[Fecha]

Iván Bezanilla López

IES Alisal

Sistema de Riego Automatizado con Arduino

Contenido

[PRESENTACIÓN 3](#_Toc167782486)

[CONTEXTO DEL PROYECTO 4](#_Toc167782487)

[OBJETIVOS 5](#_Toc167782488)

[Objetivos Principales: 5](#_Toc167782489)

[Objetivos Secundarios: 5](#_Toc167782490)

[ESQUEMA DE LA SOLUCIÓN 7](#_Toc167782491)

[Diagrama general: 7](#_Toc167782492)

[Diagrama de conexión de hardware: 9](#_Toc167782493)

[Diagrama de la estructura de AWS 12](#_Toc167782494)

[Diagrama entidad/relación de la Base de Datos 14](#_Toc167782495)

[MATERIALES, RECURSOS y SERVICIOS NECESARIOS 15](#_Toc167782496)

[Hardware: 15](#_Toc167782497)

[Software: 20](#_Toc167782498)

[PRESUPUESTO DETALLADO 24](#_Toc167782499)

[PLANIFICACIÓN DEL PROYECTO 25](#_Toc167782500)

[DESARROLLO TÉCNICO DEL PROYECTO 26](#_Toc167782501)

[Configuración del Hardware 26](#_Toc167782502)

[Ensamblaje y Cableado de los Componentes 26](#_Toc167782503)

[Desarrollo del Código para Arduino 26](#_Toc167782504)

[Implementación en Entorno Local 33](#_Toc167782505)

[Preparación del Entorno de Desarrollo (Instalación de Ubuntu 22.04) 33](#_Toc167782506)

[Configuración del Servidor Web (Instalación de Apache, PHP y MySQL) 39](#_Toc167782507)

[Integración de MQTT con WebSocket (Instalación de Mosquitto) 45](#_Toc167782508)

[Desarrollo de la Página Web y Base de Datos 49](#_Toc167782509)

[Migración a la Nube (AWS) 61](#_Toc167782510)

[Creación de una Infraestructura Virtual Privada (VPC) 61](#_Toc167782511)

[Configuración de Grupos de Seguridad 64](#_Toc167782512)

[Implementación de Instancias EC2 67](#_Toc167782513)

[Conexión a la instancia EC2 e instalación de servicios 71](#_Toc167782514)

[Configuración de IP Elástica 73](#_Toc167782515)

[Configuración de NO-IP 75](#_Toc167782516)

[Configurar HTTPS para apache 77](#_Toc167782517)

[Configuración de un Servicio de Base de Datos Relacional (RDS) 79](#_Toc167782518)

[Transición de la Página Web a la Nube (EC2) 85](#_Toc167782519)

[Migración de la Base de Datos a la Nube (RDS) 87](#_Toc167782520)

[Verificación y Validación 88](#_Toc167782521)

[Pruebas de Funcionamiento 88](#_Toc167782522)

[Confirmación del Éxito del Proyecto 88](#_Toc167782523)

[ANÁLISIS DE RIESGOS 89](#_Toc167782524)

[Fortalezas 89](#_Toc167782525)

[Debilidades 90](#_Toc167782526)

[Oportunidades 90](#_Toc167782527)

[Amenazas 90](#_Toc167782528)

[BIBLIOGRAFÍA 91](#_Toc167782529)

# PRESENTACIÓN

GreenTech Solutions se enorgullece en presentar su último proyecto: el Sistema de Riego Automatizado con Arduino. Esta innovadora solución fusiona tecnología de vanguardia con la pasión de la empresa por la agricultura y la sostenibilidad, abriendo nuevas puertas hacia una forma más inteligente y eficiente de cuidar las plantas.

El sistema se destaca por su capacidad para monitorear de manera precisa y continua las condiciones del suelo y del ambiente, gracias a la integración de múltiples sensores. Estos datos se almacenan y analizan en una base de datos MySQL, proporcionando información detallada que permite optimizar el riego y mejorar el crecimiento de las plantas.

Además, se ha desarrollado una interfaz web interactiva alojada en un servidor Apache, que brinda acceso en tiempo real a los datos de los sensores y permite el control remoto del sistema de riego desde cualquier lugar. Esta facilidad de acceso y control intuitivo se traduce en un cuidado más eficiente de las plantas, facilitando la vida tanto en hogares como en empresas agrícolas.

La comunicación entre el Arduino Uno y Apache se logra a través de MQTT, asegurando una transmisión rápida y confiable de datos. Otro aspecto destacado del sistema es la bomba de agua controlada por Arduino, que ofrece la flexibilidad de personalizar la duración y frecuencia del riego según las necesidades específicas de cada planta.

Los beneficios del Sistema de Riego Automatizado son abundantes. Desde la optimización del uso del agua hasta la reducción de costos y tiempo dedicado al riego manual, pasando por la contribución a la conservación del medio ambiente al evitar el desperdicio de agua y promover prácticas de riego más sostenibles, GreenTech Solutions está comprometido con un futuro más verde y próspero para todos.

# CONTEXTO DEL PROYECTO

El proyecto de desarrollo de un sistema de riego automatizado con Arduino se enmarca en la necesidad de encontrar soluciones innovadoras y sostenibles para optimizar el cuidado de las plantas en entornos tanto domésticos como empresariales.

Con la preocupación creciente por el uso eficiente del agua y la conservación del medio ambiente, GreenTech Solutions se ha embarcado en la misión de desarrollar una tecnología que no solo simplifique el proceso de riego, sino que también promueva prácticas agrícolas más responsables.

Este proyecto se llevará a cabo en dos escenarios distintos, cada uno adaptado a las necesidades y requisitos específicos de los usuarios involucrados: personas individuales y empresas agrícolas.

Para personas individuales, el sistema de riego automatizado ofrecerá una solución práctica y accesible para el cuidado de jardines, huertos o áreas verdes en hogares y comunidades residenciales. La interfaz intuitiva y el control remoto a través de dispositivos móviles permitirán a los usuarios monitorear y gestionar el riego de sus plantas de manera conveniente y eficiente, incluso cuando no estén en casa.

Por otro lado, en el contexto de empresas agrícolas, el sistema de riego automatizado se presenta como una herramienta indispensable para la optimización de procesos y la mejora de la productividad. Con la capacidad de adaptarse a diferentes tipos de cultivos y tamaños de terreno, esta tecnología permitirá a los agricultores controlar el riego de manera precisa y eficiente, maximizando el rendimiento de sus cosechas mientras minimizan el uso de recursos como el agua y la energía.

En ambos escenarios, el proyecto se basa en principios de innovación, eficiencia y sostenibilidad. GreenTech Solutions está comprometido con el desarrollo de soluciones tecnológicas que no solo mejoren la calidad de vida de las personas, sino que también contribuyan a la preservación del medio ambiente y la creación de un futuro más verde y próspero para todos.

# OBJETIVOS

## Objetivos Principales:

1. Desarrollar un sistema de riego automatizado con Arduino funcional para entornos individuales.
   * A corto plazo: Investigar y seleccionar los sensores adecuados para monitorear las condiciones del suelo y del ambiente.
   * A medio plazo: Diseñar e implementar el circuito electrónico que permita la comunicación entre los sensores, Arduino y la bomba de agua.
   * A largo plazo: Realizar pruebas integrales del sistema en entornos individuales, ajustar y optimizar el funcionamiento según los resultados obtenidos.
2. Implementar una interfaz web interactiva que muestre los valores de los sensores en tiempo real.
   * A corto plazo: Diseñar la estructura y la interfaz de usuario de la aplicación web.
   * A medio plazo: Desarrollar y conectar la interfaz web con la base de datos MySQL para almacenar y recuperar los datos de los sensores.
   * A largo plazo: Realizar pruebas de integración entre el sistema de riego y la interfaz web, asegurando la correcta visualización de los datos.

## Objetivos Secundarios:

1. Habilitar la funcionalidad desde la interfaz web para activar el riego durante los segundos indicados.
   * A corto plazo: Implementar la lógica de control en la interfaz web para permitir la activación remota del riego.
   * A medio plazo: Integrar la lógica de control con el sistema de riego automatizado, asegurando la sincronización y la precisión en la activación del riego.
   * A largo plazo: Realizar pruebas exhaustivas de la funcionalidad de activación remota del riego, corrigiendo posibles errores y mejorando la experiencia del usuario.
2. Adaptar el sistema de riego automatizado para entornos empresariales.
   * A corto plazo: Identificar las necesidades específicas de las empresas agrícolas y las características adicionales requeridas para el sistema.
   * A medio plazo: Desarrollar módulos adicionales o funcionalidades específicas para satisfacer los requisitos empresariales, como la gestión de múltiples zonas de riego.
   * A largo plazo: Realizar pruebas piloto del sistema adaptado en empresas agrícolas, recopilando comentarios y retroalimentación para realizar ajustes finales.
3. Diseñar un sistema de control inteligente que ajuste automáticamente el riego según las necesidades de las plantas.
   * A corto plazo: Investigar algoritmos de control y sistemas de reglas para la gestión eficiente del riego.
   * A medio plazo: Implementar el sistema de control inteligente en el software del Arduino, utilizando los datos recopilados por los sensores.
   * A largo plazo: Realizar pruebas de funcionamiento del sistema de control inteligente, evaluando su capacidad para adaptarse dinámicamente a las condiciones cambiantes del entorno.

En resumen, los objetivos principales del proyecto se centran en el desarrollo y la implementación exitosa del sistema de riego automatizado, mientras que los objetivos secundarios se enfocan en aspectos como la usabilidad, la eficiencia del riego y la adaptabilidad del sistema a las necesidades de los usuarios. Estos objetivos se han establecido con plazos claros a corto, medio y largo plazo para garantizar un progreso constante y medible a lo largo del proyecto.

# ESQUEMA DE LA SOLUCIÓN

## Diagrama general:

El presente documento tiene como objetivo realizar una detallada descripción del sistema de comunicación de datos presente en la imagen proporcionada, analizando sus componentes, flujos de información y funcionalidades.

**Componentes del sistema**

El sistema se compone de los siguientes elementos:

* **Arduino Uno:** Placa microcontroladora que sirve como núcleo del sistema, recopilando datos de los sensores y enviándolos al servidor MQTT.
* **Sensores:**
  + Sensor DHT11: Mide la temperatura y la humedad del aire.
  + Sensor de humedad del suelo: Mide la humedad del suelo.
* **Servidor MQTT:** Instancia EC2 en la nube de Amazon Web Services (AWS) que recibe los datos de los sensores, los procesa y los distribuye a otros componentes.
* **Apache:** Servidor web que aloja la aplicación web para visualizar los datos de los sensores.
* **PHP:** Lenguaje de programación que interactúa con la base de datos MySQL para almacenar los datos de los sensores.
* **MySQL:** Base de datos relacional que almacena los datos de los sensores en una tabla.
* **Interfaz web:** Página web que permite visualizar los datos de los sensores en tiempo real y establecer el tiempo de activación de la bomba de agua.
* **Bomba de agua:** Dispositivo que se activa en función del tiempo establecido en la interfaz web, controlando el riego de las plantas.

**Flujos de información**

El sistema funciona mediante la interacción de sus componentes y el intercambio de información de la siguiente manera:

1. **Recolección de datos:** Los sensores DHT11 y de humedad del suelo envían sus datos al Arduino Uno.
2. **Envío a servidor MQTT:** El Arduino Uno envía los datos de los sensores al servidor MQTT mediante el protocolo MQTT.
3. **Procesamiento y distribución:** El servidor MQTT recibe los datos de los sensores, los procesa y los distribuye a Apache y a la interfaz web.
4. **Visualización en la web:** Apache recibe los datos procesados del servidor MQTT y los utiliza para generar la interfaz web que muestra los datos de los sensores en tiempo real.
5. **Almacenamiento en base de datos:** PHP interactúa con la base de datos MySQL para almacenar los datos de los sensores en una tabla específica.
6. **Control de la bomba de agua:** La interfaz web permite al usuario establecer un tiempo en segundos para la activación de la bomba de agua. El servidor MQTT recibe este tiempo y lo envía al Arduino Uno.
7. **Activación de la bomba:** El Arduino Uno recibe el tiempo de activación de la bomba de agua del servidor MQTT y activa la bomba durante el tiempo especificado.

**Funcionalidades del sistema**

El sistema de comunicación de datos presentado ofrece las siguientes funcionalidades:

* **Monitoreo remoto de datos de sensores:** Permite visualizar en tiempo real los valores de temperatura, humedad del aire y humedad del suelo desde una interfaz web.
* **Almacenamiento histórico de datos:** Almacena los datos de los sensores en una base de datos para su análisis posterior y la toma de decisiones informadas sobre el riego de las plantas.
* **Control remoto de la bomba de agua:** Permite al usuario establecer un tiempo específico para la activación de la bomba de agua, automatizando el proceso de riego.
* **Escalabilidad:** El sistema puede ampliarse fácilmente para incluir más sensores y dispositivos, adaptándose a las necesidades específicas de cada proyecto.



## Diagrama de conexión de hardware:

Este diagrama detallará cómo se conectan físicamente los componentes del hardware.

Mostrará cómo se conectan los sensores (DHT11, sensor de humedad del suelo) al Arduino Uno, cómo se conecta el módulo ESP8266-01S al Arduino Uno y cómo se conecta la mini bomba de agua al relé conectado al Arduino Uno.

El presente documento tiene como objetivo realizar una detallada descripción del circuito electrónico presente en la imagen proporcionada, analizando sus componentes, conexiones, funcionamiento, aplicaciones, posibles mejoras y recursos adicionales.

**Componentes**

El circuito se compone de los siguientes elementos:

* **Arduino Uno:** Placa microcontroladora que sirve como núcleo del sistema, procesando y ejecutando las instrucciones del programa.
* **Sensor DHT11:** Sensor ambiental que mide la temperatura y la humedad del aire, proporcionando datos valiosos para el control del sistema.
* **Sensor de humedad del suelo:** Sensor especializado en medir la humedad del suelo, permitiendo determinar la necesidad de riego para las plantas.
* **ESP-01:** Módulo WiFi que posibilita la conexión del circuito a internet, enviando los datos de los sensores a un servidor web para su visualización y análisis.
* **Relé:** Componente electrónico que actúa como interruptor controlado eléctricamente, permitiendo encender y apagar la bomba de agua.
* **Bomba de agua:** Dispositivo que impulsa el agua para el riego de las plantas, activada por el relé.

**Conexiones**

Las conexiones entre los componentes se establecen de la siguiente manera:

**Sensor DHT11:**

VCC a 3.3V del Arduino: Proporciona alimentación al sensor.

GND a GND del Arduino: Establece la conexión a tierra para la referencia de voltaje.

DATA a pin 10 del Arduino: Transfiere los datos de temperatura y humedad al Arduino.

**Sensor de humedad del suelo:**

VCC a 5V del Arduino: Suministra alimentación al sensor.

GND a GND del Arduino: Referencia de voltaje para las mediciones.

DATA a pin A0 del Arduino: Envía los valores de humedad del suelo al Arduino.

**ESP-01:**

VCC a 3.3V del Arduino: Alimentación para el módulo WiFi.

GND a GND del Arduino: Referencia de tierra.

CH\_PD a 3.3V del Arduino: Enciende y apaga el módulo WiFi.

TX a pin 2 del Arduino: Envía datos desde el ESP-01 al Arduino.

RX a pin 3 del Arduino: Recibe datos del Arduino al ESP-01.

**Relé:**

VCC a 5V del Arduino: Alimentación para el relé.

GND a GND del Arduino: Referencia de tierra.

IN1 a pin 9 del Arduino: Recibe la señal de control del Arduino para activar o desactivar el relé.

COM a bomba de agua: Conecta el relé a la bomba de agua.

**Bomba de agua:**

COM del relé: Conexión positiva de la alimentación de la bomba.

GND del Arduino: Conexión negativa de la alimentación de la bomba.

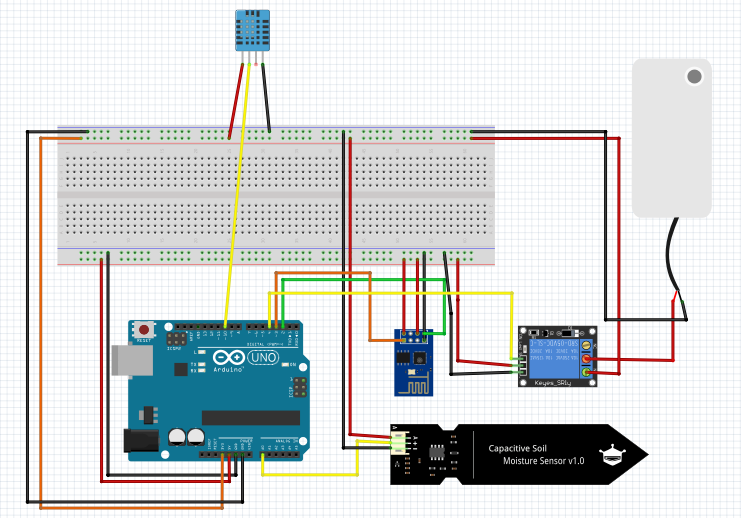
**Arduino Uno:**

5V a protoboard: Conexión positiva de 5V.

GND a protoboard: Conexión negativa.

3.3V a protoboard: Conexión positiva de 3.3V.

GND a protoboard: Conexión negativa.



## Diagrama de la estructura de AWS

En esta presentación, se explicará el diseño de una red en AWS que utiliza algunos de los servicios de red de AWS.

**Componentes principales**

La red se compone de los siguientes componentes principales:

* **Amazon Virtual Private Cloud (VPC)**: Una VPC es una red privada virtual que te permite aislar tus recursos de AWS de la red pública de Internet. Esto te ayuda a proteger tus recursos de accesos no autorizados.
* **Subredes**: Las subredes son divisiones lógicas de una VPC. Puedes utilizar subredes para segmentar tu red y controlar el tráfico entre diferentes tipos de recursos.
* **Instancias EC2**: Las instancias EC2 son servidores virtuales que puedes ejecutar en la nube de AWS. Puedes utilizar instancias EC2 para ejecutar aplicaciones, almacenar datos y mucho más.
* **Amazon RDS**: Amazon RDS es un servicio de base de datos administrado que te permite ejecutar bases de datos relacionales en la nube de AWS. Amazon RDS se encarga de la administración de la base de datos, como las actualizaciones de software, las copias de seguridad y la protección contra fallos.
* **Puerta de enlace de Internet**: Una puerta de enlace de Internet es un dispositivo que te permite conectar tu VPC a Internet. Esto te permite que tus recursos de AWS sean accesibles desde Internet.

**Flujo de tráfico**

El flujo de tráfico en la red se puede describir de la siguiente manera:

* Los usuarios de Internet pueden acceder a la base de datos RDS a través de la dirección IP pública de la subred pública. Esto permite que los usuarios de Internet se conecten a la base de datos y ejecuten consultas.
* Las instancias EC2 pueden acceder a la base de datos RDS a través de la dirección IP privada de la base de datos. Esto permite que las instancias EC2 se conecten a la base de datos y ejecuten consultas.
* Las instancias EC2 pueden acceder entre sí a través de las direcciones IP privadas dentro de la VPC. Esto permite que las instancias EC2 se comuniquen entre sí sin tener que salir a Internet.

**Consideraciones de seguridad**

La arquitectura de red de la imagen incluye algunas medidas de seguridad básicas, como:

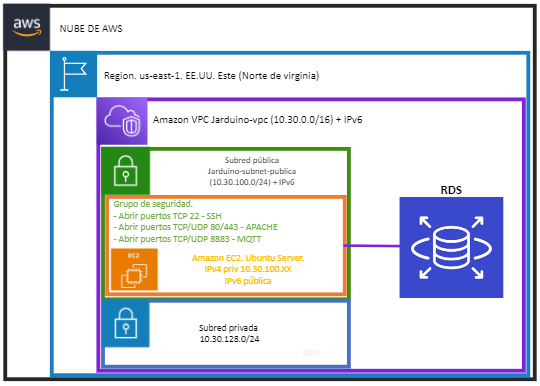
* **Grupo de seguridad**: El grupo de seguridad de la instancia EC2 permite el acceso a los puertos que son necesarios para que la instancia funcione.
* **Subredes privadas**: Los recursos de la zona de red privada no están directamente expuestos a Internet.

Sin embargo, la arquitectura podría mejorarse con la implementación de medidas de seguridad adicionales, como:

* **Lista de control de acceso (ACL)**: Una ACL es una lista de reglas que permiten o deniegan el tráfico de red a una instancia EC2. Puedes utilizar ACL para restringir aún más el acceso a los puertos de la instancia EC2.
* **Cifrado de datos**: Los datos en reposo y en tránsito deben estar cifrados para protegerlos de accesos no autorizados.

**Conclusión**

El diseño de red que se muestra en la imagen es una buena base para una red en AWS. Sin embargo, se pueden realizar algunas mejoras para mejorar la seguridad de la red.

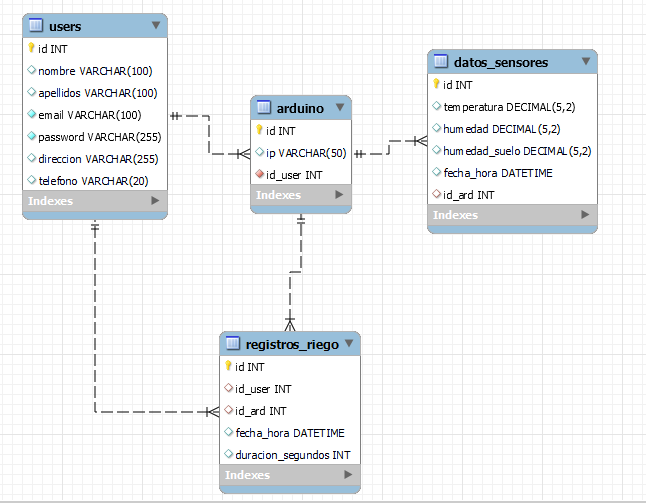


## Diagrama entidad/relación de la Base de Datos

El siguiente diagrama entidad-relación proporciona una representación visual de la estructura y las interrelaciones fundamentales que subyacen en la base de datos del proyecto. Este esquema gráfico es una herramienta invaluable para comprender la organización y la arquitectura de la base de datos, ya que muestra claramente las entidades principales, sus atributos y las conexiones entre ellas.

Al analizar el diagrama, se pueden identificar fácilmente las entidades clave del sistema, como 'arduino', 'datos\_sensores', 'registros\_riego' y 'users', así como sus atributos asociados, como 'ip', 'temperatura', 'humedad', 'fecha\_hora', entre otros. Además, se visualizan las relaciones entre estas entidades, como las claves foráneas que vinculan 'arduino' con 'users' y 'datos\_sensores', y 'registros\_riego' con 'users' y 'arduino'.

Esta representación gráfica no solo proporciona una comprensión clara de la estructura de la base de datos, sino que también ayuda en el diseño y la implementación eficiente de las funcionalidades del sistema.



# MATERIALES, RECURSOS y SERVICIOS NECESARIOS

## Hardware:

**Arduino Uno R3:**

Justificación de compra: Arduino Uno es la plataforma de desarrollo principal para el sistema de riego. Su facilidad de uso, amplia disponibilidad de bibliotecas y capacidad de interacción con una variedad de sensores y actuadores lo convierten en la opción ideal para proyectos de electrónica y automatización.

Características:

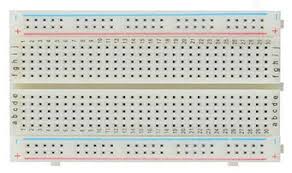
* + Microcontrolador ATmega328.
  + Características: Puerto USB, 14 pines digitales de entrada/salida (6 de ellos pueden ser PWM), 6 entradas analógicas, velocidad de reloj de 16 MHz.
  + Precio: 22,99 € (IVA incluido)
  + [Enlace](https://www.amazon.es/Arduino-UNO-A000066-microcontrolador-ATmega328/dp/B008GRTSV6/ref=sr_1_5?crid=3QBQVS22GUC7A&dib=eyJ2IjoiMSJ9.N8nF-4nrSGrUBX0eJZ8DQ3faJEQ6oU7jDfwMD55RC6PHY2xj3N9JePg85PFtW5v-nG1TLbyHvS9oNR5mPbB6sJNW75nQi8cvNy4kP0iT2t9zmy-F27ZnrA6bcqa8AZZNar1W1C-38gvaebG-VIpUeLuLxk2SnyFVRtfiXKLinuQd5iS-ovK5JxbNKNFDLE2FR01vFOMqDdFPUTTZQ6u4iDywcYGhDjz4-p_AGgxXiMcNd0uKmJsMRmqdGunroAgjvu3sF0CX8juMP2dBEHezR08bpnT6785feExFtGKqQeU.Xk6fvyzP2I-TnIgTqIMWK7zQALHMukrGGGX1qA2FSmo&dib_tag=se&keywords=arduino+uno+r3&qid=1713425949&sprefix=arduino+%2Caps%2C102&sr=8-5)

**Protoboard:**

Justificación de compra: La protoboard se utiliza para prototipar y conectar los componentes electrónicos del sistema de riego de forma temporal. Su diseño sin soldadura y su facilidad de uso permiten experimentar con diferentes configuraciones antes de la implementación final.

