



ITESM Campus Santa Fe

Nombre del bloque:

TC2007B: Integración de seguridad informática en redes y sistemas de software (Gpo 402)

Nombre del entregable:

Manual de Instalación de la aplicación web

Equipo 3:

Aquiba Yudah Benarroch Bittán, A01783710

Diego Cordova Rodríguez, A01781166

Jin Sik Yoon, A01026630

Julio César Rodríguez Figueroa, A01029680

Profesores:

Ahmed Imad Hammoodi AL-Jarah

Carlos Enrique Vega Álvarez

Edith Carolina Arias Serna

Jorge Rodríguez Ruiz

Lizbeth Peralta Malváez

Fecha de entrega:

23 de octubre de 2025

Índice

1.- Objetivo.....	3
2.- Descripción general del sistema.....	3
3.- Estructura del proyecto.....	4
4.- Requerimientos del sistema.....	5
4.1.- Hardware.....	5
4.2.- Software.....	5
5.- Instalación del entorno.....	6
Paso 1. Instalar dependencias base del proyecto.....	6
NPM y Node.....	6
Git.....	6
MongoDB.....	6
Paso 2. Ingresar al servidor/carpeta local donde se quiera ejecutar el proyecto.....	7
Paso 3. Clonar repositorio del proyecto en una carpeta.....	8
Paso 4. Instalar dependencias de la aplicación.....	8
Frontend.....	8
Backend.....	9
Paso 5. Configurar variables de entorno.....	10
Frontend.....	10
Backend.....	10
6.- Ejecución del sistema.....	11
Paso 1. Ejecutar servidor (backend).....	11
Paso 2. Ejecutar cliente (frontend).....	11
Paso 3. Acceder al sistema.....	12
7.- Despliegue del sistema.....	14
7.1.- Despliegue local (computadora).....	14
7.2.- Despliegue en servidor local (red local).....	15
7.3.- Despliegue en servidor en la nube (internet).....	16
8.- Recomendaciones finales.....	18
9.- Referencias y documentación.....	19

1.- Objetivo

Este manual fue realizado con el fin de guiar a futuros programadores o administradores en el proceso de instalación, ejecución y despliegue del sistema “Sistema de Registro de Emergencias” (SRE). Este sistema fue desarrollado por estudiantes de la carrera de Ingeniería en Tecnologías Computacionales (ITC) del Tecnológico de Monterrey, Campus Santa Fe; en el periodo comprendido por los meses agosto-octubre del año 2025.

Este documento detalla los pasos necesarios para instalar dependencias del proyecto, configurar el entorno de desarrollo, ejecutar el servidor (backend), interfaz principal (frontend) y desplegar la aplicación en un entorno local o de producción.

2.- Descripción general del sistema

El sistema consiste en una aplicación web desarrollada a través del framework React-Admin, mismo que permite una rápida y fácil conexión con distintos gestores de bases de datos.

El sistema desarrollado permite realizar las siguientes acciones por rol:

- **Administrador**
 - Visualizar/Crear/Actualizar/Eliminar reportes o notas
 - Visualizar estadísticas médicas/urbanas
 - Visualizar/Crear/Actualizar/Eliminar usuarios
 - Editar folios de reportes médicos/urbanos
- **Paramédicos:**
 - Registrar reportes/notas médicas
 - Ver reportes/notas médicas creadas por sí mismo
 - Editar folios de reportes médicos
- **Operadores urbanos**
 - Registrar reportes/notas urbanas
 - Ver reportes/notas urbanas creadas por sí mismo
 - Editar folios de reportes urbanos
- **Jefes de turno**

- Registrar reportes/notas urbanas de su turno
- Ver reportes/notas urbanas de su turno
- Editar folios de reportes médicos/urbanos

3.- Estructura del proyecto

El proyecto contiene la siguiente estructura:

- Frontend (cliente): Desarrollado mediante el framework React-Admin. Es el encargado de proveer la interfaz gráfica para todos los usuarios en un navegador web (Ej: Chrome, Firefox, Opera, Safari, etc.).
- Backend (servidor): Utiliza [Node.js](#) y Express para gestionar la base de datos, proveer autenticación de usuarios y crear/consultar reportes o notas.
- Base de datos: Se utiliza MongoDB como Sistema Gestor de Base de datos (SGBD) no relacional, proporcionando mayor flexibilidad al momento de agregar/eliminar/modificar datos.

