

# MANUAL DE DESARROLLO

El Guadalquivir nos Une.



Plataforma Interinstitucional de  
la Cuenca del río Guadalquivir



ESTADO PLURINACIONAL DE  
**BOLIVIA**

MINISTERIO DE  
MEDIO AMBIENTE Y AGUA



Gobierno Autónomo  
Departamental de  
**TARIJA**



# SISTEMA MONITOREO PDCG

## MANUAL DE DESARROLLO

### PRIMERA VERSIÓN

Tarija – Bolivia - septiembre 2023

Este documento ha sido elaborado con asesoría técnica de  
PROCUENCA GIZ - Akut Partner

## AUTORIDADES:

LIC. OSCAR MONTES BARZÓN

Gobernador del Departamento de Tarija

ABG. ALAN ECHART SOSSA

Director del Servicio Departamental de Gestión Integral del Agua (SEDEGIA)

## SERVICIO DEPARTAMENTAL DE GESTIÓN INTEGRAL DEL AGUA (SEDEGIA)

Av. Panamericana en instalaciones del PERTT.

Teléfono: 66-45098

Fax (591) (4) 66-45098

Correo electrónico: sedegia.tja@gmail.com

Tarija – Bolivia

## PUBLICACIÓN PARA CITAR COMO:

“DOCUMENTO TECNICO SISTEMA DE MONITOREO PDCG”

Se autoriza la reproducción parcial del presente documento citando previamente la fuente a la que corresponde.

## CON EL APOYO DE LA GIZ

Elaborador por:

Diagramas UML:

Luis Felipe Nuñez Humana

Carolina Andrea Llanos Rocha

Estudiantes de UCB - Pasantes

Revisado por:

Ing. Mario Veizaga

PROCUECA GIZ

Fotografía:

© Archivo PROCUECA GIZ Bolivia

## INDICE

<b>FIGURA.....</b>	<b>6</b>
<b>1 INTRODUCCIÓN.....</b>	<b>7</b>
1.1 OBJETIVO DEL SISTEMA DE MONITOREO PDCG.....	7
1.2 OBJETIVO DEL MANUAL DE DESARROLLO .....	7
<b>2 CARACTERÍSTICAS DEL SISTEMA Y FUNCIONALIDADES.....</b>	<b>7</b>
1. 2.2 FUNCIONALIDADES .....	7
<b>3 REQUISITOS PREVIOS .....</b>	<b>8</b>
3.1 REQUISITOS MÍNIMOS .....	8
3.1.1 <i>Hardware</i> .....	8
3.1.2 <i>Software</i> .....	8
3.1.3 <i>Conexión a internet</i> .....	8
3.2 HERRAMIENTAS .....	8
3.2.1 <i>Visual Studio Code</i> .....	8
3.3 INSTALACIÓN DE HERRAMIENTAS DE DESARROLLO .....	9
3.3.1 <i>Node.JS</i> .....	9
3.3.2 <i>Angular</i> .....	9
3.3.3 <i>MySQL Workbench</i> .....	10
3.3.4 <i>MariaDB</i> .....	12
<b>4 ARQUITECTURA DEL SISTEMA .....</b>	<b>13</b>
4.1 DESCRIPCIÓN GENERAL .....	13
4.2 DIAGRAMA DE ARQUITECTURA .....	13
<b>5 BASE DE DATOS.....</b>	<b>14</b>
5.1 MYSQL .....	14
5.1.1 <i>Descripción general</i> .....	14
5.2 DISEÑO DE LA BASE DE DATOS .....	14
<b>6 BACKEND .....</b>	<b>15</b>
6.1 GUÍA PARA LEVANTAR PARA EL BACKEND.....	15
6.2 TECNOLOGÍAS/DEPENDENCIAS UTILIZADAS .....	15
6.3 ESTRUCTURA DEL PROYECTO .....	17
6.4 CONEXIÓN A LA BASE DE DATOS .....	17
6.4.1 <i>Gestión de conexión y desconexión</i> .....	19
6.4.2 <i>Uso en MySQL</i> .....	19
<b>7 FRONTEND .....</b>	<b>20</b>
7.1 GUÍA PARA LEVANTAR EL FRONTEND .....	20
7.2 TECNOLOGÍAS/DEPENDENCIAS UTILIZADAS .....	20
7.3 ESTRUCTURA DEL PROYECTO .....	21

**2. 8 CONCLUSIONES Y RECOMENDACIONES.....¡ERROR! MARCADOR NO DEFINIDO.**

1. 8.1 CONCLUSIONES ..... **¡ERROR! MARCADOR NO DEFINIDO.**
2. 8.2 RECOMENDACIONES..... **¡ERROR! MARCADOR NO DEFINIDO.**

## FIGURA

Figura 1. Características del sistema monitoreo PDCG .....	7
Figura 2. Página principal Visual Studio Code.....	9
Figura 3. Instalación node.JS .....	9
Figura 4. Versión node .....	10
Figura 5. Instalación Angular CLI .....	10
Figura 6. MySQL Workbench .....	11
Figura 7. instalación MySQL Workbench .....	11
Figura 8. MySQL Workbench 8.0.34 .....	12
Figura 9. Versión de MariaDB .....	13
Figura 10. Descarga MariaDB.....	13
Figura 11. Diagrama de arquitectura del sistema .....	14
Figura 12. Modelo relacional.....	15
Figura 13. Dependencias del proyecto .....	16
Figura 14. Dependencias de desarrollo .....	17
Figura 15. Credenciales MySQL .....	18
Figura 16. Conexión y desconexión en la BD .....	19
Figura 17. Nueva conexión MySQL.....	19
Figura 18. Conexión con MySQL.....	20
Figura 19. Estructura del proyecto Angular.....	22
Figura 20. Carpeta 'src' .....	23
Figura 21. Carpeta app.....	23
Figura 22. Carpeta 'assets' .....	24
Figura 23. Carpeta 'environment' .....	25



# 1 INTRODUCCIÓN

## 1.1 OBJETIVO DEL SISTEMA DE MONITOREO PDCG

El objetivo del sistema de monitoreo Plan Director Cuenca del Guadalquivir es “Mejorar la seguridad hídrica de la cuenca del río Guadalquivir en un contexto de mayor incertidumbre climática”.

## 1.2 OBJETIVO DEL MANUAL DE DESARROLLO

El objetivo del manual es fortalecer el equipo de desarrollo con una información detallada y alineándose con las buenas prácticas, promoviendo así la calidad del código, además, se busca fomentar un ambiente de aprendizaje continuo y colaboración efectiva entre todos los miembros del equipo.

# 2 CARACTERÍSTICAS DEL SISTEMA Y FUNCIONALIDADES

El sistema presenta notables características, destacando su naturaleza de código abierto, lo que implica una escalabilidad y crecimiento progresivo sin ningún costo. Se distingue por su robusta seguridad, garantizando la protección de datos y la confidencialidad mediante medidas avanzadas.

La interfaz es fácil de usar, con una navegación amigable que facilita la experiencia del usuario y una visualización clara de datos. La capacidad de actualización en tiempo real, que incluye archivos, reportes y tablas, asegura que la información esté siempre al día.

Asimismo, se ofrece una documentación accesible al público en general, abarcando cada proyecto, normativa, actividad, así como planes, estrategias y programas. Además, el sistema permite la generación de reportes en varios formatos, como .png, .jpg, .pdf, .svg, .csv y .xls.



Figura 1. Características del sistema monitoreo PDCG

## 1. 2.2 FUNCIONALIDADES

- **Monitoreo Continuo:** El sistema permite el monitoreo constante y en tiempo real de diferentes iniciativas y proyectos, lo que brinda una visión actualizada de su progreso y rendimiento.

- Seguimiento Físico y Financiero: Ofrece la capacidad de realizar un seguimiento tanto físico como financiero de cada proyecto o acción, lo que incluye la supervisión de presupuestos, costos, avances y otros indicadores clave.
- Georreferenciación: El sistema utiliza herramientas de georreferenciación para localizar proyectos en un mapa interactivo, siendo esencial para identificar la ubicación exacta de las acciones/proyecto y facilitar su gestión.
- Disponibilidad: Se pone a disposición de la población en general la información y documentación de cada proyecto o acción así también reportes a medida, promoviendo la transparencia gubernamental y permitiendo que los ciudadanos accedan a datos clave de manera abierta y accesible.

