

**TUTORES**

**FRANCISCO ANTONIO GÓMEZ VELA Y MIGUEL GARCÍA TORRES**

# **MACHINE LEARNING APLICADO A DATOS BIOMÉDICOS**

**T R A B A J O   D E   F I N   D E   G R A D O**

**ALUMNO**

**ANDRÉS CARRILLO BEJARANO**





## Contenido

Resumen.....	5
Introducción.....	6
OBJETIVOS.....	6
ORGANIZACIÓN DEL DOCUMENTO.....	7
1.    PLAN DE PROYECTO .....	8
1.1    Introducción.....	8
1.2    Objetivos .....	8
1.3    ORGANIZACIÓN DEL PROYECTO.....	10
1.3.1 DIAGRAMA DE ORGANIZACIÓN DEL PROYECTO.....	10
1.3.2 IDENTIFICACIÓN DE LOS INTERESADOS.....	10
1.4 METODOLOGÍA DE GESTIÓN DE PROYECTO.....	11
1.5 PROGRAMA DE TRABAJO .....	11
EDT: Diagrama de desglose de Tareas.....	12
2. ANÁLISIS .....	23
1.    Definición del sistema .....	23
1.1 Alcance del sistema.....	23
1.2 Entorno Tecnológico .....	23
2.    ESPECIFICACIONES DE REQUISITOS.....	24
2.2.1 Catalogo de requisitos.....	24
2.2.2 Casos de Uso .....	26
3.    Subsistemas funcionales .....	27
3.2.1    Identificación de subsistemas .....	27
3.3.2 Relaciones entre subsistemas .....	28
3.3.3 Matriz de trazabilidad entre Subsistemas y casos de Uso .....	28
4.    ANÁLISIS DE CLASES POR SUBSISTEMAS.....	28
4.1 SUB01 – Subsistema de gestión de Usuarios .....	29
4.2 SUB02 – Subsistema de gestión de Datasets.....	29
4.3 SUB03 – Subsistema de gestión de Ejecuciones.....	30
5.    INFORMES .....	31
5.1    Subsistema “SUB-03 Gestión de Ejecuciones” .....	31
6.    COMUNICACIÓN CON SISTEMAS EXTERNOS .....	31
3. DISEÑO .....	33
1    DEFINICIÓN DEL SISTEMA.....	33
1.1    Arquitectura del sistema.....	33
2.2    Requisitos no funcionales y estándares, normas y restricciones del proyecto.....	34

2 ARQUITECTURA DE SOPORTE.....	37
3    MODELO FÍSICO DE DATOS. ....	38
3.1 Diseño del Modelo físico de datos .....	38
3.2 Acceso a los datos. ....	41
4    DISEÑO DE CASOS DE USO. ....	42
4.1 CASOS DE USO SUBSISTEMA DE USUARIO .....	42
4.2 CASOS DE USO SUBSISTEMA DE DATASETS .....	44
4.3 CASOS DE USO SUBSISTEMA DE EJECUCIONES .....	45
5 DISEÑO DE CLASES. ....	48
6    DISEÑO DE INTERFACES. ....	53
7 ESPECIFICACIONES DE CONSTRUCCIÓN.....	63
8    PLAN DE PRUEBAS.....	63
9    REQUISITOS DE IMPLANTACIÓN.....	63
9.1    REQUISITOS DE DOCUMENTACIÓN.....	63
9.2    REQUISITOS DE IMPLANTACIÓN .....	63
4. CODIFICACIÓN .....	65
4.1 SERVIDOR BACKEND.....	65
4.2    APLICACIÓN WEB .....	65
4.3    APLICACIÓN MÓVIL .....	65
5. EXPERIMENTACIÓN .....	67
5.1 DATOS DE ESTUDIO .....	67
5.2 ESPECIFICACIONES DEL EXPERIMENTO .....	67
5.3 RESULTADOS OBTENIDOS .....	67
5.4 CONCLUSIONES .....	68
6. PRUEBAS.....	69
6.1 PRUEBAS UNITARIAS .....	69
6.2 PRUEBAS DE INTEGRACIÓN.....	70
6.3 PRUEBAS DE SEGURIDAD .....	71
7. CONCLUSIONES Y COMPETENCIAS ADQUIRIDAS .....	73
7.1 CONCLUSIONES DEL TRABAJO.....	73
7.2    RELACIÓN CON EL GRADO.....	73
8. BIBLIOGRAFIA .....	75

## Resumen

Este documento se trata de la memoria del Trabajo de Fin de Grado realizado por Andrés Carrillo Bejarano, tutorizado por Francisco Gómez Vela y Miguel García Torres, con el título de “Machine learning aplicado a datos biomédicos” en el que se desarrolla un sistema enfocado a analizar datos biomédicos, y en específico se tratan los campos de machine learning de Selección de atributos y clasificación. Se compondrá un sistema con tres principales componentes: El servidor BackEnd, la aplicación Web y por último una aplicación nativa móvil.

Con este sistema se podrán realizar análisis de datasets de gran tamaño para reducir el número de atributos a utilizar en el procesamiento, y por tanto se hará uso de algoritmos de la librería [\[1\]](#) Weka pero también se incorpora otro [\[2\]](#) algoritmo desarrollado por un equipo de investigadores que componen Francisco Gómez Vela, Miguel García Torres, Belén Melián Batista y J. Marcos Moreno Vega. Con estos algoritmos se podrá reducir el número de atributos y por otra parte se incluyen algoritmos clasificadores para poder comparar el resultado obtenido con los distintos atributos seleccionados. El trabajo de fin de grado ha supuesto un reto para el alumno que ha aprendido a utilizar frameworks nuevos como [\[3\]](#) Spring Boot, [\[4\]](#) Vue, [\[5\]](#)[8.](#) [BIBLIOGRAFIA](#) Nativescript y la Librería de Weka. Y por otra parte ha trabajado sobre el campo de la selección de atributos que, aunque se mencionaba en la Asignatura de Inteligencia Artificial al desarrollar este trabajo se ha podido trabajar más a fondo y se han podido experimentar las ventajas.

## Introducción.

Debido al crecimiento exponencial de datos que se generan diariamente y la necesidad en diferentes campos de analizar estos datos en un tiempo reducido, se generan retos para realizar estas tareas. El coste de computación es elevado y aunque el coste del almacenamiento se ha reducido significativamente estos últimos años tener capacidad para analizar este gran volumen de datos es una tarea difícil.

[6] Recientemente el uso de técnicas de selección de atributos en el campo de la bioinformática ha pasado de ser un ejemplo que seguir a convertirse en un requerimiento para crear un modelo predictivo que trabaje con este tipo de datos, debido a la naturaleza de gran dimensionalidad de los datasets de este campo.

Por lo que los algoritmos de selección de atributos se ofrecen como una solución para este tipo de problemas. Con la selección de atributos se podrá reducir el número de atributos que habrá que utilizar en el momento de crear un modelo predictivo a la vez que también se podrá utilizar para acelerar el proceso de aprendizaje de un modelo basado en machine learning, mejorar la tasa de acierto de un modelo de clasificación si se elige un conjunto reducido de atributos adecuado y por último para reducir el sobre ajuste a el conjunto de datos.

## OBJETIVOS.

El objetivo principal de este trabajo de fin de grado es el desarrollo de un sistema de información que este enfocado en machine learning, este sistema debe de ser capaz de aplicar algoritmos de selección de atributos y otros clasificadores, con los que se podrán analizar datasets de carácter biomédico y que mediante una experimentación se pueda comprobar que efectivamente se obtienen buenos resultados al aplicar este tipo de técnicas. Otro objetivo motivado por el alumno es trabajar con nuevos frameworks de desarrollo para ampliar el número de frameworks y tecnologías aprendidas añadidas a las aprendidas a lo largo del grado, aparte de un interés en el aprendizaje de desarrollo de aplicaciones nativas de móvil, que no se ha trabajado en el grado.

## ORGANIZACIÓN DEL DOCUMENTO.

La estructura de la memoria del trabajo de fin de grado está dividida en 8 capítulos distintos que son listados a continuación acompañados de una breve descripción:

1. Plan de proyecto: Se desarrollan los objetivos, la planificación del trabajo, la organización y metodología de gestión del proyecto.
2. Análisis: Incorpora una primera definición del sistema, las especificaciones de requisitos, subsistemas funcionales, análisis de clases por subsistemas, los informes del sistema y la comunicación con sistemas externos.
3. Diseño: Incluye la definición del sistema, la arquitectura que da soporte al sistema, el modelo físico de datos, el diseño de casos de uso, el diseño de clases, el diseño de las interfaces, las especificaciones de construcción, el plan de pruebas y los requisitos de implantación.
4. Codificación: Descripción de la codificación de distintos componentes del sistema y aspectos importantes de su implementación.
5. Experimentación: Resultados de la experimentación con datos biomédicos aplicándolos con algoritmos que ofrece el sistema desarrollado y conclusiones de estos.
6. Pruebas: Pruebas realizadas en el sistema para cumplir sus requisitos funcionales y no funcionales, incluyendo resultados previstos y obtenidos.
7. Conclusiones y competencias adquiridas: conclusiones del trabajo realizado, competencias adquiridas y la relación con las asignaturas del grado.
8. Bibliografía.



# 1. PLAN DE PROYECTO

En este capítulo de la memoria del proyecto se desarrollará el plan de proyecto donde se detallará en una primera fase del proyecto los principales objetivos del proyecto, la organización del proyecto, la metodología de gestión del proyecto y el programa de trabajo.

## 1.1 Introducción.

Plan de desarrollo del Trabajo de fin de grado de un sistema de machine learning enfocado para tratar datos biomédicos. En el plan de proyecto se recogerá la información necesaria para el desarrollo del proyecto, como los objetivos de este, como la estructura de descomposición del trabajo (EDT), se identificarán los interesados del proyecto, metodología en la que se desarrollara el proyecto, la identificación de los riesgos y planes de contingencia para estos, además de otros temas generales del proyecto.

## 1.2 Objetivos

El objetivo principal del proyecto es emplear algoritmos de selección de atributos para reducir el volumen de datos de distintos datasets y reduciendo el mínimo la tasa de predicción obtenida, esto se hará mediante algoritmos de la librería WEKA, como otro algoritmo adicional desarrollado por un grupo de investigadores que incluye a Miguel García Torres y Francisco Gómez Vela.

Este proyecto requiere de todas las fases para desarrollar un Sistema de información, como sería Análisis, Diseño, e Implementación.

Los objetivos de alto nivel del sistema son los siguientes:

<b>OBJ-01</b>	<b>Gestión de Selección de Atributos</b>
<b>Versión</b>	1.0.0
<b>Autores</b>	Andrés Carrillo Bejarano
<b>Descripción</b>	El sistema deberá de gestionar las ejecuciones anteriores de selección de atributos. Guardando los resultados entre datasets y distintos algoritmos para el caso de que el usuario quiera repetir una de estas, el servidor no vuelva a realizar una ejecución redundante.
<b>Importancia</b>	Alta
<b>Estado</b>	Pendiente de aprobación
<b>Comentarios</b>	

<b>OBJ-02</b>	<b>Gestión de Clasificación</b>
<b>Versión</b>	1.0.0
<b>Autores</b>	Andrés Carrillo Bejarano
<b>Descripción</b>	El sistema deberá de gestionar las ejecuciones anteriores de clasificación de datasets. Guardando los resultados entre datasets y distintos algoritmos para evitar repetir una ejecución redundante.
<b>Importancia</b>	Alta
<b>Estado</b>	Pendiente de aprobación
<b>Comentarios</b>	



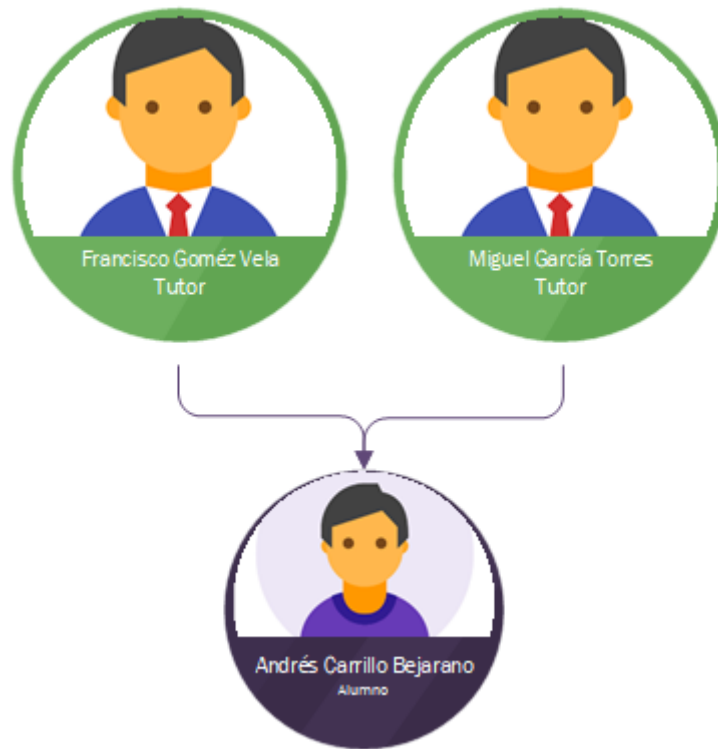
<b>OBJ-03</b>	<b>Gestión de Usuarios</b>
<b>Versión</b>	1.0.0
<b>Autores</b>	Andrés Carrillo Bejarano
<b>Descripción</b>	El sistema deberá de gestionar los usuarios de manera que estos puedan hacer uso de la aplicación mediante la página web y de la API con el mismo usuario y contraseña. El sistema deberá también gestionar el permiso que estos usuarios tienen en la aplicación y denegarles el acceso cuando sea necesario.
<b>Importancia</b>	Alta
<b>Estado</b>	Pendiente de aprobación
<b>Comentarios</b>	

<b>OBJ-04</b>	<b>Gestión de conjuntos de datos</b>
<b>Versión</b>	1.0.0
<b>Autores</b>	Andrés Carrillo Bejarano
<b>Descripción</b>	El sistema deberá de permitir que los usuarios den de alta sus conjuntos de datos para hacer sobre ellos alguna selección de atributos o clasificación para evaluar estos. Además, el sistema deberá de incluir algunos conjuntos de datos predeterminados para hacer pruebas.
<b>Importancia</b>	Alta
<b>Estado</b>	Pendiente de aprobación
<b>Comentarios</b>	

### 1.3 ORGANIZACIÓN DEL PROYECTO.

Al tratarse de un trabajo de fin de grado, el rol de analista, diseñador y programador será realizado por la misma persona, sin embargo, se encuentran dos tutores del trabajo de fin de grado involucrados en la definición del alcance y el desarrollo de este.

#### 1.3.1 DIAGRAMA DE ORGANIZACIÓN DEL PROYECTO.



Tenemos por tanto los siguientes actores:

- Tutor Miguel García Torres: Encargado de la definición del alcance y seguimiento del proyecto, así como la resolución de dudas por parte del alumno. También ofrecerá una guía sobre los algoritmos y las dudas del alumno sobre estos.
- Tutor Francisco Gómez Vela: Encargado de la definición del alcance y seguimiento del proyecto, así como la resolución de dudas por parte del alumno. También ofrecerá una guía sobre la arquitectura del proyecto y las dudas del alumno sobre esta.
- Alumno Andrés Carrillo Bejarano: Encargado de analizar los requerimientos, diseñar el sistema y programar el sistema de información.

#### 1.3.2 IDENTIFICACIÓN DE LOS INTERESADOS.

Los principales interesados en la realización del trabajo de fin de grado serán en primer lugar el alumno, pero además estarán los miembros del comité evaluador, los tutores del proyecto, el personal investigador, o parte de la comunidad tecnológica que puedan aprovechar el trabajo haciendo uso del código fuente publicado en GitHub públicamente, por esto será interesante mencionar que podrán existir futuros contribuyentes al proyecto.

