

DISEÑO DE SISTEMAS

Juan Carlos Molina Lozano
Docente

PROPÓSITO DE LA CLASE

Comprender el patrón MVC y su importancia en el desarrollo de software, identificar sus componentes y relacionar la teoría con un ejemplo práctico y sencillo.

Several white lines of varying lengths and slopes are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

OBJETIVOS DE APRENDIZAJE

- Explicar qué es el patrón MVC.
 - Describir cada componente: Modelo, Vista y Controlador.
 - Mostrar cómo se comunican entre sí.
 - Aplicarlo a un ejemplo sencillo y práctico.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

INTRODUCCIÓN

El patrón MVC (Modelo-Vista-Controlador) nace en 1979, creado por Trygve Reenskaug, un investigador noruego que trabajaba en Xerox PARC (el famoso laboratorio de innovación en Estados Unidos).

Contexto de nacimiento

En esa época, las interfaces gráficas (ventanas, botones, formularios) empezaban a desarrollarse.

Antes de eso, la mayoría del software era de línea de comandos: el usuario escribía texto, no usaba el mouse ni hacía clic en pantallas.

Crear interfaces gráficas era un gran reto porque todo (datos, pantallas, interacciones) estaba mezclado en un solo bloque de código, lo que hacía difícil mantener o cambiar las aplicaciones.

INTRODUCCIÓN

Qué problema resolvía MVC?

Reenskaug propuso separar el software en tres partes para que fuera más ordenado, fácil de modificar y más flexible:

- Modelo
 - Vista
 - Controlador
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

QUÉ ES MVC?

MVC significa: Modelo - Vista - Controlador.

Es un patrón de arquitectura de software que separa una aplicación en tres componentes principales:

- Modelo → Gestiona los datos y las reglas de negocio.
- Vista → Gestiona cómo se muestran los datos al usuario.
- Controlador → Interpreta las acciones del usuario y actualiza el Modelo o la Vista.

EXPLICACIÓN DE CADA COMPONENTE

Componente

Función

Ejemplo real

Modelo

Se encarga de los **datos**: almacenarlos, procesarlos, validarlos.

La base de datos o una estructura que guarda información como "nombre de usuario" o "lista de productos".

Vista

Muestra la **información** al usuario.

Una pantalla, una página web, o una app móvil que ves en tu dispositivo.

Controlador

Recibe las **acciones del usuario** (clics, formularios) y decide qué hacer.

Cuando presionas "Agregar al carrito" en Amazon, el controlador gestiona esa acción.

CÓMO SE COMUNICAN?

- El Controlador recibe la acción del usuario.
 - El Modelo es consultado o modificado por el Controlador.
 - La Vista se actualiza para mostrar el nuevo estado del Modelo.
- 
- Several white lines of varying lengths and orientations are positioned in the bottom right corner of the slide, creating a modern, abstract graphic element.

EJEMPLO

Modelo

```
public class Libro {  
    private String titulo;  
    private String autor;  
  
    public Libro(String titulo, String autor) {  
        this.titulo = titulo;  
        this.autor = autor;  
    }  
  
    public String getTitulo() {  
        return titulo;  
    }  
  
    public String getAutor() {  
        return autor;  
    }  
}
```

```
public class Producto {  
    private String nombre;  
    private double precio;  
  
    public Producto(String nombre, double precio) {  
        this.nombre = nombre;  
        this.precio = precio;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public double getPrecio() {  
        return precio;  
    }  
}
```

```
public class Usuario {  
    private String nombre;  
    private String email;  
  
    public Usuario(String nombre, String email) {  
        this.nombre = nombre;  
        this.email = email;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public String getEmail() {  
        return email;  
    }  
}
```

EJEMPLO

Controlador

```
public class CtrlLibro {  
    private VistaLibro vista;  
  
    public CtrlLibro(VistaLibro vista) {  
        this.vista = vista;  
    }  
  
    public void mostrarLibro(Libro libro) {  
        vista.mostrarLibro(libro);  
    }  
}
```

```
public class CtrlProducto {  
    private VistaProducto vista;  
  
    public CtrlProducto(VistaProducto vista) {  
        this.vista = vista;  
    }  
  
    public void mostrarProducto(Producto producto) {  
        vista.mostrarProducto(producto);  
    }  
}
```

```
public class CtrlUsuario {  
  
    private VistaUsuario vista;  
  
    public CtrlUsuario(VistaUsuario vista) {  
        this.vista = vista;  
    }  
  
    public void mostrarUsuario(Usuario usuario) {  
        vista.mostrarUsuario(usuario);  
    }  
}
```