### 3 REQUISITOS PREVIOS

#### 3.1 REQUISITOS MÍNIMOS

##### 3.1.1 *HARDWARE*

- ✓ Memoria RAM: 4GB de RAM
- ✓ Procesador: Procesador Intel de dos núcleos a 2,4 GHz o equivalente
- ✓ Disco Duro: 128 GB de espacio libre es suficiente

##### 3.1.2 *SOFTWARE*

- ✓ Sistema Operativo: Windows
- ✓ Navegador web: Navegador preferido (Recomendado: Chrome, Brave)
- ✓ Gestión Base de Datos: MySQL MariaDB versión superior a 10.0.0

##### 3.1.3 *CONEXIÓN A INTERNET*

- ✓ Se recomienda contar con una conexión a Internet de banda ancha para permitir una comunicación rápida y estable con usuarios y otros sistemas.

#### 3.2 HERRAMIENTAS

##### 3.2.1 *VISUAL STUDIO CODE*

Visual Studio Code es un editor de código fuente ligero pero eficaz que se ejecuta en el escritorio y está disponible para Windows, macOS y Linux. Incluye compatibilidad integrada con JavaScript, TypeScript y Node.js, y cuenta con un amplio ecosistema de extensiones para otros lenguajes (como C++, C#, Java, Python, Go, .NET). (Visual Studio Code, s.f.)



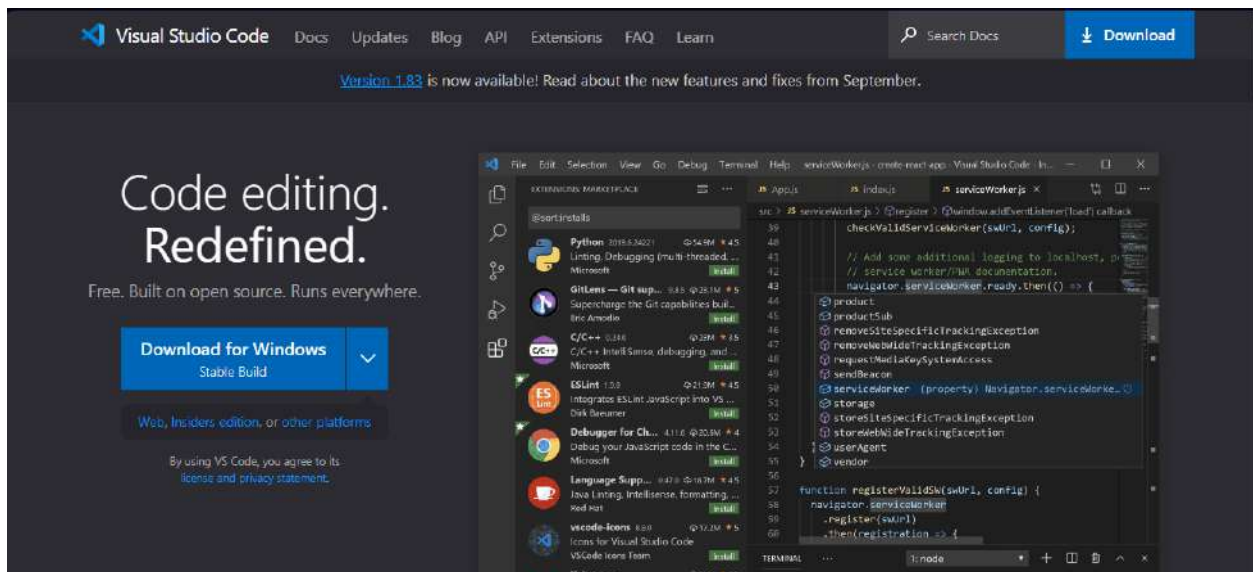


Figura 2. Página principal Visual Studio Code

### 3.3 INSTALACIÓN DE HERRAMIENTAS DE DESARROLLO

#### 3.3.1 NODE.JS

Es un entorno de ejecución para JavaScript, se utilizará una versión de 16.0.0 para adelante.

Puedes descargar la última versión estable desde su página web de Node.JS: <https://nodejs.org/es>

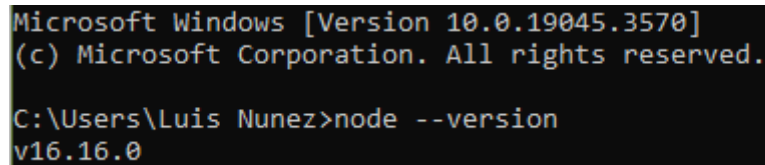


Figura 3. Instalación node.js

#### 3.3.2 ANGULAR

Angular es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones eficientes y sofisticadas de una sola página, se utilizará con una versión 15 o superior. (Angular, s.f.)

- ✓ Es necesario contar con Node.js previamente instalado, ya que Angular utiliza Node.js y npm para gestionar sus dependencias, ya que Angular utiliza Node.js y npm para gestionar sus dependencias. Puedes verificar la instalación ejecutando el siguiente comando en tu terminal:



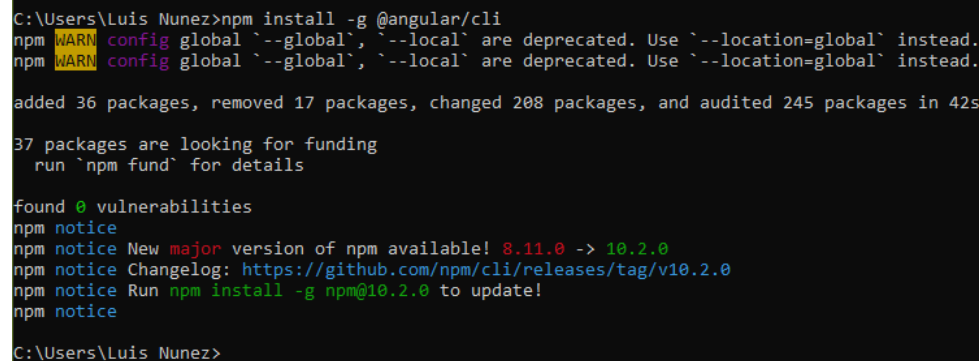
```
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Luis Nunez>node --version
v16.16.0
```

Figura 4. Versión node

- ✓ Posteriormente, instala Angular CLI (Command Line Interface), ejecutando el siguiente comando que asegurará una instalación global de Angular CLI en tu sistema.

```
npm install -g @angular/cli@15.0.0
```



```
C:\Users\Luis Nunez>npm install -g @angular/cli
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.
npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

added 36 packages, removed 17 packages, changed 208 packages, and audited 245 packages in 42s

37 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
npm notice
npm notice New major version of npm available! 8.11.0 -> 10.2.0
npm notice Changelog: https://github.com/npm/cli/releases/tag/v10.2.0
npm notice Run npm install -g npm@10.2.0 to update!
npm notice
C:\Users\Luis Nunez>
```

Figura 5. Instalación Angular CLI

### 3.3.3 MYSQL WORKBENCH

MySQL Workbench es una herramienta visual unificada para arquitectos de bases de datos, desarrolladores y administradores de bases de datos. MySQL Workbench proporciona modelado de datos, desarrollo SQL y herramientas de administración integrales para la configuración del servidor, la administración de usuarios, la copia de seguridad y mucho más. MySQL Workbench está disponible en Windows, Linux y Mac OS X (MySQL, s.f.)

A continuación, se demostrará el proceso de instalación haciendo el uso en la versión 8.0.34.