## 1.4 METODOLOGÍA DE GESTIÓN DE PROYECTO.

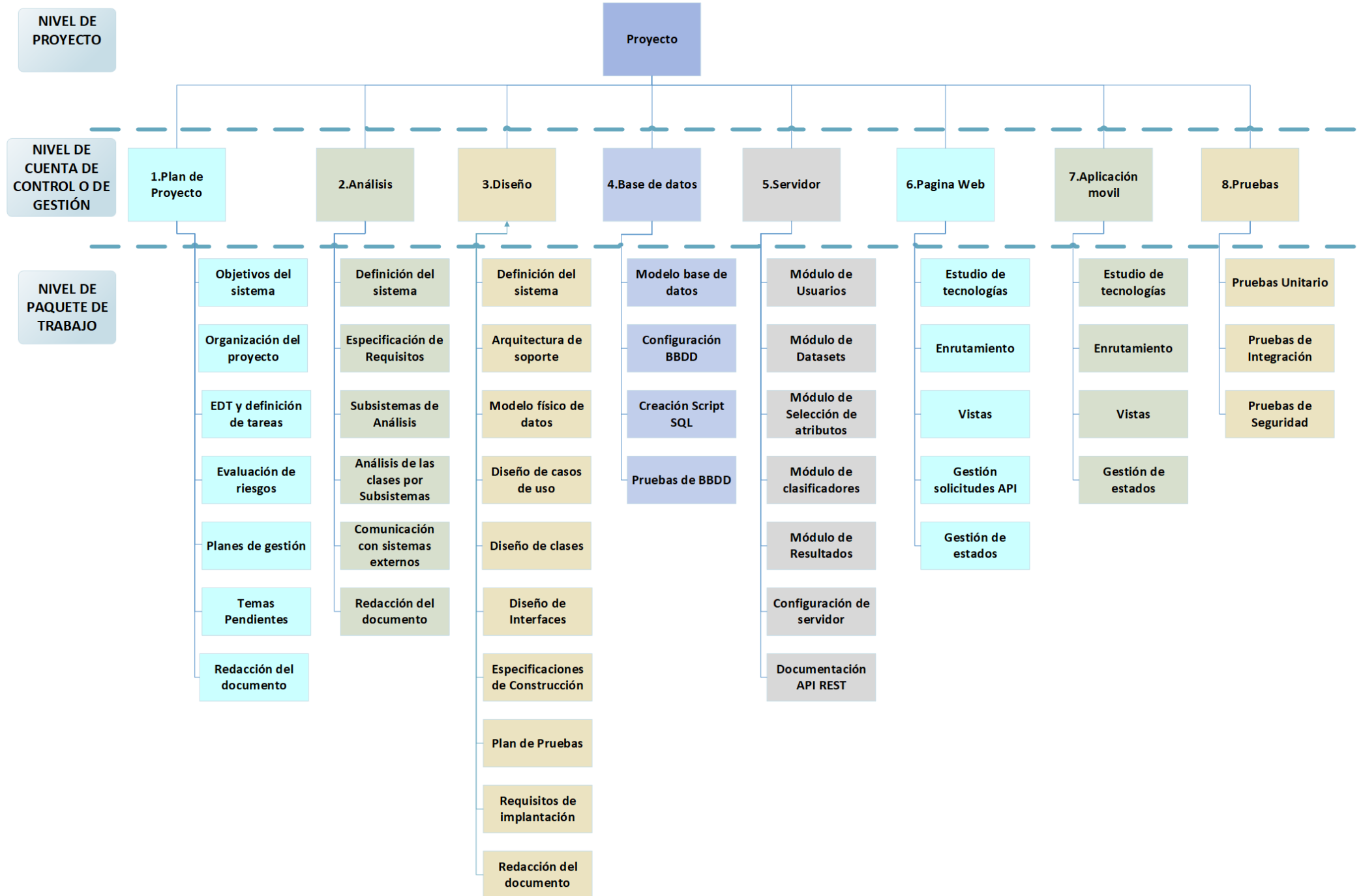
Para la gestión del proyecto se hará uso de una metodología adaptada a un proyecto unipersonal como es el trabajo de fin de grado, además que al tratarse de un proyecto que busca trabajar con tecnologías nuevas para el alumno se intentará flexibilizar en ciertas tareas el tiempo que se empleará en ellas debido a que habrá que tener en cuenta que hay un tiempo flexible de aprendizaje y en el caso de encontrar un error podrá aumentar el tiempo de la tarea considerablemente.

La gestión se ha realizado mediante una metodología ágil que se desarrollaba de manera evolutiva haciendo uso de un tablero en GitHub en el apartado Projects del repositorio y haciendo uso de los Issues para mantener un seguimiento de los cambios realizados y los problemas en los que se ha trabajado.

Se han realizado iteraciones de una semana para realizar las tareas preparadas en el tablero y hacer un seguimiento con los tutores del avance del proyecto, para esto se compartía la pantalla para mostrar el progreso o se cuestionaban dudas sobre el desarrollo.

## 1.5 PROGRAMA DE TRABAJO

El objetivo del plan de trabajo consiste en realizar un trabajo de fin de grado que tiene como principal objetivo desarrollar un sistema de machine learning que se sirva públicamente mediante una Api, por una aplicación web y aplicación móvil. La aplicación se dividirá en distintos entregables necesarios para la entrega del trabajo de fin de grado, quedan detallados cada uno de estos en el siguiente diagrama de desglose de tareas.



### 1.5.1 Plan de Proyecto

Código	Nombre	Duración
1.1	Objetivos del sistema	5 horas
Descripción: Estudio para acotar la idea del trabajo en unos objetivos principales del sistema que definirá a alto nivel las características del sistema para tener en cuenta.		

Código	Nombre	Duración
1.2	Organización del proyecto	5 horas
Descripción: Se detallará en un diagrama un esquema básico de la organización del proyecto en el equipo, la identificación de estos y las responsabilidades y funciones de los interesados.		

Código	Nombre	Duración
1.3	EDT y definición de tareas	9 horas
Descripción: Se realizará el desglose de tareas de trabajo en los tres niveles, hasta llegar a nivel de paquete de trabajo. Se detallará un calendario de hitos principales y fechas objetivo para cada uno.		

Código	Nombre	Duración
1.4	Evaluación de riesgos	5 horas
Descripción: Se creará un plan de riesgos con los principales riesgos identificados y las acciones que realizar en cada uno.		

Código	Nombre	Duración
1.5	Planes de gestión	3 horas
Descripción: Se detallarán los diferentes planes de gestión auxiliares que se tendrán en cuenta en la elaboración del proyecto.		

Código	Nombre	Duración
1.6	Temas pendientes	2 horas
Descripción: Detallamos los temas pendientes en caso de haber dejado algún tema por tratar después de finalizar el proyecto o alguna decisión.		

Código	Nombre	Duración
1.7	Redacción del documento	12 horas
Descripción: Una vez realizados todos los puntos del documento de Plan de proyecto se procede a la elaboración del documento. Así como la creación del índice y la presentación del documento.		

### 1.5.2 Análisis

Código	Nombre	Duración
2.1	Definición del sistema	8 horas
Descripción: Definición del alcance del sistema, listado de objetivos y se comentará el entorno tecnológico en que se desarrollará el proyecto.		

Código	Nombre	Duración
2.2	Especificación de Requisitos	14 horas
Descripción: Detalles sobre los requisitos funcionales a cubrir en la aplicación y a partir de estos se creará un diagrama de casos de uso y se indicará la trazabilidad entre casos de uso y requisitos.		

Código	Nombre	Duración
2.3	Subsistemas de Análisis	6 horas

**Descripción:**

La aplicación se divide en distintos subsistemas que se agrupan por funcionalidades, requisitos y casos de uso, estos están relacionados entre sí para formar el sistema, se detallara la relación entre estos y una matriz de trazabilidad casos de uso – subsistema.

Código	Nombre	Duración
2.4	Análisis de las clases por subsistemas	16 horas
<b>Descripción:</b> Se realizará un primer análisis sobre la funcionalidad de cada subsistema y se detallaran las distintas clases del modelo de negocio que habrá en cada subsistema detallando los atributos.		

Código	Nombre	Duración
2.5	Informes	16 horas
<b>Descripción:</b> Se realizará un análisis de los informes que genera el sistema, así como los campos que contienen y la utilidad que ofrecen al usuario final.		

Código	Nombre	Duración
2.6	Comunicación con sistemas externos	4 horas
<b>Descripción:</b> Se hablará de la comunicación entre los distintos componentes que forman el sistema, para aclarar esta conexión se diseñara un primer diagrama de despliegue del sistema indicando la relación entre los distintos componentes.		

Código	Nombre	Duración
2.7	Redacción del documento	12 horas
<b>Descripción:</b> Una vez realizados todos los puntos del documento de Análisis se procede a la elaboración del documento. Así como la creación del índice y la presentación del documento.		

### 1.5.3 Diseño

Código	Nombre	Duración
3.1	Definición del sistema	8 horas



<p>Descripción:</p> <p>Se profundizará sobre la primera aproximación del sistema de análisis y se incluirá la arquitectura del sistema explicada mediante un diagrama de despliegue y requisitos no funcionales.</p>
--

Código	Nombre	Duración
3.2	Arquitectura de soporte	4 horas

<p>Descripción:</p> <p>Se hablará sobre los distintos componentes del sistema y que servicio da cada uno hablando de las tecnologías en las que se sustentan.</p>
---

Código	Nombre	Duración
3.3	Modelo físico de datos	6 horas

<p>Descripción:</p> <p>Identificación de las tablas de la base de datos y se realizara una tabla con la información de los campos, claves únicas y ajenas de cada una de las tablas. También se comentará la forma en la que el sistema accede a la base de datos.</p>
--

Código	Nombre	Duración
3.4	Diseño de casos de uso	2 horas

<p>Descripción:</p> <p>Se realizarán los casos de uso reales en los que se desarrollara el proceso de interacción entre el usuario y el sistema, y las observaciones necesarias. En estos se hablará cuando sea necesario de que se conectan a el servidor BackEnd.</p>
---

Código	Nombre	Duración
3.5	Diseño de clases	14 horas

<p>Descripción:</p> <p>Se creará un diagrama de las diferentes clases que forman el sistema, los métodos que tienen, los atributos y como están conectados entre si. Se incluirá una descripción del diseño de las clases.</p>
--

Código	Nombre	Duración
3.6	Diseño de interfaces	10 horas

<p>Descripción:</p> <p>Se detallará cada una de las interfaces del sistema, se detallarán los campos de cada una, los botones y una breve descripción de la interfaz, acompañada de un Mockup.</p>
--

Código	Nombre	Duración
3.7	Especificaciones de construcción	6 horas

<p>Descripción:</p> <p>Se hablará del entorno de construcción que es necesario para levantar el sistema y las diferencias de la construcción entre producción y desarrollo.</p>
---

Código	Nombre	Duración
3.8	Plan de pruebas	6 horas

<p>Descripción:</p> <p>Elaboración de un plan de pruebas de la aplicación que verifique los requisitos funcionales y no funcionales de este mediante distintos tipos de pruebas.</p>
--

Código	Nombre	Duración
3.9	Requisitos de implantación	12 horas

<p>Descripción:</p> <p>Información sobre los requisitos del usuario para operar con el sistema en cuanto a documentación como manuales de usuario y requisitos de implantación del sistema.</p>
---

Código	Nombre	Duración
3.10	Redacción del documento	15 horas

<p>Descripción:</p> <p>Una vez realizados todos los puntos del documento de Diseño se procede a la elaboración del documento. Así como la creación del índice y la presentación del documento.</p>
--

#### 1.5.4 Base de datos

Código	Nombre	Duración
4.1	Modelo base de datos	8 horas

<p>Descripción:</p> <p>Se diseñará el modelo de base de datos con todos los campos necesarios, sus relaciones y las restricciones de este.</p>
--

Código	Nombre	Duración
4.2	Configuración BBDD	4 horas
<p>Descripción:</p> <p>Se configurará la base de datos mediante las anotaciones de Spring Boot que sean necesarias para la efectiva unión de las distintas tablas.</p>		

Código	Nombre	Duración
4.3	Creación script SQL.	1 hora
<p>Descripción:</p> <p>Se conectará a la Base de datos mediante un cliente web, en el que se realizará la exportación del Script SQL de creación de tablas, con sus campos relacionales y las restricciones.</p>		

Código	Nombre	Duración
4.4	Pruebas de BBDD.	6 horas
<p>Descripción:</p> <p>Se realizarán distintas pruebas para comprobar el correcto funcionamiento de la Base de datos. Se realizarán pruebas de conectividad, de carga y de integridad referencial.</p>		

### 1.5.5 Servidor

Código	Nombre	Duración
5.1	Módulo de usuarios	35 horas
<p>Descripción:</p> <p>Realización de la implementación del módulo de usuarios en el servidor, así como la autenticación por JSON web token, y la configuración de seguridad del servidor para que solo las peticiones al servicio de Login y registro sean accesibles sin JSON web token.</p>		

Código	Nombre	Duración
5.2	Módulo de Datasets	18 horas

**Descripción:**

Implementación del módulo de datasets, en el que se incluirá la subida y bajada de archivos, también se filtrará para aceptar los datasets que puedan ser procesados por los algoritmos disponibles. Se establecerá el límite de tamaño de estos como una variable configurable.

Código	Nombre	Duración
5.3	Módulo de selección de atributos	40 horas

**Descripción:**

Implementación del módulo de selección de atributos en el que se incluirán los algoritmos de los paquetes necesarios para que puedan ser ejecutados por los usuarios con sus propios datasets, se realizara de manera que el sistema detecte el tipo de archivo automáticamente.

Código	Nombre	Duración
5.4	Módulo de clasificadores	24 horas

**Descripción:**

Implementación del módulo de clasificadores en el que se incluirán los algoritmos de los paquetes necesarios para que el usuario los pueda ejecutar con sus propios datasets y comprobar la tasa de predicción obtenida en estos.

Código	Nombre	Duración
5.5	Módulo de Resultados	20 horas

**Descripción:**

Implementación del módulo de resultados en el que se realizarán las tareas de consultar el resultado de los algoritmos de selección de atributos y de clasificadores, y que contendrá la información relevante de estos. Se encargará de que no se realicen la misma ejecución entre un Dataset y un algoritmo.

Código	Nombre	Duración
5.6	Configuración de servidor	6 horas

**Descripción:**

Se realizará la configuración y la inclusión del gestor de dependencias Maven en el proyecto correspondiente para que pueda desplegarse en local para continuar su desarrollo en nuevas versiones y en el servidor de producción, con distintos perfiles, además de la inclusión de un archivo de propiedades que podrá ser configurado para cada perfil.

Código	Nombre	Duración
5.7	Documentación API REST	8 horas
<p>Descripción:</p> <p>Documentación de la API REST para que cualquier usuario pueda hacer uso de este servicio dentro de su aplicación, se mostraran los formatos aceptados para cada petición y la respuesta que dará el sistema en función del error encontrado.</p>		

#### 1.5.6 Página Web

Código	Nombre	Duración
6.1	Estudio de tecnologías	12 horas
<p>Descripción:</p> <p>Estudio y prototipado de distintas tecnologías para comprobar cual de estas puede funcionar mejor con el servidor de BackEnd, así como estudiar cual es mas apropiada para el proyecto.</p>		

Código	Nombre	Duración
6.2	Enrutamiento	16 horas
<p>Descripción:</p> <p>Implementación del enrutamiento entre las distintas vistas para la navegación del usuario por la página y cuando este tenga permisos para acceder al recurso solicitado.</p>		

Código	Nombre	Duración
6.3	Vistas	20 horas
<p>Descripción:</p> <p>Creación de las vistas necesarias para cada módulo de la aplicación web, con las funciones necesarias para este funcionamiento, así como la visualización correcta del contenido.</p>		

Código	Nombre	Duración
6.4	Gestión solicitudes API	18 horas
<p>Descripción:</p> <p>Creación de las funciones necesarias para tratar las solicitudes a la API y el contenido que devuelven para ser mostrado correctamente en las diferentes vistas, así como el estado de distintos componentes.</p>		

Código	Nombre	Duración
6.5	Gestión de estados	12 horas
Descripción: Implementación de la gestión de estados en la aplicación para simplificar la programación y la navegación en las vistas.		

### 1.5.7 Aplicación móvil

Código	Nombre	Duración
7.1	Estudio de tecnologías	20 horas
Descripción: Estudio y prototipado de distintas tecnologías para hacer uso del posible proyecto de la aplicación Web, y los cambios necesarios a realizar para adaptarlo a la tecnología adecuada.		

Código	Nombre	Duración
7.2	Enrutamiento	12 horas
Descripción: Implementación del enrutador de la aplicación, gestionara la navegación entre las diferentes paginas cambiando las vistas que se muestran.		

Código	Nombre	Duración
7.3	Vistas	15 horas
Descripción: Implementación de las vistas con los elementos propios de móvil para que puedan ser lo más próximas a la aplicación web.		

Código	Nombre	Duración
7.4	Gestión de estados	10 horas
Descripción: Implementación de la gestión de estados en la aplicación para simplificar la programación y la navegación en las vistas.		

### 1.5.8 Pruebas

Código	Nombre	Duración
8.1	Pruebas unitarias	12 horas
Descripción: Se realizarán pruebas unitarias a los componentes más críticos del sistema para los que se comprobaba que cumplen con los requisitos funcionales del sistema.		

Código	Nombre	Duración
8.1	Pruebas de integración	6 horas
Descripción: Se comprobará que el sistema funciona correctamente al integrar todos los módulos y componentes de este, se realizará una vez desplegado el sistema en un servidor similar al de producción.		

Código	Nombre	Duración
8.1	Pruebas de seguridad	7 horas
Descripción: Se comprobará mediante distintas peticiones que la capa de seguridad del servidor BackEnd funciona correctamente, y que en la aplicación web no se permite el acceso a interfaces restringidas para cierto perfil del usuario, al igual que en la aplicación móvil.		



## 2. ANÁLISIS

En este capítulo de la memoria se desarrollará en una segunda fase, una primera definición del sistema, las especificaciones de requisitos, se hablará de la división del sistema en subsistemas funcionales, un Análisis de clases por subsistemas, se detallarán los informes del sistema y la comunicación con sistemas externos.

