



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de un software de gestión de partes de
trabajo en movilidad para una empresa de
prestación de servicios integrales

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Sergio Rubio Salvador

Tutores: Maria Alpuente Frasnado / Roberto Monfort Casañ

2018/2019

Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales



Resumen

El objetivo de este trabajo es realizar la migración de una aplicación de gestión de partes en movilidad para una compañía que provee servicios globales esenciales, incluido el sector de las telecomunicaciones. A su vez, esto supone realizar tareas de mantenimiento dedicadas a mejorar la aplicación actual, así como el estudio de los principales *frameworks* de desarrollo y entornos de programación que pueden utilizarse para modernizar la aplicación anterior. Esto implica el uso de técnicas de ingeniería inversa para migrar las funcionalidades de la aplicación actual y el uso de patrones de diseño que mejorarán la calidad del código, mejorando así la experiencia del usuario.

Palabras clave: movilidad, ingeniería inversa, mantenimiento, experiencia del usuario.

Abstract

The goal of this work is to migrate a legacy application for worksheet management in mobility for a company that provides essential global services, including the telecommunication sector. This involves carrying out maintenance tasks that are aimed to refine the current application, as well as a study of the main development frameworks and programming environments that can be used to revamp the legacy application. It also involves the use of reverse engineering practices to migrate the functionalities of the current application and design patterns to improve the quality of the code, so that the user experience can be also improved.

Keywords: mobility, reverse engineering, maintenance, user experience.

Resum

L'objectiu d'aquest treball es realitzar la migració d'una aplicació de gestió de parts en mobilitat para una companyia que proveeix servicis globals essencials, inclòs el sector de las telecomunicacions. Així mateixa, acó implica realitzar tasques de manteniment dedicades a millorar la aplicació actual, aixina com l'estudi dels principals *frameworks* de desenvolupament y entorns de programació que poden utilitzar-se per a modernitzar l'aplicació anterior. Açò implica l'ús de tècniques de enginyeria inversa per migrar les funcionalitats de l'aplicació actual y l'ús de patrons de disseny que milloraran la qualitat del codi, millorant d'aquesta forma la experiència de l'usuari.

Paraules clau: mobilitat, enginyeria inversa, manteniment, experiència de l'usuari.

Tabla de contenidos

1.	Introducción.....	9
1.1.	Motivación.....	9
1.2.	Objetivos.....	9
1.3.	Impacto esperado.....	10
1.4.	Estructura.....	10
1.5.	Convenciones.....	11
2.	Estado del arte.....	13
2.1.	Crítica al estado del arte.....	13
2.2.	Propuesta.....	13
3.	Análisis del problema.....	17
3.1	Análisis de la seguridad.....	17
3.2.	Análisis del marco legal y ético.....	17
3.2.1.	Análisis de la protección de datos.....	17
3.2.2.	Propiedad intelectual.....	17
3.3.	Análisis de riesgos.....	17
3.4.	Identificación y análisis de soluciones posibles.....	18
3.5.	Solución propuesta.....	20
3.6.	Plan de trabajo.....	21
3.7.	Presupuesto.....	21
4.	Diseño de la solución.....	23
4.1.	Arquitectura del sistema.....	23
4.2.	Diseño detallado.....	23
4.3.	Tecnologías utilizadas.....	24
5.	Desarrollo de la solución propuesta.....	27
5.1.	Añadir un campo en el formulario de introducción de trabajos.....	28
5.2.	Realizar un nuevo informe de producción.....	28
5.3.	Diseño gráfico (Prototipado).....	31
5.3.1.	Log-in.....	32
5.3.2.	Inicio.....	33
5.3.3.	Menú.....	34



5.3.4. Parte.....	35
5.3.5. Nuevo medio.....	36
5.3.6. Nuevo operario.....	37
5.3.7. Nuevo trabajo.....	38
5.3.8. Ajustes.....	39
5.4. Implementación de la solución.....	40
5.4.1. Inicio de sesión.....	40
5.4.2. Recuperar contraseña.....	41
5.4.3. Pantalla de inicio.....	41
5.4.4. Pantalla de datos del parte.....	42
5.4.5. Pantalla añadir operario.....	44
5.4.6. Pantalla añadir medio.....	45
5.4.7. Pantalla añadir trabajo.....	45
5.4.8. Pantalla preferencias.....	46
5.4.9. Menú desplegable.....	47
5.4.10. Pantalla de contacto.....	47
6. Implantación.....	49
6.1. Pruebas.....	49
7. Conclusiones.....	51
7.1. Relación del trabajo con los estudios cursados.....	51
7.2. Relación del trabajo con las competencias transversales.....	52
8. Trabajos futuros.....	53
9. Referencias.....	55
10. Anexos.....	57

Índice de ilustraciones

Ilustración 1. Dispositivos móviles según sistema operativo 2019.	14
Ilustración 2. Dispositivos móviles según sistema operativo 2019 en Obremo. (parte 1)	14
Ilustración 3. Dispositivos móviles según sistema operativo 2019 en Obremo. (parte 2)	15
Ilustración 4. Arquitectura de la aplicación.	23
Ilustración 5. Diagrama de casos de uso actual.....	27
Ilustración 6. Correo requisitos propuestos.	27
Ilustración 7. Diagrama de casos de uso nuevo (web service)	29
Ilustración 8. Diagrama de caso de uso para la aplicación dirigida a operarios.....	29
Ilustración 9. Log-in Arquímedes prototipo atómico.	32
Ilustración 10. Inicio Arquímedes prototipo atómico.....	33
Ilustración 11. Menú Arquímedes prototipo atómico.....	34
Ilustración 12. Parte Arquímedes prototipo atómico.....	35
Ilustración 13. Medio Arquímedes prototipo atómico.	36
Ilustración 14. Operario Arquímedes prototipo atómico.	37
Ilustración 15. Trabajo Arquímedes prototipo atómico.	38
Ilustración 16. Inicio Arquímedes prototipo atómico.	39
Ilustración 17. Arquímedes pantalla inicio de sesión.....	40
Ilustración 18. Arquímedes pantalla recuperar contraseña.	41
Ilustración 19 Arquímedes pantallas de inicio	42
Ilustración 20. Arquímedes pantallas parte de trabajo.....	43
Ilustración 21. Arquímedes pantalla nuevo operario.	44
Ilustración 22. Arquímedes pantalla nuevo medio.	45
Ilustración 23. Arquímedes pantalla nuevo trabajo.....	46
Ilustración 24. Arquímedes pantalla ajustes.....	46
Ilustración 25. Arquímedes menú lateral.....	47
Ilustración 26. Arquímedes pantalla información.	48
Ilustración 27. Resultados pruebas unitarias.....	50

Índice de tablas

Tabla 1. Nuevos requisitos funcionales	28
--	----

1. Introducción.

1.1. Motivación.

La principal motivación que me impulsa a realizar este trabajo es el reto y la superación que representa este trabajo para mí. En este trabajo puedo poner en práctica todos los conocimientos que he adquirido a lo largo de la carrera, así como la posibilidad de trabajar en un proyecto tan importante como lo es la realización de esta aplicación, pues esta se va a realizar en colaboración con distintos departamentos, permitiendo a la aplicación ser un referente en cuanto a diseño de aplicaciones de esta empresa.

1.2. Objetivos.

Este trabajo de fin de grado tiene como objetivo realizar la migración de una aplicación de gestión de partes para los operarios de telecomunicaciones de la empresa Obremo. Obremo [10] es una empresa con una gran cantidad de trabajadores, cercana a los 1200 empleados, y con una capacidad de facturación cercana a los 90 millones de euros anuales. Debido a esto, es necesario el uso de aplicaciones que gestionen el registro de los trabajos que realizan los trabajadores diariamente, y debido a la naturaleza de la empresa y a la heterogeneidad de los puestos de trabajo, resulta asimismo el uso de aplicaciones específicas para mantener un registro de todos los trabajos realizados a lo largo del tiempo.

Los operarios realizan, mayoritariamente, su trabajo en puestos en movilidad. Debido a esto, el nuevo software ha de diseñarse sabiendo que esto puede ser un condicionante a la hora de elegir la tecnología a utilizar o incluso que requisitos no funcionales. Además, debemos garantizar que el software desempeñará su función de manera correcta y presentará la menor cantidad de errores que puedan perjudicar a los operarios e impedirles formalizar sus partes.

Cabe recalcar tres aspectos de gran relevancia que nos guiarán a la hora de plantear los objetivos y el alcance de este proyecto.

En primer lugar, nuestra aplicación cuenta con una versión antecesora en uso actualmente. Por lo tanto, el proyecto precisará realizar tareas de mantenimiento perfectivo de la aplicación actual en caso de ser necesario corregir algún error de esta, ofreciendo soporte a los usuarios de esta paralelamente al desarrollo de la nueva versión. Adicionalmente, podremos obtener información de éstos para mejorar aspectos como el diseño de las interfaces y posibles funcionalidades nuevas que aplicar a la nueva versión. También utilizaremos prácticas de ingeniería inversa para migrar las funcionalidades de la aplicación actual a la nueva, proporcionando a los usuarios un entorno de trabajo similar al que ellos utilizaban con anterioridad.

El segundo lugar, nuestra aplicación cuenta con dos perfiles principales de trabajadores que van a utilizar este sistema.

- Por un lado, tendremos a los operarios que serán los encargados de añadir partes el nuevo sistema y también serán capaces de realizar tareas de edición, borrado y consulta de aquellos partes en los cuales han participado. Estos usuarios son aquellos que principalmente están en movilidad mientras realizan su trabajo. Por lo tanto, estos

aspectos de la aplicación deben estar diseñados específicamente para su uso desde un aparato portátil.

· Por otro lado, tenemos a los encargados cuyo principal objetivo al utilizar esta aplicación es obtener informes relacionados con la información que desean extraer de los partes que los operarios han realizado.

Esto nos plantea un importante desafío, pues los operarios generalmente incluirán sus partes desde un dispositivo móvil con las características que esto conlleva (menor tamaño de pantalla, teclado integrado en el móvil...) mientras que los encargados realizarán el informe normalmente en su puesto de trabajo con un ordenador de sobremesa y esperan obtener la información en un archivo con la extensión propia de los documentos Excel.

1.3. Impacto esperado.

Como resultado del trabajo realizado, la usabilidad de la aplicación mejorará, añadiendo de esta manera un valor extra a la empresa, incrementando la productividad, tanto de los operarios como de los encargados y minimizando los posibles errores que puedan cometer los mismos.

Adicionalmente, mejoraremos la accesibilidad para acceder a datos y la seguridad de la aplicación, proporcionando una serie de beneficios como optimizar el uso de los recursos y mejorar la faceta de análisis con información en tiempo real o de nutrir una base de datos.

1.4. Estructura.

A lo largo de este documento vamos a tratar una serie de aspectos, repartidos a lo largo de todo el trabajo, referentes al trabajo que se ha estado realizando a lo largo del trabajo final de grado.

Entre estos apartados encontramos el apartado de “estado del arte” donde hablaremos de la situación actual del software que utiliza la empresa y realizaremos una propuesta de mejora, aprovechando un estudio realizado que compara la cantidad de sistemas móviles que dispone la empresa agrupados en los distintos sistemas operativos para móvil.

Después, podemos encontrar el apartado de análisis del problema, donde analizamos distintos aspectos a tener en cuenta a la hora de desarrollar cualquier software que presente estas características.

Seguidamente, hablaremos del diseño propuesto de la solución, explicando el funcionamiento del sistema y la comunicación entre los diversos subsistemas que lo conforman, este apartado está continuado y muy ligado a los siguientes apartados de “desarrollo de la solución propuesta” y “implantación” que hablarán del proceso que se seguirá para desplegar el nuevo software en la empresa.

A continuación, tenemos el apartado de pruebas donde vamos a recoger la implementación de las pruebas y el análisis de los resultados obtenidos.

Finalmente, vamos a encontrar tres apartados que conforman una ayuda para seguir la información que ha permitido el desarrollo de este documento, estos apartados son las referencias a documentos externos, anexos o el glosario.

1.5. Convenciones.

A lo largo de este documento vamos a utilizar ciertas convenciones generalmente aceptadas en este tipo de trabajos. Entre ellas determinamos las siguientes:

- El código fuente se mostrará en letra cursiva y el estilo de la fuente será Courier. De esta manera sabremos que si un texto contiene esta tipología se trata de un fragmento del código fuente.
- Las palabras extranjeras, aunque intentemos limitar su uso, se remarcarán en cursiva.
- Se entrecomillarán las citas textuales externas a este documento.
- Las páginas que dan inicio a una nueva sección de nuestro trabajo se encontrarán siempre en un número impar.



Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales



2. Estado del arte

A lo largo de este capítulo vamos a presentar el análisis de la situación actual de la tecnología utilizada en la empresa. En nuestro caso, el trabajo se focalizará principalmente en comparar la aplicación actual desarrollada y usada en la empresa con la nueva desarrollada en este trabajo. Haciendo énfasis en los cambios realizados a lo largo del mantenimiento de la misma y en los cambios diseñados para la nueva aplicación.

2.1. Crítica al estado del arte.

Obremo es una empresa especializada en la presentación de servicios integrales en una gran cantidad de áreas como son proyectos dedicados a la comercialización, instalaciones, proyectos de ingeniería o mantenimiento de diversos sectores.

Principalmente, la actividad de la empresa se centra en la gestión de infraestructuras públicas y privadas a nivel nacional e internacional en los sectores de agua, telecomunicaciones, gas, electricidad, energías renovables e infraestructuras.

El sector de las telecomunicaciones presenta una de las actividades más importantes de la empresa, donde los operarios realizan distintas tareas, desde la instalación y mantenimiento de estaciones base remotas de cable de red a la gestión de tiendas Vodafone.

Estos operarios realizan trabajo de campo día a día. Por lo tanto, la empresa necesita registrar el trabajo realizado por los mismos, un registro que se ha de realizar mediante un dispositivo móvil con las distintas ventajas y desventajas que presenta esta circunstancia.

Actualmente la empresa presenta un servicio web que realiza la función básica de gestionar los partes de trabajo para operarios y encargados, de modo que ambos pueden ver todas las opciones de las que dispone la aplicación, si bien sólo los encargados pueden realizar todas las acciones posibles.

Al tratarse de un servicio web, éste presenta requisitos que pueden dificultar la introducción de nuevos partes por parte de los operarios que tienen que realizar esta tarea con movilidad, siendo el más importante de todos ellos la necesidad de la conectividad a internet constante para realizar dichas tareas.

2.2. Propuesta.

Nuestra propuesta es la realización de un sistema de aplicaciones múltiples que separan las funciones de los operarios de aquellas funciones que necesitan realizar los encargados.

Debido a esto, nuestra propuesta es realizar un servicio web que funcionará como *middleware* de enlace con la nueva aplicación móvil que vamos a desarrollar.

Al margen de esto tendremos que realizar una serie de cambios sobre la aplicación actual para acomodarla a los nuevos requisitos que proponen los clientes.

A la hora de realizar una aplicación en movilidad necesitamos investigar el mercado de los dispositivos móviles. Por ello, a continuación, vamos a investigar el mercado global de los distintos dispositivos móviles y los sistemas operativos que éstos presentan [12].



Ilustración 1. Dispositivos móviles según sistema operativo 2019.

Como se puede observar, la gran mayoría de móviles en 2019 tienen Android como sistema operativo. Sin embargo, consideramos conveniente a profundizar más en la organización para la cual trabajamos. Para ello vamos a observar, dentro de la misma, los distintos dispositivos que tiene.

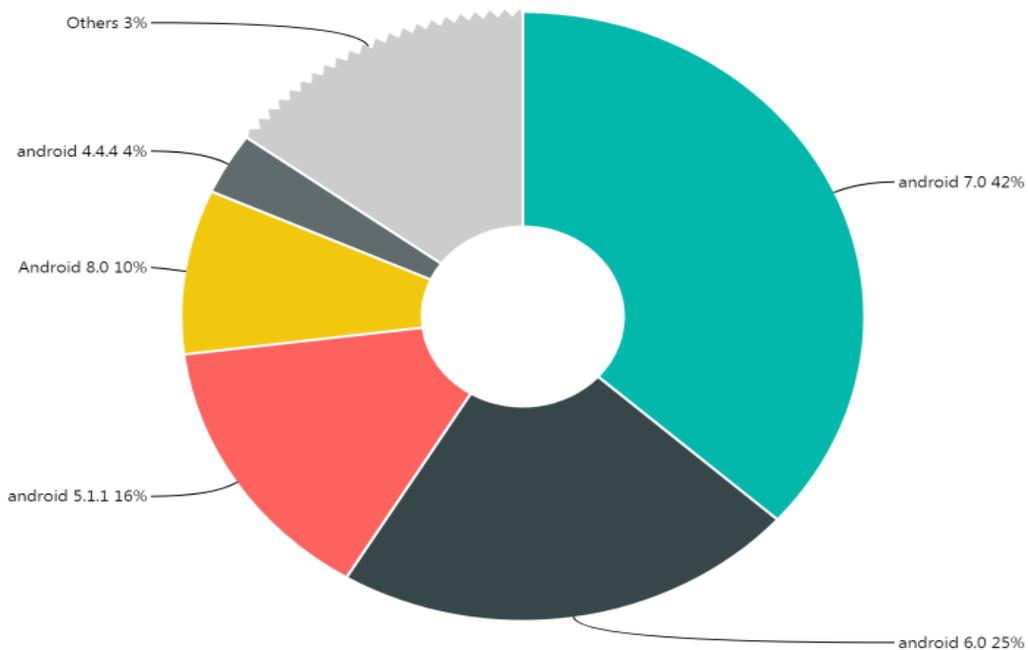


Ilustración 2. Dispositivos móviles según sistema operativo 2019 en Obremo. (parte 1)

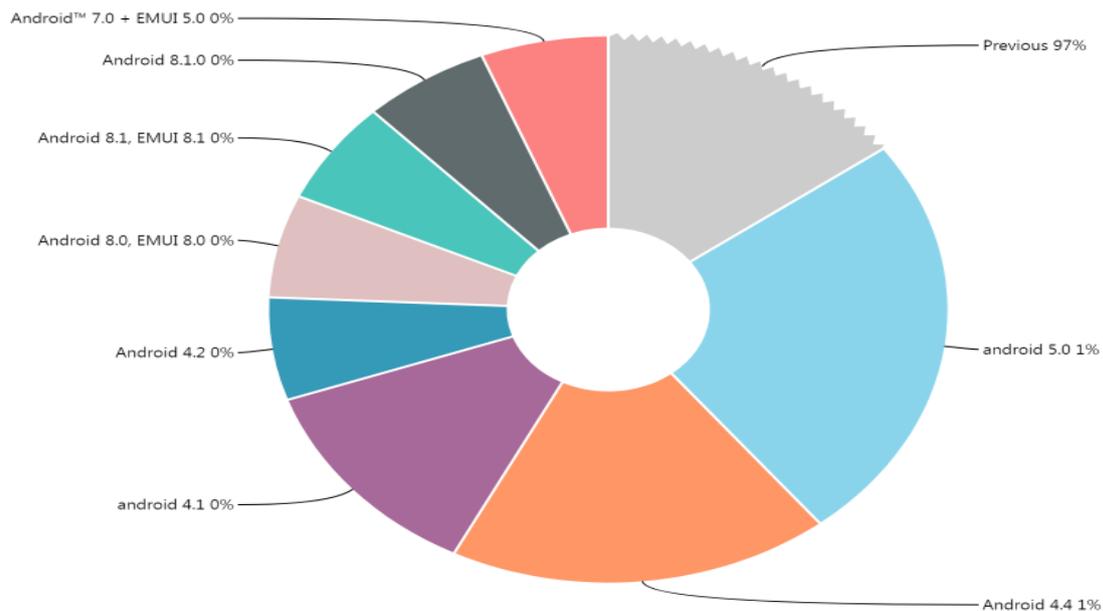


Ilustración 3. Dispositivos móviles según sistema operativo 2019 en Obremo. (parte 2)

Como podemos ver en estos dos gráficos, la gran mayoría de los móviles de la compañía presentan una versión de Android superior o igual a la versión 6.0, destacando la cantidad de dispositivos que presentan las versiones 7.0 (42%), 6.0(25%) y la versión 8.0(10%).

A continuación, discutiremos las características que presentan estas versiones.

Android Marshmallow (6.0):

Esta versión salió oficialmente al mercado el 5 de octubre de 2015. Entre sus principales características se encuentra el nuevo modelo de permisos, que permite únicamente 8 categorías de permisos, y donde las aplicaciones ya no pueden conceder automáticamente los permisos específicos en el mismo momento de la instalación.

Además, esta versión también ofrece soporte a diversas características hardware que mejoraron la usabilidad en los dispositivos con este sistema operativo. Una de ellas fue el soporte nativo al reconocimiento de huellas, permitiendo que las mismas puedan ser utilizadas para desbloquear dispositivos o autenticación en distintas aplicaciones. Otra característica es un nuevo sistema de gestión de recursos energéticos, que reduce las actividades ejecutadas en segundo plano en el dispositivo cuando este no tiene contacto físico con el usuario. Finalmente, podemos destacar la compatibilidad de esta versión con los USB tipo C que permiten una carga más rápida del terminal.

Android Nougat (7.0):

Android Nougat fue anunciada por google el 30 de junio 2016 y de esta versión destacaremos 4 características importantes que fueron añadidas en esta nueva versión:

- Multiventana: esta función permite utilizar dos aplicaciones simultáneamente dividiendo el espacio de la pantalla entre las dos aplicaciones.

- Soporte para realidad virtual: pese a ser utilizada por otros móviles como el Samsung Galaxy S6 o S7, a partir de esta versión la realidad virtual ahora es compatible de manera nativa para todos los dispositivos Android. Junto a esta característica hemos de comentar que Google presentó una nueva tienda de aplicaciones destinada a la realidad virtual llamada *Daydream*, pero que no es compatible con todos los dispositivos de esta versión.

- *Daze on the go*: ésta es una mejora de la función *Doze* de la versión anterior que permite activar esta función cuando el móvil tiene la pantalla apagada.

Android Oreo (8.0):

La versión final de Android 8.0 Oreo se publicó el 21 de agosto de 2017. Esta versión incluye entre sus novedades las siguientes características: Optimización energética, donde el sistema limita los procesos en segundo plano, mejorando el ahorro energético, y mejora en las notificaciones y en el diseño de las aplicaciones propias del sistema operativo. Sin embargo, tenemos que remarcar de esta versión presenta el problema de consumir datos móviles, aunque la red inalámbrica (WI-FI) esté disponible.

3. Análisis del problema.

3.1 Análisis de la seguridad.

El sistema es utilizado únicamente por una serie de usuarios y divididos por categorías entre los mismos, distinguiéndose operarios y encargados. Por lo tanto, a lo largo de nuestro desarrollo vamos a requerir de un sistema que valide primero si el usuario pertenece a la organización y a continuación necesitaremos comprobar que el rol de este usuario es el indicado para realizar las tareas que se propone realizar.

Además, atenderemos en la aplicación, el sistema de recuperación de contraseñas que se va a añadir desarrollado por otro compañero de la organización para automatizar el proceso de recuperación en caso de algún fallo o incidencia.

3.2. Análisis del marco legal y ético.

Como es sabido, el marco legal y ético ha de ser cumplido por cualquier profesional. A lo largo de este apartado vamos a detallar los diversos aspectos de éste que hemos considerado relevantes.

3.2.1. Análisis de la protección de datos.

Tras el análisis de la Ley Orgánica de Protección de Datos o LOPD, como la llamaremos de aquí en adelante para abreviar, hemos concluido una serie de apartados que nos han parecido significativos respecto a los datos que vamos a tratar.

Con el objetivo de reducir la posibilidad de pérdida o filtración de datos, los datos que recoge la aplicación han sido los mínimos para su funcionamiento. Por lo tanto, no se utiliza más que el nombre de los operarios o el código del operario que pertenecen al ámbito impersonal.

Sin embargo, si fuese necesario, hay protocolos preparados para la encriptación nativa que soportan los distintos dispositivos móviles. Por lo tanto, podrían encriptarse si se añadiese información de naturaleza más delicada.

3.2.2. Propiedad intelectual.

Respecto a la propiedad intelectual, la mayoría de los recursos que se han utilizado pertenecen a la empresa y han sido diseñados por el departamento de marketing. Entre ellos tenemos los diseños de los iconos de la aplicación o el logo de la empresa en todas sus variantes. Por otro lado, en cuanto a las librerías y entornos de desarrollo utilizados, siempre que son ajenos a la organización se han cumplido los términos de uso de las licencias que éstos presentaban, especialmente las licencias de Android Studio y STS pues ambos son los IDE que se han utilizado para desarrollar los distintos subsistemas.

3.3. Análisis de riesgos.

Nuestra aplicación no tiene riesgo de no ser utilizada, pues se trata de una aplicación corporativa de uso obligatorio para reportar su trabajo y, por lo tanto, está altamente ligada a su rendimiento económico. Sin embargo, esta cualidad es una espada de doble

filo, pues el descontento o el mal funcionamiento de esta aplicación puede afectar en el rendimiento de un trabajador.