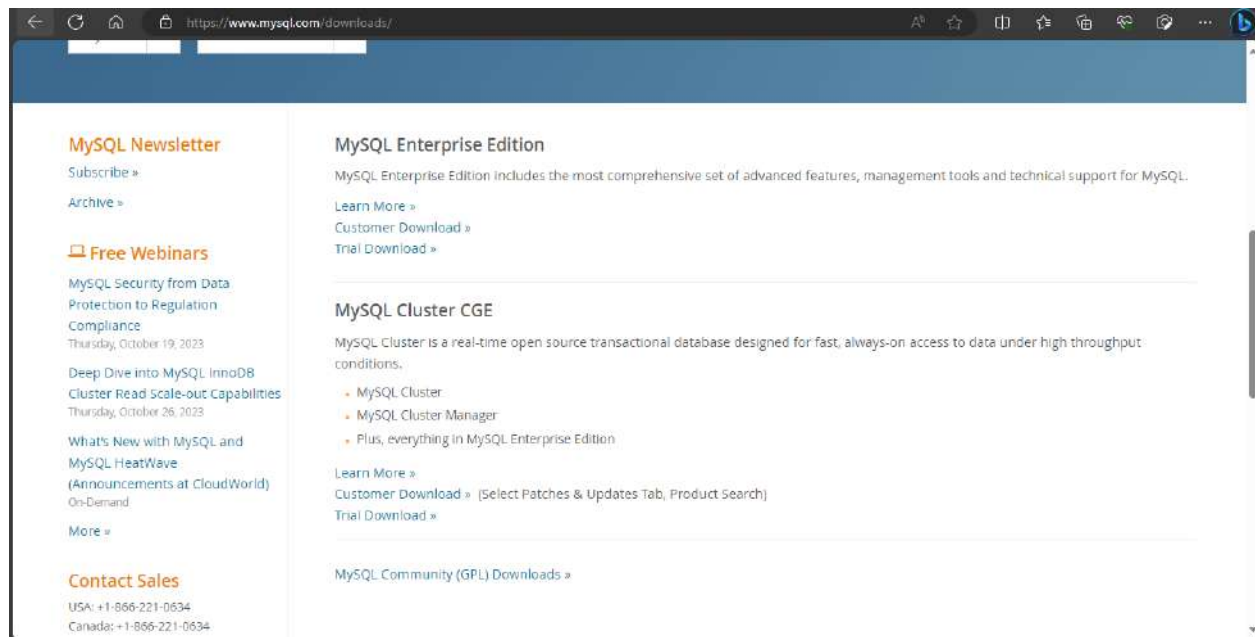


Figura 6. MySQL Workbench

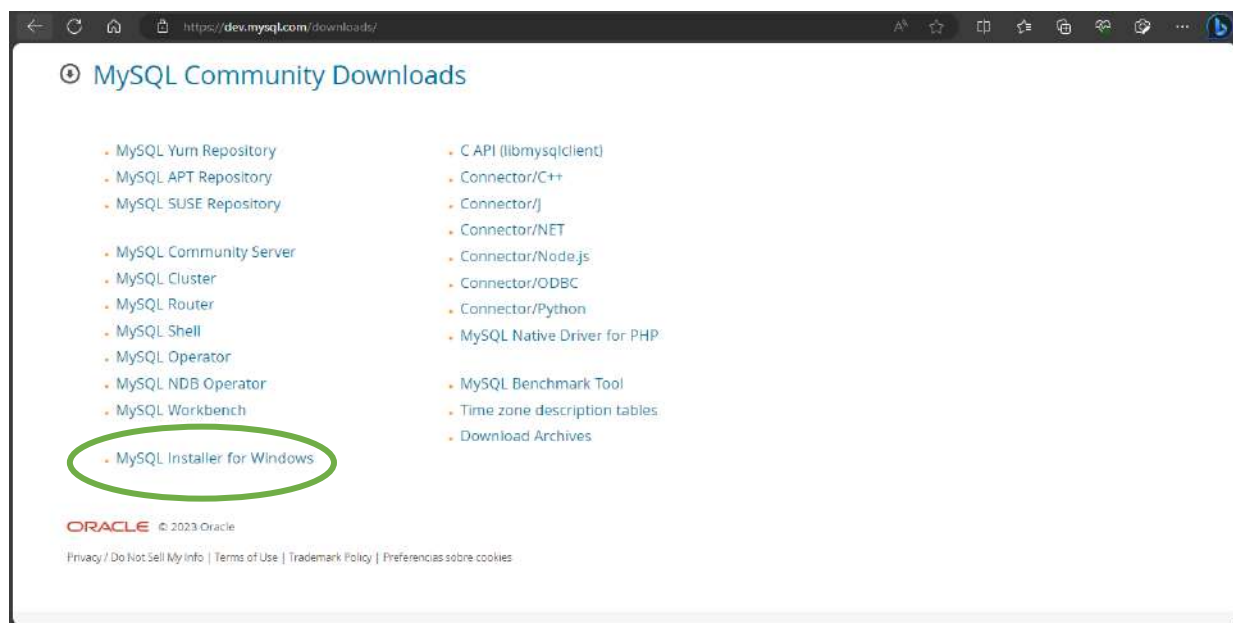


Figura 7. instalación MySQL Workbench

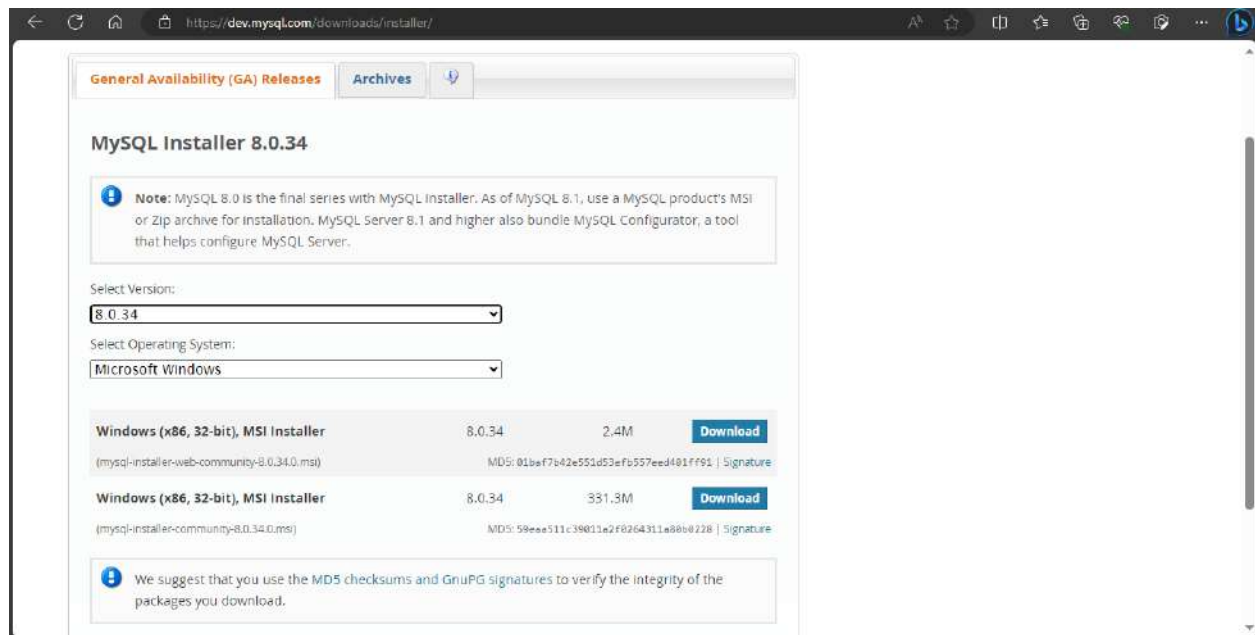


Figura 8. MySQL Workbench 8.0.34

### 3.3.4 MARIADB

MariaDB Community Server establece el estándar para las bases de datos relacionales de código abierto, con compatibilidad con Oracle Database (por ejemplo, secuencias y PL/SQL), tablas temporales, fragmentación transparente, cambios instantáneos de esquema, reversión a un momento dado y SQL moderno (es decir, expresiones de tabla comunes, funciones de ventana, funciones JSON y más). (MariaDB, s.f.)

