

UNIVERSIDAD DE TARAPACÁ



FACULTAD DE INGENIERÍA

Departamento de Ingeniería en Computación e Informática



Plan de proyecto
**BotaniDrip: Sistema de riego automatizado,
remoto e intuitivo para casas inteligentes con
enfoque asistencial**

Autor(es): Patricio Chang

Francisco Pantoja

Hernan Vazque

Asignatura: Proyecto II

Profesor y Académico: Diego Aracena P.

ARICA, 28 de noviembre de 2023

Historial de Cambios

| Fecha | Versión | Descripción | Autor(es) |
|--------------|----------------|--------------------------------|--|
| 12/09/2023 | 1.0 | Versión preliminar del formato | Patricio Chang |
| 23/09/2023 | 1.1 | Terminar informe 1 | Patricio Chang Francisco Pantoja Hernan Vazque |
| 21/11/2023 | 2.1 | Corrección Informe 1 | Francisco Pantoja |
| 26/11/2023 | 2.2 | Terminar Informe 2 | Patricio Chang Francisco Pantoja |

Tabla de contenidos

1. Panorama General
 - Introducción
 - Propósito, alcance, objetivos
 - Entregables del proyecto
 - Suposiciones y restricciones
 - Entregables
2. Organización del Proyecto
 - Personal y entidades internas
 - Roles y responsabilidades
 - Mecanismos de Comunicación
3. Planificación de los procesos de gestión
 - Planificación de estimaciones
 - Planificación de recursos humanos
 - Riesgos
 - Factores de Riesgo
4. Planificación de Actividades (Carta Gantt)
5. Planificación de Procesos Técnicos
 - Modelo de Proceso
 - Modelo de Diseño (Caso de Uso general)
 - Casos de Uso
 - Diagramas de Secuencia
 - Descripción de la arquitectura
6. Diseño de interfaces de Usuario
7. Conclusiones
8. Referencias

1. Panorama General

1.1.1 Introducción

El pasatiempo de regar y cuidar plantas tiene un impacto positivo en adultos mayores, algunos de sus beneficios son el prevenir enfermedades mentales y mejorar el estado emocional. El proyecto Botani Drip busca acercar este pasatiempo a personas con problemas de movilidad reducida o pertenecientes al grupo etario de la tercera edad. Gracias al auge de la tecnología en la vida cotidiana de las personas, este tipo de proyectos es viable e implementable, dando soluciones concretas a problemas específicos.

1.1.2 Propósito, alcance, objetivos

Propósito:

El propósito de este proyecto es lograr que personas pertenecientes a la tercera edad y/o problemas de movilidad reducida, no vean impedidas sus formas de mantenerse activos debido a sus capacidades limitadas.

Alcance:

Este proyecto suple la necesidad de regar plantas sin la necesidad de un esfuerzo físico y proveer información al respecto gracias a un sistema de monitoreo, siempre enfocado de manera asistencial. Si bien puede usarse para otros fines y otro tipo de usuario, el público objetivo de nuestra propuesta es acotado. Además de realizar la operación básica de riego, las herramientas estadísticas que provee el monitoreo del proyecto es en lugar de detallada, precisa y concisa para que cualquier usuario pueda usar la información de sus riegos de la forma que estime conveniente.

Objetivos:

General: Desarrollar un sistema de riego automatizado, remoto e intuitivo para ser implementado en casas inteligentes con enfoque asistencial.

Específicos:

- Conocer, estudiar y usar herramientas de Internet de las cosas (IOT) como Raspberry Pi y sensores analógicos
- Desarrollar software del sistema que permita interactuar al usuario final con la información de sus plantas
- Implementar Botani Drip como sistema de riego automatizado

1.1.3 Suposiciones

Al momento de realizar este proyecto, hay que considerar lo siguiente: Todas las herramientas de trabajo y sensores requeridos estarán disponibles para ser usados cuando se necesite dentro del proyecto. El grupo cumplirá con los plazos establecidos por su organización interna y el profesor. El producto elaborado cumple con todo lo requerido y propuesto en este informe. El grupo aprenderá uniformemente y en conjunto los conocimientos básicos para desarrollar e implementar el producto.

1.1.4 Restricciones

- Se debe utilizar una Raspberry Pi como elemento principal del producto -La comunicación con la Raspberry Pi debe ser mediante una aplicación para dispositivos móviles
- Los recursos son limitados, en caso de averías se deberá buscar otro tipo de tecnología para suplir el requerimiento
- No se permiten más de un usuario asociado al producto implementado

1.1.5 Entregables

Los entregables de este proyecto son:

- Carta Gantt
- Bitácoras Semanales
- Planificación de Gestión de Riesgos
- Planificación del Personal y Recursos

2. Organización del Proyecto

2.1 Personal y entidades internas

Los miembros de este proyecto son:

1. Patricio Chang Reyes
2. Francisco Pantoja González
3. Hernán Vazque Lorca

2.2 Roles y responsabilidades

Todos los miembros del equipo cumplen la función de desarrolladores y programadores. Sin embargo, los roles específicos definidos para el proyecto son:

Jefe de Grupo: Encargado de coordinar, liderar y motivar al equipo de trabajo en el cumplimiento de los plazos y tareas establecidas

Ejecutivo de Repartos: Planifica, redacta y guía la escritura de bitácoras, informes por fases y contenido de la Wiki

Analista y Programador Jefe: Encargado de recomendar e introducir a la capacitación de los demás miembros del equipo en el uso de diversas herramientas de software. Corrige inminentes errores de implementación y programación.