EJEMPLO

Controlador

```
public class CtrlLibro {  
    private VistaLibro vista;  
  
    public CtrlLibro(VistaLibro vista) {  
        this.vista = vista;  
    }  
  
    public void mostrarLibro(Libro libro) {  
        vista.mostrarLibro(libro);  
    }  
}
```

```
public class CtrlProducto {  
    private VistaProducto vista;  
  
    public CtrlProducto(VistaProducto vista) {  
        this.vista = vista;  
    }  
  
    public void mostrarProducto(Producto producto) {  
        vista.mostrarProducto(producto);  
    }  
}
```

```
public class CtrlUsuario {  
  
    private VistaUsuario vista;  
  
    public CtrlUsuario(VistaUsuario vista) {  
        this.vista = vista;  
    }  
  
    public void mostrarUsuario(Usuario usuario) {  
        vista.mostrarUsuario(usuario);  
    }  
}
```

EJEMPLO

Vista

```
public class VistaLibro {  
  
    public void mostrarLibro(Libro libro) {  
        System.out.println("Libro: " + libro.getTitulo() + " | Autor: " + libro.getAutor());  
    }  
  
}
```

```
public class VistaProducto {  
  
    public void mostrarProducto(Producto producto) {  
        System.out.println("Producto: " + producto.getNombre() + " | Precio: $" + producto.getPrecio());  
    }  
  
}
```

