

MeceAula

Alberto Anchuste Ceballos

Grado Ingeniería Informática

Desarrollo multiplataforma de aplicaciones móviles

Nombre Consultor/a

Carlos Sanchez Rosa

Jordi Almirall Lopez

Nombre Profesor/a responsable de la asignatura

Carles Garrigues Olivella

Fecha Entrega

09/01/2023



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

B) GNU Free Documentation License (GNU FDL)

Copyright © AÑO TU-NOMBRE.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>MeceAula</i>
Nombre del autor:	<i>Alberto Anchuste Ceballos</i>
Nombre del consultor/a:	<i>Carlos Sanchez Rosa Jordi Almirall Lopez</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	<i>01/2023</i>
Titulación::	<i>Grado Ingeniería Informática</i>
Área del Trabajo Final:	<i>Desarrollo multiplataforma de aplicaciones móviles</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>colecta, educación, mecenazgo</i>
<p>Resumen del Trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.</i></p>	
<p>En este trabajo se exponen todas las fases de las que consta el desarrollo de una aplicación móvil multiplataforma. La aplicación consiste en una cooperación colectiva para escuelas, es decir, un profesor de cierta escuela ve ciertas carencias de materiales en su aula. En ese momento, publica un anuncio o colecta en la aplicación para recaudar fondos. El nombre de la aplicación MeceAula, proviene de ahí, de poder brindar ayuda a una causa cultural, en este caso, un aula de una escuela.</p> <p>La metodología que se ha utilizado en el proyecto es en cascada lo que nos ha permitido definir todas las etapas del proceso en un primer momento y después ejecutarlas de manera secuencial.</p> <p>La idea de la aplicación es, a partir de un único desarrollo, poder obtener una aplicación para Android y para iOS, de ahí el nombre de multiplataforma. Esto lo hemos conseguido gracias al framework Ionic.</p> <p>La validación de la aplicación se ha realizado mediante la automatización de tests con el framework Cypress, el cuál nos permite probar cualquier aplicación web.</p>	

Abstract (in English, 250 words or less):

This document exposes all multi-platform application development phases of one application. The application is crowdfunding for schools.

The methodology implemented is the waterfall where all the development phases are established in the first moment and then we can execute them sequentially.

The main idea of the application is to have just one development. From this development, we can obtain the two artifacts for mobile: one for iOS and the other one for Android. This is possible thanks to the Ionic Framework.

The Unit Accept Testing of the application is done with Cypress Framework which let us test all web applications in the browser.

Índice

Desarrollo multiplataforma de aplicaciones móviles	i
1. Introducción	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	2
1.3 Enfoque y método seguido	2
1.4 Planificación del Trabajo	2
1.5 Breve resumen de productos obtenidos	5
1.6 Breve descripción de los otros capítulos de la memoria	5
2. DCU: Diseño centrado en el usuario	8
2.1 Usuarios y contexto de uso	8
2.1.1 Métodos de indagación	8
2.1.2 Fichas perfiles	9
2.2. Diseño conceptual	11
2.2.1. Escenarios de uso	11
2.2.1 Flujos de interacción	12
2.3. Prototipado	13
2.3.1 Sketches	13
2.3.2 Prototipado horizontal	18
2.4 Evaluación	21
3. Diseño técnico de la aplicación	22
3.1 Definición de los casos de uso	22
3.2 Diseño de la arquitectura	26
4. Desarrollo e implementación	28
4.1 Estructura del proyecto	28
4.1.1 Parte cliente o front end	28
4.2 Entornos de programación.	32
4.3 Repositorios	32
4.4 Servicios y Aplicaciones	33
4.4.1 Vmware	34
4.4.2 Postman	35
4.4.3 Firebase	36
4.4.4 AWS	36
4.6 Despliegue de nueva versión de aplicación	37
4.7 Principales dificultades en la implementación	37
4.8 Desviaciones	39
5.1 Software utilizado	43
5.2 Definición pruebas validación	43
5.3 Resultado pruebas validación	44
6. Conclusiones	45
7. Glosario	46
8. Bibliografía	47
9. Anexos	49
9.1 Transcripción de entrevista presencial	49
9.2 Preguntas de las encuestas realizadas	50

Lista de figuras

Ilustración 1 - Horas estimadas hitos	3
Ilustración 2 - Hito PEC 1 - Plan de trabajo	3
Ilustración 3 - Hito PEC 2 - Diseño y arquitectura	4
Ilustración 4 - Hito PEC 3 – Implementación	4
Ilustración 5 - Hito Entrega final	5
Ilustración 6 - Perfil Colector	10
Ilustración 7 - Perfil Mecenaz	10
Ilustración 8 - Perfil Curioso	11
Ilustración 9 - Escenario de uso 1	11
Ilustración 10 - Escenario de uso 2	12
Ilustración 11 - Escenario de uso 3	12
Ilustración 12 - Flujos de interacción	13
Ilustración 13 - Boceto Menú Principal	14
Ilustración 14 - Boceto Login	14
Ilustración 15 - Boceto Mis aportes	15
Ilustración 16 - Boceto colaborar en colecta	15
Ilustración 17 - Detalle del aporte	16
Ilustración 18 - Boceto crear colecta	16
Ilustración 19 - Boceto Mis colectas	17
Ilustración 20 - Boceto Mi perfil	17
Ilustración 21 - Boceto registrar usuario	18
Ilustración 22 - Prototipo Menu Principal	19
Ilustración 23 - Prototipo Login	19
Ilustración 24 - Prototipo Registrar usuario	20
Ilustración 25 - Prototipo Mis colectas	20
Ilustración 26 - Prototipo Detalle Aporte	21
Ilustración 27 - Prototipo Registrar	21
Ilustración 28 - Identificador: CU-001	22
Ilustración 29 - Identificador: CU-002	23
Ilustración 30 - Identificador: CU-003	23
Ilustración 31 - Identificador: CU-004	23
Ilustración 32 - Identificador: CU-005	24
Ilustración 33 - Identificador: CU-006	24
Ilustración 34 - Identificador: CU-007	24
Ilustración 35 - Identificador: CU-008	25
Ilustración 36 - Identificador: CU-009	25
Ilustración 37 - Identificador: CU-010	25
Ilustración 38 - Diagrama UML	26
Ilustración 39 - Diagrama arquitectura del sistema	27
Ilustración 40 - Estructura carpetas front end	29
Ilustración 41 - Estructura carpetas back end	31
Ilustración 42 - Repositorios meceaula	33
Ilustración 43 - Repositorio app meceaula	33
Ilustración 44 - Virtual Machine settings	34
Ilustración 45 - Postman example	35
Ilustración 46 - Postman examle 2	35
Ilustración 47 - Estimación desviación	39
Ilustración 48 - Emulador - Iniciar Sesión colectas	40
Ilustración 49 - Emulador - Listado	40
Ilustración 50 - Emulador - Registrar usuario panel lateral	41
Ilustración 51 - Emulador -	41

Ilustración 52 - Emulador Aporte/Detalle colecta5. Validación aplicación	Ilustración 53 - Emulador Crear 42	
Ilustración 54 - Definición pruebas validación		44
Ilustración 55 - Resultado pruebas validación		44

1. Introducción

1.1 Contexto y justificación del Trabajo

Este año, 2022, está siendo un año muy complicado económicamente para la mayoría de las familias. Diversos motivos, entre los que destaca un conflicto armado en territorio europeo han supuesto que la inflación se dispare, produciendo una gran pérdida de nivel adquisitivo a la población y golpeando en mayor medida a las familias más vulnerables.

Este encarecimiento de la vida repercute directamente a las familias en su relación con sus hijos y la escuela. El material escolar ha subido de precio aproximadamente entre un 20-30% lo que dificulta la vuelta al cole en muchos hogares.

Por otra parte, esta suba de precios también afecta a las administraciones públicas y, por ende, se está viendo reflejado en las aulas de las escuelas donde cada vez hay menos recursos de los que debería haber para proveer una educación de calidad a los alumnos. Algunos ejemplos de esta falta de recursos son: Pizarras en malas condiciones que no serán sustituidas, excursiones a museos que no se podrán acometer, talleres con falta de herramientas básicas para llevar a cabo proyectos, falta de materiales como papeleras, borradores, etc.

La necesidad que se pretende cubrir es que la educación de lo más pequeños no se vea limitada por esta falta de recursos en el aula. Ahora mismo este problema, la gran mayoría de veces, se soluciona gracias a la generosidad de profesores y padres, mientras que otras tantas veces, se queda sin solucionar lo que puede desembocar en falta de interés de los alumnos o apagar su creatividad.

Para ayudar con esta falta de recursos se propone el desarrollo de una aplicación para móviles que promueve el mecenazgo de aulas. Un profesor propone una necesidad de un aula y, una persona de manera altruista puede aportar su grano de arena para solucionarlo.

Examinado aplicaciones y páginas web en idioma español no hay ninguna app que se dedique exclusivamente a este tipo de apadrinamiento. Sin embargo, existe la siguiente página web en la que se realizan aportaciones como las que he comentado anteriormente:

<https://www.donorschoose.org/>

1.2 Objetivos del Trabajo

Desarrollar una aplicación para dispositivos móviles que permita realizar un apadrinamiento de una clase de una escuela. Este apadrinamiento o mecenazgo consiste en apoyar económicamente las necesidades de un aula que hayan sido publicadas previamente por un profesor. Esta aplicación funcionará en dispositivos Android e iOS.

Objetivos principales:

Como mecenas o padrino, poder realizar una donación económica a cualquier proyecto disponible mediante el uso de la aplicación.

Como profesor tener la capacidad de crear un proyecto de recaudación para mi aula, en el cual podré especificar la descripción, la fecha límite, etc.

Como desarrollador de aplicación tener un único código de desarrollo para ambos proyectos (Android e iOS). De manera que no se tenga que realizar un desarrollo individual para cada plataforma.

1.3 Enfoque y método seguido

La idea principal del proyecto es desarrollar un producto nuevo, desde cero, en el que se irán acometiendo entregas fijadas en un calendario. Cada una de estas entregas corresponde a una fase del desarrollo: Plan de trabajo, diseño y arquitectura de la aplicación, implementación y entrega final.

La metodología de software a utilizar en este proyecto será la conocida como “en cascada” o “waterfall”. Esta metodología es de forma secuencial y esto es ideal para ajustar las distintas fases del proyecto a las diferentes entregas de las PECs que también son secuenciales.

Por otra parte, conocemos de antemano todos los requisitos, funcionalidades y el tiempo que disponemos para el desarrollo del proyecto por lo que es bastante improbable que surjan imprevistos como añadir nuevas funcionalidades, malentendidos con el cliente, etc., lo que favorece la metodología en cascada. Si no tuviésemos claro el producto que queremos desarrollar y tuviésemos que ir enseñándoselo poco a poco al cliente para que se fuese haciendo una idea de lo que quiere, entonces podríamos inclinarnos por utilizar una metodología ágil.

1.4 Planificación del Trabajo

En la siguiente tabla se definen los principales grandes hitos del proyecto con el número de horas que se dedicarán a ello.

Hito	Horas estimadas*
PEC 1 - Plan de trabajo	36
PEC 2 - Diseño y arquitectura	72
PEC 3 - Implementación	108
Entrega final	52
Total	268

Ilustración 1 - Horas estimadas hitos

*El número de horas dedicadas al día son 2 los días laborales y 4 los fines de semana (sábados y domingos).

Para conocer en detalle las subtareas de cada hito, se adjuntan capturas del diagrama de Gantt.

Hito PEC 1 - Plan de trabajo

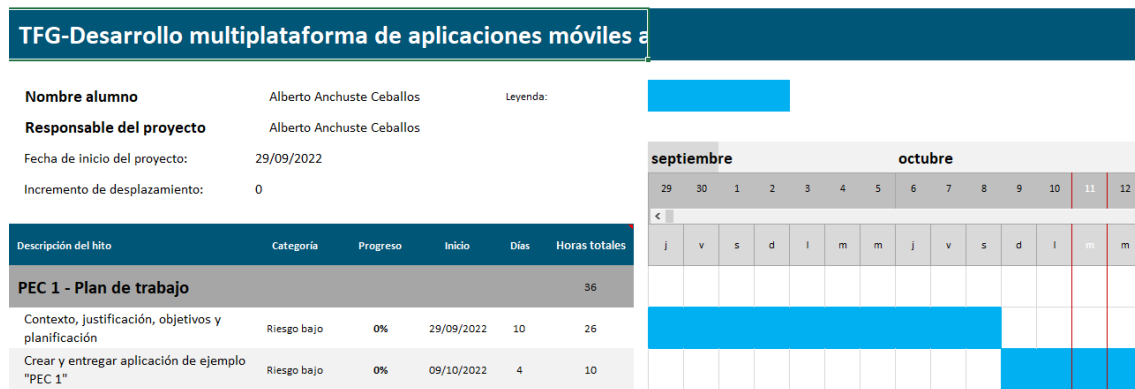


Ilustración 2 - Hito PEC 1 - Plan de trabajo

Hito PEC 2 - Diseño y arquitectura

Nombre alumno: Alberto Anchuste Ceballos Leyenda:

Responsable del proyecto: Alberto Anchuste Ceballos

Fecha de inicio del proyecto: 29/09/2022

Incremento de desplazamiento: 19

Descripción del hito	Categoría	Progreso	Inicio	Días	Horas totales
PEC 1 - Plan de trabajo					36
Contexto, justificación, objetivos y planificación	Riesgo bajo	0%	29/09/2022	10	26
Crear y entregar aplicación de ejemplo "PEC 1"	Riesgo bajo	0%	09/10/2022	4	10
PEC 2 - Diseño y arquitectura					72
Usuarios y contexto de uso	Riesgo bajo	0%	13/10/2022	4	12
Diseño conceptual	Riesgo bajo	0%	17/10/2022	5	10
Prototipado	Riesgo bajo	0%	23/10/2022	5	14
Evaluación	Riesgo bajo	0%	27/10/2022	4	12
Definición casos de uso	Riesgo bajo	0%	31/10/2022	5	10
Diseño arquitectura	Riesgo bajo	0%	05/11/2022	5	14

