



UNIVERSIDAD VALLE DEL MOMBOY; VICERRECTORADO
FACULTAD DE INGENIERÍA; PERIODO 2024B
CARRERA INGENIERÍA EN COMPUTACIÓN
FRONT END I; SECCIÓN VIV/FI
UNIDAD III; CSS & UX/UI

e-actividad 3.1:

Proyecto: Diseño Web con HTML & CSS

Informe Explicativo

Alumno:

Luis Fernando Araujo Giardinella C.I. 26.482.894

***Tutor:* Profesor Yerson González**

Contenido

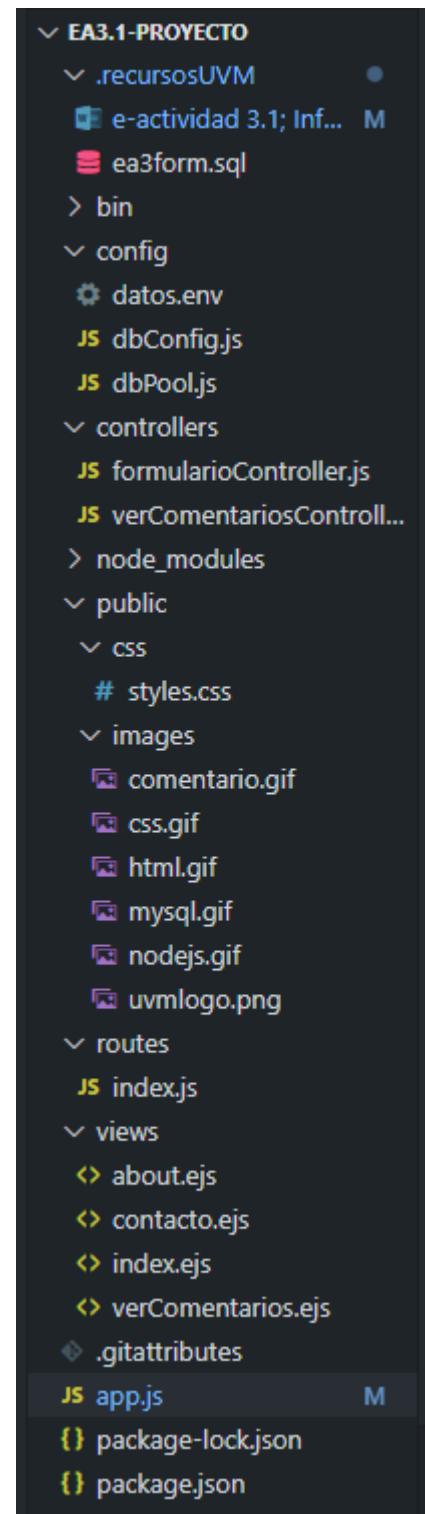
Introducción	3
Estructura del Proyecto.....	4
Frontend.....	4
Archivos y Recursos del Frontend	4
Backend	4
Archivos del Backend	4
Detalles del Código.....	5
Servidor: Archivo “app.js”	5
Controladores.....	5
Vistas	5
CSS.....	5
Funcionamiento del Proyecto	6
Página Principal	6
Página de Información	6
Página de Contacto	6
Página de Ver Comentarios	6
Funcionalidad del Formulario	6
Visualización de Comentarios	6
Instalación del Sistema	7
Requisitos	7
Visual Studio Code (VSCode)	7
Node.js	8
XAMPP	9
MySQL Workbench.....	10
GitHub	11
Despliegue	12
Paso 1: Clonando el Repositorio con GitHub	12
Paso 2: Ejecutar el Servidor Local MySQL con XAMPP.....	13
Paso 3: Crear la Base de Datos MySQL Local con MySQL Workbench	13
Paso 4: Inicializar el Archivo “app.js”	15
Accediendo a la Página Web	16
Comentarios de Desarrollo.....	17

Introducción

Este proyecto tiene como objetivo desarrollar una página web funcional utilizando Node.js, Express, MySQL, HTML y CSS. El propósito principal es proporcionar una base sólida para futuras actividades y proyectos, demostrando cómo integrar un backend robusto con un frontend interactivo y estético. La página permite a los usuarios dejar comentarios a través de un formulario, y esos comentarios se almacenan en una base de datos MySQL. Los comentarios pueden ser visualizados en una sección dedicada. Este proyecto funciona como un excelente punto de partida para aprender y aplicar principios de desarrollo web tanto en el lado del servidor como en el lado del cliente, teniendo en consideración que las siguientes actividades de esta materia serán más amplias y detalladas.

El proyecto incluye los recursos e instrucciones necesarias para tu desarrollo o réplica dentro de cualquier máquina compatible. Esto incluye tanto el despliegue del proyecto en sí, como su creación tanto en código como también en términos de base de datos y conexión correspondiente.

Imagen a la derecha: Visualización del directorio del proyecto.



Estructura del Proyecto

Frontend

El frontend incluye todas las partes visuales y de interacción del usuario. Está compuesto por archivos HTML, CSS y algunas imágenes. Las vistas están escritas en EJS (Embedded JavaScript) para facilitar la renderización dinámica de contenido.

Archivos y Recursos del Frontend

1. Archivos HTML/EJS:

- *index.ejs*: Página principal con bienvenida e información general.
- *about.ejs*: Página de información sobre el proyecto.
- *contacto.ejs*: Página de contacto con el formulario para dejar comentarios.
- *verComentarios.ejs*: Página para ver los comentarios dejados por los usuarios.

2. Archivo CSS:

- *styles.css*: Archivo de estilos que define la apariencia visual de las páginas.

3. Imágenes:

- *public/images*: Carpeta que contiene las imágenes utilizadas en el sitio web.

