

PROYECTO WEB UF1845: “ Diseño de una Aplicación Web del lado del Servidor”

APARTADO I: “ Diseño e Implementación de la Base de Datos” Método Entidad/Relación

0. Definir los **objetivos generales** de la Aplicación

1. Explicación del proceso paso a paso

1. **Identificación de entidades y atributos:** Primero, definimos todas las entidades y sus atributos básicos, asegurándonos de incluir claves primarias únicas para cada tabla.
2. **Definición de relaciones:** Luego, determinamos las relaciones entre las entidades basadas en los requisitos especificados.
3. **Establecimiento de cardinalidades:** Determinamos las cardinalidades (uno a muchos, muchos a muchos) para cada relación.
4. **Creación de tablas SQL:** Finalmente, escribimos las declaraciones SQL para crear las tablas, incluyendo claves primarias y foráneas para implementar las relaciones definidas.

2. Diseño del diagrama ER:

- Identificación de las entidades y sus atributos.
- Definición de las relaciones entre las entidades.
- Establecimiento de las cardinalidades de las relaciones.

Entidades iniciales y sus atributos o campos, teniendo en cuenta los Objetivos generales:

- **Monitores** :

(id_monitores,dni,nombre,telefono,e-mail,foto,curriculo);

- **Usuarios:**

(id_usuario,dni,nombre,telefono,e-mail,foto,comentario);

- **Equipamientos** :

(id_equipamiento,equipamiento,tipo,precio,comentario,imagen,fecha_inicio,duracion);

- **Actividades** :

(id_actividad,actividad,tipo,precio,lugar,imagen,comentario,fecha_inicio,duracion);

- **Cursos:**

(id_curso,curso,tipo,fecha_inicio,precio,duracion,imagen,descripcion);

- **Alojamiento** :

(id_alojamiento,nombre,tipo,direccion,personas,precio,descripcion,comentarios).

3. Relaciones entre las entidades (cardinalidad entre entidades):

- ✓ **Un suario** puede tener **varios monitores** y Un profesor puede tener varios usuarios;
- ✓ **Un usuario** puede alquilar **1 alojamiento** o no alquilar ningUn alojamiento y Un alojamiento puede albergar a varios usuarios o estar vacio;
- ✓ **Un usuario** puede tener **varios cursos** y Un curso puede tener varios usuarios o no tener ninguno;
- ✓ Un usuario puede alquilar varios equipamientos o no alquilar ninguno y Un equipamiento solo

pertenece a Un usuario;

- ✓ Un usuario puede contratar al menos 1 o mas actividades y una actividad puede tener varios usuarios;
- ✓ Un monitor puede tener varias actividades y 1 actividad tiene solo Un monitor;
- ✓ una actividad puede tener Un equipamiento o no tener equipamiento y Un equipamiento puede tener varias actividades.

4. **Generación de las tablas** de la base de datos (entidades y las tablas de relación entre entidades):
Usando <https://dbdiagram.io/home> y <https://app.diagrams.net/> para los **diagramas de flujo** y la relación entre tablas.

5. **Traducir a código SQL** código sql a partir de las tablas generadas

6. **Implementar el código sql** en el servidor SQL usando phpmyadmin

7. **Documentar todo el proceso** de diseño de la base de datos en tu aplicación de actividades de clase

Entidades y sus atributo

- **Monitores**

- id_monitores (PK)
- dni
- nombre
- telefono
- e-mail
- foto
- curriculum

- **Usuarios**

- id_usuario (PK)
- dni
- nombre
- telefono
- e-mail
- foto
- comentario

- **Equipamientos**

- id_equipamiento (PK)
- equipamiento
- tipo
- precio
- comentario
- imagen
- fecha_inicio
- duracion

- **Actividades**

- id_actividad (PK)
- actividad
- tipo
- precio
- lugar
- imagen
- comentario
- fecha_inicio
- duracion

- **Cursos**

- id_curso (PK)
- curso
- tipo
- fecha_inicio
- precio
- duracion
- imagen
- descripcion

