

INFORME FACTIBILIDAD IMPLEMENTATIVA

Sistema de Riego: controlado a través de Bluetooth

Asignatura: Aplicaciones móviles para IoT

Sección: TI3042/D-B50-N4-P13-C1(E-F)

Nombre del académico: Pablo Hernández Huenunguir

Nombre de los integrantes del grupo: Alejandra Miranda & Camila Cruz

Diciembre 2025

Sede Ñuñoa, Chile

ÍNDICE

Introducción	2
Objetivos	3
Uso de Microcontrolador, Integrando Sensores y Actuadores	4
a. Selección del Microcontrolador	5
b. Integración de Sensores	5
c. Integración de Actuadores	6
d. Esquema de Conexión	8
e. Pruebas y Validación	9
Empleo de Antenas Inalámbricas	10
a. Selección de Antenas	11
b. Integración con el Microcontrolador	12
c. Configuración y Programación	12
d. Consideraciones de Seguridad	14
Aplicación Móvil e IoT	16
a. Requerimientos Funcionales de la Aplicación	16
b. Requerimientos No-Funcionales de la Aplicación	17
b. Diseño de la aplicación	18
c. Integración de Hardware	22
d. Pruebas y validación	28
e. Escalabilidad	30
Análisis de entornos posibles	31
a. Identificación de entornos	32
b. Evaluación de Factibilidad	32
c. Cumplimiento de Estándares OT	33
d. Impacto de Limitaciones y Restricciones	33
e. Pruebas y Validación	34
f. Recomendaciones	36
Conclusiones	36

Introducción

A grandes rasgos, este informe documenta la implementación sumativa de un proyecto que propone una solución móvil de IoT donde se desarrolla un prototipo de Arduino que use sensores, actuadores y pueda ser manipulado y monitoreado desde una aplicación móvil mediante Bluetooth o Wifi. Para ello se realizó una serie de actividades durante el transcurso del semestre que consistieron en levantar antecedentes de una problemática IoT, diseñar y construir una serie de prototipos y realizar pruebas de integración de los diversos componentes.

La problemática que se propuso por abordar consiste puntualmente en monitorear en tiempo real la temperatura y humedad de un entorno, ya sea un invernadero o jardín de pequeña escala. Diseñamos un sistema que permite desde una aplicación móvil con conexión bluetooth acceder a varios sensores de temperatura y humedad, y en el caso de registrar determinados rangos de temperatura alerte al usuario para que este pueda activar o desactivar unos riegos automáticos.

El uso combinado de sensores de temperatura, actuadores y una aplicación con conectividad Bluetooth permite abordar de manera eficiente y escalable la problemática planteada. Esta integración posibilita la recolección de datos en tiempo real, una respuesta inmediata ante condiciones críticas mediante señales visuales, y una interacción simple e intuitiva para el usuario través de una aplicación móvil. Además, el sistema es escalable, ya que en el caso que se necesite, se pueden añadir más sensores y actuadores sin la necesidad de reestructurar el diseño base, y resulta una solución creativa al aprovechar tecnologías accesibles para optimizar el control ambiental en espacios agrícolas.

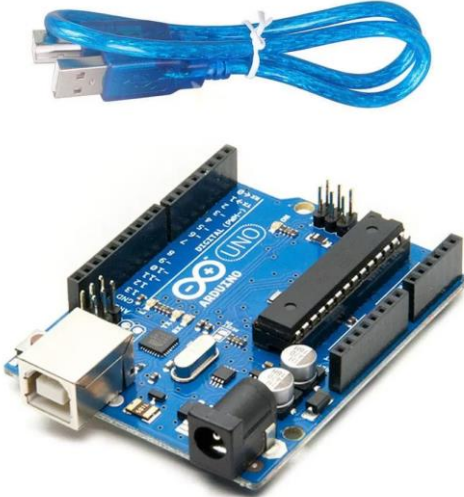
Dado este marco del proyecto, a continuación, se redacta la estructura de este informe y se resume en breve las temáticas de cada apartado: primeros se establecen los objetivos (generales y específicos) de este informe para documentarlos de forma explícita. Esto es seguido por una descripción de todos los componentes relevantes al desarrollo del proyecto, entre estos el microcontrolador, sensores y actuadoras utilizados. Luego se describe y explica el tipo de conexión inalámbrica que se aplica para hacer posible la integración entre el hardware y la aplicación móvil. En el siguiente apartado luego se realiza un análisis de la aplicación de este sistema a un entorno real, evaluación de factibilidad, sus impactos y limitaciones y consideraciones de seguridad. Finalmente se cierra este informe con documentar las pruebas de validación, recomendaciones para mejoras futuras y algunos comentarios y conclusiones finales del trabajo en general.

Objetivos

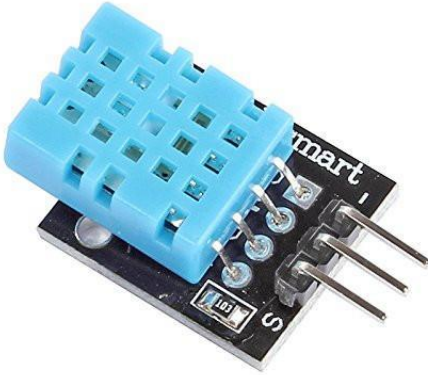
OBJETIVO GENERAL	Desarrollar un sistema IoT basado en un prototipo de Arduino que, a través de sensores de temperatura y humedad, y controlado mediante una aplicación móvil con conectividad Bluetooth, permita monitorear y gestionar en tiempo real las condiciones ambientales de un invernadero o jardín de pequeña escala, activando o desactivando riegos automáticos ante condiciones críticas de temperatura.
Objetivos Específicos	
1	Diseñar y construir un sistema IoT basado en un prototipo de Arduino que integre sensores de temperatura y humedad, actuadores y un módulo de comunicación Bluetooth, permitiendo la interacción con la aplicación móvil.
2	Desarrollar una aplicación móvil en Android utilizando Kotlin, que permita monitorear en tiempo real los valores de temperatura y humedad, y controlar el sistema de riego a través de la conexión Bluetooth.
3	Integrar e implementar el sistema completo , asegurando la correcta comunicación entre el prototipo Arduino, los sensores, los actuadores y la aplicación móvil, para permitir el control y monitoreo en tiempo real de las condiciones ambientales.
4	Evaluar la funcionalidad del sistema , realizando pruebas de validación en un entorno real y analizando su viabilidad para ser implementado en invernaderos o jardines de pequeña escala.

