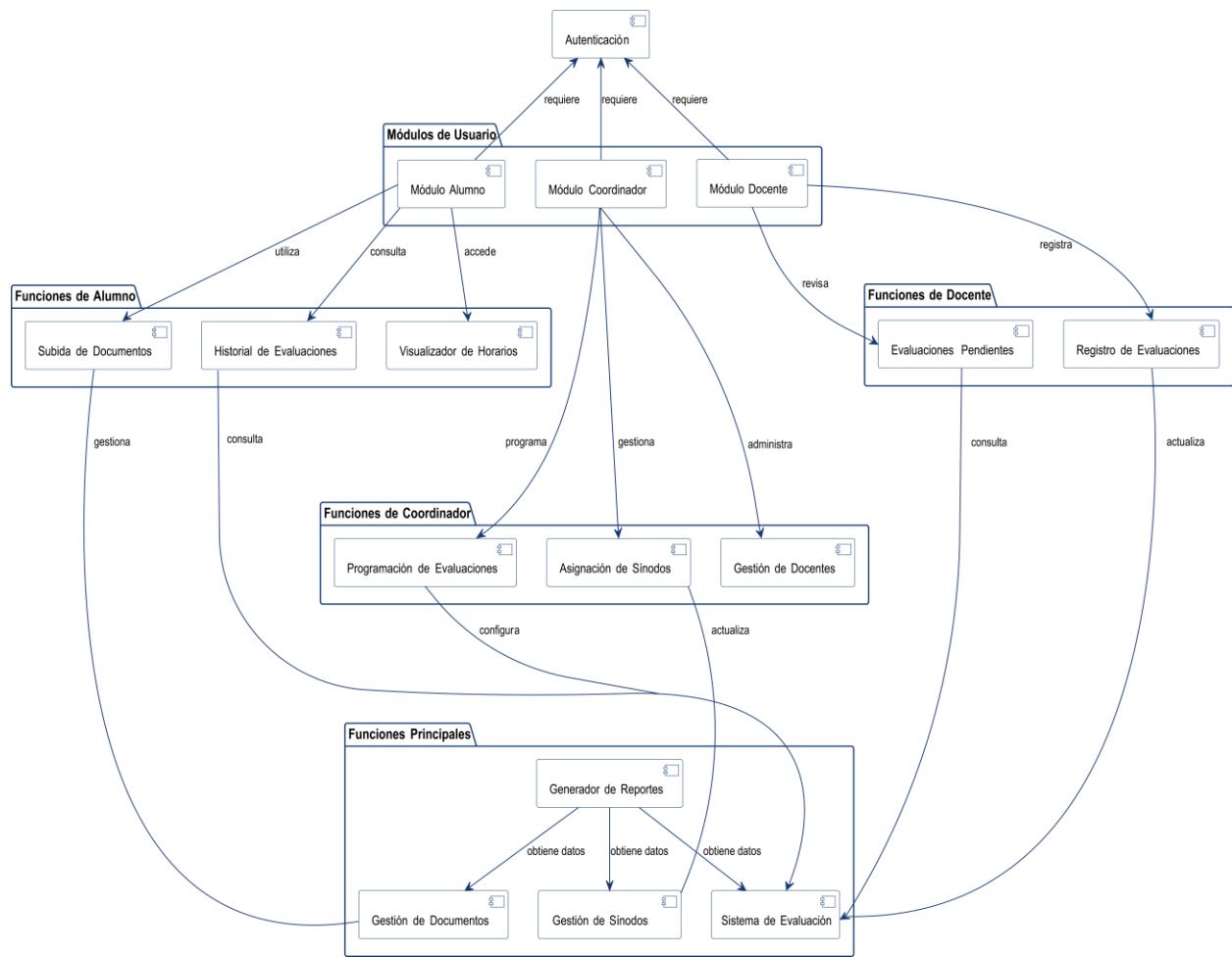


Diagrama de componentes



conexion.php

```
<?php

function Conectar(){
    $Servidor = "localhost";
    $Usuario = "root";
    $Password = "";
    $BD = "sistema_maestria";

    // Conexión orientada a objetos
    $Con = new mysqli($Servidor, $Usuario, $Password, $BD);

    // Verificar conexión
    if ($Con->connect_error) {
        die("Error de conexión: " . $Con->connect_error);
    }

    return $Con;
}

function Ejecutar($Con, $SQL){
    $Query = $Con->query($SQL) or die($Con->error);
    return $Query;
}

function Cerrar($Con){
    $Con->close();
}

?>
```

loginDesign.css

```
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

body {
    display: flex;
    justify-content: center;
    align-items: center;
    background-color: #ddefff !important;
}

.escudofif{
    display: flex;
    width: 35%;
    margin: auto;
    margin-bottom: 1.5rem;
}

.container1,
.container2 {
    background: #fff;
    color: #636363;
    top: 50%;
    left: 50%;
    position: absolute;
    transform: translate(-50%, -50%);
    box-sizing: border-box;
    padding: 2.5rem;
    border: 1px solid #9fb3c6;
    border-radius: 15px;
    font: caption;
    font-size: 1rem;
    z-index: 1;
    max-height: 90vh;
}
```

```
.container1 {
  max-width: 25vw;
}

.container2 {
  max-width: 60vw;
}

.container1 h1,
.container2 h1 {
  margin: 0;
  padding: 0 0 2rem;
  text-align: center;
  font-size: 1.7rem;
}

.container2 h2 {
  margin: 0;
  padding: 0 35px;
}

.container1 input[type="text"],
.container1 input[type="password"],
.container2 input[type="text"],
.container2 input[type="password"],
.container2 input[type="number"],
.container2 select,
.container2 input[type="email"] {
  width: 100%;
  margin-bottom: 1.5rem;
  border: none;
  border-bottom: 1px solid #636363;
  background: transparent;
  outline: none;
  height: 2rem;
  color: #000000;
  font-size: 0.9rem;
}

input::placeholder {
  color: rgb(200, 200, 200);
```

```
}

.boton_enviar {
    border: none;
    outline: none;
    height: 2.5rem;
    background: #134bb3;
    color: #fff;
    font-size: 1.125rem;
    border-radius: 5px;
    width: 50%;
    display: block;
    margin: 1rem auto 2rem auto;
    cursor: pointer;
}

.boton_enviar:hover {
    cursor: pointer;
    background: #092F75;
    color: rgb(255, 255, 255);
}

.boton_registrarse {
    text-align: center;
    border: none;
    height: 2.5rem;
    background: none;
    color: #134bb3;
    font-weight: bold;
    text-decoration: underline;
    font-size: 1.125rem;
    width: 50%;
    display: block;
    margin: 1.5rem auto 0 auto;
    cursor: pointer;
}

.boton_regresar {
    background: none;
    border: none;
    background: none;
```

```
text-align: center;
border: none;
font-size: 0.9rem;
font-weight: 700;
color: #737373;
cursor: pointer;
}

.boton_Regresar p::before {
content: "\25C4";
color: #7a7a7a;
margin-right: 0.5rem;
}

.grid-container {
display: grid;
grid-template-columns: repeat(3, 1fr);
gap: 2rem;
}

td {
padding: 0 1rem;
}

.espacio {
color: #fff;
font-size: large;
}

.año {
position: absolute;
bottom: 5%;
color: #6c757d !important;
}

/* Estilos para el pop-up */
.popup {
display: none;
position: fixed;
z-index: 1000;
left: 0;
```

```
    top: 0;
    width: 100%;
    height: 100%;
    background-color: rgb(0 0 0 / 70%);
}

.popup-content {
    top: 50%;
    left: 50%;
    position: absolute;
    transform: translate(-50%, -50%);
    background-color: #fff;
    padding: 3rem;
    width: 30%;
    align-content: center;
    text-align: center;
    border-radius: 1rem;
}

#popup-text {
    font: 1em sans-serif;
}

.close-btn {
    position: absolute;
    top: 0.5rem;
    right: 1rem;
    font-size: 1.7rem;
    cursor: pointer;
}

.copy-btn {
    margin: 2rem auto 0;
    padding: .5rem 1rem;
    font-size: 1rem;
    cursor: pointer;
}

/* Ajuste base para pantallas normales */
:root {
    font-size: 1rem;
```

```
}

@media screen and (max-width: 1600px) {
  .container1,
  .container2 {
    padding: 1.8rem;
  }

  .container1 h1,
  .container2 h1 {
    padding: 0 0 1.5rem;
  }

  .escudofif {
    margin-bottom: 1rem;
    width: 30%;
  }

  .grid-container {
    gap: 1.5rem;
  }
}
```

```
@media screen and (max-width: 1050px) {

  .container1,
  .container2 {
    max-width: 90vw;
  }

  .grid-container {
    grid-template-columns: repeat(2, 1fr);
  }

  .mt-5,
  .my-5 {
    display: none;
  }
}
```

```
.popup-content {
  width: 60%;
}

}

@media screen and (max-width: 770px) {
  .container1,
  .container2 {
    width: 90vw; /* Cambiamos a viewport width */
    max-width: none; /* Eliminamos el max-width para pantallas pequeñas */
  }
}

@media screen and (max-width: 450px) {

  .container1 h1,
  .container2 h1 {
    font-size: 1.5rem;
  }

  .container1,
  .container2 {
    font-size: 0.9rem;
    padding: 1.3rem;
  }

  .container1 input[type="text"],
  .container1 input[type="password"],
  .container2 input[type="text"],
  .container2 input[type="password"],
  .container2 input[type="number"],
  .container2 select,
  .container2 input[type="email"] {
    margin-bottom: 1rem;
    font-size: 0.8rem;
  }

  .popup-content {
    padding: 1rem;
    width: 80%;
  }
}
```

```
    height: 50%;  
}  
  
}  
  
.password-container {  
    position: relative;  
}  
  
.toggle-password {  
    position: absolute;  
    right: 0.5rem;  
    top: 50%;  
    transform: translateY(-50%);  
    background: none;  
    border: none;  
    cursor: pointer;  
}  
  
.eye-icon {  
    width: 1.5rem;  
    height: 1.5rem;  
    margin-bottom: 1.1rem;  
}  
  
/* Ocultar el ícono de mostrar/ocultar contraseña por defecto en Edge */  
input[type="password"]::-ms-reveal {  
    display: none;  
}
```

styles.css

```
@import
url('https://fonts.googleapis.com/css2?family=Poppins:ital,wght@0,100;0,200;0,300;0,400;0,500;
0,600;0,700;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

body {
    font-family: Poppins;
    margin: 0;
}

.header {
    display: flex;
    align-items: stretch;
    color: white;
    margin: 2vh 2vw;
    height: 9vh;
}

.header > div,
.header > .container-botones {
    padding: 0 1vw;
    display: flex;
    align-items: center;
    flex-grow: 1;
}

.container-titulo-header,
.container-botones,
.container-usuario,
.container-cerrar-btn {
    border-radius: clamp(.4rem, .4vw, .4rem);
    background: #001a37;
    height: 100%;
}

.container-titulo-header,
.container-botones,
.container-usuario {
    margin-right: 0.8vw;
}
```

```
}

.container-titulo-header,
.container-usuario, .container-cerrar-btn {
    justify-content: center;
}

.container-titulo-header {
    width: 15vw;
    flex-grow: 0!important;
}

.titulo-header {
    text-align: center;
    color: #fff;
    text-decoration: none;
    font-size: 2rem;
    line-height: 1.5rem;
    font-weight: bold;
}

.container-lista-botones {
    height: 100%;
    margin: 0;
    padding: 0;
    text-decoration: none;
    list-style-type: none;
    display: flex;
}

.boton-link {
    display: flex;
    align-items: center;
    margin-right: 1.5vw;
}

.boton-link:first-child {
    margin-left: 1vw; /* Agrega margen izquierdo solo al primer botón */
}

.boton-link a,
```

```
.boton-link button {
    font-family: Poppins;
    text-decoration: none;
    height: 80%;
    display: flex;
    align-items: center;
    background-color: transparent;
    color: white;
    border: none;
    cursor: pointer;
    font-size: 1rem;
    border-radius: clamp(.4rem, .4vw, .4rem);
}

.boton-link button:hover {
    background-color: #ffffff;
    color: rgb(0, 0, 0);
}

.boton-link button.active {
    background-color: #555; /* Color de fondo cuando está activo */
    color: white; /* Color del texto cuando está activo */
}

.usuario {
    font-family: Poppins;
    color: white;
    text-decoration: none;
    font-size: 1rem;
}

.container-cerrar-btn {
    background-color: rgb(255, 30, 30);
    width: 3.5vw;
    flex-grow: 0 !important; /* Asegura que no se expanda */
}

.container-cerrar-btn:hover {
    background-color: rgb(255, 64, 64);
}
```

```
.cerrar-btn {  
    text-decoration: none;  
    height: 100%;  
    display: flex;  
    align-items: center;  
}  
  
.cerrar-btn button {  
    background-color: transparent;  
    color: white;  
    border: none;  
    cursor: pointer;  
    font-size: 2vh;  
}  
  
/* Diseño para pantallas pequeñas */  
  
.container-titulo-sidebarbtn {  
    padding: 1rem;  
    display: flex;  
    align-items: center;  
    justify-content: space-between;  
    background-color: #001a37;  
    height: 5vh;  
    margin-bottom: 2vh;  
}  
  
.header-telefono {  
    display: none;  
}  
  
.sidebar {  
    position: fixed;  
    top: 0;  
    left: 0;  
    width: 100vw;  
    height: 100vh;  
    background-color: #001a37;  
    padding-top: 6vh;  
    z-index: 1000;  
    overflow-x: hidden;
```

```
    transition: 0.5s;
}

.espacio-sidebar {
    height: 7vh;
    display: block;
}

.sidebar a {
    padding: 1rem 2rem;
    text-decoration: none;
    font-size: 1.5rem;
    color: white;
    display: block;
    transition: 0.3s;
}

#usuario-sidebar {
    font-size: 2rem;
    font-weight: bold;
}

.sidebar .close-sidebar {
    color: white;
    position: absolute;
    top: 0;
    right: 0;
    padding: 0.8rem 1rem;
    font-size: 2.5rem;
}

.open-sidebar {
    font-size: 1.5rem;
    cursor: pointer;
    color: white;
    border: none;
    background-color: #001a37;
}

.cerrar-btn-telefono {
    background-color: rgb(255, 30, 30);
```

```
position: absolute;
bottom: 10%;
margin-left: 2rem;
border-radius: 1vw;
padding: 0.8rem 1.3rem !important;
}

@media screen and (max-width: 1600px) {

    .header {
        height: 11vh;
    }
}

/* Ajustes para pantallas pequeñas */
@media (max-width: 1420px) {

    .titulo-header {
        font-size: 1.7rem;
    }
}

@media (max-width: 820px) {

    .sidebar {
        width: 0;
    }

    .header {
        display: none;
    }

    .header-telefono {
        display: block;
    }

    .openbtn {
        display: block;
    }
}
```

tablas.css

```
body {  
    margin: 0;  
    padding: 0;  
    overflow: hidden;  
}  
  
:root {  
    --primary-color: rgb(26,115,232);  
    --secondary-color: #aaa;  
    --text-color: #3c4043;  
    --background-color: #fafcff;  
}  
  
.container-principal {  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    justify-content: center;  
    height: 81vh;  
    margin: 2vh 2vw;  
    padding: 2vh 2vw;  
    border-radius: clamp(.4rem, .4vw, .4rem);  
    background-color: #e9e9e9;  
}  
  
#table-container {  
    display: flex;  
    justify-content: center;  
    overflow-x: auto; /* Habilitar desplazamiento horizontal si es necesario */  
    overflow-y: auto; /* Habilitar desplazamiento vertical dentro del contenedor */  
    width: 100%;  
}  
  
table {  
    border-collapse: collapse;  
    width: 100%;  
    max-width: 100%; /* Asegurar que la tabla no sobrepase el contenedor */  
    table-layout: fixed;  
}
```

```
}

tr {
    border-top: 0.1rem solid var(--primary-color);
    border-bottom: 0.1rem solid var(--secondary-color);
}

th, td {
    border-bottom: 0.0625rem solid var(--secondary-color);
    padding: 1.25rem;
}

td {
    display: table-cell;
    text-align: center;
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    font-size: 1.1rem;
    font-weight: 500;
    color: var(--text-color);
}

th {
    letter-spacing: .01785714em;
    font-family: system-ui;
    font-weight: 600;
    font-size: 1.5rem;
    color: var(--text-color);
    padding-bottom: 2rem;
    padding-top: 3.5rem;
}

h1 {
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    text-align: center;
}

h3 {
    text-align: center;
    font-size: 2rem;
    font-family: "Google Sans", Roboto, Arial, sans-serif;
}
```

```
.input-container input {
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    font-size: 1rem;
    font-weight: 500;
    color: var(--text-color);
    border: 1px solid #ccc;
    padding: 0.5rem;
    border-radius: 0.5rem;
    width: 14vh;
}

.input-container {
    display: flex;
    align-items: center;
    justify-content: center;
}

@media screen and (max-width: 1600px) {

    .container-principal {
        height: 79vh;
    }

}

@media screen and (max-width: 820px) {

    .container-principal {
        height: 83.5vh;
    }

}

@media (max-width: 770px) {
    table {
        width: 100%;
    }

    thead {
        display: none; /* Ocultamos el encabezado en móvil */
    }
}
```

```
tr {  
  display: block;  
  margin-bottom: 1rem;  
  border: 1px solid var(--primary-color);  
  border-radius: clamp(.4rem, .4vw, .4rem);  
  padding: 1rem;  
  background: white;  
}  
  
td {  
  display: flex;  
  justify-content: space-between;  
  align-items: center;  
  padding: 0.5rem;  
  text-align: right;  
  border: none;  
}  
  
td::before {  
  content: attr(data-label);  
  font-weight: bold;  
  text-align: left;  
  padding-right: 0.5rem;  
}  
}
```

eliminar_evaluacion.php

```
<?php
// Incluir archivo de conexión si es necesario
include '../../../../../Config/conexion.php';

// Conectar a la base de datos
$Con = Conectar();

// Verificar si la solicitud es de tipo POST y si contiene el expediente
if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['accion']) && $_POST['accion'] ===
'eliminar' && isset($_POST['expediente'])) {
    // Limpiar el expediente recibido
    $expediente = intval($_POST['expediente']);

    // Preparar la consulta de eliminación
    $SQL = "DELETE FROM evaluaciones WHERE exp_alumno = '$expediente'";

    // Ejecutar la consulta
    if (Ejecutar($Con, $SQL)) {
        echo "Evaluación eliminada correctamente.";
    } else {
        echo "Error al eliminar la evaluación.";
    }

    // Cerrar la conexión
    Cerrar($Con);
} else {
    echo "Solicitud inválida.";
}
?>
```

get_documents.php

```
<?php
session_start();
include '../../../../../Config/conexion.php';

if (!isset($_SESSION['id'])) {
    echo json_encode(['success' => false, 'message' => 'Usuario no autenticado']);
    exit;
}

$Con = Conectar();
$userID = $_SESSION['id'];

$SQL = "SELECT * FROM documentos_alumno WHERE exp_alumno = ? ORDER BY fecha_subida DESC";
$stmt = $Con->prepare($SQL);
$stmt->bind_param("s", $userID);
$stmt->execute();
$result = $stmt->get_result();

$documents = [];
while ($row = $result->fetch_assoc()) {
    $documents[] = [
        'id' => $row['id'],
        'date' => $row['fecha_subida'],
        'type' => $row['tipo'],
        'fileName' => $row['nombre_archivo'],
        'fileURL' => $row['ruta']
    ];
}

echo json_encode(['success' => true, 'data' => $documents]);
?>
```

