



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ El patrón de diseño MVC

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Concepto de patrón de diseño
- ✓ Patrón de diseño MVC

LEARNING PATHWAY

¿Sobre qué temas trabajaremos?

5.5

Start! 🏁

El patrón de diseño MVC

Modelo Vista-Controlador

Controladores

Elementos de Modelo
Elementos de Vista
Elementos de Controlador

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Comprender los elementos del MVC



**Aprender a implementar el
modelo-vista-controlador en un proyecto java**



➤ Elementos del Modelo



Elementos del Modelo



En la capa **Modelo** encontraremos siempre una representación de los datos del dominio, es decir, aquellas **entidades** que nos servirán para almacenar información del sistema que estamos desarrollando. Por ejemplo, si estamos desarrollando una aplicación de facturación, en el modelo existirán las clases Factura, Cliente o Proveedor, entre otras.



Si nuestra aplicación forma parte de un sistema distribuido, es decir, consume servicios prestados por otros sistemas, en el **Modelo** encontraremos las clases de transferencia de datos (**DTO**, Data Transfer Objects) que nos permitirán intercambiar información con ellos.





Elementos del Modelo

También encontraremos la **lógica de negocio** de la aplicación, es decir, la implementación de las reglas, acciones y restricciones que nos permiten gestionar las entidades del dominio. Será por tanto el responsable de que el sistema se encuentre siempre en un estado consistente e íntegro.



Por último, el **Modelo será también el encargado de gestionar el almacenamiento y recuperación de datos y entidades del dominio**, es decir, incluirá mecanismos de persistencia o será capaz de interactuar con ellos. Dado que habitualmente la persistencia se delega a un motor de bases de datos, es muy frecuente encontrar en el Modelo la implementación de componentes tipo **DAL** (Data Access Layer, o Capa de Acceso a Datos) y ORMs.





Elementos del Modelo

En una aplicación web dinámica desarrollada con Java, los elementos típicos que pertenecen a la capa de Modelo en el patrón MVC son:

- Clases POJO (Plain Old Java Objects) que representen entidades y objetos de negocio. Por ejemplo, clases Usuario, Producto, Factura, etc.
- Objetos de acceso a datos, como DAOs (Data Access Objects), para abstraer la lógica de persistencia.
- Clases que implementen la lógica de negocio más allá de los datos, como cálculos, algoritmos, validaciones, etc.
- Objetos de dominio que modelen conceptos específicos del negocio.
- Conexiones a sistemas externos como bases de datos o APIs.
- Librerías y frameworks reusables enfocados en la capa de negocio.
- Interfaces y clases de servicios que abstraiga funcionalidades de negocio.



Elementos del Modelo

En resumen, el modelo contiene toda la lógica de negocio y datos, independiente de la visualización y la lógica de control. Solo se enfoca en encapsular las reglas y objetos de dominio.



MODEL
Representa los datos que deben
mostrarse al usuario



› Elementos de la Vista



Elementos de la Vista



Los componentes de la Vista son los responsables de generar la interfaz de nuestra aplicación, es decir, de componer las pantallas, páginas, o cualquier tipo de resultado utilizable por el usuario o cliente del sistema. De hecho, suele decirse que la Vista es una representación del estado del Modelo en un momento concreto y en el contexto de una acción determinada.



Por ejemplo, si un usuario está consultando una factura a través de una aplicación web, la Vista se encargará de representar visualmente el estado actual de la misma en forma de página visualizable en su navegador. Si en un contexto B2B el cliente de nuestro sistema es a su vez otro sistema, la vista podría ser un archivo XML con la información solicitada. En ambos casos se trataría de la misma factura, es decir, la misma entidad del Modelo, pero su representación es distinta en función de los requisitos.





Elementos de la Vista



Cuando las vistas componen la interfaz de usuario de una aplicación, deberán contener los elementos de interacción que permitan al usuario enviar información e invocar acciones en el sistema, como botones, cuadros de edición o cualquier otro tipo de elemento, convenientemente adaptados a la tecnología del cliente.



En el caso de las aplicaciones para la Web, normalmente en la Vista se encontrarán los componentes capaces de generar el lenguaje de marcado de la página que será enviada al usuario.

En la Vista encontraremos los componentes responsables de generar la interfaz con el exterior, por regla general, aunque no exclusivamente, el UI de nuestra aplicación.





