



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**título del TFG
Documentación Técnica**



Presentado por Luis Rojo Sanz
en Universidad de Burgos — 22 de mayo
de 2025

Tutor: Dr. Raúl Marticorena Sánchez
D. Yeray Pescador Calleja (Técnico de
Emprendimiento UBU)

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	iv
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	4
Apéndice B Especificación de Requisitos	11
B.1. Introducción	11
B.2. Objetivos generales	11
B.3. Catálogo de requisitos	12
B.4. Especificación de requisitos	14
Apéndice C Especificación de diseño	37
C.1. Introducción	37
C.2. Diseño de datos	37
C.3. Diseño arquitectónico	38
C.4. Diseño procedimental	40
Apéndice D Documentación técnica de programación	43
D.1. Introducción	43
D.2. Estructura de directorios	43
D.3. Manual del programador	45

D.4. Compilación, instalación y ejecución del proyecto	54
D.5. Pruebas del sistema	57
Apéndice E Documentación de usuario	61
E.1. Introducción	61
E.2. Requisitos de usuarios	61
E.3. Instalación	61
E.4. Manual del usuario	61
Apéndice F Anexo de sostenibilización curricular	63
F.1. Introducción	63
Bibliografía	65

Índice de figuras

C.1. Diseño de Datos	38
D.1. Primer paso de la instalación de Android Studio.	46
D.2. Segundo paso de la instalación de Android Studio.	47
D.3. Ruta de la instalación	48
D.4. Instalación plugins Android Studio	48
D.5. Configuración de SDK en Android Studio	49
D.6. Instalación de Herramientas de SDK	49
D.7. Descarga del archivo.	50
D.8. Pantalla de configuración avanzada.	51
D.9. Pantalla de variables de entorno.	52
D.10. Pantalla de editar variables de entorno.	53
D.11. Uso del comando “ <i>flutter doctor</i> ”	53
D.12. Descarga de un emulador	54
D.13. Seleccionando un emulador	55
D.14. Seleccionando la versión de Android del emulador	55
D.15. Verificar la configuración del emulador	56
D.16. Ejecución del emulador	56
D.17. Ejecución del proyecto	57

Índice de tablas

A.1. Costes hardware prorrateados en 8 meses	5
A.2. Costes Desarrollador	6
A.3. Coste Total	6
A.4. Costes Hardware Comercialización	7
A.5. Costes Desarrollador	7
A.6. Costes Desarrollador	8
A.7. Coste 6 meses Comercialización	8
B.1. CU-1 Creación de Plantas.	15
B.2. CU-2 Editar Plantas.	16
B.3. CU-3 Eliminación de Plantas.	17
B.4. CU-4 Estado de los sensores.	18
B.5. CU-5 Cambio de Idioma.	19
B.6. CU-6 Visualización de hitos mensuales.	20
B.7. CU-7 Visualización de hitos mensuales.	21
B.8. CU-8 Cambio de Tierra.	22
B.9. CU-9 Gráfica de Humedad.	23
B.10.CU-10 Gráfica de Humedad de Ambiente.	24
B.11.CU-11 Gráfica de Luz.	25
B.12.CU-12 Gráfica de Temperatura.	26
B.13.CU-13 Histórico de Valores en Gráficos.	27
B.14.CU-14 Consejos Plantas.	28
B.15.CU-15 Personalización en la Visualización de los Gráficos.	29
B.16.CU-16 Personalización en la Visualización de los Hitos.	30
B.17.CU-17 Personalización de la fecha del cambio de tierra.	31
B.18.CU-18 Personalización de la fecha para agregar fertilizante.	32
B.19.CU-19 Personalización de la imagen de la planta.	33
B.20.CU-20 Visualización del cuidado de la planta.	34

<i>Índice de tablas</i>	v
B.21.CU-21 Mensajes de Error.	35

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En este apéndice se desarrolla el plan de proyecto seguido para la creación de la aplicación de GreenInHouse 2.0, tratando la planificación temporal y el análisis de viabilidad del proyecto. En la planificación temporal se detalla la organización que ha seguido el proyecto con los diferentes sprints en los que se ha dividido y en lo desarrollado en cada uno de ellos. Por otro lado, también se ha llevado a cabo un estudio de la viabilidad para analizar los factores que han influido en el desarrollo de la aplicación. Se ha estudiado la viabilidad económica, evaluando los recursos necesarios, los costos y los beneficios económicos potenciales, junto con la viabilidad legal asegurando que cumple con la normativa vigente. Este plan de proyecto nos permite garantizar que aunque el proyecto sea posible a nivel técnico, también lo tiene que ser a nivel económico y legal.

A.2. Planificación temporal

La planificación temporal de este proyecto se ha basado en la realización de un sprint cada dos semanas reuniéndome con el tutor del proyecto Raúl Marticorena, con Yeray Pescador y con Oscar Valverde para ver los cambios realizadas durante ese tiempo, viendo si dichos cambios estaban bien implementados y proponiendo nuevas tareas a realizar.

Sprint 0 Planificación y Diseño Inicial 10/10/2024 - 24/10/2024

Durante el primer sprint se llevó a cabo la planificación y un pequeño diseño de la aplicación. El diseño lo realicé creando un *mockup* de las principales pantallas que tendría la aplicación como son la de inicio, la de gráficos, la de hitos, la de creación de plantas y la de ajustes. Esto lo llevé a cabo mediante el uso de una herramienta llamada “Pencil”. Se investigaron las herramientas que iba a usar en la creación de la aplicación midiendo entre otras cosas, la compatibilidad con el uso de APIs para poder consultar los datos que hay guardados en la base de datos del backend. Llegamos a la conclusión que la mejores herramientas para este fin era usar Flutter y Dart integradas ya en Android Studio. Por último instalé Android Studio descargando Flutter, Dart y diferentes librerías necesarias ya que si no me daban ciertos errores durante la instalación.

Sprint 1 Desarrollo Inicial de Pantallas 24/10/2024 - 07/11/2024

Durante este segundo sprint comencé con el desarrollo de la aplicación creando las primeras pantallas. Implementé la pantalla de inicio, en la que añadí un panel inferior donde más adelante iría implementando las referencias a otras pantallas. También creé una primera versión de la pantalla de creación de plantas aunque todavía no era del todo funcional. En ambas pantallas incorporé la función de internacionalización. Por último conseguí que la integración de la API de la raspberry con el Android Studio fuera funcional.

Sprint 2 Creación y Mejora de Pantallas 07/11/2024 - 21/11/2024

Este tercer sprint se basó sobre todo en la mejora de pantallas ya existentes y creación de otras. Creé las pantallas de cambio de idioma, consejos plantas y la de comprobación de sensores implementando la función de internacionalización de ellas. Además mejoré algunos detalles de la pantalla de inicio.

Sprint 3 Mejoras Visuales de las Pantallas 21/11/2024 - 05/12/2024

Este cuarto sprint se basó en la mejora de pantallas que ya había creado previamente. Mejoré la interfaz de la pantalla de cambio de idioma implementando iconos de banderas para que fuera mas visual para el usuario. Además, modifiqué tanto la pantalla de consejos de plantas y la pantalla de los sensores para que fuera también mas visual e intuitiva para el usuario mediante el uso de iconos y colores.

Sprint 4 Implementación de Gráficas y Visualización de Datos 05/12/2024 - 19/12/2024

Este quinto sprint se basó en la creación de una gráfica para hacer más visual la representación de los datos que llegan desde la API. Creé la pantalla de gráficos e implementé un gráfico para ver como se representaban los datos en el mismo para luego poder trabajar mejor con ellos. Además, probé a usar el método "POST" para poder enviar datos desde la aplicación a la API y que quedaran guardados en la base de datos.

Sprint 5 Optimización de Gráficas y Refinamiento de la Interfaz 19/12/2024 - 23/1/2025

Durante este sexto sprint estuve mirando como mejorar el gráfico y crearlo de una manera que fuera muchos mas visual e intuitiva. Además, creé tanto la pantalla de eliminar plantas como la de modificar plantas para que la aplicación contemplase dichas funcionalidades. Por último estuve intentando adaptar la pantalla al poner el móvil en horizontal.

Sprint 6 Creación de nuevas Gráficas y mejora de la Documentación 23/1/2025 - 6/2/2025

Este séptimo sprint me centré en la creación tanto de la gráfica de luz como la de temperatura. Además, he seguido mejorando el conjunto de todas ellas para que cada vez vayan quedando mas visible para el usuario. También me centré en hacer los apartados del plan de proyecto y el de la especificación de requisitos de la documentación.

Sprint 7 Creación de la pantalla de Hitos y de la Gráfica de Humedad de Ambiente 6/2/2025 - 27/2/2025

Durante este octavo sprint me centré en la creación de varios hitos diarios relacionados con los sensores tanto de humedad, temperatura y luz a completar diariamente. A parte, agregué la gráfica de humedad de ambiente a las ya creadas.

Sprint 8 Mejora en los Hitos y Gráficas 27/2/2024 - 13/3/2025

Durante este noveno sprint me centré sobre todo en la mejora de los hitos. Hice cambios para que la pantalla fuera más visual y mas e interactiva para el usuario separando los hitos temporalmente dependiendo de cada cuanto se tenían que completar dichos hitos. Además, agregué un nuevo hito mensual que requiere de la interacción del usuario con la aplicación para que aparezca como completado. También hice alguna pequeña mejora en las gráficas como fueron el cambio de un icono y cambios en los límites de las bandas de las gráficas.

Sprint 9 13/3/2024 - 27/3/2025

Sprint 10 27/3/2024 - 10/4/2025

Sprint 11 10/4/2024 - 24/4/2025

Sprint 12 24/4/2024 - 8/5/2025

Sprint 12 8/5/2024 - 22/5/2025

A.3. Estudio de viabilidad

La viabilidad del proyecto debe ser evaluada tanto a partir del ámbito económico como del legal, ya que, aunque un proyecto parezca correcto, puede fracasar si los costes son inviables o si no cumple con la normativa legal estipulada. Por ello, se necesita que un proyecto pueda ser sostenible en el tiempo y acate la normativa actual.

