Manual técnico

**Centro de formación**

Centro textil y de gestión industrial

**Programa de formación**

Análisis y desarrollo de sistema de información

**Desarrolladores del proyecto**

Yeison Alejandro Soto Gallego

Alejandro Buitrago Varón

**Fecha**

25/04/2021

ViveRegistro

Contenido

[1. Objetivo 4](#_Toc72743111)

[1.1. Objetivos Específicos 4](#_Toc72743112)

[2. Alcance 4](#_Toc72743113)

[3. Requerimientos Técnicos 4](#_Toc72743114)

[3.1. Software 4](#_Toc72743115)

[3.2. Hardware 6](#_Toc72743116)

[3.3. Requerimientos mínimos del hardware: 7](#_Toc72743117)

[3.4. Requerimientos mínimos software: 7](#_Toc72743118)

[4. Herramientas utilizadas para el desarrollo del sistema 7](#_Toc72743119)

[5. Instalación 9](#_Toc72743120)

[6. Configuración del proyecto 25](#_Toc72743121)

[7. Análisis 25](#_Toc72743122)

[8. Diseño de la arquitectura 36](#_Toc72743123)

[8.1. Fichas Técnicas 36](#_Toc72743124)

[8.2. Diagrama Casos de uso 38](#_Toc72743125)

[38](#_Toc72743126)

[8.3. Diagrama de clases 39](#_Toc72743127)

[8.4. Diccionario de datos 40](#_Toc72743128)

[8.5. Modelo relacional. 43](#_Toc72743129)

[8.6. Modelo entidad relación 44](#_Toc72743130)

[9. Links de instalación 45](#_Toc72743131)

[9.1. Git 45](#_Toc72743132)

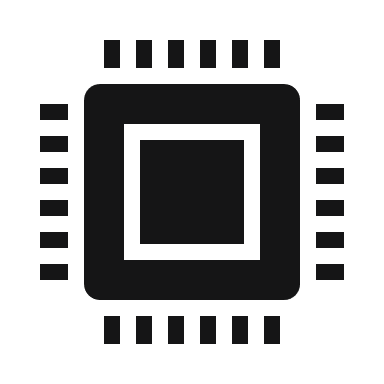
[9.2. XAMPP 45](#_Toc72743133)

[9.3. Visual Studio Code 45](#_Toc72743134)

[9.4. NodeJs 45](#_Toc72743135)

[9.5. Angular 45](#_Toc72743136)

[9.6. Arduino IDE 45](#_Toc72743137)



**ViveRegistro**

**Manual técnico**

# Objetivo

Se ha creado dicho documento con el propósito de ilustrar, mostrar y explicar el cómo fue hecho el sistema, además, dar referencias del cómo interactuar con el programa para que sea actualizado, además de que, se le pueda dar un correcto mantenimiento en caso de reportar fallas, o sea que, se hizo con el fin de dar una guía a los futuros programadores sobre el sistema de **ViveRegistro**.

## Objetivos Específicos

* Dar una guía completa del sistema.
* Mostrar código fuente y base de datos para futuras actualizaciones del sistema.
* Indicar requisitos para correr e instalar el proyecto.
* Ilustrar y mostrar la documentación, entrevistas y encuestas que ayudaron a crear el sistema.

# Alcance

El manual irá dirigido a todos los programadores que vayan a manipular, actualizar y hacer mantenimiento a este sistema.

# Requerimientos Técnicos

## Software

* **Navegador Web**: Podrá ejecutarse y manipular desde cualquier navegador (es multiplataforma), sin embargo, para apreciar mejor el diseño, se recomienda usar Google Chrome.
* **Gestores de bases de datos**: Se usaron dos tipos de gestores de bases de datos para el sistema, uno es **MYSQL,** junto a su servidor **XAMPP,** el cual se usó para manipular y gestionar las APIS, las cuales sirvieron para obtener todas las funcionalidades de un Crud (en este caso fueron las estadísticas, los formularios de entrada para crear los sensores y la tabla donde se pueden editar y eliminar dichos sensores). Por otra parte, CloudStore, se usó para crear todas las autenticaciones del usuario en el sistema, además, para guardar, editar, listar y eliminar los registros de los usuarios creados en la plataforma (tener en cuenta que cloud store es una base de datos no relacional la cual proporciona firebase).
* **Instalar Angular y TypeScript (puede ser cualquier versión desde la 10 en adelante)**: Con Angular como framework y TypeScript como lenguaje de programación, se hizo todo el Frontend del sistema, en donde están los formularios, Crud, interfaces gráficas, etc.
* **Instalar NodeJS**: Es una librería para ejecutar JavaScript del lado del servidor, con este, se hizo las APIS que obtienen, registran, editan y eliminan los registros de los sensores, estados y plantas, además, con el npm de NodeJS se instalaron todos los paquetes y dependencias del sistema.
* **Instalar o tener un editor de código**: Puede ser cualquiera, se aconseja instalar uno con terminal integrada para correr el sistema y descargar las dependencias necesarias, si no, también se puede usar el símbolo de sistema de Windows.
* **Instalar git y crearse una cuenta en GitHub**: Para el manejo de versiones del proyecto, se usó git; Para almacenar estos cambios y trabajar simultáneamente en equipo, se usó GitHub, que es su repositorio remoto donde están guardados tanto el Backend y el Frontend del sistema.
* **Instalar el IDE de Arduino**: Para la ejecución del código de Arduino y la manipulación de este mismo, se requiere tener un IDE con la capacidad de hacer dichas funciones.

## Hardware

* **Computadora con todos sus componentes:** CPU, monitor, mouse, teclado, etc.
* **Conexión a internet:** Esto es un requisito, ya que, aunque el proyecto si puede correr sin internet, ciertas funcionalidades, autenticaciones y el uso de la base de datos CloudStore no estaría disponible, ya que, funciona por medio de servidores los cuales necesitan internet para traer todos los datos y respuestas a las peticiones del cliente.
* **Arduino** (se recomienda usar el Arduino UNO): Este se usará para la conexión con los diversos sensores que se mostraran más adelante, donde se capturaran y se enviaran a otras máquinas, en este caso, a un computador de mesa.
* **Jumpers:** es un tipo de socket rectangular de plástico que a su vez tiene en su interior dos o más sockets metálicos, estos nos servirán para crear conexiones temporales entre los sensores y el Arduino.
* **Cable USB Ch340:** Este cable nos servirá para conectar el Arduino con el pc, y así, mandar los datos que llegue al Arduino por parte de los diferentes sensores.
* **Sensor de humedad:** Si se requiere comprobar y testear el código o la funcionalidad completa del sistema, se requiere un sensor “hl69”, o mejor conocido como sensor de humedad en tierra.
* **Sensor de lluvia:** Si se requiere comprobar y testear el código o la funcionalidad completa del sistema, se requiere un sensor “yl83”, o mejor conocido como sensor de lluvia.
* **Sensor de temperatura:** Si se requiere comprobar y testear el código o la funcionalidad completa del sistema, se requiere un sensor “dhd11”, o mejor conocido como sensor digital de humedad y temperatura relativa.

# Requerimientos mínimos del hardware:

* **Procesador:** Procesador Intel® Core™2 Duo all series | Procesador AMD Athlon ll x4 all series
* **Disco duro:** 100 GB
* **Memoria RAM:** 6 GB
* **tipo de unidad central de proceso:** 64 bits

## Requerimientos mínimos software:

* **Sistema operativo:** No se ha hecho un testeo del software en otros sistemas operativos, por eso**, se recomienda usar Windows 10**, ya que ahí se realizó originalmente el proyecto
* **Privilegios de administrador:** Si
* **Activación de scripts (Windows):** Si