Backend

El backend maneja la lógica del servidor, las rutas, y la interacción con la base de datos. Está implementado usando Node.js y Express.

Archivos del Backend

1. Configuración:

- *config/datos.env*: Archivo de configuración con las credenciales de la base de datos.
- *config/dbConfig.js*: Configuración de la conexión a la base de datos usando las variables de entorno.
- *config/dbPool.js*: Configuración del pool de conexiones a la base de datos.

2. Servidor:

- *app.js*: Archivo principal del servidor que configura Express, EJS, body-parser, y las rutas.

3. Controladores:

- *controllers/formularioController.js*: Controlador para manejar la lógica de guardar formularios en la base de datos.
- *controllers/verComentariosController.js*: Controlador para manejar la lógica de recuperar y mostrar los comentarios desde la base de datos.

4. Rutas:

- *routes/index.js*: Archivo que define las rutas de la aplicación.

Detalles del Código

Servidor: Archivo “app.js”

Este archivo configura el servidor Express, establece el motor de vistas en EJS, configura body-parser para manejar datos de formularios y define las rutas estáticas y dinámicas.

Controladores

Los controladores manejan la lógica de negocio para las rutas definidas. Por ejemplo, el archivo “*formularioController.js*” guarda los datos del formulario en la base de datos y el archivo “*verComentariosController.js*” recupera y muestra los comentarios almacenados.

Vistas

Las vistas son plantillas EJS que se renderizan en el servidor y se envían al cliente. Dentro del caso de este proyecto, las plantillas podrían considerarse los archivos HTML, debido a su formato y su uso como el archivo que será visualizado por parte del usuario dentro del frontend. Los archivos de vistas son: *index.ejs*, *about.ejs*, *contacto.ejs* y *verComentarios.ejs*.

CSS

Este es un único archivo llamado “*styles.css*” que define la apariencia visual de las páginas web o vistas anteriormente nombradas. Se encarga de especificar colores, tipo de texto, efectos visuales, y mucho más detalles estéticos. Además, también se encarga de limitar o ajustar tamaño de imágenes y su posición.



Imagen: Directorio del proyecto visto desde el explorador de Windows.

Funcionamiento del Proyecto

Página Principal

- *index.ejs* muestra una bienvenida y un enlace al repositorio de GitHub.

Página de Información

- *about.ejs* describe el propósito del proyecto y las tecnologías utilizadas.

Página de Contacto

- *contacto.ejs* contiene un formulario para que los usuarios dejen sus comentarios.

Página de Ver Comentarios

- *verComentarios.ejs* muestra los comentarios que han dejado los usuarios, recuperados de la base de datos.

Funcionalidad del Formulario

- Al enviar el formulario en *contacto.ejs*, los datos se envían a la ruta */contacto* que maneja el controlador *guardarFormulario*.
- Los datos se almacenan en la tabla *formularios_contacto* de la base de datos "ea3form".

Visualización de Comentarios

- La ruta */comentarios* maneja el controlador *verComentarios* que recupera y muestra los comentarios almacenados en la base de datos en la vista *verComentarios.ejs*.

Instalación del Sistema

Requisitos

Visual Studio Code (VSCode)

Es un editor de código fuente desarrollado por Microsoft. Es ligero, potente y soporta una gran variedad de lenguajes de programación y archivos. Para este proyecto, se utilizará para ejecutar el servidor local.

1. El usuario debe dirigirse al siguiente enlace de descarga:
<https://code.visualstudio.com/download>
2. A continuación, debe hacer clic en el botón de descarga para el sistema operativo que utiliza, sea Windows, macOS o Linux.
3. Tras descargar el archivo de instalación, debe abrirlo y seguir las instrucciones proporcionadas por el asistente de instalación.
4. Una vez completada la instalación, VSCode estará listo para su uso.



Imagen: Página web de descarga para VSCode

Node.js

Es el entorno de ejecución para JavaScript. En este proyecto, Node.js es necesario para ejecutar el servidor del proyecto, más específicamente nos permitirá alojar los endpoints, permitiendo la interacción con el sistema a través de protocolos HTTP.

- 1. El usuario debe dirigirse al sitio web oficial de Node.js para iniciar la descarga: <https://nodejs.org/en>
- 2. Luego, debe hacer clic en el botón de descarga que corresponda al sistema operativo que esté utilizando, ya sea Windows, macOS o Linux.
- 3. Una vez descargado el archivo de instalación, se debe de abrir y seguir las instrucciones del asistente de instalación.
- 4. Durante la instalación, es importante asegurarse de marcar la casilla que indica "Instalar herramientas de línea de comandos" para poder utilizar Node.js desde la terminal.
- 5. El usuario puede abrir la terminal y verificar que Node.js se haya instalado correctamente escribiendo el comando **node -v** y/o **npm -v**. De esta forma, debería ver la versión de Node.js y npm que han sido instaladas.

