

**REPUBLICA BOLIVARIANA DE VENEZUELA**

**UNIVERSIDAD VALLE DEL MOMBOY**

**FACULTAD DE INGENIERÍA**

**CARVAJAL- EDO TRUJILLO**

MANUAL TECNICO DE UN SERVIDOR BACKEND

**Autores:**

Steven Sifuentes C.I: 25.170.885

Luis Morillo. CI: V-27.022.220

Roque Barrios C.I: 27.677.101

Javier Giralte C.I: 27.677.371

Abril de 2021

**ÍNDICE GENERAL**

[Introducción 3](#_Toc69680381)

[Instalación de un servidor basado en NodeJS 4](#_Toc69680382)

[Pasos para la instalación del servidor con NodeJS 5](#_Toc69680383)

[Obtener NodeJS 5](#_Toc69680384)

[Instalando NodeJS en Windows 6](#_Toc69680385)

[Instalación de un entorno de desarrollo y ejecución 11](#_Toc69680386)

[**Proceso de Instalación de XAMMP** 17](#_Toc69680387)

[**Pasos para la creación de la base de datos MySQL, usando Phpmyadmin.** 21](#_Toc69680388)

[**Creando un servidor backend con NodeJS y Express** 24](#_Toc69680389)

[**Conclusión** 36](#_Toc69680390)

[REFERENCIAS 37](#_Toc69680391)

Introducción

El estudio de un lenguaje de programación es básicamente el desarrollo de un sistema computacional de sistematización extensible y adaptable, estableciendo principalmente el desarrollo del software cuya finalidad a lo largo de su inducción en la sociedad ha sido muy beneficioso y de gran impacto en el crecimiento humano, gracias a los ventajosos procedimientos tecnológicos muchas personas han podido experimentar y poner a prueba sus estudios.

Con la creación de la W.W.W (World Wide Web) por Tim Berners Lee aunado a ello se implementaron otras tres tecnologías de la web que son el HTML, HTTP y URL. Esta tecnología desemboca a la arquitectura cliente-servidor cuyo objeto es la capacidad de proceso, está repartida entre los clientes y los servidores, con una importancia significativa de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

Con esta implementación fundada principalmente en el desarrollo sistemático de la arquitectura cliente-servidor, lograron surgir distintas empresas de software y aplicaciones como son Google, Facebook, Microsoft, entre otros. Causantes de gran impacto mundial, facilitando el modo comunicacional, desempeño laboral, variedad de métodos de enseñanza, y creando a partir de esta modalidad el nacimiento de múltiples empresas y modelo de negocio que hoy mueven la economía de ciudades, países y continentes enteros.

El trabajo comienza con la selección documentada en el análisis, diseño y desarrollo del backend de un software que busca visibilizar y proveer información sobre el uso y consumo efectivo del usuario en cuestión. Diversos lenguajes existen para este entorno, la única realidad absoluta es que de momento JavaScript no tiene un rival directo en el Frontend y su uso ha crecido aún más con la implementación de este lenguaje interpretado en el Backend, utilizando NodeJS.

Instalación de un servidor basado en NodeJS

En el ambiente de la programación web existen una gran variedad de lenguajes de programación, asimismo la aparición de nuevos frameworks. Escoger entre alguno entre la multitud de lenguajes suele ser difícil, sin embargo, se debe tomar en cuenta los recursos para lograr hacer un trabajo de calidad y aprender de forma excepcional los parámetros exigidos.

Cada uno de estos lenguajes va acompañado de ciertos frameworks, los cuales permiten solucionar problemas comunes de programación, acelerar la creación de aplicaciones con calidad de producción y aumentar el rendimiento de los programadores.

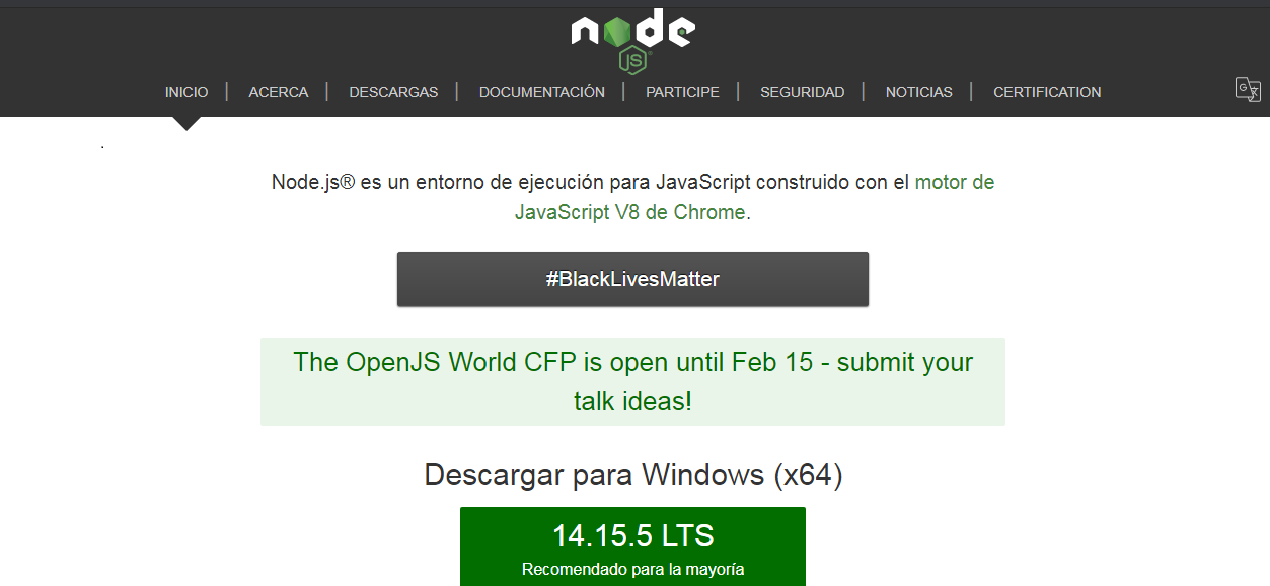
Es importante reconocer que NodeJS no es un lenguaje de programación, sino un entorno de ejecución del lenguaje JavaScript que permite construir servidores Backend con este lenguaje, con la llegada de NodeJS se abrió un nuevo mundo y empezaron a surgir los servidores web hechos con Express o con otras librerías basadas en Node, las de API Rest, posee una arquitectura basada en eventos y con énfasis en crear aplicaciones web altamente escalables. Para desarrollar en Node se realiza a través del lenguaje de programación JavaScript, que actualmente está teniendo popularidad y mejoras permitiendo el desarrollo tanto para Frontend como para backend, abriendo el camino a los profesionales fullstack. La mayor ventaja competitiva de NodeJS recae en su naturaleza asíncrona, su corazón es el *Event Loop*, un ciclo infinito donde cada evento disparado en NodeJS se procesa, pudiendo un evento delegarse a algún API y este seguir en ejecución con otro subsecuente, hasta que finalmente la ‘delegación’ culmine y el evento vuelta al hilo principal.

Generalmente, una aplicación web cada vez que realiza solicitudes a un servidor Backend, este levanta un hilo de ejecución el cual suele ocupar una cantidad pequeña de memoria en el sistema, si una tarea es muy pesada, se pueden crear más hilos de ejecución para realizar la tarea en un tiempo eficiente, y esto a su vez consume mayor memoria RAM; esto no es un problema para aplicaciones con pocos usuarios, pero a medida que está escala, si no se escalan los recursos, entonces la aplicación queda sumergida en un cuello de botella. Este tipo de cosas no suceden mientras no se bloquee el *event loop* de NodeJS, y es la principal característica que ha patrocinado la rápida adaptación de este lenguaje en grandes empresas de tecnología.

Pasos para la instalación del servidor con NodeJS

Obtener NodeJS

Para obtener NodeJS, se debe ingresar al sitio web oficial (<https://nodejs.org/es/>), al ingresar a la página web, se observa que existen dos tipos de descarga LTS (Recomendado por tener compatibilidad a largo tiempo) o la versión completa (No es tan recomendable) en mi caso lo descargaré para una versión de Windows de arquitectura de 32 bits, y una versión 12.16.1, a continuación la ilustración :



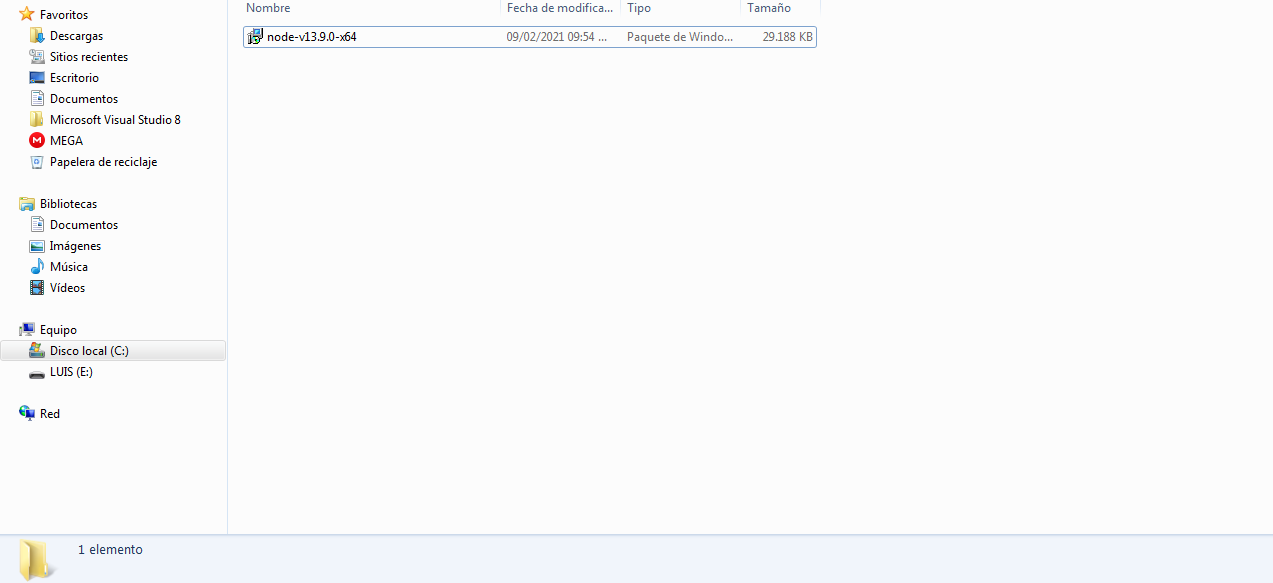
**Ilustración 1. Página web oficial de NodeJS (abril-2021).**

**Fuente:** Creación propia.

Visitando otra sección de la página web oficial de NodeJS (<https://nodejs.org/es/download/>), se logra estimar otra importante particularidad de NodeJS: Es multiplataforma, con esto se presentan diferentes opciones para adquirir el entorno de ejecución considerando el sistema operativo que tengamos instalado**.**

Instalando NodeJS en Windows

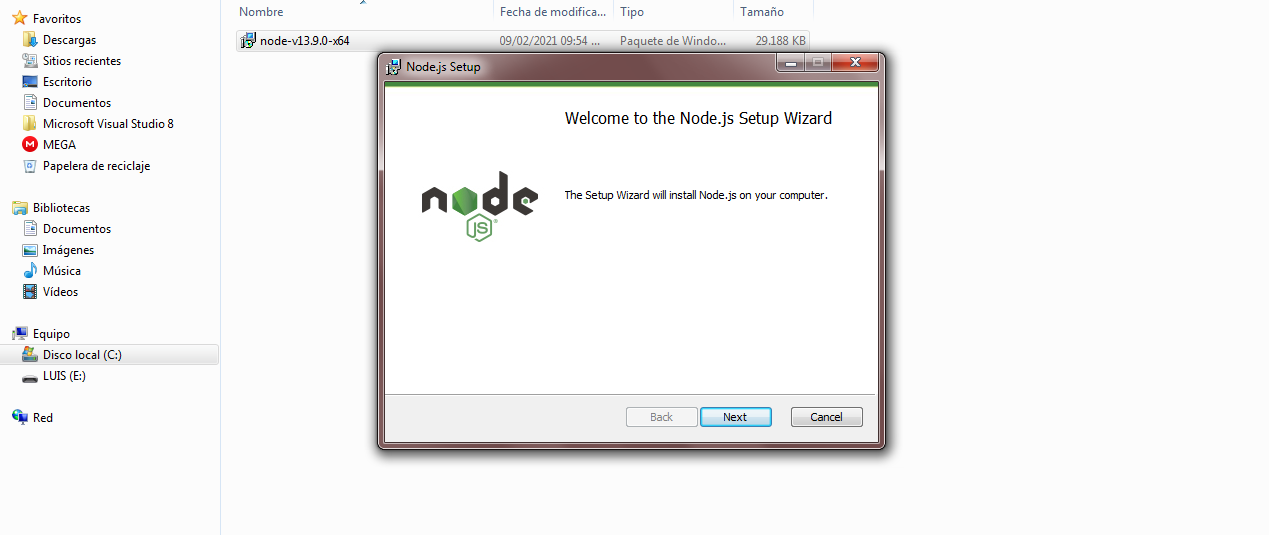
Una vez obtenido el instalador, procedemos a ubicarlo en la carpeta de descargas y ejecutarlo.