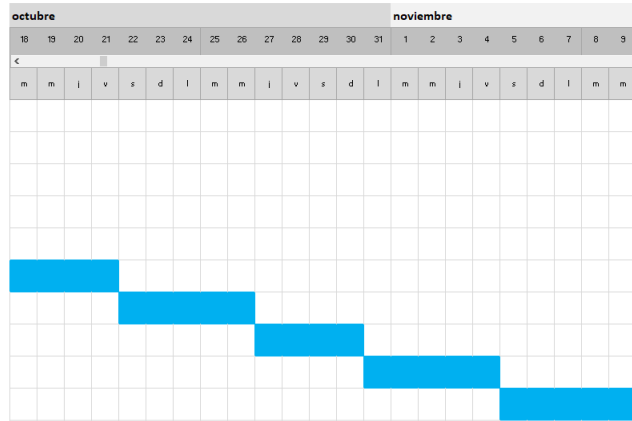


Ilustración 3 - Hito PEC 2 - Diseño y arquitectura

Hito PEC 3 – Implementación

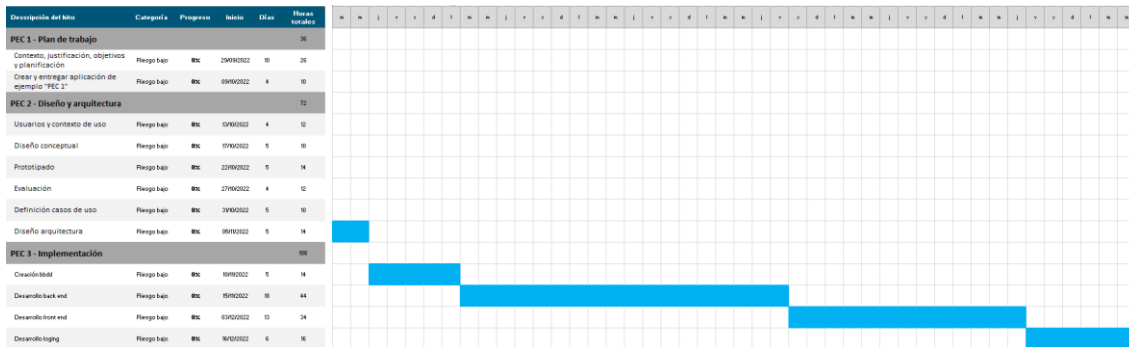


Ilustración 4 - Hito PEC 3 – Implementación

Hito Entrega final

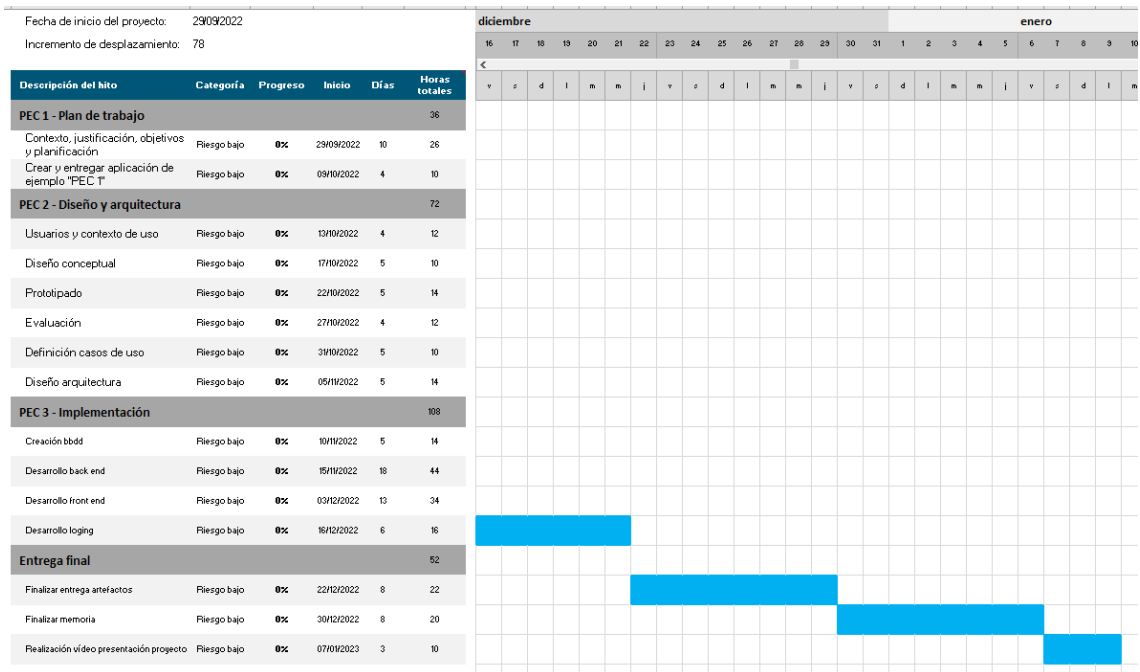


Ilustración 5 - Hito Entrega final

Con la entrega de la memoria se adjunta Excel en el cuál se puede ver con mejor calidad dicho diagrama de Gantt.

Por otra parte, los recursos necesarios para llevar a cabo el proyecto:

- Entorno de desarrollo Visual Studio Code
- Software de desarrollo de prototipos.
- Software de gestión de control de versiones GitHub
- Máquina donde se alojará servidor.
- Máquina en la nube que alojará la BBDD MySQL.
- Máquina desarrollo iOS en la nube

1.5 Breve resumen de productos obtenidos

Los artefactos que se entregarán al finalizar el proyecto son:

Un fichero .apk de la aplicación final para poder utilizarla en un teléfono móvil que trabaje con el sistema operativo Android.

Un fichero .ipa de la aplicación final o un link público para poder utilizar la aplicación en un teléfono con sistema operativo iOS.

1.6 Breve descripción de los otros capítulos de la memoria

PEC 2 - Diseño y arquitectura

Usuarios y contexto de uso:

Tiene como objetivo conocer hacia qué público irá dirigido el desarrollo. Conoceremos objetivos, necesidades y el contexto de uso.

Diseño conceptual:

En esta fase se elaborarán los distintos escenarios de uso en base a la información adquirida en el paso anterior. Los escenarios de uso describen como el usuario utilizará el producto.

Prototipado:

Consiste en elaborar un prototipo, que es una simulación de la aplicación. Aquí es donde tendremos que tomar nuestras primeras decisiones en cuanto a diseño y funcionalidad de la app.

Evaluación:

En esta etapa evaluaremos el prototipo. Realizaremos algunos test con usuarios para saber si estamos desarrollando el proyecto de forma adecuada.

Definición de casos de uso:

Estableceremos formalmente los casos de uso que sirven para implementar las funcionalidades de la aplicación.

Diseño de arquitectura:

Definimos la arquitectura y la relación entre los distintos componentes de nuestra aplicación: BBDD, APIs, procesos, servidores, etc.

PEC 3 – Implementación

Creación bbdd

Fase donde se creará la base de datos de la aplicación. En principio será la base de MySQL.

Desarrollo backend

Se desarrollará toda la parte del servidor. Es el encargado de recibir las peticiones del cliente, procesarlas y devolver la información al cliente. En un principio este desarrollo será en Java Spring.

Desarrollo frontend

En esta etapa construiremos la parte cliente de la aplicación. Se desarrollará con el Framework Ionic que nos permite desarrollar una aplicación Android o iOS de manera híbrida con un mismo código fuente.

Desarrollo login

Desarrollaremos el punto de entrada a la aplicación mediante la plataforma Firebase.

Entrega final

Finalizar entrega artefactos

Ultimaremos la generación de los artefactos release para incluir en la entrega final.

Finalizar memoria

Se abordará y finalizará los últimos puntos de la memoria.

Realización vídeo presentación proyecto

Se grabará el vídeo de presentación de todo el proyecto.

2. DCU: Diseño centrado en el usuario

2.1 Usuarios y contexto de uso

En este punto utilizaremos varios métodos de indagación en los que involucraremos al usuario en diferentes tareas para obtener información que nos ayudará a definir como debe ser nuestra aplicación.

2.1.1 Métodos de indagación

Método cualitativo

Empezaremos utilizando un método cualitativo en el que no nos importa la cantidad de información recogida sino la calidad de esta. Para esta aplicación creo que la mejor opción es realizar una entrevista en la cual utilizaremos un guion pequeño para iniciarla y luego dejaremos que siga tomando los cauces naturales para obtener información de calidad en un contexto cómodo. Nos inclinamos por utilizar la entrevista porque bastantes usuarios de la aplicación van a ser profesores que seguramente posean una gran capacidad divulgativa por lo que una entrevista abierta es el mejor medio para captar todas las ideas.

De hecho, la entrevista tomará lugar en un centro escolar por lo que se podrá tomar nota de como utiliza y comporta un usuario en el entorno donde utilizará la aplicación.

Para consultar la entrevista ir al siguiente anexo: [Transcripción de entrevista presencial](#).

De la entrevista podemos sacar datos cualitativos muy interesantes como que dado que la administración pública no hace caso a las demandas de los profesores la aplicación tendría una muy buena acogida entre el profesorado.

En general, la idea es que la app sería una buena herramienta para todos, para los profesores, para los alumnos y para los padres de estos. Nos llevamos una idea muy positiva de la entrevista.

Método cuantitativo

Utilizaremos como método cuantitativo la encuesta. Gracias a las respuestas obtenidas en el método cualitativo podemos estructurar las preguntas que se harán en la encuesta en esta segunda fase.

Las encuestas tienen la ventaja en que se pueden realizar de manera online. No es necesario desplazarse a un sitio en concreto con lo que el público que opta a realizar dicha encuesta es más amplio. Otra ventaja es que al constar

de preguntas cerradas los resultados que obtendremos podrán ser clasificados estadísticamente y será realmente fácil extraer conclusiones sobre qué desarrollar en nuestra app.

Para consultar la encuesta ir al siguiente anexo: [Preguntas de las encuestas realizadas.](#)

Gracias a las cinco encuestas que se hicieron, podemos obtener datos estadísticos. Como son:

Estadísticas favorables:

80% de los encuestados tiene un nivel alto del uso de la tecnología, lo que significa que podrían manejar la app sin dificultades.

Hay un 50% de encuestados que utiliza Android y otro 50% iOS lo que fortalece la decisión de haber desarrollado una aplicación híbrida.

85% de los encuestados están enterados de lo que significa el crowdfunding.

90% de los encuestados vería con buenos ojos el desarrollo de una aplicación crowdfunding en el ámbito escolar.

Estadísticas en contra:

85% de los encuestados nunca ha utilizado una aplicación móvil de crowdfunding para realizar una donación

2.1.2 Fichas perfiles

Con toda la información recopilada en la entrevista y encuestas pasamos a elaborar los perfiles que supuestamente utilizarán nuestra app.

Perfil **Colector**

Datos demográficos	
Sexo	Masculino/Femenino
Edad	30 – 55
Nivel estudios	Título universitario
Profesión	Maestro
Nivel uso tecnología	Medio - Alto
Intereses	Enseñanza, divulgación, movimientos sociales
Contexto de uso	
Con colector nos referimos a la persona que publica un anuncio en el que se recolectarán fondos para una causa. Esta persona utilizará la aplicación dentro de la escuela donde observe las carencias o falta de materiales de un aula	

determinado.

Tareas

Estos usuarios deben poder publicar la recolecta en la app. Para ello, deben tener una sección donde puedan crear una colecta de manera sencilla. Deberán poder hacer una fotografía de la necesidad, introducir una descripción, la cantidad que necesiten, etc.

Lo único que necesitarán estos usuarios es disponer de un smartphone que pueda soportar la aplicación (SO Android o iOS).

Ilustración 6 - Perfil Colector

Perfil **Mecenas**

Datos demográficos	
Sexo	Masculino/Femenino
Edad	40 - 70
Nivel estudios	Estudios primeros en adelante
Profesión	Indiferente
Nivel uso tecnología	Medio / Medio - Alto
Intereses	Altruismo, movimientos sociales
Contexto de uso	
<p>Con mecenas nos referimos a la persona que dona a una colecta determinada de manera altruista. Esta persona, generalmente, utilizará la aplicación dentro de su domicilio o también en reuniones de organizaciones benéficas, no gubernamentales, etc.</p> <p>Este perfil abarca más gente que los colectores, el rango de edad es más amplio porque los mecenas suelen ser personas de más avanzada edad donde el poder adquisitivo es mayor.</p>	
Tareas	
<p>Estos usuarios deben poder encontrar distintas publicaciones de necesidades en la app. La app debe tener una sección donde estén publicadas todas las necesidades que estén activas y una vez seleccionada la que interese, poder aportar a la causa. Al ser este perfil de un rango de edad que puede llegar hasta los 70 años el diseño debe ser sencillo y fácil de utilizar.</p> <p>Lo único que necesitarán estos usuarios para poder utilizar la aplicación es disponer de un smartphone que pueda soportar la aplicación (SO Android o iOS).</p>	

Ilustración 7 - Perfil Mecenas

Perfil **Curioso**

Datos demográficos	
Sexo	Masculino/Femenino
Edad	20 - 35
Nivel estudios	Estudios secundarios en adelante
Profesión	Indiferente

Nivel uso tecnología	Alto
Intereses	
Contexto de uso	
Con curioso nos referimos a esa persona que encuentra la app por casualidad y se la instala para ver lo qué es, pero sin tener un objetivo claro.	
Tareas	
Navegarán por la app con el único objetivo de curiosear. La app debe disponer de una sección donde se explique en qué consiste; por ejemplo la sección “¿Quiénes somos?”. También debe disponer de una sección donde sin registrarse se muestre el listado de las colectas disponibles.	

Ilustración 8 - Perfil Curioso

2.2. Diseño conceptual

2.2.1. Escenarios de uso

En los escenarios de uso vamos a ver a un perfil de los descritos anteriormente en una situación real utilizando la aplicación. De esta manera podremos extraer información acerca del diseño o usabilidad de la aplicación que no habíamos tenido en cuenta hasta ahora.

Escenario de uso 1.

