



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TLAXIACO

SEGURIDAD Y VIRTUALIZACIÓN

Nombre de los Integrantes de equipo:

No. Control

Edwin López Santiago

21620123

Yanet González García

21620273

Tema:

Practica 2 : Autorización y Autenticación

Docente:

Ing. Osorio Salinas Edward

Carrera:

Ingeniería en Sistemas Computacionales

Grupo: 7US

Semestre: Séptimo.

Tlaxiaco, Oaxaca. A 05 de septiembre de 2024.



Índice

Introducción.....	4
Desarrollo.....	5
Práctica 2: Autorización y Autenticación.....	5
Instrucción:	5
1.- Crea una aplicación [web mobile escritorio] que permita loguearse con un usuario y contraseña.	6
PÁGINA WEB	6
<input type="checkbox"/> La aplicación debe tener un formulario de “login” con los campos de usuario y contraseña.	6
<input type="checkbox"/> La aplicación debe tener un formulario de registro con los campos de usuario, contraseña y confirmación de contraseña.	7
<input type="checkbox"/> La aplicación debe decirme si la contraseña es segura o no (extra).....	8
<input type="checkbox"/> La aplicación debe tener una página de inicio que sea accesible para cualquier usuario.	11
<input type="checkbox"/> La aplicación debe tener una página de perfil que solo sea accesible si el usuario ha iniciado sesión.	11
<input type="checkbox"/> La aplicación debe tener una página de administración que solo sea accesible si el usuario ha iniciado sesión y tiene un rol de administrador.	12
2.- Implementa un mecanismo de autorización que permita o deniegue el acceso a ciertas rutas de la aplicación, en este caso la página de perfil y la página de administración si en dado caso el usuario no ha iniciado sesión o no tiene el rol de administrador.....	13
MECANISMO DE AUTENTICACIÓN (REGISTRO, LOGIN Y CIERRE DE SESIÓN)	13
3.- Implementa un mecanismo de autenticación que permita a los usuarios registrarse, iniciar sesión y cerrar sesión.	16
MECANISMO DE AUTORIZACIÓN.....	16
(ACCESO A CIERTAS RUTAS SEGÚN ROLES).....	16
4.- Implementa un mecanismo para cerrar sesión de un usuario si ha pasado un tiempo determinado sin actividad de 5 mins.....	17
MECANISMO DE CIERRE DE SESIÓN POR INACTIVIDAD	17
5.- Investiga y describe los siguientes servicios de autenticación:	22
o LDAP	22
o RADIUS	22
o TACACS+	22



- o Kerberos..... 22
- 6.- Investiga y describe los siguientes servicios de autorización: 23
 - o ACL..... 23
 - o RBAC 23
 - o ABAC 23
 - o PBAC 23
- Conclusión 25
- Bibliografía..... 26



Introducción

En esta práctica, el objetivo es aprender sobre Autenticación y Autorización, dos conceptos clave en la seguridad de la información. La autenticación nos permite verificar la identidad de un usuario, y la autorización define qué acciones puede realizar ese usuario según su rol. En este apartado, se muestra como se ha desarrollado una página web donde los usuarios pueden registrarse, iniciar sesión y acceder a diferentes secciones según su nivel de permisos.

También se investigó servicios de autenticación y autorización para comprender mejor cómo funcionan y cómo se aplican a la seguridad de los sistemas. La seguridad de la información es un aspecto fundamental para proteger los datos sensibles y garantizar el acceso controlado a los sistemas. A continuación, se visualizará cada paso de la práctica solicitada, de igual forma, investigaciones del tema.



Desarrollo

Práctica 2: Autorización y Autenticación

Instrucción:

1.- Crea una aplicación [web|mobile|escritorio] que permita loguearse con un usuario y contraseña.

- La aplicación debe tener un formulario de inicio de sesión con los campos de usuario y contraseña.
- La aplicación debe tener un formulario de registro con los campos de usuario, contraseña y confirmación de contraseña.
- La aplicación debe decirme si la contraseña es segura o no (extra).
- La aplicación debe tener una página de inicio que sea accesible para cualquier usuario.
- La aplicación debe tener una página de perfil que solo sea accesible si el usuario ha iniciado sesión.
- La aplicación debe tener una página de administración que solo sea accesible si el usuario ha iniciado sesión y tiene un rol de administrador.

2.- Implementa un mecanismo de autorización que permite o deniegue el acceso a ciertas rutas de la aplicación, en este caso la página de perfil y la página de administración si en dado caso el usuario no ha iniciado sesión o no tiene el rol de administrador.

3.- Implemente un mecanismo de autenticación que permita a los usuarios registrarse, iniciar sesión y cerrar sesión.

4.- Implementa un mecanismo para cerrar sesión de un usuario si ha pasado un tiempo determinado sin actividad de 5 mins.

A continuación, se presenta desde el paso 1, 2, 3 y 4.



1.- Crea una aplicación [web|mobile|escritorio] que permita loguearse con un usuario y contraseña.

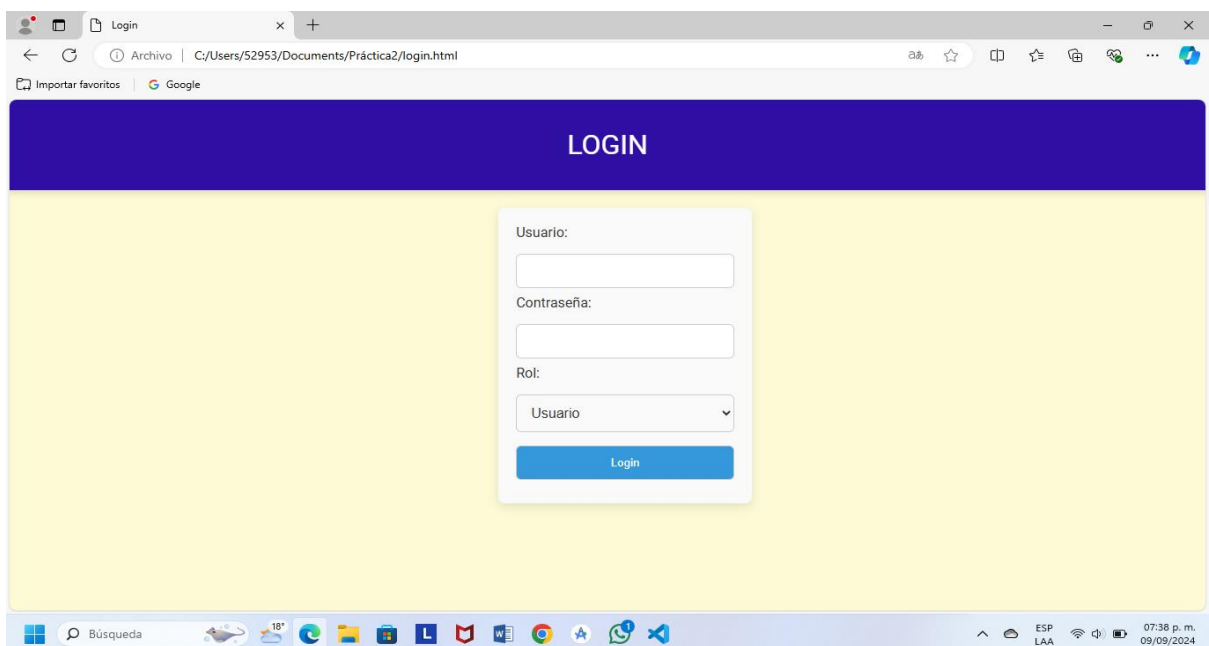
PÁGINA WEB

- La aplicación debe tener un formulario de “login” con los campos de usuario y contraseña.

1.- La página de inicio de la aplicación es la primera que los usuarios verán al acceder, y está diseñada para darles la bienvenida con un mensaje como "Bienvenido a nuestra página web."



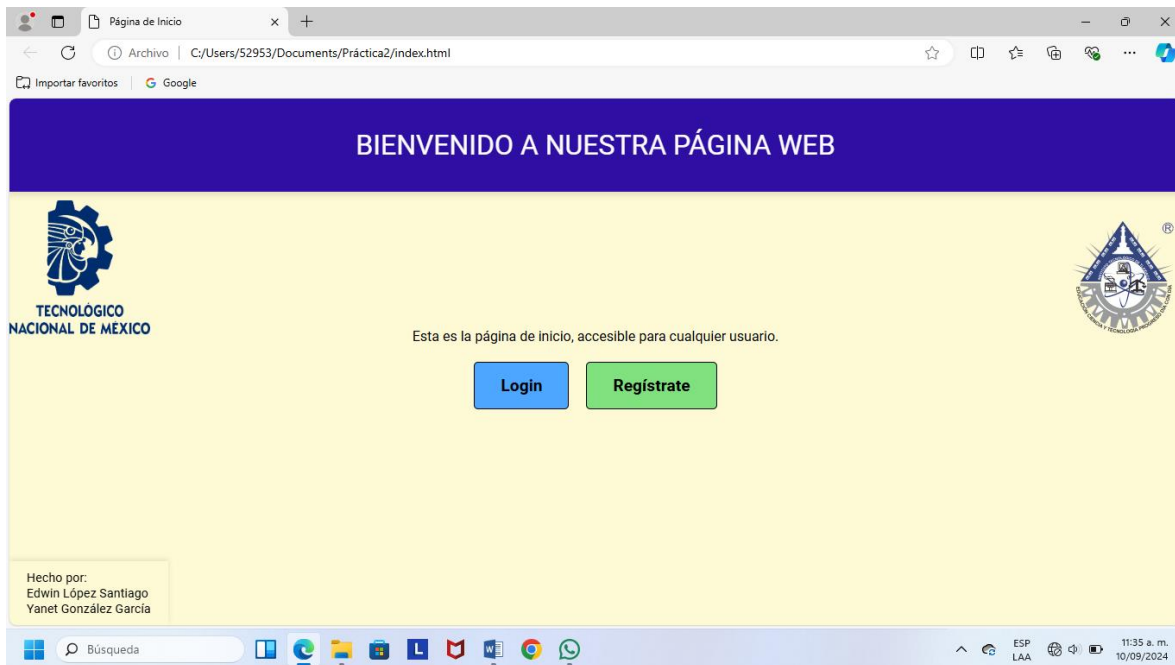
2.- Este es el apartado de Login con los campos de usuario y contraseña, en nuestro caso agregamos un campo más para el tipo de rol que tenemos.