**Ilustración 2. Instalador de NodeJS en Carpeta de Descargas.**

**Fuente**: Autoría propia.

Al ejecutar el mismo, debe aparecer una imagen tal como la siguiente, una vez ubicados aquí procedemos siguiendo las instrucciones del instalador.



**Ilustración 3. Instalador de NodeJS**

**Fuente:** Autoría propia.

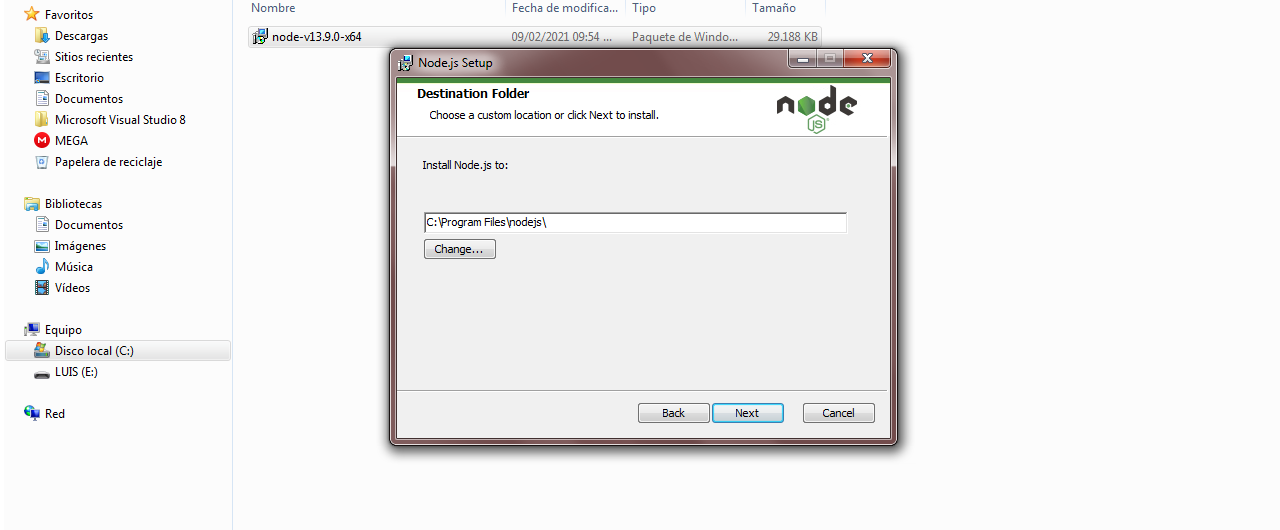
Aceptamos los términos de licencia, y damos click en siguiente



**Ilustración 4. Términos de licencia de NodeJS**

**Fuente:** Autoría propia.

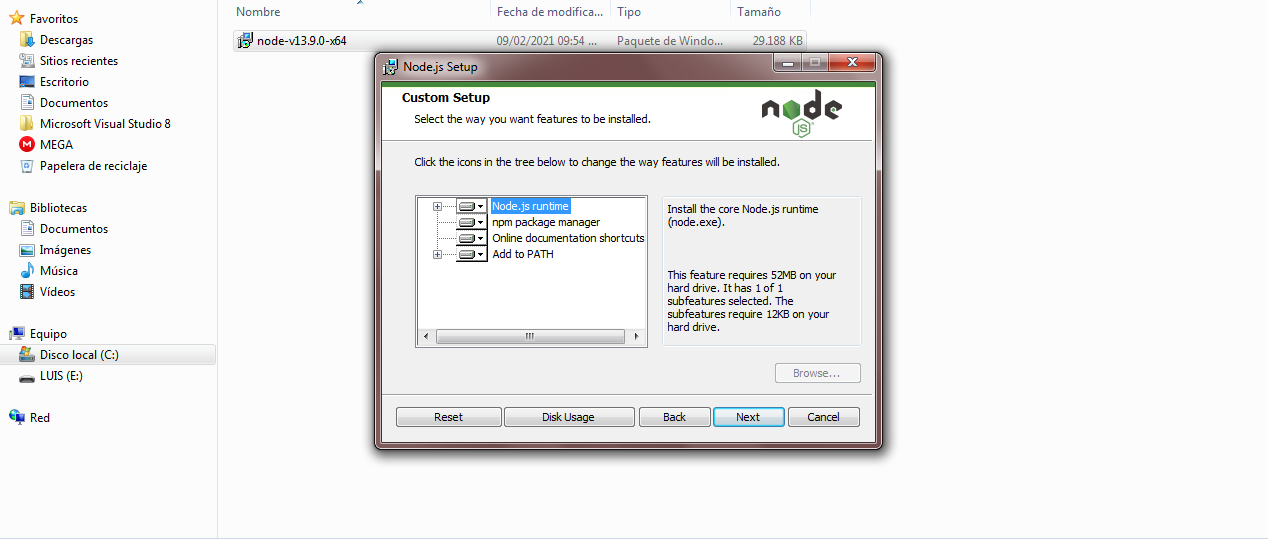
Luego, establecemos en qué dirección de nuestro equipo deseamos instalar NodeJS.



**Ilustración 5. Ubicación de NodeJS en el equipo a instalar.**

**Fuente:** Autoría propia.

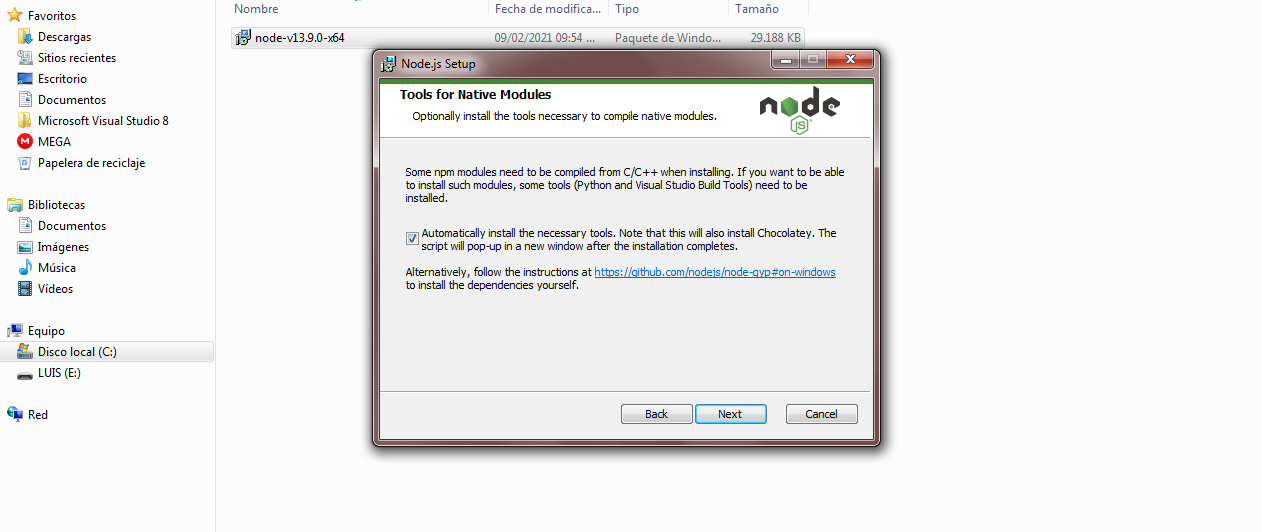
Finalmente, el instalador mostrará qué elementos instalará y qué configuraciones hará, dar click en siguiente.



**Ilustración 6. Elementos a instalar.**

**Fuente:** Autoría propia.

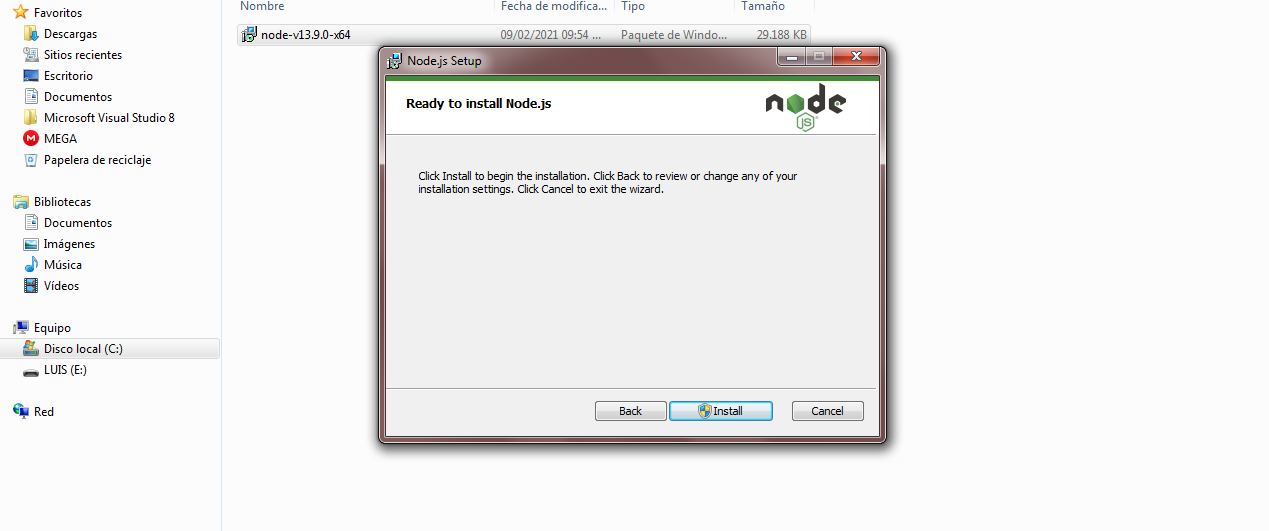
En este momento el instalador solicitará permiso para realizar la instalación del entorno NodeJS y algunos archivos extra.



**Ilustración 7. Instalación de Archivos Extra.**

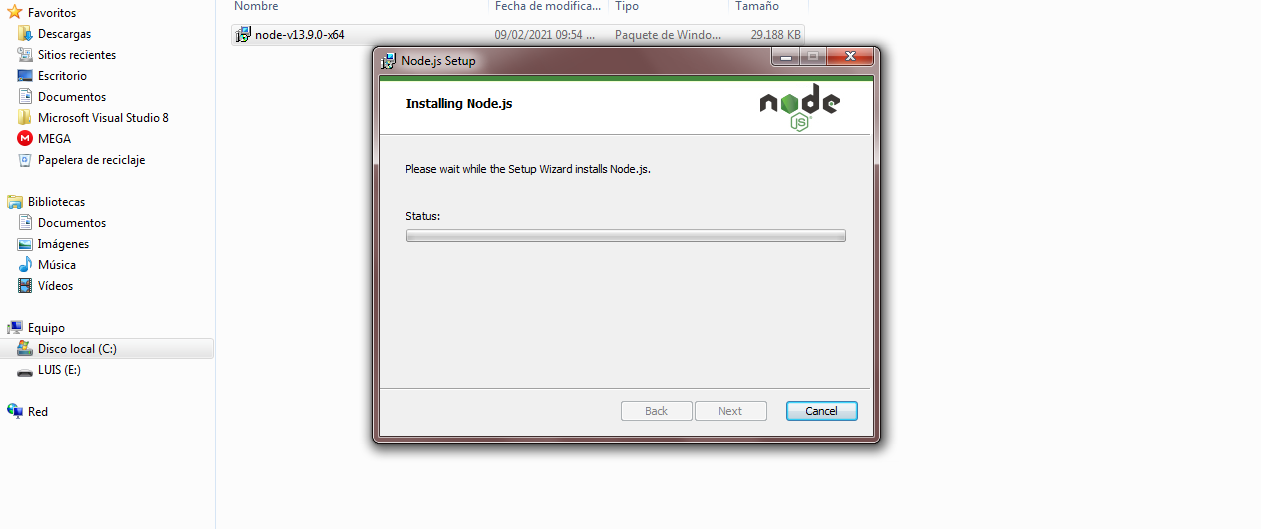
**Fuente:** Autoría propia.

Finalmente, se pedirá confirmación para instalar el entorno de ejecución.



