



# MÓDULO PROYECTO

---

CFGS Desarrollo de Aplicaciones  
Multiplataforma  
Informática y Comunicaciones

---

## PETCare – Aplicación Web

**Tutor individual:** José María Rojo Zumel

**Tutor colectivo:** Cristina Silván Pardo

**Año:** 2023/2024

**Fecha de presentación:** 27/05/2024

**Nombre y Apellidos:** Paula Nunes Ranzani  
**Email:** paula.nunran@educa.jcyl.es



## Contenido

1	Identificación proyecto .....	4
2	Organización de la memoria .....	4
3	Descripción general del proyecto .....	6
3.1	Objetivos .....	6
3.2	Cuestiones metodológicas .....	7
3.3	Entorno de trabajo (tecnologías de desarrollo y herramientas).....	7
4	Descripción general del producto.....	8
4.1	Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.....	8
4.2	Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.....	9
4.3	Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha.....	9
5	Planificación y presupuesto .....	9
6	Documentación Técnica: análisis, diseño, implementación y pruebas. ....	10
6.1	Especificación de requisitos .....	10
6.2	Análisis del sistema .....	12
6.3	Diseño del sistema: .....	15
6.3.1	Diseño de la Base de Datos .....	15
6.3.2	Diseño de la Interfaz de usuario. ....	16
6.3.3	Diseño de la Aplicación. ....	17
6.4	Implementación: .....	18
6.4.1	Entorno de desarrollo. ....	18
6.4.2	Estructura del código. ....	18
6.4.3	Cuestiones de diseño e implementación reseñables.....	18
6.5	Pruebas.....	18



7	Manuales de usuario .....	19
7.1	Manual de usuario.....	19
7.2	Manual de instalación .....	56
8	Conclusiones y posibles ampliaciones .....	56
9	Bibliografía.....	57
10	Anexos .....	57



## 1 Identificación proyecto

Logotipo	
Nombre de la aplicación	Clínica PetCare
Desarrolladores	Paula Nunes Ranzani
Fecha inicio	13/04/2024
Fecha fin	22/05/2024
Tutor	José María Rojo
Versión de la aplicación	1.0

## 2 Organización de la memoria

1. Identificación del Proyecto: Descripción del proyecto, incluyendo logotipo, título, responsables, fechas y versión.
2. Organización de la Memoria: Estructura y organización de la memoria
3. Descripción General del Proyecto: Introducción al proyecto
  - 3.1 Objetivos: Define los objetivos generales y específicos que se pretende alcanzar con el proyecto.
  - 3.2 Cuestiones Metodológicas: Describe la metodología de desarrollo y las técnicas empleadas.
  - 3.3 Entorno de Trabajo: Tecnologías, herramientas y entornos de desarrollo utilizados.
4. Descripción General del Producto: Visión global del producto final, incluyendo funcionalidad.

4.1 Visión General del Sistema: Describe el alcance del sistema, sus funciones principales, los tipos de usuarios y las posibles interacciones con otros sistemas.

4.2 Descripción Breve de Métodos, Técnicas o Arquitecturas (m/t/a) Utilizadas: Resumen de los métodos, técnicas y arquitecturas aplicadas en el desarrollo.

4.3 Despliegue de la Aplicación indicando Plataforma Tecnológica, Instalación de la Aplicación y Puesta en Marcha: Explica el proceso de despliegue de la aplicación, las plataformas tecnológicas involucradas y los pasos para su instalación y puesta en marcha.

5. Planificación y Presupuesto: Presenta la planificación temporal del proyecto y el presupuesto estimado.

6. Documentación Técnica: Análisis, Diseño, Implementación y Pruebas: Incluye toda la documentación técnica del proyecto, desde los requisitos hasta las pruebas realizadas.

6.1 Especificación de Requisitos: Detalle de los requisitos funcionales y no funcionales del sistema.

6.2 Análisis del Sistema: Análisis detallado del sistema, incluyendo diagramas y modelos de datos.

6.3 Diseño del Sistema: Expone el diseño del sistema, abarcando la base de datos, la interfaz de usuario y la aplicación en su conjunto.

6.3.1 Diseño de la Base de Datos: Descripción del esquema de la base de datos y su estructura.

6.3.2 Diseño de la Interfaz de Usuario: Presentación del diseño de la interfaz de usuario.

6.3.3 Diseño de la Aplicación: Estructura general del diseño de la aplicación, abarcando sus componentes principales.

6.4 Implementación: Detalles sobre la implementación del sistema, incluyendo el entorno de desarrollo, la estructura del código y aspectos de diseño e implementación importantes.

6.4.1 Entorno de Desarrollo: Descripción del entorno de desarrollo utilizado para implementar la aplicación.

6.4.2 Estructura del Código: Explicación de cómo está organizado el código del proyecto.

6.4.3 Cuestiones de Diseño e Implementación Reseñables: Discusión sobre aspectos notables del diseño y la implementación que merecen ser destacados.

6.5 Pruebas: Metodologías de pruebas aplicadas, tipos de pruebas realizadas y resultados obtenidos.

7. Manuales de Usuario: Instrucciones y guías destinadas a los usuarios finales para el uso y la instalación del sistema.

7.1 Manual de Usuario: Guía para los usuarios finales sobre cómo utilizar la aplicación.

7.2 Manual de Instalación: Instrucciones detalladas para la instalación del sistema en la plataforma designada.

8. Conclusiones y Posibles Ampliaciones: Reflexiones finales sobre el proyecto, incluyendo su éxito y posibles mejoras o ampliaciones futuras.

9. Bibliografía: Lista de todas las fuentes y referencias utilizadas durante la realización del proyecto.

10. Anexos: Material adicional que complementa la memoria del proyecto, como diagramas, códigos fuente, o cualquier otro documento relevante.

### 3 Descripción general del proyecto

El proyecto consiste en el desarrollo de una aplicación web para los funcionarios de una clínica veterinaria. La aplicación permitirá a los usuarios registrarse e iniciar sesión para acceder a diversas funcionalidades, como la gestión de pacientes y un chat interno para la comunicación entre los funcionarios.

#### 3.1 Objetivos

1- Principal:

El objetivo principal del proyecto es crear una aplicación web completa que facilite la gestión de pacientes y la comunicación entre los funcionarios de la clínica veterinaria.

2- Secundarios:

. Implementación de sistema de registro y autenticación: Permitir a los funcionarios registrarse con credenciales únicas y asegurar que solo los usuarios autorizados puedan acceder a la aplicación.

. Gestión de empleados: Desarrollar funcionalidades para modificar y eliminar empleados de la base de datos de la clínica.

. Gestión de pacientes: Desarrollar funcionalidades para agregar, modificar y eliminar pacientes de la base de datos de la clínica.

. Chat interno: Integrar un sistema de chat en tiempo real para que los funcionarios puedan comunicarse entre sí de manera eficiente.

. Diseño intuitivo y amigable: Asegurar que la interfaz de usuario sea fácil de usar y que los funcionarios puedan navegar sin dificultad por las diferentes secciones de la aplicación.

### **3.2 Cuestiones metodológicas**

Para el desarrollo de este proyecto, he seguido una metodología de desarrollo basada en el modelo de ciclo de vida en cascada. Este modelo consiste en una serie de fases secuenciales, donde el progreso fluye en una dirección descendente, similar a una cascada. El flujo bidireccional en el modelo en cascada indica que siempre es posible volver a una fase anterior si se detectan errores o cambios en los requisitos durante el proceso de desarrollo. Esto permite una adaptación flexible a medida que avanza el proyecto.

Las fases principales del modelo son las siguientes:

- 1- Análisis de requisitos: En esta fase, se identifican y documentan los requisitos funcionales y no funcionales del sistema. Se establecen las características y funcionalidades que la aplicación debe cumplir.
- 2- Diseño del sistema: Una vez que se comprenden los requisitos, se procede al diseño del sistema. Esto incluye la arquitectura general, el diseño de la interfaz de usuario y la estructura de la base de datos.
- 3- Implementación y codificación: En esta etapa, se traduce el diseño del sistema en código. Se desarrollan las diferentes funcionalidades de la aplicación, asegurando que cumplan con los requisitos establecidos en la fase de análisis.
- 4- Pruebas: Una vez que se completa la implementación, se realizan pruebas exhaustivas para verificar que la aplicación funcione correctamente. Se identifican y corrigen errores o defectos, asegurando la calidad del software.
- 5- Despliegue e instalación: Después de superar las pruebas, la aplicación se despliega en un entorno de producción. Se instala en los servidores de la clínica veterinaria para que los funcionarios puedan acceder a ella.
- 6- Mantenimiento: Una vez que la aplicación está en funcionamiento, se proporciona soporte continuo y se realizan actualizaciones según sea necesario. Se pueden realizar mejoras adicionales en función de la retroalimentación de los usuarios.

### **3.3 Entorno de trabajo (tecnologías de desarrollo y herramientas)**

Tecnologías:

- Modelo cliente-servidor:  
Un modelo cliente-servidor, donde el cliente es el navegador web que accede a la aplicación a través de Internet, y el servidor es donde se aloja la aplicación web y la base de datos.

Herramientas software y lenguajes de programación:

- Lenguajes:
  - . Java: Utilizado para el desarrollo del backend de la aplicación, utilizando el framework Spring Tool Suite 4 (STS4) basado en Spring Framework.
  - . JavaScript: Empleado para la programación del frontend de la aplicación web, proporcionando interactividad y dinamismo.
  - . HTML (HyperText Markup Language) y CSS (Cascading Style Sheets): Utilizados para la estructura y el diseño visual de las páginas web respectivamente.
  
- Framework y herramientas:
  - . Spring Tool Suite 4 (STS4): Entorno de desarrollo integrado (IDE) basado en Eclipse para el desarrollo de aplicaciones Java, especialmente aplicaciones basadas en Spring.
  - . Bootstrap: Framework de desarrollo front-end utilizado para crear interfaces web modernas y receptivas.
  - . Apache Tomcat: Servidor web utilizado para implementar y ejecutar la aplicación web.
  - . MySQL: Sistema de gestión de bases de datos relacional utilizado para almacenar los datos de la aplicación.

## 4 Descripción general del producto

### 4.1 Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar.

- Visión general del sistema:
  - Límites del sistema:
    - El sistema es una aplicación autónoma diseñada para la gestión interna de una clínica veterinaria.
  - Funcionalidades básicas:
    - Gestión de pacientes.
    - Registro y autenticación de usuarios.
    - Gestión de datos de empleados (modificar, eliminar).
    - Gestión de datos de pacientes (añadir, modificar, eliminar).
    - Chat interno para comunicación entre funcionarios.
  - Usuarios:
    - Existen varios tipos de usuarios, incluyendo veterinarios, asistentes y administradores.
    - Los usuarios se autentican mediante un proceso de inicio de sesión con credenciales únicas.

- Solo los administradores pueden eliminar empleados o modificar datos de los empleados.

- Sistemas con los que pueda interactuar:

- La aplicación es una aplicación web que puede ser accesible desde cualquier navegador web en diferentes sistemas operativos, lo que la hace multiplataforma.

## **4.2 Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas.**

- Se ha utilizado el patrón de diseño Modelo-Vista-Controlador (MVC) para separar la lógica de negocio de la presentación y la interacción con el usuario. Esta elección se justifica por su capacidad para mejorar la mantenibilidad y escalabilidad del sistema.

## **4.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha**

- Plataforma tecnológica:
  - Sistema operativo: La aplicación puede ser instalada en diversos sistemas operativos, incluyendo Windows, macOS y Linux.
- Instalación de la aplicación:
  - El proceso de instalación implica desplegar la aplicación en un servidor web compatible con Java, como Apache Tomcat.
  - Se puede realizar la instalación mediante la carga de los archivos de la aplicación en el servidor y la configuración adecuada de la base de datos y otros recursos necesarios.

## **5 Planificación y presupuesto**

En este punto, quise hacerlo completo para tener una base de las fases y los valores, aunque no haya completado todo, quería mostrar cómo es en realidad.

Planificación:

- Cronograma (diagrama de Gantt):

Meses	Actividades
Marzo	Análisis de requisitos y diseño del sistema
Abril	Implementación y codificación
Mayo	Pruebas y depuración
Junio	Despliegue e instalación
Julio	Mantenimiento y ajustes

Presupuesto:

- Coste de desarrollo del código:
  - . Horas dedicadas:
    - . Análisis de requisitos y diseño del sistema: 100 horas
    - . Implementación y codificación: 200 horas
    - . Pruebas y depuración: 50 horas
    - . Despliegue e instalación: 20 horas
    - . Mantenimiento y ajustes: 30 horas
    - . Precio de hora de programador: 20€/hora
    - . Coste total:  $(100 + 200 + 50 + 20 + 30) * 20\text{€}/\text{hora} = 9.000\text{€}$
- Coste de software y herramientas de programación utilizadas:
  - . Spring Tool Suite 4 (STS4): Gratuito
  - . Bootstrap: Gratuito
  - . Apache Tomcat: Gratuito
  - . MySQL: Gratuito
  - . Total: 0€
- Coste del hardware:
  - . Puede variar dependiendo de las necesidades específicas de la infraestructura, pero se estima en 500€.
- Coste del hosting:
  - . Depende del proveedor de hosting y del tipo de plan seleccionado, pero se estima en 50€/mes.
    - . Coste total (para 6 meses):  $6 \text{ meses} * 50\text{€}/\text{mes} = 300\text{€}$
- Coste total del proyecto:
  - . Coste de desarrollo del código: 9.000
  - . Coste del hardware: 500€
  - . Coste del hosting: 300€
  - . Coste total: 9.800€

## 6 Documentación Técnica: análisis, diseño, implementación y pruebas.

### 6.1 Especificación de requisitos

### Requisitos Funcionales:

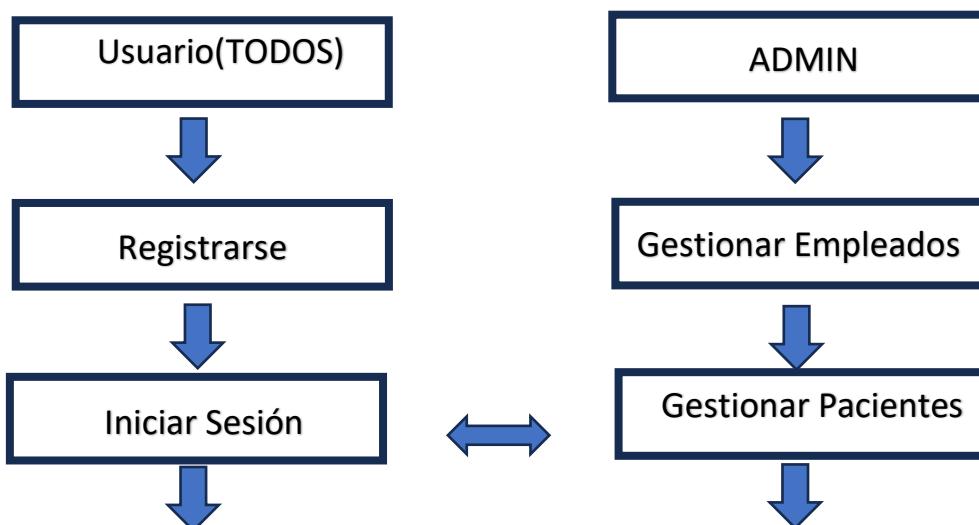
- Autenticación de Usuarios:
  - El sistema debe permitir a los usuarios registrarse con una dirección de correo electrónico y una contraseña, también nombre, apellidos, país y sexo.
  - El sistema debe permitir a los usuarios iniciar sesión utilizando sus credenciales (correo electrónico y contraseña).
- Gestión de Empleados:
  - El sistema debe permitir a los usuarios (ADMIN)modificar datos de los empleados.
  - El sistema debe permitir a los usuarios (ADMIN)eliminar datos de los empleados.
  - El sistema debe permitir a los usuarios (ADMIN)activar datos de los empleados.
  - El sistema debe permitir a los usuarios (TODOS)ver la lista de empleados.
- Gestión de Pacientes:
  - El sistema debe permitir a los usuarios (TODOS)ver la lista de pacientes.
  - El sistema debe permitir a los usuarios (TODOS)añadir datos de los pacientes.
  - El sistema debe permitir a los usuarios (TODOS)modificar datos de los pacientes.
  - El sistema debe permitir a los usuarios (TODOS)eliminar datos de los pacientes.
  - El sistema debe permitir a los usuarios (ADMIN)activar datos de los pacientes.
- Chat interno:
  - El sistema debe permitir a los usuarios (TODOS)conectaren al chat interno.
- Consultas:
  - El sistema debe permitir a los usuarios (TODOS)ver la lista de consultas.
- Inventario:
  - El sistema debe permitir a los usuarios (TODOS)ver la lista de medicamentos.