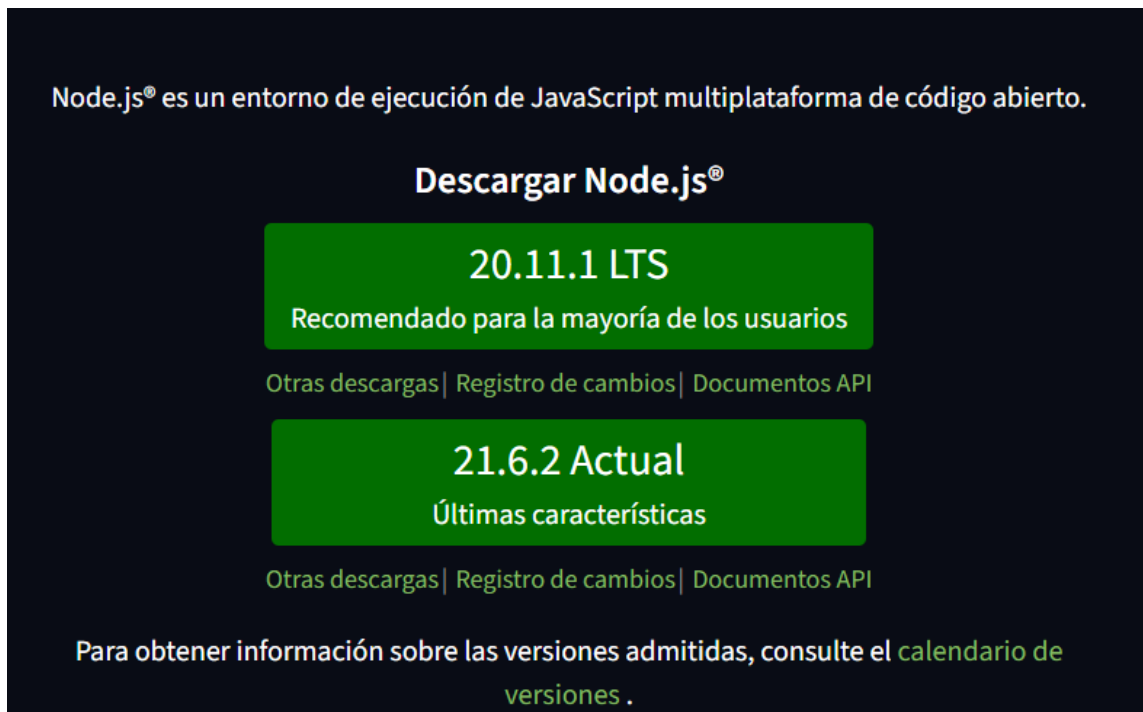


Imagen: Página web de descarga para Node.js

XAMPP

Es una distribución de Apache fácil de instalar que contiene MariaDB, PHP y Perl. Para el propósito de este proyecto, XAMPP se utilizará principalmente para su servidor MySQL, el cual proporcionará el sistema de gestión de bases de datos necesario para almacenar y gestionar la información del proyecto.

1. El usuario debe dirigirse al sitio web oficial de XAMPP para iniciar su descarga:
<https://www.apachefriends.org/index.html>
2. Debe descargar la versión de XAMPP que corresponda a su sistema operativo (Windows, macOS o Linux).
3. Una vez descargado el archivo de instalación, debe seguir las instrucciones del asistente de instalación.
4. Durante la instalación, el usuario debe elegir los componentes que desea instalar (Apache, MySQL, PHP, phpMyAdmin, etc.). Se puede dejar las opciones predeterminadas para una instalación básica.
5. El usuario debe elegir la carpeta de instalación para XAMPP. Por lo general, se recomienda instalarlo en la carpeta predeterminada.
6. Debe completar la instalación y asegurarse de que los servicios de Apache y MySQL se inicien automáticamente al finalizar la instalación.



Imagen: Página web de descarga para XAMPP

MySQL Workbench

Es una herramienta visual de diseño de bases de datos que integra desarrollo, administración, diseño, creación y mantenimiento de bases de datos en un único entorno integrado. En este proyecto, MySQL Workbench se utilizará para diseñar, crear y gestionar la base de datos local que almacena toda la información relevante del sistema.

- 1. Dirigirse al sitio web oficial de MySQL Workbench:
<https://dev.mysql.com/downloads/workbench/>
- 2. Descargar la versión de MySQL Workbench que corresponde al sistema operativo (Windows, macOS o Linux).
- 3. Una vez descargado el archivo de instalación, se deben seguir las instrucciones del asistente de instalación.
- 4. Durante la instalación, puedes elegir las opciones predeterminadas. Se recomienda al usuario instalar con las opciones por defecto.

