**Apuntes Spring con JAVA**

Modelo vista controlador (MVC):

Controlador es quien se comunica con el **modelo** y la **vista.**

El modelo lleva las validaciones y las conexiones a la base de datos

La vista no es nada mas que la interfaz de usuario.

Pasos para crear un proyecto Spring:

1. Seleccionar files y desplegar el menú new.
2. Seleccionar **Spring starter Project**
3. Dar nombre al proyecto.
4. Por ahora seleccionar **Maven** en las opciones de gradle.
5. Cambiar los campos que digan com.
6. Si es primera vez, seleccionar lo siguiente:
   1. Spring boot Devtools
   2. Spring web
   3. Thymeleaf
7. Dar clic en finish.

Es importante que en el apartado de packaging elijamos un servicio de pakage que nos convenga. Los pakage **JAR** contienen un servidor embebido que nos permite hacer la publicación de la palicación. Los pakage **WAR** se utilizan para deploy en servidores externos.

Paso para crear un controlador:

1. Dentro de la carpeta **src/main/java** se crea un nuevo package donde alojaremos todos los controladores que se necesiten.
2. Dentro de ese package se crea una nueva clase en donde irá el código del controlador.
3. Fuera de la función principal se coloca el decorador **@Controller** para indicarle al sistema que esa función se identifica como un controlador. Dentro de este se definirán métodos de tipo publico.
4. Afuera del método que creemos se coloca el decorador **@GetMapping** el cual le indica al navegador, la ruta a la que debe ingresar.

Creación de vista.

1. Dentro de la carpeta **src/main/resources** se encuentra la carpeta **templates**.
2. Dar clic derecho y desplegar el menú **new**.
3. Crear un nuevo **file**.
4. Este nuevo archivo será un archivo de código HTML.

El decorador **@RequestMapping** indica que la función a la que hace referencia va a manejar las peticiones HTTP como GET POST etc…

Este tipo de controlador se debe de incluir dentro de las clases deniminadas controladores con el decorador **@Controller.**

En este método podemos manejar los métodos de comunicación HTTP más comunes. Este decorador contiene métodos ya definidos, se declaran usando la palabra reservada **method** y se iguala al método de comunicación que se va a recibir. Si no se escribe un método de comunicación, la función soportará todos los métodos de comunicación.

Una buena practica para Spring es utilizar los decoradores directos como son el **@GetMapping** que ya son más específicos para cada método de comunicación. **@RequestMapping** se podría considerar como un método global para soportar todos los métodos de comunicación.

Dentro de estos decoradores de mapeo, se encuentra la variable **value** o **path**  el cual recibe un string en donde podemos definir una **URL** la cual, cuando escribamos dentro del navegador, nos devolverá el valor, la vista o cualquier cosa que se defina dentro del método. Esta URL puede referenciarse de muchas formas, y para indicarle esas formas al método, podemos utilizar las llaves (**{}**) y dentro escribiremos las diferentes formas o rutas que, al escribir en el navegador, nos harán referencia a ese método: value = ({“/index” , ”/” , “/home”})

Se pueden listar varios métodos dentro de un mismo controlador, para manejar diferentes tipos de acciones. Es importante que todos los controladores que se creen se guarden dentro de un package especifico para controladores, dentro del package principal.