



I E S A L B A R R E G A S

TÉCNICO SUPERIOR EN DESARROLLO DE  
APLICACIONES WEB

Departamento de Informática

PROYECTO



PET CLINIC

**Manual Técnico**

Autor/es: **Jesús Melara Martín**

Curso Académico: **2024/2025**

## Índice

- 1. Introducción**
- 2. Arquitectura de la aplicación**
  - 2.1. Frontend**
    - 2.1.1. Tecnologías usadas**
    - 2.1.2. Entorno de desarrollo**
  - 2.2. Backend**
    - 2.2.1. Tecnologías usadas**
    - 2.2.2. Entorno de desarrollo**
- 3. Documentación técnica**
  - 3.1. Análisis**
  - 3.2. Desarrollo**
  - 3.3. Pruebas realizadas**
  - 3.4. Requisitos funcionales y no funcionales**
    - 3.4.1. Requisitos funcionales**
    - 3.4.2. Requisitos no funcionales**
  - 3.5. Campos de la base de datos**
- 4. Proceso de despliegue**
- 5. Propuesta de mejoras**
- 6. Bibliografía**

## 1. Introducción

Pet Clinic es una aplicación web para la gestión de una clínica veterinaria. Permite la administración de veterinarios, clientes, mascotas, servicios y citas, con funcionalidades de consulta, creación, edición y eliminación de registros. Esta aplicación tiene como objetivo facilitar la gestión diaria de la clínica mediante una interfaz intuitiva y un backend robusto.

El proyecto se ha desarrollado utilizando tecnologías estándar para aplicaciones web basadas en Java EE y JSP, buscando una arquitectura modular, escalable y mantenible.

## 2. Arquitectura de la aplicación

### 2.1. Frontend

#### 2.1.1. Tecnologías usadas

- JSP (Java Server Pages) para la generación dinámica de contenido HTML.
- HTML5 y CSS3 para la estructura y estilo visual.
- Bootstrap 5 para diseño responsive y componentes UI.
- JavaScript para validaciones y dinámicas en el navegador.
- JSTL (Java Standard Tag Library) para simplificar el uso de expresiones y lógica en las páginas JSP.
- Font Awesome / Icons para iconografía visual.

#### Justificación:

Se optó por JSP y JSTL dado que permiten una integración directa con el backend Java EE, evitando la complejidad de frameworks frontend modernos, ajustado a los requerimientos del proyecto académico. Bootstrap se eligió por su facilidad de uso y compatibilidad con múltiples dispositivos.

#### Versiones usadas:

- Bootstrap 5
- JSTL 1.2
- JavaScript ECMAScript 5+
- Servidor Tomcat 9 compatible con JSP 2.3

## 2.1.2. Entorno de desarrollo

- NetBeans IDE 19 como entorno principal para edición, compilación y despliegue local.
- Navegadores modernos (Chrome, Firefox) para pruebas funcionales.
- Servidor Apache Tomcat 9 para la ejecución de la aplicación web.
- Herramientas de gestión de versiones (Git, GitHub).

## 2.2. Backend

### 2.2.1. Tecnologías usadas

- Java EE 7 como plataforma de desarrollo backend.
- Servlets para controladores que manejan las peticiones HTTP.
- Hibernate JPA (Java Persistence API) para el acceso y gestión de datos en base de datos.
- MySQL 8 como sistema gestor de base de datos relacional.
- Gson y org.json para manipulación de JSON.
- DAO (Data Access Object) pattern para separar lógica de acceso a datos.
- Patrón MVC (Modelo-Vista-Controlador) para organización del código.
- Commons BeanUtils para utilidades con Java Beans.
- Maven como gestor de dependencias y construcción del proyecto.

#### Justificación:

Java EE e Hibernate se eligieron por su madurez, amplia documentación y porque el proyecto académico se centra en tecnologías estándar para aplicaciones web en Java. MySQL fue elegido por su estabilidad y compatibilidad con Hibernate. Maven facilita la gestión de dependencias y despliegue.