Características:

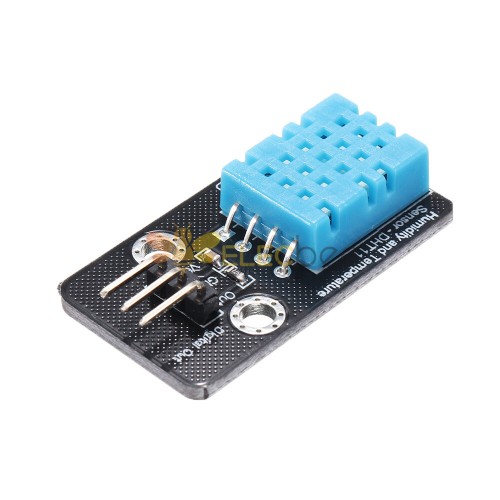
* + Incluye barras conductoras para facilitar la conexión de componentes.
  + Dimensiones: 16.5 cm x 5.5 cm.
  + Precio: 5,49 € (IVA incluido)
  + [Enlace](https://www.amazon.es/AZDelivery-Breadboard-barras-conductoras-Arduino/dp/B07KKJSFM1/ref=sr_1_5?crid=2F0BISXJE3DXY&dib=eyJ2IjoiMSJ9.qBPp7lHWWelJcqFJQYHaU0QuRzF7SqCy4PDW4Vus6H3i6FWCMCrJcUbWfocvBWiSrcpqZsns_aj-tGxVm46qMedXsErejA0wjcfdOf1kCLyJ2JrjAkkNHAGPXJnBufM0rvlByYZu-z--8kQSHVK2sqW7Pl0UG4Y7npkQe6TopjizWjn-UbA2-4b2jBGyobo__nowEZfJylMCyXCy5eWudUgpD3jMlgkeemq3v0SgJNp5GFHzYSsZk8ThjJNGFy9rRxNO2spD0ZbTRmkybW_gMfZychMC0zJ6p-fOY6aJTSI.JJS4vaJrBGw3FZS3Cp4WbMK3sYPlUuD8YwZCV7fqLIg&dib_tag=se&keywords=placa+de+pruebas+arduino&qid=1713428538&sprefix=placa+de+pruebas+ar%2Caps%2C103&sr=8-5)



**Sensor DHT11:**

Justificación de compra: El sensor DHT11 mide la temperatura y la humedad del aire, lo que proporciona datos importantes para ajustar el riego en función de las condiciones ambientales. Su bajo costo y su facilidad de uso lo hacen ideal para este propósito.

Características:

* + Sensor digital de temperatura y humedad.
  + Rango de temperatura: 0°C a 50°C.
  + Rango de humedad: 20% a 90% HR.
  + Precio: 5,99 € (IVA incluido)
  + [Enlace](https://www.amazon.es/AZDelivery-Temperatura-Compatible-Arduino-Incluido/dp/B089W8DB5P/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.J8_7o_7nkjidzjJwsOQSVGJTk_DI7naVgLCPtovUVJ7ObflO0e2acy4CTzlM1qZjY0EkKvQ3zgaUXruE6kQBgGJ88zYS0NGs-7J0HgddAVIUDNs8NCl-rW76F0seUGP0QFqYYo8JFtAgv5lILF93f1ZU2sPjxq6MfthSiuoTU9ndeVl0DgVSUwDLCXc5q7lzvQo6QVhoXGwiRi3qs5-6Yd-BooWL4U1QcterRJG73n3VAkXL0sxmcViU5cHgBCCbTmR61kVS9GCb1kN1u747vpsdhJK65mYaTMc6M_XHMho.GTL45kmFU2Jy-XPbw0l0qWKS8N4lllHWz6roPgCiqD4&dib_tag=se&keywords=dht11&qid=1713425766&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&th=1)

**Sensor de humedad del suelo:**

Justificación de compra: El sensor de humedad del suelo detecta los niveles de humedad en el suelo y activa el riego cuando es necesario. Su precisión y fiabilidad son fundamentales para garantizar un suministro adecuado de agua a las plantas.

Características:

* + Módulo higrómetro V1.2.
  + Voltaje de operación: 3.3V - 5V.
  + Salida analógica.
  + Precio: 7,99 € (IVA incluido)
  + [Enlace](https://www.amazon.es/AZDelivery-Sensor-Humedad-Suelo-Modulo_higrometro_V1-2_Parent/dp/B07YY2RN1B)

**Relé:**

Justificación de compra: El relé se utiliza para controlar la activación y desactivación de la bomba de agua u otros dispositivos de alta potencia. Su capacidad para conmutar entre circuitos de baja y alta potencia lo convierte en una opción segura y confiable para el control del sistema de riego.

Características:

* + Módulo KY-019 con relé de un canal.
  + Capacidad de conmutación: 250VAC/10A, 125VAC/10A, 30VDC/10A.
  + Precio: 5,99 € (IVA incluido)
  + [Enlace](https://www.amazon.es/ARCELI-KY-019-M%C3%B3dulo-Shield-arduino/dp/B07BVXT1ZK/ref=sr_1_5?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=24W9AAL2CKL&dib=eyJ2IjoiMSJ9.aLr9rUAXO6XYld128fwZUOdseJrgSo9EAxvVFD_K-nFhk00N-3LbBSYSuUje4iuGtdhRQoEyxGQiDVbscf2wGVuJ3yAq_iY_EryjJ5TLalakasEeVDkrV-xpzMeI-oqcOnuxC8Ht8VkP4gqMpf2H_rSNLWO-GoDRp6tPvrII9WD4GDkitzA8bEglqU7wiBYLGTg9ytEVr-fjD1b0bSAfUwKcCuBYVNym5MIQH0Gxy1hPxchIhvcAhRFToRfigwripqTth0ZZRHf-9gPuo5LbFD5xTL4KiIkjbOVoZQeztoQ.svSW2iXIKFmMxjKUAOkymVg7f7iUjMCoQftV6Og2Zvg&dib_tag=se&keywords=rele+arduino&qid=1713425822&sprefix=rele+arduino%2Caps%2C98&sr=8-5)

**Mini bomba de agua:**

Justificación de compra: La mini bomba de agua es responsable de suministrar agua al sistema de riego. Su tamaño compacto y su eficiencia energética la hacen ideal para proyectos donde el espacio y el consumo de energía son importantes.

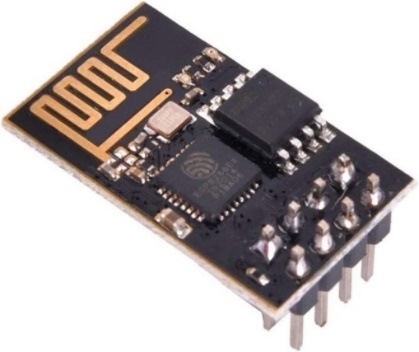
Características:

* + Bomba sumergible con cepillo y tubería para tanque.
  + Voltaje de operación: 3V - 6V.
  + Caudal máximo: 100L/h.
  + Altura máxima de elevación: 40-110cm.
  + Precio: 8,99 € (IVA incluido)
  + [Enlace](https://www.amazon.es/RUNCCI-YUN-Sumergible-Cepillo-Tuber%C3%ADa-paraTanque/dp/B082PM8L6X/ref=sr_1_5?crid=2MUPU3MCNTPQT&dib=eyJ2IjoiMSJ9.BvDnZOOEMe1jbq75fCUEGvu5WhCMollyOuhoSvwHUH2dMISFgI5-Tvh-N20KiBdzR6dcPa12N2VYjiuGCN0jt66Qc2e1J25ESUUNB4InJIQ-skXD7js1nCVtwmA25LruAR6k4HhvuFCGUoV2t_zh7XEpbr06zvCP6qfXkkhCaJuefjSL3wcWu6T2hjXErlVuPX97OvdpH8wM9M_AF3blwQsPwG44LApOT7bFTFkANZKut0WdXX3hcxv_YjzKC7NPsPK-Q5WYUK2xJkGcIhsRIeG0_Ewq2VWhKEP6pa94zOQ.fn2JSNYRdVeODKQ7FoVtuywYHlXZ8xu4gH3lW4BjWn0&dib_tag=se&keywords=bomba%2Bde%2Bagua%2Bmini%2Barduino&qid=1713425854&sprefix=mini%2Bbomba%2Bde%2B%2Barduino%2Caps%2C92&sr=8-5&th=1)

**Módulo ESP8266-01:**

Justificación de compra: El módulo ESP8266-01 es esencial para habilitar la conectividad WiFi en el sistema de riego. Permite la comunicación inalámbrica entre el sistema y otros dispositivos, como smartphones o computadoras, lo que facilita el monitoreo y control remoto del riego.

Características:

* + Adaptador para breadboard del módulo ESP8266-01.
  + Voltaje de operación: 3.3V.
  + Compatible con la mayoría de los microcontroladores.
  + Precio: 6,49 € (IVA incluido)
  + [Enlace](https://www.amazon.es/AZDelivery-ESP8266-01S-Adaptador-Breadboard-Compatible/dp/B072R6DPK7/ref=sr_1_3?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=1TAMWWNELPA1J&dib=eyJ2IjoiMSJ9.cc3weZljGH3eErmWCpTXT6inbUpVGXjTMvHCPQDanlVyv-lfEfQKtVHB-mWQ3027PwhbugX3tNSZ075GH8jqVoU_kHCSMI0-ubGB-67zyf-iE8JAV_sJnEWS53tfkLHbGqg8VJywmuCpArCUbKxUB7HRdH1OO6a_6zTFO4-xjCmpCH4LVB4kZO-J4egCkphwzgnmJRbW75CDBNCBvSx6Hr80x8cbCesCii2XyeH_AjSlHA8iKsVZDdDhoRb4yLzAfMpUUNlO_LzLoUuRIdzdl3SIaUh36PBz2dC7IsdONOg.i7E123vzvJIdndpLMyY3_Lm74n9o0NPVtW7b41dVWVg&dib_tag=se&keywords=modulo+esp8266-01&qid=1713428466&sprefix=modulo+esp8266-01%2Caps%2C108&sr=8-3)

**Soldador de estaño y Estaño**:

Justificación: Se utilizan para realizar conexiones eléctricas permanentes entre los componentes electrónicos del sistema de riego, lo que garantiza una conexión segura y duradera.

**Polímetro**:

Justificación: El polímetro se utiliza para medir voltajes, corrientes y resistencias en el circuito eléctrico del sistema de riego, lo que facilita la depuración de problemas y la verificación del funcionamiento correcto de los componentes.

**Funda termorretráctil**:

Justificación: Se utiliza para proteger y aislar las conexiones eléctricas realizadas con el soldador de estaño, lo que reduce el riesgo de cortocircuitos y mejora la durabilidad del sistema de riego.

## Software:

**Sistema Operativo Ubuntu 22.04:**

Justificación: Ubuntu es una opción ideal para el sistema operativo del servidor debido a su estabilidad, seguridad y amplio soporte comunitario. Además, su instalación sencilla y su compatibilidad con una amplia gama de software lo hacen perfecto para ejecutar el servidor web y otras aplicaciones relacionadas con el proyecto.

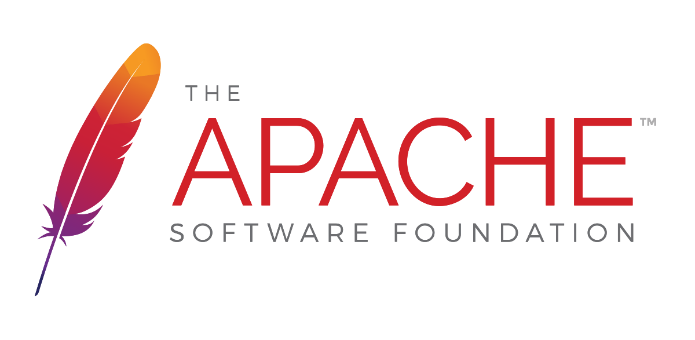
Características:

* + Distribución de Linux basada en Debian.
  + Versión: 22.04 (nombre en clave: TBD).
  + Características: Actualizaciones de seguridad regulares, soporte a largo plazo (LTS), entorno de escritorio GNOME, gran comunidad de usuarios y desarrolladores.
  + Precio: Gratuito.

**Apache:**

Justificación: Apache es un servidor web ampliamente utilizado y confiable que es compatible con PHP y otras tecnologías de servidor necesarias para el proyecto. Su robustez y su capacidad para manejar un alto volumen de solicitudes lo convierten en la elección perfecta para alojar la interfaz de usuario y la lógica del sistema de riego.

Características:

* + Servidor web HTTP de código abierto.
  + Versión: Apache 2.4.
  + Precio: Gratuito.

**MySQL:**

Justificación: MySQL es un sistema de gestión de bases de datos relacional ampliamente utilizado que es compatible con PHP y otras tecnologías de servidor. Su rendimiento, escalabilidad y facilidad de uso lo convierten en la elección ideal para almacenar datos relacionados con la configuración del sistema de riego, registros de riego y otros datos importantes.

Características:

* + Sistema de gestión de bases de datos relacionales.
  + Versión: MySQL 8.0.
  + Precio: Variable dependiendo del plan de AWS.

**PHP:**

Justificación: PHP es un lenguaje de programación popular y potente que es especialmente adecuado para el desarrollo web. Su integración con Apache y MySQL facilita la creación de páginas dinámicas y la interacción con la base de datos, lo que lo convierte en una herramienta esencial para la implementación de la interfaz de usuario del sistema de riego.

Características:

* + Lenguaje de programación del lado del servidor.
  + Versión: PHP 7.4.
  + Precio: Gratuito.

**MQTT:**

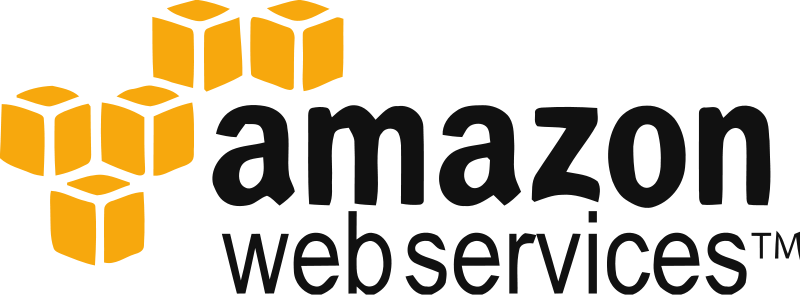
Justificación: MQTT es un protocolo ligero y eficiente diseñado para la comunicación entre dispositivos en redes IoT. Su capacidad para enviar mensajes de forma rápida y confiable lo convierte en la opción perfecta para la comunicación entre el sistema de riego y los dispositivos de monitoreo y control.

Características:

* + Protocolo de mensajería ligero para la transmisión de datos en tiempo real.
  + Precio: Gratuito.
  + Implementación: Se puede implementar utilizando bibliotecas como Mosquitto, HiveMQ, o implementaciones propias en lenguajes como Python, JavaScript, entre otros.

**AWS:**

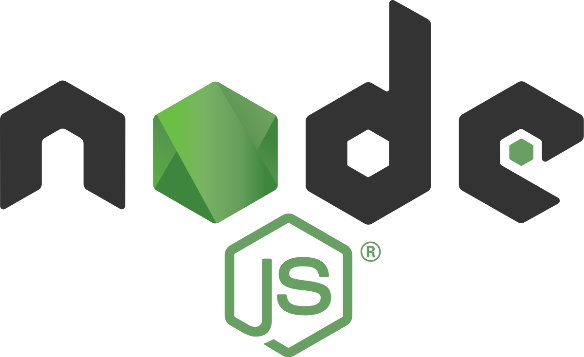
Justificación: AWS ofrece una infraestructura altamente escalable que permite aumentar o disminuir los recursos según las necesidades del proyecto de riego automatizado. Esto garantiza que el sistema pueda manejar picos de carga durante períodos de riego intensivo sin comprometer el rendimiento ni la disponibilidad.

Precio: Variable dependiendo de los servicios utilizados y su escala de uso.

**VM VirtualBox**:

Justificación: VM VirtualBox permite crear y gestionar máquinas virtuales, lo que facilita el desarrollo y la prueba del sistema de riego en diferentes entornos sin afectar al sistema operativo principal. Esto es crucial para garantizar la compatibilidad y estabilidad del sistema antes de su implementación.

**Node.js:**

Justificación: Node.js es un entorno de ejecución de JavaScript del lado del servidor que ofrece un rendimiento excepcional y una programación no bloqueante. Su capacidad para manejar múltiples conexiones simultáneas lo convierte en una opción ideal para aplicaciones web en tiempo real como la interfaz de usuario del sistema de riego.

# PRESUPUESTO DETALLADO

El siguiente desglose financiero proporciona una visión detallada del presupuesto destinado a los componentes necesarios para la realización exitosa del proyecto. Cada elemento esencial ha sido evaluado meticulosamente en términos de su costo individual, tanto sin incluir como incluyendo el Impuesto sobre el Valor Añadido (IVA), a una tasa estándar del 21%. Este análisis financiero permite una comprensión exhaustiva del gasto asociado con cada componente, proporcionando una guía clara para la asignación de recursos y la gestión financiera del proyecto.

Cada fila de la tabla presenta un componente específico junto con su respectivo precio unitario sin IVA, el monto del IVA aplicado y el precio total que incluye el impuesto. Esta estructura facilita la identificación de los costos individuales y permite una evaluación precisa del presupuesto total necesario para adquirir los materiales esenciales.

Al proporcionar esta transparencia financiera, se busca garantizar una gestión eficiente de los recursos económicos del proyecto, permitiendo así una ejecución fluida y eficaz de todas las etapas planificadas. Este desglose presupuestario sirve como una herramienta valiosa para la planificación financiera y la toma de decisiones estratégicas en el desarrollo del proyecto.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Componente | URL | Precio  (Sin IVA) | IVA (21%) | Precio (IVA) |
| Arduino UNO R3 | [Link](https://www.amazon.es/Arduino-UNO-A000066-microcontrolador-ATmega328/dp/B008GRTSV6/ref=sr_1_5?crid=3QBQVS22GUC7A&dib=eyJ2IjoiMSJ9.N8nF-4nrSGrUBX0eJZ8DQ3faJEQ6oU7jDfwMD55RC6PHY2xj3N9JePg85PFtW5v-nG1TLbyHvS9oNR5mPbB6sJNW75nQi8cvNy4kP0iT2t9zmy-F27ZnrA6bcqa8AZZNar1W1C-38gvaebG-VIpUeLuLxk2SnyFVRtfiXKLinuQd5iS-ovK5JxbNKNFDLE2FR01vFOMqDdFPUTTZQ6u4iDywcYGhDjz4-p_AGgxXiMcNd0uKmJsMRmqdGunroAgjvu3sF0CX8juMP2dBEHezR08bpnT6785feExFtGKqQeU.Xk6fvyzP2I-TnIgTqIMWK7zQALHMukrGGGX1qA2FSmo&dib_tag=se&keywords=arduino+uno+r3&qid=1713425949&sprefix=arduino+%2Caps%2C102&sr=8-5) | 18,16 € | 4,83 € | 22,99 € |
| Placa de pruebas | [Link](https://www.amazon.es/AZDelivery-Breadboard-barras-conductoras-Arduino/dp/B07KKJSFM1/ref=sr_1_5?crid=2F0BISXJE3DXY&dib=eyJ2IjoiMSJ9.qBPp7lHWWelJcqFJQYHaU0QuRzF7SqCy4PDW4Vus6H3i6FWCMCrJcUbWfocvBWiSrcpqZsns_aj-tGxVm46qMedXsErejA0wjcfdOf1kCLyJ2JrjAkkNHAGPXJnBufM0rvlByYZu-z--8kQSHVK2sqW7Pl0UG4Y7npkQe6TopjizWjn-UbA2-4b2jBGyobo__nowEZfJylMCyXCy5eWudUgpD3jMlgkeemq3v0SgJNp5GFHzYSsZk8ThjJNGFy9rRxNO2spD0ZbTRmkybW_gMfZychMC0zJ6p-fOY6aJTSI.JJS4vaJrBGw3FZS3Cp4WbMK3sYPlUuD8YwZCV7fqLIg&dib_tag=se&keywords=placa+de+pruebas+arduino&qid=1713428538&sprefix=placa+de+pruebas+ar%2Caps%2C103&sr=8-5) | 4,34 € | 1,15 € | 5,49 € |
| Módulo ESP8266-01 | [Link](https://www.amazon.es/AZDelivery-ESP8266-01S-Adaptador-Breadboard-Compatible/dp/B072R6DPK7/ref=sr_1_3?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=1TAMWWNELPA1J&dib=eyJ2IjoiMSJ9.cc3weZljGH3eErmWCpTXT6inbUpVGXjTMvHCPQDanlVyv-lfEfQKtVHB-mWQ3027PwhbugX3tNSZ075GH8jqVoU_kHCSMI0-ubGB-67zyf-iE8JAV_sJnEWS53tfkLHbGqg8VJywmuCpArCUbKxUB7HRdH1OO6a_6zTFO4-xjCmpCH4LVB4kZO-J4egCkphwzgnmJRbW75CDBNCBvSx6Hr80x8cbCesCii2XyeH_AjSlHA8iKsVZDdDhoRb4yLzAfMpUUNlO_LzLoUuRIdzdl3SIaUh36PBz2dC7IsdONOg.i7E123vzvJIdndpLMyY3_Lm74n9o0NPVtW7b41dVWVg&dib_tag=se&keywords=modulo+esp8266-01&qid=1713428466&sprefix=modulo+esp8266-01%2Caps%2C108&sr=8-3) | 5,13 € | 1,36 € | 6,49 € |
| Pack relé + bomba de agua + sensor de humedad del suelo | [Link](https://www.amazon.es/RUNCCI-YUN-Sumergible-Cepillo-Tuber%C3%ADa-paraTanque/dp/B0814HXWVV/ref=sr_1_5?crid=2MUPU3MCNTPQT&dib=eyJ2IjoiMSJ9.BvDnZOOEMe1jbq75fCUEGvu5WhCMollyOuhoSvwHUH2dMISFgI5-Tvh-N20KiBdzR6dcPa12N2VYjiuGCN0jt66Qc2e1J25ESUUNB4InJIQ-skXD7js1nCVtwmA25LruAR6k4HhvuFCGUoV2t_zh7XEpbr06zvCP6qfXkkhCaJuefjSL3wcWu6T2hjXErlVuPX97OvdpH8wM9M_AF3blwQsPwG44LApOT7bFTFkANZKut0WdXX3hcxv_YjzKC7NPsPK-Q5WYUK2xJkGcIhsRITQaHykRvN9oqPdYdsBt500.jVOSEI4JY9ReoKj4jAfbugO4NheDxsUnXwXYSlSqqEE&dib_tag=se&keywords=bomba%2Bde%2Bagua%2Bmini%2Barduino&qid=1713428234&sprefix=mini%2Bbomba%2Bde%2B%2Barduino%2Caps%2C92&sr=8-5&th=1) | 7,89 € | 2,10 € | 9,99 € |
| Sensor DHT11 | [Link](https://www.amazon.es/AZDelivery-Temperatura-Compatible-Arduino-Incluido/dp/B089W8DB5P/ref=sr_1_1_sspa?dib=eyJ2IjoiMSJ9.J8_7o_7nkjidzjJwsOQSVGJTk_DI7naVgLCPtovUVJ7ObflO0e2acy4CTzlM1qZjY0EkKvQ3zgaUXruE6kQBgGJ88zYS0NGs-7J0HgddAVIUDNs8NCl-rW76F0seUGP0QFqYYo8JFtAgv5lILF93f1ZU2sPjxq6MfthSiuoTU9ndeVl0DgVSUwDLCXc5q7lzvQo6QVhoXGwiRi3qs5-6Yd-BooWL4U1QcterRJG73n3VAkXL0sxmcViU5cHgBCCbTmR61kVS9GCb1kN1u747vpsdhJK65mYaTMc6M_XHMho.GTL45kmFU2Jy-XPbw0l0qWKS8N4lllHWz6roPgCiqD4&dib_tag=se&keywords=dht11&qid=1713425766&sr=8-1-spons&sp_csd=d2lkZ2V0TmFtZT1zcF9hdGY&th=1) | 4,73 € | 1,26 € | 5,99 € |
| Total (IVA incluido): | **50,95 €** | | | |

# PLANIFICACIÓN DEL PROYECTO

El diagrama de Gantt detalla el plan de trabajo para un proyecto distribuido en múltiples fases. Cada fase abarca una serie de tareas específicas que son cruciales para el progreso del proyecto. La primera fase, centrada en la planificación, abarca actividades como la investigación y selección de componentes, el diseño del circuito electrónico y la adquisición de materiales.

La siguiente fase se enfoca en la implementación del hardware, incluyendo actividades como el ensamblaje del circuito y la programación del Arduino Uno. Estos pasos son esenciales para construir la base física del proyecto y asegurar su funcionalidad.

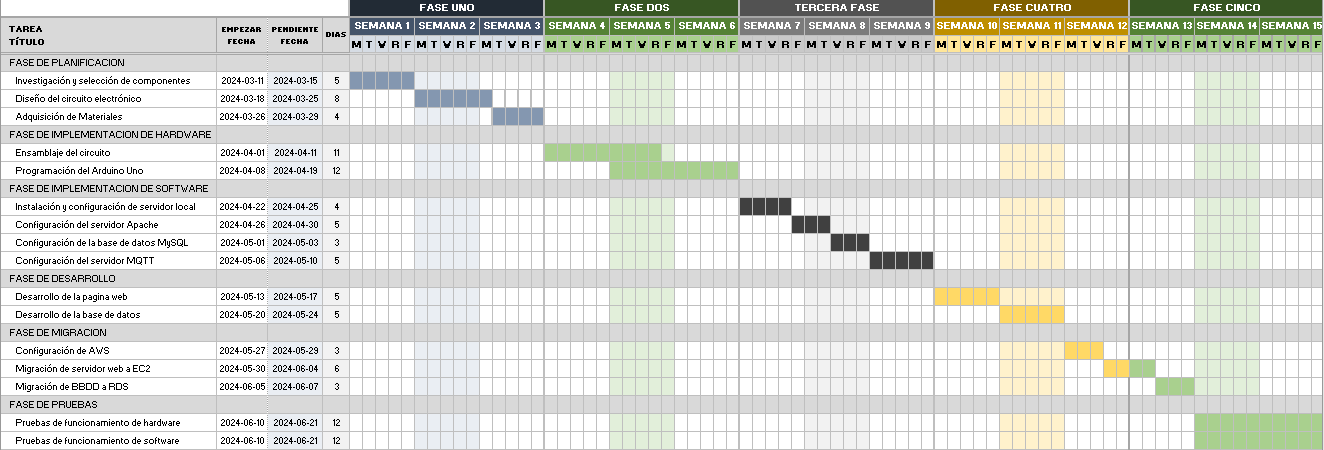
La implementación del software constituye otra fase crítica del proyecto, que involucra la instalación y configuración de componentes como el servidor local, Apache, la base de datos MySQL y el servidor MQTT. Estos aspectos son fundamentales para la interoperabilidad y el funcionamiento adecuado del sistema.

