



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona



Desarrollo de una aplicación para conocer el ciclo de vida de una colección bajo los principios DevOps

INGENIERÍA DEL SOFTWARE

ENTREGA 4: DOCUMENTO FINAL

MANGO

Claudia Sánchez Pérez

Director: Alberto Gutiérrez

Ponente: Claudia Ayala

Octubre 2020

Resumen

La necesidad de una herramienta visual e intuitiva con la que se logre estandarizar los datos relacionados con la creación de una colección de moda es una de las prioridades en Mango. Es por ese motivo, que nace la aplicación *Collection Grid*, para estandarizar y mejorar la calidad del dato, así como para poder consultar las distintas etapas que necesita una colección de ropa desde que se plantea hasta que llega a las tiendas. Estas etapas son: *Briefing*, Diseño, Compra y Producción, Entrega y Expedición y, por último, Exposición y Venta. Nuestro objetivo final será poder ofrecerle a los usuarios una herramienta con la que podrán consultar los detalles de cada una de las anteriores etapas.

Resum

La necessitat d'una eina visual i intuïtiva amb la que s'aconsegueixi estandarditzar les dades relacionades amb la creació d'una col·lecció de moda és una de les prioritats a Mango. És per aquest motiu, que neix l'aplicació *Collection Grid*, per estandarditzar i millorar la qualitat de la dada, així com per poder consultar les diferents etapes que necessita una col·lecció de roba des de que es planteja fins que arriba a les botigues. Aquestes etapes són: *Briefing*, Disseny, Compra i Producció, Lliurament i Expedició i, finalment, Exposició i Venda. El nostre objectiu final serà, doncs, poder oferir als usuaris una eina amb la qual podran consultar els detalls de cadascuna de les anteriors etapes.

Abstract

The need for a visual and intuitive tool with which to standardize the data related to the creation of a fashion collection is one of Mango's priorities. It is for this reason that the *Collection Grid* application was born, to standardize and improve the quality of the data, as well as to be able to consult the different stages that a clothing collection needs from its creation to its arrival in stores. These stages are: Briefing, Design, Purchase and Production, Delivery and Expedition and, finally, Exhibition and Sale. Our final objective will be to be able to offer users a tool with which they can consult the details of each of the previous stages.

Índice general

1. Introducción	1
1.1. Introducción	1
1.2. Identificación del problema	1
1.3. Definiciones	2
1.4. <i>Stakeholders</i>	3
2. Justificación	5
3. Alcance	7
3.1. Objetivos	7
3.2. Requisitos	7
3.2.1. Requisitos funcionales	7
3.2.2. Requisitos no funcionales	8
3.3. Riesgos	8
3.3.1. Bugs	8
3.3.2. Bloqueos	9
3.3.3. Desviaciones en el planning	9
3.3.4. Plazos de entrega ajustados	9
3.3.5. Confinamiento y posibles ERTE	9
4. Metodología y rigor	10
4.1. Metodología de trabajo	10
4.2. Metodología de desarrollo	12
5. Descripción de las tareas	14
5.1. Introducción	14
5.2. Tareas	14
5.2.1. Tareas de gestión del proyecto	14
5.2.2. Tareas de desarrollo	15
5.3. Recursos	17
6. Estimaciones y Gantt	18
7. Gestión del riesgo: Planes alternativos y obstáculos	20
8. Presupuesto	22
8.1. Identificación de los costes	22
8.1.1. Recursos humanos	22
8.1.2. <i>Hardware</i>	23
8.1.3. <i>Software</i>	24
8.1.4. Gastos generales	24
8.2. Estimación de los costes	25
8.3. Control de gestión	25

9. Informe de sostenibilidad	26
9.1. Autoevaluación	26
9.2. Dimensión Económica	26
9.3. Dimensión Ambiental	27
9.4. Dimensión Social	27

Índice de figuras

1.	<i>Proceso que sigue para la creación y venta de una colección de moda. Fuente: Mango</i>	2
2.	<i>Marco de trabajo de Scrum. Fuente: [16]</i>	10
3.	<i>Fases que se llevan a cabo en DevOps. Fuente: [26]</i>	12
4.	<i>Diagrama de ramas de Gitflow. Fuente: [29]</i>	12
5.	<i>Diagrama de Gantt para las tareas propuestas sin tener en cuenta las reuniones de equipo. Fuente: Elaboración propia</i>	19

Índice de tablas

1.	<i>Tabla DAFO para el proyecto. Fuente: Elaboración propia</i>	6
2.	<i>Tablas resumen de las tareas Fuente: Elaboración propia</i>	18
3.	<i>Tabla con los costes estimados por rol y hora. Fuente: Elaboración propia</i>	22
4.	<i>Tabla con los costes estimados de las tareas y reuniones. Fuente: Elaboración propia</i>	23
5.	<i>Tabla con los costes estimados para el hardware. Fuente: Elaboración propia</i>	23
6.	<i>Tabla con los costes estimados para el software. Fuente: Elaboración propia</i>	24
7.	<i>Tabla con los costes estimados para los gastos generales. Fuente: Elaboración propia</i>	24
8.	<i>Tabla con los costes estimados para los imprevistos. Fuente: Elaboración propia</i>	25
9.	<i>Tabla con los costes estimados del proyecto. Fuente: Elaboración propia</i>	25

Capítulo 1

Introducción

1.1. Introducción

Mi proyecto *Desarrollo de una aplicación para conocer el ciclo de vida de una colección bajo los principios DevOps* se trata de un trabajo de fin de grado para la Facultad de Informática de Barcelona (FIB), concretamente para el grado de ingeniería informática y la especialidad de ingeniería del *software*. Este trabajo será en colaboración con Mango, una empresa multinacional de moda, fundada en 1984 en España y con la sede en Palau-Solità i Plegamans, la cual facturó más de 2374 millones de euros en 2019. Además se diseñan más de 18000 prendas anualmente usando un 44 % de fibras de origen natural [1].

Esta multinacional está presente en más de 110 países con más de 2.700 tiendas. Además, tiene un equipo de más de 15000 empleados. Actualmente, Mango está dividida en 4 líneas: *Woman*, *Man*, *Violeta* y *Kids*, siendo *Woman* la línea con más peso y que mayores beneficios genera. Sin embargo, su mayor competidor en el mercado es *Inditex*, también es multinacional española, aunque la marca rival por excelencia es *Zara*.

Mango es una empresa comprometida con el medio ambiente y con la responsabilidad social, por lo que recientemente, ha publicado una lista con las fábricas de producción de sus prendas. Además dispone de la colección *COMMITTED*, una colección diseñada con criterios sostenibles que contribuyen a reducir su impacto ambiental. En 2019 se marcó los siguientes objetivos para lograr ser más sostenibles: usar el 100 % de algodón sostenible antes del 2025, utilizar el 50 % de poliéster reciclado, también, antes del 2025 y utilizar el 100 % de fibras celulósicas de origen controlado antes del 2030.

1.2. Identificación del problema

Antes de empezar con este proyecto, *Collection Grid*, el equipo del que formo parte y yo, estábamos desarrollando otro proyecto, *Visual Merchandising*, para solventar el problema que tenían los *merchandisers* de la compañía.

La idea de realizar este proyecto nace con la intención de digitalizar su proceso de trabajo y así facilitarles el día a día, de manera que los usuarios puedan poner foco en lo que realmente aportan valor: definir cómo exponer las prendas en tienda. Antes de tener la aplicación, utilizaban documentos de Excel tanto para consultar datos como para comunicarse con las tiendas, haciendo que fuera muy poco visual y estática la manera en la que se podían consultar los datos de los que se disponen.

Cuando la aplicación empezó a estar activa y los usuarios pudieron empezar a hacer uso de ella, otros departamentos de la empresa empezaron a interesarse por poder tener acceso, debido a que les resolvía el mismo problema de información no actualizada y poco accesible e intuitiva en informes Excel. Además, la situación actual en la que nos encontramos (COVID-19) y con la mayoría de departamentos teletrabajando, ha hecho que la compañía considere que se debe priorizar la digitalización de todos los procesos de trabajo por encima de otros proyectos y esta herramienta puede solucionar esos problemas.

Los nuevos usuarios de la aplicación tenían el mismo problema que tenían los *stakeholders* iniciales, una falta de estandarización de datos. La situación actual hace que estos usuarios se encuentren con más dificultades para poder obtener los datos y compartirlos, ya que están trabajando desde casa y, en ocasiones los datos los comparten de manera física. Por lo tanto, les está costando más realizar su trabajo de manera eficiente.

Actualmente, trabajan con documentos Excel que obtienen a partir de cruzar tablas filtrando los datos que necesitan y luego los comparten vía email, haciendo que sea difícil el mantenimiento y la compartición de estos datos.

Esta nueva aplicación, *Collection Grid*, pretende solucionar el problema de los nuevos *stakeholders*, así como tener una herramienta en la empresa donde poder consultar los distintos procesos que se siguen desde que se empieza a plantear una colección hasta que llega a las tiendas, bajo los principios *DevOps*. Más adelante, en la metodología de desarrollo explicaremos en qué consisten. De esta manera, podemos estandarizar los datos clave que se necesitan en los distintos estados de una colección, siendo el *visual merchandising* uno de los puntos clave antes de que las prendas de la nueva colección lleguen a tienda.

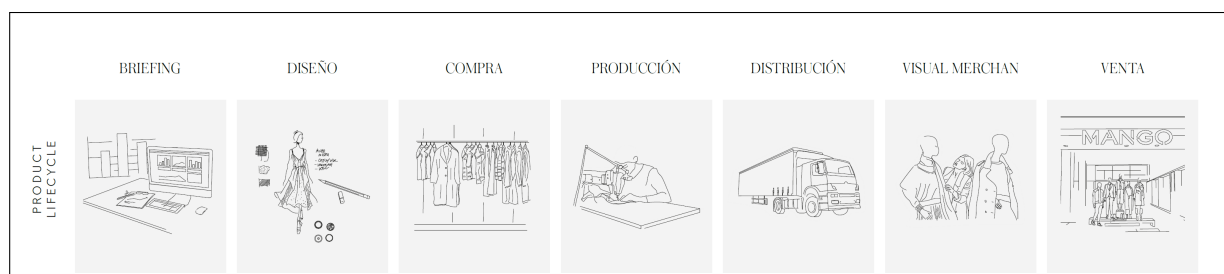


Figura 1: *Proceso que sigue para la creación y venta de una colección de moda. Fuente: Mango*

1.3. Definiciones

a) Colección de moda

Una colección de moda es un conjunto de prendas creadas por uno o varios diseñadores con una coherencia interna, siguiendo las tendencias o no, y está destinada a una temporada del año en concreto [2].

b) *Visual merchandising*

Aplicación con la que los *merchandisers* de la compañía pueden organizar la manera en la que exponer las distintas prendas a lo largo de una tienda, y las tiendas pueden consultarla para materializar la distribución de las prendas según las decisiones tomadas por los *merchandisers*. A partir de ahora cuando nos refiramos a esta aplicación, lo haremos de manera abreviada usando el término **Visual** o **VM**. Cuando nos refiramos a los *merchandisers* lo haremos usando el término **merchands**.

c) **Collection grid**

Aplicación con la que se permitirá visualizar la colección en los diferentes momentos de la evolución del producto, adaptando la información estratégica para cada momento. Será una aplicación que englobará distintas aplicaciones, es decir, será una suite de aplicaciones, para poder consultar cualquiera de las etapas de manera más cómoda desde un mismo lugar.

d) **Fase Briefing**

Primera fase de la colección. Se plantea la estructura, presupuesto, tendencias que se usarán, entre otros factores para la siguiente colección.

e) **Fase Diseño**

Fase en la que se elaboran las especificaciones técnicas de las prendas, se hacen los primeros prototipos y se hace la propuesta de la colección de manera interna.

f) **Fase Compra y Producción**

Fase en la que se seleccionan las prendas definitivas que se comprarán. Se negocia con los proveedores y se aprueban tejidos, fornituras y *fitting*. Una vez se ha dado el visto bueno por parte de la empresa, los proveedores empiezan con la fabricación de las cantidades aprobadas y desde las oficinas se hace un seguimiento.

g) **Fase Entrega y Expedición**

Fase en la que se hace el seguimiento de la entrega de los pedidos realizados a los proveedores y llegada a la aduana para garantizar que los tiempos de entrega en almacén se cumplen. Una vez llegan al almacén se organizan según el tipo de prenda que son. Después se organizan los pedidos para las distintas tiendas. Por último se preparan dichos pedidos para ser expedidos.

h) **Fase Exposición y Venta**

Fase en la que una vez se ha decidido una fecha en la que las prendas entran en tienda, el departamento de *visual merchandising* de central (oficinas de Palau) define la estrategia de exposición. Es en este punto en el que entra en juego la aplicación de la que ya disponen.