2.3 Mecanismos de Comunicación

Se ha establecido la herramienta Redmine, donde está la Wiki, Carta Gantt, Bitácoras semanales

Para coordinar reuniones fuera del horario de clases se ha creado un grupo de Whatsapp

Para compartir y editar documentos de los entregables se ha establecido una carpeta de Google Drive

3. Planificación de los procesos de gestión

3.1 Planificación de estimaciones

Jefe de Grupo: \$23.000, horas totales: 192, \$4.416.000

Ejecutivo de Entregables: \$25.000, horas totales: 150, \$3.750.000

Analista y Programador en Jefe: \$18.000, horas totales: 200, \$3.600.000 Tiempo para programación: 4 meses

Hardware:

Raspberry PI 4 Modelo B (1): \$85.000

Notebook de trabajo (2): \$600.000

Teléfono inteligente (1): \$250.000

Set Sensor Higrómetro con pines (1): \$1.600

Mini bomba de agua (1): \$2.100

Módulo de relay (relé) (1): \$2.500

Sensor de conductividad (1) : \$5.000

Tubo PVC (2): \$2.000

Macetas (4): \$7.200

Alfombrilla impermeable (1): \$500

Software:

-Sistema Operativo Raspbian

-Sistema Operativo Android 14

-Lenguaje de Programación Python 3

-Entorno de Desarrollo Integrado Visual Studio Code

-Herramienta de creación Android Studio -Plataforma Redmine

Costo en bruto del proyecto: \$12.721.900

3.2 Planificación de recursos humanos

Analistas: 1, Diseñador:1, Programador: 3, Jefe de Proyecto: 1.

Se han asignado los siguientes cargos especiales a cada integrante:

Jefe de Grupo: Francisco Pantoja

Ejecutivo de Repartos: Patricio Chang

Analista y Programador en Jefe: Hernán Vazque

3.3 Riesgos

Algunos riesgos a considerar dentro del proyecto están clasificados según probabilidad de ocurrencia y nivel de impacto (siendo 1 el menor y 2 el mayor) en la siguiente tabla:

| RIESGOS | PROBABILIDAD DE OCURRENCIA | NIVEL DE IMPACTO | ACCIÓN REMEDIAL |
|---|----------------------------|------------------|---|
| Cambios en los requerimientos | 75% | 2 | Rastrear la información para valorar el impacto de los nuevos requerimientos, pensar en soluciones que integren nuevos requerimientos con los antiguos para aminorar el tiempo extra de planeación. |
| Reestructuración organizacional | 50% | 1 | Asegurarse que la reestructuración permita cumplir con los requerimientos más eficientemente que de la forma antigua, hacer contribuciones muy importantes a las metas del proyecto. |
| Problemas financieros de la organización | 50% | 2 | Solicitar más recursos al encargado de recursos o en su defecto planificar y rehacer estimaciones de costos |
| Materiales dañados | 75% | 1 | Reemplazar el material dañado por uno nuevo |
| Problemas en el equipo | 50% | 2 | Conversar entre los miembros del equipo afectados para llegar a un acuerdo y/o decisión al respecto |
| Pequeños problemas en el proyecto | 50% | 2 | Reconocer los problemas encontrados en el proyecto, para encontrar una solución óptima y eficaz. |
| Fallo en la programación | 75% | 2 | Conversar entre todos los miembros del equipo para ajustar y/o incurrir en reprogramaciones |
| Exposición a condiciones climáticas no favorables | 75% | 2 | Ajustar la protección de la raspberry pi a las condiciones climáticas de la zona |
| Mantenimiento a largo plazo | 50% | 2 | Hacer el mantenimiento o reemplazo lo más simple posible, además de crear un manual para poder dar mantenimiento al proyecto |
| Enfermedades del Personal | 25% | 1 | Cubrir el trabajo del miembro afectado. De ser una situación grave se debe pensar en reestructurar la organización del proyecto |
| Problemas en el Reclutamiento | 25% | 2 | Reorganizar el equipo de tal forma que haya traslape en el trabajo y las personas comprenden el de los demás |

