizando dicho lenguaje, se definirán las instancias y con ello se especificará la topología del sistema de diagnóstico médico.

El dominio de aplicación de este sistema para tareas de tipo diagnóstico es el diagnóstico médico, el cual, al igual que cualquier dominio de aplicación que requiere de la pericia humana para la solución de problemas, se convierte en un escenario probable para la aplicación exitosa de un sistema experto. Dado que este tipo de sistemas requiere de un dominio de aplicación muy específico, se ha seleccionado el campo de enfermedades infecciosas infantiles. Ente las más comunes o frecuentes de ellas se encuentran: la

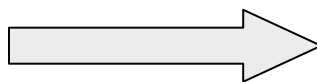
rubéola, el sarampión, la roséola, la escarlatina, la varicela, el dengue, la tifoidea, la parotiditis viral o paperas, la parotiditis bacteriana, el crup espasmódico o laringitis estridulosa, la bronquiolitis infecciosa aguda, la neumonía, la otitis viral y la otitis bacteriana.

Dicho sistema contempla en su base de conocimientos un conjunto de reglas conformadas por los signos y síntomas de dichas enfermedades que manifiesta el paciente a diagnosticar; las cuales tienen un grado de certeza avalado por un pediatra, esto es, por un experto en el tema¹. Además el sistema exhibe tres comportamientos o estrategias de razonamiento, que son unos de los más adecuados y utilizados para resolver tareas de tipo diagnóstico [Jac90]: razonamiento de reglas hacia adelante o forward, razonamiento de reglas hacia atrás o backward y razonamiento diferencial.

El sistema propuesto realiza el diagnóstico médico, donde el usuario (llámese médico y/o laboratorista) introduce datos de entrada del paciente al sistema, consistente en signos y síntomas (incluyendo estudios de laboratorio y gabinete), obteniendo como resultado el diagnóstico o enfermedad del paciente así como el tratamiento correspondiente para solucionar el problema detectado. El diagnóstico final se realiza a través de la realización de dos diagnósticos, el tipo clínico y el de laboratorio, con el objeto de que finalmente se genere un diagnóstico integral, con la aportación de ambos.. Así mismo se ha incluido el registro del paciente a través de introducir sus datos personales. Por ello el sistema ofrece al usuario las opciones de: registrar datos del paciente, visualizar datos del paciente, realizar el diagnóstico clínico, realizar el diagnóstico de laboratorio, obtener la terapia indicada al caso, así como visualizar los resultados del diagnóstico realizado.



Médico inspeccionando al paciente para obtener los signos y síntomas



Médico haciendo uso del sistema diagnosticador

Fig. 3 Vista del médico como usuario del sistema de diagnóstico médico

¹ DE LA ROSA PALACIOS, Jéssica. Comunicación personal, diciembre de 2004



Fig. 4 Entradas y salidas del sistema de diagnóstico médico

Cabe mencionar que para un paciente dado, es necesario primero realizar el diagnóstico clínico, seguido por el diagnóstico de laboratorio y finalmente obtener la terapia asociada. Para ello el sistema condiciona a dar la terapia indicada a dos casos:

- que el diagnóstico clínico y el de laboratorio correspondan a la misma enfermedad y
- que ambos diagnósticos hayan sido obtenidos con grados de certeza superiores al 0.85

Analizando este sistema se contemplan dos situaciones importantes: una de ellas es la toma de decisiones que realiza el propio sistema y la otra es sobre su uso, ya que éste debe ser también amigable. Dentro de este último punto se contempló que al usuario se le hiciera el mínimo de preguntas para llegar a la solución del problema.

La primera de ellas es referente a la adquisición de las posibles enfermedades y que se describe a continuación. Para ello, cabe reiterar que el MOTOR DE INFERENCIA obtiene de la BASE DE CONOCIMIENTOS las posibles enfermedades del paciente con el fin de poder reducir las opciones del árbol de decisión y enviarlas a la BASE DE CONOCIMIENTOS, quien adquiere el diagnóstico clínico y de laboratorio al solicitarle al USUARIO CLINICO o al USUARIO LABORATORIO los signos y síntomas de granos fino o bien los estudios de laboratorio, respectivamente, de sólo estas enfermedades y no de todas las contenidas en la BASE DE CONOCIMIENTOS.

La segunda hace mención a la obtención de los signos y síntomas, y que se describen en el aspecto funcional del componente BASE DE CONOCIMIENTOS. Ahora bien, es importante resaltar que existen signos y síntomas tanto de grano grueso como de grano fino y los resultados de los estudios de laboratorio que son comunes a varias enfermedades, por lo que para que el sistema no pregunte al usuario más de una vez sobre dichos datos, se realizaron operaciones de transacción.

2.2 Escenarios, casos de uso y diagramas de secuencia

En la definición de la arquitectura del modelo propuesto, la identificación de elementos arquitectónicos tiene como objetivo primordial la detección de los componentes del sistema, para posteriormente detectar los conectores que permiten su comunicación. Para ello, se parte de la especificación de requisitos resultante de la etapa de obtención de requisitos.

En esta etapa es primordial tener presente la definición de componente, ya que en la actualidad no existe un consenso sobre qué es un componente. Existen dos tendencias, una es más orientada a implementación y otra es más genérica. La primera recoge definiciones relacionadas con el hecho de que una componente es un paquete de código [Sou99], mientras que la segunda define un componente como un artefacto que ha sido desarrollado específicamente para ser reutilizado.

De estas dos tendencias, en este trabajo se eligió la segunda, ya que la hace independiente a la solución del problema, manteniendo el alto nivel de abstracción. Esta segunda definición permite que un componente sea tanto un caso de uso como una clase o cualquier otro elemento que surja durante el proceso de desarrollo, permitiendo que la identificación de componentes se pueda basar en diferentes criterios:

- los criterios de casos de uso, entre otros; que se citan en el trabajo realizado por [Ira00]
- los criterios de entidad (clase o tipo abstracto de datos) [Jac97] y
- el de escena [Nor98].

El concepto de escena aparece en el trabajo de [Nor98]. Este trabajo considera la escena como un criterio para la detección de componentes en la especificación de requisitos. El criterio consiste en que cada escena que se detecte en la especificación de requisitos se corresponde con un componente del sistema.

Los elementos arquitectónicos PRISMA [Per03] se obtienen identificando escenas funcionales del sistema y asignando a cada elemento una escena funcional, dependiendo de si la escena es simple o compleja, el elemento será un componente (componente simple) o un sistema (componente complejo), respectivamente.

El concepto de escena es definido con una enfoque PRISMA por [Per03b] al considerar que “Una escena se caracteriza por las tareas que se desempeñan en la actividad siguiendo un determinado protocolo, los actores que las realizan y el espacio virtual o físico donde se desarrolla”

Los casos de uso constituyen una división del sistema basada en la funcionalidad. En [Jac97, Ber99, Sou99] se propone que aquellas clases que estén relacionadas con el caso de uso, formaran parte del componente asociado al caso de uso. El diagrama de casos de uso muestra las distintas operaciones que se esperan del sistema y cómo se relaciona con su entorno (usuario).

Un diagrama de secuencias muestra la interacción de un conjunto de objetos de una aplicación a través del tiempo. Esta descripción es importante porque puede dar detalle a los casos de uso, aclarándolos al nivel de mensajes entre los objetos existentes, como también muestra el uso de los mensajes de las clases diseñadas en el contexto de una operación. Se puede crear un diagrama de secuencias por cada caso de uso. Con los diagramas de secuencia se puede modelar la implementación del escenario.

En este trabajo se realizó un análisis exhaustivo del dominio del problema (diagnóstico médico) para conocer cuáles son los requisitos que el producto final ha de satisfacer. Estos requisitos son especificados mediante los casos de uso. Ahora bien, dado que una escena da soporte a un caso de uso, en este trabajo se consideró conveniente construir un diagrama de caso de uso que se corresponda a cada escenario, así como crear un diagrama de secuencias para cada escenario, describiéndose el proceso general del sistema con una vista externa de caja negra, por lo que únicamente se establece la interacción entre el sistema y el usuario. Finalmente, y con el fin de describir cómo se realiza cada caso de uso, se crearon los diagramas de secuencias de las realizaciones de los casos de uso, ofreciendo una vista del funcionamiento del sistema (ver apartado 3.2 de este documento).

2.2.1 Casos de Uso

El diagrama que presenta los casos de uso para cada escenario descrito anteriormente y que para la **gestión de diagnóstico** realiza el sistema, puede tomar la siguiente forma:

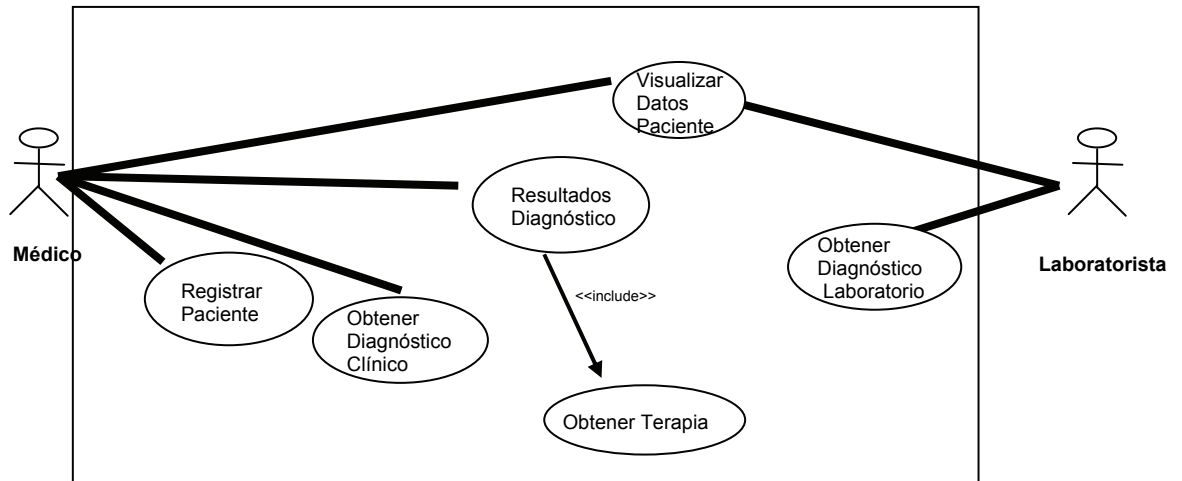


Fig. 5 Casos de uso del sistema de diagnóstico médico

Caso de Uso: Registrar paciente

- **Descripción.**- el médico puede invocar este caso de uso al haber seleccionado la opción Registrar Paciente que el sistema le muestra en una interfaz gráfica. Se introduce en el sistema los datos personales del paciente que se va a registrar.

- **Flujo de eventos:**

- **Flujo básico:**

- 1.- Ante la aparición de la correspondiente interfaz gráfica a este caso de uso, el médico introduce en el sistema los datos de registro del paciente (identificador, DNI, nombre, dirección, teléfono, email, sexo y fecha de nacimiento).
 - 2.- Una vez introducidos todos los datos del paciente, el médico podrá guardarlos pulsando el botón de aceptar, en cuyo caso se almacenará en una base de datos, quedando así dado de alta el paciente.
 - 3.- El médico deberá cerrar dicha interfaz gráfica para continuar con el uso del sistema.

- **Flujo alternativo:** No hay.

- **Precondiciones:**

- El médico ha realizado correctamente su acceso al sistema mediante su nombre de usuario y su contraseña.
 - El laboratorista no podrá invocar este caso de uso.

- **Postcondiciones:**

- En caso de haberse dado de alta a un paciente, los datos del mismo quedarán almacenados en una base de datos.

- **Puntos de extensión:** No hay.

Caso de Uso: Visualizar paciente

- **Descripción.-** el médico o el laboratorista pueden invocar este caso de uso al haber seleccionado la opción Visualizar Paciente que el sistema les muestra en una interfaz gráfica. Se deberá introducir el identificador del paciente del cual se desea consultar (datos personales).

- **Flujo de eventos:**

Flujo básico:

- 1.- Ante la aparición de la correspondiente interfaz gráfica a este caso de uso, el médico o el laboratorista introducen en el sistema el identificador del paciente y pulsan el botón de aceptar.
- 2.- El sistema muestra los datos del paciente según la base de datos.
- 3.- El médico o laboratorista deberá cerrar dicha interfaz gráfica para continuar con el uso del sistema.

Flujo alternativo:

- Si en el paso 1 el paciente no está dado de alta, ante el fracaso de la búsqueda, el usuario tendrá que cerrar la interfaz gráfica, y sólo el médico podrá realizar el caso de uso Registrar Paciente para proceder a su alta.

- **Precondiciones:**

- El médico o laboratorista ha realizado correctamente su acceso al sistema mediante su nombre de usuario y su contraseña.
- El paciente debe anteriormente haber sido registrado a través del caso de uso Registrar Paciente.

- **Postcondiciones:**

- La ejecución de este caso de uso no modifica la base de datos.

- **Puntos de extensión:**

- En el paso 1, en caso de que no exista el registro del paciente, sólo el médico podrá invocar el caso de uso Registrar Paciente para introducir un nuevo paciente en la base de datos del sistema.

Caso de uso: Realizar diagnóstico clínico

- **Descripción.-** el médico puede invocar este caso de uso al haber seleccionado la opción Realizar Diagnóstico Clínico que el sistema le muestra en una interfaz gráfica. Éste deberá introducir el identificador del paciente del cual se desea obtener el diagnóstico clínico, así como cierta información que el sistema le solicita para inferir el objetivo.

- **Flujo de eventos:**

Flujo básico:

- 1.- Ante la aparición de la correspondiente interfaz gráfica a este caso de uso, el médico introduce en el sistema el identificador del paciente así como su propio nombre.
- 2.- El sistema muestra; uno a uno; todos los signos y síntomas de grano grueso del paciente para que el médico le responda si están o no presentes mediante la pulsación de los botones si o no.
- 3.- A continuación se repite el paso 2 pero para los signos y síntomas de grano fino.
- 4.- El sistema muestra la sugerencia de si se recomienda o no realizar estudios de laboratorio al paciente.
- 5.- Finalmente el médico podrá guardar la información obtenida al pulsar el botón aceptar, en cuyo caso realizará el sistema el diagnóstico clínico del paciente y lo almacenará en la base de datos.

Flujo alternativo:

- Si en el paso 1 el paciente no está dado de alta, ante el fracaso de la búsqueda, el médico tendrá que cerrar la interfaz gráfica, y realizar el caso de uso Registrar Paciente para proceder a su alta.

• **Precondiciones:**

- El médico ha realizado correctamente su acceso al sistema mediante su nombre de usuario y su contraseña.
- El paciente debe anteriormente haber sido registrado a través del caso de uso Registrar Paciente.
- El médico deberá responder al sistema a través de la pulsación de los botones si y/o no de todos y cada uno de los signos y síntomas de grano grueso y de grano fino del paciente.

• **Postcondiciones:**

- En caso de haber ingresado toda la información solicitada por el sistema, el diagnóstico clínico es realizado y queda almacenado en una base de datos.

• **Puntos de extensión:**

- En el paso 1, en caso de que no exista el registro del paciente, el médico podrá invocar el caso de uso Registrar Paciente para introducir un nuevo paciente en la base de datos del sistema.

Caso de uso: Realizar diagnóstico de laboratorio

- **Descripción.-** el laboratorista puede invocar este caso de uso al haber seleccionado la opción Realizar Diagnóstico de Laboratorio que el sistema le muestra en una interfaz gráfica. Éste deberá introducir el identificador del paciente del cual se desea obtener el diagnóstico de laboratorio, así como cierta información que el sistema le solicita para inferir el objetivo.

- **Flujo de eventos:**

- Flujo básico:**

- 1.- Ante la aparición de la correspondiente interfaz gráfica a este caso de uso, el laboratorista introduce en el sistema el identificador del paciente así como su propio nombre.
 - 2.- El sistema muestra una sugerencia de estudios de laboratorio que deberá realizarse al paciente
 - 3.- A continuación el sistema muestra; uno a uno; todos los resultados de los estudios de laboratorio del paciente para que el laboratorista le responda si están o no presentes mediante la pulsación de los botones si o no.
 - 4.- Finalmente el laboratorista podrá guardar la información obtenida al pulsar el botón aceptar, en cuyo caso realizará el sistema el diagnóstico de laboratorio del paciente y lo almacenará en la base de datos.

- Flujo alternativo:**

- Si en el paso 1 el paciente no está dado de alta, ante el fracaso de la búsqueda, el laboratorio tendrá que cerrar la interfaz gráfica.

- **Precondiciones:**

- El laboratorista ha realizado correctamente su acceso al sistema mediante su nombre de usuario y su contraseña.
 - El paciente ha sido registrado a través del caso de uso Registrar Paciente.
 - El paciente ya cuenta con el diagnóstico clínico para proceder a realizar este caso de uso.
 - El laboratorista deberá responder al sistema a través de la pulsación de los botones si y/o no de todos y cada uno de los resultados del laboratorio del paciente.

- **Postcondiciones:**

- En caso de haber ingresado toda la información solicitada por el sistema, el diagnóstico de laboratorio es realizado y queda almacenado en una base de datos.

- **Puntos de extensión:** No hay.

Caso de uso: Resultados del diagnóstico

- **Descripción.**- el médico puede invocar este caso de uso al haber seleccionado la opción Resultados que el sistema les muestra en una interfaz gráfica. Se deberá introducir el identificador del paciente del cual se desea consulta.

- **Flujo de eventos:**

- Flujo básico:**

- 1.- Ante la aparición de la correspondiente interfaz gráfica a este caso de uso, el médico introduce en el sistema el identificador del paciente y pulsar el botón de aceptar.
- 2.- El sistema, tras inferir la terapia adecuada al caso, muestra los resultados del diagnóstico integral realizado al paciente según la base de datos, a saber el diagnóstico clínico y de laboratorio, la terapia y los nombres del médico y el laboratorista que atendieron al paciente para llevar a cabo dicho objetivo.
- 3.- El médico deberá cerrar dicha interfaz gráfica para poder seleccionar otro caso de uso o bien salir del sistema.

Flujo alternativo:

- Si en el paso 1 el paciente no está dado de alta, ante el fracaso de la búsqueda, el médico tendrá que cerrar la interfaz gráfica, y realizar el caso de uso Registrar Paciente para proceder a su alta.
- Si en el paso al paciente no se le ha realizado el diagnóstico clínico y/o el diagnóstico de laboratorio, el médico tendrá que cerrar la interfaz gráfica, y realizar el caso de uso Realizar Diagnóstico Clínico, y posteriormente, si es recomendado o deseado por el médico, el caso de uso Realizar Diagnóstico de Laboratorio.

• **Precondiciones:**

- El médico ha realizado correctamente su acceso al sistema mediante su nombre de usuario y su contraseña.
- El paciente debe anteriormente haber sido registrado a través del caso de uso Registrar Paciente.

• **Postcondiciones:**

- La ejecución de este caso de uso modifica la base de datos al agregarse la información de la terapia inferida por el sistema.

• **Puntos de extensión:**

- En el paso 1, en caso de que no exista el registro del paciente, el médico podrá invocar el caso de uso Registrar Paciente para introducir un nuevo paciente en la base de datos del sistema.

2.2.2 Diagramas de secuencia de los casos de uso

Los diagramas de secuencia que se corresponden a cada uno de los casos de uso arriba descritos son:

Para el caso de uso: Registrar paciente

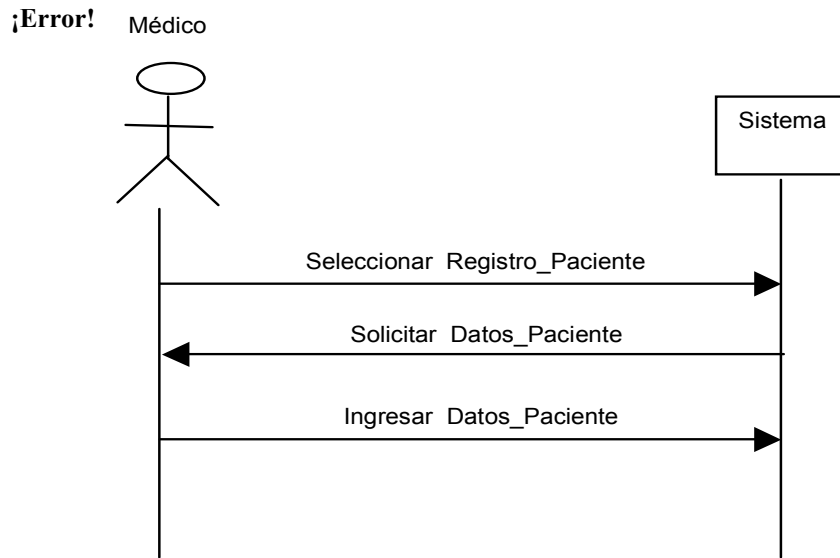


Fig. 6 Diagrama de secuencia del caso de uso: Registrar Paciente

Para el caso de uso: Visualizar paciente

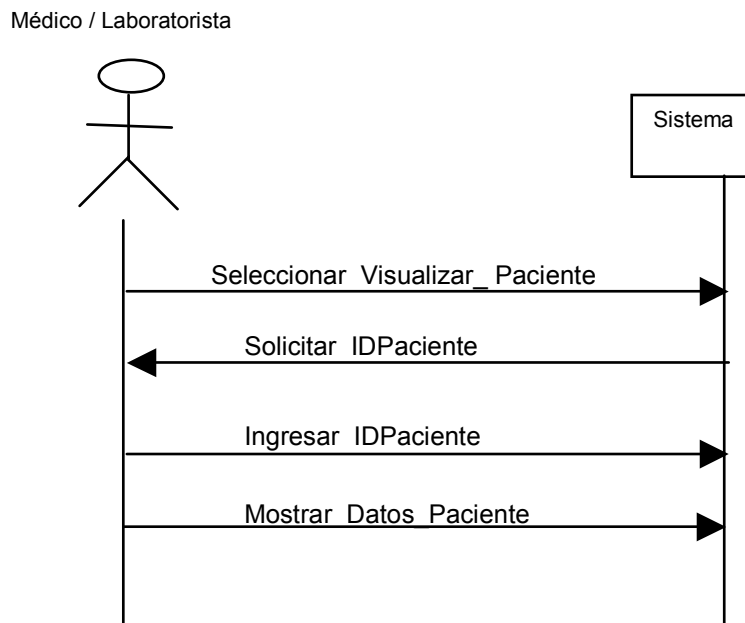


Fig. 7 Diagrama de secuencia del caso de uso: Visualizar paciente

Para el caso de uso: Realizar diagnóstico clínico

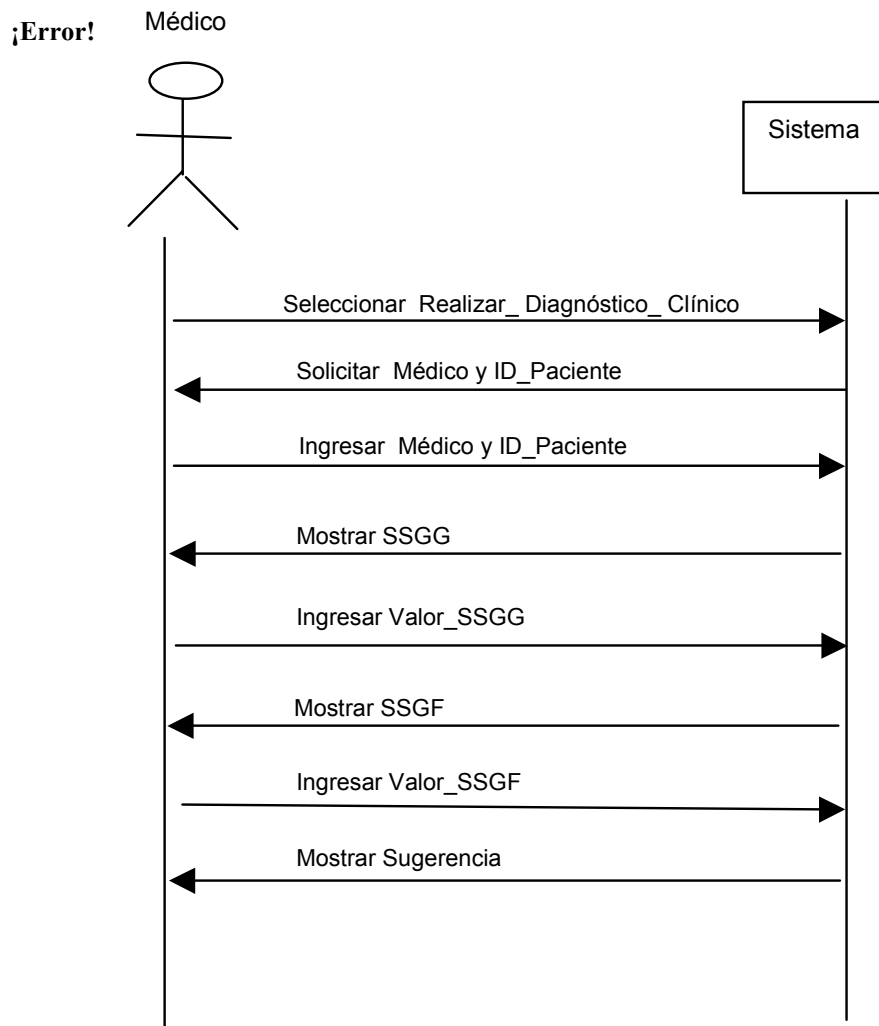
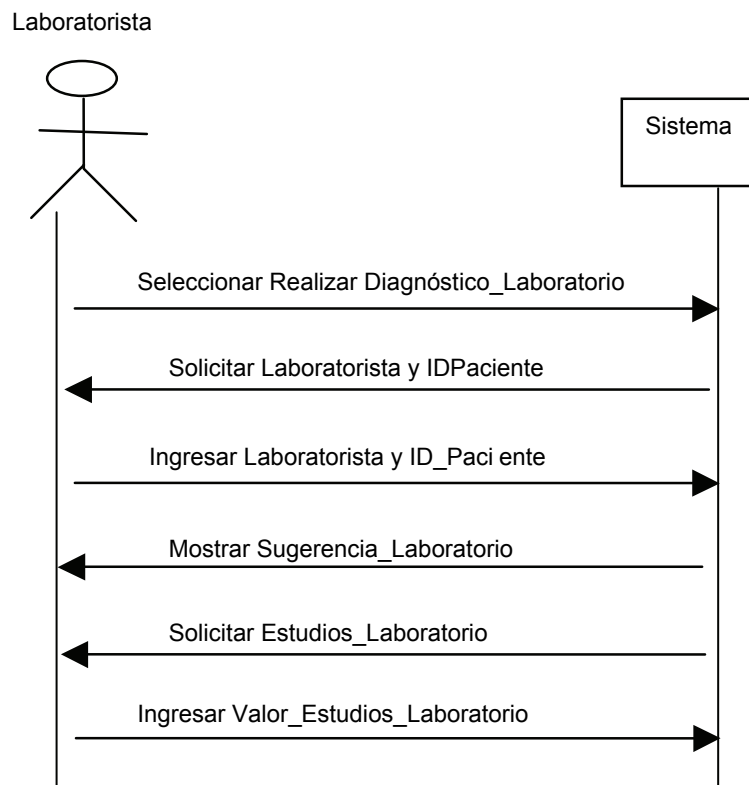


Fig. 8 Diagrama de secuencia del caso de uso: Realizar diagnóstico clínico

Para el caso de uso: Realizar diagnóstico de laboratorio



**Fig. 9 Diagrama de secuencia del caso de uso:
Realizar diagnóstico de laboratorio**

Para el caso de uso: Resultados del diagnóstico

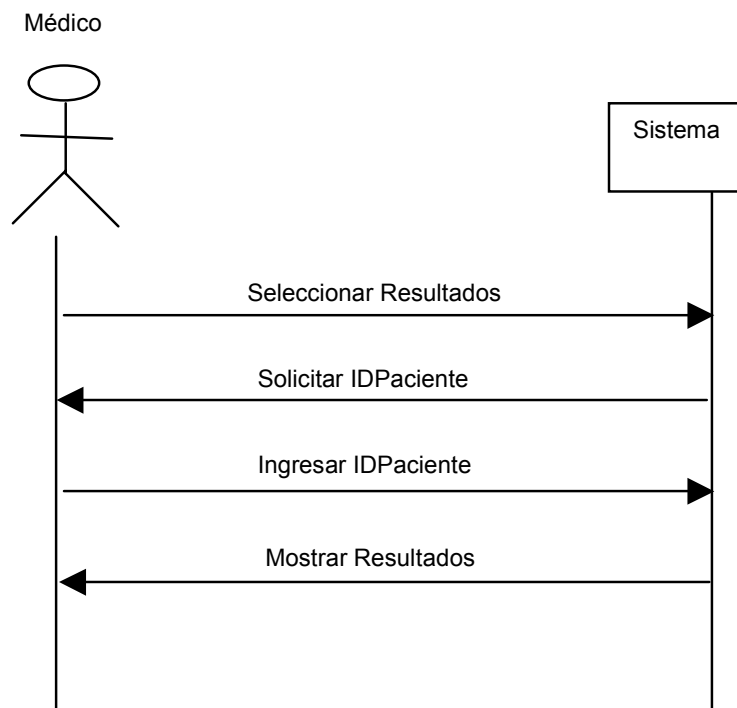


Fig. 10 Diagrama de secuencia del caso de uso: Resultados del diagnóstico

CAPÍTULO III

MODELO ARQUITECTÓNICO DE UN SISTEMA PARA TAREAS DE DIAGNÓSTICO MÉDICO

3.1. Identificación

3.1.1. Identificación de los elementos arquitectónicos

En este apartado se identificarán los elementos arquitectónicos del modelo propuesto, de un grano más bajo de abstracción a uno más alto.

Componente Motor de Inferencia.- establece el control del sistema, que construye de una forma dinámica las soluciones. Concretiza el conocimiento abstracto que posee el sistema, para obtener las conclusiones y tomar las decisiones correspondientes. Es quien da la estrategia general de resolución. Dado que es independiente del conocimiento, un mismo Motor de Inferencia puede ser la base de varios sistemas en diferentes campos del conocimiento; esto significa que una variación en la base de conocimientos puede significar una variación en el resultado pero no variará el control y por tanto éste permanecerá libre de errores debidos a cambios en la misma. Los procesos de inferencia se llevan a cabo a través de las estrategias de razonamiento como el encadenamiento de reglas hacia delante (Razonamiento Forward), el encadenamiento de reglas hacia atrás (Razonamiento Backward).

Este componente tiene un sólo puerto por donde se establece la comunicación con la Base de Conocimientos, el Usuario Clínico y Usuario Laboratorio, a través de los servicios que conforman la interfaz Inference.

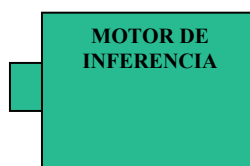


Fig. 11 Componente Motor de Inferencia

Componente Base de Conocimientos.- Este componente contiene el conocimiento del dominio en el cual es competente el programa , i.e. la información del dominio del caso de estudio.

El conocimiento tiene que representarse con el fin de que pueda incluirse en el sistema; se hace de un modo relacional entre él mismo, mediante reglas de inferencia que son cláusulas de Horn con cabeza.

La Base de Conocimientos contiene reglas y hechos (información acerca de un dominio especializado de conocimientos). Este componente contempla un almacén temporal de información dinámica; en ésta es almacenada, en forma de hechos, toda la información brindada por el usuario al sistema (respuestas a preguntas formuladas), así como las conclusiones de todas las reglas disparadas en el transcurso del proceso de inferencias. Cuando el proceso de solución de un problema particular ha concluido, el contenido de la memoria de trabajo es removido o eliminado, de forma tal que esta memoria queda limpia antes de iniciar la solución de otro problema.

Se denominan hechos a la información que es invariable de una a otra resolución, mientras que los datos son la información que varía de una a otra resolución. Los hechos representan la estructura dinámica del conocimiento ya que su número puede verse incrementado a medida que se van relacionando las reglas del dominio.

El componente Base de Conocimientos cuenta con un puerto para solicitar y enviar los servicios que han de publicarse mediante la interfaz *IDomain*, y que lo comunican con el Motor de Inferencia, el Usuario Clínico y el Usuario Laboratorio.



Fig. 12 Componente Base de Conocimientos

Conector Diagnóstico.- Este elemento arquitectónico sincroniza y conecta los componentes Motor de Inferencia y Base de Conocimientos al conformar un sistema o componente complejo. Aquí es donde se establece la principal comunicación del sistema, al redireccionar peticiones entre dichos componentes. Es decir, este conector es un repetidor, por lo que sólo reenvía los servicios solicitados por los mencionados componentes, que han de realizarse a través de sus tres puertos y que han de publicarse mediante las interfaces *IInference*, *IKnowledge*, *IDiagnostical*, respectivamente.



Fig. 13 Conector Diagnóstico

Sistema Diagnosticador.- este componente complejo o sistema conformado por el componente Motor de Inferencia, el componente Base de Conocimientos y el Conector Diagnóstico, cuenta sólomente con un puerto por el que se publican los servicios de la interfaz IDiagnostical. Este elemento arquitectónico se conecta a los componentes Usuario Clínico y Usuario Laboratorio mediante el conector Usuarios.

Considerando la relación de composición de sus elementos arquitectónicos, el comportamiento del propio sistema y de cada uno de los elementos que lo componen, la definición de la semántica de este sistema establece:

- **composición fuerte.-** no se puede acceder directamente a los elementos arquitectónicos que forman el sistema (concepto de caja negra); su acceso se realiza a través del sistema de forma indirecta . La petición de servicios se hace al sistema y la solicitud de servicios a otros elementos el sistema, a pesar de que dichos servicios, realmente sean atendidos o solicitados por los elementos arquitectonicos que lo forman.
- **composición dinámica.-** es posible alterar la topología del sistema en tiempo de ejecución aumentando su flexibilidad y dando soporte .
- **composición no localizada.-** todos los elementos arquitectónicos que forman el sistema y el propio sistema pueden ubicarse en diferentes “sites”. Por lo tanto, la movilidad de un elemento arquitectónico del sistema sólo afecta a éste y del mismo modo ocurre con el sistema.

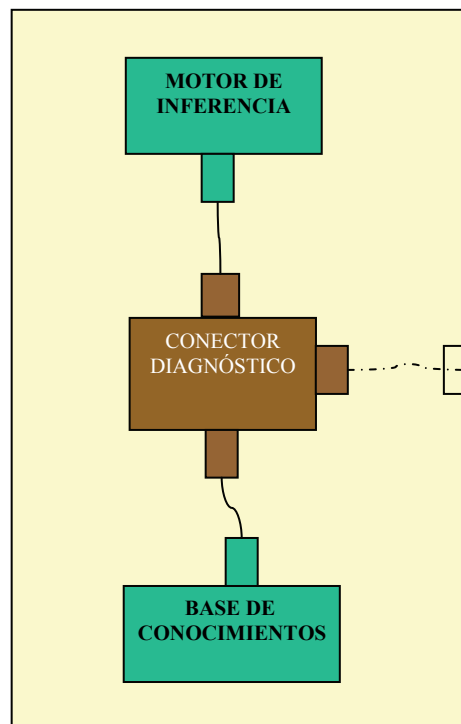


Fig. 14 Sistema Diagnosticador

Componente Usuario Clínico.- que establece la interacción humano-máquina; es la interfaz de entrada/salida que permite la comunicación entre el usuario en el contexto clínico y el sistema. A través de éste, el usuario ofrece datos iniciales al sistema o responde a preguntas formuladas por éste.

Este componente tiene un sólo puerto, para comunicarse con el sistema diagnosticador, realizando los servicios que se publican por la interfaz IClinical.



Fig. 15 Componente Usuario Clínico

Componente Usuario Laboratorio.- este componente es muy parecido al componente UsuarioClínico, pero ejerce un papel diferente, ya que permite la comunicación entre el usuario en el contexto de laboratorio y el sistema.

También cuenta con un puerto, donde mediante la interfaz ILaboratory publica los servicios

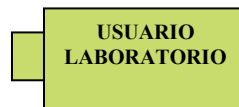


Fig. 16 Componente Usuario Laboratorio

Conector Usuarios.- el conector entre los operarios y el componente complejo hace la función de repetidor entre ambos, simplemente reenvía a ambos los servicios solicitados. Contiene tres puertos, uno de ellos recibe y envía los servicios que han de publicarse mediante la interfaz IDiagnostical para comunicar al Sistema Diagnosticador con los componentes Usuario Clínico y Usuario Laboratorio; el segundo puertos establece la comunicación entre el Sistema Diagnosticador y el Usuario Clínico que utiliza los servicios de la interfaz IClinical; y finalmente el tercer puerto que conecta al Usuario Laboratorio con el sistema Diagnosticador publicando sus servicios a través de la interfaz ILaboratory.



Fig. 17 Conector Usuarios

3.1.2. Identificación del modelo arquitectónico final

Ya identificados los elementos arquitectónicos del modelo, se está en posibilidad de establecer la arquitectura final del modelo propuesto para realizar tareas de diagnóstico médico, en el marco de PRISMA. Dicha arquitectura, descrita en la figura 18, está formado por:

- Un componente complejo o sistema que comprende:
 - Un componente simple donde se realizan los procesos de inferencia del sistema, denominado Motor de Inferencia
 - Un componente simple que contiene el conocimiento del dominio.
 - Un conector llamado Conector Dominio, que une al Motor de Inferencia y la Base de Conocimientos, por lo que permite la comunicación entre estos dos componentes.
- Dos componentes simples asociados a los usuarios del sistema. Uno para el contexto clínico (Usuario Clínico) y otro para el contexto de laboratorio y gabinete (Usuario Laboratorio).
- Un conector, el Conector Usuarios, que establece la unión, y por ende, la comunicación, entre los componentes Usuarios y el componente complejo descrito anteriormente.

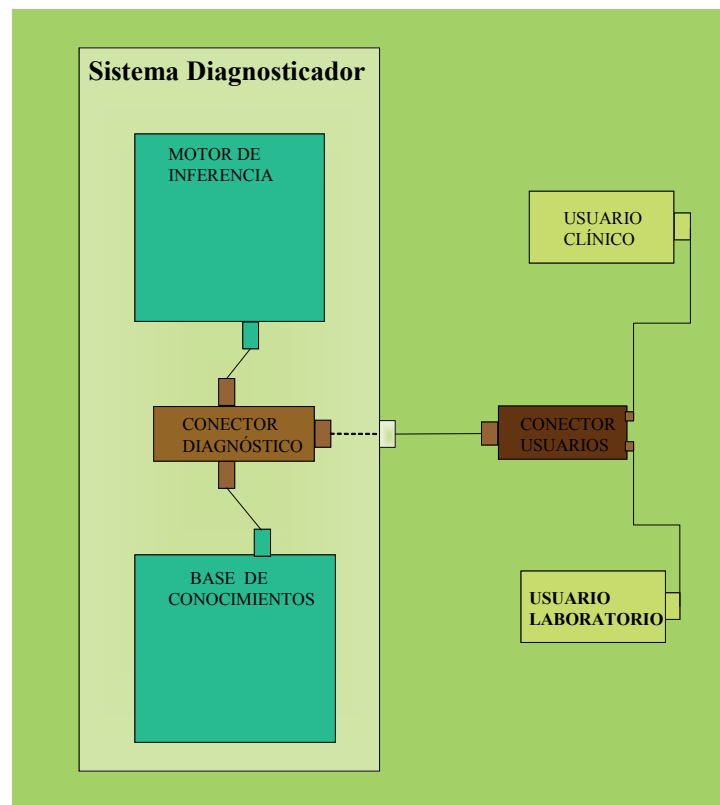


Fig. 18 Arquitectura del Sistema de Diagnóstico Médico

3.1.3. Identificación de los Concerns

Ya identificados los elementos del modelo arquitectónico del Sistema de Diagnóstico Médico, es necesario identificar los *concerns* comunes al sistema (*crosscutting concerns*) con el objetivo de separar cada uno de ellos en distintos aspectos para su reutilización .

El Sistema de Diagnóstico Médico tiene como función el realizar el diagnóstico médico de pacientes, a partir de datos observados y que son introducidos al sistema por un usuario. Esto hace emerger un concern de gran relevancia, el concern funcional.

Cabe destacar la necesidad de establecer los comportamientos de cada uno de los componentes de forma coordinada, así como la sincronización de los servicios enviados y recibidos entre ellos, por lo que surge la aparición del concern coordinación.

Finalmente, se identifica un concern que cobra relevancia al estar presente en todos y cada uno de los elementos arquitectónicos del sistema propuesto, es el concern distribución, ya que sus propiedades ubican a los componentes y conectores en la máquina.

3.1.4. Identificación de los Aspectos

Es importante separar cada uno de los concerns identificados en el modelo arquitectónico , en una entidad reutilizable denominada aspecto, con el fin de mejorar la reutilización y mantenimiento de la arquitectura.

Los aspectos de cada uno de los elementos del modelo arquitectónico, se describen a continuación:

3.1.4.1 Identificación de los aspectos del componente Motor de Inferencia

Los tres aspectos que están presentes en el Motor de Inferencia son los aspectos funcional, de distribución y de persistencia.

Aspecto funcional.- Captura la semántica de la organización, mediante la definición de su estructura y comportamiento, importando la interfaz Inference.

Aspecto de distribución.- especifica la ubicación de la instancia en la que se encuentra.

Aspecto de persistencia.- Si existe interés en guardar cierta información para después ser reutilizada o bien consultada, es conveniente contemplar un aspecto específico que permita salvar dicha información, éste es el aspecto de persistencia. De esta manera es posible realizar la gestión de aplicaciones persistentes pudiendo acceder a base de datos y almacenar documentos (archivos de los pacientes).

3.1.4.2 Identificación de los aspectos del componente Base de Conocimientos

Los aspectos funcional y de distribución, son los aspectos que conforman este componente

Aspecto funcional.- captura la semántica de la organización del conocimiento del dominio que utiliza el sistema mediante la definición de su comportamiento, a través de la interfaz IDomain.

Aspecto de distribución.- especifica la ubicación de la instancia en la que se encuentra

3.1.4.3 Identificación de los aspectos del Conector Diagnóstico

Los aspectos necesarios para especificarlo son los de coordinación y de distribución.

Aspecto de coordinación- el conector diagnóstico es un elemento arquitectónico que actúa como coordinador entre otros elementos arquitectónicos y que permite describir interacciones complejas entre componentes mediante este aspecto.

Este conector define la sincronización entre elementos arquitectónicos durante su comunicación, redireccionando peticiones a los componentes, mismos que han de de realizarse a través de sus puertos y que han de publicarse mediante las interfaces Inference, IDomain, IDiagnostical.

Este aspecto carece de estado, únicamente recibe o envía los servicios que se le solicitan, por lo que sus servicios no tienen valuaciones asociadas, tan sólo participan en un proceso modelado a través de un protocolo.

Aspecto de distribución.- especifica la ubicación de la instancia en la que se encuentra

3.1.4.4 Identificación de los aspectos del componente Usuario Clínico

Los cuatro aspectos que son importados por el componente Usuario Clínico son el funcional, el de presentación, el de distribución y el de replicación.

Aspecto funcional.- consiste en la solicitud de los servicios que le otorga el sistema, importando la interfaz IClinical. Captura la semántica de la organización del componente, mediante la definición de su comportamiento.

Aspecto de presentación.- este aspecto alberga el conjunto de gráficos que van a definir la interfaz con la que va a trabajar el usuario. Sus servicios son internos, por lo que no se publican mediante una interfaz.

Aspecto de distribución.- especifica la ubicación de la instancia desde la cual el usuario manipula el sistema así como los servicios de movilidad para desplazarse a otra máquina, en caso de que se requiera, ya que se puede gestionar el sistema desde distintas máquinas, p.e. el usuario podrá utilizar el sistema desde un consultorio, el hospital, su casa, etc.

Se hace notar que, aunque no se contempló en este trabajo, en el aspecto de distribución se podrían establecer las restricciones en las direcciones IP de acceso al sistema (p.e. el usuario sólo puede acceder al sistema desde el consultorio u hospital, pero no en su casa).

Aspecto de replicación.- permite el acceso simultáneo de diferentes usuarios, ya que es posible que haya más de un usuario utilizando el sistema (médico general o especialistas).

3.1.4.5 Identificación de los aspectos del componente Usuario Laboratorio

Usuario Laboratorio está formado por cuatro aspectos: funcional, presentación, distribución y replicación.

Aspecto funcional.- consiste en la solicitud de los servicios que le otorga el sistema, importando la interfaz ILaboratory. Captura la semántica de la organización del componente, mediante la definición de su comportamiento.

Aspecto de presentación.- este aspecto alberga el conjunto de gráficos que van a definir la interfaz con la que va a trabajar el usuario. Sus servicios son internos, por lo que no se publican mediante una interfaz.

Aspecto de distribución.- especifica la ubicación de la instancia desde la cual el usuario manipula el sistema así como los servicios de movilidad para desplazarse a otra máquina, en caso de que se requiera, ya que se puede gestionar el sistema desde distintas máquinas, p.e. el usuario podrá utilizar el sistema desde un laboratorio, el hospital, su casa, etc.

Se hace notar que, aunque no se contempló en este trabajo, en el aspecto de distribución se podrían establecer las restricciones en las direcciones IP de acceso al sistema (p.e. el usuario sólo puede acceder al sistema desde el laboratorio u hospital, pero no en su casa).

Aspecto de replicación.- permite el acceso simultáneo de diferentes usuarios, ya que es posible que haya más de un usuario utilizando el sistema (laboratoristas o equipos médicos).

3.1.4.6 Identificación de los aspectos del Conector Usuarios

Los aspectos presentes en el conector Usuarios son el de coordinación y de distribución.

Aspecto de coordinación .- define la sincronización entre elementos arquitectónicos durante su comunicación, mediante las interfaces IDiagnostical, IClinical e ILaboratory. También carece de estado, únicamente envía los servicios que se le solicitan a los usuarios o recibe los servicios del componente complejo, por lo tanto, sus servicios no

tienen valuaciones asociadas, sólomente participan en un proceso modelado a través de un protocolo.

Aspecto de distribución.- especifica las características que definen la localización dinámica del elemento arquitectónico en el cual se integra. También define los servicios necesarios para implementar estrategias de distribución de los elementos arquitectónicos (como movilidad) con el objetivo de optimizar la distribución de la topología del sistema resultante [Ali03].

3.1.5 Identificación de Interfaces

Dado que los servicios que se envían o solicitan entre los elementos arquitectónicos del sistema, se han de publicar mediante interfaces que se reutilizarán en diferentes puertos y aspectos, en este apartado se identificarán a continuación las diversas interfaces del modelo arquitectónico, haciendo mención a los aspectos y puertos de los componentes o conectores que les da semántica.

IIference.- los servicios que se publican por esta interfaz son solicitados y recibidos por el aspecto funcional del componente Motor de Inferencia a través de su único puerto y el aspecto de coordinación del Conector Diagnóstico a través de uno de sus puertos.

IKnowledge.- esta interfaz contempla los servicios entre el aspecto funcional del componente Base de Conocimientos a través de su único puerto y el aspecto de coordinación del Conector Diagnóstico a través de uno de sus puertos

IDiagnostical.- el aspecto de coordinación del Conector Diagnóstico importa esta interfaz en el puerto por el que comunica al Sistema Diagnosticador con los otros elementos arquitectónicos, misma que es reutilizada en el aspecto de coordinación del Conector Usuarios en uno de sus tres puertos que se utilizan para conectar al Sistema Diagnosticador con los componentes Usuarios.

IClinical.- el componente Usuario Clínico está formado, entre otros, por el aspecto funcional que utiliza esta interfaz a través de su puerto, y así mismo es utilizada por el aspecto de coordinación del Conector Usuarios en uno de sus puertos.

ILaboratory.- igual al componente anterior, en el aspecto funcional del componente Usuario Laboratorio se importa mediante su único puerto, a la interfaz ILaboratory, misma que es reutilizada en el aspecto de coordinación del Conector Usuarios en uno de sus puertos.

3.2 Diagramas de secuencia correspondientes a las realizaciones de los casos de uso

Ya habiéndose realizado la identificación de todos los elementos arquitectónicos del modelo propuesto, es conveniente realizar los diagramas de secuencias que corresponden a las realizaciones de los casos de uso (indicados en el apartado 2.2.2 de este trabajo). Dichos diagramas de secuencia se muestran a continuación.

Para el caso de uso: “Registrar Paciente”.- El MOTOR DE INFERENCIA solicita al USUARIO CLINICO los datos del paciente que se desea registrar, mediante el servicio *IntroducirPaciente*.

Dicha información es guardada por el MOTOR DE INFERENCIA con el servicio *Salvar*.

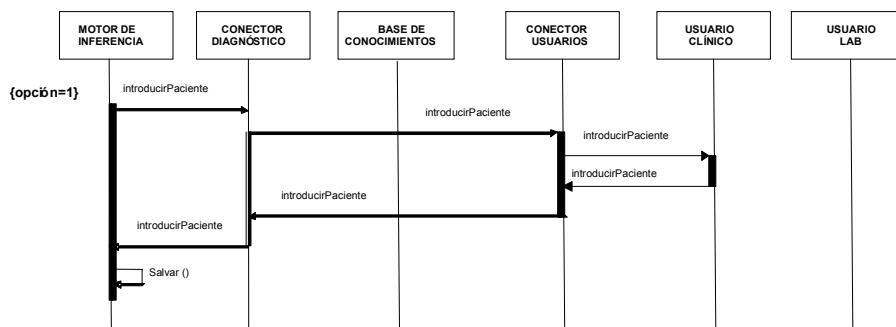


Fig. 19 Diagrama de secuencia de las realizaciones del caso de uso Registrar paciente

Para el caso de uso: “Visualizar los datos del Paciente”.- El MOTOR DE INFERENCIA solicita el identificador del paciente con el servicio *IntroducirIDPaciente*, mismo que es atendido por el USUARIO CLÍNICO y/o el USUARIO DE LABORATORIO.

Con esta información el MOTOR DE INFERENCIA a través del servicio *ConsultarPaciente* adquiere (de él mismo) los datos con que se registró al paciente, para enviarlos al USUARIO CLÍNICO y/o el USUARIO DE LABORATORIO mediante el servicio *TomarPaciente*, con el fin de que el usuario pueda visualizar los datos del registro del paciente.

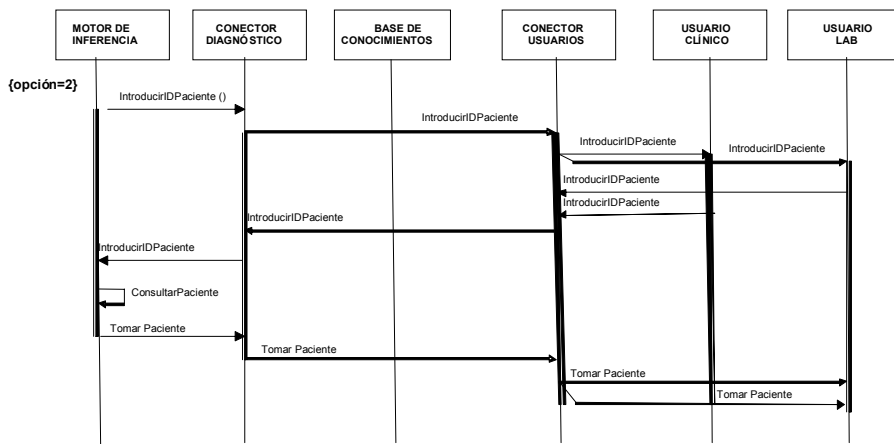


Fig. 20 Diagrama de secuencia de las realizaciones del caso de uso Visualizar datos paciente

Para el caso de uso: “Realizar el diagnóstico clínico”.- El MOTOR DE INFERENCIA obtiene del USUARIO CLINICO el nombre del médico mediante el servicio *IntroducirUsuarioClin* así como el identificador del paciente con el servicio *IntroducirIDPaciente*.

A continuación el MOTOR DE INFERENCIA envía a la BASE DE CONOCIMIENTOS el servicio *AdquirirPosEnfermedad*, para adquirir las enfermedades probables que pueda tener el paciente. Para lo cual, la BASE DE CONOCIMIENTOS solicita al USUARIO CLÍNICO los signos y síntomas de grano grueso, del paciente a través del servicio *ObtenerSSGG*. Con dichos datos la BASE DE CONOCIMIENTOS infiere las posibles enfermedades y las envía al MOTOR DE INFERENCIA a través del servicio *AdquirirPosEnfermedad*.

Cabe señalar que además de que el MOTOR DE INFERENCIA obtiene las posibles enfermedades que pueda tener el paciente, solicita una sugerencia de estudios de laboratorio con el fin de realizarle al enfermo sólo los estudios necesarios; por ello envía a la BASE DE CONOCIMIENTOS el servicio *AdquiriSugLab*, quien infiere dicha información. El MOTOR DE INFERENCIA después utiliza esta información (sugerencia de laboratorio) para realizar el diagnóstico de laboratorio.

Es importante mencionar que el MOTOR DE INFERENCIA solicita la sugerencia de laboratorio en este momento, para que la BASE DE CONOCIMIENTOS no tenga la necesidad de pedir nuevamente los signos y síntomas de grano grueso, ya que las reglas

de inferencia para adquirir tanto de las posibles enfermedades como de la sugerencia de laboratorio, utilizan la misma información para ser inferidas, es decir, los signos y síntomas de grano grueso.

El MOTOR DE INFERENCIA adquiere de la BASE DE CONOCIMIENTOS las posibles enfermedades del paciente con el fin de limitar las opciones del árbol de decisión y enviarlas como parámetro al siguiente paso, que es la solicitud del servicio *AdquirirDiagClin* para que se realice el diagnóstico clínico del paciente. Es conveniente hacer notar que esta misma información (las posibles enfermedades) se utilizará posteriormente al realizar el diagnóstico de laboratorio.

De esta forma, la BASE DE CONOCIMIENTOS al recibir el servicio *AdquirirDiagClin* solicita al USUARIO CLÍNICO con el servicio *ObtenerSSGF*, los signos y síntomas de grano fino del paciente (de solamente estas enfermedades y no de la totalidad contenida en la BASE DE CONOCIMIENTOS). El USUARIO CLINICO envía dicha información (signos y síntomas de grano fino) a la BASE DE CONOCIMIENTOS, quien infiere el diagnóstico clínico. De esta forma el MOTOR DE INFERENCIA adquiere el diagnóstico clínico del paciente.