- **Alojamiento**

- id_alojamiento (PK)
- nombre
- tipo
- direccion
- personas
- precio
- descripcion
- comentarios

tablas Generadas

Tabla: Monitores

Campo	Tipo	Descripción
id_monitores	INT	Clave primaria, autoincremental
dni	VARCHAR(10)	Documento Nacional de Identidad
nombre	VARCHAR(50)	Nombre del monitor
telefono	VARCHAR(15)	Teléfono del monitor
email	VARCHAR(50)	Correo electrónico del monitor
foto	VARCHAR(255)	URL de la foto del monitor
curriculo	TEXT	Currículum del monitor

Tabla: Usuarios

Campo	Tipo	Descripción
id_usuario	INT	Clave primaria, autoincremental
dni	VARCHAR(10)	Documento Nacional de Identidad
nombre	VARCHAR(50)	Nombre del usuario
telefono	VARCHAR(15)	Teléfono del usuario
email	VARCHAR(50)	Correo electrónico del usuario
foto	VARCHAR(255)	URL de la foto del usuario
comentario	TEXT	Comentarios del usuario

Tabla: Equipamientos

Campo	Tipo	Descripción
id equipamiento	INT	Clave primaria, autoincremental
equipamiento	VARCHAR(50)	Nombre del equipamiento
tipo	VARCHAR(50)	Tipo de equipamiento
precio	DECIMAL(10,2)	Precio del equipamiento
comentario	TEXT	Comentarios sobre el equipamiento
imagen	VARCHAR(255)	URL de la imagen del equipamiento
fecha_inicio	DATE	Fecha de inicio del alquiler
duracion	INT	Duración del alquiler en días
id_usuario	INT	Clave foránea, usuario que alquila

Tabla: Actividades

Campo	Tipo	Descripción
id_actividad	INT	Clave primaria, autoincremental
actividad	VARCHAR(50)	Nombre de la actividad
tipo	VARCHAR(50)	Tipo de actividad
precio	DECIMAL(10,2)	Precio de la actividad
lugar	VARCHAR(100)	Lugar donde se realiza la actividad
imagen	VARCHAR(255)	URL de la imagen de la actividad
comentario	TEXT	Comentarios sobre la actividad
fecha_inicio	DATE	Fecha de inicio de la actividad
duracion	INT	Duración de la actividad en días
id_monitores	INT	Clave foránea, monitor que supervisa
id equipamiento	INT	Clave foránea, equipamiento usado

Tabla: Alojamiento

Tabla: Cursos

Campo	Tipo	Descripción
id_curso	INT	Clave primaria, autoincremental
curso	VARCHAR(50)	Nombre del curso
tipo	VARCHAR(50)	Tipo de curso
fecha_inicio	DATE	Fecha de inicio del curso
precio	DECIMAL(10,2)	Precio del curso
duracion	INT	Duración del curso en días
imagen	VARCHAR(255)	URL de la imagen del curso
descripcion	TEXT	Descripción del curso
id_usuario	INT	Clave foránea, usuario que alquila

Tabla intermedia: Usuarios - Actividades

Campo	Tipo	Descripción
id_usuario	INT	Clave foránea, usuario
id_actividad	INT	Clave foránea, actividad
PRIMARY KEY (id_usuario, id_actividad)		

tablas en SQL

Paso 2: Definición de las tablas en SQL

-- Tabla Monitores

```
CREATE TABLE monitores (  
  id_monitores INT AUTO_INCREMENT PRIMARY KEY,  
  dni VARCHAR(10) NOT NULL,  
  nombre VARCHAR(50) NOT NULL,  
  telefono VARCHAR(15),  
  email VARCHAR(50),  
  foto VARCHAR(255),  
  curriculum TEXT  
);
```

-- Tabla Usuarios

```
CREATE TABLE usuarios (  
  id_usuario INT AUTO_INCREMENT PRIMARY KEY,  
  dni VARCHAR(10) NOT NULL,  
  nombre VARCHAR(50) NOT NULL,  
  telefono VARCHAR(15),  
  email VARCHAR(50),  
  foto VARCHAR(255),  
  comentario TEXT  
);
```