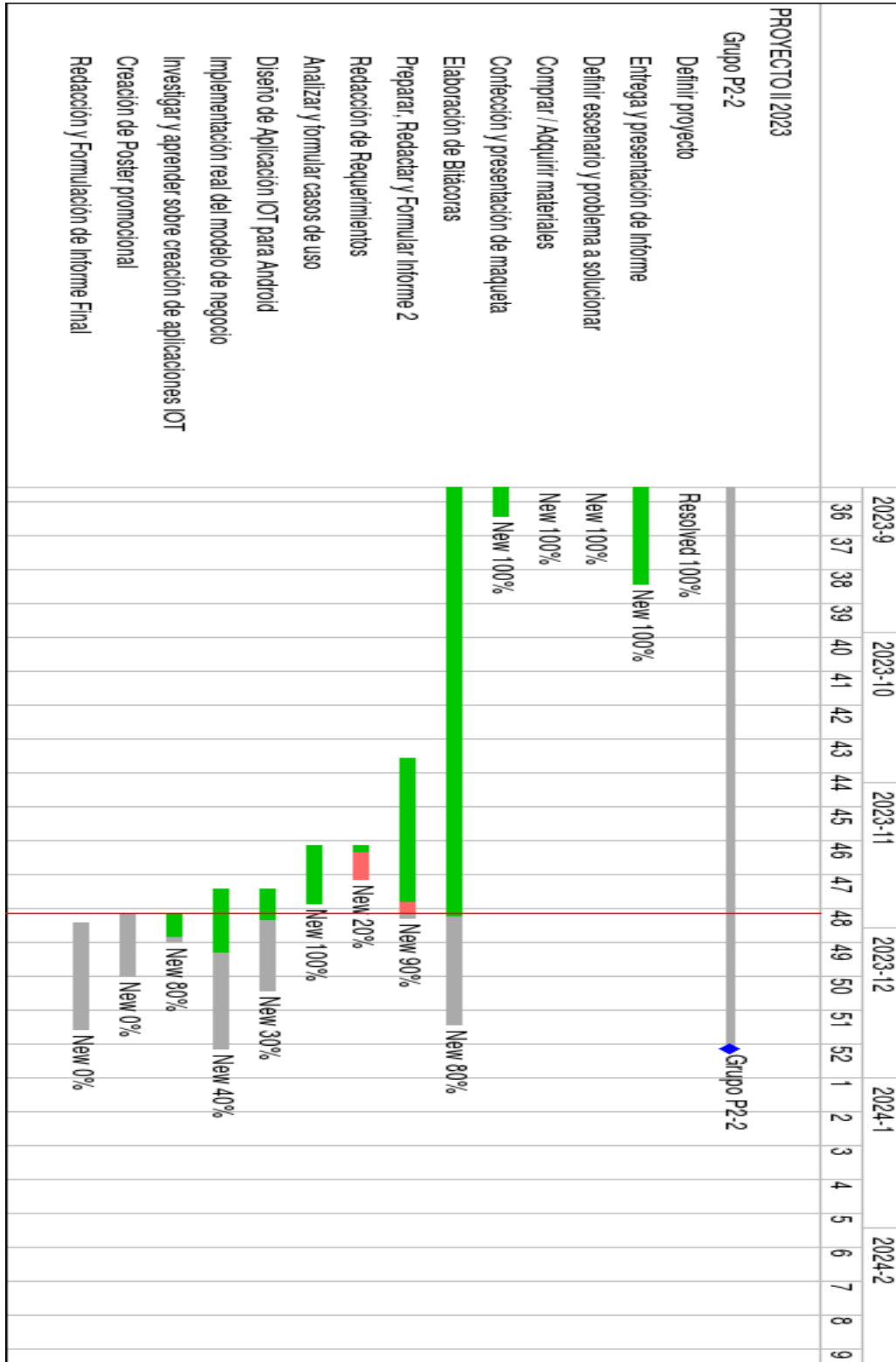
3.4 Factores de Riesgo

Otros tópicos que pueden llegar a producir riesgos a considerar son los siguientes:

| TIPO DE RIESGO | INDICADORES POTENCIALES |
|-----------------------|---|
| Tecnología | Entrega retrasada del hardware o de la ayuda del software, muchos problemas tecnológicos reportados |
| Personas | Baja moral del personal, malas relaciones entre los miembros del equipo, disponibilidad de empleo |
| Organizacional | Chismorreos organizacionales, falta de acciones por el administrador principal |
| Herramientas | Rechazo de los miembros del equipo para utilizar herramientas, peticiones de estaciones de trabajo más potentes |
| Requerimientos | Peticiones de muchos cambios en los requerimientos |
| Estimación | Fracaso en el cumplimiento de los tiempos acordados y en la eliminación de defectos reportados |
| Mantenimiento | Fallo del hardware o de los materiales usados |

4. Planificación de Actividades (Carta Gantt)

El desarrollo del proyecto ha sido plasmado en la siguiente carta Gantt, la cual va desde comienzos de Agosto hasta principios de Diciembre:

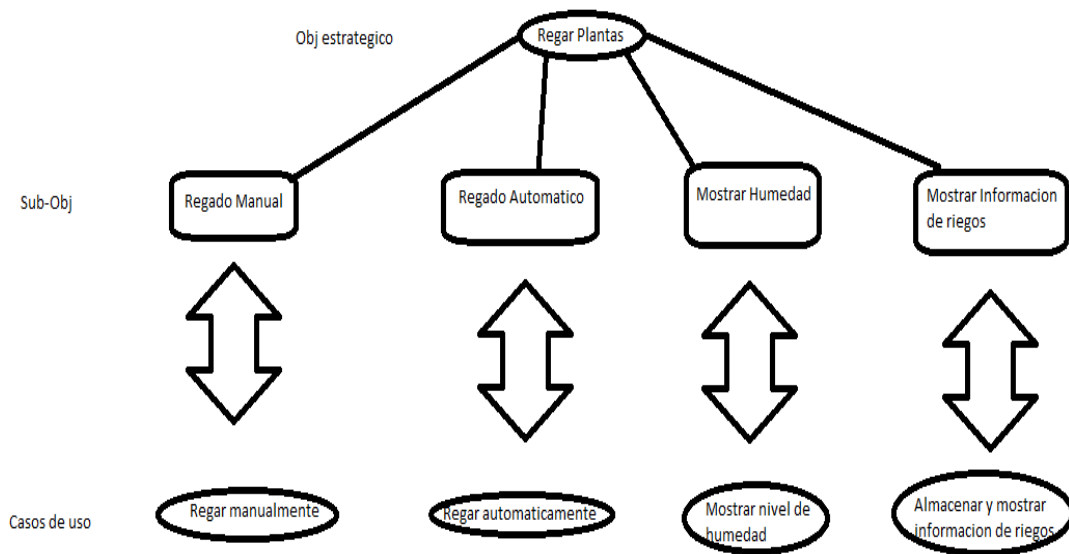


5. Planificación de Procesos Técnicos

5.1 Modelo de Proceso

Dado el objetivo estratégico de Regar plantas se desglosan los Casos de Uso:

- Regar
- Regar manualmente
- Programar riego automático
- Regar automáticamente
- Mostrar condiciones del suelo
- Mostrar historial de riego



5.1.1 Requerimientos establecidos

Requerimientos funcionales:

- Efectuar riegos remotamente por accionamiento manual y/o automático
- Recopilar y desplegar estadísticas sobre las condiciones del suelo
- Notificar al usuario cuando se haya efectuado un riego
- Almacenar información de Riegos efectuados
- Visualizar listado de Riegos efectuados

Requerimientos no funcionales:

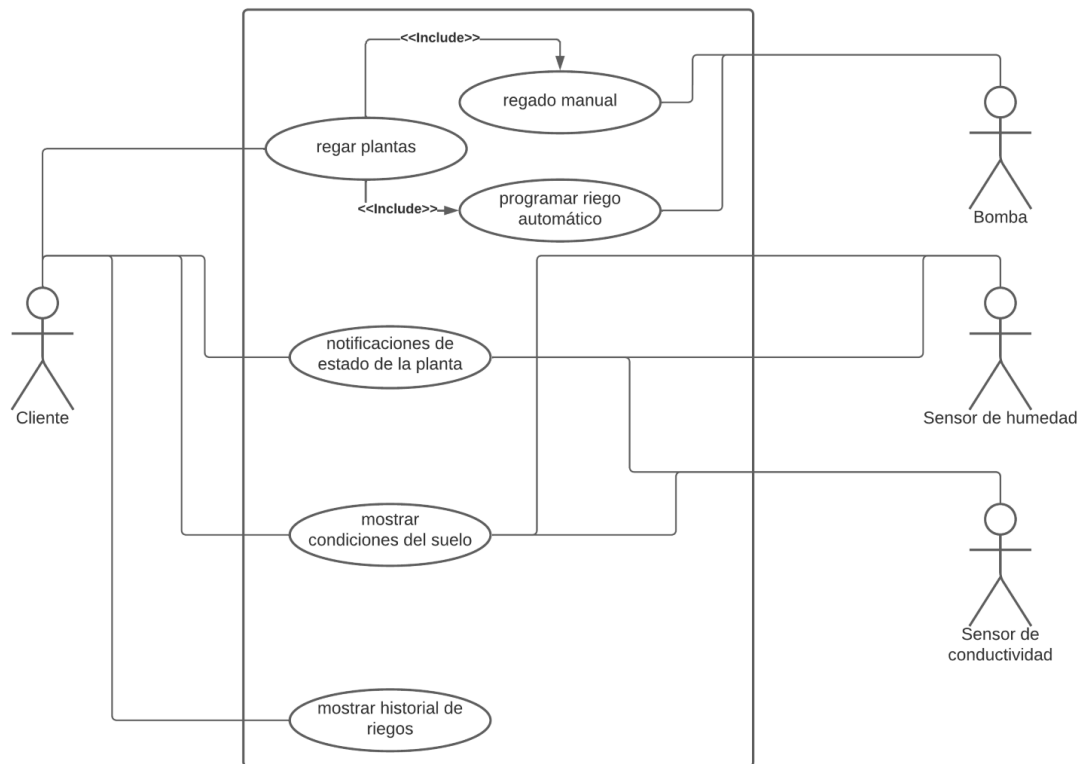
- Conexión entre usuario y raspberry debe ser por medio de una red Wi-fi implementando el modelo de paradigma cliente-servidor
- El cliente debe encontrarse en el dispositivo móvil del usuario
- El servidor debe estar alojado en la Raspberry Pi y recibir peticiones del cliente por medio del protocolo https

5.1.2 Modelo de Diseño (Caso de uso general)

El modelo de proceso cuenta con los siguientes actores:

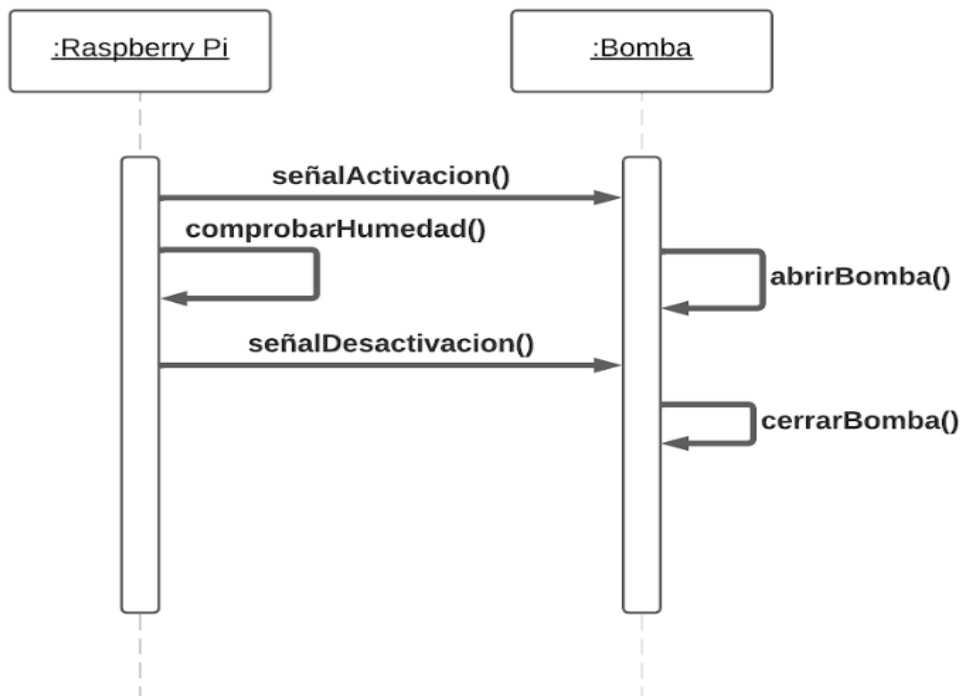
- Cliente
- Raspberry Pi
- Bomba de agua
- Sensor de humedad
- Sensor de conductividad

Se ha modelado el funcionamiento del sistema e interacción entre actores de la siguiente manera:



| | |
|-----------------|---|
| Caso de uso | Regar |
| Descripción | Las plantas son regadas por el sistema a través de la Raspberry Pi |
| Actor | Bomba, Raspberry Pi |
| Precondición | La bomba debe estar conectada a la Raspberry Pi y se debe recibir una señal de activación de esta. |
| Flujo Principal | <ol style="list-style-type: none"> 1. Al recibir una señal de activación de la Raspberry Pi se activan los mecanismos de regado 2. Mientras el porcentaje de humedad sea inferior a 100% los mecanismos de riego funcionan activando la bomba de agua |
| | Las plantas habrán sido regadas y la información del riego efectuado se añadirá al historial de riegos |

Diagrama de secuencia
"Regar"



| | |
|-------------------|---|
| Caso de uso | Regar manualmente |
| Descripción | El cliente puede regar de forma manual las plantas |
| Actor | Cliente, Raspberry Pi |
| precondición | El cliente debe tener conexión a la Raspberry Pi por medio del sistema |
| Flujo Principal | <ol style="list-style-type: none"> 1. El cliente ve el estado de humedad de la planta y decide regar la planta 2. Si la humedad de la planta es menor al 100% esta mandará una señal de activación manual a la Raspberry Pi. <ol style="list-style-type: none"> 1.2 <<include>> Regar 3. El sistema notifica al cliente que su riego manual fue realizado |
| Flujo Alternativo | <ol style="list-style-type: none"> 3. Si la humedad de la planta es 100% esta no se regará y se enviará un aviso al cliente. |
| Postcondición | Las plantas se habrán regado de forma manual |

