TRABAJO DE FIN DE GRADO

Creación y gestión de bases de conocimiento para generar modelos de enfermedades

Samir Sarmiento Padrón
alu0101028491@ull.edu.es

San Cristóbal de La Laguna, 22 de mayo de 2023

D. **Iván Castilla Rodríguez**, con N.I.F. 78.565.451-X, Profesor Contratado Doctor adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como tutor.

D. **Evelio José González González**, con N.I.F. 43.372.115-A, Profesor Titular de Universidad adscrito al Departamento de Ingeniería Informática y de Sistemas de la Universidad de La Laguna, como cotutor.

C E R T I F I C A N

Que la presente memoria titulada:

"Creación y gestión de bases de conocimiento para generar modelos de enfermedades"

ha sido realizada bajo su dirección por D. **Samir Sarmiento Padrón**, con N.I.F. 51.166.398-T.

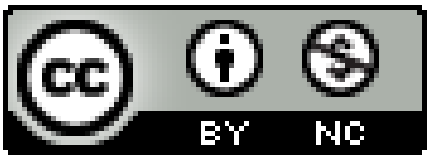
Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna, en la fecha que figura en la firma electrónica.

Agradecimientos

A nivel personal debo agradecer a todas aquellas personas que me han permitido conformar mi visión actual de la vida, tanto las que me dieron enfoques positivos como negativos, facilidades y dificultades, alegrías y agonías. No es necesaria ninguna mención directa, las personas objeto de mi gratitud ya lo saben, las que lo dudan sólo acaban de responderse, sin ellas no tendría los puntos de apoyo que me han permitido afrontar las dificultades del camino y saber quitarme el polvo tras cada caída, la mitad del trabajo es suyo, de nuevo, gracias.

Académica y profesionalmente me es imposible no mencionar a mis tutores del proyecto, ambos me han ofrecido un excelente ambiente de trabajo y de retroalimentación investigativa que creo que nos ha beneficiado a todos. Un placer, caballeros.

Licencia



Resumen

En España, la ley obliga a realizar una evaluación económica de los pros y contras de la financiación de cualquier tecnología sanitaria, ya sea esta una de nueva creación o una ampliación sobre una base preexistente. Para realizar estas evaluaciones es habitual plantear modelos que representan cómo evoluciona la enfermedad y permiten medir tanto los resultados económicos como en la salud de los individuos, aunque su desarrollo puede llegar a ser muy complejo, por lo que el uso de herramientas informáticas que permitan automatizar o semi-automatizar parte de los pasos puede ser una gran ayuda para los investigadores. Cada vez es más habitual el uso de modelos de simulación debido a que, en determinados aspectos, presentan una mayor flexibilidad que su formato tradicional de realización como parte de una evaluación económica dentro de un ensayo clínico.

En este proyecto se ha planteado el desarrollo de una ontología de dominio específico, denominada StaDiOS, para almacenar el conocimiento de distintos estudios económico-sanitarios sobre enfermedades de cualquier tipo. Una vez obtenida y modelizada dicha información, se implementa una aplicación para permitir su reutilización, facilitando la generación de análisis coste-efectividad mediante árboles de decisión. Los análisis generados permitirán la comparación entre distintas intervenciones sanitarias aplicadas a una población que sufre una determinada enfermedad, además de permitir la obtención de inferencia adicional a partir de diversas fuentes de datos en conjunto con StaDiOS.

Palabras clave: Análisis coste-efectividad, Árboles de decisión, Ontología, Web Semántica, Lenguaje Ontológico Web, Enfermedades.

Abstract

In Spain, the law requires an economic evaluation of the pros and cons of financing any health technology, whether it is a new technology or an extension of an existing one. To carry out these evaluations, it is common to propose models that represent how the disease evolves and allow the measurement of both economic and health outcomes for individuals. Although their development can be very complex, the use of computer tools that allow part of the steps to be automated or semi-automated can be a great help to researchers. The use of simulation models is becoming increasingly common because, in certain respects, they are more flexible than their traditional format of being conducted as part of an economic evaluation within a clinical trial.

In this project, the development of a specific domain ontology, called StaDiOS, has been proposed to store the knowledge of different economic-sanitary studies on diseases of any type. Once this information has been obtained and modeled, an application is implemented to allow its reuse, facilitating the generation of cost-effectiveness analysis through decision trees. The analyses generated will allow the comparison between different health interventions applied to a population suffering from a certain disease, in addition to allowing additional inference to be obtained from various data sources in conjunction with StaDiOS.

Keywords: Cost-Effectiveness Analysis, Decision Trees, Ontology, Semantic Web, Web Ontology Language, Diseases.

Índice

Índice de figuras

Índice de tablas

Acrónimos

- **RaDiOS:** Rare Disease Ontology for Simulation.
- **RaDiOS-MTT:** Rare Disease Ontology for Simulation Modeling Transformation Tool.
- **StaDiOS:** Standard Disease Ontology for Simulation.
- **URI:** Uniform Resource Identifier.
- **RDF:** Resource Description Framework.
- **RDFS** Resource Description Framework Schema.
- **W3C:** World Wide Web Consortium.
- **OWL:** Web Ontology Language.
- **SPARQL:** SPARQL Protocol and RDF Query Language.
- **QALY:** Quality Adjusted Life Years.
- **AVAC:** Años de Vida Ajustados por Calidad.
- **ICER:** Incremental Cost-Effectiveness Ratio.
- **ACE:** Análisis Coste Efectividad.
- **WIDOCO:** Wizard for Documenting Ontologies.
- **SNOMED CT:** Systematized Nomenclature of Medicine – Clinical Terms.
- **MONDO:** Mondo Disease Ontology.
- **IPC:** Índice de Precios al Consumidor.
- **HTTP:** Hypertext Transfer Protocol.
- **SQL:** Structured Query Language.
- **XML:** Extensible Markup Language.
- **TURTLE:** Terse RDF Triple Language.
- **JSON-LD:** JavaScript Object Notation for Linked Data.

1. Introducción

1.1. Contexto

En España, la legislación actual obliga a realizar una evaluación económica de los pros y contras de la financiación de cualquier tecnología sanitaria. Para realizar estas evaluaciones es común plantear modelos que analicen cómo evoluciona la enfermedad y permitan medir tanto los resultados económicos como en la salud de los individuos. El desarrollo de estos modelos puede ser muy complejo, llegando en algunos casos a ocupar una importante cantidad de recursos, de forma que el uso de herramientas informáticas que permitan automatizar o semi-automatizar parte de los pasos puede ser una gran ayuda para los investigadores. En este Trabajo de Fin de Grado se pretende crear sistemas de información que permitan recopilar la información de diversas enfermedades e intervenciones sanitarias de una forma, dentro de lo posible estandarizada, que haga viable la generación automatizada de modelos que las representen. Posteriormente estos modelos se gestionarán y visualizarán a través de herramientas relacionadas con la Web Semántica [4].

1.2. Estado del arte

Se ha tenido la oportunidad de participar en una línea de investigación que sigue activa en la universidad, de forma que se cuenta con distintos recursos en consonancia con la misma temática investigativa y proporcionan una base sólida y contrastada de conocimiento para embarcarse en este ámbito. A continuación se adjuntan los antecedentes que fundamentan este trabajo.

- **Towards the automated economic assessment of newborn screening for rare diseases [5]:** Se encuentran parámetros clave del modelado sobre epidemiología, métodos de cribado, métodos de diagnóstico, patogénesis, tratamiento y pruebas de seguimiento. También se identifican siete repositorios de datos directamente relacionados con las enfermedades raras, aunque ninguno de estos repositorios era adecuado para la generación automática de modelos de simulación. Se incorporan los parámetros identificados como clases estructuradas y propiedades de una nueva ontología, Rare Disease Ontology for Simulation (RaDiOS). Se establecen cuidadosamente las relaciones entre los parámetros para permitir la inferencia automática a partir de la ontología.
- **Automated generation of decision-tree models for the economic assessment of interventions for rare diseases using the RaDiOS ontology [6]:** Se actualiza la ontología de referencia RaDiOS para incluir más conocimientos necesarios para generar un modelo. Se implementa una herramienta de transformación, Rare Disease Ontology for Simulation Modeling Transformation Tool (RaDiOS-MTT), que utiliza el conocimiento almacenado en RaDiOS para generar automáticamente árboles de decisión para la evaluación económica de intervenciones en enfermedades raras. Se utiliza un estudio de caso para ilustrar el potencial de la herramienta y generar automáticamente un árbol de decisión que reproduce un estudio real sobre el cribado neonatal de la deficiencia profunda de biotinidasa.

1.3. Objetivos y planificación temporal del proyecto

Los objetivos principales a conseguir en este proyecto han sido:

- La implementación de un repositorio ontológico de dominio específico sobre diversas enfermedades que permita conservar el conocimiento recopilado a lo largo de la investigación. Los antecedentes de este proyecto sólo se enfocaban en la modelización de enfermedades raras, por lo que se pretende extender y extrapolar dicho conocimiento a cualquier tipo de enfermedad.
- El diseño y la configuración de dicho repositorio para que permita generar, de forma eficiente y estructurada, modelos para evaluaciones económicas de distintas tecnologías médicas.
- La implementación de métodos que permitan la generación de dichos modelos de forma semi-automatizada para las evaluaciones económicas.
- Creación de nueva inferencia a partir de la ontología a implementar, ya sea únicamente mediante los datos cargados en la misma como por el uso coordinado de múltiples fuentes externas.

En la figura 1.1 se muestra la planificación temporal seguida para este proyecto.

Name	Start Date	End Date	Duration
Proyecto de fin de grado	Nov 07, 2022	Jun 01, 2023	149 days
Creación de un repositorio personal para manipular nuestra ontología	Nov 07, 2022	Dec 12, 2022	26 days
Comprensión de las ontologías en el campo de la ciencia de datos	Nov 07, 2022	Dec 12, 2022	26 days
Estudio posibles modificaciones a los modelos ontológicos heredados	Dec 13, 2022	Feb 13, 2023	45 days
Ampliación y mejora de la ontología	Feb 13, 2023	May 22, 2023	70 days
Mejora de las herramientas de transformación e inferencia	Feb 13, 2023	May 22, 2023	70 days
Redacción y documentación - Entrega TFG	May 19, 2023	Jun 01, 2023	10 days

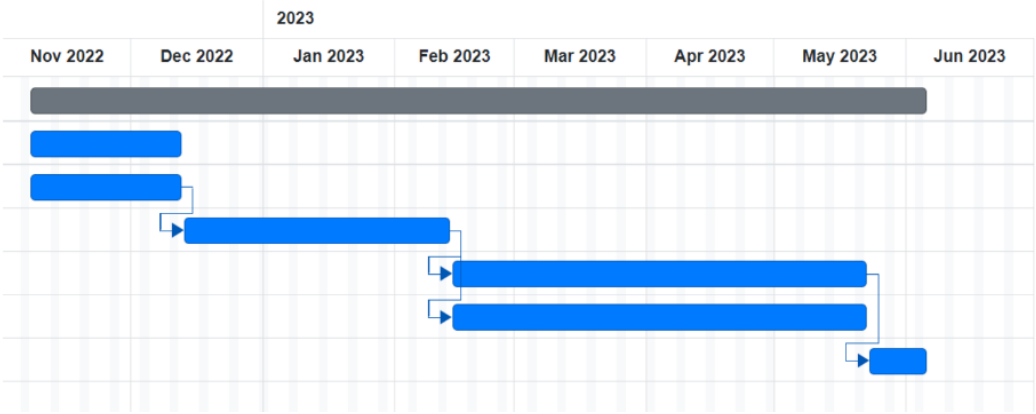


Figura 1.1: Planificación temporal - Diagrama de Gantt

2. Conceptos básicos

2.1. Web Semántica

Para explicar este concepto se debe realizar una pequeña retrospección en base a las distintas fases de desarrollo que ha experimentado la web actual (World Wide Web) [7]:

- **Web 1.0:** Concepto de web que se tenía durante el siglo XX. Se concibe la información web como un conjunto de páginas estáticas no fácilmente actualizables a las que los usuarios tenían acceso de forma unidimensional y algo tosca.
- **Web 2.0:** Concepto de web que se tenía aproximadamente a mediados de la primera década del siglo XXI. Aquí la web no era una herramienta meramente informativa, sino que se empezó a gestar la idea de que podía llegar a ser algo más bien colaborativo, un lugar donde la información se podía difundir desde múltiples puntos bajo distintas finalidades, por ejemplo, comerciales, ocio, divulgación, socialización, etc.
- **Web 3.0:** Concepto al que se quiere llegar y que comúnmente es llamado Web Semántica [4]. A partir de este punto se empieza a considerar a la web como algo inteligente. Aquí es donde, teóricamente, se consigue que las máquinas entiendan de alguna forma los datos que manejan y las relaciones que existen entre los mismos a un nivel mucho más complejo. Se destaca la pretensión de que esta compresión de la información por parte de los sistemas de cómputo se lleve a cabo de forma totalmente automática, consiguiendo la realización de diversas operaciones lógicas de la misma sin la intervención de operadores humanos.

Ya se puede ver una diferencia fundamental entre la web 2.0 y la 3.0, la Web Semántica intenta enlazar distintos tipos de conceptos o datos entre sí y que estos enlaces sean entendidos por las máquinas, mientras que la 2.0 enlaza documentos analizados a un nivel más superficial, sin compresión real por parte de los sistemas de cómputo. Dicho esto y bajo la premisa anterior, se debe destacar también que los datos con los que se trabaja en la web 3.0 necesitarán unos requisitos fundamentales para su uso en condiciones ideales.

- **Datos enlazados (Linked Data):** La información segregada por toda la web debe tener conexiones o relaciones entre sí, de forma que los sistemas de cómputo puedan entenderla a un nivel superior al que lo hacen de forma mayoritaria hoy en día. Esto se consigue mediante el uso de URIs (Uniform Resource Identifier), la definición de modelos de datos y herramientas de consulta de información bajo rigurosos estándares, y la inclusión de URIs de recursos externos dentro de las propias bases de datos, aumentando la conectividad de las mismas entre sí de una forma exponencial.
- **Datos abiertos (Open Data):** La información de distintos ámbitos que pueda tener relevancia para la sociedad, principalmente la recogida por todas las administraciones públicas a nivel global, debería estar publicada en distintos formatos electrónicos que faciliten su visualización, manipulación e interconexión sin restricciones legales notables.

2.2. Marco de descripción de recursos

Modelo estándar para representar datos interconectados en la web. Las declaraciones Resource Description Framework (RDF) [8] se utilizan para describir e intercambiar metadatos, lo que permite un intercambio normalizado de los mismos basado en relaciones que pueden ser entendidas tanto por operadores humanos como artificiales. La Web Semántica se basa en el uso del marco RDF para organizar la información basándose en significados, generando grafos dirigidos para permitir representar la información web en base a las entidades que conforman dichos grafos.

El Consorcio World Wide Web (W3C) conserva los estándares de RDF, incluidos los conceptos fundamentales, la semántica y las especificaciones de los distintos formatos. La primera sintaxis definida para RDF se basó en el Lenguaje Extensible de Marcado (XML), aunque en la actualidad se utilizan con más frecuencia otras sintaxis como Terse RDF Triple Language (Turtle), JavaScript Object Notation for Linked Data (JSON-LD) y N-Triples.

