

Eliminar Cliente PrimeFaces y Spring Boot – parte 2



GUIA: Guía: Implementación del Caso de Eliminación de Cliente en ZonaFitWeb

📌 Introducción

En esta guía, implementaremos la funcionalidad para **eliminar un cliente** en la aplicación **ZonaFitWeb**. Agregaremos el método necesario en el **controlador**, asegurando que cuando el usuario presione el botón de "Eliminar", se remueva correctamente el cliente tanto de la base de datos como de la lista en memoria.

🔍 Puntos clave que trabajaremos

- Implementaremos el método `eliminarCliente()` en el **controlador**.
- Se actualizará la tabla para reflejar la eliminación sin necesidad de recargar toda la página.
- Se gestionará el estado del objeto `clienteSeleccionado`.

🛠️ Desarrollo de la Funcionalidad

📁 Archivo: `src/main/java/gm/zona_fit/controlador/IndexControlador.java`

Este es el método que se encargará de eliminar un cliente cuando el usuario lo confirme en la interfaz.

```
public void eliminarCliente(){
    logger.info("Cliente a eliminar: " + this.clienteSeleccionado);
    this.clienteServicio.eliminarCliente(this.clienteSeleccionado);
    // Eliminar el registro de la lista de clientes
    this.clientes.remove(this.clienteSeleccionado);
    // Reset del objeto de cliente seleccionado
    this.clienteSeleccionado = null;
    FacesContext.getCurrentInstance().addMessage(null,
        new FacesMessage("Cliente Eliminado"));
    PrimeFaces.current().ajax().update("forma-clientes:mensajes",
        "forma-clientes:clientes-tabla");
}
```

📌 Explicación del Código

1. **Registro en el log:** Se imprime en la consola qué cliente será eliminado.
2. **Eliminación en la base de datos:** Se invoca el método `eliminarCliente()` del servicio, que ejecuta la eliminación en la BD.
3. **Eliminación en la lista en memoria:** Se usa `clientes.remove(clienteSeleccionado)`; para actualizar la interfaz sin recargar los datos desde la base.
4. **Reseteo del objeto `clienteSeleccionado`:** Esto previene que queden datos residuales después de eliminar un cliente.
5. **Mensaje de confirmación:** Se usa `FacesContext` para mostrar un mensaje de éxito en la interfaz.
6. **Actualización parcial de la interfaz:** Se ejecuta un `PrimeFaces.ajax().update()` para actualizar solo las secciones afectadas (mensajes y la tabla de clientes), evitando una recarga completa de la página.