Afortunadamente, nuestro despliegue en fase mientras ambas aplicaciones conviven nos permite tener margen de mejora para adecuar finalmente la aplicación a las cualidades que el usuario puede creer convenientes para su uso y así incrementar la producción o hacer que el sistema no sea un limitante para el mismo.

3.4. Identificación y análisis de soluciones posibles.

Para ofrecer la mejor respuesta al problema hemos necesitado investigar entre todas las alternativas posibles para realizar una aplicación para los móviles disponibles a día de hoy. A lo largo de este apartado vamos a observar las características que los diferencian y cuáles son los principales puntos fuertes de las distintas alternativas, así como identificar aquellas características negativas a la hora de seleccionar una u otra alternativa.

Podemos distinguir 5 grupos de entornos de programación principales, desde el desarrollo de aplicaciones web que se adaptan a las pantallas de los móviles a aplicaciones nativas desarrolladas para un sistema operativo como Android o IOS específicamente.

- **Aplicaciones *web responsive*:**

Estas son las aplicaciones web que utilizan CSS y HTML para adaptarse a las distintas resoluciones de los móviles que utilizan dicha aplicación.

Su principal ventaja es el coste de crear una aplicación de dicha manera, pues es rápido y sencillo.

Las principales desventajas son:

- No es una aplicación que se puede descargar desde una tienda como Google Store o Apple Store.
- Sencillez en las funciones que puede realizar la misma (no puede gestionar notificaciones, descargas, acceder a dichos recursos del móvil, etc.)
- Dependen directamente de la conexión a internet.

- **Aplicaciones PWA:**

Entendemos por aplicaciones PWA, aquellas aplicaciones basadas en el uso de la web pero que mediante distintas tecnologías son capaces de llegar a comportarse como aplicaciones nativas.

Las principales ventajas son:

- Es posible instalarlas como una aplicación más para el móvil.
- Es posible ejecutarlas en un ordenador personal mediante el uso de un navegador como Chrome o Firefox.

Las principales desventajas son:

- Peor rendimiento que las aplicaciones nativas.

- Mejor preparadas para Android que para IOS, donde podemos encontrarnos con posibles problemas para este último.

- **Aplicaciones híbridas:**

Esta opción permite realizar una aplicación mediante el uso de HTML, JAVASCRIPT y CSS de manera que la aplicación encapsula los elementos, permitiendo que éstos pueden generarse como una aplicación más.

Principales ventajas:

- Se puede publicar la aplicación en tiendas como Google Store o Apple Store.
- Permiten reutilizar el código para varias plataformas.

Principales desventajas:

- Aplicaciones con un rendimiento inferior al esperado.
- Actualmente en desuso.
- Abandonada por Facebook y LinkedIn.

- **Aplicaciones Bridge:**

Son aplicaciones que permiten exportar para las distintas plataformas sin perder rendimiento o fluidez, minimizando la incomodidad del usuario. Esta tecnología obtiene este nombre debido a que funciona a modo de puente entre el código nativo y el código propio del *framework* que utilizamos, C# con *Xamarin* o JavaScript con *React Native*, entre otros. De este modo, dividen el código del *frontEnd* con la parte lógica, integrándose con código nativo.

Ventajas:

- Permiten reutilizar código para cada uno de los distintos sistemas operativos.
- Rendimiento similar a las aplicaciones nativas.
- Funcionalidades cercanas a las aplicaciones nativas.

Desventajas:

- Requieren código nativo para realizar algunas funciones.

- **Aplicaciones nativas:**

Estas aplicaciones son aquellas que están desarrolladas específicamente para una plataforma en concreto. Son aplicaciones capaces de adaptarse a todas las funcionalidades del dispositivo en el cual se ejecutan y a todas las características de éste, obteniendo de esta forma una mejor experiencia para el usuario.

Ventajas:



Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales

- Permiten acceder a todas las características del dispositivo y optimizarlas para obtener el mejor rendimiento posible en una aplicación.
- Facilidad de incluir las mismas en las tiendas de cada plataforma, PlayStore en Android o AppStore en IOS.
- Amplio soporte por las compañías, que se encargan de mantener los sistemas operativos.
- Herramientas desarrolladas para facilitar la programación de nuevas aplicaciones (Android Studio, por ejemplo).

Desventajas:

- Requieren programadores especializados en el lenguaje de programación para cada sistema.
- En caso de hacer la aplicación para varios sistemas hay que trabajar en una aplicación distinta por cada sistema operativo que queramos desarrollar, necesiéndose generar diferencias entre la versión de un sistema u otro.
- En algunos casos, algunos Sistemas Operativos requieren características especiales para compilar sus aplicaciones (necesitas un MAC para compilar para IOS).
- Requieren de dos o más equipos para mantener las distintas aplicaciones en caso de que se realicen para dos o más sistemas.

En resumen, nuestra decisión dependerá de las características y los objetivos que esperamos que cumpla nuestra aplicación y de los recursos que tengamos para realizarla. Para una aplicación de lectura de datos muy sencilla, una aplicación WEB *Responsive* puede ser suficiente. Por otro lado, si necesitamos obtener el mayor rendimiento que el móvil puede ofrecernos, deberíamos de optar por la opción del desarrollo nativo, pues seremos capaces de obtener un mayor rendimiento para nuestra aplicación.

Cabe recalcar que existe un amplio abanico de posibilidades y que, en la mayoría de los casos, las aplicaciones *Bridge* pueden ser las más indicadas para realizar una aplicación lo más multiplataforma posible.

3.5. Solución propuesta.

Tras el estudio de las posibilidades realizado previamente, y tras una reunión de equipo de los trabajadores del departamento de informática, se ha concluido realizar 3 subsistemas que funcionarán conjuntamente para proporcionar a los distintos usuarios la realización de las tareas requeridas.

Estos subsistemas son los siguientes:

Para comenzar, el *web service* encargado de gestionar la conexión de las dos distintas aplicaciones con la base de datos y con el sistema de *active directory*, el cual es el encargado de la gestión de usuarios en la empresa.

En segundo lugar, con la aplicación nativa de Android que será la encargada de permitir a los operarios rellenar los partes de trabajo con la información necesaria de forma local y que se podrá mandar a la base de datos en el momento en el que el operario vea oportuno si en el momento de la creación éste no puede hacerlo.

Finalmente, la aplicación web que pueden usar los encargados para gestionar a los empleados y los centros de coste, o para generar los informes que necesiten según la información que obtienen de los partes que los operarios ingresan. Este subsistema será una ampliación del sistema que se encuentra en funcionamiento actualmente.

3.6. Plan de trabajo.

Dividiremos este trabajo en dos bloques principales:

El primero, estará dedicado a la aplicación web que está actualmente en producción pero que necesita unas mejoras para adecuarse a las peticiones de los clientes.

Esta parte nos ocupará dos meses de producción en los cuales dividiremos una porción del tiempo a aprender los entresijos de la estructura y el código que presenta la aplicación y diseñar las mejoras demandadas. Una vez terminada esta parte podremos realizar los cambios que se han propuesto.

El segundo bloque estará dedicado al desarrollo de la aplicación de Android, cual nos ocupará tres meses de duración. En los primeros quince días nos centraremos en el diseño de la aplicación y en valorar con el equipo de desarrollo de Obremo las distintas alternativas a la hora de realizar una aplicación móvil actual.

3.7. Presupuesto.

Debido a que la empresa ya tenía los recursos necesarios tanto para el mantenimiento de la base de datos, que comparte con la aplicación anteriormente usada, como los distintos servidores utilizados para los distintos subsistemas que lo requieran, el coste de esta aplicación ha sido únicamente el costo del salario del trabajador que la ha desarrollado.



Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales



4. Diseño de la solución.

4.1. Arquitectura del sistema.

En este apartado vamos a explicar la arquitectura que presentará el sistema propuesto. Como en anteriores puntos hemos comentado, vamos a estructurar nuestra aplicación en 5 subsistemas, de los cuales 3 van a ser aplicaciones desarrolladas por la empresa. La siguiente figura (Figura 4) ilustra la estructura del sistema.

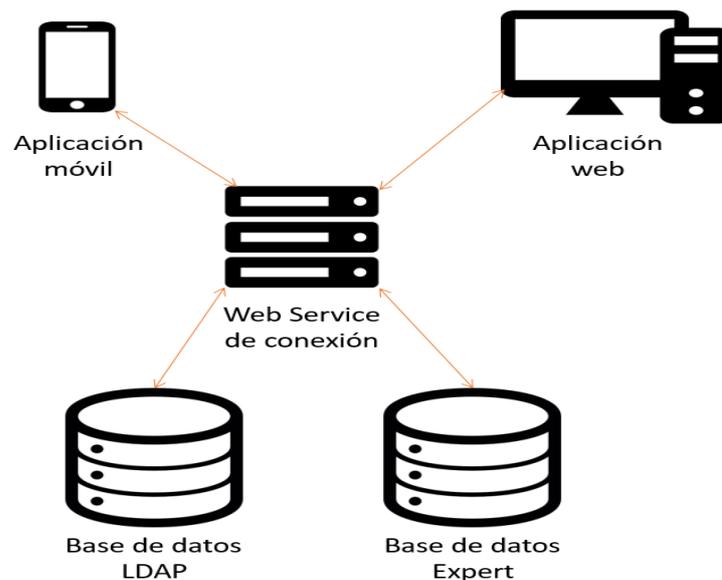


Ilustración 4. Arquitectura de la aplicación.

4.2. Diseño detallado.

A continuación, vamos a explicar los diferentes subsistemas diseñados.

En primer lugar, vamos a presentar de los dos sistemas gestores de bases de datos en los cuales almacenamos los datos que necesitaremos a lo largo de toda la vida útil de este programa.

A continuación, vamos a describir el *web Service* que funcionará como *middleware*. Este término se aplica a la capa de software que proporciona abstracción a nivel de software y actúa como intermediario entre distintos sistemas. En nuestro caso servirá de intermediario entre las aplicaciones, ya sea la aplicación móvil o web con las dos bases de datos.

Por último, tenemos el gran bloque de las aplicaciones. Por un lado, desarrollamos la aplicación móvil nativa para *Android* con *Android Studio*. Por otro lado, comentaremos el *web Service* que vamos a actualizar para ser utilizado por los usuarios con las nuevas funcionalidades que éstos nos han demandado.

Además de los subsistemas, vamos a necesitar un protocolo de intercambio de datos. En este caso vamos a utilizar el protocolo SOAP.

4.3. Tecnologías utilizadas.

A lo largo de este apartado vamos a detallar las distintas tecnologías que vamos a utilizar a lo largo de este trabajo, poniendo especial atención a los aspectos que nos son de utilidad.



Comenzamos este punto hablando de Android [6] Studio. Este entorno de desarrollo integrado o IDE, destaca por su potente editor de código, pero presenta otras funciones adicionales, que ofrece un ayuda apreciable para cualquier desarrollador que quiera programar para Android nativamente.

De estas funciones destacaremos la función *Instant Run*, que permite mostrar cambios en tiempo de ejecución, evitando así la necesidad de compilar un nuevo *Android Package* (APK). Compatibilidad con otros lenguajes a parte *Java* [2] o *Kotlin*, siendo estos dos los utilizados para el desarrollo nativo, estos lenguajes adicionales son C++ o *NDK*.

Además, una función importante es la integración que presenta con *Git-Hub*, que permite importar códigos de ejemplo que pueden ser útiles a la hora de ahorrar tiempo en el desarrollo.

Finalmente, una de las funciones más importantes es el emulador que permite analizar el comportamiento de la aplicación sin un dispositivo, permitiendo replicar el comportamiento de varios modelos del mercado, aunque también se puede utilizar un móvil conectado mediante USB o mediante conexión inalámbrica para utilizar la aplicación en cualquier móvil, siempre y cuando el dispositivo presente las características necesarias para hacer funcionar el programa deseado.



Podemos continuar con *Spring Tool Suite* o STS, como lo llamaremos a partir de este punto. Al igual que *Android Studio*, STS [7] es un IDE basado en *Eclipse* otro famoso IDE en el mercado actual. STS proporciona al usuario tras su instalación un entorno listo para la implementación, despliegue y el resto de funcionalidades necesarias para desarrollo de software.

STS presenta ciertas características especialmente útiles a la hora de desarrollar software, permitiendo desarrollos más rápidos y efectivos, convirtiendo a STS en la alternativa de la empresa cuando ésta realiza principalmente servicios web basados en Java.

Ahora vamos a comentar las características de STS que nos van a resultar de mayor ayuda a lo largo del desarrollo de este trabajo. Destacaremos el útil asistente de código, el cual nos ahorra mucho tiempo durante el desarrollo o la función de soporte a la refactorización, que nos va ayudar a lo largo de todo el desarrollo.



