

Parte 1: Portada.

Proyecto:

SISTEMA EXPERTO DE RESOLUCIÓN DE INCIDENCIAS EN EL CULTIVO DE PLANTAS. CASO DEL TOMATE CHERRY.



Foto: Cultivo tomates (*Cultivo Tomates Ventero*, 2013) con licencia Free Commons usage

Autor: FRANCISCO PICADO FERNÁNDEZ

Centro: IES CARMEN MARTIN GAITE - NAVALCARNERO

Año: 2024

Parte 2: Índice

Parte 1: Portada.....	1
Parte 2: Índice	2
Parte 3: Dedicatorias y agradecimientos.....	3
Parte 4: Introducción y descripción del proyecto.....	4
Interés del cultivo de plantas en el hogar	4
Intereses de los usuarios que quieren cultivar plantas en el hogar.....	5
Aplicaciones móviles relacionadas con el cultivo de plantas	5
Parte 5: Objetivos del proyecto y del PFC. Requisitos del sistema.	7
Objetivos del proyecto y del PFC	7
Alcance de los problemas a tratar en la aplicación	7
Requisitos del sistema.....	8
Parte 6: Marco conceptual y contextual.....	9
Arquitectura de la solución software	9
Descripción del sistema experto	9
Consideraciones sobre la gama de problemas a cubrir	10
Fases de cultivo.....	11
Parte 7: Metodología y organización	13
Planificación del proyecto y diagrama Gantt.....	13
Análisis DAFO.....	14
Viabilidad de Proyecto.....	14
Parte 8: Desarrollo	16
Base de Datos del servidor.....	16
Programación del servidor	17
Base de Datos del cliente	20
Programación del cliente.....	20
Relación Cliente-Servidor.....	23
Casos de uso.....	24
Diagrama secuencia del sistema experto	25
Recursos gráficos	25
Parte 9: Plan de pruebas	26
Parte 10: Resultados, conclusiones y vías de mejora.....	28
Resultados	28
Conclusiones	28
Mejoras	28
Parte 11: Anexos específicos.	30
Documentos de planificación.....	30
Manual de uso	32
Parte 12: Bibliografía.....	35

Parte 3: Dedicatorias y agradecimientos

Quiero agradecer a mi tutor Sergio San Martín, su asesoramiento durante todo el desarrollo del proyecto, las ideas, aportaciones, correcciones, paciencia y ánimos ante las dificultades que he ido enfrentando.

También dar las gracias a los Profesores del Centro IES Carmen Martín Gaité de Navacarnero, por las enseñanzas en los dos años de grado, y por facilitar a los alumnos un entorno de aprendizaje y de curiosidad ante el conocimiento.

Parte 4: Introducción y descripción del proyecto

Existe un interés creciente en los países occidentales por el consumo de productos agrícolas y ganaderos sin el uso de pesticidas, ni otros productos químicos empleados durante su cultivo y producción.

Muchas personas han comenzado a cultivar pequeñas cantidades de verduras y frutas de "huerta" en su casa empleando lo que se denomina huertos urbanos

No obstante, los cuidados que requieren estas plantas son más sofisticados que el simple riego y la exposición a la luz solar, por lo que dar respuesta a las incidencias que se van produciendo en el crecimiento de estas plantas acaba convirtiéndose en una pesadilla para el usuario que ha de navegar por internet en busca de páginas y blogs de consejos, sugerencias y soluciones o bien recurrir a las redes sociales en busca de grupos de interés, donde la información no es siempre muy fiable ni concreta, de modo que acaba recurriendo a la visita a los profesionales de las tiendas dedicadas a la jardinería.

La experiencia de cultivar en casa se convierte así en un “sumidero de tiempo”, buscando solución a los problemas que se nos presentan.

Por tanto, se plantea en este PFC la creación de un **sistema experto** que ofrezca respuestas a las posibles preguntas que se planteen a través del teléfono móvil.

Se trata de un sistema muy sencillo donde el usuario doméstico describa qué problema tiene su planta y el **sistema experto** le ofrezca potenciales soluciones en forma de tareas.

Como experiencia inicial para este PFC se ofrecen soluciones en la plantación del **Tomate Cherry**, complemento ideal, cuando está recién cosechado, para una succulenta ensalada casera.

Además del sistema experto se ofrece al usuario la posibilidad de pedir calendarios de tareas en función del tipo de cultivo y la fase actual de este. De esta manera el usuario se ahorra una gran cantidad de tiempo en la planificación de tareas y puede, directamente, pasar a la acción.

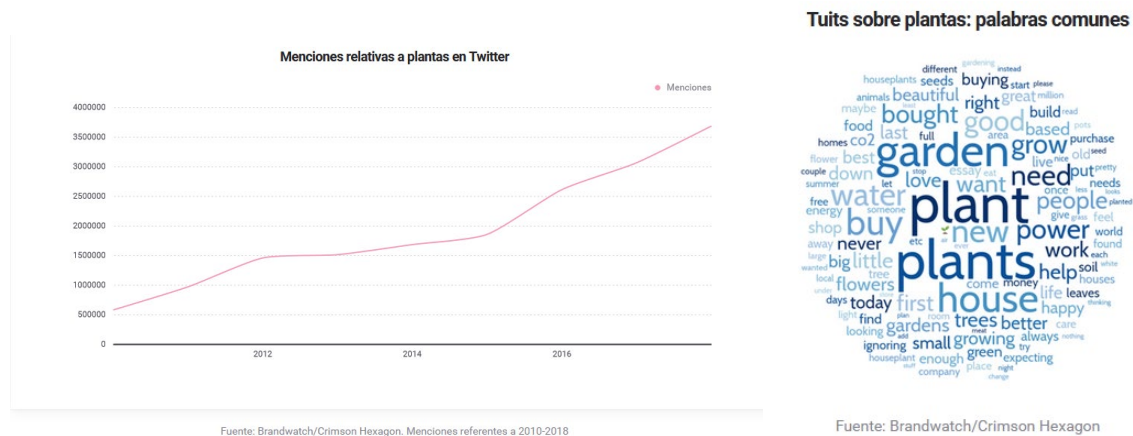
Interés del cultivo de plantas en el hogar

Entre los ciudadanos urbanos existe un boom de las huertas caseras que se impone desde terrazas y balcones como una forma sana y económica de proveer alimentos. El fenómeno arranca desde antes de la cuarentena del COVID, pero se ha afianzado después. (*Huertas Urbanas, Tendencia Que Gana Adeptos Por Cuarentenas Y Tiempos De Crisis*, 2021)

Además, según (*Los Huertos Urbanos Y La Huella Ecológica*, n.d.) “Se calcula que en el mundo actualmente hay 800 millones de personas involucradas en agricultura urbana. En Europa, Polonia es el país con más metros cuadrados de agricultura urbana por habitante (25,4 parcelas/1.000 hab.), Eslovaquia (16,3) y Alemania (12,3). España, a pesar del crecimiento experimentado en los últimos años, se encuentra lejos con 0,3 parcelas/1.000 hab.”

No obstante, en España según un estudio de Gregorio Ballesteros, del Grupo de Estudios y Alternativas GEA21, “el número de huertos urbanos supera los 15.000, en más de 300 municipios, con una superficie de más de millón y medio de metros cuadrados, unas 150 hectáreas o el equivalente a 150 campos de fútbol como el Santiago Bernabéu.”

Y fijándonos en las tendencia en internet, el interés de las plantas según menciones en twitter, (*Brandwatch/Crimson Hexagon. Menciones Referentes a 2010-2018, n.d.*):



Se relacionan como ejemplo diversas iniciativas Públicas que promueven el cultivo en el hogar, o en huertos urbanos o en huertos compartidos:

- Red de (*Huertos Urbanos*, n.d.) y escolares sostenibles en ciudades del Ministerio de Transición Ecológica, ej. Cabaña del retiro (Madrid), HuertAula en Cantarranas (Madrid), Red de huertos Agroecológicos (Zaragoza), Huertos escolares de Las Palmas de Gran Canaria, El Caminito (Málaga), etc,
- Huertos Comunitarios de Madrid (*Red De Huertos Urbanos De Madrid – La Red De Huertos Urbanos Comunitarios De Madrid*, n.d.)
- Huertas en Residencias de Mayores (*Huertas En Residencias*, n.d.).

Intereses de los usuarios que quieren cultivar plantas en el hogar

¿Qué beneficios aportan los huertos caseros?

- Disminuimos el gasto de compra de frutas y verduras.
- Proporciona acceso a alimentos frescos y orgánicos sin pesticidas, plaguicidas ni otros químicos.
- Potencia la capacidad de observación del medio natural, nos enseña sobre agricultura, nutrición y sostenibilidad a niños y adultos.
- Promueve el ejercicio físico y reduce el estrés.
- Fomenta un ecosistema urbano más diverso y apoya a polinizadores como abejas y mariposas.
- Mejora de la huella ecológica e incrementa la sensibilidad hacia la sostenibilidad

Extraído de (*Huerto Urbano: 9 Pasos Cómo Hacer Huerta En Casa Y Terrazas | SuperGuía*, n.d.) y (*Manual De Huertos Sostenibles En Casa*, n.d.)

Aplicaciones móviles relacionadas con el cultivo de plantas

La consulta del TOP 10 en la categoría Jardinería de Google Play, así como en webs especializadas (*Aplicaciones Móviles Para Plantas*, n.d.) y (Romero, n.d.) se resumen así:

Nombre de Aplicación	Des-cargas	Descripción	Funcionalidades clave
PictureThis Identificar Planta	50M	Colección de fotografías para identificar el nombre de la planta	No incluye ningún tipo de consejo sobre cultivo de plantas
Planta	1.2M	Ayudar en actividades de jardinería. Su funcionalidad incluye recordatorios para regar plantas, consejos personalizados según tipo de planta, y diario para seguir el progreso de tu jardín.	Refiere usar algoritmos de IA para asesoramiento personalizado, recordatorio de riego y cuidados
Waterbot	0.5M	Facilita la tarea de regar las plantas con recordatorios	Solo Riego, programa recordatorio de horas, duración y cantidad
Garden Organizer	100k	Planifica cultivos, sigue la plantación y dibuja las parcelas del jardín	Incluye recordatorios de riego
Sowing Calendar	100k	Guía de siembra y cosecha durante todos los meses del año	Incluye plagas y enfermedades
GreenThumb	100k	Identificar plantas, recibir consejos de cuidado, establecer recordatorios de riego y fertilización, y mucho más. Además, GreenThumb cuenta con una comunidad de jardineros aficionados para compartir experiencias	Monitoriza riego, luz, ventilación, etc. las respuestas a dudas provienen de la comunidad
GrowIt!	50k	Comunidad de amantes de la jardinería. compartir experiencias, conocimientos y hasta colgar fotos de tus plantas	Más que una aplicación es una pseudo red social
Curso de Jardinería	10k	Proporciona conocimientos acerca de especies vegetales y la realización de tareas de mantenimiento o construcción de jardines	Muy básica respecto a los cuidados y el riego. solo para verdaderos principiantes
Garden Plants Ideas	10k	100 ideas de decoración de jardín, selección de plantas y herramientas gráficas	No incluye ningún tipo de consejo sobre cultivo de plantas

Salvo **Planta**, las demás aplicaciones están dedicadas a aspectos muy concretos del cultivo o la jardinería. La gran mayoría no son totalmente gratuitas incluyendo funciones de “*compras en la aplicación*” para el acceso a todas las funcionalidades.

