3.

En el análisis pongo el diagrama de flujo y en el diseño cómo es el diseño pensado.

2. PLANIFICACIÓN

**Estado del arte** – qué implicaciones tiene para (posible punto a añadir)

Ante este problema, qué posibilidades hay y la que he escogido, por qué..

Para este problema, existe esta solución

MTODOLOGÍA

Para la realización de este proyecto se utilizó la metodología ágil Kanban[definir con pie de página?] con algunos elementos característicos de la metodología Scrum[definir con pie de pagina?]. DEFINIR AMBOS.

Inicialmente, se realizó una reunión que ayudó a establecer una pila de producto con la prioridad y ----- de cada historia de usuario. También se estableció un período de tiempo de tantas semanas para el aprendizaje de las tecnologías a usar, con una fecha –fin¿? Marcada. Esto fue lo más Scrum

Con respecto a la metodología Kanban, se …. Y así sutilmente s explica de qué va esta metodología... se fueron realizando las diferentes funcionalidades a implementar, siendo revisadas por el *Project manag*er antes de pasar a la siguiente.

Esta elección híbrida de metodologías fue dada por varios factores. En primer lugar, el desarrollo de este proyecto generó el contexto perfecto para probar esta nueva forma de trabajar sin tomar grandes riesgos al tratarse de un proyecto de menor envergadura frente a otros realizados siguiendo la metodología Scrum. Por otra parte, se creyó conveniente no establecer esprints con fechas fijas debido a la necesidad de realizar estudios relacionados con las posibles librerías a utilizar durante la implementación de la aplicación, por lo que era difícil calcular el tiempo exacto que costaría probarlas y compararlas.

No sé si se debería poner aquí lo de que la pila del producto es más usual en la metodología Scrum, o decir que se han cogido elementos de esta metodología.

Por otro lado, la creación de la pila de producto es algo usual en la metodología ágil SCRUM, es por ello por lo que, como se ha mencionado antes, a pesar de que mayoritariamente se siguió durante todo el proyecto la metodología ágil Kanban, se conservaron algunos elementos pertenecientes a la metodología ágil SCRUM.

TECNOLOGÍA (el apartado debería ir aquí, redactado en otra parte y añadido)

PLANIFICACIÓN

Primeramente, para la planificación de este proyecto, se creó una pila de producto con el fin de tener una idea más clara sobre la complejidad de cada funcionalidad y establecer un orden de prioridad a la hora de su implementación. La tabla \colorbox{CadetBlue}{ X} muestra la pila de producto, formada por las historias de usuario establecidas para representar las funcionalidades y requisitos de la aplicación. Además, durante esta primera etapa, se detallaron todas las historias de usuario, incluyendo un escenario válido e inválido para cada una. También se diseñó la base de datos necesaria para el almacenamiento de la información que usaría la aplicación y se planificó tanto la duración total del proyecto como la estimación de las tareas a realizar durante el período de prácticas.

Una vez realizada esta planificación inicial, antes de empezar con la implementación de la aplicación, se estableció un período de formación. Este período tendría una duración entre una y dos semanas para el estudio y aprendizaje de las tecnologías que se iban a usar en este proyecto, explicadas con más detalle en la sección \colorbox{CadetBlue}{X}. Es por ello por lo que se consultaron varios tutoriales sobre el \textit{*framework Angular 8}*, el lenguaje de programación \textit{*Typescript}* y el uso de escritura lambda en \textit{Java}.También se consultaron proyectos realizados anteriormente, lo que fue de gran ayuda para la estimación del tiempo y los costes, así como para entender y analizar el estilo de programación utilizado en la empresa con el fin de generar un código relativamente homogéneo y entendible.