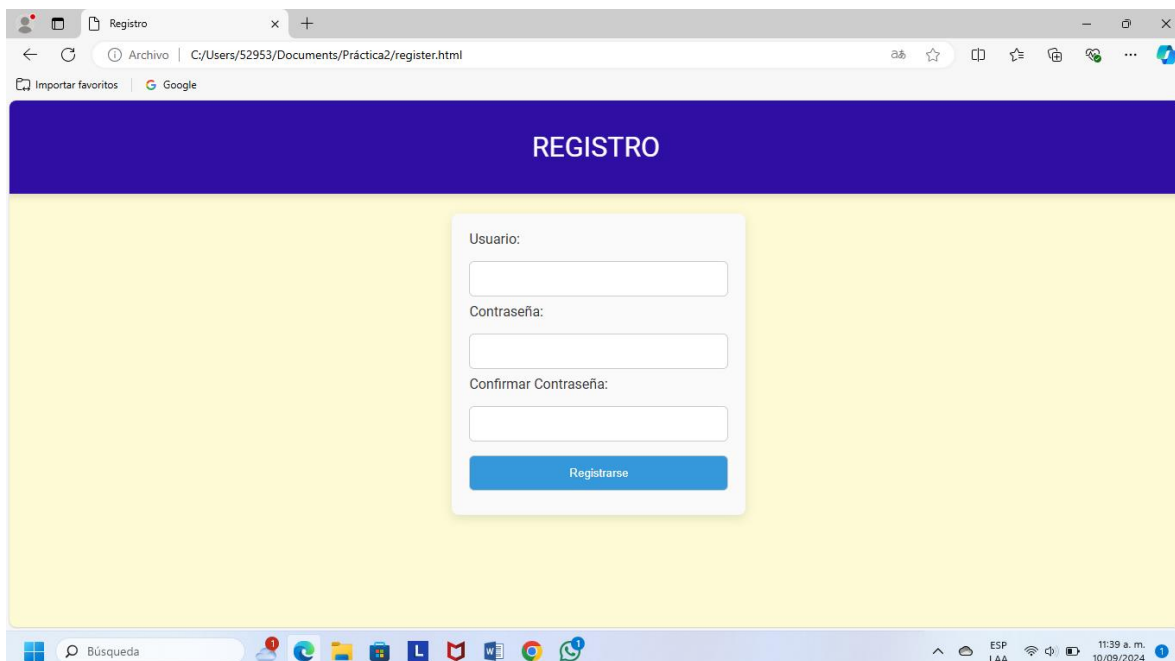


- La aplicación debe tener un formulario de registro con los campos de usuario, contraseña y confirmación de contraseña.

3.- Para este punto tenemos que estar en la página principal y seleccionar la opción de Regístrate.



4.- Y en este apartado veremos el registro con los campos de usuario, contraseña y confirmación de contraseña.





5.- Y ya cuando registramos un usuario, la contraseña y confirmación de la contraseña nos marca que “El registro es exitoso”.

Registro

Archivo | C:/Users/52953/Documents/Práctica2/register.html

Importar favoritos | Google

REGISTRO

Usuario:

Edwin

Contraseña:

Confirmar Contraseña:

Registrarse

Registro exitoso.

Búsqueda

ESP LAA

11:42 a. m.
10/09/2024

- **La aplicación debe decirme si la contraseña es segura o no (extra).**

6.- Esta es la página de inicio, accesible para cualquier usuario. En este apartado es donde indica si la aplicación debe decirme si la contraseña es segura o no.

Página de Inicio

Archivo | C:/Users/52953/Documents/Práctica2/index.html

Importar favoritos | Google

BIENVENIDO A NUESTRA PÁGINA WEB

TECNOLÓGICO NACIONAL DE MÉXICO

Esta es la página de inicio, accesible para cualquier usuario.

Login Registrarse

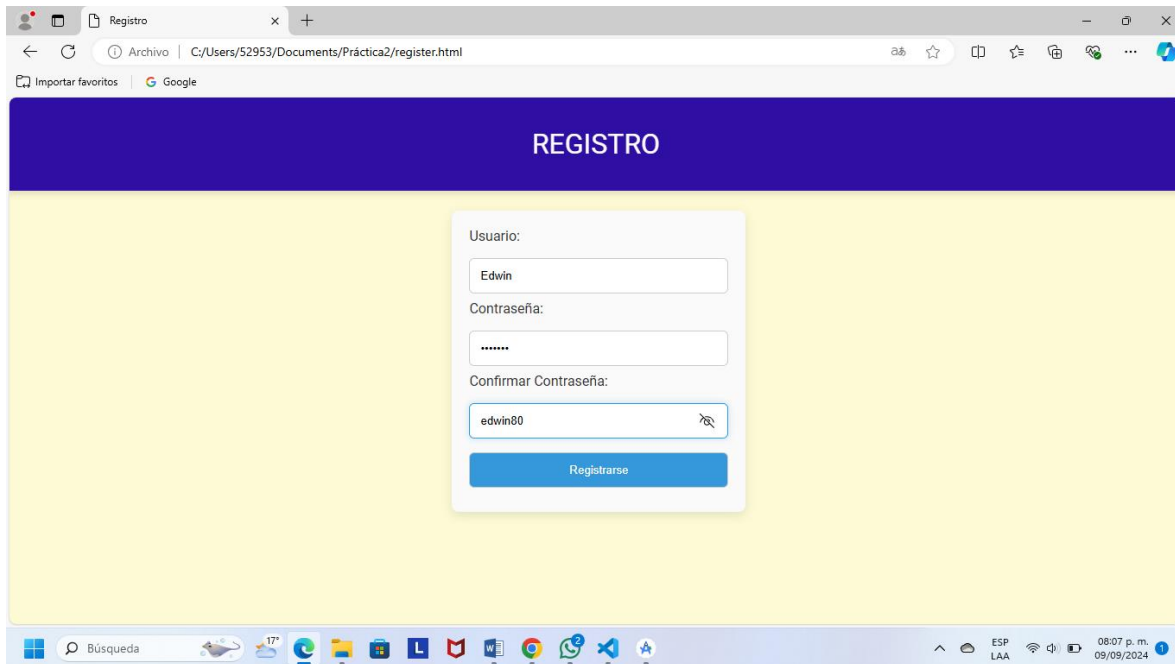
Hecho por:
Edwin López Santiago
Yanet González García

Búsqueda

ESP LAA

07:39 p. m.
09/09/2024

7.- Al darle clic al botón de registro nos arroja una ventana como aparece a continuación. En esta ventana vamos a rellenar lo que nos piden. En este caso se rellenó de la siguiente manera.



Registro

Archivo | C:/Users/52953/Documents/Práctica2/register.html

Importar favoritos | Google

REGISTRO

Usuario:

Edwin

Contraseña:

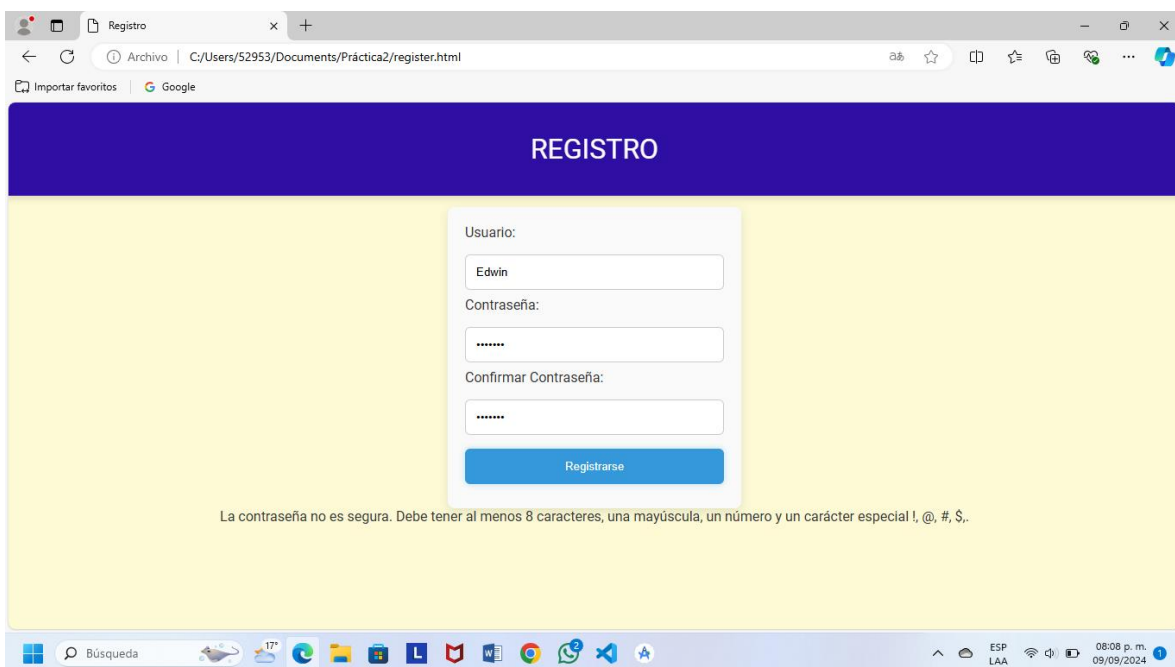
Confirmar Contraseña:

edwin80

Registrarse

Búsqueda 17° ESP LAA 08:07 p. m. 09/09/2024

8.- Al dar clic al botón de registrarse aparece el siguiente mensaje “La contraseña no es segura. ¡Debe tener al menos 8 caracteres, una mayúscula, un número y un carácter especial !, @, #, \$,.” Esto se debe que tenemos validado la contraseña para que tenga una contraseña segura y con las características que debe tener.



