



Clase 26. Programación Backend

Estrategias de autenticación con redes sociales



OBJETIVOS DE LA CLASE

- Conocer acerca del inicio de sesión mediante redes sociales.
- Realizar el inicio sesión con Twitter.
- Incorporar el módulo JSON Web Token y realizar el inicio de sesión.

CRONOGRAMA DEL CURSO

Clase 25



**Autorización y
autenticación**

Clase 26



**Estrategias de
autenticación con redes
sociales**

Clase 27



**Proceso principal del
servidor**

PASSPORT: ESTRATEGIAS DE AUTENTICACIÓN CON REDES SOCIALES

PASSPORT-TWITTER



TwitterStrategy



- La estrategia de Twitter permite a los usuarios iniciar sesión en una aplicación web utilizando su cuenta de Twitter.
- Internamente, la autenticación de Twitter funciona con **OAuth 1.0a**.
- El soporte para Twitter se implementa mediante el módulo ***passport-twitter***.



Twitter for Developers



- Para habilitar la autenticación a través de Twitter, primero tenemos que crear una Twitter App utilizando el Twitter Developers.
- Una vez creada la App, necesitamos los datos de App ID y App Secret asignados a la app. Debemos además, especificar una URL para redireccionar al usuario una vez que inicia sesión con Twitter.
- Para empezar a crear esta app, debemos primero ingresar con nuestra cuenta de Twitter al [siguiente link](#)

Vamos a la acción 😁



***¿Alguna pregunta hasta
ahora?***



BREAK

¡5/10 MINUTOS Y VOLVEMOS!

JWT ***(JSON Web Token)***



¿De qué se trata?

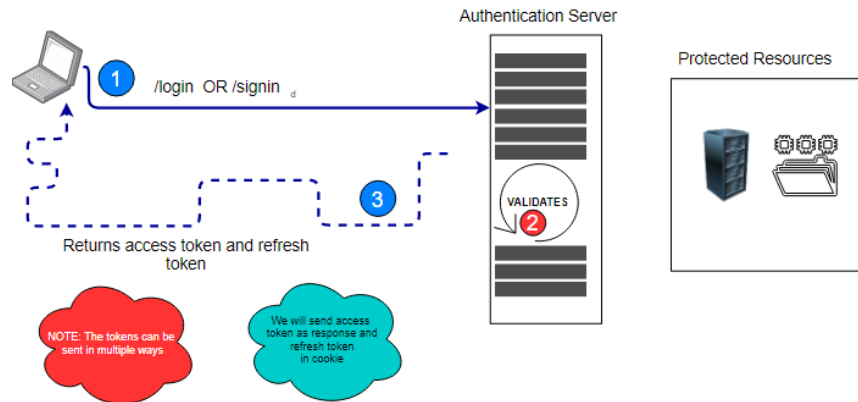
- JSON Web Token es un método estándar y abierto para representar reclamaciones de forma segura entre dos partes.
- [JWT.IO](https://jwt.io) nos permite decodificar, verificar y generar JWT.
- Básicamente, los JWT son cadenas de datos que se pueden utilizar para autenticar e intercambiar información entre un servidor y un cliente.



Flujo de funcionamiento

El flujo de funcionamiento es el siguiente:

1. El cliente envía credenciales al servidor.
2. El servidor verifica las credenciales, genera un JWT y lo envía como respuesta.
3. Las solicitudes posteriores del cliente tienen un JWT en los *headers* de la solicitud.
4. El servidor valida el token y, si es válido, proporciona la respuesta solicitada.

