

eman ta zabal zazu



Universidad del País Vasco Euskal Herriko Unibertsitatea

Grado en Ingeniería Informática

Ingeniería del Software

Trabajo de Fin de Grado

---

# Smartdevices aplicados a entornos industriales 4.0

---

Autor

Carlos Adalberto Zamora Soliz

Director

José Ángel Vadillo Zorita

Noviembre, 2016

## **AGRADECIMIENTOS**

Gracias principalmente a todas las personas que de una u otra manera han influido y participado en el desarrollo de este proyecto. Quiero agradecer a José Ángel Vadillo por ser mi director del TFG y darme las pautas necesarias para llevar a cabo este trabajo. A todos los profesores de la Facultad de Informática por guiarme y aportarme tantos conocimientos durante los últimos años.

A la empresa Innovae Augmented Reality por darme la oportunidad de desarrollar el proyecto en sus instalaciones y permitirme aprender de los profesionales que en ella trabajan. A mis compañeros en Innovae por dejarme ser parte del equipo y hacer las mañanas de trabajo muy divertidas.

Gracias a todos los compañeros y amigos conocidos durante mi etapa universitaria, porque sin ellos todos estos años de carrera no hubieran sido lo mismo.

Agradecimientos en especial a mis padres por apoyarme durante todos estos años, gracias a ellos he tenido la oportunidad de estar aquí y de haber podido estudiar lo que me gusta. También a mi hermano porque su compañía y ayuda hizo mucho más amena y divertida mi etapa en la universidad.

Agradecer también a Tamara por la corrección del documento.

---

## Resumen

---

El presente documento corresponde a la memoria del Trabajo de Fin de Grado, *Smartdevices* aplicados a entornos industriales 4.0, desarrollado para la obtención del Grado de Ingeniería Informática por la Facultad de Informática de la Universidad del País Vasco (UPV/EHU).

Este proyecto se ha llevado a cabo en la empresa Innovae Augmented Reality bajo la supervisión de Mikel Pajares y Pablo Ayala, y ha sido dirigido por José Ángel Vadillo Zorita, profesor de la Facultad de Informática de la UPV/EHU. Innovae Augmented Reality es una consultora tecnológica especializada en realidad aumentada, realidad virtual y aplicaciones móviles. Este proyecto tiene como objetivo mostrar el uso que pueden ofrecer los *smartdevices* (*smartwatches*, gafas de realidad aumentada, etc) aplicados a entornos industriales. La aplicación de estos dispositivos permitiría a los operarios acceder de manera ágil a información relativa a la planta o a ciertos procesos, minimizando el impacto en el desarrollo cotidiano de sus tareas. El proyecto ha sido desarrollado inicialmente durante mi estancia en prácticas en la empresa, como una *demo*, pero finalmente su desarrollo ha ido evolucionando e irá dirigido para su uso real en una empresa cliente de Innovae.

# Índice General

<b>1</b>	<b>Introducción</b>	<b>6</b>
<b>2</b>	<b>Antecedentes</b>	<b>7</b>
<b>3</b>	<b>Documento de Objetivos</b>	<b>8</b>
3.1	Objetivos . . . . .	8
3.2	Alcance . . . . .	8
3.2.1	Gafas de Realidad Aumentada . . . . .	10
3.2.2	Smartwatch . . . . .	10
3.3	Planificación . . . . .	11
3.3.1	EDT (Estructura de Descomposición del Trabajo) . . . . .	11
3.3.2	Entregables . . . . .	12
3.3.3	Descripción de tareas a realizar . . . . .	12
3.3.4	Dedicaciones estimadas . . . . .	15
3.3.5	Diagrama de Gantt . . . . .	16
3.4	Gestión de Riesgos . . . . .	17
<b>4</b>	<b>Tecnologías</b>	<b>18</b>
4.1	Gafas de Realidad Aumentada . . . . .	18
4.2	Relojes Inteligentes . . . . .	21
4.3	Industria 4.0 . . . . .	24
<b>5</b>	<b>Herramientas</b>	<b>27</b>
5.1	Desarrollo . . . . .	27
5.1.1	Android . . . . .	27
5.1.2	Android Studio . . . . .	29
5.1.3	Firebase Cloud Messaging . . . . .	30
5.2	Gestión . . . . .	31
5.2.1	Trello . . . . .	31
5.3	Dispositivos de prueba de la aplicación . . . . .	32
<b>6</b>	<b>Análisis de Requisitos</b>	<b>33</b>
6.1	Datos (Modelo del Dominio) . . . . .	33
6.2	Diagrama de Casos de Uso . . . . .	36
6.2.1	Diagrama Casos de Uso Smartwatch . . . . .	36
6.2.2	Diagrama Casos de Uso Gafas de Realidad Aumentada . . . . .	37
6.3	Flujo de Eventos . . . . .	38
6.3.1	Flujo de Eventos Smartwatch . . . . .	38
6.3.2	Flujo de Eventos Gafas de Realidad Aumentada . . . . .	47
6.4	Requisitos Funcionales . . . . .	54
6.4.1	Requisitos Funcionales Smartwatch . . . . .	54
6.4.2	Requisitos Funcionales Gafas de Realidad Aumentada . . . . .	54
6.5	Requisitos No Funcionales . . . . .	55
6.5.1	Requisitos No Funcionales Smartwatch . . . . .	55

6.5.2	Requisitos No Funcionales Gafas de Realidad Aumentada	55
<b>7</b>	<b>Diseño</b>	<b>56</b>
7.1	Diagramas de Secuencia Smartwatch	57
7.2	Diagramas de Secuencia Gafas de Realidad Aumentada	63
<b>8</b>	<b>Implementación</b>	<b>68</b>
8.1	Arquitectura del Sistema	68
8.1.1	Capa de Presentación	69
8.1.2	Capa de Negocio	74
8.1.3	Capa de Datos	74
8.2	Interconexiones del Sistema	75
8.3	Módulos del Sistema	76
<b>9</b>	<b>Pruebas</b>	<b>81</b>
9.1	Pruebas gafas de realidad aumentada	81
9.2	Pruebas reloj inteligente	84
<b>10</b>	<b>Gestión</b>	<b>88</b>
10.1	Comparación de Horas Estimadas y Horas Dedicadas	88
10.2	Desviaciones significativas del proyecto	88
10.2.1	Desviaciones de Diseño	88
10.2.2	Desviaciones de Implementación	89
10.2.3	Desviaciones de Pruebas	89
<b>11</b>	<b>Conclusiones</b>	<b>90</b>
11.1	Conclusiones del Trabajo	90
11.2	Conclusiones Personales	92
<b>12</b>	<b>Líneas Futuras</b>	<b>93</b>
<b>13</b>	<b>Bibliografía</b>	<b>95</b>
<b>14</b>	<b>Anexos</b>	<b>97</b>
14.1	Actas de Reunión: Director Proyecto	97
14.2	Actas de Reunión: Empresa	99

---

## 1 Introducción

---

La industria avanza a grandes pasos y con ella la idea de la anhelada 'fábrica del futuro' parece estar más cerca de lo que parece. La incorporación de tecnologías IT (Tecnologías de Información) en el sector industrial está marcando la diferencia para acelerar el camino hacia la Industria 4.0, basada en una fábrica inteligente y conectada en tiempo real.

Los tres pilares que impulsan esta industria inteligente son: *smartdevices* (dispositivos inteligentes), infraestructura de computación *cloud* y las tecnologías de análisis de información. De estos tres los dispositivos inteligentes son el corazón de la fábrica del futuro. La creación de una infraestructura digital en plantas de producción tiene como objetivo la introducción de *smartdevices* que puedan consumir dicha información y hacerla llegar a las personas interesadas, de esta manera nace un nuevo ecosistema basado en la integración de procesos, máquinas, aplicaciones y personas mejor conectadas.

El proyecto desarrollado tiene como finalidad ser utilizado en la empresa Tafisa, perteneciente al Grupo Sonae Industria, esta compañía dedica su actividad a la fabricación de tableros de madera y otros productos a partir de este material. La aplicación desarrollada para dicha empresa tiene como fin servir y dar soporte a sus operarios para que tengan la información de la fábrica más accesible. La empresa ya se encuentra embarcada en la industria 4.0, muchas de sus áreas se encuentran digitalizadas mediante servicios en la nube y las máquinas de la fábrica se encuentran conectadas con esta 'nube' de datos permitiéndoles obtener modelos que permitan hacer simulaciones y minimizar los costes. De esta manera gracias a este *cloud*, es posible el desarrollo de este proyecto cuyo fin será conectarse a su nube de datos y extraer los datos críticos que serán consumidos por los operarios, estos datos podrán ser alarmas que se muestren en momentos determinados o datos bajo demanda, que como su nombre indica, serán solicitados en cualquier momento por el propio operario. Estos datos podrán ser mostrados a los trabajadores a través de los dos *smartdevices* con los que se ha trabajado en el desarrollo de este proyecto: gafas de realidad aumentada y relojes inteligentes.

---

## 2 Antecedentes

---

La demanda actual de los consumidores en lo que a productos se refiere, está sufriendo un gran cambio en los últimos años. Cada vez requerimos productos más personalizados y que se adapten a nuestras necesidades de una manera sencilla. Lo que no se tiene en cuenta es que esta personalización supone un coste elevado para las empresas, ya que es más barato hacer el mismo producto con las mismas características una y otra vez.

Derivado de esto se entiende que uno de los factores que marcan el éxito de una empresa en la actualidad es la diferenciación frente a la competencia. Para ello y gracias a la llegada de las tecnologías digitales a la industria, las empresas están comenzando a incorporar nuevos recursos tecnológicos para reducir costes, mejorar los tiempos y conseguir mejorar la eficiencia y eficacia en toda la cadena de valor.

Aunque esta transformación industrial ya es una realidad, hoy en día siguen habiendo varias empresas que siguen utilizando los mismos métodos de producción, organización de procesos y metodologías de trabajo de antaño. La necesidad de apuntarse al carro de la Industria 4.0 no solo tiene como objetivo la innovación, sino que también trae muchas ventajas consigo para el desarrollo y la mejora de las empresas:

1. Se obtiene una gran flexibilidad en los procesos de producción y adaptabilidad a situaciones inesperadas, contribuyendo al aumento y mejora de la producción.
2. Personalización en la producción para las necesidades particulares de consumidores finales o clientes.
3. Se genera un gran y constante flujo de información, y la capacidad de transmisión de esta información tanto de forma interna como externa. Dicha información generada puede ser usada para estudio del mercado, trazabilidad, mejora de procesos, etc.
4. Creación de un 'gemelo digital' que permita simular y producir una réplica virtual de la totalidad o parte de la cadena de producción.

---

## 3 Documento de Objetivos

---

En este apartado se describen los objetivos, el alcance, una primera planificación del proyecto y el análisis y gestión de los riesgos.

### 3.1 Objetivos

El proyecto comenzó a desarrollarse como una prueba de concepto del posible uso de gafas de realidad aumentada en el ámbito industrial, se presentó una demo de prueba a la empresa Tafisa pero debido a temas de certificados el proyecto no pudo salir adelante con las gafas, en cambio vimos la posibilidad de sacar el mismo proyecto pero haciendo uso de un *smartwatch* (reloj inteligente). De todas maneras, ya que se tuvo la oportunidad de trabajar con ambos, en este trabajo se englobarán el desarrollo para los dos dispositivos. Por lo tanto el trabajo fin de grado consistirá en la implementación de las aplicaciones tanto para el reloj inteligente como para las gafas de realidad aumentada.

El proyecto tiene como principal objetivo demostrar el uso potencial de los smartdevices, tanto en el ámbito industrial como empresarial, no sólo como instrumentos de innovación, sino también como herramientas que realmente aporten valor a los trabajadores a la hora de realizar sus tareas. Esto se traducirá a su vez en una mejora en la eficiencia y eficacia de los procesos llevados a cabo en las empresas, y en su correspondiente distinción frente a la competencia.

### 3.2 Alcance

La aplicación del reloj inteligente, desarrollada para su uso en la empresa Tafisa, consistirá en una aplicación diseñada para visualizar datos a través del *smartwatch* pero el núcleo de la aplicación residirá en un teléfono móvil. Será pues el smartphone quien esté en conexión con el servidor mediante Wi-fi o 3G y con el *smartwatch* mediante Bluetooth. En el smartphone el usuario dispondrá de una aplicación, que tras iniciarla permitirá configurar si se quieren visualizar los datos asociados al usuario 1 o al usuario 2. Una vez seleccionado, en el *smartwatch* se podrán visualizar los siguientes datos:

- **Datos de Alarmas**

Por defecto la aplicación estará siempre a la “escucha” de alarmas. En caso de llegar una, se visualizará de manera inmediata en la pantalla del

reloj. Nada más llegar una alarma, el *smartwatch* vibrará, mostrará dicha alarma y a continuación el usuario podrá marcarla como leída, en cuyo caso dejaría de aparecer en primer plano, pero se añadiría a una lista de alarmas activas que el usuario puede consultar cuando desee. En el momento que la alarma es leída se notificará al servidor que se ha llevado a cabo dicha acción.

Cuando la alarma dejase de estar activa en el servidor, desaparecería de la lista de alarmas activas del *smartwatch*. En caso de llegar varias alarmas al mismo tiempo, se encolarán por nivel de criticidad (tres niveles: alarmas, warnings e información, siendo las alarmas el más crítico e información el menos crítico), mostrándose primero las alarmas, después los warnings, y por último las alarmas de información.

- **Datos Bajo Demanda**

La aplicación constará de otra sección que le permitirá al usuario acceder a datos bajo demanda. En concreto se han identificado tres grandes bloques de datos (resumen turno, valores actuales e información de producción). Estos estarán además subdivididos en grupos para poder ser visualizados de una manera más cómoda y ordenada en la interfaz del *smartwatch*.

- **Datos Google Calendar**

Los teléfonos móviles tendrán una cuenta de google vinculada y la aplicación que se implemente permitirá recibir en el reloj las notificaciones o recordatorios del calendario vinculado a esa cuenta.

Por otro lado la aplicación de las gafas también se utilizará para visualizar datos en las lentes de la misma. Al igual que en el reloj, la inteligencia de la aplicación residirá en el smartphone, ya que éste será el encargado de conectarse con el servidor mediante Wi-fi o 3G y con las gafas mediante Bluetooth. Las funcionalidades implementadas en las gafas son:

- **Datos Bajo Demanda**

Esta funcionalidad es la misma que la descrita anteriormente para el reloj, solo que en vez de visualizarse los datos en la pantalla del mismo, se visualizarán en las lentes de las gafas de realidad aumentada.

- **Modo Cámara**

Las gafas de realidad aumentada poseen una cámara incorporada, esta funcionalidad permitirá que el usuario capture fotos desde la cámara de las gafas y que se envíen a un ordenador donde podrán ser archivadas y visualizadas.

El proyecto se divide en dos apartados: una de estudio, desarrollo y pruebas con las gafas de realidad aumentada, y otra con los mismos apartados pero enfocándose en el *smartwatch*. Cabe destacar que los temas de conexión con el

servicio en la nube fueron llevados a cabo por un compañero de Innovae, mi trabajo ha sido desarrollar los front-ends del *smartwatch* y de las gafas de realidad aumentada, y encargarme de hacer las llamadas al API que se encarga de obtener los datos desde el backend. Por tanto los objetivos quedarán resumidos de la siguiente manera:

### **3.2.1 Gafas de Realidad Aumentada**

- Estudio de API y librería de las gafas de realidad aumentada.
- Desarrollo de la aplicación para las gafas de realidad aumentada.
- Pruebas de funcionamiento de la aplicación.

### **3.2.2 Smartwatch**

- Estudio de API y librerías del *smartwatch*.
- Desarrollo de la aplicación para el reloj.
- Pruebas de funcionamiento de la aplicación.

### 3.3 Planificación

En este apartado se explica todo lo que comprende la planificación del proyecto, se muestra la estructura de descomposición del trabajo, los entregables, la descripción de cada una de las tareas, con sus estimaciones y su respectivo Diagrama de Gantt.

#### 3.3.1 EDT (Estructura de Descomposición del Trabajo)

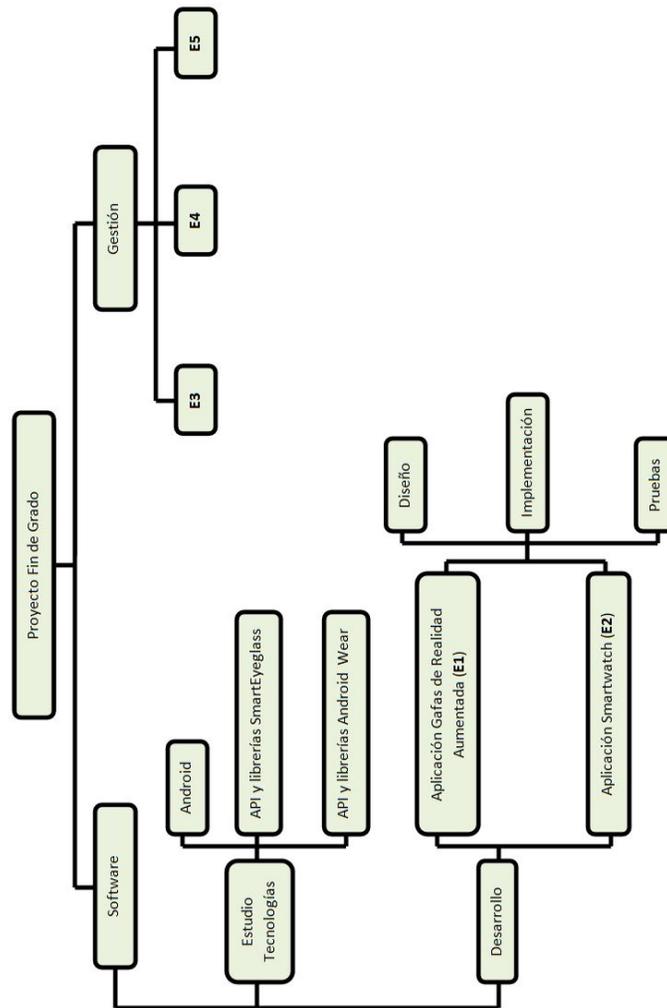


Figure 1: EDT

### 3.3.2 Entregables

Los entregables se dividirán en los propios del desarrollo del proyecto y los relacionados con la gestión del mismo.

- **Entregables propios del proyecto**
  - E1: Aplicación para las gafas de realidad aumentada.
  - E2: Aplicación para el reloj inteligente.
- **Entregables gestión del proyecto**
  - E3: Planificación.
  - E4: Seguimiento y Control.
  - E5: Memoria.

### 3.3.3 Descripción de tareas a realizar

- **Desarrollo de la aplicación para las gafas de realidad aumentada**
  - **Tarea 1.1: Diseño de la interfaz en las gafas**

En la página web de Sony se muestran una serie de guías de diseño para las aplicaciones que se crean para las *SmartEyeglass*, al llevar a cabo el diseño de la interfaz de la aplicación se tuvieron en cuenta estas guías para facilitar la interacción y consistencia con la aplicación realizada. La interfaz de las aplicaciones en el *SmartEyeglass* presentan una estructura de tarjetas que a su vez se estructuran en capas, nos podemos mover de izquierda a derecha entre tarjetas o adentrarnos en las capas. Cabe destacar que el diseño de la interfaz de las gafas lo decidí yo mismo y no tuve restricciones de ningún tipo a la hora de realizarlo.

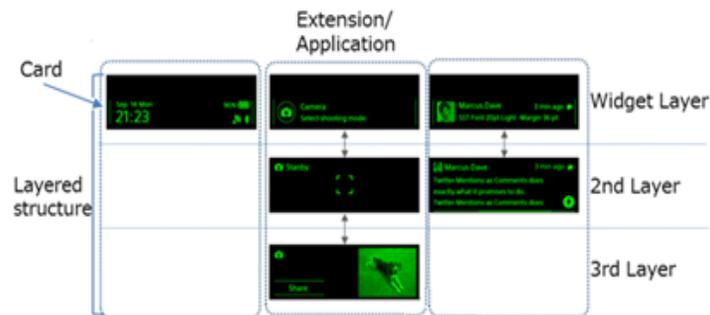


Figure 2: Interfaz Sony SmartEyeglass

– **Tarea 1.2: Implementación funcionalidad modo datos bajo demanda**

Los datos bajo demanda pueden ser consultados por el operario cuando lo desee por tanto estos datos deben estar constantemente actualizándose. Además un compañero de Innovae, que se encargó del lado backend, creó un Servicio Web de tipo REST con una serie de funciones de consulta, modificación y actualización de los datos. Para llevar a cabo la actualización constante de los datos fue necesaria la creación de un Servicio y el uso de un Gestor de Alarmas (Android AlarmManager), de manera que se tenía al servicio ejecutándose en segundo plano para obtener los datos bajo demanda cada 10 segundos. A continuación la respuesta de dicha consulta era un JSON al que se le aplicaba un parser que lo convierta en objetos de tipo Datos Bajo Demanda, que más adelante serían mostrados al usuario.

– **Tarea 1.3: Implementación funcionalidad modo cámara**

Debido a que las gafas de realidad aumentada poseían una cámara incorporada, pensé que podría ser interesante implementar un caso de uso en el que el operario tome una serie de fotos de lo que está viendo y se las transmita a un ordenador, para que sean revisadas al momento o posteriormente por otro usuario. Dicha transmisión se llevará a cabo mediante sockets, para ello nos basaremos en la arquitectura cliente-servidor, donde la aplicación de las gafas será el lado cliente y el ordenador que recibirá las imágenes funcionará como la parte servidor.

– **Tarea 1.4: Pruebas de la aplicación en las gafas**

Se realizarán una serie de pruebas del uso de la aplicación que se expondrán con mayor detalle más adelante.

• **Desarrollo de la aplicación para el reloj inteligente**

– **Tarea 2.1: Diseño de la interfaz en el smartwatch**

El sistema operativo de los relojes inteligentes para los que se ha hecho la aplicación es Android Wear, este sistema operativo también presenta una serie de patrones de diseño a la hora de crear la interfaz de sus aplicaciones dependiendo de la forma del reloj (cuadrado o circular), su resolución y el tamaño de su esfera. Entre los patrones de diseño que hay en Android Wear se decidió utilizar el patrón 2D Picker, ya que veíamos que se ajustaba a la manera en la que queríamos mostrar la información, dicho

patrón permite a los usuarios que naveguen tanto verticalmente como horizontalmente a través de páginas de datos. En lo que respecta al diseño de la interfaz se tuvo que tener en cuenta las preferencias del cliente, en la empresa Innovae se propuso una interfaz al cliente, que tras unos ajustes y correcciones, fue dada por buena. Hay que especificar que la aplicación implementada va dirigida para verse de la manera más óptima en un reloj Huawei Watch.

