

**UNIVERSIDAD NACIONAL DE ASUNCIÓN**

**FACULTAD POLITÉCNICA**

**MAESTRÍA EN INGENIERÍA EN ELECTRÓNICA CON ÉNFASIS EN TECNOLOGÍA DE  
LA INFORMACIÓN**

**Módulo: SISTEMA DE INFORMACIÓN WEB**

**Tarea 05: Realizar ABM de catálogos simples  
(Países y Departamentos)**

**Integrantes:**

- Caballero Mendoza, Oscar Aureliano
- Torres Ibarrola, Mónica Beatriz

**Docentes:**

- Dr. Julio Cesar Mello Roman
- Msc. Marcos Benítez

**19/04/2025**

# Índice

<b>1. Introducción.....</b>	<b>2</b>
<b>2. Desarrollo.....</b>	<b>3</b>
<b>2.1 Objetivo del Cambio.....</b>	<b>3</b>
<b>2.2 Tablas de la Base de Datos .....</b>	<b>3</b>
<b>2.3 Archivos Agregados y Modificados.....</b>	<b>3</b>
<b>2.3.1 Archivos agregados</b>	
2.3.1.1 Aplicaciones en Java.....	3
2.3.1.2 Interfaz de Usuario (HTML).....	4
<b>2.3.2 Archivos Modificados</b>	
<b>2.4 Funcionamiento del ABM en la Aplicación.....</b>	<b>5</b>
<b>2.5 Pruebas Funcionales Realizadas.....</b>	<b>7</b>
<b>2.6 Archivos compartidos de la Tarea.....</b>	<b>8</b>
<b>3. Conclusiones.....</b>	<b>9</b>
<b>4. Referencias.....</b>	<b>10</b>

## 1. Introducción

El presente trabajo tiene como objetivo desarrollar el módulo de Altas, Bajas y Modificaciones (ABM) de dos catálogos pertenecientes a la base de datos del sistema *control\_financiero*. Este sistema tiene como finalidad gestionar información relacionada con operaciones financieras, y para ello requiere una correcta administración de los datos maestros o catálogos que lo componen.

Dentro de este contexto, se seleccionaron dos catálogos: **países y departamentos** para implementar su respectivo ABM. Estas funcionalidades permitirán la gestión integral de cada catálogo, facilitando el ingreso, actualización, eliminación y consulta de registros a través de una interfaz de usuario amigable y segura.

El trabajo incluye la codificación de las funcionalidades, su integración con la base de datos, el control de errores, validaciones básicas, el uso de herramientas de control de versiones, así como la elaboración de un informe técnico que documenta el desarrollo realizado.

## 2. Desarrollo

En esta actualización del sistema se incorporaron las operaciones de Alta, Baja y Modificación (ABM) para las entidades "Países" y "Departamentos". Estas funcionalidades permiten al usuario administrador gestionar la información geográfica fundamental del sistema desde la interfaz gráfica. Este desarrollo forma parte del módulo "Mantenedores", en la sección "Geográficos".

### 2.1 Objetivo del Cambio

El objetivo principal de este desarrollo fue habilitar funcionalidades de **Altas, Bajas y Modificaciones (ABM)** para las entidades **Países** y **Departamentos** dentro del módulo **Mantenedores/Geográficos** del sistema *control\_financiero*. Este cambio busca mejorar la administración de estructuras geográficas desde la interfaz web, permitiendo a los usuarios del sistema gestionar directamente estos datos.

La Base de Datos Involucradas es *control\_financiero*

### 2.2 Tablas de Base de Datos

- **Países:** Esta tabla almacena la información de los países registrados en el sistema.
- **Departamentos:** Representa subdivisiones de un país. Cada departamento pertenece a un país.