Las declaraciones RDF constan de 3 partes fundamentales:

- **Sujeto (Subject):** Elemento o recurso descrito por la tripleta.
- **Predicado (Predicate):** Describe la relación entre el sujeto y el objeto.
- **Objeto (Object):** Elemento o recurso relacionado con el sujeto.

En la figura 2.1 se muestra una representación gráfica conceptual de las declaraciones RDF.

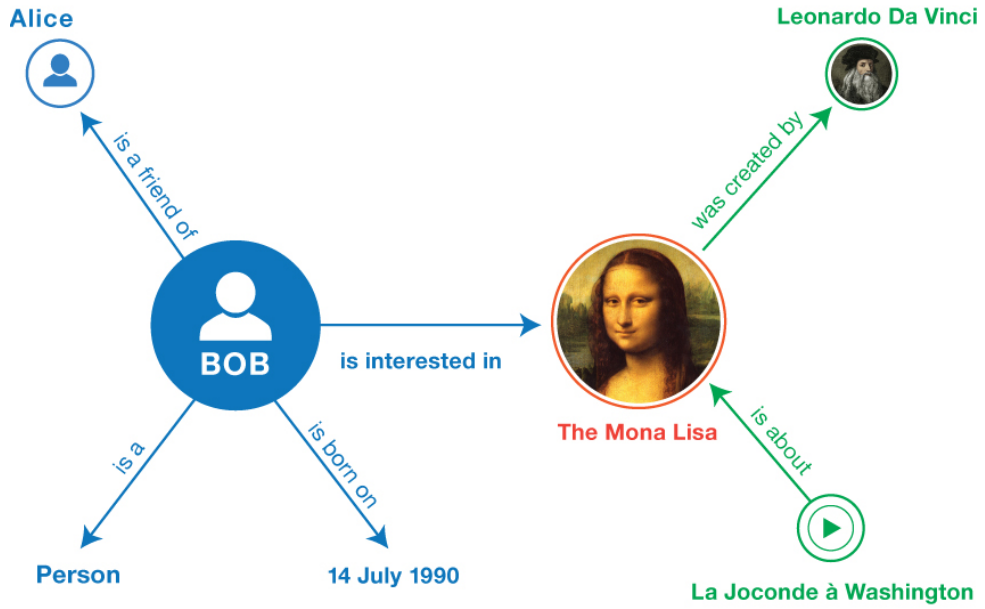


Figura 2.1: Ejemplo conceptual - Declaraciones RDF [1]

2.3. Esquema del marco de descripción de recursos

Si bien RDF permite crear enunciados simples para la descripción de relaciones existentes entre distintos elementos mediante tripletas, no le da importancia ni a dónde ni a cómo se almacenan los datos descritos, sirviendo más como un método de modelización conceptual de datos. En muchos casos se requiere, o directamente necesita, alguna herramienta que permita plasmar un tipado explícito sobre el vocabulario utilizado, pues para una máquina las relaciones y el formato de los datos que sintácticamente podrían parecer obvios no lo son.

RDF Schema (RDFS) [9] es un lenguaje diseñado para enfrentar estos casos, añadiéndose nuevo vocabulario a partir del heredado de RDF, por lo que se deduce y afirma que todo documento RDFS es también un documento RDF. RDFS permite empezar a hacer afirmaciones sobre clases de conceptos y distintos tipos de relaciones, además de permitir describir en un texto legible el significado de una relación o una clase.

El vocabulario principal introducido y ampliado en RDFS es:

- **Clases:** Los distintos recursos pueden dividirse en grupos denominados clases, siendo éstas elementos que sirven para describir de forma explícita la naturaleza de los datos asociados a cada recurso. Los miembros de una clase se conocen como instancias de la clase. Las clases son en sí mismas recursos y suelen tener identificadores únicos, además de describirse mediante propiedades RDF. En RDFS se distingue entre una clase y el conjunto de sus instancias, aunque dos clases pueden tener el mismo conjunto de instancias a pesar de ser clases diferentes. También se empieza a ver una jerarquía de conceptos, ya que si una clase A es subclase de otra B, entonces todas las instancias de A serán también instancias de B.
- **Propiedades:** Se desarrolla el concepto de propiedad RDF como una relación entre recursos sujeto y recursos objeto. También se define el concepto de subpropiedad, siendo esto nuevamente otra forma de permitir declarar de forma explícita la naturaleza de las relaciones existentes entre todos los recursos mediante un texto legible, tanto para humanos como máquinas. Si una propiedad P es una subpropiedad de la propiedad P', todos los pares de recursos relacionados por P también lo están por P'. El término superpropiedad se utiliza a menudo como inverso de subpropiedad. Si una propiedad P' es una superpropiedad de una propiedad P, entonces todos los pares de recursos que están relacionados por P también están relacionados por P'.

Puede ser algo complicado entender y explicar de forma clara las aportaciones que ofrece RDFS sobre las carencias de RDF, ya que se tratan cuestiones más bien semánticas de cómo se referencian, describen y relacionan determinados datos o conceptos a modelizar. Para facilitar la comprensión de estas aportaciones de una forma más visual y directa, se adjunta en la página a continuación la figura 2.2.

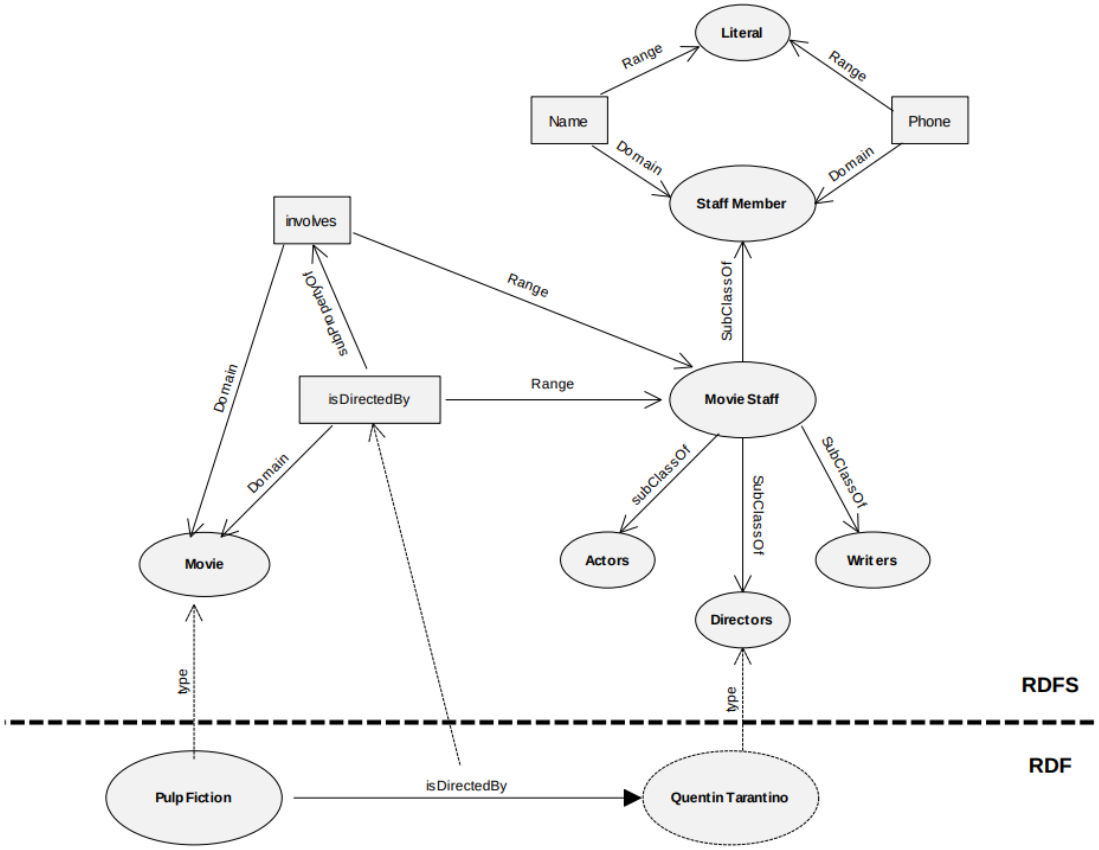


Figura 2.2: Ejemplo conceptual - RDF Schema [2]

2.4. Lenguaje ontológico web

Las ontologías son herramientas utilizadas para representar el conocimiento sobre algún dominio de interés, describiendo los conceptos fundamentales del dominio y las relaciones que existen entre los mismos, siendo Web Ontology Language (OWL) [10] el lenguaje ontológico estándar más reciente del Consorcio World Wide Web. Como se dijo anteriormente, RDFS permite expresar las relaciones entre distintos conceptos proporcionando un vocabulario RDF extendido, que se puede utilizar para describir en mayor profundidad y de forma explícita los conceptos en el área o áreas de interés, es decir, clases de conceptos, jerarquía de los mismos, tipos de datos y sus relaciones implicadas, etc. OWL es similar, pero más grande, permitiendo decir mucho más sobre un modelo de datos.

Las principales diferencias entre ambos lenguajes son las siguientes:

- **Vocabulario:** Si bien RDFS es una extensión de RDF, OWL es a su vez otra extensión de RDFS, por lo que objetivamente contiene mucho más vocabulario. OWL ofrece nuevos términos que permiten, por ejemplo, definir equivalencias entre bases de datos, restringir los valores de las propiedades, describir los datos en términos de operaciones de conjuntos, etc.
- **Consistencia lógica:** A diferencia de RDFS, OWL indica cómo se puede y cómo no se puede utilizar un determinado vocabulario. En otras palabras, mientras que RDFS no proporciona ningún mecanismo de restricción real, OWL sí lo hace. Por ejemplo, en RDFS cualquier cosa que se desee puede ser una instancia de “rdfs:Class”, siendo una instancia y una clase a la vez. Aunque esto se puede modelar en OWL, una simple comprobación de consistencia revelará la inconsistencia, es decir, es lógicamente inconsistente decir que algo puede ser tanto una clase como una instancia.
- **Anotaciones:** Cuando se desarrollan diversas ontologías que pueden estar relacionadas entre sí, al compartir un mismo dominio de conocimiento, puede ser interesante la creación de interconexiones entre las mismas. OWL facilita enormemente este tipo de cosas, pues goza de diversas opciones de importación y mapeo de términos entre ontologías, además de proporcionar una rica variedad de anotaciones como “owl:versionInfo”, “owl:backwardsCompatibleWith”, y “owl:deprecatedProperty”, que pueden usarse fácilmente para enlazar modelos de datos en una red de ontologías coherentes entre sí.

En base a lo explicado, queda expuesto que OWL proporciona un vocabulario más completo que permite desarrollar modelos de datos expresivos y rigurosos desde el punto de vista lógico. OWL permite adaptar lo que se dice en función de las realidades computacionales y los requisitos de las aplicaciones, siendo más adecuado que RDFS para su integración con bases de datos y la realización de consultas mediante sistemas gestores de bases de datos.

2.5. Protocolo y lenguaje de consulta RDF

Protocol and RDF Query Language (SPARQL) es el lenguaje oficial de consulta de la Web Semántica siendo, junto con RDF y OWL, una de sus tres tecnologías centrales. Mediante este lenguaje se obtiene la capacidad de recuperar y manipular datos almacenados en bases de datos en formato RDF, como se muestra de forma conceptual en la figura 2.3.

A diferencia de otros lenguajes de consulta de bases de datos, fue diseñado para poder operar sobre fuentes aparentemente desconectadas a través de la red, además de una base de datos local, en concreto, el protocolo SPARQL permite transmitir consultas y resultados SPARQL entre un cliente y un motor SPARQL a través del protocolo HyperText Transfer Protocol (HTTP), siendo este hecho lo que permite consultar uno o múltiples puntos finales SPARQL de forma simultánea y en tiempo real. Se debe aclarar que un punto final SPARQL (endpoint) es simplemente un servidor que expone sus datos a través del protocolo SPARQL. Este protocolo permite a los usuarios escribir consultas sobre datos que siguen la especificación RDF del W3C, por lo que toda la base de datos se entenderá como un conjunto de tripletas "sujeto-predicado-objeto", siendo esto análogo al uso que hacen algunas bases de datos NoSQL del término "documento-clave-valor", como MongoDB. También existen implementaciones para múltiples lenguajes de programación, además de herramientas, que permiten conectar y construir semi-automáticamente una consulta SPARQL para un endpoint SPARQL y traducir las consultas a otros lenguajes de consulta, como Structured Query Language (SQL) y XML Query (XQuery).

Dicho lo anterior y, de forma simplificada, se puede deducir que SPARQL será la herramienta principal a utilizar para expresar peticiones que permitan la consulta de diversas fuentes de datos, siempre que los mismos se almacenen de forma nativa como RDF o sean definidos mediante vistas RDF a través de algún sistema middleware. SPARQL contiene las capacidades para la consulta de los patrones obligatorios y opcionales de un grafo, junto con sus conjunciones y disyunciones. SPARQL también soporta la ampliación o restricción del ámbito de las consultas, indicando los grafos sobre los que se opera y donde los resultados de las consultas pueden ser conjuntos de resultados o directamente nuevos grafos RDF.

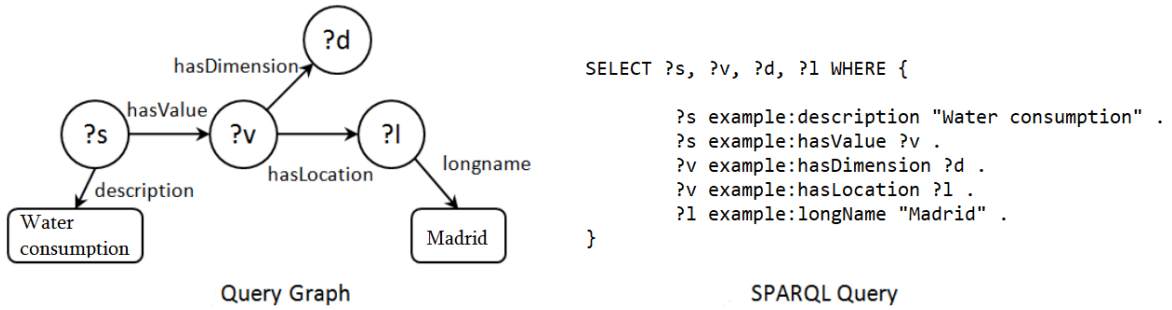


Figura 2.3: Ejemplo conceptual - Consultas SPARQL (Elaboración propia a partir de [3])

3. Ontología estándar de enfermedades para simulación

3.1. Descripción del problema

A partir de los antecedentes descritos en este proyecto se obtuvo una ontología de dominio específico, denominada RaDiOS, para almacenar de forma eficiente la información de las enfermedades raras de diversos estudios y facilitar así la generación de distintos modelos económicos. Una vez diseñada e implementada la ontología, se procedió a tratar de generar automáticamente uno de modelos de simulación más simples, los modelos de simulación basados en árboles de decisión, siendo este el punto desde el que parte este proyecto. El potencial de una ontología de esta naturaleza va más allá del subgrupo de enfermedades denominadas raras, teniendo que ser fácilmente extensible a cualquier otro tipo de enfermedad. Es por eso que, en consonancia con todos los precedentes expuestos, se ha contribuido en esa dirección creando una nueva ontología, Standard Disease Ontology for Simulation (StaDiOS).

