

Práctica 3 — Sistema de reservas de aulas o recursos

1. Objetivo de la práctica

Desarrollar un sistema web que permita gestionar reservas de aulas y recursos (proyectores, salas, laboratorios, etc.) dentro de un centro educativo. El sistema debe incluir control de acceso por roles, gestión de recursos y reservas y generar un calendario semanal.

El resultado debe ser una aplicación funcional, clara y bien estructurada.

2. Descripción general de la aplicación

La aplicación debe incluir:

- Un sistema de login con roles diferenciados.
- Un panel principal con acceso a recursos y reservas.
- Un CRUD de recursos (según rol).
- Un sistema de reservas con comprobación de solapamientos.
- Una vista semanal.

3. Requisitos funcionales obligatorios

3.1. Autenticación y roles: el sistema debe tener, como mínimo, los siguientes roles:

- **Director**
 - Permisos máximos.
 - CRUD completo de recursos.
 - Crear, listar y eliminar cualquier reserva (propia o ajena).
 - Gestionar usuarios.
- **Técnico** (rol intermedio)
 - Puede ver todas las reservas.
 - Cancelar reservas.
 - Puede gestionar recursos (alta, baja, modificación).
- **Profesor**
 - Puede ver recursos.
 - Puede crear reservas y además ver sus propias reservas.
 - Puede cancelar sus propias reservas.
- **Alumno**
 - Puede ver disponibilidad de los recursos.
 - Puede realizar reservas limitadas, solo de aulas, y cancelarlas.

Requisitos generales:

- Formulario de login con usuario y contraseña.
- Comprobación de credenciales contra la base de datos.
- Uso de sesiones para mantener la sesión iniciada.
- Páginas protegidas según rol.
- Página de logout que destruya la sesión.

3.2. Gestión de recursos (CRUD completo)

Director y técnico. Cada recurso debe tener:

- Nombre
- Tipo (aula, tablets, sala, laboratorio, carro de portátiles...)
- Capacidad (si aplica)
- Estado (activo/inactivo)

Los usuarios con permisos podrán:

- Crear nuevos recursos
- Editar recursos existentes
- Eliminar recursos
- Listar todos los recursos en una tabla HTML

3.3. Gestión de reservas

Cada reserva debe tener:

- Usuario que la realiza
- Recurso reservado
- Fecha
- Franja horaria (por ejemplo, 1 hora)
- Estado (activa, cancelada)

El sistema debe permitir:

- Crear reservas comprobando que no haya solapamientos.
- Cancelar reservas propias (profesor y alumno).
- Ver todas las reservas (director y técnico).
- Listar reservas por día, por recurso o por usuario.

3.4. Comprobación de solapamientos

Al crear una reserva:

- El sistema debe comprobar si el recurso ya está reservado en esa fecha y franja horaria.
- Si existe solapamiento, debe mostrarse un mensaje de error y no permitir la reserva.

3.5. Posibles estructuras de almacenamiento (tienen libertad si consideran otras formas de estructuración)

A) Calendario semanal (array multidimensional)

Debes generar una vista semanal:

- Días: lunes → domingo
- Franjas horarias: por ejemplo, 08:00 – 20:00
- Representación mediante un array multidimensional:

`$calendario[dia][hora] = reserva / null;`

- Mostrarlo en una tabla HTML tipo calendario.

B) Array de reservas

Al listar reservas:

- Cargar todas las reservas desde la BD
- Crear un array de objetos Reserva
- Permitir filtrar por recurso, usuario, fecha o estado
- Permitir ordenar por fecha o franja horaria

3.6. Programación orientada a objetos. Debes crear, como mínimo, las clases:

Clase Usuario

- Propiedades básicas (id, nombre, rol)
- Constructor
- Métodos getters/setters

Clase Recurso

- Propiedades: nombre, tipo, capacidad, estado
- Constructor
- Métodos getters/setters

Clase Reserva

- Propiedades: usuario, recurso, fecha, hora, estado
- Constructor
- Método `solapaCon(otraReserva)` → true si hay solapamiento
- Métodos getters/setters

3.7. Validación de formularios

- Nombre del recurso obligatorio
- Tipo válido (lista predefinida)
- Fecha válida y no anterior al día actual
- Franja horaria válida
- No permitir reservas duplicadas o solapadas
- Estado válido (lista predefinida)

Se pueden usar expresiones regulares donde sea adecuado.

4. Extras opcionales

- Notificación mediante cookie cuando el usuario tiene una reserva próxima
- Filtro avanzado por tipo de recurso, fecha y usuario
- Límite de reservas por usuario y día
- Vista mensual del calendario
- Recursos con imagen (por ejemplo, foto del aula)