



Instituto Politécnico Nacional



Unidad Profesional Interdisciplinaria en Ingeniería y Tecnologías Avanzadas

Ingeniería Web

Proyecto

Profesor:

Polanco Montelongo Francisco Antonio

Alumnos:

Anzures Toris Axel Yerain

López Cedillo Itzel

López Ventura Edgar Sajit

Grupo: 3TM2

Índice.

1. Introducción y requisitos.
2. Instalación de Componentes Esenciales para la Página Web
3. Navegación.
4. Seguridad.
5. API

1. Introducción y requisitos

Esta página web está diseñada para facilitar la compra directa de imágenes desde el sitio web de Marvel. Dentro de la plataforma, los usuarios tienen la capacidad de llevar a cabo diversas acciones, tales como modificar y recuperar contraseñas, agregar o eliminar usuarios, y, por supuesto, realizar transacciones de compra de imágenes, empleando herramientas de HBS, express, javascript y css para toda la parte del FrontEnd y nodejs y MySQL para la parte del BackEnd

Para garantizar un rendimiento óptimo y una experiencia sin problemas en la utilización de la página web, es esencial cumplir con los siguientes requisitos del sistema:

- Navegadores Compatibles: Se recomienda el uso de navegadores modernos como Google Chrome, Mozilla Firefox, Opera Safari o Microsoft Edge.
- Las últimas versiones de los navegadores son preferibles para asegurar compatibilidad total.
- JavaScript debe estar habilitado en el navegador para el correcto funcionamiento de las funcionalidades interactivas.
- Se recomienda tener habilitadas las cookies para mantener sesiones de usuario.

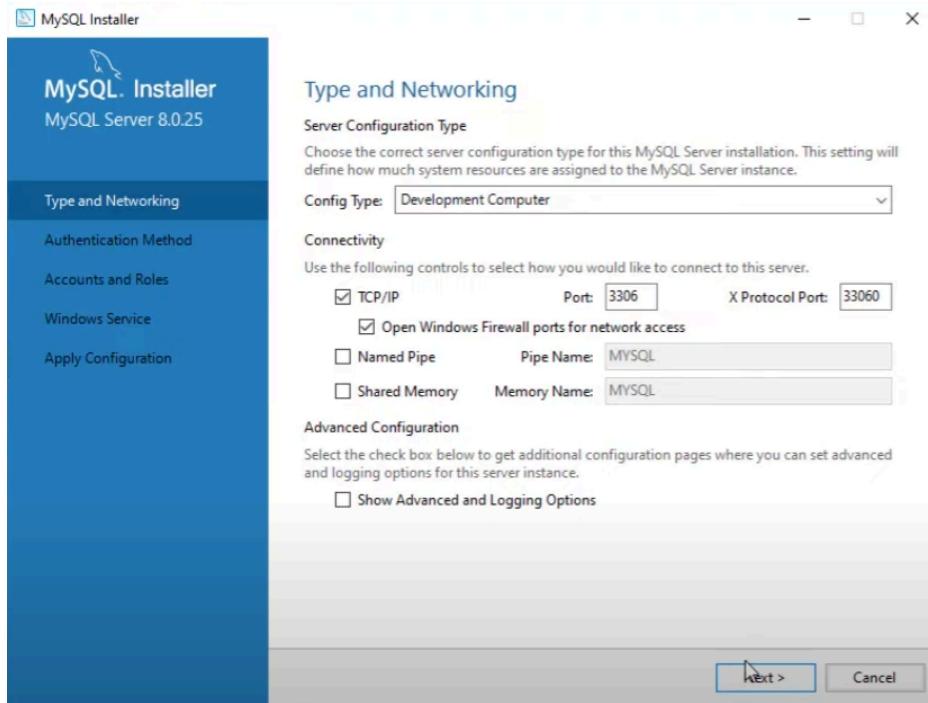
2. Instalación de Componentes Esenciales para la Página Web.