StaDiOS es una ontología diseñada para los mismos propósitos que su antecesora, únicamente diferenciándose en que se ha generalizado su conceptualización para que, dentro de lo que ha sido posible en base al conocimiento adquirido durante la realización del proyecto, pueda albergar información sobre enfermedades de cualquier tipo. Otro aspecto muy importante es que, por el momento, esta ontología se ha diseñado específicamente pensando en estructurar la información para facilitar la creación de análisis económicos mediante árboles de decisión, concretamente los análisis de tipo coste-efectividad [11].

La información cargada en StaDiOS para este proyecto, aunque ampliada, pertenece al estudio del cribado neonatal aplicado a la deficiencia profunda de biotinidasa utilizado para la validación de modelos en RaDiOS [12]. La enfermedad en cuestión es considerada muy rara y se trata de un trastorno autosómico recesivo que se caracteriza por una alteración en el metabolismo de la biotina. Durante la infancia del paciente, es común su manifestación y provoca deficiencia múltiple de carboxilasas, lo que conduce a alteraciones en la síntesis de ácidos grasos, gluconeogénesis y catabolismo de los aminoácidos. La aparición del trastorno puede variar en edad y en algunos casos es precoz, pero en su mayoría se presenta entre el segundo y quinto mes de vida. Por lo tanto, el cribado neonatal es considerado como una técnica que puede ayudar en el diagnóstico temprano de esta enfermedad.

A pesar de que algunos expertos sugieren que este desorden reúne los requisitos necesarios para ser incluido en un programa de cribado neonatal [13][14], se obviará eso por el momento e intentará, únicamente con el análisis generado a partir de este proyecto, validar de forma objetiva los pros y contras de la intervención del cribado neonatal frente a la diagnosis directa de esta enfermedad.

3.2. Implementación

3.2.1. Herramientas de desarrollo

La creación y mejora de la ontología ha ido de la mano de las siguientes herramientas:

- **OWL:** StaDiOS ha sido implementada bajo el lenguaje OWL [10] de forma que, como se explicó en los conceptos básicos y en base a otros formatos RDF posibles, goza de una mayor expresividad en la representación de conocimientos y relaciones complejas, una semántica formal y rigurosa que permite una interpretación precisa y consistente de los conceptos y sus relaciones, inferencia y razonamiento automático, interoperabilidad con otras ontologías y aplicaciones que utilizan OWL, reutilización de vocabularios existentes y una amplia gama de herramientas de soporte.
- **Protégé:** Herramienta software de código abierto ampliamente utilizada para el diseño, desarrollo y gestión de ontologías. Es una herramienta gráfica y basada en Java que ofrece una interfaz de usuario intuitiva para la creación y edición de ontologías en formatos RDF, especialmente en OWL. Protégé [15] proporciona una serie de características y funcionalidades avanzadas para el modelado del conocimiento, como la definición de clases, propiedades, instancias, axiomas y reglas de inferencia. También ofrece capacidades de razonamiento y validación de ontologías, así como la posibilidad de importar, exportar y reutilizar ontologías existentes. Protégé es ampliamente utilizado por la comunidad de la Web Semántica y es una herramienta de referencia en el diseño y desarrollo de ontologías para aplicaciones de inteligencia artificial, sistemas de gestión del conocimiento, integración de datos, entre muchas otras cosas.
- **WIDOCO:** Wizard for Documenting Ontologies [16] es una herramienta de software de código abierto que permite la generación automática de documentación en la web para ontologías RDF. WIDOCO está diseñado para facilitar la documentación de ontologías creadas en OWL y RDF mediante la generación de páginas web semánticas interactivas y navegables. La herramienta simplifica el proceso de documentación de ontologías, proporcionando una interfaz gráfica de usuario para personalizar y generar automáticamente documentación en la web a partir de los metadatos y comentarios incrustados en los archivos RDF y OWL. También permite la generación de documentación en varios idiomas y ofrece una serie de plantillas de diseño para personalizar la apariencia de la documentación generada.

La herramienta de esta sección con la que más se diferencia este trabajo de sus antecedentes es WIDOCO. Con ella y, a diferencia de las investigaciones previas, se tiene la capacidad de visualizar de una forma relativamente cómoda para cualquier usuario toda la información sobre el diseño y la información contenida en la ontología, además de incluso generar análisis sobre el grado de calidad de su diseño, aunque todo esto será expuesto de forma más clara y precisa durante las conclusiones del trabajo.

3.2.2. Referencias ontológicas

El hecho de utilizar ontologias OWL sitúa al proyecto en un lugar donde el vocabulario que se utilice y la modelización de determinados conceptos no tiene ni debe de ser algo exclusivo de su invención. Al trabajar en un ámbito tan amplio como lo es el ámbito médico, el desarrollo debería estar fundamentado por una terminología estándar, hecha previamente por profesionales o entidades con el propósito de facilitar un vocabulario de trabajo común entre investigadores.

Las fuentes terminológicas principales de este desarrollo han sido:

- **SNOMED CT:** Systematized Nomenclature of Medicine – Clinical Terms [17]. Aunque no utiliza ningún lenguaje ontológico específico, es considerada una ontología de muy alto prestigio de términos clínicos estandarizada y ampliamente utilizada en el ámbito de la salud, pues describe y representa la información relacionada con diagnósticos, procedimientos, medicamentos, hallazgos clínicos, entre otros. Esta ontología compone un vocabulario clínico completo y jerarquizado que permite la representación estructurada y semántica de la información clínica. SNOMED CT es mantenida y desarrollada por la International Health Terminology Standards Development Organisation (IHTSDO), y su uso es promovido por muchas organizaciones de salud y estándares de interoperabilidad a nivel global.
- **MONDO:** Mondo Disease Ontology [18] es una iniciativa colaborativa de la comunidad científica que tiene como objetivo desarrollar una ontología OWL para la representación de información relacionada con enfermedades humanas y su clasificación, consolidando y unificando la representación de enfermedades mediante diversas fuentes de datos y recursos terminológicos. La ontología es desarrolla en el contexto del Consorcio Monarch Initiative, que busca mejorar la representación y la interoperabilidad de datos en el campo de la biomedicina.

Haciendo una descripción más específica, SNOMED CT se ha utilizado principalmente para la terminología a nivel estructural de las clases existentes en StaDiOS, mientras que MONDO se ha utilizado principalmente para la descripción de las propiedades de datos asociadas a las instancias de las clases de StaDiOS, concretamente para los datos que permiten describir las distintas características de una enfermedad, junto a su presentación y desarrollo.

Aunque gracias a OWL se pueden crear mapeos directos al vocabulario de otras ontologías, por el momento simplemente se han utilizado las fuentes referenciadas para intentar que la generalización realizada tenga un soporte terminológico oficial y sostenible. Es importante decir que esto no se aplica únicamente a este proyecto, es extremadamente conveniente consultar estas fuentes o similares para seguir mejorando el desarrollo de StaDiOS en un futuro, pues hacerlo permitirá el acceso a una inferencia mucho más sofisticada y contrastada mediante otras bases de conocimiento.

3.2.3. Generalizaciones y mejoras

Para pasar de RaDiOS a StaDiOS la ontología ha sufrido determinados cambios a distintos niveles, concretamente los tres siguientes:

- **Estructurales:** Se han incluido nuevas clases a las ya existentes, además de crear abstracciones para aglomerar determinados conceptos que antes se entendían por separado, por ejemplo, las posibles estrategias de detección de una enfermedad. Con estos cambios se pretende poder albergar más información relevante para la generación de modelos económicos de una forma relativamente cómoda, describiendo los casos de estudio con un mayor lujo de detalles.
- **Semánticos:** Se han modificado, mejorado y ampliado las especificaciones semánticas de las relaciones existentes entre las clases de la ontología, además de incluir nuevas propiedades de datos asociadas a cada clase, es decir, los datos concretos que se espera que tenga cada instancia individual de StaDiOS. Con esto los conceptos están mejor interconectados y descritos explícitamente entre sí, de forma que resultará más cómodo tanto consultarlos como instanciarlos.
- **Documentales:** En la medida que ha sido posible, pues no está completamente documentado, también se han ampliado las anotaciones asociadas a cada elemento de la ontología de forma que ahora, tanto en la documentación generada por WIDOCO como en la propia ontología, se podrá consultar el significado y uso de cada elemento como si se estuviera trabajando con la documentación de cualquier librería de programación, por hacer un símil.

En la figura 3.1 se muestra un desglose claro, mediante Protégé, de la jerarquía de clases que componen a StaDiOS.

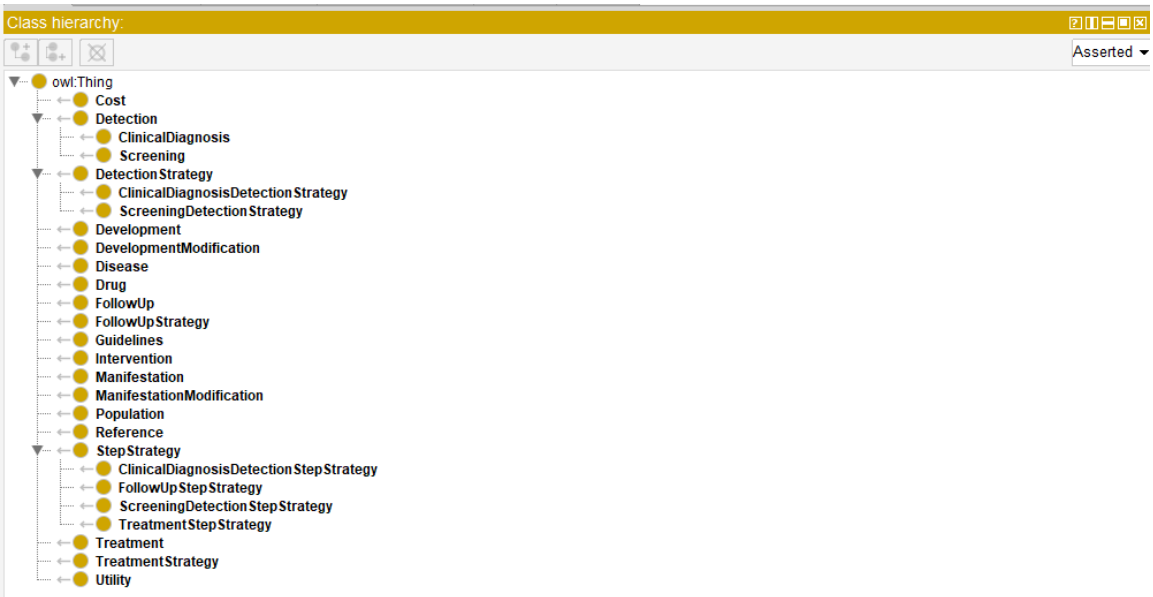


Figura 3.1: Protégé - Jerarquía de clases - StaDiOS

Las clases principales que componen StaDiOS son las siguientes:

Clase	Descripción
Disease	Define de forma estándar las enfermedades y sus posibles características. Actualmente contiene su descripción, identificación y rareza.
Manifestation	Comprende los signos y síntomas de la enfermedad, siendo clave para calcular los resultados cuantitativos del árbol de decisión, tanto por sus consecuencias económicas como sanitarias.
Development	Describe la historia natural de la enfermedad. En general las enfermedades tendrán un único desarrollo, aunque puede verse afectado por las intervenciones.
Intervention	Define las tecnologías o procedimientos utilizados para modificar la historia natural de la enfermedad. Las intervenciones se comparan en el nodo de decisión del árbol de decisión, de forma que admite diferentes estrategias de cribado, diagnóstico, tratamiento y seguimiento.
Detection Strategy	Describe las diferentes estrategias que deben estudiarse para la detección de una determinada enfermedad.
Cost	Almacena los detalles relacionados con el valor económico de los recursos implicados en intervenciones, tratamientos, pruebas de seguimiento, etc. Los costes pueden ser de distintos tipos en función de su frecuencia y momento de aplicación.
Utility	Almacena la calidad de vida relacionada con la salud que puede utilizarse para caracterizar el impacto de una manifestación de la enfermedad en la calidad y esperanza de vida de los pacientes.
Population	Representa la población que sufre la enfermedad y se somete a las intervenciones. Contiene toda la información que puede resultar descriptiva sobre una población bajo estudio.

Tabla 3.1: Clases principales - StaDiOS

Aunque pueda resultar algo confuso en un inicio, lo que debe de quedar claro es que en StaDiOS el objetivo principal es facilitar la comparación entre intervenciones sanitarias. Estas intervenciones son aplicadas a una población que sufre una enfermedad bajo un determinado desarrollo, por lo que se quiere poder averiguar qué intervención aporta más beneficios a nivel económico y de calidad de vida de los pacientes. Se debe destacar que, una vez dentro de las especificaciones únicas de cada intervención, pueden variar los recursos utilizados, por ejemplo, los métodos de detección, las estrategias de tratamiento y seguimiento, los medicamentos utilizados, etc.

Resulta algo complicado mostrar de una forma directa y clara para cualquier lector los cambios realizados, ya que se está trabajando con modelos de datos cuya representación directa puede no ser sencilla. En la figura 3.2 de la página a continuación se verá una representación esquemática de cómo se relacionan todas las clases de StaDiOs entre sí, estando destacados en rojo todos aquellos elementos a nivel estructural que se han añadido para este proyecto.

Cabe mencionar que, aun siendo un modelo relativamente sencillo, su representación gráfica empieza a ser algo compleja de seguir visualmente debido a la cantidad de interconexiones que se pueden establecer entre cada clase. Esto último puede ser algo tan positivo como negativo, aunque el modelo sea difícil de representar o comprender, significa que se pueden analizar las relaciones entre los datos expuestos desde distintos puntos de vista.