La etapa de desarrollo se centra en la creación de la parte visible y funcional del proyecto, como el desarrollo de la página web y la base de datos. Estas actividades son esenciales para la creación de una plataforma interactiva y eficiente.

La migración del proyecto a un entorno de producción constituye otra fase crucial, que incluye la configuración de AWS, la migración del servidor web a EC2 y la migración de la base de datos a RDS. Estos pasos son necesarios para garantizar la disponibilidad y la escalabilidad del proyecto en un entorno en vivo.

Finalmente, se llevan a cabo pruebas exhaustivas del hardware y el software para garantizar su correcto funcionamiento y rendimiento.

Puedes ver mejor la imagen [aquí](https://drive.google.com/file/d/1XAeJymxGkElY1-4thURDm82fCo2AUisi/view?usp=sharing)



# DESARROLLO TÉCNICO DEL PROYECTO

En este apartado se expondrán todos los pasos, procedimientos y enfoques técnicos que se emplearán en el desarrollo del proyecto. Se detallarán las tecnologías, lenguajes de programación y herramientas que se utilizarán. Se describirá el diseño detallado de las funcionalidades del sistema, utilizando imágenes para aclarar las interacciones entre los componentes. Además, se expondrá el enfoque y las prácticas de desarrollo de software que se emplearán, junto con los procesos y herramientas para la integración continua y el despliegue continuo del software. En resumen, este apartado proporcionará una guía técnica integral para la ejecución exitosa del proyecto.

## Configuración del Hardware

### Ensamblaje y Cableado de los Componentes

### Desarrollo del Código para Arduino

El código de Arduino de este proyecto gestiona un sistema de monitoreo y control de riego utilizando un sensor de temperatura y humedad DHT11, un sensor de humedad del suelo y un relé para controlar una bomba de agua. Además, se conecta a una red WiFi y envía datos a un servidor MQTT. A continuación, se presenta una explicación detallada de cada parte del código:

**Inclusión de Bibliotecas**

#include <DHT.h> // Biblioteca para el sensor DHT11

#include <WiFiEsp.h>

#include <WiFiEspClient.h>

#include <PubSubClient.h>

#include "SoftwareSerial.h"

Estas bibliotecas son necesarias para trabajar con el sensor DHT11, la conexión WiFi mediante un módulo ESP8266, y para comunicarse con el servidor MQTT.

**Definición de Pines y Variables**

#define DHTPIN 7 // Pin al que está conectado el sensor DHT11

#define DHTTYPE DHT11 // Tipo de sensor DHT que estás utilizando

#define SOIL\_MOISTURE\_PIN A0 // Pin al que está conectado el sensor de humedad del suelo

#define RELAY\_PIN 9 // Pin al que está conectado el relé

Se definen los pines a los que están conectados el sensor DHT11, el sensor de humedad del suelo y el relé.

**Inicialización de Variables y Objetos**

DHT dht(DHTPIN, DHTTYPE);

float temperature = 0;

float humidity = 0;

int soilMoisture = 0;

#define WIFI\_AP "TP-Link\_7F20"

#define WIFI\_PASSWORD "79949910"

char server[50] = "jarduino.ddns.net";

WiFiEspClient espClient;

PubSubClient client(espClient);

SoftwareSerial soft(3, 2);

unsigned long lastSend;

int status = WL\_IDLE\_STATUS;

Se inicializan objetos y variables para manejar el sensor DHT, la conexión WiFi, el cliente MQTT, y la comunicación serial.

**Configuración Inicial**

void setup() {

Serial.begin(9600);

pinMode(RELAY\_PIN, OUTPUT); // Configurar el pin del relé como salida

InitWiFi();

client.setServer(server, 1883);

client.setCallback(callback); // Establecer la función de devolución de llamada para los mensajes entrantes

lastSend = 0;

dht.begin();

pinMode(SOIL\_MOISTURE\_PIN, INPUT);

}

Se configuran los pines, se inicializa la conexión WiFi, se configura el servidor MQTT y se define la función de devolución de llamada para los mensajes entrantes. Además, se inicializa el sensor DHT.

**Bucle Principal**

void loop() {

status = WiFi.status();

if(status != WL\_CONNECTED) {

reconnectWifi();

}

if(!client.connected()) {

reconnectClient();

}

temperature = dht.readTemperature();

humidity = dht.readHumidity();

soilMoisture = analogRead(SOIL\_MOISTURE\_PIN); // Leer valor del sensor de humedad del suelo

if(millis() - lastSend > 2000 ) {

sendDataTopic();

lastSend = millis();

}

client.loop();

}

En el bucle principal, se verifica y reconecta la conexión WiFi y el cliente MQTT si es necesario. Luego, se leen los valores de temperatura, humedad y humedad del suelo. Cada 2 segundos, se envían estos datos al servidor MQTT.

**Función de Devolución de Llamada**

void callback(char\* topic, byte\* payload, unsigned int length) {

// Si se recibe un mensaje en el topic "riego", se activa el relé

if (strcmp(topic, "riego") == 0) {

int seconds = atoi((char\*)payload); // Convertir el payload a entero (segundos)

activatePump(seconds);

}

}

Esta función se ejecuta cuando se recibe un mensaje en el topic "riego". Si se recibe un mensaje, se activa la bomba de agua durante el número de segundos especificado en el mensaje.

**Envío de Datos al Servidor MQTT**

void sendDataTopic() {

String payload = "Temperatura: " + String(temperature) + "C, Humedad: " + String(humidity) + "%, Humedad del suelo: " + String(soilMoisture) + "id\_ard: 1";

char attributes[100];

payload.toCharArray(attributes, 100);

client.publish("sensores", attributes);

Serial.println(attributes);

}

Esta función crea un mensaje con los datos de los sensores y lo publica en el topic "sensores" del servidor MQTT.

**Activación de la Bomba de Agua**

void activatePump(int seconds) {

digitalWrite(RELAY\_PIN, HIGH); // Activar el relé (encender la bomba)

delay(seconds \* 1000); // Esperar el tiempo indicado en segundos

digitalWrite(RELAY\_PIN, LOW); // Desactivar el relé (apagar la bomba)

}

Esta función activa la bomba de agua (mediante el relé) durante el tiempo especificado en segundos.

**Inicialización de la Conexión WiFi**

void InitWiFi() {

soft.begin(9600);

WiFi.init(&soft);

if (WiFi.status() == WL\_NO\_SHIELD) {

Serial.println("El modulo WiFi no esta presente");

while (true);

}

reconnectWifi();

}

Esta función inicializa la conexión WiFi utilizando SoftwareSerial para comunicarse con el módulo ESP8266.

**Reconexión WiFi**

void reconnectWifi() {

Serial.println("Iniciar conección a la red WIFI");

while(status != WL\_CONNECTED) {

Serial.print("Intentando conectarse a WPA SSID: ");

Serial.println(WIFI\_AP);

status = WiFi.begin(WIFI\_AP, WIFI\_PASSWORD);

delay(500);

}

Serial.println("Conectado a la red WIFI");

}

Esta función intenta reconectar a la red WiFi si la conexión se pierde.

**Reconexión del Cliente MQTT**

void reconnectClient() {

while(!client.connected()) {

Serial.print("Conectando a: ");

Serial.println(server);

String clientId = "ESP8266Client-" + String(random(0xffff), HEX);

if(client.connect(clientId.c\_str())) {

client.subscribe("riego"); // Suscribirse al topic "riego"

Serial.println("[DONE]");

} else {

Serial.print( "[FAILED] [ rc = " );

Serial.print( client.state() );

Serial.println( " : retrying in 5 seconds]" );

delay( 5000 );

}

}

}

Esta función intenta reconectar el cliente MQTT si la conexión se pierde, y se suscribe al topic "riego" nuevamente.

**Resumen**

En resumen, este código de Arduino:

1. Lee datos de un sensor DHT11 (temperatura y humedad) y un sensor de humedad del suelo.
2. Conecta a una red WiFi utilizando un módulo ESP8266.
3. Envía datos de los sensores a un servidor MQTT cada 2 segundos.
4. Escucha mensajes en el topic "riego" y activa una bomba de agua durante el tiempo especificado en el mensaje recibido.

Puedes ver el código completo [aquí](https://drive.google.com/file/d/1yPVuzKBgMBUmTKLPZUvACDSCyaGG67rN/view?usp=sharing).

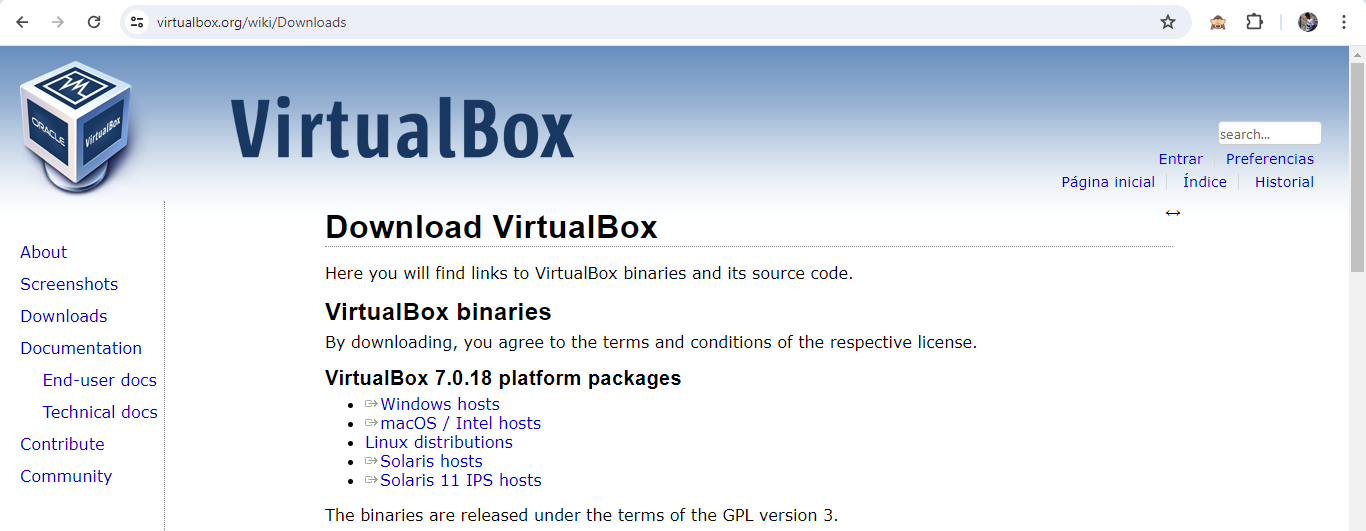
## Implementación en Entorno Local

### Preparación del Entorno de Desarrollo (Instalación de Ubuntu 22.04)

La instalación de Ubuntu se llevará a cabo en un entorno virtual utilizando VirtualBox, lo que permitirá realizar pruebas y experimentos sin comprometer un sistema operativo físico. Una vez completadas y verificadas las configuraciones en VirtualBox, el siguiente paso será trasladar todo el entorno a AWS para su despliegue en producción. Durante la instalación en VirtualBox, configuraremos los aspectos básicos del sistema operativo, como el particionamiento del disco, la configuración de usuarios y contraseñas, así como la instalación de paquetes de software necesarios.

**Paso 1: Configurar VirtualBox**

**Descarga e instala VirtualBox**: Ve al sitio web de VirtualBox (<https://www.virtualbox.org/>) y descarga la versión adecuada para tu sistema operativo. Sigue las instrucciones de instalación proporcionadas por el instalador.



**Descarga la imagen ISO de Ubuntu**: Ve al sitio web oficial de Ubuntu (<https://ubuntu.com/download>) y descarga la imagen ISO de Ubuntu 22.04 LTS.

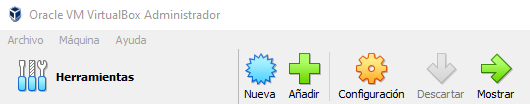


**Paso 2: Crear una nueva máquina virtual**

**Abre VirtualBox**: Ejecuta VirtualBox desde el menú de inicio o desde la ubicación donde lo hayas instalado.

**Crea una nueva máquina virtual**:

Haz clic en el botón "Nuevo" en la barra de herramientas.



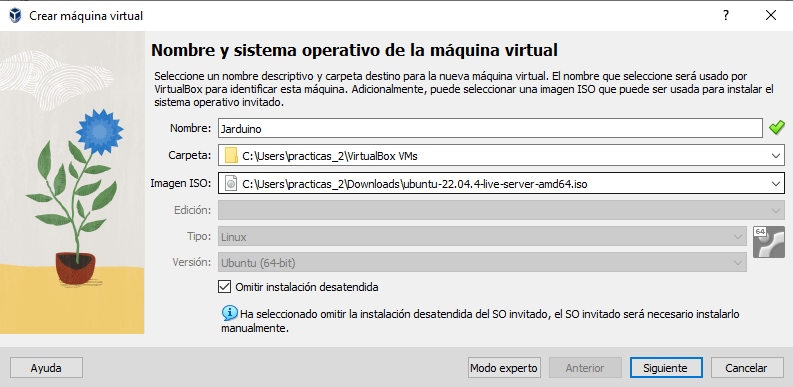
En el asistente de creación de máquinas virtuales, ingresa un nombre para la máquina virtual.

Selecciona el tipo de sistema operativo como "Linux".

Selecciona la versión como "Ubuntu (64-bit)" si has descargado la versión de 64 bits, o la versión de 32 bits según corresponda.

Puedes omitir estos pasos si seleccionamos la ISO ahora.

Haz clic en "Siguiente".



**Paso 3: Configurar la máquina virtual**

**Asigna memoria RAM**: En el siguiente paso del asistente, asigna la cantidad de memoria RAM que deseas para la máquina virtual. Se recomiendan al menos 2 GB para Ubuntu.

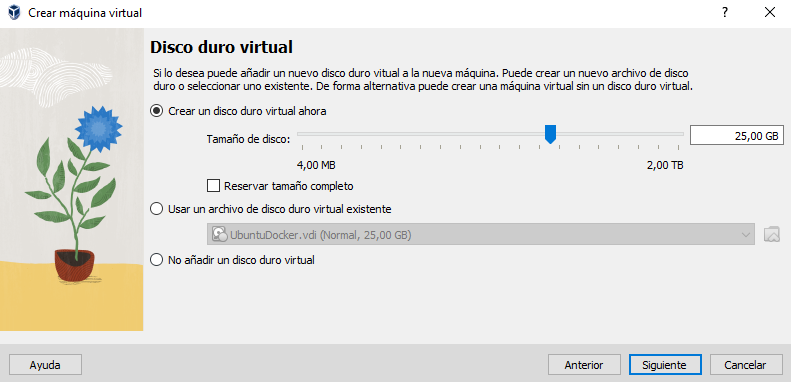


**Crea un disco duro virtual**:

Selecciona "Crear un disco duro virtual ahora".

Especifica el tamaño del disco duro virtual. Se recomiendan al menos 20 GB para una instalación básica de Ubuntu.

Haz clic en "Crear" para finalizar la configuración del disco duro.



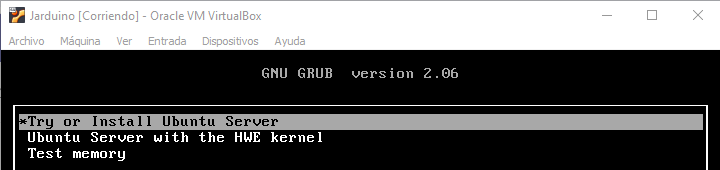
**Paso 4: Instalar Ubuntu en la máquina virtual**

**Inicia la máquina virtual**: Selecciona la máquina virtual y haz clic en "Iniciar" en la barra de herramientas.

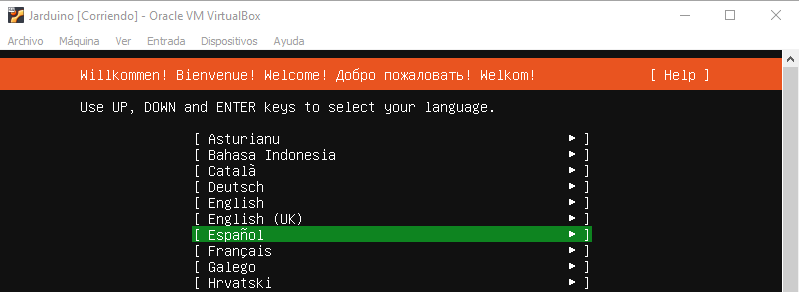


**Arranca desde la imagen ISO**: La máquina virtual debería iniciarse desde la imagen ISO de Ubuntu que configuraste. Si no lo hace, asegúrate de que la imagen esté seleccionada en la configuración de la máquina virtual.

**Selecciona "Instalar Ubuntu"** en el menú de arranque.

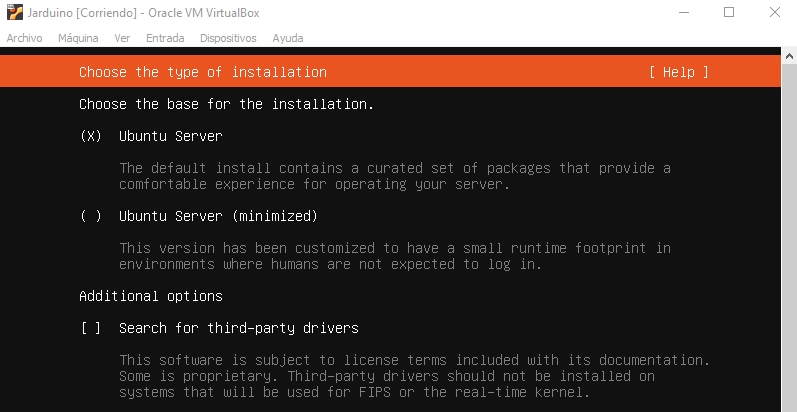


**Sigue el instalador de Ubuntu**: Sigue los pasos del instalador de Ubuntu para seleccionar el idioma, la ubicación, el tipo de teclado y otros ajustes.

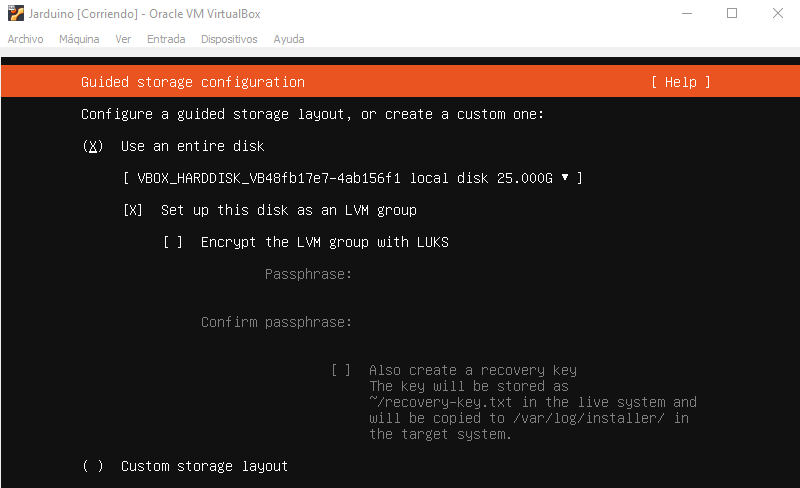


**Configura la instalación**:

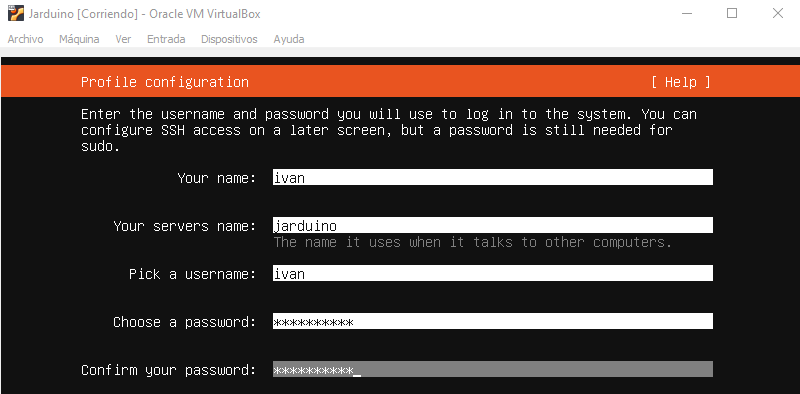
Selecciona la opción de instalación normal y elige si deseas instalar actualizaciones y software de terceros durante la instalación.



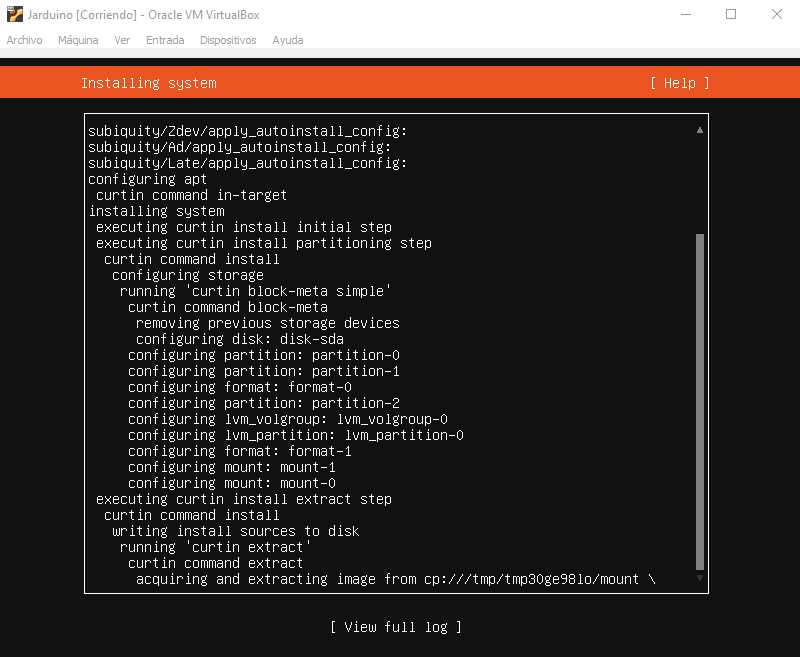
Elige el método de instalación del disco duro (puedes dejar la opción predeterminada para usar todo el disco si es para propósitos de prueba).



Crea una cuenta de usuario y establece una contraseña.



**Finaliza la instalación**: Una vez que el instalador haya completado la instalación, reinicia la máquina virtual y selecciona "Iniciar Ubuntu" desde el menú de arranque.



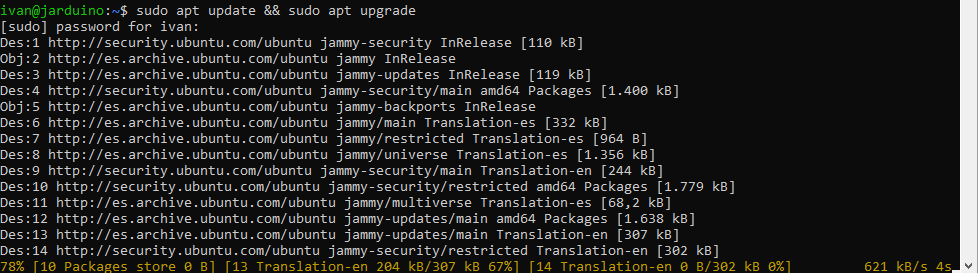
### Configuración del Servidor Web (Instalación de Apache, PHP y MySQL)

En el proceso de configuración de un servidor web en Ubuntu Server 22.04, se requiere la instalación de componentes clave que permitirán alojar y ejecutar aplicaciones web de manera efectiva. Para ello, se instalarán y configurarán Apache, un servidor web de código abierto ampliamente utilizado; PHP, un lenguaje de programación popular para el desarrollo web dinámico; y MySQL, un sistema de gestión de bases de datos relacional. Estos componentes proporcionarán una base sólida para el desarrollo y la ejecución de aplicaciones web en el entorno Ubuntu Server, permitiendo el almacenamiento, procesamiento y entrega de contenido web de manera eficiente y segura.

**Actualizar el sistema**

Antes de comenzar cualquier instalación de software, es crucial asegurarse de que el sistema esté actualizado para obtener las últimas correcciones de seguridad y actualizaciones. Puedes hacerlo ejecutando los siguientes comandos en la terminal:

sudo apt update && sudo apt upgrade

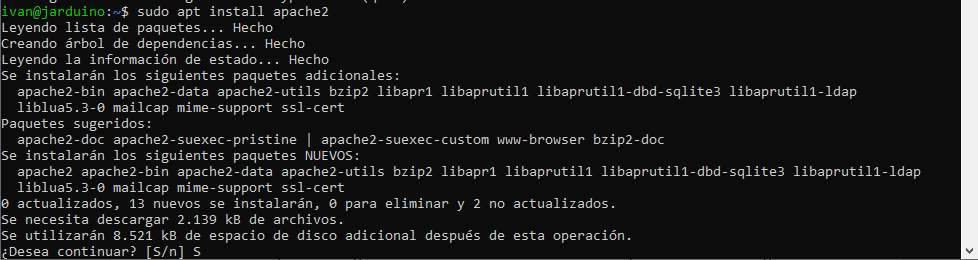


Esto actualizará los repositorios de software y aplicará las actualizaciones disponibles en tu sistema.

**Instalar Apache**

Apache es un servidor web de código abierto ampliamente utilizado en Internet. Para instalar Apache en Ubuntu, simplemente ejecuta el siguiente comando:

sudo apt install apache2



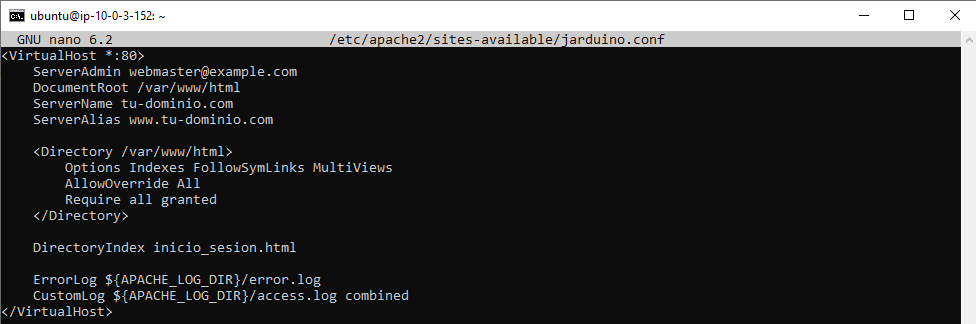
Después de ingresar tu contraseña de administrador, el sistema descargará e instalará Apache junto con todas sus dependencias. Una vez completada la instalación, Apache debería estar automáticamente configurado para iniciarse en el arranque del sistema.

