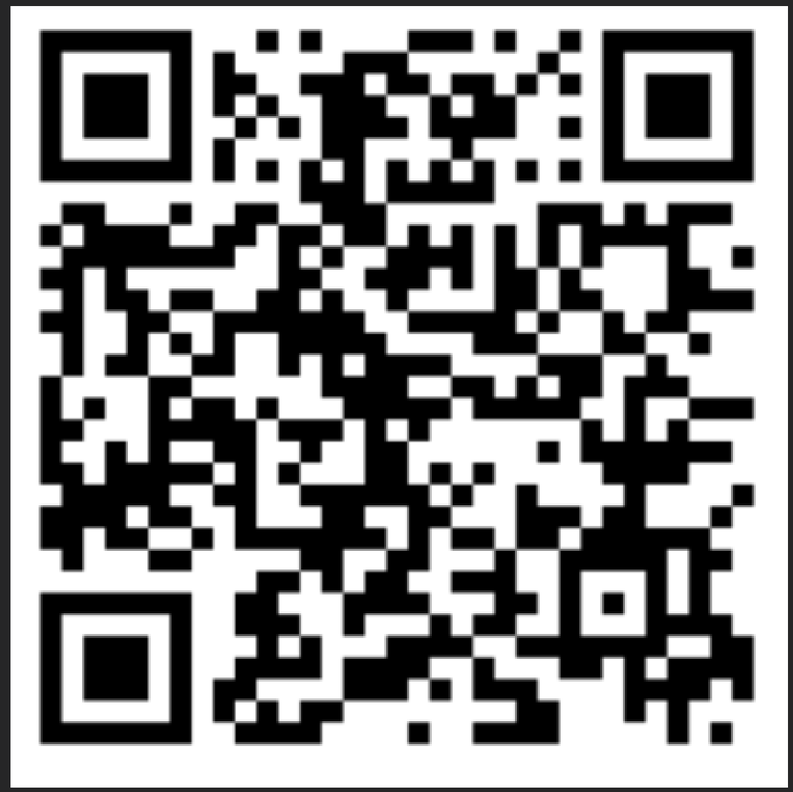

BCAMP

Challenger 01
Front End

BOOTCAMP

CONTENIDO

1. Exposición de Pablo Magiorano
2. Quiz
3. Challenger 01
4. JWT
5. Detalle de Servicios
6. Ronda de preguntas

Challenger 01

Desafío de Desarrollo Front-End

Inicio: Hoy, miércoles, a las 21:00
Fin: Domingo a las 23:59

SOCIUS

BOOTCAMP

¡BIENVENIDOS AL DESAFÍO DE DESARROLLO FRONT-END! ESTE DESAFÍO TIENE COMO OBJETIVO EVALUAR TUS HABILIDADES EN ANGULAR O REACT MEDIANTE LA CREACIÓN DE UNA APLICACIÓN WEB FUNCIONAL. AQUÍ ESTÁN LOS DETALLES DEL DESAFÍO:

LOGIN - CV

01. Crear un Login

• Debes crear una pantalla de inicio de sesión donde los usuarios puedan autenticarse utilizando un formulario.

• El formulario debe incluir campos para el nombre de usuario y la contraseña.

• No es necesario implementar una autenticación real; puedes simular la autenticación con datos estáticos.

02. Pantalla de CV

Una vez autenticados, los usuarios deben ser redirigidos a una pantalla donde puedan ver su CV (Curriculum Vitae).

• La pantalla de CV debe mostrar información básica del usuario, como nombre, experiencia, educación y habilidades.

CONSUMO DE SERVICIOS

03. Listado de Tareas.

• Debes implementar una sección donde los usuarios puedan ver una lista de tareas.
• La lista de tareas debe ser obtenida a través de una API o un servicio simulado.
• Cada tarea debe mostrar información relevante como título, descripción y estado (completada/no completada).

04. Cambio de estado de las tareas.

• Debes implementar una sección donde los usuarios puedan finalizar las tareas.
• La finalización de tareas debe ser a través de una API, indicando Hash de le tarea + Comentario de cierre.
•

¿Qué es JWT (JSON Web Token)?

Un JSON Web Token (JWT) es un estándar abierto (RFC 7519) que define una forma compacta y autónoma para transmitir información de manera segura entre partes como un objeto JSON. Esta información se puede verificar y confiar porque está firmada digitalmente.

SOCIUS

Header (Encabezado):

El encabezado generalmente consta de dos partes: el tipo de token, que es JWT, y el algoritmo de firma que se está utilizando, como HMAC SHA256 o RSA.

Payload (Carga Útil):

La carga útil contiene las afirmaciones (claims). Las afirmaciones son declaraciones sobre una entidad (generalmente, el usuario) y datos adicionales. Hay tres tipos de afirmaciones:

- Afirmaciones registradas: son un conjunto de afirmaciones predefinidas que no son obligatorias pero son recomendadas, como iss (emisor), exp (expiración), sub (tema), aud (audiencia), etc.
- Afirmaciones públicas: pueden ser definidas libremente por aquellos que usen JWT.
- Afirmaciones privadas: son las afirmaciones personalizadas que crean y comparten entre partes que acuerdan usarlas.

Signature (Firma):

Para crear la firma, necesitas tomar el header codificado, el payload codificado, un secreto, y el algoritmo especificado en el header. Por ejemplo, si estás usando el algoritmo HMAC SHA256, la firma se crea de la siguiente manera:

INSTALACIÓN

¿Cómo funciona JWT?

1. Autenticación:
• Cuando el usuario inicia sesión utilizando sus credenciales, el servidor crea un JWT y se lo devuelve al cliente.
• El cliente almacena el JWT de forma segura, típicamente en localStorage o una cookie.
2. Autorización:
• Cada vez que el cliente realiza una solicitud al servidor, incluye el JWT en los encabezados de autorización de la solicitud.
• El servidor recibe la solicitud, verifica el JWT para asegurarse de que es válido y luego procesa la solicitud.

Ventajas de JWT

• Compacto: JWT es compacto, lo que lo hace adecuado para ser transmitido en URL, encabezados HTTP o en el cuerpo de una petición HTTP.
• Autónomo: El token contiene toda la información necesaria sobre el usuario, eliminando la necesidad de consultas adicionales a la base de datos.
• Seguro: JWT puede ser firmado usando un algoritmo de clave secreta (HMAC) o un par de claves pública/privada (RSA o ECDSA).