El lenguaje que vamos a utilizar para implementar el código ya sea para el *web Service* o la aplicación Android va a ser Java [8]. Java fue desarrollado por *Sun Microsystems* y presentado en la segunda mitad del año 1995. Una de sus principales ventajas es la posibilidad de ejecutar sus programas en diversas plataformas como Windows, Mac OS o Linux, sistema operativo en el cual está basado actualmente Linux.

Los programas se compilan en un lenguaje intermedio, denominado *Bytecode*. Este código es interpretado por la máquina virtual de java del entorno de ejecución, consiguiendo con esto la portabilidad en las distintas plataformas que hemos comentado.

Como se ha comentado, una de las principales características de Java es la portabilidad, siendo ésta la característica que permite a los programas basados en este lenguaje ser ejecutado en cualquier Sistema Operativo y en cualquier dispositivo físico.



El *Simple Object Access Protocol*, también conocido como SOAP [5], es un protocolo estandarizado para el intercambio de información entre aplicaciones. La especificación escrita en SOAP define una envoltura basada en XML para la información que se transmite y el conjunto de reglas para traducir los tipos de datos específicos en representaciones XML.

Este diseño permite a SOAP ser adecuado para un gran espectro de posibilidades a la hora de enviar mensajes entre aplicaciones y patrones de integración, haciendo que la popularidad el protocolo crezca.

Para las pruebas de este protocolo utilizaremos SOAP UI, programa *open-source* que permite realizar pruebas a *web services* que utilizan protocolos SOAP. Adicionalmente también permite probar el protocolo REST y soporta la integración con diversos IDE como Eclipse o NetBeans.



Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales



5. Desarrollo de la solución propuesta.

Como ya se ha comentado en los apartados anteriores este trabajo tiene dos partes claramente diferenciadas que comentaremos separadamente.

Comenzaremos con el apartado de mantenimiento y mejora del software actual. Para ello la Figura 5 representa el diagrama de casos de uso que presenta la aplicación actualmente.

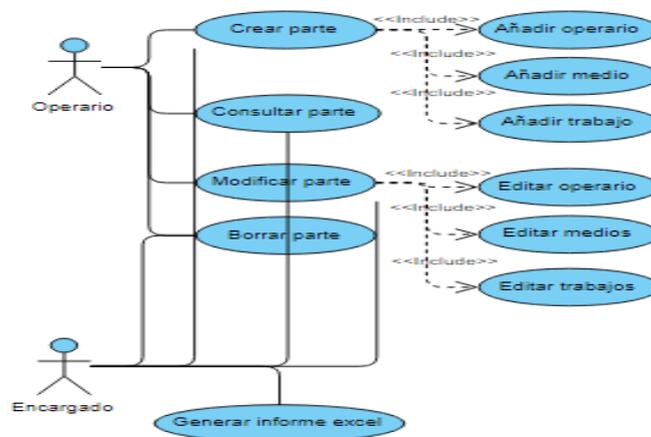


Ilustración 5. Diagrama de casos de uso actual.

En él podemos observar las principales funciones que cumple nuestra aplicación y las funcionalidades que pueden realizar cada uno de los distintos usuarios posibles.

Ahora vamos a analizar los requerimientos de la empresa para las mejoras que se han planteado para la aplicación y los nuevos casos de uso que nos plantea o la modificación de alguno de ellos.

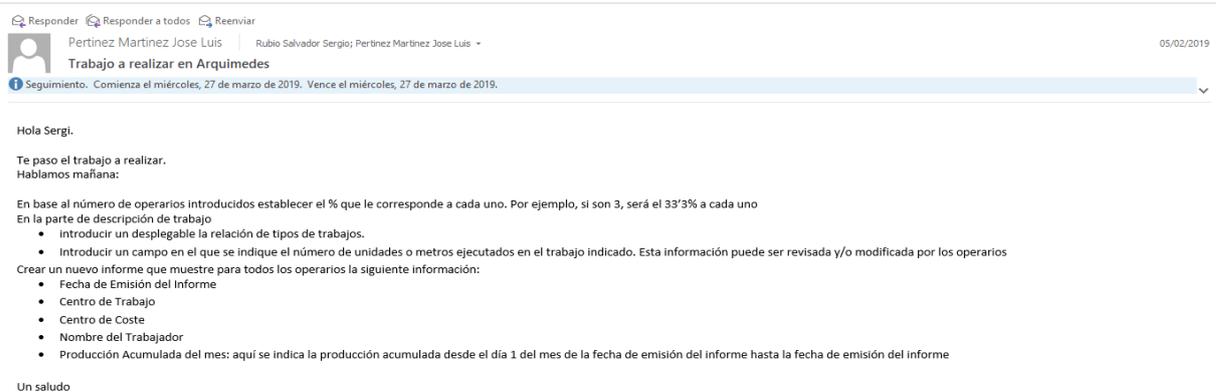


Ilustración 6. Correo requisitos propuestos.

De este correo podemos extraer los siguientes requisitos funcionales, que organizaremos en la siguiente tabla resumen:

Requisito	Establecer porcentaje de participación del operario
Definición	El sistema debe calcular el porcentaje de participación del operario, pero no debe mostrarlo en la aplicación, pues esta información solo se utilizará internamente.
Requisito	Introducir un desplegable con la relación de tipos de trabajo.
Definición	El sistema presentará un desplegable con los distintos tipos de trabajo que están clasificados en la organización.
Requisito	Introducir un campo en el que se indique el número de unidades del trabajo indicado
Definición	El sistema ha de indicar claramente cuál es la unidad de medida del trabajo seleccionado por el operario.
Requisito	Crear nuevo informe
Definición	La aplicación debe generar un nuevo informe con los campos: fecha de emisión, centro de trabajo, centro de coste, nombre del trabajador y producción acumulada. Este informe sólo estará disponible para los encargados, que lo descargarán en un documento con la extensión propia de los documentos <i>Excel</i> .

Tabla 1. Nuevos requisitos funcionales.

Tras varias reuniones y con los anteriores requisitos como base hemos obtenido los siguientes cambios que hemos de realizar.

5.1. Añadir un campo en el formulario de introducción de trabajos.

De acuerdo con lo requerido, el campo que vamos a insertar es un desplegable en el cual se encuentran los tipos de trabajo que previamente los clientes nos han especificado. Estos se han almacenado junto a sus medidas y el precio por medida en la tabla nueva creada con este propósito en la base de datos, llamada ARQUI_TIPOLOGIA_PARTES_TELECO. Esta tabla se consulta para obtener el tipo de trabajo y el precio en las nuevas funcionalidades que utilizaremos.

5.2. Realizar un nuevo informe de producción.

En este requisito podemos incluir el cálculo del porcentaje de participación en el parte, debido a que el único uso de este requisito será esta sección al generar el nuevo informe que nos han pedido.

Tras varias modificaciones del mismo, se ha llegado a la conclusión de realizar este nuevo requisito funcional de la siguiente forma.

El informe requiere de dos fechas en las que los generarán los informes de producción. Con estas dos fechas vamos a generar un documento Excel.

Por lo tanto, podemos concluir que el subsistema web ha de obedecer los requisitos funcionales que se recogen en el siguiente diagrama de casos de uso.

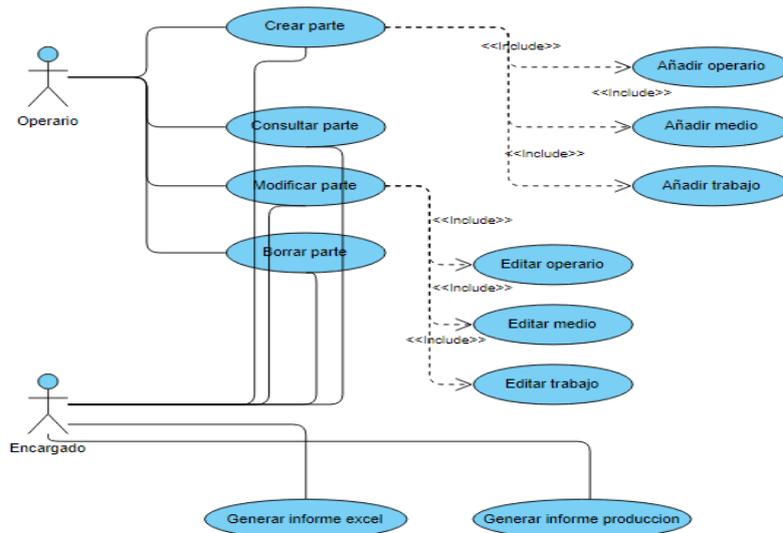


Ilustración 7. Diagrama de casos de uso nuevo (web service)

Sin embargo, optamos por realizar un subsistema que va a ser exclusivo para los operarios. De este modo para el nuevo subsistema tendremos el diagrama de casos de uso que se recoge en la Figura 6. Los casos de uso que podemos observar en él. Como se puede ver, algunos casos de uso son comunes entre el nuevo subsistema web y el subsistema de la aplicación Android.

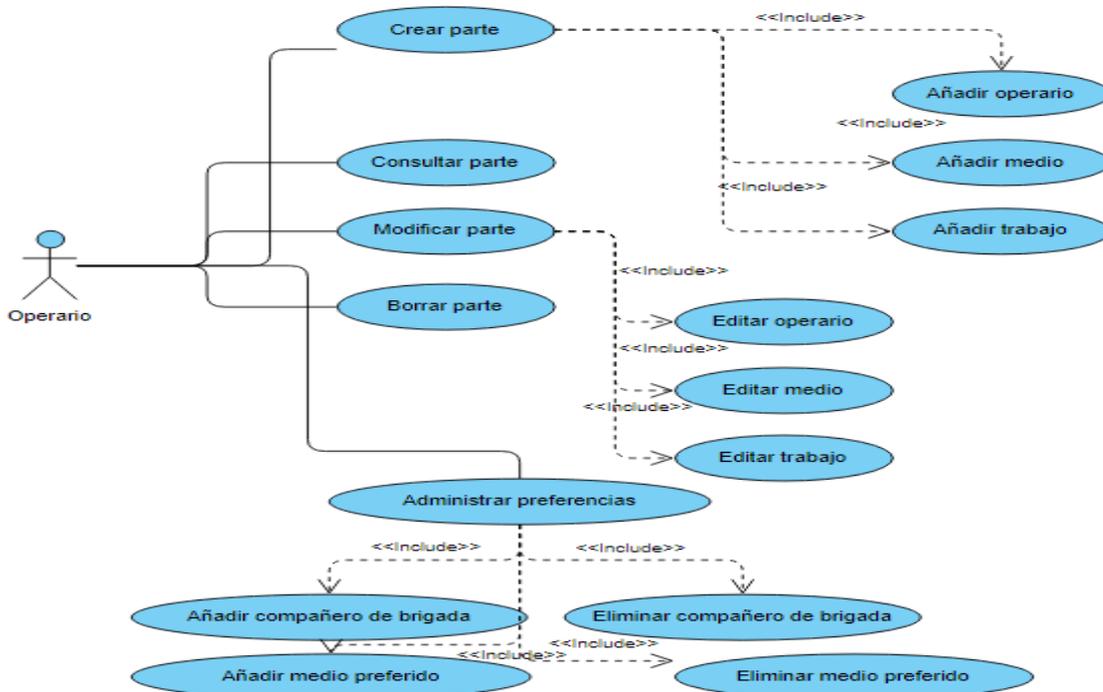


Ilustración 8. Diagrama de caso de uso para la aplicación dirigida a operarios.

Ahora que tenemos el diagrama de casos de uso, procedemos a explicar uno por uno cada uno de ellos en este apartado.

Comencemos con el caso de uso de **crear parte**, que es el principal caso de uso. Decimos que es el caso de uso principal pues es el caso de uso más recurrente y el cual planteó inicialmente la necesidad de crear esta aplicación. La finalidad de este caso de uso es introducir los partes de operario y la información que éstos contienen en la base de datos.

Crear parte incluye a su vez tres casos de uso, entre los que encontramos el **añadir un operario**; en este caso de uso tendremos que rellenar un formulario. En este formulario introduciremos el nombre del operario, el código, la hora de comienzo de jornada, la hora de finalización de la misma y las horas durante las que ha trabajado.

También incluye **añadir medio**, en el cual introduciremos la matrícula y obtendremos el modelo del vehículo que se ha utilizado en este trabajo.

Finalmente, tenemos la funcionalidad de **añadir trabajo**, donde introduciremos el tipo del trabajo realizado, una descripción del mismo y la cantidad del mismo. Donde se puede introducir cero si el trabajo no se va a cobrar en la nómina, pero nunca podrá ser negativo.

Consultar parte es el caso de uso que nos pide poder consultar la información de un parte, Esto se soluciona primero con una pequeña visualización de unos datos seleccionados y acordados con el cliente. Que son el centro de coste, la fecha del parte, el código del parte y los operarios que pertenecen a ese parte. Sin embargo, se podrá consultar con más detalle el parte posteriormente si así se requiere.