**Configurar sitio web Apache**

Después de la instalación, puedes crear el archivo de configuración para tu sitio. Crea un archivo de configuración con este comando:

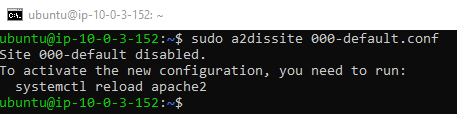
nano /etc/apache2/sites-available/tu-sitio.conf

Configura tu sitio según tus necesidades, tiene que quedar parecido a esto



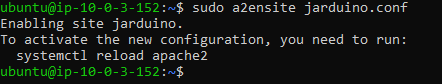
Ahora deshabilita el sitio por defecto de apache

sudo a2dissite 000-default.conf



Habilita el sitio que acabas de crear

sudo a2ensite tu-sitio.conf



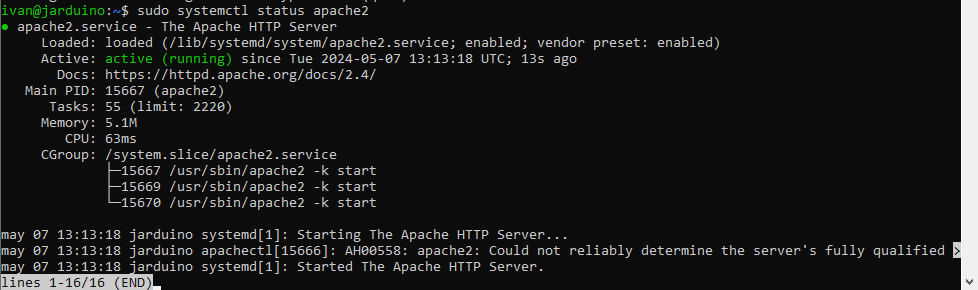
Y reinicia apache para aplicar los cambios



**Verificar el estado de Apache**

Después de la instalación y configuración, puedes verificar el estado de Apache para asegurarte de que esté en funcionamiento correctamente. Ejecuta el siguiente comando:

sudo systemctl status apache2

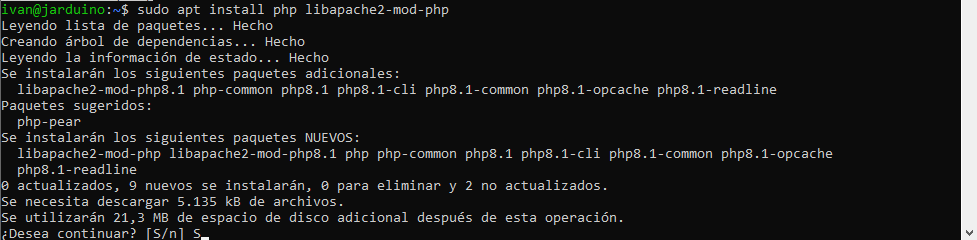


Si Apache se está ejecutando correctamente, verás un mensaje indicando que el servicio está activo y en funcionamiento.

**Instalar PHP**

PHP es un lenguaje de programación ampliamente utilizado para desarrollar aplicaciones web dinámicas. Puedes instalar PHP en Ubuntu junto con el módulo Apache necesario utilizando el siguiente comando:

sudo apt install php libapache2-mod-php

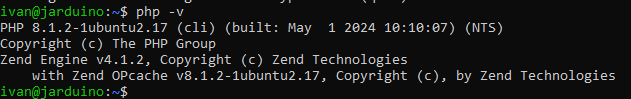


Una vez completada la instalación, PHP estará integrado con Apache y listo para su uso.

**Verificar la instalación de PHP**

Después de instalar PHP, es importante verificar que esté configurado correctamente. Puedes hacerlo ejecutando el siguiente comando:

php -v

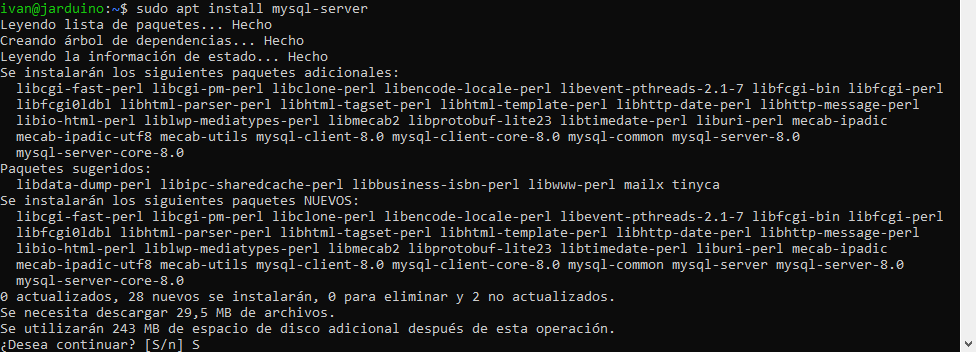


Esto mostrará la versión de PHP instalada en tu sistema y confirmará que PHP está funcionando correctamente.

**Instalar MySQL**

MySQL es un sistema de gestión de bases de datos relacional utilizado para almacenar y administrar datos. Puedes instalar MySQL en Ubuntu utilizando el siguiente comando:

sudo apt install mysql-server



Durante la instalación, se te pedirá que configures una contraseña para el usuario root de MySQL. Asegúrate de elegir una contraseña segura y memorízala o guárdala en un lugar seguro.

**Verificar la instalación de MySQL**

Después de instalar MySQL, puedes verificar su estado para asegurarte de que esté en funcionamiento correctamente. Ejecuta el siguiente comando:

sudo systemctl status mysql

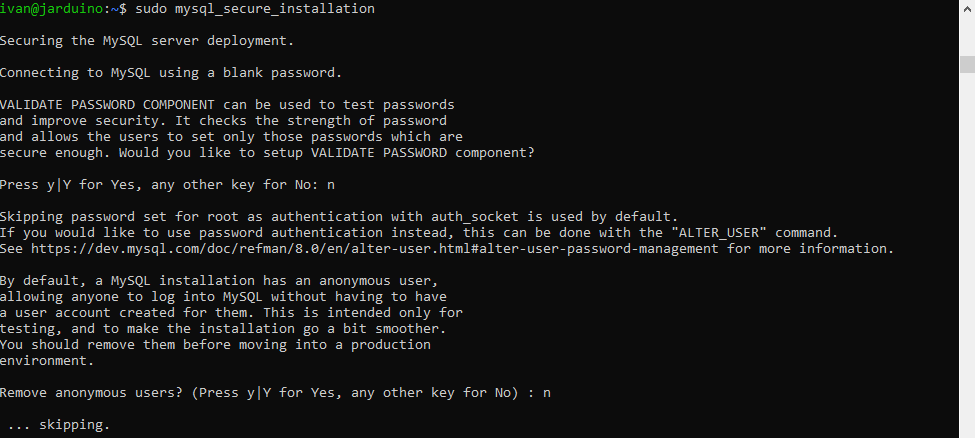


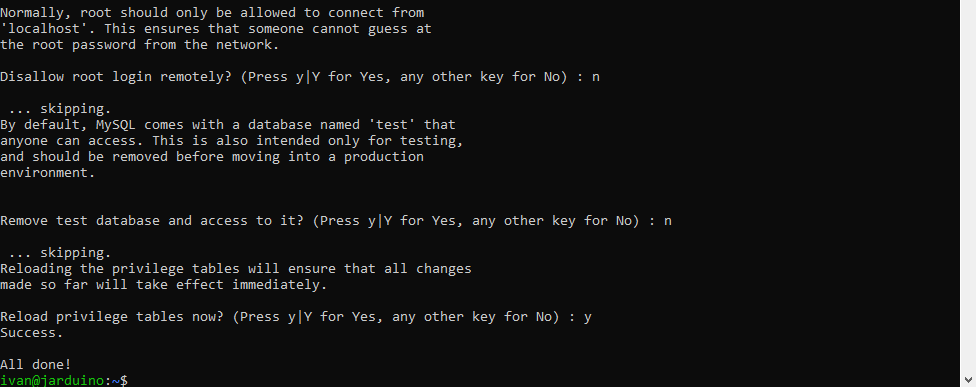
Si MySQL se está ejecutando correctamente, verás un mensaje indicando que el servicio está activo y en funcionamiento.

**Configurar MySQL**

Una vez instalado MySQL, es recomendable ejecutar el script de seguridad incluido para asegurar la instalación y configuración básica de MySQL. Puedes hacerlo ejecutando el siguiente comando:

sudo mysql\_secure\_installation





Este script te guiará a través de varios pasos para mejorar la seguridad de tu instalación de MySQL, como establecer una contraseña segura para el usuario root, eliminar usuarios anónimos, etc.

**Reiniciar Apache**

Después de instalar PHP y MySQL, es recomendable reiniciar el servidor Apache para que los cambios surtan efecto. Puedes hacerlo ejecutando el siguiente comando:

sudo systemctl restart apache2



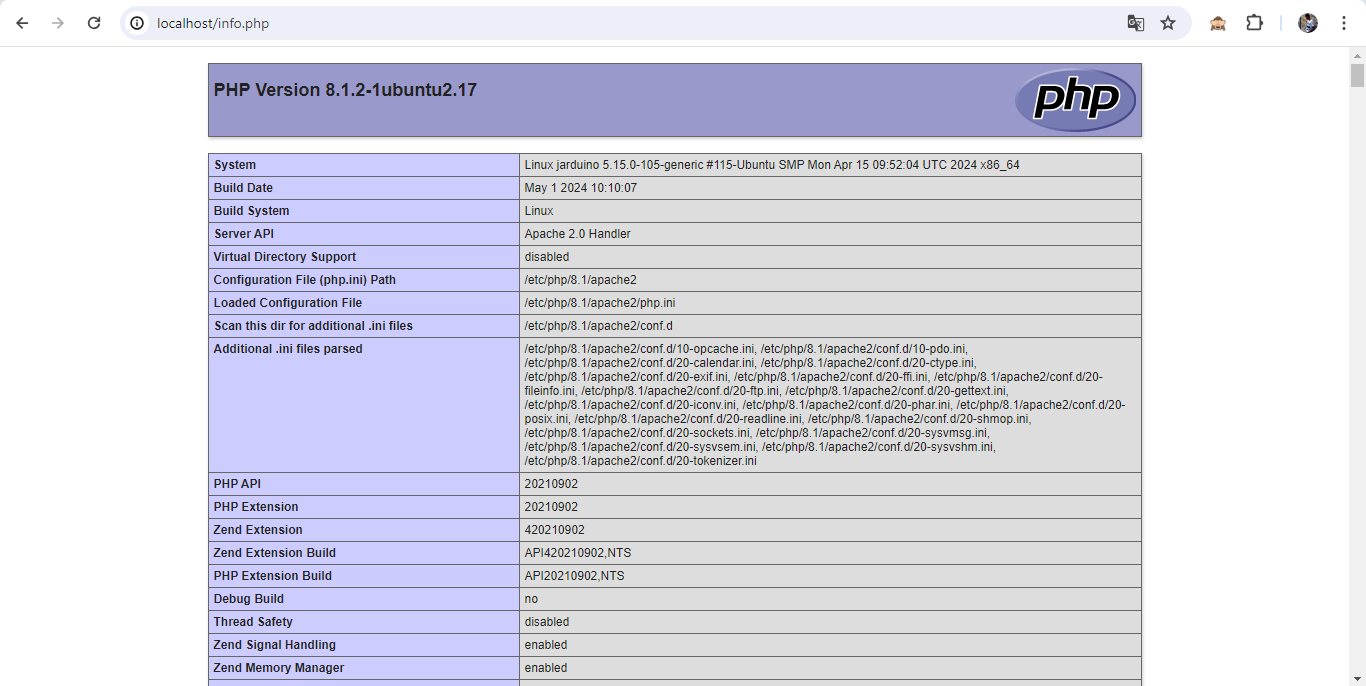
**Verificar la instalación completa**

Para asegurarte de que Apache, PHP y MySQL estén instalados y funcionando correctamente, puedes crear un archivo PHP de prueba y acceder a él desde tu navegador web.

echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php



Luego, abre tu navegador web y accede a **http://tu\_dirección\_ip/info.php**. Deberías ver una página con información detallada sobre la configuración de PHP.



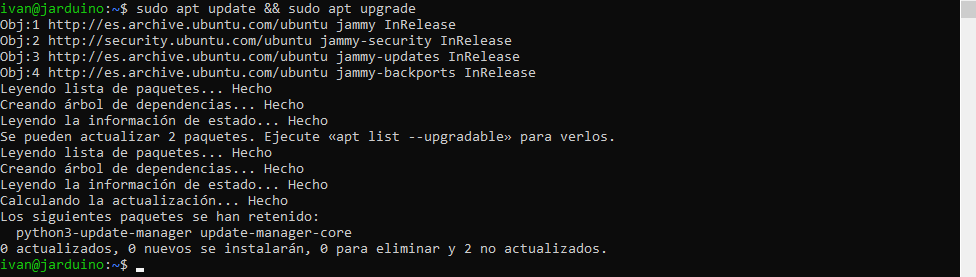
### Integración de MQTT con WebSocket (Instalación de Mosquitto)

Se instalará y configurará Mosquitto, un servidor MQTT de código abierto, junto con el soporte de WebSockets en un servidor Ubuntu 22.04. MQTT es un protocolo de mensajería ligero diseñado para dispositivos con recursos limitados y aplicaciones de Internet de las cosas (IoT). La integración de WebSockets permite a los clientes establecer conexiones bidireccionales en tiempo real con el servidor MQTT a través de conexiones web estándar. Este proceso es esencial para habilitar la comunicación entre dispositivos IoT y aplicaciones web de forma eficiente y segura.

**Agregar el repositorio de Mosquitto**

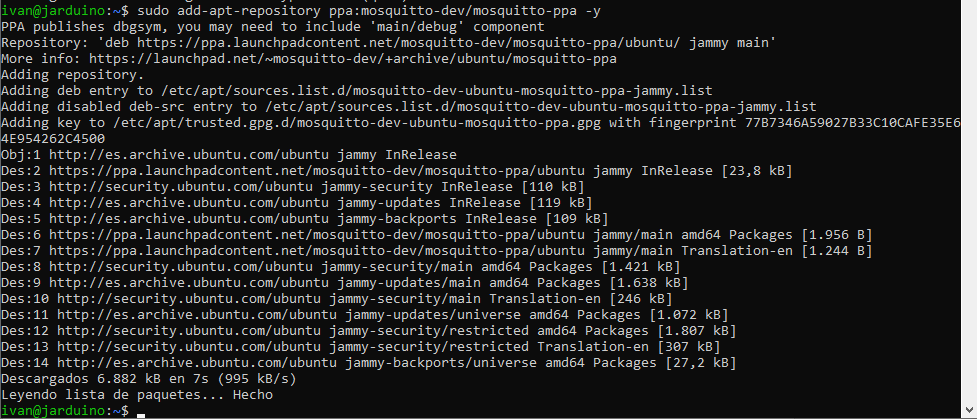
**Actualizar el sistema**: Se debe realizar una actualización del sistema antes de comenzar. Es importante asegurarse de que el sistema esté actualizado para garantizar que todas las dependencias estén al día y se puedan instalar correctamente. Se deben ejecutar los siguientes comandos:

sudo apt update && sudo apt upgrade



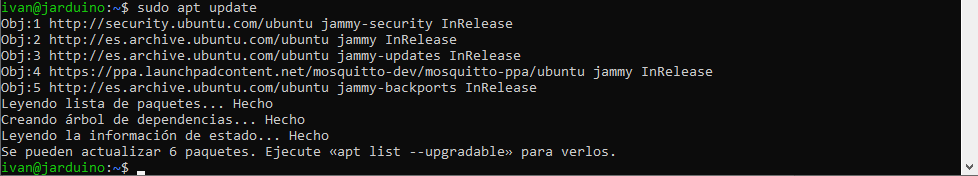
**Agregar el repositorio de Mosquitto**: Dado que Mosquitto no está incluido en los repositorios predeterminados de Ubuntu 22.04, se necesita agregar el repositorio oficial de Mosquitto. Esto se hace para obtener la última versión estable y compatible con la versión de Ubuntu utilizada. Primero, hay que importar la clave GPG del repositorio para verificar la autenticidad de los paquetes y luego hay que agregar el repositorio a nuestra lista de fuentes de software. Ejecuta los siguientes comandos:

sudo add-apt-repository ppa:mosquitto-dev/mosquitto-ppa -y



**Actualizar los repositorios**: Una vez agregado el repositorio de Mosquitto, es necesario actualizar la lista de paquetes para incluir los paquetes de Mosquitto. Esto se hace ejecutando el siguiente comando:

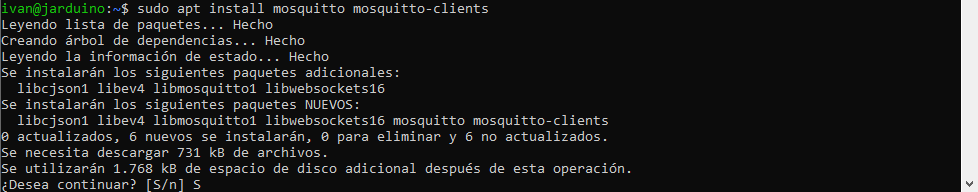
sudo apt update



**Instalar Mosquitto**

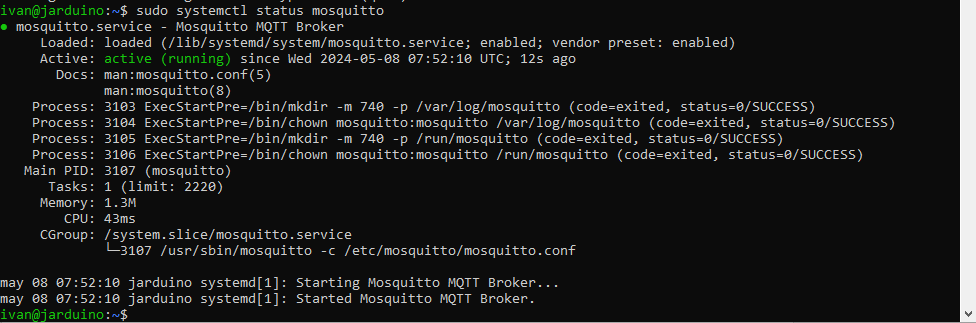
**Instalar Mosquitto**: Una vez agregado el repositorio de Mosquitto, ya se puede instalar Mosquitto y sus herramientas de cliente. Esto se logra ejecutando el siguiente comando:

sudo apt install mosquitto mosquitto-clients



**Verificar la instalación**: Una vez completada la instalación, se verifica si Mosquitto se está ejecutando correctamente ejecutando el siguiente comando:

sudo systemctl status mosquitto



**Configurar Mosquitto**

**Editar el archivo de configuración de Mosquitto**: La configuración predeterminada de Mosquitto debería ser suficiente para la mayoría de los casos de uso, pero si necesitas ajustar la configuración, puedes hacerlo editando el archivo de configuración ubicado en **/etc/mosquitto/mosquitto.conf**. Puedes usar cualquier editor de texto, como nano, para editar el archivo:

sudo nano /etc/mosquitto/mosquitto.conf

**Reiniciar Mosquitto**: Después de realizar cambios en la configuración, es importante reiniciar el servicio de Mosquitto para aplicar los cambios. Esto se puede hacer ejecutando el siguiente comando:

sudo systemctl restart mosquitto

**Instalar y configurar MQTT con WebSockets**

**Instalar el módulo WebSockets para Mosquitto**: Para habilitar el soporte de WebSockets en Mosquitto, es necesario instalar el módulo correspondiente. Esto se puede hacer instalando el paquete **libmosquitto-dev**:

sudo apt install libmosquitto-dev



**Editar el archivo de configuración de Mosquitto**: Una vez instalado el módulo, es necesario editar nuevamente el archivo de configuración de Mosquitto para habilitar explícitamente el protocolo WebSocket y especificar los puertos en los que escuchará las conexiones WebSocket. Abre el archivo de configuración con tu editor de texto preferido:

sudo nano /etc/mosquitto/mosquitto.conf

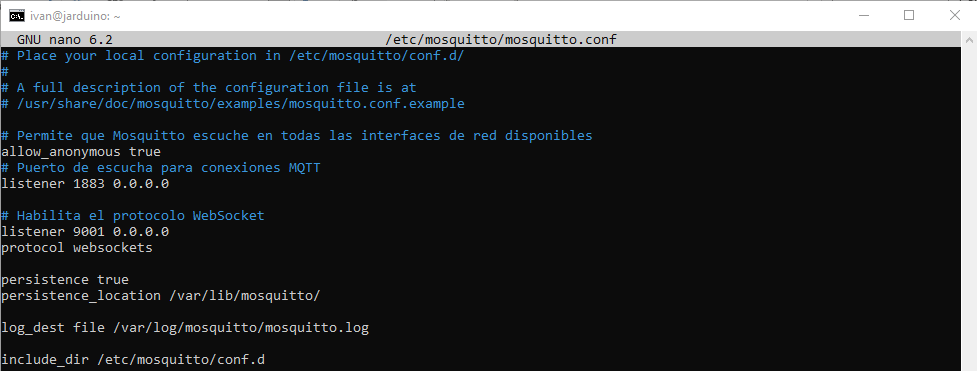
Luego, agrega las siguientes líneas al archivo:

allow\_anonymous true

listener 1883 0.0.0.0

listener 9001 0.0.0.0

protocol websockets



**Reiniciar Mosquitto**: Para que los cambios surtan efecto, es necesario reiniciar el servicio de Mosquitto:

sudo systemctl restart mosquitto

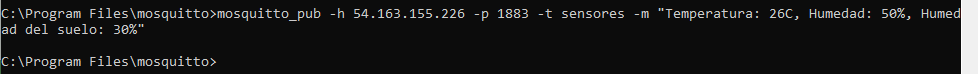


**Verificar la configuración**

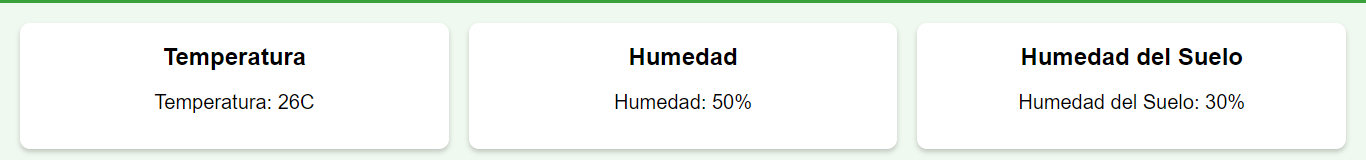
**Verificar los puertos abiertos**: Asegúrate de que los puertos 1883 y 9001 estén abiertos en el cortafuegos de tu servidor para permitir conexiones entrantes a Mosquitto.

**Probar la conexión**: Utiliza un cliente MQTT que admita WebSockets para conectarte a Mosquitto a través del puerto 9001. Crear una página web con JavaScript para enviar y recibir mensajes MQTT a través de WebSockets.

Envia unos datos simulando como los va a enviar el Arduino.



Y comprueba en la página web como los recibe y los muestra.



### Desarrollo de la Página Web y Base de Datos

GreenTech Solutions ha desarrollado un innovador Sistema de Riego Automatizado con Arduino que revoluciona la forma en que se cuidan las plantas al fusionar tecnología avanzada con el compromiso hacia la agricultura y la sostenibilidad. Este sistema se distingue por su capacidad para monitorear continuamente las condiciones del suelo y del entorno mediante la integración de múltiples sensores, cuyos datos se almacenan y analizan en una base de datos MySQL.

La base de datos cuenta con las siguientes tablas:

**users**: Esta tabla almacena información sobre los usuarios del sistema, incluyendo su nombre, apellidos, dirección de correo electrónico, contraseña, dirección y número de teléfono. Se utiliza principalmente para gestionar las cuentas de usuario y su información personal.

**arduino**: Aquí se registran los dispositivos Arduino utilizados en el sistema. Los campos incluyen un identificador único, la dirección IP del dispositivo y una referencia al usuario responsable. Esta tabla permite asociar cada dispositivo Arduino con su propietario.

**datos\_sensores**: Esta tabla almacena la información recopilada por los sensores del sistema, como la temperatura, humedad y humedad del suelo. También registra la fecha y hora de la lectura, junto con una referencia al dispositivo Arduino asociado. Permite realizar un seguimiento detallado de las condiciones del suelo y del entorno a lo largo del tiempo.

**registros\_riego**: Aquí se registran los eventos de riego en el sistema, incluyendo la fecha y hora del evento, la duración en segundos y las referencias al usuario y al dispositivo Arduino involucrados. Esta tabla permite mantener un registro de cuándo y cuánto se ha regado en el sistema.

Puedes ver el código de la base de datos [aquí](https://drive.google.com/file/d/1SrlXq2ZX2wg6i_bOLvqoxJ_J2yVLNOiU/view?usp=sharing).

Además de la base de datos, el sistema cuenta con una interfaz web alojada en un servidor Apache, que permite acceder a los datos de los sensores en tiempo real y controlar el sistema de riego de forma remota desde cualquier ubicación. La comunicación entre el Arduino y Apache se realiza mediante MQTT, asegurando una transmisión rápida y confiable de datos.

