

APLICACIONES WEB ORIENTADAS A SERVICIOS

REMEDIAL UNIDAD 2

Docente: Addai Guerrero

Oscar Ariel Quintana Merino
TID41M

Reporte de Integración de API's

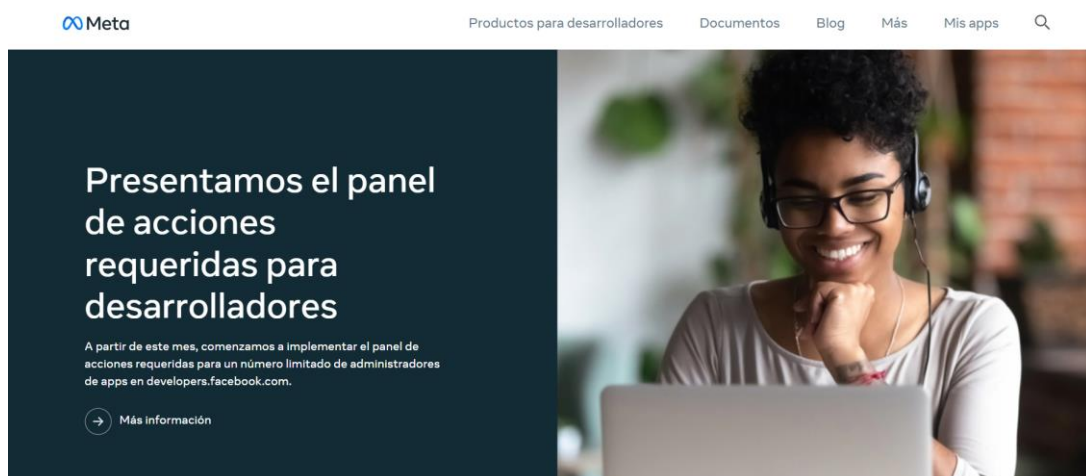
En este informe se documentará el proceso de integración de API's de redes sociales en un sitio web, centrándonos específicamente en la API de Facebook para implementar la funcionalidad de inicio de sesión en una página web. La integración de las API de redes sociales ofrece a los usuarios la posibilidad de autenticarse utilizando sus credenciales de plataformas populares, simplificando el proceso de registro y proporcionando una experiencia de usuario más fluida.

El objetivo principal de esta integración es permitir a los usuarios iniciar sesión en la página web utilizando sus cuentas de Facebook, eliminando la necesidad de crear y recordar credenciales adicionales. Este enfoque no solo simplifica el proceso de registro, sino que también mejora la seguridad al utilizar la autenticación de una plataforma confiable como Facebook.

El proceso de integración consta de varios pasos:

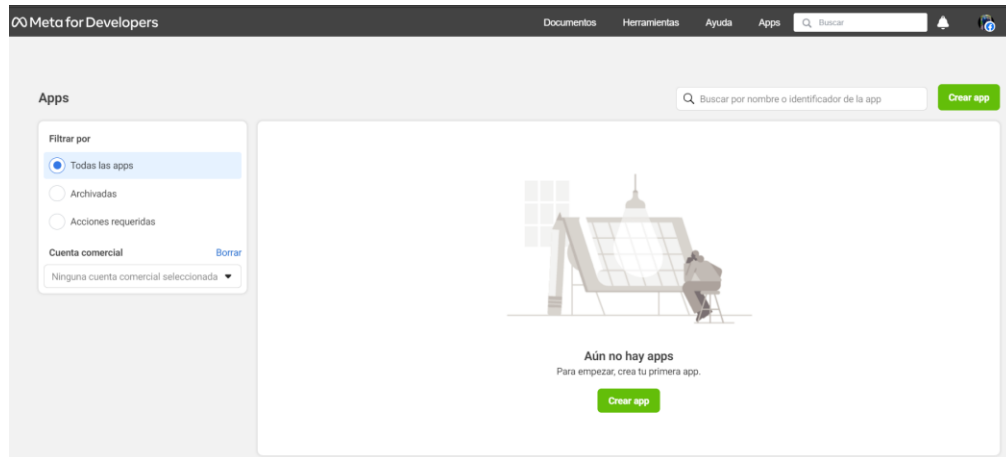
1. Acceso a Facebook for Developers:

- Visitar el portal de **Facebook for Developers** e iniciar sesión con una cuenta de Facebook.