🛠️ Código completo

📁 Archivo: `src/main/java/gm/zona_fit/controlador/IndexControlador.java`

```
package gm.zona_fit.controlador;

import gm.zona_fit.modelo.Cliente;
import gm.zona_fit.servicio.IClienteServicio;
import jakarta.annotation.PostConstruct;
import jakarta.faces.application.FacesMessage;
import jakarta.faces.context.FacesContext;
import jakarta.faces.view.ViewScoped;
```

```
import lombok.Data;
import org.primefaces.PrimeFaces;
import org.slf4j.LoggerFactory;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;
import java.util.List;
import org.slf4j.Logger;

@Component
@Data
@ViewScoped
public class IndexControlador {

    @Autowired
    IClienteServicio clienteServicio;
    private List<Cliente> clientes;
    private Cliente clienteSeleccionado;
    private static final Logger logger = LoggerFactory.getLogger(IndexControlador.class);

    @PostConstruct
    public void init(){
        cargarDatos();
    }

    public void cargarDatos(){
        this.clientes = this.clienteServicio.listarClientes();
        this.clientes.forEach(cliente -> logger.info(cliente.toString()));
    }

    public void agregarCliente(){
        this.clienteSeleccionado = new Cliente();
    }

    public void guardarCliente(){
        logger.info("cliente a guardar: " + this.clienteSeleccionado);
        // Agregar (insert)
        if(this.clienteSeleccionado.getId() == null){
            this.clienteServicio.guardarCliente(this.clienteSeleccionado);
            this.clientes.add(this.clienteSeleccionado);
            FacesContext.getCurrentInstance().addMessage(null,
                new FacesMessage("Cliente Agregado"));
        }
        // Modificar (update)
        else{
            this.clienteServicio.guardarCliente(this.clienteSeleccionado);
        }
    }
}
```

```
        FacesContext.getCurrentInstance().addMessage(null,
            new FacesMessage("Cliente Actualizado"));
    }
    // Ocultar la ventana modal
    PrimeFaces.current().executeScript("PF('ventanaModalCliente').hide()");
    // Actualizar la tabla usando ajax
    PrimeFaces.current().ajax().update("forma-clientes:mensajes",
        "forma-clientes:clientes-tabla");
    // Reset del objeto cliente seleccionado
    this.clienteSeleccionado = null;
}

public void eliminarCliente(){
    logger.info("Cliente a eliminar: " + this.clienteSeleccionado);
    this.clienteServicio.eliminarCliente(this.clienteSeleccionado);
    // Eliminar el registro de la lista de clientes
    this.clientes.remove(this.clienteSeleccionado);
    // Reset del objeto de cliente seleccionado
    this.clienteSeleccionado = null;
    FacesContext.getCurrentInstance().addMessage(null,
        new FacesMessage("Cliente Eliminado"));
    PrimeFaces.current().ajax().update("forma-clientes:mensajes",
        "forma-clientes:clientes-tabla");
}
}
```

▶ Ejecución del Proyecto

Para probar la funcionalidad de eliminación, sigue estos pasos:

1. Inicia el proyecto en **IntelliJ IDEA**.
2. Ejecuta `zonaFitWeb.java` con **Click derecho > Run**.
3. En la aplicación web, **selecciona un cliente** y haz clic en **Eliminar**.
4. Confirma la eliminación presionando "Sí".
5. Verifica que:
 - o Se muestra el mensaje "Cliente eliminado".
 - o El cliente desaparece de la tabla sin recargar la página.
 - o Si consultas la base de datos, el registro ya no existe.

Zona Fit (GYM)

ID	Nombre	Apellido	Membresía	
1	Gabriel	Flores	100	<button>Editar</button> <button>Eliminar</button>
2	Paola	Juarez	200	<button>Editar</button> <button>Eliminar</button>
6	Carlos	Gomez	300	<button>Editar</button> <button>Eliminar</button>

Zona Fit (GYM)

Confirmar

Eliminar el cliente?

✓ Si ✗ No

ID	Nombre	Apellido	Membresía	
1	Gabriel		00	<button>Editar</button> <button>Eliminar</button>
2	Paola		00	<button>Editar</button> <button>Eliminar</button>
6	Carlos	Gomez	300	<button>Editar</button> <button>Eliminar</button>

Id	Nombre	Apellido	Membresía	
1	Gabriel	Flores	100	<button>Editar</button> <button>Eliminar</button>
2	Paola	Juarez	200	<button>Editar</button> <button>Eliminar</button>

✓ Conclusión

Con esta implementación, logramos:

- **Eliminar clientes** de manera eficiente tanto en la BD como en memoria.
- **Actualizar la tabla dinámicamente** sin necesidad de recargar la página.
- **Evitar datos residuales** con el reseteo del objeto `clienteSeleccionado`.

🚀 Hemos finalizado con éxito la funcionalidad de eliminación.

¡Muchas felicidades, has completado el proyecto de Zona Fit (GYM) con PrimeFaces y SpringBoot!

Nota: Recuerda que tienes el archivo .zip en la sección de recursos de este video con todos los archivos del proyecto actualizado según la lección estudiada, el cual puedes utilizar para cualquier problema.

Sigue adelante con tu aprendizaje 🚀, ¡el esfuerzo vale la pena!

¡Saludos! 🙌

Ing. Ubaldo Acosta

Fundador de GlobalMentoring.com.mx