



UNIVERSIDAD POLITÉCNICA DE VALENCIA  
DEPARTAMENTO DE SISTEMAS INFORMÁTICOS Y COMPUTACIÓN  
INTELIGENCIA ARTIFICIAL, RECONOCIMIENTO DE FORMAS E  
IMAGEN DIGITAL

# Solución para el desarrollo de Sistemas de Ayuda a la Decisión para diagnóstico clínico

TESIS DE MÁSTER

Presentada por **Carlos Sáez Silvestre**

Dirigida por **Dr. Juan Miguel García-Gómez**

Valencia - España  
30 de noviembre de 2009



## **Agradecimientos**

El trabajo realizado en esta tesina no habría sido posible sin toda la ayuda y conocimiento aportado por mis compañeros de investigación de la línea de Minería de Datos Biomédicos del grupo IBIME Javier Vicente, Salvador Tortajada, Miguel Esparza, Alfredo Navarro, y Elías Fuster.

Quiero dar mi especial agradecimiento por el apoyo, confianza, e inspiración aportado a lo largo de estos años al Profesor Juan Miguel García-Gómez y a la Profesora Montserrat Robles Viejo.



## Resumen

Los Sistemas de Ayuda a la Decisión Clínica (CDSS) son sistemas computacionales que aportan conocimiento específico para la ayuda a la decisión en el diagnóstico, pronóstico, tratamiento o administración de pacientes. Los actuales CDSS son juzgados por su flexibilidad a la hora de incorporar información de diversas fuentes y su adaptabilidad a diferentes escenarios de aplicación.

Los objetivos de esta tesina son el diseño y desarrollo de CDSS basados en aprendizaje automático, que sean flexibles para su adaptación a diversos escenarios de diagnóstico y pronóstico clínico, de fácil integración en los sistemas sanitarios, que faciliten el mantenimiento del conocimiento empleado en la toma de decisiones, y que sean construidos de forma escalable y robusta. El trabajo se divide en los siguientes apartados:

- (1) El desarrollo de un sistema distribuido de ayuda a la decisión (dDSS) basado en agentes, en el marco del proyecto europeo del VI Programa Marco HealthAgents, que permite la ejecución distribuida de clasificadores y la recopilación de datos de diferentes fuentes para el entrenamiento de nuevos y actualizados modelos.
- (2) El desarrollo de un motor de clasificación capaz de incluir diferentes técnicas de reconocimiento de formas permitiendo ejecutar modelos predictivos desde un lenguaje XML propio.
- (3) El desarrollo de un CDSS adaptable a diferentes tareas de diagnóstico en forma de acceso a datos, interfaz gráfica de usuario y funcionalidades específicas del dominio de aplicación: Curiam.

Los tres estudios proporcionan en conjunto una solución reutilizable para el desarrollo de nuevos CDSS basados en reconocimiento de formas para el diagnóstico clínico. En concreto, se han desarrollado tres versiones de Curiam para DSS clínicos: Curiam BT, para el diagnóstico de tumores cerebrales; Curiam STT, para el diagnóstico de tumores de partes blandas; y Curiam PPD, para la predicción de depresión postparto. Finalmente, el trabajo llevado a cabo en esta tesina, individualmente o como parte de una colaboración, ha dado lugar a diversas publicaciones científicas en congresos, un capítulo de libro y una publicación en revista.



# Glosario

## Acrónimos y abreviaturas

<b>API</b>	application program interface
<b>CDSS</b>	Clinical Decision Support System
<b>CF</b>	Classification Framework
<b>DSS</b>	Decision Support System
<b>dDSS</b>	distributed Decision Support System
<b>DSS</b>	Decision Support System
<b>GUI</b>	Graphical User Interface
<b>HR-MAS</b>	High Resolution Magic-Angle Spectra
<b>IA</b>	Inteligencia Artificial
<b>ICA</b>	independent component analysis
<b>KNN</b>	k-nearest neighbour
<b>LDA</b>	linear discriminant analysis
<b>LS-SVM</b>	least square support vector machines
<b>ML</b>	Machine Learning
<b>RM</b>	Resonancia Magnética
<b>MRS</b>	Magnetic Resonance Spectroscopy
<b>PR</b>	Pattern Recognition
<b>PCA</b>	principal component analysis
<b>SVM</b>	support vector machines
<b>SOC</b>	Sistema de Orientación Clínica
<b>SV</b>	Single Voxel
<b>TE</b>	tiempo de eco
<b>XML</b>	eXtensible Markup Language





# Índice general

<b>Índice general</b>	<b>9</b>
<b>1. Introducción</b>	<b>11</b>
1.1. Motivación . . . . .	11
1.2. Reconocimiento de formas . . . . .	12
1.3. Objetivos, proyectos, y contribuciones . . . . .	15
<b>2. DSS distribuido basado en agentes en HealthAgents</b>	<b>19</b>
2.1. El framework de agentes en HEALTHAGENTS . . . . .	20
2.2. Diseño del sistema de clasificación distribuido . . . . .	21
2.3. Recolección de datos distribuida . . . . .	26
2.4. GUI de resultados de clasificación . . . . .	27
2.5. Resultados del proyecto . . . . .	28
<b>3. Classification Framework</b>	<b>33</b>
3.1. Motivación . . . . .	33
3.2. Diseño del framework . . . . .	34
3.3. Lenguaje de especificación de clasificadores basado en XML . . . . .	38
3.4. Entrada y salida XML . . . . .	47
3.5. Resultados . . . . .	49
<b>4. Curiam DSS</b>	<b>51</b>
4.1. Motivación . . . . .	51
4.2. Diseño arquitectónico . . . . .	51
4.3. Acceso a datos . . . . .	54
4.4. Sistema de clasificación . . . . .	55
4.5. Interfaz gráfica de usuario . . . . .	56
4.6. Mapeo de resultados de clasificación a ontología . . . . .	61
4.7. Implementaciones de Curiam para DSS clínicos . . . . .	67
<b>5. Conclusiones y trabajo futuro</b>	<b>77</b>
<b>Bibliografía</b>	<b>79</b>



# Capítulo 1

## Introducción

### 1.1. Motivación

Los sistemas de ayuda a la decisión clínica, o Clinical Decision Support System (CDSS), son sistemas computacionales que aportan conocimiento específico para la ayuda a la decisión en el diagnóstico, pronóstico, tratamiento o administración de pacientes. Las funcionalidades de los CDSS incluyen la alerta y/o recordatorio de situaciones de salud de pacientes concretos o epidemiológicos, asistencia al diagnóstico, pronóstico, interpretación de señales biomédicas, planificación y/o control de terapias, calidad asistencial, riesgo, coste temporal, planificación de recursos, control de calidad en la adquisición de biobancos multicéntricos y búsqueda de patrones anómalos. El uso de los sistemas de ayuda a la decisión médica pretende potenciar las habilidades médicas en el manejo de múltiples variables biomédicas de los pacientes, y nunca sustituir a los profesionales cualificados.

Los CDSS utilizan características de pacientes individuales con el propósito de generar valoraciones o recomendaciones específicas del paciente, que son presentadas a los médicos [21]. Han sido muchos los CDSS desarrollados para ayudar en la toma de decisiones en problemas médicos específicos, por ejemplo para la detección y diagnóstico de cáncer de mama [26], gastrointestinal [29, 56], hematológico [12, 33, 50], oral [31], de pulmón [5], de vejiga [44], de próstata [24], o de tumores cerebrales [47]. Sin embargo, la implementación de un CDSS genérico y multi-propósito, que cubra varios de estos problemas, que esté basado en la evidencia y aplicable tanto en asistencia primaria como especializada, está todavía en su fase de investigación [52].

Para una adecuada implantación de los CDSS en los entornos clínicos es deseable un comportamiento adaptado al problema, basado en la evidencia, referenciado, interoperable, ajustado al proceso asistencial, fidedigno y actualizado. Desde los años 80 [42] se han desarrollado y analizado una gran variedad de CDSS enfocados a distintos problemas médicos que cumplen en mayor o menor medida las funcionalidades expuestas anteriormente. Los CDSS han sido catalogados por la *National Electronic Health Records Taskforce Report* en 4 categorías según la elaboración del conocimiento específico del paciente: en el nivel I se engloban los CDSS que ofrecen información relacionada con el proceso de decisión, en el nivel II los CDSS presentan alertas que deben ser revisadas por los expertos, en el nivel III, los CDSS son capaces de realizar deducciones sobre una base de reglas, y por último, en el nivel IV, los CDSS están basados en modelos predictivos inferidos a partir de casos del problema a resolver, generalmente basados en modelos probabilísticos

y/o lógica difusa.

La mayoría de los CDSS de nivel III y IV están basados en técnicas de Inteligencia Artificial (IA). Dentro de la IA, el aprendizaje automático o Machine Learning (ML) proporciona soluciones computacionales basadas en bancos de datos masivos y ruidosos [3]. Esta propiedad ha hecho que los CDSS basados en ML hayan sido ampliamente aceptados en la ayuda a la decisión médica, especialmente en tareas de diagnóstico, debido a su capacidad para la optimización, flexibilidad, precisión para la inferencia predictiva e interpretabilidad [14, 25] en decisiones complejas. Las mejoras actuales en CDSS pasan por dos aspectos fundamentales [10]. El primero de ellos es la integración de los CDSS en los sistemas sanitarios, concretamente, mediante su integración en los flujos de trabajo, el acceso a los repositorios de datos mediante estándares de información e interoperabilidad y finalmente la propia aceptación y motivación de los clínicos para incorporar el sistema en su proceso de decisión. El segundo aspecto consiste en el mantenimiento del conocimiento empleado por el CDSS. El mantenimiento del conocimiento es debido principalmente a 1) las variaciones temporales en las muestras o datos de pacientes, lo que obliga a mantener actualizada la precisión de los modelos y 2) los cambios en el conocimiento del dominio debidos a nuevos avances en el conocimiento médico o en las técnicas de ML, que llevarán a la actualización de los modelos del CDSS, la incorporación de nuevos de ellos o incluso la posibilidad de añadir nuevas funcionalidades al sistema. Se propuso que la próxima generación de CDSS no fuera únicamente basada en la evidencia, sino que se adaptara a ella, con actualizaciones continuas y automáticas para reflejar los más recientes avances de la ciencia clínica y práctica local [43]. Los actuales CDSS serán juzgados por su flexibilidad a la hora de incorporar información de diversas fuentes y su adaptabilidad a diferentes escenarios de aplicación [32].

## 1.2. Reconocimiento de formas

El objetivo del Pattern Recognition (PR) es inferir de forma automática regularidades en conjuntos de datos, de forma que a la llegada de un nuevo objeto se pueda indicar su pertenencia a una determinada clase o categoría [3]. El aprendizaje automático proporciona al PR las herramientas para inferir el conocimiento adquirido de conjuntos de datos a un modelo formal capaz de resolver determinados problemas. El proceso de un problema de PR está dividido en dos fases (figura 1.1): 1) la adaptación o entrenamiento del modelo al conjunto de datos y 2) el reconocimiento o clasificación de una nueva muestra. Una vez entrenado un modelo, éste podrá ser incluido en un Decision Support System (DSS) permitiendo la clasificación de nuevos casos. La regla de decisión de Bayes [9], establece el clasificador de mínimo error como aquel que asigna una muestra  $\mathbf{x}$  a la clase  $c$  que maximiza la probabilidad a posteriori  $p(c | \mathbf{x})$ , por lo tanto:

$$\hat{c} = \arg \max_c p(c)p(\mathbf{x}|c), \quad (1.1)$$

En la práctica, el problema es aproximar las funciones de probabilidad, o discriminantes, a partir de la observación de un número finito de muestras. Esta estimación se lleva a cabo mediante diferentes métodos, basados o no en estadística, que darán lugar a los

correspondientes modelos predictivos. A continuación se describen los métodos utilizados en el desarrollo de esta tesina:

- LDA: Dado un conjunto de muestras, el objetivo de linear discriminant analysis (LDA) [11] es encontrar una proyección que minimice la varianza intra-clase y maximice la separación entre las proyecciones de las medias de cada clase. La solución propuesta por Fisher se reduce al ratio de las diferencias entre las medias proyectadas y la medida de dispersión de cada clase.
- KNN: Es un método de aprendizaje no paramétrico basado en instancias [6]. Basado en una métrica y dado un punto en un espacio de características, asigna la etiqueta de clase más frecuente entre las  $k$  muestras de entrenamiento más cercanas a este punto. El caso especial que la clase predecida corresponde a la clase de la muestra más cercana (por ejemplo, cuando  $k = 1$ ) es denominado clasificador por vecino más próximo (NN).
- ANN: Las redes neuronales artificiales empleadas son denominadas perceptron multicapa. Consisten en una red de unidades simples o neuronas llamadas perceptron [34]. El perceptron computa una única salida a partir de una combinación lineal de sus entradas y los pesos de las conexiones entre cada par de unidades, y posteriormente genera la salida a través de una función de activación lineal o no-lineal. Las redes neuronales perceptrón multicapa son generalmente usadas en problemas de aprendizaje supervisado, los cuales pueden ser resueltos mediante el algoritmo de retropropagación [35] basado en descenso por gradiente.
- LS-SVM: Las support vector machines (SVM) son una técnica no lineal para la clasificación y regresión de funciones definida en el contexto de la teoría estadística del aprendizaje y la minimización estructural del riesgo. Las SVM están basadas en el concepto de hiperplanos definiendo fronteras de decisión que separen muestras pertenecientes a diferentes clases. Esta técnica basada en *kernels* mapea las muestras originales a un espacio de características de alta dimensionalidad donde se encuentra un hiperplano separador de máximo margen. La construcción explícita del mapeo a alta dimensionalidad se evita usando el método del *kernel*. Las least square support vector machines (LS-SVM) [45] introducen un término de mínimos cuadrados en la función de coste de las SVM. Esto hace que el problema consista en resolver un conjunto de ecuaciones lineales.

Además de los modelos de clasificación, en ocasiones es necesario aplicar un procedimiento de pre-proceso para normalizar las muestras de datos de forma que sean comparables y puedan ser utilizadas para el entrenamiento del modelo y posteriormente para la fase de reconocimiento.

PCA: Es una técnica de extracción de características que permite reducir la dimensionalidad de conjuntos de datos [20]. PCA es una transformación lineal ortogonal que mapea los datos a un nuevo sistema de coordenadas de forma que la mayor variancia de entre cualquier proyección de los datos tiene su correspondencia en la primera coordenada (llamada primera componente principal), la segunda mayor variancia en la segunda coordenada, y así sucesivamente.

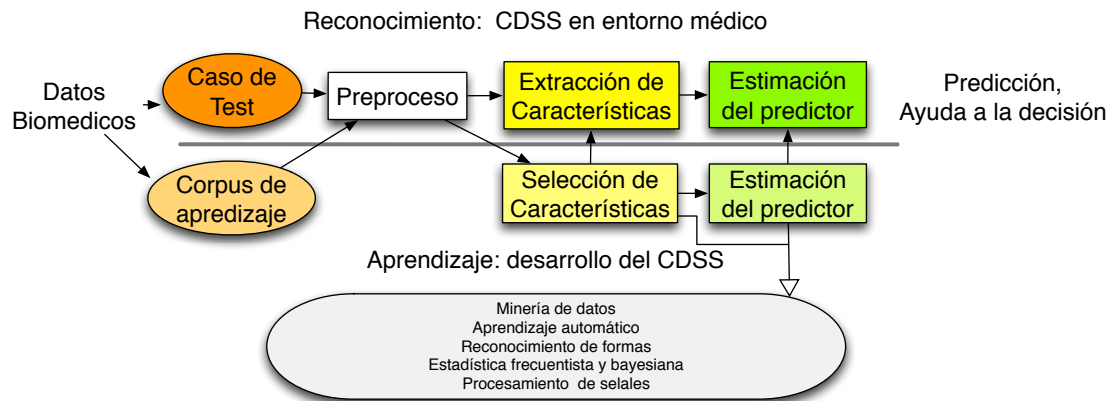


Figura 1.1: Proceso de un problema de PR. El proceso está dividido en dos partes: 1) el aprendizaje o entrenamiento del modelo al conjunto de datos y 2) el reconocimiento o clasificación de una nueva muestra.

PCA puede usarse para reducir la dimensionalidad de un conjunto de datos — manteniendo aquellas características del conjunto de datos que contribuyen en mayor medida a su varianza — tomando los componentes principales de orden menor e ignorando los de orden mayor.

Codificación: También llamada numerización, es utilizada para convertir una variable discreta o nominal en numérica. La codificación es necesaria cuando se emplean técnicas de clasificación que sólo soportan datos de entrada numéricos, por ejemplo, técnicas basadas en distancias.

Peak Integration: La integración de picos, o *peak integration*, permite extraer la intensidad en un determinado rango en una señal por medio de una aproximación de la integral de dicho rango. En determinados tipos de señales, la estimación precisa de las integrales de picos es complicada debido a diversos factores, como la línea de base no negativa, solapamiento de picos, ruido, o la naturaleza no discreta del espectro. *Peak integration* emplea la regla trapezoidal para aproximar dicha integral.

ICA: Técnica que extrae componentes estadísticamente independientes de un conjunto de señales [4, 22, 23]. El método permite extraer señales puras de un conjunto de mezclas.

Para una comparación concisa de varios modelos de clasificación, no es suficiente con obtener la clase asignada a la muestra clasificada, o incluso con obtener el orden de asignación de las diferentes clases. Se hace necesario un índice que permita comparar por igual los resultados de clasificación de varios modelos independientemente de las técnicas de PR empleadas. La regla de decisión de Bayes, asigna una muestra a la clase que maximiza la probabilidad a posteriori. Las técnicas de clasificación, estiman estas probabilidades a posteriori mediante diferentes aproximaciones. Sin embargo, estas aproximaciones no siempre obtienen una estimación de la probabilidades a posteriori, cuyos valores se encuentren en-

tre 0 y 1, y sumen en conjunto 1. Generalmente, este desajuste puede resolverse empleando la función *softmax* [2]:

$$p(c|\mathbf{x}) = \frac{e^{p_c}}{\sum_{i \neq c} e^{p_i}}, \quad (1.2)$$

donde  $p_i$  corresponde al valor de la aproximación de probabilidad a posteriori asignada a la clase  $i$ .

De esta forma, se consigue un suavizado que cumple los requisitos de probabilidad a posteriori, y permite la comparación de dichas probabilidades entre diferentes modelos. Este método se ha empleado en todas las técnicas de clasificación utilizadas en esta tesina, a excepción de la técnica de k-nearest neighbour (KNN). La técnica de KNN ya obtiene de por sí una [6] aproximación de las probabilidades a posteriori, asignando a cada clase el ratio entre el número de vecinos  $N_c$  de dicha clase entre el total de vecinos más próximos  $N$ :  $\frac{N_c}{N}$ . Sin embargo, según el número de vecinos más próximos,  $k$ , obtenido en el entrenamiento de la técnica, las aproximaciones de probabilidad a posteriori pueden no ser representativas respecto al volumen total del espacio de representación de las muestras. Esto ocurre cuando  $k$  tiene valores bajos, según el número de clases totales. Por ejemplo, en el caso más extremo de  $k = 1$  — clasificación por vecino más próximo —, las probabilidades a posteriori siempre serán de 1 para la clase ganadora y 0 para el resto. Esto puede ocasionar que modelos basados en KNN combinados con modelos basados en otras técnicas, causen un desajuste a la hora de calcular probabilidades a posteriori comunes, por ejemplo calculando las medias. Para solucionar este posible problema, las probabilidades a posteriori en la técnica de KNN se han suavizado mediante la siguiente aproximación basada en distancias:

$$p(c|\mathbf{x}) = \frac{\sum_{n=1: \mathbf{k}_n \in c}^N \frac{1}{d(\mathbf{x}, \mathbf{k}_n)}}{\sum_{n=1: \mathbf{k}_n}^N \frac{1}{d(\mathbf{x}, \mathbf{k}_n)}}, \quad (1.3)$$

donde  $N$  corresponde al número total de vecinos más próximo,  $\mathbf{k}_n$  al vecino más próximo de orden  $n$ , y  $d(\mathbf{x}, \mathbf{k}_n)$  a la distancia entre la muestra clasificada  $\mathbf{x}$  y el vecino  $\mathbf{k}_n$ . De este modo, todos los modelos incluidos en los desarrollos de esta tesina, proporcionan resultados de clasificación comparables.

## 1.3. Objetivos, proyectos, y contribuciones

### 1.3.1. Objetivos

Los objetivos de esta tesina son el diseño y desarrollo de CDSS basados en aprendizaje automático, que sean flexibles para su adaptación a diversos escenarios de diagnóstico y pronóstico clínico, de fácil integración en los sistemas sanitarios y que faciliten el mantenimiento del conocimiento empleado en la toma de decisiones. En este estudio, se ha prestado especial atención a los siguientes aspectos:

- el diseño de un motor de clasificación capaz de incluir diferentes técnicas de PR permitiendo ejecutar modelos predictivos desde un lenguaje XML propio;

- el diseño de un sistema basado en agentes para la ejecución y búsqueda distribuida de clasificadores y la recopilación de datos de diferentes fuentes para el entrenamiento de nuevos y actualizados modelos; y
- el diseño de un CDSS adaptable a diferentes tareas de diagnóstico en forma de acceso a datos, interfaz gráfica de usuario y funcionalidades específicas del dominio de aplicación.

Para demostrar la aplicabilidad de los desarrollos llevados a cabo en esta tesina, se han abordado varios problemas de diagnóstico y pronóstico médico pertenecientes a dominios y con requisitos diferentes:

- diagnóstico y pronóstico de tumores cerebrales mediante espectroscopía de resonancia magnética o Magnetic Resonance Spectroscopy (MRS);
- clasificación de grupos histopatológicos y caracteres de tumores de partes blandas a partir de hallazgos de imagen médica; y
- detección de depresión postparto en pacientes que acaban de dar a luz a partir de datos clínicos, genéticos y socioculturales.

### 1.3.2. Contribuciones

El trabajo llevado a cabo en esta tesina ha dado lugar a varias publicaciones científicas en los ámbitos de Informática, Informática Médica y Resonancia Magnética en Medicina.

Un primer prototipo de DSS para el diagnóstico de tumores cerebrales fue presentado en [16] como parte de los resultados de investigación sobre diagnóstico de tumores cerebrales.

La mayoría de las contribuciones se han desarrollado en el marco del proyecto del VI programa marco HEALTHAGENTS. Como resultado final del desarrollo en PR en el proyecto, el diseño del motor de clasificación genérico utilizado en el sistema de clasificación basado en agentes ha sido aceptado para publicación [40] en la revista *The Knowledge Engineering Review*, prevista para 2010. Dicha revista consta en el JCR<sup>1</sup> 2008 con un factor de impacto de 1.588 y un factor de impacto de 5 años de 2.763. A su vez está indexada en la primera versión del ranking de revistas CORE<sup>2</sup> 2008 con calificación C. Por otra parte, se colaboró en un capítulo del libro *Agent-Technology and e-Health* (2008) [27], donde se describió el desarrollo del distributed Decision Support System (dDSS) de HEALTHAGENTS. Dentro del mismo proyecto se han realizado colaboraciones en otras publicaciones a congreso referentes al diseño del sistema de seguridad para sistemas sanitarios desarrollado en el proyecto [54, 55] y a la representación del conocimiento en el sistema de HEALTHAGENTS[7].

El CDSS Curiam (capítulo 4) ha dado lugar a dos contribuciones en el Congreso Europeo de Resonancia Magnética en Medicina y Biología (ESMRMB) en sus ediciones 2008 y 2009. En la edición 2008 se demostraron las capacidades de Curiam como CDSS multi-propósito a partir del diagnóstico de tumores de partes blandas mediante hallazgos de imagen de Resonancia Magnética (RM) utilizando Curiam STT (sección 4.7.2), y el

---

<sup>1</sup>Journal Citation Reports, 2008 JCR Science Edition, <http://www.accesowok.fecyt.es/jcr/>

<sup>2</sup>Computing Research and Education, <http://www.core.edu.au/>



diagnóstico de tumores cerebrales mediante MRS utilizando Curiam BT (sección 4.7.1) [38]. En la edición de 2009 se mostró la primera versión oficial de Curiam BT 1.0 la cual contaba con nuevas funcionalidades específicas del dominio de tumores cerebrales y MRS como el procesado y clasificación de datos crudos de 1.5 y 3.0 Tesla de los diferentes fabricantes y la disponibilidad de clasificadores para casos de adultos y niños [39].

El DSS genérico Curiam y su implementación Curiam BT han sido registrados en el registro de patentes, categoría software, de la Universidad Politécnica de Valencia CARTA<sup>34</sup>.

Finalmente, se participó como ponente en las XVI Jornadas de Informática Sanitaria en Andalucía en 2009<sup>5</sup> dentro del taller de Diagnóstico, donde se mostraron las posibilidades de los CDSS mediante el sistema Curiam.

### 1.3.3. Proyectos

El trabajo llevado a cabo en esta tesina ha sido parcialmente desarrollado en el marco del proyecto europeo HEALTHAGENTS, el proyecto europeo eTUMOUR y el proyecto nacional CURIAM SISTEMA DE AYUDA A LA DECISIÓN MÉDICA. PLAN DE EXPLOTACIÓN Y PRUEBA DE CONCEPTO EN EL HOSPITAL UNIVERSITARIO DOCTOR PESET Y LOS HOSPITALES DEL ÁREA METROPOLITANA DE BARCELONA.

HEALTHAGENTS, AGENT-BASED DISTRIBUTED DECISION SUPPORT SYSTEM FOR BRAIN TUMOUR DIAGNOSIS AND PROGNOSIS (2006-2008). HEALTHAGENTS es un Proyecto Europeo de Investigación Específica perteneciente al Sexto Programa Marco financiado por la Unión Europea dentro del área de Sociedad de Tecnologías de la Información. Su objetivo principal fue crear un sistema de ayuda a la decisión distribuido o dDSS basado en agentes para asistir en el diagnóstico precoz de tumores cerebrales y crear un almacén de datos distribuido con la mayor red mundial interconectada de bases de datos de información clínica, histológica y de fenotipo molecular de tumores cerebrales. Los objetivos del grupo IBIME en el proyecto fueron el crear un sistema de clasificación para el dDSS y desarrollar un conjunto de clasificadores para cuestiones específicas de diagnóstico y gradación basados en datos de espectroscopia de resonancia magnética. Concretamente, el trabajo desarrollado en el marco de esta tesina consistió, en primer lugar, en el diseño del sistema de clasificación distribuida basado en agentes que permita la ejecución remota de conjuntos de clasificadores y la recopilación de datos de distintos centros para crear nuevos clasificadores (capítulo 2). En segundo lugar se desarrolló un motor de clasificación genérico que pudiera permitir la ejecución de los diferentes clasificadores y facilitara la incorporación de nuevos modelos y técnicas de PR (capítulo 3).

eTUMOUR, WEB ACCESSIBLE MR DECISION-SUPPORT SYSTEM FOR BRAIN TUMOUR DIAGNOSIS AND PROGNOSIS, INCORPORATING IN VIVO AND EX VI-

---

<sup>3</sup>CURIAM: Sistema genérico de Ayuda a la Decisión para entornos clínicos y biomédicos, Código de registro CARTA: R13391-2009

<sup>4</sup>CURIAM BT: Sistema de Ayuda a la Decisión para el diagnóstico de Tumores Cerebrales, Código de registro CARTA: R13392-2009

<sup>5</sup>[http://www.seis.es/documentos/congresos/programaspdf/Programa\\_20Df\\_20Andalucia\\_1.pdf/](http://www.seis.es/documentos/congresos/programaspdf/Programa_20Df_20Andalucia_1.pdf/)

VO GENOMIC AND METABOLOMIC DATA (2004-2009). El proyecto eTumour tuvo dos objetivos principales. En primer lugar se trató de llevar a cabo una colaboración multi-centro para adquirir una base de datos significativa de tumores cerebrales. Se recopilaron datos de MRS, High Resolution Magic-Angle Spectra (HR-MAS) y microarrays de expresión genética obtenidos de biopsias. Esta base de datos permitió el estudio de nuevos avances científicos para el diagnóstico de tumores cerebrales. Por otra parte, el segundo objetivo del proyecto consistió en el desarrollo de un CDSS para el diagnóstico de tumores cerebrales. El principal objetivo de diseño del CDSS fue el de proporcionar una herramienta para la gestión de casos de MRS, que permitiese su clasificación. En el marco de esta tesina, parte del trabajo desarrollado en el framework de clasificación (capítulo 3) fue utilizado para proporcionar ayuda a la decisión al CDSS desarrollado en el proyecto.

CURIAM SISTEMA DE AYUDA A LA DECISIÓN MÉDICA. PLAN DE EXPLOTACIÓN Y PRUEBA DE CONCEPTO EN EL HOSPITAL UNIVERSITARIO DOCTOR PESET Y LOS HOSPITALES DEL ÁREA METROPOLITANA DE BARCELONA (2009-2010). Proyecto financiado por la Universidad Politécnica de Valencia bajo el programa INNOVA 2008. El objetivo de este proyecto consiste en realizar una prueba de concepto del CDSS Curiam, pretendiendo implantar y evaluar el sistema en un entorno clínico real, concretamente en el Hospital Universitario Doctor Peset y los hospitales del área metropolitana de Barcelona con el objetivo de (1) demostrar la robustez y utilidad del CDSS, permitiendo a los clínicos familiarizarse con la herramienta y comprobar su fiabilidad; (2) divulgar Curiam, generar nuevos contactos y ampliar así las funcionalidades del mismo mediante nuevas colaboraciones; y (3) emplear la información obtenida en la evaluación y los nuevos casos que introduzcan los especialistas médicos para mejorar el sistema. En el momento de escritura de esta tesina, este proyecto se encuentra aún vigente. El desarrollo hasta el momento actual de este proyecto dentro del trabajo de esta tesina, ha consistido en el propio desarrollo del CDSS Curiam así como la mejora del sistema en base al feedback obtenido por parte de los centros evaluadores. El trabajo futuro consistirá en la definición de formularios de evaluación y cuestionarios sobre uso del CDSS en entornos clínicos reales para posteriormente comprobar e interpretar los resultados obtenidos.

## Capítulo 2

# DSS distribuido basado en agentes en el proyecto HealthAgents

Los tumores cerebrales representan una importante causa de mortalidad en la población Europea. Aunque no sean el tipo de cáncer más común, los tumores cerebrales cuentan con una gran proporción entre los tumores en jóvenes y niños, siendo en estos últimos la enfermedad terminal más común.

