



Escuela de Ingeniería en Computación
Carrera de Ingeniería en Computación
Proyecto de Ingeniería de Software

Manual Técnico: Clínica ELS

Autores

Keylor Velásquez Arias
Randy Fonseca Rivera
Deyner Navarro Badilla
Hector Guillen Ramirez

Sede Universitaria de Cartago
Costa Rica
Mayo, 2024

Tabla de contenidos

1. Objetivos	4
1.1 Objetivo General	4
1.2 Objetivos Específicos	4
2. Introducción	4
3. Requerimientos Técnicos	4
3.1 Requerimientos Mínimos de Hardware	4
Servidor Web/API	4
Servidor de Base de Datos	4
Servidor de Aplicación	5
3.2 Requerimientos Mínimos de Software	5
Servidor Web/API	5
Servidor de Aplicación	5
Servidor de Base de Datos	5
4. Herramientas Utilizadas para el Desarrollo	6
5. Instalación local	7
5.1 Instalar Backend	7
5.2 Instalar Frontend	9
7. Diseño de la Arquitectura Física	10

Tabla de Figuras

Figura 1: Ejecución comando NPM START.	9
Figura 2: Ejecución comando npm run dev	10

1. Objetivos

1.1 Objetivo General

Brindar la información necesaria para poder realizar la instalación y configuración del sistema web.

1.2 Objetivos Específicos

1. Proporcionar una visión general de la estructura del sistema web, incluyendo sus componentes principales y su interrelación.
2. Ofrecer instrucciones paso a paso para la instalación del sistema en diversos entornos.
3. Detallar los requisitos de las especificaciones de hardware y software necesarios para la instalación de la aplicación.
4. Describir las herramientas utilizadas para el desarrollo y diseño de la aplicación.

2. Introducción

Este manual técnico ha sido creado para proporcionar una guía detallada sobre la implementación, configuración y uso del sistema web de la Clínica ELS. Está dirigido a desarrolladores, administradores de sistemas, encargados de la gestión y operación del sistema. La página web de la Clínica ELS es una solución web diseñada para gestionar dicho negocio, tanto en su parte de tratamientos, como la academia que ofrecen. Su arquitectura modular y escalable permite una integración eficiente con otros sistemas.

3. Requerimientos Técnicos

3.1 Requerimientos Mínimos de Hardware

Servidor Web/API

- Procesador: CPU de 2 núcleos a 2.4 GHz
- Memoria RAM: 8 GB
- Almacenamiento: 100 GB
- Conectividad: 1 Gbps

Servidor de Base de Datos

- Procesador: CPU de 4 núcleos a 2.4 GHz
- Memoria RAM :16 GB
- Almacenamiento: 200 GB de espacio libre en disco
- Conectividad: 1 Gbps

Servidor de Aplicación

- Procesador: CPU de 4 núcleos a 2.4 GHz
- Memoria RAM: 8 GB
- Almacenamiento: 100 GB de espacio libre en disco
- Conectividad: 1 Gbps

3.2 Requerimientos Mínimos de Software

Servidor Web/API

- Sistema Operativo:
 - Windows Server 2016 o superior / Windows 10 o superior / Linux
- Servidor Web:
 - Node.js 14.x o superior
 - Express.js.
- Privilegios de Administrador: Si

Servidor de Aplicación

- Sistema Operativo:
 - Windows Server 2016 o superior / Windows 10 o superior / Linux
- Servidor de Aplicaciones:
 - Node.js 14.x o superior
- Frameworks/Dependencias:
 - React 17.x o superior
- Gestor de Paquetes:
 - npm 6.x o superior
- Privilegios de Administrador: Si

Servidor de Base de Datos

- Sistema Operativo:
 - Windows Server 2016 o superior / Windows 10 o superior / Linux
- Motor de Base de Datos:
 - MySQL 5.7 o superior
- Privilegios de Administrador: Si

4. Herramientas Utilizadas para el Desarrollo

Node.js

Este es un entorno de ejecución muy conocido para JavaScript que permite ejecutar código del lado del servidor. Con este se implementó todo el backend del sistema, manejando solicitudes HTTP/HTTPS y desarrollando la lógica del servidor y APIs RESTful para conexión con la base de datos.