Uso de Microcontrolador, Integrando Sensores y Actuadores

a. Selección del Microcontrolador

Microcontrolador: Arduino UNO	Función
	Microcontrolador programable que permite interactuar con sensores, actuadores y otros dispositivos electrónicos.
	Justificación
	Utilizamos el microcontrolador Arduino Uno porque permite la implementación de los sensores y actuadores necesarios para nuestra aplicación de riego a distancia a través de Bluetooth.
	Cantidad
	1

b. Integración de Sensores

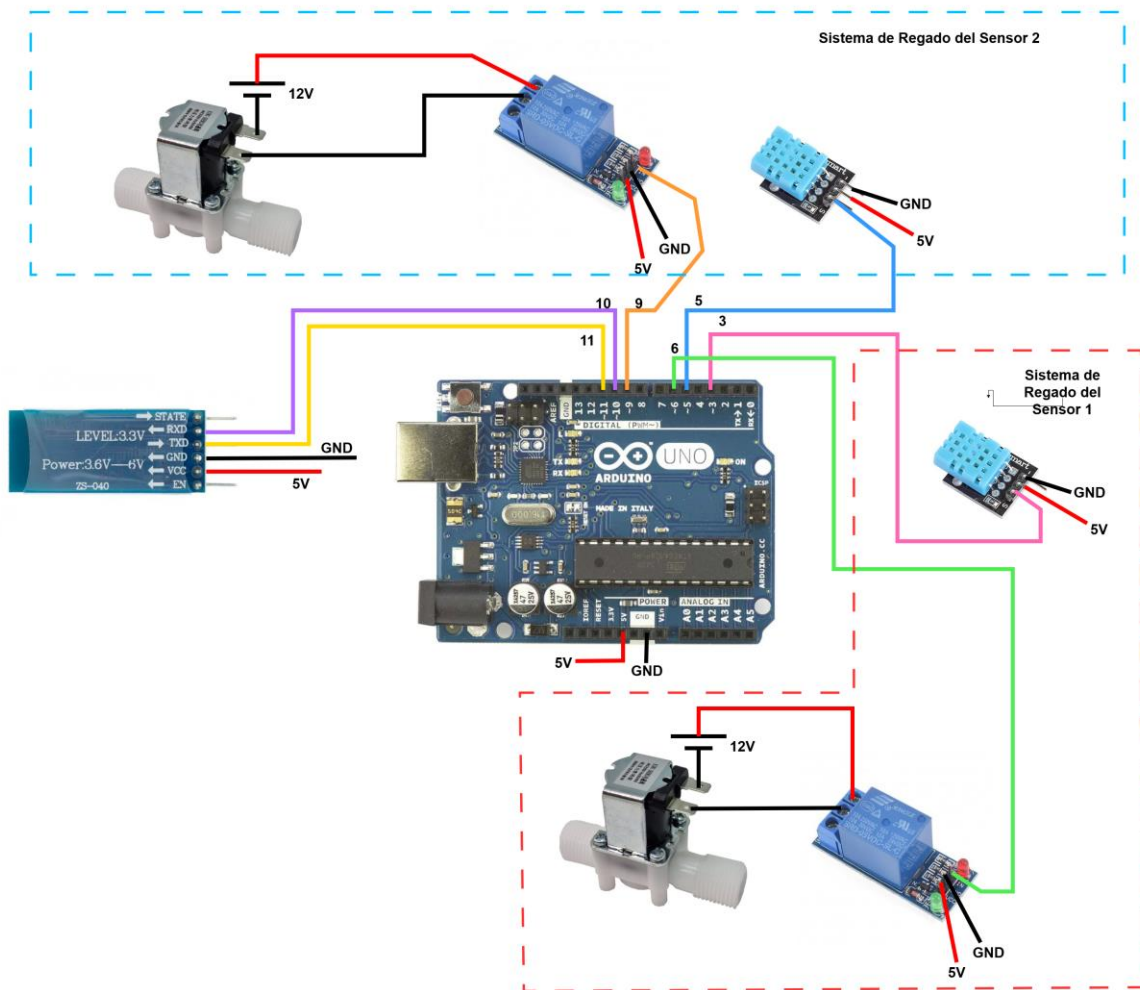
Sensor: DHT11 Temperature and Humidity Sensor	
	Función Este sensor mide la temperatura y humedad.
	Propósito Traduce variables físicas (temperatura y humedad) en información digital utilizable por el sistema para reaccionar de forma oportuna y automatizada. Sin él, no habría forma precisa ni eficiente de medir ni responder al estado térmico del entorno.
	Cantidad
	2
Integración con el microcontrolador	
#include <DHT.h>	La librería "DHT.h" permite la interpretación de las variables físicas medidas por los sensores.
DHT dht1(3, DHTTYPE); DHT dht2(5, DHTTYPE);	Aquí se crean las instancias dht1 y dht2, las cuales configuran los puertos serie que utilizarán los sensores.
dht1.begin(); dht2.begin();	Este fragmento se encuentra dentro del void loop() y realiza la inicialización de los sensores.
if (sensorActivo == '1') { float temp1 = dht1.readTemperature(); float hum1 = dht1.readHumidity(); BT.print(temp1); BT.print(";"); BT.println(hum1); }	En este fragmento se muestra cómo se obtienen las lecturas del sensor 1 y se envían a través del módulo Bluetooth.

c. Integración de Actuadores


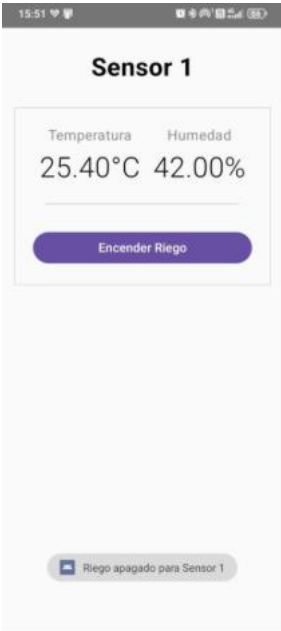
Actuadores: Módulo Relé y Válvula Solenoide	Función
	<p>Módulo Relé: Es un interruptor controlado electrónicamente, que permite activar o desactivar dispositivos de mayor potencia mediante señales de bajo voltaje. En este caso, el relé se utilizaría para controlar el encendido y apagado del sistema de riego automático.</p> <p>Válvula Solenoide: Permite controlar el flujo de agua dentro del sistema de riego automático. Se conecta a la tubería de agua y es controlada eléctricamente para abrir o cerrar el paso del agua.</p>
	Propósito
	<p>El Módulo Relé protege al microcontrolador del riesgo de daño debido a los picos de voltaje o corriente que pueden generarse al controlar dispositivos de mayor potencia. Esto permite que el microcontrolador opere de manera segura sin exponerse a altos voltajes.</p> <p>La Válvula Solenoide actúa como el componente que permite automatizar el sistema de riego.</p>
	Cantidad
	<p>2 Módulos Relé 2 Válvulas Solenoide de paso 12V</p>
Integración con el microcontrolador	
	<p>Aquí se crean las instancias rele1 y rele2, las cuales configuran los puertos serie que utilizarán los sensores.</p>
<pre>bool rele1Activo = false; bool rele2Activo = false;</pre>	<p>Aquí se declaran dos variables booleanas que almacenan el estado de cada relé:</p> <p>true indica que el relé debe estar activado. false indica que debe permanecer desactivado.</p>

<pre>pinMode(rele1, OUTPUT); pinMode(rele2, OUTPUT); digitalWrite(rele1, HIGH); digitalWrite(rele2, HIGH);</pre>	<p>Este bloque pertenece a la función setup().</p> <p>En pinMode, se establecen los pines establecidos de los módulos relé como output.</p> <p>Con el digitalWrite, se inicializan ambos módulos en estado inactivo por la lógica invertida de los módulos relé.</p>
<pre>if (comando == 'A') { rele1Activo = true; } if (comando == 'B') { rele1Activo = false; } if (comando == 'C') { rele2Activo = true; } if (comando == 'D') { rele2Activo = false; } }</pre>	<p>En este fragmento se evalúan los valores recibidos a través de Bluetooth donde 'A' activa el relé 1, 'B' desactiva el relé 1, 'C' activa el relé 2 y 'D' desactiva el relé 2.</p>
<pre>digitalWrite(rele1, rele1Activo ? LOW : HIGH); digitalWrite(rele2, rele2Activo ? LOW : HIGH);</pre>	<p>Con este fragmento se actualizan los pines físicos según el estado deseado:</p> <p>Si rele1Activo o rele2Activo se encuentran en true, el pin correspondiente se pone en LOW, activando el relé.</p> <p>Si es false, el pin se pondrá en HIGH, desactivando el relé.</p>

d. Esquema de Conexión



e. Pruebas y Validación

Módulo relé activado y desactivado y sensor DHT11	Explicación
 	<p>Aquí vemos cómo el módulo relé se activa y desactiva, al prenderse o apagarse el led verde, dependiendo de la señal mandada a través de la aplicación.</p> <p>Y en la aplicación vemos el correcto funcionamiento y procesamiento de los datos medidos y enviados por el sensor de temperatura.</p> <p>Los mismos métodos se aplican para el sensor 2 y su respectivo módulo relé y válvula.</p>