Registro

Archivo | C:/Users/52953/Documents/Práctica2/register.html

Importar favoritos | Google

REGISTRO

Usuario:

Edwin

Contraseña:

Confirmar Contraseña:

Registrarse

La contraseña no es segura. Debe tener al menos 8 caracteres, una mayúscula, un número y un carácter especial !, @, #, \$,.

Búsqueda 17° ESP LAA 08:08 p. m. 09/09/2024



9.- Al volver nuevamente a escribir **“Edwin.3917@”** y darle en registrarse nos aparecerá la siguiente ventana.

Registro

Archivo | C:/Users/52953/Documents/Práctica2/register.html

Importar favoritos | Google

REGISTRO

Usuario:

Edwin

Contraseña:

Confirmar Contraseña:

Edwin.3917@

Registrarse

Búsqueda 17° ESP LAA 08:08 p. m. 09/09/2024

10.- Al dar clic en **“registrarse”** nos aparece el siguiente mensaje, la cual ya cumple con los requisitos indicados para poder hacer un registro de un nuevo usuario.

Registro

Archivo | C:/Users/52953/Documents/Práctica2/register.html

Importar favoritos | Google

REGISTRO

Usuario:

Edwin

Contraseña:

Confirmar Contraseña:

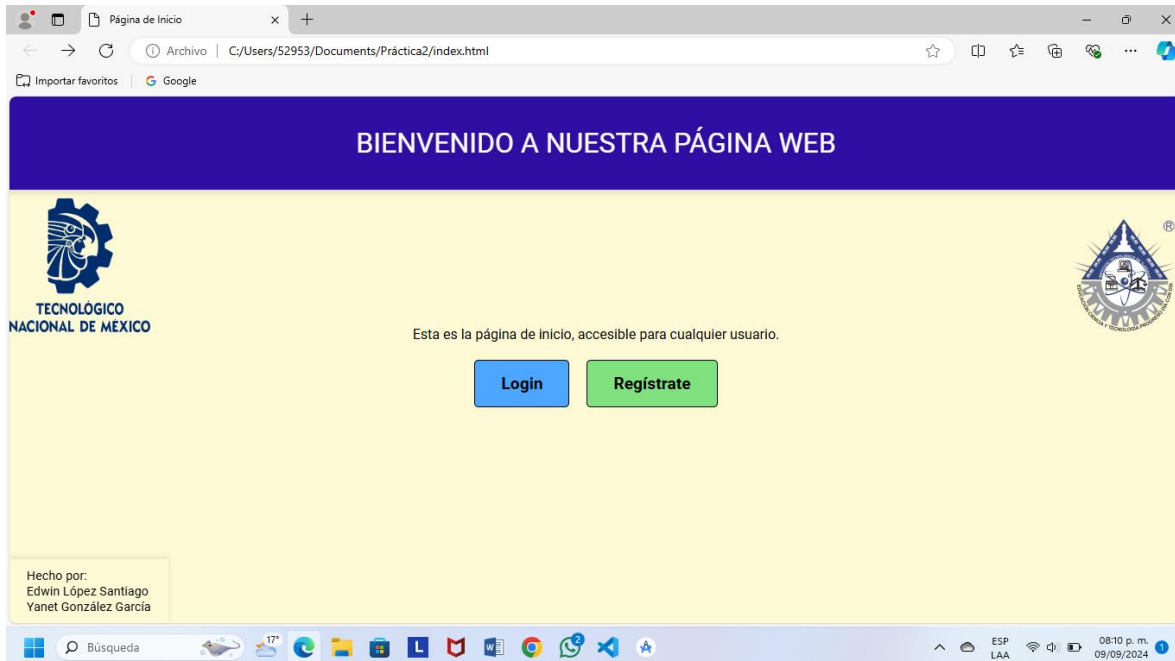
Registrarse

Registro exitoso.

Búsqueda 17° ESP LAA 08:09 p. m. 09/09/2024

- La aplicación debe tener una página de inicio que sea accesible para cualquier usuario.

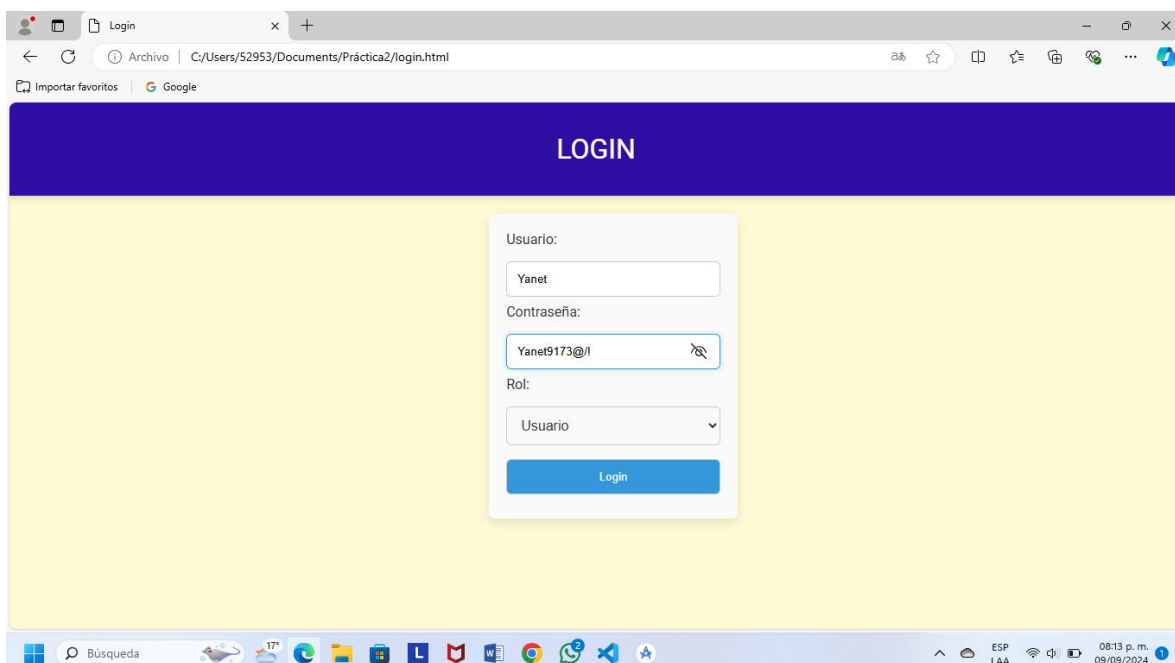
11.- Nuestra página principal es esta donde se encuentran las opciones de login y registrarse.



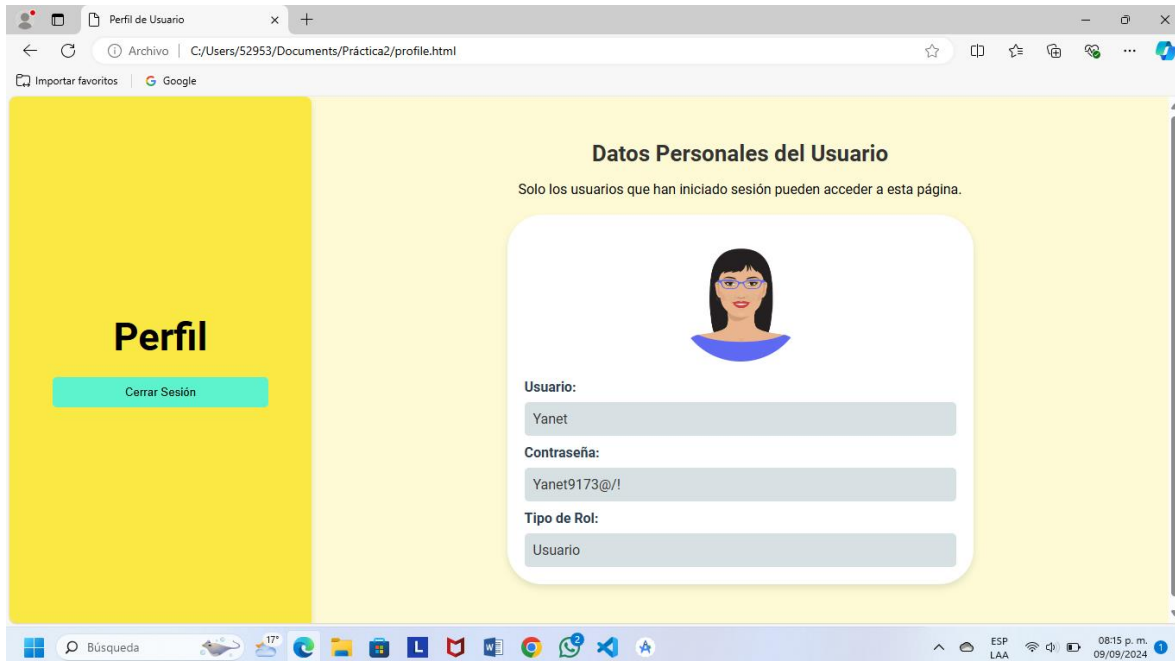
- La aplicación debe tener una página de perfil que solo sea accesible si el usuario ha iniciado sesión.