Por último, el MOTOR DE INFERENCIA envía a la BASE DE CONOCIMIENTOS el servicio *AdquirirSugerencia* para conocer la sugerencia de si es necesario o no, realizar los estudios de laboratorio.

Con lo que el MOTOR DE INFERENCIA envía al USUARIO CLINICO el servicio *TomarSugerencia* para que pueda mostrar al usuario dicha sugerencia.

El MOTOR DE INFERENCIA guarda la información resultante de este proceso con el servicio *Salvar*.

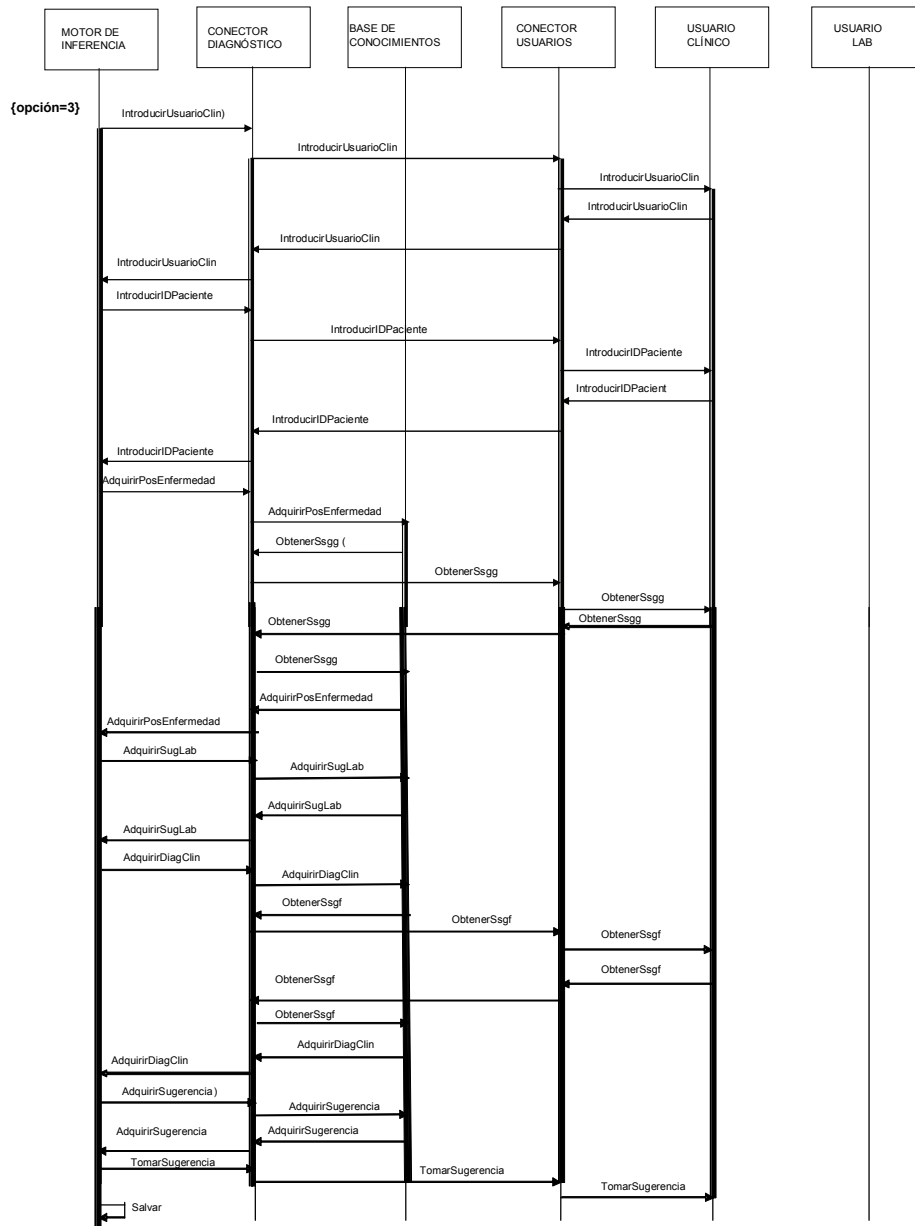


Fig. 21 Diagrama de secuencia de las
del caso de uso Realizar diagnóstico clínico

Para el caso de uso: “Realizar diagnostico de laboratorio”.- El MOTOR DE INFERENCIA solicita al USUARIO LABORATORIO el nombre del laboratorista y el identificador del paciente, mediante los servicios *IntroducirUsuarioLab* y *IntroducirIDPaciente*, respectivamente.

Acto seguido, el MOTOR DE INFERENCIA, a través del servicio *ConsultarPaciente*, adquiere (de él mismo) la sugerencia de laboratorio del paciente asociado al identificador correspondiente. Con lo que envía dicha sugerencia al USUARIO LABORATORIO por medio del servicio *TomarSugLab*, con el fin de que el usuario obtenga las sugerencias de estudios de laboratorio que le realizará al paciente.

Asímismo, el MOTOR DE INFERENCIA a través del servicio *ConsultarPaciente* adquiere (de él mismo) las posibles enfermedades del paciente asociado al identificador correspondiente. Enviando así a la BASE DE CONOCIMIENTOS el servicio *AdquirirDiagLab* para obtener el diagnóstico de laboratorio. Acto seguido, la BASE DE CONOCIMIENTOS solicita al USUARIO LABORATORIO a través del servicio *ObtenerEstsLab* los resultados de los estudios de laboratorio realizados al paciente. De esta forma la BASE DE CONOCIMIENTOS infiere el diagnóstico que envía al MOTOR DE INFERENCIA.

Por último, el MOTOR DE INFERENCIA guarda la información resultante de este proceso con el servicio *Salvar*.

A continuación, el MOTOR DE INFERENCIA guarda esta información con el servicio *Salvar*.

Finalmente, el MOTOR DE INFERENCIA envía al USUARIO CLINICO el servicio *TomarResultados* con el fin de que el usuario pueda visualizar los resultados del diagnóstico realizado a su paciente (los diagnósticos clínico y de laboratorio con sus respectivas certidumbres, la terapia indicada, así como los nombres del médico y del laboratorista responsables de los diagnósticos).

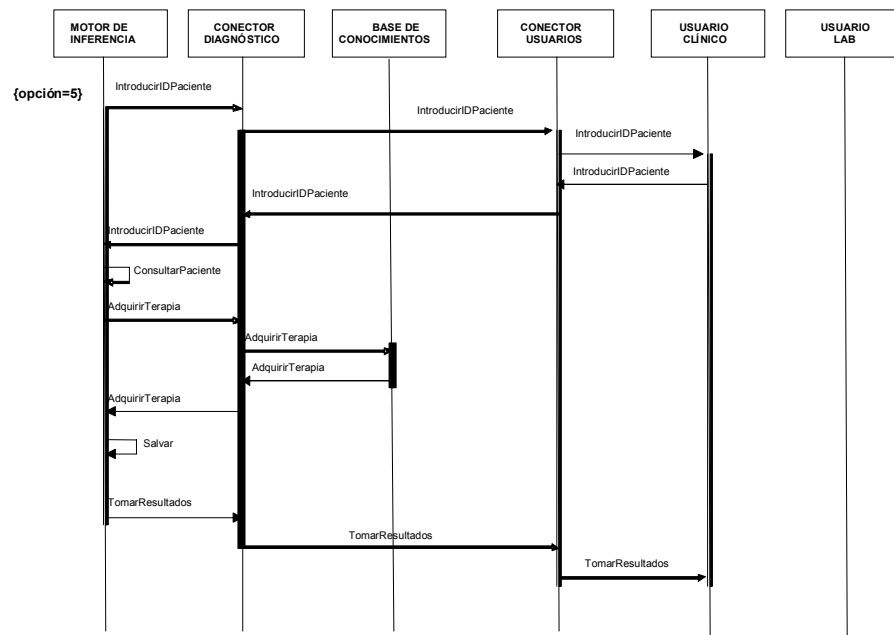


Fig. 23 Diagrama de secuencia de las realizaciones del caso de uso Resultados del diagnóstico

3.3 Especificación

3.3.1. Especificación de interfaces, aspectos y elementos arquitectónicos. Configuración de los elementos arquitectónicos.

En este apartado se van a especificar los aspectos de cada uno de los componentes y conectores del modelo arquitectónico, incluyendo las interfaces que importan, y a continuación se realizará la especificación del elemento arquitectónico que contiene dichos aspectos. Finalmente, se mostrará su configuración e instanciación.

Al igual que en se ha procedido para el diseño del modelo arquitectónico, la especificación de los elementos arquitectónicos del modelo propuesto se han especificado de forma ascendente, partiendo del nivel más bajo de abstracción y acabando en el más alto. Por ello, la especificación de un componente complejo se realizará primeramente con los componentes simples y posteriormente por el componente complejo de mayor granularidad. Por último, se se va a especificar la configuración del modelo arquitectónico final y su instanciación.

Cabe mencionar que existen dos servicios *begin* y *end* propios del modelo PRISMA, necesarios para la correcta especificación de los aspectos; ambos servicios definen el comienzo y el final de la ejecución del aspecto al que pertenece, respectivamente. El servicio *begin*, además proporciona el valor que han de tener los atributos requeridos del aspecto. La ejecución del servicio *begin* es desencadenada por el servicio que crea la instancia del elemento arquitectónico que importa el aspecto al que pertenece el servicio *begin*. Así mismo, la ejecución del servicio *end* es desencadenada por el servicio que destruye la instancia del elemento arquitectónico que importa el aspecto al que pertenece el servicio *end*.

3.3.1.1 Especificación de los aspectos del Componente Motor de Inferencia

Aspecto funcional

El sistema inicia su funcionamiento con el MOTOR DE INFERENCIA que solicita mediante el servicio *IntroducirLoginPassword* el nombre de usuario y la contraseña del usuario. Acto seguido este componente envía el servicio *IntroducirSeleccion* para adquirir la opción elegida a realizarse.

IntroducirLoginPassword (**output** Login:string, **output** Password:string)

IntroducirSeleccion (**output** Seleccion:int)

- Si se ha elegido la opción de “Registrar Paciente”, el MOTOR DE INFERENCIA solicita que se introduzcan los datos del paciente que desea registrar, mediante el servicio *IntroducirPaciente*.

IntroducirPaciente (**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string, **output** Dirección:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date)

- Si se ha seleccionado “Visualizar los datos del Paciente”, el MOTOR DE INFERENCIA solicita el identificador del paciente con el servicio *IntroducirIDPaciente*.

IntroducirIDPaciente (**output** IDPaciente:identifier)

Con esta información el MOTOR DE INFERENCIA envía el servicio *TomarPaciente* con el fin de que el usuario pueda visualizar los datos del registro de su paciente.

TomarPaciente (**input** DNI:string, **input** Nombre:string, **input** Dirección:string, **input** Tel:string, **input** Email:string, **input** Sexo:char, **input** FechaNac:string)

- Si se desea “Realizar el diagnóstico clínico”, el MOTOR DE INFERENCIA obtiene el nombre del médico mediante el servicio *IntroducirUsuarioClin* y el identificador del paciente con el servicio *IntroducirIDPaciente*.

IntroducirUsuarioClin (**output** UsuarioClin:string)

IntroducirIDPaciente (**output** IDPaciente:identifier)

A continuación el MOTOR DE INFERENCIA solicita con el servicio *AdquirirPosEnfermedad*, las enfermedades probables que pueda tener el paciente.

Cabe señalar que el MOTOR DE INFERENCIA adquiere las posibles enfermedades del paciente con el fin de limitar las posibles opciones del árbol de decisión y enviarlas como parámetro de entrada al siguiente paso, la solicitud del servicio *AdquirirDiagClin*, para que se realice el diagnóstico clínico del paciente.

AdquirirPosEnfermedad (**output** PosEnfermedades:string)

AdquirirDiagClin (**input** PosEnfermedades:string, **output** EnfermedadClin:string, **output** CertClin:real)

Es ahora conveniente hacer notar que esta misma información se utilizará posteriormente al realizar el diagnóstico de laboratorio del mismo paciente.

Cabe señalar que además de que el MOTOR DE INFERENCIA obtiene las posibles enfermedades que pueda tener el paciente, solicita una sugerencia de estudios de laboratorio con el fin de realizarle al enfermo sólo los estudios necesarios; por ello envía el servicio *AdquirirSugLab*, con el cual adquiere la información que después utilizará en el proceso para realizar el diagnóstico de laboratorio. La razón de estos últimos servicios se explicará en el aspecto funcional de la BASE DE CONOCIMIENTOS.

AdquirirSugLab (**output** SugerenciaLab:string)

Por último el MOTOR DE INFERENCIA envía el servicio *AdquirirSugerencia*, para conocer la sugerencia de si es_necesario_realizar_estudios_de_laboratorio.

AdquirirSugerencia (**output** Sugerencia:string)

Con lo que el MOTOR DE INFERENCIA envía el servicio *TomarSugerencia* para que se pueda mostrar al usuario la sugerencia de si es necesario realizar o no los estudios de laboratorio.

TomarSugerencia (**input** Sugerencia:string)

- Si la selección fue “Adquirir el diagnostico de laboratorio”, el MOTOR DE INFERENCIA primero solicita el nombre del laboratorista y el identificador del paciente, mediante los servicios *IntroducirUsuarioLab* y *IntroducirIDPaciente* respectivamente.

IntroducirUsuarioLab (**output** UsuarioLab:string)

IntroducirIDPaciente (**output** IDPaciente:identifier)

Acto seguido el MOTOR DE INFERENCIA envía el servicio *TomarSugLab* para que el usuario obtenga las sugerencias de estudios de laboratorio que le realizará al paciente.

TomarSugLab (**input** SugLab:string)

Y finalmente, envía el servicio *AdquirirDiagLab* para obtener los resultados del diagnóstico de laboratorio realizado.

AdquirirDiagLab (**output** EnfermedadLab:string, **output** CertLab:real)

- Para el caso de “Visualizar resultados del diagnóstico”, el MOTOR DE INFERENCIA con el servicio *IntroducirIDPaciente* solicita el identificador del paciente.

IntroducirIDPaciente (**output** IDPaciente: identifier)

Con ello el MOTOR DE INFERENCIA utiliza los datos del diagnóstico clínico y del de laboratorio para solicitar, con el servicio *AdquirirTerapia*, el tratamiento que se le dará al paciente para curar la enfermedad detectada.

AdquirirTerapia (**input** EnfermedadClin:string, **input** CertClin:real, **input** EnfermedadLab:string, **input** CertLab:real, **output** Terapia:string)

Con esta información el MOTOR DE INFERENCIA envía el servicio *TomarResultados* con el fin de que el usuario pueda visualizar los resultados del diagnóstico realizado a su paciente, incluyendo los diagnósticos clínico y de laboratorio con sus respectivas certidumbres, así como los nombres del médico y del laboratorista responsables de los mismos.

TomarResultados (**input** EnfermedadClin:string, **input** CertClin:real, **input** EnfermedadLab:string, **input** CertLab:real, **input** Terapia:string, **input** UsuarioClin:string, **input** UsuarioLab:string)

Los servicios antes expuestos se publican en el puerto del componente MOTOR DE INFERENCIA denominado *MotorPort*, que tiene asociada como signatura la interfaz *IInference* y que da semántica a su aspecto funcional.

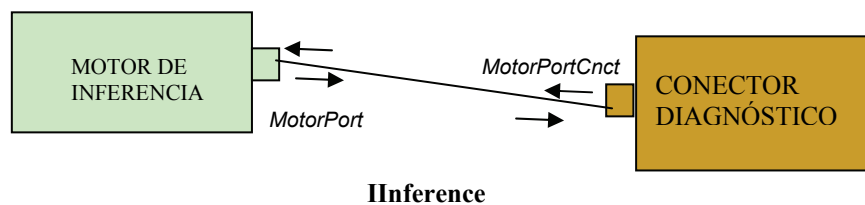


Fig. 24 Esquema del funcionamiento de la Interfaz IInference

La especificación de dicha interfaz tiene la siguiente sintaxis:

Interfaz IInference

limpiarBD ()

introducirLoginPassword (**output** Login:string , **output** Password:string)

introducirSeleccion (**output** Seleccion:int)

introducirPaciente(**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string, **output** Direccion:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date)

introducirIDPaciente (**output** IDPaciente:identifier)

introducirUsuarioClin (**output** UsuarioClin:string)

introducirUsuarioLab (**output** UsuarioLab:string)

adquirirSugLab (**output** SugLab:string).

adquirirPosEnfermedad (**output** PosEnfermedades:string)

adquirirDiagClin (**input** PosEnfermedades:string, **output** EnfermedadClin:string, **output** CertClin:real)

adquirirDiagLab (**input** PosEnfermedades:string, **output** EnfermedadLab:string, **output** CertLab:real)

adquirirTerapia (**input** EnfermedadClin:string, **input** CertClin:real, **input** EnfermedadLab:string, **input** CertLab:real, **output** Terapia:string)

tomarPaciente (**input** DNI:string, **input** Nombre:string, **input** Direccion:string, **input** Tel:string, **input** Email:string, **input** Sexo:char, **input** FechaNac:date)

tomarResultados (**input** EnfermedadClin:string, **input** CertClin:real, **input** EnfermedadLab:string, **input** CertLab:real, **input** Terapia:string, **input** UsuarioClin:string, **input** UsuarioLab:string)

tomarSugLab (**input** SugLab:string)

tomarSugerencia (**input** Sugerencia:string)

End_Interfaz Inference

Ahora bien, analizando este sistema se contemplan dos situaciones importantes sobre su uso, ya que debe ser también amigable. Dentro de este punto se contempló que al usuario se le hiciera el mínimo de preguntas para llegar a la solución del problema.

La primera de ellas es referente a la toma de decisiones y que se describe a continuación. La segunda hace mención a la obtención de los signos y síntomas, y que se describen en el aspecto funcional del componente BASE DE CONOCIMIENTOS.

Para ello, cabe reiterar que el MOTOR DE INFERENCIA obtiene de la BASE DE CONOCIMIENTOS las posibles enfermedades del paciente con el fin de poder reducir las opciones del árbol de decisión y enviarlas a la BASE DE CONOCIMIENTOS, quien adquiere el diagnóstico clínico y de laboratorio al solicitar los signos y síntomas de granos fino o bien los estudios de laboratorio, respectivamente, de solamente estas enfermedades y no de todas las contenidas en la BASE DE CONOCIMIENTOS.

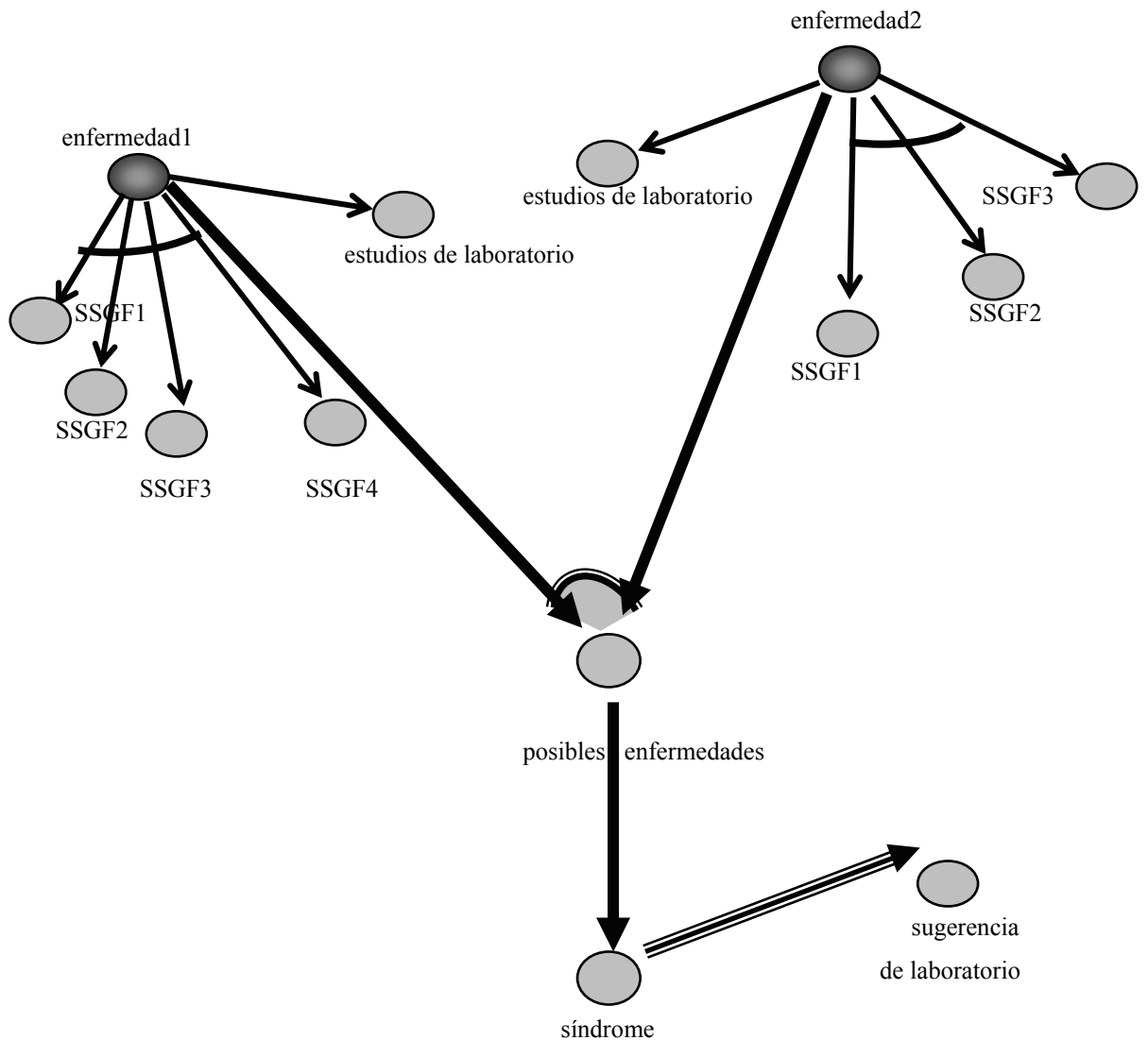


Fig. 25 Grafo del proceso de inferencia de las enfermedades

Como resultado de este análisis, la especificación del aspecto funcional del MOTOR DE INFERENCIA es el siguiente:

Functional Aspect FMotor **using** Inference

Attributes

Variable

idPaciente: identifier NOT NULL

dni:string

nombre:string

direccion:string

telefono:string

email:string

sexo:char

fechaNacimiento:date

medico: string

laboratorista: string

enfermedadClin:string
enfermedadLab:string
certidumbreClin:real
certidumbreLab:real
tratamiento:string
posEnfermedades:string
sugerenciaLab:string
sugerencia:string

Services

begin ()

out limpiarBD (..);

out introducirLoginPassword (**output** Login:string, **output** Password:string)

out introducirSeleccion (**output** Seleccion:int)

out introducirPaciente (**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string, **output** Direccion:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date)

Valuations

[**out** introducirPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac, IDPaciente)]

idPaciente:= IDPaciente

dni:= DNI

nombre:= Nombre,

direccion:= Direccion

telefono:= Tel

email:= Email;

sexo:= Sexo,

fecha_de_nac:= FechaNac;

out introducirUsuarioClin (**output** UsuarioClin:string)

Valuations

[**out** introducirUsuarioClin (UsuarioClin)] medico:= UsuarioClin;

out introducirUsuarioLab (**output** UsuarioLab:string)

Valuations

[**out** introducirUsuarioLab (UsuarioLab)] laboratorista:= UsuarioLab;

out introducirIDPaciente (**output** IDPaciente: identifier)

Valuations

[**out** introducirIDPaciente (IDPaciente)] idPaciente:= IDPaciente;

out adquirirSugLab (**output** SugLab:string)

Valuations

[**out** adquirirSugLab (SugLab)] sugerenciaLab:= SugLab;

out adquirirSugerencia (**output** Sugerencia:string).

Valuations

[**out** adquirirSugerencia (Sugerencia)] sugerencia:= Sugerencia;

out adquirirPosEnfermedad (**output** PosEnfermedades:string)

Valuations

[**out** adquirirPosEnfermedad (PosEnfermedades)]

posEnfermedades:= PosEnfermedades;

out adquirirDiagClin (**input** PosEnfermedades:string, **output** EnfermedadClin:string, **output** CertClin:real)

Valuations

[**out** adquirirDiagClin (posEnfermedades, EnfermedadClin, CertClin)]

enfermedadClin:= EnfermedadClin,
certidumbreClin:= CertClin;

out adquirirDiagLab (**input** posEnfermedades:string, **output** EnfermedadLab:string, **output** CertLab:real)

Valuations

[**out** adquirirDiagLab (posEnfermedades, EnfermedadLab, CertLab]
enfermedadLab:= EnfermedadLab,
certidumbreLab:= CertLab;

out adquirirTerapia (**input** EnfermedadClin:string, **input** CertClin:real, **input** EnfermedadLab:string, **input** CertLab:real, **output** Terapia:string)

Valuations

[**out** adquirirTerapia (enfermedadClin, certidumbreClin, enfermedadLab, certidumbreLab, Terapia)]
tratamiento:= Terapia;

out tomarPaciente (**input** DNI:string, **input** Nombre:string, **input** Direccion:string, **input** Tel:string, **input** Email:string, **input** Sexo:char, **input** FechaNac:date)

Valuations

[**out** tomarPaciente (dni, nombre, direccion, telefono, email, sexo, fechaNacimiento)];

out tomarResultados (**input** EnfermedadClin:string, **input** CertClin:real, **input** EnfermedadLab:string, **input** CertLab:real, **input** Terapia, **input** UsuarioClin:string, **input** UsuarioLab:string)

Valuations

[**out** tomarResultados (enfermedadClin, certidumbreClin, enfermedadLab, certidumbreLab, tratamiento, medico, laboratorista)];

out tomarSugLab (**input** SugLab:string)

Valuations

[**out** tomarSugLab (sugerenciaLab)];

out tomarSugerencia (**input** Sugerencia:string)

Valuations

[**out** tomarSugerencia (sugerencia)];

end;

Played Roles

MOTOR =

(Inference.IntroducirLoginPassword ! (Login, Password)

→

Inference.IntroducirLoginPassword ? (Login, Password)

→

Inference.IntroducirSeleccion ! (Seleccion)

→

Inference.IntroducirSeleccion ? (Seleccion)

→

/*INTRODUCIR PACIENTE*/

(Inference.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))

→

Inference.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))

+

/*VISUALIZAR PACIENTE*/

(Inference.IntroducirIDPaciente ! (IDPaciente)

→

Inference.IntroducirIDPaciente ? (IDPaciente)

```

→
Inference.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac) )

+
/VISUALIZAR RESULTADOS*/
(Inference.LimpiarDB ! ( )
→
Inference.IntroducirIDPaciente ! (IDPaciente )
→
Inference.IntroducirIDPaciente ? (IDPaciente)
→
Inference.AdquirirTerapia ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia)
→
Inference.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia)
→
Inference.TomarResultados ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia, UsuarioClin, UsuarioLab) )

+
/*REALIZAR DIAGNOSTICO CLINICO*/
(Inference.LimpiarDB ! ()
→
Inference.IntroducirUsuarioClin ! (UsuarioClin)
→
Inference.IntroducirUsuarioClin ? (UsuarioClin) )
→
Inference.IntroducirIDPaciente ! (IDPaciente )
→
Inference.IntroducirIDPaciente ? (IDPaciente)
→
(Inference.AdquirirPosEnfermedad ! (PosEnfermedades)
→
Inference.AdquirirPosEnfermedad ? (PosEnfermedades)
→
Inference.AdquirirSugLab ! (SugLab )
→
Inference.AdquirirSugLab ? (SugLab)
→
Inference.AdquirirDiagClin ! (PosEnfermedades, EnfermedadClin, CertClin)
→
Inference.AdquirirDiagClin ? (PosEnfermedades, EnfermedadClin, CertClin)
→
Inference.AdquirirSugerencia ! (Sugerencia)
→
Inference.AdquirirSugerencia ? (Sugerencia)
→
Inference.TomarSugerencia ! (Sugerencia) )

+
/*REALIZAR DIAGNOSTICO LABORATORIO*/
(Inference.LimpiarDB ! ()
→
Inference.IntroducirUsuarioLab ! (UsuarioLab)
→
Inference.IntroducirUsuarioLab ? (UsuarioLab)
→
Inference.IntroducirIDConsulta ! (IDConsulta)
→
Inference.IntroducirIDConsulta ? (IDConsulta)
→

```

```

IInference.TomarSugLab ! (SugLab)
→
IInference.AdquirirDiagLab ! (PosEnfermedades, EnfermedadLab, CertLab)
→
IInference.AdquirirDiagLab ? (PosEnfermedades, EnfermedadLab, CertLab) )

```

Protocols

```

FMOTOR = begin.FUNCTIONM

```

```

FUNCTIONM =
  ( MOTOR.IntroducirLoginPassword ! (Login, Password )
  →
  MOTOR.IntroducirLoginPassword ? (Login, Password) ).P0

```

```

P0 =
  ( MOTOR.IntroducirSeleccion ! (Seleccion )
  →
  MOTOR.IntroducirSeleccion ? (Seleccion) ).OPTIONALF
+
  end;

```

```

OPTIONALF =
  RegistrarPaciente
+
  VisualizarPaciente
+
  VisualizarResultados
+
  RealizarDiagnosticoClinico
+
  RealizarDiagnosticoLaboratorio
+
  Finalizar;

```

```

RegistrarPaciente=
  (MOTOR.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo,
  FechaNac )
  →
  MOTOR.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo,
  FechaNac) ). FUNCTIONM

```

```

VisualizarPaciente=
  (MOTOR.IntroducirIDPaciente ! (IDPaciente )
  →
  MOTOR.IntroducirIDPaciente ? (IDPaciente )
  →
  MOTOR.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac) ).
  FUNCTIONM

```

```

VisualizarResultados=
  (MOTOR.LimpiarDB ! ( )
  →
  MOTOR.IntroducirIDPaciente ! (IDPaciente )
  →
  MOTOR.IntroducirIDPaciente ? (IDPaciente)
  →

```

```

MOTOR.AdquirirTerapia ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia)
→
MOTOR.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia)
→
MOTOR.TomarResultados ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia, UsuarioClin, UsuarioLab) ).FUNCTIONM

```

```

RealizarDiagnosticoClinico=
(MOTOR.LimpiarDB ! ()
→
MOTOR.AdquirirPosEnfermedad ! (PosEnfermedades )
→
(MOTOR.AdquirirPosEnfermedad ? (PosEnfermedades)
→
(MOTOR.AdquirirSugLab ! (SugLab )
→
MOTOR.AdquirirSugLab ? (SugLab)
→
MOTOR.AdquirirDiagClin ! (PosEnfermedades, EnfermedadClin, CertClin)
→
MOTOR.AdquirirDiagClin ? (PosEnfermedades, EnfermedadClin, CertClin)
→
(MOTOR.AdquirirSugerencia ! (Sugerencia)
→
MOTOR.AdquirirSugerencia ? (Sugerencia)
→
MOTOR.TomarSugerencia ! (Sugerencia) ). FUNCTIONM

```

```

RealizarDiagnosticoLaboratorio=
(MOTOR.LimpiarDB ! ()
→
MOTOR.IntroducirUsuarioClin ! (UsuarioClin )
→
MOTOR.IntroducirUsuarioClin ? (UsuarioClin)
→
MOTOR.IntroducirIDPaciente ! (IDPaciente )
→
MOTOR.IntroducirIDPaciente ? (IDPaciente)
→
MOTOR.TomarSugLab ! (SugLab)
→
MOTOR.AdquirirDiagLab ! (PosEnfermedades, EnfermedadLab, CertLab)
→
MOTOR.AdquirirDiagLab ? (PosEnfermedades EnfermedadLab, CertLab) ).
FUNCTIONM

```

```

Finalizar=
end;

```

End_Functional Aspect FMotor;

Aspecto de persistencia

El MOTOR DE INFERENCIA guarda la información que solicitó a través de los servicios que envió y que fueron atendidas por los otros elementos arquitectónicos del modelo, con el fin de utilizarlos nuevamente, ya sea por que le servirán como parámetros en algunos de los servicios que solicita, o bién de los que envía para que el usuario pueda visualizar cierta información.

Cuando se realiza el registro del paciente, este componente guarda los datos del identificador del paciente, así como su DNI, nombre, dirección, teléfono, email, sexo y fecha de nacimiento con el servicio *Salvar* ().

Los resultados que se obtienen al realizar el diagnóstico clínico, a saber, la enfermedad (clínica) y la certidumbre clínica, el nombre del médico que lo atendió, así como las posibles enfermedades que pudiese tener el paciente y la sugerencia de estudios de laboratorio a realizarse (cuando se desee adquirir el diagnóstico de laboratorio), también son guardadas por el MOTOR DE INFERENCIA mediante el servicio *Salvar* ()

Similarmemente, es necesario guardar los resultados obtenidos al realizar el diagnóstico de laboratorio, por lo que la enfermedad inferida y su certidumbre asociada, así como el nombre del laboratorista son guardados por este componente a través del servicio *Salvar* ().

Una vez obtenida por el MOTOR DE INFERENCIA la terapia que se le aplicará al paciente, ésta es guardada al ejecutarse el servicio *Salvar* ().

Existe otro servicio que se encuentra presente en el aspecto de persistencia, que es utilizado por el mismo MOTOR DE INFERENCIA para poder acceder a los datos, cuando éstos son requeridos como parámetros de los servicios que pertenecen al aspecto funcional de este componente. Este servicio es *ConsultarPaciente*, el cual requiere de un argumento del tipo *identifier* para localizar la información que necesita este componente.

`ConsultarPaciente (IDPaciente)`

Por lo comentado anteriormente, este aspecto importa los atributos que se desean guardar y que se encuentran en el aspecto funcional del MOTOR DE INFERENCIA.

Import FMotor

```
idPaciente: identifier NOT NULL
dni:string
nombre:string
direccion:string
telefono:string
email:string
sexo:char
fechaNacimiento:date
medico: string
laboratorista: string
enfermedadClin:string
```

```

enfermedadLab:string
certidumbreClin:real
certidumbreLab:real
tratamiento:string
posEnfermedades:string
sugerenciaLab:string
sugerencia:string

```

End_Import FMotor

Una característica de este aspecto es que, aunque cuenta con servicios, éstos son internos y no públicos, ya que sólo son utilizados internamente y no para la comunicación con otros elementos arquitectónicos. Por ello, el aspecto de persistencia no usa ninguna interfaz ni contiene *played_roles*.

La especificación del aspecto de persistencia del MOTOR DE INFERENCIA se muestra a continuación:

Persistence Aspect TMotor

Attributes

Variable

Import FMotor

```

idPaciente: identifier NOT NULL
dni:string
nombre:string
direccion:string
telefono:string
email:string
sexo:char
fechaNacimiento:date
medico: string
laboratorista: string
enfermedadClin:string
enfermedadLab:string
certidumbreClin:real
certidumbreLab:real
tratamiento:string
posEnfermedades:string
sugerenciaLab:string
sugerencia:string

```

End_Import FMotor

Services

```

begin ( )

salvar ( )

consultarPaciente (input IDPaciente:identifier)

end;

```

Protocols

TMOTOR = begin.PERSISTENCEM

PERSISTENCEM =

```

        ConsultarPaciente (IDPaciente). TEMP
+
        end;

TEMP =
        Salvar ( ). PERSISTENCEM;
End_Persistence Aspect TMotor

```

Aspecto de distribución

El aspecto de distribución del MOTOR DE INFERENCIA va a ser estático, por lo que no va a proveer servicios de movilidad; sin embargo si deberá tener los servicios *begin* y *end* para establecer el inicio y fin de la ejecución de dicho aspecto, mismos que son utilizados internamente y no para la comunicación con otros elementos arquitectónicos. Por ello, este aspecto no utiliza ninguna interfaz ni contiene *played_roles*.

Debido a que este aspecto requiere de un atributo para la ubicación de la máquina desde la cual se manipula al sistema, el servicio *begin* tiene la siguiente sintaxis:

```
begin (input Location:loc)
```

La especificación del aspecto de distribución de este elemento arquitectónico se muestra a continuación:

Distribution Aspect DLocation

Attributes

Constant

```
location: loc, NOT NULL;
```

Services

```
begin (input Location: loc);
```

Valuations

```
[begin (Location)] location:= Location;
```

```
end;
```

Protocols

```
DLOCATION = begin (Location).DISTRIBUTION;
```

```
DISTRIBUTION = end;
```

```
End_Distribution Aspect DLocation;
```

3.3.1.2 Especificación del Componente Motor de Inferencia

Tras haberse especificado los aspectos necesarios para el componente MOTOR DE INFERENCIA, a continuación se va a definir el *Component_Type InferenceMotor*, el cual será un tipo, de forma tal que el tipo *InferenceMotor* será instanciado para crear el MOTOR DE INFERENCIA que se ha diseñado en el modelo arquitectónico.

InferenceMotor especifica un puerto necesario para comunicarse con otros elementos arquitectónicos. Para definir el puerto, se detalla el nombre (*MotorPort*), el tipo que corresponde a la interfaz declarada (*IInference*) y el *played_role* (*Motor*), que especifica el comportamiento del conjunto de servicios que forman la interfaz que tipa al puerto. El *played_role* se encuentra definidos en el aspecto funcional (*FMotor*) que importa el componente *InferenceMotor*:

```
MotorPort: IInference,  
    Played_Role FMotor.Motor;
```

El componente tipo *InferenceMotor* importa tres aspectos, el funcional (*FMotor*), el de distribución (*DLocation*) y el de persistencia (*TMotor*), con el objetivo de indicar su ubicación dentro del sistema.

```
Function Aspect import FMotor;  
Distribution Aspect import DLocation;  
Persistence Aspect import TMotor
```

En la sección *Initialize* se especifica el inicio de los aspectos (*FMotor* y *DLocation*) que lo forman. Para crear la instancia es necesario dar un valor inicial al atributo requerido *location* de su aspecto de distribución. El atributo *location* tiene la propiedad de *not null*:

```
new (Location: loc)  
    { FMotor.begin ();  
      DLocation.begin (Location);  
      TMotor.begin ( );  
    }
```

En la sección *Destruction* se especifica el término de los aspectos (*FMotor* y *DLocation*) que lo forman:

```
destroy ()  
    { FMotor.end ();  
      DLocation.end ();  
      TMotor.end ( );  
    }
```

El hecho de que se introduzcan el aspecto de persistencia en este componente, hace que se deban sincronizar mediante *weavings* los aspectos funcional y el de persistencia; de forma tal que un *weaving* define que la ejecución de un servicio del aspecto funcional puede generar la invocación de un servicio del aspecto de persistencia. Estos *weavings*

se realizan mediante los operadores *after* y *before*, los cuales establecen una sincronización temporal entre los servicios que participan en ellos.

FMotor.IntroducirIDPaciente (IDPaciente)

after

TMotor.Consultar (IDPaciente)

FMotor.TomarSugLab (SugLab)

after

TMotor.Consultar (IDPaciente)

FMotor.TomarPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

after

TMotor.Consultar (IDPaciente)

FMotor.TomarResultados (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)

after

TMotor.Consultar (IDPaciente)

FMotor.AdquirirDiagLab (PosEnfermedades, EnfermedadLab, CertLab)

after

TMotor.Consultar (IDPaciente)

FMotor.AdquirirTerapia (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)

after

TMotor.Consultar (IDPaciente)

FMotor.IntroducirPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac, IDPaciente)]

before

TMotor.Salvar ()

FMotor.AdquirirDiagClin (PosEnfermedades, EnfermedadClin, CertClin)

before

TMotor.Salvar ()

FMotor.AdquirirDiagLab (PosEnfermedades, EnfermedadLab, CertLab)

before

TMotor.Salvar ()

FMotor.AdquirirTerapia (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)

before

TMotor.Salvar ()

La semántica asociada a esta especificación es la siguiente:

El servicio *introducirIDPaciente* del aspecto funcional *FMotor* se ejecutará **después** del servicio *consultar* del aspecto de persistencia *TMotor*.

El servicio *tomarSugLab* del aspecto funcional *FMotor* se ejecutará **después** del servicio *consultar* del aspecto de persistencia *TMotor*.

El servicio *tomarPaciente* del aspecto funcional *FMotor* se ejecutará **después** del servicio *consultar* del aspecto de persistencia *TMotor*.

El servicio *tomarResultados* del aspecto funcional *FMotor* se ejecutará **después** del servicio *consultar* del aspecto de persistencia *TMotor*.

El servicio *adquirirDiagLab* del aspecto funcional *FMotor* se ejecutará **después** del servicio *consultar* del aspecto de persistencia *TMotor*.

El servicio *adquirirTerapia* del aspecto funcional *FMotor* se ejecutará **después** del servicio *consultar* del aspecto de persistencia *TMotor*.

El servicio *introducirPaciente* del aspecto funcional *FMotor* se ejecutará **antes** del servicio *salvar* del aspecto de persistencia *TMotor*.

El servicio *adquirirDiagClin* del aspecto funcional *FMotor* se ejecutará **antes** del servicio *salvar* del aspecto de persistencia *TMotor*.

El servicio *adquirirDiagLab* del aspecto funcional *FMotor* se ejecutará **antes** del servicio *salvar* del aspecto de persistencia *TMotor*.

El servicio *adquirirTerapia* del aspecto funcional *FMotor* se ejecutará **antes** del servicio *salvar* del aspecto de persistencia *TMotor*.

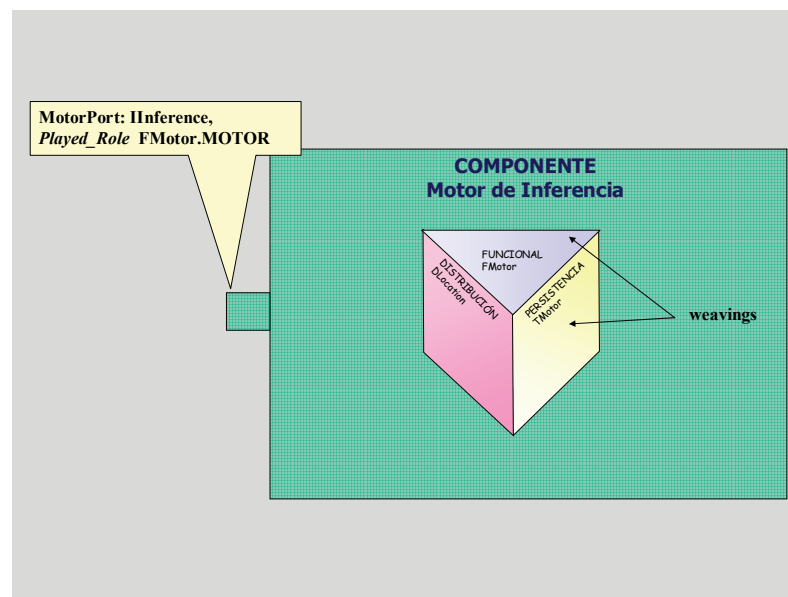


Fig. 26 Componente MOTOR DE INFERENCIA

Con base en el análisis realizado, el componente MOTOR DE INFERENCIA tiene la siguiente especificación:

Component_Type InferenceMotor

Ports

MotorPort: IIference,
Played_ROLE FMotor.Motor;

End_Ports;

Function Aspect import FMotor;

Distribution Aspect import DLocation;

Persistence Aspect import TMotor

Weavings

FMotor.IntroducirIDPaciente (IDPaciente)

after

TMotor.Consultar (IDPaciente)

FMotor.TomarSugLab (SugLab)

after

TMotor.Consultar (IDPaciente)

FMotor.TomarPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

after

TMotor.Consultar (IDPaciente)

FMotor.TomarResultados (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)

after

TMotor.Consultar (IDPaciente)

FMotor.AdquirirDiagLab (PosEnfermedades, EnfermedadLab, CertLab)

after

TMotor.Consultar (IDPaciente)

FMotor.AdquirirTerapia (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)

after

TMotor.Consultar (IDPaciente)

FMotor.IntroducirPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac, IDPaciente)]

before

TMotor.Salvar ()

FMotor.AdquirirDiagClin (PosEnfermedades, EnfermedadClin, CertClin)

before

TMotor.Salvar ()

FMotor.AdquirirDiagLab (PosEnfermedades, EnfermedadLab, CertLab)

before

TMotor.Salvar ()

FMotor.AdquirirTerapia (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)

before

TMotor.Salvar ()

Initialize

new (Location: loc)

```
{ FMotor.begin ();  
  DLocation.begin (Location);  
  TMotor.begin ( );
```

```
}
```

End_Initialize;

Destruction

destroy ()

```
{ FMotor.end ();  
  DLocation.end ();  
  TMotor.end ( );
```

```
}
```

End_Destruction;

End_Connector_Type InferenceMotor;

3.3.1.3 Instanciación del Componente Motor de Inferencia

En la configuración, el servicio *new* se invoca y da valor en la creación de la instancia, especificándose como se muestra a continuación:

```
new InferenceMotor (localhost);
```

3.3.1.4 Especificación de los Aspectos del Componente Base de Conocimientos

Aspecto funcional

El aspecto funcional de este componente interviene en los procesos para realizar el diagnóstico clínico, el diagnóstico de laboratorio y la terapia.

Para el diagnóstico clínico:

Cuando el componente BASE DE CONOCIMIENTOS recibe el servicio *AdquirirPosEnfermedad* con el fin de informar sobre las enfermedades probables que pueda tener el paciente, se encuentra en la necesidad de enviar el servicio *ObtenerSSGG* para solicitar los signos y síntomas de grano grueso, ya que las posibles enfermedades son un atributo derivado del tipo string.

Las posibles enfermedades, en este caso de estudio, denominadas paperas, parotiditis_bacteriana, bronquiolitis_neumonía_crup espasmódico, son atributos derivados que son obtenidos a través de las siguientes derivaciones, que a su vez son cláusulas de Horn con cabeza y que corresponden a las reglas de inferencia que se ejecutan:

```
{parotiditis=true} posEnfermedades:=“paperas_y_parotidis_bacteriana”
```

```
{ira =true} posEnfermedades:=“bronquiolitis_y_neumonía_y_crup_espasmodico”
```

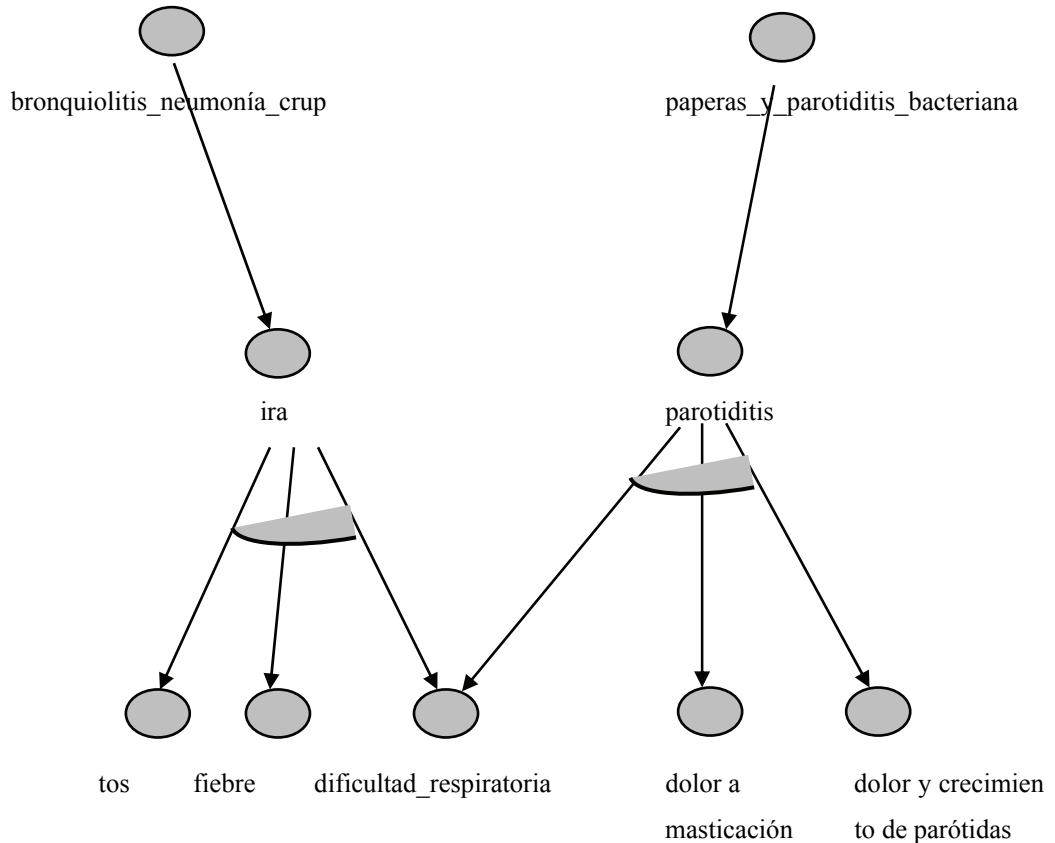
En estas cláusulas de Horn (o reglas de inferencia), las cabezas (o consecuentes de las reglas de inferencia) corresponden a los síndromes denominados parotiditis e ira, mismos que a su vez son los cuerpos (o antecedentes de las reglas de inferencia) de las siguientes cláusulas, que se corresponden con las derivaciones de la especificación PRISMA:

```
parotiditis:= {fiebre=true and dolor_a_masticacion=true and  
dolor_y_crecimiento_de_parotidas=true}
```

```
ira:= { tos=true and fiebre=true and dificultad_respiratoria=true},
```

donde los cuerpos de estas últimas cláusulas son los signos y síntomas de grano grueso, a saber fiebre, dolor a masticación, dolor y crecimientos de parótidas, tos y dificultad_respiratoria, mismos que aparecen en este aspecto como atributos variables.

El grafo que muestra esta situación es:



NOTA:  = and

Fig. 27 Grafo del proceso de inferencia de las posibles enfermedades

Los signos y síntomas de grano grueso los obtiene al recibir el servicio *ObtenerSSGG*. Con estos datos la BASE DE CONOCIMIENTOS infiere dichas enfermedades y las envía a través del servicio *AdquirirPosEnfermedad*.

ObtenerSSGG (**output** Fiebre:bool, **output** DolorMasticacion:bool, **output** DolorCrecimientoParotidas:bool, **output** Tos:bool, **output** DificultadRespiratoria:bool)

AdquirirPosEnfermedad (**output** PosEnfermedades:string)

Acto seguido este componente recibe el servicio *AdquirirSugLab*, por lo que, aplicando las reglas que a continuación se listan, infiere la sugerencia de laboratorio que se hace al usuario con el fin de realizar únicamente los estudios de laboratorio necesarios para un diagnóstico de laboratorio.

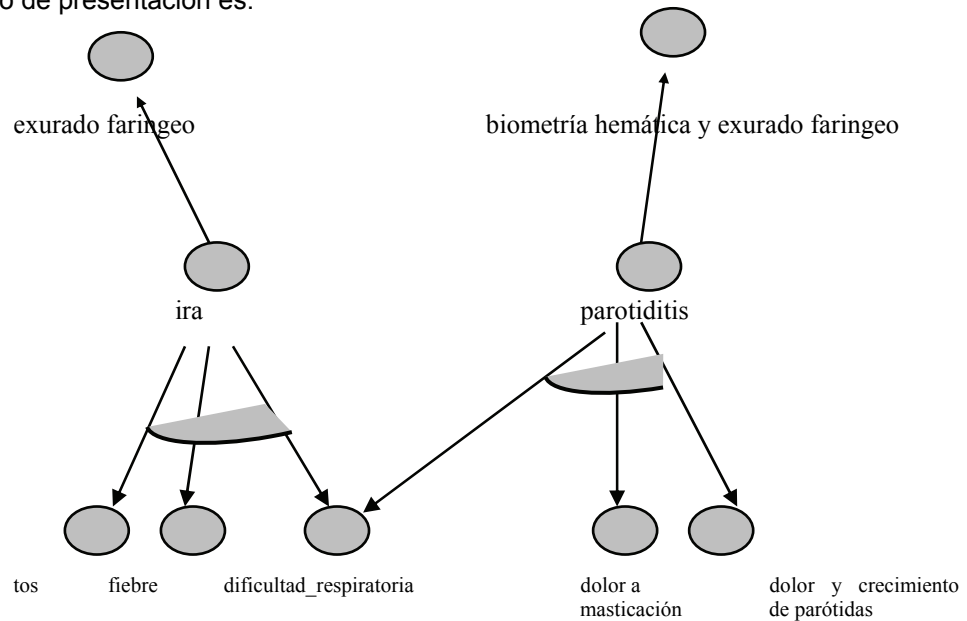
AdquirirSugLab (**output** SugerenciaLab: string)

El atributo sugerenciaLab es un atributo derivado:

{ ira=true } sugerenciaLab:= “biometria hematica y exurado faringeo”

{ parotiditis =true } sugerenciaLab:= “biometria hematica” ,

y su grafo de presentación es:



NOTA:  = and

Fig. 28 Grafo del proceso de inferencia de las sugerencias de laboratorio

En el aspecto funcional del MOTOR DE INFERENCIA se comentó que este componente, solicita la sugerencia de estudios de laboratorio en este momento para que la BASE DE CONOCIMIENTOS no tenga la necesidad de pedirle nuevamente al usuario, los signos y síntomas de grano grueso, ya que las reglas de inferencia para adquirir las posibles enfermedades y las sugerencia de laboratorio, ambas utilizan los mismos hechos, como se puede observar en los dos grafos anteriormente expuestos.

A continuación la BASE DE CONOCIMIENTOS recibe el servicio *AdquirirDiagClin* que permite realizar el diagnóstico clínico del paciente.

En este caso se procede de la misma manera a lo que sucedió con el servicio *AdquirirPosEnfermedad*. Por ello y para no ser tan reiterativo, se simplificarán los comentarios al decir que la BASE DE CONOCIMIENTOS con el servicio *ObtenerSSGF* solicita los signos y síntomas de grano fino del paciente (de solamente estas

enfermedades y no de la totalidad contenidos en la BASE DE CONOCIMIENTOS). Con ello la BASE DE CONOCIMIENTOS infiere el diagnostico clínico, que envía a través del servicio *AdquirirDiagClin*.