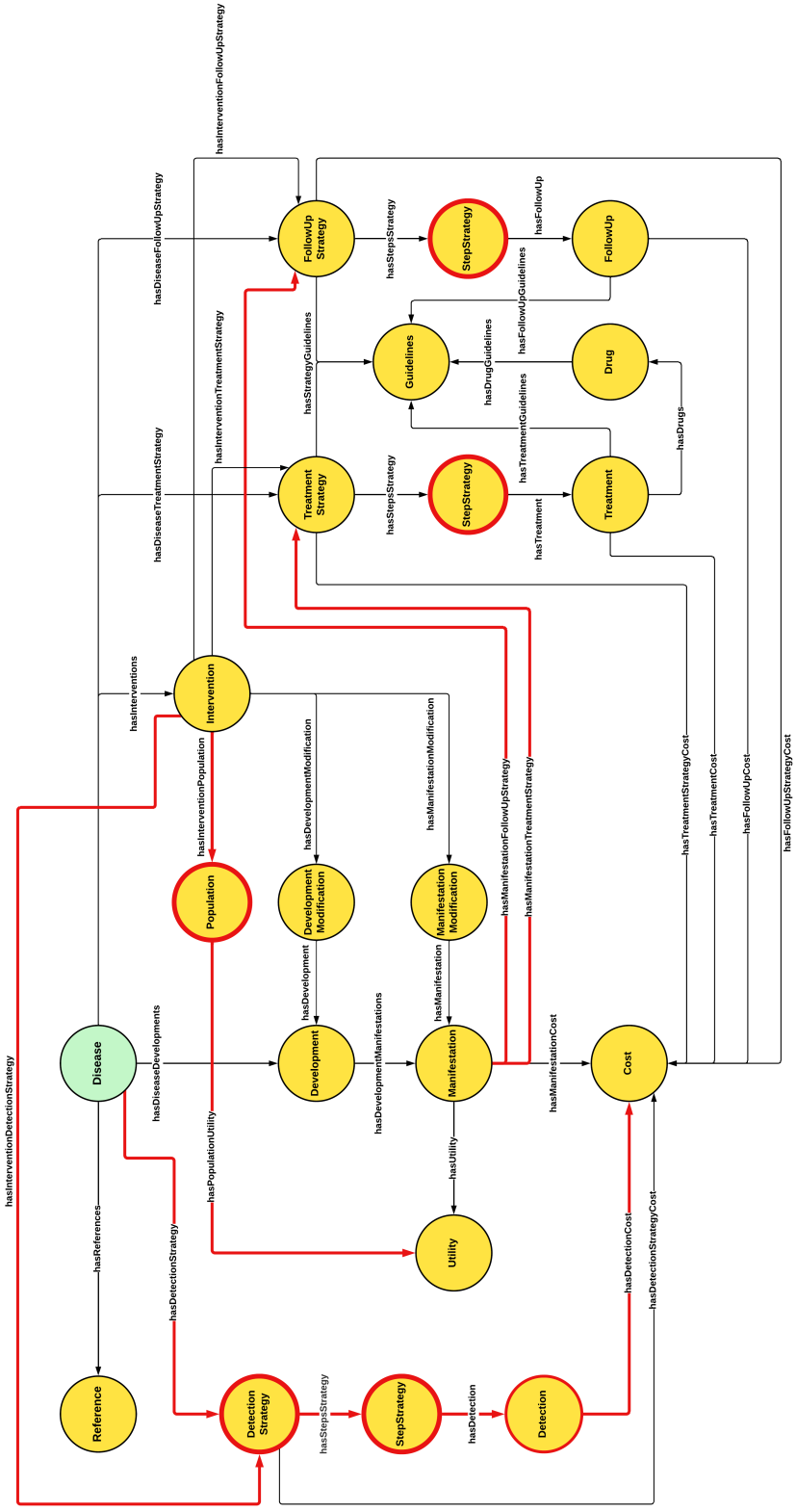


Figura 3.2: Vista esquemática - StaDiOS

4. Aplicación implementada - StaDiOS App

4.1. Descripción del problema

Una vez implementada una ontología que permite almacenar la información de los estudios fuente, se debe ser capaz de reutilizar esa información de una forma eficaz para generar lo que se ha perseguido durante todo el desarrollo previo, la creación de análisis coste-efectividad mediante árboles de decisión [11][19], más concretamente, la comparación entre intervenciones sanitarias aplicadas a determinadas enfermedades bajo unos desarrollos específicos.

Obtenidos estos elementos, se debe simplificar y representar la comparativa entre las intervenciones bajo dos parámetros clave:

- **Costes:** Representan la suma de todos los valores monetarios asociados a las ramas de cada subárbol dentro del árbol de decisión general, perteneciendo estos a cada intervención bajo estudio. Estos costes pueden ser sumados porque previamente son ponderados en base a las probabilidades asociadas a cada rama dentro de los subárboles, donde la suma de probabilidades será igual a 1.0. Los costes manejados siempre pertenecerán a estudios con un contexto geográfico y temporal, por lo que deben ser ajustados en base a la legislación pertinente, por ejemplo, para el estudio de la biotinidasa este proyecto puede actualizar los costes en base al Índice de Precios al Consumidor (IPC).
- **QALY:** Los años de vida ajustados por calidad (Quality Adjusted Life Year) son una unidad de medida en la que se relaciona la cantidad de años de vida del paciente con la calidad de esos años de vida. Un año de vida en perfecto estado de salud se considera equivalente a un QALY. El estado de salud de un individuo puede ser descrito por el par (Q, Y), donde Q es la calidad de los años de vida e Y es la cantidad de años de vida. El valor de utilidad del estado de salud se puede calcular utilizando la siguiente ecuación:

$$U(Q,Y) = V(Q) \cdot Y = N_{\text{QALY}} \tag{4.1.1}$$

  Algunos expertos sostienen que los QALY simplifican en exceso la forma en que se evaluan los riesgos y los resultados mientras que otros, a pesar de sus limitaciones, reconocen su capacidad para asignar recursos de forma equitativa. De forma similar a los costes, los QALY de cada rama del árbol también son ponderados y sumados.

Cada estudio puede variar la estructura del árbol de decisión al estar sujeto a ciertos condicionantes bastante específicos. En el caso del estudio usado de referencia [12], descrito en la sección 3.1, el árbol asociado al cribado tendrá un nivel más de profundidad respecto al árbol asociado a la diagnosis, pues en este caso se ha asumido que la diagnosis siempre acierta en la detección de la enfermedad, mientras que las pruebas asociadas al cribado tienen porcentajes que representan posibles, aunque muy improbables, resultados falsos.

Los árboles de decisión [11] son una técnica de decisión empleada en muchas áreas de conocimiento. Para construir un árbol de decisión, se comienza desde el lado izquierdo con un nodo de decisión y se van agregando a la derecha posibles eventos dentro de cada alternativa a comparar. Luego, se incorporan las probabilidades de ocurrencia, así como los costos y resultados generados por cada evento.

Los nodos y elementos dentro de un árbol pueden representar distintos conceptos o comportamientos, aunque de forma prácticamente universal a cualquier aproximación se pueden distinguir los siguientes elementos.

- **Nodos de decisión:** Por lo general, los nodos de decisión se representan en los modelos de árbol de decisión mediante un cuadrado y se utilizan para evaluar una decisión. Aunque estos nodos pueden ubicarse en diferentes partes del árbol, suelen encontrarse en la parte izquierda y representar el inicio del árbol. A partir de ellos, surgen dos o más ramas excluyentes que representan las rutas a comparar. Las ramas que salen de los nodos de decisión no tienen probabilidades asociadas.
- **Nodos de azar:** Los nodos de azar en los modelos de árbol de decisión se representan generalmente mediante un círculo y se utilizan para indicar las diferentes ramas alternativas en función de un estado previo. Pueden ubicarse en cualquier posición del árbol, excepto al inicio y al final. A diferencia de los nodos de decisión, las ramas que salen de los nodos de azar no son excluyentes y se pueden bifurcar en varias ramas. Estos nodos se van enlazando unos con otros hasta llegar a los nodos finales, estando acompañados de su respectiva probabilidad de ocurrencia. Los nodos de azar definen puntos de incertidumbre en el árbol.
- **Ramas:** Cada rama representa una combinación de eventos desde el nodo de decisión inicial hasta cada uno de los nodos terminales, y estas ramas son mutuamente excluyentes. Cada rama tiene asociada una probabilidad y un coste.
- **Nodo terminales:** Generalmente se representan con un triángulo y se utilizan para resumir los resultados y costos de cada rama del árbol. Estos nodos siempre se colocan en la parte derecha de la representación gráfica del modelo, simbolizando la finalización de una secuencia de eventos.

Para verificar correctamente un árbol de decisión, es necesario cumplir ciertas reglas derivadas de la Teoría de la Probabilidad.

- **Regla de complementariedad:** La suma de las probabilidades de las ramas que tienen como origen el mismo nodo de azar debe ser igual a uno.
- **Regla de probabilidad condicionada:** La probabilidad de que se produzca un determinado evento depende de las probabilidades de la ramas precedentes.
- **Regla de resultado promedio esperado:** El resultado promedio esperado de un suceso es equivalente al sumatorio de los productos de los resultados finales por sus probabilidades.

En la figura 4.1 se puede observar la estructura asociada al árbol de decisión de este estudio.

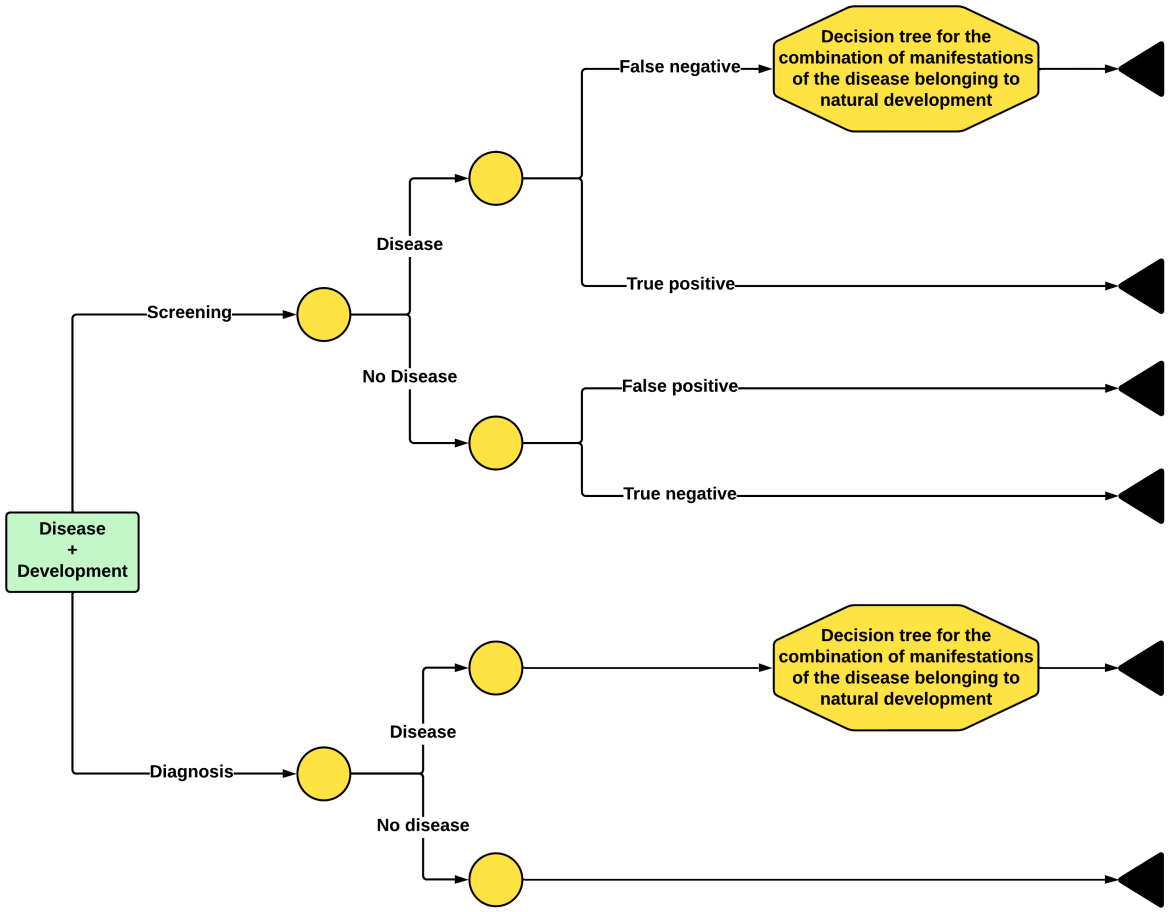


Figura 4.1: Árbol de decisión asociado - Cribado vs Diagnosis

Generados los árboles y los resultados deseados, se debe tener una cosa más en cuenta. Los tipos de evaluación económica más empleados en la bibliografía consultada son los análisis coste-efectividad y coste-utilidad [20], destacando que en España son las modalidades de evaluación económica predilectas [21]. A continuación se utilizará el término ACE para mencionar ambas evaluaciones, las cuales siempre implican la comparación de dos tecnologías sanitarias alternativas, A y B. En la realización de un ACE, normalmente se emplea algún tipo de medida que engloba tanto los costes como la efectividad de las opciones evaluadas para expresar el resultado final de la comparación, es decir, los costes y QALY descritos anteriormente. El cociente o ratio coste-efectividad incremental (Incremental Cost-Effectiveness Ratio), de acrónimo ICER, representa la relación entre la diferencia del costo y la efectividad promedios de dos tecnologías sanitarias. Este trabajo utiliza dicho parámetro para determinar cuál de las intervenciones médicas evaluadas es la mejor, como se muestra en la siguiente ecuación:

$$ICER = \frac{COST_B - COST_A}{QALY_B - QALY_A} \tag{4.1.2}$$

La forma más común de decidir si una nueva intervención es o no eficiente, en cuanto a coste-efectividad se refiere, es a través del establecimiento de umbrales como los mostrados en la figura 4.2. Estos permiten decidir si una intervención se debe o no incorporar a la cartera de servicios de un sistema sanitario, siendo lo habitual que cada país defina los suyos propios.

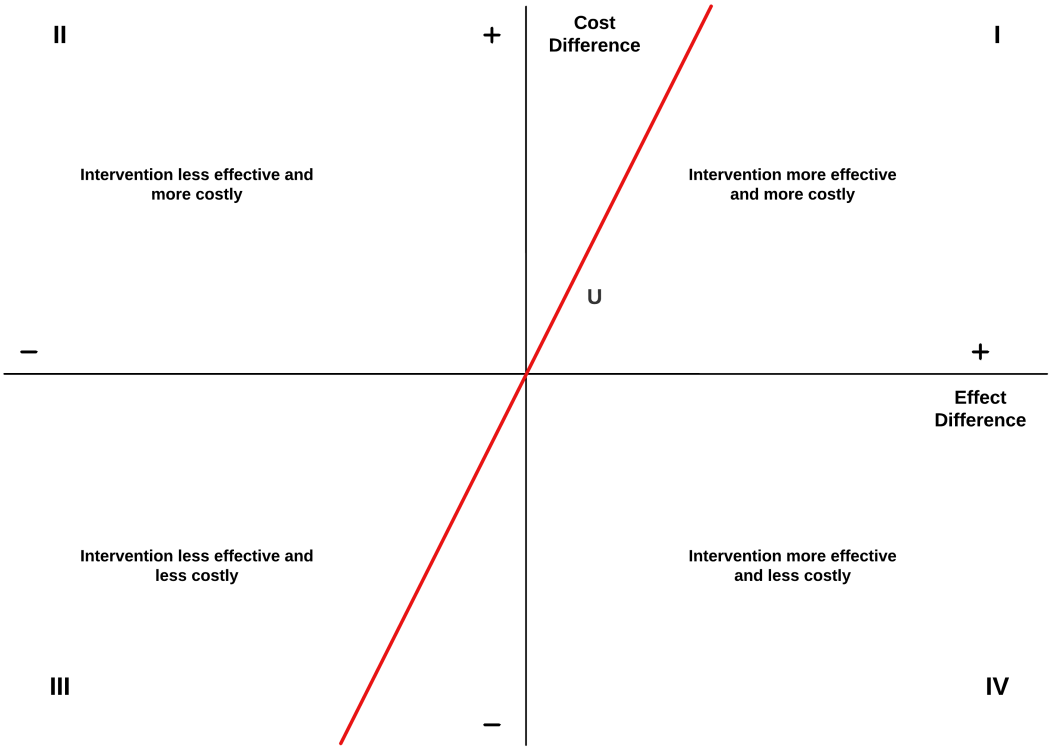


Figura 4.2: Plano coste-efectividad

Como se observa en la figura 4.2, el plano es dividido en cuatro cuadrantes que permitirán interpretar la eficiencia de la nueva intervención en función del cuadrante donde se ubiquen sus coordenadas coste-efectividad. También se ve como la recta U representa el umbral coste-efectividad asumible, dividiendo el plano en las partes aceptables y no aceptables desde el punto de vista coste-efectividad. La función que determina U depende totalmente del problema a tratar y de los parámetros que lo condicionen, la representación mostrada en la imagen es un mero ejemplo.