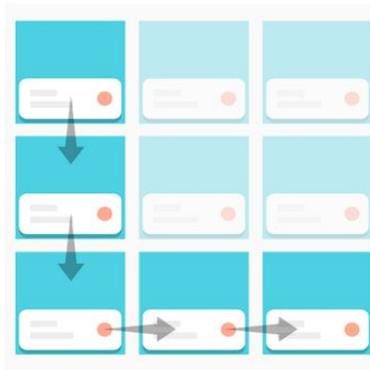


Figure 3: Patrón 2D Picker de Android Wear

– **Tarea 2.2: Implementación funcionalidad modo datos bajo demanda**

Esta funcionalidad es la misma que se explicó previamente para las gafas de realidad aumentada, sólo que en vez de mostrarse la información en las lentes de unas gafas se mostrarán en el reloj del operario.

– **Tarea 2.3: Implementación funcionalidad modo alarmas**

Para la demo de la aplicación se explicó previamente que sólo se podían elegir 2 tipos de usuarios: usuario 1 y usuario 2. Dependiendo del usuario que se elija se recibirá unas alarmas u otras. La funcionalidad de modo alarmas, al contrario que la de datos bajo demanda, muestra la información de las alarmas cuando se activan en el servidor del cliente. Para llevar a cabo esto se ha utilizado Firebase Cloud Messaging (FCM), es una herramienta de Google que permite enviar notificaciones o mensajes push cuando detecta que en el servidor hay datos nuevos disponibles para su sincronización. De esta manera se consigue que las alarmas al activarse en el servidor cliente lleguen al reloj en tiempo real.

– **Tarea 2.4: Pruebas de la aplicación en el smartwatch**

Al igual que con las gafas se realizarán una serie de pruebas del uso de la aplicación del reloj que se expondrán con mayor detalle más adelante.

- **Gestión del proyecto**

- **Planificación**

Elaboración de un documento que muestre la planificación inicial al comienzo del proyecto, las horas estimadas para la elaboración de las tareas y su ubicación en el tiempo. Tiene como finalidad el cumplimiento de todos los objetivos del proyecto definidos al comienzo del mismo.

- **Seguimiento y Control**

Esta fase tiene como objetivo la vigilancia de todas las actividades y tareas durante el desarrollo del proyecto. Es muy importante, ya que un adecuado control hace posible evitar desviaciones en costes y plazos, o al menos detectarlas cuanto antes. Por tanto controlar un proyecto se resume en hacer lo planificado, aplicando las correcciones necesarias cuando nos desviemos.

- **Memoria**

Elaboración de un documento donde se redacta todo el contenido que ha englobado el desarrollo del Trabajo Fin de Grado.

### 3.3.4 Dedicaciones estimadas

<b>FASES</b>	<b>TAREAS</b>	<b>HORAS ESTIMADAS</b>
<b>GESTIÓN DEL PROYECTO</b>	Planificación	10 horas
	Seguimiento y Control	10 horas
	Memoria	70 horas
<b>APLICACIÓN</b>	Estudio (Formación)	40 horas
	Diseño	30 horas
	Implementación	120 horas
	Pruebas	20 horas
<b>TOTAL</b>		<b>300 horas</b>

Figure 4: Tabla con las horas estimadas para el proyecto

### 3.3.5 Diagrama de Gantt

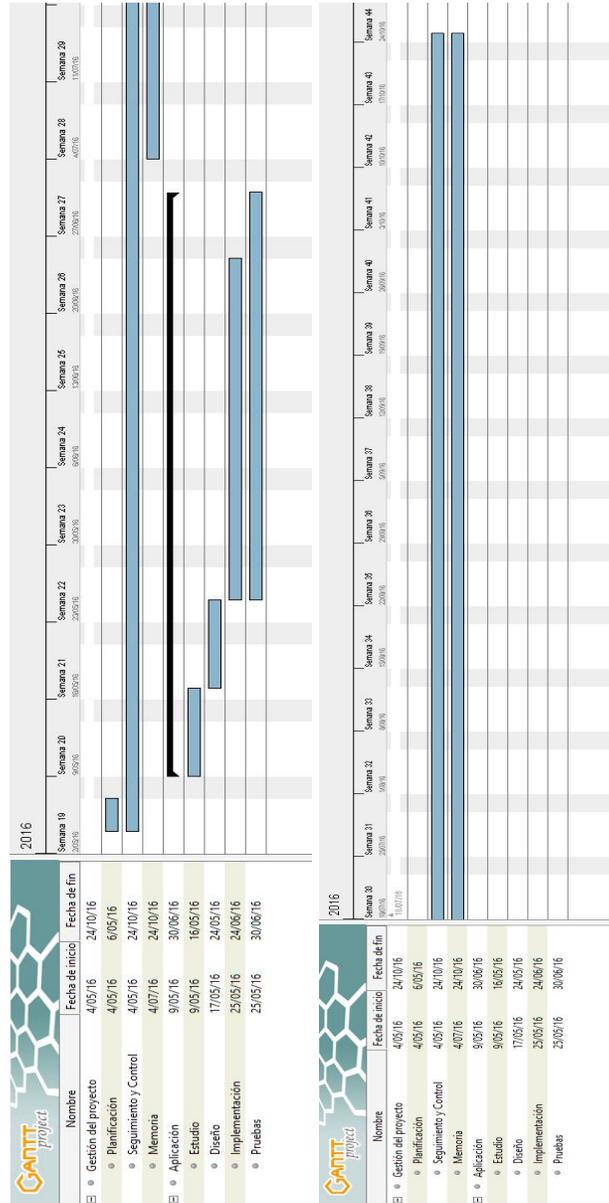


Figure 5: Diagrama de Gantt del Trabajo Fin de Grado

### 3.4 Gestión de Riesgos

- **Identificación de Riesgos**

- El desconocimiento de la programación para dispositivos *wearables*, en nuestro caso para el sistema operativo Android Wear, puede causar retrasos en las entregas y que no se cumplan los tiempos estimados.
- Disponibilidad de los datos por parte de la empresa cliente debido a retrasos o sobrecarga de trabajo en la misma empresa.
- Pérdida de los datos del proyecto por parte del autor del mismo, tanto en lo que se refiere a la aplicación como a la documentación del trabajo.
- Estimación de la dedicación muy por debajo de las horas realmente invertidas.
- Disponibilidad horaria por parte del autor del Trabajo Fin de Grado, ya que además de encontrarse realizando el proyecto, tiene un contrato laboral y se encuentra trabajando durante el desarrollo del proyecto.

- **Mitigación de riesgos**

- Durante mi estancia de prácticas en la empresa se desarrolló una aplicación para las gafas de realidad aumentada, por lo que se tenía un poco de experiencia en el desarrollo de aplicaciones para las gafas. Por otro lado, al final de las mismas prácticas se comenzó a estudiar y a leer acerca de la programación en Android Wear, por lo que se dispone de cierto conocimiento para evitar el riesgo que conlleva.
- Se han ido realizando copias de seguridad, tanto en servicios en la nube como en local, del proyecto cada semana. De esta manera evitamos la pérdida de información (aplicación y documentación) del Trabajo Fin de Grado.
- Se realizó una estimación previa de la dedicación de horas para las tareas que engloban el desarrollo del proyecto, pero en caso de que esta estimación falle se dispone de tiempo para re-ajustar las horas de dedicación a las tareas.
- Debido a que me encuentro trabajando durante el desarrollo del Trabajo Fin de Grado, en el sector de la Hostelería, mis horarios son muy flexibles, de manera que si necesito tiempo para invertir al proyecto puedo hablarlo con mi encargado y re-organizar mis horas de trabajo.

---

## 4 Tecnologías

---

En este apartado se describe el estado del arte que rodea a las tecnologías que han envuelto el desarrollo del proyecto, dichas tecnologías son: las gafas de realidad aumentada, los relojes inteligentes y, aunque no es una tecnología como tal, también la Industria 4.0.

### 4.1 Gafas de Realidad Aumentada

Las gafas de realidad aumentada y virtual han ido ganando importancia a lo largo de los años, han ido mejorando en diferentes aspectos: peso, durabilidad, autonomía, ergonomía, etc. Esto a derivado a que este tipo de dispositivos se pongan en el punto de mira de empresas para darles un uso real en el ámbito laboral.

Ivan Sutherland y posteriormente Steve Mann (Figura 6) han sido los pioneros en el estudio y la búsqueda de posibles usos para estos dispositivos, pero más enfocado en su entorno académico, con el proyecto se trata de demostrar que también tienen un gran potencial para ser usados de forma comercial.



Figure 6: Steve Mann: Evolución de los dispositivos de realidad aumentada.

Estos dispositivos han pasado por muchos cambios desde diseños nada ergonómicos y poco utilizables, a dispositivos más ligeros que brindan una experiencia e interacción con el usuario más cómoda y versátil. Existe un mercado creciente en el desarrollo de estos dispositivos para poder desvincularlos de un ordenador y que utilicen procesadores de menor consumo y tamaño y de mayor potencia. Los dispositivos de realidad aumentada más conocidos son las Google Glass y las M100 Smart Glasses de Vuzix, pero también existen otros como: el Moverio BT-300 Smart Glasses de Epson, Meta 2, las Microsoft Hololens y las Sony SmartEyeglasses de Sony (Figura 7), que son las gafas con las que se ha llevado a cabo el proyecto. A futuro se pronostica un mercado en auge, tanto en hardware como en software, en este tipo de tecnologías.



Figure 7: Gafas de Realidad Aumentada. De izquierda a derecha (Google Glass, Microsoft Hololens, Sony SmartEyeglass).

En el momento del desarrollo del proyecto, en la empresa Innovae Augmented Reality disponían de las Sony SmartEyeglass, por ello se decidió utilizar dichas gafas para llevar a cabo este proyecto. A continuación se mostrará una tabla comparativa (Figura 8) con las gafas de realidad aumentada que se encuentran actualmente en el mercado.

Dispositivo	Campo de Visión	Resolución	Conectividad	Peso	Tecnología	Procesador
<b>Vuzix M100</b> 1079,99 €	15°	(400x240)	Micro USB Wi-Fi 802.11b/g/n Bluetooth	300g	GPS Sensor de Proximidad Cámara 5 megapixel Sensor de Luz Ambiente	OMAP4460 A 1.2GHz 1 GB RAM Android ICS 4.04, API 15 4GB flash
<b>Moverio BT-300</b> 800 €	23°	(1280x720) por ojo	Wi-Fi Bluetooth	58g	Escaneo Ambiente Cámara Frontal OLED Display	Intel Atom de 4 núcleos Android 5.1
<b>Google Glass</b> 1000 €	14°	(640x360)	Wi-Fi 802.11b/g/n Bluetooth	50g	Giroscopio de 3 ejes Acelerómetro de 3 ejes Brijuja Sensor de luz ambiente Sensor de proximidad	OMAP 4430 SoC 1.2GHz Dual (ARMV7) Android 4.0.4 682 MB RAM aprox.
<b>Sony SmartEyeglass</b> 849 €	20°	(419x138)	Wi-Fi 802.11b/g Bluetooth 3.0 Micro USB NFC	77g	Pantalla monocromática Cámara 3 megapixel Acelerómetro Giroscopio Sensor Brillo Micrófono	No son independientes, deben sincronizarse con un móvil con Android 4.4 en adelante.
<b>Microsoft HoloLens</b> 2760 €	120°x120°	(1280x720) por ojo	Wi-Fi 802.11ac Bluetooth 4.1 Micro USB 2.0	579g	4 micrófonos Sensor de luz ambiente Cámara de profundidad Cámara de 2 megapixel y Cámara de vídeo HD	Intel 32 bit 1 GHz, 2 GB RAM * 1GB HPU RAM

Figure 8: Tabla comparativa de gafas de realidad aumentada que se encuentran actualmente en el mercado.

## 4.2 Relojes Inteligentes

Los relojes inteligentes o *smartwatches*, como suelen ser llamados, llevan entre nosotros desde casi 4 décadas. Desde los primeros relojes con calculadora fabricados por Casio o Seiko hasta los que podemos ver hoy en día capaces de recibir llamadas, enviar mensajes, consultar el ritmo cardíaco, etc. La tecnología de estos dispositivos ha ido de menos a más, con la introducción de los ordenadores personales en la década de 1980 muchas compañías comenzaron a desarrollar relojes con capacidad de computación más potentes y con mejores características, pero no ha sido hasta el 2012 en el que realmente nacieron los relojes inteligentes de la mano de Pebble (Figura 9), referente y pionero en el mundo de los *smartwatches*. Tras Pebble más compañías como Samsung, Sony, Huawei y otras se sumaron a la comercialización de estos pequeños dispositivos después del éxito y de la acogida por parte de los usuarios. A día de hoy, todos los grandes fabricantes de electrónica del mundo están implicados con proyectos de estas características. Entre los relojes inteligentes más conocidos de hoy en día estarían el Apple Watch de Apple, el Moto 360 de Motorola, el LG Watch Urbane de LG entre otros. (Figura 10)



Figure 9: Pebble Original, el reloj más exitoso de la historia de Kickstarter.



Figure 10: Relojes Inteligentes que se encuentran actualmente en el mercado.

Debido a mi interés por los *smartwatches* adquirí un Huawei Watch Classic de Huawei, el reloj presenta una buena resolución de pantalla y una duración de la batería más que aceptable por lo que se decidió llevar a cabo el proyecto en este mismo reloj. De todas maneras las pruebas de funcionamiento de la aplicación del reloj se realizaron tanto en dicho reloj como en un LG Watch Urbane propiedad de la empresa Innovae Augmented Reality.

Al igual que se hizo con las gafas de realidad aumentada se mostrará una tabla comparativa (Figura 11) con los relojes inteligentes que hay en la actualidad y sus respectivas especificaciones.

Dispositivo	Procesador	Memoria	Conectividad	Pantalla	Tecnologías	Batería
<b>Pebble Classic</b> 99,95 €	ARM Cortex-M3, up to 80 MHz	128MB memoria interna, 4 MB RAM	Bluetooth 2.1 Bluetooth 4.0 LE	1.26" (144 x168)	Accelerómetro Sensor de Luz Ambiente Resistente al Agua	130 mAh
<b>Motorola Moto 360 2 Gen</b> 289 € - 356 €	Snapdragon 400 (4x 1.2 GHz) // Adreno 305	4GB memoria interna, 512MB RAM	Wi-Fi 802.11b/g/n Bluetooth 4.0 LE	1.37" (360x325) 263ppi // 1.56" (360x330) 233ppi	Sensor Ritmo Cardíaco Accelerómetro Sensor de Luz Ambiente Giroscopio	300 mAh - 400 mAh
<b>Apple Watch</b> 419 € - 15.000 €	Apple S1	8GB memoria interna, 512MB RAM	Wi-Fi 802.11b/g/n Bluetooth 4.0	Retina OLED con Force Touch, (340x272 píxeles) - (390x312 píxeles)	Resistente a salpicaduras Sensor de frecuencia cardíaca Accelerómetro Giroscopio Sensor de luz ambiental	Hasta 18 horas de autonomía
<b>LG Watch Urbane</b> 349 €	Qualcomm Snapdragon 400 a 1.2 GHz	4GB memoria interna, 512MB RAM	Wi-Fi 802.11b/g/n Bluetooth 4.1 LE	1.3" circular P-OLED (320x320) 245ppi	Resistente al agua y polvo Accelerómetro Brújula Pulsómetro Sensor de frecuencia cardíaca	410 mAh
<b>Huawei Watch</b> 349 € - 449 €	Snapdragon 400 y Adreno 305	4GB memoria interna, 512MB RAM	Wi-Fi Bluetooth 4.1 LE	1.4" panel AMOLED (400x400) 286ppi	Accelerómetro Giroscopio Sensor de frecuencia cardíaca	300 mAh

Figure 11: Tabla comparativa de los relojes inteligentes.

### 4.3 Industria 4.0

La Primera Revolución Industrial surgió de la invención de la máquina de vapor y la mecanización del trabajo manual (1784). Más adelante llegó la producción en cadena, impulsada por Henry Ford (1870). La tercera transformación, y más reciente, surgió a raíz de la incorporación de la electrónica para automatizar la producción (1969). Y ahora con la llegada de las nuevas tecnologías es necesario una nueva renovación en el sector industrial a esto es lo que llamamos, Industria 4.0, esta nueva etapa de la transformación industrial que se caracteriza por la interconexión de máquinas y de sistemas en el propio lugar de producción y por un flujo ágil de la información con el exterior (Figura 8).

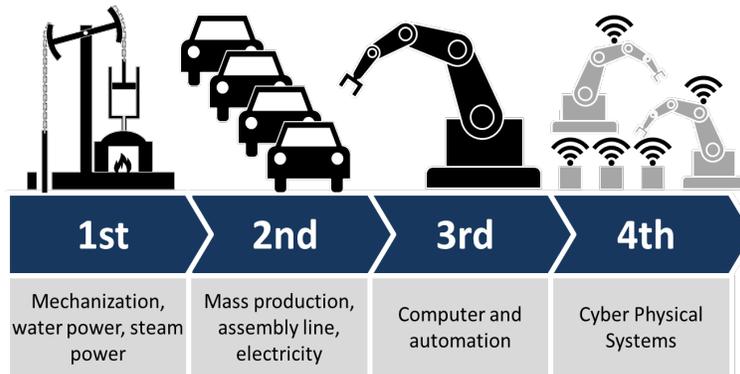


Figure 12: Evolución de la Industria

Las bases tecnológicas en las que se apoya esta Industria Conectada son los siguientes:

1. **Tecnologías de la Información:** conocida como IT (Information Technology) es la aplicación de los ordenadores y equipos de telecomunicación para la transmisión, almacenaje, transmisión, recuperación y manipulación de datos utilizados en el entorno de los negocios y las empresas. Los ERP (Enterprise Resource Planning) son un claro ejemplo de la importancia de los sistemas informáticos en el ámbito empresarial, ya que estos sistemas sirven de ayuda para la integración de la información en las firmas, la asignación de recursos disponibles, la toma de decisiones, los sistemas de control de calidad, etc.
2. **Internet de las Cosas:** conocida como IoT (Internet of Things) se trata de una revolución en las relaciones entre objetos y personas, incluso entre

los objetos directamente, estos se conectarían entre ellos y con el Internet y ofrecerían datos en tiempo real. Comenzando así la digitalización del mundo físico, esos datos generados en tiempo real pueden servir para muchos usos:

- Conocer en tiempo real las necesidades de las plantas de nuestro hogar.
- Pulseras de actividad que nos informen de nuestras constantes vitales.
- Lámparas inteligentes que se enciendan cuando estemos en casa.
- Zapatillas que indiquen la distancia recorrida a la semana.

A diferencia de algunas tecnologías mucho más populares entre las masas, el Internet de las Cosas no ha encontrado su foco de explosión en el mercado del consumo. Quizás la tecnología está aún muy poco desarrollada, o quizás los grandes del sector no han visto la oportunidad correcta para abalanzarse encima.

De todas maneras, es el sector privado donde el Internet de las Cosas se está haciendo cada vez más popular.

- **La industria de producción en masa:** la maquinaria que se encarga de controlar los procesos de fabricación, robots ensambladores, sensores de temperatura, control de producción, cada vez más en las empresas todo se encuentra conectado al Internet lo que permite centralizar el control de la infraestructura.
- **Control de infraestructura urbana:** control de semáforos, puentes, vías de tren, cámaras urbanas. Cada vez más ciudades implementan este tipo de infraestructuras, basadas en el Internet de las Cosas, que permiten monitorizar el correcto funcionamiento de sus estructuras además de adaptar más flexiblemente su funcionamiento ante nuevos eventos.
- **Control ambiental:** una de las áreas en las que está teniendo más éxito el Internet de las Cosas, pues permite acceder desde prácticamente cualquier parte a información de sensores atmosféricos, meteorológicos y sísmicos.
- **Sector de la salud:** cada vez más clínicas y hospitales alrededor del mundo confían en sistemas que les permiten al personal de salud monitorizar activamente a los pacientes de manera menos invasiva.

3. **Sistemas Ciberfísicos:** también conocidos como CPS (Cyber-Physical Systems) son todos aquellos dispositivos que integran capacidades de computación, almacenamiento y comunicación para controlar e interactuar con un proceso físico. Los sistemas ciberfísicos están, normalmente, conectados entre sí y a su vez conectados con el mundo virtual y las redes digitales

globales. El objetivo final es lograr una mejor calidad de vida mediante la monitorización y control, del mundo físico en el que vivimos utilizando las capacidades del mundo cibernético.

Se pueden aplicar principalmente en los sistemas de transporte , automóviles, fábricas, procesos industriales, hospitales, oficinas, hogares, ciudades y dispositivos personales, dando paso a una nueva generación de elementos interconectados.

---

## 5 Herramientas

---

En este apartado se describen las distintas herramientas utilizadas tanto para el desarrollo, gestión y pruebas de la aplicación.

### 5.1 Desarrollo

Las herramientas utilizadas para la creación de las dos aplicaciones, tanto de las gafas de realidad aumentada como la del reloj inteligente, han sido las que se presentan a continuación.

#### 5.1.1 Android



Figure 13: Logo Android

Android es un sistema operativo para dispositivos móviles con pantalla táctil, teléfonos inteligentes, tablets, relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y que terminó comprando en 2005. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

Android tiene una gran comunidad de desarrolladores creando aplicaciones para extender la funcionalidad de los dispositivos. Estos mismos desarrolladores pueden poner a disposición de los usuarios sus propias aplicaciones gracias a la

tienda en línea desarrollada y operada por la misma Google, la Google Play Store (anteriormente conocida como Android Market), dicha tienda ha registrado en los tres primeros meses de 2016 más de 11.000 millones de descargas de aplicaciones y posee alrededor de 2.200.000 aplicaciones en su market. Esto la convierte en la tienda de aplicaciones móviles con mayor número de aplicaciones del mercado.

Android ha visto numerosas actualizaciones desde su liberación inicial. Dichas actualizaciones al sistema operativo base arreglan bugs y agregan nuevas funciones. Generalmente cada actualización del sistema es desarrollada bajo un nombre en código de un elemento relacionado con postres.