ObtenerSSGF (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGraveAModerada, EstrindorLaringeo)

AdquirirDiagClin (PosEnfermedades, EnfermedadClin, CertClin)

Esto último puede apreciarse en el siguiente grafo:

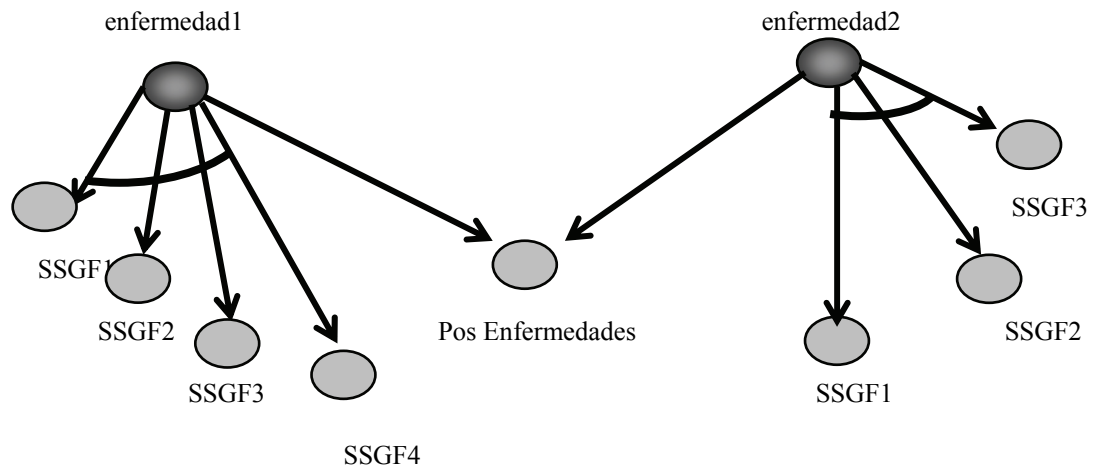


Fig. 29 Grafo del proceso de inferencia para adquirir los signos y síntomas de grano fino

Las reglas de inferencia que se utilizan para ello son:

```
{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  fiebre_mayor_a_38=true and dolor_y_crecimiento_de_parotidas=true and
  dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true ) }
enfermedadClin:= "paperas"
```

```
{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_continua=true and fiebre_mayor_a_38=true and
  dolor_y_crecimiento_de_parotidas=true and dolor_a_masticacion_espontaneo_agudo=true and
  dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true ) }
certidumbreClin:= "1.0"
```

```
{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_mayor_a_38=true and dolor_y_crecimiento_de_parotidas=true and
  dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true ) }
certidumbreClin:= "0.9"
```

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") **and**
(fiebre_continua=true **and** fiebre_mayor_a_38=true **and**
dolor_y_crecimiento_de_parotidas=true **and** dolor_a_masticacion_espontaneo_agudo=true **and**
dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true) }
certidumbreClin:= "0.8"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") **and**
(fiebre_mayor_a_37=true **and** odinofagia_de_2_a_3_dias=true **and**
ardor_de_garganta_intenso=true **and** dolor_y_crecimiento_de_parotidas=true **and**
dolor_a_masticacion_espontaneo_agudo=true) } enfermedadClin:= "parotiditis_bacteriana"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") **and**
(fiebre_continua=true **and** fiebre_mayor_a_37=true **and**
dolor_y_crecimiento_de_parotidas=true **and** dolor_a_masticacion_espontaneo_agudo=true **and**
dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true **and**
odinofagia_de_2_a_3_dias=true **and** ardor_de_garganta_intenso=true) } certidumbreClin:=
"1.0"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") **and**
(fiebre_mayor_a_37=true **and** dolor_y_crecimiento_de_parotidas=true **and**
dolor_a_masticacion_espontaneo_agudo=true **and**
dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true **and**
odinofagia_de_2_a_3_dias=true **and** ardor_de_garganta_intenso=true) } certidumbreClin:=
"0.9"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") **and**
(fiebre_continua=true **and** fiebre_mayor_a_37=true **and**
dolor_y_crecimiento_de_parotidas=true **and** dolor_a_masticacion_espontaneo_agudo=true **and**
odinofagia_de_2_a_3_dias=true **and** ardor_de_garganta_intenso=true) } certidumbreClin:=
"0.8"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") **and**
(apnea_por_periodos_largos=true **and** tos_por_accesos=true **and**
dificultad_respiratoria_grave=true **and** ardor_de_garganta_intenso=true) } enfermedadClin:=
"bronquiolitis"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") **and**
(fiebre_continua=true **and** fiebre_mayor_a_39=true **and** fiebre_de_2_a_3_dias=true **and**
tos_seca=true **and** tos_por_accesos=true **and** dificultad_respiratoria_grave=true **and**
odinofagia_de_2_a_3_dias=true **and** odinofagia_garganta_roja=true **and**
apnea_por_periodos_largos=true **and** ardor_de_garganta_intenso=true) } certidumbreClin:=
"1.0"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") **and**
(fiebre_mayor_a_39=true **and** tos_seca=true **and** tos_por_accesos=true **and**
dificultad_respiratoria_grave=true **and** odinofagia_de_2_a_3_dias=true **and**
apnea_por_periodos_largos=true **and** ardor_de_garganta_intenso=true) } certidumbreClin:=
"0.9"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") **and**
(fiebre_continua=true **and** fiebre_de_2_a_3_dias=true **and** tos_por_accesos=true **and**
dificultad_respiratoria_grave=true **and** odinofagia_garganta_roja=true **and**
apnea_por_periodos_largos=true **and** ardor_de_garganta_intenso=true) } certidumbreClin:=
"0.8"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") **and**
(dificultad_respiratoria_grave_o_polipnea_o_tiraje=true **and**
hipoventilacion_crepitante_pulmonar=true) } enfermedadClin:= "neumonía"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") **and**
(fiebre_continua=true **and** fiebre_mayor_a_39=true **and** fiebre_de_2_a_3_dias=true **and**
tos_con_flema=true **and** tos_frecuente=true **and**

```

dificultad_respiratoria_grave_o_polipnea_o_tiraje=true                                     and
hipoventilacion_crepitante_pulmonar=true } certidumbreClin:= "1.0"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (fiebre_mayor_a_39=true and tos_con_flema=true and tos_frecuente=true and
  dificultad_respiratoria_grave_o_polipnea_o_tiraje=true and
  hipoventilacion_crepitante_pulmonar=true)} certidumbreClin:= "0.90"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  ( fiebre_continua=true and dificultad_respiratoria_grave_o_polipnea_o_tiraje=true and
  hipoventilacion_crepitante_pulmonar=true } certidumbreClin:= "0.85"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (dificultad_respiratoria_grave_a_moderada =true and estrindor_laringeo=true and
  tos_por_accesos=true) } enfermedadClin:= "crup_espasmodico"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  ( fiebre_continua=true and fiebre_mayor_a_39=true and fiebre_de_2_a_3_dias=true and
  rinorea_transparente=true and tos_seca=true and tos_por_accesos=true and
  dificultad_respiratoria_grave_a_moderada =true and estrindor_laringeo=true and
  ardor_de_garganta_intenso=true and odinofagia_de_2_a_3_dias=true and
  odinofagia_garganta_roja=true } certidumbreClin:= "1.0"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  ( fiebre_continua=true and tos_por_accesos=true and
  dificultad_respiratoria_grave_a_moderada =true and estrindor_laringeo=true }
certidumbreClin:= "0.9"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  ( fiebre_mayor_a_39=true and tos_por_accesos=true and
  dificultad_respiratoria_grave_a_moderada =true and estrindor_laringeo=true and
  ardor_de_garganta_intenso=true and odinofagia_de_2_a_3_dias=true } certidumbreClin:=
"0.8"

```

Para el caso del diagnóstico de laboratorio:

LA BASE DE CONOCIMIENTOS recibe el servicio *AdquirirDiagLab* para poder realizar el diagnóstico de laboratorio, por lo que solicita con el servicio *ObtenerEstsLab* los resultados de los estudios de laboratorio realizados al paciente. De esta forma este componente infiere el diagnóstico mediante el servicio *AdquirirDiagLab*

```

AdquirirEstsLab      (output      BiometriaHematicaPositiva:bool,      output
BiometriaHematicaLeucositosis:bool,  output  BiometriaHematicaEusinoxia:bool,  output
ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool)

AdquirirDiagLab (input PosEnfermedades:string, output EnfermedadLab:string, output
CertLab:real)

```

Cabe aclarar que al igual que en el diagnóstico clínico, se solicitan los estudios de laboratorio de solamente estas enfermedades y no de la totalidad de las contenidas en la BASE DE CONOCIMIENTOS, como se aprecia en el siguiente grafo:

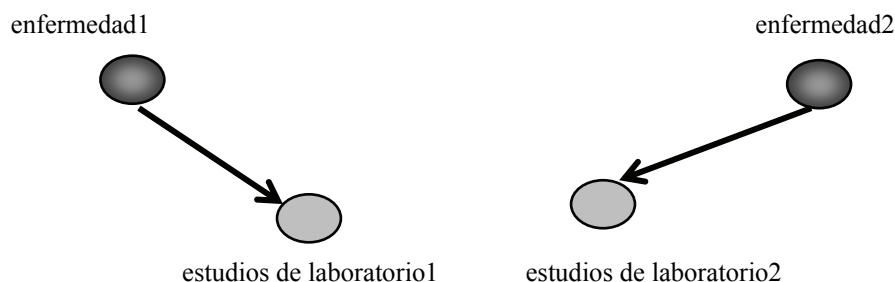


Fig. 30 Grafo del proceso de inferencia para adquirir los estudios de laboratorio

Las reglas de inferencia que se utilizan para inferir el diagnóstico de laboratorio son:

```
{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  (biometria_hematica_positiva =true) } enfermedadLab:= "paperas"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  (biometria_hematica_positiva =true) } certidumbreLab:= "1.0"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  (biometria_hematica_leucositosis =true) } enfermedadLab:= "parotiditis_bacteriana"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  (biometria_hematica_leucositosis =true) } certidumbreLab:= "1.0"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (biometria_hematica_leucositosis =true) } enfermedadLab:= "bronquiolitis"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (biometria_hematica_leucositosis =true) } certidumbreLab:= "1.0"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (exurado_faringeo_estreptococoNeumonice_o_hemofilusInfluenza_o_esquericiaColi=true) }
  enfermedadLab:= "neumonía"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (exurado_faringeo_estreptococoNeumonice_o_hemofilusInfluenza_o_esquericiaColi=true) }
  certidumbreLab:= "1.0"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (biometria_hematica_eusinofilia =true) } enfermedadLab:= "crup_espasmodico"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (biometria_hematica_eusinofilia =true) } certidumbreLab:= "1.0"
```

Ahora bien, es importante resaltar que existen signos y síntomas tanto de grano grueso como de grano fino, que son comunes a varias enfermedades, por lo que, para que el sistema no pregunte al usuario más de una vez sobre dichos datos, los servicios correspondientes muestran la siguiente sintaxis:

Para los signos y síntomas de grano grueso:

```
out obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool,
  output DolorCrecimientoParotidas:bool )
```

Valuations

```
{ fiebre=null and dolor_a_masticacion=null and dolor_y_crecimiento_de_parotidas=null }
  [out obtenerSsggParotiditis (Fiebre, DolorMasticacion, olorCrecimientoParotidas)]
    Fiebre:= fiebre,
    DolorMasticacion:= dolor_a_masticacion,
    DolorCrecimientoParotidas:= dolor_y_crecimiento de parotidas;
```

```
out obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)
```

Valuations

```
{ (fiebre<null) and (tos=null and dificultad_respiratoria=null) }
  [out obtenerSsggIra (Tos, DificultadRespiratoria)]
    Tos:= tos,
    DificultadRespiratoria:= dificultad_respiratoria;
```

Para los signos y síntomas de grano fino

```
out obtenerSsgfPaperas (output FiebreContinua:bool,
  output FiebreMayor38:bool, output DolorCrecimientoParotidas:bool,
  output DolorMasticacionEspontaneoAgudo:bool,
  output DolorMasticacionConBorramientoAnguloMandibular:bool )
```

Valuations

```
{ fiebre_continua=null and fiebre_mayor_a_38=null and
dolor_y_crecimiento_de_parotidas=null and dolor_a_masticacion_espontaneo_agudo=null
and dolor_a_masticacion_con_borramiento_del_angulo_mandibular=null }
  [out obtenerSsgfPaperas (FiebreContinua, FiebreMayor38,
    DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
    DolorMasticacionConBorramientoAnguloMandibular) ]
    FiebreContinua:= fiebre_continua,
    FiebreMayor38:= fiebre_mayor_a_38,
    DolorCrecimientoParotidas:= dolor_y_crecimiento_de_parotidas,
    DolorMasticacionEspontaneoAgudo:=
    dolor_de_masticación_espontaneo_agudo,
    DolorMasticacionConBorramientoAnguloMandibular:=
    dolor_de_masticación_acompañado_con_borramiento_del_angulo_mandibular
    ;
```

```
out obtenerSsgfParotiditisBacteriana (output FiebreMayor37:bool,
  output Odinofagia2A3Dias:bool, output ArdorGargantaIntenso:bool )
```

Valuations

```
{ ( fiebre_continua<null and dolor_y_crecimiento_de_parotidas<null
and dolor_a_masticacion_espontaneo_agudo<null
and dolor_a_masticacion_con_borramiento_del_angulo_mandibular<null) and
(fiebre_mayor_a_37=null and odinofagia_de_2_a_3_dias=null
and ardor_de_garganta_intenso=null) }
  [out obtenerSsgfParotiditisBacteriana (FiebreMayor37, Odinofagia2A3Dias,
    ArdorGargantaIntenso)]
    FiebreMayor37:= fiebre_mayor_a_37,
    Odinofagia2A3Dias:= odinofagia_de_2_a_3_dias,
    ArdorGargantaIntenso:= ardor_de_garganta_intenso;
```

```
out obtenerSsgfBronquiolitis (output FiebreContinua:bool,
  output FiebreMayor38:bool, output Fiebre2A3Dias:bool, output TosSeca:bool,
  output TosPorAccesos:bool, output DificultadRespiratoriaGrave:bool,
  output OdinofagiaGargantaRoja:bool, output Odinofagia2A3Dias:bool,
  output ApneaPeriodosLargos:bool, output ArdorGargantaIntenso:bool)
```

Valuations

```
{ fiebre_continua=null and fiebre_mayor_a_38=null and fiebre_de_2_a_3_dias=null and
tos_seca=null and tos_por_accesos=null and dificultad_respiratoria_grave=null and
odinofagia_garganta_roja=null and odinofagia_de_2_a_3_dias=null and
apnea_por_periodos_largos=null and argor_de_garganta_intenso=null }
```

[out obtenerSsgfBronquiolitis (FiebreContinua, FiebreMayor38, Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso)]

FiebreContinua:= fiebre_continua,
 FiebreMayor38:= fiebre_mayor_a_38,
 Fiebre2A3Dias:= fiebre_de_2_a_3_dias,
 TosSeca:= tos_seca,
 TosPorAccesos:= tos_por_accesos,
 DificultadRespiratoriaGrave:= dificultad_respiratoria_grave,
 OdinofagiaGargantaRoja:= odinofagia_garganta_roja,
 Odinofagia2A3Dias:= odinofagia_de_2_a_3_dias,
 ArdorGargantaIntenso:= ardor_de_garganta_intenso,
 ApneaPeriodosLargos:= apnea_por_periodos_largos;

out obtenerSsgfNeumonia (**output** FiebreMayor39:bool,
output TosConFlema:bool, **output** TosFrecuente:bool,
output DificultadRespiratoriaGravePolipneaTiraje:bool,
output HipoventilacionCrepitantePulmonar:bool)

Valuations

{ (fiebre_continua<>null **and** fiebre_de_2_a_3_dias<>null) **and**
 (fiebre_mayor_a_39=null **and** tos_con_flema=null **and** tos_frecuente=null **and**
 dificultad_respiratoria_grave_o_polipnea_o_tiraje=null **and**
 hipoventilacion_crepitante_pulmonar=null) }

[out obtenerSsgfNeumonia (FiebreMayor39, TosConFlema, TosFrecuente,
 DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]

FiebreMayor39:= fiebre_mayor_a_39,
 TosConFlema:= tos_con_flema,
 TosFrecuente:= tos_frecuente,
 DificultadRespiratoriaGravePolipneaTiraje:=
 dificultad_respiratoria_grave_o_polipnea_o_crepitante_pulmonar,
 HipoventilacionCrepitantePulmonar:= hipoventilacion_crepitante_pulmonar;

out obtenerSsgfCrup (**output** RinoreaTransparente:bool
output DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool)).

Valuations

{ (fiebre_continua<>null **and** fiebre_mayor_a_39<>null **and** fiebre_de_2_a_3_dias<>null
and tos_seca<>null **and** tos_por_accesos<>null **and** ardor_de_garganta_intenso<>null **and**
 odinofagia_de_2_a_3_dias<>null **and** odinofagia_garganta_roja<>null) **and**
 (rinorea_transparente=null **and** dificultad_respiratoria_leve_a_moderada=null **and**
 estrindor_laringeo=null) }

[out obtenerSsgfCrup (RinoreaTransparente, DificultadRespiratoriaGraveAModerada,
 EstrindorLaringeo)]

RinoreaTransparente:= rinorea_transparente;
 DificultadRespiratoriaGraveAModerada:=
 dificultad_respiratoria_grave_a_moderada;
 EstrindorLaringeo:= estrindor_laringeo;

Para la terapia:

El componente BASE DE CONOCIMIENTOS recibe el servicio *AdquirirTerapia* , con ello utiliza los datos del diagnóstico clínico y del de laboratorio, que recibe como parámetros de dicho servicio, para poder inferir la terapia indicada para el caso, utilizando dos clases de reglas de inferencia .

Cabe aclarar que este componente al recibir con dicho servicio los resultados de los dos diagnósticos realizados (i.e el clínico y el de laboratorio), debe tomar en cuenta es

existen dos condiciones que se deben cumplir para poder inferir la terapia. Dichas condiciones consisten en que las enfermedades detectadas en los dos tipos de diagnóstico deben coincidir para obtener un diagnóstico integral, y además de que las certidumbres de ambos deben ser al menos de 0.85 para el caso clínico y del 1.0 para el caso de laboratorio. Estas condiciones forman parte del aspecto funcional del componente mediante una precondition que cuya especificación es::

```
AdquirirTerapia (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)  if
    { ( enfermedadClin=enfermedadLab and
      certidumbreClin >= 0.85 and
      certidumbreLab=1.0 ) or
      ( enfermedadLab=null and certidumbreClin>=0.9 ) }
```

En esta precondition se contempla que si la certeza de la enfermedad obtenida mediante el diagnóstico clínico es por lo menos del 90 %, el usuario puede o no, desear realizar el diagnóstico de laboratorio, adquiriendo de cualquier forma el tratamiento que se le dará al paciente.

Es importante hacer notar que la BASE DE CONOCIMIENTOS cuenta con dos reglas de inferencia con las cuales el sistema toma la decisión de si debe o no realizarse el diagnóstico de laboratorio:

```
{certidumbreClin >=0.9} sugerencia:="no es_necesario_realizar_estudios_de_laboratorio"
```

```
{certidumbreClin <0.9} sugerencia:="es_necesario_realizar_estudios_de_laboratorio"
```

Sin embargo, el sistema es flexible a los deseos del usuario en este caso.

Este último proceso se realiza con la recepción del servicio *AdquirirSugerencia*, y que realiza la sugerencia de si es_necesario_realizar_estudios_de_laboratorio

```
AdquirirSugerencia(Sugerencia)
```

Con las reglas de inferencia primero se obtiene el diagnóstico integral, i.e. que la enfermedad resultante de realizar el diagnóstico clínico es la misma que la que resultó con el diagnóstico de laboratorio, lo que implica que enfermedadClin= enfermedadLab y con ello obtener la enfermedad.

```
{ (enfermedadClin="paperas" and enfermedadLab="paperas" ) or
  (enfermedadClin="paperas" and enfermedadLab = null) } enfermedad:= "paperas"
```

```
{ (enfermedadClin="parotiditis_bacteriana" and enfermedadLab=" parotiditis_bacteriana" ) or
  (enfermedadClin=" parotiditis_bacteriana" and enfermedadLab = null) } enfermedad:=
  "parotiditis_bacteriana"
```

```
{ (enfermedadClin="bronquiolitis" and enfermedadLab=" bronquiolitis" ) or
  (enfermedadClin=" bronquiolitis" and enfermedadLab = null) } enfermedad:=
  "bronquiolitis"
```

```
{ (enfermedadClin="neumonia" and enfermedadLab=" neumonia" ) or
  (enfermedadClin=" neumonia" and enfermedadLab = null) } enfermedad:= "neumonia"
```

```
{ (enfermedadClin="crup_espasmodico" and enfermedadLab=" crup_espasmodico" ) or
(enfermedadClin=" crup_espasmodico" and enfermedadLab = null) } enfermedad:=
"crup_espasmodico"
```

A continuación y dado el encadenamiento de reglas, se aplican las reglas de inferencia para inferir la terapia.

```
{ enfermedad="paperas" } tratamiento:= "antitermicos_hidratacion"
```

```
{ enfermedad="parotidits_bacteriana" } tratamiento:= "antitermicos_hidratacion_antibioticos"
```

```
{ enfermedad="bronquiolitis" } tratamiento:=
"antitermicos_broncodilatadores_oxigenoterapia_nebulizaciones"
```

```
{ enfermedad="neumonia" } tratamiento:=
"antitermicos_hidratacion_antibioticos_antitusigenos"
```

```
{ enfermedad="crup_espasmodico" } tratamiento:=
"humificacion_aireFrio_sedacion_tranquilizarPaciente"
```

Esto último puede apreciarse en el siguiente grafo:

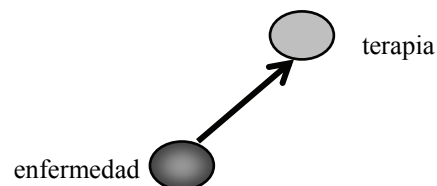


Fig. 31 Grafo del proceso de inferencia de la terapia

Ahora bien, como la BASE DE CONOCIMIENTOS ya cuenta con la información que le fue solicitada, envía con el servicio *AdquirirTerapia* el tratamiento que se le dará al paciente para curar la enfermedad detectada.

```
AdquirirTerapia (input EnfermedadClin:string, input CertClin:real, input
EnfermedadLab:string, input CertLab:real, output Terapia:string )
```

Los servicios antes expuestos se publican en el puerto del componente BASE DE CONOCIMIENTOS denominado *KnowledgePort*, que tiene asociada como signatura la interfaz *IDomain* y que da semántica a su aspecto funcional.

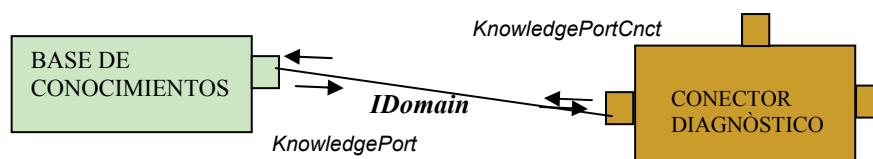


Fig. 32 Esquema del funcionamiento de la Interfaz IDomain

La especificación de dicha interfaz cuenta con la siguiente sintaxis:

Interfaz IDomain

limpiarBD ()

obtenerSsggParotiditis (**output** Fiebre:bool, **output** DolorMasticacion:bool,
output DolorCrecimientoParotidas:bool)

obtenerSsggIra (**output** Tos:bool, **output** DificultadRespiratoria:bool)

obtenerSsgfPaperas (**output** FiebreContinua:bool,
output FiebreMayor38:bool, **output** DolorCrecimientoParotidas:bool,
output DolorMasticacionEspontaneoAgudo:bool,
output DolorMasticacionConBorramientoAnguloMandibular:bool)

obtenerSsgfParotiditisBacteriana (**output** FiebreMayor37:bool,
output Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsgfBronquiolitis (**output** FiebreContinua:bool,
output FiebreMayor38:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool,
output TosPorAccesos:bool, **output** DificultadRespiratoriaGrave:bool,
output OdinofagiaGargantaRoja:bool, **output** Odinofagia2A3Dias:bool,
output ApneaPeriodosLargos:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsgfNeumonia (**output** FiebreMayor39:bool,
output TosConFlema:bool, **output** TosFrecuente:bool,
output DificultadRespiratoriaGravePolipneaTiraje:bool,
output HipoventilacionCrepitantePulmonar:bool)

obtenerSsgfCrup (**output** RinoreaTransparente:bool
output DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool)).

obtenerEstsLabPaperas (**output** BiometriaHematicaPositiva:bool)

obtenerEstsLabParotidisBacteriana (**output** BiometriaHematicaLeucositosis:bool)

obtenerEstsLabBronquiolitis (**output** BiometriaHematicaLeucositosis:bool)

obtenerEstsLabNeumonia (**output**
ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool)

obtenerEstsLabCrup (**output** BiometriaHematicaEusinofilia:bool)

adquirirSugLab (**output** SugLab:string).

adquirirPosEnfermedad (**output** PosEnfermedades:string)

adquirirDiagClin (**input** PosEnfermedades:string, **output** EnfermedadClin:string,
output CertClin:real)

adquirirDiagLab (**input** PosEnfermedades:string, **output** EnfermedadLab:string,
output CertLab:real)

adquirirTerapia (**input** EnfermedadClin:string, **input** CertClin:real,
input EnfermedadLab:string, **input** CertLab:real, **output** Terapia:string)

End_ Interfaz IDomain

Es importante aclarar en este caso que la BASE DE CONOCIMIENTOS aplica las reglas que se encuentran especificadas en el su aspecto funcional como derivaciones, realizando el razonamiento forward (al recibir la petición de obtener la sugerencia de los estudios de laboratorio y de las posibles enfermedades ya que cuenta con la información de los signos y síntomas de grano grueso, así como el adquirir la terapia al contar con los datos de enfermedad clínica y de laboratorio así como sus certidumbres), y el razonamiento backward (al recibir la petición de adquirir el diagnóstico clínico y de esta manera obtener los signos y síntomas de grano grueso y de grano fino, o bién el diagnóstico de laboratorio al obtener los resultados de los estudios de laboratorio realizados).

Cuando la BASE DE CONOCIMIENTOS infiere las dos o más posibles enfermedades, en realidad cuenta con enfermedades de igual certidumbre y muy baja. Para ello utiliza el razonamiento diferencial que discrimina entre las posibles enfermedades para diferenciar de forma más clara entre éstas, incrementando la certidumbre de la solución.

En este proceso, primero se realiza un encadenamiento hacia adelante hasta llegar a dos o más posibilidades diagnósticas (i.e. posibles enfermedades), y posteriormente un encadenamiento hacia atrás para reformar preguntas al usuario o buscar datos en la memoria de trabajo (i.e. signos y síntomas de grano fino para el diagnóstico clínico, o bién estudios de laboratorio para el diagnóstico de laboratorio) que permitan determinar cuál de las posibilidades diagnósticas es la indicada al problema analizado.



Fig. 33 Razonamientos médicos para realizar un diagnóstico

Los razonamientos comentados se muestran como grafos en la siguiente figura:

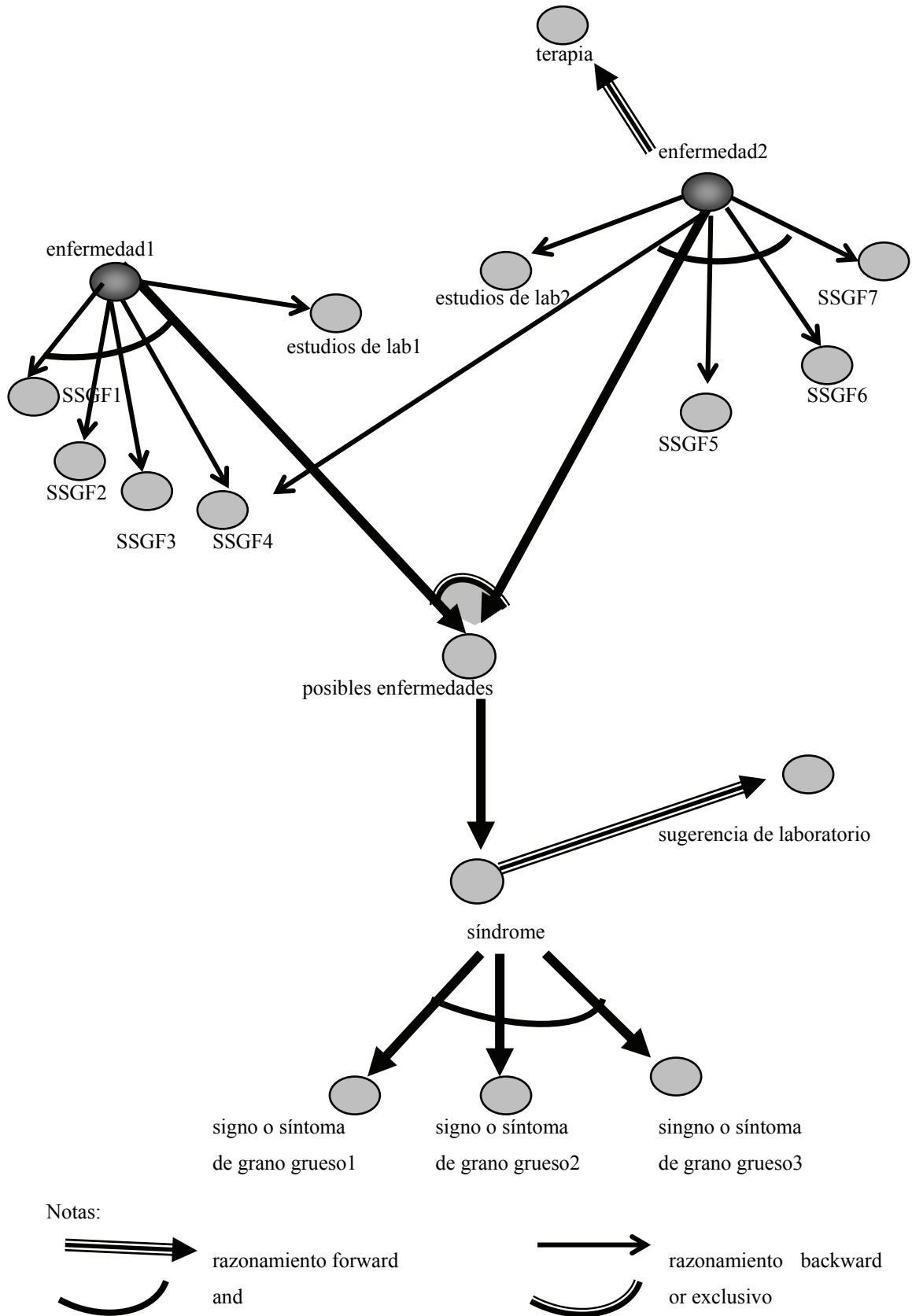


Fig. 34 Grafo de inferencia de los razonamientos forward y backward para realizar el diagnóstico médico

Con base en el análisis realizado en el aspecto funcional de la BASE DE CONOCIMIENTOS, su especificación es la siguiente:

Functional Aspect FBase using IDomain

Attributes

Variable

tos: bool
 fiebre: bool
 dificultad_respiratoria: bool
 dolor_a_masticacion: bool
 dolor_y_crecimiento_de_parotidas: bool
 fiebre_continua:bool
 fiebre_mayor_a_37:bool
 fiebre_mayor_a_38:bool
 fiebre_mayor_a_39:bool
 fiebre_de_2_a_3_dias:bool
 tos_seca:bool
 tos_por_accesos:bool
 tos_con_flema:bool
 tos_frecuente:bool
 dificultad_respiratoria_grave:bool
 dificultad_respiratoria_grave_a_moderada:bool
 dificultad_respiratoria_grave_o_polipnea_o_tiraje:bool
 hipoventilacion_crepitante_pulmonar:bool
 odinofagia_garganta_roja:bool
 odinofagia_de_2_a_3_dias:bool
 ardor_de_garganta_intenso:bool
 apnea_por_periodos_largos:bool
 dolor_y_crecimiento_de_parotidas:bool
 dolor_a_masticación_espontaneo_agudo:bool
 dolor_a_masticación_espontaneo_acompañado_con_borramiento_de_angulo_mandibular:bool
 biometria_hematica_positiva :bool
 biometria_hematica_leucositosis :bool
 biometria_hematica_eusinofilia :bool
 Nbool

Derived

parotiditis:= {fiebre=true and dolor_a_masticacion=true and dolor_y_crecimiento_de_parotidas=true};

ira:= { tos=true and fiebre=true and dificultad_respiratoria=true};

{parotiditis=true} posEnfermedades:= “paperas_y_parotiditis_bacteriana”

{ira =true} posEnfermedades:= “bronquiolitis_neumonía_crup_espasmodico”

{ ira=true } sugerenciaLab:= “biometria hematica y exurado faringeo”

{ parotiditis =true } sugerenciaLab:= “biometria hematica”

{ (posEnfermedades = “paperas_y_parotiditis_bacteriana”) and fiebre_mayor_a_38=true and dolor_y_crecimiento_de_parotidas=true and dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true) }
 enfermedadClin:= “paperas”

{ (posEnfermedades = “paperas_y_parotiditis_bacteriana”) and (fiebre_continua=true and fiebre_mayor_a_38=true and dolor_y_crecimiento_de_parotidas=true and dolor_a_masticacion_espontaneo_agudo=true) } and

```

dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true)      }
certidumbreClin:= "1.0"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_mayor_a_38=true and dolor_y_crecimiento_de_parotidas=true and
  dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true)      }
certidumbreClin:= "0.9"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_continua=true and fiebre_mayor_a_38=true and dolor_y_crecimiento_de_parotidas=true
  and dolor_a_masticacion_espontaneo_agudo=true and
  dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true)      }
certidumbreClin:= "0.8"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_mayor_a_37=true and odinofagia_de_2_a_3_dias=true and
  ardor_de_garganta_intenso=true and dolor_y_crecimiento_de_parotidas=true and
  dolor_a_masticacion_espontaneo_agudo=true) } enfermedadClin:= "parotiditis_bacteriana"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_continua=true and fiebre_mayor_a_37=true and dolor_y_crecimiento_de_parotidas=true
  and dolor_a_masticacion_espontaneo_agudo=true and
  dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true and
  odinofagia_de_2_a_3_dias=true and ardor_de_garganta_intenso=true ) } certidumbreClin:= "1.0"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_mayor_a_37=true and dolor_y_crecimiento_de_parotidas=true and
  dolor_a_masticacion_espontaneo_agudo=true and
  dolor_a_masticacion_acompañado_con_borramiento_del_angulo_mandibular=true and
  odinofagia_de_2_a_3_dias=true and ardor_de_garganta_intenso=true ) } certidumbreClin:= "0.9"

{ (posEnfermedades = "paperas_y_parotiditis_bacteriana") and
  ( fiebre_continua=true and fiebre_mayor_a_37=true and dolor_y_crecimiento_de_parotidas=true
  and dolor_a_masticacion_espontaneo_agudo=true and odinofagia_de_2_a_3_dias=true and
  ardor_de_garganta_intenso=true ) } certidumbreClin:= "0.8"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (apnea_por_periodos_largos=true and tos_por_acesos=true and
  dificultad_respiratoria_grave=true and ardor_de_garganta_intenso=true) } enfermedadClin:=
"bronquiolitis"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  ( fiebre_continua=true and fiebre_mayor_a_39=true and fiebre_de_2_a_3_dias=true and
  tos_seca=true and tos_por_acesos=true and dificultad_respiratoria_grave=true and
  odinofagia_de_2_a_3_dias=true and odinofagia_garganta_roja=true and
  apnea_por_periodos_largos=true and ardor_de_garganta_intenso=true ) } certidumbreClin:= "1.0"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  ( fiebre_mayor_a_39=true and tos_seca=true and tos_por_acesos=true and
  dificultad_respiratoria_grave=true and odinofagia_de_2_a_3_dias=true and
  apnea_por_periodos_largos=true and ardor_de_garganta_intenso=true ) } certidumbreClin:= "0.9"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  ( fiebre_continua=true and fiebre_de_2_a_3_dias=true and tos_por_acesos=true and
  dificultad_respiratoria_grave=true and odinofagia_garganta_roja=true and
  apnea_por_periodos_largos=true and ardor_de_garganta_intenso=true ) } certidumbreClin:= "0.8"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (dificultad_respiratoria_grave_o_polipnea_o_tiraje=true and
  hipoventilacion_crepitante_pulmonar=true) } enfermedadClin:= "neumonía"

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and

```

(fiebre_continua=true **and** fiebre_mayor_a_39=true **and** fiebre_de_2_a_3_dias=true **and**
 tos_con_flema=true **and** tos_frecuente=true **and**
 dificultad_respiratoria_grave_o_polipnea_o_tiraje=true **and**
 hipoventilacion_crepitante_pulmonar=true } certidumbreClin:= “1.0”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (fiebre_mayor_a_39=true **and** tos_con_flema=true **and** tos_frecuente=true **and**
 dificultad_respiratoria_grave_o_polipnea_o_tiraje=true **and**
 hipoventilacion_crepitante_pulmonar=true)} certidumbreClin:= “0.90”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (fiebre_continua=true **and** dificultad_respiratoria_grave_o_polipnea_o_tiraje=true **and**
 hipoventilacion_crepitante_pulmonar=true } certidumbreClin:= “0.85”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (dificultad_respiratoria_grave_a_moderada =true **and** estrindor_laringeo=true **and**
 tos_por_accesos=true) } enfermedadClin:= “crup_espasmodico”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (fiebre_continua=true **and** fiebre_mayor_a_39=true **and** fiebre_de_2_a_3_dias=true **and**
 rinorea_transparente=true **and** tos_seca=true **and** tos_por_accesos=true **and**
 dificultad_respiratoria_grave_a_moderada =true **and** estrindor_laringeo=true **and**
 ardor_de_garganta_intenso=true **and** odinofagia_de_2_a_3_dias=true **and**
 odinofagia_garganta_roja=true } certidumbreClin:= “1.0”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (fiebre_continua=true **and** tos_por_accesos=true **and** dificultad_respiratoria_grave_a_moderada
 =true **and** estrindor_laringeo=true } certidumbreClin:= “0.9”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (fiebre_mayor_a_39=true **and** tos_por_accesos=true **and**
 dificultad_respiratoria_grave_a_moderada =true **and** estrindor_laringeo=true **and**
 ardor_de_garganta_intenso=true **and** odinofagia_de_2_a_3_dias=true } certidumbreClin:= “0.8”
 { (posEnfermedades = “paperas_y_parotiditis_bacteriana”) **and**
 (biometria_hematica_positiva =true) } enfermedadLab:= “paperas”

{ (posEnfermedades = “paperas_y_parotiditis_bacteriana”) **and**
 (biometria_hematica_positiva =true) } certidumbreLab:= “1.0”

{ (posEnfermedades = “paperas_y_parotiditis_bacteriana”) **and**
 (biometria_hematica_leucositosis =true) } enfermedadLab:= “parotidits_bacteriana”

{ (posEnfermedades = “paperas_y_parotiditis_bacteriana”) **and**
 (biometria_hematica_leucositosis =true) } certidumbreLab:= “1.0”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (biometria_hematica_leucositosis =true) } enfermedadLab:= “bronquiolitis”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (biometria_hematica_leucositosis =true) } certidumbreLab:= “1.0”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (exurado_faringeo_estreptococoNeumonice_o_hemofilusInfluenza_o_esquericiaColi=true) }
 enfermedadLab:= “neumonía”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (exurado_faringeo_estreptococoNeumonice_o_hemofilusInfluenza_o_esquericiaColi=true) }
 certidumbreLab:= “1.0”

{ (posEnfermedades = “bronquiolitis_neumonía_crup”) **and**
 (biometria_hematica_eusino-filia =true) } enfermedadLab:= “crup_espasmodico”

```

{ (posEnfermedades = "bronquiolitis_neumonía_crup") and
  (biometria_hematica_eusinofilia =true } certidumbreLab:= "1.0"

{ (enfermedadClin="paperas" and enfermedadLab="paperas" ) or
  (enfermedadClin="paperas" and enfermedadLab = null) } enfermedad:= "paperas"

{ (enfermedadClin="parotiditis_bacteriana" and enfermedadLab=" parotiditis_bacteriana" ) or
  (enfermedadClin=" parotiditis_bacteriana" and enfermedadLab = null) } enfermedad:=
"parotiditis_bacteriana"

{ (enfermedadClin="bronquiolitis" and enfermedadLab=" bronquiolitis" ) or
  (enfermedadClin=" bronquiolitis" and enfermedadLab = null) } enfermedad:=
"bronquiolitis"

{ (enfermedadClin="neumonia" and enfermedadLab=" neumonia" ) or
  (enfermedadClin=" neumonia" and enfermedadLab = null) } enfermedad:= "neumonia"

{ (enfermedadClin="crup_espasmodico" and enfermedadLab=" crup_espasmodico" ) or
  (enfermedadClin=" crup_espasmodico" and enfermedadLab = null) } enfermedad:=
"crup_espasmodico"

{ enfermedad="paperas" } tratamiento:= "antitermicos_hidratacion"

{ enfermedad="parotidits_bacteriana" } tratamiento:= "antitermicos_hidratacion_antibioticos"

{ enfermedad="bronquiolitis" } tratamiento:=
"antitermicos_broncodilatadores_oxigenoterapia_nebulizaciones"

{ enfermedad="neumonia" } tratamiento:= "antitermicos_hidratacion_antibioticos_antitusigenos"

{ enfermedad="crup_espasmodico" } tratamiento:=
"humificacion_aireFrio_sedacion_tranquilizarPaciente"

{ certidumbreClin >=" 0.9" } sugerencia:= "no_es_necesario_realizar_estudios_de_laboratorio"

{ certidumbreClin <" 0.9" } sugerencia:= " es_necesario_realizar_estudios_de_laboratorio"

```

Services

```
begin ()
```

```
in limpiarBD ()
```

```
out obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool,
output DolorCrecimientoParotidas:bool )
```

Valuations

```
{ fiebre=null and dolor_a_masticacion=null and dolor_y_crecimiento_de_parotidas=null }
[out obtenerSsggParotiditis (Fiebre, DolorMasticacion, olorCrecimientoParotidas)]
  Fiebre:= fiebre,
  DolorMasticacion:= dolor_a_masticacion,
  DolorCrecimientoParotidas:= dolor_y_crecimiento de parotidas;
```

```
out obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)
```

Valuations

```
{ (fiebre<>null) and (tos=null and dificultad_respiratoria=null) }
[out obtenerSsggIra (Tos, DificultadRespiratoria)]
  Tos:= tos,
  DificultadRespiratoria:= dificultad_respiratoria;
```

```
out obtenerSsgfPaperas (output FiebreContinua:bool,
```

output FiebreMayor38:bool, **output** DolorCrecimientoParotidas:bool,
output DolorMasticacionEspontaneoAgudo:bool,
output DolorMasticacionConBorramientoAnguloMandibular:bool)

Valuations

{ fiebre_continua=null and fiebre_mayor_a_38=null and
dolor_y_crecimiento_de_parotidas=null and dolor_a_masticacion_espontaneo_agudo=null
and dolor_a_masticacion_con_borramiento_del_angulo_mandibular=null }
[**out** obtenerSsgfPaperas (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)]
FiebreContinua:= fiebre_continua,
FiebreMayor38:= fiebre_mayor_a_38,
DolorCrecimientoParotidas:= dolor_y_crecimiento_de_parotidas,
DolorMasticacionEspontaneoAgudo:=
dolor_de_masticación_espontaneo_agudo,
DolorMasticacionConBorramientoAnguloMandibular:=
dolor_de_masticación_acompañado_con_borramiento_del_angulo_mandibular

out obtenerSsgfParotiditisBacteriana (**output** FiebreMayor37:bool,
output Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool)

Valuations

{ (fiebre_continua<>null **and** dolor_y_crecimiento_de_parotidas<>null
and dolor_a_masticacion_espontaneo_agudo<>null
and dolor_a_masticacion_con_borramiento_del_angulo_mandibular<>null) **and**
(fiebre_mayor_a_37=null **and** odinofagia_de_2_a_3_dias=null
and ardor_de_garganta_intenso=null) }
[**out** obtenerSsgfParotiditisBacteriana (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso)]
FiebreMayor37:= fiebre_mayor_a_37,
Odinofagia2A3Dias:= odinofagia_de_2_a_3_dias,
ArdorGargantaIntenso:= ardor_de_garganta_intenso;

out obtenerSsgfBronquiolitis (**output** FiebreContinua:bool,
output FiebreMayor38:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool,
output TosPorAccesos:bool, **output** DificultadRespiratoriaGrave:bool,
output OdinofagiaGargantaRoja:bool, **output** Odinofagia2A3Dias:bool,
output ApneaPeriodosLargos:bool, **output** ArdorGargantaIntenso:bool)

Valuations

{ fiebre_continua=null **and** fiebre_mayor_a_38=null **and** fiebre_de_2_a_3_dias=null **and**
tos_seca=null **and** tos_por_accesos=null **and** dificultad_respiratoria_grave=null **and**
odinofagia_garganta_roja=null **and** odinofagia_de_2_a_3_dias=null **and**
apnea_por_periodos_largos=null **and** ardor_de_garganta_intenso=null }
[**out** obtenerSsgfBronquiolitis (FiebreContinua, FiebreMayor38, Fiebre2A3Dias,
TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja,
Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso)]
FiebreContinua:= fiebre_continua,
FiebreMayor38:= fiebre_mayor_a_38,
Fiebre2A3Dias:= fiebre_de_2_a_3_dias,
TosSeca:= tos_seca,
TosPorAccesos:= tos_por_accesos,
DificultadRespiratoriaGrave:= dificultad_respiratoria_grave,
OdinofagiaGargantaRoja:= odinofagia_garganta_roja,
Odinofagia2A3Dias:= odinofagia_de_2_a_3_dias,
ArdorGargantaIntenso:= ardor_de_garganta_intenso,
ApneaPeriodosLargos:= apnea_por_periodos_largos;

out obtenerSsgfNeumonia (**output** FiebreMayor39:bool,
output TosConFlema:bool, **output** TosFrecuente:bool,
output DificultadRespiratoriaGravePolipneaTiraje:bool,
output HipoventilacionCrepitantePulmonar:bool)

Valuations

{(fiebre_continua<>null **and** fiebre_de_2_a_3_dias<>null) and

(fiebre_mayor_a_39=null **and** tos_con_flema=null **and** tos_frecuente=null **and** dificultad_respiratoria_grave_o_polipnea_o_tiraje=null **and** hipoventilacion_crepitante_pulmonar=null) }
 [out obtenerSsgfNeumonia (FiebreMayor39, TosConFlema, TosFrecuente, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
 FiebreMayor39:= fiebre_mayor_a_39,
 TosConFlema:= tos_con_flema,
 TosFrecuente:= tos_frecuente,
 DificultadRespiratoriaGravePolipneaTiraje:=
 dificultad_respiratoria_grave_o_polipnea_o_crepitante_pulmonar,
 HipoventilacionCrepitantePulmonar:= hipoventilacion_crepitante_pulmonar;

out obtenerSsgfCrup (**output** RinoreaTransparente:bool
output DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool)).

Valuations

{ (fiebre_continua<>null **and** fiebre_mayor_a_39<>null **and** fiebre_de_2_a_3_dias<>null **and** tos_seca<>null **and** tos_por_accesos<>null **and** ardor_de_garganta_intenso<>null **and** odinofagia_de_2_a_3_dias<>null **and** odinofagia_garganta_roja<>null) **and** (rinorea_transparente=null **and** dificultad_respiratoria_leve_a_moderada=null **and** estrindor_laringeo=null) }

[out obtenerSsgfCrup (RinoreaTransparente, DificultadRespiratoriaGraveAModerada, EstrindorLaringeo)]
 RinoreaTransparente:= rinorea_transparente;
 DificultadRespiratoriaGraveAModerada:=
 dificultad_respiratoria_grave_a_moderada;
 EstrindorLaringeo:= estrindor_laringeo;

out obtenerEstsLabPaperas (**output** BiometriaHematicaPositiva:bool)

Valuations

[out obtenerEstsLabPaperas (BiometriaHematicaPositiva)]
 BiometriaHematicaPositiva:= biometria_hematica_positiva,

out obtenerEstsLabParotidisBacteriana (**output** BiometriaHematicaLeucositosis:bool)

Valuations

[out obtenerEstsLabParotidisBacteriana (BiometriaHematicaLeucositosis)]
 BiometriaHematicaLeucositosis:= biometria_hematica_leucositosis,

out obtenerEstsLabBronquiolitis (**output** BiometriaHematicaLeucositosis:bool)

Valuations

[out obtenerEstsLabBronquiolitis (BiometriaHematicaLeucositosis)]
 BiometriaHematicaLeucositosis:= biometria_hematica_leucositosis,

out obtenerEstsLabNeumonia (**output**
 ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool)

Valuations

[out obtenerEstsLabNeumonia
 (ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)]
 ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:=
 exurado_faringeo_estreptococoNeumonice_o_hemofilusInfluenza_o_esquericiaColi,

out obtenerEstsLabCrup (**output** BiometriaHematicaEusinoFilia:bool)

Valuations

[out obtenerEstsLabCrup (BiometriaHematicaEusinoFilia)]
 BiometriaHematicaEusinoFilia:= biometria_hematica_eusinoFilia,

in adquirirSugLab (**output** SugLab:string).

Valuations

[in adquirirSugLab (SugLab)] SugLab:= sugerencia_lab;

in adquirirSugerencia (**output** Sugerencia:string).

Valuations

[in adquirirSugerencia (Sugerencia)] Sugerencia:= sugerencia;

in adquirirPosEnfermedad (**output** PosEnfermedad:string)

Valuations

[in adquirirPosEnfermedad (PosEnfermedad)]
PosEnfermedad:= posEnfermedad,

in adquirirDiagClin (**input** PosEnfermedades:string, **output** EnfermedadClin:string, **output** CertClin:real)

Valuations

[in adquirirDiagClin (PosEnfermedades, EnfermedadClin, CertClin)]
posEnfermedad:= PosEnfermedad
EnfermedadClin:= enfermedadClin,
CertClin:= certClin;

in adquirirDiagLab (**input** PosEnfermedades:string, **output** EnfermedadLab:string, **output** CertLab:real)

Valuations

[in adquirirDiagLab (PosEnfermedades, EnfermedadLab, CertLab)]
posEnfermedad:= PosEnfermedad
EnfermedadLab:= enfermedadLab,
CertLab:= certLab;

in adquirirTerapia (**input** EnfermedadClin:string, **input** CertClin:real,
input EnfermedadLab:string, **input** CertLab:real **output** Terapia):string

Valuations

[in adquirirTerapia (Terapia)]
enfermedadClin:= EnfermedadClin,
certClin:= CertClin,
enfermedadLab:= EnfermedadLab,
certLab:= CertLab;
Terapia := tratamiento;

end;

Preconditions

adquirirTerapia (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia) **if**
{ (enfermedadClin=enfermedadLab **and**
certidumbreClin >= 0.85 **and**
certidumbreLab=1.0) **or**
(enfermedadLab=null **and** certidumbreClin>=0.9) }

Played_Roles

KNOWLEDGE =

/VISUALIZAR RESULTADOS*/

(IDomain.LimpiarDB ? ()

→

IDomain.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab,CertLab,
Terapia)

→

IDomain.AdquirirTerapia ! (EnfermedadClin, CertClin, EnfermedadLab,CertLab,
Terapia)

+

/*REALIZAR DIAGNOSTICO CLINICO*/

```

(IDomain.LimpiarDB ? ( )
→
IDomain.AdquirirPosEnfermedad ? (PosEnfermedades )
→
(IDomain.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas)
^
IDomain.ObtenerSsggIra ! (Tos, DificultadRespiratoria) )
→
(IDomain.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas )
^
IDomain.ObtenerSsggIra ? (Tos, DificultadRespiratoria) )
→
IDomain.AdquirirPosEnfermedad ! (PosEnfermedades)
→
IDomain.AdquirirSugLab ? (SugLab )
→
IDomain.AdquirirSugLab ! (SugLab)
→
IDomain.AdquirirDiagClin ? (PosEnfermedades, EnfermedadClin, CertClin)
→
(
( IDomain.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IDomain.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(IDomain.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38, Fiebre2A3Dias,
TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja,
Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso) ]
^
IDomain.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
^
IDomain.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(
( IDomain.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IDomain.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(IDomain.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
IDomain.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
^
IDomain.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→

```

```

IDomain.AdquirirDiagClin ! (PosEnfermedades, EnfermedadClin, CertClin)
→
IDomain.AdquirirSugerencia ? (Sugerencia) )
→
IDomain.AdquirirSugerencia ! (Sugerencia) )

+
/*REALIZAR DIAGNOSTICO LABORATORIO*/

( IDomain.LimpiarDB ? ( )
→
IDomain.AdquirirDiagLab ? (PosEnfermedades, EnfermedadLab, CertLab)
→
(
( IDomain.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
IDomain.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
(IDomain.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
^
IDomain.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
^
IDomain.ObtenerEstsLabCrup ! (BiometriaHematicaEusinoFilia) )
)
→
(
( IDomain.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
IDomain.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
(IDomain.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
^
IDomain.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
^
IDomain.ObtenerEstsLabCrup ? (BiometriaHematicaEusinoFilia) )
)
→
IDomain.AdquirirDiag ! (PosEnfermedades, EnfermedadLab, CertLab) )

```

Protocols

FBASE = begin. FUNCTIONB

```

FUNCTIONB =
  VisualizarResultados
+
  RealizarDiagnosticoClinico
+
  RealizarDiagnosticoLaboratorio
+
  Finalizar;

```

```

VisualizarResultados=
( KNOWLEDGE.LimpiarDB ? ( )
→
KNOWLEDGE.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab,
CertLab, Terapia)
→

```

KNOWLEDGE.AdquirirTerapia ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)).FUNCTIONB

RealizarDiagnosticoClinico=

(KNOWLEDGE.LimpiarDB ? ()

→

KNOWLEDGE.AdquirirPosEnfermedad ? (PosEnfermedades)

→

(KNOWLEDGE.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion, DolorCrecimientoParotidas)

^

KNOWLEDGE.ObtenerSsggIra ! (Tos, DificultadRespiratoria))

→

(KNOWLEDGE.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion, DolorCrecimientoParotidas)

^

KNOWLEDGE.ObtenerSsggIra ? (Tos, DificultadRespiratoria))

→

KNOWLEDGE.AdquirirPosEnfermedad ! (PosEnfermedades)

→

KNOWLEDGE.AdquirirSugLab ? (SugLab)

→

KNOWLEDGE.AdquirirSugLab ! (SugLab)

→

KNOWLEDGE.AdquirirDiagClin ? (PosEnfermedades, EnfermedadClin, CertClin)

→

(

(KNOWLEDGE.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular)

^

KNOWLEDGE.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias, ArdorGargantaIntenso))

+

(KNOWLEDGE.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38, Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso)]

^

KNOWLEDGE.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]

^

KNOWLEDGE.ObtenerSsgfCrup ! (RinoreaTransparente, DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))

)

→

(

(KNOWLEDGE.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular)

^

KNOWLEDGE.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias, ArdorGargantaIntenso))

+

(KNOWLEDGE.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38, Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso)]

^

```

KNOWLEDGE.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema,
TosFrecuente, DificultadRespiratoriaGravePolipneaTiraje,
HipoventilacionCrepitantePulmonar)
^
KNOWLEDGE.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
KNOWLEDGE.AdquirirDiagClin ! (PosEnfermedades, EnfermedadClin, CertClin)
→
KNOWLEDGE.AdquirirSugerencia ? (Sugerencia) )
→
KNOWLEDGE.AdquirirSugerencia ! (Sugerencia) ).FUNCTIONB

```

RealizarDiagnosticoLaboratorio=

```

(KNOWLEDGE.LimpiarDB ? ( )
→
KNOWLEDGE.AdquirirDiagLab ? (PosEnfermedades, EnfermedadLab, CertLab)
→
(
( KNOWLEDGE.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
KNOWLEDGE.ObtenerEstsLabParotidisBacteriana !
(BiometriaHematicaLeucositosis) )
+
(KNOWLEDGE.ObtenerEstsLabBronquiolititis ! (BiometriaHematicaLeucositos)
^
KNOWLEDGE.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
KNOWLEDGE.ObtenerEstsLabCrup ! (BiometriaHematicaEusino filia) )
)
→
(
( KNOWLEDGE.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
KNOWLEDGE.ObtenerEstsLabParotidisBacteriana ?
(BiometriaHematicaLeucositosis) )
+
(KNOWLEDGE.ObtenerEstsLabBronquiolititis ? (BiometriaHematicaLeucositos)
^
KNOWLEDGE.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
KNOWLEDGE.ObtenerEstsLabCrup ? (BiometriaHematicaEusino filia) )
)
→
KNOWLEDGE.AdquirirDiagLab ! (PosEnfermedades, EnfermedadLab, CertLab)
).FUNCTIONB

```

Finalizar=

```
end;
```

End_Functional Aspect FBase

Aspecto de distribución

El aspecto de distribución del componente BASE DE CONOCIMIENTOS es estático y sólo especifica la ubicación de la máquina en la que se encuentra. Este aspecto sólo va a proveer los servicios *begin* y *end* para establecer el inicio y fin de la ejecución de dicho aspecto. Estos servicios son utilizados internamente y no para la comunicación con otros elementos arquitectónicos; por ello, dicho aspecto no utiliza ninguna interfaz ni contiene `played_roles`.

