

Facultad de Informática

Grado de Ingeniería Informática

Trabajo Fin de Grado

Ingeniería del Software

Desarrollo de un sistema dual (web-móvil) para la entrega de Equipos de Protección Individual y para el Control de Accesos a las instalaciones de las empresas

Ania Berazaluce Berasategui

Junio 2018

Dirigido por: José Ángel Vadillo Zorita

Agradecimientos

Agradezco a mi familia por su apoyo incondicional. A mis padres, *Javier* y *Belén*, por sus consejos y por no dejar de creer en mí en ningún momento. A mi hermana *Maidier* por sus ánimos y por estar ahí cuando lo he necesitado. Y a mi abuela por compartir conmigo sus valores, sus pasiones y sus aventuras, por enseñarme a apreciar las pequeñas cosas y, en definitiva, por saber sacarme siempre una sonrisa, incluso en los días más grises.

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. En especial a *Irati*, por estar ahí, apoyarme y ayudarme, no solo durante el proyecto, sino durante estos cuatro años de carrera.

Quiero agradecer a *José Ángel Vadillo* por ser mi director del TFG y orientarme durante el desarrollo del proyecto.

A la empresa *Consultoría y Comunicaciones* por confiar en mí para la realización de este proyecto y permitirme aprender de los profesionales que en ella trabajan. A mis compañeros de Donosti y de Pamplona por amenizar las mañanas de trabajo. A todo el equipo de proyecto de *Unifikas*, por su paciencia y su acogida en el equipo. Y a *Ibón* y a *Ina*, por guiarme en mis primeros pasos dentro del mundo de las aplicaciones móviles.

Por último, agradecer a *David* por el apoyo moral y afectivo.

Resumen

Este documento corresponde a la memoria del proyecto de fin de grado “Desarrollo de un sistema dual (web-móvil) para la entrega de Equipos de Protección Individual y para el Control de Accesos a las instalaciones de las empresas”, desarrollado para la titulación de Grado en Ingeniería Informática en la especialidad de Ingeniería del Software, en la Facultad de Informática de la Universidad UPV/EHU de Donostia-San Sebastián.

El proyecto ha sido realizado en la empresa Consultoría y Comunicaciones, consultora tecnológica especializada en proyectos de transformación digital, a través de plataformas colaborativas, Business Analytics y movilidad. En concreto, se enmarca dentro de una amplia aplicación comercializada en empresas, *Unifikas*, de la que se hablará posteriormente.

Este proyecto tiene dos vertientes, una parte de desarrollo web y otra de movilidad. Además, la aplicación móvil hace uso de un servicio web con el objetivo de obtener el acceso a datos, que también forma parte del proyecto.

Índice de contenidos

Resumen	v
Índice de contenidos	vii
Índice de figuras	x
Índice de tablas	xii
1. Introducción	1
2. Antecedentes	4
2.1. Consultoría y Comunicaciones	5
2.2. Prácticas en CYC	5
2.3. Proyecto <i>Unifikas</i>	6
3. Planteamiento inicial	8
3.1. Objetivos	9
3.2. Alcance	10
3.3. Metodología de Desarrollo	10
3.4. Fases del proyecto.....	10
3.5. Estructura de descomposición del trabajo.....	12
3.6. Planificación temporal.....	17
3.7. Herramientas.....	20
3.7.1. Microsoft Visual Studio 2017	20
3.7.2. Microsoft SQL Server Management Studio	20
3.7.3. Microsoft Lync.....	20
3.7.4. Microsoft Word 2010	20
3.7.5. Drive	20
3.7.6. Xamarin.Forms	21
3.7.7. Xamarin.Controls.SignaturePad y Xamarin.Controls.SignaturePad.Forms	21
3.7.9. XamForms.Controls.Calendar	21
3.7.10. Toasts.Forms.Plugin	21
3.7.11. Rg.Plugins.Popup.....	21
3.7.12. Xam.Plugin.Media	21
3.8. Gestión de riesgos	22
4. Análisis de requisitos	25
4.1. Requisitos funcionales.....	26
4.1.1. Diagrama de casos de uso: Control de accesos	26

4.1.2. Diagrama de casos de uso: Entrega de EPIs	34
4.2. Requisitos de datos	47
4.2.1. Modelo de dominio: Control de accesos	47
4.2.2. Modelo de dominio: Entrega de EPIs	49
4.3. Modelo relacional de datos.....	51
4.3.1. Modelo relacional de datos: Control de accesos	52
4.3.2. Modelo relacional de datos: Entrega de EPIs.....	54
4.4. Requisitos no funcionales	55
4.4.1. Herramientas y tecnología utilizadas	55
4.4.2. Seguridad.....	55
5. Arquitectura y diseño	56
5.1. Arquitectura	57
5.1.1. Aplicación web	57
5.1.2. Aplicación móvil	58
5.1.3. Conexión entre ambas aplicaciones.....	59
5.2. Diseño de las aplicaciones.....	59
5.3.1. Diagrama de secuencia: Ver lista de entregas de un EPI en la app móvil.....	60
5.3.2. Diagrama de secuencia: Ver lista de entregas de un EPI en la app web.....	61
5.3.2. Diagrama de secuencia: Ver requisito de un trabajador en la app móvil	62
6. Implementación	63
6.1. Aplicación móvil	64
6.1.1. Aclaraciones previas.....	64
6.1.2. Vistas y su vínculo con los Modelos de Vista	65
6.1.3. Modelos de Vista y su conexión con el Servicio Web	68
6.1.4. Modelos propios de la aplicación y Modelos heredados por el Servicio Web	69
6.1.5. Servicio Web.....	71
6.2. Aplicación web	73
6.2.1. Capa de Transferencia de datos	73
6.2.2. Capa de Acceso a datos y su vínculo con las bases de datos	74
6.2.3. Capa de Lógica de negocio	75
6.2.4. Capa de presentación.....	76
7. Verificación y pruebas	78
Pruebas Prototipo 1	79
Pruebas Prototipo 2	81
Pruebas Prototipo 3	83

Pruebas Prototipo 4	84
Pruebas de Control de Accesos	85
Pruebas de Entrega de EPIS.....	88
8. Seguimiento del proyecto.....	92
8.1. Gestión del alcance	93
8.2. Gestión del tiempo	93
9. Conclusiones y futuras mejoras	96
9.1. Revisión de objetivos.....	97
9.2. Problemas durante la realización del TFG.....	97
9.3. Conclusiones del proyecto	98
9.4. Conclusiones desde un punto de vista personal	99
9.5. Trabajo futuro	100
10. Bibliografía.....	101
11. Anexos.....	104
10.1. Actas de reunión con la empresa	105
10.1. Actas de reunión con el tutor.....	108

Índice de figuras

Ilustración 1: Esquema de la Estructura de descomposición del trabajo	13
Ilustración 2: Diagrama de casos de uso, Control de accesos.....	26
Ilustración 3: Interfaz de búsqueda de trabajadores	27
Ilustración 4: Interfaz de resultados de búsqueda de trabajadores	27
Ilustración 5: Interfaz lista de pedidos	28
Ilustración 6: Interfaz detalle de un pedido	29
Ilustración 7: Interfaz ficha de trabajador.....	30
Ilustración 8: Interfaz lista de requisitos del trabajador	31
Ilustración 9: Interfaz lista de requisitos de la empresa	31
Ilustración 10: Interfaz lista de bloqueantes.....	32
Ilustración 11: Interfaz detalle de requisito	33
Ilustración 12: Diagrama de casos de uso, Entrega de EPIs	34
Ilustración 13: Interfaz búsqueda de EPIs.....	35
Ilustración 14: Interfaz resultados de la búsqueda de EPIs	35
Ilustración 15: Interfaz lista de entregas de un EPI.....	36
Ilustración 16: interfaz Pop Up filtrar entregas de un EPI.....	37
Ilustración 17: Interfaz detalle entrega de un EPI.....	38
Ilustración 18: Interfaz añadir entrega de un EPI.....	39
Ilustración 19: Interfaces Pop Up de búsqueda de una organización.....	40
Ilustración 20: Interfaces Pop Up búsqueda de un trabajador	41
Ilustración 21: Control campo de texto autocompletable para la búsqueda de productos.....	42
Ilustración 22: Control personalizado para la búsqueda de productos	42
Ilustración 23: Lista de entregas de un EPI en la web	43
Ilustración 24: Interfaz detalle de una entrega de un EPI en la web	44
Ilustración 25: Interfaz detalle agregar entrega de un EPI en la web	45
Ilustración 26: Interfaz filtros de entregas de un EPI en la web	46
Ilustración 27: Interfaz la página principal de entregas de un determinado EPI en la web	46
Ilustración 28: Modelo de dominio, Control de accesos.....	47
Ilustración 29: Modelo de dominio, Entrega de EPIs	49
Ilustración 30: Tabla de Centros de trabajo	51
Ilustración 31: Modelo relacional de datos, Control de accesos	52
Ilustración 32: Modelo relacional de datos, Entrega de EPIs.....	54

Ilustración 33: Arquitectura de <i>Unifikas</i> (Autor: Valeriano Alonso)	57
Ilustración 34: Diagrama de secuencia. Ver lista de entregas de un EPI en la app móvil	60
Ilustración 35: Diagrama de secuencia. Ver lista de entregas de un EPI en la app web	61
Ilustración 36: Diagrama de secuencia. Ver requisito de un trabajador en la app móvil	62
Ilustración 37: Estructura de la app móvil.....	64
Ilustración 38: Gráfica comparativa de la dedicación estimada y real	95

Índice de tablas

Tabla 1: Planificación temporal.....	17
Tabla 2: Calendario de trabajo del proyecto.....	19
Tabla 3: Batería de pruebas del prototipo 1	80
Tabla 4: Batería de pruebas del prototipo 2	83
Tabla 5: Batería de pruebas del prototipo 3	84
Tabla 6: Batería de pruebas del prototipo 4. Funcionalidad: Control de Accesos.....	88
Tabla 7: Batería de pruebas del prototipo 4. Funcionalidad: Entrega de EPIs.....	91
Tabla 8: Comparativa entre la dedicación estimada y real	94

1

Introducción

En este capítulo se dota de contexto al trabajo de fin de grado desarrollado. Este proyecto tiene dos vertientes, una parte de desarrollo web y otra de movilidad.

Por otro lado, se realiza una pequeña introducción de las distintas secciones en las que se divide esta memoria, de manera que el lector pueda ver el alcance del proyecto.

Este documento corresponde a la memoria del proyecto realizado por la alumna Ania Berazaluce Berasategui, en la empresa Consultoría y Comunicaciones, cuyo nombre comercial es CYC, tanto en la sede de Pamplona como en la delegación de San Sebastián. El trabajo se ha desarrollado de manera autónoma bajo la supervisión de Carolina Elduayen como jefa de proyecto, y de Ibon Gañán como responsable de movilidad de CYC.

El objetivo principal del proyecto es el desarrollo de nuevas funcionalidades para la aplicación *Unifikas*, un software diseñado para gestionar de forma integrada la Prevención de Riesgos Laborales, Medioambiente y Calidad de las empresas. Tendrá un componente de movilidad y otro de desarrollo de la aplicación web de *Unifikas*, cubriendo la funcionalidad de entrega y seguimiento de Equipos de Protección Individual (en adelante, EPIs) en ambos entornos y la funcionalidad de control de accesos a las instalaciones de la empresa por personal ajeno, ésta únicamente en la aplicación móvil.

A continuación, se describen los apartados en los que se estructura el presente documento:

- En apartado de **antecedentes** del proyecto, se describe la organización CYC y el producto *Unifikas*. Además, se menciona la experiencia previa en dicha empresa.
- En apartado de **planteamiento inicial** del proyecto, se detallan los objetivos y el alcance del mismo, la metodología seguida para su realización, las fases y las tareas de cada una de estas fases en las que se ha dividido el proyecto, la planificación temporal del proyecto y, por último, las herramientas utilizadas para el desarrollo del mismo.
- En apartado de **análisis** del proyecto, se especifican los requisitos funcionales mediante la identificación de los casos de uso; los requisitos de datos ilustrados mediante modelos de dominio, los modelos relacionales de datos utilizados para estructurar las bases de datos del proyecto; y los requisitos no funcionales a tener en cuenta a lo largo del proyecto.
- En el apartado de **arquitectura y diseño** del proyecto, se explica la arquitectura del proyecto, la cual enlaza la aplicación web y la aplicación móvil a desarrollar durante el transcurso del mismo. Para realizar el diseño se han utilizado diagramas de secuencia, una muestra de los cuales quedan reflejada en este apartado.
- En el apartado de **implementación** del proyecto, se describe el proceso de desarrollo para cada una de las aplicaciones implementadas.

- En el apartado de **pruebas** del proyecto, se especifican las pruebas realizadas para dar por concluida cada una de las iteraciones del proyecto.
- En el apartado de **seguimiento** del proyecto, se detalla la gestión del alcance y del tiempo realizadas a lo largo del ciclo de vida del proyecto
- En el apartado de **conclusiones** del proyecto, se recogen las conclusiones tanto a nivel de objetivos y del proyecto en general, como a nivel personal. Por otro lado, se describen brevemente las propuestas de extensión y mejora del trabajo realizado.

2

Antecedentes

Para la comprensión en profundidad de las características de este proyecto es importante conocer el contexto en el que se sitúa y se desarrolla. Este apartado da comienzo describiendo el contexto empresarial en el que se enmarca y el producto del que se parte.

2.1. Consultoría y Comunicaciones

Este Trabajo Fin de Grado (en adelante TFG) se desarrolla en la empresa Consultoría y Comunicaciones¹, cuyo nombre comercial es CYC, que se utilizará en adelante al referirse a ella.

CYC es una consultora tecnológica con más de 20 años de historia, especializada en proyectos de transformación digital, a través de plataformas colaborativas, *Business Analytics* y movilidad. Cuenta con el Certificado CMMI de nivel 2, avalando su capacidad de desarrollar software de alta calidad

Su principal objetivo es diseñar estrategias personalizadas y desarrollar las soluciones tecnológicas adaptadas a las necesidades de los clientes, para mejorar su forma de trabajar, de compartir documentación y conocimiento, así como para aumentar su productividad favoreciendo su transformación digital.

La sede central del Grupo se encuentra en Pamplona, además cuenta con delegaciones en San Sebastián, Zaragoza, Madrid y Bilbao.

2.2. Prácticas en CYC

El comienzo de mi vida laboral se remonta al verano de 2017, cuando, de la mano de CYC, empecé un periodo de prácticas en la empresa. El objetivo principal de estas prácticas era claro: conocer un entorno laboral profesional, integrándome en un equipo de trabajo estable, multidisciplinar y muy consolidado, con una sólida metodología de trabajo tanto a nivel de desarrollo como de gestión de versiones y despliegue de las mismas.

Las prácticas consistieron en el desarrollo de nuevas funcionalidades para la aplicación web de gestión *Unifikas*², una aplicación en .NET desarrollada en C# cuyo motor de bases de datos es SQL Server. Debido a que este proyecto lleva años en marcha, las dos primeras semanas se centraron en la formación acerca del alcance y la estructura interna de la aplicación, la metodología de trabajo utilizada y la gestión interna empleada.

Aunque durante el tercer curso de carrera ya se había trabajado en una pequeña aplicación en .NET, el salto a una aplicación de tales dimensiones con una metodología de trabajo tan rigurosa, fue una oportunidad única para el aprendizaje.

¹ <https://www.cyc.es/default.aspx>

² <https://www.cyc.es/serviciosproductos/productos/unifikas.aspx>

La realización de estas prácticas supuso una buena relación con la empresa tanto a nivel personal como por el trabajo realizado, lo que desembocó en la oportunidad de realizar ahí el proyecto de fin grado.

2.3. Proyecto *Unifikas*

El proyecto *Unifikas* es un software diseñado para gestionar de forma integrada la Prevención de Riesgos Laborales, Medioambiente y Calidad de las empresas. Esta herramienta unifica los tres ámbitos de gestión en busca de un objetivo común: la excelencia de las organizaciones.

Unifikas es un sistema integrado de gestión creado en base a la normativa internacional vigente en las áreas de Prevención de riesgos laborales (OSHAS 18001:2007), Medio ambiente (ISO 14001:2004) y Calidad (ISO 9001:2008). La aplicación recoge todos los requisitos para que las empresas puedan identificar, establecer y mantener con orden y rigor, todos los procesos para cumplir con esas normativas internacionales y conseguir la mejora y evolución positiva de las actuaciones de las organizaciones en estos ámbitos.

Los principales beneficios que ofrece la utilización de *Unifikas* son:

1. Eficiencia: La integración y estandarización de procesos permite simplificar y mejorar la gestión de las áreas previamente citadas.
2. Ahorro: Involucrar a la organización de manera más global deriva en un ahorro de tiempo y de recursos en la organización.
3. Normativa: Facilita el cumplimiento de la normativa internacional y corporativa vigente en estas áreas.
4. Facilita auditorías: Más sencillez debido a la integración y al registro completo de información y cambios durante el tiempo.
5. Plataforma común: Gestión centralizada en una aplicación multi-país, multi-cultura, multi-idioma, multi-empresa y multi-planta.
6. Adaptabilidad: Capacidad de modificar/ampliar algunos módulos para moldearlos según las necesidades específicas de la empresa.
7. Compatibilidad: Integrable con los sistemas de gestión/ERPs y hardware propios de la organización.
8. Información: Registro detallado de empresas, trabajadores, maquinaria, productos, etc. internos y externos a la organización.

Por ello, grandes empresas han decidido confiar en CYC, implantando este software para la Prevención de Riesgos Laborales, Medioambiente y Calidad.

La aplicación consta de varios módulos con la finalidad de lograr una gestión eficaz y organizada de la prevención de Riesgos Laborales, Medioambiente y Calidad.

Uno de ellos es el módulo de Productos, donde se homologan proveedores y productos además de mantener el registro de la documentación al respecto. Permite el registro de la documentación, stock y revisiones de todos los bienes, bienes prestados, equipamientos y servicios. En él se pueden definir requisitos a cumplir y flujos de aprobación entre roles para realizar nuevas compras.

En este módulo el responsable debe definir requisitos a cumplir y flujos de aprobación entre roles para realizar nuevas compras. Así se realiza una gestión de maquinaria, productos químicos y Equipos de Protección Individual (EPIs - lista de proveedores homologados). Actualmente, la única herramienta de gestión para controlar la entrega de EPIs consta en vincular un trabajador con el EPI recibido, sin embargo, esto no es suficiente ya que siempre se necesita recoger la fecha en la que fue entregado. Además, tener la opción de poder dejar documentada la fecha de reposición del equipo, las unidades entregadas y poder firmar al recibir el EPI sería un gran valor añadido para esta funcionalidad.

Otro de los módulos es el de Coordinación de actividades empresariales, que facilita la gestión de la documentación a exigir al personal externo de una empresa antes de acceder a las instalaciones y permite establecer un "Control de Accesos", aceptando o denegando la entrada a la empresa en función de dicha documentación. Además, mediante alertas de aviso, se puede informar de manera automática de aquellos documentos que vayan a caducar.

Aunque en la aplicación web todos estos módulos están ya en funcionamiento, la aplicación móvil únicamente cuenta con una pequeña parte de todas estas funcionalidades implementada. De entre todas las posibles funcionalidades a implementar en la aplicación móvil, dadas las principales necesidades de los clientes de *Unifikas*, tanto el control de accesos como la entrega de EPIs serían prioritarios.

3

Planteamiento inicial

Este proyecto de fin de grado se enmarca dentro de *Unifikas*, proyecto presentado en el punto 2.3. En concreto, dentro del módulo de Productos, el encargado de homologar proveedores y productos, además de mantener el registro de la documentación al respecto; se desarrollará la funcionalidad entrega de EPIs tanto para la aplicación web como para la aplicación móvil. Adicionalmente, como parte exclusiva de la aplicación móvil, se desarrollará la funcionalidad de control de accesos, cuya finalidad se centra en gestionar la documentación a exigir al personal externo de la empresa antes de acceder a las instalaciones. Mediante esta aportación a *Unifikas*, se pretende darle un valor añadido ampliando la capacidad de gestión de riesgos y de personal de la empresa.

A continuación, se describen los objetivos, el alcance, la metodología de trabajo seguida y la planificación del proyecto. Además, se dará un repaso a las herramientas y tecnologías utilizadas. Por último, se presentarán los riesgos potenciales a los que está expuesto el proyecto y el plan de prevención y contingencia para cada uno de ellos.

3.1. Objetivos

Uno de los principales objetivos es lograr ayudar a combatir los riesgos de accidentes y de perjuicios para la salud de los empleados mediante una gestión rigurosa de la entrega de EPIs. Se entiende por equipo de protección individual (EPI), cualquier equipo destinado a ser llevado o sujetado por el trabajador para que le proteja de uno o varios riesgos que puedan amenazar su seguridad o su salud, así como cualquier complemento o accesorio destinado a tal fin.

La segunda funcionalidad abordada radica en la necesidad de controlar el acceso de personal externo a las instalaciones de la empresa. Fijando como objetivo el control total de este tipo de subcontratas, de manera que se puedan gestionar los servicios a los que están destinados y los requisitos de acceso tanto de los trabajadores como de la organización de la que proceden.

Dentro de los objetivos secundarios del proyecto está el aprendizaje de desarrollo de aplicaciones móviles. Debido a la creciente demanda de personal especializado en el desarrollo de software y hardware para dispositivos móviles, la realización de este proyecto asentará las bases de este amplio sector, brindando nuevas oportunidades de futuro.

Por último, se fijará como objetivo la realización de un proyecto de manera autónoma. Durante los estudios de grado, la realización de trabajos grupales ha sido una de las bases del proceso formativo, estos proyectos siempre han buscado aplicar y ampliar lo aprendido en clase. Sin embargo, durante la realización de este proyecto individual se deberán aplicar los distintos conocimientos adquiridos durante el transcurso del mismo, con el fin de demostrar que se está capacitado para la vida laboral.

3.2. Alcance

El objeto del proyecto es implementar dos nuevas funcionalidades para *Unifikas*, la funcionalidad de control de accesos únicamente para la aplicación móvil y, la funcionalidad de entrega de Equipos de Protección Individual, tanto para la aplicación móvil como para la aplicación web. Teniendo en cuenta que este proyecto se desarrolla en un entorno laboral, el estado final de las funcionalidades se adecuará a las necesidades de *Unifikas*. Así pues, se implantará en las empresas que tengan contratado el software respectivo.

3.3. Metodología de Desarrollo

Se ha seguido el Proceso Unificado, cuyas principales características son el desarrollo iterativo e incremental. Es decir, completar un sistema informático en varias iteraciones dotándole en cada una de ellas de mayor funcionalidad. Este proyecto se divide en siete iteraciones, precedidas de un periodo de estudio para la correcta realización de las mismas. El alcance de cada iteración será comentado en las fases del proyecto, y su planificación temporal en el apartado con este mismo nombre. Como pilar alrededor del cual dirigir el proyecto, se han definido distintos casos de uso para cada funcionalidad, los cuales se concretarán posteriormente.

3.4. Fases del proyecto

En este apartado se describen las distintas fases y funcionalidades que caracterizarán al proyecto para poder cumplir los objetivos definidos anteriormente. Las principales fases por las que se atravesará y que desembocarán en la realización del proyecto son:

- **Gestión:** fase que durará todo el ciclo de vida del proyecto. En ella se definirá el proyecto, se concretarán las distintas tareas, se elaborará la planificación inicial y se controlará el cumplimiento de la misma. Además, si fuera necesario, se llevará a cabo una gestión de riesgos, se realizará un replanteamiento, etc.
- **Formación:** en esta fase se adquirirán los conocimientos necesarios para poder desarrollar correctamente la fase de implementación. Aunque la mayoría de dedicación a esta fase se invertirá al principio del proyecto, se alargará durante todo el ciclo de vida del mismo.
- **Análisis:** fase en la que se recogerán los requisitos y funcionalidades exigidos por la empresa para la realización del proyecto. Además de datos y requisitos no funcionales.

- **Diseño:** en esta fase se definirán tanto las interfaces de las distintas funcionalidades como las aportaciones a la base de datos y el servicio web necesarias para la realización del proyecto.
- **Implementación:** se definirán los distintos prototipos en los que se dividirá el desarrollo del proyecto y se implementarán. Adicionalmente, se desarrollarán nuevos métodos en el servicio web que importa la aplicación móvil para su posterior explotación. En este caso, debido a que se desarrollarán dos funcionalidades distintas y a que una de ellas se desarrollará tanto para la aplicación web como para la aplicación móvil, los prototipos a desarrollar serán los siguientes:
 - Funcionalidad: Control de accesos en la aplicación móvil.
 - *Prototipo 1.1:* Implementar en la aplicación móvil la funcionalidad de control de accesos.
 - *Prototipo 1.2:* Añadir nuevas características al control de accesos, refactorizar el código para que cumpla el patrón MVVM.
 - Funcionalidad: Entrega de EPIs en la aplicación web.
 - *Prototipo 2.1:* Implementar la entrega de EPIs a los trabajadores en la aplicación web.
 - Funcionalidad: Entrega de EPIs en la aplicación móvil.
 - *Prototipo 3.1:* Implementar la entrega de EPIs a los trabajadores en la aplicación móvil.
 - *Prototipo 3.2:* Conseguir firmar cuando se entrega un EPI desde la aplicación móvil.
 - *Prototipo 4:* Incluir el acceso a datos, mediante el Servicio Web, en ambas funcionalidades de la aplicación móvil.

Estos prototipos no se implementarán en orden sino que se irán adaptando a las necesidades de la empresa.

- **Pruebas:** al plantear la fase de implementación como una sucesión de prototipos, se realizará un plan de pruebas para cada prototipo para garantizar el correcto funcionamiento del mismo. Asimismo, al terminar la fase de implementación, se realizarán un conjunto de pruebas que garanticen el correcto funcionamiento de la aplicación en conjunto.
- **Documentación:** fase ejecutada en paralelo al proyecto, donde se elaborará un documento que recogerá tanto el proyecto realizado como lo acontecido durante la realización del mismo. Además, se preparará la defensa del mismo.

