



Trabajo Práctico de presentación obligatoria.

Objetivo de aprendizaje:

- Creación de un proyecto spring boot.
- Uso de trazas con Log.
- Uso de @Component, @Controller, @PostMapping, @GetMapping, @Autowired, @Qualifier, @Service
- Html – formularios
- List – ArrayList, String, Manejo de Fechas y formatos

1.- Utilizando el IDE **Spring Tool Suite**, cree un proyecto de tipo **Spring Starter Project**. Coloque como nombre de proyecto **Apellido_Nombre_LU_TP3**. (Donde **Apellido** es su apellido, **LU** es su número de libreta universitaria. Ej: **Diaz_Maria_1245_TP3**.

El nombre del paquete debe cumplir con lo siguiente:

ar.edu.unju.fi.tp3

2.- A continuación cree una clase dentro de su paquete model. La clase debe estar anotada con el estereotipo **@Component**, esto marcará a la clase como un bean de Spring.

Nombre de clase	atributos
Cliente	tipoDocumento(String) Puede tomar alguno de estos valores: DNI, Pasaporte nroDocumento (int) nombreApellido (String) email (String) password(String) fechaNacimiento(LocalDate) edad(int) se debe calcular en base a la fecha actual y la fecha de nacimiento codigoAreaTelefono (int) nroTeléfono(int) fechaUltimaCompra (LocalDate) representa la fecha de la última compra que realizó el cliente, se ingresa desde el formulario

3.- Dentro de la carpeta templates, cree los siguientes archivos html

nuevocliente.html: en este archivo confeccione un formulario para poder dar de alta un nuevo cliente.

Algunos elementos de html que debe utilizar:

- form
- select
- Input type="text"
- Input type="number"
- Input type="password"
- Input type="email"
- Input type="date"
- Input type="submit"

clientes.html: en este archivo utilice el elemento table de html para poder visualizar algunos datos de los clientes.

Los datos a visualizar son:

- Tipo de documento
- Número de documento
- Apellido y Nombres (la cadena de texto en Mayúsculas)
- E-mail
- Fecha de nacimiento (En formato dd/MM/yyyy)
- Fecha de UltimaCompra (En formato dd/MM/yyyy)
- Edad
- Teléfono Es la concatenación de código de área + nro de teléfono. Ejemplo: "388-4789150". En el formulario se ingresan por separado y en la tabla se muestra como en el ejemplo

En la misma página clientes.html luego de mostrar los datos solicitados correctamente agregue una nueva columna “Datos adicionales” al <table> y dentro de ella muestre un texto con la siguiente información:

- *T desde Ultima Compra:* Tiempo transcurrido desde la fecha de la última compra a la fecha de hoy expresado en Año-Mes-Día.
- *T desde Fecha nacimiento:* Tiempo transcurrido expresado en días desde la fecha de nacimiento hasta la fecha actual.
- *T hasta cumpleaños:* Tiempo que falta para el próximo cumpleaños del cliente. Se quiere saber el tiempo que expresado en Día-Mes-Año Hora-Min-Seg

4.- Agregue un nuevo paquete a su proyecto denominado ar.edu.unju.fi.tp3.util, en este paquete cree una clase que contenga la declaración de una lista con acceso público, en dicha lista serán almacenados los clientes, y a partir de ella se realizarán todas las demás operaciones.

Luego de completar el formulario con el nuevo cliente y enviar los datos al controlador correspondiente, se debe mostrar la página con la tabla que lista todos los clientes.

5.- En el paquete de controladores agregue el controlador específico para poder gestionar las peticiones de alta de un nuevo cliente y el listado de todos los clientes.

Nombre del controlador:

ClienteController

Agregue los métodos necesarios para responder a las siguientes peticiones:

/cliente/nuevo: de tipo **Get**, muestra la página con el formulario para dar de alta un nuevo cliente (nuevocliente.html)

cliente/guardar: de tipo **Post**, guarda el nuevo cliente en la lista de clientes que maneja la capa Service. Luego muestra la página clientes.html

cliente/listado: de tipo **Get**, muestra la página con el listado de todos los clientes en una tabla.

5.- En los paquetes **ar.edu.unju.fi.tp3.service** y **ar.edu.unju.fi.tp3.service.imp** declare la lógica necesaria para poder guardar objetos de tipo Cliente en una lista (ArrayList) pública del paquete **ar.edu.unju.fi.tp3.util** (ArrayList). En el paquete service la interface **IClienteService** (solo la declaración de los métodos necesarios para guardar un cliente en la lista y otro método para obtener todos los clientes. En el paquete imp la clase que implementa la interface: **ClienteServiceImp** (la implementación de los métodos de la interface).

6.- Agregue trazas de Log para poder identificar las acciones que se realizan en la capa Service. Ej: si se guardó un nuevo cliente, podríamos mostrar en la traza el estado del objeto que se guardó.