2. Creación de una Nueva Aplicación:

- Dentro del panel de “Mis apps”, se selecciona "Crear Aplicación" y se elige "Crear Aplicación" nuevamente.



- Después nos preguntará que queremos que realice la aplicación, en este caso autenticaremos y pediremos datos de usuario para realizar un Login con Facebook.

- Ahora se proporcionará un nombre único para la aplicación y una dirección de correo electrónico asociada. En este caso se llamará “MiAppLogin”.

- Al finalizar de rellenar los datos le daremos al botón de “Crear app”.

The screenshot shows the 'Crear una app' (Create an app) form. On the left, there's a sidebar with three items: 'Agregar caso d...' (Add use case), 'Login type', and 'Detalles de la app' (App details), with the last one being selected. The main form area has a title 'Agrega un nombre para la app' (Add a name for the app) and a text input field containing 'MiAppLogin'. Below this is a section for 'Correo electrónico de contacto de la app' (App contact email) with a text input field containing 'ariel_quintana25@hotmail.com'. There's also a section for 'Cuenta comercial' (Commercial account) with a dropdown menu showing 'No seleccionaste ninguna cuenta del administrador comercial'. At the bottom, there are links for 'Condiciones de la plataforma de Meta' and 'Políticas para desarrolladores', and buttons for 'Anterior' (Previous) and 'Crear app' (Create app).

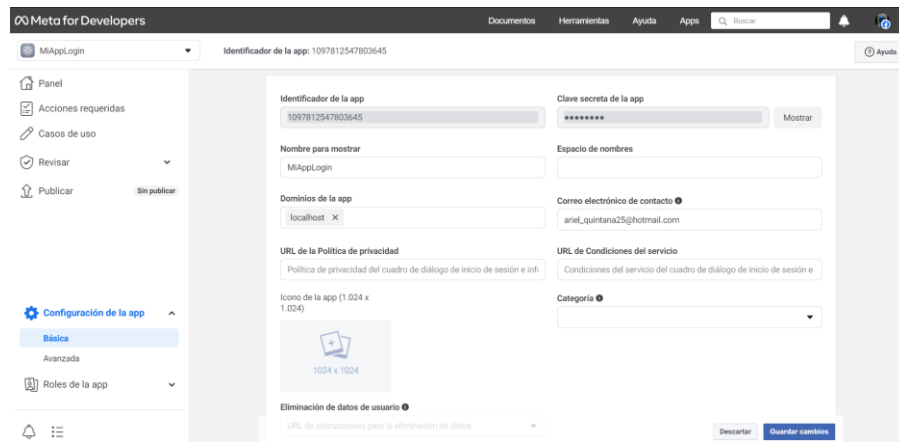
3. Configuración de la Aplicación:

- Después de crear la aplicación, accederás al panel de configuración. Aquí, seleccionamos la pestaña "Configuración" y completa la información requerida, como el dominio del sitio web y las categorías de la aplicación.

The screenshot shows the 'Panel' (Dashboard) in the Meta Developer console. The top navigation bar includes 'Meta for Developers', 'Documentos', 'Herramientas', 'Ayuda', and 'Apps'. The left sidebar has a 'Panel' section with sub-items: 'Acciones requeridas', 'Casos de uso', 'Revisar', and 'Publicar'. The main content area is titled 'Panel' and shows a progress bar for 'Crear y publicar esta app' (Create and publish this app). The progress bar indicates '0 de 3 pasos completados' (0 of 3 steps completed). The first step is '1. Personaliza la app' (Customize the app), which includes options like 'Personalizar agregar el botón de inicio de sesión con Facebook' and 'Explora y agrega más casos de uso'. The second step is '2. Preparar y enviar a revisión de apps' (Prepare and send for app review), and the third step is '3. Publica la app' (Publish the app).

4. Obtención del App ID y App Secret:

- En la sección "Configuración Básica", se encontrará el App ID de la aplicación. Junto con el App ID, se generará un App Secret. Ambas claves son esenciales para la autenticación y la comunicación segura con la API de Facebook desde el sitio web.