3.5. Estructura de descomposición del trabajo

La Estructura de Desglose del Trabajo (EDT) es una descomposición jerárquica, orientada al producto entregable del trabajo que será ejecutado por el equipo del proyecto, para lograr los objetivos del proyecto y crear los productos entregables requerido. Así pues, en el siguiente árbol jerárquico, se desglosan las tareas que componen el proyecto:

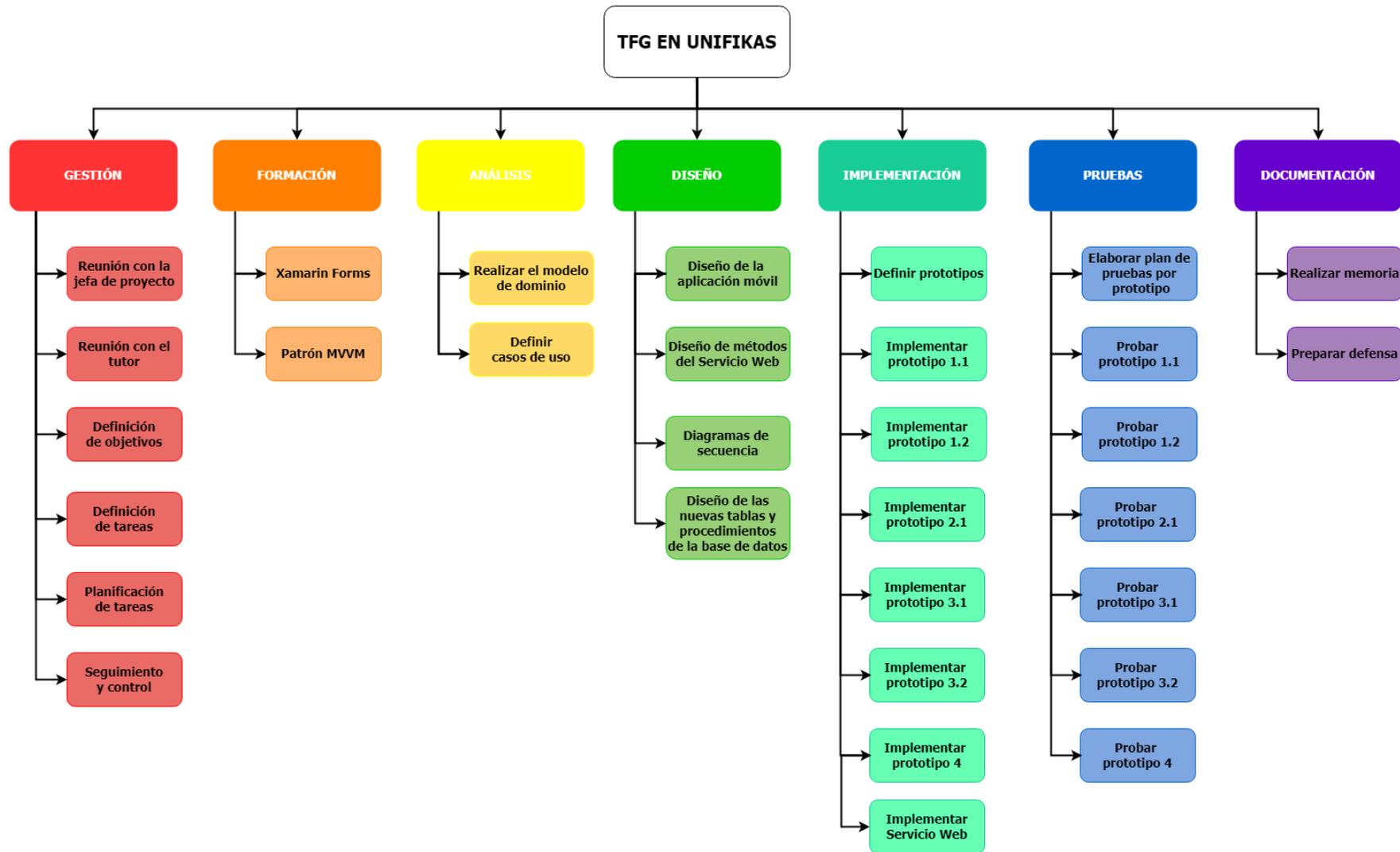


Ilustración 1: Esquema de la Estructura de descomposición del trabajo

A continuación, se detallan las tareas previamente citadas, dotándolas del número de horas estimadas a su dedicación:

Gestión:

- Reuniones con la jefa de proyecto: Además de las reuniones de inicio y fin de proyecto, se ha programado una reunión al final de cada iteración en la que se enseñarán los resultados de lo trabajado durante la misma, proponiendo mejoras y planificando la siguiente iteración a partir de ellas. Duración: 10 horas.
- Reuniones con el tutor: Aunque el proyecto esté enfocado a un entorno laboral externo y no englobado en prácticas regladas universitarias, la orientación del tutor se considera primordial para que el proyecto salga adelante. Duración: 10 horas.
- Definición de objetivos: Fijar los objetivos que se van a llevar a cabo a lo largo del proyecto. Duración: 2 horas.
- Definición de tareas: Establecer las tareas en las que se divide el proyecto, procurando desglosar las principales necesidades del mismo. Duración: 3 horas.
- Planificación de tareas: Fijar un plan con el cual desarrollar todas las tareas definidas. Duración: 3 horas.
- Seguimiento y control: Establecer el conjunto de acciones que se llevarán a cabo para la comprobación de la correcta ejecución de las actividades del proyecto establecidas en la planificación del mismo. Duración: 10 horas.

En total, se invertirán 38 horas en la fase de gestión.

Formación

Gracias a las prácticas de verano realizadas en la empresa, no es necesario incluir la formación relacionada con la aplicación web de *Unifikas*, acotando el estudio a la parte relativa a la aplicación móvil del proyecto.

- Xamarin Forms³: Estudio de la API utilizada en la empresa para la construcción de aplicaciones móviles nativas en Android, IOS y Windows. Duración: 35 horas.
- Patrón MVVM⁴: Recabar información sobre este patrón de arquitectura de software. Duración: 1 horas.

En total, se invertirán 36 horas en la fase de formación.

³ https://download.microsoft.com/DOWNLOAD/7/8/8/788971A6-C4BB-43CA-91DC-557B8BE72928/MICROSOFT_PRESS_EBOOK_CREATINGMOBILEAPPSWITHXAMARINFORMS_PDF.PDF

⁴ <https://msdn.microsoft.com/en-us/library/hh848246.aspx>

Análisis

La analista del proyecto *Unifikas* se encarga de realizar la captura de requisitos y de definir las funcionalidades de la aplicación. De modo que hay que procesar dichas funcionalidades en términos de casos de uso, resolver ambigüedades y definir el modelo del dominio:

- Definir modelo de dominio: Estudio de las necesidades de la base de datos y la relación entre las diferentes tablas, tanto las previamente creadas como las nuevas a generar para la resolución del proyecto. Duración: 3 horas.
- Definir casos de uso: Elaboración del diagrama de casos de uso de las distintas funcionalidades a desarrollar. Duración: 3 horas.

En total, se invertirán 6 horas en la fase de análisis.

Diseño

- Diseño de la aplicación móvil: Diseño en papel de las interfaces de la aplicación móvil. Duración: 2 horas.
- Diseño de métodos del Servicio Web para la aplicación móvil: La aplicación móvil obtiene los datos de un Servicio Web, el cual tendrá que ser modificado para dar soporte a las nuevas funcionalidades. Duración: 2 horas.
- Diseño de nuevas tablas y procedimientos de BD para aplicación web: Diseño de la base de datos a utilizar. Duración: 1 hora.
- Diagramas de secuencia: Realización de los diagramas de secuencia de la aplicación. Duración: 20 horas.

En total, se invertirán 25 horas en la fase de diseño.

Implementación

- Definir iteraciones: Especificar en cuantas iteraciones se divide el proyecto y el alcance de cada uno de ellas, cada iteración resultará en un nuevo prototipo de la aplicación. Duración: 1 hora.
- Implementar prototipo 1.1: Implementar en la aplicación móvil la funcionalidad de control de accesos. Duración: 50 horas.
- Implementar prototipo 1.2: Añadir nuevas características al control de accesos, refactorizar el código para que cumpla el patrón MVVM e incluir el acceso a datos mediante el Servicio Web. Duración: 45 horas.
- Implementar prototipo 2: Implementar la entrega de EPIs a los trabajadores en la aplicación web. Duración: 40 horas.

- Implementar prototipo 3.1: Implementar la entrega de EPIs a los trabajadores en la aplicación móvil. Duración: 40 horas.
- Implementar prototipo 3.2: Conseguir firmar cuando se entrega un EPI desde la aplicación móvil. Duración: 40 horas.
- Implementar Servicio web: Con la finalidad de que la aplicación móvil tenga el mismo acceso a datos que la aplicación web, se implementarán un conjunto de métodos para consumir desde el Servicio Web. Duración: 30 horas.
- Implementar prototipo 4: Incluir los nuevos métodos del servicio web en la aplicación móvil y probar su correcto funcionamiento. Duración: 20 horas.

En total, se invertirán 276 horas en la fase de implementación.

Pruebas

- Elaborar plan de pruebas por prototipo: Desarrollar un plan de pruebas que, ajustándose a la implementación llevada a cabo, cubra todas las posibilidades de error. Duración: 6 horas
- Probar prototipo 1.1. Duración: 2 horas.
- Probar prototipo 1.2. Duración: 2 horas.
- Probar prototipo 2. Duración: 2 horas.
- Probar prototipo 3.1. Duración: 2 horas.
- Probar prototipo 3.2. Duración: 2 horas.
- Probar prototipo 4. Duración: 4 horas.

En total, se invertirán 20 horas en la fase de pruebas.

Documentación

- Realizar memoria: Recoger en un documento tanto la labor realizada durante el desarrollo de las distintas funcionalidades como la documentación generada para la realización del proyecto. Duración: 50 horas.
- Preparar defensa: Preparar la defensa del proyecto con el objetivo de presentarlo frente a un tribunal. Duración: 20 horas.

En total, se invertirán 70 horas en la fase de documentación.

Tras el desglose de las fases del proyecto y la especificación del tiempo estimado para la realización de cada una, podemos concluir que la dedicación total planificada para la elaboración completa del proyecto es de 471 horas.

3.6. Planificación temporal

En este apartado se definirá la estimación de horas de dedicación para cada una de las tareas, además, así como su distribución temporal a lo largo del primer semestre de 2018.

Tareas	Dedicación estimada
Gestión	38
Reuniones con la jefa de proyecto	10
Reuniones con el tutor	10
Definición de objetivos	2
Definición de tareas	3
Planificación de tareas	3
Seguimiento y control	10
Formación	36
Xamarin Forms	35
Patrón MVVM	1
Análisis	6
Definir modelo de dominio	3
Definir casos de uso	3
Diseño	25
Diseño de la aplicación móvil	2
Diseño de métodos del Servicio Web para la app móvil	2
Diseño de nuevas tablas y procedimientos de BD para app web	1
Diagramas de secuencia	20
Implementación	276
Definir prototipos	1
Implementar prototipo 1.1	50
Implementar prototipo 1.2	45
Implementar prototipo 2	40
Implementar prototipo 3.1	40
Implementar prototipo 3.2	40
Implementar prototipo 4	40
Implementar Servicio Web	20
Pruebas	20
Elaborar plan de pruebas por prototipo	6
Probar prototipo 1.1	2
Probar prototipo 1.2	2
Probar prototipo 2	2
Probar prototipo 3.1	2
Probar prototipo 3.2	2
Probar prototipo 4	4
Documentación	70
Realizar memoria	50
Preparar defensa	20
Total	471

Tabla 1: Planificación temporal

Se ha realizado un calendario con el fin de repartir en el tiempo estas tareas, y gestionar temporalmente el proyecto con mayor facilidad.

ENERO									
	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo		
1	2	3	4	5	6	7			
8	9	10	11	12	13	14			
				INICIO PROYECTO					
				Estudio					
				Reunión inicio proyecto					
15	16	17	18	19	20	21			
Estudio									
22	23	24	25	26	27	28			
1ª Iteración: Prototipo 1.1									
29	30	31							
1ª Iteración: Prototipo 1.1									

FEBRERO									
	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo		
				1	2	3	4		
				1ª Iteración: Prototipo 1.1					
					Reunión				
5	6	7	8	9	10	11			
2ª Iteración: Prototipo 1.2									
12	13	14	15	16	17	18			
2ª Iteración: Prototipo 1.2									
					Reunión				
19	20	21	22	23	24	25			
3ª Iteración: Prototipo 2.1									
26	27	28							
3ª Iteración: Prototipo 2.1									

MARZO									
	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo		
				1	2	3	4		
				3ª Iteración: Prototipo 2.1					
					Reunión				
5	6	7	8	9	10	11			
4ª Iteración: Prototipo 3.1									
12	13	14	15	16	17	18			
4ª Iteración: Prototipo 3.1									
					Reunión				
19	20	21	22	23	24	25			
5ª Iteración: Prototipo 3.2									
Estudio									
26	27	28	29	30	31				
5ª Iteración: Prototipo 3.2									
					Reunión				

ABRIL						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
						1
2	3	4	5	6	7	8
6ª Iteración: Desarrollo Servicio Web						
9	10	11	12	13	14	15
7ª Iteración: Prototipo 4						
16	17	18	19	20	21	22
7ª Iteración: Prototipo 4				Reunión		
23	24	25	26	27	28	29
Documentación						
30						
Documentación						

MAYO						
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
	1	2	3	4	5	6
	Documentación					
	7	8	9	10	11	12
	Documentación			Documentación		
	14	15	16	17	18	19
	Preparar defensa					
	21	22	23	24	25	26
	Preparar defensa					
	28	29	30	31		

Tabla 2: Calendario de trabajo del proyecto

3.7. Herramientas

3.7.1. Microsoft Visual Studio 2017⁵

Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación, aunque únicamente se haya utilizado C#, a lo cual hay que sumarle las nuevas capacidades online bajo Windows Azure.

Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y consolas, entre otros.

3.7.2. Microsoft SQL Server Management Studio⁶

SQL Server Management Studio (SSMS) es un entorno integrado para administrar cualquier infraestructura de SQL. Se puede utilizar SSMS para acceder a todos los componentes de SQL Server, Azure SQL Database y SQL Data Warehouse, y para configurarlos, administrarlos y desarrollarlos. Además, ofrece una única utilidad integral que combina un amplio grupo de herramientas gráficas con una serie de editores de script enriquecidos que permiten a desarrolladores y administradores de bases de datos de todos los niveles acceder a SQL Server.

3.7.3. Microsoft Lync⁷

Microsoft Lync o Skype empresarial es un servicio de mensajería instantánea, gracias al cual se mantendrá el contacto con el equipo de desarrollo de *Unifikas* para coordinarse en la realización de las distintas tareas de las fases de desarrollo y prueba. Además, algunas de las reuniones de seguimiento se realizarán a través de esta herramienta.

3.7.4. Microsoft Word 2010

Microsoft Word es un programa informático orientado al procesamiento de textos, se utilizará para el desarrollo de la memoria y la documentación.

3.7.5. Drive⁸

Drive es un servicio de alojamiento de archivos multiplataforma en la nube. El servicio permite a los usuarios almacenar y sincronizar archivos en línea y entre ordenadores y compartir archivos y carpetas con otros usuarios, además de con móviles y tabletas.

⁵ <https://www.visualstudio.com/es/vs/>

⁶ <https://docs.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>

⁷ <https://products.office.com/es-ar/microsoft-lync-2013>

⁸ https://www.google.com/intl/es_ALL/drive/

NuGet es el administrador de paquetes para .NET, gracias al cual las funcionalidades del Visual Studio pueden extenderse al añadirle plugins desde la galería NuGet. Para el desarrollo de ciertas funcionalidades del proyecto móvil estas extensiones serán de gran ayuda, los paquetes instalados más relevantes para el desarrollo del proyecto son:

3.7.6. Xamarin.Forms⁹

Xamarin.Forms es un paquete que permite crear interfaces de usuario nativas para iOS, Android y Windows Phone desde una sola base de código compartida de C #.

3.7.7. Xamarin.Controls.SignaturePad y Xamarin.Controls.SignaturePad.Forms¹⁰

Estos paquetes serán utilizados para capturar, guardar y mostrar firmas dibujadas sobre la pantalla del móvil.

3.7.9. XamForms.Controls.Calendar¹¹

Este plugin contiene un control para mostrar en pantalla un Pop up con un calendario y guardar la fecha seleccionada en el mismo.

3.7.10. Toasts.Forms.Plugin¹²

Toast.Forms.Plugin será el encargado de mostrar notificaciones al realizar determinadas acciones con la aplicación móvil, como por ejemplo añadir o eliminar un elemento.

3.7.11. Rg.Plugins.Popup¹³

Gracias a este paquete se podrán crear Pop ups personalizados¹⁴ dentro de la aplicación móvil.

3.7.12. Xam.Plugin.Media¹⁵

Este plugin permite tanto hacer fotos y videos como seleccionar este tipo de elementos multimedia desde una API multiplataforma (como es el caso de este proyecto)

⁹ <https://www.xamarin.com/forms>

¹⁰ <https://github.com/xamarin/SignaturePad>

¹¹ <https://www.nuget.org/packages/XamForms.Controls.Calendar/>

¹² <https://www.nuget.org/packages/Toasts.Forms.Plugin/>

¹³ <https://www.nuget.org/packages/Rg.Plugins.Popup/>

¹⁴ <https://thatcsharpguy.com/post/popups-avanzados-xamarin-forms/>

¹⁵ <https://www.nuget.org/packages/Xam.Plugin.Media/>

3.8. Gestión de riesgos ¹⁶

El desarrollo de un proyecto conlleva una serie de riesgos. Identificarlos, trazar un plan de acción en caso de que ocurran y realizar un seguimiento de los mismos será primordial para disminuir la probabilidad y el impacto de eventos negativos para el proyecto.

- Problemas personales o de salud
 - Probabilidad: media.
 - Impacto: media.
 - Consecuencias: retraso en las fechas planificadas para la realización de las distintas tareas del proyecto.
 - Prevención: teniendo en cuenta que dependiendo del tipo de problema se realizaría un plan de prevención distinto, no se puede determinar.
 - Plan de contingencia: intentar recuperar estas horas retrasadas, dedicando más tiempo a las tareas atrasadas con el objetivo de terminarlas en plazo.

- Disponibilidad del director y la empresa a la hora de establecer alguna reunión
 - Probabilidad: alta.
 - Impacto: medio.
 - Consecuencias: retraso en las fechas planificadas para la realización de las distintas tareas del proyecto. En el caso de la empresa, ya que antes de dar por finalizada una iteración la empresa tendrá que mostrar su conformidad con el trabajo realizado. En el caso del director, porque su orientación se considera fundamental para sacar adelante el proyecto.
 - Prevención: compartir con el director y la empresa la planificación del proyecto en cuanto se concrete y establecer las reuniones con varios días de antelación.
 - Plan de contingencia: Intentar coordinarse con el director o la empresa para realizar la reunión atrasada lo antes posible.

¹⁶ <https://uacm123.weebly.com/8-gestioacuten-de-los-riesgos-del-proyecto.html>

- Problemas a la hora de instalar el software necesario
 - Probabilidad: media.
 - Impacto: medio.
 - Consecuencias: retraso en las fechas planificadas para la realización de las distintas tareas del proyecto.
 - Prevención: buscar y leer los tutoriales y manuales necesarios para la correcta instalación de todas las herramientas a utilizar.
 - Plan de contingencia: revisar la instalación para averiguar el posible fallo.

- Problemas con el funcionamiento del software
 - Probabilidad: media.
 - Impacto: medio.
 - Consecuencias: retraso en las fechas planificadas para la realización de las distintas tareas del proyecto.
 - Prevención: procurar tener el software necesario para la realización del proyecto siempre actualizado.
 - Plan de contingencia: intentar recuperar estas horas retrasadas, dedicando más tiempo a las tareas atrasadas con el objetivo de terminarlas en plazo.

- Pérdida de información almacenada localmente
 - Probabilidad: baja.
 - Impacto: alto.
 - Consecuencias: ya que el desarrollo del proyecto se almacena en distintos servidores de la empresa, la única pérdida de información sería la relacionada con la documentación del proyecto. Si esto sucediera, se perdería el trabajo relacionado con la memoria del proyecto.
 - Prevención: hacer copias de seguridad diarias y tener toda la documentación del proyecto sincronizada en Google Drive constantemente.
 - Plan de contingencia: Recuperar la copia de seguridad más reciente, ya sea mediante la obtención desde Drive o utilizando herramientas específicas para la recuperación de archivos.

- Planificación incorrecta debido a estimaciones no realistas.
 - Probabilidad: media.
 - Impacto: alto.
 - Consecuencias: retraso en las fechas planificadas para la realización de las distintas tareas del proyecto, incluso, retraso en la fecha de entrega del proyecto.
 - Prevención: realizar una planificación cuyos hitos dejen un margen de tiempo generoso desde la finalización del proyecto hasta la entrega oficial del mismo.
 - Plan de contingencia: si al realizar la gestión del proyecto se detectara un error o retraso en alguna parte de la planificación, se procederá a realizar una nueva planificación de manera inmediata, intentando no comprometer la fecha de entrega del proyecto.

- Variación del alcance del proyecto
 - Probabilidad: baja
 - Impacto: medio.
 - Consecuencias: retraso en las fechas planificadas para la realización de las distintas tareas del proyecto, incluso, retraso en la fecha de entrega del proyecto.
 - Prevención: aplicar buenas prácticas del desarrollo del software, de manera que si hubiera que realizar cambios en el alcance del proyecto, afecten lo mínimo posible a lo desarrollado hasta el momento.
 - Plan de contingencia: volver a planificar el desarrollo del proyecto, intentando variar lo mínimo posible los hitos del mismo.

4

Análisis de requisitos

La captura de requisitos es el primer paso para realizar una buena aplicación, ya que permitirá analizar las necesidades del cliente, dando una visión global del alcance y disposición del proyecto final.

En este capítulo se describe el análisis de requisitos de las dos aplicaciones. Se mostrarán los requisitos funcionales mediante la identificación de los casos de uso, explicando la secuencia de eventos de cada caso mediante flujos de eventos. Para ilustrar los requisitos de datos se utilizarán modelos de dominio. Además, se describirán las estructuras de las bases de datos de la aplicación mediante la realización de modelos relacionales de datos. Por último, se detallarán los requisitos no funcionales que describirán las características de funcionamiento del sistema.

4.1. Requisitos funcionales

4.1.1. Diagrama de casos de uso¹⁷: Control de accesos

Esta funcionalidad está centrada en el control de accesos de personal ajeno a la empresa a las instalaciones de la misma. La decisión sobre si a una persona externa a la empresa se le acepta o deniega el paso, radica en que la documentación asociada con los requisitos que debe cumplir tanto este trabajador como la empresa de la que proviene esté actualizada.

A continuación, se detalla el modelo de casos de uso de esta funcionalidad utilizando las interfaces creadas para la aplicación móvil para ilustrar las distintas funcionalidades.



Ilustración 2: Diagrama de casos de uso, Control de accesos

Nota: desde todas las listas de requisitos (requisitos del trabajador, requisitos de la empresa y bloqueantes), se puede ver el requisito, editar el requisito y descargar la documentación del mismo.

¹⁷ <https://msdn.microsoft.com/es-es/library/dd409432.aspx>

Buscar trabajador: El control de accesos comenzará con la búsqueda del trabajador sobre el que se quiere averiguar si cumple los requisitos requeridos para permitirle el acceso a las instalaciones de la empresa.

Flujo de eventos:

1. El empleado tendrá la opción de rellenar los campos DNI, nombre y apellidos, identificador y organización relativos al trabajador que quiere encontrar. Para evitar equivocaciones innecesarias, el campo de organización será un buscador de organizaciones vinculadas con la empresa en cuestión, se detallará este control posteriormente.
2. Al pulsar sobre “Buscar” aparecerá la lista de los trabajadores que concuerdan con los filtros previamente completados. Si se quiere buscar todos los empleados, bastará con dejar los filtros en blanco.
3. La lista de los trabajadores encontrados detalla el DNI, nombre, organización e identificador del trabajador. Adicionalmente, si el trabajador tiene únicamente un pedido activo y con fechas de contrata en regla o no tiene ningún pedido en absoluto, se muestra el estado de los requisitos del trabajador en cuestión. Para seleccionar el trabajador deseado bastará con pulsar sobre él.

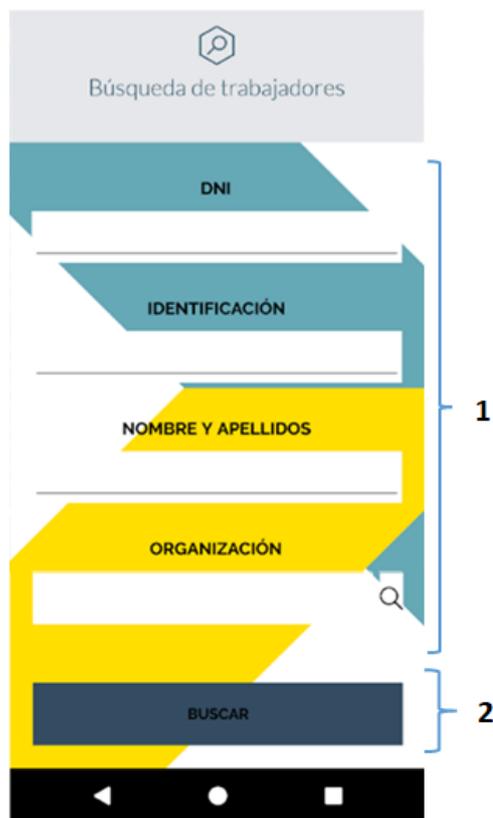


Ilustración 3: Interfaz de búsqueda de trabajadores

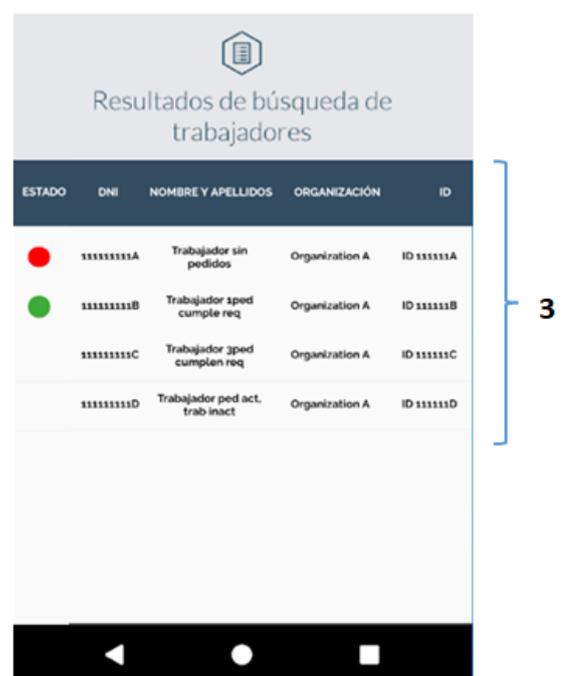


Ilustración 4: Interfaz de resultados de búsqueda de trabajadores

Ver lista de pedidos: Al pulsar sobre un trabajador desde el buscador de trabajadores, si el trabajador seleccionado tiene varios pedidos en activo y con fechas de contrata en regla, o si no tiene ningún pedido en activo pero sí alguno desactivado o con fechas de contrata caducadas, se visualizará la lista de todos los pedidos del trabajador, tanto activos como inactivos y con fechas de contrata al día o caducadas.

Flujo de eventos:

1. La lista de los pedidos del trabajador detalla el estado de los requisitos del trabajador en cuestión relacionado con dicho pedido, el número, descripción y responsable del pedido, si el trabajador está activo, las fechas de inicio y fin de contrata y si el pedido está o no validado. Para seleccionar un pedido bastará con pulsar sobre él, con el pedido seleccionado podemos realizar dos acciones.
2. Ver la ficha del trabajador con respecto al pedido seleccionado pulsando sobre el botón “Ficha del trabajador”.
3. Ver el pedido seleccionado pulsando sobre el botón “Detalle del pedido”.

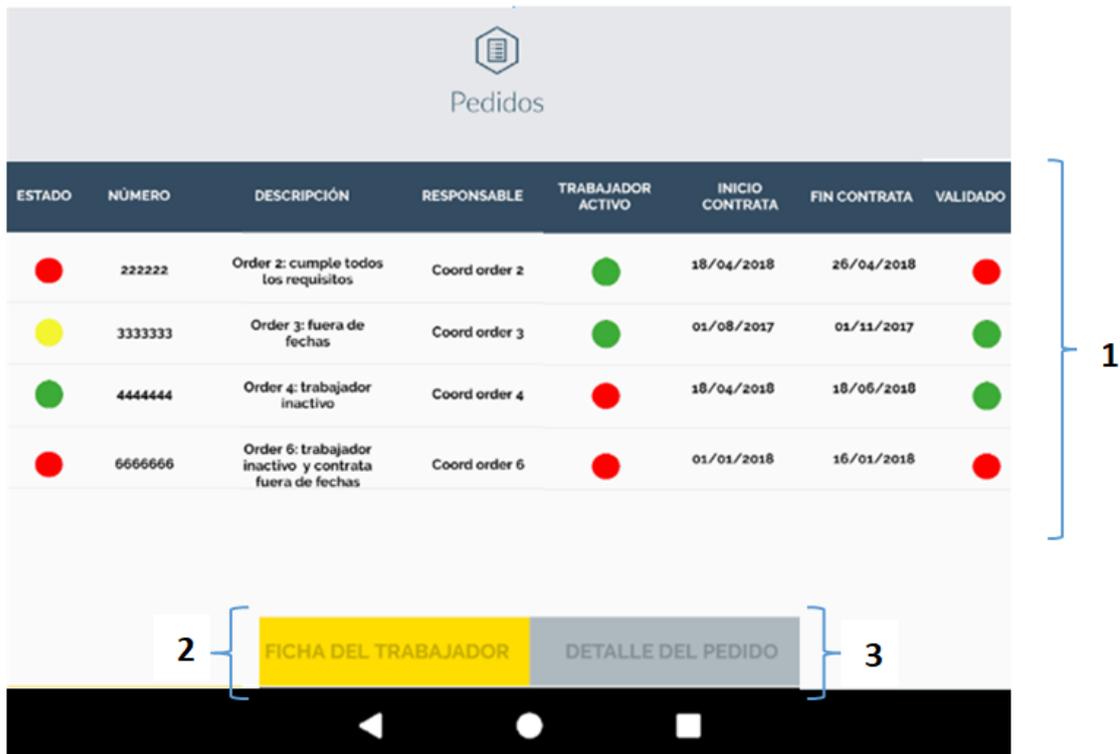


Ilustración 5: Interfaz lista de pedidos

Ver pedido: Desde la lista de pedidos del trabajador, al seleccionar un pedido y pulsar sobre el botón “Detalle del pedido”, aparecerá una nueva página mostrando toda la información disponible a cerca del pedido seleccionado.