El repositorio de GitHub del proyecto se encuentra organizado en dos carpetas principales: frontend y backend.

Carpeta	Función
Backend	Contiene la conexión con la base de datos, así como las diferentes direcciones para la comunicación con esta (endpoints).
Frontend	Contiene todos los archivos del cliente de la aplicación, desde la interfaz gráfica, listas de datos, formularios de creación/edición, filtros, entre otros.

4.- Requerimientos del sistema

A continuación se detallan los requerimientos del sistema, comprendiendo tanto la parte física donde se ejecutará (hardware), hasta el código y dependencias (software).

4.1.- Hardware

Tabla 1. Requerimientos de Hardware del sistema

Memoria RAM	Mínimo 4GB
Conexión a internet	Obligatoria

4.2.- Software

Tabla 2. Requerimientos de software del sistema

Componente	Versión recomendada
Sistema Operativo	Windows 10/11, macOS, iOS, Android
Node.js	^24.10.0
npm	^11.6.0
MongoDB	^6.20.0
Express	^5.1.0
argon2	^0.44.0
Navegador Web	Google Chrome, Firefox, Safari

Nota: Las versiones recomendadas son las utilizadas para el desarrollo del proyecto, mismas con las que fue creado y probado el sistema.

Para encontrar más detalles y dependencias, revisar el archivo “package.json” dentro de las carpetas “frontend” y “backend” del proyecto.

5.- Instalación del entorno

Paso 1. Instalar dependencias base del proyecto

NPM y Node

- NPM (Node Package Manager): Es un gestor de paquetes que permite instalar las dependencias del proyecto.
- Node: Es un entorno de ejecución de JavaScript de código abierto y multiplataforma, sirve para ejecutar código de JavaScript (Microsoft, 2025).

Sin estos elementos, no se podrá ejecutar el código de la aplicación ni instalar sus dependencias. Para acceder a la documentación completa de instalación, visitar la página de instalación de [Node.js](https://nodejs.org/es/download): <https://nodejs.org/es/download>

Git

Git es un sistema de control de versiones que permite crear un clon local de un proyecto con un control de versiones completo. Estos repositorios locales permiten trabajar sin conexión o de forma remota con facilidad (Microsoft, 2023).

Para acceder a la documentación completa de instalación, visitar la página de instalación de Git: <https://git-scm.com/downloads>

MongoDB

MongoDB es un sistema de gestión de bases de datos no relacionales (SGBD) que utiliza documentos flexibles en lugar de tablas y filas para procesar y almacenar diversas formas de datos (IBM, S.F.).

Para acceder a la documentación completa de instalación, visitar la página de instalación de MongoDB:

<https://www.mongodb.com/docs/manual/installation/#std-label-tutorial-installation>

Una vez instalado, se deberá crear una nueva base de datos:

```
None
use sre // Crea la base de datos del proyecto
```

Después, se deberán crear todas las colecciones del proyecto

```
Shell
db.createCollection("usuarios");
db.createCollection("reportes_medicos");
db.createCollection("notas_medicas");
db.createCollection("logs");
```

Para insertar datos en una colección, se puede utilizar el siguiente comando de ejemplo:

```
Shell
// Insertar un dato
db.tu_coleccion.insertOne({id:1,usuario:"usuario1"}

// Insertar varios datos
db.tu_coleccion.insertMany([ {id:1,usuario:"usuario1"}
{id:2,usuario:"usuario2"} ]);
```

Nota: Se recomienda utilizar este comando si se quieren insertar datos antes de ingresar al sistema, pues dentro de este ya se puede ingresar datos.