5. Configuración del Inicio de Sesión con Facebook:

- En el panel de desarrolladores, navegamos a la sección "Inicio de Sesión con Facebook" en el menú lateral. Aquí, se configura las opciones de inicio de sesión, como los permisos requeridos y las URL de redirección.
- En este apartado principalmente se debe de introducir principalmente el URI de direccionamiento de OAuth.
- **NOTA:** Por ultimas actualizaciones con la política de Facebook, ya no se permite utilizar la API de Login en páginas con HTTP, sino que solamente se pueden utilizar en paginas de HTTPS. Para evitar errores al utilizar Localhost, se utilizará la herramienta **ngrok**, la cual nos ayudará a crear un servidor local con credenciales de seguridad HTTPS, esta configuración se verá después al codificar la aplicación. Por ende, se introducirá el link de ngrok en el apartado de URI de direccionamiento de OAuth para permitir la conexión de la API con el sitio web.

Configuración del inicio de sesión con Facebook
Volver

Personaliza tu caso de uso para que la app funcione como quieras.

Agrega fácilmente el inicio de sesión con Facebook a tu app con nuestra [Guía de inicio rápido](#)

Configuración del cliente de OAuth

☒ **Inicio de sesión del cliente de OAuth**
Activa el flujo estándar de token de cliente de OAuth. Protege tu app y evita un uso incorrecto de esta especificando qué URI de redireccionamiento del token quieres permitir mediante las siguientes opciones. Desactiva esta opción globalmente si no la usas. [\[?\]](#)

☒ **Inicio de sesión de OAuth web**
Activa el inicio de sesión del cliente de OAuth basado en web. [\[?\]](#)

☐ **Forzar reautenticación de OAuth web**
Si se activa, solicita a las personas que ingresen su contraseña de Facebook para iniciar sesión en el sitio web. [\[?\]](#)

☐ **Aplicar HTTPS**
Aplica el uso de HTTPS para los URI de redireccionamiento y el SDK de JavaScript. Muy recomendado. [\[?\]](#)

☐ **Inicio de sesión de OAuth de navegador integrado**
Activa los URI de redireccionamiento de la vista web para el inicio de sesión del cliente de OAuth. [\[?\]](#)

☒ **Usar modo estricto para URI de redireccionamiento**
Permite solo los redireccionamientos que coinciden exactamente con los URI de redireccionamiento de OAuth válidos. Muy recomendado. [\[?\]](#)

URI de redireccionamiento de OAuth válidos

El parámetro "redirect_uri" especificado manualmente que se usa con el inicio de sesión en la web debe coincidir exactamente con uno de los URI de esta lista. El SDK para JavaScript también usa esta lista para los navegadores de la app que bloquean las ventanas emergentes. [\[?\]](#)

Copiar en el portapapeles

☐ **Iniciar sesión desde dispositivos**
Activa el flujo de acceso del cliente de OAuth a dispositivos como un Smart TV [\[?\]](#)

☒ **Iniciar sesión con el SDK para JavaScript**
Activa el inicio de sesión y la funcionalidad de sesión iniciada con el SDK para JavaScript [\[?\]](#)