Por ello nuestro sistema experto es el único que evalúa los síntomas y diagnostica los problemas que aparecen, integrando a la vez las tareas habituales con las dirigidas a resolver esos problemas.

Además, el foco de la aplicación se pone en la sencillez de uso, ofreciendo información y acciones directas sobre el cultivo, sin necesidad de realizar consultas en otros sistemas, webs, libros o bases de datos.

Parte 5: Objetivos del proyecto y del PFC. Requisitos del sistema.

Este PFC comprende el análisis del reto planteado, la arquitectura del sistema cliente servidor, las bases de datos, así como la aplicación práctica para cultivar tomates cherry durante todo el ciclo de la planta, entre la siembra y la recolección de los frutos una vez madurados.

El usuario emplea un terminal android que plantea problemas y preguntas al sistema experto, y ofrece respuestas y sugerencias que son presentadas en el terminal móvil.

El sistema experto valora la información de los síntomas del problema y genera las respuestas basadas en el contenido de la base de datos aplicable a la planta en concreto.

Objetivos del proyecto y del PFC

El proyecto desarrollará una aplicación que permita al usuario emplear un terminal Android para plantear problemas y preguntas relacionadas con el cultivo al sistema experto, que ofrecerá respuestas y sugerencias que serán presentadas en el terminal móvil. Se buscarán los objetivos siguientes:

1. Sencillez de uso, para plantear los problemas y para recibir las respuestas.
2. Crear un sistema experto que ofrezca soluciones a los problemas más típicos durante el desarrollo de la planta, permitiendo además añadir más cultivos y problemas actuando sobre la base de datos.
3. El usuario podrá recibir calendarios de tareas.
4. Arquitectura del sistema y la base de datos con flexibilidad suficiente para añadir más cultivos y plantas al sistema.
5. Permitir al usuario crear cultivos y crear tareas para su cultivo aunque este no tenga soporte en el servidor.

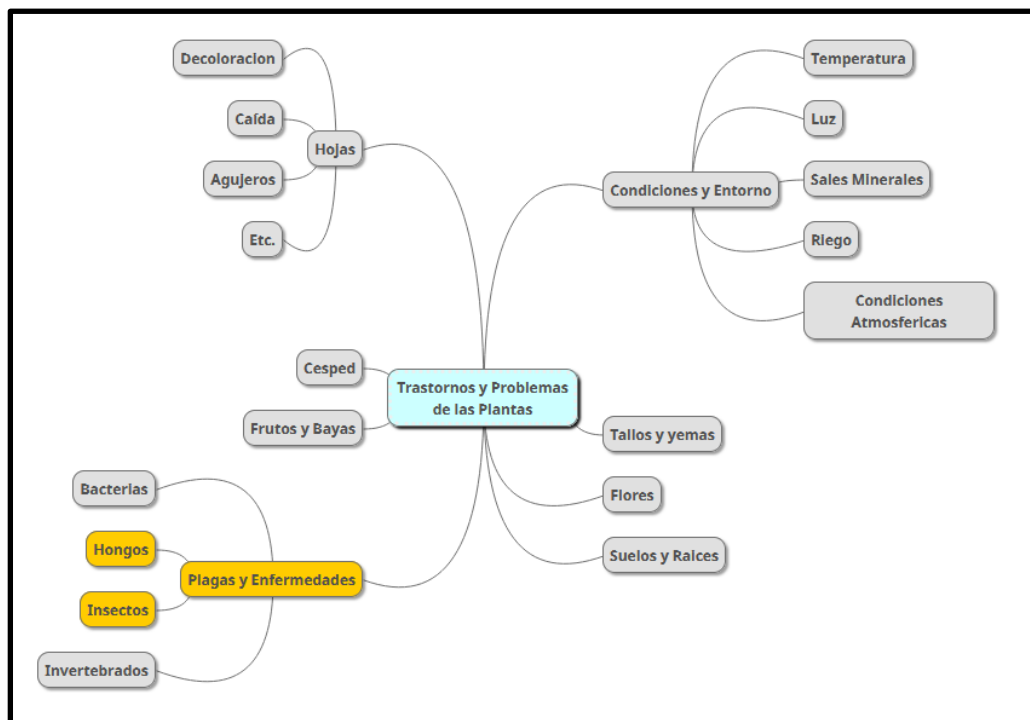
No obstante la carga lectiva correspondiente al PFC está limitada, por lo que se harán además las siguientes simplificaciones en el desarrollo del proyecto:

- No se contempla el empleo de IA o sistemas de aprendizaje en el sistema experto al exceder el tiempo asignado para completar este PFC
- Un único cultivo, el Tomate Cherry, dado que es una planta fácil de cultivar y vistosa para el usuario sin perder la capacidad de añadir más en futuro.
- Los grupos de problemas se reducen a dos: hongos e insectos.
- El protocolo de comunicación cliente-servidor será único, no universal.
- La plataforma móvil se desarrolla únicamente sobre Android

Alcance de los problemas a tratar en la aplicación

Este PFC no trata de constituirse en un tratado de agricultura sobre el cultivo del tomate cherry, por lo que haremos una simplificación basándose en los problemas más comunes encontrados en el cultivo del tomate y en otros cultivos, de modo que el software desarrollado permite el tratamiento de cualquier otra planta.

Así tendríamos este gráfico basado en los “Trastornos y Problemas de las plantas” (a efectos de la redacción de esta memoria, se nombrarán ambos como “Problemas” fuente: (*Jardineria Plantas Y Flores.com*, n.d.)



En este PFC se abordarán únicamente problemas relacionados con Plagas y Enfermedades.

Requisitos del sistema

Las pocas aplicaciones analizadas que proveen de sugerencias sobre riego o problemas en las plantas emplean la información residente en la propia aplicación.

No obstante, dada la naturaleza del problema que queremos resolver con un sistema experto, y de dotar al sistema de flexibilidad para añadir más plantas y funcionalidades extras en el futuro, entendemos que es apropiado la adopción del sistema experto en un servidor centralizado en lugar de residir en la aplicación móvil

Además, la transición de un sistema por porcentajes a uno basado en algoritmos de IA será siempre más sencillo en un modelo de servidor.

Por otra parte, Android es el sistema operativo de dispositivos móviles más popular en Europa y su entorno de desarrollo Android Studio forma parte del bagaje de conocimientos adquiridos durante el Módulo Superior del que forma parte este PFC, por lo que lo seleccionamos para la aplicación cliente.

Si bien podríamos trabajar sin necesidad de registrar al usuario, preferimos añadir la funcionalidad de modo que sirva en el futuro para crear perfiles de cultivos y ofrecer funciones o suscripciones adicionales.

Parte 6: Marco conceptual y contextual

El proyecto se enmarca inicialmente centrado en una arquitectura de la aplicación centrada en el objetivo de proporcionar soluciones a los problemas que el usuario encuentra en el cultivo de sus plantas.

Arquitectura de la solución software

La aproximación inmediata y también más simple, se centra en evaluar una aplicación de sistema experto en Android, usando Android Studio, Java y SQLite, que sea autosuficiente, no requiriendo interactuar con internet, más allá de las actualizaciones periódicas.

No obstante, evaluando las necesidades de capacidad en el sistema experto para gestionar problemas y cultivos futuros, tanto en rendimiento de la aplicación móvil, como del tamaño de la base de datos, se prefiere una solución cliente-servidor que facilite el rendimiento del cliente, así como traslade toda la potencia requerida por el sistema experto al Servidor

Esta arquitectura es más escalable que la inicial, y admite la gestión del login de usuario de cara a futuras ampliaciones de la funcionalidad y de monetizarlas a través de micro pagos o suscripciones

Al residir la base de datos en el servidor se facilita la actualización del catálogo de soluciones y el añadir nuevos cultivos a la aplicación.

Esta sería la propuesta final a desarrollar-

Opción	Cliente móvil	Servidor	Base datos (en servidor)
Planteamiento Inicial	Android Studio Java Kotlin SQLite	no requerido	no requerido
Decisión Final	Android Studio Java Kotlin SQLite	Java	MySQL

Descripción del sistema experto

El sistema experto planteado tiene la misión de agregar a la pregunta o consulta del usuario describiendo su problema, aquellos datos relevantes como la especie de planta, la fase de crecimiento, la existencia de hojas, estado del sustrato, etc, y evaluar cuál puede ser la posible solución o soluciones al problema, presentando en una lista de tareas a realizar.

Una descripción así, con una pregunta del usuario incluso en lenguaje natural podría prestarse al uso de sistemas de IA, si bien por la propia limitación de este PFC, vamos a plantearnos en su lugar la adopción de una de estas dos metodologías:

- **Sistema de Reglas:**
 - Basada en un listado de posibles factores de entradas cuyas combinaciones nos

ofrecen una lista de posibles salidas. Generalmente se trata de estructuras lógicas del tipo IF o WHILE.

- **Sistema de predicción por porcentajes**

- En este caso se aplica un peso o porcentaje de probabilidad a los diferentes factores o síntomas de entrada, de modo que el problema/solución más probable se calcula matemáticamente.

Durante el desarrollo me decanté finalmente por el sistema de porcentajes. En la parte del desarrollo del servidor se explica detalladamente esta decisión.

El usuario recibirá las respuestas en forma de un calendario de tareas, donde se detallan las actividades requeridas para cuidar la planta de un modo integral, tanto las relacionadas con el problema (aplicar insecticidas o fungicidas, fertilizantes, etc.), como aquellas habituales o regulares (exposición a luz solar, riego, temperatura).

El calendario será diario y el usuario tiene la posibilidad de marcar las tareas como completadas para reducir la lista de pendientes que muestra la aplicación.