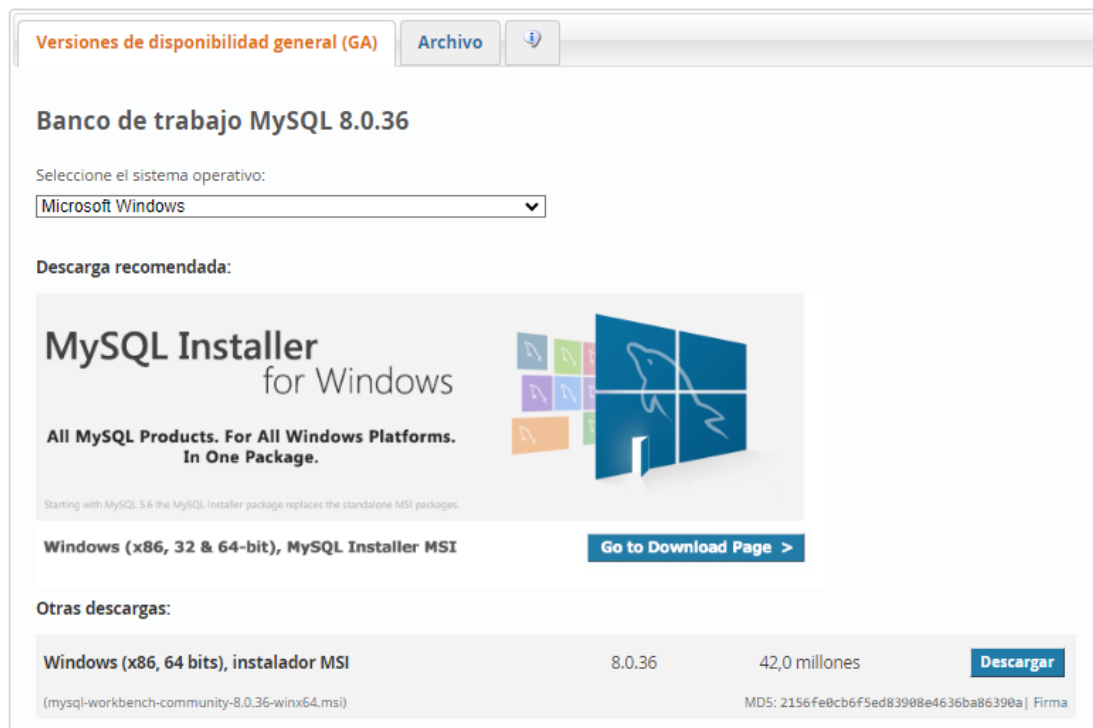


Imagen: Página web de descarga para Mysql Workbench

GitHub

Es una plataforma de desarrollo colaborativo para alojar proyectos utilizando el sistema de control de versiones Git. Se empleará para clonar el repositorio del proyecto localmente, permitiendo el acceso al código fuente y para la fácil implementación del proyecto en la máquina local. Para usarlo deberás seguir los siguientes pasos:

- 1. Crear una cuenta en GitHub: El usuario debe ir al sitio web oficial de GitHub (<https://github.com/>) para crear una cuenta propia.
- 2. Descargar e instalar la aplicación GitHub Desktop: Se necesita la aplicación para clonar el repositorio a la máquina local. A través del sitio web de descarga (<https://desktop.github.com/>) el usuario debe seguir las instrucciones para descargar y, una vez descargado, instalar la aplicación.
- 3. Iniciar sesión en la aplicación: Una vez la aplicación ha sido instalada, el usuario puede iniciar sesión para poder clonar repositorios.

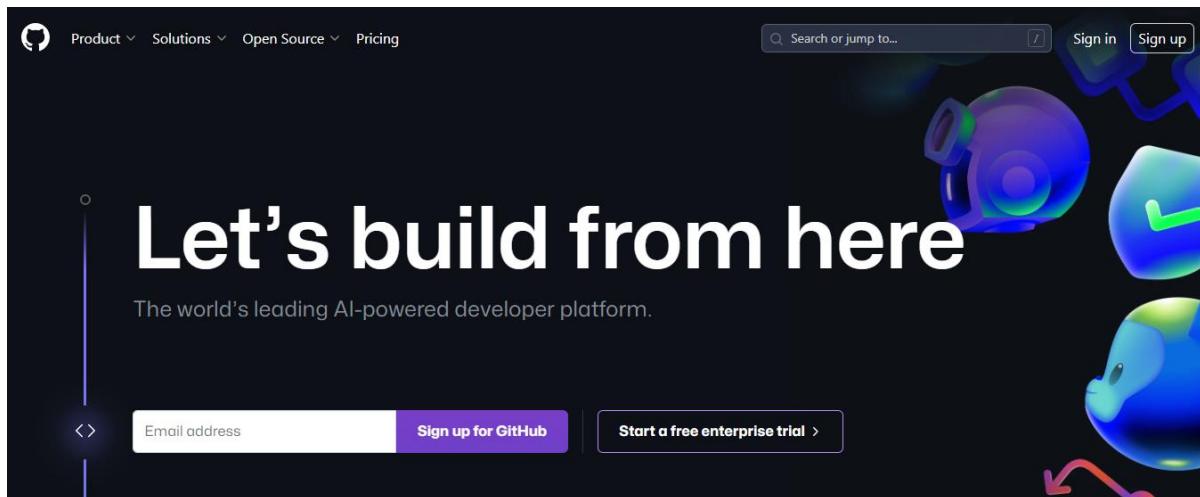


Imagen: Página web de GitHub

Despliegue

Una vez la máquina local posee los requisitos necesarios, se puede proceder a la instalación y despliegue del sistema. El correcto despliegue brindará seguridad y minimizará los inconvenientes a futuro durante el uso del proyecto..

Paso 1: Clonando el Repositorio con GitHub

El proceso de despliegue comienza al obtener los archivos necesarios del sistema a través de la aplicación de GitHub instalada. El usuario puede clonar el repositorio a través de la URL <https://github.com/Luixls/Frontend-ea3.1-proyecto> y seleccionando la opción correcta:

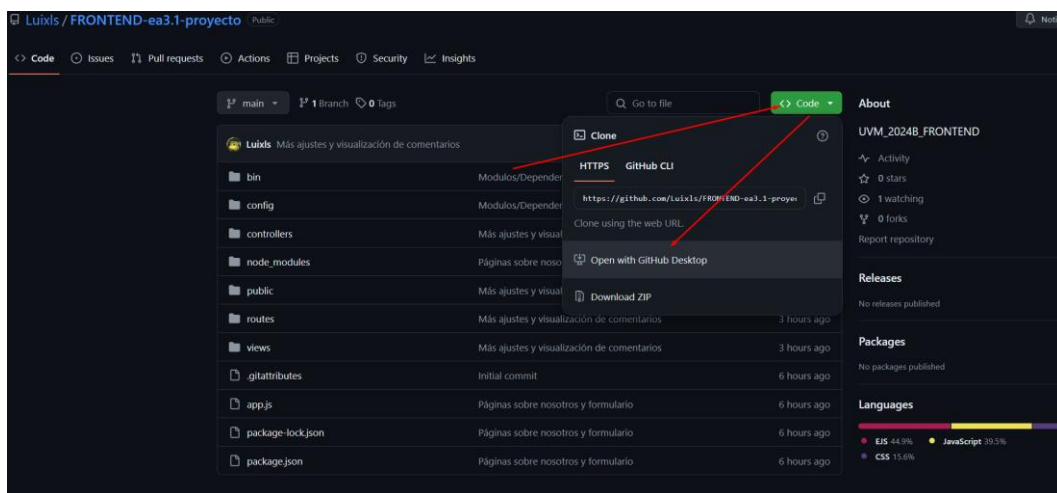


Imagen: Seleccionando la opción para obtener el repositorio con GitHub Desktop. Existe la opción de descargar el archivo zip, pero para este ejemplo, continuaremos con la aplicación GitHub Desktop.