1.4. *Stakeholders*

Debido a la envergadura del proyecto, se van a ver implicados muchos actores. A continuación se listan aquellas personas que se beneficiarán de la app, interesadas, afectadas o participantes en el desarrollo.

- **Departamento de estructura de colección:** Personas que se beneficiarán de esta nueva herramienta. Son las personas que se encargan de asistir a desfiles, de inspirarse, de recopilar inputs de otros equipos, etc.
- **Diseñadores:** Este grupo de personas, usarán la aplicación para poder consultar el plan establecido para una temporada y ver reflejado sus diseños vinculados a dicho plan.
- **Planners:** Personas que se beneficiarán de la aplicación, ya que son las encargadas de definir el plan de colección para una temporada desde el lado de la rentabilidad.
- **Compradores:** Personas que usarán la aplicación para poder hacer seguimiento de las prendas compradas y las propuestas pendientes de selección para compra.
- **Proveedores:** Nuestros proveedores se beneficiarán de esta nueva herramienta, ya que el flujo que se seguirá desde la compañía será más eficaz y unificado.
- **Departamento de distribución:** Usará la app para tener un mejor control de los procesos que se llevan a cabo para la distribución de las prendas.

- ***Departamento de GCP (Gestión Comercial del Producto):*** Se encarga de adaptar la colección y distribución a las necesidades de cada país, por tanto, hará un uso de la app similar al que hará el departamento de distribución.
- ***Departamento de logística:*** Utilizarán esta nueva herramienta para tener más visibilidad de las expediciones de producto a tienda a realizar en una fecha dada.
- ***Departamento de visual merchandising:*** Utilizarán la app para poder definir la guía base de exposición de las prendas a lo largo de una tienda.
- ***Tiendas:*** Gracias al trabajo realizado por los *merchands*, las personas de las tiendas encargadas de exponer las prendas, podrán hacerlo de una manera más cómoda y visual gracias a la app, ya que lo podrán visualizar desde allí.
- ***Clientes:*** Son parte de los *stakeholders* que se beneficiarán de esta herramienta, ya que gracias al trabajo realizado por los distintos departamentos de la compañía, podrán disfrutar de una mejor experiencia en las distintas tiendas Mango.
- ***Isak Andic:*** Al ser el propietario de la empresa, se beneficiará de la app a través de los beneficios que se generen después de que se lleve a cabo todo el proceso de diseño y venta de una colección.
- ***Desarrolladores:*** Serán los encargados de llevar a cabo el código de esta nueva herramienta.

Capítulo 2

Justificación

Desde hace un tiempo, el equipo de dirección de IT, tomó la decisión de apostar por intentar hacer el máximo número de proyectos posibles internamente, es decir, evitando así recurrir a la compra de *software* o productos ya existentes en el mercado, ya que de este modo se pueden cubrir las necesidades de los usuarios personalizando al detalle aquellos requisitos que con otras herramientas disponibles en el mercado no sería posible.

Así pues, cuando se decidió arrancar con este proyecto, se tomó la decisión de que sería una aplicación propia, es decir, no recurriríamos a ningún producto ya existente en el mercado y sería un equipo de IT de la compañía el que llevaría a cabo su desarrollo.

Esta decisión se debe a que la finalidad que tiene esta aplicación es controlar las distintas etapas que se llevan a cabo desde que se empieza a plantear una colección hasta que esta llega a tienda. Se tomó la iniciativa de desarrollar este proyecto debido a que los procesos que se necesitan son muy específicos, de manera que no existe ninguna herramienta que se adapte a esta necesidad. Por lo tanto, tal y como se menciona anteriormente, lo más lógico era que un equipo interno de IT fuera el que llevara a cabo su desarrollo con el propósito de adaptarse lo máximo posible a las necesidades de los usuarios, siempre siguiendo el *look and feel* (misma estética e interacción) de las distintas aplicaciones que hay disponibles en la compañía.

Debido a que se trata de un proyecto en el que se integrarán varias aplicaciones y herramientas ya existentes, las cuales cubren las necesidades de alguna de las etapas que se siguen en el proceso de creación de una colección y, aunque se tomara la decisión de que fuera un proyecto desarrollado por un equipo de la empresa, he hecho un pequeño análisis de mercado para ver si había alguna herramienta que pudiera encajar con el problema que se pretende solucionar con esta integración y nos ayudara a refinar los requisitos y pudiéramos considerar otros que no teníamos planteados en un principio.

Sin embargo, al tratarse de un problema tan específico, solamente he encontrado herramientas de diseño, ya sean de moda o más genéricas, para un uso más personal, como pueden ser: *Photoshop*, *Illustrator* o *CorelDraw*, entre otras [3]-[5], y herramientas de PLM (Product Lifecycle Management) [6] que se utilizan para gestionar la vida de un producto, es decir, desde el momento en el que se diseña, hasta el momento en el que se decide descatalogar. Actualmente ya contamos con un PLM (*Centric*) para gestionar algunos de los procesos de la colección. Pero con *Centric* solo se cubre el proceso de diseño, industrialización y parte de la compra. El resto de procesos se llevan a cabo en el ERP, *Enterprise Resource Planning* (Sistema de Planificación de Recursos Empresariales), (*Cobol* y *SAP*) y únicamente con el PLM no se tendría una foto completa de inicio a fin del proceso.

Además, he realizado un análisis DAFO (Debilidades, Amenazas, Fortaleza y Obstáculos), tabla 1, para ver de manera sintética estas características y determinar que la mejor opción es desarrollar nosotros el proyecto.

DEBILIDADES	AMENAZAS
Falta de conocimiento en los diferentes procesos que se llevan a cabo para lanzar una colección al mercado.	Confinamientos y ERTE.
Posibilidad de no llegar a entender a los usuarios con sus requerimientos.	Plazos de entrega ajustados.
	Bloqueos debidos a otros equipos.
FORTALEZAS	OPORTUNIDADES
Proyecto muy personalizado a las necesidades de la empresa y de los usuarios.	Posibilidad de expandir el proyecto para cubrir necesidades de otros equipos.
Disponemos de herramientas ya desarrolladas que se tendrán que integrar en esta nueva aplicación.	Mejorar el flujo de trabajo y comunicación de los usuarios.
Equipo de desarrollo experimentado y que conoce las tecnologías utilizadas en el desarrollo del proyecto.	

Cuadro 1: *Tabla DAFO para el proyecto. Fuente: Elaboración propia*

Para finalizar con este análisis, voy a nombrar algunos pros y contras para llevar a cabo este proyecto.

PROS:

- Podemos personalizar lo máximo posible las funcionalidades que nos quieren los usuarios.
- Podemos comunicarnos con los usuarios muy fácilmente, ya que utilizamos la misma herramienta de comunicación.
- Podemos reutilizar e integrar herramientas ya existentes y en uso por parte de los usuarios. Así que habrá funcionalidades de las que ya conocerán su funcionamiento.

CONTRAS:

- Es un proyecto de una envergadura considerable.
- Debido a la situación actual, *COVID-19*, el proyecto puede verse perjudicado.
- Debido a la necesidad de la aplicación, contamos con fechas de entrega ajustadas.

Es por todos los motivos expuestos anteriormente, junto con el pequeño análisis, con el que concluimos que la mejor opción es crear una herramienta única y propia para poder integrar y adaptar las herramientas de las que ya disponemos, así como añadir las funcionalidades que sean necesarias para cubrir con los requerimientos de nuestros usuarios y de la empresa.

Capítulo 3

Alcance

3.1. Objetivos

El principal objetivo de este proyecto es poder controlar las distintas etapas que se siguen en Mango para dar vida a una colección de moda. Como ya se ha mencionado anteriormente, los usuarios de esta herramienta trabajan, actualmente, con datos que no saben si están actualizados o no. Esto se debe al formato de los ficheros que utilizan y la herramienta con la que los consultan, por lo que hay una falta de consistencia en los datos y una falta de estandarización de ellos. Por lo tanto, como desarrolladores, nuestro deber es lograr construir una herramienta que les permita consultar y actualizar dichos datos para lograr una estandarización y consistencia de los datos. También, deberemos permitirles consultar las distintas etapas por las que se pasan para llevar a tienda una colección. Deberemos desarrollarlo de manera visual e intuitiva para facilitar el máximo posible su trabajo.

Sin embargo, nuestro objetivo está a alto nivel, si nos adentramos un poco más algunos de los objetivos principales de este proyecto son:

- Integrar las diferentes etapas para crear y vender una colección en un solo entorno, es decir, adaptar aplicaciones ya existentes y en uso, y crear las funcionalidades necesarias para cubrir todas las fases del proceso. Nuestro objetivo es poder diferenciarle, al usuario, de manera clara, cada una de las etapas clave para la creación de una colección
- Mejorar y unificar el flujo de datos para evitar redundancias e incongruencias. De esta manera lograremos una mejora en la calidad del dato.
- Mejorar la comunicación entre departamentos gracias a la consulta de las distintas etapas. De esta manera, los usuarios podrán tener más contexto sobre las distintas etapas y los procesos que se llevan a cabo.

3.2. Requisitos

3.2.1. Requisitos funcionales

Debido a la metodología de trabajo que usamos, más adelante exponemos cuál es, los requisitos funcionales que mencionamos a continuación, pueden ir variando en función del tiempo y de las necesidades de los usuarios.

- Poder cargar ficheros CSV (*comma-separated values*) [7] desde la herramienta para poder nutrir de datos la fase *Briefing* del proyecto y poder consultar estos datos de manera visual.