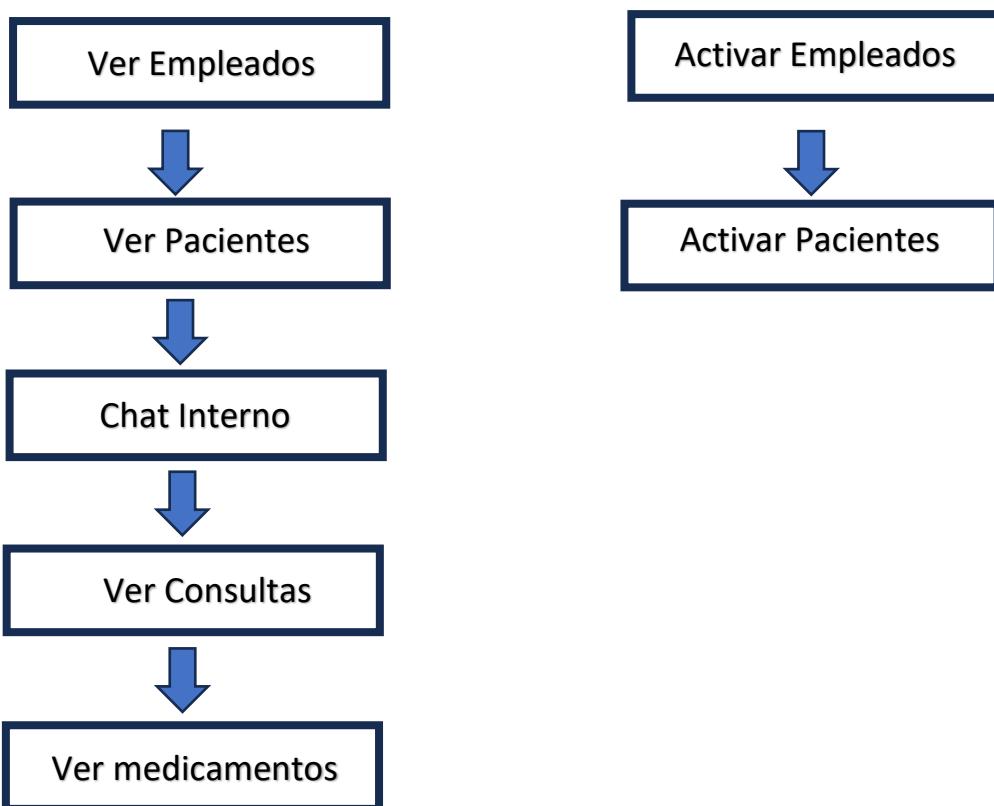
### Requisitos No Funcionales

- Seguridad:
  - El sistema debe cifrar todas las contraseñas de los usuarios en la base de datos.
- Rendimiento:

- El sistema debe ser capaz de manejar hasta 1000 usuarios concurrentes sin degradación significativa del rendimiento.
- Las páginas de la aplicación deben cargar en menos de 2 segundos bajo condiciones de carga normal.
- Escalabilidad:
  - El sistema debe ser capaz de escalar horizontalmente para soportar un aumento en el número de usuarios.
- Usabilidad:
  - El sistema debe ser intuitivo y fácil de usar, con una curva de aprendizaje mínima para nuevos usuarios.
  - La interfaz de usuario debe ser accesible y cumplir con los estándares de accesibilidad web.
- Mantenimiento:
  - El código del sistema debe estar bien documentado para facilitar el mantenimiento futuro.
  - El sistema debe seguir una arquitectura modular para permitir actualizaciones y mejoras sin afectar a todo el sistema.
- Disponibilidad:
  - El sistema debe tener una disponibilidad del 99.9%, excluyendo el tiempo de mantenimiento planificado.
- Compatibilidad:
  - El sistema debe ser compatible con los navegadores web modernos (Chrome, Firefox, Edge, Safari).
  - La aplicación debe ser responsive y funcionar correctamente en dispositivos móviles y de escritorio.

## 6.2 Análisis del sistema





#### Descripción de Casos de Uso:

- Autenticación de Usuarios:
  - Actor: Usuario (Todos)
  - Descripción: Permite a los usuarios registrarse con una dirección de correo electrónico, contraseña, nombre, apellidos, país y sexo. Además, permite a los usuarios iniciar sesión utilizando sus credenciales.
- Gestión de Empleados:
  - Actor: Administrador
  - Descripción: Permite a los administradores modificar, eliminar y activar datos de los empleados. Todos los usuarios pueden ver la lista de empleados.
- Gestión de Pacientes:
  - Actor: Usuario (Todos), Administrador
  - Descripción: Permite a todos los usuarios ver, añadir, modificar y eliminar datos de los pacientes. Solo los administradores pueden activar datos de los pacientes.
- Chat Interno:
  - Actor: Usuario (Todos)
  - Descripción: Permite a todos los usuarios conectarse al chat interno.

- Consultas:
  - Actor: Usuario (Todos)
  - Descripción: Permite a todos los usuarios ver la lista de consultas.
- Inventario:
  - Actor: Usuario (Todos)
  - Descripción: Permite a todos los usuarios ver la lista de medicamentos.

Modelo de Datos:

- Tabla Usuarios:

```
IdUsuario int NOT NULL AUTO_INCREMENT,
Nombre VARCHAR(50) NOT NULL,
Apellido VARCHAR(50) NOT NULL,
EsAdmin boolean DEFAULT false,
Email VARCHAR(50) NOT NULL,
Contrasena VARCHAR(500) NOT NULL,
Pais VARCHAR(50) NOT NULL,
Sexo VARCHAR(50) NOT NULL,
DNI varchar(9),
Telefono varchar(9),
EstaActivo boolean DEFAULT true,
PRIMARY KEY(idUsuario)
```

- Tabla Pacientes:

```
IdPaciente int NOT NULL AUTO_INCREMENT,
NumeroChip int NOT NULL ,
NombreMascota VARCHAR(50) NOT NULL,
Edad int NOT NULL,
NombreTutor VARCHAR(50) NOT NULL,
DNI VARCHAR(9) NOT NULL,
UbicacionMascota VARCHAR(50) NOT NULL,
EstaActivo boolean DEFAULT true,
PRIMARY KEY(IdPaciente)
```

- Tabla Consultas:

```
IdConsulta INT NOT NULL AUTO_INCREMENT,
IdUsuario INT NOT NULL,
IdPaciente INT NOT NULL,
Fecha VARCHAR(10),
Hora VARCHAR(5),
Descripcion VARCHAR(500) NOT NULL,
PRIMARY KEY(IdConsulta),
FOREIGN KEY (IdUsuario) REFERENCES usuarios(IdUsuario),
```

FOREIGN KEY (IdPaciente) REFERENCES pacientes(IdPaciente)

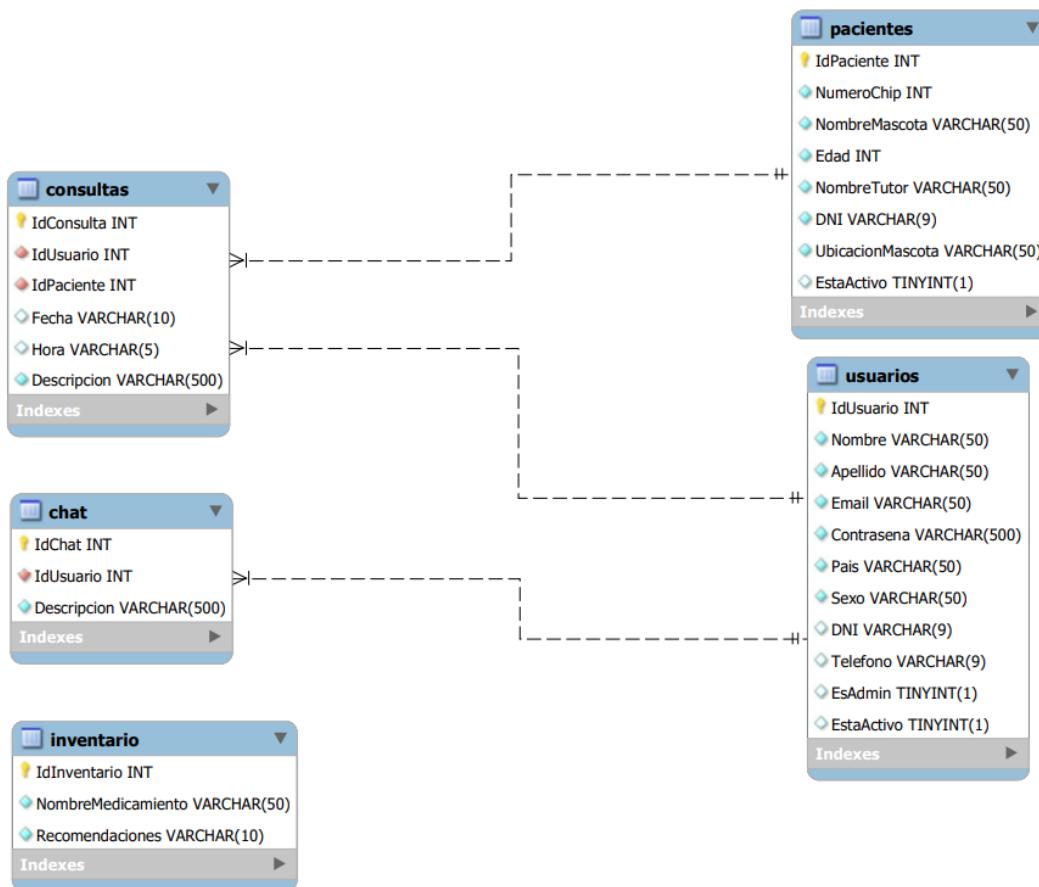
- Tabla inventario:  
IdInventario INT NOT NULL AUTO\_INCREMENT,  
NombreMedicamento VARCHAR(50) NOT NULL,  
Recomendaciones VARCHAR(10) NOT NULL,  
PRIMARY KEY(IdInventario)
- Tabla Chat:  
IdChat INT NOT NULL AUTO\_INCREMENT,  
IdUsuario INT NOT NULL,  
Descripcion VARCHAR(500) NOT NULL,  
PRIMARY KEY(IdChat),  
FOREIGN KEY (IdUsuario) REFERENCES usuarios(IdUsuario)

### **6.3 Diseño del sistema:**

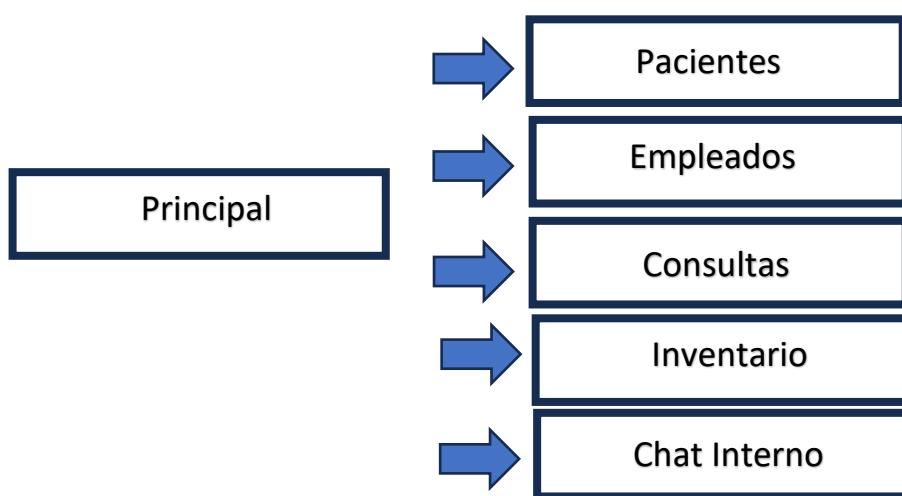
#### **6.3.1 Diseño de la Base de Datos**

Diagrama de Entidad-Relación (ER):

- La estructura de la base de datos está compuesta por las siguientes tablas: Usuarios, Pacientes, Consultas, inventario y Chat. Cada tabla está diseñada para gestionar información específica, asegurando la integridad y coherencia de los datos mediante claves primarias y relaciones definidas con claves foráneas.



### 6.3.2 Diseño de la Interfaz de usuario.



### Flujo de las pantallas:

Cuando ejecutamos la aplicación, aparecerá el inicio de sesión con el logotipo. Luego, se mostrará la pantalla para iniciar sesión, donde podré elegir entre ir a la página principal ingresando en la aplicación o registrarme. Cuando esté en la pantalla principal, podré elegir ir a las pantallas de gestión de pacientes, gestión de empleados, consultas, inventario o chat interno.

#### 6.3.3 Diseño de la Aplicación.

- Inicio de Sesión:
  - Mostrar la pantalla de inicio de sesión.
  - Validar las credenciales del usuario.
  - Permitir el acceso a usuarios registrados.
- Registro de Usuario:
  - Presentar la pantalla de registro.
  - Recopilar información del usuario (nombre, apellido, correo electrónico, contraseña, país, sexo).
  - Validar la información proporcionada.
  - Crear una nueva cuenta de usuario en la base de datos.
- Página Principal:
  - Mostrar la interfaz principal después del inicio de sesión.
  - Proporcionar acceso a diferentes funcionalidades de la aplicación.
- Gestión de Pacientes:
  - Mostrar la lista de pacientes registrados.
  - Permitir la adición, modificación, eliminación y activación de pacientes.
  - Mostrar detalles de cada paciente.
  - Permitir la búsqueda y filtrado de pacientes.
- Gestión de Empleados:
  - Mostrar la lista de empleados registrados.
  - Permitir la modificación, eliminación y activación de empleados (solo para administradores).
  - Mostrar detalles de cada empleado.
  - Permitir la búsqueda y filtrado de empleados.
- Consultas:
  - Mostrar la lista de consultas realizadas.
  - Permitir la creación de nuevas consultas, asignando paciente y fecha.
  - Mostrar detalles de cada consulta.
  - Permitir la búsqueda y filtrado de consultas.