1. Node.js

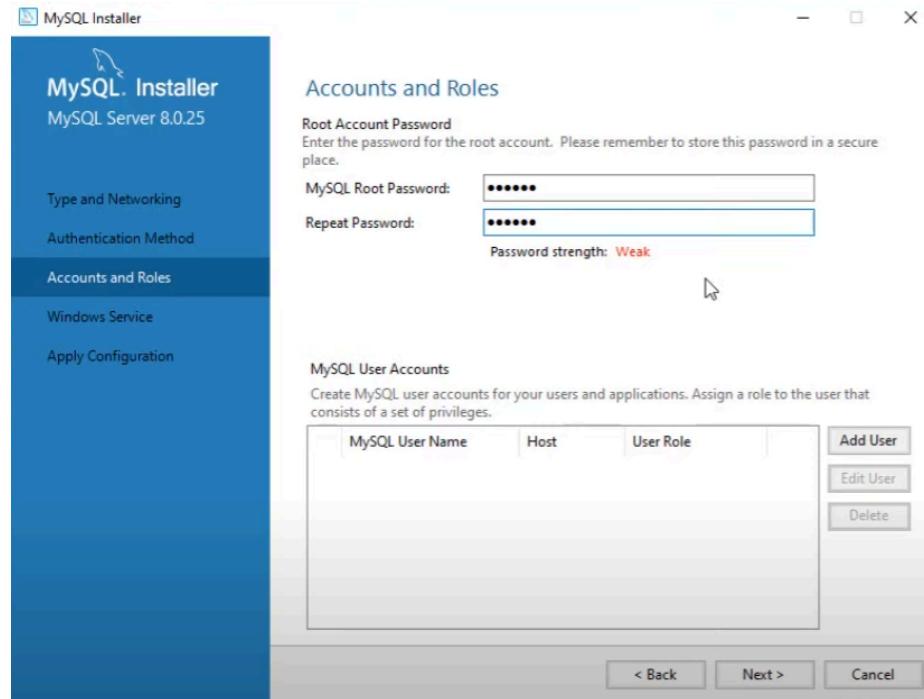
- - Descarga Node.js desde (<https://nodejs.org/en/download/current>).
- - Sigue las instrucciones de instalación para tu sistema operativo.

2. MySQL

- - Descarga MySQL desde (<https://dev.mysql.com/downloads/windows/installer/8.0.html>).
- - Inicia la instalación ejecutando el archivo descargado.
- - Selecciona la opción de instalación "FULL" para instalar todas las herramientas. Sin embargo, si solo necesitas MySQL Server y Workbench, esa selección es suficiente.
- - Continúa con la configuración de MySQL Server. Deja los datos como se muestran en la imagen proporcionada.



- Avanza hasta que llegues a la ventana para configurar la contraseña del servidor. En nuestra configuración, la contraseña se establece como "123456".



- Completa la instalación siguiendo las instrucciones, haciendo clic en "Next" en cada ventana hasta que encuentres el botón "Finish".

Una vez configurado lo anterior crearemos una base a partir del siguiente código:

```
CREATE TABLE IF NOT EXISTS Usuarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(255),
    email VARCHAR(255) UNIQUE,
    password VARCHAR(255),
    secquestion VARCHAR(255),
    secanswer VARCHAR(255),
    role VARCHAR(255)
);
CREATE TABLE Carritos (
    id INT PRIMARY KEY AUTO_INCREMENT,
    usuario_id INT,
    FOREIGN KEY (usuario_id) REFERENCES Usuarios(id)
);
CREATE TABLE PersonajesCarrito (
    id INT PRIMARY KEY AUTO_INCREMENT,
    carrito_id INT,
    nombre_personaje VARCHAR(255),
    imagen_personaje VARCHAR(255),
    FOREIGN KEY (carrito_id) REFERENCES Carritos(id)
);
```

Esta será la estructura de nuestra base en la cual se almacenará la tabla "Usuarios" la cual almacenará los siguientes datos:

- id: Un identificador único para cada usuario. Es autoincremental, lo que significa que se asignará automáticamente un valor único a medida que se agreguen nuevos registros.
- name: El nombre del usuario.
- email: La dirección de correo electrónico del usuario. Se define como única para garantizar que no haya duplicados.
- password: La contraseña del usuario. Nota importante: almacenar contraseñas en texto plano no es una práctica segura en un entorno de producción. En un escenario real, deberías utilizar técnicas de almacenamiento seguro de contraseñas, como el hash y la sal.
- secquestion: Una pregunta de seguridad asociada al usuario.
- secanswer: La respuesta a la pregunta de seguridad.
- role: El rol del usuario en el sistema. Puede ser "usuario", "admin" u otro valor según las necesidades del sistema.