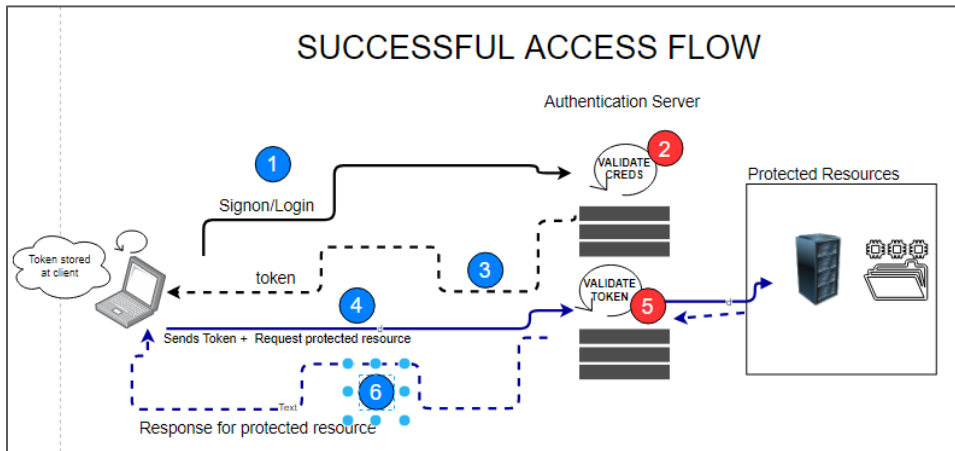




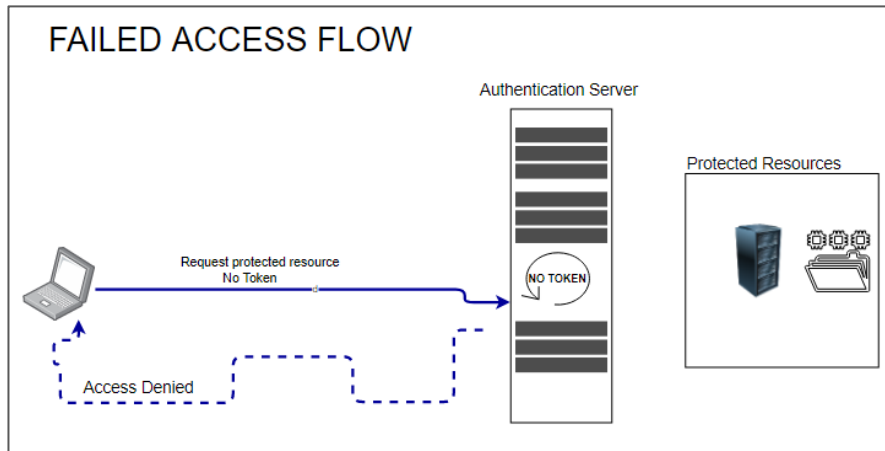
Flujo de funcionamiento

- Las solicitudes posteriores del cliente tienen un JWT en los *headers* de la solicitud.
- El servidor valida el token y, si es válido, proporciona la respuesta solicitada.
- Si no se valida el token, se niega el acceso.

SUCCESSFUL ACCESS FLOW



FAILED ACCESS FLOW



Vamos al VC 



***¿Alguna pregunta hasta
ahora?***

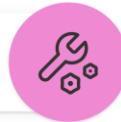


INICIO DE SESIÓN CON JWT

Tiempo: 15 minutos

Inicio de Sesión con JWT

Desafío
generico



Tiempo: 15 minutos

Utilizando la estructura de un servidor Nodejs express, realizar el siguiente desarrollo implementando las sesiones de usuario a través de JWT (JSON Web Token). Dichas sesiones tendrán un tiempo de expiración de 1 minuto.

Requerimientos del sistema:

1. Tenga un formulario de registro de usuario (nombre, password y dirección) que almacene dicha información en un array en memoria.
2. Un formulario de login (nombre y password) para permitir a los usuarios registrados iniciar una sesión.
3. Si accede un usuario no registrado o las credenciales no corresponden, el servidor enviará un error (puede ser a través de un objeto plano o de una plantilla).

Inicio de Sesión con JWT

Desafío
generico

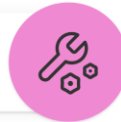


Tiempo: 15 minutos

4. Si se quiere registrar un usuario que ya está registrado, el servidor enviará un error (puede ser a través de un objeto plano o de una plantilla).
5. Al cargar la página principal ('/'), si existe una sesión iniciada, se mostrarán los datos del usuario en cuestión (obtenidos mediante una consulta con el token debidamente adjunto en el encabezado de la petición de datos). Caso contrario, se le redirigirá a una página que le informe que no posee autorización, y le ofrecerá ir a la página de login.
6. Implementar el cierre de sesión con un botón *logout* en la página de datos de usuario, que redirige la vista al formulario de login.
7. Como extra podemos implementar un contador de visitas, que se muestre sobre la vista de datos.