**Ilustración 8. Permiso para instalar NodeJS**

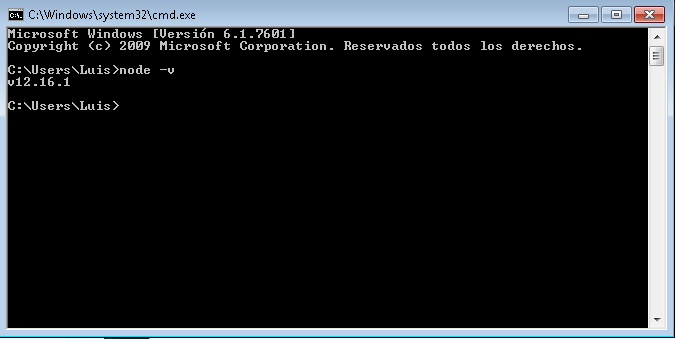
**Fuente:** Autoría propia.



**Ilustración 9. Instalación de NodeJS en el equipo.**

**Fuente:** Autoría propia.

Una vez finalizada la instalación, el instalador se cerrará y procedemos a abrir nuestra interfaz de línea de comandos y ejecutamos: **node –v**. Al hacerlo, deberíamos visualizar la versión actualmente instalada del entorno de ejecución

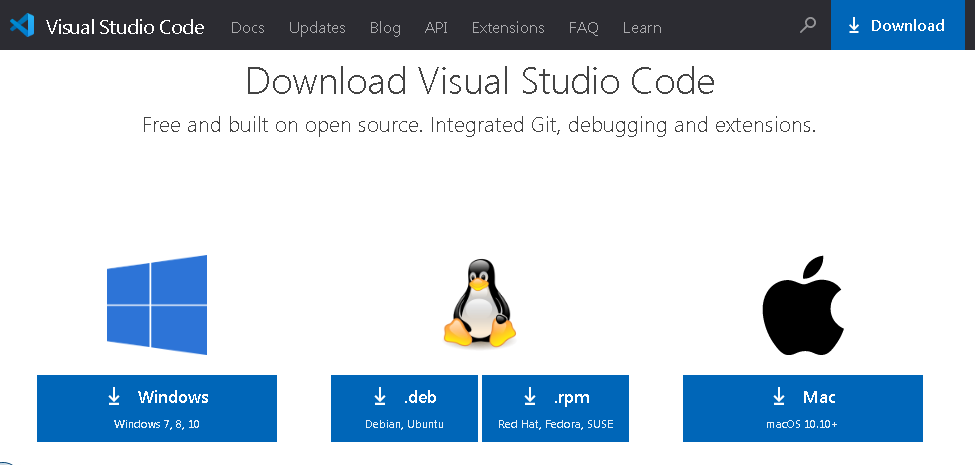


**Ilustración 10. Verificación de versión instalada de NodeJS.**

**Fuente:** Autoría propia.

## Instalación de un entorno de desarrollo y ejecución

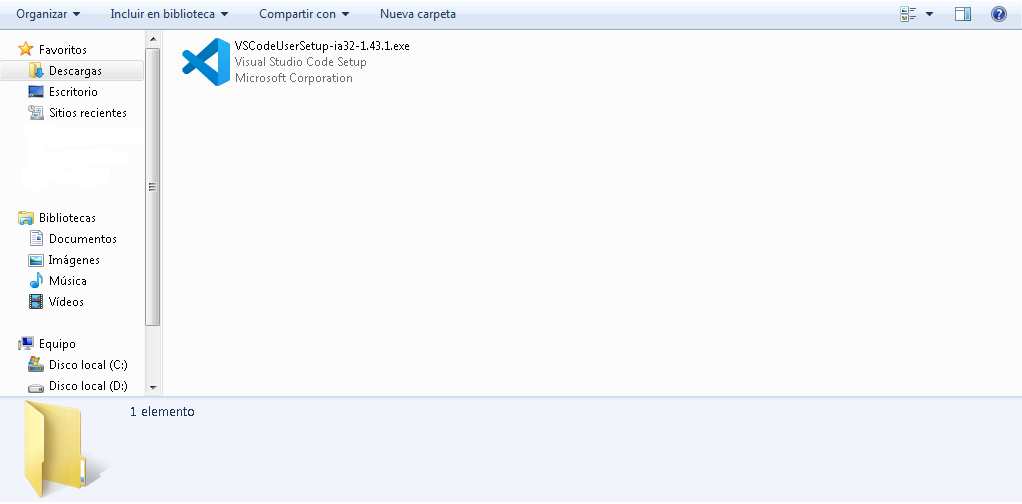
Al poder confirmar la versión de NodeJS instalada, se constata que todo está en orden para crear el servidor backend utilizando el entorno de ejecución NodeJS conjuntamente con el framework, Express, es importante recordar que también se debe contar un editor de código o entorno de desarrollo integrado que permita escribir y depurar código HTML, CSS y JavaScript, para este caso de informe se procederá a instalarse el editor de texto **Visual Studio Code** en su página oficial (https://code.visualstudio.com/download).



**Ilustración 11. Página Oficial de Descarga de Visual Studio Code**

**Fuente:** Autoría propia.

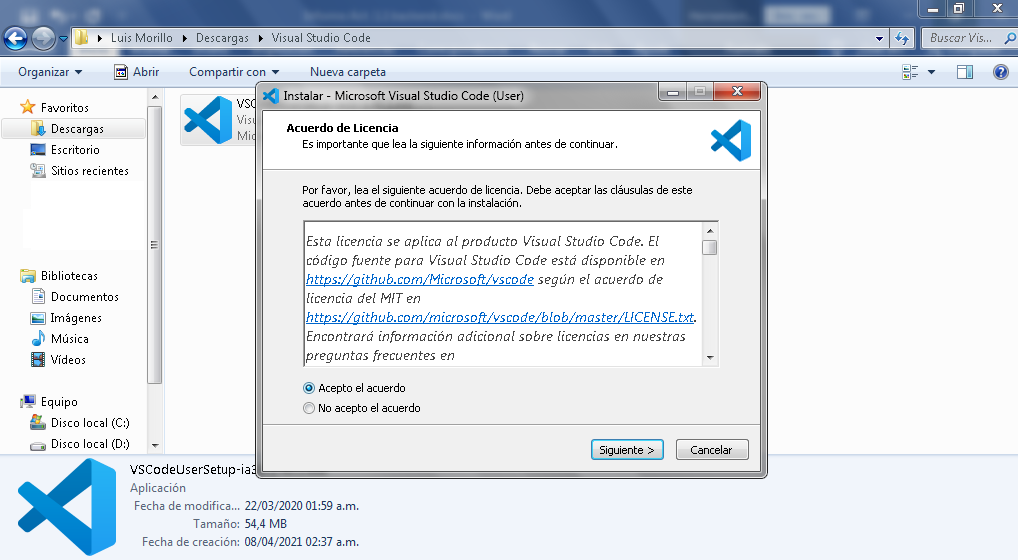
Luego de haber descargado el archivo ejecutable, almacenado en la carpeta de descarga, lo ejecutaremos para proceder a su instalación.



**Ilustración 12. Archivo de instalación de Visual Studio Code**

**Fuente:** Autoría propia.

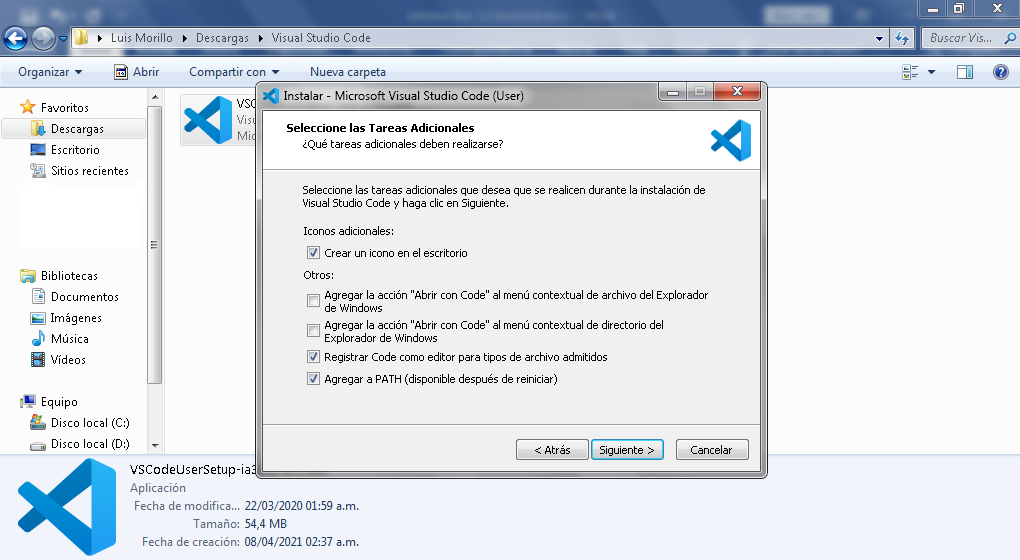
Nos abrirá una ventana de instalación, primero se deberá aceptar los términos y condiciones de uso de la misma, daremos click en “Acepto el acuerdo”.



**Ilustración 13. Acuerdo de licencia**

**Fuente:** Autoría propia.

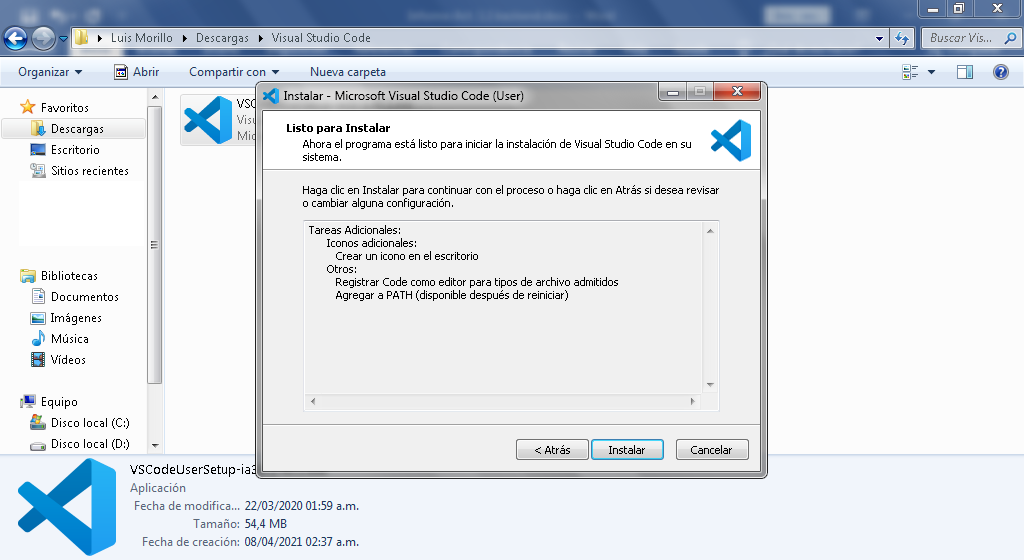
En esta parte de la instalación se debe seleccionar las tareas adicionales, es decir, crear un acceso directo desde el escritorio, agregar PATH, y ser el entorno de desarrollo por defecto.



**Ilustración 14. Agregar las tareas Adicionales**

**Fuente:** Autoría propia.

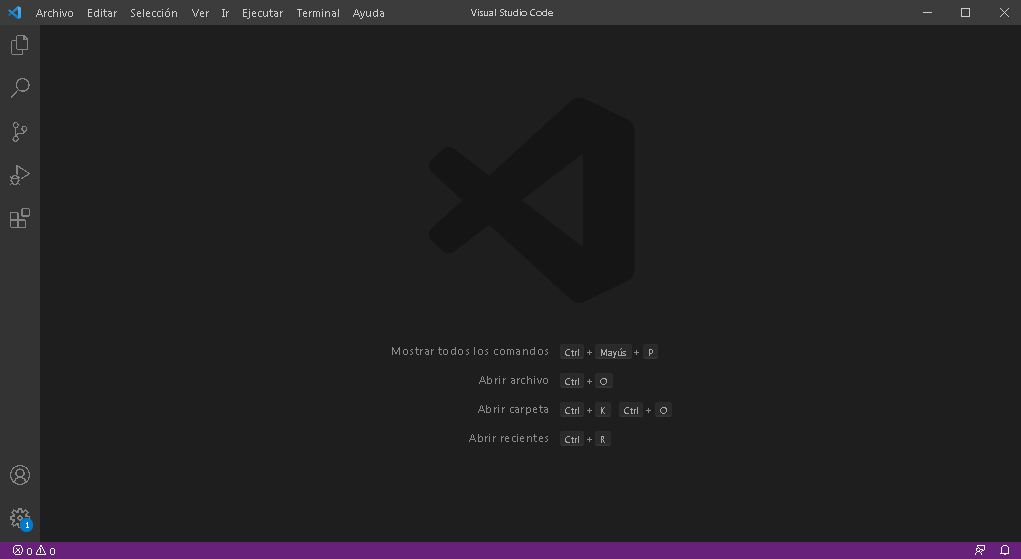
Finalmente, solamente queda en dar click en el botón “Instalar” y comenzará a instalarse el programa.



