



# **TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB**

**Departamento de Informática**

**PROYECTO**

**ENERGY**

**Manual Técnico**

Autor: Miguel Folguera Rivera

Curso Académico: 24/25

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Energy

JUNTA DE EXTREMADURA

Consejería de Educación y Empleo



## INDICE

1. Introducción.....	3
2. Arquitectura de la aplicación.....	3
2.1 Frontend.....	3
2.2 Backend.....	3
3. Documentación técnica.....	4
3.1. Análisis.....	4
3.2. Desarrollo.....	7
3.3. Pruebas realizadas.....	7
4. Proceso de despliegue.....	10
5. Propuesta de mejoras.....	11
6. Bibliografía.....	11

## 1. Introducción

Energy, es una aplicación web para la gestión integral de un gimnasio y sus actividades deportivas.

Lleva la gestión completa de un gimnasio que ofrece diversas actividades deportivas, tales como zumba, tenis, pádel, natación, baloncesto, aquagym, atletismo, entre otras.

La aplicación permite a los usuarios registrarse, consultar las actividades disponibles, inscribirse en ellas controlando el número de plazas, visualizar horarios, y gestionar su perfil. A su vez, los administradores del gimnasio podrán gestionar usuarios, actividades, horarios, entrenadores, gerentes y salas.

El ámbito de aplicación será **comercial**, ya que está pensado como una herramienta útil para gimnasios o centros deportivos que desean gestionar sus actividades de forma eficiente, moderna y centralizada.

## 2. Arquitectura de la aplicación

### 2.1 Frontend

#### 2.1.1 Tecnologías usadas

- HTML5: es el estándar web más reciente y es compatible con la gran mayoría de navegadores
- CSS3: La aplicación cuenta con un archivo principal de estilos complementado con Bootstrap
- Bootstrap : Framework CSS que facilita la creación de una interfaz responsiva y proporciona estilos prediseñados que mejoran la usabilidad de la aplicación en diferentes tamaños de pantalla.
- FontAwesome: Biblioteca de iconos que permite usar una amplia gama de iconos en diferentes secciones de la aplicación, mejorando el estilo, la usabilidad e interfaz visual sin necesidad de diseñar gráficos adicionales.
- JavaScript: Usado para dotar ciertas partes de la aplicación de un mayor dinamismo (modales y comprobaciones de los campos).

#### 2.1.2 Entorno de desarrollo

He usado principalmente Netbeans aunque tambien se ha usado visual estudio code cuando no tenia el otro.

### 2.2 Backend

## 2.2.1 Tecnologías usadas

- Workbench para trabajar con MySQL.
- Java EE como lenguaje de programación

## 2.1.2 Entorno de desarrollo: Netbeans y Visual Code.

Otros recursos: GitHub (control de versiones).

# 3. Documentación técnica

## 3.1. Análisis

### 3.1.1. Requisitos funcionales

- Registro y login de usuarios.
- Consulta de actividades deportivas, con horarios y monitores.
- Inscripción a actividades con control de plazas.
- Panel de administración para crear/editar actividades, asignar entrenadores, gestionar horarios, salas, gerentes y control de pagos.
- Control de asistencia a actividades.
- Gestión de perfiles de usuario.
- Envío de correos para recuperar contraseñas, novedades y confirmar inscripciones.

