



Fundación por México

Sistema de registro y gestión de tickets

Sistema implementado con creación y actualización de tickets dentro de un entorno visual para su posterior despliegue en base de datos.

Integrado con medidas de seguridad y registro de usuarios con permisos únicos en el sistema.

Manual de instalación del sistema

2023

Manual de instalación del sistema

MANUAL DE INSTALACIÓN DEL SISTEMA: Este Manual está dirigido al personal técnico responsable de instalar y configurar inicialmente el Sistema de registro y gestión de tickets, por lo tanto se asume que el lector está familiarizado con conceptos básicos de administración bajo Windows y/o MacOS, para la puesta en práctica de los conocimientos explicados aquí. Igualmente está familiarizado con uso básico de la terminal o uso básico del sistema operativo, puesto que aquí se explicará paso a paso las acciones a seguir para instalar el programa de manera rápida y sencilla.

Características Generales Sistemas Operativos : Windows 10, Windows 11, MacOS, Linux (Ubuntu)

Lenguaje de Desarrollo : JavaScript. TypeScript,

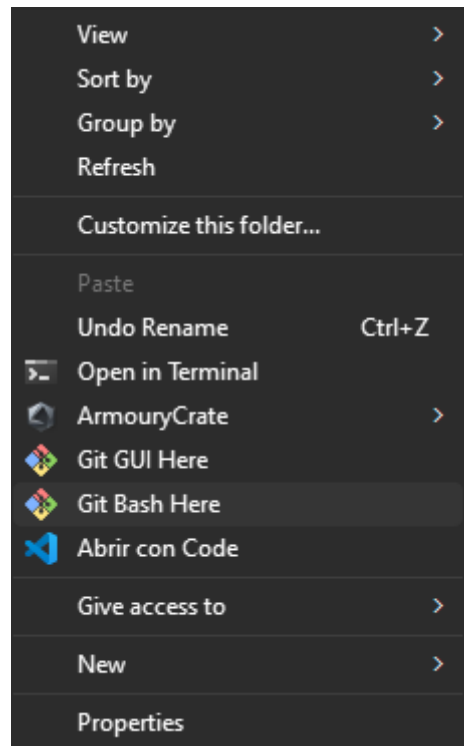
Ambiente de desarrollo: React

Modelo de acceso a Bases de Datos : Cliente / Servidor de 2 capas

Base de datos: La base de datos se encuentra activa en la nube a través de la herramienta MongoDB Atlas. Dicha base de datos corre en un servidor de AWS y se encuentra disponible en todo momento. El sistema se conecta a la base de datos por medio de una API, sin embargo no es necesario instalar ninguna base de datos localmente para que el sistema funcione.

Pasos de instalación:

1. Primeramente es necesario instalar una terminal con acceso a git, idealmente git bash. Si su computadora no cuenta con una terminal con estas características favor de leer el siguiente manual de instalación: [How to install Git Bash in Windows \(educative.io\)](https://educative.io/qa/how-to-install-git-bash-in-windows). O si bien se prefiere descargar Git desde el siguiente link: [Git - Downloads \(git-scm.com\)](https://git-scm.com/downloads)
2. Una vez habiendo instalado Git Bash, dar click en el botón derecho del mouse dentro de la ruta en la que se desea instalar el proyecto y hacer click en “Git Bash Here”. Esta opción se encuentra visible para todos los sistemas operativos, menos Windows 11, donde primero se tiene que hacer click en “Show more options / Mostrar más opciones”.



3. Una vez esté abierta la terminal se deberá ejecutar el siguiente comando:

git clone https://github.com/SamCF21/TC2007B.401.git

Esto creará una copia local del contenido del repositorio de GitHub donde se encuentra el proyecto. [SamCF21/TC2007B.401 \(github.com\)](https://github.com/SamCF21/TC2007B.401)

```
juanp@DESKTOP-E3U1B1C MINGW64 ~/Desktop/TEC/Prueba/prueba 2
$ git clone https://github.com/SamCF21/TC2007B.401.git
Cloning into 'TC2007B.401'...
remote: Enumerating objects: 73973, done.
remote: Counting objects: 100% (703/703), done.
remote: Compressing objects: 100% (455/455), done.
remote: Total 73973 (delta 192), reused 662 (delta 185), pack-reused 73270
Receiving objects: 100% (73973/73973), 82.88 MiB | 10.03 MiB/s, done.
Resolving deltas: 100% (31888/31888), done.
```

Si se clonó correctamente el repositorio debería verse un bloque de líneas de la manera que se muestra en la imagen superior.

