



**UNIVERSIDAD DE JAÉN**  
*Escuela Politécnica Superior (Jaén)*

Trabajo Fin de Máster

# **MODELADO INCREMENTAL DE INFORMACIÓN CONTEXTUAL MEDIANTE REGLAS DE ASOCIACIÓN DIFUSAS**

**Alumno/a: Pérez Cano, Francisco Daniel**

Tutor/a: José María Serrano Chica

Dpto.: Departamento de Informática

**Julio, 2018**



Universidad de Jaén  
Escuela Politécnica Superior de Jaén  
Departamento de Informática

Don José María Serrano Chica, tutor del Proyecto Fin de Carrera titulado: Modelado incremental de información contextual mediante reglas de asociación difusas, que presenta Francisco Daniel Pérez Cano, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Julio de 2018

El alumno:

Francisco Daniel Pérez Cano

Los tutores:

José María Serrano Chica

## Agradecimientos

Estos años han sido un camino largo, y aunque no han sido pocos los obstáculos que tanto mis compañeros como yo nos hemos encontrado, puedo decir con satisfacción que he conseguido recorrerlo hasta el final. Por todo ello me gustaría agradecer a mi familia el apoyo que me ha brindado a lo largo de todos estos años, ya que sin su ayuda no estaría aquí en estos momentos. Es por eso que mis palabras no pueden ser más que de gratitud hacia todos ellos, que han tratado de darme sus mejores consejos en todo momento, de ofrecerme una buena educación, y de darme un apoyo constante. He podido comprobar que trabajando duro es posible conseguir cualquier meta que nos propongamos, y es que como decía mi padre todo esfuerzo tiene su recompensa.

También me gustaría dar las gracias de manera muy especial a D. José María Serrano Chica, tutor de mi Trabajo Fin de Máster. Agradecer su crítico y constructivo acompañamiento a lo largo de este Trabajo en todo momento.

Por último, es necesario reconocer la importante labor que han tenido el resto de profesores que me han impartido alguna asignatura a lo largo de mi formación, así como su dedicación e interés.

## Contenido

Índice de figuras .....	7
Índice de tablas .....	10
Capítulo 1. Introducción .....	11
1.1. Introducción.....	11
1.2. Motivación .....	12
1.3. Objetivos .....	13
Capítulo 2. Problema a resolver .....	15
2.1. Introducción.....	15
2.2. Sistemas de redes de sensores .....	16
2.3. Análisis de datos masivos .....	17
2.4. Minería de datos.....	18
2.5. Datos difusos .....	23
Capítulo 3. Tecnologías y herramientas utilizadas .....	24
3.1. Hardware.....	24
3.1.1. Raspberry Pi 3.....	24
3.1.2. Sense Hat.....	25
3.1.3. GPIO Breakout board .....	25
3.1.4. LM393.....	26
3.1.5. HC-SR501 PIR .....	26
3.2. Tecnologías.....	27
3.2.1. MySQL.....	27
3.2.2. CodeIgniter PHP Framework.....	27
3.2.3. Python 3 .....	27
3.2.4. Bootstrap CSS Framework .....	28
3.2.5. Java Android.....	28
3.3. Software .....	28

3.3.1.	Android Studio .....	28
3.3.2.	PuTTY .....	29
3.3.3.	FileZilla .....	30
3.3.4.	RESTClient.....	30
3.3.5.	Adobe Photoshop .....	31
3.3.6.	Visual Paradigm.....	31
3.3.7.	KNIME .....	32
Capítulo 4.	Ingeniería del software .....	33
4.1.	Análisis.....	33
4.1.1.	Requisitos funcionales.....	33
4.1.2.	Requisitos no funcionales.....	34
4.2.	Diseño .....	34
4.2.1.	Modelo Vista Controlador .....	34
4.2.2.	Diagramas de secuencia .....	35
4.2.3.	Diseño de la interfaz .....	38
4.2.4.	Esquema de la base de datos .....	39
4.3.	Planificación .....	42
4.3.1.	Planificación general.....	42
4.3.2.	Planificación de la implementación.....	44
4.3.3.	Estimación de costes.....	45
4.4.	Implementación.....	49
4.5.	Pruebas.....	60
Capítulo 5.	Estudio de los datos obtenidos.....	64
5.1.	Análisis de los datos.....	64
5.2.	Planteamiento de la representación de los datos.....	71
5.3.	Evaluación y validación de los resultados .....	77
Capítulo 6.	Conclusiones y trabajo futuro .....	88

Capítulo 7. Bibliografía .....	90
Anexo I Material adjunto.....	94
Anexo II Guía de instalación y de uso .....	95

## Índice de figuras

Ilustración 1 Esquema redes de sensores .....	16
Ilustración 2 Las tres V del Big Data .....	17
Ilustración 3 Esquema general de un sistema de recomendación .....	19
Ilustración 4 Raspberry Pi .....	24
Ilustración 5 Sense Hat .....	25
Ilustración 6 Sensores BMP085 y DHT22 .....	25
Ilustración 7 Placa de expansión de los pines.....	26
Ilustración 8 Sensor de luminosidad LM393.....	26
Ilustración 9 Sensor de movimiento HC.SR501 PIR .....	26
Ilustración 10 Android Studio.....	29
Ilustración 11 Ejemplo del emulador de Android Studio .....	29
Ilustración 12 Terminal de PuTTY .....	30
Ilustración 13 FileZilla.....	30
Ilustración 14 REST Client .....	31
Ilustración 15 Interfaz Photoshop.....	31
Ilustración 16 Interfaz de Visual Paradigm .....	32
Ilustración 17 KNIME.....	32
Ilustración 18 Esquema del Modelo Vista Controlador.....	35
Ilustración 19 Diagrama de secuencia del envío de datos desde la Raspberry Pi .....	36
Ilustración 20 Diagrama de secuencia del envío de los datos desde el móvil..	37
Ilustración 21 Diagrama de secuencia de la interacción entre el usuario y el sistema para ver los datos.....	38
Ilustración 22 Prototipo de la aplicación para dispositivos Android .....	38
Ilustración 23 Storyboard de la interfaz de la aplicación web .....	39
Ilustración 24 Esquema de la base de datos (I) .....	40
Ilustración 25 Esquema de la base de datos (II) .....	41
Ilustración 26 Esquema de la base de datos (III) .....	42
Ilustración 27 Diagrama de Gantt.....	43
Ilustración 28 Pines de salida y ajustes del sensor de movimiento .....	50
Ilustración 29 Esquema de los pines del sensor de luminosidad .....	50

Ilustración 30 Sense Hat. La zona marcada indica la zona de conexión con la Raspberry Pi.....	51
Ilustración 31 Vista superior de la Raspberry Pi.....	52
Ilustración 32 Vista lateral de la Raspberry Pi.....	52
Ilustración 33 Logo de la aplicación Android.....	53
Ilustración 34 Interfaz de la aplicación Android.....	53
Ilustración 35 Código para la generación de la tabla de forma dinámica.....	56
Ilustración 36 Gráfico acelerómetro.....	56
Ilustración 37 Gráfico giroscopio.....	57
Ilustración 38 Gráfico gravedad.....	57
Ilustración 39 Gráfico magnetismo.....	57
Ilustración 40 Gráfico luminosidad.....	58
Ilustración 41 Gráfico movimiento.....	58
Ilustración 42 Gráfico humedad.....	58
Ilustración 43 Gráfico presión.....	59
Ilustración 44 Gráfico temperatura.....	59
Ilustración 45 Gráfico bluetooth.....	59
Ilustración 46 Reglas de asociación actualizadas.....	60
Ilustración 47 Tabla con el valor de las métricas utilizadas para cada regla ...	60
Ilustración 48 Reglas de asociación actualizadas teniendo en cuenta la tabla anterior.....	60
Ilustración 49 Colocación de los sensores respecto a la vivienda.....	61
Ilustración 50 RESTClient ejemplo de una petición al servidor.....	62
Ilustración 51 RESTClient datos devueltos por la petición.....	63
Ilustración 52 RESTClient envío de datos al servidor.....	63
Ilustración 53 Disposición de los ejes en un teléfono móvil.....	64
Ilustración 54 Datos del acelerómetro.....	65
Ilustración 55 Giroscopio.....	65
Ilustración 56 Datos de gravedad.....	66
Ilustración 57 Datos del sensor de gravedad.....	66
Ilustración 58 Iluminación de la vivienda.....	67
Ilustración 59 Datos del sensor de movimiento.....	68
Ilustración 60 Medida de la humedad tras un evento.....	68
Ilustración 61 Humedad del ambiente con el aire acondicionado.....	69

Ilustración 62 Medidas del sensor de presión .....	69
Ilustración 63 Medidas del sensor de temperatura.....	70
Ilustración 64 Estado del inquilino dentro de la vivienda .....	70
Ilustración 65 Gráfica de presión de la Raspberry.....	74
Ilustración 66 Presión en la estación meteorológica .....	74
Ilustración 67 Análisis exploratorio en KNIME .....	78
Ilustración 68 Datos mostrados en el nodo "Interactive Table" de KNIME .....	78
Ilustración 69 Datos mostrados en el nodo "Interactive Table" de KNIME tras tratar los valores perdidos .....	79
Ilustración 70 Estudio estadístico de los datos.....	79
Ilustración 71 Correlación lineal entre los diferentes pares de variables.....	80
Ilustración 72 Diagrama de caja .....	81
Ilustración 73 Esquema KNIME para la obtención de las reglas de asociación	81
Ilustración 74 Configuración algoritmo Apriori .....	82
Ilustración 75 Configuración del algoritmo FP-Growth .....	82
Ilustración 76 Resultados devueltos por el algoritmo Apriori .....	84
Ilustración 77 Resultados obtenidos con el algoritmo FP-Growth .....	86

## Índice de tablas

Tabla 1 Planificación general del TFM .....	43
Tabla 2 Coste de los componentes específicos del proyecto.....	46
Tabla 3 Resumen de todos los componentes y su precio .....	47
Tabla 4 Recursos humanos.....	48
Tabla 5 Resumen de todos los costes del proyecto .....	48
Tabla 6 Resumen datos recogidos.....	71
Tabla 7 Estado del usuario a través de los sensores del dispositivo móvil .....	73
Tabla 8 Resultados del algoritmo Apriori.....	85
Tabla 9 Resultados del algoritmo FP-Growth.....	86

## Capítulo 1. Introducción

### 1.1. Introducción

El desarrollo sin precedentes en las tecnologías de la información (TI), facilitando la comunicación en la sociedad, ha provocado que tengamos un gran volumen de datos a nuestra disposición. La mayor parte de esta información se encuentra encerrada en bases de datos que hacen que la información que contienen sea potencialmente importante pero aún no ha sido descubierta. Estos datos suponen una fuente de riqueza muy importante y la extracción de conocimiento sobre estos datos juega un papel fundamental en la actualidad.

La minería de datos se encarga de dar respuesta a esta necesidad permitiendo la explotación de grandes conjuntos de datos obteniendo como resultado información implícita, desconocida y potencialmente útil de los datos. El gran volumen de datos ha provocado que el análisis de los mismos ya no pueda ser manual y deba ser automático. Esta extracción de conocimiento (Knowledge Discovery in Databases, KDD) se compone de 4 fases principalmente: la selección, el pre-procesamiento, la minería de datos (en sí, en realidad la minería de datos es una etapa dentro del proceso de KDD, como vemos) y la evaluación o interpretación. La minería de datos es el paso más complejo porque engloba múltiples métodos a la hora de abordar el proceso de identificación de patrones en los datos, y no es un proceso trivial. En la extracción de conocimiento de una base de datos, un ejemplo de técnica son las reglas de asociación (RA), que permiten la identificación de asociaciones o correlaciones entre diferentes elementos de un conjunto de datos.

Por otro lado, los sistemas de redes de sensores (en inglés, wireless sensor network, WSN) son conjuntos de sensores distribuidos espacialmente para monitorizar condiciones físicas o ambientales. Constituyen actualmente una fuente inagotable de datos procedentes del entorno que pueden ser utilizados para caracterizar el contexto en el que se mueve el sistema, el usuario y cualquier elemento del mismo que precise ser monitorizado. Esta información puede ser analizada para construir un modelo contextual del entorno examinado, actuando como base a la hora de determinar un conjunto de opciones sobre las que el usuario debe tomar alguna decisión. Muchos datos de los que se recogen contienen algún tipo de incerteza por lo que pueden acarrear problemas a la hora

de extraer conocimiento de los mismo. Los conjuntos difusos son de utilidad para representar la información no numérica permitiendo un acercamiento del razonamiento computacional al que utilizamos las personas. Esta teoría permite expresar de forma matemática conceptos con cierto grado de ambigüedad.

El presente Trabajo Fin de Máster se centra en el uso de estos sistemas sobre el hábitat que rodea a una persona, para la extracción de información del entorno y la caracterización de dichos datos mediante técnicas basadas en la teoría de los subconjuntos difusos. Sobre el modelo generado, se aplicarán diferentes técnicas de análisis exploratorio teniendo en cuenta la evolución de los datos recogidos a lo largo del tiempo. Por último, se evaluarán y validarán los resultados obtenidos.

## 1.2. Motivación

La cantidad de datos que hay en el mundo no deja de crecer constantemente y no se prevé que pare a corto plazo. La facilidad a la hora de obtener los datos mediante los sistemas de redes de sensores, el avance de las tecnologías y el bajo costo que suponen la mayoría de estos sistemas han tenido un gran impacto en el aumento de la cantidad de información que se maneja actualmente. Además, la facilidad y bajo costo del almacenamiento de los datos ha provocado que se guarden cosas que antiguamente se eliminaban. Estos datos recogen nuestras decisiones, nuestras compras del supermercado, nuestros hábitos diarios, etc. Todos estos datos no aportan nada, en este estado, al mundo del comercio y la industria. Sin embargo, el análisis de estos datos nos permite obtener beneficio de ellos [1]. Las grandes corporaciones buscan patrones de comportamiento sobre sus consumidores para conocer sus gustos y necesidades. Desarrollan sus líneas de trabajo basándose en la información obtenida, mejorando su competitividad y ofreciendo a sus clientes lo que necesita y quiere realmente. Para ello utilizan métodos eficientes y eficaces para extraer y transformar los datos de una organización en información y conocimiento.

Por lo tanto, técnicas como la minería de datos nos aportan muchos beneficios en diferentes sectores como el marketing donde se mejora la competitividad de la empresa, instituciones financieras donde ayuda a determinar si un préstamo es beneficioso para la entidad o en la fabricación

donde se puede utilizar para detectar equipos defectuosos y los parámetros de control óptimos. Este proceso, combinado con la constitución de una red de sensores para la extracción de los datos me permitirá desarrollar un proceso completo, desde la recogida hasta el análisis, de tratamiento de datos poniendo en práctica gran parte de los conocimientos obtenidos durante todos mis años de formación.

### 1.3. Objetivos

El presente TFM tiene el objetivo principal de desarrollar un modelo general de representación para manejar la información contextual procedente de redes de diferentes sensores. Para cumplirlo habrá que cumplir los siguientes objetivos:

- Crear un prototipo de un sistema de redes de sensores que recojan datos del entorno para poder procesarlos posteriormente.
- Determinar un entorno concreto de trabajo para la implantación del sistema con el objetivo de utilizar los datos obtenidos para realizar trabajos futuros.
- Ajustar los sensores para obtener la mayor precisión posible.
- Preprocesar los datos obtenidos.
- Desarrollar una herramienta que sea capaz de modelar los datos numéricos y continuos mediante conjuntos de etiquetas lingüísticas.
- Realizar una comparativa entre diferentes algoritmos para la extracción de reglas de asociación difusas, estudiando sus características y los resultados obtenidos tras aplicarlos sobre los datos obtenidos.
- Crear un mecanismo asociado que se encargue de mantener actualizadas las reglas frente a cambios en los datos de entrada de forma dinámica.
- Visualización de los resultados obtenidos tras el proceso.

En este capítulo 2 de la memoria se tratará el problema a resolver. En el capítulo 3, se verá el estado en el que se encuentran las diferentes técnicas que se utilizarán a lo largo del proyecto y la importancia de las mismas. En el capítulo 4 me centraré en analizar todas las herramientas y tecnologías utilizadas, así como los datos que tendré disponibles para el correcto desarrollo del proyecto

que se detallará en el capítulo 5. En este capítulo se detallará todo lo relacionado con la implementación y las diferentes tecnologías utilizadas, así como la planificación y la estimación de costes. En el capítulo 6 se estudiarán los resultados obtenidos tras estudiar los datos y por último en el capítulo 7 se mostrarán los resultados y el trabajo futuro.

## Capítulo 2. Problema a resolver

### 2.1. Introducción

Como hemos visto en el capítulo de introducción, la minería de datos es un proceso complejo pero que aporta muchos beneficios en diferentes sectores. Esto hace que sea un campo de moda dentro de la informática. A su vez, el avance de las tecnologías ha permitido que la extracción de datos del medio natural sea más sencilla principalmente debido a los sistemas de redes de sensores. Este trabajo se centra en combinar ambos aspectos para poner en práctica gran parte de la materia trabajada durante el curso académico. Uno de los principales puntos es la contextualización del ámbito de estudio. Normalmente la minería de datos es un proceso que se suele centrar en mayor medida en el ámbito empresarial. Sin embargo, en este trabajo he decidido utilizarla para examinar el entorno que rodea a una persona con el objetivo de extraer información sobre sus hábitos diarios para analizarlos y poder trabajar con ellos en un futuro próximo como, por ejemplo, el desarrollo de una aplicación para mejora de los hábitos que siguen estas personas.

Por lo tanto, el trabajo se centra en la creación de un prototipo, de un sistema completo, encargado de la extracción de datos del entorno que rodea a una persona mediante un sistema de red de sensores y su posterior análisis utilizando diferentes técnicas como la obtención de patrones de los datos mediante la extracción de las reglas de asociación difusas. Para ello, utilizaré una Raspberry Pi [2] ya que es un hardware de bajo costo y permite montar infraestructuras sencillas pero muy potentes para los objetivos planteados en este proyecto y un teléfono móvil para obtener los datos que rodean al individuo. Un servidor se encargará de recibir los datos de los dispositivos a través de una API REST y se encargará de almacenarlos y mostrarlos a través de una interfaz web. Además, se mostrarán las reglas de asociación existentes entre las diferentes variables que forman en modelo de forma actualizada, es decir, se comprobará que reglas, de las extraídas a lo largo del estudio del conjunto de datos, se mantienen vigentes para los últimos datos recogidos. Para obtener las reglas de asociación que existen entre las diferentes variables, utilizaré los algoritmos más conocidos: el Apriori [3] y el FP-Growth [4]. En los siguientes

capítulos detallaran las tecnologías y algoritmos utilizados para llevar a cabo el proyecto y la justificación del porqué se han elegido.

## 2.2. Sistemas de redes de sensores

Los sistemas de redes de sensores son una fuente inagotable de datos procedentes del entorno que monitorizan. La facilidad y el bajo costo de implementar uno de estos sistemas ha hecho que muchos usuarios creen sus propias redes de sensores para controlar los diferentes aspectos del hogar. Esta facilidad junto con el avance en la interconexión entre diferentes dispositivos y el avance del internet de las cosas (IoT) ha provocado que se produzca un crecimiento exponencial en la cantidad de datos que son recogidos, así como la necesidad del avance de la tecnología para poder dar cabida a dichos datos.

Las redes de sensores se componen de sensores automáticos distribuidos para monitorizar las condiciones físicas o ambientales. Estas redes se componen de sensores autónomos que se utilizan para observar elementos del entorno como la temperatura, humedad, presión, posición, vibración, sonido, etc. Una de sus características principales es que no se utiliza una infraestructura de red, es decir, no existe la necesidad de alguna infraestructura para operar ya que sus nodos pueden actuar de emisores, receptores o enrutadores de la información, aunque la información generada se recoge normalmente en un ordenador y es el que hace posible la transmisión de los datos a través de otras tecnologías. A continuación, se muestra un esquema general de una red de sensores.

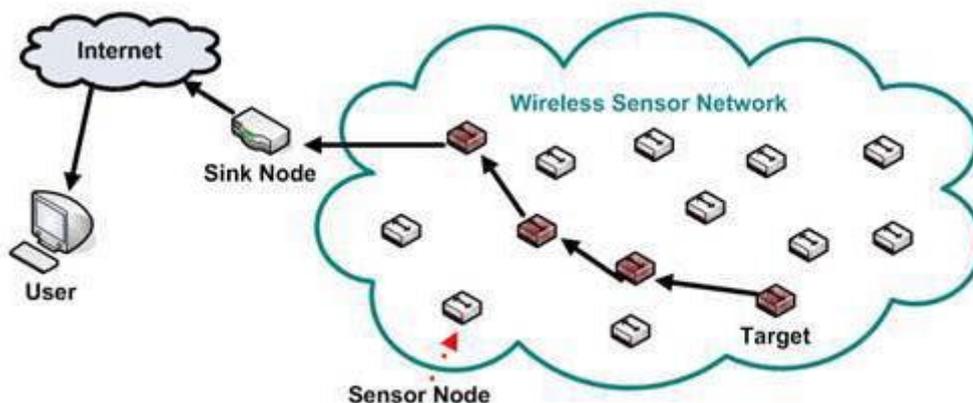


Ilustración 1 Esquema redes de sensores

### 2.3. Análisis de datos masivos

Desde los años 90 han existido problemas para analizar grandes cantidades de datos. En esa época las empresas almacenaban su información en empresas de almacenamiento terciario por lo que la migración de los sistemas de bases de datos era muy lenta. Además, las bases de datos no proporcionaban ninguna funcionalidad para los usuarios interesados en aprovechar esta información [3]. Actualmente existen diferentes enfoques y técnicas que nos ayudan a explorar y analizar la información estructurada sobre un área para descubrir tendencias o patrones, a partir de los cuales derivar ideas y extraer conclusiones [5]. El avance de las tecnologías ha permitido que la mayoría de las empresas tengan su información en almacenes de datos que proporcionan una estructura global, común e integrada de los datos permitiendo su acceso en tiempo real y con capacidades de análisis. El concepto “Big Data” nace a partir de estos avances y hace referencia a un conjunto de datos tan grande que las aplicaciones informáticas tradicionales no tienen capacidad suficiente para procesarlos. Tradicionalmente, las características más importantes del Big Data se han clasificado en tres magnitudes, más conocidas como las tres V del Big Data, relativas a volumen, variedad y velocidad.

- El volumen se refiere a la cantidad de datos que se generan en el entorno. Según Gartner, una compañía líder mundial en investigación y asesoría empresarial, el crecimiento del volumen de datos es exponencial.
- La velocidad se refiere a la rapidez con la que los datos son creados, almacenados y procesados en tiempo real.
- La variedad se refiere a las formas, tipos y fuentes en las que se registran los datos. Estos pueden ser datos estructurados o no estructurados.
- Algunos autores añaden algunas más, hasta obtener incluso 42 V's del Big Data, como la propuesta de Tom Shafer [6].

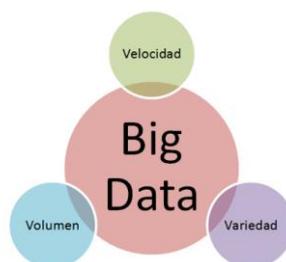


Ilustración 2 Las tres V del Big Data

## 2.4. Minería de datos

La minería de datos es un proceso que se encarga de explorar estos datos mediante diferentes técnicas y tecnologías con el objetivo de encontrar patrones relevantes, tendencias y reglas. La minería de datos es una la etapa más compleja del proceso de extracción de conocimiento por la gran complejidad y dificultad que engloba abordar el proceso de identificación de patrones. Este proceso se compone de 4 fases principalmente: la selección, el pre-procesamiento, la minería de datos y la evaluación o interpretación. La exploración, depuración y mejora de la calidad de los datos son pasos previos a la etapa de exploración y análisis de los mismos. El objetivo principal de la exploración es resaltar las características de los atributos e identificar la relación entre los diferentes atributos, es decir, busca las características más relevantes de los datos. La preparación o preprocesamiento de los datos se compone de una serie de tareas para mejorar la calidad de los datos previos al proceso de extracción de conocimiento. Entre estas tareas se pueden destacar la limpieza, la normalización o discretización, la integración o la eliminación de variables redundantes. La mejora de la calidad de los datos es crucial para obtener buenos resultados y es vital tener definida una estrategia para tratar los datos “sucios”. Podemos definir los datos “sucios” como la ausencia de valores en ciertos atributos, errores en los valores o inconsistencias. Una vez se tienen listos los datos, ya es posible comenzar la búsqueda del conocimiento. La minería de datos es un proceso iterativo e interactivo, en el cual se usan, a través de medios automáticos y semiautomáticos, técnicas de aprendizaje inteligente sobre una o varias fuentes de datos para aprender del pasado y mejorar la toma de decisiones en el futuro [7]. La elección del método utilizado para la búsqueda de patrones en los datos es una tarea esencial a la vez que complicado ya que hay que seleccionar el algoritmo más adecuado que ayude a alcanzar el objetivo final. La interpretabilidad es importante y condiciona el algoritmo a usar. Logicalis, una empresa internacional líder en operaciones con las TI, clasifica los algoritmos según las diferentes posibilidades del área en:

1. Aprendizaje automático: puede darse de dos formas: bajo supervisión, también conocido como modelado predictivo y sin supervisión, conocido como clustering. Existen numerosos algoritmos como el Naïve-Bayes

basado en métodos probabilísticos, el algoritmo C4.5 basado en árboles de decisión o el K-Means que buscan la predicción de sucesos a través de unos conjuntos de entrenamiento para conocer mejor la organización, mejorar la toma de decisiones, mejorar la competitividad del negocio, cumplir retos personales o contribuir a la investigación. Además de estos algoritmos, existen otra serie de algoritmos más orientados al análisis de transacciones y mercados principalmente.

2. Análisis de secuencias: se ocupan de descomponer una serie de tiempo en componentes de tendencia, estacionales o cíclicos e irregulares. Obtienen secuencias frecuentes o eventos en los datos.
3. Sistemas de recomendación (RS): son otro ejemplo de herramienta que utilizan las empresas para mejorar la experiencia de los usuarios. Estos sistemas forman parte del sistema de filtrado de información, se encargan de proporcionar una experiencia personalizada al usuario a través de recomendaciones basadas en las características y costumbres de los usuarios. Además, los recomendadores tienen cierto nivel de autonomía a la hora de presentar recomendaciones a los usuarios. Empresas como Amazon, Netflix o YouTube cuentan con estos sistemas desde hace muchos años. En la ilustración 2 se muestra un esquema de como manejan la información los sistemas de recomendación a la hora de realizar sus recomendaciones.

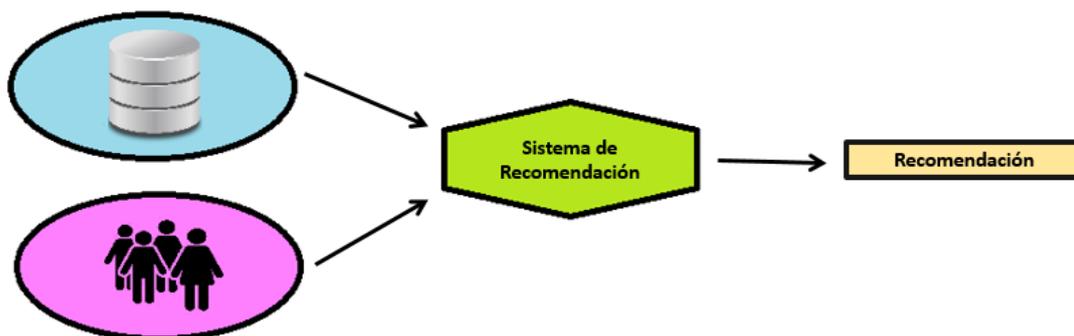


Ilustración 3 Esquema general de un sistema de recomendación

4. Minería de reglas de asociación: permiten la identificación de asociaciones o correlaciones entre diferentes elementos, es decir, encuentran reglas que predigan ocurrencias de un ítem en función de las ocurrencias de otros ítems en otra transacción. Se aplican normalmente para analizar transacciones y mercados. En el presente Trabajo Fin de

Máster me basaré en estos algoritmos a la hora de trabajar con los datos, por lo que es importante definir ciertos conceptos:

- a. Conjunto: colección de uno o más ítems.
- b. Conteo de soporte: frecuencia de ocurrencia del conjunto.
- c. Soporte: Proporción de transacciones que contienen un conjunto.
- d. Conjunto frecuente: conjunto cuyo soporte es mayor o igual que el umbral establecido.
- e. Regla de asociación: expresión de la forma  $X \rightarrow Y$  donde ambos son conjuntos.

