UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

HurbValencia :Desarrollo de una aplicación para la gestión de huertos urbanos.

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Sanz Arilla, Daniel Eduardo

Tutor/a: Penadés Gramage, María Carmen

CURSO ACADÉMICO: 2023/2024

AGRADECIMIENTOS

Quiero agradecer, en primer lugar, a Luis Navarro, Gabriel Rodríguez Díaz y Laijie Ji que hicieran posible, con su grandioso trabajo en equipo, lo imposible y llevar este proyecto más lejos de lo que hubiera imaginado cualquiera de nosotros.

Además, a lo largo de todo este camino recorrido estos últimos años, he de mostrar mi gratitud a mi pareja, Maya, que ha estado aguantándome día tras día, tanto los buenos como los malos, y a mis padres y hermanos por haber confiado en mí desde siempre.

Por último, a mi tutora del TFG, María Carmen Penadés Gramage, que, sin conocerme de prácticamente nada, me ha ayudado muchísimo y ha hecho posible la entrega de este trabajo.

RESUMEN

Hoy en día muchas personas se están dando cuenta de lo entretenido y satisfactorio que puede ser tener tu propio huerto urbano, y es por ello, que nace **HurbValencia**, con el objetivo de facilitar a los ciudadanos distintos lugares en los que poder cultivar, cómo hacerlo y una gestión de las parcelas para la comunidad, de forma que sean mucho más simples de repartir. Además, también se integran métodos para hacer que todo sea más interactivo y que puedan intercambiar cultivos con sus vecinos de huerto entre otras.

HurbValencia es un proyecto que surge como idea de la competición llamada Codethon, realizada en la Escuela Técnica Superior de Ingeniería Informática (ETSINF) durante el curso 2023/2024 donde como objetivo se proponía crear una aplicación que hiciese la ciudad de valencia una *Smart City*. **HurbValencia** ha sido realizado en equipo por cuatro alumnos, donde cada uno de nosotros tenía que realizar distintas tareas. Mi participación se ha centrado especialmente en el diseño y desarrollo de la interfaz de todo el proyecto y de algunas de sus funcionalidades, como por ejemplo toda la parte de prioridad en la página de la agenda y el iniciar sesión y registrarse. El equipo estaba formado por: Luis Navarro, Gabriel Rodríguez Díaz, Laijie Ji y Daniel Eduardo Sanz Arilla.

En este TFG desarrollaremos detalladamente cómo nos organizamos para lograr cumplir el reto de acabar el producto viable mínimo (MVP), un análisis de nuestros competidores para ver en que podíamos destacar, que tecnologías usamos y, sobre todo el trabajo realizado por el autor de esta memoria, sobre cómo funciona nuestra aplicación con sus *mockups* iniciales y resultado final.

Palabras clave: *Smart City*, Sostenibilidad, Huertos Urbanos, Aplicaciones móviles, Flutter, Dart, Supabase

ABSTRACT

Nowadays many people are realizing how entertaining and satisfying it can be to have your own urban garden, and that is why **HurbValencia** was born, with the aim of providing citizens with different places in which to grow, how to do it. and management of the plots for the community, so that they are much simpler to distribute. In addition, methods are also integrated to make everything more interactive and allow them to exchange crops with their garden neighbors, among others.

HurbValencia is a project that emerged as an idea from the competition called Codethon, held at the Higher Technical School of Computer Engineering (ETSINF) during the 2023/2024 academic year, where the objective was to create an application that would make the city of Valencia a Smart City. **HurbValencia** has been carried out as a team by four students, where each of us had to perform different tasks. My participation has focused especially on the design and development of the interface of the entire project and some of its functionalities, such as the entire priority part on the agenda page and the login and registration. The team was made up of: Luis Navarro, Gabriel Rodríguez Díaz, Laijie Ji and Daniel Eduardo Sanz Arilla.

In this TFG we will develop in detail how we organize ourselves to meet the challenge of finishing the minimum viable product (MVP), an analysis of our competitors to see where we could stand out, what technologies we use and, above all, the work carried out by the author of this memory, about how our application works with its initial mockups and final result.

Keywords: Smart City, Sustainability, Urban Gardens, Mobile applications, Flutter, Dart, Supabase

RESUM

Hui dia moltes persones s'estan donant compte de l'entretingut i satisfactori que pot ser tindre el teu propi hort urbà, i és per això, que naix **HurbValencia**, amb l'objectiu de facilitar als ciutadans diferents llocs en els quals poder cultivar, com fer-ho i una gestió de les parcel·les per a la comunitat, de manera que siguen molt més simples de repartir. A més, també s'integren mètodes per a fer que tot siga més interactiu i que puguen intercanviar cultius amb els seus veïns d'hort entre altres.

HurbValencia és un projecte que sorgix com a idea de la competició anomenada Codethon, realitzada a l'Escola Tècnica Superior d'Enginyeria Informàtica (ETSINF) durant el curs 2023/2024 on com a objectiu es proposava crear una aplicació que fera la ciutat de València una Smart City. **HurbValencia** ha sigut realitzat en equip per quatre alumnes, on cadascun de nosaltres havia de fer diferents tasques. La meua participació s'ha centrat especialment en el disseny i desenvolupament de la interfície de tot el projecte i d'algunes de les seues funcionalitats, com per exemple tota la part de prioritat en la pàgina de l'agenda i l'iniciar sessió i registrar-se. L'equip estava format per: Luis Navarro, Gabriel Rodríguez Díaz, Laijie Ji i Daniel Eduardo Sanz Arilla.

En este TFG desenvoluparem detalladament com ens organitzem per a aconseguir complir el repte d'acabar el producte viable mínim (MVP), una anàlisi dels nostres competidors per a veure en què podíem destacar, que tecnologies usem i, sobretot el treball realitzat per l'autor d'esta memòria, sobre com funciona la nostra aplicació amb les seues mockups inicials i resultat final.

Paraules clau: *Smart City*, Sostenibilitat, Huerts Urbans, Aplicacions mòbils, Flutter, Dart, Supabase

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE ILUSTRACIONES

1. Introducción

Este capítulo se describen los objetivos fundamentales de **HurbValencia**, explorando los impulsos que dieron origen al desarrollo de esta aplicación, así como los objetivos propuestos y la metodología empleada para su realización. Además, se resume la estructura de la memoria, comentando los distintos capítulos que la componen.

1.1 Motivación

En el contexto de los desafíos tecnológicos que enfrenta la ciudad de Valencia, surge Artemis Club [1], con el propósito de fomentar la creatividad y el aprendizaje entre los estudiantes mediante el desarrollo de aplicaciones tanto web como móviles, dando lugar a una competición(*Codethon*) entre estudiantes a los que les gusta la programación, independientemente de sus áreas de estudio (informática, diseño, audiovisuales, entre otros, con un reto: ¿Cómo transformar Valencia en una *Smart City*?

Pensando en cómo afrontar este reto nació **HurbValencia**, una app para móviles que quiere que la gente descubra lo interesante, reconfortante y saludable que puede ser tener tu propio huerto urbano.

Motivados por el compromiso con el bienestar colectivo y la sostenibilidad y mejorar nuestro propio aprendizaje, creamos un equipo formado por cuatro estudiantes de la ETSINF (Luis Navarro, Gabriel Rodríguez Díaz, Laijie Ji y Daniel Eduardo Sanz Arilla), que nos propusimos durante tres meses (21/03/2024 al 15/05/2024) crear una herramienta que lograse aquel reto, aspirando a fomentar una comunidad más consciente y comprometida con el cultivo urbano mediante el uso de tecnologías innovadoras y una interfaz amigable y sencilla de usar. A partir del trabajo realizado por el autor de esta memoria, y completando el proyecto, se ha realizado este TFG.

1.2 Objetivos

El objetivo principal del TFG es desarrollar una aplicación que haga posible que las personas descubran una nueva y saludable afición, cultivar su propio huerto, mediante el uso de funcionalidades innovadoras, como podría ser un mercado en el que puedas intercambiar cultivos con los integrantes de la comunidad de tu mismo huerto urbano y

una interfaz que cumpla con las heurísticas de usabilidad de Nielsen. Este TFG divide los objetivos en 2 partes: objetivos del proyecto y objetivos personales.

Objetivos proyecto:

- Determinar la idea de negocio para extraer las funcionalidades principales del proyecto
- Construir un Producto Mínimo Viable (MVP) que contribuya a la mejora tecnológica de la ciudad de Valencia.
- Diseñar interfaces de usuario que sean intuitiva, ergonómicas, cómodas y fáciles de usar.
- Aplicar metodologías ágiles para aumentar la probabilidad de éxito en la realización del MVP.

Objetivos personales:

- Mejorar mi capacidad de comunicación y escritura.
- Aprender tecnologías actuales que no se hayan usado en el grado de Ingeniería Informática, para así marcar mi inicio al camino laboral.
- Fomentar un ambiente de trabajo positivo en el equipo, promoviendo la colaboración y el apoyo mutuo para lograr un excelente trabajo en equipo.

1.3 Metodología de desarrollo

Para la realización del proyecto hay que seguir una planificación con una metodología flexible y adecuada al desarrollo de un MVP en un breve periodo de tiempo.

Por lo tanto, nos hemos decantado por seguir una metodología ágil [2], que destaca sobre todo por la flexibilidad, la mejora constante y fomentar la colaboración entre los distintos miembros del equipo. En concreto hemos empleado la Scrum [3].

Algunas de sus características más importantes son:

• Dividir el tiempo del proyecto en distintos *sprints*, que cada uno contiene distintas tareas a realizar y estas están ordenadas por importancia, de más a menos, de forma que, si hay alguna tarea que se queda sin realizar, ésta sea la menos relevante en el resultado final.

• El equipo estará formado por pocos miembros e informado diariamente sobre las tareas que esté realizando cada uno de sus componentes. Así, se harán las adaptaciones necesarias en caso de que alguno de ellos no pueda acabar la correspondiente tarea.

• En general, el rol de *Scrum Master* se caracteriza por ser responsable de facilitar la creación de tareas y asegurar que todos los miembros del equipo estén conformes con ellas y las ejecuten. En nuestro proyecto, aunque no se designó explícitamente a una persona para este rol, asumí la responsabilidad de coordinar las reuniones y guiar al equipo.

Por último, en el capítulo 3 de este documento, explicaré de forma más detallada como hemos seguido esta metodología.

1.4 Estructura de la memoria:

El documento está organizado en siete capítulos, que describen detalladamente el proceso llevado a cabo durante este proyecto para acabar teniendo un MVP funcional, cumpliendo la mayoría de nuestros objetivos. La estructura se presenta de la siguiente manera:

• Capítulo1. Introducción y objetivos:

Expone el concepto fundamental que dio origen al proyecto y su visión y valores. Este apartado ya ha sido desarrollado más arriba.

• Capítulo 2. Evaluación de idea de negocio:

En este capítulo se realiza un análisis de nuestros competidores para ver cómo superarles y qué podemos adaptar de ellos a nuestra aplicación. También haremos uso de algunas herramientas muy útiles, como podría ser la matriz DAFO, para evaluar la situación de nuestra aplicación en el mercado, y el uso de Lean Canvas para estudiar de manera clara y visual el modelo de negocio.

• Capítulo 3. **HurbValencia**, Cronología del proyecto:

Aquí comentaremos, paso a paso, como dividimos el tiempo para poder realizar con éxito nuestro MVP y explicaremos cada paso del *backlog* del proyecto, comentando los *mockups* iniciales que nos servirían de guía durante todo el proyecto. Además, durante este proceso veremos la importancia que tuvo la metodología escogida (metodología *agile*).

• Capítulo 4. Desarrollo por *sprints*:

Como hemos mencionado anteriormente, en este apartado detallaremos que tareas del *backlog* realizamos en cada *sprint* y detallaremos cada unidad de trabajo, a su vez que las pruebas de aceptación realizadas para cada una de ellas.

- Capítulo 5. Detalles de la implementación:

En esta sección hablaremos de las distintas tecnologías usadas en nuestra aplicación, como puede ser, de cómo logramos una interfaz simple y agradable de forma satisfactoria y de la estructura seguida en el proyecto para que en vez de obtener lo que se llama un código "espagueti" resultase en un código "limpio". Por último, se hará una breve mención y explicación al diagrama de clases usado.

- Capítulo 6. Pruebas de Validación:

En el penúltimo capítulo se describirán detalladamente las pruebas de validación llevadas a cabo para asegurar el éxito del proyecto. Se incluirán dos casos de uso específicos en los que un usuario interactúa con nuestra aplicación. El objetivo es observar si el usuario puede utilizar la aplicación de manera exitosa o si experimenta algún tipo de problema durante el proceso. Estas pruebas son fundamentales para garantizar que nuestra aplicación cumple con los requisitos y expectativas establecidos.

- Capítulo 7. Conclusión y trabajo futuro:

Por último, expondremos el resumen del resultado final de este trabajo de fin de grado, de la importancia de algunas de las asignaturas de la carrera de Ingeniería Informática que han permitido desarrollarlo y comentaremos varios aspectos que nos gustaría cambiar para aportarlas en nuestra aplicación final.

2. Evaluación idea de negocio

En este capítulo se presentan las distintas funcionalidades y virtudes que harán que **HurbValencia** destaque como idea de negocio y, para ello, se realizará un estudio completo del mercado. Haremos uso de la herramienta Lean Canvas con el fin de ver los distintos problemas que hay en el entorno, las soluciones que podemos aportar y qué ventajas competitivas tiene nuestra aplicación respecto al resto del mercado. Finalmente, se presenta un análisis exhaustivo del negocio mediante una matriz DAFO.

2.1 **HurbValencia** como idea de negocio

En la actualidad, son pocos quienes reconocen lo entretenido y beneficioso que puede ser tener un huerto propio. Sin embargo, la necesidad de disponer de la información necesaria para lograr obtener un huerto funcional junto con otras de sus posibles limitaciones, como son el escaso espacio disponible en tu hogar y el desconocimiento de la existencia de los huertos urbanos y sus potenciales aportaciones, lleva a que la mayoría de la población desista de emprender la creación de tu propio huerto

Por esta razón, el proyecto **HurbValencia** tiene tres objetivos esenciales. En primer lugar, busca dar a conocer los huertos urbanos más cercanos a la ubicación del usuario, proporcionando un entorno más cómodo que cultivar en casa. Además, se pretende facilitar toda la información necesaria a los usuarios sobre cómo cultivar sus propias verduras, hortalizas, frutales y plantas ornamentales, cuáles son las mejores épocas del año para hacerlo y otros aspectos relevantes. Y, por último, se le proporcionarán herramientas para facilitar la gestión de las comunidades de huertos urbanos y así conocer de forma más intuitiva información sobre el propietario de cada parcela. Finalmente, se realizará un proyecto de mercado que permita que los miembros de una misma comunidad puedan vender o intercambiar sus productos, haciendo el proceso de cultivo más entretenido y dinámico.

La idea de negocio es, de hecho, el inicio de todo proyecto empresarial. Por ello, disponer de unas bases sólidas puede ser determinante para su éxito. Tras revisar y evaluar las diferentes propuestas que estudiamos, consensuamos que el proyecto con mayor potencial futuro era crear **HurbValencia**. Nos dimos cuenta de que **HurbValencia** era una idea mucho más importante de lo que parecía inicialmente. Así, entendimos que un análisis

exhaustivo que diese lugar a una solución eficiente, podría resolver el principal problema del entorno de los huertos urbanos y ser clave para convertir nuestra idea en un proyecto de éxito.

Es por esto por lo que, en los siguientes apartados, realizaremos aquel análisis para identificar, finalmente, la mejor solución a las barreras existentes que lo dificultan.

2.2 Lean Canvas

Para encontrar la solución mencionada anteriormente, lo primero que haremos es emplear Lean Canvas [4] (ver Ilustración 1) Esta es una herramienta que permite estudiar de manera clara y visual el modelo de negocio, con el fin de aumentar la probabilidad de obtener beneficios empresariales y para la sociedad en general.

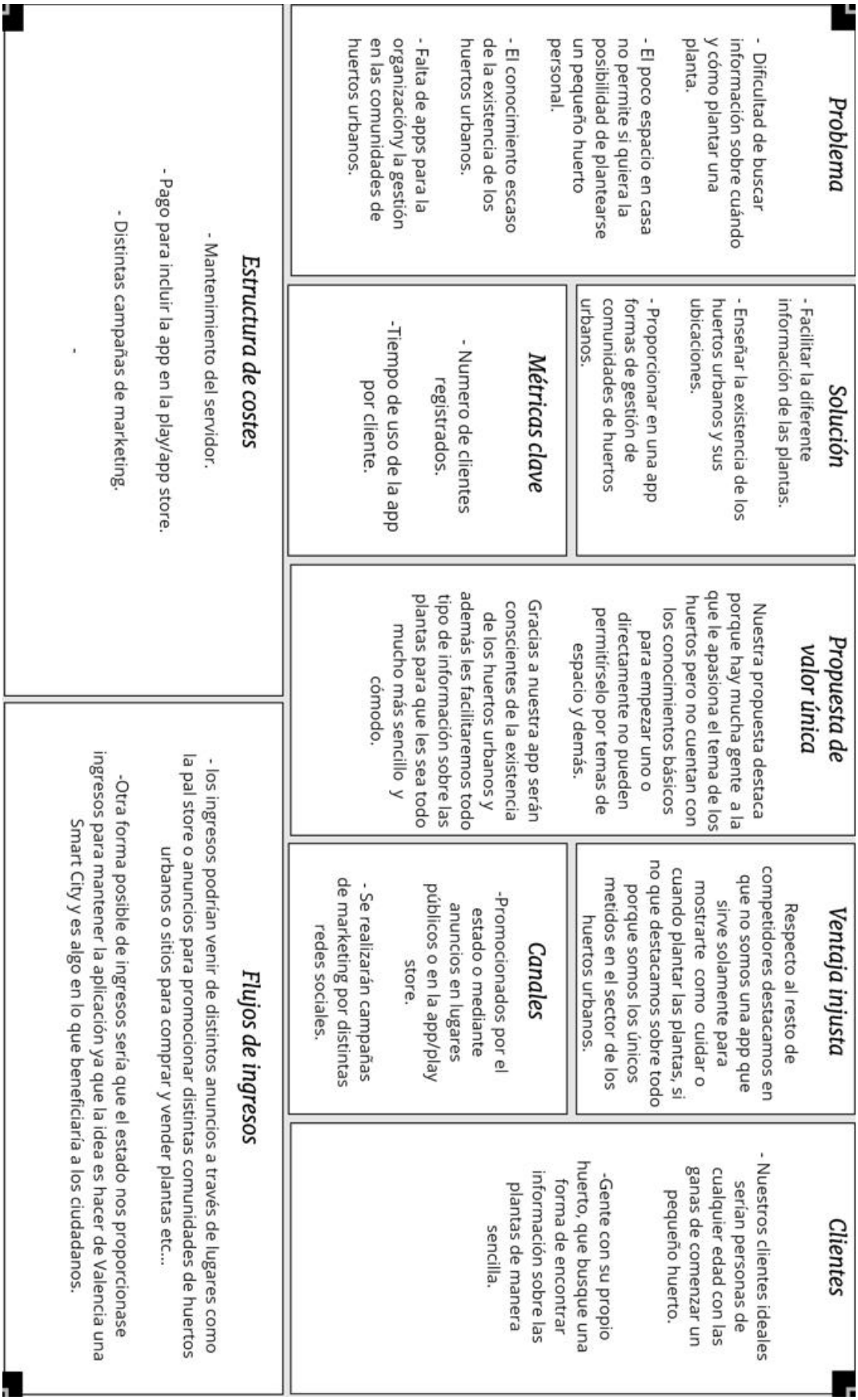


ILUSTRACIÓN 1: LEAN CANVAS

2.3 Estudio del mercado

Otra de las tareas más importantes a la hora de iniciar un proyecto innovador es llevar a cabo un profundo estudio de mercado, para detectar las soluciones que ofrecen actualmente nuestros potenciales competidores para, por un lado, poder distinguirnos de ellas, adaptarnos al entorno y superarlas con ideas y herramientas innovadoras. Los siguientes apartados explican de forma precisa dicho análisis de mercado.

2.3.1 Maceto Huerto

Esta app [5], desarrollada por Josep L. Centelles, sirve para obtener información general de cómo cuidar varias plantas y cultivos, como podría ser la cantidad de riego abonado, entre otras (ver Ilustración 2).