Flujo de eventos:

1. Se detalla el empresario principal, el departamento, el responsable de trabajos, el estado, la validez, la descripción y las fechas de inicio y fin del pedido, además del estado del trabajador en relación con el pedido en cuestión. Únicamente serán editables los campos de estado de trabajador y fecha de inicio y finalización del pedido.

Editar pedido: El empleado puede editar el estado del trabajador respecto al pedido (activo/inactivo) y su fecha de contrata.

Flujo de eventos:

2. Tras la edición de los campos previamente mencionados, se podrán guardar los cambios pulsando sobre el botón amarillo situado en la parte inferior izquierda de la pantalla.

Campos editables

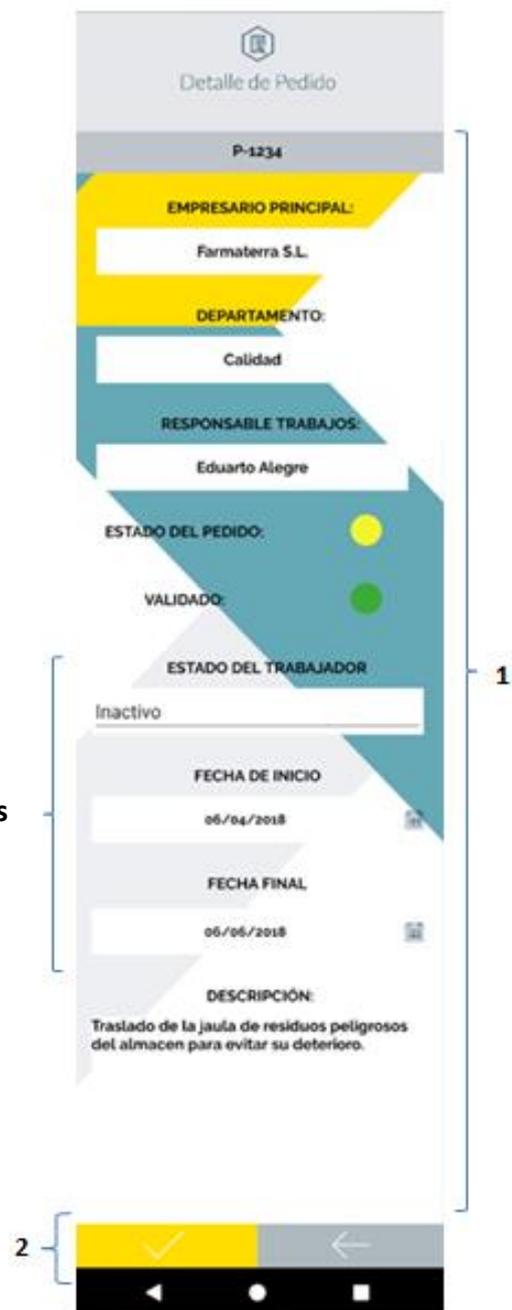


Ilustración 6: Interfaz detalle de un pedido

Ver ficha del trabajador: Al pulsar sobre un trabajador desde el buscador de trabajadores, si el trabajador seleccionado tiene únicamente un pedido activo o si no tienen ningún pedido en absoluto, se visualizará su ficha de trabajador. Adicionalmente, si desde la lista de pedidos de un trabajador se selecciona uno y se pulsa sobre el botón “Ficha del trabajador” también se visualizará esta ficha.

Flujo de eventos:

1. En esta ficha se detalla la foto, nombre y apellidos, DNI e identificación del trabajador, así como empresa y el código del pedido con el que se le ha asociado. Adicionalmente, la bola de color (definiremos así al testigo de estado) al lado del nombre del trabajador nos indica el estado de sus requisitos, de la misma manera que la bola de color al lado del nombre de la organización indica el estado de los requisitos de dicha organización.
2. Si el periodo de contratación del pedido ha expirado, en la ficha del trabajador aparecerá un aviso, y dará la opción de actualizar las fechas para que la contrata vuelva a estar activa. Para modificar la fecha de contratación, habrá que modificar las fechas de contrata y pulsar sobre el botón “Actualizar”.
3. De la misma manera, si el trabajador se encuentra inactivo, aparecerá un aviso informando de ello en la ficha del trabajador. Para activar al trabajador bastará con pulsar sobre el botón “Activar”.
4. El empleado podrá acceder a la lista de requisitos del trabajador pulsando sobre la bola de estado pertinente o sobre el botón “Requisitos trabajador”.
5. De igual forma, para acceder a la lista de requisitos de la organización, el empleado tendrá que pulsar sobre la bola de estado pertinente o sobre el botón “Requisitos empresa”.
6. Si el empleado quisiera ver los bloqueantes del trabajador y de la empresa del mismo, deberá pulsar sobre el botón “Bloqueantes”. Los bloqueantes son requisitos cuyo estado es no validado y con acceso condicionado.

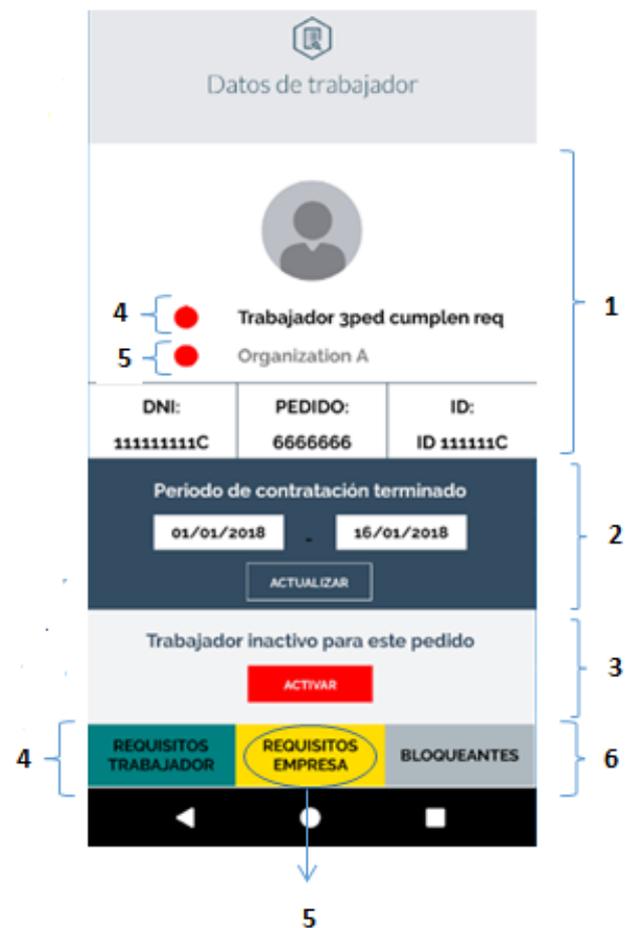


Ilustración 7: Interfaz ficha de trabajador

Ver lista de requisitos del trabajador: Desde el la ficha del trabajador, al pulsar sobre la bola de estado del trabajador o sobre el botón “Requisitos trabajador”, aparecerá una nueva página listando los requisitos del trabajador en cuestión.

Flujo de eventos:

1. La lista de los requisitos del trabajador detalla el estado, el nombre, la fecha de caducidad y el acceso del requisito. Además, cada requisito posee un botón para descargar la documentación en caso de que la tenga. Para ver el detalle del requisito deseado bastará con pulsar sobre él.
2. Si el requisito tiene documentación adjunta, el botón de “Abrir” estará habilitado. Al pulsar sobre él se mostrará la documentación correspondiente.

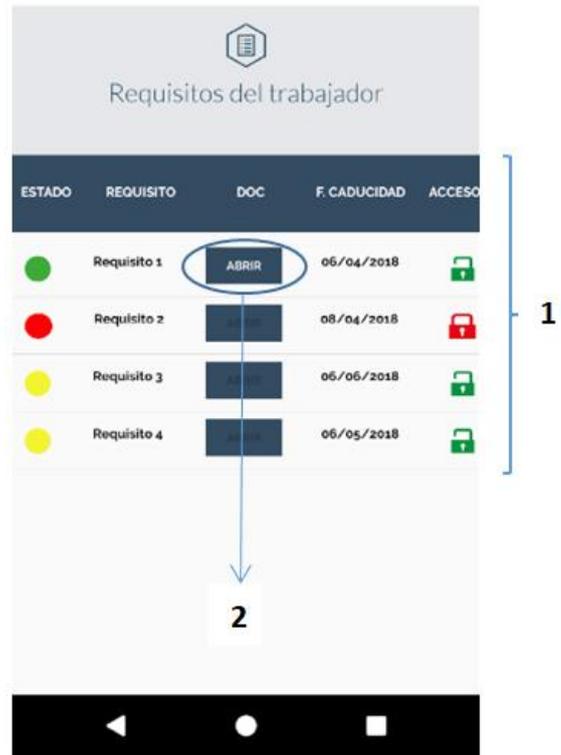


Ilustración 8: Interfaz lista de requisitos del trabajador

Ver lista de requisitos de la empresa: Desde la ficha del trabajador, al pulsar sobre la bola de estado de la empresa o sobre el botón “Requisitos empresa”, aparecerá una nueva página listando los requisitos de la empresa del trabajador en cuestión.

Flujo de eventos:

1. La lista de los requisitos de la empresa detalla el estado, el nombre, la fecha de caducidad y el acceso del requisito. Además, cada requisito posee un botón para descargar la documentación en caso de que la tenga. Para ver el detalle del requisito deseado bastará con pulsar sobre él.
2. Si el requisito tiene documentación adjunta, el botón de “Abrir” estará habilitado. Al pulsar sobre él se mostrará la documentación correspondiente.



Ilustración 9: Interfaz lista de requisitos de la empresa

Ver lista de bloqueantes: Desde el la ficha del trabajador, al pulsar el botón “Bloqueantes”, aparecerá una nueva página listando los requisitos cuyo estado es no validado y con acceso condicionado tanto del trabajador como de su empresa.

Flujo de eventos:

1. La página se divide en dos listas de bloqueantes, una para contener los bloqueantes del trabajador y la otra para abarcar los bloqueantes de la empresa del trabajador en cuestión. Ambas listas detallan el estado, el nombre, la fecha de caducidad y el acceso del requisito concreto. Además, cada requisito posee un botón para descargar la documentación en caso de que la tenga. Para ver el detalle del requisito deseado bastará con pulsar sobre él
2. Si el requisito tiene documentación adjunta, el botón de “Abrir” estará habilitado. Al pulsar sobre él se mostrará la documentación correspondiente.

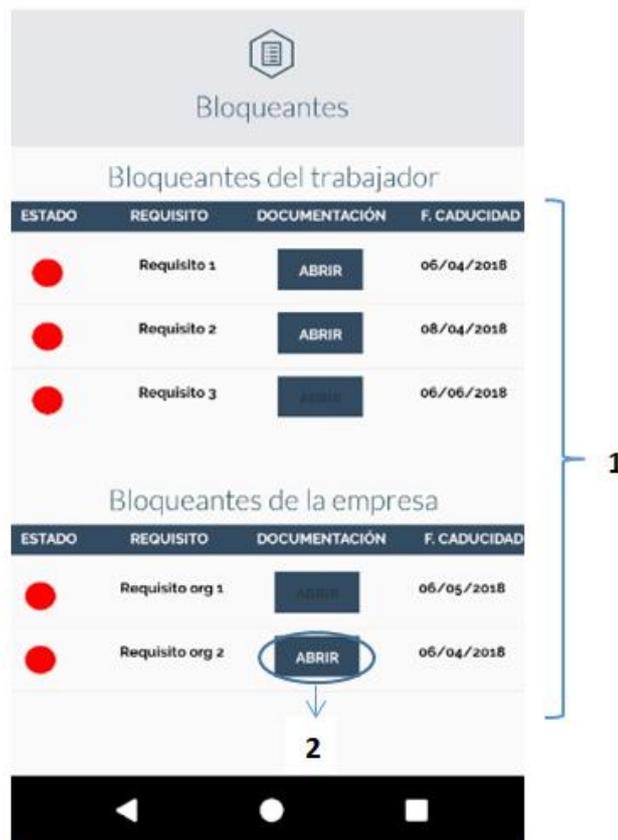


Ilustración 10: Interfaz lista de bloqueantes

Ver requisito: Desde las páginas de requisitos de trabajador, de requisitos de la empresa y de bloqueantes, al pulsar sobre uno de los requisitos, aparecerá una nueva página de detalle del requisito seleccionado.

Flujo de eventos:

1. Se detalla el nombre, el estado, la fecha de expedición, el tipo de acceso, los comentarios previos y la permanencia del pedido. Además, si el pedido no es permanente, se precisará la validez y la fecha de vencimiento del mismo. Todos estos campos son editables. También se puede consultar la documentación adjunta al requisito al pulsar sobre el botón “Ver”.

Editar requisito: El empleado puede cambiar todos los campos editables del detalle del pedido, añadir documentación y guardar los cambios.

Flujo de eventos:

2. Para añadir documentación al requisito hay que pulsar sobre el botón “Nuevo”, se abrirá un explorador de archivos para buscar el documento deseado dentro de las carpetas del móvil
3. Tras la edición de los campos previamente mencionados, se podrán guardar los cambios pulsando sobre el botón amarillo situado en la parte inferior izquierda de la pantalla

Descargar la documentación del requisito: Desde las páginas de requisitos de trabajador, de requisitos de la empresa, de bloqueantes y de detalle del requisito, al pulsar sobre el botón de “Abrir” y “Ver” respectivamente, se mostrará la documentación correspondiente al requisito deseado.

Flujo de eventos:

4. Al pulsar sobre dichos botones, se abre una ventana de vista previa del documento en cuestión.

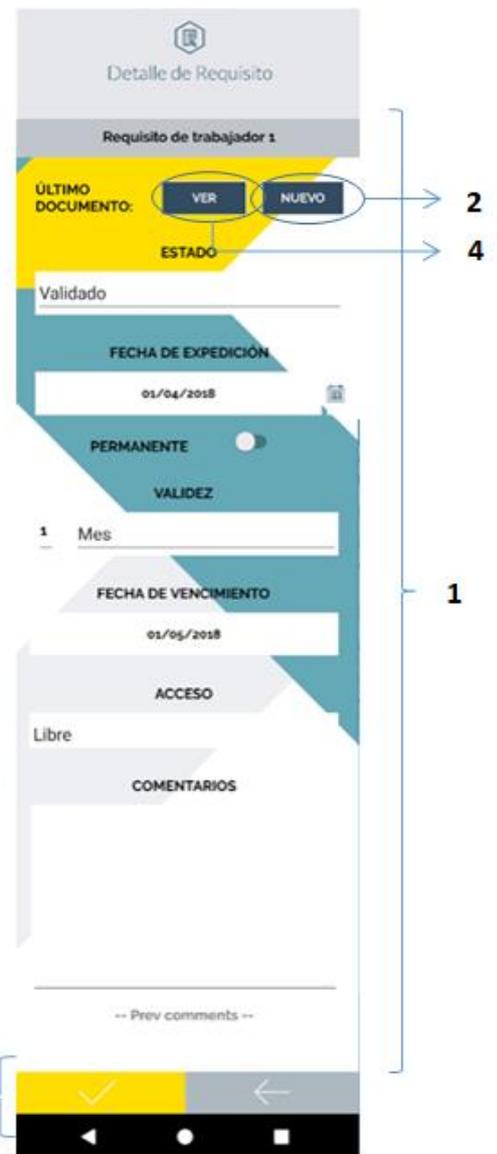


Ilustración 11: Interfaz detalle de requisito

4.1.2. Diagrama de casos de uso: Entrega de EPIs

Esta funcionalidad está centrada en la gestión del reparto de equipos de protección individual. La necesidad de utilizar estos equipos está regulada por la ley y, con el fin de reducir los riesgos laborales es importante definir qué equipos de protección individual son necesarios para garantizar la seguridad en puestos donde la salud del trabajador podría encontrarse en riesgo, formar a los trabajadores para que utilicen adecuadamente estos equipos y, por último gestionar su entrega.

Primero, detallaremos el modelo de casos de uso de esta funcionalidad utilizando las interfaces creadas para la aplicación móvil. Después, mostraremos esta misma información a partir de las interfaces de la aplicación web, de manera que queden reflejadas las dos facetas de este proyecto.



Ilustración 12: Diagrama de casos de uso, Entrega de EPIs

4.1.2.1. Aplicación móvil.

Buscar EPIs: con la finalidad de encontrar el Equipo de Protección Individual deseado, el usuario podrá filtrar los EPIs mediante el nombre, código y/o tipo del equipo, el puesto para el que es necesario y/o el centro de trabajo en el que se utiliza.

Flujo de eventos:

1. El empleado tendrá la opción de rellenar los campos tipo de EPI, código, nombre, puesto, activo y centro de trabajo relativos al EPI que quiere encontrar.
2. Al pulsar sobre “Buscar” aparecerá la lista de los EPIs que concuerdan con los filtros previamente completados. Si se quiere buscar todos los EPIs, bastará con dejar los filtros en blanco.
3. La lista de los equipos encontrados detalla el tipo, código, nombre y centro de trabajo del EPI. Para seleccionar el equipo de protección deseado bastará con pulsar sobre él.



Ilustración 13: Interfaz búsqueda de EPIs

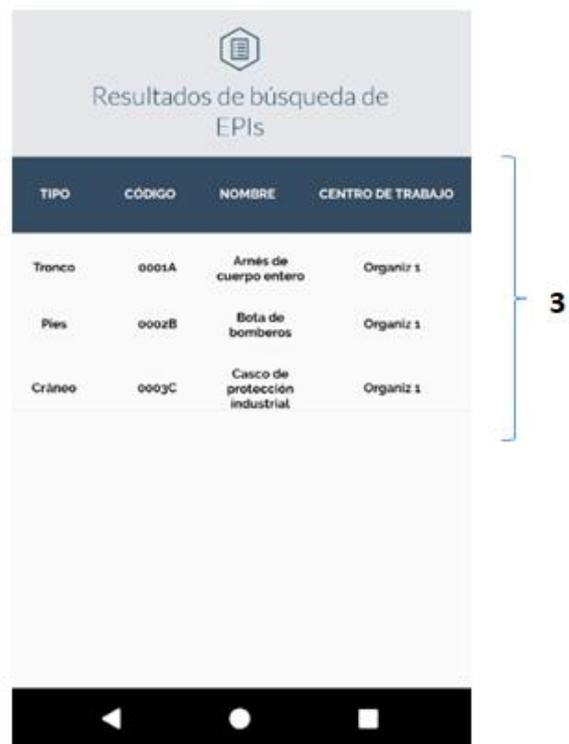


Ilustración 14: Interfaz resultados de la búsqueda de EPIs

Ver lista entregas EPI: Al pulsar sobre un EPI desde el buscador de equipos de protección, se visualizará la lista de las entregas realizadas de ese tipo de equipo relacionándolas con el trabajador al que fueron entregadas y ordenadas por fecha de entrega.

Flujo de eventos:

1. La lista de las entregas detalla la fecha de entrega del equipo, el DNI y nombre y apellidos del trabajador a quien fue entregado, si el trabajador ha recibido el equipo y, por último, el estado de la entrega. Si la fecha de reposición del equipo es anterior a la fecha actual, el estado de la entrega será rojo; si la fecha actual está comprendida entre la fecha de entrega y la fecha de reposición del equipo, el estado de la entrega será verde. Para ver el detalle de la entrega deseada bastará con pulsar sobre ella. Además, cada entrega posee un botón con el icono de una papelera para eliminarla.
2. Se podrán filtrar las entregas a visualizar, para ello habrá que pulsar sobre el botón “Aplicar filtros”.
3. Si el empleado quiere agregar una nueva entrega del EPI en cuestión, tendrá que pulsar sobre el botón “Agregar entrega”.

Eliminar entrega: Al pulsar sobre el botón con el icono de una papelera desde la lista de entregas de un determinado EPI, se procederá a su eliminación

Flujo de eventos:

1. Pulsar sobre el icono de papelera situado en la fila de la entrega que se quiere eliminar.

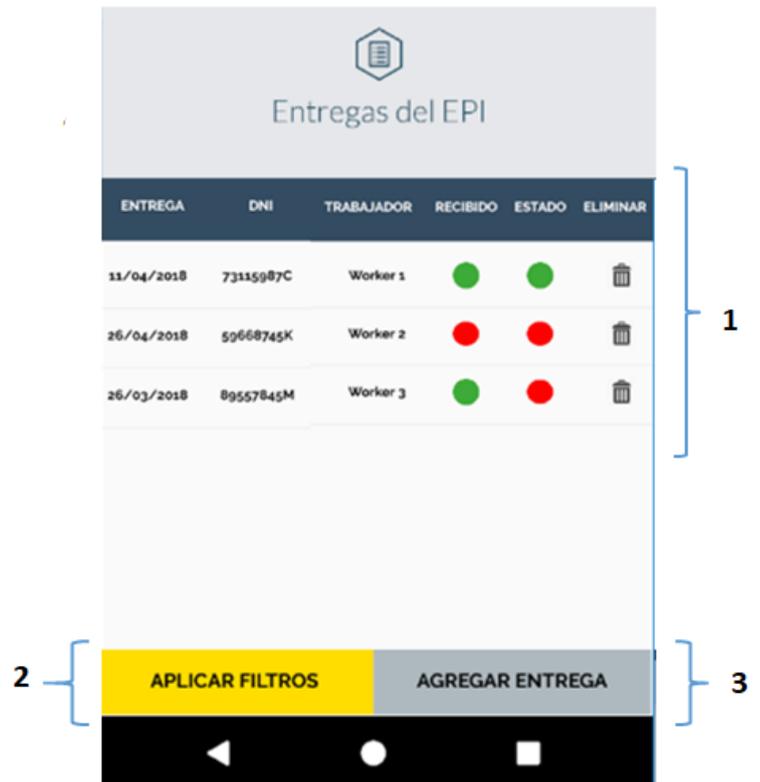


Ilustración 15: Interfaz lista de entregas de un EPI

Filtrar entregas: Al pulsar sobre el botón “Filtrar entregas” desde la lista de entregas de un determinado EPI, aparecerá un PopUp para rellenar los campos mediante los que se pueden filtrar las entregas.

Flujo de eventos:

1. El empleado tendrá la opción de rellenar los campos estado, recibido, trabajador y organización relativos a las entregas que quiere visualizar. Para evitar equivocaciones innecesarias, tanto el campo de trabajador como el de organización serán buscadores vinculados con la empresa del empleado que está realizando la búsqueda, estos controles se detallarán posteriormente.
2. Al pulsar sobre “Buscar” aparecerá la lista de las entregas que concuerdan con los filtros previamente completados.



Ilustración 16: interfaz Pop Up filtrar entregas de un EPI

Ver entrega: Al pulsar sobre una entrega del EPI en cuestión, aparecerá una nueva página de detalle de la entrega seleccionada.

Flujo de eventos:

1. Se detalla el nombre del equipo, el trabajador, la fecha en que se produjo la entrega, la fecha de reposición prevista y el número de unidades expedidas. Si la entrega ha sido firmada por el trabajador, aparecerá la imagen de la firma y la fecha en la que se produjo. Si el trabajador no ha firmado todavía la entrega, todos estos campos serán editables, en caso contrario permanecerán deshabilitados para su edición.

Editar entrega: Si el trabajador al que se le ha entregado el equipo aún no ha firmado su entrega, los campos de trabajador, fecha de entrega, fecha de reposición, unidades entregadas y firma del trabajador podrán alterarse.

Flujo de eventos:

2. Tras la edición de los campos previamente mencionados, se podrán guardar los cambios pulsando sobre el botón amarillo situado en la parte inferior izquierda de la pantalla.



Ilustración 17: Interfaz detalle entrega de un EPI

Agregar entrega: Al pulsar sobre el botón “Agregar entrega” desde la lista de entregas de un determinado EPI, aparecerá un nuevo formulario para completar con los datos de la nueva entrega.

Flujo de eventos:

1. El empleado podrá rellenar los campos de trabajador, fecha de entrega, fecha de reposición y unidades entregadas, necesitando que el trabajador al que le está entregando el equipo firme en el campo de firma del trabajador para que este entrega se dé como recibida.
2. Tras la edición de los campos previamente mencionados, se podrán guardar los cambios pulsando sobre el botón amarillo situado en la parte inferior izquierda de la pantalla.

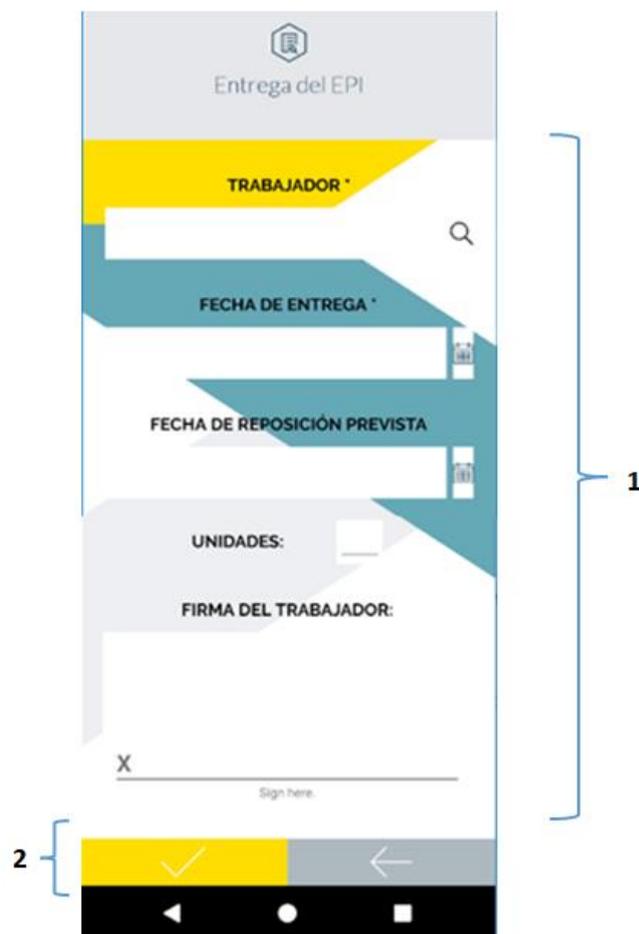


Ilustración 18: Interfaz añadir entrega de un EPI

Buscar organización: en determinados formulario, el empleado podrá filtrar por organización. Para ayudar en esta tarea, se ha desarrollado un control de búsqueda de organizaciones.

Flujo de eventos:

1. El empleado tendrá la opción de rellenar los campos número de organización, NIF, nombre, clasificación y activo relativos a la empresa que quiere encontrar.
2. Al pulsar sobre “Buscar” aparecerá la lista de las empresas que concuerdan con los filtros previamente completados. Si se quiere buscar todas las empresas, bastará con dejar los filtros en blanco.
3. La lista de las empresas encontradas detalla el NIF, nombre y clasificación de la empresa. Para seleccionar la empresa deseado bastará con pulsar sobre ella.



Ilustración 19: Interfaces Pop Up de búsqueda de una organización

Buscar trabajador: en determinados formularios, el empleado podrá filtrar por trabajador. Para ayudar en esta tarea, se ha desarrollado un control de búsqueda de trabajadores.

Flujo de eventos:

1. El empleado tendrá la opción de rellenar los campos DNI, nombre y organización relativos al trabajador que quiere encontrar.
2. Al pulsar sobre “Buscar” aparecerá la lista de los trabajadores que concuerdan con los filtros previamente completados. Si se quiere buscar todos los trabajadores, bastará con dejar los filtros en blanco.
3. La lista de los trabajadores encontrados detalla DNI, nombre y organización de los distintos trabajadores. Para seleccionar el trabajador deseado bastará con pulsar sobre él.



Ilustración 20: Interfaces Pop Up búsqueda de un trabajador

4.1.2.2. Aplicación web.

Buscar EPIs: con la finalidad de encontrar el Equipo de Protección Individual deseado, el usuario podrá buscar productos mediante un campo de texto autocompletable o mediante un buscador de productos.

Flujo de eventos:

A. Campo de texto autocompletable:

1. El empleado puede teclear en el campo de texto autocompletable el nombre del equipo de protección deseado. Si el producto seleccionado es de tipo “Protección Individual”, se mostrará una lista con las entregas del mismo.