En resumen, la tabla "Usuarios" almacena información esencial sobre los usuarios de tu aplicación. Cada fila representa un usuario individual, con un identificador único.

La tabla "Carrito" almacenará los siguientes datos:

- id (: Esta columna almacena un identificador único para cada carrito. La opción `AUTO_INCREMENT` significa que la base de datos generará automáticamente un valor único para esta columna al insertar un nuevo registro.
- usuario_id: Esta columna almacena el identificador de usuario asociado a cada carrito. La opción `FOREIGN KEY` establece una relación con la columna "id" de la tabla "Usuarios", lo que significa que cada valor en "usuario id" debe corresponder a un valor existente en la columna "id" de la tabla "Usuarios". Esta relación vincula cada carrito a un usuario específico.

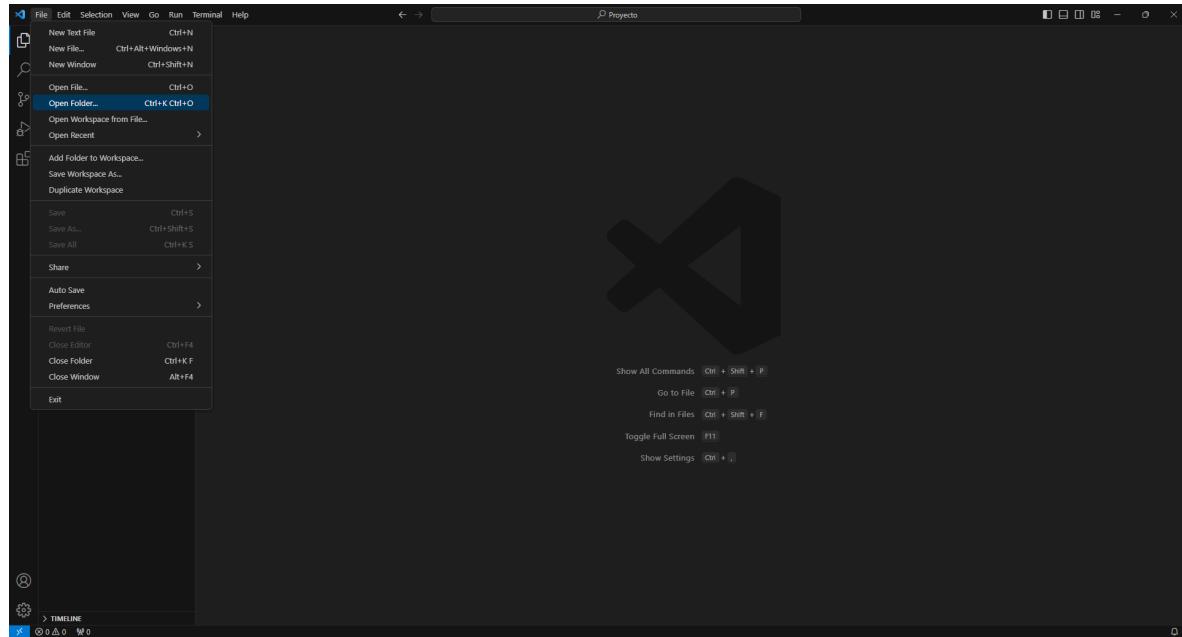
Facilita el seguimiento y la asociación de carritos de compras a usuarios individuales en una aplicación o sistema que utilice esta estructura de base de datos.

3. Visual Studio Code

- - Descarga Visual Studio Code desde (<https://code.visualstudio.com/download>).
- - Instala Visual Studio Code siguiendo las instrucciones proporcionadas.

Después de instalar y abrir Visual Studio, el siguiente paso es acceder a nuestro proyecto. Para hacerlo, ve a la esquina superior izquierda, haz clic en "File" y luego selecciona "Open Folder". Después, simplemente elige la carpeta que contiene tu proyecto y ábrela. De esta manera, tendrás tu proyecto cargado y listo

para comenzar a trabajar en Visual Studio.



Después de abrir nuestro proyecto, el siguiente paso implica trabajar en dos terminales distintas para establecer la conexión entre nuestro código y la base de datos. Aquí te detallo los pasos:

1. Terminal para la API REST:

- - Accede a la carpeta del proyecto ejecutando el comando `cd APIREST\Public`.
- - Luego, utiliza `nodemon index` para iniciar el servidor y permitir que la API REST esté en funcionamiento.