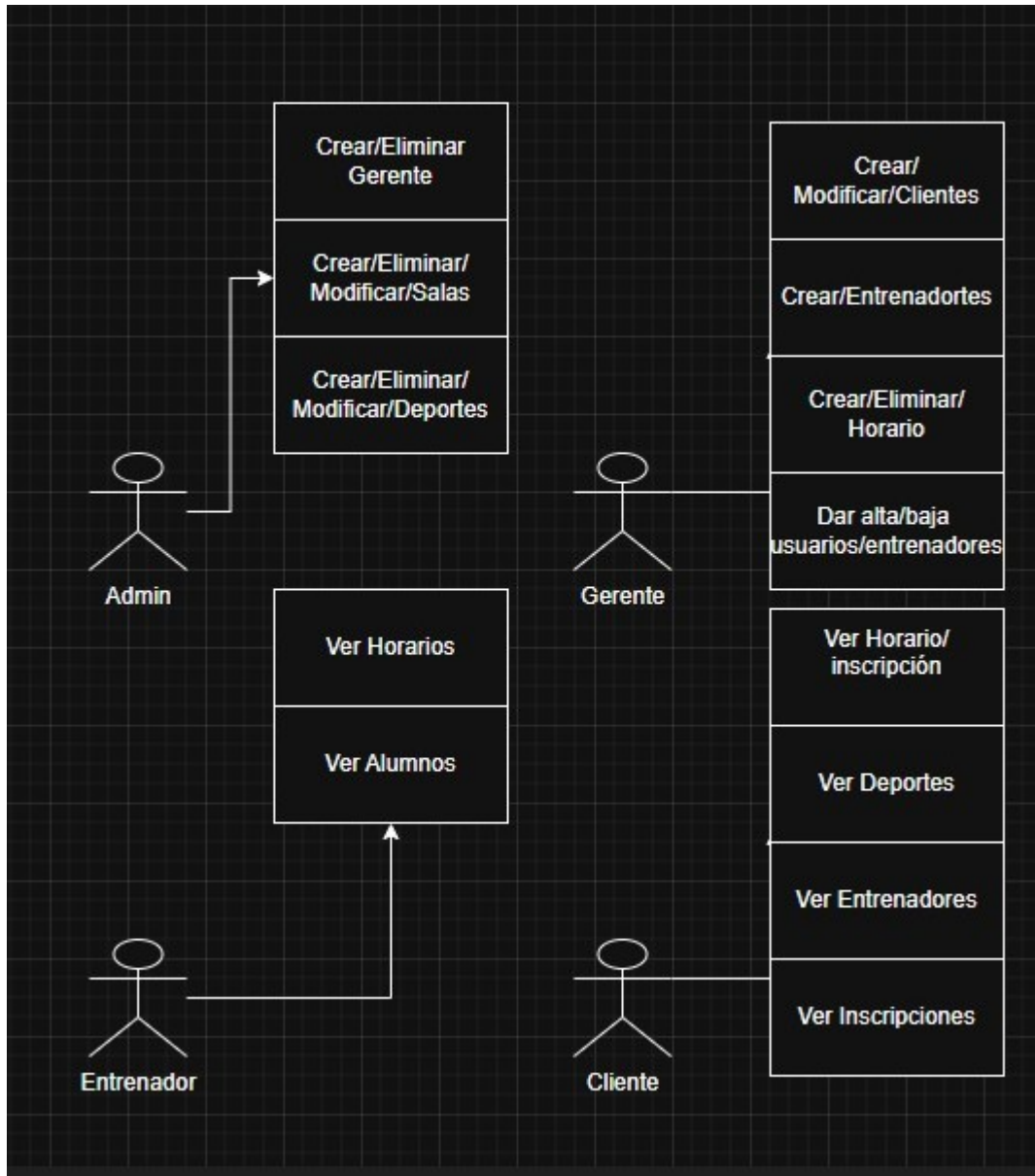
### 3.1.2. Requisitos no funcionales

- Alta disponibilidad. La aplicación estará disponible 24 horas 7 días a la semana, para que los usuarios y el resto de los roles puedan acceder a ella en cualquier momento.
- Control de errores.
- Interfaz sencilla e intuitiva
- Seguridad en el acceso mediante autenticación (login y contraseña) y roles (usuario, administrador, gerente, entrenador).