# Herramientas utilizadas para el desarrollo del sistema

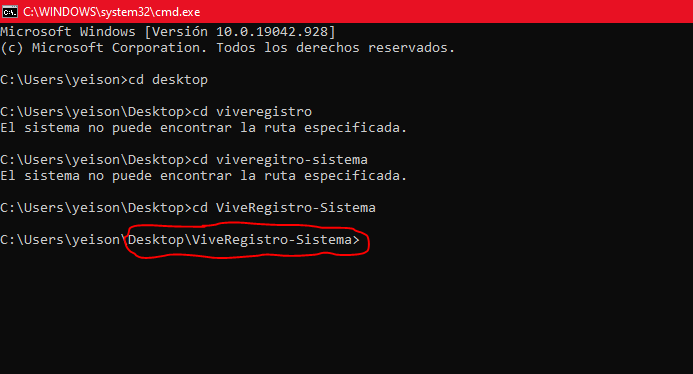
* **Navegador Web**: Se usó para el manejo de las interfaces gráficas y sus respectivos componentes, además, se usó para el testeo del aplicativo.
* **Firebase**: Se usó CloudStore (base de datos que proporciona firebase) esta, se utilizó para crear todas las autenticaciones del usuario en el sistema, además, para guardar, editar, listar y eliminar los registros de los usuarios creados en la plataforma (tener en cuenta que cloud store es una base de datos no relacional).
* **MySQL**: Se usó como gestor de base de datos del sistema, además, sirvió para hacer uso de las APIS, con las cuales se pueden crear, leer, listar, editar y eliminar todos los sensores, plantas y estados.
* **XAMPP**: Se usó este servidor para la manipulación de las bases de datos con **MYSQL,** junto a su servidor de Apache.
* **Angular**: Se usó este framework para maquetar, estructurar, crear interfaces, crear formularios, etc.
* **Bootstrap**: Se usó este framework para dar diseño a los diferentes elementos HTML del sistema.
* **NodeJS**: Es una librería para ejecutar JavaScript del lado del servidor, con este, se hizo las APIS que obtienen, registran, editan y eliminan los registros de los sensores, estados y plantas, además, con el npm de NodeJS se instalaron todos los paquetes y dependencias del sistema.
* **Editor de código (VSCODE)**: Se usó este editor de código, ya que les resultó mucho más cómodo a los programadores encargados a causa de, las múltiples herramientas que ofrece, así como, extensiones, terminal integrada, etc.

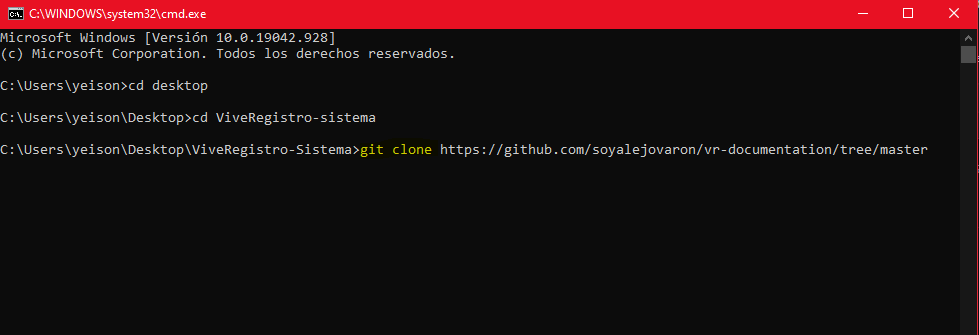
# Instalación

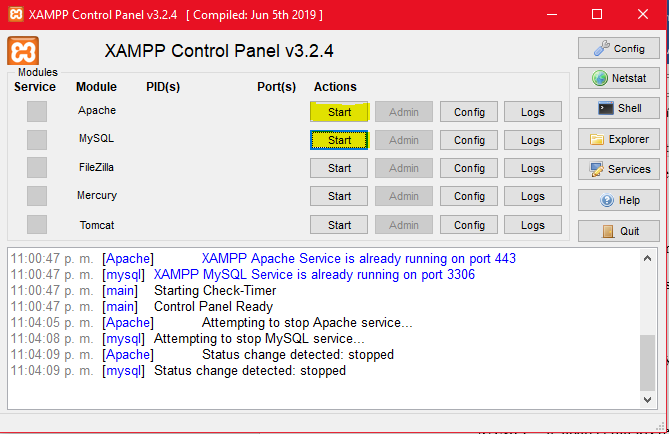
1. Primero debemos de clonar los dos proyectos (tanto el Backend como el Frontend), esto lo haremos con git, a continuación, dejaremos el link para acceder a cada uno de los repositorios:

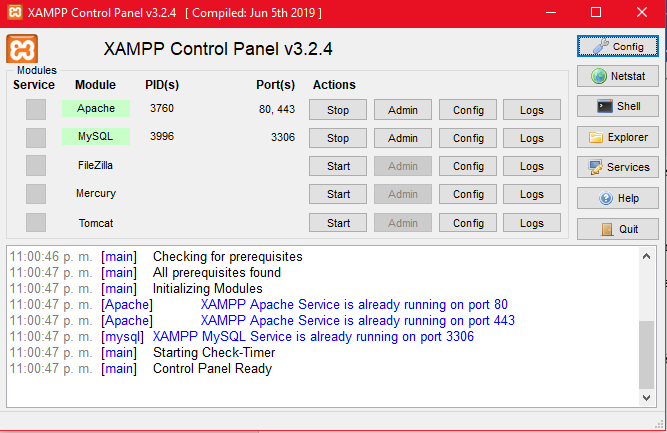
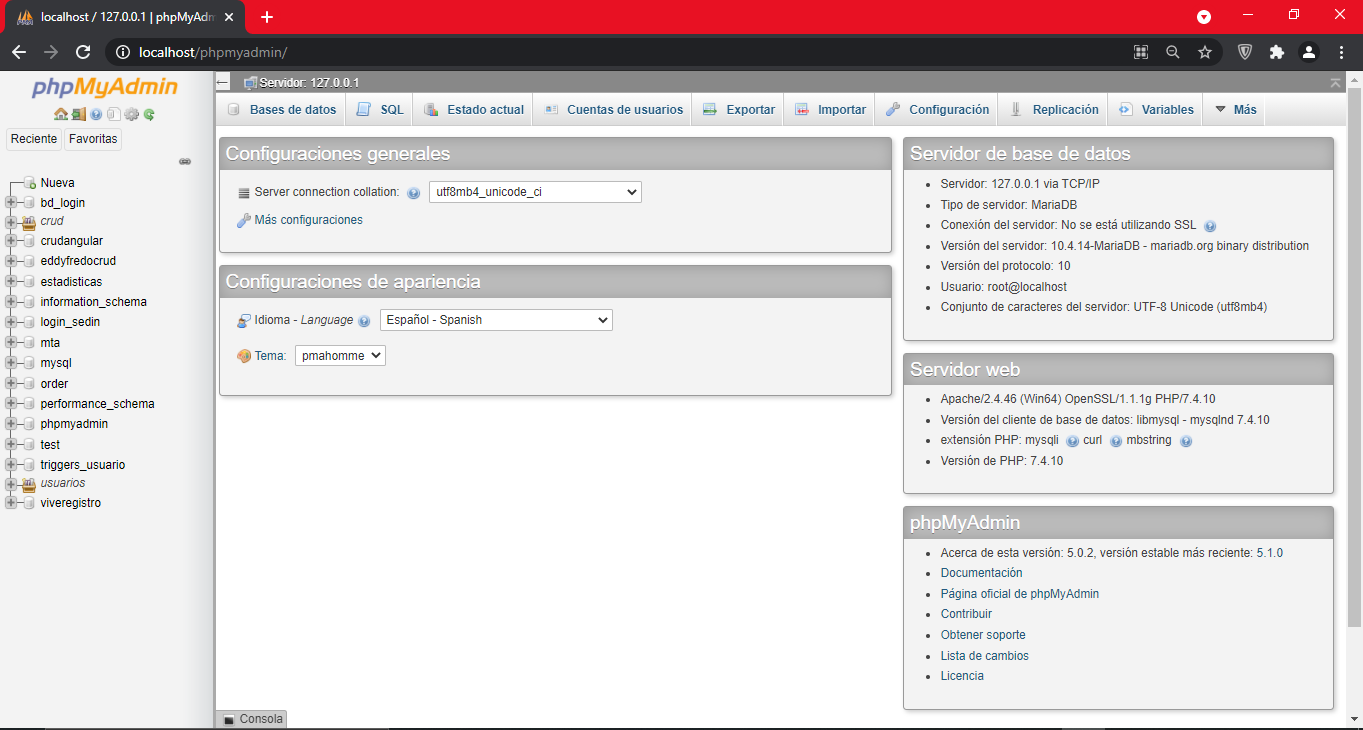
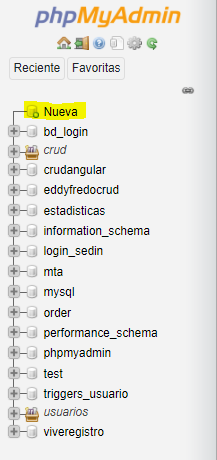
* El proyecto de **Back**en **Node.js** lo encuentras aquí: <https://github.com/soyalejovaron/vr-backend/tree/master>
* El proyecto de **Front**en **Angular** lo encuentras aquí: <https://github.com/soyalejovaron/vr-frontend/tree/master>
* La documentación completa del **sistema** lo encuentras aquí: <https://github.com/soyalejovaron/vr-documentation/tree/master>

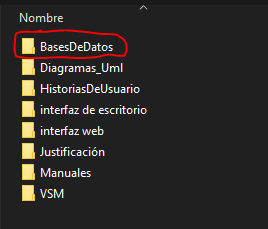
Cualquier problema repórtalo a los administradores.