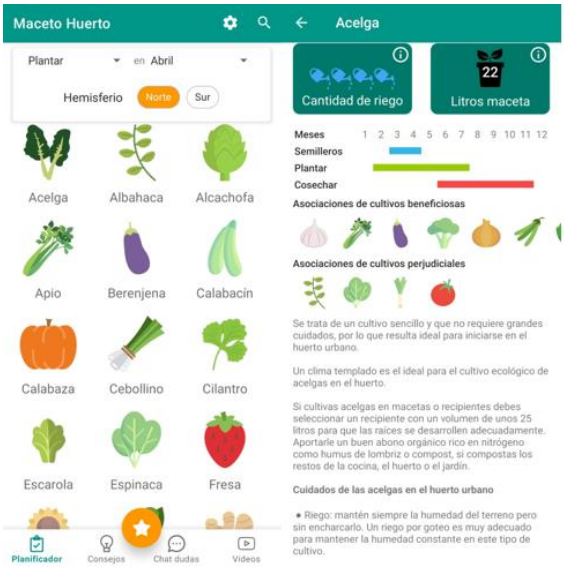


ILUSTRACIÓN 2: PLANTAS E INFORMACIÓN MACETO HUERTO.

Otra de las características más relevantes de esta aplicación es su sección de consejos. Esta es extremadamente útil, sobre todo, para quienes se estén iniciando en esta afición que aporta sostenibilidad a nuestro entorno. Aquí se les proporciona distinta información de cómo enfrentarse a algunas plagas o cómo empezar a construir su huerto, entre otras.

Por último, pero no menos importante, la aplicación cuenta, también, con una sección de dudas, preguntas y respuestas que puede resultar de gran ayuda para quienes tengan

distintos problemas con su huerto o que busquen el consejo de otros ciudadanos participantes en esta iniciativa transversal (ver Ilustración 3).

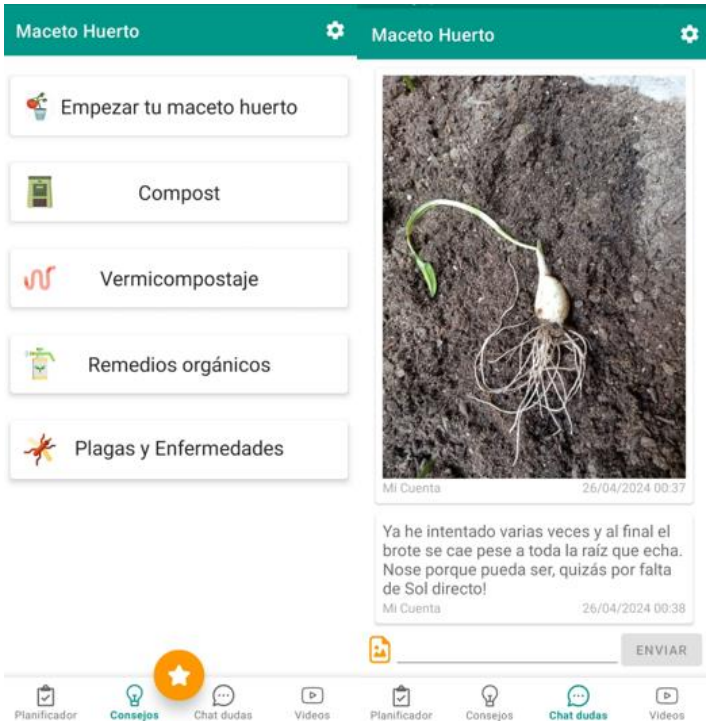


ILUSTRACIÓN 3: CONSEJOS Y DUDAS MACETO HUERTO.

2.3.2 Planificador de jardines

Esta aplicación [6], desarrollada por Bento Software, se parece a la anterior en que muestra información relevante sobre los diferentes tipos de cultivo. Sin embargo, una característica diferenciadora es que permite añadir las plantas de interés a una lista donde puedes visualizarlas fácilmente (ver Ilustración 4).

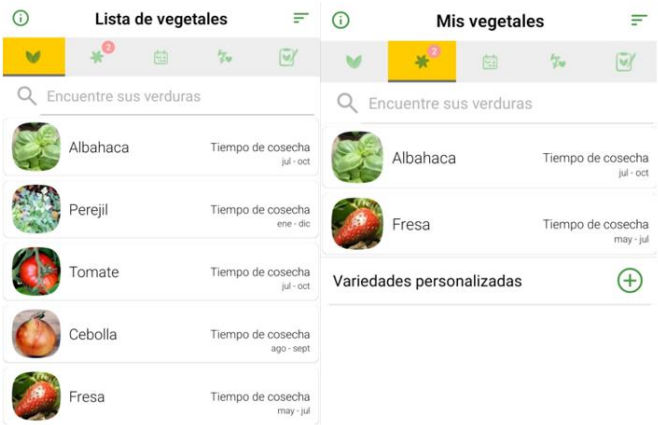


ILUSTRACIÓN 4: PLANIFICADOR DE JARDINES, LISTA DE PLANTAS.

2.3.3 Planter -Garden Planner

Desarrollada por Planter, esta es una de las apps más descargadas de su categoría [7]. Destaca por tener casi todas las características de las anteriores aplicaciones, como puede ser la información sobre las plantas y su listado. Sin embargo, lo que la distingue es su excepcional función de gestión de cultivos en parcelas, es decir, puedes simular una parcela del tamaño deseado y organizar plantas en cuadrículas, facilitando así la organización y el seguimiento del huerto (ver Ilustración 5).

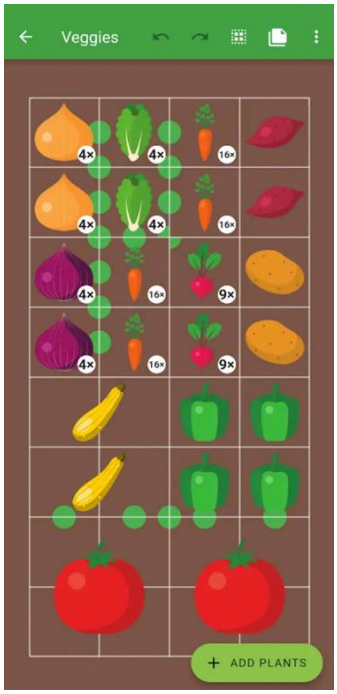


ILUSTRACIÓN 5: GESTIÓN DE CULTIVOS DE PLANTER ORGANIZER

2.3.4 Picture This

Picture This [8], desarrollada por Glority Global Group Ltd, se encuentra entre las seis aplicaciones más populares de la categoría de educación en la Play Store. A diferencia de las otras aplicaciones ya comentadas, *Picture This* ofrece una funcionalidad innovadora y única: el reconocimiento mediante inteligencia artificial de diversas plantas.

Al tomar una foto de una planta, esta aplicación te ofrece información detallada sobre ella, incluyendo una descripción y consejos muy útiles como la frecuencia de riego, poda y fertilización. Además, te alerta sobre posibles problemas como infecciones por hongos, falta de agua o si la planta es venenosa. Finalmente, te ofrece opciones de tratamiento de las enfermedades o plagas que detecta (ver Ilustración 6).

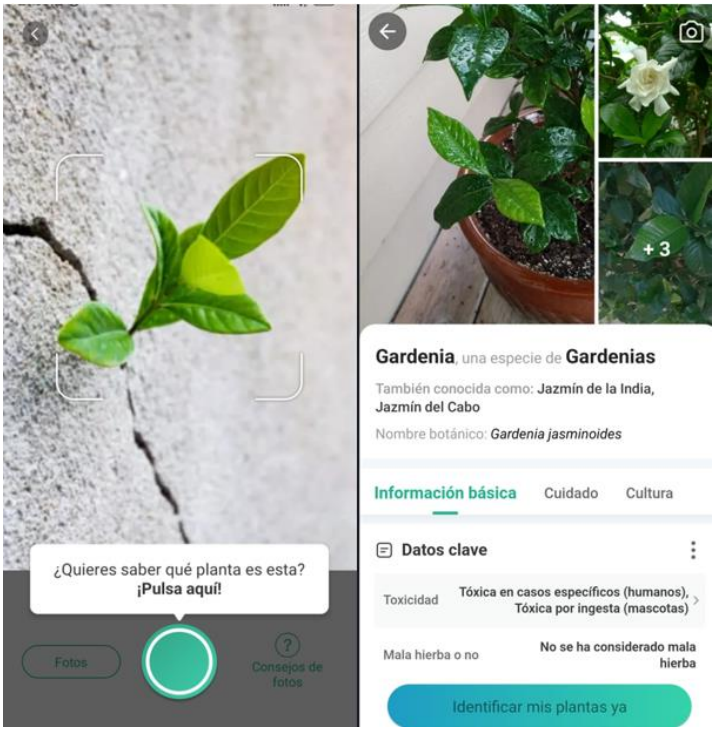


Ilustración 6: Ejemplo de la IA de Picture This

Una vez realizado el análisis de mercado, la Tabla 1 muestra las características vistas, y si estas se incorporan o no a **HurbValencia**.

TABLA 1:COMPARACIÓN DE LAS CARACTERÍSTICAS DE HURBVALENCIA CON LA COMPETENCIA.

Carácterísticas de la competencia:	Carácterísticas HurbValencia:
Información detallada sobre los cultivos.	Realizado
Matriz con los cultivos plantados.	Realizado
Inteligencia artificial para identificar plantas.	No realizado
Información de ayuda para el control de plagas y selección de cultivos.	Realizado
No realizado	Mercado para intercambiar cultivos
No realizado	Agenda
No realizado	Mapa para mostrar huertos urbanos

2.4 Análisis DAFO

Tras haber realizado el análisis de mercado, ahora vamos a proceder con su estudio empleando la matriz DAFO [9,10]. El análisis DAFO es una técnica fundamental para evaluar la situación de una empresa, institución, proyecto o persona con el propósito de tomar las decisiones estratégicas adecuadas. Este análisis permite examinar tanto las características internas (Fortalezas y Debilidades) como las externas (Oportunidades y Amenazas) de manera estructurada, a través de una matriz cuadrada.

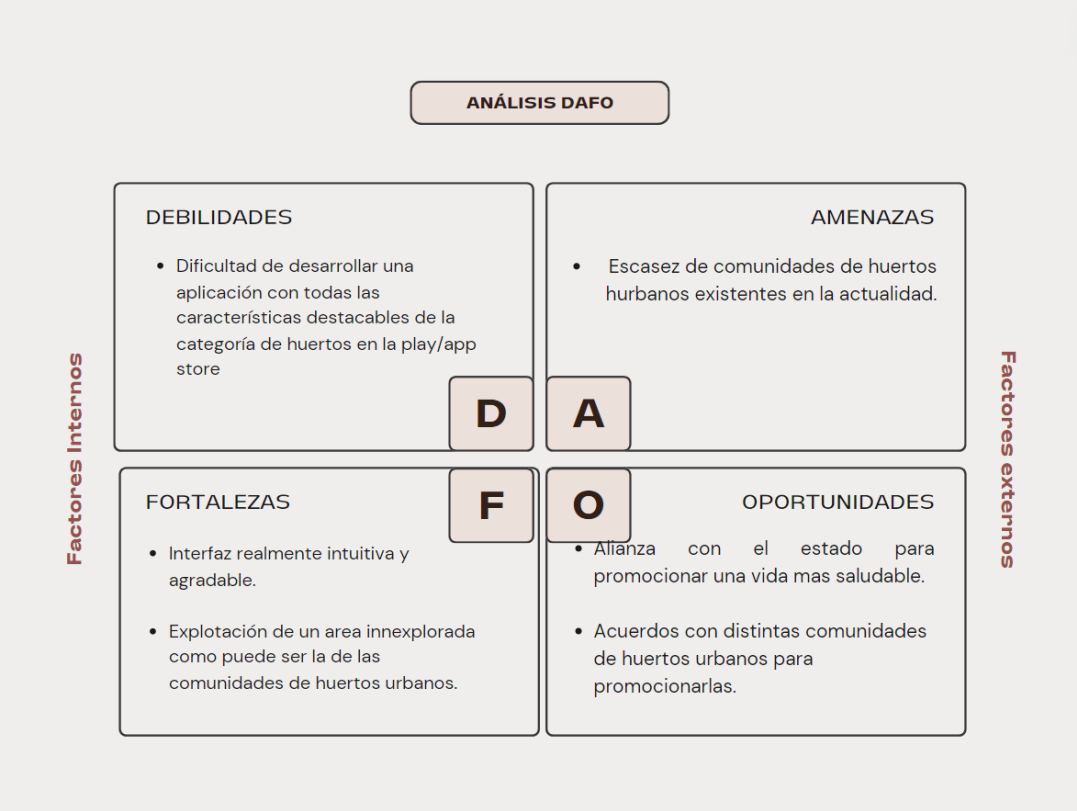


ILUSTRACIÓN 7:ANÁLISIS DAFO

- Detallando un poco más de lo que vemos en la Ilustración 7 y empezando por las debilidades, uno de nuestros mayores problemas, como es lógico, es la dificultad de desarrollar una app que contenga todo lo necesario para superar a las ya existentes en el mercado. Además, tendremos el reto de incluir novedades, como la integración de comunidades de huertos urbanos, de las que desconocemos su grado de aceptación por los usuarios.
- En cuanto a las amenazas, la principal que tendríamos que enfrentar sería la posibilidad de que **HurbValencia** tenga un éxito abrumador y que las plazas disponibles en las distintas comunidades de huertos urbanos se agotasen.

Consideramos que esto significa que habría ido muy bien, y se podrían incluso construir más comunidades, además que habríamos logrado el reto propuesto en la *codethon* de contribuir a la imagen de Valencia de una ciudad más "*smart*" y más "verde" en el entorno europeo.

- Continuando con las fortalezas, contamos con tener una interfaz amigable e intuitiva. Nuestra mayor virtud será que estamos adentrándonos en un área totalmente inexplorada, ya que en nuestra investigación de mercado hemos visto que no hay prácticamente casi ninguna app que se centre en la gestión de las comunidades de huertos urbanos. De hecho, todas están diseñadas para la gestión de tu propio huerto o identificación de plantas.

- Finalmente, al considerar las oportunidades, dado que nuestra aplicación promueve un estilo de vida sostenible y fomenta una afición saludable, existe la posibilidad de que la administración pública nos promocione o nos recompense de alguna manera. Además, podríamos llegar a algunos acuerdos con diversas comunidades de huertos urbanos o tiendas de plantas para que las promocionásemos. Finalmente, creemos que nuestra herramienta satisface de lleno los objetivos del programa europeo Horizonte 2030 y contribuye de forma relevante a uno de los pilares del programa *Horizont Europe* de la Comisión Europea. Por todo ello, y a través de la Universidad Politécnica de Valencia, tendríamos la posibilidad de optar a convocatorias de concurrencia competitiva, de alto impacto económico, en alianza con otras instituciones de reconocido prestigio, nacionales o europeas, públicas y privadas, alineadas con nuestro proyecto.

2.5 Modelo de negocio

Una vez habiendo hecho un exhaustivo análisis de la matriz DAFO y de la técnica Lean Canvas, queda por realizar una de las partes más importantes de este capítulo y del futuro funcionamiento de nuestro proyecto y que éste pueda ser rentable para una empresa. Esto es el modelo de negocio [11], que se emplea para definir cómo se va a vender un producto o servicio.

De entre los diferentes tipos de modelos disponibles, hemos descartado la opción de pago por uso. Dado que nuestro objetivo es que las personas descubran los huertos urbanos

como una nueva afición, es poco viable hacer que paguen por descargar la aplicación. Por lo tanto, hemos decidido que, al principio, nuestras ganancias provendrán de anuncios poco intrusivos colocados por la Play/App Store en nuestra aplicación. Una vez que tengamos más éxito, planeamos realizar promociones de diversas comunidades de huertos urbanos, viveros y tiendas de plantas a cambio de ingresos.

Por último, estaría la posibilidad de recibir algún tipo de ayuda económica por parte del Estado/Administración Pública para promocionarnos o cubrir los costes de los servidores. Como esto último es altamente improbable, podríamos optar a conseguir fondos en convocatorias públicas en concurrencia competitiva, ya sean europeas, nacionales o regionales, al estar este proyecto alineado con los objetivos marcados por la Unión Europea en su agenda 2030. La marca UPV, que facilita enormemente incorporar socios al proyecto, públicos y privados, es una clara ventaja para tener éxito en estas llamadas.

3. **HurbValencia**: Cronología del proyecto

Dos de los aspectos fundamentales de un proyecto son su planificación y la metodología empleada para llevarlo a cabo. En este capítulo hablaremos brevemente del *backlog* del proyecto donde se detallarán todas las tareas a hacer. Posteriormente se ofrecerá una breve explicación del desarrollo por *sprints* característico de la metodología *Scrum*, ya mencionada anteriormente y que se detallará en el capítulo 4. Finalmente, se abordará un tema de suma importancia: la explicación de la arquitectura utilizada en el proyecto, incluyendo las razones que justifican su elección.

3.1 *Backlog* del proyecto

Como parte de la metodología *Scrum*, hemos de elaborar el *backlog* de proyecto, es decir, las funcionalidades que queremos que tenga la app y los requisitos que debe cumplir. Algo a tener en cuenta es que el *backlog* sea muy flexible, es decir, que si se detectan nuevas funcionalidades se pueden añadir sin problema, además de que el objetivo es desarrollar las tareas por prioridad. De esta forma, si por falta de tiempo no podemos realizar todas las tareas del *backlog*, nos aseguramos de que las más importantes estén hechas.

TABLA 2: *BACKLOG* DEL PROYECTO AGRUPADO POR CARACTERÍSTICAS

General	*Login*
	Register
	Crear Comunidad
	Unirse Comunidad
	Unirse como invitado
	Modo Oscuro
	Editar perfil
	Unirse a varias comunidades
	Cerrar sesión
	QR para compartir comunidad
Inteligencia Artificial	Reconocimiento de plantas por foto.
	Información de la planta capturada.
	Información de ayuda para el cultivo.
API Clima	Ver clima de los próximos días.

	Notificaciones sobre la lluvia y su cantidad.
Agenda/ToDo	Crear Tareas
	Editar Tareas
	Tab para las tareas por hacer y para las hechas.
	Enviar tareas hechas a su *tab* correspondiente mediante una *checkbox*.
	Eliminar Tareas
	Ordenar tareas por prioridad
MarketPlace	Crear publicación
	Avatar con nombre de la persona creadora del post
	Al clicar sobre el chat del post de una persona que te lleve al chat con dicha persona
	Eliminar publicación
	Editar publicación
	Al clicar al chat de tu publicación que te lleve a ver la lista de gente que te ha escrito al post.
Mapa Huertos Urbanos	API Mapa
	Seleccionar huertos urbanos de Valencia
	Mostrar diferencia de distancia entre tu ubicación y los huertos
	Mostrar los huertos por proximidad a tu ubicación
Gestor de Parcelas/*Grid*	Crear *grid*
	Modificar tamaño *grid*
	Mostrar miembros comunidad
	Modificación del color de las casillas del *grid* adecuadas a sus propietarios.
	Guardar cambios en base de datos.
API Plantas	Implementación de la API
	Buscar plantas

	Que cada persona pueda añadirse las plantas que quiera
	Ver los detalles de la planta
	Ver las plantas añadidas
	Eliminar plantas

3.2 Desarrollo basado en *sprints*

Al inicio del *Codethon* empezamos seis miembros como equipo, pero dos abandonaron pronto por distintos motivos personales. Los cuatro miembros finales del equipo (Luis Navarro, Gabriel Rodríguez Díaz, Laijie Ji y el autor de esta memoria), seguimos la metodología *Scrum* y dividimos los 3 meses de duración en 4 *sprints*. Durante estos *sprints*, planificamos y estimamos las tareas en una herramienta llamada Belos (en la Ilustración 8 se muestra un ejemplo de cómo quedó el *sprint* 0 una vez finalizado) y, por último, las desarrollamos. Esta metodología nos ofreció mucha flexibilidad ya que podíamos reestimar y replanificar *sprints*. La organización de las tareas a realizar se hizo de la siguiente forma: Primero abordamos las más importantes y dejamos las menos relevantes para el final. De esta forma, si no teníamos tiempo de completar alguna tarea, esta tendría menos impacto en el resultado final. En el capítulo 4 entraremos en detalle con lo realizado en cada *sprint* y la utilidad de estos.