Empleo de Antenas Inalámbricas

a. Selección de Antenas

Modulo Bluetooth: HC-05	Función
	<p>Permite la conexión inalámbrica con Arduino. Mantiene la conexión activa entre los sensores y la aplicación mientras se recibe información, permite a la aplicación recibir datos y a su vez enviar instrucciones.</p>
	Propósito
	<p>Mejora la interacción del sistema al facilitar una conectividad directa, rápida y sin cables entre los sensores, actuadores y la aplicación del usuario. Esta conectividad permite que el sistema sea más accesible, flexible y eficiente. Ofrece una conexión estable y confiable en distancias cortas (hasta 10 metros aproximadamente), lo cual resulta ideal para el entorno controlado de un invernadero. Además, presenta un bajo consumo energético, tanto en el microcontrolador como en el dispositivo móvil, y no requiere infraestructura adicional de red como routers o acceso a internet. Su configuración es sencilla, permite transmisión de datos en tiempo real y mantiene una comunicación bidireccional eficiente entre el Arduino y la aplicación Android.</p>
	Cantidad
	1

b. Integración con el Microcontrolador

El objetivo de la integración del módulo bluetooth seleccionado con el microcontrolador Arduino es establecer un intercambio bidireccional de datos que permita al usuario monitorear en tiempo real las condiciones ambientales del invernadero y actuar en consecuencia.

A través del módulo Bluetooth, el Arduino envía a la aplicación las lecturas obtenidas por los sensores DHT11, que miden la temperatura y la humedad en diferentes zonas. La aplicación procesa y muestra esta información de manera accesible, alertando al usuario cuando las variables superan rangos críticos. A su vez, el usuario puede enviar comandos desde la aplicación para activar o desactivar el sistema de riego, permitiendo una gestión eficiente y remota del ambiente del invernadero.

c. Configuración y Programación

En los siguientes extractos del código en Arduino IDE, se documenta la configuración y programación del módulo bluetooth.

Extracto #1: Incluir la librería SoftwareSerial	#include <SoftwareSerial.h>
Explicación	La librería SoftwareSerial permite la comunicación serial con dispositivos que no están conectados al puerto serial principal (por ejemplo, el módulo Bluetooth HC-05). En este caso, se utilizarán los pines 10 y 11 de Arduino para establecer la comunicación con el HC-05.

Extracto #2: Configurar la comunicación serial con el módulo Bluetooth	SoftwareSerial BT(10,11);
Explicación	<p>Aquí se crea una instancia llamada BT de la clase SoftwareSerial. Esta instancia configura el puerto serie que utilizará el HC-05 para la comunicación:</p> <ul style="list-style-type: none"> • Pin 10 de Arduino es el RX (recepción) del módulo Bluetooth. • Pin 11 de Arduino es el TX (transmisión) del módulo Bluetooth. <p>Al configurar estos pines, Arduino podrá enviar y recibir datos desde el HC-05.</p>

Extracto #3: Inicialización del Bluetooth en la función setup()	BT.begin(9600);
Explicación	Esta línea inicializa la comunicación serie con el módulo Bluetooth BT a una velocidad de 9600 baudios. Es importante que tanto el HC-05 como Arduino estén configurados para operar a la misma velocidad de comunicación. Este paso es crucial para que el módulo HC-05 pueda comunicarse correctamente con Arduino.

Extracto #4: Lectura de datos del Bluetooth en el loop()	<pre>if (BT.available()) { comando = BT.read(); Serial.write(comando); }</pre>
Explicación	En el ciclo principal del programa (loop()), el Arduino verifica si hay datos disponibles para leer del módulo Bluetooth utilizando BT.available(). Si hay datos, los lee mediante BT.read() y los guarda en la variable comando. Este dato se envía también al monitor serial de Arduino con Serial.write(comando) para que puedas ver qué está recibiendo Arduino desde el Bluetooth.

Extracto #5: Procesamiento de comandos recibidos del Bluetooth	<pre>if (comando == 'A') { rele1Activo = true; } if (comando == 'B') { rele1Activo = false; }</pre>
Explicación	<p>Dentro del bloque if (BT.available()), los comandos recibidos se procesan y determinan las acciones a tomar. Los comandos que el código espera son:</p> <ul style="list-style-type: none"> • '1' o '0': Cambia el sensor activo (sensor 1 o sensor 2). • 'A' o 'B': Controla el estado del relé 1 (activar o desactivar). • 'C' o 'D': Controla el estado del relé 2 (activar o desactivar). <p>Este bloque de código controla los relés en función de los comandos recibidos desde el módulo Bluetooth.</p>

Extracto #6: Envío de datos al dispositivo conectado al Bluetooth	<pre>BT.print(temp1); BT.print(";"); BT.println(hum1);</pre>
Explicación	<p>Después de leer los datos de los sensores (temperatura y humedad de los sensores DHT11), el código los envía al módulo Bluetooth mediante las siguientes líneas.</p> <p>En función de qué sensor esté activo (determinado por el valor de sensorActivo), Arduino enviará los valores de temperatura y humedad al dispositivo conectado al HC-05.</p>

d. Consideraciones de Seguridad

Luego de la configuración y la programación del módulo bluetooth en el código, se debe ejecutar una serie de comandos en la terminal del Arduino IDE con la finalidad de implementar una serie de consideraciones de seguridad para proteger la comunicación inalámbrica.

Las medidas de seguridad que se aplicaron para los propósitos de este proyecto son la configuración de un nombre distintivo de la conexión del módulo y una contraseña para poder acceder a la conexión de del módulo bluetooth por el móvil. A continuación, se documentan los comandos necesarios para ejecutar en la terminal para lograr la configuración de estas medidas de seguridad.

Comando	Función	Respuesta
AT	Verificar que el módulo HC-05 está en modo AT y responde correctamente. Es un comando de prueba de comunicación. Si no responde, el módulo no está en modo AT o la velocidad Baud es incorrecta.	OK
AT+NAME= <i>setname</i>	Cambiar el nombre Bluetooth del HC-05. Cuando otro dispositivo busque dispositivos Bluetooth, el módulo aparecerá con ese nombre.	OKsetname
AT+NAME?	Consultar el nombre actual del módulo HC-05. Sirve para verificar qué nombre tiene configurado.	+NAME:setname OK
AT+PSWD= <i>setpassword</i>	Cambiar el PIN de emparejamiento del HC-05. Solo acepta números de 4 dígitos. Cuando un dispositivo quiera emparejarse, deberá ingresar ese PIN.	OKsetpswd

AT+PSWD?	Consultar el PIN actual del HC-05. Sirve para confirmar la contraseña configurada.	+PSWD:1234 OK
----------	--	------------------