El tratamiento de tumores cerebrales depende totalmente de su correcta clasificación, pero es difícil obtener un diagnóstico de forma no invasiva mediante técnicas clásicas de imagen médica en lugar de biopsias. Gracias al desarrollo de nuevas técnicas como MRS se puede incrementar la información obtenida de las imágenes médicas y mejorar la precisión en el diagnóstico, además estas técnicas pueden proveer nuevos biomarcadores para el pronóstico. Estos datos de MRS junto a imágenes de resonancia magnética, espectroscopia in-vitro, microarrays de expresión genética, datos histopatológicos e información clínica, permiten a los médicos obtener distintos puntos de vista y mejorar en la determinación del diagnóstico y pronóstico de tumores cerebrales.

La recopilación de las señales biomédicas puede permitir la investigación de soluciones basadas en aprendizaje automático y reconocimiento de patrones para obtener clasificadores automáticos de tumores cerebrales.

HEALTHAGENTS (2005-2008) [19]<sup>1</sup> es un proyecto de investigación financiado por la Unión Europea cuyo objetivo fue mejorar la clasificación de tumores cerebrales desarrollando un sistema distribuido de ayuda a la toma de decisiones o dDSS basado en agentes que implementase una serie de clasificadores automáticos basados en técnicas de reconocimiento de patrones y aprendizaje automático. Además el proyecto HealthAgents permitió la creación de la red HealthAgents Network, que proporciona un repositorio distribuido de información y conocimiento sobre tumores cerebrales y su diagnóstico y pronóstico.

Para el desarrollo del dDSS de HEALTHAGENTS se establecieron tres líneas principales: (1) el desarrollo de un framework de agentes para sustentar el sistema distribuido; (2) el desarrollo de un sistema de almacenamiento y compartición de datos clínicos; y (3) el desarrollo de un sistema de clasificación distribuida basado en PR.

El trabajo llevado a cabo en esta tesina dentro del proyecto HEALTHAGENTS se centra principalmente en el desarrollo del sistema de clasificación distribuida basado en técnicas de reconocimiento de formas, que es descrito en el capítulo 3. También se diseñó la

---

<sup>1</sup><http://www.healthagents.net/>

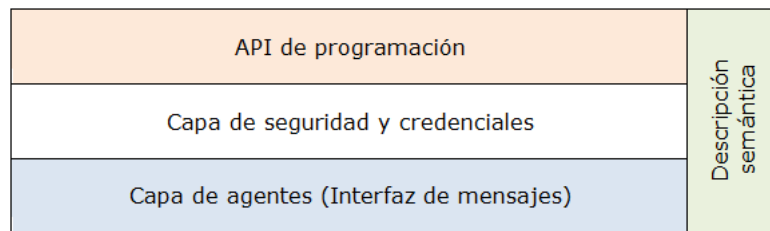


Figura 2.1: Estructura de capas del framework de agentes de HEALTHAGENTS.

Graphical User Interface (GUI) que proporcionase al facultativo los resultados de clasificación obtenidos tras la evaluación de un caso por el sistema (sección 2.4). Por otro lado se contribuyó en el diseño del sistema de recolección de datos distribuido que facilitara la búsqueda de datos con unas características determinadas para ser usados en el entrenamiento de nuevos clasificadores (sección 2.2). Todo el desarrollo en HEALTHAGENTS está basado en el framework de agentes diseñado para el proyecto, el cual se describe en la sección 2.1. Finalmente, en la sección 2.5 se describen los resultados del proyecto, indicando el despliegue del sistema y la evaluación del mismo.

## 2.1. El framework de agentes en HealthAgents

En un sistema software, el paradigma de desarrollo multi-agente es una buena elección cuando el sistema ha de ser distribuido en una red de trabajo, cuando entidades cuyo fin es resolver diferentes problemas trabajan juntas en partes de un problema común, y se comunican de forma asíncrona para compartir información. Este es el caso del dDSS de HEALTHAGENTS. Con el fin de obtener la plataforma tecnológica requerida para el desarrollo del sistema de HEALTHAGENTS se desarrolló un framework de agentes específico.

El framework de agentes de HEALTHAGENTS <sup>2</sup> esta desarrollado sobre el framework de desarrollo de agentes en Java JADE<sup>3</sup> — el cual sigue el estándar de comunicación entre agentes FIPA<sup>4</sup>. JADE está diseñado en una estructura por capas para facilitar el desarrollo de nuevos agentes y el tratamiento de los mensajes. Mediante una arquitectura vertical, el flujo del mensaje va desde la capa inferior hasta la superior, pudiéndose tomar decisiones en cualquier nivel para, por ejemplo, aceptarlo o rechazarlo. La capa inferior, o capa de agentes, es la que se encarga de la comunicación a bajo nivel entre los agentes. Esta capa gestiona el envío y recepción de mensajes mediante una encapsulación del sistema de agentes utilizado, en este caso JADE. La capa intermedia es la responsable de la seguridad en el tratamiento de los mensajes, gestionando los permisos conversacionales entre los agentes. Este sistema de seguridad está descrito en [54, 55]. Por último, la capa superior proporciona el API de programación para el desarrollo de funcionalidades específicas mediante nuevos agentes, abstrayendo el tratamiento de la seguridad y de la gestión de mensajería a los niveles inferiores.

<sup>2</sup>El autor agradece a David Dupplaw, de la Univeristy of Southampton, el desarrollo del framework de agentes de HEALTHAGENTS, así como la ayuda prestada en el desarrollo del sistema de clasificación basado en él.

<sup>3</sup>JADE, Telecom Italia, <http://jade.tilab.com/>

<sup>4</sup>Foundation of Intelligent Physical Agents, IEEE Computer Society, <http://www.fipa.org/>

Como se puede apreciar en la figura 2.1, se trabaja con una descripción semántica común a todos los niveles. Esta descripción semántica está formada por dos ontologías complementarias: la ontología de comunicación, empleada para definir la comunicación entre agentes; y la ontología del dominio, que define el contenido de los mensajes. La ontología de comunicación, llamada HEALTHAGENTS LANGUAGE o HAL, está definida en OWL<sup>5</sup>, y codificado mediante la representación Turtle<sup>6</sup>, para reducir el número de caracteres empleados en los mensajes. HAL se mapea a las performativas<sup>7</sup> FIPA para la comunicación a bajo nivel. Por otro lado, la ontología de dominio, llamada HA-Dom, establece el contexto de la aplicación, sus conceptos y relaciones entre ellos así como sus propiedades. Por ejemplo, HA-Dom consta de los diagnósticos asociados a los tumores cerebrales. HAL y HA-Dom facilitan la creación de los mensajes que los agentes del dDSS necesitan para su comunicación.

Por último, para facilitar que los agentes de la red de HEALTHAGENTS se encuentren los unos a los otros, se usa un servicio de directorio llamado *Yellow Pages*. El agente Yellow Pages permite a los agentes publicar los servicios que son capaces de ofrecer de forma que los otros agentes del sistema puedan decidir con que servicios conectarse. A lo largo de la red de HEALTHAGENTS, coexisten Agentes Yellow Pages locales — a los nodos — y globales para mejorar la eficiencia del sistema y hacerlo robusto frente a caídas en la red.

## 2.2. Diseño del sistema de clasificación distribuido

Dentro del estudio de PR en HEALTHAGENTS se establecieron dos líneas de investigación:

1. La investigación aplicada de técnicas de PR para desarrollar clasificadores automáticos para el diagnóstico de tumores cerebrales mediante información biomédica, principalmente MRS.
2. El diseño e implementación de un sistema de clasificación distribuida para que la clasificación automática de tumores cerebrales esté disponible en el sistema de HEALTHAGENTS.

El objetivo principal del diseño del dDSS fue crear un conjunto de agentes que fueran capaces de proporcionar ayuda en la toma de decisiones desde los centros clínicos utilizando clasificadores creados en centros de investigación especializados en PR. Esto implica por un lado el diseño de la arquitectura del sistema de agentes, y por otro lado, el diseño de un sistema que permita ejecutar los modelos obtenidos en el desarrollo de clasificadores. Como requisito, se tuvo que diseñar un sistema que permitiese la fácil integración de modelos en los diferentes nodos de PR. Para esto último, se desarrolló un framework de clasificación genérico, adaptable a diferentes problemas y dominios, capaz de ser extendido con nuevos métodos de PR. A este framework le llamaremos Classification Framework (CF), y dado su carácter genérico y entidad se trata exclusivamente en el capítulo 3 de esta tesina.

---

<sup>5</sup>OWL, Web Ontology Language. <http://www.w3.org/TR/owl-features/>

<sup>6</sup>Turtle, Terse RDF Triple Language. <http://www.dajobe.org/2004/01/turtle/>

<sup>7</sup>Instrucciones primitivas del framework

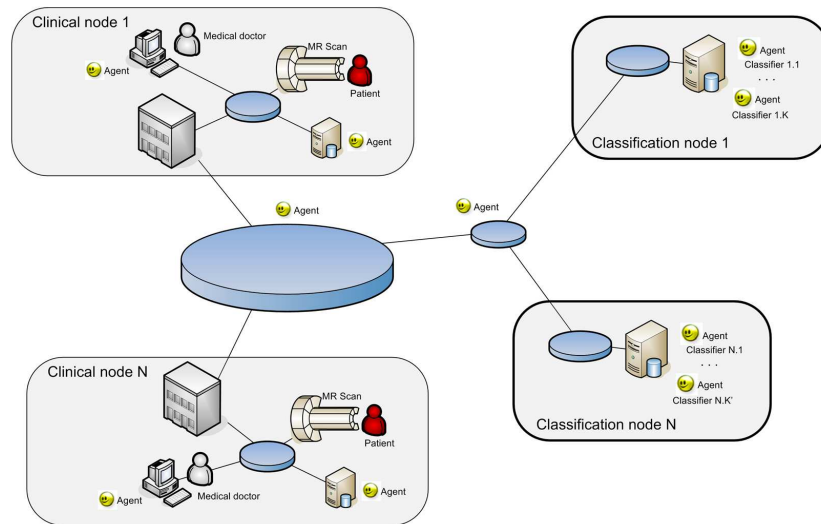


Figura 2.2: Red de HEALTHAGENTS mostrando nodos clínicos y nodos de clasificación

### 2.2.1. Arquitectura de HealthAgents

La arquitectura de agentes del dDSS en HEALTHAGENTS se distribuye en varios nodos a través de la red (ver Figura 2.2). La red consiste en un conjunto de nodos de diferente naturaleza: nodos clínicos, nodos directorio, y nodos de clasificación. Los nodos clínicos tienen un doble rol: proporcionar datos para la creación de clasificadores y ser los usuarios finales del dDSS. Cada nodo de clasificación proporciona un conjunto de clasificadores para diferentes tipos de datos y preguntas a resolver, a la vez que usando diferentes métodos de PR para la extracción de características y/o clasificación. Además, esta arquitectura permite a un centro clínico actuar simultáneamente como nodo clínico y de clasificación, permitiéndole crear sus propios clasificadores.

Los diferentes nodos del sistema pueden estar en diversas localizaciones geográficas, y pueden existir un número indeterminado de ellos. A pesar de estar físicamente separados, los agentes trabajarán de forma transparente dentro de la red de HEALTHAGENTS, gracias a los servicios de directorio distribuidos a lo largo de la red mediante instancias del agente Yellow Pages. Además, el sistema de seguridad permite que únicamente se puedan establecer conversaciones entre los agentes de la red de HEALTHAGENTS.

### 2.2.2. Classifier Petitioner Agent y Classifier Agent

De la definición del problema a desarrollar, tenemos que los nodos clínicos realizan peticiones de clasificación a un conjunto de clasificadores distribuidos en los nodos de clasificación. De esto podemos concluir que son necesarios dos tipos de agente: 1) el agente que represente a un clasificador, permitiendo recibir un caso y devolviendo un resultado de clasificación, localizado en un nodo clínico; y 2) el agente que se encargue de gestionar una petición de clasificación para un caso y realice las peticiones al conjunto de clasificadores seleccionados, localizado en un nodo de clasificación. A estos dos agentes les llamaremos Classifier Petitioner Agent y Classifier Agent respectivamente.

El Classifier Petitioner Agent es el responsable de gestionar las clasificaciones desde los nodos clínicos. Una vez el usuario de un nodo clínico haya determinado un conjunto

de clasificadores apropiado para el caso a diagnosticar, enviará tal lista junto al caso a clasificar al Classifier Petitioner Agent. El Classifier Petitioner Agent está diseñado para trabajar concurrentemente con diferentes peticiones de clasificación, dado que un nodo clínico puede estar ejecutando varios clientes del dDSS. Cuando haya recibido una petición de clasificación, el Classifier Petitioner Agent la remite a cada uno de los clasificadores especificados en la lista. Los resultados de clasificación llegan de forma asíncrona al Classifier Petitioner Agent, por lo tanto éstos son mapeados a su correspondiente petición tras la llegada. Una vez el último de los resultados de una petición haya sido recibido, el Classifier Petitioner Agent ordena los resultados según su relevancia y los envía de vuelta al agente que haya realizado la petición en el nodo clínico.

Un Classifier Agent representa a un clasificador en el sistema HEALTHAGENTS, por lo tanto existen tantas instancias del agente Classifier Agent como clasificadores hayan disponibles en el dDSS. Este agente utiliza la API proporcionada por el CF para realizar las clasificaciones y acceder a la información del clasificador. Cuando un Classifier Agent sea puesto en marcha, carga la descripción de su clasificador y, en el momento de registrarse en la red, publicita mediante el servicio de directorio a través del agente Yellow Pages las características necesarias para que otros agentes puedan identificarlo. La figura 2.3 muestra los agentes del sistema de clasificación junto al CF y los agentes Yellow Pages correspondientes a cada nodo. Al margen del tipo de agente del que se trata, el Classifier Agent publicita su pregunta a resolver y los datos necesarios para su clasificación, de forma que un cliente del dDSS sea capaz de determinar si un clasificador es apropiado o no para el caso a clasificar. Una vez el Classifier Agent reciba un mensaje conteniendo el dato a clasificar accede al CF para su clasificación y, una vez resulta la petición, crea un mensaje de respuesta que es enviado de vuelta al agente que solicitara la clasificación — en este caso el Classifier Petitioner Agent.

Además del sistema de clasificación, el dDSS de HEALTHAGENTS permite realizar otras acciones como consultas o recolecciones de datos, ofrecidas por sus agentes correspondientes. Por esta razón de diseño es conveniente que exista un agente que, desde un nodo clínico, represente al usuario en el sistema. Para esto se creó el GUI Agent. El GUI Agent recibe las órdenes desde la interfaz gráfica de usuario del dDSS y establece las conversaciones con los agentes necesarios en cada caso para resolver el problema solicitado. En el caso de la clasificación, en primer lugar el GUI Agent consulta al agente Yellow Pages por los agentes Classifier Agent apropiados para el caso a diagnosticar. Esto lo hace buscando en el servicio de directorio agentes Classifier Agent que registraran como característica la capacidad de clasificar utilizando los tipos de datos de los que el caso disponga o de clasificar un determinado tipo de tumor 2.4. El servicio de directorio devuelve las direcciones de los agentes seleccionados de forma que el agente solicitante sea capaz de localizarlos en la red de HEALTHAGENTS.

El protocolo o conversación entre agentes llevado a cabo en un proceso de clasificación es mostrado en la figura 2.5. El diagrama de secuencia UML de la figura muestra en una línea temporal vertical el intercambio de mensajes entre los cuatro agentes involucrados en el proceso, además de la entrada y salida del usuario con el GUI Agent.

Desde el punto de vista de los centros de desarrollo de clasificadores, es decir, donde residen los nodos de clasificación, se hace necesaria una herramienta que permita gestionar los clasificadores que están ejecutándose en el sistema. Con este fin se creó el gestor de agentes de clasificación tt Classifier Manager 2.6. Desde esta herramienta, los usuarios

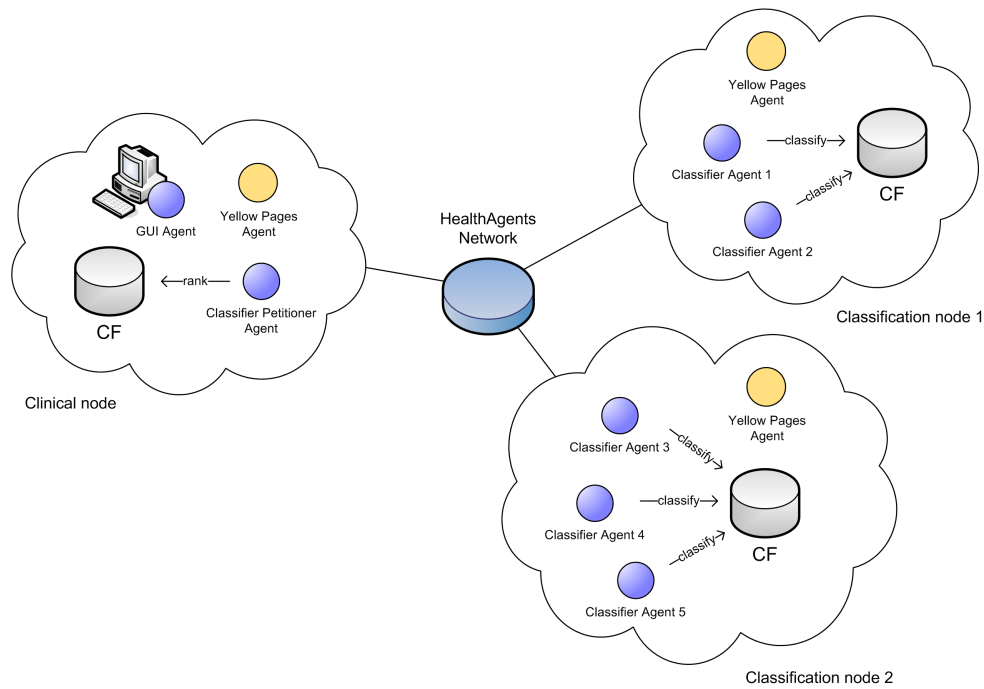


Figura 2.3: Red de HEALTHAGENTS integrando el CF. Los agentes de clasificación usan el CF para realizar las clasificaciones.

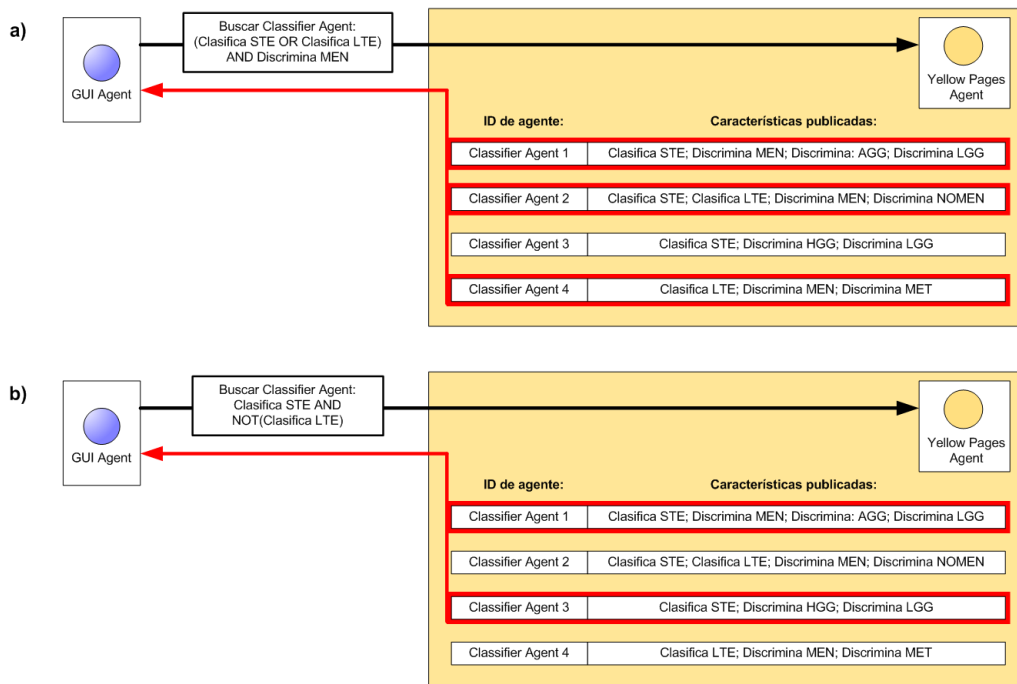


Figura 2.4: En a) el GUI Agent pregunta al agente Yellow Pages por los Classifier Agent para clasificar un caso con datos STE y LTE, indicando el tipo de tumor a discriminar 'MEN'. Se devuelven todos los Classifier Agent que son capaces de clasificar usando únicamente STE, únicamente LTE, ambos y capaces de discriminar el tipo de tumor 'MEN'. En b) se solicitan los Classifier Agent para un caso del que sólo se dispone del dato STE. En este caso hay que indicar explícitamente que no se quieren obtener aquellos Classifier Agent que necesiten el tipo de dato LTE para llevar a cabo la clasificación. Se devuelve únicamente los dos Classifier Agent capaces de clasificar tomando sólo STE.



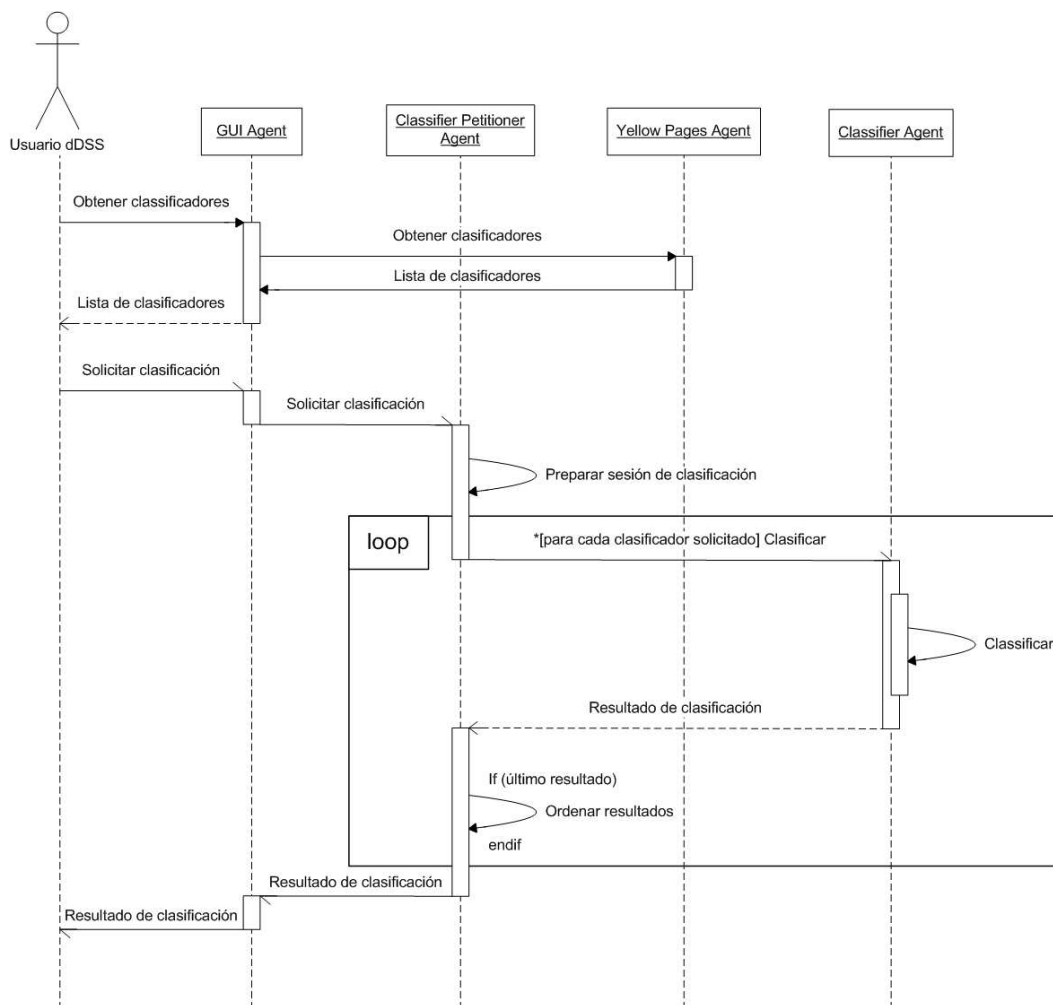


Figura 2.5: Diagrama de secuencia UML mostrando la secuencia de mensajes enviados entre los agentes involucrados en una clasificación lanzada desde un nodo clínico por un usuario del dDSS.



Figura 2.6: GUI del Classifier Manager. Desde esta herramienta los nodos de clasificación son capaces de gestionar los agentes Classifier Agent que están ejecutándose en el sistema para proceder a su mantenimiento o actualización.

de los nodos de clasificación pueden arrancar o detener los agentes de clasificación para su mantenimiento. Además es posible re-entrenar un determinado tipo de clasificadores a partir de un corpus de entrenamiento obtenido de una recolección de datos distribuida — esto es explicado en la sección siguiente —.

## 2.3. Recolección distribuida de datos

En la red de HEALTHAGENTS pueden existir múltiples fuentes de datos distribuidas en los nodos clínicos. El acceso a estos datos se realiza mediante el agente DataBase Agent. Este agente recibe peticiones para consultar datos desde otros agentes y, si los permisos son los adecuados, devuelve el resultado obtenido de la consulta. En el mensaje a enviar a un DataBase Agent, dentro del contenido en HAL se especificará una consulta en el lenguaje SPARQL<sup>8</sup> utilizando la ontología de domino HA-Dom que cada agente de base de datos traducirá al sistema gestor de bases de datos de su nodo. Una vez realizada la consulta, el resultado se devolverá mediante el mensaje correspondiente en forma de *resultset* XML.

En algunos casos, es conveniente realizar una búsqueda de un conjunto específico de datos a lo largo de todas las bases de datos de la red de HEALTHAGENTS. Este es el caso, por ejemplo, de la recopilación de un conjunto de datos para el entrenamiento o re-entrenamiento de un clasificador. Este proceso necesita de algún tipo de sincronización para difundir la petición, por esta razón se creó el agente Data Collector Agent.

Cuando un agente necesite recolectar un conjunto de datos de la red, éste debe crear la consulta SPARQL que recupere ese tipo de datos y enviarla mediante una petición al

<sup>8</sup>SPARQL Query Language for RDF, W3C. <http://www.w3.org/TR/rdf-sparql-query/>

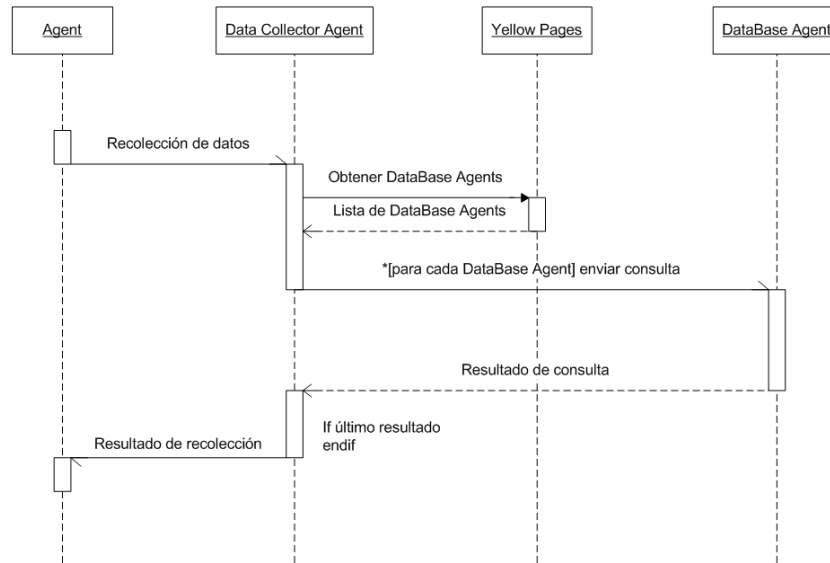


Figura 2.7: Diagrama de secuencia UML mostrando la secuencia de mensajes enviados entre los agentes involucrados en una recolección de datos distribuida.

Data Collector Agent. El Data Collector Agent busca en el servicio de directorio todos los DataBase Agents disponibles en la red sobre los que se tenga permisos. Una vez recuperada la lista, la consulta SPARQL se distribuye hacia estos agentes de base de datos. El Data Collector Agent recibe los *resultsets* de los DataBase Agents y los une en único resultado que devuelve al agente que inició la petición. Este proceso se refleja en la figura 2.7.

En HEALTHAGENTS se diseñó una prueba de concepto para el re-entrenamiento de clasificadores basados en KNN [49]. Para demostrar la funcionalidad de la recolección distribuida de datos, se permitió que un conjunto de datos obtenido a través del Data Collector Agent pueda usarse para re-entrenar un clasificador a través del Training Manager 2.8, el cual es accesible desde el Classifier Manager.

## 2.4. GUI de resultados de clasificación

Como se ha visto en el punto anterior, el GUI Agent hace de interfaz entre las acciones que un usuario del dDSS lleva a cabo y el sistema de HEALTHAGENTS. Para permitir la interacción del usuario en el sistema, HEALTHAGENTS proporciona una GUI basada en Web. La GUI del dDSS permite al usuario acceder a los repositorios de datos del sistema y clasificar los casos utilizando los clasificadores disponibles en la red. El trabajo desarrollado en esta tesina dentro de la GUI del sistema corresponde a la GUI de resultados de clasificación. El objetivo de esta GUI es proporcionar al usuario la información e interactividad necesaria para lograr una correcta ayuda en la toma de decisiones.