React con Vite

React es una biblioteca de JavaScript para construir interfaces de usuario, puede utilizar Vite que es una herramienta de desarrollo rápida y ligera que optimiza el proceso de construcción de aplicaciones React. Aquí se desarrolló toda la interfaz de usuario del sistema, creando componentes reutilizables y mantenibles y con Vite mejorando la velocidad de carga.

MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto. Es conocido por su rendimiento y facilidad de uso. Aquí se crearon los esquemas, tablas y relaciones entre datos. Además, de la ejecución de consultas SQL para la manipulación y recuperación de datos.

Visual Studio Code

Este es un editor de código fuente de Microsoft. Es gratuito, de código abierto y compatible con muchos lenguajes de programación y herramientas. Fué el principal editor utilizado para el desarrollo del código fuente del sistema web.

GitHub

GitHub es una plataforma de alojamiento de código que utiliza Git para el control de versiones. Ofrece herramientas para la colaboración, y gestión de proyectos. Se creó un repositorio centralizado del código fuente para darle seguimiento de cambios en el código y trabajar de manera colaborativa.

Postman

Esta es una herramienta de colaboración para el desarrollo de APIs. Permite realizar pruebas de APIs, documentación y monitoreo de las mismas. Ayudó en la depuración de las APIs desarrolladas en Node.js y documentación de endpoints y sus funcionalidades.

Vercel

Vercel es una plataforma de despliegue y alojamiento para aplicaciones web. Ofrece integración continua y despliegue continuo (CI/CD) y se especializa en proyectos de frontend. Se utilizó para el despliegue y alojamiento de la aplicación frontend desarrollada con React y el backend desarrollado con Express.js

5. Instalación local

5.1 Instalar Backend

Como requisito para instalar el API es necesario NodeJS y MySQL, si ya posee este entorno de tiempo de ejecución puede saltarse la siguiente sección y pasar a la sección de la instalación del proyecto.

5.1.1 Instalar MySQL

1. Instalar el gestor de base de datos, siguiendo el siguiente comando:

```
sudo apt install mysql-server
```
2. Clonar el repositorio de la base de datos para ejecutar los scripts necesarios, con el siguiente comando en una ruta elegida:

```
git clone https://github.com/keylorvela/curso_proyeto/Scripts - DB
```
3. Configurar autenticación y contraseña de MySQL, para esto debe ejecutar:

```
sudo mysql_secure_installation
```
4. Seguidamente se debe asignar una contraseña al user MySQL, con el siguiente comando, siendo la palabra dentro de las comillas la contraseña a definir:

```
sudo mysql  
ALTER USER 'root'@'localhost' IDENTIFIED WITH  
caching_sha2_password BY 'password';
```
5. Para crear la base de datos en la cual se definen las funciones y tablas necesarias para el sistema, se utilizan los siguientes:

```
mysql -u tu_usuario -p dbClinicaELS<  
/ruta/de/paso2/Creacion Tablas - Proyecto ELS.sql
```
6. Se utiliza la ruta y nombre del script sql para ejecutarlo, la ruta debe ser la misma en la cual se clonó el repositorio de la base de datos.