- Inventario:
  - Mostrar la lista de medicamentos en inventario.
  - Permitir la adición, modificación y eliminación de medicamentos.
  - Mostrar detalles de cada medicamento.
  - Permitir la búsqueda y filtrado de medicamentos.
- Chat Interno:
  - Permitir a los usuarios conectarse al chat interno.
  - Mostrar la lista de conversaciones.
  - Permitir el envío y recepción de mensajes.
  - Mostrar detalles de cada mensaje.

## **6.4 Implementación:**

### **6.4.1 Entorno de desarrollo.**

Herramientas Utilizadas:

- Spring Tool Suite 4
- Git
- Java
- JavaScript
- HTML
- CSS

### **6.4.2 Estructura del código.**

- Librerías Descargadas:
  - Utilización de Maven para la gestión de dependencias.
- Programación:
  - El código está principalmente orientado a objetos, utilizando Java como lenguaje principal.

### **6.4.3 Cuestiones de diseño e implementación reseñables.**

La implementación del patrón de diseño Modelo-Vista-Controlador (MVC) es destacable por su capacidad para mejorar la mantenibilidad y escalabilidad del sistema. Se ha integrado un sistema de mensajería en tiempo real para la comunicación interna de los empleados utilizando WebSockets.

## **6.5 Pruebas.**

Pruebas:

- Compatibilidad:
  - Pruebas con diferentes navegadores y sistemas operativos para asegurar la compatibilidad.
- Carga:
  - Pruebas con un número elevado de registros y usuarios simultáneos para evaluar el rendimiento del sistema.
- Seguridad:
  - Pruebas de seguridad para asegurar la imposibilidad de acceso no autorizado

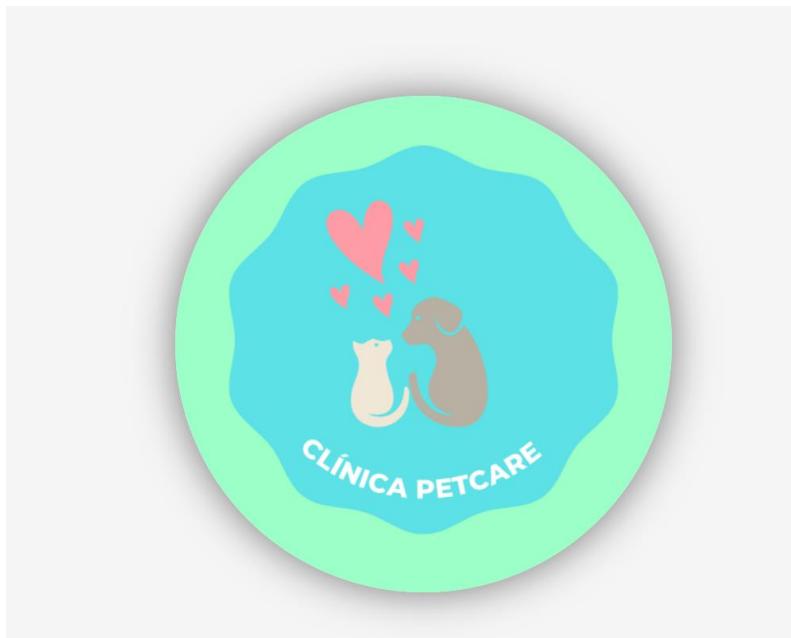
## 7 Manuales de usuario

### 7.1 Manual de usuario

Para explicar mejor la aplicación web, voy a presentar dos manuales de usuario, uno para los empleados administradores y otro para los demás.

Manual de usuario: ADMIN.

- Inicio aplicación.



Al iniciar la aplicación el logo de empresa aparece por 3 segundos y luego es direccionada para la pantalla donde se hace el login.

- Pantalla Login



# Bienvenido

Usuario:

Contraseña:

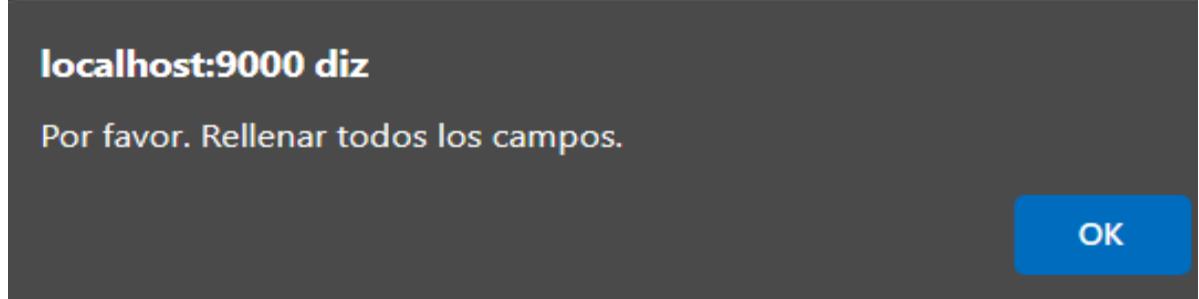
**ENTRAR**

¿No tienes cuenta? Pulse en 'REGISTRAR'

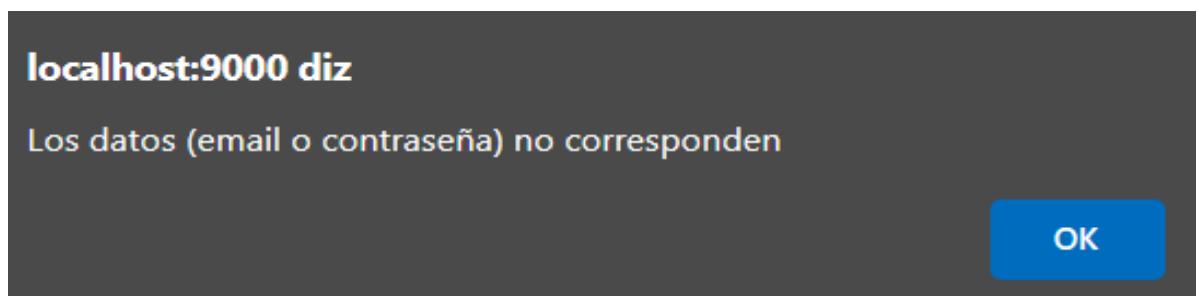
**REGISTRAR**

En la base de datos hay uno ADMIN que puede hacer todas las funcionalidades, incluso poner otros usuarios como administradores. Entonces cuando un usuario se registrar no puede elegir ser o no ADMIN. Para entrar en la aplicación el usuario debe poner el correo y la contraseña, luego pulsar en el botón ENTRAR.

Si uno de los campos o los dos están vacíos, aparecerá una alerta:



Si uno de los campos no es correcto, aparecerá una alerta:





Las contraseñas están cifradas, para hacer la prueba, abajo están los datos que debes poner en la pantalla login para entrar en la aplicación.

Usuario: [paula@hotmail.com](mailto:paula@hotmail.com)

Contraseña: 123

Result Grid   Filter Rows: [ ]					Edit: [ ]	Export/Import: [ ]	Wrap Cell Content: [ ]
	IdUsuario	Nombre	Apellido	Email	EsAdmin	Contrasenia	
▶	1	Paula	Nunes	paula@hotmail.com	1	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	2	Camila	Ranzani	camila@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	3	Sonia	Aparecida	sonia@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	4	Selmo	Costa	selmo@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	5	Solange	Ranzani	solange@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	6	Laura	Lopez	laura@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	7	Roberto	Silva	roberto@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	8	Alvaro	Souza	alvaro@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
	9	Sergio	Cruz	sergio@hotmail.com	0	a665a45920422f9d417e4867efdc4fb8a04a1f3fff1fa07e998e86f7f7a27ae3	
*	HULL	HULL	HULL	HULL	HULL	HULL	

- Pantalla principal

En todas las pantallas, la barra superior y la barra inferior son comunes. En el cuerpo de la pantalla principal, damos la bienvenida al usuario, cambiando siempre el nombre de acuerdo con el usuario que inicia sesión.

- Pantalla para gestionar datos de los empleados



Pulsamos en el desplegable que hay en la barra superior, desde la pantalla principal, y elegimos Empleados.



Se abre una nueva pantalla. En el cuerpo tenemos una tabla con los datos de los usuarios cuando se registraron, filtros para buscar un empleado en concreto y tres botones: Modificar datos de empleado, eliminar empleado y activar empleado.

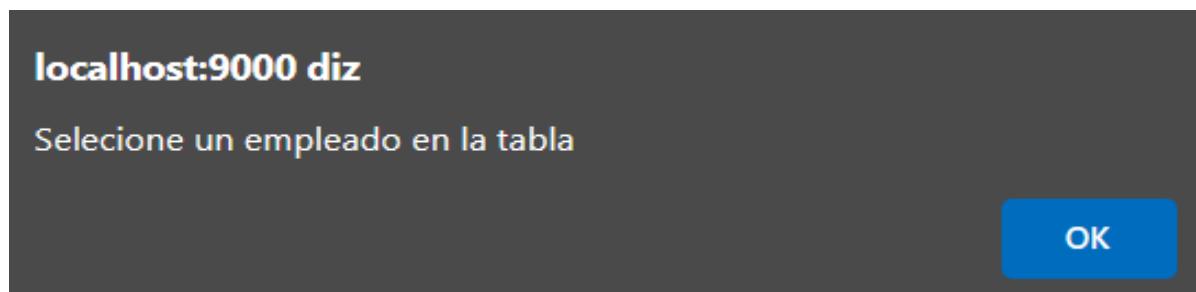
This screenshot shows the 'Empleados' management screen. At the top, there are five filter input fields: 'Filtrar por Nombre', 'Filtrar por Apellido', 'Filtrar por Email', 'Filtrar por País', and 'Filtrar por Sexo'. Below these is a table with columns: Id, Nombre, Apellido, Email, País, and Sexo. The table contains five rows of employee data:

	Id	Nombre	Apellido	Email	País	Sexo
<input type="checkbox"/>	1	Paula	Nunes	paula@hotmail.com	Brazil	mujer
<input type="checkbox"/>	2	Camila	Ranzani	camila@hotmail.com	Italia	mujer
<input type="checkbox"/>	3	Sonia	Aparecida	sonia@hotmail.com	Francia	mujer
<input type="checkbox"/>	4	Selmo	Costa	selmo@hotmail.com	Portugal	hombre
<input type="checkbox"/>	5	Solange	Ranzani	solange@hotmail.com	Brazil	mujer

At the bottom of the table are three buttons: 'Modificar Datos Empleado', 'Eliminar Empleado', and 'Activar Empleado'.

#### Botón Modificar Datos Empleado:

Debemos seleccionar un empleado. Pulsamos en uno  (checkbox) que hay en cada fila de la tabla. Si no elegimos y pulsamos el botón, aparecerá una alerta.



Cuando seleccionamos al empleado y pulsamos el botón, aparecerá un popup con los datos del empleado.

	Id	Nombre
<input type="checkbox"/>	1	Paula
<input checked="" type="checkbox"/>	2	Camila
<input type="checkbox"/>	3	Sonia
<input type="checkbox"/>	4	Selmo
<input type="checkbox"/>	5	Solange

Modificar Datos Empleado    Eliminar Empleado

Nombre: Camila    DNI: 78654325L  
Apellido: Ranzani    Telefono: 654278991  
Email: camila@hotmail.com    Es Admin: false  
Pais: Italia    Esta Activo?: true  
Sexo: mujer  
Salvar    Cancelar

Pais	Sexo
Brazil	mujer
Italia	mujer
Francia	mujer
Portugal	hombre
Brazil	mujer

El usuario ADMIN puede salvar las modificaciones o salir pulsando Cancelar.

Pero los campos: País, Sexo, EsAdmin y EstaActivo tienen algunas reglas.

En el campo País solo podemos elegir entre España, Brazil, Italia, Francia y Portugal. Si elegimos otra opción, aparecerá una alerta.

localhost:9000 diz  
En el campo pais solo acepta: España, Brazil, Italia, Francia y Portugal

Nombre	DNI
Camila	78654325L

Apellido	Telefono
Ranzani	654278991

Email	Es Admin
camila@hotmail.com	false

Pais	¿Esta Activo?
Africa	true

Sexo
mujer

**Salvar** **Cancelar**

En el campo Sexo, solo admite: hombre o mujer.

localhost:9000 diz  
En el campo sexo solo acepta: mujer o hombre

Nombre	DNI
Camila	78654325L

Apellido	Telefono
Ranzani	654278991

Email	Es Admin
camila@hotmail.com	false

Pais	¿Esta Activo?
Italia	true

Sexo
mujerssssssss

**Salvar** **Cancelar**

Los campos EsAdmin y EstaActivo solo permiten: false o true.



localhost:9000/empleados

localhost:9000 diz  
En el campo esAdmin solo acepta: true o false

Nombre	DNI
Camila	78654325
Apellido	Telefono
Ranzani	654278991
Email	Es Admin
camila@hotmail.com	falseeeeeeee
País	¿Esta Activo?
Italia	true
Sexo	
mujer	

Salvar Cancelar

Filtrar por Nombre Filtrar por Apellido

Gestionar Consultas

CLINICA PETCARE

Modificar Datos Empleado Eliminar Empleado

OK

Filtrar por Sexo

País	Sexo
Brazil	mujer
Italia	mujer
Francia	mujer
Portugal	hombre
Brazil	mujer

localhost:9000/empleados

localhost:9000 diz  
En el campo estaActivo solo acepta: true o false

Nombre	DNI
Camila	78654325
Apellido	Telefono
Ranzani	654278991
Email	Es Admin
camila@hotmail.com	false
País	¿Esta Activo?
Italia	trueqqqqqqqqqq
Sexo	
mujer	

Salvar Cancelar

Filtrar por Nombre Filtrar por Apellido

Gestionar Consultas

CLINICA PETCARE

Modificar Datos Empleado Eliminar Empleado

OK

Filtrar por Sexo

País	Sexo
Brazil	mujer
Italia	mujer
Francia	mujer
Portugal	hombre
Brazil	mujer

El usuario debe respetar el límite máximo de caracteres en los campos, ya que tenemos planeado un tamaño en la base de datos.

```
CREATE TABLE usuarios(
    IdUsuario int NOT NULL AUTO_INCREMENT,
    Nombre VARCHAR(50) NOT NULL,
    Apellido VARCHAR(50) NOT NULL,
    Email VARCHAR(50) NOT NULL,
    Contrasena VARCHAR(500) NOT NULL,
    Pais VARCHAR(50) NOT NULL,
    Sexo VARCHAR(50) NOT NULL,
    DNI varchar(9),
    Telefono varchar(9),
    EsAdmin boolean DEFAULT false,
    EstaActivo boolean DEFAULT true,
    PRIMARY KEY(idUsuario)
);
```

Si todo está correcto y pulsamos el botón, aparecerá una alerta, el popup se cerrará y la tabla se recargará con los datos modificados.

localhost:9000/empleados

CLINICA  
PETCARE

Gestionar Consultas

Filtrar por Nombre Filtrar por Apellido

	<b>Id</b>	<b>Nombre</b>
<input type="checkbox"/>	1	Paula
<input checked="" type="checkbox"/>	2	Camila
<input type="checkbox"/>	3	Sonia
<input type="checkbox"/>	4	Selmo
<input type="checkbox"/>	5	Solange