Además, existen diferentes métricas para evaluar las reglas: el soporte y la confianza. La primera métrica mide el porcentaje de transacciones que contienen ambas mientras que la confianza mide la frecuencia con la que los ítems de un conjunto aparecen en transacciones del otro conjunto. Este valor debe ser alto para asegurarnos de que las transacciones aparecen con mucha frecuencia. El soporte por su parte debe ser menor cuanto más grande es el conjunto de datos que se analiza.

Por lo general, el problema se descompone en la generación de los conjuntos frecuentes y en la generación de reglas con una alta confianza entre cada conjunto. La confianza ha recibido diversas críticas debido a que no es una métrica que no establece de forma adecuada el grado de independencia estadística, ni refleja las dependencias negativas entre el antecedente y el consecuente. Asimismo, no es del todo intuitiva de entender a la hora de definir su umbral [8].

Existen dos algoritmos principales para la generación de reglas: el algoritmo Apriori [3] y el algoritmo FP-Growth [4]. Los principales problemas que tienen, se encuentran a la hora de generar los candidatos [9]. Las reglas de asociación difusas son una herramienta muy utilizada dentro de la minería de datos. Las reglas de asociación difusas (RAD) son reglas de asociación (RA) aplicadas sobre datos en los que se aplique la teoría de conjuntos difuso. Se utilizan para encontrar patrones o secuencias dentro de un conjunto de datos. Normalmente se aplican para obtener información en transacciones y obtener la correlación entre

diferentes atributos. Entre sus elementos más característicos se puede destacar las colecciones de uno o más ítems, la frecuencia con la que ocurren, la proporción de transacciones que contienen un conjunto, las propias reglas de asociación con el formato  $X \rightarrow Y$  donde  $X$  e  $Y$  son dos conjuntos de elementos y las diferentes métricas para la evaluación de las reglas. La minería de reglas de asociación tiene como objetivo encontrar todas las reglas que cumplan las siguientes métricas:

- Soporte  $\geq$  minsup
- Confianza  $\geq$  minconf

El soporte contiene el número porcentaje de transacciones que contienen tanto al conjunto  $X$  como a  $Y$ , minsup es el soporte mínimo establecido a la hora de estudiar los datos, la confianza mide la frecuencia con la que los ítems de un conjunto  $Y$  aparecen en transacciones que contienen los ítems del conjunto  $X$  y minconf es la confianza mínima establecida al analizar los datos. Hay que tener en cuenta que las reglas obtenidas de un mismo conjunto de datos tienen el mismo soporte entre sí, pero pueden tener confianzas distintas.

El enfoque habitual permite descomponer el problema en dos pasos:

1. Generar conjuntos con ítems frecuentes cuyo soporte  $>$  minsup.
2. Generar reglas de alta confianza de cada conjunto frecuente, en las que cada regla es una partición binaria de un conjunto frecuente.

La generación de conjuntos frecuentes suele ser con proceso costoso computacionalmente por lo que se utilizan diferentes estrategias a la hora de generar los conjuntos frecuentes. Estas estrategias suelen consistir en la reducción del número de candidatos mediante técnicas de poda, reducir el número de transacciones o reducir el número de comparaciones cambiando la estructura de datos utilizada para almacenar los candidatos o las transacciones. Los principales factores que afectan a la complejidad de la generación de los conjuntos frecuentes son:

1. Elección del umbral de soporte mínimo. Permite aumentar y reducir el número de candidatos.

2. Número de ítems del conjunto de datos.
3. Tamaño de la base de datos. En el enfoque Apriori, los múltiples recorridos que realiza sobre la base de datos puede provocar un aumento en el tiempo de ejecución.
4. Longitud media de las transacciones. La longitud de las transacciones aumenta en las bases de datos más densas y puede provocar un aumento de la longitud máxima de los conjuntos frecuentes y las estructuras del árbol Hash.

Para extraer conocimiento sobre los datos descritos en el apartado anterior voy a utilizar dos de los algoritmos más conocidos como son el Apriori y el FP-Growth. El algoritmo Apriori fue propuesto en 1994 por Agrawal et al. [3]. Este algoritmo utiliza una búsqueda en anchura y un árbol de Merkle [10] para los conjuntos candidatos y posteriormente se escanea el conjunto de datos para determinar los conjuntos de ítems frecuentes entre los candidatos. Tras obtener estos conjuntos, se generan reglas candidatas uniendo dos reglas que comparten el mismo prefijo en el consecuente de la regla podando aquellas que tengan subconjuntos poca confianza.

El algoritmo FP-Growth está basado en una mejora del algoritmo Apriori y fue propuesto por Han et al. [4]. Este algoritmo define una primera fase que consiste en descubrir los conjuntos frecuentes y una segunda en la que se generan las reglas de asociación entre los elementos frecuentes encontrados. La principal diferencia con el algoritmo Apriori es que la implementación utilizada es más eficiente al hacer uso de un árbol de elementos frecuentes que puede ser procesado más rápidamente que la estructura de datos usada en el Apriori [11]. Utiliza una estructura de árbol de patrones frecuentes (FP-tree) y utiliza un enfoque recursivo divide y vencerás para minar los conjuntos frecuentes. En su estudio [4], Han demostró que su algoritmo supera a otros métodos como el Apriori [3] o el TreeProjection [12] en eficacia y rendimiento. Además, existen muchos estudios que proponen variaciones para mejorar su rendimiento.

Los algoritmos de reglas de asociación tienden a generar demasiadas reglas, y muchas, no son interesantes o son redundantes.

Por ello es importante evaluar las reglas mediante métricas como el soporte, la confianza, el coeficiente de Gini, etc.

## 2.5. Datos difusos

Los datos difusos son elementos cuyos valores dan lugar a indeterminaciones, es decir, que el valor de un elemento pertenezca a un conjunto puede ser cierto con un grado parcial de verdad. Por ejemplo, si una persona mide dos metros es claramente alta, si se establece en un metro la altura de una persona baja, ambos valores nos permiten contextualizar la altura de las personas en función de una medida métrica lineal. Los subconjuntos difusos se utilizan para modelar la representación de ciertos conocimientos y mejorar así los sistemas de decisión, de ayuda a la decisión o de inteligencia artificial. La lógica difusa permite un acercamiento entre el procesamiento de la información que hacen los ordenadores y el razonamiento de las personas.

En un conjunto difuso existe un número infinito de subconjuntos difusos mientras que en un conjunto clásico tiene un número finito de subconjuntos clásicos. La teoría de subconjuntos difusos propuesta por Zadeh [13] se utiliza para la representación de posibles indeterminaciones, permitiendo el diseño de reglas de asociación difusas (RAD). Esta teoría permite el uso de distintos tipos de incertidumbre e imprecisión en los datos y extiende las operaciones entre conjuntos usuales para su utilización en conjuntos difusos. Los primeros trabajos que empezaron a hablar de reglas de asociación difusas se centraban en el uso de etiquetas difusas para conseguir reglas de asociación cuantitativas [14] [15] donde se divide el dominio del atributo en intervalos para después extraer reglas de asociación. En este tipo de reglas se divide el dominio del atributo en intervalos para obtener reglas de asociación cuyos ítems son pares del tipo (atributo, intervalo) en lugar de (atributo, valor). Estos intervalos se suelen representar mediante etiquetas lingüísticas.

## Capítulo 3. Tecnologías y herramientas utilizadas

Con el objetivo de terminar el proyecto con éxito, se han utilizado una serie de tecnologías y herramientas que me han sido de utilidad a lo largo del desarrollo.

### 3.1. Hardware

En este apartado se incluyen todos los componentes físicos utilizados para desarrollar el proyecto.

#### 3.1.1. Raspberry Pi 3

Raspberry Pi [2] es un microcomputador de bajo costo. Soporta diferentes sistemas operativos, aunque Raspbian es el oficial. El modelo utilizado es el B cuya principal novedad frente a sus antecesores es la inclusión de wifi y bluetooth sin necesidad de adaptadores. Gracias a sus posibilidades y prestaciones ha logrado convertirse en la base de muchos proyectos. En el proyecto ha sido utilizada para procesar los datos recogidos por los sensores y enviarlos a la base de datos.



*Ilustración 4 Raspberry Pi*

Desde un principio decidí el uso de una Raspberry Pi, sin tener en cuenta otras alternativas como Arduino [16], cuyas ventajas sobre la Raspberry Pi son el precio y que dispone de entradas analógicas y digitales mientras que la Raspberry Pi solo dispone de entradas digitales. Por el contrario, la Raspberry Pi dispone de una mayor potencia, memoria y capacidad, incluso puede llegar a sustituir a un ordenador en tareas simples. Además, me será de utilidad en un futuro en el que necesitaré un dispositivo con sus características.

### 3.1.2. Sense Hat

Es un componente original de la Raspberry Pi que lo dota de varios sensores como el de presión o temperatura, un joystick y una matriz LED 8x8. Es un complemento perfecto ya que abre un gran abanico de posibilidades tanto para este proyecto como proyectos futuros. En este proyecto he utilizado el módulo para recoger información de los sensores de presión, humedad y temperatura. En ciertas etapas del proyecto, también he utilizado la matriz LED para comprobar que el funcionamiento del sistema era correcto.



Ilustración 5 Sense Hat

Aunque su precio es muy superior en comparación con el de los sensores sueltos, como el sensor digital de temperatura y humedad DHT22 o el de presión BMP085, me será de utilidad en un futuro por su matriz LED y el joystick. Por lo tanto, he decidido utilizar este módulo en el proyecto.

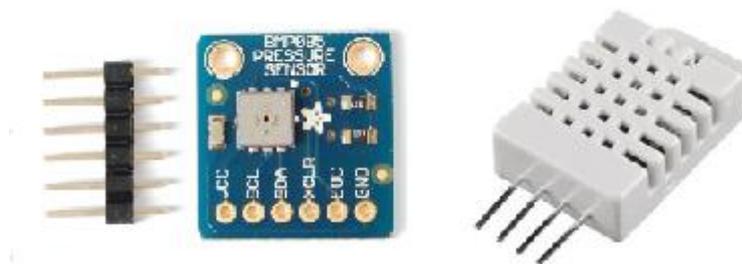


Ilustración 6 Sensores BMP085 y DHT22

### 3.1.3. GPIO Breakout board

Placa de expansión para los pines de la Raspberry Pi. Al utilizar el módulo Sense Hat, es necesario el uso de un cabezal con los pines extra largos para poder conectarlos al cable arcoíris y poder utilizar los pines desde una placa con

puentes de conexión. Aunque ocupa más espacio, me ha facilitado el cableado del proyecto.

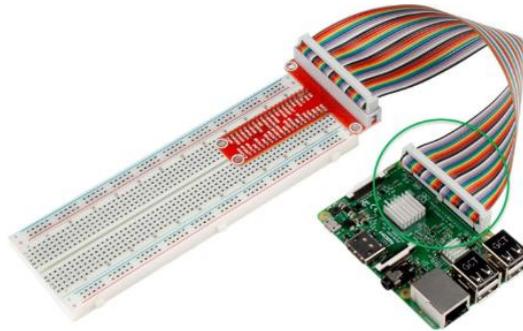


Ilustración 7 Placa de expansión de los pines

#### 3.1.4. LM393

Sensor de luz mediante una resistencia fotosensible que se utiliza para detectar el brillo de la luz ambiental. Existen dos versiones del modelo, una con tres patillas y otro con cuatro. El de tres patillas dispone de una salida digital mientras que el de cuatro dispone de una señal de salida analógica y otra digital. Es posible ajustar el umbral de brillo para la detección de luz.



Ilustración 8 Sensor de luminosidad LM393

#### 3.1.5. HC-SR501 PIR

Sensor de movimiento que utiliza un sensor infrarrojo para la detección del movimiento. Detecta movimiento dentro de un cono de 110 grados con un rango de 3 a 7 metros. Es posible ajustar la sensibilidad del sensor.



Ilustración 9 Sensor de movimiento HC.SR501 PIR

## 3.2. Tecnologías

En este apartado se incluirán todas las tecnologías utilizadas a lo largo del proyecto.

### 3.2.1. MySQL

MySQL [17] es un sistema de gestión de base de datos relacional. Está considerada como la base de datos de código abierto más popular del mundo. La utilizo para almacenar todos los datos que recojo de los sensores. En el apartado de desarrollo se detalla por qué se ha elegido este sistema de gestión de bases de datos frente a sus competidores.

Por otro lado también existen ciertas alternativas a las bases de datos relacionales como MongoDB [18] que es una base de datos NoSQL orientada a documentos. En este tipo de base de datos se crean y almacenan documentos en formato BSON, formato binario JSON. Uno de los principales beneficios que ofrece MongoDB es el uso de esquemas dinámicos sin necesidad de predefinir una estructura. Pese a que MongoDB es más eficiente con grandes volúmenes de datos, al recoger datos que se pueden mantener estructurados y el volumen de datos no va a ser demasiado grande, he optado por utilizar MySQL por su facilidad y madurez frente a MongoDB. En el prototipo desarrollado no se va a notar la diferencia de utilizar uno u otro debido al volumen de datos.

### 3.2.2. CodeIgniter PHP Framework

CodeIgniter [19] es un framework de código abierto para el desarrollo de aplicaciones web. Utiliza PHP y permite que los desarrolladores realicen sus proyectos mucho más rápido que creando toda la estructura desde cero. Está basado en el patrón Modelo-Vista-Controlador. Dispone de una gran cantidad de librerías que ayudan a completar su funcionalidad. El uso de PHP como lenguaje de programación permite que se pueda incrustar sobre el HTML. Además, el uso de PHP en los sitios web en el lado del servidor asciende a un 83.1% según un estudio realizado por W3Tech [20]. En el proyecto se utiliza la última versión de PHP junto con CodeIgniter y RESTClient para crear una API para la recolección, almacenamiento y visualización de los datos.

### 3.2.3. Python 3

Lenguaje de programación fácil de aprender cuya filosofía hace especial hincapié en la sintaxis del código [21]. El ranking de Tiobe [22] lo cataloga como

uno de los lenguajes de programación más utilizados actualmente. Permite la creación de extensiones en C o C++. En el proyecto lo he utilizado para crear leer los datos de los sensores asociados a la Raspberry y mandarlos a la API basada en REST.

#### 3.2.4. Bootstrap CSS Framework

Bootstrap [23] es un framework para el desarrollo de hojas de estilos de páginas web gratuito y de código abierto desarrollado inicialmente por Twitter. Proporciona diferentes utilidades como HTML, CSS o JavaScript para el desarrollo de sitios web siguiendo diseños responsivos. Su instalación es sencilla y lo he utilizado para simplificar el desarrollo de la parte del cliente donde se muestra la visualización de los datos recogidos por los diferentes sensores.

#### 3.2.5. Java Android

Aunque Kotlin le está quitando protagonismo en los últimos tiempos a Java, sigue siendo el lenguaje principal de programación para dispositivos Android. La aplicación desarrollada para extraer los datos de los sensores de un teléfono móvil ha sido realizada en esta plataforma. Se ha utilizado este lenguaje porque se utilizará un dispositivo Android para la recogida de datos debido a que el precio de este tipo de dispositivos es inferior.

### 3.3. Software

En este apartado se describirán todos los programas y herramientas utilizadas durante el desarrollo de todo el proyecto.

#### 3.3.1. Android Studio

Android Studio es el entorno de desarrollo oficial para el desarrollo de aplicaciones para todo tipo de dispositivos de la plataforma Android. Está disponible en varias plataformas y se basa en el software IntelliJ IDEA de JetBrains [24]. Entre las características más importantes se pueden destacar la facilidad a la hora de crear una interfaz de usuario, la gran cantidad de plantillas para crear diseños comunes en Android y un emulador de un dispositivo virtual de Android configurable para ejecutar y probar las aplicaciones. En el proyecto lo he utilizado para crear una aplicación Android con el objetivo de recolectar

datos de los sensores del dispositivo y mandarlos a la base de datos. A continuación, se muestran unas imágenes del entorno de desarrollo:

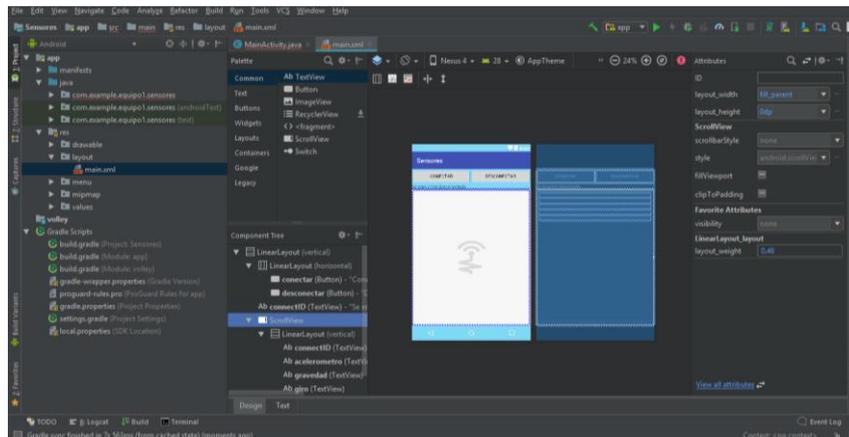


Ilustración 10 Android Studio

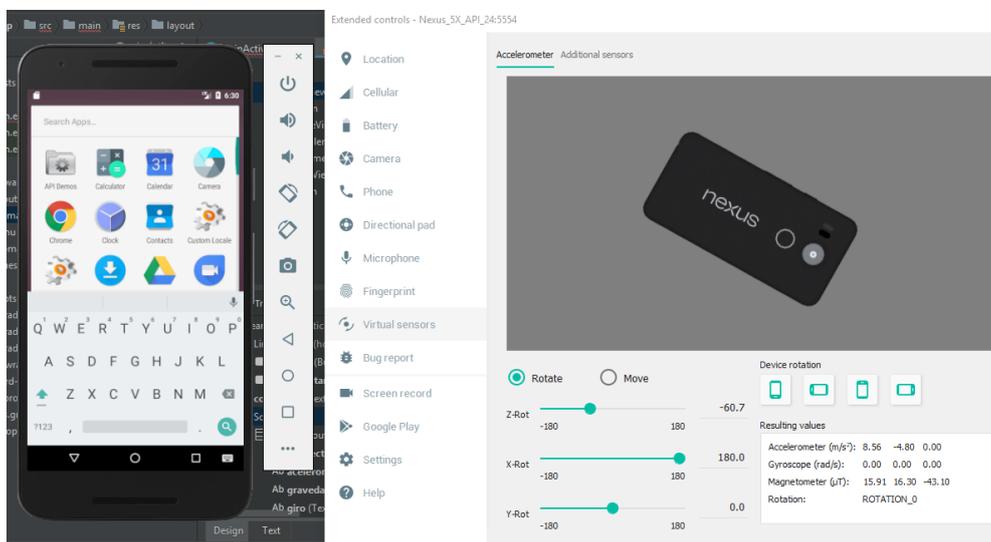


Ilustración 11 Ejemplo del emulador de Android Studio

### 3.3.2. PuTTY

Es un cliente SSH de código libre para la conexión segura con un equipo de destino [25]. Es multiplataforma y se encarga de crear una conexión mediante líneas de comandos SSH. He usado este software para gestionar las conexiones con la Raspberry y crear los scripts necesarios para la recolección y envío de datos.

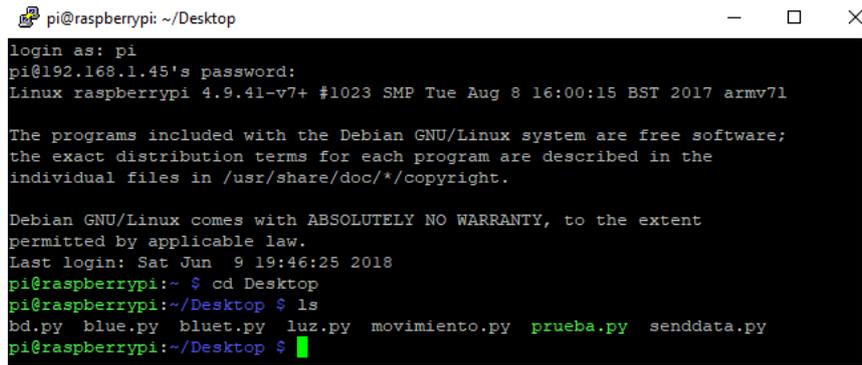


Ilustración 12 Terminal de PuTTY

### 3.3.3. FileZilla

Software libre diseñado inicialmente para funcionar en Microsoft Windows, pero desde la versión 3.0.0 gracias al uso de wxWidgets, es multiplataforma. Permite establecer conexiones con el servidor facilitando la edición de los archivos [26]. Por ello, se ha utilizado en el proyecto.

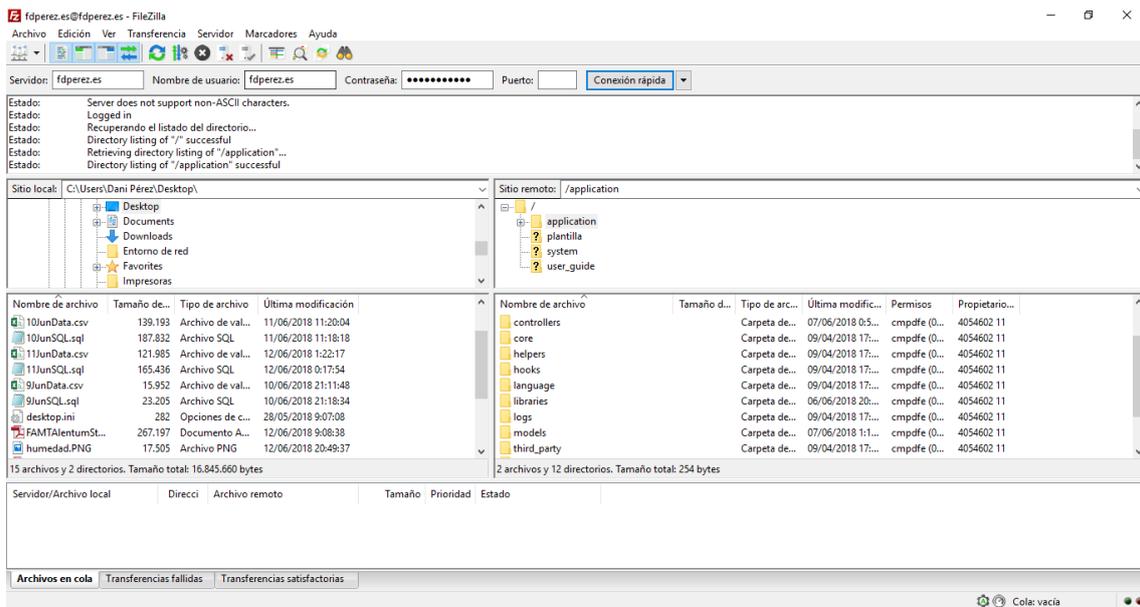


Ilustración 13 FileZilla

### 3.3.4. RESTClient

RESTClient [27] es un complemento de Mozilla Firefox que permite probar el desarrollo de una API REST mediante una interfaz sencilla e intuitiva. Me ha permitido probar la aplicación paso a paso y ahorrar mucho tiempo.

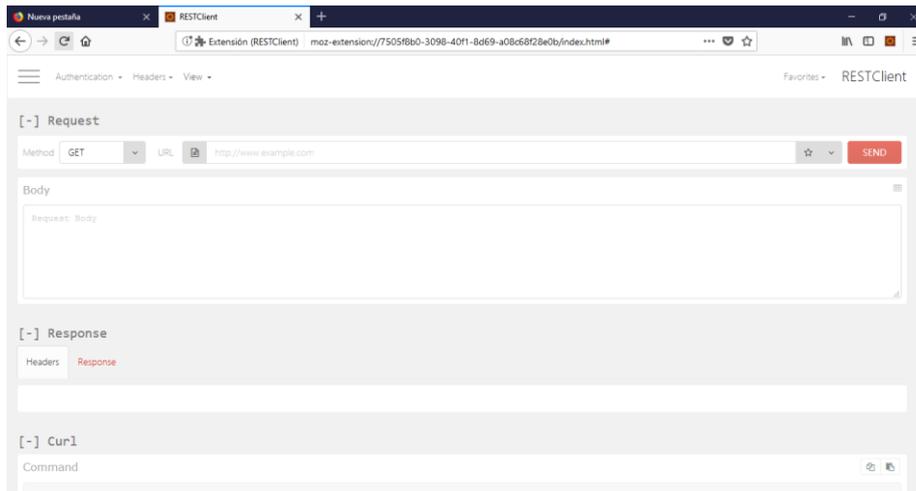


Ilustración 14 REST Client

### 3.3.5. Adobe Photoshop

Software de diseño y tratamiento de imágenes propiedad de Adobe. Es considerado el mejor software de este tipo y pone a nuestra disposición multitud de herramientas para editar cualquier tipo de imagen [28]. Lo he utilizado, principalmente, para la edición de imágenes de la memoria.

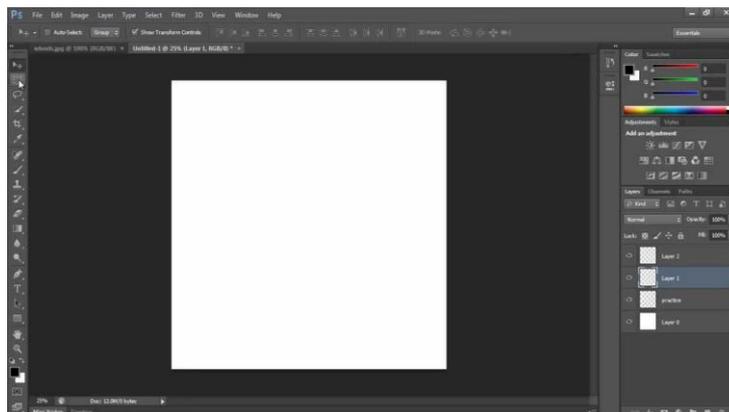


Ilustración 15 Interfaz Photoshop

### 3.3.6. Visual Paradigm

Visual paradigm [29] es un software de modelado UML que permite analizar, diseñar, codificar y probar. Permite la creación de todo tipo de diagramas UML y la generación de código fuente y de documentos a partir de los diagramas. En el proyecto ha sido utilizado a la hora de diseñar el funcionamiento del sistema.

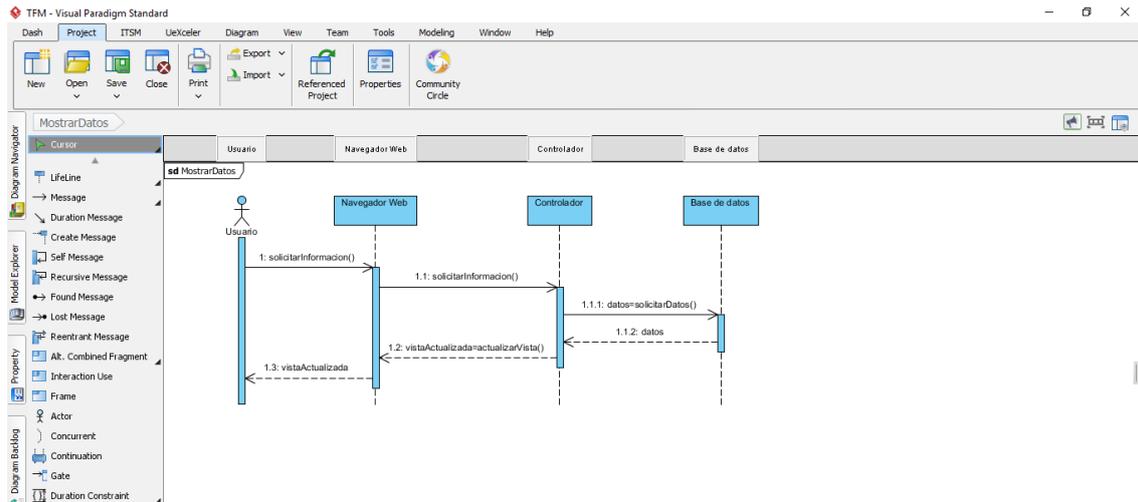


Ilustración 16 Interfaz de Visual Paradigm

### 3.3.7. KNIME

KNIME [30] es una plataforma de minería de datos con multitud de algoritmos que permite el desarrollo de modelos mediante un entorno visual. Está programado esencialmente en java y permite la instalación de complementos de otras herramientas para aumentar los algoritmos que tiene disponibles. Esta herramienta gráfica dispone de una serie de nodos, que encapsulan algoritmos, y flechas que sirven para combinar de manera gráfica los nodos. Es una herramienta de software libre. En el proyecto será utilizada para analizar los datos recogidos por los distintos dispositivos y poder extraer las reglas de asociación entre las diferentes variables de entrada.