AgendarEvaluacion.php

```
<?php
    include('../Header/MenuA.php');
    if(session_status() === PHP_SESSION_NONE){
        session_start();
    }
    include('../../Config/conexion.php');
    $Con = Conectar();
    $id = $_SESSION['id'];

    $SQL = "
SELECT e.exp, e.nombre, e.a_paterno, e.a_materno
FROM estudiantes e
LEFT JOIN evaluaciones ev ON e.exp = ev.exp_alumno
LEFT JOIN asignaciones a ON e.exp = a.exp_alumno
WHERE ev.fecha_evaluacion IS NULL
AND a.exp_alumno = '$id';
";
    $Res = Ejecutar($Con, $SQL);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Agendar Evaluación</title>
    <style>

        .input-container input {
            font-family: "Google Sans", Roboto, Arial, sans-serif;
            font-size: 1rem;
            font-weight: 500;
            color: var(--text-color);
            border: 1px solid #ccc;
            padding: 0.5rem;
            border-radius: 0.5rem;
            width: 7vw;
        }

    </style>
```

```
}

.input-container {
    display: flex;
    align-items: center;
    justify-content: center;
}

.subir-entregable {
    color: #123773;
    font-size: 1.5rem;
    padding: 0.5rem 0.9rem;
    background-color: #ffffff;
    border: none;
    cursor: pointer;
    border-radius: 0.4rem;
    border-bottom: 0.0625rem solid var(--secondary-color);
}

.subir-entregable:hover {
    background-color: #cfcfcf;
}

.confirmar-icon {
    color: #123773;
    font-size: 1.5rem;
    padding: 0.5rem 0.9rem;
    background-color: #ffffff;
    border: none;
    cursor: pointer;
    border-radius: 0.4rem;
    border-bottom: 0.0625rem solid var(--secondary-color);
}

.confirmar-icon:hover {
    background-color: #cfcfcf;
}

.subir-entregable.archivo-subido {
    background-color: #cfcfcf; /* Verde claro para indicar archivo subido */
}
```

```
.quitar-archivo {
    color: red;
    margin-left: 10px;
    cursor: pointer;
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    font-size: 0.9rem;
    display: none; /* Inicialmente oculto */
}

@media (max-width: 770px) {
    .modal-content {
        width: 80%;
    }
    .input-container input {
        width: 27vw;
    }
    input[type="checkbox"] {
        cursor: pointer;
        width: 1.5rem;
        height: 1.5rem;
    }
    .confirmar-icon {
        background-color: #118f1d;
        color: white;
        padding: 0.7rem 0.9rem;
        border: none;
        cursor: pointer;
    }

    .confirmar-icon::before {
        font-family: "Google Sans", Roboto, Arial, sans-serif;
        font-size: 1.1rem;
        font-weight: 600;
        content: 'Confirmar';
    }

    .confirmar-icon {
        font-size: 0;
    }
}
```

```
.subir-entregable {  
    background-color: #123773;  
    color: white;  
    padding: 0.7rem 0.9rem;  
    border: none;  
    cursor: pointer;  
    width: 31vw;  
}  
  
.subir-entregable::before {  
    font-family: "Google Sans", Roboto, Arial, sans-serif;  
    font-size: 1.1rem;  
    font-weight: 500;  
    content: 'Adjuntar';  
}  
  
.subir-entregable {  
    font-size: 0;  
}  
  
.quitar-archivo::before {  
    font-size: 1.5rem;  
    content: '✖';  
}  
  
.quitar-archivo {  
    font-size: 0; /* Oculta el texto original */  
}  
}  
}  
</style>  
</head>  
  
<body>  
    <div class="container-principal">  
        <h3>Agendar Evaluación:</h3>  
        <div id="table-container">  
            <table>  
                <thead>  
                    <tr>  
                        <th>Fecha</th>  
                        <th>Hora</th>
```

```

        <th>Aula</th>
        <th>Entregable</th>
        <th>Confirmar</th>
    </tr>
</thead>
<tbody>
<?php
if ($Res->num_rows > 0){
    while($Fila = $Res->fetch_assoc()){
        $exp = $Fila["exp"];
        echo "<tr>";
        echo "<td data-label='Fecha'><div class='input-container'><input type='date' id='fecha-' . $exp . ''><span class='check-icon'></span></div></td>";
        echo "<td data-label='Hora'><div class='input-container'><input type='time' id='hora-' . $exp . ''><span class='check-icon'></span></div></td>";
        echo "<td data-label='Aula'><div class='input-container'><input type='text' id='aula-' . $exp . ''><span class='check-icon'></span></div></td>";
        echo "<td data-label='Entregable'>
                <input type='file' id='file-' . $exp . '' style='display: none;' accept='.pdf,.docx,.xlsx'>
                <div style='display: flex; align-items: center; justify-content: center;''>
                    <button id='upload-btn-' . $exp . '' class='subir-entregable' onclick='uploadDocument(\"" . $exp . "\")'>📁</button>
                    <span id='quitar-' . $exp . '' class='quitar-archivo' onclick='quitarArchivo(\"" . $exp . "\")'>Quitar archivo &times;</span>
                </div>
            </td>";
        echo "<td data-label='Confirmar'><button id='confirm-btn-' . $exp . '' class='confirmar-icon' onclick='confirmarEvaluacion(\"" . $exp . "\")' disabled>✓</button></td>";
        echo "</tr>";
    }
} else{
    echo "<tr><td colspan='5'>No se encontraron evaluaciones</td></tr>";
}
Cerrar($Con);
?>
</tbody>
</table>
</div>

```

```
</div>

<script>
    function confirmarEvaluacion(expediente) {
        const fechaSeleccionada = document.getElementById('fecha-' + expediente).value;
        const horaSeleccionada = document.getElementById('hora-' + expediente).value;
        const aula = document.getElementById('aula-' + expediente).value;
        const fileInput = document.getElementById('file-' + expediente);

        // Verificar que los campos no estén vacíos
        if (!fechaSeleccionada || !horaSeleccionada || !aula || !fileInput.files.length) {
            alert('Por favor, completa todos los campos y selecciona un archivo.');
            return;
        }

        // Combina la fecha y hora en el formato DATETIME (YYYY-MM-DD HH:MM:SS)
        const fechaHoraCombinada = fechaSeleccionada + ' ' + horaSeleccionada;

        // Crear un objeto FormData
        const formData = new FormData();
        formData.append('exp', expediente);
        formData.append('fecha_evaluacion', fechaHoraCombinada);
        formData.append('aula', aula);
        formData.append('entregable', fileInput.files[0]);

        // Crear una solicitud XMLHttpRequest
        const xhr = new XMLHttpRequest();
        xhr.open('POST', '../../../../../Posgrados_FIF/insertar_evaluacion.php', true);

        // Manejo de la respuesta del servidor
        xhr.onreadystatechange = function () {
            if (xhr.readyState === 4 && xhr.status === 200) {
                alert(xhr.responseText); // Mostrar respuesta del servidor
                location.reload(); // Recargar la página
            }
        };

        // Manejo de errores de la solicitud
        xhr.onerror = function () {
            console.error('Error de red');
            alert('Ocurrió un error al procesar la solicitud');
        };
    }
</script>
```

```

    };

    // Enviar la solicitud con los datos
    xhr.send(formData); // Enviar el FormData, que contiene todos los datos
}

function uploadDocument(expediente) {

    const fileInput = document.getElementById('file-' + expediente);
    const uploadBtn = document.getElementById('upload-btn-' + expediente);
    const quitarBtn = document.getElementById('quitar-' + expediente);
    const confirmBtn = document.getElementById('confirm-btn-' + expediente);

    fileInput.click();

    fileInput.addEventListener('change', function() {
        if (fileInput.files.length > 0) {
            const file = fileInput.files[0];
            console.log('Archivo seleccionado:', file.name);

            // Cambiar apariencia del botón
            uploadBtn.classList.add('archivo-subido');
            quitarBtn.style.display = 'inline';

            // Habilitar el botón "Confirmar"
            confirmBtn.disabled = false;
        }
    });
}

function quitarArchivo(expediente) {
    const fileInput = document.getElementById('file-' + expediente);
    const uploadBtn = document.getElementById('upload-btn-' + expediente);
    const quitarBtn = document.getElementById('quitar-' + expediente);
    const confirmBtn = document.getElementById('confirm-btn-' + expediente);

    // Limpiar el input file
    fileInput.value = '';

    // Restaurar apariencia original
    uploadBtn.classList.remove('archivo-subido');
}

```

```
quitarBtn.style.display = 'none';

// Deshabilitar el botón "Confirmar"
confirmBtn.disabled = true;
}

</script>

</body>

</html>
```

detalles_evaluacion.php

```
<?php
include '../../../Config/conexion.php';

$Con = Conectar();
$id_evaluacion = $_GET['id_evaluacion'];

// Verificar que se recibió el ID de la evaluación
if (!$id_evaluacion) {
    echo "No se recibió un ID de evaluación";
    exit();
}

// Consulta SQL para obtener los detalles de los sinodales
$SQL = "SELECT de.calificacion, de.observacion
        FROM detalle_evaluaciones de
        WHERE de.id_evaluacion = '$id_evaluacion'";
$Resultado = mysqli_query($Con, $SQL);

// Crear la tabla de detalles
if ($Resultado && mysqli_num_rows($Resultado) > 0) {
    $sinodales = ["Sinodo 1", "Sinodo 2", "Sinodo 3", "Sinodo 4"];
    $i = 0;

    // Iterar sobre los resultados de la consulta
    while ($fila = mysqli_fetch_assoc($Resultado)) {
        echo "<tr>";
        echo "<td>" . $sinodales[$i] . "</td>"; // Nombre del sinodal
        echo "<td>" . $fila['calificacion'] . "</td>"; // Calificación del sinodal
        echo "<td>" . $fila['observacion'] . "</td>";
        echo "</tr>";
        $i++;
    }
} else {
    echo "<tr><td colspan='3'>No se encontraron detalles para esta evaluación</td></tr>";
}
?>
```

Documentos.php

```
<?php
    include('../Header/MenuA.php');
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Documentos</title>
</head>

<style>
    .container-subirdoc {
        font: caption;
        display: flex;
        flex-direction: column;
        justify-content: center;
        align-items: center;
        margin-bottom: 1rem;
    }

    .container-tipodoc,
    .container-subir {
        margin-bottom: 1rem;
        display: flex;
        align-items: center;
    }

    label {
        font-size: 1.1rem;
        margin-right: 0.4rem;
    }

    #document-type {
        font-size: 1rem;
    }
</style>
```

```
padding: 0.7rem 1rem;
width: 13rem;
border: 1px solid #ccc;
border-radius: 0.4rem;
line-height: 1.5rem;
}

#file-label {
    font-size: 1rem;
    display: inline-block;
    width: 13rem;
    padding: 0.7rem 1rem;
    border: 1px solid #ccc;
    border-radius: 0.4rem 0 0 0.4rem;
    cursor: not-allowed;
    white-space: nowrap;
    overflow: hidden;
    text-overflow: ellipsis;
    line-height: 1.5rem;
    vertical-align: middle;
    background-color: white;
}

#clear-file-btn {
    font-size: 1rem;
    display: none;
    margin-left: 10px;
    color: red;
    cursor: pointer;
    line-height: 1.5rem;
}

#upload-btn {
    font-size: 0.9rem;
    display: inline-block;
    padding: 0.7rem 1rem;
    background-color: grey;
    color: white;
    border: none;
    cursor: not-allowed;
    border-radius: 0 0.4rem 0.4rem 0;
}
```

```
    vertical-align: middle;
    line-height: 1.62rem;
    transition: background-color 0.3s ease;
}

#upload-btn.option-selected {
    background-color: #123773;
    /* Azul cuando se ha seleccionado una opción */
    cursor: pointer;
    pointer-events: auto;
}

#upload-btn.option-selected:hover {
    background-color: #1A4DA1;
}

#upload-btn.enabled {
    background-color: #4CAF50;
    /* Verde cuando se sube un archivo */
    cursor: pointer;
    pointer-events: auto;
}

.ver-doc {
    color: #123773;
    font-size: 2rem;
    padding: 0.25rem 0.7rem;
    background-color: #ffffff;
    border: none;
    cursor: pointer;
    border-radius: clamp(.4rem, .4vw, .4rem);
    border-bottom: 0.0625rem solid var(--secondary-color);
}
.ver-doc:hover {
    background-color: #cfcfcf;
}

@media (max-width: 770px) {
    #file-label {
        width: 6rem;
    }
}
```

```

#upload-btn {
    height: 3.01rem;
    padding: 0.5rem;
}
#document-type {
    width: 12rem;
}
.ver-doc {
    background-color: #123773;
    color: white;
    padding: 0.7rem 0.9rem;
    border: none;
    cursor: pointer;
}

.ver-doc::before {
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    font-size: 1.1rem;
    font-weight: 600;
    content: 'Ver archivo';
}

.ver-doc {
    font-size: 0;
}
}

</style>

<body>
    <div class="container-principal">
        <div class="container-subirdoc">
            <!-- Sección de selección del tipo de documento -->
            <div class="container-tipodoc">
                <label for="document-type">Tipo:</label>
                <select id="document-type" onchange="enableFileSection()">
                    <option value="">Selecciona un tipo</option>
                    <option value="Recibo">Recibo</option>
                    <option value="Protocolo">Protocolo</option>
                    <option value="Registro de tesis">Registro de tesis</option>
                </select>
            </div>
        </div>
    </div>

```

```

<!-- Sección de subir archivo -->
<div class="container-subir">
    <label for="file">Subir:</label>
    <span id="file-label" onclick="triggerFileInput()">Examinar...</span>
    <input type="file" id="file" name="file" style="display:none;" accept=".pdf,.docx,.xlsx"
        onchange="handleFileSelect()" disabled>
    <button id="upload-btn" onclick="uploadDocument()">Subir archivo</button>
    <span id="clear-file-btn" onclick="clearFile()">Quitar archivo &times;</span>
</div>
</div>
<h3>Documentos subidos:</h3>
<div id="table-container">

    <table>
        <thead>
            <tr>
                <th>Fecha</th>
                <th>Tipo</th>
                <th>Vista</th>
            </tr>
        </thead>
        <tbody id="documents-table-body">
            <!-- Los documentos se cargarán aquí dinámicamente -->
        </tbody>
    </table>
</div>

</div>

<script>
    const fileInput = document.getElementById('file');
    const fileLabel = document.getElementById('file-label');
    const clearFileBtn = document.getElementById('clear-file-btn');
    const uploadBtn = document.getElementById('upload-btn');
    const documentType = document.getElementById('document-type');

    function enableFileSection() {
        // Habilitar la parte de subir archivo si se selecciona un tipo válido
        if (documentType.value !== "") {

```

```

        inputFile.disabled = false; // Habilitar input de archivo
        fileLabel.style.cursor = 'pointer';
        uploadBtn.style.cursor = 'pointer'; // Cambiar cursor para indicar
habilitación
        uploadBtn.classList.add("option-selected"); // Cambia el botón a azul
        uploadBtn.classList.remove("enabled"); // Asegúrate de que no esté en verde
    } else {
        inputFile.disabled = true; // Deshabilitar input de archivo
        uploadBtn.style.cursor = 'not-allowed';
        fileLabel.style.cursor = 'not-allowed'; // Cambiar cursor para indicar
deshabilitación
        clearFile(); // Limpiar campos
    }
}

function triggerTextInput() {
    // Solo permitir abrir el diálogo si el input no está deshabilitado
    if (!fileInput.disabled) {
        fileInput.click();
    }
}

function handleFileSelect() {
    var inputFile = document.getElementById("file");
    var fileLabel = document.getElementById("file-label");
    var uploadBtn = document.getElementById("upload-btn");
    var clearFileBtn = document.getElementById("clear-file-btn");

    if (fileInput.files.length > 0) {
        // Actualizar el label con el nombre del archivo seleccionado
        fileLabel.innerText = fileInput.files[0].name;
        uploadBtn.classList.add("enabled"); // Cambiar el color del botón a verde
        uploadBtn.classList.remove("option-selected"); // Remueve el color azul
        clearFileBtn.style.display = "inline-block"; // Mostrar el botón de quitar
archivo
    }
}

function clearFile() {
    var inputFile = document.getElementById("file");
    var fileLabel = document.getElementById("file-label");

```

```

var uploadBtn = document.getElementById("upload-btn");
var clearFileBtn = document.getElementById("clear-file-btn");

// Limpiar el input de archivo
fileInput.value = "";
fileLabel.innerText = "Examinar..."; // Restablecer el label
uploadBtn.classList.remove("enabled"); // Volver a azul (opción seleccionada)
uploadBtn.classList.add("option-selected");
clearFileBtn.style.display = "none"; // Ocultar el botón de quitar archivo
}

function uploadDocument() {
    const fileInput = document.getElementById('file');
    const documentType = document.getElementById('document-type');

    if (fileInput.files.length > 0 && documentType.value !== "") {
        const formData = new FormData();
        formData.append('file', fileInput.files[0]);
        formData.append('documentType', documentType.value);

        uploadBtn.disabled = true;
        uploadBtn.textContent = 'Subiendo...';

        fetch('upload_document.php', {
            method: 'POST',
            body: formData
        })
        .then(response => response.json())
        .then(data => {
            if (data.success) {
                // Actualizar la tabla directamente
                updateDocumentsTable();

                // Limpiar el formulario
                clearFile();
                documentType.value = "";
                enableFileSection();

                alert('Archivo subido correctamente');
            } else {
                alert('Error: ' + data.message);
            }
        })
    }
}

```

```

        }
    })
    .catch(error => {
        console.error('Error:', error);
        alert('Error al subir el archivo');
    })
    .finally(() => {
        uploadBtn.disabled = false;
        uploadBtn.textContent = 'Subir archivo';
    });
}
}

function updateDocumentsTable() {
    fetch('../Acciones globales/get_documents.php')
        .then(response => response.json())
        .then(result => {
            if (result.success) {
                const tableBody = document.getElementById('documents-table-body');
                tableBody.innerHTML = '';

                result.data.forEach(doc => {
                    const row = tableBody.insertRow();

                    const dateCell = row.insertCell(0);
                    dateCell.setAttribute('data-label', 'Fecha');
                    dateCell.textContent = doc.date;

                    const typeCell = row.insertCell(1);
                    typeCell.setAttribute('data-label', 'Tipo');
                    typeCell.textContent = doc.type;

                    const viewCell = row.insertCell(2);
                    viewCell.setAttribute('data-label', 'Vista');
                    viewCell.innerHTML = `
                        <span class="ver-doc"
                            onclick="viewDocument('${doc.fileURL}')">@</span>
                    `;
                });
            }
        });
}

```

```
.catch(error => console.error('Error:', error));
}

function viewDocument(fileURL) {
    // Abrir el documento en una nueva pestaña
    window.open(fileURL, '_blank');
}

// Cargar documentos al iniciar la página
updateDocumentsTable();
</script>

</body>

</html>
```