Tras completar una clasificación, los resultados son recibidos en el nodo clínico y mostrados en la GUI de resultados especializada (figura 2.9). A pesar de que los repositorios de datos de HEALTHAGENTS contienen información de múltiples niveles biológicos, el dDSS de HEALTHAGENTS trabaja exclusivamente con datos de MRS, por lo que la interfaz de resultados de clasificación se adaptó a este dominio para mostrar la información fácilmente interpretable por los especialistas. Los resultados obtenidos por cada uno de

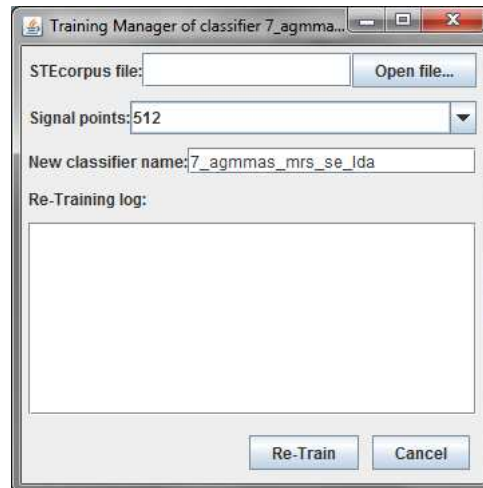


Figura 2.8: Cuadro de diálogo del Training Manager para el re-entrenamiento de clasificadores basados en acKNN

los clasificadores ejecutados se muestran de forma tabular ordenados por el criterio del Classifier Petitioner Agent. El usuario puede acceder a los detalles de un clasificador seleccionando su resultado en la tabla. Estos resultados se muestran en una GUI interactiva dividida en tres componentes:

- (1) El primer componente muestra probabilidades a posteriori resultantes en un gráfico de tarta junto a la información relativa a la descripción del clasificador: pregunta a resolver, datos utilizados y técnica de PR utilizada.
- (2) El segundo componente, dependiendo de la técnica de PR empleada por el clasificador, permite una visualización comparativa del caso con la base de datos de referencia. Por ejemplo, en clasificadores basados en Fisher LDA, visualizando una proyección del espacio latente de las muestras usadas en el entrenamiento del clasificador junto a la muestra clasificada, donde la dispersión inter-clase es maximizada a la vez que la intra-clase es minimizada.
- (3) El tercer componente visualiza las medias y desviaciones típicas por clase de los datos de MRS usados en el entrenamiento del clasificador junto al espectro del caso clasificado permitiendo una comparación visual. Esta representación frente al espectro medio refuerza los resultados obtenidos por el clasificador. La visualización además muestra los puntos de interés del espectro que están siendo usados para la predicción por el clasificador.

Además, los componentes (2) y (3) se encuentran conectados: en el caso de clasificadores Fisher LDA, desde el espacio latente es posible seleccionar un punto en particular para mostrar su correspondiente espectro en la visualización de espectros.

## 2.5. Resultados del proyecto

En el momento de la finalización del proyecto en Diciembre de 2008, la red de HEALTHAGENTS estaba compuesta por cinco nodos distribuidos a lo largo de la Unión Europea: tres

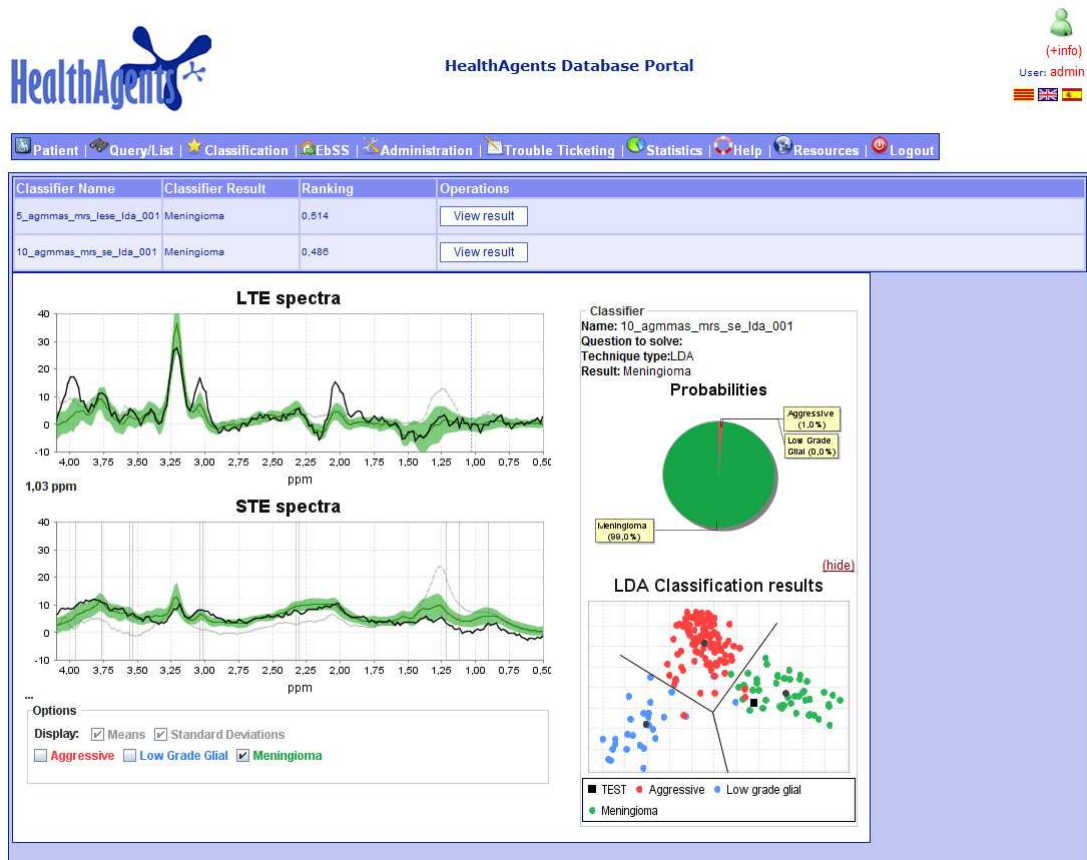


Figura 2.9: GUI de visualización de resultados de clasificación del dDSS de HEALTHAGENTS.

nodos clínicos y dos nodos de clasificación. Los nodos clínicos estuvieron localizados en el Birmingham Children's Hospital, en el Reino Unido; la Universidad de Valencia, España; y la Universitat Autònoma de Barcelona, España. Uno de los nodos de clasificación estuvo localizado en la Katholieke Universiteit Leuven, Bélgica; y el otro en el Instituto ITACA de la Universidad Politécnica de Valencia, España. Por otro lado, se dispusieron de servicios de directorio globales mediante agentes Yellow Pages en la Southampton University, Reino Unido; y en Microart S.L., España. Los nodos de clasificación ofrecieron un total de 25 clasificadores para el diagnóstico de tumores cerebrales mediante MRS respondiendo múltiples preguntas binarias y multi-clase. Los clasificadores se integraron en el sistema mediante el CF y utilizaron las siguientes técnicas de PR: Fisher LDA [11]; KNN [6]; LS-SVM [45]; principal component analysis (PCA) [20]; y independent component analysis (ICA) [4, 22, 23]. Estos clasificadores fueron entrenados con datos adquiridos durante el proyecto INTERPRET y HEALTHAGENTS. La tabla 2.1 muestra los 25 clasificadores integrados en el dDSS de HEALTHAGENTS<sup>9</sup>.

Para ilustrar el uso distribuido del sistema de clasificación mediante el HEALTHAGENTS Agent Framework, la figura 2.10 muestra capturas de las herramientas utilizadas para monitorizar los agentes involucrados en una clasificación distribuida. Los mensajes entre agentes pasan desde el GUI Agent en un nodo clínico a través del Classifier Petitioner Agent hasta cada Classifier Agent en el nodo de clasificación. El resultado final es recibido por el GUI Agent.

Para estudiar el valor añadido del dDSS de HEALTHAGENTS, se llevó a cabo una evaluación del sistema siguiendo la metodología TAM [8]. Esta evaluación fue liderada por la University of Birmingham, partner del proyecto HEALTHAGENTS. Durante este estudio fueron entrevistados 26 médicos expertos. Como respuesta general, consideraron que el uso del dDSS es beneficioso para mejorar la calidad de los diagnósticos de tumores cerebrales. Además, estuvieron de acuerdo en que el dDSS ofrecía una buena usabilidad respecto al usuario, punto importante en cualquier DSS, especialmente en un entorno clínico.

Uno de los resultados de mayor impacto del proyecto ha sido el motor de clasificación genérico para la integración de clasificadores: el CF. El trabajo desarrollado en el CF ha sido aceptado para publicación en [40], donde se ilustra su integración en HEALTHAGENTS. El CF es descrito en el capítulo siguiente y sirve de sistema de clasificación para el DSS Curiam, detallado en el capítulo 4.

---

<sup>9</sup>El autor agradece a Jan Luts de KUL por colaborar en el diseño de los clasificadores

Cuestión	Caracts.	Casos	FE	Clasif.	ACC	BAR	Desarrollador
AGG vs MEN vs LGG	LTE+STE	125 (71,13,41)	PPM	KNN	0,80 ( $\pm 0,07$ )	0,78	UPV-ITACA
AGG vs MEN vs LGG	LTE+STE	125 (71,13,41)	PPM	LDA	0,75 ( $\pm 0,07$ )	0,65	UPV-ITACA
AGG vs MEN vs LGG	STE	217 (124,58,35)	PPM	LDA	0,94 ( $\pm 0,03$ )	0,93	UPV-ITACA
AGG vs MEN vs LGG	STE	217 (124,58,35)	PI	KNN	0,86 ( $\pm 0,04$ )	0,84	UPV-ITACA
AGG vs MEN vs LGG	STE	217 (124,58,35)	PCA	LDA	0,91 ( $\pm 0,04$ )	0,90	UPV-ITACA
AGG vs MEN vs LGG (Ad.)	STE	209 (121,55,33)	PI	LDA	0,90 ( $\pm 0,04$ )	0,88	UPV-ITACA
MEN vs MET	STE	96 (58,38)	PCA	KNN	0,91 ( $\pm 0,05$ )	0,91	UPV-ITACA
MEN vs MET	STE	94 (57,37)	ICA	LS-SVM	0,88 ( $\pm 0,07$ )	0,89	KUL
MEN vs MET	STE	94 (57,37)	ICA	LS-SVM	0,90 ( $\pm 0,06$ )	0,91	KUL
HG vs LG (Ad.)	STE	237 (134,103)	PI	LDA	0,87 ( $\pm 0,05$ )	0,87	UPV-ITACA
HG vs LG	STE	262 (147,115)	PI	LDA	0,85 ( $\pm 0,04$ )	0,84	UPV-ITACA
HG vs LG (Ch.)	STE	70 (28,42)	PCA	LDA	0,69 ( $\pm 0,10$ )	0,68	UPV-ITACA
AFF vs NON-AFF	STE	304 (22,282)	PI	KNN	0,99 ( $\pm 0,01$ )	0,94	UPV-ITACA
TUM vs NONTUM	STE	282 (274,8)	PI	KNN	0,97 ( $\pm 0,02$ )	0,64	UPV-ITACA
AGG vs NON AGG	STE	244 (135,109)	PI	KNN	0,81 ( $\pm 0,05$ )	0,81	UPV-ITACA
AGG vs NON AGG (Ch.)	STE	79 (30,49)	PCA	KNN	0,72 ( $\pm 0,10$ )	0,71	UPV-ITACA
GLM vs EMB (Ch.)	STE	58 (43,15)	PCA	KNN	0,72 ( $\pm 0,11$ )	0,62	UPV-ITACA
GLM vs EMB (Ch.)	STE	48 (33,15)	PCA	KNN	0,71 ( $\pm 0,13$ )	0,66	UPV-ITACA
GBM vs LGG	STE	117 (84,33)	ICA	LS-SVM	0,83 ( $\pm 0,07$ )	0,83	KUL
GBM vs LGG	STE	117 (84,33)	ICA	LS-SVM	0,84 ( $\pm 0,07$ )	0,87	KUL
GBM vs MEN	STE	141 (84,57)	ICA	LS-SVM	0,84 ( $\pm 0,06$ )	0,79	KUL
GBM vs MEN	STE	141 (84,57)	ICA	LS-SVM	0,91 ( $\pm 0,07$ )	0,89	KUL
MEN vs LGG	STE	90 (57,33)	ICA	LS-SVM	0,84 ( $\pm 0,08$ )	0,83	KUL
MEN vs LGG	STE	90 (57,33)	ICA	LS-SVM	0,92 ( $\pm 0,06$ )	0,92	KUL
MET vs LGG	STE	70 (37,33)	ICA	LS-SVM	0,85 ( $\pm 0,08$ )	0,87	KUL

Cuadro 2.1: La tabla muestra los 25 clasificadores de tumores cerebrales para diferentes cuestiones clínicas integrados en HEALTHAGENTS. La primera columna indica los tipos/agrupaciones de tumor que el clasificador discrimina (tumor agresivo (AGG), meningioma (MEN), glial de grado bajo (LGG), metástasis (MET), tumor de grado alto (HG), tumor de grado bajo (LG), tejido afectado (AFF), tejido no afectado (NON-AFF), tumor (TUM), no tumor (NON-TUM), tumor no agresivo (NON-AGG), glioma (GLM), tumor embrional (EMB), glioblastoma (GBM)). Se indica si el clasificador es específico para adultos (Ad.) o niños (Ch.). La segunda columna indica qué señales han sido usadas por el clasificador. La tercera columna indica el tamaño del corpus de entrenamiento y la distribución de muestras por clase. La cuarta columna indica el método de extracción de características (FE) empleado por el clasificador (intensidades de pico (PPM), integración de picos (PI)). La quinta columna indica el acierto del clasificador (ACC). La sexta columna indica el ratio de acierto balanceado (BAR). La última columna indica el centro de desarrollo del clasificador (Katholieke Universiteit Leuven, Bélgica (KUL); Instituto ITACA en la Universidad Politécnica de Valencia, España (UPV-ITACA)).

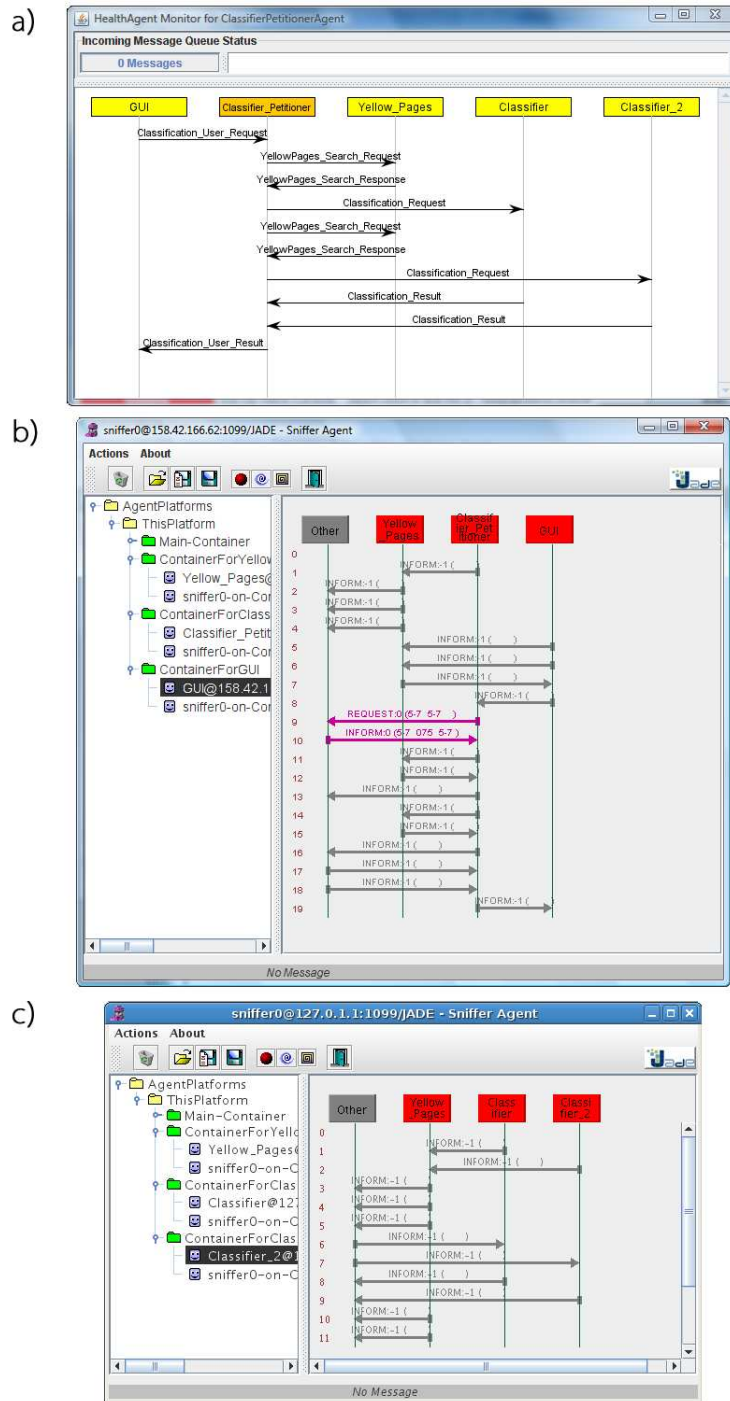


Figura 2.10: Estas tres capturas de pantalla muestran la monitorización de una clasificación llevada a cabo desde un nodo clínico usando dos clasificadores localizados en un nodo de clasificación. a) muestra el monitor de HEALTHAGENTS del Classifier Petitioner Agent. Este monitor muestra los mensajes enviados hacia y desde el Classifier Petitioner Agent. El GUI Agent pide una clasificación, el Classifier Petitioner Agent pregunta al Yellow Pages sobre Classifier Agent remotos. La petición se distribuye a los Classifier Agents y tras ejecutar las clasificaciones los resultados son devueltos al Classifier Petitioner Agent el cual los empaqueta y devuelve al GUI Agent. b) y c) muestran respectivamente el JADE Sniffer en el nodo clínico y el nodo de clasificación. El JADE Sniffer muestra los mensajes FIPA de bajo nivel enviados entre los agentes de las plataformas de cada nodo. Los mensajes enviados entre el Classifier Petitioner Agent en el nodo clínico y los Classifier Agents en el nodo de clasificación son mostrados en el sniffer como enviado o recibido a/desde otro (OTHER) agente.



# Capítulo 3

## Sistema de clasificación automática genérico: Classification Framework

### 3.1. Motivación

Todo DSS depende de modelos del conocimiento para llegar a ofrecer una ayuda al usuario para tomar decisiones. En el caso de DSS para diagnóstico médico, o generalizando, para clasificación, suelen ser técnicas de aprendizaje automático las empleadas para encontrar regularidades en los datos que permitan clasificarlos entre varias clases. El objetivo del dDSS de HEALTHAGENTS es proporcionar a los médicos una herramienta para la clasificación de casos de pacientes con tumores cerebrales mediante datos de MRS. Estos modelos, son obtenidos de la aplicación de técnicas de PR en los centros de investigación en aprendizaje automático. Desde los mismos, se ofrecerían los modelos mejor evaluados de los desarrollados mediante las metodologías particulares de cada centro. Esto podría dar lugar a modelos definidos mediante diversas técnicas de PR, y que requirieran determinados tipos de datos de los disponibles en el dDSS. Por otra parte, pese a la norma de clasificar mediante datos de MRS, cabía la posibilidad de que, si se recopilaban conjuntos de datos suficientes, pudieran emplearse datos de otros niveles biológicos, como HR-MAS o microarrays de expresión genética, para obtener nuevos modelos de clasificación. Además, nuevos avances en el diagnóstico de tumores cerebrales podrían ser contemplados a lo largo del proyecto permitiendo emplear nuevas técnicas.

Todas estos requisitos, llevaron al diseño de una herramienta de clasificación automática que ofreciera las siguientes características:

1. que permitiese integrar múltiples clasificadores desarrollados por diferentes grupos;
2. que sea capaz de incluir diferentes técnicas de PR, ya sea para preproceso de datos, extracción de características y clasificación, y ser extendido con nuevos métodos; y
3. que pudiese trabajar con diferentes tipos de datos.

Con el objetivo de proporcionar a HEALTHAGENTS con el sistema de clasificación automática requerido y pensando en su utilización por parte de futuros desarrollos en DSS, se desarrolló un sistema de clasificación automática genérico y extensible, capaz de ser adaptado a diferentes problemas y campos, y que facilitase la integración de nuevos modelos y técnicas: el Classification Framework.

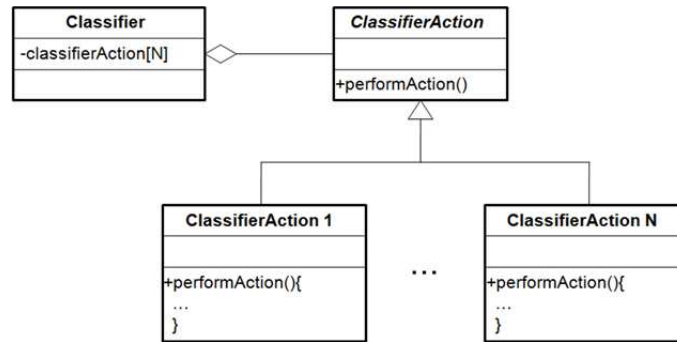


Figura 3.1: Diagrama de clases UML del diseño de las acciones de clasificación en el CF. Mediante el patrón de diseño de software orientado a objetos 'estrategia', los algoritmos de las acciones de clasificación son encapsulados por el método `performAction()`.

## 3.2. Diseño del framework

El CF se diseñó en forma de módulo software, o librería, para proporcionar clasificación automática a terceros. El CF no se puede considerar una aplicación por sí mismo, sino que provee una application program interface (API) que permite su integración en otros sistemas. En paralelo, esta API es usada para extender el framework con nuevos métodos.

El principal detalle en el diseño del CF es que separa la especificación de los clasificadores de la implementación de las técnicas de PR usadas por éstos. Como se describe mas adelante, un clasificador añadido en el sistema puede definir en sus ficheros de especificación eXtensible Markup Language (XML) cuales de las técnicas incluidas en el CF a usar. Esta aproximación permite la reutilización de las implementaciones de las técnicas que ya forman parte del sistema, evitando tener que programarlas de nuevo para nuevos clasificadores. El CF se centra principalmente en el proceso de clasificación, dejando el proceso de entrenamiento de los modelos a las herramientas de laboratorio habituales de los grupos de investigación en PR. Por lo tanto, la implementación de una nueva técnica o método a incluir en el sistema consiste en los algoritmos relativos al proceso de clasificación. Una vez incluido en el sistema, un método puede ser utilizado por nuevos modelos simplemente indicándolo en sus especificaciones XML.

Los clasificadores basados en PR, tal y como están definidos en el presente framework, pueden estar formados por diferentes acciones para llevar a cabo las predicciones, incluyendo varios métodos para selección y extracción de características, transformaciones de datos, y clasificación. Conceptualmente, estos métodos pueden considerarse 'acciones de clasificación', donde cada método toma una entrada y genera una salida. Un clasificador puede ejecutar una serie de estas acciones en forma de *pipeline* antes de obtener su resultado final. La incorporación de nuevas acciones de clasificación en el framework pretende ser una tarea fácil para los desarrolladores. Siguiendo el patrón de diseño de software orientado a objetos denominado 'estrategia', la implementación de una acción de clasificación se encapsula como un algoritmo intercambiable siguiendo una API específica (figura 3.1). Existe por tanto una clase abstracta que provee de la definición de un método abstracto, el cual debe ser escrito en la correspondiente subclase. De esta forma, al implementar una nueva acción de clasificación el desarrollador únicamente debe preocuparse de la implementación de los algoritmos específicos del método.

El punto de entrada para clientes del CF es la clase `Classifier`, clase principal del

framework que representa un clasificador. En el constructor del clasificador se le especifica el fichero de especificación del clasificador. Este fichero es leído a través de la clase `ClassifierDataFramework` e indica al clasificador qué acciones de clasificación ejecutar. Según la representación de los ficheros de especificación de clasificadores, se pueden crear subclases de `ClassifierDataFramework` para leer formatos concretos, aunque hasta el momento se usa el formato XML, utilizando las correspondientes clases de lectura. La clase abstracta que representa las acciones de clasificación se denomina `ClassifierAction`. Por cuestiones de organización de posibles funcionalidades conceptualmente relacionadas, se han incluido tres agrupaciones de acciones de clasificación mediante tres clases abstractas: `Technique`, `Transformation` y `Normalization`, para que cuelguen de ellas las técnicas de clasificación, transformaciones de datos y normalizaciones, respectivamente. De estas clases — aunque también directamente de la clase `ClassifierAction` — heredan las acciones implementadas. Normalmente, un desarrollador puede adjuntar a la clase representante de una acción, tantas clases como sean necesarias para un correcto diseño de la misma. Cuando una acción de clasificación es creada, recibe los parámetros necesarios, especificados en el fichero de especificación del clasificador. La lectura de los parámetros específicos de una acción se efectúa mediante la clase `ClassifierActionDataFramework`. Para permitir futuros cambios en el formato de especificación de los clasificadores, el desarrollador de una acción debe incluir una interfaz que defina los métodos a llamar para leer las acciones. La figura 3.2 un diagrama de clases UML con las principales clases del CF y varios ejemplos de acciones de clasificación.

En el PR estadístico, una muestra, o individuo, consiste en un vector de características. La mayoría de tipos de datos pueden representarse de esta forma de cara a la minería de datos, por ejemplo datos médicos y biológicos. Incluso si un clasificador usa combinaciones de estos datos, el vector de características puede estar formado por concatenaciones de los vectores originales. El CF puede recibir estos vectores como entrada de forma estructurada, para ser incluidos en un repositorio o *pool* de objetos de donde las acciones de clasificación pueden seleccionar el conjunto de características necesarias para llevar a cabo la clasificación. Del mismo modo, cada una de las acciones de clasificación genera una salida en la que se estructuran sus resultados — ya sean valores numéricos como transformaciones de datos o probabilidades, o meta-datos informativos como por ejemplo gráficas —, y se depositan en el *pool* de forma que puedan ser utilizados por otras acciones. Junto a las probabilidades a posteriori obtenidas de la clasificación, la especificación del clasificador indica que elementos de las salidas de las acciones se devuelven como resultado final del clasificador. Al iniciarse una clasificación, todas las acciones son ejecutadas en paralelo en forma de hilos de ejecución, activándose cuando los datos que sean necesarios para su proceso estén disponibles 3.3. Por ejemplo, acciones que únicamente necesiten de los datos de entrada se ejecutarán instantáneamente, pero otras acciones como una técnica de clasificación que requiera una transformación previa de datos, se mantendrá a la espera de que la transformación haya finalizado y depositado su salida en el *pool* de objetos. Un ejemplo de clasificación a través de varias acciones de clasificación puede verse en la figura 3.4. La entrada de datos vectorial/estructurada permite que el CF tenga una entrada genérica. Los clasificadores del CF son por tanto capaces de trabajar con la mayoría de tipos de datos y resolver las cuestiones que puedan ser expresadas mediante esos datos.

El diseño del CF proporciona un método flexible para la incorporación de nuevos clasificadores en el sistema. En el CF se han definido plantillas para la especificación

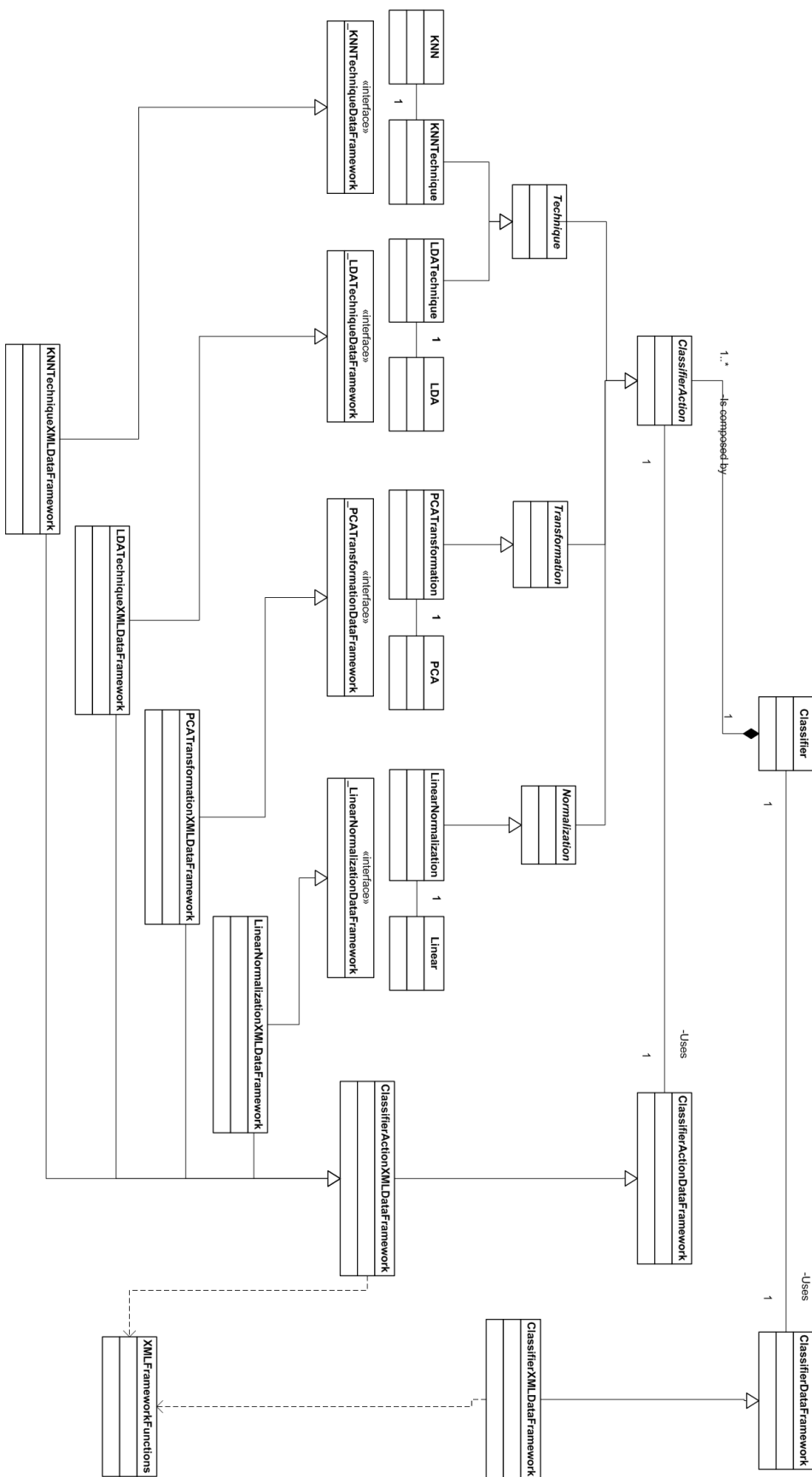


Figura 3.2: Diagrama de clases UML del CF. Se ilustran las clases principales y varios ejemplos de acciones de clasificación.