En este apartado se realizará un estudio detallado de la viabilidad económica analizando los costes que supondría el desarrollo del proyecto y formas de retorno de la inversión. Aparte, se hará un estudio de la

viabilidad legal del proyecto detallando las normativas actuales y licencias de herramientas usadas principalmente.

Ambos estudios serán explicados con ejemplos realistas para demostrar que es un proyecto viable tanto económica como legalmente.

Viabilidad económica

En este apartado se analizará la viabilidad económica del proyecto desde dos perspectivas diferentes.

En primer lugar, se estudiará la viabilidad del desarrollo llevado a cabo por el alumno, donde se explicarán los costes reales empleados para el desarrollo de la aplicación.

A continuación, se estudiará la viabilidad del proyecto orientado a un proceso de comercialización, estudiando su sostenibilidad y viabilidad.

Costes de desarrollo

Se van a explicar los costes asociados tanto al software como al hardware, junto con los costes de personal que se necesitan para la realización del proyecto:

Coste Hardware

En la tabla [A.1](#) aparecen reflejados los gastos que se han llevado a cabo tanto para el desarrollo de la aplicación, como para realizar las pruebas necesarias hasta su correcto funcionamiento. Estos costes han sido prorrateados considerando una vida útil de unos 4 años para cada dispositivo y teniendo en cuenta que la duración del proyecto ha sido de unos 8 meses.

Componente	Coste total (€)	Vida útil (años)	Coste prorrateado (€)
Ordenador portátil	1000	4	166.67
Teléfono móvil	250	4	41.67
Total			208.34

Tabla A.1: Costes hardware prorrateados en 8 meses

Coste Software

En cuanto al software y licencias, no ha sido necesario hacer ningún gasto debido a que todas las herramientas usadas en el desarrollo, como pueden ser Android Studio o Flutter, han sido completamente gratuitas.

El costo total ha sido de cero euros, por lo que se demuestra que se pueden realizar proyectos sin la necesidad de tener casi gastos en software por la amplia variedad de herramientas gratuitas que existen.

Coste Desarrollador

Aunque la aplicación la ha llevado a cabo el propio alumno, se ha planteado un coste ficticio que debería haber costado la realización de la misma.

Se ha planteado un escenario en el que el desarrollador trabaja a media jornada durante unos 8 meses, que es el tiempo que dura el proyecto. Durante ese tiempo, el desarrollador, al ser un programador junior, cobrará un salario bruto de 900 euros al mes.

A continuación, se puede ver en la tabla A.2 los gastos totales al contratar al desarrollador.

Sección	Coste (€)
Coste mensual para la empresa	1170
Retención IRPF (12 %)	108
Cotización a la Seguridad Social (30 %)	270
Salario neto mensual	792
Salario neto tras los 8 meses	9360

Tabla A.2: Costes Desarrollador

Coste Total

En la tabla A.3 se muestra la suma de todos los gastos necesarios para el desarrollo del proyecto.

Sección	Coste (€)
Coste Hardware	1250
Coste Desarrollador	9360
Coste Total	10610

Tabla A.3: Coste Total

Costes de comercialización

Se van a explicar los costes asociados tanto al software como al hardware, junto con los costes de personal que se necesitan para la comercialización de la aplicación.

Al ser un proyecto de desarrollo de una aplicación, solo se tendrán en cuenta los gastos asociados a dicha actividad. No se tendrán en cuenta gastos como pueden ser las Raspberry Pi o la fabricación de las macetas.

En este escenario planteado se van a calcular los costes únicamente teniendo un solo desarrollador. Los costes tanto de hardware como de desarrollador aumentarían proporcionalmente en función del número de desarrolladores contratados. Sin embargo, al contar con más desarrolladores, el tiempo para terminar el proyecto también se reduciría proporcionalmente, lo que compensaría el incremento de los costes.

Coste Hardware Comercialización

En la tabla A.4 aparecen reflejados los gastos que se han llevado a cabo tanto para el desarrollo de la aplicación, como para realizar las pruebas necesarias hasta su correcto funcionamiento.

Componente	Coste (€)
Ordenador portátil	1000
Teléfono móvil	250
Total	1250

Tabla A.4: Costes Hardware Comercialización

Coste Software Comercialización

En cuanto al software y licencias, han sido necesarios pocos gastos debido a que las herramientas usadas en el desarrollo de la aplicación han sido gratuitas. Estos pequeños gastos se han reflejado en la tabla A.5

Sección	Coste (€)
Cuenta de desarrollador de Google Play	25
Cuenta de desarrollador de App Store	99
Dominio web para la marca	12
Gasto Software total	136

Tabla A.5: Costes Desarrollador

Coste Desarrollador Comercialización

Se ha planteado un escenario en el que se va a contratar a un profesional con 3 años de experiencia para que trabaje a jornada completa y pueda mejorar y dar soporte a la aplicación.

En la tabla A.6 se van a calcular los gastos de los primeros 6 meses que esté contratado el desarrollador, que corresponden a los meses que va a estar en período de prueba.

Sección	Coste (€)
Coste mensual para la empresa	2751
Retención IRPF (15 %)	315
Cotización a la Seguridad Social (31 %)	651
Salario neto mensual	1785
Salario bruto tras los 6 meses de prueba	16506

Tabla A.6: Costes Desarrollador

Coste Total Comercialización

En la tabla A.7 se muestra la suma de todos los gastos necesarios de los 6 primeros meses en la parte de la aplicación móvil para la comercialización del producto.

Sección	Coste (€)
Coste Hardware	1250
Coste Software	136
Coste Desarrollador	16506
Coste Total	17892

Tabla A.7: Coste 6 meses Comercialización

Viabilidad legal

En el desarrollo de GreenInHouse 2.0 se han utilizado aplicaciones y herramientas gratuitas, respetando las licencias y condiciones de uso de cada una.

A continuación se van a detallar las aplicaciones y herramientas usadas en el desarrollo:

- **Flutter:** es un software libre y se distribuye bajo la licencia de tipo BSD la cual permite que se pueda usar con fines comerciales sin ningún coste. Esto hace que Flutter sea una de las mejores opciones para la creación de aplicaciones ya sea con fines educativos o para su futura comercialización.
- **Dart:** es un lenguaje de programación desarrollado por Google y se encuentra bajo la licencia BSD al igual que Flutter. Esta licencia hace que esté disponible de manera gratuita y que además esté permitida su uso para fines comerciales.
- **Android Studio:** es un software de uso gratuito tanto para proyectos personales como para temas comerciales. No requiere de adquisición de licencias ni suscripciones por lo que no existe ninguna restricción por usar este software en la creación de nuevas aplicaciones y distribuirlas a través de Google Play o App Store.
- **Recursos visuales y paquetes externos:** Durante el desarrollo de la aplicación se han usado iconos provenientes de bibliotecas libres como *Material Icons* que permiten su uso y comercialización y de paquetes externos obtenidos mediante pub.dev, los cuales tienen licencia BSD.

Gracias a la existencia de todo este tipo de herramientas gratuitas que permiten su uso con fines de comercialización, se ha podido crear la aplicación de GreenInHouse 2.0 y se podría comercializar sin ningún tipo de problema legal.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apéndice se van a detallar los requisitos esenciales, tanto funcionales como no funcionales, para el desarrollo del proyecto. La definición de requisitos es un paso fundamental, ya que se definen tanto las funcionalidades como las restricciones que el sistema debe cumplir para satisfacer las necesidades del usuario.

B.2. Objetivos generales

El objetivo principal del proyecto es proporcionar la posibilidad de monitorizar el estado de las plantas mediante distintos sensores que envían los datos a través de una Raspberry Pi.

Objetivos del proyecto:

- **Monitorización en tiempo real:** Obtener datos en tiempo real desde la raspberry pi para mostrárselos mediante la interfaz al usuario.
- **Historial de Mediciones:** Permitir que el usuario pueda consultar datos en un determinado periodo de tiempo.
- **Gestión de Plantas:** Permitir que el usuario pueda crear, eliminar y editar las plantas según sus intereses.
- **Estado de Sensores:** Permitir que el usuario pueda comprobar el estado de los sensores.

- **Alertas y Notificaciones:** Notificar al usuario cuando los valores que nos devuelva alguno de los sensores no estén dentro de los rangos establecidos como óptimos.
- **Interfaz intuitiva:** Creación de una interfaz intuitiva para que el usuario pueda realizar las gestiones que necesite sin que tenga dudas de cómo.
- **Interfaz Adaptativa:** Creación de una interfaz que se adapte correctamente al tamaño de la pantalla.
- **Soporte Multilingüe:** Añadir la posibilidad que el usuario pueda cambiar de idioma la aplicación y así hacerla más accesible para más usuarios.

B.3. Catálogo de requisitos

Los requisitos se van a dividir en función de si son funcionales o no funcionales:

- **Funcionales:**
 - **RF-01:** El sistema tiene que permitir que los usuarios puedan crear nuevas plantas proporcionando el nombre y tipo de planta.
 - **RF-02:** La aplicación tiene que mostrar el estado de los sensores.
 - **RF-03:** El usuario tiene que poder consultar un histórico de las mediciones en un rango de tiempo determinado.
 - **RF-04:** El sistema tiene que permitir que los usuarios puedan editar y eliminar las plantas registradas.
 - **RF-05:** El sistema tiene que notificar cuando un valor de los devueltos por los sensores se salga de los valores óptimos.
 - **RF-06:** El usuario podrá configurar los umbrales tanto de humedad, temperatura y luz en función del tipo planta que seleccione plantar.
 - **RF-07:** La aplicación deberá permitir seleccionar el idioma deseado.
 - **RF-08:** La aplicación deberá permitir al usuario registrar manualmente el cambio de tierra de la planta.