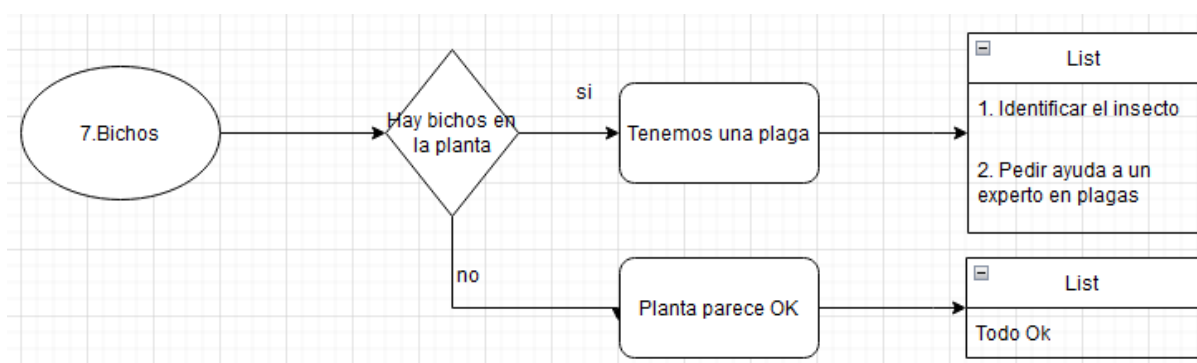
Consideraciones sobre la gama de problemas a cubrir

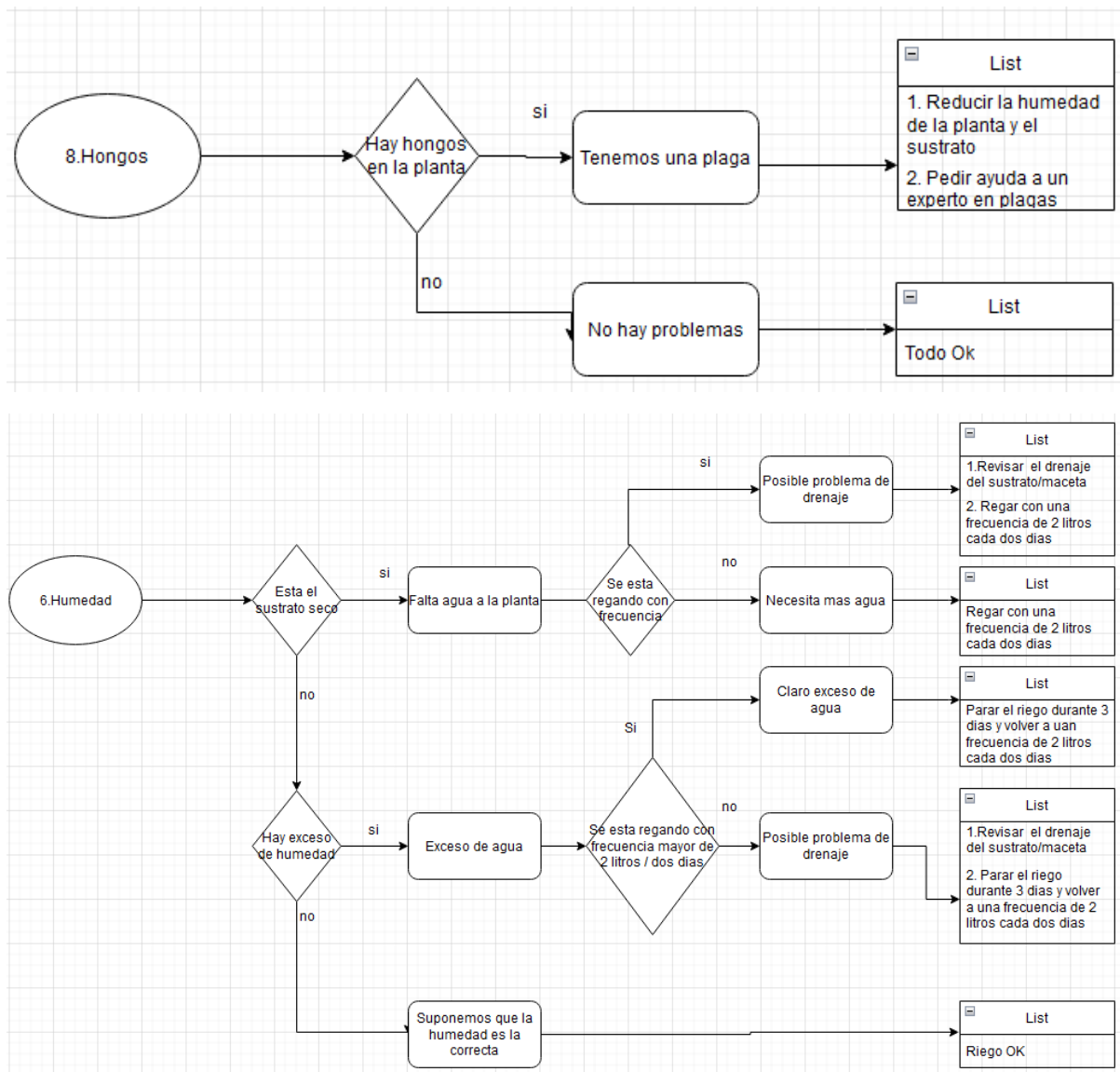
En la descripción realizada en el gráfico del Capítulo 5 basada en (*Jardinería Plantas Y Flores.com*, n.d.) se plantean hasta 8 áreas y fuentes de problema diferentes en el cultivo de plantas.

Por las limitaciones de este PFC se han seleccionado tres problemas dentro del Área de **Problemas y Enfermedades**, muy populares en cultivos, a incluir en el desarrollo a realizar, que serían:

- Enfermedades por Hongos
- Plagas de Insectos

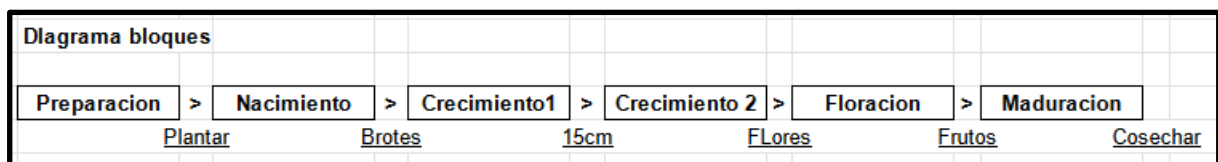
De manera simplificada se expone el grafo del sistema experto ante estos dos casos y, como ejemplo, ante problemas de Humedad si bien no ha sido incluido en el PFC:





Fases de cultivo

A efectos de estructurar adecuadamente la información en la base de datos y en las aplicaciones se consideran las siguientes fases e hitos.



En el cultivo del Tomate Cherry serían [basado en youtube (*CHERRY TOMATO From Seed to Harvest - Plant Time Lapse*, 2023)]:

- **Preparación:** del sustrato, maceta, semillero, etc, para plantar la semilla
- **Nacimiento:** Desde la siembra hasta el nacimiento de los primeros brotes.
- **Crecimiento 1:** Desde la aparición de los brotes hasta que la planta alcanza los 15 cm, donde se sugiere trasplantar al sustrato o maceta definitiva, si se había empleado una maceta pequeña o un semillero en la fase anterior

- **Crecimiento 2:** La planta continúa creciendo y llenándose de hojas, es momento de entutorar para facilitar el crecimiento posterior e incluso de podar.
- **Floración:** Esta fase arranca con la aparición de las primeras flores, hasta que salen los frutos
- **maduración:** El fruto crece hasta que transforma su color verde por el color de la maduración (rojo, amarillo, purpura, etc.) en función de la variedad de tomate plantado. es una fase delicada en la que hay que cosechar en su debido momento para que no se estropee el tomate.

A nivel de código de la aplicación vamos a condensarla en cuatro fases, con la vista en futuros cultivos y en darle la mayor simplificación al usuario:

- **Germinación:** incluyendo preparación y Nacimiento
- **Desarrollo,** uniendo las fases de crecimiento 1 y crecimiento 2
- **Floración,**
- **Maduración**

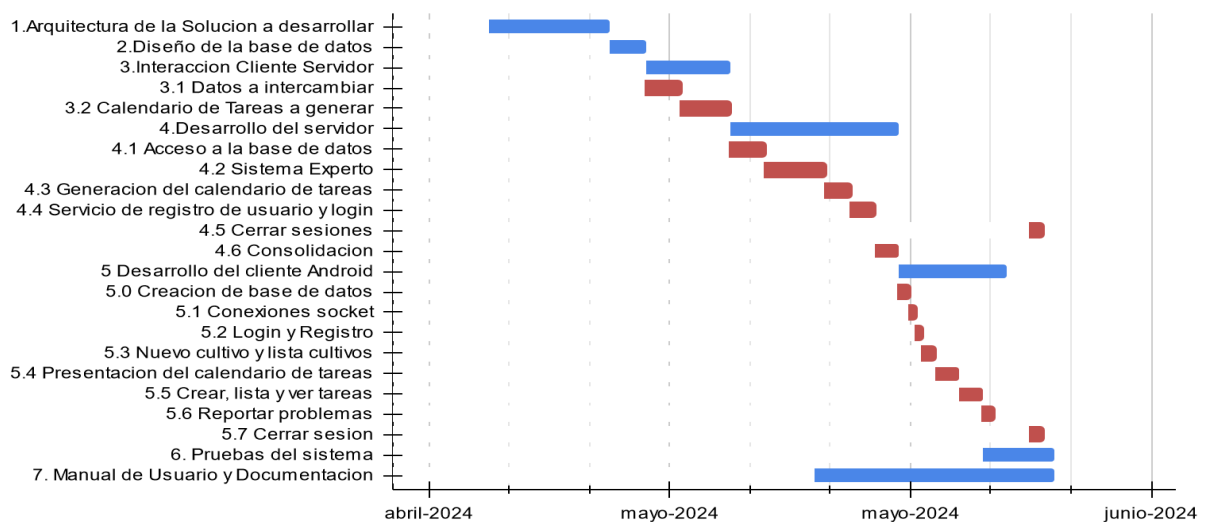
El usuario marcará el cambio de fase en la aplicación, lo que le permitirá recibir el calendario de tareas adecuado al momento vital del cultivo, dado que el Riego, Insolación y otras tareas serán diferentes dependiendo de la fase y de los problemas analizados.