2. **Terminal para el Código de Node.js y MySQL:**

- - Cambia a la carpeta correspondiente ejecutando `cd Proyecto2\src`.
- - Finalmente, ejecuta `nodemon index` en esta terminal para poner en marcha el código que interactúa con la base de datos MySQL.

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The left sidebar displays a file tree with project files like `index.js`, `db.js`, `nodemailer.js`, `nodemon.json`, and `package-lock.json`. The main editor area shows the content of `index.js`, which contains code for setting up an Express server with static files, route configurations, and cookie parsing. The terminal at the bottom shows command-line output related to the project's configuration and execution. The status bar at the bottom right indicates the current file type as JavaScript.

```
File Edit Selection View Go Run Terminal Help
... Projeto2
EXPLORER
PROYECTO2
> APIREST
> PROYECTO2
node_modules
> src
> controllers
> models
> public
> routes
> services
> views
JS index.js
JS db.js
JS nodemailer.js
(node) nodemon.json
(node) package-lock.json
(node) package.json

index.js
...
// obtiene la ruta raíz del proyecto
const PORT = 3000;
// Instancia para el servidor HTTP de la aplicación
const app = express();
app.use(cookieParser());
const __dirname = dirname(fileURLToPath(import.meta.url));
console.log(__dirname);
// Se habilita el uso de routes users.
app.use(express.urlencoded({extended:true}));
app.use(express.json());
// se establece la ruta para los recursos estáticos
app.use(express.static(__dirname + "/public"));
// configuración de hbs como motor de plantillas
app.set("view engine", "hbs");
app.set("view", __dirname + "/views");
hbs.registerPartials(__dirname + '/views/partials', function (err) {});
/*
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
+ ... x
# E:\UPTITA\Quinto Semestre\Inge Web\Proyecto2> cd APIREST\Public
# E:\UPTITA\Quinto Semestre\Inge Web\Proyecto2\APIREST\Public> nodeon index
[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] starting `node index.js`
[nodemon] starting `node index.js`
Conexión exitosa a la base de datos
Servidor en ejecución en el puerto 2000
E:\UPTITA\Quinto Semestre\Inge web\Proyecto2\Proyecto2>src
Servidor ejecutándose en el puerto 3000
|
```

Al realizar estos pasos en ambas terminales, se establecerá la conexión necesaria entre el código de tu proyecto y la base de datos. Este proceso es esencial para asegurar que tu aplicación pueda interactuar correctamente con la información almacenada en la base de datos.

3. Navegación:

4.1 Mapa del sitio



Sobre MARVEL



HISTORIA

Marvel Comics es un gigante de los cómics, creando en 1939. Los CEO actuales de Marvel son Kevin Feige, presidente de Marvel Studios y el director principal del Universo Cinematográfico de Marvel y Iker Perlmutter, ex CEO de Marvel Entertainment, responsable de operaciones y estrategia.



CEO

San Lee y Jack Kirby son dos de los principales fundadores de Marvel Comics. Junto a Stan Lee, crearon icónicos superhéroes como Spider-Man, Hulk, Los Cuatro Fantásticos y X-Men, dejando un legado duradero en la cultura pop.



FUNDADORES

Comics

The Amazing Spider-Man
Las aventuras de Peter Parker, un estudiante de secundaria que adquiere habilidades de araña y se convierte en el amado superhéroe Spider-Man.
\$400

[COMPRAR](#)

The Avengers
Reúnen a héroes como Iron Man, Capitán América, Thor, Hulk y otros para formar un equipo que lucha contra amenazas que ningún héroe podría enfrentar solo.
\$400

[COMPRAR](#)

X-Men
Un grupo de mutantes dirigidos por el Profesor X, luchan por la coexistencia pacífica entre mutantes y humanos, defendiendo un mundo que los teme.
\$400

[COMPRAR](#)

Fantastic Four
Mr. Fantástico, la Mujer Invisible, la Antorcha Humana y La Cosa exploran el espacio y enfrentan desafíos cósmicos como equipo.
\$400

[COMPRAR](#)

[VER CATALOGO](#)



DATOS CURIOSOS



El Universo Cinematográfico de Marvel (MCU)

La franquicia cinematográfica de Marvel es la más exitosa en la historia del cine, con películas interconectadas y exitosas como 'Avengers: Endgame', que se convirtió en la película más taquillera de todos los tiempos.