### 2.3 Archivos Agregados y Modificados

#### 2.3.1 Archivos agregados

##### 2.3.1.1 Aplicaciones en Java.

- En controllers (`control-financiero\src\main\java\com\example\demo\controllers`)

- ***PaísesController.java***: Controlador para operaciones ABM de países.
- ***DepartamentosController.java***: Controlador para operaciones ABM de departamentos.
- **En models (control-financiero\src\main\java\com\example\demo\models)**
  - ***Países.java***: Modelo de la tabla Países.
  - ***Departamentos.java***: Modelo de la tabla Departamentos.
- **En repositories (control-financiero\src\main\java\com\example\demo\repositories)**
  - ***PaísesRepository.java***: Repositorio de Países.
  - ***DepartamentosRepository.java***: Repositorio de Departamentos.
- **En services (control-financiero\src\main\java\com\example\demo\services)**
  - ***PaísesService.java***: Interfaz de Servicio para Países.
  - ***DepartamentosService.java***: Interfaz de Servicio para Departamentos.
- **En impl (control-financiero\src\main\java\com\example\demo\services\impl)**
  - ***PaísesServiceImpl.java***: Implementación de Servicio para Países.
  - ***DepartamentosServiceImpl.java***: Implementación de Servicio para Departamentos.

### 2.3.1.2 Interfaz de Usuario (HTML).

- **Países (control-financiero\src\main\resources\templates\países)**
  - ***formulario.html***: formulario para ingresar o editar países.
  - ***listar.html***: vista donde se listan todos los países registrados.
- **Departamentos (control-financiero\src\main\resources\templates\departamentos)**
  - ***formulario.html***: formulario para ingresar o editar departamentos.
  - ***listar.html***: listado general de departamentos.

### 2.3.2 Archivos modificados.

Algunos archivos existentes del sistema fueron modificados para integrar esta nueva funcionalidad:

- En templates (**control-financiero\src\main\resources\templates**)
  - **index.html**: se actualizó para incluir accesos a países y departamentos.
  - **error.html**: para manejar errores relacionados con las nuevas funciones.
- En usuarios (**control-financiero\src\main\resources\templates\usuarios**)
  - **formularios.html**: Formulario de usuarios
  - **listar.html**: Panel para listar usuarios
- En roles (**control-financiero\src\main\resources\templates\roles**)
  - **formularios.html**: Formulario de roles
  - **listar.html**: Panel para listar roles
- En sucursales (**control-financiero\src\main\resources\templates\sucursales**)
  - **formularios.html**: Formulario de sucursales
  - **listar.html**: Panel para listar sucursales

### 2.4 Funcionamiento del ABM en la Aplicación

Una vez que el sistema está desplegado y corriendo en **http://localhost:8080**, el usuario administrador puede acceder a la gestión de países y departamentos siguiendo estos pasos:

- Se accede con el usuario: **admin**.
- La contraseña: **123456789**

A continuación, presentamos las pantallas presentadas para visualizar el ABM en la aplicación:

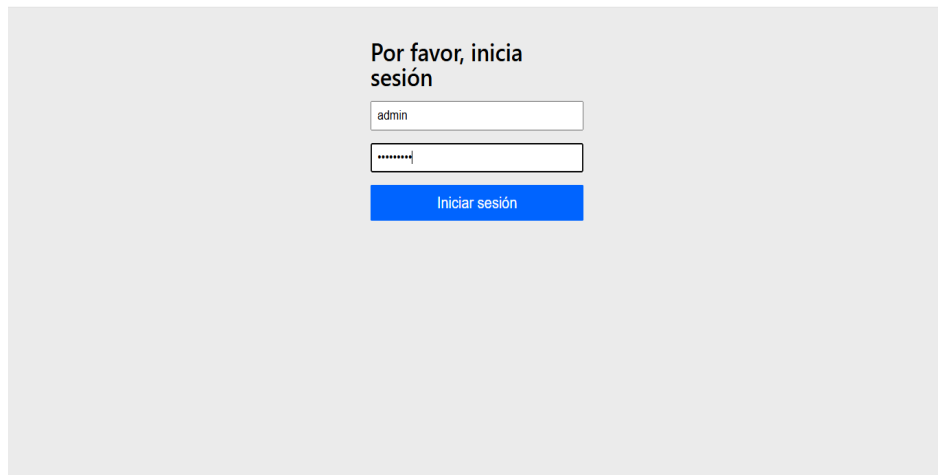


Figura 1: Pantalla de inicio

- En el menú lateral izquierdo, ir a **Mantenedores > Geográficos > Países ó Departamentos**.