A continuación, se demostrará el proceso de instalación haciendo el uso en la versión 10.4.31

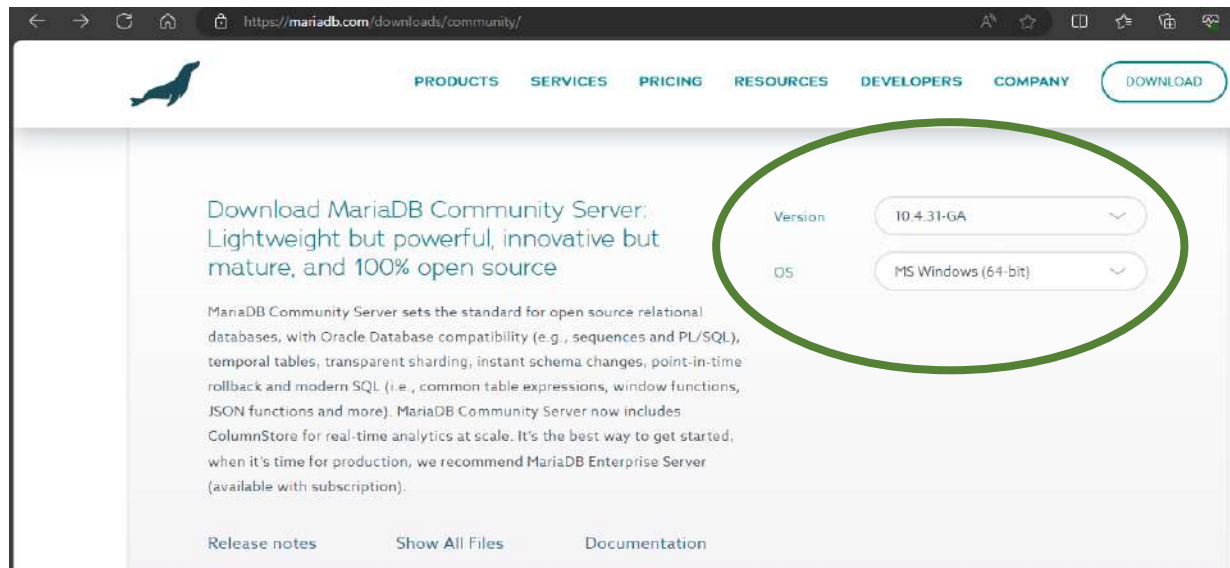


Figura 9. Versión de MariaDB

En la parte baja cuando terminas de seleccionar la versión se encuentra en botón 'Download'

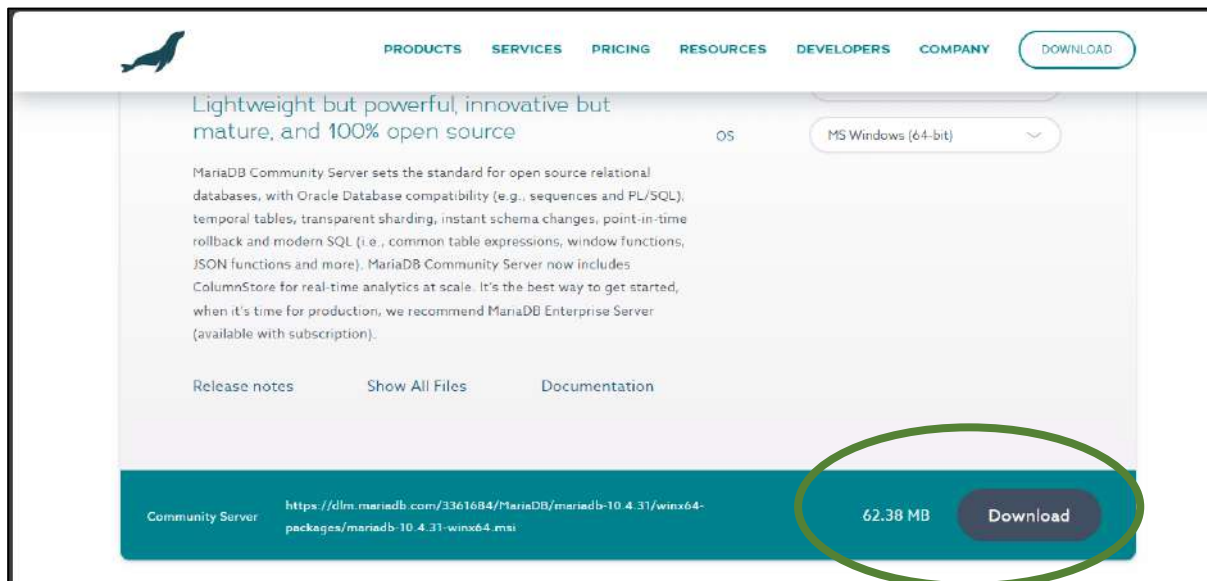


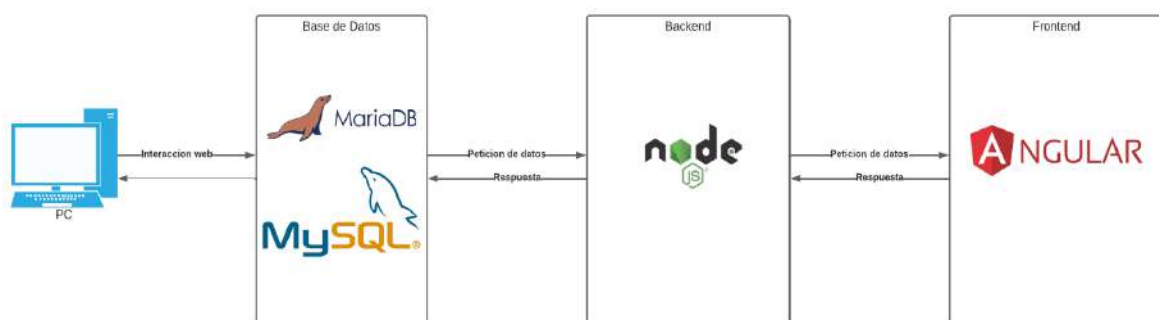
Figura 10. Descarga MariaDB

## 4 ARQUITECTURA DEL SISTEMA

### 4.1 DESCRIPCIÓN GENERAL

El sistema se basa en una arquitectura cliente-servidor, donde el cliente es una interfaz web y el servidor es responsable de procesar las solicitudes y almacenar los datos. El sistema utiliza tecnologías como HTML, CSS, TypeScript en el lado del cliente, y Nodejs y MySQL en el lado del servidor.

### 4.2 DIAGRAMA DE ARQUITECTURA





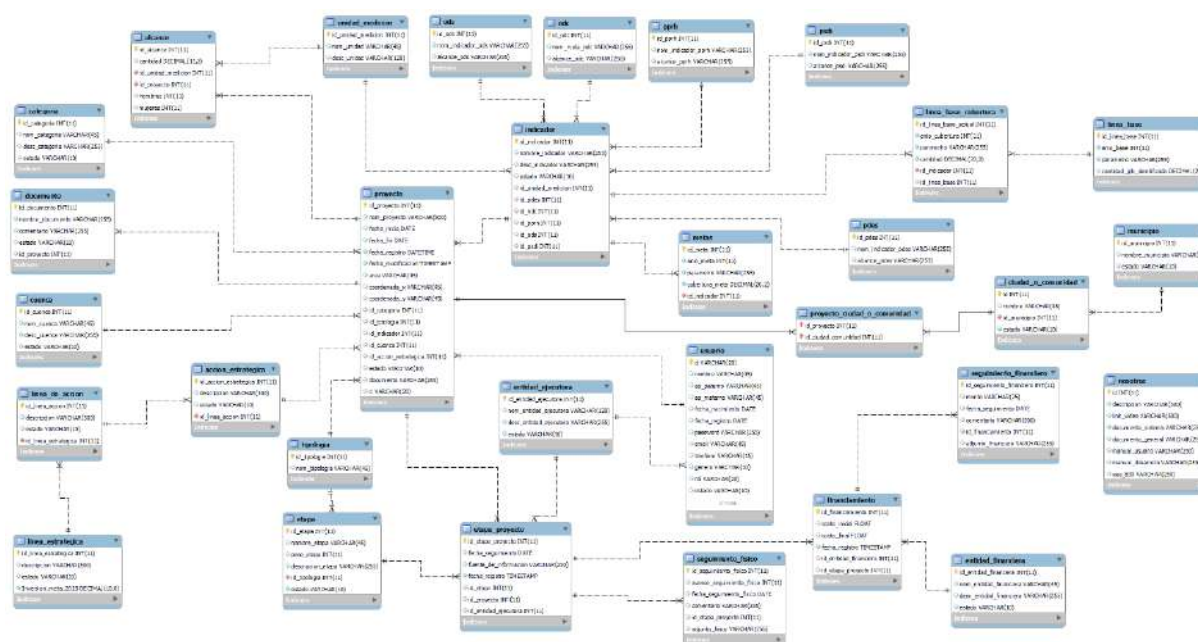
## 5 BASE DE DATOS

## 5.1 MYSQL

### 5.1.1 DESCRIPCIÓN GENERAL

- ✓ Utiliza el lenguaje SQL (Structured Query Language) para realizar operaciones en la base de datos, como la creación, modificación y consulta en los datos.
- ✓ MySQL es compatible con diferentes sistemas operativos puede ejecutarse en Windows, Linux, MacOS.
- ✓ MySQL utiliza el modelo de base de datos relacional, esto significa que organiza los datos en tablas relacionadas entre sí. Cada tabla se compone de filas y columnas, y las relaciones entre ellas se establecen mediante claves primarias y foráneas.
- ✓ Ofrece características robustas de seguridad, incluyendo autenticación basada en contraseñas y permisos.
- ✓ Ofrece características robustas de seguridad, incluyendo autenticación basada en contraseñas y permisos.