**Ilustración 15. Listo para instalar**

**Fuente:** Autoría propia.

Se creará un acceso directo en el escritorio de nuestro ordenador, y podemos ejecutar el Visual Studio Code.



**Ilustración 16. Entorno gráfico de Visual Studio Code**

**Fuente:** Autoría propia.

## **Proceso de Instalación de XAMMP**

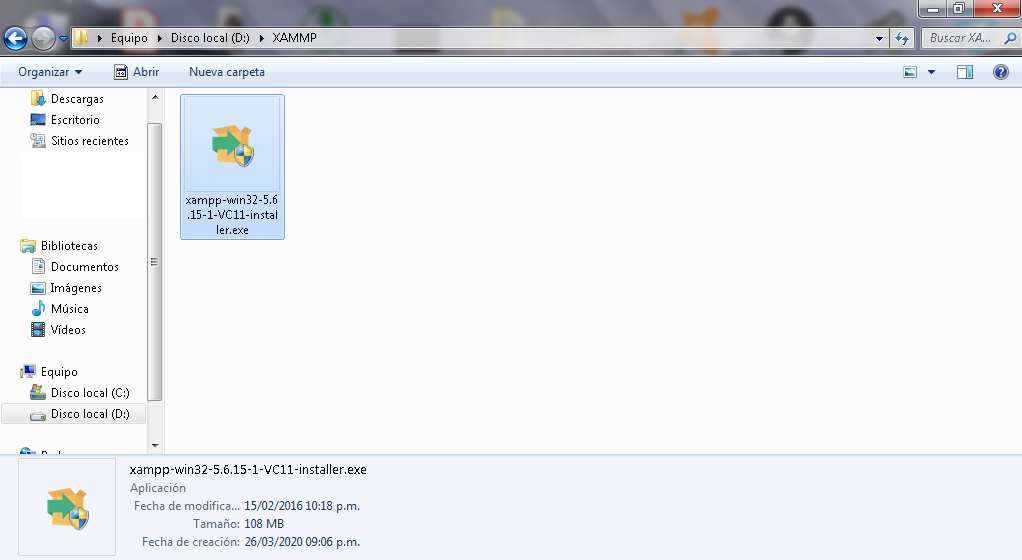
Finalizado el proceso anterior, se procede a instalarse programa **XAMMP** que es el entorno mas usado para el desarrollo con PHP, es una distribución gratuita de paquetes de apache, que contiene MariaDB, PHP y Pearl. En este caso usamos el gestor de base de datos **MySQL** con el entorno gráfico de **Phpmyadmin**. Para descargar este paquete primero debemos ingresar a su página web (<https://www.apachefriends.org/es/index.html>).



**Ilustración 17. Página Oficial de XAMPP**

**Fuente:** Autoría propia.

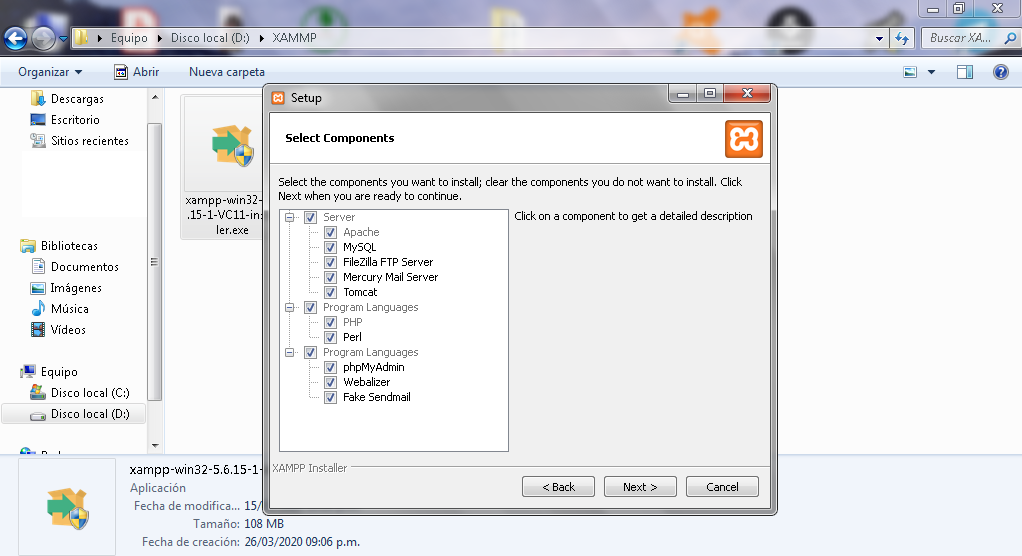
Vamos a proceder a descargar el archivo comprimido, lo descomprimimos, y lo ejecutaremos.



**Ilustración 18. Archivo para instalar XAMPP**

**Fuente:** Autoría propia.

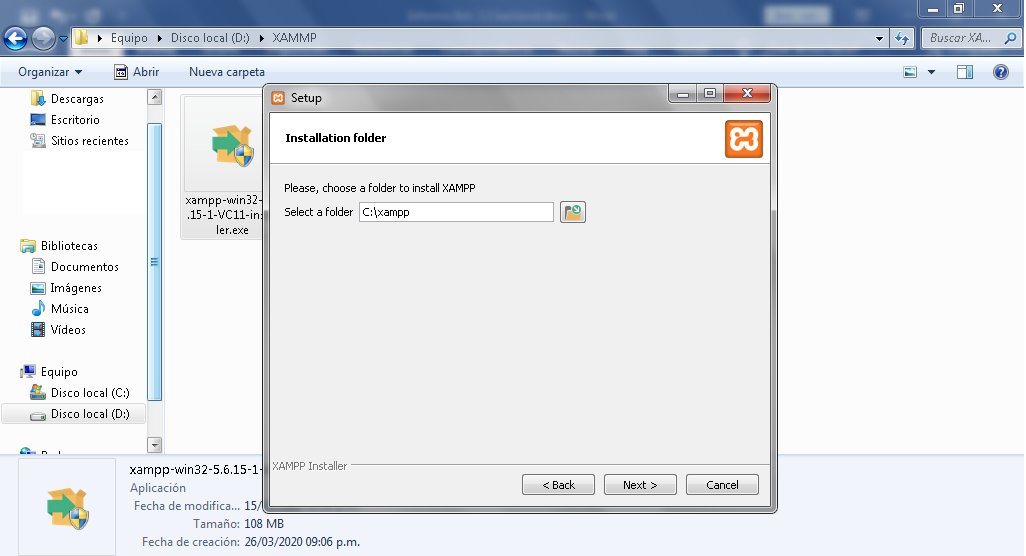
Nos tendrá que abrir una ventana de instalación y procederemos a seleccionar los componentes que queremos instalar, podemos seleccionar todas las opciones, pero por motivo de este proyecto, nos interesa **Phpmyadmin.**



**Ilustración 19. Seleccionar los componentes a instalar**

**Fuente:** Autoría propia.

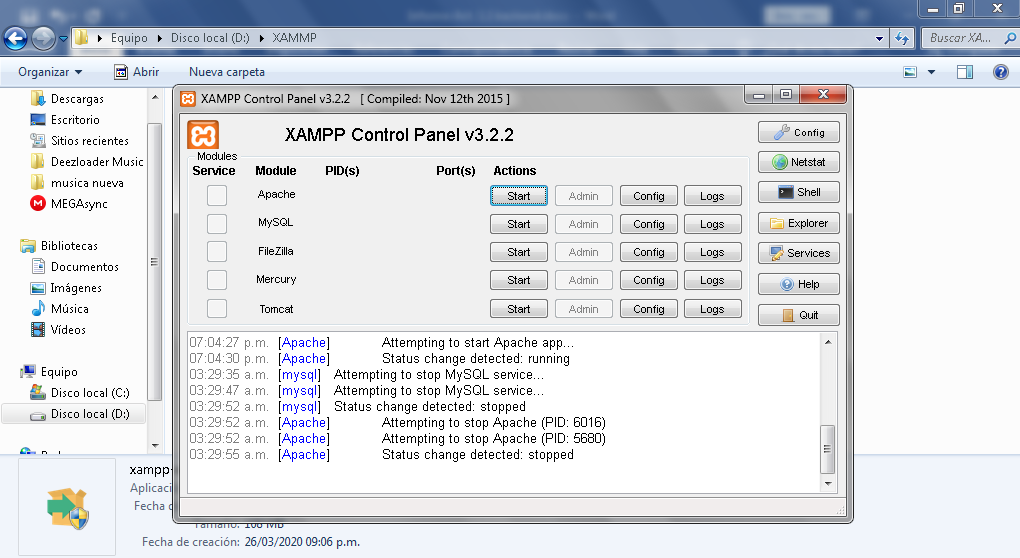
Seleccionamos la carpeta donde queremos que se instale el programa, luego daremos click en siguiente y se comenzará a instalar.



**Ilustración 20. Seleccionar la carpeta para instalar**

**Fuente:** Autoría propia.

Una vez que haya finalizado la instalación, podemos ejecutar el programa. Aparecerá una ventana con los componentes que hayamos instalado, en este caso vamos a ejecutar el servidor **MySQL**, basta con dar click en el botón “start” al lado.

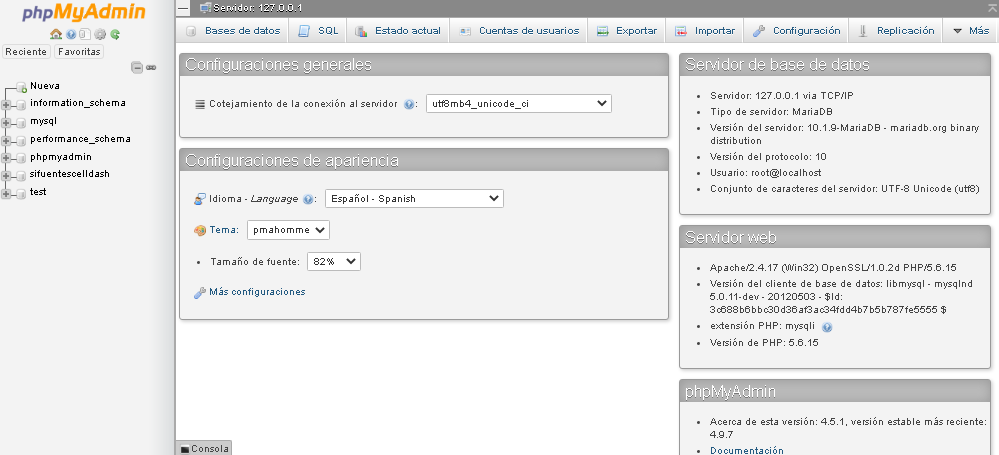


**Ilustración 21. Panel de control de XAMPP**

**Fuente:** Autoría propia.

# **Pasos para la creación de la base de datos MySQL, usando Phpmyadmin.**

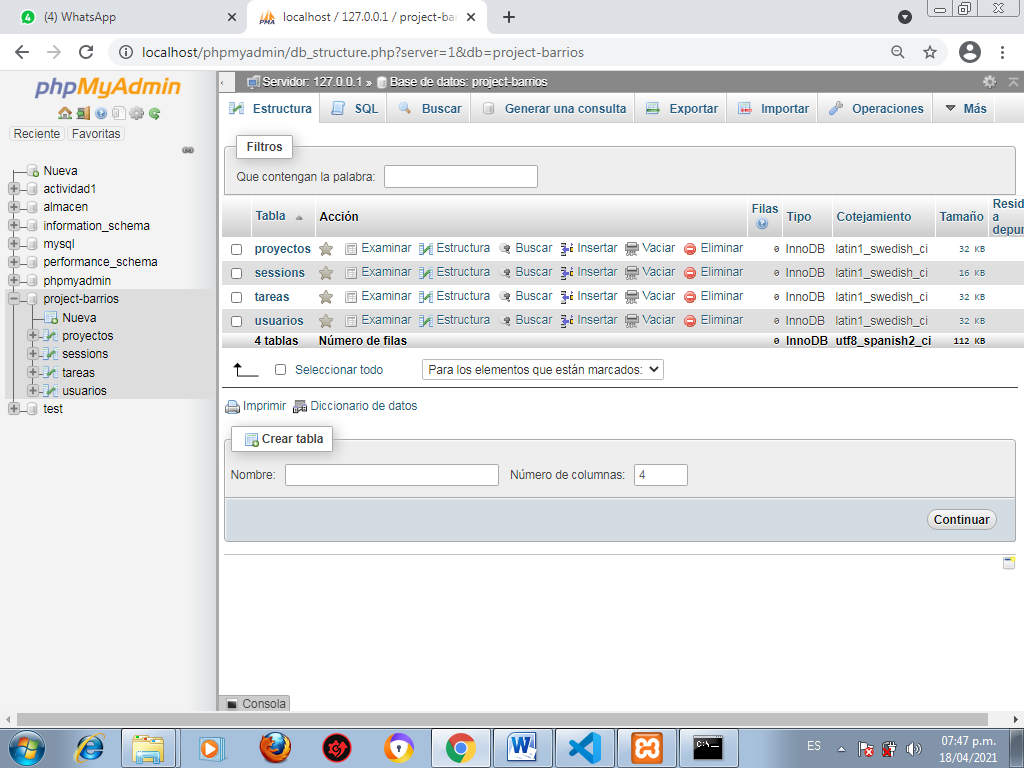
Haciendo uso de **MySQL** mediante **Phpmyadmin** es necesario acceder a su entono gráfico, podemos realizar esta acción escribiendo en el navegador (<http://localhost/phpmyadmin>).