#### Versiones usadas:

- Java SE 7
- Java EE Web API 7.0
- Hibernate 5.4.10.Final
- MySQL 8.0

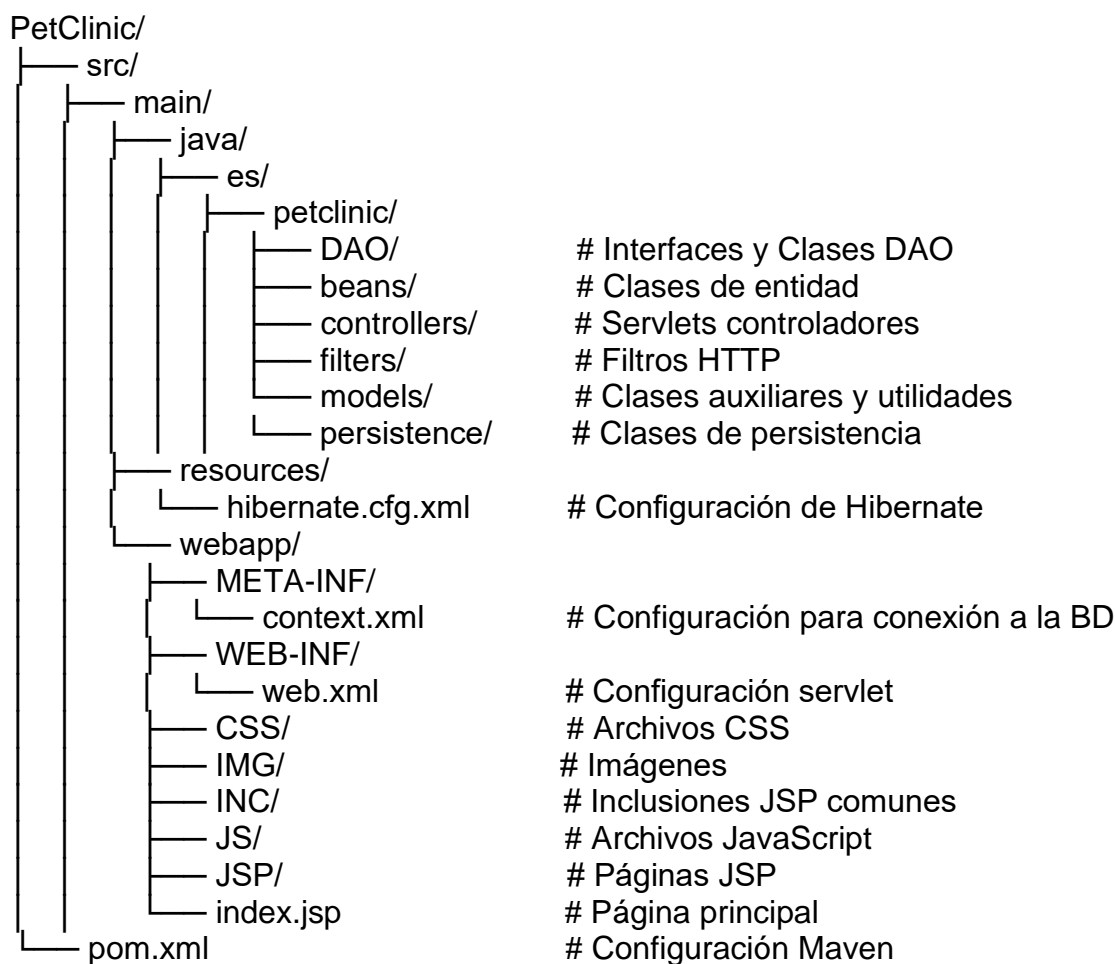
- Gson 2.8.9
- org.json 20231013
- Commons BeanUtils 1.9.4
- Maven 3.6+
- Apache Tomcat 9.0

## 2.2.2. Entorno de desarrollo

- Netbeans IDE 19 con plugins para Java EE y soporte para Maven.
- Base de datos MySQL 8.0 con consola y MySQL Workbench para gestión y consultas.
- Apache Tomcat 9 como servidor de aplicaciones para pruebas y despliegue.