Dominios permitidos para el SDK para JavaScript

Descartar
Guardar cambios

6. Configuración de Productos:

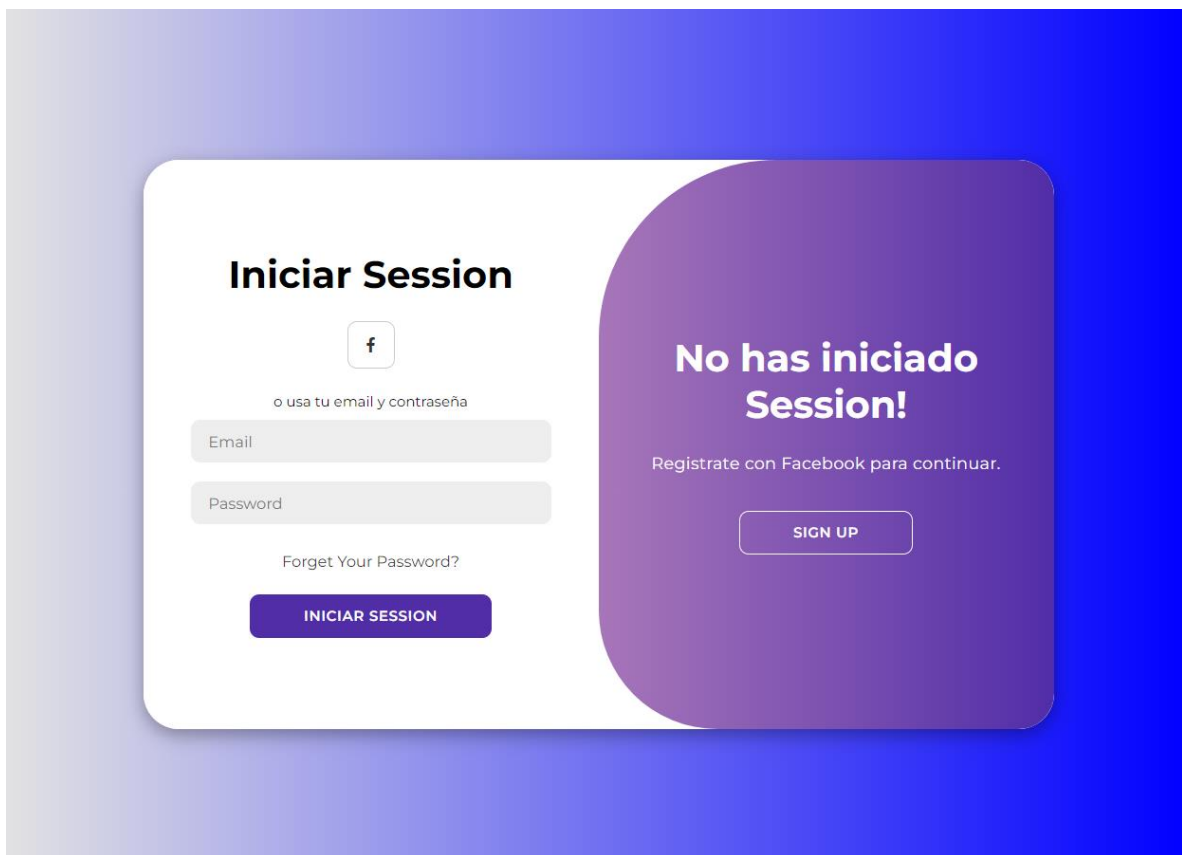
- En la pestaña "Configuración" de la aplicación, exploramos la sección "Productos". Aquí se puede habilitar y configurar productos específicos de Facebook que se desea utilizar, como el inicio de sesión con Facebook o la obtención de datos del usuario.

Permisos ↑			
Permisos	Llamadas a la API	Estado	Acción
email El permiso email permite que tu app lea la dirección de correo electrónico principal de una persona. <i>Descripción completa</i> <i>Requisitos</i>	—	—	Agregar
public_profile El permiso public_profile permite a una app leer los campos del perfil público predeterminados en el nodo User . Este permiso se otorga automáticamente a todas las apps. <i>Descripción completa</i> <i>Requisitos</i>	0	Listo para la prueba	

Creación del sitio web.

Para la creación del sitio web, utilizaremos el siguiente diseño de inicio de sesión que incorpora la autenticación a través de la API de Facebook. Este diseño proporciona a los usuarios la opción de iniciar sesión utilizando sus cuentas de Facebook, simplificando así el proceso de registro y mejorando la experiencia general del usuario.

En el código de HTML y CSS nos ayudará a proporcionar una estructura del diseño de la página web, mientras que el JavaScript se encarga de la integración con la API de Facebook. Además, se ha implementado un panel de bienvenida que se actualiza dinámicamente con el nombre y la foto de perfil del usuario después de iniciar sesión exitosamente.



Implementación del SDK

Para la implementación de la API de inicio de sesión en Facebook, se incluirá el SDK de Facebook en el sitio web incluyéndolo en el HTML y inicializando los valores en el JavaScript. El SDK (Software Development Kit) proporciona las herramientas y funciones necesarias para interactuar con la API de Facebook.

```
<script src="login.js"></script>
<script async defer crossorigin="anonymous" src="https://connect.facebook.net/en_US/sdk.js#xfbml=1&version=v10.0" nonce="YIz1GzBR"></script>
</body>
```

Dentro de un script JavaScript, se inicializará el SDK de Facebook con la información de la aplicación obtenida de Facebook for Developers. Este paso es esencial para establecer la conexión entre el sitio web y la API de Facebook:

```
window.fbAsyncInit = function() {
  FB.init({
    appId      : 'tu_app_id',
    cookie     : true,
    xfbml     : true,
    version    : 'v18.0'
  });

  FB.AppEvents.logPageView();
};
```