## 5.2 DISEÑO DE LA BASE DE DATOS





*Figura 12. Modelo relacional*

## 6 BACKEND

### 6.1 GUÍA PARA LEVANTAR PARA EL BACKEND

Pasos para levantar el Backend (Node.js)

- Abre una terminal en la ubicación del directorio del backend.
- Ejecuta ``npm install`` para instalar las dependencias del proyecto.
- Asegúrate de que la base de datos MySQL/MariaDB esté en funcionamiento con las credenciales adecuadas.
- Modifica la configuración de la base de datos en el código del backend para que coincida con tus credenciales de MySQL/MariaDB.
- Ejecuta ``node app.js`` o el comando correspondiente para iniciar el servidor backend. Por defecto, estará disponible en ``http://localhost:3000``.

### 6.2 TECNOLOGÍAS/DEPENDENCIAS UTILIZADAS

Las tecnologías que se ha utilizado:

- Nodejs para un entorno de ejecución para JavaScript al lado del servidor.
- El framework: NodeJS 'Express' que facilita la construcción de las APIs.
- El lenguaje de programación para la parte de backend: JavaScript

Dependencias utilizadas en el proyecto

1. `cors : "^2.0.5"`

Propósito: Proporciona un Middleware para habilitar el manejo de CORS (Cross-Origin Resource Sharing)

Función: Permite que el servidor responda a solicitudes HTTP desde dominios diferentes al dominio del servidor.

2. `dotenv : "^16.3.1"`

Propósito: Cargar variables de entorno desde un archivo `'.env'` en el entorno de Node.js

Función: Almacenar configuraciones sensibles como claves API o información de conexión a la base de datos.

3. `jsonwebtoken : "^9.0.1"`

Propósito: Implementar una generación y verificación de JSON Web Tokens (JWT)

Función: Autenticar y autorizar en la aplicación web.

4. `multer : "^1.4.5-lts.1"`

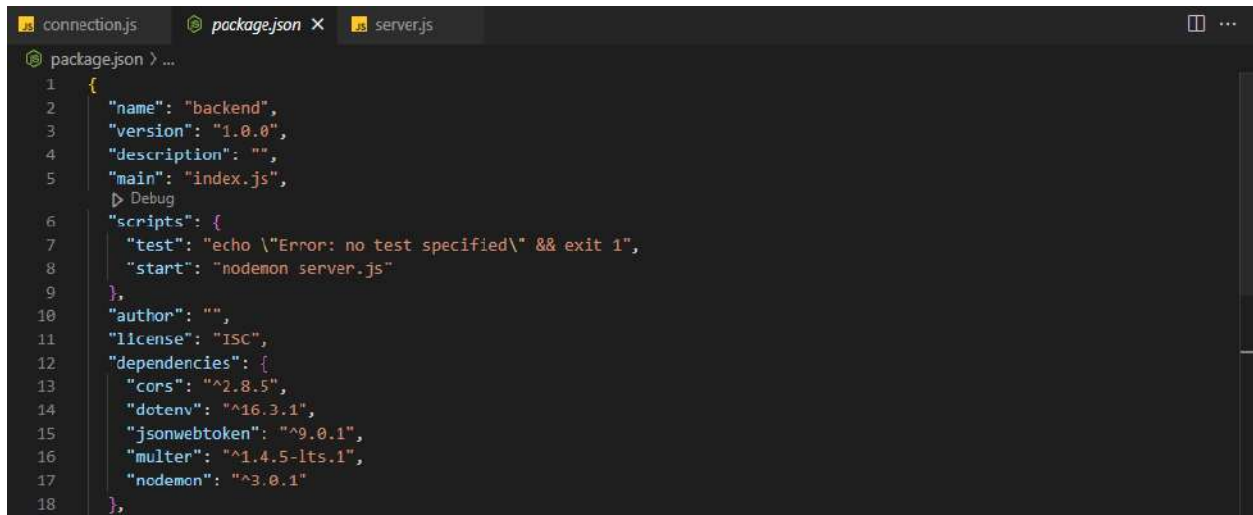
Propósito: Es un middleware para el manejo de archivos es formularios multipart/form-data

Función: Subida de archivos desde formularios en aplicaciones web

5. `nodemon : "^3.0.1"`

Propósito: Una herramienta que reinicia automáticamente la aplicación web cuando se detectan cambios en el código.

Función: Facilita el desarrollo al proporcionar un reinicio automático del servidor durante el desarrollo.



```
1 {
2   "name": "backend",
3   "version": "1.0.0",
4   "description": "",
5   "main": "index.js",
6   "scripts": {
7     "test": "echo \"Error: no test specified\" && exit 1",
8     "start": "nodemon server.js"
9   },
10  "author": "",
11  "license": "ISC",
12  "dependencies": {
13    "cors": "^2.8.5",
14    "dotenv": "^16.3.1",
15    "jsonwebtoken": "^9.0.1",
16    "multer": "^1.4.5-lts.1",
17    "nodemon": "^3.0.1"
18  },
19 }
```

Figura 13. Dependencias del proyecto

Dependencias de desarrollo:

1. chai : "^4.3.8"

Propósito: Biblioteca de garantías para pruebas en Node.js.

Función: Escribir pruebas más expresivas y legibles.

2. express : "^4.18.2"

Propósito: Framework web para Node.js.

Función: Desarrollo de la aplicación web y el desarrollo de las APIs.

3. mocha : "^10.2.0"

Propósito: Marco de prueba para Node.js.

Función: Ejecución de pruebas unitarias y de integración.

4. mysql : "^2.18.1"

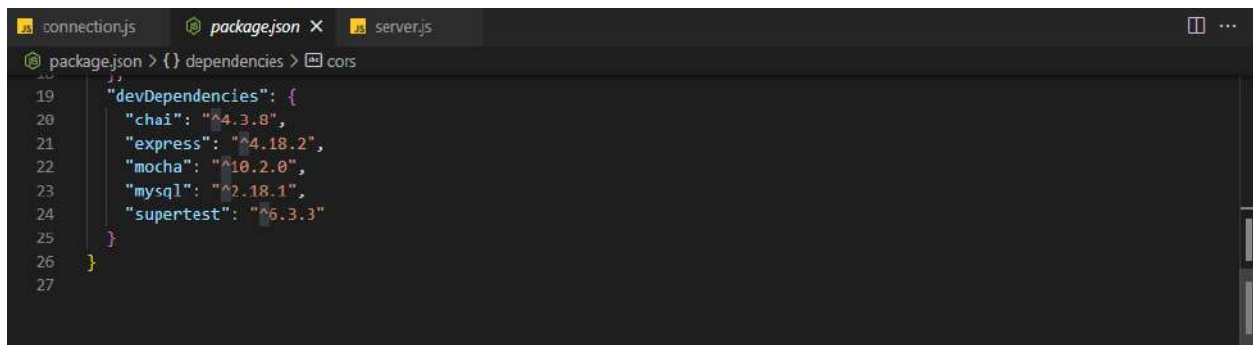
Propósito: Controlador MySQL para Node.js.

Función: Interactuar con bases de datos MySQL desde la aplicación.

5. supertest : "^6.3.3"

Propósito: Biblioteca para realizar pruebas HTTP en Node.js.