5.1.2 Instalar NodeJS

1. Por defecto en algunos sistemas operativos NodeJS viene instalado, para verificar si ya está instalado deberá de ejecutar los siguientes comandos:

```
nodejs -v
```

```
nodejs --version
```

```
node -v
```

```
node --version
```

```
sudo apt install nodejs
```

2. Ya sea si se tiene NodeJS instalado de fábrica o si lo instaló con el paso anterior es necesario instalar de igual manera el gestor de paquetes con el siguiente comando:

```
sudo apt install  
npm
```

3. Ya que el proyecto es compatible con diferentes versiones de NodeJS siempre es recomendado usar la última versión estable por lo que se puede instalar con los siguientes:
4. `sudo npm install -g`
5. En caso de querer verificar que posee la última versión de NodeJS puede ejecutar los comandos del paso 1 y comparar con la versión que se muestra en la página oficial: <https://nodejs.org/en/download/>

5.1.3 Instalar API

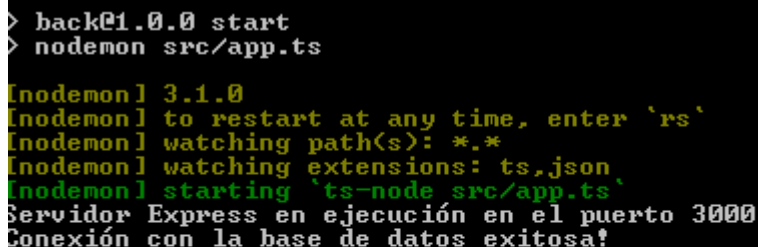
1. Es necesario descargar el repositorio del proyecto con el comando:

```
git clone  
https://github.com/keylorvela/curso_proyeto/back
```

2. Ingrese a la carpeta donde se descargó el API, el nombre de la carpeta es "back". Una vez dentro de la carpeta deberá instalar los paquetes necesarios para el funcionamiento del API, ejecutar el comando:

```
npm install
```

3. Para verificar la instalación puede ejecutar el proyecto con el comando:
`npm start`



```
> back@1.0.0 start  
> nodemon src/app.ts  
  
[nodemon] 3.1.0  
[nodemon] to restart at any time, enter `rs`  
[nodemon] watching path(s): *.*  
[nodemon] watching extensions: ts,json  
[nodemon] starting `ts-node src/app.ts`  
Servidor Express en ejecución en el puerto 3000  
Conexión con la base de datos exitosa!
```

Figura 1: Ejecución comando NPM START.

5.2 Instalar Frontend

Para instalar el repositorio que contiene las interfaces se debe:

1. Es necesario descargar el repositorio del proyecto con los comandos:

```
git clone
https://github.com/keylorvela/curso_proyeto/frontend_admin
git clone
https://github.com/keylorvela/curso_proyeto/frontend_clinica
```

2. Ingresa a las carpetas (desde consola) donde se descargaron ambas la interfaces, los nombres de las carpetas "frontend_admin" y "frontend_clinica".

3. Una vez dentro de la carpeta deberá ejecutar el comando:

```
npm install
```

El mismo descargará todas las dependencias que posee el proyecto.

4. Finalmente, para verificar que la instalación ha funcionado, puede ejecutar el comando:

```
npm run dev
```

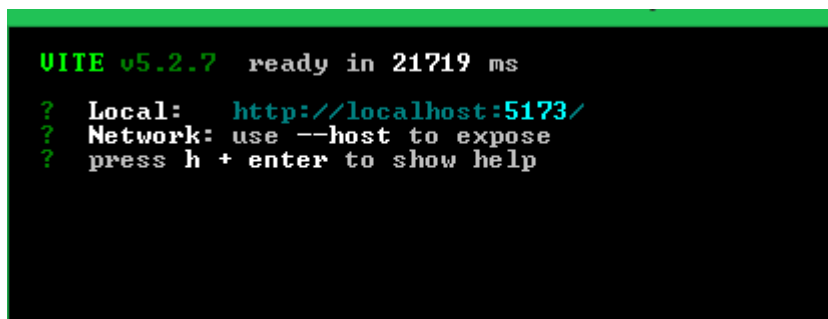


Figura 2: Ejecución comando npm run dev

7. Diseño de la Arquitectura Física