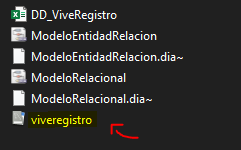
* 1.  Para clonar los proyectos, primero debes crear una carpeta, en donde vayas a guardar cada uno de estos proyectos, luego, lo único que debes hacer es irte al CMD de Windows (el símbolo de sistema), una vez allí (recordando que debes de tener git descargado en tu computadora), deberás de acceder a esa carpeta, usando el comando “cd”, el cual te permitirá acceder a carpetas del computador, una vez te posiciones en la carpeta, podrás clonar los proyectos, así como lo veras a continuación:
  2. Escribirás el comando “git clone” (es un comando de git), y seguido, pondrás la URL del repositorio que deseas clonar (puede ser el Frontend o el Backend en este caso), así como veras a continuación, en este caso, clonaremos la documentación:

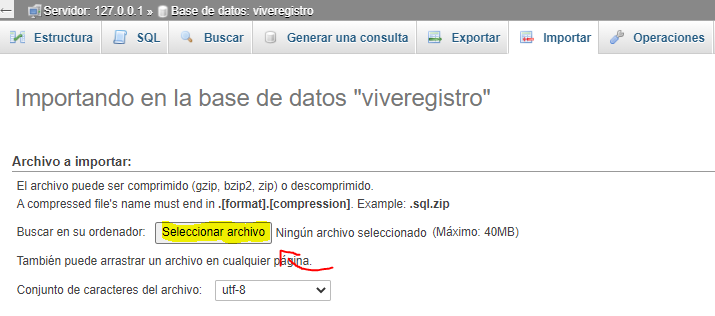


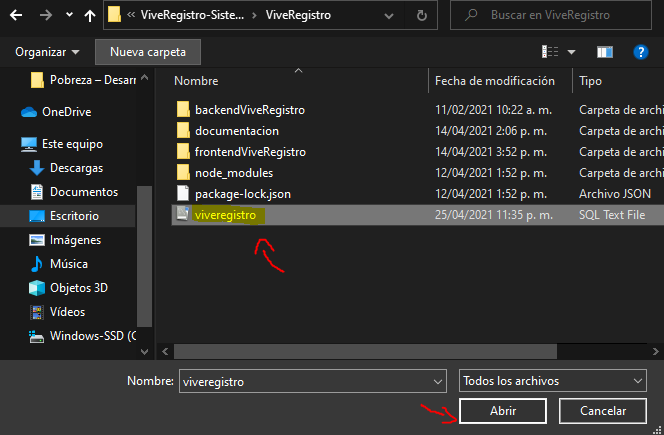
1. Activar los servicios de XAMPP y crear la base de datos en MySQL (ya debes de tener instalado MySQL y XAMPP, recuerda que los links de instalación se encuentran al final del documento):
   1. Ve al buscador de Windows o del sistema operativo que estés usando, y escribe “XAMPP”, te aparecerán los resultados de búsqueda y debes abrir la aplicación.
   2.  Después debes activar los servicios de MySQL y Apache, lo haces apretando los dos botones que están resaltados en amarillo, así como lo verás a continuación:

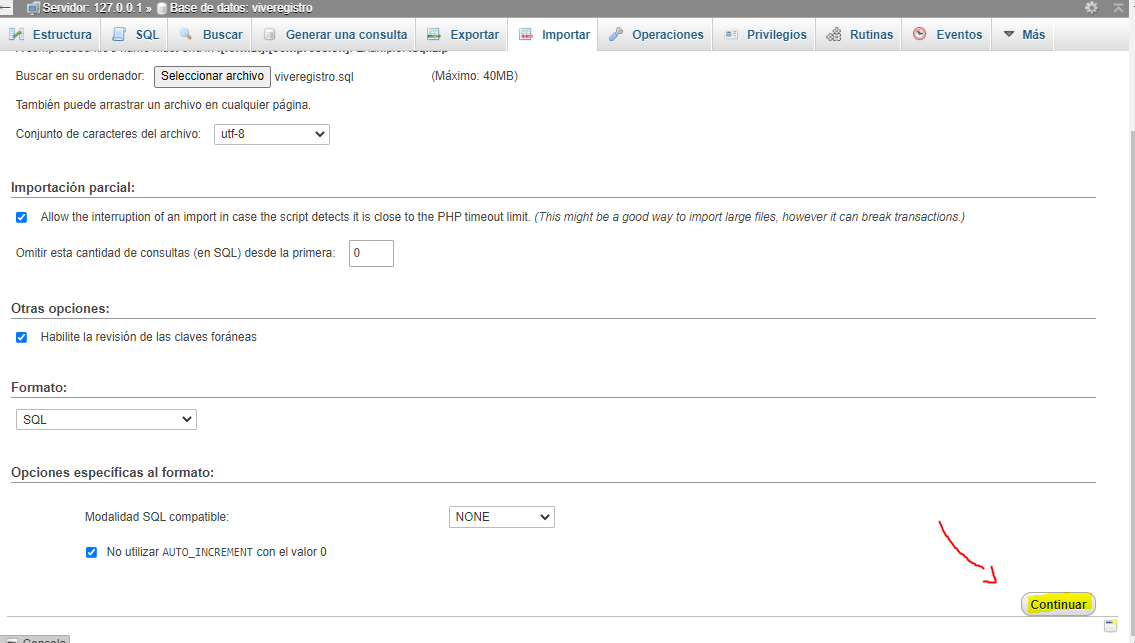
* 1.  Una vez activados los servicios, los botones deben de aparecer con un valor de “stop”, y al lado de cada botón debe aparecer el puerto por el cual está corriendo, así como lo veras a continuación:
  2.  Ahora procederemos a crear e importar la base de datos en MySQL, deberás ir primero al navegador (en este caso lo haremos en Google Chrome), después, en la barra de búsqueda, deberás ir a la siguiente dirección, tal cual cómo lo muestra la imagen:
  3.  En este punto deberás de estar en la siguiente interfaz, donde verás todo el centro de manejo para MySQL:
  4.  Ahora nos iremos al menú lateral izquierdo, donde le daremos en el botón que dice “Nuevo”:
  5. Te saldrá la interfaz que veras en la siguiente imagen, y luego, escribirás el **nombre** de la base de datos en el campo que se muestra ahí, seguido de esto, darás en el botón de “**crear**”:
  6. Después de crear la base de datos, importarás el script de la base de datos que se encuentra en un archivo “.SQL”, el cual a su vez estará ubicado en la carpeta de la documentación del proyecto, dentro de la carpeta “BasesDeDatos” con el nombre de “viveregistro”:

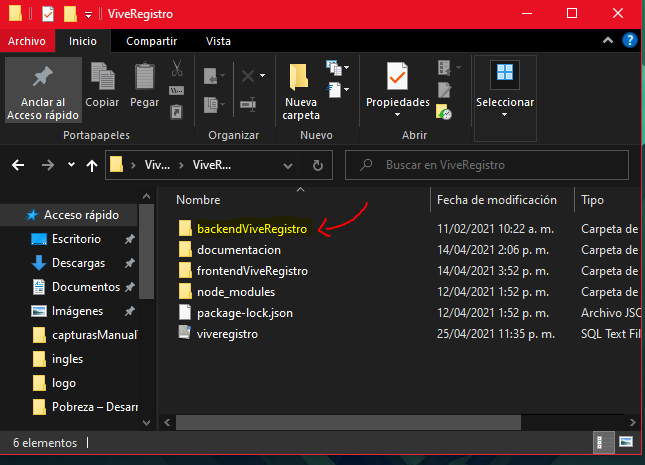


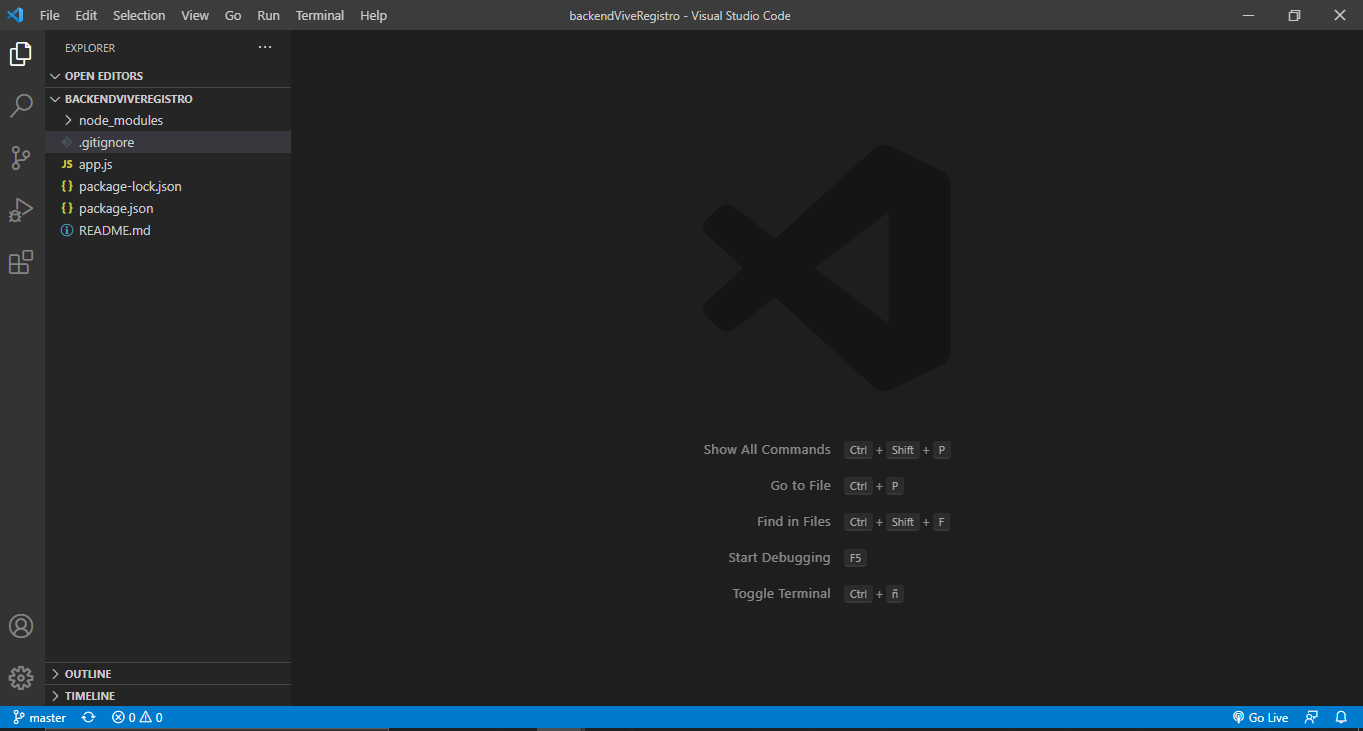


* 1. ¿Cómo importar el script?, fácil, en la interfaz de PhpMyAdmin, entras a la base de datos que creaste, y en el menú de arriba, accionarás el botón que tiene el nombre de “importar”:
  2. Ahora le darás en el botón de “Seleccionar archivo”, y buscaras el script de la base de datos (recuerda que se encuentra en la carpeta raíz del proyecto y tiene el nombre de “viveregistro.sql”), por último, le darás en el botón de “enviar”, el cual cargará el script y todo estará listo:

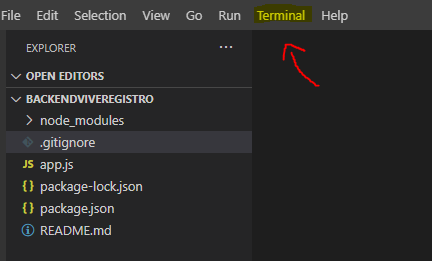




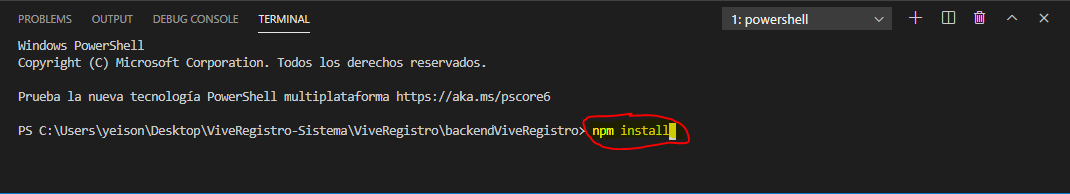
1. Ya con la base de datos creada en MySQL, estamos listos para aprender a ejecutar el backend del sistema.
   1.  Primero debemos abrir el proyecto del backed en un editor de código, aconsejamos usar Visual Studio Code, ya que es el que usaremos para la siguiente guía. Iremos a la carpeta raíz del proyecto y arrastraremos o abriremos la carpeta con el nombre de “backendViveRegistro” en el Visual Studio Code:

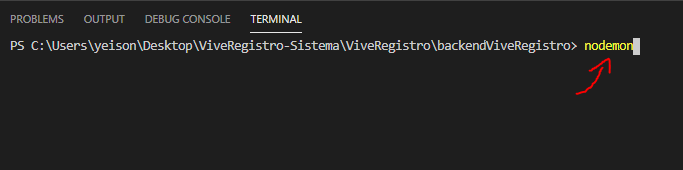
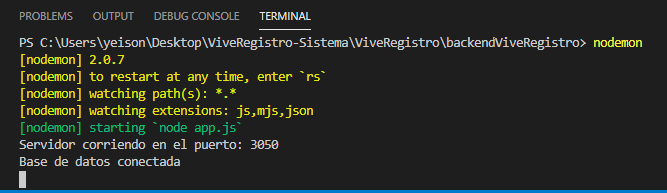


* 1. Una vez abierto el proyecto, procederemos a abrir la terminal integrada del Visual Studio Code (también puedes hacer todo este procedimiento desde el símbolo de sistema de Windows, pero esto ya es opcional), puedes usar el comando (Ctrl+shift+ñ), o también puedes ir a la parte del menú de arriba y presionar el botón que dice “terminal”, seguido de esto, dar en “new terminal”:

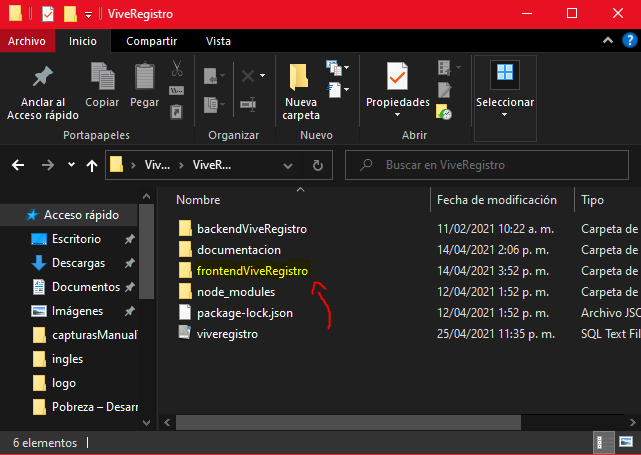


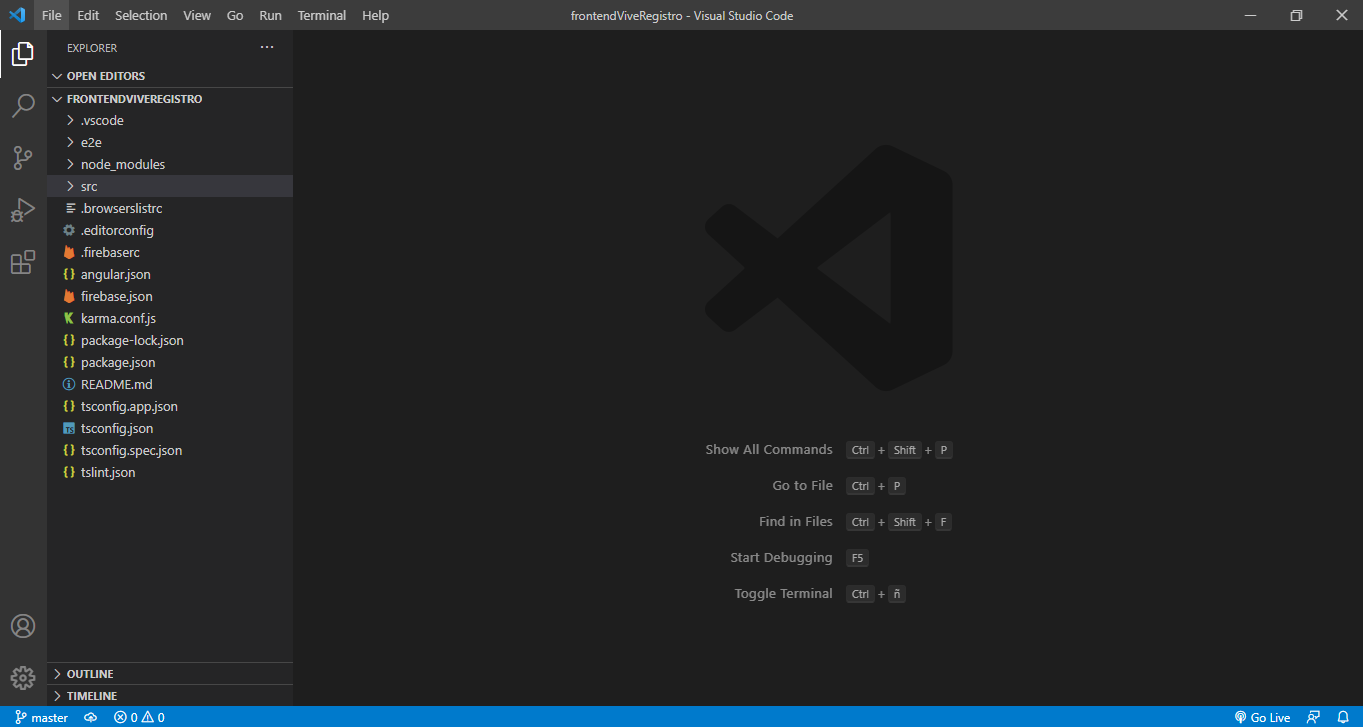
* 1. Una vez abierta la terminal integrada, deberás ingresar el siguiente comando “npm install”, así como se muestra en la siguiente imagen:



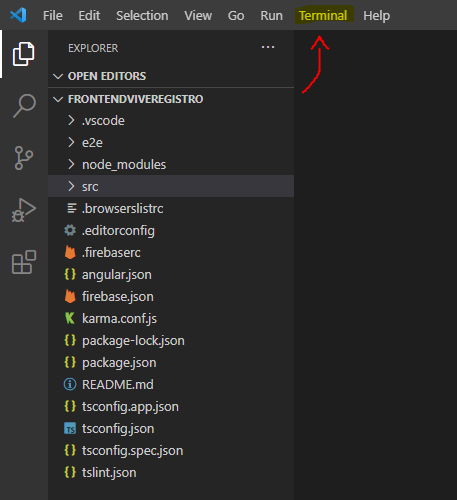
* 1. Luego le darás en la tecla “enter”, y se empezarán a cargar e instalar todas las dependencias del proyecto para que así, pueda funcionar.
  2.  Ahora, si no se ha presentado ningún fallo (lo cuál sería lo ideal), procederemos a escribir el comando “nodemon” en la terminal integrada (este comando se encargará de ejecutar todo el backend)
  3.  Ahora te debería de aparecer varios mensajes del servidor diciéndote que, todo ha ido bien y que la conexión con la base de datos fue exitosa:

1. Ahora, con las bases de datos y el backend funcionando, podemos ir al último paso que es: abrir y ejecutar el Frontend.
   1. Primero debemos abrir el proyecto del Frontend en un editor de código, como vimos antes, usaremos Visual Studio Code. Iremos a la carpeta raíz del proyecto y arrastraremos o abriremos la carpeta con el nombre de “frontendViveRegistro” en el Visual Studio Code:

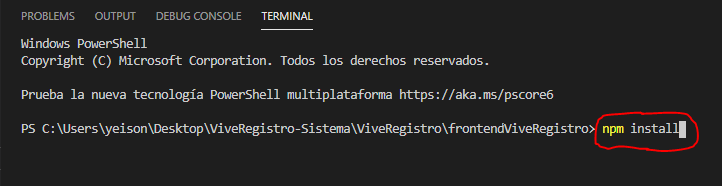


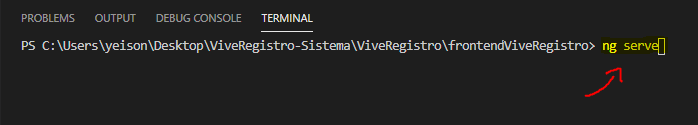


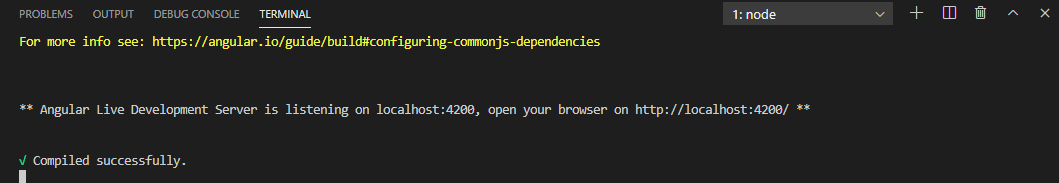
* 1. Una vez abierto el proyecto, procederemos a abrir la terminal integrada del Visual Studio Code (también puedes hacer todo este procedimiento desde el símbolo de sistema de Windows, pero, esto ya es opcional), puedes usar el comando (Ctrl+shift+ñ), o también puedes ir a la parte del menú de arriba y presionar el botón que dice “terminal”, y luego en “new terminal”:



* 1. Una vez abierta la terminal integrada, deberás ingresar el siguiente comando “npm install”, así como se muestra en la siguiente imagen:



* 1.  Ahora, si no se ha presentado ningún fallo (lo cual sería lo ideal), procederemos a escribir el comando “ng serve” en la terminal integrada, este comando se encargará de ejecutar todo el FrontEnd (esto tardará un poco dependiendo de tu computador):
  2. Una vez termine de cargar, te deberá de aparecer un mensaje, diciéndote que la compilación fue exitosa:



* 1. Por último, puedes ir al puerto que te indica angular (por defecto siempre será el puerto: 4200), allí podrás visibilizar el proyecto y el home page del sitio, ve al navegador e ingresa esa dirección, en este punto, todo estará listo:

# Configuración del proyecto

En el caso de este sistema, no se requiere una preconfiguración, ya que de por si, todo el sistema ya viene configurado, además, para poder correr el proyecto, solo se requiere de pocos comandos que ya se vieron anteriormente.

# Análisis

**Entrevista:**

**Entrevistado(a):** Alexander Babativa.

**Cargo:** Profesional en bilingüismo del C.T.G.I.

**Ficha** (Si aplica): .

**Programa de formación:** No aplica.

**Preguntas:**

1. ¿Has visto o has hecho uso del vivero del SENA?  
   R: Honestamente no era consciente de que había un vivero en el SENA.
2. ¿Por qué es importante guardar el historial de un registro?  
   R: Es importante para mirar la trazabilidad de un proyecto y en base a eso tomar mejores decisiones en base a ello.
3. ¿Para qué sirven las estadísticas?  
   R: Sirven para medir comportamientos en las personas o recursos y en base a ello dar resultados.
4. ¿Consideras importante los viveros para plantas? ¿Sí, no, por qué?  
   R: Sí, por el tema de que hoy en día la comida está muy industrializada y manipulada y es bueno que empiecen a implementar estos proyectos para que la gente lleve una vida más saludable en el tema de cultivo de alimentos.
5. ¿De qué manera se pueden mejorar estos viveros?  
   R: Divulgando información o conocimiento acerca de estos viveros para que la gente se concientice sobre su cuidado y uso.
6. ¿Qué opinas sobre la automatización de información?  
   R: Es importante porque agiliza procesos y evita reprocesos y al final permite analizar esta información
7. ¿Consideras importante cultivar tus propios alimentos? ¿Sí, no, por qué?  
   R: Absolutamente y además me gustaría aprender más sobre ese tema y también podría generar un ahorro de dinero.
8. ¿Tendrías tu propio vivero para cultivar algunas semillas? ¿Sí, no, por qué?  
   R: Sí lo tendría y sería chévere que se expandiera a nivel de apartamentos, o sea cultivar mini-viveros.
9. ¿Por qué crees que es importante la recolección de datos?  
   R: Es importante porque te permite tener un repositorio de datos para que luego otro grupo de personas hagan uso de estos datos.
10. ¿Planearías tener un vivero en tu hogar?