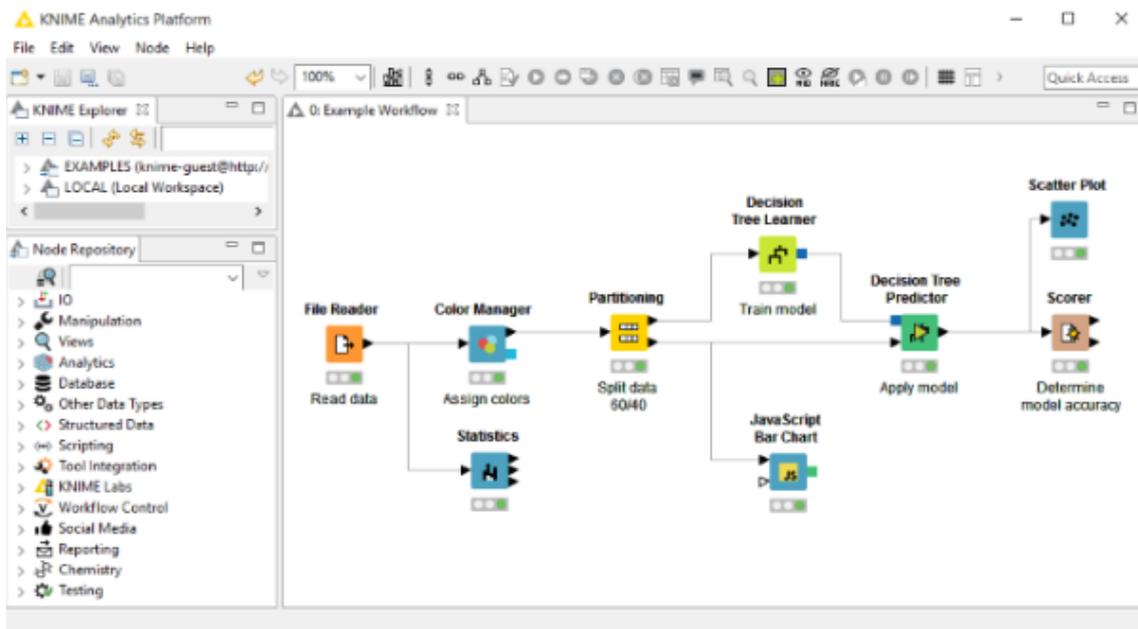


Ilustración 17 KNIME

## Capítulo 4. Ingeniería del software

En este apartado de la presente memoria se mostrarán todos los aspectos relacionados con el desarrollo del proyecto. A lo largo del capítulo se definirán los requisitos que debe cumplir, la planificación y metodología de desarrollo siguiendo un enfoque incremental genérico, diagramas para comprender como debe ser el sistema desde el punto de vista del dominio del problema, el proceso de implementación y las pruebas realizadas.

### 4.1. Análisis

En este apartado se realizará un estudio para especificar qué debe hacer el sistema desde el punto de vista del dominio del problema. Como la única funcionalidad que tiene el sistema de cara a los usuarios es la de activar/desactivar el envío de datos desde la aplicación del dispositivo Android y la de mostrar los resultados obtenidos al usuario en la aplicación web, carece de sentido la representación de los diferentes casos de uso o escenarios que se puedan dar debido a que este proyecto se centra más en el desarrollo de un sistema para la recogida y procesamiento de datos. Por lo tanto, se definirán directamente los requisitos que debe cumplir el sistema. Como en cualquier desarrollo de un producto software, es esencial reunir los requisitos básicos que debe cumplir el sistema. Para ello, se realizará un listado con todos los requisitos para comprobar, a la finalización del proyecto, que el prototipo del sistema desarrollado se ajusta a la definición inicial.

#### 4.1.1. Requisitos funcionales

Los requisitos funcionales incluyen todas las funcionalidades que el sistema debe ser capaz de realizar.

- Consultar los datos recogidos por los diferentes sensores en las últimas 24H.
- Mostrar las reglas de asociación difusas existentes actualizadas respecto a los últimos datos almacenados.
- Mostrar el estado de los sensores del dispositivo Android
- Activar y desactivar el envío de datos desde el dispositivo Android

#### 4.1.2. Requisitos no funcionales

Los requisitos no funcionales contienen todas las características que no definen una funcionalidad del sistema.

- La interfaz web seguirá una guía de estilo uniforme, utilizando una misma gama de colores, tipografía, etc.
- La interfaz web tendrá un diseño simple y sencillo para favorecer el fácil uso de los usuarios.
- La interfaz web debe presentar un diseño responsivo, es decir, la visualización de la misma se debe adaptar al tipo de dispositivo desde el que se accede.
- A la hora de mostrar los datos en la interfaz web, debe incluirse el momento en el que fue recogido dicho dato.
- La aplicación Android dispondrá de una interfaz simple para facilitar su uso.
- La aplicación Android se podrá ejecutarse en segundo plano.
- La aplicación Android realizará los envíos de los datos de forma constante y tras un periodo de tiempo determinado para evitar el alto consumo de datos.
- El programa de la Raspberry preparará los datos antes de enviarlos.
- La Raspberry Pi realizará el envío de los datos de los sensores de forma constante y tras un periodo de tiempo determinado.

## 4.2. Diseño

El diseño del proyecto se realizará, en parte, mediante modelado UML, un lenguaje de modelado utilizado para especificar los métodos y procesos de una manera visual. El objetivo de esta fase es definir cómo debe actuar el sistema. Además, también se incluirá información sobre el diseño de la interfaz y la arquitectura elegida para el sistema.

#### 4.2.1. Modelo Vista Controlador

La necesidad de una arquitectura que dé soporte a la comunicación de forma remota entre diferentes dispositivos justifica la implantación de esta arquitectura basada en servicios REST. Este tipo de arquitectura permite separar

totalmente la lógica de negocio del cliente/servidor. Además, el uso de un API REST nos ofrece una alta fiabilidad, escalabilidad y flexibilidad. CodeIgniter es un framework PHP que ayudará a minimizar el tiempo de implementación, y está basado en el Modelo Vista Controlador (MVC). La arquitectura MVC es un patrón para el desarrollo de software que se basa en separar los datos, la interfaz de usuario y la lógica interna. El objetivo de esta arquitectura es mantener al modelo lo más independiente posible de la vista y del controlador, de manera que un cambio en uno de ellos no implique la necesidad de realizar una modificación en los demás componentes, es decir, se pretende que las responsabilidades de cada uno de estos elementos estén perfectamente definidas y separadas para que el acoplamiento sea mínimo. En la siguiente ilustración se muestra el esquema de la arquitectura mencionada.

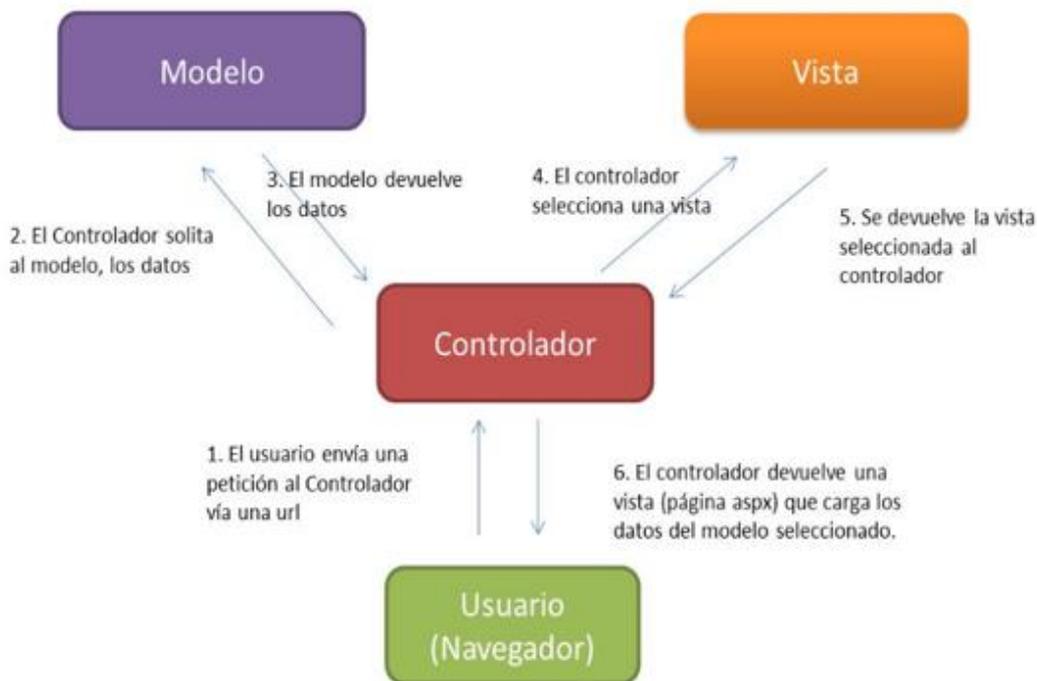


Ilustración 18 Esquema del Modelo Vista Controlador

#### 4.2.2. Diagramas de secuencia

En este apartado se representará la forma en la que debe actuar el sistema mediante modelado UML. En los siguientes diagramas se muestra la interacción existente entre los diferentes componentes que forman el sistema.

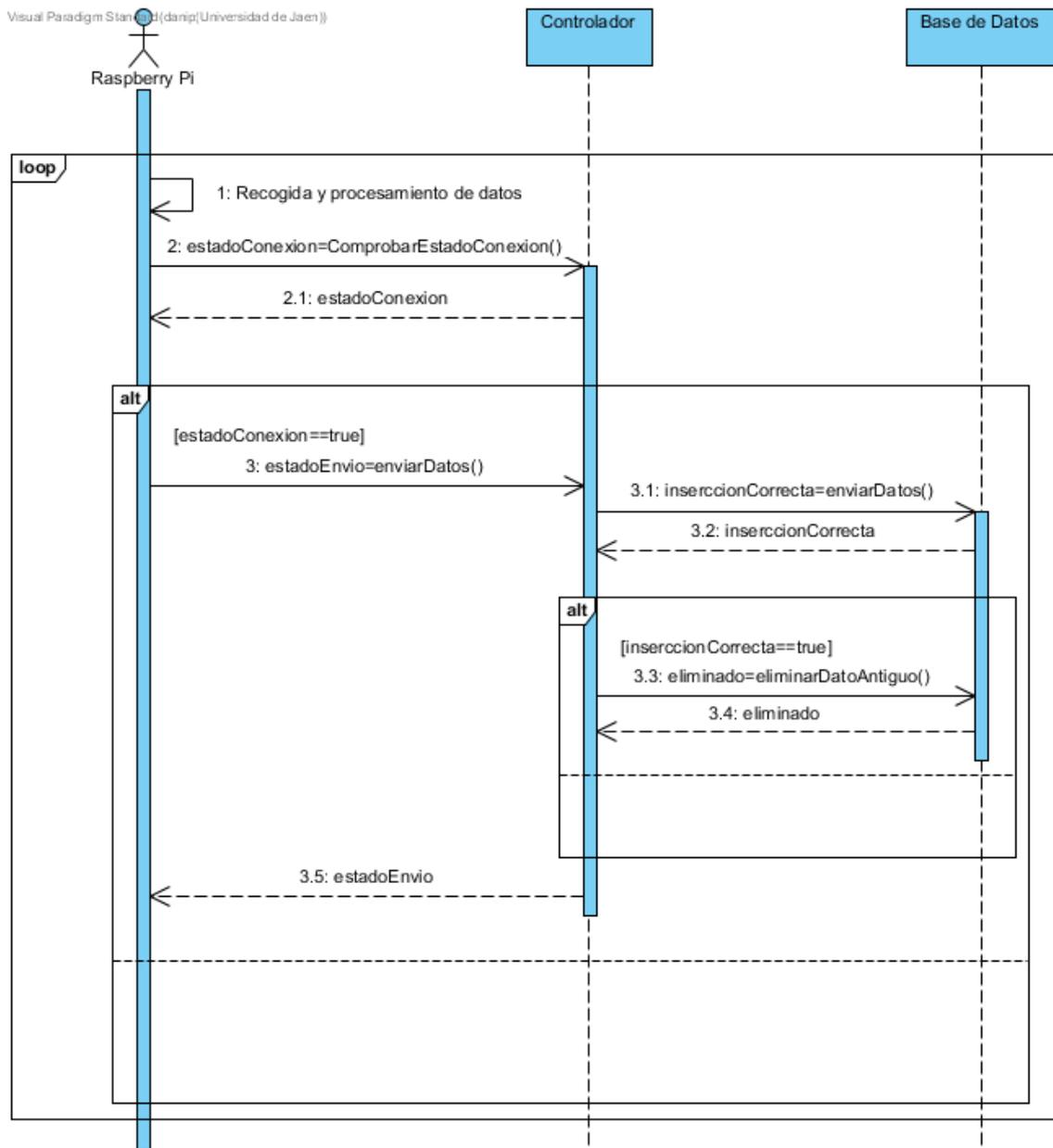


Ilustración 19 Diagrama de secuencia del envío de datos desde la Raspberry Pi

En el diagrama de secuencia anterior se muestra cómo se envían los datos recogidos desde la Raspberry Pi a la base de datos. Tras recoger los datos, se comprueba la conexión con el servidor y en el caso de que la conexión sea estable, se procede al envío de datos. Una vez enviados, el controlador se encarga de comprobar si se insertan correctamente, y en el caso de que se inserten correctamente, elimina el registro más antiguo de la base de datos.

A continuación, se muestra el diagrama de secuencia referente al envío de datos desde el dispositivo móvil. Este esquema es bastante similar al anterior ya que tras recoger los datos de los sensores comprueba el estado de la conexión y procede al envío de los mismos a la base de datos. La diferencia

reside aquí, cuando se inserta un dato en la base de datos, el sistema no elimina el más antiguo, puesto que se hace cuando se inserta un dato proveniente desde la Raspberry, ya que existe un campo que asocia los registros provenientes de los dos dispositivos como veremos en el esquema de la base de datos. Sin embargo, como ambos dispositivos están mandando datos, con una frecuencia similar, se comprobará si el último conjunto de datos proveniente desde la Raspberry tenía datos asociados del dispositivo móvil. En el caso de que no tengan ningún conjunto asociado, se asociarán.

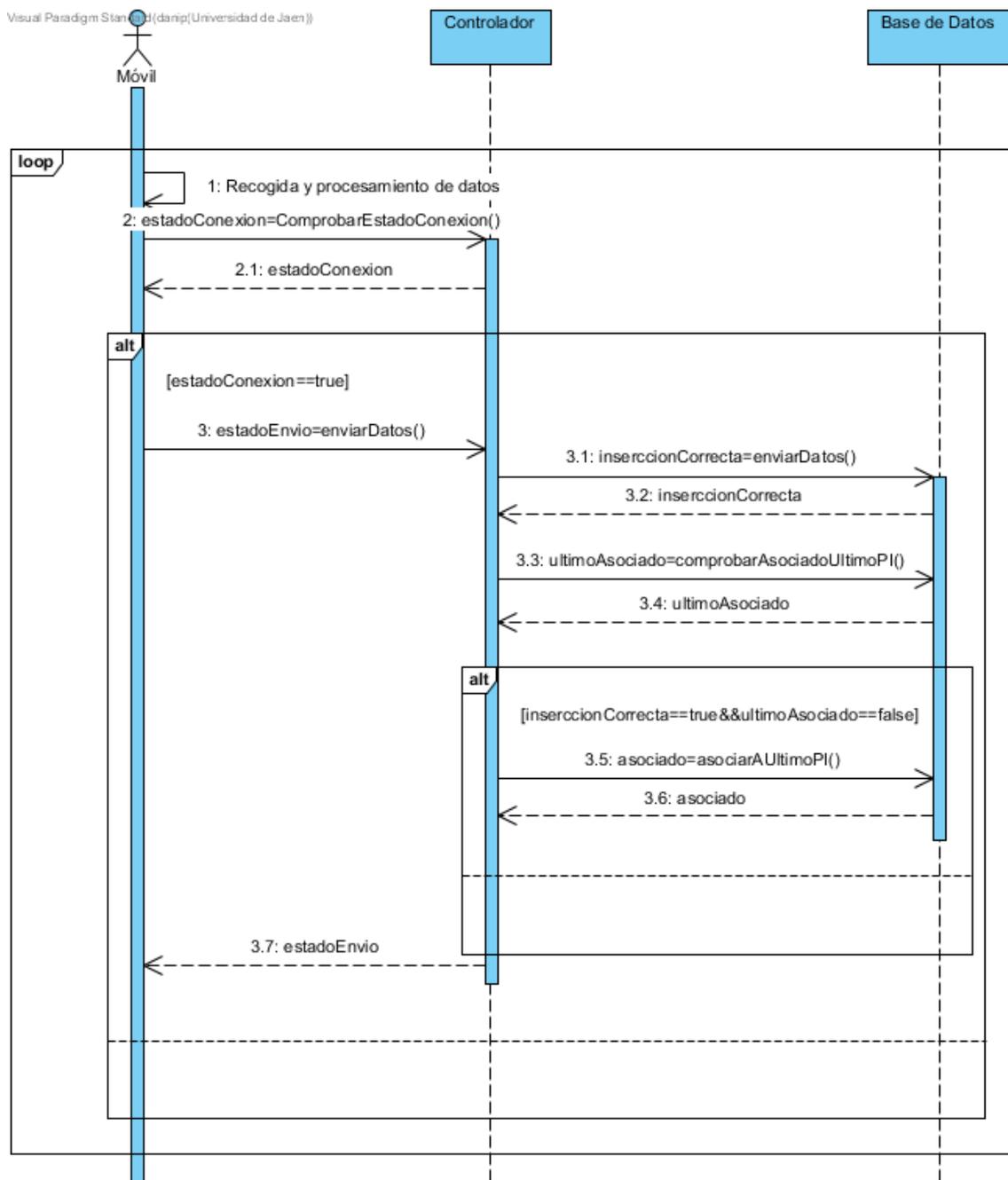


Ilustración 20 Diagrama de secuencia del envío de los datos desde el móvil

Por último, el siguiente diagrama de secuencia muestra la interacción entre el usuario y el sistema a la hora de ver los datos. El usuario accederá a la aplicación web y el controlador se encargará de acceder los datos y proporcionarle la vista con los datos correspondientes.

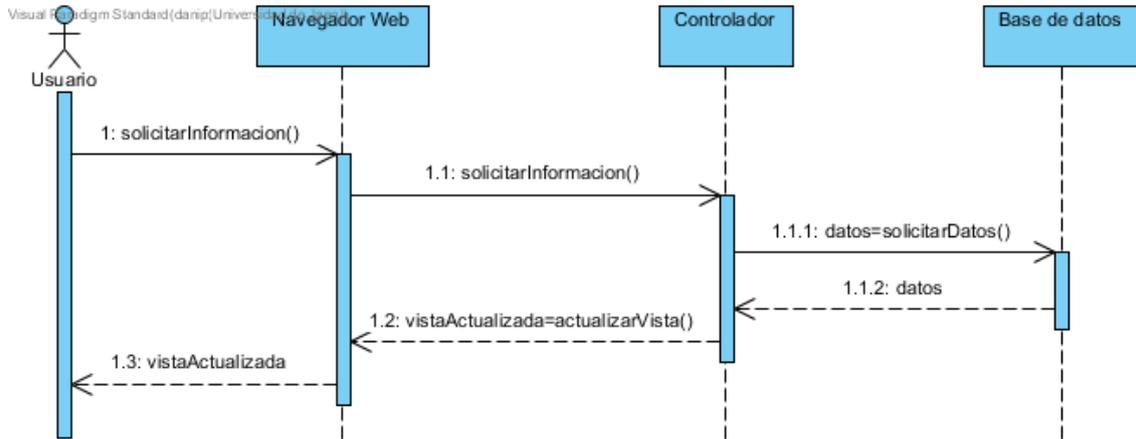


Ilustración 21 Diagrama de secuencia de la interacción entre el usuario y el sistema para ver los datos

#### 4.2.3. Diseño de la interfaz

Para decidir el diseño de la interfaz, en primer lugar, realizamos un prototipo para la aplicación del dispositivo móvil con el recurso proto.io [31].



Ilustración 22 Prototipo de la aplicación para dispositivos Android

Como podemos ver en la ilustración anterior, el prototipo de la aplicación tiene un diseño sencillo para facilitar su uso. Se han añadido el botón de conectar y desconectar para que el usuario pueda gestionar y poner en marcha o parar la captación de los datos de los sensores del dispositivo, mientras que muestra sus valores en tiempo real, y enviar los datos recogidos cada cierto tiempo. Aunque

es de vital importancia la recogida de los datos de forma continua, se ha añadido esta opción para poder pausar el envío constante de datos evitando problemas que pueda acarrear en el dispositivo móvil y un alto gasto en la tarifa de datos. En la sección de implementación se darán más detalles sobre este punto.

Por otro lado, la aplicación web ha sido diseñada mediante un Storyboard. Como la aplicación web está orientada totalmente a mostrar los datos recogidos y los resultados obtenidos tras aplicar un proceso de minería de datos al conjunto de datos, el diseño no incluirá ninguna funcionalidad para el usuario y se centra en mostrar de una forma sencilla y clara los datos obtenidos. Por lo tanto, los datos se mostrarán en una misma página. En la siguiente ilustración se muestra el diseño obtenido.

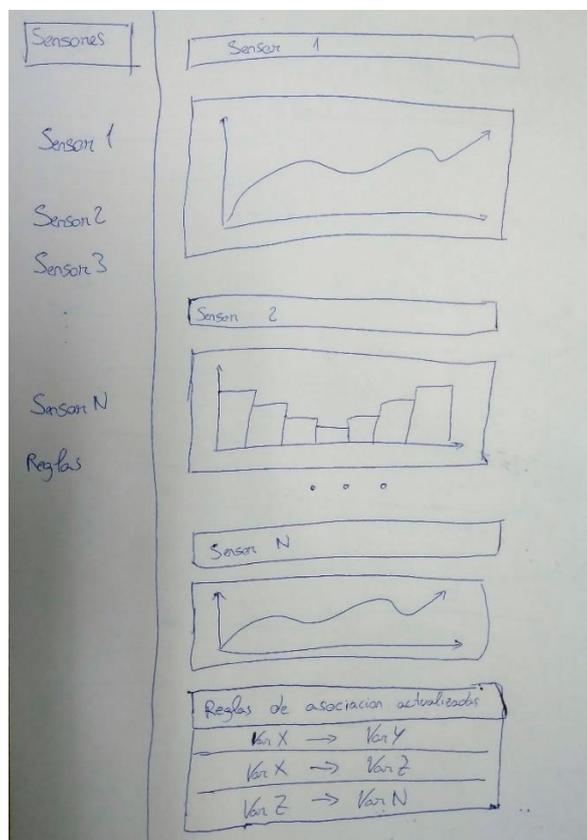


Ilustración 23 Storyboard de la interfaz de la aplicación web

#### 4.2.4. Esquema de la base de datos

Otro punto clave del sistema es diseñar el esquema de la base de datos. Se utilizará MySQL [17] como sistema de gestión de la base de datos porque es el que más se adecua para los datos que se van a almacenar. En la base de datos solo se almacenarán los datos recogidos durante las últimas 24 horas debido a que son los únicos que se mostrarán en las gráficas y se utilizarán para

actualizar las reglas de asociación difusas, aunque se realizará una copia diaria que almacenará todos los datos en una base de datos auxiliar, como un almacén de datos. En principio se iba a hacer una copia de todos los datos almacenados en un disco duro externo a las 00 horas de cada día, pero por ciertos conflictos con el servicio de alojamiento, he tenido que cambiar el plan inicial y se comentará la solución adoptada en el apartado de implementación que consiste en añadir una tabla auxiliar donde se almacenarán todos los registros. Como la cantidad de datos que se almacenarán en la base de datos no es demasiado grande, se podría utilizar prácticamente cualquier sistema de base de datos, aunque no se notaría mucha diferencia si se utilizara un sistema de base de datos no relacional como MongoDB [18] que está preparado para encargarse de realizar búsquedas de forma optimizada con grandes cantidades de datos.

Como los datos provienen de dos fuentes diferentes con una frecuencia de envío similar, se utilizarán dos tablas diferentes en las que cada una recogerá los datos provenientes de uno de los componentes y se relacionarán mediante un campo de una de ellas. También hay que tener en cuenta que se recibirán datos de forma constante desde la Raspberry Pi mientras que los datos del dispositivo móvil no serán constantes durante las 24 horas principalmente por cuestiones de eficiencia energética como se comenta en el apartado de implementación. La primera tabla contendrá los datos provenientes de la Raspberry Pi como se puede ver en la siguiente ilustración.

- **Id**
- **Hora**
- **Lum**
- **Mov**
- **Hum**
- **Pre**
- **Tem**
- **Blue**
- **Id\_mobile**

*Ilustración 24 Esquema de la base de datos (I)*

El “id” y la “hora” se generan de forma automática cuando se inserta un dato en la tabla. Los campos “lum”, “mov” y “blue” se corresponden con el sensor de luminosidad, movimiento y el bluetooth. Los datos se representan mediante un entero que representa el estado del sensor (1 o 0). “Hum”, “pre” y “tem” se corresponden con los sensores de humedad presión y temperatura. Estos datos se representan con un float. Por último, el campo “id\_mobile” hace referencia al dato de la tabla, que almacena los datos del dispositivo móvil, que se corresponde con el dato enviado por la Raspberry Pi. A continuación, se muestra el esquema de la tabla que almacena los datos del dispositivo móvil.

• Id	• Grax
• Hora	• Gray
• Girx	• Graz
• Giry	• Mag
• Girz	
• Acex	
• Acey	
• Acez	

Ilustración 25 Esquema de la base de datos (II)

Al igual que la tabla anterior, el “id” y la “hora” se generan de forma automática al insertar los datos. El controlador es el encargado de asociar el id de esta tabla con la anterior, como se muestra en el diagrama de secuencia correspondiente. Los campos que comienzan con “gir” hacen referencia a los valores del giroscopio, al igual que “ace” referencia al acelerómetro y “gra” a la gravedad. El campo “mag” se corresponde con el sensor de magnetismo del móvil. Todos los campos están representados por un float.

Por último, también existe una tabla que contiene las reglas de asociación difusas obtenidas tras aplicar un proceso de minería de datos sobre los datos almacenados en el resto de tablas. Esta tabla contiene dos columnas de tipo string que contienen el nombre de las columnas que forman los conjuntos y establecen la asociación entre ambos.

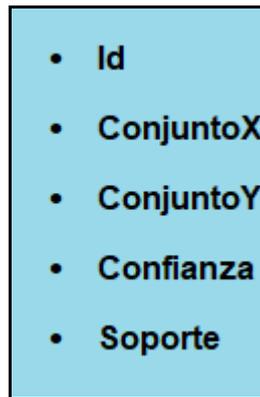


Ilustración 26 Esquema de la base de datos (III)

### 4.3. Planificación

La planificación de un proyecto, así como la estimación de las diferentes tareas son puntos críticos para el éxito de un proyecto. La planificación es difícil y la mayoría de las veces suele incumplirse, pero un buen proceso de planificación consigue que se reduzca el riesgo de fracaso del proyecto y la mejora a la hora de tomar decisiones.

#### 4.3.1. Planificación general

El proyecto se ha planificado teniendo en cuenta que el tiempo de ejecución del presente Trabajo Fin de Máster se establece en 300h. Por lo tanto, el esfuerzo establecido para cada tarea, se ve reflejado en la duración de la misma dentro de la siguiente tabla:

Fases	Duración (Horas)
Recopilación de información	20
Desarrollo en la Raspberry Pi	40
Desarrollo Android	40
Desarrollo del cliente web	60
Experimentación	40
Refinamiento	30
Análisis de los resultados	40
Extensión del sistema	30
Documentación	Tarea en paralelo con las anteriores

Tabla 1 Planificación general del TFM

En la primera fase, se tratará de obtener información sobre la minería de datos, los problemas que trata de resolver y las diferentes técnicas que podemos emplear para obtener conocimiento sobre un conjunto de datos. Las tres siguientes fases corresponden con todo el desarrollo para obtener los datos de los sensores. En estas fases, hay que tener en cuenta el estudio de los componentes que se necesitan, el diseño y la implementación en los diferentes medios. La fase de experimentación se centra en las diferentes pruebas que se llevan a cabo para probar el desarrollo realizado previamente y la integración de los mismos. La etapa de refinamiento se centra en la corrección de los errores derivados de las pruebas y las necesidades detectadas durante las pruebas. El análisis de los resultados comprende todo el proceso de estudio de los datos, desde el preprocesamiento hasta la obtención de las reglas de asociación entre las variables de estudio. En función de las reglas de asociación obtenidas, la última fase se centra en extender el sistema e instaurar un sistema para la obtención de las reglas de asociación difusas en función de los últimos datos contenidos en la base de datos del sistema. La documentación es una tarea paralela durante todo el desarrollo del proyecto. A continuación, podemos ver el diagrama de Gantt con el que se expone el tiempo de dedicación previsto para cada tarea de forma visual a lo largo del proyecto. El proyecto tendrá una duración total de 300h divididas en 15 semanas de trabajo en las que se trabajará de lunes a viernes. Cada semana se trabajará durante 20h. El proyecto comenzará el 1 de marzo y terminará el 13 de junio.