Aquí está la descripción de los archivos presentes en la página web del Sistema de Riego Automatizado de GreenTech Solutions:

**Carpeta bbdd**:

* + **Bbdd.php**: Este archivo contiene el script de conexión a la base de datos MySQL. Se utiliza para establecer una conexión con la base de datos desde el servidor web.
  + **Enviar\_sensores\_bbdd.php**: Es responsable de recibir los datos de los sensores enviados por el servidor MQTT desde el Arduino y almacenarlos en la tabla **datos\_sensores** de la base de datos. Esto garantiza que los datos recopilados por los sensores se registren de manera adecuada para su posterior análisis.
  + **phpMQTT.php**: Es una librería que proporciona funcionalidades para la comunicación MQTT en PHP. Se utiliza para facilitar la comunicación entre el servidor MQTT y el servidor web, permitiendo la transmisión de datos de los sensores en tiempo real.

**Carpeta web**:

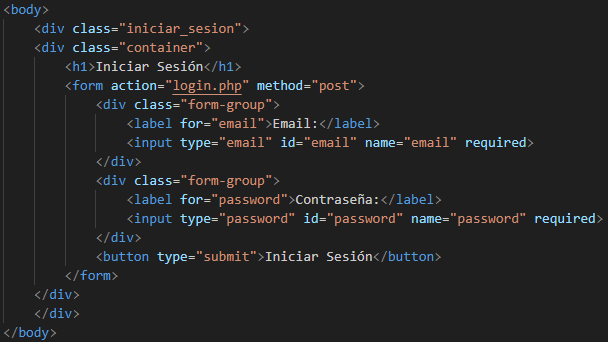
* + **principal.php**: Esta es la página principal de la aplicación web una vez que el usuario ha iniciado sesión. Aquí se muestran los datos de los sensores en tiempo real, lo que permite al usuario monitorear las condiciones del suelo y del entorno de manera continua.
  + **nosotros.php**: Esta es una pagina en la que se muestra información sobre la empresa GreenTech Solutions, así como a que se dedica y además se añade un iframe donde se muestra Google maps con la dirección de la oficina.
  + **contacto.php**: Es una página que proporciona a los usuarios la posibilidad de ponerse en contacto con el soporte técnico de GreenTech Solutions en caso de preguntas o problemas. Aquí los usuarios pueden enviar mensajes de consulta o informar sobre posibles fallas en el sistema.
  + **perfil.php**: Esta página muestra los datos personales del usuario y permite realizar cambios en caso de que haya errores. Aquí los usuarios pueden actualizar su información personal según sea necesario.
  + **actualizar\_perfil.php**: Es un script PHP que se encarga de procesar los datos recibidos del formulario de la página **perfil.php** y realizar la actualización de los datos del perfil en la base de datos. Esto asegura que los cambios realizados por el usuario se reflejen correctamente en el sistema.
  + **enviar.php**: Este archivo contiene el código PHP necesario para procesar los mensajes enviados por los usuarios a través del formulario de la página **contacto.php**. Se encarga de enviar el mensaje al soporte técnico de GreenTech Solutions por correo electrónico para su atención.
  + **registro.php**: Esta página permite a los usuarios ver el registro de la tabla **datos\_sensores**, que contiene datos registrados por los sensores del sistema. Proporciona funcionalidades para filtrar los registros por **fecha** y por **id\_arduino**.
  + **admin.php**: Esta página está diseñada para que los usuarios con privilegios de administrador puedan gestionar datos de las tablas **users** y **arduino**. Al acceder a esta página, el administrador puede ver una tabla con los registros existentes de ambas tablas, así como opciones para eliminar, insertar y actualizar registros.
  + **grafico.php**: s una página web dinámica que recopila datos de sensores almacenados en una base de datos, procesa estos datos y genera gráficos interactivos utilizando bibliotecas como Chart.js. Permite a los usuarios filtrar los datos según ciertos criterios, como el ID del Arduino o el mes, y luego visualizar los datos en forma de gráficos de temperatura, humedad y humedad del suelo.

**inicio\_sesion.php**: Es la página principal a la que se accede al buscar el dominio del sistema. Aquí los usuarios deben iniciar sesión para poder interactuar con la aplicación web y acceder a sus funcionalidades.

**login.php**: Este archivo contiene el código PHP utilizado para verificar las credenciales proporcionadas por el usuario en la página **inicio\_sesion.html**. Si las credenciales son correctas, se inicia una sesión y se redirige al usuario a la página **principal.php**, donde puede acceder a las funciones del sistema con la sesión iniciada.

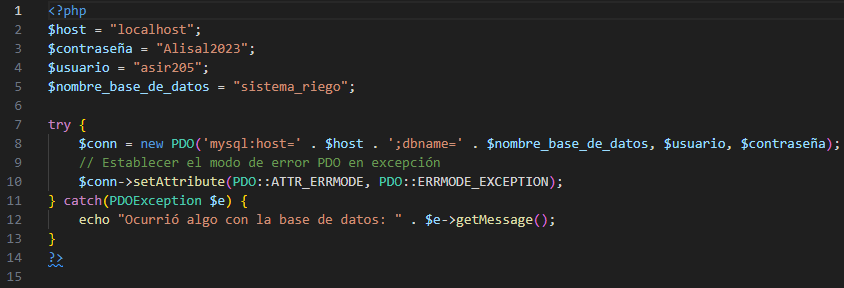
**inicio\_sesion.php**

Esta página muestra un formulario de inicio de sesión donde los usuarios pueden ingresar su correo electrónico y contraseña. Al enviar el formulario, los datos se envían al archivo **login.php** para su verificación.



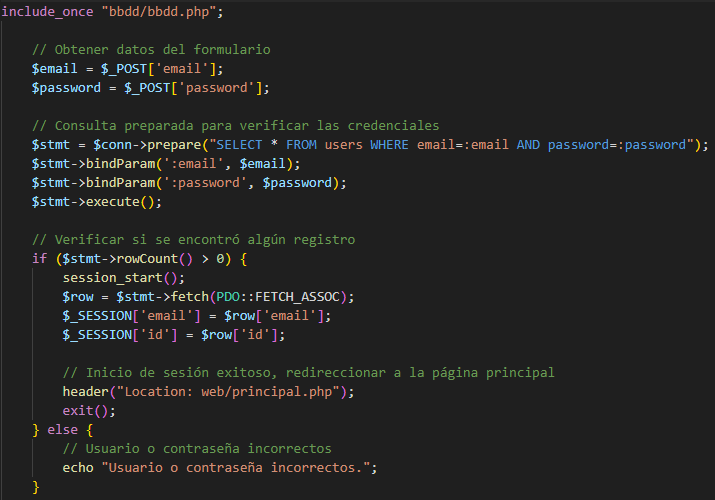
**bbdd.php**

Este archivo establece la conexión con la base de datos utilizando PDO (PHP Data Objects). Configura los parámetros de conexión, como el host, el nombre de usuario, la contraseña y el nombre de la base de datos. Luego, crea una instancia de PDO y establece el modo de error para que lance excepciones en caso de error.



**login.php**

En este archivo, se reciben los datos del formulario de inicio de sesión. Se verifica la existencia del usuario en la base de datos y la corrección de las credenciales ingresadas. Si las credenciales son válidas, se inicia una sesión y se redirige al usuario a la página principal (**principal.php**), de lo contrario, se muestra un mensaje de error.



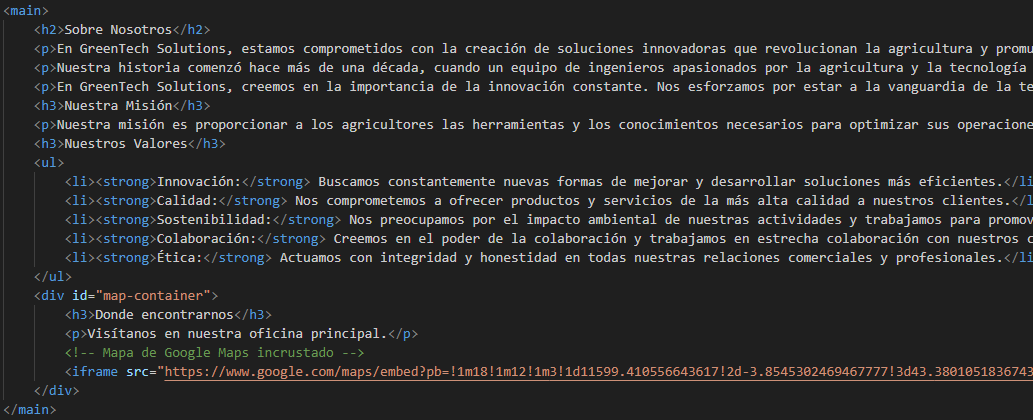
**principal.php**

Después del inicio de sesión exitoso, esta página muestra la página principal del sistema. Muestra un saludo personalizado al usuario y el contenido principal de la página.



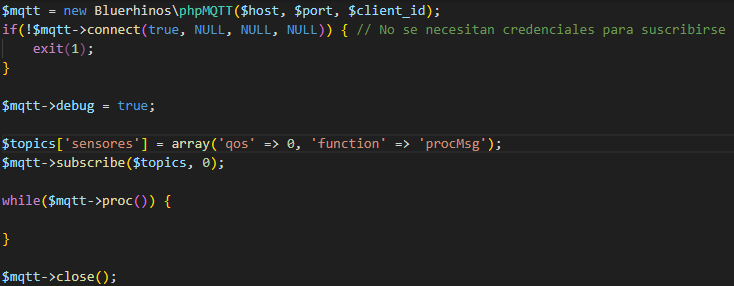
**nosotros.php**

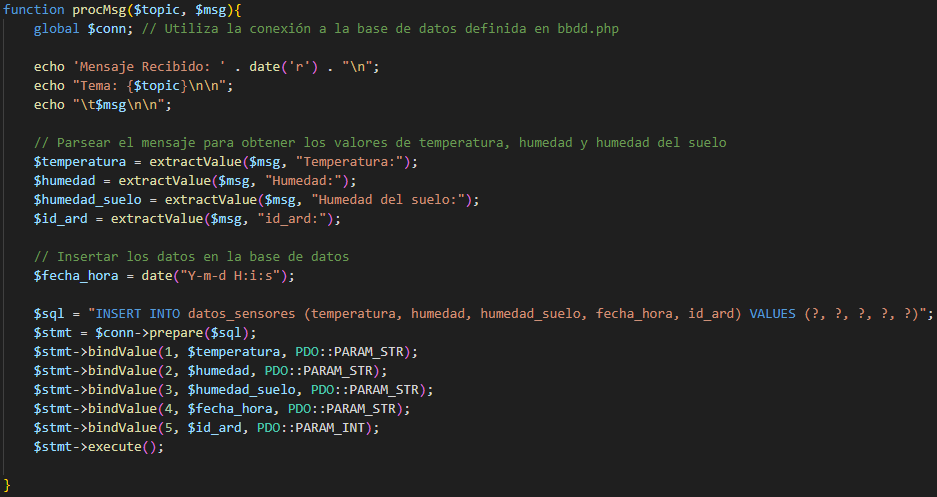
El archivo "nosotros.php" presenta una página que destaca la historia, la misión y los valores de GreenTech Solutions, una empresa dedicada a la innovación en tecnología agrícola y sostenibilidad ambiental.



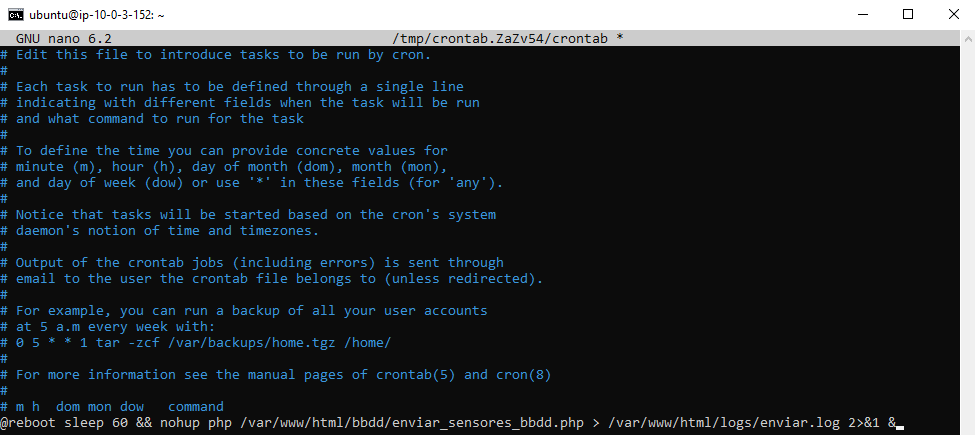
**enviar\_sensores\_bbdd.php**

Este archivo se encarga de recibir datos de sensores a través de MQTT (Message Queuing Telemetry Transport) y almacenarlos en la base de datos. Se suscribe a un tema MQTT específico donde se publican los datos de los sensores. Luego, procesa los mensajes recibidos y extrae los valores relevantes, como temperatura, humedad, etc., para almacenarlos en la base de datos.



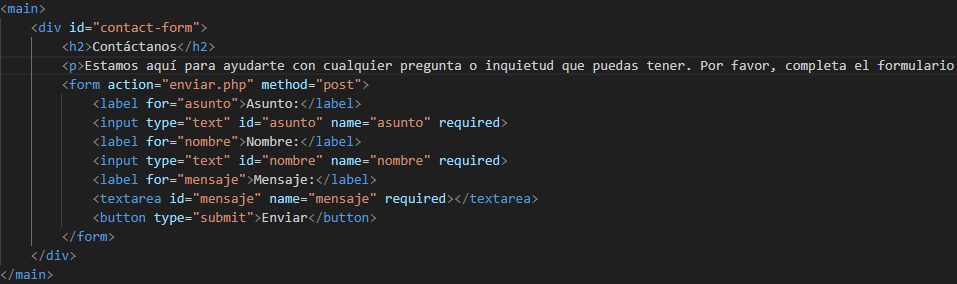


Para que este archivo PHP se ejecute automáticamente después de que el sistema se reinicie y pueda realizar sus tareas de manera confiable, se recomienda añadir el siguiente comando al cron. Este comando está diseñado para asegurar que el script se active de manera eficiente y que continúe ejecutándose incluso si la sesión del usuario se cierra. Además, facilita el registro de la salida del script para su posterior revisión y depuración.



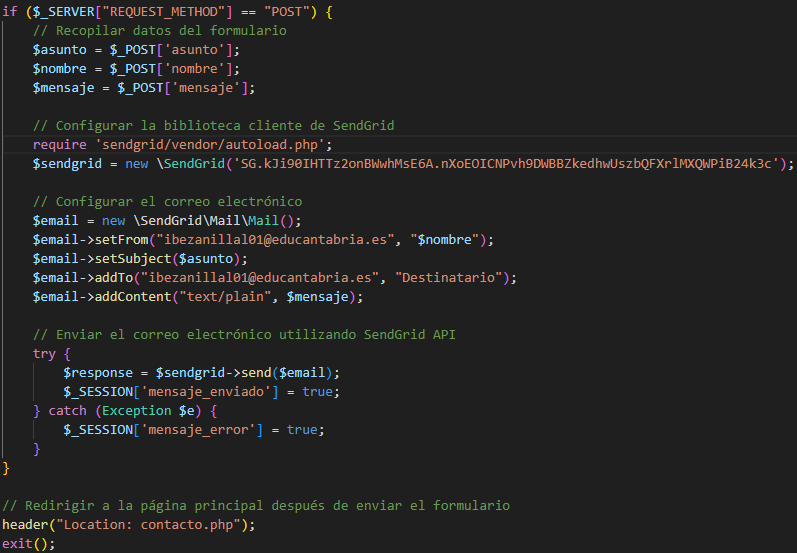
**contacto.php**

Esta página muestra un formulario de contacto donde los usuarios pueden ingresar su nombre, dirección de correo electrónico, asunto y mensaje. Al enviar el formulario, los datos se envían al archivo **enviar.php** para su procesamiento y envío por correo electrónico.



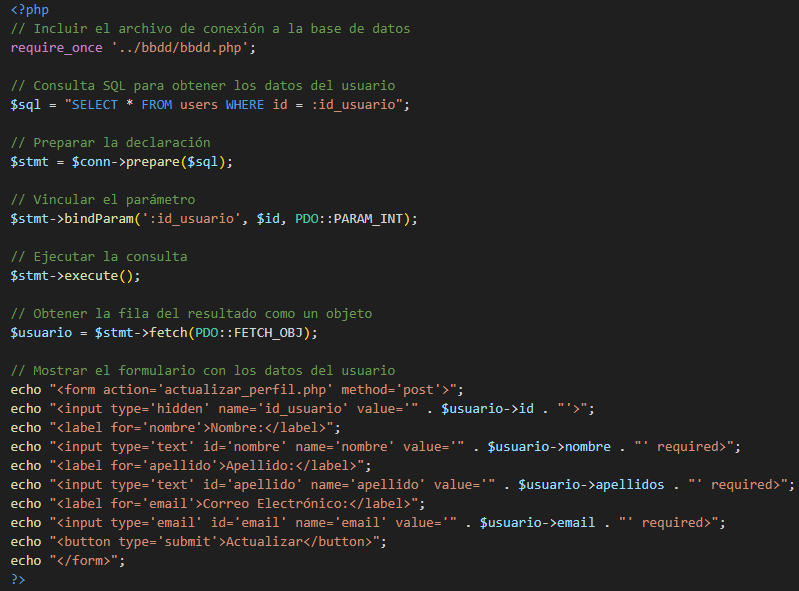
**enviar.php**

Es el archivo encargado de procesar y enviar los correos electrónicos del formulario de contacto. Al recibir los datos del formulario a través del método POST, recopila el asunto, nombre y mensaje. Luego, utiliza la biblioteca cliente de SendGrid para configurar y enviar el correo electrónico al destinatario deseado. Si ocurre algún error durante el proceso de envío, se maneja adecuadamente mediante un bloque try-catch. Finalmente, redirige al usuario de vuelta a la página de contacto, mostrando un mensaje de éxito o error según corresponda.



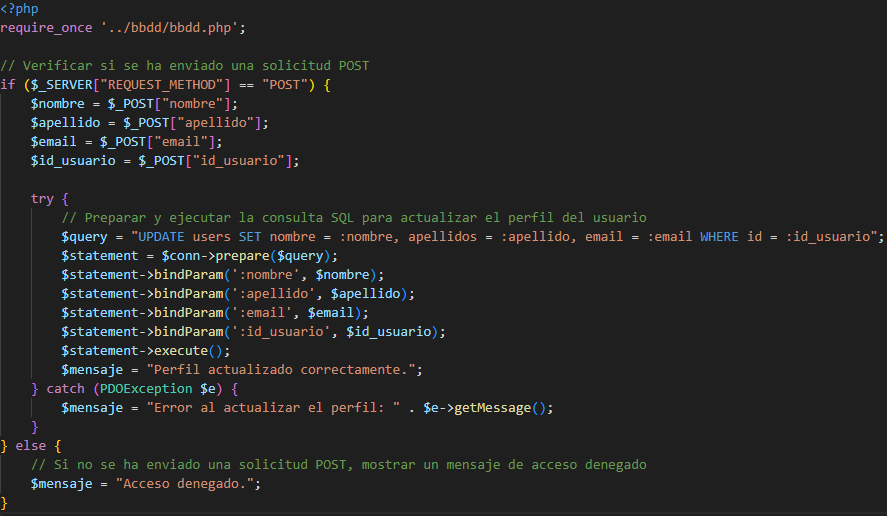
**perfil.php**

Esta página muestra el perfil del usuario actual, incluidos su nombre, apellido y correo electrónico. También permite al usuario actualizar su información mostrando un formulario con los datos actuales prellenados. Los cambios se envían al archivo **actualizar\_perfil.php** para su procesamiento.



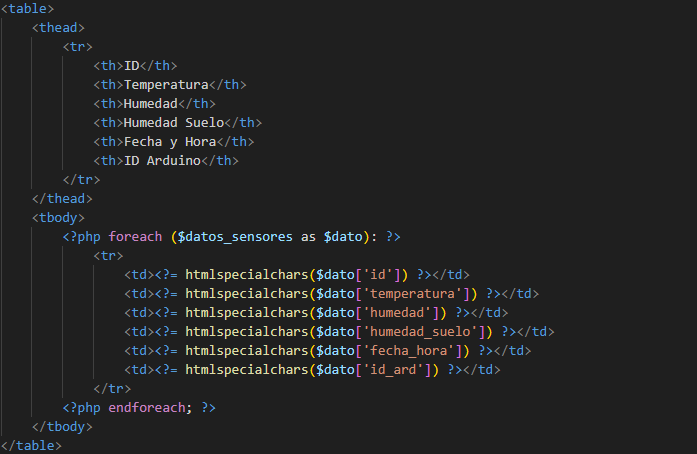
**actualizar\_perfil.php**

Aquí se procesan los datos enviados desde el formulario de actualización de perfil. Se ejecuta una consulta SQL para actualizar la información del usuario en la base de datos con los nuevos valores proporcionados. Se muestra un mensaje de éxito o error según el resultado de la operación.



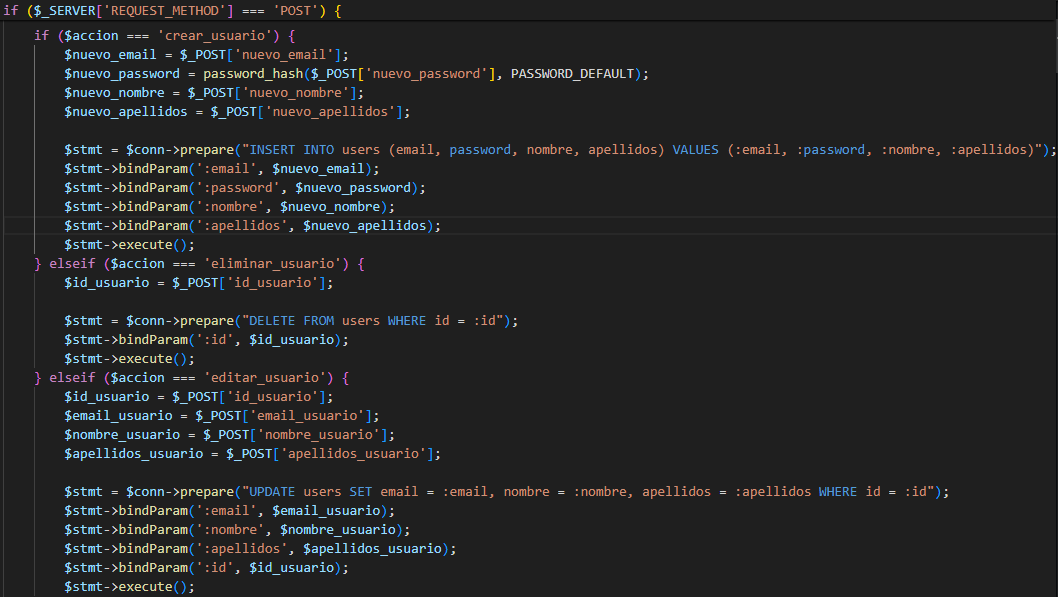
**registro.php**

Es una página que muestra el registro de datos de los sensores almacenados en la tabla **datos\_sensores**. Permite a los usuarios visualizar y analizar el historial de lecturas de los sensores. Además de mostrar los registros en una tabla, ofrece la capacidad de filtrar los datos por fecha y por el identificador único del Arduino (**id\_arduino**). Esto facilita el análisis específico de datos para un período de tiempo determinado o para un dispositivo Arduino particular.



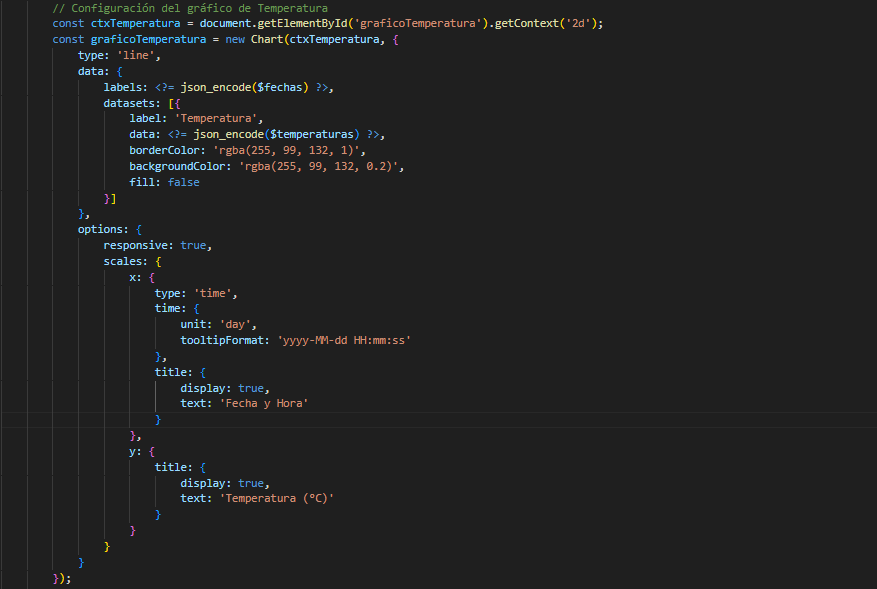
**admin.php**

Es una página dedicada a la gestión de datos por parte del usuario administrador. Proporciona funcionalidades para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en las tablas **users** y **arduino**. El usuario administrador puede ver, insertar, actualizar y eliminar registros de ambas tablas directamente desde esta interfaz. Además, muestra los datos de ambas tablas en forma de tablas HTML, lo que facilita la visualización y administración de los datos.



**grafico.php**

Es una página web que utiliza la biblioteca Chart.js para crear gráficos dinámicos. Los datos se recuperan de una base de datos MySQL y se filtran según el ID de Arduino o el mes seleccionado. Estos datos se utilizan para generar gráficos de línea que muestran la temperatura, la humedad y la humedad del suelo a lo largo del tiempo. Cada gráfico está configurado para ser interactivo y mostrar información detallada al pasar el cursor sobre los puntos. La página también utiliza HTML, CSS y PHP para proporcionar una interfaz de usuario amigable y funcionalidades de filtrado.



Puedes ver y descargar la carpeta con todos los archivos desde este [enlace](https://drive.google.com/drive/folders/1heY9S0aQkAJCrEqLw7TE5qJ3S-Yzp_8o?usp=sharing).