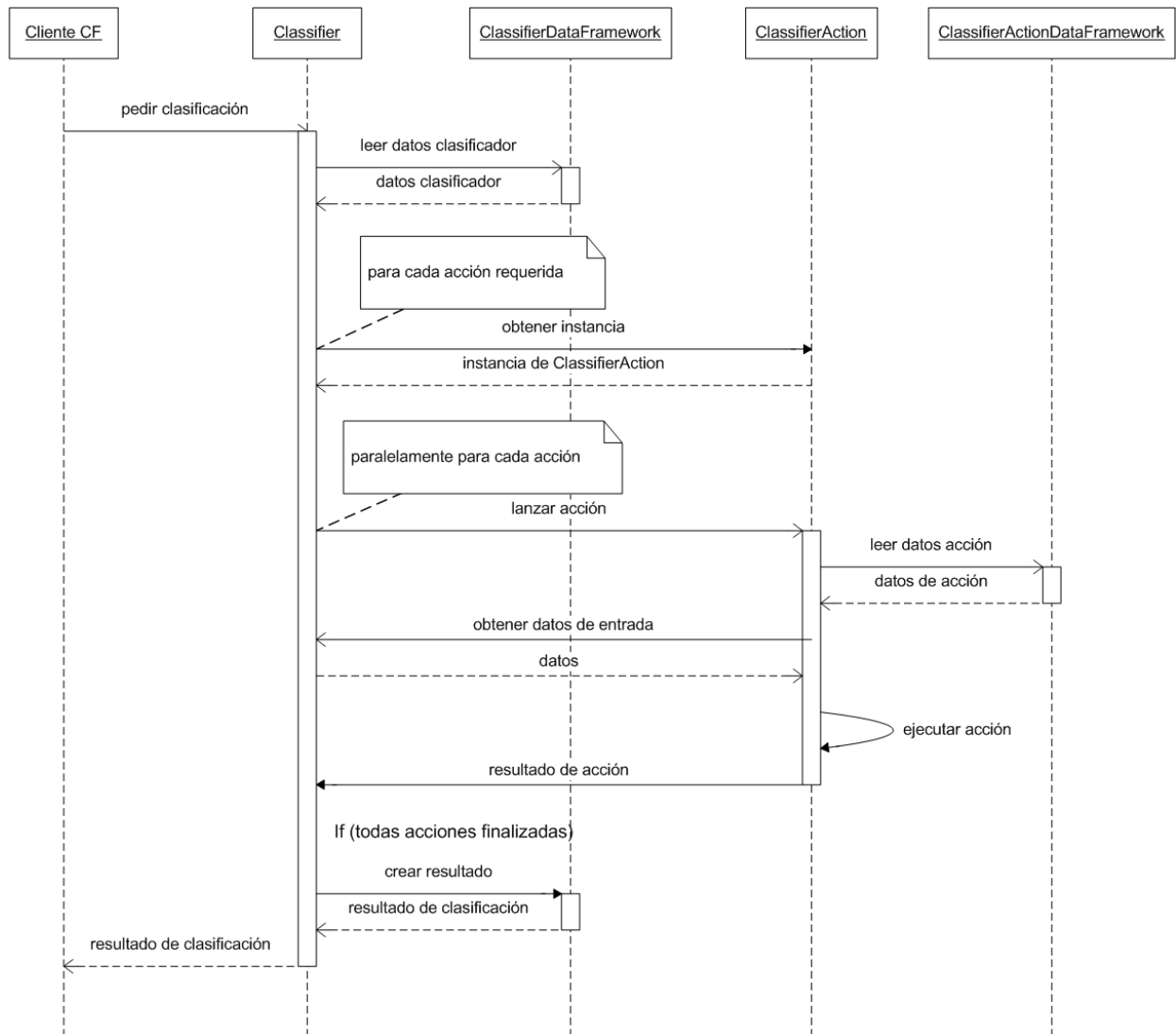


Figura 3.3: Diagrama de secuencia UML de una clasificación en el CF. Un software cliente clasifica mediante la API proporcionada por el CF a través de la clase Classifier. Una vez leída la especificación del clasificador se lanzan las acciones de clasificación de forma paralela, las cuales buscan sus datos de entrada y van depositando sus resultados en el *pool* de resultados en la clase Classifier. Cuando todas las acciones han finalizado se crea y devuelve al cliente el resultado de clasificación. .

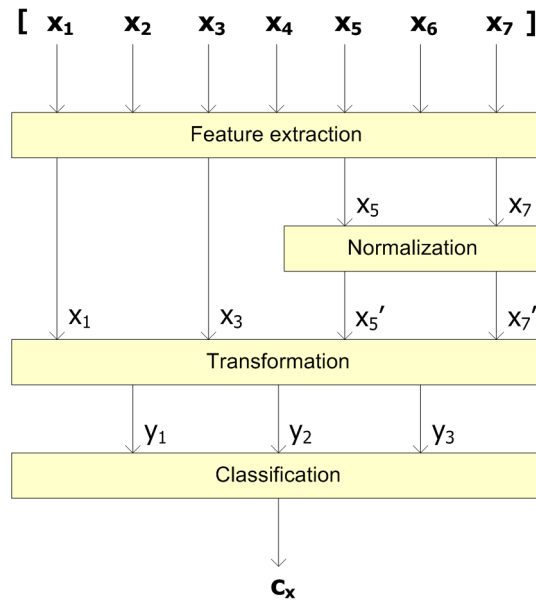


Figura 3.4: Ejemplo de un conjunto de acciones de clasificación secuenciadas para obtener una clasificación de un vector de características. De todo el conjunto de características (desde  $x_1$  a  $x_7$ ) son seleccionadas cuatro por el filtro de extracción de características ( $x_1$ ,  $x_3$ ,  $x_5$  y  $x_7$ ). Se aplica una normalización a  $x_5$  y  $x_7$ . Después se aplica una transformación a  $x_1$ ,  $x_3$  y las características normalizadas  $x_5'$  y  $x_7'$ . Finalmente, la salida de la transformación es enviada a la acción de clasificación para obtener las probabilidades a posteriori para cada clase.

de clasificadores basadas en XML. Para incorporar un clasificador al sistema, solo es necesario escribir su correspondiente especificación XML e incluirla al framework. En la sección siguiente se describen los detalles de diseño de los ficheros de especificación XML de clasificadores en el CF.

Para finalizar la sección se describe un escenario de clasificación simple usando el CF: Un módulo o software cliente solicita al API del CF las descripciones de los clasificadores disponibles para comprobar cuál se ajustan a sus necesidades. Una vez un clasificador es seleccionado, se instancia mediante la clase `Classifier`. El clasificador parsea el contenido de su especificación `method.xml` y crea un conjunto de acciones de clasificación. Dichas acciones cargan el conjunto de parámetros especificados en la especificación. Ahora que el clasificador está listo para ejecutar la clasificación, la muestra de datos puede ser recibida desde el cliente. Los datos pasan a través de la secuencia de acciones de clasificación, mientras se van generando las correspondientes salidas en cada paso. Una vez la acción final se ha completado, el clasificador genera la salida correspondiente, la cual es devuelta al cliente. Este proceso se ilustra en la 3.5.

### 3.3. Lenguaje de especificación de clasificadores basado en XML

XML es usado en la especificación personalizada de datos o lenguajes. Con XML, los desarrolladores pueden describir estándares de información específicos para aplicaciones. En el CF, los clasificadores son representados mediante documentos XML que especifican

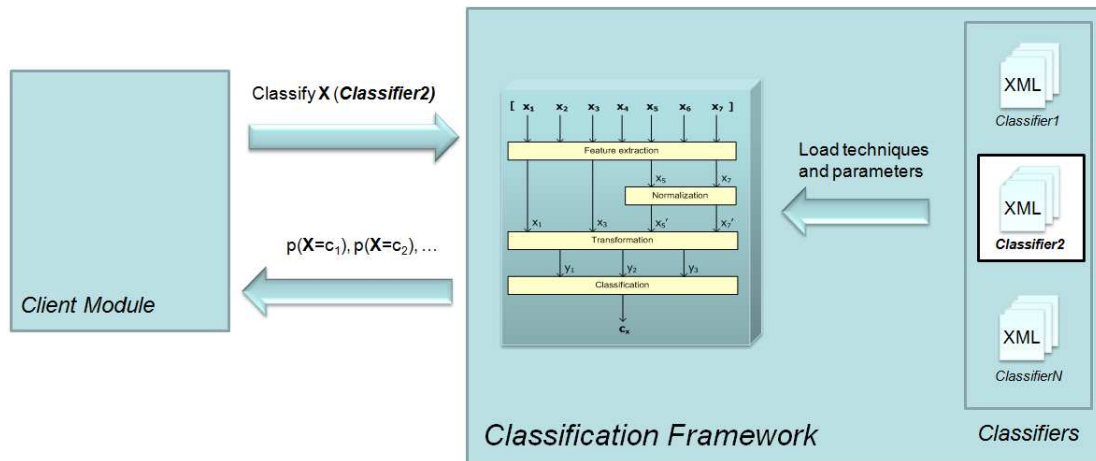


Figura 3.5: Un módulo cliente solicita la clasificación de  $X$  usando el clasificador **Classifier2**. Las acciones de clasificación especificadas en el fichero `method.xml` son cargadas con sus correspondientes parámetros para crear la secuencia de acciones. Las probabilidades son enviadas de vuelta al cliente una vez la clasificación ha finalizado.

sus descripciones y los pasos necesarios para obtener una clasificación. El uso de XML permite que el lenguaje de especificación de clasificadores puede ser extendido cuando el sistema incluya nuevas funcionalidades. Además, permite añadir información adicional para mejorar la calidad de los resultados, como por ejemplo, proyecciones de los datos de entrenamiento, o información específica del dominio biomédico del clasificador. Por otra parte, los documentos basados en XML pueden ser interpretados por la mayoría de tecnologías software, lo que facilita su tratamiento desde herramientas o aplicaciones externas.

La especificación XML de clasificadores se divide en tres ficheros XML (figura 3.6):

**description.xml**: contiene información relativa a la descripción del clasificador, como que tipo de datos de entrada usa, la pregunta a resolver, por que centro fue desarrollado, y que casos han sido usados para su entrenamiento. Por medio de este fichero, el clasificador proporciona la información **contextual** necesaria a los usuarios para saber si es adecuado para su problema o no.

**profile.xml**: contiene información relativa al perfil del clasificador como medidas de acierto e información estadística de los datos de entrenamiento. Está preparado para almacenar valores dinámicos acerca del uso del clasificador como, por ejemplo, una estimación del error mediante una estrategia de evaluación concreta.

**method.xml**: contiene la especificación del proceso de clasificación, definiendo las acciones de clasificación a ejecutar y, para cada una de ellas, su conjunto de parámetros y datos de entrada requeridos. El fichero `method.xml` describe también que salida devuelve el clasificador, la cual puede estar formada por cualquier combinación de salidas de las acciones. Además, para proporcionar una salida más completa, se pueden adjuntar elementos adicionales como gráficas o visualizaciones de datos.

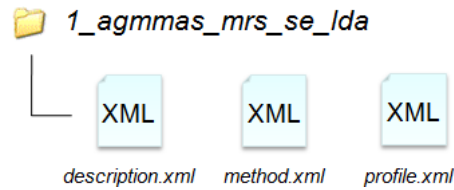


Figura 3.6: Un clasificador está representado por tres ficheros de especificación XML.

El lenguaje XML definido ha sido desarrollado en XML puro. La estructura de los documentos ha sido definida de forma sólida, lo que permite cumplir con el estándar del CF, el cual soporta tolerancia a fallos. Sin embargo, una especificación más correcta debería contar con la definición de esquemas XML (XML-Schema), a modo de guía para la extensión del lenguaje y de método de validación. La definición de los esquemas XML para el lenguaje de especificación de clasificadores del CF está incluido en el plan de trabajo futuro de desarrollo del CF. Así mismo, se pretende proporcionar al CF compatibilidad con el estándar de representación de modelos de minería de datos PMML <sup>1</sup>.

En la sección 1.2 se describieron las técnicas de PR empleadas por el CF. Como ya se ha comentado, el CF se centra en el proceso de clasificación, por lo tanto, se han desarrollado los algoritmos específicos de clasificación para cada una de estas técnicas. La mayoría de los desarrollos de estas técnicas han sido llevados a cabo en el marco de esta tesina. La implementación de la técnica de clasificación LS-SVM y la transformación ICA fue llevado a cabo por la Katholieke Universiteit Leuven <sup>2</sup> dentro del proyecto HEALTHAGENTS. La implementación de la técnica de redes neuronales fue llevada a cabo mediante unas prácticas en empresa en la Asociación ITACA <sup>3</sup>, a la cual pertenece el grupo IBIME. Estos desarrollos llevados a cabo por terceros, demuestran la escalabilidad del CF a la hora de incorporar nuevos métodos de PR. En el fichero `description.xml` se especifica el proceso de clasificación definiendo los diferentes métodos. Cada método, especifica los parámetros requeridos para su ejecución, según la estructura definida en su desarrollo. Las estructuras de algunos de los métodos de PR incluidos en el CF serán descritas en esta sección.

A continuación se describe la estructura de los tres ficheros XML que representan un clasificador en el CF:

### description.xml

```
<DESCRIPTION>
  <Classifier_Name>
    identificador del clasificador
  </Classifier_Name>
  <Accessibility>
    public/private - en HealthAgents, especifica si el clasificador puede
    ser accedido desde fuera del nodo en que se encuentra
  </Accessibility>
  <Version>
    xxx - número de versión, en caso de actualización del modelo
  </Version>
```

<sup>1</sup><http://www.dmg.org/pmml/>

<sup>2</sup>El autor agradece a Jan Luts el desarrollo de la técnica de LS-SVM e ICA para el CF

<sup>3</sup>El autor agradece a Alfonso Pérez el desarrollo de la técnica de redes neuronales para el CF



### 3.3. LENGUAJE DE ESPECIFICACIÓN DE CLASIFICADORES BASADO EN XML41

```
<Learning_Algorithm>
    identificador del algoritmo de aprendizaje y clasificación usado
</Learning_Algorithm>
<Release_Date>
    yyyy-mm-dd - fecha en que el clasificador fue desarrollado
</Release_Date>
<Question_to_solve>
    <id>
        xx_vs_xx_vx_xx... - identificador de la pregunta a resolver
    </id>
    <class>
        identificador de clase discriminada por el clasificador
        el tag se repite por cada clase discriminada
    </class>
</Question_to_solve>
<Study_Description>
    texto descriptivo del problema resuelto por el clasificador
</Study_Description>
<Developer_Centre>
    identificador del centro de desarrollo
</Developer_Centre>
<Node_mail>
    x@x - email de contacto del responsable del clasificador
</Node_mail>
<Preprocessing_Method>
    método de preproceso de los casos, si es necesario
</Preprocessing_Method>
<Needed_Data>
    <Data>
        identificador de dato o característica requerida por el clasificador
        el tag se repite por cada característica requerida
    </Data>
</Needed_Data>
<OtherInfo>
    texto para incluir información adicional
</OtherInfo>
</DESCRIPTION>
```

#### profile.xml

```
<PROFILE>
    <Project>
        identificador del proyecto al que pertenece el clasificador
    </Project>
    <Domain>
        identificador del dominio biomédico del clasificador
    </Domain>
    <Classifier_Name>
        identificador del clasificador
    </Classifier_Name>
    <Version>
        xxx - número de versión, en caso de actualización del modelo
    </Version>
    <RANKING_METRICS>
        <STATIC>
```

```

<Accuracy>
  <value>
    porcentaje acierto del clasificador
  </value>
  <Std_deviation>
    desviación típica del porcentaje de acierto del clasificador
  </Std_deviation>
</Accuracy>
<Confusion_matrix>
  matriz de confusión del clasificador
  especificada en estructura
  <Matrix><row><column>x</column></row></Matrix>
</Confusion_matrix>
</STATIC>
<DYNAMIC>
  reservado para evaluación dinámica del clasificador - en desarrollo
</DYNAMIC>
<CONTEXTUAL>
  reservado para información contextual estadística sobre los datos de
  entrenamiento del clasificador - en desarrollo
</CONTEXTUAL>
</RANKING_METRICS>
<USER_STATISTICS>
  reservado para mantener estadísticas de uso del clasificador
  - en desarrollo
</USER_STATISTICS>
</PROFILE>

```

## method.xml

```

<METHOD>
  <Classifier>
    identificador del clasificador
  </Classifier>
  <Actions>
    bajo este tag se especifica la serie de acciones de clasificación
    a ejecutar
    se añade un tag representando a cada acción, que tiene bajo sí la
    configuración de la misma
    <GROUP-NAME>
      el nombre del tag 'GROUP-NAME' es sustituido por el grupo al que
      pertenece la acción entre normalizaciones, transformaciones o
      técnicas de clasificación
    <ACTION-NAME>
      el nombre del tag 'GROUP-NAME' es sustituido por el nombre que
      identifica a la acción de clasificación
    <ID>
      código que identifica a la acción de clasificación en el proceso
      una acción puede tomar sus datos de entrada de los datos de
      salida de otra acción mediante este identificador
    </ID>
    <SF>
      estructura en árbol de los datos de entrada de la acción
      puede acceder a los datos de entrada del clasificador por su
      nombre o a los datos de salida de otra acción usando el código
      identificador

```

### 3.3. LENGUAJE DE ESPECIFICACIÓN DE CLASIFICADORES BASADO EN XML43

```
</SF>
<ACTION-PARAM>
    especifica un parámetro de la acción de clasificación
    el tag se repite por cada parámetro necesario por la
    normalización, transformación o técnica de clasificación
    cada acción dispondrá de su propio conjunto de parámetros
    especificados por el desarrollador de la misma
si se trata de matrices o vectores, se especifican mediante la
    estructura <Matrix><row><column>x</column></row></Matrix>
</ACTION-PARAM>
</ACTION-NAME>
</GROUP-NAME>
</Actions>
<Output>
<SF>
    estructura en árbol de los datos de salida del clasificador
    puede acceder a los datos de salida de las acciones usando su
    código identificador
</SF>
<Extra>
    puede especificar contenido adicional que será devuelto en
    los resultados de clasificación este contenido se estructura en XML
    y será devuelto tal cual
</Extra>
<Legend>
    define una leyenda que identifica las clases usadas en el proceso de
    clasificación con el identificador del concepto real al que representan
    <LegendItem>
        <Code>
            código de clase tomado en el proceso
        </Code>
        <ClassID>
            identificador del concepto de la clase
        </ClassID>
    </LegendItem>
</Legend>
</Output>
</METHOD>
```

A continuación se describen las especificaciones XML de algunas transformaciones, normalizaciones y técnicas de clasificación incluidas en el CF.

#### Codificación

```
<Transformation>
  <Codification>
    <ID>
      identificador
    </ID>
    <SF>
      datos de entrada
    </SF>
    <Data>
      especifica los posibles valores y codificaciones para cada
      dato de entrada
      el tag se repite por cada dato de entrada
```

```

<pos>
  posición del dato en la lista de datos de entrada
</pos>
<Option>
  especifica un posible valor y su correspondiente codificación
  para el dato de entrada
  el tag se repite por cada valor posible
  <Value>
    valor del dato de entrada al que se le asignara la
    siguiente codificación
  </Value>
  <Code>
    codificación a asignar al valor
    especificada en estructura
    <Matrix><row><column>x</column></row></Matrix>
    una columna por cada dígito
  </Code>
</Option>
</Data>
</Codification>
</Transformation>

```

## PCA

```

<Transformation>
  <PCA>
    <ID>
      identificador
    </ID>
    <SF>
      datos de entrada
    </SF>
    <MatrixTransformation>
      matriz de transformación
      especificada en estructura
      <Matrix><row><column>x</column></row></Matrix>
    </MatrixTransformation>
    <MeanTrainData>
      media de las muestras de entrenamiento
      especificada en estructura
      <Matrix><row><column>x</column></row></Matrix>
    </MeanTrainData>
  </PCA>
</Transformation>

```

## Peak Integration

```

<Transformation>
  <PeakIntegration>
    <ID>
      identificador
    </ID>
    <SF>
      datos de entrada
    </SF>

```

### 3.3. LENGUAJE DE ESPECIFICACIÓN DE CLASIFICADORES BASADO EN XML45

```
<Peaks>
  conjunto de picos a integrar
  <Peak dataType="ppm">
    el atributo dataType indica la escala del eje x, ppm es usado en
    señales espectrales como MRS
    el tag se repite por cada pico
    <Name>
      especifica un nombre al pico
    </Name>
    <Center>
      el punto central del pico a integrar, en la escala dada
    </Center>
    <Interval>
      rango que integrará con el centro en <Center>, en la escala dada
    </Interval>
  </Peak>
</Peaks>
</PeakIntegration>
</Transformation>
```

#### Normalización lineal

```
<Normalization>
  <Linear>
    <ID>
      identificador
    </ID>
    <SF>
      datos de entrada
    </SF>
    <Max>
      para cada dato de entrada especifica el valor máximo
      para su normalización
      especificada en estructura
      <Matrix><row><column>x</column></row></Matrix>
      una columna por cada dato
    </Max>
    <Min>
      para cada dato de entrada especifica el valor mínimo
      para su normalización
      especificada en estructura
      <Matrix><row><column>x</column></row></Matrix>
      una columna por cada dato
    </Min>
  </Linear>
</Normalization>
```

#### KNN

```
<Technique>
  <KNN>
    <ID>
      identificador
    </ID>
    <SF>
      datos de entrada
```

```

</SF>
<Distance>
  tipo de distancia usada para el cálculo de proximidad, por ejemplo L1
</Distance>
<k>
  parámetro k, indica número de vecinos más próximos
</k>
<TDF>
  especifica las muestras de entrenamiento
  <Data>
    matriz de muestras de entrenamiento
    especificado en estructura
    <Matrix><row><column>x</column></row></Matrix>
    una fila por muestra, una columna por característica
  </Data>
  <Labels>
    clases de las muestras de entrenamiento
    especificado en estructura
    <Matrix><row><column>x</column></row></Matrix>
    una fila por muestra, donde una única columna especifica la clase
  </Labels>
  <Identifiers>
    identificadores de las muestras de entrenamiento
    especificado en estructura
    <Matrix><row><column>x</column></row></Matrix>
    una fila por muestra, donde una única columna especifica
    el identificador
  </Identifiers>
</TDF>
</KNN>
</Technique>

```

## Fisher LDA

```

<Technique>
  <LDA>
    <ID>
      identificador
    </ID>
    <SF>
      datos de entrada
    </SF>
    <Sigma>
      matriz de covarianzas común
      especificada en estructura
      <Matrix><row><column>x</column></row></Matrix>
    </Sigma>
    <Priori>
      probabilidades a priori de las clases
      especificado en estructura
      <Matrix><row><column>x</column></row></Matrix>
      una fila por clase, probabilidad indicada en única columna
    </Priori>
    <W>
      matriz de transformación
      especificada en estructura

```

```

    <Matrix><row><column>x</column></row></Matrix>
    tantas filas como número de datos de entrada + 1
    tantas columnas como fronteras entre clases
</W>
<Boundaries>
    especifica las coordenadas de las fronteras entre clases
    <B>
        el tag se repite por cada frontera
        <From>
            identificador de una de las clases separadas
        </From>
        <To>
            identificador de la otra clase separada
        </To>
        <origin>
            punto origen de la frontera (actualmente solo 1D o 2D)
            especificado en estructura
            <Matrix><row><column>x</column></row></Matrix>
        </origin>
        <destiny>
            punto por el que pasa la frontera (actualmente solo 1D o 2D)
            especificado en estructura
            <Matrix><row><column>x</column></row></Matrix>
        </destiny>
    </B>
</Boundaries>
<Prototypes>
    especifica los centroides de cada clase
    <P>
        el tag se repite por cada centroide
        <class>
            identificador de la clase
        </class>
        <Coordinates>
            coordenadas del centroide
            especificadas en estructura
            <Matrix><row><column>x</column></row></Matrix>
        </Coordinates>
    </P>
</Prototypes>
<FDF>
    matriz que especifica los coeficientes de Fisher
    especificada en estructura
    <Matrix><row><column>x</column></row></Matrix>
    una fila por dato de entrada
    una columna por clase
</FDF>
</LDA>
</Technique>

```

### 3.4. Entrada y salida XML

Al igual que el lenguaje de especificación de clasificadores, en el CF la entrada y salida se lleva a cabo mediante mensajes XML. Esto significa que la solicitud de una clasificación al CF se efectúa mediante un mensaje XML indicando el nombre del clasificador a ejecutar y los datos de entrada. Los resultados

de clasificación se devuelven igualmente en un fichero XML que contiene las probabilidades de cada clase así como datos adicionales.

A continuación se describen los mensajes XML de entrada y salida del CF, definidos en los ficheros `input.xml` y `output.xml` respectivamente:

### input.xml

```
<INPUT>
  <Classifier>
    identificador del clasificador a ejecutar
  </Classifier>
  <User_login>
    identificador de usuario - no usado actualmente
  </User_login>
  <User_password>
    contraseña de usuario - no usado actualmente
  </User_password>
  <Data_identifier>
    identificador del caso a clasificar - no usado actualmente
  </Data_identifier>
  <Data>
    <DATA-NAME>
      el nombre del tag se sustituye por el nombre del dato
      valor del dato o característica del caso requerido por el clasificador
      el tag se repite por cada característica requerida
    </DATA-NAME>
  </Data>
</INPUT>
```

### output.xml

```
<OUTPUT>
  <ClassifierName>
    identificador del clasificador
  </ClassifierName>
  <Answer>
    estructura en árbol de los datos de salida del clasificador especificados
    en el tag <output> del fichero method.xml
    siempre contiene las probabilidades a posteriori en la forma que sigue:
    <Probabilities>
      <Probabilities>
        especificada en estructura
        <Matrix><row><column>x</column></row></Matrix>
        con una fila por cada clase
      </Probabilities>
    </Probabilities>
    <Labels>
      especifica los identificadores de cada clase en el orden que han
      sido proporcionadas las probabilidades
      están mapeados con su concepto en la leyenda
    </Labels>
  </Answer>
  <Legend>
    copia la leyenda definida en el tag <Legend> del fichero method.xml
  </Legend>
  <Extra>
```



```
        copia la información adicional definida en el tag <Extra> del
        fichero method.xml
    </Extra>
</OUTPUT>
```

La API del CF proporciona un conjunto de clases para trabajar con las entradas y salidas haciendo transparente el parseado o construcción de los mensajes XML. De este modo se facilita el uso del CF a los desarrolladores de clientes, permitiendo una fácil creación de entradas de clasificación y facilitando el tratamiento de los resultados.

## 3.5. Resultados

Con el CF se ha desarrollado un *framework* de PR genérico para proporcionar servicios de ayuda a la decisión a plataformas software, como DSS o aplicaciones clínicas. En el CF se han incluido métodos de PR implementados por diferentes desarrolladores, lo que demuestra su escalabilidad para ser ampliado con nuevos métodos. El CF se desarrolló como motor de clasificación para el dDSS de HEALTHAGENTS. Fue integrado en el sistema de clasificación distribuido de HEALTHAGENTS siendo accesible por los agentes de clasificación disponibles en los nodos de clasificación. Así, permitió la incorporación de 25 clasificadores de tumores cerebrales (Tabla 2.1). Como se describe en el capítulo 4, el CF ha servido igualmente como sistema de clasificación del DSS Curiam, dando lugar a varios CDSS para diferentes dominios biomédicos.

Respecto al rendimiento, el CF ha mostrado unos resultados correctos. El rendimiento está relacionado directamente con la capacidad de proceso del equipo en el que el sistema está ejecutándose, por lo tanto cuanto más rápido sea el sistema, mayor rendimiento tendrá el CF. Para obtener una medida de los tiempos de clasificación del CF, se llevaron a cabo un conjunto de pruebas en un PC de doble núcleo a 2.2Ghz, con 2GB de memoria RAM. Las pruebas consistieron en ejecutar 1000 clasificaciones de casos de tumores cerebrales usando aleatoriamente todos los clasificadores incluidos en el sistema de HEALTHAGENTS. Este test permitió la ejecución de las diferentes rutinas tomadas por la combinación de métodos de PR incluidos en el framework. Las pruebas mostraron una media de 125 ms por clasificación.



# Capítulo 4

## Curiam DSS

### 4.1. Motivación

Desde los comienzos del grupo de minería de datos biomédicos del grupo IBIME, se ha pretendido contribuir en la mejora de la asistencia sanitaria ofreciendo soluciones que permitan trasladar al uso clínico real los resultados de la investigación aplicada a la extracción de conocimiento de datos biomédicos. En [18] se mostró un primer prototipo de CDSS, llamado Sistema de Orientación Clínica (SOC), en el que se accedía a servicios web para la clasificación de datos de diferentes tareas médicas. Pero los continuos avances en la minería de datos biomédicos y en las tecnologías de la información, así como la obtención de conocimiento en nuevas áreas de diagnóstico clínico, requirieron el desarrollo de un nuevo sistema más escalable, robusto y mantenible, que pudiera ser especializado en dominios biomédicos específicos y ampliado con nuevas funcionalidades con el mínimo esfuerzo. La llegada del proyecto HEALTHAGENTS supuso el desarrollo dentro del consorcio de un CDSS para el diagnóstico de tumores cerebrales. Como trabajo del grupo IBIME, y en el marco de esta tesina, se desarrolló el framework o motor de clasificación de HEALTHAGENTS, el CF, con el objetivo de que no sólo pudiera aplicarse al diagnóstico de tumores cerebrales, sino a la clasificación genérica de cualquier tipo de datos biomédicos. Esto motivó el desarrollo del nuevo CDSS que, cumpliendo con las características del CF, fuera un sistema adaptable a diferentes dominios biomédicos y pudiera ser fácilmente actualizado a los requerimientos que el estado del arte en CDSS motivase.

Según [41], los DSS deben estar compuestos por tres componentes principales: (1) interfaz con los repositorio de datos, la información disponible y el conocimiento del dominio; (2) modelos de ayuda a la decisión soportados por un sistema de gestión de modelos; y (3) una potente, y a la vez que simple, GUI que permita interactividad y generación de informes. El diseño del nuevo CDSS propuesto en esta tesina, Curiam, toma en consideración estas premisas. En primer lugar se permite el acceso a repositorios de datos locales o distribuidos. El conocimiento del CDSS es tratado en forma de ontología de dominio o repositorio de conceptos, pudiendo ser extendido en cualquier momento. Los modelos predictivos son gestionados por el CF. La GUI es flexible a la hora de proporcionar información adaptada al dominio a tratar, y proporciona información avanzada sobre los resultados de los modelos. Además, el sistema permite la generación de informes clínicos basados de los resultados de clasificación. Finalmente, el sistema proporciona resultados de varios modelos simultáneamente para lograr un mayor consenso en la ayuda a la decisión proporcionada. Por esta razón, el CDSS se denominó Curiam, que proviene de la expresión latina "per curiam", la cual significa decisión en consenso.