Identificador: EU1	
Perfil	Colector
Descripción	
Luis, profesor, 38 años, profesor de Ciencias Naturales del Colegio Público Riomanzanas de Zamora. Ha realizado varias quejas contra la administración pública por falta de materiales adecuados en este aula. En particular, se queja de la necesidad de materiales decentes para impartir Ciencias Naturales, como por ejemplo un esqueleto humano. Cansado de que las administraciones públicas hagan caso omiso a sus quejas decide probar con una nueva aplicación de la que están hablando todos sus compañeros: MeceAula.	
Coge su teléfono con sistema operativo Android, accede al Play Store de Google e instala la aplicación. Comprueba que para publicar una colecta es necesario registrarse en la aplicación. Ahora no está de humor para registrarse por lo que intenta acceder utilizando la cuenta de Google. Una vez autenticado se dirige a la sección “Publicar colecta”, donde describe de forma detallada cuál es su problema y la cantidad que necesitaría para que sus alumnos disfrutaran del esqueleto.	

Ilustración 9 - Escenario de uso 1

Escenario de uso 2.

Identificador: EU2	
Perfil	Mecenas
Descripción	

Ángela, es una abogada de 55 años, especializada en derecho internacional. Lee mucho la prensa, y últimamente existen muchas noticias sobre la dura vuelta al cole que le espera a muchas familias. Ella ya ha utilizado apps de crowdfunding en numeras ocasiones. Ese día, mientras termina de desayunar leyendo la prensa, busca en el App Store de su teléfono iOS alguna aplicación de crowdfunding orientado a escuelas. Para su sorpresa, encuentra MeceAula que contiene todo lo que estaba buscando. Se registra en la aplicación y navega por todas las colectas que hay disponibles hasta que encuentra una que llama la atención. Observa que todavía faltan 40 € para completar objetivo por lo que entra dentro de la colecta, cumplimenta el formulario y realiza la donación.

Ilustración 10 - Escenario de uso 2

Escenario de uso 3.

Identificador: EU3	
Perfil	Curioso
Descripción	
Mariano, es un joven de 25 años que se dedica a la creación de contenido. Siempre está con su teléfono Android buscando nuevas apps con las que entretenerse. Busca las más populares que hay en el Play Store. En esta ocasión le ha llamado la atención el nombre de “MeceAula” y su logo, así que decide instalársela. Una vez instalada, tras navegar por las diferentes pantallas y al haber entrado en la sección “Quiénes somos” se da cuenta que no le interesa y decide desinstalar la aplicación.	

Ilustración 11 - Escenario de uso 3

2.2.1 Flujos de interacción

En el siguiente diagrama se pueden observar los flujos más importantes de la aplicación. Se han omitido aquellos que no tienen relevancia:

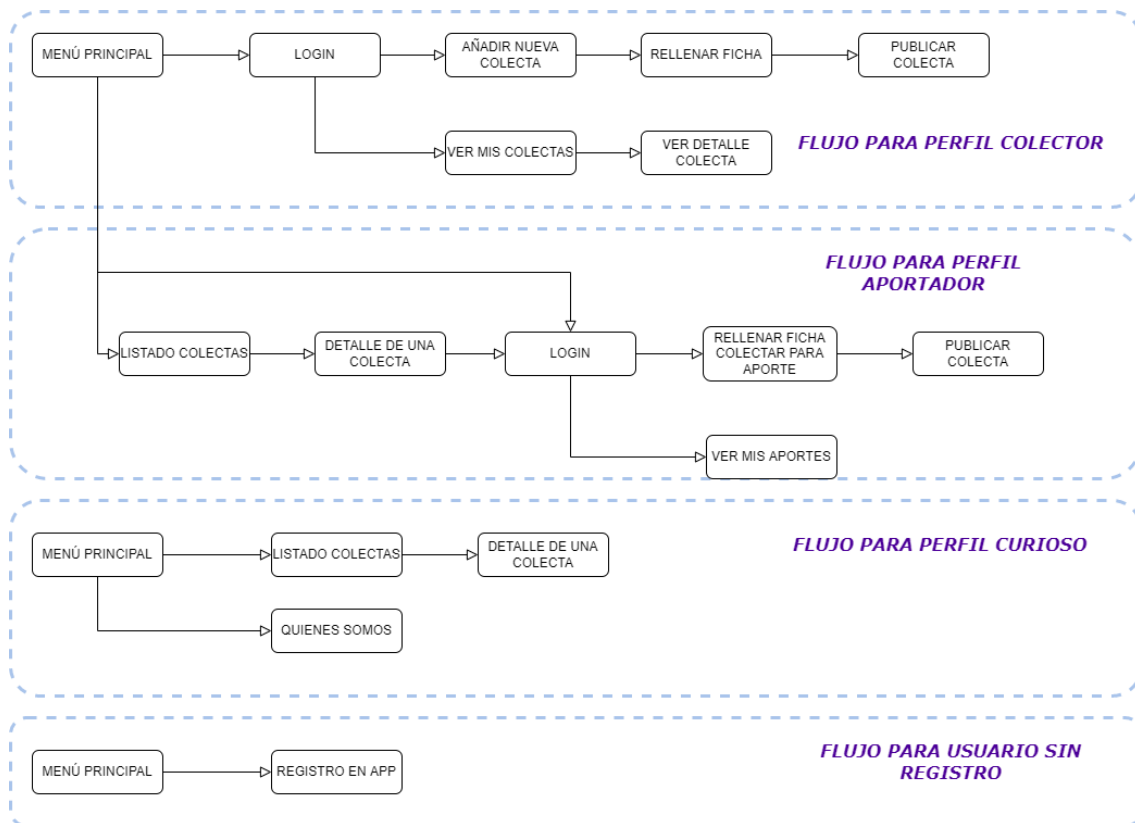


Ilustración 12 - Flujos de interacción

2.3. Prototipado

En esta fase construimos las maquetas del diseño final de nuestra aplicación utilizando toda la información obtenido en etapas anteriores. Estos prototipos se utilizarán posteriormente para que los usuarios evalúen cierta funcionalidad y usabilidad de la app.

2.3.1 Sketches

Son las ideas preliminares del diseño de la aplicación MeceAula. Son los bocetos del diseño de la aplicación. Estos sketches se han realizado a mano:

Menú principal – Pantalla a la que se entra directamente cuando se abre la aplicación. Muestra el listado de todas las colectas disponibles.

Login - Pantalla de autenticación en la aplicación.



Ilustración 13 - Boceto Menú Principal

Ilustración 14 - Boceto Login

Mis aportes – Pantalla donde se le muestra a un usuario de tipo mecenas los aportes que ha realizado.

Colaborar en colecta – Pantalla donde un usuario rellena la ficha para colaborar aportando una cantidad en una colecta.

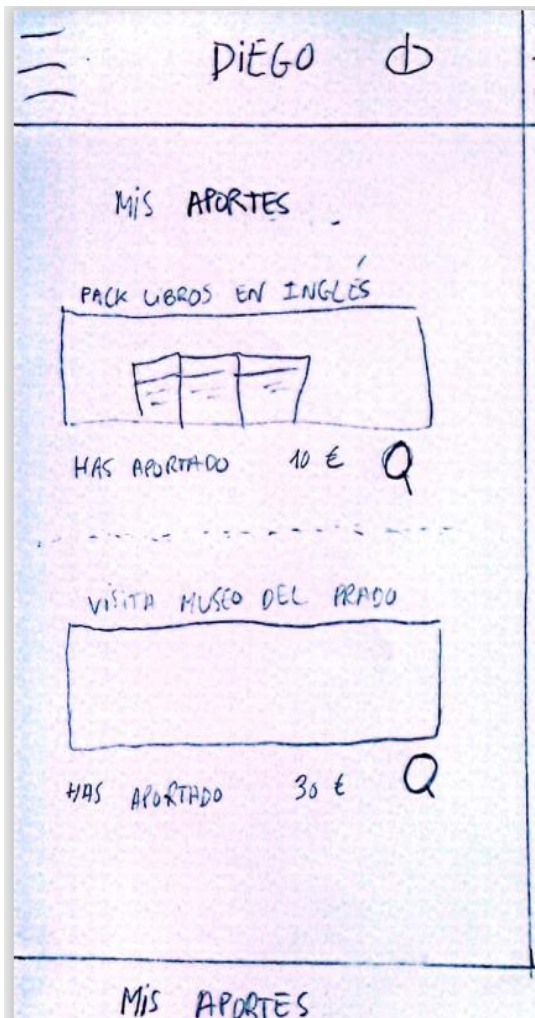


Ilustración 15 - Boceto Mis aportes

Ilustración 16 - Boceto colaborar en colecta

Detalle aporte – En esta pantalla se visualiza el detalle de un aporte realizado en una colecta.

Crear colecta – En esta pantalla se rellena la ficha para crear una colecta.

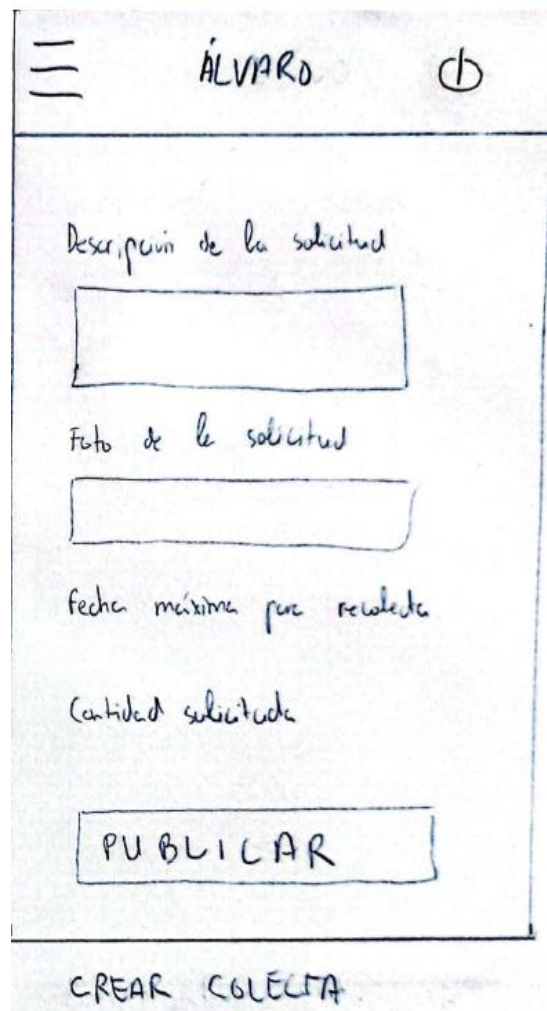
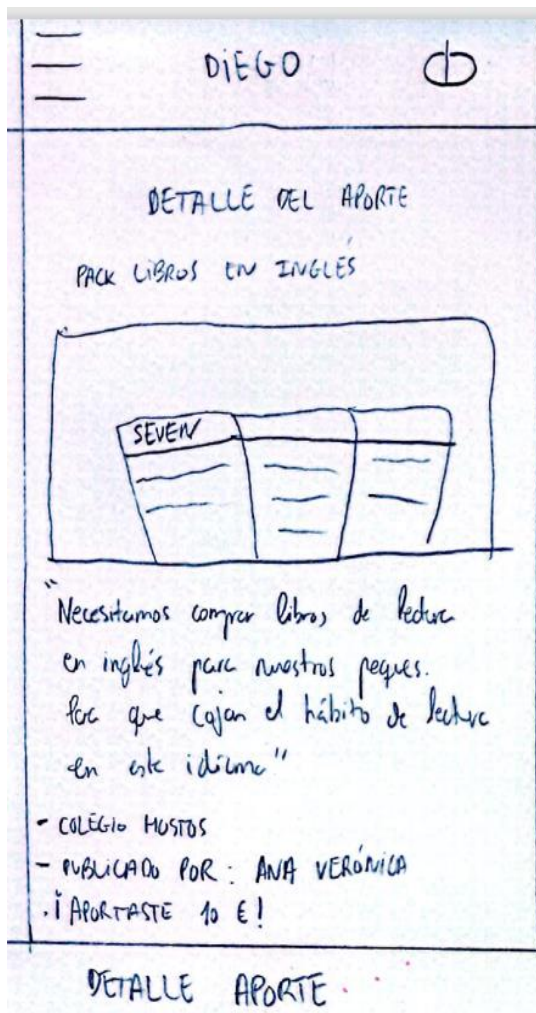


Ilustración 17 - Detalle del aporte

Ilustración 18 - Boceto crear colecta

Mis colectas – Pantalla donde se muestran todas las colectas que ha publicado un usuario.

Mi perfil – Pantalla donde se muestra y se puede editar la información de un perfil.



Ilustración 19 - Boceto Mis colectas

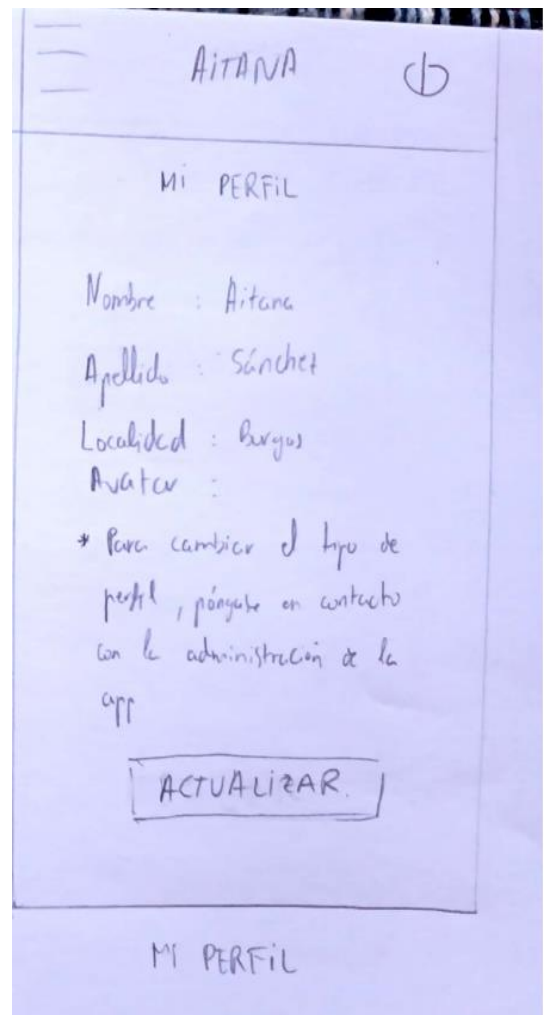


Ilustración 20 - Boceto Mi perfil

Registrar usuario – Pantalla donde se registra un usuario en la aplicación.