## 2.3. Estructura del proyecto

A continuación se presenta la estructura de carpetas principal del proyecto:

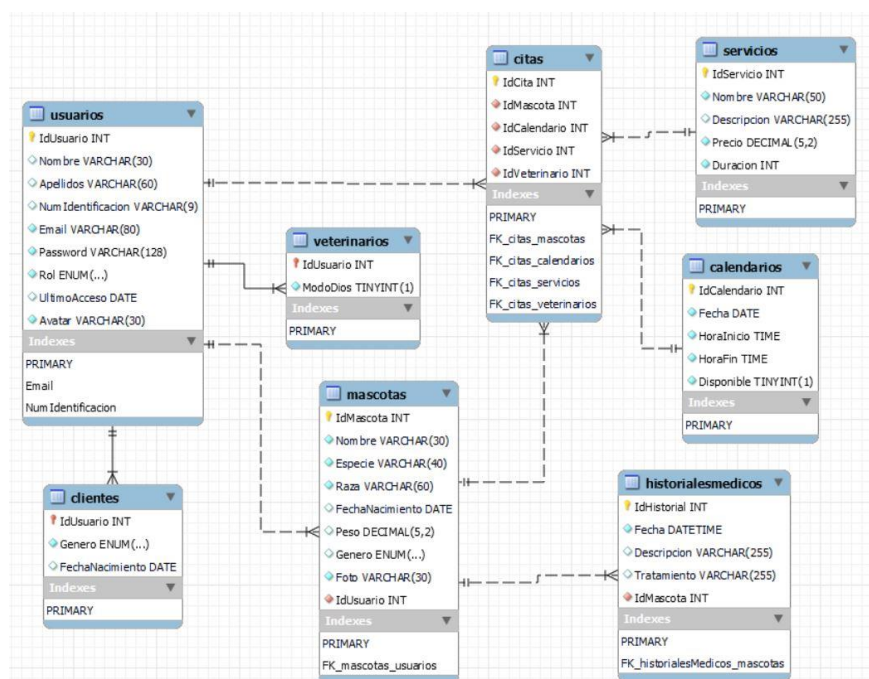


### 3.1. Análisis

```
graph LR; V((Veterinario)) --> V1[Iniciar sesión]; V --> V2[Ver citas]; V --> V3[Crear / Visualizar historiales médicos]; V --> V4[Visualizar clientes y mascotas]; C((Cliente)) --> C1[Iniciar sesión / Registrarse]; C --> C2[Editar perfil]; C --> C3[Crear / Editar / Eliminar / Visualizar mascotas]; C --> C4[Solicitar / Cancelar citas]; C --> C5[Visualizar historiales médicos]; C --> C6[Consultar servicios disponibles]; C --> C7[Visualizar infografías];
```

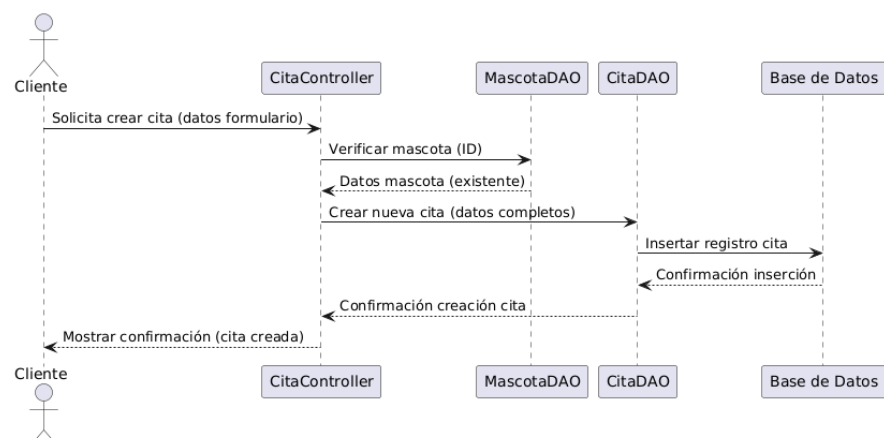
[illegible]

## Diagrama entidad relación:



## 3.2. Desarrollo

### Diagrama de secuencia (Solicitar cita):



## 3.3. Pruebas realizadas

### Prueba: Registro de usuario

- **Componente probado:**

RegistroController.doPost() y método de inserción en UsuarioDAO.  
insertOrUpdateUsuario(Usuario usuario)

- **Juego de datos de entrada:**

Datos de formulario de registro:

- Nombre: “Juan”
- Apellido: “Pérez”
- Email: [cliente1@gmail.com](mailto:cliente1@gmail.com)
- Contraseña: Cliente1\*
- Otros campos

- **Resultado esperado:**

- Usuario creado correctamente en la base de datos.
- Redirección a página del cliente.