Función: Testing de API HTTP en conjunción con un marco de prueba como Mocha.



```
19  "devDependencies": {
20    "chai": "^4.3.8",
21    "express": "^4.18.2",
22    "mocha": "^10.2.0",
23    "mysql": "^2.18.1",
24    "supertest": "^6.3.3"
25  }
26 }
27
```

Figura 14. Dependencias de desarrollo

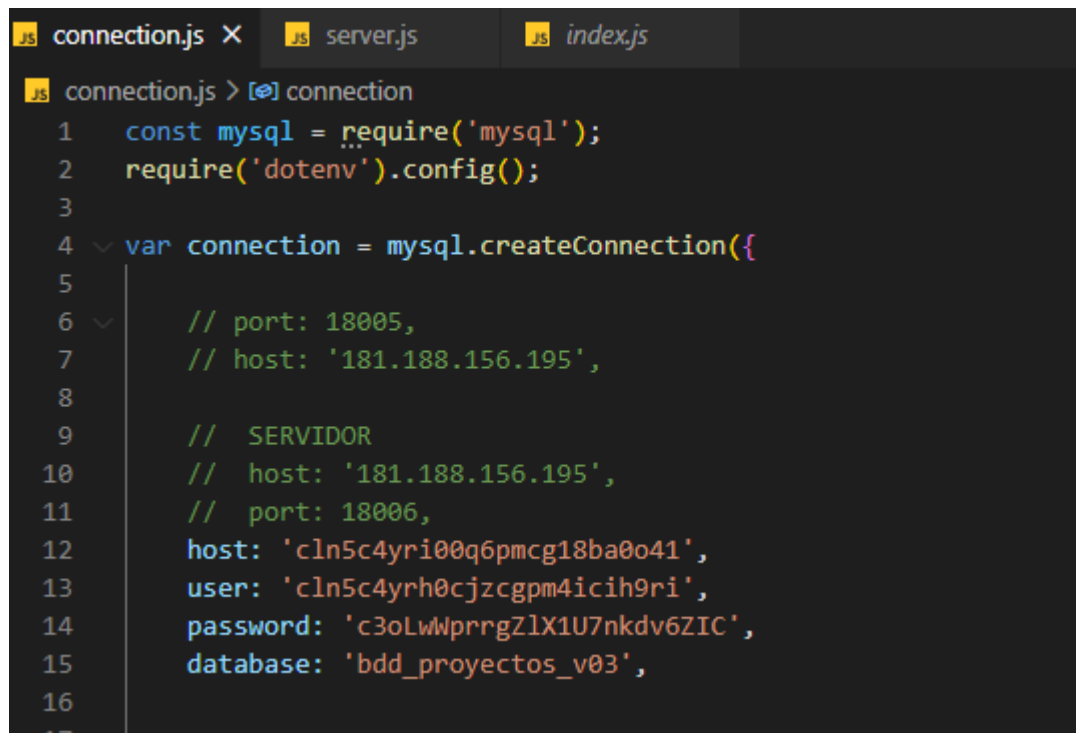
### 6.3 ESTRUCTURA DEL PROYECTO

La estructura del proyecto esta basado en Node.js, se compone con 5 carpetas que se explicaran detalladamente a continuación.

1. Libs: Abreviatura de Libraries, Bibliotecas en español. Esta carpeta contiene una dependencia, en específico multer.js
2. node\_modules: En esta carpeta se encuentra todas las dependencias utilizadas en el proyecto
3. Routes: Esta carpeta está dedicada a definir las rutas esenciales para cada módulo del proyecto:
  - Get: Recupera la lista completa de datos.
  - Create: Crea un nuevo elemento.
  - Delete: Elimina un elemento específico.
  - Update: Actualiza un elemento.
  - UpdateStatus: Actualiza el estado de un elemento a 'sí' o 'no'.
4. Services: Esta carpeta contiene un servicio con la autenticación a la base de datos, contiene un archivo llamado authentication.js
5. Uploads: Es el lugar donde se almacenan los informes y reportes cargados
6. Connection.js: El archivo donde se define el host, user, password y database. Todos esos parámetros deben tener similitudes con la base de datos
7. Index.js Este archivo es donde arranca la aplicación web
8. Server.js: aquí es donde cambias los datos para que la aplicación web arranque

### 6.4 CONEXIÓN A LA BASE DE DATOS

Para conectarte a la base de datos se debe ingresar con estas credenciales a MySQL:



```
JS connection.js X JS server.js JS index.js
JS connection.js > [?] connection
1  const mysql = require('mysql');
2  require('dotenv').config();
3
4  var connection = mysql.createConnection({
5
6    // port: 18005,
7    // host: '181.188.156.195',
8
9    // SERVIDOR
10   // host: '181.188.156.195',
11   // port: 18006,
12   host: 'c1n5c4yri00q6pmcg18ba0o41',
13   user: 'c1n5c4yrh0cjzcgpm4icih9ri',
14   password: 'c3oLwWprrgZlX1U7nkdv6ZIC',
15   database: 'bdd_proyectos_v03',
16
17 }
```

*Figura 15. Credenciales MySQL*

Host: Es el nombre del servidor donde se encuentra la base de datos

User: Es el nombre del usuario que se utiliza para autenticar la base de datos

Password: Se proporciona una contraseña al usuario, esto debe manejarse con cuidado.

Database: El nombre que se llama la Base de Datos

### 6.4.1 GESTIÓN DE CONEXIÓN Y DESCONEXIÓN

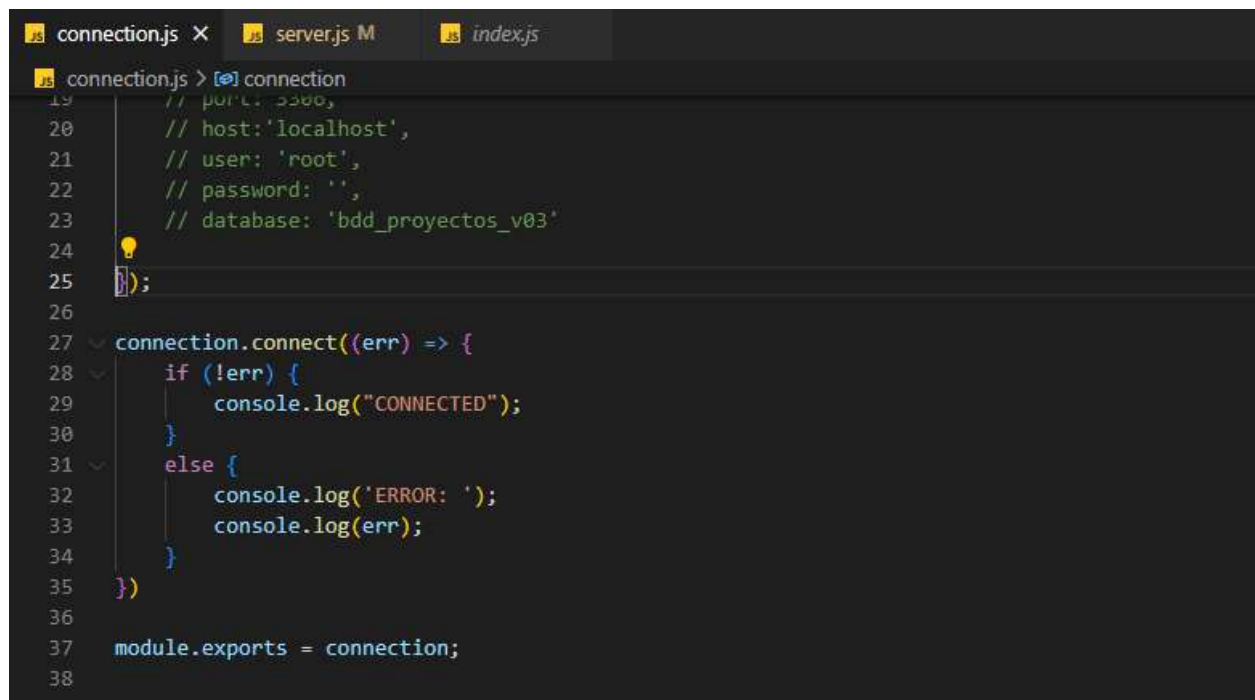


Figura 16. Conexion y desconexion en la BD

Se intenta establecer una conexión en la base de datos MySQL. Si la conexión tiene éxito, se imprimirá "CONNECTED". Si hay algún error durante la conexión, se imprimirá "ERROR:" seguido de detalles sobre el error.