Este componente tiene el mismo aspecto de distribución que el MOTOR DE INFERENCIA y EL CONECTOR DIAGNÓSTICO. La especificación de este aspecto se encuentra en el apartado del componente MOTOR DE INFERENCIA.

3.3.1.5 Especificación del Componente Base de Conocimientos

Tras haberse especificado los aspectos necesarios para el componente BASE DE CONOCIMIENTOS, a continuación se va a definir el *Component_Type KnowledgeBase*, el cual será un tipo, de forma tal que el tipo *KnowledgeBase* será instanciado para crear la BASE DE CONOCIMIENTOS que se ha diseñado en el modelo arquitectónico.

KnowledgeBase especifica un puerto necesario para comunicarse con otros elementos arquitectónicos. Para definir el puerto, se detalla el nombre (*KnowledgePort*), el tipo que corresponde a la interfaz declarada (*IDominio*,) y el `played_role` (*Knowledge*) que se le asocia para especificar el comportamiento del conjunto de servicios que forman la interfaz que tipa al puerto. El `played_role` se encuentra definido en el aspecto funcional (*FBase*) que importa el componente *KnowledgeBase*:

```
KnowledgePort: IDominio,  
    Played_Role FBase.Knowledge;
```

El componente tipo *KnowledgeBase* importa dos aspectos, el funcional (*FBase*) y el de distribución (*DLocation*) con el objetivo de indicar su ubicación dentro del sistema.

```
Function Aspect import FBase;  
Distribution Aspect import DLocation;
```

En la sección *Initialize* se especifica el inicio de los aspectos (*FBase* y *DLocation*) que lo forman. Para crear la instancia es necesario dar un valor inicial al atributo requerido `location` de su aspecto de distribución. El atributo `location` tiene la propiedad de *not null*:

```
new (Location: loc)  
    { FBase.begin ();
```

```

        DLocation.begin (Location);
    }

```

En la sección *Destruction* se especifica el término de los aspectos (*FBase* y *DLocation*) que lo forman:

```

destroy ()
    { FBase.end ();
      DLocation.end ();
    }

```

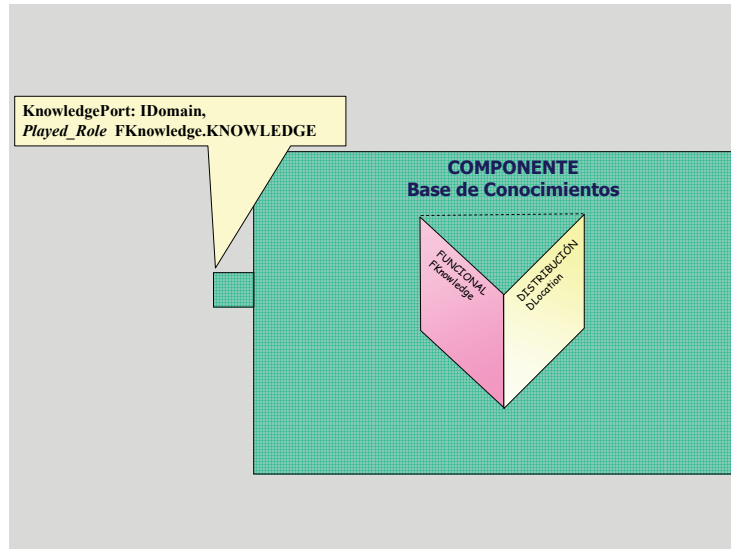


Fig. 35 Componente BASE DE CONOCIMIENTOS

Con base en el análisis realizado, el componente BASE DE CONOCIMIENTOS tiene la siguiente especificación:

Component_Type KnowledgeBase

Ports

KnowledgePort: IDominio,
Played_Role FBase.KNOWLEDGE;

End_Ports;

Function Aspect import FBase;

Distribution Aspect import DLocation;

Initialize

```

    new (Location: loc)
    { FBase.begin();
      DLocation.begin (Location);
    }

```

End_Initialize;

Destruction

```

    destroy ()
    { FBase.end();
      DLocation.end( );
    }

```

End_Destruction;

End_Connector_Type KnowledgeBase;

3.3.1.6 Instanciación del Componente Base de Conocimientos

En la configuración, el servicio *new* se invoca y da valor en la creación de la instancia, especificándose de la siguiente manera:

```
new KnowledgeBase (localhost);
```

3.3.1.7 Especificación de los aspectos del Conector Diagnóstico

Aspecto de coordinación

El aspecto de coordinación, involucra la sincronización de los servicios recibidos y enviados entre los elementos arquitectónicos que el conector coordina, misma que se muestra en el protocolo de este aspecto

El funcionamiento de los flujos de los servicios que participan en el protocolo puede observarse en la siguiente figura:

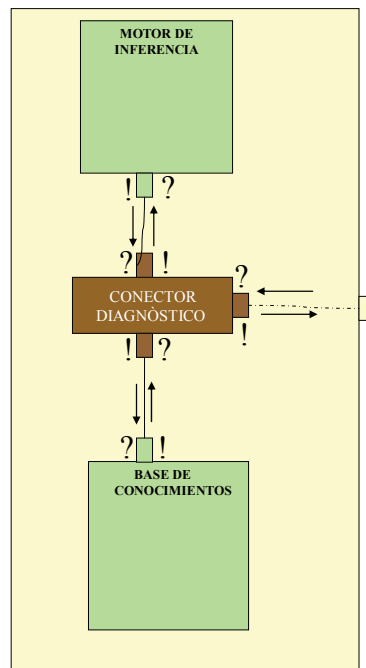


Fig. 36 Esquema que muestra el flujo de los servicios a través de los puertos del conector Diagnóstico

El protocolo del comportamiento del conector DIAGNÓSTICO se especifica de la siguiente manera:

CDIAG = begin.COORDD

COORDD = IntroducirLoginPassword.P0

P0 = IntroducirSelección.OPTIONAL

OPTIONAL =

RegistrarPaciente.P0

+

VisualizarPaciente.P0

+

RealizarDiagnosticoClinico.P0

+

RealizarDiagnosticoLaboratorio.P0

+

VisualizarResultados.P0

+

Finalizar;

Este protocolo se describe en las tablas y en el grafo de transición que a continuación se muestran.

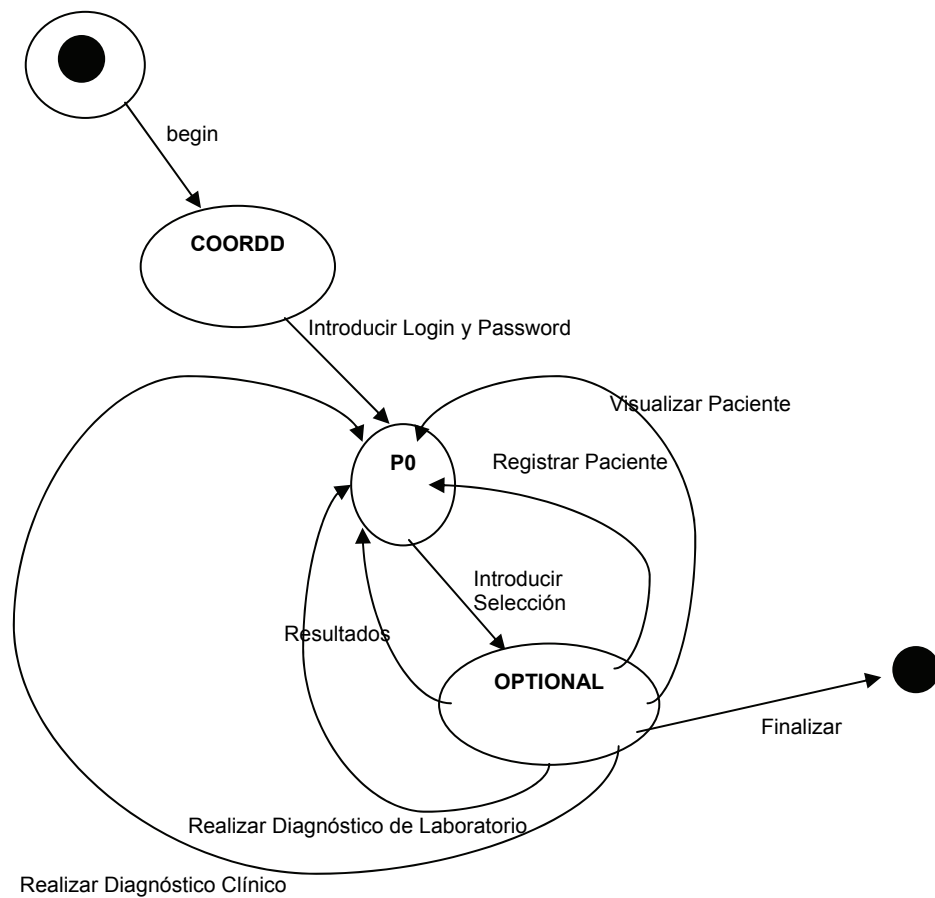


Fig. 37 Grafo de transición del protocolo del conector Diagnóstico

El conector DIAGNÓSTICO inicia su ejecución cuando el sistema obtiene el nombre del usuario y su contraseña del usuario.

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (j)	nombre del servicio
MotorPortCnct	UsersPortCnct	introducirLoginPasword (output Login:string, output Password:string)
UsersPortCnct	MotorPortCnct	introducirLoginPasword (output Login:string, output Password:string)

Acto seguido se obtiene del usuario la opción que realizará el sistema:

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (j)	nombre del servicio
MotorPortCnct	UsersPortCnct	introducirSeleccion (output Seleccion:int)
UsersPortCnct	MotorPortCnct	introducirSeleccion (output Seleccion:int)

- Si se ha elegido la opción de “Registrar Paciente”:

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (j)	nombre del servicio
MotorPortCnct	UsersPortCnct	introducirPaciente (output IDPaciente:identifier, output DNI:string, output Nombre:string, output Dirección:string, output Tel:string, output Email:string, output Sexo:char, output FechaNac:date).
UsersPortCnct	MotorPortCnct	introducirPaciente (output IDPaciente:identifier, output DNI:string, output Nombre:string, output Dirección:string, output Tel:string, output Email:string, output Sexo:char, output FechaNac:date).

- Si se ha seleccionado “Visualizar los datos del Paciente”:

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (j)	nombre del servicio
MotorPortCnct	UsersPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UsersPortCnct	MotorPortCnct	introducirIDPaciente (output IDPaciente:identifier)
MotorPortCnct	UsersPortCnct	tomarPaciente (input DNI:string, input Nombre:string, input Dirección:string, input Tel:string, input Email:string, input Sexo:char, input FechaNac:date)

- Si se desea “Realizar el diagnóstico clínico”:

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (i)	nombre del servicio
MotorPortCnct	UsersPortCnct	introducirUsuarioClin (output UsuarioClin:string)
UsersPortCnct	MotorPortCnct	introducirUsuarioClin (output UsuarioClin:string)
MotorPortCnct	UsersPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UsersPortCnct	MotorPortCnct	introducirIDPaciente (output IDPaciente:identifier)
MotorPortCnct	KnowledgePortCnct	AdquirirPosEnfermedad (output PosEnfermedades:string)
KnowledgePortCnct	UsersPortCnct	obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool, output DolorCrecimientoParotidas:bool) ^ obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)
UsersPortCnct	KnowledgePortCnct	obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool, output DolorCrecimientoParotidas:bool) ^ obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)
KnowledgePortCnct	MotorPortCnct	adquirirPosEnfermedad (output PosEnfermedades:string)
MotorPortCnct	KnowledgePortCnct	adquiriSugLab (output SugerenciaLab)
KnowledgePortCnct	MotorPortCnct	adquiriSugLab (output SugerenciaLab)
MotorPortCnct	KnowledgePortCnct	adquirirDiagClin (input PosEnfermedades:string, output EnfermedadClin:string, output CertClin:real)
KnowledgePortCnct	UsersPortCnct	(obtenerSsgfPaperas (output FiebreContinua:bool, output FiebreMayor38:bool, output DolorCrecimientoParotidas:bool, output DolorMasticacionEspontaneoAgudo:bool, output DolorMasticacionConBorramientoAnguloMandibular:bool) ^ obtenerSsgfParotiditisBacteriana (output FiebreMayor37:bool, output Odinofagia2A3Dias:bool, output ArdorGargantaIntenso:bool)) + (obtenerSsgfBronquiolitis (output FiebreContinua:bool, output FiebreMayor38:bool, output Fiebre2A3Dias:bool, output TosSeca:bool, output TosPorAccesos:bool, output DificultadRespiratoriaGrave:bool, output OdinofagiaGargantaRoja:bool, output Odinofagia2A3Dias:bool, output ApneaPeriodosLargos:bool, output ArdorGargantaIntenso:bool) ^ obtenerSsgfNeumonia (output FiebreMayor39:bool, output TosConFlema:bool, output TosFrecuente:bool, output DificultadRespiratoriaGravePolipneaTiraje:bool, output HipoventilacionCrepitantePulmonar:bool) ^ obtenerSsgfCrup (output RinoreaTransparente:bool, output DificultadRespiratoriaGarveAModerada:bool, output EstrindorLaringeo:bool))

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (i)	nombre del servicio
UsersPortCnct	KnowledgePortCnct	(obtenerSsgfPaperas (output FiebreContinua:bool, output FiebreMayor38:bool, output DolorCrecimientoParotidas:bool, output DolorMasticacionEspontaneoAgudo:bool, output DolorMasticacionConBorramientoAnguloMandibular:bool) ^ obtenerSsgfParotiditisBacteriana (output FiebreMayor37:bool, output Odinofagia2A3Dias:bool, output ArdorGargantaIntenso:bool)) + (obtenerSsgfBronquiolitis (output FiebreContinua:bool, output FiebreMayor38:bool, output Fiebre2A3Dias:bool, output TosSeca:bool, output TosPorAccesos:bool, output DificultadRespiratoriaGrave:bool, output OdinofagiaGargantaRoja:bool, output Odinofagia2A3Dias:bool, output ApneaPeriodosLargos:bool, output ArdorGargantaIntenso:bool) ^ obtenerSsgfNeumonia (output FiebreMayor39:bool, output TosConFlema:bool, output TosFrecuente:bool, output DificultadRespiratoriaGravePolipneaTiraje:bool, output HipoventilacionCrepitantePulmonar:bool) ^ obtenerSsgfCrup (output RinoreaTransparente:bool , output DificultadRespiratoriaGarveAModerada:bool, output EstrindorLaringeo:bool))
KnowledgePortCnct	MotorPortCnct	adquirirDiagClin (input PosEnfermedades:string, output EnfermedadClin:string, output CertClin:real)
MotorPortCnct	KnowledgePortCnct	adquirirSugerencia (output Sugerencia:string)
KnowledgePortCnct	MotorPortCnct	adquirirSugerencia (output Sugerencia:string)
MotorPortCnct	UsersPortCnct	tomarSugerencia (output Sugerencia:string)

- Si la selección fue “Adquirir el diagnostico de laboratorio”:

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (i)	nombre del servicio
MotorPortCnct	UsersPortCnct	introducirUsuarioLab (output UsuarioLab:string)
UsersPortCnct	MotorPortCnct	introducirUsuarioLab (output UsuarioLab:string)
MotorPortCnct	UsersPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UsersPortCnct	MotorPortCnct	introducirIDPaciente (output IDPaciente:identifier)
MotorPortCnct	UsersPortCnct	tomarSugLab (output SugLab:string)
MotorPortCnct	KnowledgePortCnct	adquirirDiagLab (input PosEnfermedades:string, output EnfermedadLab:string, output CertLab:real)

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (i)	nombre del servicio
KnowledgePortCnct	UsersPortCnct	(obtenerEstsLabPaperas (output BiometriaHematicaPositiva:bool) ^ obtenerEstsLabParotidisBacteriana (output BiometriaHematicaLeucositosis:bool)) + (obtenerEstsLabBronquiolitis (output BiometriaHematicaLeucositosis:bool) ^ obtenerEstsLabNeumonia (output ExuradoFaringeoEstreptococoNeumonicHemofilusInfluentaEsquericiaColi:bool) ^ obtenerEstsLabCrup (output BiometriaHematicaEusinofilia:bool))
UsersPortCnct	KnowledgePortCnct	(obtenerEstsLabPaperas (output BiometriaHematicaPositiva:bool) ^ obtenerEstsLabParotidisBacteriana (output BiometriaHematicaLeucositosis:bool)) + (obtenerEstsLabBronquiolitis (output BiometriaHematicaLeucositosis:bool) ^ obtenerEstsLabNeumonia (output ExuradoFaringeoEstreptococoNeumonicHemofilusInfluentaEsquericiaColi:bool) ^ obtenerEstsLabCrup (output BiometriaHematicaEusinofilia:bool))
KnowledgePortCnct	MotorPortCnct	adquirirDiagLab (input PosEnfermedades:string, output EnfermedadLab:string, output CertLab:real)

- Para el caso de “Visualizar resultados del diagnóstico”:

puerto del que recibe el servicio (?)	puerto por el que reenvía el servicio (i)	nombre del servicio
MotorPortCnct	UsersPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UsersPortCnct	MotorPortCnct	introducirIDPaciente (output IDPaciente:identifier)
MotorPortCnct	KnowledgePortCnct	adquirirTerapia (input EnfermedadClin:string, input CertClin:real, input EnfermedadLab:string, input CertLab:real output Terapia:string)
KnowledgePortCnct	MotorPortCnct	adquirirTerapia (input EnfermedadClin:string, input CertClin:real, input EnfermedadLab:string, input CertLab:real output Terapia:string)
MotorPortCnct	UsersPortCnct	tomarResultados (input EnfermedadClin:string, input CertClin:real, input EnfermedadLab:string, input CertLab:real, input Terapia:string, input UsuarioClin:string, input UsuarioLab:string)

Los servicios antes expuestos se publican en los puertos del conector denominados *MotorPortCnct*, *KnowledgePortCnct* y *UsersPortCnct* que tienen asociados como signatura las interfaces *IIInference*, *IDomain* e *IDiagnostical* respectivamente, y que dan semántica a su aspecto de coordinación.

Es conveniente visualizar los puertos del conector DIAGNÓSTICO, de acuerdo a la siguiente tabla y figura:

elemento arquitectónico del que recibe/reenvía el servicio	puerto del conector	played_role del puerto	interfaz
motor de inferencia	MotorPortCnct	MOTOR	IIInference
base de conocimientos	KnowledgePortCnct	KNOWLEDGE	IDomain
sistema diagnosticador	UsersPortCnct	USERS	IDiagnostical

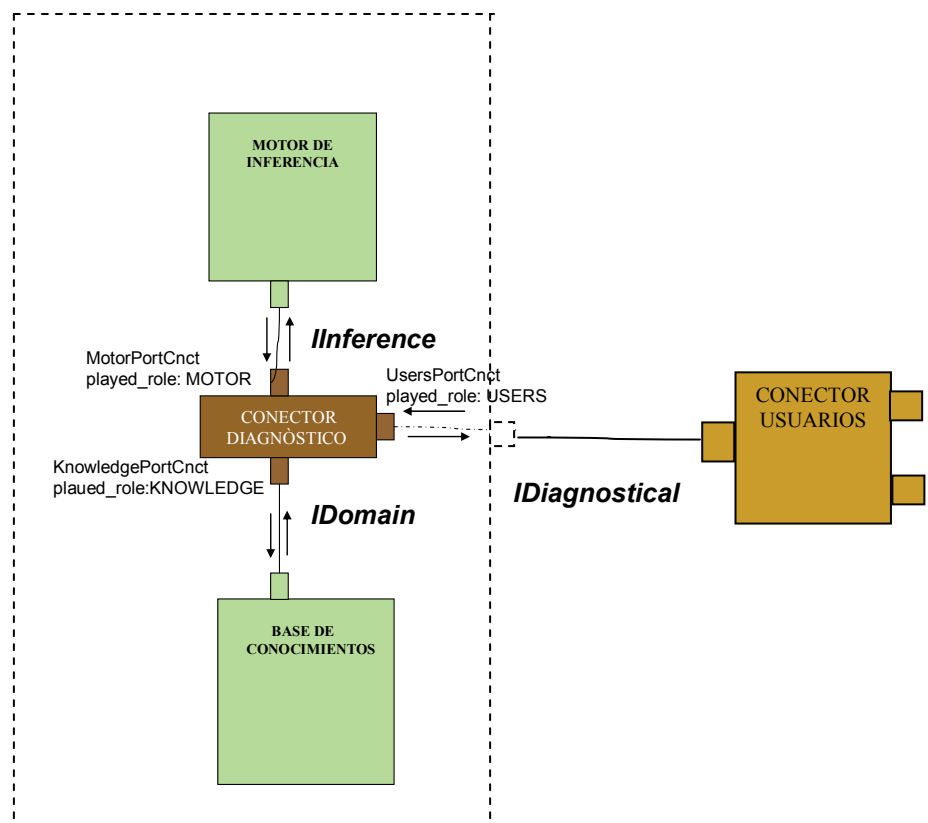


Fig. 38 Puertos del conector DIAGNÓSTICO

La especificación de la interfaz *IDiagnostical* muestra la siguiente sintaxis:

Interfaz IDiagnostical

introducirLoginPassword (**output** Login:string, **output** Password:string)

introducirSeleccion (**output** Seleccion:int)

introducirPaciente (**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string,
output Direccion:string, **output** Tel:string, **output** Email:string, **output**
Sexo:char, **output** FechaNac:date)

introducirIDPaciente (**output** IDPaciente:identifier)

introducirUsuarioClin (**output** UsuarioClin:string)

introducirUsuarioLab (**output** UsuarioLab:string)

obtenerSsggParotiditis (**output** Fiebre:bool, **output** DolorMasticacion:bool, **output**
DolorCrecimientoParotidas:bool)

obtenerSsggIra (**output** Tos:bool, **output** DificultadRespiratoria:bool)

obtenerSsgfPaperas (**output** FiebreContinua:bool,
output FiebreMayor38:bool, **output** DolorCrecimientoParotidas:bool,
output DolorMasticacionEspontaneoAgudo:bool,
output DolorMasticacionConBorramientoAnguloMandibular:bool)

obtenerSsgfParotiditisBacteriana (**output** FiebreMayor37:bool,
output Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsgfBronquiolitis (**output** FiebreContinua:bool,
output FiebreMayor38:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool,
output TosPorAccesos:bool, **output** DificultadRespiratoriaGrave:bool,
output OdinofagiaGargantaRoja:bool, **output** Odinofagia2A3Dias:bool,
output ApneaPeriodosLargos:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsgfNeumonia (**output** FiebreMayor39:bool,
output TosConFlema:bool, **output** TosFrecuente:bool,
output DificultadRespiratoriaGravePolipneaTiraje:bool,
output HipoventilacionCrepitantePulmonar:bool)

obtenerSsgfCrup (**output** RinoreaTransparente:bool
output DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool)).

obtenerEstsLabPaperas (**output** BiometriaHematicaPositiva:bool)

obtenerEstsLabParotidisBacteriana (**output** BiometriaHematicaLeucositosis:bool)

obtenerEstsLabBronquiolitis (**output** BiometriaHematicaLeucositosis:bool)

obtenerEstsLabNeumonia (**output**
ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool)

obtenerEstsLabCrup (**output** BiometriaHematicaEusinoofilia:bool)

tomarPaciente (**input** DNI:string, **input** Nombre:string, **input** Direccion:string, **input** Tel:string,
input Email:string, **input** Sexo:char, **input** FechaNac:date)

tomarResultados (**input** EnfermedadClin:string, **input** CertClin:real, **input**
EnfermedadLab:string, **input** CertLab:real, **input** Terapia, **input**
UsuarioClin:string, **input** UsuarioLab:string)

tomarSugLab (**input** SugLab:string)

tomarSugerencia (**input** Sugerencia:string)

End_ Interfaz IDiagnostical

La especificación de las interfaces *IInference* e *IDomain* se realizaron en los apartados de los componentes Motor de Inferencia y Base de Conocimientos respectivamente.

Como resultado de este análisis, la especificación del aspecto de coordinación del conector DIAGNÓSTICO es el siguiente:

Coordination Aspect CDiag using IInference, IDomain, IDiagnostical Services

begin ()

in/out limpiarBD ()

in/out introducirLoginPassword (**output** Login: string , **output** Password: string)

in/out introducirSeleccion (**output** Seleccion:int)

in/out introducirPaciente (**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string, **output** Direccion:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date)

in/out introducirIDPaciente (**output** IDPaciente:identifier)

in/out introducirUsuarioClin (**output** UsuarioClin:string)

in/out introducirUsuarioLab (**output** UsuarioLab:string)

in/out obtenerSsggParotiditis (**output** Fiebre:bool, **output** DolorMasticacion:bool, **output** DolorCrecimientoParotidas:bool)

in/out obtenerSsggIra (**output** Tos:bool, **output** DificultadRespiratoria:bool)

in/out obtenerSsgfPaperas (**output** FiebreContinua:bool, **output** FiebreMayor38:bool, **output** DolorCrecimientoParotidas:bool, **output** DolorMasticacionEspontaneoAgudo:bool, **output** DolorMasticacionConBorramientoAnguloMandibular:bool)

in/out obtenerSsgfParotiditisBacteriana (**output** FiebreMayor37:bool, **output** Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool)

in/out obtenerSsgfBronquiolitis (**output** FiebreContinua:bool, **output** FiebreMayor38:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool, **output** TosPorAccesos:bool, **output** DificultadRespiratoriaGrave:bool, **output** OdinofagiaGargantaRoja:bool, **output** Odinofagia2A3Dias:bool, **output** ApneaPeriodosLargos:bool, **output** ArdorGargantaIntenso:bool)

in/out obtenerSsgfNeumonia (**output** FiebreMayor39:bool, **output** TosConFlema:bool, **output** TosFrecuente:bool, **output** DificultadRespiratoriaGravePolipneaTiraje:bool, **output** HipoventilacionCrepitantePulmonar:bool)

in/out obtenerSsgfCrup (**output** RinoreaTransparente:bool, **output** DificultadRespiratoriaGarveAModerada:bool,

```

        output EstrindorLaringeo:bool) ).

in/out obtenerEstsLabPaperas (output BiometriaHematicaPositiva:bool)

in/out obtenerEstsLabParotidisBacteriana (output BiometriaHematicaLeucositosis:bool)

in/out obtenerEstsLabBronquiolitis (output BiometriaHematicaLeucositosis:bool)

in/out obtenerEstsLabNeumonia (output
ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool)

in/out obtenerEstsLabCrup (output BiometriaHematicaEusinofilia:bool)

in/out adquirirSugLab (output SugLab:string)

in/out adquirirSugerencia (output Sugerencia:string)

in/out adquirirPosEnfermedad (output PosEnfermedades:string)

in/out adquirirDiagClin (input PosEnfermedades:string, output EnfermedadClin:string,
output CertClin:real)

in/out adquirirDiagLab (input PosEnfermedades:string, output EnfermedadLab:string,
output CertLab:real)

in/out adquirirTerapia (input EnfermedadClin:string, input CertClin:real,
input EnfermedadLab:string, input CertLab:real, output Terapia:string)

in/out tomarPaciente (input DNI:string, input Nombre:string, input Direccion:string,
input Tel:string, input Email:string, input Sexo:char, input FechaNac:date)

in/out tomarResultados (input EnfermedadClin:string, input CertClin:real,
input EnfermedadLab:string, input CertLab:real, input Terapia,
input UsuarioClin:string, input UsuarioLab:string)

in/out tomarSugLab (input SugLab:string)

in/out tomarSugerencia (input Sugerencia:string)

end;

```

Played_Roles

```

MOTOR =
    Inference.IntroducirLoginPassword ? (Login, Password )
    →
    Inference.IntroducirLoginPassword ! (Login, Password)
    →
    Inference.IntroducirSeleccion ? (Seleccion )
    →
    Inference.IntroducirSeleccion ! (Seleccion)
→
                                                                    /*INTRODUCIR PACIENTE*/
(
( Inference.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac )
→
Inference.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac) )
+
                                                                    /*VISUALIZAR PACIENTE*/

```

(IInference.IntroducirIDPaciente ? (IDPaciente)
→
IInference.IntroducirIDPaciente ! (IDPaciente)
→
IInference.TomarPaciente ? (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))

+ /VISUALIZAR RESULTADOS*/

(IInference.LimpiarDB ? ()
→
IInference.IntroducirIDPaciente ? (IDPaciente)
→
IInference.IntroducirIDPaciente ! (IDPaciente)
→
IInference.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab. CertLab, Terapia)
→
IInference.AdquirirTerapia ! EnfermedadClin, CertClin, EnfermedadLab. CertLab, Terapia)
→
IInference.TomarResultados ? (EnfermedadClin, CertClin, EnfermedadLab. CertLab, Terapia, UsuarioClin, UsuarioLab))

+ /*REALIZAR DIAGNOSTICO CLINICO*/

(IInference.LimpiarDB ? ()
→
IInference.AdquirirUsuarioClin ? (UsuarioClin)
→
IInference.AdquirirUsuarioClin ! (UsuarioClin)
→
IInference.AdquirirIDPaciente ? (IDPaciente)
→
IInference.AdquirirIDPaciente ! (IDPaciente)
→
IInference.AdquirirPosEnfermedad ? (PosEnfermedades)
→
IInference.AdquirirPosEnfermedad ! (PosEnfermedades)
→
IInference.AdquirirSugLab ? (SugLab)
→
IInference.AdquirirSugLab ! (SugLab)
→
IInference.AdquirirDiagClin ? (PosEnfermedades, EnfermedadClin, CertClin)
→
IInference.AdquirirDiagClin ! (PosEnfermedades, EnfermedadClin, CertClin)
→
IInference.AdquirirSugerencia ? (Sugerencia)
→
IInference.AdquirirSugerencia ! (Sugerencia)
→
IInference.TomarSugLab ? (Sugerencia))

+ /*REALIZAR DIAGNOSTICO LABORATORIO*/

(IInference.LimpiarDB ? ()
→
IInference.IntroducirUsuarioLab ? (UsuarioLab)
→

```

IInference.IntroducirUsuarioLab ! (UsuarioLab)
→
IInference.IntroducirIDPaciente ? (IDPaciente )
→
IInference.IntroducirIDPaciente ! (IDPaciente)
→
IInference.TomarSugLab ? ( SugLab)
→
IInference.AdquirirDiagLab ? (PosEnfermedades, EnfermedadLab, CertLab)
→
IInference.AdquirirDiagLab ! (PosEnfermedades, EnfermedadLab, CertLab) )

```

USERS =

```

IDiagnostical.IntroducirLoginPassword ! (Login, Password )
→
IDiagnostical.IntroducirLoginPassword ? (Login, Password)
→
IDiagnostical.IntroducirSeleccion ! (Seleccion )
→
IDiagnostical.IntroducirSeleccion ? (Seleccion)

```

→

*/*INTRODUCIR PACIENTE*/*

```

(
( IDiagnostical.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac)
→
IDiagnostical.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac)

```

+

*/*VISUALIZAR PACIENTE*/*

```

(IDiagnostical.IntroducirIDPaciente ! (IDPaciente)
→
IDiagnostical.IntroducirIDPaciente ? (IDPaciente )
→
IDiagnostical.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
)

```

+

/VISUALIZAR RESULTADOS/*

```

(IDiagnostical.IntroducirIDPaciente ! (IDPaciente )
→
IDiagnostical.IntroducirIDPaciente ? (IDPaciente)
→
IDiagnostical.TomarResultados ! (EnfermedadClin, CertClin, EnfermedadLab. CertLab,
Terapia, UsuarioClin, UsuarioLab) )

```

+

*/*REALIZAR DIAGNOSTICO CLINICO*/*

```

(IDiagnostical.IntroducirUsuarioClin ! (UsuarioClin )
→
IDiagnostical.IntroducirUsuarioClin ? (UsuarioClin)
→
IDiagnostical.IntroducirIDPaciente ! (IDPaciente )
→
IDiagnostical.IntroducirIDPaciente ? (IDPaciente)
→
(IDiagnostical.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas)

```

^
 IDiagnostical.ObtenerSsggIra ! (Tos, DificultadRespiratoria))
 →
 (IDIagnostical.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 IDiagnostical.ObtenerSsggIra ? (Tos, DificultadRespiratoria))
 →
 (
 (IDiagnostical.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
 DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
 DolorMasticacionConBorramientoAnguloMandibular)
 ^
 IDiagnostical.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
 ArdorGargantaIntenso))
 +
 (IDIagnostical.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38,
 Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
 OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
 ArdorGargantaIntenso)]
 ^
 IDiagnostical.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
 DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
 ^
 IDiagnostical.ObtenerSsgfCrup ! (RinoreaTransparente,
 DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))
)
 →
 (
 (IDiagnostical.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
 DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
 DolorMasticacionConBorramientoAnguloMandibular)
 ^
 IDiagnostical.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
 ArdorGargantaIntenso))
 +
 (IDIagnostical.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
 Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
 OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
 ArdorGargantaIntenso)]
 ^
 IDiagnostical.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
 DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
 ^
 IDiagnostical.ObtenerSsgfCrup ? (RinoreaTransparente,
 DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))
)
 →
 IDiagnostical.TomarSugerencia ! (Sugerencia))

+
 /*REALIZAR DIAGNOSTICO LABORATORIO*/

(IDiagnostical.IntroducirUsuarioLab ! (UsuarioLab)
 →
 IDiagnostical.IntroducirUsuarioLab ? (UsuarioLab)
 →
 IDiagnostical.IntroducirIDPaciente ! (IDPaciente)
 →
 IDiagnostical.IntroducirIDPaciente ? (IDPaciente)
 →
 IDiagnostical.TomarSugLab ! (SugLab)

```

→
(
( IDiagnostical.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
IDiagnostical.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
(IDiagnostical.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
^
IDiagnostical.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
IDiagnostical.ObtenerEstsLabCrup ! (BiometriaHematicaEusino filia) )
)
→
(
( IDiagnostical.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
IDiagnostical.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
(IDiagnostical.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
^
IDiagnostical.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
IDiagnostical.ObtenerEstsLabCrup ? (BiometriaHematicaEusino filia) )
)
)

```

KNOWLEDGE =

/VISUALIZAR RESULTADOS*/

```

( IDomain.LimpiarDB ! ( )
→
IDomain.AdquirirTerapia ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia)
→
IDomain.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia) )

```

+

/*REALIZAR DIAGNOSTICO CLINICO*/

```

(IDomain.LimpiarDB ! ( )
→
IDomain.AdquirirPosEnfermedad ! (PosEnfermedades )
→
( (IDomain.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas )
^
IDomain.ObtenerSsggIra ? (Tos, DificultadRespiratoria) )
→
(IDomain.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas)
^
IDomain.ObtenerSsggIra ! (Tos, DificultadRespiratoria) )
→
IDomain.AdquirirPosEnfermedad ? (PosEnfermedades)
→
IDomain.AdquirirSugLab ! (SugLab )
→
IDomain.AdquirirSugLab ? (SugLab)

```

```

→
IDomain.AdquirirDiagClin ! (PosEnfermedades, EnfermedadClin, CertClin)
→
(
( IDomain.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IDomain.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(IDomain.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
IDomain.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
^
IDomain.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(
( IDomain.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IDomain.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(IDomain.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38, Fiebre2A3Dias,
TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja,
Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso) ]
^
IDomain.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
^
IDomain.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
IDomain.AdquirirDiagClin ? (PosEnfermedades, EnfermedadClin, CertClin)
→
IDomain.AdquirirSugerencia ! (Sugerencia )
→
IDomain.AdquirirSugerencia ? (Sugerencia) )

```

+ /*REALIZAR DIAGNOSTICO LABORATORIO*/

```

(IDomain.LimpiarDB ! ( )
→
IDomain.AdquirirDiagLab ! (PosEnfermedades, EnfermedadLab, CertLab)
→
(
( IDomain.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
IDomain.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
(IDomain.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)

```

```

^
IDomain.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
^
IDomain.ObtenerEstsLabCrup ? (BiometriaHematicaEusinoFilia )
)
→
(
( IDomain.ObtenerEstsLabPapeas ! (BiometriaHematicaPositiva)
^
IDomain.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
(IDomain.ObtenerEstsLabBronquiolitosis ! (BiometriaHematicaLeucositos)
^
IDomain.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
^
IDomain.ObtenerEstsLabCrup ! (BiometriaHematicaEusinoFilia )
)
→
IDomain.AdquirirDiagLab ? (PosEnfermedades, EnfermedadLab, CertLab) )

```

Protocols

CDIAG = begin.COORDD

COORDD =

```

(MOTOR.IntroducirLoginPassword ? (Login, Password)
→
USERS.IntroducirLoginPassword ! (Login, Password)
→
USERS.IntroducirLoginPassword ? (Login, Password)
→
MOTOR.IntroducirLoginPassword ! (Login, Password) ).P0

```

P0 =

```

( MOTOR.IntroducirSeleccion ? (Seleccion)
→
USERS.IntroducirSeleccion ! (Seleccion)
→
USERS.IntroducirSeleccion ? (Seleccion)
→
MOTOR.IntroducirSeleccion ! (Seleccion) ).OPTIONAL

```

OPTIONAL =

```

RegistrarPaciente
+
VisualizarPaciente
+
VisualizarResultados
+
RealizarDiagnosticoClinico
+
RealizarDiagnosticoLaboratorio
+
Finalizar;

```

RegistrarPaciente=

```

(MOTOR.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac)

```

→
 USERS.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
 →
 USERS.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
 →
 MOTOR.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)).P0

VisualizarPaciente=

(MOTOR.IntroducirIDPaciente ? (IDPaciente)
 →
 USERS.IntroducirIDPaciente ! (IDPaciente)
 →
 USERS.IntroducirIDPaciente ? (IDPaciente)
 →
 MOTOR.IntroducirIDPaciente ! (IDPaciente)
 →
 (MOTOR.TomarPaciente ? (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
 →
 USERS. TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
). P0

VisualizarResultados=

(MOTOR.LimpiarDB ? ()
 →
 KNOWLEDGE.LimpiarDB ! ()
 →
 MOTOR.IntroducirIDPaciente ? (IDPaciente)
 →
 USERS.IntroducirIDPaciente ! (IDPaciente)
 →
 USERS.IntroducirIDPaciente ? (IDPaciente)
 →
 MOTOR.IntroducirIDPaciente ! (IDPaciente)
 →
 MOTOR.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)
 →
 KNOWLEDGE.AdquirirTerapia ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)
 →
 KNOWLEDGE.AdquirirTerapia ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)
 →
 MOTOR.AdquirirTerapia ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia)
 →
 MOTOR.TomarResultados ? (EnfermedadClin, CertClin, EnfermedadLab. CertLab, Terapia, UsuarioClin, UsuarioLab)
 →
 USERS.TomarResultados ! (EnfermedadClin, CertClin, EnfermedadLab. CertLab, Terapia, UsuarioClin, UsuarioLab)). P0

RealizarDiagnosticoClinico=

(MOTOR.LimpiarDB ? ()

→
 KNOWLEDGE.LimpiarDB ! ()
 →
 MOTOR.IntroducirUsuarioClin ? (UsuarioClin)
 →
 USERS.IntroducirUsuarioClin ! (UsuarioClin)
 →
 USERS.IntroducirUsuarioClin ? (UsuarioClin)
 →
 MOTOR.IntroducirUsuarioClin ! (UsuarioClin)
 →
 MOTOR.IntroducirIDPaciente ? (IDPaciente)
 →
 USERS.IntroducirIDPaciente ! (IDPaciente)
 →
 USERS.IntroducirIDPaciente ? (IDPaciente)
 →
 MOTOR.IntroducirIDPaciente ! (IDPaciente)
 →
 MOTOR.AdquirirPosEnfermedad ? (PosEnfermedades)
 →
 KNOWLEDGE.AdquirirPosEnfermedad ! (PosEnfermedades)
 →
 (KNOWLEDGE.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 KNOWLEDGE.ObtenerSsggIra ? (Tos, DificultadRespiratoria))
 →
 (USERS.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 USERS.ObtenerSsggIra ! (Tos, DificultadRespiratoria))
 →
 (USERS.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 USERS.ObtenerSsggIra ? (Tos, DificultadRespiratoria))
 →
 (KNOWLEDGE.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 KNOWLEDGE.ObtenerSsggIra ! (Tos, DificultadRespiratoria))
 →
 (KNOWLEDGE.AdquirirPosEnfermedad ? (PosEnfermedades)
 →
 MOTOR.AdquirirPosEnfermedad ! (PosEnfermedades)
 →
 MOTOR.AdquirirSugLab ? (SugLab)
 →
 KNOWLEDGE.AdquirirSugLab ! (SugLab)
 →
 KNOWLEDGE.AdquirirSugLab ? (SugLab)
 →
 MOTOR.AdquirirSugLab ! (SugLab)
 →
 MOTOR.AdquirirDiagClin ? (PosEnfermedades)
 →
 KNOWLEDGE.AdquirirDiagClin ! (PosEnfermedades)
 →
 (

```

( KNOWLEDGE.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
KNOWLEDGE.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37,
Odinofagia2A3Dias, ArdorGargantaIntenso) )
+
(KNOWLEDGE.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
KNOWLEDGE.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema,
TosFrecuente, DificultadRespiratoriaGravePolipneaTiraje,
HipoventilacionCrepitantePulmonar)
^
KNOWLEDGE.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(
( USERS.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
USERS.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(USERS.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38, Fiebre2A3Dias,
TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja,
Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso) ]
^
USERS.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
^
USERS.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(
( USERS.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
USERS.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(USERS.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38, Fiebre2A3Dias,
TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja,
Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso) ]
^
USERS.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
^
USERS.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(

```

```

( KNOWLEDGE.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
KNOWLEDGE.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37,
Odinofagia2A3Dias, ArdorGargantaIntenso) )
+
(KNOWLEDGE.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
KNOWLEDGE.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema,
TosFrecuente, DificultadRespiratoriaGravePolipneaTiraje,
HipoventilacionCrepitantePulmonar)
^
KNOWLEDGE.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
KNOWLEDGE.AdquirirDiagClin ? (PosEnfermedades, EnfermedadClin, CertClin)
→
MOTOR.AdquirirDiagClin ! (PosEnfermedades, EnfermedadClin, CertClin)
→
MOTOR.AdquirirSugerencia ? (Sugerencia )
→
KNOWLEDGE.AdquirirSugerencia ! (Sugerencia )
→
KNOWLEDGE.AdquirirSugerencia ? (Sugerencia)
→
MOTOR.AdquirirSugerencia ! (Sugerencia)
→
MOTOR.TomarSugerencia ? (Sugerencia)
→
USERS.TomarSugerencia ! (Sugerencia) ). P0

```

```

RealizarDiagnosticoLaboratorio=
( MOTOR.LimpiarDB ? ( )
→
KNOWLEDGE.LimpiarDB ! ( )
→
MOTOR.IntroducirUsuarioLab ? (UsuarioLab )
→
USERS.IntroducirUsuarioLab ! (UsuarioLab )
→
USERS.IntroducirUsuarioLab ? (UsuarioLab)
→
MOTOR.IntroducirUsuarioLab ! (UsuarioLab)
→
MOTOR.IntroducirIDPaciente ? (IDPaciente)
→
USERS. IntroducirIDPaciente! (IDPaciente)
→
USERS.IntroducirIDPaciente ? (IDPaciente)
→
MOTOR.IntroducirIDPaciente ! (IDPaciente)
→
MOTOR.TomarSugLab ? (SugLab)
→

```

```

USERS.TomarSugLab ! (SugLab)
→
MOTOR.AdquirirDiagLab ? (PosEnfermedades)
→
KNOWLEDGE.AdquirirDiagLab ! (PosEnfermedades)
→
(
( KNOWLEDGE.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
KNOWLEDGE.ObtenerEstsLabParotidisBacteriana ?
(BiometriaHematicaLeucositosis) )
+
(KNOWLEDGE.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
^
KNOWLEDGE.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
KNOWLEDGE.ObtenerEstsLabCrup ? (BiometriaHematicaEusino filia) )
)
→
(
( USERS.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
USERS.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
(USERS.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
^
USERS.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
USERS.ObtenerEstsLabCrup ! (BiometriaHematicaEusino filia) )
)
→
(
( USERS.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
USERS.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
(USERS.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
^
USERS.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
USERS.ObtenerEstsLabCrup ? (BiometriaHematicaEusino filia) )
)
→
(
( KNOWLEDGE.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
KNOWLEDGE.ObtenerEstsLabParotidisBacteriana !
(BiometriaHematicaLeucositosis) )
+
(KNOWLEDGE.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
^
KNOWLEDGE.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
KNOWLEDGE.ObtenerEstsLabCrup ! (BiometriaHematicaEusino filia) )
)
→
KNOWLEDGE.AdquirirDiagLab ? (PosEnfermedades, EnfermedadLab, CertLab)

```

→
MOTOR.AdquirirDiagLab ! (PosEnfermedades, EnfermedadLab, CertLab)).P0

Finalizar=
end;

End_Coordination Aspect CDiag

Aspecto de distribución

El aspecto de distribución del conector DIAGNÓSTICO va a ser estático, por lo que no va a proveer servicios de movilidad; sin embargo si deberá tener los servicios *begin* y *end* para establecer el inicio y fin de la ejecución de dicho aspecto, mismos que son utilizados internamente y no para la comunicación con otros elementos arquitectónicos. Por ello, este aspecto no utiliza ninguna interfaz ni contiene *played_roles*.

Debido a que este aspecto requiere de un atributo para la ubicación de la máquina desde la cual se manipula al sistema, el servicio *begin* tiene la siguiente sintaxis:

begin (**input** Location:loc)

Este aspecto se encuentra especificado en el aspecto de distribución del componente MOTOR DE INFERENCIA, dado que ambos elementos arquitectónicos contienen dicho aspecto.

3.3.1.8 Especificación del Conector Diagnóstico

Tras haberse especificado los aspectos necesarios para el conector DIAGNÓSTICO, a continuación se va a definir el *Connector_Type DiagnosticCnct*, el cual será un tipo, de forma tal que el tipo *DiagnosticCnct* será instanciado para crear el conector DIAGNÓSTICO que se ha diseñado en el modelo arquitectónico.

El conector *DiagnosticCnct* especifica el conjunto de los tres puertos necesarios para comunicar a los elementos arquitectónicos que une. Para definir cada puerto, se detallan el nombre (*MotorPortCnct*, *KnowledgePortCnct* y *UsersPortCnct*), el tipo que corresponde a la interfaz declarada (*IIInference*, *IDomain* e *IDiagnostical*) y el *played_role* (*Motor*, *Knowledege* y *Users*) que especifica el comportamiento del conjunto de servicios

que forman la interfaz que tipa al puerto. Los `played_roles` se encuentran definidos en el aspecto de coordinación (*CDiag*) que importa el conector *DiagnosticCnct*.

```
MotorPortCnct: Inference,
    Played_Role CDiag.Motor;
KnowledgePortCnct: IDomain,
    Played_Role CDiag.Knowledge;
UsersPortCnct: IDiagnostical,
    Played_Role CDiag.Users;
```

El conector tipo *DiagnosticCnct* importa dos aspectos, el de coordinación (*CDiag*) para sincronizar (i.e. coordinar y redireccionar) los servicios entre los componentes que une, y el de distribución (*DLocation*) con el objetivo de indicar su ubicación dentro del sistema.

```
Coordination Aspect import CDiag;
Distribution Aspect import DLocation;
```

En la sección *Initialize* se especifica el inicio de los aspectos (*CDiag* y *DLocation*) que lo forman. Para crear la instancia es necesario dar un valor inicial al atributo requerido `location` de su aspecto de distribución. El atributo `location` tiene la propiedad de *not null*:

```
new (Location: loc)
{
    CDiag.begin();
    DLocation.begin (Location);
}
```

En la sección *Destruction* se especifica el término de los aspectos (*CDiag* y *DLocation*) que lo forman:

```
destroy ()
{
    CDiag.end();
    DLocation.end( );
}
```

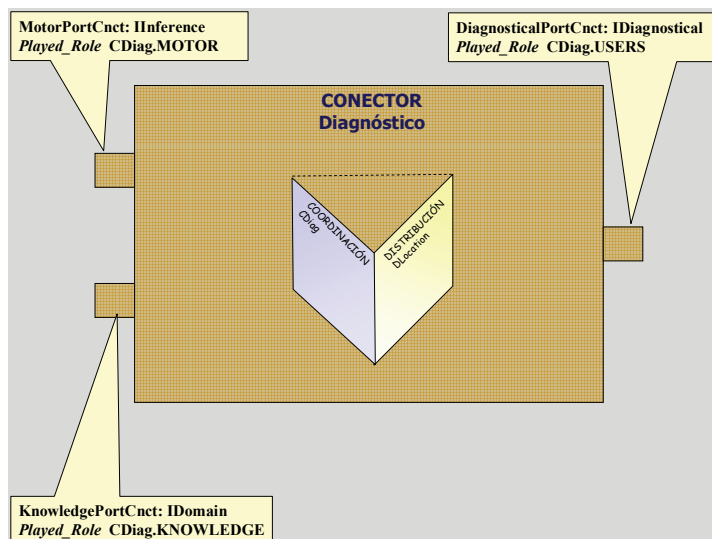


Fig. 39 Conector DIAGNÓSTICO

Con el análisis anteriormente realizado, la especificación del conector Diagnóstico presenta la siguiente sintaxis:

```
Connector_Type DiagnosticCnct
  Ports
    MotorPortCnct: IIference,
      Played_Role CDiag.Motor;
    KnowledgePortCnct: IDomain,
      Played_Role CDiag.Knowledge;
    UsersPortCnct: IDiagnostical,
      Played_Role CDiag.Users;
  End_Ports;

  Coordination Aspect import CDiag;
  Distribution Aspect import DLocation;

  Initialize
    new (Location: loc)
    {
      CDiag.begin();
      DLocation.begin (Location);
    }
  End_Initialize;

  Destruction
    destroy ()
    {
      CDiag.end();
      DLocation.end( );
    }
  End_Destruction;

End_Connector_Type DiagnosticCnct;
```

3.3.1.9 Instanciación del Conector Diagnóstico

En la configuración, el servicio *new* se invoca y da valor en la creación de la instancia, especificándose de la siguiente manera:

```
new DiagnosticCnct (localhost);
```

3.3.1.10 Especificación del Sistema Diagnosticador

Tras haberse especificado los elementos arquitectónicos necesarios para especificar el componente complejo o sistema denominado SISTEMA DIAGNOSTICADOR, a continuación se va a definir el *System_Type* *DiagnosticalSystem*, el cual será un tipo, de forma tal que será instanciado para crear el SISTEMA DIAGNOSTICADOR que se ha diseñado en el modelo arquitectónico.