- **RF-09:** La aplicación debe mostrar los hitos agrupados por frecuencia: diarios, semanales, mensuales, etc.
- **RF-10:** La aplicación tiene que mostrar los valores recogidos por los sensores tanto de humedad, como temperatura y luz.
- **RF-11:** La aplicación tiene que permitir que los usuarios puedan seleccionar los días que echar hacia atrás en el tiempo para visualizar más datos en los gráficos.
- **RF-12:** La aplicación tiene que mostrar los consejos dados para cada tipo de planta.
- **RF-13:** La aplicación deberá permitir al usuario personalizar la visualización tanto de gráficas como de hitos.
- **RF-14:** La aplicación deberá permitir al usuario personalizar el tiempo de cambio de tierra.
- **RF-15:** La aplicación debe permitir al usuario personalizar el tiempo de añadir fertilizante.
- **RF-16:** La aplicación debe permitir al usuario personalizar la imagen de la planta ya sea capturar una con la cámara o seleccionarla desde la galería.
- **RF-17:** La aplicación tiene que mostrar la barra con el porcentaje de cuidado de la planta.
- **RF-18:** La aplicación tiene que mostrar un mensajes de error si no se ha podido realizar una petición a la API.

■ **No Funcionales:**

- **RNF-01:** La aplicación debe ser compatible con dispositivos android.
- **RNF-02:** La interfaz ha de ser intuitiva para el usuario para que pueda navegar sin problemas por la aplicación.
- **RNF-03:** La aplicación debe adaptarse bien a los diferentes dispositivos desde los que se use sin afectar a la visibilidad de la información.
- **RNF-04:** La base de datos que se vaya a usar debe ser eficiente para almacenar y recuperar datos históricos sin que puedan influir en la velocidad de respuesta de la aplicación.
- **RNF-05:** El sistema debe ser escalable para permitir que se puedan añadir más sensores sin que se tengan que hacer cambios muy drásticos en el código.

- **RNF-06:** La aplicación debe funcionar correctamente en condiciones de sin conexión mostrando el contenido disponible.

B.4. Especificación de requisitos

Una muestra de cómo podría ser una tabla de casos de uso:

CU-1	Crear Planta
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-01
Descripción	El usuario podrá añadir una nueva planta indicando su nombre y tipo
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de crear planta2. El usuario elige el nombre y tipo de planta3. El usuario pulsa el botón “Añadir”
Postcondición	La planta será creada en la base de datos
Excepciones	<ul style="list-style-type: none">■ Los campos estén incompletos■ Error en la conexión con la base de datos
Importancia	Alta

Tabla B.1: CU-1 Creación de Plantas.

CU-2	Editar plantas
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-04
Descripción	El usuario podrá editar una planta existente
Precondición	Debe haber una planta creada.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Modificar Planta 2. El usuario elige el nombre de una planta activa 3. El usuario cambiará el nombre de planta o tipo 4. EL usuario dará al botón Mofificar Planta
Postcondición	La planta será modificada en la base de datos
Excepciones	<ul style="list-style-type: none"> ■ Los campos estén incompletos ■ Error en la conexión con la base de datos
Importancia	Alta

Tabla B.2: CU-2 Editar Plantas.

CU-3	Eliminar plantas
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-04
Descripción	El usuario podrá eliminar una planta existente
Precondición	Debe haber una planta creada.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Eliminar plantas 2. El usuario elige el nombre de la planta 3. El usuario pulsa el botón “Eliminar”
Postcondición	La planta será eliminada en la base de datos de plantas activas.
Excepciones	<ul style="list-style-type: none"> ■ Los campos estén incompletos ■ Error en la conexión con la base de datos
Importancia	Alta

Tabla B.3: CU-3 Eliminación de Plantas.

CU-4	Ver estado actual de los sensores
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-02
Descripción	El usuario podrá ver el estado en tiempo real de los sensores.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de sensores2. El sistema carga los estados.
Postcondición	Se muestran los estados en tiempo real.
Excepciones	<ul style="list-style-type: none">■ Error en la conexión con los sensores.■ Error en la conexión con la base de datos.
Importancia	Alta

Tabla B.4: CU-4 Estado de los sensores.

CU-5	Cambio de Idioma
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-07
Descripción	El usuario podrá cambiar el idioma de la aplicación.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de Idioma2. El usuario elegirá el idioma que prefiera.
Postcondición	Se cambia el idioma de toda la aplicación al seleccionado por el usuario.
Excepciones	El idioma no carga correctamente.
Importancia	Media

Tabla B.5: CU-5 Cambio de Idioma.

CU-6	Hitos Diarios
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-09
Descripción	El usuario podrá ver los hitos diarios si están completados o no.
Precondición	<ul style="list-style-type: none"> ■ El usuario deberá haber accedido a la aplicación. ■ Deberán haber hitos creados.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de hitos. 2. El usuario desplegará la sección de hitos mensuales.
Postcondición	Se muestran los hitos organizados.
Excepciones	<ul style="list-style-type: none"> ■ Error en la conexión con la base de datos. ■ Error al desplegar la sección.
Importancia	Media

Tabla B.6: CU-6 Visualización de hitos mensuales.

CU-7	Hitos Mensuales
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-09
Descripción	El usuario podrá ver los hitos mensuales si están completados o no.
Precondición	<ul style="list-style-type: none">■ El usuario deberá haber accedido a la aplicación.■ Deberán haber hitos creados.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de hitos.2. El usuario desplegará la sección de hitos diarios.
Postcondición	Se muestran los hitos organizados.
Excepciones	<ul style="list-style-type: none">■ Error en la conexión con la base de datos.■ Error al desplegar la sección.
Importancia	Media

Tabla B.7: CU-7 Visualización de hitos mensuales.

CU-8	Cambio de Tierra
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-08, RF-09
Descripción	El usuario podrá marcar cuándo ha cambiado la tierra de la planta.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Hitos. 2. El usuario desplegará los hitos mensuales. 3. El usuario seleccionará el hito de cambio de tierra y le dará a aceptar.
Postcondición	Se cambiará el estado del hito a completado.
Excepciones	<ul style="list-style-type: none"> ■ Error al acceder a las fechas guardadas. ■ Error a la hora de guardar la fecha del cambio de tierra.
Importancia	Media

Tabla B.8: CU-8 Cambio de Tierra.

CU-9	Gráfica de Humedad
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-10
Descripción	El usuario podrá ver los datos recogidos por el sensor de humedad en una gráfica.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de Gráficos.2. El usuario desplegará el sensor de humedad.
Postcondición	Se visualizará los datos recogidos por el sensor de humedad en la gráfica.
Excepciones	<ul style="list-style-type: none">■ Error al acceder a la base de datos.■ Error al desplegar la sección.
Importancia	Alta

Tabla B.9: CU-9 Gráfica de Humedad.

CU-10	Gráfica de Humedad de Ambiente
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-10
Descripción	El usuario podrá ver los datos recogidos por el sensor de humedad en una gráfica.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Gráficos. 2. El usuario desplegará el sensor de humedad de ambiente.
Postcondición	Se visualizará los datos recogidos por el sensor de humedad en la gráfica.
Excepciones	<ul style="list-style-type: none"> ■ Error al acceder a la base de datos. ■ Error al desplegar la sección.
Importancia	Alta

Tabla B.10: CU-10 Gráfica de Humedad de Ambiente.

CU-11	Gráfica de Luz
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-10
Descripción	El usuario podrá ver los datos recogidos por el sensor de luz en una gráfica.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de Gráficos.2. El usuario desplegará el sensor de luz.
Postcondición	Se visualizará los datos recogidos por el sensor de luz en la gráfica.
Excepciones	<ul style="list-style-type: none">■ Error al acceder a la base de datos.■ Error al desplegar la sección.
Importancia	Alta

Tabla B.11: CU-11 Gráfica de Luz.

CU-12	Gráfica de Temperatura
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-10
Descripción	El usuario podrá ver los datos recogidos por el sensor de temperatura en una gráfica.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Gráficos. 2. El usuario desplegará el sensor de temperatura.
Postcondición	Se visualizará los datos recogidos por el sensor de temperatura en la gráfica.
Excepciones	<ul style="list-style-type: none"> ■ Error al acceder a la base de datos. ■ Error al desplegar la sección.
Importancia	Alta

Tabla B.12: CU-12 Gráfica de Temperatura.

CU-13	Histórico de Valores en Gráficos
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-11
Descripción	El usuario podrá seleccionar los días que quiera volver atrás para ver el histórico de datos.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de Gráficos.2. El usuario desplegará uno de los sensores.3. El usuario seleccionará en días atrás los días que desee, hasta un máximo de 30.4. El usuario le dará a actualizar gráfica.
Postcondición	Se ajustará el gráfico para mostrar el histórico deseado por el usuario.
Excepciones	Error al acceder a la base de datos.
Importancia	Media

Tabla B.13: CU-13 Histórico de Valores en Gráficos.

CU-14	Consejos Planta
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-12
Descripción	El usuario podrá visualizar los consejos de la planta.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a la pantalla de Inicio.2. El usuario seleccionará el botón de ver consejos de plantas.
Postcondición	Se visualizará los consejos con los valores óptimos de cada sensor.
Excepciones	Error al acceder a la base de datos.
Importancia	Alta

Tabla B.14: CU-14 Consejos Plantas.

CU-15	Personalización en la Visualización de los Gráficos
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-13
Descripción	El usuario podrá personalizar los gráficos que desee ver.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Ajustes. 2. El usuario seleccionará en el apartado de mostrar gráficos los que desee ver y los que no.
Postcondición	La interfaz se ajusta a las selecciones del usuario.
Excepciones	<ul style="list-style-type: none"> ■ Error al aplicar los cambios en la pantalla de gráficos. ■ Error al guardar las preferencias.
Importancia	Alta

Tabla B.15: CU-15 Personalización en la Visualización de los Gráficos.

CU-16	Personalización en la Visualización de los Hitos
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-13
Descripción	El usuario podrá personalizar los hitos que desee ver.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Ajustes. 2. El usuario seleccionará en el apartado de mostrar hitos los que desee ver y los que no.
Postcondición	La interfaz se ajusta a las selecciones del usuario.
Excepciones	<ul style="list-style-type: none"> ■ Error al aplicar los cambios en la pantalla de hitos. ■ Error al guardar las preferencias.
Importancia	Alta

Tabla B.16: CU-16 Personalización en la Visualización de los Hitos.