EvaluacionesAgendadas.php

```
<?php
    include('../Header/MenuA.php');
    if(session_status() === PHP_SESSION_NONE){
        session_start();
    }

    include('../../Config/conexion.php');
    $Con = Conectar();
    $id = $_SESSION['id'];

    $SQL = "
        SELECT exp_alumno, fecha_evaluacion, aula FROM evaluaciones WHERE exp_alumno = '$id'
    ";

    $Res = Ejecutar($Con, $SQL);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Evaluaciones Agendadas</title>
    <style>

        .eliminar-icon {
            color: #123773;
            font-size: 1rem;
            padding: 0.7rem 0.9rem 0.75rem;
            background-color: #ffffff;
            border: none;
            cursor: pointer;
            border-radius: clamp(.4rem, .4vw, .4rem);
            border-bottom: 0.0625rem solid var(--secondary-color);
        }

        .eliminar-icon:hover {
```

```
        background-color: #cfcfcf;
    }

    @media (max-width: 770px) {
        .eliminar-icon {
            background-color: red;
            color: white;
            padding: 0.7rem 0.9rem;
            border: none;
            cursor: pointer;
        }

        .eliminar-icon::before {
            font-family: "Google Sans", Roboto, Arial, sans-serif;
            font-size: 1.1rem;
            font-weight: 600;
            content: 'Confirmar';
        }

        .eliminar-icon {
            font-size: 0;
        }
    }
}

</style>
</head>

<body>
    <div class="container-principal">
        <h3>Evaluaciones Agendadas:</h3>
        <div id="table-container">
            <table>
                <thead>
                    <tr>
                        <th>Fecha</th>
                        <th>Hora</th>
                        <th>Aula</th>
                        <th>Entregables</th>
                        <th>Eliminar</th>
                    </tr>
                </thead>
                <tbody>
```

```

<?php

if ($Res->num_rows > 0){
    while($Fila = $Res->fetch_assoc()){
        $exp = $Fila["exp_alumno"];
        $fechaCompleta = $Fila['fecha_evaluacion'];

        // Dividimos la fecha y la hora
        $fecha = date("Y-m-d", strtotime($fechaCompleta));
        $hora = date("H:i:s", strtotime($fechaCompleta));

        // Obtener lista de archivos en la carpeta de entregables
        $entregables = [];
        $dir = "../../Entregables/$exp/";
        if (is_dir($dir)) {
            $files = scandir($dir);
            foreach ($files as $file) {
                if ($file != '.' && $file != '..') {
                    $entregables[] = "<a href='$dir$file' target='_blank'$file</a>";
                }
            }
        }
        $entregablesContent = empty($entregables) ? "No disponible" : implode(", ", $entregables);

        echo "<tr id='fila-$exp'>";
        echo "<td data-label='Fecha'>$fecha</td>";
        echo "<td data-label='Hora'>$hora</td>";
        echo "<td data-label='Aula'>$Fila['aula']</td>";
        echo "<td data-label='Entregables'>$entregablesContent</td>";
        echo "<td data-label='Eliminar'><button class='eliminar-icon' onclick='eliminarEvaluacion(\"$exp\")'>X</button></td>";
        echo "</tr>";
    }
} else {
    echo "<tr><td colspan='5'>No se encontraron evaluaciones agendadas</td></tr>";
}
Cerrar($Con);
?>
</tbody>
</table>

```

```
</div>
</div>

</body>
<script>
    function eliminarEvaluacion(expediente) {
        // Confirmar eliminación
        if (confirm('¿Estás seguro de que quieres eliminar esta evaluación?')) {
            // Crear el objeto XMLHttpRequest
            var xhr = new XMLHttpRequest();

            // Configurar la solicitud
            xhr.open('POST', '../Acciones globales/eliminar_evaluacion.php', true);
            xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

            // Definir lo que sucederá cuando la solicitud se complete
            xhr.onload = function() {
                if (xhr.status === 200) {
                    // Eliminar la fila de la tabla si la solicitud fue exitosa
                    var fila = document.getElementById('fila-' + expediente);
                    if (fila) {
                        fila.remove();
                    }
                    alert('Evaluación eliminada exitosamente.');
                } else {
                    alert('Hubo un error al eliminar la evaluación.');
                }
            };
            // Enviar la solicitud con el expediente del alumno
            xhr.send('accion=eliminar&expediente=' + expediente);
        }
    }
</script>
</html>
```

upload_document.php

```
<?php
session_start();
include '../../../../../Config/conexion.php';

// Obtener la conexión
$Con = Conectar();

// Agregar log de depuración
error_log("Iniciando upload_document.php");

// Verificar si el usuario está autenticado
if (!isset($_SESSION['id'])) {
    error_log("Usuario no autenticado");
    echo json_encode(['success' => false, 'message' => 'Usuario no autenticado']);
    exit;
}

// Configurar la respuesta como JSON
header('Content-Type: application/json');

// Verificar si se recibió un archivo
if (!isset($_FILES['file']) || !isset($_POST['documentType'])) {
    error_log("No se recibió archivo o tipo de documento");
    echo json_encode(['success' => false, 'message' => 'No se recibió el archivo o el tipo de documento']);
    exit;
}

// Obtener el ID del usuario de la sesión
$userId = $_SESSION['id'];
error_log("ID de usuario: " . $userId);

// Sanitizar el nombre de la carpeta del tipo de documento
$documentType = $_POST['documentType'];
$typeFolderName = str_replace(' ', '_', $documentType);
$typeFolderName = preg_replace('/[^\w]/', '', $typeFolderName);

// Definir la ruta base para los documentos
```

```

$baseDir = '../../Documentos/';
$userDir = $baseDir . $userId . '/';
$typeDir = $userDir . $typeFolderName . '/';

error_log("Intentando crear directorios:");
error_log("baseDir: " . $baseDir);
error_log("userDir: " . $userDir);
error_log("typeDir: " . $typeDir);

// Crear directorios si no existen
foreach ($baseDir, $userDir, $typeDir] as $dir) {
    if (!file_exists($dir)) {
        if (!mkdir($dir, 0755, true)) {
            error_log("Error al crear directorio: " . $dir);
            echo json_encode(['success' => false, 'message' => 'Error al crear directorio: ' .
$dir]);
            exit;
        }
    }
}

// Obtener información del archivo
$file = $_FILES['file'];
error_log("Información del archivo:");
error_log("Nombre: " . $file['name']);
error_log("Tipo: " . $file['type']);
error_log("Tamaño: " . $file['size']);
error_log("Error: " . $file['error']);

$fileName = $file['name'];
$targetPath = $typeDir . time() . '_' . $fileName;

// Verificar el tipo de archivo
$allowedTypes = [
    'application/pdf',
    'application/msword',
    'application/vnd.openxmlformats-officedocument.wordprocessingml.document',
    'application/vnd.ms-excel',
    'application/vnd.openxmlformats-officedocument.spreadsheetml.sheet',
    'application/octet-stream'
];

```

```

if (!in_array($file['type'], $allowedTypes)) {
    error_log("Tipo de archivo no permitido: " . $file['type']);
    echo json_encode(['success' => false, 'message' => 'Tipo de archivo no permitido: ' .
$file['type']]);
    exit;
}

// Intentar mover el archivo
if (move_uploaded_file($file['tmp_name'], $targetPath)) {
    error_log("Archivo movido exitosamente a: " . $targetPath);

    // Guardar en la base de datos
    $SQL = "INSERT INTO documentos_alumno (exp_alumno, tipo, nombre_archivo, ruta,
fecha_subida)
VALUES (?, ?, ?, ?, CURDATE())";

    $stmt = $Con->prepare($SQL);
    if (!$stmt) {
        error_log("Error en prepare: " . $Con->error);
        echo json_encode(['success' => false, 'message' => 'Error al preparar la consulta']);
        exit;
    }

    $stmt->bind_param("isss", $userId, $documentType, $fileName, $targetPath);

    if ($stmt->execute()) {
        error_log("Registro guardado en la base de datos");
        echo json_encode([
            'success' => true,
            'message' => 'Archivo subido correctamente',
            'data' => [
                'id' => $stmt->insert_id,
                'fileName' => $fileName,
                'type' => $documentType,
                'path' => $targetPath,
                'date' => date('Y-m-d')
            ]
        ]);
    } else {
        error_log("Error al ejecutar la consulta: " . $stmt->error);
    }
}

```

```
    echo json_encode(['success' => false, 'message' => 'Error al guardar en la base de
datos']);
}
} else {
    error_log("Error al mover el archivo");
    echo json_encode(['success' => false, 'message' => 'Error al mover el archivo']);
}
?>
```

AltaDocentes.php

```
<?php
    include('../Header/HeaderC.php');
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Posgrado FIF</title>
    <style>
        .alta-docentes {
            width: 50%;
            top: 54%;
            left: 50%;
            position: absolute;
            transform: translate(-50%, -50%);
            box-sizing: border-box;
        }
        .alta-docentes h1 {
            font-family: "Google Sans", Roboto, Arial, sans-serif;
            color: #000000;
            margin: 0;
            margin: 0 0 2.5rem;
            text-align: center;
            font-size: 2rem;
        }
        .alta-docentes label {
            font-family: system-ui;
            color: #3c4043;
            font-weight: 600;
            font-size: 1.5rem;
            margin: 0;
            padding: 0 5px;
        }
        .alta-docentes input[type="text"],
```

```
.alta-docentes input[type="number"],  
.alta-docentes select {  
    font-family: "Google Sans", Roboto, Arial, sans-serif;  
    width: 96%;  
    margin-top: 0.5rem;  
    margin-bottom: 1.5rem;  
    border-bottom: 1px solid #636363;  
    outline: none;  
    font-size: 1rem;  
    font-weight: 500;  
    color: var(--text-color);  
    border: 1px solid #ccc;  
    padding: 1rem 0.5rem;  
    border-radius: clamp(.4rem, .4vw, .4rem);  
}  
  
input::placeholder {  
    color: rgb(200, 200, 200);  
}  
  
.boton_guardar {  
    border: none;  
    outline: none;  
    height: 3rem;  
    background: #123773;  
    color: #fff;  
    font-size: 1.5rem;  
    border-radius: clamp(.4rem, .4vw, .4rem);  
    width: 30%;  
    display: block;  
    margin: 3rem auto 0 auto;  
    cursor: pointer;  
}  
  
.grid-container {  
    margin-top: 2.5rem;  
    display: grid;  
    grid-template-columns: repeat(3, 1fr);  
    gap: 3rem;  
}
```

```
td {
    padding: 0 1rem;
}

.espacio {
    color: #fff;
    font-size: large;
}

/* Estilos para el pop-up */
.popup {
    display: none;
    position: fixed;
    z-index: 1000;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    background-color: rgb(0 0 0 / 70%);
}

.popup-content {
    top: 50%;
    left: 50%;
    position: absolute;
    transform: translate(-50%, -50%);
    background-color: #fff;
    padding: 3rem;
    align-content: center;
    text-align: center;
    border-radius: 1rem;
}

#popup-text {
    font: 1.1rem sans-serif;
}

.close-btn {
    position: absolute;
    top: 0.5rem;
    right: 1rem;
```

```
    font-size: 1.7rem;
    cursor: pointer;
}

.copy-btn {
    margin: 2rem auto 0;
    padding: .5rem 1rem;
    font-size: 1rem;
    cursor: pointer;
}

@media screen and (max-width: 1600px) {

    .alta-docentes {
        top: 56%;
        width: 60%;
    }

    .boton_guardar {
        margin: 2rem auto 0 auto;
    }

}

@media screen and (max-width: 1050px) {

    .alta-docentes {
        top: 56%;
        width: 90%;
    }

    .alta-docentes input[type="text"],
    .alta-docentes input[type="number"],
    .alta-docentes select {
        margin-bottom: 0;
    }

    .grid-container {
        grid-template-columns: repeat(2, 1fr);
        row-gap: 1rem;
    }
}
```

```
}

@media screen and (max-width: 450px) {
    .alta-docentes input[type="text"],
    .alta-docentes input[type="number"],
    .alta-docentes select {
        width: 90%;
    }

    .alta-docentes h1 {
        font-size: 1.5rem;
        margin: 0 0 3rem;
    }

    .alta-docentes label {
        font-size: 1.25rem;
    }

    .grid-container {
        gap: 1rem;
    }

    .boton_guardar {
        margin-top: 4rem;
        width: 50%;
    }

    .popup-content {
        padding: 2rem 1rem 1rem 1rem;
    }
}

</style>
</head>
<body>
<div class="container-principal">
<div class="alta-docentes" id="registrationOptions">
    <h3>Alta de Docentes</h3>
    <hr class="x-component x-component-default" style="border-top: 0; border-bottom: 0.05rem solid #196ad3; margin: auto; width: 100%;" id="box-1034">
    <form action="..../Coordinador/RegistroDocentes.php" method="POST">
```

```
<div class="grid-container">
  <div>
    <label for="expediente">Clave</label>
    <input type="number" name="clave" id="clave" required>
  </div>
  <div>
    <label for="nombre">Nombre:</label>
    <input type="text" name="nombre" id="nombre" required>
  </div>
  <div>
    <label for="apellidoPaterno">Apellido paterno:</label>
    <input type="text" name="apellidoPaterno" id="apellidoPaterno" required>
  </div>
  <div>
    <label for="apellidoMaterno">Apellido materno:</label>
    <input type="text" name="apellidoMaterno" id="apellidoMaterno" required>
  </div>
  <div>
    <label for="programa">Estatus: </label>
    <select name="status" id="status" required>
      <option value="" disabled selected>Selecciona...</option>
      <option value="A">Activ@</option>
      <option value="I">>Inactiv@</option>
    </select>
  </div>
  </div>
  <input type="submit" value="Guardar" class="boton_guardar">
</form>
</div>
</div>

<!-- Pop-up oculto inicialmente --&gt;
&lt;div id="popup" class="popup" style="display: none;"&gt;
  &lt;div class="popup-content"&gt;
    &lt;span class="close-btn" onclick="closePopup()"&gt;&amp;times;&lt;/span&gt;
    &lt;p id="popup-text"&gt;&lt;/p&gt;
    &lt;button id="copy-btn" class="copy-btn" onclick="copyToClipboard()"&gt;#x1F4CB; Copiar&lt;/button&gt;
  &lt;/div&gt;
&lt;/div&gt;
&lt;script&gt;</pre>
```

```
let operationSuccess = false; // Variable global para el estado de la operación

document.querySelector('form[action="../Coordinador/RegistroDocentes.php"]').addEventListener('submit', function(event) {
    event.preventDefault(); // Evita el envío del formulario de la manera tradicional

    const formData = new FormData(this);

    fetch('../Coordinador/RegistroDocentes.php', {
        method: 'POST',
        body: formData
    })
    .then(response => response.json())
    .then(data => {
        const popup = document.getElementById('popup');
        const popupText = document.getElementById('popup-text');
        const copyButton = document.getElementById('copy-btn');

        // Oculta el botón de copiar por defecto
        copyButton.style.display = 'none';

        if (data.status === 'success') {
            operationSuccess = true; // Marca la operación como exitosa
            // Mostrar el mensaje y el botón de copiar
            popupText.innerText = data.message;
            copyButton.style.display = 'block'; // Muestra el botón de copiar

            // Manejar la funcionalidad de copiado
            copyButton.addEventListener('click', function() {
                navigator.clipboard.writeText(data.copyText)
                .then(() => {
                    alert('Credenciales copiadas al portapapeles');
                })
                .catch(err => {
                    console.error('Error al copiar:', err);
                });
            });
        }
    } else if (data.status === 'exists') {
        // Mostrar el mensaje de que la cuenta ya existe sin el botón de copiar
        popupText.innerText = data.message;
    }
})
```

```

        } else if (data.status === 'error') {
            // Mostrar el mensaje de error
            popupText.innerText = data.message;
        }

        // Mostrar el popup
        popup.style.display = 'block';
    })
    .catch(error => {
        console.error('Error:', error);
        alert('Hubo un problema al registrar la cuenta.');
    });
});

function copyToClipboard() {
    if (window.copyText) {
        navigator.clipboard.writeText(window.copyText)
            .then(() => {
                alert("Credenciales copiadas al portapapeles");
            })
            .catch(err => {
                console.error('Error al copiar:', err);
                alert("Error al copiar el texto");
            });
    }
}

function closePopup() {
    document.getElementById("popup").style.display = "none";
    // Limpiar todos los campos del formulario

document.querySelector('form[action="../Coordinador/RegistroDocentes.php"]').reset();
}
</script>
</body>
</html>