**Ilustración 22. Página principal de Phpmyadmin**

**Fuente:** Autoría propia.

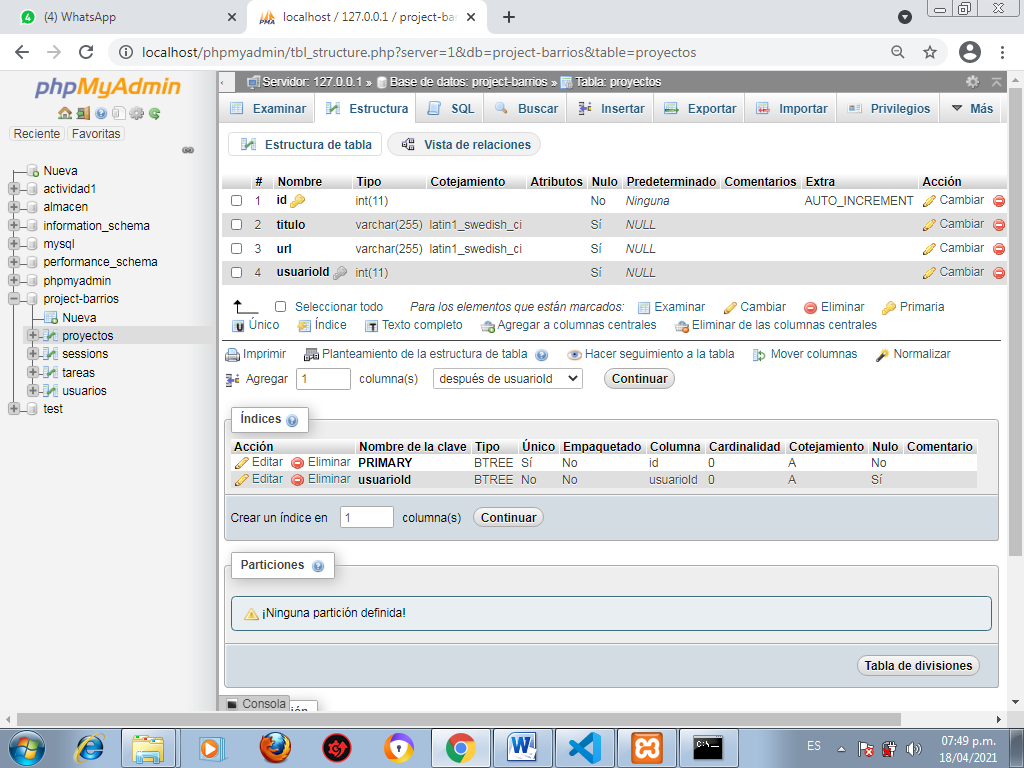
Por motivo de este proyecto la base de datos se llamará “**project-barrios”**, se crearon 4 Tablas, en el cual son: Proyetos, Sessions, tareas y usuarios.



**Ilustración 23. Base de Datos y sus campos**

**Fuente:** Autoría propia.

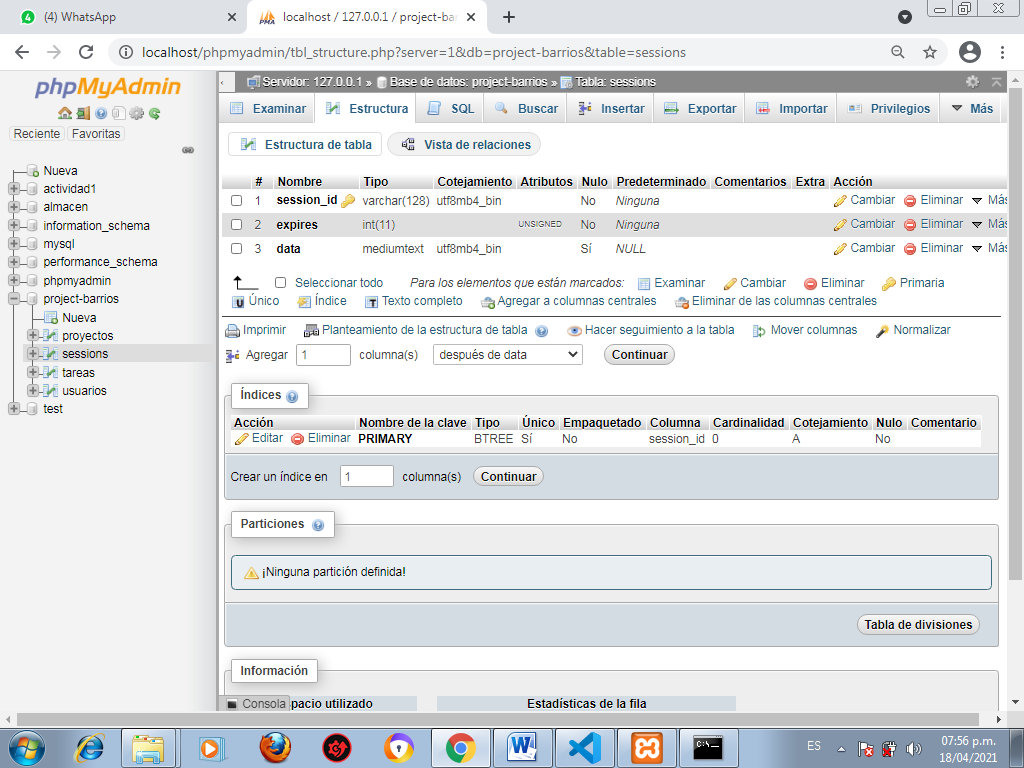
En la tabla “Proyetos” tenemos las celdas (“id, título, url y usuarioid”), aquí guardaremos los datos de los Proyectos que queremos registrar en nuestra base de datos.



**Ilustración 24. Tabla Proyectos**

**Fuente:** Autoría propia.

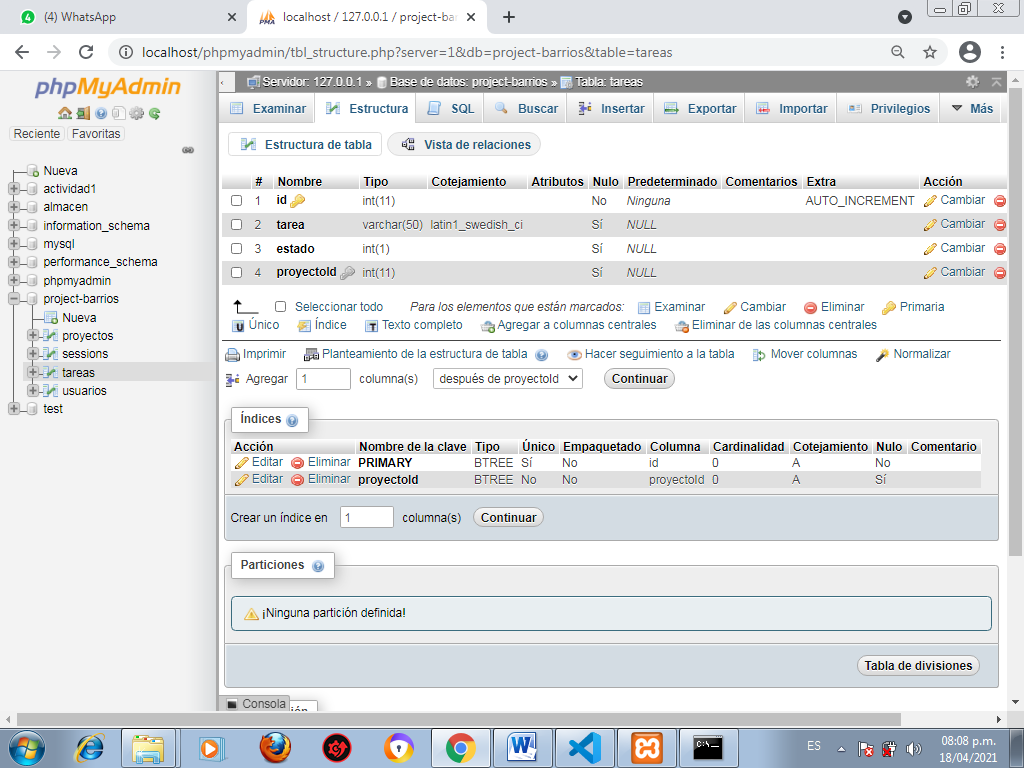
En la tabla “Sessions” tenemos las celdas (“sesión\_id, expire y data”). Podremos colocar tales datos del proyecto.



**Ilustración 25. Tablas Sessions**

**Fuente:** Autoría propia.

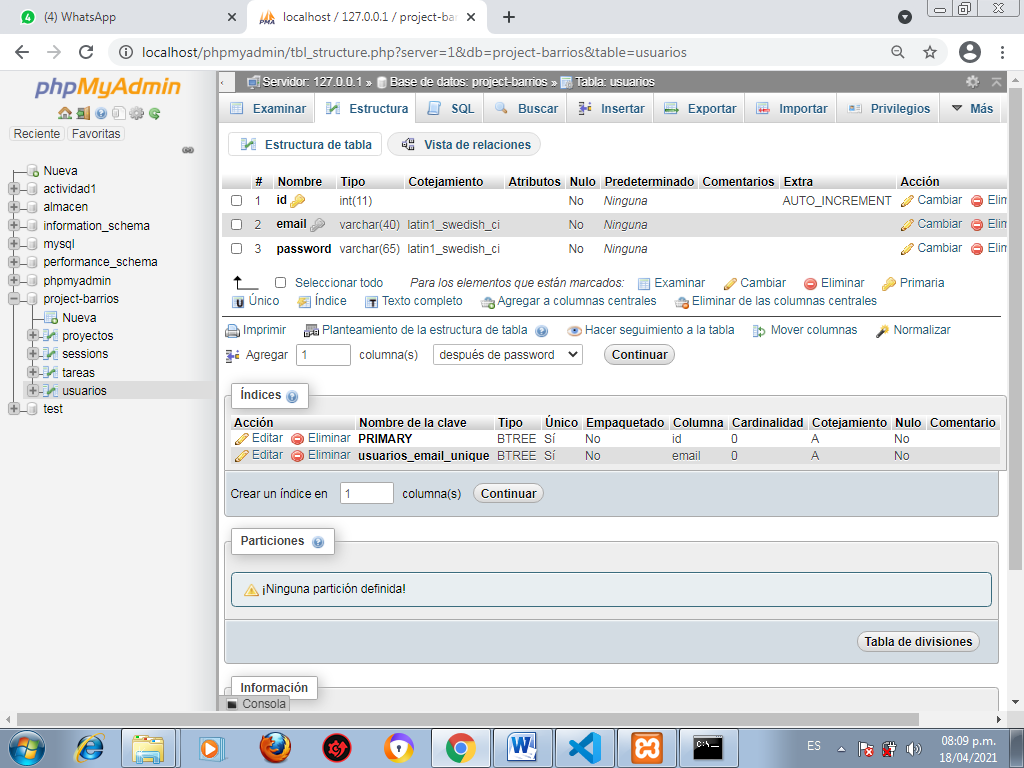
En la tabla “Tareas” tenemos las celdas (“id, estado, proyectoid, y tarea”). En esta tabla registraremos los datos de cada tarea.