Parte 7: Metodología y organización

Planificación del proyecto y diagrama Gantt

Se han seguido las siguientes fases y actividades principales:

Fecha	Actividad	Días Duración
23abr	1.Arquitectura de la Solución a desarrollar	10
3may	2.Diseño de la base de datos	3
6may	3.Interacción Cliente Servidor	7
6may	3.1 Datos a intercambiar	3
9may	3.2 Calendario de Tareas a generar	3
13may	4.Desarrollo del servidor	14
13may	4.1 Acceso a la base de datos	3
16may	4.2 Sistema experto	5
21may	4.3 Generación del calendario de tareas	2
23may	4.4 Servicio de registro de usuario y login	2
7jun	4.5 Cerrar sesiones	1
25may	4.6 Consolidación	2
27may	5 Desarrollo del cliente Android	9
27may	5.0 Creación de base de datos	1
28may	5.1 Conexiones socket	0.5
28may	5.2 Login y Registro	0.5
29may	5.3 Nuevo cultivo y lista cultivos	1
30may	5.4 Presentación del calendario de tareas	2
1jun	5.5 Crear, lista y ver tareas	2
3jun	5.6 Reportar problemas(Sistema experto)	1
7jun	5.7 Cerrar sesión	1
3jun	6. Pruebas del sistema	6
20may	7. Manual de Usuario y Documentación	20



Análisis DAFO

Del análisis de las Debilidades, Amenazas, Fortalezas y Oportunidades de la solución software planteada, se resume la información en la tabla siguiente:

Fortalezas	Oportunidades
<ul style="list-style-type: none"> • Gratuita, sin incluir anuncios. • Foco en el cultivo y sencillez uso. • Diseño flexible para admitir más cultivos y gestión de problemas. 	<ul style="list-style-type: none"> • Diseño que admite añadir funcionalidades por suscripción en un futuro. • Capacidad para añadir Comunidad de usuarios
Debilidades	Amenazas
<ul style="list-style-type: none"> • Dependencia de Internet, para interacción cliente-servidor • Fase inicial con un solo cultivo y pocos problemas cubiertos • Desarrollo por una sola persona y protocolo rígido y limitado. 	<ul style="list-style-type: none"> • Terminales con iOS excluidos • Solo en Español, no hay versión en Inglés.

Viabilidad de Proyecto

Viabilidad Técnica

En la Parte 6 de esta memoria se refiere el proceso seguido a la hora de seleccionar las herramientas de desarrollo de las aplicaciones. Dado que todas ellas son herramientas modernas contrastadas y con soporte proporcionado del fabricante entendemos que la viabilidad técnica del proyecto está asegurada.

Aplicación	Versión
Android Studio	Android Studio Hedgehog 2023.1.1
Java	servidor 1.8 aplicación 17.0.7
Kotlin	version 1.9.22
SQLite	Versión 3.22.0
MySQL	Release 8.0.33

Viabilidad económica

Consideremos las siguientes vías de Ingresos y de Costes de la aplicación

Ingresos por suscripción

- **Gratuita** para el Modelo BÁSICO (máximo de 3 cultivos activos, un cultivo nuevo al mes, publicidad integrada)

- Suscripción por Modelo PREMIUM (máximo de 5 cultivos activos, dos cultivos nuevos al mes, sin publicidad integrada)
- Suscripción por Modelo PLATINUM (cultivos ilimitados, sin límite de plantas añadidas al mes, sin publicidad integrada).
- Comisiones en venta de plantas y frutos entre usuarios registrados. Dados los costes de desarrollo de esta funcionalidad futura, no se han considerado ni ingresos ni gastos en el plan de viabilidad.

Estimación de crecimiento de usuarios de la aplicación.

Modelo	Anual	Usuarios Año 1	Usuarios Año 2	Usuarios Año 3
BÁSICO	gratuita	20.000	30.000	50.000
PREMIUM	6 eur	5.000	7.500	12.500
PLATINUM	10 eur	1.000	1.500	2.500

Respecto a los **costes** se considerarán:

- Costes de **Desarrollo**
 - Suscripción y Licencias de las Herramientas [herramientas gratuitas].
 - Mano de Obra de desarrollo los primeros tres años (*Infojobs. Salarios Desarrollador De Aplicaciones Madrid, n.d.*).
- Costes de **Operación**
 - Suscripción Google Play (*Google Play Tarifas, n.d.*).
 - Suscripción alojamiento servidor (*Hostiguer Tarifas Hosting 2024, 2023*).
 - Mano de obra de Mantenimiento, para los tres primeros años (*Infojobs. Salarios Desarrollador De Aplicaciones Madrid, n.d.*).

Resumen de Ingresos y Gastos

Modelo	Anual	Usuarios Año 1	Usuarios Año 2	Usuarios Año 3	Ingresos Año1	Ingresos Año2	Ingresos Año3	Ingresos3 años
BÁSICO	- €	20.000	30.000	50.000	- €	- €	- €	- €
PREMIUM (25% basico)	6,00 €	5.000	7.500	12.500	30.000,00 €	45.000,00 €	75.000,00 €	150.000,00 €
PLATINUM(5% basico)	10,00 €	1.000	1.500	2.500	10.000,00 €	15.000,00 €	25.000,00 €	50.000,00 €
				TOTALES	40.000,00 €	60.000,00 €	100.000,00 €	200.000,00 €
Desarrollo	Estimacion	Factor	Anual	Notas	Coste año 1	Coste año 2	Coste año 3	Coste 3 años
1.1 Licencias Herramientas	- €	12 meses	- €	gratuitas	- €	- €	- €	- €
1.2 Mano obra desarrollo	41.000,00 €	50%	26.650,00	30% cotiz	26.650,00 €	27.449,50 €	28.272,99 €	82.372,49 €
Operacion								
2.1 Cargos Google Play	facturacion	15%	15%	Tramo 1	6.000,00 €	9.000,00 €	15.000,00 €	30.000,00 €
2.2 Suscripcion Servidor	7,99 €	12 meses	95,88 €	Privado virtual	95,88 €	98,76 €	101,72 €	296,36 €
2.3 Mano obra Mantenimiento	41.000,00 €	50%	26.650,00	30% cotiz	26.650,00 €	27.449,50 €	28.272,99 €	82.372,49 €
Notas				TOTALES	- 59.395,88 €	- 63.997,76 €	- 71.647,69 €	- 195.041,33 €
30% costes empresa y cotizaciones sociales					Año 1	Año 2	Año 3	TOTAL 3 años
3% inflacion anual en costes				BALANCE	- 19.395,88 €	- 3.997,76 €	28.352,31 €	4.958,67 €

Para la estimación realizada, el proyecto sería rentable a partir del tercer año.

Parte 8: Desarrollo

Durante el desarrollo se ha utilizado Git como herramienta de control de versiones y GitHub como repositorio remoto. El desarrollo del proyecto lo empecé en la parte del servidor.

Base de Datos del servidor

Las bases de datos del servidor se han creado usando MySQL 8.0.33. La elección de esta base de datos es por su facilidad de uso y por mi experiencia en esta. El Modelo Relacional se describe a continuación dividido en tres partes:

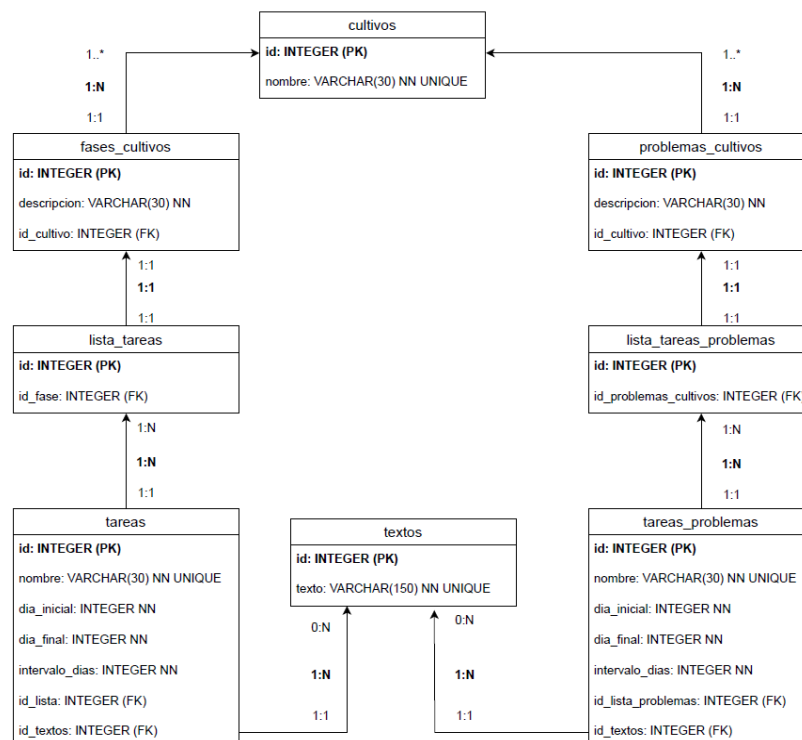
- Gestión de **Cultivos**
- Gestión de **Usuarios**
- Gestión de **Sesiones** de usuarios

La Gestión de **Cultivos**:

La base de datos “db_cultivos” utiliza las tablas “cultivos”, “fases_cultivos”, “lista_tareas”, “tareas” y “textos”. Cada fase de un cultivo tiene asociada una lista de tareas propia.

El formato de tarea dentro de la base de datos es el nombre, el día inicial en que se empieza a hacer la tarea, el día final dónde la tarea se hace por última vez y el intervalo de días de diferencia entre las repeticiones de las tareas. Por último, tiene una clave ajena de la tabla textos donde está la descripción de la tarea.

Al añadir el sistema experto se crearon tres tablas para los problemas que pueden enfrentar los cultivos. El formato de estas tablas es idéntico al de las otras. Como el sistema experto forma parte de la modalidad premium decidí crear tres nuevas tablas en vez de usar las que ya tenía para garantizar la integridad de los datos y evitar posibles trampas del usuario.



La gestión de **usuarios**:

Cree una base de datos “db_usuarios” con una tabla para los usuarios dónde la password es almacenada en hash (SHA-256) y codificada en Base64 para garantizar la seguridad del usuario.

usuarios
id: INTEGER (PK)
usuario: VARCHAR(30) UNIQUE NN
mail: VARCHAR(255)
passwd: VARCHAR(44) NN

El modelo relacional de la gestión de **sesiones**:

Para poder hacer las peticiones del sistema experto decidí crear un sistema de sesiones en una base de datos separada “db_sesiones” con una tabla de sesiones. La decisión de crear esta tercera base de datos en el servidor es para agilizar las peticiones al servidor. El servidor comprobará la sesión en la base de datos y el usuario no tendría que enviar continuamente su nombre de usuario y password.

sesiones
id: INTEGER (PK)
uuid: VARCHAR(36) UNIQUE NN
usuario: VARCHAR(30) NN

Programación del servidor

Herramientas:

Para la programación he utilizado Java 1.8 y el IDE Eclipse.

La elección de Java 1.8 se debe a mi experiencia en el diseño de servidores con este lenguaje y por su facilidad para manejar la concurrencia.

La librería “gson-2.10.1.jar” facilita la conversión de un string en formato json a objeto y viceversa. Siendo esta librería fundamental para los protocolos de comunicación.

Para la conexión con la base de datos es usado la librería “mysql-connector-j-8.0.33.jar”.

Programación:

El desarrollo del servidor se hizo módulo a módulo y luego consolidando todo al final.

La mejor forma de explicar el desarrollo es explicar las 5 peticiones básicas que el servidor puede recibir del cliente.

El servidor recibe las peticiones en formato string JSON. Gracias a la librería gson dicho string en json se puede convertir fácilmente en un objeto “Petición”.