### 1. Definición del sistema

#### 1.1 Alcance del sistema

Se pretende diseñar un sistema que integre algoritmos de selección de atributos y clasificadores para que los clientes puedan hacer uso de este servicio a partir de una página web, mediante la API publica gestionada en un servidor externo o por la aplicación móvil.

El sistema permitirá a los usuarios subir al servidor sus propios conjuntos de datos sobre los que se analizara con los algoritmos que se incluirán en la plataforma. Y podrán tener un histórico con los resultados de las ejecuciones previas.

#### 1.2 Entorno Tecnológico

El sistema BackEnd se desplegará en un sistema externo que da servicio por una API pública que gestionará todas las peticiones recibidas tanto de la aplicación web como de cualquier otro sistema que consuma la API pública mediante peticiones que contengan datos en formato JSON y que tendrán que estar autenticados previamente.

Por otro lado, el FrontEnd tiene que estar disponible en múltiples plataformas por lo que será una aplicación web, que tenga un diseño sencillo y que facilite la usabilidad de los usuarios que no tienen un perfil técnico.

El sistema necesitará una capa persistencia para guardar toda la información relativa al sistema. Por esto el servidor BackEnd hará uso de una base de datos que estará gestionada externamente para el caso en el que el sistema tenga algún problema o error, los datos puedan seguir siendo accesibles tras una recuperación del sistema lo antes posible.

Al ser una API pública tendrá que ser importante los aspectos de concurrencia, alta disponibilidad y seguridad.

## 2. ESPECIFICACIONES DE REQUISITOS

### 2.2.1 Catalogo de requisitos

RF-01	Alta Conjunto de datos
Versión	1.0.0
Autores	<ul style="list-style-type: none"><li>Andrés Carrillo Bejarano</li></ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"><li>OBJ-XX: Gestión de Conjunto de datos</li></ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"><li>Poder subir un conjunto de datos del tamaño deseado para ser procesado posteriormente, en el que contendrá el nombre del archivo, la ruta donde se guarda, el tamaño y el usuario que lo subió.</li></ul>
Actores	Usuarios
Comentarios	-

RF-02	Consultar Conjunto de datos
Versión	1.0.0
Autores	<ul style="list-style-type: none"><li>Andrés Carrillo Bejarano</li></ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"><li>OBJ-XX: Gestión de Conjunto de datos</li></ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"><li>Poder consultar los detalles de los conjuntos de datos del usuario que ha iniciado sesión.</li></ul>
Actores	Usuarios
Comentarios	-

RF-03	Alta de Usuario
Versión	1.0.0
Autores	<ul style="list-style-type: none"><li>Andrés Carrillo Bejarano</li></ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"><li>OBJ-XX: Gestión de Usuarios</li></ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"><li>Poder dar de alta al sistema a nuevos usuarios introduciendo nombre de usuario, contraseña y email</li></ul>
Actores	Invitado
Comentarios	El sistema se conectará al servidor BackEnd para esta operación.

RF-04	Baja de Usuario
Versión	1.0.0
Autores	<ul style="list-style-type: none"><li>Andrés Carrillo Bejarano</li></ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"><li>OBJ-XX: Gestión de Usuarios</li></ul>

Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Poder cerrar sesión del sistema , cuando el usuario lo desee.</li> </ul>
Actores	Usuario o Administrador
Comentarios	El sistema se conectará al servidor BackEnd para esta operación.

RF-05	Consulta de Usuario
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>• Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> </ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Poder consultar la información relacionada con el usuario registrado.</li> </ul>
Actores	Usuario
Comentarios	-

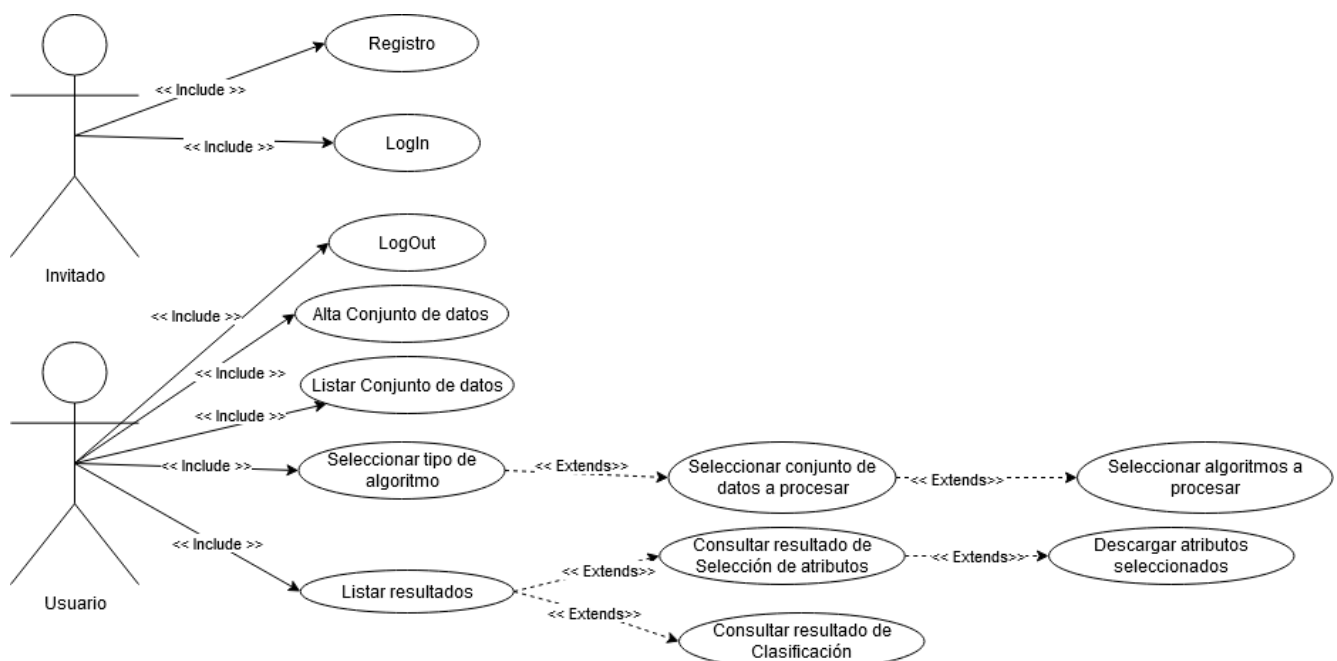
RF-06	Procesar algoritmo de clasificación.
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>• Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> </ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Permitir al usuario datasets y entre distintos algoritmos de clasificación, para valorar la tasa de predicción obtenida con distintos datasets.</li> </ul>
Actores	Invitado
Comentarios	El sistema se conectará al servidor BackEnd para esta operación.

RF-07	Procesar algoritmo de selección de atributos.
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>• Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> </ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"> <li>• Permitir al usuario seleccionar datasets y entre distintos algoritmos de selección de atributos, para así reducir el volumen de los datasets.</li> </ul>
Actores	Invitado
Comentarios	El sistema se conectará al servidor BackEnd para esta operación.

RF-08	Consultar resultados de clasificación.
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>OBJ-XX: Gestión de Usuarios</li> </ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"> <li>Permitirá al Usuario consultar los resultados de ejecuciones anteriores de clasificación.</li> </ul>
Actores	Usuario o Administrador
Comentarios	El sistema se conectará al servidor BackEnd para esta operación.

RF-09	Consultar resultados de selección de atributos.
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>OBJ-XX: Gestión de Usuarios</li> </ul>
Descripción	El sistema deberá contemplar las siguientes funcionalidades: <ul style="list-style-type: none"> <li>Permitirá al Usuario consultar los resultados de ejecuciones anteriores de selección de atributos.</li> </ul>
Actores	Usuario o Administrador
Comentarios	El sistema se conectará al servidor BackEnd para esta operación.

### 2.2.2 Casos de Uso



Los casos de uso quedan detallados en el capítulo de diseño.

### 3. Subsistemas funcionales

El sistema se ha dividido en 3 subsistemas diferentes, cada uno con una funcionalidad distinta. El usuario podrá acceder a todos los subsistemas, siendo el más perjudicado el perfil de invitado que solo podrá acceder a la gestión de usuarios para registrarse e iniciar sesión en el sistema.

#### 3.2.1 Identificación de subsistemas

Se identifican los siguientes subsistemas:

- **SUB01 – Gestión de Usuarios:** Gestionara el acceso del sistema y la información relevante de los usuarios registrados.
- **SUB02 – Gestión de Datasets:** Encargado de la gestión de los conjuntos de datos subidos por los Usuarios y de la información de estos.
- **SUB03 – Gestión de Ejecuciones:** Encargado de gestionar los algoritmos de selección de atributo, clasificadores y los resultados de las ejecuciones.

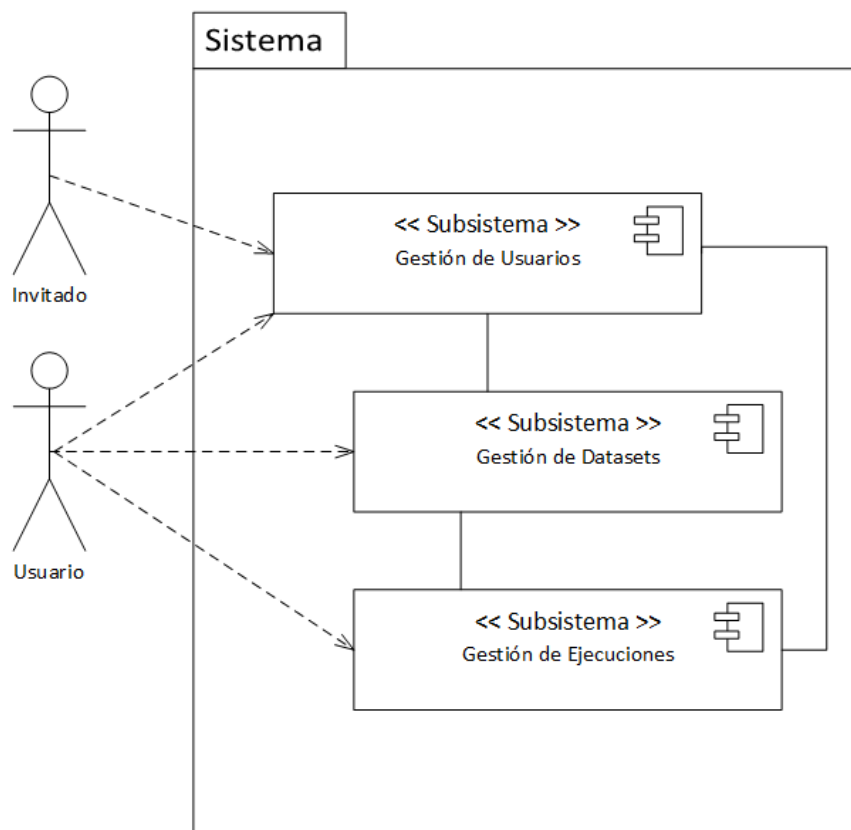


Figura 1 Subsistemas

En la Figura 1 podemos ver los tres subsistemas identificados, a que sistemas puede acceder cada tipo de usuario y como se relacionan entre si.

### 3.3.2 Relaciones entre subsistemas

Los tres subsistemas identificados están relacionados entre sí, debido a que las operaciones se realizan sobre datasets que suben los usuarios y para acceder a estas operaciones el usuario debe estar logado en el sistema para poder comprobar que tenga el acceso permitido al recurso.

### 3.3.3 Matriz de trazabilidad entre Subsistemas y casos de Uso

Incluimos la matriz entre subsistemas y casos de uso.

Subsistemas	Casos de uso
<b>SUB-01 Usuarios</b>	CU-01, CU-02, CU-03
<b>SUB-02 Datasets</b>	CU-04, CU-05
<b>SUB-03 Ejecuciones</b>	CU-06, CU-07, CU-08, CU-09, CU-10, CU-11, CU-12

## 4. ANÁLISIS DE CLASES POR SUBSISTEMAS

En la Figura 2 podemos ver el diagrama de clases de negocio donde se representan las distintas clases que pertenecen al sistema completo, como se relacionan entre ellas y sus atributos.

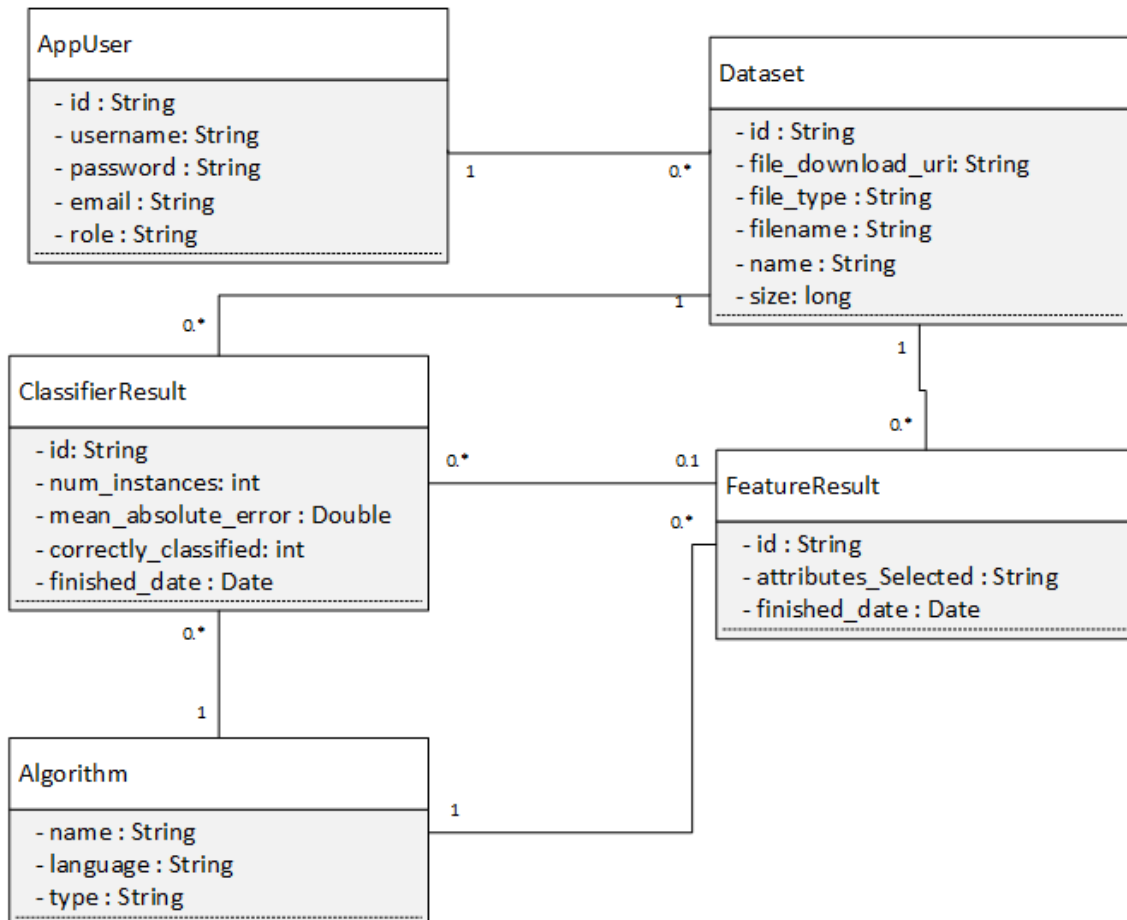


Figura 2 Diagrama de clases de negocio

#### 4.1 SUB01 – Subsistema de gestión de Usuarios

Se gestionan las cuentas de usuarios, ya sean invitados (usuarios sin registrar) o usuarios. Este se encargará de gestionar que las credenciales enviadas al servidor BackEnd sean correctas y el usuario tenga los permisos para realizar la operación que requiere en cada petición a la API.

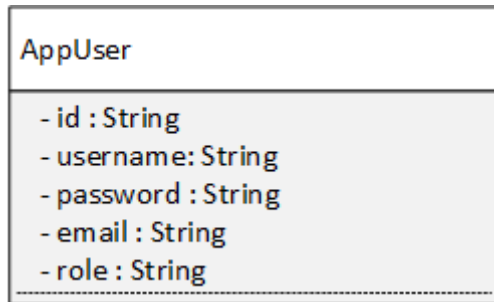


Figura 3 Clases del Subsistema gestión de Usuarios

La única clase perteneciente al subsistema de gestión de usuarios es la de Usuario, aunque en el sistema haya mas relaciones con el usuario en este subsistema solo se tratara la gestión de la autenticación del usuario.

#### 4.2 SUB02 – Subsistema de gestión de Datasets.

Se gestionarán los distintos datasets cargados por los usuarios y la información necesaria para poder identificarlos en el servidor y información que puede ser relevante para el usuario, así como el tipo de archivo y el tamaño de este, también se verificara en la parte del servidor BackEnd que el dataset sea de un tipo de archivo permitido, en caso contrario se anulara la carga del dataset.