Cambiaremos la variable appId por nuestro AppID la cual se proporciona en el panel de configuración en la página de Facebook for Developers.

```
window.fbAsyncInit = function() {
  FB.init({
    appId      : '658900186139962',
    cookie     : true,
    xfbml     : true,
    version    : 'v18.0'
  });

  FB.AppEvents.logPageView();
};
```


Manejo del Inicio de Sesión y Obtención de Información del Usuario:

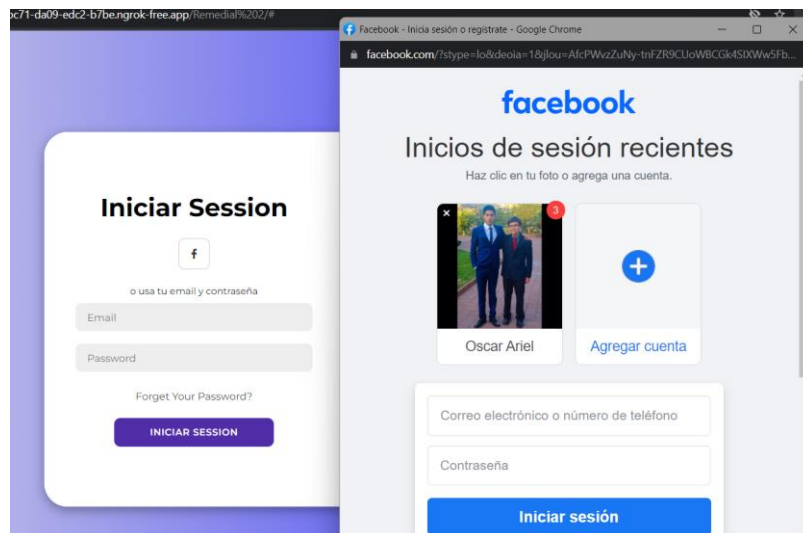
Se implementarán funciones para manejar el inicio de sesión y obtener información del usuario mediante la API de Facebook. Esto incluye el uso de FB.login para el inicio de sesión y FB.api para la obtención de datos del usuario:

```
fbLogin.addEventListener('click', function() {
  FB.login(function(response) {
    if (response.authResponse) {
      console.log('Hola! Cargando la informacion... ');
      FB.api('/me', function(response) {
        console.log('Buen Dia , ' + response.name + '.');
        updateWelcomePanel();
      });
    } else {
      console.log('Usuario Cancelo el login o no se autorizo completamente. ');
    }
  }, {scope: 'public_profile,email'});
});

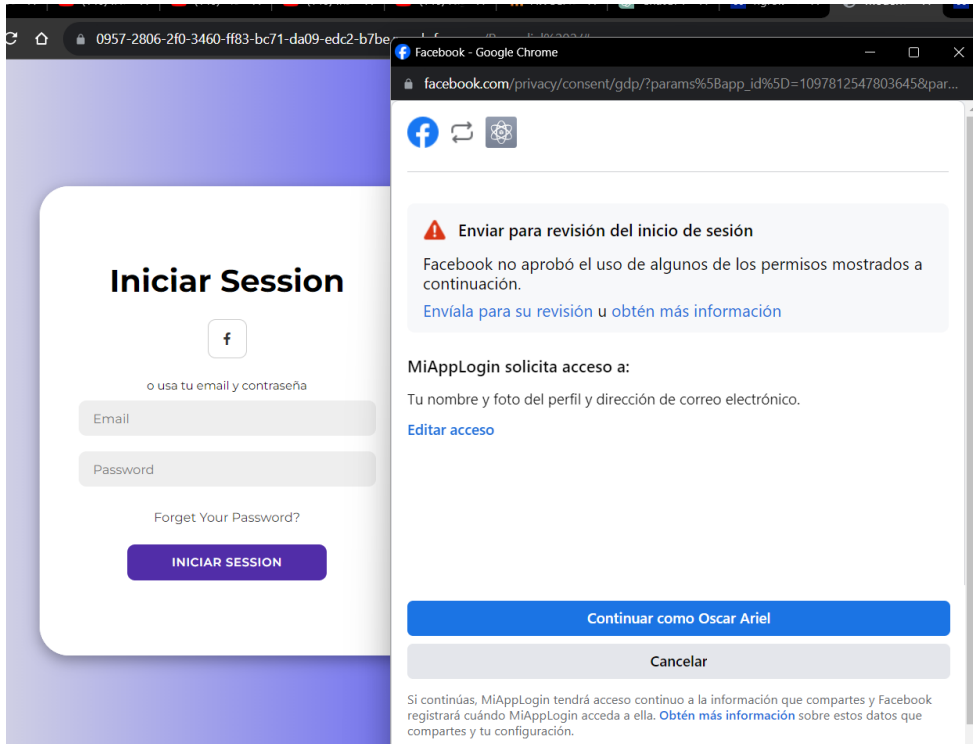
fbSignUp.addEventListener('click', function() {
  FB.login(function(response) {
    if (response.authResponse) {
      console.log('Registrando al usuario... ');
      FB.api('/me', function(response) {
        console.log('Registrado con éxito: ' + response.name);

        // TODO: se puede enviar los datos del usuario al servidor para crear una cuenta.
      });
    } else {
      console.log('Usuario Cancelo el registro o no se autorizo completamente. ');
    }
  }, {scope: 'public_profile,email'});
});
```