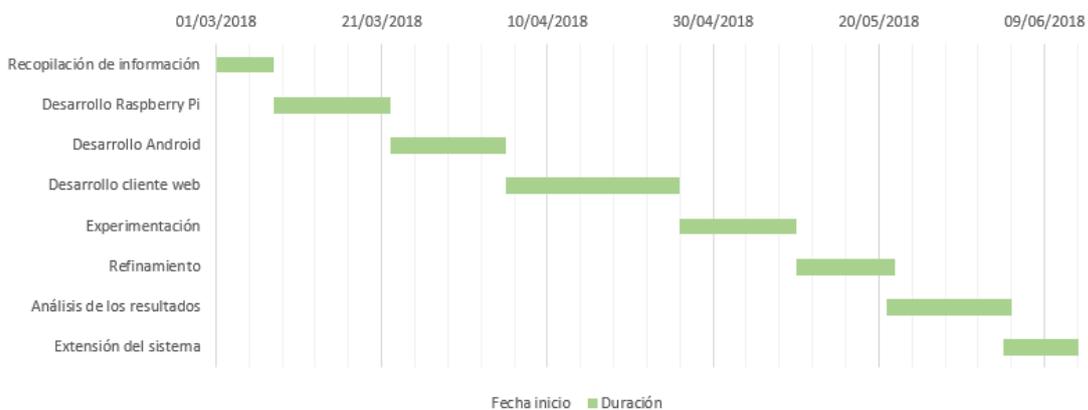


Ilustración 27 Diagrama de Gantt

#### 4.3.2. Planificación de la implementación

Considero esencial la planificación por separado de todas aquellas fases que requieren implementación ya que es de vital importancia una buena planificación para cumplir con los requisitos establecidos. Para el desarrollo del proyecto se llevará a cabo una metodología incremental genérica. Este enfoque es el más conveniente para este proyecto debido a que los requisitos no se conocen por completo al inicio del proyecto y están sujetos a cambios durante el desarrollo. Además, no tiene demasiado sentido el uso de una metodología de desarrollo ágil como Scrum ya que soy la única persona encargada de realizar las diferentes tareas y debería asumir todos los roles dentro del proyecto.

El proyecto necesitará cinco iteraciones para conseguir su estado final:

- 1) Desarrollo en la Raspberry Pi.
- 2) Desarrollo en Android.
- 3) Desarrollo del sistema web.
- 4) Integración de los diferentes desarrollos con el sistema web.
- 5) Refinamiento y extensión del sistema web.

Cada iteración tiene como resultado un prototipo funcional. A continuación, se detalla cada una de las iteraciones, aunque los detalles técnicos y las decisiones tomadas durante el desarrollo no se expondrán en la memoria hasta el apartado de implementación.

##### 4.3.2.1. Desarrollo en la Raspberry Pi (iteración 1)

Esta iteración comprende desde la definición de los requisitos iniciales, hasta el desarrollo de una aplicación que se encargue de recoger los datos de diferentes sensores. Durante esta etapa también hay que estudiar las diferentes posibilidades, tanto de microcomputadores como sensores para la recogida de datos dentro del hogar, diseñar la interconexión de la Raspberry Pi con los sensores y realizar pruebas.

##### 4.3.2.2. Desarrollo en Android (iteración 2)

En esta iteración se realizará otra aplicación desde cero para la recogida de datos desde un dispositivo Android. Como tareas más específicas se pueden destacar el diseño de la interfaz, un estudio de los sensores sobre los que se pueden obtener datos y la implementación de la aplicación.

#### 4.3.2.3. *Desarrollo del sistema web (iteración 3)*

En esta etapa se desarrollará el cliente web siguiendo un diseño responsivo. Los primeros pasos a seguir durante esta iteración será estudiar las tecnologías disponibles, diseñar la interfaz, diseñar el modelo de datos asociado y realizar pruebas.

#### 4.3.2.4. *Integración de los componentes del sistema (iteración 4)*

En esta iteración se partirá de las aplicaciones desarrolladas en las anteriores iteraciones y se desarrollarán los mecanismos necesarios para que interactúen entre sí. La comunicación entre ellas se llevará a cabo mediante un API REST. En la aplicación de la Raspberry Pi y en la de Android se harán cambios para implementar el envío de los datos a través de la API REST.

#### 4.3.2.5. *Refinamiento y extensión del sistema web (iteración 5)*

Esta iteración es la última del proyecto. Se utilizará para corregir los errores detectados en las diferentes aplicaciones. Además, se implementarán nuevas funcionalidades que permitirán ver las reglas de asociación actualizadas en función de los últimos datos registrados en la base de datos.

#### 4.3.3. *Estimación de costes*

A la hora de estimar los costes de un proyecto software es necesario tener en cuenta los costes asociados al software utilizado, los recursos materiales y el personal. El coste relativo con el software y los recursos materiales no se incluirán al completo debido a que las empresas desarrollan varios proyectos de forma simultánea. Por lo tanto, en esta estimación de costes tendré en cuenta que el coste de las licencias y recursos materiales necesarios para el desarrollo del proyecto se amortiza entre varios proyectos desarrollados por una misma empresa, exceptuando los componentes cuyo uso se para el proyecto al completo como es el caso de la Raspberry Pi o los sensores asociados.

El coste de la mayor parte del software suele abonarse de forma mensual por lo cual solo tendré en cuenta el precio durante el tiempo establecido en la planificación del proyecto (15 semanas). La licencia de Adobe Photoshop que se adapta mejor al uso para el proyecto tiene un coste de 36,29€/mes [28]. Supondré que el coste de la licencia se amortizará entre dos proyectos por lo que la cuantía total asciende hasta los 72,58€. También habría que sumarle el coste de la licencia de un mes de Visual Paradigm valorada en 6€/mes, pero solo

es necesaria durante una parte del proyecto por lo que se sumará el coste total de un mes al proyecto ya que supondré que se compra íntegramente para el proyecto. Por lo tanto, el coste total asciende a 78,58€.

El precio del servicio de alojamiento, de la empresa Strato [32], donde se almacenan los datos asciende a 28,89€ durante el primer año. He decidido utilizar el servicio anual debido a que sale más rentable que el mensual. Además, el servicio se compartirá entre varios proyectos por lo que la amortización para este proyecto se reduce hasta los 14,45€.

El resto del software y tecnologías utilizadas durante el desarrollo del proyecto son de código abierto, gratuitas o disponen de algún tipo de licencia gratuita para las pequeñas empresas como es el caso supuesto a la hora de hacer esta estimación de costes.

Al ser un sistema completo, el coste de la mayor parte de los componentes adquiridos debe incluirse por completo en la estimación de costes ya que son componentes específicos para el proyecto. En la siguiente tabla se muestran todos los componentes de este tipo que han sido adquiridos en Amazon [33]:

<b>Componente</b>	<b>Precio</b>
Raspberry Pi 3 Model B	36,26€
Sense Hat	35€
Fuente de alimentación de la Raspberry Pi	14,60€
Cabezal GPIO extra largo	6,68€
Carcasa Raspberry Pi	4,85€
GPIO Breakout board	9,99€
HC-SR501	4,99€
LM393	7,59€
Cables	6,89€

*Tabla 2 Coste de los componentes específicos del proyecto*

Como podemos ver en la tabla superior, el coste total de los componentes asciende a 126,85€. También es cierto que venían varios sensores en algunos packs y que el coste también podría ser inferior si en vez de utilizar el módulo Sense Hat se hubiesen utilizado los sensores por separado, aunque decidí comprar este módulo ya que tengo intención de utilizarlo en un proyecto futuro.

Además de estos componentes también se necesita un dispositivo móvil. El coste en Amazon [33] de un dispositivo de gama media, como el Motorola Moto G5, asciende a unos 150€ aproximadamente. Al igual que pasará con los equipos informáticos, este dispositivo puede ser reutilizado en otros proyectos de la empresa. Como la vida útil de un teléfono inteligente está establecida en unos 20 meses según un estudio realizado por El Androide Libre [34], uno de los blogs de habla hispana más famosos sobre Android, supondré que en los cuatro meses que dura el proyecto se debe amortizar un gasto de 30€. Con los equipos informáticos pasa algo similar, aunque la duración media de estos varía desde los 5 hasta los 10 años [35]. Supondré como vida útil del equipo informático de desarrollo estos 5 años de duración media. Por lo tanto, suponiendo que el coste total del equipo asciende hasta los 726,88€, como es el caso del ASUS GL553VD en Amazon [33], la cantidad total para amortizar durante los cuatro meses de trabajo es de 48,45€.

A continuación, se muestra una tabla donde se recogen todos los gastos acumulados hasta el momento:

Componente	Precio
Licencias software	78,58€
Servicio de alojamiento	14,45€
Raspberry Pi + Componentes asociados	126,85€.
Dispositivo móvil	30€
Equipo de desarrollo	48,45€

Tabla 3 Resumen de todos los componentes y su precio

Además de estos costes, también hay que estimar los costes de personal. Como todo el trabajo recae sobre mí, asumiré todos los roles durante el desarrollo del proyecto, aunque haré distinción de salario entre los diferentes roles que intervienen en el proyecto. Los salarios establecidos han sido extraídos del convenio colectivo estatal de 2009 [36] , ya que es el último disponible.

Dentro del proyecto se necesitarán los siguientes roles:

- Analista: se encargará del análisis y diseño del sistema.
- Programador: se encargará de la implementación del sistema.
- Administrador de test: se encargará de las pruebas de usuario.

- Jefe del proyecto: se encargará de la supervisión y documentación del proyecto.

Como la empresa trabaja en varios proyectos a la vez supondré que el coste asociado a este proyecto de cada trabajador es del 25% debido a su desempeño dentro del mismo, excepto el programador que se dedicará exclusivamente al proyecto. Además, hay que considerar los costes que implica tener un trabajador en nómina lo cual hace que se deba aplicar un 33% de impuestos a la empresa. En la siguiente tabla se detallan los costes totales en recursos humanos.

Rol	Sueldo	Retención	Duración	Dedicación	Total
Analista	1.539,69€	33%	3 meses	25%	1.535,84€
Programador	1.103,04€	33%	3 meses	100%	4.402,56€
Administrador de test	1.104,04€	33%	2 meses	25%	734,18€
Jefe de proyecto	1.167,29€	33%	4 meses	25%	1.552,49€
<b>Coste Total</b>					<b>8.225,07€</b>

Tabla 4 Recursos humanos

Por último, también es necesario tener en cuenta los costes indirectos del proyecto. Además de los gastos cotidianos como la luz o el agua, también hay que tener en cuenta los posibles imprevistos que puedan suceder al o largo del proyecto por lo que se establece un 25% sobre el coste del proyecto. Teniendo en cuenta todos los aspectos que hemos mencionado a lo largo de este apartado se puede concluir que el presupuesto del proyecto asciende a 10.646,75€ para cubrir con los gastos del proyecto al completo, como indica la tabla 5.

Recurso	Precio
Software	93,03€
Materiales y equipos	205,30€
Personal	8.225,07€
Costes indirectos	2.130,85€
Coste total	10.654,25€

Tabla 5 Resumen de todos los costes del proyecto

## 4.4. Implementación

El sistema implementado se compone de tres partes destacadas: el desarrollo en la Raspberry Pi, la aplicación móvil y la aplicación web.

Las conexiones con la Raspberry Pi, para la implementación del programa encargado de la recogida y envío de los datos de los sensores, han sido principalmente mediante PuTTY, un cliente SSH. El lenguaje de programación elegido para la implementación del programa encargado es Python3 debido a que es un lenguaje fácil de aprender y a que es uno de los lenguajes más utilizados actualmente. El programa se lanza cada vez que se enciende la Raspberry Pi. Para desarrollarlo, previamente he desarrollado un pequeño programa que se encargará de leer los datos de los sensores de forma individual y realizar pruebas con los mismos. Posteriormente he creado un único programa encargado de realizar la lectura de todos los sensores y el envío de los datos a la API REST. El pseudocódigo del programa desarrollado es el siguiente:

```
InicializarDatos();
While True:
    AlmacenarDatosSensores();
    ComprobarConexion();
    If(conexion):
        EnviarDatos();
    Else:
        MostrarErrorConsola();
    Wait(47s)
```

En una primera fase se inicializan los datos. Una vez están inicializados se entra en un bucle continuo en el que se recogen los datos de los sensores, se comprueba el estado de la conexión y se envían los datos si está disponible. Posteriormente se esperan 47 segundos antes de realizar el siguiente envío. El tiempo entre envío y envío difiere sensiblemente del tiempo de envío desde el dispositivo Android que está establecido en 1 minuto. Aunque se ha establecido un tiempo menor debido a que la Raspberry Pi invierte cierto tiempo cuando se escanean los dispositivos bluetooth cercanos por lo que se compensa parte de la diferencia. La extracción de los datos contiene algunas particularidades para ciertos sensores. Para conocer si el sujeto de estudio se encuentra dentro de la vivienda se realiza una búsqueda de dispositivos cercanos para comprobar si la

dirección MAC del dispositivo se encuentra dentro de rango. Para obtener estos datos es necesario que el programa haga una llamada a la función “hcitool scan” en el terminal y se estudie la salida de dicha función. En el caso que el sistema encuentre el dispositivo cerca enviará un 1 a la base de datos mientras que si no lo encuentra mandará un 0. La sensibilidad del sensor de movimiento también ha sido ajustada para evitar falsos positivos. Este sensor utiliza un rayo infrarrojo para detectar el movimiento y manda a la base de datos un 1 si detecta movimiento y un 0 si no lo detecta.

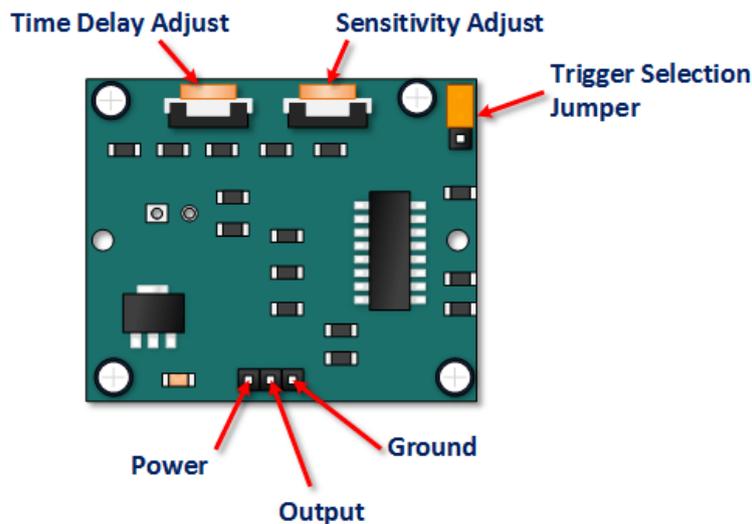


Ilustración 28 Pines de salida y ajustes del sensor de movimiento

Por su parte, el sensor de luminosidad también ha sido ajustado para evitar que detecte ciertos brillos y dé falsos positivos y negativos. En el caso de detectar iluminación mandará un 1 a la base de datos o en su defecto un 0. Al utilizar este sensor me he dado cuenta de que tiene ciertos problemas con la luz no ambiental, es decir, con las bombillas que emiten una luz blanca y no la amarilla clásica. Debe estar muy bien posicionado el sensor para detectar este tipo de luces.

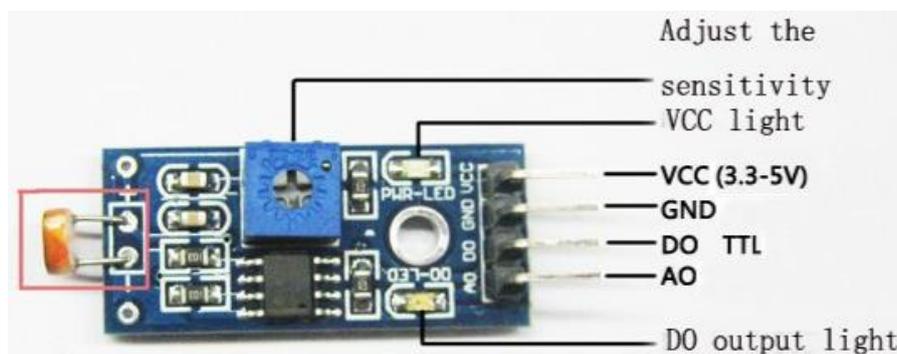


Ilustración 29 Esquema de los pines del sensor de luminosidad

Los datos del sensor de presión, humedad y temperatura se obtienen directamente del módulo Sense Hat. Este módulo incorpora una zona que se corresponde con los pines de la Raspberry y solo hay que conectarla con los pines.

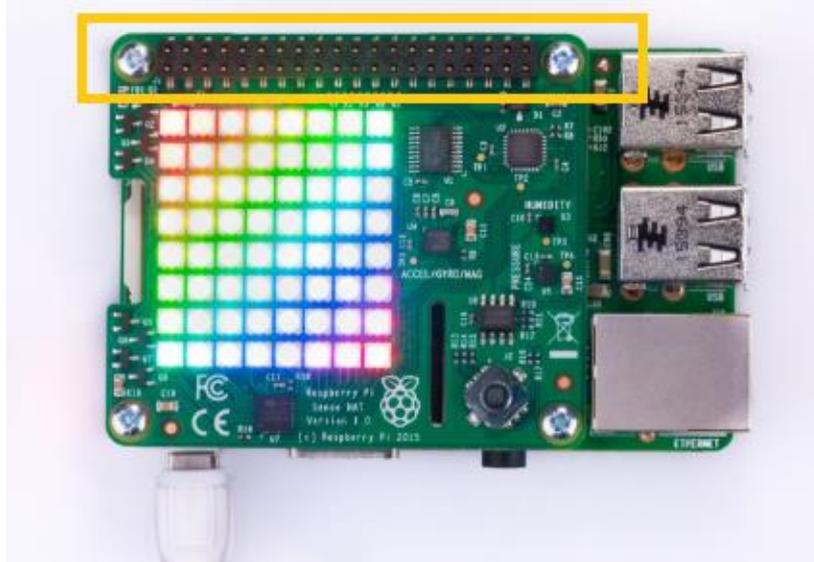


Ilustración 30 Sense Hat. La zona marcada indica la zona de conexión con la Raspberry Pi

Dispone de funciones para extraer estos datos, aunque los valores del sensor de temperatura son erróneos debido a que muestra valores elevados por el calor que genera la CPU y más si está colocado encima. Para que funcione correctamente el sensor de temperatura habría que alejar el módulo de la Raspberry Pi, provocando la pérdida de un importante beneficio como la compactación entre el módulo y el dispositivo. La otra opción es la de tratar de calibrar las lecturas del sensor de temperatura. Esta opción ha sido la que he elegido y he partido de una solución propuesta por un usuario de la comunidad de GitHub [37]. La ecuación propuesta tiene en cuenta la temperatura de la CPU y un factor de escala para calcular la temperatura.

```
temp_calibrated = temp - ((cpu_temp - temp)/FACTOR)
```

Como es posible obtener la temperatura mediante diferentes funciones asociadas al módulo Sense Hat, he utilizado la temperatura media que devuelven como temperatura base medida. El factor utilizado en la formula lo he obtenido a base de realizar pruebas y comparar con las medidas de temperaturas que marca otro dispositivo obteniendo como resultado que el factor es de 1,5. Además como la lectura de la temperatura de la CPU no es muy estable, he

calculado la tendencia media de la lectura de la temperatura obtenida en las últimas mediciones.

Tras la extracción se comprueba que existe conexión entre el dispositivo y el servidor haciendo un ping a la dirección donde se encuentra alojado, <http://www.fdperez.es> . En el caso de que la conexión exista se envían los datos al servidor tras procesarlos y redondear los valores obtenidos al segundo decimal. En las siguientes ilustraciones se muestra el resultado obtenido.



Ilustración 31 Vista superior de la Raspberry Pi

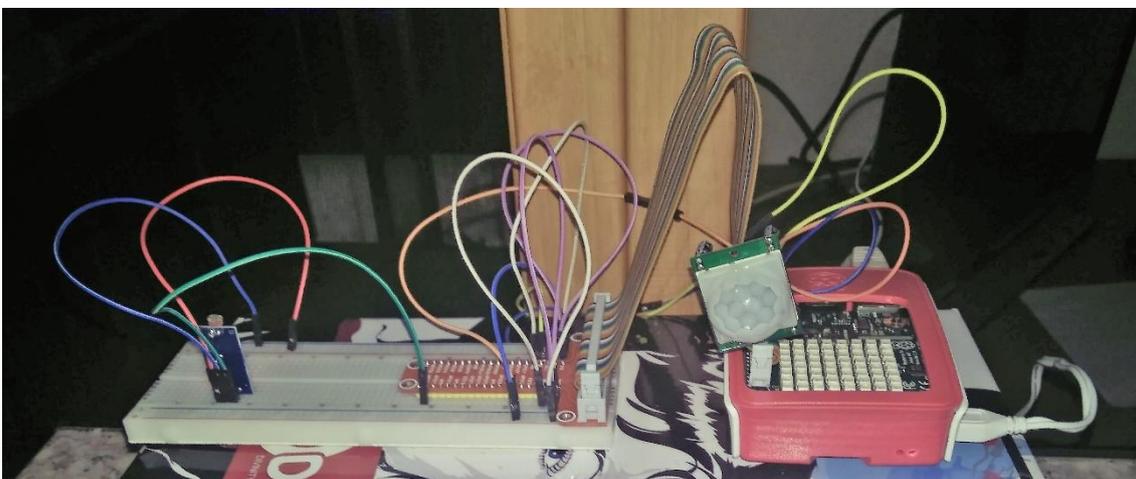


Ilustración 32 Vista lateral de la Raspberry Pi

La aplicación Android ha sido desarrollada en Android Studio utilizando Java Android. Android pone a nuestra disposición un objeto llamado “SensorManager” [38] que permite acceder de forma sencilla a los sensores del dispositivo. Es importante asegurarnos de que los sensores se puedan activar y desactivar para evitar el alto consumo de batería y datos móviles. Por ello, desde la fase de diseño se incluyeron dos botones para permitir la activación y

desactivación. El objeto dispone de la función “getDefaultSensor(param)” que recibe como parámetro el identificador del sensor sobre el que queremos obtener los datos. Una vez obtenidos los datos de los sensores de gravedad, giroscopio, acelerómetro y magnetismo, cuyos datos se representan con valores decimales y en cada uno de los ejes excepto el de magnetismo que devuelve un solo valor, se procede al envío de los datos. Es importante conceder a la aplicación todos aquellos permisos que se necesitan para el correcto funcionamiento de la aplicación. Antes de realizar el envío se comprueba la conexión con el servidor y posteriormente se realiza el envío utilizando la librería Volley [39]. Volley es una librería fácil de utilizar, rápida y de las más importantes. Permite hacer peticiones automáticas en la red, conexiones simultaneas, almacenamiento en memoria caché, herramientas de depuración y rastreo, etc. A continuación, se muestra una ilustración con el resultado obtenido tras implementar la aplicación.

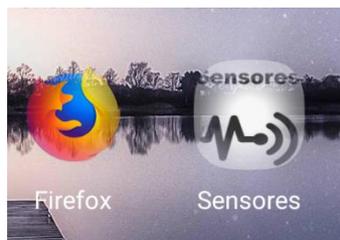


Ilustración 33 Logo de la aplicación Android



Ilustración 34 Interfaz de la aplicación Android

La aplicación web tiene como base una arquitectura Modelo Vista Controlador basada en servicios REST para favorecer la comunicación entre los diferentes dispositivos que forman el sistema. He utilizado CodeIgniter, un framework PHP basado en esta arquitectura, para minimizar el tiempo de implementación. Al seguir esta arquitectura podemos distinguir tres capas independientes en la aplicación:

- La capa del modelo es la responsable de la lógica de negocio de la aplicación. Es la encargada de acceder a la capa de almacenamiento. Cada tabla de la base de datos es un objeto que crea una capa de abstracción lo que asegura la integridad y seguridad sobre los datos almacenados.
- Las vistas forman la capa de la aplicación que ve el usuario cuando interactúa con la aplicación web.
- Los controladores se encargan de manejar las funciones que pueden llevarse a cabo en el sistema.

El modelo de esta aplicación se compone de funciones para insertar, modificar y eliminar datos en las tablas. Adicionalmente también se añadió una función para poder resetear la base de datos al completo. Esta función se diseñó de forma más específica para resetear la base de datos tras finalizar el periodo de pruebas. También se han incluido una serie de consultas específicas para estudiar la confianza que hay en las diferentes reglas de asociación obtenidas tras estudiar el conjunto de datos.

La aplicación se compone de un controlador y un controlador de recursos, es decir, un controlador de tipo REST. El primero se encarga de hacer una llamada al segundo para recoger los datos y mandarlos a la vista para que los muestre. El segundo se encarga de definir diferentes rutas indicando los métodos asociados (get, post, put y delete). Las rutas están asociadas a funciones que se encargan de trabajar con los datos a través del modelo definido. Este controlador se encargará de gestionar las peticiones y devolver los datos solicitados en formato JSON. Existe una ruta declarada para cada consulta definida en el modelo, aunque algunas no se utilizan ya que no son necesarias. Las más importantes se encargan de consultar todos los datos almacenados y de la inserción de los datos. La función de inserción de los datos provenientes del

dispositivo móvil, se encarga también de consultar si el último dato insertado en la tabla que contiene los datos de la Raspberry Pi tiene asociado algún dato del dispositivo móvil. En el caso de no tenerlo se lo asigna, Como ambos dispositivos realizan envíos con un intervalo de diferencia de aproximadamente un minuto, cuando se hace una asociación es porque se han mandado en instantes de tiempo similares. La función que se encarga de insertar los datos provenientes de la Raspberry Pi se encarga de eliminar el dato más antiguo que se encuentra almacenado en la base de datos, puesto que solo es necesario tener los de las últimas 24 horas que son los que se van a utilizar para mostrar las reglas de asociación difusas actualizadas, y de los datos recibidos del dispositivo móvil asociado si lo está. Llegados a este punto quiero comentar que inicialmente la idea era exportar periódicamente los datos contenidos en la base de datos en un disco duro externo mediante un script lanzado por la Raspberry Pi, pero el servicio de alojamiento contratado no me permite acceder a la base de datos desde fuera del servidor. Por lo tanto, el controlador se encarga a las 00 horas de hacer una copia de los datos en una tabla auxiliar donde se relacionan los campos de ambas tablas, es decir, la tabla se compone de los datos resultantes de hacer una unión(join) entre las tablas. Los datos de esta tabla deben ser exportados de forma manual para trabajar con ellos en la fase de análisis. Por último, el controlador también se encarga de comprobar qué reglas de asociación difusas, de las almacenadas en la base de datos, se cumplen con una confianza superior al 90% con los datos almacenados durante las últimas 24 horas y un soporte mínimo del 30%.

Los datos almacenados durante las últimas 24 horas en la base de datos y las reglas de asociación difusas que superan una confianza del 90% y un soporte mínimo del 30% teniendo en cuenta solo los datos de las últimas 24 horas se mostrarán a través de la vista.

```

<section id="aso">
<div class="panel-group">
<div class="panel panel-primary">
<div class="panel-heading"><b>REGLAS DE ASOCIACION</b></div>
<div class="panel-body">
<?php
$strHtml="<table border='2' solid>";
$strHtml.="<th>Conjunto X</th>";
$strHtml.="<th>Conjunto Y</th>";
foreach ($aso as $data) {
    $strHtml.= "<tr>";
    if($data->conjuntoX=="NLUZ"){
        $strHtml.= "<td>No hay luz</td>";
    }else{
        if($data->conjuntoX=="REPOSO"){
            $strHtml.= "<td>El móvil está en reposo</td>";
        }else{
            if($data->conjuntoX=="HUMBAJO"){
                $strHtml.= "<td>La humedad es baja</td>";
            }else{
                if($data->conjuntoX=="PREBAJO"){
                    $strHtml.= "<td>La presión es baja</td>";
                }else{
                    $strHtml.= "<td>El usuario no está en la vivienda</td>";
                }
            }
        }
    }
}
if($data->conjuntoY=="NLUZ"){
    $strHtml.= "<td>No hay luz</td>";
}else{
    $strHtml.= "<td>No se detecta movimiento en la vivienda</td>";
}
$strHtml.= "</tr>";
}
$strHtml.="</table>";
echo $strHtml;
?>

```

Ilustración 35 Código para la generación de la tabla de forma dinámica

Como podemos ver en el código de la imagen superior, se definen diferentes frases, según las reglas obtenidas en el apartado de análisis, para generar la tabla de forma dinámica y sustituir los valores que se almacenan dentro de ella.