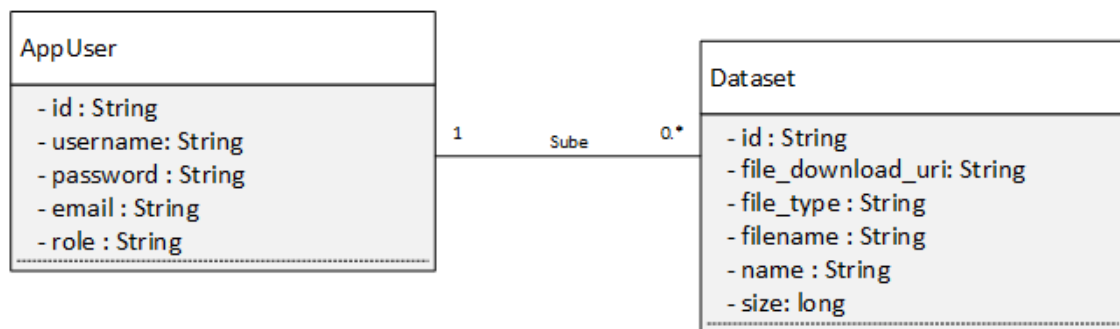


Figura 4 Clases del subsistema de gestión de datasets

Las clases relevantes al subsistema de gestión de datasets, son las de Usuario y Dataset debido a que cada dataset esta relacionado con el usuario que lo subió debido a que al ser un servicio de inteligencia artificial, y ser datasets de gran tamaño, el usuario requerirá que ningún otro usuario pueda acceder a estos datasets.



### 4.3 SUB03 – Subsistema de gestión de Ejecuciones.

Se gestionarán las distintas ejecuciones en cuanto a los algoritmos clasificadores o de selección de atributos, así como se guardarán las relaciones entre estos y el conjunto de datos sobre el que se realizan, esto es interesante para el sistema debido a que de esta manera podemos evitar realizar ejecuciones redundantes que pueden ser de un gran coste de computación.

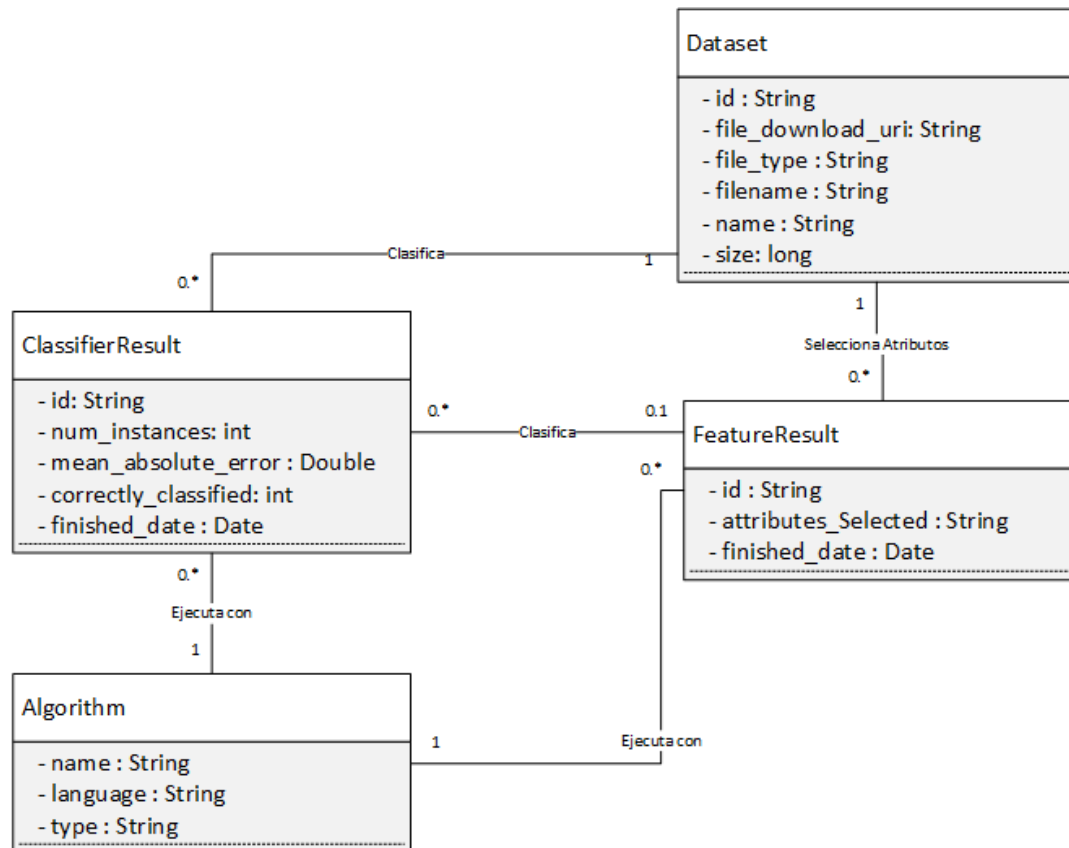


Figura 5 Clases del subsistema de Gestión de ejecuciones

En la figura podemos ver las clases que pertenecen al subsistema de gestión de ejecuciones, como son las clases de: Dataset, ClassifierResult, FeatureResult y Algoritmo.

Las ejecuciones se realizarán por un clasificador o un algoritmo de selección de atributos, usando un algoritmo. En el caso del resultado de una ejecución de selección de atributos se guardará una lista de atributos que han sido seleccionados finalmente y la relación con el dataset realizado. Por el lado de los algoritmos clasificadores podremos guardar una referencia a un algoritmo de selección de atributos esto es para realizar la clasificación sobre los atributos seleccionados previamente, aun así necesitaremos la referencia al dataset que contendrá la información del archivo sobre el que tendremos que filtrar los atributos.

## 5. INFORMES

Para este sistema solo se presenta un informe debido a que la información esta disponible a partir de la aplicación web o la aplicación móvil, sin embargo se dispone de la opción de descargar un CSV del resultado de una ejecución de selección de atributos.

### 5.1 Subsistema “SUB-03 Gestión de Ejecuciones”

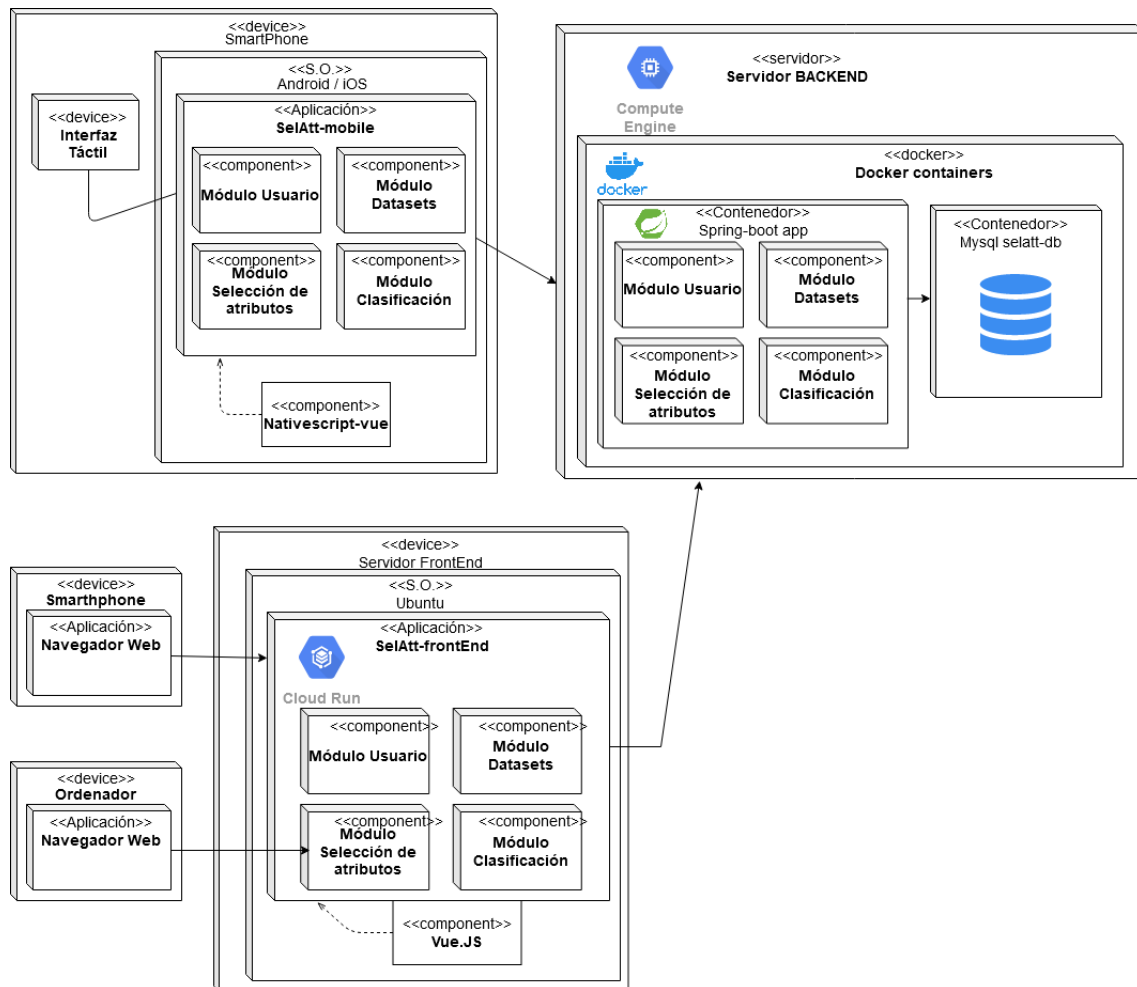
En el subsistema de gestión de ejecuciones existe un informe del resultado de la ejecución de un algoritmo de selección de atributos, para que el usuario pueda descargar los atributos seleccionados.

IF-001: Atributos Seleccionados			
Descripción	Archivo CSV descargable por el usuario en el que se guarda la lista de atributos seleccionados		
Campos	Nombre	Tipo datos	Descripción
Lista de atributos	Atributos	Lista de entero	Una lista con los índices de los atributos seleccionados que están separados por una coma.

## 6. COMUNICACIÓN CON SISTEMAS EXTERNOS

En nuestro sistema no existen unos sistemas externos a los que nos comunicamos, pero el sistema tiene un servidor BackEnd que se encuentra separado de los demás sistemas. La comunicación con el servidor BackEnd se realiza mediante peticiones a la API disponible detallada con detalle en la página web en el apartado de la API, donde se detalla los métodos disponibles, los parámetros necesarios y las respuestas que se obtendrán.

Tanto la aplicación web como la aplicación móvil hacen uso de esta API para la mayoría de las operaciones, dentro de estas se gestionarán las respuestas a esta API y se dispondrán los servicios a los usuarios de una manera más sencillos para aquellos usuarios que no tengan un perfil técnico de desarrollador.



Como podemos observar en el diagrama de despliegue la aplicación móvil y la aplicación web atacan mediante peticiones http a la API pública del servidor BackEnd Spring Boot Dockerizado.

### 3. DISEÑO

En este capítulo de la memoria del proyecto se desarrollará la tercera fase del desarrollo y previa a la implementación en la que se hablará de la definición del sistema, la arquitectura que da soporte al sistema, el modelo físico de los datos, los casos de uso, el diseño de las interfaces, especificaciones de construcción, el plan de pruebas y como último los requisitos de implantación del sistema.

#### 1 DEFINICIÓN DEL SISTEMA.

El sistema de machine learning integrara algoritmos de selección de atributos y clasificadores de Weka y otro algoritmo externo de selección de atributos, para que los clientes puedan hacer uso de este servicio a partir de una página web desarrollada en Vue , mediante la API publica RESTFul desarrollada con Spring Boot en Java y por último se dará servicio en una aplicación nativa de Android e iOS para la consulta de los resultados enviados previamente y/o solicitud de nuevos resultados de datasets cargados previamente.

El sistema permitirá a los usuarios subir al servidor sus propios conjuntos de datos sobre los que se analizara con los algoritmos que se incluirán en la plataforma. Y podrán tener un histórico con los resultados de las ejecuciones previas con los que podrán comparar los resultados obtenidos al reducir sus datasets con los algoritmos de selección de atributos.

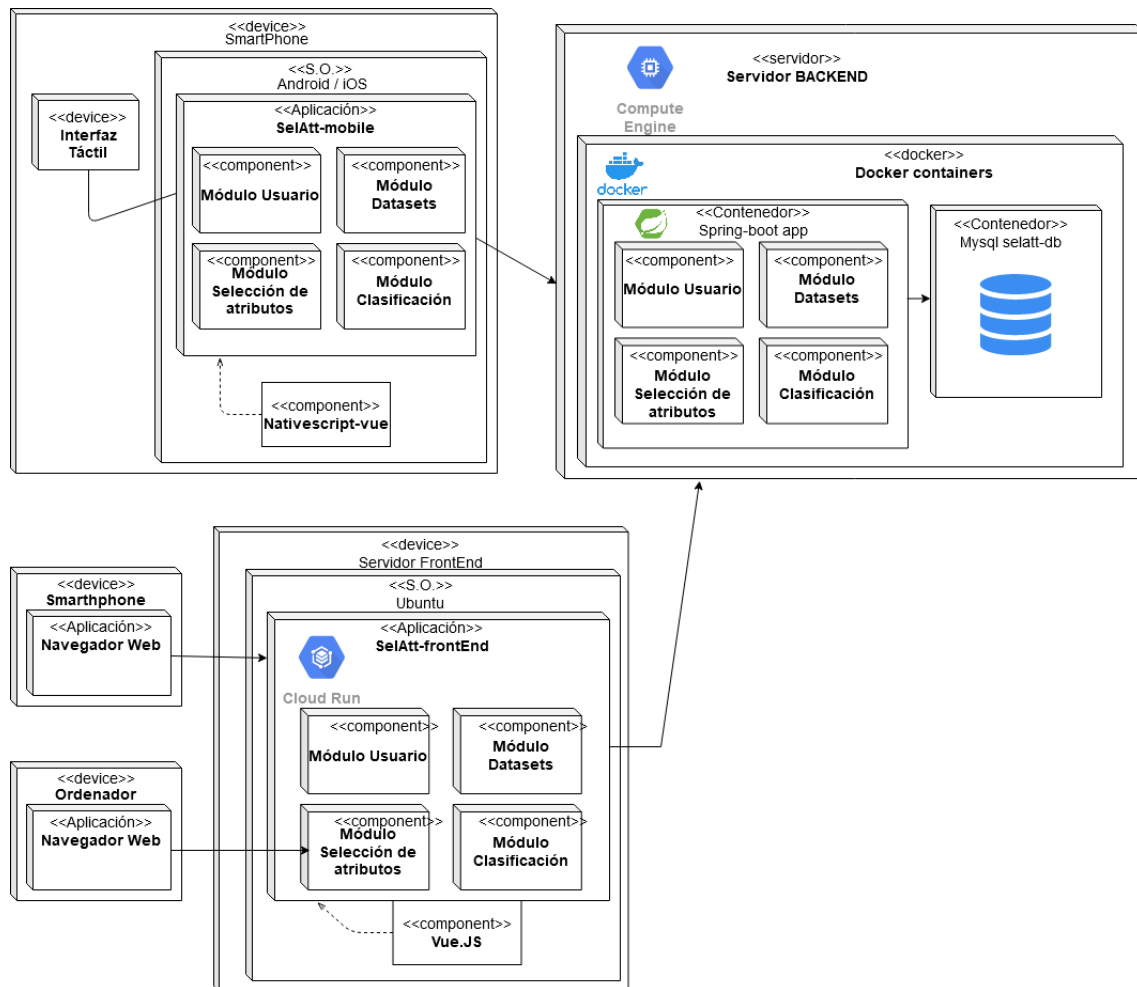
##### 1.1 Arquitectura del sistema.

La arquitectura del sistema está dividida en tres módulos principales que son el servidor BackEnd que da servicio a todo el sistema mediante la API RESTFul pública, la aplicación web que sirve de interfaz multiplataforma.

El sistema BackEnd se desplegará en un servidor Spring Boot en el que se gestionarán todas las peticiones RESTFul tanto del FrontEnd como de cualquier otro BackEnd que haga uso de la API pública mediante estas peticiones en formato JSON que harán uso de un JSON web token para validar la autenticación de usuarios. El servidor BackEnd se conecta con una base de datos MYSQL que está gestionada por un contenedor de Docker configurado previamente, en el que se gestiona adicionalmente un volumen compartido para guardar todos los datos en caso de que el servidor se caiga.

Por otro lado, el FrontEnd está desarrollado en una aplicación VueJS, que es un framework de Javascript y que haciendo uso del framework de componentes [7] Vuetify para darle un diseño de Material Design a la aplicación y hacer uso de elementos dinámicos, La gestión de distintos estados en la aplicación se realizara mediante la librería [8] Vuex y las peticiones a la API RESTFul mediante axios.

La aplicación móvil será nativa tanto en Android como en iOS desarrollada en Nativescript y con el plugin de Vue, con la que el usuario podrá consultar los resultados de sus ejecuciones y realizar nuevas ejecuciones.