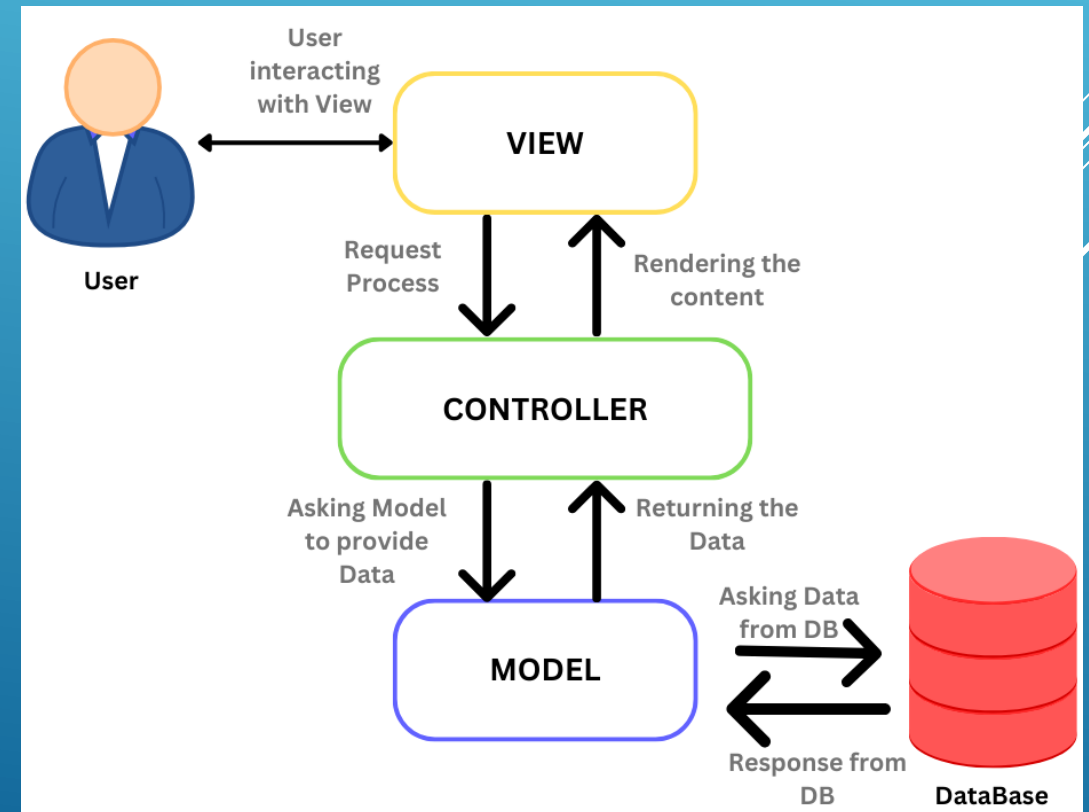
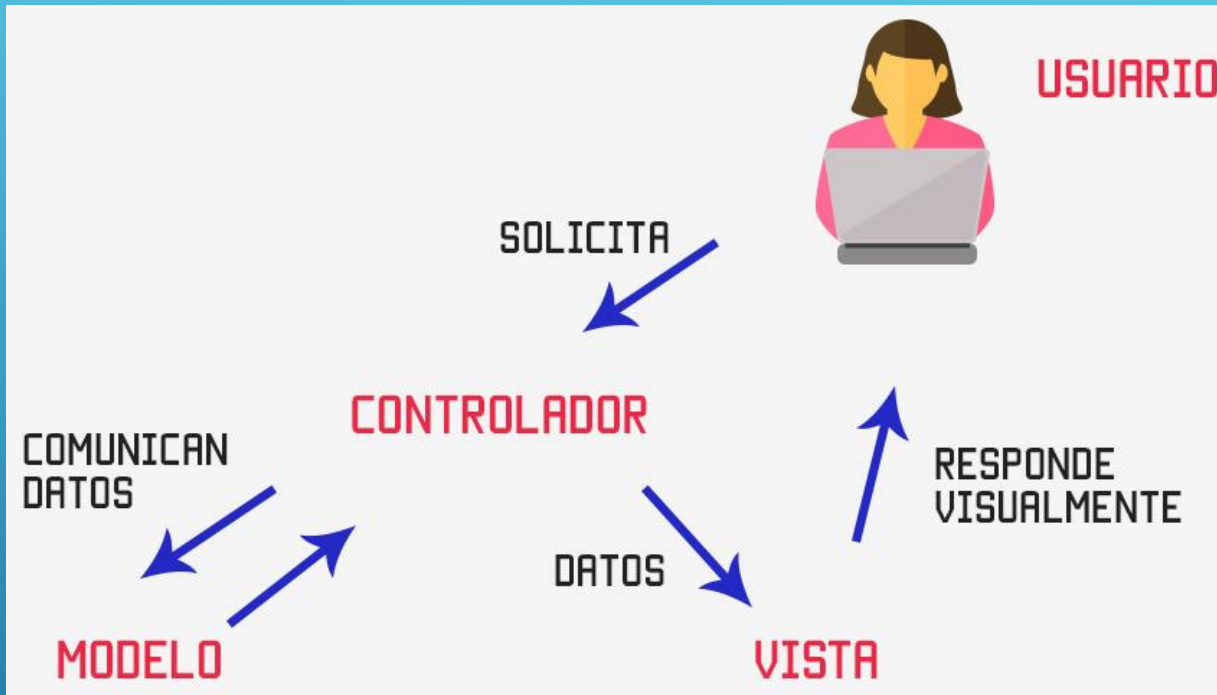
```
public class VistaUsuario {  
  
    public void mostrarUsuario(Usuario usuario) {  
        System.out.println("Usuario: " + usuario.getNombre() + " | Email: " + usuario.getEmail());  
    }  
  
}
```

EJEMPLO

Menú

```
public class Menu {  
    public static void main(String[] args) {  
        // Crear objetos del Modelo  
        Producto producto = new Producto("Laptop", 2500.0);  
        Usuario usuario = new Usuario("Ana Torres", "ana@example.com");  
        Libro libro = new Libro("El Principito", "Antoine de Saint-Exupery");  
  
        // Crear la Vista  
        //Vista vista = new Vista();  
  
        //Utilizo las vistas por aparte  
        VistaProducto vistaproducto = new VistaProducto();  
        VistaUsuario vistausuario = new VistaUsuario();  
        VistaLibro vistalibro = new VistaLibro();  
  
        // Crear el Controlador  
        //Controlador controlador = new Controlador(vista);  
  
        //Utilizo cada modelo por separado  
        CtrlProducto ctrlproducto = new CtrlProducto(vistaproducto);  
        CtrlUsuario ctrlusuario = new CtrlUsuario(vistausuario);  
        CtrlLibro ctrllibro = new CtrlLibro(vistalibro);  
  
        // Mostrar información usando el Controlador  
        //controlador.mostrarProducto(producto);  
        //controlador.mostrarUsuario(usuario);  
        //controlador.mostrarLibro(libro);  
  
        ctrlproducto.mostrarProducto(producto);  
        ctrlusuario.mostrarUsuario(usuario);  
        ctrllibro.mostrarLibro(libro);  
    }  
}
```

DIAGRAMA MVC



PREGUNTAS