12.- Para realizar dicho punto tenemos que iniciar sesión desde el login y rellenar los campos que nos piden.

Nota: Como es para iniciar sesión debemos de seleccionar el rol de usuario.



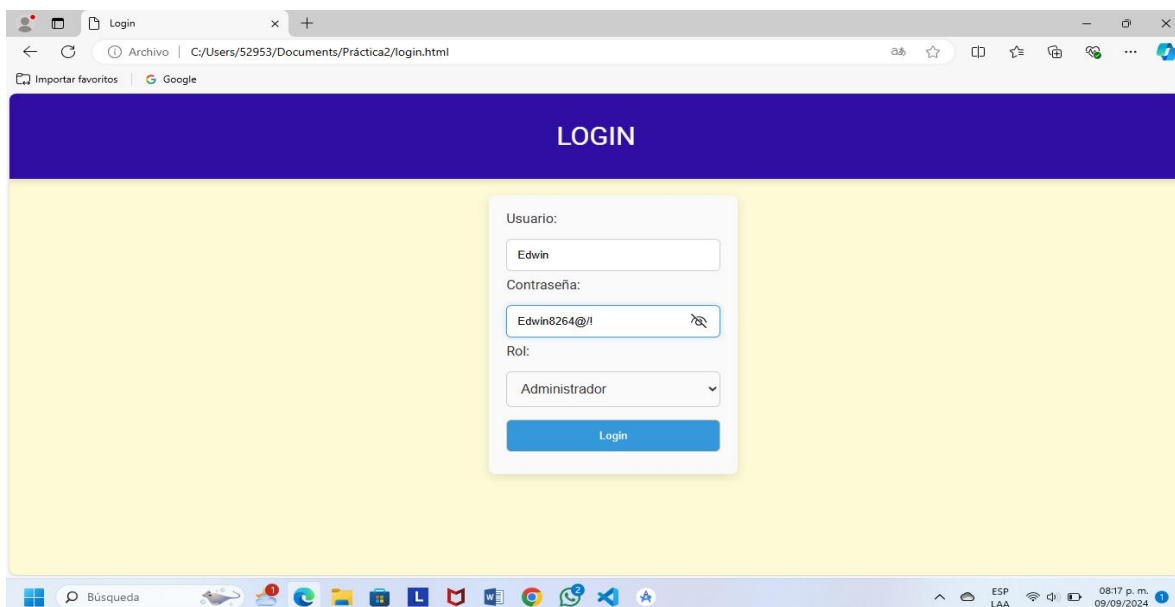
13.- Una vez que hemos accedido esta es la página para nuestro usuario. Esta página está diseñada para que solo los usuarios autenticados puedan acceder a ella, proporcionando un espacio seguro y personalizado para cada usuario registrado.



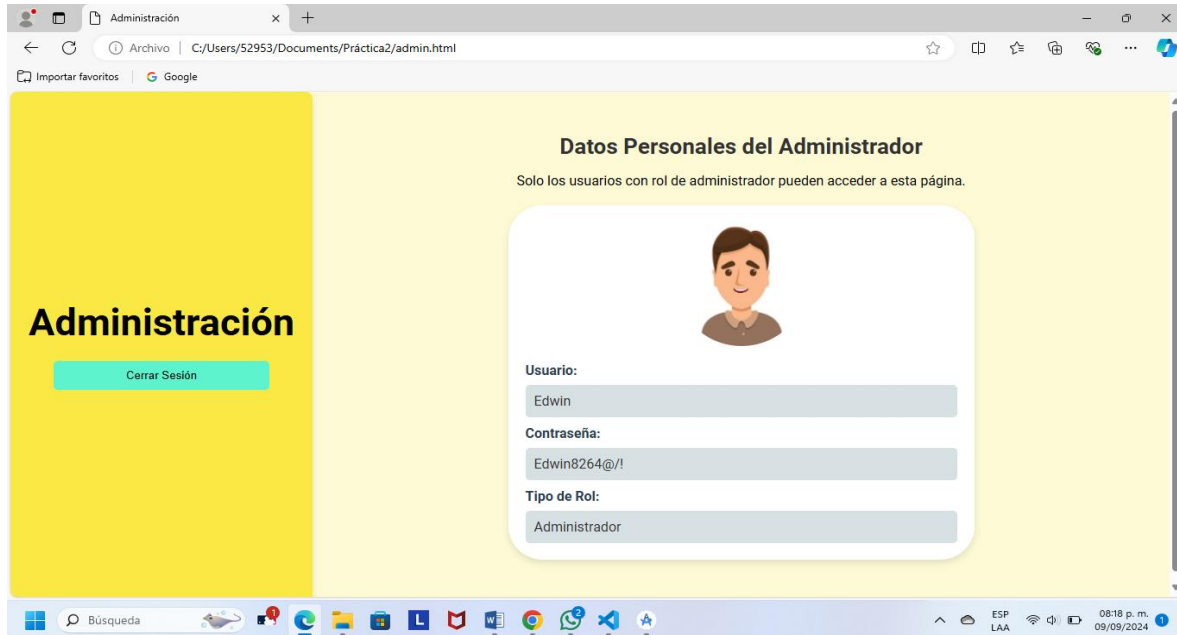
- La aplicación debe tener una página de administración que solo sea accesible si el usuario ha iniciado sesión y tiene un rol de administrador.

14.- Para realizar dicho punto tenemos que iniciar sesión desde el login y rellenar los campos que nos piden.

Nota: como es para iniciar sesión debemos de seleccionar el rol de administrador y utilizaremos otro tipo de usuario con el rol de administrador.



15.- Una vez que hemos accedido este es el diseño de nuestra página para nuestro administrador. En esta página, el administrador puede ver información relevante relacionada con su cuenta, como su nombre de usuario, detalles de perfil.



2.- Implementa un mecanismo de autorización que permita o deniegue el acceso a ciertas rutas de la aplicación, en este caso la página de perfil y la página de administración si en dado caso el usuario no ha iniciado sesión o no tiene el rol de administrador.

MECANISMO DE AUTENTICACIÓN (REGISTRO, LOGIN Y CIERRE DE SESIÓN)

16.- **Registro:** La parte de registro está en el bloque de código que escucha el evento del formulario registerForm.

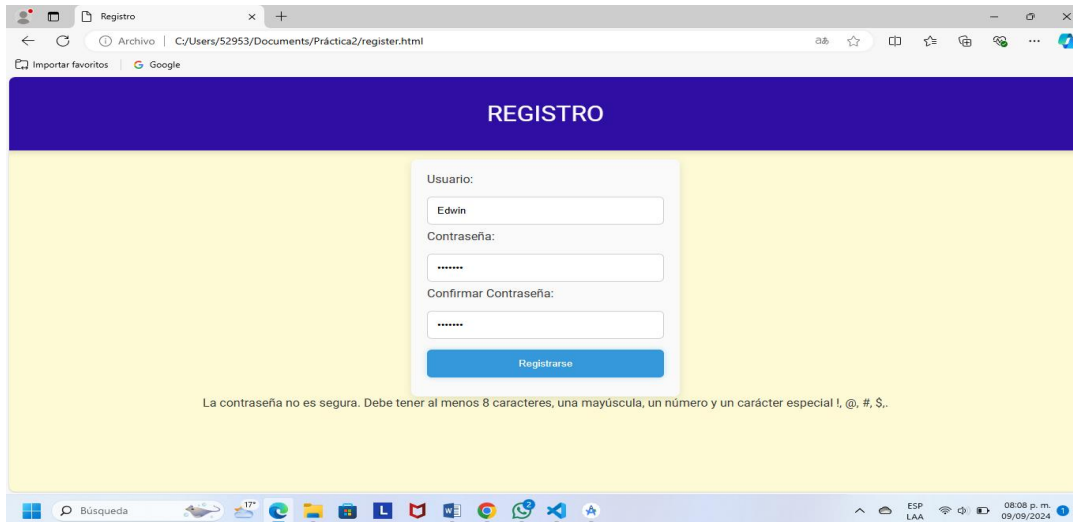
```
document.getElementById('registerForm')?.addEventListener('submit', function (e) {
    e.preventDefault();
    const username = document.getElementById('newUsername').value;
    const password = document.getElementById('newPassword').value;
    const confirmPassword = document.getElementById('confirmPassword').value;
    const msgDiv = document.getElementById('passwordMsg');

    const passwordSecurityMessage = isPasswordSecure(password);

    if (passwordSecurityMessage !== "La contraseña es segura.") {
        msgDiv.textContent = passwordSecurityMessage;
    } else if (password !== confirmPassword) {
        msgDiv.textContent = "Las contraseñas no coinciden.";
    }
});
```

```
} else {  
    msgDiv.textContent = "Registro exitoso";  
    // Guardar el nuevo usuario (simulación)  
    users.push({ username, password, role: "user" });  
}  
});
```

17.- En este apartado se ingresó una contraseña cualquiera, no cumple con todos los requisitos, no cumple con ningún carácter. Este es un breve ejemplo, la cual se debe ingresar una contraseña que cumpla con los requisitos solicitados y sea segura.



18.- **Login:** El proceso de inicio de sesión se maneja cuando el usuario envía el formulario loginForm. Se valida si el usuario existe en el arreglo users, y si las credenciales coinciden, se guarda la sesión en el localStorage mediante la función setSession.