Por consiguiente a lo anterior, los escenarios que se deben valorar a la hora de comparar dos intervenciones serán los siguientes:

- **Cuadrante I:** La nueva intervención tiene una efectividad mayor que la alternativa de referencia, pero también implica un coste superior.
- **Cuadrante II:** La nueva intervención tiene menor efectividad y mayor coste que la alternativa de referencia.
- **Cuadrante III:** La nueva intervención tiene menor efectividad y menor coste que la alternativa de referencia.
- **Cuadrante IV:** La nueva intervención tiene mayor efectividad y menor coste que la alternativa de referencia.

Mientras que las decisiones en los cuadrantes II Y IV suelen ser bastante obvias, las que se toman en los cuadrantes I y III no son tan sencillas, requiriendo de la definición de determinados elementos o criterios adicionales acordes al estudio en cuestión. En este caso no se explicarán dichos elementos, ya que abarcan cuestiones bastantes amplias sobre el campo de las evaluaciones económicas, cosa que se sale de los objetivos principales del proyecto y añade una complejidad por el momento innecesaria, pero por lo menos debe quedar claro que los casos situados en esos cuadrantes no son un campo de estudio con decisiones obvias.

4.2. Implementación

4.2.1. Herramientas de desarrollo

La creación de la aplicación implementada ha ido fundamentada por las siguientes herramientas:

- **Python:** Lenguaje de programación de alto nivel, multiparadigma y multiplataforma, con una sintaxis clara y legible, lo que lo hace fácil de aprender y usar. Python cuenta con una enorme cantidad de librerías y frameworks que nos ofrecen numerosas funcionalidades y facilitan el desarrollo de aplicaciones, como veremos posteriormente. El lenguaje tiene una amplia comunidad activa de desarrolladores que proporciona recursos y soporte, mejorándolo y ampliándolo continuamente.

- **Visual Studio Code:** Popular editor de código fuente desarrollado por Microsoft. Dispone de una interfaz de usuario moderna y personalizable, ademas de una muy alta compatibilidad con lenguajes de programación, herramientas de depuración, integración con control de versiones e integración con servicios en la nube.

- **Jupyter Notebook:** Entorno de código abierto que permite escribir y ejecutar código de forma interactiva en celdas, combinando código con texto, imágenes y visualizaciones. Jupyter Notebook es una herramienta poderosa y versátil para el análisis de datos, su visualización, la creación de simulaciones reproducibles de los mismos y la colaboración interprofesional para trabajos de desarrollo software. En este proyecto se ha integrado esta herramienta en Visual Studio Code de forma que, antes siquiera de crear la aplicación objetivo, se han podido generar por fases y reutilizar posteriormente los distintos análisis requeridos a partir de StaDiOS, siendo estos recursos elementos que represetan simulaciones estructuradas, fácilmente compartibles y reproducibles.

- **PyCharm:** Entorno de desarrollo integrado (IDE) diseñado para Python por JetBrains. Ofrece características específicas para el lenguage, una interfaz de usuario intuitiva, herramientas de depuración avanzadas, entre otros. Es ampliamente utilizado en la comunidad de desarrolladores de Python y es una herramienta completa y eficiente para el desarrollo de aplicaciones Python.

- **Git:** Sistema de control de versiones distribuido que permite a los desarrolladores rastrear y gestionar los cambios en el código fuente de un proyecto de software. En este desarrollo se ha utilizado git, bajo la plataforma GitHub, para crear y contener un repositorio en el que se ha almacenado todo el trabajo realizado, así como la publicación de documentación, aunque todo esto se verá en la sección de resultados.

- **SPARQLWrapper:** Librería de Python que proporciona una interfaz sencilla para interactuar con endpoints SPARQL, permitiendo la integración de consultas SPARQL en código Python. La librería se encarga de la invocación de las consultas y, opcionalmente, permite convertir el resultado obtenido a formatos más cómodos de manipular.

- **Apache Jena Fuseki:** Servidor de datos RDF y SPARQL desarrollado por Apache Jena, un proyecto de código abierto de la fundación Apache. Este software permite crear y gestionar conjuntos de datos RDF y SPARQL, además de procesar y gestionar consultas en función de las bases de datos que tenga cargadas.

- **Pandas:** Librería de análisis de datos de código abierto para el lenguaje de programación Python que proporciona estructuras y herramientas para la manipulación, análisis, limpieza, transformación y visualización de datos, como son DataFrame y Series. Esta es una herramienta útil en el ámbito académico para llevar a cabo análisis y manipulación de datos en proyectos de investigación en diversas disciplinas.

- **Streamlit:** Framework de Python que permite a los desarrolladores crear rápidamente aplicaciones web interactivas para la visualización de datos y la creación de interfaces de usuario utilizando únicamente código Python. Streamlit se ha vuelto popular en la comunidad de desarrollo de datos debido a su simplicidad y facilidad de uso, lo que permite a los desarrolladores crear aplicaciones web con poco esfuerzo y con un código Python limpio y legible. Este framework proporciona una amplia gama de widgets que facilitan la incorporación de elementos interactivos, como botones, cuadros de texto y gráficos en las aplicaciones web. Se debe de tener en cuenta que Streamlit está diseñado principalmente para modelar aplicaciones web a pequeña y mediana escala, por lo que puede no ser adecuado para aplicaciones web a gran escala o con requerimientos de alta carga de trabajo, además de encontrarse actualmente en una fase de desarrollo activo, por lo que tiene notables limitaciones a la hora de plantear aplicaciones que requieran de una navegación de usuario con opciones algo avanzadas.

- **Wikidata:** Wikidata [22] es una base de datos de conocimiento colaborativa y multilingüe creada por la Fundación Wikimedia. Almacena y gestiona datos estructurados en un formato legible por máquinas, utilizando un modelo de datos basado en RDF. Contiene información sobre entidades del mundo real y permite la vinculación de datos con otros recursos en la web. A diferencia de Wikipedia, que es una enciclopedia en línea, Wikidata se enfoca en la recopilación, organización y vinculación de datos estructurados, con el objetivo de ser una fuente centralizada de datos para proyectos Wikimedia y más allá.

4.2.2. Gestión de los datos

Como toda la información usada en la aplicación para los análisis coste-efectividad provendrá de StaDiOS, u otras ontologías externas, se debe tener mecanismos para consultar dicha información y extraerla de la forma más conveniente. Es aquí donde entra SPARQL, concepto ya explicado en la sección 2.5 y que será divido aquí mediante dos elementos clave. Destacar que la introducción y descripción de estos elementos ya se realizó en la sección anterior, la sección 4.2.1

El primer elemento es SPARQLWrapper. Utilizando esta librería se pueden realizar consultas SPARQL, mediante código en Python, y enviarlas a endpoints SPARQL para obtener los resultados en forma de datos RDF. Aunque se obtiene información RDF, el formato final de la respuesta con dicha información será ajustable a formatos más sencillos de manipular, siendo JSON el formato de respuesta con el que se ha decidido trabajar en este proyecto. Se puede observar cómo se está hablando de mandar peticiones a múltiples endpoints SPARQL sin, aparentemente, tener disponible ningún servidor que exponga los datos de StaDiOS y procese las solicitudes, siendo aquí donde entra el segundo elemento.

El segundo elemento es Apache Jena Fuseki. Para este caso particular, la herramienta ha permitido un despligue local, de forma que cuando se inicia la aplicación ésta se comunica con el gestor SPARQL, desplegado a nivel local, para obtener toda la información necesaria para operar correctamente. Al margen de este software existen muchos otros gestores de datos RDF, por lo que cualquier otro gestor con unas capacidades similares debería servir para conseguir el mismo resultado.

Una vez establecidos todos los mecanismos necesarios para almacenar la información en una ontología y poder consultarla y extraerla de forma cómoda, sólo falta convertir esa información RDF, transformada a su vez en formato JSON, a algún tipo de elemento en Python que facilite su manipulación, siendo aquí donde entra en uso la librería Pandas. En este caso Pandas permitirá generar un análisis coste-efectividad basado en árboles de decisión para que, en vez de visualizar un diagrama en árbol directamente, se obtenga un contenido similar transformado y simplificado a una tabla de datos que refleje la misma información y siga el mismo comportamiento de forma que, por ejemplo, cada rama del árbol será una fila de dicha tabla. Siguiendo una terminología estricta, se puede decir que a partir de una estructura esperada de un árbol de decisión se obtiene una tabla de decisión.

Finalmente, todo lo descrito en esta sección quedará envuelto en un proyecto Python bajo el framework Streamlit. Gracias a este framework y, únicamente mediante código Python, se podrá generar una aplicación funcional con todos los requisitos descritos hasta el momento, obteniéndose un prototipo funcional y razonablemente extensible.

4.2.3. Análisis coste-efectividad

Debe recordarse que lo que se quiere conseguir es generar comparaciones entre intervenciones médicas soportándose en análisis coste-efectividad mediante árboles de decisión, teniendo en cuenta que esas intervenciones son aplicadas a una población que sufre de una enfermedad con un desarrollo concreto. En la figura 4.3 se puede ver el menú principal que se le muestra al usuario al iniciar la aplicación.

Figura 4.3: Parámetros de simulación - StaDiOS App

El usuario debe de seleccionar en cada barra de selección las opciones que quiere para generar la simulación del análisis coste-efectividad, siendo los valores de cada barra obtenidos directamente de StaDiOS mediante consultas SPARQL. Las opciones obligatorias a seleccionar son la enfermedad a estudiar, su desarrollo e identificador de estudio asociado, mientras que la selección de unas estrategias de seguimiento y tratamiento son opcionales. Actualmente hay cargados los parámetros del estudio de la deficiencia profunda de biotinidasa y otros adicionales pertenecientes a un estudio de características similares totalmente inventado, por lo que estos últimos son bastante más incompletos que los primeros. En el momento en que el usuario presione el botón para generar el análisis, siempre que la información cargada en la ontología sea suficiente para generarlo, automáticamente obtendrá distintos resultados.

El primer resultado observable será una tabla interactiva con toda la información que conforma al análisis mediante el árbol de decisión, siendo que cada fila de la tabla representa a cada rama del árbol. El usuario podrá descargarse esta tabla en formato CSV para facilitar cualquier inferencia posterior que desee realizar por su cuenta. Respecto a la tabla generada, al ser bastante extensa para su inclusión de forma cómoda y legible en este documento, se recomienda la consulta del repositorio de GitHub [23] del proyecto, concretamente la sección de coste-efectividad. En esta sección se podrá seguir la simulación completa en Jupyter Notebook usada para obtener dicha tabla, junto con todos los archivos en formato CSV descargados tras la realización de los análisis aquí mencionados.

Además del análisis en bruto que representa la tabla interactiva, también se generará uno de los objetivos principales del proyecto, una tabla comparativa de las intervenciones cargadas en StaDiOS bajo las condiciones que el usuario haya seleccionado, como se puede ver en la figura 4.4. Esta tabla también puede ser descargada en formato CSV para cualquier operación posterior que el usuario quiera realizar sobre la misma. El resultado obtenido se compone de los parámetros clave que fueron descritos en la sección 4.1, aunque con algunos nuevos aspectos a mencionar. La explicación detallada de estos parámetros se realizará en el apartado de los resultados, en la sección 5.1.

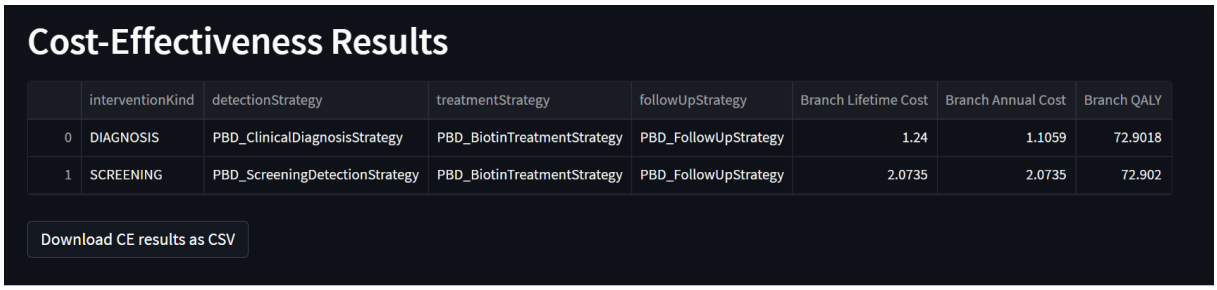


Figura 4.4: Tabla de resultados - StaDiOS App

Una vez obtenidos los resultados de las dos tablas mencionadas, el usuario también podrá agrupar los parámetros clave del análisis realizado, consiguiendo así generar representaciones gráficas de los parámetros principales dentro del árbol de decisión para obtener un análisis más profundo de los mismos, como se ve en la figura 4.5. Estos análisis gráficos pueden ser descargados en formato PNG para cualquier operación posterior que el usuario quiera realizar sobre los mismos. De igual forma que con los valores de la figura 4.4, se mostrarán y explicarán en detalle estas representaciones gráficas generadas en el apartado de los resultados, en la sección 5.1.