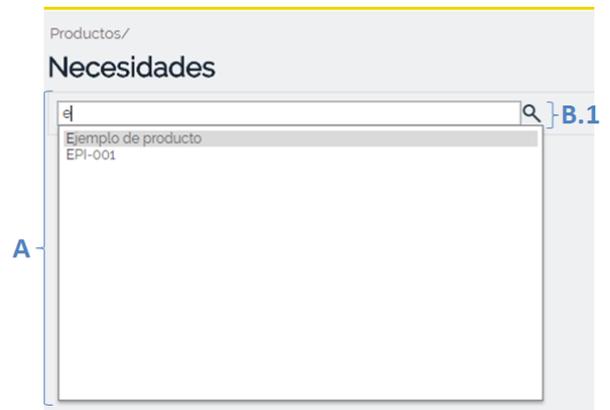


Ilustración 21: Control campo de texto autocompletable para la búsqueda de productos

B. Buscador de productos:

1. Alternativamente, el empleado puede realizar una búsqueda del pedido mediante un buscador. Para ello, tendrá que pulsar sobre el icono de la lupa.
2. El empleado tendrá la opción de rellenar los campos tipo de EPI, código, nombre, lugar, activo y centro de trabajo relativos al EPI que quiere encontrar.
3. Al pulsar sobre el botón “Buscar” aparecerá una lista de los productos que cumplan los filtros previamente rellenados.
4. Al seleccionar el producto deseado y hacer clic sobre “Aceptar”, si nuestra elección es de tipo “Protección Individual”, se mostrará una lista con las entregas del mismo.

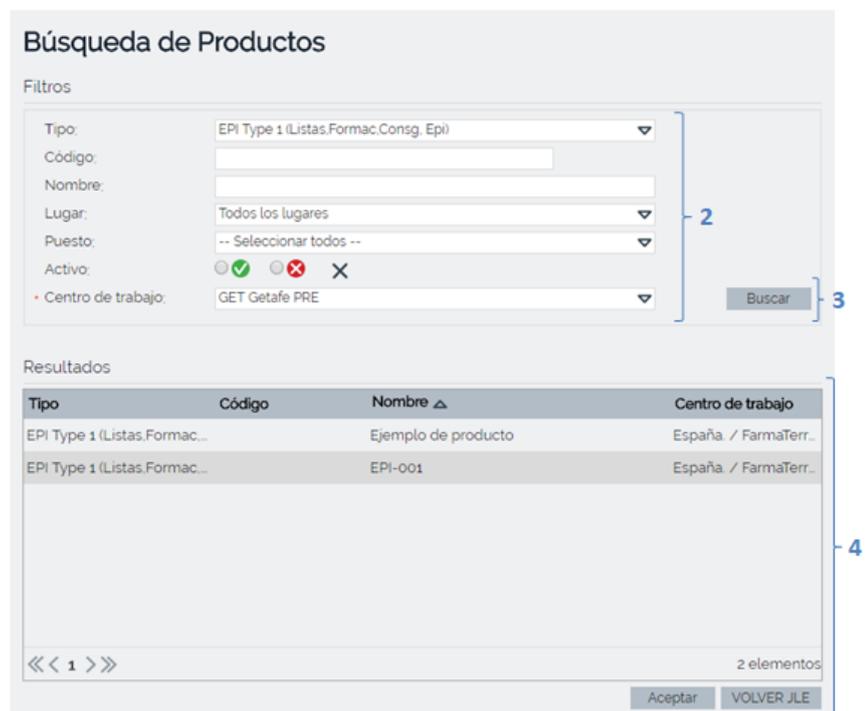


Ilustración 22: Control personalizado para la búsqueda de productos

Ver lista entregas EPI: Al seleccionar el producto deseado, si este es de tipo “Protección Individual”, se mostrará la lista de las entregas realizadas de ese equipo relacionándolas con el trabajador al que fueron entregadas y ordenadas por fecha de entrega.

Flujo de eventos:

1. La lista de los entregas detalla la fecha de entrega del equipo, la organización, el DNI y nombre y apellidos del trabajador a quien fue entregado, si el trabajador ha recibido el equipo, el estado de la entrega y, por último si tiene documentación adjunta. Si la fecha de reposición del equipo es anterior a la fecha actual, el estado de la entrega será rojo; si la fecha actual está comprendida entre la fecha de entrega y la fecha de reposición del equipo, el estado de la entrega será verde.
2. Para editar una entrega, habrá que seleccionar la entrega deseada dentro de la lista y pulsar sobre el botón “Ver”.
3. Si el empleado quiere agregar una nueva entrega al EPI en cuestión, tendrá que pulsar sobre el botón “Agregar”.
4. Para eliminar una entrega, habrá que seleccionar la entrega deseada dentro de la lista y pulsar sobre el botón “Eliminar”.

Fecha entrega	Organización	DNI	Trabajador	Recibidas	Estado	Doc. entregada
09/05/2018	ORG_NAME5	ania	Ania Berazaluze	✓	✓	✗
18/05/2018	ORG_NAME5	44168161L	Jaione Larrañaga Esnaola	✗	✓	✗

«< 1 >»

2 { Ver Agregar Eliminar } 4

3 2 elementos

Ilustración 23: Lista de entregas de un EPI en la web

Ver entrega: Al seleccionar una entrega de la lista y pulsar sobre el botón “Ver”, aparecerá una nueva página de detalle de la entrega seleccionada.

Flujo de eventos:

1. Seleccionar la entrega deseada de la lista.
2. Pulsar sobre el botón “Ver”.
 - Para ilustrar esta funcionalidad, se puede observar la ilustración que acompaña a *Ver la lista de entregas de EPI*: el punto dos corresponde con esta manera de eliminar entregas.

Editar entrega: Desde la página de detalle de la entrega seleccionada, se podrán modificar las características del elemento en cuestión.

Flujo de eventos:

1. Seleccionar la entrega deseada de la lista.
2. Pulsar sobre el botón “Ver”. Se nos abrirá la página de detalle de entrega de EPI con los datos de la entrega seleccionada.
3. Editar los campos Trabajador, fecha, fecha de reposición, unidades y/o he recibido
4. Pulsar sobre el botón “Aceptar”.
5. Para eliminar una entrega, habrá que pulsar sobre el botón “Eliminar”.

Productos/
Entrega de EPIs
Producto: EPI-001
Trabajador: Ania Berazaluce
Fecha: 09/05/2018

Detalle | Inf. Adicional

• Trabajador: Ania Berazaluce
• Fecha: 09/05/2018
Fecha de reposición prevista: 31/05/2018
Unidades: 1
He recibido:

Eliminar Aceptar VOLVER JLE

Ilustración 24: Interfaz detalle de una entrega de un EPI en la web

Eliminar entrega (opción solo válida para determinados usuarios): Al seleccionar una entrega de la lista y pulsar sobre el botón “Eliminar”, se procederá al borrado de la entrega seleccionada. Además, se podrá eliminar la entrega desde la página de detalle de la misma

Flujo de eventos:

- A. Eliminación desde la lista de entregas:
1. Seleccionar la entrega deseada de la lista.
 2. Pulsar sobre el botón “Eliminar”.
 - Para ilustrar esta funcionalidad, se puede observar la ilustración que acompaña a *Ver la lista de entregas de EPI*: el punto cuatro corresponde con esta manera de eliminar entregas.

- B. Eliminación desde la página de detalle de la entrega:
1. Seleccionar la entrega deseada de la lista.
 2. Pulsar sobre el botón “Ver”.
 3. Desde la nueva página, pulsar sobre el botón “Eliminar”.
 - Para ilustrar esta funcionalidad, se puede observar la ilustración que acompaña a *Editar entrega*: el punto cinco corresponde con esta manera de eliminar entregas.

Agregar entrega: Desde la página de entregas del EPI seleccionado, se podrá añadir una nueva entrega al EPI en cuestión.

Flujo de eventos:

1. Pulsar sobre el botón “Agregar”. Se nos abrirá la página de detalle de entrega de EPI con los campos en blanco.
2. Rellenar obligatoriamente los campos de trabajador y fecha, y optativamente los campos de fecha de reposición, unidades y/o he recibido.
3. Pulsar sobre el botón “Aceptar”.

The screenshot shows a web interface for 'Entrega de EPIs'. At the top, it says 'Productos/' and 'Entrega de EPIs'. Below that, 'Producto' is listed as 'EPI-001'. The main heading is 'Nueva entrega de EPI'. There is a 'Detalle' tab. The form contains the following fields:

- 'Trabajador:' with a search box 'Buscar por Trabajador' and a magnifying glass icon.
- 'Fecha:' with a date picker icon.
- 'Fecha de reposición prevista:' with a date picker icon.
- 'Unidades:' with a text input field.
- 'He recibido:' with a checkbox.

At the bottom right, there are three buttons: 'Eliminar', 'Aceptar', and 'VOLVER JLE'. The 'Aceptar' button is highlighted with a blue box and labeled with the number '3'. A blue bracket labeled '2' spans the 'Trabajador' and 'Fecha' fields.

Ilustración 25: Interfaz detalle agregar entrega de un EPI en la web

Filtrar entregas: Al seleccionar el producto deseado, si este es de tipo “Protección Individual”, además de mostrar la lista de las entregas realizadas de ese equipo, se mostrarán un conjuntos de filtros para acotar estas entregas.

Flujo de eventos:

1. El empleado tendrá la opción de rellenar los campos estado, recibido, trabajador y organización relativos a las entregas que quiere visualizar. Tanto el campo de trabajador como el de organización serán buscadores vinculados con la empresa del empleado que está realizando la búsqueda.
2. Al pulsar sobre el botón “Buscar” se actualizará la lista de las entregas para que concuerden con los filtros previamente completados.

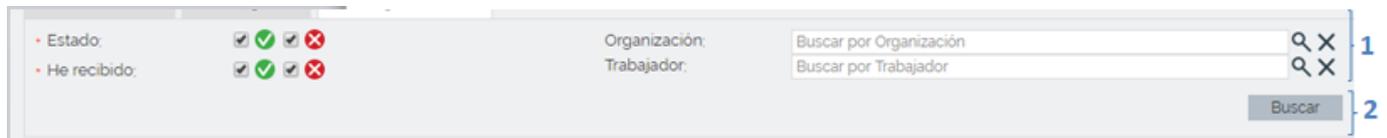


Ilustración 26: Interfaz filtros de entregas de un EPI en la web

Así pues, la página principal de entregas de un determinado EPI tendrá el siguiente aspecto:

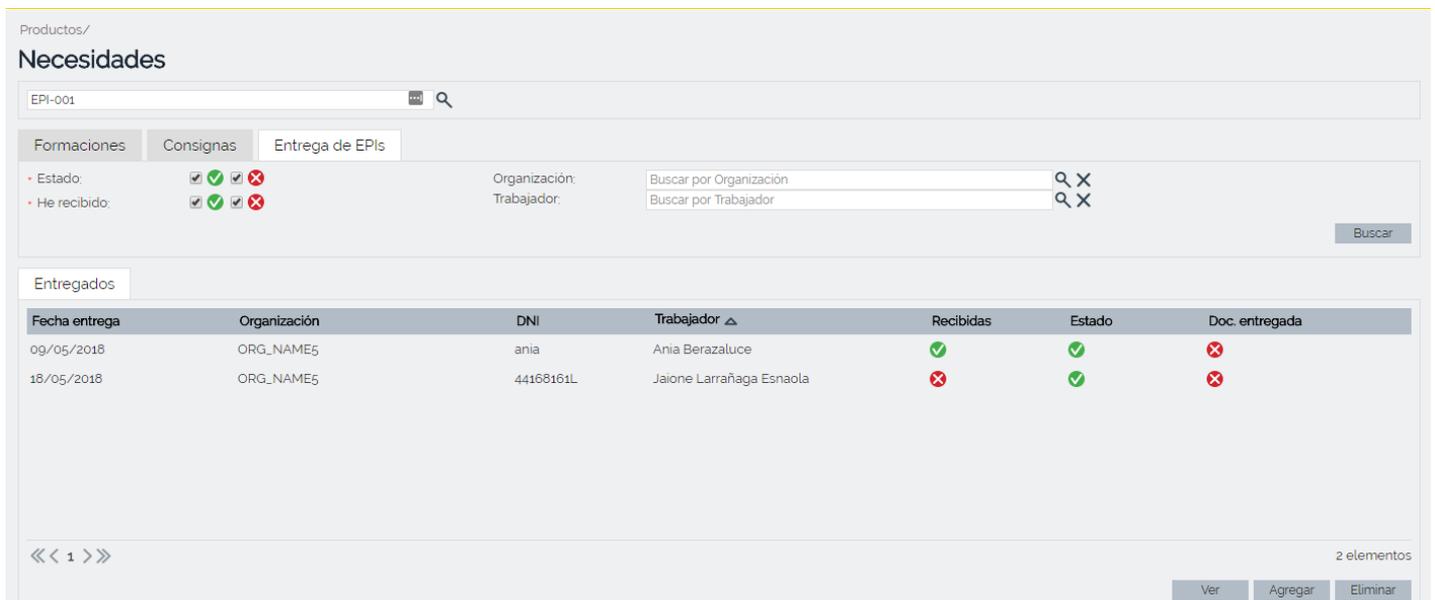


Ilustración 27: Interfaz la página principal de entregas de un determinado EPI en la web

4.2. Requisitos de datos

En el modelo de dominio¹⁸ se muestran los conceptos y relaciones asociados a la aplicación, se reflejan los datos que van a ser almacenados, así como su relación entre ellos. A continuación, se describirán los modelos de dominio de las dos funcionalidades a desarrollar.

4.2.1. Modelo de dominio: Control de accesos

En el modelo de dominio de la funcionalidad de control de accesos podemos encontrar las siguientes entidades con sus correspondientes relaciones:

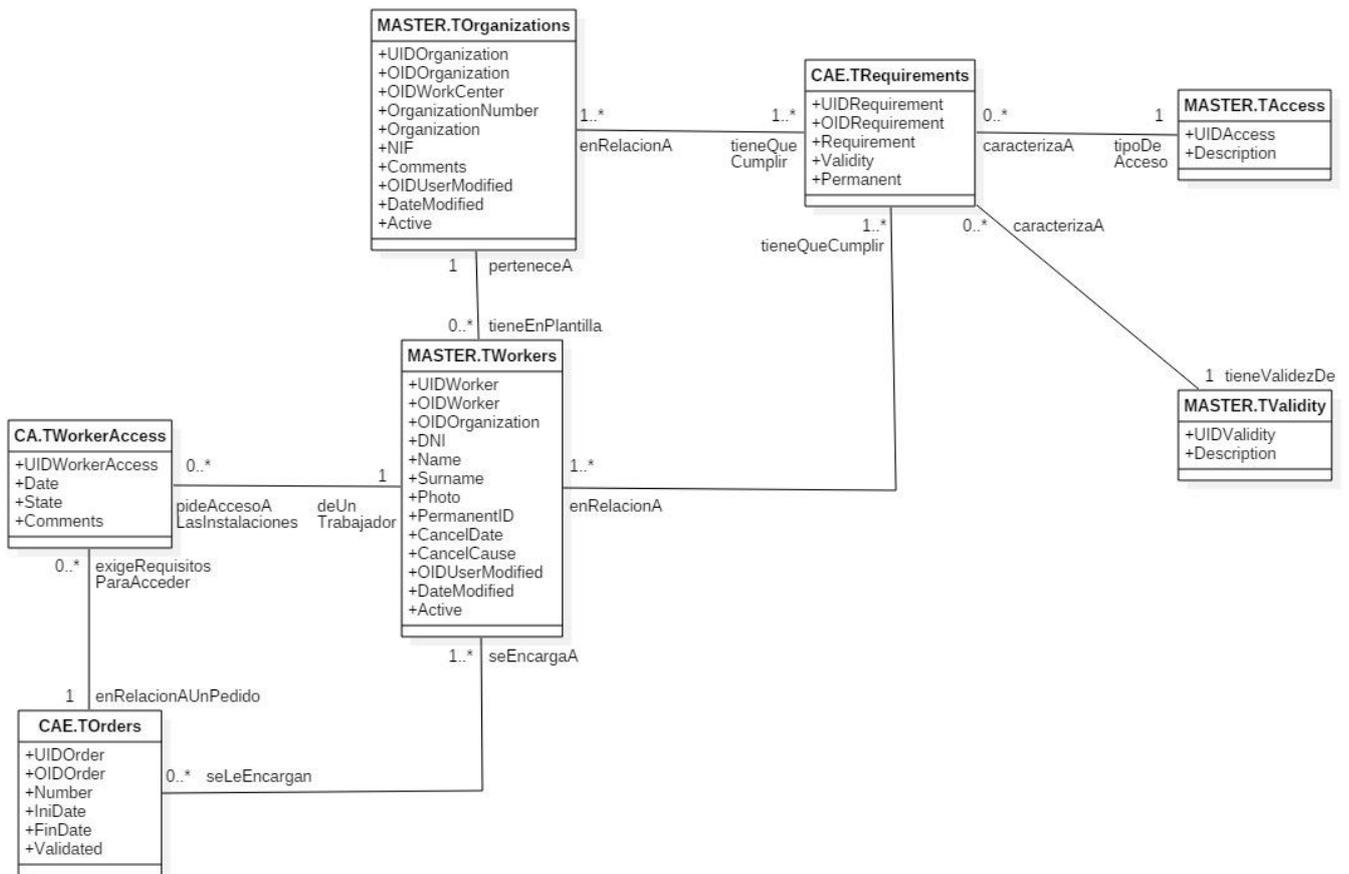


Ilustración 28: Modelo de dominio, Control de accesos

Organizaciones (*MASTER.TOrganizations*): en esta entidad se almacenan los datos que caracterizan las distintas organizaciones incluidas en la base de datos. Estas organizaciones variarán dependiendo del cliente al que estén dando servicio.

Trabajadores (*MASTER.TWorkers*): aquí se guardará la información relacionada con los trabajadores de las organizaciones previamente mencionadas.

¹⁸ [https://msdn.microsoft.com/es-es/library/ff398064\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/ff398064(v=vs.120).aspx)

Pedidos (*CAE.TOrders*): aquí se reunirán los datos pertenecientes a los pedidos que una empresa puede encargar internamente o a otras organizaciones. Se detallará información como las fechas de inicio y fin del pedido, o si ha sido validado.

Acceso de trabajador (*CA.TWorkerAccess*): cada vez que un trabajador externo a la organización quiera acceder a las instalaciones de la misma, se guardará un nuevo registro con la fecha en la que se ha producido esta acción, si el acceso ha sido o no permitido y las razones si hubiera sido denegado.

Requisitos (*CAE.TRequirements*): en esta entidad se guarda la información sobre los distintos requisitos que se pueden exigir a una empresa o a un trabajador. Como puede ser el periodo de validez del requisito o si es imprescindible su cumplimiento para acceder a las instalaciones de la empresa.

Acceso (*MASTER.TAccess*): aquí se guardarán los valores que puede tener el atributo de Acceso para los requisitos, en este caso, libre o condicionado.

Validez (*MASTER.TValidity*): aquí se almacenarán los valores que puede tomar el atributo de Validez para los requisitos, en este caso, año, mes, semana o día.

Cada **trabajador** del sistema pertenecerá a una **organización**, y para controlar si se permite o no su **acceso** se tendrá en cuenta el cumplimiento de **requisitos** tanto por su parte como por parte de la empresa de la que provienen.

La relación entre **trabajador**, **acceso** y **pedido** está presente debido a que, habitualmente, para que un trabajador ajeno a la empresa acceda a las instalaciones de la misma, será para cumplir con la realización de un pedido.

La entidad **requisitos** se relaciona con **acceso** y **validez** ya que estas dos últimas caracterizarán a la primera.

4.2.2. Modelo de dominio: Entrega de EPIs

En el modelo de dominio de la funcionalidad de entrega de equipos de protección individual podemos encontrar las siguientes entidades con sus correspondientes relaciones:

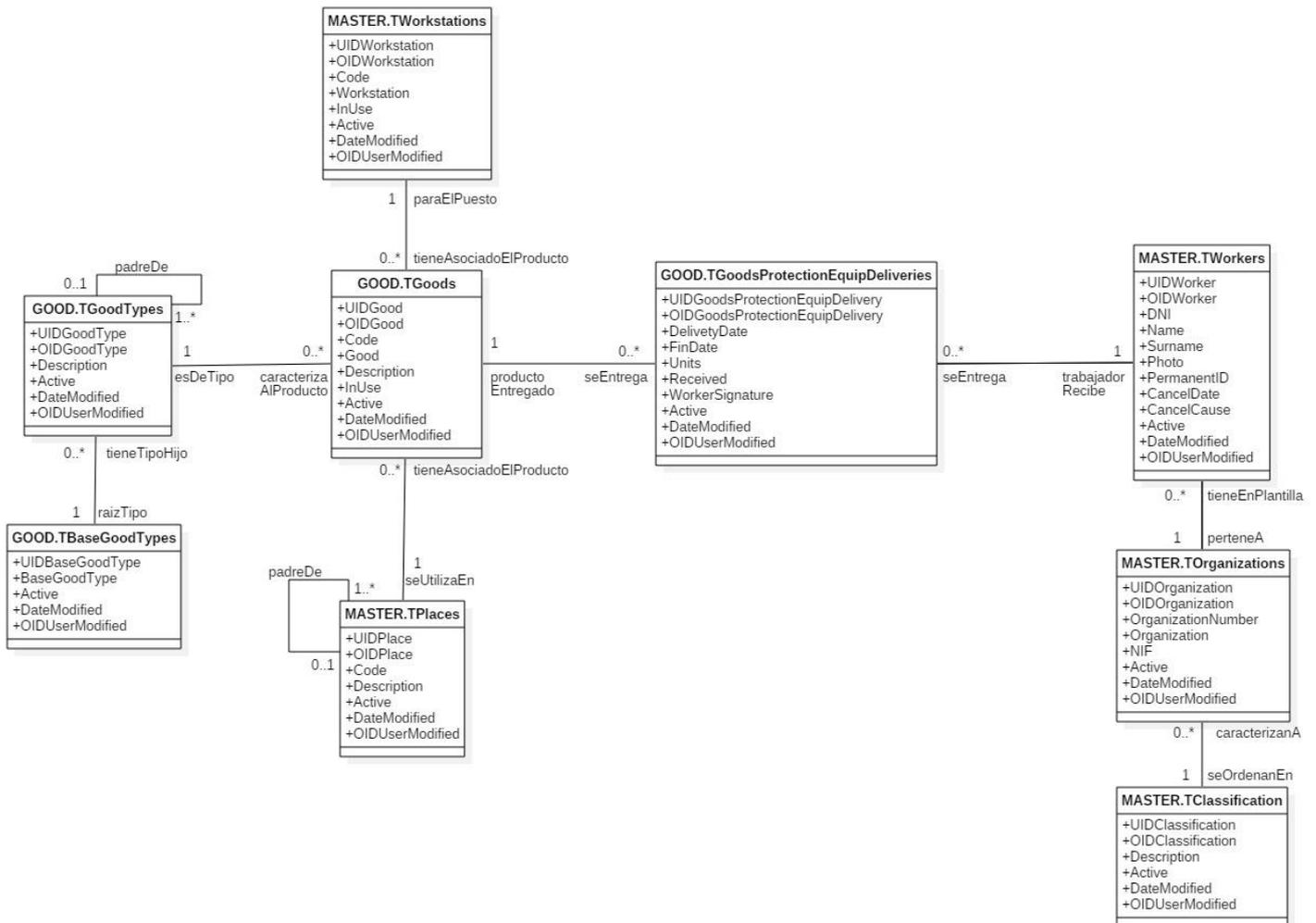


Ilustración 29: Modelo de dominio, Entrega de EPIs

Organizaciones (*MASTER.TOrganizations*): en esta entidad se almacenan los datos que caracterizan las distintas organizaciones incluidas en la base de datos. Estas organizaciones variarán dependiendo del cliente al que estén dando servicio.

Trabajadores (*MASTER.TWorkers*): aquí se guardará la información relacionada con los trabajadores de las organizaciones previamente mencionadas.

Clasificaciones (*MASTER.TClassification*): aquí se almacenarán los valores que puede tomar el atributo de clasificación de las organizaciones.

Productos (*GOOD.TGoods*): en esta entidad se guarda la información sobre los distintos productos que ha adquirido una organización.

Tipos de productos (*GOOD.TGoodsTypes*): aquí se almacenarán los valores que puede tomar el atributo de tipo de los productos.

Tipos de productos base (*GOOD.TBaseGoodsTypes*): los valores de tipo de producto se clasificarán según su tipo de producto base. En este caso las bases predefinidas pueden ser: equipos de trabajo, equipos de protección individual, instalación o químicos.

Puestos de trabajo (*MASTER.TWorkstations*): aquí se guardarán los puestos de trabajo característicos de la organización. Los trabajadores se pueden encasillar en uno o varios puestos de trabajo y determinados productos irán asociados a un puesto específico.

Lugares (*MASTER.TPlaces*): aquí se almacenará el mapa de lugares que caracteriza la empresa. Determinados productos irán asociados a un lugar específico.

Entregas de EPIs (*GOOD.TGoodsProtectionEquipDeliveries*): cuando se entrega a un trabajador un producto con tipo de producto base “Equipo de protección individual”, se añadirá un registro a esta entidad documentado la fecha de la entrega, qué producto ha sido entregado y a quién. Además, se puede ampliar la información con la fecha de reposición prevista o las unidades entregadas.

Cada **trabajador** del sistema pertenecerá a una **organización**, las cuales se catalogan por su **clasificación**.

Los **productos** se clasifican por su **tipo de producto**, estos tipos de producto se catalogan a su vez por su **tipo de producto base**. Además, los **productos** se caracterizan por el **puesto de trabajo** y el **lugar** en el que son utilizados.

La entidad **tipo de producto** cuenta con una relación hacia sí misma, esto permite saber si un producto es padre de otro y, en dicho caso, saber quién es el padre y quién el hijo. Lo mismo pasa con la entidad **lugar**, la relación que posee hacia sí misma es para indicar si un lugar es padre de otro y, en caso afirmativo, conocer quién es el padre y quién es el hijo.

La relación entre **productos**, **entregas de EPIs** y **trabajadores** sólo puede darse cuando el producto en cuestión tiene de tipo de producto base “Equipo de protección individual”. Esta relación se debe a que es necesario dejar constancia cada vez que se entrega un EPI a un trabajador.

4.3. Modelo relacional de datos¹⁹

Dada la gran magnitud del proyecto *Unifikas*, la estructura de las bases de datos está estandarizada. Estas son algunas de las características específicas de la gestión de datos del proyecto:

- **Todas sus tablas están dotadas de un prefijo**, que hace referencia al módulo en el que se trabaja con ellas.
- **Se lleva un control de los registros históricos de los datos introducidos**. Para ello, todas las tablas con registros históricos incluirán 5 columnas:
 - UID: Identificador incremental, asignado automáticamente cada vez que se introduce un nuevo registro en la tabla.
 - OID: Clave primaria, común para todas las tuplas que hagan referencia al mismo dato.
 - OIDUserModified: Usuario que ha modificado la tupla. Para representar esta clave extranjera en los diagramas, se utilizará el nombre *FK_UM*.
 - DateModified: Fecha en la que la tupla fue modificada.
 - Active: Representa el borrado lógico del dato en cuestión. Si el dato ha sido eliminado, en lugar de suprimir todos sus registros históricos de la tabla, se añade una nueva tupla con la columna Active a valor 0.

De esta manera, a la hora de hacer consultas a las bases de datos, sabremos que el registro con mayor UID corresponderá con la última modificación del dato en cuestión. Además, para referirnos a un dato específico lo haremos por su OID.

- **Aislar los datos con los que trabajan los distintos centros de trabajo** de una misma organización, de manera que la empresa pueda decidir cuándo quiere que se comparta esta información, ya sea entre centros de trabajo de su misma organización o con centros de trabajo ajenos a ella. Para ello, las tablas se dotan de una columna “OIDWorkCenter”, que denota el centro de trabajo que ha insertado el dato. Para evitar emborronar los diagramas de datos, excluirémos la tabla de centros de trabajo de estos esquemas, dando por supuesto que todas las tablas que posean entre sus columnas *OIDWorkCenter*, irán vinculada a esta. Para representar esta clave extranjera en los diagramas, se utilizará el nombre *FK_X*.