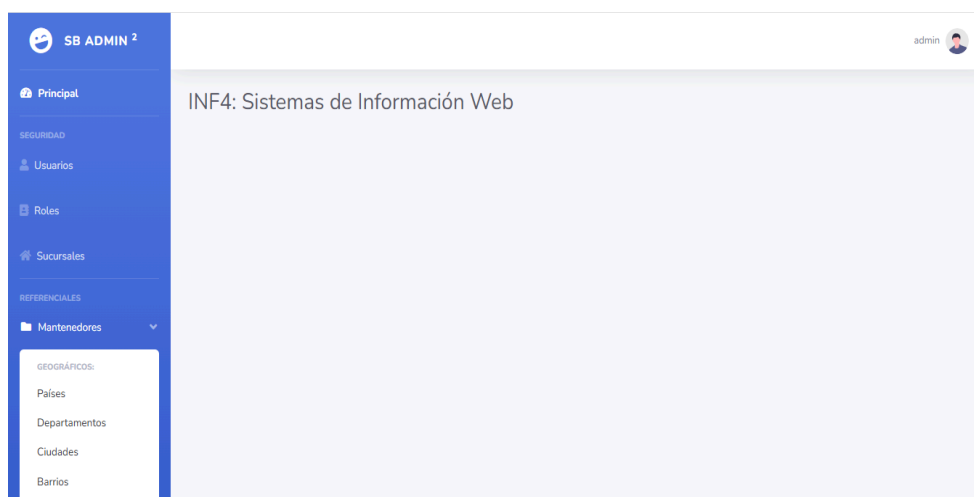


Figura 2: Menú principal

- En estas secciones se puede:
  - a. Crear un nuevo país o departamento (botón "Nuevo País/Departamentos").
  - b. Editar uno existente (botón "Editar").
  - c. Eliminar uno que ya no se necesite (botón "Eliminar").

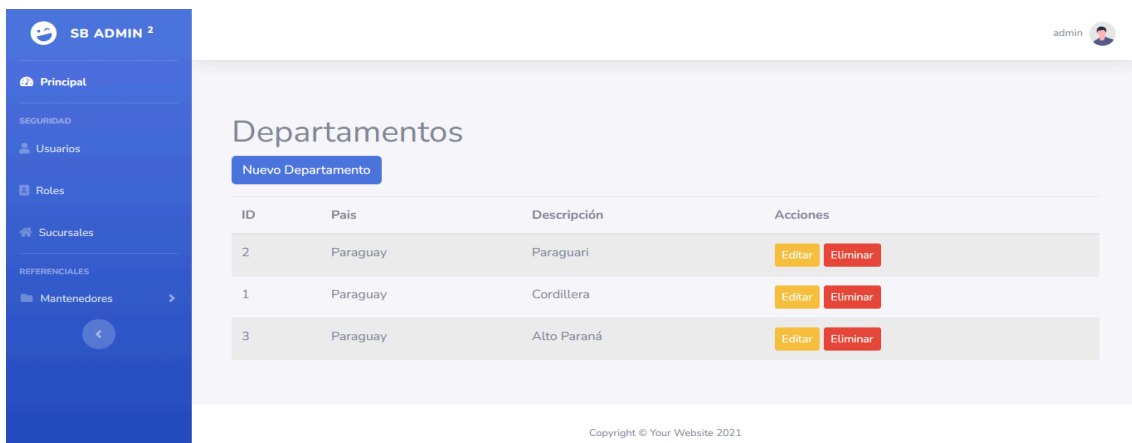


Figura 3: Pantalla de opciones de “Departamentos”