La distribución actual de versiones de Android a día de hoy es la que se puede observar en la siguiente figura:

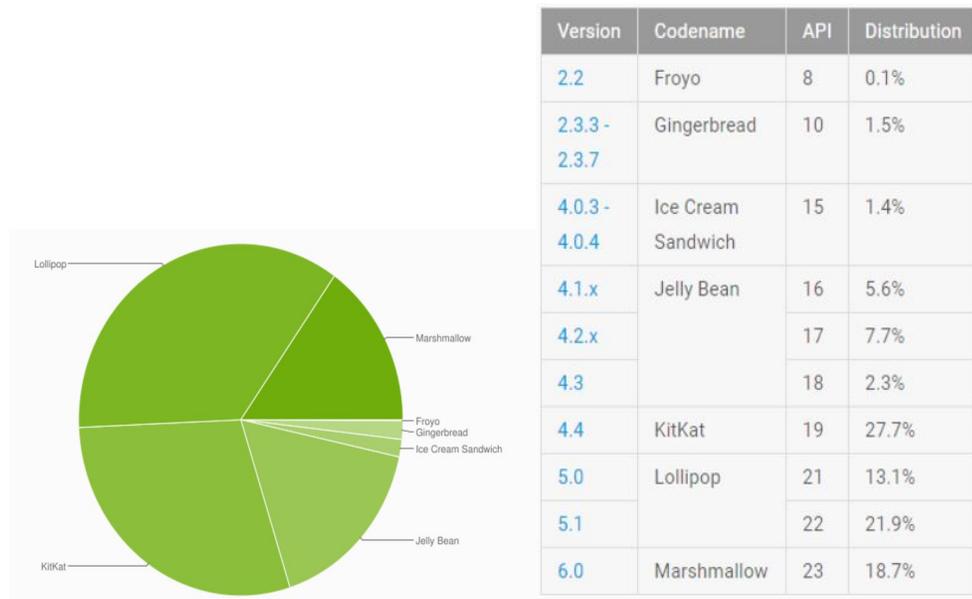


Figure 14: Distribución de versiones Android en el mes de Septiembre del 2016.

Cuanto menor sea la versión de Android para la que se crea la aplicación, esta abarcará a un mayor número de dispositivos y usuarios, por lo que se busca que la versión de Android sea la menor posible, sin embargo para el correcto funcionamiento de nuestra aplicación se presentan ciertas limitaciones en la elección de la versión de Android. Por ejemplo, para que el reloj inteligente

funcione correctamente al conectarlo con el teléfono móvil se requiere un teléfono con Android 4.3 o versiones superiores. Por tanto se ha establecido como versión mínima de Android para el uso de ambas aplicaciones, tanto la de las gafas como la del reloj, la versión Jelly Bean 4.3 (API 18). De esta manera según la imagen anterior, se abarcaría al 83,7% de los dispositivos móviles actuales que posean el sistema operativo Android.

### 5.1.2 Android Studio



Figure 15: Logo Android Studio

Android Studio es el IDE para la plataforma Android. Fue anunciado el 16 de mayo de 2013, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android.

En un principio no se tenía experiencia alguna con el IDE antes mencionado. Durante la carrera se vió muy brevemente un poco de Android pero haciendo uso del IDE Eclipse (mediante la instalación del plugin Eclipse ADT), pero a principios del 2016 se anunció que dicho plugin dejaría de tener soporte y que los proyectos en Android debían ser migrados a Android Studio. Por esa razón nos terminamos decantando por hacer uso de Android Studio para el desarrollo de las dos aplicaciones a implementar.

Algunas características de Android Studio son:

- Renderización en tiempo real.
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en Gradle.

- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.

### 5.1.3 Firebase Cloud Messaging



Figure 16: Logo Firebase

Firebase Cloud Messaging (FCM) es una solución multiplataforma que te permite enviar, de forma gratuita y segura, mensajes y notificaciones.

Al usar FCM, puedes notificarle a una aplicación cliente que un correo electrónico nuevo u otros datos están disponibles para la sincronización. Puedes enviar notificaciones para aumentar la captación y retención de usuarios. Para casos de uso como mensajería instantánea, un mensaje puede transferir una carga de hasta 4 KB a una app cliente.

Una implementación de Firebase Cloud Messaging se compone de un servidor de aplicación que interactúa con FCM a través del protocolo HTTP o XMPP y una aplicación cliente. Además, FCM incluye la consola de Notificaciones, que puedes usar para enviar notificaciones a la app cliente.

Algunas de las funciones de FCM son:

- Distribuye mensajes a tu aplicación cliente en cualquiera de las siguientes formas: a dispositivos individuales, a grupos de dispositivos o a dispositivos suscritos a temas.
- Entrega notificaciones de hasta 2 KB, cargas de datos de hasta 4 KB y envía mensajes con carga de datos y notificaciones.
- Envía reconocimientos, chats y otros mensajes desde los dispositivos a tu servidor mediante el canal de conexión de FCM confiable y que consume poca batería.

## 5.2 Gestión

Como herramientas para la organización y gestión del proyecto se hacen uso de las siguientes:

### 5.2.1 Trello



Figure 17: Logo Trello

Trello es una herramienta de gestión de proyectos con interfaz web, cliente para IOS y Android. Su uso no solo se limita a la organización de proyectos en el trabajo, sino que también para la organización de tareas del hogar, citas, viajes o cualquier otra cosa.

Antes de comenzar en la empresa Innovae Augmente Reality no conocía la herramienta de Trello, pero al comenzar el proyecto del reloj inteligente mi supervisor en la empresa me habló de la herramienta de Trello y de cómo la usaban para gestionar los distintos proyectos que acordaban, de esta manera tanto él como yo, llevábamos un seguimiento de qué tareas había que realizar y los plazos acordados para dichas tareas.

La herramienta es muy útil y organiza los proyectos en forma de tableros o “boards”. Desde la aplicación se pueden llevar a cabo muchas acciones: tomar notas, añadir listas de control, adjuntar archivos, compartir con personas, añadir etiquetas con colores, buscar en el archivo, asignar tareas a ciertos usuarios y colocar fechas límites a las tareas.

En estos tableros se añaden cards o “tarjetas” son como pequeños post-its en un tablero de corcho pero en formato digital, de modo que puede buscarse en ellas, compartirlas o añadir recordatorios.

Lo ideal es usar Trello aplicando el método Kanban para lograr hacer las cosas. Esta técnica consiste en dividir un trabajo en columnas y que en cada una de estas haya tareas. En la primera columna se coloca lo que necesita hacerse, la segunda columna contendrá las tareas que se están haciendo en ese momento, y la tercera columna tendrá las tareas que ya se han terminado. Esta es la manera

más fácil de comenzar y organizarse, pero todo esto puede editarse a medida que el proyecto avance para cubrir las necesidades del usuario.

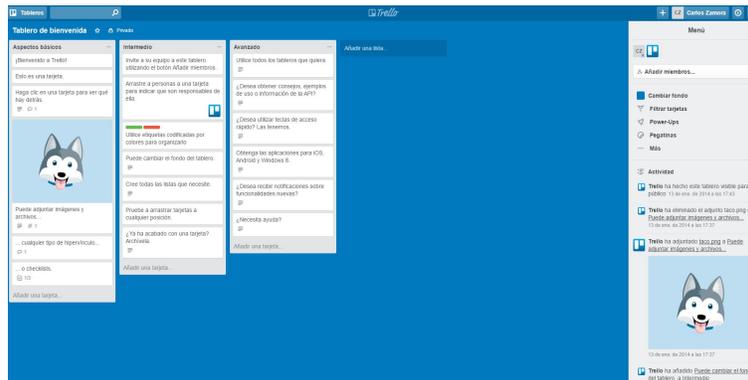


Figure 18: Tableros Trello

### 5.3 Dispositivos de prueba de la aplicación

Los dispositivos utilizados para probar el correcto funcionamiento de la aplicación han sido los siguientes:

En lo que respecta a las gafas de realidad aumentada:

- Sony SmartEyeglass

En lo referente a relojes inteligentes se utilizaron:

- Huawei Watch
- LG Watch Urbane

Y en lo que respecta a teléfonos móviles se hizo uso de:

- BQ Aquaris E4.5
- Sony Xperia Z3 Compact
- Huawei P8
- Samsung Galaxy S6

---

## 6 Análisis de Requisitos

---

En este apartado se explicará el modelo del dominio, los diagramas de casos de uso y el flujo de eventos que rodean al proyecto, así como los requisitos funcionales y no funcionales del mismo.

### 6.1 Datos (Modelo del Dominio)

En la especificación del proyecto se definió la estructura de los datos y cómo se implementarían en la Base de Datos. Dicha descripción es la siguiente:

- **Datos Bajo Demanda:** Estos datos podrán ser consultados directamente por el operario cuando lo desee, los campos que lo compondrán son:
  - Índice: identificador del dato.
  - Descripción: texto descriptivo del valor, será el dato que se muestre en el *smartwatch*.
  - Grupo: número que servirá para agrupar los valores en bloques, esto servirá para la visualización de un número concreto de valores en la pantalla del *smartwatch*.
  - Valor actual: valor de la variable en tiempo real.
  - Valor objetivo: valor objetivo para esa variable.
  - Tipo: número que indicará que tipo de dato bajo demanda es.
  - Unidad: representa la unidad de la variable.
  - Fecha y Hora: fecha y hora de la última actualización del valor actual del dato concreto.
- **Visibilidad de Alarmas:** Esta tabla definirá si un usuario (solo se definirán 2 usuarios para la demo de la aplicación) tiene o no visibilidad sobre las alarmas. Se compondrá de:
  - Índice: id de la alarma.
  - Visible Usuario 1: tendrá el valor 1 si el usuario 1 tiene visibilidad sobre la alarma y 0 sino la tiene.
  - Estado Usuario 1: número que indicará el estado de la alarma para el usuario 1, pudiendo ser: 0 (Alarma desactivada), 1 (Alarma activada pendiente de notificar), 2 (Alarma activada y notificada), 3 (Alarma activada, notificada y leída por el usuario) y 4 (Alarma sin visibilidad).

-Visible Usuario 2: tendrá el valor 1 si el usuario 2 tiene visibilidad sobre la alarma y 0 sino la tiene.

-Estado Usuario 2: número que indicará el estado de la alarma para el usuario 2, pudiendo ser: 0 (Alarma desactivada), 1 (Alarma activada pendiente de notificar), 2 (Alarma activada y notificada), 3 (Alarma activada, notificada y leída por el usuario) y 4 (Alarma sin visibilidad).

- **Alarmas:** Estos datos pertenecen a las alarmas que se mostrarán al usuario en el momento que se activen en el servidor de Tafisa. Los campos que lo componen son:

-Índice: id de la alarma.

-Fecha y Hora: registrarán la fecha y hora a la que se ha activado por última vez esa alarma.

-Descripción: texto descriptivo de la alarma, se ajustará la longitud de los textos en función de lo que ocupe su representación en el reloj.

-Nivel: determina el nivel de criticidad de la alarma, se distinguen 3 niveles: alarma, warning, información.

-Estado: número que indica el estado de la alarma, si se encuentra activada o desactivada.

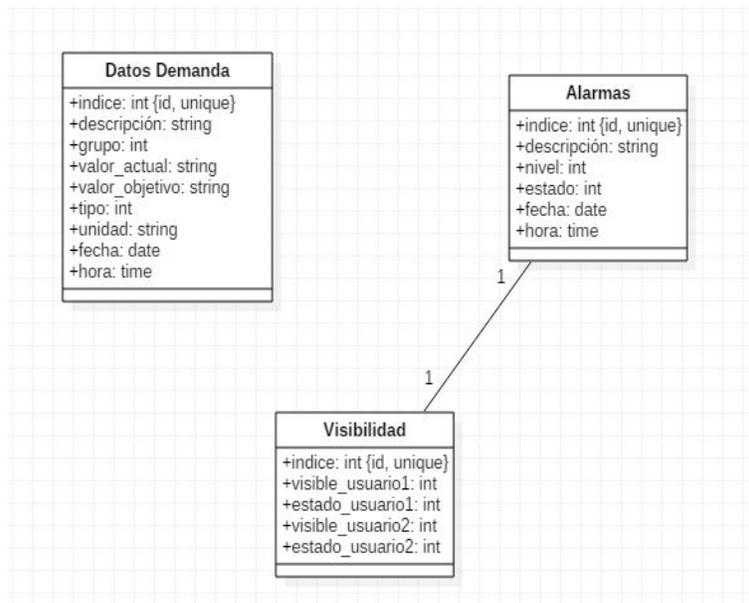


Figure 19: Modelo del Dominio

## 6.2 Diagrama de Casos de Uso

A continuación se mostrará los diagramas de casos de uso correspondientes a las dos aplicaciones, tanto la del *smartwatch* como la de las gafas de realidad aumentada:

### 6.2.1 Diagrama Casos de Uso Smartwatch

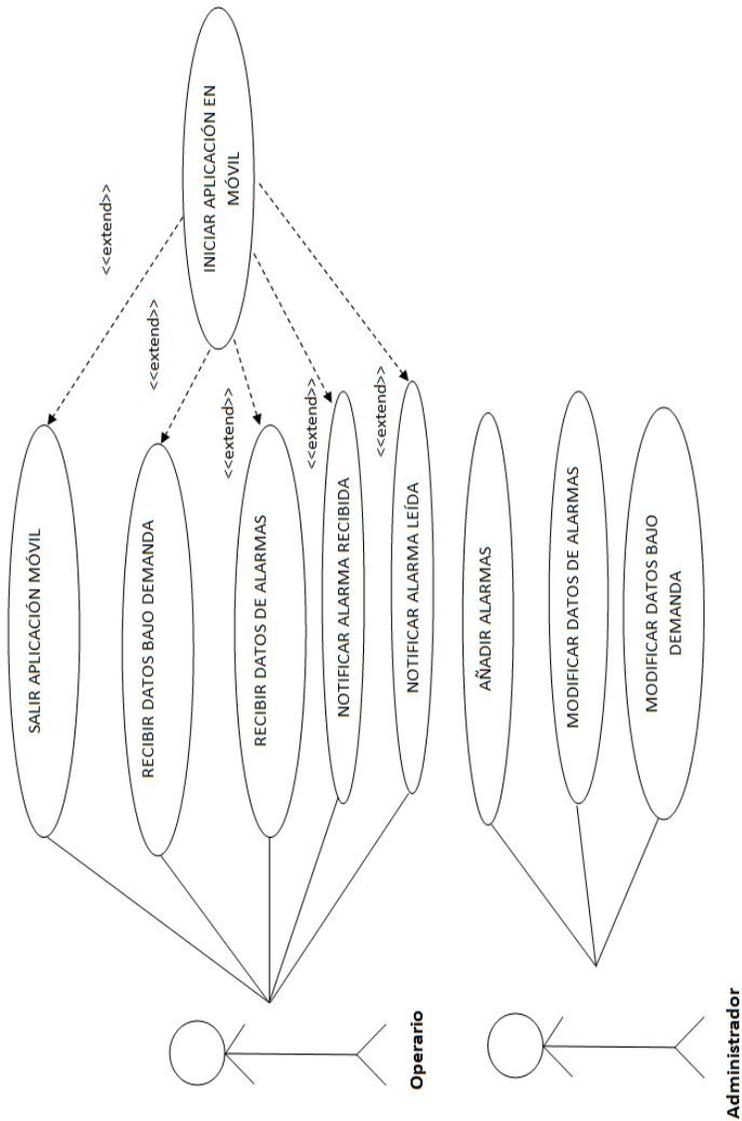


Figure 20: Casos de Uso asociados a la aplicación del reloj

## 6.2.2 Diagrama Casos de Uso Gafas de Realidad Aumentada

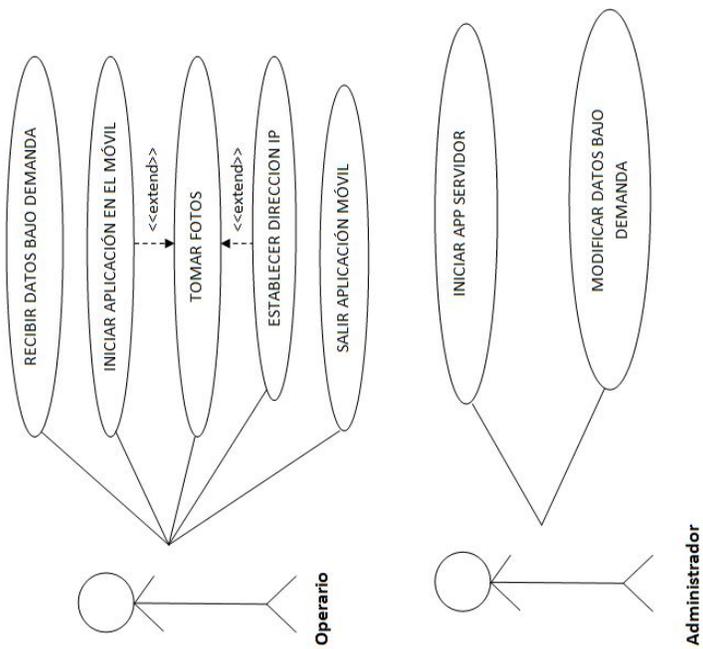


Figure 21: Casos de Uso asociados a la aplicación de las gafas

### 6.3 Flujo de Eventos

A continuación se detallará el flujo de acciones y el comportamiento para los casos de uso expuestos anteriormente.

#### 6.3.1 Flujo de Eventos Smartwatch

<b>Caso de Uso:</b>	Iniciar Aplicación Móvil
<b>Descripción:</b>	Inicia la aplicación instalada en el teléfono móvil que se encuentra sincronizado con el smartwatch utilizado por el operario.
<b>Actores:</b>	Operario
<b>Precondición:</b>	Tener el teléfono móvil con la aplicación instalada y sincronizado con el reloj a usar.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"><li>1. El operario da click en la aplicación en el teléfono móvil para iniciarla.</li><li>2. A continuación al operario le aparecerá un desplegable para poder elegir el usuario con el que desea entrar. Para la demo solo se permiten 2 usuarios. Usuario 1 o usuario 2.</li><li>3. Cuando el operario elija el usuario se habilitará un botón para poder continuar.</li><li>4. Tras presionar dicho botón se iniciará directamente la aplicación en el reloj. A partir de aquí la aplicación del teléfono móvil puede cerrarse y el teléfono puede ser bloqueado. Los servicios de petición de datos al API del servidor quedarán activos y estas peticiones se llevarán a cabo en segundo plano desde el teléfono móvil, para luego ser transmitidas al reloj.</li></ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"><li>1. En caso de no iniciarse la aplicación desde el teléfono móvil no se podrán recibir ni datos bajo demanda ni datos de alarmas. Se podrá acceder a la aplicación del reloj pero esta aparecerá sin datos.</li></ol>
<b>Post Condiciones:</b>	Se ha iniciado la aplicación del teléfono móvil y el operario será capaz de solicitar datos bajo demanda y de recibir datos de alarmas.

Table 1: Flujo de Eventos Caso de uso: Iniciar Aplicación Móvil

<b>Caso de Uso:</b>	Recibir Datos Bajo Demanda
<b>Descripción:</b>	Cuando el operario acceda al reloj a la zona de datos bajo demanda estos se mostrarán para su visualización.
<b>Actores:</b>	Operario
<b>Precondición:</b>	La aplicación tiene que haber sido iniciada desde el teléfono móvil y el usuario ha tenido que ser elegido.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Tras haber iniciado la aplicación desde el teléfono móvil y configurado el usuario con el que se va a trabajar, ya se pueden recibir los datos bajo demanda. El servicio que realiza la petición de datos bajo demanda al servidor se ejecuta cada minuto.</li> <li>2. El operario estando en la pantalla principal del reloj deberá desplazarse a la derecha a la lista de aplicaciones del reloj.</li> <li>3. En la lista de aplicaciones deberá buscar y elegir la aplicación implementada.</li> <li>4. Tras encontrarla podrá darle a la pantalla del reloj sobre la aplicación y acceder a la misma.</li> <li>5. Estando dentro de la aplicación deberá dirigirse a la zona de datos bajo demanda y desde ahí podrá visualizar los datos que desee.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de perderse la conexión entre la fábrica y nuestro servidor se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>2. En caso de perderse la conexión entre nuestro servidor y el teléfono móvil se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>3. En caso de perderse la conexión entre el teléfono móvil y el reloj se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> </ol>
<b>Post Condiciones:</b>	En la zona de datos bajo demanda se encuentran los datos que van a ser visualizados por el operario de la fábrica.

Table 2: Flujo de Eventos Caso de uso: Recibir Datos Bajo Demanda

<b>Caso de Uso:</b>	Recibir Datos de Alarmas
<b>Descripción:</b>	Cuando el operario acceda al reloj a la zona de datos de alarmas, las alarmas activadas se mostrarán. En caso de que el reloj esté bloqueado y se active una alarma este vibrará, la aplicación se iniciará y mostrará dicha alarma.
<b>Actores:</b>	Operario
<b>Precondición:</b>	La aplicación tiene que haber sido iniciada desde el teléfono móvil y el usuario ha tenido que ser elegido.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El operario estando en la pantalla principal del reloj deberá desplazarse a la derecha a la lista de aplicaciones del reloj.</li> <li>2. En la lista de aplicaciones deberá buscar y elegir la aplicación implementada.</li> <li>3. Tras encontrarla podrá darle a la pantalla del reloj sobre la aplicación y acceder a la misma.</li> <li>4. Estando dentro de la aplicación deberá dirigirse a la zona de datos de alarmas y desde ahí podrá visualizar las alarmas que se encuentren activadas.</li> <li>5. La zona de datos de alarmas comprenden la pantalla inicial al entrar a la aplicación y 3 pantallas que se encuentran a su derecha.</li> <li>6. Dependiendo del tipo de alarma que quiera consultar el operario deberá desplazarse por estas pantallas.</li> <li>7. En caso de que el reloj se encuentre bloqueado y se active una o varias alarmas, el reloj se desbloqueará, vibrará y se iniciará la aplicación mostrando dichas alarmas.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de perderse la conexión entre la fábrica y nuestro servidor se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>2. En caso de perderse la conexión entre nuestro servidor y el teléfono móvil se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>3. En caso de perderse la conexión entre el teléfono móvil y el reloj se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> </ol>
<b>Post Condiciones:</b>	En la zona de datos de alarmas se encuentran las alarmas que se encuentran activas en el servidor de la fábrica.

Table 3: Flujo de Eventos Caso de uso: Recibir Datos de Alarmas

<b>Caso de Uso:</b>	Notificar Alarma Recibida
<b>Descripción:</b>	Cuando el operario acceda al reloj a la zona de datos de alarmas y vea la alarma se notificará al servidor que esta alarma ha sido recibida.
<b>Actores:</b>	Operario
<b>Precondición:</b>	La aplicación tiene que haber sido iniciada desde el teléfono móvil y el usuario ha tenido que ser elegido.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El operario estando en la pantalla principal del reloj deberá desplazarse a la derecha a la lista de aplicaciones del reloj.</li> <li>2. En la lista de aplicaciones deberá buscar y elegir la aplicación implementada.</li> <li>3. Tras encontrarla podrá darle a la pantalla del reloj sobre la aplicación y acceder a la misma.</li> <li>4. Estando dentro de la aplicación del reloj en la pantalla inicial en la zona de datos de alarmas, se mostrarán las alarmas recibidas. En el momento que estas alarmas están visibles en la pantalla principal se notifica al servidor que estas alarmas han sido recibidas.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de perderse la conexión entre la fábrica y nuestro servidor se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>2. En caso de perderse la conexión entre nuestro servidor y el teléfono móvil se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>3. En caso de perderse la conexión entre el teléfono móvil y el reloj se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> </ol>
<b>Post Condiciones:</b>	En la zona de datos de alarmas se muestran las alarmas activadas por el servidor de la fábrica y al ser vistas por el operario se notificará que han sido recibidas al mismo servidor.