```

AsignarSinodo.php

```
<?php
    include('../Header/MenuC.php');
?>

<?php
// Incluir el archivo de conexión
include '../../../../../Config/conexion.php';

// Conectar a la base de datos
$Con = Conectar();

// Consulta SQL para obtener los datos de los estudiantes
$SQL = "
    SELECT e.exp, e.nombre, e.a_paterno, e.a_materno
    FROM estudiantes e
    INNER JOIN coordinadores c ON e.programa = c.programa
    LEFT JOIN asignaciones a ON e.exp = a.exp_alumno
    WHERE a.exp_alumno IS NULL
";

$Resultado = Ejecutar($Con, $SQL);

$SQLSinodos = "SELECT clave, nombre, a_paterno, a_materno FROM docentes"; // Consulta para
obtener los sinodos
$ResultadoSinodos = Ejecutar($Con, $SQLSinodos);
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Asignar Sinodo</title>
    <style>

        .asignar-button {
```

```
    font-size: 1.1rem;
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    padding: 0.7rem 0.9rem;
    background-color: #123773;
    color: white;
    border: none;
    cursor: pointer;
    border-radius: clamp(.4rem, .4vw, .4rem);
}

.asignar-button:hover {
    background-color: #1455bd;
}

.confirmar-icon {
    color: #123773;
    font-size: 1.5rem;
    padding: 0.5rem 0.9rem;
    background-color: #ffffff;
    border: none;
    cursor: pointer;
    border-radius: clamp(.4rem, .4vw, .4rem);
    border-bottom: 0.0625rem solid var(--secondary-color);
}

.confirmar-icon:hover {
    background-color: #cfcfcf;
}

/* Estilos para el modal */
.modal {
    display: none;
    position: fixed;
    z-index: 1;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.7);
    opacity: 0;
    transition: opacity 0.3s ease;
}
```

```
}

.modal.show {
    opacity: 1;
}

.modal-content {
    background-color: #fff;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%) scale(0.7);
    padding: 2rem;
    width: 50%;
    height: 80%;
    border-radius: 0.4rem;
    overflow-y: auto;
    padding-bottom: 0;
    opacity: 0;
    transition: all 0.3s ease;
}

.modal.show .modal-content {
    transform: translate(-50%, -50%) scale(1);
    opacity: 1;
}

.modal-table {
    max-width: 100%;
}

.modal-table {
    max-width: 100%;
}

input[type="checkbox"] {
    cursor: pointer;
    width: 1.2rem;
    height: 1.2rem;
}
```

```
.container-confirmar-button {  
    mask-image:  
        linear-gradient(to bottom, transparent, black 10%, black 100%);  
    mask-composite: intersect;  
    -webkit-mask-image:  
        linear-gradient(to bottom, transparent, black 10%, black 100%);  
    -webkit-mask-composite: source-in;  
    padding: 7vh 0;  
    display: flex;  
    justify-content: center;  
    position: sticky;  
    background-color: white;  
    bottom: 0; /* Mantendrá el botón en la parte inferior de su contenedor */  
    z-index: 10; /* Para asegurarse de que esté por encima de otros elementos como la  
    tabla */  
}  
  
.confirmar-button {  
    font-size: 1.3rem;  
    font-family: "Google Sans", Roboto, Arial, sans-serif;  
    padding: 0.7rem 3rem;  
    background-color: #123773;  
    color: white;  
    border: none;  
    cursor: pointer;  
    border-radius: clamp(.4rem, .4vw, .4rem);  
}  
  
.confirmar-button.disabled {  
    background-color: grey;  
    cursor: not-allowed;  
    opacity: 0.6;  
}  
  
.close {  
    position: absolute;  
    top: 0;  
    right: .6rem;  
    color: #aaa;  
    float: right;  
    font-size: 28px;
```

```
    font-weight: bold;
    cursor: pointer;
}

.close:hover,
.close:focus {
    color: black;
    text-decoration: none;
    cursor: pointer;
}

@media (max-width: 770px) {
    .modal-content {
        width: 80%;
    }
    input[type="checkbox"] {
        cursor: pointer;
        width: 1.5rem;
        height: 1.5rem;
    }
    .confirmar-icon {
        background-color: #118f1d;
        color: white;
        padding: 0.7rem 0.9rem;
        border: none;
        cursor: pointer;
    }

    .confirmar-icon::before {
        font-family: "Google Sans", Roboto, Arial, sans-serif;
        font-size: 1.1rem;
        font-weight: 600;
        content: 'Confirmar';
    }

    .confirmar-icon {
        font-size: 0;
    }
    .asignar-button {
        margin-left: auto; /* Empuja el botón hacia la derecha */
    }
}
```

```

    }

</style>
</head>

<body>

<div class="container-principal">
    <h3>Asignar sinodo:</h3>
    <div id="table-container">
        <table>
            <thead>
                <tr>
                    <th>Expediente</th>
                    <th>Nombre</th>
                    <th>Sinodo 1 <br> (Director)</th>
                    <th>Sinodo 2</th>
                    <th>Sinodo 3</th>
                    <th>Sinodo 4 <br> (Externo)</th>
                    <th>Confirmar</th>
                </tr>
            </thead>
            <tbody>
                <?php
                    if ($Resultado->num_rows > 0){
                        while ($Fila = $Resultado->fetch_assoc()){
                            $Nombre = $Fila["nombre"] . " " . $Fila["a_paterno"] . " " .
                            $Fila["a_materno"];
                            echo "<tr>";
                            echo "<td data-label='Expediente'>" . $Fila ["exp"] . "</td>";
                            echo "<td data-label='Nombre'>" . $Nombre . "</td>";

                            // Botones de asignar para cada sinodo
                            echo "<td data-label='Sinodo 1 (Director)'><button class='asignar-
button' onclick='openModal(this)'>Asignar</button><div class='sinodo-container'></div></td>";
                            echo "<td data-label='Sinodo 2'><button class='asignar-button'
onclick='openModal(this)'>Asignar</button><div class='sinodo-container'></div></td>";
                            echo "<td data-label='Sinodo 3'><button class='asignar-button'
onclick='openModal(this)'>Asignar</button><div class='sinodo-container'></div></td>";
                            echo "<td data-label='Sinodo 4 (Externo)'><button class='asignar-
button' onclick='openModal(this)'>Asignar</button><div class='sinodo-container'></div></td>";
                        }
                    }
                </tbody>
            </table>
        </div>
    </div>

```

```

        // Botón de confirmar
        echo "<td data-label='Confirmar'><button class='confirmar-icon'
onclick='confirmarAsignacion(\"" . $Fila['exp'] . "\")'>&#x2714;</button></td>";
        echo "</tr>";
    }
} else {
    echo "<tr><td colspan='7'>No se encontraron estudiantes</td></tr>";
}
Cerrar($Con);
?>
</tbody>
</table>
</div>
</div>

<!-- Modal --&gt;
&lt;div id="sinodoModal" class="modal"&gt;
    &lt;div class="modal-content"&gt;
        &lt;span class="close"&gt;&amp;times;&lt;/span&gt;
        &lt;h3&gt;Seleccionar sinodo&lt;/h3&gt;
        &lt;table class="modal-table"&gt;
            &lt;thead&gt;
                &lt;tr&gt;
                    &lt;th&gt;ID&lt;/th&gt;
                    &lt;th&gt;Nombre&lt;/th&gt;
                    &lt;th&gt;Seleccionar&lt;/th&gt;
                &lt;/tr&gt;
            &lt;/thead&gt;
            &lt;tbody&gt;
                &lt;?php
                if ($ResultadoSinodos-&gt;num_rows &gt; 0) {
                    while ($Sinodo = $ResultadoSinodos-&gt;fetch_assoc()) {
                        $NombreSin = $Sinodo["nombre"] . " " . $Sinodo["a_paterno"] . " " .
$Sinodo["a_materno"];
                        echo "&lt;tr&gt;";
                        echo "&lt;td data-label='ID'&gt;" . $Sinodo['clave'] . "&lt;/td&gt;";
                        echo "&lt;td data-label='Nombre'&gt;" . $NombreSin . "&lt;/td&gt;";
                        // Cambiar el value del checkbox a la clave del sinodo
                }
            }
        &lt;/tbody&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>

```

```

        echo "<td data-label='Seleccionar'><input type='checkbox'
class='sinodo-checkbox' value='".$Sinodo['clave']."' onclick='handleCheckbox(this, \"".
$NombreSin . "\")'></td>";
        echo "</tr>";
    }
}

?>

</tbody>
</table>
<div class="container-confirmar-button">
    <button class="confirmar-button" onclick="confirmSelection()">Confirmar</button>
</div>
</div>
</div>

<script>
let sinodosSeleccionadosPorEstudiante = {};// Objeto para almacenar sínodos seleccionados por
cada estudiante
let selectedSinodo = null;
let currentButton;
const confirmarButton = document.querySelector('.confirmar-button');

// Iniciar con el botón deshabilitado
confirmarButton.classList.add('disabled');
confirmarButton.disabled = true;

// Función para permitir solo un checkbox seleccionado a la vez
function handleCheckbox(checkbox, nombreSinodo) {
    let checkboxes = document.querySelectorAll('.sinodo-checkbox');

    // Si el sínodo seleccionado es el comodín (clave 0)
    if (checkbox.value === "0") {
        // Si está checkeado, permitir su selección sin desmarcar otros
        if (checkbox.checked) {
            selectedSinodo = checkbox.value;
            confirmarButton.classList.remove('disabled');
            confirmarButton.disabled = false;
            currentButton.sinodoNombre = nombreSinodo;
        } else {
            selectedSinodo = null;
            confirmarButton.classList.add('disabled');
        }
    }
}

```

```

        confirmarButton.disabled = true;
        currentButton.sinodoNombre = null;
    }
} else {
    // Para otros sínodos, mantener el comportamiento original de selección única
    checkboxes.forEach(cb => {
        if (cb !== checkbox && cb.value !== "0") {
            cb.checked = false;
        }
    });
}

if (checkbox.checked) {
    selectedSinodo = checkbox.value;
    confirmarButton.classList.remove('disabled');
    confirmarButton.disabled = false;
    currentButton.sinodoNombre = nombreSinodo;
} else {
    selectedSinodo = null;
    confirmarButton.classList.add('disabled');
    confirmarButton.disabled = true;
    currentButton.sinodoNombre = null;
}
}

// Función para abrir el modal
function openModal(button) {
    currentButton = button;
    const modal = document.getElementById("sinodoModal");
    const exp = button.closest('tr').querySelector('td:first-child').textContent;// Obtener el expediente de la fila actual

    modal.style.display = "block";
    // Forzar un reflow para que la transición funcione
    modal.offsetHeight;
    modal.classList.add('show');

    confirmarButton.classList.add('disabled');
    confirmarButton.disabled = true;

    let sinodosSeleccionados = sinodosSeleccionadosPorEstudiante[exp] || [];
}

```

```

let checkboxes = document.querySelectorAll('.sinodo-checkbox');
checkboxes.forEach(checkbox => {
    // Si el sónodo es el comodín (clave 0), siempre debe estar habilitado
    if (checkbox.value === "0") {
        checkbox.disabled = false;
    } else {
        // Para otros sónodos, bloquear solo si ya fue seleccionado para este estudiante
        if (sinodosSeleccionados.includes(checkbox.value)) {
            checkbox.disabled = true;
        } else {
            checkbox.disabled = false;
        }
    }
    checkbox.checked = false;
});

selectedSinodo = null;
}

// Función para confirmar la selección
function confirmSelection() {
    if (selectedSinodo && currentButton) {
        const exp = currentButton.closest('tr').querySelector('td:first-child').textContent;
        // Obtener el expediente de la fila actual
        let sinodoContainer = currentButton.nextElementSibling;

        if (sinodoContainer) {
            currentButton.style.display = 'none'; // Ocultar el botón de Asignar
            sinodoContainer.textContent = currentButton.sinodoNombre; // Mostrar el sónodo
            asignado (nombre)
                sinodoContainer.dataset.sinodoClave = selectedSinodo; // Almacenar la clave en un
            data attribute

            const modal = document.getElementById("sinodoModal");
            modal.classList.remove('show');
            setTimeout(() => {
                modal.style.display = "none";
            }, 300);
        }
    }
}

// Guardar el sónodo asignado en el objeto del estudiante actual

```

```

        if (!sinodosSeleccionadosPorEstudiante[exp]) {
            sinodosSeleccionadosPorEstudiante[exp] = [];
        }
        sinodosSeleccionadosPorEstudiante[exp].push(selectedSinodo); // Añadir sínodo al
estudiante
    }
} else {
    alert("Por favor selecciona un sínodo");
}
}

// Función para insertar en la base de datos cuando se confirma la asignación
function confirmarAsignacion(exp) {
    let sinodos = [];
    document.querySelectorAll('.sinodo-container').forEach(container => {
        if (container.dataset.sinodoClave) {
            sinodos.push(container.dataset.sinodoClave);
        }
    });

    if (sinodos.length === 4) {
        console.log(sinodos);
        const xhr = new XMLHttpRequest();
        xhr.open("POST", "../Coordinador/insertar_sinodos.php", true);
        xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
        xhr.onreadystatechange = function() {
            if (xhr.readyState === 4 && xhr.status === 200) {
                alert("Asignación hecha correctamente!"); // Mostrar respuesta del servidor
                // console.log(xhr.responseText);
                location.reload();
            }
        };
        xhr.send("exp=" + exp + "&sinodo1=" + sinodos[0] + "&sinodo2=" + sinodos[1] +
"&sinodo3=" + sinodos[2] + "&sinodo4=" + sinodos[3]);
    } else {
        alert("Debes asignar los 4 sinodos antes de confirmar");
    }
}

// Función para cerrar el modal
let closeModalButton = document.querySelector('.close');

```

```

closeModalButton.onclick = function() {
    const modal = document.getElementById("sinodoModal");
    modal.classList.remove('show');
    setTimeout(() => {
        modal.style.display = "none";
    }, 300);
}

// Cerrar el modal si se hace clic fuera del contenido
window.onclick = function(event) {
    let modal = document.getElementById("sinodoModal");
    if (event.target === modal) {
        modal.classList.remove('show');
        setTimeout(() => {
            modal.style.display = "none";
        }, 300);
    }
}
</script>

</body>
</html><?php
    include('.../../Header/MenuC.php');
?>

<?php
// Incluir el archivo de conexión
include '../../../../../Config/conexion.php';

// Conectar a la base de datos
$Con = Conectar();

// Consulta SQL para obtener los datos de los estudiantes
$SQL = "
    SELECT e.exp, e.nombre, e.a_paterno, e.a_materno
    FROM estudiantes e
    INNER JOIN coordinadores c ON e.programa = c.programa
    LEFT JOIN asignaciones a ON e.exp = a.exp_alumno
    WHERE a.exp_alumno IS NULL
";

```

```
$Resultado = Ejecutar($Con, $SQL);

$SQLSinodos = "SELECT clave, nombre, a_paterno, a_materno FROM docentes"; // Consulta para
obtener los sinodos
$ResultadoSinodos = Ejecutar($Con, $SQLSinodos);
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Asignar Sinodo</title>
    <style>
        /* Estilos aquí */
    </style>
</head>

<body>
    <!-- Contenido aquí -->
</body>

</html>
```

eliminar_asignacion.php

```
<?php
header('Content-Type: application/json');
include('../../../Config/conexion.php');

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['expediente'])) {
    $Con = Conectar();
    $expediente = $_POST['expediente'];

    try {
        $SQL_DELETE_ASIGNACION = "DELETE FROM asignaciones WHERE exp_alumno = ?";
        $stmt = mysqli_prepare($Con, $SQL_DELETE_ASIGNACION);
        mysqli_stmt_bind_param($stmt, 's', $expediente);
        mysqli_stmt_execute($stmt);

        if (mysqli_affected_rows($Con) > 0) {
            echo json_encode([
                'status' => 'success',
                'message' => 'Asignación eliminada exitosamente'
            ]);
        } else {
            throw new Exception('No se encontró la asignación');
        }
    } catch (Exception $e) {
        echo json_encode([
            'status' => 'error',
            'message' => 'Error al eliminar la asignación: ' . $e->getMessage()
        ]);
    }

    mysqli_close($Con);
} else {
    echo json_encode([
        'status' => 'error',
        'message' => 'Solicitud inválida'
    ]);
}
?>
```

EvaluacionesAgendadasC.php

```
<?php
    include('../Header/MenuC.php');
?>

<?php
include('.../Config/conexion.php');
$Con = Conectar();

$SQL = "
    SELECT ev.exp_alumno, e.nombre, e.a_paterno, e.a_materno, ev.fecha_evaluacion, ev.aula
    FROM evaluaciones ev
    JOIN estudiantes e ON ev.exp_alumno = e.exp;
";

$Res = Ejecutar($Con, $SQL);
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Evaluaciones Agendadas</title>
    <style>

        .eliminar-icon {
            color: #123773;
            font-size: 1rem;
            padding: 0.7rem 0.9rem 0.75rem;
            background-color: #ffffff;
            border: none;
            cursor: pointer;
            border-radius: clamp(.4rem, .4vw, .4rem);
            border-bottom: 0.0625rem solid var(--secondary-color);
        }

        .eliminar-icon:hover {
```

```
        background-color: #cfcfcf;
    }

    @media (max-width: 770px) {
        .eliminar-icon {
            background-color: red;
            color: white;
            padding: 0.7rem 0.9rem;
            border: none;
            cursor: pointer;
        }

        .eliminar-icon::before {
            font-family: "Google Sans", Roboto, Arial, sans-serif;
            font-size: 1.1rem;
            font-weight: 600;
            content: 'Confirmar';
        }

        .eliminar-icon {
            font-size: 0;
        }
    }
}

</style>
</head>

<body>
    <div class="container-principal">
        <h3>Evaluaciones Agendadas:</h3>
        <div id="table-container">
            <table>
                <thead>
                    <tr>
                        <th>Expediente</th>
                        <th>Nombre</th>
                        <th>Fecha</th>
                        <th>Hora</th>
                        <th>Aula</th>
                        <th>Eliminar</th>
                    </tr>
                </thead>
            </table>
        </div>
    </div>
</body>
```

```

        </thead>
        <tbody>
            <?php
            if ($Res->num_rows > 0){
                while($Fila = $Res->fetch_assoc()){
                    $NombreCom = $Fila["nombre"] . " " . $Fila["a_paterno"] . " " .
$Fila["a_materno"];
                    $exp = $Fila["exp_alumno"];
                    $fecha = date('Y-m-d', strtotime($Fila['fecha_evaluacion']));
                    $hora = date('H:i', strtotime($Fila['fecha_evaluacion']));
                    echo "<tr id='fila-$exp>" . $exp . "</tr>";
                    echo "<td data-label='Expediente'>" . $exp . "</td>";
                    echo "<td data-label='Nombre'>" . $NombreCom . "</td>";
                    echo "<td data-label='Fecha'>" . $fecha . "</td>";
                    echo "<td data-label='Hora'>" . $hora . "</td>";
                    echo "<td data-label='Aula'>" . $Fila['aula'] . "</td>";
                    echo "<td data-label='Eliminar'><button class='eliminar-icon'
onclick='eliminarEvaluacion(\"" . $exp . "\")'>X</button></td>";
                    echo "</tr>";
                }
            }else{
                echo "<tr><td colspan = '6'>No se encontraron evaluaciones agendadas </td></tr>";
            }
            Cerrar($Con);
        ?>
    </tbody>
</table>
</div>
</div>

</body>
<script>

    function eliminarEvaluacion(expediente) {
        // Confirmar eliminación
        if (confirm('¿Estás seguro de que quieres eliminar esta evaluación?')) {
            // Crear el objeto XMLHttpRequest
            var xhr = new XMLHttpRequest();

            // Configurar la solicitud
            xhr.open('POST', '../Acciones globales/eliminar_evaluacion.php', true);
            xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

```

```

// Definir lo que sucederá cuando la solicitud se complete
xhr.onload = function() {
    if (xhr.status === 200) {
        // Eliminar la fila de la tabla si la solicitud fue exitosa
        var fila = document.getElementById('fila-' + expediente);
        if (fila) {
            fila.remove();
        }
        alert('Evaluación eliminada exitosamente.');
    } else {
        alert('Hubo un error al eliminar la evaluación.');
    }
};

// Enviar la solicitud con el expediente del alumno
xhr.send('accion=eliminar&expediente=' + expediente);
}
}

</script>
</html>

```

insertar_sinodos.php

```

<?php
session_start();
// Incluir el archivo de conexión
include '../../../../../Config/conexion.php';

// Conectar a la base de datos
$Con = Conectar();

// Verificar si los datos han sido enviados por POST
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Obtener los datos enviados desde la función de JavaScript
    $exp_alumno = $_POST['exp']; // Cambié 'exp' a 'exp_alumno' para coincidir con tu campo
    $director = $_POST['sinodo1'];
    $sinodo2 = $_POST['sinodo2'];
    $sinodo3 = $_POST['sinodo3'];
}

```

```

$externo = $_POST['sinodo4']; // En el código JS 'sinodo4' se refiere al sínodo externo
$clave_coordinador = $_SESSION['id']; // Aquí debes reemplazarlo con el valor
correspondiente, si lo tienes en alguna parte de tu sistema

// Consulta SQL para insertar o actualizar los sinodos asignados en la tabla
'asignaciones'
$SQL = "INSERT INTO asignaciones (exp_alumno, director, sinodo2, sinodo3, externo,
clave_coordinador)
VALUES ('$exp_alumno', '$director', '$sinodo2', '$sinodo3', '$externo',
'$clave_coordinador')
ON DUPLICATE KEY UPDATE
director = '$director', sinodo2 = '$sinodo2', sinodo3 = '$sinodo3', externo =
'$externo', clave_coordinador = '$clave_coordinador'";
echo $SQL;
// Ejecutar la consulta
if (Ejecutar($Con, $SQL)) {
    // Si la inserción es exitosa
    echo "Asignación de sínodos guardada con éxito para el expediente: $exp_alumno.";
} else {
    // Si hay un error al ejecutar la consulta
    echo "Error al asignar los sínodos para el expediente: $exp_alumno.";
}

// Cerrar la conexión a la base de datos
Cerrar($Con);
} else {
    // Si la solicitud no fue enviada correctamente
    echo "Método de solicitud no válido.";
}
?>
```

