Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Ángel Miguel García Vico y Don Cristóbal José Carmona del Jesus, tutores del Proyecto Fin de Carrera titulado: Aplicación Android para análisis de datos generados por bandas inteligentes, que presenta David Padilla Montero, autorizan su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Julio de 2022

El alumno:

Firmado por PADILLA MONTERO DAVID - 75573486V el día 03/07/2022

David Padilla Montero

Los tutores:

CARMONA DEL JESUS CRISTOBAL JOSE - 75102603J

Firmado digitalmente por CARMONA DEL JESUS CRISTOBAL JOSE - 75102603J
Fecha: 2022.07.03 11:08:26 +02'00'

Firmado digitalmente por GARCIA VICO ANGEL MIGUEL - ***5667** el día 03/07/2022 con un certificado emitido por AC FNMT Usuarios

Cristóbal José Carmona del Jesus

Ángel Miguel García Vico

Agradecimientos

A los profesores de la Universidad de Jaén por enseñarme durante mis años en la carrera.

A mis tutores Ángel y Cristóbal por su supervisión y recomendaciones.

A Estefanía Torrón Blasco y Alicia Alejandra Meza Meza por ayudarme con la revisión gramatical de la memoria.

A mis padres y amigos porque su apoyo me ha acompañado en los momentos más difíciles.

Índice

Índices de figuras

Índices de tablas

Índices de códigos

Capítulo 1: Introducción

Durante los últimos años el uso de aplicaciones que capturan de manera constante la actividad que realiza una persona en su día a día ha aumentado significativamente. Estas aplicaciones permiten el seguimiento del rendimiento físico o mejoras en los hábitos de sueño, entre otras.

Este tipo de aplicaciones, además, se benefician del uso de los dispositivos *wearables*[1] por el uso de sus sensores, que pese a tener un rendimiento no profesional, proporcionan a los usuarios datos del ritmo de sueño, pulsaciones o actividad.

El éxito de estos dispositivos se puede ver reflejado en el aumento de las ventas de los mismos. Por ejemplo, según Lei Jun, CEO de Xiaomi, las ventas de la Xiaomi Mi Band 6 habían excedido el millón de ventas en 2021 [1].

1.1 Motivación

El incremento de *wearables*[1] ha supuesto el aumento del uso de las aplicaciones de salud propias de cada compañía. Las aplicaciones de salud recogen los datos que se obtienen de estos dispositivos para después mostrarlos a los usuarios, con la finalidad de que puedan ver cómo ha sido su día.

La cantidad de aplicaciones de salud es extensa. Un ejemplo sería la aplicación "Google Fit", que no solo recoge información de las *smartbands*[2] sino también de otros teléfonos móviles y aplicaciones externas, con el objetivo de tener un centro disponible para el usuario. Las aplicaciones registran los datos las veinticuatro horas al día, siempre y cuando los dispositivos que los recogen estén generando valores que puedan almacenar.

Esto genera una gran cantidad de datos que actualmente se muestran al usuario sin ninguna finalidad de estudio posterior. Sin embargo, estos pueden ser un buen campo de trabajo dentro de la inteligencia artificial (IA) dado el gran volumen de

[1] Dispositivos que se pueden vestir
[2] Bandas inteligentes

información generado por la cantidad de dispositivos que se encuentran en el mercado, lo que permite extraer comportamientos o patrones de ellos.

Es posible utilizar este tipo de datos de salud (ritmo cardíaco, actividad o futuros sensores como los de oxígeno en sangre) dentro de la IA como han demostrado algunos proyectos como “AIHealth”, orientada al monitoreo de pacientes y control sanitario de los mismos [2].

1.2 Objetivos

El principal objetivo de este proyecto es la generación de un prototipo de aplicación móvil, que se encargará de la extracción de datos. Además, se incorporan algoritmos de IA que permitan obtener y generar reglas de conocimiento y comportamiento aplicando modelos de Ciencia de Datos.

La Ciencia de Datos (o *Data Science*) es un campo interdisciplinario que se puede definir como “la disciplina que convierte los datos en conocimiento útil” [3]. Al ser interdisciplinario, utiliza la minería de datos para la extracción de estos, la estadística, para la toma de decisiones y la inteligencia artificial o *Machine learning* para el aprendizaje y la creación de conocimiento.

Los datos se pueden obtener de las bandas inteligentes que, gracias a su extensión en el mercado, son fáciles de adquirir y poseen diversos sensores.

Para alcanzar este objetivo principal se deben cumplir los siguientes subobjetivos:

- Realizar un estudio de mercado para conocer las *smartbands* de posible uso.
- Generar un análisis sobre los posibles tipos de datos a tomar, tanto en cantidad como variedad.
- Creación de una aplicación prototipo.
- Extracción del conocimiento sobre la información que se ha generado en las *smartbands.*

1.3 Estimación de costes

Después de establecer cuáles son los objetivos del proyecto, se realizará un análisis sobre la viabilidad del mismo. Se utilizará el coste tanto en tiempo, como en material necesario para conocer el coste total de desarrollo y generación del prototipo funcional.

Usando la Figura 1 como medida temporal del proyecto y tomando como referencia que se realizarán 32.5 horas semanales al coste de 35 euros la hora, podemos obtener el coste total de personal asociado al proyecto, además de poder obtener el coste de cada tarea de manera independiente tal y como se muestra en la Tabla 2.

Para el desarrollo de este proyecto se utilizan dispositivos *smartbands*, ya que tienen los sensores necesarios para el prototipo. Por ello se van a incluir estos dispositivos, que han de ser comprados, en el coste de generación del prototipo, viendo dicho precio en la Tabla 1.

Material	Coste Asociado
Huawei Band 4 pro	43,61 €
Xiaomi Mi Band 5	32,90 €
Samsung Galaxy Fit 2	52,00 €
TOTAL	**128,51 €**

Tabla 1.- Coste dispositivos smartbands.

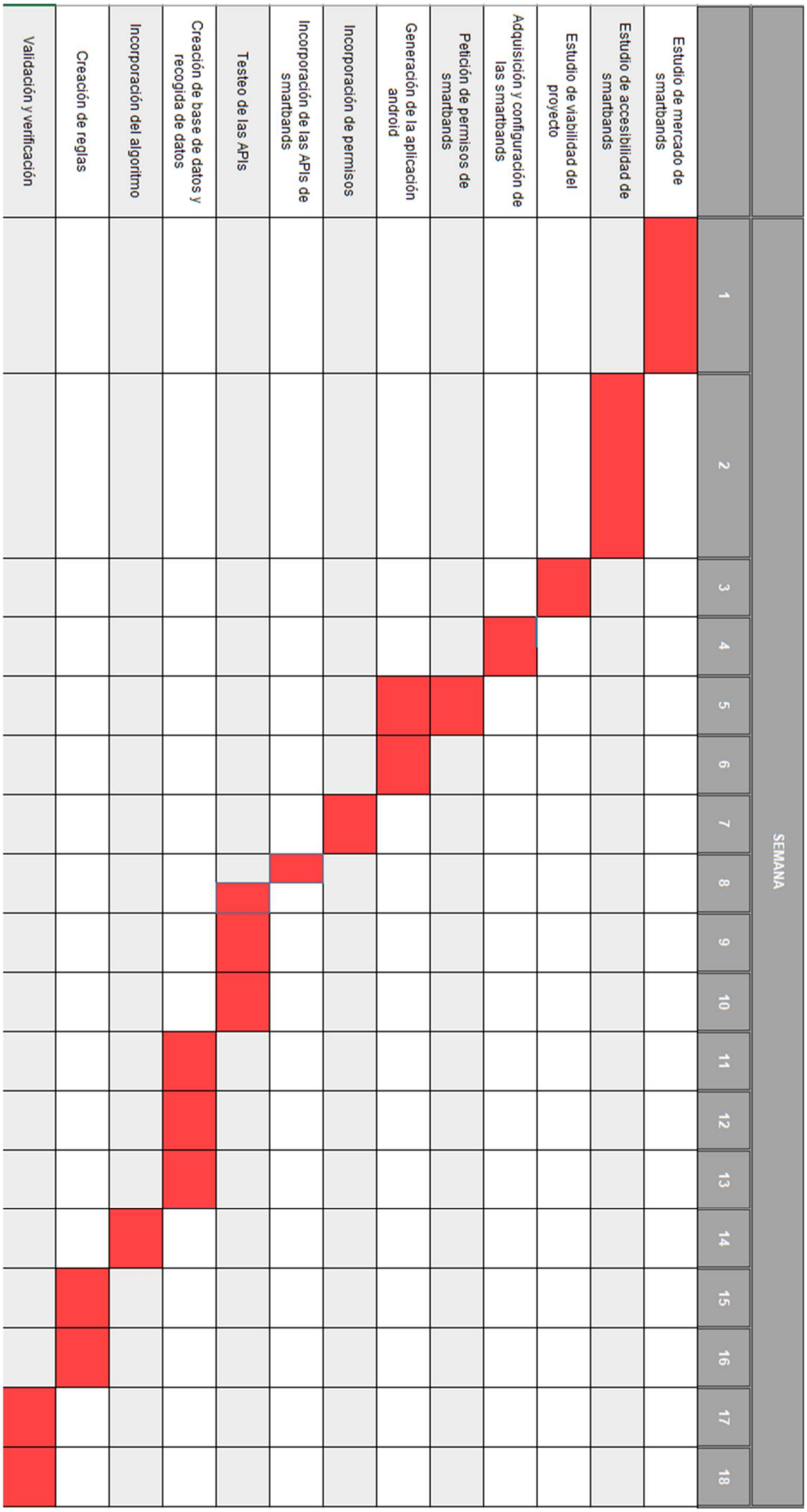

Figura 1.- Diagrama de Grantt. Fuente propia.

Tarea	Duración	Coste Asociado
Estudio de mercado de *smartbands*	1 semana	1.137,50 €
Estudio de accesibilidad de *smartbands*	1 semanas	1.137,50 €
Estudio de viabilidad del proyecto	1 semana	1.137,50 €
Adquisición y configuración de las *smartbands*	1 semana	1.137,50 €
Petición de credenciales de *smartbands*	1 semanas	1.137,50 €
Generación de la aplicación Android	2 semanas	2.275,00 €
Incorporación de permisos	1 semana	1.137,50 €
Incorporación de las APIs de *smartbands*	media semana	568,75 €
Testeo de las APIs	2 semanas y media	2.843,75 €
Creación de base de datos y recogida de datos	3 semanas	3.412,50 €
Incorporación del algoritmo	1 semana	1.137,50 €
Creación de reglas	2 semanas	2.275,00 €
Validación y verificación	2 semanas	2.275,00 €
TOTAL	**18 semanas**	**20.475,00 €**

Tabla 2.- Costes asociados a la distribución temporal.

Capítulo 2: Situación actual

Las *smartbands* son dispositivos autosuficientes, dotados de batería y que presentan una pantalla con información, pero eso no es suficiente para poder hacer un estudio dado que los datos simplemente están para mostrarse sin ser procesados. Por ello se va a investigar sobre las tecnologías disponibles para realizar este proyecto y que pueden ser de utilidad para su desarrollo o su planteamiento, empezando por la plataforma donde correrá el prototipo del proyecto, un análisis de los lenguajes que pueden ser utilizados y de las tecnologías que se disponen actualmente.

También tomará parte del estudio las *smartbands* seleccionadas, características y el proceso de adquisición de credenciales de estas, además de las APIs de cada uno de ellas.

2.1 Plataformas

En este primer apartado se lleva a cabo el análisis de plataformas de dispositivos móviles, ya que es el destino de la aplicación a desarrollar durante este trabajo.

2.1.1 Android

Android nació pensado originalmente para teléfonos móviles. Está basado en Linux, por lo que posee un núcleo o *kernel* multiplataforma y es gratuito. Asimismo, permite que se desarrollen aplicaciones basadas en un lenguaje de programación extendido y conocido como Java o Kotlin. Android además se encarga de proporcionar todas aquellas interfaces que sean necesarias para el acceso a funciones del propio dispositivo móvil.

Gracias a ello, se ha generalizado el uso de Android y han aparecido diferentes herramientas de programación gratuitas que han hecho posible la existencia de una gran cantidad de aplicaciones, otorgando una experiencia de usuario muy amplia.

Algunas de las características principales de Android son:

- Gratuito: AOSP (Android Open Source Project) es el núcleo de Android, siendo de código abierto, aunque generalmente se incluyen servicios de Google. Sin embargo, está disponible de manera gratuita siempre que se acepten las cláusulas de Google [4].

- Personalizable: Al ser un sistema de código abierto, la comunidad de programadores ha hecho que se tenga una gran libertad a la hora de personalizarlo.
- Multiplataforma: Se puede encontrar el sistema de Android en una enorme cantidad de dispositivos, como Smartphones, tabletas, PDA, relojes, reproductores, etc.

Esto no exime a Android de disponer de más de un defecto o problema, ya que el hecho de poder estar en casi cualquier dispositivo se puede convertir en una desventaja. Esto se debe a que cada dispositivo tiene una configuración de componentes y hardware distinta, por lo que adaptarse a todos ellos puede suponer que se pierda eficiencia, es decir, que Android no siempre sea capaz de obtener el máximo porcentaje de funcionalidad en todos los componentes de estos dispositivos.

Un problema que puede originarse debido a lo comentado anteriormente es que, teniendo dos dispositivos con una misma versión de Android y la aplicación desarrollada, uno de los dos dispositivos puede no ser capaz de ejecutar Android o la aplicación. Sin embargo, este tipo de errores se están intentando paliar con cada nueva versión que Android saca al mercado.

2.1.2 iOS

El siguiente sistema operativo que analizar es el sistema operativo móvil de Apple, iOS, competidor actual de Android que fue creado para los productos móviles de Apple como iPhone, iPad, iPod, etc.

iOS presenta como característica principal su calidad y su originalidad, las cuales ha mantenido en su interfaz elegante e intuitiva. Un punto fuerte que mantiene toda su comunidad de Internet es que iOS fue pionero en el multigesto y en el desbloqueo del dispositivo con el rostro, que fueron revolucionarios en sus orígenes.

Apple se ha encargado de proporcionar un sistema operativo que sea capaz de obtener todo el rendimiento del hardware, garantizando que sus productos sean de alta calidad. Este hecho implica que los usuarios sientan que han comprado algo que funcionará correctamente con alta tolerancia a fallos. Pero esta característica se debe a que un dispositivo iOS no es totalmente configurable y de que no es posible incluir iOS en un dispositivo externo a Apple.

Los dispositivos Apple se encuentran en un rango de precios mayor, por lo que adquirir alguno de estos dispositivos puede ser considerado caro pero la calidad y el soporte que ofrece hace que cada día iOS sea más popular.

2.1.3 Comparativa

Para poder compararlos vamos a tener en cuenta el porcentaje de dispositivos que hay actualmente en el mercado durante este último año (desde marzo de 2021 a marzo de 2022).

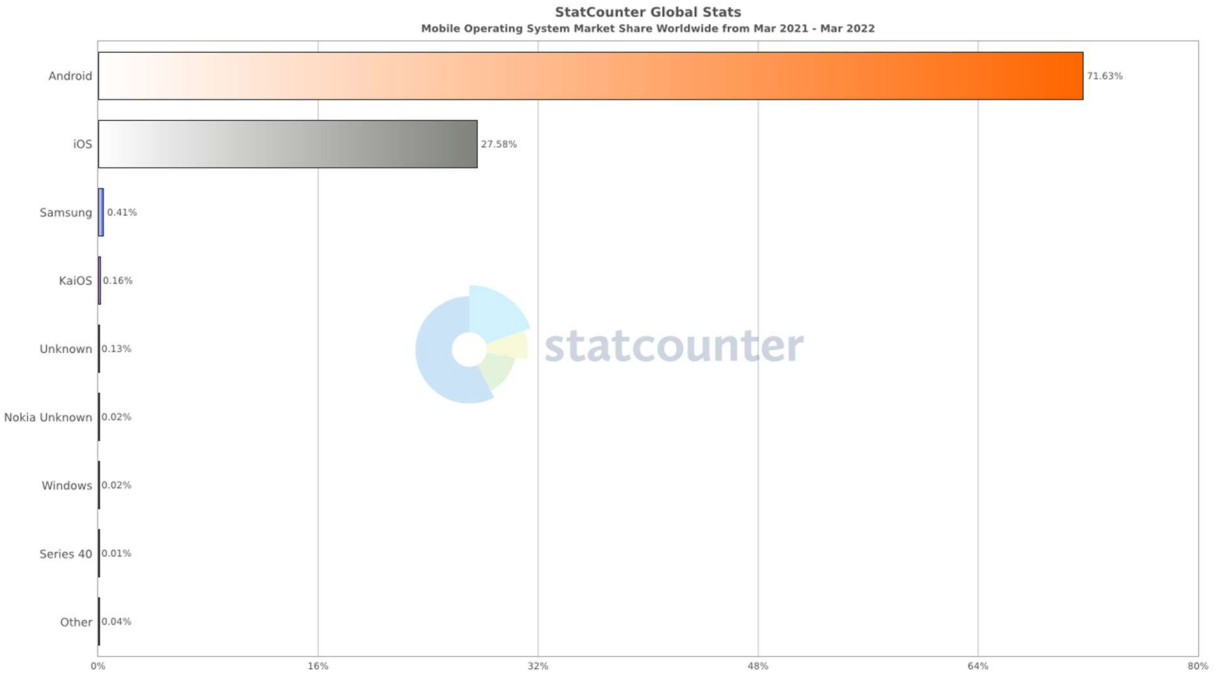

Figura 2.- Comparativa OS móviles.
Fuente: Statcounter

Observando el gráfico, Figura 2, se puede ver que más del 70% de los dispositivos móviles en el último año utilizan Android. Por ello, para la creación de una aplicación lo más cómoda y que permita acceder a un amplio abanico de dispositivos, seleccionaremos Android, ya que ofrece un sistema operativo gratuito, amplio y que supone el 71% de los dispositivos móviles actuales, lo cual lo hace el sistema operativo más idóneo a utilizar.

2.2 Lenguajes de programación

En el siguiente apartado se van a analizar los lenguajes de desarrollo que pueden usarse en el sistema operativo seleccionado.

2.2.1 HTML5

HTML5 es un lenguaje *markup*, que se utiliza para la creación de contenido en la web. HTML5 establece los ajustes de aspecto que dispondrá la web y proporciona esta información a los navegadores. Con ello tendrán conocimiento de cómo deben colocar el texto en una página web, así como el resto de elementos que tenga como imágenes, videos, etc.

Poco a poco, HTML5 ha mejorado para evitar la dependencia de *plugins* externos y ser capaz de incluir más contenido multimedia.

2.2.2 Java

Tal y como lo describe la empresa desarrolladora, “Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa. Con más de 9 millones de desarrolladores en todo el mundo, Java le permite desarrollar, implementar y utilizar de forma eficaz interesantes aplicaciones y servicios” [5].

Java es un lenguaje de programación orientado a objetos. Este fue diseñado con la idea de tener pocas dependencias y que permitiese a los desarrolladores escribir únicamente una vez el programa y se pueda ejecutar en cualquier dispositivo. Esto se conoce como el término WORA (write once, run anywhere), lo cual se produce cuando el código se convierte en instrucciones máquina simplificadas y una máquina virtual (JVM) interpreta y ejecuta el código de manera nativa.

2.3 Algoritmos evolutivos

El término algoritmo evolutivo se utiliza para describir un sistema de resolución de problemas de optimización o búsqueda, basados en principios de la evolución biológica, siendo este el elemento clave en su diseño e implementación. Pertenecen a las metaheurísticas basadas en poblaciones y simulan este proceso biológico en un ordenador, usando optimización probabilística. Estos métodos, con bastante frecuencia, mejoran a los algoritmos clásicos en problemas complejos, esta mejora se debe a la mayor capacidad de exploración del espacio de soluciones [6] [7],

Esta clase de algoritmos obtienen su base del libro “El origen de las especies” de Darwin, donde planteaba que las especies nacen, evolucionan y las especies menos aptas desaparecen para que las más aptas se adapten al medio y perpetúen sus aptitudes. De acuerdo con esa visión la computación tomó las mejores soluciones temporales, que se cruzan y mezclan generando nuevos individuos o soluciones. Estos individuos tendrán parte del código genético de sus antecesores, generando una nueva generación de individuos como en las especies [8].

Los algoritmos evolutivos son una de las técnicas bioinspiradas más empleadas en la actualidad, donde un individuo es representado con soluciones análogas a cromosomas. Posteriormente se realizan transformaciones que se aprecian en la evolución de las especies tales como cruce, selección, inversión o mutación [9].

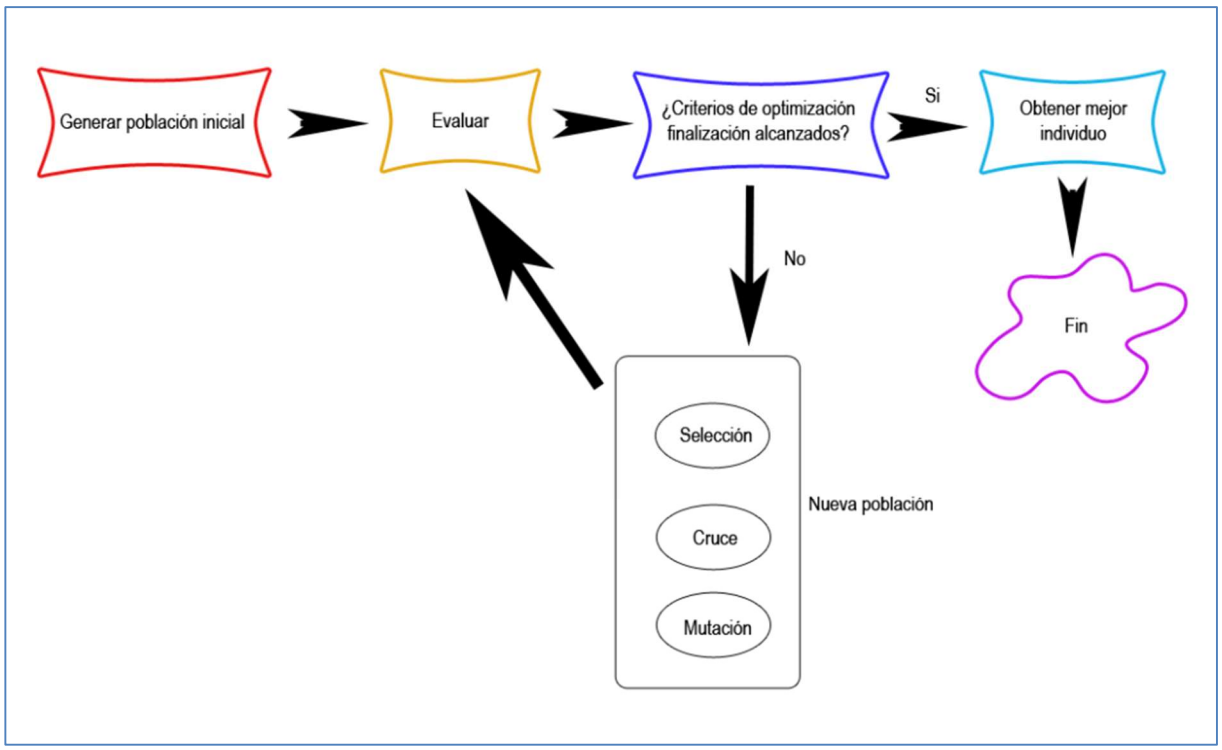

Figura 3.- Funcionamiento de algoritmos evolutivos. Fuente propia.

Viendo el esquema de la Figura 3 se puede observar que para iniciar el algoritmo se necesita generar una población inicial, que puede ser completamente aleatoria o seguir unas pautas indicadas. Cada nueva generación se realiza desde una población anterior.

El operador evolutivo de selección es un operador que intenta recrear el concepto de que los mejores y más adaptados individuos sean los que sobrevivan en

las siguientes generaciones. Existen diversos métodos de selección como selección escalada, por torneo o por rango entre otros.

El operador genético de recombinación o cruce, es el que implica tomar dos individuos obtenidos en el proceso de selección e intercambiar fragmentos de su código genético. Los nuevos individuos son descendientes de estos anteriores y obtienen el código genético mezclado.

El operador genético de mutación es más sencillo, donde se toma cada individuo y por azar se elige si sufrirá una mutación genética, que consiste en seleccionar un gen y modificarlo, ya sea por intercambio con otro gen del mismo, o por un gen aleatorio, haciendo así una modificación nueva de este individuo.

Este proceso se realizará generación tras generación hasta satisfacer el criterio de optimización, es decir, un valor deseable o satisfacer un criterio de parada, donde se establece un máximo de generaciones que se van a poder dar ya que la solución no es conocida, pero se desea mejorar cada generación hasta obtener el mejor individuo.

Para poder establecer el mejor individuo y poder guiar eficazmente la búsqueda se debe generar un sistema de evaluación del individuo. Esta evaluación depende de la representación del cromosoma, del problema, del algoritmo usado y de la solución del problema.

2.4 Aplicaciones de salud

En este apartado se van a explorar las aplicaciones de salud existentes, en este caso de los dispositivos *smartbands* que se han escogido.

2.4.1 Mi band 5. Xiaomi

A continuación, en la Tabla 3, se presentan las características de la *smartband* Xiaomi Mi Band 5 y la aplicación que utiliza Mi Fit.

	Xiaomi Mi Band 5
Pantalla	OLED a color 1,1 pulgadas. 126 x 294 píxeles
Peso	22 gramos (estimado)
Batería	125 mAh
Resistencia	Hasta 50 metros bajo el agua
Duración	14 días (según fabricante)
Conectividad	Bluetooth 5,0
Compatibilidad	iOS y Android
Sensores	Acelerómetro
	Giroscopio
	Sensor de ritmo cardíaco
	Reconocimiento de 11 deportes
	PPG

Tabla 3.- Especificaciones Mi Band 5.

El reloj parece completo a excepción de la falta GPS, pero por el hecho de que esté conectado a un dispositivo móvil no debería de haber ningún problema. Dado su precio, está bien equipado con los siguientes sensores:

- Ritmo cardíaco.
- Pasos.
- Actividad.
- Sueño.

Figura 4.- Aplicación Mi Fit. Fuente propia.

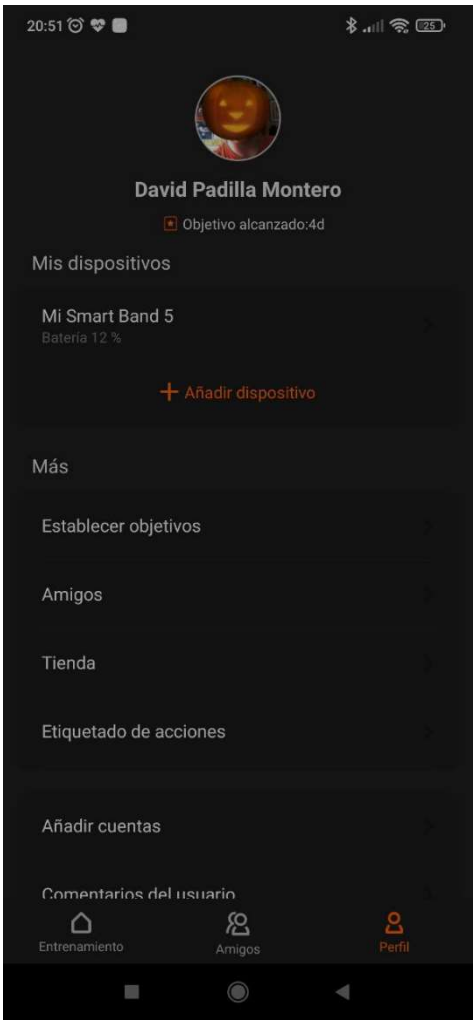

Figura 5.- Configuración Mi Fit. Fuente propia.

Existen muchas posibles vistas y partes que ver en la aplicación de Mi Fit, la más importante es la vista principal de la Figura 4. En ella se hace un muestreo sencillo de los datos, permitiendo además ver el resto de estos de manera más precisa y detallada uno a uno, es decir, eligiendo qué se quiere ver, por ejemplo, sueño, actividad o ritmo cardíaco.

En la Figura 5 aparece la configuración, que es simple y sencilla. Se puede tener más de un dispositivo *smartwatch* conectado simultáneamente, donde se puede ver la *smartband*, Mi Band 5, conectada.

2.4.2 Huawei Band 4 pro

Similar al dispositivo anterior, a continuación, se van a exponer las características del Huawei Band 4 pro que se usará durante el proyecto.

	Huawei Band 4 Pro
Pantalla	AMOLED a color 0,95 pulgadas. 120 x 240 píxeles
Peso	25 gramos (estimado)
Batería	100 mAh
Resistencia	Hasta 50 metros bajo el agua
Duración	12 días (según fabricante)
Conectividad	Bluetooth 4,2
Compatibilidad	iOS y Android
Sensores	Acelerómetro
	Giroscopio
	Sensor de ritmo cardíaco
	Reconocimiento de 11 deportes
	PPG
	GPS

Tabla 4.- Especificaciones Band 4 Pro.

Las características de Huawei Band 4 Pro, Tabla 4, son similares al Mi Band 5, con la diferencia de que esta *smartband* sí dispone de GPS, además de un poco más de peso, pero no supone una gran diferencia. Los sensores de Huawei Band 4 Pro son más precisos según Huawei ya que son sensores IMU (*Inertial Measurement Unit*).

Figura 6.- Huawei Health principal. Fuente propia.

Figura 7.- Huawei Health configuración. Fuente propia.

La vista principal de la aplicación de salud ”Huawei Health”, Figura 6, es compacta y directa. Muestra los datos del día (siempre que hayan sido sincronizados) y posteriormente se puede acceder a cada uno de ellos para una vista más detallada. La parte de configuración de dispositivos, Figura 7, está separada de la parte de configuración en una pestaña independiente, para que esta pestaña se encargue de la conectividad y de la personalización del smartband.

2.4.3 Samsung Galaxy Fit 2

El último dispositivo usado pertenece a Samsung con el modelo, Galaxy Fit 2 color negro, Tabla 5.

	Samsung Galaxy Fit 2
Pantalla	AMOLED a color 1,1 pulgadas. 126 x 294 píxeles
Peso	21 gramos sin correa
Batería	159 mAh
Resistencia	Hasta 50 metros bajo el agua
Duración	15 días (según fabricante)
Conectividad	Bluetooth 5,1
Compatibilidad	iOS y Android
Sensores	Acelerómetro
	Giroscopio
	Sensor de ritmo cardíaco
	Reconocimiento de actividad

Tabla 5.- Especificaciones Galaxy Fit 2.

Comparativamente con respecto a los anteriores este es el dispositivo *smartband* que tiene mayor capacidad de batería y por ello mayor duración de uso, pero por el contrario, es aquel que posee menor cantidad de sensores.

Respecto a la aplicación se divide en dos partes, una primera Galaxy Wearable que es la encargada de la conectividad con el dispositivo, mientras que por otra parte está Samsung Health que se encarga de la toma de datos y del muestreo de estos al usuario.

Figura 8.- Samsung Health. Fuente smartwatchzone.

La aplicación de Samsung Health, Figura 8, muestra una pantalla inicial con un resumen de los datos obtenidos y después la posibilidad de poder acceder a cada uno de ellos de manera independiente en caso de que se quiera más información.

2.4.4 Comparativa

Se muestra ahora la Tabla 6 con los elementos más interesantes de cada dispositivo smartband, para una fácil comparación entre ellos.

	Xiaomi Mi Band 5	Samsung Galaxy Fit 2	Huawei Band 4 Pro
Pantalla	OLED 1,1 pulgadas. 126 x 294 píxeles	AMOLED 1,1 pulgadas. 126 x 294 píxeles	AMOLED 0,95 pulgadas. 120 x 240 píxeles
Batería	125 mAh	159 mAh	100 mAh
Duración	14 días	15 días	12 días
Sensores	Acelerómetro	Acelerómetro	Acelerómetro
	Giroscopio	Giroscopio	Giroscopio
	Sensor de ritmo cardíaco	Sensor de ritmo cardíaco	Sensor de ritmo cardíaco
	Reconocimiento de 11 deportes	Reconocimiento de actividad	Reconocimiento de actividad
	PPG		PPG Y GPS

Tabla 6.- Comparativa smartbands.

2.5 Credenciales

Para el uso de las peticiones a los sensores, es necesario obtener credenciales de los desarrolladores oficiales de las aplicaciones, dado que sin esas credenciales no es posible obtener datos desde las *smartbands*.

Se ha creado primero un contenedor de claves, Figura 9, ya que es obligatorio para poder desarrollar el proyecto en un IDE.