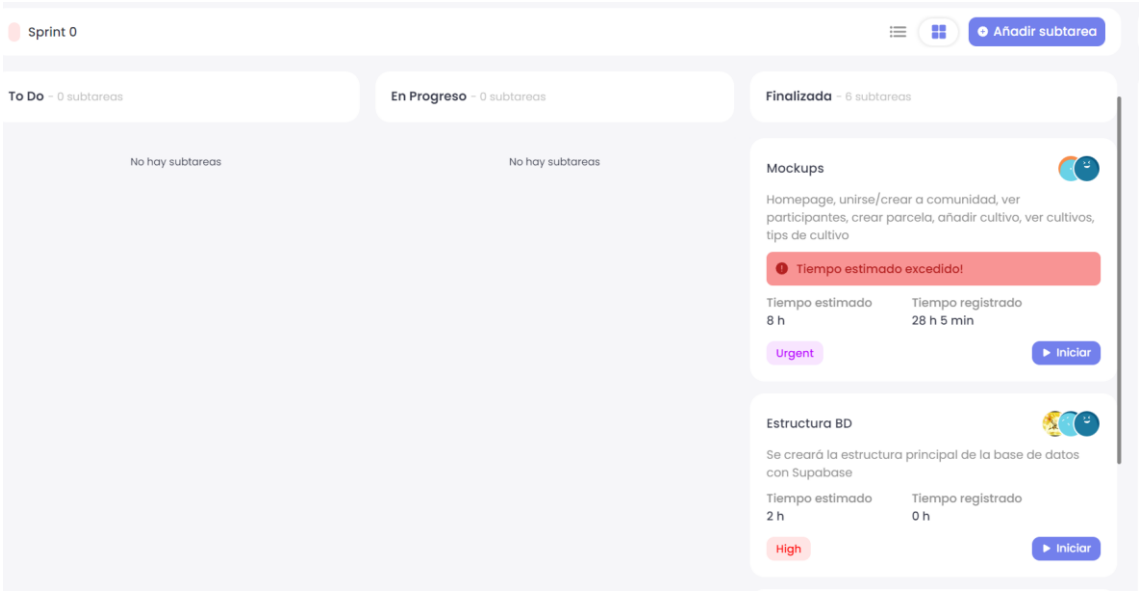


ILUSTRACIÓN 8: USO DE BELOS PARA MEDIR EL TIEMPO

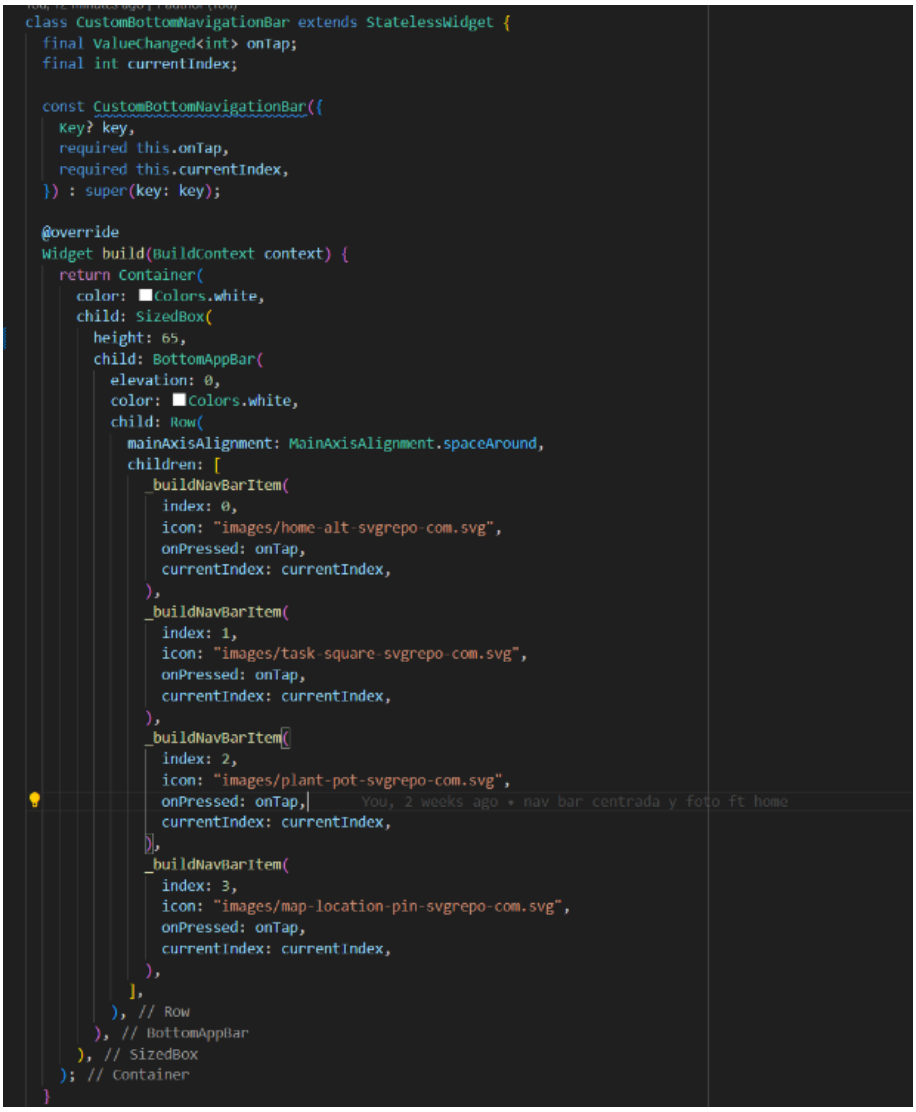
3.3 Arquitectura de **HurbValencia**

En el desarrollo de **HurbValencia** la arquitectura *frontend* ha sido el SDK (*software development kit*) de Flutter mientras que el *backend* ha sido Supabase. En esta sección se comentará en detalle por qué hemos realizado dicha selección.

3.3.1 Flutter

Flutter es un SDK de código fuente abierto para el desarrollo de aplicaciones móviles creado en 2017 por Google y programado en Dart [12], cuya popularidad ha crecido muchísimo en los últimos 2 años. Este es utilizado principalmente para el desarrollo de aplicaciones multiplataforma en Android, iOS y web. Flutter era una de las mejores opciones a elegir, ya que con un mismo código podemos tener la app en los entornos ya mencionados, sin mayores inconvenientes, por ejemplo, para descargar la APK (*Android Application Package*) en nuestro caso sería tan simple como ir a nuestro editor de código fuente (Visual Studio Code) y estando en la carpeta de nuestro proyecto escribir en la terminal *flutter build apk*, y esto nos descargaría la aplicación para Android.

Flutter destaca muchísimo a la hora de programar porque, como tal, todos los elementos de la interfaz son *widgets*. Cada widget tiene un uso distinto y dentro de cada uno puede haber varios *widgets*. Esto puede resultar difícil de comprender al principio, pero proporciona mucha flexibilidad y te permite aprender a usar esta tecnología de una forma muy rápida.



```dart
class CustomBottomNavigationBar extends StatelessWidget {
  final ValueChanged<int> onTap;
  final int currentIndex;

  const CustomBottomNavigationBar({
    Key? key,
    required this.onTap,
    required this.currentIndex,
  }) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Container(
      color: Colors.white,
      child: SizedBox(
        height: 65,
        child: BottomAppBar(
          elevation: 0,
          color: Colors.white,
          child: Row(
            mainAxisAlignment: MainAxisAlignment.spaceAround,
            children: [
              _buildNavBarItem(
                index: 0,
                icon: "images/home-alt-svgrepo-com.svg",
                onPressed: onTap,
                currentIndex: currentIndex,
              ),
              _buildNavBarItem(
                index: 1,
                icon: "images/task-square-svgrepo-com.svg",
                onPressed: onTap,
                currentIndex: currentIndex,
              ),
              _buildNavBarItem(
                index: 2,
                icon: "images/plant-pot-svgrepo-com.svg",
                onPressed: onTap,       You, 2 weeks ago • nav bar centrada y foto ft home
                currentIndex: currentIndex,
              ),
              _buildNavBarItem(
                index: 3,
                icon: "images/map-location-pin-svgrepo-com.svg",
                onPressed: onTap,
                currentIndex: currentIndex,
              ),
            ],
          ), // Row
        ), // BottomAppBar
      ), // SizedBox
    ); // Container
  }
```

ILUSTRACIÓN 9: USO DE BELOS PARA MEDIR EL TIEMPO

```dart
Widget _buildNavBarItem({
  required int index,
  required String icon,
  required ValueChanged<int> onPressed,
  required int currentIndex,
}) {
  return GestureDetector(
    onTap: () => onPressed(index),
    child: Container(
      width: 50,
      height: 50,
      alignment: Alignment.center,
      child: SvgPicture.asset(
        icon,
        width: 30,
        height: 30,
        color: currentIndex == index
            ? OurColors().primeWhite
            : OurColors().navBarDefault,
      ), // SvgPicture.asset
    ), // Container
  ); // GestureDetector
}
}
```

ILUSTRACIÓN 10: EJEMPLO DE USO DE FLUTTER PARTE 2

Creación de la barra de navegación inferior para las páginas de **HurbValencia**

```dart
bottomNavigationBar: CustomBottomNavigationBar(
  onTap: (index) {
    if (index != _currentIndex) {
      setState(() {
        _currentIndex = index;
      });      You, 2 weeks ago • Custom bottom nav bar
      if (_currentIndex == 0) {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => HomePage(widget.userId),
          ), // MaterialPageRoute
        );
      } else if (_currentIndex == 1) {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => ToDo(widget.userId),
          ), // MaterialPageRoute
        );
      } else if (_currentIndex == 2) {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => UserPlants(widget.userId),
          ), // MaterialPageRoute
        );
      } else if (_currentIndex == 3) {
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (context) => MapPage(widget.userId),
          ), // MaterialPageRoute
        );
      }
    }
  },
  currentIndex: _currentIndex,
), // CustomBottomNavigationBar
); // Scaffold
}
```

ILUSTRACIÓN 11: EJEMPLO DE USO DE FLUTTER PARTE FINAL

Uso del *widget* CustomnavBar en la página principal.

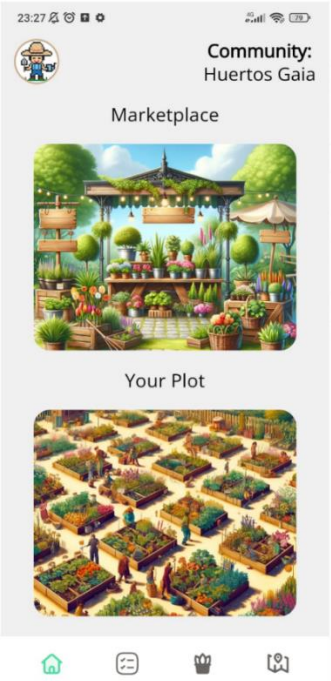


ILUSTRACIÓN 12: EJEMPLO DE LA CUSTOM NAV BAR

Y como podemos ver en la parte inferior de la Ilustración 12 así habría quedado el widget de la barra de navegación inferior en nuestra página principal.

Por último, otras de sus muchas características destacables podrían ser la compatibilidad, es decir, ser compatible con muchas herramientas de desarrollo populares como Visual Studio Code o Android Studio, su alto rendimiento y, por último, lo que se llama "*hot reload*", que nos permite observar los cambios efectuados en el código en tiempo real, tanto en el emulador como en el dispositivo, sin necesidad de reiniciar la aplicación, aumentando así la eficiencia. [13]

3.3.2 Supabase

Al inicio de nuestro proyecto teníamos que decidir qué tecnología íbamos a usar para el *backend*. Entre las distintas ideas surgieron Firebase y Supabase, donde optamos por Supabase. Supabase es una plataforma de *backend* como servicio (BaaS) basada en la nube, que ofrece a los desarrolladores un conjunto extenso de herramientas para desarrollar y administrar servicios *backend* [14].

Supabase se caracteriza por el uso de una base de datos relacional, la cual preparamos al principio del proyecto y fuimos desarrollando y adaptando a nuestras nuevas necesidades. Además de todo esto, el hecho de que fuese una base de datos (BD) relacional nos

facilitaba el aprendizaje, por haberlas usado en el grado. Por otro lado, también cuenta con más aspectos positivos como que es de código abierto, una comunidad en constante crecimiento y te otorga una flexibilidad y facilidad de uso que es de agradecer.

A continuación, en las Ilustraciones 13 y 14 podemos observar cómo hacemos en nuestra aplicación para obtener el id de un *crop* con el servicio de Supabase.

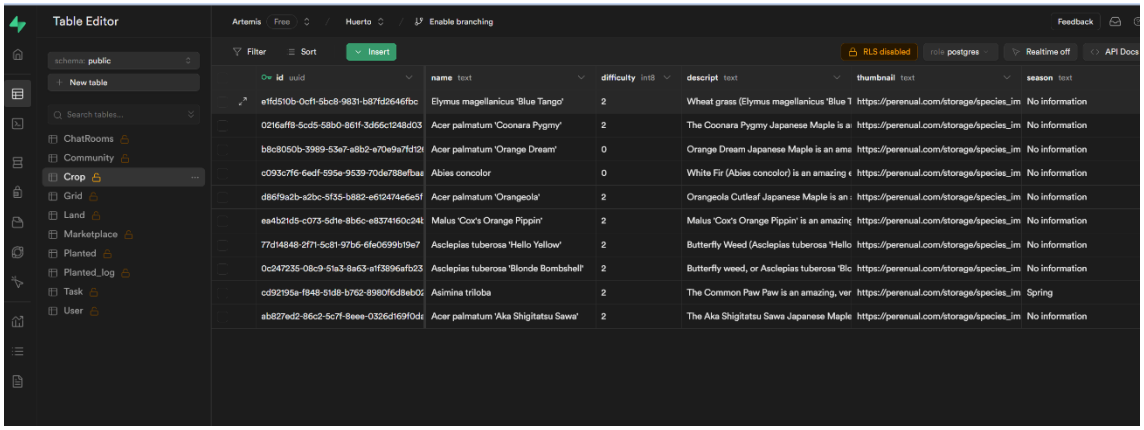


ILUSTRACIÓN 13: EJEMPLO DE USO DE SUPABASE

```dart
class CropSupabase {
  final client = SupabaseService().client;

  Future<void> addCrop(Crop crop) async {
    try {
      Map<String, dynamic> cropMap = Crop.toJson(crop);
      // Llamar a Supabase().addData con el Map del cultivo
      await SupabaseService().addData("Crop", cropMap);
    } catch (e) {
      print('Error al insertar datos: $e');
    }
  }

  Future<Crop?> getCropById(Guid id) async {
    final data = await SupabaseService().readDataById("Crop", id.value);
    if (data.length == 0) {
      return null;
    } else {
      final crop = Crop.fromJson(data);
      return crop;
    }
  }
}
```

ILUSTRACIÓN 14: EJEMPLO DE USO DE SUPABASE EN CÓDIGO

4. **HurbValencia**: Desarrollo por *sprints*

En este punto hablaremos de forma desarrollada sobre lo comentado en el *backlog* del capítulo anterior y veremos unos *mockups* elaborados de manera rápida, debido a la limitación temporal de tres meses con la que se contaba para el desarrollo de la aplicación. Explicaremos de forma detallada la utilidad de cada ventana, su unidad de trabajo (UT) relacionada y las pruebas de aceptación realizadas para cada UT. En los anexos podremos apreciar la interfaz final, viendo la gran flexibilidad que nos proporciona la metodología Scrum y destacando la gran diferencia visual respecto los mockups.

4.1 *Sprint* 0:

En este primer *sprint*, que se desarrolló en equipo, pensamos cual iba ser nuestra idea de aplicación. Luego, se realizó una planificación de tareas entre todos los miembros dando lugar a las siguientes: 1) Realizar los mockups de los que nos encargaríamos Rubén y el autor de este TFG, 2) estructurar la base de datos en Supabase de lo que se encargarían Luis Navarro y, también el autor de este TFG, 3) investigar tecnologías, tarea conjunta del equipo, en la que llegamos a la conclusión de usar Flutter y 4) formarnos en dicha tecnología.

4.2 *Sprint* 1:

Para el *sprint* 1 de nuevo nos reunimos todo el equipo e hicimos una planificación de tareas, en la que el autor de este TFG se encargaría de realizar toda la pestaña inicial que incluiría la gestión de las comunidades, es decir, realizar como una especie de iniciar sesión en la que el usuario podrá darle clic a un botón para unirse o crear una comunidad. Evidentemente, esto iba de la mano de realizar las 2 páginas enteras de unirse y crear una comunidad que estaban enlazada a nuestra base de datos de Supabase.

Luis Navarro se encargaría de hacer las funcionalidades de la Agenda/Todo. El autor del TFG le ayudaría a realizar toda la parte del diseño de esa ventana y añadiría una nueva funcionalidad, ordenar la lista de tareas por prioridad. Por otra parte, Michelle, Rubén, Laijie y Gabriel, se encargaron de realizar una página para el Clima que permitiera: 1) avisar a los usuarios en casos de lluvia grave para que protegiesen de alguna forma a sus plantas y 2) de hacer un Mapa que indicase cuál era tu ubicación y mostrase todos los

huertos urbanos de la ciudad de Valencia y cuáles eran los más cercanos al usuario. A continuación, vamos a hablar detalladamente de las UTs realizadas en este *sprint:*

4.2.1 Pagina Unirse o crear una comunidad

En la Ilustración 15 se puede observar la página a la que accedemos tras iniciar sesión. Esta ventana como podemos observar en la Ilustración 16 sirve para que podamos unirnos a una comunidad de huertos urbanos ya creada, a través de una ID que obtendremos mediante un QR que se muestra en la Ilustración 29 o como vemos en la Ilustración 17 para crear una nueva comunidad.

Por falta de tiempo, nos quedaron pendientes de programar las UTs de unirse como invitado y a varias comunidades. Por ello, para acceder a la página principal y a las características de la app, si no perteneces previamente a una comunidad tendrás que crear una nueva y no te podrás unir simultáneamente a varias comunidades.

ILUSTRACIÓN 15: MOCKUP PRIMERA PÁGINA

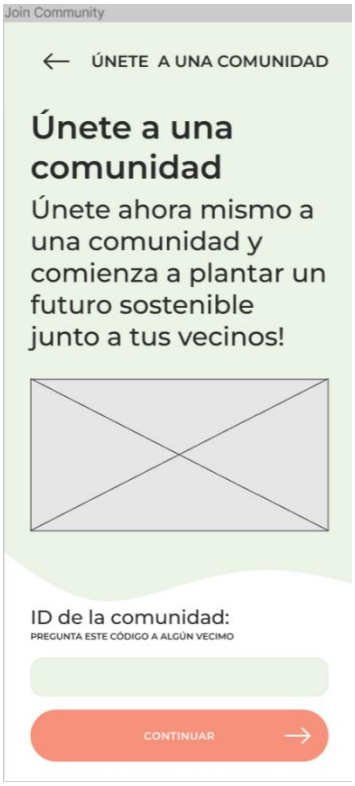


ILUSTRACIÓN 16: MOCKUP UNIRSE COMUNIDAD

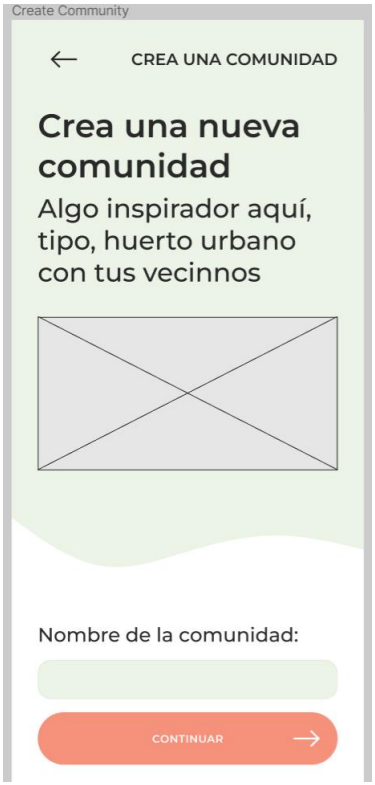


ILUSTRACIÓN 17: MOCKUP CREAR COMUNIDAD

UTS vinculadas:

• Crear comunidad.

• Unirse a una comunidad

• Unirse a varias comunidades

• Unirse como invitado

En cuanto a las pruebas de aceptación para Unirse a una Comunidad hemos establecido las siguientes:

- En caso de que escribas el ID incorrecto se te devolverá un mensaje de error avisándote de ello.

En cuanto a la parte de crear una comunidad:

- Cuando te unas por ID a la comunidad podrás crear la comunidad con el nombre que quieras, esté repetido o no.

4.2.2 Página principal (*Homepage*).

La Ilustración 19 nos muestra la idea general de cómo iba ser la página principal de nuestra aplicación, a la que accederías tras haberte unido a una comunidad de forma correcta o al haber creado la comunidad. En esta página mostraríamos el *widget* del tiempo, la parte de nuestra parcela/*grid*, que explicaremos más adelante, y el apartado "mis tareas", que representaría la agenda. Este diseño de *mockup* tendría que ser adaptado a la realidad, ya que cada vez que añadíamos alguna nueva funcionalidad como podría ser el *marketplace* haría que esta página fuese mejor o peor. Por tanto, tuvimos que readaptarla por completo. Esto lo veremos más adelante cuando se muestre la versión final de la app.

