

Guardar Cliente con PrimeFaces y Spring Boot



Guía paso a paso: Guardar Cliente con PrimeFaces y Spring Boot

❖ Introducción

En esta lección, implementaremos el método que permitirá guardar un cliente en la base de datos desde nuestra ventana modal. Este proceso incluirá la validación del cliente, el almacenamiento en la base de datos y la actualización del listado de clientes en la interfaz.

1 Implementación del método Guardar Cliente en el controlador

Vamos a agregar el método `guardarCliente()` en nuestro controlador `IndexControlador.java`.

📁 Archivo: `src/main/java/com/zonaFit/controlador/IndexControlador.java`

```
public void guardarCliente(){
    logger.info("cliente a guardar: " + this.clienteSeleccionado);
    // Agregar
    if(this.clienteSeleccionado.getId() == null){
```

```
this.clienteServicio.guardarCliente(this.clienteSeleccionado);
this.clientes.add(this.clienteSeleccionado);
FacesContext.getCurrentInstance().addMessage(null,
    new FacesMessage("Cliente Agregado")));
}
// 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;
}
```

■ Explicación del código

1. **Registro en logs:** Se imprime la información del cliente que se va a guardar.
2. **Verificación del ID:**
 - o Si el ID es null, significa que es un nuevo cliente.
 - o Se guarda en la base de datos usando `clienteServicio.guardarCliente(clienteSeleccionado)`.
 - o Se agrega el cliente a la lista `clientes` en memoria para actualizar la tabla sin hacer una nueva consulta a la base de datos.
 - o Se muestra un mensaje de éxito "Cliente agregado".
3. **Ocultar la ventana modal:** Se utiliza `PrimeFaces.current().executeScript("PF('ventanaModalCliente').hide()")` para cerrar la ventana después de guardar.
4. **Actualizar la interfaz:** Se actualiza la tabla y los mensajes con `PrimeFaces.current().ajax().update("forma-clientes:mensajes", "forma-clientes:clientes-tabla")`.
5. **Reset del objeto clienteSeleccionado:** Se asigna un nuevo objeto para futuras inserciones.

2 Prueba del Guardado de Cliente

Para probar la funcionalidad, seguimos los siguientes pasos:

► Ejecutar el proyecto

1. Abrir el proyecto en **IntelliJ IDEA**.
2. En el explorador de archivos, ubicar `ZonaFitWeb.java`.
3. **Clic derecho** sobre el archivo y seleccionar `Run ZonaFitWeb`.
4. Acceder a la aplicación en el navegador.

🛠 Prueba en la interfaz

1. Hacer clic en "**Nuevo Cliente**" en la barra de navegación.
2. Completar los campos con la información de un nuevo cliente.
3. Presionar el botón **Guardar**.
4. Verificar que aparece el mensaje `Cliente agregado` y que la tabla se actualiza con el nuevo cliente.
5. Si se presiona **Cancelar**, la ventana debe cerrarse sin guardar los cambios.

<http://localhost:8080/index.xhtml>

The screenshot shows a dark-themed modal window titled "Detalles Cliente". It contains three input fields: "Nombre *" with the value "Carlos", "Apellido" with the value "Gomez", and "Membresía *" with the value "300". At the bottom are two buttons: a blue "Guardar" button with a checkmark icon and a grey "Cancelar" button with a cross icon.

Id	Nombre	Apellido	Membresía
1	Gabriel	Flores	100
2	Paola	Juarez	200
6	Carlos	Gomez	300

3 Código final

📁 Archivo: src/main/java/com/zonaFit/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  
        if(this.clienteSeleccionado.getId() == null){  
            this.clienteServicio.guardarCliente(this.clienteSeleccionado);  
            this.clientes.add(this.clienteSeleccionado);  
            FacesContext.getCurrentInstance().addMessage(null,  
                new FacesMessage("Cliente Agregado"));  
        }  
        // 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;  
    }  
}
```

Conclusión

En esta lección:

- ✓ Implementamos el método `guardarCliente()` en el controlador.
 - ✓ Agregamos la lógica para insertar un nuevo cliente en la base de datos.
 - ✓ Refrescamos la interfaz de manera dinámica para mostrar los nuevos clientes sin necesidad de recargar la página.
 - ✓ Probamos la funcionalidad en la aplicación web agregando un nuevo Cliente.
-

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](https://www.globalmentoring.com.mx)