- Tener una herramienta única, es decir, integrar la aplicación de *Visual Merchandising* en la nueva herramienta para así no tener que tener las fases de *Briefing* y *VM* unificadas. De esta manera, lograremos tener un único login y disponer de un menú con el que poder navegar entre fases del proceso y aplicaciones de estas.
- Diseño de la funcionalidad *dashboard* [8]. Un dashboard no es nada más que un tablero donde se expresan gráficos a partir de unos datos disponibles. Con esta funcionalidad, pretendemos permitir hacer gráficos de los datos cargados y configurarlos dinámicamente según los atributos disponibles. De este modo, el usuario podrá analizar aquello que necesite en ese momento.
- Integrar propuestas de diseño aún no compradas en la aplicación.

3.2.2. Requisitos no funcionales

En cuanto a requisitos no funcionales o de calidad, nos centraremos, sobre todo, en las integraciones y mejora de la calidad de los datos. Los siguientes requisitos son una muestra.

- Integrar los datos del *briefing* en el PLM que usa la empresa. Definir la integración técnica entre la aplicación y el PLM, de manera que el PLM también tenga cargado los datos subidos a la app.
- Refactorizar procesos de carga de datos de producto. Analizar cómo se prevee que los datos a mostrar en la herramienta crezcan. Se tendrá que realizar una nueva infraestructura de integración de datos más robusta para evitar seguir nutriéndonos de los datos de aplicaciones más antiguas. De esta manera, nuestro proyecto podrá evolucionar sin sufrir grandes bloqueos debido a la obtención de datos.

3.3. Riesgos

A la hora de desarrollar un proyecto de software, aunque, aparentemente, parezca que no hay ningún riesgo visible, siempre nos encontramos con unos que son intrínsecos, como los que se mencionan a continuación.

3.3.1. Bugs

Cuando estamos llevando a cabo el desarrollo de algún tipo de software, uno de los problemas que más tiempo nos hacen perder son los *bugs*, es decir, errores en el código. Podemos encontrarnos *bugs* de varios tipos, algunos son más fáciles de detectar que otros.

Los más fáciles de detectar son lo que nos encontramos mientras estamos desarrollando una funcionalidad o antes de subirla a la *rama develop*, más adelante se explica como funcionan las ramas y la metodología que usamos. Por otro lado, los errores más difíciles de detectar son aquellos que aparecen una vez tenemos nuestra funcionalidad en producción y el usuario hace uso de ella, ya que no obtiene el resultado esperado y se puede deber a un fallo de código o a no haber entendido bien lo que se pretendía obtener con esa funcionalidad.

3.3.2. Bloqueos

Debido a la estructura de equipos que hay en la empresa, aunque sea uno el que se encargue del desarrollo de una aplicación, siempre se van a ver involucrados otros equipos indirectamente. Por lo tanto, durante el desarrollo, nos podemos encontrar, por ejemplo, con que necesitemos crear una base de datos o añadirle una columna a alguna tabla ya existente. Esto, nos supondrá tener que crear un ticket para solicitar al equipo correspondiente, que se encargue de realizar este cambio, y nos puede suponer un bloqueo de días.

Actualmente, trabajamos con tickets para poder controlar y registrar todas las incidencias de los distintos empleados. Un ticket no es nada más que una notificación al equipo pertinente para que se pueda registrar un seguimiento de la incidencia que debe completarse [9].

3.3.3. Desviaciones en el planning

Al trabajar en diferentes proyectos a la vez, puede ser que, en algún momento durante el desarrollo de esta aplicación, el equipo tenga que dar prioridad a alguno de los otros proyectos. Podemos encontrarnos estas desviaciones por varios motivos, como pueden ser: incidencias encontradas por alguno de los usuarios, priorización de nuevas funcionalidades en algún proyecto existente, creación de algún proyecto nuevo que urja su desarrollo, cambios en el equipo, entre otros factores.

3.3.4. Plazos de entrega ajustados

Teniendo en cuenta que el proyecto tiene una envergadura considerable y la metodología con la que trabajamos, es importante cumplir con las fechas que estén establecidas para que los usuarios puedan hacer uso de las funcionalidades entregadas, pero debido a posibles incidencias, *bugs* o bloqueos se pueden ver comprometidas las fechas de entrega o parte de las funcionalidades que formaban parte de la entrega.

3.3.5. Confinamiento y posibles ERTE

El confinamiento y un posible ERTE (Expediente de Regulación Temporal de Empleo) son una posibilidad que pueden suponer un riesgo para el proyecto. Aunque el confinamiento, aparentemente, parece que únicamente nos afecta en el lugar de trabajo, puede suponer un problema de comunicación entre compañeros y departamentos, ya que hay ocasiones en las que podemos encontrarnos con algún problema que presencialmente se solucionaría rápidamente.

En cuanto a si se produjera, de nuevo, un ERTE, esto supondría un posible cambio en el equipo, es decir, una reducción del número de miembros y, en el peor de los casos, una suspensión del proyecto. De todas maneras, este escenario es el menos probable, ya que se trata de un proyecto necesario para la empresa.

Capítulo 4

Metodología y rigor

4.1. Metodología de trabajo

Actualmente, existen diversas metodologías de trabajo, aunque no hay ninguna que sea la mejor [10] [11], ya que cada una está pensada para diferentes problemas y entornos de trabajo. Se tienen que tener en cuenta varios factores a la hora de decidir qué metodología se adapta mejor a las necesidades tanto del proyecto como del equipo. Es por eso, que para nuestro caso se tomó la decisión de seguir un *mindset Agile* [12], en concreto, el marco de trabajo *Scrum* [13]-[15].

Aunque se tomara la decisión de trabajar utilizando este marco de trabajo, el equipo ha ido adaptando aquello que mejor funcionaba de este para sacar adelante los proyectos. En la figura 2, se puede observar como funciona teóricamente.

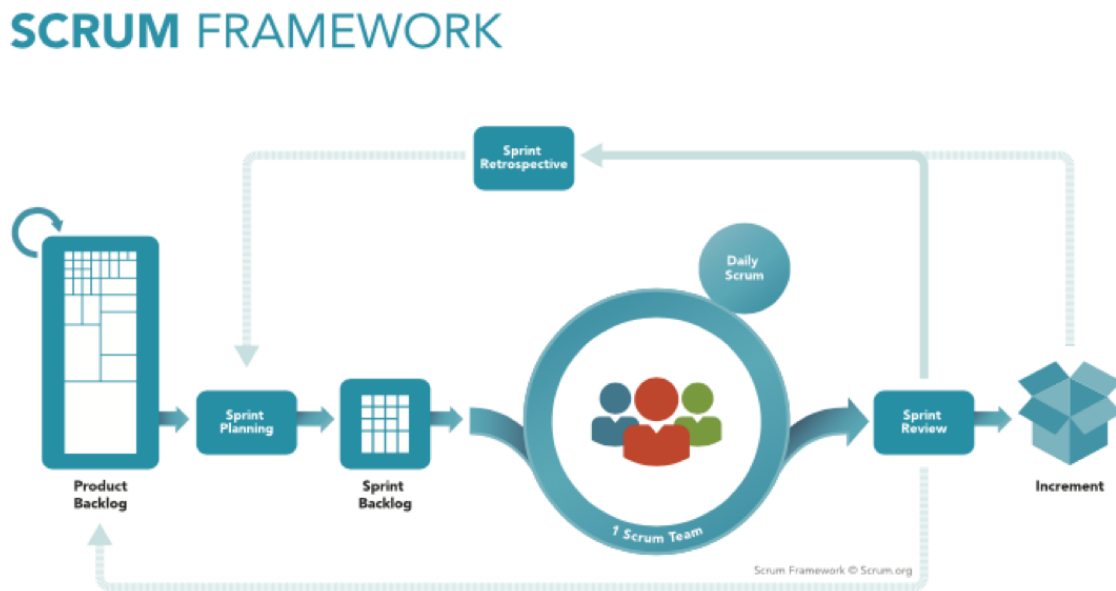


Figura 2: Marco de trabajo de Scrum. Fuente: [16]

Tal y como funciona este *framework*, tenemos que destacar cuatro aspectos clave que se explicarán seguidamente. El primero es el *backlog*, que no es nada más que la lista de tareas pendiente [17]. Hay dos tipos de tareas pendientes, las tareas de todo el proyecto y las tareas que se pretenden hacer durante un *sprint*, un período de tiempo fijo en el que el equipo *Scrum* trabaja para completar las tareas definidas para ese período de tiempo [18]. Nuestras tareas son historias de usuario [19], que son la unidad más pequeña con la que podemos trabajar y son una explicación general de un requerimiento o necesidad escrita desde la perspectiva de un usuario final.

El segundo aspecto es el equipo *Scrum* [20], donde destacan las figuras del *Product Owner* (PO), la figura del *Scrum Master* (SM) y, por último, la figura del equipo de desarrollo.

- **Product Owner:** Es la persona encargada de optimizar y maximizar el valor del producto, así como de priorizar este producto a través del *Product Backlog* (lista de tareas pendientes global).
- **Scrum Master:** Es la persona encargada de gestionar el proceso *Scrum* y de eliminar impedimentos para agilizar el desarrollo del producto. Actualmente no contamos con esta figura en el equipo.
- **Equipo de Desarrollo:** Equipo que suele estar formado entre 3 y 9 personas, en nuestro caso somos 4 personas dedicadas al proyecto. Es el encargado de desarrollar el producto.

El tercer aspecto a destacar son las ceremonias *Agile*, entre las que tenemos la *Sprint Planning*, la *Daily Scrum*, la *Sprint Review* y la *Sprint Retrospective* [21].

- **Sprint Planning:** Es una reunión que se suele hacer al inicio de cada *sprint* y tiene como finalidad preparar a todo el equipo para conseguir los objetivos fijados con éxito. Suele tener una duración de una hora.

En nuestro equipo nos reunimos cada dos semanas para iniciar un nuevo *sprint* y el día antes de esta reunión para hacer un refinamiento de las tareas que hay en el *backlog*, para así agilizar el planning.

- **Daily Scrum:** Es una reunión diaria de unos 15 minutos de duración en la que cada miembro del equipo informa del estado de sus tareas al resto del equipo intentando resolver estas tres preguntas:
 - ¿Qué completé ayer?
 - ¿En qué trabajaré yo?
 - ¿Estoy bloqueado con alguna cosa?

En nuestro equipo nos reunimos cada mañana para, así, empezar el día una visión general del estado de las tareas de cada compañero y poder ayudarle en caso de que se encuentre bloqueado.

- **Sprint Review:** Es una reunión que se realiza al finalizar un *sprint* o lograr un hito. El objetivo es revisar las iteraciones realizadas y exponer el trabajo realizado por el equipo.
- **Sprint Retrospective:** Es una reunión que se realiza al finalizar una iteración y suele tener una duración de una hora. El objetivo de esta reunión es analizar qué ha ido bien y qué ha ido mal para mejorar de cara al futuro.

Actualmente en el equipo no solemos hacer este tipo de reunión, ya que consideramos el que el equipo funciona correctamente. Sin embargo, cuando llevamos dos *sprints* o más sin que salga adelante lo que nos habíamos propuesto sí nos reunimos. De esta manera, podemos analizar qué no está funcionando.