Diagrama de secuencia "Regar manualmente"

caso humedad<100%

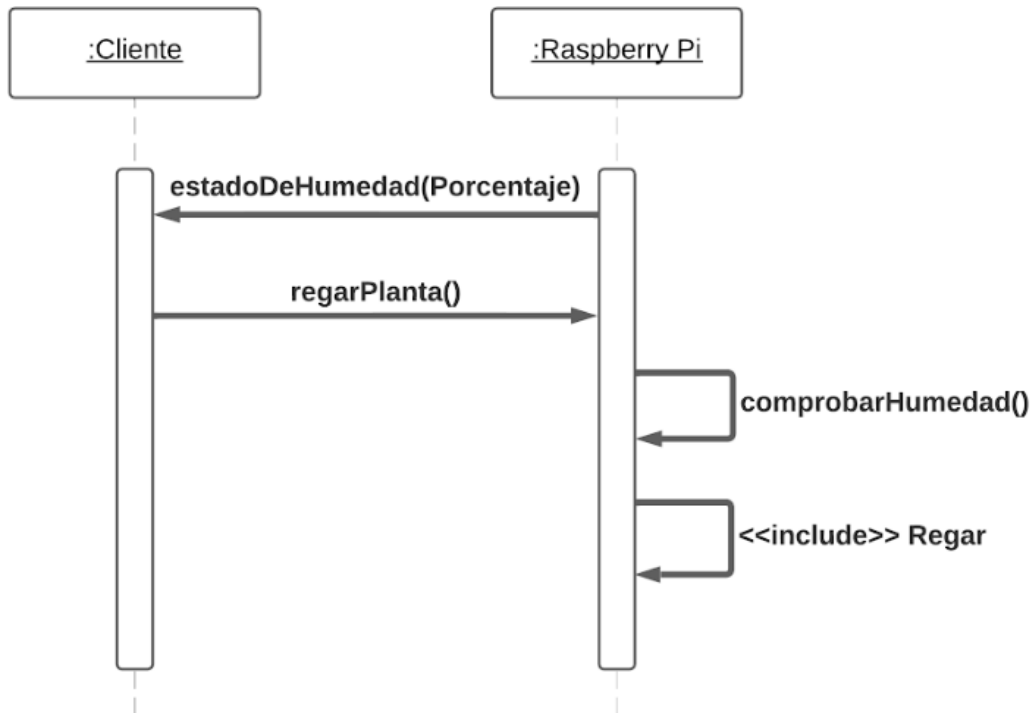
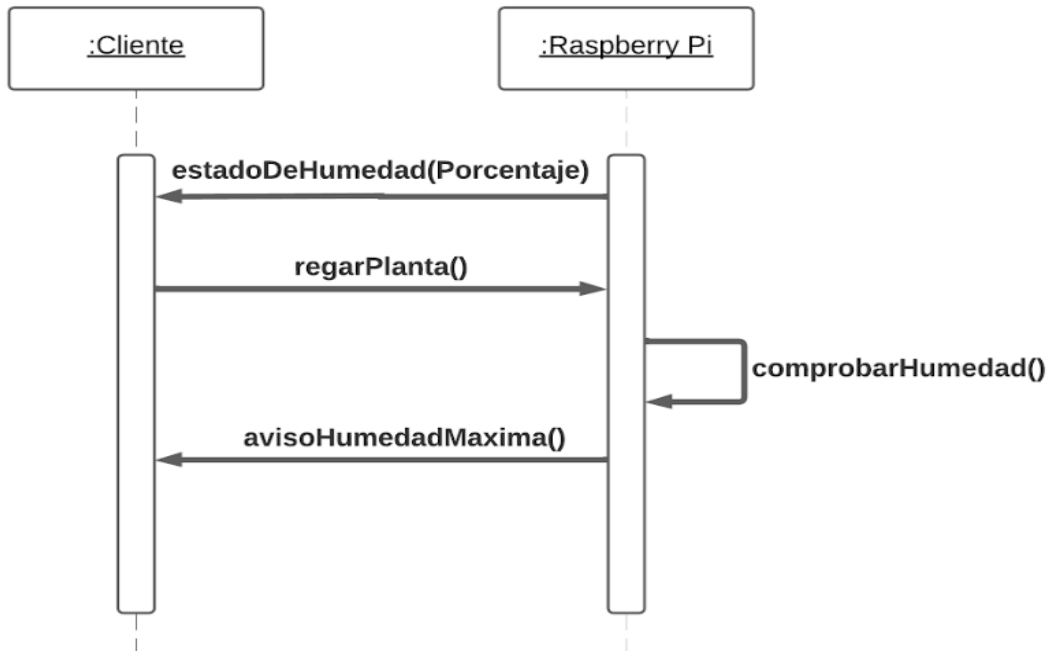


Diagrama de secuencia "Regar manualmente"

caso humedad=100%



| | |
|-------------------|--|
| Caso de uso | Programar riego automático |
| Descripción | El cliente agenda horarios en los cuales se efectúan riegos a lo largo del día |
| Actor | Cliente |
| Flujo Principal | <ol style="list-style-type: none"> 1. El cliente indica al sistema una hora del día en la cual regar las plantas 2. El sistema verifica que haya un intervalo de más de una hora de diferencia hacia adelante y hacia atrás del horario escogido <ol style="list-style-type: none"> 2.1 De existir el intervalo de tiempo adecuado, agenda el horario elegido por el cliente para el regado automático 2.2 De no existir el intervalo, indica al cliente que ingrese otro horario |
| Flujo Alternativo | <ol style="list-style-type: none"> 2. Si la hora ya está agendada, el sistema indica al cliente que ya existe ese horario |
| Postcondición | El riego queda agendado para ser utilizado |