Cabe añadir que, durante esta etapa de formación, se estuvo estudiando las posibles mejoras previstas para la aplicación y cómo sería su implementación. Puesto que dichas mejoras están relacionadas con las predicciones de consumo de agua, se optó por la opción de usarlas como pretexto para la introducción al mundo del \textit{machine learning} por parte del equipo de desarrollo \textit{software}. Además, estas mejoras a largo plazo son independientes del producto originalmente diseñado, por lo que no se incluyen en el coste y tiempo estimado del proyecto, documentado en el \colorbox{CadetBlue}{apartado x}.

con la introducción de machine learning en el proyecto para los cálculos de predicciones

Por último, se procedió a la implementación del proyecto, empezando por su configuración inicial, prosiguiendo con la creación de la base de datos y continuando con las funcionalidades de la aplicación, de mayor a menor prioridad.

TABLA PILA DEL PRODUCTO

/\* no se añadirá por ahora \*/

Esto último fue indispensable no solo para entender y habituarse a la lectura de código ajeno, sino también para fomentar la homogenización de un estilo de escritura de código concreto, fomentando la fácil comprensión y comunicación dentro del equipo.

con el objetivo de una rápida adaptación al estilo de escritura de código y homogenización del mismo en todos los proyectos. Esto es algo sumamente útil para poder y entender el código de los compañeros y fomentar la comunicación y entendimiento entre el equipo de manera fluida.

\*\*\*\*\*\*\*\*\*/

\*IMPORTANTE

Hay que comentar que la parte de machine learning está pensada como mejora y no se tiene cuenta ni en la planificación ni en los costes totales. Bueno, dentro de la planificación se tiene en cuenta realizar un estudio.

\*después de la tabla

Pregunta: ¿se debe poner en la planificación la parte de los otros compañeros? ¿mencionar cuántas personas iban a realizar el proyecto?

SEGUIMIENTO DEL PROYECTO

Cabe añadir que, pese a que estaba previsto llegar a implementar a nivel muy básico la funcionalidad relacionada con la parte del *machine learning*, debido

las prioridades de las funcionalidades que se debían implementar y el tiempo de realización de las mismas en función de su complejidad. Cabe mencionar

aunque la pila de producto suele ser más propia de la metodología ágil SCRUM,

y, teniendo en cuenta las circunstancias de cambio reciente en la metodología de trabajo en la empresa,

Teniendo en cuenta la metodología ágil Kanban escogida para la realización de este proyecto, la planificación realizada consistió primeramente en establecer la pila del producto.

METODOLOGÍA

Debido a la naturaleza de este proyecto, sujeto a la posibilidad de cambios y mejoras en algunos de sus aspectos menos definidos, se optó por seguir la metodología ágil Kanban, con algunas características más propias de la metodología SCRUM. Por una parte, .. pila de producto (propia de SCRUM)

En este caso, la pila de producto la conforman las historias de usuario creadas a partir de los requisitos iniciales. Cabe añadir que, aunque ésta sea una práctica habitual a la hora de conformar la pila de producto, no es siempre así e incluso puede estar simplemente formada por tareas.

2.3. ESTIMACIÓN DE RECURSOS Y COSTES DEL PROYECTO (COCOMO)

Para la estimación de los recursos y costes de un proyecto se utilizó la estimación LCD y el modelo COCOMO básico. También se tuvo en cuenta tanto el coste del *hardware* como del *software* utilizado, reflejados en la \colorbox{CadetBlue}{ tabla X}.

TABLA DE HARDWARE Y SOFTWARE – Intellij, bbdd(¿?), ordenador, ratón, pantalla extra.

2.3.1 ESTIMACIÓN LCD

El modelo de estimación LCD calcula las líneas de código que aproximadamente tendrá una aplicación en base a la cantidad de líneas de código estimadas [referencia, traspas de classe]. Para ello, se usa la fórmula mostrada en la figura X, donde – hace referencia a -, - a x y x a x.

FORMULA [añadir referencia en la figura si hiciera falta, traspas de clase]

Los valores pesimistas, optimistas y probables usados para el cálculo del valor esperado de cada módulo del sistema, se obtuvieron teniendo en cuenta proyectos de envergadura similar realizados anteriormente en la empresa. También se tuvo en cuenta los lenguajes de programación usados en proyectos pasados, puesto que la cantidad de líneas puede fluctuar dependiendo del lenguaje de programación utilizado. Para este proyecto en concreto, se acordó la utilización del lenguaje de programación Typescript para el frontend y de Java para el backend.