La aplicación detectará que se quiere clonar un repositorio a la máquina local, y presentará el cuadro de confirmación para que el usuario seleccione qué nombre desea colocarle localmente al repositorio y en qué ubicación guardarlo.

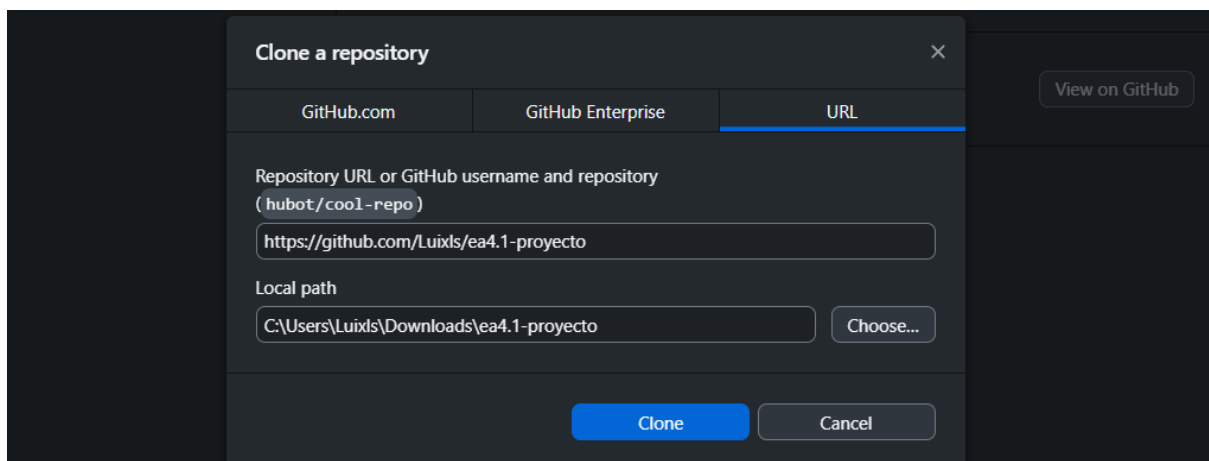


Imagen: Cuadro de confirmación de GitHub Desktop.

Paso 2: Ejecutar el Servidor Local MySQL con XAMPP

Procedemos a ejecutar el servidor para la base de datos utilizando el programa XAMPP, **se recomienda ejecutar XAMPP como administrador** para minimizar incompatibilidades. Una vez la ventana de XAMPP está presente, podemos iniciar el servidor apretando el botón “Start” para Apache y MySQL.

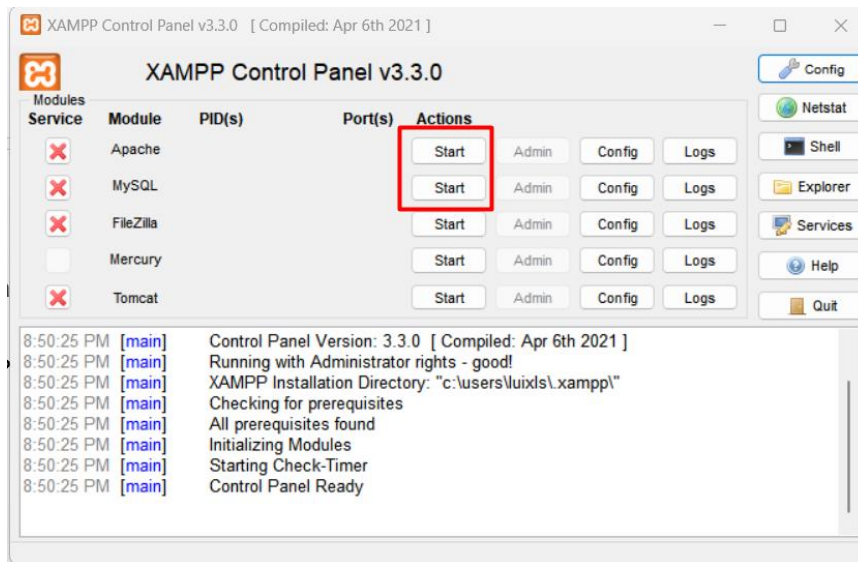


Imagen: Ventana de XAMPP, con los botones para iniciar el servidor resaltados.

Paso 3: Crear la Base de Datos MySQL Local con MySQL Workbench

Con el servidor MySQL ejecutándose, se procede a abrir MySQL Workbench y se establece la conexión al servidor local.

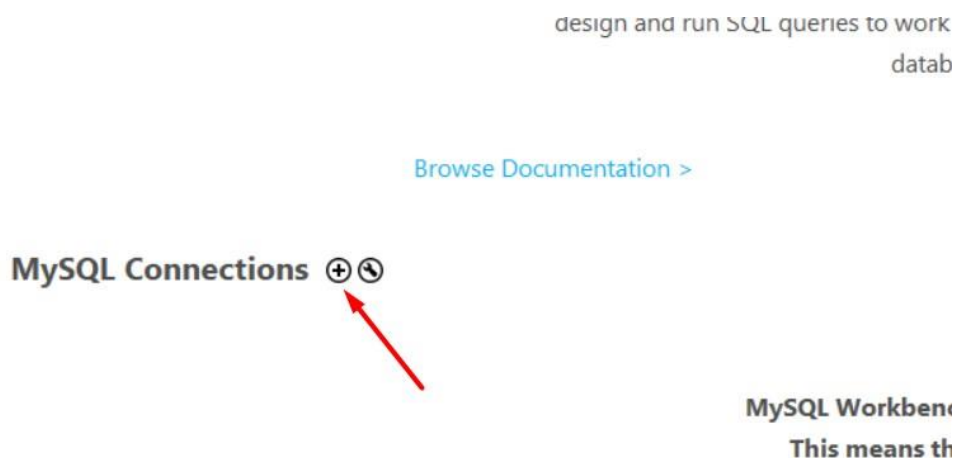


Imagen: Dentro de MySQL Workbench, botón para agregar conexión.