Modificar Datos Empleado Eliminar Empleado

localhost:9000 diz  
Modificado

OK

<b>Nombre</b>	<b>DNI</b>
Camilaqwqiwuqwiqyiqeu	78654325L

<b>Apellido</b>	<b>Telefono</b>
Ranzani	654278991

<b>Email</b>	<b>Es Admin</b>
camila@hotmail.com	false

<b>País</b>	<b>¿Esta Activo?</b>
Italia	true

<b>Sexo</b>
mujer

Salvar Cancelar

Filtrar por Sexo

<b>País</b>	<b>Sexo</b>
Brazil	mujer
Italia	mujer
Francia	mujer
Portugal	hombre
Brazil	mujer



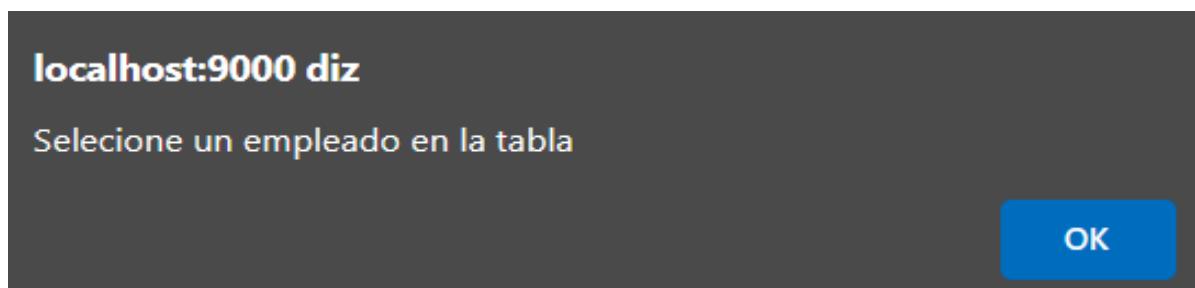
The screenshot shows a software application window titled "CLINICA PETCARE". At the top, there is a logo with the text "CLINICA" and "PETCARE" above a cartoon animal icon. To the right of the logo are menu items: "Gestionar", "Consultas", "Inventario", and "Comunicacion Interna". Below the menu is a search bar with four input fields: "Filtrar por Nombre", "Filtrar por Apellido", "Filtrar por Email", and "Filtrar por País". There is also a "Filtrar por Sexo" button. The main area is a table with the following columns: "Id", "Nombre", "Apellido", "Email", "País", and "Sexo". The table contains five rows of data:

	Id	Nombre	Apellido	Email	País	Sexo
<input type="checkbox"/>	1	Paula	Nunes	paula@hotmail.com	Brazil	mujer
<input type="checkbox"/>	2	Camilaqwqiwuqwqiqyiqyeuw12	Ranzani	camila@hotmail.com	Italia	mujer
<input type="checkbox"/>	3	Sonia	Aparecida	sonia@hotmail.com	Francia	mujer
<input type="checkbox"/>	4	Selmo	Costa	selmo@hotmail.com	Portugal	hombre
<input type="checkbox"/>	5	Solange	Ranzani	solange@hotmail.com	Brazil	mujer

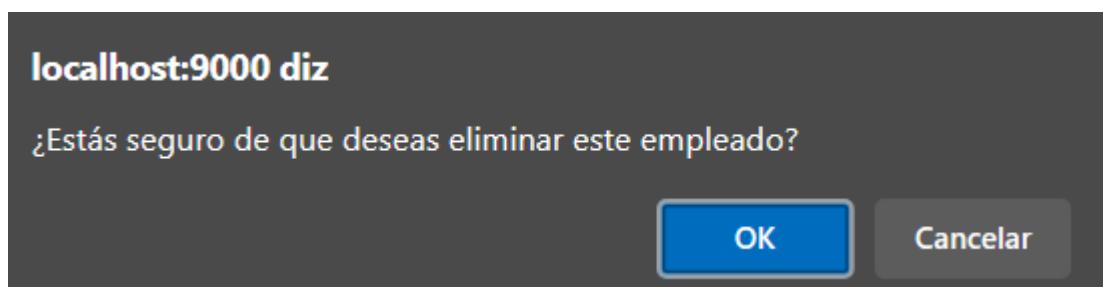
At the bottom of the table are three buttons: "Modificar Datos Empleado", "Eliminar Empleado", and "Activar Empleado".

### Botón Eliminar Empleado:

Debemos seleccionar un empleado. Pulsamos en uno  (checkbox) que hay en cada fila de la tabla. Si no elegimos y pulsamos el botón, aparecerá una alerta.



Cuando seleccionamos al empleado y pulsamos el botón, aparece una alerta.



Podemos confirmar pulsando "OK" o cancelar la operación pulsando "Cancelar".



**localhost:9000 diz**

Eliminado.

OK

**localhost:9000 diz**

Eliminación cancelada.

OK

Cuando se elimina un empleado, la tabla se recarga y el empleado eliminado ya no aparecerá.

The screenshot shows a web application interface for managing employees. At the top, there is a logo for 'CLINICA PETCARE' featuring a cartoon animal. To the right of the logo are navigation links: 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. Below the header, there are five input fields with dropdown menus: 'Filtrar por Nombre', 'Filtrar por Apellido', 'Filtrar por Email', 'Filtrar por País', and 'Filtrar por Sexo'. A table below these filters displays a list of employees with the following columns: Id, Nombre, Apellido, Email, País, and Sexo. The data in the table is as follows:

	<b>Id</b>	<b>Nombre</b>	<b>Apellido</b>	<b>Email</b>	<b>País</b>	<b>Sexo</b>
<input type="checkbox"/>	1	Paula	Nunes	paula@hotmail.com	Brazil	mujer
<input type="checkbox"/>	3	Sonia	Aparecida	sonia@hotmail.com	Francia	mujer
<input type="checkbox"/>	4	Selmo	Costa	selmo@hotmail.com	Portugal	hombre
<input type="checkbox"/>	5	Solange	Ranzani	solange@hotmail.com	Brazil	mujer
<input type="checkbox"/>	6	Laura	Lopez	laura@hotmail.com	Espana	mujer

At the bottom of the table area, there are three buttons: 'Modificar Datos Empleado', 'Eliminar Empleado', and 'Activar Empleado'.

Importante: los datos no se eliminaron; el usuario permanecerá en la base de datos como inactivo. Si el ADMIN desea volver a activarlo, debe pulsar el botón 'Activar empleado'.

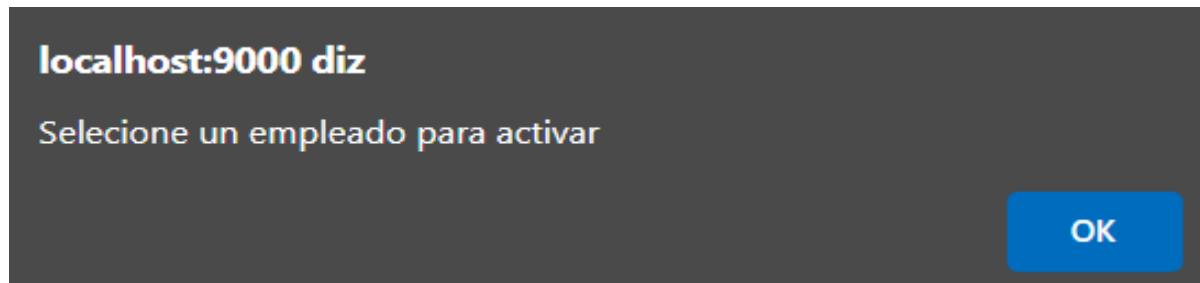


	IdUsuario	Nombre	Apellido	Email	EsAdmin	EstaActivo
▶	1	Paula	Nunes	paula@hotmail.com	1	1
	2	Camila...	Ranzani	camila@hotmail.com	0	0
	3	Sonia	Aparecida	sonia@hotmail.com	0	1
	4	Selmo	Costa	selmo@hotmail.com	0	1
	5	Solange	Ranzani	solange@hotmail.com	0	1
	6	Laura	Lopez	laura@hotmail.com	0	1
	7	Roberto	Silva	roberto@hotmail.com	0	1
	8	Alvaro	Souza	alvaro@hotmail.com	0	1
	9	Sergio	Cruz	sergio@hotmail.com	0	1
*	NULL	NULL	NULL	NULL	NULL	NULL

### Botón Activar Empleado:

Solo los administradores pueden activar empleados. Cuando se presione el botón 'Activar Empleado', aparecerá un popup con una tabla que contiene los empleados inactivos. El administrador debe seleccionar uno de los empleados. Si no se realiza ninguna selección y se presiona el botón 'Activar', se mostrará una alerta.

The screenshot shows the PETCARE application interface. At the top, there is a navigation bar with the logo, followed by 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. Below the navigation bar, there is a search bar with four filters: 'Filtrar por Nombre', 'Filtrar por Apellido', 'Filtrar por Email', and 'Filtrar por País'. A large table lists employees with columns for Id, Nombre, Apellido, and Email. The 'Email' column shows the active status of each employee. A modal dialog box is overlaid on the page, containing a smaller table with columns 'Id' and 'Email'. This table lists employees with inactive status (Email column shows 0). Each row has a checkbox next to the Id. At the bottom of the modal are two buttons: 'Activar' (Activate) and 'Cancelar' (Cancel). The background of the main page shows a list of employees with their details and various action buttons at the bottom.



Cuando se selecciona un empleado y se presiona el botón 'Activar', aparecerá una alerta de confirmación. Al hacer clic en 'OK', el popup se cerrará y la tabla en la pantalla de empleados se recargará, mostrando al empleado activado.

The screenshot shows a web application interface for managing employees. On the left, there's a sidebar with icons for 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. The main area has four filter boxes: 'Filtrar por Nombre', 'Filtrar por Apellido', 'Filtrar por Email', and 'Filtrar por País'. Below these filters is a table of employees:

	Id	Nombre	Apellido
<input type="checkbox"/>	1	Paula	Nunes
<input type="checkbox"/>	3	Sonia	Aparecida
<input type="checkbox"/>	4	Selmo	Costa
<input type="checkbox"/>	5	Solange	Ranzani
<input type="checkbox"/>	7	Roberto	Silva

At the bottom of this table are buttons for 'Modificar Datos Empleado', 'Eliminar Empleado', and 'Activar'. A modal dialog is overlaid on the page, containing another table with columns 'Id' and 'Email':

	Id	Email
<input type="checkbox"/>	2	camila@hotmail.com
<input checked="" type="checkbox"/>	6	laura@hotmail.com
<input type="checkbox"/>	8	alvaro@hotmail.com
<input type="checkbox"/>	9	sergio@hotmail.com

At the bottom of this modal are 'Activar' and 'Cancelar' buttons. To the right of the modal, there's another table for 'Filtrar por País' and 'Filtrar por Sexo'.





The screenshot shows a software application window titled 'CLINICA PETCARE'. At the top, there's a logo with a colorful butterfly and the text 'CLINICA PETCARE'. To the right of the logo are navigation links: 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. Below these are five filter input fields: 'Filtrar por Nombre', 'Filtrar por Apellido', 'Filtrar por Email', 'Filtrar por País', and 'Filtrar por Sexo'. The main area is a table with columns: 'Id', 'Nombre', 'Apellido', 'Email', 'País', and 'Sexo'. The table contains six rows of data:

	Id	Nombre	Apellido	Email	País	Sexo
<input type="checkbox"/>	1	Paula	Nunes	paula@hotmail.com	Brazil	mujer
<input type="checkbox"/>	3	Sonia	Aparecida	sonia@hotmail.com	Francia	mujer
<input type="checkbox"/>	4	Selmo	Costa	selmo@hotmail.com	Portugal	hombre
<input type="checkbox"/>	5	Solange	Ranzani	solange@hotmail.com	Brazil	mujer
<input type="checkbox"/>	6	Laura	Lopez	laura@hotmail.com	Espana	mujer

At the bottom of the table area are three buttons: 'Modificar Datos Empleado', 'Eliminar Empleado', and 'Activar Empleado'.

Si pulsamos en Cancelar el popup cierra y no activamos el empleado.

- **Pantalla para gestionar datos de los pacientes**

Pulsamos en el desplegable que hay en la barra superior, desde la pantalla principal, y elegimos Pacientes.

The screenshot shows the same software interface as the previous one, but with a different selection in the top navigation bar. The 'Gestionar' dropdown menu is open, and the 'Pacientes' option is highlighted with a red box. The other options in the dropdown are 'Consultas', 'Inventario', and 'Comunicacion Interna'. The rest of the interface is identical to the previous screenshot, showing the employee management table and buttons.

Se abre una nueva pantalla. En el cuerpo tenemos una tabla con los datos de los pacientes, filtros para buscar un paciente en concreto y cuatro botones: Añadir Paciente, Modificar Datos paciente, Eliminar paciente y Activar Paciente.



The screenshot shows a software application titled "CLINICA PETCARE". At the top, there is a logo with a colorful geometric pattern and the text "CLINICA PETCARE". Below the logo, there is a navigation bar with links: "Gestionar", "Consultas", "Inventario", and "Comunicacion Interna". A search bar at the top has five input fields: "Filtrar por NumeroChip", "Filtrar por Nombre Mascota", "Filtrar por Edad", "Filtrar por Nombre Tutor", and "Filtrar por DNI Tutor". Below the search bar is a table with columns: "Id", "Numero Chip", "Nombre Mascota", "Edad", "Nombre Tutor", "DNI Tutor", and "Ubicacion Mascota". The table contains five rows of data:

Id	Numero Chip	Nombre Mascota	Edad	Nombre Tutor	DNI Tutor	Ubicacion Mascota
1	48596	Mel	5	Paulinha	12345678P	Calle Galana,54
2	38495	Toto	6	Juan	22998811J	Paseo Zorrilla,12
3	39475	Mial	8	Carmen	87654321C	Plaza España,34
4	20495	Jorge	1	Maria	37261839J	Plaza Madrid,26
5	48379	Samuka	3	Juana	78462748S	Av.Zamora,7

At the bottom of the table are four buttons: "Añadir Paciente", "Modificar Datos Paciente", "Eliminar Paciente", and "Activar Paciente".

### Botón Añadir Paciente:

Cuando el usuario pulsa el botón, surge un popup con campos que deben ser rellenados.

The screenshot shows the same software interface as the previous one, but with a modal dialog box open over the patient list. The dialog box is titled "Añadir Paciente" and contains several input fields grouped by category:

- Numero Chip**: ID Chip (input field)
- DNI Tutor**: DNI del Tutor (input field)
- Nombre Mascota**: Nombre de la Mascota (input field)
- Ubicación**: Ubicación de la Mascota (input field)
- Edad**: Edad (input field)
- Esta Activo?**: Esta Activo? (input field)
- Nombre Tutor**: Nombre del Tutor (input field)

At the bottom of the dialog box are two buttons: "Salvar" (Save) and "Cancelar" (Cancel).

Si no rellena algún campo, aparecerá una alerta.



**localhost:9000 diz**

Por favor. Rellenar todos los campos.

OK

Importante: El número de chip no puede ser igual. Si existe aparecerá una alerta.

**localhost:9000 diz**

El paciente ya está registrado.