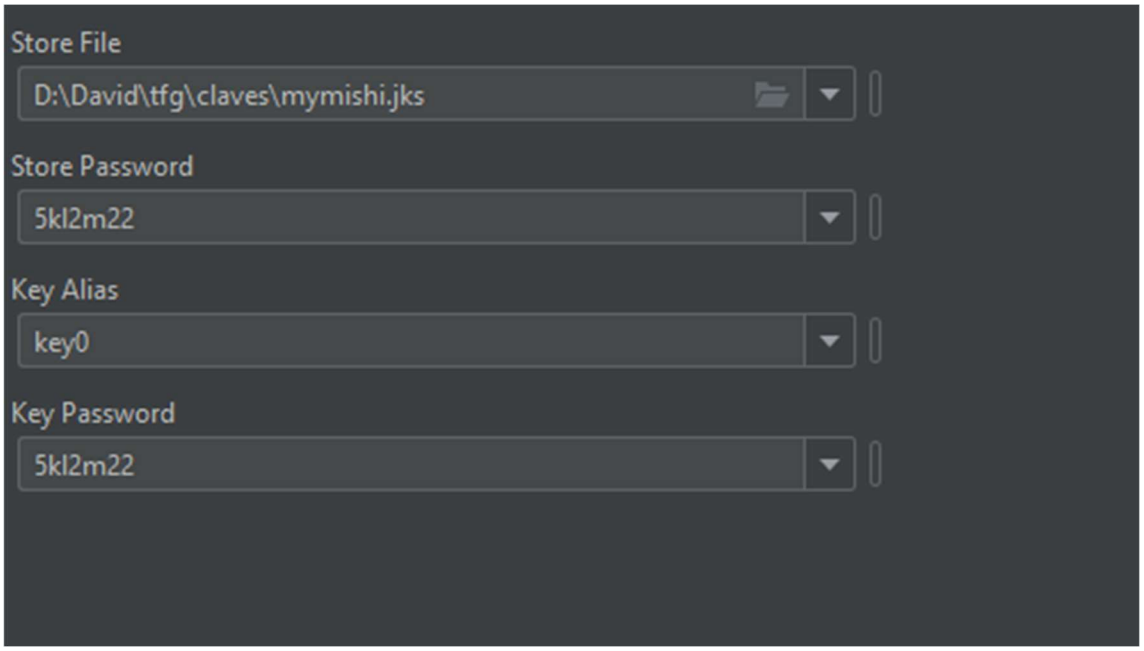

Figura 9.- SHA Store Key. Fuente propia.

Posteriormente, todas las credenciales necesitan de una clave SHA-1 y SHA-256, por lo que se va a crear una y que se usará de manera global en el proyecto.

Una vez obtenidos el contenedor y la clave, esta última se almacena con una firma certificada. En el caso del proyecto con el nombre y apellidos del desarrollador junior encargado de la adquisición de los certificados, tal y como se ve en la Figura 10. Esta clase de certificados son usados para identificarse posteriormente en las páginas de certificación de las diferentes empresas.

Finalmente se ha tenido que dar nombre a la aplicación, dado que sin él no se permite obtener acceso. Con ello también se debe establecer los paquetes que se van a usar y el nombre del proyecto, en este caso su nombre será lerning (IERNING), se usará el formato de paquetes com.tfg.ierning.

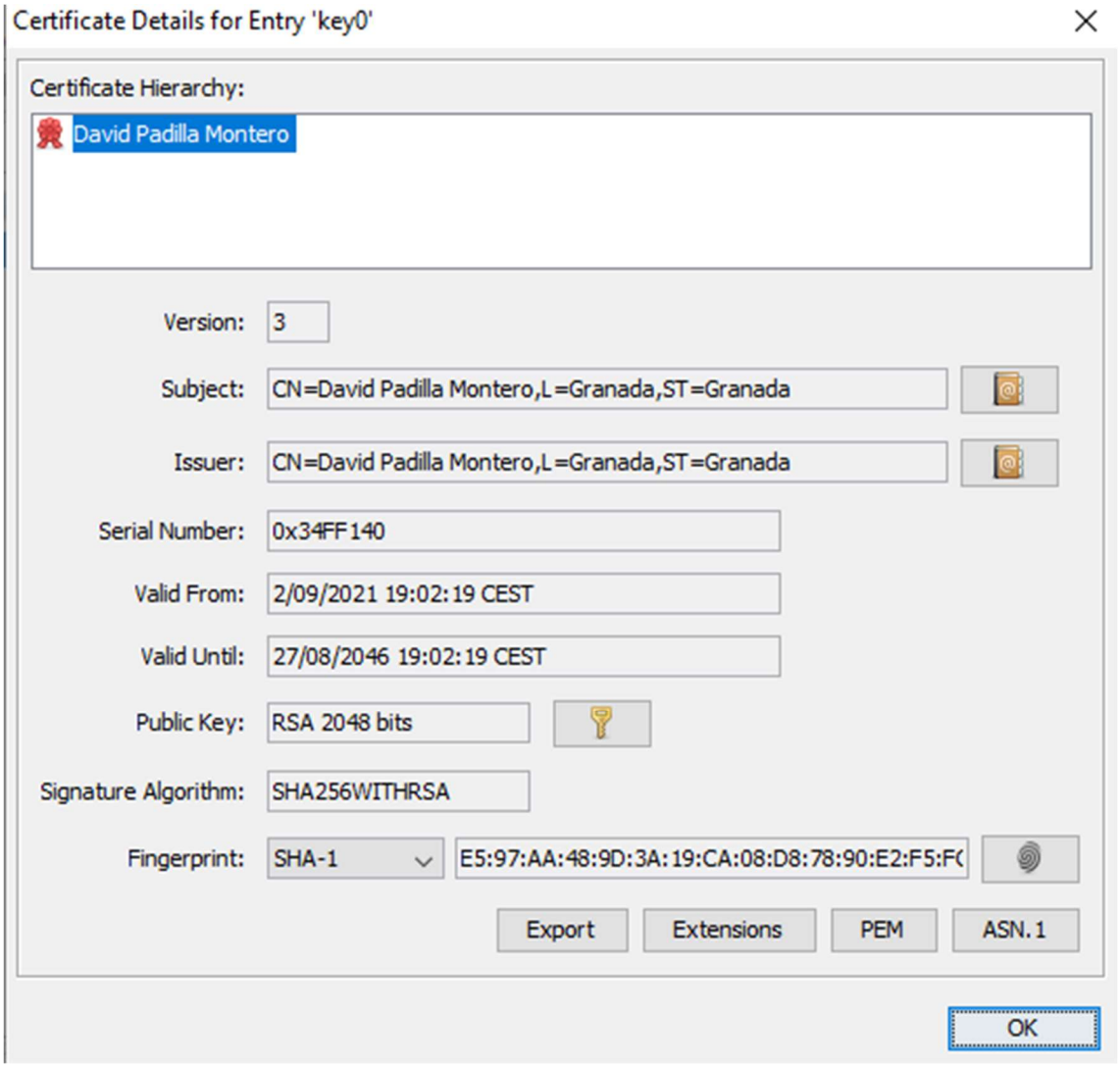

Figura 10.- Key 0 SHA. Fuente propia.

2.5.1. Huawei

En el primer caso, Huawei, se necesita de una de sus plataformas de servicios, en concreto la que Huawei proporciona. Esta se llama HMS Core, la cual Huawei define como “una colección de utilidades libres proporcionadas por HUAWEI Mobile Services (HMS). Este paquete se encuentra entre la aplicación y el sistema operativo del dispositivo, funcionando como una plataforma de servicios. HMS también ofrece servicios en la nube como HUAWEI CLOUD” [10].

HMS es el centro donde este proyecto va a trabajar, ya que incluye todas las aplicaciones y operaciones referentes a los dispositivos móviles y *wearables*. Por ello se pedirá acceso como desarrollador y así obtener credenciales de uso, además de obtener guías de cómo se deben realizar el uso de sus APIs, ya que estas guías solo están disponibles para desarrolladores verificados.

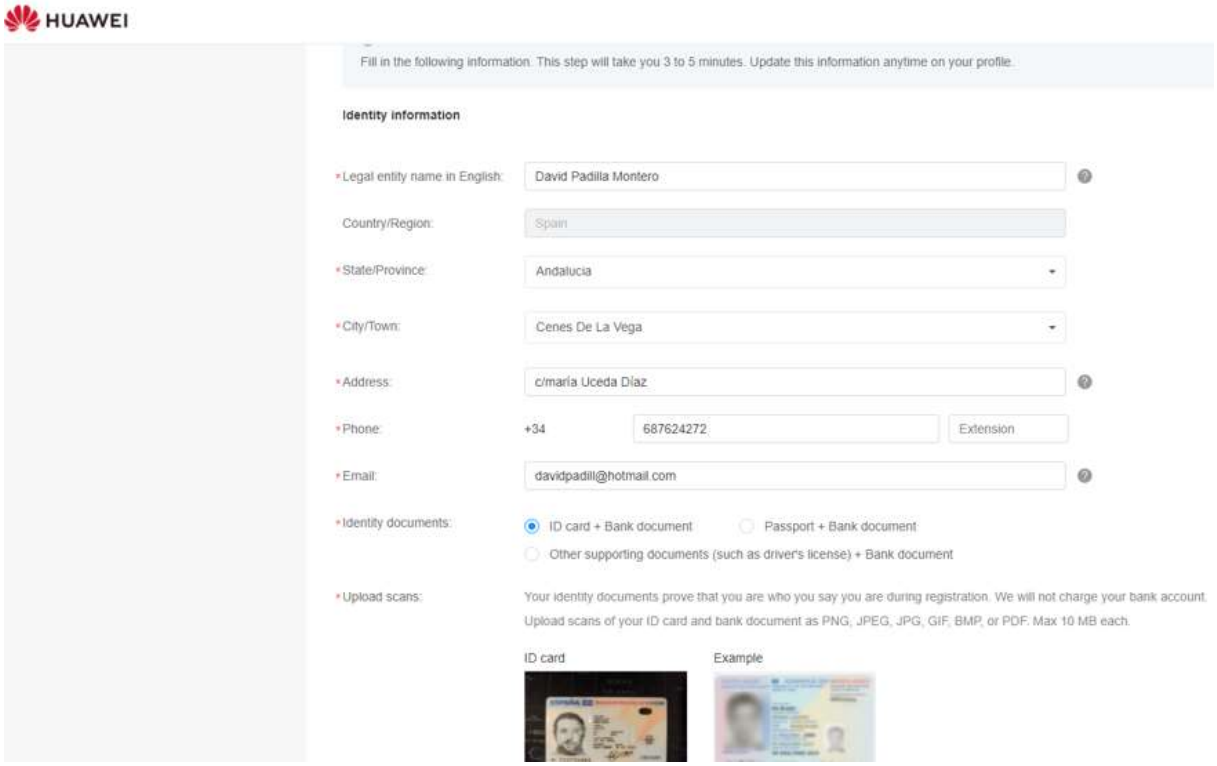

Figura 11.- Huawei developer Access. Fuente propia.

La cantidad de datos que Huawei solicita para dar acceso es bastante abrumadora, ya que comienza por necesidad de DNI, tarjeta de crédito, teléfono y localización, como muestra la Figura 11.

Una vez cumplidos estos requerimientos es necesario obtener verificación como desarrolladores, Figura 12. Para el caso de lerning se han logrado los permisos como desarrollador y obtener el acceso a poder crear una aplicación, pero ese proceso requiere de rellenar más información referente a qué es lo que se va a desarrollar y a qué datos de HMS se requiere acceso.

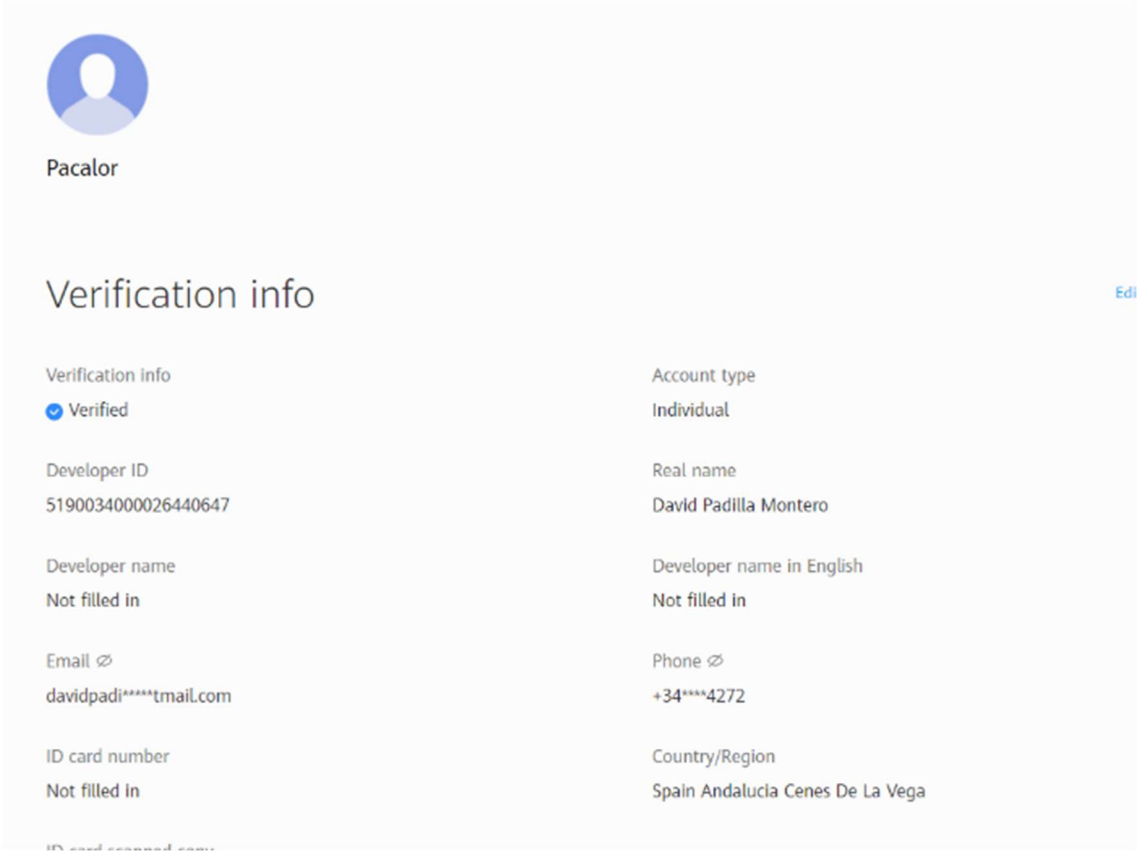

Figura 12.- Huawei developer data. Fuente propia.

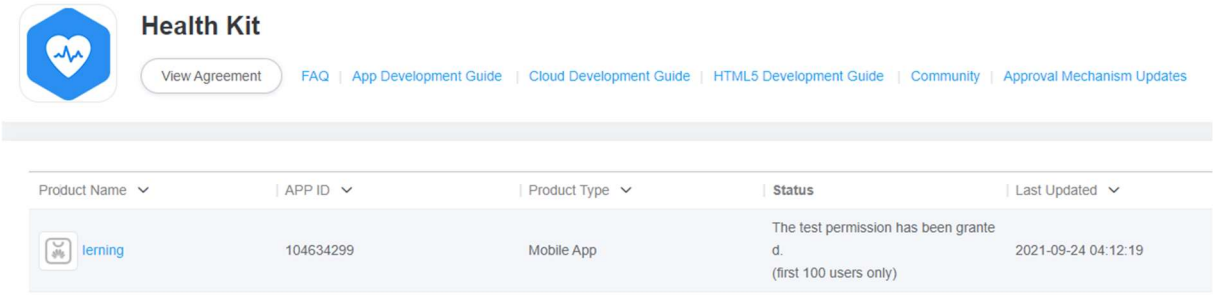

Figura 13.- Permisos de lerning. Fuente propia.

Para el resto de la documentación es necesario especificar donde aparecerán los datos, cómo se van a usar y si la aplicación usará información sensible del usuario. En lerning, serán sensibles, pero no entran en la categoría de riesgo. En la categoría de riesgo se incluyen tales como: análisis sanguíneo, seguimiento del ciclo menstrual o azúcar en sangre.

Después de incluir todo el papeleo necesario se van a obtener identificadores de cliente OAuth 2.0, además de tener los permisos para el desarrollo del prototipo como aparece en la Figura 13.

2.5.2 Google

Debido a que Xiaomi no dispone de una API para sus dispositivos, únicamente posee una aplicación de salud, se ha tomado la decisión de redirigir estas peticiones a un tercero que el propio programa de salud recomienda, Google Fit. Este dispone de la opción de establecer conexión con la aplicación de salud de Xiaomi (Mi Fit) y mantener los datos actualizados. Debido a que Mi Fit permite exportarlos como Excel y Google Fit lo ordena para su posterior acceso, por lo que con ello es posible tener una API cómoda.

La obtención de permisos pasa entonces por Google, que, a diferencia de Huawei, su obtención es algo más obtusa ya que algunas guías que se usan están algo desactualizadas. Pero gracias a un gestor de cuentas de Google es posible obtener una configuración fácil y cómoda, como por ejemplo Firebase que se muestra en la Figura 14.

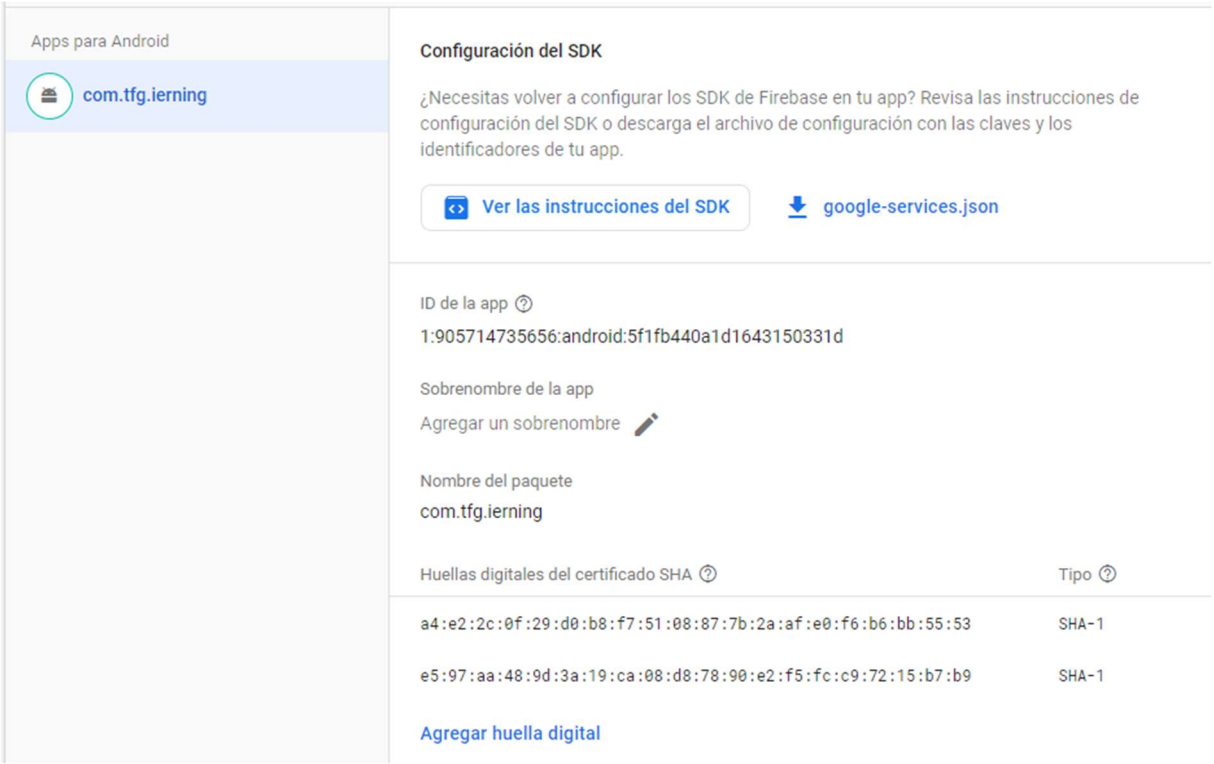

Figura 14.- Firebase Ierning configuración. Fuente propia.

Firebase, que viene integrado en el IDE Android Studio, permite seguir unos sencillos pasos y obtener los permisos, simplemente añadiendo las claves SHA que

ya se habían generado y siguiendo los pasos de los tutoriales que ellos mismos proporcionan.

Gracias a ello es posible ver en la plataforma de Google los permisos para la aplicación, Figura 15. Con ello solo debemos incluirlos dentro de la aplicación Android y enviar una petición con la clave e ID de cliente, así las peticiones de datos se completarán y es posible obtener las respuestas.

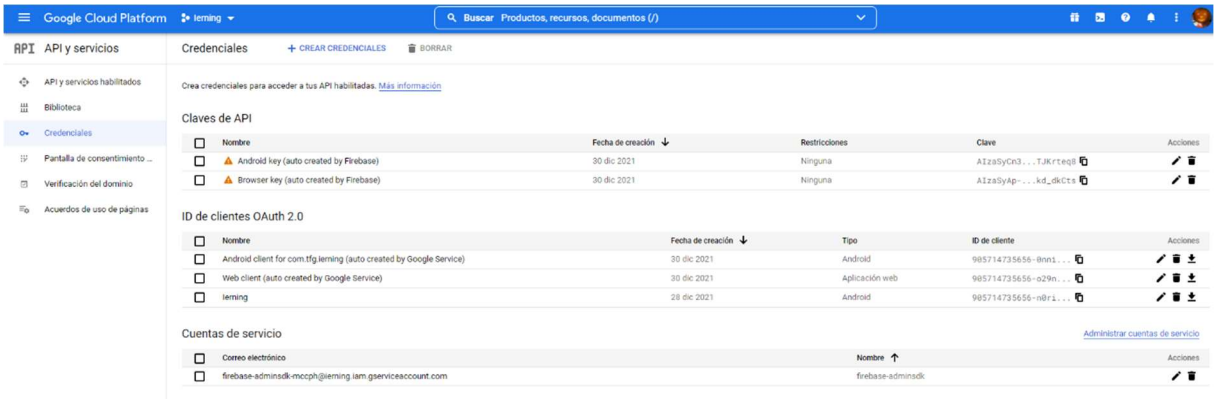

Figura 15.- Google credenciales. Fuente propia.

2.5.3 Samsung

Por desgracia con Samsung surgió un problema importante. Durante la creación de la aplicación no surgió ningún problema, pero a la hora de realizar peticiones apareció un error. Aunque Samsung no requería de credenciales para usar su API, la aplicación de salud requería unos permisos especiales en su configuración, estos permisos eran unas credenciales únicas para la aplicación. Lamentablemente en estos momentos se encuentra cerrado el plazo de peticiones para estas credenciales, ya fuese rellenando sus formularios o completando diferentes pautas, no era posible acceder a dicho proceso, por lo que Samsung se ha tenido que quedar fuera del proyecto debido a la imposibilidad de obtener datos del mismo.

Capítulo 3: Análisis y extracción de requisitos

En busca de poder hacer un buen diseño de la aplicación, es necesario hacer previamente un buen análisis del mismo, obteniendo conocimiento de qué es lo que se pretende hacer y con ello favorecer una correcta comprensión del funcionamiento de la aplicación a desarrollar.

Se va a dividir este apartado en los diferentes aspectos que se presentan en el análisis y especificación de requisitos, además de un análisis del algoritmo que se utilizará para extraer conocimiento y reglas.

3.1 Escenarios y Actores

El escenario que se utilizará está centrado en el uso de la aplicación para el estudio y la obtención de reglas, usando lerning como una herramienta que permite generar fácilmente estas reglas.

Francisco es un investigador hábil de las nuevas tecnologías y ha decidido realizar una aplicación de estudio para enfermedades del corazón. Necesita obtener datos constantes de una persona e información de cómo está evolucionando, por ello decide usar lerning para la extracción de reglas con las que generar su estudio. Pide estas funcionalidades:

- Tomar datos de su *smartband.*
- Poder establecer parámetros en el algoritmo.
- Extraer reglas de conocimiento.
- Una interfaz sencilla que no requiera de trabajo extra por su parte.

Con ello aparecen los siguientes actores implicados, Tabla 7 y 8.

A01 - Sistema

Tipo	Actor principal.
Descripción	Sistema que gestiona el aprendizaje, los datos y la aplicación móvil.
Interacción	Interacción directa desde la aplicación móvil.

Tabla 7.- Actor sistema.

A02 - Usuario	
Tipo	Actor principal.
Descripción	Persona de quien se van a tomar datos y quien generará eventos en la aplicación.
Interacción	Interacción directa desde la aplicación móvil.

Tabla 8.- Actor Usuario.

3.2 Casos de uso

- **CU01 – Modificar interfaz.** Se pueden modificar los parámetros del algoritmo en la interfaz de la aplicación.
- **CU02 – Ver reglas.** Se pueden ver las reglas generadas por el algoritmo.
- **CU03 – Lanzar IA.** Se lanza el algoritmo de estudio con la información y los parámetros que se han seleccionado.
- **CU04 – Generar ficheros.** Se generan ficheros con los datos, las reglas y demás salidas que se obtienen del algoritmo.
- **CU05 – Crear reglas.** Se ejecutará el algoritmo y se crearán reglas que se almacenan en memoria.
- **CU06 – Conectar APIs.** Se establece conexión con las APIs de las *smartband*.
- **CU07 – Obtener datos.** Se realizan peticiones a las APIs y se almacenan en memoria.

3.3 Diagrama de casos de uso

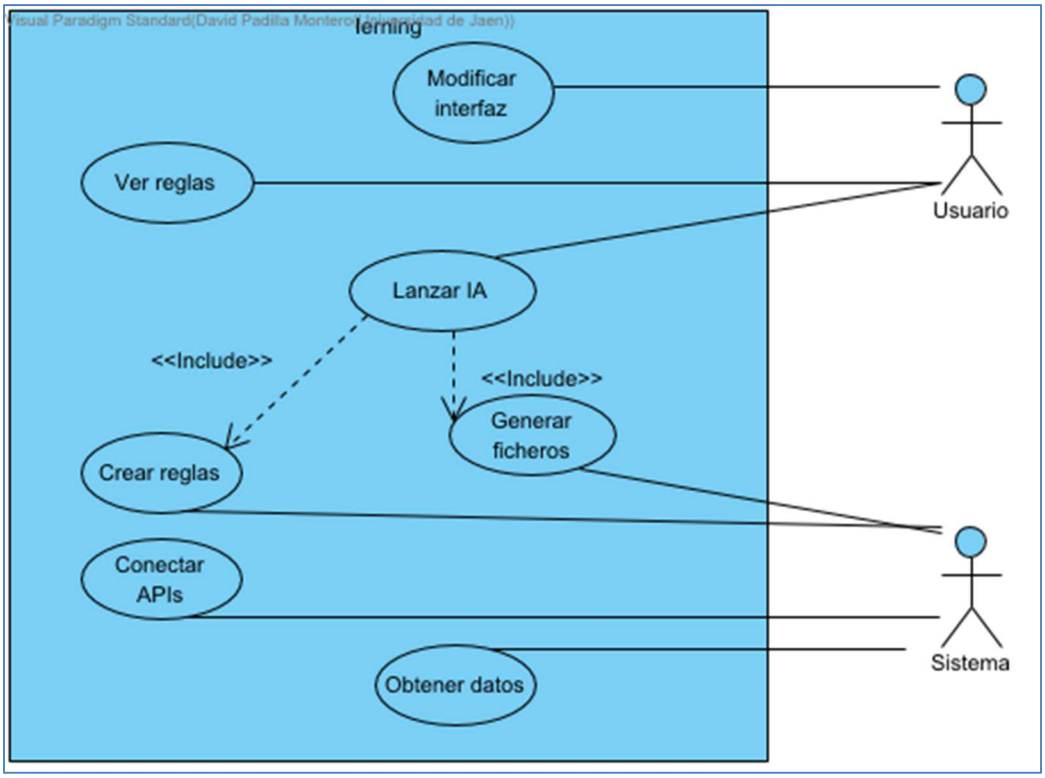

Figura 16.- Casos de uso. Fuente propia.

3.3.1 Modificar interfaz

Caso de uso	Modificar interfaz
Actor primario	Usuario
Sistema	Ierning
Nivel	Objetivo de usuario
Precondición	-
Flujo normal	
1	Seleccionar interfaz
2	Insertar nuevo número
3	Terminar
Flujos alternativos	
1.A	Selecciona días
1.B	Selecciona etiquetas

Tabla 9.- Análisis textual. Modificar interfaz.

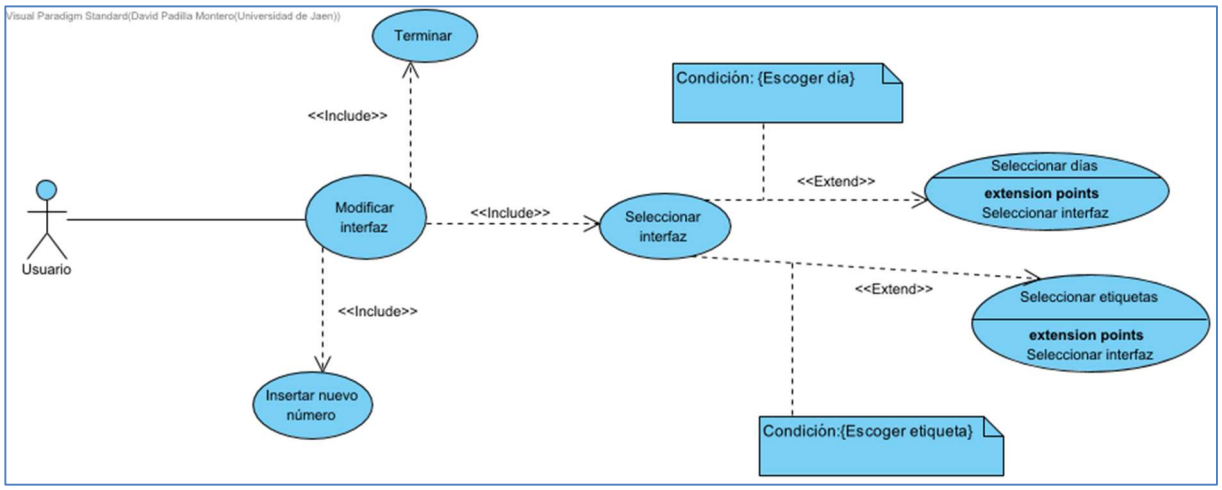

Figura 17.- Caso de uso. Modificar interfaz. Fuente propia.

3.3.2 Ver reglas

Caso de uso	Ver reglas
Actor primario	Usuario
Sistema	Ierning
Nivel	Objetivo de usuario
Precondición	Tener reglas
Flujo normal	
1	Pulsar botón
2	Obtener presentación de reglas
3	Terminar

Tabla 10.- Análisis textual. Ver reglas.

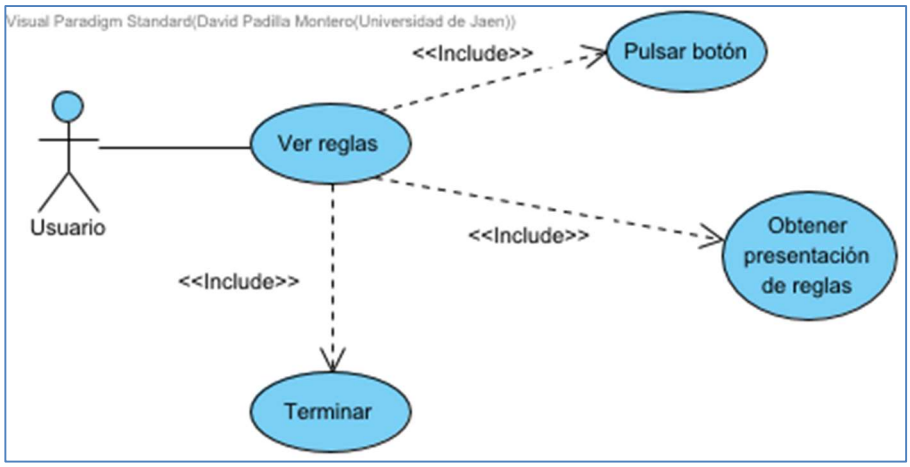

Figura 18.- Caso de uso. Ver reglas. Fuente propia.

3.3.3 Lanzar IA

Caso de uso	Lanzar IA
Actor primario	Usuario
Sistema	Ierning
Nivel	Objetivo de usuario
Precondición	Modificar interfaz
Flujo normal	
1	Presionar botón de IA
Flujos alternativos	
1.A	Mensaje de error

Tabla 11.- Análisis textual. Lanzar IA.