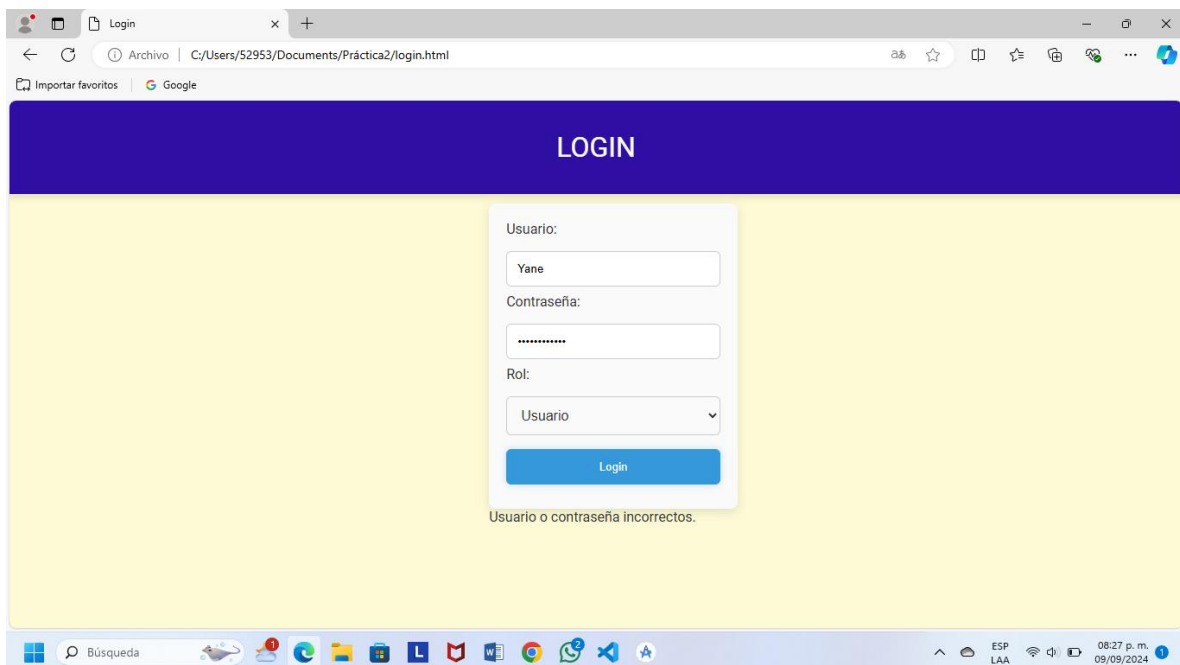
```
document.getElementById('loginForm')?.addEventListener('submit', function  
(e) {  
    e.preventDefault();  
    const username = document.getElementById('username').value;  
    const password = document.getElementById('password').value;  
    const errorMsg = document.getElementById('errorMsg');  
  
    const user = users.find(u => u.username === username && u.password ===  
password);  
  
    if (user) {  
        errorMsg.textContent = "";  
        setSession(user); // Establecer la sesión
```

```
if (user.role === "admin") {  
    window.location.href = "admin.html";  
} else {  
    window.location.href = "profile.html";  
}  
} else {  
    errorMsg.textContent = "Usuario o contraseña incorrectos.";  
}  
});
```

19.- Cuando se validan las credenciales del login podemos acceder a la página del administrador o usuario y nos dirige a la página de usuario o administrador.



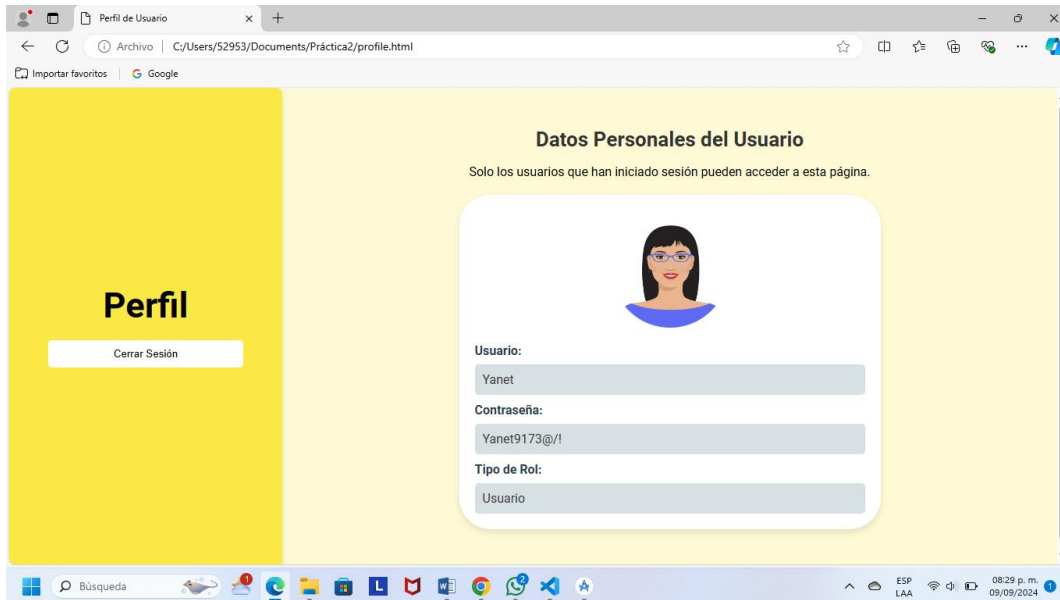
20.- De igual forma pasa si no anotamos bien al usuario nos mandara al siguiente mensaje.



21.- **Cierre de Sesión:** Para cerrar sesión manualmente, se elimina la sesión del localStorage y se redirige al usuario a la página de inicio de sesión.

```
document.getElementById('logoutButton')?.addEventListener('click', function  
( ) {  
    clearSession();  
});
```

22.- Para el cierre de sesión tenemos implementado un botón llamado “Cerrar sesión”.



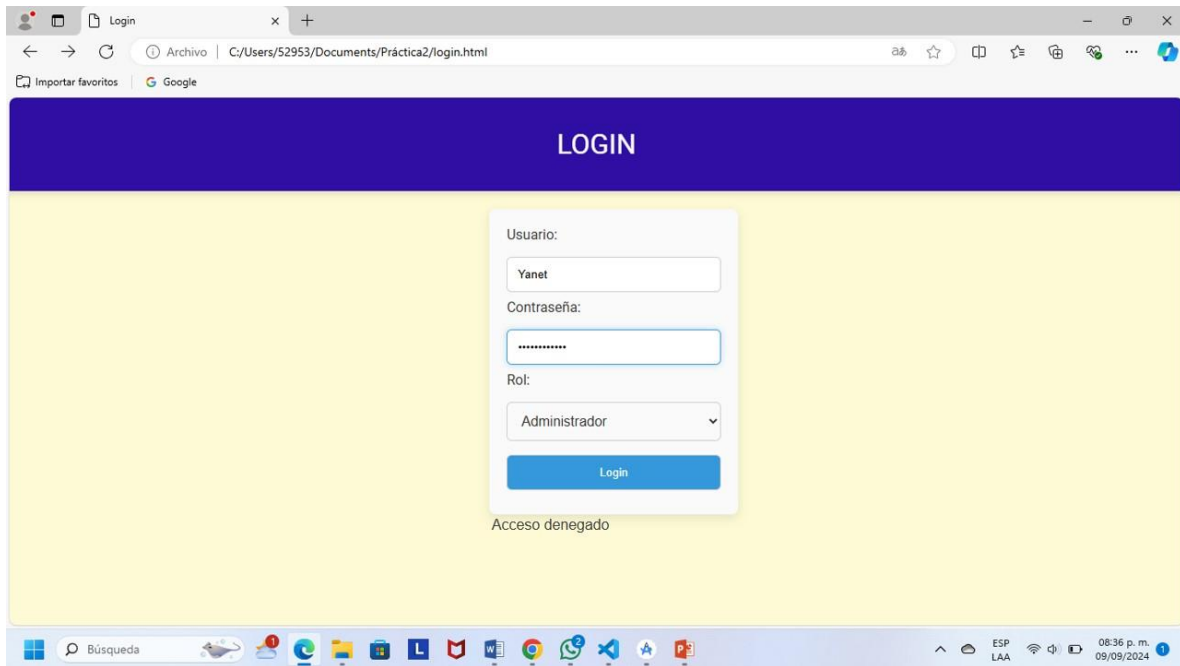
3.- Implementa un mecanismo de autenticación que permita a los usuarios registrarse, iniciar sesión y cerrar sesión.

MECANISMO DE AUTORIZACIÓN (ACCESO A CIERTAS RUTAS SEGÚN ROLES)

23.- **Autorización para Rutas:** La función checkAuth se utiliza para verificar si el usuario tiene acceso a ciertas páginas. Se verifica si el usuario ha iniciado sesión y si tiene el rol adecuado para acceder a páginas específicas como el perfil o la administración. Si el usuario no está autenticado o no tiene el rol adecuado, se redirige a la página de inicio de sesión.

```
// Redirigir si no está autenticado o no tiene el rol adecuado  
function checkAuth(role = null) {  
    const user = getSession();  
    if (!user || (role && user.role !== role)) {  
        alert("Acceso denegado");  
        window.location.href = 'login.html';  
    }  
}
```


24.- Si el usuario quiere iniciar un acceso con el rol del administrador se denegará el acceso ya que solamente los verdaderos administradores pueden acceder a ella.



