

# Introducción al desarrollo web utilizando Java y Spring Boot



## Objetivos de la sesión.

Explicar las herramientas de gestión de construcción.

Explicar la estructura del proyecto Spring Boot

Explicar la gestión de dependencias

## Mentalidades:

- ORIENTACIÓN A FUTURO
- RESPONSABILIDAD PERSONAL
- MENTALIDAD DE CRECIMIENTO
- PERSISTENCIA

## Habilidades:

- TRABAJO EN EQUIPO
- ORIENTACIÓN AL DETALLE
- PROACTIVIDAD
- COMUNICACIÓN

## 1. ¿Qué es un framework?



Un framework (o marco de trabajo) es una estructura de software que sirve como base para organizar y desarrollar una aplicación. En lugar de empezar desde cero, un framework te proporciona un esqueleto con código, herramientas y convenciones predefinidas para resolver problemas comunes

# Construir la tienda de Oxxo SIN un framework (Desde Cero)

Si decides hacerlo por tu cuenta, tendrías que:

- Diseñar el plano: ¿Dónde pones los refrigeradores? ¿La caja? ¿La máquina de café? ¿Los hot dogs? Tienes que planear cada centímetro desde cero.
- Buscar proveedores: Negociar uno por uno con Coca-Cola, Sabritas, Bimbo, los del hielo, los de las recargas telefónicas, etc.
- Crear un sistema de cobro: Programar o comprar un software para la caja registradora, el inventario y el pago de servicios.
- Definir la operación: Crear manuales para los empleados sobre cómo hacer un café, cómo rellenar los estantes, cómo hacer el corte de caja.
- Crear una marca: Diseñar un logo, elegir los colores de la tienda, pensar en un nombre.

Es un trabajo enorme, lento y muy propenso a errores. Estás "reinventando la rueda" de cómo funciona una tienda de conveniencia.



# Construir la tienda CON un framework (El Modelo de Franquicia OXXO)

Ahora, imagina que en lugar de empezar de cero, adquieres una franquicia OXXO. La franquicia OXXO es tu framework.

- ▶ La Estructura (El Layout): Te dan los planos exactos de la tienda. Sabes que los refrigeradores van al fondo, el café cerca de la entrada y la caja en una posición específica. Esta es la arquitectura predefinida del framework.
- ▶ Las Librerías (Proveedores y Productos): OXXO ya tiene los contratos con todos los proveedores. Te dicen qué productos vender y te conectan directamente con la red de distribución. No tienes que buscar a nadie. Estas son las librerías y módulos listos para usar.
- ▶ Las Herramientas (El Sistema y la Marca): Te dan su sistema de punto de venta ya programado (Spin by Oxxo, pago de servicios), el uniforme de los empleados y toda la marca (el logo, los colores rojo y amarillo). Estas son las herramientas y utilidades que incluye el framework.
- ▶ La Convención sobre Configuración (Las Reglas de OXXO): No puedes pintar la tienda de azul. Debes seguir sus reglas de operación, precios y promociones. Al seguir estas convenciones, todo es más rápido y coherente.



# El principio de Convención sobre Configuración



Convención sobre Configuración, también conocido como CoC, es un paradigma de programación de software que busca minimizar el número de decisiones que un desarrollador necesita hacer, ganando así en simplicidad pero no perdiendo flexibilidad por ello.

Por ejemplo, si hay una clase «Product» en el modelo, por convención la tabla correspondiente en la base de datos se llamará «products». Solo si alguien se aparta de esta convención, como llamar a la tabla «sold\_products», entonces deberá escribir código respecto a ese nombre.



# Resumen de Convención sobre Configuración

Concepto	Analogía con OXXO	En el Código
<b>Convención</b>	Las reglas y estándares que todos los OXXO siguen por defecto (colores, productos base, layout).	El framework asume dónde están los archivos y cómo se llaman, y proporciona funcionalidades base sin que tengas que pedir las.
<b>Configuración</b>	Cambiar el horario a 24/7, vender quesos locales o crear una promo para estudiantes.	Crear un archivo de configuración para cambiar un valor por defecto, instalar una librería extra o escribir una nueva clase con tu propia lógica.

— Los planos dicen que aquí va un estacionamiento.  
— Pero señor, los edificios no permiten...  
— UN ES-TA-CIO-NA-MIEN-TOOO, DIJE.





# Spring Framework



Spring es un framework de desarrollo de aplicaciones para Java. Su objetivo principal es simplificar el desarrollo de aplicaciones empresariales complejas (Java EE), promoviendo un código más limpio, desacoplado y fácil de probar.

Spring tiene como objetivo facilitar los desafíos de los desarrolladores, como la creación de aplicaciones web, el trabajo con bases de datos, la protección de aplicaciones y los microservicios.

# Spring Boot



Spring Boot no es un reemplazo de Spring, sino una extensión del framework Spring que facilita y acelera drásticamente la creación de aplicaciones

Spring Boot se basa en Spring para simplificar aún más su uso con una gestión de dependencias simplificada, configuración automática e información sobre el tiempo de ejecución.