Además, hay que mencionar que, al no ser expertos en aplicaciones móviles, se nos olvidó por completo en el *mockup* hacer lo que se dice el NavBar, siendo esta la barra de navegación que irá en la parte más inferior de la *homepage* y que vemos en la Ilustración 18.

ILUSTRACIÓN 18: BARRA DE NAVEGACIÓN

ILUSTRACIÓN 19: PÁGINA PRINCIPAL

UTs vinculadas:

• Para la barra de navegación necesitaremos tener las páginas de la agenda, mapa y API de plantas.

• QR

• Editar perfil, modo oscuro

• Para la vista general necesitaremos *el grid,marketplace* y API del clima, ya realizados.

En cuanto a las pruebas de aceptación para Unirse a una Comunidad hemos establecido las siguientes:

- Necesitaremos una página principal extremadamente intuitiva para que el usuario sepa en todo momento lo que puede hacer.
- Todas las rutas tendrán que funcionar de forma adecuada para navegar bien entre las distintas páginas.

- Imágenes e iconos que representen de forma clara lo que queremos mostrar. Por ejemplo, para poder acceder a la API de las plantas, tendrás que clicar sobre la planta de la barra de navegación.

4.2.3 Página de la Agenda/ToDo

Las Ilustraciones 20 y 21 ofrecen unos mockups casi idénticos a la página en la app final. Esta muestra una lista de tareas que puedes ir añadiendo e indicando la prioridad, alta, media o baja, y que se verá reflejada en el círculo que vemos a la izquierda del título de la tarea. El círculo será de color rojo si la prioridad es alta, amarillo si es media y verde si es baja.

En el caso del mockup de la Ilustración 20 nos encontramos sobre el *tab*/ventana de "Por hacer" que está subrayado en verde. En caso de que terminemos la tarea, clicaremos sobre la *checkbox* que se ve en la esquina superior derecha de la tarjeta y está pasará a la parte de "Hecho" (Ilustración 21), donde si clicamos sobre dicho texto pasaremos a la ventana de "hecho" que nos mostrará el texto subrayado y las tareas que ya hemos realizado con la *checkbox* marcada.

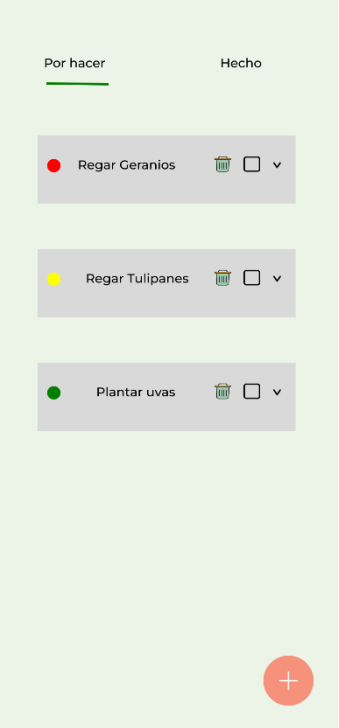


ILUSTRACIÓN 20: MOCKUP DE LA AGENDA "POR HACER"

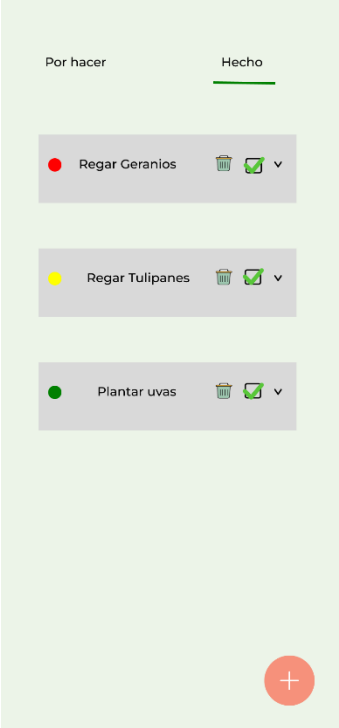


ILUSTRACIÓN 21: MOCKUP DE LA AGENDA "HECHO"

UTs vinculadas:

• Añadir tareas.

• Eliminar tareas.

• Modificar tareas.

• Enviar tareas a “hecho”.

• Ordenar tareas por prioridad.

En cuantos a las pruebas de aceptación para la agenda hemos establecido las siguientes:

- Se guardarán las tareas por usuario.
- Se podrán escribir tantas tareas como el usuario desee.
- En caso de borrar una tarea saldrá un mensaje de confirmación.
- Si la tarea creada no tiene nombre te saldrá un error de validación para indicar que tienes que escribir el título de la tarea.

4.3 *Sprint* 2:

En este *sprint* es donde vimos el verdadero potencial de la metodología ágil, ya que tanto Luis Navarro como el autor del TFG cumplieron con sus tareas. Sin embargo, el resto de los compañeros no las terminaron, entre otros motivos porque, como se ha comentado previamente, dos miembros del equipo lo abandonaron por falta de tiempo para participar en la *Codethon* (Michelle y Rubén).

En este *sprint* reorganizamos las tareas que nos tocaba a cada uno, donde Luis y Daniel serían los que implementarían la API del mapa. Además de esto, Daniel 1) retocaría las interfaces actuales para, así, hacerlas más agradables y cómodas, arreglando ciertos bugs que dificultaban la interacción sencilla con la app, y 2) desarrollaría las interfaces de las páginas que iban a realizar tanto Gabriel como Laijie.

Por último, Gabriel y Laijie tenían que acabar la implementación de la API del clima del pasado *sprint*, realizar una API de plantas para poder guardar las plantas que más te gusten en una lista, y realizar la página de la gestión de la parcela de una comunidad de huertos urbanos.

Procedemos a detallar las UTs realizadas en el *sprint*:

4.3.1 Mapa de Huertos Urbanos

En la Ilustración 22 observamos el *mokcup* de lo que queremos que sea nuestra página del mapa. Aquí tendremos una lista de todos los huertos urbanos de Valencia. De este modo, donde vemos "Huerto Benimaclet", nos saldrán una lista de huertos que, además, nos mostrará su proximidad a nuestra ubicación y nos indicará la distancia en metros existente entre el usuario y el huerto. Por último, al dar clic sobre el icono verde de la ubicación se nos abrirá el Google Maps, indicándonos directamente la ruta exacta desde nuestra posición a la del huerto.

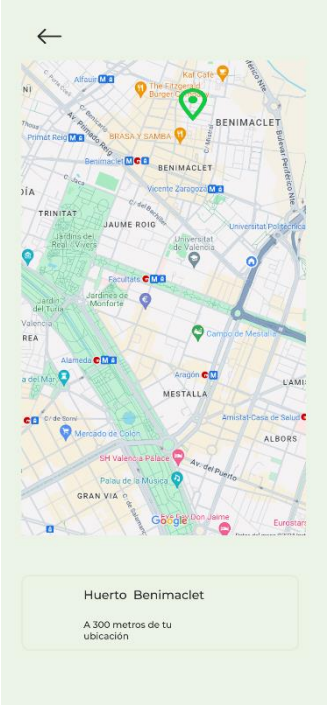


ILUSTRACIÓN 22: MAPA

UTs vinculadas:

• API Mapa

• Lista huertos urbanos de Valencia

• Mostrar diferencia de distancia entre tu ubicación y los huertos

• Mostrar los huertos por proximidad a tu ubicación

En cuantos a las pruebas de aceptación para el Mapa hemos establecido las siguientes:

- Al darle clic sobre el icono de la ubicación del huerto que queramos se nos abrirá el Google Maps, indicando la ruta exacta desde nuestra ubicación hasta dicho huerto.
- Será un mapa parecido al de Google Maps, para que de esta forma sea muy intuitivo.

4.3.2 Página del Tiempo

En la Ilustración 23 tenemos una vista de cómo queremos que sea nuestra página del clima. Esta mediante una API mostrará el clima actual y el de los próximos días. La parte de las notificaciones es algo que no nos dio tiempo a realizar ya que priorizamos otras tareas.

ILUSTRACIÓN 23: MOCKUP DEL CLIMA

UTs vinculadas:

• API Clima

En cuantos a las pruebas de aceptación para el QR hemos establecido las siguientes:

- Ver clima de los próximos días.
- Notificaciones sobre la lluvia y su cantidad.

4.3.3 Página de información y almacenamiento de las Plantas

En la Ilustración 24, pretendemos mostrar en el *mockup* una ventana donde poder almacenar las plantas que tenemos en nuestro huerto. Estas provendrán de una API, cuya idea es que al clicar sobre estas plantas se muestre información básica, como cantidad de agua necesaria, tiempo de crecimiento, estación y clima adecuados para plantarla.

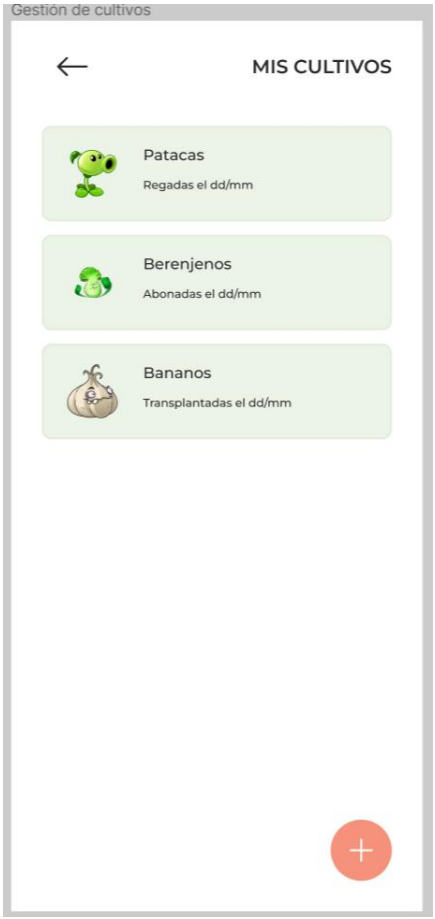


ILUSTRACIÓN 24: MOCKUP LISTA DE PLANTAS

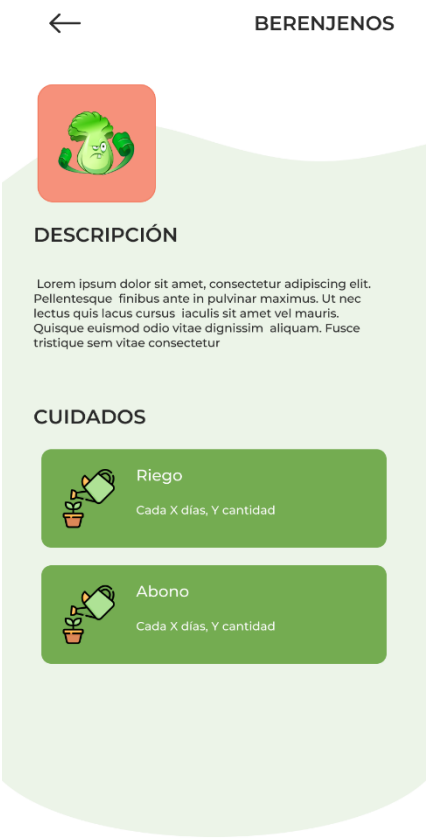


ILUSTRACIÓN 25: MOCKUP INFORMACIÓN PLANTA

UTs vinculadas:

• API Plantas

• Añadir plantas

• Ver información de las plantas

• Eliminar plantas

• Ver los detalles de la planta

• Ver plantas añadidas

En cuantos a las pruebas de aceptación para la API de las plantas hemos establecido las siguientes:

- Que cada usuario pueda almacenar las plantas que quiera y que se guardarán en la BD. De esta forma el usuario, al entrar, siempre podrá ver que plantas tiene en su huerto
- La información de las plantas será siempre correcta y breve, destacando las partes más importantes de la información para que sea lo más fácil de entender por el usuario.
- En caso de querer borrar una planta saldrá un mensaje de confirmación de la acción.

4.4.4 Página de la parcela de la Comunidad

En la Ilustración 26 observamos el *mockup* de la gestión de parcelas. Tendremos una matriz con un tamaño modificable, cuya idea principal es que el creador de la comunidad al clicar sobre el nombre del usuario que quiera de la comunidad podrá cambiar el color del "cuadrado", indicando así que esa parte de la parcela es de dicho usuario.

Para comprenderlo mejor en la Ilustración 26 también observaríamos como la primera fila le corresponde a Dani, la segunda a Gabriel, etc. Este tamaño, como ya se ha dicho, se puede modificar dependiendo de lo grande que sea nuestra parcela. Además, a la hora de modificar el tamaño, nos saldrá un aviso para modificarlo ya que se borrará el estado actual de la matriz y se pondrán todos los "cuadrados" de color blanco, volviendo al estado inicial. Evidentemente, todo esto estará guardado en la base de datos en su correspondiente comunidad. En la versión final que veremos en los anexos observaremos un ejemplo mejor.

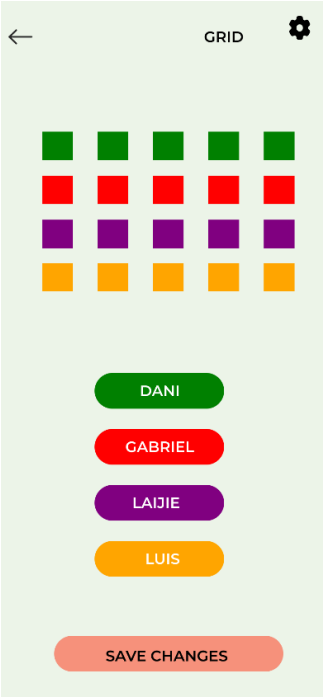


ILUSTRACIÓN 26: MOCKUP GESTIÓN DE PARCELAS

UTs vinculadas:

• Crear *grid*.

• Modificar tamaño *grid*.

• Mostrar miembros de la comunidad.

• Pintar *grid*.

En cuantos a las pruebas de aceptación para la gestión de parcelas hemos establecido las siguientes:

- A la hora de modificar el *grid* te saldrá un aviso de confirmación.
- Solo el creador de la comunidad podrá modificar a quien le corresponde cada parte de la parcela para evitar posibles casos de caos.
- Los cambios se guardarán en la página para todos los usuarios pertenecientes a dicha comunidad.

4.4 *Sprint* 3:

En el último y más intenso de los *sprints*, el autor del TFG debía implementar una IA (Inteligencia Artificial) para reconocer el estado de las plantas, mejorar la interfaz. Pero, finalmente, el tema de la IA quedaría descartado, ya que de ninguna forma se encontró alguna API funcional gratuita, y al quedar un par de semanas para que acabase el *sprint* final y con ello la *Codethon*, se descartó, como ya se ha comentado. Por lo tanto, todos los esfuerzos se centraron en mejorar la interfaz actual, lo que le llevó a un rediseño completo de ella. Es por ello por lo que se puede observar claramente la mejora del diseño final de la app en comparación con los mockups. Además, en caso de que los compañeros realizasen alguna funcionalidad nueva, también se realizaría dicha interfaz y, por último, con la ayuda de Luis, realizaría toda la parte referente a iniciar sesión y registrarse.

Por el contrario, se pudieron realizar dos nuevas páginas. Un "*marketplace*" que es un mercado en el que se puede poner una publicación sobre algo que se vende, a la vez que ves las distintas publicaciones de los miembros de la misma comunidad. Además, en caso de que alguien esté interesado en tu publicación o esté interesado en la de otra persona se podría establecer una conversación por chat. Finalmente, completamos en la página principal una funcionalidad para ofrecer un QR con el objetivo de poder compartir con amigos o nuevos miembros de la comunidad el *id* de esta para que así se puedan unir a ella.

Las UTs realizadas fueron las siguientes:

4.4.1 Iniciar sesión y registro

Como vemos a continuación en las Ilustraciones 28 y 29, nada más el usuario abra la aplicación por primera vez, deberá registrarse completando los campos requeridos y en caso de que tenga cuenta deberá iniciar sesión para acceder a la siguiente página.

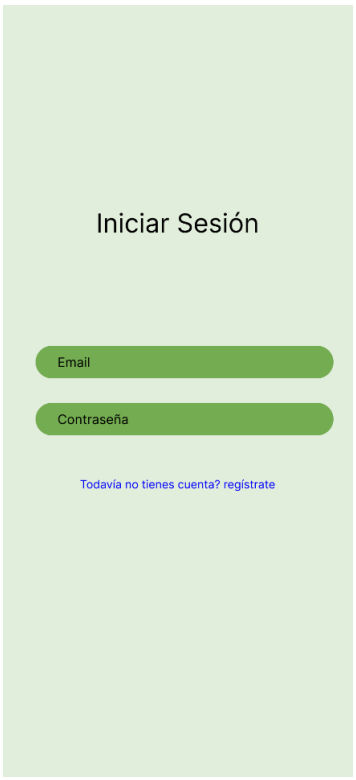


ILUSTRACIÓN 27: MOCKUP INICIAR SESIÓN

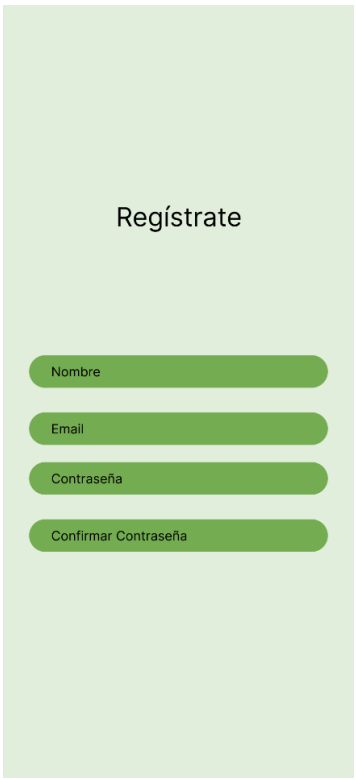


ILUSTRACIÓN 28: MOCKUP REGISTRARSE

UTs vinculadas:

• Registro con Supabase

• Login con Supabase

En cuantos a las pruebas de aceptación del inicio de sesión hemos establecido las siguientes:

- Se comprobará que el usuario y la contraseña sean correctos. En el caso de que lo sean y en algún momento ya te hayas unido a una comunidad se te redirigirá a la página principal (Ilustración 19) y en caso de que no estuvieses unido a una comunidad se te redirigirá a la parte de crear o unirse a una comunidad (Ilustración

15). En caso de que las credenciales sean incorrectas se te enviará un mensaje de error, avisándote de que te has equivocado en el email o la contraseña.

En cuanto al registro:

- Se comprobará que el email está en un formato adecuado.
- La contraseña deberá tener más de 6 caracteres.
- Las contraseñas y confirmar contraseña tendrán que coincidir.
- Si algún campo está vacío, se nos avisará de ello.

4.4.2 Compartir QR

A la Ilustración 29 accederemos a través de darle un clic sobre el nombre de la comunidad de la página principal (Ilustración 19) Escaneando el QR o dándole clic sobre el botón de *Coppy Community ID* haré exactamente lo que indica su traducción; es decir, nos dará el ID de la comunidad en la que nos encontramos para así poder enviárselo de una forma muy cómoda a nuestros compañeros.

Podemos fijarnos en que la página del QR, junto a las que veremos adelante cuando hable en detalle del Marketplace, están en inglés ya que la app se diseñó originalmente en castellano, pero el idioma final acabó siendo el inglés.

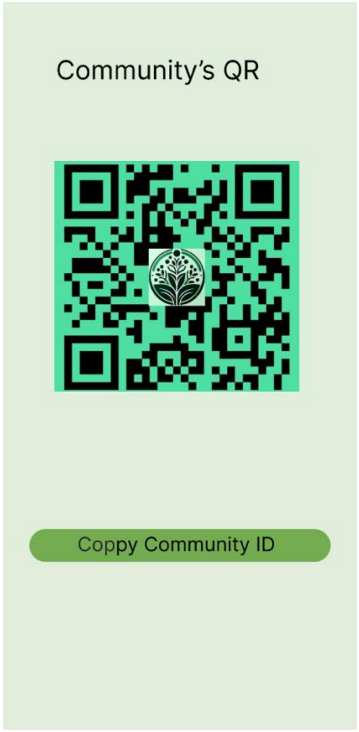


ILUSTRACIÓN 29: MOCKUP QR

UTs vinculadas:

• QR

En cuantos a las pruebas de aceptación para el QR hemos establecido las siguientes:

- Se podrá copiar el QR clicando sobre el botón de copiar.
- Se podrá escanear el QR directamente.