**Ilustración 26. Tabla tareas**

**Fuente:** Autoría propia.

En la tabla “usuarios” tenemos las celdas (“id, email, password”). En esta tabla podemos registrar a los encargados de usar la base de datos, se debe crear un nombre de usuario y una contraseña.



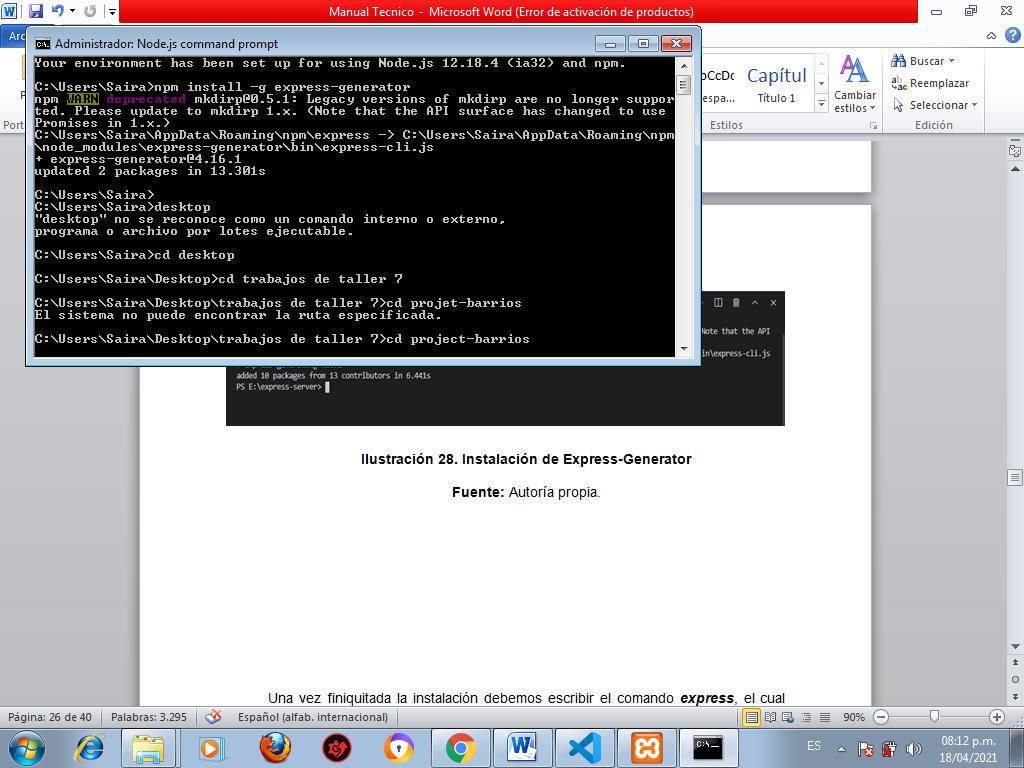
**Ilustración 27. Tabla usuarios**

**Fuente:** Autoría propia.

**Creando un servidor backend con NodeJS y Express**

Una vez instalado NodeJS en el computador, se debe crear una carpeta donde se alojarán los archivos correspondientes al servidor backend, para casos del presente informe esta carpeta se llamará **proyecto**, ahora es necesario instalar la librería Express, la cual arma una abstracción sobre NodeJS que permite crear funcionalidades propias de un servidor backend de una manera más intuitiva y fácil.

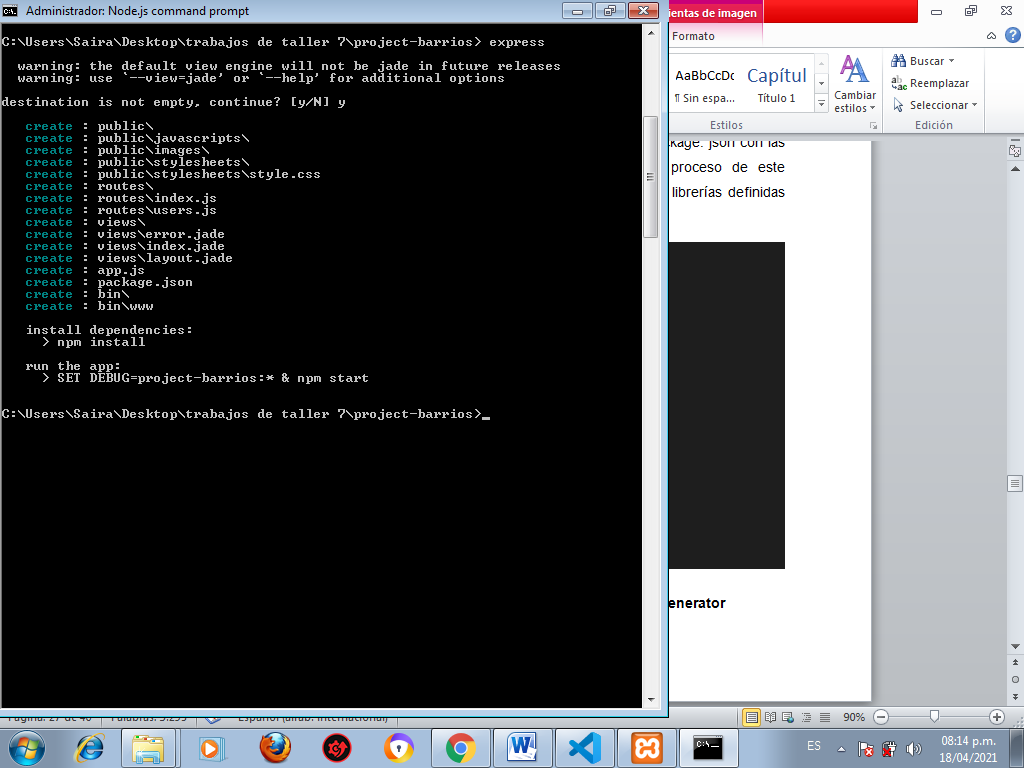
Para instalar está librería, entrar a la carpeta **express-server** utilizando la interfaz de línea de comandos y ejecutar el siguiente comando: ***npm install -g express-generator,*** esto instalará un generador de estructura para el servidor backend, destacando que la directiva npm install, le indica al gestor de paquetes de Node, NPM, instalada durante la instalación de NodeJS, instalar la librería de generación de forma global.



**Ilustración 28. Instalación de Express-Generator**

**Fuente:** Autoría propia.

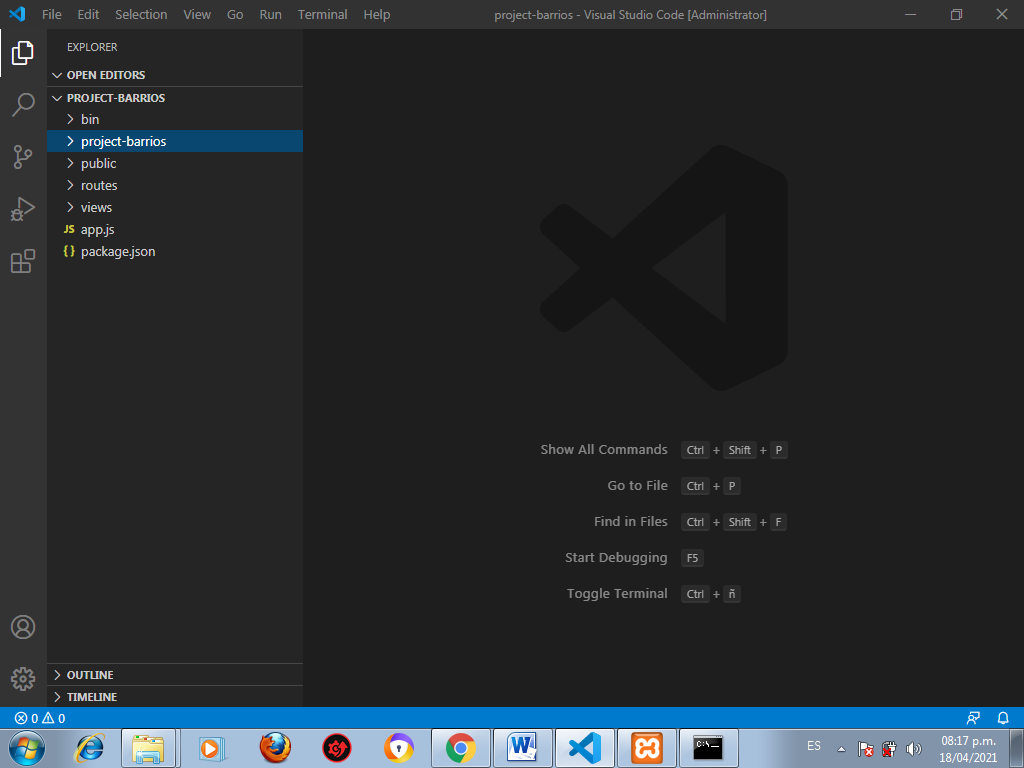
Una vez finiquitada la instalación debemos escribir el comando ***express***, el cual se encargará de crear la estructura de carpetas y crear el archivo package. json con las dependencias para el servidor backend, una vez terminado el proceso de este comando, debemos ejecutar ***npm install***, encargado de instalar las librerías definidas en package. json.



**Ilustración 29. Estructura generada por Express-generator**

**Fuente:** Autoría propia.

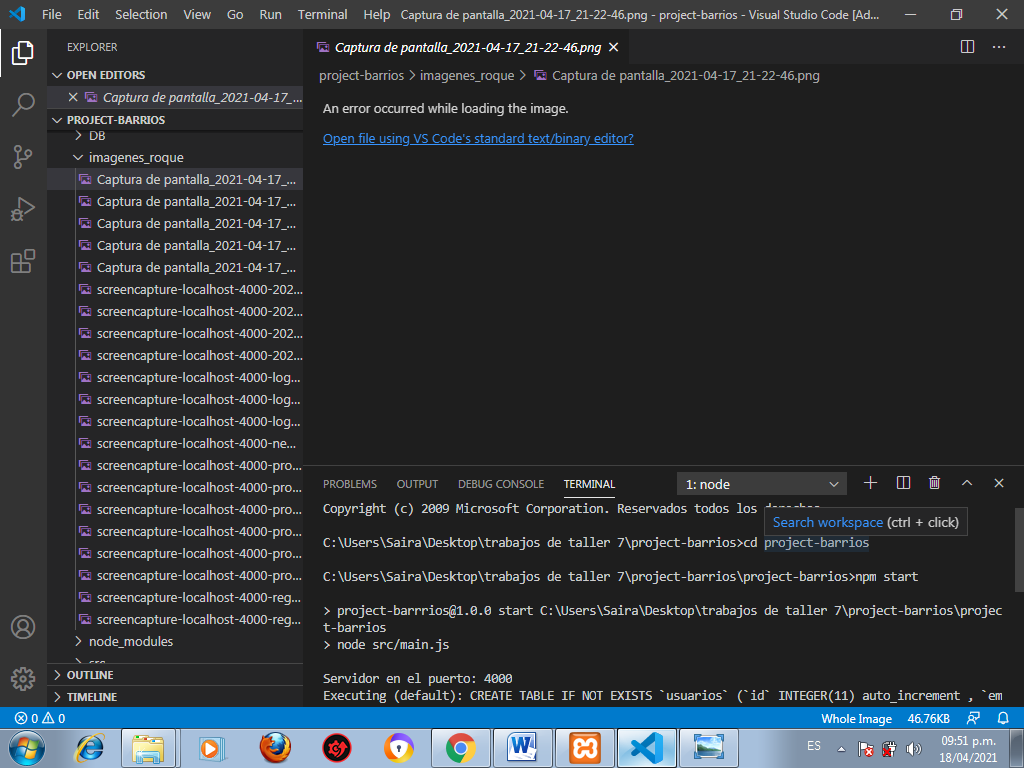
Las carpetas de nuestro proyecto son las que se muestran a continuación, como podemos apreciar tenemos una carpeta para la base de datos, el routes, views, el archivo app.js y el código correspondiente para hacer la conexión a la base de datos denominada conexión.js.