Table 4: Flujo de Eventos Caso de uso: Notificar Alarma Recibida

<b>Caso de Uso:</b>	Notificar Alarma Leída
<b>Descripción:</b>	Cuando el operario acceda al reloj a la zona de datos de alarmas y vea la alarma se notificará al servidor que esta alarma ha sido recibida, para notificar que la alarma ha sido leída el operario deberá tocar sobre la pantalla donde se muestra la alarma.
<b>Actores:</b>	Operario
<b>Precondición:</b>	La aplicación tiene que haber sido iniciada desde el teléfono móvil y el usuario ha tenido que ser elegido.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El operario estando en la pantalla principal del reloj deberá desplazarse a la derecha a la lista de aplicaciones del reloj.</li> <li>2. En la lista de aplicaciones deberá buscar y elegir la aplicación implementada.</li> <li>3. Tras encontrarla podrá darle a la pantalla del reloj sobre la aplicación y acceder a la misma.</li> <li>4. Estando dentro de la aplicación del reloj en la pantalla inicial en la zona de datos de alarmas, se mostrarán las alarmas recibidas. Para notificar que estas alarmas han sido leídas el operario deberá tocar sobre la pantalla del reloj en la zona donde se muestran las alarmas.</li> <li>5. Al tocar se notificará al servidor de la fábrica que dichas alarmas han sido leídas, desaparecerán de la pantalla inicial de la aplicación y se colocarán en unas listas de alarmas activadas (que se encuentran a la derecha de la pantalla inicial) donde estarán hasta que sean desactivadas por el servidor.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de perderse la conexión entre la fábrica y nuestro servidor se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>2. En caso de perderse la conexión entre nuestro servidor y el teléfono móvil se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> <li>3. En caso de perderse la conexión entre el teléfono móvil y el reloj se mostrará en la página inicial de la aplicación del reloj un mensaje notificando el error. En la zona de datos bajo demanda se seguirán visualizando los últimos datos recibidos antes de que se perdiera la conexión.</li> </ol>
<b>Post Condiciones:</b>	En la zona de datos de alarmas se muestran las alarmas activadas por el servidor de la fábrica y al ser vistas y tocadas por el operario se notificará que dichas alarmas han sido leídas.

Table 5: Flujo de Eventos Caso de uso: Notificar Alarma Leída

<b>Caso de Uso:</b>	Salir Aplicación Móvil
<b>Descripción:</b>	Salir de la aplicación que se encuentra en el teléfono móvil que está sincronizado con el smartwatch.
<b>Actores:</b>	Operario
<b>Precondición:</b>	La aplicación tiene que haber sido iniciada desde el teléfono móvil y el usuario ha tenido que ser elegido.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El operario para cerrar la aplicación deberá hacerlo desde el teléfono móvil.</li> <li>2. Al abrir la aplicación del teléfono móvil deberá presionar sobre el botón "Salir Aplicación".</li> <li>3. Al presionar dicho botón la aplicación del teléfono móvil se cerrará inmediatamente, que la aplicación se cierre implicará que los servicios, que trabajan en segundo plano, de obtención de datos tanto bajo demanda como de alarmas se detengan, impidiendo que la app sature la memoria del teléfono móvil.</li> <li>4. Derivado de lo explicado anteriormente al cerrar la aplicación se dejarán de recibir datos al reloj.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de no salir de la aplicación desde el teléfono móvil los servicios quedarán activos lo que conllevará la saturación de la memoria por parte de la app.</li> </ol>
<b>Post Condiciones:</b>	Se sale de la aplicación desde el teléfono móvil sincronizado al smartwatch y se dejan de recibir datos al mismo.

Table 6: Flujo de Eventos Caso de uso: Salir Aplicación Móvil

<b>Caso de Uso:</b>	Añadir Alarmas
<b>Descripción:</b>	Añade alarmas para que sean visualizadas por el operario.
<b>Actores:</b>	Administrador
<b>Precondición:</b>	No hay.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Esta parte será ejecutada por el administrador de la fábrica para ello un compañero de la empresa elaboró un API con unas funciones para modificar las tablas de la Base de Datos que están en nuestro servidor, entre estas funciones se encuentra la de añadir alarmas.</li> <li>2. Cuando el administrador de la fábrica añada una alarma en su BBDD, se hará una llamada a la función de nuestro API y añadirá la alarma a nuestra tabla de alarmas de nuestra BBDD.</li> <li>3. Al añadir la alarma esta quedará a la espera de ser activada, también por el mismo administrador, para mostrarse en el reloj.</li> </ol>
<b>Flujo Alternativo:</b>	No hay.
<b>Post Condiciones:</b>	Se añade una alarma al sistema que al ser activada será visualizada por el operario desde el reloj.

Table 7: Flujo de Eventos Caso de uso: Añadir Alarmas

<b>Caso de Uso:</b>	Modificar Datos de Alarmas
<b>Descripción:</b>	Modifica los datos de ciertas alarmas que deban ser editadas por el administrador.
<b>Actores:</b>	Administrador
<b>Precondición:</b>	No hay.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Esta parte será ejecutada por el administrador de la fábrica para ello un compañero de la empresa elaboró un API con unas funciones para modificar las tablas de la Base de Datos que están en nuestro servidor, entre estas funciones se encuentra la de modificar los datos de las alarmas.</li> <li>2. Cuando el administrador de la fábrica modifique una alarma en su BBDD, se hará una llamada a la función de nuestro API y modificará dicha alarma en nuestra tabla de alarmas de nuestra BBDD.</li> </ol>
<b>Flujo Alternativo:</b>	No hay.
<b>Post Condiciones:</b>	Se modifican los datos de una alarma en el sistema.

Table 8: Flujo de Eventos Caso de uso: Modificar Datos de Alarmas

<b>Caso de Uso:</b>	Modificar Datos Bajo Demanda
<b>Descripción:</b>	Modifica los datos bajo demanda, ya definidos, que deban ser editados por el administrador.
<b>Actores:</b>	Administrador
<b>Precondición:</b>	No hay.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Esta parte será ejecutada por el administrador de la fábrica para ello un compañero de la empresa elaboró un API con unas funciones para modificar las tablas de la Base de Datos que están en nuestro servidor, entre estas funciones se encuentra la de modificar datos bajo demanda.</li> <li>2. Cuando el administrador de la fábrica modifique los valores de un dato bajo demanda en su BBDD, se hará una llamada a la función de nuestro API y modificará dichos valores en nuestra tabla de datos bajo demanda de nuestra BBDD. Cabe destacar que en la aplicación demo realizada, sólo se permite la modificación de los datos bajo demanda definidos al comienzo del proyecto, pero no agregar nuevos datos.</li> </ol>
<b>Flujo Alternativo:</b>	No hay.
<b>Post Condiciones:</b>	Se modifican los datos bajo demanda en el sistema.

Table 9: Flujo de Eventos Caso de uso: Modificar Datos Bajo Demanda

### 6.3.2 Flujo de Eventos Gafas de Realidad Aumentada

<b>Caso de Uso:</b>	Iniciar Aplicación Móvil
<b>Descripción:</b>	Inicia la aplicación instalada en el teléfono móvil que se encuentra sincronizado con las gafas de realidad aumentada.
<b>Actores:</b>	Operario
<b>Precondición:</b>	Tener el teléfono móvil con la aplicación instalada y sincronizado con las gafas de realidad aumentada a usar.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El operario da click en la aplicación en el teléfono móvil para iniciarla.</li> <li>2. A continuación al operario, en el teléfono móvil, le aparecerá un mensaje y dos botones uno para salir de la aplicación y otro para poder iniciar el modo cámara.</li> <li>3. Al iniciar la aplicación del teléfono móvil automáticamente se iniciará la aplicación de las gafas.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de no iniciarse la aplicación del teléfono móvil no se podrá iniciar el modo cámara desde las gafas. El modo de datos bajo demanda funcionará aunque no se inicie la aplicación del teléfono móvil.</li> </ol>
<b>Post Condiciones:</b>	Se ha iniciado la aplicación del teléfono móvil y el operario será capaz de solicitar datos bajo demanda y de hacer fotos desde el modo cámara.

Table 10: Flujo de Eventos Caso de uso: Iniciar Aplicación Móvil

<b>Caso de Uso:</b>	Establecer dirección IP.
<b>Descripción:</b>	Desde la aplicación del teléfono móvil configurar la dirección IP del servidor donde se mandarán las fotos tomadas desde el modo cámara de las gafas.
<b>Actores:</b>	Operario
<b>Precondición:</b>	Tener el teléfono móvil con la aplicación instalada y sincronizado con las gafas de realidad aumentada a usar.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Dentro de la aplicación del teléfono móvil pulsar el botón "Modo Cámara".</li> <li>2. Desde aquí se introducirá la dirección IP del servidor a donde se enviarán las fotos tomadas desde las gafas.</li> <li>3. Finalmente tras introducir la IP pulsar el botón "Establecer Dirección IP".</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de no introducirse la IP desde el teléfono móvil no se podrá iniciar el modo cámara desde las gafas. Es necesario configurar la IP desde el móvil para iniciar el Modo Cámara desde las gafas.</li> </ol>
<b>Post Condiciones:</b>	Se ha establecido la dirección IP del servidor y se puede iniciar el Modo Cámara desde las gafas de realidad aumentada.

Table 11: Flujo de Eventos Caso de uso: Establecer Dirección IP

<b>Caso de Uso:</b>	Recibir Datos Bajo Demanda
<b>Descripción:</b>	Cuando el operario acceda en las gafas al modo de datos demanda estos se mostrarán para su visualización.
<b>Actores:</b>	Operario
<b>Precondición:</b>	Tener el teléfono móvil con la aplicación instalada y sincronizado con las gafas de realidad aumentada a usar.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Para que el operario pueda recibir los datos baja demanda no es necesario haber iniciado previamente la aplicación del teléfono móvil. El servicio que realiza la petición de datos bajo demanda al servidor se ejecuta cada minuto, es el mismo servicio utilizado para el reloj.</li> <li>2. El operario estando en la pantalla principal de las gafas deberá desplazarse hasta encontrar la aplicación.</li> <li>3. Al presionar en el control de las gafas el botón de selección e iniciar la aplicación se mostrarán las funcionalidades a las que puede acceder, en este caso: Modo Datos Demanda y Modo Cámara.</li> <li>4. Luego deberá acceder al Modo Datos Demanda y desde ahí podrá visualizar los datos que desee.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de perderse la conexión entre nuestro servidor y el teléfono móvil se mostrará una ventana de dialogo con un mensaje notificando el error.</li> <li>2. En caso de perderse la conexión entre el teléfono móvil y las gafas de realidad aumentada por defecto el sistema notifica la desconexión por bluetooth.</li> </ol>
<b>Post Condiciones:</b>	En el Modo Datos Demanda se encuentran los datos que van a ser visualizados por el operario de la fábrica.

Table 12: Flujo de Eventos Caso de uso: Recibir Datos Bajo Demanda

<b>Caso de Uso:</b>	Tomar Fotos.
<b>Descripción:</b>	Desde la aplicación de las gafas se podrán tomar fotos que serán enviadas a un ordenador servidor para poder ser revisadas y visualizadas.
<b>Actores:</b>	Operario
<b>Precondición:</b>	<ul style="list-style-type: none"> <li>- Tener el teléfono móvil con la aplicación instalada y sincronizado con las gafas de realidad aumentada a usar.</li> <li>- Además debería haberse configurado la dirección IP del servidor previamente desde la aplicación del teléfono móvil.</li> <li>- También es necesario iniciar el servidor para que esté a la "escucha" de nuevas fotos.</li> </ul>
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Dentro de la aplicación de las gafas buscar y seleccionar el Modo Cámara.</li> <li>2. Dentro de este modo cada vez que se pulse el touchpad del mando de control de las gafas se llevará a cabo una foto y se enviará al servidor.</li> <li>3. En caso de querer salir del Modo Cámara se deberá mantener pulsado el touchpad para salir al menú de la aplicación.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de no introducirse la IP desde el teléfono móvil no se podrá iniciar el modo cámara desde las gafas. Es necesario configurar la IP desde el móvil para iniciar el Modo Cámara desde las gafas.</li> <li>2. En caso de no iniciar el servidor estando en el Modo Cámara se podrán llevar a cabo fotos pero estas no serán enviadas al servidor.</li> </ol>
<b>Post Condiciones:</b>	Tomar fotos con la cámara de a bordo de las gafas de realidad aumentada y enviarlas a un servidor para su revisión y visualización posterior.

Table 13: Flujo de Eventos Caso de uso: Tomar Fotos

<b>Caso de Uso:</b>	Salir Aplicación Móvil
<b>Descripción:</b>	Salir de la aplicación que se encuentra en el teléfono móvil que está sincronizado con las gafas.
<b>Actores:</b>	Operario
<b>Precondición:</b>	No hay.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. El operario para cerrar la aplicación deberá hacerlo desde el teléfono móvil.</li> <li>2. Al abrir la aplicación del teléfono móvil deberá presionar sobre el botón "Salir Aplicación".</li> <li>3. Al presionar dicho botón la aplicación del teléfono móvil se cerrará inmediatamente, que la aplicación se cierre implicará que el servicio que trabaja en segundo plano, de obtención de datos bajo demanda se detenga, impidiendo que la app sature la memoria del teléfono móvil.</li> <li>4. Derivado de lo explicado anteriormente al cerrar la aplicación se dejarán de recibir datos en las gafas.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de no salir de la aplicación desde el teléfono móvil los servicios quedarán activos lo que conllevará la saturación de la memoria por parte de la app.</li> </ol>
<b>Post Condiciones:</b>	Se sale de la aplicación desde el teléfono móvil sincronizado a las gafas y se dejan de recibir datos al mismo.

Table 14: Flujo de Eventos Caso de uso: Salir Aplicación Móvil

<b>Caso de Uso:</b>	Iniciar App Servidor
<b>Descripción:</b>	Desde el ordenador que hará como servidor, iniciar la aplicación que estará a la " <i>escucha</i> " de las fotos para almacenarlas.
<b>Actores:</b>	Administrador
<b>Precondición:</b>	- Tener en el ordenador que hará como servidor el programa FileServer.exe
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. En el ordenador que servirá como servidor iniciamos el programa FileServer.exe, dicho programa estará a la "<i>escucha</i>" del envío de las fotos tomadas por las gafas.</li> <li>2. Éstas fotos se guardarán en una ruta ya prefijada anteriormente.</li> </ol>
<b>Flujo Alternativo:</b>	<ol style="list-style-type: none"> <li>1. En caso de no iniciar la aplicación del servidor, estando en el Modo Cámara se podrán llevar a cabo fotos, pero estas no serán enviadas al servidor.</li> </ol>
<b>Post Condiciones:</b>	Iniciar la app del servidor para poder recibir las fotos tomadas por las gafas de realidad aumentada.

Table 15: Flujo de Eventos Caso de uso: Iniciar App Servidor

<b>Caso de Uso:</b>	Modificar Datos Bajo Demanda
<b>Descripción:</b>	Modifica los datos bajo demanda, ya definidos, que deban ser editados por el administrador.
<b>Actores:</b>	Administrador
<b>Precondición:</b>	No hay.
<b>Flujo Normal:</b>	<ol style="list-style-type: none"> <li>1. Esta parte será ejecutada por el administrador de la fábrica para ello un compañero de la empresa elaboró un API con unas funciones para modificar las tablas de la Base de Datos que están en nuestro servidor, entre estas funciones se encuentra la de modificar datos bajo demanda.</li> <li>2. Cuando el administrador de la fábrica modifique los valores de un dato bajo demanda en su BBDD, se hará una llamada a la función de nuestro API y modificará dichos valores en nuestra tabla de datos bajo demanda de nuestra BBDD. Cabe destacar que en la aplicación demo realizada, sólo se permite la modificación de los datos bajo demanda definidos al comienzo del proyecto, pero no agregar nuevos datos.</li> </ol>
<b>Flujo Alternativo:</b>	No hay.
<b>Post Condiciones:</b>	Se modifican los datos bajo demanda en el sistema.

Table 16: Flujo de Eventos Caso de uso: Modificar Datos Bajo Demanda

## 6.4 Requisitos Funcionales

Los requisitos funcionales son definiciones de los servicios que el sistema debe proporcionar, describen su funcionamiento, cómo debe reaccionar a una entrada particular y cómo se debe comportar ante situaciones particulares. Los requisitos funcionales asociados a las aplicaciones del reloj y de las gafas son:

### 6.4.1 Requisitos Funcionales Smartwatch

- La aplicación permitirá la selección de sólo 2 usuarios, ésto distinguirá las alarmas que pueden ser visualizadas por cada uno.
- La aplicación del reloj debe mantener actualizados constantemente los datos bajo demanda que se muestran a los operarios.
- La aplicación del reloj debe funcionar aunque la aplicación del teléfono móvil se encuentre en segundo plano o bloqueado.
- La aplicación del reloj debe notificar al operario en tiempo real de la aparición de nuevas alarmas.
- La aplicación del reloj debe alertar al operario si hay una pérdida de conexión entre el teléfono móvil y nuestro servidor.
- La aplicación del reloj debe alertar al operario si hay una pérdida de conexión entre nuestro servidor y el servidor de Tafisa.
- La aplicación del reloj debe alertar al operario si hay una pérdida de conexión entre el teléfono móvil y el reloj inteligente.
- La aplicación del reloj debe permitir al operario poder acceder a la información de la planta tanto desde las propias instalaciones como desde el exterior.
- Si se activan varias alarmas a la vez estas se mostrarán al operario en el reloj por orden de criticidad (alarma, warning, información).
- La aplicación permitirá recibir recordatorios o notificaciones asociadas a la cuenta de Google vinculada al teléfono móvil.

### 6.4.2 Requisitos Funcionales Gafas de Realidad Aumentada

- La aplicación de las gafas debe funcionar aunque la aplicación del teléfono móvil se encuentre en segundo plano o bloqueado.
- La aplicación de las gafas debe mantener actualizados constantemente los datos bajo demanda que se muestran a los operarios.
- La aplicación de las gafas debe alertar al operario si hay una pérdida de conexión entre el teléfono móvil y nuestro servidor.

- La aplicación de las gafas debe alertar al operario si hay una pérdida de conexión entre el teléfono móvil y el reloj inteligente.
- La aplicación de las gafas debe permitir al operario poder acceder a la información de la planta tanto desde las propias instalaciones como desde el exterior.
- Las fotos tomadas desde el modo cámara de las gafas serán guardadas en el servidor para ser visualizadas posteriormente.

## **6.5 Requisitos No Funcionales**

Los requisitos no funcionales o también llamados atributos de calidad, especifican una serie de criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de su funcionamiento. En el caso de nuestras dos aplicaciones estos requisitos serían:

### **6.5.1 Requisitos No Funcionales Smartwatch**

- La aplicación del reloj inteligente debe ser clara y fácil de usar por parte de los operarios de la fábrica.
- Debido a que los operarios están constantemente realizando labores la aplicación del reloj inteligente debe ser lo menos invasiva posible.
- Los textos que se muestran en la aplicación del reloj inteligente, tanto de los bajo demanda como de las alarmas, debido a su reducido tamaño deben ser cortos, claros y entendibles por el operario.
- La aplicación debe estar lista para poder añadir y visualizar más alarmas en función de las necesidades de la fábrica.

### **6.5.2 Requisitos No Funcionales Gafas de Realidad Aumentada**

- La aplicación de las gafas debe ser clara y fácil de usar por parte de los operarios de la fábrica.
- Debido a que los operarios están constantemente realizando labores la aplicación de las gafas debe ser lo menos invasiva posible.
- Los textos que se muestran en las lentes de las gafas de los datos bajo demanda deben ser cortos, claros y entendibles por el operario.
- La toma de fotos en el modo cámara debe ser fácil de usar.

---

## 7 Diseño

---

En este apartado se describirán los diagramas de secuencia de los casos de uso expuestos en el capítulo anterior.

Los diagramas de secuencia modelan la secuencia lógica a través del tiempo de los mensajes entre instancias. Se podría definir como la pila de llamadas resultante de realizar las diferentes operaciones, forman un mapeado de la traza de llamadas que se realizan cuando un participante realiza una acción.

Cabe destacar que los diagramas de secuencia de los casos de uso llevados a cabo por el administrador, no se incluyen en la documentación, esto se debe a que no han sido realizados por mí sino por terceros, y no se sabe exactamente la traza de llamadas que se llevan a cabo en dichos casos.

## 7.1 Diagramas de Secuencia Smartwatch

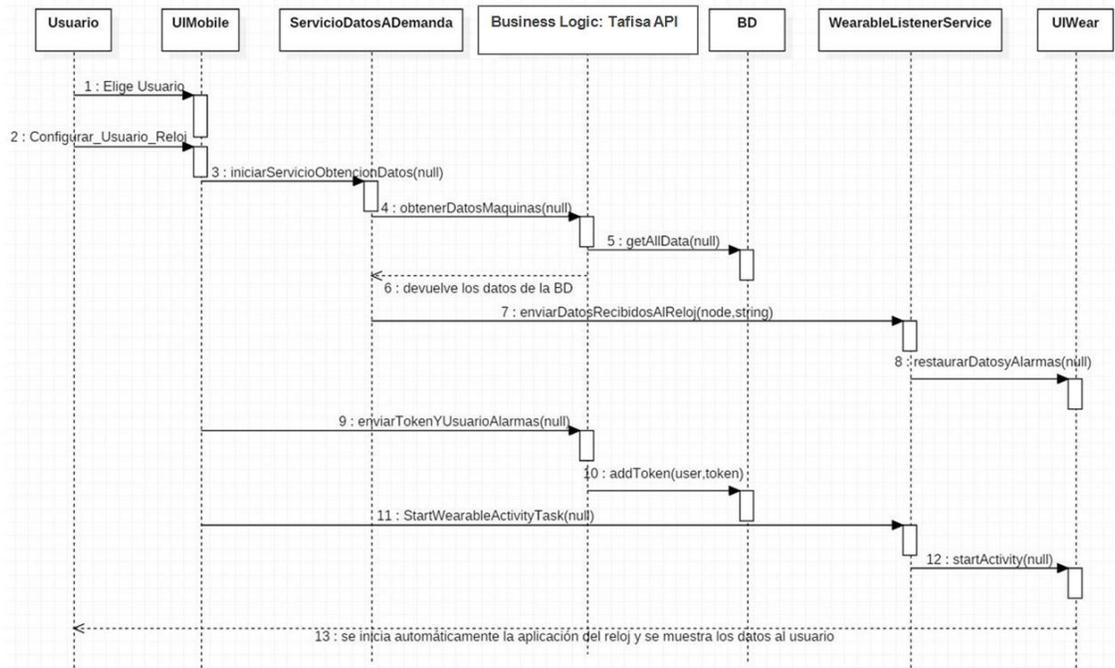


Figure 22: Diagrama de Secuencia: Inicia Aplicación Móvil (Reloj)

En el caso de uso Inicia Aplicación Móvil se lleva a cabo la elección del usuario con el que se desea entrar al sistema, por especificación del proyecto solo hay dos usuarios. Al configurar dicho usuario desde la aplicación del teléfono se inicia el servicio de consulta de datos al servidor externo haciendo uso del REST API, tras esto, los datos recibidos son enviados al reloj. Además, tras elegir el usuario se envía al servidor el usuario elegido y el token de identificación en la plataforma de Firebase (para recibir las notificaciones push). Por último, desde la aplicación móvil se envía un mensaje para que la aplicación del reloj inteligente se inicie automáticamente.