## 2.2 Requisitos no funcionales y estándares, normas y restricciones del proyecto.

RNF-XX	Concurrencia
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>OBJ-XX: Gestión de Usuarios</li> <li>OBJ-XX: Gestión de Conjunto de datos</li> <li>OBJ-XX: Gestión de Ejecuciones</li> </ul>
Descripción	<p>El sistema deberá contemplar las siguientes funcionalidades:</p> <ul style="list-style-type: none"> <li>Para las peticiones al servidor BackEnd habrá que tener en cuenta que debe estar disponible para ser consumido por múltiples peticiones</li> </ul>
Actores	Usuarios
Comentarios	-

RNF-XX	Alta disponibilidad
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>Andrés Carrillo Bejarano</li> </ul>

Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> <li>• OBJ-XX: Gestión de Conjunto de datos</li> <li>• OBJ-XX: Gestión de Ejecuciones</li> </ul>
Descripción	El sistema debe garantizar un cierto grado de absoluta continuidad operacional durante el mayor periodo posible.
Actores	Usuarios
Comentarios	-

RNF-XX	Seguridad del sistema
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>• Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> <li>• OBJ-XX: Gestión de Conjunto de datos</li> <li>• OBJ-XX: Gestión de Ejecuciones</li> </ul>
Descripción	El sistema debe estar preparado en aspectos de seguridad para proteger el acceso a datos privados de los usuarios.
Actores	Usuarios
Comentarios	-

RNF-XX	Robustez
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>• Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> <li>• OBJ-XX: Gestión de Conjunto de datos</li> <li>• OBJ-XX: Gestión de Ejecuciones</li> </ul>
Descripción	El sistema debe de tener una capacidad alta para hacer frente a errores, al tratarse de una API pública el sistema debe estar preparado para las distintas casuísticas.
Actores	Usuarios
Comentarios	-

RNF-XX	Caída del servidor
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>• Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> <li>• OBJ-XX: Gestión de Conjunto de datos</li> <li>• OBJ-XX: Gestión de Ejecuciones</li> </ul>
Descripción	El sistema debe estar preparado para que si hay una caída del servidor se restablezca el servicio en menos de 5 minutos.
Actores	Usuarios
Comentarios	-

RNF-XX	Escalabilidad
Versión	1.0.0
Autores	<ul style="list-style-type: none"> <li>• Andrés Carrillo Bejarano</li> </ul>
Fuentes	Usuarios participantes
Objetivos Asociados	<ul style="list-style-type: none"> <li>• OBJ-XX: Gestión de Usuarios</li> <li>• OBJ-XX: Gestión de Conjunto de datos</li> <li>• OBJ-XX: Gestión de Ejecuciones</li> </ul>
Descripción	El sistema debe estar diseñado para poder escalar fácilmente y permitir dar servicio a un mayor volumen de usuarios/peticiones.
Actores	Usuarios
Comentarios	-



## 2 ARQUITECTURA DE SOPORTE.

Hablaremos de los distintos frameworks de desarrollos utilizados en cada parte del sistema.

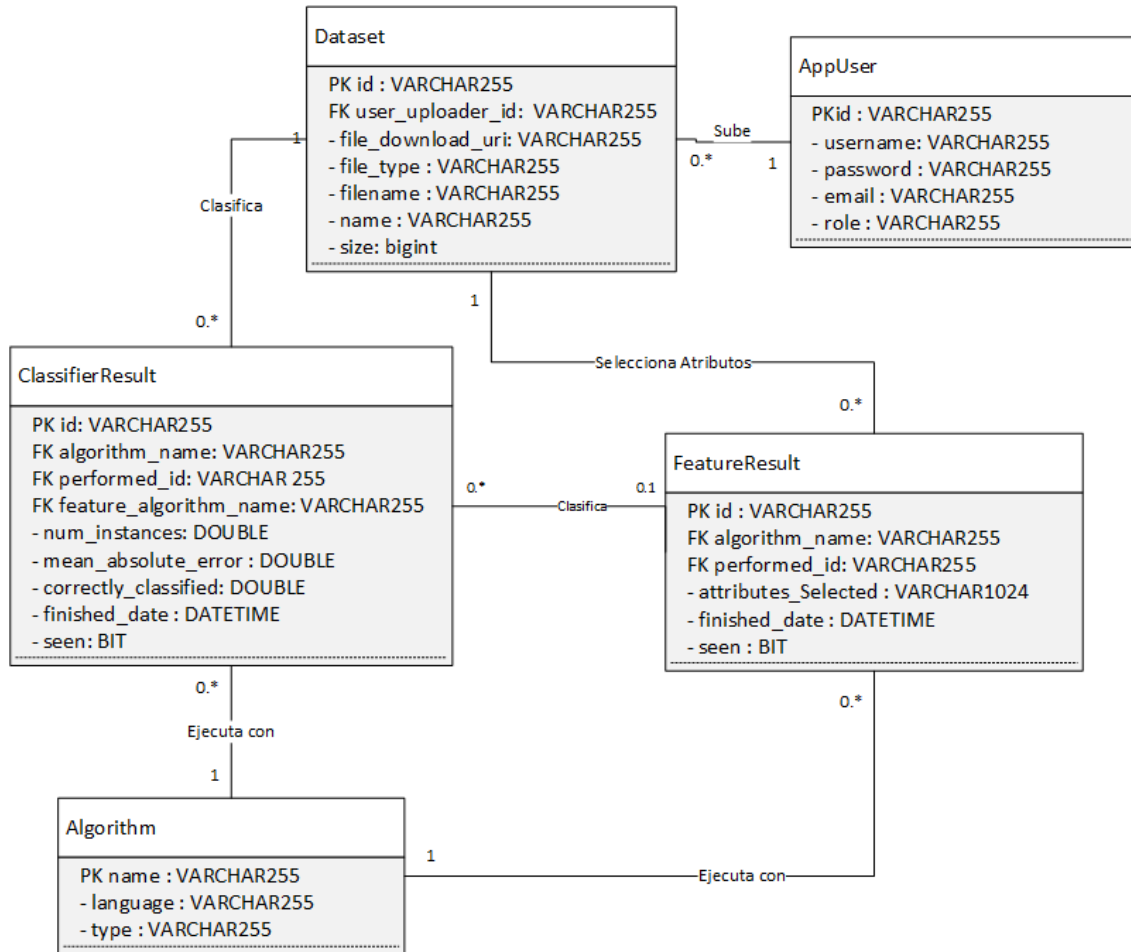
Servidor BackEnd: Desarrollado con Spring Framework un framework de desarrollo Java de código abierto, se implementa una aplicación Spring RESTFul API, con la que se da servicio a las aplicaciones de interfaces que se dan en el sistema, o por otro sistema externo.

FrontEnd Web: Desarrollado con Vue.JS un framework de desarrollo web de código abierto que utiliza el patrón MVVM modelo-vista-modelo de vista, sirve para construir interfaces de usuario en SPA (Single-page application) de manera que el usuario navega entre la misma página, pero va cambiando los elementos de manera fluida.

Aplicación móvil: Desarrollado con Nativescript, un framework para desarrollar aplicaciones en iOS y Android, se utiliza mediante código de Vue.js a partir de un complemento desarrollado por la comunidad. Estas aplicaciones son totalmente nativas. Esto hace que sea posible reusar código del FrontEnd Web.

### 3 MODELO FÍSICO DE DATOS.

#### 3.1 Diseño del Modelo físico de datos



Nombre	AppUser		
Descripción	Usuario de la Aplicación		
Atributos			
Campo	Tipo	Obligatorio	Descripción
Id	VARCHAR255	S	Identificador
username	VARCHAR255	S	Nombre de usuario
password	VARCHAR255	S	Contraseña de usuario
email	VARCHAR255	S	Correo del usuario
role	VARCHAR255	S	Rol del usuario
Clave Primaria			
Nombre	Columnas		
Id	Id		
Claves Únicas			
Nombre	Columnas		
username	username		
Restricciones			
Nombre	Columnas		Restricción
-	-		-

Nombre	Dataset		
Descripción	Conjunto de datos para procesar		
Atributos			
Campo	Tipo	Obligatorio	Descripción
Id	VARCHAR255	S	Identificador
file_download_uri	VARCHAR255	S	Ubicación del archivo
file_type	VARCHAR255	S	Tipo de archivo
filename	VARCHAR255	S	Nombre del archivo
name	VARCHAR255	N	Nombre del dataset
size	BIGINT	S	Tamaño del archivo
Clave Primaria			
Nombre	Columnas		
Id	Id		
Claves Únicas			
Nombre	Columnas		
file_download_uri	file_download_uri		
filename	nombre del archivo		
Claves Ajenas			
Nombre	Destino	Columnas	
user_uploader_id	AppUser.id	user_uploader_id	
Restricciones			
Nombre	Columnas	Restricción	
-	-	-	

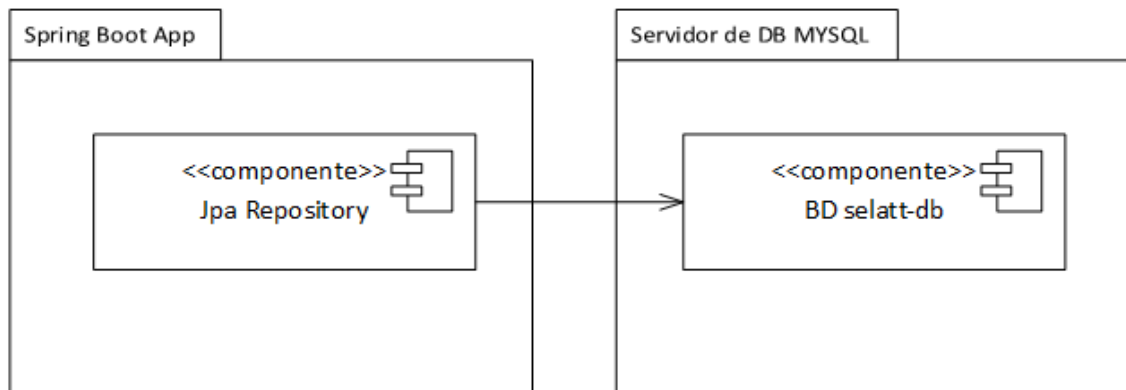
Nombre	ClassifierResult		
Descripción	Resultado de clasificador		
Atributos			
Campo	Tipo	Obligatorio	Descripción
Id	VARCHAR255	S	Identificador
num_instances	VARCHAR255	S	Número de instancias clasificadas
mean_absolute_error	VARCHAR255	S	Mediana de error absoluto
correctly_classified	VARCHAR255	S	Instancias correctamente clasificadas
finished_date	VARCHAR255	S	Fecha de finalización
seen	BIT	S	Indica si el resultado fue visto
Clave Primaria			
Nombre	Columnas		
Id	Id		
Claves Ajenas			
Nombre	Destino	Columnas	
performed_id	Dataset.id	performed_id	
algorithm_name	Algorithm.name	algorithm_name	
feature_algorithm_name	FeatureResult.id	feature_algorithm_name	
Restricciones			
Nombre	Columnas	Restricción	
-	-	-	

Nombre	FeatureResult		
Descripción	Resultado de Selección de atributos		
Atributos			
Campo	Tipo	Obligatorio	Descripción
Id	VARCHAR255	S	Identificador
attributes_selected	VARCHAR1024	S	Lista de atributos seleccionados
finished_date	VARCHAR255	S	Fecha de finalización
seen	BIT	S	Indica si el resultado fue visto
Clave Primaria			
Nombre	Columnas		
Id	Id		
Claves Ajenas			
Nombre	Destino	Columnas	
performed_id	Dataset.id	performed_id	
algorithm_name	Algorithm.name	algorithm_name	
Restricciones			
Nombre	Columnas	Restricción	
-	-	-	

Nombre	Algorithm		
Descripción	Algoritmo del sistema		
Atributos			
Campo	Tipo	Obligatorio	Descripción
name	VARCHAR255	S	Nombre del algoritmo
language	VARCHAR255	S	TODO
type	VARCHAR255	S	indica si es clasificador o selección de atributos
Clave Primaria			
Nombre	Columnas		
name	name		
Restricciones			
Nombre	Columnas		Restricción
-	-		-

### 3.2 Acceso a los datos.

El servidor BackEnd se conecta la base de datos MYSQL mediante la Api JPA en la capa de persistencia. Para cada entidad se dispondrá una interfaz extendida de CRUDRepository con la que se realizaran las consultas necesarias a la base de datos.



## 4 DISEÑO DE CASOS DE USO.

FrontEnd Web:

### 4.1 CASOS DE USO SUBSISTEMA DE USUARIO

<b>CU-01</b>	<b>Registro</b>	
<b>Descripción</b>	El sistema permitirá registrarse como usuarios a los invitados.	
<b>Actores</b>	Invitado	
<b>Precondición</b>	-	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El invitado solicita al sistema darse de alta
	<b>2</b>	El sistema redirige a IU-06 y pide al usuario que ingrese los datos necesarios
	<b>3</b>	El invitado ingresa los datos requeridos y pulsa el botón de registro.
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta aceptada del servidor BackEnd y acepta los datos y da de alta al invitado.
	<b>5</b>	Finaliza el caso de uso y redirige a IU-05.
<b>Flujos Alternativos</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta denegada del servidor BackEnd e indica que los datos introducidos no son correctos.
	<b>5</b>	El sistema pide al usuario que escriba los datos de nuevo mostrando un mensaje de error.
	<b>6</b>	Vuelve al paso 3 del flujo normal.
	<b>PASO</b>	<b>ACCIÓN</b>
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta denegada del servidor BackEnd debido a que ya existe un cliente con el mismo nombre de usuario.
	<b>5</b>	El sistema pide al usuario que escriba los datos de nuevo mostrando un mensaje de error.
	<b>6</b>	Vuelve al paso 3 del flujo normal.
<b>Observaciones</b>	Durante el proceso de registro el actor podrá cancelar el proceso.	

<b>CU-02</b>	<b>Login</b>
<b>Descripción</b>	El sistema permitirá logarse como usuarios a los invitados.
<b>Actores</b>	Invitado
<b>Precondición</b>	-
<b>Postcondición</b>	-
<b>Puntos Extensión</b>	-
<b>Activa</b>	

<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El invitado solicita al sistema iniciar sesión
	<b>2</b>	El sistema redirige a IU-05 y pide al usuario que ingrese el usuario y contraseña.
	<b>3</b>	El invitado ingresa los datos requeridos y pulsa el botón de Login.
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta aceptada del servidor BackEnd e inicia la sesión del usuario.
	<b>5</b>	Finaliza el caso de uso y redirige a IU-07.
<b>Flujos Alternativos</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta denegada del servidor BackEnd debido a que los datos introducidos no son correctos.
	<b>5</b>	El sistema pide al usuario que escriba los datos de nuevo mostrando un mensaje de error.
	<b>6</b>	Vuelve al paso 3 del flujo normal.
<b>Observaciones</b>	Durante el proceso de Login el actor podrá cancelar el proceso.	

<b>CU-03</b>	<b>LogOut</b>	
<b>Descripción</b>	El sistema permitirá cerrar sesión a los usuarios	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado.	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema cerrar sesión
	<b>2</b>	El sistema redirige a IU-07 y muestra el botón de Logout.
	<b>3</b>	El usuario y pulsa el botón de Login.
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta aceptada del servidor BackEnd y cierra la sesión del usuario.
	<b>5</b>	Finaliza el caso de uso y redirige a IU-05.
<b>Flujos Alternativos</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta denegada del servidor BackEnd debido a que los datos introducidos no son correctos.
	<b>5</b>	El sistema pide al usuario que escriba los datos de nuevo mostrando un mensaje de error.
	<b>6</b>	Vuelve al paso 3 del flujo normal.
<b>Observaciones</b>	Durante el proceso de Logout el actor podrá cancelar el proceso.	