DiagnosticalSystem no contiene aspectos y especifica sólo un puerto necesario para comunicarse con el otro elemento arquitectónico. Para definir el puerto, se detalla el nombre (*DiagnosticalPort*) y el tipo que corresponde a la interfaz declarada (*IDiagnostical*). Este puerto no tiene un *played_role* asociado, ya que únicamente se comporta como un repetidor del comportamiento del puerto al que está conectado mediante un *binding*:

```
DiagnosticalPort: IDiagnostical,
```

La definición de variables permite especificar los componentes y el conector que forman parte del sistema. Cada componente importado se asocia con una variable, cuyo tipo es el del componente a importar. *DiagnosticalSystem* está formado por tres variables, anteriormente definidas. En la sección *Variables* se especifican en la forma:

```
VarMotordeInferencia: InferenceMotor;  
VarBasedeConocimientos: KnowledgeBase;  
VarConectorDiagnostico: DiagnosticCnct
```

La construcción y destrucción de dichos componentes se desencadena en el momento de la creación o destrucción del sistema. Estas operaciones se especifican en las secciones *Initalize* y *Destruction* respectivamente, y permiten indicar los parámetros que necesita cada componente para su creación, así como definir la secuencia de destrucción de los componentes:

```
new (Location:loc)  
  
VarMotordeInferencia = new InferenceMotor (Location);  
VarBasedeConocimientos = new KnowledgeBase (Location);  
VarConectorDiagnostico = new DiagnosticCnct (Location);  
}  
  
destroy ()  
{  
VarMotordeInferencia.destroy ();  
VarBasedeConocimientos.destroy ();  
VarConectorDiagnostico.destroy ();  
}
```

Dentro del sistema es necesario declarar los elementos arquitectónicos que lo forman y los *attachments* y *bindings* que los unen. La definición del sistema especifica las tres relaciones de conexión (*attachments*) y la de composición (*binding*) entre los elementos arquitectónicos que contiene. Para establecerse la comunicación interna entre los tres elementos arquitectónicos que conforman este sistema, se diseñaron dos *attachments*. .

Los *attachments* establecen la conexión entre los puertos *MotorPort* y *KnowledgePort* de *VarMotordeInferencia* y *VarBasedeConocimientos* respectivamente, y el puerto *MotorPortCnct* del *VarConectorDiagnostico*:

```
VarMotordeInferencia.MotorPort ↔ VarConectorDiagnostico.MotorPortCnct;
VarBasedeConocimientos.KnowledgePort ↔ VarConectorDiagnostico.KnowledgePortCnct;
```

La comunicación de este sistema (componente complejo) con el exterior se realiza mediante un *binding* que va del *VarConectorDiagnostico* (a través de su puerto *UsersPortCnct*) al *DiagnosticalSystem* (mediante su puerto *DiagnosticalPort*).

```
VarConectorDiagnostico.UsersPortCnct ↔ DiagnosticalSystem.DiagnosticalPort;
```

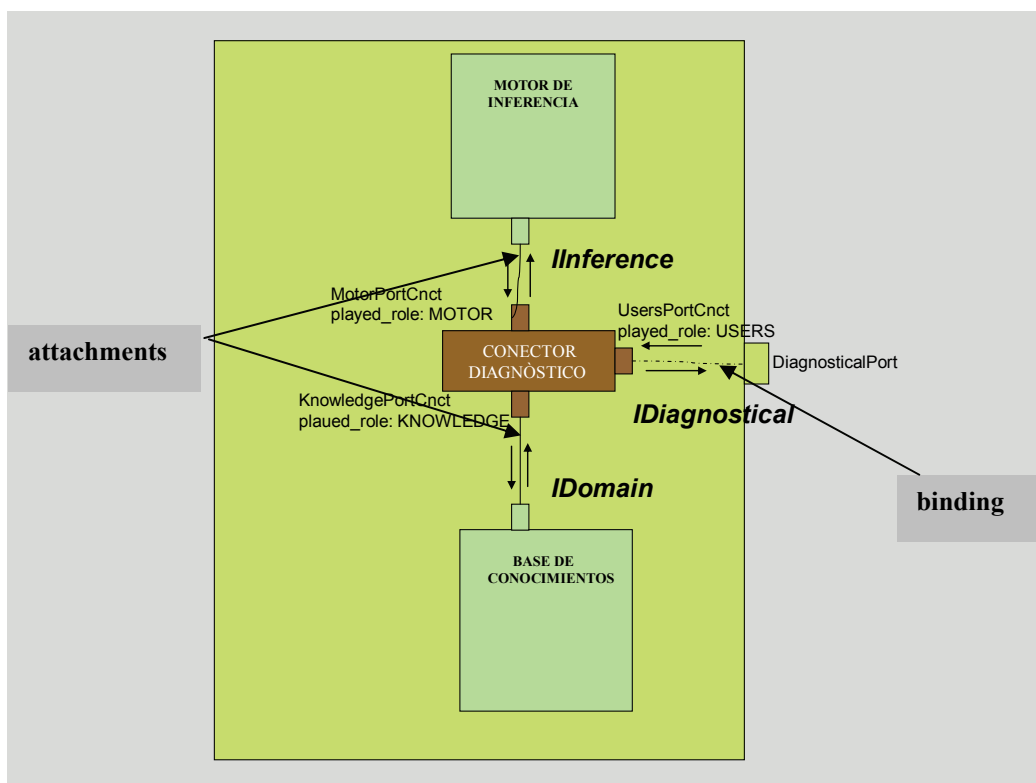


Fig. 40 Sistema Diagnosticador

Con base en el análisis realizado, la especificación del elemento arquitectónico SISTEMA DIAGNOSTICADOR tiene la siguiente sintaxis:

System_Type DiagnosticalSystem

Ports

UsersDiagPort: IDiagnostical,

End_Ports;

Variables

VarMotordeInferencia: InferenceMotor;

VarBasedeConocimientos: KnowledgeBase;

VarConectorDiagnostico: DiagnosticCnct

End_Variables;

Attachments

```
VarMotordeInferencia.MotorPort ↔ VarConectorDiagnostico.MotorPortCnct;  
VarBasedeConocimientos.KnowledgePort ↔ VarConectorDiagnostico.KnowledgePortCnct;
```

End_Attachments;**Bindings**

```
VarConectorDiagnostico.UsersPortCnct ↔ DiagnosticalSystem.DiagnosticalPort;  
End_Bindings;
```

Initialize

```
    new (Location:loc )  
{  
  VarMotordeInferencia = new InferenceMotor (Location);  
  VarBasedeConocimientos = new KnowledgeBase (Location);  
  VarConectorDiagnostico = new DiagnosticCnct (Location);  
}
```

End_Initialize;**Destruction**

```
    destroy ()  
{  
  VarMotordeInferencia.destroy ();  
  VarBasedeConocimientos.destroy ();  
  VarConectorDiagnostico.destroy ();  
}
```

End_Destruction;**End_System_Type** DiagnosticalSystem;

3.3.1.11 Instanciación del Sistema Diagnóstico

Ahora se va a abordar la instanciación de las partes que componen el sistema. La creación de instancias se realiza mediante el servicio `new`, al cual se le pasan como parámetros el conjunto de atributos requeridos para crear la instancia, los cuales tienen la propiedad de `not null` en su definición dentro de cada uno de los aspectos del tipo. En la configuración, el servicio `new` se invoca y da valor en la creación de la instancia del sistema, como se muestra a continuación:

```
new DiagnosticalSystem (localhost);
```

3.3.1.12 Especificación de los Aspectos del Componente Usuario Clínico

Aspecto funcional

El USUARIO CLINICO inicia su comportamiento al recibir el servicio `IntroducirLoginPassword`, con el cual se le solicita al usuario teclear su nombre de usuario

y contraseña. A continuación, este componente recibe el servicio *IntroducirSeleccion* para que el usuario seleccione la opción del sistema a realizar.

IntroducirLoginPassword (**output** Login:string, **output** Password:string)

IntroducirSeleccion (**output** Seleccion:int)

- Si se ha elegido la opción de “Registrar Paciente”, al USUARIO CLÍNICO se se solicita mediante el servicio *IntroducirPaciente* los datos del paciente que desea registrar.

IntroducirPaciente (**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string, **output** Dirección:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date).

- En la selección “Visualizar Paciente” se solicita al USUARIO CLINICO con el servicio *IntroducirIDPaciente*, el identificador del paciente del cual se desean consultar sus datos con el que se registró..

IntroducirIDPaciente (**output** IDPaciente:identifier)

Finalmente el USUARIO CLINICO recibe el servicio *TomarPaciente* que le sirve para adquirir la información del paciente que se mostrará al usuario (ver aspecto de presentación).

TomarPaciente (**input** DNI:string, **input** Dirección:string, **input** Tel:string, **input** Email:string, **input** Sexo;char, **input** FechaNac:date).

- Si se desea “Visualizar los Resultados” del diagnóstico realizado, al USUARIO CLINICO le es requerido con el servicio *IntroducirIDPaciente* el identificador del paciente de quien se mostrarán dicha información.

IntroducirIDPaciente (**output** IDPaciente:identifier)

- Para el caso “Realizar Diagnóstico Clínico”, primeramente se solicita el nombre del médico que atendió al paciente mediante el servicio *IntroducirUsuarioClin*.

IntroducirUsuarioClin (**output** UsuarioClin:string)

Acto seguido el servicio *IntroducirIDPaciente* es recibido para obtener el identificador del paciente para identificar al paciente que se le realizará el diagnóstico clínico.

IntroducirIDPaciente (**output** IDPaciente:identifier)

A continuación el servicio *ObtenerSSGG* es recibido por el USUARIO CLINICO para obtener los signos y síntomas de grano grueso del paciente a diagnosticar.

Inmediatamente después a este componente le llega el servicio *ObtenerSSGF* con el que son requeridos los signos y síntomas de grano fino que se encontraron en el paciente.

obtenerSsgfPaperas (**output** FiebreContinua:bool,
output FiebreMayor38:bool, **output** DolorCrecimientoParotidas:bool,
output DolorMasticacionEspontaneoAgudo:bool,
output DolorMasticacionConBorramientoAnguloMandibular:bool)

obtenerSsgfParotiditisBacteriana (**output** FiebreMayor37:bool,
output Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsgfBronquiolitis (**output** FiebreContinua:bool,
output FiebreMayor38:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool,
output TosPorAccesos:bool, **output** DificultadRespiratoriaGrave:bool,
output OdinofagiaGargantaRoja:bool, **output** Odinofagia2A3Dias:bool,
output ApneaPeriodosLargos:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsgfNeumonia (**output** FiebreMayor39:bool,
output TosConFlema:bool, **output** TosFrecuente:bool,
output DificultadRespiratoriaGravePolipneaTiraje:bool,
output HipoventilacionCrepitantePulmonar:bool)

obtenerSsgfCrup (**output** RinoreaTransparente:bool
output DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool))

Por último, al USUARIO CLINICO le es enviado el servicio *TomarSugerencia* para que este componente pueda mostrar al usuario la sugerencia de si es necesario realizar o no los estudios de laboratorio.

TomarSugerencia (**input** Sugerencia:string)

Los servicios antes expuestos se publican en el puerto denominado *UserClinPort* del componente USUARIO CLINICO, que tiene asociado como signatura la interfaz *IClinical*, y que da semántica a su aspecto funcional.

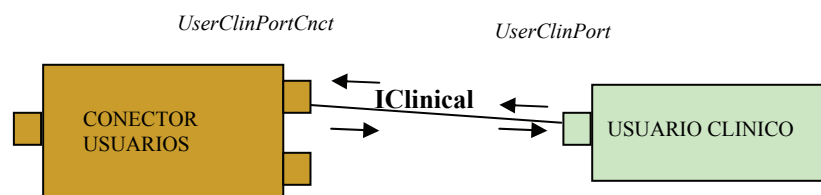


Fig. 41 Esquema del funcionamiento de la Interfaz IClinical

La especificación de *IClinical* tiene la siguiente sintaxis:

Interfaz IClinical

introducirLoginPassword (**output** Login:string, **output** Password:string)
introducirSeleccion (**output** Seleccion:int)

introducirPaciente (**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string, **output** Direccion:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date)

introducirIDPaciente (**output** IDPaciente:identifier)

introducirUsuarioClin (output UsuarioClin:string)

obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool, output DolorCrecimientoParotidas:bool)

obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)

obtenerSsggPaperas (output FiebreContinua:bool, **output** FiebreMayor38:bool, **output** DolorCrecimientoParotidas:bool, **output** DolorMasticacionEspontaneoAgudo:bool, **output** DolorMasticacionConBorramientoAnguloMandibular:bool)

obtenerSsggParotiditisBacteriana (output FiebreMayor37:bool, **output** Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsggBronquiolitis (output FiebreContinua:bool, **output** FiebreMayor38:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool, **output** TosPorAccesos:bool, **output** DificultadRespiratoriaGrave:bool, **output** OdinofagiaGargantaRoja:bool, **output** Odinofagia2A3Dias:bool, **output** ApneaPeriodosLargos:bool, **output** ArdorGargantaIntenso:bool)

obtenerSsgfNeumonia (output FiebreMayor39:bool, **output** TosConFlema:bool, **output** TosFrecuente:bool, **output** DificultadRespiratoriaGravePolipneaTiraje:bool, **output** HipoventilacionCrepitantePulmonar:bool)

obtenerSsgfCrup (output RinoreaTransparente:bool **output** DificultadRespiratoriaGarveAModerada:bool, **output** EstrindorLaringeo:bool)).

tomarPaciente (input DNI:string, input Nombre:string, input Direccion:string, input Tel:string, **input** Email:string, **input** Sexo:char, **input** FechaNac:date)

tomarResultados (**input** EnfermedadClin:string, **input** CertClin:real, **input** EnfermedadLab:string, **input** CertLab:real, **input** Terapia, **input** UsuarioClin:string, **input** UsuarioLab:string)

End_Interfaz IClinical

Con base en el análisis anterior, el aspecto funcional del componente USUARIO CLÍNICO se especifica a continuación:

Functional Aspect FClinical **using** IClinical

Services

begin

in introducirLoginPassword (**output** Login:string, **output** Password:string)

in introducirSeleccion (**output** Seleccion:int)

in introducirPaciente (**output** IDPaciente:identifier, **output** DNI:string, **output** Nombre:string, **output** Direccion:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date)

in introducirIDPaciente (**output** IDPaciente:identifier)

```

in introducirUsuarioClin (output UsuarioClin:string)

in obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool,
output DolorCrecimientoParotidas:bool )

in obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)

in obtenerSsgfPaperas (output FiebreContinua:bool,
output FiebreMayor38:bool, output DolorCrecimientoParotidas:bool,
output DolorMasticacionEspontaneoAgudo:bool,
output DolorMasticacionConBorramientoAnguloMandibular:bool )

in obtenerSsgfParotiditisBacteriana (output FiebreMayor37:bool,
output Odinofagia2A3Dias:bool, output ArdorGargantaIntenso:bool )

in obtenerSsgfBronquiolitis (output FiebreContinua:bool,
output FiebreMayor38:bool, output Fiebre2A3Dias:bool, output TosSeca:bool,
output TosPorAccesos:bool, output DificultadRespiratoriaGrave:bool,
output OdinofagiaGargantaRoja:bool, output Odinofagia2A3Dias:bool,
output ApneaPeriodosLargos:bool, output ArdorGargantaIntenso:bool)

in obtenerSsgfNeumonia (output FiebreMayor39:bool,
output TosConFlema:bool, output TosFrecuente:bool,
output DificultadRespiratoriaGravePolipneaTiraje:bool,
output HipoventilacionCrepitantePulmonar:bool)

in obtenerSsgfCrup (output RinoreaTransparente:bool
output DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool) ).

in tomarPaciente (input DNI:string, input Nombre:string,
input Direccion:string, input Tel:string, input Email:string,
input Sexo:char, input FechaNac:date)

in tomarResultados (input EnfermedadClin:string, input CertClin:real,
input EnfermedadLab:string, input CertLab:real, input Terapia,
input UsuarioClin:string, input UsuarioLab:string)

in tomarSugerencia (input Sugerencia:string)

end;

```

Played_Roles

```

USERCLIN=
  IClinical.IntroducirLoginPassword ? (Login, Password)
  →
  IClinical.IntroducirLoginPassword ! (Login, Password)
  →
  IClinical.IntroducirSeleccion ? (Seleccion )
  →
  IClinical.IntroducirSeleccion ! (Seleccion)
  →
(
  /*REGISTRAR PACIENTE*/
  ( IClinical.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo,
  FechaNac)
  →
  IClinical.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo,
  FechaNac) )
)

```

```

+
/*VISUALIZAR PACIENTE*/
(IClinical.IntroducirIDPaciente ? (IDPaciente)
→
IClinical.IntroducirIDPaciente ! (IDPaciente)
→
IClinical.TomarPaciente? (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac) ))

+
/*VISUALIZAR RESULTADOS*/
(IClinical.IntroducirIDPaciente ? (IDPaciente)
→
IClinical.IntroducirIDPaciente ! (IDPaciente)
→
IClinical.TomarResultados ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
Terapia, UsuarioClin, UsuarioLab) )

+
/*REALIZAR DIAGNOSTICO CLINICO*/
(IClinical.IntroducirUsuarioClin ? (UsuarioClin )
→
IClinical.IntroducirUsuarioClin ! (UsuarioClin) )
→
IClinical.IntroducirIDPaciente ? (IDPaciente )
→
IClinical.IntroducirIDPaciente ! (IDPaciente)
→
( IClinical.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas )
^
IClinical.ObtenerSsggIra ? (Tos, DificultadRespiratoria) )
→
( IClinical.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas )
^
IClinical.ObtenerSsggIra ! (Tos, DificultadRespiratoria) )
→
(
( IClinical.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IClinical.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(IClinical.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
IClinical.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
^
IClinical.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(
( IClinical.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IClinical.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
)

```

```

+
(IClinical.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38, Fiebre2A3Dias,
TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja,
Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso) ]
^
IClinical.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
^
IClinical.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
IClinical.TomarSugerencia ? (Sugerencia ) )
)

```

Protocols

```
FCLINICAL = begin.FUNCTIONALUC;
```

```
FUNCTIONALUC =
  ( USERCLIN.IntroducirLoginPassword ? (Login, Password)
    →
    USERCLIN.IntroducirLoginPassword ! (Login, Password) ).P0
```

```
P0 =
  ( USERCLIN.IntroducirSeleccion ? (Seleccion)
    →
    USERCLIN.IntroducirSeleccion ! (Seleccion) ).OPTIONC
```

```
OPTIONC =
  RegistrarPaciente
+
  VisualizarPaciente
+
  VisualizarResultados
+
  RealizarDiagnosticoClinico
+
  Finalizar;
```

```
RegistrarPaciente =
  (USERCLIN.IntroducirPaciente ? (IDPaciente,DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac )
  →
  USERCLIN.IntroducirPaciente ! (IDPaciente,DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac) ).P0
```

```
VisualizarPaciente =
  (USERCLIN.IntroducirIDPaciente ? (IDPaciente)
  →
  USERCLIN.IntroducirIDPaciente ! (IDPaciente)
  →
  USERCLIN.TomarPaciente ? (DNI, Nombre, Direccion, Tel, Email, Sexo,
FechaNac) ).P0
```

```
VisualizarResultados =
  (USERCLIN.IntroducirIDPaciente ? (IDPaciente)
  →
  USERCLIN.IntroducirIDPaciente ! (IDPaciente)
  →
```

USERCLIN.TomarResultados ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)). P0

RealizarDiagnosticoClinico =

(USERCLIN.IntroducirIDPaciente ? (IDPaciente))

→

USERCLIN.IntroducirIDPaciente ! (IDPaciente)

→

USERCLIN.IntroducirUsuarioClin ? (UsuarioClin))

→

USERCLIN.IntroducirUsuarioClin ! (UsuarioClin)

→

(USERCLIN.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion, DolorCrecimientoParotidas))

^

USERCLIN.ObtenerSsggIra ? (Tos, DificultadRespiratoria))

→

(USERCLIN.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion, DolorCrecimientoParotidas))

^

USERCLIN.ObtenerSsggIra ! (Tos, DificultadRespiratoria))

→

(

(USERCLIN.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular)

^

USERCLIN.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias, ArdorGargantaIntenso))

+

(USERCLIN.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38, Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso)]

^

USERCLIN.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)

^

USERCLIN.ObtenerSsgfCrup ? (RinoreaTransparente, DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))

)

→

(

(USERCLIN.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular)

^

USERCLIN.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias, ArdorGargantaIntenso))

+

(USERCLIN.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38, Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso)]

^

USERCLIN.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)

^

USERCLIN.ObtenerSsgfCrup ! (RinoreaTransparente, DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))

)

→
USERCLIN.TomarSugerencia ? (Sugerencia)).P0

Finalizar =
end;

End Functional Aspect FClinical

Aspecto de presentación

En este aspecto, que alberga la parte gráfica para la comunicación del usuario con el sistema, son utilizados unos servicios para mostrar la información, ya sea la requerida por el usuario al visualizar los datos del registro del paciente (*showPatient*) o bien los resultados del diagnóstico realizado (*showResults*), así como la que se le es mostrada por el propio sistema (*showMessage*, *showSSGG*, *showSSGF*), para que con los servicios *onclickLoginPassword*, *onclickSelect*, *onclickRegistrerPatient*, *onclickIDPatient*, *onclickUserClinIDPatient*, *onClickSSGG*, *onClickSSGF* se obtenga del usuario la información requerida para el funcionamiento del sistema (nombre del usuario y contraseña, opción elegida, datos del registro del paciente, identificador del paciente, nombre del médico, signos y síntomas de grano grueso y de grano fino encontrados en el paciente).

onClickRegistrerPatient (**output** IDPaciente:identifier, **output** DNI: string,
output Nombre: string, **output** Direccion: string, **output** Tel: string,
output Email: string, **output** Sexo: string, **output** FechaNac: string);

showPatient (**input** DNI:string, **input** Nombre:string, **input** Direccion:string,
input Tel:string, **input** Email:string, **input** Sexo:string, **input** FechaNac:string);

onclickIDPatient (**output** IDPaciente: identifier);

onclickUserClinIDPatient (**output** UsuarioClin: string **output** IDPaciente: identifier);

onClickResults (**output** IDPaciente: identifier);

showResults (**input** EnfClin:string, **input** CertClin:real, **input** EnfLab:string,
input CertLab:real, **input** Terapia:string, **input** UsuarioClin:string,
input UsuarioLab:string);

showSSGG (**input** Fiebre:bool, **input** Tos:bool, **input** DificultadRespiratoria:bool,
input DolorMasticacion:bool, **input** DolorCrecimientoParotidas:bool)

onClickSSGG (**output** Fiebre:bool, **output** Tos:bool, **output** DificultadRespiratoria:bool,
output DolorMasticacion:bool, **output** DolorCrecimientoParotidas:bool);

showSSGF (**input** FiebreContinua:bool, **input** FiebreMayor37:bool,
input FiebreMayor38:bool, **input** FiebreMayor39:bool, **input** Fiebre2A3Dias:bool,
input TosSeca:bool, **input** TosPorAccesos:bool, **input** TosConFlema:bool,
input TosFrecuente:bool, **input** DificultadRespiratoriaGrave:bool,
input DificultadRespiratoriaPolipnea:bool, **input** DificultadRespiratoriaTiraje:bool,
input HipoventilacionCrepitantePulmonar:bool, **input** OdinofagiaGargantaRoja:bool,

input Odinofagia2A3Dias:bool, **input** ArdorGargantaIntenso:bool,
input ApneaPeriodosLargos:bool, **input** DolorCrecimientoParotidas:bool,
input DolorMasticacionEspontaneoAgudo:bool,
input DolorMasticacionConBorramientoAnguloMandibular:bool,
input RinoreaTransparente:bool, **input** DificultadRespiratoriaGarveAModerada:bool,
input EstrindorLaringeo:bool)

onClickSSGF (**output** FiebreContinua:bool, **output** FiebreMayor37:bool,
output FiebreMayor38:bool, **output** FiebreMayor39:bool, **output** Fiebre2A3Dias:bool,
output TosSeca:bool, **output** TosPorAccesos:bool, **output** TosConFlema:bool,
output TosFrecuente:bool, **output** DificultadRespiratoriaGrave:bool,
output DificultadRespiratoriaPolipnea:bool, **output** DificultadRespiratoriaTiraje:bool,
output HipoventilacionCrepitantePulmonar:bool, **output** OdinofagiaGargantaRoja:bool,
output Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool,
output ApneaPeriodosLargos:bool, **output** DolorCrecimientoParotidas:bool,
output DolorMasticacionEspontaneoAgudo:bool,
output DolorMasticacionConBorramientoAnguloMandibular:bool,
output RinoreaTransparente:bool, **output** DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool);

showSugerencia (**input** Sugerencia: string);

showWindow1 (**input** Forma1: window);

showWindow2 (**input** Forma2: window);

showWindow3 (**input** Forma3: window);

showWindow4 (**input** Forma4: window);

showWindow5 (**input** Forma5: window);

showWindow6 (**input** Forma6: window);

closeWindow1 (**input** Forma1: window);

closeWindow2 (**input** Forma2: window);

closeWindow3 (**input** Forma3: window);

closeWindow4 (**input** Forma4: window);

closeWindow5 (**input** Forma5: window);

closeWindow6 (**input** Forma6: window);

onclickSelect (**output** Seleccion: int);

onclickLoginPassword (**output** Login:string, **output** Contraseña:string);

showMessage (**input** Mensaje: string);

Una muestra de los gráficos anteriormente comentados adquiere la siguiente vista:

CONTRASEÑAS		-	⏏	x
Login	<input type="text"/>			
Password	<input type="text"/>			
		ACEPTAR		

MENU		-	⏏	x
1.- Introducir datos del paciente 2.- Visualizar datos del paciente 3.- Diagnóstico Clínico 5.- Resultados 6.- Salir				
Selección	<input type="text"/>			
		ACEPTAR		

INTRODUCIR DATOS PACIENTE		-	⏏	x
ID Paciente	<input type="text"/>			
DNI	<input type="text"/>			
Nombre	<input type="text"/>			
Dirección	<input type="text"/>			
Teléfono	<input type="text"/>			
E-mail	<input type="text"/>			
Sexo	<input type="text"/>			
Fecha Nac.	<input type="text"/>			
		ACEPTAR		

VISUALIZAR DATOS PACIENTE		-	⏏	x
ID Paciente	<input type="text"/>			
DNI	<input type="text"/>			
Nombre	<input type="text"/>			
Dirección	<input type="text"/>			
Teléfono	<input type="text"/>			
E-mail	<input type="text"/>			
Sexo	<input type="text"/>			
Fecha Nac.	<input type="text"/>			
		ACEPTAR		

Fig. 42 Gráficos que puede utilizar el aspecto de presentación del USUARIO CLÍNICO

Una característica de este aspecto es que, a pesar de tener servicios, no utiliza ninguna interfaz ni contiene played_roles, ya que dichos servicios son públicos, i.e. sólo son utilizados internamente y no para la comunicación con otros elementos arquitectónicos.

Con base en el análisis anterior, el aspecto de presentación del componente USUARIO CLÍNICO se especifica a continuación:

Presentation Aspect Pclinical

Attributes

Variable

titleWindow: window;
 label1, label2, label3, label4, label5, label6, label7, label8: label
 button1, button2, button3: button;
 text1, text2, text3, text4, text5: text;

Derivations

forma1:= {titleWindow=CONTRASEÑAS and label1=Log in and label2=Contraseña and button1=ACEPTAR }

forma2:= {titleWindow=MENU and text1=1.-Introducir datos del Paciente and text2=2.- Visualizar datos del paciente and text3=3.-Diagnostico Clinico and text4=5.-Resultados and text5:-6.-Salir and label1=Selecccion and button1=ACEPTAR }

forma3:= {titleWindow=INTRODUCIR DATOS PACIENTE **and** label1=ID Paciente **and** label2=DNI **and** label3= Nombre **and** label4= Direccion **and** label5=Teléfono **and** label6=Email **and** label7:Sexo **and** label8:Fecha Nac **and** button1=ACEPTAR }

forma4:= {titleWindow=VISUALIZAR DATOS DEL PACIENTE **and** label1= ID Paciente **and** label2=DNI **and** label3= Nombre **and** label4= Direccion **and** label5=Teléfono **and** label6=Email **and** label7:Sexo **and** label8:Fecha Nac **and** button1=ACEPTAR }

forma5:= {titleWindow =DIAGNOSTICO CLÍNICO **and** label1=Médico **and** label2= ID Paciente **and** label3= SSGG presente: **and** label4= SSGF presente: **and** label5=Sugerencia **and** button1=SI **and** button2=NO **and** button3=SI **and** button4=NO **and** button5=ACEPTAR}

forma6:= {titleWindow=RESULTADOS **and** label1=Diagnostico Clínico **and** label2=Cert Clinica **and** label3=Diagnostico Laboratorio **and** label4=Cert Lab **and** label5=Terapia **and** label6=Médico **and** label7=Laboratorista **and** button1=ACEPTAR }

Services

begin ();

onClickRegistrerPatient (**output** IDPaciente:identifier, **output** DNI: string, **output** Nombre: string, **output** Direccion: string, **output** Tel: string, **output** Email: string, **output** Sexo: string, **output** FechaNac: string);

showPatient (**input** DNI:string, **input** Nombre:string, **input** Direccion:string, **input** Tel:string, **input** Email:string, **input** Sexo:string, **input** FechaNac:string);

onclickIDPatient (**output** IDPaciente: identifier);

onclickUserClinIDPatient (**output** UsuarioClin: string **output** IDPaciente: identifier);

onClickResults (**output** IDPaciente: identifier);

showResults (**input** EnfClin:string, **input** CertClin:real, **input** EnfLab:string, **input** CertLab:real, **input** Terapia:string, **input** UsuarioClin:string, **input** UsuarioLab:string);

showSSGG (**input** Fiebre:bool, **input** Tos:bool, **input** DificultadRespiratoria:bool, **input** DolorMasticacion:bool, **input** DolorCrecimientoParotidas:bool)

onClickSSGG (**output** Fiebre:bool, **output** Tos:bool, **output** DificultadRespiratoria:bool, **output** DolorMasticacion:bool, **output** DolorCrecimientoParotidas:bool);

showSSGF (**input** FiebreContinua:bool, **input** FiebreMayor37:bool, **input** FiebreMayor38:bool, **input** FiebreMayor39:bool, **input** Fiebre2A3Dias:bool, **input** TosSeca:bool, **input** TosPorAccesos:bool, **input** TosConFlema:bool, **input** TosFrecuente:bool, **input** DificultadRespiratoriaGrave:bool, **input** DificultadRespiratoriaPolipnea:bool, **input** DificultadRespiratoriaTiraje:bool, **input** HipoventilacionCrepitantePulmonar:bool, **input** OdinofagiaGargantaRoja:bool, **input** Odinofagia2A3Dias:bool, **input** ArdorGargantaIntenso:bool, **input** ApneaPeriodosLargos:bool, **input** DolorCrecimientoParotidas:bool, **input** DolorMasticacionEspontaneoAgudo:bool, **input** DolorMasticacionConBorramientoAnguloMandibular:bool, **input** RinoreaTransparente:bool, **input** DificultadRespiratoriaGarveAModerada:bool, **input** EstrindorLaringeo:bool)

onClickSSGF (**output** FiebreContinua:bool, **output** FiebreMayor37:bool, **output** FiebreMayor38:bool, **output** FiebreMayor39:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool, **output** TosPorAccesos:bool, **output** TosConFlema:bool, **output** TosFrecuente:bool, **output** DificultadRespiratoriaGrave:bool, **output** DificultadRespiratoriaPolipnea:bool, **output** DificultadRespiratoriaTiraje:bool,

output HipoventilacionCrepitantePulmonar:bool, **output** OdinofagiaGargantaRoja:bool,
output Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool,
output ApneaPeriodosLargos:bool, **output** DolorCrecimientoParotidas:bool,
output DolorMasticacionEspontaneoAgudo:bool,
output DolorMasticacionConBorramientoAnguloMandibular:bool,
output RinoreaTransparente:bool, **output** DificultadRespiratoriaGarveAModerada:bool,
output EstrindorLaringeo:bool);

showSugerencia (**input** Sugerencia: string);
 showWindow1 (**input** Forma1: window);
 showWindow2 (**input** Forma2: window);
 showWindow3 (**input** Forma3: window);
 showWindow4 (**input** Forma4: window);
 showWindow5 (**input** Forma5: window);
 showWindow6 (**input** Forma6: window);
 closeWindow1 (**input** Forma1: window);
 closeWindow2 (**input** Forma2: window);
 closeWindow3 (**input** Forma3: window);
 closeWindow4 (**input** Forma4: window);
 closeWindow5 (**input** Forma5: window);
 closeWindow6 (**input** Forma6: window);
 onclickSelect (**output** Seleccion: int);
 onclickLoginPassword (**output** Login:string, **output** Contraseña:string);
 showMessage (**input** Mensaje: string);
 end;

Protocols

PCLINICAL = begin.PRESENTATIONC;

PRESENTATIONC =

ShowWindow1 (Forma1).Po
 +
 end

Po =

CloseWindow1 (Forma1).PRESENTATIONC
 +
 OnclickLoginPaswword (Login, Contraseña).P1

P1

ShowMessage (Mensaje).Po
 +
 (CloseWindow1 (Forma1)
 →

```

ShowWindow2 (Forma2) ).P2

P2 =
  OnclickSelect (Seleccion).P3

P3 =
  ShowMessage (Mensaje).P2
  +
  CloseWindow2 (Forma2).SELECTC

SELECTC =
  (ShowWindow3 (Forma3).Pa1
  /*INTRODUCIR PACIENTE*/

Pa1 =
  CloseWindow3 (Forma3).SELECTC
  +
  (OnclickRegistrerPatient (IDPaciente, DNI, Nombre, Direccion, Tel, Email,. Sexo,
  FechaNac)
  →
  CloseWindow3 (Forma3).SELECTC ) )

+
  (ShowWindow4 (Forma4). Pb1
  /*VISUALIZAR PACIENTE*/

Pb1 =
  (CloseWindow4 (Forma4) . SELECTC
  +
  (OnclickIDPatient (IDPaciente)
  →
  ShowPatient (DNI, Nombre, Direccion, Tel, Email,. Sexo, FechaNac)
  →
  CloseWindow4 (Forma4) ).SELECTC ) )

+
  (ShowWindow5 (Forma5). Pd1
  /*REALIZAR DIAGNOSTICO CLINICO*/

Pd1 =
  CloseWindow5 (Forma5). SELECTC
  +
  (OnclickUserClinIDPatient (UsuarioClin, IDPaciente)
  →
  ShowSsgg (Fiebre, DolorMasticacionl, DolorCrecimientoParotidas, Tos,
  DificultadRespiratoria)
  →
  OnclickSsgg (Fiebre, DolorMasticacionl, DolorCrecimientoParotidas, Tos,
  DificultadRespiratoria)
  →
  ShowSsgf (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39,
  Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente,
  DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje,
  HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias,
  ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas,
  DolorMasticacionEspontaneoAgudo,
  DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente,
  DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)
  →
  OnclickSsgF (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39,
  Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente,
  DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje,
  HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias,
  ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas,

```

```

DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente,
DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)
→
ShowSugerencia (Sugerencia)
→
CloseWindow5 (Forma5).SELECTC )

+
/*VISUALIZAR RESULTADOS*/
(ShowWindow6 (Forma6). Pi1

Pi1 =
(CloseWindow6 (Forma6) . SELECTC
+
(OnClickIDPatient (IDPaciente)
→
ShowResults (EnfClin, CerClin, EnfLab, CetLab, Terapia, UsuarioClin, UsuarioLab )
→
CloseWindow6 (Forma6) ).SELECTC ) )

+
/*FINALIZAR*/
end;

```

End_Presentation Aspect PClinical;

Aspecto de distribución

Este aspecto va a especificar la ubicación de la máquina desde la cual el usuario manipula al sistema, permitiendo que se pueda utilizar el sistema desde distintas máquinas. Por lo que el aspecto de distribución del USUARIO CLINICO va a ser dinámico y va a proveer servicios de movilidad.

La especificación de este aspecto presenta la siguiente sintaxis:

Distribution Aspect DUsers

Attributes

Constant

```

location: loc, NOT NULL;
locationMax: loc, NOT NULL;
locationMin: loc, NOT NULL;

```

Services

```

begin (input Loc: loc, LocMin:loc, LocMax:loc);

```

Valuations

```

[begin (Loc; LocMin, LocMax)] location:= Loc, locationMin:=LocMin,
locationMax:=LocMax;

```

```

checkLocation (input Loc:loc, output Check:bool)

```

Valuations

```

{Loc>locMin and Loc<locMax}
[checkLocation (Loc, Check)] Check:= true;

{Loc<locMin and Loc>locMax}
[checkLocation (Location, Check)] Check:= false;

move (input NewLocation:loc)
Valuations
{NewLocation > locMin and NewLocation <locMax}
[move (NewLocation)] location:= NewLocation;

end;

```

Constraints

```

always {location>locationMin and location <locationMax};

```

Protocols

```

DUSERS = begin(Loc, LocMax, LocMin).DISTRIBUTION;

DISTRIBUTION =
    checkLocation(Loc, Check).DISTRIBUTION
+
    move(NewLocation).DISTRIBUTION
+
    end;

```

End_Distribution Aspect DUsers;

Aspecto de replicación

En este aspecto, el componente USUARIO CLINICO con el servicio *replicate* permite que más de un usuario pueda estar lanzando peticiones al sistema de forma simultánea.

```

replicate (input NewLocation:loc);

```

Este aspecto aunque cuenta con el servicio *replicate*, éste es interno y no público, ya que sólo es utilizado internamente y no para la comunicación con otros elementos arquitectónicos. Por ello, el aspecto no usa ninguna interfaz ni contiene *played_roles*.

Con base en el análisis anterior, el aspecto de replicación del componente USUARIO CLÍNICO se especifica a continuación:

Replication Aspect RUsers

Services

```

begin;
replicate (input NewLocation:loc);
end;

```

Protocol

```

RUSERS = begin.REPLICATION;

```

```

REPLICATION =
    replicate (NewLocation ).REPLICATION
+
    end;

```

End_Replication Aspect RUsers;

3.3.1.13 Especificación del Componente Usuario Clínico

Tras haberse especificado los aspectos necesarios para el componente USUARIO CLÍNICO, a continuación se va a definir el *Component_Type ClinicUser*, el cual será un tipo, de forma tal que el tipo *ClinicUser* será instanciado para crear la USUARIO CLÍNICO que se ha diseñado en el modelo arquitectónico.

ClinicUser especifica un puerto necesario para comunicarse con otros elementos arquitectónicos. Para definir el puerto, se detalla el nombre (*UserClinPort*), el tipo que corresponde a la interfaz declarada (*IClinical*) y el *played_role* (*UserClin*), que especifica el comportamiento del conjunto de servicios que forman la interfaz que tipa al puerto. El *played_role* se encuentra definidos en el aspecto funcional (*FClinical*) que importa el componente *ClinicUser*:

```

UserClinPort: IClinical
Played_Role FClinical.UserClin

```

El componente tipo *ClinicUser* importa dos aspectos, el funcional (*FClinical*) y el de distribución (*DUsers*) con el objetivo de indicar su ubicación dentro del sistema:

```

Function Aspect import FClinical;
Distribution Aspect import DUsers;
Replication Aspect import RUsers;
Presentation Aspect import PClinical;

```

En la sección *Initialize* se especifica el inicio de los aspectos (*FClinical*, *DUsers*, *RUsers* y *PClinical*) que lo forman. Para crear la instancia es necesario dar un valor inicial a los atributos requeridos *location*, *locationMin* y *locationMax* de su aspecto de distribución. Estos atributos tienen la propiedad de *not null*:

```

new (NewLocation:loc, Loc:loc, LocMin:loc, LocMax:loc)
{
    FClinical.begin ();
    DUsers.begin (Location, LocMin, LocMax);
    RUsers.begin (NewLocation);
    PClinical.begin ();
}

```

En la sección *Destruction* se especifica el término de los aspectos (*FClinical*, *DUsers*, *RUsers* y *PClinical*) que lo forman:

```
destroy ()
{
    FClinical.end ();
    DUsers.end ();
    RUsers.end ();
    PClinical.end ();
}
```

El hecho de que se introduzcan el aspecto de presentación en este componente, hace que se deban sincronizar mediante *weavings* los aspectos funcional y el de presentación; de forma tal que un *weaving* define que la ejecución de un servicio del aspecto funcional puede generar la invocación de un servicio del aspecto de presentación.

Estos *weavings* se realizan mediante los operadores *before*, *afterif*, e *instead*. El operador *before* establece una sincronización temporal entre los servicios que participan en ellos. El segundo operador *afterif* además de establecer una sincronización temporal entre los servicios que participan en ellos, introduce una semántica de obligación, ya que sólo se ejecutará el segundo evento si y sólo si se satisface la condición asociada al operador. Finalmente el tercer operador *instead* establece que el primer servicio es ejecutado en lugar del segundo servicio.

A continuación se relacionan cada uno de los *weavings*, con la semántica asociada a su especificación:

Weaving:

```
FClinical.IntroducirLoginPassword (Login, Password)
before if (login=usuarioMedico) and (Password =contraseñaMedico)
PClinical.ShowWindow2 (Forma2)
```

Semántica: El servicio *introducirPassword* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *showWindow2* del aspecto de presentación *PClinical*, **sí y sólo si** los parámetros *Login* y *Password* del servicio *introducirPassword* son igual a *usuarioMédico* y *contraseñaMédico*, respectivamente.

Weaving

```
FClinical.IntroducirLoginPassword (Login, Password)
before if (Login<>usuarioMedico) and (Password<>contraseñaMedico )
PClinical.ShowMessage (“Tus datos no son correctos”)
```

Semántica: El servicio *introducirLoginPassword* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *showMessage* del aspecto de presentación *PClinical*, **sí y sólo si** los parámetros *Login* y *Password* del servicio *introducirLoginrPassword* no son igual a *usuarioMédico* y *contraseñaMédico*, respectivamente.

Weaving

```
FClinical.IntroducirSeleccion (Seleccion)
```

before if Seleccion=5
PClinical.ShowMessage (“No tienes permiso para esta seleccion”)

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *showMessage* del aspecto de presentación *PClinical*, **sí y sólo si** el parámetro *Seleccion* del servicio *introducirSeleccion* es 5.

Weaving

FClinical.IntroducirSeleccion (Seleccion)
before if Seleccion=1
PClinical.IntroducirPatient (IDPaciente, DNI, Nombre, Direccion, Tel, Email,. Sexo, FechaNac)

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *introducirPatient* del aspecto de presentación *PClinical*, **sí y sólo si** el parámetro *Seleccion* del servicio *introducirSeleccion* es 1.

Weaving

FClinical.IntroducirSeleccion (Seleccion)
before if (Seleccion=2 **or** Seleccion=3 **or** Seleccion=4)
PClinical.IntroducirIDPatient (IDPaciente)

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *introducirPatient* del aspecto de presentación *PClinical*, **sí y sólo si** el parámetro *Seleccion* del servicio *introducirSeleccion* es 2 ó 3 ó 4.

Weaving

FClinical.IntroducirSeleccion (Seleccion)
before if Seleccion=6
PClinical.End

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *end* del aspecto de presentación *PClinical*, **sí y sólo si** el parámetro *Seleccion* del servicio *introducirSeleccion* es 6.

Weaving

FClinical.IntroducirIDPaciente (IDPaciente)
after
PClinical.OnclickUserClinIDPatient (UsuarioClin, IDPaciente)

Semántica: El servicio *introducirIDPaciente* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclickUserClinIDPatient* del aspecto de presentación *PClinical*.

Weaving

FClinical.IntroducirUserClin (UsuarioClin))
after
PClinical.OnclickUserClinIDPatient (UsuarioClin, IDPaciente)

Semántica: El servicio *introducirUserClin* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclickUserClinIDPatient* del aspecto de presentación *PClinical*.

Weaving

FClinical.IntroducirLoginPassword (Login, Password)

after

PClinical.OnclickLoginPassword (Login, Contraseña)

Semántica: El servicio *servicio introducirLoginPassword* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclickLoginPassword* del aspecto de presentación *PClinical*.

Weaving

FClinical.IntroducirSelection (Seleccion)

after

PClinical.OnclickSelection (Seleccion)

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclickSelection* del aspecto de presentación *PClinical*.

Weaving

FClinical.IntroducirIDPaciente (IDPaciente)

after

PClinical.OnclickIDPatient (IDPaciente)

Semántica: El servicio *introducirIDPaciente* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclickIDPatient* del aspecto de presentación *PClinical*.

Weaving

FClinical.IntroducirPaciente (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

after

PClinical.OnclickDataPatient (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

Semántica: El servicio *introducirPaciente* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclickDataPatient* del aspecto de presentación *PClinical*.

Weaving

FClinical.IntroducirIDPaciente (IDPaciente)

after

PClinical.OnclickResults (IDPaciente)

Semántica: El servicio *introducirIDPaciente* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclickResults* del aspecto de presentación *PClinical*.

Weaving

FClinical.ObtenerSsgg (Fiebre, DolorMasticacionI, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

instead

PClinical.ShowSSGG (Fiebre, DolorMasticacionI, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

Semántica: El servicio *obtenerSsgg* del aspecto de presentación *FClinical* se ejecutará **en lugar de** el servicio *showSSGG* del aspecto funcional *PClinical*.

Weaving

FClinical.ObtenerSsgg (Fiebre, DolorMasticacionI, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

after

PClinical.OnclickSSGG (Fiebre, DolorMasticacionI, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

Semántica: El servicio *obtenerSsgg* del aspecto funcional *FClinical* se ejecutará **después** del servicio *onclikSSGG* del aspecto de presentación *PClinical*.

Weaving

FClinical.ObtenerSsgf (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)

instead

PClinical.ShowSSGF (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)

Semántica: El servicio *obtenerSsgf* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *showSSGF* del aspecto de presentación *PClinical*.

Weaving

FClinical.ObtenerSsgf (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)

before

PClinical.OnclickSSGF (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)

Semántica: El servicio *obtenerSsgf* del aspecto funcional se ejecutará **después** del servicio *FClinical onclickSSGF* del aspecto de presentación *PClinical*.

Weaving

FClinical.TomarPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

before

PClinical.ShowPatient (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

Semántica: El servicio *tomarPaciente* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *showPatient* del aspecto de presentación *PClinical*.

Weaving

FClinical.TomarResultados (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)

before

PClinical.ShowResults (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)

Semántica: El servicio *tomarResultados* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *showResults* del aspecto de presentación *PClinical*.

Weaving

FClinical.TomarSugerencia (Sugerencia)

before

PClinical.ShowSugerencia (Sugerencia)

Semántica: El servicio *tomarSugerencia* del aspecto funcional *FClinical* se ejecutará **antes** del servicio *showSugerencia* del aspecto de presentación *PClinical*.

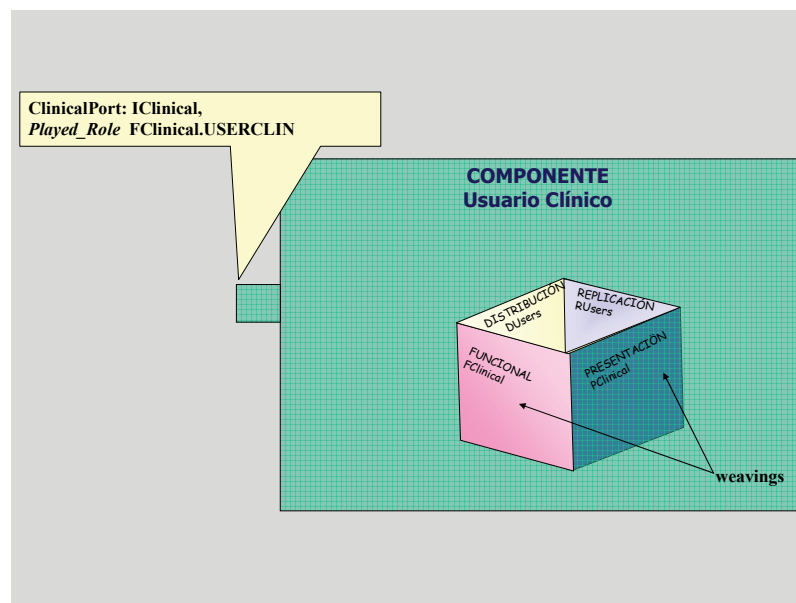


Fig. 49 Componente USUARIO CLÍNICO

Con base en el análisis anterior, el componente USUARIO CLÍNICO se especifica a continuación:

Component_Type ClinicUser

Ports

UserClinPort: IClinical

Played_Role FClinical.USERCLIN

End_Ports;

Function Aspect import FClinical;

Distribution Aspect import DUsers;
Replication Aspect import RUsers;
Presentation Aspect import PClinical;

Weavings

FClinical.IntroducirLoginPassword (Login, Password)
before if (login=usuarioMedico) **and** (Password =contraseñaMedico)
PClinical.ShowWindow2 (Forma2)

FClinical.IntroducirLoginPassword (Login, Password)
before if (Login<>usuarioMedico) **and** (Password<>contraseñaMedico)
PClinical.ShowMessage (“Tus datos no son correctos”)

FClinical.IntroducirSeleccion (Seleccion)
before if Seleccion=5
PClinical.ShowMessage (“No tienes permiso para esta seleccion”)

FClinical.IntroducirSeleccion (Seleccion)
before if Seleccion=1
PClinical.IntroducirPatient ()

FClinical.IntroducirSeleccion (Seleccion)
before if (Seleccion=2 **or** Seleccion=3 **or** Seleccion=4)
PClinical.IntroducirIDPatient ()

FClinical.IntroducirSeleccion (Seleccion)
before if Seleccion=6
PClinical.End

FClinical.IntroducirIDPaciente (IDPaciente)
after
PClinical.OnclickUserClinIDPatient (UsuarioClin, IDPaciente)

FClinical.IntroducirUserClin (UsuarioClin))
after
PClinical.OnclickUserClinIDPatient (UsuarioClin, IDPaciente)

FClinical.IntroducirLoginPassword (Login, Password)
after
PClinical.OnclickLoginPassword (Login, Contraseña)

FClinical.IntroducirSelection (Seleccion)
after
PClinical.OnclickSelection (Seleccion)

FClinical.IntroducirIDPaciente (IDPaciente)
after
PClinical.OnclickIDPatient (IDPaciente)

FClinical.IntroducirPaciente (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
after
PClinical.OnclickDataPatient (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

FClinical.IntroducirIDPaciente (IDPaciente)
after
PClinical.OnclickResults (IDPaciente)

FClinical.ObtenerSsgg (Fiebre, DolorMasticacionl, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)
instead

PClinical.ShowSSGG (Fiebre, DolorMasticacionl, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

FClinical.ObtenerSsgg (Fiebre, DolorMasticacionl, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

after

PClinical.OnclickSSGG (Fiebre, DolorMasticacionl, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

FClinical.ObtenerSsgg (Fiebre, DolorMasticacionl, DolorCrecimientoParotidas, Tos, DificultadRespiratoria)

instead

PClinical.ShowSSGF (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)

FClinical.ObtenerSsgf (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)

before

PClinical.OnclickSSGF (FiebreContinua, FiebreMayor37, FiebreMayor38, FiebreMayor39, Fiebre2A3Dias, TosSeca, TosPorAccesos, TosConFlema, TosFrecuente, DificultadRespiratoriaGrave, DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar, OdinofagiaGargantaRoja, Odinofagia2A3Dias, ArdorGargantaIntenso, ApneaPeriodosLargos, DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo, DolorMasticacionConBorramientoAnguloMandibular, RinoreaTransparente, DificultadRespiratoriaGarveAModerada, EstrindorLaringeo)

FClinical.TomarPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

before

PClinical.ShowPatient (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

FClinical.TomarResultados (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)

before

PClinical.ShowResults (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)

FClinical.TomarSugerencia (Sugerencia)

before

PClinical.ShowSugerencia (Sugerencia)

End_Weavings

Initialize

```
new (NewLocation:loc, Loc: loc. LocMin:locMin, LocMax:locMax)
{
    FClinical.begin ();
    DUsers.begin (Loc. LocMin, LocMax);
    RUsers.begin (NewLocation);
    PClinical.begin ();
}
```

End_Initialize;

Destruction

```

    destroy ()
    {
        FClinical.end();
        DUsers.end ();
        RUsers.end ();
        PClinical.end ();
    }
End_Destruction;

End_Connector_Type ClinicUser;

```

3.3.1.14 Instanciación del Componente Usuario Clínico

En la configuración, el servicio *new* se invoca y da valor en la creación de la instancia, como se muestra a continuación:

```
new ClinicUser (newLocalhost, localhost, 198.120.0.0, 198.120.0.100);
```

3.2.1.15 Especificación de los aspectos del Componente Usuario Laboratorio

Aspecto funcional

- En la selección “Visualizar Paciente” se solicita al USUARIO LABORATORIO con el servicio *IntroducirIDPaciente*, el identificador del paciente del cual se desean consultar sus datos con el que se registró.

IntroducirIDPaciente (**output** IDPaciente:identifier)

Finalmente el USUARIO LABORATORIO recibe el servicio *TomarPaciente* que le sirve para adquirir la información del paciente que se mostrará al usuario (ver aspecto de presentación).

TomarPaciente (**input** DNI:string, **input** Nombre:string, **input** Dirección:string, **input** Tel:string, **input** Email:string, **input** Sexo:char, **input** FechaNac:date).

- Para “Realizar el Diagnóstico de Laboratorio”, al USUARIO LABORATORIO se le solicita con el servicio *IntroducirUsuarioLab* el nombre del laboratorista responsable de realizarle los estudios de laboratorio al paciente.

IntroducirUsuarioLab (**output** UsuarioLab:string)

Posteriormente este componente recibe a través del servicio *IntroducirIDPaciente* el identificador del paciente.

IntroducirIDPaciente (**output** IDPaciente:identifier)

A continuación el USUARIO LABORATORIO recibe el servicio *TomarSugLab* con el fin de que el usuario obtenga las sugerencia de estudios de laboratorio que se le podrán realizar al paciente.

TomarSugLab (**input** Sugerencia:string)

Finalmente, el servicio *ObtenerEstsLab* es recibido por el USUARIOLAB con el que se solicitan los resultados de los estudios de laboratorio realizados al paciente.

ObtenerEstsLabPaperas (**output** BiometriaHematicaPositiva:bool)

obtenerEstsLabParotidisBacteriana (**output** BiometriaHematicaLeucositosis:bool)

ObtenerEstsLabBronquiolitis (**output** BiometriaHematicaLeucositosis:bool)

ObtenerEstsLabNeumonia (**output**
ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi:bool)

ObtenerEstsLabCrup (**output** BiometriaHematicaEusinofilia:bool)

Los servicios antes expuestos se publican en el puerto denominado *UserLabPort* del componente USUARIO LABORATORIO, que tiene asociado como signatura la interfaz *ILaboratory* y que da semántica a su aspecto funcional.