4.- Implementa un mecanismo para cerrar sesión de un usuario si ha pasado un tiempo determinado sin actividad de 5 mins.

MECANISMO DE CIERRE DE SESIÓN POR INACTIVIDAD

25.- **Temporizador de Inactividad:** Se resetea el temporizador de inactividad cada vez que el usuario interactúa con la página (moviendo el mouse o presionando teclas). Si pasan 5 minutos sin actividad, se cierra la sesión automáticamente.

```
// Tiempo máximo de inactividad (5 minutos en milisegundos)
const MAX_INACTIVITY_TIME = 5 * 60 * 1000;

// Variables para controlar la sesión y el tiempo de inactividad
let inactivityTimer;

// Guardar sesión en localStorage
function setSession(user) {
  localStorage.setItem('currentUser', JSON.stringify(user));
  resetInactivityTimer(); // Resetea el contador de inactividad
}

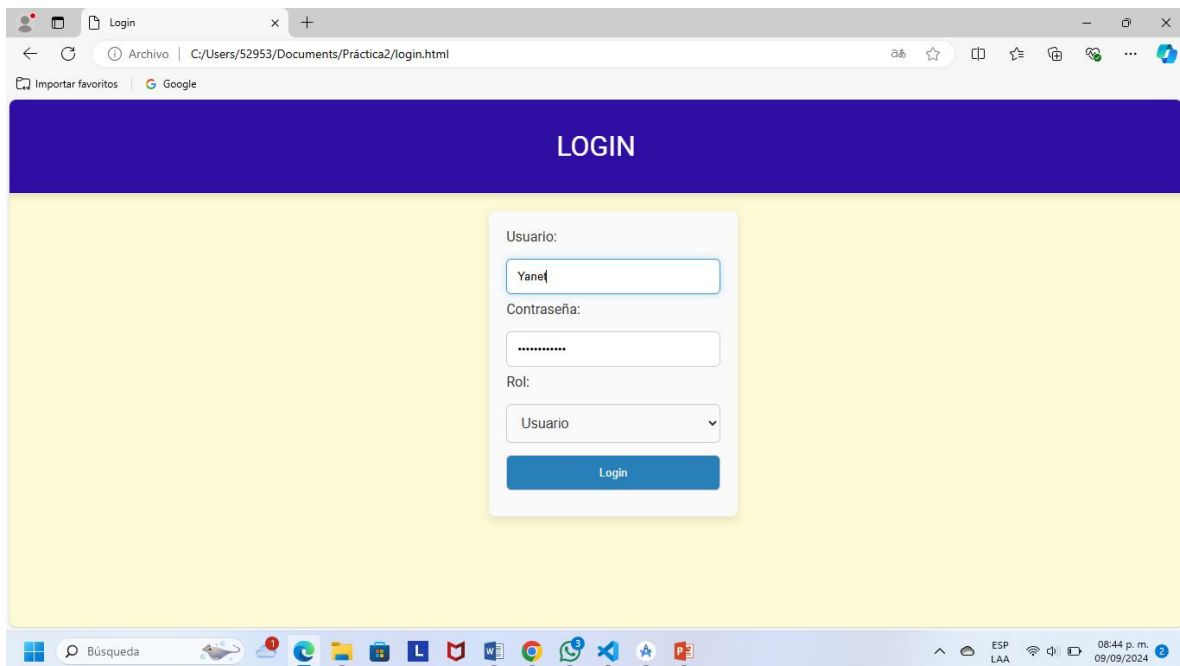
// Eliminar sesión (Cerrar sesión)
function clearSession() {
```

```
localStorage.removeItem('currentUser');
window.location.href = 'index.html';
}

// Resetea el temporizador de inactividad
function resetInactivityTimer() {
  if (inactivityTimer) {
    clearTimeout(inactivityTimer);
  }
  inactivityTimer = setTimeout(() => {
    alert("Sesión expirada por inactividad.");
    clearSession();
  }, MAX_INACTIVITY_TIME);
}

// Control de inactividad: resetea el temporizador con cualquier interacción
window.onload = resetInactivityTimer;
window.onmousemove = resetInactivityTimer;
window.onkeypress = resetInactivityTimer;
```

26.- Para hacer este punto lo haremos iniciando sesión con el login para acceder a la página de cualquier rol y si no hacemos un movimiento con el mouse o el teclado se expira nuestra sesión esto está habilitado para las sesiones de usuario y administrador.



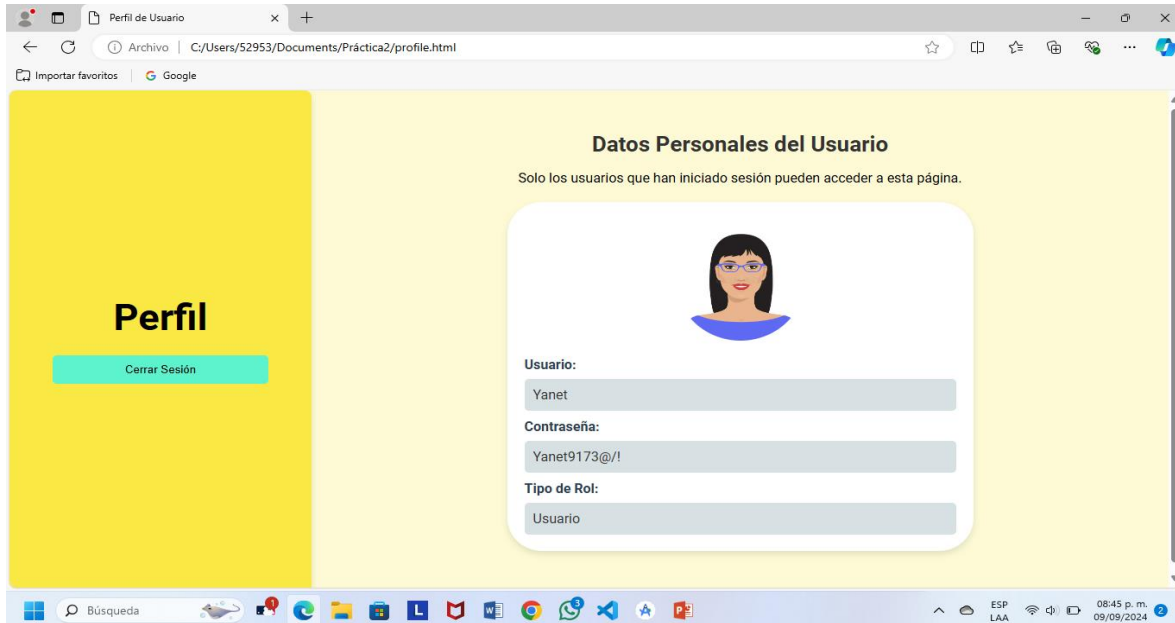
La imagen muestra una ventana de navegador web con la URL `C:/Users/52953/Documents/Práctica2/login.html`. El título de la pestaña es "Login". La interfaz de usuario tiene un encabezado azul oscuro con el texto "LOGIN" en blanco. El fondo principal es amarillo. En el centro, hay un formulario de login con los siguientes campos:

- Usuario:** Un campo de texto con el valor "Yanel".
- Contraseña:** Un campo de texto con caracteres ocultos por asteriscos.
- Rol:** Un menú desplegable con la opción "Usuario" seleccionada.
- Botón:** Un botón azul con el texto "Login".

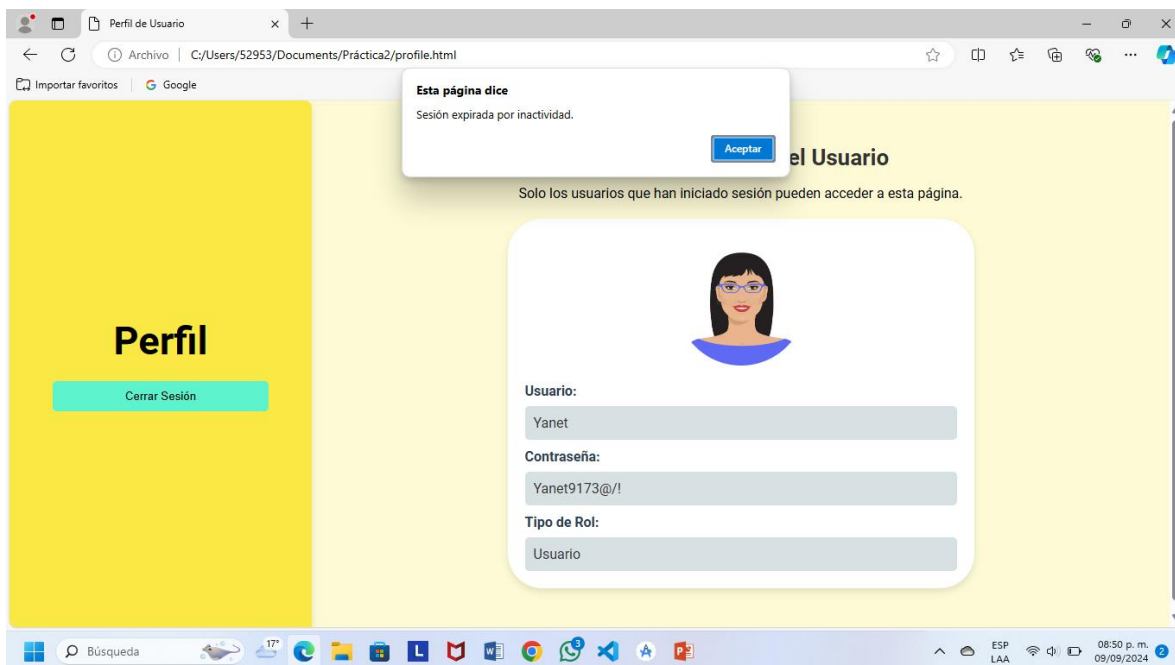
La barra de tareas de Windows está visible en la parte inferior, mostrando la hora como 08:44 p. m. el 09/09/2024.

