****

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** | **Foto actual del alumno** |

Contenido

[1 Identificación proyecto 3](#_Toc166671644)

[2 Organización de la memoria 3](#_Toc166671645)

[3 Descripción general del proyecto 3](#_Toc166671646)

[3.1 Objetivos 3](#_Toc166671647)

[3.2 Cuestiones metodológicas 3](#_Toc166671648)

[3.3 Entorno de trabajo (tecnologías de desarrollo y herramientas) 4](#_Toc166671649)

[4 Descripción general del producto 5](#_Toc166671650)

[4.1 Visión general del sistema: límites del sistema, funcionalidades básicas, usuarios y/o otros sistemas con los que pueda interactuar. 5](#_Toc166671651)

[4.2 Descripción breve de métodos, técnicas o arquitecturas(m/t/a) utilizadas. 6](#_Toc166671652)

[4.3 Despliegue de la aplicación indicando plataforma tecnológica, instalación de la aplicación y puesta en marcha 6](#_Toc166671653)

[5 Planificación y presupuesto 6](#_Toc166671654)

[6 Documentación Técnica: análisis, diseño, implementación y pruebas. 8](#_Toc166671655)

[6.1 Especificación de requisitos 8](#_Toc166671656)

[6.2 Análisis del sistema 8](#_Toc166671657)

[6.3 Diseño del sistema: 8](#_Toc166671658)

[6.3.1 Diseño de la Base de Datos 8](#_Toc166671659)

[6.3.2 Diseño de la Interfaz de usuario. 8](#_Toc166671660)

[6.3.3 Diseño de la Aplicación. 8](#_Toc166671661)

[6.4 Implementación: 8](#_Toc166671662)

[6.4.1 Entorno de desarrollo. 8](#_Toc166671663)

[6.4.2 Estructura del código. 8](#_Toc166671664)

[6.4.3 Cuestiones de diseño e implementación reseñables. 8](#_Toc166671665)

[6.5 Pruebas. 8](#_Toc166671666)

[7 Manuales de usuario 8](#_Toc166671667)

[7.1 Manual de usuario 8](#_Toc166671668)

[7.2 Manual de instalación 8](#_Toc166671669)

[8 Conclusiones y posibles ampliaciones 8](#_Toc166671670)

[9 Bibliografía 9](#_Toc166671671)

[10 Anexos 9](#_Toc166671672)

# Identificación proyecto

# Organización de la memoria

# 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.

## 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.

1. 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.

## 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.

## 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.

# Descripción general del producto

## 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.

## 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.

## 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.

# 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€/hora = 9.000€

* 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 meses \* 50€/mes = 300€

* Coste total del proyecto:

. Coste de desarrollo del código: 9.000

. Coste del hardware: 500€

. Coste del hosting: 300€

. Coste total: 9.800€

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

## Especificación de requisitos

## Análisis del sistema

## Diseño del sistema:

### Diseño de la Base de Datos

### Diseño de la Interfaz de usuario.

### Diseño de la Aplicación.

## Implementación:

### Entorno de desarrollo.

### Estructura del código.

### Cuestiones de diseño e implementación reseñables.

## Pruebas.

# Manuales de usuario

## 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.

Diagrama

Descripción generada automáticamente con confianza media

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

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

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:

Texto

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

* Pantalla principal

Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente

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.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

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 empleados, eliminar empleado y activar empleado.