RegistroDocentes.php

```
<?php
header('Content-Type: application/json');

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    // Obtener datos del formulario y limpiar las entradas
    $Clave = trim($_POST['clave']);
    $Nombre = trim($_POST['nombre']);
    $ApellidoP = trim($_POST['apellidoPaterno']);
    $ApellidoM = trim($_POST['apellidoMaterno']);
    $Status = trim($_POST['status']);

    // Quitar espacios al inicio y al final
    $Nombre=trim( $Nombre);
    $ApellidoM=trim($ApellidoM);
    $ApellidoP=trim($ApellidoP);

    // Convertir a mayúsculas y quitar acentos
    include("../Index/LimpiaCadenas.php");
    $Nombre = strtoupper(eliminar_acentos($Nombre));
    $ApellidoP = strtoupper(eliminar_acentos($ApellidoP));
    $ApellidoM = strtoupper(eliminar_acentos($ApellidoM));

    // Crear contraseña
    $Password = $ApellidoP . substr($Clave, -2);

    //Variable de nombre completo
    $NombreC = "$ApellidoP $ApellidoM $Nombre";

    // Conectar a la base de datos
    include("../Config/conexion.php");
    $Con = Conectar();
    if (!$Con) {
        die(json_encode(["status" => "error", "message" => "Error de conexión: " .
        mysqli_connect_error()]));}
    }

    // Verificar si el expediente ya existe
    $SQL1 = "SELECT * FROM docentes WHERE clave = ?";
```

```

$stmt1 = mysqli_prepare($Con, $SQL1);
mysqli_stmt_bind_param($stmt1, 's', $Clave);

if (!$stmt1->execute()) {
    echo json_encode([
        'status' => 'error',
        'message' => 'Error en la consulta: ' . mysqli_error($Con)
    ]);
    exit;
}

$result = mysqli_stmt_get_result($stmt1);

if (mysqli_num_rows($result) == 0) {
    // Insertar datos en la base de datos
    $SQL2 = "INSERT INTO docentes (clave, nombre, a_paterno, a_materno, status)
              VALUES (?, ?, ?, ?, ?)";
    $SQL3 = "INSERT INTO cuentas (id, contrasena, tipo)
              VALUES (?, ?, 'D')";

    $stmt2 = mysqli_prepare($Con, $SQL2);
    $stmt3 = mysqli_prepare($Con, $SQL3);

    if (!$stmt2 || !$stmt3) {
        echo json_encode([
            'status' => 'error',
            'message' => 'Error al preparar las consultas: ' . mysqli_error($Con)
        ]);
        exit;
    }

    mysqli_stmt_bind_param($stmt2, 'issss', $Clave, $Nombre, $ApellidoP, $ApellidoM,
    $status);
    mysqli_stmt_bind_param($stmt3, 'is', $Clave, $Password);

    if (mysqli_stmt_execute($stmt2) && mysqli_stmt_execute($stmt3)) {
        echo json_encode([
            'status' => 'success',
            'message' => "Cuenta registrada correctamente.\n\n".
                "Bienvenido $NombreC.\n" .
                "Tus credenciales de acceso son:\n" .
        ]);
    }
}

```

```
        "Usuario: $Clave\n" .
        "Password: $Password\n\n" .
        "Recuerda guardar tus credenciales para acceder al sistema.",
    'copyText' => "Tus credenciales de acceso son:\n" .
        "Usuario: $Clave\n" .
        "Password: $Password"
    ]);
} else {
    echo json_encode([
        'status' => 'error',
        'message' => "Error al registrar la cuenta: " . mysqli_error($Con)
    ]);
}
} else {
    echo json_encode([
        'status' => 'exists',
        'message' => "La cuenta ya ha sido registrada anteriormente.\n\n" .
            "Para recuperar tus credenciales de acceso, envía un correo a:
francisco.javier.paulin@uaq.mx"
    ]);
}

// Cerrar la conexión
mysqli_close($Con);
}
?>
```

SinodosAsignados.php

```
<?php
include('../Header/MenuC.php');
if(session_status() === PHP_SESSION_NONE){
    session_start();
}

include('../../../Config/conexion.php');
$Con = Conectar();

// Consulta para obtener los sinodos asignados
$SQL = "
SELECT
    a.exp_alumno,
    e.nombre as nombre_alumno,
    e.a_paterno as ap_alumno,
    e.a_materno as am_alumno,
    d1.nombre as nombre_director,
    d1.a_paterno as ap_director,
    d2.nombre as nombre_sinodo2,
    d2.a_paterno as ap_sinodo2,
    d3.nombre as nombre_sinodo3,
    d3.a_paterno as ap_sinodo3,
    d4.nombre as nombre_externo,
    d4.a_paterno as ap_externo
FROM asignaciones a
INNER JOIN estudiantes e ON a.exp_alumno = e.exp
LEFT JOIN docentes d1 ON a.director = d1.clave
LEFT JOIN docentes d2 ON a.sinodo2 = d2.clave
LEFT JOIN docentes d3 ON a.sinodo3 = d3.clave
LEFT JOIN docentes d4 ON a.externo = d4.clave
ORDER BY e.a_paterno, e.a_materno, e.nombre
";

$Res = Ejecutar($Con, $SQL);
?>

<!DOCTYPE html>
<html lang="en">
```

```
<head>

    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Sinodos Asignados</title>
    <style>

        .eliminar-icon {
            color: #123773;
            font-size: 1rem;
            padding: 0.7rem 0.9rem 0.75rem;
            background-color: #ffffff;
            border: none;
            cursor: pointer;
            border-radius: 0.4rem;
            border-bottom: 0.0625rem solid var(--secondary-color);
        }

        .eliminar-icon:hover {
            background-color: #cfcfcf;
        }

        @media (max-width: 770px) {
            .eliminar-icon {
                background-color: red;
                color: white;
                padding: 0.7rem 0.9rem;
                border: none;
                cursor: pointer;
            }
        }

        .eliminar-icon::before {
            font-family: "Google Sans", Roboto, Arial, sans-serif;
            font-size: 1.1rem;
            font-weight: 600;
            content: 'Eliminar';
        }

        .eliminar-icon {
            font-size: 0;
        }
    </style>
```

```

        </style>
    </head>

<body>
    <div class="container-principal">
        <h3>Sinodos Asignados:</h3>
        <div id="table-container">
            <table>
                <thead>
                    <tr>
                        <th>Alumno</th>
                        <th>Sinodo Director</th>
                        <th>Sinodo 2</th>
                        <th>Sinodo 3</th>
                        <th>Sinodo Externo</th>
                        <th>Eliminar</th>
                    </tr>
                </thead>
                <tbody>
                    <?php
                    if ($Res->num_rows > 0){
                        while($Fila = $Res->fetch_assoc()){
                            $exp = $Fila["exp_alumno"];
                            echo "<tr id='fila-" . $exp . "'>";
                            echo "<td data-label='Alumno'>" . $Fila['ap_alumno'] . " " .
$Fila['am_alumno'] . " " . $Fila['nombre_alumno'] . "</td>";
                            echo "<td data-label='Sinodo Director'>" . $Fila['ap_director'] .
" " . $Fila['nombre_director'] . "</td>";
                            echo "<td data-label='Sinodo 2'>" . $Fila['ap_sinodo2'] . " " .
$Fila['nombre_sinodo2'] . "</td>";
                            echo "<td data-label='Sinodo 3'>" . $Fila['ap_sinodo3'] . " " .
$Fila['nombre_sinodo3'] . "</td>";
                            echo "<td data-label='Sinodo Externo'>" . $Fila['ap_externo'] . " " .
$Fila['nombre_externo'] . "</td>";
                            echo "<td data-label='Eliminar'><button class='eliminar-icon' onclick='eliminarAsignacion(\"" . $exp . "\")'>X</button></td>";
                            echo "</tr>";
                        }
                    } else {
                        echo "<tr><td colspan='6'>No hay sinodos asignados</td></tr>";
                    }
                </tbody>
            </table>
        </div>
    </div>

```

```

        Cerrar($Con);
    ?>
</tbody>
</table>
</div>
</div>

<script>
    function eliminarAsignacion(expediente) {
        if (confirm('¿Estás seguro de que quieres eliminar esta asignación de sinodos?'))
{
            var xhr = new XMLHttpRequest();
            xhr.open('POST', 'eliminar_asignacion.php', true);
            xhr.setRequestHeader('Content-type', 'application/x-www-form-urlencoded');

            xhr.onload = function() {
                try {
                    var response = JSON.parse(xhr.responseText);
                    if (response.status === 'success') {
                        var fila = document.getElementById('fila-' + expediente);
                        if (fila) {
                            fila.remove();
                        }
                        alert('Asignación eliminada exitosamente.');
                        location.reload();
                    } else {
                        alert('Error: ' + response.message);
                    }
                } catch (e) {
                    console.error('Error parsing response:', xhr.responseText);
                    alert('Error al procesar la respuesta del servidor');
                }
            };
            xhr.onerror = function() {
                alert('Error de conexión al servidor');
            };

            xhr.send('expediente=' + expediente);
        }
    }

```

```
</script>
</body>
</html>
```

actualizar_detalle_evaluaciones.php

```
<?php
// Segundo archivo (actualizar_detalle_evaluaciones.php)
session_start();
include '../../Config/conexion.php';
$Con = Conectar();

if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['expediente'])) {
    $expediente = $_POST['expediente'];
    $periodo = isset($_POST['periodo']) ? $_POST['periodo'] : '';
    $esDirector = isset($_POST['esDirector']) && $_POST['esDirector'] === 'true';

    if (isset($_SESSION['id'])) {
        $id_sinodo = $_SESSION['id'];

        // Verificar si ya existe un registro
        $checkSQL = "SELECT id_evaluacion FROM detalle_evaluaciones
                     WHERE id_sinodo = ? AND
                           id_evaluacion = (SELECT id FROM evaluaciones WHERE exp_alumno = ?)";

        $stmt = $Con->prepare($checkSQL);
        $stmt->bind_param("ss", $id_sinodo, $expediente);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($esDirector) {
            // Manejo para director (sin calificación)
            if ($result->num_rows > 0) {
                $SQL = "UPDATE detalle_evaluaciones
                        SET d_observacion1 = ?,
                            d_observacion2 = ?,
                            d_observacion3 = ?,
                            observacion = ?,
                            periodo = ?";

                $stmt = $Con->prepare($SQL);
                $stmt->bind_param("ssssss", $d_observacion1, $d_observacion2, $d_observacion3, $observacion, $periodo);
                $stmt->execute();
            }
        }
    }
}
```

```

        WHERE id_sinodo = ? AND
        id_evaluacion = (SELECT id FROM evaluaciones WHERE exp_alumno = ?)";

$stmt = $Con->prepare($SQL);
$d_obs1 = $_POST[ 'd_observacion1' ] ?? '';
$d_obs2 = $_POST[ 'd_observacion2' ] ?? '';
$d_obs3 = $_POST[ 'd_observacion3' ] ?? '';
$observacion = $_POST[ 'observacion' ] ?? '';
$stmt->bind_param(
    "sssssss",
    $d_obs1,
    $d_obs2,
    $d_obs3,
    $observacion,
    $periodo,
    $id_sinodo,
    $expediente
);
} else {
    $SQL = "INSERT INTO detalle_evaluaciones
        (id_sinodo, id_evaluacion, d_observacion1, d_observacion2,
d_observacion3, observacion, periodo)
        SELECT ?, id, ?, ?, ?, ?, ?
        FROM evaluaciones
        WHERE exp_alumno = ?";

$stmt = $Con->prepare($SQL);
$d_obs1 = $_POST[ 'd_observacion1' ] ?? '';
$d_obs2 = $_POST[ 'd_observacion2' ] ?? '';
$d_obs3 = $_POST[ 'd_observacion3' ] ?? '';
$observacion = $_POST[ 'observacion' ] ?? '';
$stmt->bind_param(
    "sssssss",
    $id_sinodo,
    $d_obs1,
    $d_obs2,
    $d_obs3,
    $observacion,
    $periodo,
    $expediente
);
}

```

```

        }

    } else {
        // Manejo para no director (con calificación)
        $calificacion = isset($_POST['calificacion']) ? $_POST['calificacion'] : 0;

        if ($result->num_rows > 0) {
            $SQL = "UPDATE detalle_evaluaciones
                SET calificacion = ?,
                    observacion = ?,
                    periodo = ?
                WHERE id_sinodo = ? AND
                    id_evaluacion = (SELECT id FROM evaluaciones WHERE exp_alumno = ?)";

            $stmt = $Con->prepare($SQL);
            $observacion = $_POST['observacion'] ?? '';
            $stmt->bind_param(
                "dssss",
                $calificacion,
                $observacion,
                $periodo,
                $id_sinodo,
                $expediente
            );
        } else {
            $SQL = "INSERT INTO detalle_evaluaciones
                (id_sinodo, id_evaluacion, calificacion, observacion, periodo)
                SELECT ?, id, ?, ?, ?
                FROM evaluaciones
                WHERE exp_alumno = ?";

            $stmt = $Con->prepare($SQL);
            $observacion = $_POST['observacion'] ?? '';
            $stmt->bind_param(
                "sdsss",
                $id_sinodo,
                $calificacion,
                $observacion,
                $periodo,
                $expediente
            );
        }
    }
}

```

```
}

if ($stmt->execute()) {
    echo "success";
} else {
    echo "Error: " . $stmt->error;
}

$stmt->close();
} else {
    echo "Error: ID de sínodo no encontrado en la sesión";
}

Cerrar($Con);
} else {
    echo "Error: Datos incompletos";
}
?>
```

EvaluacionesPendientes.php

```
<?php
include('../Header/MenuD.php');

// Verificar si ya hay una sesión activa
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

// Incluir el archivo de conexión
include(' ../../Config/conexion.php');

// Conectar a la base de datos
$Con = Conectar();

$id_sinodo = $_SESSION['id'];

// Consulta principal para obtener evaluaciones pendientes
$SQL = "
    SELECT DISTINCT
        a.exp_alumno,
        e.nombre,
        e.a_paterno,
        e.a_materno,
        ev.aula,
        ev.fecha_evaluacion,
        CASE
            WHEN a.director = $id_sinodo THEN 1
            ELSE 0
        END as es_director
    FROM
        asignaciones a
    LEFT JOIN
        estudiantes e ON a.exp_alumno = e.exp
    LEFT JOIN
        evaluaciones ev ON a.exp_alumno = ev.exp_alumno
    LEFT JOIN
        detalle_evaluaciones de ON (
            ev.id = de.id_evaluacion AND
```

```

        de.id_sinodo = $id_sinodo
    )
WHERE
    (a.director = $id_sinodo OR a.sinodo2 = $id_sinodo OR
     a.sinodo3 = $id_sinodo OR a.externo = $id_sinodo)
    AND (de.id_evaluacion IS NULL OR ((de.calificacion IS NULL OR de.calificacion = 0) AND
de.observacion IS NULL))
;

$Resultado = Ejecutar($Con, $SQL);
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Evaluaciones Pendientes</title>
    <style>
.inputs {
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    height: 2vh;
    border-bottom: 1px solid #636363;
    outline: none;
    font-size: 1rem;
    font-weight: 500;
    color: var(--text-color);
    border: 1px solid #ccc;
    padding: 0.8rem 0.9rem;
    border-radius: clamp(.4rem, .4vw, .4rem);
}

.observaciones {
    font-size: 1rem;
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    padding: 0.8rem 0.9rem;
    background-color: #ffffff;
    border: none;
    cursor: pointer;
}

```

```
    color: var(--text-color);
    border-radius: clamp(.4rem, .4vw, .4rem);
    border-bottom: 0.0625rem solid var(--secondary-color);
}

.observaciones:hover {
    background-color: #cfcfcf;
}

.confirmar-icon {
    color: #ffffff;
    font-size: 1.5rem;
    padding: 0.5rem 0.9rem;
    background-color: #123773;
    border: none;
    cursor: pointer;
    border-radius: clamp(.4rem, .4vw, .4rem);
    border-bottom: 0.0625rem solid var(--secondary-color);
}

.confirmar-icon:disabled {
    cursor: not-allowed;
}

.confirmar-icon:hover {
    background-color: #1455bd;
}

.confirmar-icon:disabled:hover {
    background-color: #123773;
}

.modal {
    display: none;
    position: fixed;
    z-index: 1;
    left: 0;
    top: 0;
    width: 100%;
    height: 100%;
    background-color: rgba(0, 0, 0, 0.7);
```

```
    opacity: 0;
    transition: opacity 0.3s ease;
}

.modal.show {
    opacity: 1;
}

.modal-content {
    background-color: #fff;
    position: absolute;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%) scale(0.7);
    padding: 2rem;
    border-radius: 0.4rem;
    overflow-y: auto;
    width: 100%;
    max-width: 35%;
    max-height: 70%;
    overflow-x: hidden;
    opacity: 0;
    transition: all 0.3s ease;
}

.modal.show .modal-content {
    transform: translate(-50%, -50%) scale(1);
    opacity: 1;
}

.close {
    position: absolute;
    top: 0;
    right: .6rem;
    color: #aaa;
    float: right;
    font-size: 28px;
    font-weight: bold;
    cursor: pointer;
}
```

```
.close:hover,  
.close:focus {  
    color: black;  
    text-decoration: none;  
    cursor: pointer;  
}  
  
.label-observaciones {  
    letter-spacing: .01785714em;  
    font-family: system-ui;  
    font-weight: 600;  
    font-size: 1.5rem;  
    color: var(--text-color);  
}  
  
.observacion-input {  
    color: #555;  
    font-family: "Google Sans", Roboto, Arial, sans-serif;  
    font-size: 1rem;  
    font-weight: 500;  
    width: 98%;  
    margin: .5rem 0 1rem;  
    padding: 0.5rem;  
    border: 1px solid #ddd;  
    border-radius: clamp(.4rem, .4vw, .4rem);  
    resize: none;  
}  
  
.confirmar-button {  
    display: flex;  
    font-size: 1.3rem;  
    font-family: "Google Sans", Roboto, Arial, sans-serif;  
    padding: 0.7rem 3rem;  
    background-color: #123773;  
    color: white;  
    border: none;  
    cursor: pointer;  
    border-radius: clamp(.4rem, .4vw, .4rem);  
    margin: auto;  
}
```

```
.confirmar-button.disabled {  
    background-color: grey;  
    cursor: not-allowed;  
    opacity: 0.6;  
}  
  
. @media (max-width: 770px) {  
    .inputs {  
        width: 16vw;  
    }  
  
    .modal-content {  
        max-width: 80%;  
    }  
  
    .observacion-input {  
        height: 6rem;  
    }  
  
    .observaciones {  
        color: white;  
        width: 20vw;  
        background-color: #123773;  
    }  
  
    .confirmar-icon {  
        background-color: #118f1d;  
        color: white;  
        padding: 0.7rem 0.9rem;  
        border: none;  
        cursor: pointer;  
    }  
  
    .confirmar-icon::before {  
        font-family: "Google Sans", Roboto, Arial, sans-serif;  
        font-size: 1.1rem;  
        font-weight: 600;  
        content: 'Confirmar';  
    }  
  
    .confirmar-icon {
```

```

        font-size: 0;
    }
}

</style>
</head>

<body>
<div class="container-principal">
    <h3>Evaluaciones pendientes:</h3>
    <div id="table-container">
        <table>
            <thead>
                <tr>
                    <th>Expediente</th>
                    <th>Nombre</th>
                    <th>Fecha</th>
                    <th>Aula</th>
                    <th>Entregable</th>
                    <th>Calificación</th>
                    <th>Observaciones</th>
                    <th>Confirmar</th>
                </tr>
            </thead>
            <tbody>
                <?php
                    if ($Resultado->num_rows > 0) {
                        while ($Fila = $Resultado->fetch_assoc()) {
                            $Expediente = $Fila['exp_alumno'];
                            $esDirector = $Fila['es_director'] == 1;

                            // Obtener lista de archivos en la carpeta de entregables
                            $entregables = [];
                            $dir = "../../Entregables/$Expediente/";
                            if (is_dir($dir)) {
                                $files = scandir($dir);
                                foreach ($files as $file) {
                                    if ($file != '.' && $file != '..') {
                                        $entregables[] = "<a href='$dir$file'
target='_blank'>$file</a>";
                                    }
                                }
                            }
                        }
                    }
                </tbody>
            </table>
        </div>
    </div>
</body>