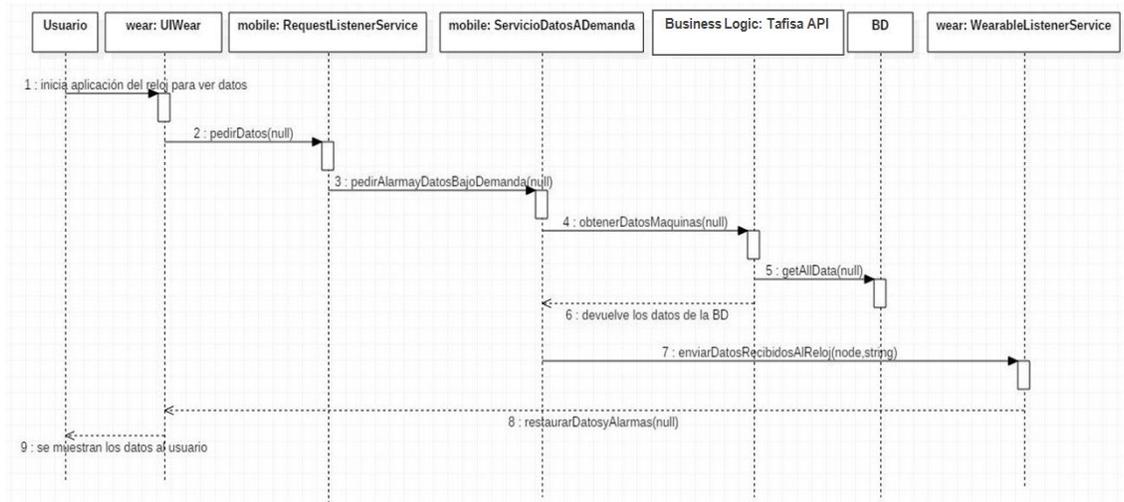


Figure 23: Diagrama de Secuencia: Recibir Datos Bajo Demanda (Reloj)

En el diagrama de secuencias del caso de uso Inicia Aplicación Móvil al configurar el usuario de la aplicación se inicia por primera vez el servicio de consulta de datos, permitiendo que desde ese momento podamos recibir los datos bajo demanda y alarmas desde el reloj. En el diagrama de secuencias del caso de uso Recibir Datos Bajo Demanda se muestra la traza de llamadas que se realizarán cuando el usuario inicia la aplicación del reloj. Tras iniciar la aplicación del reloj, para no tener que esperar a que el sistema haga la siguiente consulta de datos, se pide al módulo *mobile* que envíe los datos bajo demanda más actuales y las alarmas ya activadas; esto hará que se inicie en el momento el servicio de consulta de datos *ServicioDatosADemanda*, si no estaba activado ya previamente, tras eso los datos devueltos se envían al reloj y se muestran al usuario.

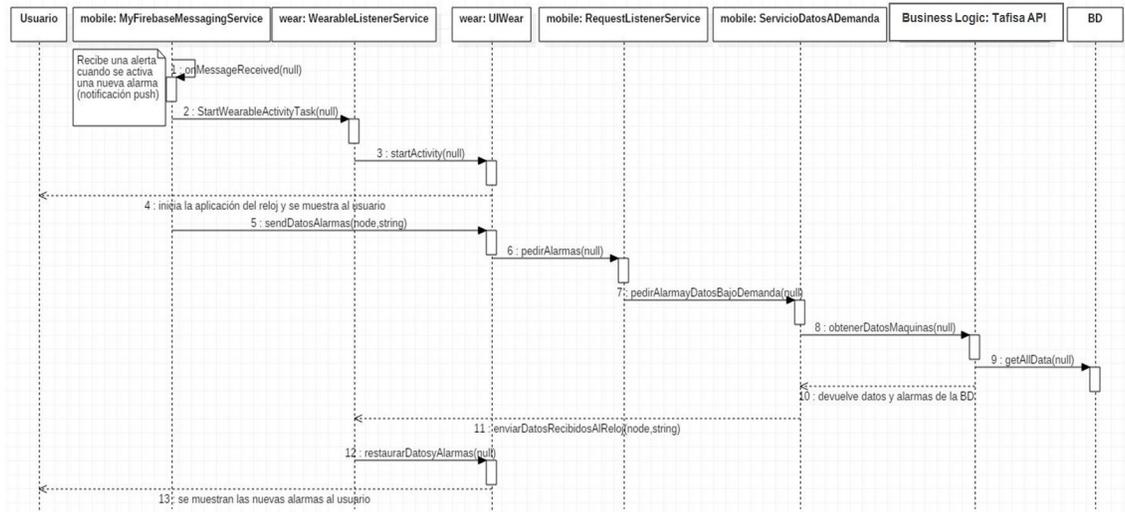


Figure 24: Diagrama de Secuencia: Recibir Datos de Alarmas (Reloj)

La funcionalidad de Recibir Datos de Alarmas se puede llevar a cabo de dos maneras: La primera es que cada vez que el usuario entra a la aplicación se hace la petición de los datos al servicio, este servicio trae tanto los datos bajo demanda como las alarmas activadas, por tanto el diagrama de secuencia sería exactamente el mismo que el de Recibir Datos Bajo Demanda. La otra manera sería que el usuario no entre a la aplicación, y que al activarse la alarma sea el mismo servidor que alerte de la presencia de dicha alarma y la muestre en el reloj, ese diagrama de secuencia es el que se refleja en la parte de arriba.

Se hace uso de una serie de servicios que funcionan en segundo plano, entre estos servicios está *MyFirebaseMessagingService*, cuyo fin es recibir alertas por parte del servidor cuando se activan nuevas alarmas. Tras recibir las alertas la aplicación del reloj se inicia automáticamente, se envía un mensaje al módulo *mobile* para activar el servicio que pide las alarmas y datos bajo demanda (entre los que están las nuevas alarmas activadas), se procede a enviarlos al reloj y finalmente son mostrados al usuario.

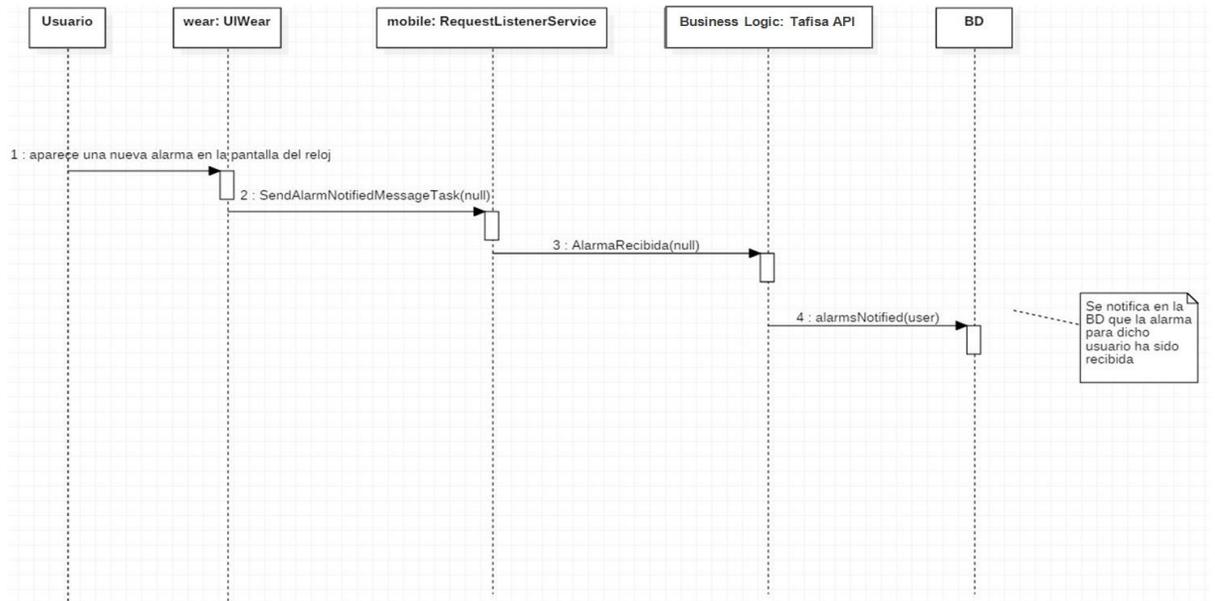


Figure 25: Diagrama de Secuencia: Notificar Alarma Recibida

El caso de uso Notificar Alarma Recibida consiste en notificar al servidor que la alarma fue recibida por el usuario y cambia su estado en la tabla de alarmas de la Base de Datos. Para ello, cuando la alarma es 'escrita en pantalla' se envía un mensaje primero al móvil para avisar que la alarma llegó al reloj, y luego desde el módulo *mobile*, utilizando la función *alarmsNotified()* del REST API, se procede a cambiar su estado en la BD para el usuario con el que se haya configurado inicialmente la aplicación.

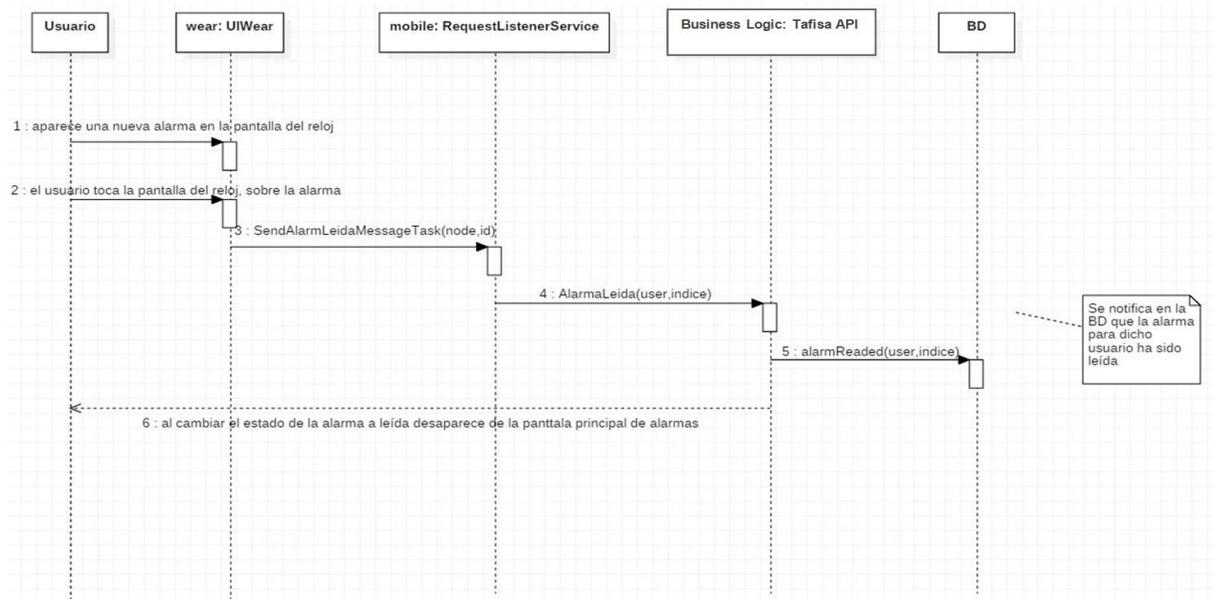


Figure 26: Diagrama de Secuencia: Notificar Alarma Leída

El caso de uso Notificar Alarma Leída consiste en notificar al servidor que la alarma fue leída por el usuario y cambia su estado en la tabla de alarmas de la Base de Datos. Para determinar que la alarma es leída, es necesario que esta sea 'escrita en pantalla' y además que el usuario toque la pantalla sobre dicha alarma. Cuando esas dos acciones ocurran se procede a enviar un mensaje primero al móvil para avisar que la alarma fue leída por el usuario, y luego desde el módulo *mobile*, utilizando la función *alarmReaded()* del REST API, se procede a cambiar su estado en la BD para el usuario con el que se haya configurado inicialmente la aplicación. Al tocar la pantalla del reloj y cambiar el estado de la alarma, esta desaparece de la pantalla principal de alarmas activadas pero se coloca en una lista de alarmas leídas según su tipo (Alarma, Warning, Información). Esta sección de la aplicación puede ser consultada por el usuario cuando lo requiera. Dichas alarmas permanecen en las listas hasta que el administrador las desactive.

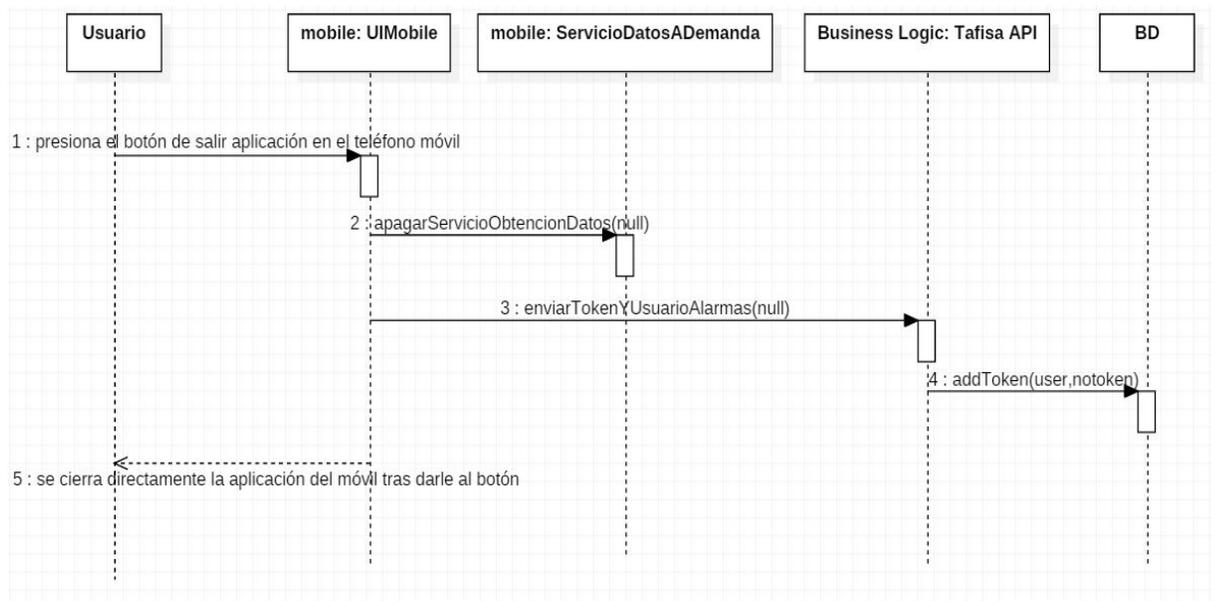


Figure 27: Diagrama de Secuencia: Salir Aplicación Móvil (Reloj)

El caso de uso Salir Aplicación Móvil tiene como fin desactivar la aplicación del reloj, es decir, que no siga llegando al mismo ni datos ni alarmas. Para ello, desde la aplicación del teléfono móvil, tras presionar el botón *Salir Aplicación*, se desactiva el servicio de consulta de datos y se envía mediante el método `enviarTokenYUsuarioAlarmas()`, un token vacío al servidor para el usuario configurado inicialmente, de esta manera al salir de la aplicación el token de dicho usuario quedará libre, permitiendo que otro operario pueda usar dicho usuario más adelante. Tras esto, la aplicación del teléfono se cierra directamente. Es importante que cuando no se desee recibir datos ni alarmas se cierre la aplicación del teléfono móvil, ya que el servicio de consulta de datos queda activo y puede llegar a saturar la memoria del teléfono móvil.

## 7.2 Diagramas de Secuencia Gafas de Realidad Aumentada

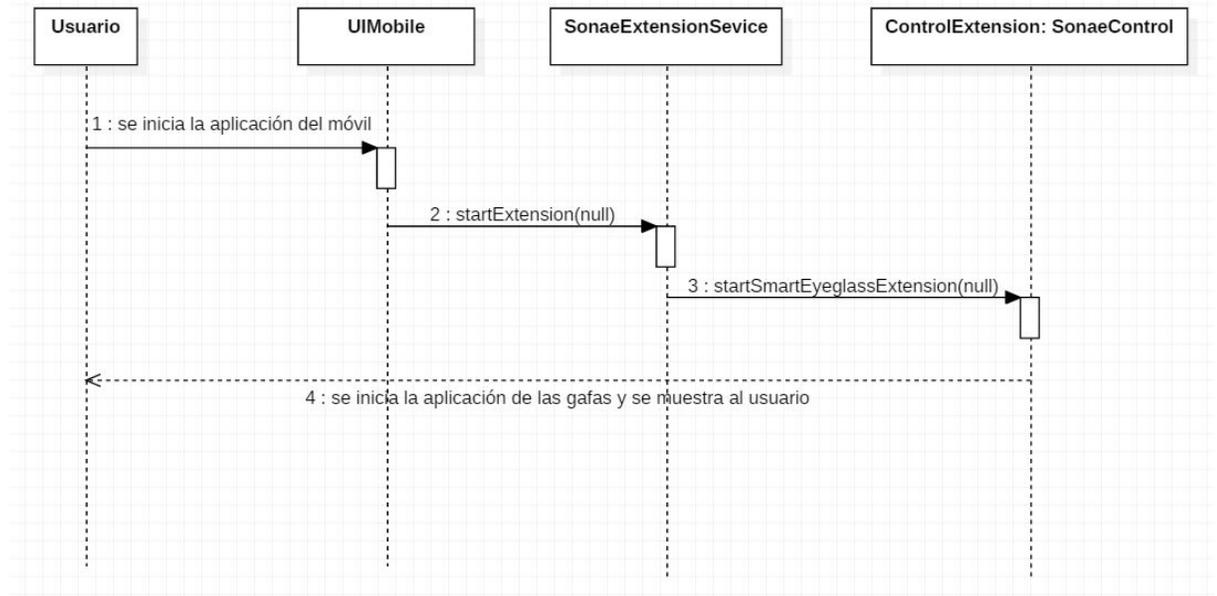


Figure 28: Diagrama de Secuencia: Iniciar Aplicación Móvil (Gafas)

El caso de uso Iniciar Aplicación Móvil de las gafas es distinto al del reloj. No se tiene que configurar ningún usuario y sólo consta de dos botones: uno para configurar la IP del servidor donde se enviarán las fotos de las gafas y el de salir de la aplicación. Además al iniciar la aplicación del teléfono móvil, se inicia la aplicación de las gafas, para ello se envía una petición de extensión al servicio SonaeExtensionService y este iniciará la clase SonaeControl, que se encarga de gestionar la pantalla de las gafas y por tanto de mostrar el menú de nuestra aplicación.

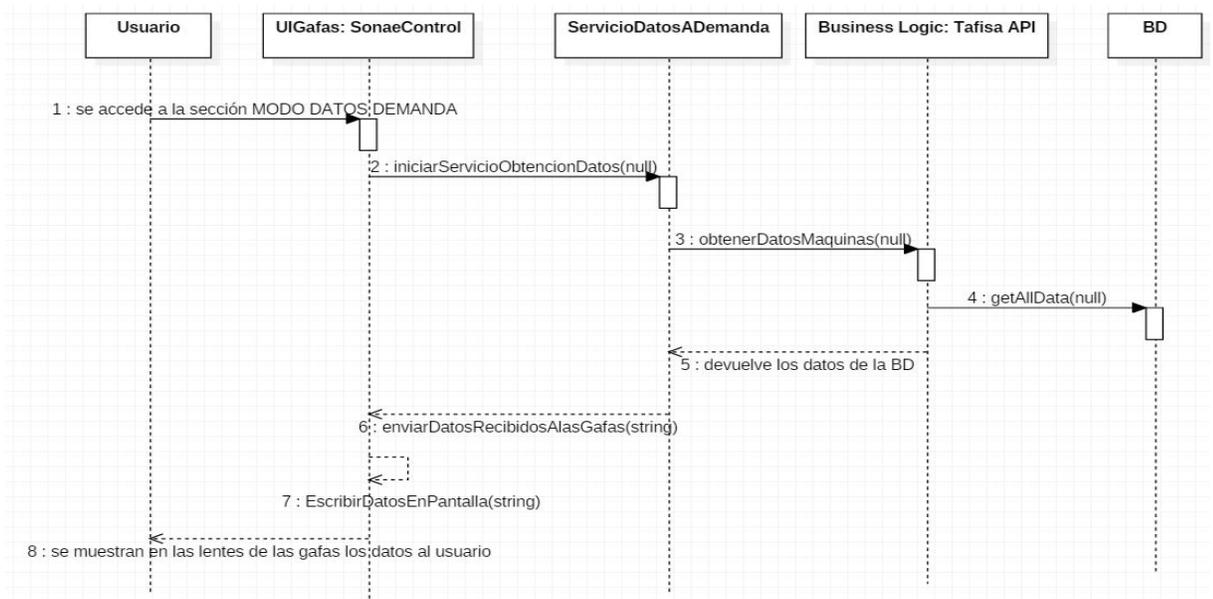


Figure 29: Diagrama de Secuencia: Recibir Datos Bajo Demanda (Gafas)

El caso de uso Recibir Datos Bajo Demanda muestra la traza de llamadas que se realizan cuando el usuario entra a la sección de *Modo Datos Demanda* de las gafas. Tras entrar en dicha sección se inicia el servicio de consulta de datos *ServicioDatosADemanda*, este servicio es el mismo que se utiliza para la aplicación del reloj, tras eso los datos devueltos (como un string) son parseados y convertidos en objetos de tipo *Datos*, y por último son escritos en las lentes de las gafas y mostrados al usuario.