Modificar parte es el caso de uso que nos permite editar o cambiar ciertos elementos de un parte y guardarlos en la base de datos. Al igual que en el caso de la creación del parte, aquí podemos distinguir tres casos de uso más.

En primer lugar, identificamos caso de uso de **editar operario** que nos permite editar cualquier elemento necesario y de esta forma cambiar un operario o cambiar las horas que ha trabajado, así como la hora de entrada del mismo o la hora de salida.

Editar medio nos permite modificar un medio por otro sin la necesidad de borrar y crear uno nuevo.

Editar trabajo nos permite modificar un aspecto concreto de un trabajo, ya sea la descripción, el tipo o la cantidad sin tener que modificar el resto, pero nos ofrece también la posibilidad de cambiar cualquier valor del trabajo simultáneamente.

Borrar parte nos permite, tras seleccionar un parte, poder borrarlo si así lo deseamos de la base de datos en su totalidad.

Administrar preferencias es el caso de uso que nos permite organizar nuestra brigada y el medio recurrente, así como el centro de coste habitual. De este modo el usuario puede incrementar su eficacia a la hora de rellenar un parte. Como se puede observar en este caso de uso encontramos cuatro casos de uso ligados a él.

Uno de ellos es **añadir compañero de brigada**, el cual es posible seleccionar un operario ya sea por su código o su nombre y guardarlo en la base de datos del móvil.

Paralelamente, podemos borrar el operario de esta base de datos, mediante el caso de uso **eliminar compañero de brigada**.

También tenemos **añadir medio preferido**, que es semejante al de introducir un nuevo medio al parte, pero lo añadiremos a la base de datos interna del móvil permitiendo para poder recuperarlo cuando lo necesite el usuario. Al igual que con los operarios, también se podrá eliminar uno de estos.

5.3. Diseño gráfico (Prototipado)

El diseño gráfico que vamos a implementar tiene su representación en la aplicación específica para Android. En este apartado vamos a explicar el diseño atómico que se planteó para el prototipo de la aplicación y más tarde veremos el diseño final de la aplicación.

Atomic Desing [1] es una metodología de diseño basada en desglosar una aplicación desde los elementos más básicos que presentan información por sí solos a las instancias específicas de plantillas. Para todo este trabajo nos apoyaremos en el libro propio de la metodología que podemos encontrar referenciado en la bibliografía. [1]

Tal y como hemos comentado comenzaremos con los elementos más básicos que presentan información independiente del resto de elementos, pero que, sin embargo, no pueden representar una funcionalidad por ellos mismo. Este es el caso de una imagen, un texto o un botón.

A continuación, entendemos por molécula la agrupación de átomos que comienzan a tener una funcionalidad más clara.

Juntando moléculas podemos formar organismos, o grupos capaces de formar estructuras y secciones complejas en una interfaz. Cabe recalcar que éstos pueden estar formados por grupos de moléculas iguales o diferentes.

Las plantillas representan, mediante la agrupación de distintos organismos unidos, las primeras aproximaciones a las páginas. Éstas son muy concretas y proporcionan contexto a todas las moléculas y organismos anteriormente abstractos.

Finalmente nos encontramos las páginas, que son instancias específicas de plantillas, las cuales son la representación de más alto nivel dentro de la metodología. Este estado es esencial si se pretende comprobar la efectividad del diseño del sistema. Poder ofrecer una vista en contexto nos permite realizar cambios sobre el resto de elementos para mejorar su integración con el resto de elementos.

A continuación, vamos a presentar unas imágenes de los distintos apartados comentados anteriormente aplicados en nuestro prototipo.

En las siguientes imágenes utilizaremos un código de colores para separar cada elemento que encontremos según la clasificación que hemos descrito previamente. El color morado lo utilizaremos para marcar los átomos, el naranja para las moléculas, los organismos con el color negro y el amarillo marcará los límites de la plantilla. En todas las ilustraciones el diseño final que veremos es el de la página.

5.3.1. Log-in.

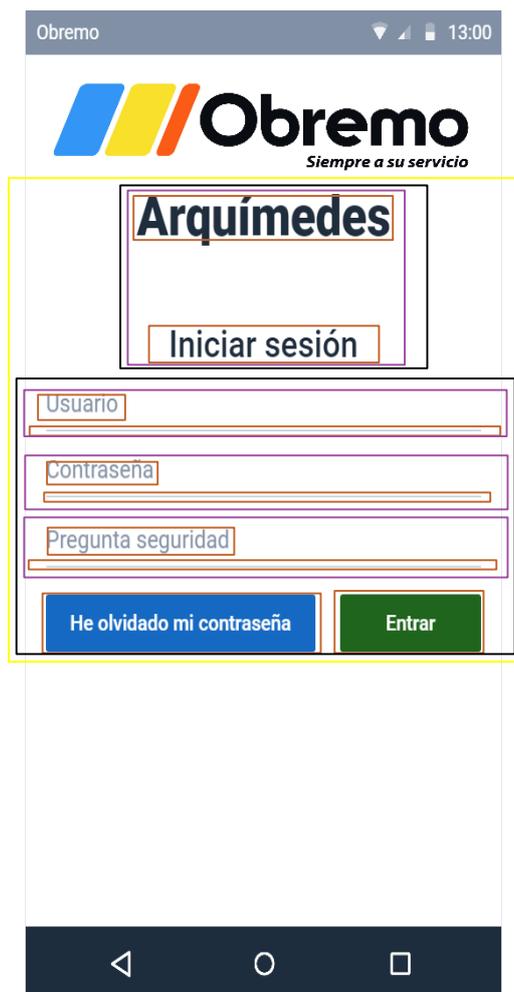


Ilustración 9. Log-in Arquímedes prototipo atómico.

Como hemos comentado con anterioridad los distintos recuadros marcan la clasificación actual de los elementos que conforman esta pantalla. Como podemos ver, esta página está formada por una plantilla que incluye dos organismos diferenciados, el organismo del título formado por la molécula que componen los átomos que son los campos de texto “Arquímedes” e “Iniciar sesión”. Por el otro lado tenemos el organismo formado por elementos más funcionales de la aplicación, ellos son los átomos de los botones “Entrar” y “He olvidado mi contraseña” junto a las moléculas de los campos de inserción de datos, formadas las tres mediante la unión de un campo de texto, como por ejemplo “usuario” y el elemento que contiene la futura entrada de texto (que vendría marcado por la línea inferior).

5.3.2. Inicio.

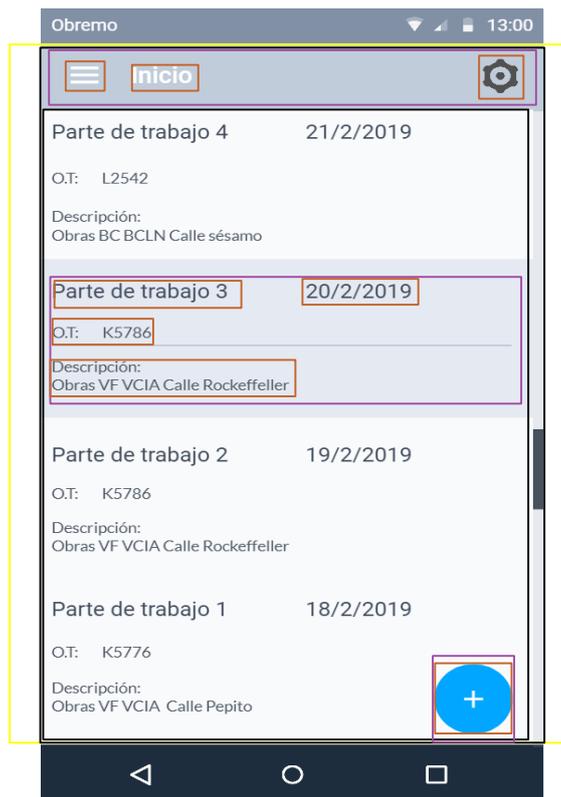


Ilustración 10. Inicio Arquímedes prototipo atómico.

En esta pantalla podemos observar que la página esta vez está formada por la plantilla que contiene dos organismos.

El primero es el organismo del menú superior, formado por una única molécula compuesta del icono de tres líneas que representa la acción desplegar menú, el campo de texto con el nombre de la pantalla y el botón de la rueda que representa la función de acceso al menú. Esta estructura se repite a en la mayoría de las pantallas siguientes por lo tanto no lo volveremos a comentar.

El segundo es el organismo que representa la información que queremos consultar y los elementos funcionales que nos permiten realizar diversas acciones sobre él. Como podemos ver está compuesto por una cantidad indefinida de moléculas que representan la información del parte, formada por los átomos del campo de texto “Parte de trabajo 3”, el campo de texto que presenta la fecha como el ejemplo “20/2/2019”, el campo que contiene el código del centro de trabajo como “O.T: K5786” y el campo descripción del mismo. La otra molécula que lo conforma sería el botón encargado de realizar la acción de crear un nuevo parte, que a su vez es su propio átomo.

5.3.3. Menú.

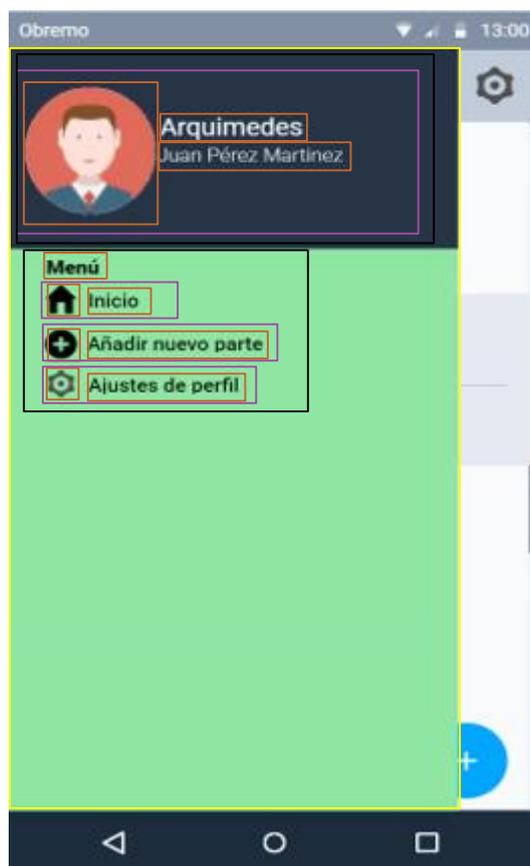


Ilustración 11. Menú Arquímedes prototipo atómico.

Al igual que en la pantalla anterior, de la plantilla podemos diferenciar dos organismos.

El primero nos presenta la información del usuario, siendo un organismo monomolecular que está formado por los átomos de la imagen y los campos de texto “Arquímedes” y “Juan Pérez Martínez”.

Éstos representan a su vez el nombre de la aplicación a la que está conectado y el nombre del operario que está conectado.

El segundo, a su vez, presenta las moléculas encargadas de proporcionar la información necesaria de las acciones que el usuario puede realizar, que están formadas por dos átomos, un icono y un campo de texto. Adicionalmente este organismo tiene el campo de texto “Menú”, que representa el título de esta función.

5.3.4. Parte.

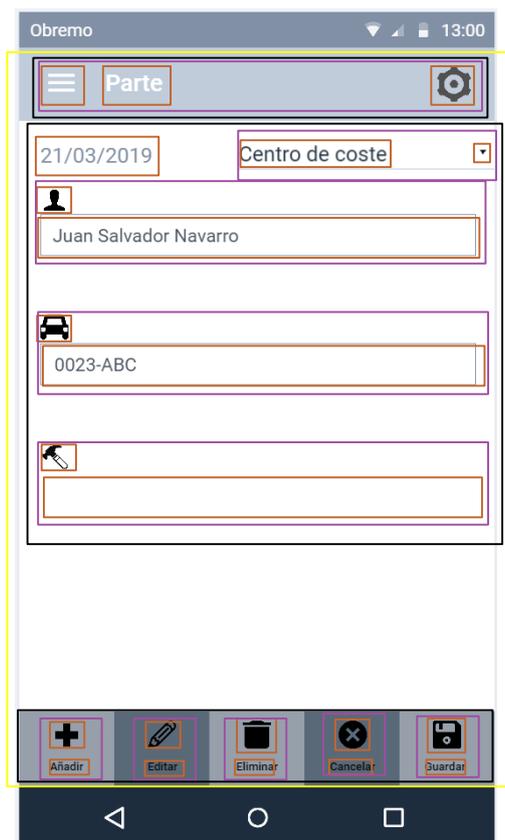


Ilustración 12. Parte Arquímedes prototipo atómico.

A diferencia de la pantalla de inicio, aquí la plantilla presenta tres organismos. El primero de ellos no lo vamos a comentar pues es igual que en el de la misma.