### 3.1.3. Ciclo de vida

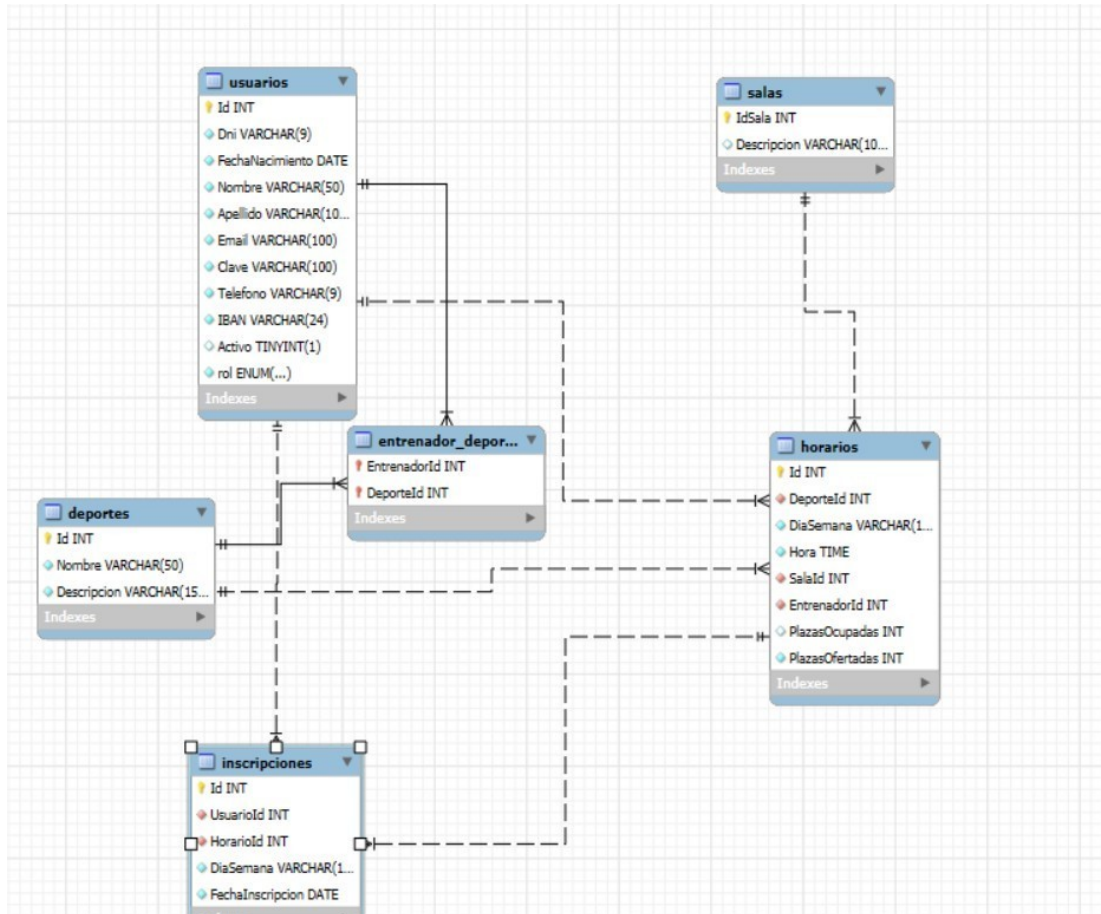
Se ha optado por un modelo en cascada con retroalimentación ya que permite avanzar de forma estructurada a través de cada fase del desarrollo con la posibilidad de retroceder y corregir cualquier error o mejora que surja en las etapas posteriores.

## 3.1.4. Casos de uso



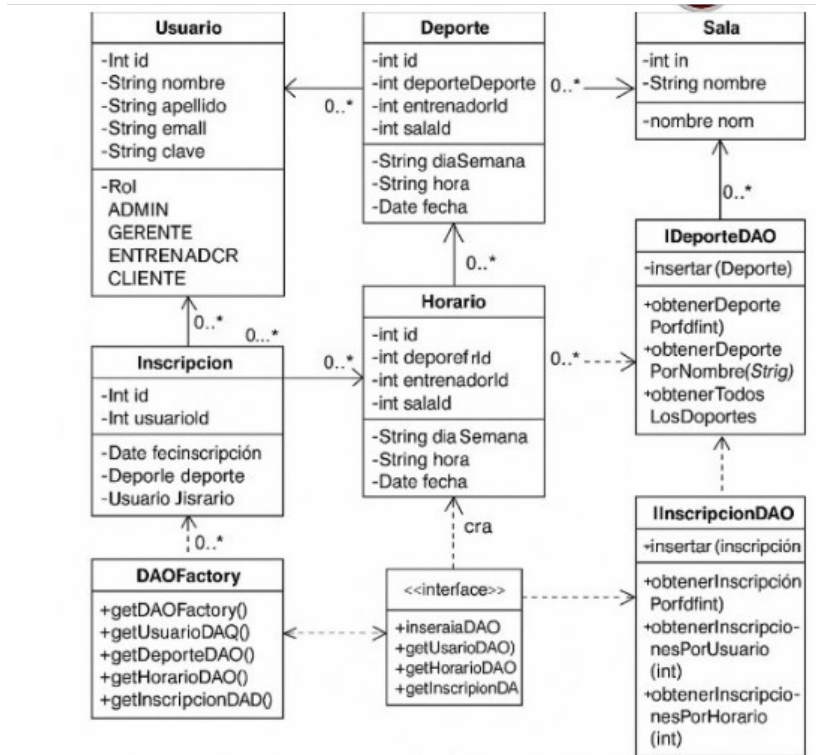
## 3.1.5 Diseño de datos.

Se incluye el Diagrama relacional correspondiente al Modelo lógico de la BD.



## 3.2. Desarrollo

### Diagrama de clases



## 3.3. Pruebas realizadas

Se han realizado pruebas de caja blanca como inscribir usuarios, ver horarios , registrar usuario comprobando que se han hecho las inserciones oportunas

- Componente (clase, método,...) al que se le han realizado.
- Juego de datos de entrada.
- Resultado de salida.
- Herramientas de pruebas usadas.