4.4.3 Página del *Marketplace*

Esta ventana se nos ocurrió al final del último *sprint*. Aquí, cada usuario podrá subir una o varias publicaciones de lo que quiera y estas podrán ser vistas por todos los miembros de la misma comunidad. En el ejemplo de la Ilustración 30 tenemos al usuario Moncho que dice que vende patatas. En caso de que nosotros queramos comunicarnos con el simplemente tendremos que darle al icono de chat y ahí entraremos directamente en un chat con Moncho. Esto lo podemos observar en la Ilustración 31. El usuario creador de la publicación, en este caso Moncho, podrá clicar sobre el chat de su publicación (Ilustración 30) para ver la lista de personas que le han hablado, lo que se muestra en la Ilustración 32. El objetivo de todo esto es hacer la aplicación más útil (chat de la comunidad e intercambios con los vecinos)

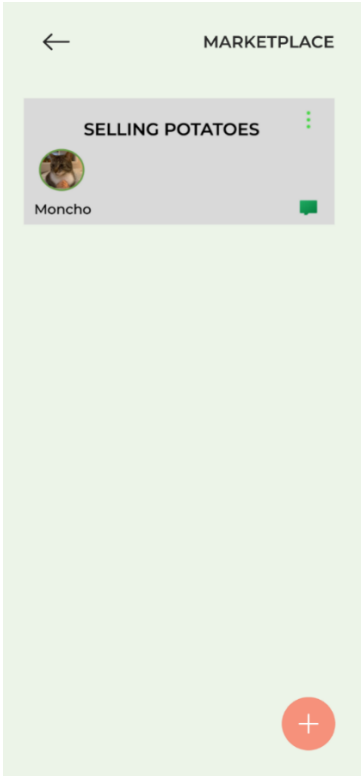


ILUSTRACIÓN 30: MOCKUP VISTA GENERAL MARKETPLACE

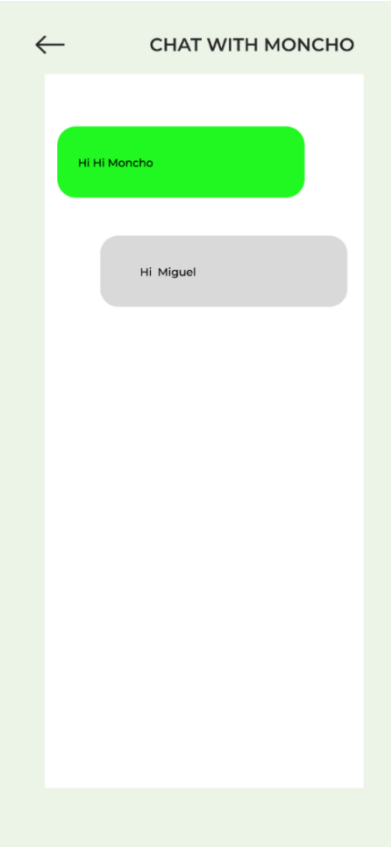


ILUSTRACIÓN 31: MOCKUP CHAT DEL CLIENTE CON MONCHO

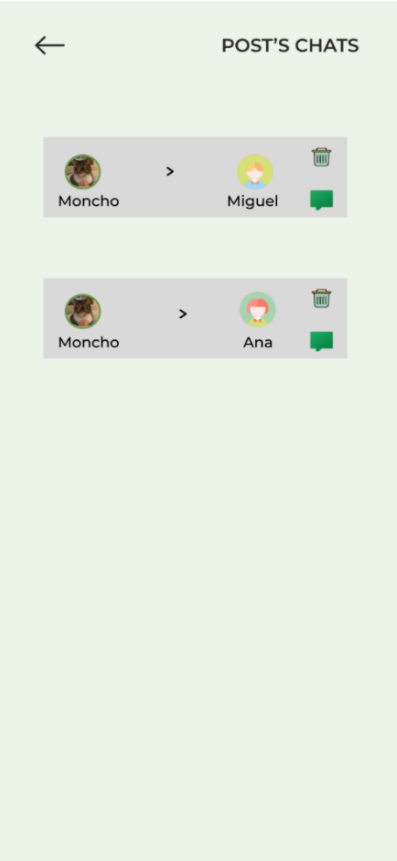


ILUSTRACIÓN 32: MOCKUP LISTA DE CHATS DE MONCHO CON SUS CLIENTES

UTs vinculadas:

• Crear publicación.

• Avatar con nombre de la persona creadora del post.

• Eliminar publicación.

• Editar publicación.

• Lista de chats de los clientes.

• Chat con comprador.

En cuantos a las pruebas de aceptación para la gestión de parcelas hemos establecido las siguientes:

- Al clicar sobre el chat del post de una persona que te lleve a su chat.
- Al clicar al chat de tu publicación que te lleve a ver la lista de gente que te ha escrito al post.
- A la hora de eliminar una publicación te saltará un mensaje de aviso.

5. Detalles de la implementación

En apartados anteriores, hemos hablado del *frontend* y del *backend* de la aplicación. En este, comentaremos en detalle las tecnologías utilizadas para la realización de la aplicación. Además, detallaremos la importancia de Jakob Nielsen y sus principios de diseño y, por último, puntualizaremos cómo hemos estructurado el proyecto para tener un código limpio y aplicar buenas prácticas.

5.1 Tecnología empleada.

De acuerdo con la arquitectura de la aplicación, presentada en el capítulo 3, destacamos las distintas tecnologías que nos han facilitado la realización del proyecto.

5.1.1 GitHub

Para trabajar de forma coordinada y tener el proyecto constantemente actualizado usamos un sistema de versión de controles, GitHub. Aquí, cada usuario puede realizar diversos comandos, entre ellos, puede crear lo que se denominan "ramas", que sirven para facilitar mucho el desarrollo de una aplicación en conjunto. A continuación, se explica un ejemplo del uso de *gitHub* en el desarrollo de **HurbValencia**. Cada miembro del equipo, a la hora de desarrollar una tarea, se crearía su propia rama a partir de la "*main*" y haría su tarea sin tener conflictos con nadie. Una vez acabada y revisada, subiría sus cambios a la rama *main*, siendo esta la principal del proyecto (ver Ilustración 33).

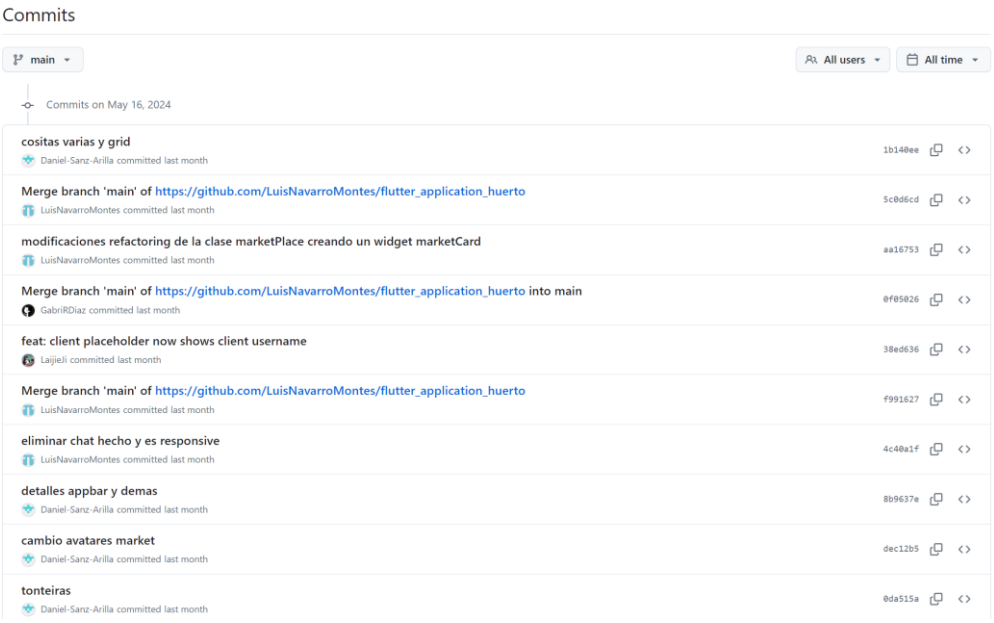


ILUSTRACIÓN 33: EJEMPLO DE USO DE GITHUB

No obstante, GitHub no es solamente eso, sino que como repositorio compartido de código permite, entre otras cosas, recuperar versiones previas de una forma sencilla, pudiendo así deshacer cambios (lo cual fue muy útil para el equipo).

5.1.2 Android Studio

Android Studio es un entorno de desarrollo oficial para la plataforma Android, basado en el popular IDE IntelliJ IDEA [15]. Este se puede utilizar para programar como tal, pero el autor del TFG lo utilizó en modo emulador. Android Studio nos ofrece una enorme facilidad a la hora de probar nuestra aplicación en el móvil, pues nos ofrece un emulador en el que seleccionamos el tipo de móvil que queremos emular y una vez seleccionado, probar de forma muy simple nuestra aplicación.

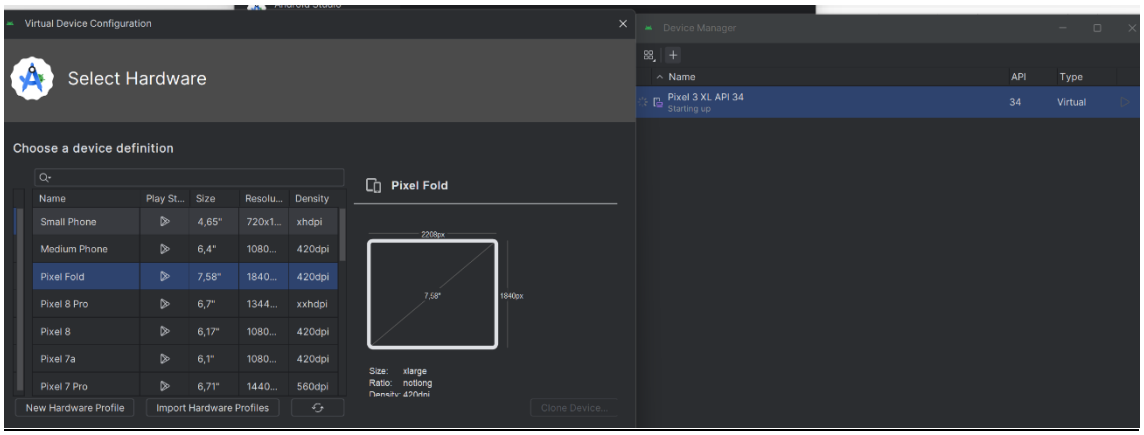


ILUSTRACIÓN 34: EJEMPLO DE USO DE ANDROID STUDIO

Como vemos en la Ilustración 34, este nos ofrece una amplia variedad de móviles para emular, en mi caso, emulé todo el desarrollo con el móvil Pixel 3 XL API 34.

5.1.3 Weatherapi

Esta es un API simple de la página [16] que proporciona los datos climatológicos necesarios para nuestra aplicación. Entre ellos hay datos como cuándo va a llover, los grados y las probabilidades de lluvia (ver Ilustración 35).

ILUSTRACIÓN 35: EJEMPLO DEL RESULTADO FINAL DE WEATHERAPI

5.1.4 Perenual API

Esta API [17] y nos proporciona todos los datos necesarios sobre las plantas para nuestra aplicación, entre ellos algunos como cada cuanto hay que regar las plantas, su dificultad, donde ponerlas o en que estación plantarlas. En las siguientes Ilustraciones (36,37,38,39) voy a mostrar cómo han quedado las páginas de nuestra app que hacen uso de esta API.

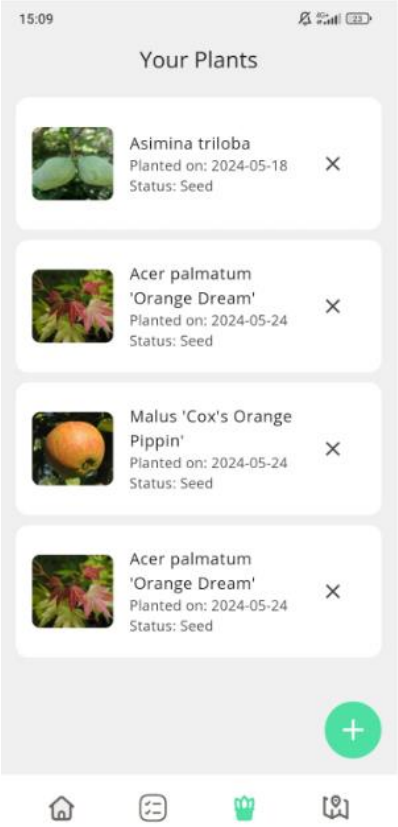


ILUSTRACIÓN 36: LISTA DE PLANTAS CON PERNUAL API

ILUSTRACIÓN 37: BÚSQUEDA DE PLANTAS CON PERENUAL API

ILUSTRACIÓN 38: INFORMACIÓN DE LA PLANTA

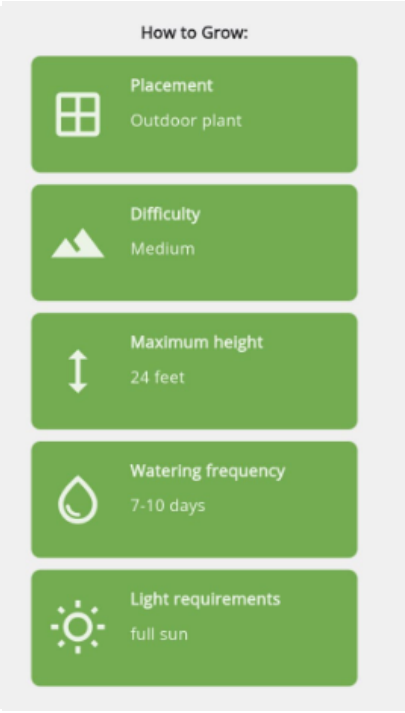


ILUSTRACIÓN 39: INFORMACIÓN MÁS IMPORTANTE

5.1.5 Map API

Para el mapa se ha utilizado [18], una API que nos proporciona un mapa visualmente atractivo y que funciona de forma correcta.

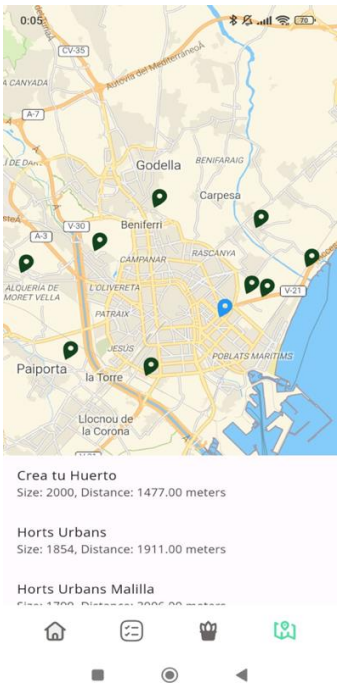


ILUSTRACIÓN 40: EJEMPLO DEL USO DE MAPTILER

Se distribuyeron las localizaciones exactas de los distintos huertos urbanos de Valencia sobre el mapa, dando lugar a la Ilustración 40 que muestra nuestra ubicación y todos los huertos urbanos en el mapa. Además, permite visualizar una lista de los huertos urbanos con sus nombres, ordenados por proximidad a nuestra ubicación.

5.2 Desarrollo de la Interfaz y principios de diseño.

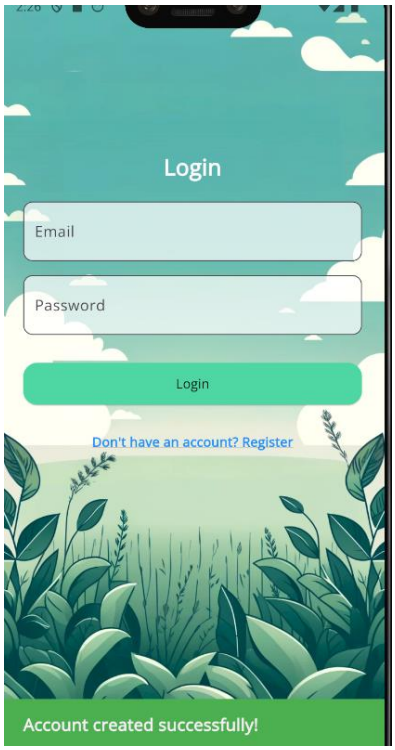
Como ya hemos explicado en varias ocasiones, la interfaz de nuestra aplicación es una de sus componentes más importantes, ya que de poco sirve que tengas muy buenas funcionalidades si la interfaz no es amigable o es de difícil comprensión.

Esta fue una de mis principales tareas durante el proyecto, el diseño y desarrollo de toda la interfaz en general, incluyendo los *mockups*. Para mejorar la interfaz desde un principio, se repasaron conceptos básicos sobre diseño centrado en el usuario, cuestiones sobre qué colores usar, cuándo, cómo ,y adaptarlos a nuestro estilo de aplicación. Además, se siguieron los principios de usabilidad de Jacob Nielsen, para comprender cómo hacer una buena interfaz. Hay que destacar que este estudio se ve reflejado conforme avanza la aplicación, ya que su diseño inicial era realmente simple en comparación con el final.

5.2.1 Principios de usabilidad de Jacob Nielsen

Este investigador consiguió marcar 10 principios de usabilidad para el diseño web que se siguen usando hoy en día [19 y 20]. Estos, también conocidos como las heurísticas de usabilidad de Nielsen, establecen las bases para realizar interfaces efectivas y eficientes, es decir, proporcionan una experiencia satisfactoria para el usuario, siendo los principios de una buena interfaz. A continuación, veremos ejemplos de algunos de estos principios usados en nuestra aplicación para lograr una interfaz óptima.

• **Principio de la Visibilidad y relación entre el sistema y el mundo real:** La visibilidad indica que nuestra interfaz debe mantener siempre informado al usuario de lo que está ocurriendo. Por ejemplo, si el usuario se registra con éxito, verá una notificación como la que podemos observar en el inferior de la Ilustración 41. De esta forma se evitarán confusiones.



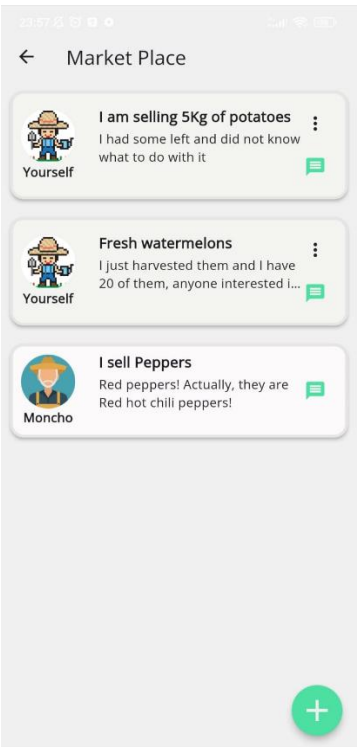


ILUSTRACIÓN 41: PRINCIPIO DE VISIBILIDAD **ILUSTRACIÓN 42: CONTROL Y LIBERTAD DEL USUARIO**

• **Control y libertad del usuario** El usuario siempre debe tener cierta autonomía en la aplicación, poder explorar la aplicación cuanto quiera y retroceder a un paso anterior si así lo desea. Esta ventaja la podemos observar, por ejemplo, con la barra de navegación inferior, donde, clicando sobre cualquiera de los iconos, podrán acceder a cualquiera de las distintas páginas principales y luego, en caso de que estemos dentro de los posts del *marketplace*, darle la opción de que le dé a la flecha hacia atrás por si se ha equivocado. Esto lo podemos apreciar en la Ilustración 42, donde tenemos el icono de la flecha en la esquina superior izquierda.

• **Prevención de errores:** Este principio es uno de los más importantes de la heurística e indica que hay que evitar y prever cualquier error que pueda cometer el usuario. En la Ilustración 43 mostramos el ejemplo de un usuario que ha presionado sobre el icono de la basura que sirve para eliminar una tarea. Al darle clic sobre este nos saltará el siguiente "*pop up*", que preguntará al usuario si está seguro de lo que quiere hacer, previniendo equivocaciones.