Para la vista he utilizado una plantilla de Bootstrap [23] con un diseño responsivo. A continuación, vamos a ver un ejemplo de cada gráfico que se muestra en la vista de la aplicación.

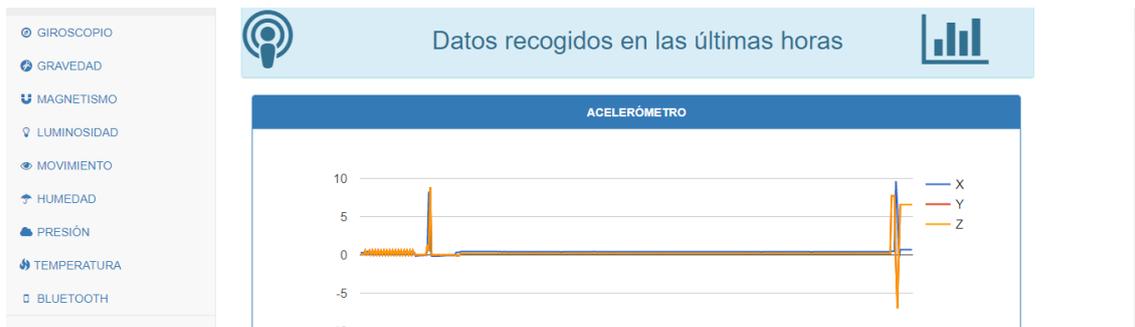


Ilustración 36 Gráfico acelerómetro

En el gráfico superior podemos ver los datos que hacen referencia al acelerómetro en cada uno de los ejes. Los valores se expresan en  $m/s^2$ .

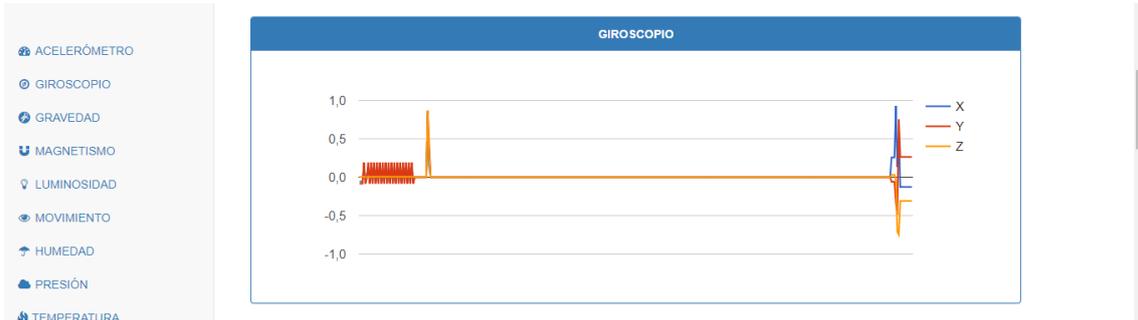


Ilustración 37 Gráfico giroscopio

En este gráfico se muestran los datos del giroscopio sobre cada uno de los ejes. Muestra la rotación sobre cada uno de los ejes. Se expresa en rad/s.

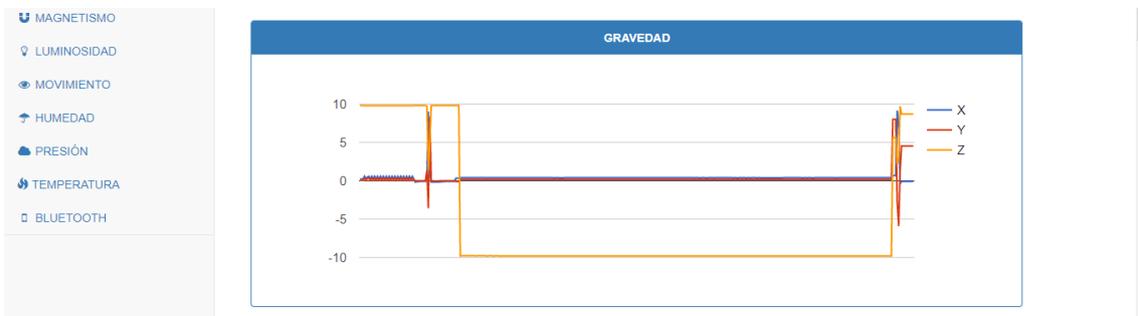


Ilustración 38 Gráfico gravedad

La ilustración superior muestra los datos referentes a la gravedad sobre cada uno de los ejes. Se expresa en  $m/s^2$

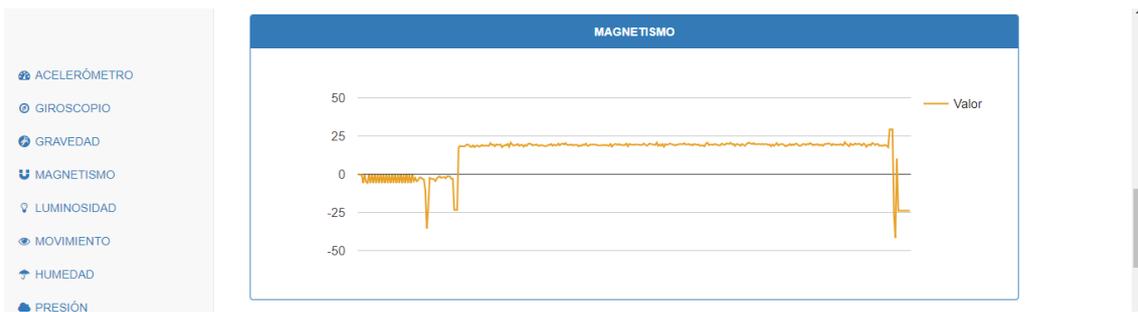


Ilustración 39 Gráfico magnetismo

El gráfico de la parte superior muestra el magnetismo medido por el dispositivo móvil. Los datos obtenidos se miden en  $\mu T$ .

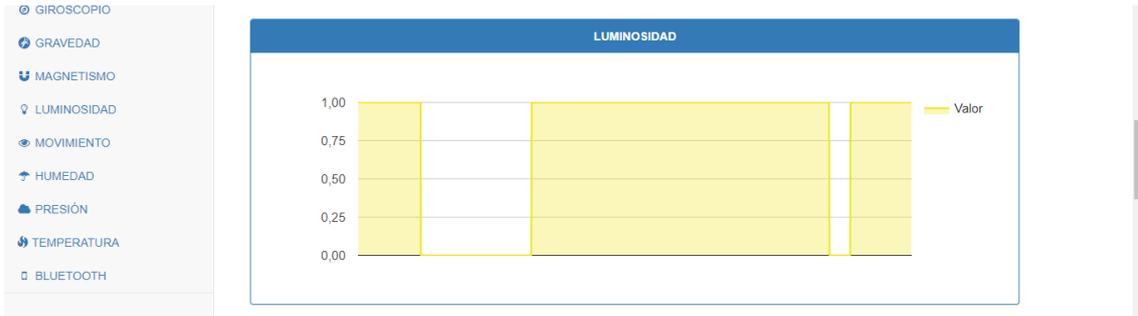


Ilustración 40 Gráfico luminosidad

La ilustración superior muestra el estado del sensor de luminosidad, 1 si el sensor detecta un brillo sobre el fotosensor y 0 si no lo detecta. Por su parte la siguiente gráfica muestra de la misma forma los datos del sensor de movimiento.

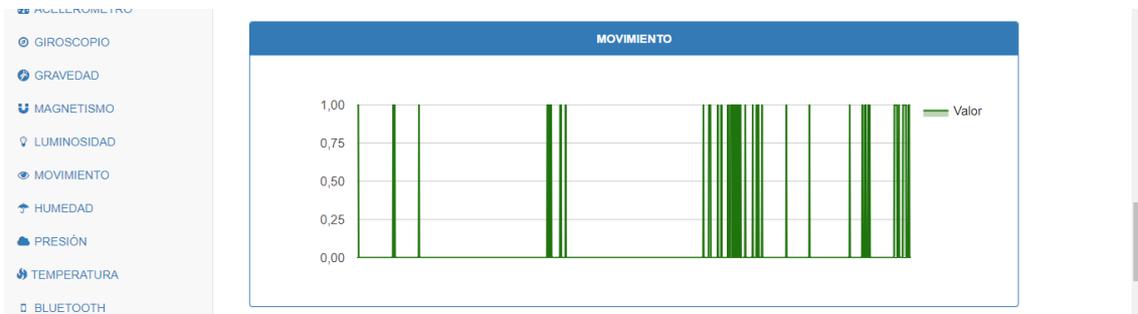


Ilustración 41 Gráfico movimiento

La siguiente ilustración muestra los datos relativos a la humedad en porcentaje. Como se puede ver en la gráfica los datos que lee el sensor son un poco inestables.

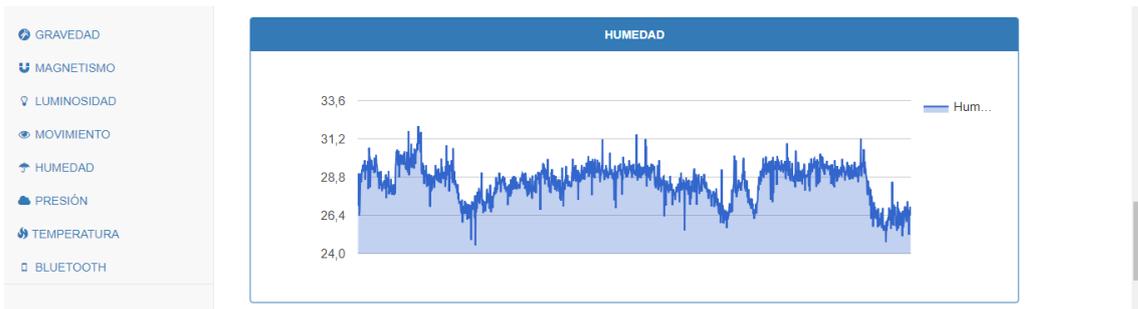


Ilustración 42 Gráfico humedad

A continuación, se muestra la gráfica del módulo de presión. Estos datos se miden en milibares.

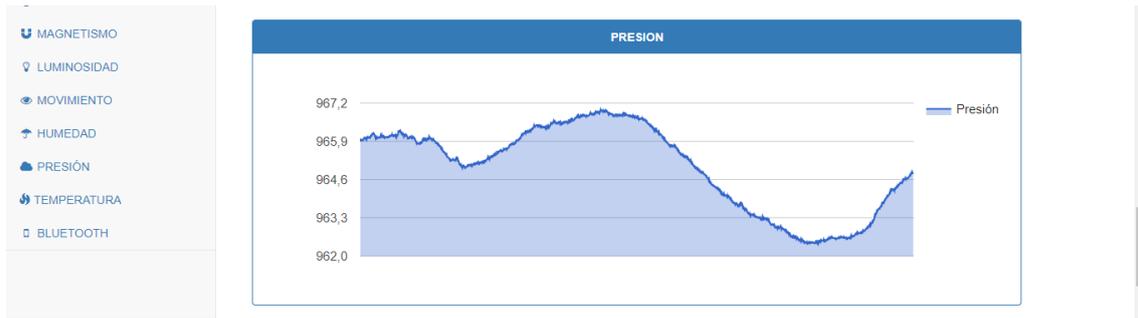


Ilustración 43 Gráfico presión

La siguiente gráfica es la de temperatura. Se ha utilizado los grados centígrados como unidad de medida. Como podemos ver, al igual que pasa en la gráfica de humedad, los datos son un poco inestables pese a que en la implementación he tratado de corregir un poco estos valores estudiando la tendencia.

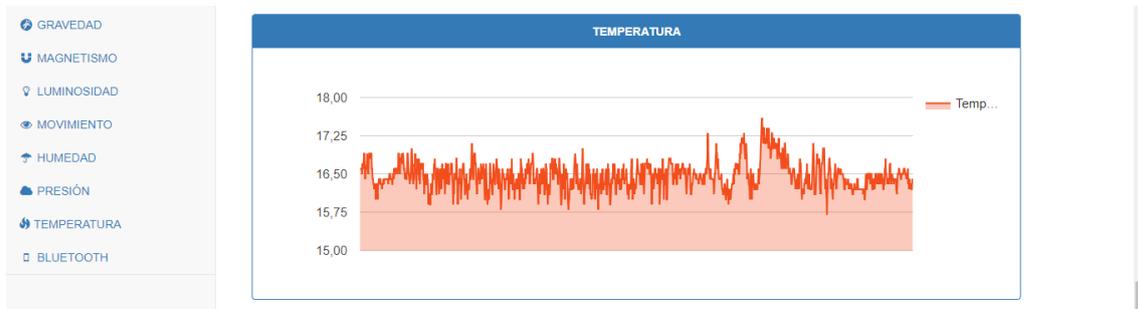


Ilustración 44 Gráfico temperatura

A continuación, se muestra la gráfica que se corresponde con el sensor de bluetooth. Indica si el dispositivo móvil se encuentra en el rango bluetooth de la Raspberry Pi indicando si el usuario se encuentra en casa (1) o si está fuera (0),

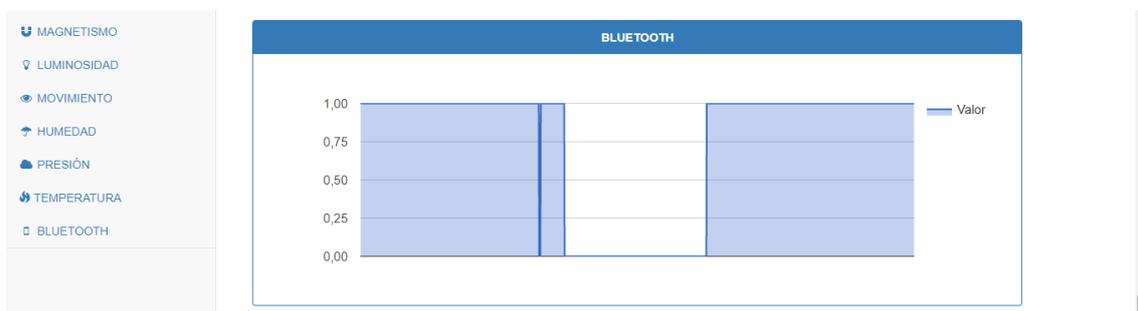


Ilustración 45 Gráfico bluetooth

A continuación, se muestran varias ilustraciones con las reglas de asociación actualizadas. La tabla se genera de forma dinámica en función de los datos que devuelve el servidor tras comprobar las reglas que tienen una

confianza superior al 90% y un soporte superior al 30% teniendo en cuenta los datos de las últimas 24 horas.

<ul style="list-style-type: none"> <li>PRESIÓN</li> <li>TEMPERATURA</li> <li>BLUETOOTH</li> <li>REGLAS DE ASOCIACIÓN</li> </ul>	<table border="1"> <thead> <tr> <th colspan="2">REGLAS DE ASOCIACION</th> </tr> <tr> <th>Conjunto X</th> <th>Conjunto Y</th> </tr> </thead> <tbody> <tr> <td>No hay luz</td> <td>No se detecta movimiento en la vivienda</td> </tr> <tr> <td>El móvil está en reposo</td> <td>No hay luz</td> </tr> <tr> <td>La humedad es baja</td> <td>No se detecta movimiento en la vivienda</td> </tr> <tr> <td>La presión es baja</td> <td>No se detecta movimiento en la vivienda</td> </tr> <tr> <td>El usuario no está en la vivienda</td> <td>No se detecta movimiento en la vivienda</td> </tr> </tbody> </table>	REGLAS DE ASOCIACION		Conjunto X	Conjunto Y	No hay luz	No se detecta movimiento en la vivienda	El móvil está en reposo	No hay luz	La humedad es baja	No se detecta movimiento en la vivienda	La presión es baja	No se detecta movimiento en la vivienda	El usuario no está en la vivienda	No se detecta movimiento en la vivienda
REGLAS DE ASOCIACION															
Conjunto X	Conjunto Y														
No hay luz	No se detecta movimiento en la vivienda														
El móvil está en reposo	No hay luz														
La humedad es baja	No se detecta movimiento en la vivienda														
La presión es baja	No se detecta movimiento en la vivienda														
El usuario no está en la vivienda	No se detecta movimiento en la vivienda														

Ilustración 46 Reglas de asociación actualizadas

id	conjuntoX	conjuntoY	confianza	soporte
1	NLUZ	NDETMOV	1.000	0.004
2	REPOSO	NLUZ	0.990	0.320
3	HUMBAJO	NDETMOV	0.986	0.900
4	PREBAJO	NDETMOV	0.985	0.273
5	NESTAVIV	NDETMOV	1.000	0.385

Ilustración 47 Tabla con el valor de las métricas utilizadas para cada regla

REGLAS DE ASOCIACION	
Conjunto X	Conjunto Y
El móvil está en reposo	No hay luz
La humedad es baja	No se detecta movimiento en la vivienda
El usuario no está en la vivienda	No se detecta movimiento en la vivienda

Ilustración 48 Reglas de asociación actualizadas teniendo en cuenta la tabla anterior

## 4.5. Pruebas

Para crear el script de la Raspberry Pi, he creado pequeños programas previos que se encargan de leer los datos de los sensores de forma individual con el objetivo de hacer pruebas con cada uno de ellos. Una vez todos funcionaban de forma correcta los integré e implementé el envío de los datos tal y como se comenta en el apartado anterior. Al utilizar este enfoque a la hora de desarrollar el programa, pude corregir los problemas surgidos con los sensores de movimiento, temperatura y luminosidad durante la implementación de forma sencilla y agilizando el tiempo en el que eran solucionados. Cuando solucioné el problema relacionado con el sensor de temperatura, me di cuenta de que los valores que devolvía el sensor eran muy estables, principalmente porque el algoritmo desarrollado estudia la tendencia de los datos evitando que las

medidas sean inestables y porque el dispositivo se encontraba en el interior de la vivienda en una zona de sombra donde nunca le da el sol de forma directa, lo que favorece a que las mediciones durante el día sean estables. Por lo tanto, opté por realizar mediciones en el exterior y observé que los datos que devolvía el sensor variaban bastante en función del tramo horario y sobre todo cuando recibía la luz directa del sol. También realicé pruebas en el interior utilizando el aire climatizado durante un tiempo y se podía apreciar que la temperatura ambiente variaba a lo largo del tiempo. En consecuencia, pude determinar que los datos que devolvía el sensor eran correctos.

En cuanto a los problemas detectados con el sensor de luminosidad y movimiento, los resolví ajustando la sensibilidad de los dispositivos para evitar falsos positivos como se comenta en el apartado anterior. Aunque la colocación de los mismos dentro del hogar también ha tenido un papel fundamental. A continuación, se muestra una imagen con la disposición de los sensores respecto a la vivienda.

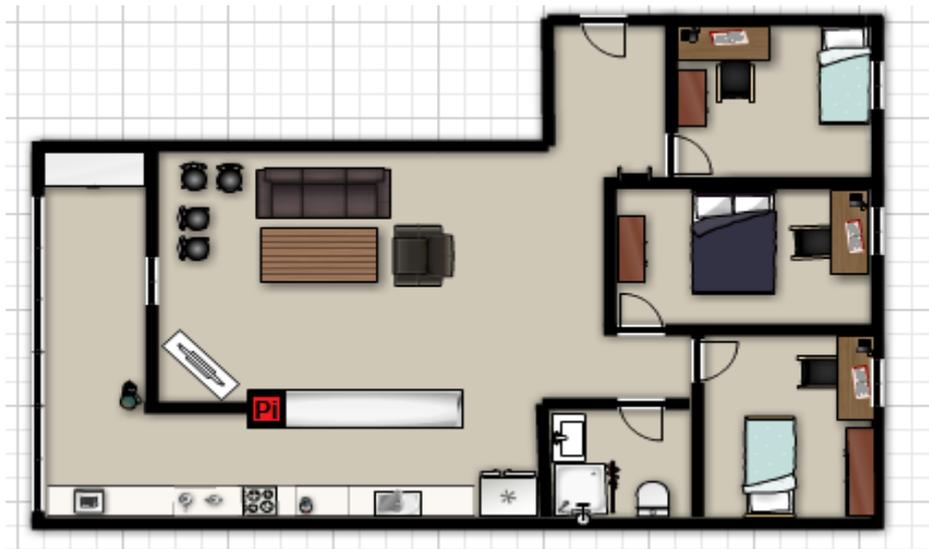


Ilustración 49 Colocación de los sensores respecto a la vivienda

La zona de colocación de la Raspberry ha sido elegida de forma estratégica para optimizar el funcionamiento de los sensores. El sensor de movimiento está orientado hacia la zona de las habitaciones y por su rango acapara casi por completo toda la zona del salón y la cocina. Además, el sensor de luminosidad recibe desde esa posición el brillo de la luz natural del día, la luz de la cocina y la del salón, que pese a haber ciertos problemas por ser una luz

blanca, la forma en la que incide sobre esta posición permite que funcione sin problemas.

Las pruebas desarrolladas sobre la aplicación Android han consistido en ver que los datos se mandaban correctamente al servidor. Al comprobar que existe una conexión estable entre el móvil y el servidor de forma previa al envío de los datos, evitamos aquellos problemas o errores que pueden surgir al realizar los envíos. Un problema que surgió fue que al tener activada la rotación de la pantalla al girar el dispositivo la aplicación se reiniciaba cada vez que rotaba. Tomé la decisión de establecer la pantalla en vertical y no girarla, aunque el usuario hiciera lo propio con el dispositivo.

Las pruebas realizadas en la aplicación web han consistido casi en su totalidad a probar el correcto funcionamiento de la API REST. Para ello he utilizado una extensión del navegador llamada RESTClient [27]. Esta herramienta me ha permitido probar todas las funciones definidas. A continuación, se muestran unas capturas de pantalla con algunas pruebas realizadas.

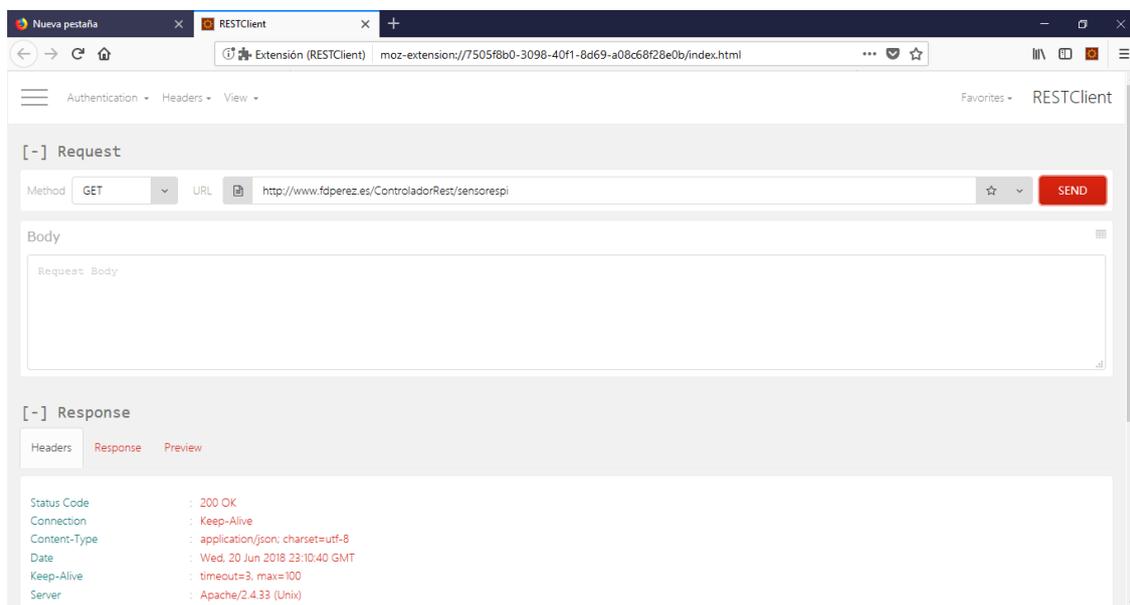


Ilustración 50 RESTClient ejemplo de una petición al servidor

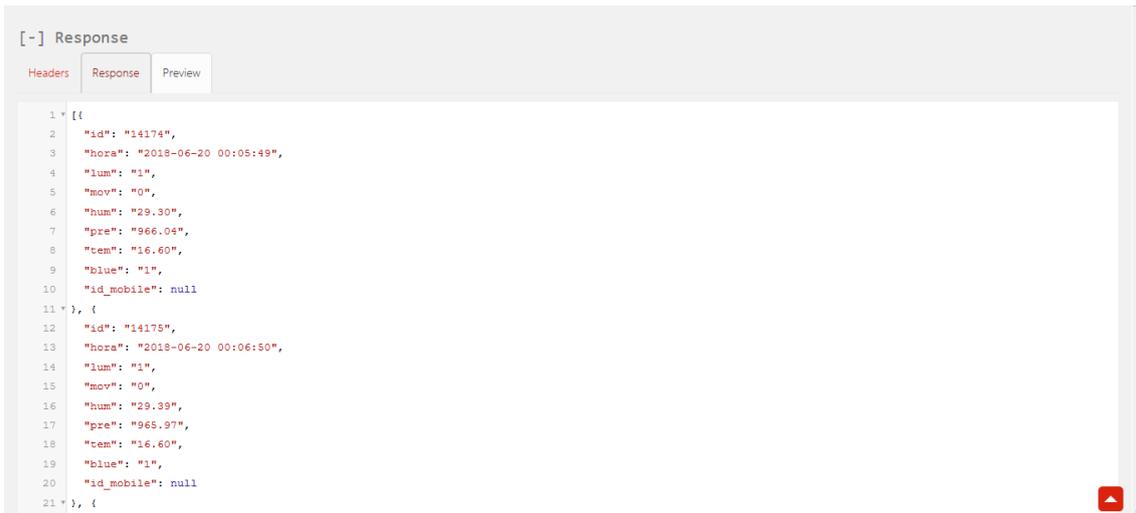


Ilustración 51 RESTClient datos devueltos por la petición

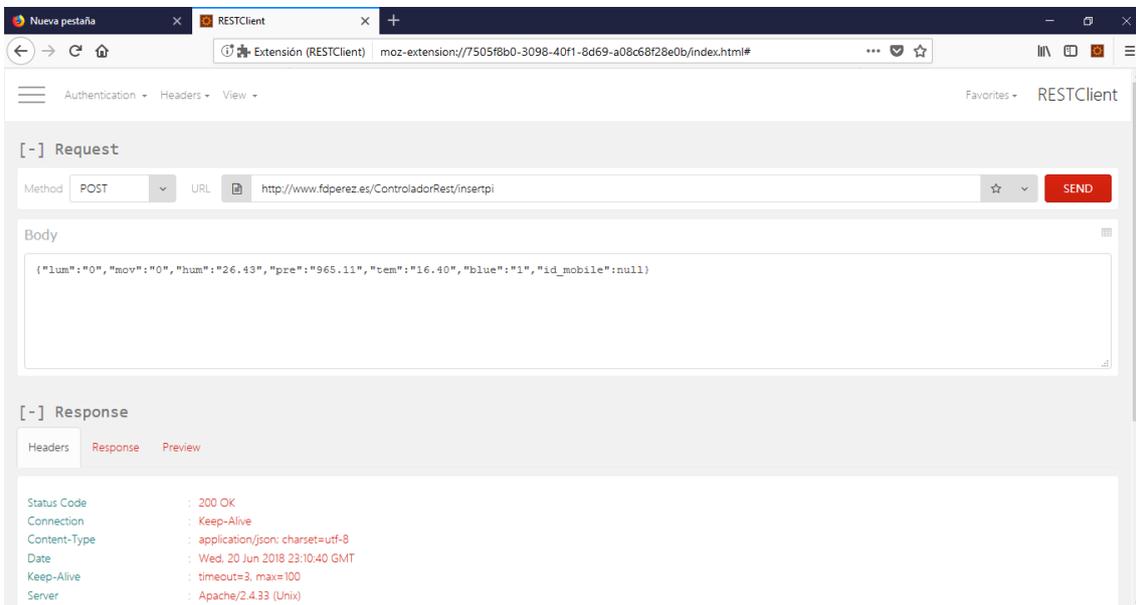


Ilustración 52 RESTClient envío de datos al servidor

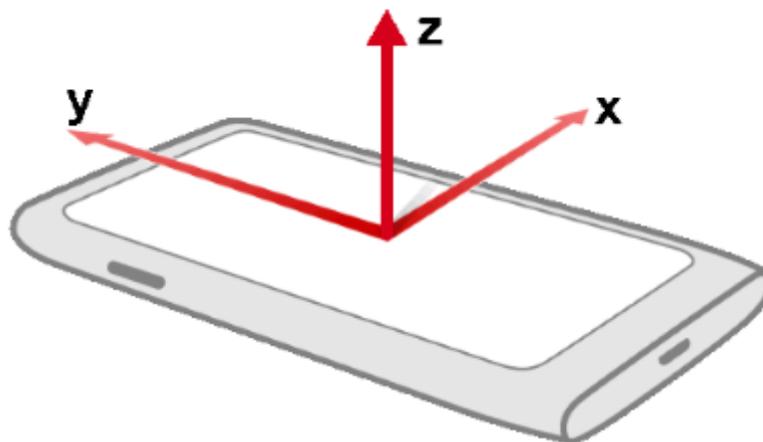
## Capítulo 5. Estudio de los datos obtenidos

En este capítulo se analizarán las características de los datos recogidos mediante los sensores, así como su representación para poder analizarlos y extraer conocimiento tras aplicar diferentes técnicas diferentes algoritmos. Por último, se evaluarán los resultados obtenidos tras aplicar estas técnicas.