CU-17	Personalización en el cambio de tierra
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-14
Descripción	El usuario podrá personalizar la fecha del cambio de tierra.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Ajustes. 2. El usuario desplazará el punto hacia la derecha para aumentar los días y hacia la izquierda para disminuirlos en el apartado de frecuencia de cambio de tierra.
Postcondición	La interfaz se ajusta a los cambios del usuario.
Excepciones	<ul style="list-style-type: none"> ■ Error al aplicar los cambios en la pantalla de hitos. ■ Error al guardar las preferencias.
Importancia	Alta

Tabla B.17: CU-17 Personalización de la fecha del cambio de tierra.

CU-18	Personalización en el cambio de tierra
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-15
Descripción	El usuario podrá personalizar la fecha para agregar fertilizante.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Ajustes. 2. El usuario desplazará el punto hacia la derecha para aumentar los días y hacia la izquierda para disminuirlos en el apartado de frecuencia del fertilizante.
Postcondición	La interfaz se ajusta a los cambios del usuario.
Excepciones	<ul style="list-style-type: none"> ■ Error al aplicar los cambios en la pantalla de hitos. ■ Error al guardar las preferencias.
Importancia	Alta

Tabla B.18: CU-18 Personalización de la fecha para agregar fertilizante.

CU-19	Personalización de la imagen de la planta
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-16
Descripción	El usuario podrá personalizar la foto de la planta.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Inicio. 2. El usuario pulsará el icono de la cámara. 3. El usuario elegirá entre las opciones dadas ya sea agregar nueva imagen haciendo una foto o desde la galería o eliminar la imagen.
Postcondición	Se modifica la imagen en función de la elección del usuario.
Excepciones	<ul style="list-style-type: none"> ■ Error al agregar la imagen. ■ Error al borrar la imagen.
Importancia	Media

Tabla B.19: CU-19 Personalización de la imagen de la planta.

CU-20	Visualización del cuidado de la planta
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-17
Descripción	El usuario podrá visualizar el cuidado de la planta.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la pantalla de Inicio. 2. Se mostrará una barra con el porcentaje de cuidado completado de la planta
Postcondición	Se muestra el porcentaje que irá variando en función de si se completan los hitos.
Excepciones	<ul style="list-style-type: none"> ■ Error al cargar los hitos completados.
Importancia	Media

Tabla B.20: CU-20 Visualización del cuidado de la planta.

CU-21	Mensajes de Error
Versión	1.0
Autor	Luis Rojo
Requisitos asociados	RF-18
Descripción	El usuario podrá visualizar mensajes de errores tras imposibilidad de comunicación con la API.
Precondición	El usuario deberá haber accedido a la aplicación
Acciones	<ol style="list-style-type: none">1. El usuario accede a cualquier pantalla que requiera comunicación con la API.2. Se mostrará un mensaje de error comunicando la imposibilidad de contactar con la API debido a error de conexión.
Postcondición	Se devuelve al usuario a la pantalla de inicio.
Excepciones	<ul style="list-style-type: none">■ Error al devolver al usuario a la pantalla de inicio.
Importancia	Alta

Tabla B.21: CU-21 Mensajes de Error.

Apéndice C

Especificación de diseño

C.1. Introducción

En esta sección se describe el diseño de la aplicación GreenInHouse 2.0, desarrollada para gestionar mejor el cuidado de las plantas mediante el uso de sensores. La aplicación ha sido diseñada mediante la integración con herramientas previamente creadas por el autor del primer proyecto GreenInHouse, Óscar Valverde Escobar, ya que ha sido necesaria la interacción entre la API REST y la aplicación para poder trabajar con los datos almacenados en la base de datos.

El diseño se ha creado teniendo en cuenta los requisitos funcionales previamente definidos en la anterior sección del proyecto.

C.2. Diseño de datos

El diseño de datos de la aplicación se basa en los datos recogidos de la base de datos previamente creada por el autor del primer proyecto GreenInHouse. De esta manera, la aplicación puede trabajar tanto con los datos recogidos de los sensores como con los datos almacenados sobre las plantas.

La aplicación no accede directamente a la base de datos, sino que debe comunicarse con ella a través de la API REST, la cual permitirá realizar las llamadas a los diferentes endpoints, ya sea para obtener información de la base de datos como para modificarla. La información de la base de datos también puede ser modificada o eliminada a través de la aplicación mediante diferentes pantallas que sirven para la creación, modificación o eliminación de plantas creadas.

El diseño del flujo de datos que tiene el proyecto es el siguiente:[C.1](#):

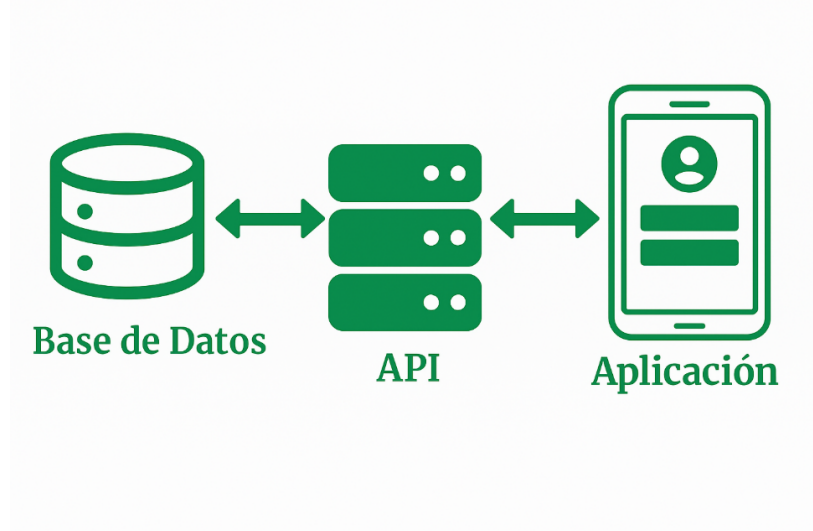


Figura C.1: Diseño de Datos

C.3. Diseño arquitectónico

En cuanto al diseño arquitectónico, nos referimos a la manera en que se organiza y estructura un proyecto, en este caso la aplicación GreenInHouse 2.0.

Este diseño muestra cómo el usuario interactúa con la interfaz de la aplicación, ya sea para llevar a cabo alguna acción en la que interviene también la interacción con la base de datos a través de la API, o alguna acción que no requiera de dicha comunicación con la API, como por ejemplo, cambiar el idioma de la aplicación.

Para este proyecto se ha seguido una arquitectura basada en el patrón Modelo-Vista-Controlador(MVC) extendido con una capa de servicios usada para la gestión de los datos a través de una API:

1. **Modelo:** Se refiere a las clases de datos que se usan en la aplicación de GreenInHouse 2.0, como pueden ser:
 - a) Planta
 - b) Sensor
 - c) Consejo

2. **Vista:** Se refiere a las pantallas con las que el usuario va a interactuar en la aplicación, como pueden ser:
 - a) Pantalla de Gráficos
 - b) Pantalla de Hitos
 - c) Pantalla de Inicio
 - d) Pantalla de Ajustes
3. **Controlador:** Se refiere a la lógica que hay detrás de cada acción realizada por el usuario, como puede ser:
 - a) Cambio de Idioma
 - b) Creación de Planta
 - c) Cambio de Tierra
4. **Servicio:** Se refiere a todas esas llamadas HTTP (GET, POST, PUT, DELETE) que el usuario puede realizar, las cuales actúan de intermediarias entre la lógica del programa y la API.

Para que este diseño arquitectónico sea completamente funcional, se va a necesitar que haya conexión a internet. Dependiendo de si hay conexión a internet o no, se van a dar los siguientes casos:

1. **Con Internet:** La mayoría de las funcionalidades de la aplicación dependen de la comunicación con la base de datos a través de la API. Para ello, se va a necesitar que tanto la maceta como el teléfono móvil estén conectados a la misma red. Estas funcionalidades incluyen:
 - a) Consulta de Sensores
 - b) Consulta de Gráficos
 - c) Consulta de Consejos
 - d) Crear Plantas
 - e) Editar Plantas
 - f) Eliminar Plantas
2. **Sin Internet:** Sin esta conexión a internet, la aplicación va a ser parcialmente funcional y entre estas cosas están:
 - a) La interfaz carga sin problema
 - b) Los datos guardados en el archivo “SharedPreferences” si que se usan.
 - c) Funciones que no requieran de la API como el cambio de idioma

C.4. Diseño procedimental

A diferencia del diseño de arquitectura que se basaba en conocer la organización de la aplicación, el diseño procedimental se centra en ver cómo se comporta la aplicación durante su ejecución y uso. Va a mostrar la interacción entre el usuario y la aplicación y los resultados de las decisiones tomadas en tiempo de ejecución.

A continuación se verán ejemplos de diseños procedimentales de las diferentes acciones posibles dentro de la aplicación:

Creación de una Planta:

1. El usuario abre la aplicación.
2. Accede a la pantalla de creación de plantas.
3. Rellena los campos necesarios y pulsa el botón crear planta.
4. El sistema valida los campos.
5. Si son válidos se envía una petición POST a la API y en caso contrario te da error y vuelves al paso 3.
6. La planta se crea y queda registrada en la base de datos.