Aplicación Móvil e IoT

a. Requerimientos Funcionales de la Aplicación

RF1: Conexión y comunicación Bluetooth	
RF1.1	La aplicación debe escanear y listar los dispositivos Bluetooth disponibles en el entorno.
RF1.2	La aplicación debe permitir seleccionar y conectar un dispositivo Bluetooth HC-05/HC-06.
RF1.3	La aplicación debe mantener la conexión activa mientras se reciben datos del Arduino.
RF1.4	La aplicación debe enviar comandos al módulo Bluetooth para controlar los relés (activar/desactivar riego).
RF1.5	La aplicación debe recibir datos en formato de texto enviados por Arduino (temperatura y humedad).
RF2: Autenticación y seguridad	
RF2.1	La aplicación debe requerir que el usuario se autentique antes de acceder al sistema.
RF2.2	La aplicación debe validar las credenciales ingresadas por el usuario.
RF2.3	La aplicación debe proteger los datos locales mediante medidas básicas de seguridad (encriptación de credenciales y control de acceso)
RF3: Lectura y visualización de sensores	
RF3.1	La aplicación debe mostrar en pantalla los valores de temperatura y humedad de cada sensor.
RF3.2	La aplicación debe actualizar los datos en tiempo real a medida que recibe nuevas lecturas vía Bluetooth.
RF3.3	La aplicación debe usar elementos visuales (indicadores) para representar los valores recibidos.
RF4: Control del sistema de riego	
RF4.1	La aplicación debe incluir un botón para activar o desactivar el riego.
RF4.2	La aplicación debe enviar el comando correspondiente al Arduino para modificar el estado del riego.

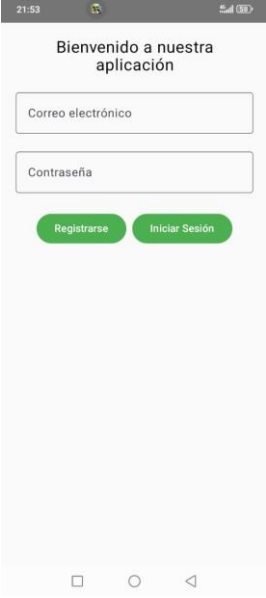
RF4.3	La aplicación debe mostrar en la interfaz el estado actual del riego (“Riego activo” / “Riego apagado”).
RF5: Almacenamiento local de datos (SQLite)	
RF5.1	La aplicación debe almacenar en SQLite los registros históricos de temperatura y humedad.
RF6: Interfaz de Usuario	
RF6.1	Pantalla debe mostrar: <ul style="list-style-type: none"> • Estado de conexión Bluetooth • Lecturas de sensores • Indicadores de riesgo • Botón para controlar el riego

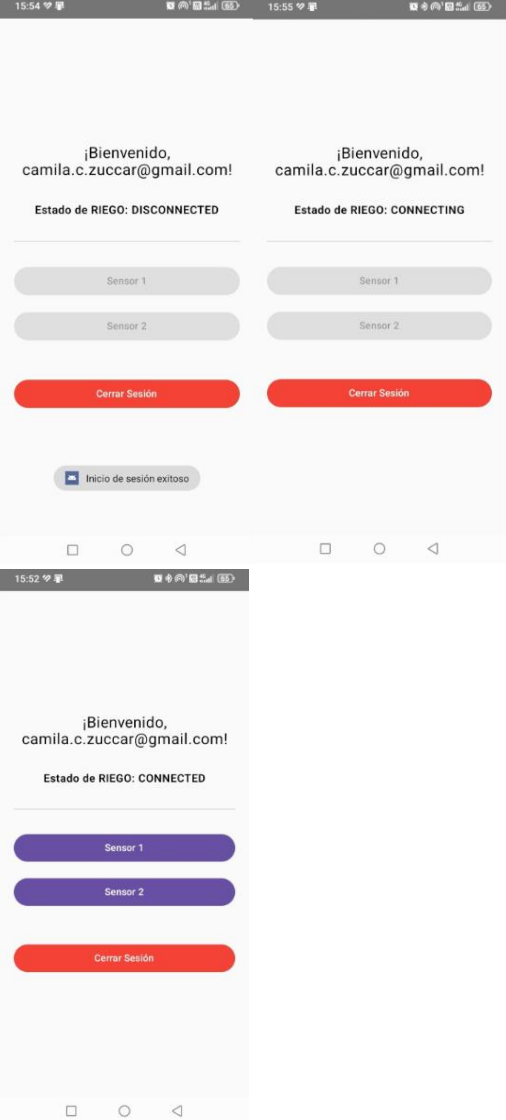
b. Requerimientos No-Funcionales de la Aplicación

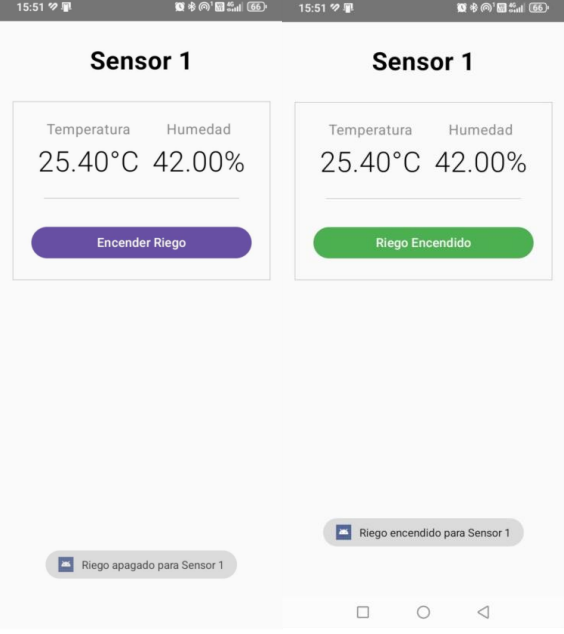
RNF1: Rendimiento	
RNF1.1	La aplicación debe procesar y mostrar las lecturas de sensores en menos de 1 segundo después de recibirlas.
RNF1.2	La aplicación debe mantener un consumo de batería moderado durante la conexión Bluetooth.
RNF1.3	Las consultas a la base de datos SQLite deben ejecutarse en menos de 200 ms.
RNF2: Usabilidad	
RNF2.1	La interfaz debe ser intuitiva y fácil de usar para usuarios sin conocimientos técnicos.
RNF2.2	Los elementos visuales deben ser claros e interpretables (colores, íconos, etc.).
RNF2.3	La aplicación debe ofrecer retroalimentación visual inmediata (estado de conexión, comandos enviados, etc.).
RNF3: Seguridad	
RNF3.1	Los datos de autenticación deben almacenarse de forma segura (encriptación o almacenamiento seguro de Android).
RNF3.2	La comunicación Bluetooth debe evitar el envío de datos sensibles sin autorización.
RNF3.3	La aplicación debe limitar accesos no autorizados a la base de datos.