### 5.1. Análisis de los datos

Como se comenta a lo largo de la memoria, este prototipo se encarga de recoger los datos a través de una Raspberry Pi y un móvil. Los datos son procesados y enviados al servidor. En este apartado se revisarán todos los datos que se envían al servidor, su forma de obtención, la frecuencia de envío y las unidades utilizadas. A continuación, se listan todos los datos que se extraen del sistema:

- Aceleración: este dato se obtiene a través del dispositivo móvil. Esta magnitud indica la variación de velocidad en una unidad de tiempo, en este caso, los datos que recibe el servidor son en metros por segundo al cuadrado( $m/s^2$ ). Los datos recogidos se obtienen sobre cada uno de sus ejes. En la siguiente ilustración se muestran la disposición de los ejes en un teléfono móvil.



*Ilustración 53 Disposición de los ejes en un teléfono móvil*

Un móvil que no se está acelerando, situado sobre una superficie plana horizontal tiene un valor de 0 para el eje X e Y mientras que el eje Z  $9.81m/s^2$  si está mirando al techo de la habitación o  $-9.81m/s^2$  si mira hacia

la superficie. Esto es debido a que la fuerza de gravedad influye a la hora de calcular las fuerzas de aceleración sobre los ejes. Los valores recogidos del sensor se representan mediante un número con una precisión de tres decimales. A continuación, se muestra una gráfica con los valores recogidos durante un periodo de tiempo determinado. Cada línea representa los valores de los datos obtenidos a través de uno de los ejes.

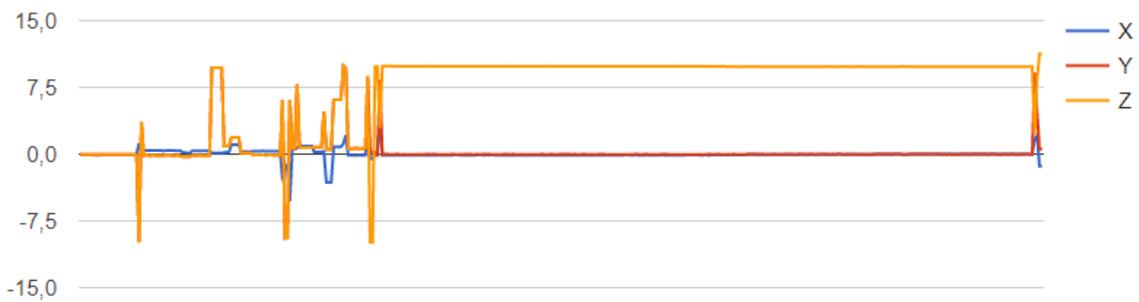


Ilustración 54 Datos del acelerómetro

- **Orientación:** dato obtenido a través del teléfono móvil mediante el giroscopio que se encarga de medir velocidad de rotación sobre cada uno de los ejes. Los datos recogidos por el giroscopio son enviados al servidor una vez cada minuto y se expresan en radianes por segundo (rad/s). La precisión de los datos recogidos es de 4 decimales. Cuando el dispositivo se encuentra en reposo sobre una superficie la velocidad angular sobre los ejes es 0. A continuación se muestra una gráfica donde cada línea representa la velocidad sobre cada uno de los ejes.



Ilustración 55 Giroscopio

- **Gravedad:** este fenómeno mide la fuerza de atracción entre dos cuerpos. También se ha obtenido a través de los sensores del teléfono móvil y su unidad son los  $m/s^2$ . Utiliza el mismo sistema de coordenadas que el acelerómetro y el giroscopio para indicar la dirección y la magnitud de la gravedad. La mayoría de los dispositivos no disponen de un sensor de

gravedad y calculan su valor a través de los datos del acelerómetro al aplicarle el filtro de Kalman [40]. Los datos medidos tienen una precisión de 8 decimales, aunque se redondean al 3er decimal cuando se envían al servidor. En la siguiente gráfica se pueden ver los datos obtenidos durante un día completo.



Ilustración 56 Datos de gravedad

Cada línea representa a la fuerza medida en cada uno de los ejes. Como podemos ver en la ilustración superior, cuando el dispositivo se encuentra en un estado de reposo, sobre una superficie, los datos en los ejes X e Y tienden a 0 mientras que el eje Z del dispositivo marca aproximadamente el valor 9.81 si se encuentra mirando al techo o -9.81 si se encuentra al revés.

- Campo magnético: este valor medido a través del teléfono móvil muestra el valor del campo magnético ambiental en los ejes X, Y y Z. Sus valores se representan con micro Teslas( $\mu\text{T}$ ). Se envían los datos al servidor una vez por minuto y la precisión de medida es al sexto decimal mientras que se envía al servidor redondeado al cuarto decimal. A continuación, se muestra una gráfica de ejemplo donde se muestran sus valores.



Ilustración 57 Datos del sensor de gravedad

- Luminosidad: valor medido con la Raspberry Pi a través del módulo LM393 y enviado al servidor en intervalos de un minuto aproximadamente.

El sensor mide valores entre 0 y 1 en función del brillo que recibe de la luz. El 0 representaba inicialmente la recepción de luz y el 1 la no recepción de la misma, aunque he aplicado una conversión para cambiar los valores y que el 1 indique que detecta iluminación y el 0 que no la detecta. Por su colocación estratégica dentro del hogar, detecta desde el primer rayo de luz que entra por la ventana de la vivienda hasta que se pierde la luz natural. En estos instantes es capaz de capturar tanto la luz de la cocina como la del comedor, pese a los problemas que tiene con las luces blancas, gracias al ángulo de incidencia que tienen sobre la posición adoptada. Por lo tanto, detecta cuando existe iluminación en la vivienda. A continuación, se muestra una gráfica de ejemplo con sus valores durante un periodo de tiempo determinado.



Ilustración 58 Iluminación de la vivienda

- **Movimiento:** dato obtenido mediante el módulo HC-SR501 PIR de la Raspberry Pi. Se realizan mediciones cada minuto y se envían los datos al servidor. En un principio planteé la opción de enviar los cambios de estado del sensor para determinar cuándo se encontraba el inquilino en la vivienda, pero como el resto de sensores se envían con una misma frecuencia determinada, he creído más conveniente enviar la detección de movimiento en la habitación principal ya que el sensor de bluetooth me indica si el usuario está o no en la vivienda. Este sensor detecta el movimiento en su zona de rango a través de un rayo infrarrojo y devuelve un 1 si detecta movimiento y un 0 si no lo detecta. Su colocación permite maximizar el rango al que llega el sensor gracias al ángulo y distancia que es capaz de detectar. Al haber bajado la sensibilidad del sensor se evita la detección de falsos positivos. A continuación, se muestra un ejemplo de los datos registrados por el sensor.

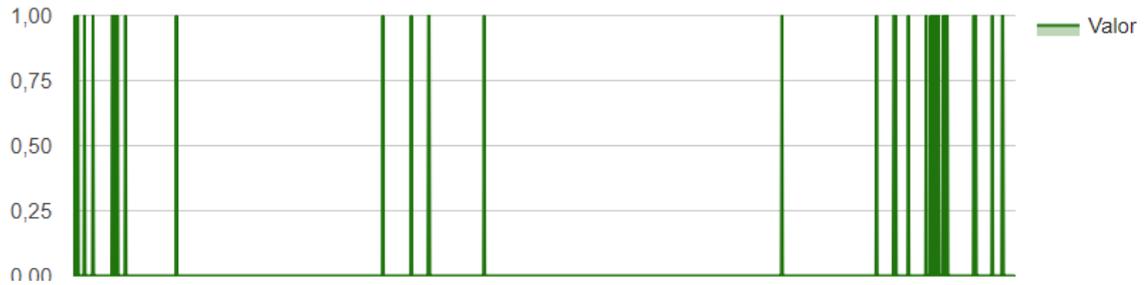


Ilustración 59 Datos del sensor de movimiento

- **Humedad:** este dato contiene la cantidad de vapor de agua en el aire. El dato está representado como un porcentaje, es decir la cantidad de humedad en el aire en comparación con lo que el aire puede contener esa temperatura. A esta medida se le llama humedad relativa. El módulo Sense HAT de la Raspberry devuelve este valor y se encarga de procesarlo y mandarlo al servidor con dos decimales. Las medidas del sensor son un poco inestables y los valores que miden se encuentran entre 0 y 100. El módulo se encuentra localizado en la misma zona que la Raspberry Pi y se observan ciertas diferencias con las acciones diarias. Por ejemplo, se producen picos en las medidas de humedad cuando se utiliza la ducha o se hacen carnes que cuya composición contiene mucha agua, ya que se incrementa la humedad del ambiente. A continuación, podemos ver una muestra de este suceso en un gráfico.

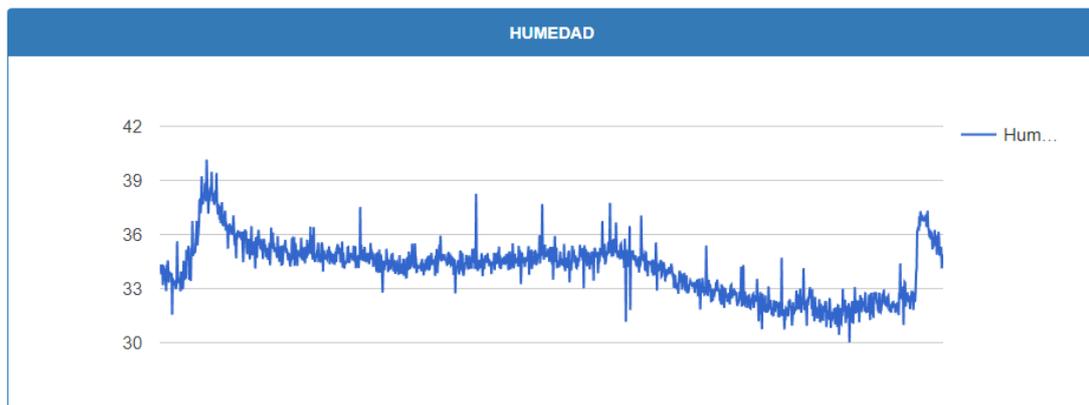


Ilustración 60 Medida de la humedad tras un evento

A su vez también se pueden detectar bajadas debido al uso del aire acondicionado ya que elimina humedad del aire. A continuación, podemos ver otro ejemplo de este fenómeno.

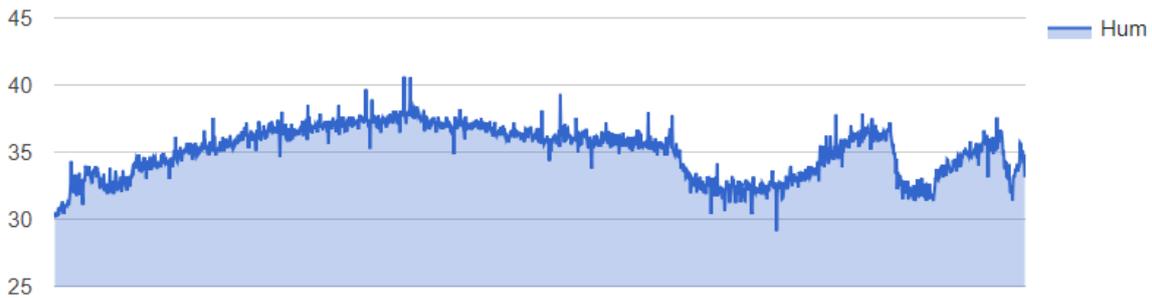


Ilustración 61 Humedad del ambiente con el aire acondicionado

También hay que destacar que al encontrarse dentro de una vivienda y no en el exterior, las variaciones de los datos que se producen por respecto al cambio de temperaturas y a las posibles lluvias del exterior son muy pequeñas.

- Presión: esta medida representa la fuerza por unidad de superficie que ejerce el aire de la atmósfera sobre la superficie terrestre. Se ha medido utilizando el módulo Sense HAT y se representa mediante milibares(mbar). Los milibares tienen el mismo valor que los hectopascales (hPa), por lo que son intercambiables. Este dato se recoge y envía cada minuto redondeándolo al segundo decimal. La presión normal se establece en 1013hPa [41] sobre el nivel del mar y disminuye conforme aumenta la altitud. Esta medida es la que se utiliza para definir una alta presión, si supera la presión normal, lo que se corresponde con un anticiclón o baja correspondiéndose con una borrasca. Además de la diferencia de altura, hay que tener que los valores medidos son en interior por lo que el rango de los valores medidos va a diferir de los valores del exterior. En la siguiente gráfica se muestra la representación de los datos a lo largo de un día completo.



Ilustración 62 Medidas del sensor de presión

- Temperatura: los datos de temperatura que recoge el módulo Sense HAT se expresan en grados centígrados y son enviados al servidor una vez por

minuto. Los datos de la temperatura se pueden obtener a través del sensor de presión y de humedad. Como se comenta en el apartado de implementación, el valor obtenido, aproximado al segundo decimal, se obtiene de realizar la media entre ambos valores y de estudiar la tendencia de los mismos. Aunque en el gráfico parece que existe inestabilidad, si nos fijamos en la diferencia que existe entre los valores, podemos ver que es mínima. Esto es consecuencia del estudio de la tendencia de los últimos valores medidos. Al igual que pasa con el resto de sensores, las medidas realizadas se ven limitadas porque se realizan en el interior de la vivienda donde los datos varían de forma considerable con los del exterior. A continuación, se muestra un gráfico con los datos obtenidos durante un periodo de tiempo determinado.

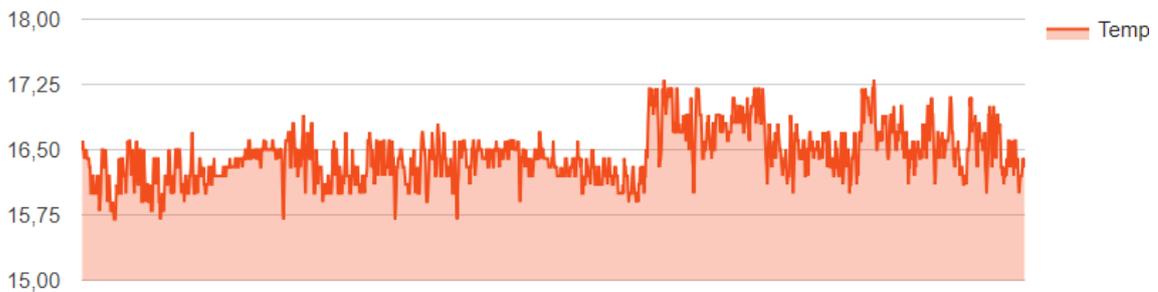


Ilustración 63 Medidas del sensor de temperatura

- Estado: este dato se corresponde con el sensor de bluetooth e indica si el inquilino se encuentra dentro de la vivienda. Este sensor busca los dispositivos disponibles y devuelve un 1 si encuentra la dirección MAC del teléfono móvil en el rango y un 0 si no está dentro del rango del bluetooth. El proceso dura unos segundos y se realiza cada minuto para enviar los datos al servidor. A continuación, se muestran los datos registrados por este sensor en un rango de tiempo determinado.

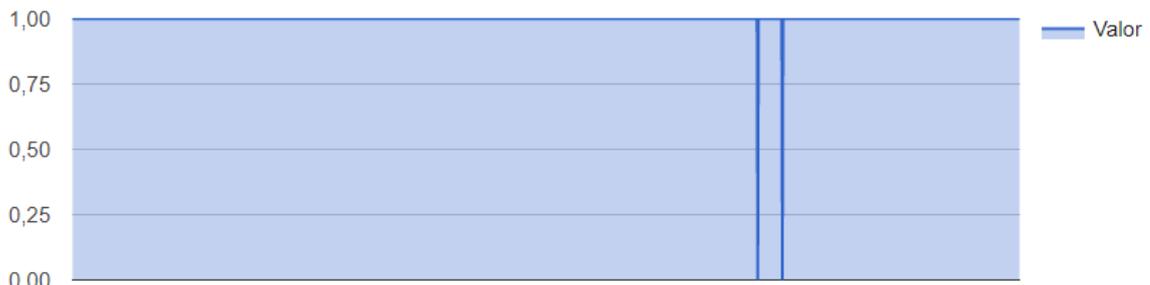


Ilustración 64 Estado del inquilino dentro de la vivienda

## 5.2. Planteamiento de la representación de los datos

En este apartado se trata la representación de los datos mencionados en el apartado anterior. Algunos datos se utilizarán tal cual están, y otros deben ser preparados para poder utilizarlos correctamente y poder obtener conocimiento de los mismos. Para ello, he partido de un conjunto de datos recogido durante diez días y exportado en un archivo de valores separados por comas (CSV). Estos datos se adjuntarán a la memoria del proyecto. Los datos recogidos son los que aparecen en la siguiente tabla.

Aceleración en el eje X	Aceleración en el eje Y	Aceleración en el eje Z
Orientación en el eje X	Orientación en el eje Y	Orientación en el eje Z
Gravedad en el eje X	Gravedad en el eje Y	Gravedad en el eje Z
Campo magnético	Luminosidad	Movimiento
Humedad	Presión	Temperatura
Estado	Identificador	Fecha de los datos

Tabla 6 Resumen datos recogidos

La mayor parte de los datos recogidos por el teléfono móvil son poco representativos ya que se basan en estudiar el movimiento, inclinación y gravedad del dispositivo. Debido a los problemas de energía por los que no me ha sido posible recolectar datos durante las 24 horas del día, he recogido datos todas las noches mientras el dispositivo se cargaba y se mantenía estático y a lo largo del día, cuando salía a la calle para recoger datos del movimiento. Por lo tanto, se podrían distinguir tres estados del dispositivo: reposo, movimiento leve o movimiento, en función de la actividad realizada durante la recogida de datos. Estos datos no aportan mucho, tal como están, a la hora de establecer rutinas o estudiar el comportamiento de los casos de estudio. Por lo tanto, he añadido nuevas columnas en el archivo y he utilizado diferentes fórmulas para rellenar sus valores. Estos valores van a establecer, en función de los datos recogidos sobre cada uno de los ejes de los sensores mencionados anteriormente, de manera que se distinga entre las variaciones de los datos medidos y los valores que se obtendrían en un estado de reposo. También he establecido un pequeño umbral para suplir las pequeñas variaciones que puedan existir. A continuación,

se muestra en pseudocódigo las funciones establecidas para detectar si existe movimiento o no en el acelerómetro, giroscopio y el sensor de gravedad.

```

if Acelerometro=NULL:
    Campo=NULL
Else:
    Si Acelerometro.x>-1 && Acelerometro.x<1 &&
    Acelerometro.y>-1 && Acelerometro.y<1 &&
    ((Acelerometro.z>9.7 && Acelerometro.z<9.9)||
    (Acelerometro.z<-9.7 && Acelerometro.z>-9.9))
        Campo=Reposo
    Else:
        Campo=Movimiento
    
```

```

if Giroscopio=NULL:
    Campo=NULL
Else:
    Si Giroscopio.x>-1 && Giroscopio.x<1 &&
    Giroscopio.y>-1 && Giroscopio.y<1 &&
    Giroscopio.z>-1 && Giroscopio.z<1
        Campo=Reposo
    Else:
        Campo=Movimiento
    
```

```

if Gravedad=NULL:
    Campo=NULL
Else:
    Si Gravedad.x>-1 && Gravedad.x<1 &&
    Gravedad.y>-1 && Gravedad.y<1 &&
    ((Gravedad.z>9.7 && Gravedad.z<9.9)||
    (Gravedad.z<-9.7 && Gravedad.z>-9.9))
        Campo=Reposo
    Else:
        Campo=Movimiento
    
```

Puede ser que uno de estos sensores tenga ciertas alteraciones mientras otros estén en un estado de reposo, por ejemplo, puede que se esté realizando una actividad física en la que el móvil se encuentra en una zona en la que no se detecta ningún giro mientras que el resto de sensores si varían. Por lo tanto, he añadido una columna que indica el estado del usuario, indicando si está en reposo o movimiento, en función del estado de estos tres sensores. En la siguiente tabla se muestra el criterio seguido a la hora de establecer el estado del usuario.

<b>Acelerómetro</b>	<b>Giroscopio</b>	<b>Gravedad</b>	<b>Estado</b>
Reposo	Reposo	Reposo	Reposo
Reposo	Reposo	Movimiento	Reposo
Reposo	Movimiento	Reposo	Reposo
Reposo	Movimiento	Movimiento	Movimiento
Movimiento	Reposo	Reposo	Reposo
Movimiento	Reposo	Movimiento	Movimiento
Movimiento	Movimiento	Reposo	Movimiento
Movimiento	Movimiento	Movimiento	Movimiento

Tabla 7 Estado del usuario a través de los sensores del dispositivo móvil

Los datos provenientes del sensor de campo magnético son irrelevantes, no tiene relación con el resto de variables y no tiene utilizad a la hora de extraer las reglas de asociación entre las diferentes variables. Estos datos son totalmente independientes y no tiene sentido discretizarlos por lo que en el momento de preprocesar las entradas dentro de KNIME se obviará esta variable.

En cuanto a los datos provenientes de la Raspberry Pi, he categorizado los datos provenientes de los sensores de temperatura, presión y humedad. Los datos con respecto al sensor de humedad los he categorizado en poca humedad si el valor se encuentra por debajo del 35% y alta humedad si supera el 35%. Este valor que divide ambos intervalos lo he establecido teniendo en cuenta los valores medidos dentro de la vivienda mientras que llovía en el exterior. Los datos del sensor de temperatura los he categorizado en baja temperatura si los datos se encuentran por debajo de los 16 grados y alta temperatura si se superan los 16 grados dentro del hogar. En esta categoría los datos se podrían clasificar

en baja ( $\infty$ -8), un poco baja (8-16), un poco alta (16-24) y alta temperatura (24- $\infty$ ), pero como la diferencia entre el valor mínimo y máximo es de apenas dos grados debido a que la variación de la misma dentro del hogar es muy pequeña, los he clasificado teniendo en dos categorías ya que todos se agrupaban en las categorías centrales. Para que los datos se hubiesen podido clasificar en las cuatro categorías descritas anteriormente, sería necesario la recolección de datos durante un año completo para observar una variación mucho mayor.

La mayoría de los problemas han sido a la hora de determinar el límite que determina las altas y bajas presiones, para categorizar los datos de presión, dentro de la vivienda debido a que son inferiores a las que hay en el exterior. Como indican los estudios, 1013hPa es el límite que establece la diferencia entre las altas y bajas presiones [41]. Como las mediciones en el interior son inferiores, he utilizado las mediciones del exterior de la Universidad de Jaén [42] para estudiar la relación existente entre las medidas realizadas en el interior y el exterior. A continuación, se muestran los datos obtenidos en un mismo periodo de tiempo entre la estación de meteorológica y la Raspberry Pi a través del módulo.



Ilustración 65 Gráfica de presión de la Raspberry

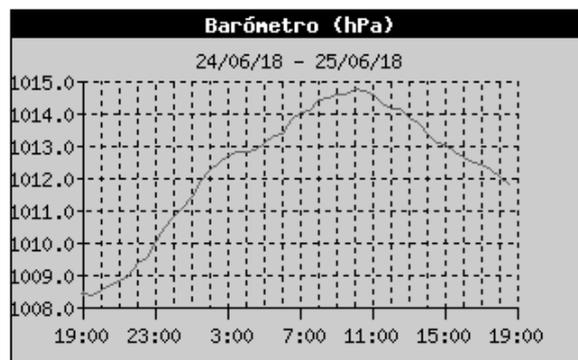


Ilustración 66 Presión en la estación meteorológica

Como podemos ver, ambas ilustraciones presentan una función de distribución similar. Además, me he asegurado cogiendo varios valores aleatorios en horas similares y estudiando la relación existente entre ambas gráficas. Como resultado he obtenido que los valores medidos por la Raspberry en interior guardan una relación del 0,9518hPa por cada 1hPa en la gráfica que representa la presión en la estación meteorológica exterior de la Universidad de Jaén. Por lo tanto, teniendo en cuenta este coeficiente que relaciona ambas gráficas, el límite que determina si una presión es alta o baja dentro de la vivienda lo he establecido en 964,8hPa.

En el resto de variables, cuyos datos se componen de 0 y 1 para determinar el estado de los sensores, se ha añadido una serie de etiquetas lingüísticas para determinar los estados de los sensores. El sensor de luminosidad tendrá la etiqueta “LUZ” si el sensor está activado o “NLUZ” si no lo está. El detector de movimiento utilizará la etiqueta “DETMOV” si detecta movimiento o “NDETMOV” si no lo detecta. Por último, el sensor bluetooth valdrá “ESTAVIV” si el usuario se encuentra dentro de la vivienda o “NESTAVIV” si no se encuentra dentro de la misma. A continuación, se muestra un resumen con todas las columnas que componen el conjunto de datos final tras aplicar los cambios comentados a lo largo de este apartado.

- Id: entero que indica el número de registro de la tabla.
- Hora: indica la fecha y hora en la que el dato fue introducido en la base de datos.
- Acex: indica la aceleración en el eje X y se representa mediante un número con tres decimales.
- Acey: indica la aceleración en el eje Y y se representa mediante un número con tres decimales.
- Acez: indica la aceleración en el eje Z y se representa mediante un número con tres decimales.
- Girx: indica la posición del teléfono móvil en el eje X. Se representa mediante un número con cuatro decimales.
- Giry: indica la posición del teléfono móvil en el eje Y. Se representa mediante un número con cuatro decimales.

- Girz: indica la posición del teléfono móvil en el eje Z. Se representa mediante un número con cuatro decimales.
- Grax: indica la gravedad medida sobre el eje X. Se representa mediante un número con tres decimales.
- Gray: indica la gravedad medida sobre el eje Y. Se representa mediante un número con tres decimales.
- Graz: indica la gravedad medida sobre el eje Z. Se representa mediante un número con tres decimales.
- Mag: indica el campo magnético. Se representa mediante un número con tres decimales.
- Lum: indica el estado del sensor de luminosidad con 0 y 1.
- Mov: indica el estado del detector de movimiento con 0 y 1.
- Hum: indica la humedad relativa. Se indica con un número entre 0 y 100.
- Pre: indica la presión atmosférica. Se indica con un número aproximado al segundo decimal.
- Tem: indica la temperatura. Se aproxima el valor medido al primer decimal.
- Blue: indica si el usuario se encuentra dentro de la vivienda con 0 y 1.
- Acemov: indica el estado del acelerómetro del móvil. Los estados son “REPOSO” y “MOVIMIENTO”.
- Girmov: indica el estado del giroscopio. Los estados son “REPOSO” y “MOVIMIENTO”.
- Gramov: indica el estado del dispositivo en función de los cambios medidos en el sensor de gravedad. Los estados son “REPOSO” y “MOVIMIENTO”.
- Movdis: indica el estado del dispositivo móvil en función del estado del acelerómetro, giroscopio y sensor de gravedad. Los estados son “REPOSO”, y “MOVIMIENTO”.
- Humdis: indica el estado en función del valor medido por el sensor de humedad. Sus valores pueden ser “BAJO” y “ALTO”.
- Predis: indica el estado en función del valor medido por el sensor de presión. Sus valores pueden ser “BAJO” y “ALTO”.