Tabla

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Aplicación, Sitio web

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

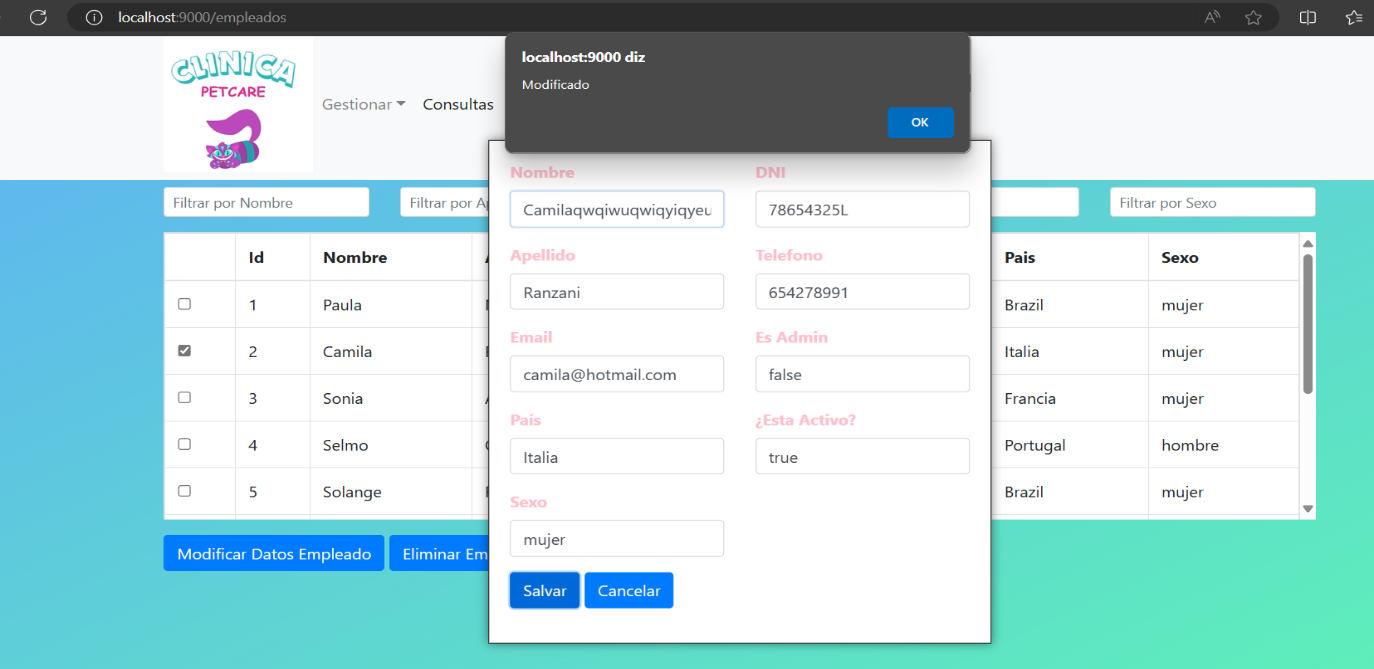
Descripción generada automáticamente

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

Imagen que contiene Texto

Descripción generada automáticamente

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



Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

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

Forma

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza baja

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

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

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'.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

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.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Texto

Descripción generada automáticamente con confianza media

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.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Forma

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Si pulsamos em 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.

Imagen que contiene Escala de tiempo

Descripción generada automáticamente

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.

Tabla

Descripción generada automáticamente

Botón Añadir Paciente:

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si no rellena algún campo, aparecerá una alerta.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Importante: El numero de chip no puede ser existe, es la chave. Si existe aparecerá una alerta.

Texto

Descripción generada automáticamente con confianza media

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

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Después de rellenar 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.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Botón Modificar Datos Paciente:

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

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Importante: No podemos cambiar el numero Chip

Los campos EstaActivo solo permiten: false o true.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Botón Eliminar Paciente:

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

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

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

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

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

Forma

Descripción generada automáticamente con confianza media

Podemos confirmar pulsando "OK”.

Forma

Descripción generada automáticamente

Botón Activar Paciente:

* Pantalla para el chat interno(funcionarios)

## 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

# 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.

# Bibliografía

* [Curso em Vídeo - Aprenda informática com cursos grátis. (cursoemvideo.com)](https://www.cursoemvideo.com/)
* [Inicio | Academia OW (openwebinars.net)](https://openwebinars.net/academia/)
* [Curso Java. MVC II. Vídeo 249 (youtube.com)](https://www.youtube.com/watch?v=Jl8qoHqrjd8&t=475s)

# Anexos