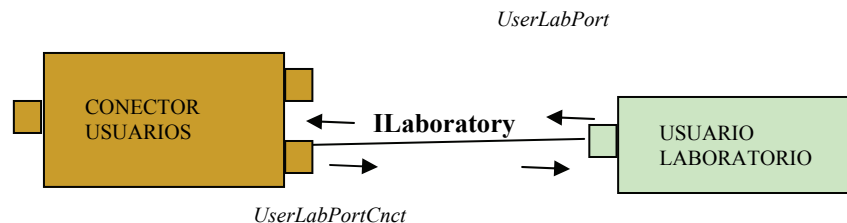


Fig. 44 Esquema del funcionamiento de la Interfaz ILaboratory

La especificación de *ILaboratory* muestra la siguiente sintaxis:

Interfaz ILaboratory

introducirLoginPassword (**output** Login:string, **output** Password:string)

introducirSeleccion (**output** Seleccion:int)

introducirIDPaciente (**output** IDPaciente:identifier)

introducirUsuarioLab (**output** UsuarioLab:string)

obtenerEstsLabPaperas (**output** BiometriaHematicaPositiva:bool)

obtenerEstsLabParotidisBacteriana (**output** BiometriaHematicaLeucositosis:bool)

obtenerEstsLabBronquiolitis (**output** BiometriaHematicaLeucositos:bool)

obtenerEstsLabNeumonia (**output**
ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi:bool)

obtenerEstsLabCrup (**output** BiometriaHematicaEusinoftia:bool)
tomarSugLab (**input** SugLab:string)

End_Interfaz ILaboratory

Con base en el análisis anterior, el aspecto funcional del componente USUARIO LABORATORIO se especifica a continuación:

Functional Aspect FLaboratory **using** ILaboratory

Services

begin

in introducirLoginPassword (**output** Login:string, **output** Password:string)

in introducirSeleccion (**output** Selección:int)

in introducirIDPaciente (**output** IDPaciente:identifier)

in introducirUsuarioLab (**output** UsuarioLab:string)

in obtenerEstsLabPaperas (**output** BiometriaHematicaPositiva:bool)

in obtenerEstsLabParotidisBacteriana (**output** BiometriaHematicaLeucositosis:bool)

in obtenerEstsLabBronquiolitis (**output** BiometriaHematicaLeucositosis:bool)

in obtenerEstsLabNeumonia (**output**
ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi:bool)

in obtenerEstsLabCrup (**output** BiometriaHematicaEusinoftia:bool)

in tomarPaciente (**input** DNI:string, **input** Nombre:string, **input** Direccion:string,
input Tel:string, **input** Email:string, **input** Sexo:char, **input** FechaNac:date)

in tomarSugLab (**input** SugLab:string)

end;

Played_Roles

USERLAB=

ILaboratory.IntroducirPassword ? (Password)

→

ILaboratory.IntroducirPassword ! (Password)

→

ILaboratory.IntroducirSeleccion ? (Seleccion)

→

ILaboratory.IntroducirSeleccion ! (Seleccion)

→

(

/*VISUALIZAR PACIENTE*/

(ILaboratory.IntroducirIDPaciente ? (IDPaciente)

→

ILaboratory.IntroducirIDPaciente ! (IDPaciente)

→

ILaboratory.TomarPaciente? (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))

```

+
/*REALIZAR DIAGNOSTICO LABORATORIO*/
(ILaboratory.IntroducirUsuarioLab ? (UsuarioLab )
→
ILaboratory.IntroducirUsuarioLab ! (UsuarioLab) )
→
ILaboratory.IntroducirIDPaciente ? (IDPaciente)
→
ILaboratory.IntroducirIDPaciente ! (IDPaciente)
→
ILaboratory.TomarSugLab ? (SugLab)
→
(
(
ILaboratory.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
ILaboratory.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
(ILaboratory.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
^
ILaboratory.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
ILaboratory.ObtenerEstsLabCrup ? (BiometriaHematicaEusinoFilia) )
)
→
(
(
ILaboratory.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
ILaboratory.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
(ILaboratory.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
^
ILaboratory.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
ILaboratory.ObtenerEstsLabCrup ! (BiometriaHematicaEusinoFilia) )
).

```

Protocols

```

FLABORATORY = begin.FUNCTIONALUL;

```

```

FUNCTIONALUL =
  ( USERLAB.IntroducirLoginPassword ? (Login, Contraseña)
  →
  USERLAB.IntroducirLogin, Password ! (Login, Contraseña) ).P0

```

```

P0 =
  ( USERLAB.IntroducirSeleccion ? (Seleccion)
  →
  USERLAB.IntroducirSeleccion ! (Seleccion) ).OPTIONL

```

```

OPTIONL =
  VisualizarPaciente
+
  RealizarDiagnosticoLaboratorio
+
  Finalizar;

```

```

VisualizarPaciente =
    (USERLAB.IntroducirIDPaciente ? (IDPaciente )
    →
    USERLAB.IntroducirIDPaciente ! (IDPaciente )
    →
    USERLAB.TomarPaciente? (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
    ).PO

RealizarDiagnosticoLaboratorio =
    (USERLAB.IntroducirUsuarioLab ? (UsuarioLab)
    →
    USERLAB.IntroducirUsuarioLab ! (UsuarioLab) )
    →
    (USERLAB.IntroducirIDPaciente ? (IDPaciente )
    →
    USERLAB.IntroducirIDPaciente ! (IDPaciente)
    →
    USERLAB.TomarSugLab ? (SugLab)
    →
    (
    ( USERLAB.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
    ^
    USERLAB.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
    +
    (USERLAB.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
    ^
    USERLAB.ObtenerEstsLabNeumonia ?
    (ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
    ^
    USERLAB.ObtenerEstsLabCrup ? (BiometriaHematicaEusino filia) )
    )
    →
    (
    ( USERLAB.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
    ^
    USERLAB.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
    +
    (USERLAB.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
    ^
    USERLAB.ObtenerEstsLabNeumonia !
    (ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
    ^
    USERLAB.ObtenerEstsLabCrup ! (BiometriaHematicaEusino filia) )
    ).

Finalizar =
    end;

```

End Functional Aspect FLaboratory

Aspecto de presentación

En este aspecto, que alberga la parte gráfica para la comunicación del usuario con el sistema, son utilizados unos servicios para mostrar la información, ya sea la requerida

por el usuario al visualizar los datos del registro del paciente (*showPatient*), así como la que se le es mostrada por el propio sistema (*showMessage*, *showEstsLab*), para que con los servicios *onclickLoginPassword*, *onclickSelect*, *onclickIDPatient*, *onclickUserLabIDPatient*, *onClickEstsLab* se obtenga del usuario la información requerida para el funcionamiento del sistema (nombre del usuario y contraseña, opción elegida, identificador del paciente, nombre del laboratorista, resultados de los estudios de laboratorio).

```
onclickIDPatient (output IDPaciente: identifier);

showPatient (input DNI:string, input Nombre:string, input Direccion:string,
             input Tel:string, input Email:string, input Sexo:string, input FechaNac:string);

showSugLab ( input SugLab: string);

showEstsLab ( input BiometriaHematicaPositiva:bool,
             input BiometriaHematicaLeucositosis:bool, input BiometriaHematicaLeucositos:bool
             input BiometriaHematicaEusinoofilia:bool,
             input ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool)

onclickEstsLab (output BiometriaHematicaPositiva:bool,
               output BiometriaHematicaLeucositosis:bool, output BiometriaHematicaLeucositos:bool
               output BiometriaHematicaEusinoofilia:bool,
               output ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool);

onclickUserLabIDPatient (output UsuarioLab: string, output IDPaciente: identifier);

showWindow1 (input Forma1: window);

showWindow2 (input Forma2: window);

showWindow3 (input Forma3: window);

showWindow4 (input Forma4: window);

closeWindow1 (input Forma1: window);

closeWindow2 (input Forma2: window);

closeWindow3 (input Forma3: window);

closeWindow4 (input Forma4: window);

onclickSelect (output Seleccion: int);

onclickLoginPassword (output Login:string, output Contraseña:string);

showMessage (input Mensaje: string);
```

Una característica de este aspecto de presentación es que no utiliza ninguna interfaz ni contiene `played_roles`, ya que aunque cuenta con servicios que son utilizados internamente y no para la comunicación con otros elementos arquitectónicos, i.e. no son servicios públicos.

Una muestra de los gráficos anteriormente comentados adquiere la siguiente vista:

The figure displays four screenshots of a software interface for a laboratory user, arranged vertically. Each screenshot shows a different screen with a title bar and standard window controls (minimize, maximize, close).
1. **CONTRASEÑAS**: A login screen with two text input fields labeled 'Login' and 'Password', and a green 'ACEPTAR' button at the bottom right.
2. **MENU**: A menu screen with three options: '2.- Visualizar datos del paciente', '4.- Diagnóstico Laboratorio', and '6.- Salir'. Below the options is a 'Selección' field and a green 'ACEPTAR' button.
3. **VISUALIZAR DATOS PACIENTE**: A screen for viewing patient data with eight text input fields: 'ID Paciente', 'DNI', 'Nombre', 'Dirección', 'Teléfono', 'E-mail', 'Sexo', and 'Fecha Nac.'. A green 'ACEPTAR' button is at the bottom right.
4. **DIAGNÓSTICO LABORATORIO**: A diagnostic screen with four text input fields: 'Laboratorista', 'ID Paciente', 'Sugerencia', and 'Resultados laboratorio'. There are also two small buttons (one green with 'i', one red with 'o') and a green 'ACEPTAR' button at the bottom right.

Fig. 45 Gráficos que puede utilizar el aspecto de presentación del USUARIO LABORATORIO

Con base en el análisis anterior, el aspecto de presentación del componente USUARIO CLÍNICO se especifica a continuación:

Presentation Aspect PLaboratory

Attributes
Variable

titleWindow: window;
label1, label2, label3, label4, label5, label6, label7: label
button1, button2, button3, button4: button;
text1, text2, text3: text;

Derivations

forma1:= {titleWindow=CONTRASEÑAS **and** label1=Log in **and** label2=Contraseña **and** button1=ACEPTAR }

forma2:= {titleWindow=MENU **and** text1=2.-Visualizar datos del paciente **and** text2=3.-Diagnostico Laboratorio **and** text3:-6.-Salir **and** label1=Selección **and** button1=ACEPTAR }

forma3:= {titleWindow=VISUALIZAR DATOS DEL PACIENTE **and** label1=ID Paciente, label2=Nombre **and** label3= Direccion **and** label4=Teléfono **and** label5=Email **and** label6:Sexo **and** label7:Fecha Nac **and** button1=ACEPTAR }

forma4:= {titleWindow =DIAGNOSTICO DE LABORATORIO **and** label1=Laboratorista **and** label2=ID Paciente **and** label3=Sugerencia de estudios **and** label4=Resultados de laboratorio presentes: **and** button1=SI **and** button2=NO button3=ACEPTAR }

Services

begin ();

onclickIDPatient (**output** IDPaciente: identifier);

showPatient (**input** DNI:string, **input** Nombre:string, **input** Direccion:string,
input Tel:string, **input** Email:string, **input** Sexo:string, **input** FechaNac:string);

showSugLab (**input** SugLab: string);

showEstsLab (**input** BiometriaHematicaPositiva:bool,
input BiometriaHematicaLeucositosis:bool, **input** BiometriaHematicaLeucositos:bool
input BiometriaHematicaEusinoofilia:bool,
input ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool)

onclickEstsLab (**output** BiometriaHematicaPositiva:bool,
output BiometriaHematicaLeucositosis:bool, **output** BiometriaHematicaLeucositos:bool
output BiometriaHematicaEusinoofilia:bool,
output ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool);

onclickUserLabIDPatient (**output** UsuarioLab: string, **output** IDPaciente: identifier);

showWindow1 (**input** Forma1: window);

showWindow2 (**input** Forma2: window);

showWindow3 (**input** Forma3: window);

showWindow4 (**input** Forma4: window);

closeWindow1 (**input** Forma1: window);

closeWindow2 (**input** Forma2: window);

closeWindow3 (**input** Forma3: window);

closeWindow4 (**input** Forma4: window);

onclickSelect (**output** Seleccion: int);

onclickLoginPassword (**output** Login:string, **output** Contraseña:string);

```

showMessage (input Mensaje: string);

end;

```

Protocols

```

PLABORATORY = begin.PRESENTATIONL;

PRESENTATIONL =
    ShowWindow1 (Forma1).Po
+
    end;

Po =
    CloseWindow1 (Forma1).PRESENTATIONL
+
    OnclickLoginPaswword (Login, Contraseña).P1

P1 =
    ShowMessage (Mensaje).Po
+
    (CloseWindow1 (Forma1)
    →
    ShowWindow2 (Forma2) ).P2

P2 =
    OnclickSelect (Seleccion).P3

P3 =
    ShowMessage (Mensaje).P2
+
    CloseWindow2 (Forma2).SELECTL

SELECTL =
    (ShowWindow3 (Form3). Pb1
    /*VISUALIZAR PACIENTE*/

Pb1 =
    (CloseWindow3 (Forma3). SELECTL
+
    (OnclickIDPatient (IDPaciente)
    →
    ShowPatient (DNI, Nombre, Direccion, Tel, Email. Sexo, FechaNac)
    →
    CloseWindow3 (Forma3) ).SELECTL ) )

+
    /*REALIZAR DIAGNOSTICO LABORATORIO*/
    (ShowWindow4 (Forma4).Ph1

Ph1 =
    CloseWindow4 (Forma4).SELECTL
+
    (OnclickUserLabIDPatient (UsuarioLab, IDPaciente )

    ShowSugLab (SugLab)
    →
    ShowEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis,
    BiometriaHematicaLeucositos, BiometriaHematicaEusinoфиlia,
    ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi))
    →

```

```

OnclickEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis,
BiometriaHematicaLeucositos, BiometriaHematicaEusinofilia,
ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
→
CloseWindow4 (Forma4 ).SELECTL ) ) )

```

```

+
end;
/*FINALIZAR*/

```

End_Presentation Aspect PLaboratory;

Aspecto de distribución

Este aspecto va a especificar la ubicación de la máquina desde la cual el usuario manipula al sistema, permitiendo que se pueda utilizar el sistema desde distintas máquinas.

Su especificación se puede encontrar en el apartado del aspecto de distribución del componente USUARIO CLÍNICO, dado que ambos componentes utiimportan dicho aspecto.

Aspecto de replicación

En este aspecto, se contempla que más de un usuario pueda estar lanzando peticiones al sistema de forma simultánea.

El aspecto de replicación de los componentes USUARIO LABORATORIO y USUARIO CLINICO es el mismo, por lo que su especificación no será (nuevamente) dada en este apartado.

3.3.1.16 Especificación del Componente Usuario Laboratorio

Tras haberse especificado los aspectos necesarios para el componente USUARIO LABORATORIO, a continuación se va a definir el *Component_Type LaboratoryCabinetUser*, el cual será un tipo, de forma tal que el tipo *LaboratoryCabinetUser* será instanciado para crear la USUARIO LABORATORIO que se ha diseñado en el modelo arquitectónico.

LaboratoryCabinetUser especifica un puerto necesario para comunicarse con otros elementos arquitectónicos. Para definir el puerto, se detalla el nombre (*UserLabPort*), el tipo que corresponde a la interfaz declarada (*ILaboratory*) y el *played_role* (*UserLab*), que especifica el comportamiento del conjunto de servicios que forman la interfaz que tipa al puerto. El *played_role* se encuentra definidos en el aspecto funcional (*FLaboratory*) que importa el componente *LaboratoryCabinetUser*:

```
UserLabPort: ILaboratory
  Played_Role FLaboratory.UserLab
```

El componente tipo *LaboratoryCabinetUser* importa cuatro aspectos, el funcional (*FLaboratory*), el de distribución (*DUsers*), el de replicación (*RUsers*) y el de presentación (*PLaboratory*), con el objetivo de indicar su ubicación dentro del sistema.

```
Function Aspect import FLaboratory;
Distribution Aspect import DUsers;
Replication Aspect import RUsers;
Presentation Aspect import PLaboratory;
```

En la sección *Initialize* se especifica el inicio de los aspectos (*FLaboratory*, *DUsers*, *RUsers* y *PLaboratory*) que lo forman. Para crear la instancia es necesario dar un valor inicial a los atributos requeridos *location*, *locationMin* y *locationMax* de su aspecto de distribución. Estos atributos tienen la propiedad de *not null*.

```
new (NewLocation:loc, Loc: loc, LocMin:locMin, LocMax:locMax)
  { FLaboratory.begin ();
    DUsers.begin (Loc, LocMin, LocMax);
    RUsers.begin (NewLocation);
    PLaboratory.begin ();
  }
```

En la sección *Destruction* se especifica el término de los aspectos (*FLaboratory*, *DUsers*, *RUsers* y *PLaboratory*) que lo forman:

```
destroy ()
  { FLaboratory.end ();
    DUsers.end ();
    RUsers.end ();
    PLaboratory.end ();
  }
```

El hecho de que se introduzcan el aspecto de presentación en este componente, hace que se deban sincronizar mediante *weavings* los aspectos funcional y el de presentación, de forma tal que un *weaving* define que la ejecución de un servicio del aspecto funcional puede generar la invocación de un servicio del aspecto de presentación.

Estos *weavings* se realizan mediante los operadores *before*, *afterif*, e *instead*. El operador *before* establece una sincronización temporal entre los servicios que participan en ellos. El segundo operador *afterif* además de establecer una sincronización temporal

entre los servicios que participan en ellos, introduce una semántica de obligación, ya que sólo se ejecutará el segundo evento si y sólo si se satisface la condición asociada al operador. Finalmente el tercer operador *instead* establece que el primer servicio es ejecutado en lugar del segundo servicio.

A continuación se relacionan cada uno de los weavings, con la semántica asociada a su especificación:

Weaving:

```
FLaboratory.IntroducirLoginPassword (Login, Password)
before if (Login=usuarioLab) and (Password=contraseñaLab )
PLaboratory.ShowWindow2 (Forma2)
```

Semántica: El servicio *introducirLoginPassword* del aspecto funcional *FLaboratory* se ejecutará **antes** del servicio *showWindow2* del aspecto de presentación *PLaboratory*, **sí y sólo si** los parámetros *Login* y *Password* del servicio *introducirLoginPassword* no son igual a *usuarioLab* y *contraseñaLab*, respectivamente.

Weaving:

```
FLaboratory.IntroducirPassword (Password)
before if (Login<> usuarioLab) and (Password <> contraseñaLab)
PLaboratory.ShowMessage (“Tus datos no son correctos”)
```

Semántica: El servicio *introducirPassword* del aspecto funcional *FLaboratory* se ejecutará **antes** del servicio *showMessage* del aspecto de presentación *PLaboratory*, **sí y sólo si** los parámetros *Login* y *Password* del servicio *introducirPassword* no son igual a *usuarioLab* y *contraseñaLab*, respectivamente.

Weaving:

```
FLaboratory.IntroducirSeleccion (Seleccion)
before if (Seleccion=1 or Selección= 3 or Selección= 4)
PLaboratory.ShowMessage (“No tienes permiso para esta seleccion”)
```

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FLaboratory* se ejecutará **antes** del servicio *showMessage* del aspecto de presentación *PLaboratory*, **sí y sólo si** el parámetro *Seleccion* del servicio *introducirSeleccion* es 1 ó 3 ó 4.

Weaving:

```
FLaboratory.IntroducirSeleccion (Seleccion)
before if (Seleccion=2 or Seleccion=5)
PLaboratory.IntroducirIDPatient(IDPaciente)
```

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FLaboratory* se ejecutará **antes** del servicio *introducirIDPatient* del aspecto de presentación *PLaboratory*, **sí y sólo si** el parámetro *Seleccion* del servicio *introducirSeleccion* es 2 ó 5.

Weaving:

```
FLaboratory.IntroducirSeleccion (Seleccion)
before if Seleccion=6
```

PLaboratory.End

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FLaboratory* se ejecutará **antes** del servicio *end* del aspecto de presentación *PLaboratory*, **sí y sólo si** el parámetro *Seleccion* del servicio *introducirSeleccion* es 6.

Weaving:

FLaboratory.IntroducirLoginPassword (Login, Password)

after

PLaboratory.OnclickLoginPassword (Login, Contraseña)

Semántica: El servicio *introducirLoginPassword* del aspecto funcional *FLaboratory* se ejecutará **después** del servicio *onclickLoginPassword* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.IntroducirSeleccion (Seleccion)

before

PLaboratory.OnclickSelection (Seleccion)

Semántica: El servicio *introducirSeleccion* del aspecto funcional *FLaboratory* se ejecutará **después** del servicio *onclickSelection* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.IntroducirIDPaciente (IDPaciente)

after

PLaboratory.OnclickIDPatient (IDPaciente)

Semántica: El servicio *introducirIDPaciente* del aspecto funcional *FLaboratory* se ejecutará **después** del servicio *onclickIDPatient* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.IntroducirIDPaciente (IDPaciente)

after

PLaboratory.OnclickUserLabIDPatient (UsuarioLab, IDPaciente)

Semántica: El servicio *introducirIDPaciente* del aspecto funcional *FLaboratory* se ejecutará **después** del servicio *onclickUserLabIDPatient* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.IntroducirUserLab (UsuarioLab))

after

PLaboratory.OnclickUserLabIDPatient (UsuarioLab, IDPaciente)

Semántica: El servicio *introducirUserLab* del aspecto funcional *FLaboratory* se ejecutará **después** del servicio *onclickUserLabIDPatient* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.ObtenerEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis, BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia, ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)

instead

PLaboratory.ShowEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis, BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia, ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)

Semántica: El servicio *obtenerEstsLab* del aspecto funcional *FLaboratory* se ejecutará **en lugar de** el servicio *showEstsLab* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.ObtenerEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis, BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia, ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)

after

PLaboratory.OnclickEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis, BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia, ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)

Semántica: El servicio *obtenerEstsLab* del aspecto funcional *FLaboratory* se ejecutará **después** del servicio *onclickEstsLab* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.TomarPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

before

PLaboratory.ShowPatient (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

Semántica: El servicio *tomarPaciente* del aspecto funcional *FLaboratory* se ejecutará **antes** del servicio *showPatient* del aspecto de presentación *PLaboratory*.

Weaving:

FLaboratory.TomarSugLab (SugLab)

before

PLaboratory.ShowSugLab (SugLab)

Semántica: El servicio *tomarSugLab* del aspecto funcional *FLaboratory* se ejecutará **antes** del servicio *showSugLab* del aspecto de presentación *PLaboratory*.

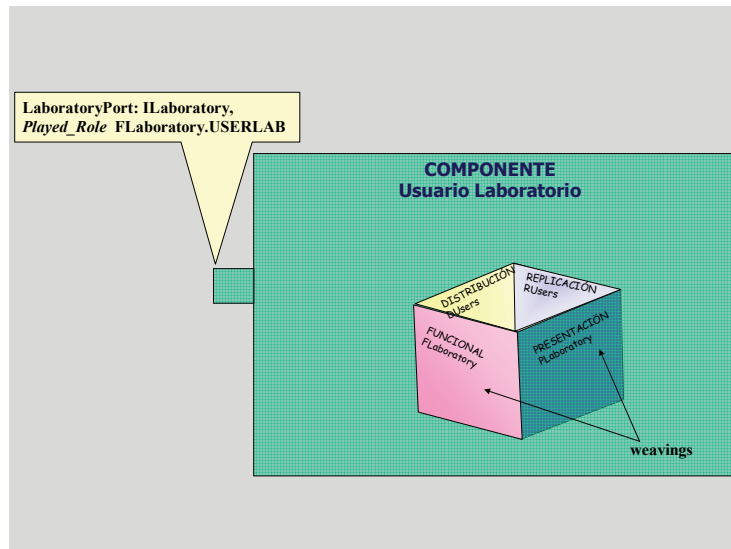


Fig. 46 Componente USUARIO LABORATORIO

Posteriormente al haber realizado el análisis del componente USUARIO LABORATORIO, se presenta a continuación su especificación:

Component_Type LaboratoryCabinetUser

Ports

UserLabPort: ILaboratory

Played_Role FLaboratory.UserLab

End_Ports;

Function Aspect import FLaboratory;

Distribution Aspect import DUsers;

Replication Aspect import RUsers;

Presentation Aspect import PLaboratory;

Weavings

FLaboratory.IntroducirLoginPassword (Login, Password)

before if (Login=usuarioLab) **and** (Password=contraseñaLab)

PLaboratory.ShowWindow2 (Forma2)

FLaboratory.IntroducirPassword (Password)

before if (Login<> usuarioLab) **and** (Password <> contraseñaLab)

PLaboratory.ShowMessage (“Tus datos no son correctos”)

FLaboratory.IntroducirSeleccion (Seleccion)

before if (Seleccion=1 **or** Selección= 3 **or** Selección= 4)

PLaboratory.ShowMessage (“No tienes permiso para esta seleccion”)

FLaboratory.IntroducirSeleccion (Seleccion)

before if (Seleccion=2 **or** Seleccion=5)

PLaboratory.IntroducirIDPatient()

FLaboratory.IntroducirSeleccion (Seleccion)

before if Seleccion=6

PLaboratory.End

FLaboratory.IntroducirLoginPassword (Login, Password)

after

PLaboratory.OnclickLoginPassword (Login, Contraseña)

```

FLaboratory.IntroducirSeleccion (Seleccion)
before
PLaboratory.OnclickSelection (Seleccion)

FLaboratory.IntroducirIDPaciente (IDPaciente)
after
PLaboratory.OnclickIDPatient (IDPaciente)

FLaboratory.IntroducirIDPaciente (IDPaciente)
after
PLaboratory.OnclickUserLabIDPatient (UsuarioLab, IDPaciente)

FLaboratory.IntroducirUserLab (UsuarioLab)
after
PLaboratory.OnclickUserLabIDPatient (UsuarioLab, IDPaciente)

FLaboratory.ObtenerEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis,
    BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia,
    ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
instead
PLaboratory.ShowEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis,
    BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia,
    ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)

FLaboratory.ObtenerEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis,
    BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia,
    ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
after
PLaboratory.OnclickEstsLab (BiometriaHematicaPositiva, BiometriaHematicaLeucositosis,
    BiometriaHematicaLeucositos, BiometriaHematicaEusinoofilia,
    ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)

FLaboratory.TomarPaciente (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)
before
PLaboratory.ShowPatient (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)

FLaboratory.TomarSugLab (SugLab)
before
PLaboratory.ShowSugLab (SugLab)
End_Weavings

```

Initialize

```

new (NewLocation:loc, Loc: loc. LocMin:locMin, LocMax:locMax)
{
    FLaboratory.begin ();
    DUsers.begin (Loc. LocMin, LocMax);
    RUsers.begin (NewLocation);
    PLaboratory.begin ();
}

```

End_Initialize;

Destruction

```

destroy ()
{
    FLaboratory.end();
    DUsers.end ();
    RUsers.end ();
    PLaboratory.end ();
}

```

End_Destruction;

End_Connector_Type LaboratoryCabinetUser;

3.3.1.17 Instanciación del Componente Usuario Laboratorio

En la configuración, el servicio *new* se invoca y da valor en la creación de la instancia, como se muestra a continuación:

```
new LaboratoryCabinetUser (newLocalhost, localhost, 198.120.0.0, 198.120.0.100);
```

3.3.1.18 Especificación de los aspectos del Conector Usuarios

Aspecto de coordinación

El aspecto de coordinación, involucra la sincronización de los servicios recibidos y enviados entre los elementos arquitectónicos que el conector coordina, misma que se muestra en el protocolo de este aspecto

El funcionamiento de los flujos de los servicios que participan en el protocolo puede observarse en la siguiente figura:

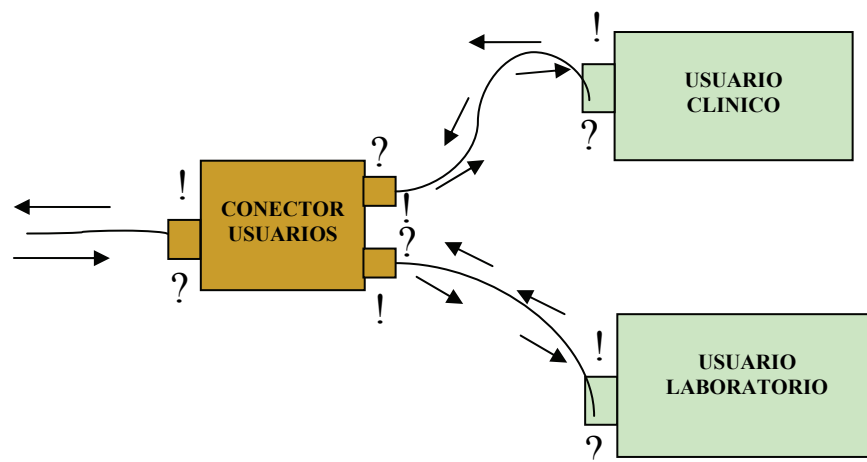


Fig. 47 Esquema que muestra el flujo de los servicios a través de los puertos del Conector Usuarios

El protocolo del comportamiento del conector DIAGNÓSTICO se especifica de la siguiente manera:

```
CUSERS = begin.COORDU;
```

```
COORDU = IntroducirPassword.P0
```

```
P0 =IntroducirSeleccion.OPTION
```

```

OPTION =
  RegistrarPaciente.P0
+
  VisualizarPacienteClin.P0
+
  VisualizarPacienteLab.P0
+
  VisualizarResultados.P0
+
  RealizarDiagnosticoClinico.P0
+
  RealizarDiagnosticoLaboratorio.P0
+
  Finalizar;

```

Este protocolo se describe en las tablas y en el grafo de transición que a continuación se muestran .

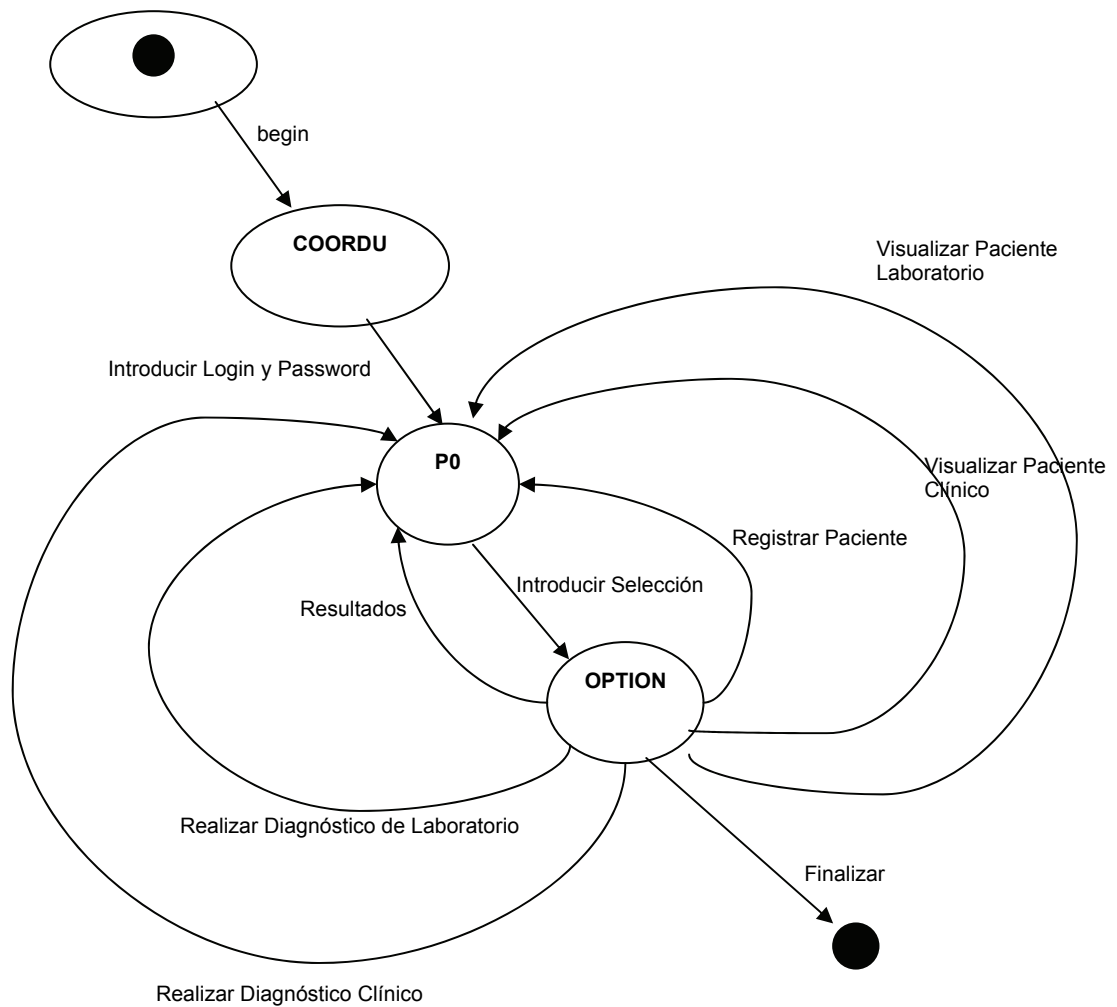


Fig. 48 Grafo de transición del protocolo del conector USUARIOS

El conector USUARIOS inicia su ejecución cuando el sistema obtiene el nombre de usuario y la contraseña del usuario.

puerto del que recibe el servicio (?)	puerto del que reenvía el servicio (i)	nombre del servicio
UserDiagPortCnct	UserClinPortCnct UserLabPortCnct	introducirLoginPassword (output Login:string, output Password:string)
UserClinPortCnct UserLabPortCnct	UserDiagPortCnct	introducirLoginPassword (output Login:string, output Password:string)

Acto seguido se obtiene del usuario la opción que realizará el sistema:

puerto del que recibe el servicio (?)	puerto del que reenvía el servicio (i)	nombre del servicio
UserDiagPortCnct	UserClinPortCnct UserLabPortCnct	introducirSeleccion (output Seleccion:int)
UserClinPortCnct UserLabPortCnct	UserDiagPortCnct	introducirSeleccion (output Seleccion:int)

- Si se ha elegido la opción de “Registrar Paciente”:

puerto del que recibe el servicio (?)	puerto del que reenvía el servicio (i)	nombre del servicio
UserDiagPortCnct	UserClinPortCnct	introducirPaciente (output IDPaciente:identifier, output DNI:string, output Nombre:string, output Dirección:string, output Tel:string, output Email:string, output Sexo:char, output FechaNac:date).
UserClinPortCnct	UserDiagPortCnct	introducirPaciente (output IDPaciente:identifier, output DNI:string, output Nombre:string, output Dirección:string, output Tel:string, output Email:string, output Sexo:char, output FechaNac:date).

- Si se ha seleccionado “Visualizar los datos del Paciente”:

puerto del que recibe el servicio (?)	puerto del que reenvía el servicio (i)	nombre del servicio
UserDiagPortCnct	UserClinPortCnct UserLabPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserClinPortCnct UserLabPortCnct	UserDiagPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserDiagPortCnct	UserClinPortCnct UserLabPortCnct	tomarPaciente (input DNI:string, input Nombre:string, input Dirección:string, input Tel:string, input Email:string, input Sexo:char, input FechaNac:date)

- Si se desea “Realizar el diagnóstico clínico”:

puerto del que recibe el servicio (?)	puerto del que reenvía el servicio (¡)	nombre del servicio
UserDiagPortCnct	UserClinPortCnct	introducirUsuarioClin (output UsuarioClin:string)
UserClinPortCnct	UserDiagPortCnct	introducirUsuarioClin (output UsuarioClin:string)
UserDiagPortCnct	UserClinPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserClinPortCnct	UserDiagPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserDiagPortCnct	UserClinPortCnct	obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool, output DolorCrecimientoParotidas:bool) ^ obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)
UserClinPortCnct	UserDiagPortCnct	obtenerSsggParotiditis (output Fiebre:bool, output DolorMasticacion:bool, output DolorCrecimientoParotidas:bool) ^ obtenerSsggIra (output Tos:bool, output DificultadRespiratoria:bool)
UserDiagPortCnct	UserClinPortCnct	(obtenerSsgfPaperas (output FiebreContinua:bool, output FiebreMayor38:bool, output DolorCrecimientoParotidas:bool, output DolorMasticacionEspontaneoAgudo:bool, output DolorMasticacionConBorramientoAnguloMandibular:bool) ^ obtenerSsgfParotiditisBacteriana (output FiebreMayor37:bool, output Odinofagia2A3Dias:bool, output ArdorGargantaIntenso:bool)) + (obtenerSsgfBronquiolitis (output FiebreContinua:bool, output FiebreMayor38:bool, output Fiebre2A3Dias:bool, output TosSeca:bool, output TosPorAccesos:bool, output DificultadRespiratoriaGrave:bool, output OdinofagiaGargantaRoja:bool, output Odinofagia2A3Dias:bool, output ApneaPeriodosLargos:bool, output ArdorGargantaIntenso:bool) ^ obtenerSsgfNeumonia (output FiebreMayor39:bool, output TosConFlema:bool, output TosFrecuente:bool, output DificultadRespiratoriaGravePolipneaTiraje:bool, output HipoventilacionCrepitantePulmonar:bool) ^ obtenerSsgfCrup (output RinoreaTransparente:bool , output DificultadRespiratoriaGarveAModerada:bool, output EstrindorLaringeo:bool))
UserClinPortCnct	UserDiagPortCnct	(obtenerSsgfPaperas (output FiebreContinua:bool, output FiebreMayor38:bool, output DolorCrecimientoParotidas:bool, output DolorMasticacionEspontaneoAgudo:bool, output DolorMasticacionConBorramientoAnguloMandibular:bool) ^ obtenerSsgfParotiditisBacteriana (output FiebreMayor37:bool, output Odinofagia2A3Dias:bool, output ArdorGargantaIntenso:bool)) + (obtenerSsgfBronquiolitis (output FiebreContinua:bool, output FiebreMayor38:bool, output Fiebre2A3Dias:bool, output TosSeca:bool, output TosPorAccesos:bool, output DificultadRespiratoriaGrave:bool, output OdinofagiaGargantaRoja:bool, output Odinofagia2A3Dias:bool, output ApneaPeriodosLargos:bool, output ArdorGargantaIntenso:bool) ^ obtenerSsgfNeumonia (output FiebreMayor39:bool, output TosConFlema:bool, output TosFrecuente:bool, output DificultadRespiratoriaGravePolipneaTiraje:bool, output HipoventilacionCrepitantePulmonar:bool) ^ obtenerSsgfCrup (output RinoreaTransparente:bool , output DificultadRespiratoriaGarveAModerada:bool, output EstrindorLaringeo:bool))
UserDiagPortCnct	UserClinPortCnct	tomarSugerencia (input Sugerencia:string)

- Si la selección fue “Adquirir el diagnostico de laboratorio”:

puerto del que recibe el servicio (?)	puerto del que reenvía el servicio (;)	nombre del servicio
UserDiagPortCnct	UserLabPortCnct	introducirUsuarioLab (output UsuarioLab:string)
UserLabPortCnct	UserDiagPortCnct	introducirUsuarioLab (output UsuarioLab:string)
UserDiagPortCnct	UserLabPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserLabPortCnct	UserDiagPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserDiagPortCnct	UserLabPortCnct	tomarSugLab (input SugLab:string)
UserDiagPortCnct	UserLabPortCnct	(obtenerEstsLabPaperas (output BiometriaHematicaPositiva:bool) ^ obtenerEstsLabParotidisBacteriana (output BiometriaHematicaLeucositosis:bool)) + (obtenerEstsLabBronquiolitis (output BiometriaHematicaLeucositosis:bool) ^ obtenerEstsLabNeumonia (output ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsqu ericiaColi:bool) ^ obtenerEstsLabCrup (output BiometriaHematicaEusino ^{filia} :bool))
UserLabPortCnct	UserDiagPortCnct	(obtenerEstsLabPaperas (output BiometriaHematicaPositiva:bool) ^ obtenerEstsLabParotidisBacteriana (output BiometriaHematicaLeucositosis:bool)) + (obtenerEstsLabBronquiolitis (output BiometriaHematicaLeucositosis:bool) ^ obtenerEstsLabNeumonia (output ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsqu ericiaColi:bool) ^ obtenerEstsLabCrup (output BiometriaHematicaEusino ^{filia} :bool))

- Para el caso de “Visualizar resultados del diagnóstico”:

puerto del que recibe el servicio (?)	puerto del que reenvía el servicio (;)	nombre del servicio
UserDiagPortCnct	UserClinPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserClinPortCnct	UserDiagPortCnct	introducirIDPaciente (output IDPaciente:identifier)
UserDiagPortCnct	UserClinPortCnct	tomarResultados (input EnfermedadClin:string, input CertClin:real, input EnfermedadLab:string, input CertLab:real, input Terapia:string, input UsuarioClin:string, input UsuarioLab:string)

Es conveniente realizar un análisis de los puertos del conector USUARIOS, de acuerdo a la siguiente tabla y figura:

elemento arquitectónico del que recibe/reenvía el servicio	puerto del conector	played_role del puerto	interfaz
usuario clinico	UserClinPortCnct	USERCLIN	IClinical
usuario laboratorio	UserLabPortCnct	USERLAB	ILaboratory
sistema diagnosticador	UserDiagPortCnct	USERDIAG	IDiagnostical

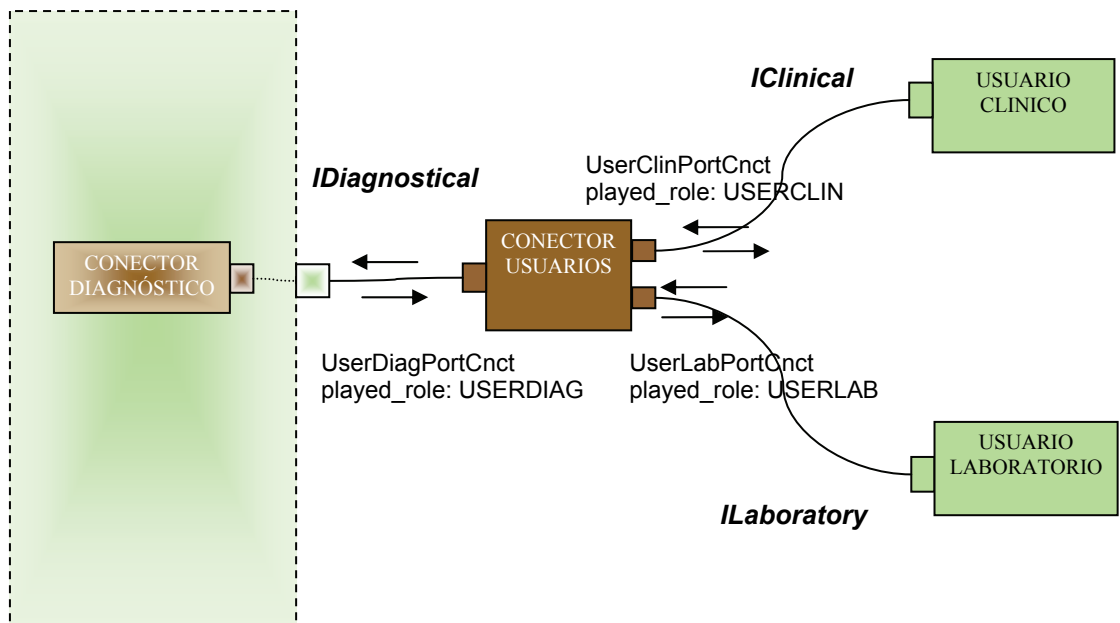


Fig. 49 Puertos del conector USUARIOS

Los servicios antes expuestos se publican en los puertos del conector denominados *UserDiagPortCnct*, *UserClinPortCnct* y *UserLabPortCnct* que tienen asociados como signatura las interfaces *IDiagnostical*, *IClinical* e *ILaboratory* respectivamente, y que dan semántica a su aspecto de coordinación.

La interfaz *IDiagnostical* del puerto de este conector, es reutilizada en el puerto del conector DIAGNÓSTICO del modelo arquitectónico propuesto. Por ello no se realizará la especificación en este apartado, ya que anteriormente fue especificada en el mencionado elemento arquitectónico. Similarmente, la especificación de las interfaces *IClinical* e

ILaboratory se muestran en los apartados de los componentes Usuario Clinico y Usuario Laboratorio.

Como resultado de este análisis, la especificación del aspecto de coordinación del conector USUARIOS es el siguiente:

Coordination Aspect CUsers **using** IClinical, ILaboratory, IDiagnostical

Services

begin ();

in/out introducirLoginPassword (**output** Login:string, **output** Password:string);

in/out introducirSeleccion (**output** Selección:int);

in/out introducirPaciente (**output** IDPaciente:identifier , **output** DNI:string, **output** Nombre:string, **output** Direccion:string, **output** Tel:string, **output** Email:string, **output** Sexo:char, **output** FechaNac:date);

in/out introducirIDPaciente (**output** IDPaciente:identifier);

in/out obtenerSsggParotiditis (**output** Fiebre:bool, **output** DolorMasticacion:bool, **output** DolorCrecimientoParotidas:bool);

in/out obtenerSsggIra (**output** Tos:bool, **output** DificultadRespiratoria:bool);

in/out obtenerSsgfPaperas (**output** FiebreContinua:bool, **output** FiebreMayor38:bool, **output** DolorCrecimientoParotidas:bool, **output** DolorMasticacionEspontaneoAgudo:bool, **output** DolorMasticacionConBorramientoAnguloMandibular:bool);

in/out obtenerSsgfParotiditisBacteriana (**output** FiebreMayor37:bool, **output** Odinofagia2A3Dias:bool, **output** ArdorGargantaIntenso:bool);

in/out obtenerSsgfBronquiolitis (**output** FiebreContinua:bool, **output** FiebreMayor38:bool, **output** Fiebre2A3Dias:bool, **output** TosSeca:bool, **output** TosPorAccesos:bool, **output** DificultadRespiratoriaGrave:bool, **output** OdinofagiaGargantaRoja:bool, **output** Odinofagia2A3Dias:bool, **output** ApneaPeriodosLargos:bool, **output** ArdorGargantaIntenso:bool);

in/out obtenerSsgfNeumonia (**output** FiebreMayor39:bool, **output** TosConFlema:bool, **output** TosFrecuente:bool, **output** DificultadRespiratoriaGravePolipneaTiraje:bool, **output** HipoventilacionCrepitantePulmonar:bool);

in/out obtenerSsgfCrup (**output** RinoreaTransparente:bool **output** DificultadRespiratoriaGraveAModerada:bool, **output** EstrindorLaringeo:bool) ;

in/out introducirUsuarioClin (**input** UsuarioClin:string);

in/out introducirUsuarioClin (**output** UsuarioClin:string);

in/out obtenerEstsLabPaperas (**output** BiometriaHematicaPositiva:bool);

in/out obtenerEstsLabParotidisBacteriana (**output** BiometriaHematicaLeucositosis:bool);

in/out obtenerEstsLabBronquiolitis (**output** BiometriaHematicaLeucositos:bool);

```

in/out obtenerEstsLabNeumonia (output
    ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi:bool);

in/out obtenerEstsLabCrup (output BiometriaHematicaEusinofilia:bool);

in/out introducirUsuarioLab (output UsuarioLab:string);

in/out tomarPaciente (output DNI:string, output Nombre:string, output Direccion:string,
output Tel:string, output Email:string, output Sexo:char, output FechaNac:date);

in/out tomarResultados (output EnfermedadClin:string, output CertClin:real,
output EnfermedadLab:string, output CertLab:real, output Terapia,
output UsuarioClin:string, output UsuarioLab:string)

in/out tomarSugLab (output SugLab:string) ;

in/out tomarSugerencia (output Sugerencia:string);

end;

```

Played_Roles

```

USERCLIN=
    (IClinical.IntroducirLoginPassword ! (Login, Password )
    →
    IClinical.IntroducirLoginPassword ? (Login, Password)
    →
    (IClinical.IntroducirSeleccion ! (Seleccion )
    →
    IClinical.IntroducirSeleccion ? (Seleccion)
    →
    (
    /*INTRODUCIR PACIENTE*/
    (IClinical.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo,
    FechaNac )
    →
    IClinical.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email, Sexo,
    FechaNac) )

+
    /*VISUALIZAR PACIENTE*/
    (IClinical.IntroducirIDPaciente ! (IDPaciente )
    →
    IClinical.Introducir IDPaciente ? (IDPaciente) )
    →
    IClinical.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac) )

+
    /*VISUALIZAR RESULTADOS*/
    ( IClinical.IntroducirIDPaciente ! (IDPaciente )
    →
    IClinical.Introducir IDPaciente ? (IDPaciente)
    →
    IClinical.TomarResultados ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab,
    Terapia, UsuarioClin, UsuarioLab) ).

+
    /*REALIZAR DIAGNOSTICO CLINICO*/
    ( IClinical.IntroducirUsuarioClin ! (UsuarioClin )
    →
    IClinical.IntroducirUsuarioClin ?! (UsuarioClin) )
    →
    IClinical.IntroducirIDPaciente ! (IDPaciente )

```

→
 IClinical.Introducir IDPaciente ? (IDPaciente)
 →
 (IClinical.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 IClinical.ObtenerSsggIra ! (Tos, DificultadRespiratoria))
 →
 (IClinical.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 IClinical.ObtenerSsggIra ? (Tos, DificultadRespiratoria))
 →
 (
 (IClinical.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
 DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
 DolorMasticacionConBorramientoAnguloMandibular)
 ^
 IClinical.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
 ArdorGargantaIntenso))
 +
 (IClinical.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38, Fiebre2A3Dias,
 TosSeca, TosPorAccesos, DificultadRespiratoriaGrave, OdinofagiaGargantaRoja,
 Odinofagia2A3Dias, ApneaPeriodosLargos, ArdorGargantaIntenso)]
 ^
 IClinical.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
 DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
 ^
 IClinical.ObtenerSsgfCrup ! (RinoreaTransparente,
 DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))
)
 →
 (
 (IClinical.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
 DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
 DolorMasticacionConBorramientoAnguloMandibular)
 ^
 IClinical.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
 ArdorGargantaIntenso))
 +
 (IClinical.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
 Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
 OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
 ArdorGargantaIntenso)]
 ^
 IClinical.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
 DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
 ^
 IClinical.ObtenerSsgfCrup ? (RinoreaTransparente,
 DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))
)
 →
 IClinical.TomarSugerencia ! (Sugerencia))

USERLAB=

(ILaboratory.IntroducirLoginPassword ! (Login, Password)
 →
 ILaboratory.IntroducirLoginPassword ? (Login, Password)
 →
 (ILaboratory.IntroducirSeleccion ! (Seleccion))

```

→
ILaboratory.IntroducirSeleccion ? (Seleccion)
→
(
                                                                    /*VISUALIZAR PACIENTE*/
(ILaboratory.IntroducirIDPaciente ! (IDPaciente )
→
ILaboratory.Introducir IDPaciente ? (IDPaciente)
→
ILaboratory.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac) )
+
                                                                    /*REALIZAR DIAGNOSTICO LABORATORIO*/
(ILaboratory.IntroducirUsuarioLab ! (UsuarioLab )
→
ILaboratory.IntroducirUsuarioLab ? (UsuarioLab)
→
ILaboratory.IntroducirIDPaciente ! (IDPaciente )
→
ILaboratory.Introducir IDPaciente ? (IDPaciente)
→
ILaboratory.TomarSugLab ! (SugLab)
→
(
( ILaboratory.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
ILaboratory.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
(ILaboratory.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
^
ILaboratory.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
^
ILaboratory.ObtenerEstsLabCrup ! (BiometriaHematicaEusinoFilia) )
)
→
(
( ILaboratory.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
ILaboratory.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
(ILaboratory.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
^
ILaboratory.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
^
ILaboratory.ObtenerEstsLabCrup ? (BiometriaHematicaEusinoFilia) )
)

USERDIAG=
(IDiagnostical.IntroducirLoginPassword ? (Login, Password )
→
IDiagnostical.IntroducirLoginPassword ! (Login, Password)
→
(IDiagnostical.IntroducirSeleccion ! (Seleccion )
→
IDiagnostical.IntroducirSeleccion ? (Seleccion)
→
(
                                                                    /*INTRODUCIR PACIENTE*/

```

```

(IDiagnostical.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac )
→
IDiagnostical.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
Sexo, FechaNac) )

+
/*VISUALIZAR PACIENTE- MEDICO*/
(IDiagnostical.IntroducirIDPaciente ? (IDPaciente )
→
IDiagnostical.IntroducirIDPaciente ! (IDPaciente)
→
IDiagnostical.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))

+
/*VISUALIZAR PACIENTE- LABORATORISTA*/
(IDiagnostical.IntroducirIDPaciente ? (IDPaciente )
→
IDiagnostical.IntroducirIDPaciente ! (IDPaciente)
→
IDiagnostical.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))

+
/*VISUALIZAR RESULTADOS*/
( IDiagnostical.IntroducirIDPaciente? (IDPaciente )
→
IDiagnostical.IntroducirIDPaciente ! (IDPaciente)
→
IDiagnostical.TomarResultados ? (EnfermedadClin, CertClin, EnfermedadLab,
CertLab, Terapia, UsuarioClin, UsuarioLab) ).

+
/*REALIZAR DIAGNOSTICO CLINICO*/
(IDiagnostical.IntroducirUsuarioClin ? (UsuarioClin )
→
IDiagnostical.IntroducirUsuarioClin ! (UsuarioClin)
→
IDiagnostical.IntroducirIDPaciente ? (IDPaciente )
→
IDiagnostical.IntroducirIDPaciente ! (IDPaciente) )
→
(IDiagnostical.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas )
^
IDiagnostical.ObtenerSsggIra ? (Tos, DificultadRespiratoria) )
→
(IDiagnostical.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
DolorCrecimientoParotidas)
^
IDiagnostical.ObtenerSsggIra ! (Tos, DificultadRespiratoria) )
→
(
( IDiagnostical.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IDiagnostical.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odirofagia2A3Dias,
ArdorGargantaIntenso) )
+
(IDiagnostical.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdirofagiaGargantaRoja, Odirofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^