En esta sección de código se realiza la lógica del Login y SignUp donde utilizaremos la función FB.login proporcionada por la API de Facebook donde nos abrirá una ventana de inicio de sesión de Facebook.



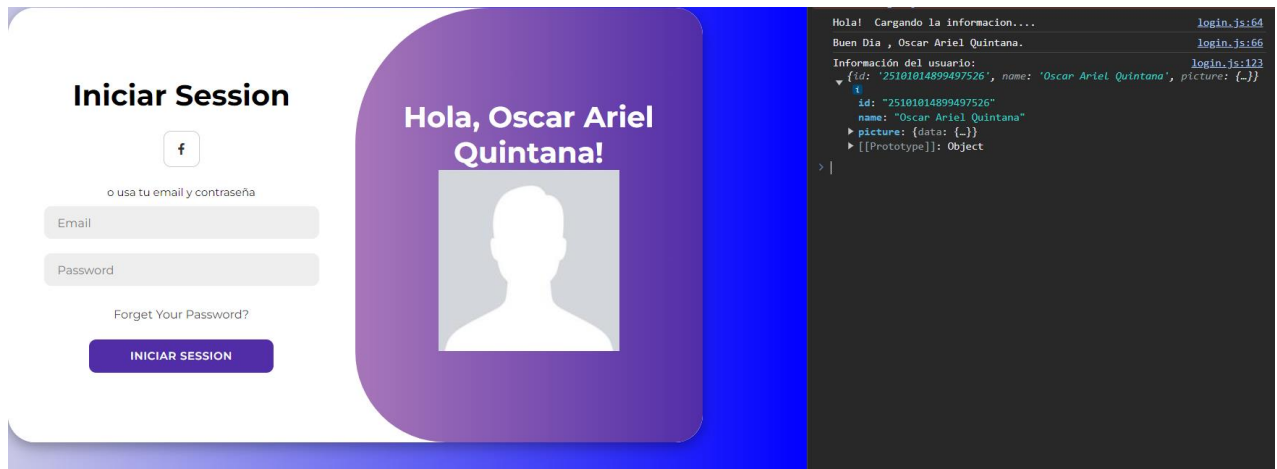
Debido a ser una aplicación de prueba básica sin subir a la web y hosteándolo en un servidor local puede aparecer la siguiente advertencia:



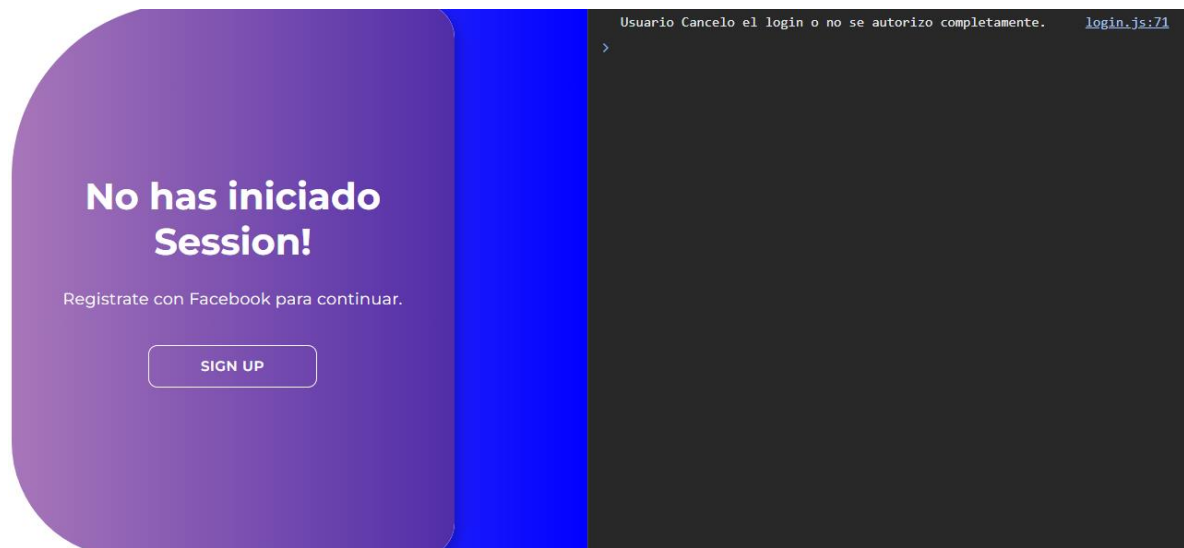
Para procesar la información de un usuario simplemente realizaremos un GET a la API con los datos del usuario y nos devolverá sus datos.

```
function getUserInfo() {  
  FB.api('/me', 'GET', { fields: 'id,name,picture' }, function(userInfo) {  
    // Procesar la información del usuario (nombre, foto, etc.)  
    console.log('Información del usuario:', userInfo);  
  });  
}
```

Una vez iniciado sesión correctamente se mostrará nuestro nombre así como nuestra foto de perfil de Facebook indicándonos que iniciamos sesión correctamente.



Si no se realiza la conexión correctamente el programa no cargará la sesión y mostrará por consola que no se pudo realizar la conexión correctamente.

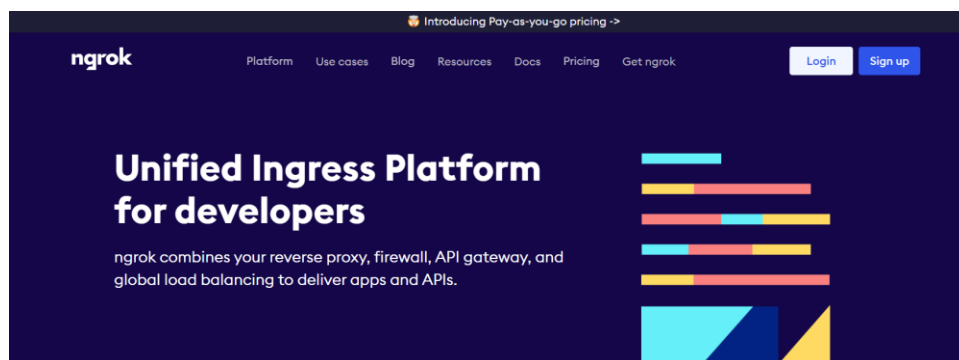


Crear credenciales de seguridad a un servidor local.

Por medidas de seguridad Facebook ya no permite realizar llamadas de su API mediante páginas con protocolo HTTP, sino que ahora es obligatorio que las páginas web tengan credenciales de seguridad mediante el protocolo HTTPS. Por ende, ya que para aprender y conocer la API de Facebook no queremos realizar la compra de un Host con protocolo HTTPS sino que se quiere realizar un servidor local con dichas credenciales de seguridad.

Para ello se utilizó la plataforma ngrok, la cual nos proporciona un link de nuestro localhost pero en un enlace con credenciales de seguridad HTTPS.

Para ello se descargará la aplicación de ngrok en su página web oficial.