```

```

        }

        // Generar contenido para la columna "Entregable"
        $entregablesContent = empty($entregables) ? "No disponible" :
        implode(", ", $entregables);

        // Generar nombre completo
        $Nombre = $Fila["nombre"] . " " . $Fila["a_paterno"] . " " .
        $Fila["a_materno"];

        // Separar fecha y hora
        $Fecha = $Fila["fecha_evaluacion"];
        $FechaSola = !empty($Fecha) ? date('Y-m-d', strtotime($Fecha)) :
        "Pendiente";

        if ((int)substr($FechaSola, 5, 2) < 6) {
            $Periodo = substr($FechaSola, 0, 4) . "-" . "2";
        } else {
            $Periodo = substr($FechaSola, 0, 4) . "-" . "1";
        }

        echo "<tr data-expediente='" . $Expediente . "'>";
        echo "<td data-label='Expediente'" . $Expediente . "</td>";
        echo "<td data-label='Nombre'" . $Nombre . "</td>";
        echo "<td data-label='Fecha'" . $FechaSola . "</td>";
        echo "<td data-label='Aula'" . (!empty($Fila["aula"])) ?
        $Fila["aula"] : "Pendiente" . "</td>";
        echo "<td data-label='Entregable'" . $entregablesContent .
        "</td>";

        // Modificar la visualización de calificación según el rol
        echo "<td data-label='Calificación'" . $Expediente . "</td>";
        if ($esDirector) {
            echo "No assignable";
        } else {
            echo "<input type='number' class='inputs' name='calificacion_"
            . $Expediente . "'"
            . "id='calificacion_" . $Expediente . "' step='0.01'"
            . "min='0' max='10'"
            . "placeholder='Calificación... required"
            . "onchange='checkFields(\"" . $Expediente . "\")'" . "
```

```

                oninput='limitDigits(this, 4)">';
            }
            echo "</td>";

            echo "<td data-label='Observaciones'>
                    <button class='observaciones' onclick='abrirModal(\"".
$Expediente . "\", \" . ($esDirector ? 'true' : 'false') . \")'>
                        Agregar
                    </button>
                    <input type='hidden' id='observacion_" . $Expediente . "'"
value=''
                </td>";
            echo "<td data-label='Acción'>
                    <button class='confirmar-icon' id='btn_" . $Expediente . "'"
onclick='actualizarEvaluacion(\"\" . $Expediente . \"")'
disabled>&#x2714;</button>
                </td>";
            echo "</tr>";
        }
    } else {
        echo "<tr><td colspan='8'>No se encontraron evaluaciones
pendientes</td></tr>";
    }
}

Cerrar($Con);
?>
</tbody>
</table>
</div>
</div>

<div id="modalObservaciones" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        <h3>Observaciones y recomendaciones</h3>
        <hr class="x-component x-component-default" style="border-top: 0; border-bottom:
0.05rem solid #196ad3; margin: auto; width: 100%; margin-bottom: 2rem;">
        <div id="observacionesContent">
            <!-- El contenido se llenará dinámicamente -->
        </div>
        <button onclick="guardarObservaciones()" class="confirmar-button">Guardar</button>
    </div>
</div>
```

```

        </div>
    </div>

<script>
    // Iniciar con el botón guardar del modal deshabilitado
    /* const confirmarButton = document.querySelector('.confirmar-button');
    confirmarButton.classList.add('disabled');
    confirmarButton.disabled = true; */

    function actualizarEvaluacion(expediente) {
        const fila = document.querySelector(`tr[data-expediente="${expediente}"]`);
        const fecha = fila.querySelector('td:nth-child(3)').innerText;
        const calificacionInput = document.getElementById('calificacion_' + expediente);
        const observacionInput = document.getElementById('observacion_' + expediente);
        const esDirector = !calificacionInput;

        // Calcular el periodo basado en la fecha
        let periodo = '';
        if (fecha !== "Pendiente") {
            const mes = parseInt(fecha.split('-')[1]);
            const año = fecha.split('-')[0];
            periodo = año + "-" + (mes < 6 ? "2" : "1");
        }

        // Preparar los datos para enviar
        const formData = new FormData();
        formData.append('expediente', expediente);
        formData.append('periodo', periodo);
        formData.append('esDirector', esDirector);

        if (esDirector) {
            // Para director, separar las observaciones
            const observaciones = observacionInput.value.split('|');
            formData.append('d_observacion1', observaciones[0] || '');
            formData.append('d_observacion2', observaciones[1] || '');
            formData.append('d_observacion3', observaciones[2] || '');
            formData.append('observacion', observaciones[3] || '');
        } else {
            // Para sínodo
            formData.append('calificacion', calificacionInput.value);
            formData.append('observacion', observacionInput.value);
        }
    }
</script>

```

```

        }

        // Enviar la solicitud al servidor
        fetch('actualizar_detalle_evaluaciones.php', {
            method: 'POST',
            body: formData
        })
        .then(response => response.text())
        .then(result => {
            if (result === 'success') {
                alert('Evaluación actualizada correctamente');
                // Recargar la página para actualizar la lista
                location.reload();
            } else {
                alert('Error al actualizar la evaluación: ' + result);
            }
        })
        .catch(error => {
            console.error('Error:', error);
            alert('Error al procesar la solicitud');
        });
    }
    let expedienteActual = '';
    let esDirectorModal = false;

    function checkFields(expediente) {
        const btn = document.getElementById('btn_' + expediente);
        const fila = document.querySelector(`tr[data-expediente="${expediente}"]`);
        const fecha = fila.querySelector('td:nth-child(3)').innerText;
        const aula = fila.querySelector('td:nth-child(4)').innerText;
        const calificacionInput = document.getElementById('calificacion_' + expediente);
        const observacionInput = document.getElementById('observacion_' + expediente);

        // Verificar si es director para este expediente específico
        const esDirector = !calificacionInput; // Si no hay input de calificación, es
        director

        let isValid = false;

        if (esDirector) {
            // Validación para director: todas las observaciones deben estar llenas

```

```

        const observaciones = observacionInput.value.split('|');
        isValid = observaciones.length === 4 &&
            observaciones.every(obs => obs.trim() !== '') &&
            fecha !== "Pendiente" &&
            aula !== "Pendiente";
    } else {
        // Validación para sínodo: calificación y observación requeridas
        isValid = calificacionInput?.value &&
            observacionInput?.value &&
            fecha !== "Pendiente" &&
            aula !== "Pendiente";
    }

    btn.disabled = !isValid;
}

function limitDigits(input, maxDigits) {
    if (input.value.length > maxDigits) {
        input.value = input.value.slice(0, maxDigits);
    }
}

function abrirModal(expediente, esDirector) {
    expedienteActual = expediente;
    esDirectorModal = esDirector;
    const modal = document.getElementById('modalObservaciones');
    const contenido = document.getElementById('observacionesContent');

    // Recuperar observaciones existentes
    const existingObservations = document.getElementById('observacion_' +
expediente)?.value || '';
    contenido.innerHTML = '';

    if (esDirector) {
        // Formato para director
        const obsSections = existingObservations.split('|');
        contenido.innerHTML = `

<div>
    <label class='label-observaciones'>Sobre Avance gradual:</label>
    <textarea class="observacion-input" id="obs1_${expediente}" rows="5">

```

```

        onchange="checkFields('${expediente}')">${obsSections[0]} ||
''}</textarea>
      </div>
      <div>
        <label class='label-observaciones'>Sobre entregable y resultados
esperados:</label>
        <textarea class="observacion-input" id="obs2_${expediente}" rows="5"
onchange="checkFields('${expediente}')">${obsSections[1]} ||
''}</textarea>
      </div>
      <div>
        <label class='label-observaciones'>Sobre el Avance del
Proyecto:</label>
        <textarea class="observacion-input" id="obs3_${expediente}" rows="5"
onchange="checkFields('${expediente}')">${obsSections[2]} ||
''}</textarea>
      </div>
      <div>
        <label class='label-observaciones'>Comentario Individual:</label>
        <textarea class="observacion-input" id="obs4_${expediente}" rows="5"
onchange="checkFields('${expediente}')">${obsSections[3]} ||
''}</textarea>
      </div>`;
    } else {
      // Formato para sínodo
      contenido.innerHTML =
        <div>
          <textarea class="observacion-input" id="obs_${expediente}" rows="5"
onchange="checkFields('${expediente}')"
required>${existingObservations}</textarea>
        </div>`;
    }

    modal.style.display = "block";
    // Forzar un reflow para que la transición funcione
    modal.offsetHeight;
    modal.classList.add('show');
  }

  function guardarObservaciones() {

```

```

        const observacionInput = document.getElementById('observacion_' +
expedienteActual);

        if (esDirectorModal) {
            // Guardar observaciones de director
            const obs1 = document.getElementById(`obs1_${expedienteActual}`).value || '';
            const obs2 = document.getElementById(`obs2_${expedienteActual}`).value || '';
            const obs3 = document.getElementById(`obs3_${expedienteActual}`).value || '';
            const obs4 = document.getElementById(`obs4_${expedienteActual}`).value || '';

            observacionInput.value = [obs1, obs2, obs3, obs4].join('|');
        } else {
            // Guardar observación de sínodo
            const obsInput = document.getElementById(`obs_${expedienteActual}`);
            if (obsInput) {
                observacionInput.value = obsInput.value;
            }
        }

        checkFields(expedienteActual);
        cerrarModal();
    }

    function cerrarModal() {
        const modal = document.getElementById('modalObservaciones');
        modal.classList.remove('show');
        // Esperar a que termine la animación antes de ocultar el modal
        setTimeout(() => {
            modal.style.display = "none";
        }, 300); // Este tiempo debe coincidir con la duración de la transición en CSS
    }

    // Event listeners
    document.querySelector('.close').onclick = cerrarModal;

    // Cerrar modal al hacer clic fuera
    window.onclick = function(event) {
        const modal = document.getElementById('modalObservaciones');
        if (event.target === modal) {
            cerrarModal();
        }
    }

```

```
    };
</script>
</body>

</html>
```

HistorialEvaluaciones.php

```
<?php
include('../Header/MenuD.php');
// Verificar si ya hay una sesión activa
if (session_status() === PHP_SESSION_NONE) {
    session_start();
}

// Incluir el archivo de conexión
include '../../../../../Config/conexion.php';

// Conectar a la base de datos
$Con = Conectar();

$clave_coordinador = $_SESSION['id'];
$SQL = "
SELECT DISTINCT
    a.exp_alumno,
    e.nombre,
    e.a_paterno,
    e.a_materno,
    ev.aula,
    ev.fecha_evaluacion,
    de.calificacion,
    de.observacion,
    de.periodo
FROM
    asignaciones a
LEFT JOIN
    estudiantes e ON a.exp_alumno = e.exp
LEFT JOIN
    evaluaciones ev ON a.exp_alumno = ev.exp_alumno
```

```

INNER JOIN
    detalle_evaluaciones de ON (
        ev.id = de.id_evaluacion AND
        de.id_sinodo = $clave_coordinador
    )
WHERE
    (a.director = $clave_coordinador OR a.sinodo2 = $clave_coordinador OR
    a.sinodo3 = $clave_coordinador OR a.externo = $clave_coordinador)
    AND de.calificacion != 0
    AND de.calificacion IS NOT NULL
    AND de.observacion IS NOT NULL
;

$SQL2 = "SELECT DISTINCT periodo FROM detalle_evaluaciones WHERE periodo <> '' AND id_sinodo =
'$clave_coordinador' " ;
$Resultado = Ejecutar($Con, $SQL);
$Periodos = Ejecutar($Con, $SQL2);
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/tablas.css">
    <title>Historial de Evaluaciones</title>
</head>
<style>

.buscar {
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    border-bottom: 1px solid #636363;
    outline: none;
    font-size: 0.8rem;
    font-weight: 500;
    color: var(--text-color);
    border: 1px solid #ccc;
    padding: 0.5rem;
    border-radius: clamp(.4rem, .4vw, .4rem);
    width: 7vw;
}

```

```
}

/* Estilo placeholder en select */
.buscar.periodo {
    color: #757575; /* Color del texto del placeholder */
}

.buscar.periodo option {
    color: var(--text-color); /* Color del texto de las opciones */
}

.buscar.periodo:focus {
    outline: none;
}

.buscar.periodo option:first-child {
    display: none; /* Ocultar el primer option si es el placeholder */
}

.ver-observacion {
    font-size: 1rem;
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    padding: 0.8rem 0.9rem;
    background-color: #ffffff;
    border: none;
    cursor: pointer;
    color: var(--text-color);
    border-radius: clamp(.4rem, .4vw, .4rem);
    border-bottom: 0.0625rem solid var(--secondary-color);
}

.ver-observacion:hover {
    background-color: #cfcfcf;
}

@media (max-width: 770px) {
    .ver-observacion {
        color: white;
        background-color: #123773;
    }
}
```

```
.modal {  
    display: none;  
    position: fixed;  
    z-index: 1;  
    left: 0;  
    top: 0;  
    width: 100%;  
    height: 100%;  
    background-color: rgba(0, 0, 0, 0.7);  
    opacity: 0;  
    transition: opacity 0.3s ease;  
}  
  
.modal.show {  
    opacity: 1;  
}  
  
.modal-content {  
    background-color: #fff;  
    position: absolute;  
    top: 50%;  
    left: 50%;  
    transform: translate(-50%, -50%) scale(0.7);  
    padding: 2rem;  
    border-radius: 0.4rem;  
    width: 80%;  
    max-width: 500px;  
    opacity: 0;  
    transition: all 0.3s ease;  
}  
  
.modal.show .modal-content {  
    transform: translate(-50%, -50%) scale(1);  
    opacity: 1;  
}  
  
.close {  
    position: absolute;  
    top: 0;  
    right: .6rem;
```

```
        color: #aaa;
        float: right;
        font-size: 28px;
        font-weight: bold;
        cursor: pointer;
    }

.close:hover,
.close:focus {
    color: black;
    text-decoration: none;
    cursor: pointer;
}

#observacionContent {
    font-family: "Google Sans", Roboto, Arial, sans-serif;
    font-size: 1rem;
    color: var(--text-color);
    line-height: 1.5;
    margin-top: 1rem;
}

```

</style>

```
<body>
<div class="container-principal">
    <h3>Historial de evaluaciones:</h3>
    <div id="table-container">
        <table>
            <thead>
                <tr>
                    <th>Expediente</th>
                    <th>
                        Nombre
                        <br>
                        <input class="buscar nombre" type="text" id="search-name"
placeholder="Buscar..."/>
                    </th>
                    <th>Fecha</th>
                    <th>
                        Periodo
                        <br>

```

```

<select class="buscar periodo" id="search-periodo">
    <option value="" disabled selected hidden>Seleccionar...</option>
    <?php
        if ($Periodos->num_rows > 0) {
            while ($row = $Periodos->fetch_assoc()) {
                echo "<option value='" . htmlspecialchars($row['periodo']) .
. "'>" . htmlspecialchars($row['periodo']) . "</option>";
            }
        } else {
            echo "<option value='>No se encontraron periodos</option>";
        }
    ?>
</select>
</th>
<th>Calificación</th>
<th>Observaciones</th>

</tr>
</thead>
<tbody id="table-body">
    <?php
        if ($Resultado->num_rows > 0){
            while ($Fila = $Resultado->fetch_assoc()){
                $Nombre = $Fila["nombre"] . " " . $Fila["a_paterno"] . " " .
$Fila["a_materno"];
                echo "<tr data-expediente='".$Fila['exp_alumno']."' data-nombre='"
. $Nombre . "' data-periodo='".$Fila['periodo']."'>";
                echo "<td data-label='Expediente'>" . $Fila ["exp_alumno"] . "</td>";
                echo "<td data-label='Nombre'>" . $Nombre . "</td>";
                $Fecha = $Fila["fecha_evaluacion"];
                $FechaSola = !empty($Fecha) ? date('Y-m-d', strtotime($Fecha)) :
"Pendiente";
                echo "<td data-label='Fecha'>" . $FechaSola . "</td>";
                echo "<td data-label='Periodo'>" . $Fila["periodo"] . "</td>";
                echo "<td data-label='Calificación'>" . $Fila["calificacion"] .
"</td>";
                echo "<td data-label='Observaciones'><button class='ver-observacion'
data-observacion='".$Fila["observacion"]." '>Ver</button></td>";
                echo "</tr>";
            }
        } else {

```

```

        echo "<tr><td colspan='7'>No se encontraron evaluaciones</td></tr>";
    }
    Cerrar($Con);
    ?>
</tbody>
</table>
</div>
</div>

<div id="modalObservaciones" class="modal">
    <div class="modal-content">
        <span class="close">&times;</span>
        <h3>Observaciones y Recomendaciones</h3>
        <hr class="x-component x-component-default" style="border-top: 0; border-bottom: 0.05rem solid #196ad3; margin: auto; width: 100%; margin-bottom: 2rem;">
        <div id="observacionContent"></div>
    </div>
</div>

<script>
    document.getElementById('search-name').addEventListener('input', filterTable);
    document.getElementById('search-periodo').addEventListener('change', filterTable);

    function filterTable() {
        const nameInput = document.getElementById('search-name').value.toLowerCase();
        const periodoInput = document.getElementById('search-periodo').value;
        const rows = document.querySelectorAll('#table-body tr');

        rows.forEach(row => {
            const nombre = row.getAttribute('data-nombre').toLowerCase();
            const periodo = row.getAttribute('data-periodo');
            const matchesName = nombre.includes(nameInput);
            const matchesPeriodo = periodoInput === '' || periodo === periodoInput;

            if (matchesName && matchesPeriodo) {
                row.style.display = '';
            } else {
                row.style.display = 'none';
            }
        });
    }
}