- Prueba 1: Login exitoso como ADMIN

Entradas:

- email: admin@correo.com
- password: a (la contraseña correcta, encriptada coincide)
- El email existe
- Rol: ADMIN

Ruta ejecutada:

1 → 2 → 3 (email existe) → 4 (usuario encontrado) → 5 (contraseña coincide)  
→ 6 (rol ADMIN o ENTRENADOR)

Resultado esperado:

- Sesión creada: usuarioLogueado
- Redirección a indexAdmin.jsp

Estado: Correcto

- Prueba 2: Login exitoso como CLIENTE

Entradas:

- email: a@correo.com
- password: a correcta
- Rol: CLIENTE

Ruta ejecutada:

1 → 2 → 3 (email existe) → 4 (usuario encontrado) → 5 (contraseña coincide) → 6 (rol CLIENTE)

Resultado esperado:

- Sesión creada
- Redirección a index.jsp
- Estado: Correcto

- Prueba 3: Contraseña incorrecta

Entradas:

- email: a@correo.com
- password:ds incorrecta
- El email existe
- Contraseña encriptada no coincide

Ruta ejecutada:

1 → 2 → 3 (email existe) → 4 (usuario encontrado) → 5 (contraseña NO coincide)

Resultado esperado:

- No se establece sesión
- Atributo error con mensaje: *"La contraseña es incorrecta."*



# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Energy

JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

- Redirección a index.jsp

Estado: Correcta

- Prueba 4: Email no registrado

Entradas:

- email: noexiste@correo.com
- password: cualquier valor

Ruta ejecutada:

1 → 2 → 3 (email NO existe)

Resultado esperado:

- No se consulta contraseña
- Atributo error con mensaje: *"El email es incorrecto."*
- Redirección a index.jsp

Estado: Correcta

- Prueba 5: Sin parámetro "login" en request

Entradas:

- No se incluye el parámetro "login" en el request

Ruta ejecutada:

Se omite todo el bloque de login.

Resultado esperado:

- No se ejecuta lógica de autenticación
- No cambia URL, no se setea ningún atributo

Estado: Correcto

RUTAS LÓGICAS TOTALES CUBIERTAS

Camino	Descripción	Estado
C1	Email válido, contraseña válida, rol admin	Correcto
C2	Email válido, contraseña válida, rol cliente	Correcto
C4	Email inválido	Correcto
C5	Falta parámetro "login"	Correcto

## 4. Proceso de despliegue

### 4.1 Desde un hosting (producción)

#### 1. Archivo .war generado:

- Debe ser exportado desde el entorno de desarrollo (por ejemplo, NetBeans o Eclipse).
- Contiene todo el proyecto empaquetado (servlets, JSPs, clases, librerías).

#### 2. Servidor de aplicaciones (Apache Tomcat o compatible):

- El hosting debe tener Tomcat o GlassFish configurado para aceptar despliegue de aplicaciones Java EE.

#### 3. Subida del .war al servidor:

- Se puede hacer mediante el **panel de administración de Tomcat** (Manager App) o vía FTP/SFTP.
- Al subirlo, el servidor lo desplegará automáticamente y generará una URL de acceso como:

CopiarEditar

<https://miapp.hosting.com/NombreMiProyecto>

Se ha subido al servidor del IES Albarregas.

<http://informatica.iesalbarregas.com:7001/Energy/>

### 4.2 Local (desarrollo o pruebas)

#### 1. Apache Tomcat instalado localmente:

- Debes descargarlo e instalarlo desde <https://tomcat.apache.org/>
- Configura el puerto (por defecto: 8080) y define tu proyecto como aplicación web.

#### 2. IDE compatible (NetBeans, Eclipse, IntelliJ):

- Asegúrate de tener el plugin de Java EE/Web instalado para trabajar con netbeans

#### 3. Configuración del servidor en el IDE:

- Desde NetBeans: Servicios → Servidores → Añadir Servidor → Apache Tomcat

→ Configurar ruta.

- Agregar tu proyecto a Tomcat y ejecútalo.

4. Base de datos local o remota:

- Verificar que la base de datos (MySQL) esté accesible con la configuración adecuada.

5. Compilación sin errores:

- Toda la aplicación debe compilar correctamente, incluyendo:
  - Clases DAO
  - Servlets
  - JSPs
  - Archivos de configuración (web.xml, context.xml )

## 5. Propuesta de mejoras

- Hacerlo completamente responsivo, hay que mejorar esa parte.
- Que se puedan cambiar los entrenadores conservando sus deportes

## 6. Bibliografía

Se ha consultado frecuentemente páginas web para el desarrollo.