1. **principal.php**: Esta página suele ser la página de inicio del sitio web. En ella se presenta contenido relevante para los visitantes, como información sobre la empresa, sus productos o servicios destacados, posiblemente testimonios de clientes, enlaces a páginas importantes del sitio y cualquier otra información relevante que se desee destacar.
2. **nosotros.php**: En esta página se proporciona una visión general de la empresa. Puede incluir detalles sobre su historia, misión, visión, valores fundamentales, información sobre el equipo detrás de la empresa y cualquier otro aspecto que se considere importante para que los visitantes conozcan quiénes son y qué representan.
3. **contacto.php**: Aquí los visitantes pueden encontrar información de contacto de la empresa, como dirección física, números de teléfono, dirección de correo electrónico y posiblemente un formulario de contacto. La página puede incluir también un mapa interactivo con la ubicación de la empresa y horarios de atención.
4. **sensores.php**: Esta página ofrece detalles técnicos sobre los sensores utilizados por la empresa. Puede incluir especificaciones técnicas, información sobre su uso, ventajas, ejemplos de aplicación y cualquier otra información relevante para los clientes interesados en adquirir o utilizar estos sensores.
5. **grafico.php**: Aquí se generan gráficos a partir de los datos recopilados por los sensores. Los gráficos muestran la evolución de variables como temperatura, humedad y humedad del suelo en función del tiempo. Los usuarios pueden filtrar los datos por ID de Arduino y mes para obtener información más específica.
6. **inicio\_sesion.php**: Es la página donde los usuarios pueden iniciar sesión en el sitio web. Presenta un formulario donde los usuarios ingresan su correo electrónico y contraseña. También puede incluir enlaces para restablecer la contraseña o registrarse como nuevo usuario si aún no tienen una cuenta.
7. **login.php**: Este script procesa los datos del formulario de inicio de sesión. Verifica las credenciales ingresadas por el usuario en la base de datos y, si son válidas, inicia una sesión de usuario, lo que permite al usuario acceder a contenido restringido a usuarios registrados.
8. **cerrar-sesion.php**: Este script cierra la sesión de usuario, eliminando todas las variables de sesión y redirigiendo al usuario de vuelta a la página principal del sitio web.
9. **enviar\_sensores\_bbdd.php**: Recibe datos de los sensores y los almacena en la base de datos para su posterior análisis y visualización. Es una parte importante del sistema de monitoreo y gestión de datos de la empresa.
10. **perfil.php**: Esta página muestra información sobre el perfil del usuario logueado. Puede incluir detalles como nombre de usuario, dirección de correo electrónico, historial de compras o cualquier otra información personal relevante.
11. **actualizar\_perfil.php**: Permite a los usuarios actualizar la información de su perfil, como la contraseña o la dirección de correo electrónico. Procesa los cambios realizados y actualiza la información en la base de datos.
12. **enviar.php**: Este archivo maneja el envío de datos desde un formulario, como un formulario de contacto. Puede validar los datos ingresados por el usuario, enviar correos electrónicos de notificación y almacenar la información en la base de datos para su posterior revisión o seguimiento.

## Migración a la Nube (AWS)

### Creación de una Infraestructura Virtual Privada (VPC)

Una Virtual Private Cloud (VPC) es un servicio de AWS que te permite crear una red virtual aislada en la nube. Funciona de manera similar a una red tradicional en un centro de datos físico, pero con la flexibilidad y escalabilidad que ofrece la nube. Con una VPC, puedes definir tus propias subredes, configurar las reglas de enrutamiento y controlar la seguridad de la red.

Se trata del primer paso para la creación de cualquier elemento en línea de AWS. A través de la opción VPC y más se desplegará todo lo necesario para establecer una VPC con las especificaciones deseadas.

**Pasos Detallados para Crear una VPC en AWS:**

**Inicia sesión en la Consola de AWS:**

Accede a la consola de AWS con tus credenciales de inicio de sesión en <https://aws.amazon.com/>.

**Navega a la Consola de VPC:**

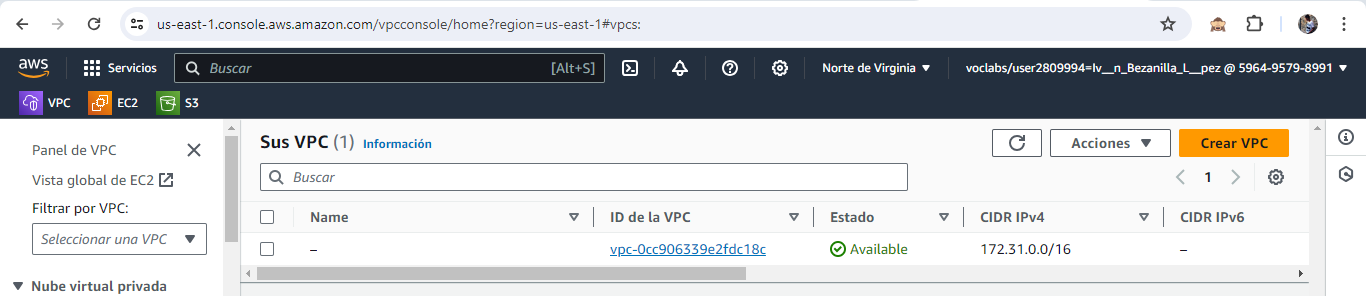
En la barra de búsqueda, escribe "VPC" y selecciona "VPC" en los resultados.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

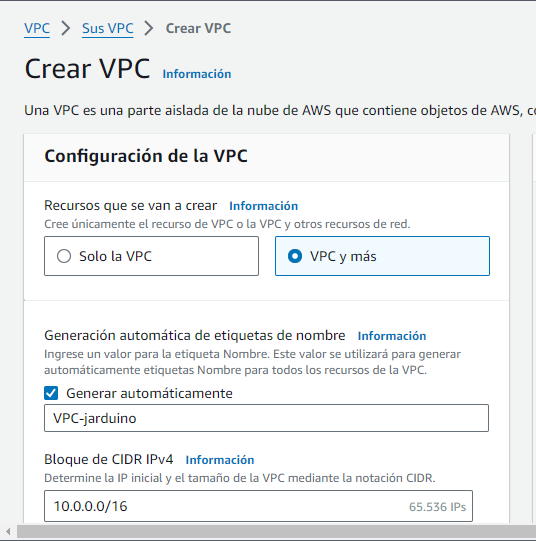
**Inicia el Asistente de Creación de VPC:**

En la página de VPC, haz clic en "Crear VPC".



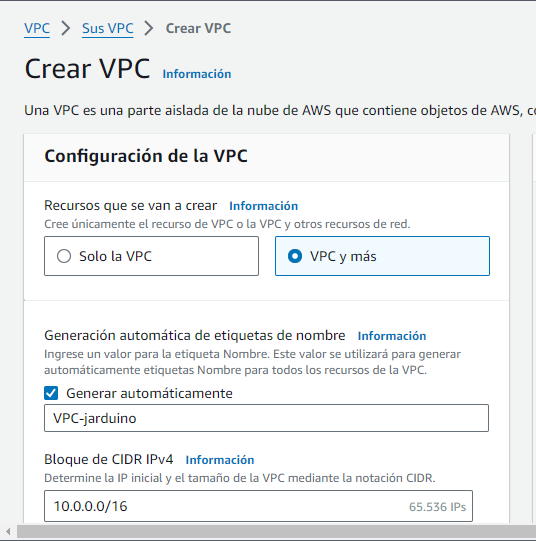
**Selecciona una Configuración de Red:**

Puedes elegir entre varias configuraciones predefinidas (por ejemplo, una VPC con una única subnet pública) o seleccionar "VPC con subredes públicas y privadas".

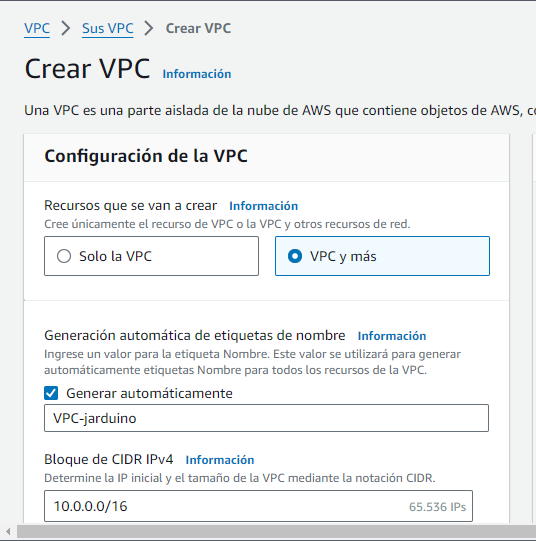


**Configura los Detalles de la VPC:**

Proporciona un nombre para tu VPC.

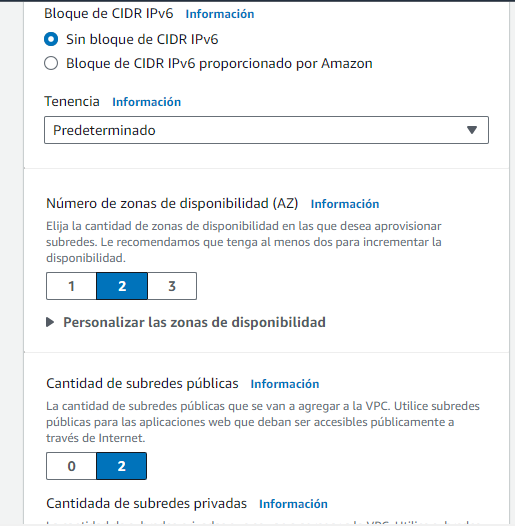


Ingresa el bloque de direcciones IP para la VPC (en formato CIDR, por ejemplo, 10.0.0.0/16).



**Configura las Subredes:**

Define las subredes dentro de la VPC. Puedes especificar subredes públicas y privadas.



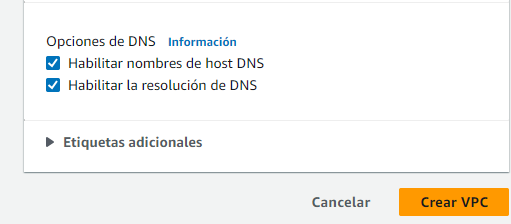


**Revisa la Configuración:**

Revisa todas las configuraciones antes de crear la VPC para asegurarte de que todo esté configurado según tus necesidades.

**Crea la VPC:**

Haz clic en "Crear VPC" para finalizar el proceso.



### Configuración de Grupos de Seguridad

Un grupo de seguridad en AWS es un mecanismo de control de acceso que actúa como un firewall virtual para tus recursos en la nube. Es una parte fundamental de la configuración de la red dentro de una VPC (Virtual Private Cloud) y te permite especificar las reglas de tráfico de red entrante y saliente para tus instancias EC2 (Elastic Compute Cloud), bases de datos RDS (Relational Database Service), y otros servicios de AWS.

Una vez creada la VPC, se dispondrá de la pestaña de grupos de seguridad, donde se pueden crear nuevos grupos y editar las reglas de entrada y salida de información dentro de nuestra red. Todos los EC2 relacionados con el proyecto serán incluidos en el mismo grupo de seguridad. Este grupo de seguridad tendrá permitido cualquier salida de datos, pero tendrá acceso restringido a sus reglas de entrada, teniendo permiso los puertos HTTP 80, SSH 22, MySQL 3306 y HTTPS 443.

**Pasos Detallados para Crear un Grupo de Seguridad en AWS:**

**Navega a la Consola de VPC:**

En la barra de búsqueda, escribe "VPC" y selecciona "VPC" en los resultados.

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

**Ve a la sección "Grupos de seguridad" en la barra lateral izquierda:**

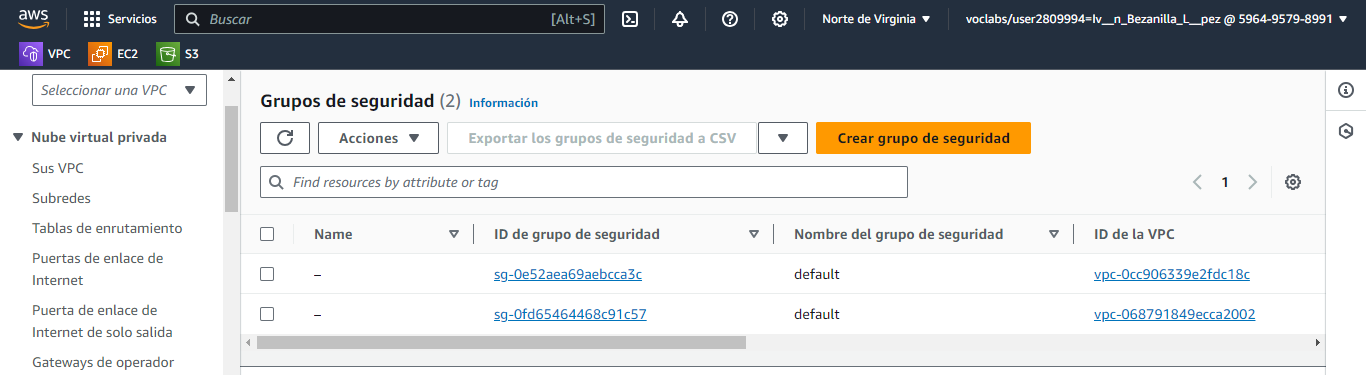
En el panel de navegación izquierdo, busca y selecciona "Grupos de seguridad" bajo la sección "Seguridad".

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

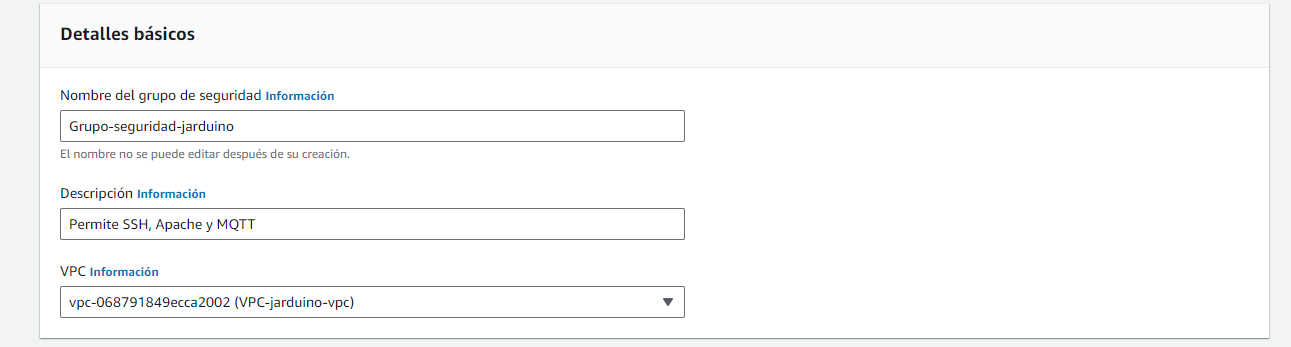
**Haz clic en el botón "Crear grupo de seguridad":**

En la parte superior de la página "Grupos de seguridad", haz clic en el botón "Crear grupo de seguridad".

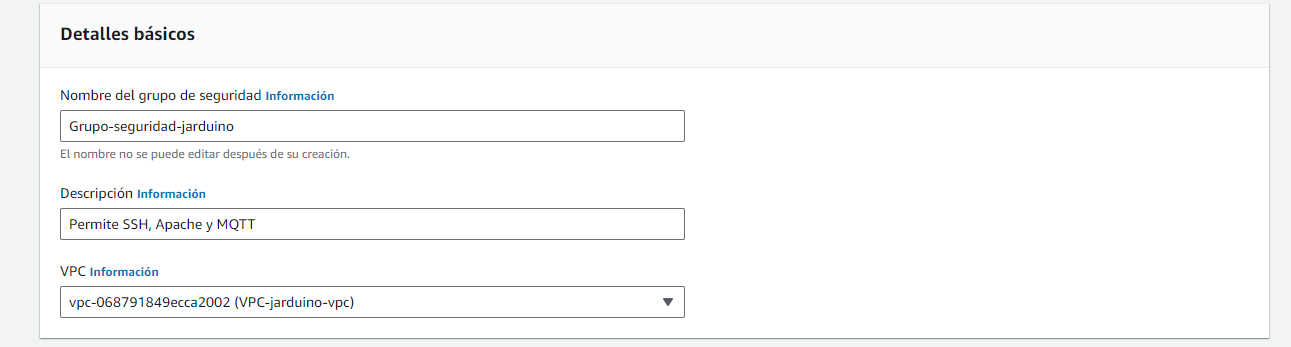


**Configura los detalles básicos:**

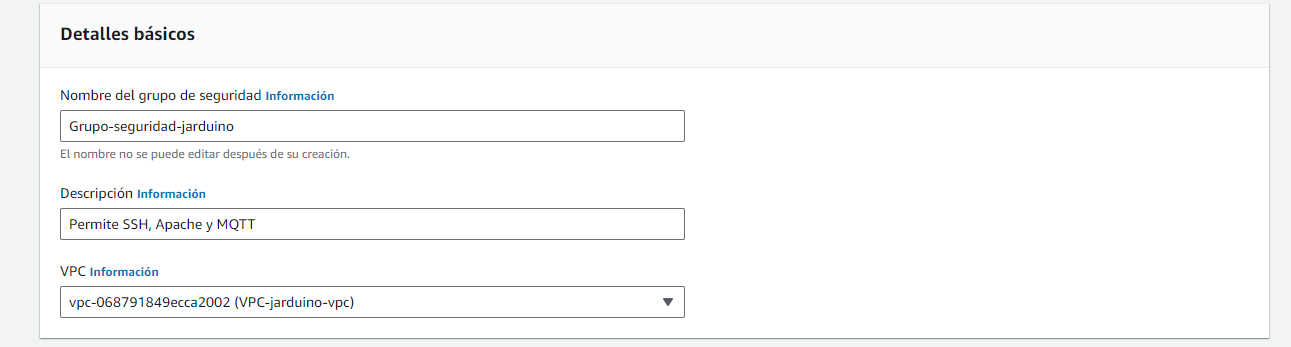
Ingresa un nombre descriptivo para tu grupo de seguridad.



Proporciona una descripción opcional.



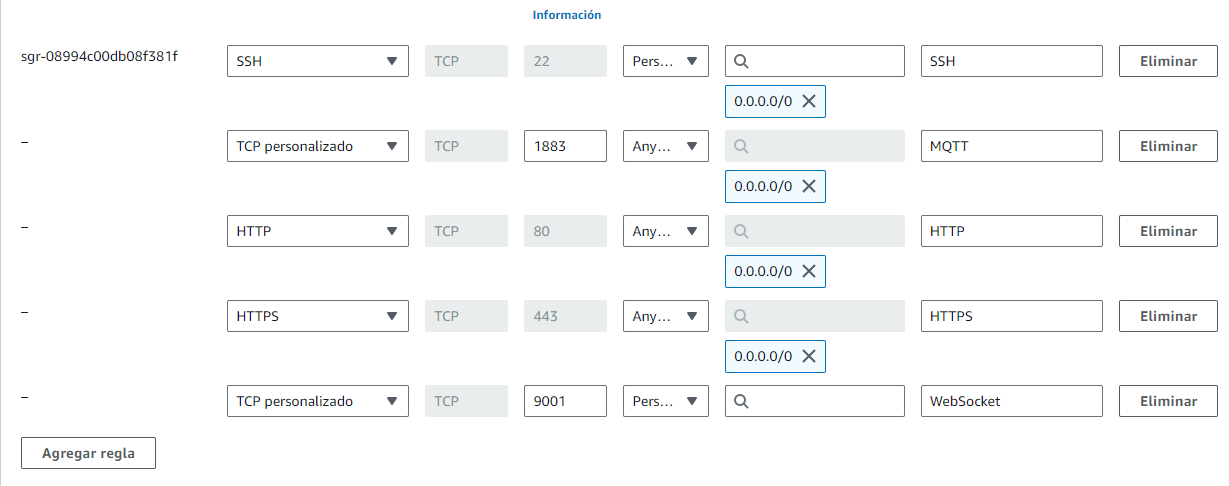
Proporciona la VPC en la cual quieres asignar este grupo de seguridad



**Configura las reglas de entrada:**

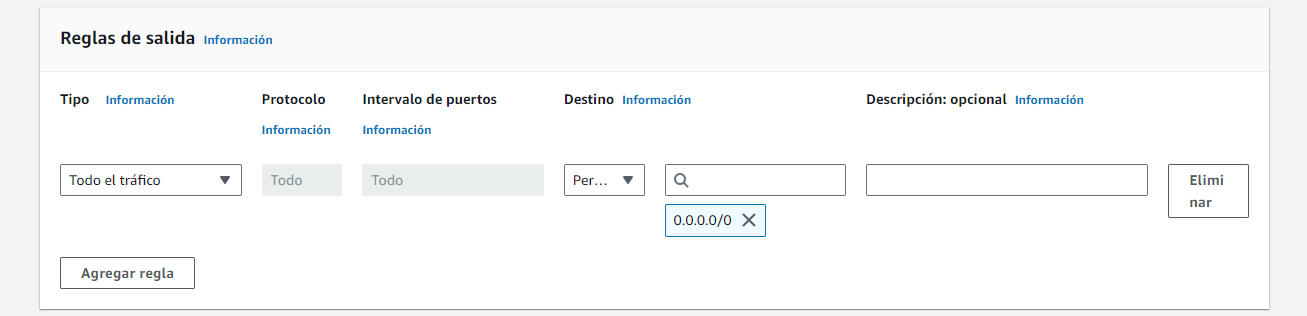
En la sección "Reglas de entrada", especifica las reglas para permitir el tráfico entrante. Por ejemplo, puedes permitir el tráfico SSH (para conexiones SSH), el tráfico HTTP (para conexiones web), el tráfico MQTT (para mosquitto) y el tráfico para el servidor WebSocket.

Define el tipo de tráfico (SSH, HTTP, etc.), el rango de direcciones IP permitido y el puerto.



**Configura las reglas de salida (opcional):**

En la sección "Reglas de salida", puedes especificar reglas para el tráfico que sale de las instancias.

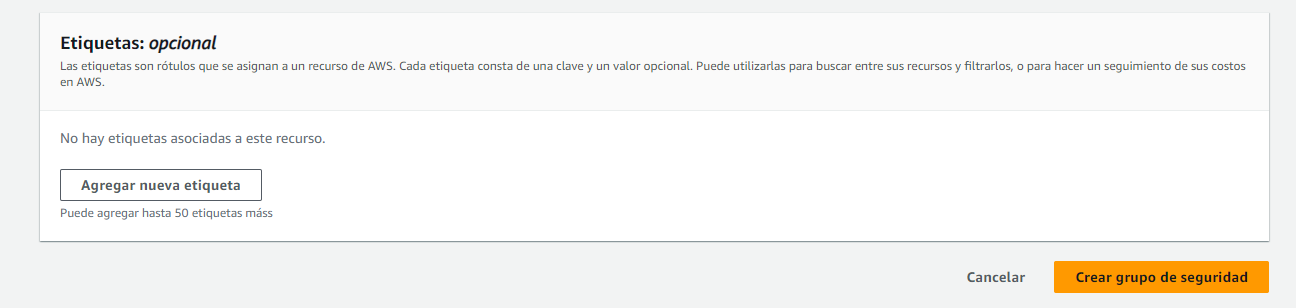


**Revisa la configuración:**

Revisa todas las configuraciones para asegurarte de que sean correctas.

**Haz clic en "Crear grupo de seguridad":**

Una vez que estés satisfecho con la configuración, haz clic en el botón "Crear grupo de seguridad".



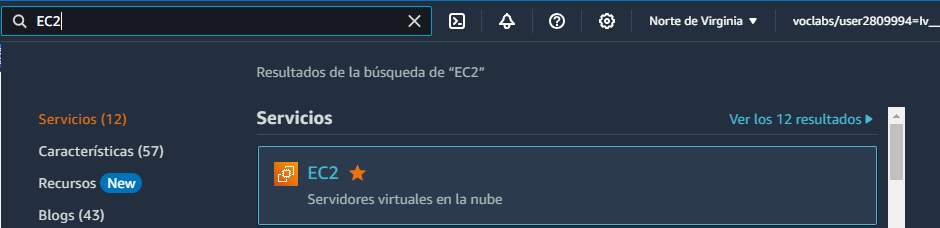
### Implementación de Instancias EC2

Una instancia EC2 (Elastic Compute Cloud) es un servicio de AWS que te permite lanzar y gestionar servidores virtuales en la nube. Estas instancias pueden ejecutar una amplia variedad de sistemas operativos y software, y son altamente escalables y configurables según tus necesidades.

**Pasos Detallados para una instancia EC2 en AWS:**

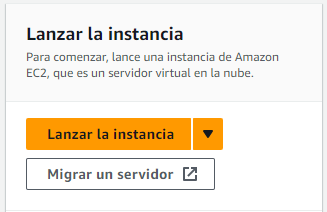
**Dirígete al Servicio EC2**

En la consola de AWS, busca y selecciona el servicio "EC2" en la sección "Compute".



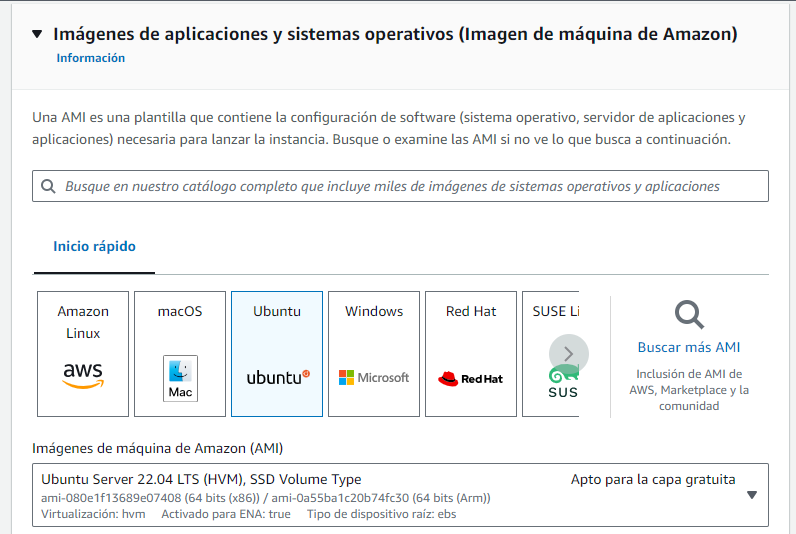
**Haz Clic en "Launch Instance"**

En el panel de control de EC2, haz clic en el botón "Launch Instance" para iniciar el proceso de lanzamiento de una nueva instancia.