```

```

/* Estilo placeholder en select */
document.getElementById('search-periodo').addEventListener('change', function() {
    if (this.value) {
        this.style.color = 'var(--text-color)'; // Cambia al color fuerte
    } else {
        this.style.color = '#757575'; // Mantiene el color tenue si no hay selección
    }
});

// Obtener todos los botones de ver observación
const botonesObservacion = document.querySelectorAll('.ver-observacion');
const modal = document.getElementById('modalObservaciones');
const observacionContent = document.getElementById('observacionContent');

// Agregar evento click a cada botón
botonesObservacion.forEach(boton => {
    boton.addEventListener('click', () => {
        const observacion = boton.getAttribute('data-observacion');
        observacionContent.textContent = observacion;
        modal.style.display = "block";
        // Forzar un reflow para que la transición funcione
        modal.offsetHeight;
        modal.classList.add('show');
    });
});

// Cerrar modal con el botón X
document.querySelector('.close').onclick = () => {
    cerrarModal();
};

// Cerrar modal al hacer clic fuera
window.onclick = (event) => {
    if (event.target === modal) {
        cerrarModal();
    }
};

function cerrarModal() {
    modal.classList.remove('show');
}

```

```
        setTimeout(() => {
            modal.style.display = "none";
        }, 300); // Este tiempo debe coincidir con la duración de la transición en CSS
    }
</script>

</body>

</html>
```

Cerrar.php

```
<?php
    session_start();
    session_unset();
    session_destroy();
    print('<META HTTP-EQUIV="REFRESH" CONTENT="1;URL=../Index/index.html"> </head>');
?>
```

MenuA.php

```
<?php
session_start();
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/styles.css">
    <title>Alumno</title>
</head>
<body>
<header class="header">
    <div class="container-titulo-header">
        <a class="titulo-header" href="../Fondo estatico/Alumno.php">SSAP FIF UAQ</a>
    </div>
    <div class="container-botones">
        <ul class="container-lista-botones">
            <li class="boton-link">
                <a class="agendar-evaluacion" href="../Alumno/AgendarEvaluacion.php"><button type="button">Agendar Evaluación</button></a>
            </li>
            <li class="boton-link">
                <a class="evaluaciones-agendadas" href="../Alumno/EvaluacionesAgendadas.php"><button type="button">Evaluaciones Agendadas</button></a>
            </li>
            <li class="boton-link">
                <a class="mis-evaluaciones" href="../Alumno/MisEvaluaciones.php"><button type="button">Mis Evaluaciones</button></a>
            </li>
            <li class="boton-link">
                <a class="documentos" href="../Alumno/Documentos.php"><button type="button">Documentos</button></a>
            </li>
            <li class="boton-link">
                <a class="avisos" href="Avisos.php" ><button type="button">Avisos</button></a>
            </li>
        </ul>
    </div>
</header>
<div class="container-principal">
    <div class="container-encabezado">
        <h1>Alumno</h1>
    </div>
    <div class="container-contenido">
        <p>Este es el menú principal para el Alumno. Puedes acceder a las siguientes secciones:</p>
        <ul class="listado-secciones">
            <li>Agendar Evaluación</li>
            <li>Evaluaciones Agendadas</li>
            <li>Mis Evaluaciones</li>
            <li>Documentos</li>
            <li>Avisos</li>
        </ul>
    </div>
</div>
</body>
</html>
```

```

        </li>
    </ul>
</div>
<div class="container-usuario">
    <a class="usuario">Usuario: <?php echo(isset($_SESSION['Nombre']) ?  

$_SESSION['Nombre'] : 'Invitado'); ?></a>
</div>
<div class="container-cerrar-btn">
    <a class="cerrar-btn" href="../Header/Cerrar.php"><button  

type="button">Cerrar</button></a>
</div>
</header>

<div class="header-telefono">
    <div class="container-titulo-sidebarbtn">
        <a class="titulo-header" href="../Fondo estatico/Alumno.php">SSAP FIF UAQ</a>
        <button class="open-sidebar" onclick="openNav()">☰</button>
    </div>

    <div id="mySidebar" class="sidebar">
        <span href="javascript:void(0)" class="close-sidebar"  

onclick="closeNav()">✖</span>
        <span class="espacio-sidebar"></span>
        <a>Usuario: <?php echo(isset($_SESSION['Nombre']) ? $_SESSION['Nombre'] :  

'Invitado'); ?></a>
        <a href="../Alumno/AgendarEvaluacion.php">Agendar Evaluación</a>
        <a href="../Alumno/EvaluacionesAgendadas.php">Evaluaciones Agendadas</a>
        <a href="../Alumno/MisEvaluaciones.php">Mis evaluaciones</a>
        <a href="../Alumno/Documentos.php">Documentos</a>
        <a href="Avisos.php">Avisos</a>
        <a class="cerrar-btn-telefono" href="../Header/Cerrar.php">Cerrar</a>
    </div>
</div>
<script>
    function openNav() {
        document.getElementById("mySidebar").style.width = "100vw";
    }

    function closeNav() {
        document.getElementById("mySidebar").style.width = "0";
    }
</script>

```

```

        }
    </script>
</body>
</html>
```

MenuC.php

```

<?php
session_start();
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/styles.css">
    <title>Coordinador</title>
</head>
<body>
    <header class="header">
        <div class="container-titulo-header">
            <a class="titulo-header" href="../Fondo estatico/Coordinador.php">SSAP FIF UAQ</a>
        </div>
        <div class="container-botones">
            <ul class="container-lista-botones">
                <li class="boton-link">
                    <a class="asignar-sinodo" href="../Coordinador/AsignarSinodo.php"><button type="button">Asignar Sínodo</button></a>
                </li>
                <li class="boton-link">
                    <a class="asignar-sinodo"
                        href="../Coordinador/SinodosAsignados.php"><button type="button">Sinodos asignados</button></a>
                </li>
                <li class="boton-link">
                    <a class="asignar-sinodo"
                        href="../Coordinador/EvaluacionesAgendadasC.php"><button type="button">Evaluaciones Agendadas</button></a>
                </li>
            </ul>
        </div>
    </header>
</body>
```

```

        </li>
        <li class="boton-link">
            <a class="agendar-evaluacion"
href="../Coordinador/AltaDocentes.php"><button type="button">Alta de docentes</button></a>
        </li>
        <li class="boton-link">
            <a class="seguimiento" href=""><button
type="button">Seguimiento</button></a>
        </li>
    </ul>
</div>
<div class="container-usuario">
    <a class="usuario">Usuario: <?php echo(isset($_SESSION['Nombre']) ?
$_SESSION['Nombre'] : 'Invitado');?></a>
</div>
<div class="container-cerrar-btn">
    <a class="cerrar-btn" href="../Header/Cerrar.php"><button
type="button">Cerrar</button></a>
</div>
</header>

<div class="header-telefono">
<div class="container-titulo-sidebarbtn">
    <a class="titulo-header" href="../Fondo estatico/Alumno.php">SSAP FIF UAQ</a>
    <button class="open-sidebar" onclick="openNav()">☰</button>
</div>

<div id="mySidebar" class="sidebar">
    <span href="javascript:void(0)" class="close-sidebar"
onclick="closeNav()">×</span>
    <span class="espacio-sidebar"></span>
    <a id="usuario-sidebar" >Usuario: <?php echo(isset($_SESSION['Nombre']) ?
$_SESSION['Nombre'] : 'Invitado');?></a>
    <a href="../Coordinador/AsignarSinodo.php">Asignar sínodo</a>
    <a href="../Coordinador/SinodosAsignados.php">Sinodos asignados</a>
    <a href="../Coordinador/EvaluacionesAgendadasC.php">Evaluaciones agendadas</a>
    <a href="../Coordinador/AltaDocentes.php">Alta de docentes</a>
    <a href="">Seguimiento</a>
    <a class="cerrar-btn-telefono" href="../Header/Cerrar.php">Cerrar</a>
</div>
</div>

```

```

<script>
    function openNav() {
        document.getElementById("mySidebar").style.width = "100vw";
    }

    function closeNav() {
        document.getElementById("mySidebar").style.width = "0";
    }
</script>
</body>
</html>

```

MenuD.php

```

<?php
session_start();
?>

<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="../../CSS/styles.css">
    <title>Docente</title>
</head>
<body>
    <header class="header">
        <div class="container-titulo-header">
            <a class="titulo-header" href="../Fondo estatico/Docente.php">SSAP FIF UAQ</a>
        </div>
        <div class="container-botones">
            <ul class="container-lista-botones">
                <li class="boton-link">
                    <a class="evaluaciones-pendientes"
href="../Docente/EvaluacionesPendientes.php"><button type="button">Evaluaciones
Pendientes</button></a>
                </li>
                <li class="boton-link">

```

```

        <a class="historial-evaluaciones"
href="../Docente/HistorialEvaluaciones.php"><button type="button">Historial de
Evaluaciones</button></a>
    </li>
</ul>
</div>
<div class="container-usuario">
    <a class="usuario">Usuario: <?php echo(isset($_SESSION['Nombre']) ? $_SESSION['Nombre'] : 'Invitado');?></a>
</div>
<div class="container-cerrar-btn">
    <a class="cerrar-btn" href="../Header/Cerrar.php"><button
type="button">Cerrar</button></a>
</div>
</header>

<div class="header-telefono">
<div class="container-titulo-sidebarbtn">
    <a class="titulo-header" href="../Fondo estatico/Alumno.php">SSAP FIF UAQ</a>
    <button class="open-sidebar" onclick="openNav()">☰</button>
</div>

<div id="mySidebar" class="sidebar">
    <span href="javascript:void(0)" class="close-sidebar"
onclick="closeNav()">✖</span>
    <span class="espacio-sidebar"></span>
    <a>Usuario: <?php echo(isset($_SESSION['Nombre']) ? $_SESSION['Nombre'] :
'Invitado');?></a>
        <a href="../Docente/EvaluacionesPendientes.php">Evaluaciones Pendientes</a>
        <a href="../Docente/HistorialEvaluaciones.php">Historial de Evaluaciones</a>
        <a class="cerrar-btn-telefono" href="../Header/Cerrar.php">Cerrar</a>
    </div>
</div>
<script>
    function openNav() {
        document.getElementById("mySidebar").style.width = "100vw";
    }

    function closeNav() {
        document.getElementById("mySidebar").style.width = "0";
    }

```

```
</script>
</body>
</html>
```

index.html

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Posgrado FIF</title>
    <link rel="stylesheet" href="../../CSS/loginDesign.css">
</head>

<body>
    <div class="container1" id="initialContent">
        <!-- Contenido inicial -->
        <form method="POST" action="ValidaAcceso.php">
            
            <h1>Iniciar sesión</h1>
            <label for="inputUsuario">Usuario:</label>
            <input type="text" name="inputUsuario" id="inputUsuario">
            <br>
            <label for="inputPassword">Contraseña:</label>
            <div class="password-container">
                <input type="password" name="inputPassword" id="inputPassword">
                <button type="button" id="togglePassword" class="toggle-password">
                    
                </button>
            </div>
            <br>
            <input type="submit" value="Enviar" class="boton_enviar">
            <hr class="x-component x-component-default" style="border-color:#ececce; margin:auto; width:60%;"
                id="box-1034">
        </form>
    </div>
</body>
```

```

        <a href="#" class="boton_registrarse"
    onclick="showRegistrationOptions()">Registrarse</a>
    </form>
</div>
<p class="año">&copy; 2024</p>
<div class="container2" id="registrationOptions" style="display: none;">
    <h1>Registro de alumno</h1>
    <span class="espacio">espacio</span>
    <form action="RegistroAlumno.php" method="POST">
        <div class="grid-container">
            <div>
                <label for="expediente">Expediente:</label>
                <input type="number" name="expediente" id="expediente" required>
            </div>
            <div>
                <label for="nombre">Nombre:</label>
                <input type="text" name="nombre" id="nombre" required>
            </div>
            <div>
                <label for="apellidoPaterno">Apellido paterno:</label>
                <input type="text" name="apellidoPaterno" id="apellidoPaterno" required>
            </div>
            <div>
                <label for="apellidoMaterno">Apellido materno:</label>
                <input type="text" name="apellidoMaterno" id="apellidoMaterno" required>
            </div>
            <div>
                <label for="telefono">Teléfono:</label>
                <input type="text" name="telefono" id="telefono" maxlength="10"
                    oninput="this.value = this.value.replace(/\[^0-9]/g, '').slice(0, 10)" required>
            </div>
            <div>
                <label for="correo">Correo electrónico:</label>
                <input type="email" name="correo" id="correo" required>
            </div>
            <div>
                <label for="programa">Programa:</label>
                <select name="programa" id="programa" required>
                    <option value="" disabled selected>Selecciona...</option>

```

```

        <option value="DITE">DITE (Doctorado en Innovación en Tecnología
educativa)</option>
        <option value="DTE">DTE (Doctorado en Tecnología Educativa)</option>
        <option value="DCC">DCC (Doctorado en Ciencias de la
Computación)</option>
        <option value="MCC">MCC (Maestría en Ciencias de la
Computación)</option>
        <option value="MSC">MSC (Maestría en Sistemas
Computacionales)</option>
        <option value="MIEVEA">MIEVEA (Maestría en Innovación de Entornos
Virtuales de Enseñanza y Aprendizaje)</option>
        <option value="MCD">MCD (Maestría en Ciencia De Datos)</option>
    </select>
</div>
</div>
<input type="submit" value="Guardar" class="boton_enviar">
<button type="button" class="boton_regresar" onclick="showInitialContent()">
    <p>Regresar</p>
</button>
</form>
</div>


<div id="popup" class="popup" style="display: none;">
    <div class="popup-content">
        <span class="close-btn" onclick="closePopup()">&times;</span>
        <p id="popup-text"></p>
        <button id="copy-btn" class="copy-btn" onclick="copyToClipboard()">Copiar</button>
    </div>
</div>

<script src="../../JS/script.js"></script>
</body>

</html>

```

LimpiaCadenas.php

```
<?php
function eliminar_acentos($cadena){

    //Reemplazamos la A y a
    $cadena = str_replace(
        array('Á', 'À', 'Â', 'Ã', 'á', 'à', 'â', 'ã'),
        array('A', 'A', 'A', 'A', 'a', 'a', 'a', 'a'),
        $cadena
    );

    //Reemplazamos la E y e
    $cadena = str_replace(
        array('É', 'È', 'Ê', 'Ë', 'é', 'è', 'ë', 'ê'),
        array('E', 'E', 'E', 'E', 'e', 'e', 'e', 'e'),
        $cadena );
    //Reemplazamos la I y i
    $cadena = str_replace(
        array('Í', 'Ì', 'Ï', 'Î', 'í', 'ì', 'ï', 'î'),
        array('I', 'I', 'I', 'I', 'i', 'i', 'i', 'i'),
        $cadena );
    //Reemplazamos la O y o
    $cadena = str_replace(
        array('Ó', 'Ò', 'Ö', 'Ô', 'ó', 'ò', 'ö', 'ô'),
        array('O', 'O', 'O', 'O', 'o', 'o', 'o', 'o'),
        $cadena );
    //Reemplazamos la U y u
    $cadena = str_replace(
        array('Ú', 'Ù', 'Û', 'Ü', 'ú', 'ù', 'û', 'û'),
        array('U', 'U', 'U', 'U', 'u', 'u', 'u', 'u'),
        $cadena );
    //Reemplazamos la N, n, C y c
    $cadena = str_replace(
        array('Ñ', 'ñ', 'Ҫ', 'ҫ'),
        array('N', 'n', 'C', 'c'),
```

```

$cadena
);

return $cadena;
}

?>

```

RegistroAlumno.php

```

<?php
if ($_SERVER[ "REQUEST_METHOD" ] == "POST") {
    // Obtener datos del formulario y limpiar las entradas
    $Expediente = trim($_POST['expediente']);
    $Nombre = trim($_POST['nombre']);
    $ApellidoP = trim($_POST['apellidoPaterno']);
    $ApellidoM = trim($_POST['apellidoMaterno']);
    $Telefono = trim($_POST['telefono']);
    $Correo = trim($_POST['correo']);
    $Programa = trim($_POST['programa']);

    // Quitar espacios al inicio y al final
    $Nombre=trim( $Nombre);
    $ApellidoM=trim($ApellidoM);
    $ApellidoP=trim($ApellidoP);
    $Telefono=trim($Telefono);

    // Convertir a mayúsculas y quitar acentos
    include("LimpiaCadenas.php");
    $Nombre = strtoupper(eliminar_acentos($Nombre));
    $ApellidoP = strtoupper(eliminar_acentos($ApellidoP));
    $ApellidoM = strtoupper(eliminar_acentos($ApellidoM));

    // Crear contraseña
    $Password = $ApellidoP . substr($Telefono, -2);

    //Variable de nombre completo
    $NombreC = "$ApellidoP $ApellidoM $Nombre";
}

```



```
"Recuerda guardar tus credenciales para acceder al  
sistema.";  
  
$response['copyText'] = "Tus credenciales de acceso son:\n" .  
    "Usuario: $Expediente\n" .  
    "Password: $Password";  
} else {  
    $response['status'] = 'error';  
    $response['message'] = "Error al registrar la cuenta: " . mysqli_error($Con);  
}  
} else {  
    $response['status'] = 'exists';  
    $response['message'] = "La cuenta ya ha sido registrada anteriormente.\n\n" .  
        "Para recuperar tus credenciales de acceso, envía un correo a:  
francisco.javier.paulin@uaq.mx";  
}  
  
echo json_encode($response);  
  
// Cerrar la conexión  
mysqli_close($Con);  
}  
?>
```

ValidaAcceso.php

```
<?php
session_start(); // Iniciar la sesión

// Obtener datos del formulario
$Expediente = $_POST['inputUsuario'];
>Password = $_POST['inputPassword']; // Esto debería ser la contraseña en texto plano

include("../Config/conexion.php");
$Con = Conectar();

// Verificar la conexión
if (!$Con) {
    die("Error de conexión: " . mysqli_connect_error());
}

// Consulta SQL para verificar el usuario
$query = "SELECT * FROM cuentas WHERE id = ?";
$stmt = $Con->prepare($query);
$stmt->bind_param("i", $Expediente);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows == 1) {
    $row = $result->fetch_assoc();

    // Comparar directamente si ambos son hashes
    if ($Password === $row['contraseña']) {
        $_SESSION['id'] = $row['id'];
        $_SESSION['tipo'] = $row['tipo'];

        // Consulta para obtener el nombre completo según el tipo de usuario
        switch ($row['tipo']) {
            case 'A':
                $query_usuario = "SELECT nombre, a_paterno, a_materno FROM estudiantes WHERE
exp = ?;";
                break;
            case 'D':
```

```

$query_usuario = "SELECT nombre, a_paterno, a_materno FROM docentes WHERE
clave = ?";
break;
case 'C':
$query_usuario = "SELECT nombre, a_paterno, a_materno FROM coordinadores WHERE
clave = ?";
break;
default:
$query_usuario = null;
}

if ($query_usuario) {
$stmt_usuario = $Con->prepare($query_usuario);
$stmt_usuario->bind_param("i", $Expediente);
$stmt_usuario->execute();
$result_usuario = $stmt_usuario->get_result();

if ($result_usuario->num_rows == 1) {
$row_usuario = $result_usuario->fetch_assoc();
// Guardar el nombre completo del usuario en la sesión
$_SESSION['Nombre'] = $row_usuario['nombre'] . ' ' . $row_usuario['a_paterno']
. ' ' . $row_usuario['a_materno'];
}
$stmt_usuario->close();
}