R: off course, me encantaría tener un vivero en mi casa a futuro o a corto plazo si me dan los conocimientos acerca de ello.

**Entrevistado(a):** Ana Isabel Casas.

**Cargo**: Productora multimedia independiente.

**Ficha** (Si aplica): 1354812.

**Programa de Formación:** Producción de multimedia.

1. ¿Has visto o has hecho uso del vivero del SENA?  
   R: Si lo he visto y solo le he tomado fotos para trabajos.
2. ¿Por qué es importante guardar el historial de un registro?  
   R: porque hay momentos en los que uno requiere de cierta información y con esto, puede agilizar el proceso de búsqueda.
3. ¿Para qué sirven las estadísticas?  
   R: Para llevar un control y mejorar procesos y resultados.
4. ¿Consideras importante los viveros para plantas? ¿Sí, no, por qué?  
   R: Sí, porque se requiere más espacio donde se pueda cultivar y cuidar plantas y además, se necesita de espacios donde las plantas puedan tener un mejor cuidado.
5. ¿De qué manera se pueden mejorar estos viveros?  
   R: Poniendo el sistema de riego, iluminación, un mayor conocimiento para cuidarlos y sistematizarlo o automatizarlo.
6. ¿Qué opinas sobre la automatización de información?  
   R: Es una gran idea, ya que esto les facilita trabajo a las personas y nos permite mantener un control sobre la información.
7. ¿Consideras importante cultivar tus propios alimentos? ¿Sí, no, por qué?  
   R: Honestamente si porque es muy saludable y llevas un mejor control en lo que comer ya que en el mercado hay muchos productos que se manipula mucho y se le echan cosas que no se debería con tal de mejorar su imagen.
8. ¿Tendrías tu propio vivero para cultivar algunas semillas? ¿Sí, no, por qué?  
   R: Yo no, ya que soy muy descuidada, al menos de que estuviera sistematizada.
9. ¿Por qué crees que es importante la recolección de datos?  
   R: Me sirven para hacer cambios, basados en una información y para así, generar mejores conclusiones.
10. ¿Planearías tener un vivero en tu hogar?

R: Si, en un futuro donde tenga más tiempo para cuidar de él y tenga los conocimientos necesarios.

**Conclusiones de la entrevista:**

-Encontramos que las personas son conscientes de que los productos son industrializados y manipulados químicamente y por ende, esto causa efectos secundarios tanto en los productos como en las personas consumidoras, ya que los productos no son 100% natural y a esto se quiere llegar, una forma natural de cultivar y una forma de consumirlo con naturalidad sin alteraciones.

-Se encontró que, hay cierta desinformación o desconocimiento acerca del vivero del SENA, lo cual nos indica que, hay que hacer una retroalimentación o dar información de una forma masificada sobre este tema a todas las personas que conforman el Sena complejo norte.

**Tipo de población:**

Todo tipo de persona que esté interesado con la agricultura o cuidado del medio ambiente.

**Encuesta:**

Se hizo una entrevista a las personas de la comunidad SENA con el fin de recaudar datos e información para luego realizar una encuesta la cual nos permitiría ver la pertinencia del proyecto académico.

**Preguntas realizadas:**

Esta encuesta emplea preguntas cerradas que tienen como opción de respuesta Sí/No y una de valoración del 1 al 10 sabiendo que 1 es menor importancia y 10 es mayor importancia.

1. ¿Has visto o tienes conocimiento del vivero que está ubicado en el SENA?

2. ¿Le ves utilidad al vivero del SENA?

3. ¿Cuál crees que es la causa de que no todos tienen conocimiento sobre el vivero del SENA?

4. ¿Crees que es importante guardar el registro histórico de los procesos de un sistema?

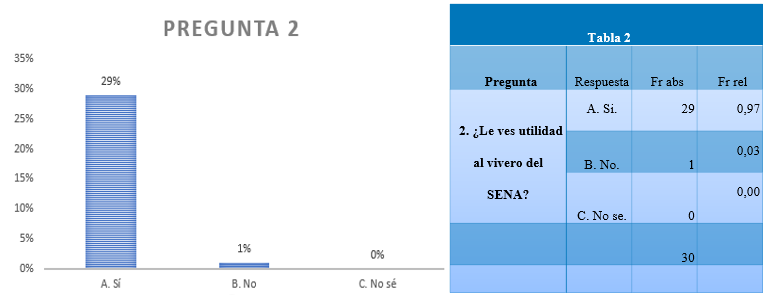
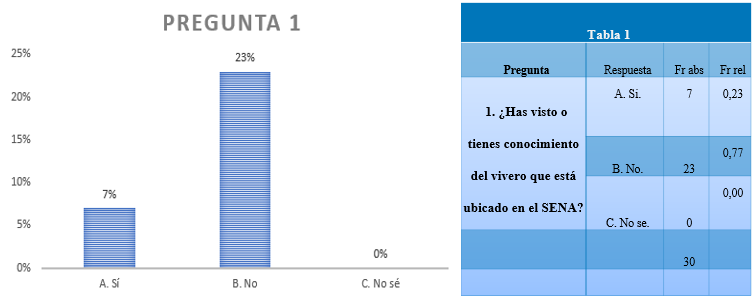
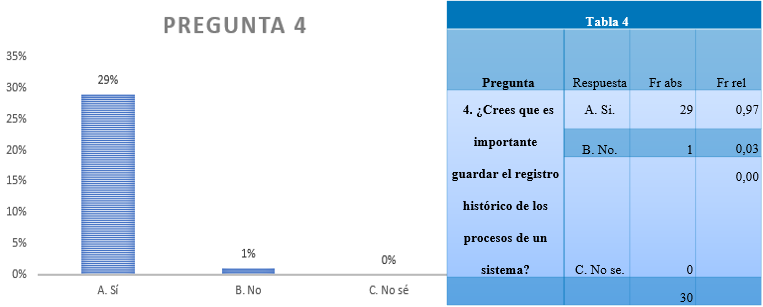
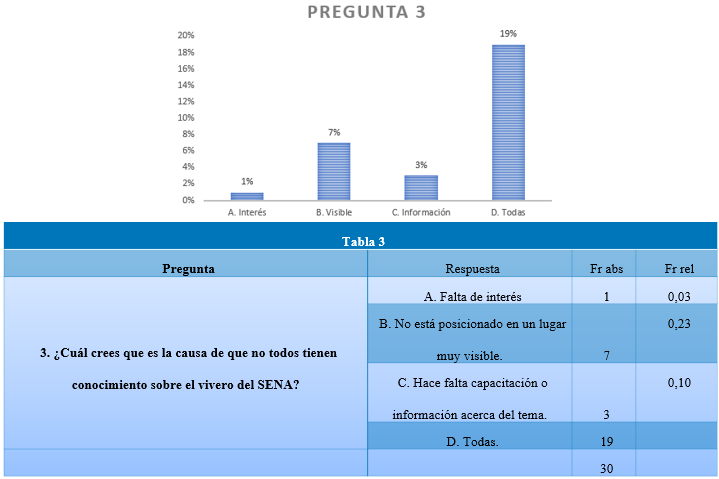
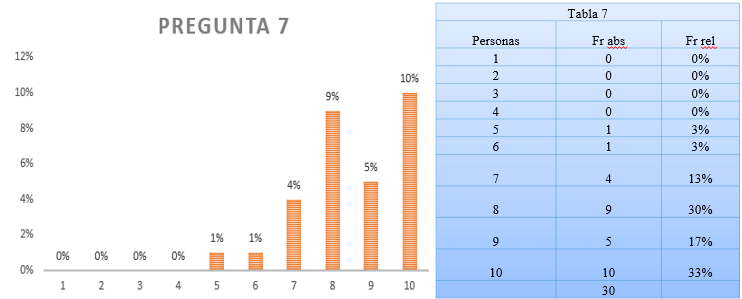
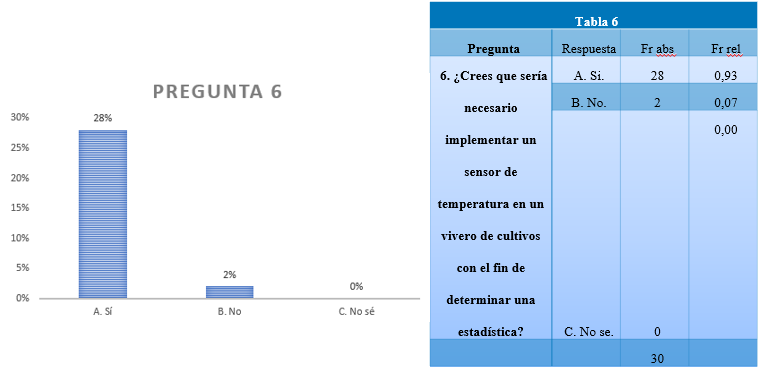
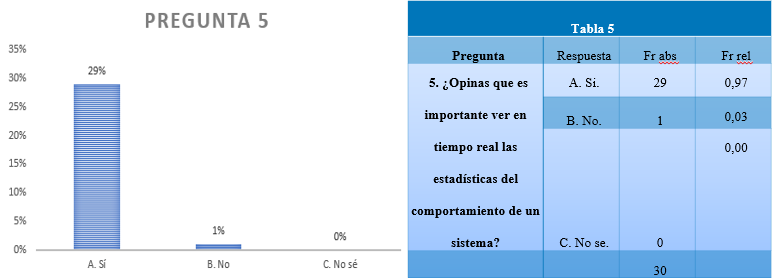
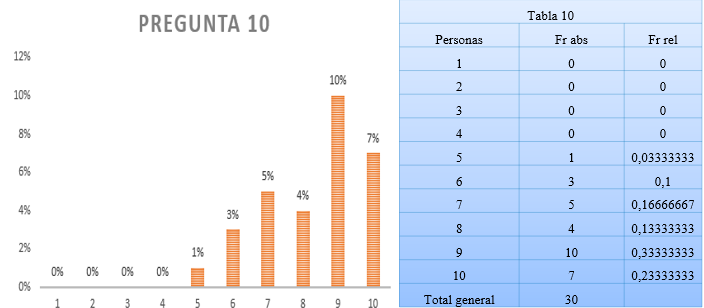
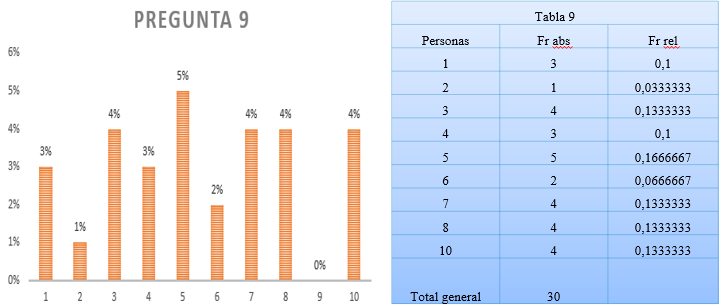
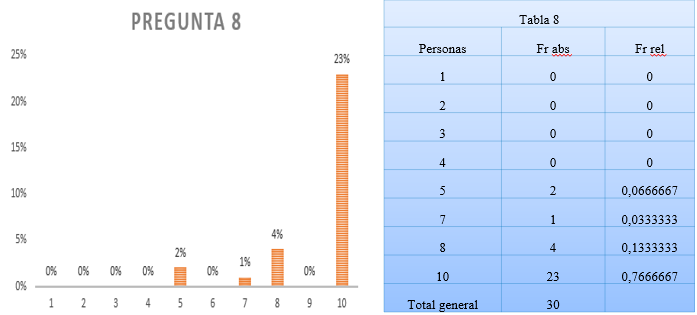
5. ¿Opinas que es importante ver en tiempo real las estadísticas del comportamiento de un sistema?