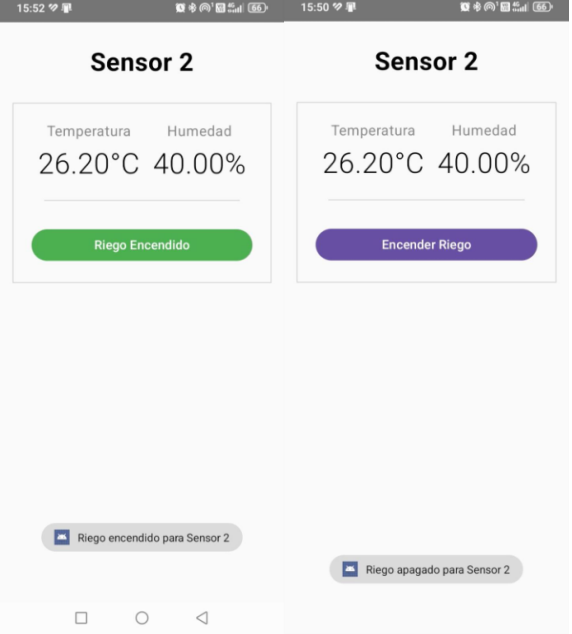
RNF4: Mantenibilidad	
RNF4.1	El código debe desarrollarse en Kotlin siguiendo buenas prácticas de clean code.
RNF4.2	La estructura del proyecto debe permitir futuras ampliaciones (otros sensores, nuevos actuadores).
RNF5: Compatibilidad	
RNF5.1	La aplicación debe ser compatible con Android 7.0 (API 24) o superior.
RNF5.2	La aplicación debe ser funcional en teléfonos con Bluetooth clásico (no BLE), ya que HC-05 usa Bluetooth 2.0.

b. Diseño de la aplicación

Inicio de Sesión en la Aplicación	Descripción
	<p>Esta es la interfaz de inicio de sesión de la aplicación. Es lo primero que aparece al abrir por primera vez la aplicación.</p> <p>En la parte superior se lee el encabezado “Bienvenido a nuestra aplicación”, centrado y en tipografía destacada. Debajo se encuentran los campos “Correo electrónico” y “Contraseña”, los cuales deben ser rellenados correctamente con las credenciales de un usuario registrado, o registrarse como un usuario nuevo.</p>

Estados de Conexión de la Aplicación al Módulo Bluetooth	Descripción
 <p>The image displays three screenshots of a mobile application interface. The top two screenshots show the 'DISCONNECTED' and 'CONNECTING' states, respectively, with buttons for 'Sensor 1', 'Sensor 2', and 'Cerrar Sesión'. The third screenshot shows the 'CONNECTED' state, with buttons for 'Sensor 1', 'Sensor 2', and 'Cerrar Sesión'.</p>	<p>Luego de haber iniciado sesión, el usuario es llevado a la siguiente interfaz donde se le da la bienvenida y se muestra el estado de conexión al Bluetooth del sistema de riego.</p> <p>El usuario no tendrá acceso a los sensores o las válvulas a menos que se haya establecido una conexión al sistema de riego.</p> <p>Una vez que se establece la conexión Bluetooth con el sistema de riego el usuario puede consultar la información de los sensores y encender o apagar el sistema de riego de cada uno.</p> <p>El usuario también puede cerrar sesión desde esta interfaz.</p>

Estados del Sensor 1	Descripción
	<p>Cuando el usuario haya establecido una conexión Bluetooth con el sistema de riego y apretado el botón del sensor uno para consultar sus medidas será llevado a esta interfaz, la cual le mostrará las medidas de temperatura y humedad constante del sensor 1 y le permitirá encender y apagar el riego vinculado a este sensor.</p>

Estados del Sensor 2	Descripción
	<p>Cuando el usuario haya establecido una conexión Bluetooth con el sistema de riego y apretado el botón del sensor uno para consultar sus medidas será llevado a esta interfaz, la cual le mostrará las medidas de temperatura y humedad constante del sensor 2 y le permitirá encender y apagar el riego vinculado a este sensor.</p>

c. Integración de Hardware

Código Arduino Extractos

1. Inicialización de la comunicación Bluetooth (en el setup)

```
SoftwareSerial BT(10, 11); // Se configura la comunicación serial por Bluetooth (RX, TX)
```

```
BT.begin(9600);           // Inicia la comunicación a 9600 baudios
```

Aquí se establece la conexión con el módulo Bluetooth utilizando los pines 10 y 11 del microcontrolador. La comunicación se realiza a 9600 baudios.

2. Lectura de datos enviados desde la aplicación móvil (en el loop)

```
if (BT.available()) {      // Verifica si hay datos disponibles desde la aplicación móvil
  comando = BT.read();      // Lee el siguiente byte de datos (el comando enviado por la aplicación)
  Serial.write(comando);    // Opcional: muestra el comando en el puerto serie (para depuración)
```

```
  // Condiciones para activar o desactivar sensores o actuadores
  if (comando == '1' || comando == '0') {
    sensorActivo = comando; // Cambia el sensor activo (sensor 1 o sensor 2) según el comando
                              recibido
  }
}
```

```
  // Control de los actuadores (relés)
  if (comando == 'A') { rele1Activo = true; } // Activa el primer relé
  if (comando == 'B') { rele1Activo = false; } // Desactiva el primer relé

  if (comando == 'C') { rele2Activo = true; } // Activa el segundo relé
  if (comando == 'D') { rele2Activo = false; } // Desactiva el segundo relé
}
```

El código revisa constantemente si hay datos disponibles (BT.available()).

Lee el byte recibido desde la aplicación móvil con BT.read(). Este byte representa un comando que puede ser uno de los siguientes:

Comando '1' o '0': Cambia el sensor activo (sensorActivo), permitiendo elegir entre los dos sensores (dht1 y dht2).

Comandos 'A', 'B', 'C', 'D': Controlan el estado de los relés, activando o desactivando los actuadores.

3. Envío de datos del sensor seleccionado a la aplicación móvil

```
if (sensorActivo == '1') {      // Si el sensor activo es el 1 (dht1)
    float temp1 = dht1.readTemperature(); // Lee la temperatura del sensor 1
    float hum1 = dht1.readHumidity();    // Lee la humedad del sensor 1

    BT.print(temp1);              // Envía la temperatura al dispositivo móvil
    BT.print(";");               // Separa los valores con un delimitador
    BT.println(hum1);            // Envía la humedad al dispositivo móvil
}
else if (sensorActivo == '0') {    // Si el sensor activo es el 2 (dht2)
    float temp2 = dht2.readTemperature(); // Lee la temperatura del sensor 2
    float hum2 = dht2.readHumidity();    // Lee la humedad del sensor 2

    BT.print(temp2);              // Envía la temperatura al dispositivo móvil
    BT.print(";");               // Separa los valores con un delimitador
    BT.println(hum2);            // Envía la humedad al dispositivo móvil
}
```

Dependiendo del sensor activo (sensorActivo), el código lee los valores de temperatura y humedad de los sensores DHT11 (dht1 o dht2). Estos datos son enviados a la aplicación móvil utilizando el objeto BT que está conectado al módulo Bluetooth. El formato utilizado para enviar los datos es:

Se envía primero la temperatura, seguida de un punto y coma (;), y luego se envía la humedad.

El comando BT.print() envía los datos sin un salto de línea, y BT.println() envía los datos con un salto de línea al final.

4. Retraso para estabilizar la lectura de los sensores

```
delay(800); // Espera 800 ms antes de realizar la siguiente lectura
```

Este retraso asegura que las lecturas de los sensores no sean demasiado rápidas, lo cual podría resultar en lecturas incorrectas. Además, también ayuda a dar tiempo a la aplicación móvil para procesar los datos recibidos.

Código Android Studio

Cambio entre Sensores (HomeActivity.kt y BluetoothManager.kt):

En estos fragmentos de código de los archivos HomeActivity.kt y BluetoothManager.kt podemos observar el intercambio y las funciones que permiten cambiar entre los sensores.

