

## Desafío - Crear proyecto Contact Manager

- Para realizar este desafío debes haber estudiado previamente todo el material disponible correspondiente a la unidad.
- Una vez terminado el desafío, comprime la carpeta que contiene el desarrollo de los requerimientos solicitados y sube el .zip en el LMS.
- Puntaje total: 10 puntos.
- Desarrollo desafío:
  - El desafío se debe desarrollar de manera Individual

### Capítulos

- Patrón de diseño MVC
- Spring
- Creación de un proyecto con Spring MVC
- Controlador y la notación @Controller
- Dispatcher
- Creando la vista
- Creando el controlador

## Descripción

El desafío consiste en crear y comenzar el desarrollo de un proyecto llamado ContactManager, el cual debe permitir mediante una pantalla de usuario, el agregar y eliminar un contacto personal.

Los atributos de un contacto en particular, serán los siguientes:

Mantenedor de Contactos

Nombre:	<input type="text"/>	Apellido Paterno:	<input type="text"/>	Apellido Materno:	<input type="text"/>
Dirección:	<input type="text"/>			Teléfono:	<input type="text"/>
<input type="button" value="Agregar"/> <input type="button" value="Eliminar"/>					

Lista de contactos

Id	Nombre	Apellido Paterno	Apellido Materno	Dirección	Teléfono
----	--------	------------------	------------------	-----------	----------

  

Mantenedor de Contactos

Nombre:	<input type="text" value="Papo"/>	Apellido Paterno:	<input type="text" value="Paiote"/>	Apellido Materno:	<input type="text" value="Maicote"/>
Dirección:	<input type="text" value="Calle las Hormigas"/>			Teléfono:	<input type="text" value="9999125"/>
<input type="button" value="Agregar"/> <input type="button" value="Eliminar"/>					

Lista de contactos

Id	Nombre	Apellido Paterno	Apellido Materno	Dirección	Teléfono
----	--------	------------------	------------------	-----------	----------

- Nombre
- Apellido Paterno
- Apellido Materno
- Dirección
- Teléfono

El objetivo final de este proyecto, es crear la Vista y el Controlador de la aplicación. La vista debe ser como la siguiente:

Mantenedor de Contactos

Nombre:	<input type="text"/>	Apellido Paterno:	<input type="text"/>	Apellido Materno:	<input type="text"/>
Dirección:	<input type="text"/>			Teléfono:	<input type="text"/>
<input type="button" value="Agregar"/> <input type="button" value="Eliminar"/>					

Lista de contactos

Id	Nombre	Apellido Paterno	Apellido Materno	Dirección	Teléfono
----	--------	------------------	------------------	-----------	----------

## Aspectos funcionales

1. Para entrar a la aplicación mediante el navegador web, se debe realizar mediante la siguiente URL:
  - <http://localhost:8080/contactManager/>
2. Al presionar el botón Agregar, los datos ingresados en los cuadros de texto asociados a cada atributo del contacto, deben aparecer cargados en la tabla o grilla, como se muestra en la siguiente imagen:

Mantenedor de Contactos

Nombre:	<input type="text" value="Pepito"/>	Apellido Paterno:	<input type="text" value="Palote"/>	Apellido Materno:	<input type="text" value="Malote"/>
Dirección:	<input type="text" value="Calle las Hormigas"/>			Teléfono:	<input type="text" value="9999125"/>
<input type="button" value="Agregar"/>		<input type="button" value="Eliminar"/>			

Lista de contactos

Id	Nombre	Apellido Paterno	Apellido Materno	Dirección	Teléfono
----	--------	------------------	------------------	-----------	----------

## Requerimientos

1. rear proyecto con <https://start.spring.io/> maven y Spring Boot 2.1.6, considerando los siguientes datos del proyecto: **(1 Punto)**
  - groupId: cl.desafiolatam.
  - artifactId: contactmanager.
  - packaging: war.
  - Utilizar Java 8.
2. Crear la estructura del proyecto, clase controller, vista y archivo web.xml de la siguiente manera: **(2 Puntos)**
  - Crear Packages Controller - cl.desafiolatam.controller.
  - Crear clase controller - ContactController.java.
  - Crear archivo web.xml, configurando el dispatcher de Spring.
  - Crear método getContactList y devolver la palabra Hola Mundo!! (Esto es, solo para comprobar que el proyecto fue creado correctamente).
  - Crear la Vista contactManager.jsp bajo la carpeta WEB-INF/views del proyecto.
  - Configurar la vista en el archivo application.properties.
  - Probar en la url <http://localhost:8080/contactManager/>.

3. Crear la interfaz como se muestra en Requerimientos funcionales. Una vez creada la interfaz, programar el método `getContactList` para que muestre una lista de datos tipo `Contacto`. **(3 Puntos)**
  - a. El método debe ser de tipo GET.
  - b. Este método actuará como inicio de la aplicación, por lo que para acceder a él se debe hacer mediante la siguiente URL: <http://localhost:8080/contactManager/>.
  - c. La lista, debe ser una Lista de Contactos, por lo que se debe desarrollar la clase `Contacto.java` en base a los atributos del contacto.

List `<Contacto>` listaContacto.

Podemos tener un solo contacto el cual puede ser reemplazado y eliminado.

```
<div style="page-break-after: always;"></div>
```

4. Programar los siguientes métodos restantes en el controlado: **(4 Puntos)**

**addContact:** Método encargado de agregar un contacto. Este método debe ser de tipo `POST` y debe ser accedido mediante la siguiente URL: <http://localhost:8082/contactManager/addContact>.

- a. Utilizar `@ModelAttribute`.
- b. Recibir desde la Vista, el objeto `contact`. Utilizar `<form:form />` de Spring MVC.
- c. El método, debe crear un `id` para cada ítem de la lista.

**deleteContact:** Método encargado de eliminar un contacto. Este método debe ser de tipo `POST` y debe ser accedido mediante la siguiente URL: <http://localhost:8082/contactManager/deleteContact?id=>.

## Recursos

Como recomendación, se podría utilizar el siguiente código javascript para la vista:

```
<script type="text/javascript">
var idSelect = 0;
function submitFormAction(action, method){
document.getElementById("frmAcciones").action = action;
document.getElementById("frmAcciones").method = method;
document.getElementById("frmAcciones").submit();
}
function setIdSelect(id){
idSelect = id;
}
function deleteRow(){
window.location.href = "/contactManager/deleteContact/?id=" +
idSelect;
}
</script>
```