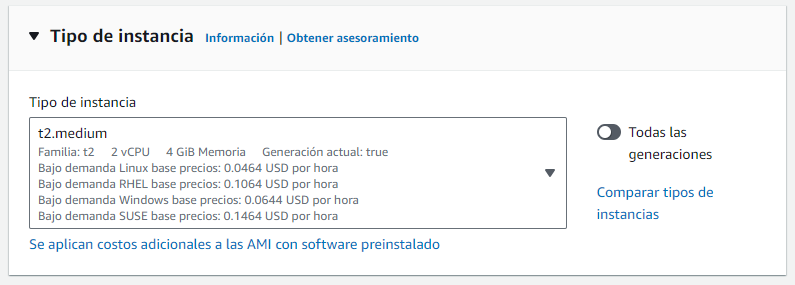
**Selecciona una Amazon Machine Image (AMI)**

En el primer paso del asistente de lanzamiento de instancias, elige una AMI que contenga el sistema operativo y el software que deseas utilizar en tu instancia. Puedes seleccionar una AMI prediseñada por AWS o una AMI personalizada.



**Elige un Tipo de Instancia**

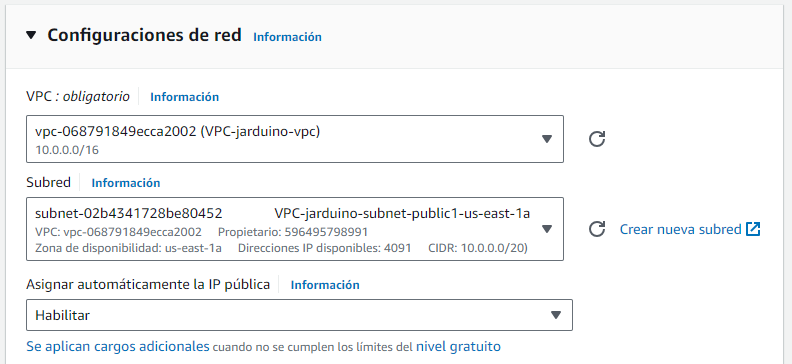
En este paso, selecciona el tipo de instancia que se ajuste a tus necesidades de capacidad de cómputo, memoria y almacenamiento. AWS ofrece una variedad de tipos de instancias, desde instancias de propósito general hasta instancias optimizadas para cómputo, memoria, almacenamiento y otros casos de uso específicos.



**Configura los Detalles de la Instancia**

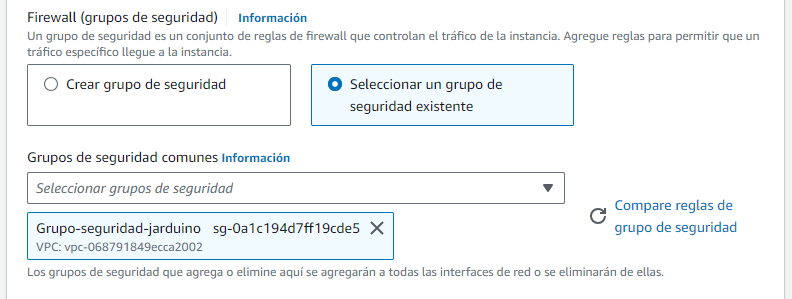
Aquí puedes configurar detalles como el número de instancias que deseas lanzar, la red en la que deseas lanzarlas, las subredes, las asignaciones de direcciones IP, etc.

Le indicamos la VPC que me hemos creado.



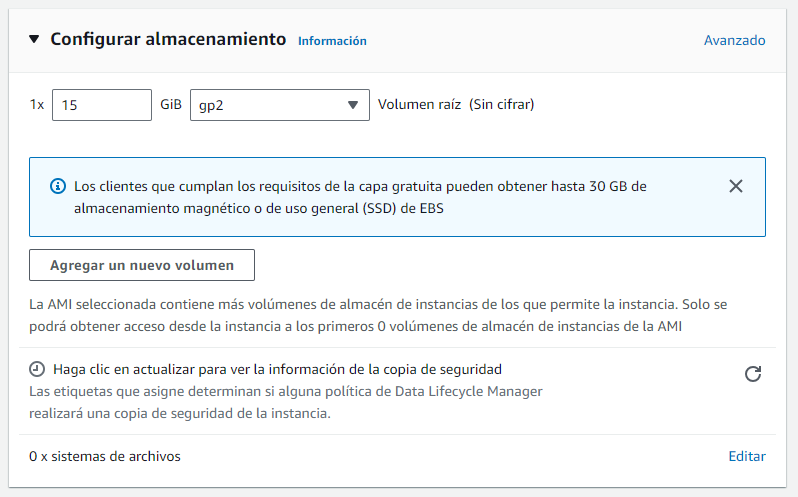
**Configura el Grupo de Seguridad**

Aquí puedes seleccionar un grupo de seguridad existente o crear uno nuevo para tu instancia EC2. Un grupo de seguridad actúa como un firewall virtual y te permite controlar el tráfico de red hacia y desde tu instancia, seleccionaremos el que hemos creado anteriormente.



**Añade Almacenamiento**

En este paso, puedes añadir y configurar el almacenamiento para tu instancia EC2. Puedes especificar el tamaño y el tipo de volumen, así como configurar opciones avanzadas como la encriptación.



**Añade Etiquetas (Tags)**

Puedes añadir etiquetas a tu instancia para facilitar la organización y la gestión de tus recursos en AWS. Las etiquetas son pares de clave-valor que puedes utilizar para identificar y filtrar recursos en la consola de AWS.

**Revisa y Lanza la Instancia**

En la pantalla de revisión, asegúrate de que toda la configuración sea correcta y luego haz clic en "Launch" para lanzar tu instancia EC2.



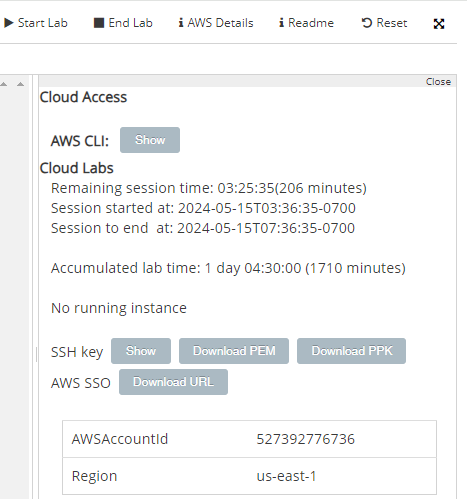
### Conexión a la instancia EC2 e instalación de servicios

Conectarse a una instancia EC2 en AWS es una tarea esencial para configurar y administrar tus recursos en la nube. Aquí tienes los pasos para hacerlo:

**Obtén la clave privada**

Si has lanzado tu instancia EC2 con una clave privada asegúrate de tenerla a mano. Esta clave se utiliza para autenticarte en la instancia.

Se puede conseguir en el apartado “AWS Details” y la puedes descargar en el botón “Download PEM”.

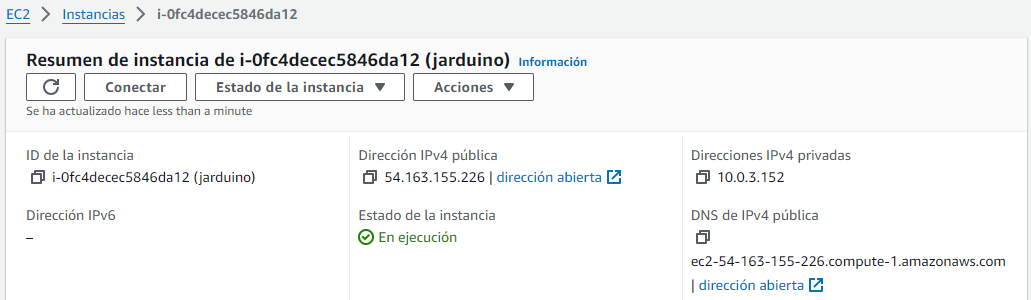


**Encuentra la dirección IP de tu instancia EC2**

Ve al panel de EC2 en la consola de AWS.

Haz clic en "Instancias" en el panel de navegación izquierdo.

Identifica tu instancia en la lista y toma nota de su dirección IP pública o DNS público.



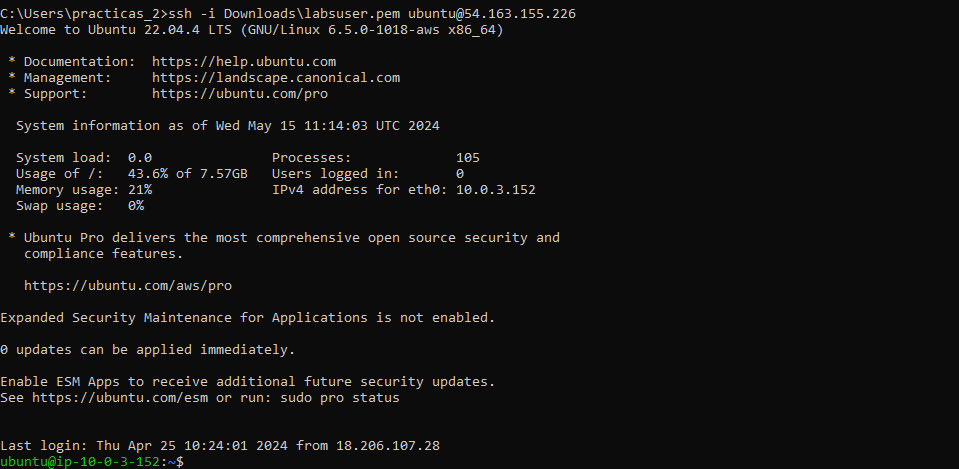
**Conéctate a la instancia EC2**

Utiliza el comando **ssh** para conectarte a tu instancia. La sintaxis es:

ssh -i /ruta/a/tu/clave-privada.pem usuario@direccion-ip-o-dns



Reemplaza **/ruta/a/tu/clave-privada.pem** con la ruta a tu clave privada, **usuario** con el usuario predeterminado del sistema operativo de tu instancia (por ejemplo, para Amazon Linux es "ec2-user"), y **direccion-ip-o-dns** con la dirección IP pública o DNS público de tu instancia.



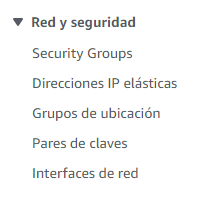
Una vez se ha accedido a la instancia EC2 por SSH, se procederá a instalar los servicios de Apache, PHP y Mosquitto, tal como se ha realizado en el entorno local. Puedes hacer clic [aquí](#_Configuración_del_Servidor) para ir a los pasos de instalación.

### Configuración de IP Elástica

Las direcciones IP son identificadores únicos asignados a dispositivos en redes informáticas para permitir la comunicación entre ellos. En el contexto de la nube, como AWS, las direcciones IP elásticas son direcciones IP estáticas que puedes asignar y reasignar a tus recursos de manera dinámica. Se denominan "elásticas" porque puedes asociarlas y desasociarlas de tus recursos en la nube según sea necesario, lo que te brinda flexibilidad y control sobre tu infraestructura.

**Crear una dirección IP elástica:**

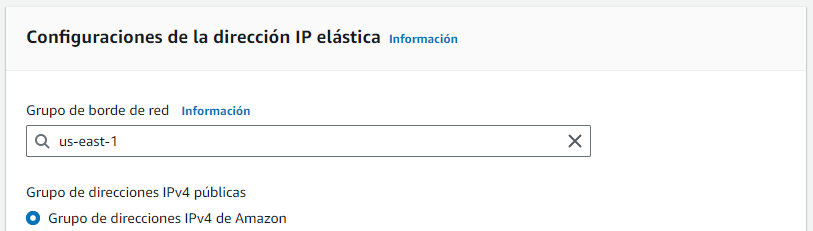
En el panel de navegación de la izquierda de EC2, haz clic en "Direcciones IP elásticas" bajo "Red y Seguridad".



Luego, selecciona "Asignar nueva dirección IP elástica".



Escoge la región de AWS donde deseas crear la dirección IP y haz clic en “Asignar”.

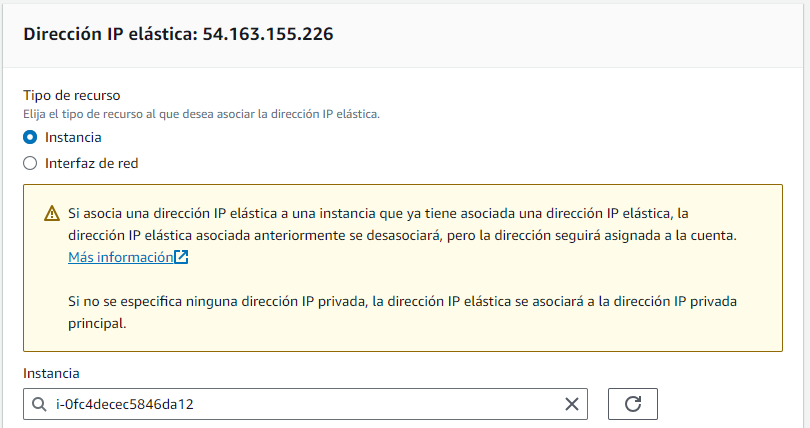


**Asignar la dirección IP elástica a una instancia EC2:**

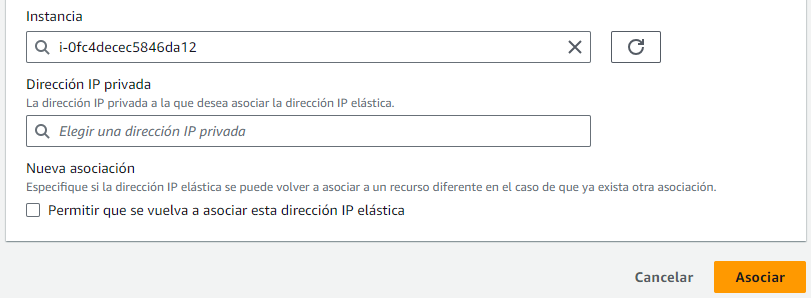
Ve al panel de "Direcciones IP Elásticas" y selecciona la dirección IP que acabas de crear. En el menú de acciones, elige "Asociar dirección IP elástica".



Selecciona la instancia EC2 a la que le quieres asignar la dirección IP Elástica.



Confirma la asignación.



### Configuración de NO-IP

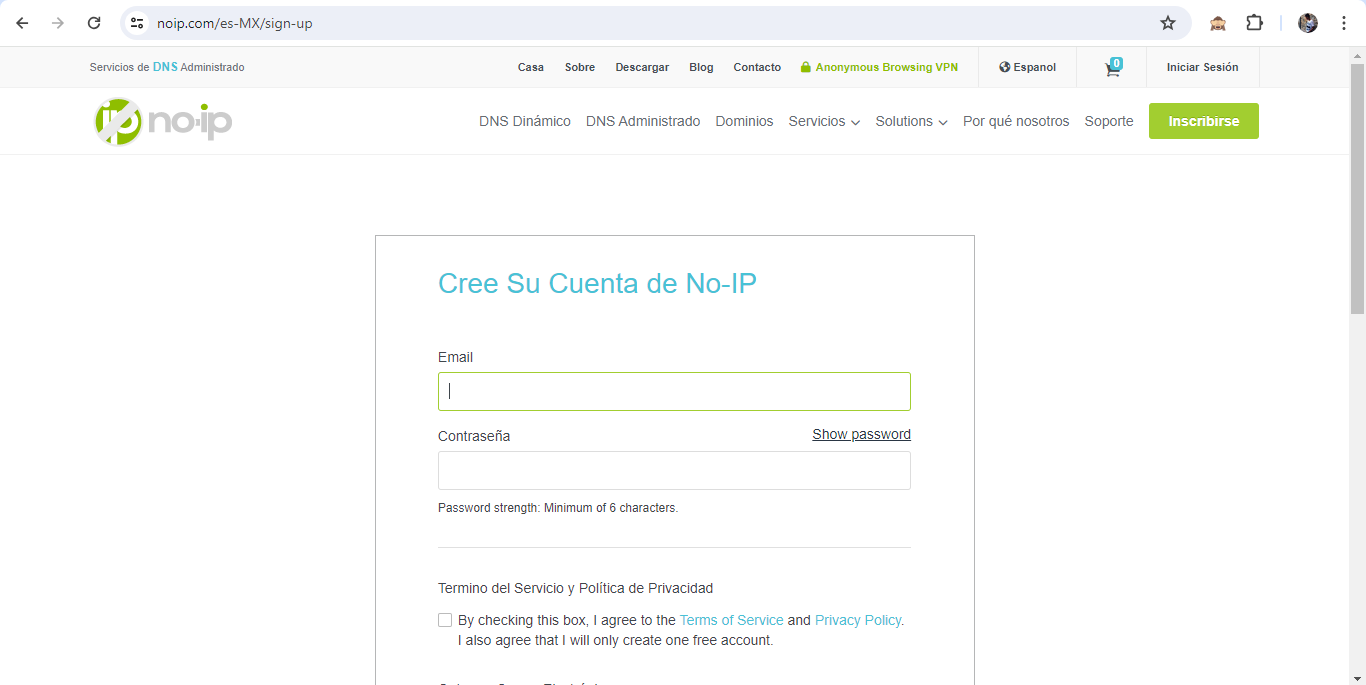
Antes de sumergirnos en la configuración de una dirección IP elástica con No-IP, es importante comprender por qué esto puede ser útil. En el mundo de la informática y las redes, las direcciones IP son como las señas de una casa en Internet: identifican de manera única un dispositivo en una red. Sin embargo, muchas veces, las direcciones IP asignadas por los proveedores de servicios de Internet (ISP) son dinámicas, lo que significa que pueden cambiar periódicamente.

No-IP es un servicio de Dynamic DNS (DNS dinámico) que te permite asignar un nombre de dominio fácil de recordar a tu dirección IP dinámica. De esta manera, aunque tu dirección IP cambie, el nombre de dominio permanece constante, lo que facilita el acceso remoto a tu servidor o dispositivo.

**Registro en No-IP:**

Ve al sitio web de No-IP (noip.com) y regístrate para obtener una cuenta.

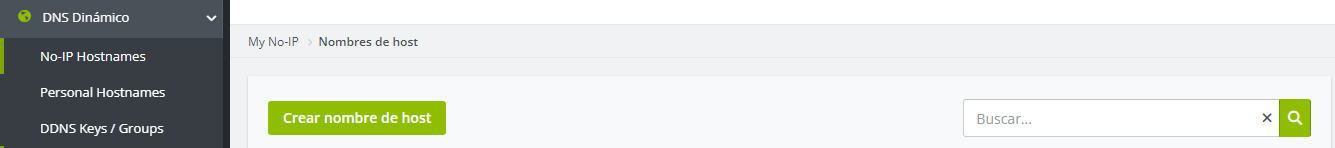
Una vez que te hayas registrado, inicia sesión en tu cuenta.



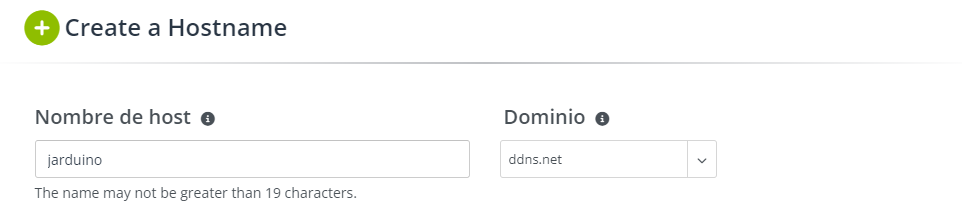
**Crear un nombre de host:**

Después de iniciar sesión, ve al panel de control de No-IP.

En la sección "Dynamic DNS", haz clic en "Crear nombre de host".



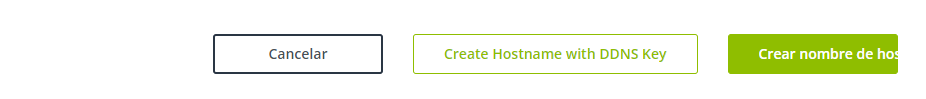
Elige un nombre de host que deseas usar para acceder a tu servidor o dispositivo y selecciona el dominio de No-IP que prefieras.



Introduce la dirección IP de la IP Elástica que has creado anteriormente.

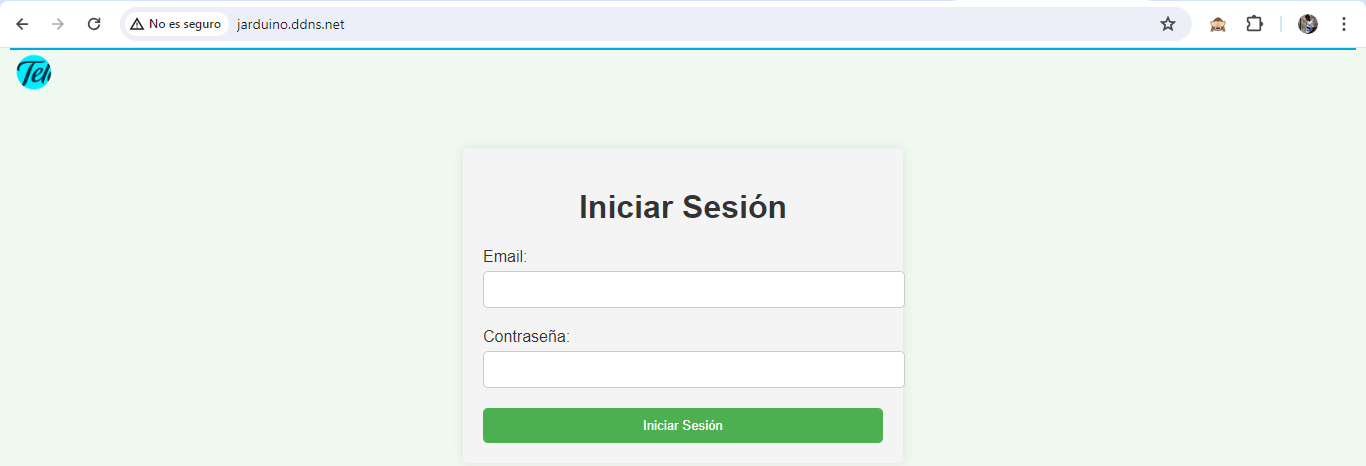


Haz clic en "Crear nombre de host".



**Prueba la configuración:**

Puedes probar la configuración ingresando el nombre de host en un navegador web o usando herramientas de ping para verificar que responda con la dirección IP correcta.

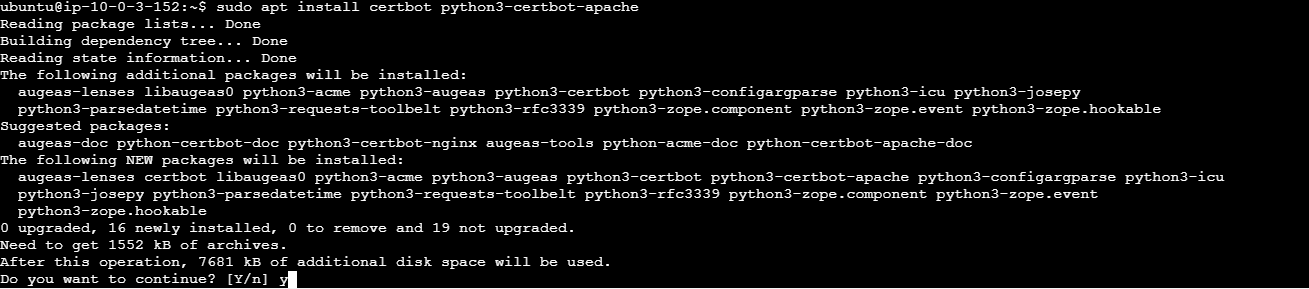


### Configurar HTTPS para apache

**Obtener un Certificado SSL/TLS**

**Instala Certbot en tu instancia EC2**: Certbot es una herramienta que facilita la obtención y la renovación de certificados SSL/TLS de Let's Encrypt. Puedes instalarlo ejecutando los siguientes comandos:

sudo apt install certbot python3-certbot-apache



Esto instalará Certbot y su plugin para Apache.

**Ejecuta Certbot para obtener el certificado SSL/TLS**: Utiliza Certbot para obtener un certificado SSL/TLS para tu nombre de dominio. Por ejemplo, ejecuta el siguiente comando:

sudo certbot --apache -d jarduino.ddns.net



Esto configurará automáticamente Apache para usar el certificado SSL/TLS y lo renovará automáticamente cuando sea necesario.

**Configurar Apache para usar HTTPS**

**Asegúrate de que Apache esté configurado correctamente**: Certbot debería haber configurado Apache automáticamente para utilizar HTTPS. Sin embargo, verifica el archivo de configuración de Apache para asegurarte de que la configuración sea correcta.

El archivo de configuración de Apache generalmente se encuentra en **/etc/apache2/sites-available/**. Deberías tener un archivo de configuración para tu sitio web, por ejemplo, **jarduino.ddns.net.conf**. Abre este archivo y asegúrate de que tenga las siguientes líneas o similares:



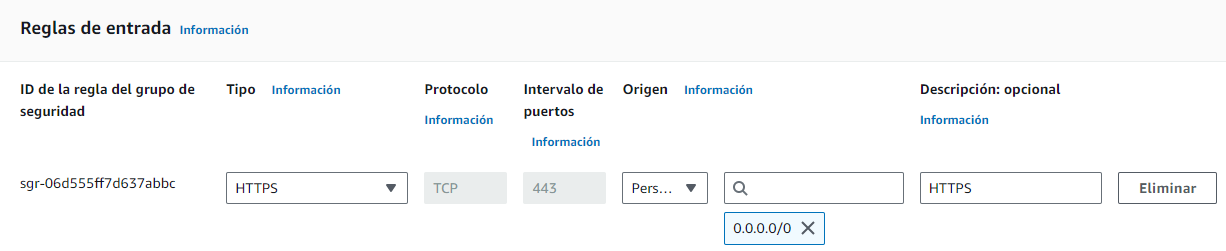
Asegúrate de que las rutas de los archivos **SSLCertificateFile** y **SSLCertificateKeyFile** coincidan con la ubicación de los certificados SSL/TLS generados por Certbot.