OK

El usuario debe respetar el límite máximo de caracteres en los campos, ya que tenemos planeado un tamaño en la base de datos.

```
CREATE TABLE pacientes(
    IdPaciente int NOT NULL AUTO_INCREMENT,
    NumeroChip int NOT NULL ,
    NombreMascota VARCHAR(50) NOT NULL,
    Edad int NOT NULL,
    NombreTutor VARCHAR(50) NOT NULL,
    DNIITutor VARCHAR(9) NOT NULL,
    UbicacionMascota VARCHAR(50) NOT NULL,
    EstaActivo boolean DEFAULT true,
    PRIMARY KEY(IdPaciente)
);
```

Después de llenar todos los campos, pulsa el botón 'Salvar'. Aparecerá una alerta de confirmación. Los campos del popup estarán vacíos para que el



usuario pueda añadir otro paciente. Si deseas salir, pulsa el botón 'Cancelar', el popup se cerrará y la tabla se recargará.

The screenshot shows a web-based application for a veterinary clinic named 'CLINICA PETCARE'. The main interface has tabs for 'Gestionar', 'Consultas', and 'Inventario'. A modal dialog box is open in the center, titled 'localhost:9000 diz' with the sub-titler 'Paciente Añadido'. The dialog contains fields for 'Numero Chip' (12345), 'DNI Tutor' (12345678P), 'Nombre Mascota' (TITI), 'Ubicación' (Av. Titi), 'Edad' (1), 'Esta Activo?' (true), and 'Nombre Tutor' (Paula). At the bottom are 'Salvar' and 'Cancelar' buttons. In the background, there's a table with columns 'Id', 'Numero Chip', 'Nombre Mascota', 'Ubicación', and 'DNI Tutor', showing several rows of data. The row for the new patient (Id 8, Nombre Mascota TITI) is highlighted with a red border.

This screenshot shows the same application interface after the new patient has been added. The table now includes the new row (Id 8, Nombre Mascota TITI) at the bottom. The row is highlighted with a red border. Below the table are buttons for 'Añadir Paciente', 'Modificar Datos Paciente', 'Eliminar Paciente', and 'Activar Paciente'.

#### Botón Modificar Datos Paciente:

Si pulsa el botón y no selecciona un dato en la tabla aparecerá una alerta.

**localhost:9000 diz**

Seleccione un paciente en la tabla

**OK**



Cuando se seleccione al paciente y se pulse el botón, aparecerá el popup relleno con los datos del paciente.

The screenshot shows a modal dialog box over a main application window. The main window has a header with 'CLINICA PETCARE' and navigation links: 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. The modal dialog has two tabs: 'Filtrar por NumeroChip' and 'Filtrar por Nombre'. The 'Numero Chip' tab is active, showing a table with columns: 'Numero Chip', 'Nombre Mascota', 'Edad', and 'Tutor'. A row for '8' is selected, showing '12345', 'TITI', '1', and 'TITI'. The 'Nombre Tutor' section contains a dropdown menu with entries: 'Plaza Espana,34', 'Plaza Madrid,26', 'Av. Zamora,7', 'Av.Salamanca,8', 'Av.Segovia,12', and 'Av. Titi'. The 'DNI Tutor' and 'Ubicación' sections have empty input fields. The 'Esta Activo?' field contains 'true'. At the bottom of the dialog are 'Salvar' and 'Cancelar' buttons.

Importante: No podemos cambiar el numero Chip  
Los campos EstaActivo solo permiten: false o true.

The screenshot shows a dark gray modal dialog box with the text 'localhost:9000 diz' at the top. Below it is the message 'En el campo estaActivo solo acepta: true o false'. In the bottom right corner is a blue 'OK' button.

#### Botón Eliminar Paciente:

Hay que seleccionar un paciente en la tabla; de lo contrario, aparecerá una alerta.

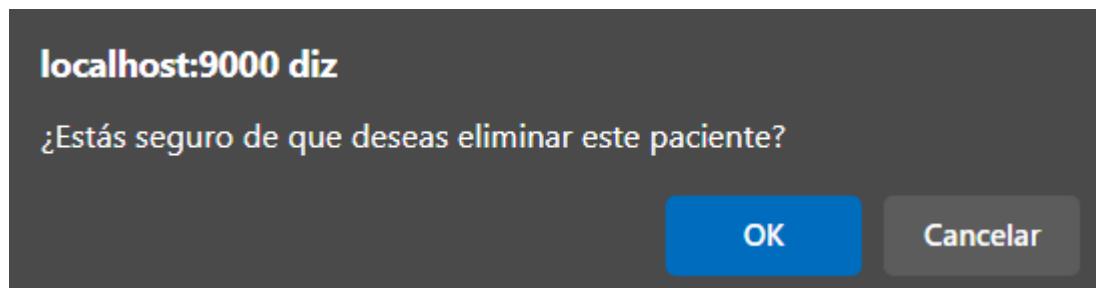


IES RIBERA DE CASTILLA

PETCARE



Cuando seleccionamos al paciente y pulsamos el botón, aparece una alerta.



Si pulsamos Cancelar, aparecerá una alerta y la operación no se realiza.



Podemos confirmar pulsando "OK".



Botón Activar Paciente:

Solo los administradores pueden activar pacientes. Cuando se presione el botón 'Activar Empleado', aparecerá un popup con una tabla que contiene los



pacientes inactivos. El administrador debe seleccionar uno de los pacientes. Si no se realiza ninguna selección y se presiona el botón 'Activar', se mostrará una alerta.

The screenshot shows a web-based application interface for a veterinary clinic named 'CLINICA PETCARE'. At the top, there is a logo with a colorful paw print and the text 'CLINICA PETCARE'. Below the logo, there is a navigation bar with links: 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. The main content area has four filter boxes: 'Filtrar por NumeroChip', 'Filtrar por Nombre Mascota', 'Filtrar por Edad', 'Filtrar por Nombre Tutor', and 'Filtrar por DNI Tutor'. Below these filters is a table with columns 'Id', 'Numero Chip', and 'Nombre Mascota'. The table contains eight rows of data. A modal dialog box is overlaid on the page, containing a smaller table with columns 'Id' and 'Numero Chip'. It also includes two buttons: 'Activar' (highlighted in blue) and 'Cancelar'. In the background, to the right of the modal, there is another table with columns 'DNI Tutor' and 'Ubicacion Mascota', showing five rows of data. At the bottom of the page, there are several buttons: 'Añadir Paciente', 'Modificar Datos Paciente', 'Eliminar Paciente', and 'Activar Paciente'.

This screenshot shows a confirmation dialog box with a dark gray background. At the top, it displays the text 'localhost:9000 diz'. Below that, the message 'Seleccione un paciente para activar' is centered. In the bottom right corner of the dialog, there is a large blue button labeled 'OK'.

Cuando se selecciona un paciente y se presiona el botón 'Activar', aparecerá una alerta de confirmación. Al hacer clic en 'OK', el popup se cerrará y la tabla en la pantalla de pacientes se recargará, mostrando el paciente activado.

This screenshot shows the same application interface as the previous one, but with a key difference: the second row in the main table now has a checked checkbox in the 'Id' column. This indicates that a patient has been selected for activation. The rest of the interface, including the modal dialog and the background tables, remains the same.



Si pulsamos em Cancelar el popup cierra y no activamos el empleado.

- Pantalla para el chat interno(funcionarios)

The screenshot shows a table with columns: Id, Numero Chip, Nombre Mascota, Edad, Nombre Tutor, DNI Tutor, and Ubicacion Mascota. The data is as follows:

Id	Numero Chip	Nombre Mascota	Edad	Nombre Tutor	DNI Tutor	Ubicacion Mascota
1	48596	Mel	5	Paulinha	12345678P	Calle Galana,54
2	38495	Toto	6	Juan	2299881J	Paseo Zorrilla,12
3	39475	Mial	8	Carmen	87654321C	Plaza Espana,34
5	48379	Samuka	3	Juana	78462748S	Av.Zamora,7
7	19384	Tiko	2	Cris	62734827R	Av.Segovia,12

Buttons at the bottom: Añadir Paciente, Modificar Datos Paciente, Eliminar Paciente, Activar Paciente.

Desde la pantalla principal, elegimos la opción Comunicación Interna, una nueva ventana se abre.

The screenshot shows a chat interface with sections for connecting to the server, sending messages, and viewing received messages and users in the room.

**Conectar con el servidor:** Includes 'Conectar' and 'Desconectar' buttons.

**Escribe tu nickname:** Input field with placeholder 'Paula'.

**Unirse al chat** button.

**Mensajes recibidos:** Placeholder for received messages.

**Usuarios en la sala:** Placeholder for users in the room.

**Escribe tu mensaje:** Input field.

**Enviar mensaje** button.



Todos los usuarios pueden conectar.  
Debe seguir los pasos para empezar el chat:  
1- Pulsar el botón Conectar

The screenshot shows the PETCARE application interface. At the top, there is a header with the logo 'CLINICA PETCARE' and navigation links 'Gestionar', 'Consultas', and 'OK'. Below this, a modal window titled 'localhost:9000 diz' displays the message 'Conectado' and an 'OK' button. The main interface has a teal gradient background. On the left, it says 'Conectar con el servidor' with 'Conectar' and 'Desconectar' buttons. In the center, there is a text input field labeled 'Escribe tu nickname:' with the placeholder 'Paula' and a 'Unirse al chat' button. To the right, there are two panels: 'Mensajes recibidos:' and 'Usuarios en la sala:', both currently empty.

2- Escriba tu nombre y pulse el botón Unirse al chat si el campo está vacío aparecerá una alerta.

A dark gray dialog box with the title 'localhost:9000 diz' at the top. The main message 'El nickname no puede estar vacío' is displayed in white text. In the bottom right corner, there is a blue 'OK' button.

The screenshot shows the PETCARE application interface after connecting. The top header now includes 'Inventario' and 'Comunicacion Interna'. The main interface shows the connection status 'Conectado' in the top right. The 'nickname' field now contains 'Paulinha'. The 'Unirse al chat' button is visible. The 'Mensajes recibidos:' panel shows the message 'Paulinha\' -- entro en el chat -- \''. The 'Usuarios en la sala:' panel shows the user 'Paulinha'.

3- Ahora escriba un mensaje y pulse el botón Enviar Mensaje, si está vacío aparecerá una alerta.



localhost:9000 diz

El mensaje no puede estar vacío

OK

The screenshot shows a web-based internal communication system for a veterinary clinic. At the top, there's a logo for 'CLINICA PETCARE' featuring a colorful stylized animal head. The navigation bar includes links for 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. On the left, a sidebar displays the name 'Paula'. The main area has a teal background. It features a section titled 'Conectar con el servidor' with 'Conectar' and 'Desconectar' buttons. A nickname input field contains 'Paulinha', and a blue 'Unirse al chat' button is below it. To the right, there's a message input field with 'Escribe tu mensaje:' placeholder text and a blue 'Enviar mensaje' button. A sidebar on the right shows a list of received messages: 'Paulinha:\\" -- entro en el chat -- \\'', 'Paulinha:Hola Soy Paulinha', and a list of users in the room: 'Paulinha'.

Ahora vamos a ver otro funcionario para conectar:

This screenshot shows the same internal communication interface. The sidebar now displays the name 'Camila'. The main area shows an empty message input field with 'Escribe tu mensaje:' placeholder text and a blue 'Enviar mensaje' button. The message history and user list on the right side are empty, indicating that Camila has not yet connected or sent any messages.



IES RIBERA DE CASTILLA

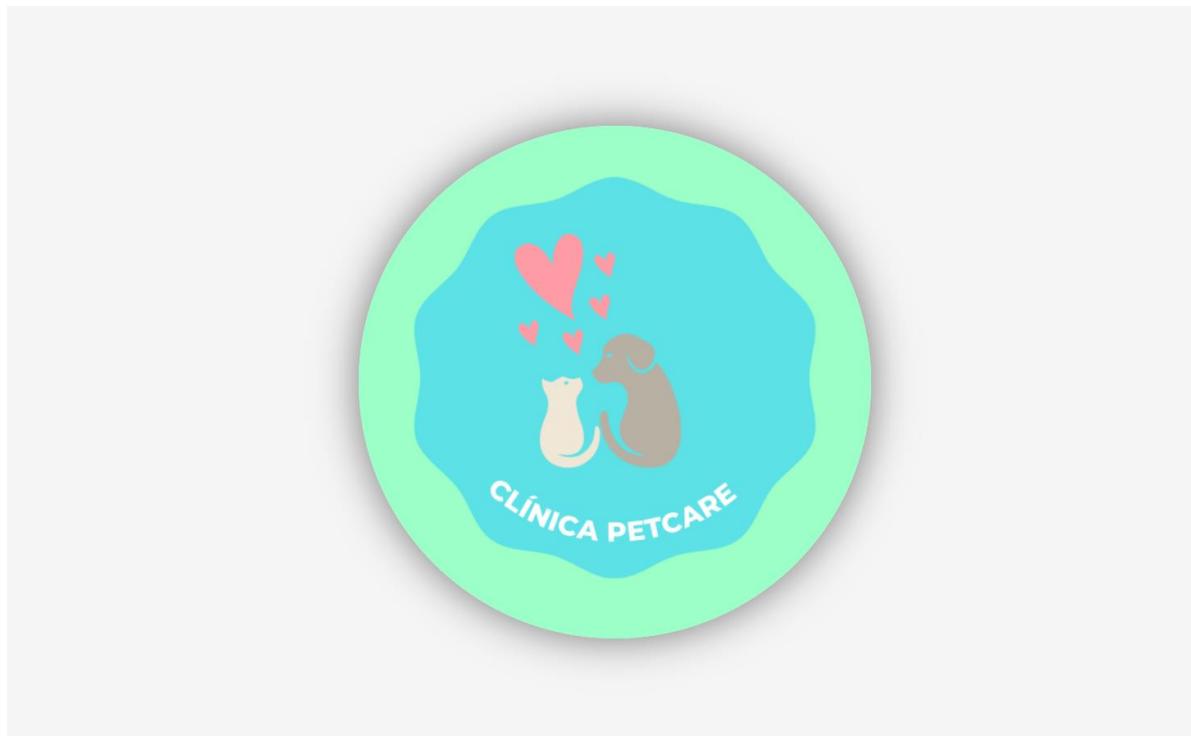
PETCARE

The screenshot shows the PETCARE chat interface. At the top, there's a logo for 'CLINICA PETCARE' featuring a colorful paw print. The menu bar includes 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. On the left, a sidebar says 'Conectar con el servidor' with 'Conectar' and 'Desconectar' buttons. The main area has a teal background. A user profile 'Camila' is shown on the left. In the center, there's an input field for 'Escribe tu nickname:' containing 'Camilinha', a 'Unirse al chat' button, and a message input field with 'Enviar mensaje' at the bottom. On the right, a 'Mensajes recibidos:' section shows messages from 'Camilinha': 'Camilinha:\\" -- entro en el chat -- \\'', 'Camilinha:Hola soy Camilinha', and a 'Usuarios en la sala:' section showing 'Camilinha, Paulinha'.

The screenshot shows the PETCARE chat interface. The layout is identical to the previous one, with the 'CLINICA PETCARE' logo at the top. The menu bar includes 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. On the left, a sidebar says 'Conectar con el servidor' with 'Conectar' and 'Desconectar' buttons. The main area has a teal background. A user profile 'Paula' is shown on the left. In the center, there's an input field for 'Escribe tu nickname:' containing 'Paulinha', a 'Unirse al chat' button, and a message input field with 'Enviar mensaje' at the bottom. On the right, a 'Mensajes recibidos:' section shows messages from 'Paulinha': 'Paulinha:\\" -- entro en el chat -- \\'', 'Paulinha:Hola Soy Paulinha', and a 'Usuarios en la sala:' section showing 'Camilinha, Paulinha'.