#### 4.2 CASOS DE USO SUBSISTEMA DE DATASETS

<b>CU-04</b>	<b>Alta conjunto de datos</b>	
<b>Descripción</b>	El sistema permitirá subir conjuntos de datos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado.	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema dar de alta un conjunto de datos
	<b>2</b>	El usuario selecciona el archivo del dataset y pulsa el botón de Enviar.
	<b>3</b>	El sistema envía la petición con el archivo al servidor BackEnd, recibe la respuesta aceptada del servidor BackEnd y guarda el conjunto de datos, mostrando un mensaje de confirmación.
	<b>5</b>	Finaliza el caso de uso.
<b>Flujos Alternativos</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta denegada del servidor BackEnd debido a que el archivo no es correcto.
	<b>5</b>	El sistema pide al usuario que seleccione otro archivo de nuevo mostrando un mensaje de error.
	<b>6</b>	Vuelve al paso 3 del flujo normal.
<b>Observaciones</b>	-	

<b>CU-05</b>	<b>Listar conjunto de datos</b>	
<b>Descripción</b>	El sistema permitirá subir conjuntos de datos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado.	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema listar el conjunto de datos
	<b>2</b>	El usuario pulsa el botón de Datasets en la barra de navegación.
	<b>3</b>	El sistema se redirige a IU-08 envía la petición con el archivo al servidor BackEnd, recibe la respuesta aceptada del servidor BackEnd y carga los conjuntos de datos.
	<b>4</b>	Finaliza el caso de uso.
<b>Observaciones</b>	-	



#### 4.3 CASOS DE USO SUBSISTEMA DE EJECUCIONES

<b>CU-06</b>	<b>Seleccionar tipo de algoritmo</b>	
<b>Descripción</b>	El sistema permitirá subir conjuntos de datos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado.	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	CU-X+1	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema seleccionar el tipo de algoritmo a ejecutar.
	<b>2</b>	El sistema muestra los tipos de algoritmos posibles.
	<b>3</b>	El usuario pulsa el botón del tipo de algoritmo deseado
	<b>4</b>	El sistema envía la petición al servidor BackEnd, recibe la respuesta aceptada del servidor BackEnd y carga los conjuntos de datos para este tipo de algoritmo.
	<b>5</b>	Finaliza el caso de uso.
<b>Observaciones</b>	Solo habrá dos tipos de algoritmos: Los de selección de atributos y los clasificadores-	

<b>CU-07</b>	<b>Seleccionar conjunto de datos a procesar</b>	
<b>Descripción</b>	El sistema permitirá subir conjuntos de datos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	tipo de algoritmo seleccionado y conjunto de datos cargados	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema seleccionar el conjunto de datos a procesar.
	<b>2</b>	El sistema carga los conjuntos de datos disponibles.
	<b>3</b>	El usuario selecciona el conjunto de datos que desea procesar y pulsa continuar.
	<b>4</b>	El sistema guarda el conjunto de datos a procesar y carga los algoritmos que se podrán elegir para el conjunto de datos elegido.
	<b>5</b>	Finaliza el caso de uso.
<b>Observaciones</b>	-	

<b>CU-08</b>	<b>Seleccionar algoritmo a procesar</b>	
<b>Descripción</b>	El sistema permitirá subir conjuntos de datos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Algoritmo seleccionado y conjunto de datos cargados	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>

	<b>1</b>	El usuario solicita al sistema seleccionar el algoritmo a procesar.
	<b>2</b>	El sistema carga los posibles algoritmos a procesar.
	<b>3</b>	El usuario selecciona el algoritmo y pulsa Enviar.
	<b>4</b>	El sistema para el conjunto de datos elegido.
	<b>5</b>	Finaliza el caso de uso.
<b>Observaciones</b>	-	

<b>CU-09</b>	<b>Listar Resultados</b>	
<b>Descripción</b>	El sistema permitirá listar los resultados de las ejecuciones.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado.	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema listar los resultados
	<b>2</b>	El usuario pulsa el botón de Resultados en la barra de navegación.
	<b>3</b>	El sistema se redirige a IU-10 envía la petición con el archivo al servidor BackEnd, recibe la respuesta aceptada del servidor BackEnd y carga resultados.
	<b>4</b>	Finaliza el caso de uso.
<b>Observaciones</b>	-	

<b>CU-10</b>	<b>Consultar resultado de selección de atributos</b>	
<b>Descripción</b>	El sistema permitirá consultar resultados de selección de atributos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado.	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema consultar un resultado de selección de atributos.
	<b>2</b>	El sistema muestra los resultados de selección de atributos y un botón en cada uno para acceder a los detalles.
	<b>3</b>	El usuario pulsa el botón de uno de ellos para acceder a los detalles.
	<b>4</b>	El sistema muestra los detalles del resultado de selección de atributo.
	<b>5</b>	Finaliza el caso de uso.
<b>Observaciones</b>	-	

<b>CU-11</b>	<b>Consultar resultado de Clasificación</b>	
<b>Descripción</b>	El sistema permitirá subir conjuntos de datos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado.	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema consultar un resultado de clasificación.
	<b>2</b>	El sistema muestra los resultados de clasificación y un botón en cada uno para acceder a los detalles.
	<b>3</b>	El usuario pulsa el botón de uno de ellos para acceder a los detalles.
	<b>4</b>	El sistema muestra los detalles del resultado de clasificación.
	<b>5</b>	Finaliza el caso de uso.
<b>Observaciones</b>	-	

<b>CU-12</b>	<b>Descargar atributos seleccionados</b>	
<b>Descripción</b>	El sistema permitirá consultar resultados de selección de atributos.	
<b>Actores</b>	Usuarios	
<b>Precondición</b>	Usuario logado y resultado de selección de atributos seleccionado	
<b>Postcondición</b>	-	
<b>Puntos Extensión</b>	-	
<b>Activa</b>		
<b>Flujo normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	<b>1</b>	El usuario solicita al sistema descargar la lista de atributos de un resultado de selección de atributos.
	<b>2</b>	El sistema muestra un botón para descargar la lista en formato CSV.
	<b>3</b>	El usuario pulsa el botón de descargar.
	<b>4</b>	El sistema abre una ventana emergente para confirmar la descarga del archivo CSV.
	<b>5</b>	El confirma la operación.
	<b>6</b>	El sistema envía el archivo CSV al Usuario y finaliza el caso de uso.
<b>Observaciones</b>	-	

## 5 DISEÑO DE CLASES.

### Capa Controladora:

Capa encargada de separar la respuesta del servidor con el modelo, es la encargada de coordinar las peticiones y la respuesta de estas.

<b>Clase-01: AuthController</b>	
<b>Descripción</b>	Clase controladora que recibe las peticiones del servidor relacionadas con la autenticación.
<b>Comentarios</b>	

<b>Clase-02: EvaluationController</b>	
<b>Descripción</b>	Clase controladora que recibe las peticiones del servidor relacionadas con las clasificaciones.
<b>Comentarios</b>	

<b>Clase-03: FeatureSelectionController</b>	
<b>Descripción</b>	Clase controladora que recibe las peticiones del servidor relacionadas con la selección de atributos.
<b>Comentarios</b>	

<b>Clase-04: FileManagerController</b>	
<b>Descripción</b>	Clase controladora que recibe las peticiones del servidor relacionadas con la subida de datasets al servidor.
<b>Comentarios</b>	

### Capa de servicios:

Capa encargada de encapsular la lógica de negocio de la aplicación, centralizar el acceso a los datos , definen las transacciones con la BBDD y coordinan las operaciones del sistema.

<b>Clase-05: AuthService</b>	
<b>Descripción</b>	Interfaz de los servicios que ofrece la autenticación.
<b>Comentarios</b>	-

<b>Clase-06: EvaluationService</b>	
<b>Descripción</b>	Interfaz de los servicios que ofrece la clasificación.
<b>Comentarios</b>	-

<b>Clase-07: FeatureSelectionService</b>	
<b>Descripción</b>	Interfaz de los servicios que ofrece la selección de atributos.
<b>Comentarios</b>	-

<b>Clase-08: FileStorageService</b>	
<b>Descripción</b>	Interfaz de los servicios que ofrece la gestión de archivos.
<b>Comentarios</b>	-

<b>Clase-09: LoadData</b>	
<b>Descripción</b>	Interfaz de los servicios que ofrece la gestión de datasets.
<b>Comentarios</b>	-

<b>Clase-10: AuthServiceImpl</b>	
<b>Descripción</b>	Clase que implementa los servicios que ofrece la autenticación.
<b>Comentarios</b>	-

<b>Clase-11: EvaluationServiceImpl</b>	
<b>Descripción</b>	Clase que implementa los servicios que ofrece la clasificación.
<b>Comentarios</b>	-

<b>Clase-12: FeatureSelectionServiceImpl</b>	
<b>Descripción</b>	Clase que implementa los servicios que ofrece la selección de atributos.
<b>Comentarios</b>	-

<b>Clase-13: FileStorageServiceImpl</b>	
<b>Descripción</b>	Clase que implementa los servicios que ofrece la gestión de archivos.
<b>Comentarios</b>	-

<b>Clase-14: LoadDataImpl</b>	
-------------------------------	--

<b>Descripción</b>	Clase que implementa los servicios que ofrece la gestión de datasets.
<b>Comentarios</b>	-

#### Capa JPA -Acceso BBDD:

Capa encargada del acceso a la base de datos para la persistencia del sistema y las consultas necesarias para el funcionamiento del sistema.

<b>Clase-15: AppUserRepository</b>	
<b>Descripción</b>	Interfaz que implementa las operaciones básicas de gestión de la entidad AppUser.
<b>Comentarios</b>	-

<b>Clase-16: ClassifierResultRepository</b>	
<b>Descripción</b>	Interfaz que implementa las operaciones básicas de gestión de la entidad ClassifierResult.
<b>Comentarios</b>	-

<b>Clase-17: DatasetRepository</b>	
<b>Descripción</b>	Interfaz que implementa las operaciones básicas de gestión de la entidad Dataset.
<b>Comentarios</b>	-

<b>Clase-18: FeatureResultRepository</b>	
<b>Descripción</b>	Interfaz que implementa las operaciones básicas de gestión de la entidad FeatureResult.
<b>Comentarios</b>	-

<b>Clase-19: AlgorithmRepository</b>	
<b>Descripción</b>	Interfaz que implementa las operaciones básicas de gestión de la entidad Algorithm.
<b>Comentarios</b>	-

#### Capa de modelos:

Es la capa encargada de representar la información con la que opera el sistema gestiona las operaciones que se realizan sobre ellos y configura el mapeo de una tabla en la base de datos por cada modelo.

<b>Clase-20: AppUser</b>
--------------------------

<b>Descripción</b>	Representa la información del usuario que interactúa con el sistema.
<b>Comentarios</b>	-

#### **Clase-21: ClassifierResult**

<b>Descripción</b>	Representa el resultado de realizar una clasificación sobre un dataset, ya sea filtrado o no.
<b>Comentarios</b>	-

#### **Clase-22: FeatureResult**

<b>Descripción</b>	Representa el resultado de realizar una selección de atributos sobre un dataset.
<b>Comentarios</b>	-

#### **Clase-23: Dataset**

<b>Descripción</b>	Representa el conjunto de datos.
<b>Comentarios</b>	-

#### **Clase-21: Algorithm**

<b>Descripción</b>	Representa el algoritmo que se puede utilizar en el sistema de información.
<b>Comentarios</b>	-

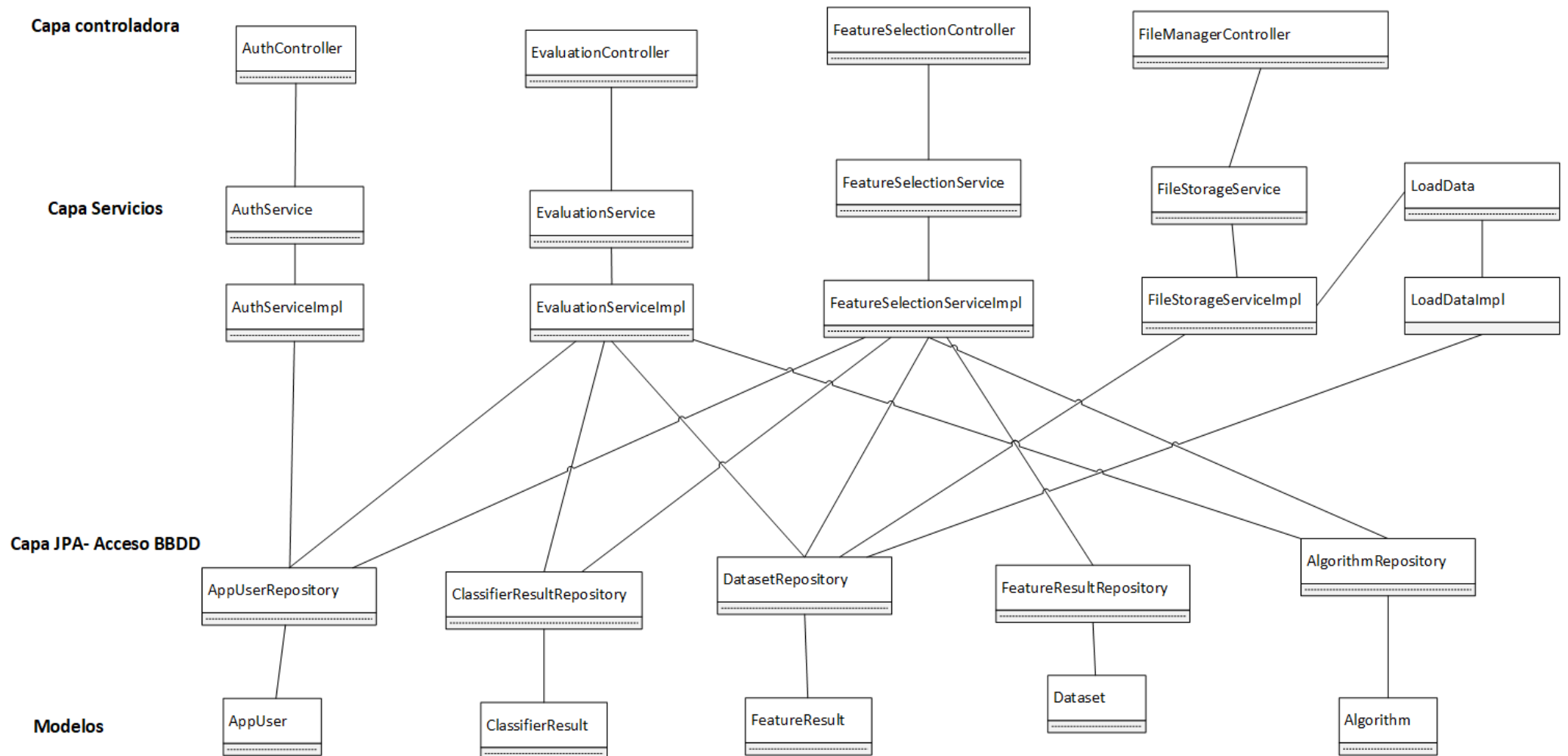


Ilustración 1 Diagrama de clases



## 6 DISEÑO DE INTERFACES.

Las interfaces se diseñarán teniendo en cuenta que se realizarán con material Design.

Interfaz del módulo IU-01 Landing page

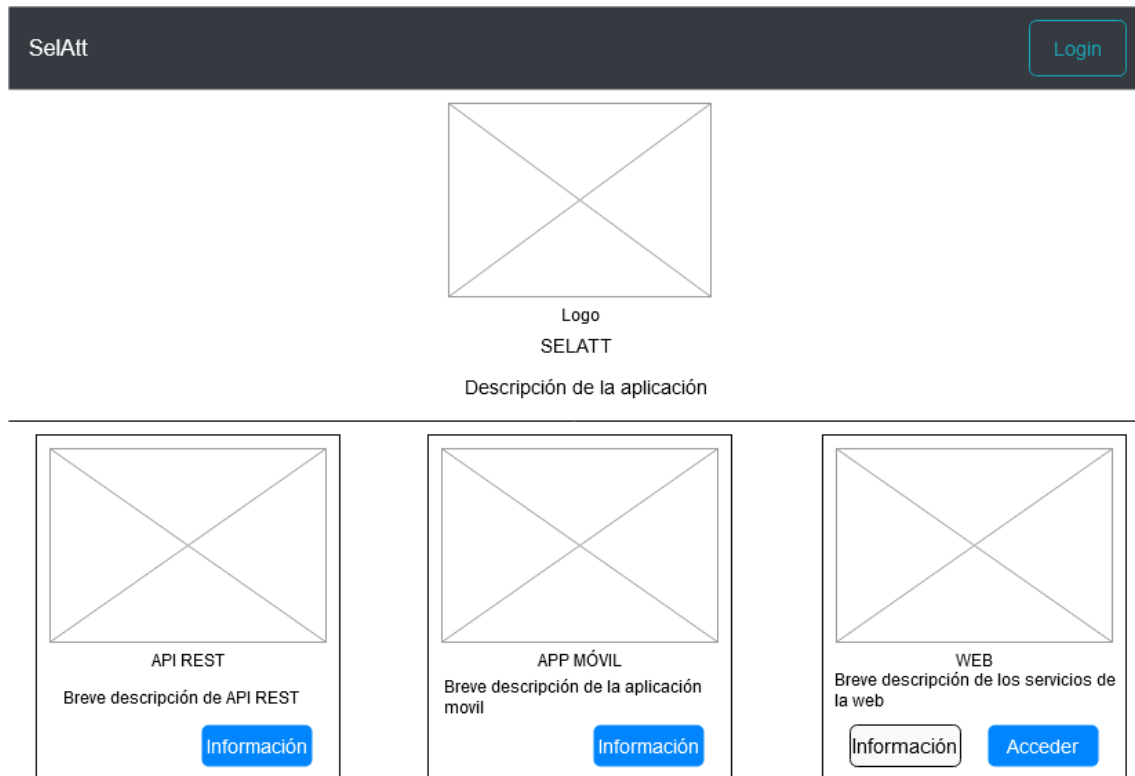


Ilustración 2 Interfaz IU-01 Landing page

IU-01 Landing page		
<b>Descripción</b>	Interfaz de la que partirá el usuario, se presentará el sistema brevemente y varios botones con los que navegar.	
<b>Botones/Enlaces</b>	<b>Nombre</b>	<b>Acción</b>
	Login	Al ser pulsado, el usuario podrá iniciar sesión en la Interfaz IU-05.
	Información – API REST	Al ser pulsado, el usuario será redirigido a la IU-02.
	Información – APP móvil	Al ser pulsado, el usuario será redirigido a la IU-04.
	Información – Web	Al ser pulsado, el usuario será redirigido a la IU-03.
	Acceder – Web	Al ser pulsado, el usuario será redirigido a la IU-09 en caso de estar logado, de lo contrario irá a la IU-05.