Aquí el link con los comandos para introducir el Dummy data propuesto por nosotros:

https://docs.google.com/document/d/1hbvkYQzD8-U_j9170UFTa19GUYDUJpr5oWETn-m2IMo/edit?usp=sharing

Paso 2. Ingresar al servidor/carpeta local donde se quiera ejecutar el proyecto

```
Shell
cd tu_carpeta // Ingresa a la carpeta deseada
```

Paso 3. Clonar repositorio del proyecto en una carpeta

Una vez dentro de la carpeta en donde se quiera instalar el proyecto, se deberá clonar el repositorio de GitHub con todos los archivos necesarios.

```
Shell
// Dentro de la carpeta donde se desee instalar el proyecto
git clone git@github.com:diegocrdz/TC2007B_Equipo3.git
```

Este comando instalará todos los archivos necesarios para ejecutar el proyecto dentro de la carpeta deseada.

Paso 4. Instalar dependencias de la aplicación

Una vez instalados todos los archivos del proyecto, se deberá ingresar a la carpeta de frontend (cliente) y backend (servidor) e instalar las dependencias de cada uno. Estas dependencias se tratan de librerías necesarias para que el código sea ejecutado sin problemas.

Frontend

```
Shell
cd TC2007B_Equipo3 // Entrar a la carpeta del repositorio
cd frontend // Entrar a la carpeta del cliente (frontend)
npm install // Instalar dependencias
```

Se deberá obtener una salida en la terminal similar a la siguiente:


```

PS C:\Users\dcord\Documents\repos\TCB2007B_Equipo3\backend> npm install package.json
npm warn deprecated package.json@2.0.1: Use pkg.json instead.

added 186 packages, and audited 187 packages in 5s

21 packages are looking for funding
  run `npm fund` for details

10 vulnerabilities (1 low, 2 moderate, 2 high, 5 critical)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
❖ PS C:\Users\dcord\Documents\repos\TCB2007B_Equipo3\backend>

```

Backend

Shell

```

cd TC2007B_Equipo3 // Entrar a la carpeta del repositorio
cd backend // Entrar a la carpeta del servidor (backend)
npm install // Instalar dependencias

```

Se deberá obtener una salida en la terminal similar a la siguiente:

```

PS C:\Users\dcord\Documents\repos\TCB2007B_Equipo3\frontend> npm install package.json
npm warn deprecated package.json@2.0.1: Use pkg.json instead.

added 73 packages, changed 68 packages, and audited 553 packages in 17s

150 packages are looking for funding
  run `npm fund` for details

11 vulnerabilities (2 low, 2 moderate, 2 high, 5 critical)

To address all issues, run:
  npm audit fix

Run `npm audit` for details.
❖ PS C:\Users\dcord\Documents\repos\TCB2007B_Equipo3\frontend>

```

Paso 5. Configurar variables de entorno

En este paso se crearán los archivos .env para “frontend” y “backend”. El propósito de estos archivos es contar con variables que puedan ser accedidas desde el cliente y servidor sin exponerlas directamente en el código de la aplicación. Sin estos archivos, no se podrá tener comunicación entre el servidor, cliente y base de datos.

Estos archivos no se encuentran en el repositorio del proyecto para proteger la integridad de la aplicación, llave secreta para tokens y la dirección de la base de datos.

Frontend

Dentro de la carpeta /frontend, crear un archivo llamado “.env” con la siguiente información:

```
None
VITE_BACKEND=https://127.0.0.1:3000
```

Esto permitirá que el cliente (frontend) se comuniquen con el backend a través de la dirección 127.0.0.1 (localhost), corriendo en el puerto 3000.

Backend

Dentro de la carpeta /backend, crear un archivo llamado “.env” con la siguiente información:

```
None
DB=mongodb://127.0.0.1:27017/sre
JWTKEY=secretKey
```

Esto permitirá que el servidor (backend) sea capaz de comunicarse con la base de datos, en la dirección 127.0.0.1 (localhost), a través del puerto 27017. Nótese que la dirección de la base de datos termina en “sre”. En este caso, este se trata del nombre de la base de datos en MongoDB, correspondiente a las siglas del Sistema de Registro de Emergencias (SRE).

JWTKEY se trata de una clave utilizada para crear un token para los usuarios al momento de ingresar al sistema. Este viaja entre cliente y servidor para autenticar/autorizar peticiones de los usuarios sin guardar sus datos en el servidor. En caso de querer cambiar la clave secreta, modificar el valor en el archivo .env creado, de la forma JWTKEY=tu_nueva_clave.