4. Una vez tengamos el repositorio clonado localmente, navegaremos a la carpeta *backend*, la cual se encuentra dentro de la carpeta *SistemaTickets*. Esto se puede hacer entrando manualmente a la carpeta y utilizando el botón “Git Bash Here” dando click derecho en algún lugar vacío de la carpeta o si no ejecutando el siguiente comando desde la misma terminal en donde se hizo el clon.

```
cd TC2007B.401/SistemaTickets/backend
```

5. Una vez nos encontramos en la carpeta *backend* del proyecto, ejecutaremos el siguiente comando:

```
npm install
```

Una vez ejecutado este comando se deberían desplegar las siguientes líneas que muestran la instalación satisfactoria de los paquetes:

```
$ npm install
npm WARN deprecated package.json@2.0.1: Use pkg.json instead.

added 229 packages, and audited 230 packages in 4s

16 packages are looking for funding
  run 'npm fund' for details

9 vulnerabilities (5 moderate, 3 high, 1 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.
```

6. Una vez se haya instalado npm tendremos que instalar otras librerías necesarias para la ejecución del backend del proyecto. Para esto correremos los siguientes comandos:

```
npm install dotenv
```

```
npm install express
```

Se deberá desplegar un mensaje similar al del punto anterior con la instalación de los paquetes.

7. Una vez teniendo las librerías instaladas será necesario probar que el código se ejecute. para esto utilizaremos el siguiente comando:

```
node index.js
```

A esto recibiremos un mensaje de error que dicta lo siguiente:

```
Secret key not found in .env file.
```

Esto ocurre debido a que la llave generada para el Json Web Token se generará directamente desde el servidor para evitar brechas de seguridad.

8. Para crear la llave secreta que se solicita en el punto anterior tendremos que ejecutar el siguiente comando:

```
node keyGenerator.js
```

A lo que obtendremos la siguiente respuesta:

```
Random secret key saved to .env
```

9. Ahora ejecutaremos de nuevo el siguiente comando:

```
node index.js
```

Pero esta vez obtendremos el siguiente mensaje:

*Servidor escuchando en puerto 1338
conectado a la base de datos*

El mensaje *conectado a la base de datos* solo será desplegado si se realizó una conexión exitosa.

10. Ahora vamos a levantar el frontend. Para esto podemos ingresar manualmente a la carpeta frontend y hacer “Git Bash Here”. Si queremos hacerlo desde la terminal utilizaremos los siguiente comandos:

```
cd ..  
cd frontend/
```

Nota importante***

Necesitaremos hacer uso de 2 terminales distintas para correr el backend y el frontend, por lo que es necesario abrir una segunda terminal dentro de dicha carpeta sin cerrar la terminal que estábamos trabajando anteriormente.

11. Una vez nos encontremos en la carpeta *frontend* ejecutaremos desde esa dirección el siguiente comando a través de la terminal.

```
npm install react-scripts --save
```

12. Ahora por último ejecutaremos la siguiente línea de código para levantar el la web app.

```
npm start
```

El código debería levantar en el navegador predeterminado la web app de manera automática, pero si no pasa esto es necesario ingresar al navegador de nuestra preferencia e ingresar a la ruta <https://localhost:3000>

Si se levantó la la App correctamente debemos obtener un mensaje similar a este:

```
Compiled successfully!  
  
You can now view frontend in the browser.  
  
Local:      https://localhost:3000  
On Your Network: https://10.31.58.147:3000  
  
Note that the development build is not optimized.  
To create a production build, use npm run build.  
  
webpack compiled successfully
```

13. Aunque ya hemos instalado el sistema dentro de nuestra computadora todavía es necesario importar los certificados a nuestro navegador. Para esto consultaremos la siguiente fuente: [Cómo importar y exportar un certificado digital en Microsoft Edge \(xataka.com\)](https://xataka.com/como-importar-y-exportar-un-certificado-digital-en-microsoft-edge/) e importaremos un total de 2 certificados a nuestro navegador. Estos se encuentran en la carpeta *backend* y *frontend* respectivamente. Dejaremos que el sistema defina por defecto el tipo de certificado que es en la pestaña *Almacén de certificados* e importaremos de manera separada los archivos *frontend.cer* y *backend.cer*

Felicidades. Has completado la instalación del sistema. Ahora tienes la aplicación web ejecutando en https dentro de tu computadora servidor.

Para consultar cómo funciona el sistema favor de consultar el manual de usuario.