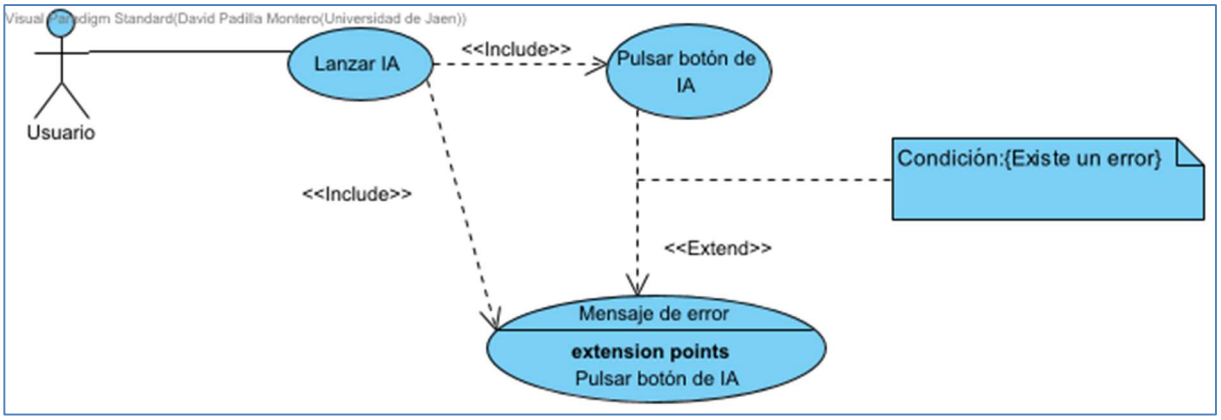

Figura 19.- Caso de uso. Lanzar IA. Fuente propia.

3.3.4 Generar ficheros

Caso de uso	Generar ficheros
Actor primario	Sistema
Sistema	Ierning
Nivel	Objetivo de sistema
Precondición	Haber ejecutado la IA
Flujo normal	
1	Recopilar resultados de la IA
2	Generar un escritor
3	Escribir los datos en memoria

Tabla 12.- Análisis textual. Generar ficheros.

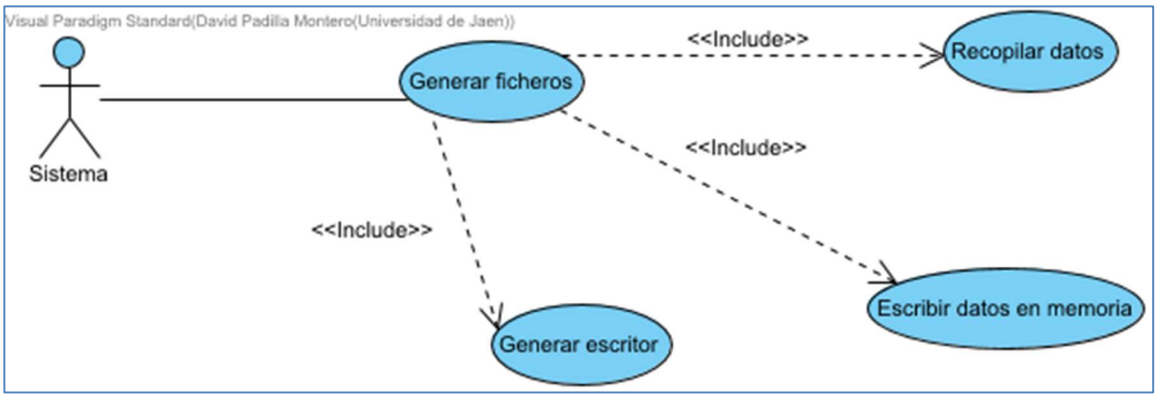

Figura 20.- Caso de uso. Lanzar IA. Fuente propia.

3.3.5 Crear reglas

Caso de uso	Crear reglas
Actor primario	Sistema
Sistema	Ierning
Nivel	Objetivo de sistema
Precondición	Tener datos en memoria
Flujo normal	
1	Preprocesar datos
2	Generar archivos ARFF
3	Ejecutar algoritmo de inteligencia artificial
4	Recopilar reglas
5	Guardar reglas
Flujo alternativo	
3.A	Si falla saltar a 5

Tabla 13.- Análisis textual. Crear reglas.

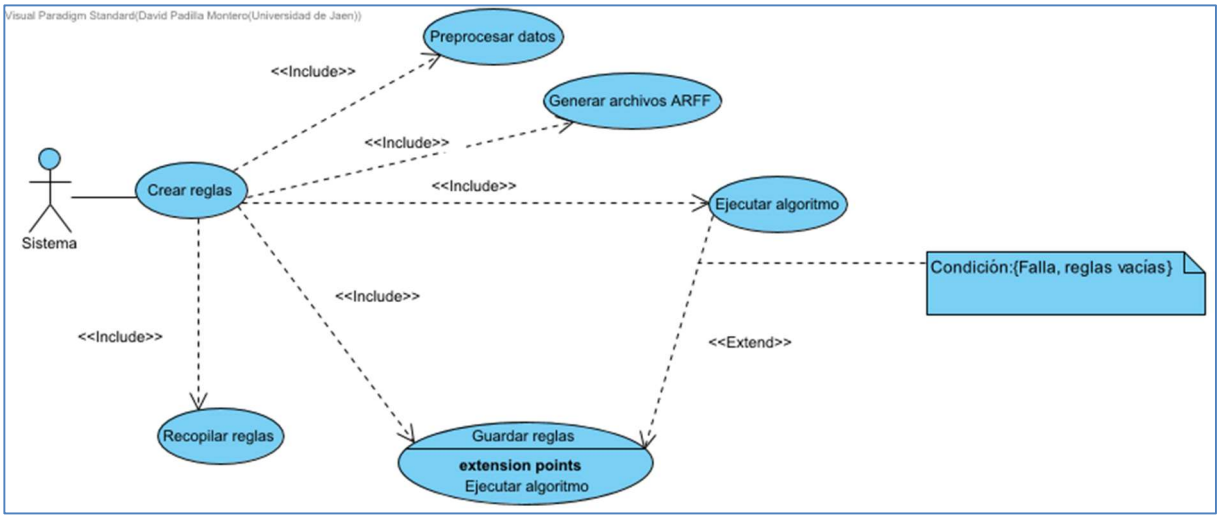

Figura 21.- Caso de uso. Crear reglas. Fuente propia.

3.3.6 Conectar APIs

Caso de uso	Conectar APIs
Actor primario	Sistema
Sistema	Ierning
Nivel	Objetivo de sistema
Precondición	Iniciar la app con conexión a internet
Flujo normal	
1	Enviar claves
2	Esperar respuesta
3	Comprobar resultados
4	Establecer conexión
5	Confirmar finalizado
Flujos alternativos	
2.A	No llega respuesta, saltar a 5.A
3.A	Resultado erróneo, saltar a 5.A
5.A	Error de confirmación

Tabla 14.- Análisis textual. Conectar APIs.

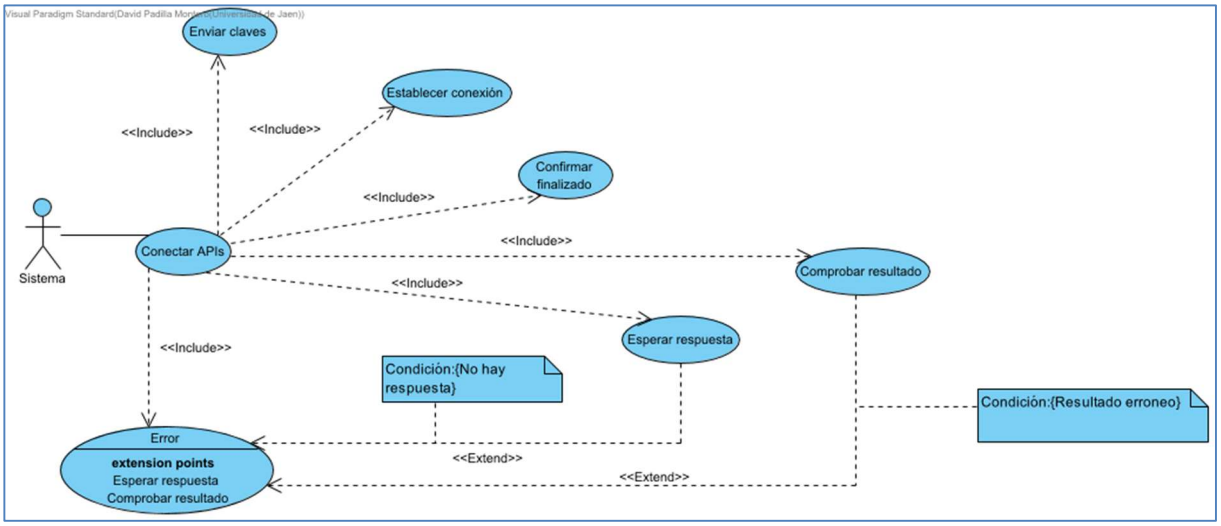

Figura 22.- Caso de uso. Conectar APIs. Fuente propia.

3.3.7 Obtener datos

Caso de uso	Obtener datos
Actor primario	Sistema
Sistema	Ierning
Nivel	Objetivo de sistema
Precondición	Tener APIs funcionales
Flujo normal	
1	Establecer fechas
2	Enviar petición a la API
3	Esperar resultados
4	Guardar datos
5	Finalizar
Flujos alternativos	
3.A	No hay resultados, saltar a 5

Tabla 15.- Análisis textual. Obtener datos.

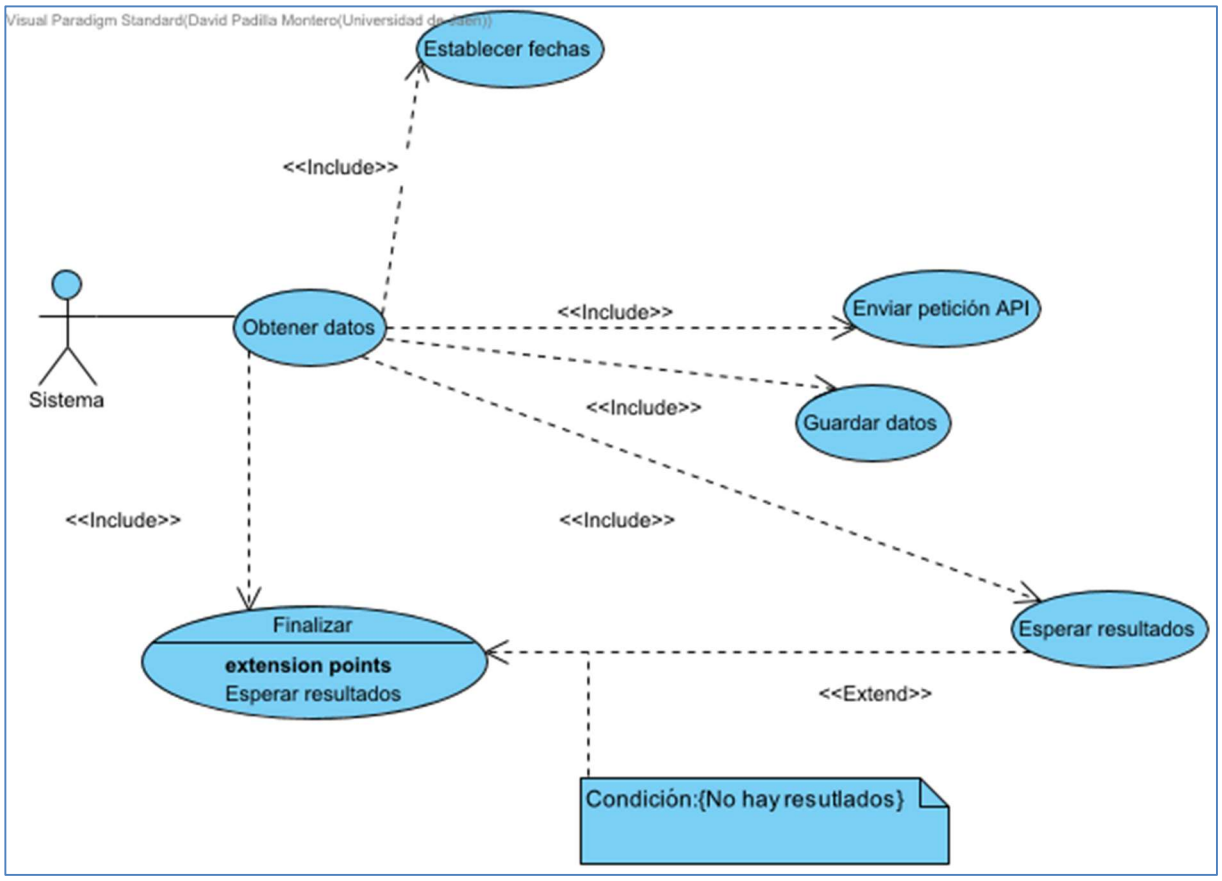

Figura 23.- Caso de uso. Obtener datos. Fuente propia.

3.4 Diagrama de secuencia y comunicación

Similar al apartado anterior 3.3, este apartado se va a exponer los diagramas de secuencia y sus respectivos diagramas de comunicación. Las tres parejas de diagramas que se expondrán son las referentes al usuario, a las *smartbands* y al algoritmo de inteligencia artificial que vamos a emplear.

3.4.1 Diagrama de secuencia de usuario

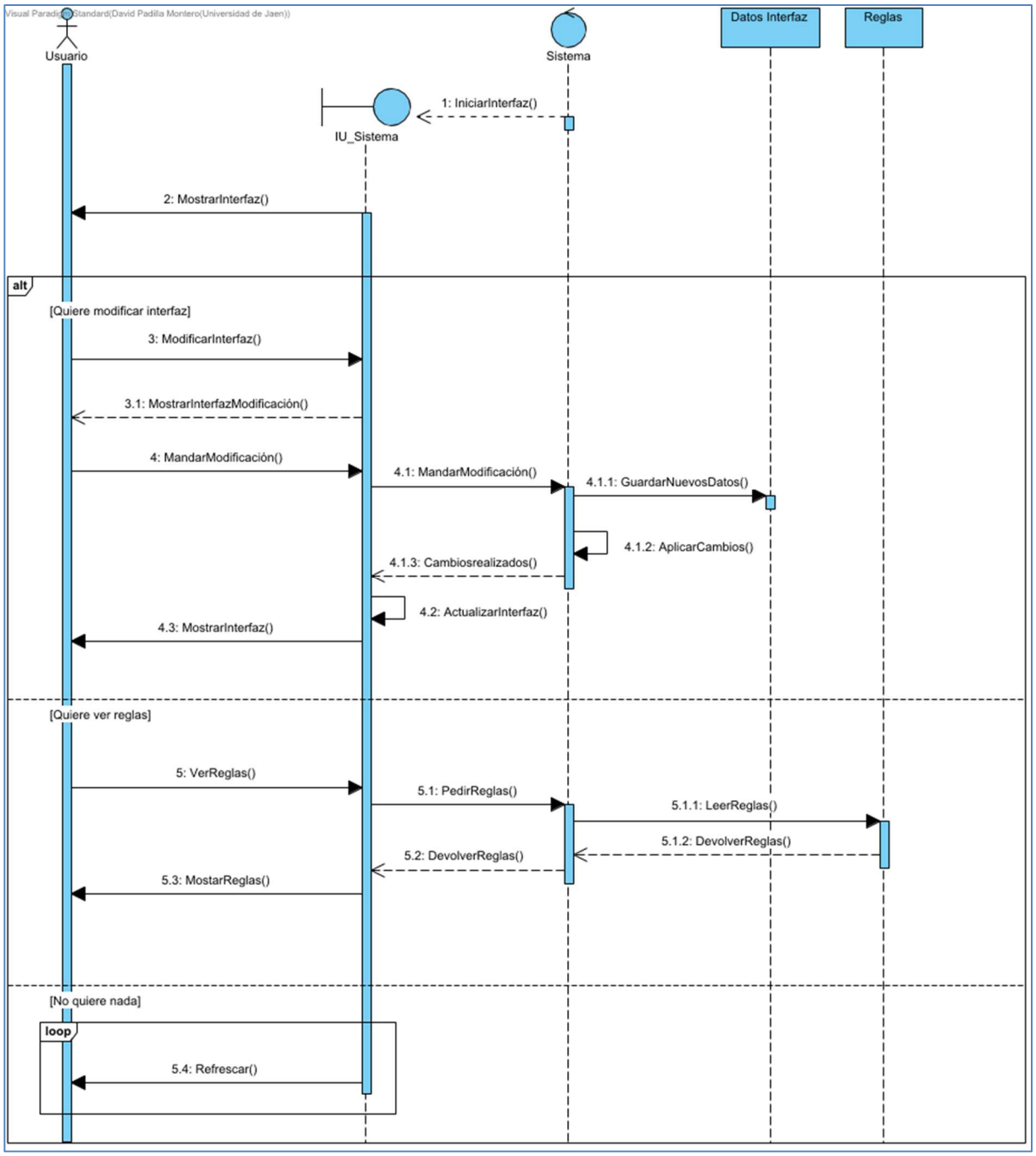

Figura 24.- Diagrama de secuencia de usuario. Fuente propia.

3.4.2 Diagrama de comunicación de usuario

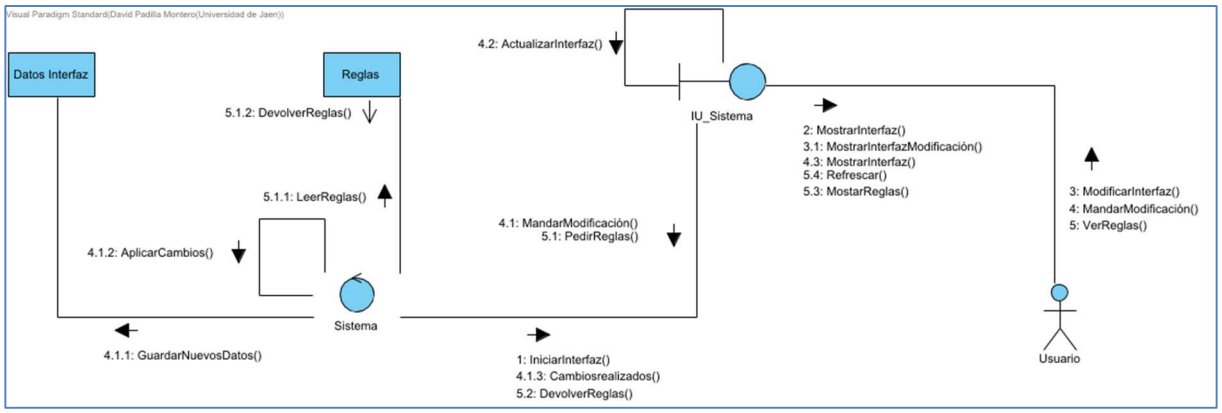

Figura 25.- Diagrama de comunicación de usuario. Fuente propia.

3.4.3 Diagrama de secuencia smartband

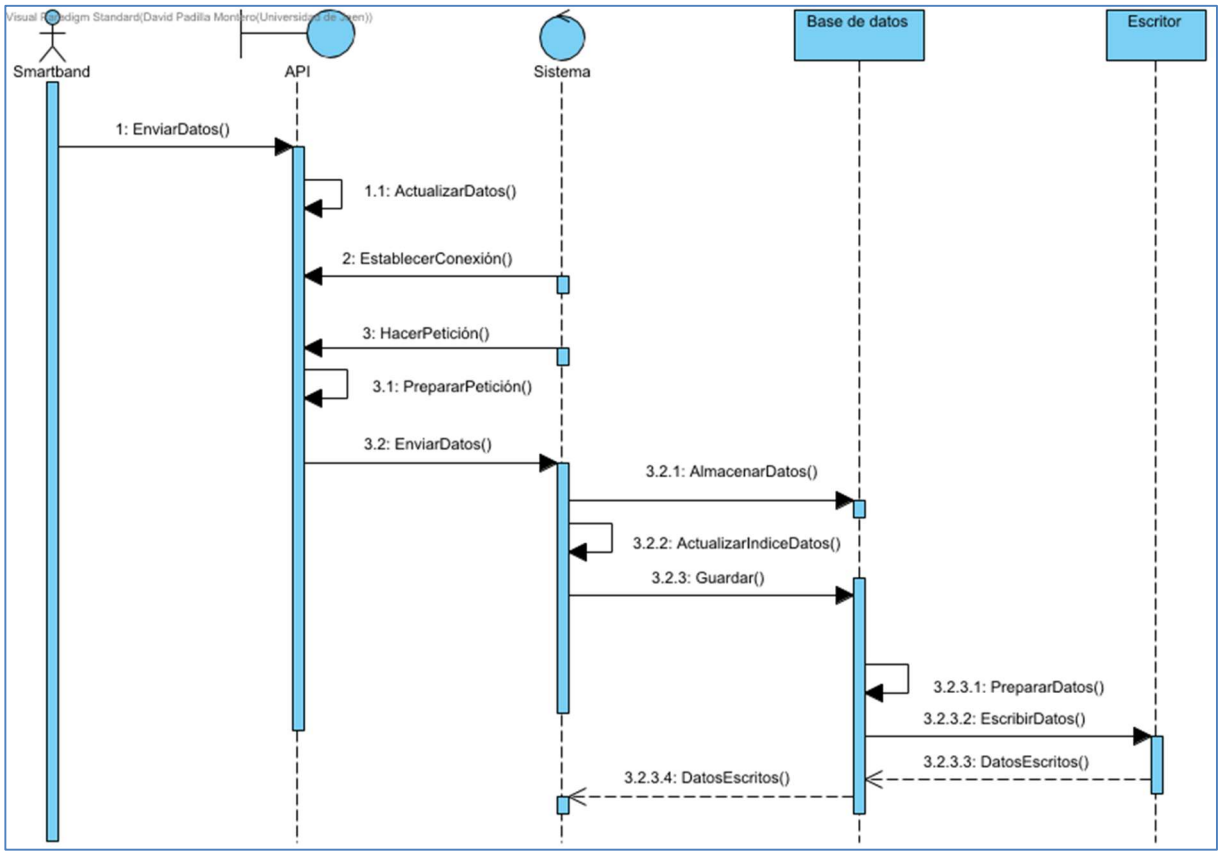

Figura 26.- Diagrama de secuencia smartband. Fuente propia.

3.4.4 Diagrama de comunicación smartband

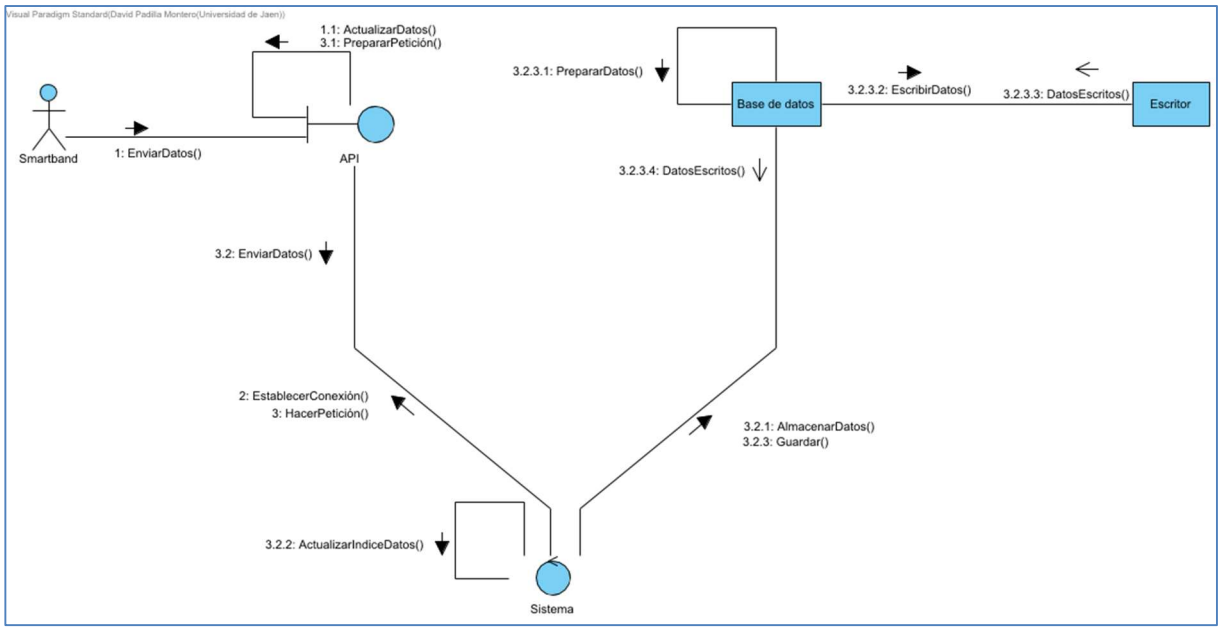

Figura 27.- Diagrama de comunicación smartband. Fuente propia.

3.4.5 Diagrama de secuencia algoritmo Inteligencia Artificial

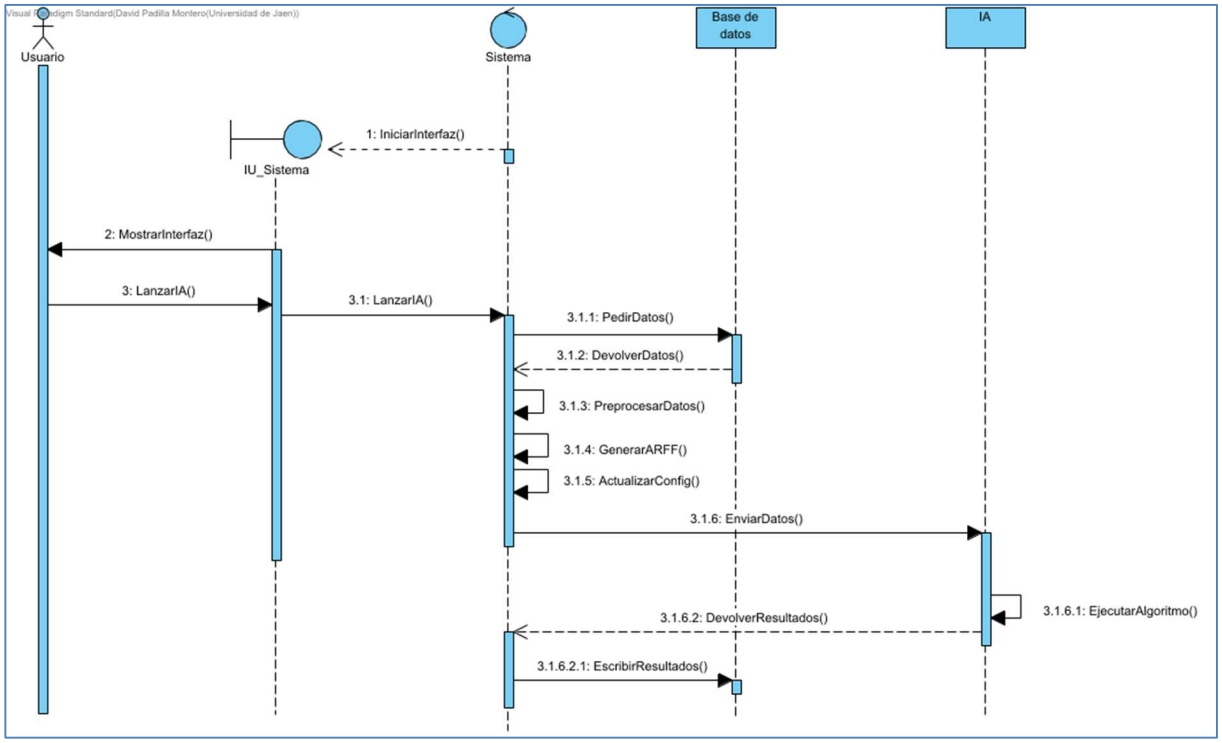

Figura 28.- Diagrama de secuencia algoritmo Inteligencia Artificial.

3.4.6 Diagrama de comunicación algoritmo Inteligencia Artificial

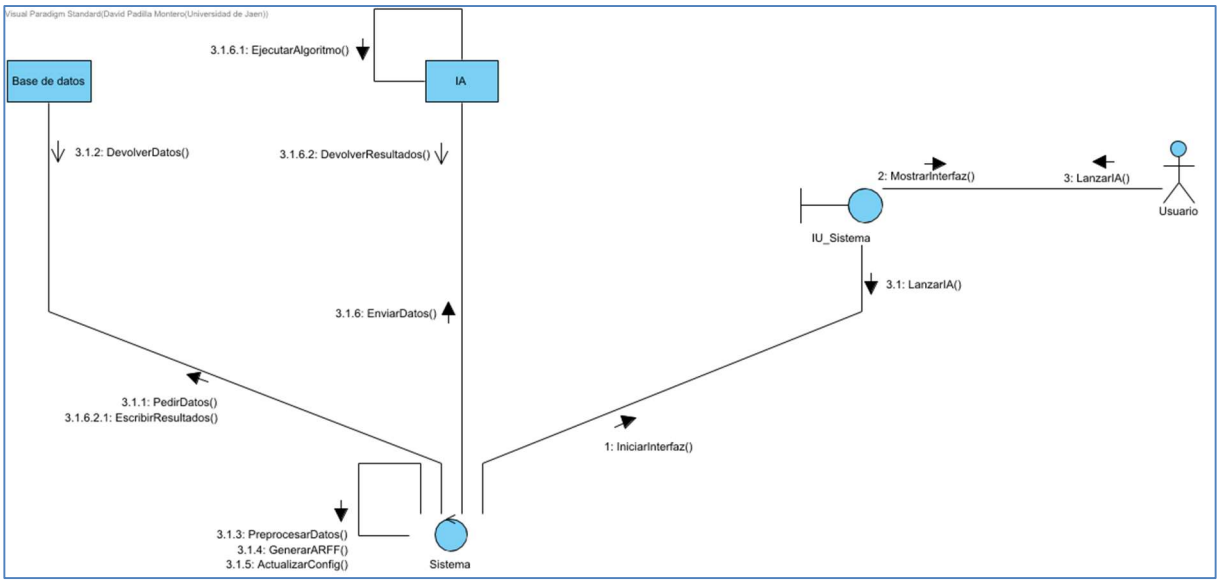

Figura 29.- Diagrama de comunicación algoritmo Inteligencia Artificial. Fuente propia.

3.5 Wireframe de la aplicación

En este apartado se tratará el aspecto que tendrá la aplicación, en concreto la vista principal de la Figura 30, ya que todo se gestiona principalmente desde ella.

Se debe tener en cuenta que esta será una aplicación usada para el estudio de datos, por ello va a ser una interfaz sencilla que disponga toda la utilidad a mano. Con ello en mente aparecen dos espacios de texto donde incluir el número de etiquetas y días que se van a utilizar, después cuatro diferentes botones. El primero junto a los espacios de texto para lanzar el algoritmo, el segundo para mostrar en pantalla las reglas, el tercero se encarga de escribir en la memoria accesible del teléfono las reglas que existen, es decir, se encarga de colocarla en una carpeta de uso compartido y no en la memoria propia de la aplicación. El último es el encargado de forzar la descarga de más de un día de datos en la aplicación.

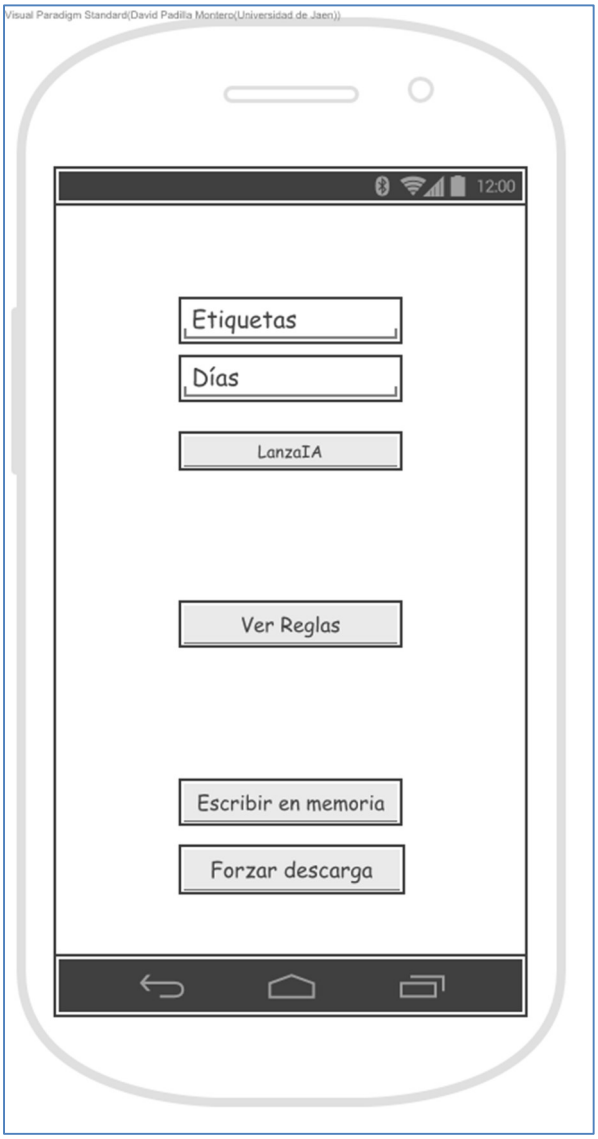

Figura 30.- Wireframe lerning. Fuente propia.