### 6.4.2 USO EN MYSQL

Se debe presionar en la figura del signo + para agregar una nueva conexión MySQL



Figura 17. Nueva conexión MySQL

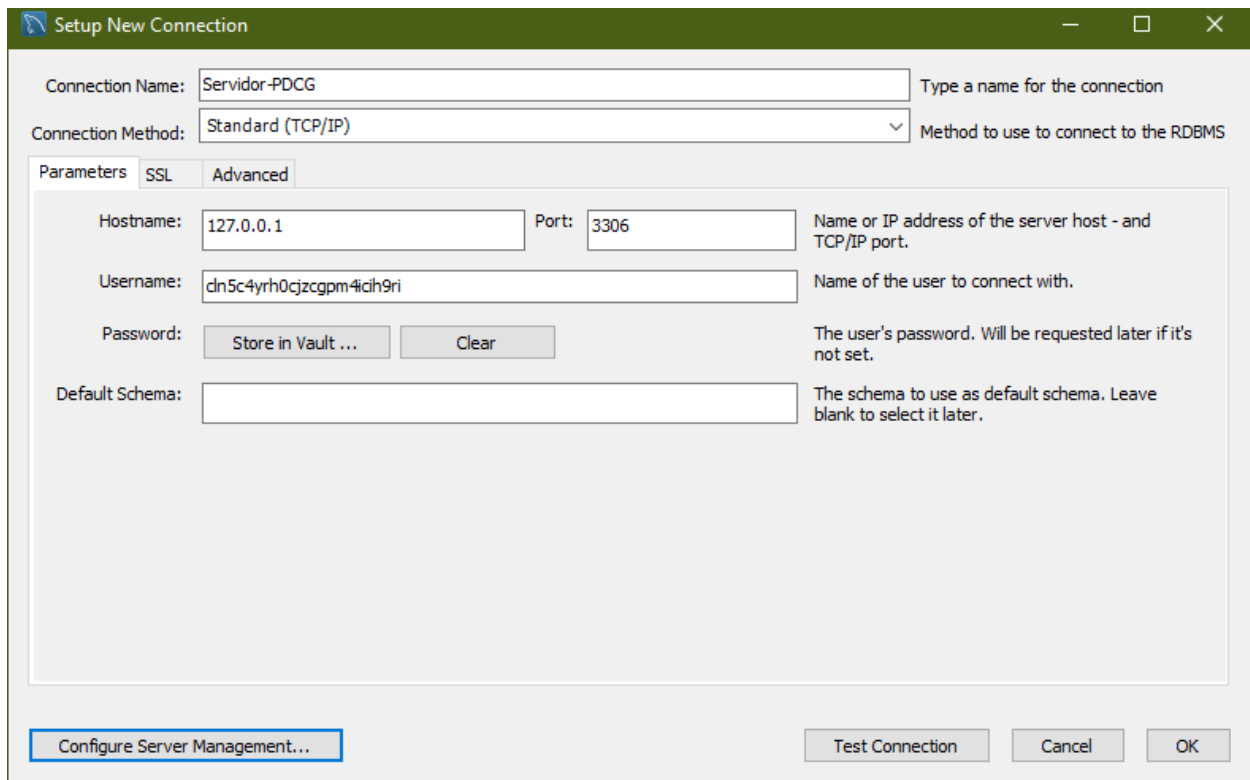


Figura 18. Conexión con MySQL

Y por último paso después de presionar el botón 'OK' debes colocar la contraseña establecida.

## 7 FRONTEND

### 7.1 GUÍA PARA LEVANTAR EL FRONTEND

Levantar el Frontend (Angular 15)

- Abre una terminal en la ubicación del directorio del frontend.
- Ejecuta ``npm install`` para instalar las dependencias del proyecto.
- Luego, ejecuta ``ng serve`` para iniciar el servidor de desarrollo del frontend. Por defecto, estará disponible en ``http://localhost:4200``.

### 7.2 TECNOLOGÍAS/DEPENDENCIAS UTILIZADAS

Las tecnologías que se ha utilizado:

- Angular v15 el framework principal.
- El lenguaje de programación para la parte de frontend: HTML, CSS y TypeScript.

Las dependencias utilizadas en el proyecto:

1. `"@angular/cdk": "^15.2.9"`.

Contiene una serie de herramientas de desarrollo para la creación de componentes Angular.



2. "@angular/common": "^15.2.0".

Contiene módulos comunes que son fundamentales para el desarrollo de aplicaciones Angular

3. "@angular/compiler": "^15.2.0".

Proporciona el compilador de Angular que se utiliza para compilar plantillas y código TypeScript en código JavaScript que se puede ejecutar en el navegador.

4. "@angular/forms": "^15.2.0".

Proporciona el compilador de Angular que se utiliza para compilar plantillas y código TypeScript en código JavaScript que se puede ejecutar en el navegador.

5. "@angular/material": "^15.2.9".

Un conjunto de componentes y estilos diseñados conforme a las directrices de Material Design.

6. "highcharts": "^11.1.0".

Una biblioteca popular para crear gráficos interactivos en el navegador. Ofrece una variedad de tipos de gráficos.

7. "file-saver": "^2.0.5".

Una biblioteca que proporciona utilidades para la descarga de archivos en el navegador.

8. "html2canvas": "^1.4.1".

Una biblioteca que permite la captura de la representación visual de elementos HTML y la creación de imágenes a partir de ellos. Útil para generar capturas de pantalla o imágenes de contenido HTML.

9. "jspdf": "^2.5.1".

Una biblioteca para generar documentos PDF en el navegador.

10. "pdfmake": "^0.2.7".

Una biblioteca para la creación de documentos PDF en el navegador. Ofrece un conjunto de funciones para definir la estructura y el contenido de documentos PDF.

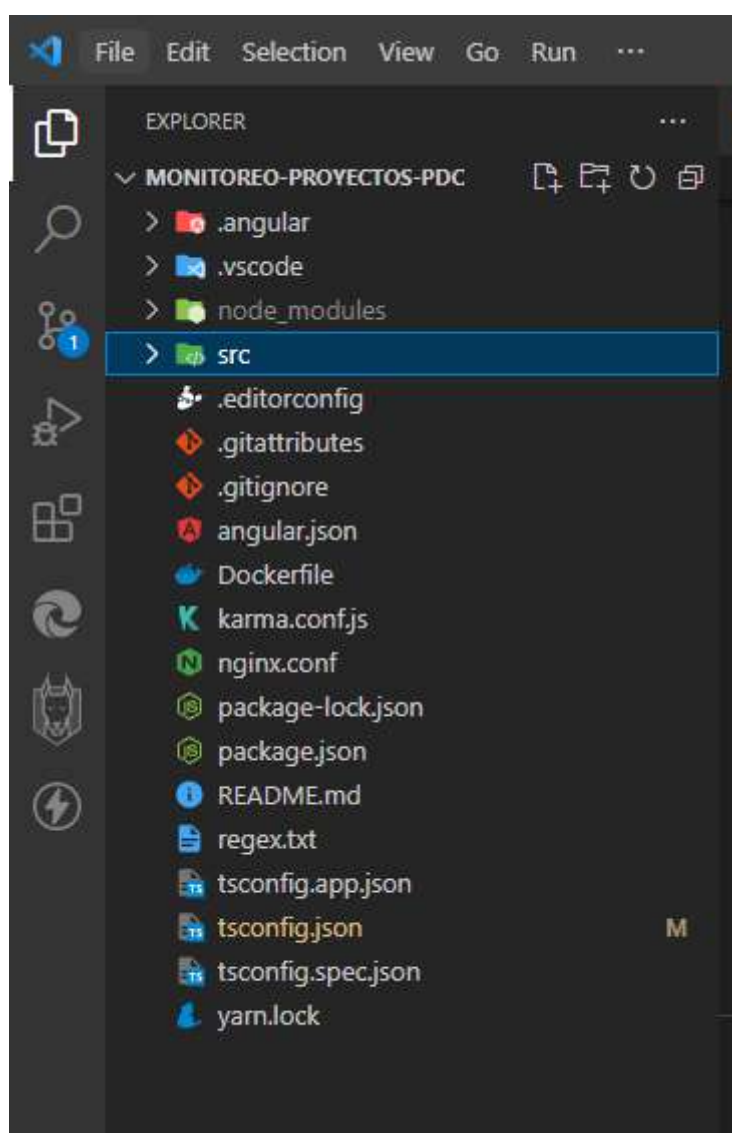
11. "xlsx": "^0.18.5".