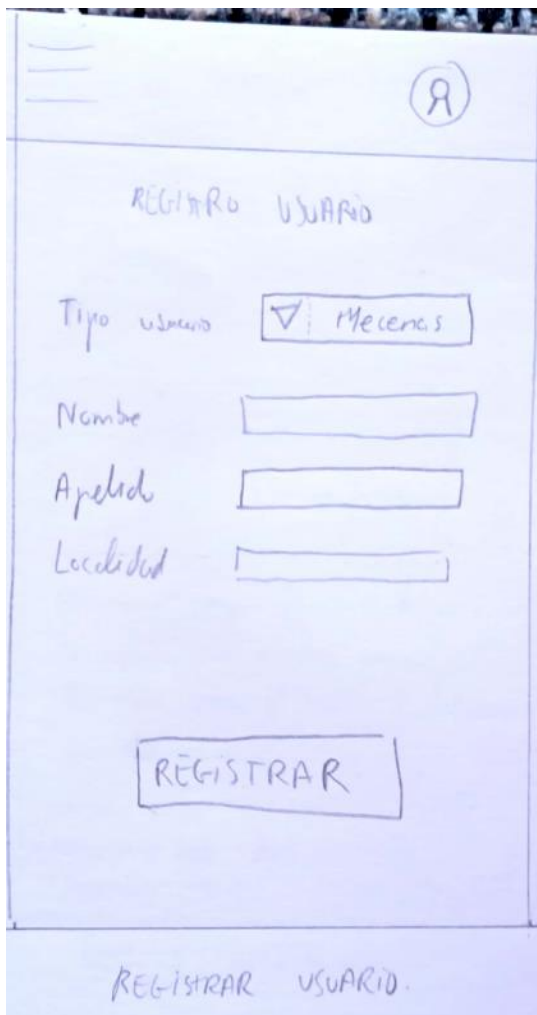


Ilustración 21 - Boceto registrar usuario

2.3.2 Prototipado horizontal

Se trata de un prototipado que se acerca bastante a la realidad del diseño del producto final, pero sin funcionalidad. Este prototipado lo he realizado con la herramienta Balsamiq.

Menú principal

Autenticación



Ilustración 22 - Prototipo Menu Principal



Ilustración 23 - Prototipo Login

Mis aportes



Aportar o colaborar en colecta



Registrar usuario



Mis colectas

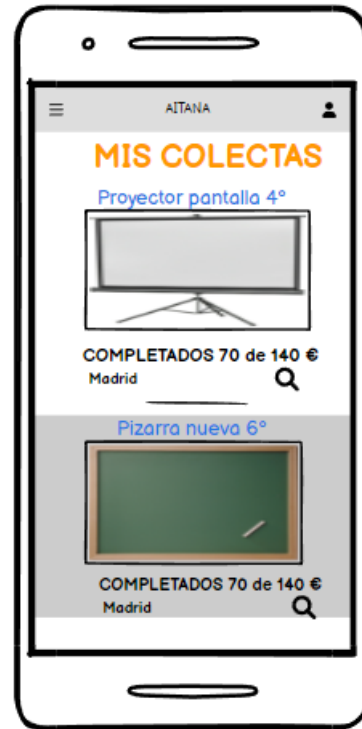


Ilustración 24 - Prototipo Registrar usuario

Ilustración 25 - Prototipo Mis colectas

Publicar colecta



Mis aportes



Detalle de Aporte

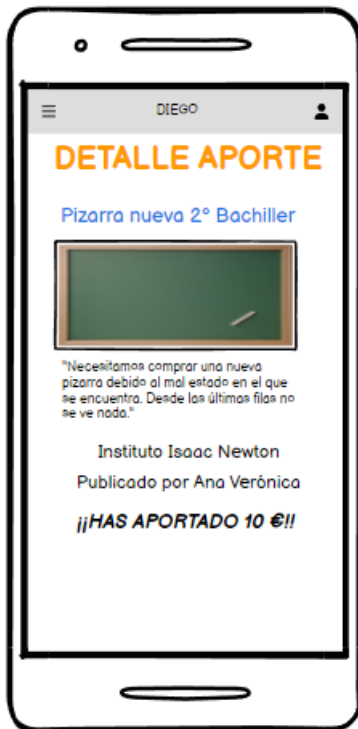


Ilustración 26 - Prototipo Detalle Aporte

Mi perfil

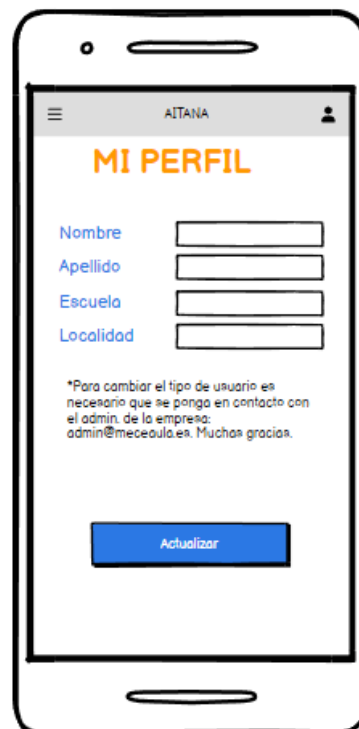


Ilustración 27 - Prototipo Registrar

2.4 Evaluación

Se dan unas directrices básicas de cómo se harían las pruebas de los prototipos con los usuarios. De momento, en esta entrega no se implementa la realización de los test.

Durante el test: Describa en voz alta el proceso de registrar un usuario en la aplicación.

Al final del test: ¿Le ha resultado sencillo? ¿Cuáles son los inconvenientes con los que se ha encontrado?

Durante el test: Describa en voz alta el proceso de publicar una nueva colecta. Recuerde que para este test debe utilizar el perfil Colector.

Al final del test: ¿Le ha resultado sencillo? ¿Cuáles son los inconvenientes con los que se ha encontrado?

Durante el test: Describa en voz alta el proceso de aportar cierta cantidad a una colecta. Recuerde que para este test debe utilizar el perfil Mecenass.

Al final del test: ¿Le ha resultado sencillo? ¿Cuáles son los inconvenientes con los que se ha encontrado?

Durante el test: Describa en voz alta el proceso de modificar su perfil.

Al final del test: ¿Le ha resultado sencillo? ¿Cuáles son los inconvenientes con los que se ha encontrado?

3. Diseño técnico de la aplicación

3.1 Definición de los casos de uso

Identificador: CU-001	
Nombre	Dar alta usuario
Prioridad	Alta
Descripción	El usuario debe poder darse de alta en la aplicación. Esto es básico para poder realizar acciones como aportar en una colecta o crear una nueva.
Actores	Usuarios que no están registrados en la aplicación.
Precondiciones	Ninguna
Iniciado por	Pulsar botón regístrate
Flujo	<ul style="list-style-type: none">• Acceder a la aplicación.• Ir a la pantalla de autenticación.• Pulsar en el botón regístrate.• Rellenar el formulario de nuevo usuario.• Pulsar el botón "Registrar".
Postcondiciones	El nuevo usuario quedará registrado tanto en firebase como en la base de datos.
Notas	

Ilustración 28 - Identificador: CU-001

Identificador: CU-002	
Nombre	Añadir nueva colecta
Prioridad	Alta
Descripción	El usuario debe poder dar de alta una colecta.
Actores	Usuarios con perfil colector.
Precondiciones	Estar autenticado en la aplicación.
Iniciado por	Pulsar botón "crear colecta".
Flujo	<ul style="list-style-type: none">• Acceder a la aplicación.• Autenticarse.• Pulsar en el botón crear colecta.• Rellenar el formulario de la colecta.• Pulsar el botón "Publicar".

Postcondiciones	La nueva colecta quedará registrada en base de datos. Esta colecta se visualizará en el listado de colectas.
Notas	

Ilustración 29 - Identificador: CU-002

Identificador: CU-003	
Nombre	Aportar a una colecta
Prioridad	Alta
Descripción	El usuario debe poder colaborar con una donación en una colecta.
Actores	Usuarios con perfil Mecenass.
Precondiciones	Estar autenticado en la aplicación. Debe existir una colecta.
Iniciado por	Pulsar botón "crear colecta".
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación. • En el listado de colectas, escoger una donde realizar el aporte. • Acceder al detalle de dicho aporte. • Autenticarse en la aplicación. • Rellenar el formulario de colaborar con un aporte a una colecta. • Pulsar el botón "Aportar".
Postcondiciones	La nueva colecta quedará registrada en base de datos. Esta colecta se visualizará en el listado de colectas.
Notas	

Ilustración 30 - Identificador: CU-003

Identificador: CU-004	
Nombre	Actualizar perfil
Prioridad	Baja
Descripción	El usuario debe poder actualizar los datos de su perfil desde la aplicación.
Actores	Usuarios con perfil Colector y Mecenass.
Precondiciones	Estar autenticado en la aplicación.
Iniciado por	Pulsar botón "Mi perfil".
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación. • Autenticarse en la aplicación. • Acceder a la sección mi perfil. • Modificar algún dato existente. • Rellenar el formulario de colaborar con un aporte a una colecta. • Pulsar el botón "Actualizar"
Postcondiciones	Los datos del perfil se actualizarán en la bbdd.
Notas	

Ilustración 31 - Identificador: CU-004

Identificador: CU-005	
Nombre	Listar colectas
Prioridad	Baja
Descripción	Cualquier usuario debe poder visualizar todas las colectas que están disponibles en la aplicación
Actores	Cualquier usuario
Precondiciones	Ninguna
Iniciado por	Ninguno
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación. • Visualizar todas las colectas de la aplicación
Postcondiciones	Los datos del perfil se actualizarán en la bbdd
Notas	El punto de entrada de la aplicación son todas las colectas que hay disponibles

Ilustración 32 - Identificador: CU-005

Identificador: CU-006	
Nombre	Iniciar sesión
Prioridad	Media
Descripción	Cualquier usuario registrado en la aplicación puede autenticarse
Actores	Cualquier usuario
Precondiciones	El usuario debe estar registrado en la aplicación
Iniciado por	Pulsar botón Iniciar sesión
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación • Ir a la sección Iniciar Sesión • Introducir usuario y contraseña • Pulsar botón "Iniciar sesión"
Postcondiciones	Se indicará de algún modo que hemos iniciado sesión correctamente.
Notas	

Ilustración 33 - Identificador: CU-006

Identificador: CU-007	
Nombre	Cerrar sesión
Prioridad	Baja
Descripción	Cualquier usuario autenticado en la aplicación puede terminar su sesión
Actores	Cualquier usuario
Precondiciones	El usuario debe estar autenticado en la aplicación
Iniciado por	Pulsar botón Cerrar sesión
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación. • Desplegar menú lateral • Pulsar botón "Cerrar sesión"
Postcondiciones	La aplicación debe estar en la pantalla de login
Notas	

Ilustración 34 - Identificador: CU-007

Identificador: CU-008	
Nombre	Visualizar mis aportes
Prioridad	Media
Descripción	Visualizar los aportes realizados por un usuario
Actores	Cualquier usuario
Precondiciones	El usuario debe estar autenticado en la aplicación
Iniciado por	Pulsar botón Mis aportes
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación. • Iniciar sesión • Desplegar menú lateral • Pulsar botón “Mis aportes” • Visualizar los aportes realizados
Postcondiciones	
Notas	

Ilustración 35 - Identificador: CU-008

Identificador: CU-009	
Nombre	Visualizar mis colectas
Prioridad	Media
Descripción	Visualizar las colectas creadas por un usuario
Actores	Cualquier usuario
Precondiciones	El usuario debe estar autenticado en la aplicación
Iniciado por	Pulsar botón Mis colectas
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación • Iniciar sesión • Desplegar menú lateral • Pulsar botón “Mis colectas” • Visualizar las colectas realizadas
Postcondiciones	
Notas	

Ilustración 36 - Identificador: CU-009

Identificador: CU-010	
Nombre	Detalle colecta
Prioridad	Media
Descripción	Visualizar el detalle de una colecta
Actores	Cualquier usuario
Precondiciones	Ninguna
Iniciado por	Pulsar en una de las colectas disponibles
Flujo	<ul style="list-style-type: none"> • Acceder a la aplicación • Seleccionar una colecta • Visualizar detalle de una colecta
Postcondiciones	
Notas	