6.- Ejecución del sistema

Paso 1. Ejecutar servidor (backend)

Dentro de la carpeta /backend, ejecutar el siguiente comando en la terminal:

```
Shell
node index.js // Inicia el servidor
```

Deberá aparecer el siguiente resultado en la terminal:



```
❖ diego.crdz@MacBook-Air-de-DiegoC backend % node index.js
HTTPS Server running on port 3000
conectado a la base de datos
█
```

Paso 2. Ejecutar cliente (frontend)

Dentro de la carpeta /frontend, ejecutar el siguiente comando:

```
Shell
npm run dev // Inicia la aplicación en modo desarrollador
```

Deberá aparecer el siguiente resultado en la terminal:

```
diego.crdz@MacBook-Air-de-DiegoC frontend % npm run dev
VITE v6.3.5 ready in 170 ms

→ Local:   https://localhost:5173/
→ Network: https://10.49.92.236:5173/
→ press h + enter to show help
```

Paso 3. Acceder al sistema

Ingresar en un navegador web a la página de internet <https://127.0.0.1:5173/>. Esta página deberá redireccionar a la página de inicio de sesión del sistema:



Nota: Si se presenta un error en el que no se permita realizar el inicio de sesión a pesar de usar credenciales de inicio de sesión correctas, se recomienda utilizar el navegador web Firefox. Este error ocurre debido a que la página https ha sido autofirmada y no fue validada por una autoridad certificada.

A continuación se presenta una lista de usuarios de prueba para ingresar al sistema:

Usuario	Contraseña
admin	123
paramedico1	123
paramedico2	123
paramedico3	123
urbano1	123
urbano2	123
urbano3	123
jefe_turno1	123
jefe_turno2	123
jefe_turno3	123

***Nota:** La contraseña dentro de la base de datos está cifrada mediante el algoritmo Argon2, por lo que cuando se consulten los usuarios dentro de la aplicación, no se podrán ver las contraseñas en texto plano, sino su huella hash.*

7.- Despliegue del sistema

Al momento de desplegar el sistema en un entorno de producción, la aplicación se debe ejecutar en un servidor siempre encendido. Se debe tener un dominio HTTPS y base de datos en MongoDB. A continuación se presentan las opciones recomendadas para realizar el despliegue del sistema.

7.1.- Despliegue local (computadora)

Esta opción consiste en realizar la instalación y ejecución del sistema en una sola computadora.

¿Cuándo conviene?

Esta opción es conveniente cuando se quieren realizar pruebas internas o solo una persona quiere acceder al sistema para comprobar su funcionamiento.

¿Qué se necesita?

Una computadora que cumpla con todos los requisitos de instalación y ejecución del sistema. Revisar secciones 5 y 6 del documento para obtener más detalles de la instalación y ejecución de la aplicación.

¿Cómo se hace?

1. Encender la computadora
2. Instalar el sistema ([5.- Instalación del entorno](#))
3. Ejecutar el sistema ([6.- Ejecución del sistema](#))
4. Ingresar al sistema a través de un navegador de internet, en la dirección <https://127.0.0.1:5172>
5. Iniciar sesión con una de las credenciales válidas al sistema.

7.2.- Despliegue en servidor local (red local)

Esta opción consiste en instalar y ejecutar el sistema en una computadora de la red de la alcaldía local.

¿Cuándo conviene?

Esta opción es conveniente cuando solo se quiere ejecutar la aplicación dentro de la misma oficina, donde todos los dispositivos comparten la misma red de internet.

¿Qué se necesita?

Una computadora que cumpla con todos los requisitos de instalación y ejecución del sistema. Revisar secciones [5.- Instalación del entorno](#) y [6.- Ejecución del sistema](#) para obtener más detalles de la instalación y ejecución de la aplicación.

¿Cómo se hace?