El valor más importante de este objeto “Peticion” es el atributo tipo que indicará que quiere el usuario (“login”, “registro”, “cultivo”, “problema”, “cerrar”). Una vez identificada, la petición se ejecutará.

Petición Login y Registro

De las peticiones de tipo “login” y “registro” se encarga la clase “GestorUsuarios”, esta clase es la encargada de las operaciones de CRUD de “db_usuarios”. Los objetos de la clase “Operacion” contienen información que necesita la clase anterior para saber que tipo de operación se quiere realizar.

Los atributos obligatorios son “usuario” y “password”.

La “respuesta” esperada por el cliente es “OK” para Registro. Para Login “OK” y un valor para “sesión”. Dicho valor sesión lo genera la clase “GestorSesiones”.

Petición Cultivo

De las peticiones de tipo “cultivo” se utiliza la clase “ConectorBDCultivos” que devolverá las tareas del cultivo en el formato de la base de “db_cultivos” con la clase “TareasBD”. Luego estas tareas son convertidas a formato lista de “TareasDia” a través de “CreadorDeCalendario”. Los atributos obligatorios son “cultivo” y “fase”.

La “respuesta” esperada por el cliente es “OK” y una lista de tareas del atributo “calendario”.

Petición Problema.

Las peticiones de tipo “problema” son las relacionadas con el sistema experto. Primero se comprueba si la sesión del usuario es correcta en “GestorSesiones”, que tiene conexión con “db_sesiones”, y luego se aplica el sistema experto de hongos en insectos. “GestorSesiones” depende de la clase “Sesion” para poder aplicar las operaciones de CRUD correctamente.

El sistema experto recibe una serie de parámetros en formato String (“OK” y “NOK”), estos parámetros se procesan y se van sumando porcentajes en cascada. El diagnóstico será positivo si el resultado está por encima del 50%. Los “SistemaExpertoHongos” y “SistemaExpertoInsectos” han sido diseñados con una estructura que permite hacer una transición futura a un sistema experto basado en Reglas. Dado que el objetivo es una pequeña demo y el número de “inputs” de entrada es reducido se decidió durante el desarrollo hacer que el sistema experto se basará en porcentajes manteniendo la flexibilidad de un futuro sistema experto basado en reglas (siendo este sistema mejor para una mayor cantidad de inputs).

Los atributos obligatorios son “cultivo” y el atributo “problema” del objeto “PeticionProblema” que contiene los “inputs” que usará el sistema experto.

La respuesta esperada es la misma que para las peticiones cultivo.

Petición Cerrar Sesión.

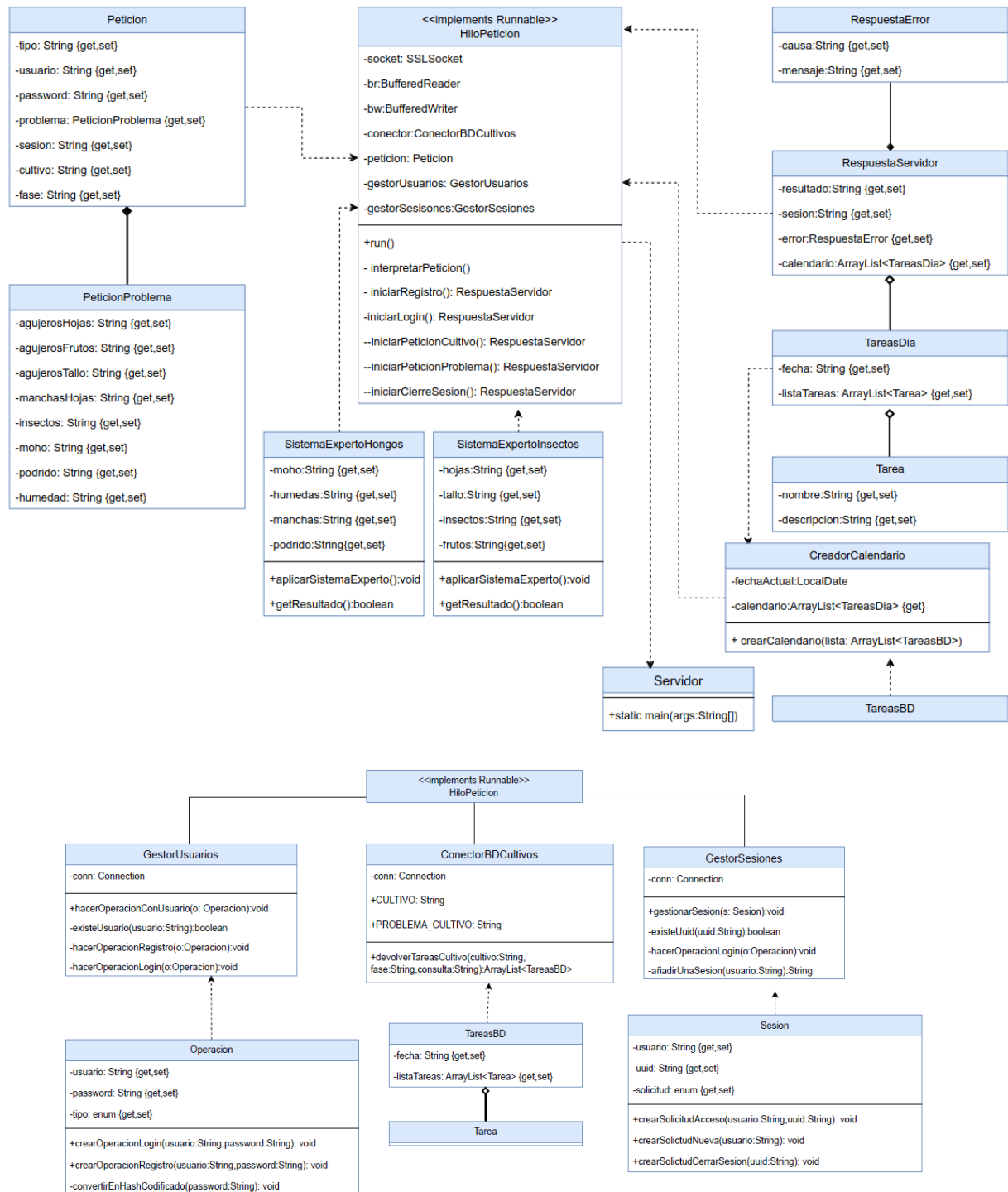
El valor de “tipo” es “cerrar”. El atributo obligatorio es “sesion”.

La respuesta esperada es “OK”.

Respuestas del Servidor.

La clase “RespuestaServidor” es la encargada de contestar al cliente dónde el atributo más importante es “respuesta”. Anteriormente se ha explicado cuáles son los mensajes esperados por el cliente para cada petición. En caso de petición fallida la respuesta será “NOK” acompañado del atributo error de la clase “RespuestaError” que contiene información de la causa del error y una pequeña descripción.

Diagrama de clases del servidor



Base de Datos del cliente

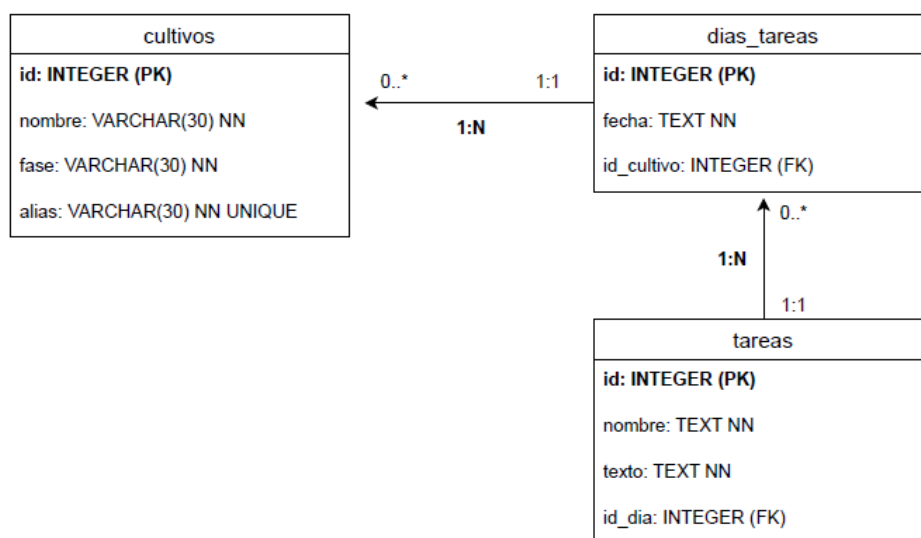
La base de datos del cliente se programa con SQLite de Android Studio.

SQLite es una base de datos ligera, fácil de usar y está bien integrada en los sistemas operativos Android.

Consta de tres tablas:

- cultivos: con el nombre del cultivo y un alias identificativo único.
- dias_tareas: almacena días que tienen tarea para un cultivo.
- tareas: son las tareas para un día de la tabla anterior.

Su Modelo Relacional es este:



Programación del cliente

Herramientas:

El cliente se ha desarrollado usando “Android Studio Hedgehog | 2023.1.1” y como lenguajes de programación Kotlin 1.9.22 y Java 17.0.7.

Java se ha usado para la lógica del programa y Kotlin para la parte gráfica.

La versión mínima SDK soportable es 28 y máxima es 34.

El teléfono más óptimo es el Nexus 5X aunque sirve para otros dispositivos Android.

Para gestionar las dependencias Gradle 8.2.

El cliente también usa la librería “`com.google.code.gson:gson:2.8.9`”

Programación:

El sistema de petición-respuesta es bastante parecido al servidor utilizando la misma estructura de clases “Petición”, “PeticiónProblema”(para el sistema experto), “RespuestaError” y “RespuestaServidor”. Además de otras clases como “Tareas” o “TareasDia”.