Diagrama de secuencia
"Programar riego automatico"
caso se pude agendar horario

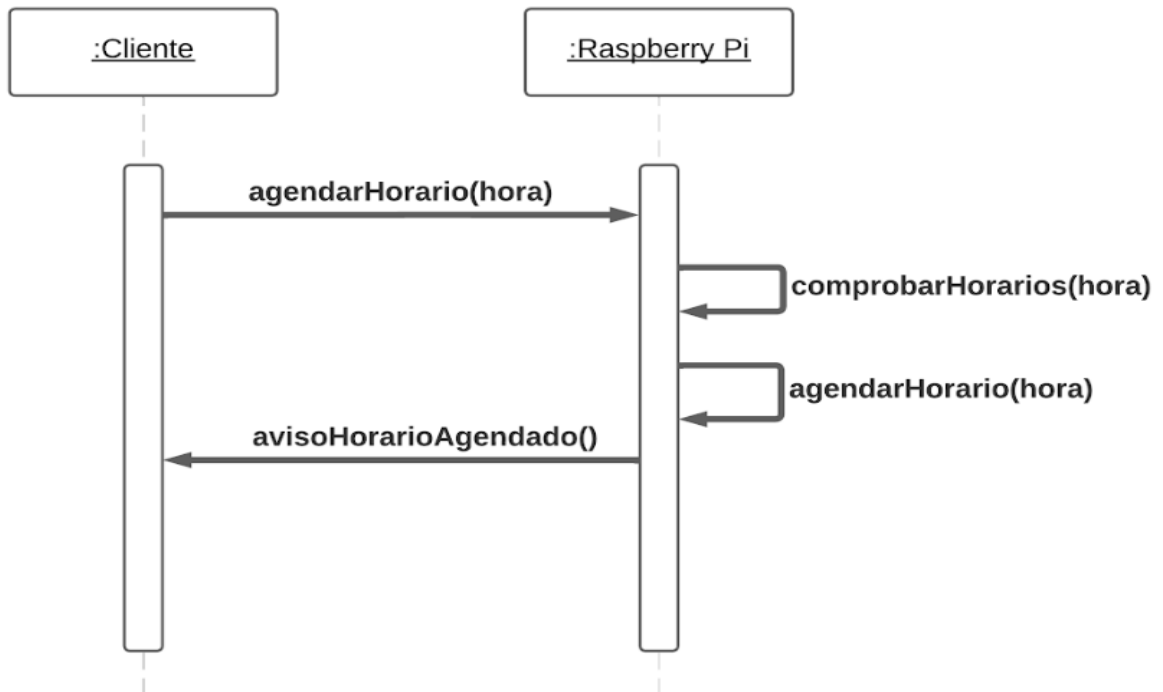
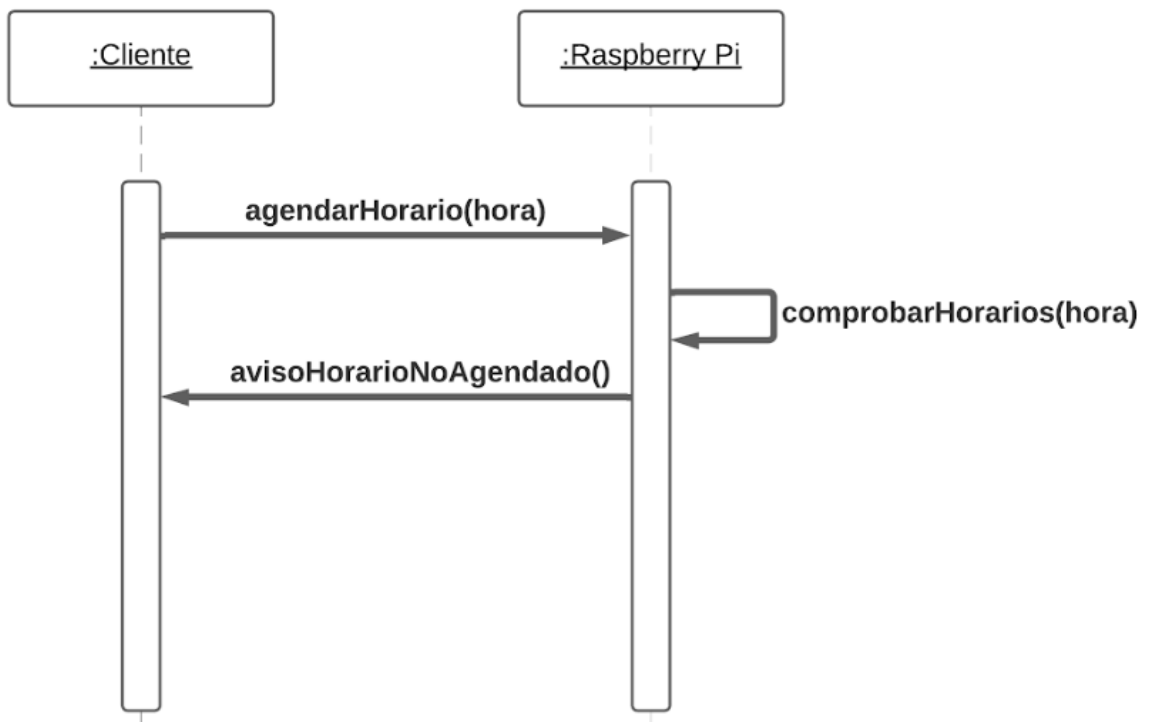