MASTER.TWorkCenters
+UIDWorkCenter
+OIDWorkCenter
+Code
+Description
+Adress
+InUse
+Active
+DateModified
+OIDUserModified

Ilustración 30: Tabla de Centros de trabajo

¹⁹ <https://msdn.microsoft.com/es-es/library/dd409416.aspx>

4.3.1. Modelo relacional de datos: Control de accesos

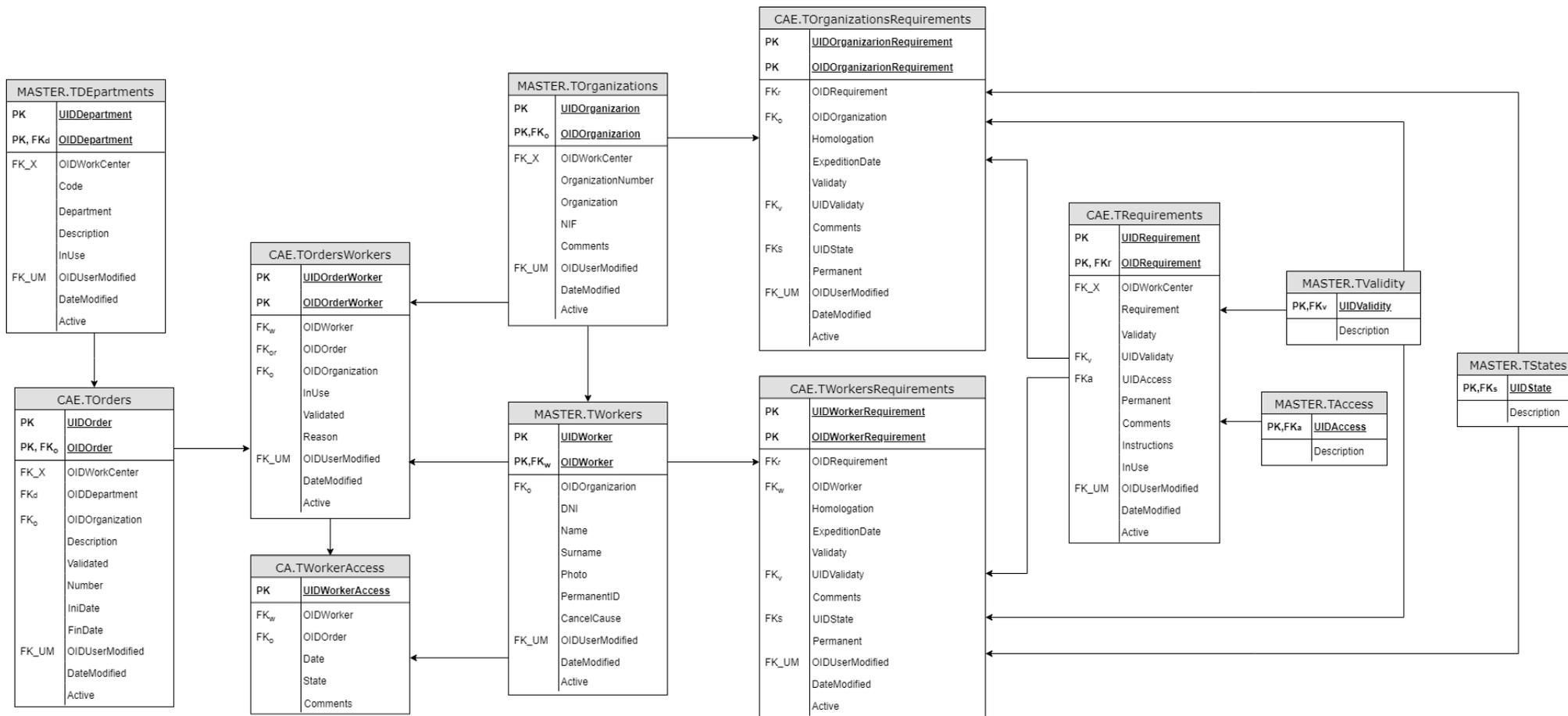


Ilustración 31: Modelo relacional de datos, Control de accesos

Como podemos observar en la siguiente imagen, todas las tablas están dotadas de control de históricos excepto *MASTER.TValidity*, *MASTER.TStates*, *MASTER.TAccess* y *CA.TWorkerAccess*. En el caso de las tres primeras es debido a que siempre van a tener los mismos datos, la validez siempre será día, semana, mes o año; el estado siempre será activo o inactivo; y el acceso libre o condicionado. Sin embargo, la tabla de acceso de trabajadores no tiene históricos porque cada vez que un trabajador intente acceder a las instalaciones se debe generar un registro distinto.

Además, a parte de las tablas principales como son trabajadores, organizaciones, pedidos y requisitos, tenemos algunas tablas cuya funcionalidad es crear asociaciones entre estas tablas principales para cubrir nuevas funcionalidades. Así pues, cuando se encarga un pedido a un trabajador, se añadirá un nuevo registro a la tabla *CAE.TOrdersWorkers*. De la misma manera, cuando queremos documentar el cumplimiento de un requisito por parte de una organización o de un empleado, se añadirá un nuevo registro a las tablas *CAE.TOrganizationsRequirements* y *CAE.TWorkersRequirements*, respectivamente.

4.3.2. Modelo relacional de datos: Entrega de EPIs

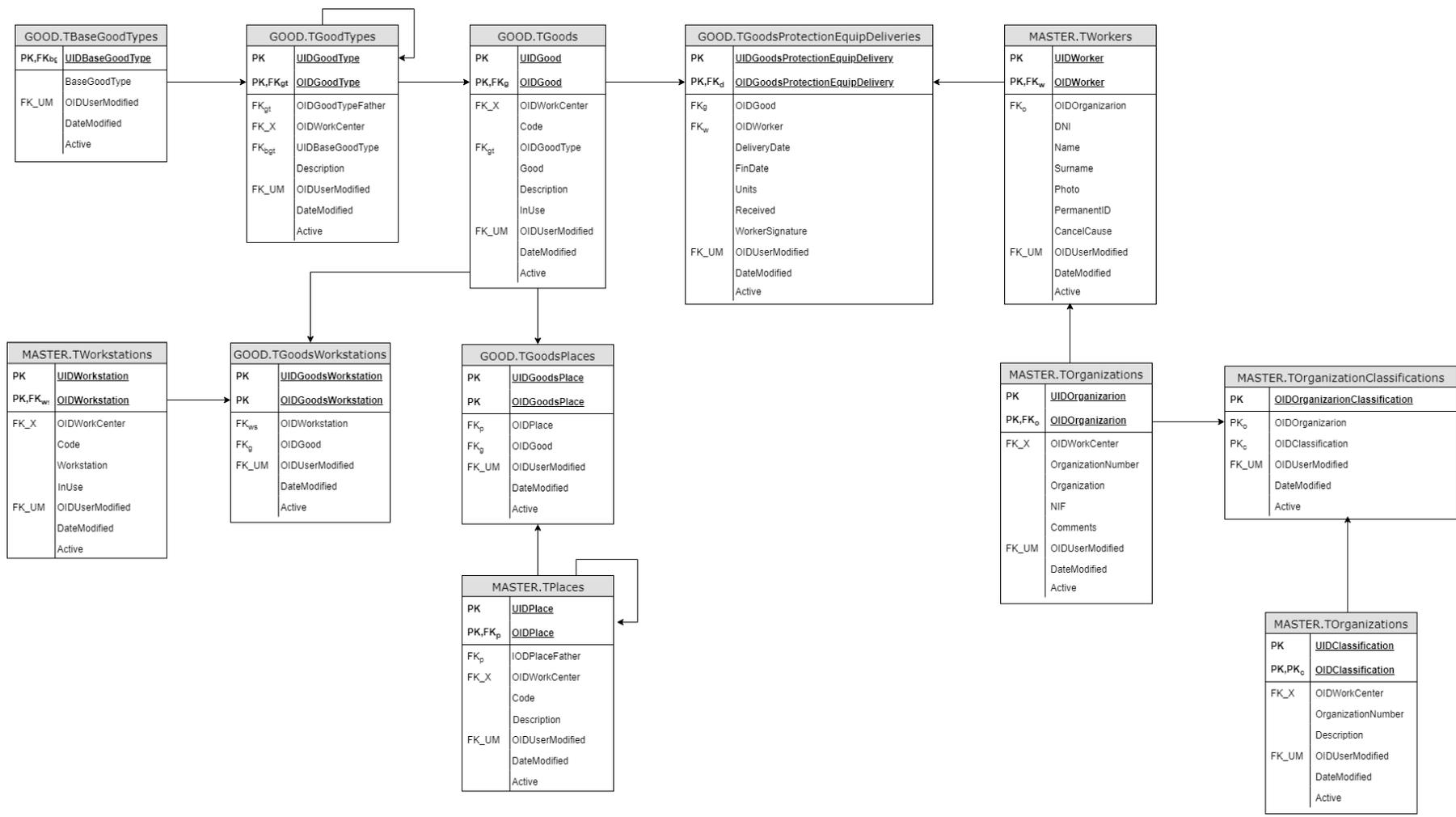


Ilustración 32: Modelo relacional de datos, Entrega de EPIs

De la misma manera que en el modelo anterior, cuando una tabla está destinada a almacenar siempre los mismos valores, carecerá de las columnas asignadas al control de registros históricos, este es el caso de *GOOD.TBaseGoodTypes*.

En esta funcionalidad hay dos tipos de tablas que se encargan de vincular clases principales: Como novedad respecto al modelo de datos anterior, Las tablas *GOOD.TGoodsWorkstations* y *GOOD.TGoodsPlaces* dotarán de puestos y lugares de trabajo a los productos, mientras que *MASTER.TOrganizationsClassification* permitirá clasificar las distintas organizaciones de las bases de datos. De esta manera, utilizamos estas tablas para caracterizar algunas de las tablas principales (productos y organizaciones).

Al igual que en el diagrama anterior, hay una tabla que se encarga de vincular dos clases para dotar de una nueva funcionalidad al proyecto: cuando se entrega un EPI a un trabajador, se añadirá un nuevo registro a la tabla *GOOD.TGoodsProtectionEquipDeliveries*.

4.4. Requisitos no funcionales

4.4.1. Herramientas y tecnología utilizadas

Debido a que el proyecto está enmarcado dentro de un proyecto ya existente mayor, las herramientas base utilizadas, como son Visual Studio para desarrollar las aplicaciones web, la extensión Xamarin.Forms para realizar la aplicación móvil, SQL Server Management Azure para almacenar las Bases de Datos y Lync para las comunicaciones con el resto del equipo de Unifikas, han sido impuestas por la empresa.

Además, los plugins utilizados para dotar de extensiones funcionales a la aplicación móvil se han elegido de entre una lista acotada por el uso de estas herramientas base.

4.4.2. Seguridad

Dada la gran magnitud del proyecto *Unifikas* en su versión web, todos los formularios de la aplicación están dotados de, al menos, un permiso de lectura y otro permiso de escritura, que gestionan qué usuarios que deberían acceder y/o modificar la información almacenada en los mismos.

5

Arquitectura y diseño

A lo largo del capítulo se detalla la arquitectura global de la aplicación, esta arquitectura englobará y establecerá la conexión entre la aplicación web y la aplicación móvil del proyecto. Además, con el objetivo de mostrar la interacción de los datos con el sistema, se mostrará un diagrama de secuencia caracterizando cada una de las dos funcionalidades implementadas, tanto para la parte móvil como para la parte web.

5.1. Arquitectura

Ya que *Unifikas* tiene tanto una versión web como una versión móvil, la arquitectura del proyecto se adaptará a esta característica. Se empezará explicando las principales características de arquitectura en la aplicación web, tras lo que se dará paso a exponer el patrón arquitectural utilizado en la aplicación móvil. Por último, se concretará el vínculo entre las dos. La ilustración treinta y tres, servirá de apoyo para obtener una visión global de la arquitectura de la aplicación.

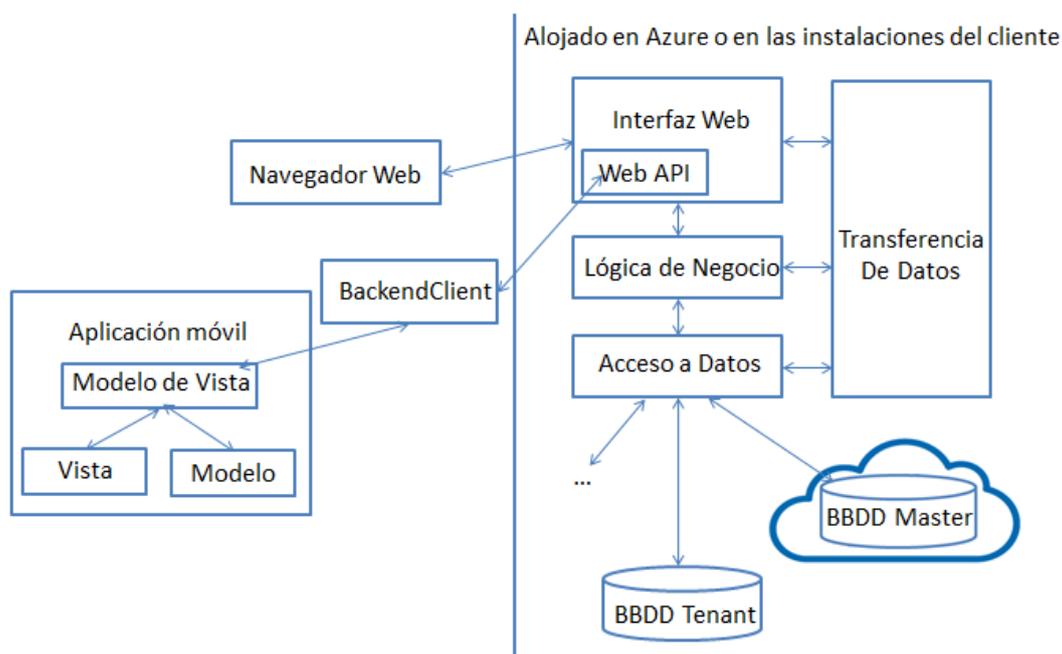


Ilustración 33: Arquitectura de *Unifikas* (Autor: Valeriano Alonso)

5.1.1. Aplicación web

Aunque *Unifikas* utiliza un software común para todas las empresas a las que da servicio, cabe destacar que se personalizan sus funciones dependiendo de las necesidades de cada organización para dar un servicio óptimo. Desde la aplicación web se accede a una base de datos alojada en Azure (BBDD Master) o en las instalaciones del cliente, específica para cada empresa (BBDD Tenant).

Como se puede observar en la ilustración, la aplicación web se estructura en cuatro capas:

- Interfaz Web: la interfaz de usuario. Se encarga de presentar el sistema, comunicar y capturar la información del usuario. Comprende el Web API, que en ASP.NET es un marco que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen exploradores y dispositivos móviles.

- Lógica de negocio: expone la lógica necesaria a la capa de presentación para que el usuario, a través de la interfaz, interactúe con las funcionalidades de la aplicación.
- Acceso a datos: se gestiona el acceso a los datos de la aplicación, realizando la recuperación y el almacenamiento físico de los datos a partir de solicitudes de la capa de negocio.
- Capa de transferencia de datos: contiene distintos objetos cuya funcionalidad es transportar datos entre procesos.

Estas tres últimas capas se dividen en 10 módulos cada una, debido a la extensión de la aplicación.

Además, aunque no esté reflejado en la imagen, cabe resaltar que para acceder a la capa de acceso a datos se utiliza el **patrón factoría**²⁰. Este patrón define una interfaz para crear objetos, permitiendo que sean las subclasses quienes decidan qué clase concreta necesitan instanciar. De esta manera, cuando una clase no puede anticipar el tipo de objetos que debe crear, delegará en sus subclasses la responsabilidad de creación de objetos. El principal objetivo de su utilización es evitar el acoplamiento entre clases en el momento de creación de los objetos, dando flexibilidad a la aplicación frente a futuros cambios.

5.1.2. Aplicación móvil

Como se ve reflejado en la imagen y, a diferencia de la aplicación web, la aplicación móvil utiliza el patrón de arquitectura **Modelo-Vista-Modelo de Vista** (MVVM).

Modelo-Vista-Modelo de Vista es un patrón de arquitectura de software, que propone separar una aplicación en tres componentes principales:

- Modelo: representa la capa de datos, de manera que contiene la información pero nunca las acciones o servicios que la manipulan.
- Modelo de Vista: actúa de intermediario entre el modelo y la vista, contiene toda la lógica de presentación y se comporta como una abstracción de la interfaz. La comunicación entre la vista y el modelo de vista se realiza por medio de los enlaces de datos.
- Vista: La misión de la vista es representar la información a través de los elementos visuales que la componen.

²⁰ <https://msdn.microsoft.com/es-es/library/bb972258.aspx>

Este patrón se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

5.1.3. Conexión entre ambas aplicaciones

Como se ha explicado anteriormente, en la aplicación web los objetos que forman la capa de acceso a datos se encargan de almacenar y recuperar los datos de la base de datos. Sin embargo, desde la aplicación móvil esta transferencia de información se produce de manera distinta.

Se desarrolla un nuevo proyecto (BackendClient) que, utilizando tanto los objetos de transferencia de datos, como los métodos de acceso a datos implementados para la aplicación web, completa un servicio web listo para ser consumido desde la aplicación móvil. Así pues, la aplicación móvil importa el proyecto como si fuera una librería y, a través del mismo, interactúa con la base de datos pertinente.

Debido a que el servicio web contiene las definiciones de algunos objetos de la capa de transferencia de datos de la aplicación web, además de darnos acceso a las bases de datos, completa el componente *Modelo* de nuestra aplicación móvil.

5.2. Diseño de las aplicaciones

Para realizar el diseño se ha utilizado el artefacto definido por el proceso unificado, el diagrama de secuencia²¹. Los diagramas de secuencia muestran la interacción de un conjunto de objetos en un sistema a través de tiempo. Estos diagramas detallan la implementación de un sistema en un escenario concreto.

A continuación, se detalla el diagrama de secuencia de la acción “Ver lista de entregas de un EPI”, perteneciente a la funcionalidad de entrega de EPIs, tanto en la aplicación web como en la aplicación móvil. El empleado seleccionará de la lista de EPIs el que desee para consultar sus entregas y el sistema le mostrará la lista de las mismas. En la aplicación web esta lista se muestra en la misma página de elección de EPI, mientras que en la aplicación móvil se abrirá una nueva página para mostrar todas las entregas.

Después, se mostrará el diagrama de secuencia de la acción “Ver detalle de requisito de un trabajador”, perteneciente a la funcionalidad de control de accesos, únicamente en la aplicación móvil. El empleado seleccionará de la lista de requisitos de trabajador el que desee para que el sistema le muestre los detalles del mismo en una nueva página.

²¹ <https://msdn.microsoft.com/es-es/library/dd409389.aspx>

5.3.1. Diagrama de secuencia: Ver lista de entregas de un EPI en la app móvil

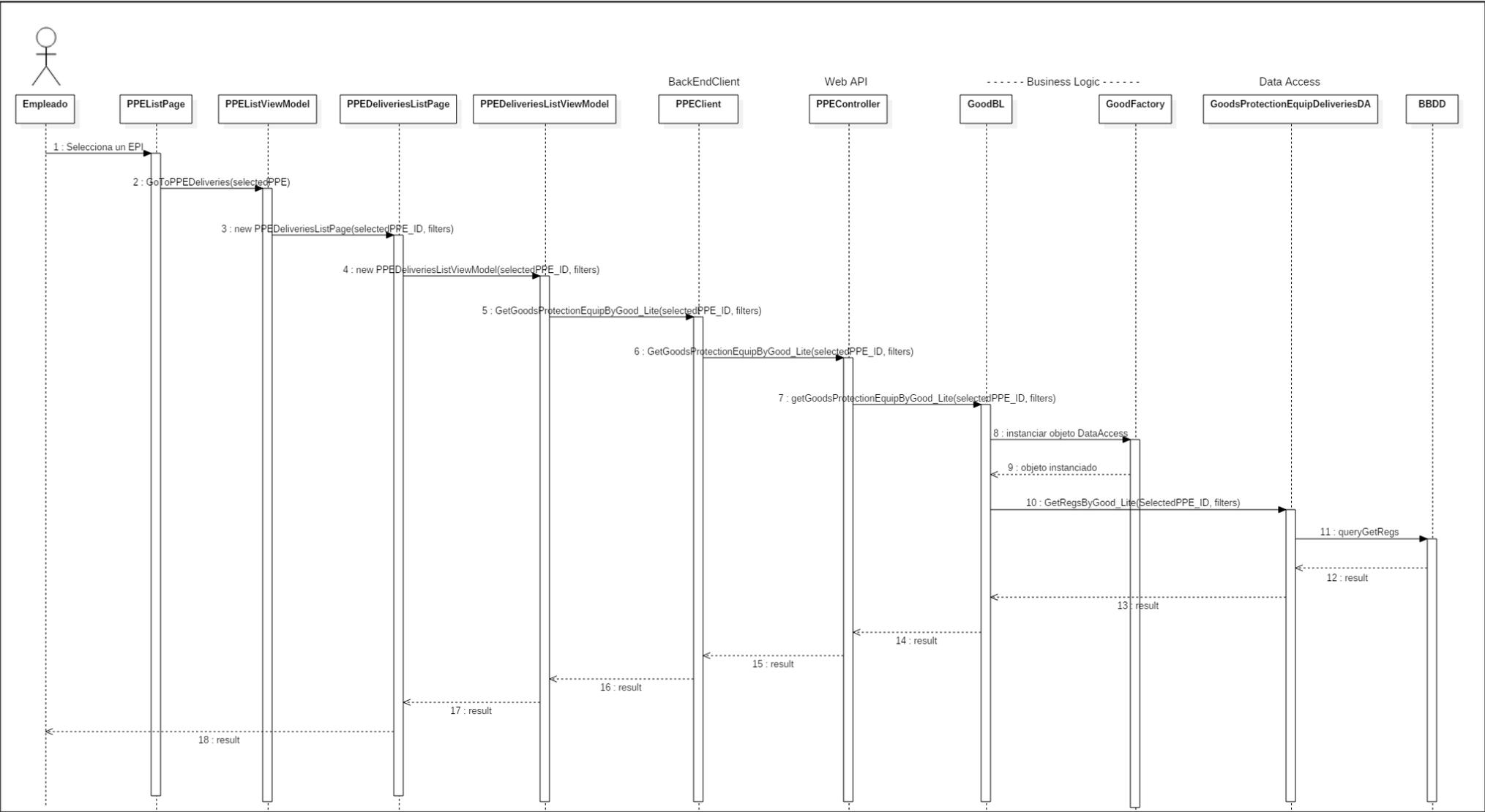


Ilustración 34: Diagrama de secuencia. Ver lista de entregas de un EPI en la app móvil

5.3.2. Diagrama de secuencia: Ver lista de entregas de un EPI en la app web

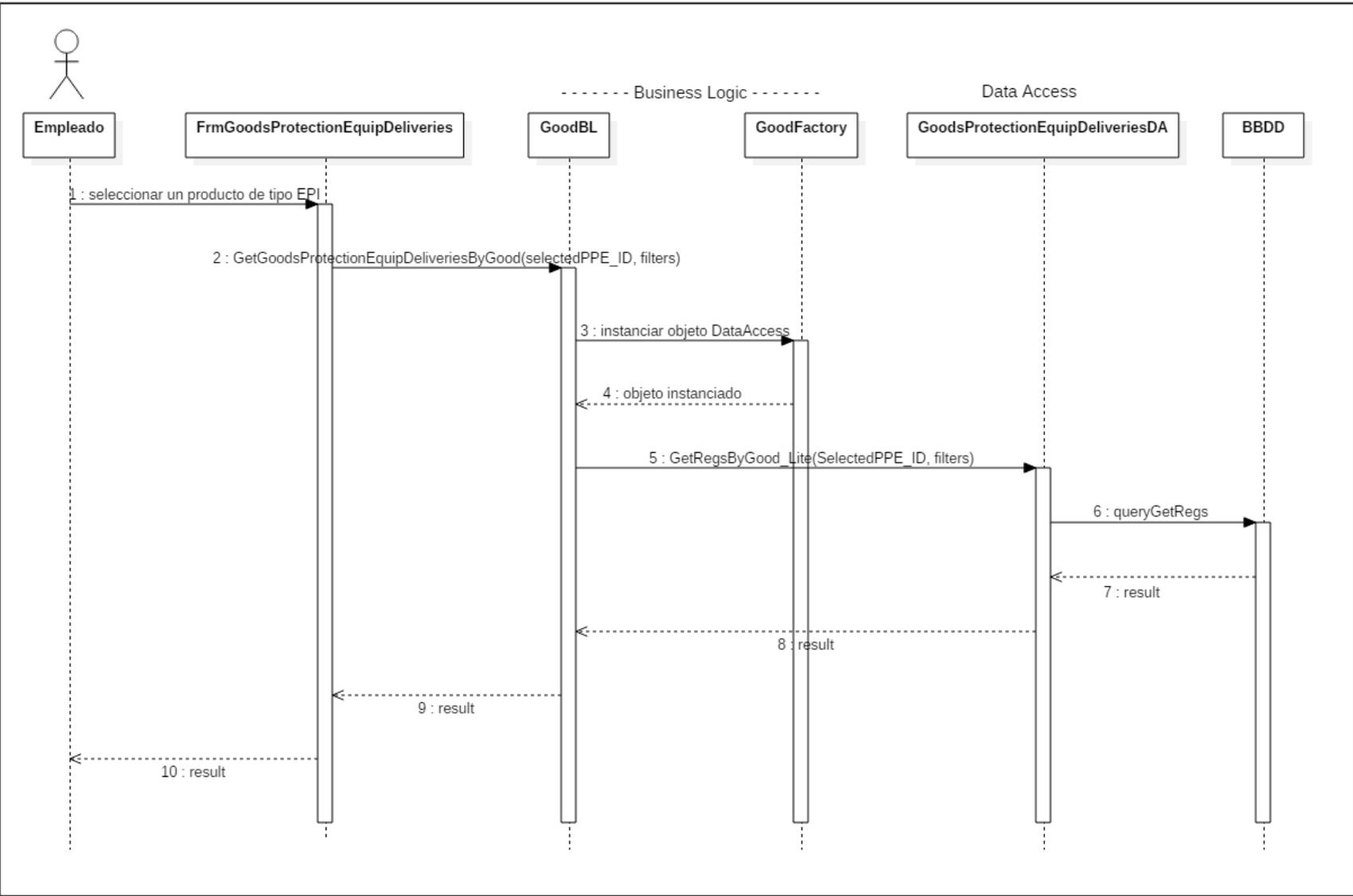


Ilustración 35: Diagrama de secuencia. Ver lista de entregas de un EPI en la app web

5.3.2. Diagrama de secuencia: Ver requisito de un trabajador en la app móvil

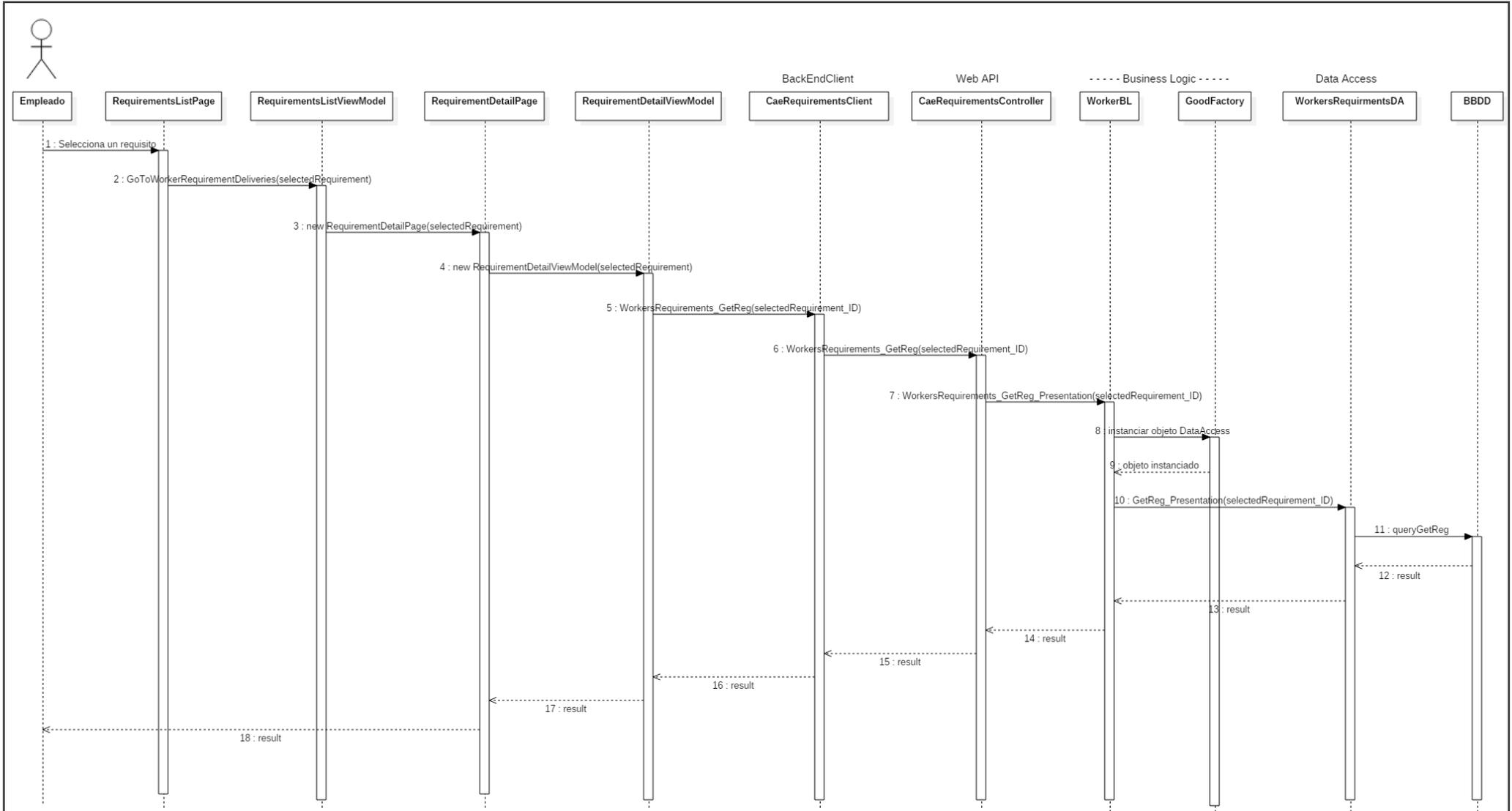


Ilustración 36: Diagrama de secuencia. Ver requisito de un trabajador en la app móvil

6

Implementación

Como se ha explicado en el apartado de arquitectura (5.1), el funcionamiento de las dos aplicaciones desarrolladas es muy distinto, por lo que la implementación de las mismas también será diferente. En este apartado se mostrarán las principales características de la implementación de ambas aplicaciones, dividiendo la explicación en las capas en las que se divide cada parte. Además, se expondrán pequeñas partes del código donde se ve reflejada la separación de dichas capas y la interacción entre ellas.