Consulta de Hitos Diarios:

1. El usuario abre la aplicación.
2. Accede a la pantalla de hitos.
3. Se hace una petición GET a la API para que devuelva tanto los datos recogidos por los sensores como los de los consejos.
4. El sistema compara los datos recogidos por los sensores con los de los consejos y determina el estado del hito

Consulta de Gráficos:

1. El usuario abre la aplicación.
2. Accede a la pantalla de gráficos.

3. Se hace una petición GET a la API para que devuelva tanto los datos recogidos por los sensores como los de los consejos.
4. El sistema compara los datos recogidos por los sensores con los de los consejos y determina el estado del gráfico

Consulta de Sensores:

1. El usuario abre la aplicación.
2. Accede a la pantalla de comprobar sensores.
3. Se hace una petición GET a la API para que devuelva el estado de los sensores

Modificación de una Planta:

1. El usuario abre la aplicación.
2. Accede a la pantalla de modificar planta.
3. Selecciona el tipo de planta y el nombre de la planta a modificar.
4. El sistema valida los campos.
5. Si son válidos se envía una petición PUT a la API y en caso contrario te da error y vuelves al paso 3.
6. La planta se modifica y queda registrada en la base de datos.

Eliminación de una Planta:

1. El usuario abre la aplicación.
2. Accede a la pantalla de eliminar planta.
3. Selecciona la planta a eliminar y pulsa el botón eliminar planta.
4. El sistema valida los campos.
5. Si son válidos se envía una petición DELETE a la API y en caso contrario te da error y vuelves al paso 3.
6. La planta se elimina de plantas activas en la base de datos y se registra como planta marchita.

Apéndice D

Documentación técnica de programación

D.1. Introducción

Este apéndice proporciona una guía detallada sobre la estructura del código del proyecto, las herramientas usadas para el funcionamiento de la aplicación junto con su configuración. Se analiza la estructura de directorios que hay en la aplicación, además se incluye un manual del programador en el que se explica cómo instalar y configurar el entorno de desarrollo junto con la ejecución del proyecto. Con este apéndice se intenta que futuros desarrolladores que vayan a modificar o mejorar la aplicación sepan cómo dejar todo instalado y configurado de la mejor forma posible.

D.2. Estructura de directorios

El proyecto **GreenInHouse 2.0** sigue una estructura de directorios bien organizada, permitiendo una gestión eficiente del código, los recursos y la configuración. A continuación, se describen las principales carpetas y su contenido:

- **Directorio raíz** (`greeninhouse2/`) Este es el directorio principal del proyecto y contiene archivos de configuración esenciales para Flutter y Dart, además de la estructura del código fuente.
 - `.dart_tool/`: Carpeta generada automáticamente por Flutter y Dart con herramientas internas necesarias para la compilación.

- **.idea/**: Archivos de configuración de Android Studio para la organización del proyecto.
- **build/**: Carpeta generada durante la compilación, almacena archivos de salida del proyecto.
- **ios/, android/, linux/, macos/, web/, windows/**: Carpetas para cada plataforma soportada, con la configuración y código necesarios para compilar en cada sistema operativo.
- **assets/** Carpeta con todas las fotos que se van a usar en la aplicación.
- **flags/**: Carpeta con imágenes de iconos y estados de las plantas:
- **lib/** Directorio principal del código fuente en Flutter. Aquí se encuentra la lógica de la aplicación.
 - **generated/**: Contiene archivos generados automáticamente para la internacionalización.
 - ◊ **intl/l10n.dart** – Configuración de localización.
 - **l10n/**: Dedicada a la internacionalización del proyecto.
 - ◊ **intl_en.arb** – Traducciones en inglés.
 - ◊ **intl_es.arb** – Traducciones en español.
 - **api_service.dart** – Implementación de las llamadas a la API.
 - **botones_inicio.dart** – Definición de la barra de navegación inferior.
 - **main.dart** – Punto de entrada principal de la aplicación.
 - **pantalla_graficas.dart** – Pantalla principal con gráficos de sensores.
 - **pantalla_inicio.dart** – Pantalla inicial de la aplicación.
 - **pantalla_comprobacion_sensores.dart** – Revisión del estado de sensores.
 - **pantalla_creacionplantas.dart** – Creación de nuevas plantas.
 - **pantalla_modificarplanta.dart** – Modificación de plantas registradas.
 - **pantalla_eliminarplanta.dart** – Eliminación de plantas registradas.
 - **grafica_humedad.dart** – Representación de datos de humedad.
 - **grafica_luz.dart** – Visualización de datos de luminosidad.

- **grafica_temperatura.dart** – Evolución de la temperatura.
- **.gitignore** – Definición de archivos y carpetas a excluir del repositorio Git.
- **.metadata** – Archivo interno de configuración de Flutter.
- **analysis_options.yaml** – Reglas de análisis de código en Dart.
- **greeninhouse2.iml** – Configuración de IntelliJ/Android Studio.
- **pubspec.yaml** – Definición de dependencias, assets e información del proyecto.
- **pubspec.lock** – Registro de versiones de las dependencias utilizadas.
- **README.md** – Documentación inicial sobre el proyecto y su instalación.

D.3. Manual del programador

El objetivo del manual del programador es proporcionar una guía del desarrollo y mantenimiento de la aplicación. Este manual está dirigido a desarrolladores que necesiten saber cómo funcionan las herramientas utilizadas en el desarrollo de esta aplicación. También se han incluido las instrucciones necesarias para configurar el entorno de desarrollo y ejecutar la aplicación. Además, se han detallado aspectos que son clave, como la estructura de datos que sigue la aplicación y la comunicación con la API, entre otras cosas, con el objetivo de facilitar mejoras que se puedan llevar a cabo en un futuro.

Instalación de Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial que se usa en el desarrollo de *apps* para Android. Basado en el potente editor de código y las herramientas para desarrolladores de IntelliJ IDEA. Se puede descargar desde su página oficial [1].

Al descargar la aplicación y ejecutarla, nos saldrá lo que vemos en la figura D.1 a continuación y le daremos al botón de “Next”.

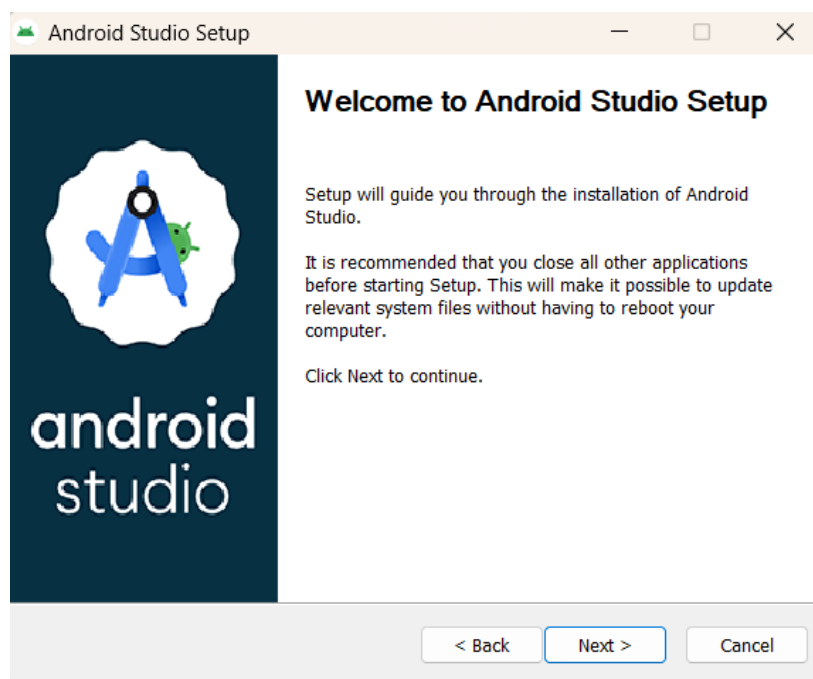


Figura D.1: Primer paso de la instalación de Android Studio.

Tras esto nos saldrá la ventana [D.2](#) en la que dejaremos lo que viene por defecto y le volveremos a dar a “Next”.

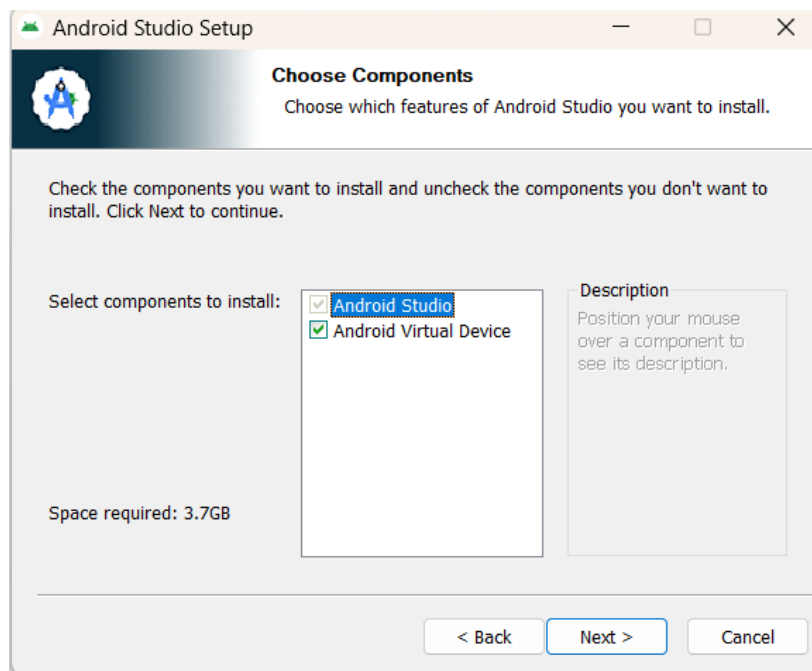


Figura D.2: Segundo paso de la instalación de Android Studio.

Por último, decidiremos la ruta donde instalar la aplicación, como se puede ver en la figura D.3 y le daremos a “Next” para que instale la aplicación en la ruta seleccionada.