Ilustración 37 - Identificador: CU-010

3.2 Diseño de la arquitectura

3.2.1 Diagrama UML correspondiente al diseño de la base de datos

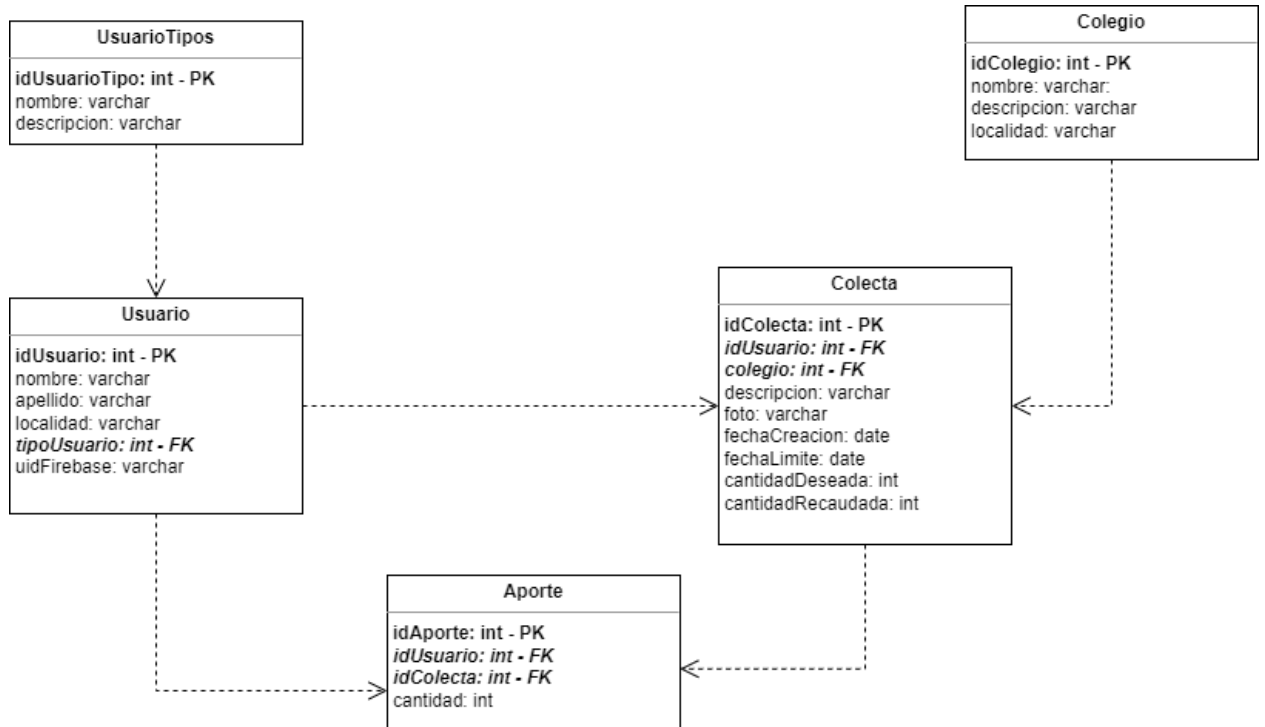


Ilustración 38 - Diagrama UML

3.2.2 Diagrama explicativo de la arquitectura del sistema

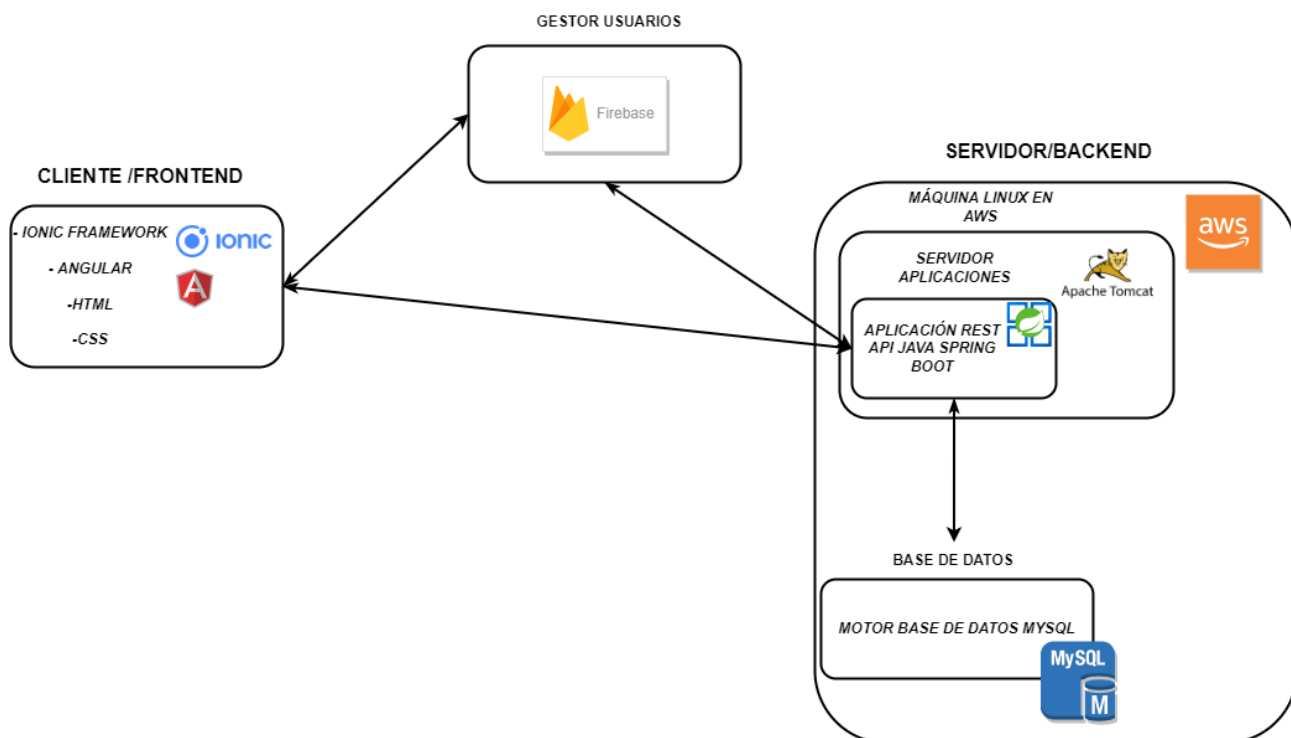


Ilustración 39 - Diagrama arquitectura del sistema

Lo que muestra el gráfico son las interacciones entre los distintos componentes de nuestra aplicación.

El cliente hará peticiones contra Firebase para autenticarse y hará peticiones contra el servidor para obtener la información necesaria para cada una de sus vistas.

El servidor está montado en una máquina AWS con SO Amazon Linux 2. Dentro de esta máquina, se encuentra instalado el contenedor de aplicaciones Java, Apache Tomcat, que es donde se encuentra nuestro aplicativo servidor. Este aplicativo servidor se comunicará con la base de datos para obtener la información pertinente. La base de datos también está en AWS.

Un ejemplo práctico de ver como funciona un flujo en la arquitectura sería “Obtener todas las colectas”:

Un usuario entra en la sección “Colectas”, por lo tanto, se envía una petición al servidor demandando esas colectas.

La aplicación servidor dentro del contenedor apache tomcat recepciona esta petición, la evalúa, y realiza la petición a la base de datos para obtener todas las colectas. Una vez las tiene, las devuelve a la aplicación servidor del tomcat y después las envía de vuelta al cliente.

Una vez que el cliente tiene las colectas, las establece en la vista para su visualización.

4. Desarrollo e implementación

4.1 Estructura del proyecto.

La aplicación está separada en dos grandes bloques. Uno de ellos es la parte cliente o front end que es la aplicación móvil (artefacto apk o ipa) y el otro bloque es la parte servidor o back end, que es la parte que recibe peticiones por parte del cliente y devuelve información. Cada uno de estos bloques tiene un desarrollo independiente que hace ser a la aplicación totalmente escalable. Si un día se quedase obsoleta la parte front end de nuestro proyecto sólo deberíamos sustituir la parte front end.

4.1.1 Parte cliente o front end

El front end se ha desarrollado con Ionic Framework, que es un conjunto de herramientas UI (User Interface) que nos permite desarrollar una aplicación nativa híbrida (iOS y Android) utilizando tecnología web: HTML, CSS y JavaScript. Ionic Framework se integra fácilmente con los framework web más modernos del mercado: Angular, Vue y React.

Por otra parte, la gran ventaja de Ionic es que con un solo desarrollo obtienes un artefacto para iOS y otro para Android. No hay que mantener dos desarrollos paralelos, por lo que es perfecto para esta aplicación multiplataforma.

Las tecnologías utilizadas en la parte front son Ionic Framework 6 + Angular 15.

Siguiendo diversas guías de estilo tanto de fuentes oficiales como de usuarios experimentados, se ha decidido utilizar la siguiente estructura de carpetas:

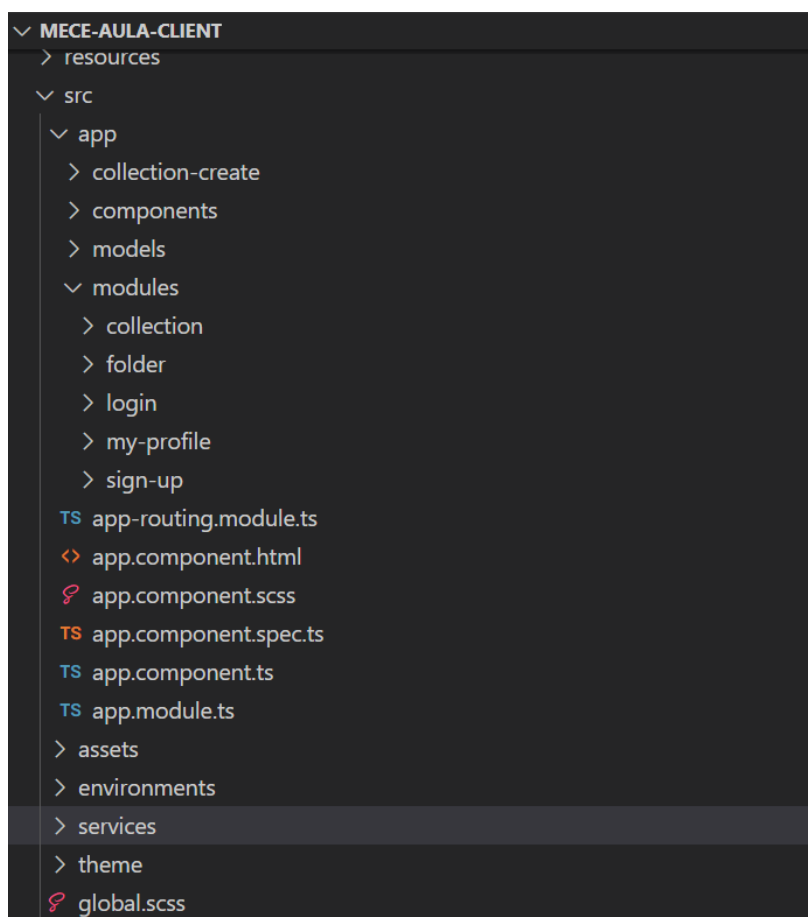


Ilustración 40 - Estructura carpetas front end

Contenido de las carpetas:

Components:

Suelen ser aquellos componentes que se utilizan más de una vez en diferentes lugares del proyecto. Por ejemplo, una cabecera con un título se utilizará en casi todas las pantallas de la aplicación (con diferentes parámetros).

Models

Ficheros o clases que representan el esquema de una tabla de base de datos en la aplicación cliente. Ejemplo: User.

Modules

Ficheros que representan cada una de las partes en las que se divide la aplicación. En nuestro caso: Aportes, colectas, autenticación, login, etc.

Assets

Recursos utilizados por la aplicación. Pueden ser imágenes de logos, iconos, ficheros de traducciones, etc.

Environments

Ficheros que representan variables globales de la aplicación con respecto al dominio al que tienen que apuntar, servicios, etc.

Existen dos ficheros: `environment.prod.ts` y `environment.ts`. Cada uno de ellos se utilizará dependiendo del objetivo de la compilación: artefacto para desarrollo o producción.

Útil

Funciones para utilizar en varios puntos de la aplicación. Por ejemplo, funciones para formatear salidas de datos.

4.1.2 Parte servidor o backend

El backend se ha desarrollado en Java utilizando Spring Framework, el framework más reconocido de Java para aplicaciones web. Se ha utilizado Spring Boot que permite crear aplicaciones Spring de manera sencilla desde cero y además posee un tomcat en el que desplegar la aplicación mientras desarrollas en local. Spring también ofrece opciones de seguridad que son muy fáciles de implementar. Por último, cabe destacar que tiene una amplia cantidad de documentación en sus canales oficiales y una gran comunidad de gente soportándola.

El gestor de proyectos y dependencias para el proyecto servidor es Maven que se integra a la perfección en un proyecto de Spring.

Las tecnologías utilizadas en la parte back son: Java 1.8 + Spring Boot 2.7.6 + Spring Security.

Siguiendo diversas guías de estilo tanto de fuentes oficiales como de usuarios experimentados, se ha decidido utilizar la siguiente estructura de carpetas:

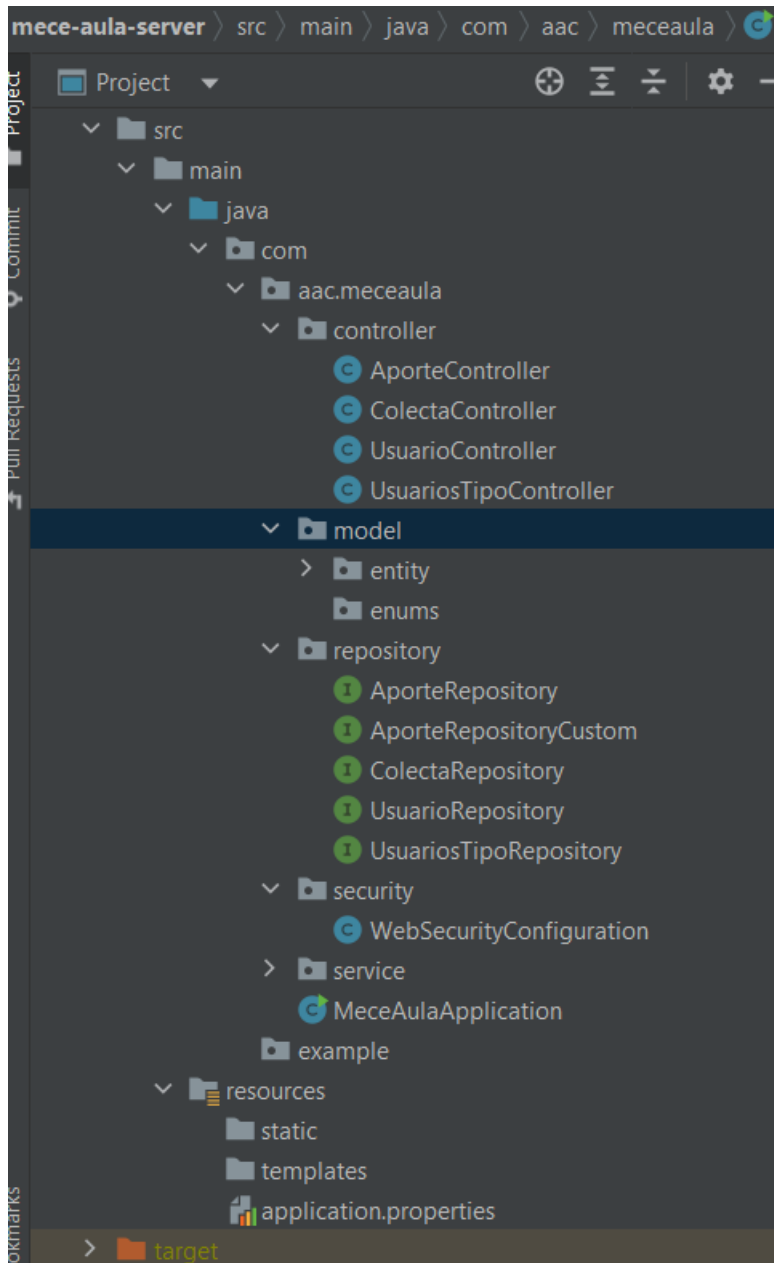


Ilustración 41 - Estructura carpetas back end

Contenido de las carpetas:

Controller:

Son las clases donde están los end points, es decir, las clases que reciben las peticiones de un cliente y las procesan hacia el servidor.