27.- Ya estamos en la sesión del **usuario** y no haremos ningún movimiento por 5 minutos y vamos a esperar a que expire nuestra sesión.

Nota: En mi reloj son a las 8:45 por default nos tiene que expirar la sesión a las 8:50 minutos.

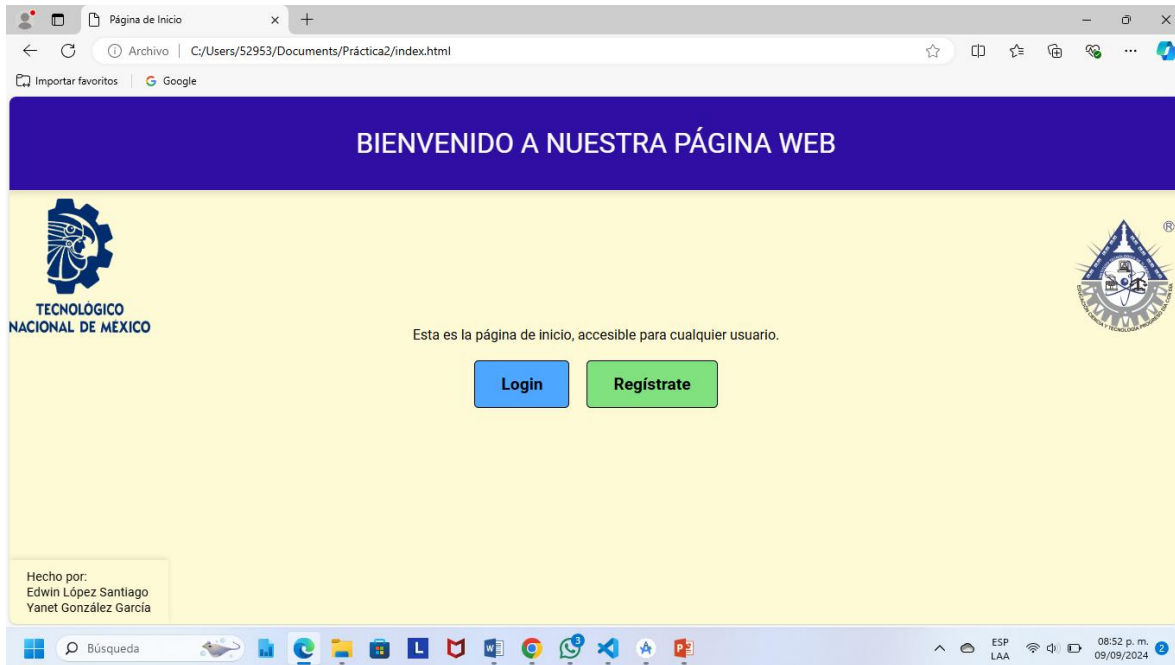


28.- Como lo mencione siendo a las 8:50 nuestra sesión ha sido expirada y nos manda la ventana emergente con el siguiente mensaje de "Esta página dice: Sesión expirada por inactividad" y nos da la opción de aceptar la cual tenemos que aceptar por defecto.

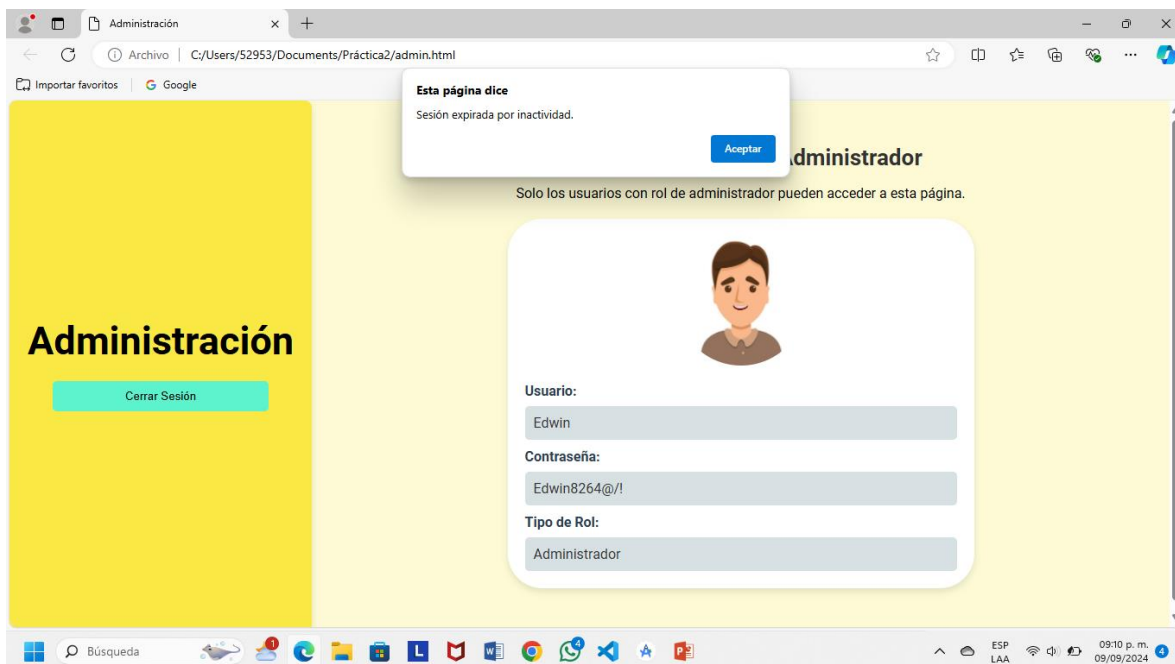




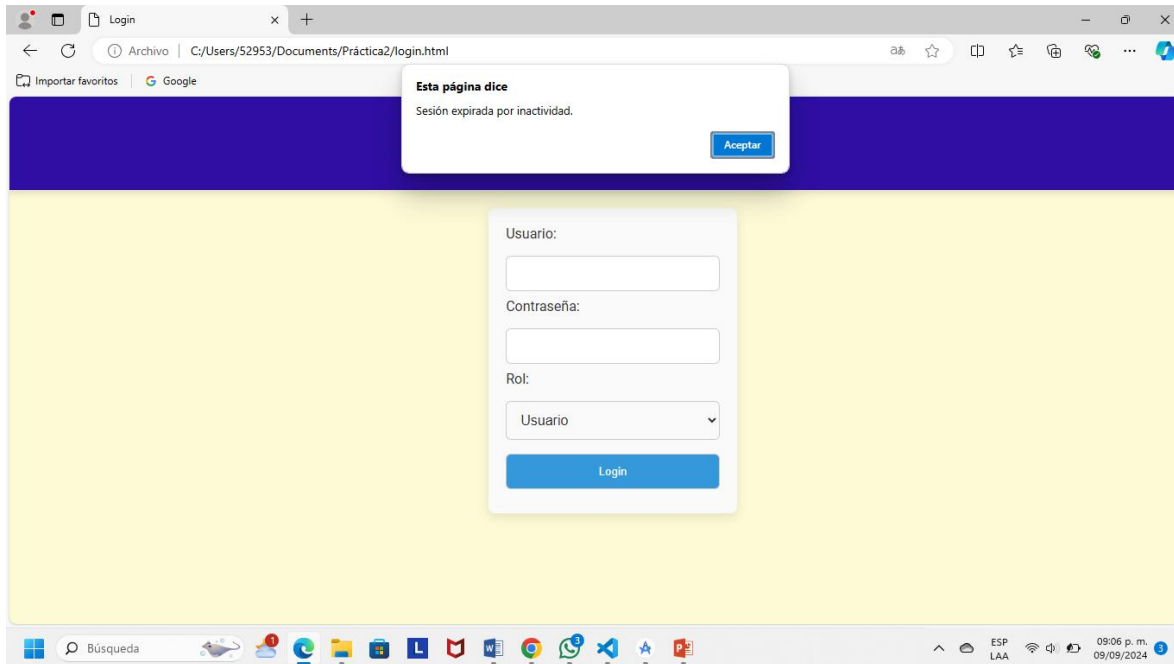
29.- Al presionar el botón de **Aceptar** nos regresa a la página principal donde cualquier usuario puede acceder.



30.- El mismo mecanismo tiene el **rol del administrador** si no se hace ningún movimiento se expira la página en 5 minutos.



31.- El mismo mecanismo tiene en el **Login** si no se hace ningún movimiento se expira la página en 5 minutos.



Esta página dice
Sesión expirada por inactividad.