```

```

IDiagnostical.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
^
IDiagnostical.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(
( IDiagnostical.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
IDiagnostical.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(IDiagnostical.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
IDiagnostical.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
^
IDiagnostical.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
IDiagnostical.TomarSugerencia ? (Sugerencia)

+
/*REALIZAR DIAGNOSTICO LABORATORIO*/
(IDiagnostical.IntroducirUsuarioLab ? (UsuarioLab)
→
IDiagnostical.IntroducirUsuarioLab ! (UsuarioLab) )
→
IDiagnostical.IntroducirIDPaciente ? (IDPaciente)
→
IDiagnostical.IntroducirIDPaciente ! (IDPaciente)
→
IDiagnostical.TomarSugLab ? (SugLab)
→
(
( IDiagnostical.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
^
IDiagnostical.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
(IDiagnostical.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
^
IDiagnostical.ObtenerEstsLabNeumonia ?
(ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
^
IDiagnostical.ObtenerEstsLabCrup ? (BiometriaHematicaEusinofilia) )
)
→
(
( IDiagnostical.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
^
IDiagnostical.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
(IDiagnostical.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
^

```

```

IDiagnostical.ObtenerEstsLabNeumonia !
(ExuradoFaringeoEstreptococoNeumonicHemofilusInfluenzaEsquericiaColi)
^
IDiagnostical.ObtenerEstsLabCrup ! (BiometriaHematicaEusinofilia) )
)

```

Protocols

```

CUSERS = begin.COORDU;

COORDU =
  USERDIAG.IntroducirLoginPassword ? (Login, Password )
  →
  ( USERCLIN.IntroducirLoginPassword ! (Login, Password )
  ||
  USERLAB.IntroducirLoginPassword ! (Login, Password ) )
  →
  ( USERCLIN.IntroducirLoginPassword ? (Login, Password)
  +
  USERLAB.IntroducirLoginPassword ? (Login, Password ) ).P0

P0 =
  USERDIAG.IntroducirSeleccion ? (Seleccion )
  →
  ( USERCLIN.IntroducirSeleccion! (Seleccion )
  ||
  USERLAB.IntroducirSeleccion! (Seleccion ) )
  →
  ( USERCLIN.IntroducirSeleccion ? (Seleccion)
  +
  USERLAB.IntroducirSeleccion ? (Seleccion) ).OPTION

OPTION =
  RegistrarPaciente
+
  (VisualizarPacienteClin
+
  VisualizarPacienteLab)
+
  VisualizarResultados
+
  RealizarDiagnosticoClinico
+
  RealizarDiagnosticoLaboratorio
+
  Finalizar;

RegistrarPaciente =
  (USERDIAG.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
  Sexo, FechaNac )
  →
  USERCLIN.IntroducirPaciente ! (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
  Sexo, FechaNac )
  →
  USERCLIN.IntroducirPaciente ? (IDPaciente, DNI, Nombre, Direccion, Tel, Email,
  Sexo, FechaNac)
  →

```

USERDIAG.IntroducirPaciente ! (IDPaciente; DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac)). P0

VisualizarPacienteClin =

(USERDIAG.IntroducirIDPaciente ? (IDPaciente))
→
USERCLIN.IntroducirIDPaciente ! (IDPaciente))
→
USERCLIN.IntroducirIDPaciente ? (IDPaciente))
→
USERDIAG.IntroducirIDPaciente ! (IDPaciente))
→
USERDIAG.TomarPaciente ? (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))
→
USERCLIN.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))
). P0

VisualizarPacienteLab =

(USERDIAG.IntroducirIDPaciente ? (IDPaciente))
→
USERLAB.IntroducirIDPaciente ! (IDPaciente))
→
USERLAB. IntroducirIDPaciente ? (IDPaciente))
→
USERDIAG. IntroducirIDPaciente ! (IDPaciente))
→
USERDIAG.TomarPaciente ? (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))
→
USERLAB.TomarPaciente ! (DNI, Nombre, Direccion, Tel, Email, Sexo, FechaNac))
).P0

VisualizarResultados =

(USERDIAG.IntroducirIDPaciente ? (IDPaciente))
→
USERCLIN.IntroducirIDPaciente ! (IDPaciente))
→
USERCLIN.IntroducirIDPaciente ? (IDPaciente))
→
USERDIAG.IntroducirIDPaciente ! (IDPaciente))
→
USERDIAG.TomarResultados ? (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab))
→
USERCLIN.TomarResultados ! (EnfermedadClin, CertClin, EnfermedadLab, CertLab, Terapia, UsuarioClin, UsuarioLab)). P0

RealizarDiagnosticoClinico =

(USERDIAG.IntroducirIDPaciente ? (IDPaciente))
→
USERCLIN.IntroducirIDPaciente ! (IDPaciente))
→
USERCLIN.IntroducirIDPaciente ? (IDPaciente))
→
USERDIAG.IntroducirIDPaciente ! (IDPaciente))
→
USERDIAG.IntroducirUsuarioClin ? (UsuarioClin)

→
 USERCLIN.IntroducirUsuarioClin ! (UsuarioClin)
 →
 USERCLIN.IntroducirUsuarioClin ? (UsuarioClin)
 →
 USERDIAG.IntroducirUsuarioClin ! (UsuarioClin)
 →
 (
 (USERDIAG.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 USERDIAG.ObtenerSsggIra ? (Tos, DificultadRespiratoria))
 →
 (USERCLIN.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 USERCLIN.ObtenerSsggIra ! (Tos, DificultadRespiratoria))
 →
 (USERCLIN.ObtenerSsggParotiditis ? (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 USERCLIN.ObtenerSsggIra ? (Tos, DificultadRespiratoria))
 →
 (USERDIAG.ObtenerSsggParotiditis ! (Fiebre, DolorMasticacion,
 DolorCrecimientoParotidas)
 ^
 USERDIAG.ObtenerSsggIra ! (Tos, DificultadRespiratoria))
)
 →
 (
 (USERDIAG.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
 DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
 DolorMasticacionConBorramientoAnguloMandibular)
 ^
 USERDIAG.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
 ArdorGargantaIntenso))
 +
 (USERDIAG.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
 Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
 OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
 ArdorGargantaIntenso)]
 ^
 USERDIAG.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
 DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
 ^
 USERDIAG.ObtenerSsgfCrup ? (RinoreaTransparente,
 DificultadRespiratoriaGraveAModerada, EstrindorLaringeo))
 →
 (USERCLIN.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
 DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
 DolorMasticacionConBorramientoAnguloMandibular)
 ^
 USERCLIN.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
 ArdorGargantaIntenso))
 +
 (USERCLIN.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38,
 Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
 OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
 ArdorGargantaIntenso)]
 ^

```

USERCLIN.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
^
USERCLIN.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
( USERCLIN.ObtenerSsgfPaperas ? (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
USERCLIN.ObtenerSsgfParotiditisBacteriana ? (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(USERCLIN.ObtenerSsgfBronquiolitis ? (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
USERCLIN.ObtenerSsgfNeumonia ? (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)]
^
USERCLIN.ObtenerSsgfCrup ? (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
(
USERDIAG.ObtenerSsgfPaperas ! (FiebreContinua, FiebreMayor38,
DolorCrecimientoParotidas, DolorMasticacionEspontaneoAgudo,
DolorMasticacionConBorramientoAnguloMandibular)
^
USERDIAG.ObtenerSsgfParotiditisBacteriana ! (FiebreMayor37, Odinofagia2A3Dias,
ArdorGargantaIntenso) )
+
(USERDIAG.ObtenerSsgfBronquiolitis ! (FiebreContinua, FiebreMayor38,
Fiebre2A3Dias, TosSeca, TosPorAccesos, DificultadRespiratoriaGrave,
OdinofagiaGargantaRoja, Odinofagia2A3Dias, ApneaPeriodosLargos,
ArdorGargantaIntenso) ]
^
USERDIAG.ObtenerSsgfNeumonia ! (FiebreMayor39, TosConFlema, TosFrecuente,
DificultadRespiratoriaGravePolipneaTiraje, HipoventilacionCrepitantePulmonar)
^
USERDIAG.ObtenerSsgfCrup ! (RinoreaTransparente,
DificultadRespiratoriaGraveAModerada, EstrindorLaringeo) )
)
→
USERDIAG.TomarSugerencia ? (Sugerencia )
→
USERCLIN.TomarSugerencia ! (Sugerencia) ).P0

```

```

RealizarDiagnosticoLaboratorio =
( USERDIAG.IntroducirUsuarioLab ? (UsuarioLab)
→
USERLAB.IntroducirUsuarioLab ! (UsuarioLab)
→
USERLAB.IntroducirUsuarioLab ? (UsuarioLab)
→
USERDIAG.IntroducirUsuarioLab ! (UsuarioLab)
→
USERDIAG.IntroducirIDPaciente ? (IDPaciente)

```

```

→
USERLAB.IntroducirIDPaciente ! (IDPaciente )
→
USERLAB.IntroducirIDPaciente ? (IDPaciente)
→
USERDIAG.IntroducirIDPaciente ! (IDPaciente)
→
USERDIAG.TomarSugLab ? (SugLab )
→
USERLAB.TomarSugLab ! (SugLab )
→
(
(
  USERDIAG.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
  ^
  USERDIAG.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
  (USERDIAG.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
  ^
  USERDIAG.ObtenerEstsLabNeumonia ?
  (ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
  ^
  USERDIAG.ObtenerEstsLabCrup ? (BiometriaHematicaEusino filia) )
)
→
(
(
  USERLAB.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
  ^
  USERLAB.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
  (USERLAB.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
  ^
  USERLAB.ObtenerEstsLabNeumonia !
  (ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
  ^
  USERLAB.ObtenerEstsLabCrup ! (BiometriaHematicaEusino filia) )
)
→
(
(
  USERLAB.ObtenerEstsLabPaperas ? (BiometriaHematicaPositiva)
  ^
  USERLAB.ObtenerEstsLabParotidisBacteriana ? (BiometriaHematicaLeucositosis) )
+
  (USERLAB.ObtenerEstsLabBronquiolitis ? (BiometriaHematicaLeucositos)
  ^
  USERLAB.ObtenerEstsLabNeumonia ?
  (ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
  ^
  USERLAB.ObtenerEstsLabCrup ? (BiometriaHematicaEusino filia) )
)
→
(
(
  USERDIAG.ObtenerEstsLabPaperas ! (BiometriaHematicaPositiva)
  ^
  USERDIAG.ObtenerEstsLabParotidisBacteriana ! (BiometriaHematicaLeucositosis) )
+
  (USERDIAG.ObtenerEstsLabBronquiolitis ! (BiometriaHematicaLeucositos)
  ^
  USERDIAG.ObtenerEstsLabNeumonia !
  (ExuradoFaringeoEstreptococoNeumoniceHemofilusInfluenzaEsquericiaColi)
  ^
  USERDIAG.ObtenerEstsLabCrup ! (BiometriaHematicaEusino filia) )
)

```

```
)  
) ).PO
```

```
Finalizar =  
    end;
```

End_Coordination Aspect CUsers

Aspecto de distribución

Este aspecto va a especificar la ubicación de la máquina desde la cual el usuario manipula al sistema, permitiendo que se pueda utilizar el sistema desde distintas máquinas. Por lo que el aspecto de distribución del conector USUARIOS va a ser dinámico y va a proveer servicios de movilidad.

Debido a que este aspecto requiere de un atributo para la ubicación de la máquina desde la que se manipula al sistema, el servicio *begin* tiene la siguiente sintaxis:

```
begin (input Location:loc, LocMax:loc, LocMin:loc),
```

que contempla un rango permitido con la condición {Location>locMin **and** Location<locMax}, lo que implica que con el servicio *checkLocation* se comprueba que se está dentro de los límites permitidos (Check:= true).

```
checkLocation (input Location:loc, output Check:bool)
```

Con el servicio *move* obtiene su nueva localización en la máquina, la cual debe de estar dentro del rango permitido ({NewLocation > locMin **and** NewLocation <locMax}).

```
move (input NewLocation:loc)
```

Una característica este aspecto de distribución es que, aunque cuenta con servicios, éstos son internos y no públicos, pues sólo son utilizados internamente y no para la comunicación con otros elementos arquitectónicos. Por ello, este aspecto no usa ninguna interfaz ni contiene played_roles.

La especificación de este aspecto se muestra en el apartado del aspecto de distribución del componente USUARIO CLÍNICO, ya que ambos elementos arquitectónicos importan el mismo aspecto de distribución.

3.3.1.19 Especificación del Conector Usuarios

Tras haberse especificado los aspectos necesarios para el conector USUARIOS, a continuación se va a definir el *Connector_Type UsersCnct*, el cual será un tipo, de forma tal que el tipo *UsersCnct* será instanciado para crear el conector USUARIOS que se ha diseñado en el modelo arquitectónico.

El conector *UsersCnct* especifica el conjunto de los tres puertos necesarios para comunicar a los elementos arquitectónicos que une. Para definir cada puerto, se detallan el nombre (*UserDiagPortCnct*, *UserClinPortCnct* y *UserLabPortCnct*), el tipo que corresponde a la interfaz declarada (*IDiagnostical*, *IClinical* e *ILaboratory*) y el *played_role* (*UserDiag*, *UserClin* y *UserLab*) que especifica el comportamiento del conjunto de servicios que forman la interfaz que tipa al puerto. Los *played_role* se encuentran definidos en el aspecto de coordinación (*CUsers*) que importa el conector *UsersCnct*:

```
UserDiagPortCnct: IDiagnostical,  
    Played_Role CUsers.UserDiag;  
UserClinPortCnct: IClinical,  
    Played_Role CUsers.UserClin;  
UserLabPortCnct: ILaboratory,  
    Played_Role . CUsers.UserLab;
```

El conector tipo *UsersCnct* importa dos aspectos, el de coordinación (*CUsers*) para sincronizar (i.e. coordinar y redireccionar) los servicios entre los componentes que une, y el de distribución (*DUsers*) con el objetivo de indicar su ubicación dentro del sistema.

```
Coordination Aspect import CUsers;  
Distribution Aspect import DUsers;
```

En la sección *Initialize* se especifica el inicio de los aspectos (*CUsers* y *DUsers*) que lo forman. Para crear la instancia es necesario dar un valor inicial a los atributos requeridos *location*, *locationMin* y *locationMax* de su aspecto de coordinación. Estos atributos tienen la propiedad de *not null*:

```
new (Loc: loc, LocMin:loc, LocMax:loc)  
{  
    CUsers.begin ();  
    DUsers.begin (Loc, LocMin, LocMax);  
}
```

En la sección *Destruction* se especifica el término de los aspectos (*CUsers* y *DUsers*) que lo forman:

```
destroy ()  
{  
    CUsers.end ();  
    DUsers.end ();  
}
```

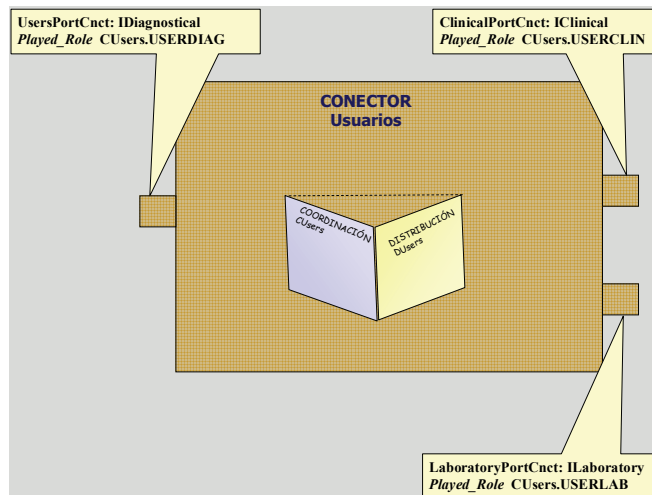


Fig. 50 Conector USUARIOS

Finalmente, y con base en el análisis realizado, la especificación del conector Usuarios muestra la siguiente sintaxis:

```

Connector_Type UserCnct
  Ports
    UserDiagPortCnct: IDiagnostical,
      Played_Role CUsers.USERDIAG;
    UserClinPortCnct: IClinical,
      Played_Role CUsers.USERCLIN;
    UserLabPortCnct: ILaboratory,
      Played_Role . CUsers.USERLAB;
  End_Ports;

  Coordination Aspect import CUsers;
  Distribution Aspect import DUsers;

  Initialize
    new (Loc:loc, LocMin:loc, LocMax:loc)
    {
      CUsers.begin ();
      DUsers.begin (Loc, LocMin, LocMax);
    }
  End_Initialize;

  Destruction
    destroy ()
    {
      CUsers.end();
      DUsers.end();
    }
  End_Destruction;

End_Connector_Type UserCnct;

```

3.3.1.20 Instanciación del Conector Usuarios

En la configuración, el servicio *new* se invoca y da valor en la creación de la instancia, como se muestra a continuación:

```
new UserCnct (newLocalhost, localhost, 198.120.0.0, 198.120.0.100);
```

3.3.2. Especificación del modelo arquitectónico y configuración final

3.3.2.1 Especificación del modelo arquitectónico final

Para acabar la arquitectura completa del sistema de diagnóstico médico, se realizará la definición del modelo arquitectónico final. El nivel de configuración del lenguaje permite definir las instancias y la topología del modelo arquitectónico final, como a continuación se describe:

La cabecera **Architectural_Model** define el nombre del modelo arquitectónico: `DiagMed`

La cláusula **Variables** define los componentes, conectores y sistemas que forman el modelo arquitectónico, que a la vez que importa los tipos de la librería PRISMA, las asocia con un nombre de variable para posteriormente poderles hacer referencia en las siguientes secciones:

```
VarUsuarioClinico: ClinicUser;  
VarUsuarioLaboratorio: LaboratoryCabinetUser;  
VarConectorUsuarios: UserCnct;  
VarSistemaDiagnosticador: DiagnosticalSystem;
```

La especificación de **Attachments** especifica, al igual que en los sistemas, la conexión entre los diferentes elementos arquitectónicos que conforman el modelo:

```
VarSistemaDiagnosticador.DiagnosticalPort ↔ VarConectorUsuarios.UserDiagPortCnct;  
VarConectorUsuarios.UserClinPortCnct ↔ VarUsuarioClinico.UserClinPort;  
VarConectorUsuarios.UserLabPortCnct ↔ VarUsuarioLaboratorio.UserLabPort;
```

El bloque **Initialize** especifica como instanciar los elementos arquitectónicos que conforman el modelo, proporcionando los valores de inicialización que requieren sus constructores. La aproximación consiste en definir el modelo arquitectónico como un tipo, es decir, no especificar los valores de inicialización que tomarán las instancias, tan sólo la secuencia de llamadas a los constructores de los elementos arquitectónicos que forman el modelo arquitectónico. Más tarde cuando se decidiese instanciar el modelo arquitectónico definido, se proporcionarían los valores de inicialización requeridos. Con esto se favorece la reutilización del modelo arquitectónico definido.

```
new (NewLocation:loc, Loc:loc, LocMin:loc, LocMax:loc)  
{  
  VarUsuarioClinico = new ClinicUser (NewLocation, Loc, LocMin, LocMax);  
  VarUsuarioLaboratorio = new LaboratoryCabinetUser (NewLocation, Loc, LocMin, LocMax);  
  VarConectorUsuarios = new UserCnct (Loc, LocMin, LocMax);  
  VarSistemaDiagnosticador = new DiagnosticalSystem (Location);  
}
```

El bloque de **Destruction** define qué secuencias de destrucción de los componentes debe seguirse:

```
destroy ()  
{  
  VarUsuarioClinico.destroy ();  
  VarUsuarioLaboratorio.destroy ();  
}
```

```

VarConectorUsuarios.destroy ( );
VarSistemaDiagnosticador.destroy ( );
}

```

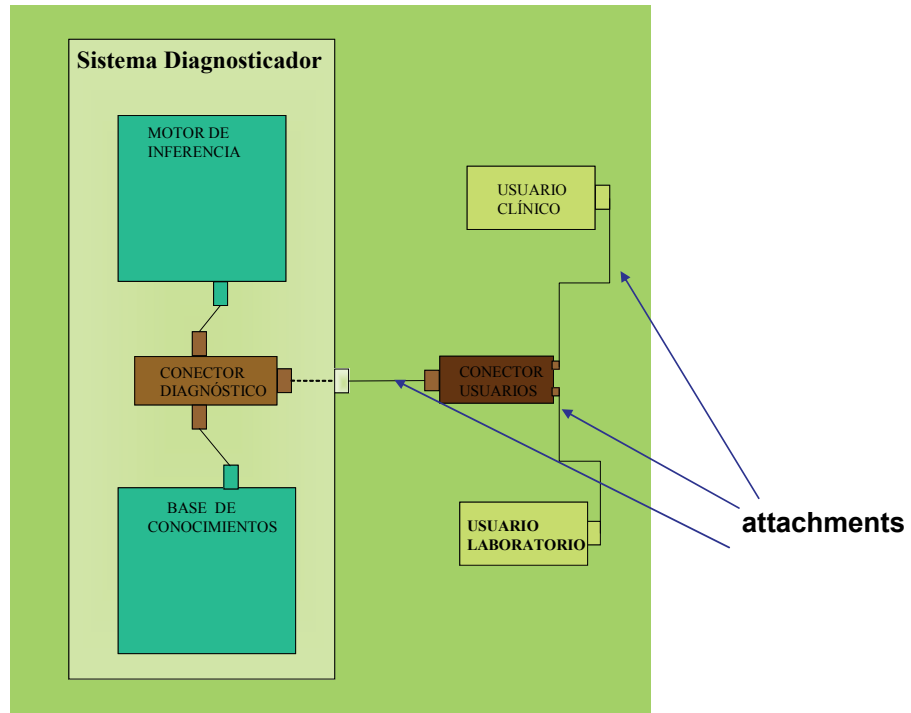


Fig. 51 Model Arquitectónico final del Sistema de Diagnóstico Médico *DiagMed*

Con base en lo anteriormente comentado, la especificación del modelo arquitectónico final presenta la siguiente sintaxis:

Architectural_Model DiagMed

Variables

```

VarUsuarioClinico: ClinicUser;
VarUsuarioLaboratorio: LaboratoryCabinetUser;
VarConectorUsuarios: UserCnct;
VarSistemaDiagnosticador: DiagnosticalSystem;

```

End_Variables;

Attachments

```

VarSistemaDiagnosticador.DiagnosticalPort ↔ VarConectorUsuarios.UserDiagPortCnct;
VarConectorUsuarios.UserClinPortCnct ↔ VarUsuarioClinico.UserClinPort;
VarConectorUsuarios.UserLabPortCnct ↔ VarUsuarioLaboratorio.UserLabPort;

```

End_Attachments;

Initialize

```

new (NewLocation:loc, Loc:loc, LocMin:loc, LocMax:loc)
{
VarUsuarioClinico = new ClinicUser (NewLocation, Loc, LocMin, LocMax);
VarUsuarioLaboratorio = new LaboratoryCabinetUser (NewLocation, Loc, LocMin, LocMax);
VarConectorUsuarios = new UserCnct (Loc, LocMin, LocMax);
VarSistemaDiagnosticador = new DiagnosticalSystem (Location);
}

```

End_Initialize;

Destruction

```

destroy ()
{
  VarUsuarioClinico.destroy ();
  VarUsuarioLaboratorio.destroy ();
  VarConectorUsuarios.destroy ();
  VarSistemaDiagnosticador.destroy ();
}
End_Destruction;

End_Architectural_Model DiagMed;

```

3.3.2.2 Instanciación de la configuración final

El bloque de inicialización únicamente define cómo construirlo, pero no lo instancia, con el objetivo de favorecer la reutilización del modelo arquitectónico. La instanciación del modelo arquitectónico se haría de la siguiente forma:

```

SISTEMA DIAGNÓSTICO MÉDICO = new DiagMed (newLocalhost, localhost, 0, 100)
{
  CLIN_OPERATOR = new ClinicUser (newLocalhost, localhost, 0, 100);
  LAB_OPERATOR = new LaboratoryCabinetUser (newLocalhost, localhost, 0, 100);
  CONNECTOR = new UserCnct (localhost, 0,100);
  EXPERT_SYSTEM = new DiagnosticalSystem (localhost )
  {
    INFERENCE = new InferenceMotor (localhost);
    DOMAIN= new KnowledgeBase (localhost);
    DIAG_CNCT= new DiagnosticCnct (localhost);
  }
}

```

Obsérvese cómo se han ido anidando los constructores para inicializar los componentes que forman el sistema. Se ha definido el constructor del sistema final con tantos argumentos como los que tenga cada uno de los constructores de los componentes/conectores que forman el sistema.

3.4 Ventajas de realizar la especificación en el LDA de PRISMA.

Con el fin de demostrar empíricamente que las propiedades de PRISMA se satisfacen, además de la especificación utilizando el LDA de PRISMA del sistema propuesto en este trabajo, se ha realizado la implementación de este sistema (como un típico sistema experto basado en reglas) codificado en el lenguaje PROLOG, y que se presenta en el apéndice D de este documento. En ambas aplicaciones se realizó el diagnóstico médico considerando el caso de estudio de las enfermedades infecciosas infantiles.

En este apartado, se mencionarán algunas de las ventajas que tiene el realizar la especificación en el LDA de PRISMA frente a la implementación realizada en PROLOG, y a continuación se comentarán algunas similitudes entre ambos lenguajes.

Las ventajas que aportan la especificación en el LDA de PRISMA para la construcción de sistemas, contemplan la mejora de las siguientes propiedades:

Complejidad.- mediante diferentes niveles de granularidad se ha gestionado la complejidad del sistema a través de la separación de unidades funcionales en elementos arquitectónicos.

Mantenimiento.- se favorece el mantenimiento del sistema al separar los *crosscutting concerns* en aspectos. Asimismo, los elementos arquitectónicos de este modelo son independientes, de tal forma que se pueden añadir o intercambiar componentes y/o conectores, sin afectar al funcionamiento general del sistema ni al resto de los elementos que lo conforman.

Reutilización.- tanto los aspectos como los elementos arquitectónicos son reutilizables, ya sea dentro de la misma arquitectura en dominios diferentes o bien en arquitecturas diferentes.

Distribución.- este sistema se puede utilizar de forma tanto centralizada como distribuida, ya que mediante el aspecto de distribución se especifican las características que definen la localización dinámica del elemento arquitectónico en el cual se integra, así como los servicios necesarios para implementar estrategias de distribución de los elementos arquitectónicos (como la movilidad) con el fin de optimizar la distribución de la topología del sistema resultante.

Independencia de plataforma.- el sistema es independiente de plataforma, lo que evita que los cambios de infraestructura sean una fuente de variabilidad ya que el alto nivel de abstracción de las especificaciones hace que éstas sean independientes de la tecnología aplicada. Ante un cambio tecnológico, la especificación no se ve afectada, solo el proceso de traducción de la especialización a código.

Generalidad.- las características de reutilización, distribución e independencia de plataforma, implican una generalidad en nuestra aproximación, permitiendo realizar

tareas de diagnóstico o tareas similares que comparten características comunes, como las de interpretación y predicción, aplicado a diversos dominios.

En PROLOG se observa que dichas propiedades tienen las siguientes características:

Complejidad.- no existen diferentes niveles de granularidad. Sin embargo aunque se realiza una separación de unidades funcionales en elementos arquitectónicos, solamente se contemplan componentes en los que la codificación se realiza en un único paquete de software, donde se definen los tipos de datos, las bases de datos y los predicados de todos los componentes de forma conjunta.

Mantenimiento.- debido a que la implementación del sistema se realiza en un solo paquete de software, el modificar alguno de sus componentes implica cambios en el código que repercuten de forma no favorable al mantenimiento del propio sistema.

Reutilización.- los componentes son reutilizables en dominios diferentes, aunque el código de la implementación de todo el sistema se encuentra en un sólo paquete. Adicionalmente, dado que no contiene aspectos, no se aprovecha la reutilización de estos “trozos de código”.

Distribución.- no se definen la localización de los componentes que integran el sistema, ni se implementan estrategias de distribución (como la movilidad) de los elementos arquitectónicos.

Independencia de plataforma.- el PROLOG es un lenguaje que necesita ser codificado sobre una plataforma específica, según la versión utilizada.

Generalidad.- la característica de reutilización se encuentra presente, de forma limitada (ya que es posible cambiar las reglas del dominio para que este sistema sea aplicado a otro caso de estudio dentro del diagnóstico médico), sin embargo las características de distribución e independencia de plataforma no se encuentran presentes. Por ello, no se considera una generalidad en la aproximación.

Algunas similitudes entre el lenguaje de programación PROLOG y la especificación en el LDA de PRISMA, se describen a continuación:

1.- En el LDA de PRISMA se especifican elementos arquitectónicos (componentes, conectores y sistemas) y aspectos. Cada uno especificados en paquetes de software independientes.

En PROLOG se codifican solo componentes, mismos que se encuentran en un solo paquete de software dentro de la sección *clauses* del programa.

2.- En PROLOG estos componentes están formados por reglas del tipo “IF <premisa> THEN <conclusión>”. Existen reglas del dominio (que forman la base de conocimientos del caso de estudio) y reglas de control.

En el LDA de PRISMA las reglas del dominio se corresponden con las *derivations* y las reglas de control con los *protocols*. Todas estas reglas se encuentran en los aspectos que conforman los elementos arquitectónicos del sistema.

3.- En PROLOG los predicados que se encuentran en la sección de *predicates* del programa, se corresponden con los servicios del LDA de PRISMA que se encuentran en la sección *services* de los aspectos.

4.- Los parámetros de los servicios del LDA de PRISMA se corresponden con los parámetros de los predicados de PROLOG.

5.- En el LDA de PRISMA las variables forman parte de los aspectos en la sección de *attributes*, los cuales pueden ser constantes, variables o derivados. Las variables pueden ser del tipo string, real, char, bool, integer, etc.

En PROLOG las variables se colocan en la sección *domains* del programa. Se incluyen todo tipo de variables sin distinción entre constantes, variables o derivadas. Las variables pueden ser del tipo string, real, char, bool, integer, etc.

6.- En el LDA de PRISMA las variables derivadas se especifican a través de *derivations* dentro del aspecto respectivo.

En PROLOG estas *derivations* son reglas que se encuentran en la sección *clauses* del programa.

7.- En el LDA de PRISMA el cambio de estado se realiza con los atributos variables a través de las *valuations*. En PROLOG para los cambios de estado se utilizan los predicados que manipulan bases de datos: *load*, *assert*, *asserta*, *assertz*, *retract*, *save*, *consult* y *reconsult*.

8.- Los *weavings* del LDA de PRISMA pueden considerarse como el encadenamiento de algunas de las reglas de control de PROLOG.

CONCLUSIONES Y TRABAJO FUTURO

CONCLUSIONES

En este trabajo se ha presentado el diseño y análisis de un sistema de diagnóstico, en el marco del modelo PRISMA, el cual integra las aproximaciones del desarrollo de software basado en componentes y orientado a aspectos.

Se ha especificado su arquitectura mediante el LDA de PRISMA, en el nivel de definición de tipos y en el nivel de configuración. Se han descrito los elementos arquitectónicos que forman el modelo, en tres niveles de abstracción (del más simple al más complejo). Se han tratado diversos aspectos (funcional, coordinación, distribución, replicación, presentación y persistencia); sin embargo han sido tratados con diferente detalle, debido a que algunos de ellos son elementales para el modelo básico (coordinación, funcional) y otros se utilizan para ámbitos específicos de aplicación (presentación, distribución, replicación, persistencia). Se han utilizado tres estrategias de razonamiento (encadenamiento de reglas hacia adelante y hacia atrás, y el diferencial), las cuales han sido las más útiles al realizar un diagnóstico, tanto en los sistemas expertos basados en reglas como en las personas que realizan este tipo de tareas.

Dado que se requiere de un dominio de aplicación muy específico, y que el diagnóstico médico ha cobrado importancia en nuestros días, se ha elegido el caso de estudio de las enfermedades infecciosas infantiles (la rubéola, el sarampión, la roséola, la escarlatina, la varicela, el dengue, la tifoidea, la parotiditis viral o paperas, la parotiditis bacteriana, el crup espasmódico o laringitis estridulosa, la bronquiolitis infecciosa aguda, la neumonía, la otitis viral y la otitis bacteriana). Los conocimientos y experiencia de un pediatra² en funciones han avalado la información del dominio contenida en el sistema presentado en este trabajo.

Cabe señalar que el modelo PRISMA ha sido aplicado a otros dominios, como los sistemas de tele-operación para la limpieza de buques por medio de un brazo-robot [Cab05a], así como en sistemas de manejo de cuentas bancarias [Per04], comprobándose la mejora del desarrollo y mantenimiento de dichos sistemas a través de la aproximación de aspectos y componentes.

El sistema realiza tomas de decisión al establecer:

- cuáles son las posibles enfermedades que pudiese tener el paciente a diagnosticar, y con ello obtener solamente los signos y síntomas que padece el paciente de esas enfermedades para llegar a un diagnóstico;

² De la Rosa Palacios, Jéssica. Comunicación personal, diciembre de 2004

- si son o no necesarios los estudios de laboratorio;
- sólo aquellos estudios necesarios y adecuados para inferir la enfermedad, en caso de decidir realizar un diagnóstico de laboratorio.

El sistema propuesto permite:

- presentar un enfoque integrado y flexible para describir modelos de arquitectura complejos, distribuidos y reutilizables;
- establecer un modelo arquitectónico para sistemas de diagnóstico médico;
- reducir el costo del diseño y mantenimiento en comparación con otros modelos arquitectónicos, gracias a que los componentes especificados se pueden reutilizar en diferentes arquitecturas;
- representar una utilidad importante, en la detección de fallas o desperfectos de un ente, sin la presencia física del personal experto o perito;
- dirigir otros sistemas bajo este modelo a diversas clases de usuarios;
- estandarizar el proceso de diagnóstico, al permitir la unificación de criterios diagnósticos, así como la disponibilidad del conocimiento de los expertos con el fin de alcanzar una alta calidad diagnóstica. Cuando el proceso de diagnóstico proporcionado por este sistema se aplica, se logra que el diagnóstico sea de la misma calidad para todo lo que será diagnosticado (cada paciente, en el caso de la medicina). La capacidad del sistema para aplicar diferentes estrategias de razonamiento asegura, además, suficiente profundidad en las posibilidades del diagnóstico, la cual, en algunos casos, incluso rebasa la capacidad de un experto humano (el médico en el caso de la medicina), por la amplitud del conocimiento específico contenido en el propio sistema;
- permitir automatizar tareas rutinarias elevando la calidad de trabajo y eficiencia de las personas, disminuyendo con esto, la probabilidad de error, sobretodo en situaciones complejas, donde la subjetividad humana puede llevar a conclusiones erróneas;
- optimizar los recursos económicos, ya que para llegar a un diagnóstico y tratamiento final, el sistema propone sólo aquellos estudios de laboratorio necesarios y útiles. Este hecho exige del proceso diagnóstico que se efectúen los estudios que se consideren necesarios y adecuados, evitando el dispendio de recursos o la solicitud de estudios incompletos, los cuales, en la gran mayoría de las veces, incrementan considerablemente el tiempo dedicado al proceso diagnóstico, y por supuesto al cargo económico asociado a éste. Por lo anterior, el modelo propuesto contribuye a reducir el costo asociado al proceso de diagnóstico;
- mejorar la productividad, permitiendo ahorrar tiempo y dinero, dado que con los conocimientos fácilmente accesibles, los problemas se pueden ir resolviendo a medida que se van planteando y las decisiones se pueden tomar rápidamente;

- conservar los conocimientos importantes, ya que el componente Base de Conocimientos permite guardar y encapsular los valiosos conocimientos de uno o varios especialistas en forma de paquetes de software;
- contener conocimientos predefinidos que se utilizan para tomar todas las decisiones, adquiriendo información contenida en el propio sistema, con el fin de limitar las posibles opciones del árbol de decisión para que se realice el diagnóstico.

Las ventajas de la aproximación presentada en este trabajo, frente a la gran variedad de sistemas expertos existentes para resolver tareas de diagnóstico en el dominio de la medicina, se centran en que se ofrece una aportación al desarrollo de estos sistemas, ya que dicho modelo arquitectónico:

- se puede utilizar en sistemas centralizados o distribuidos, ya que mediante el aspecto de distribución se especifican las características que definen la localización dinámica del elemento arquitectural en el cual se integra, así como los servicios necesarios para implementar estrategias de distribución de los elementos arquitectónicos (como la movilidad) con el fin de optimizar la distribución de la tipología del sistema resultante;
- reduce la complejidad del sistema al gestionar la separación de las unidades funcionales, mediante diferentes niveles de abstracción,;
- está formado por elementos arquitectónicos independientes, de forma tal que se pueden añadir o intercambiar componentes y/o conectores, sin afectar el funcionamiento general del sistema ni el resto de sus elementos que lo conforman;
- favorece el mantenimiento del sistema al considerar los diferentes intereses (*crosscutting concerns*) del sistema y separarlos en aspectos;
- reutiliza tanto los aspectos como los componentes, ya sea dentro de la misma arquitectura en dominios diferentes o bien en arquitecturas diferentes;
- es independiente de plataforma y es abordado desde la perspectiva del problema y no de la solución, permitiendo generalidad en esta aproximación para tareas de diagnóstico aplicado a diversos dominios;
- evita que los cambios de infraestructura sean una fuente de variabilidad ya que el alto nivel de abstracción de las especificaciones del modelo arquitectónico hace que éstas sean independientes de la tecnología aplicada. Por ello, ante un cambio tecnológico, la especificación no se ve afectada, sólo el proceso de traducción automático de la especialización a código.

Aunadas las ventajas de un sistema experto con las del modelo PRISMA, la arquitectura propuesta en este trabajo posee una importancia, pues conforme a lo consultado en la literatura, aún no se ha creado un sistema con las características y funciones que éste realiza, dada la incorporación de aspectos en su arquitectura.

Sin embargo, es imposible representar o plasmar todo el conocimiento y experiencia en un sistema artificial, aunque la cantidad y calidad del conocimiento contenido en el sistema puede ser mayor a la de un experto humano promedio. Así mismo, el buen desempeño del sistema depende en gran parte de la habilidad y destreza del usuario para realizar el planteamiento inicial del problema, i.e. la introducción correcta de los datos de entrada (signos y síntomas).

TRABAJO FUTURO

Con base en el modelo propuesto en esta investigación, en el futuro se podría mejorar la arquitectura de este sistema, al:

- modificar este modelo para realizar tareas de diagnóstico en todos los campos del conocimiento (generalización de la aproximación), i.e. para cualquier caso de uso que se eligiera, con sólo incorporar las respectivas reglas del dominio en la base de conocimientos;
- ampliar la base de conocimientos, para enriquecer al sistema en la aplicación de las reglas;
- añadir al modelo varias Bases de Conocimientos (conocimiento distribuido);
- cambiar el Motor de Inferencia por otro que contemple adicionalmente otras estrategias de razonamiento adecuadas para el proceso de diagnóstico;
- contemplar además del diagnóstico, otras tareas como la interpretación y predicción, ya que comparten características comunes;
- especificar de manera más detallada los aspectos de presentación y persistencia;
- incorporar un nuevo aspecto de decisión, al separarlo del aspecto funcional, ya que se ha detectado que dicho aspecto puede ser de gran ayuda al desarrollo de los sistemas de toma de decisión;
- estudiar y determinar cuáles son las necesidades de evolución del sistema, tras un análisis de los requisitos, para dotar al futuro sistema de propiedades reflexivas a través de un metanivel que le dé la capacidad de evolución;
- finalmente, y a fin de conseguir la validación del modelo propuesto, llevar a cabo la traducción de la especificación realizada en el LDA de PRISMA a código C#, utilizando el compilador de modelos PRISMA [Cab05a], lo que permitirá ejecutar dicho modelo sobre el middleware PRISMANET [Cos05].

BIBLIOGRAFÍA Y REFERENCIAS

[Ade92] Adedeji B. *Expert Systems Applications in Engineering and Manufacturing*, EUA: Prentice Hall, 1992. 436 p.

[Adr01] Adriaenssens V, Goethals P and De Pauw N. *Development of a fuzzy expert system for the prediction of macroinvertebrate taxa* Meded Rijksuniv Gent Fak Landbouwkd Toegep Biol Wet. 2001;66(4):225-8

[Agos05] D'Agostino E, Casali A, Corti R, Torres A y Siragusa M. *EndoDiag II: Un sistema de apoyo a la enseñanza diagnóstica en Endodoncia*. Reporte técnico, Depto. de sistemas e informática, Escuela de Ing. Electrónica, Fac. de Cs. exactas, ingenierías y agrimensura, Universidad Nacional de Rosario, Argentina, mayo 2005

[Ali03a] Ali N., Silva J., Jaén J., Ramos I., Carsí J. y Pérez J. *Distribution Patterns in Aspect-oriented component Based Software Architectures*, Actas IV Jornadas de Trabajo de Distributed Objects, Languages, Methods and Environments, DOLMEN. Alicante, España, 2003

[Ali03b] Ali N., Silva J., Jaen J., Ramos I., Carsi J. and Pérez J. *Mobility and Replicability Patterns in Aspect-Oriented Component-Based Software Architectures*. Proceedings of 15th IASTED, Parallel and Distributed Systems, Acta Press: Marina del Rey, C.A., USA, 2003.

[Ali04a] Ali N, Pérez J. y Ramos I. *High Level Specification of Distributed and Mobile Information Systems*, Second International Symposium on Innovation in Information & Communication Technology(ISSICT2004), 2004

[Ali04b] Ali N., Carsí J. y Ramos I. *Analysis of a Distribution Dimension for PRISMA*, Actas de las IX Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Málaga, España, 2004

[Ali04c] Ali N., Cercos J., Ramos I., Letelier P. y Carsí J. *Distribution in PRISMA*, Actas de las II Jornadas de DYNAMICA, Malaga, España, 2004

[Ali04d] Ali N. *Distribution in an Aspect-Oriented Component based Software Architecture through Prisma*, Trabajo de investigación dentro del programa de doctorado de Programación Declarativa e Ingeniería de la Programación, Universidad Politécnica de Valencia, España, 2004

- [Ali04e] Ali N. *A Compiler for the Automatic Generation of the Distribution Aspect to Distributed Applications*, Trabajo de investigación dentro del programa de doctorado de Programación Declarativa e Ingeniería de la Programación, Universidad Politécnica de Valencia, España, 2004
- [Ali04f] Ali N., Pérez J. and Ramos I. *High Level Specification of Distributed and Mobile Information Systems*, Proceedings of Second International Symposium on Innovation in Information & Communication Technology ISIICT 2004, Amman, Jordan, 2004.
- [Ali05] Ali N., Ramos I. and Carsí J. *A Conceptual Model for Distributed Aspect-Oriented Software Architectures*, International Conference on Information Technology Coding and Computing, ITCC 2005, IEEE Computer Society, Las Vegas, NV, USA, 2005
- [All97] Allen, R. *Formal Approach to software Architecture*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1997.
- [And99] Andrade L. and Fiadeiro J. *Interconnecting Objects via Contracts*. OOPSLA'99.
- [Aos] Aspect Oriented Software Development, <http://aosd.net>
- [Bac00] Bachman F., Bass L., Chastek G., Donohoe P. and Peruzzi F. *The Architecture Based Design Method*. Technical Report CMU/SEI-2000-TR-001, Carnegie Mellon University, USA, January 2000.
- [Bar89] Barr A. and Feigenbaum E. *The Hand Book of Artificial Intelligence*, vol. I, II y IV. E.U.A.: Addison Wesley, 1989.
- [Bra02] Bracciali A., Brogi A. and Canal C., *Systematic Component Adaptation*. In Proc IDEAS, La Habana, Cuba, 2002.
- [Bra83] Brachman R., Amarel S., Engelman C., Engelmores R., Feigenbaum E. and Wilkins D. *What are Expert Systems?. Building Expert Systems*. E.U.A.: Addison Wesley, 1983.
- [Bre92] Brena R. *Sistema Experto de Diagnóstico de Fallas y Ayuda a la Decisión*, Reporte de desarrollo CIA-RD-003, Centro de Inteligencia Artificial, Campus Monterrey, ITESM, 1992. <http://homepages.mty.itesm.mx/rbrena/CV.html>

- [Bre94] Brena R., Barrón O. y Sahagún R. *Sistema Experto para Detección de Fallas en Máquinas de Polietileno*, Reporte de desarrollo CIA-RD-029, Centro de Inteligencia Artificial, Campus Monterrey, ITESM, Mayo 1994.
- [Buch89] Buchanan B. and Smith R. *Fundamentals of Expert Systems. The Hand Book of Artificial Intelligence*. vol. IV. E.U.A.: Addison Wesley, 1989.
- [Cab05a] Cabedo R, Pérez J. y Ramos I., “Aplicación del lenguaje de descripción de arquitecturas PRISMA a un sistema robótico industrial”, Informe técnico DSIC-II/12/05, Universidad Politécnica de Valencia, Valencia, España, 2005, 201 p.
- [Cab05b] Cabedo R., Pérez J., Carsí J.A. y Ramos I., “Modelado y Generación de Arquitecturas PRISMA con DSL Tools”, en Actas del IV Workshop DYNAMICA, Archena, Murcia, España, 2005, pp. 79-86.
- [Car02] Carlos S M. *Sistema experto de diagnóstico médico del síndrome de Guillian Bare (SEGBS)*, Monografía, Universidad de Lima, Perú, 2002.
- [Car99] Carsí J A, *OASIS como Marco Conceptual para la Evolución de Software*. Tesis doctoral, Universidad Politécnica de Valencia, 1999.
- [Chae98] Chae Y., Park Q., Park K. and Young M. *Development of Medical Decision Support System for Leukemia Management*, Expert Systems with Applications. E.U.A. (vol. 15-1998). Pag. 309-315.
- [Che05] Cherrie JW, Hughson GW. *The validity of the EASE expert system for inhalation exposures*, Ann Occup Hyg. 2005 Mar;49(2):125-34
- [Cos05] Costa C., Pérez J., Ali N., Carsí J. y Ramos I. *PRISMANET middleware: Soporte de la Evolución Dinámica de Arquitecturas Software Orientadas a Aspectos*. Actas X Jornadas de Ingeniería de Software y Bases de Datos, JISBD, Granada, España, 2005.
- [Fel04] Feldon S.E. *Computerized expert system for evaluation of automated visual fields from the ischemic optic neuropathy decompression trial: methods, baseline fields, and six-month longitudinal follow-up*, Trans Am Ophthalmol Soc. 2004;102:269-303
- [For84] Forsyth R. *Expert Systems Principles and Case Studies*. E.U.A.: Chapman and Hall Computing, 1984.

- [Gar01] Garlan D., Cheng S. and Kompanek A. J. *Reconciling the Needs of Architectural Description with Object Modeling Notations*. Science of Computer Programming Journal, Special UML Edition, Elsevier Science, 2001.
- [Ger04] Gerner I, Barratt MD, Zinke S, Schlegel K, Schlede E. *Development and prevalidation of a list of structure-activity relationship rules to be used in expert systems for prediction of the skin-sensitising properties of chemicals*, Altern Lab Anim. 2004 Nov;32(5):487-509.
- [Gon95] González P. *Sistemas Expertos Facultativamente Asociados en Red Cooperativa con Arquitecturas de Pizarrón. Una aplicación en la consulta e interconsulta médica*. Tesis. Instituto de Investigaciones Biomédicas, UNAM. México, 1995.
- [Gon97] González P. and Negrete J. *REDIEX: A Cooperative Network of Expert Systems with Blackboard Architectures*". Expert Systems. (vol. 14 - no. 4-1997). Pag. 180-189.
- [Gon99] González P. and Negrete J. *INTERMED: A Cooperative Network of Expert Systems for Medical Consultation and Interconsultation*. Comunicación personal, enero de 1999.
- [Gue97] Guevara M., Rodríguez M. y González N. *DIAG: un Sistema Experto para el Diagnóstico de Anomalías Craneofaciales*. Revista Cubana de Investigación Biomédica [online], jul.-dic. 1997, vol.16, no.2, p.93-97. ISSN 0864-0300. Disponible en <http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-03001997000200003&lng=es&nrm=iso>..
- [Hay83] Hayes F., Waterman D. and Lenat D. *On Overview of Expert Systems. Building Expert Systems*. E.U.A.: Addison Wesley, 1983.
- [Hug05] Hughson GW, Cherrie JW. *Comparison of measured dermal dust exposures with predicted exposures given by the EASE expert system*, Ann Occup Hyg. 2005 Mar;49(2):111-23
- [Ili05] Iliffe S, Kharicha K, Harari D, Swift C, Stuck AE. *Health risk appraisal for older people in general practice using an expert system: a pilot study*, Health Soc Care Community. 2005 Jan;13(1):21-9.
- [Ira00] Irastrorza D., Jaime A. and Díaz O. *Diseño Basado en Componentes: Alternativas en el Paso de Partición*. Jornadas de Ingeniería del Software y Bases de Datos, JISBD-2000.

- [Jac90] Jackson P. *Introduction to Expert Systems*. 2^a ed. E.U.A.: Addison-Wesley Publishing Company, 1990. 526 p.
- [Jac97] Jacobson I., Griss M. and Johnson P. *Software Reuse. Architecture, Process and Organization for Business Success*. Addison-Wesley, 1997.
- [Jan05]. Jansson ET. *Alzheimer disease is substantially preventable in the United States -- review of risk factors, therapy, and the prospects for an expert software system*. Med Hypotheses. 2005;64(5):960-7
- [Kar04] Karlsson D, Forsum U. *Medical decision-support systems and the concept of context*. Med Inform Internet Med. 2004 Jun;29(2):109-18.
- [Kic01] Kiczales G., Hilsdale E., Huguin J., Kersten M., Plam J. and Griswold W. *An Overview of AspectJ*, Proceeding of the European Conference on Object-Oriented Programming, Springer Verlag, 2001.
- [Kic97] Kiczales G. et al., *Aspect Oriented Programming (ECOOP'97)*, LNCS 124, Springer Verlag, 1997.
- [Kop04] Kopec D, Shagas G, Selman J, Reinharth D, Tamang S. *Development of an expert system for differentiating tension type headaches from migraines*. Stud Health Technol Inform. 2004;103:81-92.
- [Kot04] Kotze B, Brdaroska B. *Clinical decision support systems in psychiatry in the information age*. Australas Psychiatry. 2004 Dec;12(4):361-4.
- [Let98] Letelier P., Sánchez P, Ramos I., Pastor O., *OASIS 3.0: Un Enfoque Formal para el Modelado Conceptual Orientado a Objetos*. Universidad Politécnica de Valencia, SPUPV-98.4011, ISBN 84-7721-663-0, 1998
- [Lie98] Liebewitz J., "The Handbook of Applied Expert Systems", CRC Press, 1998.
- [Lia03] Liao S.-H., "Knowledge Management Technologies and Applications- Literature Review from 1995-2002", in Expert Systems with Applications, Vol. 25, Issue 2, 2003, pp. 155-164.
- [Lia05] Liao S.-H., "Expert Systems Methodologies and Applications- a Decade Review from 1995-2004", in Expert Systems with Applications, Vol. 28, Issue 1, 2005, pp. 93-103.

- [Mar05] Marten K, Grillhosi A, Seyfarth T, Obenauer S, Rummeny EJ, Engelke C. Computer-assisted detection of pulmonary nodules: evaluation of diagnostic performance using an expert knowledge-based detection system with variable reconstruction slice thickness settings, *Eur Radiol.* 2005 Feb;15(2):203-12. Epub 2004 Dec 2.
<http://www.e-ghost.deusto.es/docs/2005/conferencias/Bayes05.pdf>
- [McG02] McGurren F. and Conroy D., *X-ADAPT: An Architecture for Dynamic Systems*, Workshop on Component Oriented Programming. ECOOP, Málaga, Spain, 2002.
- [Mda] *Model Driven Architecture. A Thechnical Perspective*. Object Management Group, Jan. 2001. OMG document ab/2001-01-01
- [Med00] Medvidovic N., Taylor R.N., *A Classification and Comparison Framework for Software Architectures*. In Proc IDEAS, La Habana, Cuba, 2002.
- [Mer89] Merritt D. *Building Expert Systems in Prolog*. E.U.A.: Spring Verlag, 1989.
- [Mill03] J. Miller and J. Mukerji. *MDA Guide*. Object Management Group, Jan. 2003.
- [Mon02] Monge R. and Alves C. and Vallecillo A., *A Graphical Representation of COTS-Based Software Architecture*. In Proc IDEAS, La Habana, Cuba, 2002.
- [Mont03] Montes M. *Sistemas Expertos*. Universidad EAFIT, 2003:
http://dis.eafit.edu.co/labs/labgic/ARTICULOS_%20PUBLICAR/Sistemas%20expertos.doc
- [Mycin] <http://www.fortunecity.com/skyscraper/chaos/279/docs/mycin.htm>
- [Nav03] Navarro E., Ramos I. and Pérez J. *Software Requeriments dor Arquitectured Systems*. 11 th IEEE International Requeriments Engineering Conference, Monterey, California, USA, 2003
- [Neg96] Negrete J., González P. y Guerra A. *Pericia Artificial: Un Aprendizaje Constructivista de Sistemas Expertos*. Editorial Limusa, México, 1996, 259 p.
- [Nor98] Noriega P. y Sierra C. *Subastas y Sistemas Multiagente*. Revista Iberoamericana de Inteligencia Artificial, Número 6, pp. 68-84, 1998
- [Omg] Object Management Group (OMG), <http://www.omg.org>

[Ovi03] Oviedo N J. <http://www.fi.uba.ar/laboratorios/lie/Revista/Articulos/010100/A3%20set%202003.pdf>

[Pat90] Patterson D. *Introduction to Artificial Intelligence and Expert Systems*. England: Prentice Hall. 1990. 24 p.

[Per02]. Pérez J., Ramos I., Lorenzo A., Letelier P. and Jaén J. *PRISMA: Plataforma OASIS para Modelos Arquitectónicos*. Actas VII Jornadas de Ingeniería de Software y Bases de Datos, JISBD, Escorial (Madrid), España, 2002.

[Per03a] Pérez J., Ramos I., Jaén J., Letelier P. and Navarro E: *PRISMA: Towards Quality, Aspect Oriented and Dynamic Software Architectures*. 3rd IEEE International Conference on Quality Software, Dallas, Texas, USA, 2003. IEEE Computer Society Press ISBN 0-7695-2015-4, pp 59-66

[Per03b] Pérez J. *OASIS como Soporte Formal para la Definición de Modelos Hipermedia Dinámicos, Distribuidos y Evolutivos*. Trabajo de investigación dentro del programa de doctorado de Programación Declarativa e Ingeniería de la Programación, Universidad Politécnica de Valencia, España, 2003.