Inicio de Sesión con JWT

Desafío
generico



Tiempo: 15 minutos

Nota:

- No utilizar passport.
- No utilizar session de express.
- Priorizar la funcionalidad del backend (especialmente el manejo de firma y verificación de tokens) antes que el desarrollo de las vistas.
- Se puede usar fetch para hacer la petición a la API de datos (adjuntando la cabecera necesaria para la autenticación)
- También se pueden usar plantillas públicas (handlebars) para la carga dinámica de la página de datos, dependiendo del resultado de la autenticación.



INICIO DE SESIÓN

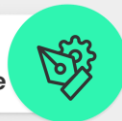
Retomemos nuestro trabajo para agregar inicio de sesión a nuestro sitio.

INICIO DE SESIÓN

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



>> Consigna:

Implementar sobre el entregable que venimos realizando un mecanismo de autenticación. Para ello:

Se incluirá una vista de registro, en donde se pidan email y contraseña. Estos datos se persistirán usando MongoDB, en una (nueva) colección de usuarios, cuidando que la contraseña quede encriptada (sugerencia: usar la librería *bcrypt*).

Una vista de login, donde se pida email y contraseña, y que realice la autenticación del lado del servidor a través de una estrategia de passport local.

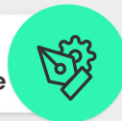
Cada una de las vistas (logueo - registro) deberá tener un botón para ser redirigido a la otra.

INICIO DE SESIÓN

Formato: link a un repositorio en Github con el proyecto cargado.

Sugerencia: no incluir los node_modules

Desafío
entregable



Una vez logueado el usuario, se lo redirigirá al inicio, el cual ahora mostrará también su email, y un botón para desolguearse.

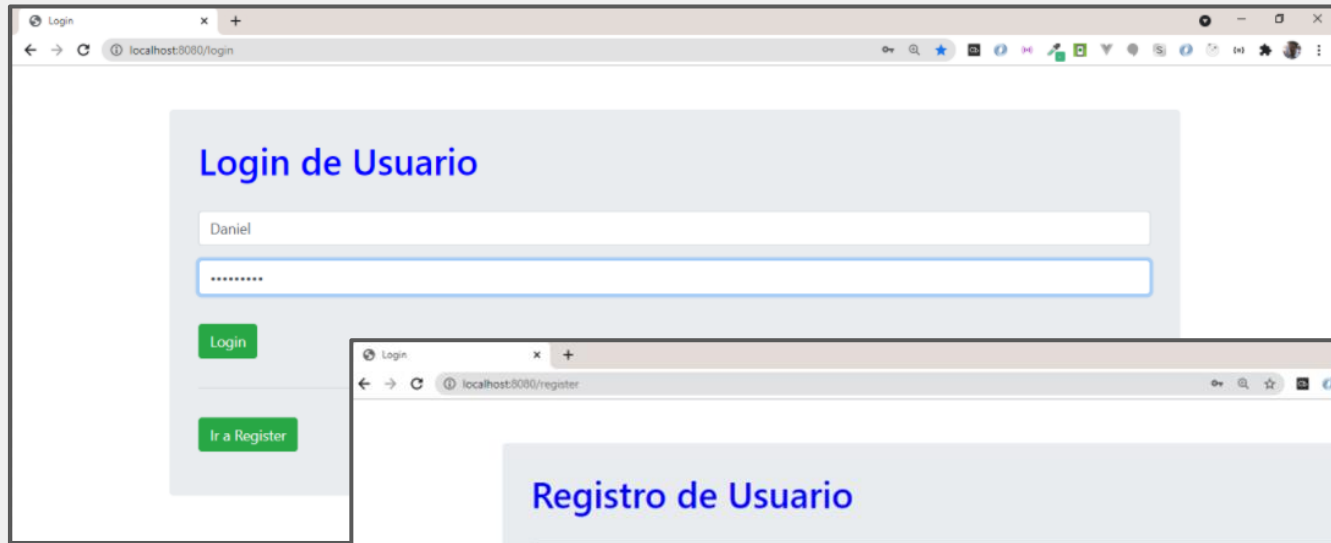
Además, se activará un espacio de sesión controlado por la sesión de passport. Esta estará activa por 10 minutos y en cada acceso se recargará este tiempo.