- **Resultado obtenido:**

- Se insertó correctamente el cliente en la tabla usuarios y clientes.
- No hubo errores y el sistema me redirigió a la página del cliente.
- Se verificó en la base de datos la presencia del nuevo cliente.

- **Herramientas usadas:**

- Navegador (Chrome) para enviar formulario.
- MySQL Workbench para consulta de base de datos.
- Visualizar las consultas SQL generadas por Hibernate en la consola de NetBeans.

## 3.4. Requisitos funcionales y no funcionales

### 3.4.1. Requisitos funcionales

Los requisitos funcionales definen el comportamiento que debe ofrecer la aplicación para satisfacer las necesidades del usuario. En el caso de Pet Clinic, se identifican los siguientes:

- 1. El sistema debe permitir a los usuarios registrarse e iniciar sesión.
- 2. El sistema debe redirigir al usuario a una vista específica según su rol al iniciar sesión.
- 2. El sistema debe permitir al cliente editar sus datos personales y cambiar su contraseña desde su perfil.



- 3. El sistema debe permitir que un cliente registre una o varias mascotas.
- 4. El sistema debe permitir crear, consultar y cancelar citas veterinarias.
- 5. El sistema debe permitir al veterinario registrar y consultar historiales médicos.
- 6. El sistema debe validar formularios mediante JavaScript y AJAX para evitar recargar completas.

## 3.4.2. Requisitos no funcionales

Los requisitos no funcionales definen atributos de calidad del sistema como rendimiento, seguridad, o mantenibilidad:

- 1. El sistema debe seguir el patrón de arquitectura MVC.
- 2. La aplicación debe funcionar correctamente en navegadores modernos (Chrome, Firefox, etc.).
- 3. Las contraseñas deben ser almacenadas cifradas en la base de datos.
- 4. La base de datos debe ser relacional (MySQL) y garantizar la integridad referencial entre tablas.
- 5. El sistema debe ser desplegable en Apache Tomcat 9 mediante archivo .war.
- 6. El proyecto debe utilizar Maven para la gestión de dependencias.

## 3.5. Campos de la base de datos

A continuación se describen los campos de las tablas de la base de datos de la aplicación:

### Tabla usuarios:

- **IdUsuario.** Clave primaria. Identificador único del usuario.
- **Nombre.** Nombre del usuario.
- **Apellidos.** Apellidos del usuario.
- **NumIdentificacion.** Número de identificación de la persona.
- **Email.** Correo electrónico del usuario.
- **Password.** Contraseña cifrada del usuario.
- **Rol.** Rol del usuario (VETERINARIO, CLIENTE)
- **UltimoAcceso.** Fecha y hora del último inicio de sesión.
- **Avatar.** Nombre del archivo del avatar del usuario.

### Tabla veterinarios:

- **IdUsuario.** Clave primaria. Relación con la tabla usuarios.
- **ModoDios.** Campo booleano que indica si el veterinario tiene permisos elevados (admin).

### Tabla clientes:

- **IdUsuario.** Clave primaria. Relación con la tabla usuarios.
- **Genero.** Género del cliente (MUJER, HOMBRE, OTRO).
- **FechaNacimiento.** Fecha de nacimiento del cliente.

### Tabla mascotas:

- **IdMascota.** Clave primaria. Identificador único de la mascota.
- **Nombre.** Nombre de la mascota.
- **Especie.** Especie de la mascota.
- **Raza.** Raza específica dentro de la especie.
- **FechaNacimiento.** Fecha de nacimiento de la mascota.
- **Peso.** Peso de la mascota en kg.
- **Genero.** Género de la mascota (MACHO, HEMBRA).
- **Foto.** Nombre del archivo de la foto de la mascota.
- **IdUsuario.** Clave foránea que referencia al dueño (cliente) en usuarios.

# Proyecto “Desarrollo de Aplicaciones Web”

Título del Proyecto: Pet Clinic



JUNTA DE EXTREMADURA

Consejería de Educación y Empleo

## Tabla citas:

- **IdCita.** Clave primaria. Identificador único de la cita.
- **IdMascota.** Mascota que será atendida en la cita.
- **IdCalendario.** Bloque horario reservado para la cita.
- **IdServicio.** Servicio veterinario que se realizará.
- **IdVeterinario.** Veterinario asignado a la cita.