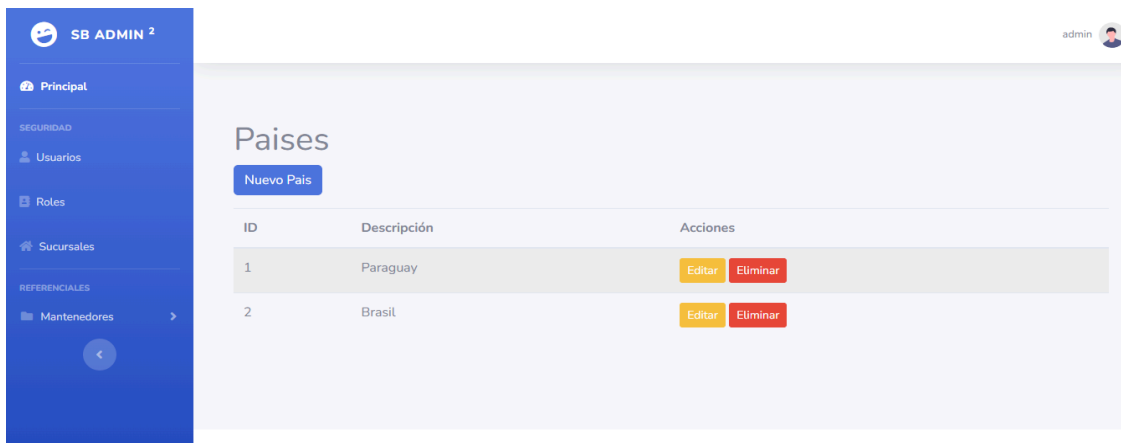


Figura 4: Pantalla de opciones de “Países”

Estas acciones afectan directamente a las tablas **Países** y **Departamentos** en la base de datos.

## 2.5 Pruebas Funcionales Realizadas

Se indica que:

- Se realizaron pruebas exitosas para crear, editar y eliminar registros de países y departamentos.
- Se accede correctamente a través del usuario administrador.



## 2.6 Archivos compartidos de la tarea

Los archivos creados y modificados están en las carpetas **java** y **resources**, que serán compartidos en forma completa para verificación de la tarea 05.

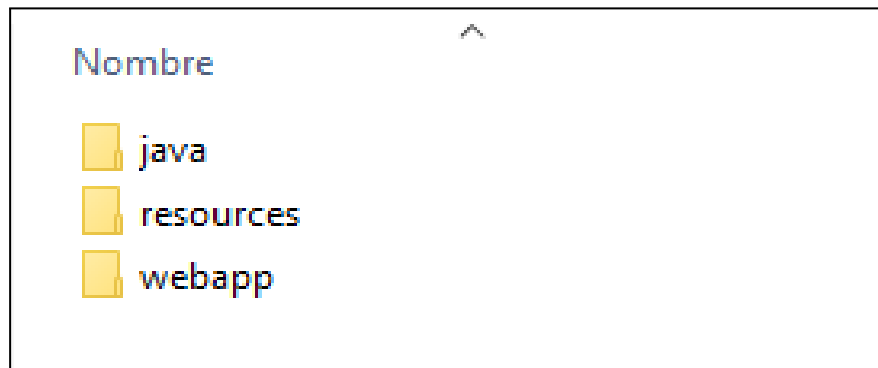


Figura 5: Carpetas modificadas

### 3. Conclusiones

La implementación del ABM para los catálogos de países y departamentos en el sistema *control\_financiero* permitió integrar funcionalidades clave para una gestión geográfica más eficiente. A través del uso de tecnologías modernas como Java, Spring Boot y PostgreSQL, se logró desarrollar una solución robusta y escalable, con una interfaz amigable para el usuario contribuyendo a mejorar la administración de datos del sistema, facilitando su mantenimiento y escalabilidad.

Además, el uso de herramientas como Git y GitHub permitió un control efectivo del código y su integración en el proyecto principal mediante la generación de un Pull Request, lo cual promueve el trabajo colaborativo y ordenado dentro del entorno de desarrollo.

Este trabajo representa un paso fundamental en la construcción modular del sistema y sienta las bases para la incorporación de nuevos catálogos o la mejora continua de los existentes, garantizando así un sistema más robusto, eficiente y adaptable a las necesidades de la organización.

## 4. Referencias

Las tecnologías y herramientas utilizadas son:

- **Java:** lenguaje de programación base del sistema.
- **Spring Boot:** framework para desarrollar aplicaciones web en Java, que facilita la creación de controladores, servicios y conexiones con la base de datos.
- **PostgreSQL:** sistema de gestión de bases de datos relacional donde se almacenan países y departamentos.
- **HTML:** para crear las interfaces visuales del sistema (formularios, listados, etc).