Visualización de los datos de un contador

Visualización de la gestión de datos adicionales

Visualización de la gestión de alarmas

Visualización del mapa interactivo

Visualización de la gestión de datos

Tratamiento de excepciones

Acceso a la base de datos

Acceso a la API de Sensores\*

Acceso a la API de Contadores\*

Acceso a la API de Alarmas\*

Procesamiento de los datos

La tabla X muestra la cantidad de líneas que se estimaron para cada módulo aplicando la fórmula mostrada en la Figura X y el total obtenido al sumarlas.

\*final del apartado (no se si añadirlo)

Se debe tener en cuenta que a pesar de todo y, como su nombre indica, es una estimación aproximada. Es por ello que no se ha tenido en cuenta ni posibles riesgos ni el tiempo de formación requerido, así como las mejoras planteadas.

Cabe añadir que la cantidad de líneas utilizadas para cada módulo varía dependiendo del lenguaje de programación utilizado. La implementación del frontend está realizada principalmente con Typescript y la del backend en Java. También es importante tener en cuenta el estilo de programación de cada persona, ya que esto podría suponer una variación considerable de líneas. Afortunadamente, esto último no se ha tenido en cuenta debido al esfuerzo del equipo de Software por mantener un estilo de programación homogéneo para todos los proyectos.

tenía el fin de determinar el tamaño del proyecto. Esta estimación consistía en y el modelo básico COCOMO para calcular los costes.

los recursos humanos, de *hardware* y de *software*. Para el primero

2.3.2 COCOMO BÁSICO

Para la estimación de costes de un proyecto software, el modelo COCOMO (Modelo Constructivo de Costos) es uno de los más usados. Este modelo se basa en estimaciones matemáticas que miden el tamaño del producto final, principalmente a través del conteo de líneas de código [referencia a wiki].

**TECNOLOGÍA**

A la hora de hablar de las tecnologías que se utilizaron en la elaboración del código, de debe distinguir entre dos partes bien diferenciadas: \textit{front-end} y \textit{back-end}. La primera hace referencia a la parte web de la aplicación, la capa de presentación, aquella que interactúa con los usuarios \cite{bib:front-end}. Por contra, la capa de acceso a datos, el \textit{back-end}, es la parte encargada de interactuar con el servidor, la base de datos u otras aplicaciones con el objetivo de procesar la información proveniente tanto de alguno de los tres anteriores para enviarla al \textit{front-end}, como viceversa \cite{bib:frontendybackend}.

Bib:front-end - [referencia Wikipedia - <https://es.wikipedia.org/wiki/Front_end_y_back_end>: *Primer glosario de comunicación estratégica en español*. Fundéu BBVA.]

@inbook{bib:front-end,

author = "Estanislao Echazú / Ramiro Rodríguez",

title = "Primer glosario de comunicación estratégica en español",

publisher="fundéu BBVA",

year = "2018",

keywords="front-end"

}

Bib:frontendybackend - [https://descubrecomunicacion.com/que-es-backend-y-frontend](https://descubrecomunicacion.com/que-es-backend-y-frontend/#:~:text=El%20back%20end%20del%20sitio,para%20que%20todo%20funcione%20correctamente.)

NOTA: la palabra framework, poner un numerito la primera vez que sale!!!!

En primer lugar, con respecto a la implementación del \textit{frontend}, se utilizó el \textit{framework}\footnote{\colorbox{CadetBlue}{ \*\*\*6. definición de framework y referencia, pero se debe mirar si ésta es la primera vez de la palabra} } \textit{AngularJS}, concretamente Angular 8, una de sus versiones más recientes. Este \textit{framework} incluye el uso del lenguaje de programación \textit{Typescript} y los lenguajes de etiquetas \textit{HTML} y \textit{CSS}, con los cuales se obtuvo una interfaz gráfica cómoda y agradable a la vez que un código bien estructurado, siguiendo el patrón MVC (modelo-vista-controlador). Además, en relación a la interfaz de usuario, para lograr que ésta fuera \textit{responsive}{\colorbox{CadetBlue}{ footage}, se utilizó otro \textit{framework} llamado \textit{Bootstrap}.