6.1. Aplicación móvil

Como se ha descrito anteriormente, la aplicación móvil seguirá el patrón MVVM. En este apartado se mostrará la implementación de dicho patrón, explicando el funcionamiento de cada una de las capas de la aplicación y la interacción entre ellas.

6.1.1. Aclaraciones previas

La herramienta elegida para desarrollar la aplicación móvil ha sido Visual Studio, el cual, tras la instalación del paquete Xamarin.Forms, permite crear interfaces de usuario nativas para iOS, Android y Windows Phone desde una única base de código compartida en C#.

Así pues, la implementación de esta aplicación ha sido realizada en el proyecto compartido para todos los entornos y, en casos muy puntuales, se ha tenido que realizar algún pequeño cambio en los proyectos característicos de los distintos entornos para el correcto funcionamiento de alguna de las extensiones añadidas al proyecto.

Además, cabe destacar que, con estas herramientas, las aplicaciones para iOS únicamente se pueden desarrollar desde ordenadores de marca Apple, por lo que no se ha podido probar la aplicación personalmente en este entorno. Sin embargo, el responsable de movilidad de la empresa ha sido el encargado de verificar su correcto funcionamiento en iOS y posteriormente se implementaron las correcciones pertinentes.

Como se puede observar en la imagen, los elementos que componen las distintas capas de la aplicación se encuentran almacenados y divididos por carpetas en el proyecto común a los tres entornos: las vistas estarán en la carpeta *View*, los modelos se guardarán en la carpeta *Model* y los Modelos de Vista se recogerán en la carpeta *ViewModel*. Asimismo, los iconos e imágenes utilizados para ilustrar la interfaz se almacenarán en la carpeta *Images*.

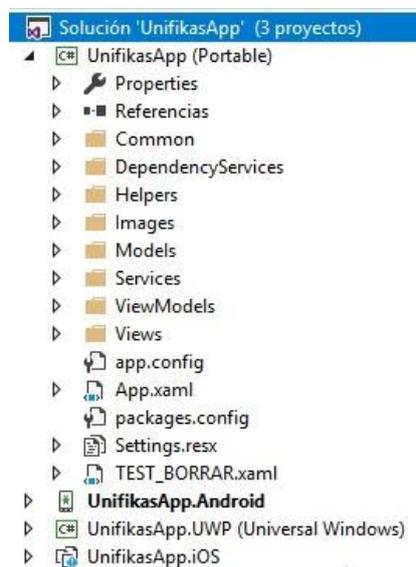


Ilustración 37: Estructura de la app móvil

6.1.2. Vistas y su vínculo con los Modelos de Vista

En el modelo MVVM, las vistas están pensadas exclusivamente para encapsular la lógica de presentación. En MVVM la vista tiene un rol activo, esto significa que en algún momento la vista recibirá o maneja algún evento y tendrá que comunicarse con el modelo de vista, para poder cumplir el requerimiento.

Cada vista se divide en dos elementos: el diseño de página con su respectivo *codebehind*.

6.1.2.1. Diseño de página

Para dotar la vista de un rol activo, los datos que se muestran se actualizan automáticamente desde el modelo de vista. De la misma manera, los datos que se recogen desde su interfaz refrescan la información almacenada en el objeto correspondiente del modelo. Esta característica se gestiona a través de la propiedad *Binding*, que vinculará la información mostrada en la interfaz de usuario a su respectivo objeto guardado en el modelo de vista.

A continuación, se mostrarán distintas maneras de configurar esta propiedad dependiendo del comportamiento que esperamos de la misma.

Cuando se quiere **recoger información simple** facilitada por el usuario, como pueden ser datos introducidos por el teclado del móvil, únicamente hay que vincular el texto del elemento en cuestión con la instancia guardada en el modelo de vista del objeto del modelo deseado. En el caso reflejado por las siguientes imágenes, se asociará el cuadro de entrada de texto referente al NIF de una empresa y el *Switch* que marca si una organización está o no activa con sus respectivas instancias de objetos en el modelo de vista, respectivamente.

En el diseño de página se configurará la entrada de texto de la siguiente manera:

```
<Entry x:Name="NIF" Text="{Binding NIF}" Grid.Column="1" Margin="5,5,20,0" Style="{StaticResource FieldEntry}"/>
```

Si en lugar de tener un campo de texto se quiere utilizar otro control, como puede ser un *Switch*, la propiedad a enlazar será *IsToggled* (está activo).

```
<Switch IsToggled="{Binding Active}" Grid.Column="1" VerticalOptions="CenterAndExpand" HorizontalOptions="StartAndExpand" Margin="20,15,20,0"/>
```

De este modo, la información recogida en el cuadro de texto se almacenará en el siguiente objeto instanciado en el modelo de vista:

```
public string NIF { get; set; }  
public bool Active { get; set; }
```

En el caso de querer dar una **lista de selección** al usuario, habrá que utilizar esta propiedad en dos ocasiones: al configurar el origen de los datos a mostrar y al establecer el elemento seleccionado por defecto, el cual se actualizará tanto en la interfaz como en el modelo cuando el usuario cambie de selección. En la siguiente ilustración se muestra la configuración de una lista de selección que presentará las distintas clasificaciones dentro de las cuales se puede categorizar una organización.

En el diseño de página se configurará un elemento *Picker* de la siguiente manera:

```
<Picker x:Name="Classification" Grid.Column="1" SelectedIndex="{Binding SelectedClassificationViewModel}"
ItemsSource="{Binding ClassificationViewModel}" Margin="5,5,20,0" Style="{StaticResource FieldPicker}"/>
```

De este modo, en el modelo de vista, la lista *ClassificationViewModel* contendrá los posibles valores de la selección, mientras que el objeto *SelectedClassificationViewModel* guardará el valor seleccionado. La definición de estos elementos será la siguiente:

```
#region Classification atributes

List<CyC.Unifikas.Common.DataTransfer.Master.Classification> _classification;

public IEnumerable<DescriptionItemViewModel> ClassificationViewModel
{
    get { return this.GetValue<IEnumerable<DescriptionItemViewModel>>(); }
    private set { this.SetValue(value); }
}

public DescriptionItemViewModel SelectedClassificationViewModel
{
    get { return GetValue<DescriptionItemViewModel>(); }
    set
    {
        if (value != null)
            SetValue(value);
    }
}

#endregion
```

Para instanciar estos elementos se añadirá un primer valor “Todos”, que recogerá la totalidad de los posibles valores que puede tomar la lista. Después, se accederá al servicio web para obtener la lista de los valores a elegir por el usuario. Por último se completará la lista *ClassificationViewModel* con los elementos previamente mencionados y se dotará al objeto *SelectedClassificationViewModel* del primer elemento, en este caso, “Todos”:

```
_classification = new List<CyC.Unifikas.Common.DataTransfer.Master.Classification>
{
    new CyC.Unifikas.Common.DataTransfer.Master.Classification { Description = Constants.PICKER_ALL_VALUE } };
_classification.AddRange(client.OrganizationClassifications_GetAll());
ClassificationViewModel = _classification.Select(p => new DescriptionItemViewModel(p.Description)).ToList();
SelectedClassificationViewModel = ClassificationViewModel.FirstOrDefault();
```

Para manejar los **eventos** desde la vista también es necesario utilizar la propiedad *Binding*. Así pues, esta vez dicha propiedad no irá asociada a un control de entrada o muestra de información, sino que se vinculará con un comando. En la siguiente imagen se mostrará como asociar un evento al pulsar sobre un botón.

En el diseño de página se configurará el botón de la siguiente manera:

```
<Button x:Name="OrganizationSearchButton" Command="{Binding SearchOrganizations}" Margin="20,25,20,10"
VerticalOptions="EndAndExpand" AbsoluteLayout.LayoutFlags="All"
| | | AbsoluteLayout.LayoutBounds="0,0,1,1" Text="BUSCAR" Style="{StaticResource SearchButton}" />
```

De este modo, en el modelo de vista se definirá un comando que haga referencia al comando utilizado en el diseño de página:

```
public ICommand SearchOrganizations { get; set; }
```

Al inicializar el modelo, se inicializará también el comando, asociándolo con un método del modelo:

```
SearchOrganizations = new Command((arg) => SearchOrganizationsCommand(page));
```

Por último, se definirá el método que implemente el comportamiento deseado al presionar el botón en cuestión de la interfaz gráfica:

```
private async void SearchOrganizationsCommand(FindOrganizationPopUp page)
```

6.1.2.2. Codebehind

Para cumplir con el patrón MVVM, hay que intentar tener el menor número de líneas de código en el codebehind de la vista. Para conseguir esto se ha confeccionado una clase *BaseContentPage*, encargada de realizar todas las configuraciones comunes sobre las vistas. De esta manera, la vista heredará dichos métodos de la clase base y únicamente tendrá que inicializar la vista e indicar el modelo de vista con el que se quiere que interactúe la vista.

Así pues, este sería el contenido de la vista del buscador de trabajadores:

```
public partial class WorkerSearchPage : BaseContentPage
{
    WorkerSearchViewModel viewModel;

    public WorkerSearchPage()
    {
        InitializeComponent();

        BindingContext = viewModel = new WorkerSearchViewModel();
    }
}
```

6.1.3. Modelos de Vista y su conexión con el Servicio Web

Los modelos de vista son los responsables de implementar el comportamiento de su respectiva vista para responder a las acciones del usuario y de exponer los datos del modelo de forma tal que sea fácil usar bindings en la vista.

Anteriormente, se ha explicado el vínculo entre la vista y el modelo de vista utilizando bindings, por lo que este apartado se centrará en el acceso a datos de la aplicación móvil. Este intercambio de información con la base de datos se produce a través de un servicio web, importado en forma de librería al proyecto.

Esta librería se divide en dos partes: los clientes y los objetos de transferencia de datos. Los clientes serán los encargados de interactuar con la base de datos, mientras que los objetos de transferencia de datos acompañarán a los modelos en su función de gestionar los datos de la aplicación.

Se han utilizado los objetos de transferencia de datos suministrados por el servicio web en dos tipos de situaciones distintas, para definir variables puntuales dentro de los distintos métodos del modelo de vista, y para definir variables de clase utilizadas para la transferencia de información con la vista correspondiente.

Un ejemplo de **utilización de los objetos de transferencia de datos suministrados por el servicio web como variable puntual** es el representado por la siguiente imagen, donde se inicializa y completa un objeto de tipo *WorkerSearchPresentationFilters*, encargado de guardar los filtros de búsqueda de trabajadores:

```
var filters = new WorkerSearchPresentationFilters
{
    DNI = DNI,
    PermanentID = Identification,
    Worker = NameComplete,
    OIDOrganization = Organization.OIDOrganization
};
```

Para ilustrar la **utilización de los objetos de transferencia de datos suministrados por el servicio web como variable de clase**, se muestra la declaración de las variables entrega y trabajador, que formarán parte del detalle de la entrega de EPIs:

```
public GoodsProtectionEquipDeliveriesPresentation Delivery { get; set; }
public WorkerSearchPresentationResult Worker { get; set; }
```

Se declararán también dos objetos de tipo `ProtectionEquipClient` y `WorkerClient`, respectivamente. De manera que, tras inicializarlos, podrán ser utilizados para hacer consultas a la base de datos:

```
ProtectionEquipClient goodsClient;
WorkerClient workerClient;
```

Al instanciar el modelo de vista, se inicializarán las clases cliente del servicio web. Estos clientes serán utilizados para que, mediante sus respectivas consultas a las bases de datos, inicialicen las variables de clase previamente declaradas.

```
string backEndUrl = Settings.URL_CONNECTION;
goodsClient = new ProtectionEquipClient(backEndUrl);
workerClient = new WorkerClient(backEndUrl);

Delivery = goodsClient.GoodProtectionEquipDelivery_GetReg_Presentation(_oidGoodProtectionEquipDelivery);
Worker = workerClient.Workers_GetReg(Delivery.OIDWorker);
```

Debido a que los controles de las vistas muestran siempre datos de tipo simple: binario, fecha, texto, numérico, etc. no se puede utilizar la propiedad `binding` directamente sobre los objetos heredados del servicio web. Sin embargo, ya estos objetos están formados por conjuntos de datos simples para denotar sus propiedades, se vinculará el control de la vista con la propiedad específica deseada del objeto. Siguiendo el ejemplo anterior, para mostrar el nombre del trabajador al que se le ha entregado el equipo de protección, se utilizará la propiedad `binding` de la siguiente manera:

```
<Label x:Name="WokerName" Text="{Binding Worker.NameComplete}" IsEnabled="{Binding NotSignedYet}" HorizontalOptions=
"FillAndExpand" Style="{StaticResource FieldValueLabel}">
```

6.1.4. Modelos propios de la aplicación y Modelos heredados por el Servicio Web

El modelo es responsable de todos los datos de la aplicación. Contiene la información, pero no los comportamientos o servicios que manipulan la información. La lógica de negocio generalmente se mantiene separada del modelo y se encapsula en otras clases que actúan sobre el modelo.

Como se ha explicado anteriormente, el Servicio Web proporciona objetos de transferencia de datos que son utilizados como parte del modelo de la aplicación móvil.

Sin embargo, en algunas ocasiones, los objetos suministrados por el Servicio Web no están diseñados para albergar la información necesaria para gestionar la información trata. En estos casos, se han diseñado nuevos objetos en el modelo de la aplicación que, aprovechando el alcance de la información que pueden almacenar los objetos suministrados por el Servicio Web, extienden su funcionalidad para abarcar toda la información en cuestión.

Un claro ejemplo de esta situación se da cuando, para mostrar y recoger información sobre uno o varios requisitos, bastaría con la información que se puede almacenar en el objeto heredado por el servicio web *WorkersRequirementsPresentation*. Sin embargo, para mostrar cierta información como el estado, la validez y el acceso del requisito en cuestión, utilizar elementos gráficos como pueden ser bolas de distintos colores (rojo, amarillo o verde) dependiendo del estado de cada uno de estos atributos, aportaría un valor añadido a la interfaz gráfica. Así pues, se creará un nuevo objeto *WorkersRequirementsExtended*, cuya función consistirá en guardar las imágenes correspondientes con las características nombradas con anterioridad, además de recoger toda la información que se puede almacenar en el objeto heredado por el servicio web *WorkersRequirementsPresentation*.

En otras ocasiones, extender la funcionalidad de los objetos suministrados por el servicio web es indispensable para llevar a cabo ciertas funcionalidades de la vista. Por ejemplo, si se quiere asociar un botón a cada requisito para que, al pulsarlo, se descargue la documentación vinculada al mismo, habrá que extender la clase *WorkersRequirementsExtended* previamente nombrada para incluir un dato de tipo comando, que posteriormente se vinculará con su respectivo método.

Además, hay veces en las que se quiere mostrar cierto dato que no se guarda en el objeto principal, como puede ser cuando se quiere trabajar con un dato autogenerado. Para seguir con el ejemplo anterior, la fecha de expiración de un requisito se calcula automáticamente teniendo en cuenta la permanencia y la validez del mismo, por lo que es un dato que no se almacena en las bases de datos. Así pues, si quisiéramos incluir este dato en nuestra interfaz, deberíamos volver a extender la clase *WorkersRequirementsExtended* para guardar esta información adicional.

Teniendo estos atributos disponibles en un único objeto, la tarea de utilizar la propiedad binding para vincular y gestionar la información con la interfaz será más sencilla y limpia.

Para ilustrar los casos anteriores, se muestra una pequeña parte del modelo de la aplicación móvil, la definición del objeto *WorkersRequirementsExtended*:

```
public class WorkersRequirementsExtended
{
    public WorkersRequirementsPresentation Requirement { get; set; }
    public ImageSource StateColor { get; set; }
    public ImageSource ValidityColor { get; set; }
    public ImageSource AccessIcon { get; set; }
    public string ExpirationDate { get; set; }

    public ICommand DownloadReq { get; set; }
}
```

6.1.5. Servicio Web

Previamente, se ha hablado del alcance y las múltiples funciones del servicio web en la aplicación móvil, por lo que este apartado se centrará en cómo fue creado el servicio web.

Como se ha explicado en el apartado de arquitectura (5.1.), el servicio web es desarrollado en el proyecto web e importado como librería en la aplicación móvil.

Es importante hacer un buen diseño previo de los métodos que será necesario desarrollar en el servicio web ya que cada vez que se modifiquen estos procedimientos, tardarán un tiempo hasta poder ser utilizados desde la aplicación móvil. Además, a la hora de diseñar estos nuevos métodos habrá que fijarse en los métodos ya implementados para desarrollar funcionalidades en la aplicación web y aprovechar, en la medida de lo posible, los métodos ya existentes.

Tras diseñar los nuevos métodos, habrá que implementar, si fuera necesario, los procedimientos que no se hayan podido aprovechar de entre los ya desarrollados para la aplicación web. Para no corromper la arquitectura de la aplicación web, estos métodos se implementarán en la capa de acceso a datos y se accederá a ellos a través de la capa de lógica de negocio tras haber sido instanciados por su respectivo patrón factoría.

Cuando todos los métodos están listos para ser explotados por el servicio web se desarrollarán nuevos controladores para el Web Api del proyecto. Se recuerda que en ASP.NET, Web API es un marco que facilita la creación de servicios HTTP disponibles para una amplia variedad de clientes, entre los que se incluyen dispositivos móviles. En el Web API, los controladores son objetos que reciben solicitudes HTTP y devuelven una respuesta en este mismo lenguaje.

Estos controladores recibirán las solicitudes del servicio web, llamarán a la lógica de negocio para explotar los métodos correspondientes, y devolverán la respuesta pertinente.

La siguiente imagen muestra un ejemplo del método utilizado por el controlador para obtener un registro de entrega de equipos de protección dado su identificador:

```
[HttpPost]
public GoodsProtectionEquipDeliveries GoodProtectionEquipDelivery_GetReg(int oidGoodProtectionEquipDelivery)
{
    UnifikasUser user = AuthController.ApiController_GetUnifikasUser();
    try
    {
        GoodsProtectionEquipDeliveries delivery = _goodBL.GetGoodProtectionEquipDelivery(oidGoodProtectionEquipDelivery);
        return delivery;
    }
    catch (Exception ex)
    {
        _btsBL.WriteErrorBL(ex);
        throw;
    }
}
```

Una vez desarrollados los nuevos controladores para la Web API de nuestro proyecto, se pasará a implementar el servicio web. Para ello, se definirá una nueva clase por cada nuevo controlador creado, donde se desarrollarán los nuevos métodos, cuya función será llamar al Web API para obtener o guardar la información pertinente.

La siguiente imagen muestra un ejemplo del método utilizado por el servicio web para obtener un registro de entrega de equipos de protección dado su identificador. Como se puede apreciar, primero se hace una solicitud HTTP al cliente en cuestión, pasándole los parámetros necesarios para llevar a cabo la consulta mediante un objeto de tipo *Diccionario*. Tras esto, se convierte la respuesta obtenida de HTTP al objeto de tipo pertinente, en este caso, un registro de entrega de equipos de protección individual:

```
public GoodsProtectionEquipDeliveries GoodProtectionEquipDelivery_GetReg(int oidGoodProtectionEquipDelivery)
{
    Dictionary<string, string> prms = new Dictionary<string, string>();
    prms.Add("oidGoodProtectionEquipDelivery", Convert.ToString(oidGoodProtectionEquipDelivery));

    string responseApi = ApiHelper.CallApi(_backEndUrl, "/api/ProtectionEquip/GoodProtectionEquipDelivery_GetReg?", prms, null);
    var result = JsonConvert.DeserializeObject<GoodsProtectionEquipDeliveries>(responseApi);
    return result;
}
```

Para certificar que los métodos desarrollados funcionan correctamente, se desarrolla un pequeño programa de pruebas y se comprueba que todos devuelvan los resultados esperados.

Por último, se vincularán las nuevas clases del servicio web y los objetos de transferencia de datos utilizados para desarrollar las nuevas funcionalidades al proyecto *MobileClient*, que posteriormente se ejecutará para obtener un archivo de tipo biblioteca de vínculos dinámicos. Esta será la biblioteca que se importará en el proyecto de la aplicación móvil con el objetivo de poder explotar su contenido.

6.2. Aplicación web

Como se ha nombrado anteriormente, la aplicación web seguirá una estructura basada en 4 capas. En este apartado se mostrará la implementación de dicha estructura, explicando el funcionamiento de cada una de las capas de la aplicación y la interacción entre ellas.

6.2.1. Capa de Transferencia de datos

Esta capa es la encargada de suministrar **Objetos de Transferencia de Datos (DTO en Inglés)**, que se ocupan de transportar datos entre procesos. La motivación para su uso es que la comunicación entre procesos generalmente se realiza recurriendo a interfaces remotas, donde cada llamada es una operación costosa.

Los DTOs son contenedores de datos para mover datos entre capas. Estos objetos carecen de ningún tipo de lógica de negocio, aunque pueden contener mecanismos de serialización y deserialización para facilitar la transferencia de datos.

Ya que, habitualmente, la información con la que se trabajará en la aplicación hará referencia a registros de las bases de datos, todas las tablas incluidas en la base de datos tendrán su respectivo DTO en la aplicación web. De esta manera, cuando se haga una consulta a la base de datos, el resultado que ésta devuelva se podrá convertir en un objeto de tipo DTO.

Los DTOs son utilizados para transportar datos entre todas las capas de la aplicación y, debido al gran volumen de DTOs contenidos en el proyecto, esta capa se dividirá en 10 módulos.

En la siguiente imagen se observa el DTO correspondiente con la tabla de entrega de EPIs de las bases de datos, *GoodsProtectionEquipDeliveries*:

```
namespace CyC.Unifikas.Good.DataTransfer
{
    public class GoodsProtectionEquipDeliveries
    {
        [PrimaryKey()]
        public int UIDGoodProtectionEquipDelivery { get; set; }

        public int OIDGoodProtectionEquipDelivery { get; set; }
        public int OIDGood { get; set; }
        public int OIDWorker { get; set; }
        public DateTime DeliveryDate { get; set; }
        public DateTime? FinDate { get; set; }
        public int? Units { get; set; }
        public bool? Received { get; set; }
        public byte[] WorkerSignature { get; set; }

        public bool Active { get; set; }
        public DateTime DateModified { get; set; }
        public int OIDUserModified { get; set; }
    }
}
```

Instancias de este objeto serán utilizadas en todas las capas de la aplicación, con el objetivo de llevar a cabo la funcionalidad de Entrega de Equipos de Protección Individual. Conforme se vayan explicando las distintas capas se seguirá desarrollando este ejemplo.

6.2.2. Capa de Acceso a datos y su vínculo con las bases de datos

En la **capa de acceso a datos (DAL)** se gestiona el acceso simplificado a los datos almacenados en las bases de datos de la aplicación.

Este tipo de modelo suele ser implementado mediante la creación de una clase de métodos de acceso a datos que hacen referencia directamente a un conjunto correspondiente de procedimientos almacenados de base de datos. Aunque anteriormente la comunicación con la base de datos de la aplicación se produjera de esta manera, ahora estas consultas son plasmadas directamente en el código, permitiendo corregir los errores de una forma más rápida y ahorrando espacio en las bases de datos al suprimir estos procedimientos almacenados de su capacidad.

Siguiendo el ejemplo anterior, en la imagen se observa cómo una de las clases de acceso a datos guarda un nuevo registro de una entrega de Equipo de Protección dado el identificador de la misma, si este identificador fuera 0 significaría que se ha agregado una nueva entrega y se sobrescribirá el 0 del identificador con un nuevo número adjudicado en orden incremental:

```
public GoodsProtectionEquipDeliveries SetReg(GoodsProtectionEquipDeliveries elemento)
{
    var prms = new Dictionary<string, object>
    {
        {"@OIDGoodProtectionEquipDelivery", elemento.OIDGoodProtectionEquipDelivery},
        {"@OIDGoodProtectionEquip", elemento.OIDGoodProtectionEquip},
        {"@OIDWorker", elemento.OIDWorker},
        {"@DeliveryDate", elemento.DeliveryDate},
        {"@FinDate", elemento.FinDate},
        {"@Units", elemento.Units},
        {"@Received", elemento.Received},
        {"@Active", elemento.Active},
        {"@OIDUserModified", elemento.OIDUserModified}
    };
    elemento.OIDGoodProtectionEquipDelivery = Convert.ToInt32(DataConector.ExecQueryScalar(@"
INSERT INTO [GOOD].[TGoodsProtectionEquipDeliveries](
    OIDGoodProtectionEquipDelivery,
    OIDGoodProtectionEquip,
    OIDWorker,
    DeliveryDate,
    FinDate,
    Units,
    Received,
    Active,
    DateModified,
    OIDUserModified
) VALUES (
    " + ((elemento.OIDGoodProtectionEquipDelivery == 0) ?
        "IDENT_CURRENT('GOOD.TGoodsProtectionEquipDeliveries')" : "@OIDGoodProtectionEquipDelivery") + @",
    @OIDGoodProtectionEquip,
    @OIDWorker,
    @DeliveryDate,
    @FinDate,
    @Units,
    @Received,
    @Active,
    GETUTCDATE(),
    @OIDUserModified
);
SELECT scope_identity();", prms));

    if (elemento.OIDGoodProtectionEquipDelivery == 0)
        elemento.OIDGoodProtectionEquipDelivery = elemento.OIDGoodProtectionEquipDelivery;
    return elemento;
}
```

En el caso de que hubiera varias posibles instancias de una clase de acceso a datos dependiendo del contexto en el que se ejecutara la aplicación, estas clases deberían implementar los mismos métodos, de manera que, utilizando el patrón factoría, se pudiera elegir fácilmente la clase a consultar en cada caso. Teniendo en cuenta esta posibilidad, toda clase DAL vendrá acompañada de una interfaz con la definición de sus métodos, de modo que, cuando se quisiera implementar una nueva instancia para obtener otro tipo de objeto, esta nueva clase herede de la interfaz, homogeneizando la creación de la instancia.

6.2.3. Capa de Lógica de negocio

La capa de lógica de negocio (BLL) actúa como intermediario para el intercambio de datos entre la capa de presentación y la capa de acceso a datos. El objetivo es centralizar la mayor cantidad de lógica común posible en la BLL para que no esté acoplada a ninguna interfaz de usuario particular. Al hacerlo, se separa el "núcleo" de la aplicación que, con un buen diseño, debería admitir cualquier número de interfaces de usuario con una cantidad mínima de duplicación de código.

