

Ingeniería de Software

Materia: Ingeniería de Software

Nombre del sistema: LOOK IDEAL

Apellido y Nombre: Tomas Meneces Fernandez Benjamin

Localización: Castelar

Comisión:3-A

Turno: Noche

Año de cursada: 3er año

Docente: Julián Martin Rodríguez

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

ÍNDICE:

G00 – Descripción global del producto.....	3
G01. Propósito.....	3
G02. Descripción funcional del producto y Alcance.....	4
G05. Descripción de las personas participantes en el desarrollo del sistema y los usuarios (Roles).....	5
G06. Otros Requisitos.....	5
G07: Diagrama de clases parcial de todos los módulos implementado (diagrama de clases).....	6
G08. Modelo de datos parcial de todos los módulos implementados.....	7
N01. Especificación funcional por proceso de negocio.....	10
N02. Especificaciones de Casos de Uso.....	13
T01. Arquitectura Base.....	57
T02. Gestión de Log In / Log Out del Sistema.....	59
T03. Gestión de Encriptado.....	67
T04. Gestión de Perfiles de Usuario.....	68
T05. Gestión de Múltiples Idiomas.....	70
T06a. Gestión de bitácora.....	71
T06b. Control de cambios.....	72
T07. Gestión de Dígitos Verificadores.....	74

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				Versión: 1.0
	<Nombre de Sección / Etapa>				

D00. Documentación Adicional.....	76
D03. Material de apoyo al usuario final.....	76

HISTORIAL DE REVISIONES

Fecha	Versión	Autor	Descripción
9/5/25	1.0	Tomas Meneces	Primer Entrega

G00 – Descripción global del producto

El sistema *Look Ideal* es un software de gestión de ventas mayoristas de indumentaria que permite registrar clientes, ventas, y generar reportes de desempeño comercial. Está destinado a asesores de ventas y administradores de la empresa.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

El producto permite registrar clientes, registrar ventas por cantidades de media docena, una docena, dos y tres docenas, y asociar cada venta al asesor responsable. Genera reportes mensuales que facilitan la toma de decisiones: cantidad de ventas por mes, ganancia total mensual, vendedor con mayor número de ventas y vendedor que generó mayor ganancia. Además incorpora una clasificación automática de clientes en categorías (Básico, Plata, Oro) basada en la cantidad de compras realizadas, para identificar a los clientes más frecuentes y habilitar beneficios en el precio. El sistema aporta control operativo y datos procesados que ayudan al negocio a mejorar ventas y toma de decisiones.

G01. Propósito

Elegí desarrollar este sistema porque pienso usarlo en mi vida cotidiana, el programa está pensado para mí, ya que vendo ropa por internet al por mayor y necesito un programa práctico que me ayude a registrar ventas, controlar el rendimiento de los vendedores(que estoy pensando en contratar) y reconocer a los clientes frecuentes. Me interesa optimizar y tener registros útiles para reconocer las cosas en las cuales me conviene enfocarse en mi negocio, aparte de un registro de ventas acerca de la ropa que me diga las ganancias totales del mes y otros datos.

G02. Descripción funcional del producto y Alcance

- Registro, modificación y eliminación de clientes (nombre, DNI, código postal, teléfono, dirección opcional).
- Registro de ventas mayoristas con fecha y cliente.
- Cálculo y almacenamiento del monto de cada venta.
- Clasificación automática de clientes por cantidad de compras:
 - **Básico:** < 5 compras
 - **Plata:** 5–9 compras

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

○ **Oro:** ≥ 10 compras

- Generación de reportes mensuales: número de ventas, ganancia total, vendedor con más ventas, vendedor con mayor ganancia.
- Módulo de gestión de usuarios (ABM): alta, baja, modificación de usuarios, roles y permisos.
- Otras funcionalidades: log in / log out, encriptación de contraseñas, bitácora de operaciones.

Qué no incluye (fuera de alcance actual)

- Gestión de inventario o control físico de stock (se puede incorporar en futuras versiones).
- Cálculo automático de comisiones a asesores (pendiente).
- Procesamiento de pagos y facturación electrónica.
- Integración con métodos de pago o sistemas financieros.

G05. Descripción de las personas participantes en el desarrollo del sistema y los usuarios (Roles)

El desarrollo del sistema fue realizado por Tomás Meneses, responsable del análisis, diseño e implementación del sistema de información.

Los roles definidos dentro del sistema son los siguientes:

- **Gerente:** tiene acceso total al sistema y puede realizar todas las gestiones.
- **Administrador:** puede agregar, modificar y eliminar usuarios, además de gestionar compras y datos generales.

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

- **Usuario / Asesor:** puede usar las funciones principales del sistema, como registrar o consultar información, pero sin permisos de modificación.

Nombre	Descripción	Responsabilidad
Juan García	Gerente General de la empresa Look Ideal.	Define lineamientos generales de negocio, tomando decisiones con ayuda de los reportes e información útil que da el sistema
Sofía Herrera	Encargada de Ventas y Atención al Cliente.	Establece procesos de ventas, carga de clientes y pedidos asociados.
Diego Fernández	Responsable de Sistemas (administrador)	Supervisa la implementación técnica, gestiona varias partes integrales del programa

Nombre	Rol del Sistema	Acceso dentro del sistema
--------	-----------------	---------------------------

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Versión: 1.0

Juan García Gerente

- Reportes completos (ventas, cantidades, clientes, usuarios)
- Acceso total a Bitácora
- Visualización de ABM Usuarios y Permisos
- Personalización de pedidos
- Exportación de Reportes PDF