Parte Clave en HomeActivity.kt (El Disparador):

```
Button(  
    onClick = { context.startActivity(Intent(context,  
DetalleSensorActivity::class.java).putExtra("SENSOR_ID", "Sensor 1")) },  
    modifier = Modifier.fillMaxWidth(),  
    enabled = isConnected  
) {  
    Text("Sensor 1")  
}  
  
Spacer(modifier = Modifier.height(16.dp))  
  
Button(  
    onClick = { context.startActivity(Intent(context,  
DetalleSensorActivity::class.java).putExtra("SENSOR_ID", "Sensor 2")) },  
    modifier = Modifier.fillMaxWidth(),  
    enabled = isConnected  
) {  
    Text("Sensor 2")  
}
```

Parte Clave en BluetoothManager.kt (El Ejecutor):

```
fun sendData(data: String) {  
    scope.launch {  
        try {  
            outputStream?.write(data.toByteArray())  
        } catch (e: IOException) {  
            disconnect()  
        }  
    }  
}
```

Resumen del Flujo de Cambio de Sensor:

1. Usuario pulsa "Sensor 1" en HomeActivity.
2. Tu app llama a `BluetoothManager.sendData("1")`.
3. El `BluetoothManager` envía el carácter '1' por Bluetooth.
4. El Arduino recibe el '1' y actualiza su variable `sensorActivo = '1'`, por lo que a partir de ese momento, empezará a leer y enviar los datos del `dht1`.

Mostrar las Medidas de los Sensores (`BluetoothManager.kt` y `DetalleSensorActivity.kt`)

En estos fragmentos de código de los archivos `BluetoothManager.kt` y `DetalleSensorActivity.kt` podemos observar la recepción de los datos enviados por el Arduino.

Parte Clave en `BluetoothManager.kt` (El Receptor y Procesador):

```
private fun listenForData() {
    scope.launch {
        val buffer = ByteArray(1024)
        val messageBuilder = StringBuilder()
        while (_connectionState.value == ConnectionState.CONNECTED) {
            try {
                val bytes = inputStream?.read(buffer) ?: -1
                if (bytes > 0) {
                    messageBuilder.append(String(buffer, 0, bytes))
                    var newlineIndex: Int
                    while (messageBuilder.indexOf('\n').also { newlineIndex = it } != -1) {
                        val line = messageBuilder.substring(0, newlineIndex).trim()
                        messageBuilder.delete(0, newlineIndex + 1)

                        val parts = line.split(';')
                        if (parts.size == 2 && parts[0].isNotEmpty() && parts[1].isNotEmpty()) {
                            _sensorData.value = SensorData(temperature = parts[0], humidity = parts[1])
                        }
                    }
                }
            } catch (e: IOException) {
```

```
        disconnect()
        break
    }
}
}
```

Parte Clave en DetalleSensorActivity.kt (El Observador y Visualizador):

```
@Composable
fun PantallaDetalleSensor(
    sensorId: String,
    bluetoothViewModel: BluetoothViewModel
) {
    val context = LocalContext.current
    var riegoActivado by remember { mutableStateOf(false) }


    // Los datos y el estado se siguen obteniendo del ViewModel
    val connectionState by bluetoothViewModel.connectionState.collectAsState()
    val sensorData by bluetoothViewModel.sensorData.collectAsState()

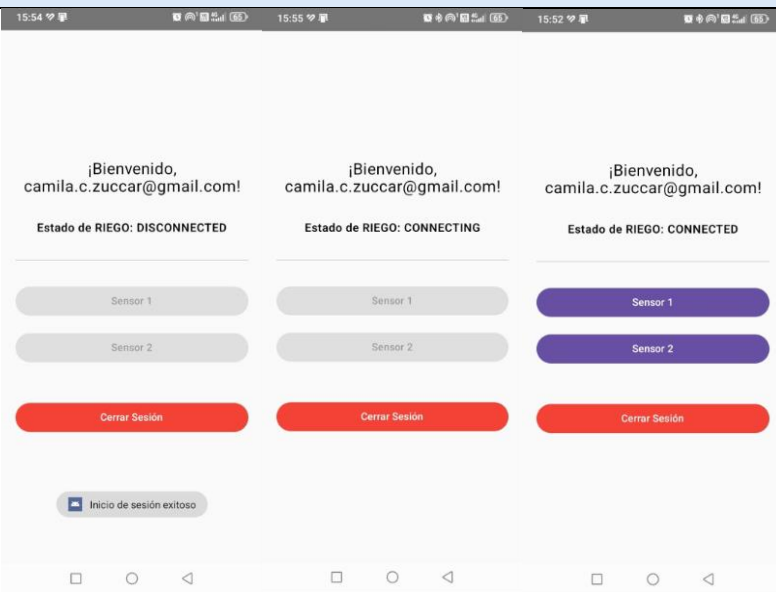
    // Se asume que la conexión ya está hecha. Se envía el ID del sensor al entrar.
    LaunchedEffect(Unit) {
        val sensorIdentifier = if (sensorId == "Sensor 2") "0" else sensorId.filter { it.isDigit() }
        if (sensorIdentifier.isNotEmpty()) {
            bluetoothViewModel.sendData(sensorIdentifier)
        }
    }
}
```

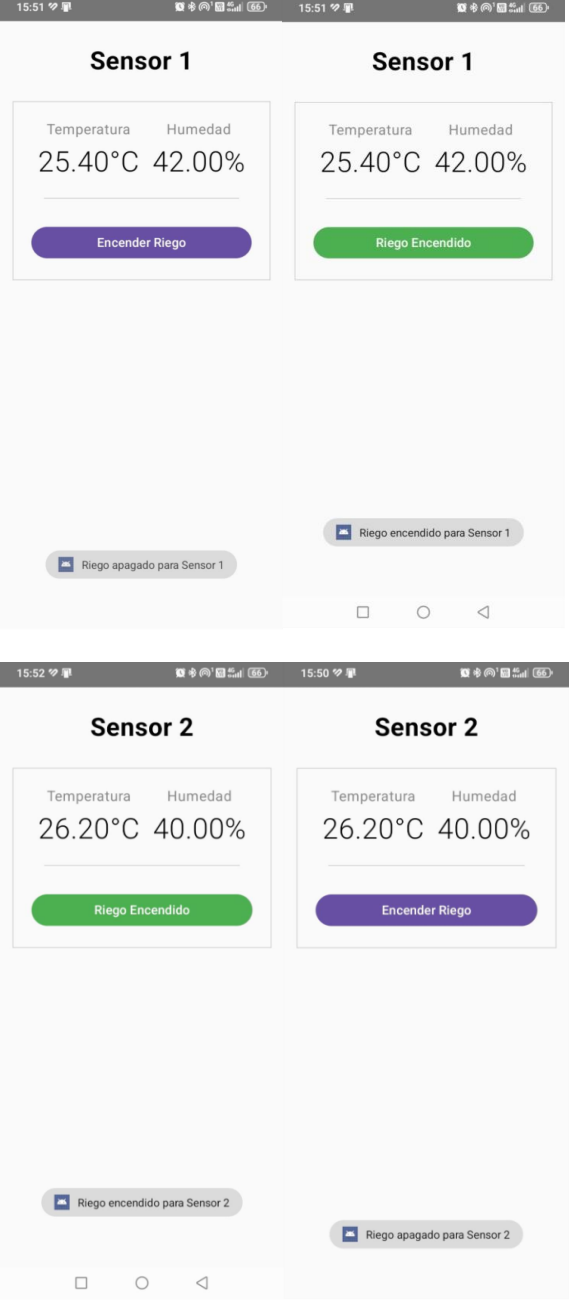
Resumen del Flujo de Visualización de Datos:

- 1.El Arduino lee el sensor activo (ej. dht1) y envía el string "23.4,65.2\n".
- 2.La función listenForData() en BluetoothManager recibe este string.
- 3.Lo divide por la coma , y actualiza el _sensorData con los nuevos valores.
- 4.La PantallaDetalleSensor, que está "observando" los cambios, detecta la actualización.
- 5.Jetpack Compose automáticamente "redibuja" los componentes Text con los nuevos valores de temperatura y humedad.