El origen del nombre "Marvel"

La empresa ha tenido varios nombres antes de convertirse en Marvel Comics; se llamó Timely Publications y Atlas Comics antes de adoptar el nombre que todos conocemos hoy.

FRASES EMBLEMATICAS

“With great power, comes great responsibility”

-TIO BEN (Spider-Man(2002))



Vista login



Vista de registro



Catalogo Personajes

INICIAR SESION



Lista de usuarios registrados

[Registrar Nuevo Usuario](#)

nombre correo acciones

Sajit sajitlove@hotmail.com admin

[Borrar](#)
[Editar Usuario](#)

Vista posterior al inicio de sesión como administrador.



Vista edición de usuarios.

MARVEL

Catalogo Personajes Cerrar Sesión Carrito INICIAR SESIÓN

CATALOGO

Buscar:
Nombre del personaje...
Buscar

 3-D Man Agregar al Carrito Ver más	 A-Bomb (HAS) Agregar al Carrito Ver más	 A.I.M. Agregar al Carrito Ver más	 Aaron Stack Agregar al Carrito Ver más	 Abomination (Emil Blonsky) Agregar al Carrito Ver más
 Abomination (Ultimate) Agregar al Carrito Ver más	 Absorbing Man Agregar al Carrito Ver más	 Abyss Agregar al Carrito Ver más	 Abyss (Age of Apocalypse) Agregar al Carrito Ver más	 Adam Destine Agregar al Carrito Ver más
 Adam Warlock Agregar al Carrito Ver más	 Aegis (Trey Rollins) Agregar al Carrito Ver más	 Aero (Aero) Agregar al Carrito Ver más	 Agatha Harkness Agregar al Carrito Ver más	 Agent Brand Agregar al Carrito Ver más
 Agent X (Nijo) Agregar al Carrito Ver más	 Agent Zero Agregar al Carrito Ver más	 Agents of Atlas Agregar al Carrito Ver más	 Aginar Agregar al Carrito Ver más	 Air-Walker (Gabriel Lan) Agregar al Carrito Ver más

Vista posterior al inicio de sesión como Operador(Vista Catalogo).

MARVEL

Catalogo Personajes



A.I.M.



Eliminar del Carrito

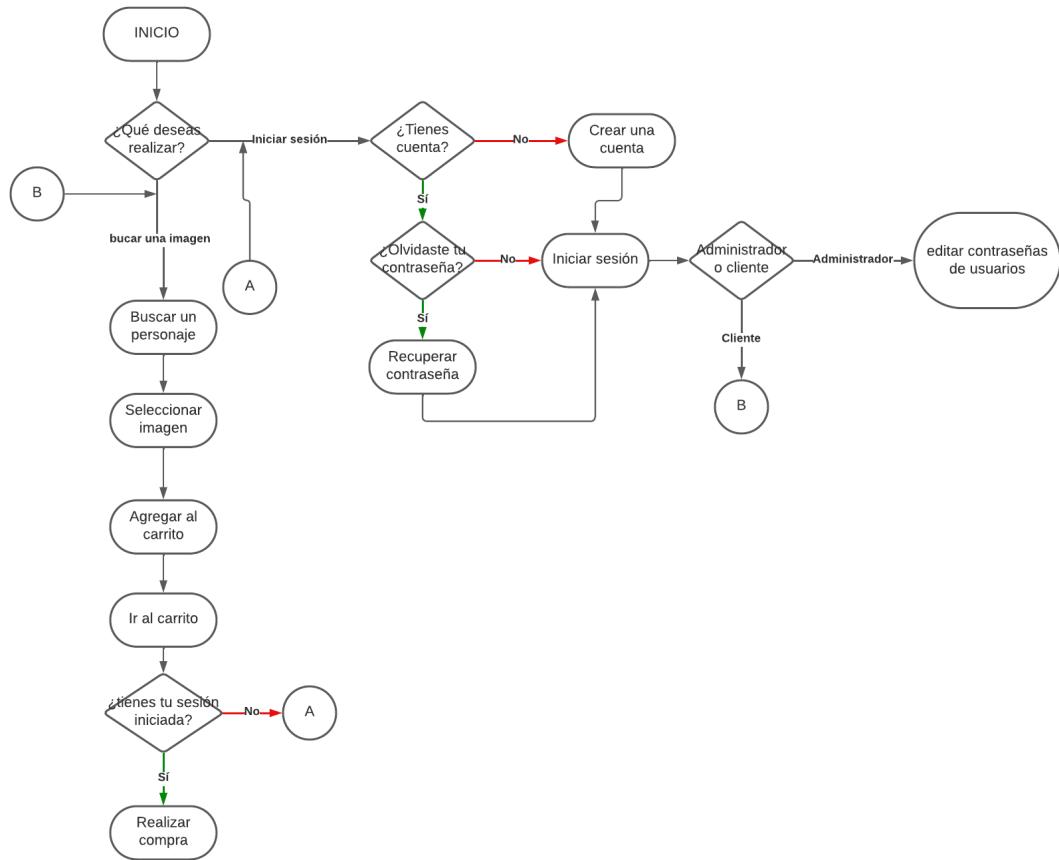
A-Bomb (HAS)