**Ilustración 30. Carpetas de nuestro proyecto**

**Fuente:** Autoría propia.

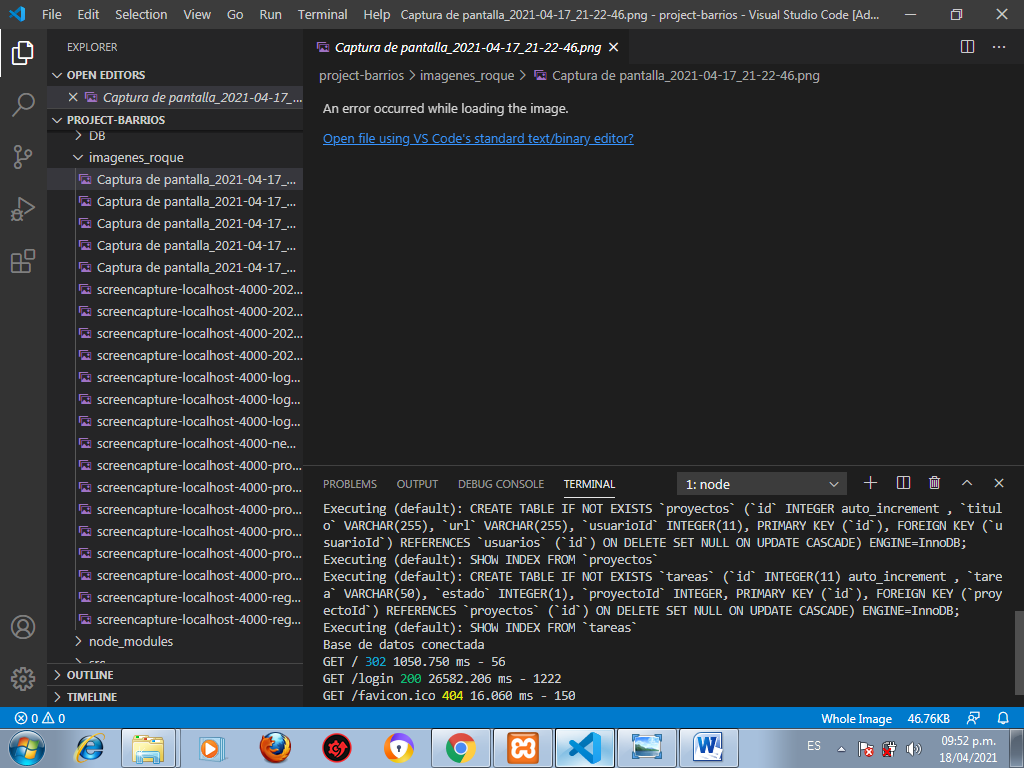
Abrimos el terminal de desarrollo y procederemos a ejecutar el proyecto, cabe destacar que para que todos los módulos sean ejecutados de buena manera, es necesario primero ejecutar la base de datos MySQL mediante XAMPP y luego escribir y ejecutar en el terminal (en este caso) de Visual Studio Code, la línea **npm i y luego npm start**.



**Ilustración 31. Ejecución del Proyecto**

**Fuente:** Autoría propia.

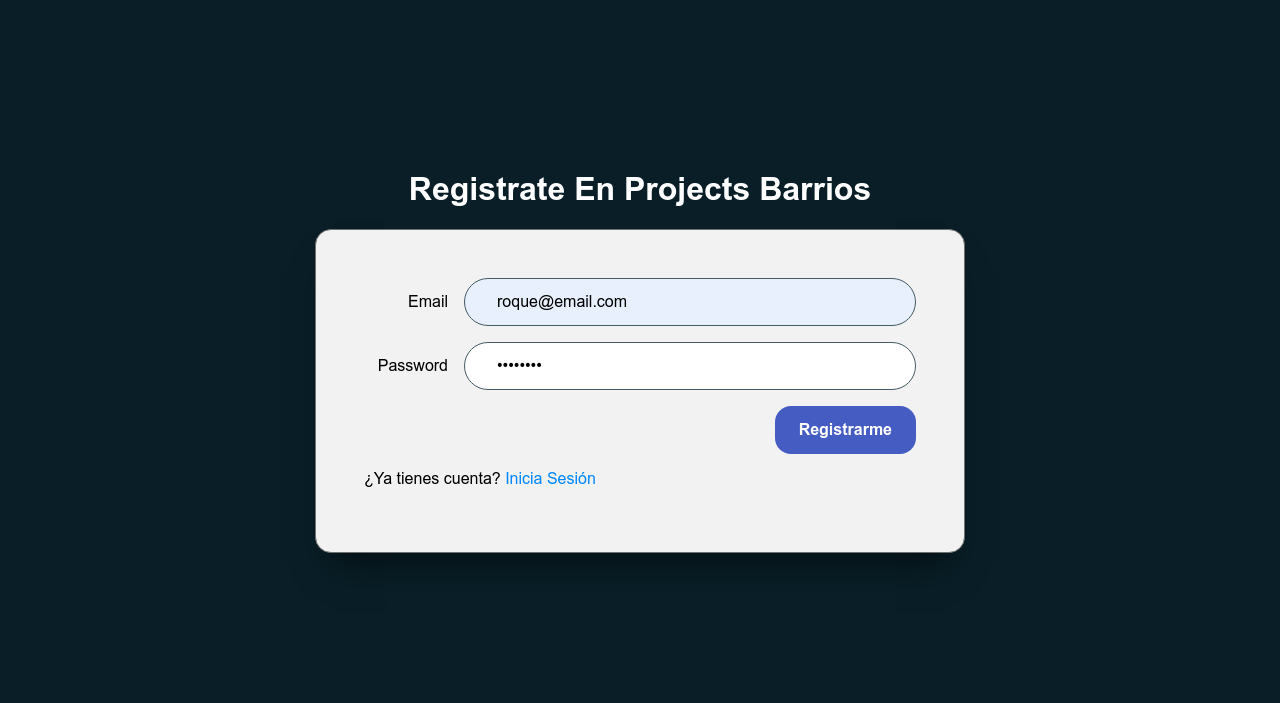
Tendremos los mensajes de que se ha conectado al base de datos y el código se ha escrito correctamente, en caso de esto no se cierto, en la consola habrá un mensaje que lo indique.



**Ilustración 32. Correcto funcionamiento de la base de datos y el código**

**Fuente:** Autoría propia.

La base de datos ha sido conectada con éxito, ahora solo nos queda por entrar al navegador y colocar en la URL **http://localhost:4000,** pulsamos la tecla enter y podremos ver la pantalla de registro, en donde colocaremos los que nos piden daremos click al botón **Registrarse**.



**Ilustración 33. Registrarse**

**Fuente:** Autoría propia.

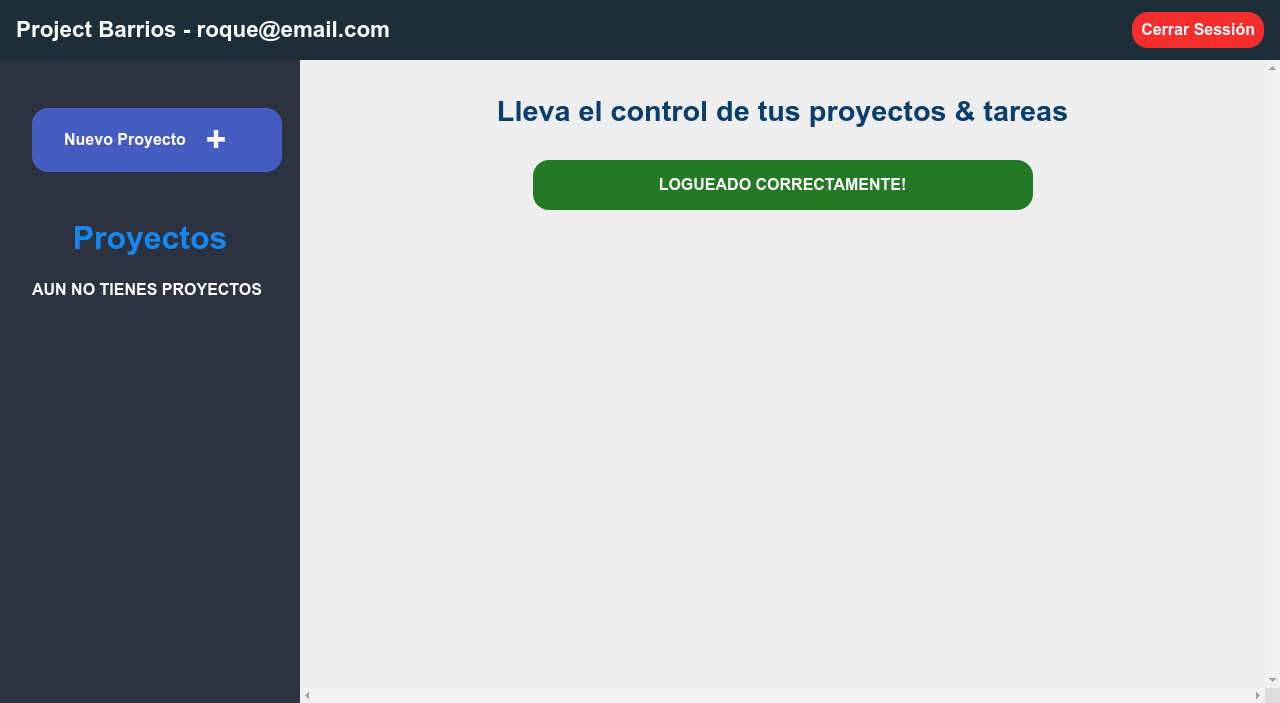
Luego de registrarse Inicias sesión para entrar a tu cuenta en project-barrios



**Ilustración 34. Iniciar sesión**

**Fuente:** Autoría propia.

Una vez completado el registro, podemos dar click en **login** o en la URL para entrar con nuestro nombre y usuario previamente registrado te llevara a la página principal.



**Ilustración 35. Página Principal**

**Fuente:** Autoría propia.

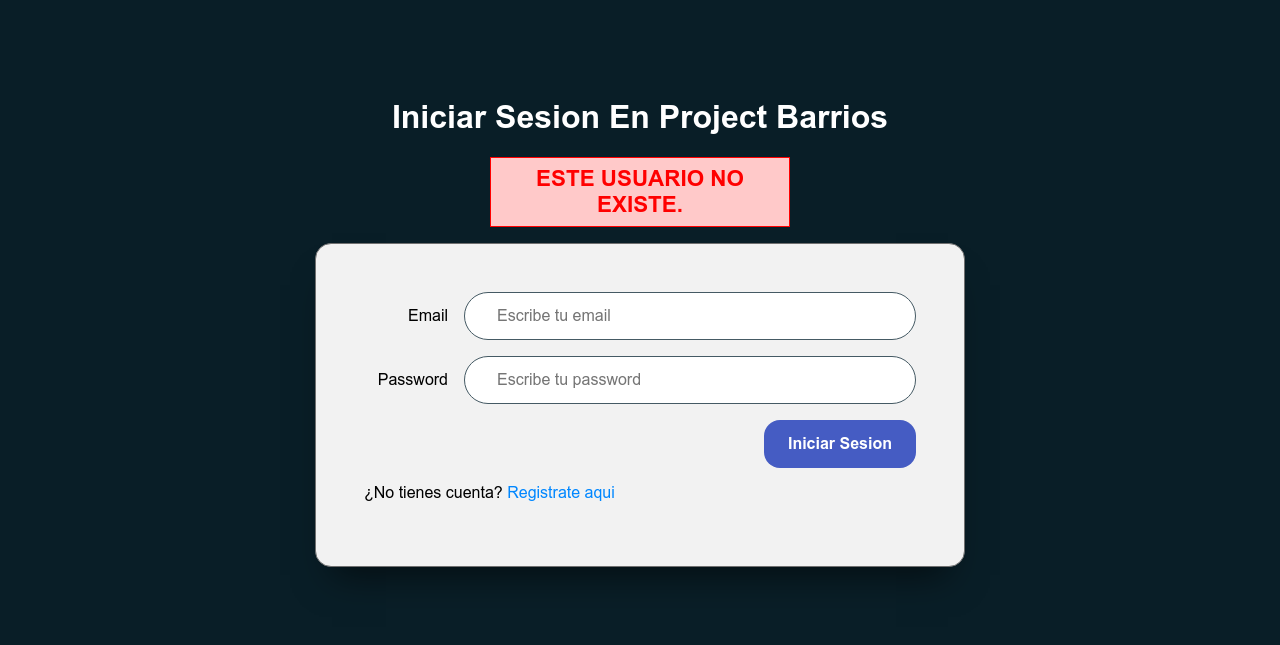
Luego de Entrar a tu cuenta te aparece la página principal donde te sale un mensaje que dice que aún no tienes proyectos



**Ilustración 36. Proyecto Creado**

**Fuente:** Autoría propia.

Cuando inicias sesión con un usuario o contraseña incorrecta te envía estos mensajes ilustrados.