**API PUBLICA**

Descripción de la API breve

**Indice**

1. Autenticación
2. Datasets
3. Selección de atributos
4. Clasificadores

**Lista de metodos por apartado**

Metodo http : /url	Descripción metodo
Tabla de parametros	
Nombre	Descripción
Campo	Descripción breve del campo y ejemplo Json
Tabla de respuestas	
Nombre	Descripción
Campo	Descripción breve del campo y ejemplo Json

*Ilustración 3 Interfaz IU-02 Api Docs Page*

IU-02 API Docs Page		
<b>Descripción</b>	Interfaz en la que el invitado podrá consultar la información relevante del uso de la Api Pública, donde se listaran los distintos métodos disponibles, los parámetros necesarios y las posibles respuestas.	
<b>Botones/Enlaces</b>	<b>Nombre</b>	<b>Acción</b>
	Login	Botón que iniciara la sesión del usuario con los datos proporcionados. Redirige a la IU-07.
	Registro	Al ser pulsado, el usuario será redireccionado a la IU-06.
	SelAtt - Navbar	Al ser pulsado, el usuario será redireccionado a la IU-01.

**Primeros pasos**

Necesario crear una cuenta

[Crear cuenta](#)

Información de tipos de archivo aceptados

[Subir Archivo](#)

Descripción de algoritmos

[Algoritmos](#)

Lista de algoritmos disponibles

Descripción breve de Weka

[Weka](#)

Descripción de resultados

[Resultados](#)

Ilustración 4 Interfaz IU-03 Web Docs Page

IU-03 Web Docs Page		
<b>Descripción</b>	Interfaz en la que el invitado podrá consultar una guía de uso de la página web en la que se listaran los servicios disponibles.	
<b>Botones/Enlaces</b>	<b>Nombre</b>	<b>Acción</b>
	Crear Cuenta	Al ser pulsado, el usuario será redireccionado a la IU-06.
	Subir Archivo	Al ser pulsado, el usuario será redireccionado a la IU-08 en caso de estar logado, de lo contrario navegará a la IU-05.
	Algoritmos	Al ser pulsado, el usuario será redireccionado a la IU-09 en caso de estar logado, de lo contrario navegará a la IU-05.
	Weka	Al ser pulsado, el usuario será redireccionado a la página web de Weka.
	Resultados	Al ser pulsado, el usuario será redireccionado a la IU-10 en caso de estar logado, de lo contrario navegará a la IU-05.
	Login	Al ser pulsado, el usuario será redireccionado a la IU-05.
	Registro	Al ser pulsado, el usuario será redireccionado a la IU-06.
	SelAtt - Navbar	Al ser pulsado, el usuario será redireccionado a la IU-01.

**Aplicación móvil**

Descripción de la App

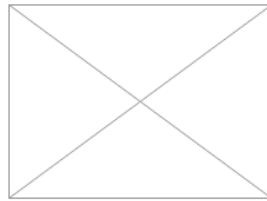
Instalación de apk



Capturas de la aplicación

[Descargar](#)

IU-04 Mobile Docs Page		
<b>Descripción</b>	Interfaz en la que el invitado podrá observar un primer vistazo de la aplicación y un enlace para descargarla.	
<b>Botones/Enlaces</b>	<b>Nombre</b>	<b>Acción</b>
	Descargar	Al ser pulsado, el usuario será redireccionado a la página de releases del repositorio en GitHub, donde se encuentra la descarga de la aplicación.
	Login	Al ser pulsado, el usuario será redireccionado a la IU-05.
	Registro	Al ser pulsado, el usuario será redireccionado a la IU-06.
	SelAtt - Navbar	Al ser pulsado, el usuario será redireccionado a la IU-01.



Logo

Usuario

Contraseña

Login

Registro

**IU-05 Login**

<b>Descripción</b>	Interfaz en la que el invitado podrá iniciar sesión como usuario introduciendo su usuario y contraseña.				
<b>Campos</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Editable</b>	<b>Oblig.</b>	<b>Descripción</b>
	Usuario	String	Editable	Si	Usuario del cliente que desea logarse.
	Contraseña	String	Editable	Si	Contraseña del usuario que desea logarse.
<b>Botones/Enlaces</b>	<b>Nombre</b>	<b>Acción</b>			
	Login	Botón que iniciara la sesión del usuario con los datos proporcionados. Redirige a la IU-07.			
	Registro	Al ser pulsado, el usuario será redireccionado a la IU-06.			
	SelAtt - Navbar	Al ser pulsado, el usuario será redireccionado a la IU-01.			

**Registro**

Usuario

Correo




Contraseña

Registrarse

Login

IU-06 Register					
<b>Descripción</b>	Interfaz en la que el invitado podrá darse de alta en el sistema introduciendo un usuario, correo y contraseña.				
<b>Campos</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Editable</b>	<b>Oblig.</b>	<b>Descripción</b>
	Usuario	String	Editable	Si	Usuario del usuario que desea logarse.
	Correo	String	Editable	Si	Correo del usuario que desea logarse.
	Contraseña	String	Editable	Si	Contraseña del usuario que desea logarse.
<b>Botones/Enlaces</b>	<b>Nombre</b>	<b>Acción</b>			
	Login	Al ser pulsado, el usuario será redireccionado a la IU-05.			
	Registrarse	Botón que al pulsar dará de alta en el sistema al invitado como usuario.			
	SelAtt - Navbar	Al ser pulsado, el usuario será redireccionado a la IU-01.			

**Datasets**


Dataset	Size	Tipo archivo
 iris.arff	209 Kb	arff
 sample_data.csv	197 Kb	text/csv
 supermarket.arff	360 Mb	arff

Subir dataset

Subir

Seleccionar archivo

IU-08 Datasets					
<b>Descripción</b>	Interfaz en la que el usuario podrá consultar los datasets subidos, y subir nuevos.				
<b>Campos</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Editable</b>	<b>Oblig.</b>	<b>Descripción</b>
	Archivo	Archivo	Editable	Si	Archivo que el usuario desea subir.
	Filtro	String	Editable	Si	Palabra por la que se filtraran los datasets mostrados.
	Lista de datasets	String	Consulta	No	Listado de datasets que muestra el nombre, el tamaño y el tipo de archivo para cada uno de los datasets registrados.
<b>Botones/Enlaces</b>	<b>Nombre</b>		<b>Acción</b>		
	Subir		Botón que subirá el archivo seleccionado a el servidor.		
	SelAtt - Navbar		Al ser pulsado, el usuario será redireccionado a la IU-01.		
	Resultados - Navbar		Al ser pulsado, el usuario será redireccionado a la IU-10.		
	Algoritmos – Navbar		Al ser pulsado, el usuario será redireccionado a la IU-09.		
	User – Navbar		Al ser pulsado, el usuario será redireccionado a la IU-07.		

SelAtt
Datasets
Resultados
Algoritmos
\*Usuario\*

### Algoritmos

☒ Clasificadores  
☐ Selección de atributos

Subir dataset

Subir
Seleccionar archivo

Seleccionar dataset

Dataset a procesar

Seleccione un algoritmo

Algoritmo a procesar

Enviar
Cancelar

Respuesta de la petición.

IU-09 Algoritmos					
<b>Descripción</b>	Interfaz en la que el usuario podrá consultar los datasets subidos, y subir nuevos.				
<b>Campos</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Editable /Consulta</b>	<b>Oblig.</b>	<b>Descripción</b>
	Tipo de Algoritmo	Booleano	Editable	Si	Indicara si es selección de atributos en caso de ser verdadero y clasificador en caso contrario.
	Dataset	Archivo	Editable	No	Archivo para subir en caso de ser necesario.
	Dataset	String	Editable	Si	Nombre del dataset a utilizar en el procesamiento.
	Algoritmo	String	Editable	Si	Nombre del algoritmo a usar en el procesamiento.
	Respuesta	String	Consulta	No	Respuesta del servidor BackEnd de la petición.
<b>Botones/Enlaces</b>	<b>Nombre</b>		<b>Acción</b>		
	Subir		Botón que subirá el archivo seleccionado a el servidor.		
	Enviar		Botón que enviara la petición al servidor BackEnd con los parámetros elegidos.		
	Cancelar		Botón que reinicia la operación desde la selección del tipo de algoritmo.		
	SelAtt - Navbar		Al ser pulsado, el usuario será redireccionado a la IU-01.		
	Resultados - Navbar		Al ser pulsado, el usuario será redireccionado a la IU-10.		
	Algoritmos – Navbar		Al ser pulsado, el usuario será redireccionado a la IU-09.		
	User – Navbar		Al ser pulsado, el usuario será redireccionado a la IU-07.		






## Resultados

## Clasificaciones

## Selección de atributos

 Filtrar

Dataset	Algoritmo	Fecha	Porcentaje de acierto	Info
 iris.arff/Ranker	Naive Bayes	Sabado 22 de Febrero	96,4%	<a href="#">Ver</a>
 sample_data.csv	Multi Layer Perceptron	Lunes 24 de Febrero	45%	<a href="#">Ver</a>
 supermarket.arff	lbk	Miercoles 26 de Febrero	67.4%	<a href="#">Ver</a>

## Clasificador: Naive Bayes

Dataset "Iris.arff" de 7Kb.  
Filtrado con :Ranker  
Numero de instancias:  
Instancias clasificadas correctamente:  
Porcentaje de acierto: 96,4%  
Mediana de error absoluto:

[Cerrar](#)

## Selección de atributos: Ranker

Dataset "Iris.arff" de 7Kb.  
Atributos: '0,1,2,3'

[Cerrar](#)[Descargar CSV](#)

IU-10 Resultados					
<b>Descripción</b>	Interfaz en la que el usuario podrá consultar los resultados obtenidos.				
<b>Campos</b>	<b>Nombre</b>	<b>Tipo</b>	<b>Editable /Consulta</b>	<b>Oblig.</b>	<b>Descripción</b>
	Filtro	String	Editable	Si	Palabra por la que filtrara los resultados.
	Lista de Resultados	String	Consulta	No	Lista de resultados previos.
	Resultado de clasificación	String	Consulta	No	Detalles del resultado de clasificación.
<b>Botones/Enlaces</b>	<b>Nombre</b>		<b>Acción</b>		
	Clasificaciones		Botón que hará que se muestre el listado de resultados de clasificaciones.		
	Selección de atributos		Botón que hará que se muestre el listado de resultados de selección de atributos.		
	Ver - Resultado		Botón que seleccionara un resultado para mostrar los detalles, dependerá del tipo los detalles que se mostraran.		
	Cerrar – Detalles		Botón que cerrara los detalles del resultado elegido.		
	Descargar CSV – Detalles		Botón que iniciara la descarga del archivo CSV que contiene una lista de los atributos obtenidos.		
	SelAtt - Navbar		Al ser pulsado, el usuario será redireccionado a la IU-01.		
	Resultados - Navbar		Al ser pulsado, el usuario será redireccionado a la IU-10.		
	Algoritmos – Navbar		Al ser pulsado, el usuario será redireccionado a la IU-09.		
	User – Navbar		Al ser pulsado, el usuario será redireccionado a la IU-07.		

## 7 ESPECIFICACIONES DE CONSTRUCCIÓN.

Durante el desarrollo se podrá utilizar cualquier IDE de desarrollo debido a que el proyecto está mavenizado con las dependencias necesarias para funcionar tanto en desarrollo como en producción, es por esto por lo que existen dos perfiles distintos, el por defecto usado para producción y el de prueba para desarrollo. Las tablas de la base de datos se montan mediante el componente de Spring Boot de JPA sin embargo en producción, aunque se conecte mediante este componente es necesario la creación previa de una secuencia SQL para la importación de la estructura de la base de datos, en desarrollo la estructura puede ir cambiando mediante las anotaciones en los modelos del servidor BackEnd. Para desplegar el sistema se hará uso de Docker para la base de datos, en producción también se desplegará el sistema BackEnd en Docker y la aplicación web se desplegará por separado ligado a las releases en la rama master de GitHub.

## 8 PLAN DE PRUEBAS.

En este punto se describirá el plan de pruebas realizado para comprobar el funcionamiento y cumplimiento de requisitos no funcionales del sistema.

Las pruebas que se van a realizar sobre el sistema son:

- **Pruebas Unitarias:** Deben realizarse pruebas unitarias sobre los componentes que realizan operaciones críticas, sobre todo las relacionadas con la conexión entre el FrontEnd y el BackEnd. Los elementos que serán objeto de pruebas unitarias serán los relacionados con ejecutar procesamientos de algoritmos, la subida de datasets y la consulta de resultados, estas pruebas se realizarán de manera manual con una tabla que describa los pasos a realizar, los resultados esperados y los obtenidos.
- **Pruebas de integración:** Se realizarán pruebas de integración sobre cada subsistema verificando que la integración entre sistemas funciona correctamente. Se detallarán los resultados obtenidos en una tabla
- **Pruebas de seguridad:** Se realizarán pruebas de seguridad para comprobar que el sistema es seguro ante ciertos aspectos. Se comprobará el posible acceso a los métodos sin el token de autenticación, el acceso a usuarios mediante el hash de una contraseña de la Base de datos, para comprobar que en posible caso de inyección SQL el acceso al sistema de otros usuarios sea imposible.

## 9 REQUISITOS DE IMPLANTACIÓN.

### 9.1 REQUISITOS DE DOCUMENTACIÓN

La documentación necesaria para operar con el Sistema BackEnd se encuentra dentro de la propia documentación incluida en la página web accesible para los invitados del sistema, en la que se desarrollara una guía de primeros pasos del sistema, Una guía de los métodos disponibles de la API, una pequeña guía de la aplicación móvil y el acceso de la descarga de esta.

### 9.2 REQUISITOS DE IMPLANTACIÓN

Los Requisitos de implantación del sistema son varias opciones:

1. **Docker:** Tener instalado Docker en el equipo, por el cual se podrá levantar el sistema BackEnd y la base de dato.

2. **Maven, java y MySQL:** Tener instalado Java, Maven con la que se podrá levantar el sistema BackEnd con Maven y un servidor MySQL.

Para la aplicación web será necesario tener instalado NodeJs, aunque en futuras versiones se creara la posibilidad de incluir la aplicación web en un contenedor Docker.

## 4. CODIFICACIÓN

En este capítulo se describirán brevemente los componentes del sistema para entender mejor la codificación y el aspecto tecnológico de estos.

La codificación del sistema se encuentra en un repositorio público de GitHub, en el siguiente enlace: <https://github.com/Kazzpa/tfg-kazzpa>, además de entregarse adjunto un archivo comprimido zip que contiene los ejecutables necesarios para desplegar la aplicación.

### 4.1 SERVIDOR BACKEND

La base del sistema como es la lógica de negocio y las principales operaciones están gestionadas por un servidor BackEnd desarrollado con Spring Boot. Este framework de Java provee robustez al sistema, una capa de seguridad completamente configurable y la creación efectiva de una API pública en la que mediante las anotaciones permiten configurar el comportamiento de los métodos. Ofrece una arquitectura en la que las peticiones pasan primero por una controladora, asistida por la capa de seguridad que verifica que las peticiones estén autenticadas según la configuración, estas pasan a la capa de servicios que realiza las operaciones necesarias que utiliza la capa de persistencia y devuelve un resultado con las cabeceras http necesarias.

### 4.2 APLICACIÓN WEB

La aplicación Web esta desarrollada sobre un framework de JavaScript que permite desarrollar aplicaciones de pagina única, es decir que todos los códigos de HTML, Javascript y CSS se cargan una única vez, los recursos necesarios van cargando dinámicamente cuando sea necesario según la respuesta del Usuario. Uno de los principales objetivos de la aplicación web era que fuese fácilmente usable por el usuario por lo que se ha apoyado en un framework de componentes como es Vuetify, con un diseño Material Design que permite un ligero desarrollo con multitud de componentes configurables con los que la aplicación web puede contar.

La aplicación web ha de consumir la API pública del servidor BackEnd por lo que existía la necesidad de utilizar un gestor de eventos que nos ayudase a gestionar dentro de la aplicación las distintas peticiones a la API y actualizar la página cuando obtuviésemos un resultado de esta, para esto también nos apoyamos sobre el cliente HTTP axios, que ayuda a configurar las peticiones a la API de manera sencilla y lógica.

Esta aplicación se desarrollo en distintos componentes que son las interfaces de la vista, y unos componentes comunes a todas las vistas como son la cabecera y el pie de página. Se utilizaron cookies para facilitar el uso de la aplicación, como no exigir en cada momento el usuario y contraseña, si no que si se encuentran en las cookies se inicia sesión automáticamente en el sistema.