6. ¿Crees que sería necesario implementar un sensor de temperatura en un vivero de cultivos con el fin de determinar una estadística?  
7. ¿Del 1 al 10 qué tan importante consideras que existan viveros para cultivar alimentos?

8. ¿Del 1 al 10 qué tan importante consideras que la humanidad comience a cultivar en sus propios hogares?

9. ¿Del 1 al 10 qué tan capacitado te sientes para mantener en tu vivienda un vivero acondicionado al cultivo de alimentos?

10. ¿Del 1 al 10 qué calidad de vida crees que tendrías al implementar un vivero para alimentos en tu localidad?

**Resultados:  
  
  
  
  
  
  
**

**Conclusión de los datos recolectados:**

Gracias a la entrevista y a la encuesta, se puede observar que en realidad el proyecto es viable y sí es de interés para la población del Sena, y, además, se notó que, la principal causa de que el vivero no sea tan conocido, es principalmente por el lugar donde está posicionado y porque no se ha hecho una alimentación sobre el tema.

**Matriz FODA:**



# Diseño de la arquitectura

## Fichas Técnicas

En esta sección se mostrarán las fichas técnicas, donde se ilustra de forma gráfica y se explican las diferentes interfaces que contiene el sistema (además de otras especificaciones).



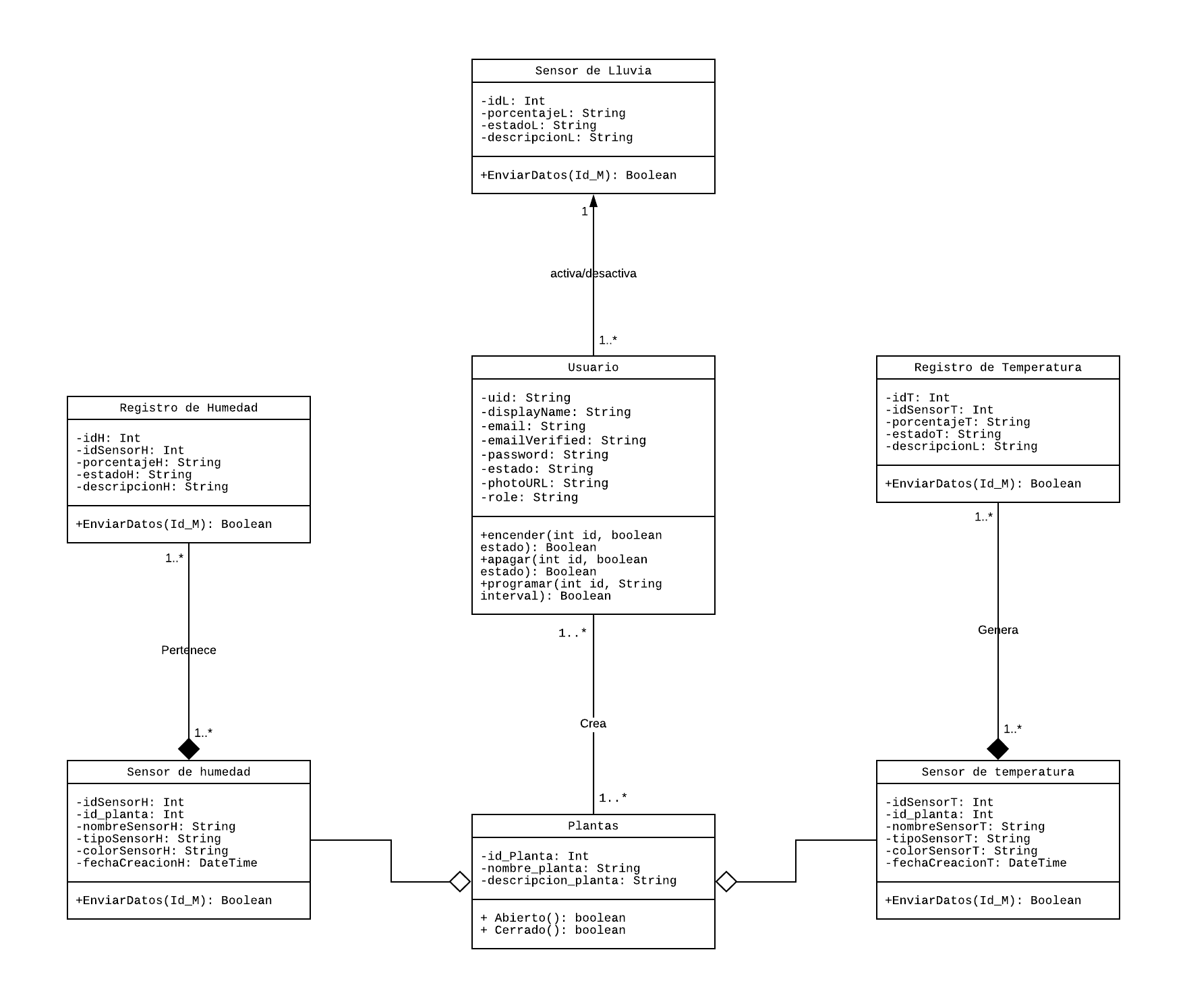


## Diagrama Casos de uso

En esta sección mostraremos el diagrama de casos de uso, donde gráficamente se podrá reconocer y entender los diferentes actores que participan e interactúan con el sistema, más específicamente, mostrar las acciones que hará cada actor dependiendo de su rol.