El último aspecto que nos queda mencionar es el incremento. El incremento del producto es el objetivo de cada *sprint*, ya que es parte del producto terminado y listo para ser, potencialmente, lanzado al mercado o, en nuestro caso, puesto en producción para que los usuarios puedan disfrutar de estas nuevas funcionalidades. Gracias al incremento podemos adaptar el *Product Backlog* [22].

4.2. Metodología de desarrollo

La metodología de desarrollo que utilizaremos para esta aplicación es, como se puede ver en el título de este trabajo, *DevOps*. *DevOps* hace referencia a *Development*, desarrollo en inglés, y a *Operations*, operaciones en inglés, y se refiere a una metodología de desarrollo de *software* que tiene como objetivo mejorar la comunicación, colaboración e integración entre los desarrolladores y los equipos de sistemas que se encargan de las operaciones para lograr que un proyecto pueda estar en el entorno de producción y los usuarios puedan hacer uso de él [10] [23]-[25].

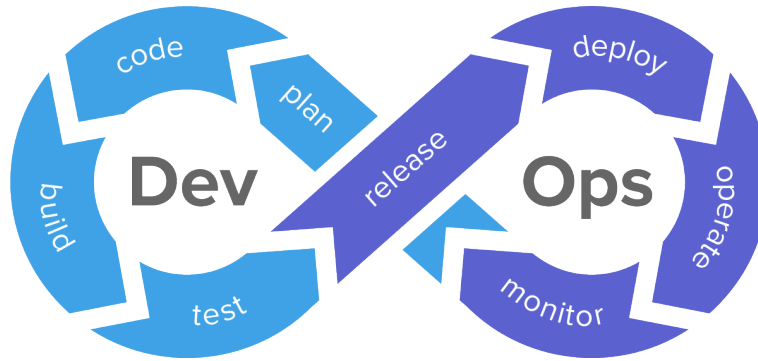


Figura 3: Fases que se llevan a cabo en DevOps. Fuente: [26]

En la figura 3 se pueden observar las distintas fases que forman parte de esta metodología. A continuación, se explica cada una de ellas, siguiendo su flujo natural, para entender como funcionan para nuestra empresa. Empezaremos por la fase de *plan*, ya que es la primera, y terminaremos con la fase de *monitor* [27] [28].

- A. **Plan:** Fase de planificación. En nuestro caso, como trabajamos por *sprints*, planificamos tareas para dos semanas usando una pizarra para poder controlar el estado de cada una de las tareas. Hay varias herramientas, pero nosotros utilizamos *Jira*.
- B. **Code:** Fase de codificación. Los desarrolladores realizan sus tareas de la pizarra, siguiendo el flujo de trabajo de *Gitflow*. Una vez terminadas, suben sus cambios a un repositorio de código fuente centralizado, en nuestro caso *Bitbucket*. En esta fase también desarrollamos los tests, ya sean unitarios, de integración o *e2e* (*end to end*, tests automáticos), que son imprescindibles para asegurar la calidad de nuestro código.

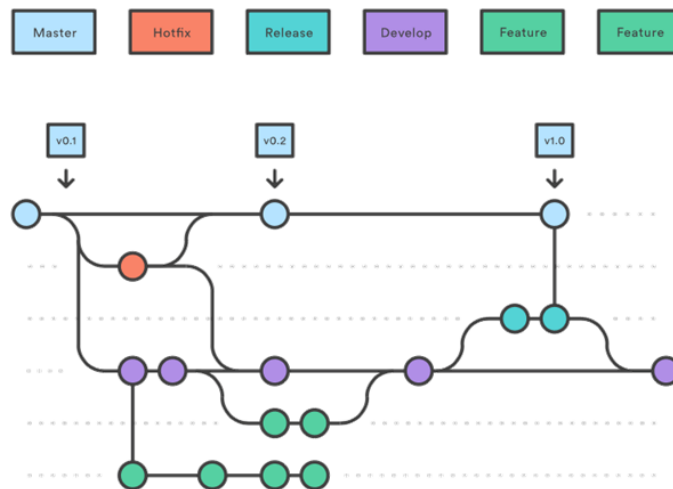


Figura 4: Diagrama de ramas de Gitflow. Fuente: [29]

El flujo de trabajo de *Gitflow* es un diseño de flujo de trabajo de *Git*. Es un modelo de basado en ramas diseñado entorno a la publicación del proyecto.

En lugar de tener una única rama *master* donde está todo el proyecto publicado, tenemos una rama llamada *develop*, y otras ramas de función, llamadas *feature* que nacen a partir de la *develop* y una vez la función está completada puede fusionarse a la rama *develop* de nuevo. También tenemos ramas de publicación, llamadas *release*. Se utilizan para poder subir la rama *develop* a *master* una vez *develop* tiene suficientes funciones para publicar. Estas ramas suelen ir acompañadas del número de versión correspondiente. Por último, tenemos las ramas de correcciones, llamadas *hotfix*, que se utilizan para reparar rápidamente las *releases* a producción. Son parecidas a las ramas *feature* excepto porque parten de la rama *master* y a la hora de fusionarlas, subirán a *master*. En la figura 4 podemos ver como funciona el sistema de ramas gráficamente.

- C. **Build:** Fase de construcción. Esta fase implica que se recupere el código de *Bitbucket* a través de una herramienta automatizada llamada *Jenkins*. Esta herramienta compila el código en un artefacto binario, ejecuta los tests que hay y, por último, publica el artefacto en el repositorio centralizado. Gracias a esto, logramos una integración continua (CI). La CI es la práctica en la que se integran los cambios del código en la rama principal del repositorio y se testan estos cambios de manera regular [30]. Además, la CI es garantizar que aquellos cambios que subes funcionan y no romperás el código existente gracias a los tests.
- D. **Test:** Fase de pruebas. Aunque se pasan pruebas en la fase anterior, los desarrolladores tendrán que realizar las pruebas pertinentes para asegurar el correcto funcionamiento del producto. Actualmente utilizamos *JUnit*.
- E. **Release:** Fase de *release*. Llegados a este punto, la *build* está lista para hacer el despliegue a producción. Para esta fase utilizamos *Jenkins*. Para crear la *release* usamos *Docker* y la guardamos en los repositorios de *sonatype*, en concreto, en *Nexus*.
- F. **Deploy:** Fase de despliegue. Gracias a utilizar esta metodología, conseguimos eliminar el proceso de tener que hacer manualmente el despliegue a producción de nuestro proyecto. Con esto logramos un despliegue continuo (CD), donde cada *build* que pasa todos los *checks* y el *pipeline* está balanceado se despliega automáticamente en producción. En nuestro caso, se coge el elemento de Nexus y se despliega en el orquestador, *Kubernetes*.
- G. **Operate:** Fase de operaciones. En esta fase ya tenemos en producción nuestro proyecto, por tanto, el equipo de operaciones tendrá que comprobar que todo funciona correctamente. En esta fase utilizamos *Kubernetes* para orquestar instancias que se encuentran en pequeños despliegues con microservicios.
- H. **Monitor:** Fase de monitoreo. En esta fase, tanto el equipo de desarrollo como el de operaciones, tendrán que tener un control del correcto funcionamiento de la aplicación en producción. Para ello, deberán monitorear continuamente la aplicación para comprobar el estado de los procesos que se llevan a cabo para su correcto funcionamiento. Actualmente, trabajamos con *ElasticSearch* y con *Kibana*.

Capítulo 5

Descripción de las tareas

5.1. Introducción

El proyecto arrancó a finales del mes de junio y todavía no tiene una fecha de finalización, es por ese motivo que he tomado la decisión de reflejar las tareas que voy a desarrollar entre el 1 de septiembre y el 15 enero, ya que la presentación final es a finales de ese mes. Por lo que, teniendo en cuenta que tengo una jornada laboral de 5 horas diarias, dedicaré aproximadamente 380 horas al desarrollo de este proyecto y cualquiera de sus herramientas implicadas. Si a esto le sumamos las horas dedicadas a la documentación del trabajo, así como a la preparación de la presentación final, unas 200 horas aproximadamente, tenemos que en total le dedicaré unas 580 horas a este proyecto.

5.2. Tareas

Al trabajar por *sprints*, como ya mencioné anteriormente, ha sido más difícil detallar las tareas, ya que lo hacemos en el *sprint planning* cada dos semanas y no tenemos el *Product Backlog* desgranado a nivel de tarea, sino que lo tenemos a nivel de historia de usuario. Es por ese motivo, que junto con mi PO (*Product Owner*), Núria Silva, nos hemos reunido para intentar desgranar las historias que tenemos pendientes hasta el mes de enero, para poder tener sus respectivas tareas. Sin embargo, a medida que avance el proyecto, y con ello, sus respectivos *sprints*, estas tareas podrán sufrir modificaciones, aparecer nuevas tareas e incluso desaparecer otras. Para cada tarea se detalla su descripción y se especifica el número de horas aproximada que necesitará, así como sus dependencias con otras tareas. A continuación, se detallan las tareas en función de si son de gestión del proyecto o de desarrollo. Para las tareas de desarrollo, se detallan en función de los hitos que hay marcados para la entrega de febrero de 2021.

5.2.1. Tareas de gestión del proyecto

- **TG1 ⇒ Alcance (25 h):** Redacción de la primera entrega de la gestión del proyecto, donde hay el contexto del proyecto, su justificación, el alcance y la metodología a usar.
- **TG2 ⇒ Planificación (9 h):** Redacción de la segunda entrega de la gestión del proyecto. En ella se encuentra la propuesta de planificación de tareas, su diagrama de Gantt y la gestión de los riesgos observados en la primera entrega. **Dependencias:** TG1.
- **TG3 ⇒ Gestión económica y sostenibilidad (10 h):** Redacción de la tercera entrega de la gestión del proyecto. En ella se encuentran los presupuestos y el informe de sostenibilidad. **Dependencias:** TG2.

- **TG4 ⇒ Documentación de la fase inicial (19 h):** Se recogen las tres entregas anteriores y se les aplican los cambios pertinentes. **Dependencias:** TG3.
- **TG5 ⇒ Documentación de la fase de seguimiento (30 h):** Se prepara la entrega y se hace una reunión con mi ponente, Claudia Ayala, para tener conocimiento del progreso y estado del proyecto. **Dependencias:** TG4.
- **TG6 ⇒ Documentación de la fase final (70 h):** Se redacta lo restante del trabajo. **Dependencias:** TG5.
- **TG7 ⇒ Preparación de la presentación final (36 h):** Preparación de la presentación del trabajo. **Dependencias:** TG6.
- **TG8 ⇒ Daily Scrum (15 min):** Cada mañana el equipo se reúne para poner en común el estado de las tareas. No hay dependencias.
- **TG9 ⇒ Refinamiento (1 h):** Cada dos semanas, al finalizar el *sprint* nos reunimos para refinar las historias de usuario y tener un poco encaminado el *sprint planning* del día siguiente. No hay dependencias.
- **TG10 ⇒ Sprint planning (1 h):** Cada dos semanas, nos reunimos para desglosar en tareas las historias de usuario que se pretenden realizar durante ese *sprint*. La **TG9** podría parecer una dependencia, pero se puede realizar el *sprint planning* sin necesidad de ella.