Se recomienda al usuario dejar los valores por defecto al momento de realizar la conexión, para que no haga falta editar archivos dentro del directorio del proyecto.

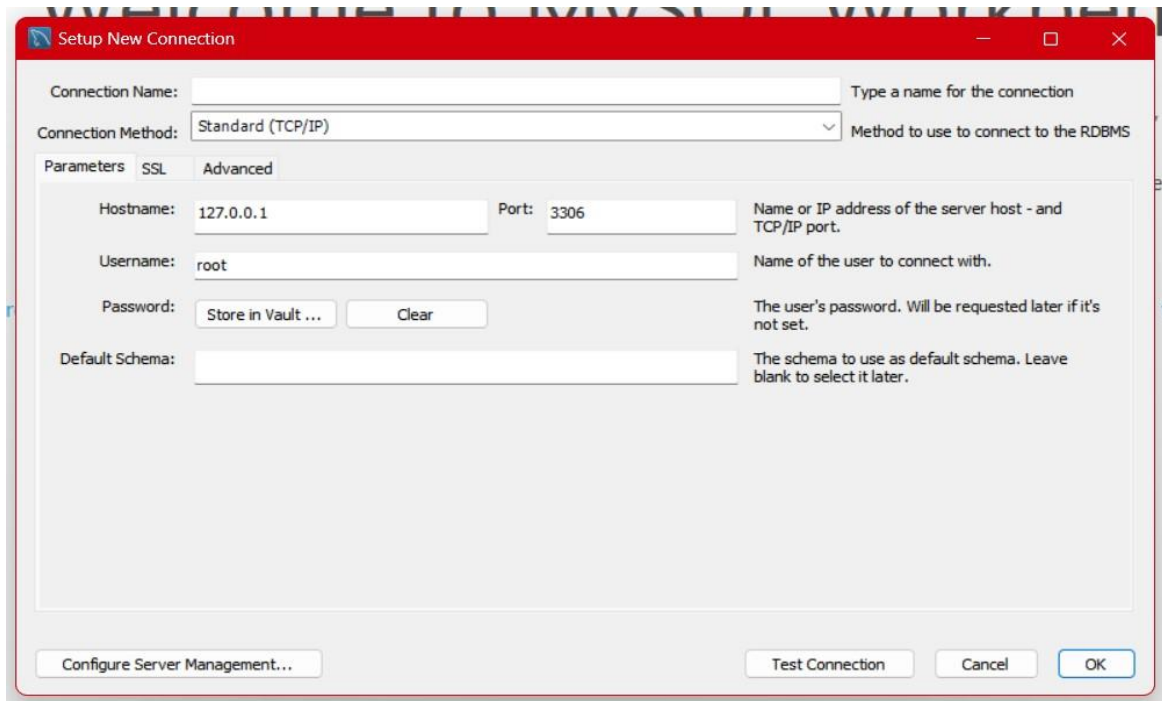


Imagen: Dentro de MySQL Workbench, ventana de configuración de conexión.

Una vez la conexión se establece, procedemos a abrir el script SQL proveído en la carpeta “.recursosUVM” dentro del directorio del proyecto. Para ello, seleccionamos la opción de “Open SQL Script...” dentro de MySQL Workbench.

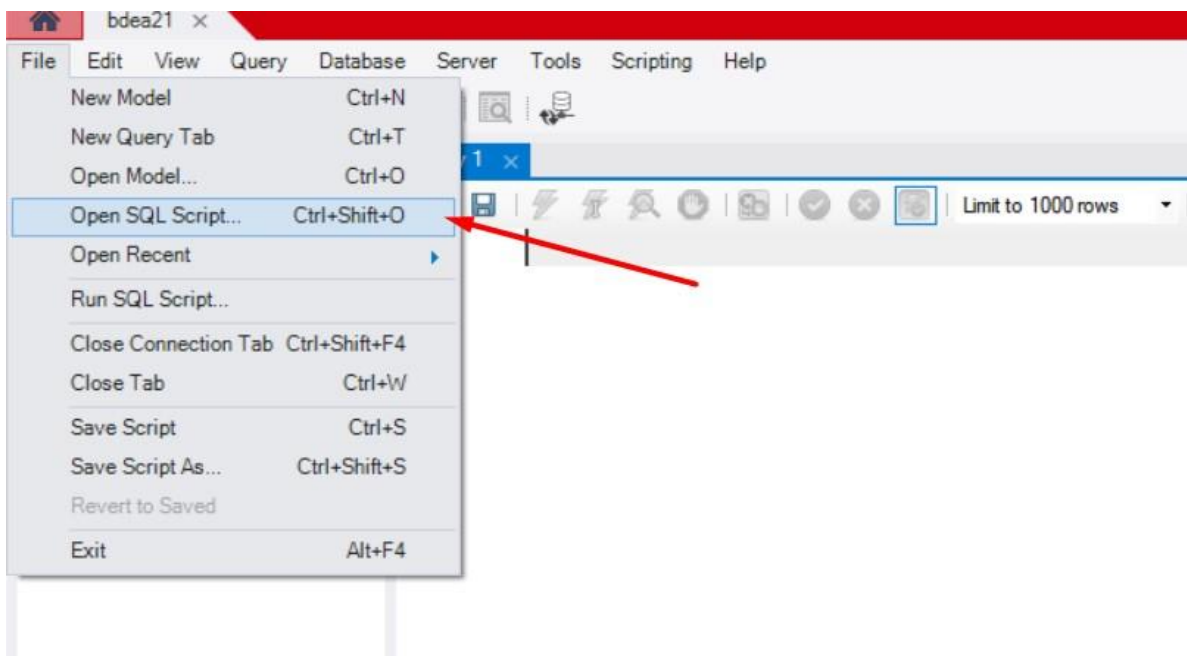


Imagen: Dentro de MySQL Workbench, opción para abrir un archivo script SQL.