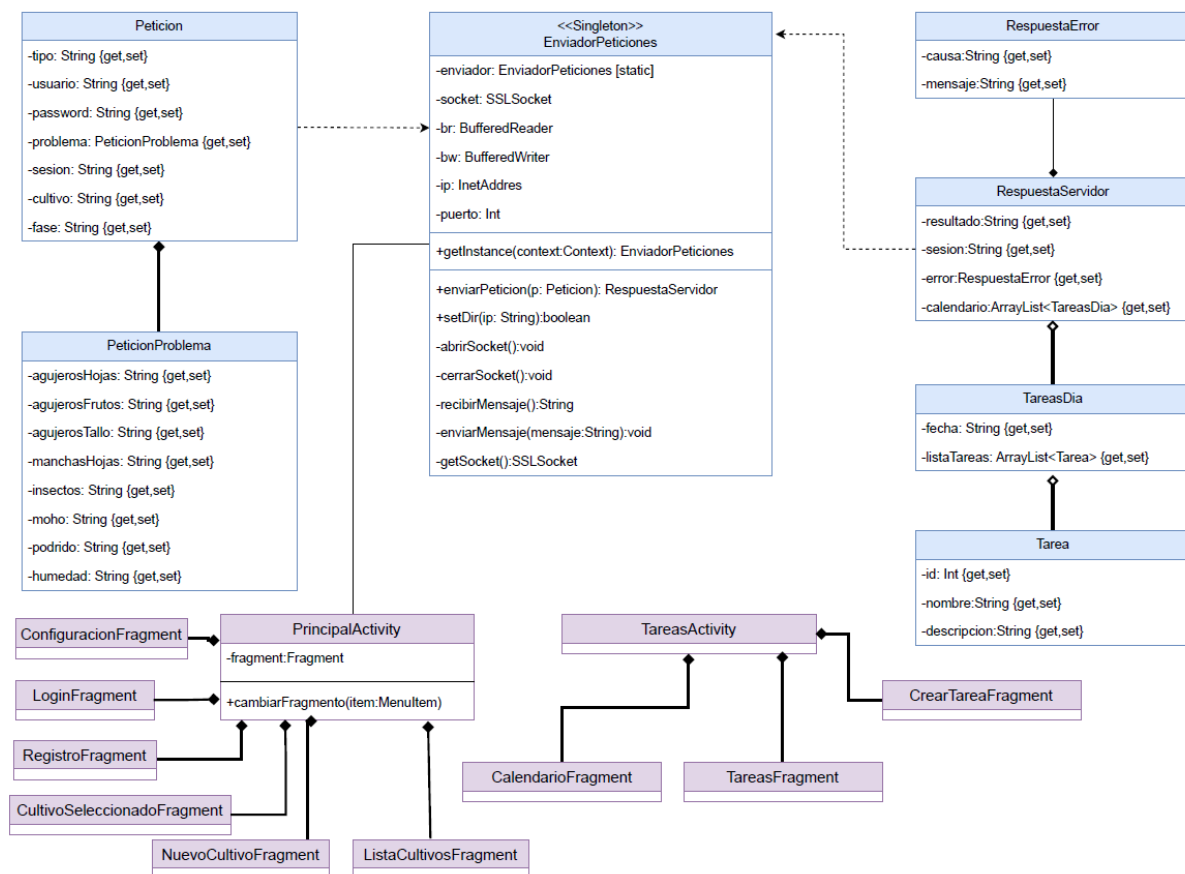
Lo más relevante son las clases “EnviadorPeticiones” y “GestorDeBaseDeDatos” que utilizan el patrón de “singleton”. Dónde la primera se encarga de crear el socket, enviar una petición y recibir la respuesta, y el segundo se encarga de realizar las operaciones CRUD en la base de datos que la aplicación demanda.

También hay que mencionar el uso de “recyclerview” para mostrar gráficamente las tareas, días del calendario o la lista cultivos.

La app consta de dos “activity” y varios fragmentos. Estos fragmentos acceden a “Enviar Peticiones” y “Gestor Base De Datos” a través de sus instancias de clase.

Gracias a las corrutinas puedo realizar la parte lógica sin comprometer al proceso principal, que es el encargado de la parte gráfica.

A continuación el diagrama de clases del cliente (el diagrama está simplificado pero manteniendo las clases y métodos más relevantes) dónde las clases en Java son azules y en morado las clases en Kotlin:





Relación Cliente-Servidor

La comunicación cliente-servidor se produce a través de SocketSSL. Se garantiza la seguridad en las comunicaciones usando un protocolo TLS.

La clave privada del servidor es almacenada en un almacén “pkcs12” mientras que la clave pública del cliente se almacena en “bks”.

Cliente y Servidor utilizan un protocolo JSON para comunicarse.

La estructura JSON de petición cliente es:

```
{
  "tipo": "problema",
  "usuario": "pepe",
  "password": "contraseña",
  "problema": {
    "agujerosHojas": "OK",
    "agujerosFrutos": "NOK",
    "agujerosTallos": "NOK",
    "manchasHojas": "NOK",
    "insectos": "OK",
    "moho": "NOK",
    "podrido": "OK",
    "humedad": "NOK"
  },
  "sesion": "csfd12-dsfsc2-sfs23-qyq2",
  "cultivo": "tomate cherry",
  "fase": "germinacion"
}
```

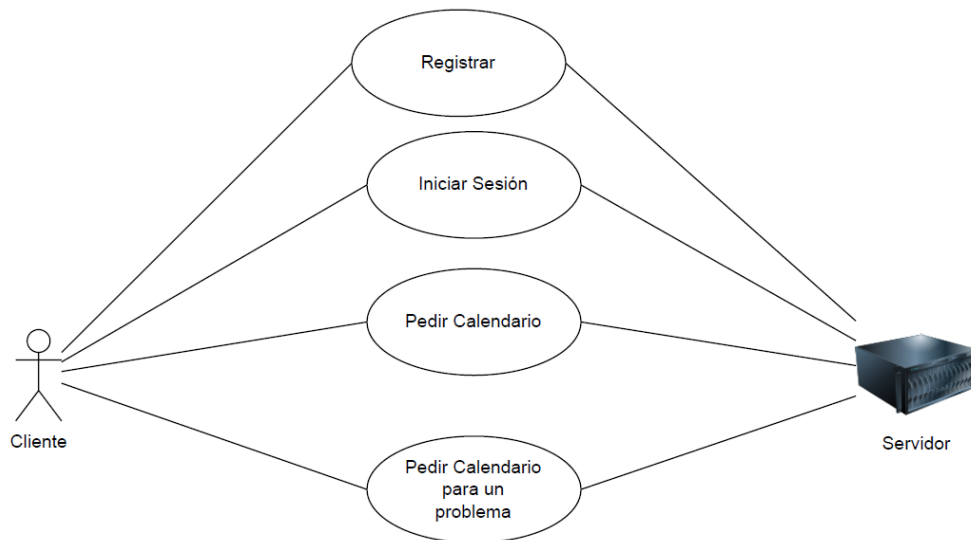
La respuesta del servidor sigue la estructura:

```
{
  "resultado": "NOK",
  "sesion": "csfd12-dsfsc2-sfs23-qyq2",
  "error": {
    "causa": "SERVIDOR",
    "mensaje": "VULEVA A INTENTARLO"
  },
  "calendario": [
    {
      "fecha": "22/05/2024",
      "listaTareas": [
        {
          "nombre": "riego",
          "descripcion": "Regar con 1 litro de agua"
        },
        {
          "nombre": "luz",
          "descripcion": "6 horas de iluminación."
        }
      ]
    }
  ]
}
```

A través de los siguientes diagramas se verá mejor como es la comunicación entre cliente-servidor.

Casos de uso

Se presentan los cuatro casos de uso más importantes del sistema cliente-servidor como ejemplo, ligados a la gestión de usuarios, sesiones, petición de cultivo y petición problema.



Nombre	Registrar
Actores	Cliente, Servidor
Propósitos	Registrar al cliente en el sistema "premium"
Precondiciones	El nombre de usuario y password cumplen los criterios
Flujo Normal	Se inserta en la bases de datos los datos del usuario
Flujo Alternativo	Ninguno
Postcondiciones	El nombre del usuario ya no estará disponible para un nuevo registro

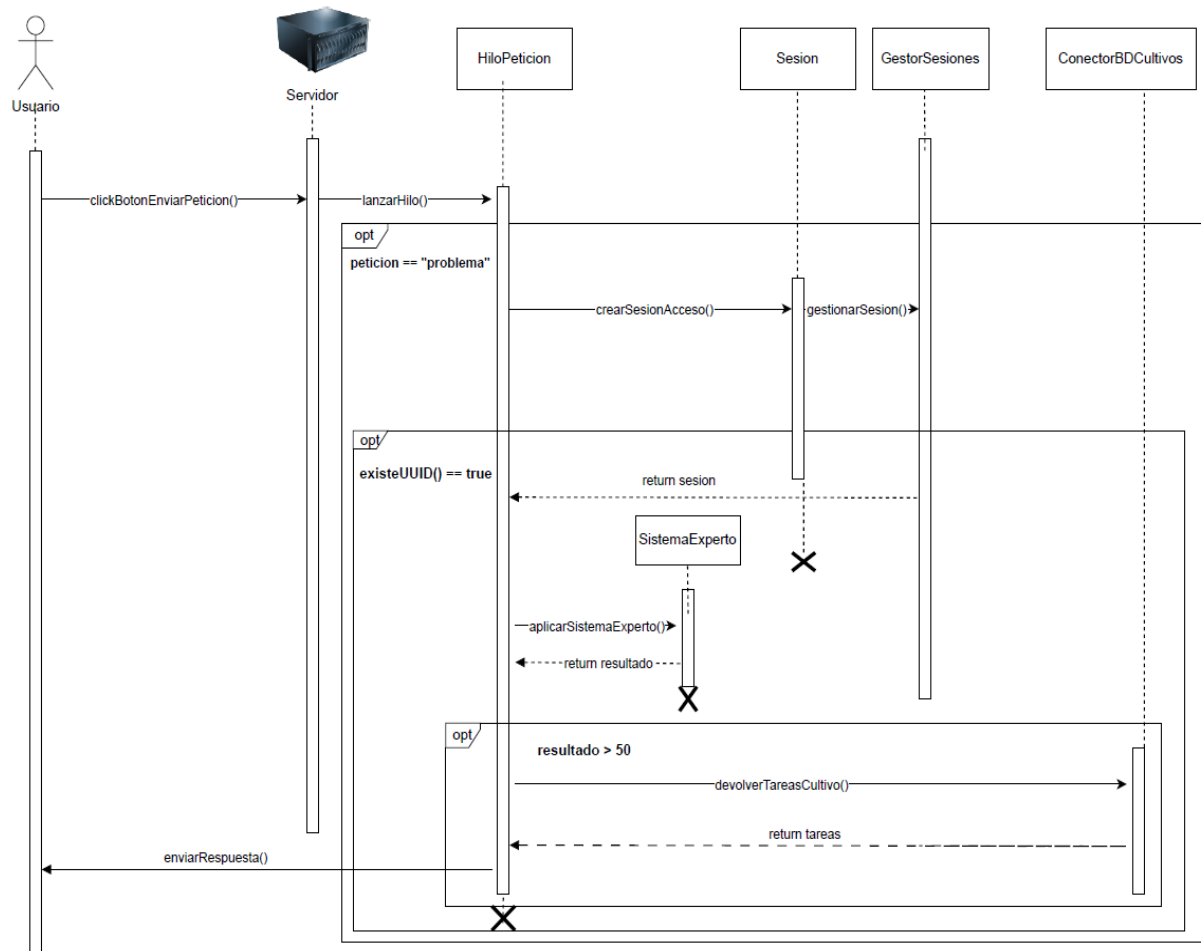
Nombre	Pedir Calendario
Actores	Cliente, Servidor
Propósitos	Enviar un calendario al cliente en función del cultivo
Precondiciones	Ninguno
Flujo Normal	Se devuelve una lista de tareas para ese cultivo
Flujo Alternativo	El cultivo no tiene soporte en la base de datos y no se devuelve un calendario
Postcondiciones	Ninguno

Nombre	Iniciar sesión
Actores	Cliente, Servidor
Propósitos	Devolver al cliente un id de sesión.
Precondiciones	El usuario está registrado
Flujo Normal	Se genera un id de sesión único y se responde al cliente
Flujo Alternativo	Fallo en el Login
Postcondiciones	Ninguno

Nombre	Pedir Calendario para un problema
Actores	Cliente, Servidor
Propósitos	Determinar si el cliente tiene un problema y devolver tareas para solucionarlo
Precondiciones	El cultivo tiene soporte y el id de sesión no ha expirado
Flujo Normal	Se devuelve una lista de tareas para ese problema
Flujo Alternativo	El cultivo no problemas
Postcondiciones	Ninguno

Diagrama secuencia del sistema experto

A continuación se presenta un diagrama secuencia para el caso de petición de un problema desde el cliente al servidor. En él se puede observar el funcionamiento del sistema experto. Está simplificado al no poder incluir todo en la memoria, pero la idea general se mantiene.