## Tabla servicios:

- **IdServicio.** Clave primaria. Identificador del servicio ofrecido.
- **Nombre.** Nombre del servicio.
- **Descripción.** Descripción del servicio.
- **Precio.** Precio del servicio.
- **Duracion.** Duración del servicio en minutos.

## Tabla calendarios:

- **IdCalendario.** Clave primaria. Identificador del bloque horario.
- **Fecha.** Fecha concreta del bloque horario.
- **Horainicio.** Hora de inicio del bloque.
- **HoraFin.** Hora de fin del bloque.
- **Disponible.** Indica si ese bloque está disponible (1 para sí, 0 para no).

## Tabla historialesmedicos:

- **IdHistorial.** Clave primaria. Identificador único del historial.
- **Fecha.** Fecha y hora en la que se realizó la atención médica.
- **Descripcion.** Descripción general de la revisión o diagnóstico.
- **Tratamiento.** Tratamiento prescrito o realizado.
- **IdMascota.** Clave foránea que indica a qué mascota pertenece el historial.

## 4. Proceso de despliegue

### Software necesario

- **Java SE Development Kit 8** (versión 1.8) — <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- **Apache Tomcat 9.0** — <https://tomcat.apache.org/download-90.cgi>
- **MySQL Server 8.0** — <https://dev.mysql.com/downloads/mysql/>
- **MySQL Workbench** (opcional) — <https://dev.mysql.com/downloads/workbench/>
- **Maven 3.6+** — <https://maven.apache.org/download.cgi>

### Proceso detallado

#### 1. Configurar base de datos:

- Crear base de datos petclinic en MySQL.

- Ejecutar script SQL incluido



para crear tablas y relaciones.

#### 2. Configurar Hibernate:

- Editar hibernate.cfg.xml para configurar la conexión entre Hibernate y la base de datos MySQL, así como las clases mapeadas para la persistencia.

#### 3. Construir proyecto:

- Ejecutar en Clean and Build para generar el archivo .war.

#### 4. Desplegar en Tomcat:

- Abrir el navegador.
- Insertar el .war generado en la sección:

Archivo WAR a desplegar

Seleccione archivo WAR a cargar  Ningún archivo seleccionado

en <http://informatica.iesalbarregas.com:7001/manager/html>

- En la parte superior de la página aparecerá nuestra aplicación subida

/PetClinic		Ninguno especificado	true	1	Expirar sesiones	sin trabajar > 30 minutos	Arrancar	Parar	Recargar	Replegar
					Expirar sesiones	sin trabajar > 30 minutos	Arrancar	Parar	Recargar	Replegar

Hacer clic en el nombre y directamente accedemos a <http://informatica.iesalbarregas.com:7001/PetClinic/>

## 5. Propuesta de mejoras

Futuras mejoras que me gustaría implementar en la aplicación:

- Migrar el frontend a un framework SPA (Single Page Application) como React o Angular significa cambiar la forma en que se construye la interfaz de usuario de la aplicación.
- Incluir paginación y más filtros mejor optimizados.
- Ofrecer más servicios.
- Cambiar / optimizar la forma de crear el calendario del veterinario.
- Integrar notificaciones (correo, SMS) para citas y recordatorios.
- Tienda online donde se vendan productos para el cuidado de los animales.
- Mejorar aún más el diseño gráfico de la página.

## 6. Bibliografía

Durante el desarrollo de la aplicación PetClinic se han consultado y utilizado diversas fuentes de información online, así como documentación oficial y recursos educativos:

- **Stack Overflow:** <https://stackoverflow.com/>
- **YouTube:** Vídeos tutoriales y explicativos sobre desarrollo web, Java EE, Hibernate y diseño frontend con Bootstrap.
- **Documentación oficial de Bootstrap:** <https://getbootstrap.com/>
- **Iconos usados:** <https://www.flaticon.es/>
- **Recursos web varios:** Diferentes blogs, foros y páginas técnicas especializadas consultadas para solucionar problemas concretos y ampliar conocimientos sobre tecnologías usadas.
- **Páginas de clínicas veterinarias de referencia:**
  - <https://www.kivet.com/>