



Universidad de Murcia – Facultad de Informática

DESARROLLO DE SISTEMAS INTELIGENTES

Alertas de ECG: Informe 1

Curso académico 2025-26

Profesores: Ricardo Javier Sendra Lázaro – Aurora González Vidal

Juan Rondán Ramos (j.rondanramos@um.es)

Domingo Pujante López de Ochoa (domingo.pujantel@um.es)

David Madrid del Amor (d.madridamor@um.es)

1. INTRODUCCIÓN

El principal objetivo de este informe es explicar las decisiones de diseño de una ontología para relacionar los datos extraídos de un electrocardiograma (ECG) con las posibles patologías que se pueden detectar a raíz de este.

2. DETERMINAR ALCANCE Y DOMINIO

Para garantizar que el proyecto sea manejable y enfocado, se han establecido unos límites claros sobre lo que la ontología incluye y lo que excluye.

Dentro de este alcance se abarcará: los conceptos del ECG, como los diferentes **tipos de ondas, de intervalos, complejos, relaciones...** Así como conceptos de **ritmo cardíaco, ciclos y picos máximos**. Por otro lado, se determinan las patologías específicas, donde únicamente se modelarán las patologías propuestas específicamente para esta práctica. Queda excluido cualquier otro aparato o prueba médica, enfermedades no propuestas, posibles historiales médicos de un paciente y cualquier otra información clínica adicional.

3. CONSIDERAR USO DE ONTOLOGÍA EXISTENTE

En cuanto a la consideración de reutilizar una ontología ya existente, por supuesto, se evaluó la posibilidad. Sin embargo, tras su análisis, se concluyó que, para los fines específicos de esta práctica, la opción más eficaz y directa era desarrollarla desde cero. La razón principal es que este proyecto tiene un alcance muy reducido y de bajo nivel, centrado exclusivamente en reglas de diagnóstico concretas que relacionan métricas específicas del ECG con una lista cerrada de patologías.

Las ontologías generales suelen ser excesivamente amplias y complejas para una tarea tan acotada. Adaptarlas o extraer de ellas solo el subconjunto necesario habría supuesto una carga de trabajo mayor que crear una ontología ligera y optimizada desde cero para cumplir con el objetivo concreto.

4. ENUMERACIÓN DE TÉRMINOS IMPORTANTES

Una vez establecido el alcance y dominio de la ontología a crear, se enumeran los términos que se deben incluir de algún modo en ella:

Onda, ECG, amplitud, ciclos, pico máximo, segmento, intervalo, complejo, enfermedad, patología, frecuencia cardíaca, resultado, diagnóstico.

Algunos de estos términos pueden ser considerados sinónimos, pero se ha creído conveniente añadirlos, en cualquier caso. En próximos pasos es posible que aparezcan nuevos términos o que algunos de estos, se agrupen en uno.

5. DEFINICIÓN DE CLASES

Para la creación de clases, se ha optado por utilizar un enfoque combinado, ya que, una vez definidos los conceptos más generales, posteriormente se especializarán y generalizarán apropiadamente.

En primera instancia se crearon las siguientes clases: **Onda y Resultado**. También en un principio, se consideró la opción de que estas fueran clases abstractas, de las que heredarían otras Clases como **OndaP, OndaQ...** o **Bradycardia, Taquicardia...** Sin embargo, para evitar una creación de clases que se consideró innecesaria, se optó por resolver ambos casos con dos enumerados diferentes. El primero es **Tipos_Ondas**, dónde únicamente se permiten los posibles tipos de ondas (**P/Q/R/S/T**). El segundo es **Enfermedades**, donde se hace algo parecido, incluyendo únicamente las posibles enfermedades a diagnosticar (**TAQUICARDIA, BRADICARDIA, HIPOCALCEMIA, HIPOPOTASEMIA, INFARTO_AGUDO_DE_MIOCARDIO, CONTRACCIÓN_VENTRICULAR_PREMATURA, ISQUEMIA_CORONARIA**). De este modo, se ideó gestionar el tipo de onda y de enfermedad, desde un atributo dentro de esa misma clase, acotado por el mismo enumerado.

En el transcurso de la práctica, a la hora de crear reglas para el diagnóstico de enfermedades donde fuera necesario utilizar un intervalo concreto, quedó en evidencia la falta de recursos que se tenía en el sistema. Es por ello que se optó por añadir una nueva clase: **Intervalo**. En esta clase no se ha definido un enumerado para acotar los tipos de segmentos a tratar, ya que, el análisis de los mismos es tan complejo que se habría tenido que modificar el enumerado en más de una ocasión. A esto se le suma otra clase, **ContadorCiclos**, que, principalmente ha ayudado a identificar ondas que se encuentren en el mismo ciclo, para que los intervalos solo puedan ser tratados con ondas de diferentes tipos y del mismo ciclo.

6. DEFINICIÓN DE SLOTS, FACETAS Y RESTRICCIONES

Para complementar las clases creadas, es necesario añadir ciertos conceptos para dar sentido a las clases.

Clase **Onda**: Se añaden los atributos: **tipo de onda** (**Tipos_Ondas**), **inicio** (float), **fin** (float), **pico máximo** (float), **duración** (float) y **ciclo** (int). El atributo **duración**, es un atributo calculado a partir de la diferencia entre fin e inicio. El atributo **ciclo** indicará el ciclo al que pertenece la onda. El **resto de atributos** son extraídos directamente del fichero de entrada y serán necesarios para realizar diversos cálculos.

Clase **Resultado**: Se añaden los atributos: **número de ciclos**, (float) **ritmo cardiaco** (float) y una lista, **enfermedad**, con todas las enfermedades que se han detectado. Todo esto es lo necesario para ofrecer un diagnóstico completo.

Clase **Intervalo**: Se añaden los atributos: **tipo** (string), **duración** (float), **ciclo** (int). De esta forma se asegura que las ondas que componen un intervalo pertenecen al mismo ciclo.

Clase **Contador Ciclos**: Se añade el atributo **ciclo actual** (int) para poder asignar a cada onda su ciclo correspondiente.

En cuanto a las restricciones, se añade:

Restricción 1: El valor fin de una **Onda** no puede ser anterior al inicio de esa misma **Onda**.

Restricción 2: El atributo duración, tanto de una **Onda** como de un **Intervalo**, no puede tener un valor negativo.

Restricción 3: El atributo ciclo, tanto de una **Onda** como de un **Intervalo**, nunca podrá ser superior al atributo número de ciclos de la clase **Resultado**.

7. CREACIÓN DE INSTANCIAS Y LECTURA DEL FICHERO DE ENTRADA

Por otro lado, fuera de lo que a la ontología se refiere, se han creado dos clases para la lectura del fichero de los datos de entrada, así como para almacenar las ondas leídas. Estas clases son: **LectorECG**, que con ayuda de expresiones regulares se consigue separar los diferentes datos (**tipo de onda, inicio, fin, pico**), creando una instancia de **Onda** por cada línea leída, asignando cada grupo leído a su correspondiente atributo, y almacenándolas en una lista. Esta lista será utilizada para crear una instancia de otra clase **EntradaElectro**, cuyo atributo es dicha lista con todas las Ondas leídas.

Además, en el main, se crea una instancia de: **Resultado, Intervalo y Contador de Ciclos**, cuyos atributos serán modificados conforme se vayan lanzando las reglas de Drools, atendiendo al pertinente estudio de posibles síntomas y enfermedades.