Con respecto al \textit{back-end}, se usó \textit{Spring}, el cual es un \textit{framework} de código abierto para el desarrollo de aplicaciones con \textit{Java}, lenguaje de programación que se utilizó conjuntamente \cite{bib:spring}. Este \textit{framework} ayudó en la simplificación de las configuraciones, el acceso a la base de datos y las peticiones a las APIs (\textit{Application Programming Interface}) \colorbox{CadetBlue}{9. Definir API en caso de no haber aparecido antes} externas utilizadas en la aplicación.

Bib:spring - [referencia: [https://openwebinars.net/blog/que-es-spring-framework/](https://openwebinars.net/blog/que-es-spring-framework/#:~:text=Como%20definici%C3%B3n%20podemos%20decir%20que,las%20necesidades%20de%20la%20aplicaci%C3%B3n.)]

6. Poner definición de framework

7. Definir machine learning y poner referencia o ver si se ha hecho antes, si es la primera vez que aparece (seguramente la primera vez sea en la intro, asi que OJO!)

8. definir librería si es la primera vez que aparece

9. ver si es la primera vez que se usa la palabra API y definirla

Por otro lado, cabe destacar la parte de las posibles mejoras que se plantearon para la aplicación. Estas mejoras están relacionadas con el análisis predictivo a modo de introducción al \textit{machine learning}\colorbox{CadetBlue}{7}. Para poder llevar a cabo su implementación en un futuro, se decidió estudiar el uso de \textit{Python}, así como de \textit{R}. Esto se debió a que ambos lenguajes de programación tienen librerías\colorbox{CadetBlue}{8} muy populares para \textit{machine learning}, con un gran soporte técnico y documentación.

Por último, en relación con los recursos de \textit{hardware} utilizados durante el desarrollo del proyecto, sólo se requirió de un ordenador de sobremesa para cada miembro del equipo. Aunque durante el estado de alarma debido al COVID-19, se tuvieron que usar también los ordenadores personales.

**HERRAMIENTAS**

Para este proyecto se utilizaron una serie de herramientas con la finalidad de ayudar en la correcta gestión, documentación desarrollo del mismo, así como para mantener una buena comunicación dentro del equipo de trabajo.

HERRAMIENTAS DE GESTIÓN Y DOCUMENTACIÓN

JIRA (¿mejor sin títulos y se van nombrando?)

En primer lugar, se utilizó la herramienta Jira Software, una plataforma web que ayuda en la planificación, desarrollo, gestión y supervisión de proyectos software ágiles\colorbox{CadetBlue}{1} \cite{bib: atlassian}. Dispone de un tablero Kanban para visualizar el flujo de trabajo y hacer un seguimiento de las tareas, lo que hizo tener una visión general del avance del proyecto en todo momento.

Bib: atlassian – [<https://www.atlassian.com/es/software/jira>]

IMAGEN DE EJEMPLO Kanban

Puesto que la herramienta Jira está especializada en el desarrollo de proyectos informáticos, ésta permite la creación de tareas de diferente tipo relacionadas con la implementación de funcionalidades de una aplicación, como serían las historias de usuario, errores y las épicas2. Además, es posible detallarlas añadiéndoles descripción, prioridad, puntos de historia3 y subtareas. Cada tarea también puede ser asignada al miembro del equipo responsable de realizarla y/o al responsable de su supervisión.

e imagen de una tarea detallada

Por otra parte, cabe destacar que Jira permite también la creación y gestión de esprints, que son propios de la metodología ágil SCRUM. A pesar de que en este proyecto no se realizaron esprints, es importante tener en cuenta esta posibilidad, ya que es otra forma más de facilitar la combinación de las metodologías ágiles SCRUM y Kanban, las cuales se utilizaron elementos de ambas para el desarrollo de este proyecto, como se explica en la sección x.