### 4.2. Diseño arquitectónico

El objetivo de Curiam es ser un DSS extensible y adaptable, por lo que un correcto diseño es fundamental para tal fin. Además, un buen diseño facilita la mantenibilidad del sistema y así mismo puede mejorar la robustez del mismo. Curiam ha sido diseñado mediante un arquitectura de tres capas<sup>4.1</sup>. La arquitectura de tres capas separa los componentes dedicados a la persistencia del sistema, la lógica de la aplicación, y la interfaz gráfica de usuario. Además, el diseño de Curiam contempla una capa transversal

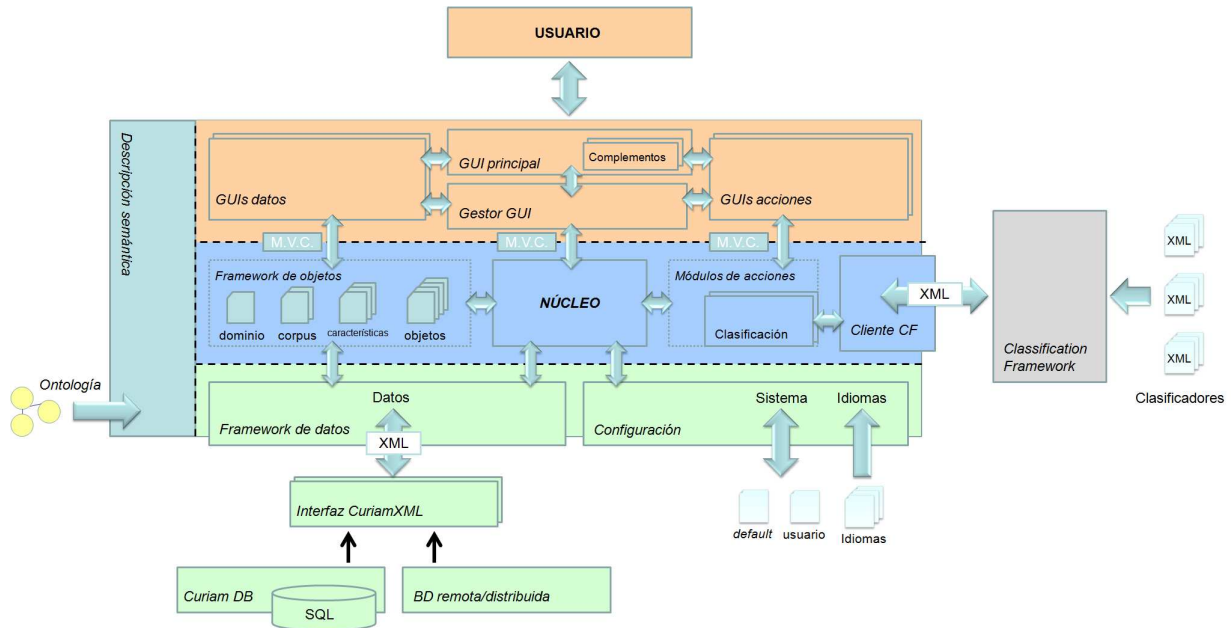


Figura 4.1: Diseño arquitectónico de Curiam. Curiam está desarrollado en una arquitectura de tres capas más una capa vertical de conocimiento.

que otorga una descripción semántica a los conceptos tratados en la aplicación. El objetivo de esta capa es proporcionar significado a los conceptos que el usuario trata para obtener ayuda a la decisión. Su primera aproximación se basa en adquirir el conocimiento del dominio de la aplicación del CDSS y relacionarlo con los posibles diagnósticos de los modelos predictivos a la hora de proporcionar ayuda a la decisión.

En esta sección se describen los principales componentes que forman la arquitectura de Curiam entre las tres capas. Las secciones siguientes describen en más detalle los componentes relacionados con el acceso a datos, clasificación, y la interfaz gráfica de usuario. Además, como futura línea de investigación relacionada con Curiam, la sección 4.6 introduce una primera aproximación sobre una visualización conjunta de resultados de clasificación basada en la ontología de dominio.

La capa lógica de Curiam consta de tres componentes principales: un sistema de objetos, un sistema de módulos de ejecución, y un núcleo gestor de la aplicación.

En primer lugar describiremos el sistema de objetos. Curiam es aplicable a diferentes dominios biomédicos, por lo tanto se tiene un objeto principal que representa el dominio actual de la aplicación, bajo el cual se organizan el resto de datos relativos al mismo. Como CDSS de diagnóstico, los datos básicos con los que Curiam trata son los diferentes elementos susceptibles de ser clasificados. Estos elementos corresponden a los casos médicos almacenados en una determinada base de datos. Cada uno de estos casos está formado por un conjunto de características que definen el propio contenido del caso, o lo que es lo mismo, los datos del paciente. Se han definido cuatro tipos de posibles características: numérica, categórica, vector de datos, e imagen. Consecuentemente, estas características son las utilizadas por el CDSS para la extracción de conocimiento, por lo que deben ser trasladadas al correspondiente sistema de clasificación para la clasificación de un caso. A continuación se describen los cuatro tipos de característica:

**Numérica:** consiste en un único valor numérico dentro de un determinado rango. La descripción de la misma indica el tipo de unidad. Pueden utilizarse para medidas como la tensión arterial de un paciente o su edad.

**Categórica:** una característica categórica consiste en un valor discreto seleccionado de entre un grupo de posibles valores. A priori los valores se tratan como cadenas de texto y el sistema de clasificación es el encargado de codificar dichos valores para su computación numérica. Como ejemplos de utilización, pueden indicar posibles formas de una masa tumoral o el nivel de expresión de un determinado gen.

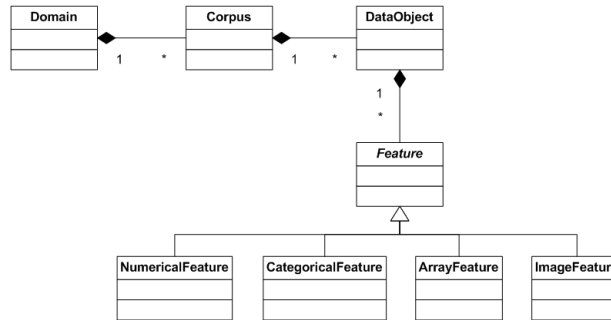


Figura 4.2: Diagrama UML que representa el modelado de datos lógicos en Curiam.

Vector de datos: una característica en forma de vector de datos consiste en una serie de valores numéricos o discretos. Según el dominio de aplicación, dicha característica puede ser interpretada de diversas formas, aunque generalmente se tratará como una señal o serie numérica. Puede representar señales médicas como adquisiciones de encefalograma o espectros de resonancia magnética. Esta característica puede llevar asociada la variable independiente sobre la que se define la señal (p.e. tiempo, frecuencia). Esta característica puede llegar a representar texto plano para aplicaciones de minería de texto.

Imagen: a priori una imagen puede ser representada mediante una característica de vector de datos, pero para facilitar su tratamiento se ha incluido como característica independiente. Representa pues, una adquisición en forma de imagen.

Por lo tanto, un caso está representado de un conjunto de características. Dentro de un dominio biomédico, podemos disponer de diferentes corpora representados con diferentes conjuntos de características — es decir, los casos de un corpus solo pueden disponer como máximo de ese conjunto de características —. Sobre estos corpora, se puede estar interesado en aplicar diferentes tipos de extracción de conocimiento. Esta organización se representa en Curiam agrupando en diferentes corpus los casos de un dominio. De esta forma, se puede acceder a diferentes tareas de clasificación o a diferentes bases de datos biomédicas según el corpus seleccionado. La figura 4.2 ilustra el modelado del sistema de objetos en Curiam. Bajo el dominio, se organizan los conjuntos de datos o corpora y bajo estos los diferentes casos, representados por el objeto `DataObject`. Los casos están formados por conjuntos de características, definidas por la interfaz `Feature`.

Curiam está concebido como CDSS principalmente para diagnóstico, si bien está preparado para pronóstico y tratamiento. Esto implica que el principal caso de uso del sistema es la ejecución de clasificaciones sobre casos médicos para obtener ayuda en su diagnóstico. Pero como ya se ha comentado, se pretende hacer de Curiam un sistema extensible a nuevas funcionalidades para mejorar la ayuda a la toma de decisiones. Por esta razón, se ha diseñado un sistema de acciones ejecutables de las que el sistema puede hacer uso y cuyo número puede ser extendido por nuevos desarrollos. Este sistema de módulos de acciones está dirigido por la interfaz `WorkAction` siguiendo el patrón de diseño 'estrategia' — el cual, como se vio en la sección 3.2, encapsula procedimientos mediante una interfaz —. Dicha interfaz será implementada por las nuevas acciones a desarrollar. Consecuentemente, la clasificación en Curiam se lleva a cabo mediante una implementación de `WorkAction`, la cual se denomina `ClassificationWorkAction`. Los detalles del sistema de clasificación de Curiam son descritos en la sección .

La capa de lógica contiene también el núcleo de la aplicación. El núcleo, representado por la clase `BMGDSS`, gestiona las relaciones entre los componentes del sistema. Hace de creador de instancias mediante métodos de fabricación, para el inicio del sistema o cuando se requiera establecer un nuevo módulo en memoria. Por ejemplo, al iniciar el sistema, el núcleo pone en marcha los componentes de acceso a datos, cargando en memoria aquellos objetos requeridos al comienzo de la aplicación. El acceso a los objetos del sistema de objetos de Curiam se realiza mediante el núcleo. Como otro ejemplo, la GUI puede requerir el lanzamiento de un módulo de acción. Para esto se realiza la petición al núcleo el cual pone dicho módulo en funcionamiento y provee el acceso a sus correspondientes componentes gráficos. Además, el núcleo proporciona el acceso a la configuración de Curiam, ya sea para las propiedades del sistema como para acceder a los idiomas de la aplicación.

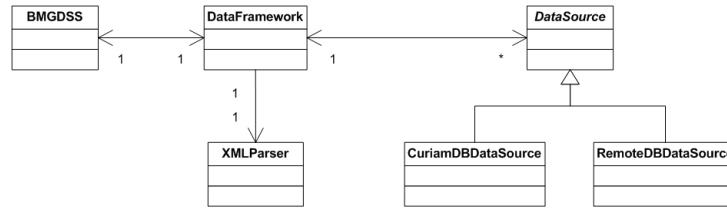


Figura 4.3: Diagrama UML del sistema de acceso a datos en Curiam.

La capa de datos de Curiam está dividida entre el componente de acceso a datos, o *framework* de datos, y el componente de configuración. El framework de datos provee a Curiam del acceso a la persistencia de los objetos y a la configuración del dominio biomédico. Éste es descrito en la sección 4.3. Por otro lado, el componente de configuración gestiona la configuración del sistema y los idiomas en los que Curiam puede ser visualizado. Tanto la configuración como los idiomas, son almacenados en ficheros de propiedades de tipo clave-valor. Los ficheros de idiomas contienen las traducciones de los elementos comunes de Curiam, sin embargo, la adaptación un dominio biomédico, provee además las traducciones de los términos específicos del dominio que se utilicen en los componentes específicos del dominio. En todo caso, los componentes gráficos de Curiam acceden al contenido por medio de su clave, y a la hora de representarlos de cara al usuario comprueban el valor correspondiente al idioma actual. En cuanto a la configuración del sistema, se han definido dos niveles de configuración: configuración por defecto y configuración de usuario. Los ficheros de configuración proporcionan valores de configuración estáticos o dinámicos. Los valores estáticos pueden ser definidos a la hora de empaquetar el software, por ejemplo los valores de ciertas cadenas de texto que representen títulos de componentes gráficos, o tamaños predefinidos de dichos componentes para un dominio en particular. Estos valores suelen ser definidos en el fichero de configuración por defecto, el cual es de solo-lectura y por tanto no es accesible por el usuario final. Los valores dinámicos son valores que pueden ser establecidos por el usuario a lo largo de la vida del sistema, como por ejemplo la identificación de usuario o el último directorio accedido. Estos valores se almacenan cada vez que el sistema es cerrado, y se definen en el fichero de propiedades de usuario.

Finalmente, la capa de GUI de Curiam es la encargada de proporcionar al usuario la interfaz para interactuar con el CDSS y para recibir la correspondiente información. Básicamente, la GUI es gestionada por componente de gestión de GUI, y se construye mediante los componentes de GUI principal, GUI para datos, y GUI para acciones. Éstas dos últimas son además extensibles según el dominio de aplicación y la existencia de nuevas acciones. Además, se pueden incluir extensiones adicionales en forma de complementos de la GUI principal. La interfaz gráfica de usuario de Curiam es descrita con detalle en la sección 4.5.

### 4.3. Acceso a datos

Todo DSS necesita de una interfaz con repositorios de datos para llevar a cabo la extracción de conocimiento. Especialmente en los CDSS, dicha interfaz debe proporcionar flexibilidad para la interconexión con sistemas de historia clínica en los centros clínicos. Con este fin, Curiam gestiona el acceso a los repositorios de datos mediante el intercambio de información en XML. Esto permite que sistemas de información diseñados con diferentes tecnologías o estándares, puedan ser accedidos desde el CDSS a través de una capa que transforme su interfaz a XML. La figura 4.3 muestra el modelado del sistema de acceso a datos en Curiam. Desde el núcleo del sistema (clase `BMGDSS`) se tiene acceso a la clase que gestiona los accesos a datos, la clase `DataFramework`. Esta clase puede utilizar varios accesos de datos simultáneos, a través de la clase abstracta `DataSource`, que proporciona una interfaz de métodos de intercambio de mensajes — mediante XML — a implementar por los diferentes accesos.

Al margen de los propios datos, es necesario establecer una descripción lógica de ellos mismos para un dominio biomédico en particular. Esta información también puede ser accedida mediante XML. Sin embargo, con el objetivo de facilitar el uso e instalación de Curiam en la mayoría de centros clínicos o de investigación, se desarrolló dentro del grupo IBIME una meta-base de datos capaz de albergar al mismo tiempo la descripción lógica de los datos y los propios datos. Esta base de datos fue implementada

mediante el sistema gestor de bases de datos MySQL. Se desarrolló la correspondiente interfaz de acceso a datos para permitir a Curiam el acceso a la base de datos mediante los mensajes XML. El módulo desarrollado, `CuriamBDDataSource`, traduce las consultas y escrituras de XML a SQL, y devuelve los resultados transformando los correspondientes *ResultSet*, o conjuntos de resultados, a XML.

Actualmente, se está trabajando en proporcionar a Curiam el acceso a sistemas de información hospitalaria distribuidos a través de motores de integración de sistemas de información clínica. Esto permitirá llevar a cabo tareas de ayuda a la decisión desde diferentes puntos de un centro clínico usando datos distribuidos y heterogéneos del sistema de información del hospital.

## 4.4. Sistema de clasificación

El sistema de clasificación en Curiam está basado en el *framework* de clasificación surgido del desarrollo en el proyecto HEALTHAGENTS (capítulo 3), el CF. De este modo el CF proporciona a Curiam la capacidad de clasificación de datos de diferentes dominios biomédicos y la posibilidad de incluir múltiples modelos que empleen diferentes técnicas de PR.

La propia librería del CF proporciona la interfaz para que clientes como Curiam se comuniquen con el *framework*. Esta interfaz permite por un lado que Curiam obtenga listados de los clasificadores disponibles — es decir, el conjunto de clasificadores representados por las especificaciones XML a los que el CF tenga acceso — y por otro la propia clasificación de un caso.

En Curiam, existe una capa vertical que regula que la semántica del sistema sea consistente 4.1. La implicación de esta capa con el sistema de clasificación consiste en que los conceptos o términos utilizados por los clasificadores del CF — ya sea para especificar los diagnósticos tratados, o los tipos de datos utilizados — son los mismos que el sistema de datos de Curiam emplea, y contemplan las mismas relaciones semánticas entre ellos. Esto es esencial para dos situaciones: la elección de clasificadores apropiados para un determinado caso, y la representación de los resultados de modelos diferentes de forma simultánea. La primera de ellas se describe a continuación, y la segunda, actualmente en experimentación, en la sección 4.6.

El proceso de clasificación se ha definido en Curiam de forma que el usuario sea libre de escoger cómo realizar una clasificación de entre los modelos disponibles guiado por un sistema de búsqueda sobre los clasificadores disponibles para un dominio biomédico. Curiam entiende como compatibles con el caso a clasificar aquellos cuyas características requeridas para su ejecución están disponibles para el caso de estudio. Un médico acude a un CDSS en las dos situaciones siguientes: cuando tiene dudas a la hora de diagnosticar un caso y para confirmar un diagnóstico. En el primer caso, el médico requerirá resolver determinadas preguntas clínicas, distinguiendo entre dos o más diagnósticos posibles. Incluso puede solicitar una recomendación de entre todos los posibles diagnósticos. Para el segundo caso, el médico simplemente indicará cual es su hipótesis de diagnóstico y solicitará una confirmación por parte del CDSS. Curiam contempla ambas situaciones utilizando un sistema de selección de clasificadores.

El sistema de clasificación de Curiam permite en primer lugar seleccionar uno o más diagnósticos, o clases, de entre el conjunto de todas aquellas que son discriminadas por los clasificadores del dominio compatibles con los datos seleccionados. Esto incluye, además de clases, posibles agrupaciones o propiedades de las mismas, siempre y cuando existan clasificadores capaces de discriminar dichas agrupaciones o propiedades, o incluso clases que pertenezcan a las mismas. Tras cada cambio en la selección de clases, se mostrarán todas las preguntas a resolver 4.4 — igualmente de entre el conjunto de clasificadores — capaces de discriminar cada una de las clases seleccionadas de forma individual. El médico seleccionará aquellas preguntas en las que esté interesado, por defecto todas dado la selección previa de clases, y solicitará la clasificación. Se ejecutarán todos los clasificadores correspondientes a las preguntas seleccionadas. Esta selección semántica de clasificadores se desarrolló en [36]. En dicho trabajo, se utilizó como ontología de dominio una versión propia de la taxonomía de tumores cerebrales WHO [28], en donde se incluyeron las clases de tumor y sus propiedades. Para representar los clasificadores, se creó otra ontología de clasificadores a los cuales se les asignaban las propiedades de discriminar una determinada clase o propiedad del dominio. Según la selección de clases y propiedades tomada por el usuario, se creaba de forma dinámica una consulta SPARQL para la obtención de los clasificadores apropiados. La incorporación de la selección semántica de clasificadores en Curiam esta prevista dentro del próximo trabajo futuro, sin embargo, la

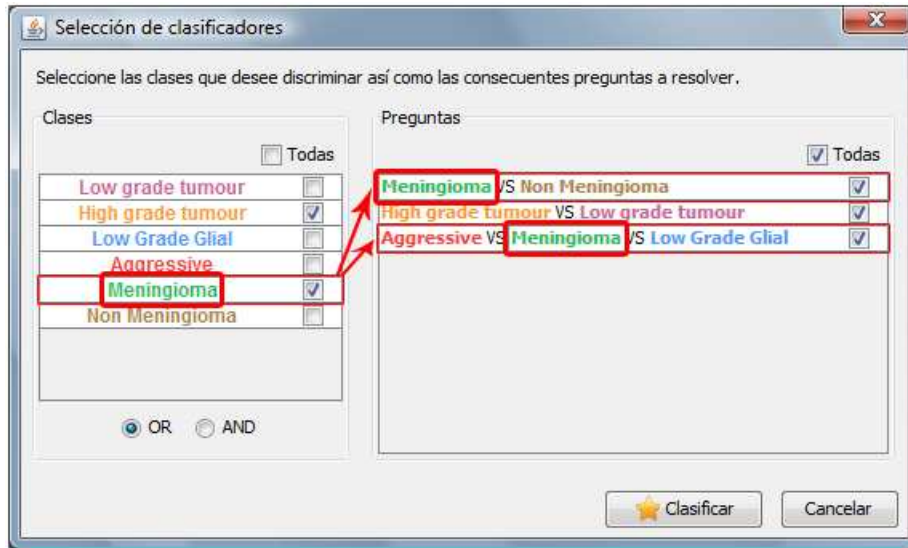


Figura 4.4: Cuadro de diálogo de selección de clasificadores Curiam Al seleccionar la clase 'Meningioma' en la lista de clases, aquellas preguntas capaces de discriminarla son mostradas en la lista de preguntas.

versión actual de Curiam lleva a cabo el mismo sistema de selección de clasificadores pero comprobando de forma exacta — no semántica — las clases, agrupaciones o propiedades seleccionadas con las posibles preguntas a resolver. Por ejemplo, si un usuario selecciona la confirmación de que un caso pertenece a una agrupación, son mostradas aquellas preguntas que discriminen directamente dicha agrupación, y no las clases o propiedades de la misma — lo cual sí es llevado a cabo mediante el sistema de selección semántica.

Una vez se haya determinado qué clasificadores ejecutar, Curiam envía el caso a clasificar junto con el listado de clasificadores seleccionados al CF. El CF ejecuta cada uno de los clasificadores indicados y una vez haya obtenido sus resultados, estos son devueltos a Curiam. Finalmente Curiam interpreta los resultados recibidos y los muestra al usuario mediante la GUI de la acción de ejecución de clasificación, en donde se puede comprobar de forma detallada cada uno de los resultados obtenidos por cada uno de los clasificadores. La GUI de resultados de clasificación es descrita en la sección siguiente.

## 4.5. Interfaz gráfica de usuario

La interfaz gráfica de Curiam puede considerarse uno de los bloques más importantes de Curiam, ya que es el que más capacidades de extensión requerirá a la hora de adaptar Curiam a dominios biomédicos específicos o de incluir nuevas funcionalidades para el usuario — al margen del desarrollo de clasificadores para el sistema de clasificación —. Para este fin, se han establecido dos focos para el diseño del modelado de la GUI según el tipo de extensibilidad. Por un lado, se tienen los componentes gráficos de las características (o variables) de un caso, en donde, por ejemplo, según el dominio, un vector de datos puede tratarse de diferente manera. Por otro lado, se dispone de componentes extensibles para el desarrollo de nuevas funcionalidades. De forma secundaria, se permite la inclusión de complementos adicionales que puedan interactuar con diferentes partes de la interfaz gráfica y sean accedidos desde el menú o barras de herramientas.

Los componentes que representan características de una muestra constan de un modelado equivalente al del modelo lógico de datos en Curiam (Figura 4.5). Del mismo modo, se tiene una clase `FeatureComponent` que representa de forma genérica un componente de característica. Posteriormente, se tienen los componentes relacionados con cada uno de los tipos de características ya conocidas: `NumericalFeatureComponent` para características de tipo numérico, `CategoricalFeatureComponent` para características de tipo categórico, `ArrayFeatureComponent` para características de tipo vector de datos, e `ImageFeatureComponent` para características de imagen. Ahora, cuando un dominio requiera un componente gráfico específico para una de estas características, puede generalizar uno de estos componentes mediante una nueva clase y asociarla



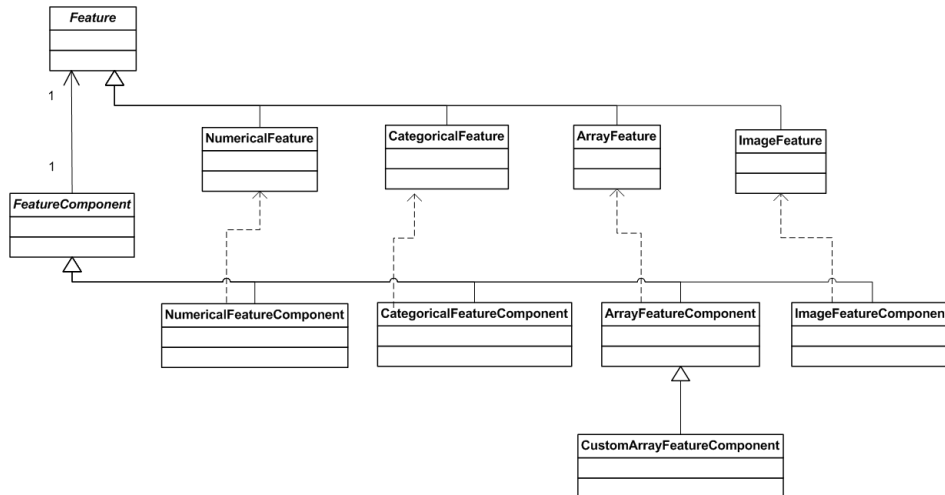


Figura 4.5: Diagrama UML de los componentes GUI relacionados con los datos en Curiam.

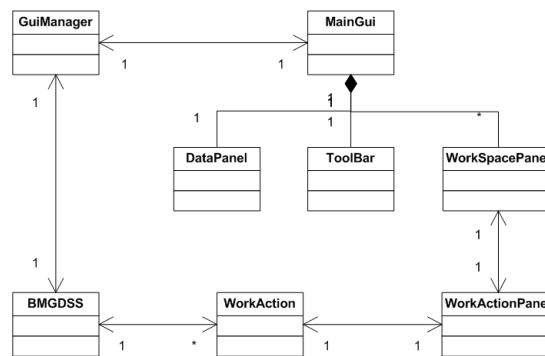


Figura 4.6: Diagrama UML que representa las acciones de ejecución asociadas con sus componentes GUI.

al dominio. Este puede ser el caso del uso de la característica de vector de datos como una señal de MRS, en dónde se requiere una representación gráfica específica. Además, estas componentes personalizadas pueden utilizar cualquier clase o librería externa para cumplir propósitos específicos. Para introducir un nuevo caso en Curiam, es necesario introducir los valores de sus características. Esto es sencillo al tratarse de características numéricas o categóricas, en donde simplemente hay que cumplir las restricciones de rango o valores. Sin embargo, características de vector de datos o imagen pueden requerir algún tipo de procesamiento especial. Por ejemplo, para cargar una señal en una característica de vector de datos puede ser requerido un pre-proceso previo en base a una metodología de pre-proceso definida. Esto puede llevarse a cabo en el desarrollo del componente gráfico de característica personalizada.

Como ya se describió en la sección 4.1, Curiam pretende ser un sistema que pueda incorporar nuevas funcionalidades, y para ello se diseñó un módulo que permita la inclusión de nuevas acciones de ejecución. Dichas acciones, obviamente necesitan de un componente gráfico para permitir la interacción con el usuario. En este caso, el componente lógico `WorkAction` lleva asociado un componente gráfico `WorkActionPanel` (Figura 4.6). Estos componentes pueden tener la estructura interna o acceso a clases externas que el desarrollador desee para proporcionar el funcionamiento deseado. Para visualizar en la GUI de Curiam los componentes gráficos de acciones, se tiene el componente `WorkspacePanel`. Este componente hace de espacio de trabajo, donde se irán añadiendo pestañas sucesivas por cada acción ejecutada, sean del tipo que sean. Así, por un lado se visualizarán los datos del sistema en un panel de datos (`DataPanel`), y por otros los resultados de las acciones realizadas sobre ellos.

Se ha introducido el modelado de la GUI de Curiam. A continuación se describe la GUI de Curiam tal y como es proporcionada al usuario. Para ello nos apoyaremos de capturas de pantalla para mostrar

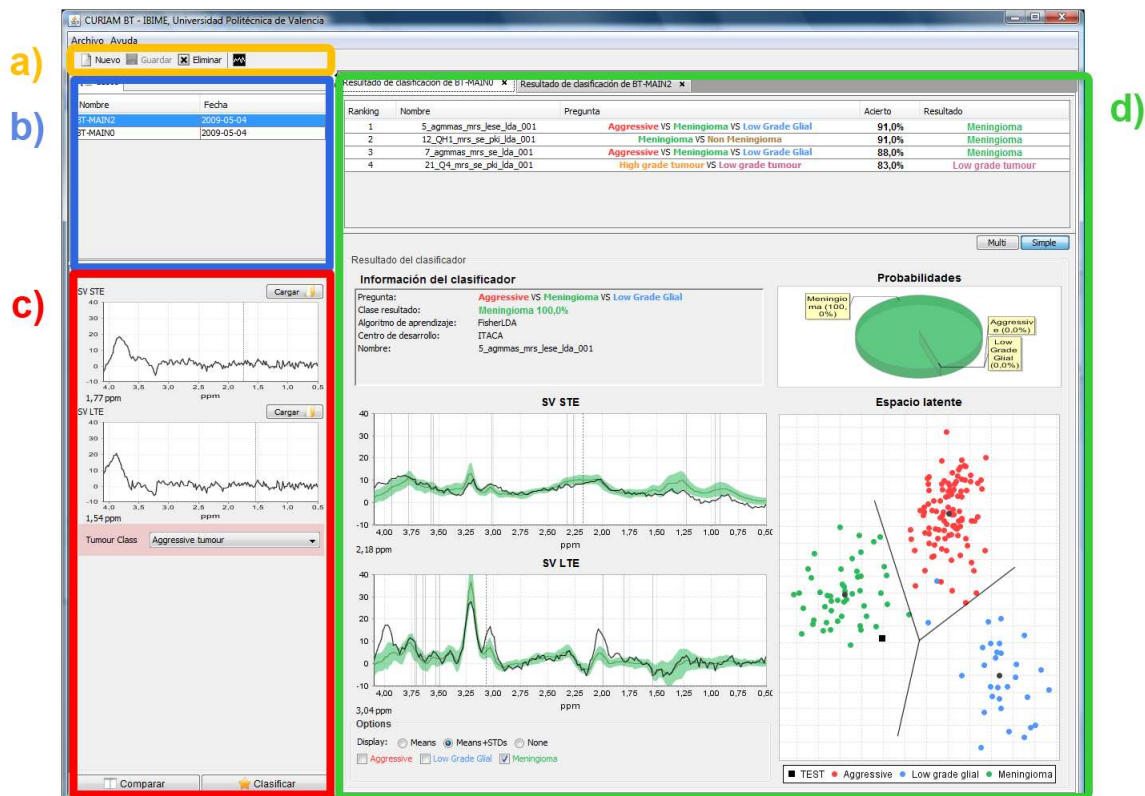


Figura 4.7: Espacio de trabajo de Curiam.

los componentes genéricos de Curiam en el marco de la aplicación Curiam BT, versión de Curiam para el diagnóstico de tumores cerebrales mediante datos de MRS.