Manual de usuario: NO es ADMIN.

- Inicio aplicación.



Al iniciar la aplicación el logo de empresa aparece por 3 segundos y luego es direccionada para la pantalla donde se hace el login.

- Pantalla Login



# Bienvenido

Usuario:

Contraseña:

**ENTRAR**

¿No tienes cuenta? Pulse en 'REGISTRAR'

**REGISTRAR**

Si el usuario no tiene registro debe pulsar el botón Registrar. Se abre una nueva ventana.

## Registrar

Nombre

Apellido

Usuario

Contraseña

País  España

Sexo  Hombre  Mujer

**Salir** **Confirmar**

Si algún campo está vacío aparecerá una alerta.

**localhost:9000 diz**

Por favor. Rellenar todos los campos.

**OK**

El campo Usuario hay una regla, si no cumple la regla aparecerá una alerta.

**localhost:9000 diz**

Por favor, el campo 'Usuario' es un correo. Cantidad de caracteres permitida en el campo (22). Formato del correo: Solo es permitido letras y numeros y debe terminar @hotmail.com

**OK**

Si cumple todas las reglas y pulsamos el botón Confirmar, aparecerá un alerta confirmando, los campos estarán limpios. Para salir pulsamos el botón Salir. Y volvemos a la pantalla de login.

**localhost:9000 diz**

Registro confirmado!

**OK**

Para entrar en la aplicación el usuario debe poner el correo y la contraseña, luego pulsar en el botón ENTRAR.

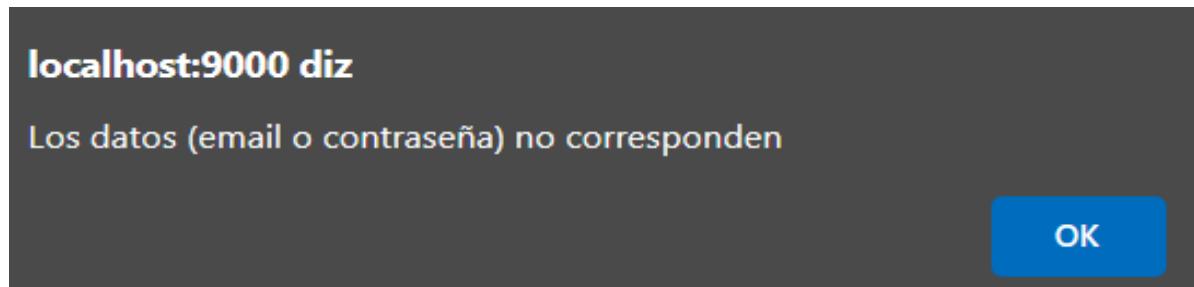
Si uno de los campos o los dos están vacíos, aparecerá una alerta:

**localhost:9000 diz**

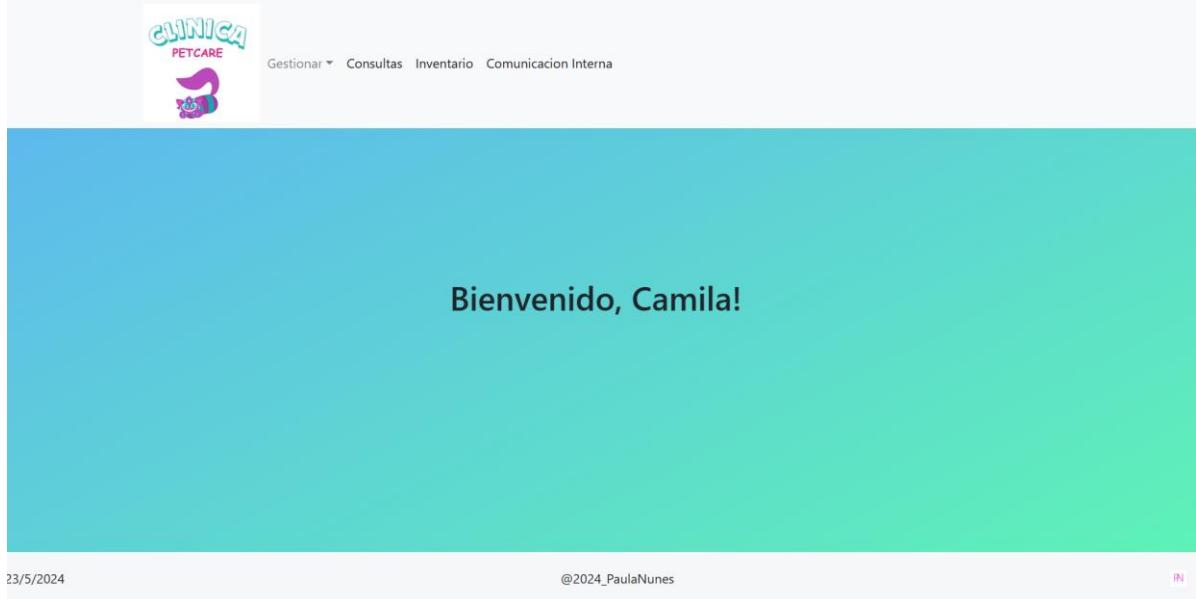
Por favor. Rellenar todos los campos.

**OK**

Si uno de los campos no es correcto, aparecerá una alerta:



- Pantalla principal



Desde la pantalla principal podemos elegir, las acciones que queremos hacer. En todas las pantallas, la barra superior y la barra inferior son comunes. En el cuerpo de la pantalla principal, damos la bienvenida al usuario, cambiando siempre el nombre de acuerdo con el usuario que inicia sesión.

- Pantalla para gestionar datos de los empleados

Pulsamos en el desplegable que hay en la barra superior, desde la pantalla principal, y elegimos Empleados.



The screenshot shows the PETCARE software interface. At the top left is the logo 'CLINICA PETCARE' with a cartoon cat icon. To the right are navigation links: 'Gestionar' (with a dropdown arrow), 'Consultas', 'Inventario', 'Comunicacion Interna', 'Pacientes', and 'Empleados'. The 'Empleados' link is highlighted with a white box.

Se abre una nueva pantalla. En el cuerpo tenemos una tabla con los datos de los usuarios cuando se registraron, filtros para buscar un empleado en concreto y tres botones: Modificar datos de empleado, eliminar empleado y activar empleado. Pero solo el ADMIN puede hacer las funcionalidades.

The screenshot shows the 'Empleados' management screen. At the top left is the 'CLINICA PETCARE' logo. Below it is a table with columns: Id, Nombre, Apellido, Email, País, and Sexo. The table contains five rows of data:

	Id	Nombre	Apellido	Email	País	Sexo
<input type="checkbox"/>	1	Paula	Nunes	paula@hotmail.com	Brazil	mujer
<input type="checkbox"/>	2	Camila	Ranzani	camila@hotmail.com	Italia	mujer
<input type="checkbox"/>	3	Sonia	Aparecida	sonia@hotmail.com	Francia	mujer
<input type="checkbox"/>	4	Selmo	Costa	selmo@hotmail.com	Portugal	hombre
<input type="checkbox"/>	5	Solange	Ranzani	solange@hotmail.com	Brazil	mujer

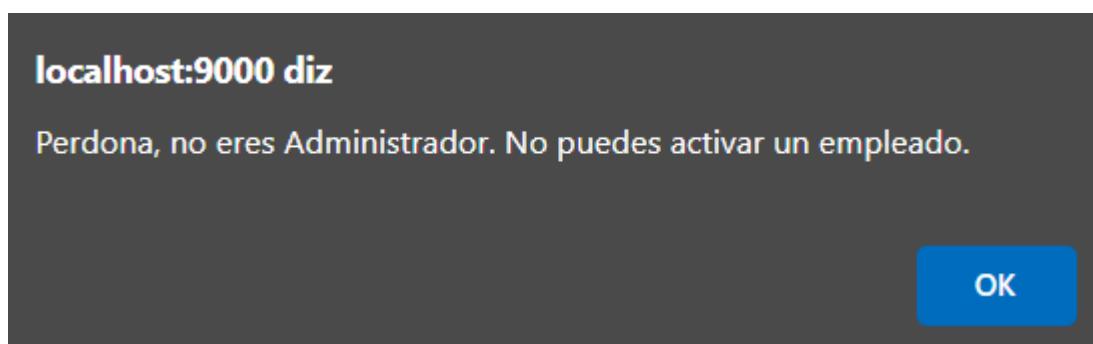
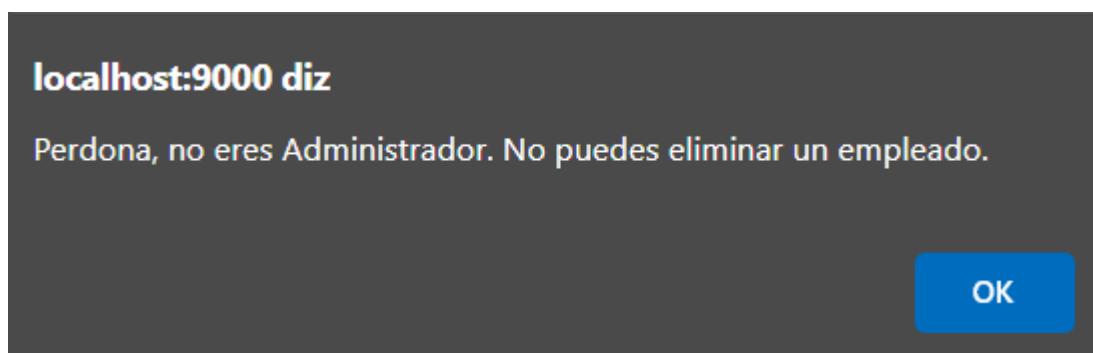
At the bottom of the table are three buttons: 'Modificar Datos Empleado', 'Eliminar Empleado', and 'Activar Empleado'. The entire interface has a light blue background.

18/5/2024

@2024\_PaulaNunes

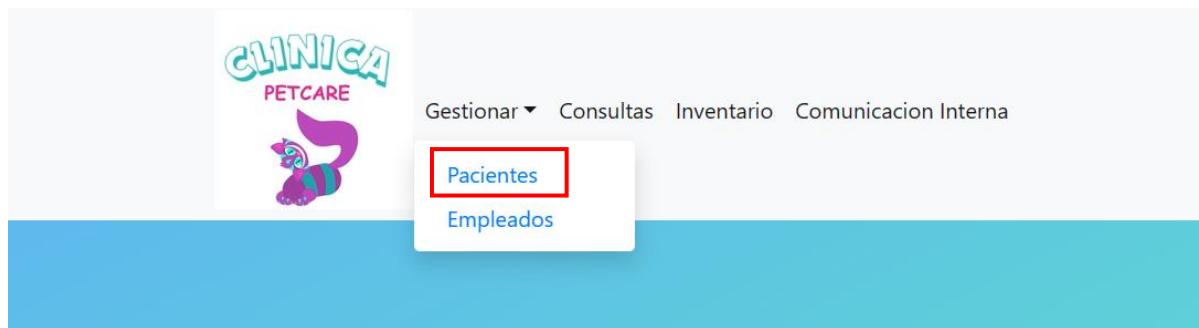
IN

The screenshot shows a modal dialog box with a dark gray background. At the top left is the text 'localhost:9000 diz'. In the center is a message: 'Perdona, no eres Administrador. No puedes modificar los datos.' (Sorry, you are not an administrator. You cannot modify the data.). At the bottom right is a blue 'OK' button.



- Pantalla para gestionar datos de los pacientes

Pulsamos en el desplegable que hay en la barra superior, desde la pantalla principal, y elegimos Pacientes.



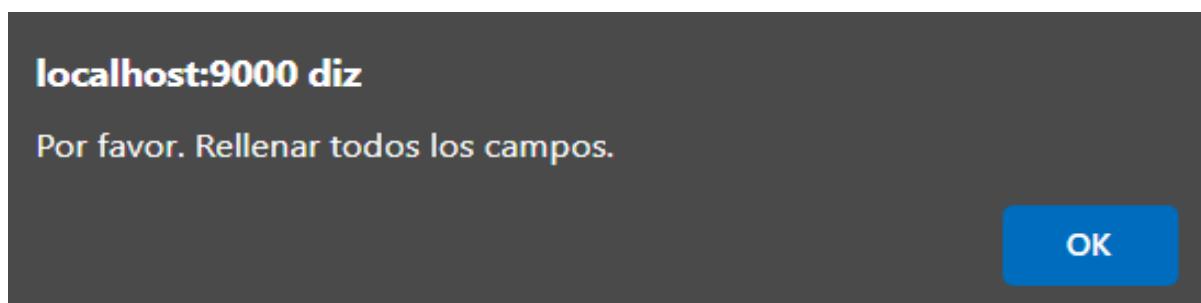
Se abre una nueva pantalla. En el cuerpo tenemos una tabla con los datos de los pacientes, filtros para buscar un paciente en concreto y cuatro botones: Añadir Paciente, Modificar Datos paciente, Eliminar paciente y Activar Paciente.

Botón Añadir Paciente:

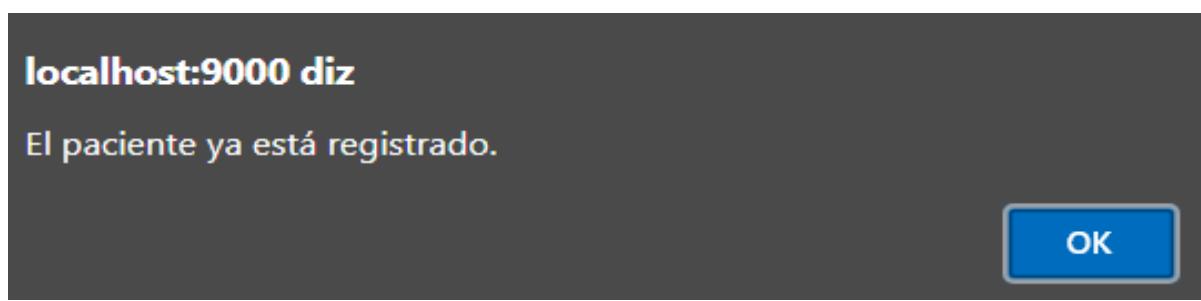
Cuando el usuario pulsa el botón, surge un popup con campos que deben ser rellenados.

The screenshot shows a software application window titled "CLINICA PETCARE". At the top, there are navigation links: "Gestionar", "Consultas", "Inventario", and "Comunicacion Interna". Below this is a table with columns "Id", "Numero Chip", "Nombre Mascota", and "Ubicación". The table contains five rows of data. Overlaid on this table is a modal dialog box with fields for "ID Chip", "DNI del Tutor", "Nombre Mascota", "Ubicación", "Edad", "Esta Activo?", and "Nombre Tutor". At the bottom of the dialog are "Salvar" and "Cancelar" buttons.

Si no rellena algún campo, aparecerá una alerta.



Importante: El número de chip no puede ser igual. Si existe aparecerá una alerta.