d. Pruebas y validación

Pruebas al Inicio de Sesión	Explicación
	<p>Se realizaron pruebas a la seguridad y fiabilidad del inicio de sesión de la aplicación colocando datos erróneos y con valores distintos en cada prueba.</p> <p>Finalmente, se ingresaron las credenciales de prueba y se ingresó a la aplicación.</p>

Pruebas de Conexión	Explicación
	<p>Se comprobaron los estados de conexión al módulo Bluetooth, permitiendo el acceso a las otras funciones de la aplicación.</p>

Pruebas de los sensores y actuadores	Explicación
	<p>Aquí verificamos que las medidas enviadas por los sensores a través de la conexión Bluetooth sean correctos y constantes.</p> <p>Además, verificamos que los módulos relé se enciendan o apaguen dependiendo de si se encendió el botón de riego o se apagó. Así determinando el correcto funcionamiento de la aplicación.</p>

e. Escalabilidad

Escalabilidad	Descripción
Agregar más elementos visuales para la lectura y visualización de sensores	Se podrían agregar más elementos visuales para representar los valores de temperatura y humedad que la aplicación recibe, como por ejemplo generar gráficos de los cambios de los valores en el tiempo.
Agregar alertas y monitoreo de riesgos más riguroso	Se podría a futuro agregar alertas o notificaciones visuales y/o auditivas cuando se detectan valores críticos de temperatura y humedad en el entorno.
Agregar consulta al historial de datos	Se podría agregar una interfaz para consultar el historial de mediciones y registros de los valores en un periodo de tiempo y poder generar informes a partir de estos datos.
Agregar interfaz para agregar y configurar nuevos sensores	Se podría agregar interfaz para poder agregar y configurar nuevos sensores que se conectan al sistema.

Análisis de entornos posibles

a. Identificación de entornos

Identificamos tres entornos que la presente solución IoT podría aplicarse: un invernadero, una plantación de pequeña a mediana escala y para cuidado de plantas del hogar.

Un invernadero es un entorno cerrado y controlado donde se cultivan plantas de manera intensiva. Aquí, el control de temperatura y humedad es crucial para maximizar el crecimiento de las plantas. La implementación de una solución IoT en este entorno permitiría monitorear y ajustar en tiempo real las condiciones ambientales, optimizando el uso de recursos como agua y energía.

Mientras tanto, en una plantación pequeña o mediana, la automatización del riego y el monitoreo de condiciones ambientales también sería beneficioso, pero con un enfoque ligeramente diferente al de un invernadero. Este entorno generalmente tiene más exposición a factores climáticos externos, por lo que la integración de sensores y actuadores permitiría ajustar el riego y otros factores en respuesta a condiciones cambiantes del clima.

Y, por último, esta solución aplicada al cuidado de plantas del hogar, aunque de menor escala, también se beneficia del monitoreo y control automatizado de condiciones ambientales. Con una implementación IoT, el usuario podría recibir alertas si la temperatura o humedad de su jardín o macetas se desvían de los niveles ideales, permitiendo un cuidado más eficiente y sencillo de las plantas en interiores o exteriores del hogar.

b. Evaluación de Factibilidad

La implementación de la solución IoT propuesta en los distintos entornos (invernadero, plantación pequeña a mediana escala y cuidado de plantas del hogar) presenta diversas consideraciones tanto técnicas como económicas. A continuación, se realiza un análisis integral de la factibilidad de llevar a cabo el proyecto en cada uno de estos entornos.

Factibilidad Técnica	
Invernadero	El uso de sensores de temperatura y humedad, actuadores para control de riego y sistemas IoT (como Arduino con Bluetooth) es totalmente viable. Sin embargo, en un entorno de invernadero, es fundamental asegurar que los sensores y actuadores sean resistentes a condiciones de alta humedad y temperaturas variables. Además, la conectividad Bluetooth podría tener limitaciones si el invernadero es de gran tamaño; en tal caso, se podría considerar el uso de WiFi o redes LoRa para una mejor cobertura.
Plantación	Técnicamente, el sistema IoT es completamente aplicable, ya que los sensores de temperatura y humedad pueden ser distribuidos a través del área para monitorear múltiples puntos. Sin embargo, la conectividad y la alimentación energética pueden ser

	más complejas en áreas de gran extensión. El uso de sensores de bajo consumo de energía y redes con mayor alcance, como LoRa o 4G, sería ideal para este entorno.
Hogar	Este entorno es el más sencillo de implementar técnicamente. Los sensores son pequeños y fáciles de integrar en sistemas de bajo costo, y la conectividad Bluetooth será suficiente para la comunicación entre el dispositivo móvil y el prototipo. Las restricciones tecnológicas son mínimas en este entorno.
Factibilidad Económica	
Invernadero	Los costos pueden ser mayores debido a la necesidad de sensores más robustos y de mayor alcance, así como una infraestructura más compleja para gestionar el sistema a gran escala. Sin embargo, la reducción en el uso de recursos como agua y energía, y la mejora en los rendimientos del cultivo, justificarían la inversión inicial.
Plantación	Los costos serían intermedios, ya que es posible implementar una solución IoT eficaz utilizando componentes estándar de bajo costo, como sensores DHT y relés. La mayor parte de la inversión se destinaría a asegurar una buena cobertura de red y a la instalación de sensores adicionales.
Hogar	Este entorno sería el más asequible de todos, con una solución IoT más pequeña y accesible en cuanto a costos. Los sensores y actuadores necesarios son relativamente baratos y fácilmente integrables con el dispositivo móvil.

c. Cumplimiento de Estándares OT

El proyecto se alinea con varios estándares de la industria OT (Tecnología Operacional), que buscan garantizar la confiabilidad, seguridad y eficiencia en la operación de sistemas industriales. Algunos aspectos clave incluyen:

- **Monitoreo en Tiempo Real:** El sistema IoT proporciona monitoreo en tiempo real de las variables de temperatura y humedad, lo cual es un estándar en OT para la supervisión continua de procesos críticos. Esto asegura que las condiciones se mantengan dentro de los parámetros deseados.
- **Automatización de Procesos:** Al integrar actuadores de riego, el sistema permite la automatización de procesos, una práctica común en sistemas OT para reducir la intervención manual y mejorar la eficiencia operativa.
- **Integración de Sistemas y Escalabilidad:** La capacidad de agregar más sensores y actuadores a medida que el entorno crece o cambia es un principio fundamental de los sistemas OT, que deben ser flexibles y escalables.
- **Seguridad y Fiabilidad:** Al tratar con tecnologías que afectan directamente la operación de cultivos o jardines, la seguridad y fiabilidad son cruciales. El sistema debe garantizar que no haya fallos en la comunicación o en los controles automáticos, lo que se alinea con las buenas prácticas de la industria OT.

d. Impacto de Limitaciones y Restricciones

Limitaciones Técnicas	
Invernadero	La conectividad Bluetooth puede ser insuficiente para áreas grandes, lo que podría requerir redes más robustas como WiFi o LoRa.
Plantación	Los sensores deben cubrir áreas más grandes y podrían enfrentarse a obstáculos físicos (árboles, cercas) que afecten la conectividad.
Hogar	En este entorno, el alcance de la conectividad Bluetooth no debería ser un problema, pero la precisión de los sensores debe ser adecuada para áreas más pequeñas.
Restricciones	
Invernadero	Los sensores deben ser resistentes a condiciones de alta humedad y temperaturas extremas. Además, la integración del sistema en un entorno cerrado requiere garantizar que los sensores no se vean afectados por los cambios repentinos de temperatura o la condensación.
Plantación	La variabilidad del clima podría afectar la precisión de los datos si los sensores no están distribuidos adecuadamente.
Hogar	Las condiciones estables del hogar permiten un uso más sencillo de los sensores, pero la solución debe ser compacta y fácil de instalar, sin afectar la estética.