**Ilustración 37. Inicios incorrectos**

**Fuente:** Autoría propia.

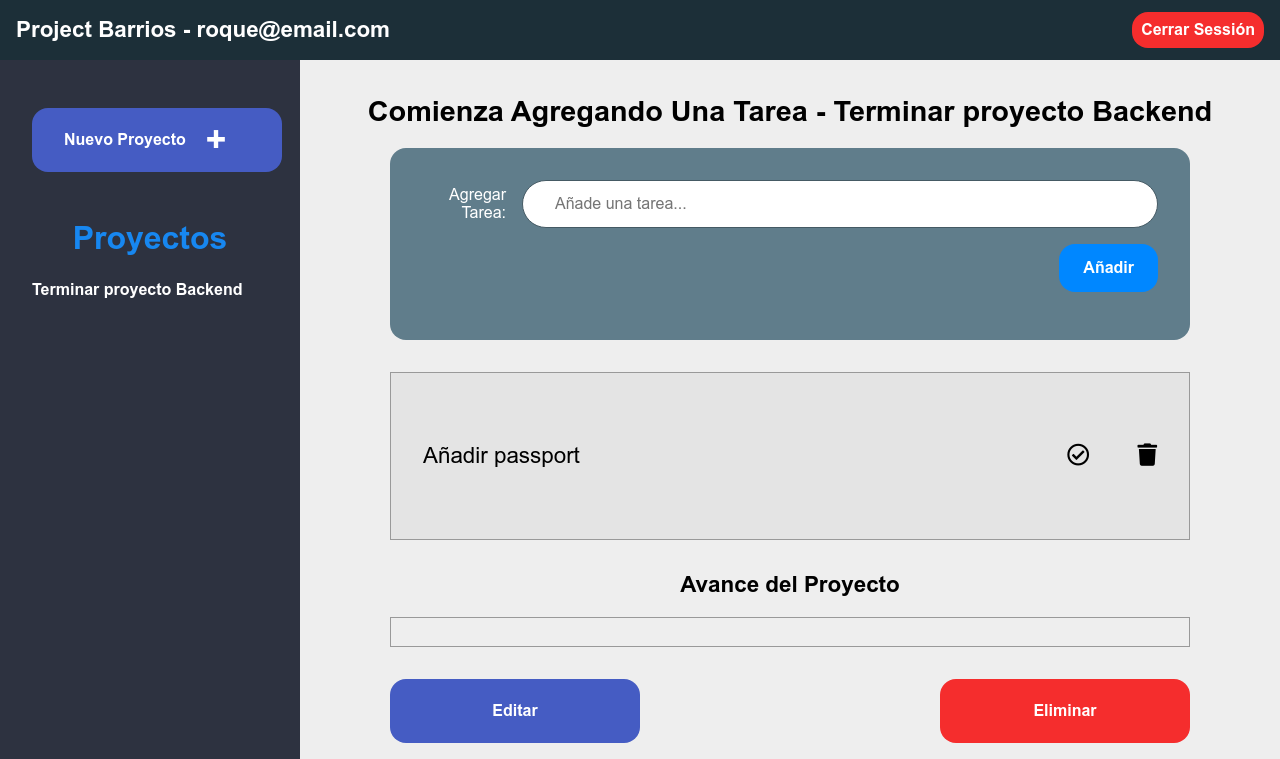
Si quieres editar o actualizar un proyecto solo tienes que dar click en actualizar como lo muestra la siguiente imagen



**Ilustración 38. Actualizar**

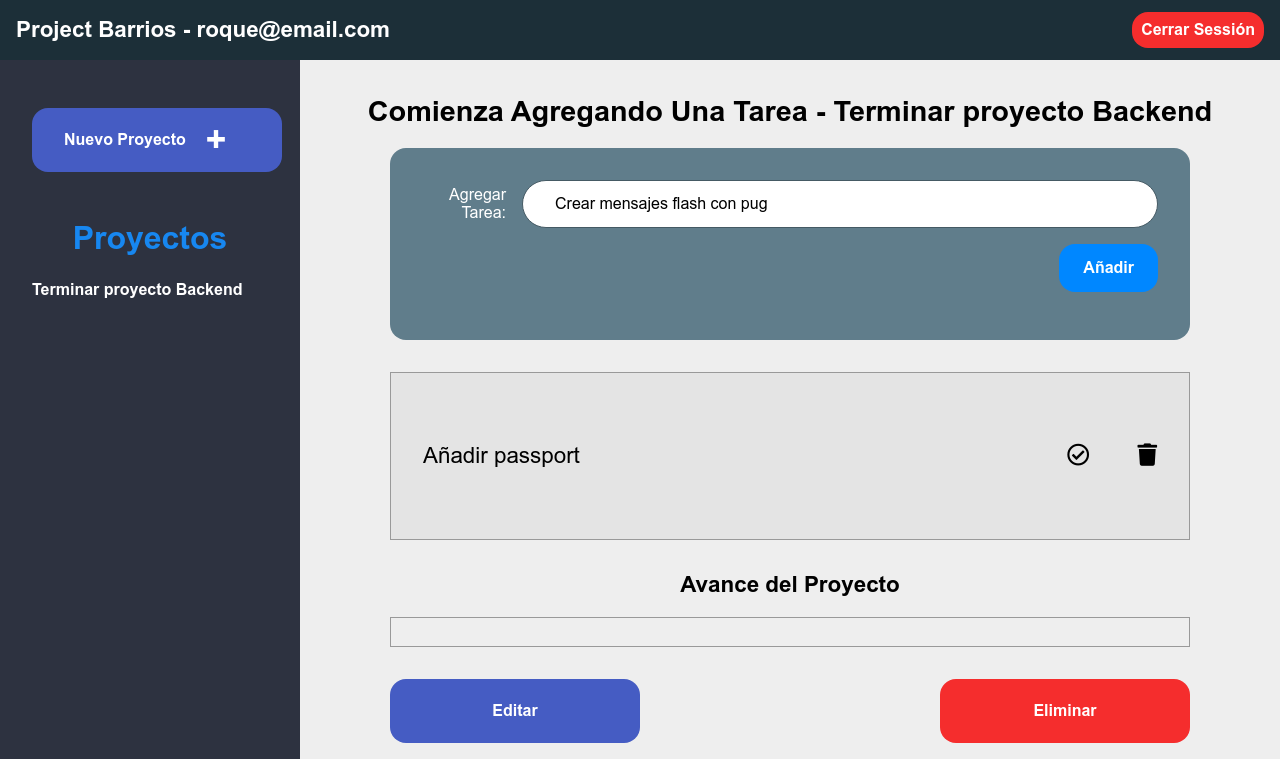
**Fuente:** Autoría propia.

Para añadir una tarea a ese proyecto solo le das click a uno de los proyectos.



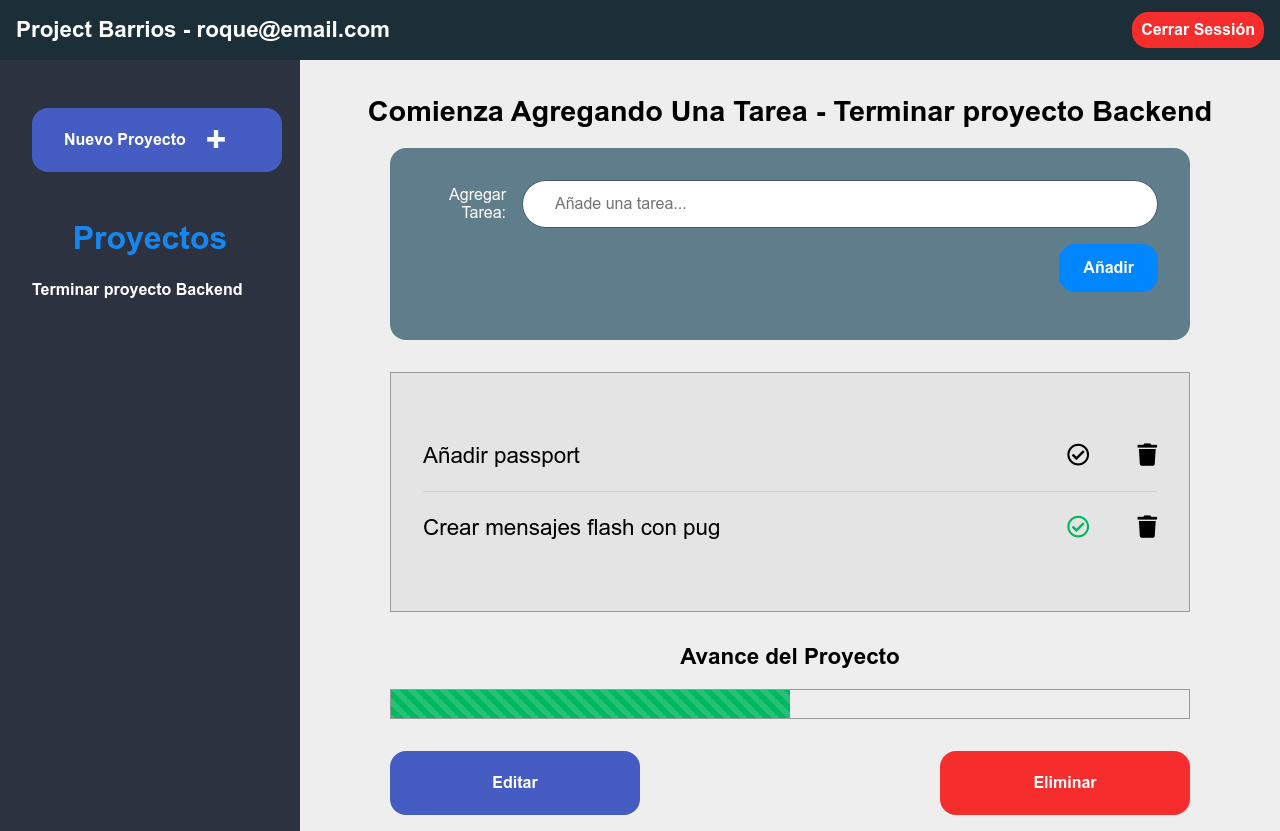
**Ilustración 39. Añadir tarea**

**Fuente:** Autoría propia.



**Ilustración 40. Añadir tarea**

**Fuente:** Autoría propia.



**Ilustración 41. Añadir tarea**

**Fuente:** Autoría propia.

**Diagrama de flujo**

¿Estas registrado?

No

Registrarse

Inicio

Si

Pantalla principal

Iniciar sesión

Añadir tarea

Actualizar proyecto

Añadir proyeto

Agregar proyecto

Fin

Cerrar sesión

**Conclusión**

Aprovechar JavaScript para hacer servidores backend con ayuda de NodeJS es provechoso, es bueno aprender y saber manejar este lenguaje ya que conocemos la forma de crear servidores, este lenguaje se puede usar tanto para el desarrollador Frontend como para el desarrollador backend, en un mismo lenguaje de programación, trabajando en conjunto con el mismo se encuentran las soluciones a los problemas, con el uso de Express se permite crear servidores backend de un forma más entendible y reduciendo las líneas de código para su creación .

NodeJS debe su notoriedad en gran parte a esto, es fácil de usar y para quienes emprenden una ruta autoeducada de aprendizaje en desarrollo web, les brinda la posibilidad de poder construir un aplicativo integral considerando tanto la lógica del negocio como la visualización de la interfaz, NodeJS, si bien contiene módulos para gestionar plantillas, tiene un mejor uso al limitarse a exponer servicios web mediante una API REST, con este mismo orden de ideas, si bien NodeJS posee documentación suficiente en su documentación oficial para crear un servidor HTTP sin agregar librerías o dependencias, utilizar una herramienta como *Express*, ahorra tiempo dinero y crea una capa de abstracción sobre conceptos no tan importantes, lo cual permite al programador centrarse en lo más importante: Crear valor para quién trabaja.

Ahora se está en capacidad para crear servidores web que pueden gestionar una variedad de solicitudes y respuestas. Con este conocimiento, puedo crear un servidor que devuelva muchas páginas HTML al usuario en diferentes puntos finales, comprender los principales beneficios de Express y Node, y más o menos cómo se verían las partes principales de una aplicación Express (rutas, middleware, manejo de errores y plantillas).

REFERENCIAS

*NodeJS - Node Foundation*. (08 de abril de 2021). Obtenido de https://nodejs.dev/

*Visual Studio Code EcuRed (08 de abril de 2021). Obtenido por: https://www.ecured.cu/Visual\_Studio\_Code*

*Lenguaje de Programación más recomendables (08 de abril de 2021) Obtenido por: https://indexdesarrollo.com/lenguajes-de-programacion-web-mas-recomendables/*

*¿Por qué usar Node.JS? (08 de abril de 2021). Obtenido por: https://openwebinars.net/blog/por-que-usar-nodejs/*

*Página Oficial de Visual Studio Code (08 de abril de 2021). Obtenido por: https://code.visualstudio.com*

*Página Oficial de XAMPP (08 de abril de 2021). Obtenido por:*

[*https://www.apachefriends.org/es/index.html*](https://www.apachefriends.org/es/index.html)