Una vez descargada e instalada la aplicación se nos abrirá su terminal donde podremos crear servidores locales HTTPS asignándole nosotros el puerto al que queremos utilizar.

```
C:\Users\Ariel\Downloads\ngrok x + v
NAME:
  ngrok - tunnel local ports to public URLs and inspect traffic

USAGE:
  ngrok [command] [flags]

DESCRIPTION:
  ngrok exposes local networked services behinds NATs and firewalls to the
  public internet over a secure tunnel. Share local websites, build/test
  webhooks consumers and self-host personal services.
  Detailed help for each command is available with 'ngrok help <command>'.
  Open http://localhost:4040 for ngrok's web interface to inspect traffic.

Author:
  ngrok - <support@ngrok.com>

TERMS OF SERVICE: https://ngrok.com/tos

EXAMPLES:
  ngrok http 80 # secure public URL for port 80 web server
  ngrok http --domain baz.ngrok.dev 8080 # port 8080 available at baz.ngrok.dev
  ngrok http foo.dev:80 # tunnel to host:port instead of localhost
  ngrok http https://localhost # expose a local https server
  ngrok tcp 22 # tunnel arbitrary TCP traffic to port 22
  ngrok tls --domain=foo.com 443 # TLS traffic for foo.com to port 443
  ngrok start foo bar baz # start tunnels from the configuration file

COMMANDS:
  api use ngrok agent as an api client
  completion generates shell completion code for bash or zsh
```

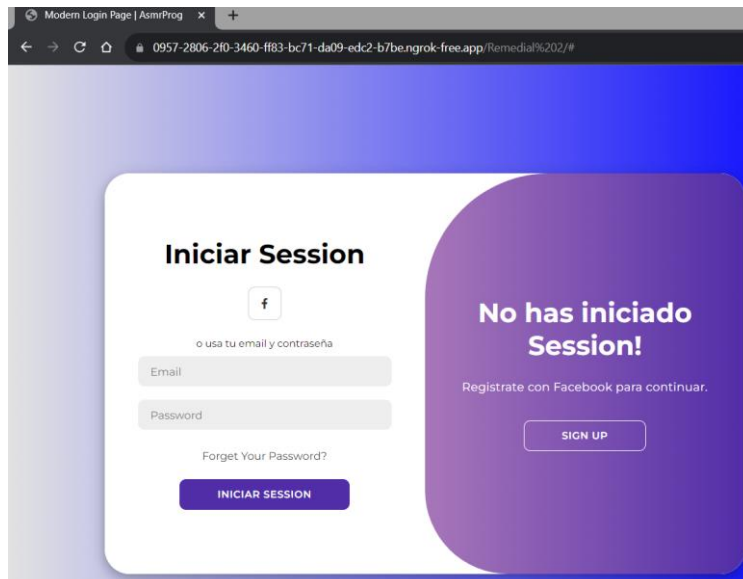
Por último introduciremos nuestras credenciales de nuestra cuenta de ngrok y crearemos un servidor HTTPS asignándole el puerto que queramos utilizar.

```
Introducing Pay-as-you-go pricing: https://ngrok.com/e/payg
Session Status online
Account Ariel Quintana (Plan: Free)
Version 3.4.0
Region United States (us)
Latency 47ms
Web Interface http://127.0.0.1:4040
Forwarding https://0957-2806-2f0-3460-ff83-bc71-da09-edc2-b7be.ngrok-free.app -> http://localhost:550

Connections
  ttl  opn  rt1  rt5  p50  p90
  171   0    0.00 0.00  5.14 89.31

HTTP Requests
-----
GET /Remedial 2/styles.css 304 Not Modified
GET /Remedial 2//ws 101 Switching Protocols
GET /Remedial 2/Login.js 304 Not Modified
GET /Remedial 2/ 304 Not Modified
GET /Remedial 2//ws 101 Switching Protocols
GET /Remedial 2/Login.js 304 Not Modified
GET /Remedial 2/ 304 Not Modified
GET /Remedial 2/styles.css 304 Not Modified
GET /Remedial 2//ws 101 Switching Protocols
GET /Remedial 2/Login.js 200 OK
```

Una vez realizado ingresaremos al link que se nos proporciona y obtendremos nuestra página web funcional.



FUENTES DE CONSULTA

- Facebook. (s. f.). Facebook for Developers. Recuperado de <https://developers.facebook.com/>
- Ngrok. (s. f.). Ngrok Dashboard. Recuperado de <https://dashboard.ngrok.com/>
- Patel, J. (2022, [Fecha de publicación]). Login with Facebook | Integrate Login with Facebook | Javascript. YouTube. <https://www.youtube.com/watch?v=NGFoNyeno7s>
- Code Java. (2023, [Fecha de publicación]). Spring Boot OAuth2 Social Login with Google and Facebook in One Application. YouTube. <https://www.youtube.com/watch?v=MLVp4P2Cty8&t=307s>