- Temdis: indica el estado en función del valor medido por el sensor de temperatura. Sus valores pueden ser “BAJO” y “ALTO”.
- Lumdis: indica el estado del sensor de luminosidad. Sus valores pueden ser “LUZ” y “NLUZ” en función de si detecta iluminación o no.
- Movdis: indica el estado del sensor encargado de la detección de movimiento. Sus valores pueden ser “DETMOV” si detecta movimiento y “NDETMOV” si no detecta movimiento.
- Bluedis: indica si el usuario se encuentra en la vivienda en función de los valores detectados por el bluetooth. Tendrá el valor “ESTAVIV” si el usuario se encuentra en la vivienda o “NESTAVIV” si no se encuentra dentro de la vivienda.

Aunque algunas variables se podrían haber categorizado en tres categorías diferentes, como la humedad o temperatura, se ha optado por dos categorías diferentes debido a que uno de los algoritmos utilizados en KNIME requiere como entrada el uso de variables binarias.

### 5.3. Evaluación y validación de los resultados

Tras preparar el conjunto de datos para estudiarlo, solo nos falta realizar un estudio de los datos recogidos durante los diez días de estudio que se adjuntan en la entrega del proyecto. En la preparación se han discretizado las variables numéricas, y se han categorizado en dos categorías diferentes para que estén adaptados a las entradas que necesitan los algoritmos descritos en el apartado anterior. Antes de aplicar los algoritmos para la obtención de las reglas de asociación difusas entre las variables, he realizado un análisis exploratorio de los datos recogidos por los sensores. A continuación, se muestra un esquema del diseño de la parte de análisis de los datos.

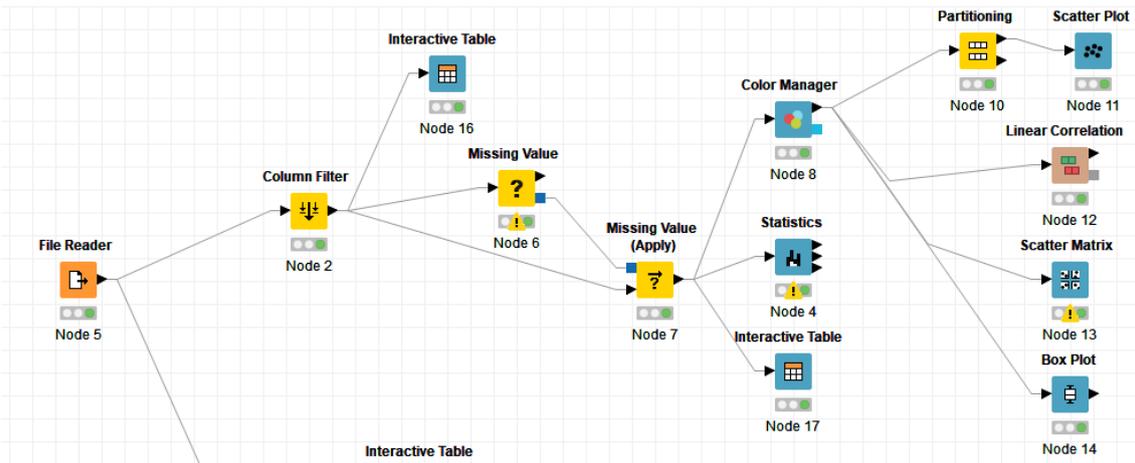


Ilustración 67 Análisis exploratorio en KNIME

El nodo “File Reader” se encarga de leer los datos del documento. En el nodo “Column Filter” se realiza un filtrado para quedarnos solo con aquellas variables que contienen datos numéricos ya que son las que vamos a analizar. En los nodos “Interactive Table”, se muestra el estado de los datos. A continuación, se muestra un ejemplo de lo que nos muestra este nodo.

Row ID	S_acex	S_acey	S_acez	S_girx	S_giry	S_girz	S_grax	S_gray	S_graz	S_mag	i_lum	i_mov	S_hum	S_pre	S_tem	i_blue
1	?	?	?	?	?	?	?	?	?	?	0	0	32,93	962,33	17,4	1
2	0,193	-0,08	-0,08	0,0003	0	0,0004	0,193	-0,074	9,759	15,009	0	0	35,77	962,33	17,2	1
3	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793	1	0	35,77	962,35	16,9	1
4	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793	0	0	34,82	962,38	16,6	1
5	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793	0	1	35,26	962,42	16,6	1
6	?	?	?	?	?	?	?	?	?	?	0	0	37,19	962,39	16,9	1
7	?	?	?	?	?	?	?	?	?	?	0	0	35,36	962,4	16,8	1
8	?	?	?	?	?	?	?	?	?	?	0	0	35,23	962,4	16,8	1
9	?	?	?	?	?	?	?	?	?	?	0	0	35,28	962,35	16,7	1
10	?	?	?	?	?	?	?	?	?	?	0	0	35,8	962,37	16,6	1
11	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793	0	1	35,56	962,33	16,6	1
12	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,065	-4,793	0	1	36,37	962,33	16,6	1
13	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,017	-4,793	0	1	35,57	962,35	16,7	1
14	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,017	-4,793	0	1	36,7	962,33	16,4	1
15	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,017	-4,793	1	0	35,95	962,34	16,4	1
16	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,017	-4,793	1	0	36,28	962,31	16,4	1

Ilustración 68 Datos mostrados en el nodo "Interactive Table" de KNIME

Como podemos ver, aparecen todos los valores e interrogaciones en rojo en los valores perdidos. Estos valores hacen referencia a la falta de datos en ciertas filas. Como el prototipo de sistema no recolecta los datos del teléfono móvil las 24 horas y se han recolectado de forma parcial mientras he cargado el dispositivo por las noches o cuando he salido a la calle, se puede apreciar una mayoría de filas en las que faltan valores en estos campos. Para poder trabajar con los datos es necesario tomar una decisión con este tipo de valores. Existen tres opciones principalmente:

1. Eliminar todas las filas que contienen valores perdidos.
2. Sustituir estos valores por el valor medio.
3. Sustituir los valores perdidos por el valor más común, es decir, la moda.

Como en las variables discretizadas se han establecido en dos categorías diferentes, y con el objetivo de utilizar un criterio común tanto en la fase de

análisis exploratorio como en el estudio de las reglas de asociación entre los diferentes conjuntos generados, no tiene sentido el sustituir estos valores por la media ya que en las variables discretizadas este valor no existe. Además, si se eliminan todas las filas que contienen valores perdidos, se perdería mucha información ya que son la mayor parte de los registros. Como el criterio seguido para la recolección de estos datos ha sido recogerlos durante toda la noche y en salidas al exterior, la mayor parte de las medidas no tomadas se corresponden con estados de reposo dentro o fuera de casa. Por lo tanto, considero que el mejor criterio consiste en sustituir estos valores por el más común. Tras aplicar este criterio, los datos quedarían de la siguiente forma.

Row ID	S acex	S acey	S acez	S girx	S giry	S girz	S grax	S gray	S graz	S mag
1	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
2	0,193	-0,08	-0,08	0,0003	0	0,0004	0,193	-0,074	9,759	15,009
3	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793
4	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793
5	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793
6	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
7	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
8	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
9	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
10	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
11	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,068	-4,793
12	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,065	-4,793
13	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,017	-4,793
14	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,017	-4,793
15	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	4,017	-4,793
16	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	3,875	-4,793
17	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	3,875	-4,793
18	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	3,875	-4,793
19	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	3,75	-4,793
20	0,62	-0,377	-0,377	0,3625	-0,0451	-0,2666	0,538	-0,358	3,75	-4,793
21	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
22	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
23	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108
24	0,332	-0,035	-0,035	0	0	0	0,329	-0,036	9,806	6,108

Ilustración 69 Datos mostrados en el nodo "Interactive Table" de KNIME tras tratar los valores perdidos

Como podemos ver, el conjunto de datos no tiene valores perdidos y los valores que se han tomado son los de un estado de reposo. Mediante el nodo "Statistics" se puede obtener información estadística sobre las diferentes variables estudiadas como el valor mínimo, máximo, media o moda.

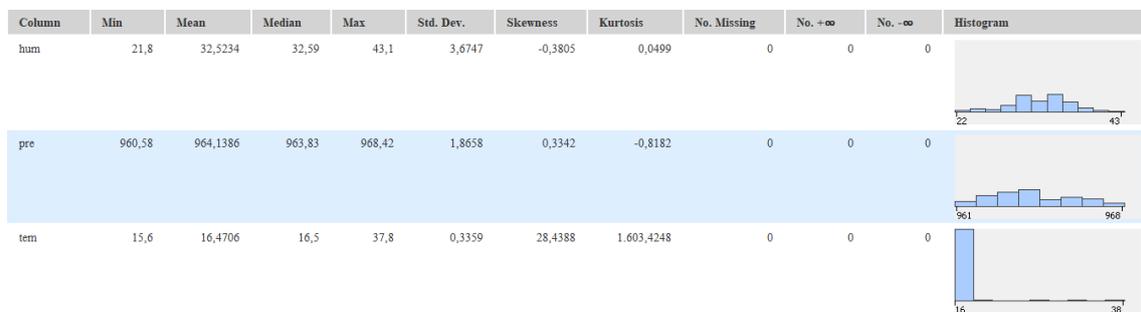


Ilustración 70 Estudio estadístico de los datos

Como podemos ver en la ilustración superior, la humedad media es del 32,52%, la presión máxima medida dentro de la vivienda durante el periodo que se han recogido estos datos es de 968,42hPa y la temperatura mínima recogida ha sido de 15,6°. La desviación típica de los datos no es muy grande, lo que quiere decir que los datos no están muy dispersos.

Los nodos “Scatter Plot” y “Scatter Matrix” nos permiten ver la matriz de dispersión para estudiar la relación existente entre dos pares de variables. Es complicado detectar si existe correlación entre dos variables viendo la matriz por lo que he optado por calcular el coeficiente de correlación lineal existente, con el nodo “Linear Correlation”, entre las diferentes variables ya que se ve con más claridad.

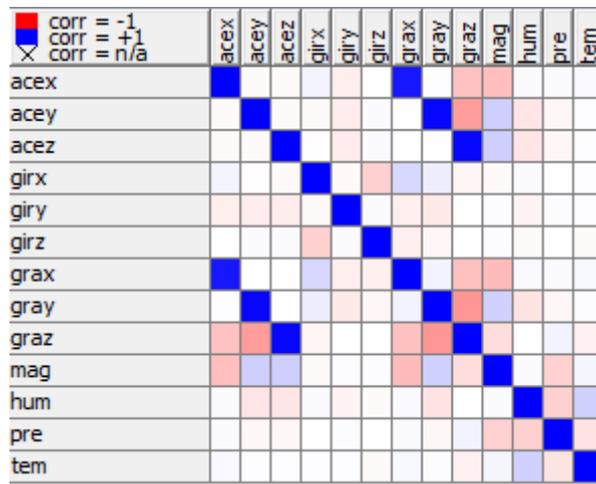


Ilustración 71 Correlación lineal entre los diferentes pares de variables

Como podemos ver en la ilustración superior, existe una alta correlación entre los valores medidos por el acelerómetro y el sensor de gravedad en sus ejes. Como se indicó en el primer apartado de este capítulo, la mayoría de los teléfonos móviles no disponen de un sensor de gravedad y estos datos se obtienen a través del acelerómetro principalmente. Esto explicaría la alta correlación entre los datos de ambos sensores.

Por último, en este análisis exploratorio, también he obtenido el diagrama de caja para cada variable con el objetivo de detectar valores anómalos.

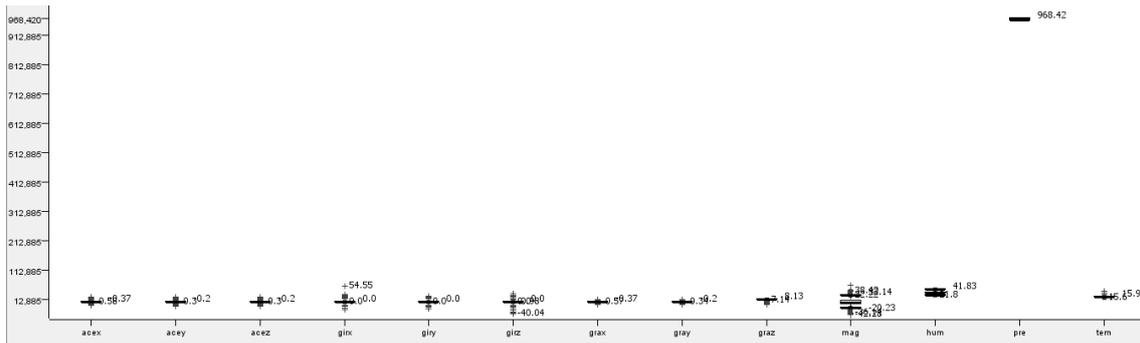


Ilustración 72 Diagrama de caja

A continuación, se muestra el esquema final obtenido para la obtención de las reglas de asociación entre los diferentes conjuntos.

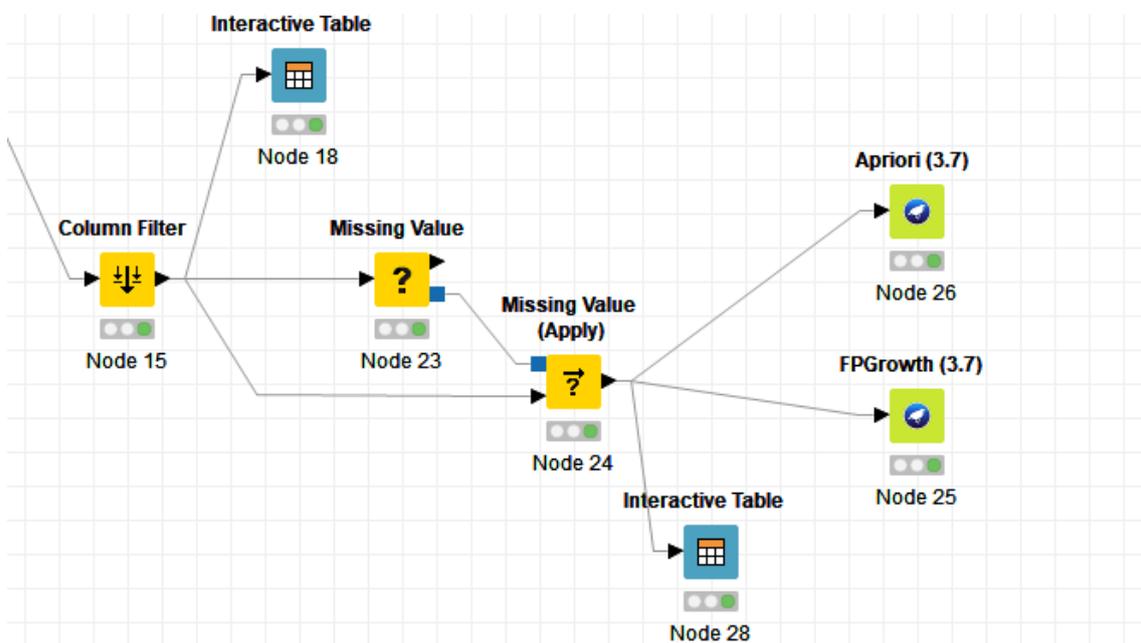


Ilustración 73 Esquema KNIME para la obtención de las reglas de asociación

Como podemos ver en el esquema, se realiza un filtrado en el que se seleccionan solo aquellas variables que contienen los datos discretizados. Además, se tratan los valores perdidos sustituyendo estos valores por el valor más frecuente al igual que se determinó en la parte de exploración de los datos. Tras tenerlo todo listo se ha procedido a configurar los algoritmos.

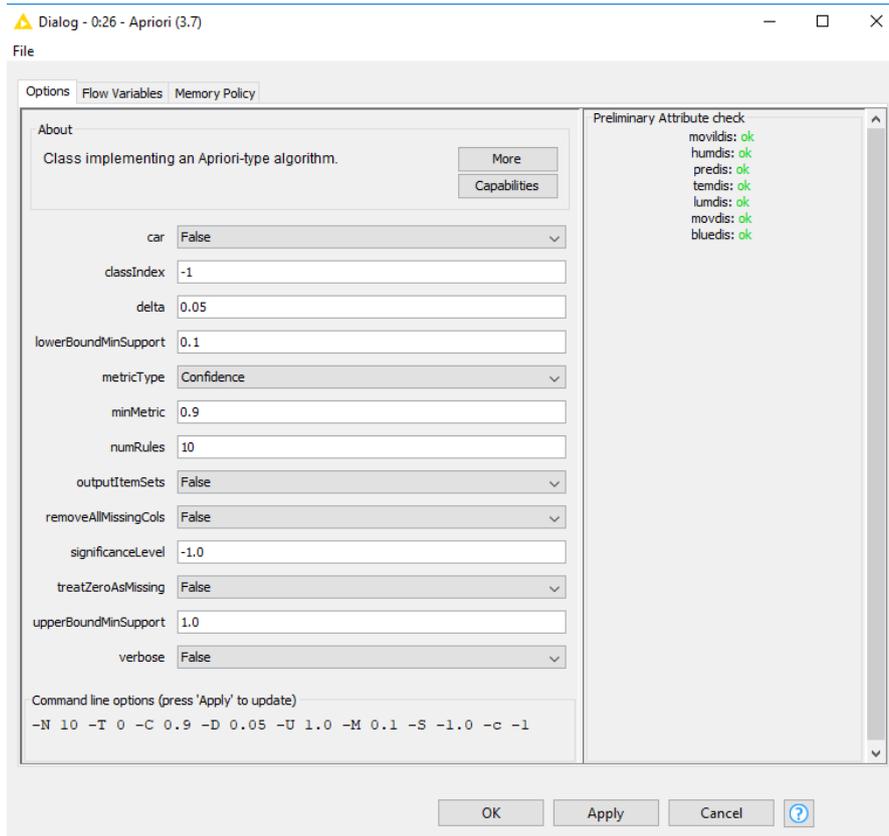


Ilustración 74 Configuración algoritmo Apriori

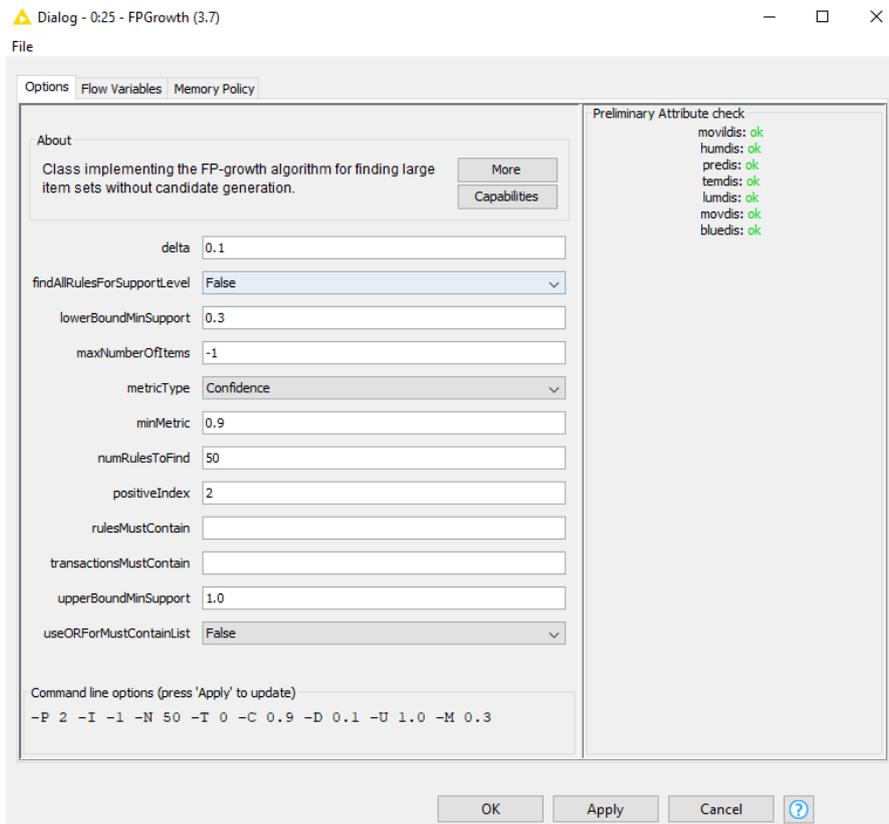


Ilustración 75 Configuración del algoritmo FP-Growth

Las siguientes propiedades son las más importantes y se encuentran en ambos algoritmos:

- numRules: indica el número de reglas que queremos obtener. Lo he establecido en 50.
- metricType: indica la métrica utilizada para clasificar las reglas. He utilizado la confianza.
- minMetric: indica la confianza mínima que debe tener una regla. He establecido esta variable en 0,9.
- delta: indica el delta por el que se disminuye el soporte mínimo en cada iteración. He utilizado el valor 0,01.
- upperBoundMinSupport: indica el límite superior para el soporte mínimo. Establecido por defecto en 1.
- lowerBoundMinSupport: indica el límite inferior del soporte mínimo. El valor lo he establecido en 0,3.

A continuación, se muestran los resultados obtenidos tras aplicar ambos algoritmos:

1. predis=BAJO 8632 ==> temdis=ALTO 8598 <conf:(1)> lift:(1) lev:(0) [34] conv:(1.95)  
 2. movdis=NDETMOV bluedis=ESTAVIV 8958 ==> temdis=ALTO 8893 <conf:(0.99)> lift:(1) lev:(0) [5] conv:(1.07)  
 3. bluedis=ESTAVIV 9532 ==> temdis=ALTO 9461 <conf:(0.99)> lift:(1) lev:(0) [4] conv:(1.05)  
 4. movildis=REPOSO movdis=NDETMOV bluedis=ESTAVIV 8574 ==> temdis=ALTO 8510 <conf:(0.99)> lift:(1) lev:(0) [3] conv:(1.04)  
 5. movildis=REPOSO bluedis=ESTAVIV 9087 ==> temdis=ALTO 9017 <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.01)  
 6. movildis=REPOSO movdis=NDETMOV 12872 ==> temdis=ALTO 12772 <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1.01)  
 7. movdis=NDETMOV 13546 ==> temdis=ALTO 13440 <conf:(0.99)> lift:(1) lev:(0) [1] conv:(1)  
 8. movildis=REPOSO 13418 ==> temdis=ALTO 13312 <conf:(0.99)> lift:(1) lev:(0) [0] conv:(0.99)  
 9. lumdis=LUZ movdis=NDETMOV 8768 ==> temdis=ALTO 8698 <conf:(0.99)> lift:(1) lev:(0) [0] conv:(0.98)  
 10. movildis=REPOSO lumdis=LUZ 8769 ==> temdis=ALTO 8698 <conf:(0.99)> lift:(1) lev:(0) [-1] conv:(0.96)  
 11. lumdis=LUZ 9358 ==> temdis=ALTO 9282 <conf:(0.99)> lift:(1) lev:(0) [-1] conv:(0.96)  
 12. movildis=REPOSO humdis=BAJO movdis=NDETMOV 9641 ==> temdis=ALTO 9548 <conf:(0.99)> lift:(1) lev:(0) [-16] conv:(0.81)  
 13. movildis=REPOSO humdis=BAJO 10016 ==> temdis=ALTO 9917 <conf:(0.99)> lift:(1) lev:(0) [-19] conv:(0.79)  
 14. humdis=BAJO movdis=NDETMOV 10007 ==> temdis=ALTO 9908 <conf:(0.99)> lift:(1) lev:(0) [-19] conv:(0.79)  
 15. humdis=BAJO 10405 ==> temdis=ALTO 10300 <conf:(0.99)> lift:(1) lev:(0) [-22] conv:(0.78)  
 16. humdis=BAJO temdis=ALTO movdis=NDETMOV 9908 ==> movildis=REPOSO 9548 <conf:(0.96)> lift:(1.02) lev:(0.01) [154] conv:(1.43)  
 17. humdis=BAJO movdis=NDETMOV 10007 ==> movildis=REPOSO 9641 <conf:(0.96)> lift:(1.02) lev:(0.01) [153] conv:(1.42)  
 18. humdis=BAJO temdis=ALTO 10300 ==> movildis=REPOSO 9917 <conf:(0.96)> lift:(1.02) lev:(0.01) [151] conv:(1.39)  
 19. movildis=REPOSO humdis=BAJO temdis=ALTO 9917 ==> movdis=NDETMOV 9548 <conf:(0.96)> lift:(1.01) lev:(0) [56] conv:(1.15)  
 20. humdis=BAJO 10405 ==> movildis=REPOSO 10016 <conf:(0.96)> lift:(1.02) lev:(0.01) [151] conv:(1.39)  
 21. movildis=REPOSO humdis=BAJO 10016 ==> movdis=NDETMOV 9641 <conf:(0.96)> lift:(1.01) lev:(0) [54] conv:(1.14)  
 22. humdis=BAJO temdis=ALTO 10300 ==> movdis=NDETMOV 9908 <conf:(0.96)> lift:(1.01) lev:(0) [49] conv:(1.12)  
 23. humdis=BAJO 10405 ==> movdis=NDETMOV 10007 <conf:(0.96)> lift:(1) lev:(0) [48] conv:(1.12)  
 24. movildis=REPOSO temdis=ALTO 13312 ==> movdis=NDETMOV 12772 <conf:(0.96)> lift:(1) lev:(0) [30] conv:(1.06)  
 25. movildis=REPOSO 13418 ==> movdis=NDETMOV 12872 <conf:(0.96)> lift:(1) lev:(0) [29] conv:(1.05)  
 26. temdis=ALTO 14041 ==> movdis=NDETMOV 13440 <conf:(0.96)> lift:(1) lev:(0) [1] conv:(1)  
 27. movdis=NDETMOV bluedis=ESTAVIV 8958 ==> movildis=REPOSO 8574 <conf:(0.96)> lift:(1.01) lev:(0.01) [81] conv:(1.21)  
 28. temdis=ALTO movdis=NDETMOV bluedis=ESTAVIV 8893 ==> movildis=REPOSO 8510 <conf:(0.96)> lift:(1.01) lev:(0.01) [78] conv:(1.2)  
 29. humdis=BAJO movdis=NDETMOV 10007 ==> movildis=REPOSO temdis=ALTO 9548 <conf:(0.95)> lift:(1.01) lev:(0.01) [135] conv:(1.29)  
 30. bluedis=ESTAVIV 9532 ==> movildis=REPOSO 9087 <conf:(0.95)> lift:(1.01) lev:(0) [50] conv:(1.11)  
 31. movildis=REPOSO humdis=BAJO 10016 ==> temdis=ALTO movdis=NDETMOV 9548 <conf:(0.95)> lift:(1) lev:(0) [36] conv:(1.08)  
 32. humdis=BAJO 10405 ==> movildis=REPOSO temdis=ALTO 9917 <conf:(0.95)> lift:(1.01) lev:(0.01) [130] conv:(1.26)  
 33. temdis=ALTO bluedis=ESTAVIV 9461 ==> movildis=REPOSO 9017 <conf:(0.95)> lift:(1.01) lev:(0) [47] conv:(1.1)  
 34. humdis=BAJO 10405 ==> temdis=ALTO movdis=NDETMOV 9908 <conf:(0.95)> lift:(1) lev:(0) [27] conv:(1.05)  
 35. movildis=REPOSO 13418 ==> temdis=ALTO movdis=NDETMOV 12772 <conf:(0.95)> lift:(1) lev:(0) [29] conv:(1.04)  
 36. temdis=ALTO movdis=NDETMOV 13440 ==> movildis=REPOSO 12772 <conf:(0.95)> lift:(1) lev:(0) [29] conv:(1.04)  
 37. movdis=NDETMOV 13546 ==> movildis=REPOSO 12872 <conf:(0.95)> lift:(1) lev:(0) [29] conv:(1.04)  
 38. movdis=NDETMOV bluedis=ESTAVIV 8958 ==> movildis=REPOSO temdis=ALTO 8510 <conf:(0.95)> lift:(1.01) lev:(0.01) [84] conv:(1.19)  
 39. temdis=ALTO 14041 ==> movildis=REPOSO 13312 <conf:(0.95)> lift:(1) lev:(0) [0] conv:(1)  
 40. bluedis=ESTAVIV 9532 ==> movildis=REPOSO temdis=ALTO 9017 <conf:(0.95)> lift:(1.01) lev:(0) [51] conv:(1.1)  
 41. movildis=REPOSO temdis=ALTO bluedis=ESTAVIV 9017 ==> movdis=NDETMOV 8510 <conf:(0.94)> lift:(0.99) lev:(-0.01) [-120] conv:(0.76)  
 42. movildis=REPOSO bluedis=ESTAVIV 9087 ==> movdis=NDETMOV 8574 <conf:(0.94)> lift:(0.99) lev:(-0.01) [-123] conv:(0.76)  
 43. movdis=NDETMOV 13546 ==> movildis=REPOSO temdis=ALTO 12772 <conf:(0.94)> lift:(1) lev:(0) [30] conv:(1.04)  
 44. temdis=ALTO bluedis=ESTAVIV 9461 ==> movdis=NDETMOV 8893 <conf:(0.94)> lift:(0.98) lev:(-0.01) [-162] conv:(0.71)  
 45. bluedis=ESTAVIV 9532 ==> movdis=NDETMOV 8958 <conf:(0.94)> lift:(0.98) lev:(-0.01) [-165] conv:(0.71)  
 46. temdis=ALTO lumdis=LUZ 9282 ==> movildis=REPOSO 8698 <conf:(0.94)> lift:(0.99) lev:(-0.01) [-101] conv:(0.82)  
 47. temdis=ALTO lumdis=LUZ 9282 ==> movdis=NDETMOV 8698 <conf:(0.94)> lift:(0.98) lev:(-0.01) [-185] conv:(0.68)  
 48. lumdis=LUZ 9358 ==> movildis=REPOSO 8769 <conf:(0.94)> lift:(0.99) lev:(-0.01) [-103] conv:(0.82)  
 49. lumdis=LUZ 9358 ==> movdis=NDETMOV 8768 <conf:(0.94)> lift:(0.98) lev:(-0.01) [-188] conv:(0.68)  
 50. movildis=REPOSO bluedis=ESTAVIV 9087 ==> temdis=ALTO movdis=NDETMOV 8510 <conf:(0.94)> lift:(0.99) lev:(-0.01) [-119]