Sofía Herrera Administrador

- ABM Usuarios
- ABM Idiomas
- ABM Permisos
- Control de cambios e historial

Matías Roldán Usuario

- Agregar, modificar y borrar Pedidos
- Agregar, modificar y borrar Cliente

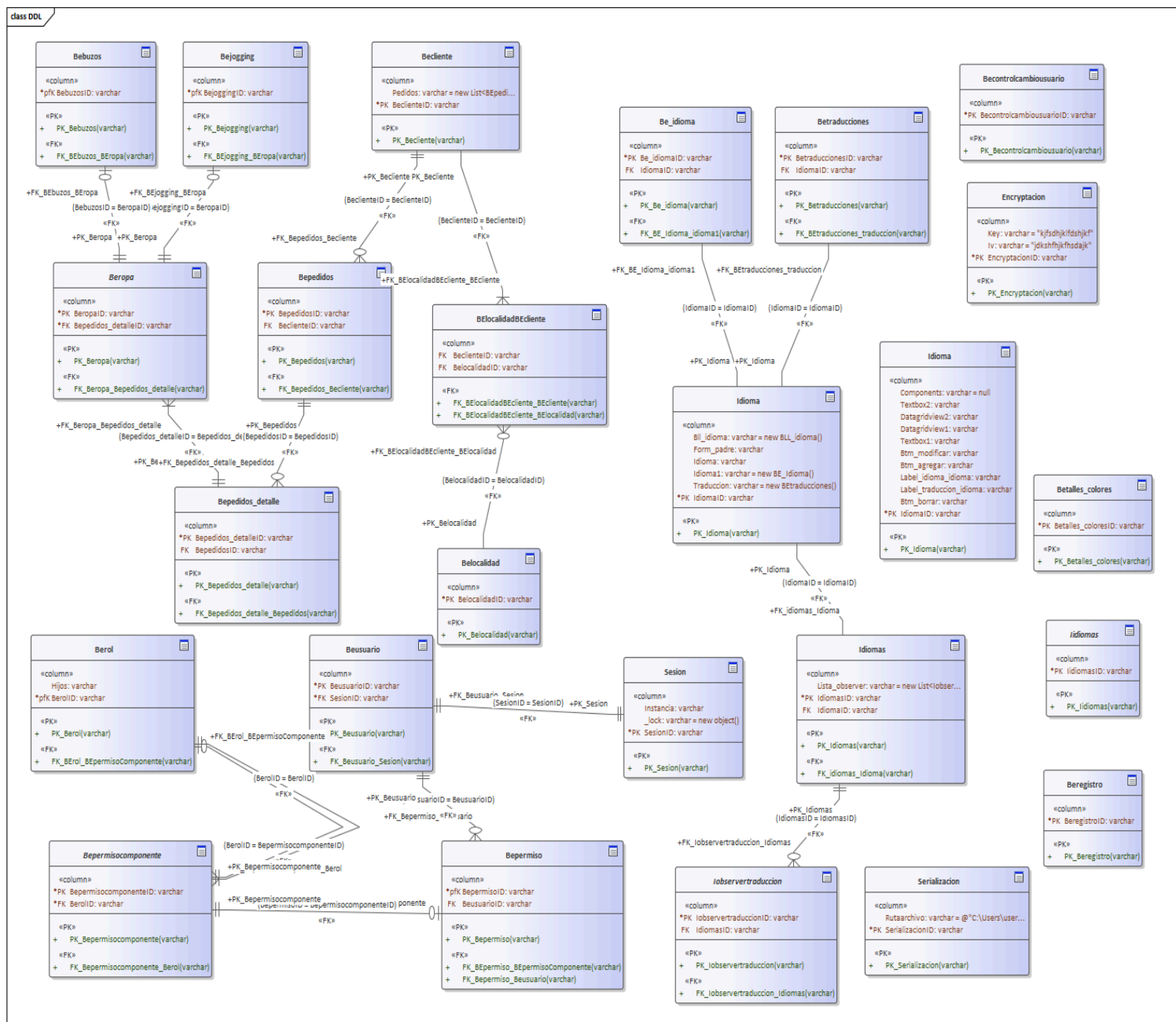
G06. Otros Requisitos

- Modelado UML para documentación (casos de uso, diagramas de clases y secuencia).
- Buenas prácticas de POO: bajo acoplamiento, alta cohesión, separación de capas (UI, BLL, DAL, BE).
- Base de datos: SQL Server
- Manual de usuario (procedimientos básicos para asesores y administradores).
- Documentación técnica: diagramas de arquitectura, diagramas de clases, DER y especificación de casos de uso.

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				Versión: 1.0
	<Nombre de Sección / Etapa>				

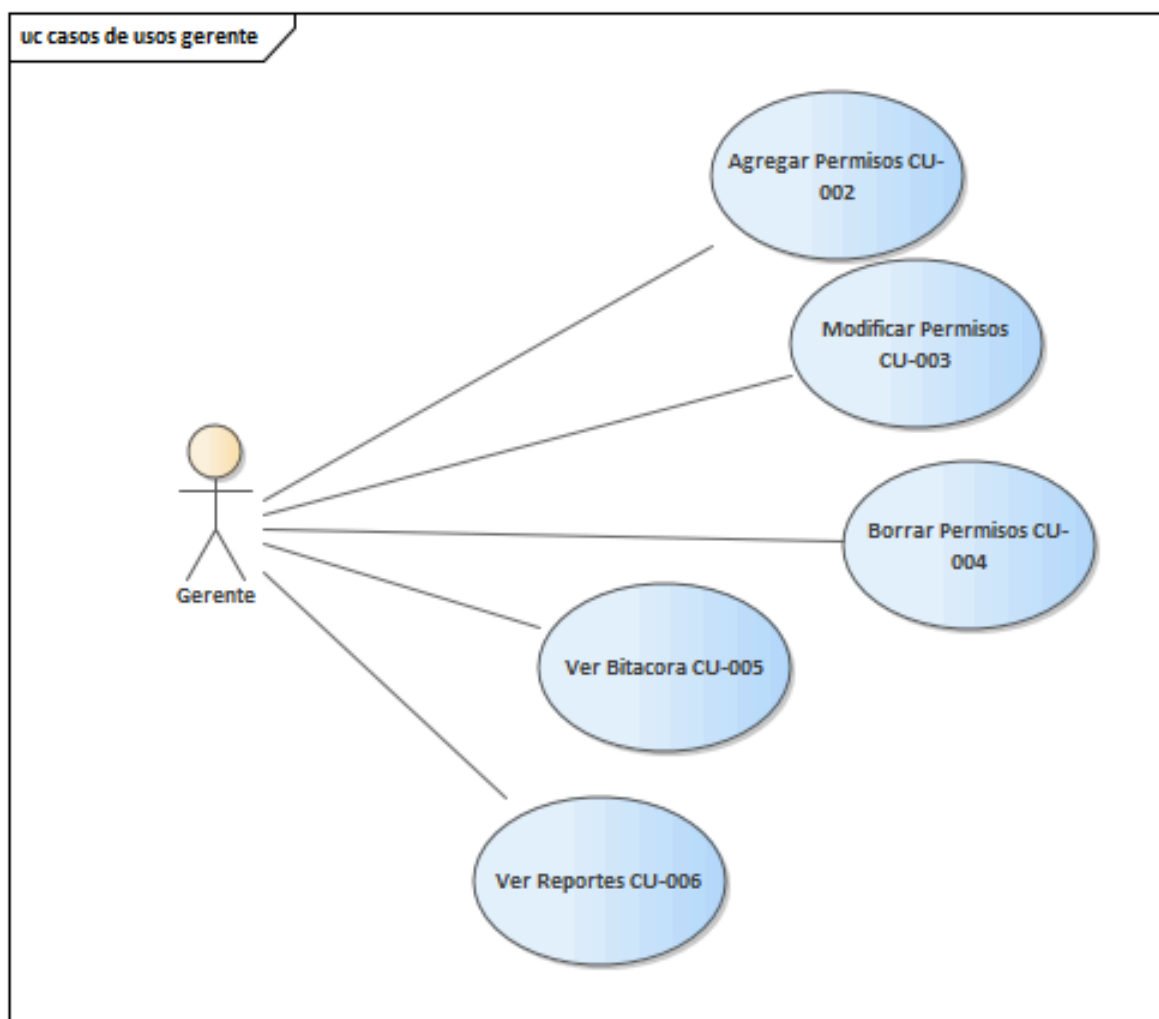
G08. Modelo de datos parcial de todos los módulos implementados