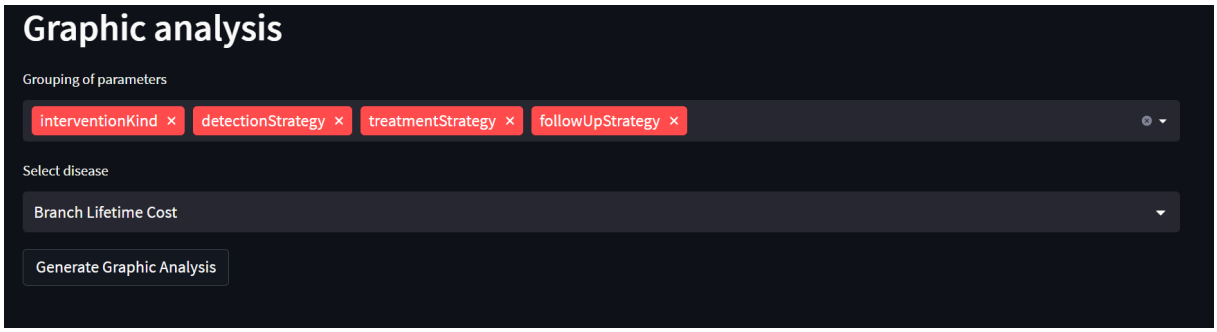
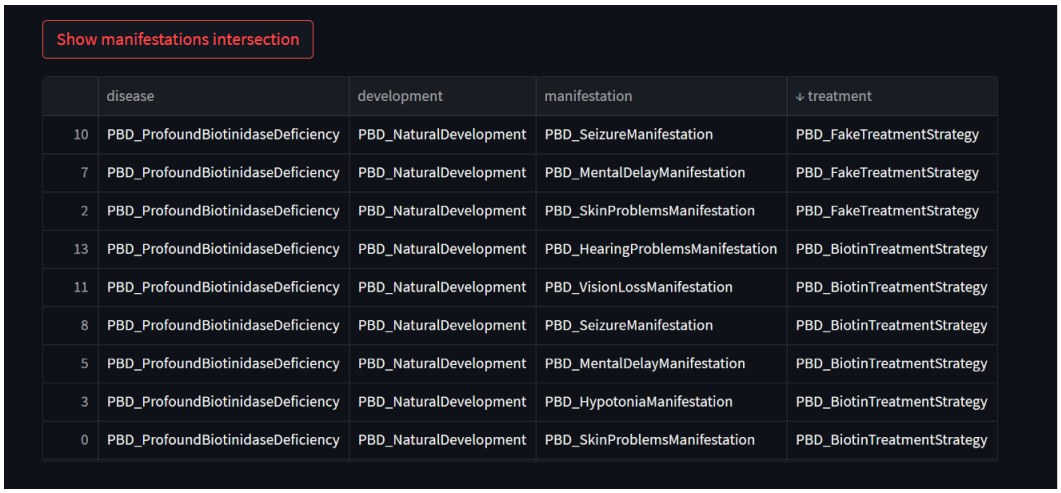


Figura 4.5: Opciones de representación gráfica - StaDiOS App

4.2.4. Inferencia adicional

Una parte muy importante de este proyecto y que no se abordó en los proyectos previos al mismo, es que no se tenía ninguna forma de empezar a obtener inferencia de la ontología a un nivel más general. No se tenían opciones para realizar de forma cómoda la extracción, o deducción, de cierta información que se podría querer conocer sobre los datos actualmente cargados. Cuando se manejan diferentes estudios de enfermedades, podría darse el caso de que las manifestaciones asociadas a estos estudios tengan puntos en común que no se han notado en un primer momento. Actualmente, tras seleccionar una enfermedad y un desarrollo, como se ve en la figura 4.6, se pueden obtener los tratamientos y estrategias de seguimiento que tienen en común las manifestaciones de dicha enfermedad. Esto permite saber qué manifestaciones de todos los estudios cargados tienen esos puntos en común, como demuestra la figura 4.7.

Figura 4.6: Intersección de manifestaciones - StaDiOS App



	disease	development	manifestation	↓ treatment
10	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_SeizureManifestation	PBD_FakeTreatmentStrategy
7	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_MentalDelayManifestation	PBD_FakeTreatmentStrategy
2	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_SkinProblemsManifestation	PBD_FakeTreatmentStrategy
13	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_HearingProblemsManifestation	PBD_BiotinTreatmentStrategy
11	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_VisionLossManifestation	PBD_BiotinTreatmentStrategy
8	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_SeizureManifestation	PBD_BiotinTreatmentStrategy
5	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_MentalDelayManifestation	PBD_BiotinTreatmentStrategy
3	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_HypotoniaManifestation	PBD_BiotinTreatmentStrategy
0	PBD_ProfoundBiotinidaseDeficiency	PBD_NaturalDevelopment	PBD_SkinProblemsManifestation	PBD_BiotinTreatmentStrategy

Figura 4.7: Tratamientos en común de manifestaciones - StaDiOS App

SPARQL permite consultar múltiples puntos finales simultáneamente, de forma que se puede crear nueva inferencia a partir de distintas bases de datos siempre y cuando se use una sintaxis equivalente o, por lo menos, compatible. En este trabajo se ha elegido Wikidata para complementar el conocimiento cargado en StaDiOS y ejemplificar lo que se podrá extrapolar a muchas otras bases de datos RDF y derivados.

Lo que se ha conseguido aquí, mediante el conocimiento conjunto proporcionado por StaDiOS y Wikidata, es detectar cuáles de los parámetros de cada uno de los estudios actuales pertenecen a países de la Unión Europea, siendo esto mostrado en la figura 4.9. Actualmente se pueden ver los parámetros europeos dentro de un estudio concreto, en base a su identificador único, u obtenerlos todos a la vez mediante el menú de la figura 4.8.

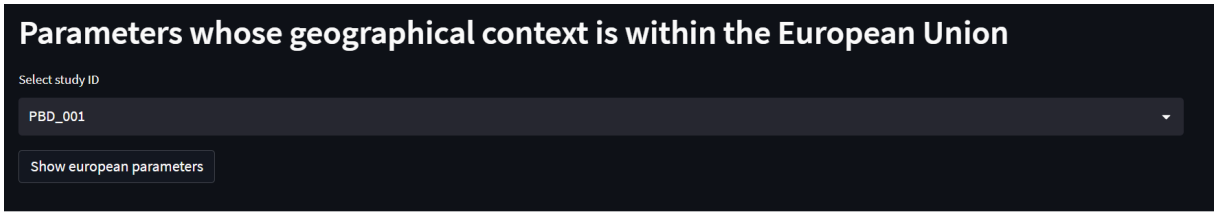
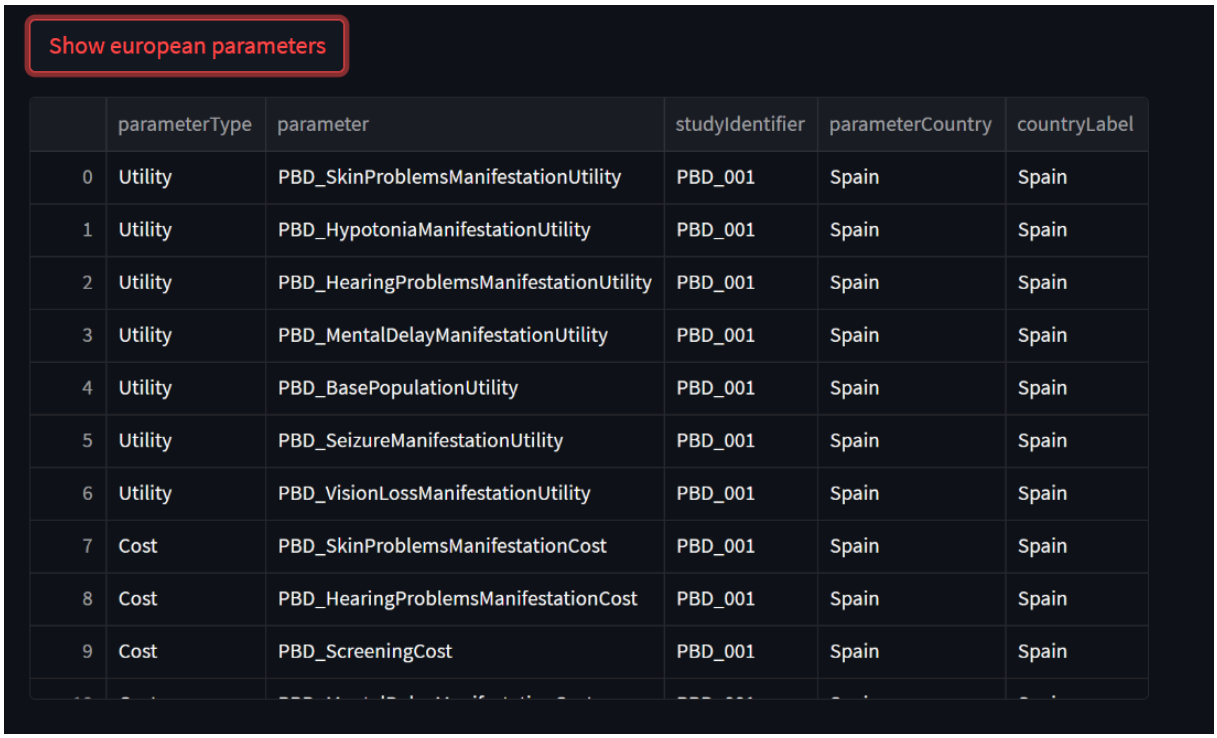


Figura 4.8: Selección de parámetros europeos - StaDiOS App

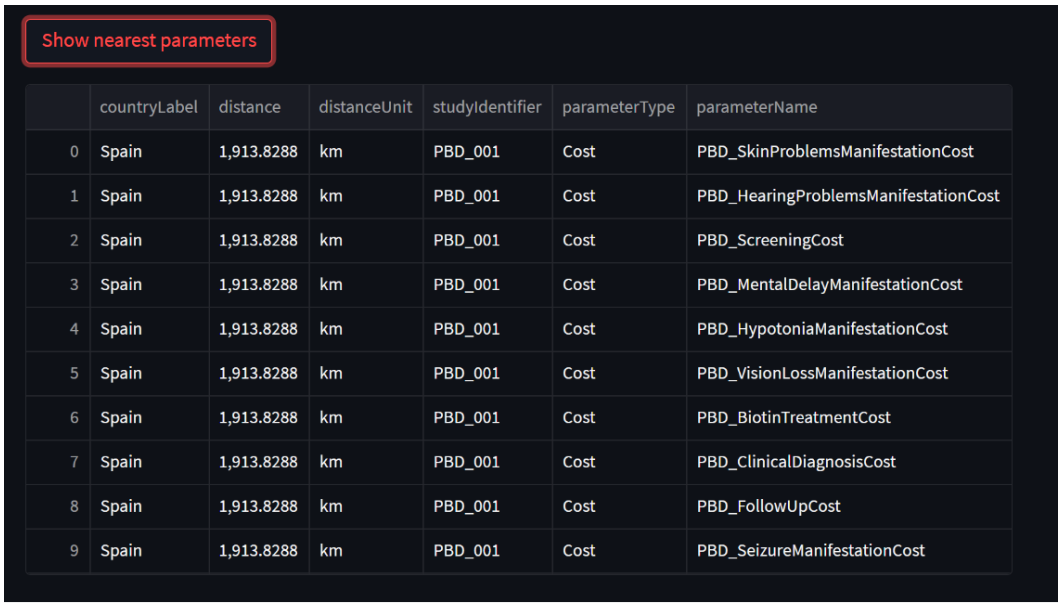


	parameterType	parameter	studyIdentifier	parameterCountry	countryLabel
0	Utility	PBD_SkinProblemsManifestationUtility	PBD_001	Spain	Spain
1	Utility	PBD_HypotoniaManifestationUtility	PBD_001	Spain	Spain
2	Utility	PBD_HearingProblemsManifestationUtility	PBD_001	Spain	Spain
3	Utility	PBD_MentalDelayManifestationUtility	PBD_001	Spain	Spain
4	Utility	PBD_BasePopulationUtility	PBD_001	Spain	Spain
5	Utility	PBD_SeizureManifestationUtility	PBD_001	Spain	Spain
6	Utility	PBD_VisionLossManifestationUtility	PBD_001	Spain	Spain
7	Cost	PBD_SkinProblemsManifestationCost	PBD_001	Spain	Spain
8	Cost	PBD_HearingProblemsManifestationCost	PBD_001	Spain	Spain
9	Cost	PBD_ScreeningCost	PBD_001	Spain	Spain

Figura 4.9: Resultado de parámetros europeos - StaDiOS App

Además de contener los países europeos, en Wikidata se pueden encontrar sus coordenadas, de forma que se han conseguido obtener los parámetros de un tipo específico, usando un identificador de estudio y país que sirven como punto de origen mediante el menú de la figura 4.10, que queden más cerca por distancia geográfica. El resultado observable en la figura 4.11 usa de referencia el estudio secundario inventado para este proyecto, de forma que el resultado apunta a los parámetros del estudio heredado de RaDiOS [12]. Actualmente no tiene un uso directo, pero esto podrá servir para, por ejemplo, complementar información faltante de estudios cargados en la ontología con parámetros que se consideren de mejor calidad o afinidad por distintos criterios, en este caso, por distancia geográfica al estudio y país de origen. Los parámetros que se pueden analizar por el momento son los costes, utilidades y población.

Figura 4.10: Selección de cercanía geográfica - StaDiOS App



	countryLabel	distance	distanceUnit	studyIdentifier	parameterType	parameterName
0	Spain	1,913.8288	km	PBD_001	Cost	PBD_SkinProblemsManifestationCost
1	Spain	1,913.8288	km	PBD_001	Cost	PBD_HearingProblemsManifestationCost
2	Spain	1,913.8288	km	PBD_001	Cost	PBD_ScreeningCost
3	Spain	1,913.8288	km	PBD_001	Cost	PBD_MentalDelayManifestationCost
4	Spain	1,913.8288	km	PBD_001	Cost	PBD_HypotoniaManifestationCost
5	Spain	1,913.8288	km	PBD_001	Cost	PBD_VisionLossManifestationCost
6	Spain	1,913.8288	km	PBD_001	Cost	PBD_BiotinTreatmentCost
7	Spain	1,913.8288	km	PBD_001	Cost	PBD_ClinicalDiagnosisCost
8	Spain	1,913.8288	km	PBD_001	Cost	PBD_FollowUpCost
9	Spain	1,913.8288	km	PBD_001	Cost	PBD_SeizureManifestationCost

Figura 4.11: Resultado de cercanía geográfica - StaDiOS App

4.2.5. Estructura de la aplicación

Los paquetes fundamentales que componen a StaDiOS-App son los siguientes:

- **Main:** Contiene la definición e implementación del programa principal. Desde aquí se inicia la interfaz gráfica mediante el framework Streamlit, mostrando la ventana principal de la aplicación, siendo esta el menú para la generación de análisis coste-efectividad explicado en la sección 4.2.3. También se importan todos los elementos del proyecto que permiten su correcto funcionamiento.
- **Pages:** Permite la definición de las distintas ventanas secundarias de la aplicación, implementadas bajo el framework Streamlit. Actualmente contiene el código para la creación y el control de la actividad de la ventana con las opciones de inferencia adicional de StaDiOS, explicada en la sección 4.2.4.
- **Analyzer:** Define las clases que permiten la interacción con StaDiOS mediante SPARQL y la generación de distintos tipos de modelos de evaluación económica. Actualmente sólo genera los análisis de tipo coste-efectividad mostrados en este proyecto, aunque está enfocado a su ampliación.
- **Sparql:** Define la clase que permite la generación de inferencia adicional a partir de StaDiOS y otras fuentes externas mediante SPARQL, como Wikidata. A diferencia del paquete Analyzer, centrado en definir métodos de análisis económico, este paquete va enfocado para la definición de todos los comportamientos que permitan obtener un análisis adicional de la información disponible en StaDiOS.
- **Data:** Este paquete no contiene código. Aquí se definen las distintas consultas SPARQL que deberán ser llamadas e integradas en el código Python de los elementos que componen la aplicación.