Como se ha explicado en el apartado de arquitectura (5.1), la lógica de negocio de la aplicación se caracteriza por implementar el patrón factoría. Una factoría es un objeto que maneja la creación de otros objetos. En lugar de que sea el propio objeto quien se encargue de todos los aspectos relativos a la creación, se crea otro objeto que lo haga. De esta manera, se libera al objeto que va a ser creado de aquellas responsabilidades que no le corresponden pero que son necesarias para su creación, manteniendo su independencia.

Las factorías se utilizan cuando la creación de un objeto implica algo más que una simple instanciación, en este caso concreto, utilizar el patrón factoría es necesario debido a que no se sabe qué tipo concreto de objeto se va a instanciar hasta el momento preciso de la instanciación.

Así pues, primero se crea un nuevo método en la factoría para implementar los distintos objetos dependiendo del contexto de la aplicación. Como se ha explicado anteriormente, para lograr que esto sea posible, el patrón deberá devolver siempre un objeto de tipo interfaz para que la lógica de negocio sea consciente de que los únicos métodos que puede utilizar son los definidos en la interfaz, ya que serán los únicos que implementen todas las clases que la hereden.

Siguiendo el ejemplo anterior, el patrón factoría se encargará de inicializar las distintas clases que hereden de la interfaz común, *IGoodsProtectionEquipDeliveriesDA*. En este caso,

únicamente se ha implementado la opción de que el contexto de la aplicación sea normal, por lo que si el contexto fuera testear saltaría una excepción:

```
public static IGoodsProtectionEquipDeliveriesDA CreateIGoodsProtectionEquipDeliveriesDA()
{
    if (Context == BaseDataAccessFactoryContext.Normal)
    {
        return new GoodsProtectionEquipDeliveriesDA();
    }
    else if (Context == BaseDataAccessFactoryContext.Testing)
    {
        throw new NotImplementedException("IGoodsProtectionEquipDeliveriesDA Mock not implemented");
    }
    throw new NotImplementedException("GoodCtx value not implemented");
}
```

Una vez definida la factoría se obtendrá en la lógica de negocios el objeto de acceso a datos a través de la factoría. Como se aprecia en la imagen, se inicializa un objeto de tipo *IGoodsProtectionEquipDeliveriesDA* para poder acceder a sus métodos y, con ello, obtener y guardar información en las bases de datos:

```
private IGoodsProtectionEquipDeliveriesDA _goodsProtectionEquipDeliveriesData = null;
private IGoodsProtectionEquipDeliveriesDA GoodsProtectionEquipDeliveriesData
{
    get
    {
        if (_goodsProtectionEquipDeliveriesData == null)
        {
            _goodsProtectionEquipDeliveriesData = GoodFactory.CreateIGoodsProtectionEquipDeliveriesDA();
        }
        return _goodsProtectionEquipDeliveriesData;
    }
}
```

Tras inicializar el acceso a datos se podrán utilizar los métodos definidos en el mismo. Así pues, guardar un nuevo registro de entrega de un EPI se implementaría de la siguiente manera:

```
public GoodsProtectionEquipDeliveries SetGoodsProtectionEquipDelivery(GoodsProtectionEquipDeliveries elemento, int oidUserModified)
{
    elemento.OIDUserModified = oidUserModified;
    elemento.Active = true;
    return GoodsProtectionEquipDeliveriesData.SetReg(elemento);
}
```

6.2.4. Capa de presentación

La capa de presentación es la interfaz de usuario, encargada de presentar el sistema, comunicar y capturar la información del usuario.

Para llevar a cabo su función conectará con la lógica de negocio, de manera que la información presentada y recogida al usuario sea pertinentemente obtenida y guardada en las bases de datos. Para ello, se instanciarán objetos de tipo lógica de negocio, accediendo a sus métodos con el objetivo de llevar a cabo las funcionalidades de la interfaz.

Se termina con el mismo ejemplo que se ha ido siguiendo a lo largo de las explicaciones de las distintas capas de la aplicación. Esta vez, la acción de guardar un nuevo registro de entrega

estará vinculada al evento de pulsar un botón de “Aceptar”. Así pues, primero se comprobará si el registro es nuevo o es una modificación de una entrega previamente almacenada en las bases de datos. Después se preparará el objeto a guardar consultando la información proporcionada en los controles del diseño de la interfaz y se llamará a la lógica de negocio para que almacene el nuevo registro. Por último, se mostrará un mensaje de inserción o modificación correcta por pantalla:

```
private void Form_SaveData()
{
    GoodsProtectionEquipDeliveries item = new GoodsProtectionEquipDeliveries { OIDGoodProtectionEquipDelivery = _oidGoodsProtectionEquipDelivery };
    bool insertion = true;
    if (item.OIDGoodProtectionEquipDelivery > 0)
    {
        insertion = false;
        item = _goodBL.GetGoodProtectionEquipDelivery(item.OIDGoodProtectionEquipDelivery);
    }

    item.OIDGood = _oidGood;
    item.OIDWorker = findworker.Value;
    item.DeliveryDate = (DateTime) calDate.Value;
    item.FinDate = calReplacementDate.Value;
    if (tbUnits.HasValue())
        item.Units = Convert.ToInt32(tbUnits.Value);
    item.Received = cbEPIReceived.Checked;

    item = _goodBL.SetGoodsProtectionEquipDelivery(item, Me.OIDUser);

    if (insertion)
    {
        _oidGoodsProtectionEquipDelivery = item.OIDGoodProtectionEquipDelivery;
    }

    ctlMessages.OKMsgRes(insertion ? "ReviewInsertCorrect" : "ReviewModifyCorrect");
}
```

7

Verificación y pruebas

En este capítulo se detallan algunas de las pruebas realizadas a lo largo de la implementación del proyecto. Mediante la ejecución de las pruebas, se han ido identificando y rectificando los errores encontrados en los prototipos realizados. Además de las pruebas aquí expuestas, se han realizado múltiples pruebas para solventar los errores existentes.

Las pruebas de la aplicación móvil se han realizado sobre emuladores y dispositivos Android únicamente, ya que para compilar la aplicación iOS desde un ordenador Windows se necesita un certificado no gratuito. Por otro lado, para realizar las pruebas de la aplicación móvil se han utilizado los navegadores Chrome e Internet Explorer.

La dinámica de pruebas llevada a cabo ambas aplicaciones ha sido muy distinta debido a la política de testeo instaurada dentro del equipo de proyecto de *Unifikas web*.

En la aplicación móvil las pruebas finales han sido realizadas íntegramente por la alumna desarrolladora, comprobando su correcto funcionamiento junto a la jefa de proyecto al finalizar cada iteración.

En la aplicación web, los cambios realizados tienen que pasar tres filtros antes de instalar el producto final en los clientes. Primero, deben pasar el filtro de desarrollo, cuyas pruebas serán realizadas por el desarrollador que ha implementado los cambios. Después, tendrán que pasar el filtro de preproducción, donde otro miembro del equipo de desarrollo del proyecto o la analista realizará las pruebas. Por último, los cambios pasarán el filtro de producción, donde, nuevamente, alguno de los integrantes del equipo que no haya realizado las pruebas sobre los cambios en cuestión. De esta manera, se pretende dar a los clientes un software de calidad, debidamente testado y, en la medida de lo posible, sin errores de funcionamiento.

Pruebas Prototipo 1

Se comprueba que la funcionalidad de control de accesos de personal ajeno a la empresa a las instalaciones de la misma se ha realizado correctamente. Este prototipo no tendrá acceso a datos, puesto a que el servicio web se le añadirá en el prototipo 4, de modo que las pruebas completas sobre esta funcionalidad se realizarán tras implementar dicho prototipo.

Código	Descripción	Resultado esperado	Resultado obtenido	Observaciones
1	Comprobar que los <i>Modelos de Vista</i> están preparados para poder integrar los nuevos métodos del Servicio Web en cuanto estén disponibles	Comentar las futuras llamadas al Servicio Web y, en su lugar, simular su funcionamiento con datos insertados desde código. Desde este estado, realizar las pruebas 2, 3, 4 y 5	Con las futuras llamadas al Servicio Web comentadas y simulando su funcionamiento con datos insertados desde código. Desde este estado, se realizan las pruebas 2, 3, 4 y 5	Correcto*
2	Comprobar la correcta navegación entre las páginas de la funcionalidad	La navegación entre páginas debería ser correcta tanto hacia adelante como al pulsar las flechas de retroceso	La navegación entre páginas es correcta tanto hacia adelante como al pulsar las flechas de retroceso	Correcto

3	Comprobar la aparición y desvanecimiento de los PopUps	Los PopUps deberían aparecer al pulsar sobre determinados campos y desaparecer cuando su función haya concluido o cuando se pulse fuera del PopUp	Los PopUps aparecen al pulsar sobre determinados campos y desaparecen cuando su función ha concluido y cuando se pulsa fuera del PopUp	Correcto
4	Comprobar la visualización de los datos del <i>Modelo de Vista</i> desde la Interfaz	La Interfaz debería mostrar los datos guardados en las instancias de objetos guardadas en el <i>Modelo de Vista</i>	La Interfaz muestra los datos guardados en las instancias de objetos guardadas en el <i>Modelo de Vista</i>	Correcto
5	Comprobar la captura de datos de la Interfaz, de modo que en un futuro puedan utilizarse para interactuar con la BBDD	Al modificar información en la interfaz, los datos se deberían actualizar en las instancias de objetos correspondientes guardadas en el <i>Modelo de Vista</i>	Al modificar información en la interfaz, los datos se actualizan en las instancias de objetos correspondientes guardadas en el <i>Modelo de Vista</i>	Correcto
6	Simular descargar documentación de un requisito	Al pulsar sobre el botón “Abrir” en las listas de requisitos; o sobre el botón “Ver” en el detalle de requisitos debería mostrarse una imagen guardada en la BD	Al pulsar sobre el botón “Abrir” en las listas de requisitos; o sobre el botón “Ver” en el detalle de requisitos se muestra una imagen guardada en la BD	Correcto
7	Simular subir documentación a un requisito	Como se ha probará con el emulador, debería abrir las carpetas internas del mismo. Como el emulador carece de documentos, no se comprobará la funcionalidad completa.	La aplicación redirecciona a las carpetas internas del emulador.	Correcto

Tabla 3: Batería de pruebas del prototipo 1

Pruebas Prototipo 2

Se comprueba la funcionalidad de entregas de EPIs en la aplicación web. Al entregar el equipo, bastará con seleccionar el checkbox “He recibido” para indicar la correcta entrega del mismo, la posibilidad de firmar al entregar el EPI no ha sido implementada en la web.

Cabe destacar que los controles personalizados de búsqueda de productos ya estaban implementados, por lo que no se incluirán sus pruebas como parte del proyecto.

Código	Descripción	Resultado esperado	Resultado obtenido	Observaciones
1	Visualizar las entregas de un EPI sin entregas	La lista de las entregas debería estar vacía.	La lista de las entregas debería estar vacía.	Correcto
2	Añadir entrega de EPI sin rellenar campos obligatorios	Aparecerá un mensaje indicando que se rellenen dichos campos.	Aparece un mensaje indicando que se rellenen dichos campos.	Correcto
3	Añadir entrega a un EPI rellenando los campos pertinentes	La entrega debería guardarse en la BD y aparecer en la lista de entregas del EPI. Aparecerá un mensaje indicando que la entrega ha sido creada correctamente.	La entrega se guarda en la BD y aparece en la lista de entregas del EPI. Aparece un mensaje indicando que la entrega ha sido creada correctamente.	Correcto
4	Modificar la entrega previamente guardada	Los cambios deberían reflejarse en la BD, en el detalle de la entrega y en la lista de entregas. Aparecerá un mensaje indicando que la entrega ha sido modificada correctamente.	Los cambios se reflejan en la BD, en el detalle de la entrega y en la lista de entregas. Aparece un mensaje indicando que la entrega ha sido creada correctamente.	Correcto
5	Añadir varias entregas y probar únicamente el filtro de organización (esta prueba se realizará con distintas organizaciones)	La lista de entregas debería contener únicamente aquellas cuyo trabajador pertenezca a la organización filtrada.	La lista de entregas contiene únicamente aquellas cuyo trabajador pertenezca a la organización filtrada.	Correcto

6	Probar el filtro de organización con una empresa que no esté entre la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto
7	Probar únicamente el filtro de trabajador (esta prueba se realizará con distintos trabajadores)	La lista de entregas debería contener únicamente aquellas cuyo trabajador corresponda con el filtrado.	La lista de entregas contiene únicamente aquellas cuyo trabajador corresponda con el filtrado.	Correcto
8	Probar el filtro de trabajador con un empleado que no esté entre la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto
9	Probar únicamente el filtro de estado (esta prueba se realizará con los dos posibles estados)	La lista de entregas debería contener únicamente aquellas con estado como el filtrado.	La lista de entregas contiene únicamente aquellas con estado como el filtrado.	Correcto
10	Probar el filtro de estado con un valor que no esté en la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto
11	Probar únicamente el filtro de recibido (esta prueba se realizará con los dos posibles valores del recibido)	La lista de entregas debería contener únicamente aquellas con el campo de recibido como el filtrado.	La lista de entregas contiene únicamente aquellas con el campo de recibido como el filtrado.	Correcto
12	Probar el filtro de recibido con un valor que no esté en la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto

13	Eliminar entrega desde el detalle de la entrega	Se redirigirá a la página de lista de entregas, entrega borrada ya no formará parte de la lista. Aparecerá un mensaje indicando que la entrega ha sido eliminada correctamente.	Se redirige a la página de lista de entregas, entrega borrada ya no forma parte de la lista. Aparece un mensaje indicando que la entrega ha sido eliminada correctamente.	Correcto
14	Eliminar entrega desde la página de la lista de entregas	La entrega borrada desaparecerá de la lista. Aparecerá un mensaje indicando que la entrega ha sido eliminada correctamente.	La entrega borrada desaparece de la lista. Aparece un mensaje indicando que la entrega ha sido eliminada correctamente.	Correcto

Tabla 4: Batería de pruebas del prototipo 2

Pruebas Prototipo 3

Se comprueba que la funcionalidad de entrega de EPIs se ha realizado correctamente en la aplicación móvil. Este prototipo no tendrá acceso a datos, puesto a que el servicio web se le añadirá en el prototipo 4, de modo que las pruebas completas sobre esta funcionalidad se realizarán tras implementar dicho prototipo.

Código	Descripción	Resultado esperado	Resultado obtenido	Observaciones
1	Comprobar que los <i>Modelos de Vista</i> están preparados para poder integrar los nuevos métodos del Servicio Web en cuanto estén disponibles	Comentar las futuras llamadas al Servicio Web y, en su lugar, simular su funcionamiento con datos insertados desde código. Desde este estado, realizar las pruebas 2, 3, 4 y 5	Con las futuras llamadas al Servicio Web comentadas y simulando su funcionamiento con datos insertados desde código. Desde este estado, se realizan las pruebas 2, 3, 4 y 5	Correcto*
2	Comprobar la correcta navegación entre las páginas de la funcionalidad	La navegación entre páginas debería ser correcta tanto hacia adelante como al pulsar las flechas de retroceso	La navegación entre páginas es correcta tanto hacia adelante como al pulsar las flechas de retroceso	Correcto

3	Comprobar la aparición y desvanecimiento de los PopUps	Los PopUps deberían aparecer al pulsar sobre determinados campos y desaparecer cuando su función haya concluido o cuando se pulse fuera del PopUp	Los PopUps aparecen al pulsar sobre determinados campos y desaparecen cuando su función ha concluido y cuando se pulsa fuera del PopUp	Correcto
4	Comprobar la visualización de los datos del <i>Modelo de Vista</i> desde la Interfaz	La Interfaz debería mostrar los datos guardados en las instancias de objetos guardadas en el <i>Modelo de Vista</i>	La Interfaz muestra los datos guardados en las instancias de objetos guardadas en el <i>Modelo de Vista</i>	Correcto
5	Comprobar la captura de datos de la Interfaz, de modo que en un futuro puedan utilizarse para interactuar con la BBDD	Al modificar información en la interfaz, los datos se deberían actualizar en las instancias de objetos correspondientes guardadas en el <i>Modelo de Vista</i>	Al modificar información en la interfaz, los datos se actualizan en las instancias de objetos correspondientes guardadas en el <i>Modelo de Vista</i>	Correcto
6	Capturar y mostrar la firma del trabajador al entregarle un EPI	Tras firmar en el detalle de la entrega de EPIs, debería refrescarse la página mostrando la firma en formato imagen	Tras firmar en el detalle de la entrega de EPIs, se refresca la página y la imagen de firma está vacía	Incorrecto

Tabla 5: Batería de pruebas del prototipo 3

Pruebas Prototipo 4

Se comprueba que las funcionalidades de control de accesos de personal ajeno a la empresa a las instalaciones de la misma y de entrega de EPIs han sido realizadas correctamente en la aplicación móvil. En este prototipo se añaden el servicio web al proyecto, por lo que se realizarán las pruebas de la aplicación móvil en conjunto.

Como se muestra a continuación, las pruebas de este prototipo han sido divididas por funcionalidad.

Pruebas de Control de Accesos

Código	Descripción	Resultado esperado	Resultado obtenido	Observaciones
1	Comprobar el correcto funcionamiento del PopUp de búsqueda de organización	Al pulsar sobre el icono correspondiente el PopUp se debería abrir, sus controles funcionar correctamente y permitir seleccionar la organización deseada	Al pulsar sobre el icono correspondiente el PopUp se abre, sus controles funcionan correctamente y permite seleccionar la organización deseada	Correcto
2	Buscar trabajadores únicamente con el filtro de organización (obligatorio)	Debería aparecer una lista con todos los trabajadores de dicha organización	Aparece una lista con todos los trabajadores de dicha organización	Correcto
3	Buscar trabajadores con más filtros	Debería aparecer una lista con todos los trabajadores que cumplan los requisitos	Aparece una lista con todos los trabajadores que cumplan los requisitos	Correcto
4	Pulsar sobre un trabajador sin pedidos	Debería redirigir a la ficha de dicho trabajador sin vincularlo con ningún pedido	Redirige a la ficha de dicho trabajador sin vincularlo con ningún pedido	Correcto
5	Pulsar sobre un trabajador con un pedido activo, con fechas de contrata en regla y activo para dicho pedido	Debería redirigir a la ficha de dicho trabajador vinculándolo con dicho pedido	Redirige a la ficha de dicho trabajador vinculándolo con dicho pedido	Correcto
6	Pulsar sobre un trabajador con varios pedidos activos, con fechas de contrata en regla y activo para dichos pedidos	Debería redirigir a una página con la lista de todos los pedidos del trabajador (activos e inactivos, con fecha de contrata caducada o en regla y con trabajador activo o inactivo para dicho pedido)	Redirige a una página con la lista de todos los pedidos (activos e inactivos, con fecha de contrata en regla o caducadas y en los que el trabajador este activo o inactivo para dicho pedido)	Correcto

7	Pulsar sobre un trabajador con algún pedidos inactivo, con fechas de contrata caducadas y/o inactivo para dicho pedido	Debería redirigir a una página con la lista de todos los pedidos del trabajador (activos e inactivos, con fecha de contrata en regla o caducadas y en los que el trabajador este activo o inactivo para dicho pedido)	Redirige a una página con la lista de todos los pedidos (activos e inactivos, con fecha de contrata en regla o caducadas y en los que el trabajador este activo o inactivo para dicho pedido)	Correcto
8	Desde la página de lista de pedidos de un trabajador, seleccionar un pedido y pulsar sobre "Ver ficha"	Debería redirigir a la ficha de dicho trabajador vinculándolo con dicho pedido	Redirige a la ficha de dicho trabajador vinculándolo con dicho pedido	Correcto
9	Desde la página de lista de pedidos de un trabajador, seleccionar un pedido y pulsar sobre "Detalle del pedido"	Debería redirigir a la página de detalle del pedido seleccionado, vinculándolo con el trabajador en cuestión	Redirige a la página de detalle del pedido seleccionado, vinculándolo con el trabajador en cuestión	Correcto
10	Desde la página de detalle del pedido, editar las fechas de contrata	Debería actualizar las fechas de contrata del pedido tanto en la BD como en la aplicación	Actualiza las fechas de contrata del pedido tanto en la BD como en la aplicación	Correcto
11	Desde la página de detalle del pedido, editar si el trabajador está activo o inactivo	Debería actualizar el estado del trabajador respecto al pedido tanto en la BD como en la aplicación	Actualiza el estado del trabajador respecto al pedido tanto en la BD como en la aplicación	Correcto
12	Desde la ficha de un trabajador vinculado con un pedido con fechas de contrata caducadas, actualizar las fechas en cuestión	Debería actualizar las fechas de contrata del pedido tanto en la BD como en la aplicación y eliminar el aviso de fechas en la ficha	Actualiza las fechas de contrata del pedido tanto en la BD como en la aplicación y elimina el aviso de fechas en la ficha	Correcto

13	Desde la ficha de un trabajador vinculado con un pedido en el que esté inactivo, actualizar el estado del trabajador respecto al pedido	Debería actualizar el estado del trabajador respecto al pedido tanto en la BD como en la aplicación y eliminar el aviso de inactividad de trabajador en la ficha	Actualiza el estado del trabajador respecto al pedido tanto en la BD como en la aplicación y elimina el aviso de inactividad de trabajador en la ficha	Correcto
14	Desde la ficha de un trabajador vinculado a un pedido, pulsar sobre el código del pedido	Debería redirigir a la página de detalle del pedido en cuestión, vinculándolo con el trabajador en cuestión	Redirige a la página de detalle del pedido en cuestión, vinculándolo con el trabajador en cuestión	Correcto
15	Desde la ficha de un trabajador pulsar sobre el botón "Requisitos trabajador"	Debería redirigir a una página con la lista de todos los requisitos del trabajador	Redirige a una página con la lista de todos los requisitos del trabajador	Correcto
16	Desde la ficha de un trabajador pulsar sobre el botón "Requisitos empresa"	Debería redirigir a una página con la lista de todos los requisitos de la empresa a la que pertenece el trabajador	Redirige a una página con la lista de todos los requisitos de la empresa a la que pertenece el trabajador	Correcto
17	Desde la ficha de un trabajador pulsar sobre el botón "Bloqueantes"	Debería redirigir a una página con la lista de todos los requisitos tanto del trabajador como de la empresa a la que pertenece que estén por entregar y que su acceso sea condicionado	Redirige a una página con la lista de todos los requisitos tanto del trabajador como de la empresa a la que pertenece que estén por entregar y que su acceso sea condicionado	Correcto
18	Desde cualquiera de las páginas de listas de requisitos, pulsar sobre el botón "Abrir" de alguno de los pedidos de la lista	Debería mostrar la documentación más reciente del pedido en cuestión	Muestra la documentación más reciente del pedido en cuestión	Correcto

19	Desde cualquiera de las páginas de listas de requisitos, pulsar sobre alguno de los pedidos de la lista	Debería redirigir a la página de detalle del requisito	Redirige a la página de detalle del requisito	Correcto
20	Desde el detalle de pedido, pulsar sobre el botón "Ver"	Debería mostrar la documentación más reciente del pedido en cuestión	Muestra la documentación más reciente del pedido en cuestión	Correcto
21	Desde el detalle de pedido, pulsar sobre el botón "Subir"	Debería redireccionarse a las carpetas internas del móvil, para elegir el archivo a subir. Al seleccionarlo, debería guardarse la nueva documentación en la BD	Redirecciona a las carpetas internas del móvil, para elegir el archivo a subir. Al seleccionarlo, se guarda la nueva documentación en la BD	Correcto
22	Desde el detalle de pedido, realizar alguna modificación sobre el mismo	Deberían guardarse los cambios en la BD y mostrar las modificaciones por pantalla	Se guardan los cambios en la BD y se muestran las modificaciones por pantalla	Correcto

Tabla 6: Batería de pruebas del prototipo 4. Funcionalidad: Control de Accesos

Pruebas de Entrega de EPIS

Código	Descripción	Resultado esperado	Resultado obtenido	Observaciones
1	Buscar EPI sin filtros	Debería mostrar una lista con todos los EPIS	Muestra una lista con todos los EPIS	Correcto
2	Buscar EPI con filtros	Debería mostrar una lista con todos los EPIS que cumplan los filtros	Muestra una lista con todos los EPIS que cumplan los filtros	Correcto
3	Visualizar las entregas de un EPI sin entregas	La lista de las entregas debería estar vacía.	La lista de las entregas debería estar vacía.	Correcto
4	Añadir entrega de EPI sin rellenar campos obligatorios	Aparecerá un mensaje indicando que se rellenen dichos campos.	Aparece un mensaje indicando que se rellenen dichos campos.	Correcto

5	Añadir entrega a un EPI rellenando los campos pertinentes	La entrega debería guardarse en la BD y aparecer en la lista de entregas del EPI. Aparecerá un mensaje indicando que la entrega ha sido creada correctamente.	La entrega se guarda en la BD y aparece en la lista de entregas del EPI. Aparece un mensaje indicando que la entrega ha sido creada correctamente.	Correcto
6	Modificar la entrega previamente guardada	Los cambios deberían reflejarse en la BD, en el detalle de la entrega y en la lista de entregas. Aparecerá un mensaje indicando que la entrega ha sido modificada correctamente.	Los cambios se reflejan en la BD, en el detalle de la entrega y en la lista de entregas. Aparece un mensaje indicando que la entrega ha sido creada correctamente.	Correcto
7	Capturar y mostrar la firma del trabajador al entregarle un EPI	Tras firmar en el detalle de la entrega de EPIs, debería guardar la imagen en la BBDD y refrescarse la página mostrando la firma en formato imagen	Tras firmar en el detalle de la entrega de EPIs, se guarda la imagen en la BBDD y se refresca la página y la imagen de firma está vacía	Correcto
8	Añadir varias entregas y probar únicamente el filtro de organización (esta prueba se realizará con distintas organizaciones)	La lista de entregas debería contener únicamente aquellas cuyo trabajador pertenezca a la organización filtrada.	La lista de entregas contiene únicamente aquellas cuyo trabajador pertenezca a la organización filtrada.	Correcto
9	Probar el filtro de organización con una empresa que no esté entre la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto
10	Comprobar el correcto funcionamiento del	Al pulsar sobre el icono correspondiente el	Al pulsar sobre el icono correspondiente el	Correcto

	PopUp de búsqueda de trabajador	PopUp se debería abrir, sus controles funcionar correctamente y permitir seleccionar el trabajador deseado	PopUp se abre, sus controles funcionan correctamente y permite seleccionar el trabajador deseado	
11	Probar únicamente el filtro de trabajador (esta prueba se realizará con distintos trabajadores)	La lista de entregas debería contener únicamente aquellas cuyo trabajador corresponda con el filtrado.	La lista de entregas contiene únicamente aquellas cuyo trabajador corresponda con el filtrado.	Correcto
12	Probar el filtro de trabajador con un empleado que no esté entre la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto
13	Probar únicamente el filtro de estado (esta prueba se realizará con los dos posibles estados)	La lista de entregas debería contener únicamente aquellas con estado como el filtrado.	La lista de entregas contiene únicamente aquellas con estado como el filtrado.	Correcto
14	Probar el filtro de estado con un valor que no esté en la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto
15	Probar únicamente el filtro de recibido (esta prueba se realizará con los dos posibles valores del recibido)	La lista de entregas debería contener únicamente aquellas con el campo de recibido como el filtrado.	La lista de entregas contiene únicamente aquellas con el campo de recibido como el filtrado.	Correcto
16	Probar el filtro de recibido con un valor que no esté en la lista	Aparecerá un mensaje indicando que no hay valores que mostrar. La lista de entregas deberá estar vacía	Aparece un mensaje indicando que no hay valores que mostrar. La lista de entregas está vacía	Correcto

17	Eliminar entrega desde el detalle de la entrega	Se redirigirá a la página de lista de entregas, entrega borrada ya no formará parte de la lista. Aparecerá un mensaje indicando que la entrega ha sido eliminada correctamente.	Se redirige a la página de lista de entregas, entrega borrada ya no forma parte de la lista. Aparece un mensaje indicando que la entrega ha sido eliminada correctamente.	Correcto
18	Eliminar entrega desde la página de la lista de entregas	La entrega borrada desaparecerá de la lista. Aparecerá un mensaje indicando que la entrega ha sido eliminada correctamente.	La entrega borrada desaparece de la lista. Aparece un mensaje indicando que la entrega ha sido eliminada correctamente.	Correcto

Tabla 7: Batería de pruebas del prototipo 4. Funcionalidad: Entrega de EPIS

8

Seguimiento del proyecto

En este capítulo se ilustra la fase de seguimiento y control del proyecto.