5.2.2. Tareas de desarrollo

Antes de empezar a describir las tareas que desarrollaré a lo largo de estos meses, hay que remarcar que cada tarea debe tener los tests unitarios pertinentes para asegurar el correcto funcionamiento de la funcionalidad que se ha desarrollado.

Carga de datos y consulta de la fase *Briefing*

Tareas relacionadas con la fase *Briefing* y sus datos.

- **CDB1 ⇒ Cambiar entre líneas de negocio (frontend) (15 h):** Necesitamos un menú con el que poder navegar entre líneas (*Woman, Man, Violeta y Kids*). De esta manera, podemos filtrar los datos de *Briefing* en función de la línea. **Dependencias:** No hay dependencias para desarrollar esta tarea.
- **CDB2 ⇒ Subida fichero Excel (frontend) (20 h):** Una vez tenemos las tablas de bases de datos pertinentes creadas y en el *backend* de nuestra aplicación tenemos los métodos necesarios creados, se desarrollan los métodos *frontend* para poder subir un fichero Excel con extensión csv con una sola hoja, pudiendo arrastrar este fichero desde la carpeta origen hasta la zona permitida para colgar el fichero. **Dependencias:** No hay dependencias para esta tarea, ya que cuando se llevó a cabo, las dependencias que había ya estaban resueltas.
- **CDB3 ⇒ Hacer las llamadas a backend (frontend) (5 h):** Para esta tarea, cuando se llevó a cabo, ya estaban todos los *endpoints* necesarios creados. El objetivo es dejar de mostrar *mock data* y empezar a mostrar los datos que se obtienen de hacer una carga Excel. **Dependencias:** CDB2.
- **CDB4 ⇒ Paginación en el modo galería (frontend) (2 h):** Con la estructura del listado en modo galería (se le da importancia a las imágenes por encima de otros atributos) debemos mostrar las páginas de productos disponibles para que el usuario tenga constancia de ello. Además, cuando cambias a modo listado clásico, la página se debe mantener. **Dependencias:** No hay, ya que ya estaban implementadas las tareas relacionadas con las estructuras de listado clásico y galería.

- **CDB5 ⇒ Redirigir a la vista principal cuando se clica en el logo y al enlace de *Collection* (frontend) (1 h):** **Dependencias:** No hay dependencias, ya que estaban desarrolladas las tareas que incluían el logo y el enlace de *Collection*.

Herramienta única

Tareas relacionadas en la primera integración de aplicaciones existentes en una sola.

- **HU1 ⇒ Unificar los menús de *Briefing* y *VM* (frontend) (20 h):** La tarea consiste en tener un único menú y poder navegar entre las dos apps. Para ello, cogeremos las funcionalidades que tenemos en el menú de *VM* junto con las de *Briefing* y las integraremos en el menú de *Collection Grid* adaptando el diseño según los *mockups* realizados por el equipo de *Experience Design*. **Dependencias:** No hay dependencias, ya que la única que podía haber es que no estuviera diseñado el *mockup*, pero ya lo tenemos disponible.
- **HU2 ⇒ Tener un login único (frontend) (30 h):** Dejar de utilizar el login para acceder a *VM*, ya que se podrá acceder desde la app de *Collection Grid*, donde se hará login, navegando por el menú. **Dependencias:** HU1
- **HU3 ⇒ Implementar buscador general (backend) (50 h):** Necesitaremos crear uno o varios *endpoints* para poder tener un buscador general en el que se podrán buscar productos (ya sea a partir de referencia comercial, de su nombre, su modelo fábrica...) y muebles (por nombre, tipo...). **Dependencias:** No hay, ya que ya disponemos del modelo de datos necesario.
- **HU4 ⇒ Implementar buscador general (frontend) (10 h):** En el menú principal necesitamos disponer de un buscador general para poder buscar cualquier dato relacionado con la app, utilizando los *endpoints* necesarios. **Dependencias:** HU1, HU3

Dashboard

Tareas relacionadas con el *dashboard* para poder analizar datos clave.

- **DB1 ⇒ Poder definir el diseño del gráfico (frontend) (35 h):** Tarea para definir el estilo de gráfico que se desea visualizar. **Dependencias:** No hay debido a que tenemos la estructura de datos definida, aunque habría una, que sería la referente al *mockup* de esta vista.
- **DB2 ⇒ Poder definir la estructura del gráfico en función de los atributos (frontend) (35 h):** Con esta tarea se pretende poder reorganizar los datos a analizar en los distintos gráficos y su jerarquía. **Dependencias:** DB1, DB3.
- **DB3 ⇒ Poder definir la estructura del gráfico en función de los atributos (backend) (45 h):** Se deben poder analizar los distintos datos que se muestran en un gráfico y poder determinar cuando no se pueden relacionar distintos tipos de datos entre ellos. **Dependencias:** No hay porque ya tenemos la estructura de datos definida, así como sus tablas de bases de datos.
- **DB4 ⇒ Consultar gráfico por atributo (frontend) (30 h):** Tarea para poder visualizar los gráficos según el atributo que se desee analizar. **Dependencias:** DB1, DB2, DB3.

Tareas de *Visual Merchandising*

- **TVM1 ⇒ Mostrar un *loader* cuando se está añadiendo un producto a un mueble mediante el escáner (40 h):** Debemos poder notificar de algún modo al usuario que se ha añadido un producto a un mueble, una vez se ha escaneado su código EAN (*European Article Number*). **Dependencias:** No hay porque los desarrollos previos necesarios ya están en producción.

5.3. Recursos

Seguidamente, menciono los recursos necesarios para que el desarrollo del proyecto sea exitoso.

- **Recursos humanos:** Yo [C] seré parte del equipo de desarrollo del proyecto. El equipo de desarrollo [ED] del que formo parte son una parte imprescindible para lograr el éxito. También el equipo de *Experience Design* [ExD] para poder consultar detalles de *mockups* y de maquetación. También necesitaremos a nuestra PO [PO], y en ocasiones necesitaremos a otros equipos como los de *Sistemas* [S] o *Comunicaciones* [Comms]. Periódicamente, mi director, Albeto Gutiérrez [AG] me irá dando *feedback* sobre mis progresos con el trabajo. Claudia Ayala [CA] también me irá dando soporte y ayudando cuando sea necesario.
- **Recursos materiales:** Aquello imprescindible para poder realizar el proyecto y las reuniones de equipo. Para ello, ya sea en casa o en la oficina, necesitaré una mesa, una silla, un monitor, un teclado y un ratón. Además, necesitaré mi portátil de la empresa para poder desarrollar. En cuanto a *software*, necesitaré:
 - [VS] **Visual Studio Code:** IDE para el *framework* de *Angular*.
 - [IJ] **IntelliJ IDEA:** IDE para *Java* y *Kotlin*.
 - [GIT] **Bitbucket, Git y SourceTree:** Herramientas para el control de versiones y repositorios.
 - [K8S] **Kubernetes:** Herramienta para poder automatizar subidas a producción de contenedores.
 - [J] **Jenkins:** Servidor de automatización para *CI*.
 - [OFF] **Office 365:** herramientas de *Microsoft* que incluyen: Outlook, Teams, Word, entre otras.
 - [TEX] **TeXstudio:** herramienta con la que redactar el trabajo.
 - [AT] **Atenea:** fuente de donde sacar los recursos necesarios para la redacción y desarrollo del trabajo.
 - [GOO] **Google Chrome:** Navegador que usaré tanto para buscar la información necesaria como para probar los desarrollos que lleve a cabo.
 - [S3] **AWS S3:** Servicio de *Amazon Web Services* para tener almacenados objetos. En nuestro proyecto, lo usaremos para tener imágenes y vídeos.
 - [GANTT] **Gantt Project:** Herramienta con la que crear el diagrama de Gantt.

Capítulo 6

Estimaciones y Gantt

En la tabla 2 se muestran resumidamente las tareas que llevaré a cabo.

ID	TAREA	HORAS	DEPENDENCIAS	RECURSOS
TG1	Alcance	25	-	C, PO, AG, CA, OFF, TEX, AT, GOO
TG2	Planificación	9	TG1	C, PO, AG, CA, GIT, OFF, TEX, AT, GOO
TG3	Gestión económica y sostenibilidad	10	TG2	C, PO, AG, CA, GIT, OFF, TEX, AT, GOO
TG4	Documentación de la fase inicial	19	TG3	C, PO, AG, CA, GIT, OFF, TEX, AT, GOO
TG5	Documentación de la fase de seguimiento	30	TG4	C, PO, AG, CA, GIT, OFF, TEX, AT, GOO
TG6	Documentación de la fase final	70	TG5	C, PO, AG, CA, GIT, OFF, TEX, AT, GOO
TG7	Preparación de la presentación final	36	TG6	C, PO, AG, CA, GIT, OFF, TEX, AT, GOO
TG8	Daily Scrum	0.25	-	C, ED, ExD, PO
TG9	Refinamiento	1	-	C, ED, ExD, PO
TG10	Sprint Planning	1	TG9	C, ED, ExD, PO
CDB1	Cambiar entre líneas de negocio	15	-	C, ED, ExD, PO, VS, GIT, K8S, J, OFF, GOO
CDB2	Subida fichero Excel (frontend)	20	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
CDB3	Hacer las llamadas a backend (frontend)	5	CDB2	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
CDB4	Paginación en el modo galería (frontend)	2	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
CDB5	Redirigir a la vista principal cuando se clica en el logo y al enlace de Collection (frontend)	1	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
HU1	Unificar los menús de Briefing y VM (frontend)	20	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
HU2	Tener un login único (frontend)	30	HU1	C, ED, ExD, PO, IT, VS, IJ, K8S, OFF, GOO
HU3	Implementar buscador general (backend)	50	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
HU4	Implementar buscador general (frontend)	10	HU1, HU3	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
DB1	Poder definir el diseño del gráfico (frontend)	35	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
DB2	Poder definir la estructura del gráfico en función de los atributos (frontend)	35	DB1, DB3	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
DB3	Poder definir la estructura del gráfico en función de los atributos (backend)	45	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
DB4	Consultar gráfico por atributo (frontend)	30	DB1, DB2, DB3	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
TVM1	Mostrar un loader cuando se está añadiendo un producto a un mueble mediante el escáner (frontend)	40	-	C, ED, ExD, PO, GIT, VS, IJ, K8S, OFF, GOO
TOTAL		579.5		

Cuadro 2: Tablas resumen de las tareas Fuente: Elaboración propia

En el diagrama siguiente, 6, podemos ver que cada color corresponde a un *sprint* distinto, excepto las tareas de documentación (TG1-TG7). Además, en este diagrama no se incluyen las reuniones de equipo, por ese motivo, no suma el total de horas planificado.

