

Ejemplo de API Rest usando NodeJS Express y MariaDB

Descripción del proyecto

El proyecto contiene las siguientes **dependencias de producción**:

1. **express** módulo Infraestructura web rápida, minimalista y flexible para Node.js
2. **dotenv** módulo de carga Variables de entorno
3. **mariadb** módulo nativo de javascript para acceder a bases de datos MariaDB.
4. **cors** módulo de middleware de express para configurar Cross-Origin Resource Sharing en una aplicación express.

También las siguientes **dependencias de desarrollo**:

1. **nodemon** Es una herramienta que ayuda a desarrollar aplicaciones basadas en node.js al reiniciar automáticamente la aplicación de Node.js cuando se detectan cambios en los archivos de código javascript en el directorio del proyecto.
2. **jest** Es un marco de pruebas automatizadas javascript.
3. **supertest** Es un marco de pruebas para solicitudes http hacia una aplicación express.

Instale las siguientes herramientas

Para que funcione esta aplicación web necesitan tener instalado las siguientes Herramientas.

1. **NodeJS** se puede descargar desde <https://nodejs.org/es/download/>
2. **MariaDB** se puede descargar desde <https://mariadb.org/download/>
3. Una herramienta para trabajar con la base de datos recomendamos **DBeaver Community** se puede descargar desde <https://dbeaver.io/>
4. Recomendamos que tengan instalado **git** (se puede descargar desde <https://git-scm.com/downloads>) en su computadora para enviar el proyecto.
5. **Visual Studio Code** se puede descargar desde <https://code.visualstudio.com/>

Instale las siguientes extensiones en Visual Studio Code

1. **npm** comandos de npm.
2. **npm intellisense** ayuda con el autocompletado de funciones, métodos, etc. de módulos javascript instalados en el proyecto
3. **Prettier - Code formatter** ayuda a indentar el código
4. **Bracket Pair Colorizer** ayuda en la visualización de bloques de código marcando llaves, corchetes y parentesis en distintos colores.
5. **import cost** nos permite ver cuanto espacio ocupa la importación de un módulo en nuestros archivos
6. **vscode-icons** juegos de iconos para los archivos del proyecto.
7. **path intellisense** ayuda con el autocompletado de nombres de archivos y directorios del proyecto
8. **REST Client** nos permite interactuar con una api REST

Restaurar la base de datos de ejemplo

Una vez instalado el motor de base de datos **MariaDB** (es muy importante anotar la contraseña del usuario **root** de la base de datos) y la herramienta de administración de bases de datos **DBeaver Community**.

1. Crear una conexión a la base de datos usando el usuario y contraseña del servidor de base de datos.
2. Crear una nueva base de datos **Recordar el nombre de la base de datos**
3. Restaurar la base de datos desde el archivo **respaldo-employees-mariadb.sql** en la base de datos creada anteriormente

Instalar los módulos de los que depende la aplicación

Con una terminal ubicada en el directorio del proyecto ejecutar el siguiente comando

```
npm install
```

Configurar la aplicación

1. Copiar el archivo **.env-ejemplo**.
2. Renombrar la copia del archivo a **.env**.
3. Especificar los valores adecuados a las variables según su entorno local.

Ejecutar la aplicación

Ejecutar el servidor en modo producción

Con una terminal ubicada en el directorio del proyecto ejecutar el siguiente comando

```
npm start
```

Ejecutar el servidor en modo desarrollo, recargando el servidor cada vez que se guarda un cambio en algún archivo

Con una terminal ubicada en el directorio del proyecto ejecutar el siguiente comando

```
npm run dev
```

Ejecutar las pruebas automatizadas por única vez

Con una terminal ubicada en el directorio del proyecto ejecutar el siguiente comando

```
npm test
```

Ejecutar las pruebas automatizadas, recargando cada vez que se guarda un cambio en algún archivo

Con una terminal ubicada en el directorio del proyecto ejecutar el siguiente comando

```
npm run test:watch
```