3.6 Extracción de requisitos

Se analizarán los requisitos por parte de la aplicación en cuestión, se busca un software que permita al usuario estudiar datos de manera fácil y correcta.

3.6.1 Requisitos funcionales

Respecto a los requisitos funcionales se ha visto una descripción de los mismos en los casos de uso en el apartado 3.3, de manera que quedaría resumido de la siguiente manera:

- **RF01 - Modificar la interfaz.** Permitirá modificar elementos de la interfaz.
- **RF02 - Ver reglas generadas.** Será capaz de obtener las reglas generadas en el algoritmo y mostrarlas en pantalla.
- **RF03 - Ejecutar el algoritmo.** Podrá ejecutar el algoritmo de inteligencia artificial.
- **RF04 - Crear ficheros de datos.** La aplicación permitirá almacenar en la memoria interna de la aplicación los ficheros con información.
- **RF05 - Utilizar las APIs.** Se podrá conectar y usar las APIs de las diferentes *smartbands*.
- **RF06 - Obtención de datos.** Obtendrá los datos de las *smartbands* y las almacenará para su uso.

3.6.2 Requisitos no funcionales

Se verán también los requisitos no funcionales de la aplicación que se pueden resumir en los siguientes:

- **RNF01 - Flexibilidad:** Se está buscando un trabajo a largo plazo, por lo que la aplicación tiene que estar adaptada a todo cambio posible ya que el mercado de bandas inteligentes y dispositivos Android está en constante cambio.
- **RNF02 - Eficiencia:** El objetivo de la aplicación es que funcione en dispositivos móviles por lo que es necesario consumir el menor número de recursos posible para ahorrar memoria y batería.
- **RNF03 - Consistencia:** El proyecto va orientado al uso de bandas inteligentes conectadas por Bluetooth, por lo que hace necesario que haya el menor número de errores posible y de haberlos que comunique al usuario qué ha ocurrido de manera sencilla y legible, además debe ser capaz de detectar dichos fallos y corregirlos.
- **RNF04 - Usabilidad:** Como se ha comentado anteriormente el proyecto va orientado a personas que usarán la aplicación con fines de estudio. Por lo tanto, debe permitir obtener los resultados a estudiar de manera cómoda, y poder exportarlos sin necesidad de programas externos.
- **RNF05 - Seguridad:** Se está tratando con información de usuarios que son de carácter personal, por lo que hay que almacenar esa información de manera segura y no pueda acceder cualquier persona.

- **RNF06 - Disponibilidad:** La aplicación deberá estar lo más disponible posible ya que nunca se sabe cuándo se va a usar y quien la va a usar.
- **RNF07 - Concurrencia:** Varias bandas inteligentes pueden estar conectadas simultáneamente por lo que la concurrencia será un aspecto a tener en cuenta.
- **RNF08 - Personal:** El proyecto está compuesto por un analista, un programador senior y un programador junior.
- **RNF09 - Software:** La aplicación estará disponible para Android.
- **RNF10 - Hardware:** Disponibilidad para cualquier dispositivo que sea capaz de llevar el software comentado anteriormente, disponga de bluetooth y de las bandas inteligentes.
- **RNF11 - Procesamiento local:** Los datos con los que se va a trabajar contienen información sensible del usuario, con diferente información personal, por lo que el procesamiento no debe requerir de envío de datos. Por ello se mantendrá toda funcionalidad dentro de la propia aplicación.
- **RNF12 - Control de fallos:** Los datos tomados de las *smartbands* son susceptibles a tener errores o tener franjas de tiempo vacías, ya sea por una pérdida de la comunicación de la misma o un fallo en los sensores. Se crearán medidas de preprocesamiento de información para eliminar estos posibles errores.
- **RNF13 - Multiplataforma:** Dada la amplia gama de *smartbands* que existen en el mercado la aplicación a desarrollar debe ser capaz de trabajar con diferentes marcas de los mismos sin que ello suponga un cambio en el método de funcionamiento de la aplicación.

3.7 Arquitectura del sistema

En la Figura 31 se puede ver un diagrama del funcionamiento del sistema y la relación que se genera a través de ellos, además de que ayudará a tener un mejor conocimiento del propio sistema.

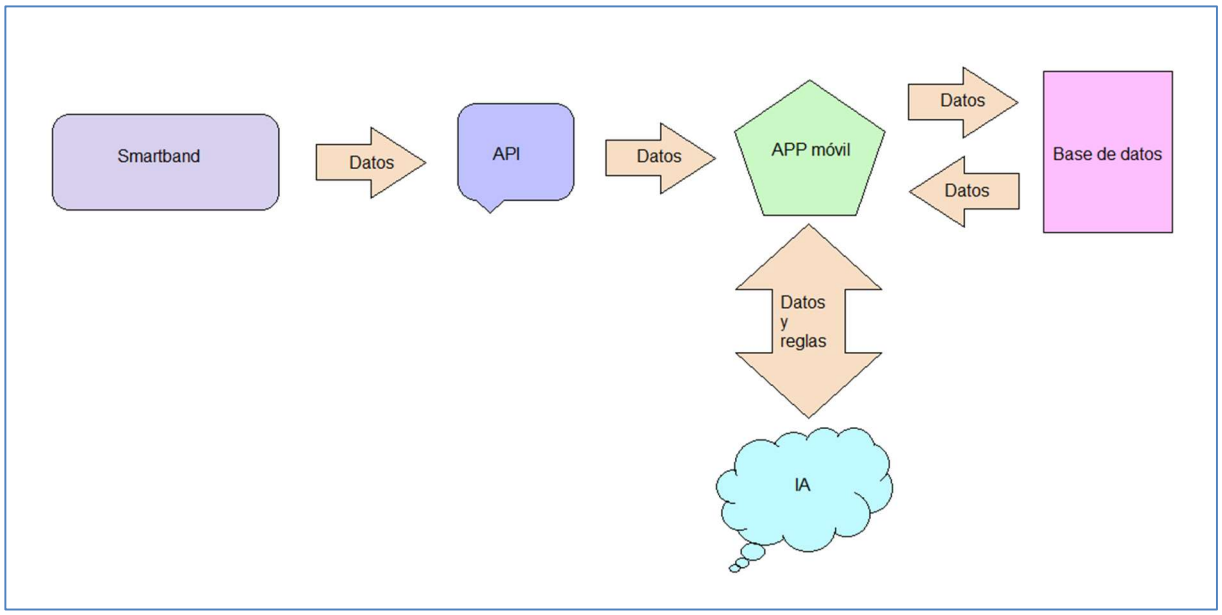

Figura 31.- Arquitectura del sistema. Fuente propia.

El funcionamiento del sistema está centralizado en la aplicación móvil lerning, que será la encargada de coordinar los datos recibidos, con los estudiados y la base de datos, además es quien da al usuario la interfaz con la que interactuar, por lo que será el eje central de la arquitectura.

Las *smartbands* son dispositivos externos al teléfono móvil, que serán usados como sensores, pero que son intrínsecamente necesarias en el sistema, ya que de ellas se van a obtener los datos y son el objetivo de estudio. Estos van a dirigirse hacia los programas de salud correspondientes, de allí pasa a las peticiones de las APIs. Estas facilitan la comunicación y dan acceso a la información de las *smartbands*, permitiendo incluirla posteriormente en la base de datos.

La base de datos será autogestionada a través de ficheros de texto. Debido a la potencia de los dispositivos móviles, incluir una base de datos más compleja puede suponer una ralentización del todo el sistema, por lo que, usar una más simple permite tener un acceso rápido y un control total sobre ella. Aunque esto incluye un aumento de dificultad, ya que depende de la propia aplicación el control de que esté funcional.

Por último, el algoritmo usado para la inteligencia artificial será el encargado de obtener reglas de conocimiento. La aplicación usará la información obtenida de las *smartbands* para dicha creación de conocimiento.

3.8 Análisis de datos y funciones de intercambio

Para poder explicar las funciones y datos usados es importante referenciar la Figura 32, donde aparece la estructuración de clases del proyecto lerning. La clase principal es `MainActivity`, ya que en Android es quien controla la funcionalidad de la aplicación, motivo por el cual muchas clases deben tener una asociación con ella. Esta asociación supone dar posibilidad a la clase de realizar funcionalidades que una clase normalmente no podría, como leer un archivo o escribir en uno.

La siguiente clase de gran tamaño es `HealthAIDataPackage`, la clase que funciona como base de datos y con ello debe de encargarse de controlar aquellos elementos que se almacenan.

En los siguientes apartados se verá en más detalle los principales TDA (tipos de datos abstractos) usados, y los métodos de apoyo con más importancia para ser analizados a continuación:

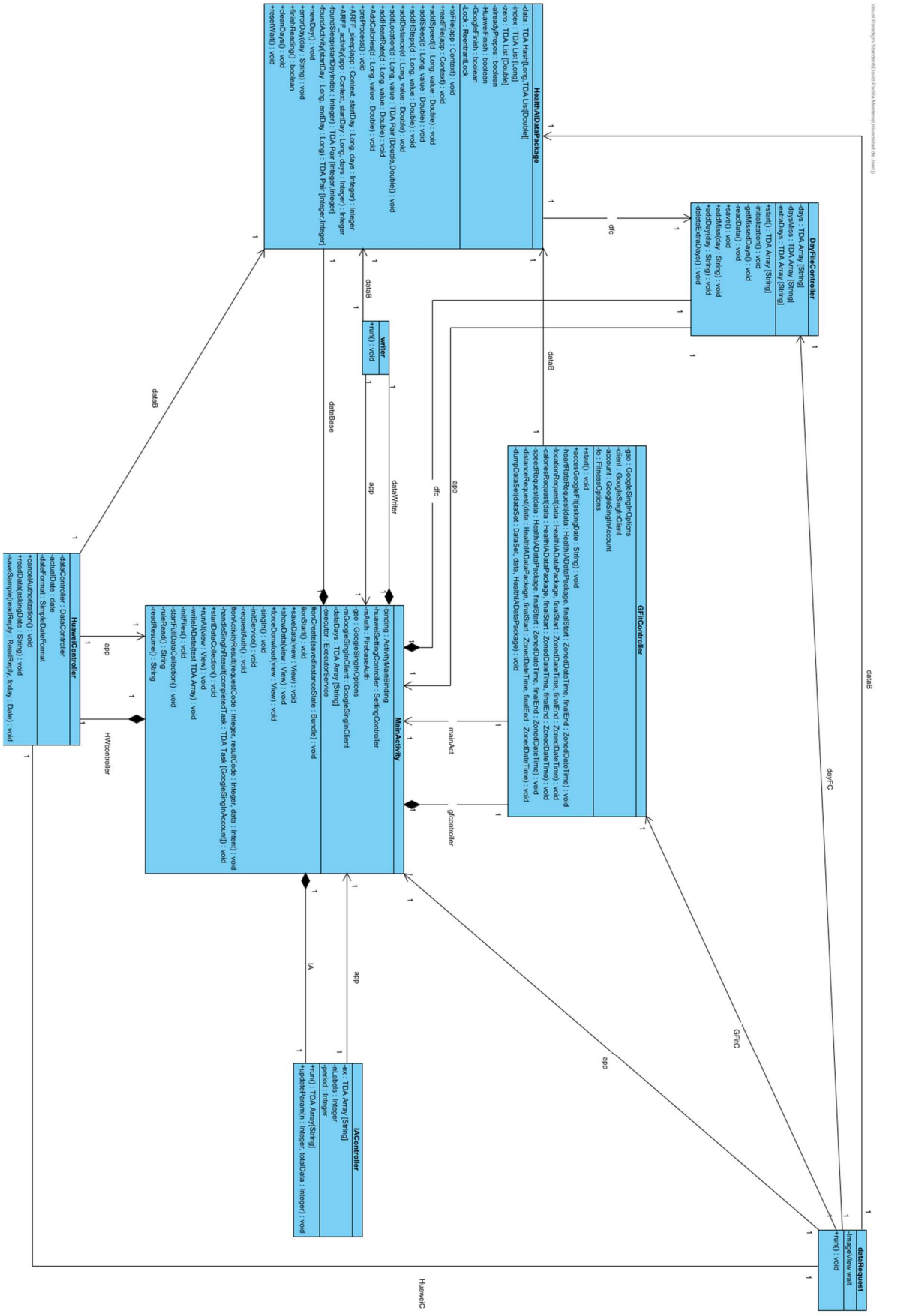

Figura 32.- Diagrama de clases lerning. Fuente propia.

3.8.1 Tipos de datos

`HealthAIDataPackage` – esta clase es la base de datos (BBDD) del proyecto. Lo primero será ver un ejemplo del archivo de texto donde se almacena:

```
Hour;Beats per second;1: light sleep, 2: REM sleep,3: deep sleep,4:
awake,5: nap;Steps;Steps Until Now;Metres;Metres until
now;latitude;Longitude;metres per second;calories;
```

Código 1.- Formato fichero base de datos.

La primera línea de texto, que ha sido partida para su mejor visualización, es la cabecera del archivo de texto "`IerningDB`", en el cual se indica la distribución de los datos durante el archivo. Explicando el modelo de distribución, se separa con el símbolo ";" entre diferentes atributos del TDA. Además, esta cabecera indica que tipo es y en qué unidad de medida se encuentra, a excepción del sueño, que puede poseer cinco valores diferentes, estos están marcados y separados por una coma. Los demás valores están en una unidad del sistema internacional y se dividen a su vez en diferentes contenedores que vamos a separar:

- **`TDA HashMap [Long, TDA Lista [Double]]- data`**: Este es el contenedor principal de la BBDD. La lista de `Doubles` almacena los datos en el orden que aparece en la imagen Código 1, mientras que el *Long* sirve de identificador en el `HashMap`. El primer campo es el tiempo donde se tomaron y está en formato *Unix epoch o POSIX*, que hace que ese TDA Lista sea inequívoco y único.
- **`TDA Lista [Long]- index`**: Contenedor auxiliar que funciona de índice para el mapa, así pues, contiene los mismos identificadores de "data" pero permite una ordenación rápida o una exploración por tiempo cargando la menor cantidad de información posible.
- **`boolean alreadyPrepos`**: Se necesita preprocesar la BBDD, dado que, al ser datos reales, algunos de ellos pueden venir defectuosos o con instancias de tiempo vacías, por lo que es necesario conocer si estos datos han sido ya pre procesados o no.
- **`boolean HuaweiFinish`**: Para garantizar que antes de comenzar la escritura de datos en fichero la API de Huawei ha terminado su petición.

- **`boolean GoogleFinish`**: Para garantizar que antes de comenzar la escritura de datos en fichero la API de Google ha terminado su petición.
- **`Lock lock`**: Bloqueo para la escritura en BBDD, dado que ambas APIs intentarán escribir simultáneamente hay que evitar que exista sobrescritura, ya que pueden leer el mismo bloque simultáneamente y escribir diferentes datos.

En primer lugar, el `HashMap`, como se había explicado anteriormente, se compone de un identificador que será el tiempo en formato *POSIX*. Este formato permite que el tiempo esté contabilizado en milisegundos, así que es fácil conocer cuando se ha almacenado. Un `HashMap` permite un acceso rápido al paquete de datos (lista de `Double`) teniendo solo que indicar un valor usado como llave.

En segundo lugar, la lista, que será usada como un índice. El motivo principal para usar un índice es el hecho de que no conocemos el orden en el que se guardarán las peticiones con datos.

El índice se utilizará para tener conocimiento de los elementos que existen en la base de datos, poder buscarlos y ordenarlos. Gracias al índice, es posible buscar de manera ordenada en tiempo, sin cargar el resto de información, ya que, a excepción del algoritmo, el interés para la aplicación es la fecha de toma de los sensores.

Es también de interés explicar la selección de la lista como clase contenedora. Debido a que la clase HealthAIDataPackage no tiene conocimiento sobre cuántos días se deben almacenar o cuantos hay actualmente, ese trabajo es de la clase `DayFileController`. Por ello, el uso de una lista más simple permite almacenarlos ordenados, que posteriormente `DayFileController` eliminará el exceso según sea el límite de días establecido. Además, no solo se eliminan del índice, sino también de la base de datos, por lo que este borrado debe hacerse simultáneamente.

`DayFileController` – Esta clase se encarga de controlar los días que actualmente se han pedido a las APIs y están almacenados en la BBDD.

```
2022-05-29
2022-05-30
2022-05-31
2022-06-01
```

Código 2.- Formato fichero días.

Se controlan los días que están almacenados en memoria y con ello se puede saber qué tipo de peticiones se deben hacer o cuáles no. El archivo "`IerningDAY`" muestra el formato de la imagen Código 2, los días que se tienen actualmente almacenados en memoria, además dispone de algunos contenedores como:

- **`TDA Lista [String]- days`**: Contiene los días que actualmente están en memoria y cuya petición a las APIs ha sido ya realizada. Con ello se conoce la cantidad de datos que hay y si se ha terminado de completar la base datos.
- **`TDA Lista [String]- daysMiss`**: Conociendo el número de días total que se deben tener, se va a almacenar en este contenedor los días que aún no se han almacenado en memoria y, por lo tanto, no se dispone de datos.
- **`TDA Lista [String]- extraDays`**: En caso de que existan días de más, se almacenarán en este contenedor. Para evitar una saturación de la memoria solo se mantendrán un número determinado de días en memoria de manera variable, por lo que este control es necesario mantenerlo.

3.8.2 Procedimientos de apoyo

Este apartado contiene algunas de las funciones y procedimientos más importantes utilizados durante el proyecto, pudiendo ver su relación en la Figura 33, los cuales son:

1. Para los datos
    a. **`toFile (contexto de la aplicación):`** Esta función se encarga de crear el fichero de la base de datos. Escribe el fichero "IerningDB" con el formato que se mostraba en Código 1.
    b. **`preProcess ():`** Preprocesa la BBDD, para evitar que aparezcan datos vacíos ya que, al ser datos reales, los sensores pueden (y van a) fallar, por lo que aparecen estos vacíos que no se pueden utilizar.
    c. **`foundSleep (entero):`** Este procedimiento busca toda la etapa de sueño que existe dentro de un día seleccionado, devolverá un marcador con el inicio y el final de la etapa de sueño.
    d. **`foundActivity (fecha epoch, fecha epoch):`** Función que encuentra las etapas de actividad existentes dentro del periodo seleccionado y devolverá una lista con los tiempos donde hay actividad.

e. **`cleanDays ():`** Comprueba si existe un excedente de días en memoria y los datos correspondientes a estos, se encarga de eliminarlos.

f. **`newDay ():`** Marcador del inicio de cada día, establecido antes de cada petición y colocando un patrón reconocible para el preprocesado.

2. Para el Algoritmo

a. **`ARFF_sleep (contexto de la aplicación, fecha epoch, entero):`** procedimiento de preparación de datos para el algoritmo. Dado que el algoritmo utiliza ficheros ARFF prepara el fichero con la información en este formato, solo toma el estudio de sueño.

b. **`ARFF_activity (contexto de la aplicación, fecha epoch, entero):`** procedimiento de preparación de datos para el algoritmo. Dado que el algoritmo utiliza ficheros ARFF prepara el fichero con la información en este formato, solo toma el estudio de actividad.

3. Para las Apis

a. **`readData(fecha):`** Petición a la API de Huawei referente al día seleccionado.

b. **`saveSample (respuesta, fecha):`** Una vez la API de Huawei ha terminado, se debe almacenar la BBDD usando sus métodos de añadir correspondientes al tipo de dato.

c. **`accessGoogleFit (fecha):`** Petición a la API de Google referente a un día seleccionado.

d. **`dumpDataSet (set de datos, HealthAIDataPackage):`** Una vez la API de Google ha terminado la petición, se debe almacenar en la BBDD usando sus métodos de añadir correspondientes al tipo de dato.

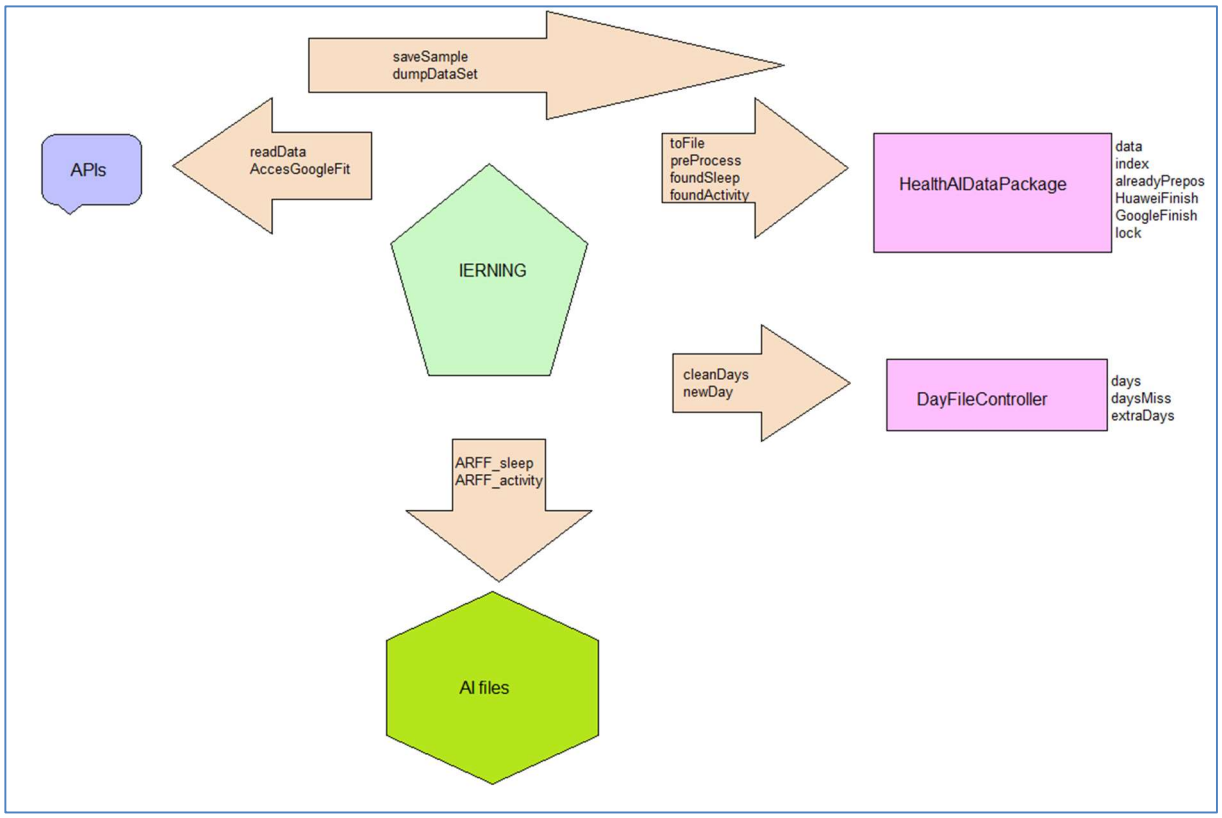

Figura 33.- Diagrama funciones apoyo. Fuente propia.

3.9 Algoritmo SDATEP

El algoritmo que se ha desarrollado recibe el nombre de SDATEP*, Smartbands drift analysis through emerging patterns*. En este apartado se comentará qué tipo de algoritmo se utiliza como base, por lo tanto, requiere un estudio para comprender su funcionamiento y así poder utilizarlo correctamente.

En este estudio se podrán ver diferentes apartados, un estudio inicial del estado del arte actual, seguido después de explicaciones de partes integradas en el algoritmo, así como el sistema de reglas difusas, los patrones emergentes, cromosomas y su representación o qué es un EFS.

Este algoritmo es una adaptación a la minería de cambio de concepto del proyecto FEPDS de Ángel M. García-Vico, Cristóbal J. Carmona, Pedro González, Huseyin Seker y María J. del Jesus. En concreto de su presentación el 16 de junio del 2020, es una implementación realizada en Java disponible en el repositorio Git Hub de Ángel M. García-Vico [11].

3.9.1 Estado del arte actual

La minería de datos es todo el procedimiento de extracción de conocimiento o patrones útiles en grandes volúmenes de datos. Hay muchas utilidades para la minería de datos, por ejemplo, para IBM “es un método innovador de aprovechar la información ya existente a fin de mejorar procesos, mejorar el rendimiento u optimizar el uso de recursos” [12].

Tradicionalmente en la minería de datos existen dos posibles formas de acercarse, el aprendizaje supervisado y el no supervisado. Existen los algoritmos de descubrimiento de reglas descriptivas (*Emerging Pattern Mining*) que se encuentran a medio camino entre la predicción de los algoritmos supervisados y la descripción de los algoritmos no supervisados. Esta descripción debe ser generalizada, simple y tan precisa como sea posible, además aquí aparece una variante en esta minería conocida como minería de cambio, *change mining*. Estudia la obtención de reglas descriptivas frecuentes en un conjunto de información y descubrir los cambios o anomalías que aparecen a lo largo del tiempo. Por ese motivo, este tipo de minería es útil con un flujo constante de información [13] [14] [15].

Es importante comentar qué es un *data streaming* (o flujo de datos en español), siendo un tipo especial de datos potencialmente infinitos, que poseen una indicación de tiempo y que van a llegar al algoritmo de manera continuada con una velocidad variable. Siendo así un campo de información cargado de conocimiento, donde *change mining* es capaz de obtener un alto nivel de utilidad [16].

Los algoritmos que utilizan un sistema difuso evolutivo multiobjetivo establecen un alto nivel de explotación y exploración de los datos durante el proceso de búsqueda. El sistema difuso evolutivo multiobjetivo, es robusto frente a cambios pequeños y tiene una buena adaptación para los desvíos que pueden existir. Se ha decidido utilizar este sistema ya que se deben de optimizar varios objetivos, tales como compresión, generalidad, precisión o interés [17].

3.9.2 Sistema de reglas difusas

Los sistemas de reglas difusas o FRBS (*Fuzzy rule-based system*), son una extensión de los sistemas de reglas clásicos que se expresan de la forma “SI A

ENTONCES B" donde A y B son conjuntos difusos. A y B son llamados antecedente y consecuencia de la regla respectivamente [18].

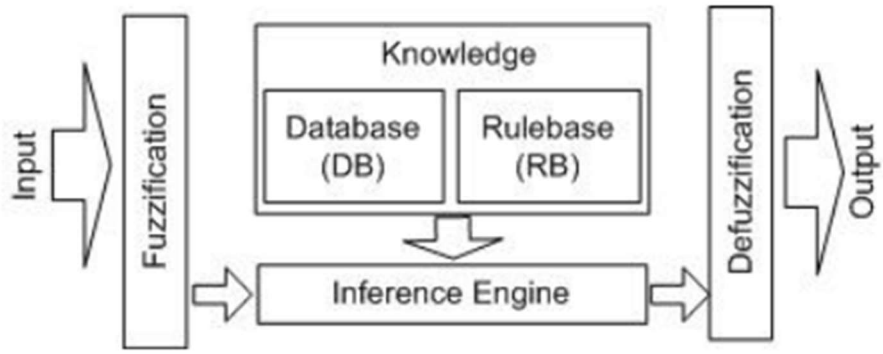

Figura 34.- Representación de un sistema de reglas difusas. Fuente Digiburg UGR.

El conocimiento se obtiene de dos componentes principales: la base de reglas o *Rulebase* (RB) y los conjuntos difusos de la base de datos (DB), Figura 34. Para la construcción de un FRBS existen dos posibles estrategias.

La primera, donde se necesita información de un experto que define manualmente el conocimiento del FRBS, aunque esto hace que este método casi nunca sea posible, no solo por la dificultad del conocimiento preciso de los datos sino la disponibilidad de un experto. El segundo método se realiza de la extracción del conocimiento a través de los datos usando métodos de aprendizaje.

Aquí aparecen entonces los EFS (*Evolutionary Fuzzy System*), sistemas evolutivos difusos, que están en la rama de los FRBS donde se obtiene conocimiento a través de los datos.

Todo este sistema de FRBS se basa en el uso de la lógica difusa [19], la cual permite tomar cierto grado de decisiones con distinta intensidad, tal y como hace el ser humano, es decir, es posible indicar "no está muy lejos" y supone una expresión que permite entender la distancia, pese a no indicar una distancia exacta.

Con ello la lógica difusa permite diferenciar entre cuantificadores, los cuales en el algoritmo serán llamadas *Labels* o etiquetas y supondrán la cantidad de divisiones del dominio de los valores, es decir, si se usan tres etiquetas se permite hacer tres diferentes divisiones, poco, medio y mucho.

3.9.3 Adaptación al cambio

En el apartado 3.9.1 se introdujo el concepto de *data stream*, pero se va hablar un poco más extendido del mismo.

Debido a que un *data stream* es una continua llegada de datos, esto genera diferentes problemas, por ejemplo, la necesidad del procesamiento, ya que pueden venir de diferentes dispositivos, sensores, tipo de conexiones, etc. Además, surgen otro tipo de dificultades a tener en cuenta, como su heterogeneidad, su imperfección (pérdida de datos) o el hecho de que sean irrepetibles. Todo eso son dificultades que se añaden, pero gracias a ello, es posible disponer de un alto volumen de información que llega constantemente.

Debido al alto volumen de datos, los algoritmos de *data mining* deben de estudiar estos en instancias, separándolos en paquetes donde sea posible realizar un estudio sin que esto suponga un tiempo excesivo. Con ello es posible que el modelo aprendido con paquetes anteriores no sea totalmente útil en el siguiente, provocando un error.

Este error es debido al cambio de concepto, *concept drift*, lo cual significa que la distribución existente en los datos ha sufrido una variación y provoca que los modelos aprendidos hasta ahora fallen. Como aparece en la Figura 35, es posible que se dé una modificación real o virtual [20].

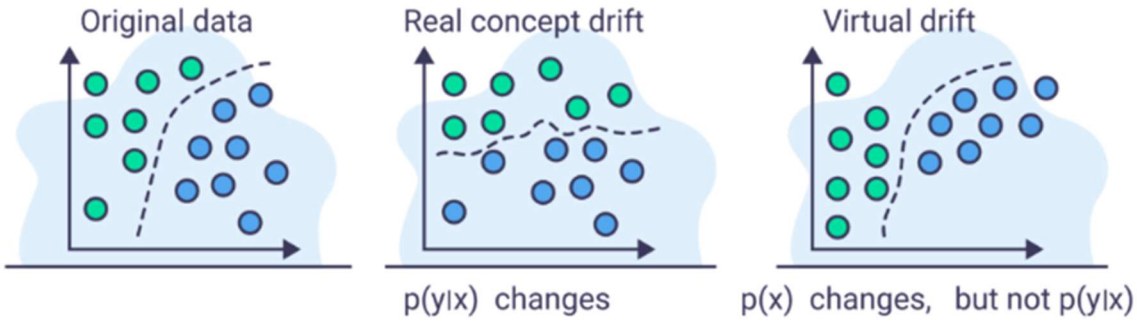

Figura 35.- Cambios de concepto. Fuente Aporia Blog.

Para enfrentar dicho problema se utilizan diferentes elementos para poder trabajar correctamente y modificar el comportamiento de los modelos en función de los posibles cambios, tales como:

- Memoria: donde el algoritmo va olvidando datos antiguos mientras que aprende de las nuevas, esta está definida como *Windowing*, donde la información y comportamientos del modelo se consideran más relevantes cuanto más recientes son.
- Proceso de aprendizaje: se introducen los mecanismos mencionados de olvido y se controla la complejidad del sistema para poder dar una respuesta rápida. Además, se extrae el conocimiento de los datos durante ese proceso y este evoluciona para descartar conceptos obsoletos.
- Monitorización de cambios: aparecen los sistemas de apoyo del proceso de aprendizaje para la detección de los posibles *concept drift*.

3.9.4 Proceso de aprendizaje del algoritmo

Se parte del algoritmo de obtención de patrones, EFS sobre MOA *(Massive Online Analysis*) [21], donde dichos patrones EP que se obtienen deben de ser relevantes, es decir, que tengan información de utilidad para un experto, también deben de tener la mayor precisión posible para que no dé posibilidad a errores y debe de cubrir la mayor cantidad posible de datos e instancias de una clase. Además, se realizan diferentes modificaciones para su adaptación a *Change Mining*, en espacios de una cantidad reducida de muestras.