Model

Ficheros o clases que representan las tablas de base de datos en el servidor.

Repository

Son las clases que, con ayuda de las clases models, se comunican con la base de datos y obtienen la información.

Service

Ficheros donde se almacena la lógica de negocio del servidor.

Security

En esta carpeta se encuentran los ficheros de configuración de la seguridad del servidor. Ficheros donde se puede establecer que end points están abiertos, a todo el mundo cuáles cerrados, el tipo de fichero al que tendrán acceso las peticiones de los clientes, etc.

Resources

Contiene recursos estáticos que se utilizarán en la aplicación como por ejemplo plantillas de errores.

Y también tiene un fichero de configuración donde se establece contra qué base de datos hay que conectarse, la ruta donde se servirá la aplicación, etc. Este fichero es el conocido App.properties.

4.2 Entornos de programación.

Se han utilizado dos entornos diferentes de programación distintos para cada parte del proyecto.

Para la parte front-end se ha utilizado VisualCode, que es totalmente gratuito:

<https://code.visualstudio.com/>

Mientras que para la parte back-end se ha utilizado la versión gratuita de IntelliJ (Community Edition):

<https://www.jetbrains.com/es-es/idea/download/#section=windows>

Se acopla muy bien para proyectos con Maven y Java.

4.3 Repositorios

El código fuente de ambas partes (front end y back end) se está almacenando en repositorios de la plataforma GitHub. Ambos repositorios están marcados

como privados; ningún usuario no autorizado puede ver el contenido del código:



Ilustración 42 - Repositorios meceaula

En cada repositorio se indica cual es el lenguaje de programación más utilizado.

Para agregar o modificar contenido del repositorio lo hacemos sobre la rama por defecto máster. Existen guías sobre diversas metodologías de trabajo en Git. Sin embargo, al haber trabajado una sola persona en este proyecto, se ha utilizado la forma tradicional: trabajar sobre una única rama, es decir, subir todos los cambios a dicha rama.

A continuación, podemos observar el detalle de uno de los repositorios utilizados:

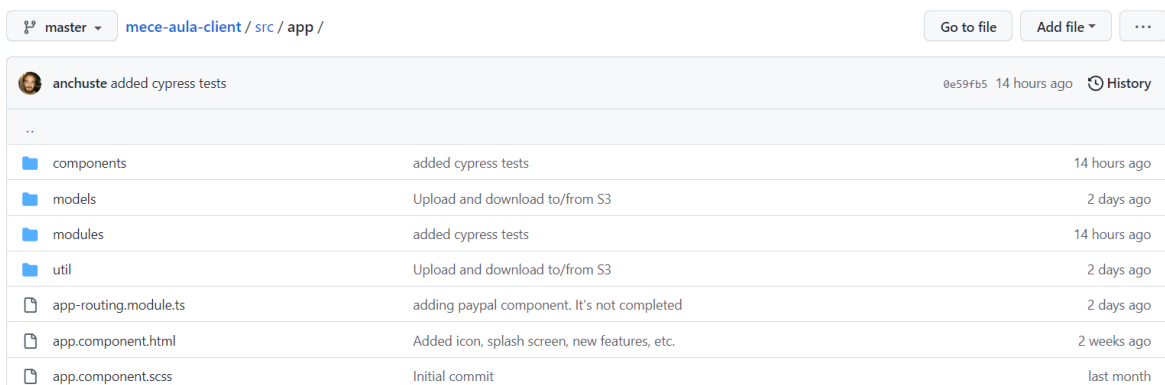


Ilustración 43 - Repositorio app meceaula

Se puede observar cuándo ha sido la última vez que se ha tocado un fichero y la descripción que se le dio a esta modificación.

4.4 Servicios y Aplicaciones

4.4.1 Vmware

El desarrollo e implementación de la aplicación se ha desarrollado dentro de una máquina virtual desde cero, para que ninguna aplicación existente en el entorno de trabajo influya en el proyecto pudiendo afecta a la compilación, rendimiento, etc. Es decir, dentro de esta máquina virtual se ha instalado Windows 10, Visual Studio Code, IntelliJ, Nodejs, Angular, Maven, etc.

La máquina virtual instalada es de la empresa software Vmware especialista en virutalización y computación en la nube.

A continuación, se muestra la configuración y las especificaciones técnicas de la máquina virtual:

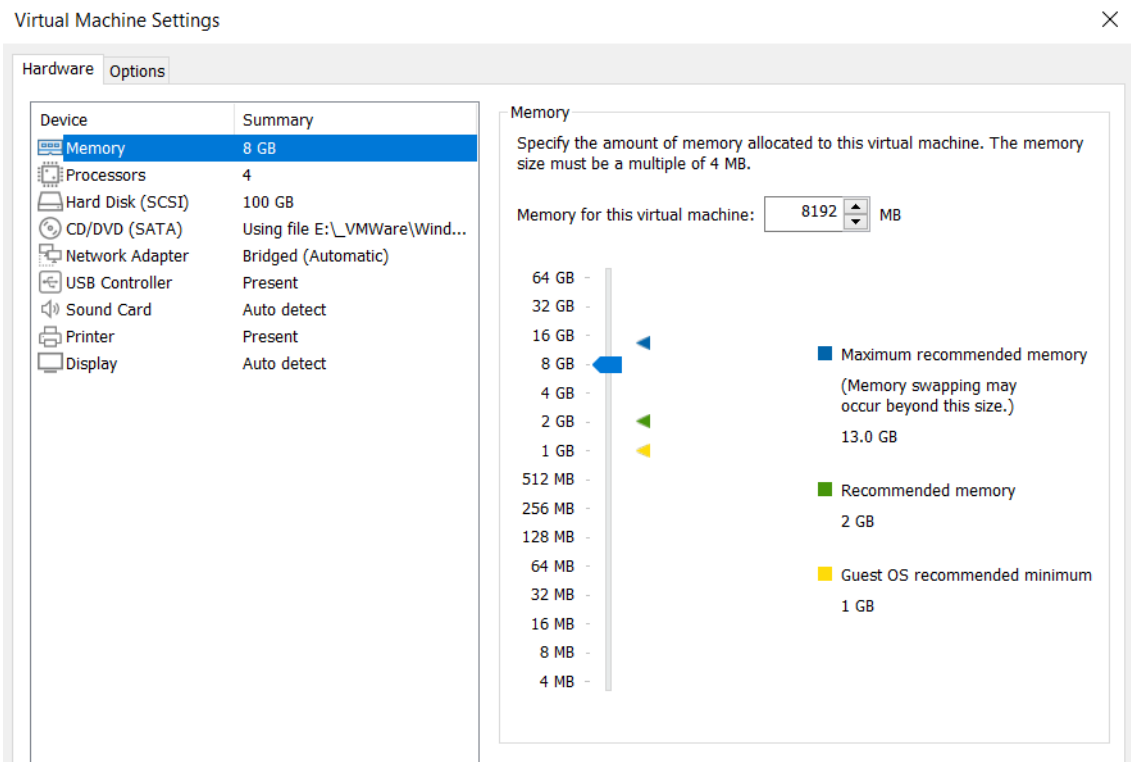


Ilustración 44 - Virtual Machine settings

4.4.2 Postman

Es una aplicación para testear APIs. Se crean peticiones de manera sencilla y se envían a la URL deseada.

En este proyecto se ha utilizado para probar algunas llamadas a Firebase y a nuestro servidor mece-aula-server. Es una gran herramienta ya que permite lanzar llamadas al servidor sin tener montado un cliente. Una vez que se comprueba desde Postman cómo se deben hacer las peticiones al servidor se implementa en nuestro cliente. Soporta diferentes tipos de peticiones POST, GET, PUT, DELETE, etc.

La versión utilizada de Postman ha sido: v10.6.0

A continuación, se muestran algunos ejemplos de peticiones utilizadas:

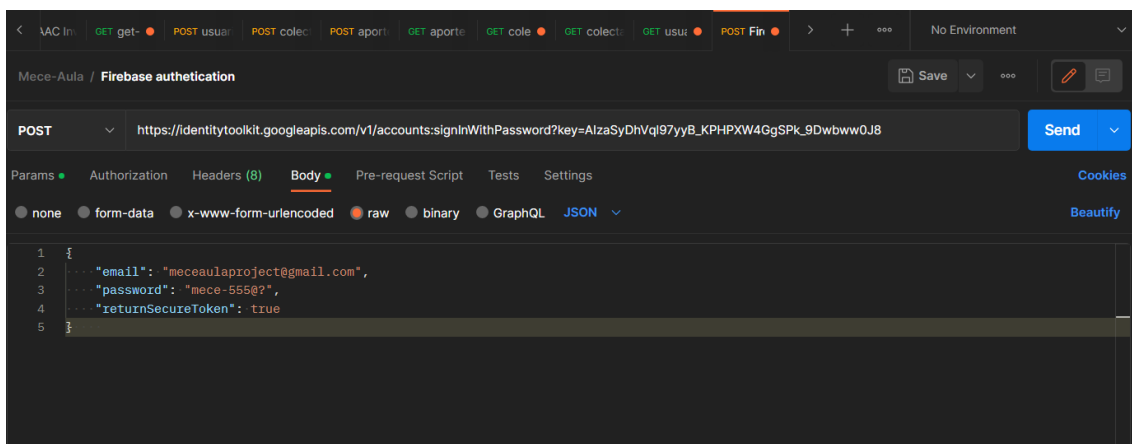


Ilustración 45 - Postman example

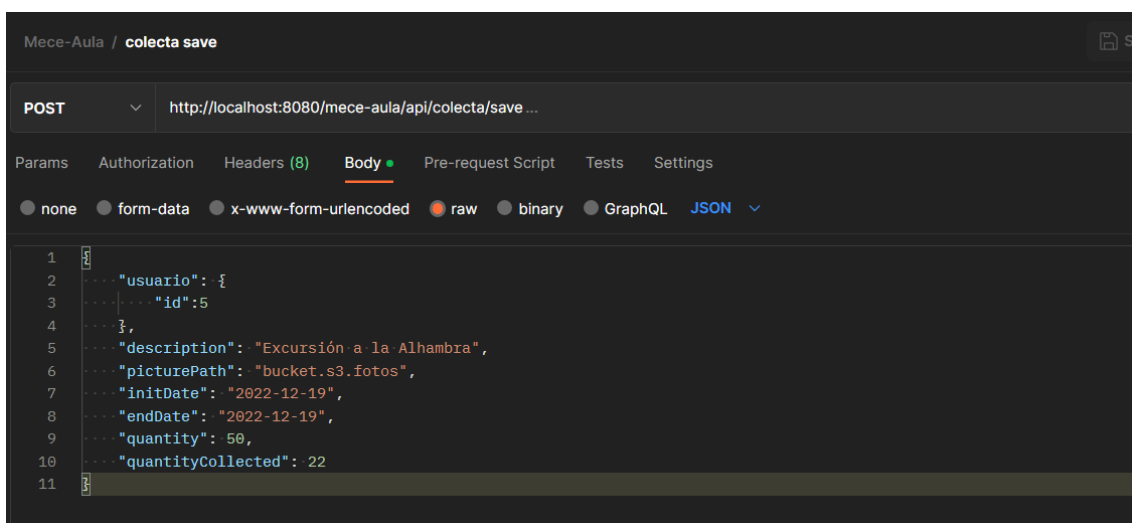


Ilustración 46 - Postman examle 2

4.4.3 Firebase

Se ha creado una cuenta de Firebase para la gestión de usuarios. Firebase proporciona una API que se integra a la perfección con Ionic y Angular. Se ha realizado toda la parte de autenticación mediante correo electrónico.

Por otra parte, se ha securizado el servidor mediante el token de Firebase. Si no se realizan las llamadas al back-end con un token que se obtiene tras autenticarse el servidor no devolverá los datos.

4.4.4 AWS

El servidor de aplicaciones (Tomcat) y la base de datos (MySQL) se han montado en máquinas de AWS.

El servidor de aplicaciones, que es el contenedor (Tomcat) donde se suben los artefactos generados en el proyecto de la parte servidor, se ha montado en una máquina AWS EC2 con el sistema operativo Amazon Linux 2:

<https://aws.amazon.com/es/amazon-linux-2/>

El servidor Tomcat instalado en esta máquina es la versión: apache-tomcat-9.0.70

Por otra parte, la base de datos se ha montado con el servicio de AWS, RDS, la cuál permite levantar una base de datos en pocos minutos:

<https://aws.amazon.com/es/rds/>

El motor de la base de datos es: MySQL 8.0.21

4.5 Dominios

Se ha realizado la compra del dominio aac-software.com en la plataforma godaddy.com.

Dentro de este dominio se han establecido dos nuevos DNS para la aplicación:

mece-aula.aac-software.com: Está asociado a la máquina servidor de AWS de producción.

mece-aula-pre.aac-software.com: Está asociado a la máquina servidor de AWS de preproducción.

¿Por qué es importante tener asociado un dominio a la máquina del servidor?

Puede ser que en una de estas máquinas de AWS se produzca un reinicio o reinicio. Cuando esto se produce es posible que la dirección IP de estas máquinas cambie. Cuando hagamos peticiones a la dirección IP anterior desde nuestro cliente no vamos a encontrar nuestra máquina y por lo tanto el aplicativo no va a funcionar. Sin embargo, si tenemos un dominio asociado nuestras peticiones se realizarán contra ese nombre sin importar la IP que tengan por debajo.

4.6 Despliegue de nueva versión de aplicación

Para realizar el despliegue en producción de la aplicación se realiza la siguiente lista de tareas:

- Se genera artefacto .war de la aplicación servidor.
- Nos conectamos al Tomcat de preproducción y paramos la aplicación mece-aula.
- Desplegamos el nuevo artefacto .war en el Tomcat.
- Comprobamos que la aplicación funciona correctamente en el servidor de pre-producción: mece-aula-pre.aac-software.com.
- Se realizan las pruebas de cypress (framework de testeo) con la parte cliente del aplicativo apuntando al Tomcat de pre-producción.
- Una vez validadas las pruebas, se despliega el artefacto en el Tomcat de producción: mece-aula.aac-software.com.
- Se genera el artefacto de la parte cliente y se distribuye.