// Redirigir según el tipo de usuario
switch ($row['tipo']) {
case 'A':
header('Location: ../Fondo estatico/Alumno.php');
break;
case 'C':
header('Location: ../Fondo estatico/Coordinador.php');
break;
case 'D':
header('Location: ../Fondo estatico/Docente.php');
break;
default:
session_destroy();
echo '<script type="text/javascript">
alert("Tipo de cuenta no válido.");

```

```
        window.location.href="../../index.html";
    </script>';
break;
}
} else {
echo '<script type="text/javascript">
    alert("Usuario o Contraseña Incorrecta");
    window.location.href="../../index.html";
</script>';
}
} else {
echo '<script type="text/javascript">
    alert("Usuario no encontrado.");
    window.location.href="../../index.html";
</script>';
}

// Cerrar la conexión
$stmt->close();
mysqli_close($Con);
?>
```

script.js

```
function validarFormulario() {
    var username = document.getElementById("username").value;
    var password = document.getElementById("password").value;

    if (username === "" || password === "") {
        alert("Por favor, ingresa el usuario y la contraseña.");
        return false;
    }
}

function showRegistrationOptions() {
    var initialContent = document.getElementById("initialContent");
    var registrationOptions = document.getElementById("registrationOptions");

    initialContent.style.display = "none"; // Oculta content1
    registrationOptions.style.display = "block"; // Muestra content2
}

function showInitialContent() {
    var initialContent = document.getElementById("initialContent");
    var registrationOptions = document.getElementById("registrationOptions");
    var form = document.querySelector('form[action="RegistroAlumno.php"]');

    // Muestra el contenido inicial y oculta el formulario de registro
    initialContent.style.display = "block";
    registrationOptions.style.display = "none";

    // Limpia los campos del formulario
    if (form) {
        form.reset();
    }
}

function validarTelefono(event) {
    event.target.value = event.target.value.replace(/\D/g, '').slice(0, 10);
}
```

```
let operationSuccess = false; // Variable global para el estado de la operación
document.querySelector('form[action="RegistroAlumno.php"]').addEventListener('submit',
function(event) {
    event.preventDefault(); // Evita el envío del formulario de la manera tradicional

    const formData = new FormData(this);

    fetch('RegistroAlumno.php', {
        method: 'POST',
        body: formData
    })
    .then(response => {
        console.log('Response status:', response.status);
        return response.text().then(text => {
            console.log('Response text:', text);
            try {
                return JSON.parse(text);
            } catch (e) {
                console.error('Error parsing JSON:', e);
                throw new Error('Invalid JSON response: ' + text);
            }
        });
    })
    .then(data => {
        const popup = document.getElementById('popup');
        const popupText = document.getElementById('popup-text');
        const copyButton = document.getElementById('copy-btn');

        // Oculta el botón de copiar por defecto
        copyButton.style.display = 'none';

        if (data.status === 'success') {
            operationSuccess = true; // Marca la operación como exitosa
            // Mostrar el mensaje y el botón de copiar
            popupText.innerText = data.message;
            copyButton.style.display = 'block';
            window.copyText = data.copyText; // Guardamos el texto para copiar
        } else if (data.status === 'exists') {
            // Mostrar el mensaje de que la cuenta ya existe sin el botón de copiar
            popupText.innerText = data.message;
        } else {

```

```

        popupText.innerText = data.message || 'Error desconocido al registrar la cuenta';
    }

    // Mostrar el popup
    popup.style.display = 'block';
})
.catch(error => {
    console.error('Error completo:', error);
    const popup = document.getElementById('popup');
    const popupText = document.getElementById('popup-text');
    popupText.innerText = 'Error al procesar la solicitud: ' + error.message;
    popup.style.display = 'block';
});
});

function copyToClipboard() {
    if (window.copyText) {
        navigator.clipboard.writeText(window.copyText)
        .then(() => {
            alert("Credenciales copiadas al portapapeles");
        })
        .catch(err => {
            console.error('Error al copiar:', err);
            alert("Error al copiar el texto");
        });
    }
}

function closePopup() {
    document.getElementById("popup").style.display = "none";
    if (operationSuccess) {
        window.location.href = 'index.html'; // Redirige al inicio
    }
}

document.addEventListener('DOMContentLoaded', function() {
    const togglePassword = document.querySelector('#togglePassword');
    const password = document.querySelector('#inputPassword');

    // Función para mostrar/ocultar la contraseña
    togglePassword.addEventListener('click', function (e) {

```

```

        const type = password.getAttribute('type') === 'password' ? 'text' : 'password';
        password.setAttribute('type', type);
        this.querySelector('img').src = type === 'password' ? '../../../../../Imagenes/eye-icon.png' :
        '../../../../../Imagenes/eye-slash-icon.png';
    });

    // Función para mostrar el ícono solo si hay texto
    password.addEventListener('input', function() {
        if (this.value) {
            togglePassword.style.display = 'block'; // Muestra el ícono
        } else {
            togglePassword.style.display = 'none'; // Oculta el ícono
        }
    });

    // Inicialmente oculta el ícono
    togglePassword.style.display = 'none';
});

```

ActaSeminarioTutorial.php

```

<?php
$url = "https://www.timeapi.io/api/time/current/zone?timeZone=AMERICA%2FMEXICO_CITY";

// Obtener datos de la API
$response = file_get_contents($url);
$data = json_decode($response, true);

// Obtener la fecha actual en formato ISO
$fechaActual = $data['year']; // Ejemplo: "2024-11-19T12:34:56-06:00"
$fechaDate = $data['day'];

// Mostrar solo la fecha
$year = substr($fechaActual, 3, 1); // "2024-11-19"

function obtenerNombreMes($mesNum)
{
    $meses = [

```

```

1 => 'Enero',
2 => 'Febrero',
3 => 'Marzo',
4 => 'Abril',
5 => 'Mayo',
6 => 'Junio',
7 => 'Julio',
8 => 'Agosto',
9 => 'Septiembre',
10 => 'Octubre',
11 => 'Noviembre',
12 => 'Diciembre'

];
return $meses[(int)$mesNum];
}

$mesNum = $data['month'];
$mesNombre = obtenerNombreMes($mesNum);

if (session_status() == PHP_SESSION_NONE) {
    session_start();
}
include("../Config/conexion.php");

$id_alumno = $_SESSION['id'];
$SQL = "SELECT d.nombre, d.a_paterno, d.a_materno, de.observacion
FROM asignaciones a
INNER JOIN evaluaciones ev ON a.exp_alumno = ev.exp_alumno
INNER JOIN detalle_evaluaciones de ON ev.id = de.id_evaluacion
INNER JOIN docentes d ON de.id_sinodo = d.clave
WHERE a.exp_alumno = '$id_alumno'";

$SQL2 = "SELECT c.nombre AS coor_nombre,
c.a_paterno AS coor_paterno,
c.a_materno AS coor_materno,
e.nombre, e.a_paterno, e.a_materno,
ev.cal_final
FROM asignaciones a
INNER JOIN coordinadores c ON a.clave_coordinador = c.clave
INNER JOIN estudiantes e ON a.exp_alumno = e.exp

```

```

INNER JOIN evaluaciones ev ON a.exp_alumno = ev.exp_alumno
WHERE a.exp_alumno = '$id_alumno';

$SQL3 = "SELECT
          de.d_observacion1,
          de.d_observacion2,
          de.d_observacion3,
          de.observacion AS observacion_general,
          CONCAT(d.nombre, ' ', d.a_paterno, ' ', d.a_materno) AS nombre_sinodo,
          de.observacion AS observacion_sinodo
      FROM
          detalle_evaluaciones de
      JOIN
          evaluaciones e ON de.id_evaluacion = e.id
      JOIN
          asignaciones a ON e.exp_alumno = a.exp_alumno
      JOIN
          docentes d ON de.id_sinodo = d.clave
      WHERE
          e.exp_alumno = '$id_alumno' AND
          de.id_evaluacion = e.id;
      ";

$Con = Conectar();
$Resultado = Ejecutar($Con, $SQL);
$Resultado2 = Ejecutar($Con, $SQL2);
$Resultado3 = Ejecutar($Con, $SQL3);
if (mysqli_num_rows($Resultado2) > 0) {
    $Fila = mysqli_fetch_array($Resultado2);
    $Nom_Coordinador = $Fila['coor_nombre'] . " " . $Fila['coor_paterno'] . " " .
    $Fila['coor_materno'];
    $Nom_Alumno = $Fila['nombre'] . " " . $Fila['a_paterno'] . " " . $Fila['a_materno'];
    $Cal_Final = $Fila['cal_final'];
}

if (mysqli_num_rows($Resultado3) > 0){
    $Fila = mysqli_fetch_array($Resultado3);
    $obs1 = $Fila['d_observacion1'];
    $obs2 = $Fila['d_observacion2'];
    $obs3 = $Fila['d_observacion3'];
}

```

```

$obs = $Fila['observacion_general'];
$nomSinodo = $Fila['nombre_sinodo'];
$obssinodo = $Fila['observacion_sinodo'];

}

Cerrar($Con);

require('fpdf.php');

// Establecer márgenes (izquierdo, superior, derecho)

// Crear una clase personalizada que hereda de FPDF
class PDF extends FPDF
{
    // Función para el encabezado
    function Header()
    {
        // Posicionamiento de la primera imagen (izquierda)
        $this->Image('../libraries/logoFIF.png', 24, -10, 55); // Ajusta la ruta y el tamaño

        // Posicionamiento de la segunda imagen (derecha)
        $this->Image('../libraries/mcclogo.png', 134, 6, 55); // Ajusta la ruta y el tamaño
    }
}

// Crear instancia de la clase PDF personalizada
$pdf = new PDF();
$pdf->AddPage();
$pdf->SetMargins(25, 40, 25);

// Definir el interlineado deseado (altura de la celda)
$lineHeight = 6;

$pdf->Ln(30);

// Título principal del documento
$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(0, $lineHeight, "Campus Juriquilla a $fechaDate de $mesNombre del 20$year", 0, 1,
'R');

```

```

$pdf->Ln(10);

// Saludo
$pdf->SetFont('Arial', '', 12);
$pdf->Cell(0, $lineHeight, utf8_decode('Dr. ' . " " . $Nom_Coordinador), 0, 1);
$pdf->Cell(0, $lineHeight, utf8_decode('Coordinador del programa de la Maestría en Ciencias de
la Computación'), 0, 1);
$pdf->Ln(12);
$pdf->SetFont('Arial', 'B', 12);

$pdf->Cell(0, $lineHeight, utf8_decode('P r e s e n t e'), 0, 1);
$pdf->Ln(10);

$pdf->SetFont('Arial', '', 12);

// Cuerpo principal
$pdf->MultiCell(0, $lineHeight, utf8_decode("Por este medio, le informamos que el comité
académico del estudiante $Nom_Alumno\n"
. "con número de expediente $id_alumno, se reunió de manera virtual, para realizar la
evaluación\n"
. "del avance del proyecto de investigación.\n\n"
. "De acuerdo con lo planteado en el cronograma de trabajo y a los requerimientos del
_____ semestre\n"
. "del programa del Doctorado en Ciencias de la Computación del cual se tienen los
siguientes resultados:"), 0, 'J');
$pdf->Ln($lineHeight);

// Secciones de evaluación
$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(0, $lineHeight, utf8_decode('Sobre Avance gradual'), 0, 1);
$pdf->SetFont('Arial', '', 12);
$pdf->MultiCell(0, $lineHeight, utf8_decode("Observaciones y recomendaciones:\n\n" . $obs1),
0, 'J');
$pdf->Ln(10); // Espacio entre secciones

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(0, $lineHeight, utf8_decode("Sobre entregable y resultados esperados de acuerdo con
el semestre a evaluar"), 0, 1);
$pdf->Cell(0, $lineHeight, utf8_decode("(ver anexo)'), 0, 1);
$pdf->SetFont('Arial', '', 12);

```

```

$pdf->MultiCell(0, $lineHeight, utf8_decode("Observaciones y recomendaciones:\n\n" . $obs2),
0, 'J');
$pdf->Ln(10); // Espacio entre secciones

$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(0, $lineHeight, utf8_decode('Sobre el Avance del Proyecto'), 0, 1);
$pdf->SetFont('Arial', '', 12);
$pdf->MultiCell(0, $lineHeight, utf8_decode("(Avances en la metodología y pertinencia en la
estrategia experimental)\n\n"
. "Observaciones y recomendaciones:\n\n" . $obs3), 0, 'J');
$pdf->Ln(5); // Espacio entre secciones

// Calificación
$pdf->Ln(10); // Calificación
$pdf->SetFont('Arial', 'B', 12);
$pdf->Cell(0, $lineHeight, utf8_decode("Calificación: $Cal_Final"), 0, 1);
$pdf->Ln($lineHeight);

// Comité
$pdf->SetFont('Arial', 'B', 12);
$columna = 80;
$pdf->Cell($columna, $lineHeight, utf8_decode('INTEGRANTES DEL COMITÉ'), 1, 0, 'C');
$pdf->Cell($columna, $lineHeight, utf8_decode('COMENTARIO INDIVIDUAL'), 1, 1, 'C');

// Configuración de las columnas
$colWidth = 80; // Ancho de cada columna
$minHeight = 30; // Altura mínima de cada fila
$fontHeight = 4; // Altura de cada línea de texto

$pdf->SetFont('Arial', '', 10);

if (mysqli_num_rows($Resultado) > 0) {
    while ($sinodo = mysqli_fetch_array($Resultado)) {
        $nombreSinodo = utf8_decode($sinodo['nombre'] . ' ' . $sinodo['a_paterno'] . ' ' .
$sinodo['a_materno']);
        $comentario = utf8_decode($sinodo['observacion']);

        // Calcular altura necesaria para el comentario
        $pdf->SetFont('Arial', '', 10);
        $comentarioHeight = $pdf->GetStringWidth($comentario) / $colWidth * $fontHeight;
    }
}

```

```

$cellHeight = max($minHeight, $comentarioHeight + 10);

// Verificar si hay espacio suficiente en la página
if ($pdf->GetY() + $cellHeight > $pdf->GetPageHeight() - 30) {
    $pdf->AddPage();
    $pdf->SetFont('Arial', 'B', 12);
    $pdf->Cell($columna, $lineHeight, utf8_decode('INTEGRANTES DEL COMITÉ'), 1, 0,
'C');
    $pdf->Cell($columna, $lineHeight, utf8_decode('COMENTARIO INDIVIDUAL'), 1, 1,
'C');
    $pdf->SetFont('Arial', '', 10);
}

// Guardar posición inicial
$startY = $pdf->GetY();
$startX = $pdf->GetX();

// Dibujar celdas con la altura calculada
$pdf->Cell($colWidth, $cellHeight, '', 1, 0);
$pdf->SetXY($startX + $colWidth, $startY);
$pdf->Cell($colWidth, $cellHeight, '', 1, 0);

// Nombre centrado verticalmente
$pdf->SetXY($startX, $startY + ($cellHeight/2) - 2);
$pdf->Cell($colWidth, 5, $nombreSinodo, 0, 0, 'C');

// Comentario con MultiCell
$pdf->SetXY($startX + $colWidth + 1, $startY + 2);
$pdf->MultiCell($colWidth - 2, $fontHeight, $comentario, 0, 'L');

// Mover a la siguiente línea
$pdf->SetXY($startX, $startY + $cellHeight);
}

} else {
    $pdf->Cell($colWidth, $minHeight, utf8_decode('Sin información de sinodales'), 1, 0, 'C');
    $pdf->Cell($colWidth, $minHeight, utf8_decode('Sin comentarios'), 1, 1, 'C');
}

// Generar el PDF
$pdf->Output('I', 'ActaSeminarioTutorial.pdf'); // 'I' para mostrar en el navegador, 'F' para
guardar en servidor

```

insertar_evaluacion.php

```
<?php

// Incluir el archivo de conexión
include 'Config/conexion.php';

// Conectar a la base de datos
$Con = Conectar();

// Asegúrate de que la solicitud sea POST
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $exp_alumno = mysqli_real_escape_string($Con, $_POST['exp']);
    $fecha_evaluacion = mysqli_real_escape_string($Con, $_POST['fecha_evaluacion']);
    $aula = mysqli_real_escape_string($Con, $_POST['aula']);

    $archivoGuardado = "";

    // Manejo de archivo (si existe)
    if (isset($_FILES['entregable']) && $_FILES['entregable']['error'] == 0) {
        $basePath = "Entregables/$exp_alumno";

        // Crear la carpeta si no existe
        if (!file_exists($basePath)) {
            mkdir($basePath, 0777, true); // Crear carpetas con permisos adecuados
        }

        // Guardar el archivo
        $fileName = basename($_FILES['entregable']['name']);
        $targetPath = $basePath . '/' . $fileName;

        if (move_uploaded_file($_FILES['entregable']['tmp_name'], $targetPath)) {
            $archivoGuardado = $targetPath; // Ruta del archivo guardado
        } else {
            echo "Error al guardar el archivo.";
            exit;
        }
    } else {
        echo "No se seleccionó ningún archivo o hubo un error en la carga.";
        exit;
    }
}
```

```

// Iniciar transacción
mysqli_begin_transaction($Con);

try {
    // Insertar en evaluaciones
    $SQL = "INSERT INTO evaluaciones (exp_alumno, fecha_evaluacion, aula) VALUES (?, ?, ?)";
    $stmt = mysqli_prepare($Con, $SQL);
    mysqli_stmt_bind_param($stmt, "sss", $exp_alumno, $fecha_evaluacion, $aula);
    mysqli_stmt_execute($stmt);
    $id_evaluacion = mysqli_insert_id($Con);

    // Obtener sinodales de asignaciones
    $SQL_sinodales = "SELECT director, sinodo2, sinodo3, externo FROM asignaciones WHERE
exp_alumno = ?";
    $stmt_sinodales = mysqli_prepare($Con, $SQL_sinodales);
    mysqli_stmt_bind_param($stmt_sinodales, "s", $exp_alumno);
    mysqli_stmt_execute($stmt_sinodales);
    $result = mysqli_stmt_get_result($stmt_sinodales);
    $asignacion = mysqli_fetch_assoc($result);

    // Insertar en detalle_evaluaciones para cada sinodal
    $sinodales = array_filter([$asignacion['director'], $asignacion['sinodo2'],
$asignacion['sinodo3'], $asignacion['externo']]);
    $SQL_detalle = "INSERT INTO detalle_evaluaciones (id_evaluacion, id_sinodo) VALUES (?, ?)";
    $stmt_detalle = mysqli_prepare($Con, $SQL_detalle);

    foreach ($sinodales as $sinodal) {
        mysqli_stmt_bind_param($stmt_detalle, "is", $id_evaluacion, $sinodal);
        mysqli_stmt_execute($stmt_detalle);
    }

    // Confirmar la transacción
    mysqli_commit($Con);
    echo "Evaluación agendada correctamente para el expediente: $exp_alumno.";
} catch (Exception $e) {
    // Revertir la transacción en caso de error
    mysqli_rollback($Con);
    echo "Error al agendar evaluación: " . $e->getMessage();
}

```

```
}

// Cerrar la conexión a la base de datos
Cerrar($Con);

} else {
    // Si la solicitud no fue enviada correctamente
    echo "Método de solicitud no válido.";
}

?>
```