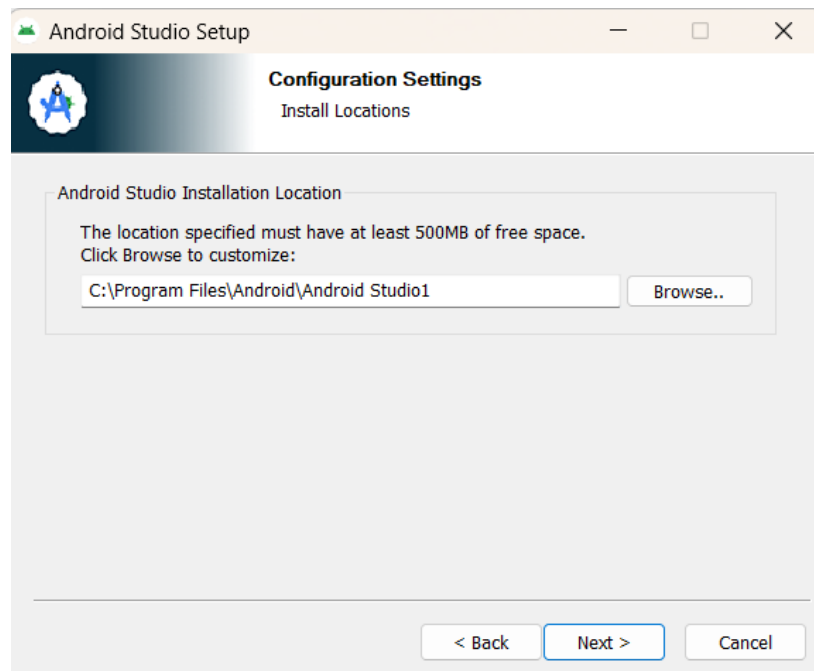


Figura D.3: Ruta de la instalación

Tras instalar Android Studio, vamos a pasar a la configuración de las herramientas instalando tanto Flutter como Dart. Esto se hace entrando en la pestaña de ajustes y en Plugins, como vemos en la figura D.4.

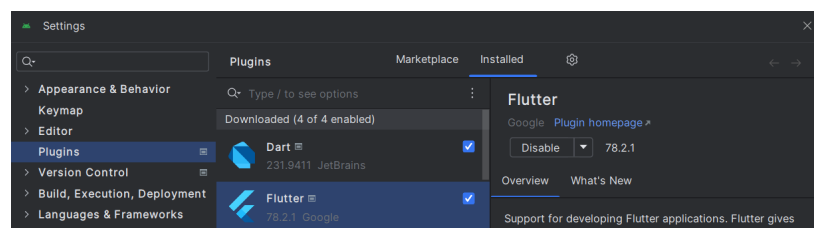


Figura D.4: Instalación plugins Android Studio

El siguiente paso es la configuración de SDK en el Android Studio. Esto se hace accediendo a ajustes y allí desde Languages & Frameworks como vemos en la figura D.5.

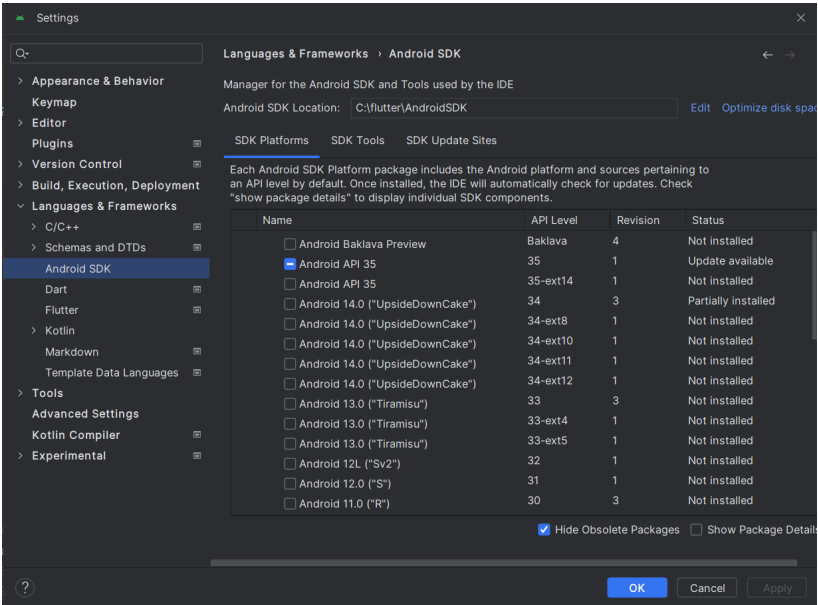


Figura D.5: Configuración de SDK en Android Studio

Por último, vamos a instalar unas herramientas en SDK Tools que vienen a la derecha de la de SDK Platforms, como vemos en la figura D.6, necesarias para el buen funcionamiento de la herramienta.

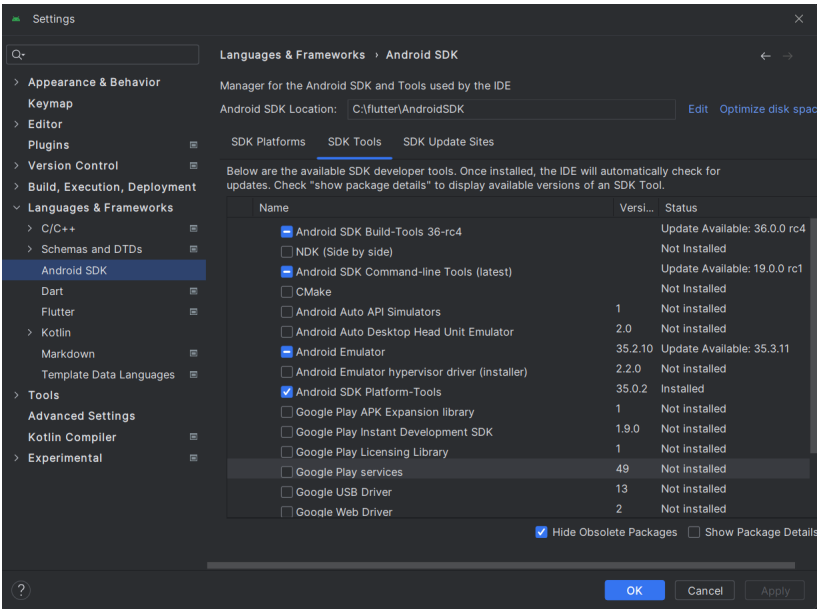


Figura D.6: Instalación de Herramientas de SDK

Instalación de Flutter

Para la instalación de Flutter tendremos que ir a su página web [2] y descargarlo. Allí nos preguntarán por nuestro sistema operativo, tras elegirlo nos preguntarán por el tipo de aplicación que queremos ya sea “Android” o “Web” o “Desktop”. En este caso, se selecciona “Windows” y “Android”. Tras esto, nos descargaremos el .zip que aparece en la web, como se puede ver en la figura D.7.

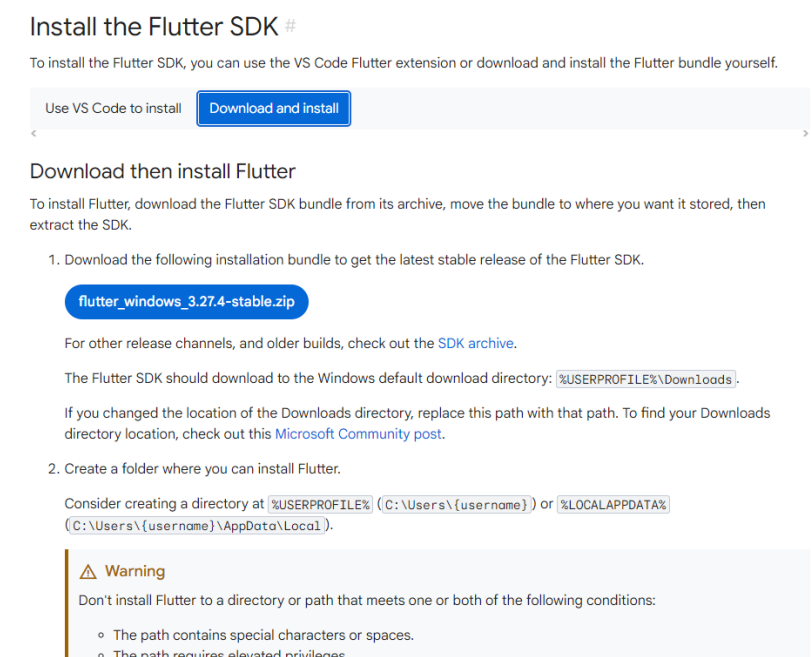


Figura D.7: Descarga del archivo.

A continuación, deberemos descomprimir el .zip y guardar la carpeta en un directorio que elijamos para de esta manera poder añadir en las variables de entorno la ruta del fichero “\bin”.

Primero, abriremos la configuración avanzada del sistema desde el buscador de Windows y nos saldrá la pantalla que vemos en la figura D.8. Daremos al botón que pone “Variables de entorno” para continuar.

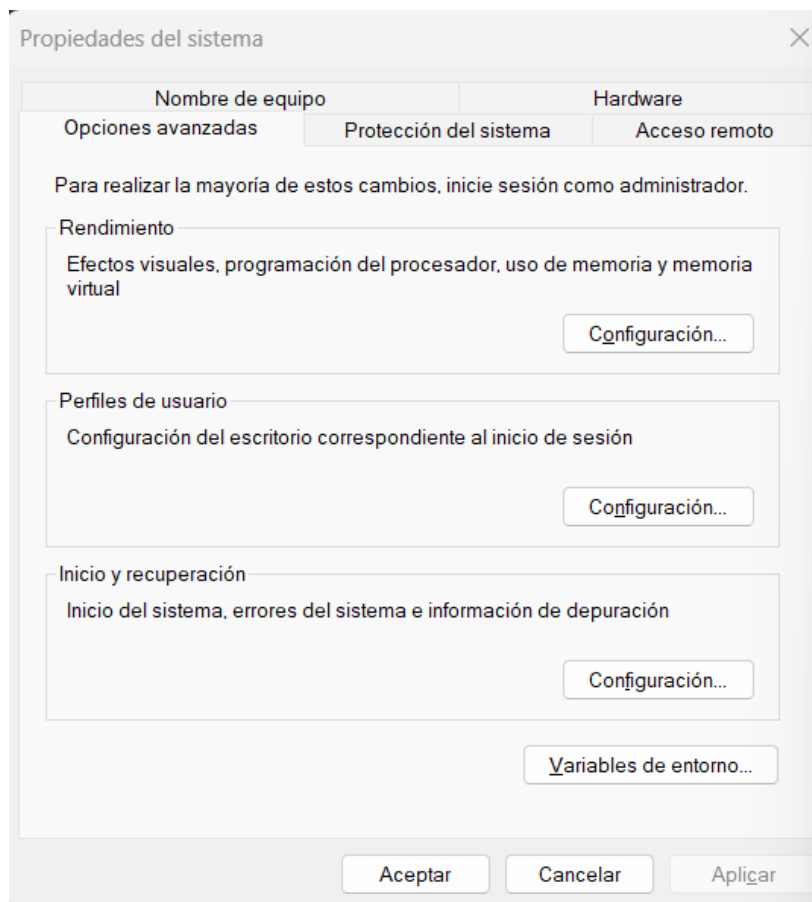


Figura D.8: Pantalla de configuración avanzada.