[Per03c] Pérez J., Ali N., Ramos I., Pastor J., Sánchez P. y Álvarez B. *Desarrollo de un Sistema de Teleoperación utilizando el enfoque PRISMA*, VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Alicante, España, 2003

[Per03d] Pérez J., Ali N., Ramos I. y Carsí, J. *PRISMA: Arquitecturas Software Orientadas a Aspectos y Basadas en Componentes*, AOSD workshop in collaboration with VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Alicante, España, 2003

[Per03e] Pérez J., Ramos I., Jaén J. y Letelier P. *PRISMA: Development of Software Architectures with an Aspect Oriented, Reflexive and Dynamic Approach*, Dagstuhl Seminar N° 03081, Report N° 36 "Objects, Agents and Features", copyright (c) IBFI gem. GmbH, Schloss Dagstuhl, D-66687 Wadern, Germany . Eds.H.-D. Ehrich (Univ. Braunschweig, D), J.-J. Meyer (Utrecht, NL), M. Ryan (Univ. of Birmingham, GB), 2003

[Per03f] Pérez J., Ramos I., Carsí J. *Compilador para la Generación Automática del Metanivel de una Especificación mediante la Reificación de Propiedades del Nivel Base*, Informe Técnico DSIC-II/23/03 Universidad Politécnica de Valencia, 2003

[Per04] Pérez J., Cabedo R., Sánchez P., Carsí J., Pastor J., Ramos I y Álvarez B. *Arquitectura PRISMA para el Caso de Estudio: Brazo Robot*, Actas del II Workshop DYNAMICA – Dynamic and Aspect-Oriented Modeling for Integrated Component-based

Architectures, pags. 119-127, junto a Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Málaga, España, 2004

[Per05] Pérez J., Ali N., Carsí J., Ramos I. y Navarro E. *Designing Software Architectures with an Aspect-Oriented Language*, Journal on Aspect Orientation, Published by Houman Younessi, RISE @ADVICE 2004 ISSN 1548-3851, Volumen 1.1. pags. 20, 2005

[Pro05] Prochaska JO, Velicer WF, Redding C, Rossi JS, Goldstein M, DePue J, Greene GW, Rossi SR, Sun X, Fava JL, Laforge R, Rakowski W, Plummer BA. *Stage-based expert systems to guide a population of primary care patients to quit smoking, eat healthier, prevent skin cancer, and receive regular mammograms*, Prev Med. 2005 Aug; 41(2):406-16.

[Rich94] Rich E. y Knight K. *Inteligencia Artificial*. 2ª ed. España: Mc Graw Hill, 1994. 703p.

[Rod01] Rodríguez R. *Diseño y desarrollo de un SE auxiliar en el diagnóstico temprano de cáncer en niños*, Revista mexicana de pediatría, Vol 68, Num. 4, Jul-ago 2001, pp 160-161.

[Rod99] Rodríguez H.; Gálvez D.; Molina O. y Mora F. *COOPERATIVE: Un shell para el desarrollo de sistemas basados en el conocimiento utilizando la solución cooperativa de problemas distribuidos*. Universidad Central de las Villas, Cuba, 1999

[Ros00] De la Rosa J. *YOTANPOX: Un sistema experto para el diagnóstico de enfermedades cardiovasculares*. Tesis. Fundación Arturo Rosenblueth. México: 2000.

[Ros04] De la Rosa J., Comunicación personal, diciembre de 2004.

[Rui04] Ruiz M E. *Sistemas expertos para la realización del diagnóstico de parálisis facial con electromiografía: PARFAC*, Revista de investigación en sistemas informáticos, FISI-UNMSM, RISI 1(1), 68-72 (2004)

[Rus96] Russell S. y Norbving P. *Inteligencia Artificial: Un Enfoque Moderno*. México: Prentice Hall Hispanoamericana, S.A, 1996. 979 p.

[San05] Sancipriano GP. *Artificial intelligence--the knowledge base applied to nephrology*, G Ital Nefrol. 2005 Jan-Feb;22(1):47-62

[San90] SÁNCHEZ Y BELTRÁN, Juan Pablo. *Sistemas Expertos. Una Metodología de Programación*. México: Macrobit Editores, S.A. de C.V., 1990. 261 p.

[San95] Sánchez A. *Aplicación de los Sistemas Expertos en Contabilidad*. Departamento de Contabilidad, Universidad de Valencia, España, 1995: <http://ciberconta.unizar.es/Biblioteca/0002/Sanchez95.html>

[Sch05] Schurink CA, Lucas PJ, Hoepelman IM, Bonten MJ. *Computer-assisted decision support for the diagnosis and treatment of infectious diseases in intensive care*, Lancet Infect Dis. 2005 May;5(5):305-12. units.

[Shi04] Shi H, Paolucci U, Vigneau-Callahan KE, Milbury PE, Matson WR, Kristal BS. *Development of biomarkers based on diet-dependent metabolic serotypes: practical issues in development of expert system-based classification models in metabolomic studies*, OMICS. 2004 Fall;8(3):197-208.

[Sil03] Silva J., Ali N., Carsí J. y Ramos I. *El Aspecto de Distribución de PRISMA*, VIII Jornadas de Ingeniería del Software y Bases de Datos (JISBD), Alicante, España, 2003.

[Sou99] D'Souza D. and Wills A. *Objects, Components and Frameworks with UML. The Catalysis Approach*, Addison-Wesley, 1999.

[Sta04] Statnikov A, Aliferis CF, Tsamardinos I. *Methods for multi-category cancer diagnosis from gene expression data: a comprehensive evaluation to inform decision support system development*, Medinfo. 2004;11(Pt 2):813-7.

[Tick05] Tickner J, Friar J, Creely KS, Cherrie JW, Pryde DE, Kingston J. *The development of the EASE model*. Ann Occup Hyg. 2005 Mar;49(2):103-10.

[Van04] Van Hoof V, Wormek A, Schleutermann S, Schumacher T, Lothaire O, Trendelenburg C. *Medical expert systems developed in jMD, a Java based expert system shell: application in clinical laboratories*, Medinfo. 2004;11(Pt 1):89-93.

[Yeu04] Yeung DS, Wang XZ, Tsang EC. *Handling interaction in fuzzy production rule reasoning*, IEEE Trans Syst Man Cybern B Cybern. 2004 Oct;34(5):1979-87

APÉNDICE A:

ONTOLOGÍA DEL DIAGNÓSTICO MÉDICO:

El diagnóstico médico

El diagnóstico médico es entendido como el proceso encaminado a la identificación o reconocimiento de una enfermedad sobre la base de los signos y síntomas presentes, con el apoyo de los estudios de laboratorio y gabinete. El diagnóstico médico indica todo el proceso de investigación del paciente, a partir de las observaciones y razonamientos del médico para determinar la enfermedad.

El grafo RDF correspondiente al diagnóstico médico se puede apreciar en la siguiente figura:

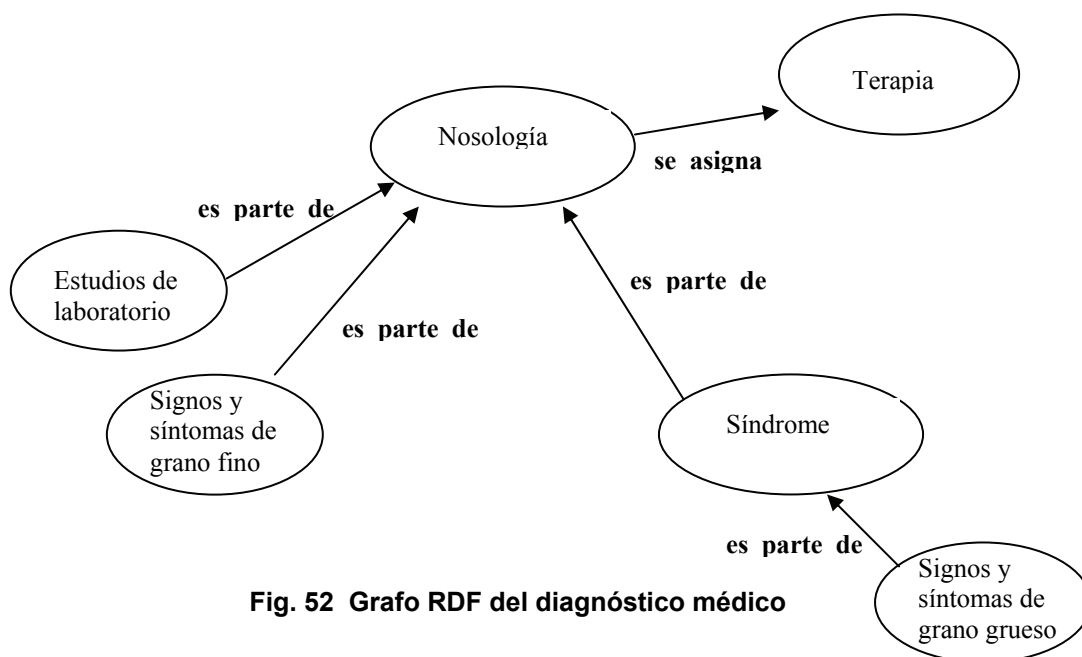


Fig. 52 Grafo RDF del diagnóstico médico

Algunos conceptos y términos médicos

Para lograr una mayor comprensión del dominio de aplicación y funcionamiento del sistema, es necesario introducir los siguientes conceptos y términos médicos relacionados: signos, síntomas, síndrome o entidad etiofisiopatológica, enfermedad o entidad nosológica y terapia o entidad terapéutica³.

Para ello, y para una mejor comprensión de dichos términos, se hará referencia a una de las más comunes o frecuentes de las enfermedades infecciosas infantiles, la bronquiolitis infecciosa aguda. Cabe señalar que entre otras enfermedades, dentro de este dominio, se encuentran: la rubéola, el sarampión, la roséola, la escarlatina, la varicela, el dengue, la tifoidea, la parotiditis viral o paperas, la parotiditis bacteriana, el crup espasmódico o laringitis estridulosa, la neumonía, la otitis viral y la otitis bacteriana.

Signos

Un signo es la evidencia objetiva de una enfermedad, en especial cuando ésta es observada e interpretada más por el médico que por el paciente. Un signo físico es una indicación de la condición corporal que puede ser percibida directamente (por ejemplo, a través de la auscultación) cuando el médico examina al paciente. El signo es parte principal del proceso de diagnóstico médico, ya que su principal virtud es el ser un dato objetivo.

Como ejemplo de signos se pueden citar los siguientes asociados al diagnóstico de la enfermedad bronquiolitis infecciosa aguda. Todos ellos se obtienen durante el proceso de exploración física y por lo tanto sólo son detectados por el médico que realiza la exploración:

<u>DIAGNÓSTICO</u>	Bronquiolitis Infecciosa Aguda
<u>Signos</u>	Fiebre mayor a 39 Fiebre continua Fiebre de 2 a 3 días Tos seca Tos por accesos Dificultad respiratoria grave Odinofagia garganta roja Odinofagia de 2 a 3 días Apnea por periodos largos

Pruebas de laboratorio

³ DE LA ROSA PALACIOS, Jéssica. Comunicación personal, diciembre de 2004.

Las pruebas de laboratorio forman parte de los signos que presenta el paciente. Estas, junto con el resto de los signos y síntomas, son una parte muy importante dentro de las reglas que se aplican en los encadenamientos, dado que el resultado del laboratorio ayuda al sistema a tomar una decisión sobre el diagnóstico diferencial que se aplica.

Como ejemplo de prueba de laboratorio, se puede citar el siguiente asociado al diagnóstico de la enfermedad bronquiolitis infecciosa aguda:

<u>DIAGNÓSTICO</u>	Bronquiolitis Infecciosa Aguda
<u>Pruebas de laboratorio</u>	Biometría hemática leucocitosis

Síntomas

El síntoma es cualquier prueba subjetiva de enfermedad o del estado de un paciente. Se refiere a los datos proporcionados por el paciente, relacionados con alguna enfermedad. Son subjetivos, matizados por el padecimiento y las características psicológicas, sociológicas y biológicas del paciente. Es decir, el síntoma “dolor de ganglios” referido por un paciente pudiera estar matizado por su capacidad para resistir el dolor, su historia sobre antecedentes de dolores semejantes y su fortaleza para soportarlo.

Durante el proceso de diagnóstico médico, los síntomas ayudan a localizar al sistema anatómico que pudiera estar comprendido o asociado a la enfermedad del paciente, como es el caso de los síntomas: dolor de garganta (odinofagia), dificultad para deglutir (disfagia), o síntomas digestivos (diarrea, estreñimiento). También existen síntomas totalmente inespecíficos y que pudieran estar, por lo tanto, asociados a muchas enfermedades y aunque no orientan directamente al sistema anatómico afectado, si refieren sobre la intensidad o evolución del padecimiento. Por ejemplo, astenia, adinamia y anorexia, los cuales pudieran presentarse desde el inicio o aparecer durante la evolución del padecimiento.

Como ejemplo de síntoma se pueden citar el siguiente asociado al diagnóstico de la enfermedad bronquiolitis infecciosa aguda:

<u>DIAGNÓSTICO</u>	Bronquiolitis Infecciosa Aguda
<u>Síntomas</u>	Ardor de garganta intenso

Entidad etiofisiopatológica o síndrome

La entidad etiofisiopatológica, comúnmente llamada síndrome, es el grupo de signos y síntomas que se manifiestan simultáneamente y que, considerados como un todo, son característicos de cierta enfermedad. Existen procesos mórbidos, o grupo de enfermedades que se caracterizan siempre por la presencia de ciertos signos y síntomas,

y cuando éstos no están presentes en su totalidad, no es posible arribar a ese diagnóstico sindromático. En el proceso diagnóstico, el síndrome puede ocupar el diagnóstico final o el paso previo a este diagnóstico cuando no se puede encontrar la causa definitiva y única de ese síndrome. Como ejemplo de síndrome podemos citar el síndrome exantemático, el cual se caracteriza por la presencia de exantema y fiebre generalizados. Este síndrome generalmente es percibido por el paciente (síntoma) y durante el proceso de exploración lo describe el médico (signo). Si faltara alguno de estos datos, ya no estaríamos en presencia de un síndrome exantemático.

Como ejemplo de síndrome podemos citar el siguiente, asociado al diagnóstico de la enfermedad bronquiolitis infecciosa aguda:

<u>DIAGNÓSTICO</u>	Bronquiolitis Infecciosa Aguda
<u>Síndrome</u>	Ira

Entidad nosológica o enfermedad

Relacionado a la nosología (del griego nosos: enfermedad y logos: palabra, razón, tratado), la enfermedad es un dato físico y funcional, manifestado por síndromes y comprobado por análisis de laboratorio y estudios de gabinete.

Como ejemplo de las enfermedades en este dominio podemos citar a la bronquiolitis infecciosa aguda, que es una enfermedad infecciosa infantil, la cual es de las más frecuentes entre los niños.

Entidad terapéutica o terapia

La entidad terapéutica o terapia es el tratamiento dado al paciente.

Como ejemplo de la terapia aplicada a la bronquiolitis infecciosa aguda, se puede citar la siguiente:

<u>DIAGNÓSTICO</u>	Bronquiolitis Infecciosa Aguda
<u>Terapia</u>	Sintomático y de soporte (antitérmicos, broncodilatadores, oxigenoterapia y nebulizaciones)

APÉNDICE B

ELEMENTOS DEL DOMINIO DEL DOMINIO DEL DIAGNÓSTICO MÉDICO: LAS ENFERMEDADES INFECCIOSAS INFANTILES

Los signos, síntomas, entidades etiofisiopatológicas o síndromes, entidades nosológicas o enfermedades y entidades terapéuticas o terapias, asociados al diagnóstico de la enfermedades infecciosas infantiles más comunes, dominio de aplicación del caso de estudio, son:

Signos y síntomas de grano grueso

signos:

fiebre
tos
dificultad respiratoria
odinofagia
apnea
estrindor
rinorea
hipoventilación
exantema
ademopatías
conjuntivitis
enantema
coriza
linfoademopatías
vómito
crecimiento de parótidas
inflamación del tímpano
otorrea
pruebas de laboratorio

síntomas:

ardor de garganta
cefalea
trastornos digestivos
dolor a la masticación
dolor de parótidas
otalgia
hipoacusia

Signos y síntomas de grano fino

signos:

fiebre continua
fiebre discontinua
fiebre de 1 a 2 días

síntomas

ardor de garganta intenso
cefalea constante
cefalea dolor retocular

fiebre de 2 a 3 días
fiebre de 3 a 5 días
fiebre en picos
fiebre vespertina
fiebre mayor a 39
fiebre mayor a 38
fiebre mayor a 37
fiebre de 38 a 39
fiebre de 38 a 39.5
tos seca
tos por accesos
tos con flegma
tos frecuente
dificultad respiratoria leve a moderada
dificultad respiratoria grave
dificultad respiratoria polipnea
dificultad respiratoria tiraje
odinofagia de 2 a 3 días
odinofagia garganta roja
rinorea transparente
estrindor laríngeo
apnea periodos largos
hipoventilación crepitante pulmonar
crecimiento de parótidas
exantema máculas
exantema maculopapuloso
exantema vesículas
exantema no confluyente
exantema confluyente
exantema inicia en cara
exantema inicia en tronco
exantema inicia en cuero cabelludo
exantema inicia en palmas y plantas
exantema inicia en abdomen y pecho
exantema petequias
exantema dispersas
exantema eritematoso
exantema costras
exantema puede producir escamas
exantema piel de lija
exantema escamas

trastornos digestivos diarrea
trastornos digestivos estreñimiento
dolor a la masticación espontáneo aguda
dolor de parótidas
otalgia intensa
otalgia continua
hipoacusia de 2 a 3 días

exantema rosa
exantema rosa pálido
exantema rojo
exantema evanescente
adempatías retroauricular
adempatías dolorosas
adempatías todos los ganglios
linfoadempatías en todos los ganglios
conjuntivitis intensa
conjuntivitis moderada
coriza moderada
enanema placas de Koplick
vómito intenso
dolor a masticación acompañado con borramiento del ángulo mandibular
crecimiento de parótidas
inflamación del tímpano de 2 a 3 días
inflamación del tímpano rojo
otorrea abundante
otorrea escasa
otorrea purulenta
otorrea transparente
otorrea fétida
otorrea sin olor
otorrea no existe
estrindor laríngeo
resultados de laboratorio biimetría hemática positiva
resultados de laboratorio biimetría hemática leucopenia
resultados de laboratorio biimetría hemática linfocitosis
resultados de laboratorio biimetría hemática leucocitosis
resultados de laboratorio biimetría hemática neutrofilia
resultados de laboratorio biimetría hemática neutropenia
resultados de laboratorio biimetría hemática eosinofilia
resultados de laboratorio exurado faríngeo estreptococo neumonice
resultados de laboratorio exurado faríngeo hemofilus influenza
resultados de laboratorio exurado faríngeo escherichia coli
resultados de laboratorio exurado de oído estreptococo hemolítico

Entidades etiofisiopatológicas o síndromes:

enfermedad exantemática
parotiditis

ira
otitis

Entidades nosológicas o enfermedades:

rubeóla
sarampión
roseóla
escarlatina
varicela
dengue
tifoidea
parotiditis viral o paperas
parotiditis bacteriana
crup espasmódico o laringitis estridulosa
bronquiolitis infecciosa aguda
neumonía
otitis viral
otitis bacteriana

Entidades terapéuticas o terapia:

sintomático y de soporte:
antitérmicos
antitérmicos (excepto aspirina)
antibióticos: penicilina, eritromicina, ampicilina, clorafenicol, trimatropima-sulfametaxazol,
cefuroxona, penicilina G, amoxicilina
hidratación
humidificación
aire frío
sedación
tranquilizar al paciente
broncodilatadores
oxigenoterapia
nebulizaciones
antitusígenos

Las reglas para inferir las entidades nosológicas son:

IF enfermedad extantemática

THEN (rubeóla,0.60) **or**
 (sarampión,0.60) **or**
 (roseóla,0.60) **or**
 (escarlatina,0.60) **or**
 (varicela,0.60) **or**
 (dengue,0.60) **or**
 (tifoidea,0.60).

IF parotiditis

THEN (parotiditis viral,0.60) **or**
 (parotiditis bacteriana,0.60).

IF ira

THEN (crup espasmódico,0.60) **or**
 (bronquiolitis infecciosa aguda,0.60) **or**
 (neumonía,0.60).

IF otitis

THEN (otitis viral,0.60) **or**
 (otitis bacteriana,0.60)

IF enfermedad exantemática **and**

 exantema=maculopapuloso **and**

 exantema=no confluyente **and**

 exantema=inicia en cara **and**

 exantema=rosa pálido **and**

 fiebre=continua **and**

 fiebre=de 1 a 2 días **and**

 adenopatías=retroauricular **and**

 adenopatías=dolorosas **and**

 resultados de laboratorio=biometría hemática leucopenia

THEN (rubeóla, 1.0)

IF enfermedad exantemática **and**

 exantema=maculopapuloso **and**

 exantema=no confluyente **and**

 fiebre=continua **and**

 adenopatías=retroauricular

THEN (rubeóla, 0.80)

IF enfermedad exantemática **and**
exantema=maculopapuloso **and**
exantema=no confluyente **and**
exantema=rosa pálido **and**
adenopatías=retroauricular **and**
resultados de laboratorio=biometría hemática leucopenia

THEN (rubeóla, 0.90)

IF enfermedad exantemática **and**
exantema=maculopapuloso **and**
exantema=confluyente **and**
exantema=inicia en cara **and**
exantema=eritematoso **and**
exantema=puede producir escamas **and**
fiebre=continua **and**
fiebre=de 2 a 3 días **and**
conjuntivitis=intensa **and**
exantema=placas de Koplick **and**
resultados de laboratorio=biometría hemática leucopenia **and**
resultados de laboratorio=biometría hemática linfocitosis

THEN (sarampión, 1.0)

IF enfermedad exantemática **and**
exantema=maculopapuloso **and**
exantema=eritematoso **and**
fiebre=continua **and**
conjuntivitis=intensa

THEN (sarampión, 0.80)

IF enfermedad exantemática **and**
exantema=maculopapuloso **and**
exantema=eritematoso **and**
exantema=puede producir escamas **and**
exantema=placas de Koplick **and**
resultados de laboratorio=biometría hemática leucopenia **and**
resultados de laboratorio=biometría hemática linfocitosis

THEN (sarampión, 0.85)

IF enfermedad exantemática **and**

exantema=evanescente **and**
exantema=inicia en tronco **and**
exantema=rosa **and**
fiebre=continua **and**
fiebre=de 1 a 2 días **and**
resultados de laboratorio=biometría hemática leucopenia **and**
resultados de laboratorio=biometría hemática linfocitosis
THEN (roseola, 1.0)

IF enfermedad exantemática **and**
 exantema=evanescente **and**
 exantema=rosa **and**
 fiebre=continua
THEN (roseola, 0.75)

IF enfermedad exantemática **and**
 exantema=evanescente **and**
 exantema=inicia en tronco **and**
 fiebre=de 1 a 2 días **and**
 resultados de laboratorio=biometría hemática leucopenia **and**
 resultados de laboratorio=biometría hemática linfocitosis
THEN (roseola, 0.85)

IF enfermedad exantemática **and**
 exantema=rojo **and**
 exantema=piel de lija **and**
 exantema=escamas **and**
 exantema=rosa pálido **and**
 fiebre=continua **and**
 fiebre=de 3 a 5 días **and**
 adenoopatías=todos los ganglios **and**
 resultados de laboratorio=biometría hemática linfocitosis **and**
 resultados de laboratorio=biometría hemática leucocitosis
THEN (escarlatina, 1.0)

IF enfermedad exantemática **and**
 exantema=rojo **and**
 exantema=piel de lija **and**
 fiebre=continua **and**
 adenoopatías=todos los ganglios
THEN (escarlatina, 0.80)

IF enfermedad exantemática **and**
exantema=rojo **and**
exantema=piel de lija **and**
exantema=escamas **and**
resultados de laboratorio=biometría hemática linfocitosis **and**
resultados de laboratorio=biometría hemática leucocitosis
THEN (escarlatina, 1.0)

IF enfermedad exantemática **and**
exantema=máculas **and**
exantema=vesículas **and**
exantema=costras **and**
exantema=inicia en cuero cabelludo **and**
exantema=dispersas **and**
fiebre=en picos **and**
fiebre=vespertina **and**
fiebre=de 2 a 3 días **and**
resultados de laboratorio=biometría hemática leucocitosis **and**
resultados de laboratorio=biometría hemática neutrofilia
THEN (varicela, 1.0)

IF enfermedad exantemática **and**
exantema=vesículas **and**
exantema=costras **and**
exantema=dispersas **and**
fiebre=en picos
THEN (varicela, 0.85)

IF enfermedad exantemática **and**
exantema=vesículas **and**
exantema=inicia en cuero cabelludo **and**
fiebre=vespertina **and**
resultados de laboratorio=biometría hemática leucocitosis **and**
resultados de laboratorio=biometría hemática neutrofilia
THEN (varicela, 0.90)

IF enfermedad exantemática **and**
exantema=rosa **and**
exantema=maculopapuloso **and**
exantema=inicia en palmas y plantas **and**

exantema=petequias **and**
fiebre=mayor a 38 **and**
fiebre=continua **and**
conjuntivitis=moderada **and**
coriza=moderada **and**
resultados de laboratorio=biometría hemática leucositosis
THEN (dengue, 1.0)

IF enfermedad exantemática **and**
 exantema=maculopapuloso **and**
 exantema=inicia en palmas y plantas **and**
 exantema=petequias **and**
 fiebre=continua
THEN (dengue, 0.80)

IF enfermedad exantemática **and**
 exantema=maculopapuloso **and**
 exantema=inicia en palmas y plantas **and**
 conjuntivitis=moderada **and**
 coriza=moderada **and**
 resultados de laboratorio=biometría hemática leucositosis
THEN (dengue, 0.85)

IF enfermedad exantemática **and**
 fiebre=vespertina **and**
 fiebre=en picos **and**
 fiebre=de 3 a 5 días **and**
 exantema=maculopapuloso **and**
 exantema=rosa **and**
 exantema=inicia en abdomen y pecho **and**
 linfadenopatías=en todos los ganglios **and**
 cefalea=constante **and**
 cefalea=dolor retrocular **and**
 vómito=intenso **and**
 trastornos digestivos=diarrea **or**
 trastornos digestivos=estreñimiento **and**
 resultados de laboratorio=biometría hemática leucopenia **and**
 resultados de laboratorio=biometría hemática neutropeña
THEN (tifoidea, 1.0)

IF enfermedad exantemática **and**

IF parotiditis **and**
fiebre=continua **and**
fiebre=mayor a 37 **and**
dolor de parótidas **and**
crecimiento de parótidas **and**
dolor a la masticación espontáneo agudo **and**
dolor a la masticación acompañado con borramiento del ángulo mandibular **and**
odinofagia=2 a 3 días **and**
ardor de garganta=intenso **and**
resultados de laboratorio=biometría hemática leucocitosis
THEN (parotiditis bacteriana, 1.0)

IF parotiditis **and**
fiebre=continua **and**
fiebre=mayor a 37 **and**
dolor a la masticación espontáneo agudo
THEN (parotiditis bacteriana, 0.80)

IF parotiditis **and**
fiebre=mayor a 37 **and**
dolor de parótidas **and**
crecimiento de parótidas **and**
dolor a la masticación acompañado con borramiento del ángulo mandibular **and**
resultados de laboratorio=biometría hemática leucocitosis
THEN (parotiditis bacteriana, 0.90)

IF ira **and**
fiebre=mayor a 39 **and**
fiebre=continua **and**
fiebre=de 2 a 3 días **and**
rinoreea=transparente **and**
tos=seca **and**
tos=por accesos **and**
dificultad respiratoria=leve a moderada **and**
estrindor=laríngeo **and**
ardor de garganta intenso **and**
odinofagia=de 2 a 3 días **and**
odinofagia=garganta roja **and**
resultados de laboratorio=biometría hemática eosinofilia
THEN (crup espasmódico, 1.0)

IF sarampión

THEN (sintomático y de soporte (antitérmicos),1.0).

IF ruseóla

THEN (sintomático y de soporte (antitérmicos),1.0).

IF escarlatina

THEN (sintomático y de soporte (antitérmicos y antibióticos:penicilina y eritromicina),1.0).

IF varicela

THEN (sintomático y de soporte (antitérmicos),1.0).

IF dengue

THEN (sintomático y de soporte (antitérmicos (excepto aspirina) e hidratación),1.0).

IF tifoidea

THEN (sintomático y de soporte (antitérmicos, hidratación y antibióticos (ampicilina, clorafenicol, trimatropima-sulfametaxazol, cefuroxona)),1.0).

IF paperas

THEN (sintomático y de soporte (antitérmicos e hidratación),1.0).

IF parotiditis bacteriana

THEN (sintomático y de soporte (antitérmicos, hidratación y antibióticos),1.0).

IF crup espasmódico

THEN (sintomático y de soporte (antitérmicos, humidificación, aire frío, sedación y tranquilizar al paciente),1.0).

IF bronquiolitis infecciosa aguda

THEN (sintomático y de soporte (antitérmicos, broncodilatadores, oxigenoterapia y nebulizaciones),1.0).

IF neumonía

THEN (sintomático y de soporte (antitérmicos, antitusígenos, hidratación y antibióticos (penicilina G, eritromicina, amoxicilina)),1.0).

IF otitis viral

THEN (sintomático y de soporte (antitérmicos),1.0).

IF otitis bacteriana

THEN (sintomático y de soporte (antitérmicos),1.0).

APENDICE D:

IMPLEMENTACIÓN DEL SISTEMA *DIAGMED*

(CODIFICADO EN TURBO PROLOG 1.1)

/ Programa: Diagnóstico de Enfermedades Infecciosas Infantiles */*
/ solo tiene 4 enfermedades: */*

domains

enfermedad=string
tipo_diagnostico, signo_sintoma_gg, signo_sintoma_gf = string
valor_certidumbre = real
respuesta = char
resultado_lab, laboratorio = string
terapia= string
expediente=string
nombre=string
edad= integer
sexo=char
email=string
sindrome, nombres_lab = string
seleccion=integer
medico, laboratorista= string
password= string
folio=string
opcion=integer
archcons=string
num= integer

database

db_signo_sintoma_gg(signo_sintoma_gg, respuesta)
db_signo_sintoma_gf(signo_sintoma_gf, respuesta)
db_diagnostico(enfermedad, tipo_diagnostico, valor_certidumbre)
db_resultado_lab(resultado_lab, respuesta)
db_terapia(terapia)
db_personal(expediente, nombre, sexo, edad, email)
db_sindrome(sindrome)
db_posible_enfermedad(enfermedad)
db_sugerencia_lab(nombres_lab)
db_medico(medico)
db_laboratorista(laboratorista)
db_folio(string, nombres_lab)

predicates

diagnostico(enfermedad, tipo_diagnostico, valor_certidumbre)
terapia(enfermedad, terapia)
tratamiento
signo_sintoma_gg(signo_sintoma_gg)
signo_sintoma_gf(signo_sintoma_gf)
resultado_lab(resultado_lab)
diagnostico_clinico_gg
diagnostico_clinico_gf
diagnostico_laboratorio

resultados_diagnostico_clinico_gg
resultados_diagnostico_clinico_gf
resultados_diagnostico_laboratorio
resultados_tratamiento
resultados_sindrome
resultados_sugerencia_lab
resultados_posible_enfermedad
limpia_db
bienvenida
motor
identificacion
proceso1
proceso2
proceso3
proceso4
proceso5
crear_ventana1
crear_ventana2
crear_ventana3
crear_ventana4
crear_ventana5
crear_ventana6
crear_ventana7
crear_ventana8
crear_ventana9
sugerencia_lab(nombres_lab)
sindrome(sindrome)
posible_enfermedad(enfermedad)
posibles_enfermedades
menu
proceso(seleccion)
password_medico(password)
password_laboratorista(password)
password_medico_laboratorista (password)
mostrar_res_lab
letrero_menu
password

goal
motor.

clauses

/* * BASE DE CONOCIMIENTOS *** */**

sindrome("parotiditis):-

 signo_sintoma_gg("fiebre"),
 signo_sintoma_gg("dolor_a_masticacion"),
 signo_sintoma_gg("dolor_de_parotidas"),
 signo_sintoma_gg("crecimiento_de_parotidas").

sindrome("ira):-

 signo_sintoma_gg("fiebre"),
 signo_sintoma_gg("tos"),
 signo_sintoma_gg("dificultad_respiratoria").

sugerencia_lab("biometria_hematica):-

 posible_enfermedad("paperas").

sugerencia_lab("biometria_hematica):-

 posible_enfermedad("parotiditis_bacteriana").

sugerencia_lab("biometria_hematica):-

 posible_enfermedad("bronquiolitis_infecciosa").

sugerencia_lab("exurado_faringeo):-

 posible_enfermedad("neumonia").

posible_enfermedad("paperas):-

 sindrome("parotiditis").

posible_enfermedad("parotiditis_bacteriana):-

 sindrome("parotiditis").

posible_enfermedad("bronquiolitis_infecciosa):-

 sindrome("ira").

posible_enfermedad("neumonia):-

 sindrome("ira").

diagnostico("paperas","clinica",1.0):-

 signo_sintoma_gf("fiebre_continua"),
 signo_sintoma_gf("fiebre_mayor_a_38"),
 signo_sintoma_gf("dolor_de_parotidas"),
 signo_sintoma_gf("crecimiento_de_parotidas"),
 signo_sintoma_gf("dolor_a_masticacion_espontaneo_agudo"),
 signo_sintoma_gf("dolor_a_masticacion_acompañado_con_borramiento_angulo_mandibular").

diagnostico("paperas","laboratorio", 1.0):-

 resultado_lab("biometria_hematica = positiva").

diagnostico("paperas","clinica",0.90):-

 signo_sintoma_gf("fiebre_mayor_a_38"),
 signo_sintoma_gf("dolor_de_parotidas"),
 signo_sintoma_gf("crecimiento_de_parotidas"),

signo_sintoma_gf("dolor_a_masticacion_acompañado_con_borramiento_angulo_mandibular").

diagnostico("paperas","clinica", 0.80):-

 signo_sintoma_gf("fiebre_continua"),
 signo_sintoma_gf("fiebre_mayor_a_38"),
 signo_sintoma_gf("dolor_de_parotidas"),

```

signo_sintoma_gf("crecimiento_de_parotidas"),
signo_sintoma_gf("dolor_a_masticacion_espontaneo_agudo").

diagnostico("parotiditis_bacteriana","clinica",1.0):-
    signo_sintoma_gf("fiebre_continua"),
    signo_sintoma_gf("fiebre_mayor_a_37"),
    signo_sintoma_gf("dolor_de_parotidas"),
    signo_sintoma_gf("crecimiento_de_parotidas"),
    signo_sintoma_gf("dolor_a_masticacion_espontaneo_agudo"),
    signo_sintoma_gf("dolor_a_masticacion_acompañado_con_borramiento_angulo_mandi-
bular"),
    signo_sintoma_gf("odinofagia_de_2_a_3_dias"),
    signo_sintoma_gf("ardor_de_garganta_intenso").

diagnostico("parotiditis_bacteriana","laboratorio",1.0):-
    resultado_lab("biometria_hematica = leucositosis").

diagnostico("parotiditis_bacteriana","clinica",0.9):-
    signo_sintoma_gf("fiebre_mayor_a_37"),
    signo_sintoma_gf("dolor_de_parotidas"),
    signo_sintoma_gf("crecimiento_de_parotidas"),
    signo_sintoma_gf("dolor_a_masticacion_acompañado_con_borramiento_angulo_mandi-
bular"),
    signo_sintoma_gf("odinofagia_de_2_a_3_dias"),
    signo_sintoma_gf("ardor_de_garganta_intenso").

diagnostico("parotiditis_bacteriana","clinica",0.8):-
    signo_sintoma_gf("fiebre_continua"),
    signo_sintoma_gf("fiebre_mayor_a_37"),
    signo_sintoma_gf("dolor_de_parotidas"),
    signo_sintoma_gf("crecimiento_de_parotidas"),
    signo_sintoma_gf("dolor_a_masticacion_espontaneo_agudo"),
    signo_sintoma_gf("odinofagia_de_2_a_3_dias"),
    signo_sintoma_gf("ardor_de_garganta_intenso").

diagnostico("bronquiolitis_infecciosa","clinica",1.0):-
    signo_sintoma_gf("fiebre_continua"),
    signo_sintoma_gf("fiebre_mayor_a_39"),
    signo_sintoma_gf("fiebre_de_2_a_3_dias"),
    signo_sintoma_gf("tos_seca"),
    signo_sintoma_gf("tos_por_accesos"),
    signo_sintoma_gf("dificultad_respiratoria_grave"),
    signo_sintoma_gf("odinofagia_de_2_a_3_dias"),
    signo_sintoma_gf("odinofagia_garganta_roja"),
    signo_sintoma_gf("apnea_periodos_largos"),
    signo_sintoma_gf("ardor_de_garganta_intenso").

diagnostico("bronquiolitis_infecciosa","laboratorio",1.0):-
    resultado_lab("biometria_hematica = leucositosis").

diagnostico("bronquiolitis_infecciosa","clinica",0.8):-
    signo_sintoma_gf("fiebre_continua"),
    signo_sintoma_gf("fiebre_de_2_a_3_dias"),
    signo_sintoma_gf("tos_por_accesos"),
    signo_sintoma_gf("dificultad_respiratoria_grave"),
    signo_sintoma_gf("odinofagia_garganta_roja"),
    signo_sintoma_gf("apnea_periodos_largos"),
    signo_sintoma_gf("ardor_de_garganta_intenso").

diagnostico("neumonia","clinica",1.0):-
    signo_sintoma_gf("fiebre_continua"),
    signo_sintoma_gf("fiebre_mayor_a_39"),
    signo_sintoma_gf("fiebre_de_2_a_3_dias"),

```

```
signo_sintoma_gf("tos_con_flema"),
signo_sintoma_gf("tos_frecuente"),
signo_sintoma_gf("dificultad_respiratoria_grave_o_polipnea_o_tiraje"),
signo_sintoma_gf("hipoventilacion_crepitante_pulmonar").
```

```
diagnostico("neumonia", "laboratorio", 1.0):-
    resultado_lab("exurado_faringeo = estreptococo neumonice o hemofilus
influenza o esqericia coli").
```

```
diagnostico("neumonia", "clinica", 0.90):-
    signo_sintoma_gf("fiebre_continua"),
    signo_sintoma_gf("dificultad_respiratoria_grave_o_polipnea_o_tiraje"),
    signo_sintoma_gf("hipoventilacion_crepitante_pulmonar").
```

```
diagnostico("neumonia", "clinica", 0.85):-
    signo_sintoma_gf("fiebre_mayor_a_39"),
    signo_sintoma_gf("tos_con_flema"),
    signo_sintoma_gf("tos_frecuente"),
    signo_sintoma_gf("dificultad_respiratoria_grave_o_polipnea_o_tiraje"),
    signo_sintoma_gf("hipoventilacion_crepitante_pulmonar").
```

```
terapia("paperas", "sintom tico y de soporte: antit,rmicos e hidrataciϕn").
```

```
terapia("parotiditis_bacteriana", "sintom tico y de soporte: antit,rmicos, hidrataciϕn y
antibiϕticos").
```

```
terapia("bronquiolitis_infecciosa", "sintom tico y de soporte: antit,rmicos,
broncodilatadores, oxigenoterpia y nebulizaciones").
```

```
terapia("neumonia", "sintom tico y de soporte: antit,rmicos, antitusigenos, hidrataciϕn y
antibiϕticos (penicilina G, eritromicina, amoxicilina)").
```

```
password_laboratorista("clavelaboratorista").
```

```
password_medico_laboratorista("clavemedico").
```

```
password_medico_laboratorista ("clavelaboratorista").
```

```
password_medico("clavemedico").
```

```
password:- password_medico("clavemedico"),
password_laboratorista("clavelaboratorista").
```

```
/* *** MECANISMO DE INFERENCIA *** */
```

```
motor:-
    identificacion,
    menu.
```

```
menu:-
    letrero_menu,
    readint(Seleccion),
    proceso(Seleccion),
    menu.
```

```
proceso1:-
    crear_ventana4,
    limpia_db,
    nl, write("Introduce los siguientes datos:"),nl,
    write("Expediente: "),
    readln(X),
```

```

write("Nombre: "),
readln(N),
write("Sexo (f/m): "), nl,
readchar(S),
write("Edad: "),
readint(D),
write("email: "),
readln(M),
assertz(db_personal(X,N,S,D,M)),
nl, write("Dame iniciales del paciente: "),
readln(Firstname),
save(Firstname),
nl, write("Teclea para ir al menu"),
readchar(_),
removewindow.

```

proceso2:-

```

crear_ventana5,
limpia_db,
diagnostico_clinico_gg,
resultados_diagnostico_clinico_gg,
diagnostico_clinico_gf,
resultados_diagnostico_clinico_gf,!,
nl, nl, write("Iniciales del paciente y numero de consulta: "),
readln(FilenameNum),
nl, write("Se enviar... la petición al laboratorio con el folio: ", FilenameNum),
sugerencia_lab(SL),
assertz(db_folio(FilenameNum,SL)),
nl, write("Dame tu nombre médico: "),
readln(ME),
assertz(db_medico(ME)),
save(FilenameNum),
nl, nl, write("Teclea para ir al menu"),
readchar(_),
limpia_db,
removewindow.

```

proceso3:-

```

crear_ventana6,
limpia_db,
nl, write("Ingresa folio:"),
readln(FilenameNum),
consult(FilenameNum),
db_folio(FilenameNum,SL),
nl, write("Folio: ", FilenameNum),
nl, write("Peticion: " , SL),
diagnostico_laboratorio,
resultados_diagnostico_laboratorio,
nl,nl, write("Dame tu nombre laboratorista: "),
readln(LA),
assertz(db_laboratorista(LA)),
save(FilenameNum),
nl, nl, write("Teclea para ir al menu"),
readchar(_),
removewindow,
menu.

```

proceso4:-

```

crear_ventana7,
limpia_db,
nl, write("Dame iniciales de paciente y numero de consulta: "),
readln(FilenameNum),

```

```

consult (FilenameNum),
db_diagnostico(E,"clinica",FC_C),
nl, write("DIAGNOSTICO DEL PACIENTE"),
nl, write("Diagnóstico clínico: ", E),nl,
write("Certeza clínica: ",FC_C),nl,
db_medico(ME),
write("Médico: ", ME),nl,
mostrar_res_lab,
tratamiento,
resultados_tratamiento!,
save(FilenameNum),
limpia_db,
nl, nl, write("Teclea para ir al menu"),
readchar(_),
removewindow.

```

proceso5:-

```

crear_ventana7,
limpia_db,
nl, write("Dame iniciales del paciente: "),
readln(Filename),
consult (Filename),
db_personal(X,N,S,D,M),
nl,nl, write("DATOS DEL PACIENTE:"),nl,
write("Expediente: ", X),nl,
write("Nombre: ", N),nl,
write("Sexo: ",S),nl,
write("Edad: ",D),nl,
write("email: ",M),nl,
nl, write("Teclea para ir al menu"),
readchar(_),
removewindow.

```

limpia_db :-

```
retract(_),fail.
```

limpia_db.

signo_sintoma_gg(SSGG) :-

```

db_signo_sintoma_gg(SSGG,Resp),
Resp='s';
not(db_signo_sintoma_gg(SSGG,_)),
nl, write("El paciente tiene ", SSGG, " (s/n)? "),
readchar(Resp),
write(Resp),
assertz(db_signo_sintoma_gg(SSGG,Resp)),
Resp='s'.

```

signo_sintoma_gf(SSGF) :-

```

db_signo_sintoma_gf(SSGF,Resp),
Resp='s';
not(db_signo_sintoma_gf(SSGF,_)),
nl, write("El paciente tiene ", SSGF, " (s/n)? "),
readchar(Resp),
write(Resp),
assertz(db_signo_sintoma_gf(SSGF,Resp)),
Resp = 's'.

```

resultado_lab(RL):-

```

db_resultado_lab(RL,Resp),
Resp='s';
not(db_resultado_lab(RL,_)),

```

```
nl, write("Sus resultados de laboratorio son: ", RL, " (s/n)? "),
readchar(Resp),
write(Resp),
assertz(db_resultado_lab(RL,Resp)),
Resp='s'.
```

tratamiento:-

```
db_diagnostico(E,"clinica",FC_C),
FC_C<0.9, FC_C>=0.85,
db_diagnostico(E,"laboratorio",FC_L),
FC_L=1.0,
terapia(E,P),
assertz(db_terapia(P));
db_diagnostico(E,"clinica",FC_C),
FC_C>=0.9,
not(db_diagnostico(E,"laboratorio", _)),
terapia(E,P),
assertz(db_terapia(P));
db_diagnostico(E,"clinica",FC_C),
FC_C>=0.9,
db_diagnostico(E,"laboratorio",FC_L),
FC_L=1.0,
terapia(E,P),
assertz(db_terapia(P)).
```

tratamiento.

posibles_enfermedades:-

```
posible_enfermedad(E),
assertz(db_posible_enfermedad(E)),
fail.
```

posibles_enfermedades.

diagnostico_clinico_gg:-

```
nl, nl, write("Dame datos clínicos de grano grueso "),
sindrome(X),
assertz(db_sindrome(X)),
/*sugerencia_lab(SL),
assertz(db_sugerencia_lab(SL)),*/
posibles_enfermedades.
```

diagnostico_clinico_gg.

diagnostico_clinico_gf:-

```
nl,nl, write("Dame signos y síntomas de grano fino"),
db_posible_enfermedad(E),
diagnostico(E, "clinica", FC_C),
assertz(db_diagnostico(E, "clinica", FC_C)),
fail.
```

diagnostico_clinico_gf.

diagnostico_laboratorio:-

```
nl, write("Dame tus resultados de los estudios de laboratorio"),
db_posible_enfermedad(E),
diagnostico(E, "laboratorio", FC_L),
assertz(db_diagnostico(E, "laboratorio", FC_L)),
fail.
```

diagnostico_laboratorio.

/*INTERFAZ DEL USUARIO */

identificacion:-

```
    crear_ventana1,  
    crear_ventana9,  
    nl, write ("Password como médico o laboratorista: "),  
    readln(Password),  
    password_medico_laboratorista (Password),  
    bienvenida;  
    nl, write ("PASSWORD INCORRECTO"), nl,  
    nl, write ("Volver a introducir el password?, teclea i "),  
    nl, write ("Salir del programa?, teclea s "), nl,  
    readln(Resp),  
    Resp = "i",  
    identificacion;  
    Resp="s",  
    exit.
```

bienvenida:-

```
    crear_ventana2,  
    nl, write("***** BIENVENIDO AL SISTEMA EXPERTO *****"),  
    nl, write("Te ayudamos a diagnosticar las siguientes enfermedades"),  
    nl, write ("          infecciosas infantiles: "),  
        nl, nl, write ("rubeola, sarampión, roseola, escarlatina, varicela,"),  
        nl, write ("dengue, tifoidea, paperas, parotiditis bacteriana,"),  
    nl, write ("crup espasmódico, bronquiolitis infecciosa, neumonía,"),  
    nl, write ("otitis viral y otitis bacteriana."),  
    nl,nl, write("          Presione una tecla para iniciar"),  
    readchar(_),  
    removewindow.
```

letrero_menu:-

```
    removewindow,  
    crear_ventana3,  
    nl, write("MENU PRINCIPAL"), nl,  
    write("1.- Insertar datos personales del paciente"), nl,  
    write("2.- Realizar un diagnóstico clínico"), nl,  
    write("3.- Realizar un diagnóstico de laboratorio"), nl,  
    write("4.- Obtener resultados del diagnóstico integral"), nl,  
    write("5.- Visualizar datos de un paciente (datos personales)"),nl,  
    write("6.- Salir"),nl,nl,  
    write("Selecciona una opción: ").
```

proceso(1):-

```
    password_medico("clavemedico"),  
    proceso1;  
    nl, write ("No tienes permiso para esta opcion"),  
    readchar(_),  
    menu.
```

proceso(2):-

```
    password_medico("clavemedico"),  
    proceso2;  
    nl, write ("No tienes permiso para esta opcion"),  
    readchar(_),  
    menu.
```

proceso(3):-

```
    password_laboratorista("clavemedico"),  
    proceso3;  
    nl, write ("No tienes permiso para esta opción"),
```

```
readchar(_),  
menu.
```

```
proceso(4):-  
password_medico("clavemedico"),  
proceso4;  
nl, write ("No tienes permiso para esta opcion"),  
readchar(_),  
menu.
```

```
proceso(5):-  
password,  
proceso5;  
nl, write ("No tienes permiso para esta opción"),  
readchar(_),  
menu.
```

```
proceso(6):-  
crear_ventana8,  
nl, nl, write("***** ADIOS---, Hasta la próxima *****"),  
sound(4000,392), sound(4000,440), sound(4000,494), sound(8000,440),  
sound(4000,494), sound(12000,392),sound(4000,494), sound(12000,392),  
removewindow,  
exit.
```

```
resultados_tratamiento:-  
db_terapia(P),  
write("TERAPIA: ", P).
```

```
resultados_tratamiento:-  
not(db_terapia(_)),  
nl, write("Lo siento no puedo darte terapia").
```

```
resultados_tratamiento.
```

```
resultados_diagnostico_clinico_gg:-  
resultados_sindrome,  
/*resultados_sugerencia_lab,*/  
resultados_posible_enfermedad.
```

```
resultados_diagnostico_clinico_gg.
```

```
resultados_sindrome:-  
db_sindrome(S),  
nl,nl, write("SINDROME ASOCIADO: ", S).
```

```
resultados_sindrome:-  
not(db_sindrome(_)),  
nl, write("No hay síndrome diagnosticado").
```

```
resultados_sindrome.
```

```
resultados_sugerencia_lab:-  
db_sugerencia_lab(SL),  
nl,nl, write("SUGERENCIA: ", SL).
```

```
resultados_sugerencia_lab:-  
not(db_sugerencia_lab(_)),  
nl, write("No hay sugerencia de estudios de laboratorio").
```

```
resultados_sugerencia_lab.
```

```

resultados_posible_enfermedad:-
  db_posible_enfermedad(E),
  nl, write("POSIBLE ENFERMEDAD: ", E),
  fail.

resultados_posible_enfermedad:-
  not(db_posible_enfermedad(_)),
  nl, write("No hay varias enfermedades asociadas a este sindrome").

resultados_posible_enfermedad.

resultados_diagnostico_clinico_gf :-
  db_diagnostico(E,"clinica",FC_C),
  FC_C >= 0.9,
  nl,nl, write("RESULTADOS DEL DIAGNOSTICO REALIZADO: "), nl,
  write("Diagnóstico: ", E),nl,
  write("Tipo de diagnóstico: clinico"),nl,
  write("Certeza: ", FC_C), nl,
  write("Sugerencia: No es necesario realizar estudios de laboratorio");
  db_diagnostico(E,"clinica",FC_C),
  FC_C < 0.9,
  sugerencia_lab(SL),
  assertz(db_sugerencia_lab(SL)),
  nl,nl, write("RESULTADOS DEL DIAGNOSTICO REALIZADO: "), nl,
  write("Diagnóstico: ", E),nl,
  write("Tipo de diagnóstico: clinico"),nl,
  write("Certeza: ", FC_C), nl,
  write("Sugerencia: Realizar estudios de laboratorio de " , SL).

resultados_diagnostico_clinico_gf:-
  not(db_diagnostico(_, "clinica", _)),
  nl,nl, write("Lo siento, no ha sido posible obtener el diagnóstico"),nl.

resultados_diagnostico_clinico_gf.

resultados_diagnostico_laboratorio :-
  db_diagnostico(E,"laboratorio",FC_L),
  nl,nl, write("RESULTADOS DEL DIAGNOSTICO REALIZADO: "), nl,
  write("Diagnóstico: ", E),nl,
  write("Tipo de diagnóstico: laboratorio"),nl,
  write("Certeza: ",FC_L).

resultados_diagnostico_laboratorio:-
  not(db_diagnostico(_, "laboratorio", _)),
  nl,nl, write("Lo siento, no ha sido posible obtener el diagnóstico"),nl.

resultados_diagnostico_laboratorio.

mostrar_res_lab:-
  db_diagnostico(E,"laboratorio",FC_L),
  nl, write ("Diagnóstico de laboratorio: ", E),nl,
  write("Certeza laboratorio: ", FC_L),nl,
  db_laboralista(LA),
  write("Laboralista: ", LA),nl,nl;
  not(db_diagnostico(_, "laboratorio", _)),
  nl, write("No se realizó diagnóstico de laboratorio"),nl,nl.

crear_ventana1:-
  makewindow(1,112,7," ", 0,0,24,79),
  gotowindow(1),
  clearwindow.

```

```
crear_ventana2:-
    makewindow(2,30,7,"BIENVENIDA", 7,12,15,60),
    gotowindow(2),
    clearwindow.

crear_ventana3:-
    makewindow(3,30,7,"MENU", 7,12,15,60),
    gotowindow(3),
    clearwindow.

crear_ventana4:-
    makewindow(4,30,7,"INSERTAR DATOS DEL PACIENTE", 2,3,20,73),
    gotowindow(4),
    clearwindow.

crear_ventana5:-
    makewindow(5,30,7,"DIAGNOSTICO CLINICO", 2,3,20,73),
    gotowindow(5),
    clearwindow.

crear_ventana6:-
    makewindow(6,30,7,"DIAGNOSTICO DE LABORATORIO", 2,3,20,73),
    gotowindow(6),
    clearwindow.

crear_ventana7:-
    makewindow(7,30,7,"VISUALIZAR A UN PACIENTE", 2,3,20,73),
    gotowindow(7),
    clearwindow.

crear_ventana8:-
    makewindow(8,30,7,"DESPEDIDA", 7,12,15,60),
    gotowindow(8),
    clearwindow.

crear_ventana9:-
    makewindow(9,30,7,"IDENTIFICACION", 8,15,14,50),
    gotowindow(9),
    clearwindow.
```