Agregar también vistas de error para login (credenciales no válidas) y registro (usuario ya registrado).

El resto de la funciones, deben quedar tal cual estaban el proyecto original.

>> Ejemplos de vistas de acceso a continuación.

>> Ejemplo

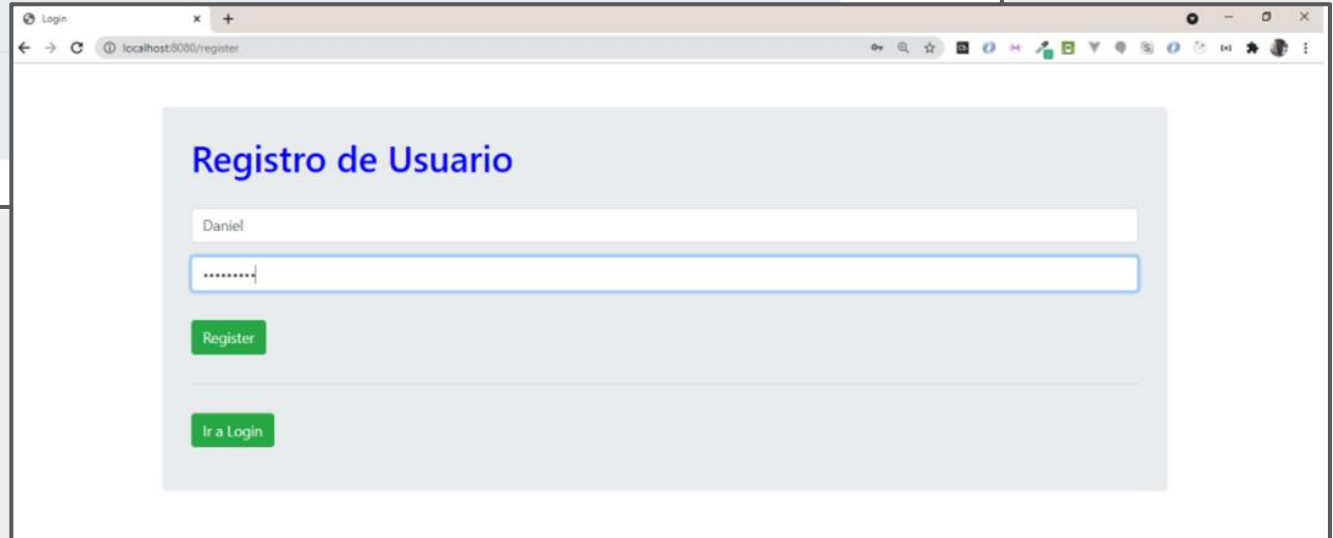


Login de Usuario

Daniel

Login

[Ir a Register](#)



Registro de Usuario

Daniel

Register

[Ir a Login](#)