Se empezará explicando la evolución del alcance del proyecto y las causas por las que se ha modificado. Después, se hará una reflexión sobre la gestión del tiempo durante el proyecto, comparando la planificación inicial con las horas reales invertidas. Por último, se hará una revisión del cumplimiento de los objetivos iniciales del proyecto.

8.1. Gestión del alcance

Debido a que el desarrollo del proyecto se ha ido ajustando a las necesidades de la empresa, el alcance inicial de la aplicación ha sufrido algunas variaciones. La dinámica seguida durante el desarrollo del proyecto se ha basado en realizar una reunión con la jefa de proyecto al finalizar cada iteración, de manera que se realizaba una crítica constructiva sobre el trabajo realizado hasta el momento y se acordaba el alcance de la siguiente iteración. Debido a esto, dentro del alcance de la aplicación se han realizado varias modificaciones.

En un principio se iba a desarrollar la funcionalidad de firmar las entregas de los Equipos de Protección Individual no sólo en la aplicación móvil, sino también en la aplicación web. La problemática de implementar esta funcionalidad en la web, radica en que es necesario un aparato externo para recoger la firma, ya que intentar firmar con el ratón no es una opción viable. Aunque se gestionó la posibilidad de adquirir un dispositivo externo para llevar a cabo esta funcionalidad, al final no se llevó a cabo.

Otra variación del alcance inicial del proyecto fue la implementación de los nuevos métodos para que el servicio web diera servicio a las nuevas funcionalidades implementadas en la aplicación móvil. Inicialmente el equipo de desarrollo de *Unifikas* iba a encargarse de realizar la totalidad de estos nuevos métodos, sin embargo, y con excepción de un par de métodos concretos, la realización del servicio web se convirtió en parte del alcance del proyecto.

8.2. Gestión del tiempo

La gestión del tiempo ha sido una de las principales dificultades del proyecto. El compaginar la realización del proyecto en la empresa con las clases del segundo cuatrimestre fue más compleja de lo esperado, provocando una menor dedicación semanal al desarrollo del proyecto, lo que desembocó en un retraso sistemático de las fecha de fin planificadas para cada iteración.

Además, en algunas partes fases de la planificación se ha pecado de ser demasiado de ser demasiado optimista a la hora de establecer la dedicación estimada, de manera que, por regla general, las dedicaciones reales han sido algo superiores a las estimadas. La siguiente imagen muestra la desviación de estas estimaciones respecto a las horas reales invertidas, desglosando estos datos en las tareas iniciales del proyecto:

Tareas	Dedicación estimada	Dedicación real
Gestión	38	36
Reuniones con la jefa de proyecto	10	10
Reuniones con el tutor	10	8
Definición de objetivos	2	2
Definición de tareas	3	2
Planificación de tareas	3	2
Seguimiento y control	10	12
Formación	36	32
Xamarin Forms	35	30
Patrón MVVM	1	2
Análisis	6	15
Definir modelo de dominio	3	5
Definir casos de uso	3	10
Diseño	25	23
Diseño de la aplicación móvil	2	2
Diseño de métodos del Servicio Web para la app móvil	2	4
Diseño de nuevas tablas y procedimientos de BD para app web	1	2
Diagramas de secuencia	20	15
Implementación	276	291
Definir prototipos	1	1
Implementar prototipo 1.1	50	50
Implementar prototipo 1.2	45	40
Implementar prototipo 2	40	45
Implementar prototipo 3.1	40	35
Implementar prototipo 3.2	40	35
Implementar prototipo 4	40	50
Implementar Servicio Web	20	35
Pruebas	20	22
Elaborar plan de pruebas por prototipo	6	6
Probar prototipo 1.1	2	2
Probar prototipo 1.2	2	2
Probar prototipo 2	2	3
Probar prototipo 3.1	2	2
Probar prototipo 3.2	2	2
Probar prototipo 4	4	5
Documentación	70	100
Realizar memoria	50	80
Preparar defensa	20	20
Total	471	519

Tabla 8: Comparativa entre la dedicación estimada y real

Como se puede comprobar en el gráfico y la tabla anterior, la necesidad de aumentar el tiempo de dedicación ha sido general, exceptuando en las dos primeras fases.

Para la tarea de análisis se necesitó más tiempo para realizar los modelos de dominio y los casos de uso, debido a la extensión de las funcionalidades desarrolladas y a la extensión de la aplicación.

En la tarea de implementación, las dedicaciones de las cuatro primeras tareas sufren una pequeña desviación respecto a la planificación. Sin embargo, tanto para la implementación del Servicio Web como para su instauración en la aplicación móvil el tiempo estimado resultó ser insuficiente.

Para realizar las pruebas también se necesitó más tiempo al haber estimado menos horas de las realmente necesarias para realizar todas las tareas de implementación planificadas.

En último lugar, la estimación de las horas para realizar la documentación también resultó ser escasa a la hora de realizar la memoria.

En total, ha habido que dedicar casi cincuenta horas adicionales al tiempo estimado, como se muestra en la siguiente gráfica comparativa entre la dedicación estimada y real del proyecto:

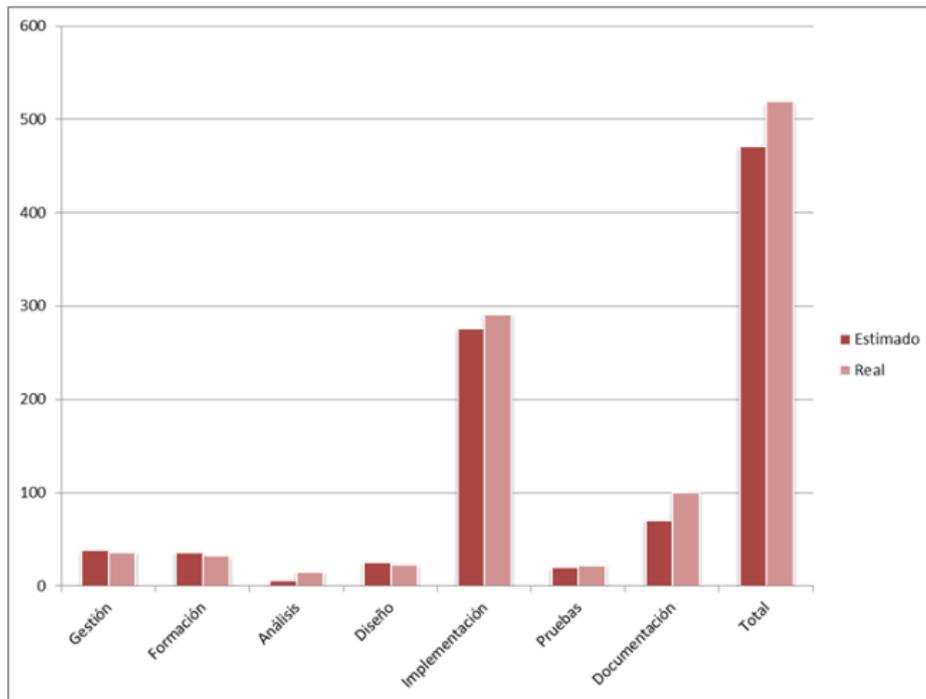


Ilustración 38: Gráfica comparativa de la dedicación estimada y real

9

Conclusiones y líneas futuras

En este capítulo se recogen las conclusiones obtenidas a lo largo del proyecto, tanto a nivel de desarrollo, como las conclusiones finales desde un punto de vista crítico y personal.

9.1. Revisión de objetivos

Una de las principales tareas del proyecto es definir los objetivos que se pretenden conseguir con su realización. Al finalizar el proyecto, hay que de revisar si esos objetivos han sido completados o no y, en caso afirmativo, determinar en qué medida, y, en caso negativo, determinar las causas.

Los dos primeros objetivos trataban de ayudar a combatir los riesgos de accidentes y de perjuicios para la salud de los empleados y de ayudar a gestionar el control de accesos de personal ajeno a la empresa a las instalaciones de la misma, respectivamente. Teóricamente, debido a que la empresa ha mostrado su conformidad respecto al trabajo realizado, estos objetivos se han cumplido. Sin embargo, hasta que las organizaciones que han comprado *Unifikas* no utilicen las nuevas funcionalidades implementadas, no se podrá saber a ciencia cierta si el enfoque del proyecto ha sido el apropiado.

Otro de los objetivos se basaba en el aprendizaje de la alumna en el desarrollo de aplicaciones móviles. Este objetivo se puede dar por conseguido, ya que, de no haberlo cumplido, el proyecto hubiera fracasado. Teniendo en cuenta que hasta el momento únicamente se habían desarrollado aplicaciones para ordenador, gracias al proyecto se ha podido ampliar la capacidad de desarrollo a nuevas plataformas. Además, cabe destacar que la realización del proyecto puede desembocar en la futura especialización laboral en esta área de desarrollo.

El último objetivo trataba de realizar el proyecto de manera autónoma. Cabe destacar que durante la carrera se han realizado proyectos, aunque de una envergadura mucho más modesta que éste. También se ha aprendido a realizar planificaciones, pero nunca de un modo tan real como en este caso. Sumando todos los factores favorables y en contra para que este proyecto saliera bien auto gestionándolo, el resultado final ha superado las expectativas iniciales del proyecto y, aunque se haya alargado más de lo previsto, la desviación temporal ha merecido la pena. Aunque durante las prácticas de verano en la empresa se conoció lo que es un desarrollo real de una aplicación, tener la oportunidad de planificar y gestionar el proyecto ha supuesto un crecimiento, tanto personal como profesional, en la autora del TFG.

9.2. Problemas durante la realización del TFG

Teniendo en cuenta la magnitud del TFG, se podía prever la aparición de problemas durante su desarrollo, ya que entran en juego demasiadas variables y controlarlas todas es una tarea complicada.

Uno de los problemas recurrentes a la hora de realizar el proyecto fueron los problemas con el funcionamiento del software. En algunas ocasiones, el problema fue que el software no estaba actualizado. Otras veces, extensiones de la herramienta Visual Studio se deshabilitaron, provocando la imposibilidad de trabajar con el Emulador hasta descubrir el problema.

Otro de los principales problemas fue que, debido a la falta de investigación por parte de la desarrolladora, cuando se llevó a cabo la implementación de los nuevos métodos para el servicio web, ignoraba que podía crear un Internet Information Service²² (IIS) Local para probar los métodos que se encontraban en fase de desarrollo directamente en la aplicación móvil. Debido a esto, se tuvo que esperar hasta que los nuevos métodos pasaran a la fase de producción y se publicaran para probarlos por primera vez. Más adelante, este comportamiento se corrigió, permitiendo probar las correcciones de los métodos tras sus modificaciones.

El último problema surgido fue poder alcanzar la dedicación de horas planificadas semanales durante los meses de marzo y abril. Como se ha explicado anteriormente, esto se debió a la necesidad de compaginar la realización del proyecto con las clases del segundo cuatrimestre.

9.3. Conclusiones del proyecto

A rasgos generales, la metodología utilizada ha sido un acierto. Desde el punto de vista de la empresa, ha facilitado evaluar si se iban consiguiendo los resultados esperados, asimismo, la división en iteraciones ha sido de ayuda a la hora de proponer mejoras sobre el trabajo realizado. Desde el punto de vista personal, la división del proyecto en iteraciones ha facilitado la planificación del mismo, además de que, al conseguir un resultado tangible al final de cada iteración ha incentivado el desarrollo continuado del producto. Por otro lado, que cada iteración se desarrolle durante un periodo corto, facilita el progreso continuo y favorece la detección de dificultades durante el proceso de implementación.

Aunque el proceso de familiarización con nuevas tecnologías y patrones de desarrollo siempre es un reto, tener la oportunidad de aprender sobre un campo de especialización tan puntero como lo son los dispositivos móviles, ha sido una experiencia muy enriquecedora.

Una de las características más interesantes del proyecto es su dualidad, implementar la misma funcionalidad tanto para aplicaciones web como para aplicaciones móviles ayuda a comprender las diferencias y dificultades a la hora de desarrollar ambos tipos de proyectos.

²² [https://msdn.microsoft.com/es-es/library/hh831725\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831725(v=ws.11).aspx)

Los objetivos prioritarios establecidos con la empresa sobre la aplicación han sido cumplidos y, tras la reunión de fin de proyecto, se puede concluir que se ha desarrollado el proyecto de forma exitosa.

9.4. Conclusiones desde un punto de vista personal

La autora de este TFG está muy satisfecha con el trabajo realizado, la dedicación invertida para finalizarlo de acuerdo a los requisitos de la empresa ha merecido la pena. Se ha adquirido experiencia no sólo en áreas tecnológicas desconocidas hasta el momento, sino también en un entorno laboral con un equipo de profesionales.

El conocimiento adquirido ha sido muy amplio. Estudiar nuevos patrones de diseño tanto para desarrollar aplicaciones web como aplicaciones móviles, ha ayudado a aumentar las habilidades como desarrolladora. Aprender a diseñar y desarrollar aplicaciones móviles desde cero, además de la familiarización con plugins para ampliar la funcionalidad de las mismas, ha hecho que surja en la autora de este proyecto un interés por este campo de desarrollo, planteando un futuro laboral entorno a él. Instruirse para desarrollar, implementar, probar e implantar servicios web, ha completado el estudio alrededor de la programación para dispositivos móviles ya que, de no haber tenido la oportunidad de desarrollar el servicio web para la aplicación, no se podría decir que se ha profundizado en este campo. Además, la integración en un equipo de trabajo estable, multidisciplinar y muy consolidado que ha establecido una sólida metodología de trabajo tanto a nivel de desarrollo como de gestión de versiones y despliegue de las mismas, ha hecho de esta experiencia una inmersión total en el entorno laboral. Todo ello ha supuesto un reto y un aprendizaje tanto desde el punto de vista profesional como personal.

Asimismo, ver la capacidad de solucionar conflictos y aprender nuevas tecnologías de manera autónoma durante el proyecto, ha ayudado a conocer los límites personales de la autora.

9.5. Trabajo futuro

Al ser un proyecto en desarrollo, *Unifikas* está en constante evolución. Las necesidades de los clientes actuales y futuros de la aplicación, sustentan los nuevos requisitos y mejoras a realizar. Como *Unifikas* es un proyecto de gran magnitud, en este apartado únicamente se analizarán las propuestas relacionadas con las funcionalidades desarrolladas.

Las mejoras sobre la aplicación web serían:

- Sustituir el campo “He recibido” del formulario de detalle de entrega de un EPI por un campo donde el usuario pueda firmar, ya sea con el ratón como con un aparato externo.
- Permisos en la funcionalidad de entrega de EPIS:
 - Además de los permisos propios de los formularios característicos de esta funcionalidad, crear permiso de lectura “Mis entregas” para que el usuario activo solo pueda ver las líneas donde él sea el trabajador. Este permiso permitirá firmar al trabajador pero no podrá editar ningún campo más.
 - Como añadido a la mejora anterior, quitar el permiso de escritura genérico para poder editar la firma
- En el dossier del empleado de la aplicación web:
 - Actualizar el cálculo de entrega de EPIs en dossier del empleado, tanto en la pestaña de EPIs como para las columnas de las pestañas de lugares y puestos
 - Añadir columna fecha reposición en la pestaña de EPIs del dossier del empleado. Que aparezca entre producto y centro de trabajo
 - Actualizar las entidades de consultas: EPIs del trabajador, puestos del trabajador y lugares del trabajador

Las mejoras sobre la aplicación web serían:

- Añadir permisos a toda la aplicación móvil. Lo ideal sería que estos permisos fueran los mismos que en la aplicación web.
- Sustituir la firma en el móvil por un mecanismo avalado por los abogados de la empresa, que justifique su legalidad. Algunas de las opciones barajadas son: la compra de un plugin específico, que además de capturar la firma guarde datos biométricos del firmante, la firma mediante huella digital, y un mecanismo de verificación en dos pasos, que envíe al móvil del firmante una clave secreta que deberá introducir en la aplicación para que se le permita firmar.

10

Bibliografía

En este apartado se muestran las referencias bibliográficas más relevantes en relación al proyecto de fin de grado realizado.

- [1]. CyC.es. (Fecha). *Consultoría y Comunicaciones*. Recuperado el 20 de enero de 2018, de <https://www.cyc.es>
- [2]. CyC.es. (Fecha). *Unifikas*. Recuperado el 20 de enero de 2018, de <https://www.cyc.es/serviciosproductos/productos/unifikas.aspx>
- [3]. Charles Petzold. (2016). *Creating Mobile Apps with Xamarin.Forms*. Recuperado el 11 de enero de 2018, de https://download.microsoft.com/DOWNLOAD/7/8/8/788971A6-C4BB-43CA-91DC-557B8BE72928/MICROSOFT_PRESS_EBOOK_CREATINGMOBILEAPPSWITHXAMARINFORMS_PDF.PDF
- [4]. Microsoft.com. (2012). *The MVVM Pattern*. Recuperado el 11 de enero de 2018, de <https://msdn.microsoft.com/en-us/library/hh848246.aspx>
- [5]. VisualStudio.com. (Fecha). *VisualStudio*. Recuperado el 15 de febrero de 2018, de <https://www.visualstudio.com/es/vs/>
- [6]. Microsoft.com. (2018). *SQL Server Management Studio (SSMS)*. Recuperado el 15 de febrero de 2018, de <https://docs.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>
- [7]. Microsoft.com. (2013). *Microsoft Lync*. Recuperado el 15 de febrero de 2018, de <https://products.office.com/es-ar/microsoft-lync-2013>
- [8]. Google.com. (Fecha). *Drive*. Recuperado el 15 de febrero de 2018, de https://www.google.com/intl/es_ALL/drive/
- [9]. Microsoft.com. (Fecha). *Xamarin.Forms*. Recuperado el 11 de enero de 2018, de <https://www.xamarin.com/forms>
- [10]. Matthew Leibowitz. (2018). *SignaturePad*. Recuperado el 19 de marzo de 2018, de <https://github.com/xamarin/SignaturePad>
- [11]. RebeccaXam. (2017). *XamForms.Controls.Calendar*. Recuperado el 15 de febrero de 2018, de <https://www.nuget.org/packages/XamForms.Controls.Calendar/>
- [12]. Adam Pedley, Egor Bogatov. (2018). *Toasts.Forms.Plugin*. Recuperado el 15 de febrero de 2018, de <https://www.nuget.org/packages/Toasts.Forms.Plugin/>
- [13]. Kirill Lyubimov. (2016). *Rg.Plugins.Popup*. Recuperado el 15 de febrero de 2018, de <https://www.nuget.org/packages/Rg.Plugins.Popup/>
- [14]. Antonio Feregrino. (2017). *PopUps avanzados en Xamarin.Forms*. Recuperado el 5 de febrero de 2018, de <https://thatcsharpguy.com/post/popups-avanzados-xamarin-forms/>
- [15]. James Montemagno. (2018). *Xam.Plugin.Media*. Recuperado el 15 de febrero de 2018, de <https://www.nuget.org/packages/Xam.Plugin.Media/>

- [16]. Dávila Becerril Ángel Jesús. (2012). *Gestión de riesgos del proyecto*. Recuperado el 3 de marzo de 2018, de <https://uacm123.weebly.com/8-gestioacuten-de-los-riesgos-del-proyecto.html>
- [17]. Microsoft.com. (Fecha). *Diagrama de Casos de uso UML: Instrucciones*. Recuperado el 20 de enero de 2018, de <https://msdn.microsoft.com/es-es/library/dd409432.aspx>
- [18]. Microsoft.com. (Fecha). *Modelar casos de usuario*. Recuperado el 3 de febrero de 2018, de [https://msdn.microsoft.com/es-es/library/ff398064\(v=vs.120\).aspx](https://msdn.microsoft.com/es-es/library/ff398064(v=vs.120).aspx)
- [19]. Microsoft.com. (Fecha). *Diagrama de clases UML: Instrucciones*. Recuperado el 17 de enero de 2018, de <https://msdn.microsoft.com/es-es/library/dd409416.aspx>
- [20]. León Welicki. (Fecha). *Patrones de fabricación: Fábricas de objetos*. Recuperado el 19 de febrero de 2018, de <https://msdn.microsoft.com/es-es/library/bb972258.aspx>
- [21]. Microsoft.com. (Fecha). *Diagramas de secuencia UML: Instrucciones*. Recuperado el 20 de enero de 2018, de <https://msdn.microsoft.com/es-es/library/dd409389.aspx>
- [22]. Microsoft.com. (Fecha). *Introducción al servidor web (IIS)*. Recuperado el 5 de abril de 2018, de [https://msdn.microsoft.com/es-es/library/hh831725\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831725(v=ws.11).aspx)

11



Anexos

ACTA DE REUNIÓN CON LA EMPRESA

1. Datos sobre la reunión:

- **Fecha:** 11 de enero de 2018
- **Hora de inicio:** 9:30
- **Hora de finalización:** 10:30
- **Número de reunión:** 1

2. Planificación:

a. Orden del día:

- Presentación del proyecto

3. Desarrollo:

a. Asistentes:

- Carolina Elduayen
- Ania Berazaluce

b. Temas tratados:

- Estado actual de *Unifikas*
- Alcance del proyecto
- Hitos y fechas relevantes

4. Conclusión:

Actualmente, el proyecto *Unifikas* tiene una versión web y otra móvil, sin embargo, mientras que la versión web goza de un amplio abanico de funcionalidades para gestionar los riesgos de Medio Ambiente, Calidad y Laborales de las empresas, la aplicación móvil carece de muchas de estas funcionalidades.

Así pues, uno de los principales objetivos a corto plazo es contar con una aplicación móvil para gestionar las funcionalidades más solicitadas y utilizadas por los clientes.

De entre todas las futuras funcionalidades de las que se quiere dotar a la aplicación móvil, una de las más prioritarias es el control de accesos de personal externo a las instalaciones de la empresa. Esta será la primera tarea a realizar por la alumna, que además le servirá para empezar a familiarizarse con el desarrollo de aplicaciones móviles.

Aunque todavía no hay fecha fijada para que ésta funcionalidad esté en funcionamiento para los clientes, se estima que debería de completarse antes de verano.

Por último, se ha hecho una pequeña introducción sobre el responsable de movilidad de la empresa, quien guiará y ayudará a la alumna con la parte de desarrollo en dispositivos móviles del proyecto.

ACTA DE REUNIÓN CON LA EMPRESA

5. Datos sobre la reunión:

- **Fecha:** 23 de febrero de 2018
- **Hora de inicio:** 13:00
- **Hora de finalización:** 13:30
- **Número de reunión:** 3

6. Planificación:

a. Orden del día:

- Mostrar el progreso hasta el momento y recibir las mejoras a llevar a cabo sobre el trabajo realizado
- Marcar las prioridades y acotar el alcance de la siguiente iteración

7. Desarrollo:

a. Asistentes:

- Carolina Elduayen
- Ania Berazaluce

b. Temas tratados:

- Correcciones y posibles mejoras del trabajo realizado hasta el momento
- Fijar el alcance de la siguiente iteración

8. Conclusión:

Tras mostrar el funcionamiento de la funcionalidad de control de accesos sobre la aplicación móvil, se han concretado un par de cambios a realizar antes de comenzar con la segunda funcionalidad a implementar:

- Se cambiará la apariencia de la ficha de trabajador, de manera que se incluyan los avisos si el periodo de contratación del pedido con el que está relacionado está fuera de fechas o si el trabajador está inactivo para dicho pedido.
- Se añadirá el campo de "Clasificación" en el buscador de organizaciones.

Tras estas modificaciones se comenzará a trabajar en la aplicación web, coordinándose con el equipo de trabajo de *Unifikas*, para desarrollar la funcionalidad de entrega de EPIS.

ACTA DE REUNIÓN CON LA EMPRESA

9. Datos sobre la reunión:

- **Fecha:** 25 de mayo de 2018
- **Hora de inicio:** 13:00
- **Hora de finalización:** 14:00
- **Número de reunión:** 7

10. Planificación:

a. Orden del día:

- Fin del proyecto

11. Desarrollo:

a. Asistentes:

- Carolina Elduayen
- Ania Berazaluce

b. Temas tratados:

- Mostrar el resultado final obtenido tras la realización del proyecto
- Obtener la conformidad/no conformidad de la empresa con el trabajo realizado
- Puesta en común de las futuras mejoras

12. Conclusión:

La jefa de proyecto ha mostrado su satisfacción con el trabajo realizado y se han puesto en común las futuras mejoras a efectuar en un futuro sobre el trabajo realizado, algunas de ellas son:

- Dotar a la aplicación móvil de permisos específicos para los distintos roles que se pueden dar a los usuarios de la aplicación.
- Profundizar en el estudio de las posibilidades de firma para la entrega de EPIs, hablando con los abogados de la empresa para verificar su legalidad.
- Añadir nuevos permisos a la funcionalidad de entrega de EPIs en la web, para que añadir, modificar y eliminar entregas sea más restrictivo.

ACTA DE REUNIÓN CON EL TUTOR

13. Datos sobre la reunión:

- **Fecha:** 27 de febrero de 2018
- **Hora de inicio:** 16:30
- **Hora de finalización:** 17:00
- **Número de reunión:** 1

14. Planificación:

a. Orden del día:

- Analizar la viabilidad del proyecto.
- Mostrar la primera de las funcionalidades implementadas para la empresa.
- Revisar la documentación realizada hasta la fecha, recibiendo las correcciones del último borrador de la memoria enviado al tutor.
- Planificar el trabajo a realizar durante las siguientes semanas

15. Desarrollo:

a. Asistentes:

- J. Ángel Vadillo
- Ania Berazaluce

b. Temas tratados:

- Ver la primera de las funcionalidades implementada
- Corrección del último borrador de la memoria enviado al tutor, cómo se ha enfocado el planteamiento inicial del proyecto y los objetivos del mismo.
- Guiar el trabajo a realizar durante las siguientes semanas.
- Análisis de viabilidad del proyecto.

16. Conclusión:

El enfoque y la implementación realizados hasta el momento son del agrado del tutor. Se han resaltado los puntos a modificar y las posibles mejoras sobre la documentación previamente entregada, ésta incluía los antecedentes y el planteamiento inicial del proyecto.

Se ha decidido comenzar a documentar la captura de requisitos, los modelos de casos de uso serán la primera de las tareas a realizar, ya que facilitarán tanto la implementación como las pruebas del proyecto. Aunque ya se hizo un primer borrador del modelo de casos de uso de la funcionalidad implementada, hay que explicarlo mejor, incluyendo interfaces para ilustrar sus procesos.

Teniendo en cuenta el estado actual del proyecto y la fecha de finalización del mismo, terminamos con la conclusión de que el proyecto es viable y se podrá presentar en plazo.

ACTA DE REUNIÓN CON EL TUTOR

17. Datos sobre la reunión:

- **Fecha:** 10 de mayo de 2018
- **Hora de inicio:** 16:30
- **Hora de finalización:** 17:00
- **Número de reunión:** 3

18. Planificación:

a. Orden del día:

- Analizar la viabilidad del proyecto.
- Revisar la documentación realizada hasta la fecha, recibiendo las correcciones del último borrador de la memoria enviado al tutor.
- Contrastar la ejecución de los modelos de dominio de la memoria
- Planificar el trabajo a realizar durante las siguientes semanas

19. Desarrollo:

a. Asistentes:

- J. Ángel Vadillo
- Ania Berazaluze

b. Temas tratados:

- Corrección del último borrador de la memoria enviado al tutor, poniendo especial énfasis en la mejor manera de realizar los modelos de dominio y cómo presentar los casos de uso para la aplicación web.
- Marcar las pautas para describir los requisitos no funcionales del proyecto. Además de guiar el trabajo a realizar durante las siguientes semanas.
- Análisis de viabilidad del proyecto.

20. Conclusión:

Con algunas excepciones, la documentación realizada hasta el momento es correcta, además, se han resaltado los puntos a modificar y las posibles mejoras.

Se ha decidido realizar únicamente un par de diagramas de secuencia por funcionalidad, de manera que ilustren el proceso sin recargar la documentación.

Respecto a las siguientes semanas, además de seguir con la implementación del proyecto, se documentarán los requisitos no funcionales, se finalizará el análisis y se empezará a documentar el desarrollo del proyecto.

Teniendo en cuenta el estado actual del proyecto y la fecha de finalización del mismo, terminamos con la conclusión de que el proyecto es viable y se podrá presentar en plazo.