Elementos de la Vista



En el patrón MVC, los **elementos típicos que pertenecen a la capa de Vista en una aplicación web dinámica con Java** son:

- Páginas JSP que definen la interfaz de usuario. Contienen HTML, CSS y código JSP para dinamismo.
- Archivos estáticos como imágenes, CSS y JavaScript utilizados para la presentación.
- Componentes visuales reusables como custom tags o macros.
- Layouts y templates para estructurar contenidos comunes.
- Librerías de etiquetas JSTL para facilitar la visualización de datos.
- Internacionalización mediante archivos de propiedades.
- Formularios para recolectar inputs del usuario.
- Clases que implementen funcionalidades de renderizado.



Elementos de la Vista

En resumen, la capa de Vista contiene todo lo relacionado a la interfaz de usuario y la presentación visual de la información, separado de la lógica de negocios.



VIEW

Renderiza el modelo en la vista,
aceptar la interacción del usuario



➤ Elementos del Controlador



Elementos del Controlador



La misión principal de los componentes incluidos en el Controlador es actuar como intermediarios entre el usuario y el sistema. Serán capaces de capturar las acciones de éste sobre la Vista, como puede ser la pulsación de un botón o la selección de una opción de menú, interpretarlas y actuar en función de ellas. Por ejemplo, retornando al usuario una nueva vista que represente el estado actual del sistema, o invocando a acciones definidas en el Modelo para consultar o actualizar información.





Elementos del Controlador



Realizarán también tareas de transformación de datos para hacer que los componentes de la Vista y el Modelo se entiendan. Así, traducirán la información enviada desde la interfaz, por ejemplo los valores de campos de un formulario recibidos mediante el protocolo HTTP, a objetos que puedan ser comprendidos por el Modelo, como pueden las clases o las entidades del dominio.



Y de la misma forma, el Controlador tomará la información procedente del Modelo y la adaptará a formatos o estructuras de datos que la Vista sea capaz de manejar.





Elementos del Controlador



Por todo ello, podríamos considerar **el Controlador como un coordinador general del sistema**, que regula la navegación y el flujo de información con el usuario, ejerciendo también como intermediario entre la capa de Vista y el Modelo.



En el Controlador se encuentran los componentes capaces de procesar las interacciones del usuario, consultar o actualizar el Modelo, y seleccionar las Vistas apropiadas en cada momento.





Elementos del Controlador

Los principales elementos que pertenecen a la capa de Controlador en aplicaciones web MVC con Java son:



- Servlets que implementan la lógica de control y manejo de requests y responses.
- Filtros (filters) para interceptar y modificar requests/responses.
- Clases controladoras Front Controller cuando se utiliza ese patrón.
- Acciones específicas para manejar lógica de control de ciertas peticiones.
- APIs de mapeo de URLs a métodos como en los frameworks MVC.
- Objetos que implementan workflow y navegación de la aplicación.
- Control de sesión y autenticación de usuarios.
- Transiciones y redireccionamiento según estado de la aplicación.
- Excepciones relacionadas al control de flujo.





Elementos del Controlador

En resumen, el controlador gestiona el flujo de la aplicación, delegando trabajo al modelo y las vistas según corresponda.



CONTROLLER

Interactúa con el modelo y la vista



Elementos del patrón MVC

MVC

Model – View – Controller



MODEL

Representa los datos que deben mostrarse al usuario

VIEW

Renderiza el modelo en la vista, aceptar la interacción del usuario

CONTROLLER

Interactúa con el modelo y la vista



LIVE CODING

Ejemplo en vivo

¡Creando un controlador!

Vamos a crear un controlador que retorne una pagina index.html con un saludo por pantalla.

  **Tiempo: 15 minutos**



Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.





Ejercicio N° 1

Controladores



Controladores

Manos a la obra: 🙌

¡Finalmente vamos a unificar lo aprendido! Vas a trabajar sobre un nuevo proyecto donde organizarás las clases nuevamente para respetar el patrón DAO.

Consigna: ✍️

- 1- Crear las clases necesarias para interactuar con UserDAO, AccountDAO y CurrencyDAO .
- 2- Crear una nueva BD Wallet e insertar registros.
- 3- Modificar el nombre de al menos dos usuarios.

Tiempo 🕒: 30 minutos



○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la implementación del patrón MVC**
- ✓ **Aprender el concepto de modelo-vista-controlador.**



#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 🙌 🙌 🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. *Lectura Modulo 5, Lección 5: páginas 3 - 5*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 🙌



Momento: ✚

Time-out!

🕒 5 min.