En cambio, el segundo está formado por un átomo, que indica la fecha del parte, y cuatro moléculas, tres de ellas iguales y que representan, mediante un icono, el tipo de elemento del parte, una lista con los elementos que contiene el mismo y una que permite indicar el centro de coste. Esta última está formada por un texto representativo de esta función y un desplegable con los centros de coste cargados a los que el operario tiene acceso.

El tercer organismo está formado por una serie de moléculas que representan distintas funciones que pueden realizar con el parte o sus elementos. Cada una de ellas está formada por dos átomos uno con un icono representativo de la acción y el otro con un campo de texto que le acompaña.

5.3.5. Nuevo medio.

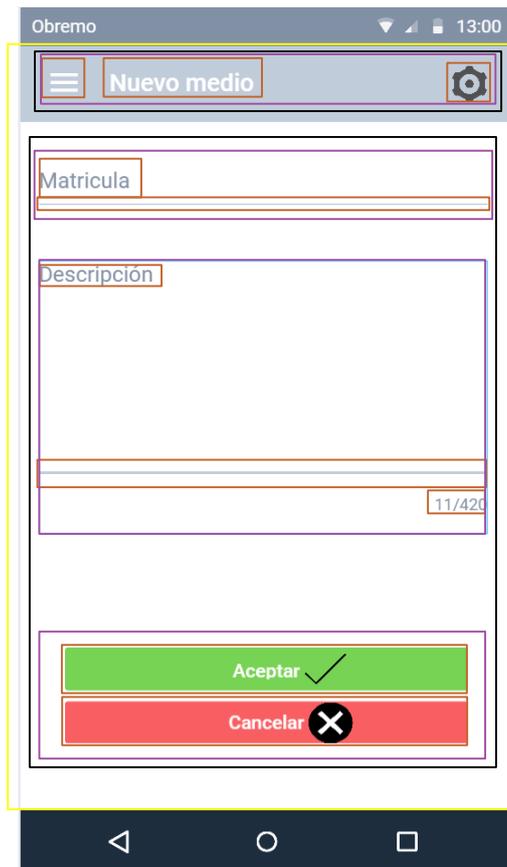


Ilustración 13. Medio Arquímedes prototipo atómico.

Esta página es similar a la pantalla de inicio, aquí la plantilla presenta dos organismos, el primero no lo vamos a comentar pues es igual que en la página anteriormente comentada.

El segundo organismo está compuesto por los elementos que contienen información sobre el nuevo medio o elementos de control de la misma, por ello encontramos dos moléculas que contienen información. La primera contiene información sobre la matricula, formada por un campo de inserción y un texto representativo, mientras que el otro adicionalmente de esto presenta un campo que informa sobre los caracteres máximos permitidos y los que hemos consumido. La molécula de control contiene dos átomos, el botón aceptar y el botón cancelar.

5.3.6. Nuevo operario.

The screenshot shows a mobile application interface for creating a new worker. The title bar at the top reads 'Obremo' and 'Nuevo operario'. The main content area contains several input fields: 'Nombre' (text), 'Codigo' (text), 'Hora entrada' (time, 00:00), 'Hora salida' (time, 00:00), and 'Horas trabajadas' (number, 0.00). At the bottom, there are two buttons: 'Aceptar' (green) and 'Cancelar' (red). The interface is framed by a yellow border.

Ilustración 14. Operario Arquímedes prototipo atómico.

Al igual que en la pantalla anterior, aquí la plantilla presenta dos organismos, el primero no lo vamos a comentar pues es igual que en la página anteriormente comentada.

El segundo contiene moléculas formadas por la dupla de texto representativo e información a introducir, como son los campos nombre, código, hora entrada y hora salida, luego el campo horas trabajadas tiene la característica de requerir tres átomos, el primero para el texto representativo, el segundo para la información a introducir y el tercero un delimitador del campo. Al igual que en la plantilla de la página anterior tenemos la molécula que compone los mismos elementos de control.

5.3.7. Nuevo trabajo.



Ilustración 15. Trabajo Arquímedes prototipo atómico.

Al igual que en las dos páginas anteriores, aquí la plantilla presenta dos organismos, el primero no lo vamos a comentar pues es igual que en la página anteriormente comentada.

El segundo presenta cuatro moléculas similares a algunas que ya hemos comentado anteriormente, la molécula descripción es similar a la que podemos encontrar en la página “nuevo medio”, la molécula cantidad es igual a la molécula nombre de la plantilla “nuevo operario” y las moléculas de “tipo de trabajo” o la formada por los botones de control de operaciones las podemos encontrar explicadas en la plantilla “parte” o en las otras dos anteriores respectivamente.

5.3.8. Ajustes.

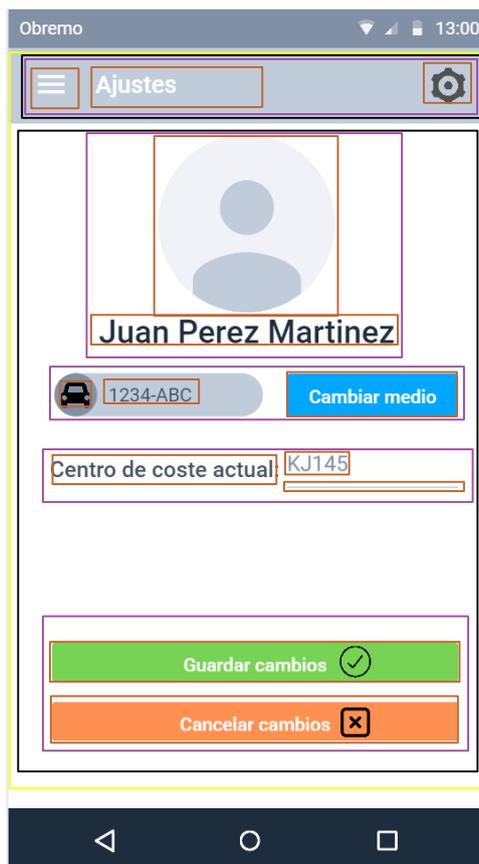


Ilustración 16. Inicio Arquímedes prototipo atómico.

Siguiendo el patrón las dos páginas anteriores, aquí la plantilla presenta dos organismos. Al igual que en los apartados anteriores, el primero no lo vamos a comentar pues es igual que en la página anteriormente comentada.

El segundo presenta moléculas similares a algunas que hemos comentado con anterioridad como son la molécula de elementos de control o el campo de “centro de coste actual”. Sin embargo, encontramos la molécula formada por el icono representativo, el campo de texto de la matrícula y el botón encargado de realizar el cambio. La última molécula que veremos es la molécula que conforman los átomos de la imagen y el campo de texto con el nombre del operario.

5.4. Implementación de la solución.

En el apartado anterior hemos perfilado el prototipo de la solución. Sin embargo, ahora vamos a presentar el diseño final de la misma después de varios meses de desarrollo y reuniones con clientes, mejorando ciertos aspectos que nos han ido solicitando. En este apartado también explicaremos las funcionalidades que representa cada pantalla que nuestro sistema presenta.

Un comentario importante a realizar es la modificación en ciertos colores para adecuar la interfaz gráfica a la marca de la empresa. La elección del color no es al azar, tratamos de crear uniformidad entre las apps de empresa, para que en todas las apps tengan coherencia entre sí. Esto se consigue con repetición de elementos o, en este caso, en el color ya que tratamos de repetir en todos los 3 colores del logotipo. En este caso hemos escogido el color hexadecimal “#2480bb”, que es el color azul del logotipo de Obremo [11].

5.4.1. Inicio de sesión.

Ahora vamos a analizar la pantalla de inicio de sesión. Esta pantalla permite al usuario acceder a la aplicación y es la primera pantalla que el mismo puede observar nada más ingresar en la misma.



Ilustración 17. Arquímedes pantalla inicio de sesión.

Como se puede ver, se han modificado varios elementos respecto al prototipo inicial. Con esto conseguimos que los elementos del centro queden mejor representados y dotamos al botón de “olvidé mi contraseña” de un plano secundario, pero de acceso cómodo.

Esta vista no tiene mayor complejidad que realizar la tarea de iniciar sesión del usuario. Tras comprobar que el usuario es correcto, la aplicación actualiza el contenido de la base de datos interna del programa. En caso de no poder realizar esta acción, el usuario puede intentar recordar la contraseña pulsando en el botón “olvidé mi contraseña”.

5.4.2. Recuperar contraseña.

En esta pantalla se encuentra la funcionalidad que permite al usuario acceder al portal de recuperación de contraseña del que la empresa dispone. Por lo tanto, esta pantalla sólo gestiona la vista *web* de dicho portal, tal y como podemos ver en la siguiente imagen.



Ilustración 18. Arquímedes pantalla recuperar contraseña.

5.4.3. Pantalla de inicio.

Esta pantalla presenta la principal vista de la aplicación. Tal y como se puede observar, en ella tenemos representada la información más relevante de los partes de manera resumida, as acceso a las principales acciones de la aplicación.

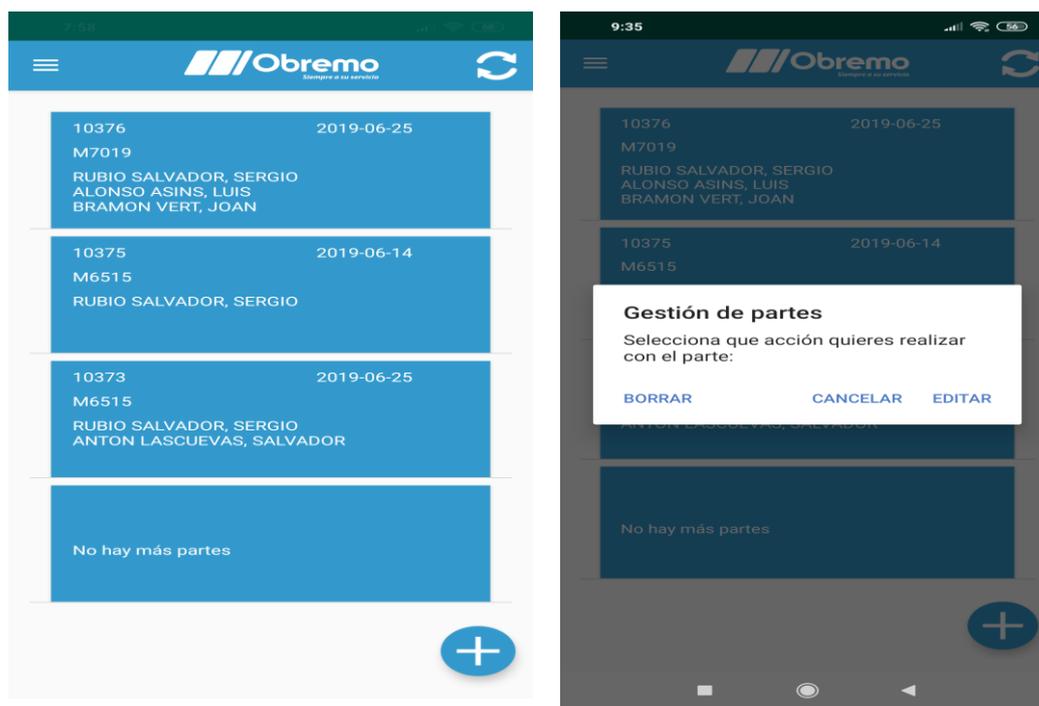


Ilustración 19 Arquímedes pantallas de inicio

Tal y como se puede observar en ambas imágenes, desde aquí realizaremos o accederemos a las funciones relacionadas con la gestión de partes. Podemos ver representados los casos de uso de consulta, edición, borrado o creación de partes.

5.4.4. Pantalla de datos del parte.

Esta vista es la encargada de ofrecer al usuario información sobre un parte que desea consultar en detalle y permitir la edición de datos de éste, o incluso permitir al usuario crear nuevos elementos para un parte.

Cada uno de los iconos de la barra inferior representa una acción que se puede realizar con los elementos que éste presenta. El primero de la izquierda resulta el icono de borrado de un elemento previamente seleccionado, situado en esa posición debido a su acceso e impidiendo así borrar un elemento no deseado por error, pues su acceso es posible, pero costoso.

El siguiente es el botón de cancelar, el cual permite volver al estado anterior en el que se encontraba el parte, cancelando cualquier modificación realizada sobre el mismo.

El siguiente es el botón de guardado, que envía el parte al *web service* si tiene disponible conexión a internet, Si el móvil no tiene conexión lo guardará en la base de datos interna del móvil.

El lápiz representa la acción de editar. Tras seleccionar un elemento, esta acción carga la información extendida de dichos elementos y permite la edición de éstos.