3.9.4.1 Codificación de los cromosomas

El algoritmo obtiene los patrones, EP, usados y los representa usando el método DNF (forma normal disyuntiva), que puede ser explicada como:

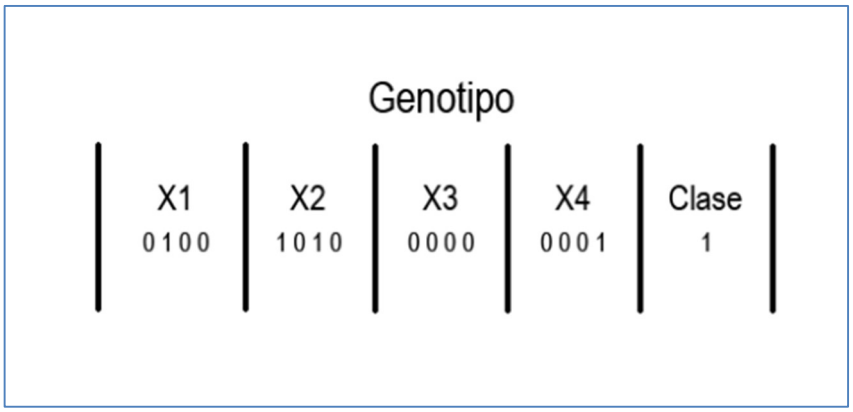

Figura 36.- Representación DNF. Fuente propia.

Una representación de conocimiento con reglas DNF es una forma de conocimiento compacta y flexible que permite la generación y extracción de reglas

más generales. Una representación de una cadena binaria de longitud establecida, donde cada bit corresponde a cada uno de los posibles valores, siendo el valor 0 para indicar que ese valor no será utilizado y el valor 1 para la indicación de que sí lo será, como aparece en la Figura 36.

3.9.4.2 Operadores genéticos

Los operadores que se utilizan durante el algoritmo son:

- Inicialización: Primero se incorpora el actual modelo de reglas del sistema para que pueda ir evolucionando y adaptándose, posteriormente se añaden el resto de individuos de la población de manera aleatoria hasta que dicha población esté completa.
- Selección: Se realiza una selección de torneo binario clásico, donde se toman dos individuos aleatorios y se enfrentan, solo seleccionando al mejor de ellos.
- Cruce: Cruce en dos puntos.
- Mutación: Se tienen dos tipos de mutaciones, una de ellas cambia aleatoriamente un bit del cromosoma, mientras que la otra elimina completamente una variable entera.

3.9.4.3 Método de reinicio

El mecanismo de reinicio se utiliza para evitar caer en un máximo local, con ese objetivo el método de reinicio se lanzará cuando en el frente se de alguno de estos dos casos: cuando no ha evolucionado o cuando no ha sido capaz de cubrir nuevos ejemplos durante el veinticinco por ciento del total de las evaluaciones realizadas. Además, también se realiza durante los procesos finales del modelo [22].

Aquellos individuos que hayan pasado el criterio de no dominancia y la competición de tokens se incorporarán, los restantes pueden tener interés a la hora de explorar zonas no cubiertas, por lo que se utilizarán para una nueva población que se encargará de realizar una exploración espacios de búsqueda no cubiertos anteriormente.

3.9.5 Esquema operacional

Durante este apartado se explicará cómo es el funcionamiento del algoritmo, siendo un ejemplo resumido la Figura 37.

```
1    entrada: M - instancia de datos
2
3    salida: fPS - modelo de patrones
4  ------------------------------------------------
5    SI batch no lleno HACER
6         Guardar en batch(M)
7    FIN SI
8    SINO
9
10   inicializarBaseDatos(batch)
11   generacion <- 0
12   Pg <- inicializar(datos)
13   Evaluar(Pg)
14
15   MIENTRAS generacion < maxGen HACER
16        Offg <- OperadoresGenéticos(Pg)
17        Evaluar(Offg)
18        Rg <- Unión (Pg,Offg)
19        F <- OrdenaciónDominancia(Rg)
20        SI F no evoluciona ENTONCES
21             Pg+1 <- reinicialización()
22        FIN SI
23        g <- g+1
24   FIN MIENTRAS
25
26   fPS <- CompeticiónDeTokens(F)
27   evaluarfPS()
28   escribirfPSMemoria()
29   FIN SINO
```

Código 3.- Pseudocódigo SDATEP.

El Código 3 se repite por cada nueva instancia de datos que llega, se va a explicar que ocurre durante su ejecución. De las líneas 5 a la 7, se realiza una condicional donde se van a recolectar. Estos datos deben de llenar un *batch*, que será la medida de tamaño para un bloque. Este *batch*, deberá de llenarse antes de continuar el algoritmo.

Si el condicional no se cumple, se preparará la generación 0. De las líneas 8 a 13, se realizará el proceso de inicialización de esta generación, además de evaluarse inicialmente dicha población.

Posteriormente se pasará de las líneas 14 a la 23, donde se utiliza un bucle generacional, es decir, se va a repetir el algoritmo evolutivo hasta una condición de parada. Los operadores genéticos comentados en el apartado 3.9.4.2, se realizan en este orden, selección, cruce y mutación, posteriormente en la línea 16 se evaluará esta población tras los operadores genéticos. Los objetivos de esta evaluación son WRAcc_Norm, Confidence y TPR, que aparecen explicados en el apartado 5.3 de

esta memoria. La población ya evaluada tendrá una unión con la población inicial del bucle y ordenación de no dominado, con ello se obtendrá la población F.

Si el frente de pareto no evoluciona, se obtendrá una nueva población con una competición de tokens. Finalmente, después del bucle generacional se realiza una última competición de tokens y se obtienen las reglas o patrones, que se van a almacenar en memoria y estas serán las reglas que se obtendrán de ese *batch* de estudio.

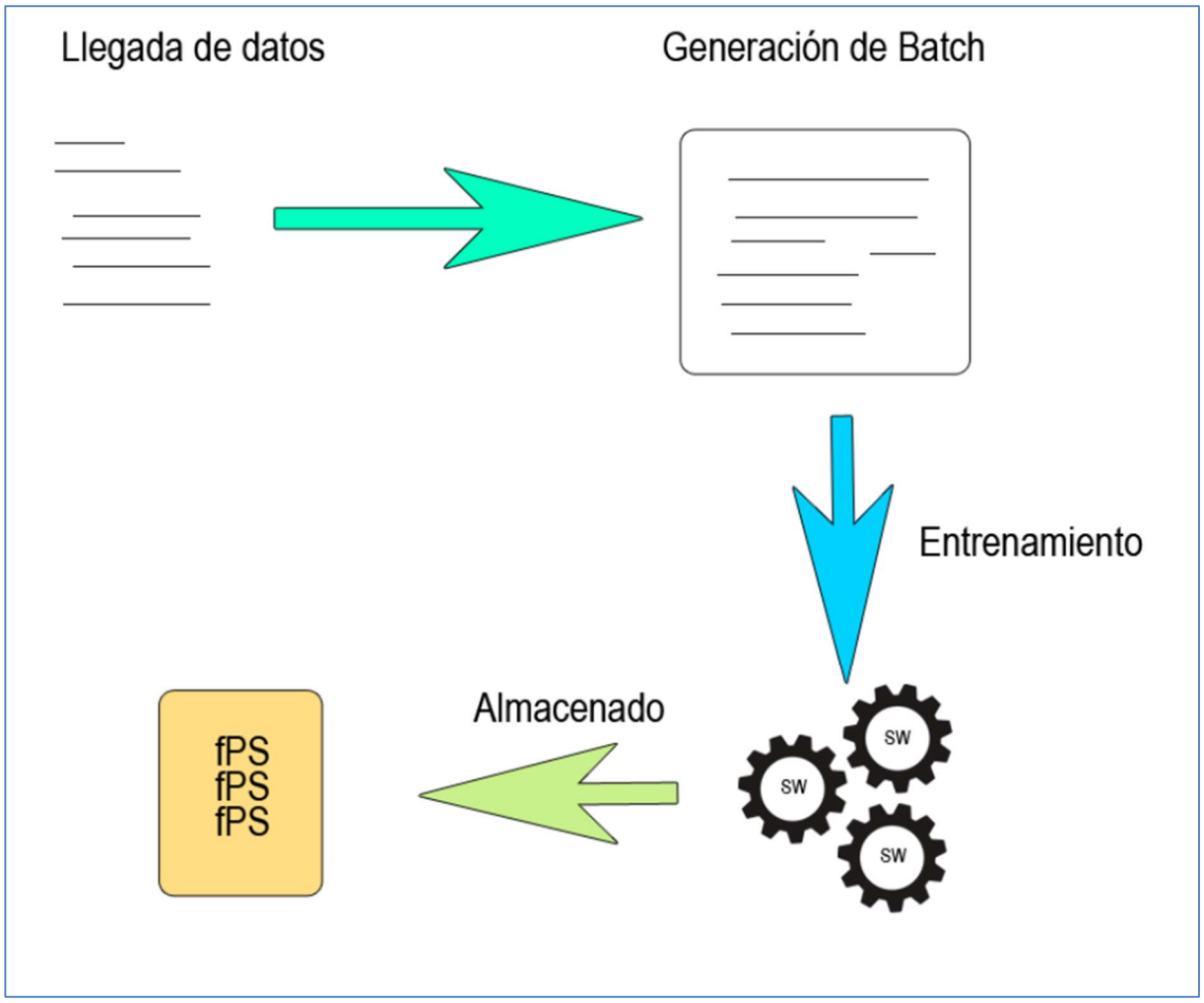

Figura 37.- Funcionamiento SDATEP. Fuente propia.

Capítulo 4: Implementación

En este apartado se va a mostrar cómo se ha realizado la implementación del proyecto y que puntos importantes deben ser comentados, así como, las partes donde se ha debido dedicar más tiempo y cuidado.

4.1 Toma de datos y almacenamiento

Se va a tratar uno de los temas principales del proyecto, la toma y recogida de datos para el análisis, ya que, sin ellos, la posibilidad de realizar un estudio es nula. Con ello se va a dividir qué es y cómo se trata la información, la toma y su posterior procesamiento.

4.1.1 Peticiones de datos

La petición de datos se realiza sobre las dos APIs que se están usando actualmente, Huawei y Google/Xiaomi, sobre sus diferentes aplicaciones de salud.

Ambas peticiones se realizan de manera similar. A las APIs se les asigna un día sobre el que deben realizar una petición. Se tomó de medida un día, ya que, al realizar pruebas de toma de datos se ha comprobado empíricamente que era uno de los periodos más grandes que se podían tomar, sin que ello repercutiese en la calidad o cantidad de información que se estaba obteniendo. Siendo así, se prepara la llamada a la API pidiendo los permisos adecuados, ya que en caso de Huawei estos permisos requieren un protocolo de aceptación por parte de la compañía, así que solo se accederán a los datos cuyo permiso está ya adquirido.

Después de enviar la petición de forma asíncrona, llegará una respuesta que se almacena en la memoria y posteriormente se guardará en la base de datos utilizando una clase auxiliar de escritura. Esta clase escribirá en memoria en segundo plano, por motivos de peso y tamaño, puede tardar desde varios segundos hasta varios minutos y puede suponer molesto que la aplicación se ralentice o bloquee durante ese tiempo, por ello la ejecución en segundo plano libra parcialmente de notar este proceso.

Existe una diferencia entre las peticiones que se realizan en las APIs. Huawei permite realizar una única petición con todos los tipos de datos, es una petición algo más pesada pero solo se debe realizar una única vez. Google de manera similar a

Huawei, es capaz de realizar una petición con todos los tipos necesarios, pero se genera un problema grave. Las peticiones de la API de Google únicamente son capaces de devolver un tipo cada vez, el resto se pierde, por ello se realizan diferentes peticiones de menor tamaño en cascada, es decir, se almacenan y se genera la siguiente. Esto no supone demasiado incremento en tiempo, ya que se habla de uno o dos segundos de diferencia.

4.1.2 Preprocesamiento

Uno de los problemas que se han enfrentado durante la toma de datos, es la inestabilidad existente en los sensores. Por motivos físicos, las *smartbands* y por ello los sensores, no están siempre en contacto con el cuerpo, ya sea por ejemplo están cargando o porque se está tomando una ducha, pero durante ese tiempo se van a hacer registros de toma de datos, lo que implica la aparición del valor 0 (nulo).

Para enfrentarse a ello se han preprocesado los datos intentando rellenar este valor nulo con interpolaciones entre los dos más cercanos posibles, o en caso de un valor constante, como es la cantidad de pasos, replicando el último valor válido obtenido. Se puede ver un ejemplo en la Figura 38.

El preprocesamiento se prepara cuando se incluyen datos nuevos en BBDD. Al estar ordenados por fecha, el primero de ellos se marca con un identificador, así al lanzar el preprocesamiento se busca dicha marca para comenzar. Además de esta marca, se tiene un control de si la base de datos necesita ser preprocesada o no, dado que cada vez que se desea realizar se debe realizar una búsqueda

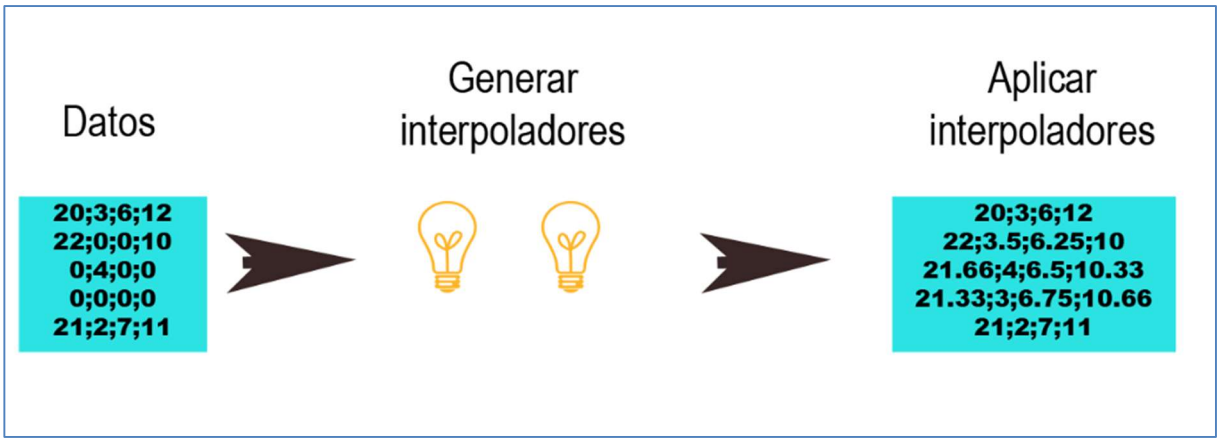

Figura 38.- Ejemplo de interpolación de datos. Fuente propia.

4.1.3 Búsqueda de bloques

¿Qué es una búsqueda de bloques?, este proceso viene dado a la hora de generar los archivos ARFF que van a ser introducidos dentro del algoritmo. Dado que no todos los datos de tiempo son útiles, se van a tener que buscar los bloques de tiempo donde realmente existen datos relevantes. Con esto se genera la búsqueda de los bloques de tiempo útiles, como se muestra en la Figura 39.

Por ello se han desarrollado dos funciones dedicadas a ello, una referente a la actividad y otra al sueño.

- Actividad: se encarga de buscar durante el día seleccionado todo aquel periodo de tiempo en el que los sensores han captado movimiento, es decir, pasos o desplazamiento. Devuelve una agrupación de todos estos periodos donde hay actividad y que pueden ser incluidos.
- Sueño: buscará los periodos donde los sensores han detectado sueño. No se limita a un día, ya que no tendría sentido si alguien se acuesta un día y se despierta al siguiente, por ello se estudia el bloque completo de sueño. Se busca el inicio y el final de estos periodos de sueño, ya que no se sabe cuándo el usuario que usaba las *smartband* ha dormido (ha generado datos de sueño) o cuánto ha durado el mismo.

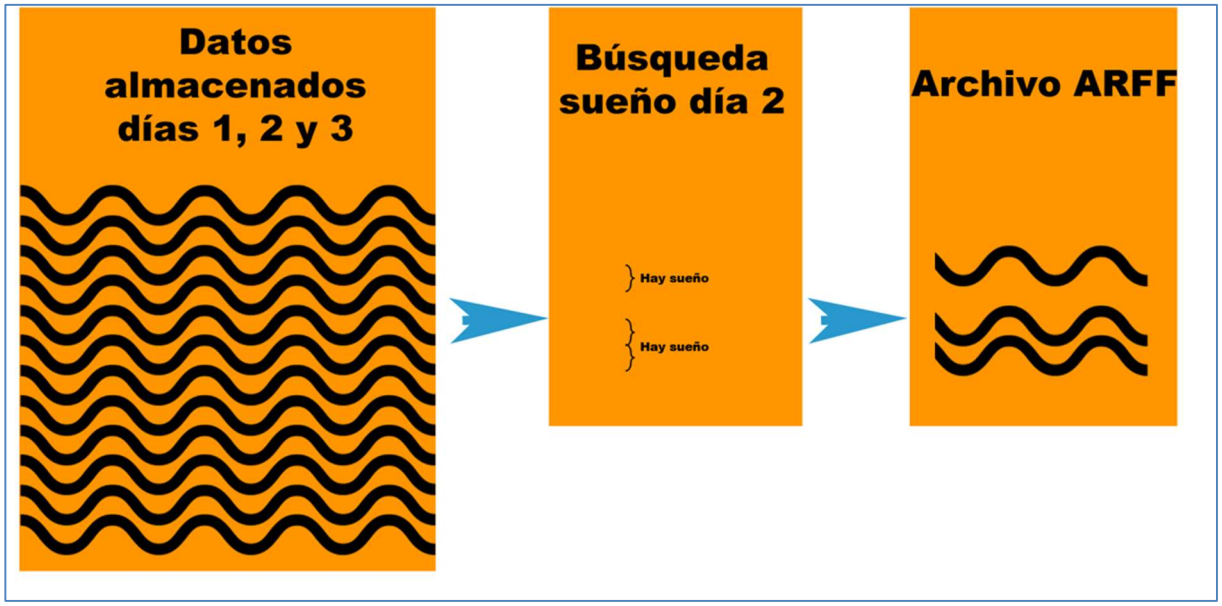

Figura 39.- Ejemplo búsqueda del sueño. Fuente propia.

4.2 Tolerancia a fallos

Una de las ideas que se ha planteado durante el proyecto es la existencia de diferentes errores y fallos que son conocidos, pero no siempre evitables. Por lo que se han desarrollado medidas para poder tolerar dichos errores y serán explicadas en este apartado. Asimismo, también se debe de ajustar el proyecto a los requisitos que piden las APIs, ya que no cumplirlos supone no obtener ningún dato.

4.2.1 Conexión a internet

Las APIs, en especial la API de acceso a Huawei, necesita tener acceso a internet, ya sea vía wifi o 4G, para hacer las peticiones de datos. En caso de que no se disponga de internet durante la petición o durante el establecimiento de la conexión con la API devolverá un error, ya que no ha podido completar alguna de estas acciones y por lo tanto no se completa la petición.

Incluso un fallo de conexión breve genera error, por lo tanto, si se produce dicho error no se va a considerar ese día válido, es decir, el día de la petición se añadirá de nuevo a la cola de días pendientes para ser pedidos, esta solución es la más cómoda. Debido a que no es posible asegurar constantemente si se dispone o no de conexión a Internet. Además, podemos tener un control de que en caso de que falle por un micro corte, tomaremos igualmente medidas.

4.2.2 Concurrencia

Dado que únicamente se dispone de una base de datos, tanto en memoria como en archivo de texto almacenado, es obligatorio controlar dicha concurrencia. Este problema se refleja principalmente a la hora de realizar las peticiones, debido a que las diversas APIs generan una petición asíncrona que se irá solventando tan pronto como sea posible. Por ello, al no tener un control directo sobre el orden en el que se genera el almacenamiento de datos, se controla la concurrencia con un bloqueo en las llamadas de almacenamiento en la BBDD.

El almacén de datos en memoria es compartido por todas las APIs, por lo que se pedirá permisos al mismo objeto y las peticiones se irán intercalando entre ellas. Para que, si se realizan peticiones durante el mismo instante, estas no se pierdan y machaquen, sino que se almacenen ambos.

4.3 Algoritmo SDATEP

4.3.1 Inclusión

Antes de comenzar la inclusión se va a comentar cuales han sido los entornos de programación que se van a usar, conocidos como IDE, siendo dos de ellos los que se usarán en el proyecto y los cuales se recomienda su uso en caso de querer continuar o ejecutar el proyecto.

Android Studio, IDE recomendado por los desarrolladores de Android. El cual permite compilar proyectos Java y Kotlin directamente en una apk que puede instalarse en un dispositivo Android. Para la aplicación se utiliza Java como lenguaje principal, se añaden además algunas características para Android, (como unión de pantallas, lectura de ficheros, interfaces, etc.) usando librerías que vienen por defecto.

Apache NetBeans, IDE utilizado para la edición y compilación del algoritmo SDATEP. El algoritmo SDATEP está totalmente implementado en Java, Apache NetBeans maneja automáticamente todas las dependencias y es un IDE disponible en cualquier sistema operativo, al igual que Java. Se utiliza este IDE para generar un archivo .jar (Java) que contendrá el algoritmo SDATEP, posteriormente este archivo .jar se incluye en Android Studio para su uso.

Esto lleva al primer intento fallido, pese a que el proyecto está escrito completamente en Java, al realizar la compilación del proyecto en una aplicación para dispositivos móviles, Android Studio no es capaz de usar todas las librerías necesarias. En concreto las librerías “com.yahoo.labs.samoa:samoa-instances:0.1.0” y “nz.ac.waikato.cms.moa:moa:2019.05.0” generaban un conflicto entre ellas dentro de Android Studio, pese a que no debería de darse tal situación.

Con ello el equipo se vio obligado a realizar otro tipo de inclusión del algoritmo SDATEP. Se tomó el algoritmo y se compiló de manera externa con todas las librerías, en un fichero java ejecutable. Este fichero se podría incluir en la aplicación a desarrollar directamente e importarlo como una librería externa.

Realizando pruebas se comprobó que era posible ejecutar el proyecto del algoritmo dentro de la aplicación Android, así que se hicieron las modificaciones para que se adaptara a cómo iba a funcionar el proyecto lerning y SDATEP.

4.3.2 Ficheros de datos

Para poder ejecutar adecuadamente el algoritmo SDATEP se va a generar ficheros en el formato ARFF, que debe de seguir unas pautas concretas mencionadas por Gordon Paynter, Len Tringg y Eibe Frank [23].

Se utilizan dos tipos de ficheros ARFF, dado que por la naturaleza de los experimentos los datos a estudiar serán diferentes.

```
1 @relation "ierningData"
2
3 @attribute heartRate numeric
4 @attribute steps numeric
5 @attribute acumulatedSteps numeric
6 @attribute distance numeric
7 @attribute acumulatedDistance numeric
8 @attribute class {groupA,groupB}
9
10 @data
```

Código 4.- Cabecera ARFF actividad.

El primero de ellos es referente al estudio de la actividad, así que se han tomado estos datos mostrados Código 4, que son aquellos que se toman como más relevantes a la hora de realizar un periodo de actividad.

```
1 @relation 'ierningData'
2
3 @attribute heartRate numeric
4 @attribute sleep numeric
5 @attribute class {groupA,groupB}
6
7 @data
```

Código 5.- Cabecera ARFF sueño.

El segundo de ellos, Código 5, está relacionado con el estudio del sueño. Pero no es necesario tantos datos, solo se quiere tener en cuenta los básicos, ya que, por la naturaleza del sueño, no debería de haber actividad.

4.3.3 Modificaciones

En este apartado se van a ver cuáles han sido las modificaciones que se han realizado para el algoritmo SDATEP, comentando de manera general se ha modificado el entrenamiento del algoritmo para que obtenga todos los datos de manera simultánea y no en *stream*, se ha añadido control de individuos vacíos y

simplificado el algoritmo para realizar solo estudio de un grupo A sobre un grupo B y no el opuesto.

4.3.3.1 Entrenamiento

El entrenamiento normal del algoritmo se realiza con diferentes *batchs* de un tamaño fijo, este tipo de estructura se replanteó para una gran cantidad de datos y de continua entrada de los mismos. Pero en este caso con SDATEP, tenemos una cantidad de información relativamente pequeña y junto a una potencia de cómputo y almacenamiento limitada, ya que este algoritmo se va a ejecutar dentro de un dispositivo móvil.

El algoritmo SDATEP funciona para problemas más restringidos, donde la cantidad de datos o muestras es escasa. Pese a ello se debe mantener su funcionamiento y obtener reglas expertas de una calidad considerable.

Con la modificación de los problemas a unos más pequeños, ha supuesto un cambio en el *batch* del algoritmo. Donde en lugar de usar diferentes *batchs* con un tamaño fijo, se ha tomado uno único de un tamaño variable, ya que depende del tamaño del experimento y de los datos que haya disponibles. Este funcionamiento se puede ver en las Figuras 40 y 41,

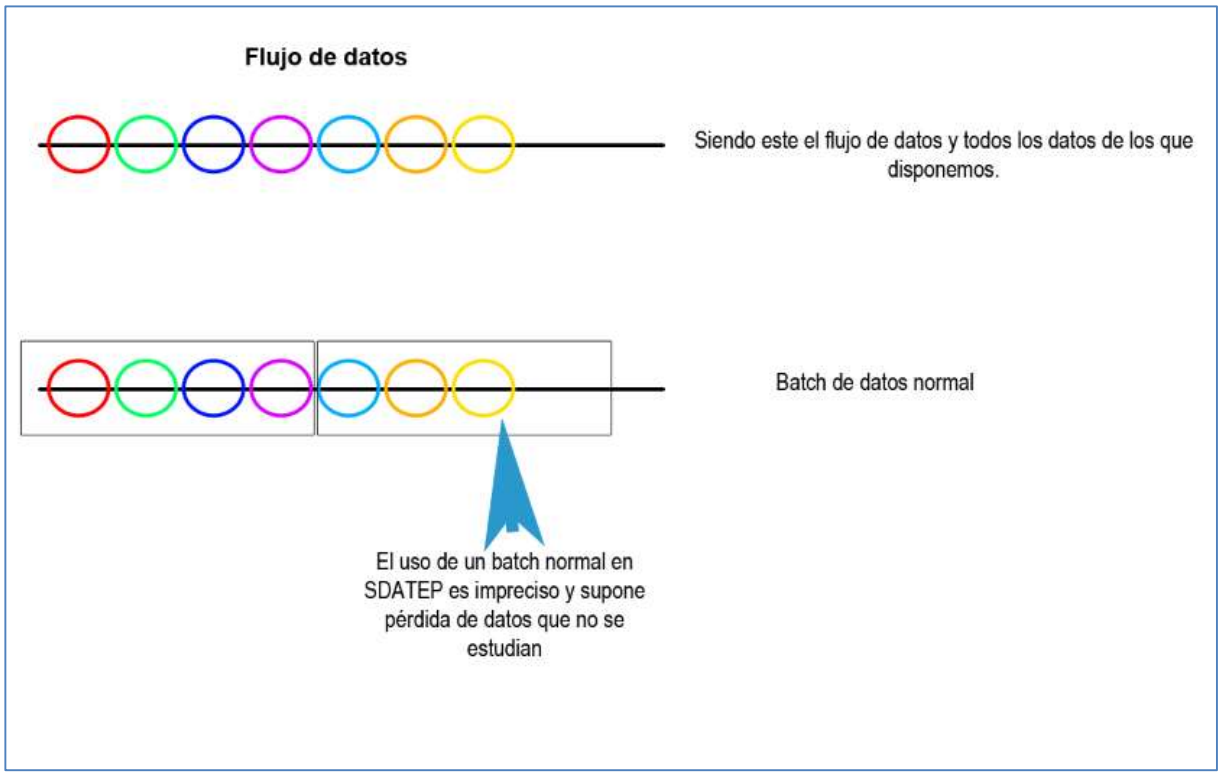

Figura 40.- Batch normal en SDATEP. Fuente propia.

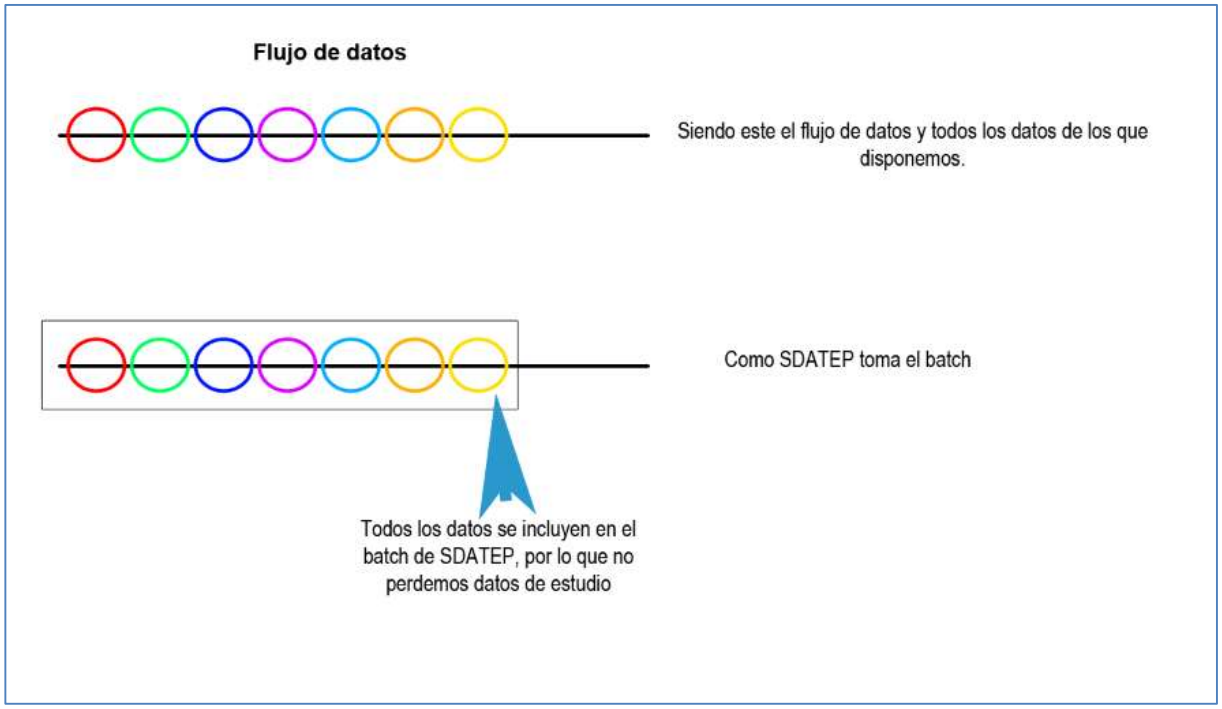

Figura 41.- Batch de SDATEP. Fuente propia.

4.3.3.2 Control de reglas vacías

El origen de este apartado aparece de uno de los problemas que surgieron durante el uso de SDATEP en el dispositivo móvil, realizando los estudios de prueba de las *smartbands*.

Tras haber realizado las modificaciones de los anteriores apartados dentro de este capítulo 4.3.3 modificaciones, los resultados de los test estaban siendo erróneos. El uso de datos reales, que pueden ser poco precisos, estaba resultando en la aparición de reglas expertas vacías, lo cual no debe existir bajo ningún concepto.

Para controlar este tipo de reglas vacías en SDATEP, se han modificado los operadores genéticos de manera que se tenga mucho más control de estas reglas vacías y se eliminen adecuadamente, ya que, al ser un tamaño de datos menor, no estaban siendo correctamente descartados. Las modificaciones que se han realizado se aplicaron al evaluador, el torneo de selección, la mutación y el ranking de dominación.

En el evaluador: originalmente si un individuo estaba vacío no se aplicaba ningún tipo de evaluación. Se ha añadido a eso una capa extra de tratamiento, por lo que en caso de que se encuentre vacío vamos a inicializarlo con las peores variables posibles. Este tratamiento se debe a que las reglas vacías no son deseadas y deben tener el peor desempeño posible para ser eliminadas intentando que no haya probabilidad de su selección posterior.

Respecto al torneo de selección: a la hora de la selección del ganador del torneo, se ha añadido un condicional antes del propio torneo, en el cual, en caso de que existiese un individuo vacío, directamente se tomará como ganador al otro individuo, sin importar la calidad de este. Con este paso se puede evitar que se seleccionen un individuo vacío siempre que no sea un torneo entre dos individuos vacíos.

En la mutación se hacían dos posibles tipos de mutaciones dependiendo de una probabilidad del cincuenta por ciento, una de estas mutaciones elimina uno de los genes, con ello se generaba un individuo considerado vacío. Para evitarlo, se ha eliminado esta probabilidad de eliminar uno de los genes y dejado solo la probabilidad de que durante la mutación se cambie el gen por uno diferente, sin modificar la probabilidad de mutación original.