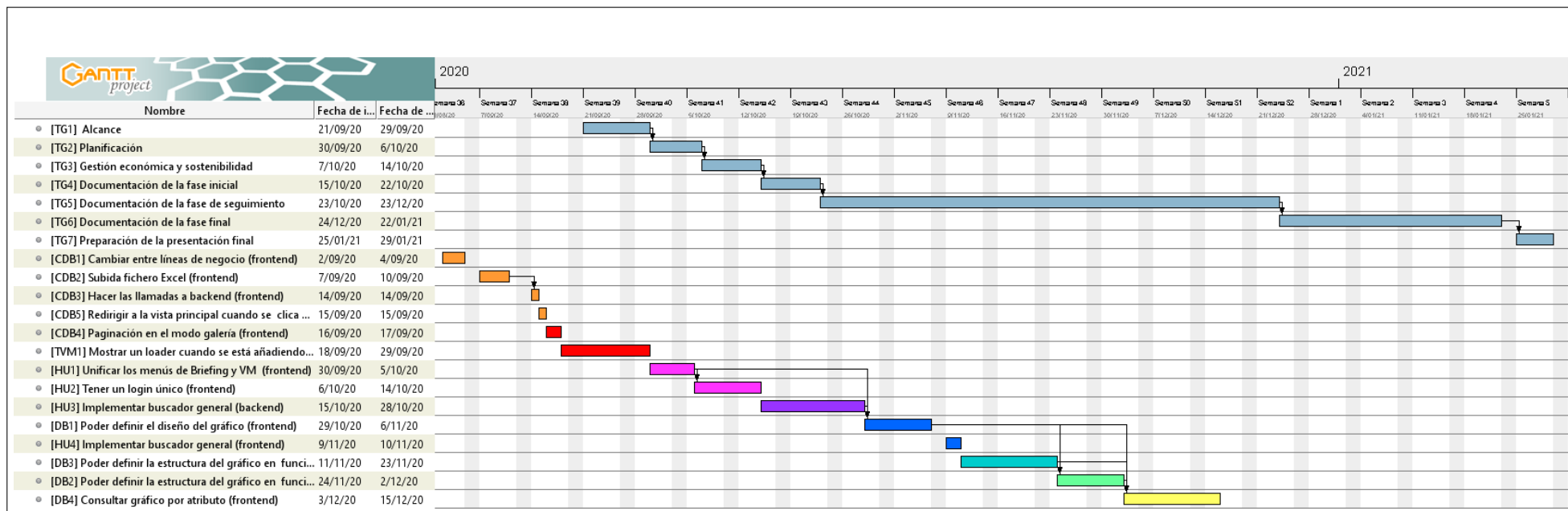


Figura 5: Diagrama de Gantt para las tareas propuestas sin tener en cuenta las reuniones de equipo. Fuente: Elaboración propia

Capítulo 7

Gestión del riesgo: Planes alternativos y obstáculos

Como ya habíamos visto anteriormente, el proyecto tiene unos riesgos que pueden retrasar la entrega prevista o, al menos parte de ella. Sin embargo, tenemos que proponer posibles soluciones por si se produjera alguno de estos riesgos.

Empezando con el riesgo con el que es más probable que nos encontremos, los *bugs* o errores en el código, ya que en prácticamente, todos los proyectos desarrollados en la historia hay alguno. Tenemos que tenerlos presentes e intentar minimizarlos al máximo haciendo tests unitarios tanto para el *frontend* como para el *backend*, practicando, en ocasiones, el *pair programming*, así como pruebas con usuarios para ver si el resultado obtenido es el que esperan. De esta manera, podemos asegurar una mejor calidad tanto en el código como en el producto final.

Otro riesgo que es posible que tengamos que solventar son los bloqueos. Tenemos bloqueos de dos tipos. El primero, son los que dependen de otros equipos. Son impredecibles de determinar los días que nos tendrán bloqueados. Por eso, lo máximo que podemos hacer, al no tener la figura del *Scrum Master*, es intentar apretar al equipo que le toque para que nos desbloquee aquello que nos impide seguir. El segundo tipo son los bloqueos producidos por necesidad de algún *software* del que no disponemos o dar de alta permisos para tener acceso a alguna aplicación o herramienta. Para eso, se tiene que abrir un ticket, para que nos concedan los permisos necesarios. Normalmente, este tipo de incidencia suele tardar unos tres días en resolverse.

Planificar un proyecto es una tarea complicada, ya que se tienen que tener en cuenta muchos factores. En nuestro caso, al usar una metodología ágil, es más difícil de cuadrar las fechas con las historias de usuario, así como con las tareas que necesitarán para cumplir con las necesidades de los usuarios, por lo que podemos tener unos plazos de entrega ajustados.

Actualmente tenemos una entrega para octubre donde en ella se pretende entregar la carga del fichero de datos, la visualización de estos ya sea en un listado como su detalle, la navegación entre la app de *VM* y *Briefing*. Siempre que no nos encontremos con ningún bloqueo o *bug* importante, o ninguna desviación en el planning debido a alguna incidencia en otro de los proyectos que tenemos activos, seremos capaces de cumplir con el hito propuesto. Para lidiar con las desviaciones, lo mejor que podemos hacer es intentar tener una buena calidad en nuestro código para evitar *bugs*.

Para finalizar, tenemos un riesgo que, hasta ahora, era impensable tener que considerar. Sin embargo, debido a la situación en la que vivimos, debemos tener en cuenta la posibilidad de que esto suceda, ya que podría causar un impacto importante en el desarrollo del proyecto. Por ejemplo, en caso de que se produjera un ERTe, muchos participantes del proyecto podrían perder temporalmente su puesto, hecho que causaría grandes retrasos. No obstante, al ser un proyecto priorizado por la empresa, es una casuística que es poco probable que ocurra o, al menos, es difícil que el equipo de desarrollo se vea afectado. En cuanto al confinamiento, es más probable que se dé, pero aunque la comunicación entre compañeros se pueda ver afectada, durante los últimos meses se ha demostrado que nos hemos podido adaptar adecuadamente y somos capaces de sobrellevar cualquier incidencia desde casa, por lo que no supondría un impacto muy negativo y es poco probable que este riesgo supusiera retrasos en las entregas.

Capítulo 8

Presupuesto

8.1. Identificación de los costes

Con el fin de poder identificar correctamente los costes que va a suponer este proyecto, tenemos que tener en consideración los siguientes factores: recursos humanos, *hardware* y *software* que se necesitan para llevarlo a cabo, así como gastos generales (luz, Internet, etc) e impuestos. Para poder calcular correctamente los costes que va a suponer, debemos tener en cuenta, también, las contingencias y los posibles imprevistos.

8.1.1. Recursos humanos

Los recursos humanos hacen referencia a los diferentes roles de empleados que se necesitan para desarrollar las tareas descritas en el diagrama de Gantt. Para poder planificar los costes que supondrán, tendré que dar una aproximación del coste por hora para los distintos roles. El equipo de desarrollo se divide entre desarrolladores *frontend* y *backend* seniors. Para los distintos roles de sistemas he hecho una media de los salarios por hora brutos para englobarlos. Estas aproximaciones las he hecho a partir de la media obtenida en la página *glassdoor* [31].

Rol	Salario/hora (bruto)	Salario/hora (bruto) + Seguridad Social (x1.3)
Desarrollador frontend senior	23.53€	30.59€
Desarrollar backend senior	20.79€	27.03€
Product Owner	24.78€	32.21€
Experience Designer	20.86€	27.12€
Sistemas	24.26€	31.54€
Desarrollador becario	9€	11.7€

Cuadro 3: Tabla con los costes estimados por rol y hora. Fuente: Elaboración propia

Una vez que tenemos definidos los costes por rol, en la tabla 3, podemos determinar de manera detallada el coste de cada tarea a partir del coste por hora, una vez calculado con la Seguridad Social. En la tabla 4 podemos observar el coste que supone cada tarea. En el apartado de recursos, 5.3, podemos encontrar el significado de cada rol (C, FE, BE, SIS, ExD, PO).

ID	TAREA	TOTAL HORAS	HORAS					COSTE (€)	
			C	FE	BE	SIS	ExD	PO	
TG1	Alcance	25	24.5					0.5	302.76
TG2	Planificación	9	8.5					0.5	115.56
TG3	Gestión económica y sostenibilidad	10	9.5					0.5	127.26
TG4	Documentación de la fase final	19	18.5					0.5	232.56
TG5	Documentación de la fase de seguimiento	30	29.5					0.5	361.26
TG6	Documentación de la fase final	70	69					1	839.51
TG7	Preparación de la presentación final	36	35					1	441.71
TG8	Daily Scrum	0.25(x92)	0.25	0.25	0.25		0.25	0.25	4284.21
TG9	Refinamiento	1(x10)	1	1	1		1	1	1862.7
TG10	Sprint Planning	1(x10)	1	1	1		1	1	1862.7
CDB1	Cambiar entre líneas de negocio	15	13.5	0.5			0.5	0.5	202.91
CDB2	Subida fichero Excel (frontend)	20	17.5	0.5		0.5	1	0.5	263.27
CDB3	Hacer las llamadas a backend (frontend)	5	4.5	0.5					67.95
CDB4	Paginación en el modo galería (frontend)	2	1.5	0.5					32.85
CDB5	Redirigir a la vista principal cuando se clica en el logo y al enlace de Collection (frontend)	1	0.9	0.1					13.59
HU1	Unificar los menús de Briefing y VM (frontend)	20	17.5	0.5			2		274.29
HU2	Tener un login único (frontend)	30	27	1	1	1			405.06
HU3	Implementar buscador general (backend)	50	49		1				600.33
HU4	Implementar buscador general (frontend)	10	9	0.5			0.5		134.16
DB1	Poder definir el diseño del gráfico (frontend)	35	33	1			1		443.81
DB2	Poder definir la estructura del gráfico en función de los atributos (frontend)	35	33	1			1		443.81
DB3	Poder definir la estructura del gráfico en función de los atributos (backend)	45	44		1				541.83
DB4	Consultar gráfico por atributo (frontend)	30	28	1			1		385.31
TVM1	Mostrar un loader cuando se está añadiendo un producto a un mueble mediante el escáner (frontend)	40	38	1			1		502.31
TOTAL		579.5							14731.71

Cuadro 4: Tabla con los costes estimados de las tareas y reuniones. Fuente: Elaboración propia

8.1.2. Hardware

Para poder llevar a cabo el desarrollo de la app requerimos de un material *hardware* básico. Este es el que se mencionó en el apartado de recursos. Este *hardware* necesario, tal y como Hacienda permite, lo vamos a amortizar en 4 años. Con el fin de poder hacer una aproximación del coste que nos va a suponer este *hardware* durante los meses que voy a participar en el desarrollo y teniendo en cuenta que este año hay unas 1760 horas laborables (220 días laborables), podemos calcular dicho coste. El cálculo para obtener las amortizaciones es: *(Coste hardware €/ (4 años * 220 días laborables al año * 5 horas dedicación al día)) * 92 días de dedicación al desarrollo del proyecto * 5 horas al día*. En la tabla 5 podemos observarlos.