1. Proyecto el cual está desarrollado con metodologías ágiles, explicadas en la sección X.
2. Historia de usuario de gran tamaño que se descompone de varias de tamaño menor para poder ser gestionadas [referencia a <https://www.scrummanager.net/bok/index.php?title=Epic#:~:text=Se%20denomina%20Epic%20a%20una,seguimiento%20cercano%20(normalmente%20diario).&text=Historia%20de,Requisitos%20%C3%A1giles>]
3. Medida subjetiva para estimar la dificultad y tiempo de realización de una tarea
4. Plataforma web que se utiliza para la colaboración, documentación y recopilación de información [referencia a la Wikipedia: <https://es.wikipedia.org/wiki/Software_para_wikis>]

\*NOTA: detallar en qué consiste el estado de cada tarea cuando se explique la metodología, en el apartado anterior. ¿Poner los nombres Jira, Kanban, Confluence y todo eso en cursiva?

CONFLUENCE

Otra herramienta utilizada fue Confluence, para la consulta la documentación de las diferentes APIs ya implementadas por la empresa o uso de algunos programas para la comunicación a distancia del equipo. Confluence es una wiki\colorbox{CadetBlue}{4} utilizada en entornos corporativos para la colaboración en equipo \cite{bib: wiki}.

Bib:wiki - [referencia de la Wikipedia:  Woods, Dan; Thoeny, Peter (23 de julio de 2007). [*Wikis for dummies*](http://books.google.com/books?id=5VXgXlU7g-YC). [John Wiley & Sons](https://es.wikipedia.org/wiki/John_Wiley_%26_Sons). p. [193](http://books.google.com/books?id=5VXgXlU7g-YC&pg=PA193). [ISBN](https://es.wikipedia.org/wiki/ISBN) [978-0-470-04399-8](https://es.wikipedia.org/wiki/Especial:FuentesDeLibros/978-0-470-04399-8).]

Esta herramienta fue imprescindible, puesto que se usaron varias de estas APIs ya documentadas en esta plataforma, para la implementación de la aplicación.

Cabe añadir que Confluence también permite redactar informes sobre la evolución y reuniones que se han realizado a lo largo del desarrollo de un proyecto, para poder tener un seguimiento bastante preciso del mismo.

Por último, debido a situación derivada a causa del COVID-19, se pasó a trabajar en remoto. Esto evento se detalla mejor en la sección X. Es por ello por lo que la herramienta Microsoft Teams pasó a tener una gran relevancia a lo largo del desarrollo del proyecto, ya que mantuvo a todos los miembros del equipo en comunicación continua. A través de esta plataforma fue posible organizar reuniones e informar del estado del proyecto en todo momento tanto al mánager como al resto del equipo.

HERRAMIENTAS DE DESARROLLO

GitLab, la IDE Intellij, phpMyAdmin

Puesto que la aplicación web a desarrollar de este proyecto requería del almacenamiento de diferentes tipos de datos, se utilizó para ello \textit{MySQL}, un sistema de gestión de bases de datos relacional de código abierto [referencia wiki:  <http://www.oracle.com/us/products/mysql/overview/index.html>]. Además, con la intención de crear de manera gráfica, fácil y rápida tanto la base de datos necesaria para la aplicación web \colorbox{CadetBlue}{Water Clients (lo pongo con el nombre?} como los datos de prueba iniciales, se usó la herramienta \textit{phpMyAdmin}.

phpMyAdmin es una herramienta destinada a la administración de bases de datos creadas con MySQL de manera gráfica y cómoda.

GITLAB

Por otro lado, una parte crítica en el desarrollo de un proyecto informático es la mezcla de código, puesto que a lo largo de la implementación de una aplicación el equipo trabaja paralelamente en las diferentes funcionalidades. La herramienta \textit{GitLab}, , fue fundamental en esta etapa del proyecto.

INTELLIJ

Con respecto al entorno de trabajo, se utilizó la ide5 Intellij. DEFINIR QUÉ ES UNA IDE Y TODO ESO.