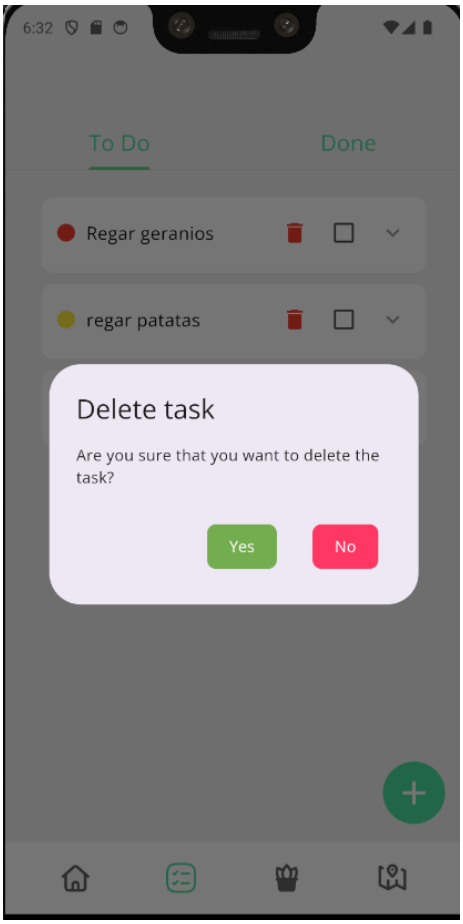


ILUSTRACIÓN 43: PREVENCIÓN DE ERRORES

• **Reconocer antes que recordar:** Este principio señala que en vez de hacer que el usuario tenga que acordarse de ciertos aspectos, solo los tenga que reconocer. Esto se puede hacer mediante menús, iconos y selectores, entre otros. A lo largo de la aplicación tenemos varios ejemplos como el siguiente (ilustración 44), donde si nos fijamos en la parte del toDo, a la hora de añadir tarea podemos observar que el Add task está a la izquierda y las tareas de aceptar siempre se encontrarán a la izquierda mientras que las de cancelar estarán a la derecha. Igualmente, esto se observa en la Ilustración 43 donde el sí está a la izquierda y el no a la derecha. Luego, en vez de hacer que el usuario tenga que recordar que tres tipos de prioridad hay, le ofrecemos un selector donde simplemente tendrá que escoger el que quiera.

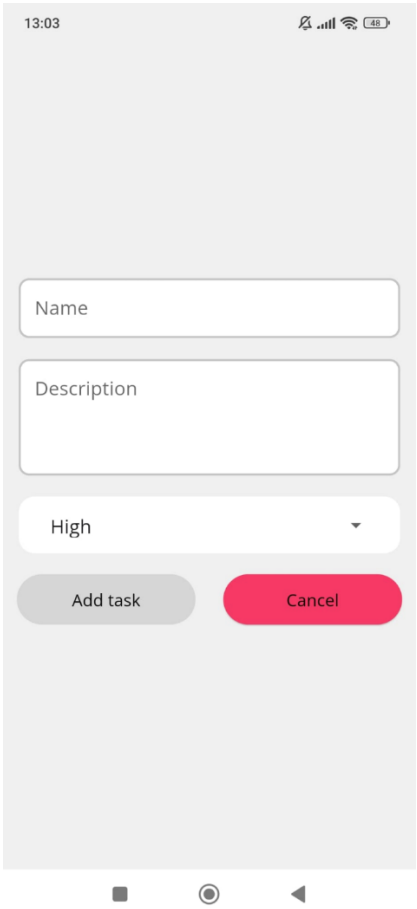


ILUSTRACIÓN 44: RECONOCER ANTES QUE RECORDAR

• **Consistencia y estándares:** Este principio consiste en adaptarse a unas normas no escritas basándose en experiencias previas de los usuarios y mantenerlas a lo largo de tu aplicación. Por ejemplo, los botones verdes se suelen asociar a acciones de aceptar o confirmar mientras que los rojos a acciones de cancelar o rechazar, además esto no solo lo puedes hacer una vez en tu aplicación, es decir, si tienes varios botones de aceptar y varios de cancelar estos siempre tendrán que seguir un estilo similar para no confundir al usuario. Esto se aplica también para los colores en la aplicación, iconos y estilo. Así, por ejemplo, si usamos generalmente colores fríos, no es lógico que de repente hagamos una página con colores cálidos.

En la Ilustración 45 tenemos el archivo Colors.dart de nuestro proyecto, donde tendríamos todos los colores a usar por nuestra app y, como vemos, son todos colores fríos, manteniendo una consistencia, menos algunas excepciones que son usadas para botones de cancelar y rechazar.

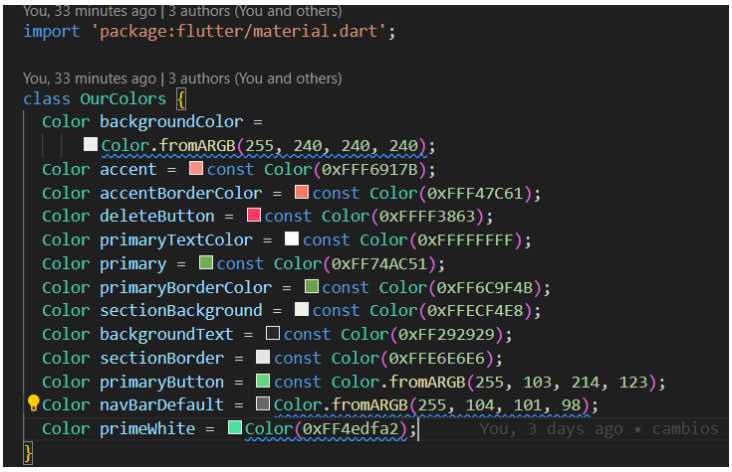


ILUSTRACIÓN 45: CONSISTENCIA Y ESTÁNDARES

Otro ejemplo podría ser el botón de añadir que es circular (+) que lo podemos observar en varios lugares de la aplicación como la Ilustración 42, que siempre tiene el mismo color y que se usa ese botón con el mismo color en todas las páginas en las que queremos que el usuario pueda añadir algo.

• **Estética y diseño minimalista:** La heurística mencionada es una que aplicamos prácticamente desde el inicio, empezando por los *mockups*, ya que es una de las más importantes. Esta describe que tenemos que mostrar solamente la información necesaria, implicando que no hay que tener sobrecargas, es decir, para conseguir una buena interfaz ha de ser simple y eficiente. Las sobrecargas innecesarias llevan a confusiones y distracciones que el usuario no necesita.

• **Flexibilidad y eficiencia de uso:** Es muy importante lograr que todos los usuarios, independientemente de su experiencia, puedan navegar de forma agradable y fácil por nuestra aplicación y, es por ello, por lo que hay que lograr esta flexibilidad. El ejemplo de dicha heurística lo podemos ver a lo largo de toda la aplicación, ya que usamos iconos reconocibles y formas fáciles de navegación, como lo son las flechas hacia atrás para volver a la ventana anterior, o la barra inferior para la navegación entre las distintas páginas principales.

• **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** Aquí se explica que cuando se produce un código de error, en vez de por ejemplo enviar al usuario a una página de error 404, hay que explicarle por qué ha sucedido para que así no se frustre y entienda de forma simple lo que ha sucedido. Esto no lo hemos aplicado en

nuestra aplicación, ya que está más orientado para webs y nuestra app es para dispositivos móviles.

• **Ayuda y documentación:** Por último, existe un principio que no hemos implementado en nuestro proyecto debido a la falta de tiempo y a que priorizamos otros aspectos que considerábamos más cruciales en esta fase. Sin embargo, es importante tenerlo en cuenta para futuras mejoras. Este principio sugiere crear un botón de ayuda, permitiendo que el usuario pueda contactar con los programadores de manera sencilla si necesita asistencia.

También es importante incluir documentación de cómo funciona la aplicación por si el usuario quiere hacer uso de ella. Este principio sí lo aplicamos, ya que en el GitHub tenemos un *readme* en el que explicamos el funcionamiento de la aplicación paso a paso.

5.3 Estructura del proyecto.

Es muy conveniente que, antes de comenzar a programar, se planifique cómo se va a estructurar el proyecto. En la Ilustración 46 observamos la estructura general del proyecto de **HurbValencia** y a continuación la explicaremos detalladamente.

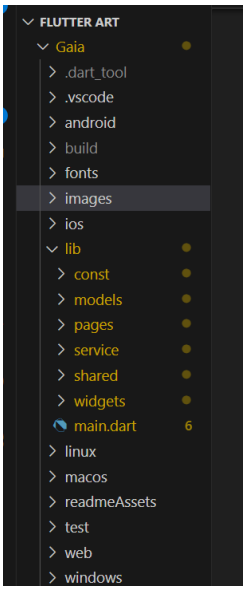


ILUSTRACIÓN 46: ESTRUCTURA DEL CÓDIGO DE HURBVALENCIA

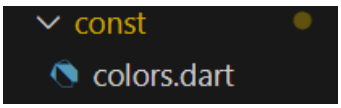


ILUSTRACIÓN 47: ESTRUCTURA DEL CÓDIGO: CONST

En *const* (Ilustración 47) guardamos las constantes como es el caso de los colores ya vistos, donde se encuentran las variables de los colores que usa la aplicación.

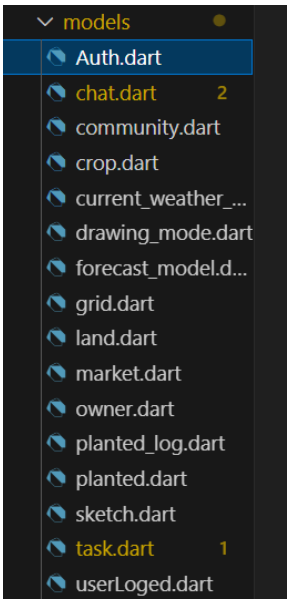


ILUSTRACIÓN 48: ESTRUCTURA DEL CÓDIGO: MODELS

La carpeta *models* (Ilustración 48) contiene las definiciones de clases que representan los datos y estructuras que maneja la aplicación. Por ejemplo, el auth.dart sirve para manejar la autenticación y la información del usuario autenticado. Otro ejemplo podría ser el task.dart, que sirve para manejar y estructurar la información relacionada con las tareas del toDo.

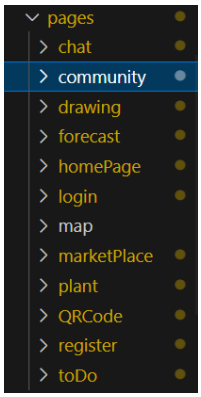


ILUSTRACIÓN 49: ESTRUCTURA DEL CÓDIGO PAGES

Esta carpeta (Ilustración 49) contiene las distintas páginas de la aplicación.

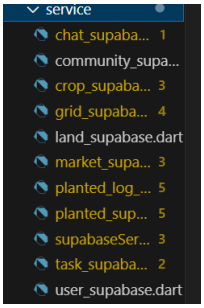


ILUSTRACIÓN 50: ESTRUCTURA DEL CÓDIGO SERVICE

La carpeta de los servicios (Ilustración 50) sirve para definir y organizar la comunicación con la base de datos y *backend* de SupaBase.

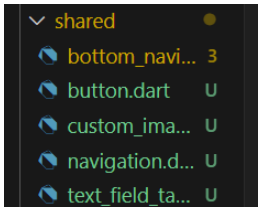


ILUSTRACIÓN 51: ESTRUCTURA DEL CÓDIGO: SHARED

La carpeta *shared* (Ilustración 51) sirve para mostrar los *widget*s que serán compartidos por todas o la mayoría de las páginas, como es el caso de la ya vista *bottom navigation bar*.

ILUSTRACIÓN 52: ESTRUCTURA DEL CÓDIGO *WIDGETS*

La carpeta *widgets* se utiliza para organizar componentes reutilizables de la interfaz de usuario. Cuando un elemento se repite varias veces en diferentes páginas de la aplicación, lo encapsulamos en un widget personalizado y lo colocamos en esta carpeta.

5.3.1 Diagrama de clases

Para analizar el dominio de aplicación, y representar la parte estática o de estructura de aplicación, hemos realizado un diagrama de clase UML (Lenguaje Unificado de Modelado) [21], representando las clases con sus atributos y métodos, y las relaciones entre ellas. En la Ilustración 53 podemos observar el diagrama de clases de nuestra aplicación ya como tablas relacionales. Un breve resumen de lo que plasma cada clase sería el siguiente:

Tabla *task* representaría las tareas para el toDo/agenda, el *marketplace* a un item del mercado, los *chatRooms* a un chat con sus mensajes del cliente y el vendedor, el *planted* log al historial de las plantas que se han cultivado a lo largo del tiempo, el *planted* y el *crop* están relacionados también, donde el *planted* registra una plantación actual y el *crop* a una planta con sus distintos atributos. La *land* describe al terreno de una comunidad con su ubicación y distancia, entre otras. El *grid* representa a la cuadrícula para la comunidad

con sus dimensiones y la distribución, el *user* al usuario *logeado* y la comunidad a la comunidad de huertos urbanos.

ILUSTRACIÓN 53: DIAGRAMA DE CLASES COMO TABLAS RELACIONALES

6. Pruebas de Aceptación

En este capítulo, se presentan las pruebas de validación [22]. Este proceso se sigue generalmente en el ámbito del software, donde se evalúa una aplicación en desarrollo o finalizada. Estas pruebas son fundamentales para garantizar que nuestra aplicación cumpla con los requisitos y expectativas establecidos, cara a los usuarios finales.

6.1 Tipos de pruebas empleadas

A continuación, vamos a mostrar algunos ejemplos de pruebas de aceptación realizadas en nuestra app.

• **Pruebas de navegación:** Comprueban que la navegación entre las distintas pantallas funcione de forma correcta.

• **Pruebas de interfaz de usuario:** Verifican que todos los elementos se muestren de una forma correcta en pantalla.

• **Pruebas de estado:** Demuestran que los estados de la aplicación actúen de manera adecuada. Por ejemplo, al crear una tarea en la agenda, se debe mostrar de forma dinámica y casi instantánea.

• **Pruebas de rendimiento:** Se confirma que la aplicación sea escalable y tenga una rapidez de carga adecuada a lo esperado.

6.2 Escenarios de uso

Para verificar la mayoría de estas pruebas, realizaremos diversos escenarios de uso en los que un usuario interactuará con diferentes partes de nuestra aplicación para lograr un objetivo específico [23].

• El primer escenario será un usuario nuevo llamado Paco que va a querer unirse a una comunidad de huerto urbano a la que pertenece su amigo Moncho y creará su primera tarea de alta prioridad que le servirá para acordarse de que quiere comenzar en esta afición plantando naranjas y, por último, registrará las naranjas en la lista de plantas que quiere cultivar.

En primer lugar, Moncho iniciará sesión y accederá la a parte del QR de su comunidad para copiar la id y mandársela a Paco.

ILUSTRACIÓN 54: UNIRSE A LA COMUNIDAD - QR

Posteriormente, Paco podrá clicar sobre el botón azul que pone *Register* (en la Ilustración 55) ya que actualmente Paco no tiene cuenta. A continuación, sin equivocarse en el formato del email ni de la contraseña, Paco rellenará todo con sus datos personales. En caso de error, como es el caso de la Ilustración 56, le saltará un error de validación que le avisará para que pueda registrarse exitosamente. Como vemos en la Ilustración 57 Paco será notificado finalmente de su correcto registro.

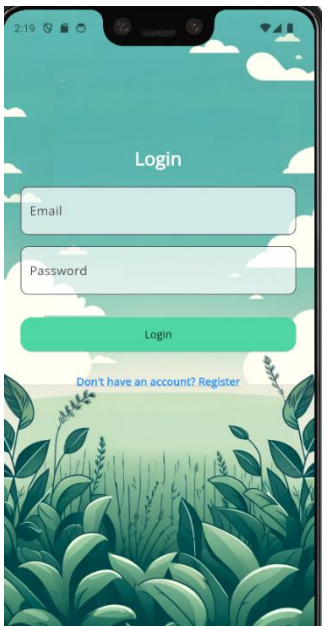


ILUSTRACIÓN 55: CASO DE USO: INICIAR SESIÓN

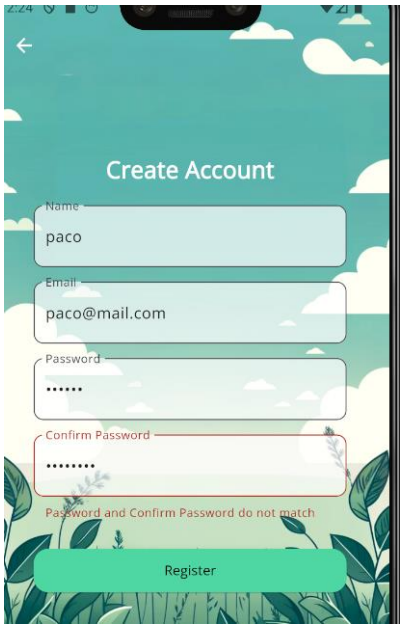


ILUSTRACIÓN 56: CASO DE USO: REGISTRARSE

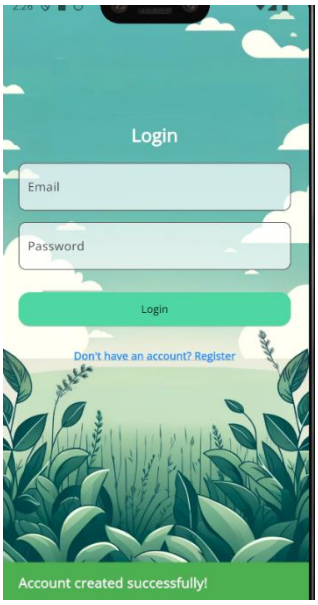


ILUSTRACIÓN 57: CASO DE USO: CONFIRMACIÓN REGISTRO

Al escribir correctamente los datos para iniciar sesión, Paco al ser un nuevo usuario accederá a la siguiente ventana, donde tendrá que elegir si quiere crear una comunidad o si va a unirse a una comunidad ya existente. Su caso es el de unirse (Ilustración 59) a una ya creada, por lo que, tras darle clic, tendrá que simplemente que introducir el id de comunidad que le ha enviado su amigo Moncho.

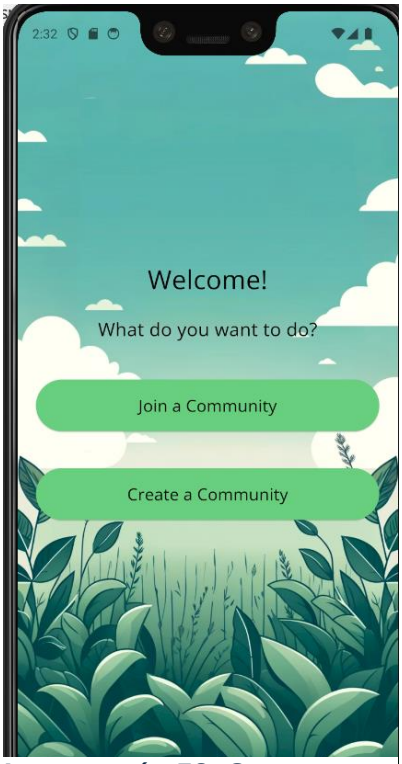


ILUSTRACIÓN 58: CASO DE USO: CREAR O UNIRSE COMUNIDAD

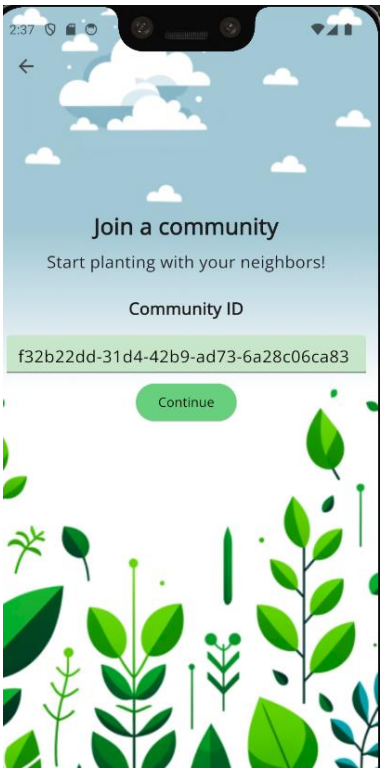


ILUSTRACIÓN 59: CASO DE USO: UNIRSE COMUNIDAD

Tras haber insertado de forma correcta el id de la comunidad Paco accederá a la página principal (ilustración 60). A continuación, tendrá que clicar sobre el icono de las tareas en la barra de navegación para acceder a la página de la agenda (ver Ilustración 61).

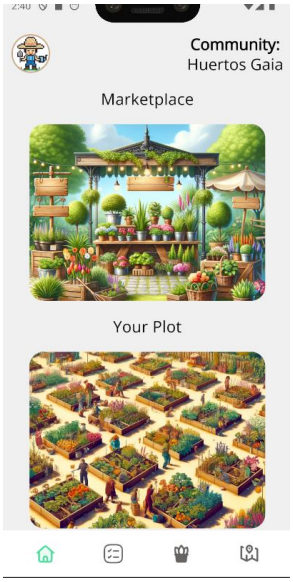


ILUSTRACIÓN 60: CASO DE USO: PÁGINA PRINCIPAL

Ahora Paco creará su primera tarea (Ilustración 62) y cuando la haya acabado solo tendrá que clicar sobre la *checkbox* y esta pasará de estar en "*To Do*" a "*Done*" como vemos en la Ilustración 63.