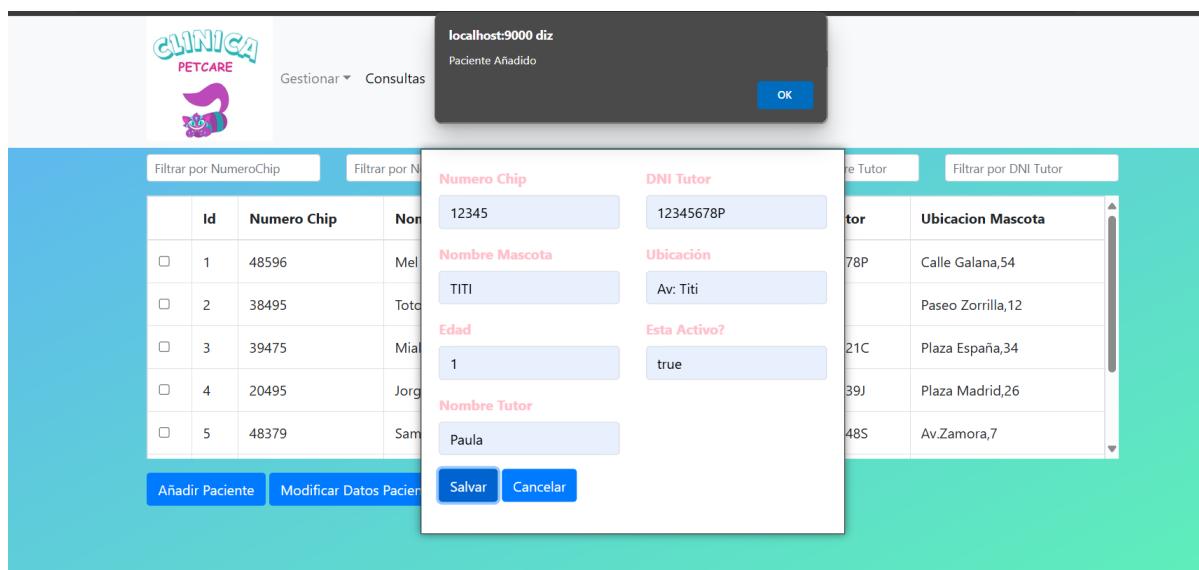


El usuario debe respetar el límite máximo de caracteres en los campos, ya que tenemos planeado un tamaño en la base de datos.

```

CREATE TABLE pacientes(
    IdPaciente int NOT NULL AUTO_INCREMENT,
    NumeroChip int NOT NULL ,
    NombreMascota VARCHAR(50) NOT NULL,
    Edad int NOT NULL,
    NombreTutor VARCHAR(50) NOT NULL,
    DNI Tutor VARCHAR(9) NOT NULL,
    UbicacionMascota VARCHAR(50) NOT NULL,
    EstaActivo boolean DEFAULT true,
    PRIMARY KEY(IdPaciente)
);
    
```

Después de llenar todos los campos, pulsa el botón 'Salvar'. Aparecerá una alerta de confirmación. Los campos del popup estarán vacíos para que el usuario pueda añadir otro paciente. Si deseas salir, pulsa el botón 'Cancelar', el popup se cerrará y la tabla se recargará.





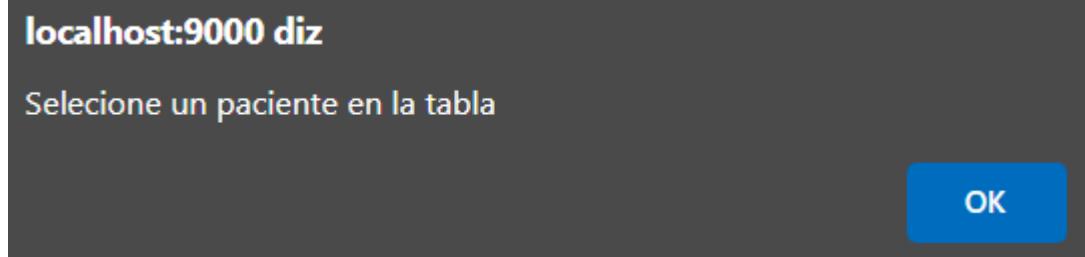
The screenshot shows a table with columns: Numero Chip, Nombre Mascota, Edad, Nombre Tutor, DNI Tutor, and Ubicación. The row for patient ID 8 is selected.

Filtrar por NumeroChip	Filtrar por Nombre Mascota	Filtrar por Edad	Filtrar por Nombre Tutor	Filtrar por DNI Tutor	
3 33473	Iván	0	Carmen	07034521C	Plaza Espana,34
4 20495	Jorge	1	Maria	37261839J	Plaza Madrid,26
5 48379	Samuka	3	Juana	78462748S	Av.Zamora,7
6 18493	Titi	4	Beatriz	11111882T	Av.Salamanca,8
7 193845	Tiko	2	Cris		Av.Segovia,12
8 12345	TITI	1	Paula	12345678P	Av. Titi

Añadir Paciente    Modificar Datos Paciente    Eliminar Paciente    Activar Paciente

### Botón Modificar Datos Paciente:

Si pulsa el botón y no selecciona un dato en la tabla aparecerá una alerta.



Cuando se seleccione al paciente y se pulse el botón, aparecerá el popup lleno con los datos del paciente.

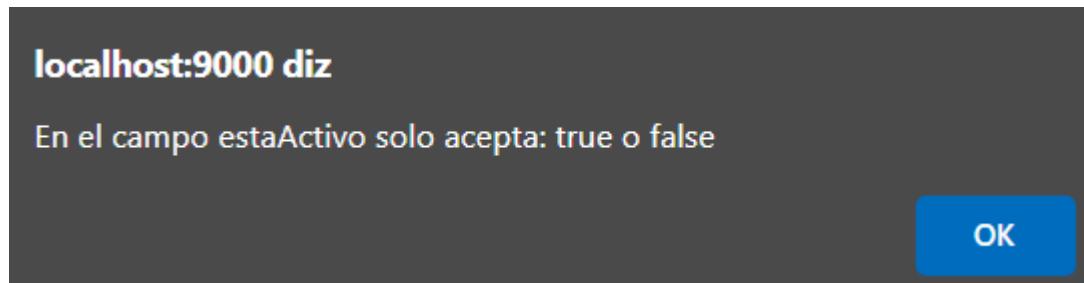
The dialog box displays the following data for patient ID 8:

Numero Chip	DNI Tutor
12345	DNI del Tutor
Nombre Mascota	Ubicación
TITI	Av. Titi
Edad	Esta Activo?
1	true
Nombre Tutor	
Paula	

Añadir Paciente    Modificar Datos Paciente    Salvar    Cancelar



Importante: No podemos cambiar el numero Chip  
Los campos EstaActivo solo permiten: false o true.

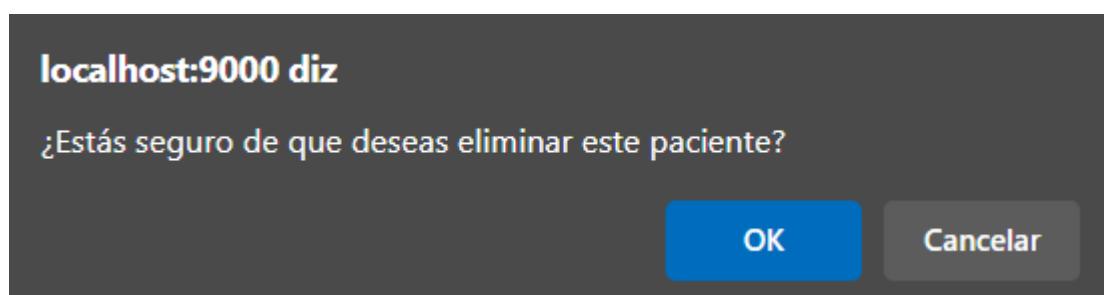


#### Botón Eliminar Paciente:

Hay que seleccionar un paciente en la tabla; de lo contrario, aparecerá una alerta.



Cuando seleccionamos al paciente y pulsamos el botón, aparece una alerta.



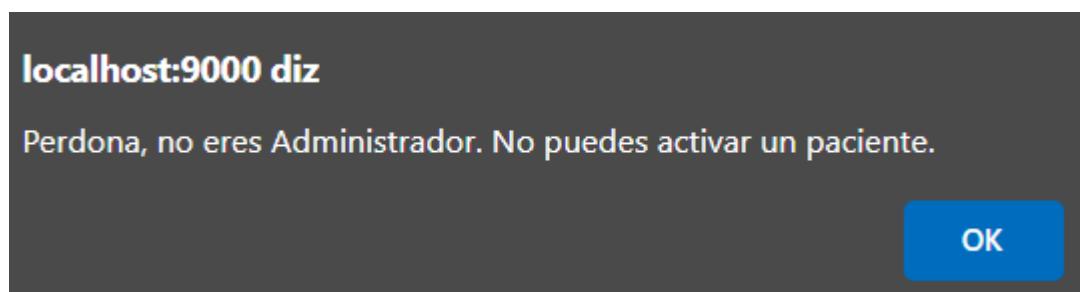
Si pulsamos Cancelar, aparecerá una alerta y la operación no se realiza.



Podemos confirmar pulsando "OK".

Botón Activar Paciente:

Solo los administradores pueden activar pacientes.



- Pantalla para el chat interno(funcionarios)

Id	Numero Chip	Nombre Mascota	Edad	Nombre Tutor	DNI Tutor	Ubicacion Mascota
1	48596	Mel	5	Paulinha	12345678P	Calle Galana,54
2	38495	Toto	6	Juan	22998811J	Paseo Zorrilla,12
3	39475	Mial	8	Carmen	87654321C	Plaza España,34
5	48379	Samuka	3	Juana	78462748S	Av.Zamora,7
7	19384	Tiko	2	Cris	62734827R	Av.Segovia,12

Desde la pantalla principal, elegimos la opción Comunicación Interna, una nueva ventana se abre.



The screenshot shows the PETCARE application interface. At the top, there is a logo for 'CLINICA PETCARE' featuring a colorful butterfly. The menu bar includes 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. Below the menu, a section titled 'Conectar con el servidor' (Connect to the server) has two buttons: 'Conectar' (Connect) and 'Desconectar' (Disconnect). A text input field labeled 'Escribe tu nickname:' is empty. A blue button labeled 'Unirse al chat' (Join the chat) is visible. On the left, the name 'Paula' is displayed. In the center, a text input field labeled 'Escribe tu mensaje:' is empty, with a blue 'Enviar mensaje' (Send message) button below it. To the right, two sections are shown: 'Mensajes recibidos:' (Received messages) and 'Usuarios en la sala:' (Users in the room), both of which are currently empty.

Todos los usuarios pueden conectar.  
Debe seguir los pasos para empezar el chat:  
4- Pulsar el botón Conectar

The screenshot shows the PETCARE application interface. A connection confirmation dialog box is overlaid on the screen, displaying 'localhost:9000 diz' and 'Conectado' (Connected) in white text on a black background, with an 'OK' button at the bottom right. The main interface below the dialog shows the 'Conectar con el servidor' section with the 'Conectar' button highlighted in green. The nickname input field is empty. The 'Unirse al chat' button is blue. The central message area is empty. The 'Mensajes recibidos:' and 'Usuarios en la sala:' sections are also empty.

- 5- Escriba tu nombre y pulse el botón Unirse al chat si el campo está vacío aparecerá una alerta.

**localhost:9000 diz**

El nickname no puede estar vacío

OK



The screenshot shows the PETCARE communication interface. At the top, there's a logo for 'CLINICA PETCARE' with a colorful paw print graphic. Below the logo, a navigation bar includes 'Gestionar', 'Consultas', 'Inventario', and 'Comunicacion Interna'. On the left, a sidebar titled 'Conectar con el servidor' has 'Conectar' and 'Desconectar' buttons. In the center, a text input field asks 'Escribe tu nickname:' with 'Paulinha' typed in, and a blue 'Unirse al chat' button below it. To the right, a message box shows 'Mensajes recibidos:' with the message 'Paulinha:\\" -- entro en el chat -- \\\"'. Below this is a section titled 'Usuarios en la sala:' showing 'Paulinha'. At the bottom, a text input field asks 'Escribe tu mensaje:' with a blue 'Enviar mensaje' button.

- 6- Ahora escriba un mensaje y pulse el botón Enviar Mensaje, si está vacío aparecerá una alerta.

**localhost:9000 diz**

El mensaje no puede estar vacío

OK

The screenshot shows the PETCARE communication interface after a successful connection. The 'Escribe tu nickname:' field still contains 'Paulinha'. The message box now shows two messages: 'Paulinha:\\\" -- entro en el chat -- \\\"' and 'Paulinha:Hola Soy Paulinha'. The 'Usuarios en la sala:' section still shows 'Paulinha'. The 'Enviar mensaje' button at the bottom is visible.

Ahora vamos a ver otro funcionario para conectar:



IES RIBERA DE CASTILLA

PETCARE

The screenshot shows the PETCARE communication interface. At the top, there is a logo for "CLINICA PETCARE" featuring a stylized purple and yellow paw print. The menu bar includes "Gestionar", "Consultas", "Inventario", and "Comunicacion Interna". On the left, a section titled "Conectar con el servidor" has two buttons: "Conectar" (green) and "Desconectar" (red). A user named "Camila" is listed. In the center, there is a text input field for "Escribe tu nickname:" with a placeholder "Camilinha" and a blue button "Unirse al chat". Below it is another text input field for "Escribe tu mensaje:" with a blue "Enviar mensaje" button. On the right, a teal sidebar displays "Mensajes recibidos:" and "Usuarios en la sala:". The "Mensajes recibidos:" area is empty. The "Usuarios en la sala:" area also contains the placeholder text.

The screenshot shows the PETCARE communication interface after a user has connected. The "Escribe tu nickname:" field now contains "Camilinha". The "Mensajes recibidos:" area shows the message "Camilinha:\\" -- entro en el chat -- \\\" followed by "Camilinha:Hola soy Camilinha". The "Usuarios en la sala:" area now lists "Camilinha, Paulinha".

The screenshot shows the PETCARE communication interface with two users connected. The "Escribe tu nickname:" field for the second user contains "Paulinha". The "Mensajes recibidos:" area shows the messages "Paulinha:\\\" -- entro en el chat -- \\\" followed by "Paulinha:Hola Soy Paulinha", "Camilinha:\\\" -- entro en el chat -- \\\"", and "Camilinha:Hola soy Camilinha". The "Usuarios en la sala:" area lists both "Camilinha, Paulinha".

## 7.2 Manual de instalación

- Requisitos básicos:
  - Instalar JDK.
  - Instalar Spring Tool Suite 4.
  - Instalar MYSQL.
- Para ejecutar la aplicación:
  - Abrir el GitHub y bajar el proyecto.
  - Abrir Spring Tool Suite 4 con el proyecto.
  - Ejecutar.
  - En la web: localhost:9000

## 8 Conclusiones y posibles ampliaciones

- Conclusiones:

Durante el desarrollo del proyecto, he seguido una metodología basada en el modelo de ciclo de vida en cascada, lo que ha permitido avanzar de manera estructurada a través de las diferentes fases de análisis, diseño, implementación, pruebas, despliegue y mantenimiento (mismo que no por completo). Esta metodología ha proporcionado una guía clara para la planificación y ejecución del proyecto, permitiendo una gestión eficiente de los recursos y un control adecuado sobre el avance del trabajo.

He utilizado una arquitectura Modelo-Vista-Controlador (MVC) para organizar el código de manera modular y separar las preocupaciones, lo que ha facilitado la escalabilidad y la mantenibilidad de la aplicación a lo largo del tiempo. Además, se han empleado tecnologías y herramientas modernas como Java, Spring Framework, JavaScript, HTML, CSS, y Bootstrap, entre otras, para garantizar un desarrollo ágil y robusto.

