

## Desafío - Contact Manager (Parte II)

- 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.
- Desarrollo desafío:
  - El desafío se debe desarrollar de manera Individual
  - Para la realización del desafío necesitarás apoyarte del archivo Apoyo Desafío - Contact Manager (parte II)

### Descripción

En este desafío, se debe finalizar el proyecto Contact Manager, en la carpeta .zip llamada Apoyo Desafío -Contact Manager (Parte II), podrás descargar el proyecto Contact Manager que es la base para este desafío.

#### En este desafío se requiere realizar lo siguiente:

- Crear la capa de Servicio (Interfaz e Implementación)
- Implementar la lógica de la implementación del servicio (Lógica que tenemos en el controlador)
- Implementar la función de eliminar en la capa de servicio
- Implementar Test JUnit
- Empaquetar la aplicación

## Aspectos funcionales

1. Al seleccionar un registro o fila de la grilla, este debe poder ser eliminado mediante el botón Eliminar.

Mantenedor de Contactos

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

Lista de contactos

	Id	Nombre	Apellido Paterno	Apellido Materno	Dirección	Teléfono
<input checked="" type="radio"/>	1	Pepito	Palote	Malote	Calle las Hormigas	9999125
<input type="radio"/>	2	Oliver	Aton		China	99999

Mantenedor de Contactos

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

Lista de contactos

	Id	Nombre	Apellido Paterno	Apellido Materno	Dirección	Teléfono
<input type="radio"/>	2	Oliver	Aton		China	99999

## Requerimientos

1. En el proyecto Contact Manager, crear la capa de servicio: **(1 Punto)**
  - Crear packages `cl.desafiolatam.service`
  - Crear packages `cl.desafiolatam.service.impl`
  - Crear la Interfaz `ContactService.java`
  - Crear la clase implementación `ContactServiceImpl.java`
  - Crear los siguientes métodos:
    - `getContactList`
    - `addContact`
2. Crear la lógica para cada uno de los métodos (1) de la clase service: **(4 Puntos)**
  - La clase Contact, se debe inyectar a la clase Service
  - Declarar de forma privada, la lista de contactos
  - Crear el constructor de la clase Service, en donde se inicialice o se cree `listContact`

- Método `getContactList`: Debe agregar a `listContact` el objeto `contact`, si y sólo si este último tiene un `id` mayor a cero.
- Método `addContact`: Debe crear un `id` al contacto con `listContact.size() + 1`. Luego, debe agregar a la lista de contacto, el objeto `contact`.
- Método `deleteContact`: Debe ser capaz de eliminar un contacto o ítem de la lista de contacto.

```
private List <Contact> listContact;
```

3. Crear el método `deleteContact()`, en la capa de Servicio para borrar un registro de la lista. Luego Implementar el resto de la lógica, según corresponda, en el controlador y en la vista: **(4 Puntos)**

- `deleteContact`: Método encargado de eliminar un contacto. Este método debe ser de tipo GET y debe recibir por parámetro el `id` correspondiente al ítem a eliminar de la lista. Para acceder a este método, la url debe ser como sigue: <http://localhost:8082/contactManager/deleteContact/?id=2>, siendo `id` el parámetro que recibirá el método del controlador.

a. Utilizar `@RequestParam`

4. Empaquetar la aplicación **(1 Punto)**

## 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>
```