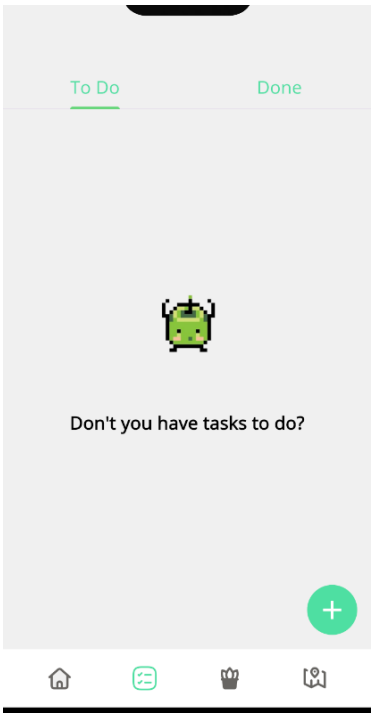


ILUSTRACIÓN 61: CASO DE USO: SIN TAREAS

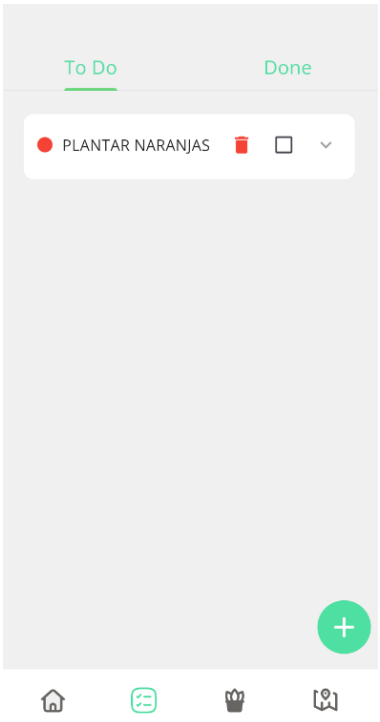


ILUSTRACIÓN 62: CASO DE USO: TAREAS EN TO DO

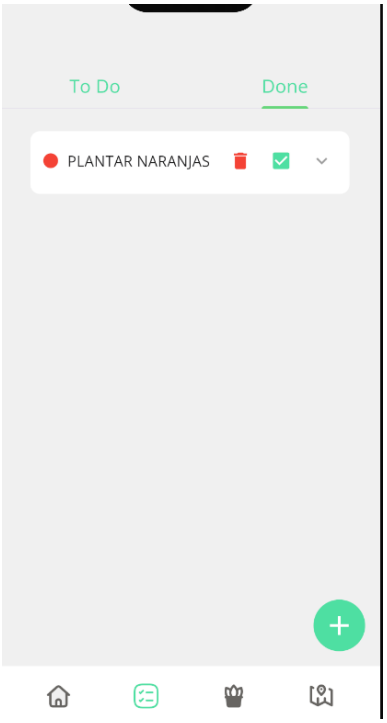


ILUSTRACIÓN 63: CASO DE USO: TAREAS EN DONE

A continuación, Paco procederá a buscar las naranjas en la sección de plantas y registrarlas, ya sea como favoritas o como plantadas. De este modo, podrá acceder fácilmente a la información sobre cómo cultivar estas plantas. Para ello, simplemente deberá hacer clic en el icono de la planta en la barra de navegación (Ilustración 64) y luego al darle clic en el botón de "+" buscará el cultivo deseado en la barra de búsqueda (Ilustración 65), y una vez seleccionada la planta que se va a añadir, aparecerá información detallada sobre esta (Ilustración 66).

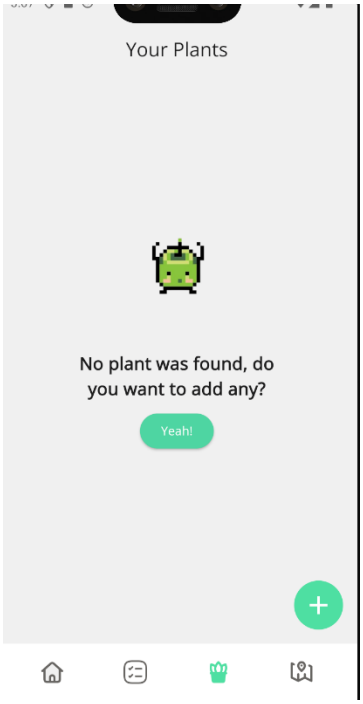


ILUSTRACIÓN 64: CASO DE USO: SIN PLANTAS AÑADIDAS

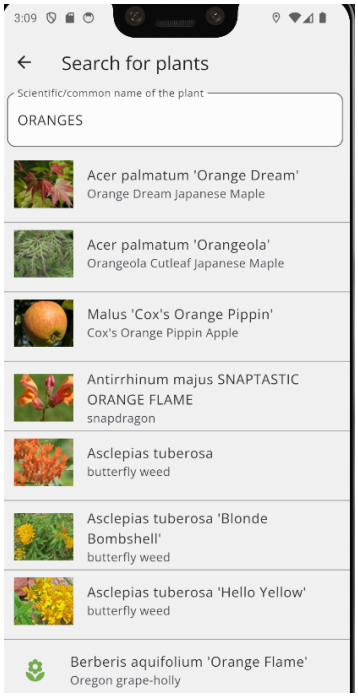


ILUSTRACIÓN 65: CASO DE USO: BUSCANDO PLANTAS

ILUSTRACIÓN 66: CASO DE USO: INFORMACIÓN PLANTA

Por último, Paco solamente tendrá que darle al botón de + para añadirla y le saldrá el “pop up” de la Ilustración 67 donde al darle clic sobre “*yes*” se añadirá la planta a nuestra lista de cultivos como vemos en la Ilustración 68. Hay que tener en cuenta, que, de forma intuitiva, en caso de querer eliminar la tarea sería tan lógico como clicar sobre la X de la Ilustración 68.

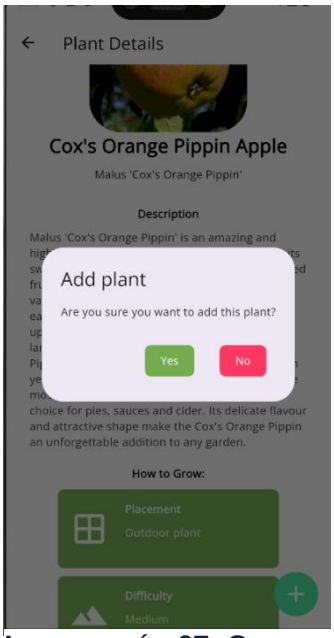


ILUSTRACIÓN 67: CASO DE USO: AÑADIR PLANTA

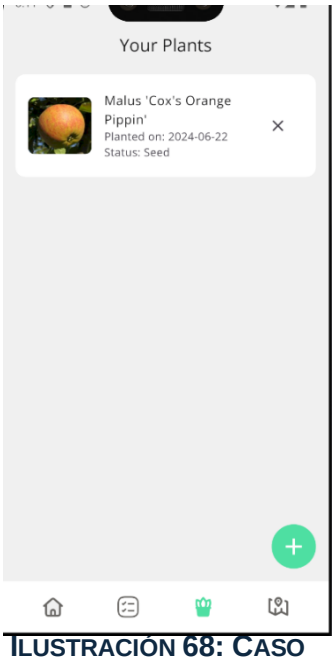


ILUSTRACIÓN 68: CASO DE USO: PLANTA AÑADIDA

De esta forma tan simple, lógica e intuitiva Paco habría conseguido su objetivo con éxito.

• En el segundo escenario de uso, a Paco le habrán crecido las naranjas y querrá intercambiar algunas con sus compañeros de comunidad. Para ello, creará una oferta.

En primer lugar, para acceder al mercado de intercambios, Paco tendrá que clicar sobre Marketplace (Ilustración 69). En dicha página como vemos en la ilustración 70, observaremos las distintas ofertas que han publicado otros miembros de la comunidad. Si Paco estuviese interesado en alguna oferta, clicaría sobre el ícono de chat y enviaría un mensaje como se puede apreciar en la Ilustración 71.

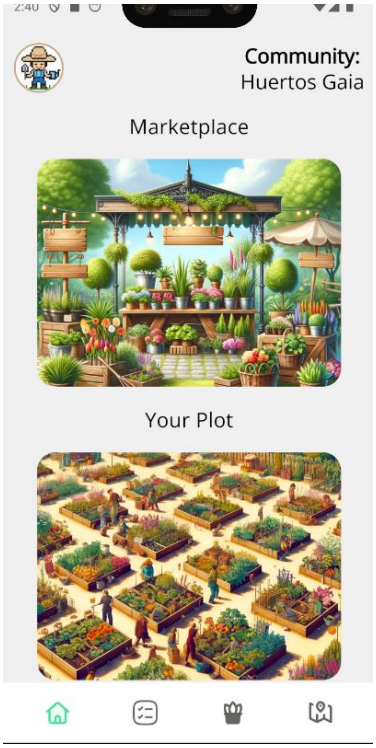


ILUSTRACIÓN 69: CASO DE USO: VUELTA A LA PRINCIPAL

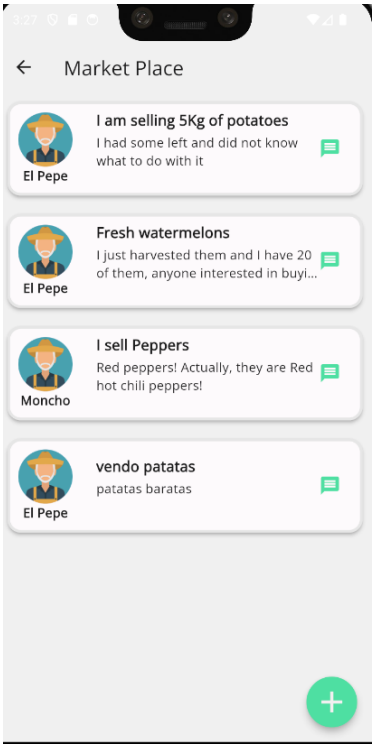


ILUSTRACIÓN 70:CASO DE USO: MERCADO COMUNITARIO

ILUSTRACIÓN 71: CASO DE USO: CHAT CON EL PEPE

En este caso Paco va a crear su propia oferta. Por lo tanto, siguiendo el mismo patrón que para crear un Tarea, clicará sobre la figura ovalada + y, como vemos en la Ilustración 72, le saltará un “pop up” para crearla. Al crear su publicación, aparecerá instantáneamente de forma dinámica (Ilustración 73) y Paco podrá clicar sobre el icono de chat que sale sobre su post para acceder a ver los chats de las personas que le han hablado sobre su oferta. En este caso como vemos en la Ilustración 74 está vacío, ya que todavía nadie le ha respondido.

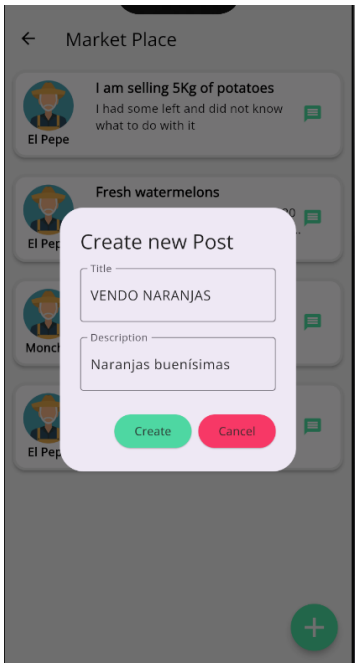


ILUSTRACIÓN 72: CASO DE USO: CREAR OFERTA

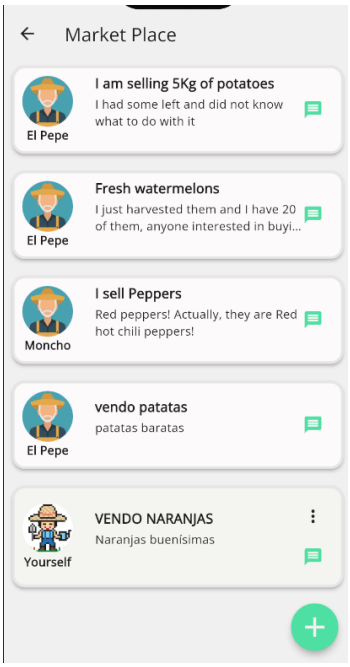


ILUSTRACIÓN 73:CASO DE USO: OFERTAS CREADAS

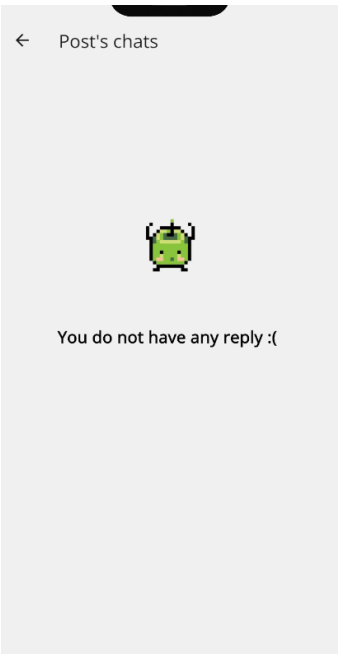


ILUSTRACIÓN 74: CASO DE USO: SIN CHATS EN EL POST

De esta forma, Paco habría creado su publicación con éxito y ahora solo tendría que esperar para negociar con las personas que le contesten.

7. Conclusión y trabajo futuro

En este último apartado comentaremos el futuro de **HurbValencia**, las conclusiones obtenidas del desarrollo del proyecto, los trabajos futuros que quedan planteados, y finalmente, y el valor concreto de algunas de sus asignaturas para la consecución de este TFG

7.1 Conclusiones

En esta memoria se ha realizado un análisis del progreso de desarrollo de la *app* **HurbValencia**, incluyendo algunos aspectos como las tecnologías y arquitectura empleada, la metodología usada y un estudio de mercado que nos permite descifrar nuestra posición respecto a la competencia y nuestras oportunidades en el mercado.

En este proyecto he tenido el papel del desarrollo completo de la interfaz. Mi tarea se puede presenciar en cada una de las ventanas realizadas, ya que para cada una de ellas siempre hay que crear una interfaz y mi labor ha sido crucial para el éxito del proyecto. Como hemos mostrado a lo largo del TFG no tiene sentido que realicemos una aplicación sin una buena interfaz. En ese caso, nadie la usaría y se escogería a la de la competencia.

Por último, en cuanto al aspecto económico tenemos que destacar lo siguiente:

- En la feria de proyectos de la UPV donde mostramos **HurbValencia** a distintas personas y empresas que asistieron, se nos ofreció la oportunidad de entrar en Lanzadera, empresa de Juan Roig Alfonso, que se dedica a impulsar pequeñas empresas (*startups*). Rechazamos esta oferta ya que dos de los miembros del equipo (Gabriel y Laijie) son de segundo de carrera y, para estar en Lanzadera, no puedes estar cursando la carrera. Además, los otros dos miembros (Luis Navarro y Daniel Sanz), que estamos en cuarto de carrera, no estamos muy interesados en dicha oferta, pues no tenemos la suficiente seguridad como para considerarnos expertos y sacar un buen proyecto con el que ganar dinero.
- Por otro lado, como ya he mencionado anteriormente, hay un pilar del programa *Horizont Europe* de la Unión Europea al que este proyecto podría optar en

concurrencia competitiva con la marca UPV para poder seguir desarrollándolo, ya que el proyecto cumple los requisitos para ganarlo.

En conclusión, la combinación de una sólida formación académica y la aplicación práctica de nuevos conocimientos ha sido clave para el desarrollo exitoso de **HurbValencia**, posicionándonos de manera competitiva en el mercado.

7.2 Trabajo futuro

A lo largo de la memoria hemos detallado varios aspectos que nos habría gustado haber completado al finalizar el proyecto, pero que no pudimos abordar debido a limitaciones de tiempo.

- En cuanto a los detalles de implementación, observando el *backlog* del proyecto en la tabla 2, podemos notar que la parte del reconocimiento de plantas mediante Inteligencia Artificial quedó pendiente. Como mencioné en el capítulo 4, *sprint* 3, esta parte se complicó debido a la falta de tiempo que requería desarrollar una Inteligencia Artificial eficiente desde cero o a la inversión monetaria de adquirir una existente.
- Otro aspecto que destacar es la funcionalidad de permitir que un usuario se una a múltiples comunidades de huertos urbanos o se retire de ellas. Actualmente, un usuario solo puede unirse a una comunidad y no tiene la opción de salir de ella. Por lo tanto, si desea cambiar de comunidad, tendría que crear una nueva cuenta. Además, si el usuario participa en varias comunidades de huertos urbanos, la aplicación solo le permitirá estar en una a la vez.
- Por último, en cuanto a los detalles del desarrollo, también nos habría gustado implementar un rol de Invitado. Esto permitiría a cualquier usuario acceder a la aplicación sin necesidad de crear una cuenta, facilitando la visualización de huertos urbanos cercanos mediante el mapa. Actualmente, para ver el mapa es necesario tener una cuenta.

7.3 Relación con las asignaturas cursadas

A lo largo de este proyecto hemos podido constatar que hay asignaturas cursadas en el grado que, en su momento podían parecernos poco importantes y a la hora de hacer un proyecto como este, nos han servido de mucho. Entre ellas destacaría Lenguajes Teoría y Paradigmas de la Programación, donde aprendimos las bases de la programación, Base de Datos, donde se nos enseñó como actuar con Bases de datos Relacionales y, por último, Diseño Centrado en el Usuario, donde adquirí los conceptos de lo que es una buena interfaz. Además, el conjunto de todos los conocimientos adquiridos durante la carrera ha contribuido a que nuestro aprendizaje sea extremadamente rápido, y ello a pesar de que Flutter sea una nueva tecnología no vista en la carrera.

8 Referencias

[1] Artemis Club. Consultada el 20/06/2024. Disponible en: https://artemisclub.es/

[2] Agile Alliance(2001)". "Manifesto for Agile Software Development".Consultada el 20/06/2024. Disponible en: https://agilemanifesto.org/

[3]Scrum oficial. Schwaber, K., & Sutherland, J. (2020). "The Scrum Guide". Scrum.org. Consultada el 20/06/2024. Disponible en: https://scrumguides.org/scrum-guide.html

[4] "¿Qué es el Lean Canvas?". CECARM. Consultada el 28/04/2024. Disponible en: https://www.cecarm.com/emprendedor/estrategia/consultas-y-faqs/que-es-el-lean-canvas-3801

[5] Google Play Store. "MacetoHuerto". Consultada el 29/04/2024. Disponible en: https://play.google.com/store/apps/details?id=com.centelles.josep.macetohuerto&hl=es_419&pli=1

[6] Google Play Store. "Planificador de Jardines". Consultada el 29/04/2024. Disponible en: https://play.google.com/store/apps/details?id=com.bentosoftware.gartenplaner&hl=es_419

[7] Google Play Store. "Garden Planner". Consultada el 29/04/2024. Disponible en: https://play.google.com/store/apps/details?id=com.perculacreative.peter.gardenplanner&hl=es

[8] Google Play Store. "Picture This". Consultada el 29/04/2024. Disponible en: https://play.google.com/store/apps/details?id=cn.danatech.xingseus&hl=es

[9] Wikipedia. "Análisis DAFO". Consultada el 29/04/2024. Disponible en: https://es.wikipedia.org/wiki/An%C3%A1lisis_FODA

[10] Infoautónomos. "Análisis DAFO". Consultada el 29/04/2024. Disponible en: https://www.infoautonomos.com/plan-de-negocio/analisis-dafo/

[11]Modelos de negocio. Consultada el 29/04/2024. Disponible en: Osterwalder, A., & Pigneur, Y. (2010). Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers. Wiley.

[12] "Flutter ". Flutter.dev. Consultada el 17/05/2024. Disponible en: https://flutter.dev/

[13] Samsung Developer Program. "¿Por qué elegir Flutter como framework de desarrollo móvil?". Consultada el 17/05/2024. Disponible en: https://www.europe-samsung.com/smsdev/Noticias/Detalle/por_que_elegir_flutter_como_framework_de_desarrollo_movil/a29bff94-e2b7-4ef3-a28e-e566b752cfb9

[14] "Supabase - The Open Source Firebase Alternative". Supabase. Consultada el 18/05/2024. Disponible en: https://supabase.io/

[15] "Android Studio". Android Developers. Consultada el 18/05/2024. Disponible en: https://developer.android.com/studio

[16] WeatherAPI. Consultada el 18/05/2024 https://www.weatherapi.com/

[17] Perenual API. Consultada el 18/05/2024 https://perenual.com/docs/api

[18] MapTiler. Consultada el 18/05/2024 https://www.maptiler.com/

[19] Digital Var. "10 Principios de Nielsen" Consultada el 18/06/2024 https://digitalvar.es/articulos-diseno-web/10-principios-de-nielsen/

[20] Nielsen Norman Group. "10 Usability Heuristics for User Interface Design". Consultada el 18/06/2024. Disponible en: https://www.nngroup.com/articles/ten-usability-heuristics/

[21] Diagrama de clases. Consultada 20/06/2024 disponible en: Schneider, M., & Winter, R. (2013). UML@Classroom: An Introduction to Object-Oriented Modeling. Springer.