La GUI de Curiam se divide en cuatro áreas principales (Figura 4.7):

- Barra de herramientas. Permite que el usuario realice acciones sobre los datos del sistema o acciones personalizadas por los complementos GUI desarrollados.
- Listado de casos. Permite al usuario la navegación entre los casos del sistema de información. Resalta los casos no guardados o modificados.
- Panel de datos. Muestra las características asociadas al caso seleccionado en el listado de casos. En el ejemplo, los casos están compuestos de dos espectros de RM y una variable categórica que indica el tipo de tumor. Desde la parte inferior del panel de datos se pueden lanzar las acciones de ejecución sobre el caso, como puede ser la clasificación o comparación de casos.
- Espacio de trabajo. En este área se muestran los resultados de las acciones realizadas sobre los casos. Cada resultado de acción se muestra en forma de pestaña y se podrá navegar entre ellos. El ejemplo muestra un resultado de clasificación.

La principal tarea de Curiam es la de ofrecer recomendaciones en el diagnóstico a través de la clasificación de casos mediante los modelos predictivos disponibles. A continuación describiremos la GUI ofrecida como resultado de las acciones de clasificación. Una vez clasificado un caso, se añade al espacio de trabajo una pestaña de vista de resultados de clasificación. La vista de resultados consta de tres partes:

- Listado de resultados de clasificadores. Lista los resultados obtenidos por todos los clasificadores ejecutados, ordenados por la tasa o porcentaje de acierto. Esta lista permite obtener una rápida interpretación de los resultados. Se muestran las clases o agrupaciones ganadoras por cada clasificador y se les asigna colores que les identifique. El usuario puede hacerse un

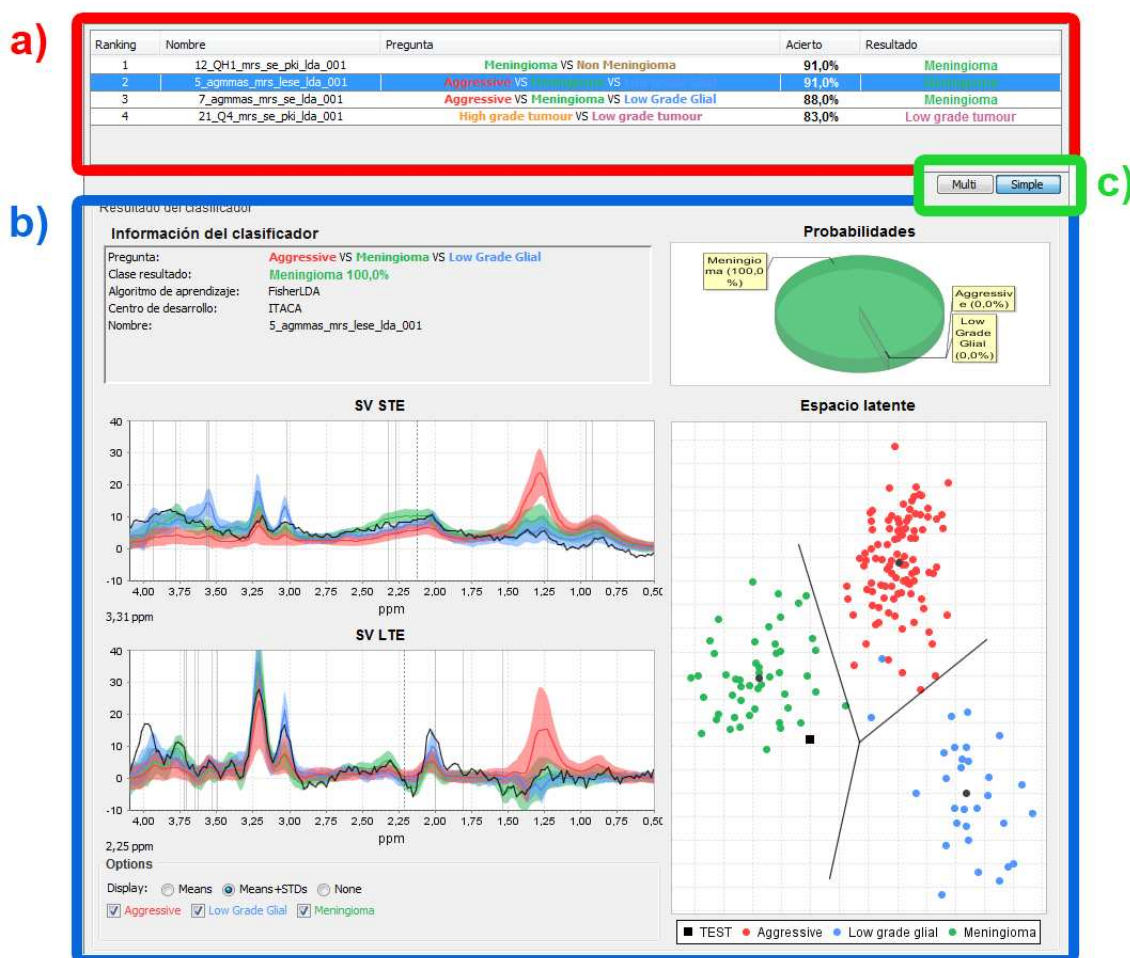


Figura 4.8: GUI de resultados de clasificación en Curiam.

mapa mental de las relaciones entre los diferentes resultados para llegar a una conclusión, o lo que es lo mismo, para tomar una decisión. A su vez, permite la navegación entre los diferentes resultados avanzados de cada clasificador, mostrando en el panel de resultado aquel seleccionado.

- (b) Panel de resultado. Puede mostrar el resultado completo de un clasificador o un listado resumen de todos los resultados. Esto lo hace según el tipo de resultado seleccionado en los botones 'Multi' o 'Simple'. Posteriormente se describen cada uno de ellos.
- (c) Botones de selección de tipo de panel de resultado. Permiten alternar entre la vista de resultado completa o la vista resumen de múltiples resultados.

La vista de resultado completa del clasificador muestra toda la información que éste proporciona al usuario para la ayuda al diagnóstico 4.9. Está dividido en cuatro áreas:

- (a) Información del clasificador. Muestra información relativa al propio clasificador, como la pregunta a resolver, el algoritmo de aprendizaje utilizado, el centro desarrollador o el nombre que lo identifica. Además muestra la clase con mayor probabilidad.
- (b) Tarta de probabilidades. Ilustra en una gráfica de tarta las probabilidades a posteriori de que el caso clasificado pertenezca a cada una de las clases discriminadas por el clasificador.
- (c) Visualización específica de datos. Según el dominio biomédico en que se esté trabajando, el clasificador puede proporcionar datos para mostrar una visualización específica del dominio para reforzar la decisión tomada. Por ejemplo, un clasificador puede proporcionar las

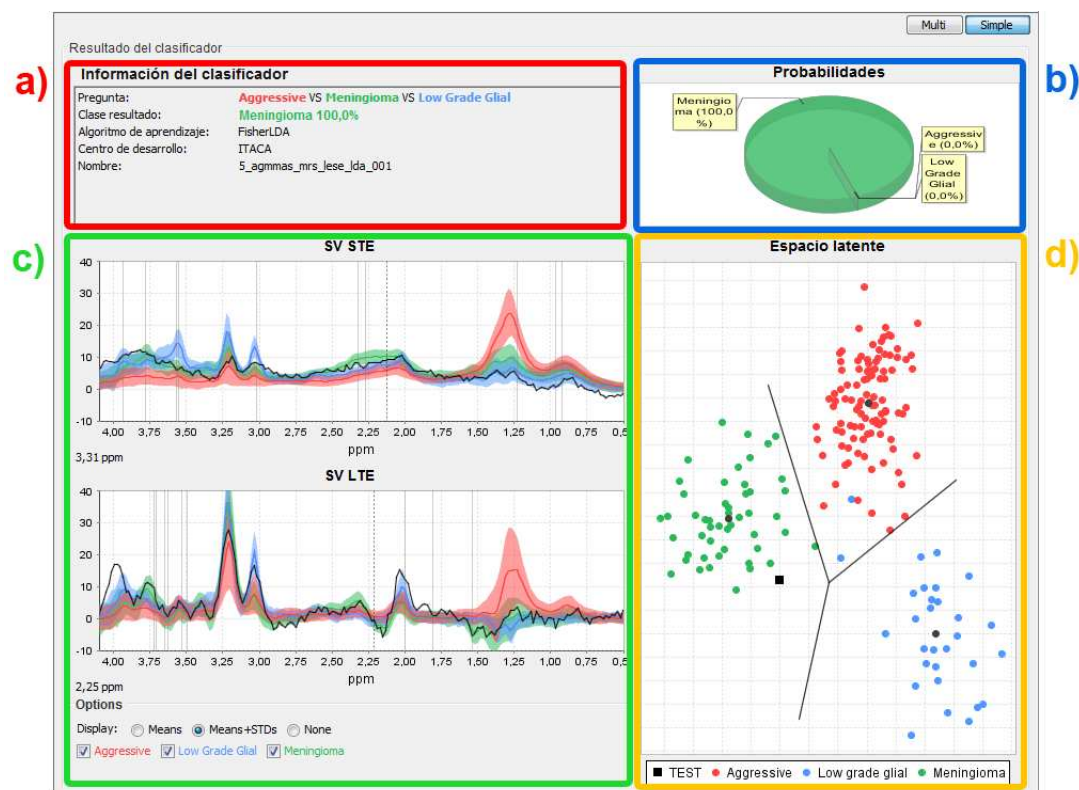


Figura 4.9: Vista de resultados completa de clasificación.

medias y desviaciones típicas de sus datos de entrenamiento para cada una de las clases, y compararlas con el dato clasificado.

- (d) Visualización específica del clasificador. En ocasiones, algunos clasificadores pueden mostrar una visualización de resultados específica a la técnica o algoritmo de clasificación empleado. Este panel muestra tal información. Por ejemplo, los clasificadores basados en la técnica LDA pueden mostrar un espacio latente con una representación visual de los casos usados en el entrenamiento, separados por clase o diagnóstico, dónde se incluye el caso clasificado para su comparación.

La vista de resultados múltiple de clasificación permite una rápida revisión de los diagnósticos ofrecidos por todos los clasificadores. Esta vista ofrece en forma de listado todos los resultados obtenidos por los clasificadores mostrando la información del clasificador y los gráficos de tarta correspondientes a las probabilidades de cada clase.

Como se indicó al comienzo del capítulo, entre otros atributos, un CDSS debe ofrecer la capacidad de generación de informes que ilustre los resultados o decisiones obtenidos por el sistema. En las versiones beta de Curiam esta capacidad no fue incluida, pero dicho requerimiento surgió dentro del proyecto de prueba de concepto de Curiam BT en el Hospital Universitario Doctor Peset y los Hospitales del área metropolitana de Barcelona. Por ello, se incluyó un componente para la generación de informes. La generación de informes de resultados de clasificación en Curiam se puede ejecutar una vez recibida una clasificación y mostrada en el panel de acción. A la hora de generar el informe, visualizando la vista de resultado completa, el usuario puede seleccionar uno de los clasificadores del listado. El informe contiene tanto el listado de todos los resultados como la visualización de resultado completa del clasificador seleccionado, siendo éste indicado en el listado previo. La figura 4.11 muestra un ejemplo de informe generado para un caso de tumores cerebrales en Curiam BT.

Como se ha podido observar en la interfaz de clasificación, los nombres de las posibles clases o diagnósticos llevan siempre un color asociado. Curiam muestra siempre de forma homogénea los nombres

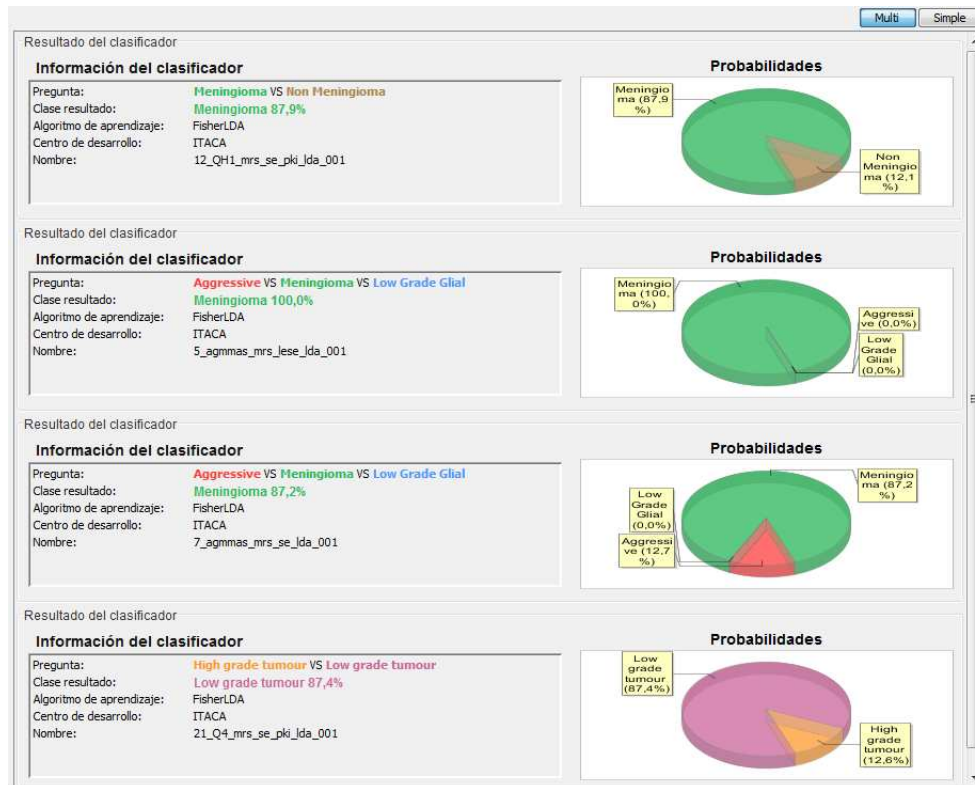


Figura 4.10: Vista de resultados completa de clasificación.

de dichas clases con su color asociado. De este modo el usuario puede localizar de forma rápida aquellas clases de su interés al mismo tiempo que se facilita la interpretación de la información mostrada por el sistema. Los colores asignados pueden modificarse por los usuarios en los ficheros de configuración de Curiam.

## 4.6. Mapeo de resultados de clasificación a ontología: prueba de concepto

Especialmente en el ámbito clínico, los DSS han de ser herramientas que proporcionen resultados de forma rápida y eficaz. Los resultados han de ser al mismo tiempo fácilmente interpretables. Esto es debido a que uno de los objetivos principales en el diseño de un CDSS es minimizar el tiempo invertido por el facultativo en la consulta del sistema. Por esta razón, son importantes los esfuerzos en obtener métodos de visualización de resultados de ayuda a la decisión cada vez mas eficaces.

Curiam permite al usuario la ejecución de múltiples clasificadores simultáneos para obtener respuesta a diferentes preguntas a resolver. En Curiam, un dominio biomédico está definido por su ontología de dominio. En ella se definen todas las posibles clases o diagnósticos a las que un caso puede estar asociado. Además se incluyen agrupaciones o propiedades de dichas clases. Los clasificadores, discriminan entre diferentes clases, agrupaciones o propiedades. Consecuentemente, las posibles preguntas a resolver por el CDSS son descritas por estos elementos. La ejecución de un conjunto reducido de clasificadores, que discriminen a la vez un número reducido de clases, puede proporcionar un pequeño listado de resultados fácilmente interpretable. Sin embargo, cuando el número de clasificadores aumenta, y/o se tiene una mayor diversidad de preguntas a resolver, la representación de los resultados pueden volverse demasiado heterogénea. Debido a este problema, se buscó una forma alternativa de proporcionar resultados simultáneos de un conjunto de modelos o clasificadores.

Como caso base, los resultados de un clasificador, proporcionan un conjunto de probabilidades a

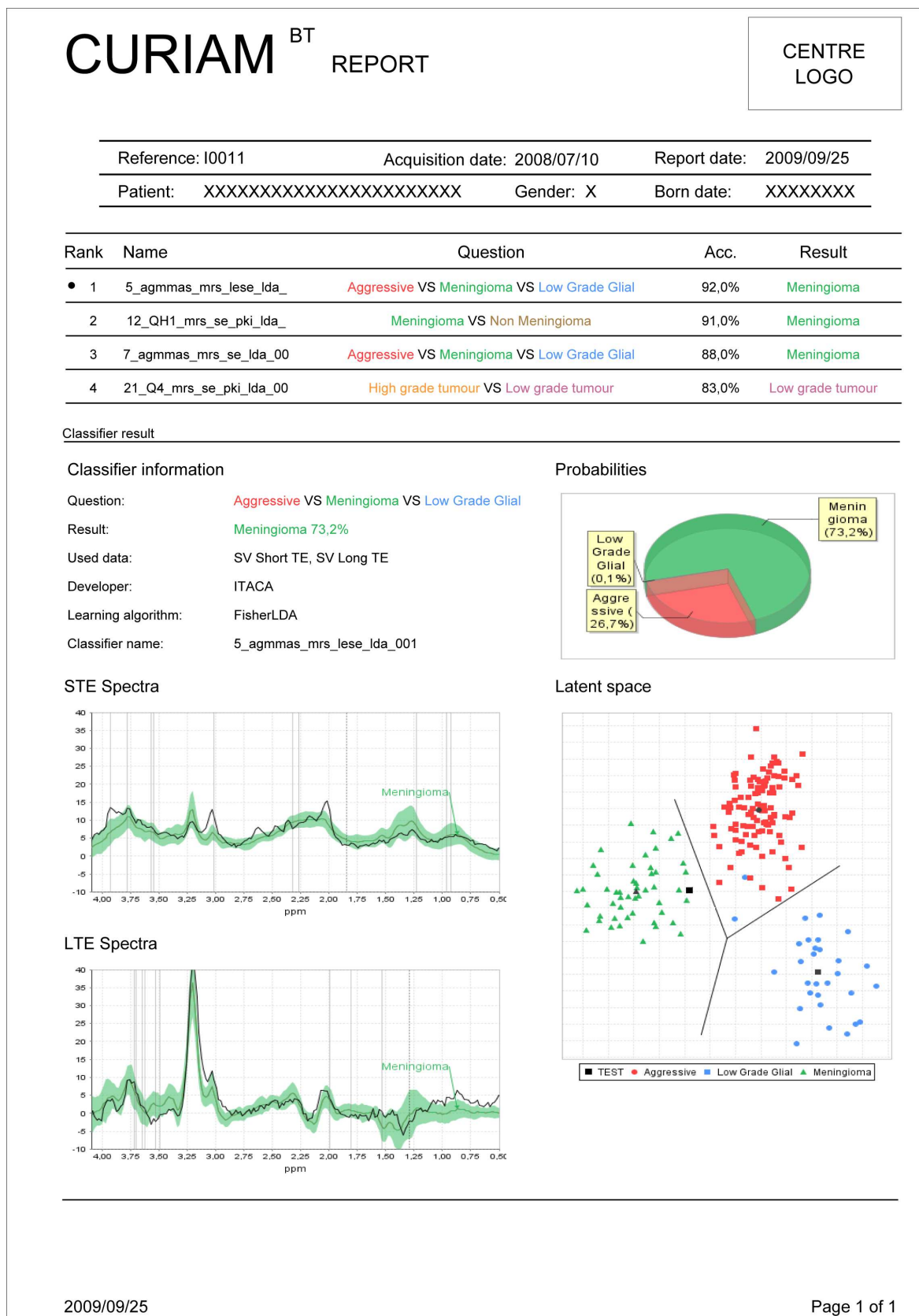


Figura 4.11: Informe de resultados de clasificación generado para un caso de tumores cerebrales en Curiam BT.

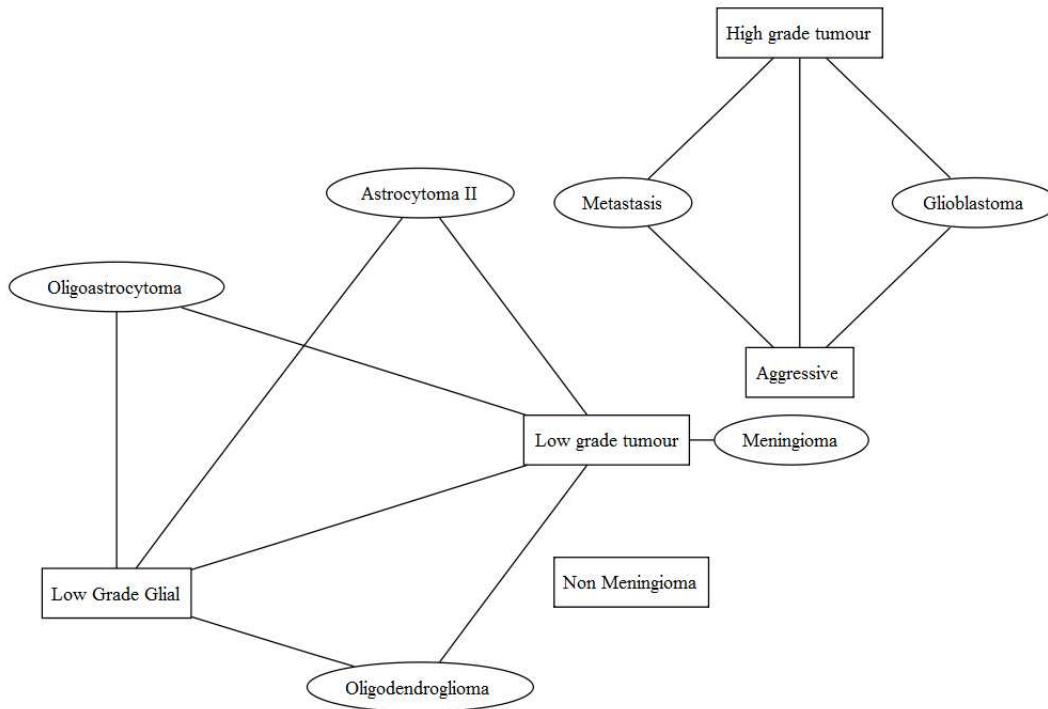


Figura 4.12: Ontología de dominio de tumores cerebrales en Curiam BT. Las elipses muestran clases de tumor y los rectángulos agrupaciones (grados de tumor) o propiedades (carácter agresivo o tumores no meningioma). Las flechas muestran las relaciones semánticas entre clases (es-un, no mostrado en este ejemplo) o entre clases y propiedades (tiene-caracter o pertenece-a-grupo).

posteriori para cada uno de los elementos discriminados por éste. Estas probabilidades dan un índice de la pertenencia del caso clasificado a cada uno de estos elementos. Generalizando, los resultados de un conjunto de clasificadores, proporcionan un conjunto de probabilidades a posteriori para un conjunto de los elementos definidos en la ontología del dominio. Esto nos lleva a buscar un método eficaz para trasladar, o 'mapear', este conjunto de probabilidades a una representación de la ontología del dominio, de forma que se obtenga una única visualización global de los resultados obtenidos por todos los clasificadores. El hecho de proporcionar una única representación de todos los resultados, y hacerlo sobre una ontología o mapa conceptual conocido por el facultativo, le permitiría interpretar de forma rápida la ayuda a la decisión propuesta por el CDSS.

Pese a estar en fase de concepción, de cara a trabajo futuro, durante el desarrollo de Curiam se diseñó una prueba de concepto del método. La descripción semántica en Curiam dada por la capa de conocimiento vertical, une los conceptos del sistema de clasificación con los de la ontología de dominio. De este modo se diseñó un método simple para mapear el conjunto de probabilidades obtenidas de los clasificadores a la ontología de dominio y proporcionar una visualización que pueda ser interpretada de forma simple. La figura 4.12 muestra la ontología de dominio de los posibles diagnósticos en Curiam BT.

El objetivo para obtener la visualización de los resultados fue asignar un mapa de calor a la visualización de ontología de dominio que represente el índice global obtenido por cada elemento. Según dicho índice, los elementos mostrarán desde un color blanco — mínimo índice — a un color rojo — máximo índice —. El método empleado para obtener el índice global fue un método *naive* que suma a cada elemento la probabilidad a posteriori obtenida por aquellos clasificadores que lo discriminan, y finalmente la divide por la suma de probabilidades a posteriori total — de todas las clases entre todos los modelos, y dado que la probabilidad total de un modelo es 1, equivale al número de modelos —.

El índice utilizado tiene las siguientes carencias:

1. Existe un *bias* generado por el número de modelos discriminando una misma clase. En el caso de que todos los modelos del conjunto resuelvan la misma pregunta, el método equivale al cálculo de

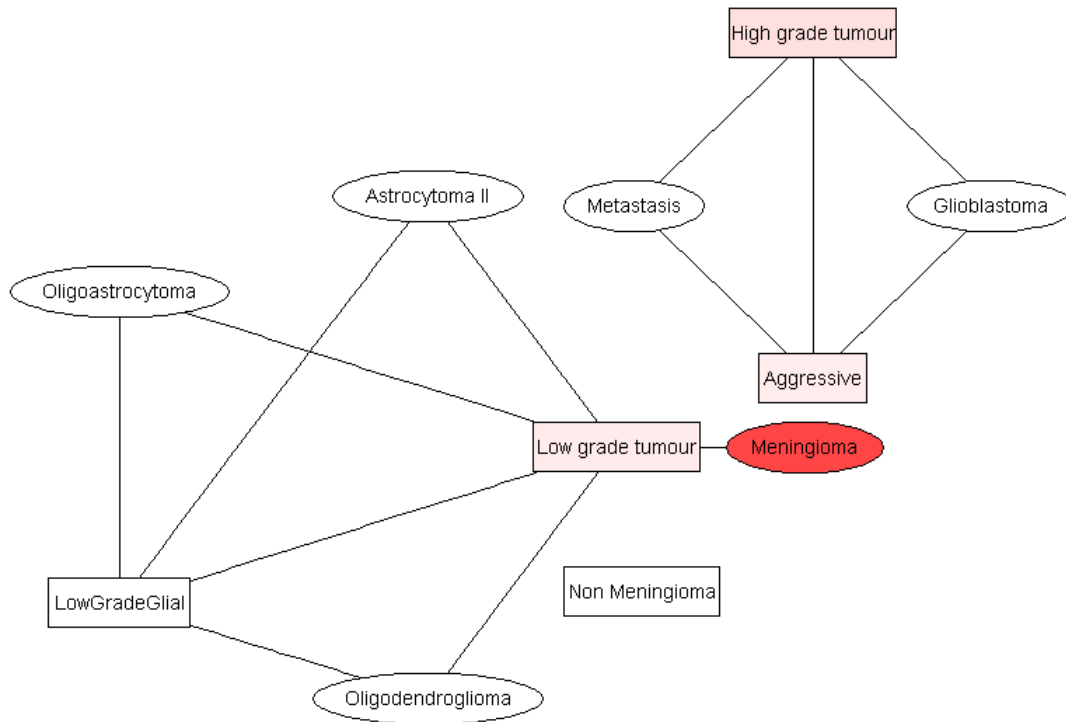


Figura 4.13: Visualización de resultados global de un caso de tumores cerebrales. La clase *Meningioma* ha obtenido el mayor índice global, por lo que muestra el mayor nivel de rojo de entre todos los elementos.

la media de las probabilidades a posteriori de cada clase. Sin embargo, con modelos heterogéneos, una clase discriminada por un elevado número de modelos, obtendrá a priori un mayor índice global que otra que sea discriminada por un número mínimo de modelos.

2. Una clase o agrupación representada en la visualización y que sea correcta para un caso, no obtiene ningún índice global si no se encuentra entre las clases discriminadas por los clasificadores.
3. No existe conexión semántica entre los índices globales asignados a los elementos, por ejemplo, que una superclase obtenga un elevado índice global no afectará a la probabilidad de una subclase con una relación es-un.

Éste último es un punto importante, ya que los índices globales podrían unirse a la inferencia o razonamiento automático — mediante motores de inferencia o razonamiento — para llegar a concluir proposiciones desde premisas definidas por los resultados. Las figuras 4.13 y 4.14 muestran las visualizaciones de esta prueba de concepto obtenidas tras diferentes clasificaciones.

Por el hecho de complementar la prueba de concepto, se incluyó una función que ocasiona que los elementos puedan 'activarse', sin influir en su índice global, cuando los índices globales de sus vecinos superan determinados umbrales.

La activación se ilustra cambiando el color de la forma que representa la clase, de negro — color por defecto — a rojo. Esta función permite que elementos que no estén entre los discriminados por los modelos, pero cuyo valor es verdadero para un determinado caso, sean activados al superar sus vecinos los umbrales establecidos. Esto puede apreciarse en las figuras 4.15 y 4.16. La agrupación de resultados de clasificación de clasificadores que discriminan categorías semánticamente relacionadas será uno de los temas de estudio en las fases posteriores a esta tesina.

En esta sección se ha descrito la prueba de concepto realizada para visualizar resultados de un conjunto de clasificadores mediante el mapeo de las probabilidades a posteriori obtenidas por los modelos a la ontología del dominio. Pese a tratarse de una prueba simple, fue incluido en las versiones beta de Curiam BT y Curiam STT. Fue ocultado en la primera versión oficial de Curiam BT — la única hasta el momento



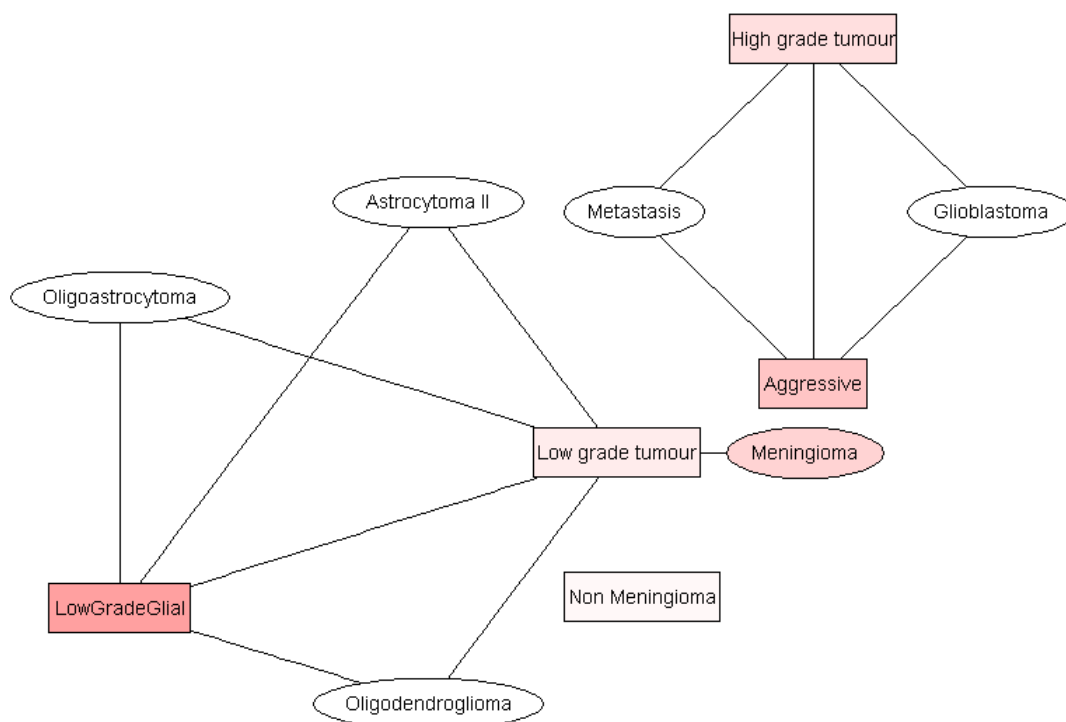


Figura 4.14: Visualización de resultados global de un caso de tumores cerebrales. En este caso el índice global se encuentra más distribuida entre todos los elementos, siendo la clase *Low Grade Glial* la ganadora.