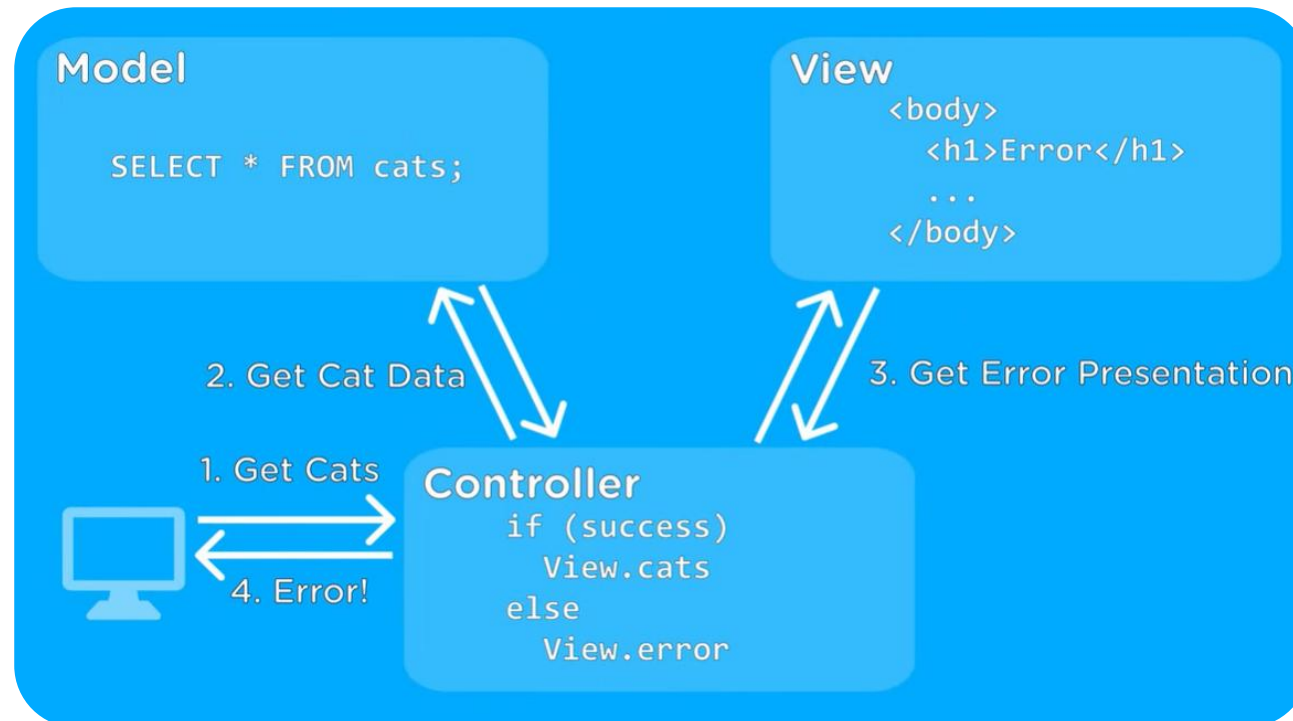
# Resumen de Spring y Spring Boot

Característica	Spring Framework	Spring Boot
Enfoque	Flexibilidad y control total.	Rapidez y convención sobre configuración.
Configuración	Manual y explícita (XML o JavaConfig).	Automática e implícita.
Servidor	Requiere un servidor externo (Tomcat, etc.).	Trae un servidor embebido.
Dependencias	Se gestionan manualmente una por una.	Se gestionan con "starters" que agrupan todo.
Recomendación	<b>Aprende sus conceptos</b> , porque es la base de todo.	<b>Úsalo para desarrollar</b> , porque es el estándar moderno y es mucho más productivo.

# El patrón MVC (Modelo-Vista-Controlador)



Es un estilo de arquitectura de software que separa una aplicación en tres componentes principales interconectados. El objetivo es organizar el código separando la lógica de negocio, los datos y la interfaz de usuario. Esta separación hace que la aplicación sea mucho más fácil de mantener, escalar y probar.



# La vista



La Vista es todo lo que el usuario (el cliente) puede ver e interactuar. Es la interfaz de usuario (UI).

- En OXXO: Es la tienda misma: los pasillos, los refrigeradores, el mostrador, las etiquetas de precios. Es la presentación de los productos.
- En la Web: Es el archivo HTML renderizado en el navegador, con sus botones, formularios y CSS.

Su único trabajo es mostrar la información que le dan y capturar las acciones del usuario (como un clic o el envío de un formulario). No procesa datos.

# El controlador



El Controlador es el intermediario. Actúa como el cerebro que recibe las peticiones del usuario y decide qué hacer.

- En OXXO: Es el cajero. Cuando el cliente llega al mostrador con la Coca-Cola (una petición del usuario desde la Vista), el cajero toma el producto. Él no sabe el precio de memoria ni gestiona el inventario, pero sabe a quién preguntar.
- En la Web: Es una clase o función que recibe una petición HTTP (ej: GET /productos/123). Escucha al usuario, entiende lo que quiere y coordina a los otros componentes.

El Controlador recibe la interacción del usuario, se comunica con el Modelo para obtener datos y luego le dice a la Vista qué debe mostrar.

# El modelo



El Modelo representa los datos y la lógica de negocio de la aplicación. Es el único que sabe cómo acceder y manipular la información.

- En OXXO: Es el sistema informático de la tienda. Este sistema contiene la base de datos de productos, sus precios, el stock actual y las reglas de negocio (ej: si el producto es una Coca-Cola, su precio es \$17, al vender una, descuenta 1 del inventario).
- En la Web: Son las clases que interactúan directamente con la base de datos. Se encargan de guardar, leer, actualizar y borrar datos, así como de ejecutar la lógica de negocio (validaciones, cálculos, etc.).

El Modelo no sabe nada de la interfaz de usuario; su única preocupación son los datos.