Por último, en el ranking de dominación, de manera similar a como se había realizado en el torneo de selección, a la hora de establecer quién domina a quien, se comprueba si alguno de los dos individuos se encuentra vacío. En caso afirmativo, ese individuo será directamente dominado, dado que no queremos que ese individuo aparezca en el frente de pareto.

Capítulo 5: Experimentación

Durante este capítulo se va a realizar el procedimiento de obtención de conocimiento del proceso de ciencia de datos que se lleva a cabo en este TFG. Ahora que ya se dispone del algoritmo modificado y de la información, es posible hacer un estudio más exhaustivo de los resultados que se van a obtener y de cómo deben ser interpretados. Además de la preparación de dicho procedimiento, de la preparación del algoritmo y de su ejecución.

Antes de nada, cabe explicar el cómo va a ser el tipo estudio que se quiere realizar con el proyecto lerning, dado que será un estudio de datos en el cual no se toman todos estos sin previo conocimiento y se realiza un estudio, sino que se dividirá en varias fases.

Una primera fase que será la división del estudio en dos apartados, sueño y actividad, que, aunque comparten algún tipo de datos se realiza durante periodos de tiempo completamente diferentes. Después debe de hacerse referencia a cómo se van a realizar estos estudios, donde se va a establecer un grupo A, siendo este el grupo de estudio de un día seleccionado, y un grupo B, siendo este segundo grupo el resto de datos contra los que se va a estudiar el grupo A.

5.1 Preparación de la ejecución

Este apartado va a mostrar la preparación del algoritmo. Se hará mención de procedimientos que se han visto anteriormente.

El Código 6 muestra cómo es la preparación del algoritmo. En primer lugar, se cargan los datos de la interfaz, en este caso las etiquetas que se van a usar en el experimento y el número de días. Este número de días es conocido como *SlidingWindowSize,* pero dado que estará teniendo un uso diferente en SDATEP, es necesario que se encuentre a disposición del usuario.

```
cargarNúmeroEtiquetas()
cargarNúmeroDías()

IAtype <- cargarTipoDeTest()

baseDatos.preprocesar()

SI IAtype==0
    ENTONCES día<-prepararNuevoDíaActividad()
SINO
    ENTONCES día<-prepararNuevoDíaSueño()
FIN SI

SI IAtype==0
    ENTONCES baseDatos.ARFF_activity()
SINO
    ENTONCES baseDatos.ARFF_sueño()
FIN SI

actualizarParámetrosAlgoritmo()

ejecutarAlgoritmo()
escribirResultados()
```

Código 6.- Pseudocódigo Ejecución IA.

El preprocesamiento de los datos es tal como se comenta en el apartado 4.1.2, pero la preparación de los días implica un tiempo algo más largo, donde primero hay una creación de la fecha, en la que se establece un punto central para la búsqueda. Después la preparación del archivo ARFF, este es el formato de los archivos de entrada que lee el algoritmo SDATEP, por ello se debe preparar el formato de la base de datos a este formato ARFF. Además, dependiendo del tipo de experimento que se desee realizar debe tener formatos diferentes.

Por último, se actualiza los parámetros del algoritmo SDATEP, un archivo en formato texto, con el nombre de "`param.txt`", este fichero se introduce junto al algoritmo donde toma los siguientes datos:

- Algoritmo usado (`algorithm`).
- Nombre y localización de los ficheros de entrada (`inputData`).
- Nombre y localización de los ficheros de salida (`outputData`).
- Número de instancias en un *batch,* tamaño del mismo (`period`).
- Semilla del algoritmo (`seed`).
- Formato de representación de reglas (`RulesRepresentation`).
- Número de etiquetas (`nLabels`).

- Número de generaciones máximas (`nGen`).
- Longitud de la población (`popLength`).
- Probabilidad de cruce (`crossProb`).
- Probabilidad de mutación (`mutProb`).
- Tipo de evaluador utilizado (`Evaluator`).
- Tamaño de la ventana (`SlidingWindowSize`).
- Objetivos de calidad (`Obj1, Obj2, Obj3`).
- Medida de diversidad (`diversity`).
- Tipo de filtro (`filter`).

5.2. Parámetros de ejecución

A la hora de querer usar el algoritmo y ejecutarlo, se va a necesitar especificar una serie de parámetros.

Primero aquellos que pueden ser modificados por el usuario o se van a modificar automáticamente ya que, debido a la naturaleza de las modificaciones en el algoritmo, cambian con frecuencia.

- **`nLabels`**: Es el número de etiquetas que se utilizarán en el algoritmo.
- **`Días`**: Es el número de días contra el que se va a estudiar el día actual, estos se introducirán dentro del algoritmo con diferentes marcas.
- **`Period`**: Aunque originalmente era el número de instancias que se iban a utilizar por ciclo del algoritmo, ahora marca el total de datos que tenemos, este hecho fue explicado en el apartado 4.3.3.1. Este valor va a marcar el tamaño del batch que SDATEP usará.
- **`TOTAL_DAYS_SAVED`**: Indica el máximo de días que pueden estar almacenados dentro de memoria. Se utiliza para las peticiones y control sobre qué días estarán dentro de la base de datos.

Segundo se verán también el resto de los parámetros relacionados con SDATEP, en concreto los que no se van a modificar durante las ejecuciones y que tendrán el valor que indica la Tabla 16.

Para la realización de los experimentos se va a tener diferentes variaciones, en concreto sobre el tipo de experimento, días y las etiquetas, siguiendo el siguiente esquema de la cantidad de experimentos que haremos.

		Valor
Parámetro	outputData	taxis_tra_qua.txt taxis_tst_qua.txt taxis_rules.txt
	RulesRepresentation	dnf
	nGen	70
	popLength	50
	crossProb	0,8
	mutProb	0,1
	Evaluator	byDiversity
	diversity	WRAccNorm
	Obj1	WRAccNorm
	Obj2	Confidence
	Obj3	TPR
	filter	TokenCompetition

Tabla 16.- Parámetros SDATEP.

		Etiquetas		
		3	5	7
Días	1	Sueño/Actividad	Sueño/Actividad	Sueño/Actividad
	7	Sueño/Actividad	Sueño/Actividad	Sueño/Actividad
	14	Sueño/Actividad	Sueño/Actividad	Sueño/Actividad
	21	Sueño/Actividad	Sueño/Actividad	Sueño/Actividad

Tabla 17.- Total de experimentos.

5.3 Resultados

Todos los algoritmos y medidas aquí mencionadas se encuentran definidas en las publicaciones [13] [22] [24], de las cuales serán usadas las siguientes:

- **NumRules:** Indica el número de reglas que se han extraído del problema. Este es un resultado que se desea minimizar.
- **NumVars:** Son el número de variables que forman parte de las reglas, indica la complejidad de las mismas. Este es un resultado que se desea minimizar.
- **CONF:(*Confidence*)** En este resultado es el total de aciertos respecto a los ejemplos que cumplen el antecedente de la misma. Este es un resultado a maximizar.
- **FPR: (*False positive rate*)** Proporción de falsos positivos, este resultado marca el porcentaje de datos que han sido mal cubiertos respecto al total de ejemplos que no pertenecen a la regla. Este es un parámetro que se busca minimizar.
- **GR_PCT: (*Growth percentage*)** El valor que indica el porcentaje de los ejemplos cubiertos en el grupo A frente a los del grupo B. Esta es una medida que se busca maximizar.
- **TPR: (*True positive rate*)** Este valor establece el porcentaje de ejemplo que se han cubierto correctamente en comparación con el total de positivos que ha habido, esta es una medida que se busca maximizar.
- **SuppDiff: (*Support difference*)** Muestra este valor la diferencia que existe entre TPR y FPR, esta es una medida que se busca maximizar.
- **WRAcc_Norm: (*Weighted relative accuracy normalized*)** Este valor representa el equilibrio que existe entre la cobertura del patrón y la precisión del mismo. Pondera la confianza y el TPR de manera normalizada para que se pueda comparar. Esta es una medida que se busca maximizar.
- **ExecTime_ms: (*Execution time*)** Tiempo de ejecución, establece cuánto ha tardado el algoritmo en ejecutarse y obtener las reglas, se muestra con milisegundos como unidad de medida, haciendo de este valor uno que se busca minimizar.

- **Memory_mb**: Memoria utilizada durante el tiempo de procesamiento, se encuentra expresada en megabytes, es una medida que se busca minimizar.

Durante las siguientes tablas de este apartado se van a utilizar abreviaturas de las medidas indicadas anteriormente, quedando en orden: Et(etiquetas), NR(NumRules), NV(NumVars), CONF, FPR, GR(GR_PCT), SD(SuppDiff), WRAcc(WRAcc_Norm), Exec(ExecTime_ms) y Mem(Memory_mb).

Establecidos cuales van a ser los diferentes experimentos a realizar, Tabla 17, se va a mostrar los diferentes resultados de dichos experimentos en forma de tabla.

5.3.1 Resultados de los experimentos con datos de sueño

Este apartado refleja en la Tabla 18 los resultados que se han obtenido durante las diferentes ejecuciones del algoritmo. En este caso, la modificación se realiza con el número de días, que incrementa el número de datos, y el número de etiquetas que usa el algoritmo SDATEP.

Como se muestra en la Tabla 17, se realizan doce experimentos, establecidos en bloques de etiquetas similares y diferentes días. Se muestran todos juntos con la intención de favorecer la lectura de los mismos y que se pueda tener un visionado general rápido.

En siguientes apartados se explicará el valor de los resultados y qué significan dichos valores, teniendo siempre esta Tabla 18 como referencia principal.

Et	Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
3	1	3	1,33	0,76	0,38	1,00	0,30	0,68	0,65	1810	64
3	7	2	1,00	0,15	0,75	1,00	0,11	0,87	0,56	1894	83
3	14	2	1,50	0,07	1,00	1,00	0,00	1,00	0,50	2085	71
3	21	3	1,33	0,05	1,00	1,00	0,00	1,00	0,50	2225	59
5	1	3	1,33	0,76	0,38	1,00	0,30	0,07	0,65	2111	66
5	7	2	1,50	0,13	0,95	1,00	0,04	0,99	0,82	1928	27
5	14	1	1,00	0,07	0,99	1,00	0,01	1,00	0,51	2206	29
5	21	2	1,50	0,05	0,99	1,00	0,01	1,00	0,50	2360	32
7	1	3	1,67	0,83	0,33	0,67	0,01	0,35	0,51	2030	39
7	7	3	1,33	0,16	0,54	1,00	0,14	0,68	0,57	2115	20
7	14	1	1,00	0,07	0,89	1,00	0,10	0,99	0,55	2363	42
7	21	1	1,00	0,06	0,69	1,00	0,20	0,89	0,60	2239	32

Tabla 18.- Resultados experimentos de sueño.

5.3.2 Resultados de los experimentos con datos de actividad

De manera análoga que en el apartado 5.3.1, se realiza un muestreo de los resultados que se han obtenido del algoritmo SDATEP tras realizar las ejecuciones y experimentos con los datos de actividad.

Se tomará de nuevo el mismo orden, siendo grupos de etiquetas con diferentes números de días, seguidos de los valores que se han obtenido en cada uno de los valores. Queda así la Tabla 19 con los resultados agrupados y ordenados para posteriormente su explicación y análisis.

Et	Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
3	1	5	3,40	0,45	0,45	1,00	0,55	1,00	0,77	2647	51
3	7	1	4,00	0,11	0,73	1,00	0,27	1,00	0,64	2239	35
3	14	2	3,50	0,07	0,76	1,00	0,23	0,99	0,62	1970	37
3	21	2	3,50	0,06	0,74	1,00	0,25	0,99	0,62	2401	53
5	1	1	2,00	1,00	0,00	1,00	0,02	0,02	0,51	2176	37
5	7	2	5,00	0,21	0,05	1,00	0,09	0,13	0,54	2651	31
5	14	5	2,80	0,27	0,48	1,00	0,29	0,77	0,65	2477	36
5	21	1	5,00	0,08	0,59	1,00	0,41	1,00	0,71	2780	41
7	1	5	2,40	0,55	0,28	1,00	0,59	0,72	0,80	2922	51
7	7	6	3,00	0,15	0,50	1,00	0,46	0,96	0,73	3217	28
7	14	4	5,00	0,16	0,29	1,00	0,36	0,65	0,68	3234	26
7	21	1	4,00	0,19	0,04	1,00	0,16	0,20	0,58	2853	32

Tabla 19.- Resultados experimentos de actividad.

5.4 Análisis de los resultados

Durante este apartado se va a explicar qué se puede extraer de estos datos y que aprender de ellos, al igual que la calidad de los mismos o si se podría escalar estos experimentos. Se estudiará la calidad de ambos tipos de experimentos, con sueño y con actividad.

5.4.1 Estudio de calidad

Antes de comenzar con el apartado es interesante hablar un poco del usuario del que se han tomado los datos. En este caso, David Padilla Montero, quien ha llevado los dispositivos *smartbands* durante toda la creación del proyecto.

Una media de sueño de siete horas, salvo algunos días puntuales, respecto al tiempo de actividad, se debe comentar que es algo escueto dado que vive una vida sedentaria y el ejercicio que realiza es complicado de medir, ya que se realiza en el sitio, como las pesas.

Una vez comentado esto, se va a realizar el estudio de calidad para cada una de las partes.

5.4.1.1 Estudio de sueño

En la Tabla 18, se puede ver cómo se obtienen unos buenos resultados en cuanto a número de reglas y variables en mayor cantidad de días, incluso siendo un experimento con más cantidad de datos, solo se obtiene una regla y una única variable.

La Tabla 18 es una agrupación de todos los resultados en orden, es decir, resultados de tres etiquetas, luego de cinco y luego de siete etiquetas, para tenerlos todos agrupados y de fácil visualización.

Durante el estudio del sueño se ve como el número de reglas varía entre una, siendo este el mínimo y el número más deseado, hasta tres, que no llega a ser una cantidad excesiva, pero dado que la cantidad de datos que se están manejando no es tampoco de un gran tamaño. Pese a ello, el objetivo del algoritmo es la búsqueda del mínimo número de reglas, por ello, al tener una cantidad baja se obtienen resultados menos complejos.

Se hablará ahora de otros elementos interesantes a tener en cuenta, en la Tabla 16 se puede ver como los objetivos en el algoritmo son WRAcc_Norm, Confianza y TPR, por lo que estos valores serán los que más se desean estudiar.

El WRAcc_Norm es bastante mejorable. Pero se puede ver cómo evoluciona dicho valor, debido a que en pocos días (pocos datos) funcionan mejor una cantidad menor de etiquetas. Mientras en más cantidad de días, a partir de catorce, funciona menor la máxima cantidad de etiquetas y permite obtener unos mejores valores de este resultado.

Ahora para el estudio del último, TPR se puede ver en los experimentos de menor número de días este número era bajo. Incrementando el número de días y por ello de datos, incrementa, indicando que se mejora la precisión de las reglas que se han obtenido. En los experimentos de catorce o más días, se muestra un TPR de cercano al 1, lo cual indica que la regla cubre casi el mayor porcentaje de los datos que realmente son positivos.

5.4.1.2 Estudio de actividad

Se volverá a hablar de las tres principales variables que son de interés, WRAcc_Norm, Confianza y TPR, ya que con ayuda de la Tabla 16, será fácil poder estudiarlas.

Por otro lado, la Tabla 19 contiene todos los resultados en orden, es decir, resultados de tres etiquetas, luego de cinco y luego de siete etiquetas, para tenerlos todos agrupados y de fácil visualización

Durante este apartado se va a comentar también los resultados que se han obtenido de los experimentos con los datos de actividad. Es de interés empezar con la cantidad de reglas y de variables, que, a diferencia del sueño, son mucho más diversas. Esto provoca que puedan llegar a haber seis reglas o cinco variables, lo cual incrementa la complejidad. Al igual que en estudio del sueño, se está estudiando un *batch* cada vez, por lo que, al incrementar el número de datos, es deseado que estas reglas no incrementen. Por ejemplo, en los experimentos con veintiún días, el número de reglas es de uno y el de variables está entre 3.5 y 5, tal y como en el experimento anterior. Es el objetivo del algoritmo obtener la mínima cantidad de reglas con la mayor cobertura posible.

Se procede a estudiar WRAcc_Norm, ya que esta medida es la ponderación entre la confianza y el TPR, es decir, del resto de los valores que queremos de estudio.

Se puede ver que la medida de WRAcc_Norm en el estudio de actividad tiene un mejor resultado que los de sueño. Puede verse debido a que se tienen una mayor cantidad de datos o a que existe una variación mayor entre ellos, debido a que los del sueño son más homogéneos. Pero en WRAcc_Norm es posible ver diversos casos que tienen resultados más bajos, el experimento de cinco etiquetas y un día, el experimento de cinco etiquetas y tres días y el último de la Tabla 19. En todos estos casos el TPR baja de gran manera, lo que afecta a la ponderación, sobre todo en el primero mencionado donde FPR es de 0.

Anteriormente en el estudio de sueño, se incrementa el TPR conforme se incrementa el número de días, debido a que los datos son muy homogéneos. Esto favorece la creación de reglas sencillas que abarquen con precisión mayor cantidad posible de los mismos. Pero en caso de actividad no son tan homogéneos, el tiempo

de actividad entre diferentes días no es el mismo, al igual que la intensidad de la actividad.

En conclusión, el estudio da una idea de lo pequeña que es posible hacer la ventana de días y que se obtengan unas reglas útiles para el estudio. El proyecto plantea obtener un estudio sobre los datos de salud con algoritmo SDATEP en un espacio de poca cantidad de muestras. Siendo la tendencia normal es un incremento de la cantidad de estas para una generación mayor de conocimiento.

5.4.1.3 Tiempo y tamaño

Antes de comenzar la explicación de estas partes debemos tener en cuenta que se han realizado en un dispositivo móvil, el cual, no tenemos control de la cantidad de memoria, CPU y tiempo que dedica.

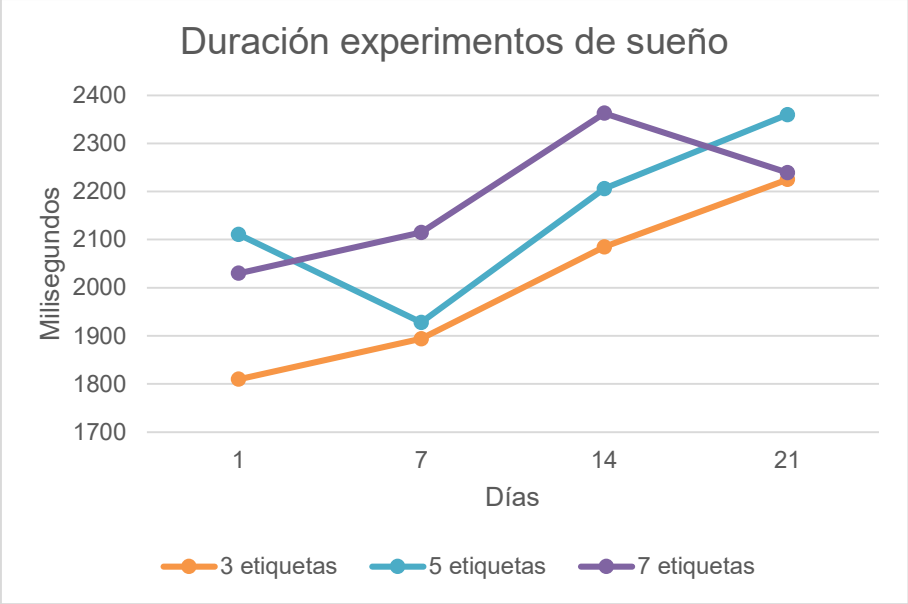

Figura 42.- Tiempos sueño.

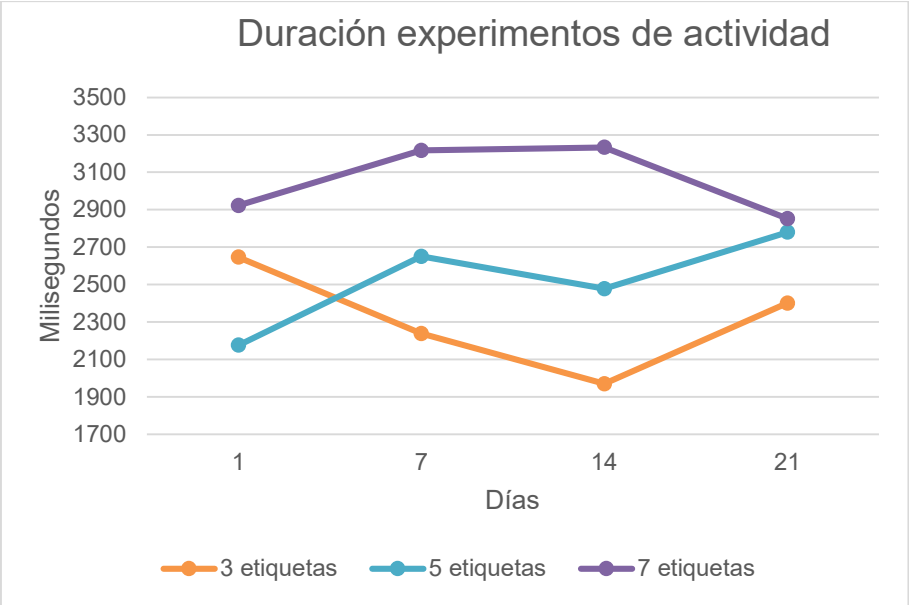

Figura 43.- Tiempos actividad.

Se va a observar ahora los tiempos de los estudios de sueño y actividad, Figuras 42 y 43. Como era de esperar, la duración de los experimentos de actividad es mayor en general, puesto que la diferencia entre los datos y la mayor cantidad hacen que se dedique más tiempo.

Por un lado, el sueño sigue un orden lógico, a mayor cantidad de datos se dedica una mayor cantidad de tiempo. En cambio, en el estudio de actividad no es así, la cantidad de tiempo empleada no sigue un orden lógico. Por ejemplo, con cinco etiquetas se obtiene un incremento de duración, pero con tres, el estudio de un solo día requería más tiempo que el de ningún otro.

El mayor tiempo que se ha obtenido es superior a los tres segundos mientras que el menor de los tiempos ronda los dos segundos, 1810 milisegundos en el primer experimento de sueño. No son tiempos excesivos, pero se debe tener en cuenta que el usuario del teléfono notará una ligera pausa en la aplicación, por ello debemos controlar estos tiempos. La aplicación debe funcionar en primer plano, debido a restricciones en las APIs, por lo que no es posible ocultar plenamente este tiempo, aunque si mitigarlo.

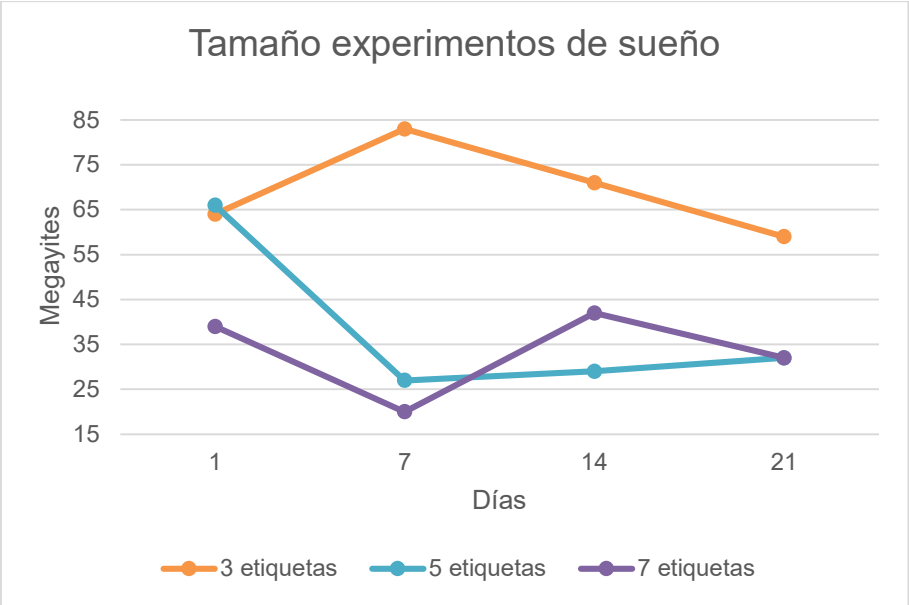

Figura 44.- Tamaños sueño.

Una de las cosas que se pueden ver respecto al tamaño en memoria, es que el uso de tres etiquetas supone un mayor consumo de memoria, Figuras 44 y 45. Además, de que el incremento de etiquetas supone un decremento en el tamaño de memoria utilizado para el experimento. Esto es un valor importante a tener en cuenta, ya que la aplicación desarrollada trabaja con dispositivos móviles y con ello una memoria de tamaño limitado.

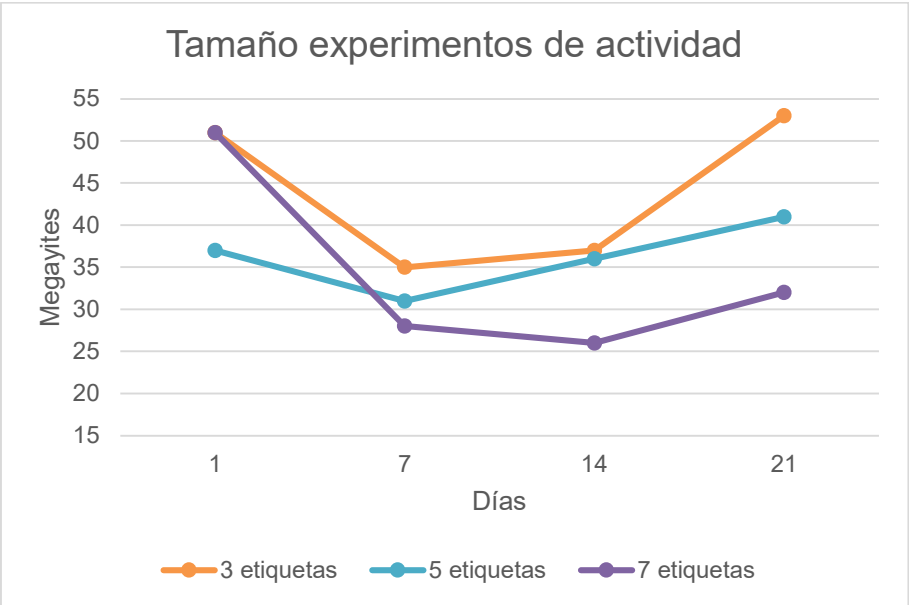

Figura 45.- Tamaños actividad.

El tamaño de los datos usados para el experimento supone entre 2.5 y 3 megabytes, por lo que en comparación al experimento es muy menor en consumo de memoria. Con ello en mente se debe controlar el tamaño de los experimentos, aunque gracias al incremento de almacenamiento en los dispositivos móviles aún existe mucho margen con el que poder trabajar sin suponer un peso real en el dispositivo.

5.4.2 Comparación de mejores resultados

Los resultados obtenidos de sueño y de actividad pueden ser comparados entre ellos, pudiendo ver cuál ha tenido un mejor desempeño usando el algoritmo SDATEP. Se debe tener en cuenta que en sueño se tiene una cantidad inferior de información por día, es decir, hay menos instancias de datos.

Para definir cuál es el mejor resultado se va a tener en cuenta: el número de reglas más bajo posible, mayor cantidad de datos, el WRAcc_Norm más alto, el TPR más alto y el mayor SuppDiff. Para responder a la pregunta de qué sería un mejor resultado, cabe señalar que la regla obtenida debe tener una mayor cobertura y poseer la menor cantidad de fallos.

Primero, se va a ver cuál es el mejor resultado obtenido de sueño, usando la Tabla 20, donde se ha tomado de cada cantidad de etiquetas una (o dos en caso de tres etiquetas) fila de la tabla.

Mejor sueño 3 etiquetas

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
7	2	1,00	0,15	0,75	1,00	0,11	0,87	0,56	1894	83
14	2	1,50	0,07	1,00	1,00	0,00	1,00	0,50	2085	71

Mejor sueño 5 etiquetas

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
14	1	1,00	0,07	0,99	1,00	0,01	1,00	0,51	2206	29

Mejor sueño 7 etiquetas

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
21	1	1,00	0,06	0,69	1,00	0,20	0,89	0,60	2239	32

Tabla 20.- Mejores resultados para el sueño.

Siguiendo los criterios establecidos anteriormente, se toma como mejor resultado el referente a siete etiquetas y veintiún días, debido a que teniendo la mayor cantidad de información, mantiene buenos valores de TPR y WRAcc_Norm.

Mejor actividad 3 etiquetas

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
7	1	4,00	0,11	0,73	1,00	0,27	1,00	0,64	2239	35

Mejor actividad 5 etiquetas

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
21	1	5,00	0,08	0,59	1,00	0,41	1,00	0,71	2780	41

Mejor actividad 7 etiquetas

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
21	1	4,00	0,19	0,04	1,00	0,16	0,20	0,58	2853	32

Tabla 21.- Mejores resultados para la actividad.

Segundo, se va a realizar el mismo procedimiento con los valores de actividad, aplicando los mismos criterios, y con ello, obteniendo el mismo tipo de resultado, encontrar el mejor resultado dentro de la actividad. Se tomará la Tabla 21 como referencia, donde aparece una fila de la tabla para cada valor de etiquetas. En este caso se toma como mejor resultado, la fila de veintiún días con 5 etiquetas dado el alto WRAcc_Norm.

Mejor sueño

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
21	1	1,00	0,06	0,69	1,00	0,20	0,89	0,60	2239	32

Mejor Actividad

Días	NR	NV	CONF	FPR	GR	SD	TPR	WRAcc	Exec	Mem
21	1	5,00	0,08	0,59	1,00	0,41	1,00	0,71	2780	41

Tabla 22.- Mejores resultados.

La Tabla 22 es una comparación entre los mejores resultados que se han obtenido de sueño y de actividad, ambos de ellos en veintiún días pero usando sueño con siete etiquetas y actividad con cinco.

En términos de reglas tanto sueño como actividad disponen únicamente de una, es decir, el mínimo posible, teniendo así ambas la misma complejidad a la hora de

estudio. Por otro lado, sabiendo que ambas disponen de la misma cantidad de datos, el WRAcc_Norm de la actividad es mayor, mostrando así que la cobertura de esta regla es bastante amplia, además su TPR siendo de 1 para actividad.

No es posible comparar el número de variables, debido a que la actividad posee mayor cantidad de tipos de datos es normal que el resultado de NumVars(número de variables) sea mayor. Por el mismo motivo recién comentado, el tiempo de ejecución es mayor, más información requiere más tiempo.

En conclusión, la regla con mejor cobertura y posee un TPR mayor (al igual que un FPR menor en este caso), es la regla obtenida de actividad. Está es la mejor regla que se ha obtenido del algoritmo SDATEP en los datos usando durante los experimentos.

5.4.3 Reglas obtenidas

Como continuación del análisis es importante ver cómo se muestran las reglas resultado del algoritmo. Las reglas son de tipo condicional, por lo que se dispone de una condición que divide las posibles situaciones entre las etiquetas posibles que se disponen. Aparecen los resultados siempre, `Consecuent: groupA`, esto es referente al grupo A que se menciona en el apartado 5, al inicio del capítulo. El grupo A es el día que se desea estudiar y el grupo B todos los demás días contra los que se va a estudiar. La aparición de `groupA` es una indicación de que estas reglas son referentes al día de estudio seleccionado.

Para esta explicación se ha tomado un ejemplo de sueño con siete días. Además, se tomarán cinco etiquetas, mientras que de actividad ha sido de catorce días y se volverá a tomar cinco etiquetas.

Al ser cinco etiquetas se dividen en grupos *Label* 0, 1, 2, 3 y 4, todos ellos determinando el nivel en el que se encuentran, siendo 0 el nivel más bajo y 4 el nivel más alto, dentro de las divisiones. Pudiendo considerar 0-muy bajo,1-bajo, 2-medio, 3-alto y 4-muy alto, por lo que se va a ver cómo se leen estas reglas, empezando por el Código 7.

```
1 Rule 1 (ID: 1912490658)
2    Variable heartRate = Medio (41.536064 52.49344 63.450813)
3    Variable sleep = Muy Bajo (-3.4028235E38 1.0 1.5)
4                     Medio (1.5 2.0 2.5)
5                     Muy Alto (2.5 3.0 3.4028235E38)
6    Consecuent: groupA
```

Código 7.- Ejemplo reglas sueño.

La regla 1, contiene dos variables, una para las pulsaciones del corazón y otra para el sueño. Las variables están asociadas a los diferentes tipos de datos que se introducen en el algoritmo, en concreto se ha obtenido un *Label* para las pulsaciones y tres para el sueño.