Vista de Productos

localhost:8080/login

Bienvenido Daniel

Desloguear

Ingrese Producto

Nombre

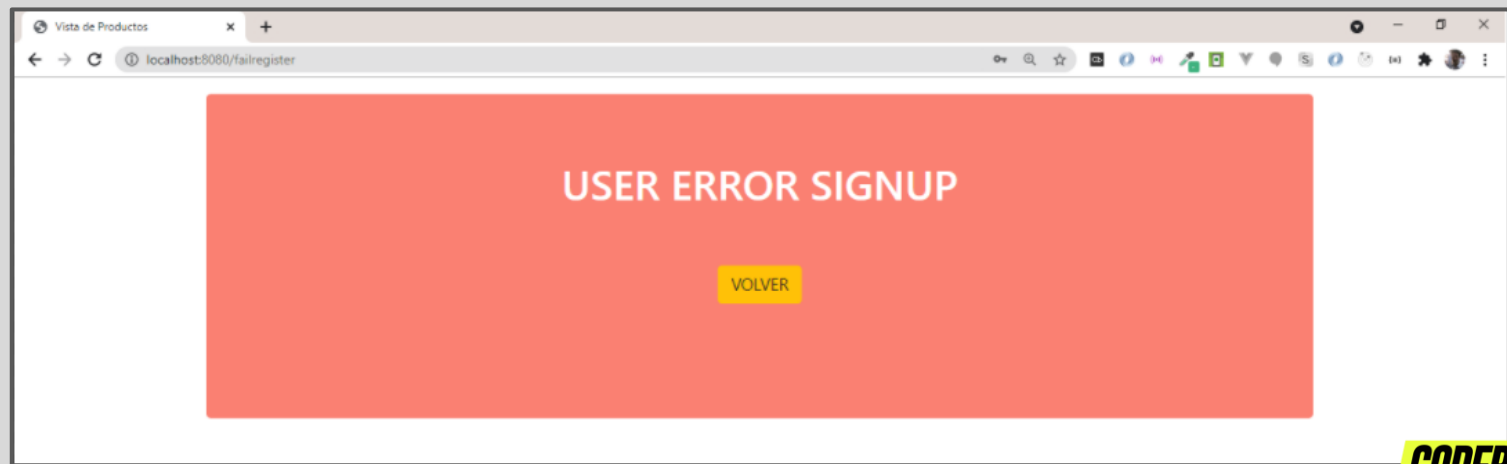
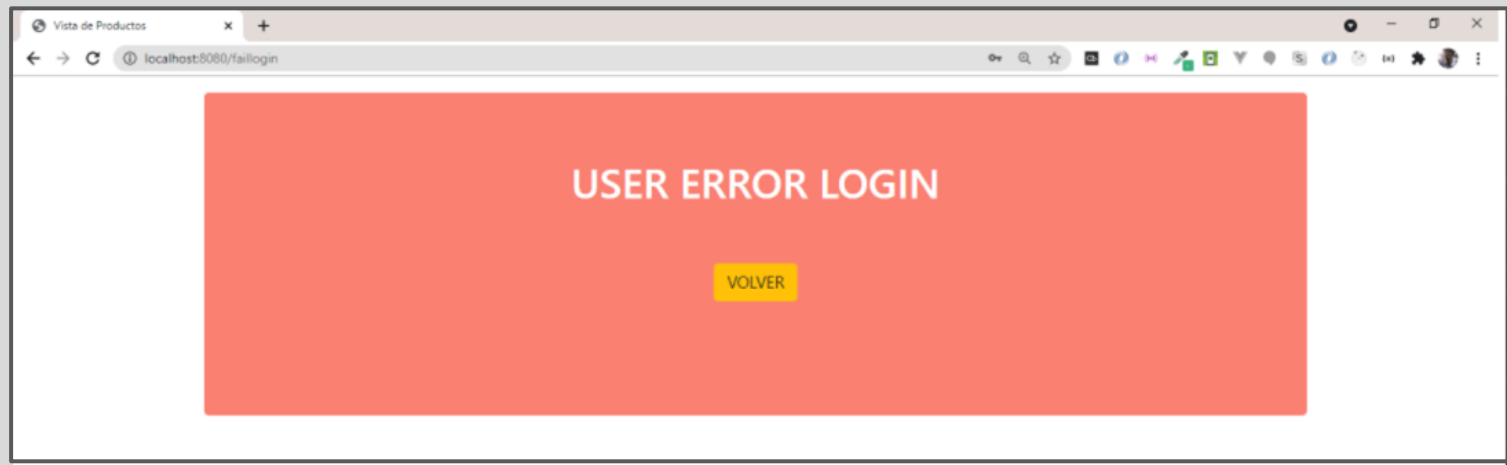
Precio

Foto URL

Enviar

Lista de Productos

Nombre	Precio	Foto
--------	--------	------



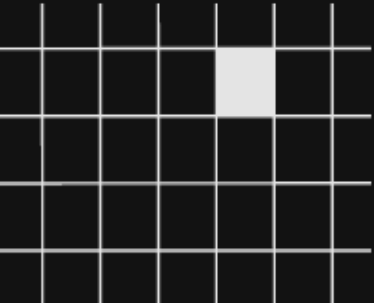
¿PREGUNTAS?





¡MUCHAS GRACIAS!

Resumen de lo visto en clase
hoy:

- Passport-Facebook.
 - Passport-Twitter.
 - JWT.
- 



OPINA Y VALORA ESTA CLASE

#DEMOCRATIZANDO LA EDUCACIÓN