Ilustración 76 Resultados devueltos por el algoritmo Apriori

Antecedente	Consecuente	Confianza
{{predis,BAJO}}	{{temdis,ALTO}}	1
{{(movdis,NDETMOV),(bluedis,ESTAVIV)}}	{{temdis,ALTO}}	0,99
{{(bluedis,ESTAVIV)}}	{{temdis,ALTO}}	0,99
{{(movildis,REPOSO),(movdis,NDETMOV),(bluedis,ESTAVIV)}}	{{temdis,ALTO}}	0,99
{{(movildis,REPOSO),(bluedis,ESTAVIV)}}	{{temdis,ALTO}}	0,99
{{(movildis,REPOSO),(movdis=NDETMOV)}}	{{temdis,ALTO}}	0,99
{{(movdis,NDETMOV)}}	{{temdis,ALTO}}	0,99
{{(movildis,REPOSO)}}	{{temdis,ALTO}}	0,99
{{(lumdis,LUZ),(movdis,NDETMOV)}}	{{temdis,ALTO}}	0,99
{{(movildis,REPOSO),(lumdis,LUZ)}}	{{temdis,ALTO}}	0,99
{{(lumdis,LUZ)}}	{{temdis,ALTO}}	0,99
{{(movildis,REPOSO),(humdis,BAJO),(movdis,NDETMOV)}}	{{temdis,ALTO}}	0,99
{{(movildis,REPOSO),(humdis,BAJO)}}	{{temdis,ALTO}}	0,99
{{(humdis,BAJO),(movdis,NDETMOV)}}	{{temdis,ALTO}}	0,99

{{(humdis,BAJO)}	{{(temdis,ALTO)}	0,99
{{(humdis,BAJO),(temdis,ALTO),(movdis,NDETMOV)}	{{(movildis,REPOSO)}	0,96
{{(humdis,BAJO),(movdis,NDETMOV)}	{{(movildis,REPOSO)}	0,96
{{(humdis,BAJO),(temdis,ALTO)}	{{(movildis,REPOSO)}	0,96
{{(movildis,REPOSO),(humdis,BAJO),(temdis,ALTO)}	{{(movdis,NDETMOV)}	0,96
{{(humdis,BAJO)}	{{(movildis,REPOSO)}	0,96
{{(movildis,REPOSO),(humdis,BAJO)}	{{(movdis,NDETMOV)}	0,96
{{(humdis,BAJO),(temdis,ALTO)}	{{(movdis,NDETMOV)}	0,96
{{(humdis,BAJO)}	{{(movdis,NDETMOV)}	0,96
{{(movildis,REPOSO),(temdis,ALTO)}	{{(movdis,NDETMOV)}	0,96
{{(movildis,REPOSO)}	{{(movdis,NDETMOV)}	0,96
{{(temdis,ALTO)}	{{(movdis,NDETMOV)}	0,96
{{(movdis,NDETMOV),(bluedis,ESTAVIV)}	{{(movildis,REPOSO)}	0,96
{{(temdis,ALTO),(movdis,NDETMOV),(bluedis,ESTAVIV)}	{{(movildis,REPOSO)}	0,96
{{(humdis,BAJO),(movdis,NDETMOV)}	{{(movildis,REPOSO),(temdis,ALTO)}	0,95
{{(bluedis,ESTAVIV)}	{{(movildis,REPOSO)}	0,95
{{(movildis,REPOSO),(humdis,BAJO)}	{{(temdis,ALTO),(movdis=NDETMOV)}	0,95
{{(humdis,BAJO)}	{{(movildis,REPOSO),(temdis,ALTO)}	0,95
{{(temdis,ALTO),(bluedis,ESTAVIV)}	{{(movildis,REPOSO)}	0,95
{{(humdis,BAJO)}	{{(temdis,ALTO),(movdis=NDETMOV)}	0,95
{{(movildis,REPOSO)}	{{(temdis,ALTO),(movdis=NDETMOV)}	0,95
{{(movdis,NDETMOV)}	{{(movildis,REPOSO)}	0,95
{{(movdis,NDETMOV),(bluedis,ESTAVIV)}	{{(movildis,REPOSO),(temdis,ALTO)}	0,95
{{(temdis,ALTO)}	{{(movildis,REPOSO)}	0,95
{{(bluedis,ESTAVIV)}	{{(movildis,REPOSO)}	0,95
{{(movildis,REPOSO),(temdis,ALTO),(bluedis,ESTAVIV)}	{{(movdis,NDETMOV)}	0,94
{{(movildis,REPOSO),(bluedis,ESTAVIV)}	{{(movdis,NDETMOV)}	0,94
{{(movdis,NDETMOV)}	{{(movildis,REPOSO),(temdis,ALTO)}	0,94
{{(temdis,ALTO),(bluedis,ESTAVIV)}	{{(movdis,NDETMOV)}	0,94
{{(bluedis,ESTAVIV)}	{{(movdis,NDETMOV)}	0,94
{{(temdis,ALTO),(lumdis,LUZ)}	{{(movildis,REPOSO)}	0,94
{{(temdis,ALTO),(lumdis,LUZ)}	{{(movdis,NDETMOV)}	0,94
{{(lumdis,LUZ)}	{{(movildis,REPOSO)}	0,94
{{(lumdis,LUZ)}	{{(movdis,NDETMOV)}	0,94
{{(movildis,REPOSO),(bluedis,ESTAVIV)}	{{(temdis,ALTO),(movdis=NDETMOV)}	0,94

Tabla 8 Resultados del algoritmo Apriori

1. [lumdis=NLUZ]: 4795 ==> [movdis=NDETMOV]: 4778 <conf:(1)> lift:(1.04) lev:(0.01) conv:(11.43)
2. [movildis=REPOSO, lumdis=NLUZ]: 4649 ==> [movdis=NDETMOV]: 4632 <conf:(1)> lift:(1.04) lev:(0.01) conv:(11.08)
3. [bluedis=NESTAVIV]: 4621 ==> [movdis=NDETMOV]: 4588 <conf:(0.99)> lift:(1.04) lev:(0.01) conv:(5.83)
4. [movildis=REPOSO, bluedis=NESTAVIV]: 4331 ==> [movdis=NDETMOV]: 4298 <conf:(0.99)> lift:(1.04) lev:(0.01) conv:(5.46)
5. [lumdis=NLUZ]: 4795 ==> [movildis=REPOSO]: 4649 <conf:(0.97)> lift:(1.02) lev:(0.01) conv:(1.69)
6. [movdis=NDETMOV, lumdis=NLUZ]: 4778 ==> [movildis=REPOSO]: 4632 <conf:(0.97)> lift:(1.02) lev:(0.01) conv:(1.69)
7. [lumdis=NLUZ]: 4795 ==> [movdis=NDETMOV, movildis=REPOSO]: 4632 <conf:(0.97)> lift:(1.06) lev:(0.02) conv:(2.65)
8. [movdis=NDETMOV, humdis=BAJO, predis=BAJO]: 5411 ==> [movildis=REPOSO]: 5222 <conf:(0.97)> lift:(1.02) lev:(0.01) conv:(1.48)
9. [movdis=NDETMOV, humdis=BAJO]: 10007 ==> [movildis=REPOSO]: 9641 <conf:(0.96)> lift:(1.02) lev:(0.01) conv:(1.42)
10. [humdis=BAJO, predis=BAJO]: 5660 ==> [movildis=REPOSO]: 5452 <conf:(0.96)> lift:(1.02) lev:(0.01) conv:(1.41)
11. [humdis=BAJO]: 10405 ==> [movildis=REPOSO]: 10016 <conf:(0.96)> lift:(1.02) lev:(0.01) conv:(1.39)
12. [movildis=REPOSO, humdis=BAJO]: 10016 ==> [movdis=NDETMOV]: 9641 <conf:(0.96)> lift:(1.01) lev:(0) conv:(1.14)
13. [humdis=BAJO]: 10405 ==> [movdis=NDETMOV]: 10007 <conf:(0.96)> lift:(1) lev:(0) conv:(1.12)
14. [movildis=REPOSO]: 13418 ==> [movdis=NDETMOV]: 12872 <conf:(0.96)> lift:(1) lev:(0) conv:(1.05)
15. [movildis=REPOSO, humdis=BAJO, predis=BAJO]: 5452 ==> [movdis=NDETMOV]: 5222 <conf:(0.96)> lift:(1) lev:(0) conv:(1.01)
16. [humdis=BAJO, predis=BAJO]: 5660 ==> [movdis=NDETMOV]: 5411 <conf:(0.96)> lift:(1) lev:(0) conv:(0.97)
17. [movildis=REPOSO, predis=BAJO]: 8134 ==> [movdis=NDETMOV]: 7747 <conf:(0.95)> lift:(1) lev:(0) conv:(0.9)
18. [movdis=NDETMOV]: 13546 ==> [movildis=REPOSO]: 12872 <conf:(0.95)> lift:(1) lev:(0) conv:(1.04)
19. [predis=BAJO]: 8632 ==> [movdis=NDETMOV]: 8190 <conf:(0.95)> lift:(0.99) lev:(-0.01) conv:(0.84)
20. [movdis=NDETMOV, predis=BAJO]: 8190 ==> [movildis=REPOSO]: 7747 <conf:(0.95)> lift:(1) lev:(0) conv:(0.96)
21. [predis=BAJO]: 8632 ==> [movildis=REPOSO]: 8134 <conf:(0.94)> lift:(0.99) lev:(0) conv:(0.9)
22. [bluedis=NESTAVIV]: 4621 ==> [movildis=REPOSO]: 4331 <conf:(0.94)> lift:(0.99) lev:(0) conv:(0.82)
23. [movdis=NDETMOV, bluedis=NESTAVIV]: 4588 ==> [movildis=REPOSO]: 4298 <conf:(0.94)> lift:(0.99) lev:(0) conv:(0.82)
24. [bluedis=NESTAVIV]: 4621 ==> [movdis=NDETMOV, movildis=REPOSO]: 4298 <conf:(0.93)> lift:(1.02) lev:(0.01) conv:(1.29)
25. [lumdis=BAJO]: 10405 ==> [movdis=NDETMOV, movildis=REPOSO]: 9641 <conf:(0.93)> lift:(1.02) lev:(0.01) conv:(1.23)
26. [humdis=BAJO, predis=BAJO]: 5660 ==> [movdis=NDETMOV, movildis=REPOSO]: 5222 <conf:(0.92)> lift:(1.01) lev:(0.01) conv:(1.17)

Ilustración 77 Resultados obtenidos con el algoritmo FP-Growth

Antecedente	Consecuente	Confianza
{(lumdis,NLUZ)}	{(movdis,NDETMOV)}	1
{(movildis,REPOSO),(lumdis,NLUZ)}	{(movdis,NDETMOV)}	1
{(bluedis,ESTAVIV)}	{(movdis,NDETMOV)}	0,99
{(movildis,REPOSO),(bluedis,ESTAVIV)}	{(movdis,NDETMOV)}	0,99
{(lumdis,NLUZ)}	{(movildis,REPOSO)}	0,97
{(movdis,NDETMOV),(lumdis,NLUZ)}	{(movildis,REPOSO)}	0,97
{(lumdis,NLUZ)}	{(movdis,NDETMOV),(movildis,REPOSO)}	0,97
{(movdis,NDETMOV),(humdis,BAJO),(predis,BAJO)}	{(movildis,REPOSO)}	0,97
{(movdis,NDETMOV),(humdis,BAJO)}	{(movildis,REPOSO)}	0,96
{(humdis,BAJO),(predis,BAJO)}	{(movildis,REPOSO)}	0,96
{(humdis,BAJO)}	{(movildis,REPOSO)}	0,96
{(humdis,BAJO),(movildis,REPOSO)}	{(movdis,NDETMOV)}	0,96
{(humdis,BAJO)}	{(movdis,NDETMOV)}	0,96
{(movildis,REPOSO)}	{(movdis,NDETMOV)}	0,96
{(movildis,REPOSO),(humdis,BAJO),(predis,BAJO)}	{(movdis,NDETMOV)}	0,96
{(humdis,BAJO),(predis,BAJO)}	{(movdis,NDETMOV)}	0,96
{(movildis,REPOSO),(predis,BAJO)}	{(movdis,NDETMOV)}	0,95
{(movdis,NDETMOV)}	{(movildis,REPOSO)}	0,95
{(predis,BAJO)}	{(movdis,NDETMOV)}	0,95
{(movdis,NDETMOV),(predis,BAJO)}	{(movildis,REPOSO)}	0,95
{(predis,BAJO)}	{(movildis,REPOSO)}	0,95
{(bluedis,NESTAVIV)}	{(movildis,REOSO)}	0,94
{(movdis,NDETMOV),(bluedis,NESTAVIV)}	{(movildis,REOSO)}	0,94
{(bluedis,NESTAVIV)}	{(movdis,NDETMOV),(movildis,REPOSO)}	0,93
{(humdis,BAJO)}	{(movdis,NDETMOV),(movildis,REPOSO)}	0,93
{(humdis,BAJO),(predis,BAJO)}	{(movdis,NDETMOV),(movildis,REPOSO)}	0,92

Tabla 9 Resultados del algoritmo FP-Growth

Como podemos ver en las ilustraciones superiores, se demuestra que el algoritmo Apriori es más ineficiente y genera muchas más reglas, muy redundantes, con respecto a las generadas por el algoritmo FP-Growth con los mismos parámetros de configuración. Por lo tanto, utilizaré los resultados devueltos por este último algoritmo. A continuación, se muestra un listado con las reglas obtenidas tras eliminar aquellas que son redundantes:

- Si el sensor no detecta luz, no se detectará movimiento.
- Si el móvil se encuentra en reposo, no se detecta luz.
- Si el sensor de humedad marca un nivel bajo, no se detecta movimiento
- Si el sensor de presión marca un nivel bajo, no se detecta movimiento.
- Si el usuario no está en la vivienda, no se detecta movimiento.

Por los parámetros establecidos, se puede asegurar que estas reglas se cumplen en al menos el 90% de las transacciones y estas reglas se dan, al menos, con una frecuencia del 30%.

## Capítulo 6. Conclusiones y trabajo futuro

La minería de datos es una etapa del análisis del conocimiento de las bases de datos (KDD). Hoy en día, con el avance de las tecnologías, se hace imprescindible el uso de técnicas de inteligencia artificial, aprendizaje automático y estadística sobre las bases de datos para poder convertir los grandes volúmenes de datos en conocimiento. Este conocimiento es aprovechado por las empresas para mejorar sus servicios al cliente y su productividad. A pesar de ello, no es un proceso trivial ya que hay que aplicar diferentes técnicas para preparar el conjunto de datos de forma correcta para obtener unos resultados de calidad. La preparación de los datos analizados ha sido un proceso más complejo y tedioso que la evaluación de los mismos.

Durante la creación del prototipo se han utilizado diferentes dispositivos para extracción de los datos que rodean a una persona. De cara a un futuro, cabe la idea de incluir una mayor cantidad de sensores distribuidos por toda la vivienda, así como una serie de dispositivos que recojan los datos del exterior de la misma para poder crear un conjunto de datos más amplio. Sería interesante controlar el estado de todas las habitaciones, así como la posición del usuario dentro de la vivienda. Además, la limitación que supone la recogida de datos desde el dispositivo móvil por la eficiencia energética, se podría solucionar utilizando una pulsera inteligente que se encargue de monitorizar al usuario de forma constante durante todo el día. Esto también permitiría obtener datos como la frecuencia cardíaca y el movimiento del individuo de una forma más precisa respecto con los datos obtenidos del teléfono móvil.

Al ser un prototipo, el servidor sólo se encarga de comprobar el estado y actualizar las reglas de asociación obtenidas teniendo en cuenta los datos de las últimas 24 horas. Sería interesante plantear, en una versión final, la opción de utilizar una base de datos operacional y tener en cuenta una mayor cantidad de datos a la hora de evaluar estas reglas. Al tener una mayor cantidad de datos puede que también fuera necesario cambiar el sistema de gestión de base de datos por uno no relacional como el comentado en el apartado de análisis de tecnologías utilizadas en el proyecto, MongoDB [18], obteniendo así un mayor rendimiento y eficacia en las consultas.

Aunque las reglas de asociación entre conjuntos se utilizan principalmente sobre transacciones, hemos podido comprobar que también son de utilidad a la hora de analizar la relación existente entre otros tipos de conjuntos de datos. A pesar de ello, debido principalmente a la limitación existente al ser un Trabajo Fin de Máster, no he obtenido los resultados esperados en un principio al evaluar los datos recogidos. Uno de los objetivos, por el que me he centrado en el entorno que rodea a una persona dentro de una vivienda, era obtener información sobre vida cotidiana y costumbres de un individuo para comenzar a sentar las bases de una herramienta que se encargara de ayudar a las personas en su día a día. Aunque los sensores me han aportado gran cantidad de información, los resultados no han arrojado ninguna información que pueda ser relevante y que permita desarrollar en un futuro un sistema que permita mejorar los hábitos de las personas. Se podría solucionar aumentando la cantidad de sensores y datos recogidos.

Por último, al analizar los datos se ha visto de manera clara la superioridad del algoritmo FP-Growth frente al Apriori a la hora de generar los conjuntos candidatos para crear las reglas de asociación.

## Capítulo 7. Bibliografía

- [1] I. H. Witten, E. Frank, M. A. Hall y C. J. Pal, Data Mining: Practical Machine Learning. Tools and Techniques, Morgan Kaufmann, 2017.
- [2] Raspberry Pi Foundation, [En línea]. Available: <https://www.raspberrypi.org/>. [Último acceso: 20 Abril 2018].
- [3] T. I. ., A. S. R. Agrawal, «Mining associations between sets of items in large databases,» 1993.
- [4] J. Han, J. Pei y Y. Yin, «Mining frequent patterns without candidate generation,» 2000.
- [5] J. Cano, Business intelligence, competir con información, 2012.
- [6] T. Shafer, «Elder Reseach,» [En línea]. Available: <https://www.elderresearch.com/blog/42-v-of-big-data>. [Último acceso: 18 Mayo 2018].
- [7] J. ., G. J. Molina, Técnicas de análisis de datos: Aplicaciones prácticas utilizando Microsoft Excel y Weka, 2006.
- [8] A. P. Alonso, *Mantenimiento incremental de reglas de asociación y sus extensiones mediante bases de datos activas*.
- [9] C. Zhang y S. Zhang, Association rule mining: models and algorithms., 2002.
- [10] R. Merkle, « Secrecy, authentication and public key systems,» 1979.
- [11] J. Torres y C. Abad, «Análisis comparativo de mecanismos de minería de datos,» 2015.
- [12] R. Agarwal, C. Aggarwal y V. Prasad, «A tree projection algorithm for generation of frequent itemsets. Journal of Parallel and Distributed Computing,» 2001.
- [13] L. Zadeh, «Fuzzy sets,» 1965.
- [14] T. Hong y Y. Lee, «An overview of mining fuzzy association rules,» 2008.
- [15] M. Delgado, N. Marín y D. Sánchez, «Fuzzy association rules: General model and applications,» pp. 214-225, 2003.

- [16] Arduino, «Arduino,» [En línea]. Available: <https://www.arduino.cc/>. [Último acceso: 2018 Marzo 12].
- [17] Oracle Corporation, «MySQL,» [En línea]. Available: <https://www.mysql.com/>. [Último acceso: 20 Abril 2018].
- [18] MongoDB, «MongoDB,» [En línea]. Available: <https://www.mongodb.com/>. [Último acceso: 12 Marzo 2018].
- [19] British Columbia Institute of Technology, «CodeIgniter,» [En línea]. Available: <https://codeigniter.com/>. [Último acceso: 15 Abril 2018].
- [20] «W3Tech,» [En línea]. Available: <https://w3techs.com/technologies/details/pl-php/all/all>. [Último acceso: 22 Enero 2018].
- [21] Python Software Foundation, «Python,» [En línea]. Available: <http://www.python.org/>. [Último acceso: 18 Marzo 2018].
- [22] TIOBE, «TIOBE,» [En línea]. Available: <https://www.tiobe.com/tiobe-index/>. [Último acceso: 1 Junio 2018].
- [23] Bootstrap, «Bootstrap,» [En línea]. Available: <https://getbootstrap.com/>. [Último acceso: 05 Marzo 2018].
- [24] «JetBrains,» [En línea]. Available: <https://www.jetbrains.com/>. [Último acceso: 20 Mayo 2018].
- [25] S. Tatham. [En línea]. Available: <https://putty.org/>. [Último acceso: 15 Abril 2018].
- [26] T. Kosse, «FileZilla,» [En línea]. Available: <https://filezilla-project.org/>. [Último acceso: 10 Marzo 2018].
- [27] C. ZHOU, «RESTClient,» [En línea]. Available: <https://addons.mozilla.org/es/firefox/addon/restclient/>. [Último acceso: 15 Abril 2018].
- [28] Adobe Systems Software, «Adobe Photoshop,» [En línea]. Available: <https://www.adobe.com/es/products/photoshop.html>. [Último acceso: 15 Mayo 2018].

- [29] Visual Paradigm, «Visual Paradigm,» [En línea]. Available: <https://www.visual-paradigm.com/>. [Último acceso: 2018 Abril 19].
- [30] KNIME, «KNIME,» [En línea]. Available: <https://www.knime.com/>. [Último acceso: 12 Mayo 2018].
- [31] Proto.io, «Proto.io,» [En línea]. Available: <https://proto.io/>. [Último acceso: 16 Mayo 2018].
- [32] STRATO, «STRATO,» [En línea]. Available: <https://www.strato.es>. [Último acceso: 25 Mayo 2018].
- [33] Amazon, «Amazon,» [En línea]. Available: <https://www.amazon.es/>. [Último acceso: 2018 03 12].
- [34] El androide libre S.L., «El androide libre,» [En línea]. Available: <https://elandroidelibre.elespanol.com/2017/02/vida-media-movil-espana-20-meses.html>. [Último acceso: 2018 Mayo 20].
- [35] Ecosistemas Digitales de Negocio S.L., «Cuéntica,» [En línea]. Available: <https://cuentica.com/asesoria/tabla-anos-porcentajes-amortizacion-simplificada-autonomos-y-profesionales/>. [Último acceso: 15 Mayo 2018].
- [36] Ministerio de Trabajo e Inmigración, «Resolución de 18 de marzo de 2009, de la Dirección General de Trabajo, por la que se registra y publica el XVI Convenio colectivo estatal de empresas de consultoría y estudios de mercado y de la opinión pública,» 4 Abril 2009. [En línea]. Available: <https://www.boe.es/boe/dias/2009/04/04/pdfs/BOE-A-2009-5688.pdf>. [Último acceso: 23 Mayo 2018].
- [37] Jrbail01, «Weather API GitHub Project,» [En línea]. Available: <https://github.com/initialstate/wunderground-sensehat/wiki/Part-3.-Sense-HAT-Temperature-Correction>. [Último acceso: 12 04 2018].
- [38] Android, «Sensor Manager,» [En línea]. Available: <https://developer.android.com/reference/android/hardware/SensorManager>. [Último acceso: 5 Abril 2018].
- [39] Android, «Volley,» [En línea]. Available: <https://developer.android.com/training/volley/>. [Último acceso: 2 Mayo 2018].

- [40] G. Welch y G. Bishop, «An Introduction to the Kalman Filter,» 2006.
- [41] D. Benito, «La Rioja meteo,» [En línea]. Available: <http://lariojameteo.es/2017/09/presion-atmosferica-que-son-anticiclones-y-borrascas/>. [Último acceso: 20 Mayo 2018].
- [42] Universidad de Jaén, «Meteorología en Jaén,» [En línea]. Available: <http://www.ujaen.es/dep/fisica/estacion/estacion3.htm>. [Último acceso: 02 Junio 2018].
- [43] R. M. D. S. D. Delgado M., «New approaches for discovering exception and anomalous rules,» pp. 361-399, 2011.
- [44] «Gartner, Inc.,» [En línea]. Available: <https://www.gartner.com/en>. [Último acceso: 15 Abril 2018].
- [45] Logicalis, «Logicalis,» [En línea]. Available: <https://blog.es.logicalis.com/>. [Último acceso: 20 Mayo 2018].
- [46] S. Barik, D. Mishra y S. Mishra, «Pattern Discovery using Fuzzy FP-growth Algorithm from Gene Expression Data,» 2010.
- [47] M. Delgado, M. Ruiz y D. Sánchez, «A formal model for mining fuzzy rules using the RL representation theory,» pp. 5194-5213, 2011.
- [48] A. Fernández, C. Carmona y M. del Jesus, «A View on Fuzzy Systems for Big Data: Progress and Opportunities,» pp. 69-80, 2016.
- [49] C. Fernandez, M. Ruiz y M. M., «Extraction of Association Rules using Big Data technologies,» pp. 178-185, 2016.

## Anexo I Material adjunto

En este apartado se describen todos los archivos que se adjuntan con la memoria del proyecto.

- sendata.py → script en Python que contiene el código que se ejecuta en la Raspberry Pi para la recolección y envío de datos al servidor.
- sensores.zip → archivo comprimido que contiene la aplicación desarrollada en Android Studio para recoger y enviar datos al servidor desde un dispositivo móvil. Dentro del archivo en la carpeta “app” se encuentra el archivo. apk para su instalación en dispositivos Android.
- codeigniter.zip → contiene la aplicación web. En la carpeta plantilla se encuentra la plantilla de bootstrap, en la ruta “application/controllers” se encuentran los controladores, en la ruta “application/models” se encuentra el modelo de la base de datos y en “application/views” se encuentran las vistas de la aplicación web.
- knime.zip → contiene el proyecto de KNIME con modelo desarrollado.
- sensoresProc.xlsx → contiene los datos procesados que se han analizado.

Además, se puede acceder a la web a través del siguiente enlace:

<http://www.fdperez.es>

## Anexo II Guía de instalación y de uso

Para poder lanzar el script de la Raspberry Pi es necesario conectar los sensores tal y como se comenta en el apartado de implementación. También hay que instalar Python3. Para lanzar el script basta con utilizar el siguiente comando

```
Python3 senddata.py
```

Para instalar la aplicación en un dispositivo Android basta con lanzar el archivo con la extensión apk. La interfaz de la aplicación consta de dos botones, como se ha visto en la fase de implementación, para activar y desactivar el envío de datos al servidor.

Para instalar la aplicación web basta con disponer de un servidor con PHP y MySQL instalados. Habría que configurar CodeIgniter adaptándolo al nuevo servidor para que funcione de forma correcta. La interfaz de la aplicación web solo cuenta con un menú lateral que se encarga de facilitar la visualización de los datos que componen la vista. También se podrá acceder a la aplicación a través de <http://www.fdperez.es>

Para lanzar el proyecto en KNIME hace falta activar el plugin de WEKA porque se utiliza para la extracción de las reglas de asociación.