HARDWARE	COSTE (€)	AMORTIZACIÓN (€)
Lenovo ThinkPad X270 (ordenador portátil [32])	1128.95	118.03
Logitech B100 (ratón [33])	4.99	0.53
Trust ClassicLine (teclado [34])	10.77	1.13
BenQ GW2280 (monitor [35])	99.99	10.46
TOTAL		130.15

Cuadro 5: Tabla con los costes estimados para el hardware. Fuente: Elaboración propia

8.1.3. Software

Siguiendo con los costes, a continuación se mencionan los costes ocasionados por el *software* utilizado en el día a día para el desarrollo del proyecto y la comunicación entre compañeros. En nuestro caso, al usar herramientas con licencias anuales o mensuales, no tenemos en cuenta la amortización para poder cambios de *software* obsoleto, donde Hacienda permite hacer cambios cada 2 años, ya que con las licencias podemos obtener actualizaciones. El cálculo para obtener las amortizaciones es: *(Coste software €/ (220 días laborables al año * 5 horas dedicación al día)) * 92 días de dedicación al desarrollo del proyecto * 5 horas al día*. En la tabla 6 podemos ver los costes asociados al *software*.

SOFTWARE	COSTE (€)	AMORTIZACIÓN (€)
Visual Studio Code	Gratuito	-
IntelliJ IDEA	Gratuito	-
Bitbucket	5602 anual /100 usuarios	23.43
Git	Gratuito	-
Sourcetree	Gratuito	-
Kubernetes	Gratuito	-
Jenkins	Gratuito	-
TeXstudio	Gratuito	-
Office 365	204.24 anual	85.41
Google Chrome	Gratuito	-
AWS S3	0.020 / GB	2.1
TOTAL		110.94

Cuadro 6: Tabla con los costes estimados para el software. Fuente: Elaboración propia

8.1.4. Gastos generales

Para finalizar con los costes, tenemos que tener en consideración otro tipo de costes, como pueden ser: el material de oficina necesario para poder llevar a cabo el proyecto, la luz, los gastos de Internet, el desplazamiento hasta el trabajo, entre otros. Para el material de oficina, tenemos que tener en cuenta que Hacienda lo permite amortizar en 20 años [36]. Puesto que actualmente voy a la oficina la mitad de los días y la planificación del proyecto es de 92 días, tendré que tener en consideración 46 días de desplazamientos. Para calcular los gastos de electricidad hay que tener en cuenta la potencia del portátil, unos 150W, y de un fluorescente, unos 18W, donde tendré en consideración, para calcular el coste, un par de fluorescentes. El cálculo para obtener las amortizaciones es: *(Coste software €/ (20 años amortización * 220 días laborables al año * 5 horas dedicación al día)) * 92 días de dedicación al desarrollo del proyecto * 5 horas al día*. En la tabla 7 podemos ver los costes de dichos recursos.

RECURSO	COSTE (€)	AMORTIZACIÓN (€)
Mesa oficina [37]	99	2.07
Silla oficina [38]	209.9	4.39
Desplazamiento (gasolina)	4.104 €/día	188.79
Electricidad	0.10352 €/kWh	8.86
Internet	55.99 €/mes	171.71
TOTAL		375.82

Cuadro 7: Tabla con los costes estimados para los gastos generales. Fuente: Elaboración propia

8.2. Estimación de los costes

Para finalizar con la estimación de los costes tenemos que tener en cuenta los imprevistos y la contingencia, que es del 15 % en este sector, que se traduce en **2287.98€**. En la tabla 8 podemos ver los costes asociados a los imprevistos con los que nos podemos encontrar durante el desarrollo del proyecto.

RIESGO	PROBABILIDAD	TIEMPO	COSTE ESTIMADO (€)	COSTE (€)
Bugs	55 %	30 horas	677.09	372.4
Bloqueos	15 %	10 horas	304.49	45.68
Desviaciones en el planning	10 %	10 horas	594.38	59.44
Plazos de entrega ajustados	10 %	10 horas	594.38	59.44
ERTE	5 %	50 horas	1425.82	71.3
TOTAL				608.26

Cuadro 8: *Tabla con los costes estimados para los imprevistos. Fuente: Elaboración propia*

En la tabla 9 podemos ver resumidos los distintos costes y el coste total estimado para el proyecto.

	COSTE (€)
Costes de personal por actividad (CPA)	14732
Costes Generales (CG)	617
Contingencia	2288
Imprevistos	609
TOTAL	18246

Cuadro 9: *Tabla con los costes estimados del proyecto. Fuente: Elaboración propia*

8.3. Control de gestión

En esta sección, se explica el control que se llevará sobre las posibles desviaciones en el presupuesto que nos podemos encontrar a lo largo del proyecto. Para ello, vamos a intentar anotar el tiempo real que nos ha llevado desarrollar cada tarea. De este modo, podemos saber la desviación que tenemos en nuestro presupuesto y si estamos dentro de lo que se ha presupuestado en un inicio. Con el fin de tener este control definimos las siguientes métricas:

- Desvío de mano de obra en precio (desvío en coste por tarifa) =

$$(\text{coste estimado} - \text{coste real}) * \text{consumo horas real}$$

- Desvío de mano de obra en consumo (desvío en eficiencia) =

$$(\text{consumo horas estimadas} - \text{consumo horas reales}) * \text{coste estimado}$$

Al finalizar el período del proyecto que voy a desarrollar, realizaremos los cálculos con el fin de obtener los desvíos. De este modo podremos hacer un análisis del presupuesto y determinar si, finalmente, tenemos que hacer uso o no de la contingencia calculada al inicio del proyecto.

Capítulo 9

Informe de sostenibilidad

9.1. Autoevaluación

A lo largo de estos años en la universidad, hemos ido dando pinceladas sobre sostenibilidad en distintas asignaturas. En alguna de ellas se ha hecho hincapié, sobre todo, en la dimensión ambiental con el impacto que dejan los residuos de *hardware* obsoletos y los conocidos cementerios tecnológicos en países subdesarrollados, donde aquí también podemos ver la dimensión social y el impacto que causan a la población de esos países.

En otra asignatura tuvimos que desarrollar un proyecto de cooperación, en nuestro caso, decidimos desarrollar una app con la que una persona que se encontrara en una situación de peligro pudiera solicitar ayuda a los otros usuarios de la app que se encontraran cerca. Con este ejemplo, podemos ver la dimensión social más en profundidad, ya que la finalidad del proyecto era mejorar la sociedad y aumentar la calidad de vida de las personas que hicieran uso de dicha aplicación.

La dimensión que me falta por comentar es la económica y, es de la que menos conozco. Esto se debe a que no me he visto involucrada en procesos de toma de decisiones económicas a la hora de arrancar un proyecto nuevo. Lo único que conozco es intentar rentabilizar lo máximo posible las horas de trabajo para poder sacar un proyecto adelante con el mínimo gasto posible.

Reflexionando sobre mis conocimientos sobre la sostenibilidad, me he dado cuenta de que aún me queda bastante por aprender sobre este tema, sobre todo en las dimensiones económicas y sociales, ya que creo que podría ser útil y enriquecedor para proyectos en el futuro. Sin embargo, me he dado cuenta de que las tres dimensiones están muy relacionadas entre ellas y si no tienes en cuenta una de las tres, esta puede verse afectada negativamente, causando un impacto negativo al proyecto. De modo que de cara al futuro sería útil tener en cuenta estas tres dimensiones para poder desarrollar un proyecto lo más sostenible posible, para lograr un mayor éxito.

9.2. Dimensión Económica

Referente a PPP: ¿Has estimado el coste de la realización del proyecto (recursos humanos y materiales)?

A la hora de analizar si un proyecto será viable o no, tenemos que tener en cuenta su presupuesto, ya que la empresa tiene que tomar la decisión de si vale la pena invertir en ese proyecto para mejorar sus procesos internos. En mi caso, la empresa quería desarrollar este proyecto porque existía una necesidad real, por lo que era necesario presupuestarlo. Como yo no formaba parte del equipo encargado del presupuesto, he realizado un presupuesto basándome en salarios y costes medios que he encontrado en Internet, así como de las tareas que voy a realizar y el tiempo que me llevarán desarrollarlas.

Referente a la Vida Útil: ¿Cómo se resuelve actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejorará económicamente tu solución a las existentes?

Como ya comenté anteriormente, antes de arrancar con este proyecto, los *stakeholders* trabajaban con documentos Excel y tenían una falta de estandarización en los datos que utilizaban. Por eso, en cuanto tuvieron la posibilidad de consultar la app de VM, nos dimos cuenta, tanto ellos como nosotros, el equipo de desarrollo y la PO, que necesitaban una herramienta más visual y de donde pudieran consultar los datos y estos estuvieran actualizados y estandarizados. En cuanto a la mejora económica, se verá reflejado en que los usuarios de esta app podrán optimizar y aumentar su eficiencia, ya que no habrá conflictos con los datos y el hecho de tener una herramienta visual facilita el trabajo que tienen que realizar.

9.3. Dimensión Ambiental

Referente a PPP: ¿Has estimado el impacto ambiental que tendrá la realización del proyecto? ¿Te has planteado minimizar el impacto, por ejemplo, reutilizando recursos?

A la hora de plantear el proyecto, no se planteó el impacto ambiental que este tendría. Cabe destacar que este proyecto nació en un momento de necesidad donde las condiciones no eran las más deseadas debido a la situación que se estaba viviendo (*COVID-19*), por lo que se planteó lo esencial para sacar el proyecto adelante. Sin embargo, de este proyecto podemos sacar en claro que se reducirá el uso de papel y tóner, ya que los usuarios dispondrán de una herramienta visual desde donde podrán consultar y actualizar datos.

Referente a la Vida Útil: ¿Cómo se resuelve actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejorará ambientalmente tu solución a las existentes?

Como ya he comentado, los *stakeholders* trabajan con documentos Excel, a menudo, estos documentos son impresos para su posterior análisis y consulta. Gracias a este nuevo proyecto, podremos reducir considerablemente, por no decir el 100 %, el consumo de papel y tóner que actualmente utilizan. De este modo, contribuimos a reducir los residuos tanto de papel como de tóner, siendo estos últimos muy perjudiciales para el medio ambiente.

9.4. Dimensión Social

Referente a PPP: ¿Qué crees que te va a aportar a nivel personal la realización de este proyecto?

A nivel personal creo que este proyecto me puede enriquecer en muchos sentidos, no solamente con conocimientos sobre el desarrollo de *software*, sino también en aprender cómo funciona un equipo de desarrollo en una empresa y cómo aplicar metodologías ágiles. También aprenderé a cómo llevar a cabo sesiones de demostraciones con los usuarios, donde en ellas se les presentan las nuevas funcionalidades y ellos deberán ser capaces de saber cómo utilizarlas. Si no fuera así, deberemos revisar dichas funcionalidades para ver si tenemos que replantearlas con el fin de que sean más intuitivas y usables. Además, gracias a este proyecto aprenderé los procesos que se llevan a cabo para desarrollar una colección de moda.

Referente a la Vida Útil: ¿Cómo se resuelve actualmente el problema que quieres abordar (estado del arte)? ¿En qué mejorará socialmente (calidad de vida) tu solución a las existentes?

Actualmente y como ya he comentado anteriormente, los usuarios utilizan documentos Excel y también consultan la app de VM para poder llevar a cabo su trabajo. Gracias a este proyecto los usuarios podrán utilizar datos estandarizados y que saben que están actualizados, así como, los usuarios clave, podrán ser capaces de actualizar dichos datos y dejarán de utilizar los documentos Excel como herramienta de consulta.

Referente a la Vida Útil: ¿Existe una necesidad real del proyecto?