Siguiente, simplemente se debe ejecutar dicho archivo apretando el botón correspondiente dentro de MySQL Workbench. La base de datos local será creada con los parámetros necesarios, e incluirá entradas (datos) de ejemplo pre-existentes.

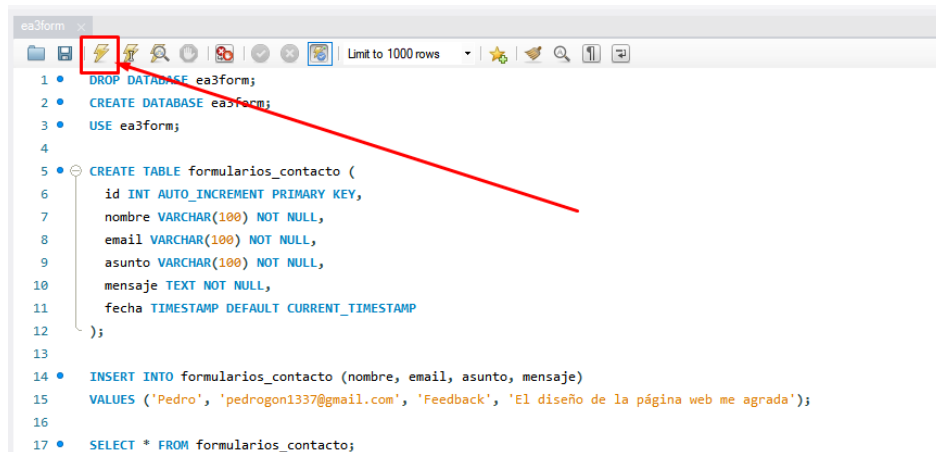


Imagen: Dentro de MySQL Workbench, botón para ejecutar el script con la creación de la base de datos local.

Ejecutar este script **asignará una contraseña a la base de datos**. En este ejemplo, será “123” y se trabajará con esta contraseña por defecto para proseguir.

Paso 4: Inicializar el Archivo “app.js”

Para el siguiente paso, se debe ejecutar el archivo que corresponde al servidor del proyecto, localizado en la raíz del directorio. Para ello, a través de VSCode, debemos elegir la opción de abrir una carpeta, la cual es el proyecto que ha sido clonado desde GitHub.

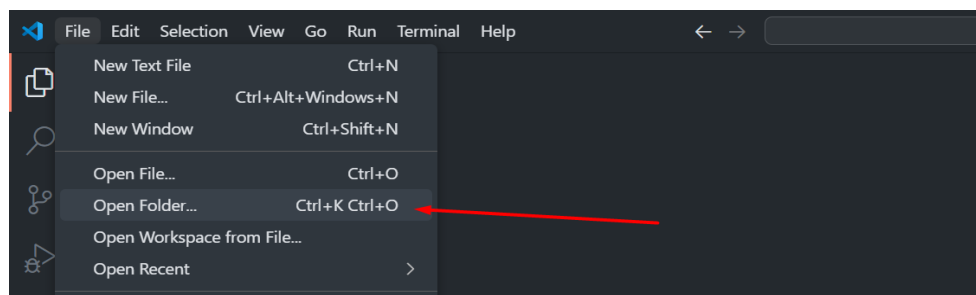


Imagen: Dentro de VSCode, seleccionando la opción de abrir carpeta.

Por último, solo es cuestión de utilizar el comando “***npx nodemon app.js***” para ejecutar el servidor local del proyecto a través de la consola de VSCode. Un mensaje de conexión exitosa será mostrado cuando el servidor arranque.

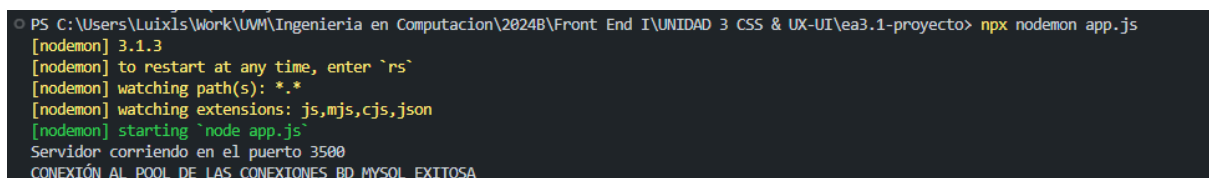


Imagen: Dentro de VSCode, ejecutando el comando para iniciar el servidor local del proyecto.

Accediendo a la Página Web

Una vez se han seguido los pasos para el correcto despliegue del proyecto, podemos acceder a la página web a través de nuestro navegador de internet de preferencia. Por defecto, el proyecto trabaja en el puerto local 3500 y por ende, el usuario debe dirigirse a la URL <http://localhost:3500/> para visualizar la página web principal, o de “landing”.



Imagen: Dentro del navegador, la página web principal del proyecto.

El usuario puede navegar a través del dominio gracias a los botones que se encuentran al tope de la página web.