Los valores de sueño van de 1 a 5 como se puede ver en Código 1, comenzando hablando de las pulsaciones del corazón. Sale que está en *Label* 2, lo que sería una medida de nivel medio, y aparecen tres valores 41.5 como valor mínimo, 63.45 como valor máximo con un valor medio de 52.49. Este es el modo en el que debe leerse y ser hará de manera equivalente en el resto de variables, siendo divididas en las diferentes *Labels* si dispone de más de una, cabe aclarar simplemente el valor (+/-)3.40028235E38, siendo este valor el referente a más infinito o menos infinito dependiendo de su símbolo de signo.

Este tipo de representaciones se dan por que se usa una representación en DFN con lo que para leerlo quedaría que cada variable se puede encontrar en los estados:

- Pulsaciones: nivel medio
- Sueño: nivel bajo, medio o alto

También se va a explicar las implicaciones del Código 7. Sabemos que el sueño puede tomar valores entre 1 y 5, en este caso los valores medios son 1, 2 y 3 para los diferentes niveles de sueño. Siendo como aparece en código 1, 1- sueño ligero, 2-fase REM y 3-Sueño profundo, pero solo hay un valor para las pulsaciones. Esto indica que las pulsaciones no han variado mucho durante el sueño y que se encuentran en un nivel medio para cualquiera de los 3 momentos de sueño conocidos.

```
1 Rule 2 (ID: -1425126442)
2    Variable heartRate = Muy Bajo(-3.4028235E38 39.986744 72.13486)
3                         Bajo (39.986744 72.13486 104.282974)
4                         Medio (72.13486 104.282974 136.43109)
5    Variable steps = Muy Bajo (-3.4028235E38 1.0 33.25)
6                     Medio (33.25 65.5 97.75)
7    Variable acumulatedSteps = Bajo (2.0 1956.75 3911.5)
8                               Alto (3911.5 5866.25 7821.0)
9                               Muy Alto (5866.25 7821.0 3.4028235E38)
10   Variable distance = Muy Alto (67.5 90.0 3.4028235E38)
11   Variable acumulatedDistance = Muy Bajo (-3.4028235E38 2.0 1251.25)
12   Consecuent: groupA
```

Código 8.- Ejemplo reglas actividad.

Para el Código 8 se puede ver un ejemplo de actividad, aparecen cinco variables y cada una de ellas con diferentes *Labels*, aunque se mantiene el mismo método de lectura que el anteriormente mencionado. Por lo que la representación en estados se puede ver cómo quedaría:

- Pulsaciones: nivel Muy Bajo, Bajo o Medio
- Pasos: nivel Muy Bajo o Medio
- Pasos acumulados: nivel Bajo, Alto o Muy Alto
- Distancia: nivel Muy Alto
- Distancia acumulada: nivel Muy Bajo

Algunas variables, por ejemplo, las distancias, disponen únicamente de un estado, mientras que pulsaciones o pasos todas ellas disponen de diferentes estados en los que se pueden encontrar. En este experimento el WRAcc_Norm es de 0,65 y con un TPR de 0,77, lo cual es una buena cobertura de los datos que se han estudiado.

Igual que en Código 7 se explica también las implicaciones del Código 8, ya que, en el caso de esta regla hay muchas posibles variables. Por ejemplo, si miramos las pulsaciones, se observa que el nivel máximo es medio con un valor de 104 de media, es decir, la actividad conocida no tiene unas pulsaciones altas. Esta actividad la podemos ver en pasos y distancia. La distancia solo tiene un único valor alto, mientras que los pasos pueden ser medio o muy bajos, esto se debe al ejercicio en casa. Los pasos se miden, pero la distancia no se mide del todo bien, por ello, cuando se logra medir distancia (fuera de casa), esta llega a niveles muy altos. Comparando la distancia acumulada es muy baja, debido a que sí, un día se hace ejercicio fuera de casa tendrá mucha distancia, pero dentro de casa no, por lo que, el día de estudio, fue día sin ejercicio fuera.

5.4.4 Estudio de escalabilidad

El algoritmo en sí SDATEP, es un estudio con reducción de cantidad de muestras, por lo que existe la posibilidad de poder volver a un estudio de *data streaming* si se da en caso, es decir, que la cantidad de datos que se tomen y se almacenen para estudio sea suficiente.

En el caso del algoritmo SDATEP la escalabilidad de memoria no debería de suponer un problema, los dispositivos móviles actuales disponen de varias gigas de memoria y como se ha comprobado en el apartado 5.5.1.3, la memoria en uso no supera los 100 mb más la memoria de la aplicación, 2.7 mb, si se redondea al alza serían 103 mb en total de uso de la aplicación. Solo los 3 mb de memoria se quedan de forma permanente.

El principal problema a tener en cuenta a la hora de la escalabilidad es el tiempo de ejecución. Como se muestra en las Figuras 53 y 54 los tiempos de ejecución suponen entre 1.8 y 3 segundos, pero existe un tiempo máximo que se puede tener a un usuario esperando.

Para el conocimiento del tiempo de retraso, haciendo referencia al estudio de Nielsen, “para el usuario un tiempo entre 0.1 y 1 segundos pasa desapercibido si espera respuesta. El límite máximo de tiempo es de 10 segundos, siendo este momento cuando el usuario pierde la atención” [25].

Si bien existe la posibilidad de realizar la ejecución del algoritmo en segundo plano, esto supone una carga alta en el dispositivo que puede conllevar a incrementar mucho el tiempo de ejecución y de suponer una ralentización global del dispositivo.

En términos globales, la escalabilidad es posible, incluso la posibilidad de volver a un *data streaming*, sin las modificaciones de reducción para Android, si el número de datos es suficiente. Debe de tenerse en cuenta en gran medida el tiempo, ya que la CPU de los dispositivos móviles tiene menor potencia que una de ordenador.

Capítulo 6: Conclusiones

6.1 Conclusión

Este proyecto ha sido uno de los primeros en introducir un estudio de obtención de patrones de aprendizaje en un dispositivo móvil y la extracción de datos se realice a través de los sensores de diferentes *smartbands* conectadas a un usuario. Además, la unión de diversas familias de *smartbands* es algo que pocas veces suele ocurrir por el poco trabajo entre ellas, con ello en lerning se ha logrado una aplicación más global y con una amplitud de estudio mayor que algunas aplicaciones de salud de las mismas familias.

La parte relacionada con las *smartbands* se ha realizado en un entorno sencillo. Se utilizan las APIs que las propias empresas o familias de dispositivos dejan a disposición de la aplicación y el objetivo ha sido centralizar y unificar todos los datos de manera homogénea, por lo que el estudio de estos se puede hacer independientemente de su procedencia. Estos se almacenan de manera local, debido a ello no se tiene acceso a ellos desde fuera de la propia aplicación, protegiendo así la integridad y la confidencialidad de los datos. Asimismo, se almacenan codificados numéricamente, evitando poder tomar o trabajar con información de carácter personal.

La parte relacionada con el algoritmo se centra en una cantidad de datos mucho menor, que no pueden llegar a algunos umbrales de cantidad preestablecidos. Por ello la modificación propuesta, SDATEP, supone una simplificación del algoritmo, que genera unas reglas de una calidad aceptable mientras que sus datos de estudio pueden ser muy pequeños. No se utiliza una cantidad de memoria alta, ni de tiempo de ejecución, por lo que aún es posible realizar estudios de SDATEP con mayor cantidad de muestras.

Los datos obtenidos se dividen en dos partes, los de sueño son muy homogéneos en todos los días, mientras que los de actividad son muy desbalanceados y diferentes, haciendo así que ambos hayan resultado interesantes como estudio.

6.2 Trabajo futuro

Existen dos líneas de trabajo que pueden ser de interés para el trabajo futuro, ya que, por complejidad o disponibilidad, no son posibles realizarlas ahora.

La primera sería referente al número de muestras que se han adquirido. Se puede ver la posibilidad de incrementar el número de *smartbands* que se usan para las peticiones e incrementar las bases de datos usadas. Esto permite que en el mismo transcurso de tiempo (días) puedan existir mayor cantidad de datos, debido a las pequeñas discrepancias de tiempo entre diferentes *smartbands*.

La otra línea de trabajo futuro es referente al algoritmo usado, SDATEP. Por un lado, es un algoritmo que puede ser modificado de diversas maneras según esté evolucionando el estudio, no solo de manera central sino también en los parámetros del mismo. De esta manera mejora la calidad de los patrones de comportamiento que se obtienen.

También existe la posibilidad de cambiar este algoritmo que utiliza la aplicación, ya que, solo necesita ejecutarlo dentro de su aplicación local. Los datos y el sistema de conexión de *smartbands* no deben de verse modificados si se ejecuta un algoritmo distinto, siendo así la aplicación una plataforma de estudio de posibles cambios.

Para finalizar, se destaca que se abren unas puertas interesantes a la posibilidad de estudio de unos patrones humanos que, no solo pueden ser interesantes para la salud o el control de uno mismo, ya que las reglas nos permitirían conocer casos raros donde se obtienen datos fuera de lo normal, sino para estudios sociales o psicológicos de las personas.

Bibliografía.

[1] E. UDIN, «Gizchina,» 26 Abril 2021. [En línea]. Available: https://www.gizchina.com/2021/04/26/xiaomi-mi-band-6-official-sales-exceed-one-million-units/. [Último acceso: 9 Junio 2022].

[2] AIHealth, «AiHealth,» AI Health , LLC, 2021. [En línea]. Available: https://www.aihealth.ai/. [Último acceso: 9 Junio 2022].

[3] R. Ferrero, «Maxima Formación,» Septiembre 2020. [En línea]. Available: https://www.maximaformacion.es/blog-dat/que-es-la-ciencia-de-datos/. [Último acceso: 26 Mayo 2022].

[4] E. Pérez, «Xataka,» 20 Mayo 2019. [En línea]. Available: https://www.xataka.com/aplicaciones/aosp-asi-al-android-open-source-google-que-queda-como-opcion-para-huawei. [Último acceso: 16 Junio 2022].

[5] Oracle, «Java Oracle,» [En línea]. Available: https://www.java.com/es/about/. [Último acceso: 24 Abril 2022].

[6] J. S. A.E. Eiben, Introduction to Evolutionary Computation, Berlin: Springer Verlag, 2003.

[7] J. H. Holland, Adaptation in Natural and Artificial Systems, London: MIT Press, 1992.

[8] D. E. Goldberg, Genetic Algoriths, Champaign: Pearson Education India, 2013.

[9] A. M. Andaluz, «Universidad carlos tercero de madrid,» Agosto 2018. [En línea]. Available: http://www.it.uc3m.es/~jvillena/irc/practicas/estudios/aeag. [Último acceso: 26 ABril 2022].

[10] Huawei, «Huawei developer,» [En línea]. Available: https://developer.huawei.com/consumer/en/doc/development/hmscore-common-Guides/get-started-hmscore-0000001212585589.

[11] Á. M. G. Vico, «Github,» 16 Junio 2020. [En línea]. Available: https://github.com/SIMIDAT/FEPDS. [Último acceso: 5 Febrero 2022].

[12] IBM, «IBM data mining goals,» 27 Febrero 2021. [En línea]. Available: https://www.ibm.com/docs/es/db2/11.1?topic=overview-data-mining-goals. [Último acceso: 11 Mayo 2022].

[13] C. J. C. P. G. H. S. y. M. J. d. J. Ángel M.García-Vico, «An overview of emerging pattern mining in supervised descriptive rule discovery: taxonomy, empirical

study, trends, and prospects,» *WIREs Data Mining and knowledge discovery,* vol. 8, nº 1, p. e1231, 2017.

[14] K. C. C. Wai-Ho Au, «Mining Changes in Association Rules: A Fuzzy Approach,» *Fuzzy Sets and Systems,* vol. 149, pp. 637-646, 2005.

[15] A.-L. C. H.-H. C. Mu-Chen Chen, «Mining changes in customer behavior in retail marketing,» *Expert Systems with Applications,* vol. 28, nº 4, pp. 773-781, 2005.

[16] J. Gama, Knowledge Discovery from Data Streams, Boca Raton: CRC Press, 2010.

[17] F. Herrera, «Genetic fuzzy systems: Taxonomy, current research trends and prospects,» *Evolutionary Intelligence ,* vol. 1, pp. 27-46, 2008.

[18] C. B. F. H. J. M. B. Lala Septem Riza, « Fuzzy Rule-Based Systems for Classification,» *Journal of Statistical Software,* vol. 65, nº 6, pp. 1-30, Mayo 2015.

[19] L. A. Zadeh, «Fuzzy logic and approximate reasoning,» *Synthese,* vol. 30, nº 3, pp. 407-428, 1975.

[20] A. L. F. D. F. G. J. G. G. Z. J. Lu, «Learning under Concept Drift: A Review,» *IEEE Transactions on Knowledge and Data Engineering,* vol. 31, nº 12, pp. 2346-2363, 2019.

[21] G. H. R. K. B. P. A. Bifet, «MOA: massive,» *Journal of Machine Learning Research,,* vol. 11, pp. 1601-1604, 2010.

[22] C. J. C. P. G. H. S. y. M. J. d. J. Ángel M.García-Vico, «FEPDS: A Proposal for the Extraction of Fuzzy,» *IEEE Transactions on Fuzzy Systems,* vol. 28, nº 12, pp. 3193-3203, 2020.

[23] G. Paynter, «ARFF file format,» 1 Noviembre 2008. [En línea]. Available: https://www.cs.waikato.ac.nz/ml/weka/arff.html. [Último acceso: 9 Mayo 2022].

[24] C. J. C. D. M. M. G. M. J. d. J. A.M. García-Vico, «An overview of emerging pattern mining in supervised descriptive rule discovery: taxonomy, empirical study, trends, and prospects,» *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery,* vol. 8, nº 1, p. e1231, 2018.

[25] J. Nielsen’s, Usability Engineering, Morgan Kaufmann, 1993.

[26] w3school, «w3school,» [En línea]. Available: https://www.w3schools.com/html/.

[27] Varios, «Stack overflow,» [En línea]. Available: https://stackoverflow.com/.

[28] G. F. API, «Developers google,» Google, 27 Abril 2021. [En línea]. Available: https://developers.google.com/fit/. [Último acceso: 26 Abril 2022].

[29] P. G. C. C. y. M. d. J. A.M. García-Vico, « Impact of the Type of Rule in Fuzzy Emerging Pattern Mining on a Big Data Approach,» de *Proc. of the II International symposium on Fuzzy and Rough Sets (ISFUROS 2017)*, 2017.

[30] A. M. Andaluz, «IT UC3M,» Agosto 2018. [En línea]. Available: http://www.it.uc3m.es/~jvillena/irc/practicas/estudios/aeag. [Último acceso: 26 abril 2022].

[31] Apple, «Support Apple,» Apple, 1 junio 2022. [En línea]. Available: https://support.apple.com/es-es/HT208109. [Último acceso: 22 Abril 2022].

[32] D. Caballero, «Memoria en movil,» Movil Zona, 25 Mayo 2022. [En línea]. Available: https://www.movilzona.es/tutoriales/rendimiento/almacenamiento-necesario-movil/. [Último acceso: 16 Mayo 2022].

[33] conceptodefininicon, «conceptodefinicion.de,» conceptodefininicon, 30 Enero 2021. [En línea]. Available: https://conceptodefinicion.de/ios/. [Último acceso: 19 Mayo 2022].

[34] N. L. Dinkal Shah, «Literature Review about Emerging Pattern and Frequent Pattern Growth algorithm apply on gene Expression data,» *International journal of innovative research in technology,* vol. 1, nº 11, 2014.

[35] G. d. españa, «protección de datos personales,» Gobierno de españa, 5 Diciembre 2018. [En línea]. Available: https://www.chj.es/es-es/ciudadano/Atencionalciudadano/Paginas/Preguntasfrecuentesprotecci%C3%B3ndatospersonales.aspx#:~:text=La%20protecci%C3%B3n%20de%20datos%20personales,integridad%20y%20disponibilidad%20de%20esta.. [Último acceso: 17 Mayo 2022].

[36] geeksforgeeks, «Geeks For geeks,» Geeks For geeks, 2 Noviembre 2020. [En línea]. Available: https://www.geeksforgeeks.org/. [Último acceso: 27 Abril 2022].

[37] D. Geyshis, «Aporia,» 4 Abril 2021. [En línea]. Available: https://www.aporia.com/blog/concept_drift_in_machine_learning_101/. [Último acceso: 19 Mayo 2022].

[38] A. N. Gonzalez, «Xatakandroid,» 9 Febrero 2011. [En línea]. Available: https://www.xatakandroid.com/sistema-operativo/que-es-android. [Último acceso: 19 Mayo 2022].

[39] Huawei, «Band4 pro especificaciones,» Huawei, Diciembre 2019. [En línea]. Available: https://consumer.huawei.com/es/wearables/band4-pro/specs/. [Último acceso: 25 Abril 2022].

[40] Huawei, «Huawei developer,» 20 Octubre 2021. [En línea]. Available: https://developer.huawei.com/consumer/en/doc/development/. [Último acceso: 26 Abril 2022].

[41] LaGaceta, «La gaceta,» 8 Enero 2019. [En línea]. Available: https://www.lagacetadesalamanca.es/hemeroteca/10-caracteristicas-ios-10-debes-conocer-CEGS185043. [Último acceso: 19 Mayo 2022].

[42] I. Linares, «Mi Band 5 análisis,» 19 Enero 2021. [En línea]. Available: https://www.xataka.com/analisis/xiaomi-mi-band-5-analisis-caracteristicas-precio-especificaciones. [Último acceso: 24 Abril 2022].

[43] L. López, «smartwatchzone,» 18 Agosto 2018. [En línea]. Available: https://www.smartwatchzone.net/samsung-health-servicios-conectados/. [Último acceso: 25 Abril 2022].

[44] Á. M.García-Vico, «SIMIDAT FEPDS Github,» 16 Junio 2020. [En línea]. Available: https://github.com/SIMIDAT/FEPDS. [Último acceso: 4 Mayo 2022].

[45] E. Medina, «MuyLinux,» 14 Marzo 2017. [En línea]. Available: https://www.muylinux.com/2017/03/14/lenguajes-programacion-2017/. [Último acceso: 19 Mayo 2022].

[46] J. Penalva, «Galaxy Fit 2 análisis,» 27 Enero 2021. [En línea]. Available: https://www.xataka.com/analisis/samsung-galaxy-fit2-analisis-caracteristicas-precio-especificaciones. [Último acceso: 25 Abril 2022].

[47] Samsung, «Galaxy fit 2 especificaciones,» Septiembre 2020. [En línea]. Available: https://www.samsung.com/es/watches/galaxy-fit/galaxy-fit2-black-sm-r220nzkaeub/. [Último acceso: 25 Abril 2022].

[48] Statcounter, «statcounter,» 22 Abril 2022. [En línea]. Available: https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202103-202203-bar. [Último acceso: 22 Abril 2022].

[49] Monche, «Timetoast,» 2018. [En línea]. Available: https://www.timetoast.com/timelines/versiones-del-sistema-operativo-ios. [Último acceso: 22 Abril 2022].

[50] WebGenio, «WebGenio,» [En línea]. Available: https://webgenio.com/blog/que-es-android-y-que-es-un-telefono-movil-android/. [Último acceso: 20 Abril 2022].

[51] Wikipedia, «Minería de datos,» 24 Mayo 2020. [En línea]. Available: https://es.wikipedia.org/wiki/Miner%C3%ADa_de_datos. [Último acceso: 11 Mayo 2022].

[52] Wikipedia, «Wikipedia,» 29 Enero 2022. [En línea]. Available: https://es.wikipedia.org/wiki/Ciencia_de_datos. [Último acceso: 26 Mayo 2022].

[53] R. E. Y. C. Yuhui Shi, «sci2s Implementation of Evolutionary Fuzzy Systems,» *IEEE TRANSACTIONS ON FUZZY SYSTEMS,* vol. 7, nº 2, pp. 109-119, Abril 1999.

Apéndice A: Manual de usuario

Se van a explicar en este apartado cómo realizar una conexión con las APIs usando Android Studio, cómo instalar lerning, cómo utilizarlo y qué significan los *Logs* mostrados por la aplicación.

Es importante remarcar que todos los proyectos aquí abiertos serán entregados conjuntamente con la memoria, por lo que, el manual de usuario usará esos proyectos para la explicación y guía de los mismos.

A.1 Apertura del algoritmo SDATEP

Como se había comentado en el apartado 4.3.1 el algoritmo SDATEP está montado en un proyecto del IDE NetBeans, por ello se va a explicar cómo incluirlo para poder modificar o compilarlo.

El primer paso deberá ser tener NetBeans instalado, se encuentra disponible en “https://netbeans.apache.org/download/index.html" y posteriormente seleccionar la pestaña “File” y la apertura de un proyecto existente, como muestra la Figura 46.

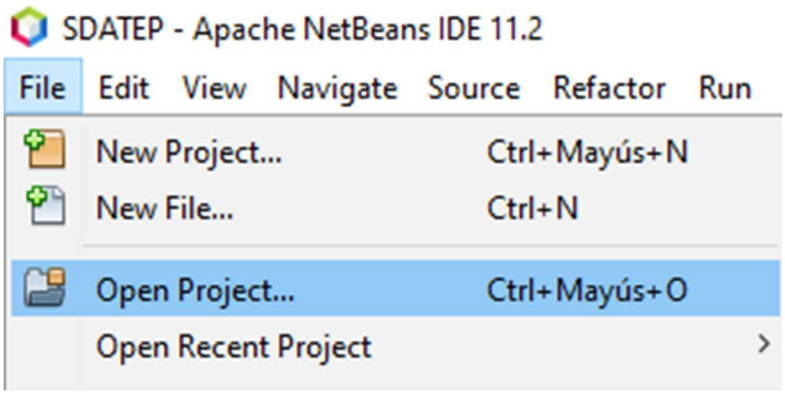

Figura 46.- Apertura de proyecto SDATEP.

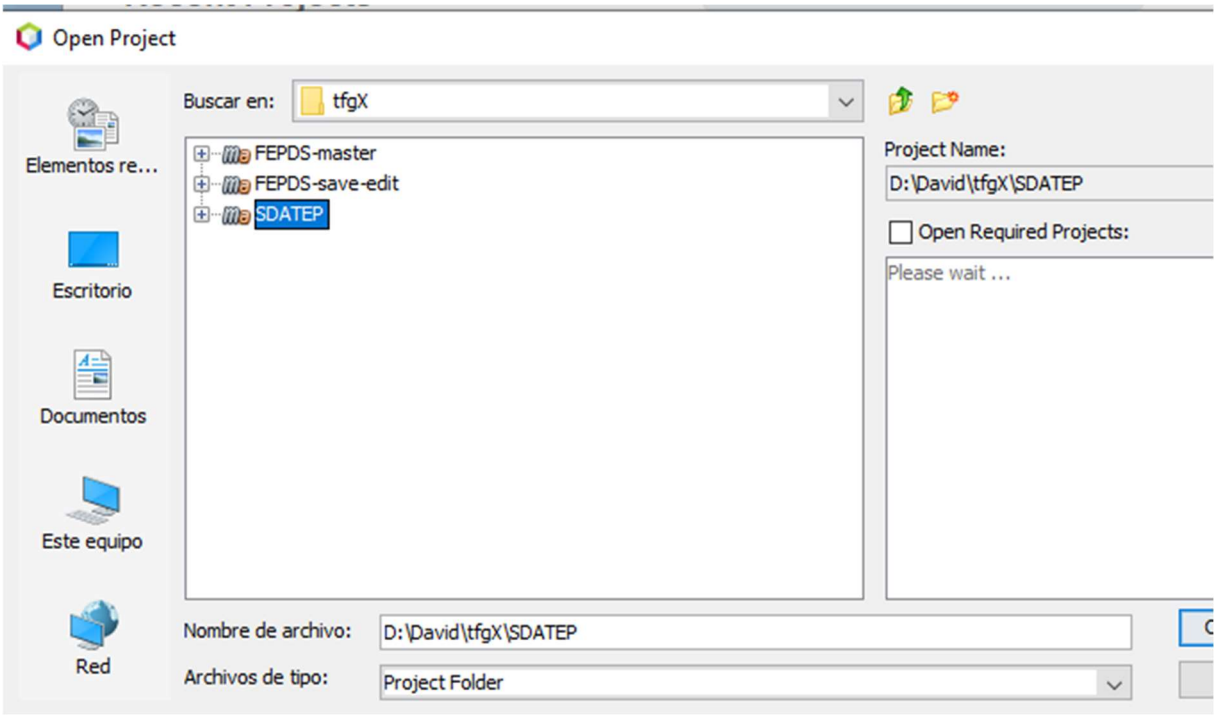

Figura 47.- Selección del proyecto SDATEP.

Tras ello aparecerá la ventana de la Figura 47, se debe seleccionar el proyecto SDATEP para abrirlo. Con ello se importará y abrirá correctamente el proyecto donde está el algoritmo SDATEP, además, NetBeans escaneará las dependencias que existan dentro del proyecto y las descargará automáticamente, esto puede llevar un tiempo dependiendo de la conexión a internet de la máquina y de la potencia de la misma.

A.2 Apertura de lerning

De igual manera que en el apartado 1.1 se va a incluir una explicación de como importar y abrir el proyecto lerning, donde se encuentra la aplicación Android. El IDE utilizado es Android Studio, por ello es necesario una instalación previa del mismo, se encuentra disponible en “https://developer.android.com/studio?hl=es”.

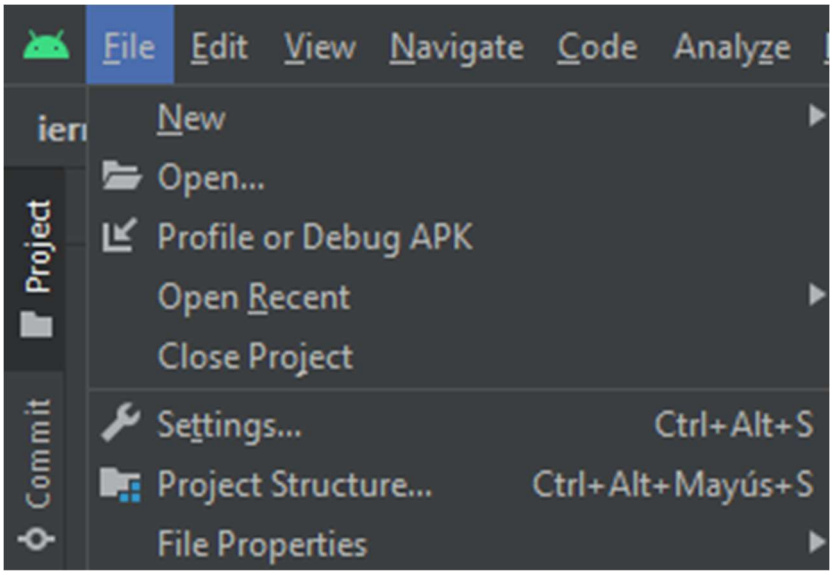

Figura 48.- Apertura del proyecto lerning.

Una vez abierto Android Studio, se debe seleccionar la pestaña “File”, posteriormente la pestaña “Open…”, para pasar a la siguiente pestaña y poder escoger el proyecto que se desea abrir, Figura 48.

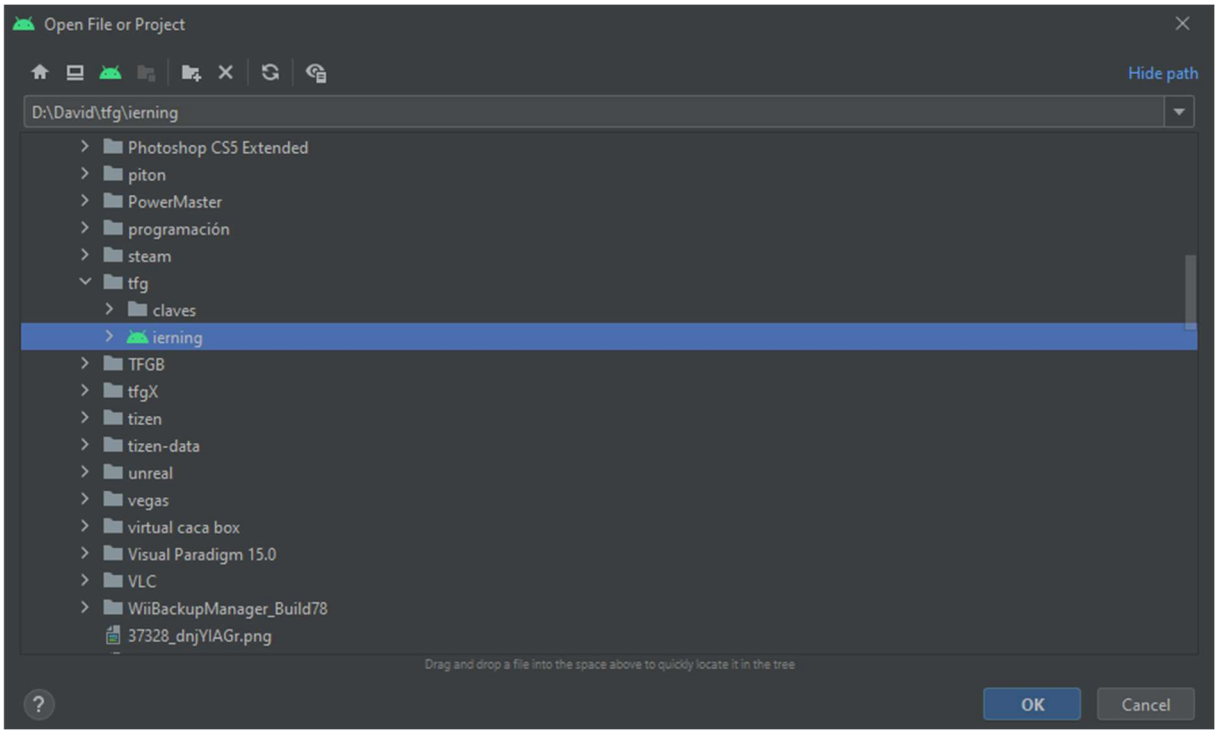

Figura 49.- Selección del proyecto lerning.

En la Figura 49 se muestra un explorador de archivos, en el cual, se debe buscar la carpeta donde se encuentra el proyecto lerning, marcarlo y pulsar al final el botón de “OK”. Con ello se abrirá el proyecto, las dependencias existentes se instalarán en

su mayoría, pero debido a que lerning requiere de instalación de credenciales y permisos se va a realizar un apartado 1.5 exclusivo para esta explicación.

A.3 Instalación de lerning y aplicaciones de salud

La instalación de lerning debe de realizarse desde la apk resultante de la compilación del proyecto o desde la apk suministrada conjuntamente con la memoria.

Google Fit, Mi Fit y Salud Huawei pueden encontrarse en Google Play Store, tal y como muestran las Figuras 61, 62 y 63.

Figura 50.- Google Fit Store. Fuente Google Play Store.

Mi Fit recibió recientemente un cambio de nombre a Zepp Life, pero la aplicación continúa funcionando de manera similar y transparente para la API.

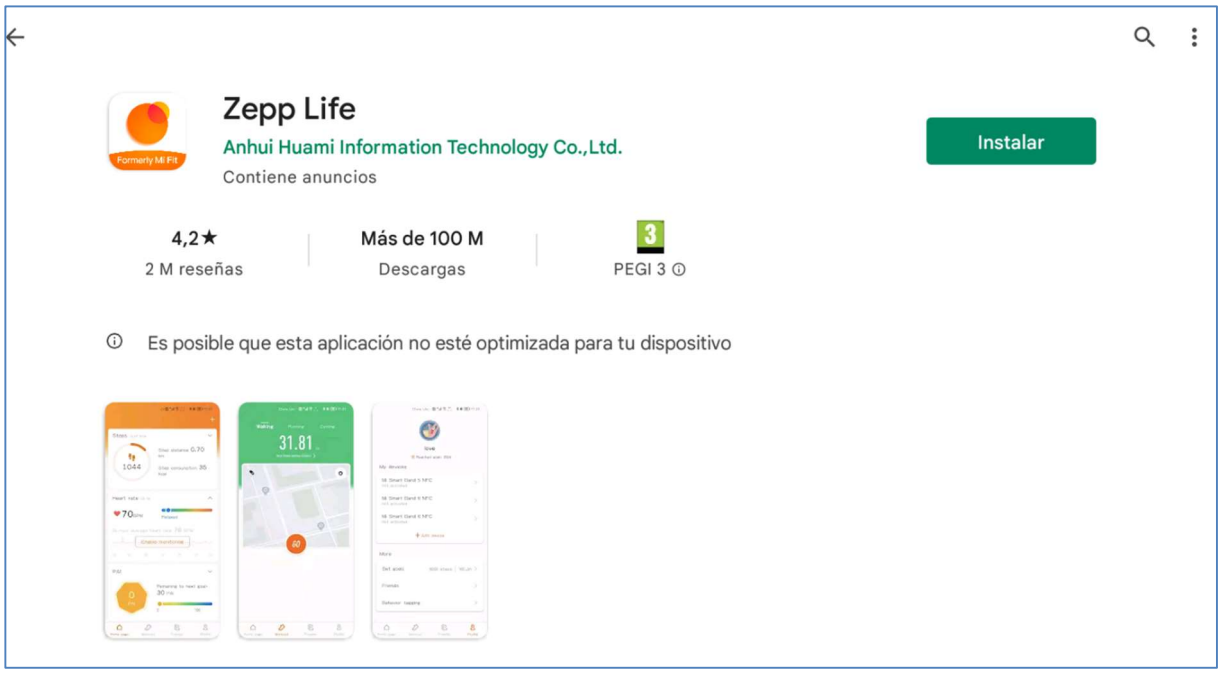

Figura 51.- Mi Fit. Fuente Google Play Store.