## 

## Diagrama de clases

En esta sección se mostrarán las diferentes clases que existen en el sistema, donde por medio de diferentes tipos de relaciones, se mostrará e ilustrarán la interacción que tienen estas entre sí.

## Diccionario de datos

En esta sección se mostrarán cada una de las entidades o tablas que existen en la base de datos del sistema, donde se podrá ver cada uno de los campos con su respectiva descripción, con el fin de dar una información más especifica sobre cada una de estas.



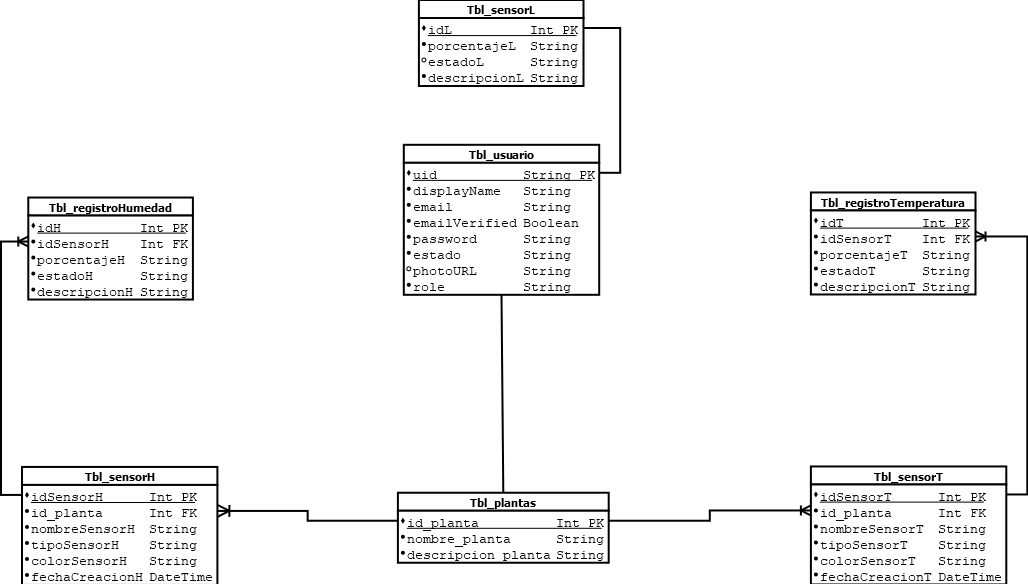






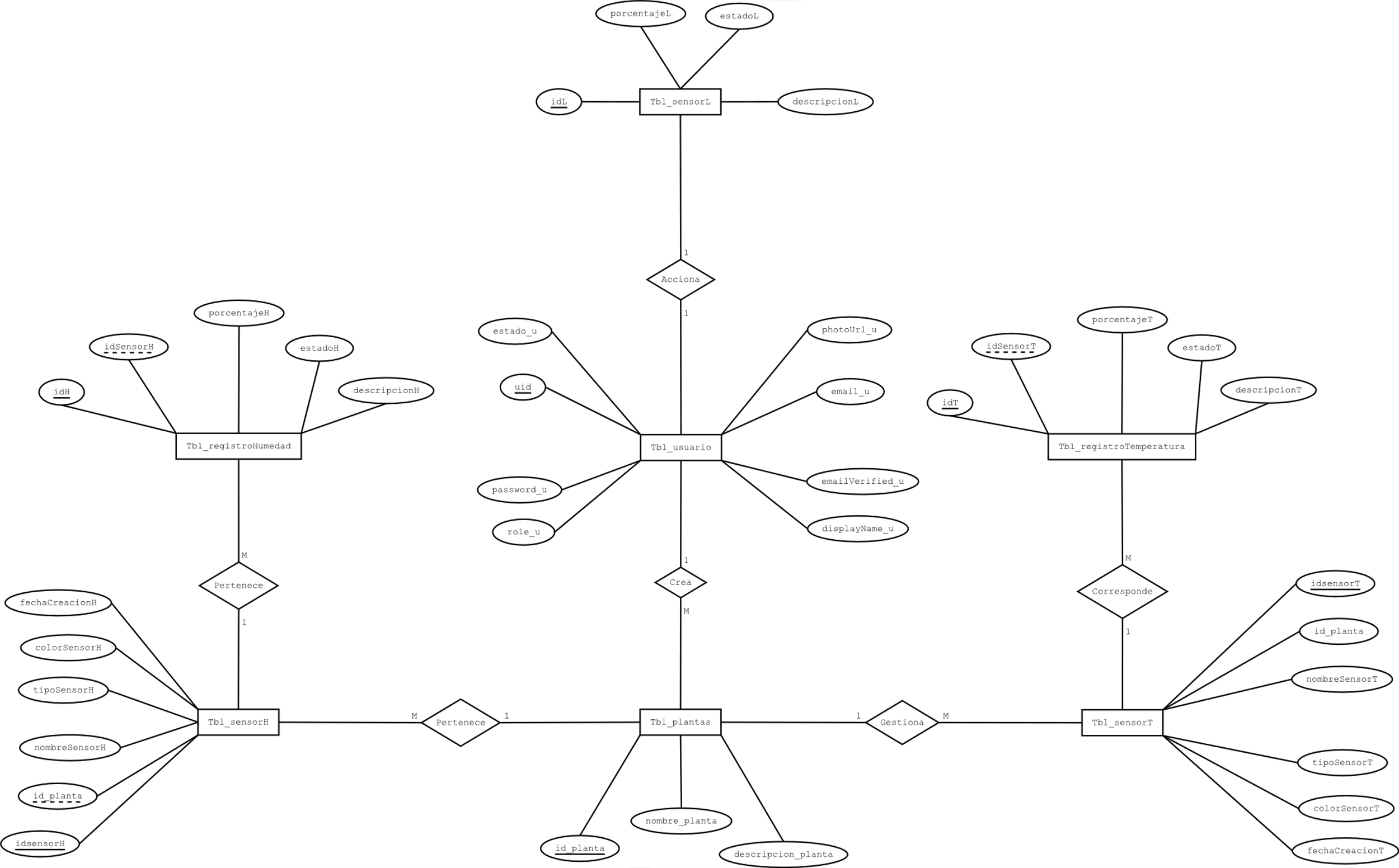
## Modelo relacional.

En esta sección se mostrarán cada una de las tablas que pertenecen a la base de datos del sistema, cuyo fin es poder tener un panorama más acercado sobre la base de datos y el cómo se organiza esquemáticamente cada tabla dentro de ella.



## Modelo entidad relación

En esta sección se mostrarán cada una de las entidades de las bases de datos junto a sus respectivos atributos, donde por medio de tipos de relaciones, se tendrá una vista más objetiva sobre las diferentes entidades que participan en la base de datos.



# Links de instalación

Se recomienda seguir el orden de instalación planteado, en caso de tener

problemas, acudir al administrador o a los videos recomendados).

* 1. Git: <https://git-scm.com/downloads>
* Video ayuda: <https://www.youtube.com/watch?v=h9ZH2wFpSUc>
  1. XAMPP: <https://www.apachefriends.org/es/download.html>
* Video ayuda: <https://www.youtube.com/watch?v=DOZPG4V6-JU>
  1. Visual Studio Code: <https://code.visualstudio.com/download>
* Video ayuda: <https://www.youtube.com/watch?v=EZAbKNLkrSM>
  1. NodeJs: <https://nodejs.org/es/download/> (Se recomienda descargar la versión LTS)
* Video ayuda: <https://www.youtube.com/watch?v=BgtB31gXkoA>
  1. Angular: <https://desarrolloweb.com/articulos/angular-cli.html> (recuerda primero descargar NodeJs y seguir puntualmente las instrucciones, además, se recomienda usar desde angular 10 hacía arriba)
* Video ayuda: <https://www.youtube.com/watch?v=8UjLKoIr5Fg>
  1. Arduino IDE: <https://www.arduino.cc/en/software> (Se recomienda usar cualquier versión desde la 1.8.0 en adelante)
* Video ayuda: <https://www.youtube.com/watch?v=Hz1rLRyFqf4>