Recursos gráficos

Los iconos utilizados en la aplicación cliente se extraen de las páginas web siguientes:

Portal de iconos	URL
PNGWING	https://www.pngwing.com/
Icons8	https://icons8.com/
Flat Icon	https://www.flaticon.com/

Parte 9: Plan de pruebas

Aunque el plan de pruebas se definió inicialmente como de Caja Blanca y Negra, se realizaron cambios, a la vez que se avanzaba en la codificación de cada parte de la aplicación y se hacían pruebas inmediatas.

Al terminar la codificación y las pruebas del servidor se utilizó el programa generado para completar las pruebas del cliente, por lo que hay una mezcla de pruebas de caja blanca y de caja negra.

La parte de registro de usuario, login y cierre de sesión se indica aparte, puesto que es una funcionalidad añadida a la aplicación una vez iniciado el desarrollo.

1. Servidor (caja blanca y caja negra)
 - a. Accesos a la base de datos
 - b. Módulo de Cultivos: Interrogar y Presentar
 - c. Crear mensaje saliente en JSON
 - d. Registro de usuarios y sesiones
 - e. Pruebas contra simulación de cliente en Java
2. Cliente (caja blanca y caja negra)
 - a. Acceso a la base de datos local
 - b. Pruebas de crear cultivos, crear tareas, calendario, elegir problemas, etc.
3. Pruebas del sistema cliente-servidor (caja blanca y negra)
 - a. Funcionalidad de Registro de usuario y login
 - b. Función Crear Cultivos
 - c. Función Solicitar Tareas y Calendario. y Función Cambio de Fase
 - d. Función Preguntar por Problemas
 - e. Función Cierre de Sesión
 - f. Función de Gestión Errores de conexión entre cliente y servidor

Durante el desarrollo del servidor se hicieron pruebas módulo a módulo. Una vez consolidados todos los módulos se creó un cliente pequeño para hacer pruebas enviando diferentes JSON y comprobando los resultados obtenidos.

Durante el desarrollo del cliente se utilizó el servidor ya compilado para hacer las pruebas del cliente. Las pruebas de peticiones se hicieron contra el servidor. El Logcat y el debug ayudaron a la realización de las pruebas y la solución de errores.

En lo posible se han usado tablas de resultados como este ejemplo de las Pruebas del sistema cliente-servidor. Los datos de entrada-salida son una simplificación del JSON:

Prueba	Dato de Entrada	Salida Esperada	Salida Obtenida	Resultado
registro	tipo: "registro" usuario: "pepe" password: "holahola"	respuesta: "OK"	respuesta: "OK"	OK
login	tipo: "login" usuario: "pepe" password: "holahola"	respuesta: "OK" sesión: "afsy-12sj"	respuesta: "OK" sesión: "afsy-12sj"	OK

pedir tareas	tipo: "cultivo" cultivo: "tomate cherry" fase: "germinacion"	respuesta:"OK" calendario: "[...]"	respuesta:"OK" calendario: "[...]"	OK
pedir tareas problema	tipo: "problema" sesion: "afsy-12sj" cultivo: "tomate cherry" problema: {moho: "OK"}	respuesta:"OK" calendario: "[...]"	respuesta:"OK" calendario: "[...]"	OK
cerrar sesión	tipo: "cerrar" sesion: "afsy-12sj"	respuesta:"OK"	respuesta:"OK"	OK

Parte 10: Resultados, conclusiones y vías de mejora.

Resultados

Comparando con los objetivos propuestos y la visión del PFC desde el primer día:

- He diseñado y desarrollado una aplicación completa y funcional, incluso sin conexión al servidor.
- Se puede dar soporte a cultivos adicionales solo añadiendo las tareas y problemas en las bases de datos del cliente y del servidor, sin cambios en el código.
- Se añadió la funcionalidad de registro y gestión de usuarios, no planteada al principio del desarrollo, pero que proporcionará ingresos, darle una viabilidad económica al proyecto con los tres niveles de suscripción y añadirle mayor funcionalidad en el futuro.
- Y lo más importante, la aplicación es sencilla de usar para el usuario centrándose su gestión del cultivo en seguir las tareas que ofrece la aplicación, sin otras distracciones ni necesidad de acudir a buscar información a otro sitio.

Conclusiones

Mis conclusiones personales en la creación de este PFC son:

- He puesto en práctica los conocimientos adquiridos durante el curso:
 - Creación y acceso a Bases de Datos,
 - Uso de Sockets.
 - Diseño de servidores.
 - Sistemas de cifrado y seguridad.
 - Uso de protocolos.
 - Diseño de aplicaciones Android.
 - Programación orientada a objetos.
 - Programación concurrente.
 - Uso de lenguajes de marcas.
 - Aplicación de conocimiento de entornos de desarrollo.
- Me he dado cuenta de lo importante que es planificar y analizar minuciosamente las necesidades de la aplicación y sus funciones, antes de ponerse a escribir líneas de código, pues el tiempo para modificar y cambiar las cosas una vez programadas es muy grande.
- He tenido que emplear bastante tiempo para documentarme en el modo que hay que cultivar las plantas y entender cómo encontrar un modelo común para que el cultivo de plantas diferentes pudiera ser gestionado por el mismo código de cliente servidor y bases de datos.
- Importancia de la planificación inicial para la futura implantación del sistema.

Mejoras

Siempre hay aspectos que pueden ser mejores. Destacar estos:

- En la aplicación, modificar, corregir, probar lleva mucho tiempo y no se le han podido prestar atención suficiente a mejorar el aspecto visual de la aplicación.

- La documentación del proyecto hay que tenerla en cuenta desde el mismo día que se comienza a diseñar, pues los días finales no hay tiempo material para revisar o añadir.
- Añadir un sistema de recordatorios para las tareas.
- Apostar por “frameworks” que facilitan el diseño de servidores, de esta manera ahorro horas en el diseño de funcionalidades básicas como el Registro, Login, etc.
- Darle más funcionalidades a un usuario registrado.
- Añadir la funcionalidad que permite a los usuarios premium vender productos entre ellos.

Parte 11: Anexos específicos.

Documentos de planificación.

Una vez decidida la estructura cliente-servidor de la aplicación, he tenido tres momentos principales en la planificación, todos ellos tras la revisión-discusión con el Tutor de PFC. He puesto color verde en las actividades nuevas o con cambios importantes respecto a la versión revisada. Las duraciones inicialmente estimadas han tenido que corregirse casi siempre

- Planteamiento inicial

Fecha	Actividad	Días Duración
23abr	1.Arquitectura de la Solución a desarrollar	5
28abr	2.Diseño de la base de datos	5
3may	3.Interacción Cliente Servidor	7
3may	3.1 Datos a intercambiar	3
6may	3.2 Calendario de Tareas a generar	5
13may	4.Desarrollo del servidor	14
13may	4.1 Acceso a la base de datos	3
16may	4.2 Sistema experto	7
23may	4.3 Generación del calendario de tareas	4
27may	5 Desarrollo del cliente Android	8
27may	5.1 Conexiones socket	1
28may	5.2 Presentación del calendario de tareas	5
2jun	5.3 Reportar problemas	2
4jun	6. Pruebas del sistema	2
1jun	7. Manual de Usuario y Documentación	5

- Primera Revisión

Se añade la funcionalidad de Login y Registro de usuario y adelantó el comienzo de la documentación del PFC para no dejarla para el final.

Fecha	Actividad	Días Duración
23abr	1.Arquitectura de la Solución a desarrollar	10
3may	2.Diseño de la base de datos	3
6may	3.Interacción Cliente Servidor	7
6may	3.1 Datos a intercambiar	3
9may	3.2 Calendario de Tareas a generar	3
13may	4.Desarrollo del servidor	14
13may	4.1 Acceso a la base de datos	3
16may	4.2 Sistema experto	5
21may	4.3 Generación del calendario de tareas	2
23may	4.4 Servicio de registro de usuario y login	2

25may	4.5 Consolidación	2
27may	5 Desarrollo del cliente Android	8
27may	5.1 Conexiones socket	1
28may	5.2 Login y Registro	1
29may	5.3 Cliente principal de cultivos	2
31may	5.4 Presentación del calendario de tareas	2
2jun	5.6 Reportar problemas	2
4jun	6. Pruebas del sistema	3
24may	7. Manual de Usuario y Documentación	15

- Segunda Revisión:

Será la revisión final, sin cambios hasta la entrega del PFC. Ya incluye más trabajo en la generación de tareas y presentación del calendario y de las tareas de modo más simple. La función de seleccionar IP del servidor y cerrar sesiones. Ampliación de la actividad de documentación

Fecha	Actividad	Días Duración
23abr	1.Arquitectura de la Solución a desarrollar	10
3may	2.Diseño de la base de datos	3
6may	3.Interacción Cliente Servidor	7
6may	3.1 Datos a intercambiar	3
9may	3.2 Calendario de Tareas a generar	3
13may	4.Desarrollo del servidor	14
13may	4.1 Acceso a la base de datos	3
16may	4.2 Sistema experto	5
21may	4.3 Generación del calendario de tareas	2
23may	4.4 Servicio de registro de usuario y login	2
7jun	4.5 Cerrar sesiones	1
25may	4.6 Consolidación	2
27may	5 Desarrollo del cliente Android	9
27may	5.0 Creación de base de datos	1
28may	5.1 Conexiones socket	0.5
28may	5.2 Login y Registro	0.5
29may	5.3 Nuevo cultivo y lista cultivos	1
30may	5.4 Presentación del calendario de tareas	2
1jun	5.5 Crear, lista y ver tareas	2
3jun	5.6 Reportar problemas(Sistema experto)	1
7jun	5.7 Cerrar sesión	1
3jun	6. Pruebas del sistema	6
20may	7. Manual de Usuario y Documentación	20

Manual de uso

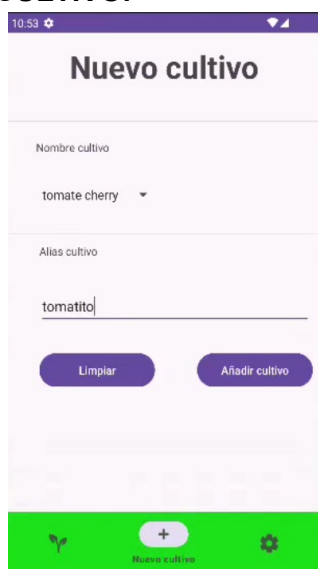
Primero crear las bases de datos MySQL es importante que el usuario sea “root”, la password sea “123qwe” y el puerto de MySQL sea 3306.