[22] Wikipedia. "Pruebas de validación". Consultada el 20/06/2024

https://es.wikipedia.org/wiki/Pruebas_de_validaci%C3%B3n

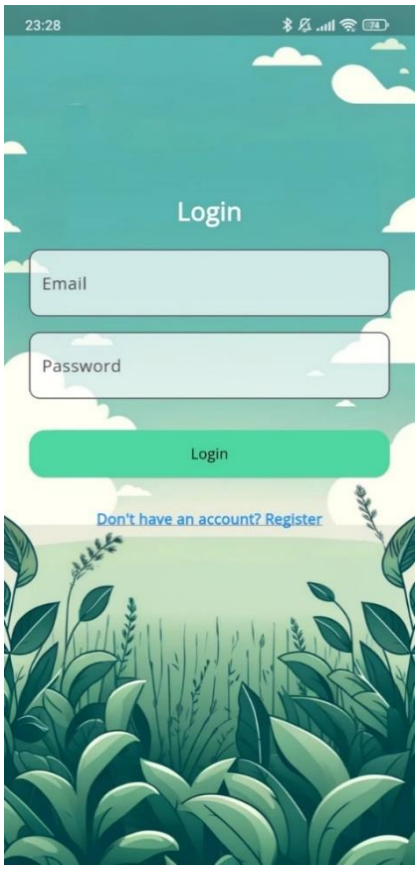
[23] Wikipedia. "Caso de uso". Consultada el 20/06/2024. Disponible en: https://es.wikipedia.org/wiki/Caso_de_uso

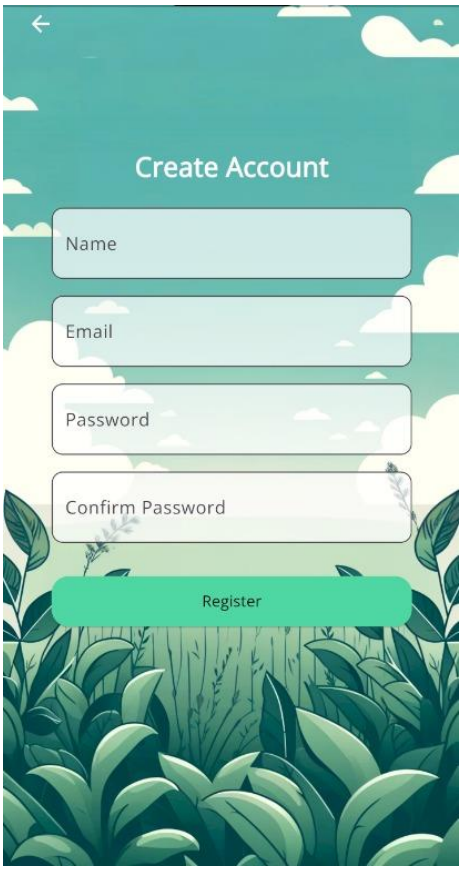
9 Anexos

9.1 Aplicación Final

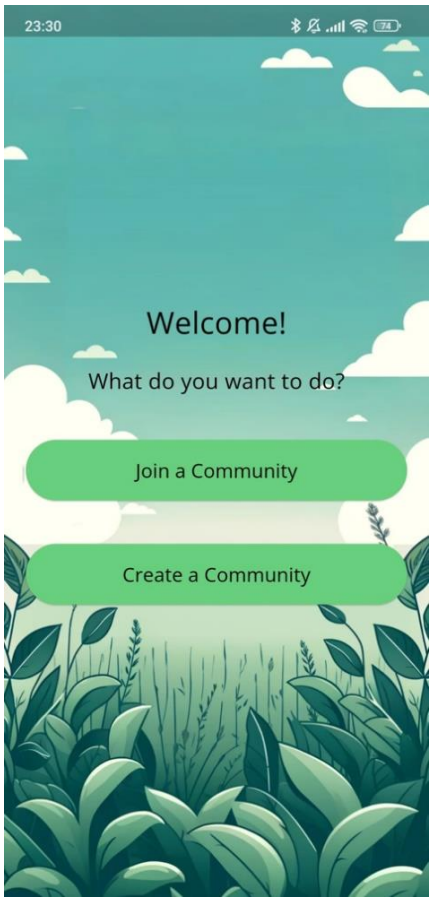
En este punto se mostrarán todas las pantallas de la aplicación final de HurbValencia.

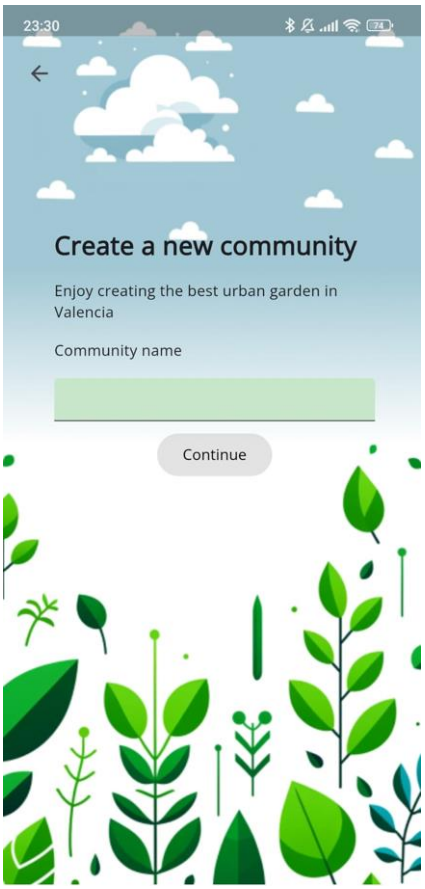
• Pantallas iniciales para Registrarse e iniciar sesión.

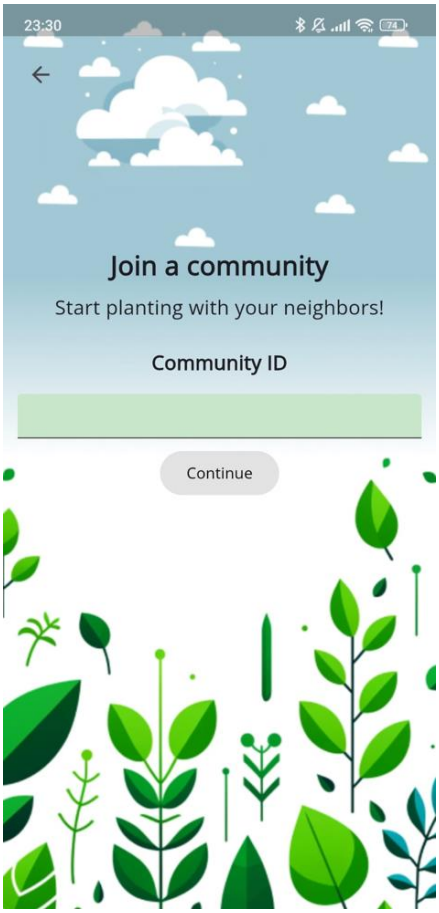




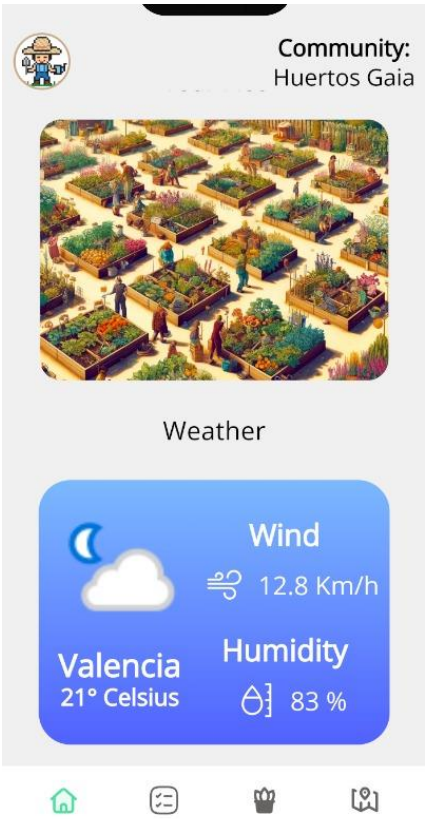
• Pantallas para crear y unirse a una comunidad tras iniciar sesión.

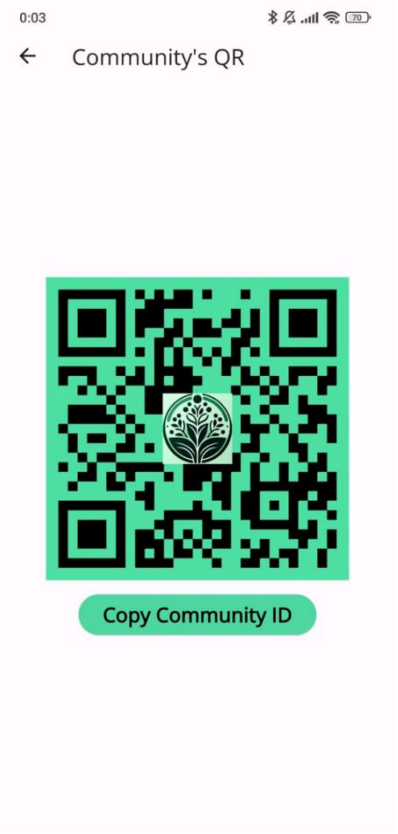




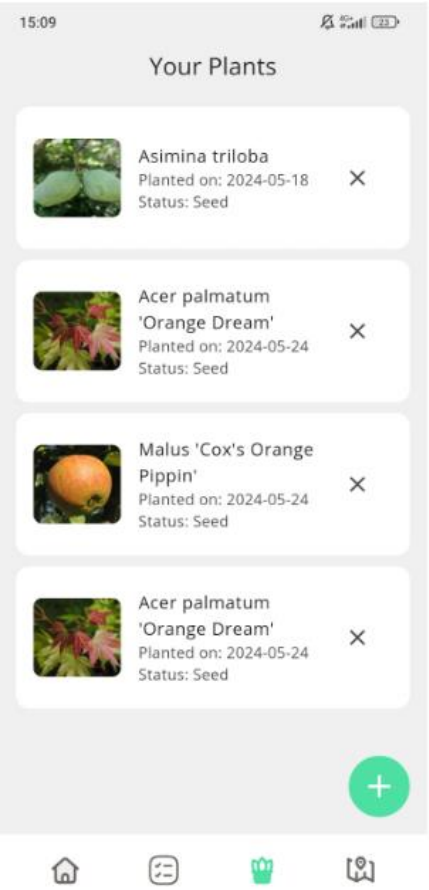


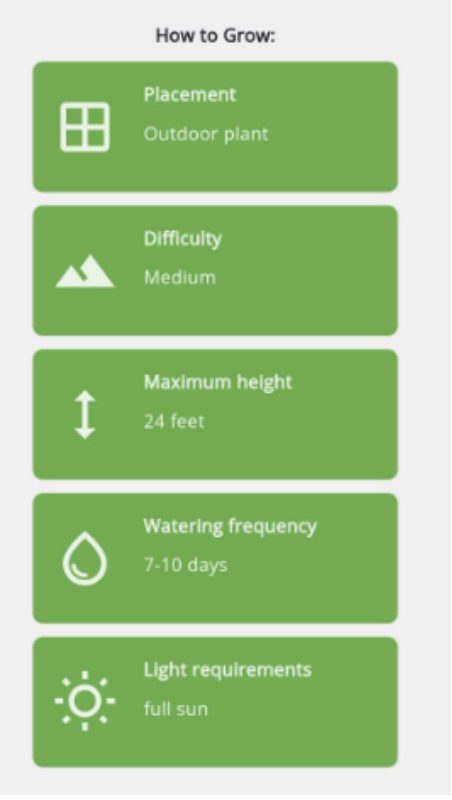
• Pantalla principal y ventana para compartir el QR de tu comunidad



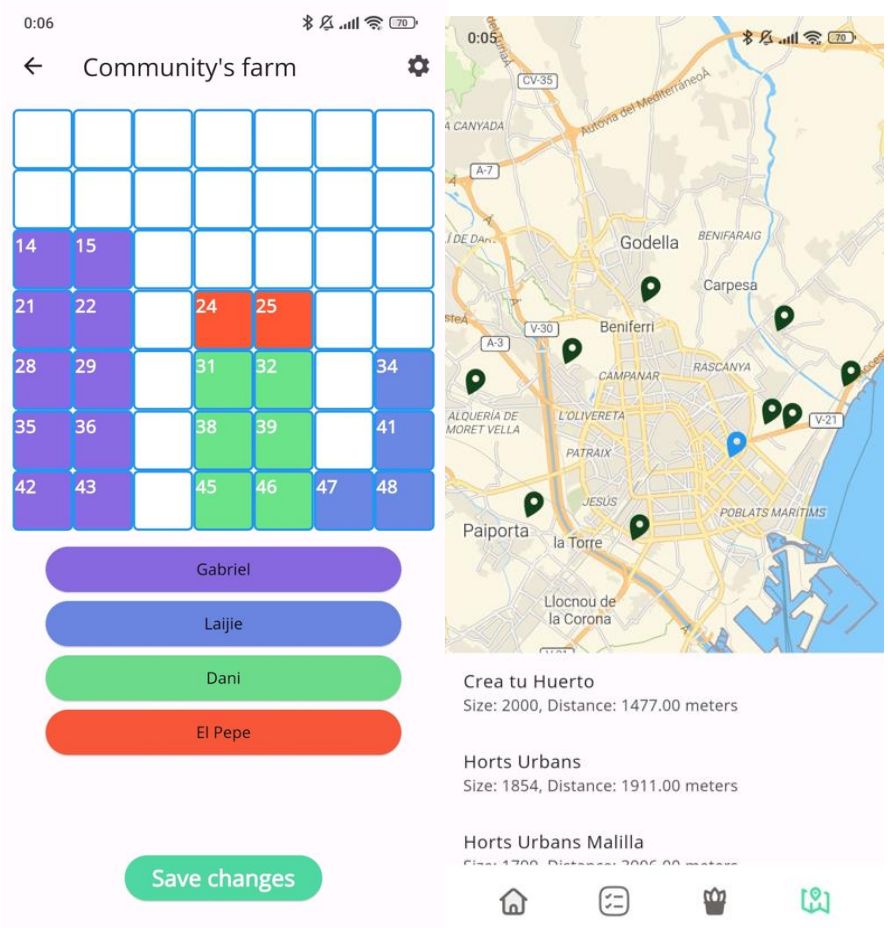


• Pantallas para añadir plantas a tu lista de favoritas/plantadas y buscar plantas con su información

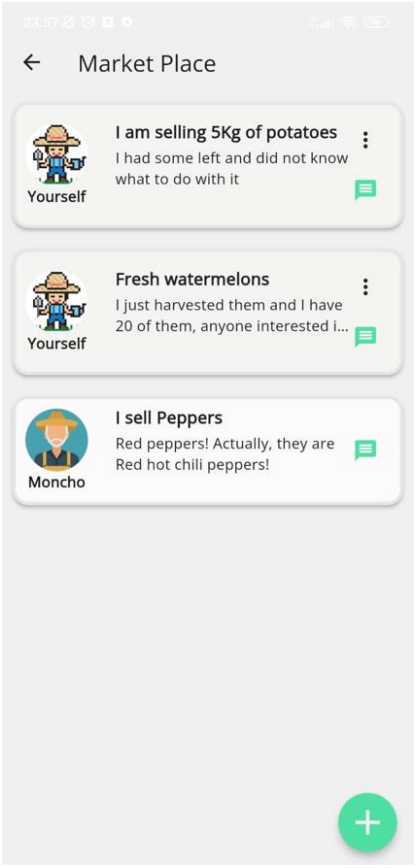


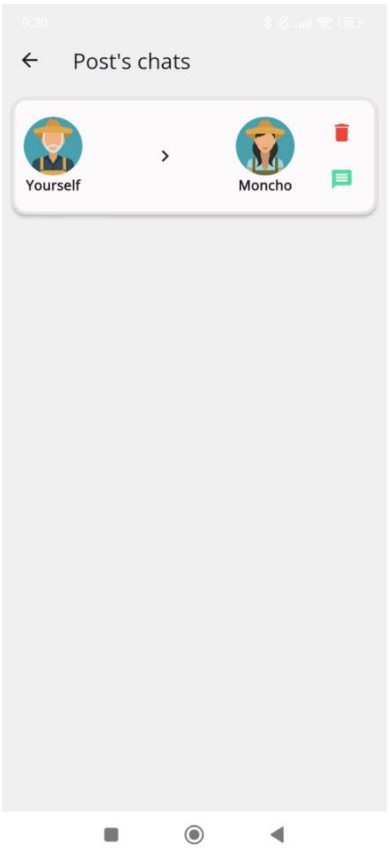


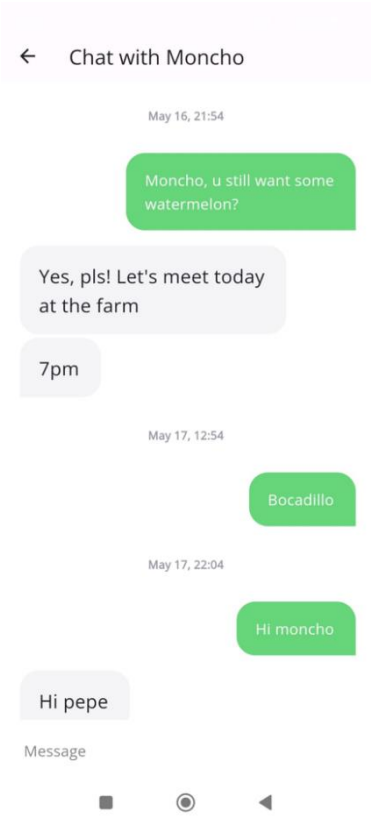
• Ventana de la matriz/*grid* para gestionar las parcelas de la comunidad y Mapa para visualizar las comunidades de huertos urbanos más cercanas.



• Páginas del mercado para crear tus ofertas de intercambio y el chat para comunicarte con los vecinos de la comunidad que te hayan contactado.







9.2 (ODS)Objetivos de desarrollo sostenible

TABLA 3: TABLA ODS

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
A. **Fin de la pobreza.**			**X**	
B. **Hambre cero.**		**X**		
C. **Salud y bienestar.**	**X**			
D. **Educación de calidad.**				**X**
E. **Igualdad de género.**				X
F. **Agua limpia y saneamiento.**				**X**
G. **Energía asequible y no contaminante.**			**X**	
H. **Trabajo decente y crecimiento económico.**				**X**
I. **Industria, innovación e infraestructuras.**				**X**
J. **Reducción de las desigualdades**.				**X**
K. **Ciudades y comunidades sostenibles**.	**X**			
L. **Producción y consumo responsables.**				**X**
M. **Acción por el clima.**		X		
N. **Vida submarina.**				**X**
O. **Vida de ecosistemas terrestres.**				**X**
P. **Paz, justicia e instituciones sólidas.**			**X**	
Q. **Alianzas para lograr objetivos.**				**X**

• Fin de la pobreza, hambre cero, salud y bienestar:

Nuestra aplicación fomenta el cultivo de distintas plantas alimenticias que harán que no tengas que gastar dinero en dichos productos indispensables para una buena salud. Esto mismo sería también muy útil para el hambre cero ya que te comerías tus propios productos e incluso si haces de más, como ya hemos explicado anteriormente, podrías intercambiarlos con distintos miembros de tu comunidad.

• Acción por el clima,energía asequible y no contaminante:

Al impulsar a las distintas personas a que creen su propio huerto urbano logramos que estas no dependan al 100% de algunos productos de los supermercados, que al ser importados desde otros países suponen el gasto de una gran cantidad de energía para que lleguen a su destino final.

• Ciudades y comunidades sostenibles:

Como he comentado a lo largo de todo este trabajo de fin de grado, el objetivo principal era contribuir a convertir a Valencia en una Smart City. Creemos que con nuestra aplicación ayudamos a su desarrollo y hacerla más sostenible.