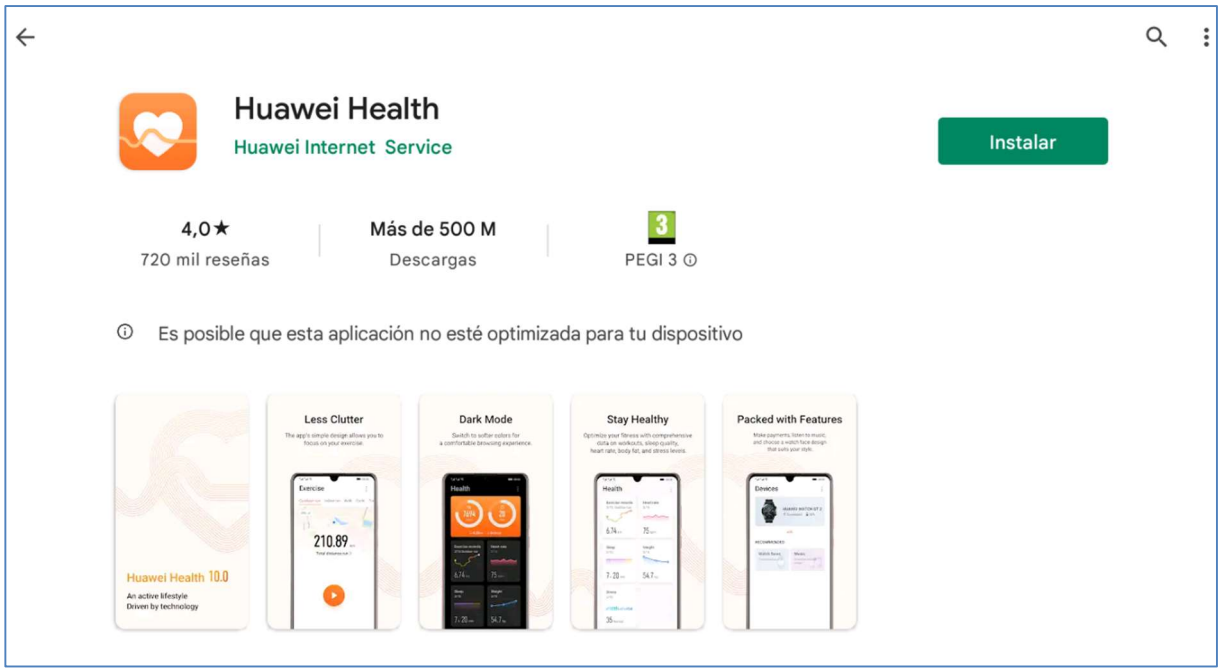

Figura 52.- Huawei Health. Fuente Google Play Store.

Huawei Health pedirá la instalación de HMS core, esta instalación se realiza de manera automática por lerning tras solicitar acceso de desarrollador.

A.4 Conexión con las bandas inteligentes

Se mostrará como realizar la conexión de las bandas inteligentes con Huawei y con Mi Fit.

A.4.1 Huawei

Habiendo descargado los programas desde el apartado 1.3, dentro de la aplicación de Huawei se abrirá el apartado de dispositivos, como se muestra en la Figura 53.

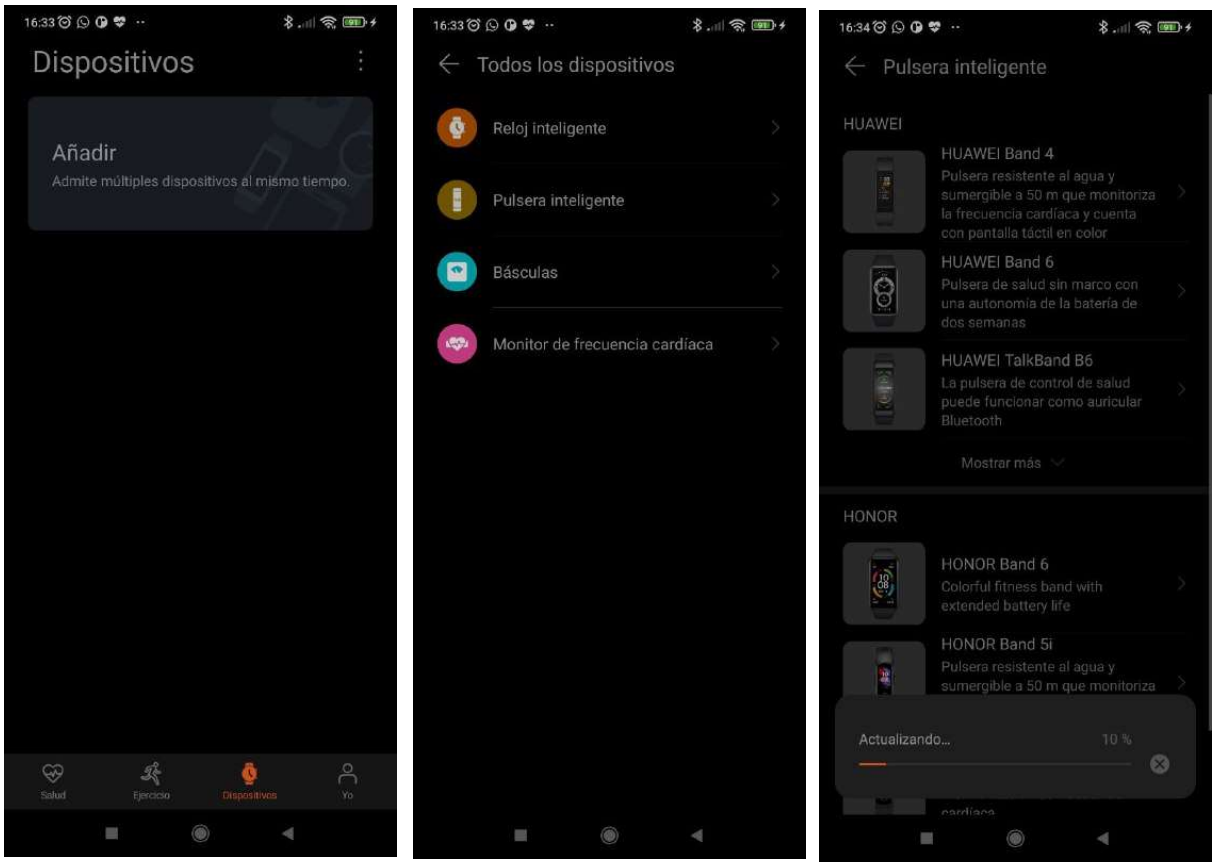

Figura 53.- Pasos pantalla de dispositivos Huawei. Fuente propia.

Una vez pulsado el botón de añadir, se pasará a la parte 2 de la Figura 53, donde se debe seleccionar que tipo de dispositivo se desea vincular con la aplicación. Para enlazar las bandas inteligentes se debe pulsar el botón de “Pulsera inteligente”, yendo de esa manera hacia la pantalla que muestra la parte final de la Figura 53, donde se debe escoger la banda/pulsera inteligente a vincular. En el caso de ejemplo se realiza con la banda Huawei Band 4 pro.

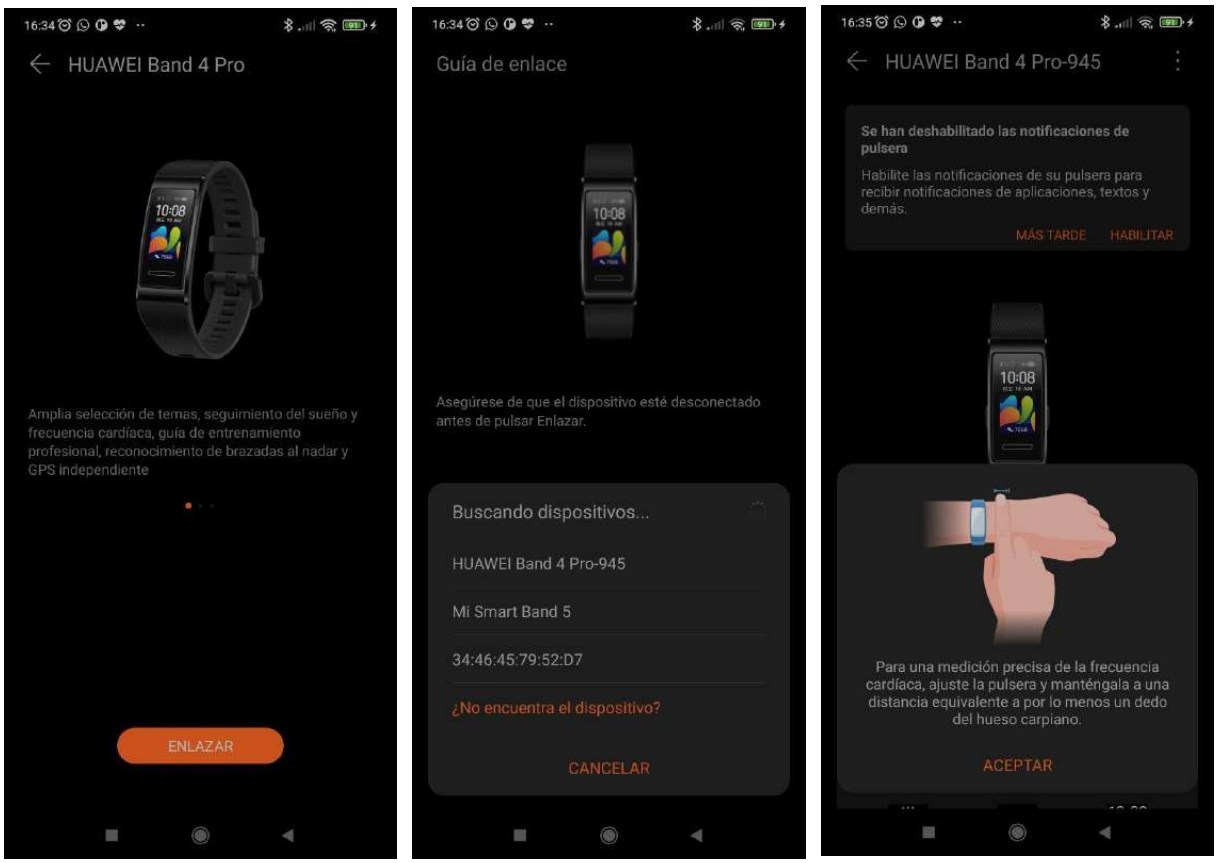

Figura 54.- Pasos enlazado de dispositivo Huawei. Fuente propia.

En la Figura 54 se realiza el enlazado del dispositivo seleccionado, primero en la imagen de la izquierda se podrá confirmar que es el dispositivo deseado, pasando posteriormente a la imagen central de la Figura 54, donde se debe realizar una conexión bluetooth con la banda inteligente. Una vez seleccionada la banda inteligente HUAWEI Band 4 pro, el enlace estará realizado, es posible que pida un código de identificación, en ese caso la propia banda inteligente proporcionará un código de cuatro o cinco dígitos para introducirlo en la aplicación. Acabados los pasos se tendrá la banda completamente enlazada.

A.4.2 Mi Fit

De manera similar que el apartado anterior, se van a indicar los pasos que deben seguirse para conectar una banda inteligente, mi Band 5, a la aplicación Mi Fit.

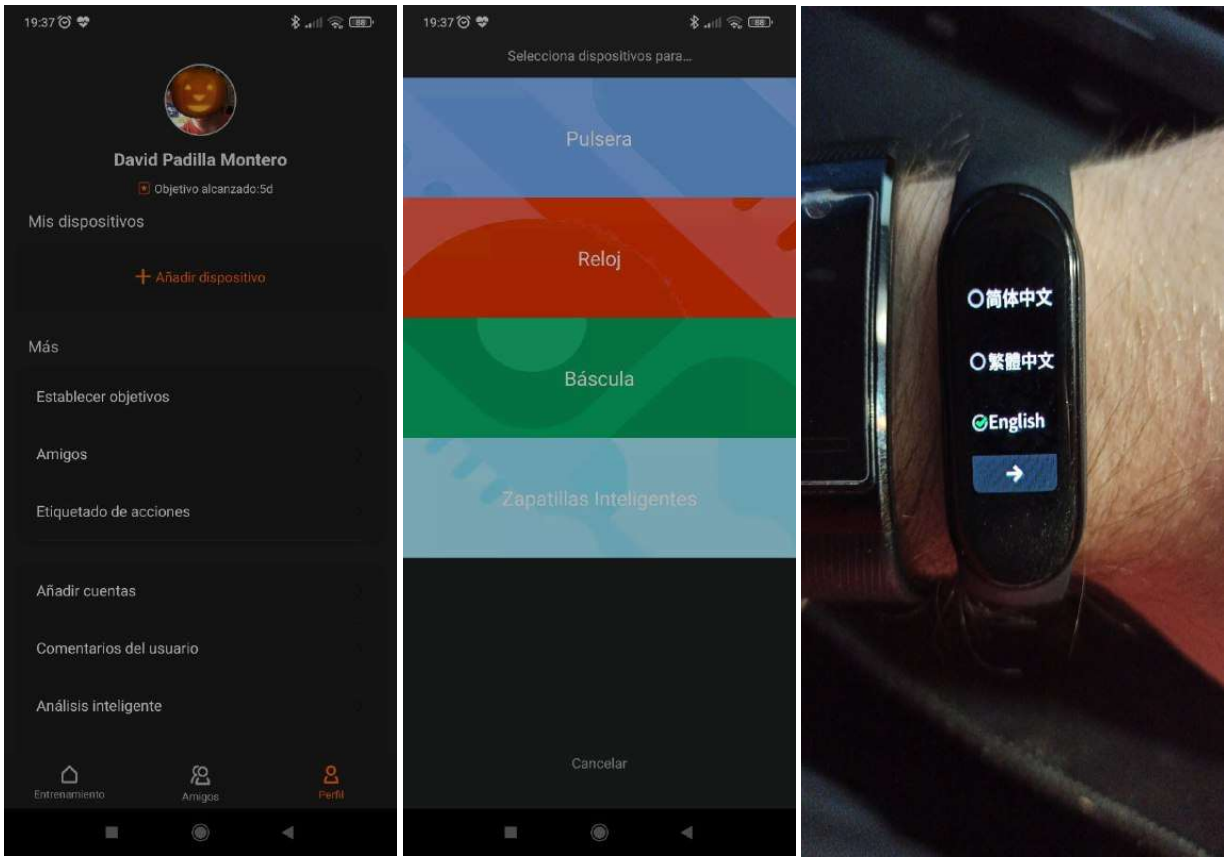

Figura 55.- Pasos selección de dispositivo Mi Fit. Fuente propia.

Primero debe abrirse la aplicación “Mi Fit” para posteriormente seleccionar el perfil, como aparece en la primera imagen de la Figura 55. Posteriormente aparecerá la segunda imagen, donde se debe seleccionar el tipo de dispositivo a enlazar, para poder enlazar la Mi Band 5 se debe pulsar en pulsera.

La banda inteligente Mi Band 5 aparece como la última imagen de la Figura 55, donde se pide seleccionar el idioma, por defecto en inglés, pero es posible seleccionar aquel idioma con el que el usuario se sienta cómodo.

Se verá en la aplicación “Mi Fit” la primera imagen de la Figura 56, donde comenzará a buscar las bandas inteligentes cercanas, siendo así como aparece en la segunda imagen, la banda inteligente que se desea conectar pedirá confirmación. Tras confirmar dicho enlazado en la banda inteligente, la aplicación “Mi Fit” mostrará una pantalla de confirmación de que el enlazado ha sido correcto.

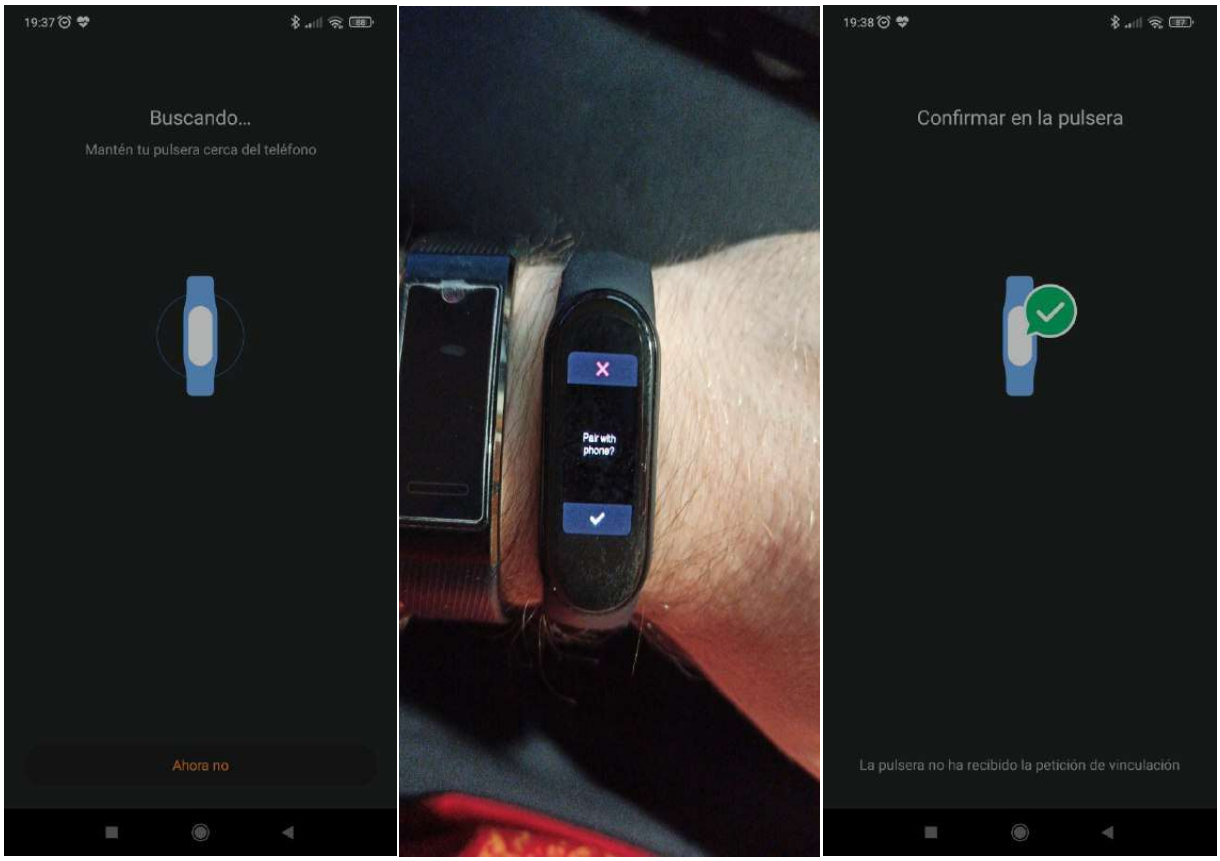

Figura 56.- Pasos enlazado Mi Fit. Fuente propia.

A.5 Configuración de APIs

Siguiendo los pasos del apartado 1.2 el proyecto lerning estará abierto y descargando las dependencias que aún no estén presentes en el IDE, pero aun así existen algunas configuraciones que aparecerán como error. En este apartado se va a ayudar a la configuración de esos puntos revisando los archivos de configuración y comentando los cambios realizados.

A.5.1 AndroidManifest.xml

En la Figura 57 se muestra la configuración del archivo, AndroidManifest.xml, en este archivo los cambios importantes son:

- Campos meta-data: ya que son parte de la configuración de HSM core y es necesario tenerlos.
- Campos uses-permission: es necesario la petición de permisos de escritura y lectura de archivos, además del permiso de actividad, que será necesario para las peticiones a las APIs.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.tfg.ierning">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Ierning"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Ierning">

        <meta-data
            android:name="com.huawei.hms.client.appid"
            android:value="104634299"/>
        <meta-data
            android:name="com.huawei.hms.client.channel.androidMarket"
            android:value="false" />

        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:label="Ierning">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>



    </application>


    <uses-permission android:name="android.permission.ACTIVITY_RECOGNITION"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
</manifest>
```

Figura 57.- AndroidManifest.xml. Fuente propia.

A.5.2 build.gradle(lerning)

Tal y como aparece en la Figura 58 no debería de realizarse ningún cambio en dicho archivo, ya que únicamente es un archivo de repositorios y deberían funcionar correctamente al realizar la importación del proyecto.

```
// Top-level build file where you can add configuration options common to all sub-projects/modules.
buildscript {
    repositories {
        google()
        mavenCentral()
        maven {url 'https://developer.huawei.com/repo/'}

    }
    dependencies {
        classpath "com.android.tools.build:gradle:7.0.1"
        classpath 'com.google.gms:google-services:4.3.10'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}
```

Figura 58.- build.gradle(lerning). Fuente propia.

A.5.3 build.gradle(:app)

En las Figuras 59 y 60 aparece la configuración del fichero build.gradle(:app), este fichero si requiere de un poco de configuración por parte del usuario, en concreto el apartado de “signingConfigs”, debido a que se debe establecer rutas de ficheros dentro del propio ordenador donde se está lanzando el IDE. Siendo en “debug” el apartado “storeFile file”, donde se debe poner la dirección de la key del proyecto lerning, que se entrega conjuntamente con la memoria.

En el apartado “debugFit” y el subapartado “storeFile file” se debe de indicar la localización por defecto de la localización de claves de Android Studio, este proceso es automático, pero es recomendable revisarlo y cambiarlo antes de que pueda generar algún error.

En el apartado de dependencias se han incluido bastantes de estas para la utilización de las APIs y los servicios de Google, al importar el proyecto todas las usadas estarán incluidas por defecto, pero en caso de error, la Figura 60 contiene todas las dependencias que se usan.

```
plugins {
    id 'com.android.application'
    id 'com.google.gms.google-services'
}

android {
    signingConfigs {
        debug {
            storeFile file('D:\\David\\tfg\\claves\\mymishi.jks')
            storePassword '5kl2m22'
            keyAlias 'key0'
            keyPassword '5kl2m22'
        }
        debugFit {
            storeFile file('C:\\Users\\Pacalor\\.android\\debug.keystore')
            storePassword 'android'
            keyPassword 'android'
            keyAlias 'androiddebugkey'
        }
    }
    compileSdk 31

    defaultConfig {
        applicationId "com.tfg.ierning"
        minSdk 21
        targetSdk 31
        versionCode 1
        versionName "1.0"

        testInstrumentationRunner 'androidx.test.runner.AndroidJUnitRunner'
        resConfigs "en", "zh-rCN"
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
```

Figura 59.- build.gradle(:app). Fuente propia.

```
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    buildFeatures {
        viewBinding true
    }
}

dependencies {


    implementation 'androidx.appcompat:appcompat:1.0.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'androidx.lifecycle:lifecycle-livedata:2.0.0'
    implementation 'androidx.lifecycle:lifecycle-viewmodel:2.0.0'
    implementation 'androidx.navigation:navigation-fragment:2.0.0-rc02'
    implementation 'androidx.navigation:navigation-ui:2.0.0-rc02'
    implementation 'com.google.android.material:material:1.0.0'
    implementation 'com.google.firebase:firebase-auth:21.0.1'
    implementation files('libs\\SDATEP.jar')
    testImplementation 'junit:junit:4.+'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.0'
    implementation 'com.huawei.hms:health:6.2.0.300'

    implementation 'com.google.android.gms:play-services-fitness:20.0.0'
    implementation 'com.google.android.gms:play-services-auth:19.2.0'

}
```

Figura 60.- build.gradle(:app) continuación. Fuente propia.

A.5.4 proguard-rules.pro

La configuración de este archivo la exige la API de Huawei, por lo que el archivo importado del proyecto debería de quedar de la misma manera que la Figura 61, ya que este archivo se ha completado siguiendo el asesoramiento de Huawei para el correcto funcionamiento de su API.

```
-ignorewarnings
-keepattributes *Annotation*
-keepattributes Exceptions
-keepattributes InnerClasses
-keepattributes Signature
-keepattributes SourceFile,LineNumberTable
-keep class com.huawei.hianalytics.**{*;}
-keep class com.huawei.updatesdk.**{*;}
-keep class com.huawei.hms.**{*;}
```

Figura 61.- proguard-rules.pro. Fuente propia.

A.5.5 settings.gradle

En este archivo se han incluido los repositorios de Google y de Huawei, tal y como se muestra en la Figura 62.

```
dependencyResolutionManagement { DependencyResolutionManagement it ->
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories { RepositoryHandler it ->
        google()
        mavenCentral()
        jcenter() // Warning: this repository is going to shut down soon
        maven {url 'https://developer.huawei.com/repo/'}
    }
}
rootProject.name = "Ierning"
include ':app'
```

Figura 62.- settings.gradle. Fuente propia.

A.6 Uso de lerning

Se va a mostrar ahora cómo es la interfaz de la aplicación lerning y cómo usarla.

La Figura 63 muestra la vista principal de la aplicación, que se divide en siete apartados que se irán explicando a continuación.

1. Caja para introducir valor numérico, representa la cantidad de etiquetas que usará el algoritmo.
2. Caja para introducir valor numérico, representa la cantidad de días que usará el algoritmo.
3. Desplazable, para indicar si se quiere realizar el estudio con valores de actividad o de sueño.
4. Botón que ejecutará el algoritmo con los valores escritos en 1 ,2 y seleccionado 3, en caso de no tener ninguno escrito, se tomarán valores por defecto.
5. Botón que permite ver las reglas de la última ejecución del algoritmo.
6. Botón que escribe en una memoria de uso común los resultados del algoritmo. Estos archivos se pueden enviar como cualquier otro a un ordenador, por correo, programas de mensajería, etc.
7. Botón que realiza la descarga de todos los datos de las *smartbands* a la base de datos.

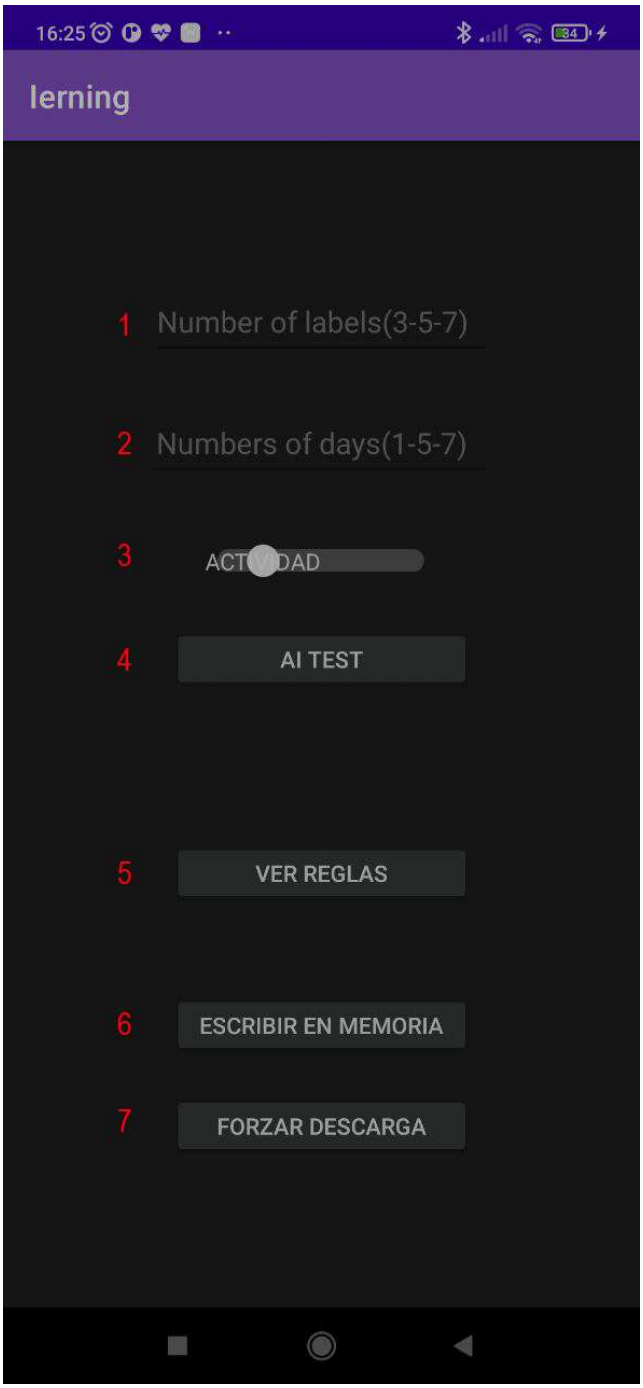

Figura 63.- lerning. **Fuente propia.**

A.7 Explicación de los Logs

En referencia a los *Logs*, existen dos tipos diferentes, el primero, referente a la petición de datos de un día a las APIs y el segundo referente a la ejecución del algoritmo SDATEP.

```
2022-05-24 12:59:50.556 32338-32472/com.tfg.ierning E/Ierning: Mon May 23 00:00:00 GMT+02:00 2022
2022-05-24 12:59:50.557 32338-32472/com.tfg.ierning E/Ierning: Tue May 24 00:00:00 GMT+02:00 2022
2022-05-24 12:59:51.100 32338-32472/com.tfg.ierning I/Ierning: FIT Range Start:2022-05-23T00:00+02:00[Europe/Monaco]
2022-05-24 12:59:51.101 32338-32472/com.tfg.ierning I/Ierning: FIT Range End:2022-05-24T00:00+02:00[Europe/Monaco]
2022-05-24 12:59:57.621 32338-32338/com.tfg.ierning I/Ierning: Success read a SampleSets from HMS core
2022-05-24 12:59:57.621 32338-32338/com.tfg.ierning I/Ierning: 5
2022-05-24 12:59:57.622 32338-32338/com.tfg.ierning I/Ierning: 5
2022-05-24 13:00:03.984 32338-32472/com.tfg.ierning I/System.out: /data/user/0/com.tfg.ierning/files/IerningDB
2022-05-24 13:00:03.984 32338-32472/com.tfg.ierning E/Ierning: Data base writing finish
```

Código 9.- Log petición.

El primer *Log* referente al Código 9, muestra primeramente en rojo, el día del que se ha hecho la petición a la API de Huawei, posteriormente la misma fecha para la API de Google Fit con el prefijo “FIT” al inicio, esas serían las primeras cuatro líneas de tiempo que aparecen.

Aparecerá después una confirmación de HSM core, donde se indica su correcto funcionamiento y la cantidad de datos que se han extraído, aparece dos a modo de confirmación.

Por último, la localización de la base de datos que se está usando y la confirmación de que se ha terminado de escribir en ella.

```
2022-05-24 13:00:10.692 32338-32338/com.tfg.ierning E/Ierning: 2022-05-23T13:00:10.691+02:00[Europe/Madrid]
2022-05-24 13:00:10.773 32338-32338/com.tfg.ierning E/Ierning: /data/user/0/com.tfg.ierning/files/param.txt
2022-05-24 13:00:13.902 32338-32338/com.tfg.ierning E/Ierning: labels3days5
2022-05-24 13:00:13.902 32338-32338/com.tfg.ierning E/Ierning: test10
```

Código 10.- Log ejecución SDATEP.

Para el Código 10, referente al segundo tipo de *Log* que tenemos, relacionado con la ejecución del algoritmo, el cual se divide en cuatro líneas, la primera, referente al tiempo de estudio, es decir, qué día es el grupo de estudio. La segunda línea es la ubicación del archivo de parámetros, en caso de necesitarlo. La tercera línea indica el número de etiquetas usadas para el experimento y el número de días contra el que se enfrentará el día de estudio. Por último, un mensaje de confirmación, siendo 10 el número de OK en el resultado, un número menor supone un error en algún apartado del algoritmo.