Finalmente, podemos observar el icono de adición. Este es el que realiza la inserción de un elemento en nuestro parte.

Si deseáramos cambiar la fecha a otro día distinto al indicado, podremos acceder a un calendario que ofrece al usuario una manera fácil e interactiva de seleccionar una nueva

fecha para el parte. Si lo que deseásemos fuera cambiar el centro de coste, tan solo necesitaríamos seleccionar el desplegable para seleccionar algún centro de coste a los que el operario tiene acceso.

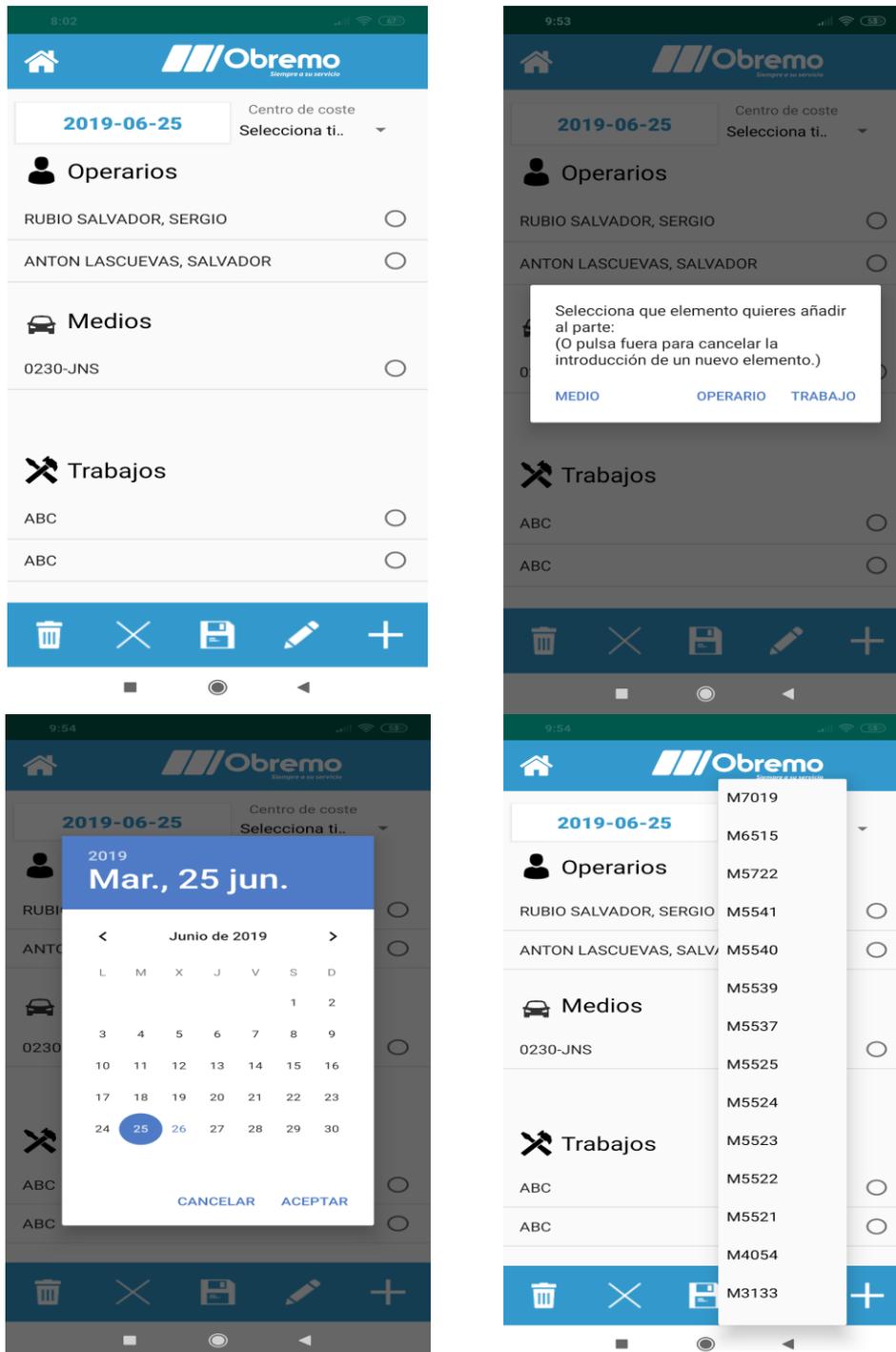


Ilustración 20. Arquímedes pantallas parte de trabajo

5.4.5. Pantalla añadir operario.

Esta pantalla cumple con la función de añadir un nuevo operario. Previamente a esta pantalla se ha utilizado un asistente que ha permitido al usuario rellenar esta información, aunque también tiene la posibilidad de omitir el uso de ese asistente si así lo desea y llegar a esta pantalla directamente.



The screenshot shows a mobile application interface for adding a worker. At the top, there is a blue header with a back arrow and the Obremo logo. Below the header, the title 'Operario' is displayed. The form contains the following fields: a name field with 'RUBIO SALVADOR, SERGIO', a code field with '6604', two time selection fields for 'Hora entrada' (08:00) and 'Hora salida' (14:00), and a field for 'Horas trabajadas' (4.0). At the bottom, there are two buttons: 'AÑADIR OPERARIO' and 'CANCELAR'.

Ilustración 21. Arquímedes pantalla nuevo operario.

Como se puede observar, la información introducida será la utilizada para generar la información del parte. Al igual que en el apartado anterior, la inserción de la hora de entrada y la hora de salida se ha realizado mediante una simulación de un reloj que ayuda al usuario a reconocer el trabajo que está realizando. De este modo se facilita el uso de éste creando una analogía con la realidad. Si insertan horas trabajadas, cambia el teclado del dispositivo permitiendo únicamente la inserción de valores numéricos decimales.

Por otro lado, el campo del nombre del operario se trata de un campo autocompletable que despliega opciones según los datos introducidos, reduciendo el tiempo de inserción del nombre del empleado.

El código del empleado, en caso de haber sido introducido primero completa automáticamente el campo del nombre del operario.

5.4.6. Pantalla añadir medio.

Esta pantalla permite al operario añadir un medio al parte. Para ello el operario ha de introducir un valor correcto en el campo matrícula y pulsar en el botón de búsqueda. Si es correcta la matrícula, podremos obtener el modelo del medio que queremos añadir al parte. Si no es correcto, obtendremos un valor que nos indica que no se ha encontrado ningún elemento correspondiente a dicho valor como matrícula.



Ilustración 22. Arquímedes pantalla nuevo medio.

5.4.7. Pantalla añadir trabajo.

El operario utilizará esta pantalla para añadir la información referente a los trabajos que se han realizado en este parte. Para ello utilizará el desplegable que proporciona el tipo de trabajo que va a realizar, escribiendo en el campo de texto central una descripción del trabajo realizado. Finalmente indicará la cantidad de trabajo que ha realizado, que se medirá en unidades o metros según la unidad de medida especificada en cada uno.

Al igual que con el apartado de añadir operario, esta vista también dispone de una opción guiada para añadir la información. Sin embargo, se puede cancelar u omitir esta opción.



Ilustración 23. Arquímedes pantalla nuevo trabajo.

5.4.8. Pantalla preferencias.

En esta pantalla el usuario puede organizar sus preferencias. Algunos son posibles datos que el operario añadirá en sus partes normalmente.

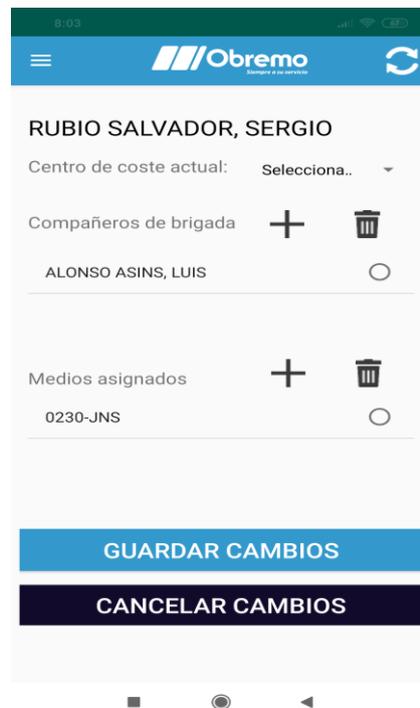


Ilustración 24. Arquímedes pantalla ajustes.

En la pantalla el operario puede acceder a la información, que puede organizar de modo que el asistente recupere automáticamente esta información al generar un nuevo parte.

Al acceder a la sección donde podemos ver los operarios guardados como compañeros de brigada, el asistente preguntará si deseamos unirlos al parte respecto a los medios que funciona de la misma manera que la forma de añadir operarios a la brigada.

También podemos guardar el centro de coste actual para almacenarlo si el usuario va a trabajar varios días en el mismo centro de coste. De esta forma se cargaría en el parte como primera opción.

5.4.9. Menú desplegable.

Pese a que este apartado no es una vista como tal, sí que es un apartado importante de nuestra aplicación.

En esta sección podemos encontrar una serie de botones que nos ofrecen acceso a distintas vistas, funcionando, así como un navegador entre las distintas pantallas.

También tenemos acceso a una breve información del usuario que está utilizando la aplicación, así como el botón que realiza la sincronización con la base de datos manualmente.

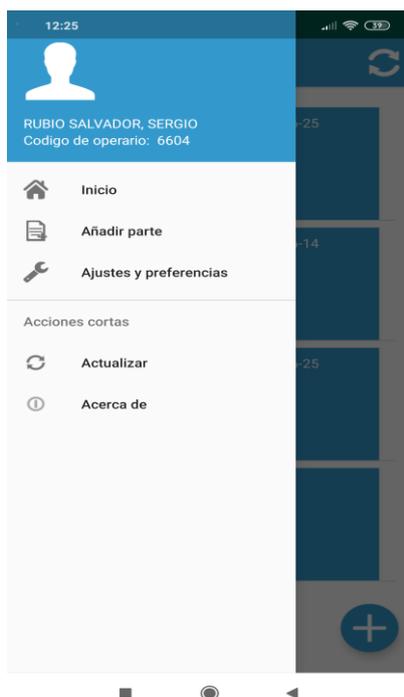


Ilustración 25. Arquímedes menú lateral.

5.4.10. Pantalla de contacto.

Esta pantalla es una pequeña pantalla que sirve de contacto y almacena distinta información referente a la aplicación, como su número de versión, un correo de contacto para notificar errores, sugerencias o felicitar aspectos positivos o información referente a los derechos de autor.

Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales

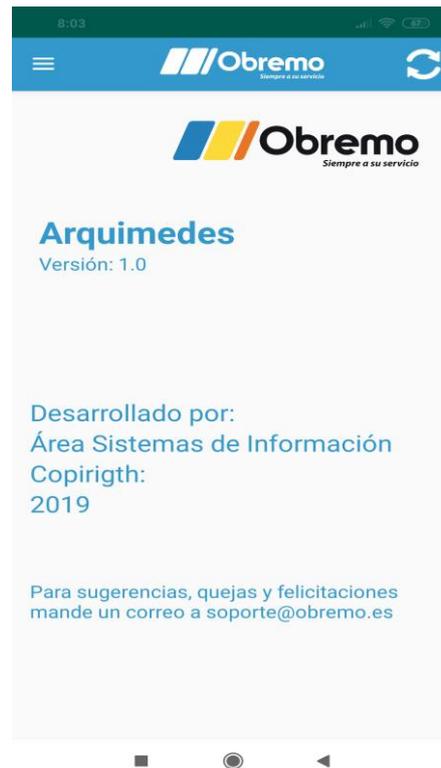


Ilustración 26. Arquímedes pantalla información.

6. Implantación.

En este apartado vamos a comentar el proceso de implantación del nuevo software desarrollado. En nuestro caso, este proceso se ha realizado escaladamente, aprovechando la estructura de la empresa primero se ha ofrecido a los usuarios probar y revisar los nuevos cambios que propusieron. Este cambio se realizó a lo largo de la segunda semana de marzo. Durante este tiempo, los usuarios no experimentaron ningún error, sin embargo, más tarde los usuarios experimentaron un error que producía ciertos fallos a la hora de obtener el importe por los trabajos realizados y propusieron un cambio que consistía en modificar el plazo del informe a una fecha seleccionada y no solo realizarlo del mes actual, tal y como se proponía.

Sin embargo, la aplicación para Android tiene el despliegue planteado a lo largo de la segunda semana de Julio tras finalizar el último Sprint el día 04/07/2019. Esto no impide que ciertos operarios se encuentren ya probando simultáneamente la aplicación a día de hoy, permitiendo ultimar los últimos detalles que pueden mejorar la aplicación.