4.7 Principales dificultades en la implementación

A continuación, se detallan situaciones que han dificultado o han retrasado el proyecto:

Montar la infraestructura AWS

Aunque se disponía de unos conocimientos básicos de cómo funcionan los servicios de Amazon en la nube, se ha requerido, aproximadamente, el doble de tiempo pensado para montar toda la infraestructura.

Al tener dos máquinas contratadas, una para alojar el apache tomcat, donde va la aplicación servidor y otra donde se aloja la base de datos ha habido algún problema en configurar las reglas de seguridad que permitiese a la aplicación servidor comunicarse con la máquina RDS que contiene la base de datos.

Configurar Tomcat

En este paso, no se tenía experiencia previa, lo que se ha hecho notar a la hora de configurar un Apache Tomcat.

Recordemos que la aplicación servidor, genera un artefacto .war que sube al Apache Tomcat y este lo hace funcionar.

Mientras la aplicación de la parte servidor se está desarrollando en local, todo funciona perfecto, ya que Spring Boot, consta de un Tomcat interno. Cuando se despliega la aplicación en desarrollo, se despliega automáticamente en el Tomcat de desarrollo.

Para configurar el Tomcat en producción se utilizaron diversos tutoriales vistos en la web. Están detallados en la bibliografía.

Una anécdota que sucedió fue el tópico de “en local funciona y en producción no”.

Como he comentado anteriormente, ejecutando el servidor en local (spring boot tiene su propio tomcat) la aplicación funcionaba perfectamente y, cuando se generaba el .WAR y se subía al tomcat de producción, no había manera. En eso se perdió pues prácticamente un día o dos.

Lo que ocurría es que el .WAR se estaba generando en una versión de Java determinada 1.7 mientras que el Tomcat instalada en producción utilizaba Java 1.8. El error que lanzaba, obviamente, no era: “El .WAR está compilado en una versión que no soporta el Tomcat actual”. No daba pistas de lo que podía ser.

Pagos con Paypal

Siempre que se intenta implementar un sistema de pago en una aplicación es una tarea complicada. En esta ocasión no iba a ser diferente.

La documentación oficial de Paypal para implementar la solución en Angular e Ionic es bastante escasa por no decir nula. Se ha tenido que utilizar una librería de Nodejs para solucionarlo y se ha tenido que abrir una cuenta de desarrollador de Paypal.

Aún así se ha implementado la solución para el entorno de desarrollo de Paypal, por lo tanto, habría que invertir más tiempo para implementar el entorno de producción.

Si se quiere realizar un pago en la aplicación con Paypal, estas son las credenciales para poder hacerlo funcionar:

Dirección correo electrónico:

sb-f47zv624322379@personal.example.com

Contraseña:

Gv.MzI0)

4.8 Desviaciones

Al finalizar el [Hito PEC 3 – Implementación](#), todavía faltaba por desarrollar las siguientes tareas:

Tarea	Horas estimadas
Interfaces gráficas	
Mis aportes	4
Detalle aporte	4
Mis colectas	4
Mi perfil	4
Mejorar el aspecto visual de la aplicación	8
Cuando se realice una aportación, sea real, te lleve a la plataforma paypal y se realice el pago.	16
Tests	12
Total	60

Ilustración 47 - Estimación desviación

Como mostraba la tabla faltaban por desarrollar los tests, es otra de las cosas en las que no se pensó al iniciar al desarrollo y eran cruciales. Si se hubiera optado por una metodología TDD, los tests se hubieran entregado en el tiempo correcto.

**Estas tareas se incorporaron y se realizaron en el [Hito Entrega final](#).

A continuación, se muestran unas capturas del estado del proyecto actual:

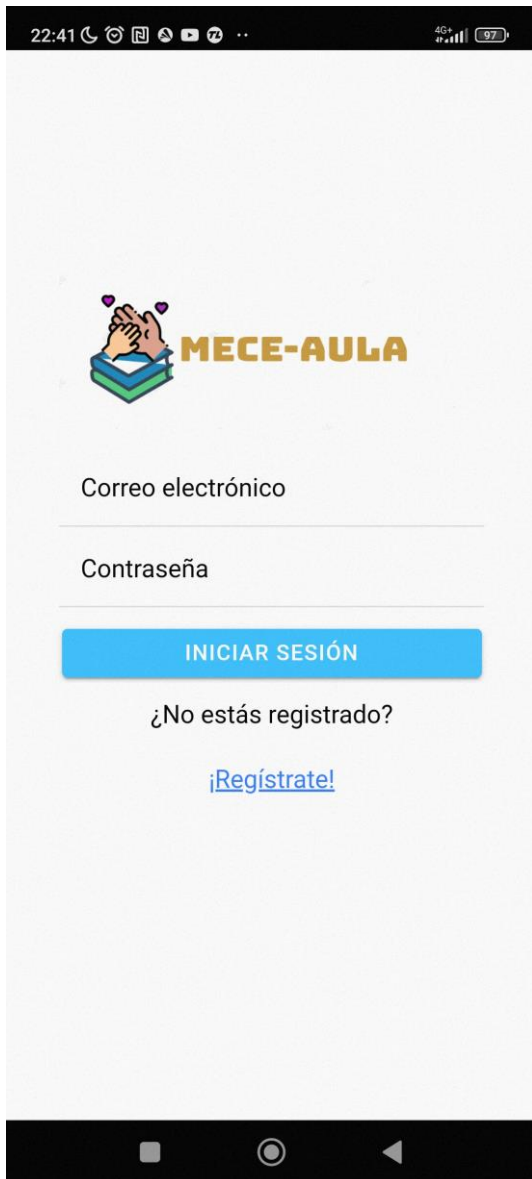


Ilustración 48 - Emulador - Iniciar Sesión

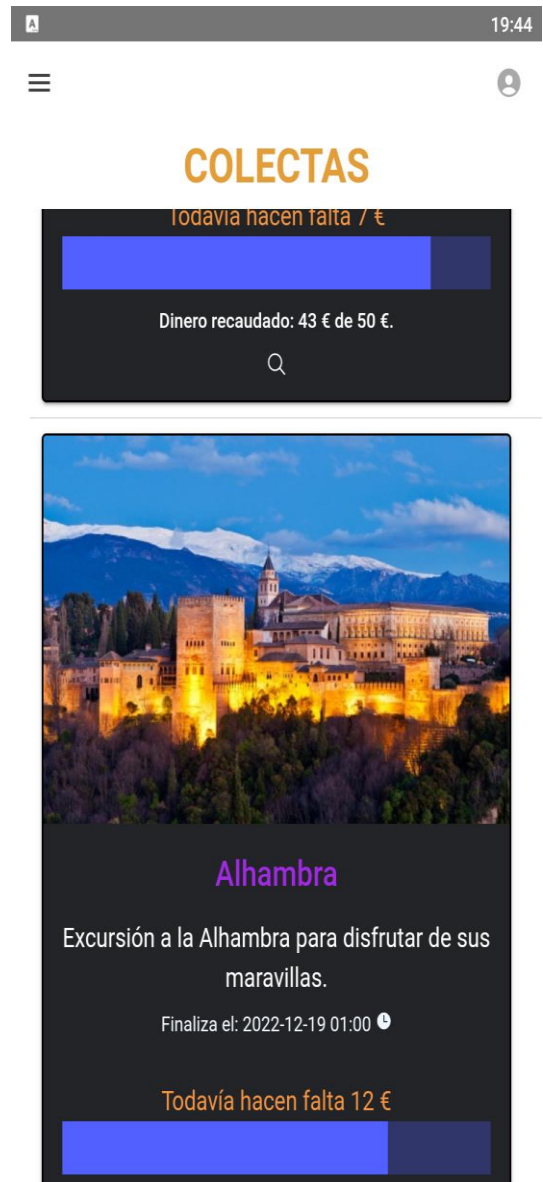


Ilustración 49 - Emulador - Listado colectas

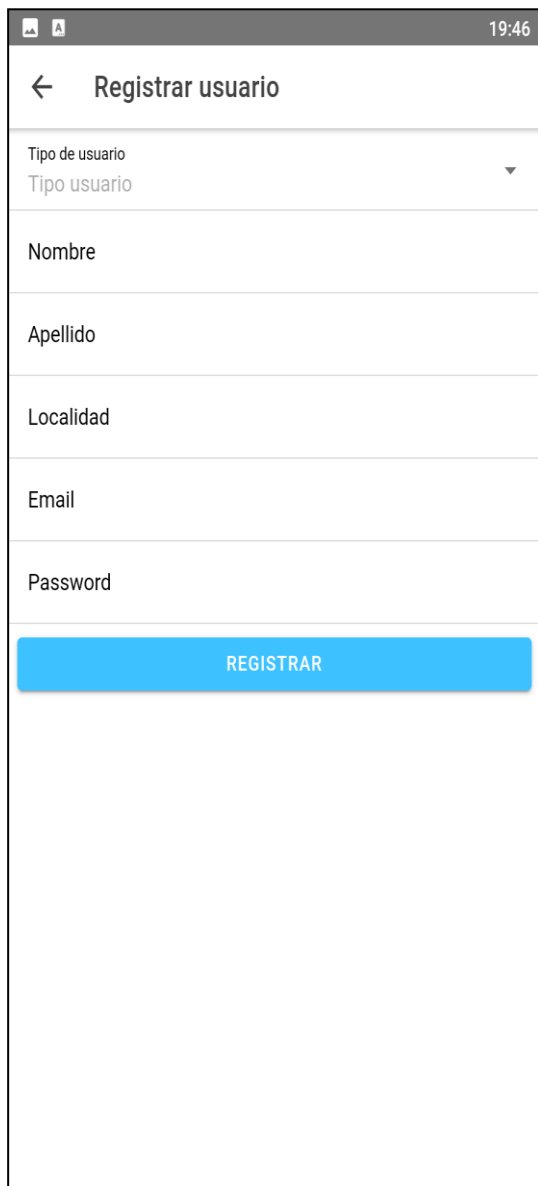


Ilustración 50 - Emulador - Registrar usuario

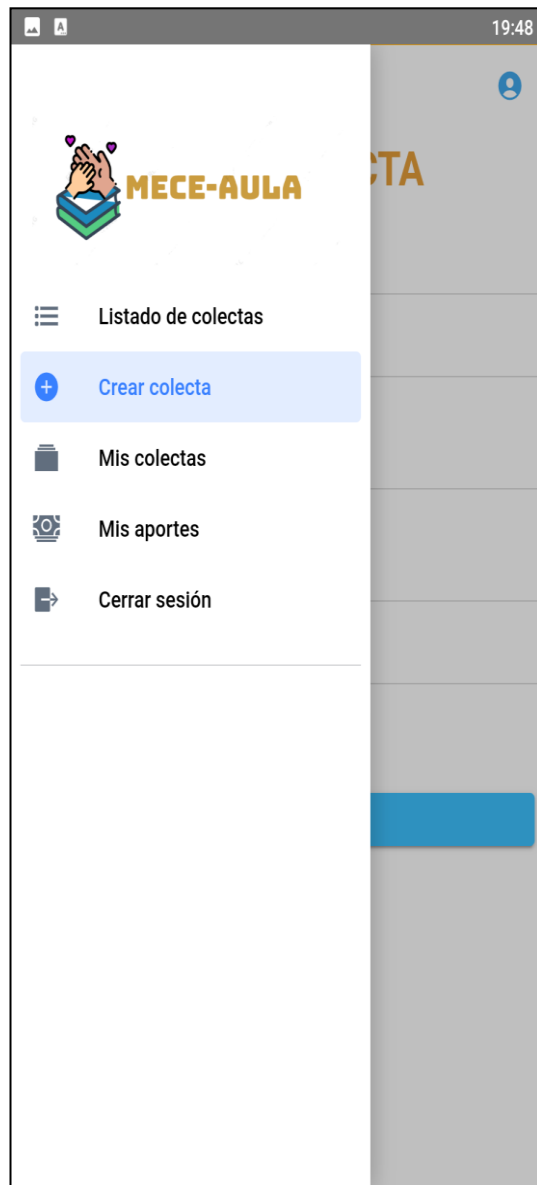


Ilustración 51 - Emulador - panel lateral

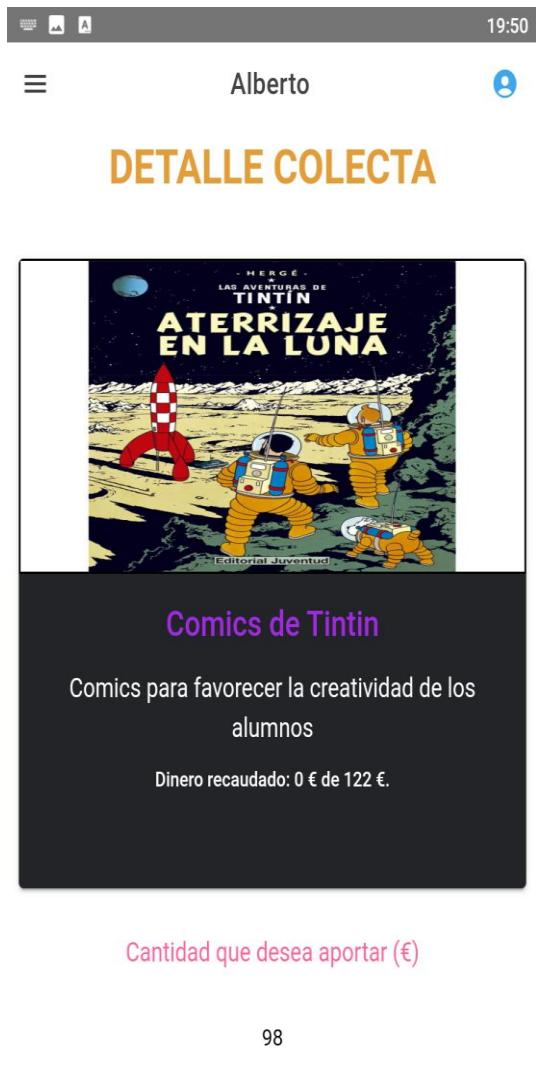


Ilustración 52 - Emulador Aporte/Detalle colecta

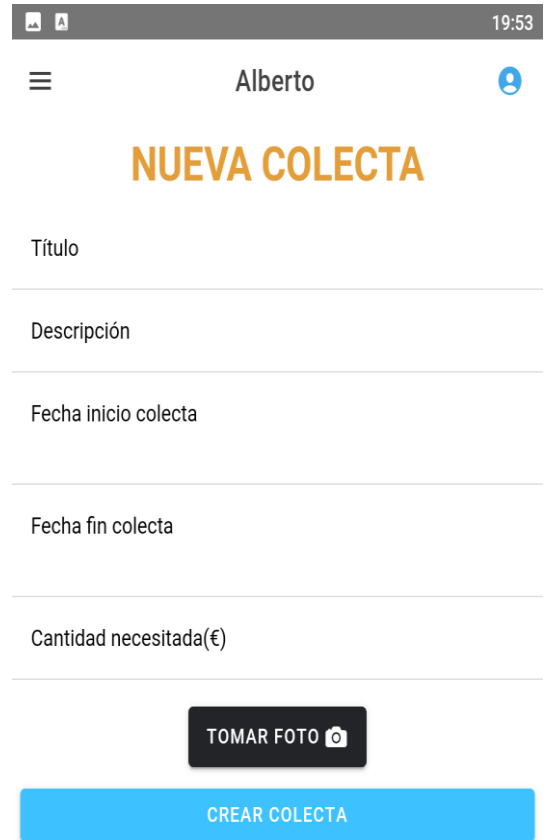


Ilustración 53 - Emulador Crear

5. Validación aplicación

Para validar el correcto funcionamiento de la aplicación se ha optado por la implementación de la automatización de test en la parte front.