Seguido, buscaremos la variable que se llame “Path” y le daremos al botón de “Editar” como vemos en la figura D.9.

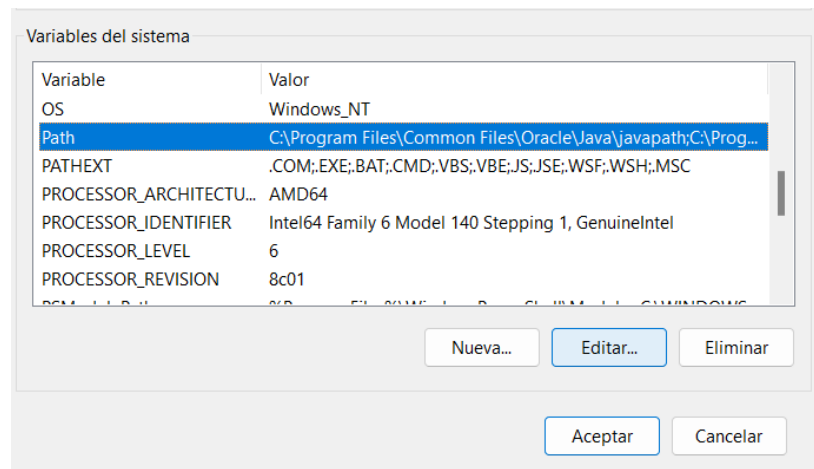


Figura D.9: Pantalla de variables de entorno.

Por último, le daremos al botón de “Nuevo” y añadiremos la ruta donde se encuentre nuestra carpeta “\bin”. Nos quedará añadido, como se puede ver en la figura [D.10](#) y le daremos a “Aceptar” para que se quede guardado.

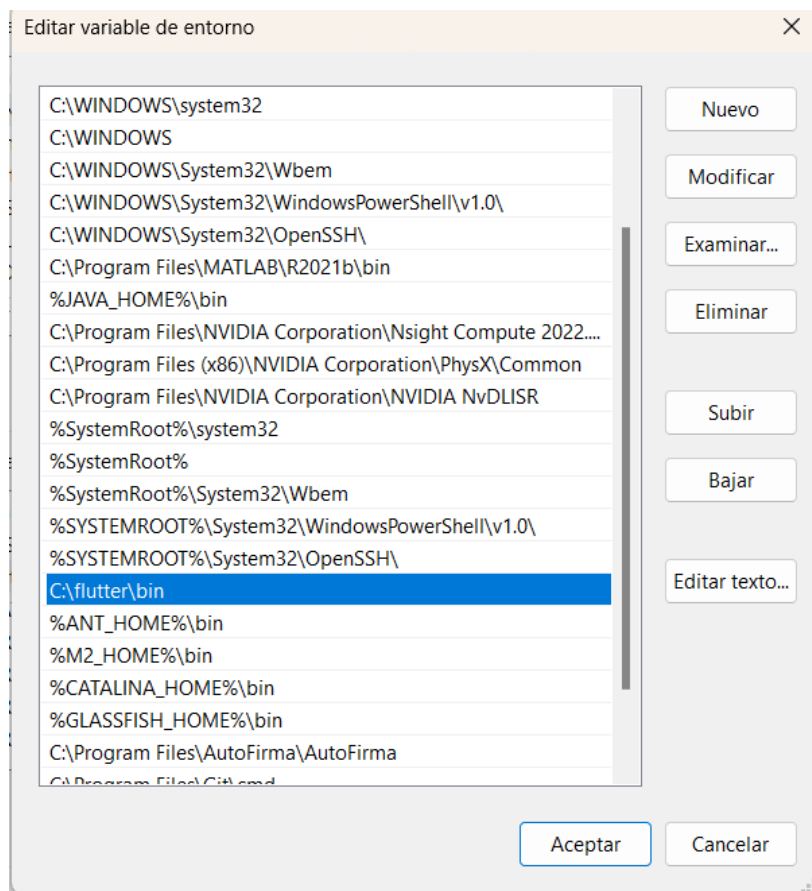


Figura D.10: Pantalla de editar variables de entorno.

Para comprobar que todo se ha instalado correctamente, abriremos una terminal y, mediante el comando “*flutter doctor*” verificaremos que se ha instalado bien y si falta alguna configuración o dependencia, como podemos ver en la figura D.11.

```
C:\Users\Luis Rojo>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.24.3, on Microsoft Windows [Version 10.0.22631.5039], locale es-ES)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✓] Android toolchain - develop for Android devices (Android SDK version 35.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.3.3)
[✓] Android Studio (version 2023.1)
[✓] VS Code (version 1.98.2)
[✓] Connected device (3 available)
[✓] Network resources

• No issues found!
```

Figura D.11: Uso del comando “*flutter doctor*”

D.4. Compilación, instalación y ejecución del proyecto

Para poder probar y utilizar la aplicación de GreenInHouse 2.0 durante su desarrollo ha sido necesario el uso de un emulador. El propio AndroidStudio tiene sus propios emuladores, los cuales puedes descargar y configurar en función de tus intereses de la aplicación.

Para poder llevar a cabo la ejecución del programa primero deberemos descargarnos el proyecto mediante el uso del comando: `git clone https://github.com/LuisRojoSanz/GreenInHouse2_APP.git`

Tras esto se habrá descargado el contenido del repositorio de github y se podrá ejecutar el proyecto.

Primero se deberá pulsar el botón del “+” en *Device Manager* para empezar a descargar el emulador como se ve en la figura D.12.

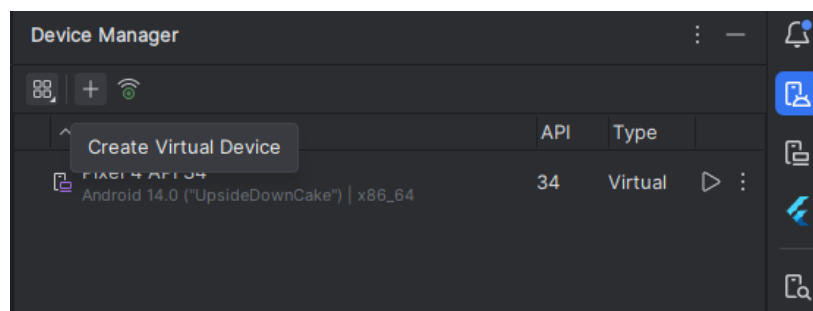


Figura D.12: Descarga de un emulador

Una vez se haya pulsado el “+” aparecerá la ventana que vemos en la figura D.13 en la que se seleccionará el modelo de emulador deseado y se pulsará el botón “Next”.

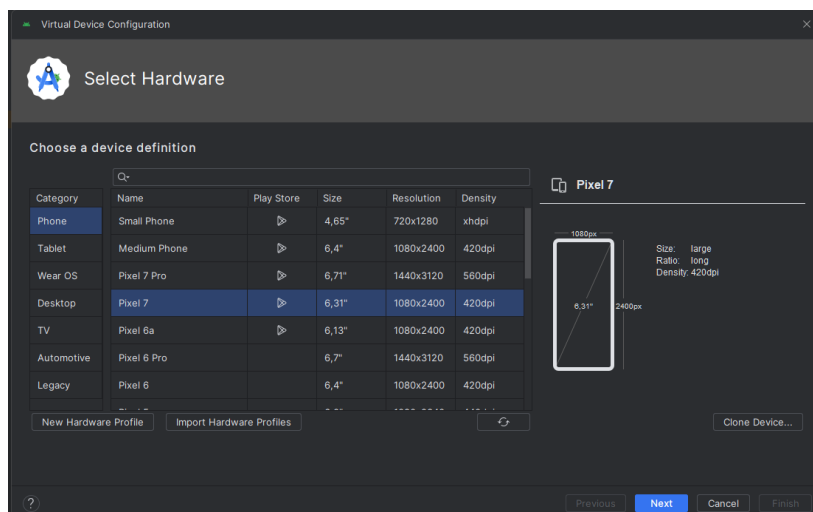


Figura D.13: Seleccionando un emulador

Tras esto habrá que seleccionar la versión de Android que vemos en la figura D.14 o una superior para el correcto funcionamiento del proyecto y darle al botón de “Next”

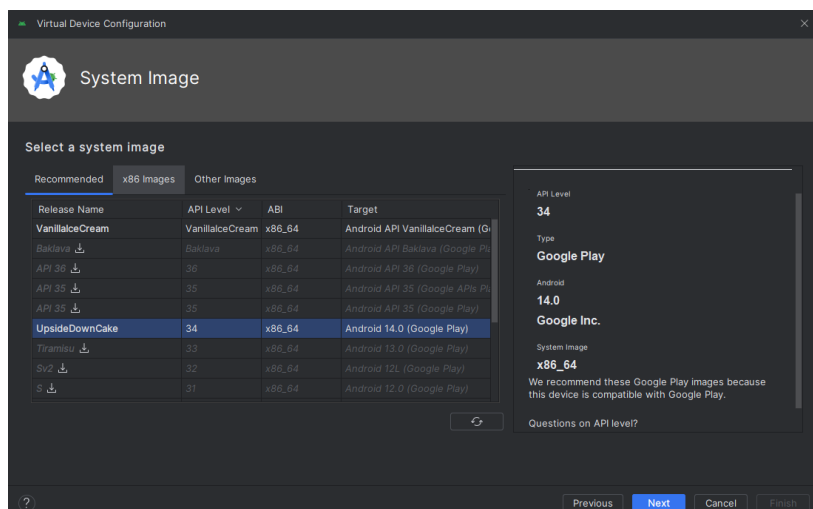


Figura D.14: Seleccionando la versión de Android del emulador

En la pantalla que vemos en la figura D.15 se deberá verificar la configuración del emulador y seleccionar el nombre que se desee. Una vez verificado, se dará al botón de “Next”.