Tal y como he comentado anteriormente, este proyecto nació de una necesidad real en la empresa. Los usuarios necesitaban poder consultar datos de manera continua, por lo que estos debían estar actualizados en todo momento. Gracias a la app de VM, se dieron cuenta de que necesitaban una herramienta igual de visual para poder trabajar de manera más eficiente. Además, gracias a este proyecto, más departamentos se van a ver beneficiados, ya que tendrán funcionalidades desarrolladas exclusivamente para ellos y podrán ser conscientes de los otros procesos por los que se pasan para desarrollar una colección.

Bibliografía

- [1] *Prendas de Mujer 2020 — Mango España*. URL: https://shop.mango.com/iframe.faces?state=she.001_ES&ts=1600959954312 (visitado 22-09-2020).
- [2] *Colección (moda) - Wikipedia, la enciclopedia libre*. URL: [https://es.wikipedia.org/wiki/Colecci%7B%5C'%7Bo%7D%7Dn_\(moda\)](https://es.wikipedia.org/wiki/Colecci%7B%5C'%7Bo%7D%7Dn_(moda)) (visitado 26-09-2020).
- [3] *Las 10 mejores aplicaciones y programas de diseño de modas*. URL: <https://mott.pe/noticias/aplicaciones-para-disenar-ropa-que-todo-disenador-de-modas-debe-tener/> (visitado 22-09-2020).
- [4] *CorelDRAW: software de diseño gráfico, ilustración y técnico*. URL: https://www.coreldraw.com/la/?x-sitetest=cdgs-b&utm-expid=.3Q67IJ2MTh6dy6g0Lq4uGQ.2&utm_referrer=https%3A%2F%2Fwww.google.com%2F (visitado 22-09-2020).
- [5] *Herramientas para el diseño de moda con CorelDRAW de Mesdits*. URL: <http://www.mesdits.com/> (visitado 22-09-2020).
- [6] *Software de PLM / Software de gestión del ciclo de vida del producto*. URL: <https://www.plm.automation.siemens.com/global/es/our-story/glossary/product-lifecycle-management-plm-software/12506> (visitado 22-09-2020).
- [7] *What is a .CSV file and what does it mean for my...* — *BigCommerce*. URL: <https://www.bigcommerce.com/ecommerce-answers/what-csv-file-and-what-does-it-mean-my-ecommerce-business/> (visitado 25-09-2020).
- [8] *Dashboard: Qué es y Para qué sirve - Glosario*. URL: <https://www.workana.com/i/glosario/que-es-un-dashboard/> (visitado 25-09-2020).
- [9] *Aprender sobre los tiques con Jira Software — Atlassian*. URL: <https://www.atlassian.com/es/agile/tutorials/issues> (visitado 24-09-2020).
- [10] *Top 4 software development methodologies — Synopsys*. URL: <https://www.synopsys.com/blogs/software-security/top-4-software-development-methodologies/> (visitado 25-09-2020).
- [11] *Agile-Waterfall Hybrid: Is It Right for Your Team?* — *Lucidchart Blog*. URL: <https://www.lucidchart.com/blog/is-agile-waterfall-hybrid-right-for-your-team> (visitado 25-09-2020).
- [12] *Principles behind the Agile Manifesto*. URL: <http://agilemanifesto.org/principles.html> (visitado 25-09-2020).
- [13] *Scrum: qué es, cómo funciona y por qué es excelente*. URL: <https://www.atlassian.com/es/agile/scrum> (visitado 25-09-2020).
- [14] *Scrum Methodology and Project Management*. URL: <https://www.mountangoatsoftware.com/agile/scrum> (visitado 25-09-2020).
- [15] *What is Scrum?* URL: <https://www.scrum.org/resources/what-is-scrum> (visitado 25-09-2020).
- [16] *The Scrum Framework Poster — Scrum.org*. URL: <https://www.scrum.org/resources/scrum-framework-poster> (visitado 26-09-2020).
- [17] *El backlog del producto: la lista de tareas pendientes definitiva*. URL: <https://www.atlassian.com/es/agile/scrum/backlogs> (visitado 26-09-2020).
- [18] *sprints — Atlassian*. URL: <https://www.atlassian.com/es/agile/scrum/sprints> (visitado 26-09-2020).

- [19] *Historias de usuario — Ejemplos y plantilla — Atlassian*. URL: <https://www.atlassian.com/es/agile/project-management/user-stories> (visitado 26-09-2020).
- [20] *Scrum: roles y responsabilidades — Deloitte España*. URL: <https://www2.deloitte.com/es/es/pages/technology/articles/roles-y-responsabilidades-scrum.html> (visitado 26-09-2020).
- [21] *Cuatro protocolos ágiles desmitificados — Atlassian*. URL: <https://www.atlassian.com/es/agile/scrum/ceremonies> (visitado 26-09-2020).
- [22] *Inspección del Incremento al Final del Sprint o Inspección Continua en el Sprint — Scrum.org*. URL: <https://www.scrum.org/resources/blog/inspeccion-del-incremento-al-final-del-sprint-o-inspeccion-continua-en-el-sprint> (visitado 26-09-2020).
- [23] *La metodología ágil y DevOps: ¿amigos o enemigos?* URL: <https://www.atlassian.com/es/agile/devops> (visitado 26-09-2020).
- [24] *¿Qué es DevOps? - Amazon Web Services (AWS)*. URL: <https://aws.amazon.com/es/devops/what-is-devops/> (visitado 26-09-2020).
- [25] *Qué es DevOps (y sobre todo qué no es DevOps) - Paradigma*. URL: <https://www.paradigmadigital.com/techbiz/que-es-devops-y-sobre-todo-que-no-es-devops/> (visitado 26-09-2020).
- [26] *DevOps in Practice - AdoraLabs*. URL: <https://www.adoralabs.io/curso/devops-in-practice/> (visitado 26-09-2020).
- [27] *Informe sobre herramientas DevOps 2020 — Podcast República Web*. URL: <https://republicaweb.es/podcast/informe-sobre-herramientas-devops-2020/> (visitado 26-09-2020).
- [28] *Cómo desmitificar el proceso DevOps, paso a paso*. URL: <https://searchdatacenter.techtarget.com/es/consejo/Como-desmitificar-el-proceso-DevOps-paso-a-paso> (visitado 26-09-2020).
- [29] *Tutorial: Gitflow and UML/SysML — LieberLieber Modelling Expert*. URL: <https://blog.lieberlieber.com/2019/09/11/gitflow-and-uml-sysml/> (visitado 26-09-2020).
- [30] *Continuous integration — Atlassian*. URL: <https://www.atlassian.com/agile/software-development/continuous-integration> (visitado 26-09-2020).
- [31] *Sueldos de la empresa — Glassdoor.es*. URL: <https://www.glassdoor.es/Sueldos/index.htm> (visitado 05-10-2020).
- [32] *Lenovo ThinkPad X270 20HN I5-7200U / HD 620 / 8GB / 256GB SSD / 12.5 Portátil*. URL: <https://www.coolmod.com/lenovo-thinkpad-x270-20hn-i5-7200u-hd-620-8gb-256gb-ssd-125-portatil-precio> (visitado 06-10-2020).
- [33] *Logitech B100 Ratón con Cable, 3 Botones, Seguimiento Óptico, Ambidiestro, PC/Mac/Portátil, Negro: Amazon.es: Informática*. URL: https://www.amazon.es/Logitech-B100-Rat%C3%B3n-con-Cable-3-Botones-Seguimiento-Optico-Ambidiestro-PC-Mac-Portatil-Negro%7B%5C'%7Bo%7D%7Dn-%7B%5C'%7Bo%7D%7Dptico-color/dp/B00AZKNPZC/ref=sr_1_2?__mk_es_ES=%7B%5CAA%7DM%7B%5CAA%7D%7B%5Cv%7BZ%7D%7D%7B%5C~%7BO%7D%7D%7B%5C~%7BN%7D%7D&dchild=1&keywords=mouse+trust&qid=1602001616&sr=8-27 (visitado 06-10-2020).
- [34] *Trust ClassicLine USB QWERTY Inglés Negro - Teclado (Estándar, Alámbrico, USB, QWERTY, Negro): Amazon.es: Informática*. URL: [https://www.amazon.es/Trust-ClassicLine-Teclado-Universal-Al%C3%B3n-%7Ba%7D%7Dmbrico/dp/B016WGGL0I/ref=sr_1_2?adgrpid=56500733672&dchild=1&gclid=CjwKCAjwq_D7BRADEiwAVMDdHnS6JCaSnE4kKa_R7rgWABJmUQSuB_a2vvNuch6FTPqfiPOqT1hoCgKYQAvD_BwE&hvadid=275309405673&hvdev=c&hvlocphy=20270&hvnetw=g&hvqmt=e&hvrnd=3119315988231458610&hvtargid=kwd-373901888614&hydadcr=23359_1796958&keywords=teclado+trust+classicline&qid=1602001735&sr=8-2&tag=hydes-21](https://www.amazon.es/Trust-ClassicLine-Teclado-Universal-Al%C3%B3n-%7Ba%7D%7Dmbrico/dp/B016WGGL0I/ref=sr_1_2?adgrpid=56500733672&dchild=1&gclid=CjwKCAjwq_D7BRADEiwAVMDdHnS6JCaSnE4kKa_R7rgWABJmUQSuB_a2vvNuch6FTPqfiPOqT1hoCgKYQAvD_BwE&hvadid=275309405673&hvdev=c&hvlocphy=20270&hvnetw=g&hvqmt=e&hvrnd=3119315988231458610&hvtargid=kwd-373901888614&hydadcr=23359_1796958&keywords=teclado+trust+classicline&qid=1602001735&sr=8-2&>tag=hydes-21) (visitado 06-10-2020).
- [35] *BenQ GW2280 21.5" LED FullHD — PcComponentes.com*. URL: https://www.pccomponentes.com/benq-gw2280-215-led-fullhd?gclid=CjwKCAjwq_D7BRADEiwAVMDdHqPd0HHREOlyx4aGd8tn79sNhoCm7IQAvD_BwE& (visitado 06-10-2020).

- [36] *3.5.4 Tabla de amortización simplificada - Agencia Tributaria*. URL: https://www.agenciatributaria.es/AEAT.internet/Inicio/Ayuda/Manuales_Folletos_y_Videos/Manuales_practicos/_Ayuda_Folleto_Actividades_economicas/3_Impuesto_sobre_la_Renta_de_las_Personas_Fisicas/3.5_Estimacion_directa_simplificada/3.5.4_Tabla_de_amortizacion_simplificada/3.5.4_Tabla_de_amortizacion_simplificada.html (visitado 06-10-2020).
- [37] *Comprar Work uno mesa auxiliar — Ofiprix ®*. URL: <https://www.ofiprix.com/muebles-de-oficina/mesas-de-oficina/mesas-individuales/work-uno-mesa-auxiliar.html#/12-tamano-100x60cm/211-color-marfil> (visitado 06-10-2020).
- [38] *Silla de Oficina MILO en Tela color Gris - Ofisillas.es*. URL: <https://www.ofisillas.es/silla-de-oficina-milo-brazos-ajustables-soporte-lumbar-en-tela-color-gris.html> (visitado 06-10-2020).