**1. Introducción**

**Descripción del Proyecto**

Este proyecto es una API CRUD desarrollada en Node.js con el framework Express. Permite gestionar usuarios mediante operaciones de crear, leer, actualizar y eliminar, con validación de datos como la unicidad de correo electrónico y nombre de usuario, y almacenamiento seguro de contraseñas. Los datos se guardan en una base de datos MySQL.

**Tecnologías Usadas**

* Node.js y Express: Framework y entorno para el desarrollo de la API.
* MySQL: Base de datos relacional para almacenar los datos de los usuarios.
* Postman: Herramienta utilizada para probar los endpoints de la API.
* npm: Administrador de paquetes de Node.js para gestionar dependencias.

**2. Requisitos Previos**

Antes de ejecutar el proyecto, asegúrate de tener instalados y configurados los siguientes requisitos:

* **Node.js (versión 20 o superior)**
  + Descarga e instala Node.js desde [nodejs.org](https://nodejs.org/). Al instalar Node.js, también se incluye npm, que se utilizará para gestionar las dependencias del proyecto.
* **XAMPP**
  + XAMPP proporciona una solución para instalar y ejecutar **MySQL** junto con **phpMyAdmin** para la administración de la base de datos. Puedes descargar XAMPP desde [apachefriends.org](https://www.apachefriends.org/). Utiliza MySQL de XAMPP para crear y gestionar la base de datos del proyecto.
* **phpMyAdmin (incluido en XAMPP)**
  + phpMyAdmin, incluido en XAMPP, es una herramienta visual para administrar bases de datos MySQL. Puedes acceder a phpMyAdmin desde <http://localhost/phpmyadmin> una vez que XAMPP esté en ejecución.
* **Postman**
  + Utiliza Postman para probar los endpoints de la API. Es una herramienta gráfica que facilita la creación y ejecución de solicitudes HTTP hacia la API.

**3. Instalación y Configuración del Entorno**

Sigue los pasos a continuación para configurar el entorno de desarrollo y ejecutar el proyecto:

* **Clonar el Repositorio desde GitHub:**

Primero, necesitas clonar el proyecto desde el repositorio de GitHub en tu máquina local.

Link: <https://github.com/Danie0822/Node-Api.git>

* **Configurar el Entorno (.env)**

El proyecto incluye un archivo .env. para configurar tu entorno, abre el archivo .env y configura los parámetros de la base de datos (por ejemplo, MySQL) y otros ajustes del entorno.

Texto

Descripción generada automáticamente

* **Iniciar el Servidor Local**

Finalmente, para iniciar el servidor local de Laravel y poder acceder a la API desde Postman, ejecuta el siguiente comando:



4. **Ejemplos de Uso con Herramientas como Postman**

A continuación, se muestran ejemplos de cómo interactuar con la API para realizar operaciones CRUD sobre los usuarios.

**Lista usuarios**

* **Postman**:
  + Método: **GET**
  + API Key : (API Key del .env)
  + URL: <http://localhost:4000/api/usuario>

Texto

Descripción generada automáticamente

**Crear un Usuario**

* **Postman**:
  + Método: **POST**
  + API Key : (API Key del .env)
  + URL: <http://localhost:4000/api/usuario/save>

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

**Obtener un Usuario por ID**

* **Postman**:
  + Método: **GET**
  + API Key : (API Key del .env)
  + URL: <http://localhost:4000/api/usuario/1>
  + Sustituye {id} por el ID del usuario que deseas obtener, por ejemplo.

Texto

Descripción generada automáticamente

**Actualizar un Usuario**

* **Postman**:
  + Método: **PUT**
  + API Key : (API Key del .env)
  + URL: <http://localhost:4000/api/usuario/update>

Imagen que contiene Texto

Descripción generada automáticamente

**Eliminar un Usuario**

* **Postman**:
  + Método: **DELETE**
  + API Key : (API Key del .env)
  + URL: <http://localhost:4000/api/usuario/delete/1>

Sustituye {id} por el ID del usuario que deseas eliminar, por ejemplo.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

**Notas Finales**

**¡Importante!** No olvides iniciar la base de datos que se incluye en el repositorio bajo el nombre DBPrueba.