**siguientes tareas:**

**1- En el archivo navbar-logueado.php, no deberia haber un if para profesor y estudiante tambien? Solo pusiste este: <?php if ($\_SESSION['usuario']['rol'] == 'administrador'): ?> <li class="nav-item"> <a class="nav-link" href="administrador.html">Admin</a> </li> <?php endif; ?>**

**2- El diseño del navbar no me gusto mucho, quizas luego busquemos algo mejor en bootstrap, quizas el de hamburguesa, con los botones mejor organizados.**

**4- Hacer que, al momento de iniciar sesion, en el navbar aparezca el nombre del usuario que inicio sesion. Creo que intentaste hacer algo en el archivo navbar-logueado.php, pero no funciona, quizas necesita del apoyo de un archivo JS.**

**3- Necesitamos hacer que, el formulario de registro de Estudiante, que lo maneja el archivo registro.html, al momento de hacer un registro, quede asentado que se registro si o si un Rol Estudiante. Actualmente registra un usuario, pero sin indicar el Rol.**

**Paso 1: Implementar el sistema de autenticación**

1. **Login y creación de sesión:**
   * En login.php, utilizaremos session\_start() para iniciar una sesión y guardaremos los datos clave del usuario (como id, nombre, tipo\_usuario -si lo tienen o similar-) en $\_SESSION. Así, el sistema reconocerá si hay una sesión activa.
   * Modificar los archivos HTML y JavaScript para que, al iniciar sesión correctamente, redirijan al usuario según su rol a vista-estudiante.html, vista-profesor.html, o administrador.html.
2. **Redirección al no estar logueado:**
   * Como mencionaste, cada archivo de PHP que requiere al usuario estar logueado (por ejemplo, enviarMensaje.php, consultarMensajesEnviados.php, obtenerMensajes.php) necesitará verificar la existencia de la sesión.
   * Incluir session\_start() y un if (!isset($\_SESSION['id'])) para redirigir al index-login.html si la sesión no existe.

**Paso 2: Ajustar funcionalidades con sesiones en lugar de IDs fijos**

1. **Remplazar IDs harcodeados en mensajes y cursos:**
   * En enviarMensaje.php, consultarMensajesEnviados.php, y obtenerMensajes.php, los id del remitente y destinatario deberán obtenerse de $\_SESSION['id'].
   * Actualizar las funciones JavaScript en mensajes.js para que no necesiten enviar un id fijo, sino que utilicen la sesión activa.
2. **Redirección en funciones de carga de páginas principales:**
   * En index.php, configuraremos un navbar que incluya enlaces a inicio de sesión y registro (o el nombre del usuario activo si ya está logueado) para simplificar la navegación y brindar un mejor flujo a los estudiantes.

**Paso 3: Controlar el acceso a páginas por tipo de usuario**

1. **Añadir lógica de autorización:**
   * Index.php estará disponible solo para los estudiantes; vista-profesor.html y administrador.html se redirigirán según el rol guardado en la sesión (tipo\_usuario o similar).
   * En PHP, realizaremos una verificación $\_SESSION['tipo\_usuario'] para limitar el acceso a cada página o funcionalidad específica.
2. **Logout:**
   * Crear un archivo logout.php que cierre la sesión (llamando a session\_unset() y session\_destroy()) y redirija al index-login.html.
3. **Verificación en páginas HTML/Javascript:**
   * Para las páginas principales, agregar en vista-estudiante.html y las demás vistas un código JavaScript que revise si la sesión existe y redirija al login si no lo está. Esto es útil para evitar que los usuarios lleguen a esas páginas sin estar logueados.

**Paso 4: Pruebas y ajustes**

1. **Prueba de flujo de usuario:**
   * Realizar un testeo completo de flujo para cada rol: iniciar sesión, intentar acceder a páginas de otros roles, enviar mensajes, y consultar mensajes enviados y recibidos.
2. **Ajuste de detalles:**
   * Cualquier redirección adicional, mensajes de error para usuarios sin acceso, y la actualización de menús según el tipo de usuario.