Una biblioteca para la manipulación de archivos Excel en el navegador. Permite leer y escribir archivos Excel en formato .xlsx.

### 7.3 ESTRUCTURA DEL PROYECTO

La estructura del proyecto se basó en Angular.

La carpeta 'src' es el núcleo de la aplicación y aquí se almacena todo el código fuente.



*Figura 19. Estructura del proyecto Angular*

Dentro de la carpeta 'src' existe carpetas como:

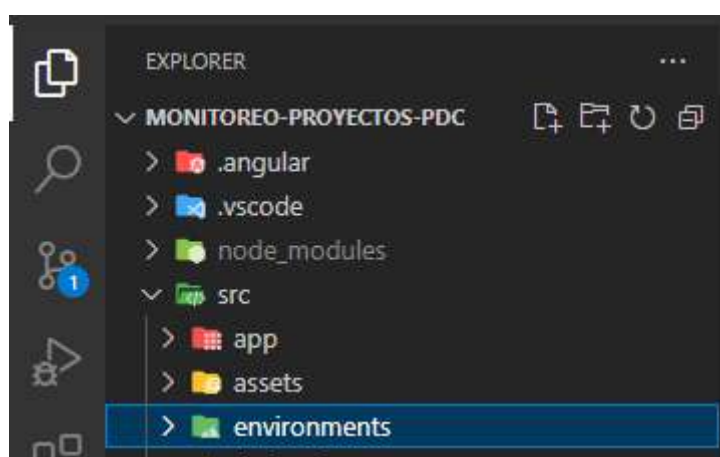


Figura 20. Carpeta 'src'

- app: La subcarpeta contiene los componentes del sistema, el login y los servicios.

En la parte de components se presenta una carpeta llamada dialog, aquí se encuentra subcarpetas adicionales que representa cada módulo. Cada subcarpeta contiene los dialogs de cada módulo.

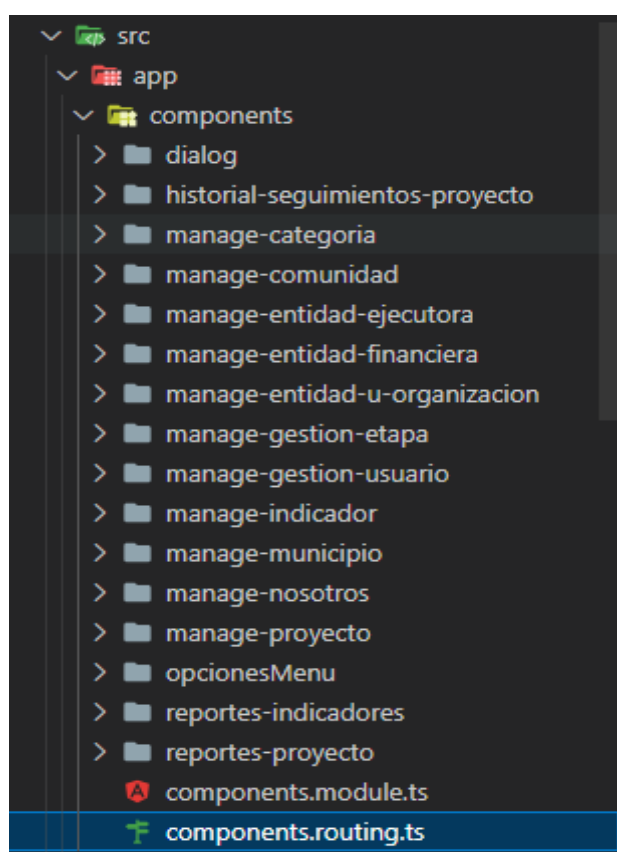


Figura 21. Carpeta app

- assets: La carpeta almacena archivos en PDF, PNG en la subcarpeta 'img', luego cabe recalcar en la subcarpeta capas se encuentra archivos GEOJSON que se encarga de proporcionar información geográfica.

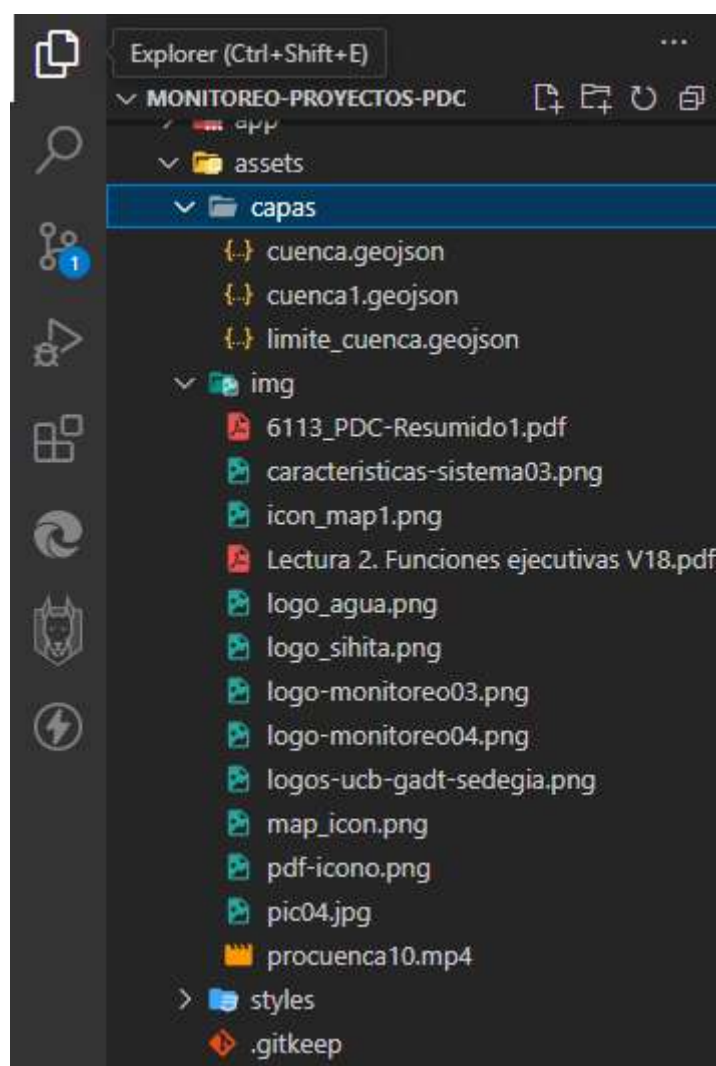


Figura 22. Carpeta 'assets'

- environment: La carpeta contiene las configuraciones de entorno, cuenta con la URL del sistema.

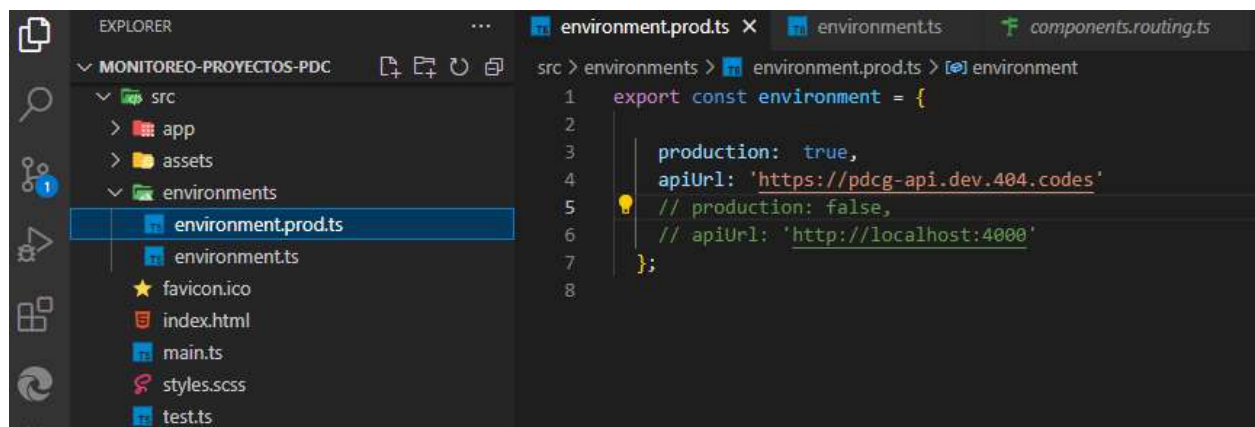


Figura 23. Carpeta 'environment'