En la figura 4.12 de la página a continuación se representa y explica, de forma gráfica y simplificada, la estructura e interacción existente entre los distintos paquetes y herramientas principales que componen a la aplicación desarrollada. Aunque no se especifique de forma explícita, todas las herramientas descritas en la sección 4.2.1 son integradas en los diferentes elementos expuestos.

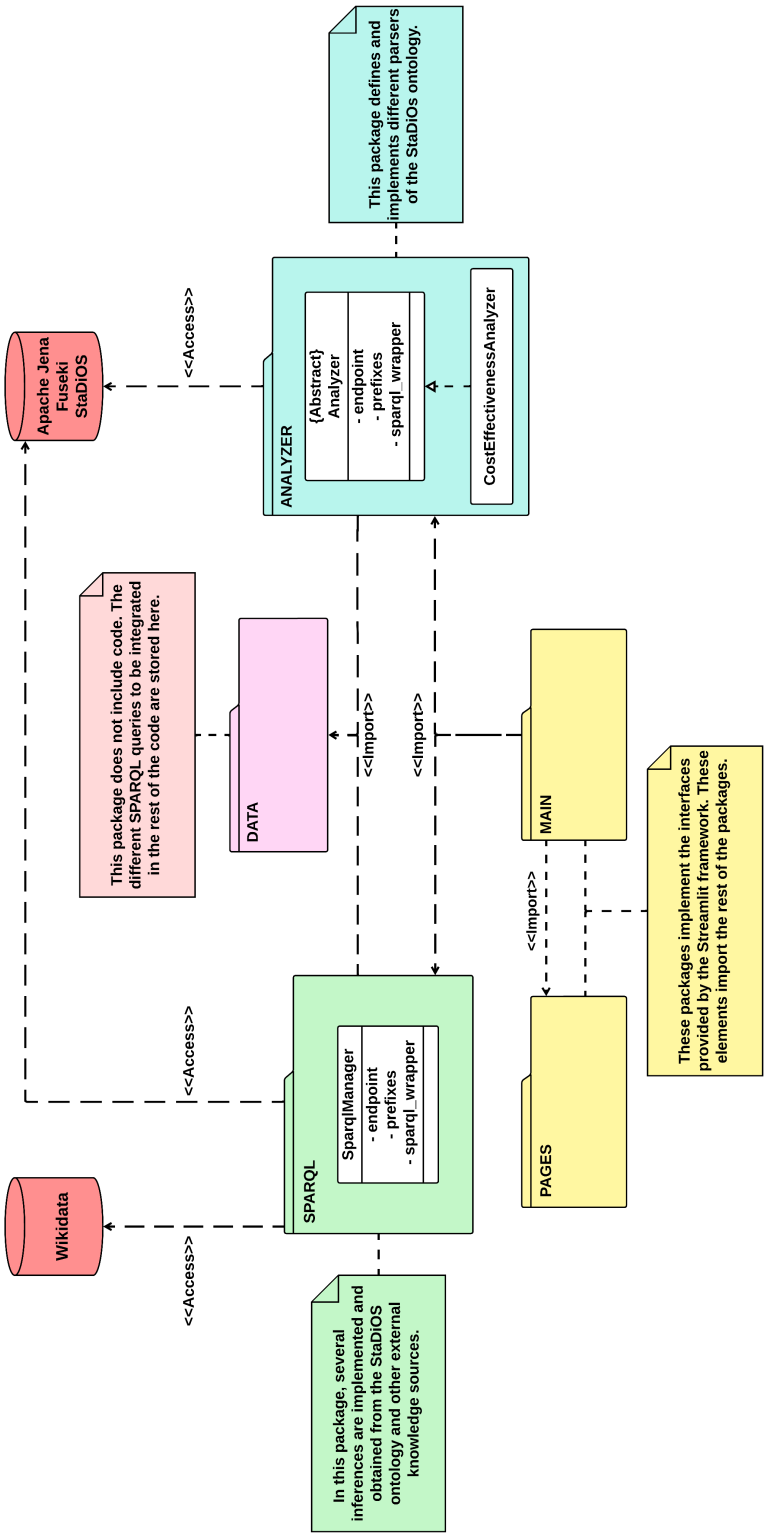


Figura 4.12: Estructura de StaDiOS App - Diagrama UML

5. Validación de caso de estudio

Estrictamente hablando, todo lo implementado es un resultado de este TFG. Ya mostrado y explicado todo lo anterior, ahora sólo queda verificar que los resultados tienen sentido y proporcionalidad respecto a los resultados de los proyectos que lo preceden.

5.1. Análisis coste-efectividad

interventionKind	Cost	QALY
DIAGNOSIS	1.32	72.9019
SCREENING	1.76	72.9020

Tabla 5.1: Resultados originales - RaDiOS

interventionKind	Cost	QALY
DIAGNOSIS	1.05	72.9018
SCREENING	1.76	72.9019

Tabla 5.2: Resultados originales - StaDiOS

En la tabla 5.1 se observa el resultado de la comparación generada por RaDiOS entre las intervenciones del cribado neonatal frente a la diagnosis para el estudio de la deficiencia profunda de biotinidasa. Se debe mencionar que estos resultados son respecto a los costes del momento en que se hizo su estudio de origen, es decir, el año 2013.

En la tabla 5.2 se observa el resultado de la comparación generada por StaDiOS para el mismo supuesto con los costes ajustados al año de origen también. Se puede ver que el análisis parece ser similar, excepto por el resultado obtenido para la diagnosis, pero esto tiene una explicación para el modelo desarrollado. A pesar de que inicialmente se orientó la estructura del árbol de decisión como se muestra en la figura 4.1, de cara a la finalización del proyecto, se detectó en archivos ajenos a los antecedentes propocionados inicialmente que el árbol de decisión asignado a la diagnosis tenía un subnivel de decisión más que no se había comentado en un principio, sin embargo, no se consideró relevante para los resultados que se tenían hasta el momento, de forma que se ignoró. Otro aspecto que afecta ligeramente al resultado es que, para obtener la tabla que simplifica el diagrama del árbol de decisión, se aplica una normalización a las probabilidades asignadas a las manifestaciones de la enfermedad en cuestión, de forma que se pierde cierto nivel de información o, mejor dicho, se ponderan de forma distinta los factores de decisión.

Aunque estos dos factores generan la discrepancia entre los resultados, esto no ocurre de forma generalizada, es decir, también depende de las ponderaciones concretas del caso en cuestión puesto que en el cribado, a pesar de sufrir la misma normalización, no denota ninguna pérdida significativa de información. Esto indica que según se configuren las ponderaciones de las ramas de los árboles se notará más, o menos, los efectos de la normalización de las probabilidades en los resultados de la simulación generada. Es importante decir que en la tabla de decisión, el usuario también tiene las probabilidades sin normalizar, de forma que puede ponderar los datos bajo distintos criterios si así lo desea.

Dicho todo lo anterior, se verifica que el análisis obtenido tiene una concordancia y proporcionalidad respecto a los antecendentes del proyecto, validando el procesamiento de la información. Se considera positivo tener identificados y saber de forma clara los puntos que hacen que se difiera de los estudios padre, pues esto verifica que el proceso de análisis se realiza de una forma esquemática y controlada.

interventionKind	Branch Lifetime Cost	Branch Annual Cost	Branch QALY
DIAGNOSIS	1.24	1.10	72.9018
SCREENING	2.07	2.072	72.9019

Tabla 5.3: Resultados actuales - StaDiOS

En la tabla 5.3 se muestra el resultado de la misma comparación realizada en las tablas 5.1 y 5.2, únicamente variando en que los costes se han actualizado a la fecha de realización de este proyecto. Como se puede ver, los resultados dividen los costes en dos términos que pueden resultar algo confusos a primera vista, pero que dotan a la ontología de una mayor flexibilidad a la hora de poder definirlos.

- **Annual Cost:** Representa los costes anuales de los parámetros que necesitan ser tratados para obtener un valor estimado de su coste real aplicado a la esperanza de vida de la población en cuestión.
- **Lifetime Cost:** Representa los costes que aglomeran los costes anuales de cada parámetro y cualquier otro elemento externo. Es una forma de simplificar la definición de los costes si en un estudio ya se tiene esa información disponible.

Aunque en un primer momento se podría pensar que los resultados de ambos costes deberían dar lo mismo, esto no es siempre así. Hay veces que en los estudios médicos se tiene distintos grados de desglose en la definición de los costes a usar, siendo que para este caso particular no toda la información asociada a los costes de las manifestaciones de la deficiencia profunda de biotinidasa tenía un coste anual aproximado, sino que se trabajaba directamente con su aproximación total (incluyendo otros parámetros externos al coste) ajustada a la esperanza de vida de la población estudiada. Esto es lo que genera el desajuste entre los dos tipos de costes, puesto que algunos parámetros tienen su aproximación de coste anual a cero. De nuevo, la ponderación en cada rama de los árboles también determinará una mayor o menor disonancia entre unos resultados y otros.

Para aproximarse lo más posible a los estudios de origen del proyecto, se guiarán las conclusiones utilizando principalmente el parámetro ”Lifetime Cost”, obteniéndose el siguiente resultado.

$$ICER = \frac{2{,}07 - 1{,}24}{72{,}9019 - 72{,}9018} = 8300\ €/\text{QALY} \tag{5.1.1}$$

A partir del parámetro ICER se deduce que se necesitan 8.300€ adicionales para obtener una ganancia de 1 QALY al cambiar de la intervención de diagnosis al cribado neonatal. En España, acorde a la bibliografía [24], toda intervención cuyo ICER sea inferior a 20000 €/AVAC se considera coste-efectiva.

Parece ser que el cribado neonatal es menos rentable en cuanto a lo que el costo-efectividad se refiere, aunque se puede considerar una intervención coste-efectiva. Se debe remarcar que estas comparativas no resuelven el problema de una forma sencilla, es decir, pueden haber muchos más factores que permitan decidir la implementación de una intervención u otra. Para ilustrar esto mejor, mediante la figura 5.1, se adjunta el análisis gráfico obtenido para el parámetro "Lifetime Cost" por StaDiOS-App entre las dos intervenciones.

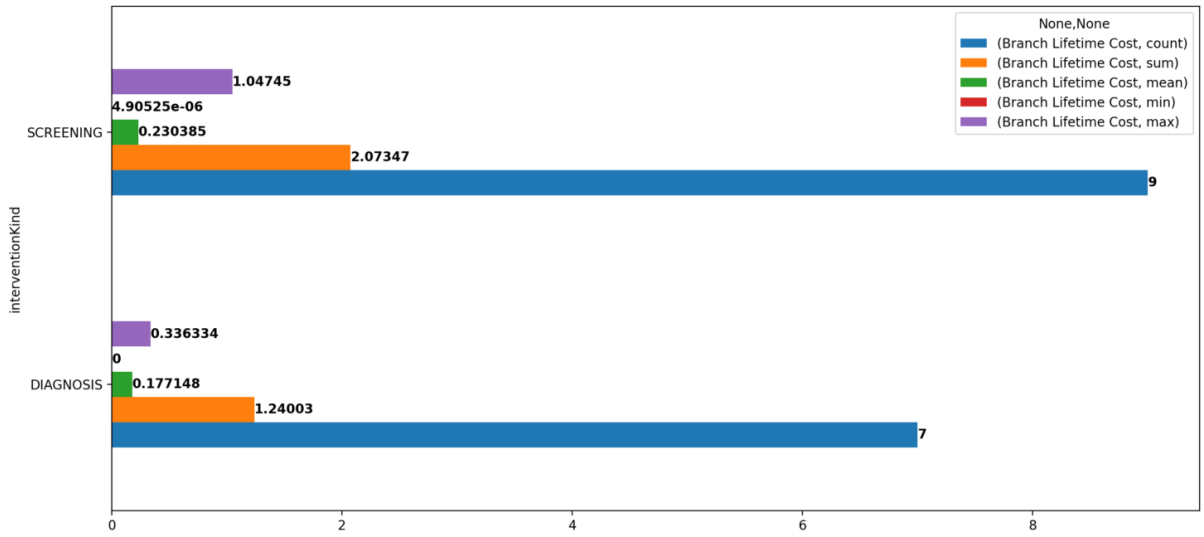


Figura 5.1: Representación gráfica - Cribado vs Diagnosis

La diagnosis parece seguir siendo la clara ganadora frente al cribado en cuanto a costes se refiere, ya que se observan menores valores máximos, mínimos y de media. También se verifica que la suma de los costes es igual al resultado de la tabla 5.3 y que el número de ramas implicadas tiene sentido de cara al árbol de decisión esperado. A pesar de todo esto, se debe insistir en que aquí no acaba el análisis, simplemente es un factor más a tener en cuenta. El cribado neonatal es una intervención coste-efectiva en base a los resultados anteriores que permite, aunque aparentemente de forma más cara, evitar con una mayor eficacia que los pacientes sufran los efectos más adversos de la enfermedad. La diagnosis directa simplemente ataca el problema una vez se presenta. Una de las razones fundamentales por las que la diagnosis parece ganar aquí es porque se está trabajando con una enfermedad muy rara, hay tan poca incidencia de forma natural que económicamente tiene sentido el no querer buscarla directamente pero, de nuevo, son decisiones complejas que deben ser tomadas por profesionales de las entidades médicas.

No se debe creer que las posibilidades del análisis generado hasta el momento se limitan a lo mostrado. Como bien se explicó en la sección 4.2.3, el usuario tiene que seleccionar unas opciones obligatorias para el análisis, mientras que otras son opcionales, concretamente, las opciones del tratamiento y seguimiento a seguir. Si no se selecciona alguna de estas dos, o ambas, se podrá obtener un análisis de los costes derivados únicamente de lo seleccionado, excluyendo el resto. Mediante este simple desglose de los factores del análisis se pueden deducir los elementos que más afectan a la toma de decisiones. En este caso la estrategia de seguimiento parece ser lo que más encarece los costes, mientras que la diferencia económica entre ambas intervenciones parece mantenerse con una proporción estable.

interventionKind	Branch Lifetime Cost	Branch Annual Cost	Branch QALY
DIAGNOSIS	0.262	0.127	72.9018
SCREENING	1.095	1.095	72.9019

Tabla 5.4: Resultados actuales - Intervenciones

interventionKind	Branch Lifetime Cost	Branch Annual Cost	Branch QALY
DIAGNOSIS	0.356	0.222	72.9018
SCREENING	1.190	1.190	72.9019

Tabla 5.5: Resultados actuales - Intervenciones + Tratamiento

interventionKind	Branch Lifetime Cost	Branch Annual Cost	Branch QALY
DIAGNOSIS	1.145	1.011	72.9018
SCREENING	1.978	1.978	72.9019

Tabla 5.6: Resultados actuales - Intervenciones + Seguimiento

5.2. Inferencia adicional

Como se ha mostrado en las figuras desde la 4.6 a la 4.11 dentro de la sección 4.2.4, ahora se puede obtener y generar nueva información útil a partir de la ontología y otras fuentes de datos externas. No hay demasiado que comentar aquí de cara a los resultados obtenidos pues esto, aunque muy prometedor, no deja se ser una primera aproximación para validar el potencial que tiene por delante esta herramienta y otros futuros proyectos bajo el uso de SPARQL y bases de datos RDF.