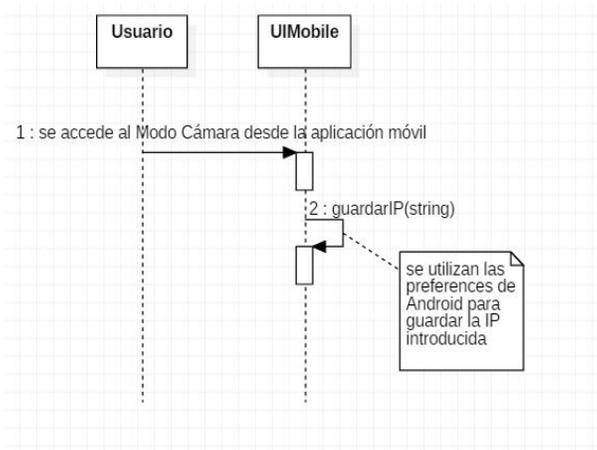


Figure 30: Diagrama de Secuencia: Establecer Dirección IP

El caso de uso Establecer dirección IP es muy sencillo pero necesario para la ejecución de posteriores casos de uso. Simplemente se accede al modo cámara en la aplicación del teléfono móvil, se escribe la dirección IP del servidor donde se desea enviar las fotos tomadas desde las gafas y se configura dicha dirección. Para evitar el uso de bases de datos para guardar dicha dirección, se utilizó las *preferencias compartidas* o *SharedPreferences* de Android, estas preferencias se almacenan en forma de clave-valor y se guardan en ficheros XML.

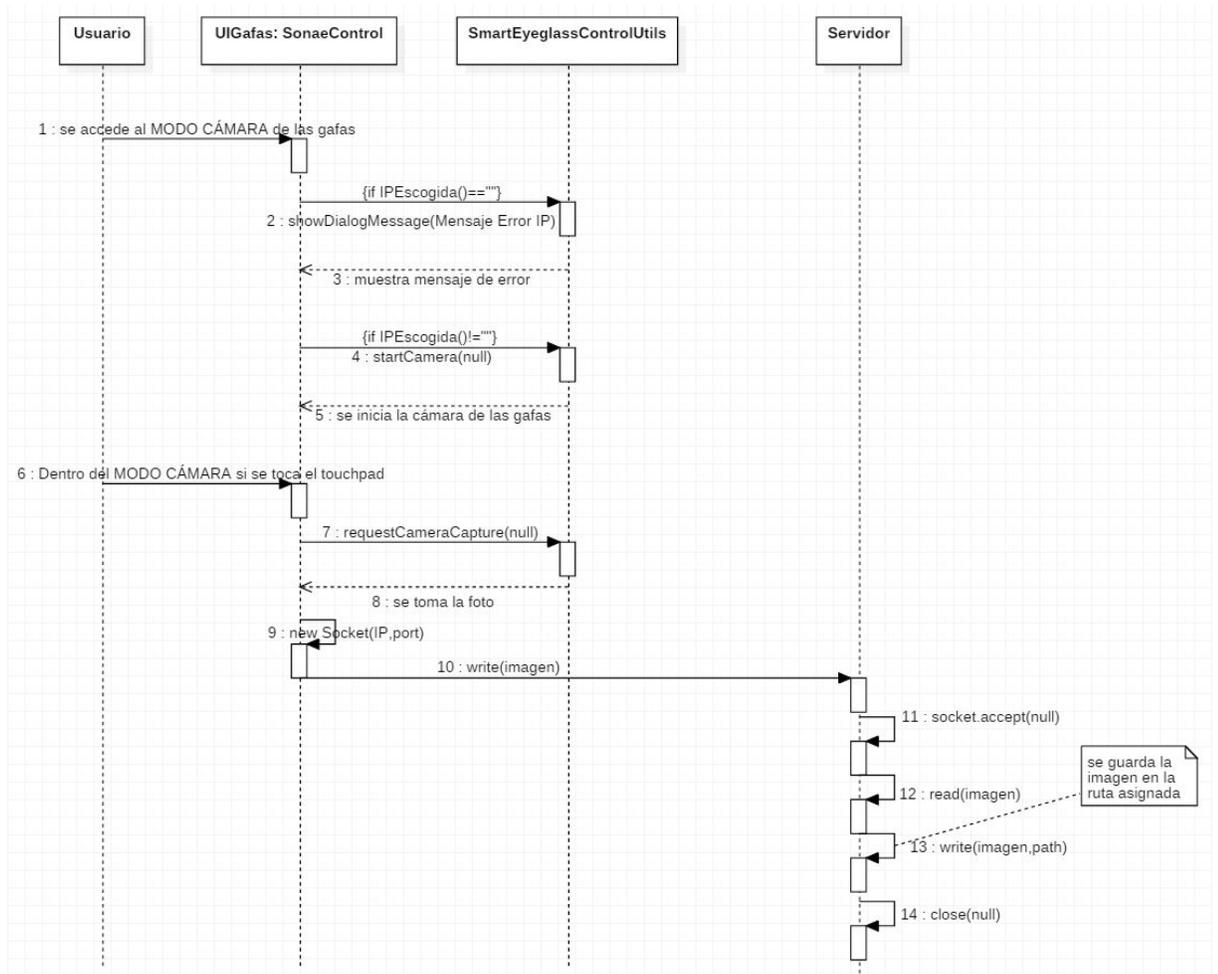


Figure 31: Diagrama de Secuencia: Tomar Fotos

El caso de uso Tomar Fotos consiste en realizar una serie de fotos con la cámara de las gafas de realidad aumentada, y enviarlas a un servidor para que sean visualizadas en el momento o posteriormente. Para ello primero se accede al Modo Cámara desde el menú de las gafas, en caso de que la dirección IP no haya sido configurada desde la aplicación móvil, se muestra un mensaje de error alertando que hay que configurarla y vuelve al menú inicial. Por el contrario, si la dirección IP está configurada accedemos directamente al Modo Cámara. Para llevar a cabo las fotos hay que pulsar el touchpad de las gafas de realidad aumentada, con una pulsación simple se toman las fotos y si se mantiene pulsado

se vuelve al menú inicial. Para iniciar la cámara y realizar las fotos se hace uso de la clase *SmartEyeglassControlUtils*, esta clase provee una serie de métodos y funciones que permite realizar muchas de las funcionalidades específicas de las gafas.

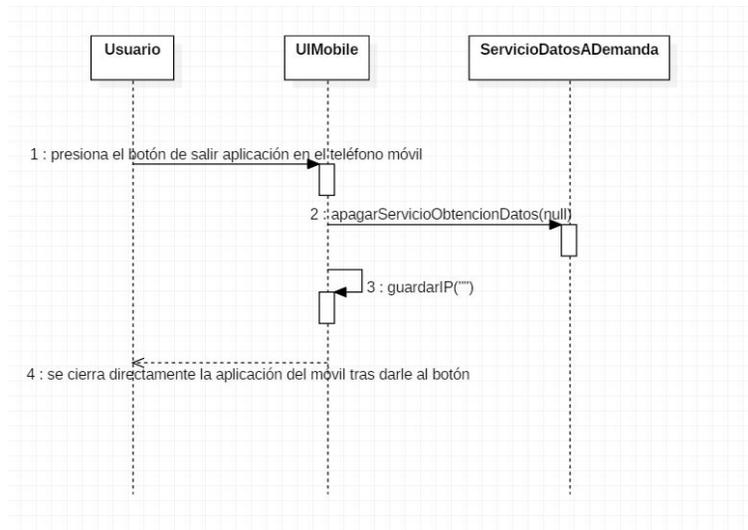


Figure 32: Diagrama de Secuencia: Salir Aplicación Móvil (Gafas)

El caso de uso Salir Aplicación Móvil tiene como fin desactivar la aplicación de las gafas, es decir, que no siga llegando datos al mismo. Para ello, desde la aplicación del teléfono móvil, tras presionar el botón *Salir Aplicación*, se desactiva el servicio de consulta de datos y en las preferencias compartidas de Android se guarda una IP vacía, de esta manera si se quiere volver a tomar fotos se tiene que configurar nuevamente la dirección IP. Tras esto la aplicación del teléfono se cierra directamente.

---

## 8 Implementación

---

En este capítulo se presenta la arquitectura del sistema, una descripción del diseño e interconexiones del mismo y los módulos utilizados para la implementación final del proyecto.

### 8.1 Arquitectura del Sistema

La arquitectura software consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software. También se encarga de definir el diseño y especificación de la estructura global de un sistema. Para el proyecto se eligió adoptar la arquitectura de 3 niveles o capas ya que, además de estar entre las arquitecturas más utilizadas en el desarrollo software, permite la división de la carga de trabajo del sistema en 3 capas claramente diferenciadas. La ventaja principal de esta arquitectura es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de hacer cambios, solo se ataca al nivel requerido sin tener que revisar entre todo el código.

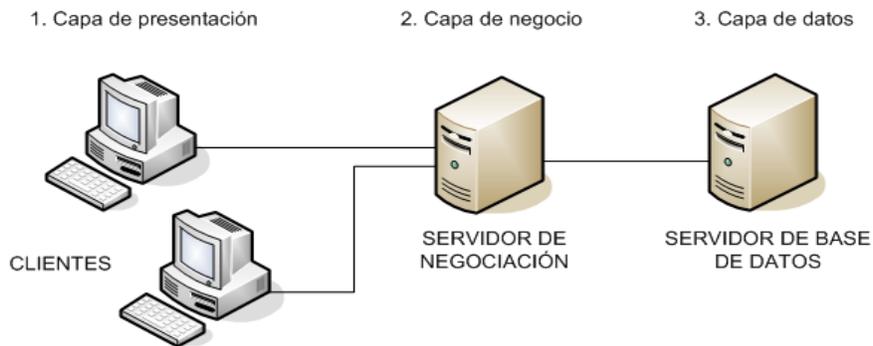


Figure 33: Arquitectura Software de Tres Niveles o Capas

### 8.1.1 Capa de Presentación

La capa de presentación, también conocida como interfaz gráfica, es la capa que ve el usuario, presenta el sistema al usuario y se encarga de comunicar y capturar la información del mismo cuando este interactúa con la interfaz. Al desarrollar aplicaciones se debe tratar que dicha capa sea clara y fácil de usar por el usuario. Esta capa se comunica únicamente con la capa de negocio. En nuestro proyecto hay 4 interfaces: las dos interfaces de las aplicaciones del teléfono móvil para el reloj y para las gafas, y dos más, una para la aplicación del reloj inteligente y una para la aplicación las gafas de realidad aumentada.

Se muestra las interfaces desarrolladas para el proyecto, primero se muestra las interfaces correspondientes a la aplicación de las gafas de realidad aumentada y posteriormente las del *smartwatch*.

- **Capa de Presentación Gafas de Realidad Aumentada**

La interfaz de la aplicación móvil de las gafas de realidad aumentada es muy sencilla, ya que la única funcionalidad que ofrece es la de establecer la dirección IP del servidor donde se enviarán las fotos de las gafas.

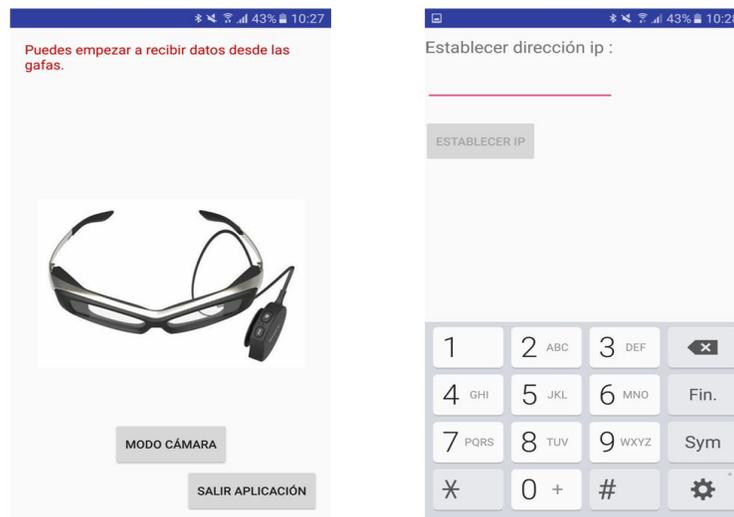


Figure 34: Interfaz aplicación móvil de las gafas de realidad aumentada

Por otro lado la interfaz de la aplicación de las gafas de realidad aumentada suponía un reto, ya que nunca se había desarrollado una interfaz para unas gafas,

pero tras estudiar el API de Sony y revisar las guías de diseño de las aplicaciones de las gafas, se pudo llevar a cabo una interfaz simple pero consistente con la aplicación a desarrollar. Como se explicó previamente, la interfaz de las aplicaciones en el *SmartEyeglass* presentan una estructura de tarjetas que a su vez se estructuran en capas, nos podemos mover de izquierda a derecha entre tarjetas o adentrarnos en las capas. Para moverse por el menú se hace uso del *touchpad* que viene incorporado con las gafas, si se pasa el dedo sobre el *touchpad* de izquierda a derecha nos desplazamos en el menú de izquierda a derecha, y si hacemos clic sobre el *touchpad* nos adentramos en las opciones del menú. Estos son los gestos básicos para interactuar con las gafas.

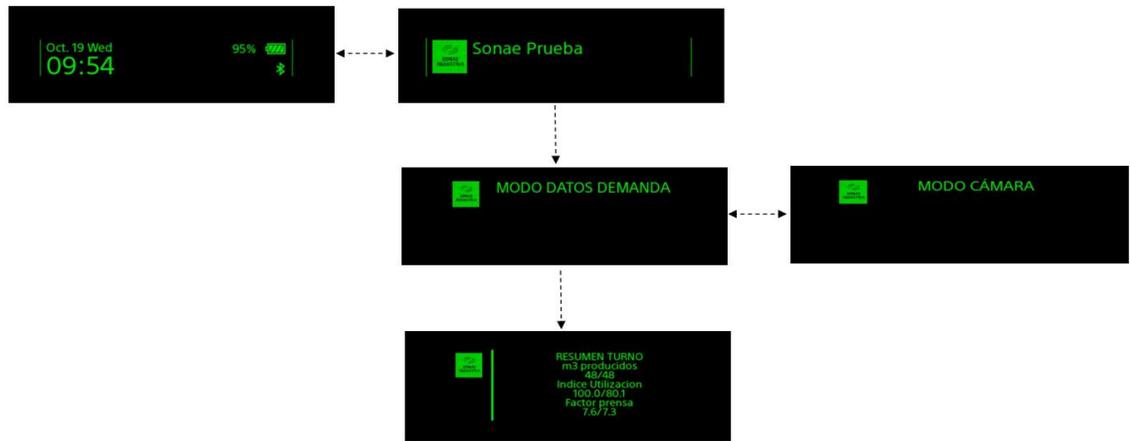


Figure 35: Interfaz aplicación gafas de realidad aumentada

- **Capa de Presentación Reloj Inteligente**

La interfaz de la aplicación móvil del reloj inteligente es la siguiente:

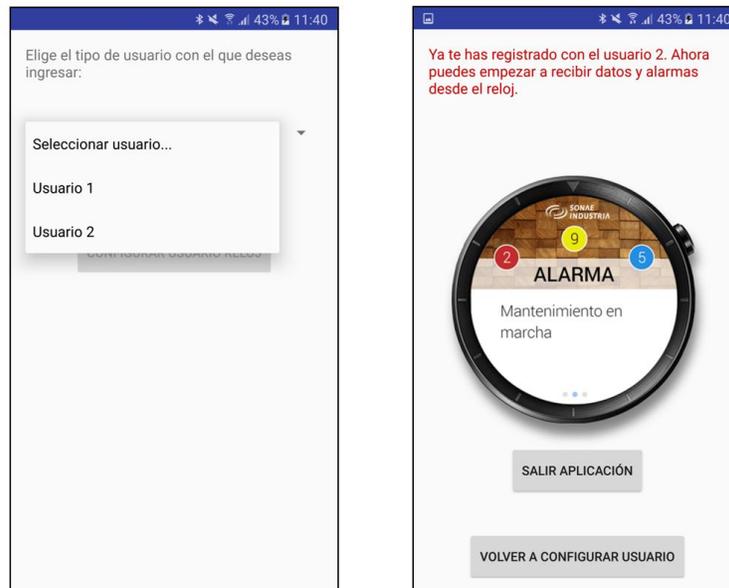


Figure 36: Interfaz aplicación móvil del reloj inteligente

Como se observa en la imagen anterior la interfaz móvil es sencilla, se presenta un desplegable para la selección del usuario con el que se desea entrar en el sistema. Como se explicó en apartados anteriores, solo se definió en la especificación del proyecto dar soporte a dos usuarios. Tras seleccionar el usuario, nos aparecerá la pantalla de la derecha indicándonos qué usuario ha sido elegido y los botones respectivos para “*Volver a Configurar Usuario*”, con el que volvemos a la pantalla inicial (izquierda), y el de “*Salir Aplicación*”, que cierra directamente la aplicación.

En lo referente a la interfaz de la aplicación del reloj inteligente, se desarrolló una interfaz sencilla pero a la vez completa y útil para los operarios que trabajen con ella. No se había trabajado con el sistema operativo Android Wear, pero tras leer la documentación se apreció el parecido que tiene con Android, por lo que no supuso gran trabajo en cuanto a aprender el lenguaje. Por otro lado, Android Wear presenta una serie de patrones de diseño para sus aplicaciones, entre las cuales se decidió utilizar el patrón 2D Picker, ya que como se

explicó anteriormente, se vio que se ajustaba a la manera en la que queríamos mostrar la información, este patrón permite a los usuarios que naveguen tanto verticalmente como horizontalmente a través de páginas de datos. La interfaz de la aplicación del *smartwatch* desarrollada es el siguiente:

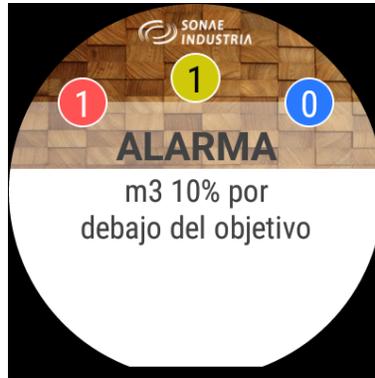


Figure 37: Interfaz aplicación reloj inteligente (pantalla inicial)

En la pantalla inicial de la aplicación se observan las alarmas que están activadas y que aún no han sido notificadas como leídas por el usuario, por ello se muestran en primer plano. Anteriormente se explicó que para notificar una alarma como leída era necesario tocar la pantalla sobre dicha alarma, en cuyo caso desaparecería de la pantalla inicial y quedaría el mensaje “*En espera de alarmas...*”. Por otro lado, se observan tres círculos de colores con números en su interior, los colores de los círculos indican el tipo de alarma (rojo: alarma, amarillo: warning, azul: información) y el número indica cuántas alarmas de dicho tipo están activadas. Por tanto se aprecia que hay: una alarma normal y una alarma de warning activas.



Figure 38: Interfaz aplicación reloj inteligente, lista de alarmas activas por tipo

Desde la pantalla inicial podemos desplazarnos hacia todas las direcciones mediante gestos de arrastre con el dedo. Cabe indicar que los gestos son invertidos a la dirección a la que deseamos ir. Por tanto, si nos queremos desplazar hacia la derecha hay que arrastrar el dedo hacia la izquierda y viceversa, lo mismo se aplica para los gestos de arriba y abajo. Desde la pantalla inicial si nos dirigimos a la derecha de la aplicación, se nos muestran tres pantallas más con las listas de las alarmas activas en el sistema (imagen superior). Otro atajo para llegar a estas pantallas sería tocar sobre los círculos de la pantalla inicial.



Figure 39: Interfaz aplicación reloj inteligente, datos bajo demanda

Por último, desde la pantalla inicial si nos desplazamos hacia abajo se accede a la zona de datos bajo demanda, en esta zona se podrán observar una serie de datos de interes para el operario. En la imagen superior sólo se muestra una de

las pantallas, pero hay más, con la misma interfaz pero con distintos tipos de datos.

### **8.1.2 Capa de Negocio**

La capa de negocio, también conocida como lógica de negocio, es la capa en la que residen los programas que se ejecutan, recibe las peticiones del usuario y envía las respuestas tras procesarlas. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al BBDD el almacenaje o recuperación de datos de él. En el proyecto, para ambas aplicaciones, la lógica de negocio es el API proporcionado por el Servicio Web de tipo REST creado exclusivamente para la comunicación entre el servidor de la fábrica y nuestro servidor externo, y entre dicho servidor externo y el teléfono móvil.

### **8.1.3 Capa de Datos**

La capa de datos, también conocida como capa de acceso a datos, es la capa donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de los datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio. En nuestro proyecto para ambas aplicaciones, debido a que tienen que acceder a datos de la fábrica, esta capa de datos la compondrán: una BBDD MySQL, correspondiente al servidor externo, y otra BBDD SQL Server, correspondiente al servidor de la fábrica. Además para la funcionalidad de tomar fotos de las gafas, el servidor donde se almacenarán dichas fotos será el ordenador donde se ejecute la aplicación servidor.

## 8.2 Interconexiones del Sistema

En la fábrica de Tafisa se recogen una serie de valores relacionados con distintos procesos productivos en una BBDD SQL Server. De todos esos valores, se eligieron los más representativos para ser utilizados y visualizados en la aplicación del *smartwatch* a desarrollar. La comunicación entre la aplicación del *smartwatch* y su BBDD, para la consulta de los datos, no puede ser posible por temas de seguridad. En cambio se decidió instalar y configurar un servidor externo, donde se mantendrá una copia actualizada de los datos a los que se dará visibilidad desde la aplicación del reloj. Al llevar a cabo la actualización de los datos el administrador de la fábrica actualizará también los datos en la BBDD MySQL del servidor externo. Es precisamente desde este servidor desde donde la aplicación del *smartwatch* consultará los datos. De esta manera se puede lograr que un operario pueda acceder a la información de la planta desde las propias instalaciones de la fábrica, como desde el exterior.