Segundo se ejecuta el servidor con el comando “java -jar servidor.jar”.

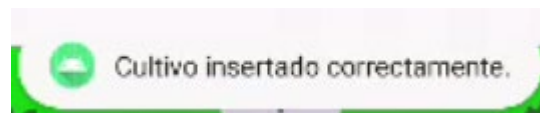
Acceso a la aplicación y Crear primer cultivo

Abrir la aplicación marcando en el icono. Se presenta una pantalla con el listado de los cultivos disponibles. Sin embargo, la primera vez no habrá cultivos creados y pulsaremos en el icono del signo **+** para crear uno.

Seleccionamos el cultivo deseado en la lista de **Nombre de cultivo**, p.ej TOMATE CHERRY y en la caja de debajo añadimos un **alias**, p.ej. TOMATITO, y pulsamos el botón **AÑADIR CULTIVO**.

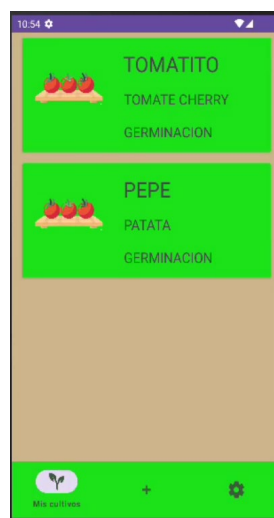


El sistema nos devuelve un mensaje de que se añadió correctamente:



Gestión de Cultivos

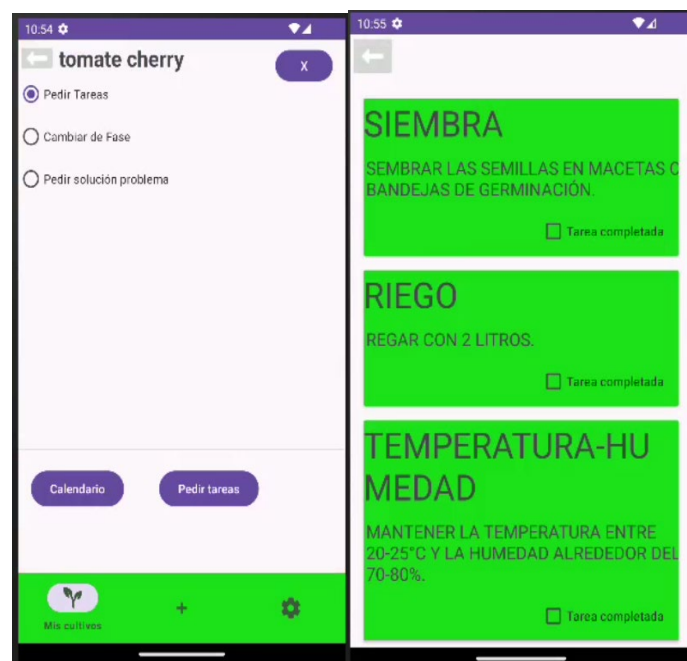
Desde la pantalla principal hacemos click en el icono **MIS CULTIVOS** nos aparece un listado de cultivos ya creados identificándose por su **alias**, la **planta a cultivar** y la **fase** del cultivo. En este ejemplo tenemos dos cultivos.



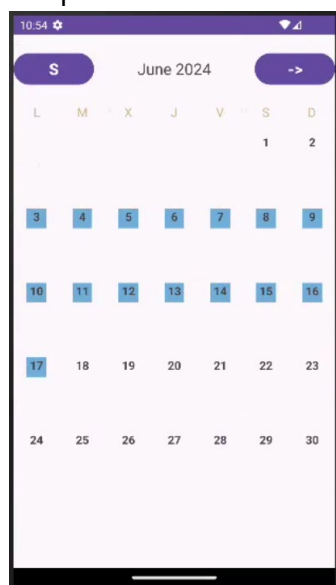
Pulsando sobre el primero identificado en el ejemplo como TOMATITO, accedemos a la lista de opciones donde hay tres acciones definidas:

- **Pedir Tareas:** que nos devuelve la lista de tareas a realizar hoy
- **Cambiar de Fase:** indicando a la aplicación que el cultivo se mueve a la fase siguiente (**Germinación, Desarrollo, Floración, Maduración**). Este aspecto es primordial, al ser diferentes las necesidades de insolación, riego y temperatura en cada fase.
- **Pedir solución a un problema:** Que presenta un formulario con una lista de síntomas a seleccionar y con las que el sistema experto nos ofrecerá una lista de tareas para solucionarlo.

Pulsando en el botón de acción **PEDIR TAREAS**, la aplicación nos devolverá una lista de tareas a realizar.



Otra opción es marcar Calendario pudiendo revisar las tareas pendientes de los días anteriores. Pulsamos el día deseado para revisarlas



Cada tarea tiene una pequeña descripción con la acción a realizar. Marcando sobre la casilla de **completada**, la tarea desaparece de la lista.



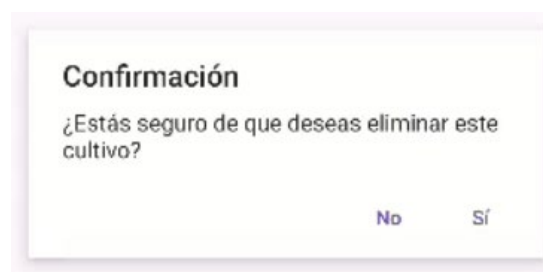
Añadir Tareas

El usuario puede añadir manualmente nuevas tareas pulsando en el NUEVA TAREA y añadiendo la descripción y la fecha.



Eliminar cultivos

Desde la pantalla del cultivo pulsar el botón borrado en la parte superior derecha y confirmar:



FIN DEL MANUAL DE USUARIO

Parte 12: Bibliografía

- Aplicaciones Mviles para plantas*. (n.d.). Retrieved May 29, 2024, from <https://idplantae.com/aplicaciones-recursos-y-herramientas/aplicaciones-mviles-para-plantas-comprobadas-y-confiables/>
- Brandwatch/Crimson Hexagon. Menciones referentes a 2010-2018*. (n.d.). Wikipedia. Retrieved May 29, 2024, from <https://brandwatch.com/es/blog/plantas-hogar/>
- CHERRY TOMATO from Seed to Harvest - Plant Time Lapse*. (2023, August 23). YouTube. Retrieved June 6, 2024, from <https://www.youtube.com/watch?v=rLW8pxwXZuM>
- Cultivo tomates ventero*. (2013, enero 15). Homo Agrícola- 15ene2013. Retrieved June 9, 2024, from <https://elhocino-adra.blogspot.com/2013/01/mucho-por-poco.html>
- Google Play Tarifas*. (n.d.). Información sobre los cargos del servicio de Google Play - Ayuda de Play Console. Retrieved June 5, 2024, from <https://support.google.com/googleplay/android-developer/answer/11131145?hl=es-419>
- Hostiguer Tarifas Hosting 2024*. (2023, December 20). Hostinger. Retrieved June 5, 2024, from https://www.hostinger.es/tutoriales/cuanto-cuesta-un-hosting#Hosting_compartido
- Huertas en Residencias*. (n.d.). Retrieved May 29, 2024, from <https://artsandculture.google.com/story/cQXhTPEsZMfELQ?hl=es-419>
- Huertas urbanas, tendencia que gana adeptos por cuarentenas y tiempos de crisis*. (2021, May 16). El País de Cali. Retrieved May 29, 2024, from <https://www.elpais.com.co/familia/huertas-urbanas-tendencia-que-gana-adeptos-por-cuarentenas-y-tiempos-de-crisis.html>
- Huertos urbanos*. (n.d.). Ministerio para la Transición Ecológica y el Reto Demográfico. Retrieved May 29, 2024, from <https://www.miteco.gob.es/es/ceneam/recursos/pag-web/huertos-urbanos.html>
- Huerto urbano: 9 Pasos cómo hacer huerta en casa y terrazas | SuperGuía*. (n.d.). OVACEN. Retrieved May 29, 2024, from <https://ovacen.com/huerto-urbano/>
- Infojobs. Salarios Desarrollador de aplicaciones Madrid*. (n.d.). Retrieved June 5, 2024, from <https://salarios.infojobs.net/desarrollador-de-software-de-aplicaciones/madrid>
- Jardineria Plantas y Flores.com*. (n.d.). Retrieved May 29, 2024, from <https://jardineriaplantasyflores.com/problemas-plantas>
- Los huertos urbanos y la huella ecológica*. (n.d.). Fundación Aquae. Retrieved May 29, 2024, from <https://www.fundacionaquae.org/wiki/huertos-urbanos-una-manera-contribuir-disminuir-la-huella-ecologica/amp/>
- Manual de huertos sostenibles en casa*. (n.d.). Ministerio para la Transición Ecológica y el Reto Demográfico. Retrieved May 29, 2024, from <https://www.miteco.gob.es/es/ceneam/recursos/pag-web/manual-huertos-sostenibles-casa.html>
- Ministerio para la Transición Ecológica y el Reto Demográfico. (n.d.). *Manual de huertos sostenibles en casa*. Ministerio para la Transición Ecológica y el Reto Demográfico. Retrieved May 29, 2024, from <https://www.miteco.gob.es/es/ceneam/recursos/pag-web/manual-huertos-sostenibles-casa.html>
- Red de Huertos Urbanos de Madrid – La red huertos urbanos comunitarios Madrid*. (n.d.). Retrieved May 29, 2024, from <https://redhuertosurbanosmadrid.wordpress.com/>
- Romero, H. (n.d.). *11 mejores app de jardinería para cuidar plantas (2023)*. Retrieved May 29, 2024, <https://www.lavanguardia.com/andro4all/aplicaciones/apps-de-jardineria>