5.3. Repositorio de trabajo

Todo lo desarrollado en este proyecto ha conformado un repositorio de trabajo [23] que queda abierto para el uso, mejora y exploración de todo lo creado. A partir de este repositorio cualquiera que esté interesado puede acceder a la ontología, verificar los análisis coste-efectividad realizados en Jupyter Notebook, y descargar la aplicación gráfica implementada e instalar de forma cómoda sus dependencias mediante Python.

Otro aspecto fundamental de este apartado es que se ha conseguido generar un punto de origen para la actualización y revisión continua de la ontología; ahora cada vez que se modifica y actualiza StaDiOS en el repositorio se generará automáticamente, mediante GitHub Actions, GitHub Pages y WIDOCO, la documentación asociada a la ontología junto con revisiones de las deficiencias sintácticas de la misma. Gracias a todo esto, ahora el proyecto ontológico está vivo y disponible para la consulta de cualquier usuario, además de que la documentación generada facilita el entendimiento de la información almacenada en el mismo.

- **Repositorio de GitHub:** `https://github.com/alu0101028491/TFG_StaDiOS`
- **Documentación - GitHub Pages:** `https://alu0101028491.github.io/TFG_StaDiOS/`

6. Conclusiones y trabajos futuros

6.1. Conclusiones

La toma de decisiones por parte de las entidades sanitarias para la inclusión de intervenciones médicas requiere de una gran cantidad de información y evaluaciones de distintos ámbitos. Poder facilitar de alguna forma la semi-automatización para la generación de modelos coste-efectividad mediante árboles de decisión para los equipos en cuestión parece una línea de investigación que, aunque bastante compleja y laboriosa, puede tener un desenlace realmente útil.

- La ontología desarrollada ya puede albergar la información necesaria para generar árboles de decisión básicos para cualquier enfermedad, además de dar cierto nivel de maniobra a la hora de diseccionar los parámetros a tener en cuenta en cada análisis.
- La aplicación implementada ofrece una forma sencilla de generar los modelos en cuestión sin que el usuario deba ser consciente de todo lo que pasa realmente, además de poder descargarse todos los análisis generados para cualquier operación externa que desee realizar.
- Se ha logrado mostrar, mediante algunas herramientas de la Web Semántica, que el proyecto no está limitado a la información cargada en StaDiOS. Siempre que se usen tipos de datos y un lenguaje estándar para el campo de estudio en cuestión, se podrán valorar infinitas fuentes de datos oficiales para obtener nueva inferencia, como los portales de datos abiertos de distintos gobiernos, aunque realmente estos últimos podrían llegar a ser también las principales fuentes de recolección de información para StaDiOS.

Uno de los aspectos más positivos del trabajo realizado es que abre la puerta a múltiples aplicaciones futuras que se escapan del ámbito temporal de un Trabajo de Fin de Grado, siendo algunas de las más interesantes mencionadas en la sección a continuación, la sección 6.2.

6.2. Trabajos futuros

- StaDiOS no es un elemento estático, es un modelo de conocimiento vivo. Sus conceptos, datos, vocabulario y la relación entre los mismos están abiertos a su ampliación y mejora para poder definir nuevos supuestos.
- En StaDiOS se traduce la información mínima requerida de un estudio médico al modelo de conocimiento diseñado, por lo que mediante futuras iteraciones con más estudios de referencia se irán ampliando y mostrando sus deficiencias conceptuales. Mediante StaDiOS se debe aprender a jugar a encontrar esa cantidad mínima de información que permite describir el problema a solucionar de forma óptima.
- Para la ampliación y mejora de la aplicación, se abre un enorme abanico de posibilidades a la hora de enfocar cómo se van a plantear las opciones de personalización para la generación de los árboles de decisión de cara a estudios que, con antelación, ya se sabe que serán completamente distintos y cuyos parámetros de referencia necesitarán un procesamiento igual de distinto. El desarrollo de esta aplicación, o cualquiera derivada de StaDiOS, requerirá un buen diseño a la hora de afrontar la definición por parte de los usuarios de distintos parámetros de análisis para poder ajustarse al gran número y tipos de estudios a los que se enfrentará.
- La información que actualmente se puede visualizar sobre los elementos cargados en StaDiOS y los obtenidos mediante inferencia externa podría ser útil y aplicable directamente a los análisis. Se deben estudiar formas de intercambiar parámetros entre distintos estudios para mejorar o complementar la información de los análisis, además de posibilitar criterios de evaluación y selección para dichos parámetros.
- StaDiOS requiere del uso de herramientas externas para la carga manual de información. Uno de los puntos clave para empezar a ver el verdadero potencial del proyecto es la integración de técnicas y herramientas, sobretodo de inteligencia artificial, para la extracción y carga automática de información. A fecha de la realización de este trabajo, muchos portales abiertos están habilitando endpoints SPARQL para el acceso a toda la información de los mismos, por lo que se considera que sería algo muy interesante el que una inteligencia artificial analizara y entendiera los contenidos bajo el estándar RDF y sus derivados que ahí se encuentran, extrayendo la información necesaria y cargándola directamente en la ontología.
- StaDiOS puede ser desplegado en cualquier servidor SPARQL y ser puesto a disposición de todos los interesados para su desarrollo vía online [25], facilitando todavía más la idea anterior de la carga automática de la información. StaDiOS-App también debería ser capaz de seguir su camino. Una vez desplegada la aplicación en algún servicio, ésta debería ser capaz de extraer la información de StaDiOS vía web, permitiendo trabajar de forma similar a como lo hace ahora.
- Actualmente, cuando no hay información suficiente para generar un análisis coste-efectividad, se muestra un mensaje al usuario y se le dirige a la documentación con toda la información asociada. Sería interesante analizar los parámetros que no están bien definidos en StaDiOS en el momento de la consulta y explicar mejor al usuario de qué información se carece.

7. Conclusions and future work

7.1. Conclusions

Decision-making by healthcare entities for the inclusion of medical interventions requires a large amount of information and evaluations from different fields. To be able to facilitate in some way the semi-automation for the generation of cost-effectiveness models using decision trees for the teams in question seems to us to be a line of research that, although quite complex and laborious, can have a really useful outcome.

- The ontology developed can already hold the information necessary to generate basic decision trees for any disease, in addition to providing a certain level of maneuver when it comes to dissecting the parameters to be taken into account in each analysis.
- The implemented application offers a simple way to generate the models in question without the user having to be aware of everything that is happening, in addition to being able to download all the generated analyses for any external operation it may wish to perform.
- It´s been possible to show, using some Semantic Web tools, that the project is not limited to the information uploaded to StaDiOS. As long as standard data types and language are used for the field of study in question, an infinite number of official data sources can be used to obtain new inferences, such as the open data portals of different governments, although, in reality, the latter could also become the main sources of information collection for StaDiOS.

One of the most positive aspects of the work done is that it opens the door to multiple future applications beyond the time scope of a Final Degree Project, some of the most interesting of which are mentioned in the section below, section 7.2.

7.2. Future work

- StaDiOS is not a static element. It is a living knowledge model. Its concepts, data, vocabulary, and the relationship between them are open to expansion and improvement to define new assumptions.
- In StaDiOS, the minimum information required from a medical study is translated into the designed knowledge model, so that through future iterations with more reference studies, they will expand and show their conceptual deficiencies. Through StaDiOS you must learn to play to find that minimum amount of information that allows you to describe the problem to be solved optimally.
- For the extension and improvement of the application, an enormous range of possibilities opens up when it comes to approaching the customization options for the generation of decision trees for studies that, in advance, are known to be completely different and whose reference parameters will require equally different processing. The development of this application, or any derived from StaDiOS, will require a good design when facing the definition by the users of different analysis parameters to be able to adjust to the large number and types of studies that will be faced.
- The information it can currently be displayed, on the elements loaded into StaDiOS and those obtained by external inference could be useful and directly applicable to the analyses. Ways of exchanging parameters between different studies should be explored to improve or complement the information in the analyses. As well as to enable evaluation and selection criteria for these parameters.
- StaDiOS requires the use of external tools for manual data loading. One of the key points to begin to see the project's potential is the integration of techniques and tools, especially artificial intelligence, for the extraction and automatic information loading. As of the date of this work, many open portals are enabling SPARQL endpoints for access to all the information in them, so it's considered that it would be very interesting for artificial intelligence to analyse and understand the contents under the RDF standard and its derivatives that are found there, extracting the necessary information and loading it directly into the ontology.
- StaDiOS can be deployed on any SPARQL server and made available to all interested parties for development via online [25], further facilitating the above idea of automatic loading of the information. StaDiOS-App should also be able to follow suit. Once the application is deployed in some service, it should be able to extract the data from StaDiOS via the web, allowing it to work similarly as it does now.
- Currently, when there is insufficient information to generate a cost-effectiveness analysis, a message is displayed to the user and it's directed to the documentation with all the associated data. It would be interesting to analyse the parameters not well defined in StaDiOS at the time of the query and better explain to the user what information is lacking.

8. Presupuesto

La obtención y el cálculo del presupuesto asociado a este proyecto se basa principalmente en una estimación del coste total por hora, fundamentando esta estimación en el número de horas requeridas para la asignatura en cuestión y en los gastos derivados de la actividad profesional a realizar.

- El tiempo estimado para la realización de este proyecto, también acorde a la guía docente de la asignatura en la fecha de realización del mismo, consta de un total de 300 horas y 12 créditos ECTS, siendo cada crédito ECTS equivalente a 25 horas de trabajo.
- La estimación realizada incluye el coste de la mano de obra, amortización de equipo para el trabajo realizado, fungibles, suministros, impuestos, etc. Hay que tener en cuenta que todas las herramientas usadas en este proyecto son de uso gratuito, de forma que lo único necesario para su realización es un ordenador y el tiempo de desarrollo profesional asociado a un ingeniero informático de nivel junior, con sus gastos derivados.

Recurso	**Coste**
Programador junior (14 €/hora)	4200 €
Herramientas de desarrollo	0 €
Gastos generales	1200 €
Subtotal	**5400 €**
IGIC (7 %)	378 €
Coste total del proyecto	**5778 €**

Tabla 8.1: Presupuesto del proyecto

Bibliografía

[1] W. W. W. Consortium *et al.*, "Rdf 1.1 primer," 2014.

[2] T. Costa and J. Leal, *Publishing Linked data with DaPress*, 08 2013, pp. 67–81.

[3] V. Bicer, T. Tran, A. Abecker, and R.Ñedkov, "Koios: Utilizing semantic search for easy-access and visualization of structured environmental data," in *The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part II 10*. Springer, 2011, pp. 1–16.

[4] L. Codina and C. Rovira, "La web semántica," in *Tendencias en documentación digital*. Trea, 2006.

[5] D. Prieto-González, I. Castilla-Rodríguez, E. González, and M. L. Couce, "Towards the automated economic assessment of newborn screening for rare diseases," *Journal of Biomedical Informatics*, vol. 95, p. 103216, 2019.

[6] ——, "Automated generation of decision-tree models for the economic assessment of interventions for rare diseases using the radios ontology," *Journal of Biomedical Informatics*, vol. 110, p. 103563, 2020.

[7] Wikipedia, "World Wide Web — Wikipedia, the free encyclopedia," https://en.wikipedia.org/wiki/World_Wide_Web, 2023.

[8] W. W. W. Consortium *et al.*, "Rdf 1.1 concepts and abstract syntax," 2014.

[9] ——, "Rdf schema 1.1," 2014.

[10] ——, "Owl 2 web ontology language document overview," 2012.

[11] R. Edlin, C. McCabe, C. Hulme, P. Hall, and J. Wright, "Cost effectiveness modelling for health technology assessment," 2019.

[12] L. Vallejo-Torres, I. Castilla, M. L. Couce, C. Pérez-Cerdá, E. Martín-Hernández, M. Pineda, J. Campistol, A. Arrospide, S. Morris, and P. Serrano-Aguilar, "Cost-effectiveness analysis of a national newborn screening program for biotinidase deficiency," *Pediatrics*, vol. 136, no. 2, pp. e424–e432, 2015.

[13] P. Weber, S. Scholl, and E. R. Baumgartner, "Outcome in patients with profound biotinidase deficiency: relevance of newborn screening," *Developmental medicine and child neurology*, vol. 46, no. 7, pp. 481–484, 2004.

[14] B. Wolf, "Clinical issues and frequent questions about biotinidase deficiency," *Molecular genetics and metabolism*, vol. 100, no. 1, pp. 6–13, 2010.

[15] M. A. Musen, "The protégé project: a look back and a look forward," *AI Matters*, vol. 1, no. 4, pp. 4–12, 2015. [Online]. Available: https://doi.org/10.1145/2757001.2757003

[16] D. Garijo, "Widoco: a wizard for documenting ontologies," in *International Semantic Web Conference*. Springer, Cham, 2017, pp. 94–102. [Online]. Available: http://dgarijo.com/papers/widoco-iswc2017.pdf

[17] K. Donnelly *et al.*, "Snomed-ct: The advanced terminology and coding system for ehealth," *Studies in health technology and informatics*, vol. 121, p. 279, 2006.

[18] N. A. Vasilevsky, N. A. Matentzoglu, S. Toro, J. E. Flack IV, H. Hegde, D. R. Unni, G. F. Alyea, J. S. Amberger, L. Babb, J. P. Balhoff *et al.*, "Mondo: Unifying diseases for the world, by the world," *medRxiv*, pp. 2022–04, 2022.

[19] J. S. Álvarez, "Evaluación económica de medicamentos y tecnologías sanitarias," *DOI*, vol. 10, pp. 978–84, 2012.

[20] M. F. Drummond, M. J. Sculpher, K. Claxton, G. L. Stoddart, and G. W. Torrance, *Methods for the economic evaluation of health care programmes*. Oxford university press, 2015.

[21] J. A. Sacristán, J. Oliva, C. Campillo-Artero, J. Puig-Junoy, J. L. Pinto-Prades, T. Dilla, C. Rubio-Terrés, and V. Ortún, "¿ qué es una intervención sanitaria eficiente en españa en 2020?" *Gaceta Sanitaria*, vol. 34, pp. 189–193, 2020.

[22] F. Erxleben, M. Günther, M. Krötzsch, J. Mendez, and D. Vrandečić, "Introducing wikidata to the linked data web," in *The Semantic Web–ISWC 2014: 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014. Proceedings, Part I 13*. Springer, 2014, pp. 50–65.

[23] S. S. Padrón, "Tfg - stadios," https://github.com/alu0101028491/TFG_StaDiOS, 2023, [Online; accessed 2023-04-24].

[24] L. Vallejo-Torres, B. García-Lorenzo, and P. Serrano-Aguilar, "Estimating a cost-effectiveness threshold for the spanish nhs," *Health economics*, vol. 27, no. 4, pp. 746–761, 2018.

[25] A. Alobaid, D. Garijo, M. Poveda-Villalón, I. Santana-Perez, A. Fernández-Izquierdo, and O. Corcho, "Automating ontology engineering support activities with ontoology," *Journal of Web Semantics*, vol. 57, p. 100472, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570826818300465