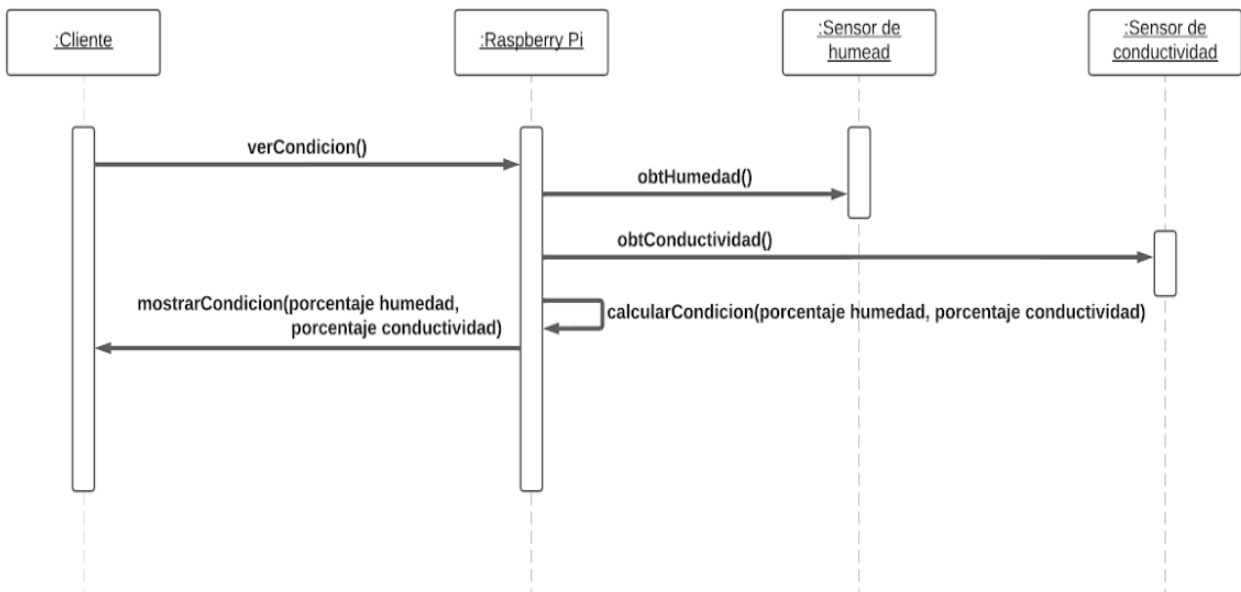
Diagrama de secuencia
"Programar riego automatico"
caso no se pudo agendar horario



| | |
|-----------------|---|
| Caso de uso | Regar automáticamente |
| Descripción | Las plantas son regadas automáticamente y se le da aviso al cliente. |
| Actor | Cliente, Raspberry Pi |
| precondición | El cliente debe tener conexión a la Raspberry Pi por medio del sistema y tener riegos programados previamente |
| Flujo Principal | <ol style="list-style-type: none">1. Cuando se recibe una señal de activación automática correspondiente al horario en el cual está agendado un riego automático en la Raspberry Pi, se activan los mecanismos de riego hasta que la humedad llegue al 100%2. <<include>> Regar3. El sistema notifica al cliente que su riego automático fue realizado |
| Postcondición | Las plantas se habrán regado de forma automática |

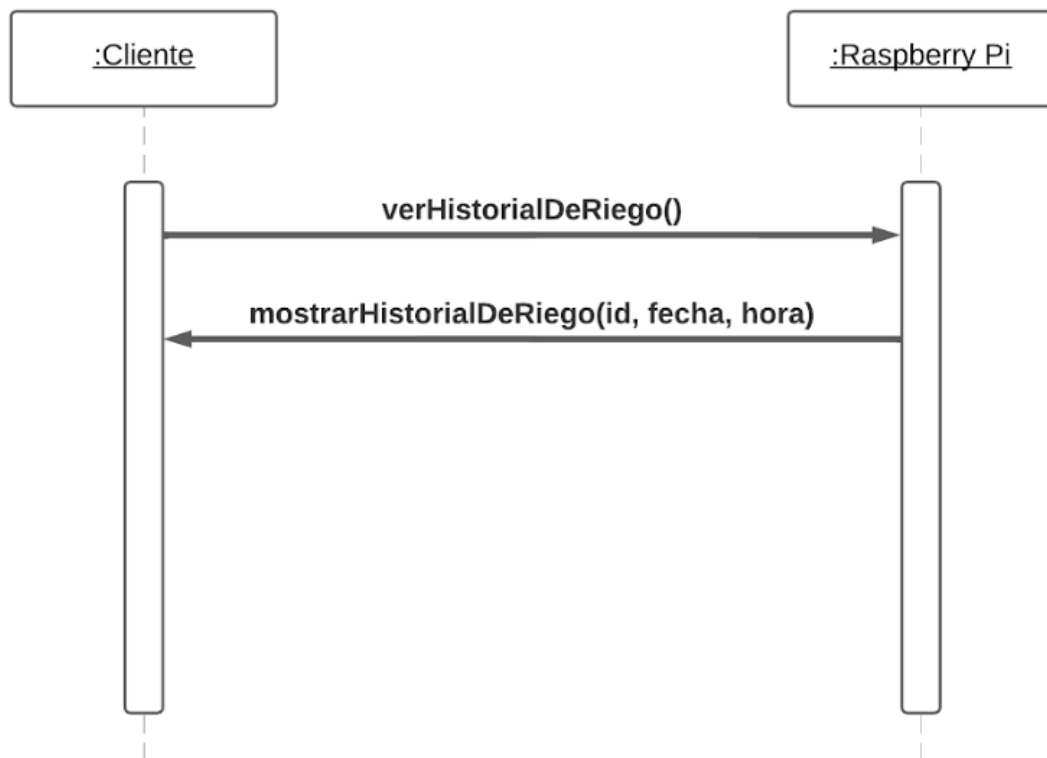
| | |
|-----------------|---|
| Caso de uso | Mostrar condiciones del suelo |
| Descripción | El cliente puede revisar las condiciones del suelo en tiempo real |
| Actor | Cliente, Raspberry Pi |
| Flujo Principal | <ol style="list-style-type: none"> El cliente solicita visualizar estadísticas sobre el suelo en el cual se efectúan los regados, estos datos son recopilados por el sensor de humedad y sensor de conductividad. El sistema muestra al cliente el porcentaje de humedad y porcentaje de conductividad del suelo |

Diagrama de secuencia
"Mostrar condiciones de suelo"



| | |
|-----------------|--|
| Caso de uso | Mostrar historial de riegos |
| Descripción | El cliente puede ver los riegos que se han hecho con anterioridad. |
| Actor | Cliente |
| precondición | Se debe tener almacenada la información de los riegos. |
| Flujo Principal | <ol style="list-style-type: none">1. El cliente puede ver el historial de riego de sus plantas.2. Por cada riego efectuado se muestra el id, fecha y hora de realización. |

Diagrama de secuencia "Mostrar historial de riego"



5.1.3 Descripción de la arquitectura

Una vez consolidada la idea, es necesario establecer cómo será el proceso de conexión entre el usuario final y el modelo de negocio. En este caso se ha optado por una implementación usando el paradigma cliente-servidor, donde se encuentran las siguientes tecnologías:

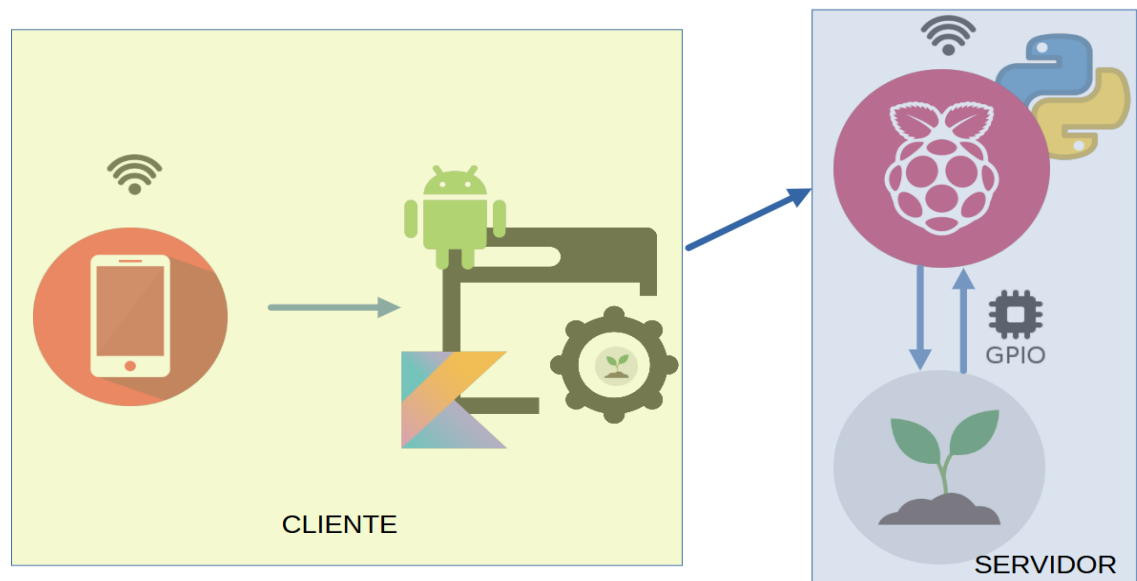
Cliente:

El usuario usará su dispositivo móvil con sistema operativo Android. La aplicación del sistema está escrita en Kotlin, lenguaje de programación para la creación de aplicaciones nativas en Android.

Servidor:

El servidor está compuesto principalmente por la Raspberry Pi, sobre ella se ejecuta un software que funciona como intermediario entre el hardware y el usuario. Se usan las librerías de Python: Django y RPI.GPIO, para el backend como para la manipulación de entradas y salidas GPIO respectivamente

Ambos apartados tanto cliente como servidor deben estar conectados a una red Wi-Fi o en su defecto, una conexión Ethernet.

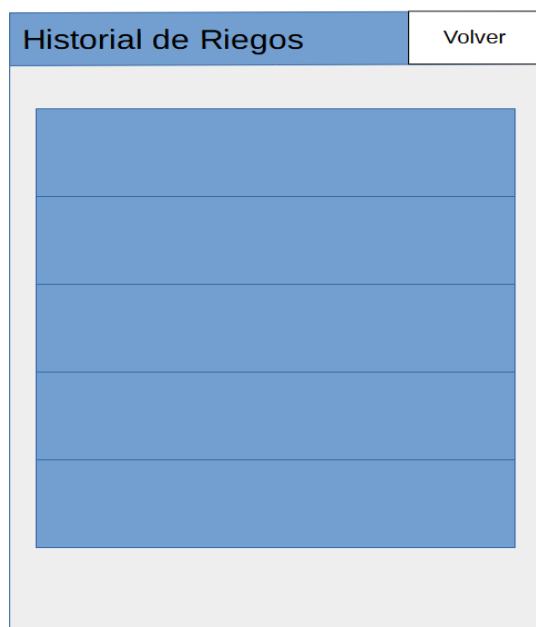


6. Diseño de interfaces de Usuario

Debido al enfoque asistencial y destinado a personas pertenecientes a la tercera edad en su mayoría. Se ha optado por la realización de 3 vistas sencillas: Vista principal, Historial de Riego, y Ver Estadísticas



Diseño de GUI:
Vista principal
- Ver historial de riegos
- Ver condiciones del suelo
- Regar



Diseño de GUI:
Historial de Riego



Diseño de GUI: Ver Estadísticas

7. Conclusiones

Se ha optado por brindar una solución concisa e implementable a un problema específico. Como grupo se espera que lo plasmado en esta fase de planificación sea de guía para todo el transcurso del proyecto, aplicando conocimientos de gestión y planificación de proyectos. De la misma forma se espera que el reconocimiento de posibles riesgos y factores de riesgos acompañado de sus respectivas acciones remediales permitan que frente a cualquier problema presentado, el grupo de trabajo será capaz de enfrentarlo, aplicando así principios de ingeniería en el desarrollo de cada miembro como estudiante, y con ello, preparándose para el mundo laboral.

A través de un modelado de los procesos referentes al modelo de negocio es posible contextualizar y traer a la realidad la idea planteada en la fase de planeación. UML es una herramienta muy poderosa para realizar esta labor

Finalmente, la descripción de la arquitectura del producto es de vital importancia para la fase de implementación, ya que traza las directrices a seguir para cumplir a cabalidad con la planeación del grupo.

Botani Drip tiene base sólida tecnológica y lógica para llevarse a cabo.

8. Referencias

- Beneficios de la jardinería para personas mayores:
<https://teleasistencia.es/es/blog/ocio-en-la-tercera-edad/beneficios-de-la-jardineria-para-personas-mayores>
- Raspberry Pi 4: <https://raspberrypi.cl/raspberry-pi-4/>
- Raspberry Pi 4 Modelo B en el mercado:
<https://mcielectronics.cl/shop/product/raspberry-pi-4-modelo-b-8gb-ram-raspberry-pi-28296/>
- Android Studio (Jetpack Compose):
<https://developer.android.com/jetpack/compose?hl=es-419>