



2ºDAW - Módulo de Desarrollo de Aplicaciones Entorno Servidor

Nombre Alumno:	Fecha: 21-Nov-20245
Evaluación de RA2, RA5, RA6	Docente: Sarah González

Nota:

- Se permite el uso de todos los ejercicios realizados en clase por Visual Studio Code. Reutilización.
- El examen en su totalidad debe ser distinto al resto de compañeros.
- Anulación examen:
 - Si hay 2 ó más iguales ó parcialmente iguales la evaluación de todos ellos será de 0; por copia o plagio. Anulación del examen.
 - No se permite el uso de chatGPT. Anulación del examen.
 - No se permite el uso de documentación, ni consultas por Google, PHP, etc
En caso de anulación del examen.
 - Tendrá pérdida de los RA evaluados en este examen.
 - Directo a Ordinaria con los RA evaluados en este examen.

ENTREGABLE:

Al finalizar todos los ejercicios del examen: ApellidoAlumno.ZIP ej:gonzalez.zip

Realizar Capturas de Pantalla - pueden ser tipo jpg o en un word/writer.
Separar cada ejercicio por carpeta. Por ejemplo: /sesiones y /crud

1. Captura de pantalla del localhost	
2. Captura de pantalla del Visual SC con el contenido del proyecto y subcarpetas de los ejercicios.	
3. Captura de pantallas de todas las salidas por ej: ejecuciones, cookies, BD, mensajes de errores, etc.	
4. Entrega de todos los archivos en el .zip / .7z / .rar con todos los HTML, PHP, TXT, pdf, doc o odt.	

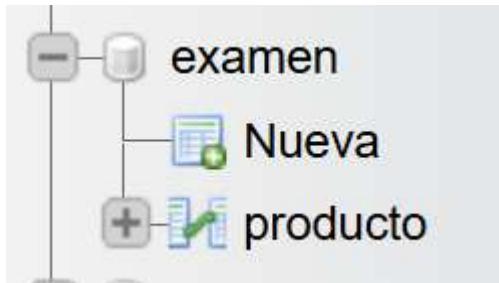


2ºDAW - Módulo de Desarrollo de Aplicaciones Entorno Servidor

1. Realizar una aplicación WEB con PHP-HTML-MySQL 4pts

Estructura /examen/

```
├── conexion.php  
├── index.php  
├── crear.php  
├── actualizar.php  
├── eliminar.php  
├── consultar.php  
└── borrar_tabla.php  
style.css
```



Requisitos técnicos

- Usar PHP con conexión a MySQL mediante PDO.
- Implementar formularios HTML para entrada de datos.
- Validar los datos del formulario y usar excepciones para los errores.
- Crear un menú de opciones CRUD.

Actividades:

1. Crear la base de datos llamada examen
2. Opciones del menú con formulario CRUD
 1. Crear una tabla llamada producto con los siguientes campos:
 - INT AUTO_INCREMENT PRIMARY KEY
 - VARCHAR(100)
 - DECIMAL(6,2)
 - INT
 2. Insertar productos diferentes desde un formulario HTML. (Mínimo 2 productos)
INSERT INTO producto (nombre, precio, cant)
VALUES ('\$nombre', \$precio, \$cant)"
 3. Actualizar datos de un producto existente.
UPDATE producto SET precio = \$precio, cant = \$cant WHERE id = \$id
 4. Consultar todos los productos y mostrarlos en una tabla HTML.
SELECT * FROM producto
 5. Eliminar un producto por ID.
DELETE FROM producto WHERE id = \$id
 6. Eliminar la tabla desde una opción del menú.
3. Validar los datos del formulario (precio > 0, cant ≥ 0)
4. **Manejo de excepciones si hay errores. Siempre informar al cliente.**



2ºDAW - Módulo de Desarrollo de Aplicaciones Entorno Servidor

2. Realizar una aplicación WEB con PHP-HTML-MySQL 6pts

Estructura /cine/

```
└── conexion.php  
└── login.php  
└── logout.php  
└── cookies.php  
└── cine.php  
└── reservar.php  
└── sesiones.php  
└── style.css
```

Requisitos técnicos

- Conexión a base de datos cine mediante PDO.
- Formulario de login/logout Validar usuario y contraseña
 - Si no existen, lanzar excepción
 - Si existen, iniciar sesión y contar sesiones

Aceptación de cookies

- Mostrar mensaje: "Debe aceptar las cookies para continuar", "Al aceptar, permite el uso de cookies según nuestra política de privacidad."
- Si acepta, guardar cookies y redirigir a la WEB de cine. **Hacer captura de cookies.**
- Si no acepta, no puede entrar a la WEB de cine

Actividades: Web de cine con:

1. Crear la base de datos llamada cine
2. Crear la tabla de usuarios
 - id INT AUTO_INCREMENT PRIMARY KEY,
usuario VARCHAR(50) NOT NULL,
contrasena VARCHAR(50) NOT NULL
 - con estos valores
 - admin, admin
 - cliente1, 1234
 - cliente2, 1234
3. Crear la tabla de peliculas
 - id INT AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(100) NOT NULL,
sillas_total INT NOT NULL,
sillas_reservadas INT DEFAULT 0
4. Crear la tabla de reservas
 - id INT PRIMARY KEY,
usuario_id INT NOT NULL,
pelicula_id INT NOT NULL,
sillas_reservadas INT NOT NULL,
FOREIGN KEY (usuario_id) REFERENCES usuarios(id),
FOREIGN KEY (pelicula_id) REFERENCES peliculas(id)
5. Formulario de compra de tickets con: nombre de la pelicula, total de sillas disponibles y sillas que quiere reservar (Todo debe estar validado, numerico mayor de 0, etc)



Lógica de reserva

1. Validar que haya sillas disponibles
2. Insertar reserva en tabla reservas
3. Actualizar sillas_reservadas en tabla peliculas
4. Mostrar:
 - Si se ha podido reservar, confirmar el nombre de la pelicula, sillas reservadas
 - Y las sillas que quedan libres.
 - Número total de sesiones iniciadas
5. Al finalizar se debe cerrar sesión; Destruir sesión y Redirigir al login.