<p align="center">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p align="center">Facultad de Tecnología Informática</p>					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				Versión: 1.0
	<Nombre de Sección / Etapa>				



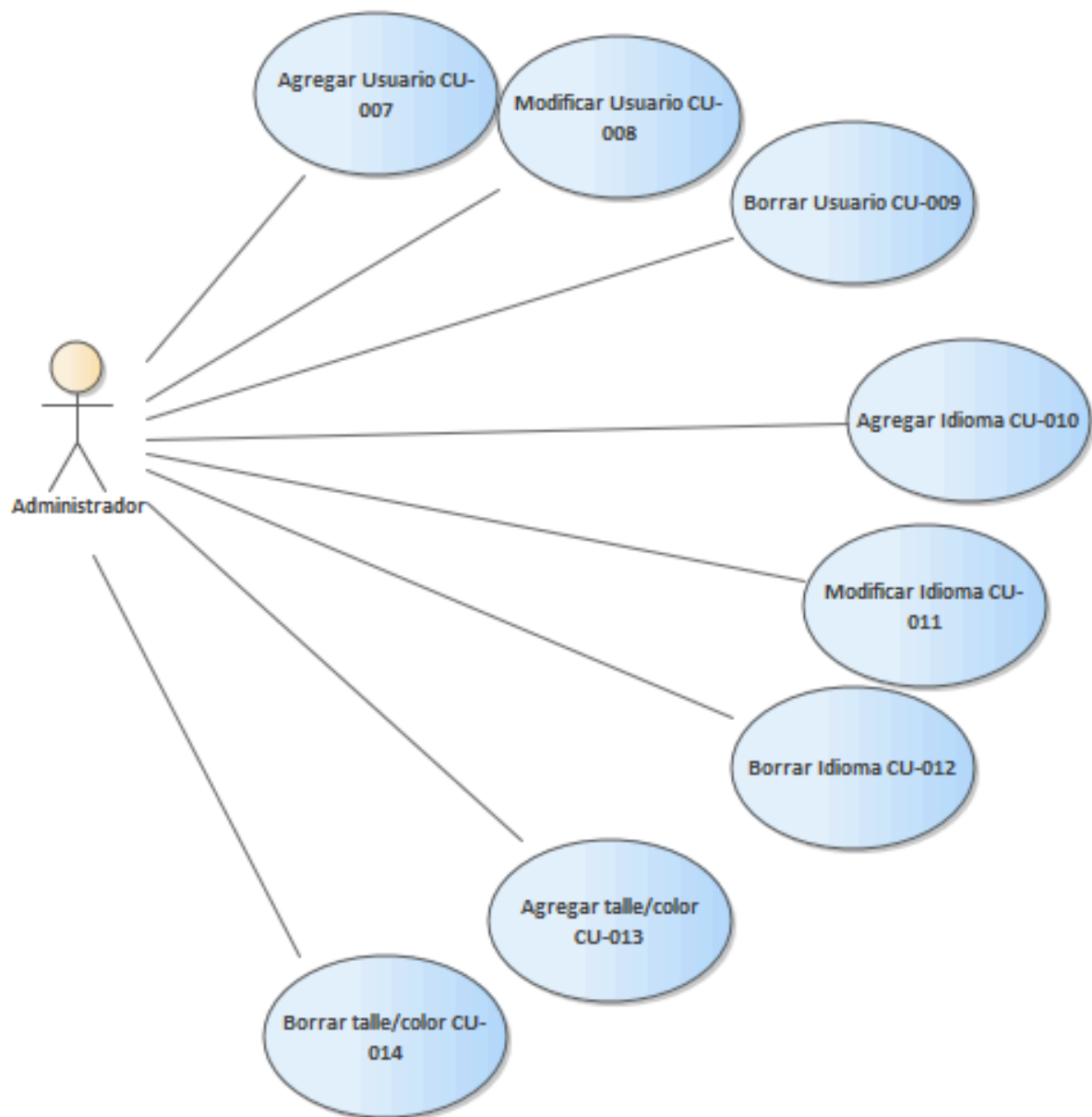
UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