Usuario:

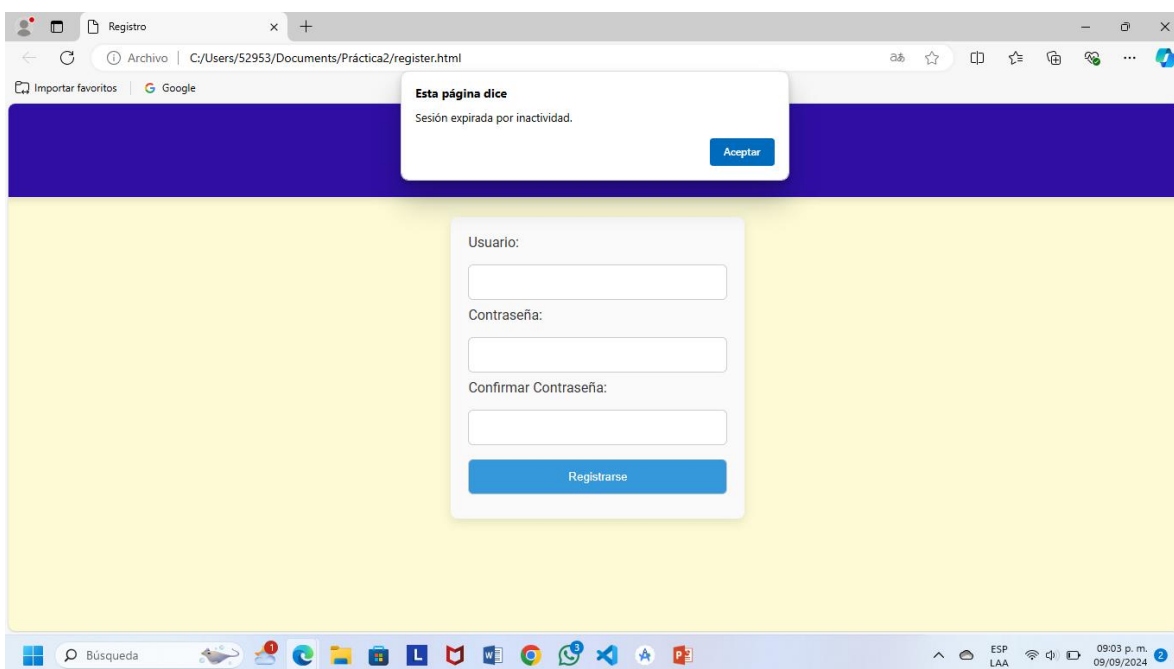
Contraseña:

Rol:

Login

32.- El mismo mecanismo tiene en la **“página de Registro”** si no se hace ningún movimiento se expira la página en 5 minutos.

En todas las páginas dice el mismo mensaje y la misma opción, dicha opción los manda hasta el inicio donde cualquier usuario puede acceder.



Esta página dice
Sesión expirada por inactividad.

Usuario:

Contraseña:

Confirmar Contraseña:

Registrarse

5.- Investiga y describe los siguientes servicios de autenticación:

- LDAP
- RADIUS
- TACACS+
- Kerberos

LDAP (Lightweight Directory Access Protocol): Es un protocolo utilizado para acceder y mantener servicios de directorio distribuidos sobre una red IP. LDAP almacena y organiza la información en una jerarquía, permitiendo que los administradores gestionen usuarios, grupos y recursos dentro de una red. Es ampliamente utilizado para la autenticación en aplicaciones empresariales, ya que permite a los sistemas validar credenciales contra un repositorio centralizado. Es compatible con varios sistemas operativos y puede integrarse con otros servicios como Active Directory.

RADIUS (Remote Authentication Dial-In User Service): Es un protocolo de red que proporciona autenticación centralizada, autorización y contabilidad (AAA) para usuarios que se conectan y usan un servicio de red. RADIUS es comúnmente utilizado por proveedores de servicios de Internet y redes empresariales para autenticar usuarios en conexiones inalámbricas o VPNs. Funciona como intermediario entre los servidores de autenticación y los puntos de acceso, verificando las credenciales del usuario antes de permitir el acceso.

TACACS+ (Terminal Access Controller Access-Control System Plus): Es un protocolo que proporciona servicios AAA, especialmente en redes empresariales y grandes infraestructuras. TACACS+ separa los procesos de autenticación, autorización y contabilidad, lo que ofrece un mayor control y flexibilidad para los administradores de redes. Se utiliza principalmente en dispositivos de red como routers y switches para controlar el acceso de los usuarios administrativos a los recursos del sistema.

Kerberos: Es un protocolo de autenticación diseñado para funcionar en redes abiertas y de cliente-servidor, proporcionando un método seguro para que los usuarios se autentifiquen entre sí. Kerberos utiliza criptografía de clave simétrica y un sistema de "tickets" para permitir que los usuarios se autentifiquen de manera segura sin enviar contraseñas en texto plano. Es muy utilizado en sistemas operativos como Windows y en redes empresariales donde se requiere una autenticación fuerte y centralizada.

6.- Investiga y describe los siguientes servicios de autorización:

- ACL
- RBAC
- ABAC
- PBAC

ACL (Listas de Control de Acceso): Es un mecanismo de autorización que especifica qué usuarios o sistemas tienen permisos para acceder a ciertos recursos en una red o sistema. Las listas de control de acceso contienen una lista de entradas donde cada una define un permiso para un usuario o grupo específico. Estos permisos pueden incluir lectura, escritura, ejecución, etc. LCA es una forma directa y sencilla de controlar el acceso, aunque su administración puede volverse complicada a medida que crece el número de usuarios y recursos.

Características:

- **Simplicidad:** Las listas de control de acceso (LCA) son simples de entender y usar, ya que asignan permisos directamente a usuarios o grupos específicos.
- **Permisos Específicos:** Cada entrada en la lista define qué tipo de acceso tiene un usuario o grupo a un recurso (lectura, escritura, ejecución, etc.).
- **Administración:** Puede volverse compleja a medida que aumenta el número de usuarios y recursos, requiriendo una gestión constante para mantener las listas actualizadas.
- **Granularidad:** Ofrece un control de acceso detallado, pero cada cambio requiere una actualización manual de la lista.

RBAC (Role-Based Access Control): Es un modelo de control de acceso basado en roles, donde los permisos para realizar operaciones están asociados con roles específicos dentro de una organización. En lugar de asignar permisos directamente a los usuarios, se les asignan roles, y cada rol tiene un conjunto de permisos predefinidos. Esto facilita la administración, ya que los permisos no se asignan individualmente, sino en función de los roles que desempeñan los usuarios, lo que mejora la eficiencia y seguridad en sistemas grandes.

Características:

- **Basado en Roles:** Los permisos se asignan a roles en lugar de a usuarios individuales. Los usuarios obtienen permisos al ser asignados a uno o más roles.
- **Escalabilidad:** Facilita la administración en grandes organizaciones, ya que los permisos se gestionan a nivel de rol y no de usuario individual.

ABAC (Attribute-Based Access Control): Es un modelo de autorización en el que los permisos se otorgan en función de atributos de los usuarios, los recursos y el entorno. Los atributos pueden ser cualquier cosa, desde la ubicación del usuario, la hora del día, el tipo de dispositivo que está utilizando o cualquier característica del usuario o recurso. Esto permite un control de acceso más granular y flexible en comparación con RBAC, ya que las decisiones de acceso pueden basarse en una combinación de múltiples atributos.

Características:

- **Basado en Atributos:** Los permisos se basan en una combinación de atributos de usuarios, recursos y contexto (como hora del día, ubicación, tipo de dispositivo).
- **Flexibilidad:** Permite un control de acceso más granular y adaptativo al considerar múltiples factores al tomar decisiones de acceso.
- **Políticas Complejas:** Utiliza políticas basadas en atributos para decidir el acceso, lo que puede permitir configuraciones más complejas.
- **Dinamismo:** Permite adaptaciones en tiempo real basadas en cambios en los atributos o contexto.

PBAC (Policy-Based Access Control): Este modelo de autorización está basado en políticas específicas definidas por una organización. Las políticas determinan las reglas de acceso basadas en diferentes condiciones y atributos, que pueden incluir roles, atributos de usuarios y otros factores contextuales. PBAC es similar a ABAC, pero se enfoca más en la definición y aplicación de políticas específicas que gestionan quién puede acceder a qué y bajo qué condiciones.

Características:

- **Basado en Políticas:** La autorización se gestiona mediante políticas definidas por la organización, que especifican las reglas y condiciones para el acceso.
- **Definición de Reglas:** Las políticas pueden incluir una combinación de roles, atributos y otros factores contextuales para determinar el acceso.
- **Flexibilidad en Políticas:** Permite una amplia personalización de reglas y condiciones de acceso, adaptándose a las necesidades específicas de la organización.
- **Centralización:** La administración de acceso se basa en la definición y aplicación de políticas, lo que puede simplificar la gestión en entornos complejos.



Conclusión

En esta práctica, se aplicó de manera práctica los conceptos de autenticación y autorización, que son fundamentales para garantizar la seguridad en aplicaciones web. Desarrollar la página web con formularios de registro e inicio de sesión me permitió entender cómo proteger el acceso a diferentes secciones, dependiendo de si el usuario está autenticado y su rol en el sistema.

Implementar estas funcionalidades no solo nos ayudó a comprender la importancia de controlar quién puede acceder a qué, sino también a explorar servicios de autenticación y autorización como LDAP, RADIUS, TACACS+, Kerberos, ACL, RBAC, ABAC y PBAC. Conocer estos servicios es esencial para aplicar medidas de seguridad más avanzadas en aplicaciones reales. Este trabajo fue muy útil para reforzar nuestros conocimientos y habilidades en seguridad de la información, logrando una aplicación segura y funcional. Fue una experiencia muy enriquecedora y práctica.



Bibliografía

<https://jumpcloud.com/es/blog/what-is-ldap-authentication>

<https://www.redeszone.net/tutoriales/servidores/que-es-servidor-radius-funcionamiento/>

https://www.cisco.com/c/es_mx/support/docs/security-vpn/remote-authentication-dial-user-service-radius/13838-10.html

<https://www.ibm.com/docs/es/aix/7.3?topic=network-kerberos>

<https://cloud.google.com/storage/docs/access-control/lists?hl=es-419>

<https://www.entrust.com/es/resources/learn/what-is-role-based-access-control>

<https://www.safepaas.com/es/articles/why-pbac-is-the-new-rbac/>