Eliminar del Carrito

Carrito

4.2 diagramas de flujo



4.4 funcionalidades

a. Visualización de Imágenes de Marvel

- Descripción:

- Implementa una interfaz de usuario que muestre imágenes disponibles de Marvel.

- Desarrollo:

- Utiliza HTML y CSS para estructurar y diseñar la página de visualización.
- Emplea JavaScript para cargar y mostrar dinámicamente las imágenes en la interfaz.

b. Carrito de Compras

- Descripción:

- Permite a los usuarios agregar imágenes al carrito y revisar su selección antes de comprar.

- Desarrollo:

- Implementa botones de "Aregar al Carrito" junto a cada imagen.
- Utiliza una estructura de datos (puede ser un array en JavaScript) para almacenar los elementos del carrito.
- Desarrolla una sección donde los usuarios pueden ver y gestionar los elementos en su carrito.

c. Autenticación de Usuarios

- Descripción:

- Proporciona funciones para que los usuarios creen cuentas, inicien sesión y recuperar contraseñas.

- Desarrollo:

- Diseña formularios de registro e inicio de sesión.
- Implementa lógica de autenticación del lado del servidor (puede ser con Node.js y Express).
- Crea mecanismos para restablecer contraseñas utilizando tokens de restablecimiento.

d. Administración de Usuarios

- Descripción:

- Permite a los administradores agregar y eliminar usuarios.

- Desarrollo:

- Implementa una interfaz de administrador accesible solo para usuarios autorizados.
- Desarrolla funciones para agregar y eliminar usuarios desde esta interfaz.

4.5 casos de uso y escenarios

4.6 Reglas de navegación

4.API

La interfaz de programación de aplicaciones basada en REST (APIREST) representa la conexión esencial entre la base de datos y la aplicación principal. Previo a la recuperación o envío de datos a la base de datos, es imperativo que dichos datos atraviesen la APIREST. Esta interfaz se encuentra interconectada con el proyecto principal mediante servicios, los cuales hacen uso de la dependencia node-fetch.

En este contexto, los servicios envían datos en formato de cadena JSON (mediante `Json.stringify`) a través de una URL específica en el puerto 2000. Los datos que los servicios transmiten son recuperados directamente desde los controladores.

5. Seguridad

Se implementaron 2 herramientas, una para la navegación segura dentro de la página web y otra para la encriptación de datos sensibles, a continuación su explicación.

1. Navegación Segura:

jsonwebtoken:

- Descripción: Utilizamos una herramienta llamada jsonwebtoken para garantizar que la navegación en la página web sea segura y autenticada.
- ¿Cómo Funciona?: Después de iniciar sesión, el servidor crea un "token" especial que se envía al navegador del usuario. Este token es como una credencial que se incluye en cada solicitud para confirmar que el usuario está autorizado. Esto ayuda a proteger la identidad del usuario durante su interacción con la aplicación.

2. Encriptación de Datos Sensibles:

crypto.createHash:

- Descripción: Empleamos `crypto.createHash` para asegurar que los datos sensibles, como contraseñas, estén protegidos mediante un proceso llamado "hashing".
- Cómo Funciona: Antes de guardar las contraseñas en la base de datos, las convertimos en una forma irreversible llamada "hash". Este hash es único para cada contraseña y proporciona una capa adicional de seguridad, ya que no se puede revertir para revelar la contraseña original.