1. Encender la computadora
2. Instalar el sistema ([5.- Instalación del entorno](#))
 - a. Configurar una dirección de la red local de la oficina en las variables de entorno del sistema ([Paso 5. Configurar variables de entorno](#)).
3. Ejecutar el sistema ([6.- Ejecución del sistema](#))
4. Ingresar al sistema a través de un navegador de internet, en la dirección configurada anteriormente en las variables de entorno.
5. Iniciar sesión con una de las credenciales válidas al sistema.

7.3.- Despliegue en servidor en la nube (internet)

Debido a las necesidades del proyecto, esta es la opción recomendada para que cualquier dispositivo con acceso a internet sea capaz de ingresar el sistema; desde administradores hasta operadores en cualquier punto del país, con cualquier dispositivo con acceso a un navegador de internet y acceso a una red.

¿Cuándo conviene?

Esta opción es conveniente cuando se quiere tener acceso al sistema desde cualquier lugar o dispositivo, las 24 horas del día.

¿Qué se necesita?

Una cuenta registrada en un servicio de proveedor de nube. Ej: AWS (Amazon Web Services), Google Cloud, entre otros. De igual manera, se necesita una base de datos en línea. Ej: MongoDB Atlas.

¿Cómo se hace?

1. Crear el servidor en la nube y subir el sistema
2. Configurar la base de datos en la nube
3. Asignar un nombre al sitio donde se accede para ingresar al sistema (Ej: sre.alcaldiacuajimalpa.gob.mx).
4. Configurar el servicio HTTPS para que el sitio muestre el mensaje de “Conexión segura”.

Para configurarlo, ingresar a la carpeta “/frontend” de la aplicación e ingresar los siguientes comandos:

```
Shell
// Ingresar a la carpeta backend
cd backend

// Generar llave RSA de 4096 bytes (Llave privada)
openssl genrsa -out backend.key 4096
```



```
// Crear un certificado HTTPS para verificar la página
openssl req -x509 -new -key backend.key -out backend.crt -days 365
```

Nota: Si ya existen los archivos de llave y certificado, eliminarlos para volverlos a generar.

Se deberán utilizar los mismos comandos para generar la llave y certificado dentro de la carpeta “/backend”:

```
Shell
// Ingresar a la carpeta frontend
cd frontend

// Generar llave RSA de 4096 bytes (Llave privada)
openssl genrsa -out backend.key 4096

// Crear un certificado HTTPS para verificar la página
openssl req -x509 -new -key backend.key -out backend.crt -days 365
```

Nota: Si ya existen los archivos de llave y certificado, eliminarlos para volverlos a generar.

Para cada uno de estos comandos, se requerirá llenar la información pertinente solicitada, como el nombre del dominio, propietario, entre otros.

8.- Recomendaciones finales

Considerar las siguientes recomendaciones finales para la instalación, ejecución y despliegue del sistema:

- Navegadores recomendados: Google Chrome, Firefox
- Usuarios y contraseñas: Crear nuevas cuentas de usuarios con contraseñas con las siguientes características:
 - Al menos 9 caracteres de longitud
 - Al menos una letra mayúscula
 - Al menos un carácter especial
 - Al menos un número
- Leer el documento de manual de usuario
- Mantenimiento de usuarios: Si un empleado deja de trabajar en la alcaldía, eliminar su cuenta de usuario; de otra forma, seguirá teniendo acceso al sistema con su usuario y contraseña.

9.- Referencias y documentación

Git. (2025). *Documentación - Git*. <https://git-scm.com/docs>

IBM. (s.f.). *¿Qué es MongoDB?* <https://www.ibm.com/es-es/think/topics/mongodb>

Marmelab. (2025). *Documentación - React-Admin*.
<https://marmelab.com/react-admin/documentation.html>

Microsoft. (4 de octubre, 2023). *¿Qué es Git?*
<https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>

Microsoft. (3 de mayo, 2025). *¿Qué es NodeJS?*
<https://learn.microsoft.com/es-es/windows/dev-environment/javascript/nodejs-overview>

MongoDB. (2025). *Documentación - MongoDB*. <https://www.mongodb.com/docs/>

NodeJS. (2025). *Documentación - NodeJS*. <https://nodejs.org/docs/latest/api/>

NPM. (2025). *Documentación - NPM*. <https://docs.npmjs.com/>