6.1. Pruebas.

En el mundo del desarrollo de software una importante actividad que a menudo no es valorada respecto a su importancia es la realización de pruebas. Estas pruebas permiten al desarrollador detectar fallos y adelantarse a posibles cambios que pueden ser necesarios para mejorar los programas.

Nuestro software no es una excepción de esta regla, y como se ha comentado antes, este apartado es de vital importancia para nosotros, pues nos va a permitir detectar los flancos que puede presentar nuestra aplicación y permitir corregirlo antes de ser entregado a los operarios.

Debido a la naturaleza de nuestro software las pruebas elegidas para este van a ser pruebas de aceptación y pruebas de campo, ya que la funcionalidad de la aplicación y la movilidad son las principales características que diferencian nuestro sistema, especialmente la aplicación móvil, del resto de sistemas.

Mediante estas pruebas vamos a medir el cumplimiento de las especificaciones y los requisitos que el cliente ha demandado. Con esta técnica vamos a medir el grado de satisfacción de los *stakeholders* con el producto desarrollado, para ello vamos a diseñar un cuestionario que rellenarán, este cuestionario lo podemos encontrar en el apartado de anexos.

A parte de las pruebas de aceptación se ha probado la aplicación, debido a su naturaleza, en distintos ambientes. Por lo que se han realizado una serie de pruebas que permitan saber si la misma funciona correctamente en el ambiente de trabajo.

Para no molestar a los operarios lo que se ha realizado es salir a la calle a probar a insertar un parte de trabajo, modificarlo y borrarlo para ver la función principal en funcionamiento. Gracias a esto se han modificado diversos aspectos gráficos que han mejorado la visibilidad de información en distintas condiciones meteorológicas.

Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales

El cambio más relevante ha sido cambiar el color de la fuente al blanco, previamente con el color negro sobre el fondo azul con cierta exposición a la luz algunos puntos de la aplicación no se veían correctamente o costaban mucho de ver.

Por otro lado, a lo largo del desarrollo de nuestro código hemos realizado pruebas de *testing* unitario mediante JUnit [4]. Junit es un *framework open source* diseñado con el propósito de realizar pruebas unitarias para Java. Hemos escogido este framework pues *Android Studio* está preparado para integrar dentro de su sistema estas pruebas.

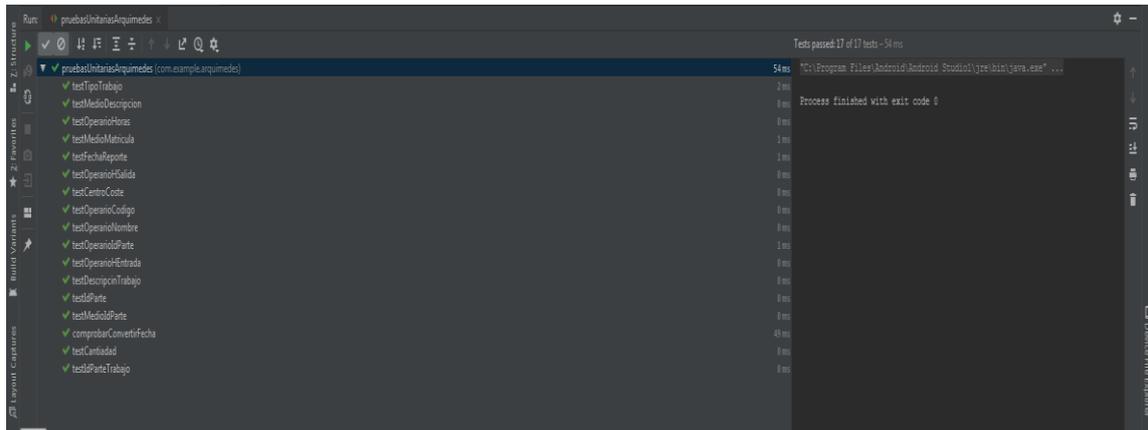


Ilustración 27. Resultados pruebas unitarias.

Esta es la pantalla de resultados de las pruebas al final de la etapa de desarrollo de software, en ellas se puede ver que los métodos que manejan los datos propios de los elementos del programa funcionan correctamente, además algunos métodos que gestionan el cambio de parámetros, como lo es “comprobarConvertirFecha”.

Gracias a la ejecución de estas pruebas podemos concluir que nuestro código que gestiona esta información no produce ningún error.

7. Conclusiones

A lo largo del trabajo realizado se han superado una serie de retos y de problemas que me han permitido mejorar y ampliar mis conocimientos. Además, me ha permitido experimentar el ámbito de desarrollo dentro de una empresa grande y la responsabilidad y presión que esto supone.

Como conclusión podemos decir que se han realizado las tareas que esperábamos, pues el sistema se ha migrado a la nueva versión web y la aplicación Android está en pruebas actualmente, muy próxima a su publicación para el uso por parte de los operarios que podrán disfrutarla en los próximos días.

Uno de los principales retos ha sido la integración entre dos bases de datos que contienen distinta información de usuarios, siendo el principal problema el obtener información de usuarios de *Active Directory* [9], pues la información no tenía un acceso claro y la especificación de LDAP que se ha encontrado no era del todo clara. Además, esta información era de naturaleza delicada, pues presenta información sobre todos los trabajadores de la organización. También la realización de la comunicación entre la aplicación móvil y el *web Service* ha planteado una serie de problemas que, con el tiempo, se han ido resolviendo, como el envío de información desde el cliente al servidor.

Sin embargo, pese a todos los contratiempos que he podido sufrir, el trabajo ha sido gratificante. En el tiempo que ha durado el mismo además de realizar algo que me gustaba y aprender de ello, los resultados obtenidos me han gustado, tanto la parte visual como el rendimiento de la aplicación. Además, he podido sentirme una pieza importante del equipo de trabajo, aumentando mi confianza en futuros proyectos.

7.1. Relación del trabajo con los estudios cursados.

Este trabajo denota el final de un grado de cuatro años, por lo que es interesante analizar qué conocimiento ha podido aplicar a lo largo de éste, así como los conocimientos adquiridos durante la relación de este trabajo.

Respecto a los conocimientos previos aplicados, comenzaremos con el uso de JAVA. Este lenguaje se comienza a dar desde primero de carrera en la asignatura de Introducción a la programación, y se va perfeccionando su conocimiento a lo largo de los cursos. Sin embargo, aplicar este lenguaje a las tecnologías y el estilo de programación para servicios web con STS o el uso del mismo para el desarrollo de la aplicación para Android utilizando los elementos propios del sistema operativo ha resultado un desafío importante.

Asimismo, se han adquirido nuevos conocimientos, como es el uso de SOAP para realizar el intercambio de información entre las distintas aplicaciones para agilizar la comunicación entre ellas. O el uso de la librería *Hibernate* para acceder a la base de datos.

También se han utilizado conocimientos de asignaturas como proyecto de ingeniería software o proceso de software, que han facilitado poder amoldarse a la metodología de desarrollo ágil [3] que presenta la empresa, mejorando la eficacia a la hora de desarrollar



las tareas y realizar una estimación del tiempo de desarrollo de cada una de los requisitos funcionales que el sistema presenta.

7.2. Relación del trabajo con las competencias transversales.

A lo largo de este trabajo hemos utilizado diversas competencias transversales que nos han permitido amoldarse a este trabajo, entre ellas podemos destacar las competencias de Diseño y proyecto, la cual se puede observar a lo largo del desarrollo del software. Sin embargo, la competencia transversal que más debemos destacar es la Competencia Transversal 6, trabajo en equipo y liderazgo, pues esta es la competencia que nos ha permitido gestionar la comunicación con el equipo de desarrollo de la empresa y con el resto de las áreas de la empresa (márquetin, los clientes, etc.).

También hemos de comentar algunas competencias que fueron utilizadas en menor medida, pero que también han sido importantes para el trabajo, entre ellas tenemos innovación, creatividad y emprendimiento o planificación y gestión del tiempo.

8. Trabajos futuros.

Este sistema no es un sistema cerrado y es posible que, en el futuro, los clientes puedan necesitar realizar cambios para amoldar el programa a nuevas necesidades o incluso expandir el área de trabajo a otras necesidades de la empresa, como el departamento de energías renovables, gas o los operarios encargados del mantenimiento de aguas. Por ello es interesante analizar los posibles flecos que pueden dificultar este hecho o las mejoras que se pueden realizar a lo largo del tiempo.

De entre las posibles mejoras, podemos destacar la inclusión de un panel de mando para mejorar la gestión de las brigadas por parte de los encargados desde el sistema web, o incluso una mejora que permita a los operarios mandar a los encargados revisiones sobre partes que ellos puedan crear necesarias, y que los encargados clasificarán para proponer una solución a esas revisiones.

Los centros de coste son una variable constante, así que permitir a los encargados definir nuevos centros de coste desde el mismo portal web puede ser una característica que permita al programa incrementar su eficacia.

Por otro lado, en cuanto a los trabajos en curso podemos encontrar los siguientes:

El primero de ellos sería la posibilidad de editar un compañero de brigada o editar un medio desde la pantalla de ajustes, sin tener que borrar y añadir un operario o un medio, a su vez sería interesante poder priorizar u ordenar esta lista, ofreciendo la posibilidad de organizar a sus compañeros según la preferencia del operario que rellena el parte.

El segundo trabajo en curso importante a destacar es aumentar un poco más la capacidad de personalización del usuario, permitiéndole escoger entre una serie de temas de su preferencia (modo nocturno, modo para daltónicos, etc) mejorando la accesibilidad del sistema.

La última tarea prevista, aunque no menos importante, es la posibilidad de permitir al operario realizar solo el *login* una vez al día, permitiendo el acceso a la aplicación de manera más directa si así él lo desea. Esta tarea también implica añadir la función de cerrar sesión, por si el operario la necesita.

Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales



9. Referencias.

- [1] Brad Frost, Atomic Design, versión 2016.
- [2] Android ACT team, Java Fundamentals for Android Development, Android ACT, versión 7 (2017).
- [3] Ken Schwaber, Agile Project Management with Scrum, Microsoft Press, versión 2004.
- [4] Vincent Massol with Ted Husted, Junit in action, Manning, 2004.
- [5] Doug Tidwell, James Snell and Pavel Kulchenko, Programming web services with SOAP, editor O'Reilly, versión 1 2001.
- [6] Guía de usuario Android Developers. Fecha última consulta: 12/04/2019 <https://developer.android.com/studio/intro?hl=es-419>
- [7] Spring tool suite features. Fecha última consulta: 02/07/2019 <https://spring.io/tools3/sts>
- [8] Jorge Martínez Ladrón de Guevara, Fundamentos de programación en JAVA, editorial EME
- [9] Joe Richards, Alistair G. Lowe-Norris y Robbie Allen, Active Directory, Grupo Anaya Comercial, 2007
- [10] Página principal Obremo. Fecha última consulta: 18/02/2019 <https://www.obremo.es/>
- [11] Manual de la marca. Fecha última consulta 10/04/2019 <https://socrates.obremo.es/SitePages/Manual%20de%20Marca%20Obremo.aspx>
- [12] Estadísticas sistemas operativos móviles más utilizados. Fecha última consulta: 23/04/2019 <https://netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Mobile%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Trend%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platform%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsMobile%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222018-07%22%2C%22dateEnd%22%3A%222019-06%22%2C%22segments%22%3A%22-1000%22%2C%22plotKeys%22%3A%5B%7B%22platform%22%3A%22Android%22%7D%2C%7B%22platform%22%3A%22iOS%22%7D%2C%7B%22platform%22%3A%22Unknown%22%7D%5D%7D>

10. Anexos.

Anexo 1. Encuesta de satisfacción de usuario.

Satisfacción usuario

Califica el desempeño de la aplicación según estas afirmaciones

	Muy bien	Bien	Regular	Mal	Muy mal
Arquímedes me parece una aplicación fácil de usar	<input type="radio"/>				
Creo que Arquímedes me permite reducir el tiempo de reportar un trabajo	<input type="radio"/>				
Me gusta el nuevo diseño	<input type="radio"/>				
La aplicación es rápida	<input type="radio"/>				
La información de la aplicación es clara	<input type="radio"/>				
Los colores de la aplicación representan a la empresa	<input type="radio"/>				
He tenido problemas utilizando la aplicación	<input type="radio"/>				
Creo que la aplicación podría mejorarse	<input type="radio"/>				

Desarrollo de un software de gestión de partes de trabajo en movilidad para una empresa de prestación de servicios integrales

El diseño me parece adecuado	<input type="radio"/>				
La aplicación es cómoda de usar	<input type="radio"/>				
Realiza la función que se espera de la misma	<input type="radio"/>				
Valoración general sobre la aplicación	<input type="radio"/>				

Si tiene alguna sugerencia de mejora puedes escribirla aquí:

Tu respuesta

ENVIAR

Nunca envíes contraseñas a través de Formularios de Google.