La aplicación resultante ofrece una interfaz intuitiva y amigable que permite a los usuarios llevar a cabo tareas como la gestión de pacientes, la comunicación interna a través del chat, y la administración de datos de manera eficiente y efectiva.

En conclusión, este proyecto ha sido una experiencia enriquecedora que ha permitido aplicar conocimientos teóricos en un contexto práctico y desarrollar habilidades técnicas y de gestión de proyectos.
- Posibles ampliaciones:
  - Integración con sistemas de terceros: Ampliar la aplicación para integrarla con sistemas de terceros, como sistemas de facturación, sistemas de gestión de inventario o sistemas de citas online para clientes.
  - Agregar funcionalidades adicionales, como la capacidad de programar recordatorios de citas, generar informes estadísticos sobre el

rendimiento de la clínica veterinaria o implementar un sistema de gestión de ventas para productos relacionados con mascotas.

## 9 Bibliografía

- [Curso em Vídeo - Aprenda informática com cursos grátis. \(cursoemvideo.com\)](#)
- [Inicio | Academia OW \(openwebinars.net\)](#)
- [Curso Java. MVC II. Vídeo 249 \(youtube.com\)](#)

## 10 Anexos

Código fuente:

### Pantalla registrar.

```

1 package com.example.demo.Controllers;
2
3 import java.util.HashMap;
4
5
6 @Controller
7 public class UsuarioController {
8
9     @Autowired
10    private MessageSource meassageSource;
11
12    @Autowired
13    private HttpServletRequest request;
14
15
16    //*****método para mostrar la vista(archivo .jsp) y tambien hacer lista(datos que busco)
17    @RequestMapping("/usuario")
18    public ModelAndView usuario() {
19
20        ModelAndView mv = new ModelAndView();
21        Map<String, Object> model = new HashMap<>();
22        try {
23            mv = new ModelAndView("usuario","model",model);
24        }catch (Exception e){
25            e.getMessage();
26        }
27        return mv;
28    }
29
30
31    //método para crear la vista(archivo .json) y tambien hacer lista(datos que busco)
32    @RequestMapping("/crearUsuario")
33    public Object crearUsuario() {
34
35        Object crearUsuario(
36            @RequestParam(value="nombre") String nombre,
37            @RequestParam(value="apellido") String apellido,
38            @RequestParam(value="email") String email,
39            @RequestParam(value="contrasena") String contrasena,
40            @RequestParam(value="pais") String pais,
41            @RequestParam(value="sexo") String sexo
42        ){
43
44            Map<String, Object> model = new HashMap<>();
45            try {
46                //Verifican se o email já existe
47                if (!Usuario.existeUsuario(email)) {
48                    // Se o email não existe, inserir o usuário
49                    UsuarioPOJO datos = new UsuarioPOJO(nombre, apellido, email, contrasena, pais, sexo);
50                    Usuario.insertarUsuario(datos);
51                    model.put("mensaje", "Registro confirmado!");
52                    model.put("success", true);
53                } else {
54                    // Se o email já existe, retornar uma mensagem indicando isso
55                    model.put("mensaje", "El correo electrónico ya está registrado.");
56                    model.put("success", false);
57                }
58            } catch(Exception e) {
59                e.printStackTrace();
60                model.put("mensaje", "Ocurrió un error al crear/modificar el usuario");
61                model.put("success", false);
62            }
63        }
64    }
65
66    return model;
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81

```



```
/*
 * mapeando para mod Usuario
 * @RequestMapping("/modusuario.json")
 * @ResponseBody
 public Object modificarUsuario(
     @RequestParam(value="idusuario") int idUsuario,
     @RequestParam(value="nombre") String nombre,
     @RequestParam(value="apellido") String apellido,
     @RequestParam(value="email") String email,
     @RequestParam(value="pais") String pais,
     @RequestParam(value="sexo") String sexo,
     @RequestParam(value="DNI") String DNI,
     @RequestParam(value="telefono") String telefono,
     @RequestParam(value="esAdmin") String esAdmin,
     @RequestParam(value="estaActivo") String estaActivo
){}

    ...
    Map<String, Object> model = new HashMap<>();
    try {
        // Verificar se pais esta na lista de opciones
        if (Usuario.existePais(pais)) {
            // Verificar se sexo es mujer o hombre
            if (sexo.equals("mujer") || sexo.equals("hombre")) {
                // Verificar se esAdmin es true o false
                if (esAdmin.equals("true") || esAdmin.equals("false")) {
                    // Verificar se estaActivo es true o false
                    if (estaActivo.equals("true") || estaActivo.equals("false")) {
                        boolean esAdm = Boolean.parseLong(esAdmin);

                        UsuarioPOJO datos = new UsuarioPOJO(idUsuario, nombre, apellido, email, pais, sexo, DNI, telefono, esAdm, Boolean.valueOf(estaActivo));
                        Usuario.modificarUsuario(datos);

                        model.put("mensajeMod", "OK");
                        model.put("success", true);
                    } else {
                        model.put("mensajeMod", "En el campo estaActivo solo acepta: true o false");
                        model.put("success", false);
                    }
                } else {
                    model.put("mensajeMod", "En el campo esAdmin solo acepta: true o false");
                    model.put("success", false);
                }
            } else {
                // Se o pais nao existe, retornar uma mensagem indicando isso
                model.put("mensajeMod", "En el campo pais solo acepta: Espana, Brazil, Italia, Francia y Portugal");
                model.put("success", false);
            }
        } else {
            model.put("success", false);
        }
    } catch(Exception e) {
        e.printStackTrace();
        model.put("mensaje", "Ocurrio un error al crear/modificar el usuario");
        model.put("success", false);
    }
    return model;
}
 *****/
```



```
*****  
//mapeando para eliminar Usuario  
@RequestMapping("/eliminarUsuario.json")  
@ResponseBody  
public Object eliminarUsuario(  
    @RequestParam(value="idUsuario") int idUsuario  
{  
  
    Map<String, Object> model = new HashMap<>();  
    try {  
  
        Usuario.eliminarUsuario(idUsuario);  
  
        model.put("mensajeElim", "OK");  
        model.put("success", true);  
  
    } catch(Exception e) {  
        e.printStackTrace();  
        model.put("mensaje", "Ocurrió un error al eliminar el usuario");  
        model.put("success", false);  
    }  
    return model;  
}  
*****
```

```
105  
106 //*****  
107 //mapeando para activar Usuario  
108 @RequestMapping("/activarUsuario.json")  
109 @ResponseBody  
110 public Object activarUsuario(  
111     @RequestParam(value="idUsuario") int idUsuario  
112     ){  
113  
114     Map<String, Object> model = new HashMap<>();  
115     try {  
116         Usuario.activarUsuario(idUsuario);  
117  
118         model.put("mensajeAct", "OK");  
119         model.put("success", true);  
120  
121     } catch(Exception e) {  
122         e.printStackTrace();  
123         model.put("mensaje", "Ocurrió un error al activar el usuario");  
124         model.put("success", false);  
125     }  
126     return model;  
127 }  
128 *****  
129 }  
130 *****
```



```
1 |<%@ page language="java" contentType="text/html; charset=ISO-8859-1" pageEncoding="ISO-8859-1"%>
2 |<!DOCTYPE html>
3 |<html>
4 |<head>
5 |<meta charset="ISO-8859-1">
6 |<title><%= ResourceBundle.getBundle("code").getString("registrar") %></title>
7 |<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
8 |<link rel="icon" href="/icon.ico" type="image/x-icon">
9 |
10|<style>
11|body {
12|    display: flex;
13|    justify-content: center; /* Centraliza o conteúdo horizontalmente */
14|    align-items: center; /* Centraliza o conteúdo verticalmente */
15|    height: 100vh; /* Ocupa toda a altura da tela */
16|    margin: 0; /* Remove a margem padrão */
17|    background: linear-gradient(to bottom right, #5EB1F5, #5EF5B4); /* Fundo degradê de azul para verde */
18|
19|}</style>
20|.container {
21|    text-align: center; /* Alinha o conteúdo no centro horizontal */
22|    border: 1px solid #cccc; /* Adiciona uma borda para visualização */
23|    padding: 20px;
24|    border-radius: 10px;
25|    background-color: #ffffff; /* Cor de fundo do contêiner */
26|    box-shadow: 0 4px 8px rgba(0, 0, 0, 0.5); /* Adiciona uma sombra de fundo */
27|    max-width: 400px; /* Define a largura máxima do contêiner */
28|    width: 100%; /* Garante que o contêiner não ultrapasse a largura máxima */
29|}</div>
30|h1{
31|    color: #FDB7D7;
32|}
33|
34|/* Cor padrão dos botões */
35|.btn-primary {
36|    background-color: #1EE4F2;
37|    border-color: #1EE4F2;
38|
39|}</style>
40|/* Cor dos botões ao passar o mouse */
41|.btn-primary:hover {
42|    background-color: #1EF2C8;
43|    border-color: #1EF2C8;
44|}</div>
45|</body>
46|</html>
47|<div class="container">
48|<form action="" method="post" class="row">
49|<div class="col-md-12">
50|    <label><h1><%= ResourceBundle.getBundle("code").getString("registrar") %></h1></label>
51|</div>
52|<div class="col-md-4">
53|    <div class="form-group">
54|        <label for=""><h5><%= ResourceBundle.getBundle("code").getString("registrar.nombre") %></h5></label>
55|    </div>
56|    <div class="form-group">
57|        <label for=""><h5><%= ResourceBundle.getBundle("code").getString("registrar.apellido") %></h5></label>
58|    </div>
59|    <div class="form-group">
60|        <label for=""><h5><%= ResourceBundle.getBundle("code").getString("registrar.usuario") %></h5></label>
61|    </div>
62|    <div class="form-group">
63|        <label for=""><h5><%= ResourceBundle.getBundle("code").getString("registrar.contrasena") %></h5></label>
64|    </div>
65|    <div class="form-group">
66|        <label for=""><h5><%= ResourceBundle.getBundle("code").getString("registrar.pais") %></h5></label>
67|    </div>
68|    <div class="form-group">
69|        <label for=""><h5><%= ResourceBundle.getBundle("code").getString("registrar.sexo") %></h5></label>
70|    </div>
71|    <div class="form-group">
72|        <label for=""><h5><%= ResourceBundle.getBundle("code").getString("registrar.senha") %></h5></label>
73|    </div>
74|<div class="col-md-8">
```



```

75<div class="form-group">
76    <input type="text" class="form-control" id="nombre" name="nombre" maxlength="15" required>
77</div>
78<div class="form-group">
79    <input type="text" class="form-control" id="apellido" name="apellido" maxlength="40" required>
80</div>
81<div class="form-group">
82    <input type="text" class="form-control" id="email" name="usuario" maxlength="22" required>
83</div>
84<div class="form-group">
85    <input type="password" class="form-control" id="contrasena" name="contrasena" maxlength="10" required>
86</div>
87<div class="form-group">
88    <select id="desplegablePais" class="form-control form-control-sm">
89        <option value="España"><%= ResourceBundle.getBundle("code").getString("registrar.pais.España") %></option>
90        <option value="Brazil"><%= ResourceBundle.getBundle("code").getString("registrar.pais.Brazil") %></option>
91        <option value="Italia"><%= ResourceBundle.getBundle("code").getString("registrar.pais.Italia") %></option>
92        <option value="Francia"><%= ResourceBundle.getBundle("code").getString("registrar.pais.Francis") %></option>
93        <option value="Portugal"><%= ResourceBundle.getBundle("code").getString("registrar.pais.Portugal") %></option>
94    </select>
95</div>
96<div class="form-group">
97    <input type="radio" name="sexo" value="homem" id="sexo_H"><%= ResourceBundle.getBundle("code").getString("registrar.sexo.hombre") %>
98    <input type="radio" name="sexo" value="mulher" id="sexo_M"><%= ResourceBundle.getBundle("code").getString("registrar.sexo.mujer") %>
99</div>
100</div>
101<div class="col-md-12">
102    <div class="col-md-12 mb-5">
103        <a href="#" id="btnSalir" type="button" class="btn btn-primary ">!-- volta para login -->
104        <%= ResourceBundle.getBundle("code").getString("registrar.salir") %>
105    </a>
106    <a href="#" javascript:crearUsuario() id="btnConfirmar" type="button" class="btn btn-primary ">
107        <%= ResourceBundle.getBundle("code").getString("registrar.confirmar") %>
108    </a>
109</div>
110</div>

111    </div>
112</form>
113</div>
114<%@ include file="/WEB-INF/views/footer.jsp"%>
115<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script><!-- Link para o jQuery fazer a solicitação AJAX -->
116<script>
117    function crearUsuario() {
118        // Verifica se os campos obrigatórios estão preenchidos
119        if ($("#nombre").val() === "" || $("#apellido").val() === "" || $("#email").val() === "" || $("#contrasena").val() === "" || $("#desplegablePais").val() === "") {
120            alert("Por favor. Rellenar todos los campos.");
121            return;
122        }
123
124        // Verifica o formato do campo de email(usuario), somente letras e numeros, e deve terminar @hotmail.com
125        var formatoEmail = /^[a-zA-Z0-9]+@[hotmail\.com$/;
126
127
128        if (!formatoEmail.test($("#email").val())) {
129            alert("Por favor, el campo 'Usuario' es un correo. Cantidad de caracteres permitida en el campo (22). Formato del correo: Solo es permitido letras y numeros");
130            return;
131        }
132
133        // Codifica a senha
134        crypto.subtle.digest('SHA-256', new TextEncoder().encode($("#contrasena").val())).then(function(hash) {
135            // Converte o hash para uma string hexadecimal
136            var hashedPassword = Array.from(new Uint8Array(hash)).map(b => b.toString(16).padStart(2, '0')).join('');
137
138            // Envia os dados apenas se todos os campos estiverem preenchidos
139            $.post("creaUsuario.json", {
140                nome: $("#nombre").val(),
141                apellido: $("#apellido").val(),
142                email: $("#email").val(),
143                contrasena: hashedPassword, // Envie a senha codificada
144                pais: $("#desplegablePais").val(),
145                sexo: $("input[name='sexo']:checked").val()
146            })
147        })
148    }
149</script>

```



```
147      //o .done() é usado para executar código específico quando a requisição AJAX é bem-sucedida,
148      //permitindo que atualize a interface do usuário,
149      //processe os dados recebidos ou execute outras operações necessárias após a conclusão da requisição.
150      .done(function (dados) {
151
152          // Verifica a mensagem recebida do servidor
153          if (dados.mensaje === "Registro confirmado!") {
154              // Limpa os campos após o registro ser confirmado
155              $("#nombre").val("");
156              $("#apellido").val("");
157              $("#email").val("");
158              $("#contrasena").val("");
159              $("#desplegablePais").val("");
160              $("input[name='sexo']").prop('checked', false);
161              alert("Registro confirmado!");
162          } else {
163              alert(datos.mensaje); //el alert aparece el mensaje que ha pasado el model del controller
164          }
165      })
166
167
168      //o .fail() é usado para lidar com erros que ocorrem durante a execução de uma requisição AJAX,
169      //permitindo que exiba mensagens de erro,
170      //registre os erros no console ou tome outras medidas necessárias para lidar com a situação de erro.
171      .fail(function (jqXHR, textStatus, errorThrown) {
172          console.error("Error al enviar la solicitud: ", errorThrown);
173          alert("Ocorreu um erro ao enviar a solicitação. Por favor, tente novamente.");
174      });
175  });
176 }
177 </script>
178
179 </body>
180 </html>
```