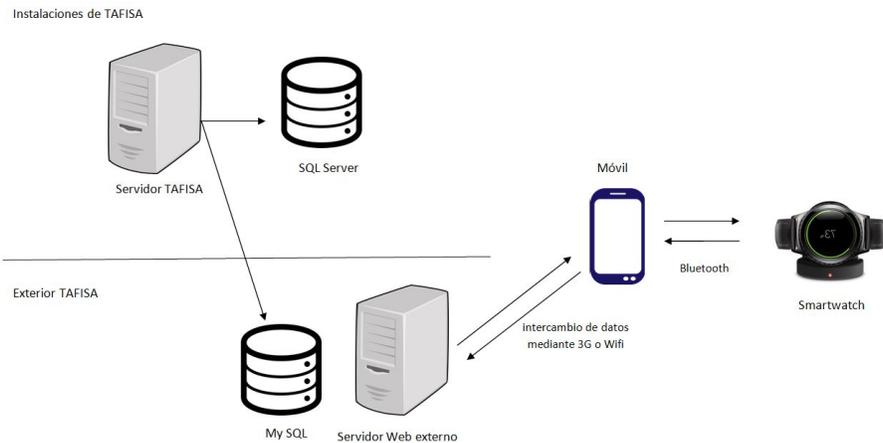


Figure 40: Interconexiones del Sistema de la aplicación del reloj inteligente y la fábrica

Las comunicaciones entre el servidor de la fábrica y el servidor externo, y entre el servidor externo y el teléfono móvil que se sincronizará con el reloj inteligente, se realizarán mediante el uso de una API proporcionada por un Servicio Web de tipo REST creado por un compañero de Innovae. Este Servicio Web servirá para que el administrador de la fábrica al actualizar los datos de su BBDD actualice también los datos de la BBDD externa, y también para que se lleven a cabo las consultas de datos desde el teléfono móvil al servidor externo.

### 8.3 Módulos del Sistema

En este apartado se explican los módulos implementados que componen ambas soluciones, tanto para el reloj inteligente como para las gafas de realidad aumentada.

- En los proyectos para un reloj inteligente con sistema operativo Android Wear, que es para el que se ha implementado, se diferencian dos módulos: un módulo *mobile* donde se implementará todo lo que comprende a la aplicación del teléfono móvil que acompaña al reloj, y otro módulo *wear* donde se implementará todo lo que comprende a la aplicación del reloj inteligente.

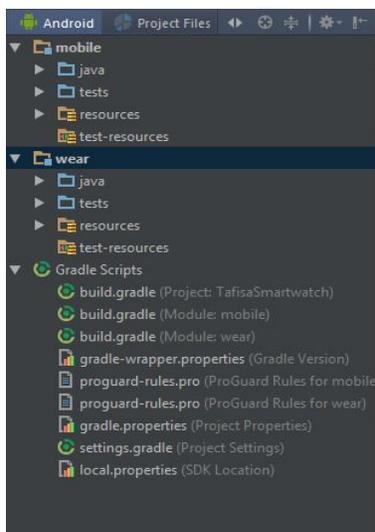


Figure 41: Módulos de la aplicación del reloj

- En el módulo *mobile* se implementó la interfaz de la aplicación del teléfono móvil y las distintas clases y servicios que comprenden la aplicación. Entre los servicios implementados se destacan dos: *ServicioDatosADemanda* y *MyFirebaseMessagingService*.

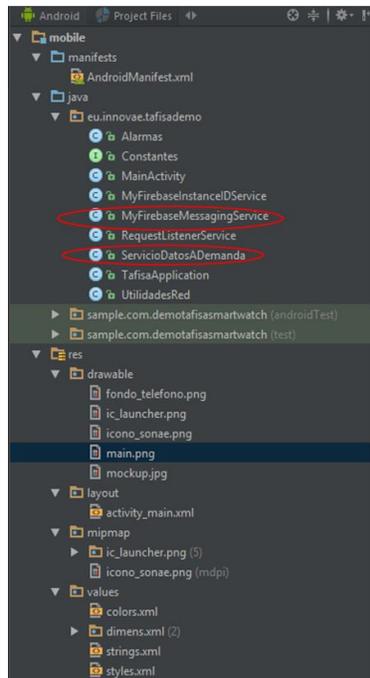


Figure 42: Módulo *mobile* de la aplicación del reloj

- El servicio *ServicioDatosADemanda* es muy importante para la aplicación en general, ya que es el encargado de llevar a cabo la consulta de datos cada minuto haciendo uso del REST API, este servicio fue implementado en el módulo *mobile* debido a que desde el módulo *wear* no se puede llevar a cabo las peticiones a la red directamente, lo cual habría sido lo ideal. De esta manera los datos se consultan desde el módulo *mobile* y luego se envían al módulo *wear*, haciendo uso del Wearable Data Layer API, encargado de la comunicación entre la aplicación del móvil y la del *smartwatch*.<sup>1</sup>

---

<sup>1</sup> Nota: Para este otoño se liberaba la versión de Android Wear 2.0, esta versión trae muchas novedades entre las que se destaca la posibilidad de hacer peticiones directamente a la red desde el módulo wear.

```

public ServicioDatosADemanda() { super(TAG_NOMBRE_SERVICIO); }

public static void activarAlarmaGestionService(Context contexto, boolean activo) {
    Log.d(TAG_NOMBRE_SERVICIO, "activarAlarmaGestionService: " + String.valueOf(activo));
    @Contexto;
    Intent intentServicio = ServicioDatosADemanda.nuevoIntent(contexto);
    PendingIntent tokenIntentServicio = PendingIntent.getService(contexto, ID_ALARMA, intentServicio, 0);

    AlarmManager gestorAlarmasSistema = (AlarmManager)contexto.getSystemService(Context.ALARM_SERVICE);

    if (activo) {
        if (!estaActivoServicioAlarma(contexto)) {
            contexto.startService(intentServicio);
            gestorAlarmasSistema.setInexactRepeating(AlarmManager.ELAPSED_REALTIME_WAKEUP, SystemClock.elapsedRealtime(), INTERVALO_CONSULTA_DATOS, tokenIntentServicio);
        }
    } else {
        gestorAlarmasSistema.cancel(tokenIntentServicio);
        tokenIntentServicio.cancel();
    }
}
}

```

Figure 43: Fragmento de código del servicio *ServicioDatosADemanda*

- El servicio *MyFirebaseMessagingService* también es importante, ya que se encarga de recibir *notificaciones push* para alertar que nuevas alarmas han sido activadas en el servidor, de esta manera al recibir la notificación se inicia la aplicación del *smartwatch* y se muestra la o las nuevas alarmas.

```

public class MyFirebaseMessagingService extends FirebaseMessagingService implements
    GoogleApiClient.ConnectionCallbacks,
    GoogleApiClient.OnConnectionFailedListener,
    DataApi.DataListener,
    MessageApi.MessageListener {

    private static final String TAG = "firemessaging";
    private GoogleApiClient mGoogleApiClient;
    private TafisaApplication instanceTafisaApp;

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        // TODO(developer): Handle FCM messages here.
        // If the application is in the foreground handle both data and notification messages here.
        // Also if you intend on generating your own notifications as a result of a received FCM
        // message, here is where that should be initiated. See sendNotification method below.
        instanceTafisaApp=(TafisaApplication) getApplication();
        mGoogleApiClient = new GoogleApiClient.Builder(this)
            .addApi(Wearable.API)
            .build();

        mGoogleApiClient.connect();
        StartWearableActivity();
        Log.d(TAG, "From: " + remoteMessage.getFrom());
        Log.d(TAG, "Data : " + remoteMessage.getData().get("message"));
    }
}

```

Figure 44: Fragmento de código del servicio *MyFirebaseMessagingService*

- En el módulo *wear* se implementó la interfaz de la aplicación del *smartwatch* y las distintas clases y servicios que comprenden la aplicación del reloj. Entre los servicios implementados se destaca: *MyService*.

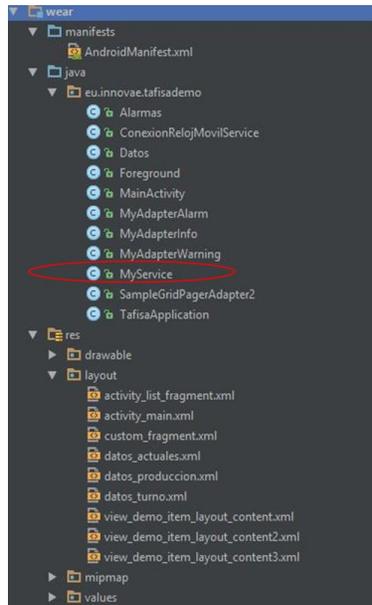


Figure 45: Módulo *wear* de la aplicación del reloj

- El servicio *MyService* es importante, ya que se encarga de recibir los mensajes enviados por el Wearable Data Layer API, es decir, se encarga de recibir los datos (alarmas y datos bajo demanda) y mensajes desde la aplicación del teléfono móvil y los transmite al reloj.

```

public class MyService extends WearableListenerService {
    private static final String START_ACTIVITY_PATH = "/start-activity";
    private static final String EXIT_APP_PATH = "/exit-application";
    private static final String ERROR_PATH = "/error";
    private static final String START_SERVICEBLUETOOTH_PATH = "/start-service-bt";
    private long[] pattern = {0, 500, 50, 300};
    private NotificationCompat.BigTextStyle bigStyle;
    private Handler handler;
    private static final String TAG = "MyService";
    Intent startIntent;
    GoogleApiClient mGoogleApiClient;

    @Override
    public void onCreate() {
        super.onCreate();
        handler = new Handler();
        mGoogleApiClient = new GoogleApiClient.Builder(this)
            .addApi(Wearable.API)
            .build();
        mGoogleApiClient.connect();
    }

    @Override
    public void onMessageReceived(MessageEvent messageEvent) {
        if (messageEvent.getPath().equals(START_ACTIVITY_PATH)) {
            Log.i(TAG, "onMessageReceived");
            startIntent = new Intent(this, MainActivity.class);
            startIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK|Intent.FLAG_ACTIVITY_CLEAR_TASK);
            startActivity(startIntent);
        }
    }
}

```

Figure 46: Fragmento de código del servicio *MyService*

- En lo que respecta a la estructura de los proyectos de las gafas, en este

caso para las *Sony Smarteyeglass*, se compone del propio módulo de la aplicación de las gafas, y de tres módulos adicionales: *SmartExtensionAPI*, *SmartExtensionUtils* y *SmartEyeglassAPI*. Dichos módulos son necesarios para el correcto funcionamiento de la *app* y vienen por defecto en todas las aplicaciones de las gafas de Sony.

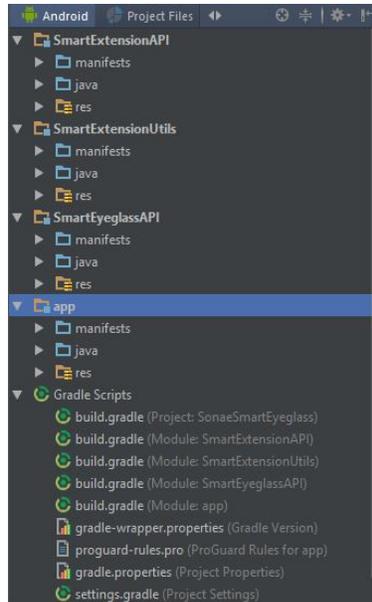


Figure 47: Módulos de la aplicación de las gafas

---

## 9 Pruebas

---

En este apartado se explica las pruebas realizadas para verificar el funcionamiento de las dos aplicaciones desarrolladas durante el proyecto.

### 9.1 Pruebas gafas de realidad aumentada

- Caso de uso Recibir Datos Bajo Demanda

<b>Problema:</b>	No se reciben los datos bajo demanda en segundo plano.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos a Demanda.
<b>Escenario:</b>	Fábrica.
<b>Descripción:</b>	<p>Al realizar pruebas en distintos modelos de teléfonos móviles se evidenció que en algunos los datos bajo demanda se actualizaban siempre, tanto si la aplicación se encontraba en primer como en segundo plano, pero en otros sólo se actualizaban cuando la aplicación estaba ejecutándose en primer plano.</p> <p>Tras investigar un poco e ir descartando posibles fuentes del problema, se averiguó que en ciertos dispositivos si se elige el modo de ahorro de energía se desactivan los datos móviles, Wi-Fi y se deshabilitan los servicios que se ejecutan en background.</p> <ul style="list-style-type: none"><li>- En el Sony Xperia Z3 Compact si el Modo STAMINA se encuentra activado no permite la ejecución de servicios en background.</li><li>- El móvil Huawei P8 por defecto ahorra energía, si no se indica que se desea permitir la ejecución de aplicaciones en segundo plano estas quedarán deshabilitadas.</li></ul>
<b>Solución:</b>	<ul style="list-style-type: none"><li>-En el móvil Sony Xperia Z3 Compact nos dirigimos a Ajustes&gt;Administración de energía&gt;Modo Stamina y lo desactivamos.</li><li>-En el móvil Huawei P8 nos dirigimos al menú de Ajustes, elegimos la opción <i>Power Saving</i>(Ahorro de energía) y buscamos la opción de <i>Protected apps</i>, luego buscamos nuestra aplicación y la habilitamos para que se ejecute en background.</li></ul>

Figure 48: Problema: no se reciben datos bajo demanda en segundo plano

<b>Problema:</b>	No se actualizan los datos bajo demanda.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos a Demanda.
<b>Escenario:</b>	Fábrica.
<b>Descripción:</b>	Al realizar pruebas de la aplicación en la misma fábrica se detectó que había ciertas localizaciones en las que la red Wi-Fi perdía calidad, por tanto los datos bajo demanda no se actualizaban.
<b>Solución:</b>	-En la misma fábrica decidieron mejorar su infraestructura Wi-Fi para que no se pierda calidad en ningún lugar de la misma. -Como modo alternativo, cuando se pierda la conexión con la red teniendo el Wi-Fi activado la aplicación envía un mensaje de error, si el operario ve este mensaje puede optar por activar los datos móviles del teléfono para mantener la conexión.

Figure 49: Problema: no se actualizan los datos bajo demanda

<b>Problema:</b>	No se visualiza la interfaz donde se muestran los datos bajo demanda.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos a Demanda.
<b>Escenario:</b>	Empresa del proyecto, Fábrica.
<b>Descripción:</b>	Al tratar de visualizar los datos bajo demanda en la interfaz de las gafas no nos mostraba nada y se quedaba la pantalla vacía.
<b>Solución:</b>	En el AndroidManifest.xml de la aplicación hay un Broadcast Receiver ( <i>SonaeExtensionReceiver</i> ), este receiver se encarga de responder ante todos los eventos de las gafas, dentro de este receiver en su <intent-filter> hay que añadir las acciones que corresponden a mostrar un menú en las gafas. En nuestro caso fueron: <pre>&lt;action android:name="com.sonyericsson.extras.aef.control.LIST_REFERESH_REQUEST" /&gt; &lt;action android:name="com.sonyericsson.extras.aef.control.LIST_REQUEST_ITEM" /&gt; &lt;action android:name="com.sonyericsson.extras.aef.control.LIST_ITEM_CLICK" /&gt; &lt;action android:name="com.sonyericsson.extras.aef.control.LIST_ITEM_SELECTED" /&gt; &lt;action android:name="com.sonyericsson.extras.aef.control.MENU_ITEM_SELECTED" /&gt; &lt;action android:name="com.sonyericsson.extras.aef.control.MENU_SHOW" /&gt;</pre>

Figure 50: Problema: no se visualiza la interfaz de datos bajo demanda

- Caso de uso Tomar Fotos

<b>Problema:</b>	No se inicia la cámara de las gafas de realidad aumentada.
<b>Caso de Uso del problema:</b>	Caso de Uso Tomar Fotos.
<b>Escenario:</b>	Empresa del proyecto.
<b>Descripción:</b>	Al tratar de entrar al Modo Cámara de las gafas, tras establecer el IP del servidor, no se inicia la cámara de las gafas.
<b>Solución:</b>	-En el AndroidManifest.xml de la aplicación hay que añadir los permisos respectivos a la cámara de las gafas. Estos permisos serían: <code>&lt;uses-permission android:name="com.sony.smarteyeglass.permission.SMARTEYEGGLASS" /&gt;</code> -Si se añade este permiso y sigue sin iniciarse la cámara de las gafas, nos dirigimos desde el mismo teléfono móvil a los Ajustes>Aplicaciones>Administrador de Aplicaciones> buscamos nuestra aplicación y nos aseguramos que los permisos del manifest se encuentren activados en el teléfono, si no lo están los activamos.

Figure 51: Problema: no se inicia la cámara de las gafas de realidad aumentada

<b>Problema:</b>	No se reciben las imágenes tomadas por las gafas en el servidor.
<b>Caso de Uso del problema:</b>	Caso de Uso Tomar Fotos.
<b>Escenario:</b>	Empresa del proyecto, Fábrica.
<b>Descripción:</b>	Al tomar fotos con la cámara incorporada de las gafas estas no se envían al servidor.
<b>Solución:</b>	-Al establecer la IP del servidor donde se guardarán las fotos, tanto el teléfono móvil como el ordenador, deben encontrarse dentro de la misma red para que se envíen las fotos y se reciban del lado servidor.

Figure 52: Problema: no se reciben las imágenes tomadas por las gafas en el servidor

## 9.2 Pruebas reloj inteligente

- Caso de Uso Recibir Datos Bajo Demanda

<b>Problema:</b>	No se reciben los datos bajo demanda en segundo plano.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos Bajo Demanda.
<b>Escenario:</b>	Fábrica.
<b>Descripción:</b>	Al realizar pruebas en distintos modelos de teléfonos móviles se evidenció que en algunos los datos bajo demanda se actualizaban siempre, tanto si la aplicación se encontraba en primer como en segundo plano, pero en otros sólo se actualizaban cuando la aplicación estaba ejecutándose en primer plano. Tras investigar un poco e ir descartando posibles fuentes del problema, se averiguó que en ciertos dispositivos si se elige el modo de ahorro de energía se desactivan los datos móviles, Wi-Fi y se deshabilitan los servicios que se ejecutan en background. - En el Sony Xperia Z3 Compact si el Modo STAMINA se encuentra activado no permite la ejecución de servicios en background. - El móvil Huawei P8 por defecto ahorra energía, si no se indica que se desea permitir la ejecución de aplicaciones en segundo plano estas quedarán deshabilitadas.
<b>Solución:</b>	-En el móvil Sony Xperia Z3 Compact nos dirigimos a Ajustes>Administración de energía>Modo Stamina y lo desactivamos. -En el móvil Huawei P8 nos dirigimos al menú de Ajustes, elegimos la opción <i>Power Saving</i> (Ahorro de energía) y buscamos la opción de <i>Protected apps</i> , luego buscamos nuestra aplicación y la habilitamos para que se ejecute en background.

Figure 53: Problema: no se reciben datos bajo demanda en segundo plano

<b>Problema:</b>	No se actualizan los datos bajo demanda.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos Bajo Demanda.
<b>Escenario:</b>	Fábrica.
<b>Descripción:</b>	Al realizar pruebas de la aplicación en la misma fábrica se detectó que había ciertas localizaciones en las que la red Wi-Fi perdía calidad, por tanto los datos bajo demanda no se actualizaban.
<b>Solución:</b>	-En la misma fábrica decidieron mejorar su infraestructura Wi-Fi para que no se pierda calidad en ningún lugar de la misma. -Como modo alternativo, cuando se pierda la conexión con la red teniendo el Wi-Fi activado la aplicación envía un mensaje de error, si el operario ve este mensaje puede optar por activar los datos móviles del teléfono para mantener la conexión. -El mensaje de error se envía al reloj para alertar al operario que sus datos no están actualizados.

Figure 54: Problema: no se actualizan los datos bajo demanda

<b>Problema:</b>	La pantalla del reloj se refresca cada vez que recibe nuevos datos bajo demanda.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos Bajo Demanda.
<b>Escenario:</b>	Empresa del proyecto.
<b>Descripción:</b>	Cada vez que se actualizan los datos la pantalla del reloj se pone en negro y vuelve a cargarse el contenido con los datos bajo demanda actualizados.
<b>Solución:</b>	-Este problema se arreglo utilizando la función <i>notifyDataSetChanged()</i> de la clase <i>Adapter</i> , éste método alerta que los datos que posee el <i>Adapter</i> han cambiado y cualquier vista que tenga los datos debe actualizarse por sí misma. De esta manera evitamos que la pantalla se ponga en negro y vuelva a cargarse todo el contenido.

Figure 55: Problema: La pantalla del reloj se refresca cada vez que recibe nuevos datos bajo demanda

- **Caso de Uso Recibir Datos de Alarmas**

<b>Problema:</b>	No se reciben los datos de alarmas en segundo plano.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos de Alarmas.
<b>Escenario:</b>	Empresa del Proyecto, Fábrica.
<b>Descripción:</b>	Al realizar pruebas en distintos modelos de teléfonos móviles se evidenció que en algunos las alarmas cuando se activaban en el servidor aparecían en el reloj, tanto si la aplicación se encontraba en primer como en segundo plano, pero en otros sólo aparecían cuando la aplicación estaba ejecutándose en primer plano. Tras investigar un poco e ir descartando posibles fuentes del problema, se averiguó que en ciertos dispositivos si se elige el modo de ahorro de energía se desactivan los datos móviles, Wi-Fi y se deshabilitan los servicios que se ejecutan en background. - En el Sony Xperia Z3 Compact si el Modo STAMINA se encuentra activado no permite la ejecución de servicios en background. - El móvil Huawei P8 por defecto ahorra energía, si no se indica que se desea permitir la ejecución de aplicaciones en segundo plano estas quedarán deshabilitadas.
<b>Solución:</b>	-En el móvil Sony Xperia Z3 Compact nos dirigimos a Ajustes>Administración de energía>Modo Stamina y lo desactivamos. -En el móvil Huawei P8 nos dirigimos al menú de Ajustes, elegimos la opción <i>Power Saving</i> (Ahorro de energía) y buscamos la opción de <i>Protected apps</i> , luego buscamos nuestra aplicación y la habilitamos para que se ejecute en background.

Figure 56: no se reciben los datos de alarmas en segundo plano

<b>Problema:</b>	No se reciben las notificaciones <i>push</i> cuando se activa una alarma en el servidor.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos de Alarmas.
<b>Escenario:</b>	Empresa del Proyecto, Fábrica.
<b>Descripción:</b>	Al activar alarmas desde el servidor no llega la notificación <i>push</i> al teléfono móvil y por tanto no se muestra la nueva alarma activada.
<b>Solución:</b>	Este problema puede originarse por muchos motivos pero todos tienen que ver con la configuración de <i>Firebase</i> en nuestro proyecto: - Asegurarnos que tenemos en nuestro módulo <i>mobile</i> de la aplicación el fichero <i>google-services.json</i> . - Añadir en el <i>gradle</i> del módulo <i>mobile</i> las dependencias referentes a <i>firebase</i> y a los servicios de Google. En nuestro caso fueron: <pre>compile 'com.google.android.gms:play-services:9.2.0'</pre> <pre>compile 'com.google.firebase:firebase-messaging:9.2.0'</pre> - Añadir en el <i>AndroidManifest.xml</i> del módulo <i>mobile</i> los servicios <i>FirebaseInstanceIdService</i> y <i>FirebaseMessagingService</i> .