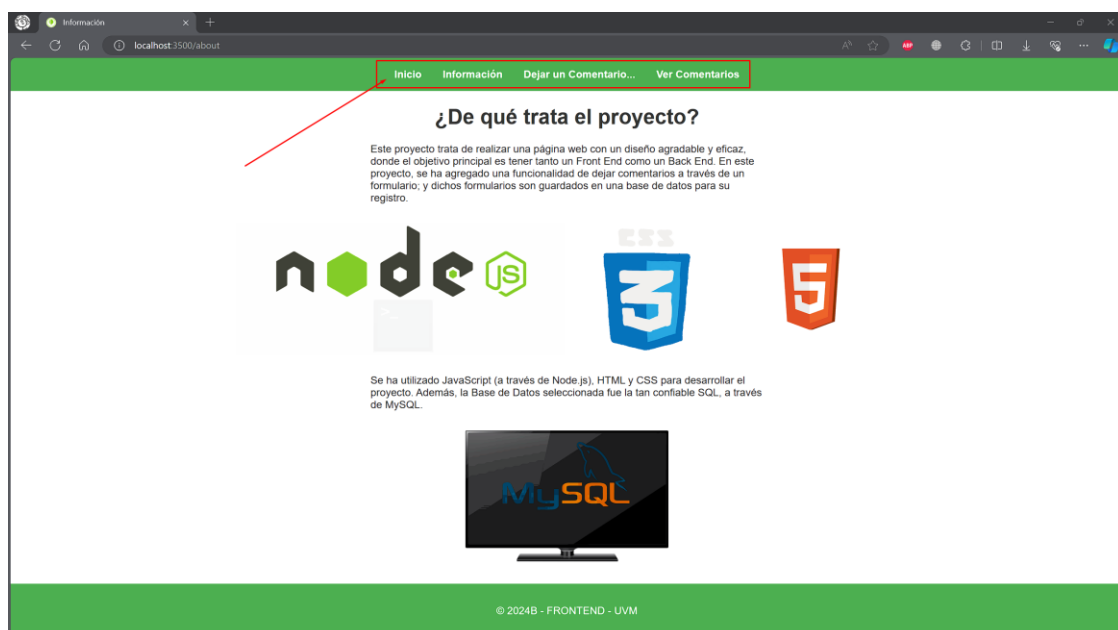


Imagen: Dentro del navegador, la página web “Información” que muestra de qué trata el proyecto.

Comentarios de Desarrollo

El proyecto demuestra cómo integrar de manera efectiva el frontend y el backend para crear una aplicación web funcional y estética. Dentro de los desafíos principales, se puede resaltar la integración de las tecnologías de backend (Node.js, Express, MySQL) con las tecnologías de frontend (HTML, CSS, EJS) ya que puede ser complicada, especialmente al manejar la comunicación entre el servidor y la interfaz de usuario. Por otro lado, en mi experiencia personal, siempre he tenido la opinión de que diseñar en CSS no es para nada intuitivo; sin embargo esta es una opinión sesgada en el hecho de que he trabajado como diseñador gráfico anteriormente utilizando programas como Photoshop. La diferencia de facilidad de diseño es, sin duda alguna, como noche y día para mí.

El repositorio puede ser accedido a través de la siguiente URL a GitHub:

<https://github.com/Luixls/FRONTEND-ea3.1-proyecto>

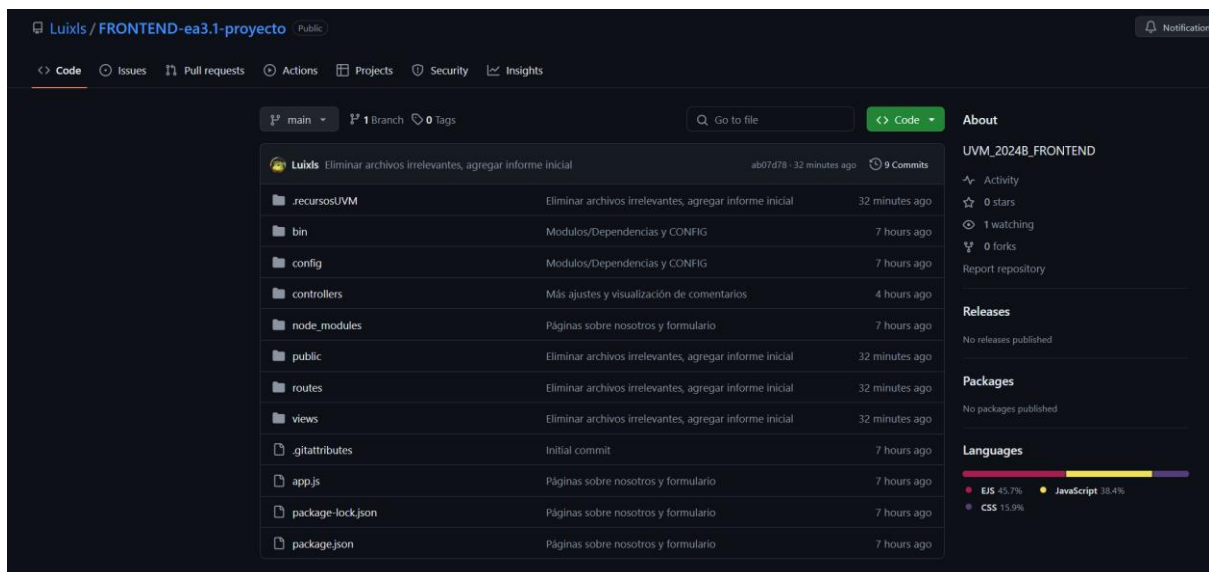


Imagen: Repositorio del proyecto en GitHub.

Por otro lado, los commits pueden ser accedidos desde el siguiente enlace:

<https://github.com/Luixls/FRONTEND-ea3.1-proyecto/commits/main/>

Y para culminar, el video explicativo del proyecto puede ser accedido a través de la siguiente URL a YouTube: <https://youtu.be/R7PoRKBaBYQ>