e. Pruebas y Validación

Invernadero: Para garantizar el correcto funcionamiento del sistema IoT en un invernadero, se aplicarán las siguientes pruebas específicas.	
Pruebas de Conectividad de Red	Dado que en el invernadero es probable que el alcance de Bluetooth sea insuficiente, se realizará una prueba de conectividad de red utilizando WiFi o LoRa. Esta prueba asegurará que la red de comunicación elegida tenga el alcance necesario para cubrir todo el espacio sin pérdidas de señal.
Pruebas de Resistencia a Condiciones Ambientales	Los sensores y actuadores estarán expuestos a altas temperaturas y niveles elevados de humedad. Se llevarán a cabo pruebas de resistencia ambiental para verificar que los dispositivos continúen operando correctamente bajo estas condiciones extremas.
Pruebas de Calibración de Sensores:	Debido a la importancia de obtener mediciones precisas de temperatura y humedad, se realizarán pruebas de calibración de sensores para asegurar que los dispositivos estén proporcionando datos precisos y consistentes en el entorno del invernadero.
Plantación: En una plantación pequeña a mediana escala, se realizarán las siguientes pruebas.	
Pruebas de Cobertura de Red	Similar al invernadero, se realizará una prueba de cobertura de red utilizando LoRa para asegurarse de que la comunicación entre sensores y la aplicación móvil sea confiable, especialmente en áreas de gran extensión y con posibles obstáculos físicos.
Pruebas de Consumo Energético	Como se utilizarán sensores de bajo consumo y posiblemente paneles solares, se llevarán a cabo pruebas de autonomía energética para comprobar que los sensores funcionen de manera eficiente durante largos períodos sin necesidad de recarga o intervención manual.
Pruebas de Integración de Componentes	En este entorno, se realizará una prueba de integración para verificar que todos los sensores y actuadores se comuniquen correctamente entre sí y con la aplicación móvil, sin interferencias o fallos de comunicación.
Hogar: Para el cuidado de las plantas del hogar, se aplicarán las siguientes pruebas.	
Pruebas de Instalación y Configuración Rápida	Dado que la solución está dirigida a usuarios sin conocimientos técnicos, se realizarán pruebas de facilidad de instalación para asegurar que los sensores sean fáciles de conectar y configurar en el entorno doméstico.

Pruebas de Notificaciones y Alertas	Se validará el sistema de notificaciones en tiempo real, asegurándose de que los usuarios reciban alertas precisas sobre el estado de sus plantas, como cuando la temperatura o humedad excedan los rangos óptimos.
Pruebas de Usabilidad	Se realizarán pruebas de usabilidad para verificar que la aplicación móvil sea intuitiva y fácil de usar, permitiendo a los usuarios monitorear las condiciones ambientales y tomar decisiones rápidamente sin complicaciones.

f. Recomendaciones

En el caso del invernadero, se recomienda utilizar redes de comunicación más robustas, como WiFi o LoRa, para superar las limitaciones de alcance de la tecnología Bluetooth, que podría no ser suficiente en grandes espacios cerrados. Además, es fundamental asegurarse de que los sensores utilizados sean resistentes a las condiciones extremas de humedad y temperatura, características comunes en este tipo de entornos, para garantizar su funcionamiento continuo y confiable.

Para una plantación pequeña a mediana escala, se sugiere emplear sensores de bajo consumo energético que permitan operar de manera eficiente durante períodos prolongados sin necesidad de un mantenimiento frecuente. También se recomienda considerar el uso de fuentes de energía alternativas, como paneles solares, para alimentar los sensores y actuadores, lo que haría más sostenible la solución. Además, se debe integrar una red de comunicación más robusta, como LoRa, que asegure la confiabilidad de la comunicación en áreas extensas y sin obstáculos, proporcionando una cobertura adecuada en el terreno.

En el cuidado de las plantas del hogar, es clave mantener la solución simple y accesible, utilizando sensores compactos y fáciles de instalar, lo que facilita su implementación por parte de usuarios no técnicos. También es importante implementar un sistema de notificaciones que permita a los usuarios recibir alertas sobre el estado de sus plantas, ayudándoles a mantenerlas en condiciones óptimas sin necesidad de intervención constante, lo que aporta comodidad y eficiencia al proceso de cuidado.

Conclusiones

El desarrollo de este sistema IoT para monitorear la temperatura y humedad en entornos agrícolas ha avanzado significativamente, logrando establecer las bases para una solución escalable y accesible para una variedad de entornos, desde invernaderos hasta pequeños jardines domésticos. La integración de Arduino con sensores y actuadores, conectados a una aplicación móvil vía Bluetooth, ha permitido ofrecer una solución efectiva que no solo optimiza el control ambiental en espacios agrícolas, sino que también simplifica la gestión de recursos como agua y energía. La elección de Bluetooth como método de comunicación ha sido adecuada para entornos domésticos y pequeños, pero a medida que el proyecto evoluciona hacia implementaciones más grandes (como invernaderos o plantaciones medianas), se ha identificado la necesidad de explorar redes más robustas como WiFi o LoRa para mejorar la cobertura y fiabilidad de la comunicación.

Uno de los aspectos más destacados de este trabajo ha sido la integración de la automatización, especialmente la capacidad de activar o desactivar el sistema de riego de manera remota desde la aplicación móvil. Este componente, que responde a las condiciones de los sensores en tiempo real, representa un avance importante para la eficiencia operativa, permitiendo una respuesta inmediata ante condiciones críticas como temperaturas o humedades fuera de rango. A nivel técnico, se ha considerado el uso de sensores de bajo consumo energético como una medida clave para la eficiencia del sistema, y se ha planteado a futuro la opción de utilizar fuentes de energía renovables, como paneles solares, en el futuro para mejorar la sostenibilidad en entornos con grandes demandas energéticas o cuando se busque una mayor escalabilidad, como en plantaciones de mayor escala.

A lo largo del proceso, se han identificado áreas de profundización que podrían llevar a mejoras sustanciales en el proyecto. Una de estas áreas es la optimización de la conectividad en grandes entornos, donde la solución actual podría necesitar una revisión en términos de alcance de señal. La opción de integrar tecnologías de red de largo alcance como LoRa se presenta como una alternativa prometedora, especialmente en plantaciones más grandes. Además, en cuanto a la interfaz de usuario de la aplicación, aún queda espacio para mejorar la experiencia del usuario, simplificando los procesos de configuración y gestión, y ofreciendo más opciones personalizadas para usuarios con diferentes niveles de conocimiento técnico.

Este proyecto también ha dejado en evidencia el gran potencial de las soluciones IoT en el sector agrícola, especialmente en el contexto de la automatización y el monitoreo remoto. En el futuro, sería interesante explorar la integración con otras tecnologías como sensores de calidad del aire, sistemas de predicción meteorológica o incluso inteligencia artificial para optimizar aún más la gestión de los recursos y prevenir condiciones futuras.

En conclusión, este trabajo ha proporcionado una solución funcional y efectiva para el monitoreo ambiental en entornos agrícolas, con la capacidad de ser escalada y mejorada según las necesidades de cada usuario. Sin embargo, a medida que se avanza hacia una implementación más amplia, será esencial seguir explorando nuevas tecnologías y adaptando el sistema a las realidades de los diferentes entornos de aplicación. La integración de la automatización y la conectividad, junto con el potencial de personalización y expansión, hace de este proyecto una base sólida para futuros desarrollos que pueden transformar la forma en que gestionamos los recursos en el sector agrícola.