N01. Especificación funcional por proceso de negocio

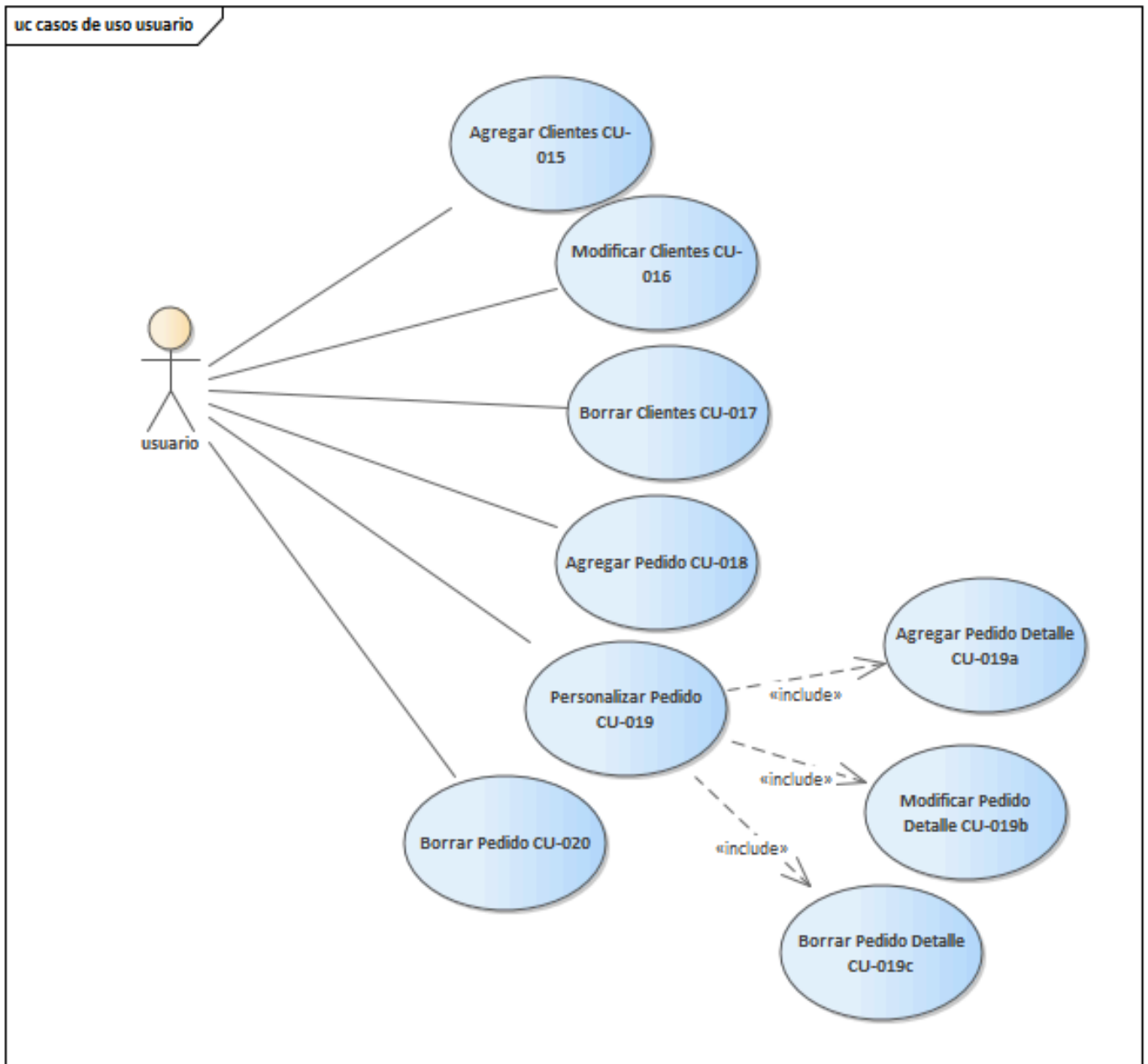


UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

uc casos de uso administrador



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

N02. Especificaciones de Casos de Uso

Nombre del Caso de uso: CU 002-Agregar Permisos
Actores: Gerente
Objetivo: Permitir que el usuario ingrese sus credenciales y acceder al sistema validando su identidad y cargando sus permisos.
Precondiciones: El actor debe haber iniciado sesión. Debe haber seleccionado un nodo del árbol (rol).
Postcondiciones: Se agrega un nuevo permiso o rol al árbol. El cambio queda persistido en la base de datos. Se actualiza la vista con el árbol modificado.
Eventos Disparadores: El gerente presiona el botón "Agregar" luego de seleccionar un nodo rol.
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El actor selecciona un nodo del árbol de permisos. 2. El actor presiona el botón "Agregar". 3. El sistema verifica que el nodo seleccionado no sea nulo. 4. El sistema determina si el gerente quiere agregar es un Permiso (según radiobutton). 5. El sistema verifica si existe un permiso con ese nombre para ese rol. 6. Si es válido, se crea el nuevo objeto Permiso.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

7. El sistema agrega el nuevo permiso y lo persiste en la base.
8. El sistema actualiza la UI mostrando los cambios.

Flujo Alternativo:

- 4A. El sistema determina que el gerente quiere agregar un Rol.
- 4A.2 El sistema verifica que no exista otro rol con ese nombre.
- 4A.3 Si el nombre es válido, el sistema crea el nuevo rol.
- 4A.4 El sistema agrega el nuevo rol al árbol y guarda los cambios en la base de datos.
- 4A.5 El sistema actualiza la interfaz mostrando el árbol con el nuevo rol.

2A1 – Nodo no seleccionado

El actor presiona “Agregar” sin haber seleccionado un nodo.

El sistema muestra un mensaje de error: *“Debe seleccionar un nodo del árbol.”*

Se regresa al paso 1 del flujo principal.

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

Materia: Ingeniería de software

Docente: Julián Martín Rodríguez

Fecha: 15/4/25

Alumno: Tomas Meneces Fernandez Benjamin

Legajo:

Localización: Castelar

Comisión: 2-A

Turno: Noche

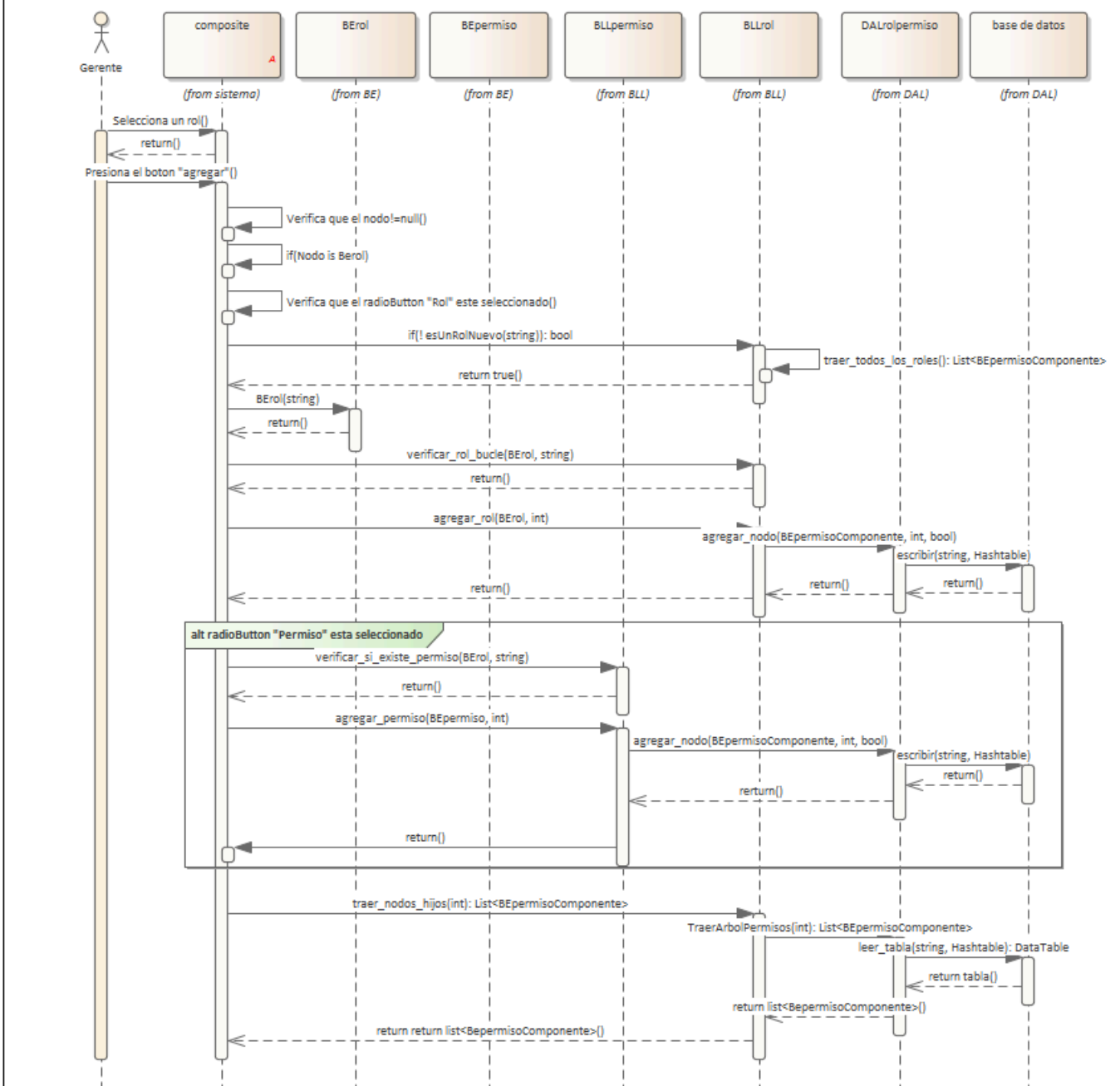
Año: 3

sistema de ropa

<Nombre de Sección / Etapa>

Versión: 1.0

sd agregar_nodo_permisos



<p>UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p>Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 003-Modificar Permisos
Actores: Gerente
Objetivo: Modificar los datos de un permiso o un rol existente.
Precondiciones:El actor debe haber iniciado sesión. Debe haber seleccionado un nodo del árbol (rol).
Postcondiciones: Se modifica el permiso o rol al árbol. El cambio queda persistido en la base de datos. Se actualiza la vista con el árbol modificado.
Eventos Disparadores: El gerente presiona el botón "Modificar" luego de seleccionar un nodo rol.
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El actor selecciona un nodo del árbol de permisos. 2. El actor presiona el botón "Modificar". 3. El sistema verifica que el nodo seleccionado no sea nulo. 4. El sistema pregunta al gerente si quiere borrar un Permiso/rol. 5. El gerente confirma la acción. 6. El sistema elimina el objeto Permiso/Rol. 7. El sistema persiste los cambios en la base de datos.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

8. El sistema actualiza la UI reflejando la eliminación.

Flujo alternativo

-Modificar un Rol (en vez de un Permiso)

5A. El sistema determina que el elemento a modificar es un Rol.

7A. El sistema valida que no exista otro rol con el mismo nombre.

8A. Si es válido, el sistema actualiza el objeto Rol.

9A. El sistema guarda el rol actualizado en la base.

10A. El sistema actualiza la UI mostrando los cambios.

-Nodo inválido

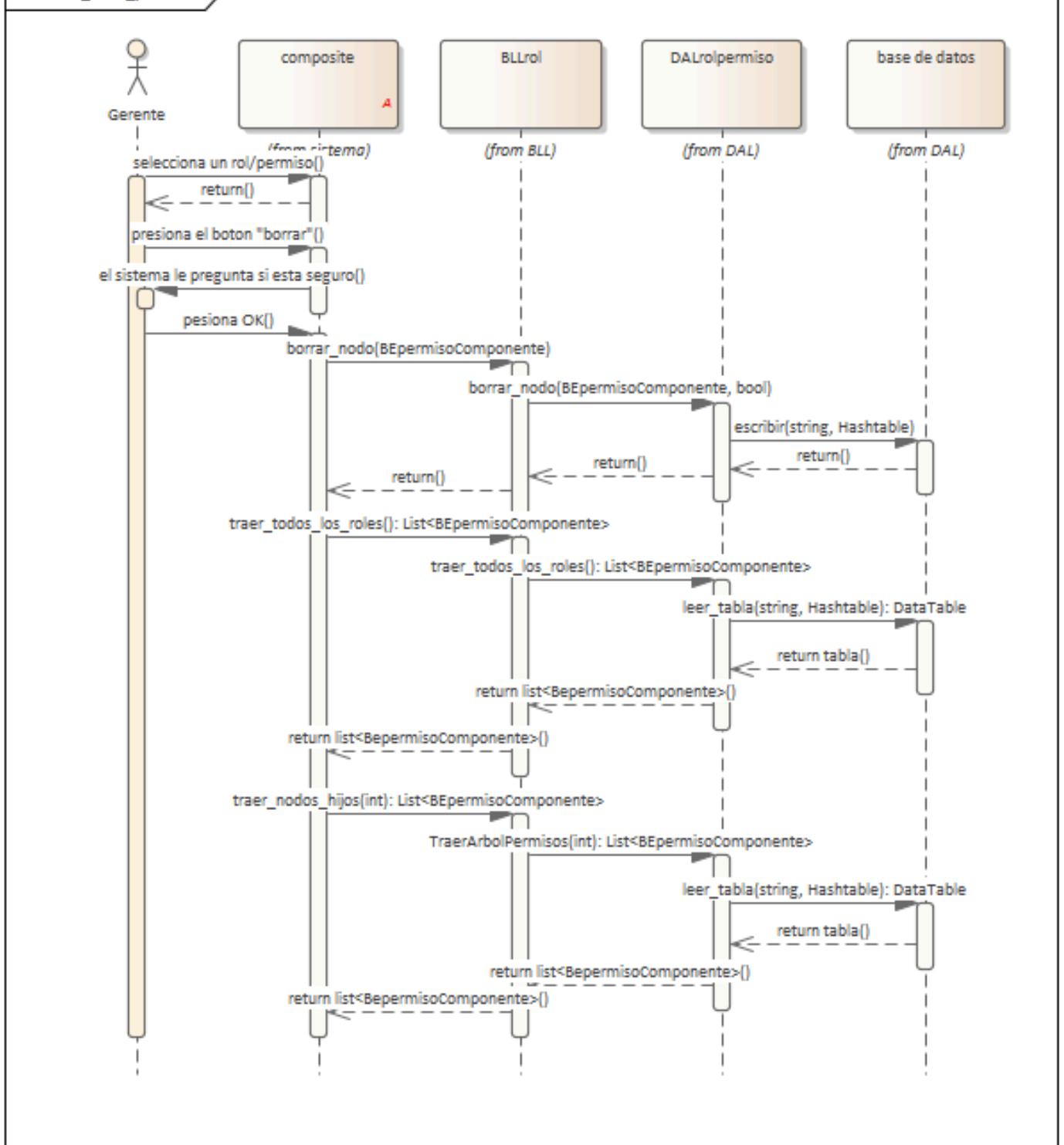
3B. Si el nodo seleccionado es nulo, el sistema muestra un mensaje de error:
"Debe seleccionar un permiso o rol para modificar".