Figure 57: no se reciben las notificaciones push cuando se activa una alarma en el servidor

<b>Problema:</b>	No se reciben las alarmas por problemas con la selección del usuario.
<b>Caso de Uso del problema:</b>	Caso de Uso Recibir Datos de Alarmas.
<b>Escenario:</b>	Empresa del Proyecto, Fábrica.
<b>Descripción:</b>	Si al entrar a la aplicación elegimos un usuario y después desde otro móvil se elige el mismo usuario, el último que eligió será el que reciba las notificaciones push cuando cambie el estado de las alarmas.
<b>Solución:</b>	Este problema no se resolvió pero se avisó a la empresa cliente que sólo puede haber un móvil con un usuario 1 y otro móvil con un usuario 2. Como continuación del proyecto se podría resolver este problema y dar alta a más usuarios.

Figure 58: no se reciben las alarmas por problemas con la selección del usuario

- Otras pruebas

<b>Problema:</b>	No se reciben datos por desconexión Bluetooth entre teléfono y reloj.
<b>Caso de Uso del problema:</b>	Todos.
<b>Escenario:</b>	Empresa del Proyecto, Fábrica.
<b>Descripción:</b>	Hay veces en que la conexión por Bluetooth falla, tanto por la lejanía entre los dispositivos como por interferencias, esto causa que no se puedan transferir datos del teléfono al reloj y viceversa.
<b>Solución:</b>	Se creó un servicio <i>ConexionRelojMovilService</i> que se encuentra chequeando constantemente la conexión entre el teléfono y el reloj, si detecta que ambos dispositivos se han desconectado envía una alerta al reloj para que el usuario sepa y trate de recuperar la conexión nuevamente.

Figure 59: no se reciben datos por desconexión Bluetooth entre teléfono y reloj

---

## 10 Gestión

---

En este capítulo se presenta la gestión llevada a lo largo del ciclo de vida del proyecto.

### 10.1 Comparación de Horas Estimadas y Horas Dedicadas

En este apartado se muestra una comparativa entre las horas previamente estimadas para el proyecto (capítulo 3) y las horas que se dedicaron verdaderamente para el desarrollo del mismo.

FASES	TAREAS	HORAS ESTIMADAS	HORAS DEDICADAS	DESVIACIONES
GESTIÓN DEL PROYECTO	Planificación	10 horas	10 horas	0
	Seguimiento y Control	10 horas	10 horas	0
	Memoria	70 horas	70 horas	0
APLICACIÓN	Estudio (Formación)	40 horas	40 horas	0
	Diseño	30 horas	35 horas	+5
	Implementación	120 horas	130 horas	+10
	Pruebas	20 horas	25 horas	+5
TOTAL		300 horas	320 horas	20 horas

Figure 60: Comparativa Horas Estimadas y Horas Dedicadas

### 10.2 Desviaciones significativas del proyecto

En este apartado se presenta y se explica el motivo de las desviaciones que ha tenido el proyecto frente a lo planificado.

Tras observar la tabla anterior puede comprobarse que ha habido una pequeña desviación entre las horas estimadas y las dedicadas, pero dicha desviación no supone un contratiempo para llevar a cabo el proyecto en las fechas establecidas.

#### 10.2.1 Desviaciones de Diseño

En el apartado de diseño se presentó una desviación debido a que tras establecer un diseño de interfaz inicial para la aplicación del reloj, la empresa cliente sugería nuevos cambios en la interfaz que tenían que cambiarse sobre la marcha. Por ejemplo, en las pantallas de la aplicación del reloj donde se muestran las listas de alarmas activadas por su tipo, en un principio solo se iban a mostrar tres

alarmas por pantalla (las últimas tres alarmas activadas por cada tipo), pero al ir avanzando en el proyecto se decidió mostrar todas las alarmas que estén activadas, por ello se creó una lista desplegable en la que podamos movernos haciendo *scroll*.

### 10.2.2 Desviaciones de Implementación

Por otro lado, las desviaciones en el apartado de implementación se debieron a la demora en la disponibilidad de los datos por parte de la empresa cliente, ya que tenían que definir que datos querían mostrar y que los textos de dichos datos sean los adecuados para su interpretación por parte de los trabajadores. Por ejemplo, debido a que la pantallas del reloj son pequeñas los encargados de definir los datos tenían que pensar en la mejor forma de resumir dichos datos, de manera que sean útiles y entendibles por los operarios.

Además, la mayoría de los operarios de la fábrica utilizan gafas, por lo que era necesario que los textos se vieran lo suficientemente grandes tras haber definido una primera vez los datos y haber entregado una versión de la aplicación. Los datos tuvieron que ser acortados más para poder aumentar el tamaño de los mismos y que se sigan viendo bien en la pantalla del reloj.

### 10.2.3 Desviaciones de Pruebas

Por último, las desviaciones en el apartado de pruebas se debieron a que tras ir entregando versiones de la aplicación a la empresa cliente para su testeo en la fábrica, se nos comunicara de ciertos errores o correcciones en la aplicación,. Por ejemplo, tras entregar una de las primeras versiones de la aplicación, se nos avisó de una serie de errores entre los que estaban:

- Tras activarse las alarmas y notificar su lectura por parte del operario volvían a aparecer dichas alarmas en la pantalla del reloj.
- Cuando se perdía la conexión entre el teléfono y el reloj, el mensaje de fallo de conexión no se enviaba al reloj.
- Al añadir una nueva alarma a la base de datos no se registraba bien. Cuando consultaban la BBDD no veían la nueva alarma y por tanto tampoco llegaba al reloj.

Tras ir corrigiendo los errores que nos comunicaba la empresa cliente, se volvía a enviar nuevas versiones de la aplicación con los *bugs* arreglados hasta que todas las pruebas fueron positivas.

---

## 11 Conclusiones

---

Tras haber afrontado el desarrollo de este proyecto se presentan una serie de conclusiones sobre el trabajo desarrollado y el cumplimiento de los objetivos, y otras conclusiones a nivel personal.

### 11.1 Conclusiones del Trabajo

El objetivo de este proyecto consistió en demostrar el uso potencial de los smart-devices en el ámbito industrial, para ello se realizó la implementación de dos aplicaciones que expriman las capacidades de estos dispositivos, tanto para un reloj inteligente como para unas gafas de realidad aumentada.

A continuación analizamos si se han cumplido los objetivos y tareas definidos en el apartado de planificación del proyecto.

- **Estudio (Formación)**

- Se estudió como realizar aplicaciones en Android mediante el diseño y la implementación de pequeñas aplicaciones básicas para ganar experiencia con el código y su estructura.
- También se leyó y familiarizó con el API de Sony y con las librerías de las gafas de realidad aumentada. Se revisaron una serie de ejemplos de proyectos de las gafas de Sony, que se encontraban en su página web para desarrolladores.
- El estudio del API y librerías de Android Wear mediante el desarrollo de aplicaciones básicas fue muy útil. También desde su página web se revisó una serie de proyectos de ejemplo que sirvieron para saber cómo diseñar la interfaz, las comunicaciones entre el móvil y el reloj, etc.

- **Diseño**

- Se realizó el diseño de las interfaces de ambas aplicaciones, en el caso de la aplicación de las gafas este diseño fue idea mía, pero en el caso de la interfaz del reloj el diseño fue acordado entre la empresa cliente e Innovae, tras revisar las distintas posibilidades de cómo mostrar la información en este dispositivo.

- **Implementación**

- **Gafas de Realidad Aumentada:** Se implementó las dos funcionalidades definidas en el apartado de objetivos del proyecto.
  - \* La funcionalidad modo datos bajo demanda permite al operario consultar una serie de datos cuando lo desee por tanto estos datos deben estar constantemente actualizándose. Además un compañero de Innovae, que se encargó del lado backend, creó un Servicio Web de tipo REST con una serie de funciones de consulta, modificación y actualización de éstos datos. Para llevar a cabo esta actualización constante de los datos fue necesaria la creación de un Servicio y el uso de un Gestor de Alarmas (Android AlarmManager), de esta manera se tenía al servicio ejecutándose en segundo plano realizando la consulta para obtener los datos bajo demanda cada 10 segundos. A continuación la respuesta de dicha consulta era un JSON al que se le aplicaba un parser que lo convierta en objetos de tipo Datos Bajo Demanda, que más adelante serían mostrados al usuario.
  - \* La funcionalidad modo cámara permite al operario tomar una serie de fotos, con la cámara de las gafas, y enviarlas a un ordenador, para que estas sean revisadas al momento o posteriormente por otro usuario. Dicha transmisión se lleva cabo mediante sockets, para ello nos basamos en la arquitectura cliente-servidor, donde la aplicación de las gafas es el lado cliente y el ordenador que recibe las imágenes funciona como la parte servidor.
- **Reloj Inteligente:** Se implementó las tres funcionalidades del *smart-watch* también definidas en el apartado de objetivos.
  - \* La funcionalidad modo datos bajo demanda es la misma que se explicó previamente para las gafas de realidad aumentada, sólo que en vez de mostrarse la información en las lentes de unas gafas se mostrarán en el reloj del operario.
  - \* La funcionalidad modo alarmas, al contrario que la de datos bajo demanda, muestra la información de las alarmas cuando estas se activan en el servidor del cliente. Para llevar a cabo esto se ha utilizado Firebase Cloud Messaging (FCM), es una herramienta de Google que permite enviar notificaciones o mensajes push cuando detecta que en el servidor hay datos nuevos disponibles para su sincronización. De esta manera se consigue que las alarmas al activarse en el servidor cliente lleguen al reloj en tiempo real.
  - \* También se pidió la posibilidad de poder recibir recordatorios o notificaciones en el reloj asociadas a la cuenta de Google vinculada al teléfono móvil, pero esta funcionalidad no supuso implementación alguna, ya que tras sincronizar el teléfono móvil con un reloj por defecto se sincronizan los datos asociados a la cuenta de Google.

- **Pruebas**

- A lo largo del desarrollo del proyecto, mientras se implementó cada una de las funcionalidades de ambas aplicaciones (gafas y reloj) se iban haciendo pruebas para asegurarnos de su correcto funcionamiento. En caso de que hubieran errores o fallos se corregían y se volvían a hacer pruebas y así sucesivamente.

## 11.2 Conclusiones Personales

Tras mi estancia en una empresa, tanto para el desarrollo de las prácticas como para la elaboración de mi Trabajo Fin de Grado, he aprendido una serie de cosas a nivel personal que resumo a continuación en forma de conclusiones:

- El desarrollo de aplicaciones y proyectos en la Universidad es una muy buena práctica para afrontar nuestro futuro, pero la experiencia y aptitudes que brinda el trabajar en un entorno laboral con un equipo de trabajo, bajo la supervisión de alguien y tomando decisiones, es algo igual o más importante.
- La oportunidad de poder trabajar en Innovae, que se dedica al desarrollo de soluciones informáticas haciendo uso de nuevas tecnologías, ha sido una experiencia muy positiva. No sólo por la experiencia que se ha conseguido, sino también por haber tenido la oportunidad de desarrollar mi proyecto haciendo uso de dispositivos de última generación, como son los *smartwatches* y las gafas de realidad aumentada.
- Durante mi estancia en la Universidad no se estudió mucho acerca del desarrollo de aplicaciones móviles para el sistema operativo Android, pero afrontar este proyecto requirió un estudio y formación previa en dicha tecnología. La necesidad de adaptarte a nuevas tecnologías y la búsqueda de información es algo que si se aprendió en el día a día durante mi etapa universitaria. Como buenos informáticos siempre tenemos que estar preparados e informados de la nuevas tecnologías y tendencias que rodean nuestro oficio.
- Son de mucha importancia las reuniones con la empresa cliente, para poder definir los requisitos y la especificación del proyecto, de esta manera quedan claros los objetivos a realizar, los plazos para cumplir dichos objetivos y la ejecución de los mismos.
- La oportunidad de poder desarrollar una aplicación para un reloj inteligente, que comenzó siendo una demo, pero que finalmente será utilizada realmente en una fábrica, es algo que me llena de mucha alegría y satisfacción. Es muy positivo ver como el esfuerzo de tantos años de estudio concluye en la creación de algo que servirá de utilidad para terceros.

---

## 12 Líneas Futuras

---

Tras la elaboración de este Trabajo Fin de Grado, se ha conseguido demostrar las capacidades que tienen los dispositivos inteligentes para ser utilizados en entornos industriales, de manera que ofrezcan a los trabajadores nuevas maneras de poder ver la información y que les sea más accesible. Pero aún queda mucho camino por recorrer y muchas cosas nuevas por probar. Como extensión de este proyecto se proponen una serie de posibilidades a futuro tanto para la empresa donde desarrollé el proyecto, Innovae Augmented Reality, como para la empresa cliente.

Para la empresa Innovae se propone:

- La posibilidad de probar el desarrollo de aplicaciones para *smartwatch* pero enfocándose en otras plataformas, por ejemplo para IOS (Apple Watch) o para Tizen (Samsung Gear S2), de esta manera se pueden estudiar las características que nos ofrecen dichos sistemas, y ampliar el abanico de dispositivos para los que desarrollar.
- La posibilidad de probar el desarrollo de aplicaciones para gafas de realidad aumentada pero enfocándose en otros dispositivos, por ejemplo para las gafas Hololens (Microsoft), de esta manera se pueden estudiar las nuevas características que nos ofrecen estas gafas, y explotar una amplia gama de nuevas funcionalidades para desarrollar.
- Explotar el desarrollo de aplicaciones para otros dispositivos inteligentes como pueden ser drones, *smartbands*, etc.

Para la empresa cliente se propone:

- La incorporación de nuevos dispositivos inteligentes como las gafas de realidad aumentada, pero cuando se resuelvan los problemas con los certificados para poder ser usadas en una fábrica.
- La ampliación de la aplicación realizada permitiendo dar de alta a un mayor número de usuarios, y que además de ver la información de los datos y las alarmas solo en reloj, también pueda verse desde la aplicación del teléfono móvil.<sup>1</sup>

---

<sup>1</sup>Nota: La ampliación de la aplicación realizada está siendo valorada actualmente para su posible desarrollo por ambas empresas.

- El despliegue de la aplicación desarrollada a otras fábricas de la misma empresa.

---

## 13 Bibliografía

---

- [1] José María Megino Barquinero. Tipos de relaciones en diagramas de casos de uso. uml., 2013. URL: <http://www.seas.es/blog/informatica/tipos-de-relaciones-en-diagramas-de-casos-de-uso-uml/>.
- [2] Pablo G. Bejerano. Tecnología wearable para modernizar el mundo industrial. 2014. URL: <http://blogthinkbig.com/wearable-en-la-industria/>.
- [3] Instituto Tecnológico de Aragón. Análisis: Wearables en el entorno de las factorías del futuro. Technical report, Gobierno de Aragón, 2014.
- [4] David Millan Escrivá. *Posibilidades de la Realidad Aumentada en entornos laborables. Estudio comparativo entre dispositivos video see-through y optical see-through*. PhD thesis, Universidad Politécnica de Valencia, 2014.
- [5] Firebase. Firebase cloud messaging, 2012. URL: <https://firebase.google.com/docs/cloud-messaging/?hl=es>.
- [6] Jesús Tomás Gironés. *El gran libro de Android (4º Edición)*. 2014.
- [7] Alex Ho. Android wear book. the gridviewpager layout sample., 2015. URL: <http://www.learnandroidwear.com/sample-3/>.
- [8] LyX. Lyx - the document processor, 1999. Url: <https://www.lyx.org/Home>.
- [9] MKLab. Staruml., 2014. URL: <http://staruml.io/>.
- [10] Salvador Gómez Oliver. Tareas en segundo plano en android (ii): Intentser-vice, 2012. URL: <http://www.sgoliver.net/blog/tareas-en-segundo-plano-en-android-ii-intentservice/>.
- [11] James Revelo. Operaciones http en android con el cliente httpURLConnection, 2015. URL: <http://www.hermosaprogramacion.com/2015/01/android-httpURLConnection/>.
- [12] James Revelo. Parsear datos json en android con jsonreader y gson, 2015. URL: <http://www.hermosaprogramacion.com/2015/01/android-json-parsing/>.

- [13] Sony. Develop apps for smarteyeglass, 2016. URL: <https://developer.sony.com/develop/wearables/smarteyeglass-sdk/>.
- [14] Android Studio. Cómo descargar android studio y sdk tools., 2014. URL: <https://developer.android.com/studio/index.html?hl=es-419>.
- [15] Synergix. Tipos de requisitos: Funcional vs. no funcional, 2008. URL: <https://synergix.wordpress.com/2008/07/07/requisito-funcional-y-no-funcional/>.
- [16] Jesús Tomás, Antonio Albiol, Mohamed Falhi, and Vicente Carbonell. *Dispositivos Wearables, Vision artificial, Google Glass y Android TV*. 2015.
- [17] Android Wear. Android wear developer, 2015. URL: <https://developer.android.com/wear/index.html>.
- [18] Ahmet Yildirim. Livestreaming app for smarteyeglass available as open source project, 2016. URL: <http://developer.sonymobile.com/2016/01/28/livestreaming-app-for-smarteyeglass-available-as-open-source-project/>.

---

## 14 Anexos

---

En este apartado se muestra las actas de reunión realizadas a lo largo del desarrollo del Trabajo Fin de Grado, tanto las que se llevaron a cabo con mi director del proyecto, como las realizadas en la empresa.

### 14.1 Actas de Reunión: Director Proyecto

#### ACTA DE REUNIÓN: DIRECTOR TFG

##### ACTA N° 1

**FECHA:** 28 de Junio de 2016

**ASUNTO:** Exposición al profesor de mi propuesta de proyecto para que acepte ser mi director.

**LUGAR:** Facultad de Informática.

- **ASISTENTES:**

- Carlos Zamora Soliz

- José Ángel Vadillo.

- **ORDEN DEL DÍA:**

- Exposición de mi propuesta para el TFG.

- **DESARROLLO:**

Se expuso al profesor lo que había hecho en la empresa durante mi estancia en las prácticas, para que tenga conocimientos de lo que había estado haciendo hasta ahora. Luego se expuso la idea del proyecto y se explicó cómo surgió. Y por último, se explicó los objetivos y el alcance que iba a tener el proyecto.

- **DECISIONES TOMADAS:**

- El profesor José Ángel Vadillo accedió a ser mi director de proyecto.

- **TAREAS A REALIZAR:**

- Empezar con la redacción de la memoria del TFG.

Figure 61: Acta de Reunión 1

## ACTA DE REUNIÓN: DIRECTOR TFG

### ACTA N° 2

**FECHA:** 11 de Octubre de 2016

**ASUNTO:** Mostrar al director el avance en el desarrollo del TFG.

**LUGAR:** Facultad de Informática.

- **ASISTENTES:**

- Carlos Zamora Soliz
- José Ángel Vadillo

- **ORDEN DEL DÍA:**

- Mostrar al director todo lo avanzado en el desarrollo de las aplicaciones del TFG.
- Mostrar al director todo lo avanzado en el desarrollo de la documentación del TFG.

- **DESARROLLO:**

Se muestra al profesor lo que se ha avanzado en el desarrollo de las aplicaciones (reloj y gafas), ambas aplicaciones se encuentran terminadas. También se muestra lo que se ha avanzado en la documentación del TFG, en este caso nos falta una serie de apartados para terminarlo.

- **DECISIONES TOMADAS:**

- Corrección de una serie de errores en la documentación y mover apartados que no están en su lugar.
- Realizar los apartados que faltan: Diseño, Implementación, Pruebas, Gestión, Conclusiones, Líneas Futuras, Bibliografía y Anexos.

- **TAREAS A REALIZAR:**

- Finalizar la redacción de la memoria del TFG.
- Enviar al director un borrador completo de la memoria para el 24 de Octubre de 2016.

Figure 62: Acta de Reunión 2

## 14.2 Actas de Reunión: Empresa

### ACTA DE REUNIÓN: EMPRESA

#### ACTA N° 1

**FECHA:** 24 de Mayo de 2016

**ASUNTO:** Definir la interfaz de usuario que tendrá la aplicación del *smartwatch*.

**LUGAR:** Empresa Innovae Augmented Reality.

- **ASISTENTES:**

- Carlos Zamora Soliz
- Mikel Pajares (Supervisor)

- **ORDEN DEL DÍA:**

- Reunión con el supervisor de la empresa para definir la interfaz que tendrá la aplicación del reloj.

- **DESARROLLO:**

Tras haber leído y estudiado las librerías y el API de Android Wear, se realizó una reunión para tratar de definir cómo quedaría la interfaz de la aplicación del reloj, de manera que se ajuste a las funcionalidades que hay que implementar y que sea intuitiva para los trabajadores.

- **DECISIONES TOMADAS:**

- Se define la interfaz por parte de nosotros, pero se espera la aprobación de dicha interfaz por parte de la empresa cliente.

- **TAREAS A REALIZAR:**

- Seguir dándole vueltas a otras posibilidades de interfaz, en caso de que la que se plantea no sea aprobada.
- Estudiar la posibilidad de que la interacción por parte de los operarios con el reloj, pueda llevarse a cabo por gestos, en vez de tener que tocar la pantalla del reloj.

Figure 63: Acta de Reunión 1

## ACTA DE REUNIÓN: EMPRESA

### ACTA N° 2

**FECHA:** 8 de Julio de 2016

**ASUNTO:** Skype con el encargado de tecnología de la empresa cliente.

**LUGAR:** Empresa Innovae Augmented Reality.

- **ASISTENTES:**

- Carlos Zamora Soliz
- Mikel Pajares (Supervisor)
- Ángel García Bombin (Jefe de Producción)

- **ORDEN DEL DÍA:**

- Skype con el encargado de tecnología de la empresa cliente para darnos su *feedback* de la aplicación.

- **DESARROLLO:**

Se revisaron las interfaces de todas las secciones de la aplicación del reloj, y se aclaró una serie de cuestiones que tienen que ver con la actualización de los datos en el servidor.

- **DECISIONES TOMADAS:**

- Se aprobó la interfaz del apartado de alarmas de la aplicación.
- En el apartado de datos bajo demanda, en la sección de resumen turno, corrección en los textos que se visualizan.
- En el apartado de datos bajo demanda, en la sección de valores actuales, corrección en los textos que se visualizan.
- En el apartado de datos bajo demanda, en la sección de informe producción, corrección en los textos que se visualizan.
- Aumentar el tamaño de los textos.

- **TAREAS A REALIZAR:**

- Corregir los textos de los apartados indicados y aumentar su tamaño.

Figure 64: Acta de Reunión 2