### 4.3 APLICACIÓN MÓVIL

La aplicación móvil tenia como objetivo principal que los usuarios pudiesen acceder desde cualquier dispositivo y que pudiese seguir una lógica similar a la de la aplicación web, es por es que se ha hecho uso del Framework de Nativescript que ofrece la posibilidad de construir aplicaciones nativas para Android e iOS. Ofrece un aprendizaje rápido debido a que se construye mediante unas plantillas mediante etiquetas propias de la interfaz de Nativescript similares a los de HTML, CSS para el estilo y Javascript para el procesamiento de la aplicación. Por parte de la comunidad ofrecían un plugin de Vue por lo que se puede utilizar el potencial de Vue, sin embargo, al ser un plugin no permite la migración total de un proyecto web a un proyecto de aplicación móvil, hay que hacer muchos cambios en cuanto a el tratamiento de

estados, el tratamiento del enrutador entre distintas vistas debido a que ya no se trata de una aplicación de página única SPA. La forma en la que se consume la API del servidor BackEnd es muy similar a la de la aplicación Web, haciendo uso del gestor de estados Vuex, con ciertos cambios, y el cliente HTTP axios.

## 5. EXPERIMENTACIÓN

En este capítulo se realizará la experimentación de unos conjuntos de datos biomédicos en el sistema, realizando ciertas selecciones de atributos y usando varios clasificadores para observar los resultados obtenidos.

### 5.1 DATOS DE ESTUDIO

Los datos utilizados para la experimentación son datos biomédicos que son públicos, estos datos tienen relación con enfermedades. Se trabajará con 4 datasets distintos de las enfermedades: diabetes, linfoma, cáncer de mama, cáncer de ovario, son datasets de tamaño variable que contienen una gran cantidad de atributos. La fuente de los datasets está en la bibliografía.

Dataset de diabetes: Se trata de un dataset que contiene 8 atributos y la clase.

Dataset de linfoma: Se trata de un dataset que contiene 4026 atributos y la clase.

Dataset de cáncer de mama: Se trata de un dataset que contiene 24481 atributos y la clase.

Dataset de cáncer de ovario: Se trata de un dataset que contiene 15154 atributos y la clase.

### 5.2 ESPECIFICACIONES DEL EXPERIMENTO

Para el experimento se subirán los datasets al sistema y se ejecutaran distintos algoritmos de selección de atributos y se procederá a analizar mediante clasificadores el resultado obtenido, además de comparar el número de atributos con el que se ha obtenido el resultado de cada clasificación.

### 5.3 RESULTADOS OBTENIDOS

En este apartado se expondrán tablas en las que se representarán los resultados obtenidos por cada dataset:

Diabetes:

Filtrado	Clasificación	Atributos	Tasa de acierto
Sin filtrar	Bayes Net	8	75.78%
Sin filtrar	Naive Bayes	8	75%
Sin filtrar	IBk	8	70.96%
VNS	Naive Bayes	3	68.36%
SCS	Bayes Net	3	68.23%
SCS	Naive Bayes	3	68.1%
VNS	Bayes Net	3	67.71%

Se obtienen buenos resultados sin filtrar, como podemos comprobar al tratarse de un dataset con un número pequeño de atributos no se obtiene una mejoría en la predicción de estos, ni supondría una mejora sustancial en el tiempo de procesamiento de un dataset de estas características.

Linfoma:

Filtrado	Clasificación	Atributos	Tasa de acierto
Sin filtrar	Bayes Net	4026	98.48%
Sin filtrar	IBk	4026	98.48%
FCBF	IBk	110	96.97%

FCBF	Naive Bayes	110	92.42%
Sin filtrar	Naive Bayes	4026	92.42%
FCBF	Bayes Net	110	92.42%
Sin filtrar	Bayes Net	4026	98.48%

Se puede observar como al reducirse los atributos se siguen obteniendo buenos resultados parecidos a el dataset original, por lo que podría suponer una ventaja trabajar con un dataset que solo tenga en cuenta estos atributos, se trata de un porcentaje de disminución de los atributos de un 97.27%.

Cáncer de mama:

Filtrado	Clasificación	Atributos	Tasa de acierto
FCBF	IBk	92	60.82%
Sin filtrar	IBk	24481	59.79%
FCBF	Naive Bayes	92	52.58%
FCBF	Bayes Net	92	52.58%
Sin filtrar	Naive Bayes	24481	50.52%
FCBF	IBk	92	60.82%

Se obtienen incluso mejores resultados al reducir el número de atributos, y podemos observar que se obtienen mejores resultados al reducir significativamente este conjunto de atributos, el porcentaje de disminución de atributos se trata del 99.62%.

Cáncer de ovario:

Filtrado	Clasificación	Atributos	Tasa de acierto
Sin filtrar	IBk	15154	94.86%
FCBF	IBk	30	94.47%
Sin filtrar	Naive Bayes	15154	93.28%
FCBF	Naive Bayes	30	92.49%

Se obtienen peores resultados al reducir el número de atributos, pero apenas se pierde un 1% de acierto, por lo que se pueden entender que se ha seleccionado un buen subconjunto de atributos. El porcentaje de disminución de atributos es del 99.80%.

## 5.4 CONCLUSIONES

Podemos concluir que al aplicar algoritmos de selección de atributos se puede reducir enormemente la dimensionalidad de estos y se puede acelerar el tiempo de entrenamiento de un modelo de predicción, se puede disminuir el sobre ajuste a los atributos. Sin embargo, hay que tener en cuenta que esto tiene relación con el número de atributos inicial. Es por esto por lo que en el dataset de la diabetes no se mejoran los resultados ni supone un problema trabajar con 8 atributos. En los otros datasets podemos ver como la tasa de acierto se mantiene e incluso se mejora en algún caso y se reducen hasta un 99% el número de atributos a utilizar.



## 6. PRUEBAS

En este capítulo se hablará de las pruebas que se han realizado sobre el sistema para comprobar que cumple los requisitos funcionales y no funcionales, detallando los resultados esperados, los obtenidos y las posibles correcciones realizadas.

### 6.1 PRUEBAS UNITARIAS

En estas pruebas se realizarán pruebas a métodos concretos en los que se detallara una descripción del proceso, los pasos a realizar, el resultado esperado, el obtenido y comentarios.

PU-01: Iniciar sesión	
Descripción	Se realizará el proceso de inicio de sesión en el sistema
Prerrequisitos	-
Pasos	<ol style="list-style-type: none"><li>1. Navegar a pantalla de LogIn.</li><li>2. Introducir los campos de usuario y contraseña.</li><li>3. Pulsar el botón de LogIn.</li></ol>
Resultado esperado	Se inicia la sesión del usuario correctamente.
Resultado Obtenido	El resultado esperado.

PU-02: Registro	
Descripción	Se realizará el proceso de registro en el sistema de un usuario nuevo
Prerrequisitos	-
Pasos	<ol style="list-style-type: none"><li>1. Navegar a pantalla de registro.</li><li>2. Introducir los campos de usuario, email y contraseña.</li><li>3. Pulsar el botón de Registro.</li></ol>
Resultado esperado	Se da de alta el usuario en el sistema correctamente.
Resultado Obtenido	El resultado esperado.

PU-03: Subida de dataset	
Descripción	Se realizará el proceso seleccionar un archivo para subir a el sistema.
Prerrequisitos	Sesión iniciada.
Pasos	<ol style="list-style-type: none"><li>1. Navegar a pantalla de Datasets.</li><li>2. Seleccionar un archivo para subir de la lista de formatos permitidos.</li><li>3. Pulsar el botón de Subir.</li></ol>
Resultado esperado	El dataset es cargado con éxito en el sistema y se muestra un mensaje de confirmación.
Resultado Obtenido	El resultado esperado.

PU-04: Solicitar selección de atributos de un dataset	
Descripción	Se realizará el proceso de solicitar una selección de atributos sobre un dataset.
Prerrequisitos	Sesión iniciada. Dataset cargado previamente

<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Navegar a pantalla de Algoritmos.</li> <li>2. Seleccionar selección de atributos.</li> <li>3. Seleccionar un datasets de los cargados previamente.</li> <li>4. Seleccionar un algoritmo de los disponibles.</li> <li>5. Pulsar el botón de Enviar.</li> </ol>
<b>Resultado esperado</b>	Se realiza la selección de atributos sobre el dataset y se devuelve la respuesta de confirmación del servidor BackEnd además de una notificación de que la selección ha finalizado.
<b>Resultado Obtenido</b>	El resultado esperado.

<b>PU-05: Solicitar clasificación de un dataset</b>	
<b>Descripción</b>	Se realizará el proceso solicitar una clasificación sobre un dataset.
<b>Prerrequisitos</b>	Sesión iniciada. Dataset cargado previamente
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Navegar a pantalla de Algoritmos.</li> <li>2. Seleccionar Clasificadores.</li> <li>3. Seleccionar un datasets de los cargados previamente.</li> <li>4. Seleccionar un algoritmo de los disponibles.</li> <li>5. Pulsar el botón de Enviar.</li> </ol>
<b>Resultado esperado</b>	Se realiza la clasificación sobre el dataset y se devuelve la respuesta de confirmación del servidor BackEnd además de una notificación de que la clasificación ha finalizado.
<b>Resultado Obtenido</b>	El resultado esperado.

<b>PU-06: Filtrar resultados por nombre de un dataset</b>	
<b>Descripción</b>	Se realizará el proceso de filtrar la lista de resultados mediante el campo de un dataset.
<b>Prerrequisitos</b>	Sesión iniciada. Resultados realizados previamente.
<b>Pasos</b>	<ol style="list-style-type: none"> <li>1. Navegar a pantalla de Resultados.</li> <li>2. Seleccionar resultados de selección de atributos.</li> <li>3. Rellenar la casilla de filtro por el nombre de un dataset cargado previamente.</li> </ol>
<b>Resultado esperado</b>	La lista de los datasets se filtra mediante el campo de búsqueda por el nombre del dataset introducido.
<b>Resultado Obtenido</b>	El resultado esperado.

## 6.2 PRUEBAS DE INTEGRACIÓN

En este apartado vamos a detallar los resultados de las pruebas de integración del sistema. En la que se realizaran pruebas en un entorno similar al de producción en el que al integrar todos los módulos del sistema se comprobara que funcionan correctamente.

Para realizar esta prueba se realizó el paso de construcción de los componentes del sistema se conectan adecuadamente la conexión entre ellos es correcta y el funcionamiento de los componentes es el esperado.

<b>PI – 01: Aplicación Web</b>	
<b>Conexión con</b>	Servidor BackEnd
<b>Descripción</b>	Se comprobará la correcta integración con el servidor BackEnd.
<b>Resultado esperado</b>	Los métodos de todos los módulos funcionan correctamente y la conexión entre estos componentes sea correcta.
<b>Resultado obtenido</b>	El resultado esperado.

<b>PI – 02: Aplicación Móvil</b>	
<b>Conexión con</b>	Servidor BackEnd
<b>Descripción</b>	Se comprobará la correcta integración con el servidor BackEnd.
<b>Resultado esperado</b>	Los métodos de todos los módulos funcionan correctamente excepto la alta de conjunto de datos y la conexión entre estos componentes sea correcta.
<b>Resultado obtenido</b>	El resultado esperado.

### 6.3 PRUEBAS DE SEGURIDAD

En este apartado veremos el resultado de las pruebas de seguridad donde se comprobará el acceso a distintos métodos de la API del servidor BackEnd, comprobando que la configuración del servidor es correcta y solo permite el acceso a esos métodos cuando debe.

<b>PS-01: Acceso a subir dataset sin JWT</b>	
<b>Descripción</b>	Se intentará acceder a el método para subir un archivo sin ningún token de autenticación.
<b>Resultado esperado</b>	El servidor devuelve un error 401 HTTP, sin permitir el acceso a el método. El sistema queda intacto.
<b>Resultado Obtenido</b>	El resultado esperado.

<b>PS-02: Acceso a procesar algoritmo de selección de atributos sin JWT</b>	
<b>Descripción</b>	Se intentará acceder a el método para procesar un algoritmo de selección de atributos de un dataset dado de alta en el sistema sin ningún token de autenticación.
<b>Resultado esperado</b>	El servidor devuelve un error 401 HTTP, sin permitir el acceso a el método. El sistema queda intacto.
<b>Resultado Obtenido</b>	El resultado esperado.

<b>PS-03: Acceso a registro sin JWT</b>	
<b>Descripción</b>	Se comprobará que el sistema permita acceder a los métodos de autenticación como el registro sin token de autenticación.
<b>Resultado esperado</b>	El servidor permite el acceso a el método. El usuario queda registrado en el sistema.
<b>Resultado Obtenido</b>	El resultado esperado.

<b>PS-04: Acceso a LogIn sin JWT</b>	
<b>Descripción</b>	Se comprobará que el sistema permita acceder a los métodos de autenticación como el LogIn sin token de autenticación.
<b>Resultado esperado</b>	El servidor permite el acceso a el método. El usuario inicia sesión en el sistema correctamente.
<b>Resultado Obtenido</b>	El resultado esperado.

<b>PS-05: Acceso a Resultados sin JWT</b>	
<b>Descripción</b>	Se intentará acceder a los resultados de ejecuciones sin token de autenticación
<b>Resultado esperado</b>	El servidor devuelve un error 401 HTTP, sin permitir el acceso a el método. El sistema queda intacto.
<b>Resultado Obtenido</b>	El resultado esperado.

## 7. CONCLUSIONES Y COMPETENCIAS ADQUIRIDAS

En este capítulo de la memoria se tratarán las conclusiones sobre el trabajo realizado, las competencias adquiridas, las futuras mejoras a el sistema desarrollado y la relación con las asignaturas del grado.

### 7.1 CONCLUSIONES DEL TRABAJO

Finalmente tras haber finalizado el trabajo de fin de grado y haber desarrollado la aplicación tal como estaba planeada, me siento orgulloso de todos estos meses de trabajo y constante evolución personal que me ha supuesto enfrentarme a el reto de trabajar con frameworks de desarrollo nuevos, que me suponían un gran interés debido a que son frameworks de desarrollo que se utilizan en el mercado y me parecía un reto interesante poder aprender a utilizar estas tecnologías nuevas a la vez que me enfrento a el trabajo de un sistema que esta enfocado en uno de los campos de informática que más me atraen que es la Inteligencia Artificial.

Desde un principio estuve explorando distintas tecnologías y estudiando sus ventajas e inconvenientes mientras hacía prototipados rápido para ver como funcionaban y como era el desarrollo de estas. Hay tecnologías que se han dejado de lado para trabajar con otras, pero me parece que esto ha supuesto un estudio de nuevas tecnologías de desarrollo web muy interesante.

Me satisface personalmente el haber logrado enfrentarme a problemas de un proyecto de este tipo y haberlos podido solucionar de manera autónoma. También he podido enfrentarme al ejercicio que es organizarme y resolver las trabas que iban apareciendo en el desarrollo del proyecto.

Respecto a lo aprendido en el grado creo que, aunque no se haya desarrollado directamente con tecnologías que se han estudiado en el grado, los conocimientos adquiridos en el transcurso de este han sido esenciales para el desarrollo del trabajo.

### 7.2 RELACIÓN CON EL GRADO

Durante el desarrollo del proyecto han sido esenciales los conocimientos adquiridos en el grado es por lo que realizo esta lista de asignaturas que han aportado en cierta manera a la realización del trabajo.

- Ingeniería del Software I II: Análisis de los requisitos y metodologías de desarrollo software.
- Asignaturas de Bases de datos: Por la comprensión y análisis del uso de una base de datos relacional.
- Ingeniería de proyectos: Documentación del proyecto.
- Planificación de proyectos: Metodologías de trabajo y seguimiento de proyectos.
- Programación Avanzada: Comportamiento de aplicaciones Web, uso de HTML, CSS y Javascript.
- Integración de tecnologías: Iniciación a los Web Services RESTFul.
- Calidad: Uso de GitHub, Docker, Maven y mención de Google Cloud Platform.
- Inteligencia Artificial: Algoritmos clasificadores e introducción a Selección de atributos.
- Inteligencia de negocio: Uso de librería de data mining Weka.
- Redes de computadores: Conexión entre distintos sistemas externos.
- Sistemas operativos: Concurrencia de procesos.



## 8. BIBLIOGRAFIA

- [1] Weka Plataforma de software enfocada a la data mining que cuenta con varios servicios <https://www.cs.waikato.ac.nz/ml/weka/>
- [2] Miguel García Torres, Francisco Gómez Vela, Belén Melián Batista, J. Marcos Moreno Vega: “*High-dimensional feature selection via feature grouping: A Variable Neighborhood Search approach*” <https://www.sciencedirect.com/science/article/pii/S0020025515005460>
- [3] Spring Framework <https://spring.io/>
- [4] Vue Javascript Framework <https://vuejs.org/>
- [5] NativeScript Native mobile apps Framework <https://nativescript.org/>
- [6] Yvan Saeys, Iñaki Inza, Pedro Larrañaga: “*A review of feature selection techniques in bioinformatics*” <https://academic.oup.com/bioinformatics/article/23/19/2507/185254>
- [7] Material design component Framework <https://vuetifyjs.com/en/>
- [8] State Management library <https://vuex.vuejs.org/>
- [9] Http client for browsers <https://github.com/axios/axios>
- [10] Fuente de datasets experimentados: <http://csse.szu.edu.cn/staff/zhuzx/Datasets.html>  
<https://storm.cis.fordham.edu/~qweiss/data-mining/weka-data/diabetes.arff>