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				
	<Nombre de Sección / Etapa>				Versión: 1.0

sd borrar_nodo_permisos



<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 004-Borrar Permisos
Actores: Gerente
Objetivo: Borrar un nodo del árbol
Precondiciones: Debe haber seleccionado un nodo del árbol (rol).
Postcondiciones: Se borra un nodo del árbol. El cambio queda persistido en la base de datos. Se actualiza la vista con el árbol modificado.
Eventos Disparadores: El gerente presiona el botón “borrar” luego de seleccionar un nodo rol.
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El actor selecciona un nodo del árbol de permisos. 2. El actor presiona el botón "Borrar". 3. El sistema verifica que el nodo seleccionado no sea nulo. 4. El sistema determina si se quiere borrar un Permiso. 5. El sistema solicita confirmación al actor. 6. El actor confirma la acción. 7. El sistema elimina el objeto Permiso. 8. El sistema persiste los cambios en la base de datos. 9. El sistema actualiza la UI reflejando la eliminación.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Flujo Alternativo

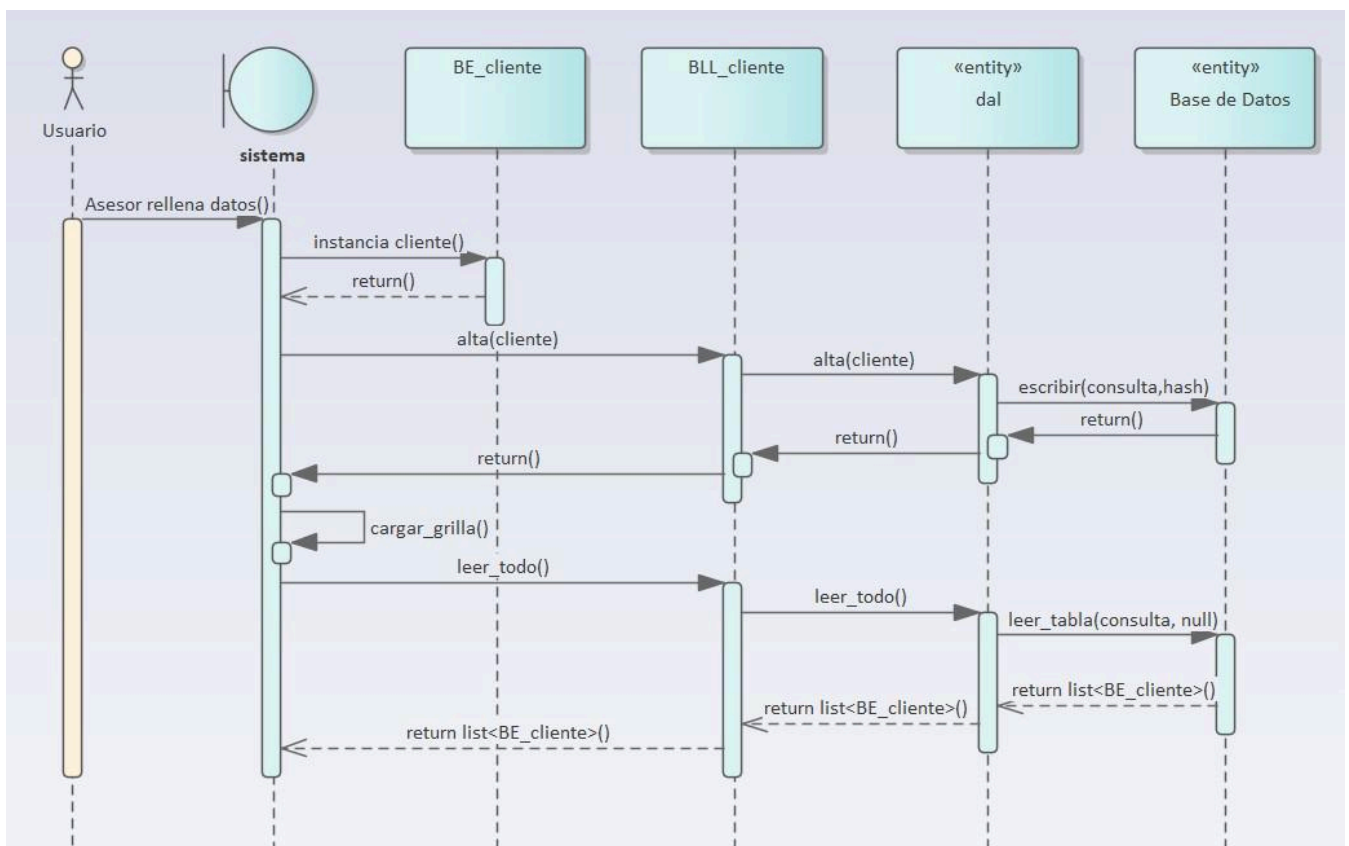
5a. Si el actor cancela la confirmación, el caso de uso termina sin cambios.

Flujo Alternativo b – Nodo inválido

3b. Si el nodo seleccionado es nulo, el sistema muestra:

“Debe seleccionar un permiso o rol para borrar”.

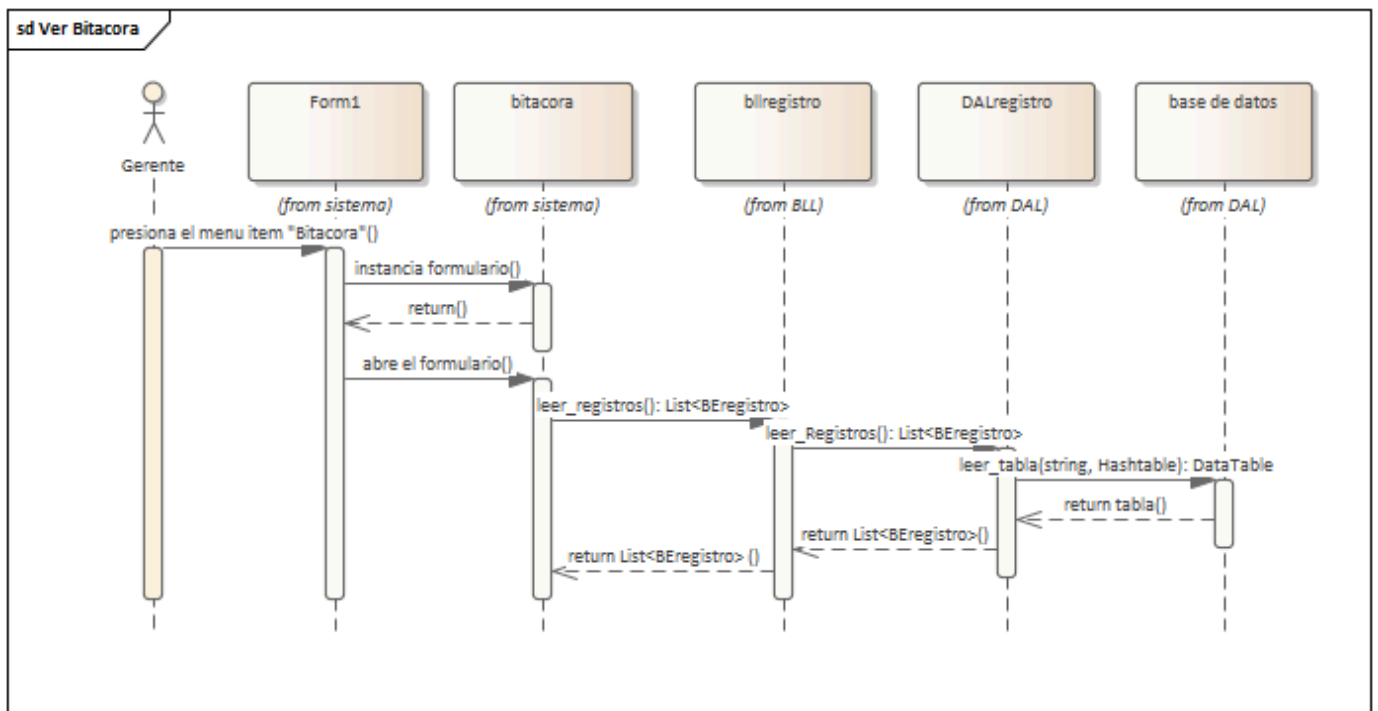
4b. El caso de uso finaliza.