**Reinicia Apache**: Después de realizar cambios en la configuración de Apache, reinicia el servicio Apache para que los cambios surtan efecto:

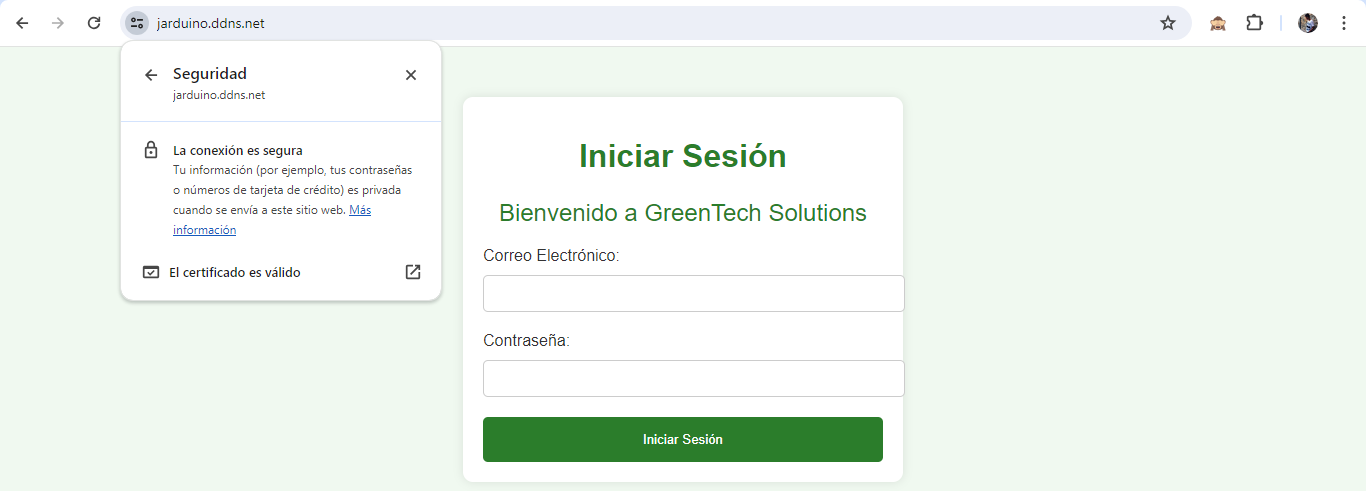
sudo systemctl restart apache2

**Actualizar el Firewall y el Grupo de Seguridad de EC2**

**Actualiza las reglas de seguridad del grupo de seguridad de EC2**: Si estás utilizando un grupo de seguridad de EC2, asegúrate de que esté configurado para permitir el tráfico HTTPS en el puerto 443 tanto para entradas como para salidas.



Una vez completados estos pasos, tu servidor Apache debería estar configurado para utilizar HTTPS con éxito. Puedes probar accediendo a tu sitio web utilizando **https://jarduino.ddns.net** en un navegador web y verificar si la conexión es segura.



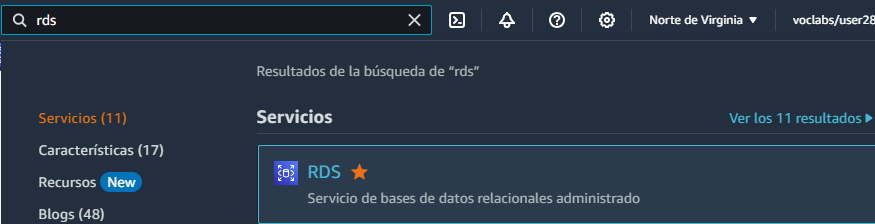
### Configuración de un Servicio de Base de Datos Relacional (RDS)

**Amazon Relational Database Service (RDS)** es un servicio de base de datos relacional totalmente administrado ofrecido por Amazon Web Services (AWS). Proporciona una forma sencilla de configurar, operar y escalar una base de datos relacional en la nube. RDS elimina la necesidad de realizar tareas de administración pesadas, como aprovisionar hardware, instalar software de base de datos, configurar la infraestructura de red y realizar copias de seguridad, ya que AWS se encarga de estas tareas para ti.

Con esta comprensión de lo que es Amazon RDS y sus capacidades, ahora podemos proceder a los pasos detallados para crear una base de datos MySQL en RDS.

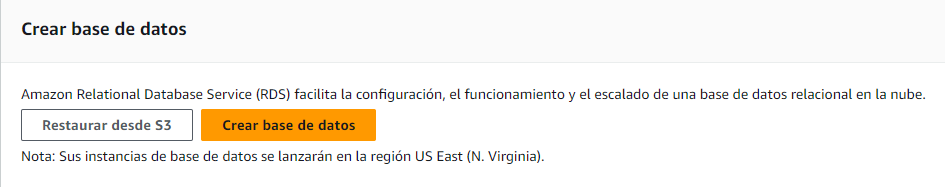
**Navega a RDS:**

En la barra de búsqueda en la parte superior, escribe "RDS" y selecciona el servicio "Amazon RDS" cuando aparezca en los resultados.



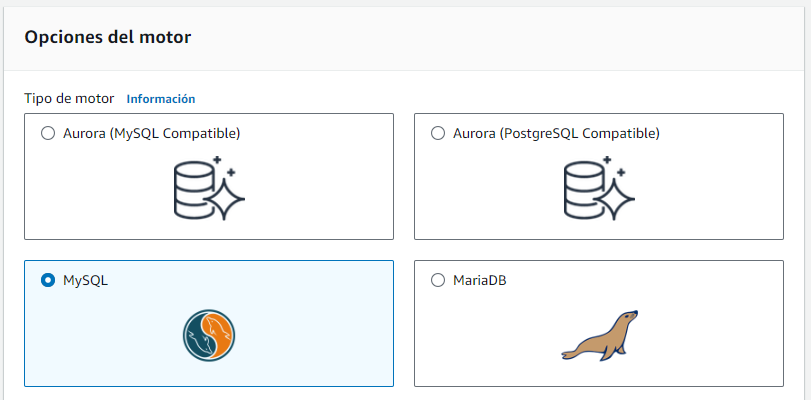
**Crea una instancia de base de datos:**

En el panel de control de RDS, busca y haz clic en "Crear base de datos".



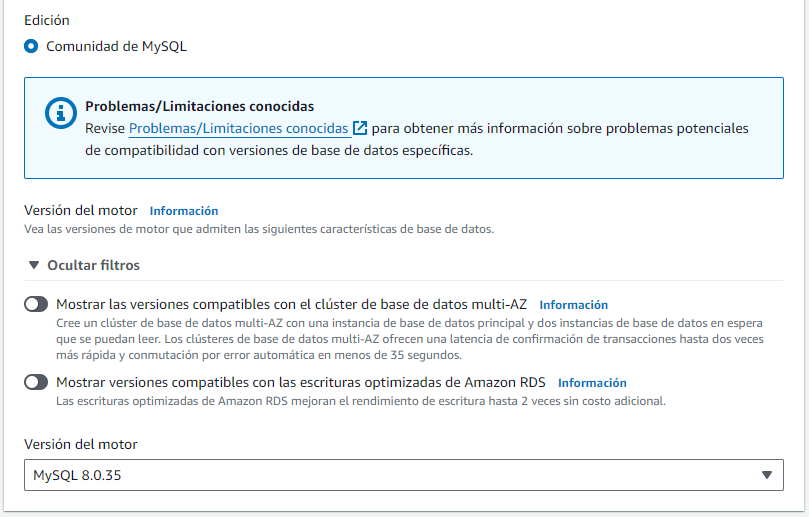
**Selecciona el motor de base de datos MySQL:**

En la página de creación de base de datos, asegúrate de que el motor de base de datos seleccionado sea "MySQL".

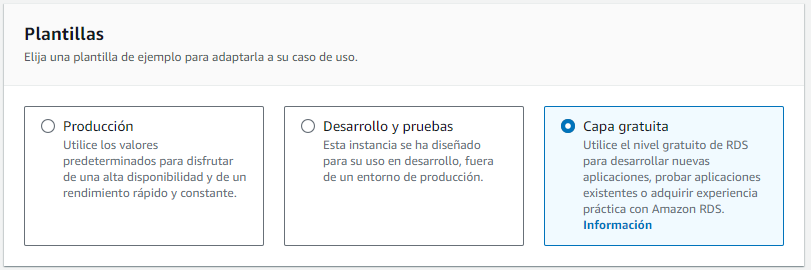


**Especifica las configuraciones de la instancia:**

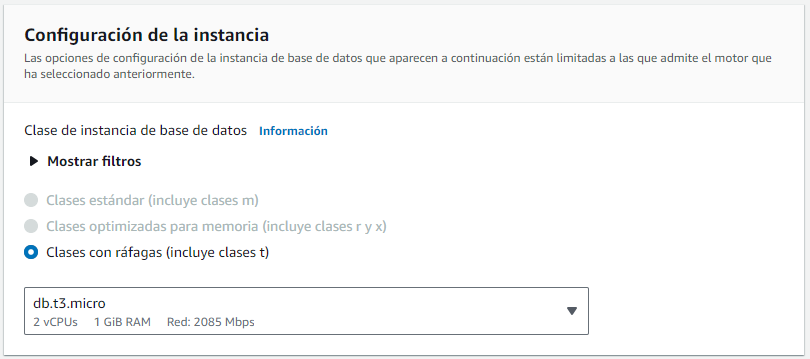
Motor: Selecciona la versión de MySQL que desees utilizar. Puedes elegir entre varias versiones disponibles.



Plantilla de uso: Elige entre las opciones de plantilla preconfiguradas según el uso que le darás a tu base de datos, como producción, desarrollo, etc.



Capacidad de instancia: Define la cantidad de CPU y memoria RAM que deseas asignar a tu instancia de base de datos.

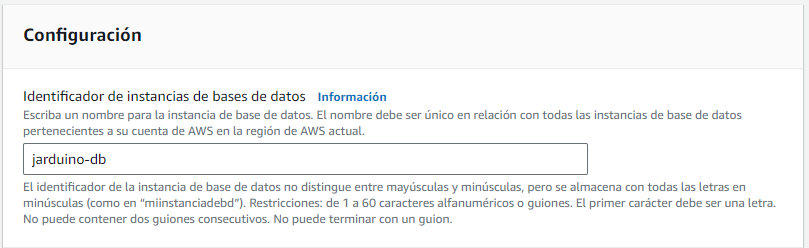


Almacenamiento: Especifica el tamaño del almacenamiento que necesitas para tu base de datos.

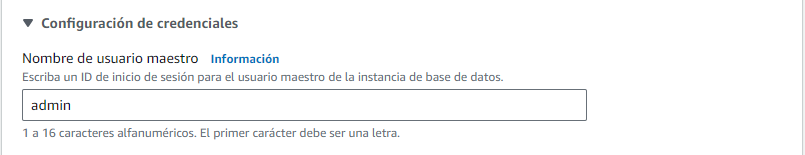


**Configura los detalles de la base de datos:**

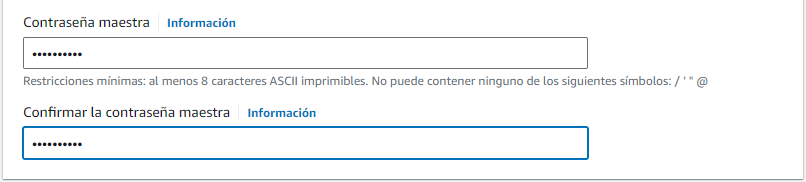
Identificador de la instancia: Ingresa un nombre único para identificar tu instancia de base de datos dentro de AWS.



Nombre de usuario maestro: Define un nombre de usuario para acceder a la base de datos.

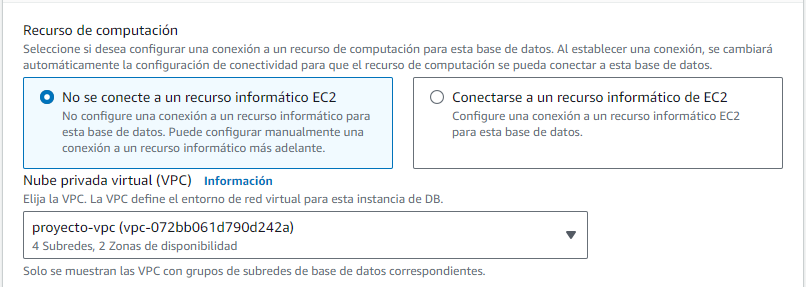


Contraseña maestra: Establece una contraseña segura para el usuario maestro.

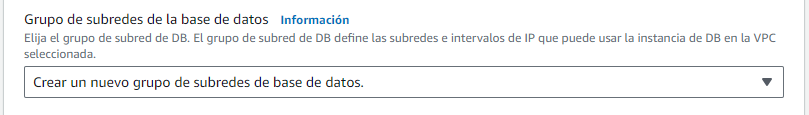


**Configura la conectividad de red:**

VPC: Selecciona la red virtual (VPC) en la que deseas lanzar tu instancia de base de datos.

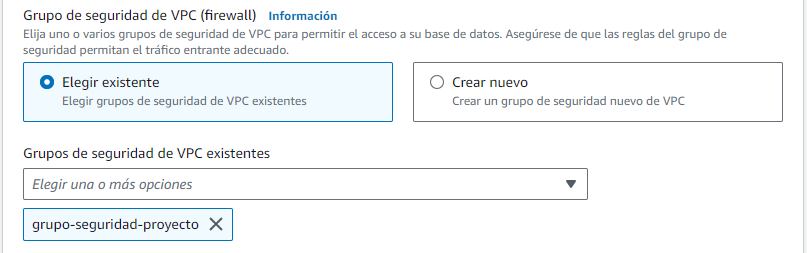


Subredes: Elige las subredes dentro de la VPC en las que deseas que esté disponible tu instancia de base de datos.



**Configura la seguridad de la base de datos:**

Grupos de seguridad: Define los grupos de seguridad que controlarán el acceso a tu instancia de base de datos. Configura los puertos y las direcciones IP permitidas.



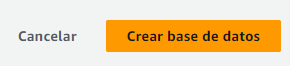
**Configura opciones adicionales:**

En esta sección, puedes establecer opciones avanzadas como la copia de seguridad automatizada, la configuración de monitoreo, la habilitación de la recuperación de bases de datos y otras configuraciones específicas de MySQL.

**Revisa y lanza la instancia:**

Antes de lanzar la instancia, revisa todas las configuraciones para asegurarte de que sean correctas.

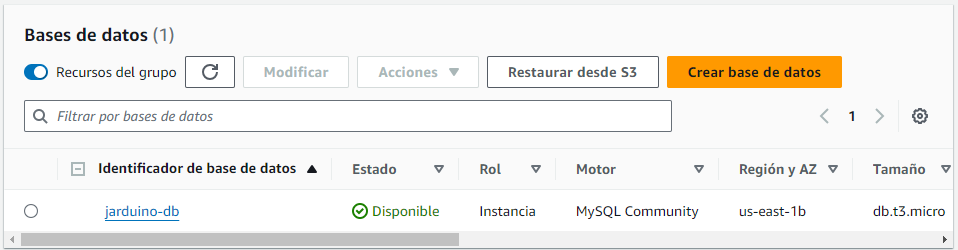
Haz clic en "Crear base de datos" para iniciar el proceso de creación de la instancia de base de datos.



**Espera a que se complete el lanzamiento:**

Una vez que hayas lanzado la instancia, espera unos minutos mientras se completa el proceso.

Una vez que la instancia esté disponible, podrás verla en el panel de control de RDS junto con su estado y detalles de conexión.

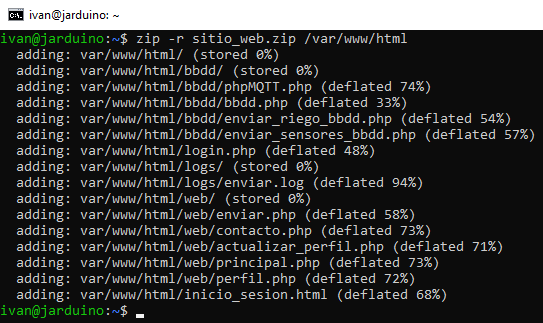


### Transición de la Página Web a la Nube (EC2)

La migración de una base de datos desde un entorno local a Amazon RDS puede realizarse de manera efectiva utilizando herramientas y comandos disponibles en sistemas Linux. En este proceso, exportaremos la base de datos desde tu entorno

**Comprimir la carpeta de la página web:** Desde la máquina virtual, abre una terminal y navega hasta la ubicación de la carpeta de la página web. Luego, comprime la carpeta en un archivo zip:

zip -r sitio\_web.zip carpeta\_sitio\_web



Esto creará un archivo **sitio\_web.zip** que contiene todos los archivos de tu página web.

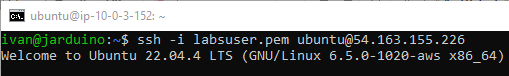
**Transferir el archivo comprimido al servidor AWS:** Utiliza SCP (Secure Copy Protocol) para transferir el archivo desde la máquina virtual al servidor AWS. Asegúrate de tener el archivo **.pem** (en este caso, **labsuser.pem**) en la máquina virtual y usa la dirección IP o el nombre de host del servidor AWS:

scp -i labsuser.pem sitio\_web.zip ubuntu@ip\_servidor\_aws:/ruta/destino



**Verificar que se ha pasado el archivo:** Conéctate al servidor AWS a través de SSH:

ssh -i labsuser.pem ubuntu@ip\_servidor\_aws

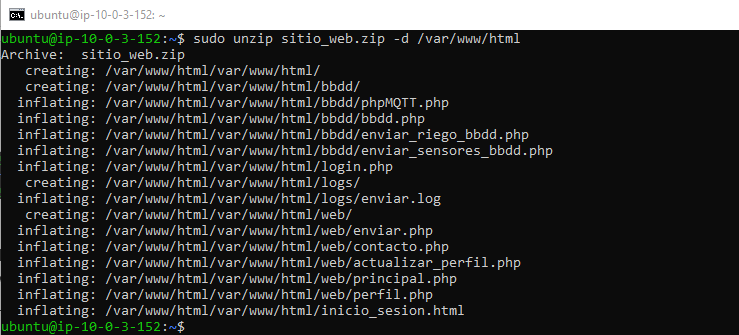


Luego, verifica que el archivo sitio\_web.zip se haya transferido correctamente:



**Descomprimir el archivo en el servidor:** Descomprime el archivo sitio\_web.zip en el directorio /var/www/html del servidor AWS:

sudo unzip /ruta/destino/sitio\_web.zip -d /var/www/html



Esto extraerá todos los archivos de la página web en el directorio adecuado del servidor.

Ahora tu página web debería estar disponible en el servidor AWS en el directorio **/var/www/html**. Recuerda ajustar los permisos de los archivos y configurar adecuadamente el servidor web si es necesario.

### Migración de la Base de Datos a la Nube (RDS)

La migración de una base de datos desde un entorno local a Amazon RDS puede realizarse de manera efectiva utilizando herramientas y comandos disponibles en sistemas Linux. En este proceso, exportaremos la base de datos desde tu entorno local utilizando comandos MySQL, luego transferiremos el archivo de exportación a una instancia EC2 utilizando SCP (Secure Copy), y finalmente importaremos los datos a tu instancia de RDS desde la instancia EC2.

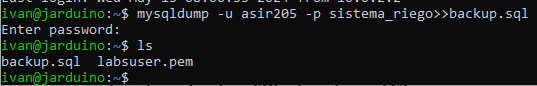
Es esencial que ambos, tu instancia de RDS y tu instancia EC2, estén en la misma VPC y tengan las configuraciones de seguridad adecuadas para permitir la transferencia de datos entre ellas. Además, asegúrate de tener acceso a tu base de datos local y a tus instancias en AWS antes de comenzar este proceso.

**Exportar la base de datos desde tu entorno local:**

Accede a tu servidor local que contiene la base de datos que deseas migrar.

Utiliza el comando mysqldump para exportar la base de datos a un archivo SQL.

mysqldump -u <nombre\_de\_usuario> -p<contraseña> nombre\_de\_base\_de\_datos > > backup.sql



**Transferir el archivo SQL a la instancia EC2:**

Conéctate a tu instancia EC2 utilizando SSH.

Utiliza el comando scp para copiar el archivo SQL desde tu máquina local a la instancia EC2. Por ejemplo:

scp -i <ruta\_a\_tu\_clave\_privada> backup.sql usuario@direccion\_de\_la\_instancia\_ec2:/ruta\_en\_la\_instancia\_ec2

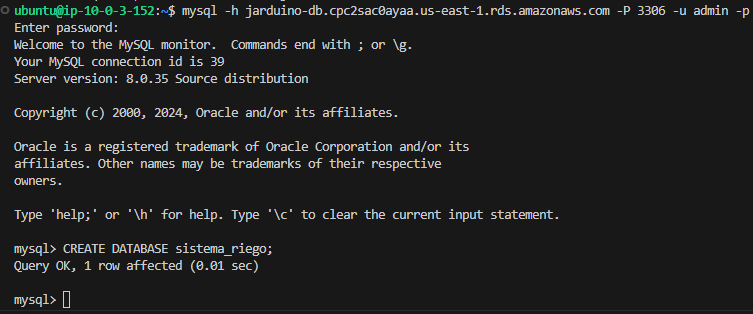


Y compruebas en la instancia EC2 que este el archivo sql que acabas de pasar.



**Importar la base de datos en el RDS:**

Conéctate a tu instancia EC2 utilizando SSH. Después conéctate al RDS desde tu instancia EC2 para crear una base de datos vacía que será donde se importará el archivo SQL.



Utiliza el comando mysql para importar el archivo SQL a la base de datos en el RDS. Por ejemplo:

mysql -h <host\_de\_RDS> -u <nombre\_de\_usuario> -p<contraseña> <nombre\_de\_base\_de\_datos> < backup.sql



## Verificación y Validación

### Pruebas de Funcionamiento

### Confirmación del Éxito del Proyecto

# ANÁLISIS DE RIESGOS



## Fortalezas

**Innovación Tecnológica**: El proyecto se basa en el uso de tecnología avanzada como Arduino, sensores y actuadores para ofrecer una solución moderna y eficiente.

**Sostenibilidad Ambiental**: Promueve prácticas agrícolas responsables al minimizar el uso de agua y energía, lo que lo hace atractivo para usuarios preocupados por el medio ambiente.

**Flexibilidad de Aplicación**: El sistema se adapta a diferentes entornos, desde hogares hasta empresas agrícolas, lo que amplía su alcance y potencial de mercado.

**Interfaz Intuitiva**: La interfaz de usuario intuitiva facilita su uso tanto para usuarios domésticos como para agricultores, lo que aumenta su accesibilidad y aceptación.

## Debilidades

**Costo Inicial**: La implementación del sistema puede requerir una inversión inicial significativa en equipos y componentes electrónicos, lo que podría ser una barrera para algunos usuarios.

**Dependencia de la Tecnología**: La efectividad del sistema está ligada al correcto funcionamiento de la tecnología utilizada, lo que podría generar problemas en caso de fallos técnicos o errores de programación.

**Complejidad de Instalación**: Para algunos usuarios, especialmente aquellos con poca experiencia técnica, la instalación y configuración del sistema podría resultar complicada y requerir asistencia especializada.

## Oportunidades

**Mercado en Crecimiento**: Existe una creciente demanda de soluciones sostenibles y automatizadas para el cuidado de plantas, especialmente en el contexto de la agricultura urbana y la preocupación por el cambio climático.

**Colaboraciones Estratégicas**: El proyecto podría beneficiarse de alianzas con empresas de tecnología, fabricantes de equipos agrícolas o instituciones de investigación para mejorar y promover su desarrollo.

**Expansión Internacional**: La tecnología utilizada en el proyecto tiene el potencial de ser adoptada en mercados internacionales donde la agricultura sostenible y la automatización están en aumento.

## Amenazas

**Competencia del Mercado**: Existe la posibilidad de que surjan competidores que ofrezcan soluciones similares a precios más bajos o con características adicionales, lo que podría afectar la participación en el mercado.

**Regulaciones Ambientales**: Cambios en las regulaciones relacionadas con el uso del agua y la gestión de residuos podrían impactar en la demanda y la aceptación del sistema en el mercado.

**Riesgos Tecnológicos**: La rápida evolución de la tecnología podría hacer que el sistema se vuelva obsoleto en poco tiempo si no se realizan actualizaciones y mejoras continuas para mantener su relevancia y competitividad.

# BIBLIOGRAFÍA

En este apartado se recogen todas las referencias que se ha empleado para llevar a cabo el proyecto: libros, recursos web, etc

<http://www2.imse-cnm.csic.es/~juanle/Documents/MasterBacherlorTheses/AntonioBarciaGarcia_2017.pdf>

<https://digitalocean.com/community/tutorials/>

<https://www.digitalocean.com/community/tutorial-collections/how-to-install-and-secure-the-mosquitto-mqtt-messaging-broker>

<http://codigoelectronica.com/blog/instalar-mosquitto-ubuntu>

<https://forum.arduino.cc/t/arduino-connected-to-esp-01-and-send-data-to-sql/862744/34>

<https://www.arduino.cc/en/software>

<https://chat.openai.com/>

<https://www.circuito.io/app?components=512,10167,11021,13322,13678,3061987>

<https://fritzing.org/>

<https://forum.fritzing.org/>

<https://github.com/WayinTop>

<https://docs.aws.amazon.com/ec2/>

<https://docs.aws.amazon.com/rds/>

<https://www.infoautonomos.com/plan-de-negocio/analisis-dafo/>

<https://dafo.ipyme.org/Home>

<https://www.canva.com/es_es/graficos/analisis-dafo/>

<https://pi3g.com/es/correct-way-to-instantiate-paho-client-in-javascript-for-wss-secure-websocket/>

<https://www.geeksforgeeks.org/how-to-create-a-websocket-connection-in-javascript/>