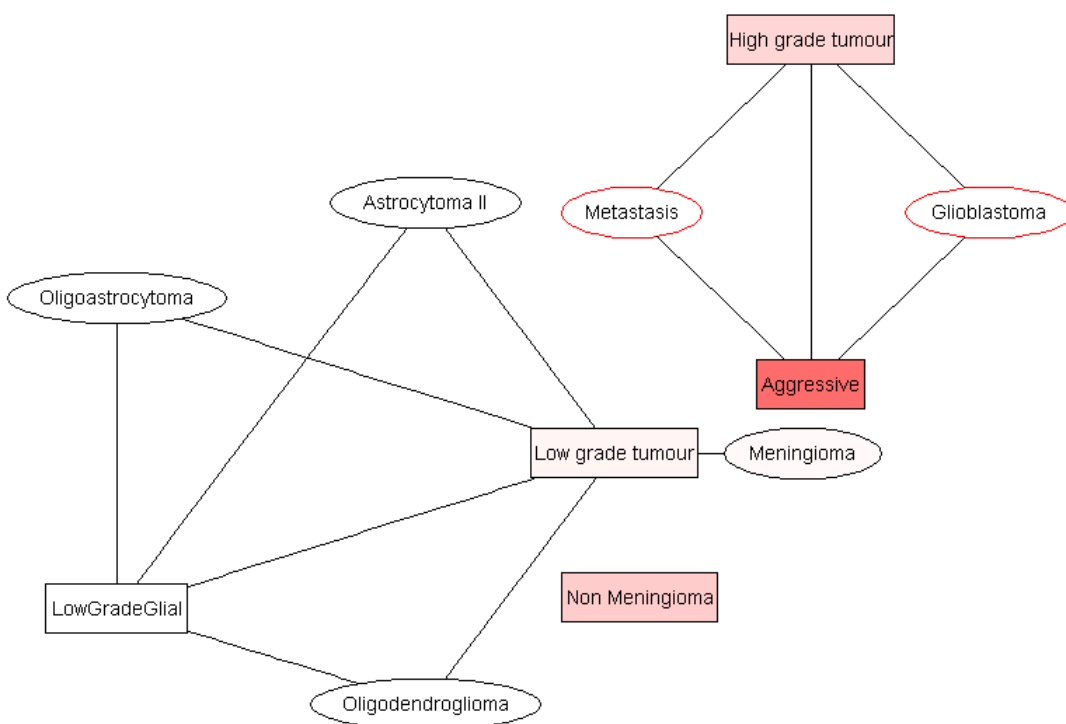


Figura 4.15: Visualización de resultados global de un caso de tumores cerebrales. Este resultado ilustra la activación de las clases *Metastasis* y *Glioblastoma* debido a sus elementos vecinos, la propiedad *Aggressive* y la agrupación *High grade tumour* han superado los umbrales especificados. En el gráfico de barras se puede apreciar que dichas clases activadas no estuvieron incluidas entre las discriminadas por los clasificadores.

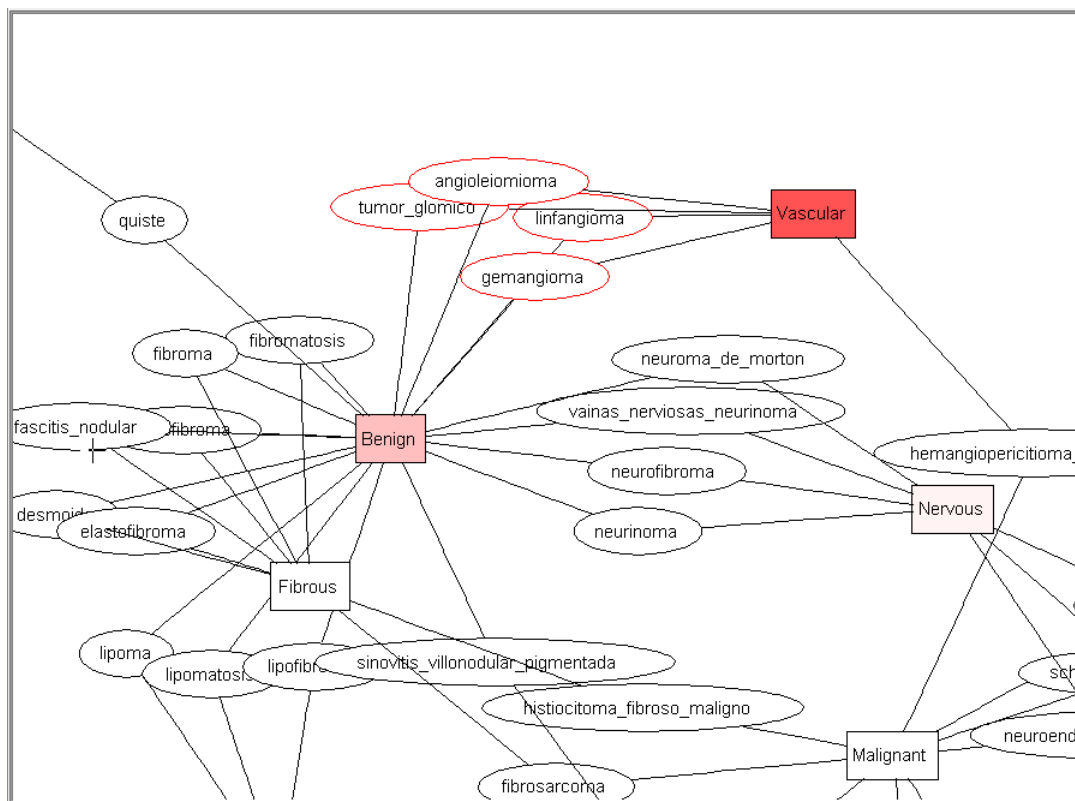


Figura 4.16: Subconjunto de la visualización de resultados global de un caso de tumores de partes blandas. El índice global de los tipos de tumor *Vascular* y *Benigno* provoca la activación de las clases relacionadas entre ambos.

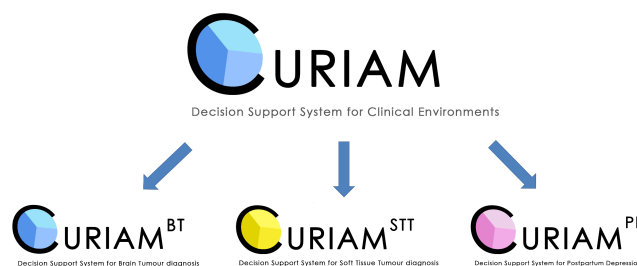


Figura 4.17: Logotipos de las tres versiones de Curiam: Curiam BT, Curiam STT y Curiam PPD.

de las descritas en la sección siguiente —, a la espera de un estudio más conciso. La investigación en este método de representación deja abierto un trabajo futuro de relevancia para la incorporación de sistemas de ayuda a la decisión a la práctica clínica.

## 4.7. Implementaciones de Curiam para DSS clínicos

Actualmente se han desarrollado tres implementaciones de Curiam para DSS clínicos (Figura 4.17). Cada una de las implementaciones está basada en un dominio biomédico diferente, con el fin de demostrar la capacidad de adaptación de Curiam a dominios médicos diferentes. Dos de las implementaciones están actualmente desarrolladas a modo de prueba de concepto, en forma de versión beta, para trasladar a un sistema real los resultados de estudios de minería de datos realizados en dos ámbitos: diagnóstico de tumores de partes blandas a partir de hallazgos de imagen [17, 53], y predicción de la depresión postparto [48]. La otra implementación se encuentra en un estado más avanzado, concretamente en su primera versión oficial, debido a que recientemente se han dedicado más recursos al estudio del dominio en los proyectos eTumour y HEALTHAGENTS. Esta implementación consiste, por tanto, en la versión de Curiam para el diagnóstico de tumores cerebrales.

Las siguientes tres secciones describen cada una de las tres implementaciones existentes de Curiam.

### 4.7.1. Curiam BT: DSS para el diagnóstico de tumores cerebrales

La aplicación de técnicas de PR para el diagnóstico automático de tumores cerebrales se ha venido estudiando en el grupo IBIME desde el inicio de los proyectos eTumour y HEALTHAGENTS, en los años 2004 y 2006 respectivamente. En sendos proyectos se construyeron prototipos de CDSS para el diagnóstico de tumores cerebrales desde la perspectiva de sistema de información de datos de MRS en el primero, y de DSS basado en Web en el segundo. Consecuentemente, la aplicabilidad del conocimiento adquirido a su correspondiente implementación del sistema Curiam fue directa, dando lugar a Curiam BT. La primera prueba de concepto de Curiam BT fue presentada a la comunidad científica en [38] en el año 2008. La primera versión oficial de Curiam BT fue presentada en [39] en 2009.

La ayuda a la decisión en Curiam BT está basada en la clasificación de datos de MRS Single Voxel (SV), adquiridos en tiempo de eco (TE) corto o TE largo. Los datos de MRS SV consisten en una señal espectral que proporciona información de concentraciones de metabolitos en un único vóxel localizado en una zona tumoral del cerebro. En la adaptación a Curiam, estos datos se trataron mediante el tipo de característica *vector de datos*. Para una óptima visualización de dichos datos, se desarrolló un componente GUI de datos específico. Este componente representa el espectro de MRS, sobre el cual el usuario puede realizar zoom horizontal para una visualización más o menos precisa de un determinado rango de frecuencia (en partes por millón de la frecuencia de referencia del espectrómetro, PPM). Además, como la información metabólica en los espectros de RM está asociada a la frecuencia se facilita el valor PPM sobre el cual pasa el cursor para una mejor localización de los metabolitos.

Sin embargo, el propósito de este componente no es únicamente la visualización de datos. Al introducir los datos de un nuevo caso en Curiam, es sencillo si se trata de variables numéricas o categóricas, pero en el caso de datos MRS, éstos se encuentran en varios formatos, desde señales en texto plano, hasta datos

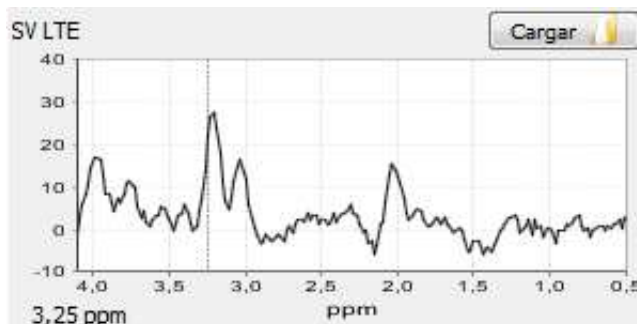


Figura 4.18: Componente GUI de Curiam BT para la visualización de datos de MRS.

en crudo generados por las máquinas de RM. Además, según la potencia del campo magnético generado por la máquina de RM — actualmente las máquinas comerciales suelen trabajar en 1.5 o 3.0 Tesla —, los datos se encuentran en diferente resolución espectral. Con el objetivo de permitir a Curiam BT trabajar con toda resolución espectral, se desarrolló un módulo de procesamiento de espectros de MRS, el cual, unido al componente GUI del dato permite la carga de estos tipos de datos en Curiam BT.

Para realizar el procesamiento de datos de MRS, el módulo desarrollado accede a dos herramientas externas de referencia en el campo del procesamiento de MRS: jMRUI <sup>1</sup> y jDMS [46], y entre ellas se realiza una corrección de fase desarrollada de forma específica. El módulo de procesamiento de Curiam BT envía al software jMRUI los datos crudos junto a una secuencia de comandos para un procesamiento en modo *batch*. Éste, genera un fichero en su propio formato, el cual es cargado por el módulo de procesamiento para realizar la corrección de fase. Para realizar este paso, se diseñó un método manual, que permitiese a los usuarios corregir la fase del espectro — correcciones de orden 0 y orden 1 — con una mayor exactitud (Figura 4.19). Se optó por el método manual por ser el método más adecuado en la actualidad. Finalmente, se llevan a cabo unas últimas correcciones al espectro fasado mediante la librería jDMS (normalización del espectro por Unit Length Normalization, principalmente). Tras finalizar este proceso, el espectro es visualizado e incluido en el caso.

En Curiam BT se ha desarrollado otro componente GUI para facilitar la interpretación de los resultados obtenidos en una clasificación (Figura 4.20). Se trata de una visualización específica de datos en los resultados completos de un clasificador. Esta visualización permite al usuario comparar espectros medios y desviaciones típicas de los datos usados en el entrenamiento del clasificador con el caso clasificado.

Un último componente adicional fue incluido para completar la adaptación de la GUI de Curiam al dominio de tumores cerebrales. En este caso se trata de un complemento GUI que permite alternar la combinación de colores empleada para visualizar los espectros de RM. Este cambio consiste en ofrecer una visualización de alto contraste con el aspecto de ciertos sistemas de radiología, en donde los espectros son mostrados sobre un fondo negro — estilo de las imágenes médicas —. La figura 4.21 muestra el efecto de esta visualización en la GUI de Curiam BT.

Curiam BT incluye actualmente 16 clasificadores de tumores cerebrales para resolver siete preguntas clínicas: agresivo vs. meningioma vs. glial de grado bajo; tumor de grado alto vs. tumor de grado bajo; meningioma vs. no meningioma; glial grado I vs. glial grado II; grados I-II vs. grados III-IV; glial de grado bajo vs. glial de grado bajo vs. embrional; y glial no ependimoma grados II-III-IV vs. glial grado I vs. embrional. En Curiam BT, y siguiendo los protocolos de adquisición de espectros SV, un caso puede disponer de dos tipos de datos: espectro MRS SV en TE corto o espectro MRS SV en TE largo — además de poder incluir una imagen de referencia del caso, no usada para clasificación —. Los clasificadores se distribuyen entre los que requieren uno de los dos TE para la clasificación, o ambos. Existen varios clasificadores capaces de responder la misma pregunta clínica pero usando uno de los espectros o los dos [13]. Esto permite que según la disponibilidad de datos, siempre se pueda utilizar al menos un clasificador disponible. Algunos clasificadores han sido tomados de los desarrollados para el dDSS de HEALTHAGENTS utilizando los datos del proyecto INTERPRET [51]. El resto fueron desarrollados a partir de nuevos estudios para el diagnóstico de tumores infantiles, o tras el desarrollo de nuevos experimentos de PR. Los clasificadores de

<sup>1</sup>Magnetic Resonance User Interface, <http://www.mrui.uab.es/mrui/>

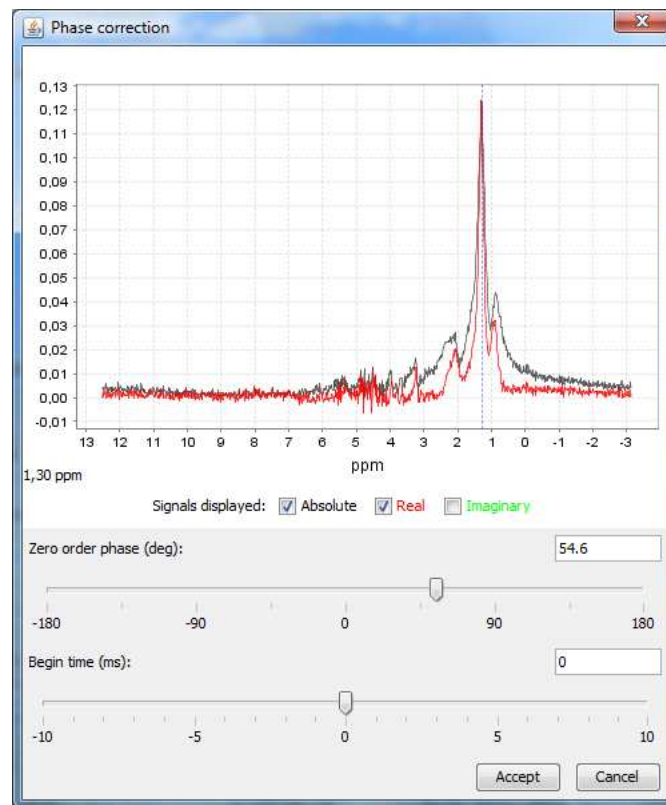


Figura 4.19: Corrección de fase manual en espectros de RM en Curiam BT.

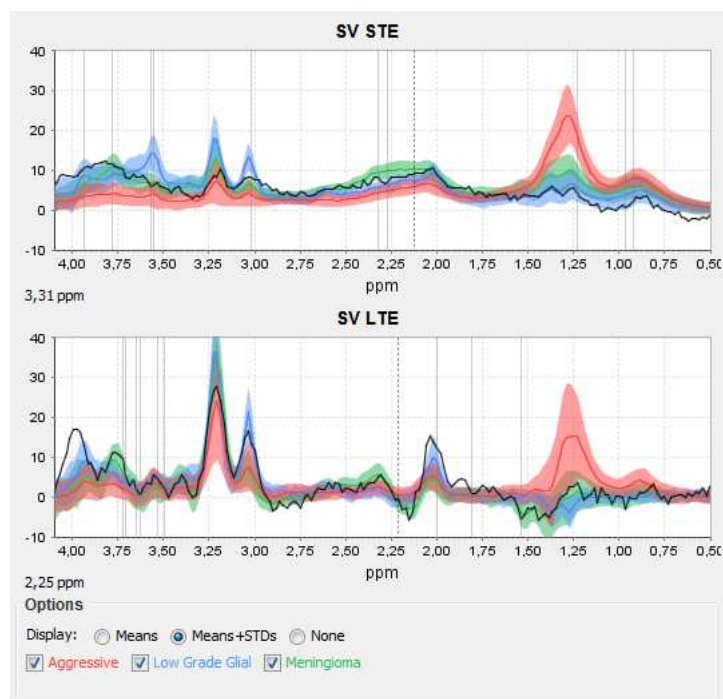


Figura 4.20: Visualización de datos MRS en resultados de clasificación en Curiam BT. Esta visualización permite comparar espectros medios y desviaciones típicas de los datos usados en el entrenamiento del clasificador con el caso clasificado.

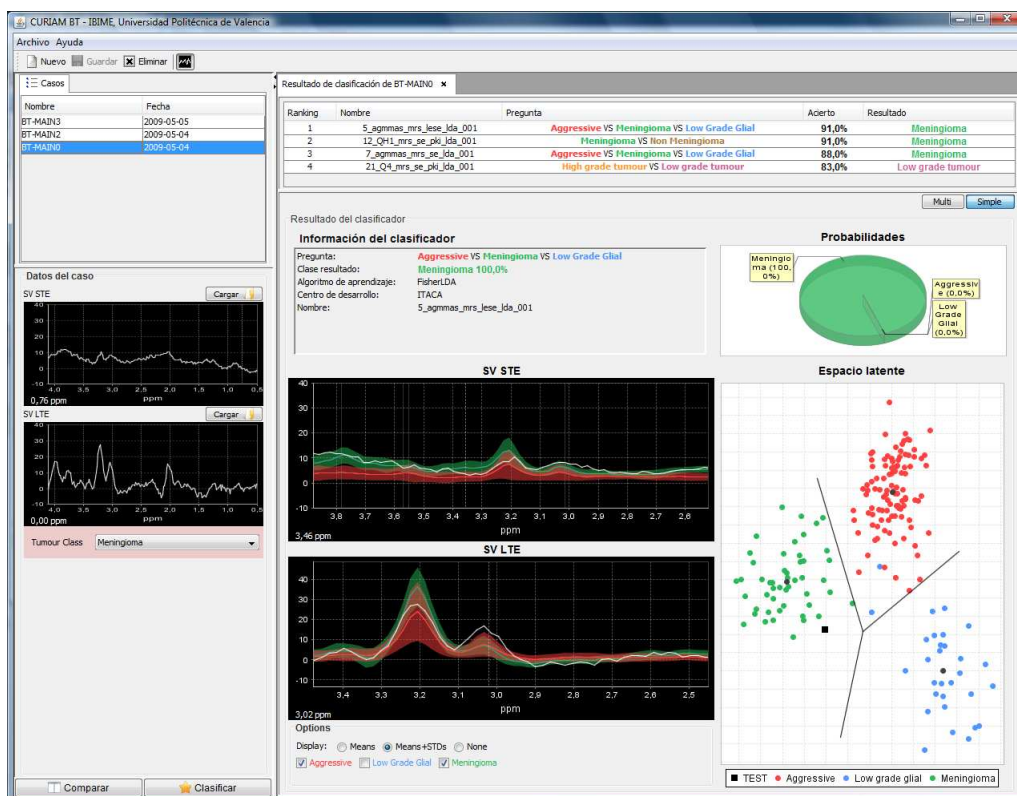


Figura 4.21: Aspecto de Curiam BT activada la visualización de espectros de RM con el aspecto de sistemas de radiología. Esta visualización se activa y desactiva desde su componente asociado en la barra de herramientas.

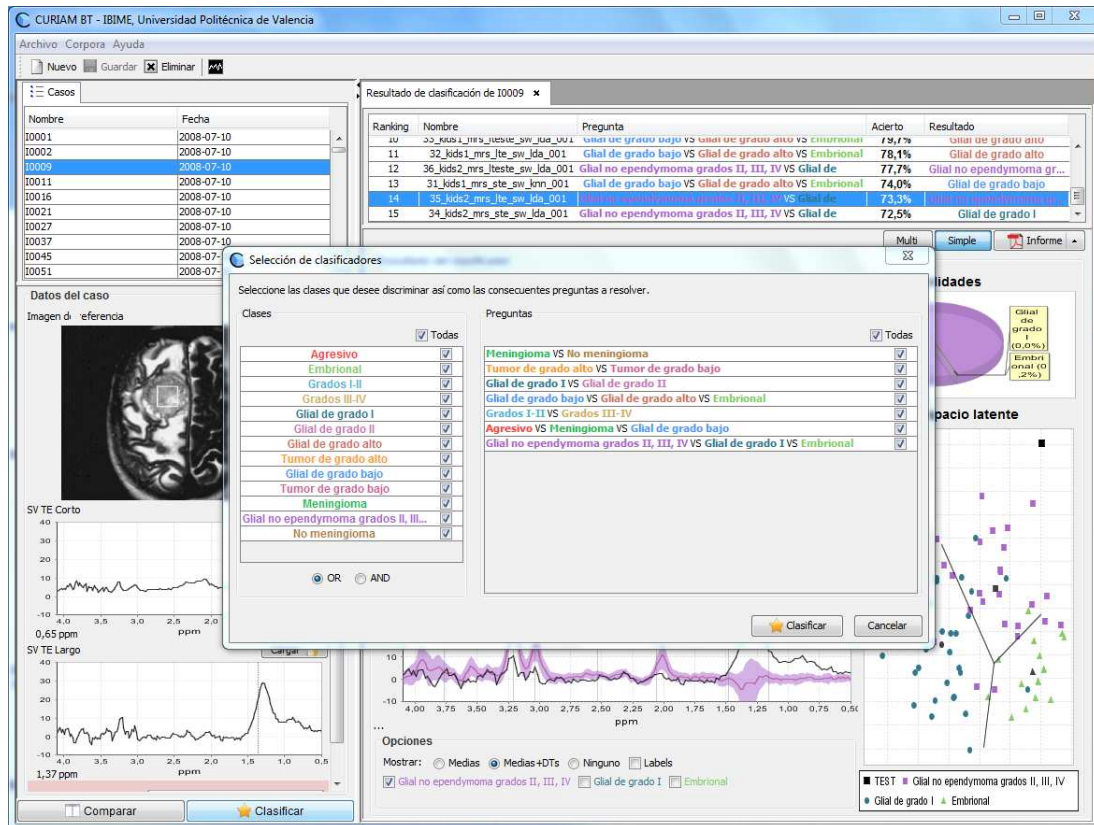


Figura 4.22: Captura de pantalla de Curiam BT en donde se puede apreciar el cuadro de diálogo de selección de clasificadores. Mediante este cuadro de diálogo el usuario puede seleccionar las clases y preguntas a discriminar según su conveniencia.

Curiam BT están por tanto basados en diferentes métodos de PR. Como técnicas de clasificación emplean Fisher-LDA y KNN; y como extracción de características emplean *peak integration* y selección de puntos. La figura 4.22 muestra una captura de pantalla de Curiam BT en donde se puede apreciar el cuadro de diálogo de selección de clasificadores. Mediante este cuadro de diálogo el usuario puede seleccionar las clases y preguntas a discriminar según su conveniencia.

En la prueba de concepto de Curiam BT se incluyó la visualización de resultados global basada en la ontología del dominio. Se empleó para ello un subconjunto de la taxonomía WHO, ampliado con las agrupaciones y propiedades de clases de tumores cerebrales que en ese momento eran discriminadas por los clasificadores. Debido a que dicha visualización de resultados se trataba del mismo modo de una prueba de concepto, ésta fue deshabilitada para la primera versión oficial de Curiam BT. Obviamente, para completar la adaptación de Curiam al dominio, se incluyeron las traducciones de los terminos a los idiomas inglés y español, y se asignaron diferentes colores para representar en el sistema las diferentes clases de tumores cerebrales.

Actualmente, la versión Curiam BT 1.0 se encuentra en evaluación clínica bajo el proyecto 'CURIAM Sistema de ayuda a la decisión médica. Plan de explotación y prueba de concepto en el Hospital Universitario Doctor Peset y los Hospitales del área metropolitana de Barcelona', financiado por la Universidad Politécnica de Valencia bajo el programa INNOVA 2008.

Las figuras y muestran dos capturas de pantalla de Curiam BT.

#### 4.7.2. Curiam STT: DSS para el diagnóstico de tumores de partes blandas

Los tumores de partes blandas son una enfermedad poco frecuente en la población general. Por esta razón, la mayoría de los médicos y radiólogos no están familiarizados con su apariencia y sus diferentes

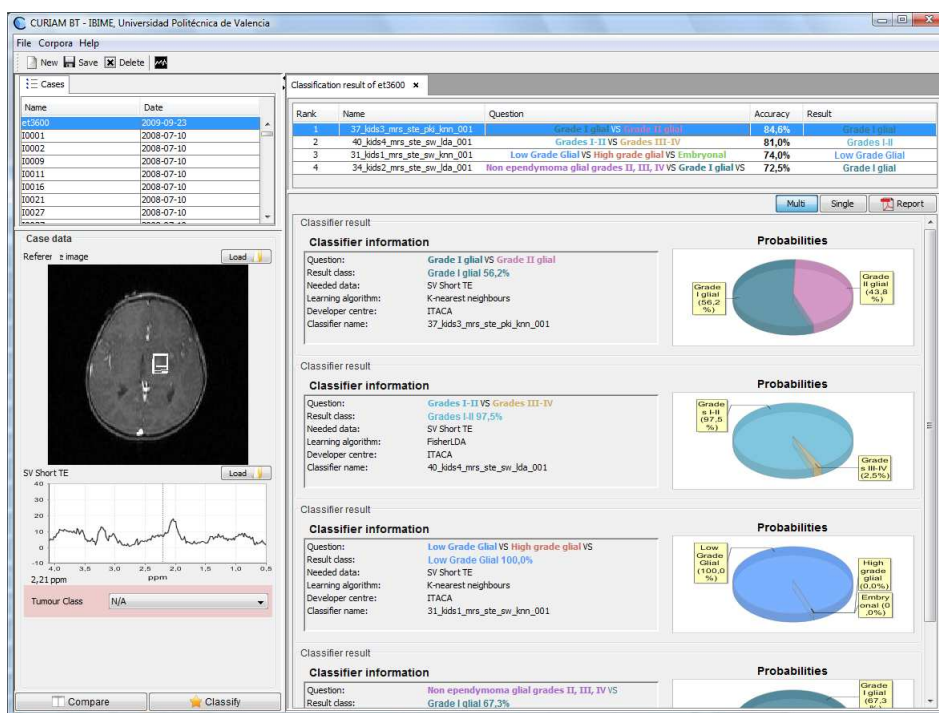


Figura 4.23: Captura de pantalla de Curiam BT. Muestra la clasificación de un caso de tumor cerebral infantil mediante la vista de resultados múltiple.

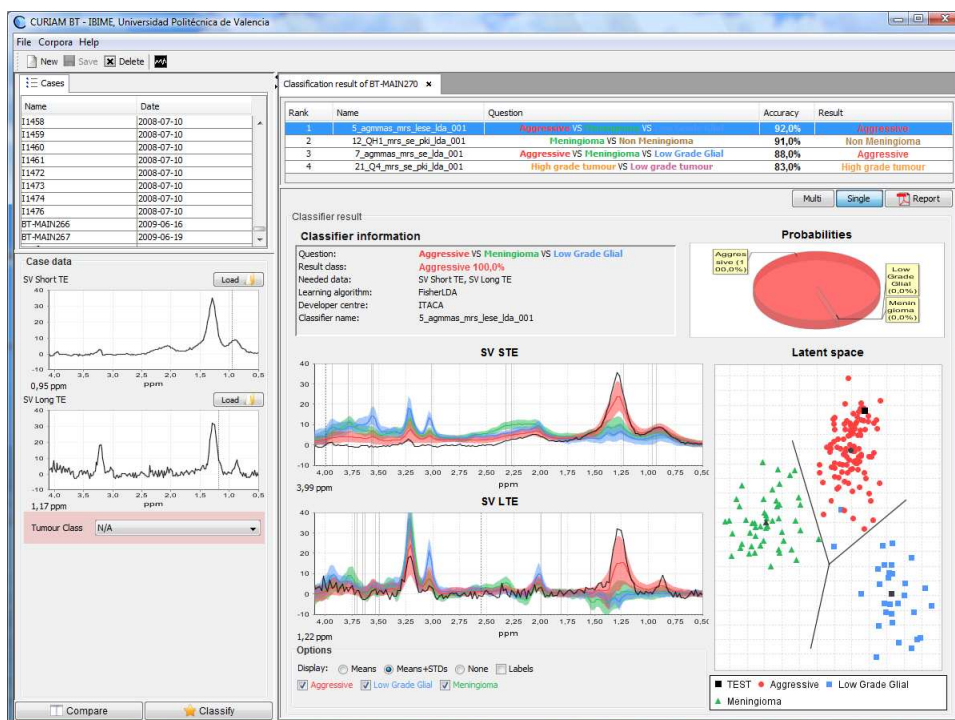


Figura 4.24: Captura de pantalla de Curiam BT. Clasificación de un tumor agresivo.