-- Tabla Equipamientos

```
CREATE TABLE equipamientos (  
  id_equipamiento INT AUTO_INCREMENT PRIMARY KEY,  
  equipamiento VARCHAR(50) NOT NULL,  
  tipo VARCHAR(50),  
  precio DECIMAL(10,2),  
  comentario TEXT,  
  imagen VARCHAR(255),  
  fecha_inicio DATE,  
  duracion INT,  
  id_usuario INT,  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)  
);
```

-- Tabla Actividades

```
CREATE TABLE actividades (  
  id_actividad INT AUTO_INCREMENT PRIMARY KEY,  
  actividad VARCHAR(50) NOT NULL,  
  tipo VARCHAR(50),  
  precio DECIMAL(10,2),  
  lugar VARCHAR(100),  
  imagen VARCHAR(255),  
  comentario TEXT,  
  fecha_inicio DATE,  
  duracion INT,  
  id_monitores INT,  
  id_equipamiento INT,  
  FOREIGN KEY (id_monitores) REFERENCES monitores(id_monitores),  
  FOREIGN KEY (id_equipamiento) REFERENCES equipamientos(id_equipamiento)  
);
```

-- Tabla Cursos

```
CREATE TABLE cursos (  
  id_curso INT AUTO_INCREMENT PRIMARY KEY,  
  curso VARCHAR(50) NOT NULL,  
  tipo VARCHAR(50),  
  fecha_inicio DATE,  
  precio DECIMAL(10,2),  
  duracion INT,  
  imagen VARCHAR(255),  
  descripcion TEXT  
);
```

-- Tabla Alojamiento

```
CREATE TABLE alojamiento (  
  id_alojamiento INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(50) NOT NULL,  
  tipo VARCHAR(50),  
  direccion VARCHAR(100),  
  personas INT,  
  precio DECIMAL(10,2),  
  descripcion TEXT,  
  comentarios TEXT,  
  id_usuario INT,  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario)  
);
```

-- Tabla intermedia Usuarios - Monitores

```
CREATE TABLE usuarios_monitores (  
  id_usuario INT,  
  id_monitores INT,  
  PRIMARY KEY (id_usuario, id_monitores),  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),  
  FOREIGN KEY (id_monitores) REFERENCES monitores(id_monitores)  
);
```

-- Tabla intermedia Usuarios - Cursos

```
CREATE TABLE usuarios_cursos (  
  id_usuario INT,  
  id_curso INT,  
  PRIMARY KEY (id_usuario, id_curso),  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),  
  FOREIGN KEY (id_curso) REFERENCES cursos(id_curso)  
);
```

-- Tabla intermedia Usuarios - Actividades

```
CREATE TABLE usuarios_actividades (  
  id_usuario INT,  
  id_actividad INT,  
  PRIMARY KEY (id_usuario, id_actividad),  
  FOREIGN KEY (id_usuario) REFERENCES usuarios(id_usuario),  
  FOREIGN KEY (id_actividad) REFERENCES actividades(id_actividad)  
);
```

APARTADO II: “Construcción de la Aplicación”

DISEÑO de tu Aplicación Web a partir de la base de datos generada en el apartado anterior; Teniendo en cuenta los Principios generales del Diseño Web (archivo adjunto)

1. Definir y Construir una arquitectura inicial básica para la Aplicación: Sistema de carpetas y subcarpetas para implementar los elementos de la aplicación (css, js, php, img, videos, ...) de forma gráfica, con diagramas.

2. Generar un CRUD con alguna tabla de la base de datos

3. Construir un login para que los usuarios y monitores puedan loguearse para ver y editar sus datos

4. Construir los archivos necesarios para generar las páginas de la aplicación para el público visitante, para usuarios de la aplicación y los administradores

4. Diseñar una interfaz administrativa para que el administrador pueda editar todos los elementos de la aplicación

5. Diseñar una interfaz que haga accesible los elementos de la aplicación a los visitantes de la aplicación en la que puedan logearse los usuarios inscritos

6. Tener en cuenta la Usabilidad y accesibilidad según sus Principios generales, generando un informe de los elementos tenidos en cuenta para ello

7. La aplicación debe ser “responsive”, adaptable a los distintos tamaños de pantallas.