	<b>Taller de Lenguajes de programación I</b>	Código:	mpb-01
		Versión:	01
		Páginas	
		Fecha emisión	Fecha actual


## **Guía Trabajo de estudio No : 1** **Trabajo Practico Taller programación I**

---




---

Manolo Pájaro Borrás

	<b>Taller de Lenguajes de programación I</b>	Código:	mpb-01
		Versión:	01
		Páginas	
		Fecha emisión	Fecha actual

### Objetivo:

Realizar una Aplicación utilizando el modelo Vista Controlador y trabajando con archivos de texto

### Actividades:

Para estructurar y trabajar el aplicativo de administración de una veterinaria en Java escritorio utilizando el modelo MVC, puedes seguir los siguientes pasos:

#### 1. Estructura del Proyecto:


- Crea un proyecto en Java y organiza los paquetes según el patrón MVC:
- modelo: Para las clases que representan los datos y la lógica de negocio.
- vista: Para las interfaces gráficas.
- controlador: Para manejar la interacción entre la vista y el modelo.

#### 2. Clases Modelo:

- Crea las clases modelo que representarán las entidades y la lógica de negocio:
- Persona: Con atributos cédula, nombre, edad y rol.
- Mascota: Con los atributos mencionados.
- HistoriaClinica: Con los detalles de la historia clínica.
- OrdenMedica: Con los detalles de las órdenes médicas.
- Factura: Con los detalles de las facturas.

#### 3. Clases Vista:

- Diseña las interfaces gráficas para interactuar con el usuario:
- RegistroVeterinarioFrame: Ventana para registrar veterinarios.
- RegistroVendedorFrame: Ventana para registrar vendedores.
- RegistroDuenoMascotaFrame: Ventana para registrar dueños de mascotas.
- RegistroMascotaFrame: Ventana para registrar mascotas.
- RegistroHistoriaClinicaFrame: Ventana para registrar historias clínicas.
- RegistroFacturaFrame: Ventana para registrar facturas.
- ConsultaHistoriaClinicaFrame: Ventana para consultar historias clínicas.

	<b>Taller de Lenguajes de programación I</b>	Código:	mpb-01
		Versión:	01
		Páginas	
		Fecha emisión	Fecha actual

- ConsultaOrdenesFrame: Ventana para consultar órdenes.

#### 4. Clases Controlador:


- Crea las clases controlador que manejarán la interacción entre la vista y el modelo:
- RegistroVeterinarioController: Controlador para la ventana de registro de veterinarios.
- RegistroVendedorController: Controlador para la ventana de registro de vendedores.
- ... y así sucesivamente para las demás ventanas.

#### 5. Implementación del MVC:

- En los controladores, maneja las interacciones de la vista y actualiza el modelo en consecuencia.
- En las vistas, muestra la información y solicita la entrada del usuario.
- En el modelo, maneja la lógica de negocio y el almacenamiento temporal de datos.
- 6. Gestión de Sesiones:
- Crea una clase Sesion que almacene la información del usuario actual y el rol.
- Permite cambiar entre roles manteniendo la información del usuario actual.

#### 7. Utilización de Vectores:

- Crea vectores en la clase principal que representa la veterinaria para almacenar la información en memoria.
- Usa los vectores para almacenar veterinarios, vendedores, dueños, mascotas, historias clínicas, órdenes, etc.

	<b>Taller de Lenguajes de programación I</b>	Código:	mpb-01
		Versión:	01
		Páginas	
		Fecha emisión	Fecha actual

## 8. Pruebas y Validaciones:

- Realiza pruebas exhaustivas de las diferentes funcionalidades.
- Asegúrate de que los roles tengan acceso correcto y se pueda cambiar entre ellos.

## 9. Documentación:

- Crea documentación que explique la estructura del proyecto, las clases y sus responsabilidades.
- Agrega comentarios en el código para explicar la funcionalidad clave.

## 10. Interfaz Gráfica:

- Diseña interfaces amigables y coherentes para cada ventana.
- Asegúrate de que el flujo de navegación entre ventanas sea claro y lógico.

*Recuerda que este es un enfoque general y simplificado. Tendrás que ajustar y detallar cada componente según las necesidades específicas del proyecto. La clave es descomponer el proyecto en piezas más pequeñas y manejables, y luego integrarlas para lograr un aplicativo completo y funcional.*