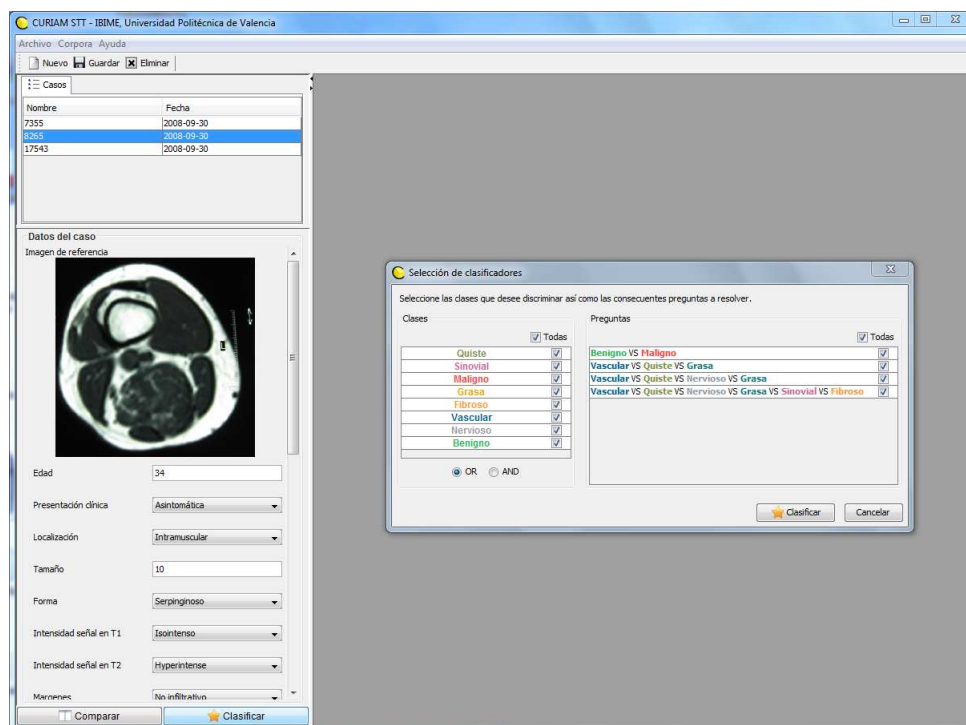


Figura 4.25: Captura de pantalla de Curiam STT en donde se puede apreciar el cuadro de diálogo de selección de clasificadores. Se muestran las cuatro preguntas clínicas, una de caracteres y tres de estirpes, asociadas cada una de ellas a un clasificador.

diagnósticos. En este sentido, en [17] y [53] se desarrollaron métodos para la clasificación de caracteres y estirpes de casos de tumores de partes blandas mediante hallazgos de imagen. Este dominio biomédico fue el tratado en el primer prototipo de CDSS del grupo IBIME, SOC [18]. Actualmente, también se ha desarrollado una adaptación de Curiam para tal dominio: Curiam STT.

Curiam STT fue adaptado en primer lugar mediante la configuración de descripción lógica de los datos. Los hallazgos de imagen son representados mediante variables numéricas y categóricas, por lo que no se requirió el desarrollo de componentes GUI de datos adicionales. En segundo lugar se definieron los idiomas para representar los conceptos del dominio y los colores asignados a las diferentes clases. Finalmente, se incluyeron en el CF los clasificadores desarrollados en los diferentes estudios. Estos consisten en un clasificador entre caracteres benigno vs. maligno, y tres clasificadores de estirpes: vascular vs. quiste vs. grasa; vascular vs. quiste vs. nervioso vs. grasa; vascular vs. quiste vs. nervioso vs. grasa vs. sinovial vs. fibroso. Todos los clasificadores están basados en la técnica de KNN y requieren de la normalización lineal para algunas características numéricas y codificación de características categóricas.

La vista de resultado global basada en ontología fue incluida al tratarse de una versión beta. Ésta, además de mostrar los diferentes grupos de caracteres y estirpes, muestra los diagnósticos posibles al relacionar un par carácter-estirpe (Figura 4.16).

Las figuras 4.25 y 4.26 muestran dos capturas de pantalla de Curiam STT. En ellas se puede apreciar el cuadro de diálogo de selección de clasificadores con las posibles clases y preguntas; y la vista de resultado de clasificación múltiple mostrando los resultados de la ejecución de dichas preguntas.

### 4.7.3. Curiam PPD: DSS para la predicción de depresión postparto

La depresión postparto afecta aproximadamente a entre el 10 % y el 15 % de las mujeres que dan a luz. Las mujeres con depresión postparto sufren un deterioro cognitivo y emocional que puede afectar al desarrollo del hijo. La identificación precoz de mujeres en riesgo de desarrollar depresión postparto

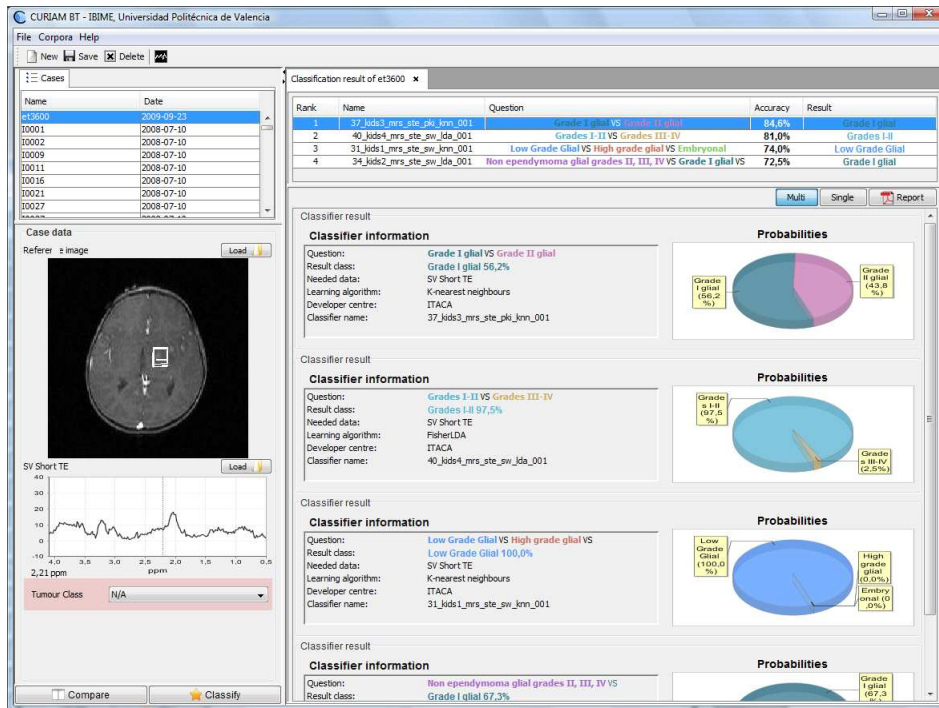


Figura 4.26: Captura de pantalla de Curiam STT. Muestra los resultados de los cuatro clasificadores de tumores de partes blandas mediante la vista de resultados múltiple.

tras haber dado a luz permitiría realizar intervenciones preventivas dirigidas a estas pacientes. En [48], el grupo IBIME desarrolló un método basado en redes neuronales artificiales para la detección precoz de esta patología. Esta predicción se lleva a cabo a partir de diferentes variables clínicas, genéticas y socioeconómicas de la paciente, junto a los resultados de sus respuestas en cuestionarios clínicos. Con el fin de trasladar este estudio a un sistema real, se desarrolló una adaptación de Curiam a la predicción de la detección postparto: Curiam PPD. La adaptación se trató en primer lugar de la configuración de descripción lógica de los datos, para representar las variables requeridas por el modelo, que en este caso se trataron de variables numéricas y categóricas. En segundo lugar se definieron los idiomas para representar los conceptos del dominio y los colores asignados a las diferentes clases. Paralelamente, la red neuronal desarrollada en el estudio, fue definida mediante la definición de clasificadores XML del CF, para su inclusión en el sistema. En este caso únicamente se tienen dos posibles clases o diagnósticos: riesgo de sufrir depresión postparto o sin riesgo de sufrir depresión postparto, un único clasificador fue suficiente para mostrar los resultados. Por lo tanto se deshabilitó la vista de resultados basada en ontología. Actualmente Curiam PPD se encuentra en estado de prueba de concepto. La figura 4.27 muestra una captura de pantalla de Curiam PPD.

The screenshot displays the CURIAM PPD software interface. On the left, a 'Datos del caso' panel lists various clinical variables such as 'Antecedentes psiquiátricos', 'Alteraciones emocionales durante embarazo', and 'Número de personas viviendo juntas'. The main workspace shows classification results for a specific case, including a table with columns for 'Ranking', 'Nombre', 'Pregunta', 'Acerto', and 'Resultado'. Below this, a 'Probabilidades' section features a pie chart showing the distribution of predicted classes: 'Riesgo de DPP (56,4%)' and 'Sin riesgo de DPP (43,6%)'. In the foreground, a 'Selección de clasificadores' dialog box is open, allowing the user to select the classes to be discriminated and the questions to be resolved, with options for 'OR' or 'AND' logic.

Figura 4.27: Captura de pantalla de Curiam PPD. En el panel de datos se puede apreciar un conjunto de las variables de un caso. El espacio de trabajo muestra los resultados de una clasificación. En la parte inferior derecha se observa el cuadro de diálogo de selección de clasificadores, dónde se muestran las dos posibles clases a discriminar: riesgo de depresión postparto o sin riesgo de depresión postparto.



# Capítulo 5

## Conclusiones y trabajo futuro

Esta tesina se ha centrado en la investigación y desarrollo de sistemas de ayuda a la decisión clínica basados en aprendizaje automático. El fin último de la tesina era obtener en conjunto una solución novedosa y reutilizable para el desarrollo de nuevos CDSS basados en PR para el diagnóstico clínico. Las principales conclusiones se citan a continuación:

1. Un nuevo framework para proporcionar servicios de clasificación ha sido desarrollado y testeado en entornos reales mediante el dDSS de HEALTHAGENTS y el CDSS Curiam.
2. El dDSS de HEALTHAGENTS permite que diferentes centros puedan crear sus propios clasificadores y sean incluidos en la plataforma, permitiendo que puedan ser accedidos desde diferentes centros clínicos para obtener ayuda a la decisión en el diagnóstico de tumores cerebrales.
3. La escalabilidad del CF ha sido demostrada mediante la incorporación exitosa de nuevos métodos de clasificación, así como de nuevos clasificadores, desarrollados por otros laboratorios y terceras personas.
4. El diseño del sistema de clasificación de HEALTHAGENTS, basado en el CF ha sido aceptado para publicación en la revista *The Knowledge Engineering Review* [40], lo que demuestra su novedad científico-tecnológica.
5. Curiam ha demostrado su adaptabilidad al haber sido la plataforma sobre la que desarrollar tres implementaciones para tres dominios biomédicos diferentes: tumores cerebrales, tumores de partes blandas y depresión postparto.
6. Curiam fue introducido en el congreso ESMRMB 2008 [38], mostrando sus versiones beta Curiam BT y Curiam STT. La madurez adquirida en la clasificación de tumores cerebrales, permitió obtener la primera versión oficial de Curiam BT, que fue presentada en el congreso ESMRMB 2009 [39], y se encuentra actualmente en evaluación clínica.

Los resultados obtenidos en esta tesina ofrecen la base para multitud de trabajos futuros en CDSS. Concretamente, algunos de los próximos objetivos a llevar a cabo, basados en el CF y Curiam, son los siguientes:

- Actualmente Curiam se encuentra en evaluación clínica multicéntrica. La interpretación de los resultados obtenidos dará a conocer el *feedback* de usuarios clínicos reales, lo que permitirá una sólida mejora del sistema. De este modo se contribuirá a mejorar el sistema y demostrar su valor añadido.
- Los sistemas de información hospitalaria suelen estar formados por bases de datos heterogéneas, pocas veces definidas mediante estándares de facto para sistemas de historia clínica (HL7 <sup>1</sup>, CEN-13606 <sup>2</sup>). Actualmente se están desarrollando nuevas herramientas que permiten la integración y estandarización de estos sistemas de información (como por ejemplo Pangea [1], LinKEHR [30] o

---

<sup>1</sup>www.hl7.org

<sup>2</sup>CEN/TC251. EN13606-Medical Informatics-Electronic health record communication

Ensembl<sup>3</sup>). Esto abre una gran puerta a la incorporación de CDSS en los sistemas hospitalarios. Curiam fue diseñado con el objetivo de poder acceder a nuevos sistemas de información mediante la comunicación basada en XML. De este modo, uno de los futuros trabajos consistirá en el desarrollo del módulo de acceso a datos que permita a Curiam el tratamiento de estándares de historia clínica para así poder ser integrado en la mayoría de nuevos sistemas hospitalarios.

- Los CDSS deben encontrar un nexo de unión en el proceso clínico. Los centros clínicos y hospitalarios suelen tener definidos procesos, o guías clínicas, que dirigen el estado del paciente así como las pruebas o intervenciones clínicas realizadas a través de un sistema de grafos. Los CDSS pueden contribuir a mejorar la efectividad y eficiencia de dichas guías clínicas. Por esta razón, se pretende evaluar la integración de Curiam, el CF, o un módulo basado en ellos, en éstos procesos. Como punto de partida, esta integración puede llevarse a cabo incorporando un CDSS que proporcione ayuda a la decisión en determinados nodos a la hora de escoger a qué estado continuar. En un sistema completamente integrado con el sistema de información del hospital, la ayuda a la decisión podría proporcionarse sin la intervención manual del usuario, lo cual beneficiaría el rendimiento global del proceso y aumentaría la apreciación del personal clínico por estos sistemas. Más adelante, sistemas de evaluación podrían llegar a proporcionar *feedback* sobre la eficiencia de las guías existentes y sugerir nuevos caminos o incluso recomendaciones en las pruebas realizadas a la hora de tomar decisiones.
- Hasta el momento, en el CF han sido integrados determinados métodos de minería de datos. Como trabajo futuro en el CF se pretende incorporar nuevos métodos que permitan la instanciación de clasificadores basados en reglas, como por ejemplo los modelos C4.5, C5 o CART. Como consecuencia de la incorporación de dichos modelos, así mismo como de otros nuevos, se hace interesante proporcionar al CF la compatibilidad con el estándar de representación de modelos de minería de datos PMML. Consiste esta tarea, por tanto, en otro trabajo futuro, que permitiría integrar en el CF modelos entrenados desde herramientas de minería de datos externas, como Clementine o Weka.
- En [37] se definió un método para la caracterización de patrones de datos de MRS usando mixturas para datos truncados y agrupados, basado en los métodos descritos en [14, 15]. Este método fue aplicado de forma experimental para el pre-proceso de datos de MRS para clasificación de tumores cerebrales. Queda como trabajo futuro, su inclusión como técnica de preproceso en el CF para permitir así el uso de clasificadores desarrollados con este método.
- Otro de los objetivos para Curiam es la incorporación de un sistema que permita la evaluación dinámica de los clasificadores mediante un modelo estadístico y otro que proporcione la capacidad de re-entrenar automáticamente dichos modelos con nuevos datos. Estos trabajos están siendo el desarrollo de dos tesis doctorales en el grupo IBIME.
- La visualización de resultados de clasificación basada en la ontología de dominio desarrollada como parte de Curiam — sección 4.6 — se trata actualmente de una prueba de concepto, a la espera de un estudio más conciso. La investigación en este método de representación de conocimiento supone un trabajo futuro importante para mejorar la interpretabilidad de los DSS en general, especialmente en el ámbito clínico.

---

<sup>3</sup>Intersystems Ensembl, <http://www.intersystems.com/ensemble/>

# Bibliografía

- [1] C. Angulo, P. Crespo, J. A. Maldonado, D. Moner, D. Pérez, I. Abad, J. Mandingorra, and M. Robles. Non-invasive lightweight integration engine for building ehr from autonomous distributed systems. *International Journal of Medical Informatics*, 76(Supplement 3):S417 – S424, 2007. Ubiquity: Technologies for Better Health in Aging Societies - MIE 2006.
- [2] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, November 1995.
- [3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.
- [4] P. Comon. Independent component analysis, a new concept? *Signal Process.*, 36(3):287–314, 1994.
- [5] G. Coppini, S. Diciotti, M. Falchini, N. Villari, and G. Valli. Neural networks for computer-aided diagnosis: detection of lung nodules in chest radiograms. *IEEE Trans Inf Technol Biomed*, 7(4):344–357, 2003.
- [6] T. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
- [7] M. Croitoru, B. Hu, S. Dasmahapatra, P. H. Lewis, D. Dupplaw, A. Gibb, M. Julià-Sapé, J. Vicente, C. Sáez, J. M. García-Gómez, R. Roset, F. Estanyol, X. Rafael, and M. Mier. Conceptual graphs based information retrieval in healthagents. In *CBMS*, pages 618–623. IEEE Computer Society, 2007.
- [8] F. D. Davis. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, (Sept.):319–339, September 1989.
- [9] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, New York, 2nd. edition, 2000.
- [10] B. ES. Clinical decision support systems: State of the art. *Agency for Healthcare Research and Quality*, June 2009. AHRQ Publication No. 09-0069-EF.
- [11] R. A. Fisher. *Statistical methods for research workers*. Oliver and Boyd, Edinburgh, UK, 1925.
- [12] D. J. Foran, D. Comaniciu, P. Meer, and L. A. Goodell. Computer-assisted discrimination among malignant lymphomas and leukemia using immunophenotyping, intelligent image repositories, and telemicroscopy. *IEEE Trans Inf Technol Biomed*, 4(4):265–273, 2000.
- [13] J. García-Gómez, S. Tortajada, C. Vidal, M. Juliá-Sapé, J. Luts, A. Moreno-Torres, S. Van Huffel, C. Arús, and M. Robles. The effect of combining two echo times in automatic brain tumor classification by mrs. *NMR in Biomedicine*, 2008.
- [14] J. M. García-Gómez. Pattern recognition approaches for biomedical data in computer-assisted cancer research, chapter 4. *PhD. Thesis, Universidad Politécnica de Valencia*, 2009.
- [15] J. M. Garcia-Gomez, M. Robles, S. Van Huffel, and A. Juan-Císcar. Modelling of magnetic resonance spectra using mixtures for binned and truncated data. In Springer, editor, *Proceedings of the 1st Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, volume 2 of *Lecture Notes in Computer Science Series 4478*, pages 266–273, Girona, Spain, June 2007. Springer-Verlag.

- [16] J. M. García-Gómez, S. Tortajada, J. Vicente, C. Sáez, X. Castells, J. Luts, M. Julià-Sapé, A. Juan-Císcar, S. Van Huffel, A. Barceló, J. Ariño, C. Arus, and M. Robles. Genomics and metabolomics research for brain tumour diagnosis based on machine learning, 2007.
- [17] J. M. Garcia-Gomez, C. Vidal, L. Marti-Bonmati, J. Galant, N. Sans, M. Robles, and F. Casacuberta. Benign/malignant classifier of soft tissue tumors using mr imaging. *Magn. Reson. Mater. Phy.*, 16(4):194–201, 2004.
- [18] J. M. García-Gómez, C. Vidal, L. Martí-Bonmatí, and M. Robles. Distributed decision support architecture for soft tissue tumor diagnosis. In *MEDINFO 2004: 11th World Congress on Medical Informatics*, page 1608, 2004.
- [19] H. González-Vélez, M. Mier, M. Julià-Sapé, T. N. Arvanitis, J. M. García-Gómez, M. Robles, P. H. Lewis, S. Dasmahapatra, D. Dupplaw, A. Peet, C. Arús, B. Celda, S. V. Huffel, and M. Lluch i Ariet. HealthAgents: Distributed multi-agent brain tumor diagnosis and prognosis. *Applied Intelligence*, 30(3):191–202, June 2009. Thomson Scientific Impact Factor (2007): 0.500.
- [20] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, August 2001.
- [21] D. L. Hunt, R. B. Haynes, S. E. Hanna, and K. Smith. Effects of computer-based clinical decision support systems on physician performance and patient outcomes: A systematic review. *JAMA*, 280(15):1339–1346, 1998.
- [22] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.
- [23] J.-F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEEE Proceedings-F*, 140(6):362–370, 1993.
- [24] B. M. Kelm, B. H. Menze, C. M. Zechmann, K. T. Baudendistel, and F. A. Hamprecht. Automated estimation of tumor probability in prostate magnetic resonance spectroscopic imaging: pattern recognition vs quantification. *Magn Reson Med*, 57(1):150–159, 2007.
- [25] P. J. G. Lisboa. A review of evidence of health benefit from artificial neural networks in medical intervention. *Neural Networks*, 15(1):11–39, 2002.
- [26] P. J. G. Lisboa, H. Wong, P. Harris, and R. Swindell. A Bayesian neural network approach for modelling censored data with an application to prognosis after surgery for breast cancer. *Artif Intell Med*, 28(1):1–25, 2003.
- [27] M. Lluch-Ariet, F. Estanyol, M. Mier, C. Delgado, H. González Velez, T. Dalmas, M. Robles, C. Sáez, J. Vicente, S. Huffel, J. Luts, C. Arús, A. P. C. Silveira, M. Julià-Sapé, A. Peet, Y. Gibb, Alexand Sun, B. Celda, M. C. M. Bisbal, G. Valsecchi, B. Dupplaw, Davida nd Hu, and P. Lewis. *Agent-Technology and e-Health*, chapter On the Implementation of HealthAgents: Agent-Based Brain Tumour Diagnosis, pages 5–24. 2008. 10.1007/978-3-7643-8547-7\_2.
- [28] D. Louis, H. Ohgaki, O. Wiestler, W. Cavenee, P. Burger, A. Jouvet, B. Scheithauer, and P. Kleihues. The 2007 WHO Classification of Tumours of the Central Nervous System. *Acta Neuropatol*, pages 114:97–109, 2007.
- [29] P. J. Lucas, H. Boot, and B. G. Taal. Computer-based decision support in the management of primary gastric non-hodgkin lymphoma. *Methods Inf Med*, 37(3):206–219, 1998.
- [30] J. A. Maldonado, D. Moner, D. Boscá, J. T. Fernández-Breis, C. Angulo, and M. Robles. Linkehr-ed: A multi-reference model archetype editor based on formal semantics. *International Journal of Medical Informatics*, 78(8):559 – 570, 2009.



- [31] G. S. Nayak, S. Kamath, K. M. Pai, A. Sarkar, S. Ray, J. Kurien, L. D’Almeida, B. R. Krishnanand, C. Santhosh, V. B. Kartha, and K. K. Mahato. Principal component analysis and artificial neural network analysis of oral tissue fluorescence spectra: classification of normal premalignant and malignant pathological conditions. *Biopolymers*, 82(2):152–166, 2006.
- [32] G. P. Purcell. What makes a good clinical decision support system. *BMJ*, 330, April 2005. *BMJ* 2005,330 740/1.
- [33] R. Ratei, L. Karawajew, F. Lacombe, K. Jagoda, G. Del Poeta, J. Kraan, M. De Santiago, J. Kappelmayer, E. Bjorklund, W.-D. Ludwig, J. W. Gratama, and A. Orfao. Discriminant function analysis as decision support system for the diagnosis of acute leukemia with a minimal four color screening panel and multiparameter flow cytometry immunophenotyping. *Leukemia*, 21(6):1204–1211, 2007.
- [34] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.
- [35] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. pages 318–362, 1986.
- [36] C. Sáez. Desarrollo de una ontología y un módulo software para la selección de clasificadores de tumores cerebrales. *Agentes de Información y Web Semántica, Máster de Posgrado en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital 2007, Universidad Politécnica de Valencia*, 2007.
- [37] C. Sáez. Modelling of Magnetic Resonance Spectra patterns using mixtures for binned and truncated data. *Statistical Pattern Analysis, Máster de Posgrado en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital 2009, Universidad Politécnica de Valencia*, 2009.
- [38] C. Sáez, J. M. García-Gómez, J. Vicente, S. Tortajada, M. Esparza, A. T. Navarro, E. Fuster-García, M. Robles, L. Martí-Bonmatí, and C. Arús. A generic Decision Support System featuring an assembled view of predictive models for Magnetic Resonance and clinical data. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 21:483, Sep 2008. 10.1007/s10334-008-0126-2.
- [39] C. Sáez, J. M. García-Gómez, J. Vicente, S. Tortajada, E. Fuster-García, M. Esparza, A. T. Navarro, and M. Robles. Curiam BT 1.0, decision support system for brain tumour diagnosis. *Magnetic Resonance Materials in Physics, Biology and Medicine*, 22:538, Oct 2009. 10.1007/s10334-009-0178-y.
- [40] C. Sáez, J. M. García-Gómez, J. Vicente, S. Tortajada, J. Luts, D. Dupplaw, S. V. Huffel, and M. Robles. A generic and extensible automatic classification framework applied to brain tumour diagnosis in healthagents. *The Knowledge Engineering Review*, 2010. Accepted for publication.
- [41] J. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson. Past, present, and future of decision support technology. *Decision Support Systems*, 2002.
- [42] E. H. Shortliffe, A. C. Scott, M. B. Bischoff, A. B. Campbell, W. va. Melle, and C. D. Jacobs. Oncocin: An expert system for oncology protocol management. In *Seventh International Joint Conference on Artificial Intelligence*, Vancouver, 1981.
- [43] I. Sim, P. Gorman, R. Greenes, R. Haynes, B. Kaplan, H. Lehmann, and P. Tang. Clinical decision support systems for the practice of evidence-based medicine. *Journal of the American Medical Informatics Association*, 2001. 8(6),527/4.
- [44] P. Spyridonos, D. Cavouras, P. Ravazoula, and G. Nikiforidis. A computer-based diagnostic and prognostic system for assessing urinary bladder tumour grade and predicting cancer recurrence. *Med Inform Internet Med*, 27(2):111–122, 2002.
- [45] J. A. K. Suykens and J. Vandewalle. Least squares support vector machine classifiers. *Neural Process. Lett.*, 9(3):293–300, 1999.

- [46] A. R. Tate, J. Underwood, D. M. Acosta, M. Julià-Sapé, C. Majós, A. Moreno-Torres, F. A. Howe, M. van der Graaf, V. Lefournier, M. M. Murphy, A. Loosemore, C. Ladroue, P. Wesseling, J. L. Bosson, M. E. Cabañas, A. W. Simonetti, W. Gajewicz, J. Calvar, A. Capdevila, P. R. Wilkins, B. A. Bell, C. Rémy, A. Heerschap, D. Watson, J. R. Griffiths, and C. Arús. Development of a decision support system for diagnosis and grading of brain tumours using in vivo magnetic resonance single voxel spectra. *NMR Biomed.*, 19(4):411–434, 2006.
- [47] A. R. Tate, J. Underwood, D. M. Acosta, M. Julià-Sapé, C. Majós, A. Moreno-Torres, F. A. Howe, M. van der Graaf, V. Lefournier, M. M. Murphy, A. Loosemore, C. Ladroue, P. Wesseling, J. Luc Bosson, M. E. Cabanas, A. W. Simonetti, W. Gajewicz, J. Calvar, A. Capdevila, P. R. Wilkins, B. A. Bell, C. Remy, A. Heerschap, D. Watson, J. R. Griffiths, and C. Arús. Development of a decision support system for diagnosis and grading of brain tumours using in vivo magnetic resonance single voxel spectra. *NMR Biomed*, 19(4):411–434, Jun 2006.
- [48] S. Tortajada, J. García-Gomez, J. Vicente, J. Sanjuán, F. R. de, R. Martín-Santos, L. García-Esteve, I. Gornemann, A. Gutiérrez-Zotes, F. Canellas, A. Carracedo, M. Gratacos, R. Guillamat, E. Baca-García, and M. Robles. Prediction of postpartum depression using multilayer perceptrons and pruning. *Methods Inf Med*, 48(3):291–8, 2009.
- [49] S. Tortajada, J. García-Gómez, J. Vicente, and M. Robles. Dynamic learning of Brain Tumour classifiers based on 1H-MRS. In SpringerLink, editor, *Book of Abstracts ESMRMB 2008 - Supplement 1, Journal Magnetic Resonance Materials in Physics, Biology and Medicine*, volume 21, pages 14–15, 2008.
- [50] W. L. Tung and C. Quek. Genso-fdss: a neural-fuzzy decision support system for pediatric all cancer subtype identification using gene expression data. *Artif Intell Med*, 33(1):61–88, 2005.
- [51] Universitat Autònoma de Barcelona. INTERPRET project. web site, 2000–2002. <http://azizu.uab.es/INTERPRET/> (Last accessed: 5 Jan 2007).
- [52] H. Varonen, T. Kortteisto, and M. Kaila. What may help or hinder the implementation of computerized decision support systems (cdss): a focus group study with physicians. *Family Practice Advance Access*, May 2008.
- [53] C. Vidal, J. M. García-Gómez, L. Martí-Bonmatí, and M. Robles. Soft tissue tumour vascular group discriminatios. 14:661, 2004.
- [54] L. Xiao, A. Peet, P. Lewis, S. Dasmahapatra, C. Sáez, M. Croitoru, J. Vicente, H. González-Vélez, M. Ariet, D. Dupplaw, and A. Gibb. A security model and its application to a distributed decision support system for healthcare. In *Proceedings of the 3rd International Conference on Availability, Reliability and Security (ARES'08)*. IEEE Computer Society, 2008.
- [55] L. Xiao, A. Peet, P. Lewis, S. Dasmahapatra, C. Sáez, M. Croitoru, J. Vicente, H. González-Vélez, and M. Lluch i Ariet. An adaptive security model for multi-agent systems and application to a clinical trials environment. In *Proceedings of the 31st IEEE Annual International Computer Software and Applications Conference (COMPSAC '07)*, volume II, pages 261–268. IEEE Computer Society, 2007.
- [56] M. M. Zheng, S. M. Krishnan, and M. P. Tjoa. A fusion-based clinical decision support for disease diagnosis from endoscopic images. *Comput Biol Med*, 35(3):259–274, 2005.