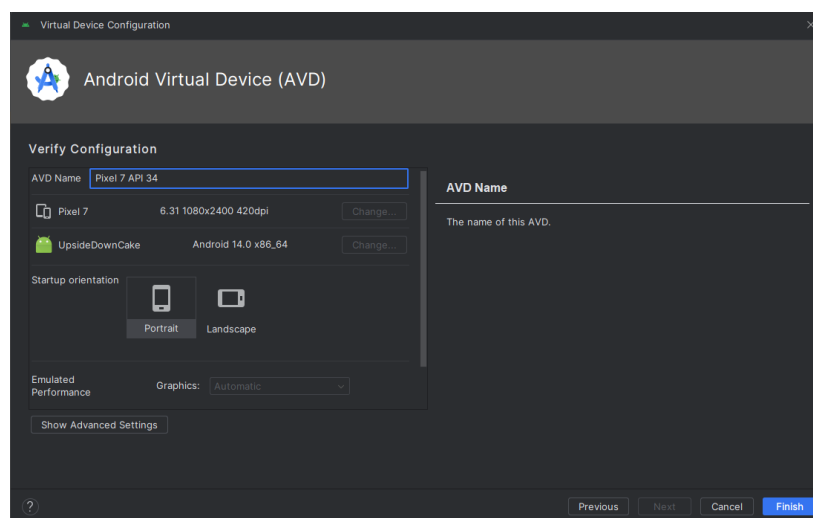


Figura D.15: Verificar la configuración del emulador

Descargado ya el emulador, tocará seleccionarlo como se muestra en la figura D.16 resaltado en azul. Una vez seleccionado, se empezará a ejecutar el emulador.

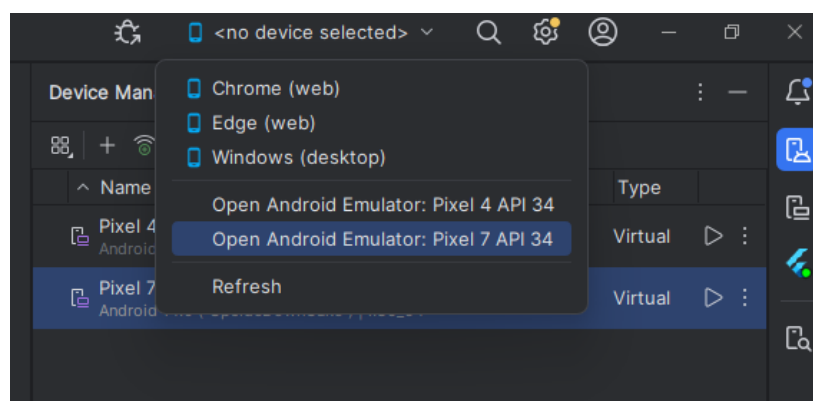


Figura D.16: Ejecución del emulador

Por último, una vez que el emulador ya ha iniciado, como vemos en la figura D.17, se deberá pulsar el botón verde de ejecutar de arriba a la izquierda que está rodeado en azul. Una vez pulsemos este botón, se comenzará a ejecutar el proyecto.

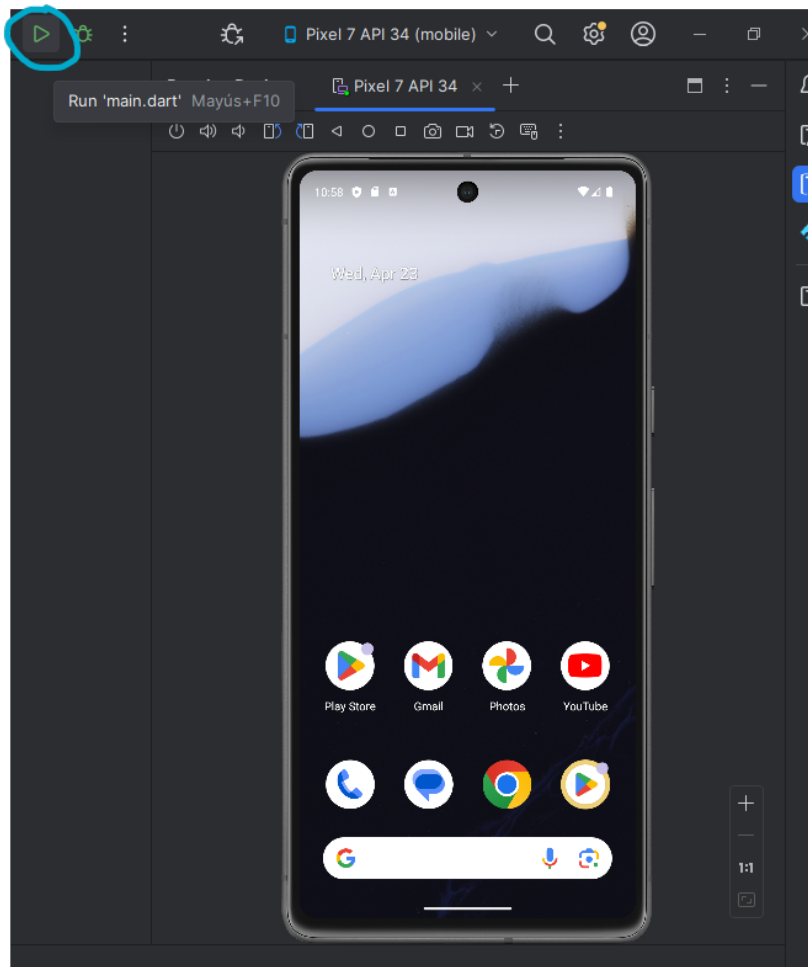


Figura D.17: Ejecución del proyecto

D.5. Pruebas del sistema

Para garantizar que la aplicación GreenInHouse 2.0 funciona de manera correcta, se han llevado a cabo diferentes tipos de pruebas:

Pruebas Funcionales:

Estas pruebas se basan en comprobar que cada una de las funcionalidades de los requisitos funciona de manera correcta, como por ejemplo:

1. **Crear Planta:** El sistema permite que el usuario pueda seleccionar el nombre y tipo de planta y crear la planta.

2. **Consultar Hitos:** La comunicación con la base de datos a través de la API hace que se puedan ver los hitos diarios.
3. **Cambiar Idioma:** El usuario selecciona el idioma que desee y automáticamente se cambia el idioma en toda la aplicación.
4. **Seleccionar Imagen de Planta:** El usuario puede sacar una foto con la cámara o seleccionarla de la galería.

Pruebas de Integración:

Estas pruebas se realizan comprobando la comunicación entre la aplicación y la base de datos mediante diferentes peticiones HTTP dentro de la red local. Estas peticiones pueden ser:

1. **GET:** Mediante esta llamada se realizan las solicitudes de lectura a la API como pueden ser la obtención del estado de los sensores.
2. **POST:** Mediante esta llamada se realizan las solicitudes a la API para la creación de nuevos datos en la base de datos, como puede ser la creación de una nueva planta.
3. **PUT:** Mediante esta llamada se realizan las solicitudes a la API para la modificación de datos ya creados, como puede ser la modificación del tipo de planta.
4. **DELETE:** Mediante esta llamada se realizan las solicitudes a la API para la eliminación de datos creados en la API, como puede ser la eliminación de una planta.

Pruebas de Interfaz de Usuario:

Este tipo de pruebas se basan en el correcto funcionamiento de la interfaz de usuario, como puede ser:

1. **Funcionamiento de botones y demás *widgets*:** Se han hecho pruebas para comprobar que tanto los botones como los demás elementos con los que puede interactuar el usuario de la interfaz funcionen de manera correcta.
2. **Adaptabilidad:** Se han hecho pruebas para comprobar que usando teléfonos móviles de diferentes resoluciones la interfaz se adapta de manera correcta.

Pruebas de robustez:

Se han hecho también pruebas simulando fallos para ver cómo reaccionaba la aplicación. Este tipo de pruebas han sido, por ejemplo:

1. **Sin conexión a internet:** Se han hecho pruebas para ver cómo reaccionaba la aplicación en caso de que no hubiera conexión a internet y por tanto no se pudiera comunicar la aplicación con la API. Se ha observado que la aplicación sigue funcionando solo que no ofrece dichos servicios que necesitan de los datos de la base de datos.
2. **Plantas con el mismo nombre:** Se han hecho pruebas intentando crear una planta con el mismo nombre que otra ya existente. Al intentar esta prueba la aplicación ha dado error debido a la existencia de una planta con ese mismo nombre.

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Apéndice F

Anexo de sostenibilización curricular

F.1. Introducción

Este anexo incluirá una reflexión personal del alumnado sobre los aspectos de la sostenibilidad que se abordan en el trabajo. Se pueden incluir tantas subsecciones como sean necesarias con la intención de explicar las competencias de sostenibilidad adquiridas durante el alumnado y aplicadas al Trabajo de Fin de Grado.

Más información en el documento de la CRUE https://www.crue.org/wp-content/uploads/2020/02/Directrices_Sostenibilidad_Crue2012.pdf.

Este anexo tendrá una extensión comprendida entre 600 y 800 palabras.

Bibliografía

- [1] Descarga de android studio. <https://developer.android.com/studio?hl=es-419>.
- [2] Descarga de flutter sdk. <https://docs.flutter.dev/get-started/install>.