La misión de estos tests es que cada vez que se añadan nuevas funcionalidades a la aplicación sigan funcionando las funcionalidades anteriores.

La principal idea de la validación es que las funcionalidades que han sido definidas como casos de uso sean aceptadas o validadas. Por tanto, el objetivo es intentar cubrir con test todas esas funcionalidades.

5.1 Software utilizado

Se ha utilizado Cypress para automatizar los tests. Cypress es una herramienta JavaScript para probar aplicaciones web en el navegador. Cypress toma el control de los distintos navegadores y realiza pruebas sobre la aplicación en cuestión.

¿Pero Ionic no es una aplicación para dispositivos móviles? ¿Qué tiene que ver con probar una web?

Como se comentó en [apartados anteriores](#), Ionic empaqueta como apk una aplicación web (JavaScript + HTML + CSS), por lo que, en realidad, estamos hablando de una aplicación web.

Ionic tiene una opción en desarrollo donde se despliega la aplicación en un navegador web y es ahí donde se aprovechará para realizar la validación de la aplicación, tal y como se explica en la documentación oficial de Cypress.

5.2 Definición pruebas validación

Se definen las pruebas de validación que deben pasar los casos de uso definidos por la aplicación. Recordemos que estas pruebas de validación se harán bajo las pruebas de Cypress.

Identificador	Descripción	Caso uso asociado
PV-001	Dar alta usuario	CU-001
PV-002	Añadir nueva colecta	CU-002
PV-003	Aportar a una colecta	CU-003
PV-004	Actualizar perfil	CU-004
PV-005	Listar colectas	CU-005
PV-006	Iniciar sesión	CU-006
PV-007	Cerrar sesión	CU-007
PV-008	Visualizar mis aportes	CU-008
PV-009	Visualizar mis colectas	CU-009

PV-010	Detalle colecta	CU-010
--------	-----------------	--------

Ilustración 54 - Definición pruebas validación

5.3 Resultado pruebas validación

Identificador	Descripción	Resultado
PV-001	Dar alta usuario	OK
PV-002	Añadir nueva colecta	OK
PV-003	Aportar a una colecta	OK
PV-004	Actualizar perfil	OK
PV-005	Listar colectas	Test no implementado
PV-006	Iniciar sesión	OK
PV-007	Cerrar sesión	Test no implementado
PV-008	Visualizar mis aportes	Test no implementado
PV-009	Visualizar mis colectas	Test no implementado
PV-010	Detalle colecta	CU-010

Ilustración 55 - Resultado pruebas validación

6. Conclusiones

Me he dado cuenta que a la hora de planificar es importante diferenciar entre especificar horas para tareas de las que ya tenemos conocimiento y tareas para lo cual no tenemos conocimiento. Si no se tiene conocimiento sobre alguna tecnología, seguramente estas tareas se multipliquen por dos en tiempo, ya que para aprenderlas necesitamos leer documentación sobre ella, realizar ejemplos y finalmente adaptarlo a lo que nosotros queremos.

Otra lección aprendida ha sido que la tarea de configurar toda la infraestructura del proyecto: servidores, contenedores, cuentas, etc., no es cosa baladí. Yo creía que se tardaría menos que en desarrollar la app y los tiempos han estado muy parejos.

Todos los objetivos no han sido logrados. Ha faltado empaquetar la aplicación para utilizarla en iOS y realizar más test que cubran todos los casos de uso. Los test ha sido una mala planificación en el desarrollo. Debería haberlos incluido desde el primer momento en que se empezó a desarrollar y no al revés.

Quiere decir que por cada funcionalidad que se vaya desarrollando hay que implementar los test y no dejarlo para el final. En cuanto al tema del empaquetado de la aplicación para iOS me he confiado y al final no ha habido tiempo para desarrollarlo. He leído la documentación y se tarda más de lo que esperaba.

El desarrollo en cascada se ha llevado a cabo de manera correcta casi todo el proyecto. Se han seguido los plazos fijados de una manera correcta, hasta el final del hito 3 donde había que entregar el desarrollo. Ahí hubo una pequeña desviación debido a falta de algunas funcionalidades y la total falta de tests. Esta desviación fue solventada incluyendo los desarrollos y test en el hito final, pero provocó un extra de esfuerzo y algunas funcionalidades sin desarrollar (empaquetado iOS). Quitando estos pequeños problemas volvería a utilizar esta metodología.

Para finalizar, en el futuro de la aplicación me gustaría acometer los siguientes puntos:

Ampliar los tests funcionales. Siempre que se implemente una nueva funcionalidad acompañarlo con un test.

Empaquetar la aplicación para iOS.

Implementar búsquedas de colectas por nombre.

7. Glosario

Término	Definición
Front end	Parte cliente de una aplicación web
Back end	Parte servidor de una aplicación web
Android	Sistema operativo de Google
iOS	Sistema operativo de Apple
Ionic	Framework que permite desarrollar aplicaciones nativas de iOS y Android como si fuese una aplicación web
Angular	Framework web
AWS	Servicio de la nube de Amazon
EC2	Máquina virtual disponible en AWS
Tomcat	Contenedor para aplicaciones Java. Soporta artefactos .war y .jar

8. Bibliografía

Diseño centrado en el usuario: PID_00176058 -
ESP_Diseño_Centrado_Usuario.pdf - Muriel Garreta Domingo y Enric Mor Pera

https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software
Octubre 2022

https://es.wikipedia.org/wiki/Desarrollo_en_cascada
Octubre 2022

Controlador/Servicio/Repositorio:

<https://tom-collings.medium.com/controller-service-repository-16e29a4684e5>

Noviembre - Diciembre 2022

Tomcat:

<https://ubunlog.com/en/tomcat-9-installation-ubuntu-18-04/>

<https://serverfault.com/questions/978731/how-do-i-fix-my-systemctl-status-tomcat>

<https://medium.com/@yashpatel007/how-to-connect-your-amazon-ec2-instance-with-a-domain-name-80ad8959078>

Diciembre 2022

Ionic

<https://ionicframework.com/docs/>

<https://ionicframework.com/docs/components>

<https://ionicframework.com/docs/angular/your-first-app/deploying-mobile>

Octubre 2022 – Enero 2023

MySQL

<https://medium.com/@centizennationwide/mysql-naming-conventions-e3a6f6219efe>

<https://linuxhint.com/connect-aws-rds-mysql-workbench/>

Diciembre 2022

RDS

<https://linuxhint.com/connect-aws-rds-mysql-workbench/>

Noviembre - Diciembre 2022

Spring

<https://spring.io/guides/gs/rest-service/>

<https://spring.io/guides/tutorials/spring-security-and-angular-is/>

<https://www.linkedin.com/pulse/spring-boot-best-practices-developers-omar-ismail>

Noviembre - Diciembre 2022

Cypress

<https://www.cypress.io/blog/2020/07/08/end-to-end-testing-mobile-apps-with-ionic-and-cypress/>

<https://docs.cypress.io/api/commands/last>

<https://docs.cypress.io/guides/references/configuration#Configuration-File>

9. Anexos

9.1 Transcripción de entrevista presencial

A continuación, se expone la transcripción de la entrevista presencial que se realizó a un profesor en el Colegio Público Lepanto, ubicado en Madrid.

Nos referiremos con A, al entrevistador (Alberto) y con P, al entrevistado, el profesor.

A: Buenos días, ¿qué tal? ¿cómo se llama?

P: Mi nombre es Pablo.

A: ¿Me podría decir su edad?

P: No hace falta que me tutees. ¡Ja, ja, ja!

A: Perfecto, pues, ¿qué edad tienes?

P: Tengo 33 años.

A: ¿Cuál es tu sexo?

P: Como puedes observar, masculino.

A: ¿A qué te dedicas?

P: Soy profesor en el colegio público Lepanto, que está en Madrid.

A: ¿Podrías decirme qué haces exactamente dentro de la escuela?

P: Me dedico a enseñar la asignatura “Conocimiento del medio natural”.

A: ¿Cuántos alumnos tienes en el aula?

P: La verdad es que yo tengo 26. Otros compañeros pueden llegar a tener 28 o 29.

A: Vaya pasada, entiendo que a mayor número de alumnos el nivel de atención e ilusión de los alumnos disminuirá.

P: Sí, es correcto.

A: ¿Has notado en este arranque del curso (22-23), falta de material o recursos en el aula o instalaciones del colegio?

P: Si te digo la verdad es lo de todos los años, falta bastante material. Se nota sobre todo en las aulas de los alumnos, falta de proyectores, pupitres desgastados, etc.

A: Entiendo. Dejando de lado la falta de recursos del colegio, ¿ha notado que los alumnos tengan dificultades para presentar los materiales adecuados?

P: Sí, desde luego, este año con el tema de la inflación los materiales escolares para los alumnos se han puesto por las nubes y yo, personalmente, estoy viendo muchas carencias en los alumnos. Falta de libros, cuadernos, no poder realizar ciertas actividades, etc.

A: Es una realidad bien triste la que me estás contando. ¿Conoces o has oído hablar del crowdfunding o micromecenazgo?

P: Sí, sí, esto del crowdfunding está muy de moda los últimos años. Es esto de que publicas cierto proyecto que te gustaría realizar pero no tienes fondos para acometerlo. Entonces lo publicas en internet con la cantidad que necesitas y

personas que no conoces de nada pueden aportar fondos a este proyecto. Micromecenazgo supongo que será lo mismo, será la palabra en español.

A: Sí, es justo lo que has comentado. ¿Qué te parecería aplicar esta idea del micromecenazgo a los problemas que me has comentado de la escuela? ¿Realizar una app? Cierta profesor, de cierta escuela, publica un crowdfunding y una persona de forma altruista realiza una aportación.

P: Me parece una idea muy buena. De esta manera podríamos obtener bastantes materiales que, de verdad, son muy necesarios para una educación de calidad. Aunque creo que no es la mejor vía, ya que estas cosas las tendría que solucionar la administración pública.

A: ¿Crees que tendría buena aceptación por parte de los padres?

P: Sí, claro que sí, se trata de mejorar la educación de sus hijos, por lo tanto, no tengo ninguna duda que esta app les encantará.

A: Dime cuál sería la primera colecta que publicarías en la app.

P: Pues te vas a reír, pero la primera colecta que publicaría sería obtener un esqueleto para el aula. A estas edades, seguro que se les queda más en la cabeza cada hueso del cuerpo humano viéndolo, tocándolo, etc., que recitándolo de memoria, jajaja.

A: Y por último, ¿cómo cree que puede influir esta app de mecenazgo en el rendimiento de sus alumnos?

P: Sin duda de manera positiva, creo que se centrarían más en lo que realmente importa, dejando de lado las dificultades que muchas de sus familias están pasando. Creo también que obteniendo ciertos materiales para los alumnos es fundamental para dar rienda suelta a su creatividad.

A: Perfecto, pues hasta aquí hemos llegado. ¡¡Muchas gracias por todo!! Nos vemos.

P: Gracias a ti. Ha sido un auténtico placer. Avisadme cuando tengáis la app en marcha.

9.2 Preguntas de las encuestas realizadas.

Nos referiremos con E, a la pregunta de la encuesta.

E: Nombre

E: Edad
E: Sexo
E: Profesión
E: Nivel de estudios: *Primarios / Secundarios / Bachillerato / Universitario*
Preguntas de las encuestas realizadas.
E: Localidad
E: Nivel uso tecnologías: *Alto / Medio / Bajo*
E: Sistema operativo teléfono: *Android / iOS*
E: ¿Sabe usted lo que es el crowdfunding? *Sí/No*
E: ¿Tiene aplicaciones instaladas de crowdfunding en el teléfono? *Sí/No*
E: ¿Ha realizado alguna colecta para un crowdfunding mediante una app? *Sí/No*
E: ¿Ha realizado alguna aportación para un crowdfunding mediante una app? *Sí/No*
E: ¿Le parece importante para el rendimiento de los alumnos llevar unos materiales escolares en buenas condiciones? *Sí/No*
E: ¿Le parece importante para el rendimiento de los alumnos que los materiales del aula estén en buenas condiciones? *Sí/No*
E: ¿Qué le parecería disponer de una app para financiar de manera altruista el material escolar para un determinado aula? *Buena idea / Esto le corresponde al estado / Esto le corresponde al estado, pero en su ausencia es buena idea /*
E: ¿Realizaría una donación por medio de la app descrita anteriormente? *Sí / No / Quizás*
E: ¿Publicaría una colecta por medio de la app descrita anteriormente? *Sí / No / Quizás*