<p>UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p>Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 005-Ver Bitácora
Actores: Gerente
Objetivo: Permitir que el gerente visualice todos los registros de la bitácora del sistema.
Precondiciones: El gerente inició sesión previamente. El actor posee permisos para ver la bitácora.
Postcondiciones: Se muestra al gerente la lista de registros de la bitácora disponibles.
Eventos Disparadores: El gerente presiona la opción de menú “Bitácora”.
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El actor selecciona la opción “Bitácora” en el menú. 2. El sistema crea e inicia la ventana o formulario de bitácora. 3. El sistema abre el formulario y lo muestra al actor. 4. El sistema obtiene los registros de la bitácora de la base de datos. 5. El formulario muestra al actor el listado completo de registros de la bitácora.
<p>Flujo Alternativo</p> <p>4A. El sistema detecta un error al intentar consultar la base de datos.</p> <p>5A. El sistema notifica al formulario que ocurrió un error.</p> <p>6A. El formulario muestra un mensaje informando que no fue posible cargar la bitácora.</p>

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				



Nombre del Caso de uso: CU007-Agregar Usuario
Actores: Administrador
Objetivo: Permitir que el administrador cree un nuevo usuario en el sistema.
Precondiciones: El gerente inició sesión previamente. El actor posee permisos para ver la bitácora.

<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

<p>Postcondiciones: El nuevo usuario queda registrado en la base de datos. Se crea un registro de control de cambios asociado a la creación. La lista de usuarios se actualiza y muestra el nuevo usuario.</p>
<p>Eventos Disparadores: El administrador presiona la opción “Agregar Usuario”.</p>
<p>Puntos de extensión: Ninguno</p>
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El administrador ingresa los datos del nuevo usuario en los cuadros de texto. 2. El administrador presiona el botón “Agregar”. 3. El sistema verifica que los cuadros de texto obligatorios no estén vacíos. 4. El sistema crea el nuevo usuario con los datos ingresados. 5. El sistema encripta la información sensible del usuario. 6. El sistema guarda al nuevo usuario en la base de datos. 7. El sistema registra la creación en la bitácora de control de usuario. 8. El sistema actualiza y muestra la lista de usuarios incluyendo el nuevo registro.
<p>Flujo Alternativo</p> <p>A1 – Campos vacíos (reemplaza el paso 3)</p> <ol style="list-style-type: none"> 3A. El sistema detecta uno o más campos obligatorios vacíos. 4A. El sistema muestra un mensaje indicando que deben completarse. 5A. El sistema no crea el usuario. Fin del flujo alternativo.

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

Materia: Ingeniería de software

Docente: Julián Martín Rodríguez

Fecha: 15/4/25

Alumno: Tomas Meneces Fernandez Benjamin

Legajo:

Localización: Castelar

Comisión: 2-A

Turno: Noche

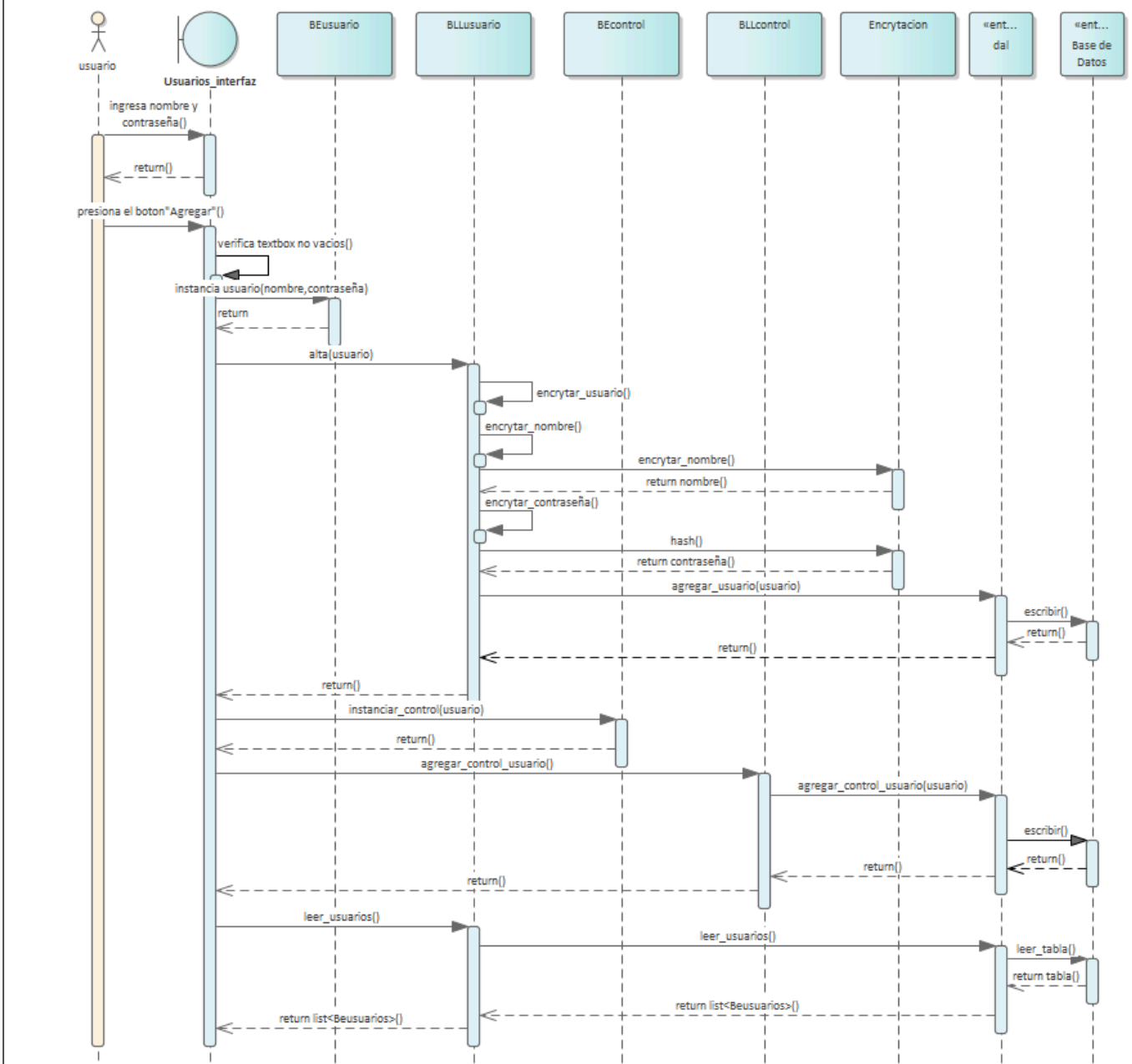
Año: 3

sistema de ropa

Versión: 1.0

<Nombre de Sección / Etapa>

sd diagrama agregar_usuario



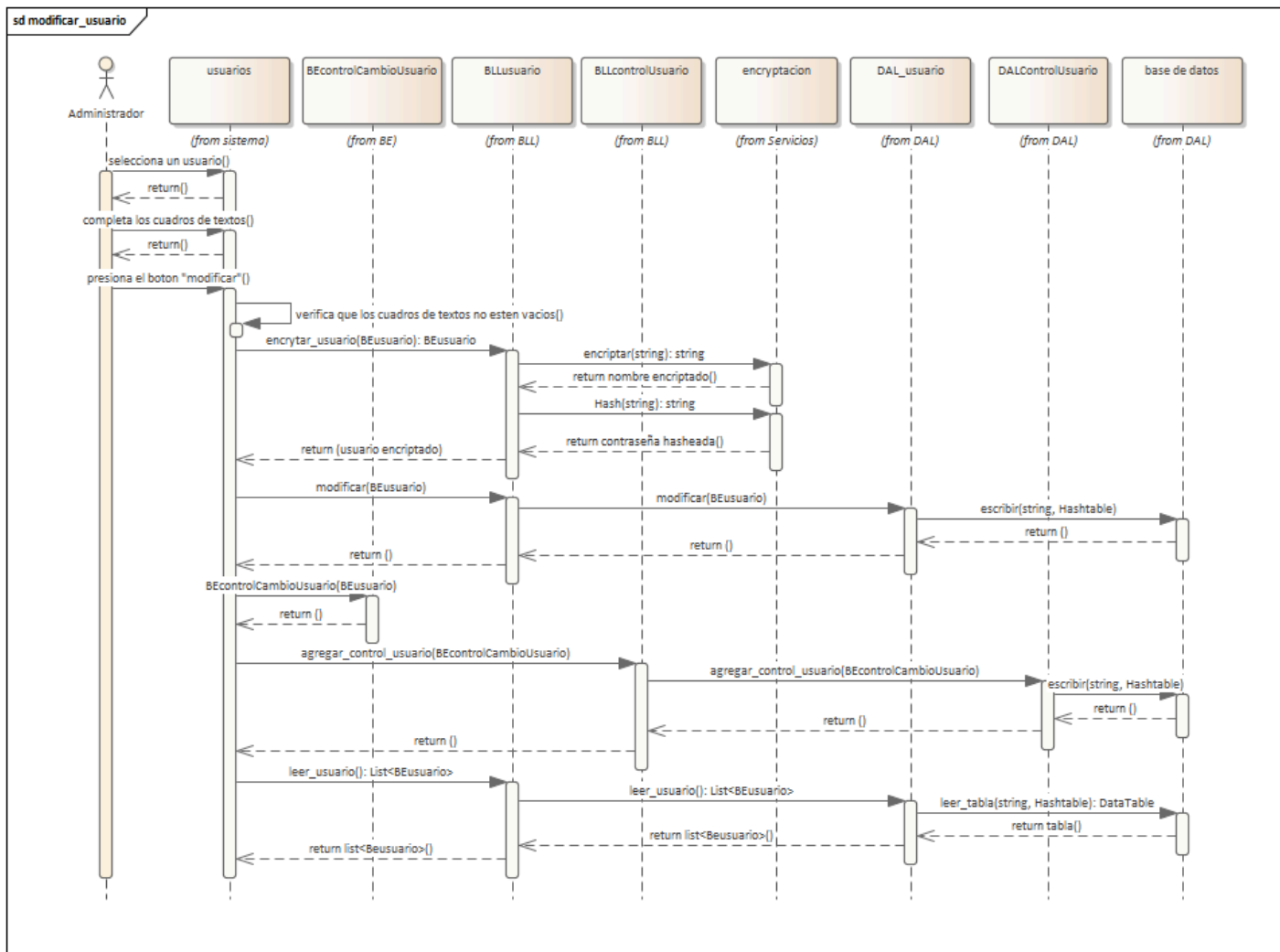
UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU008-Modificar Usuario
Actores: Administrador
Objetivo: Permitir al administrador actualizar los datos de un usuario (nombre, contraseña) y dejar la lista de usuarios actualizada.
Precondiciones: El administrador ha iniciado sesión. El administrador tiene permisos para modificar usuarios. Existe al menos un usuario listado.
Postcondiciones: Los datos del usuario quedan actualizados en el sistema. La lista de usuarios se muestra con la información modificada. Se registra la acción en la bitácora (control de cambios).
Eventos Disparadores: El administrador selecciona la opción “Modificar” o abre el formulario de edición desde la lista de usuarios.
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> El actor selecciona un usuario de la lista. El sistema completa los campos del formulario con los datos actuales. El actor modifica los datos necesarios. El actor presiona el botón “Modificar”. El sistema verifica que los campos obligatorios no estén vacíos. El sistema actualiza los datos del usuario (incluyendo nombre y contraseña). El sistema guarda los cambios en la base de datos. El sistema registra la modificación en la bitácora de control de cambios.

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

17. El sistema muestra al actor la lista de usuarios.
<p>Flujo Alternativo</p> <p>-Campos vacíos</p> <p>5A. El sistema detecta que faltan datos obligatorios.</p> <p>6A. El sistema muestra un mensaje solicitando completar los campos.</p> <p>7A. El sistema no modifica nada.</p> <p>8A. Fin del flujo alternativo.</p>

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



Nombre del Caso de uso: CU009-Borrar Usuario
Actores: Administrador
Objetivo: borrar un usuario del sistema

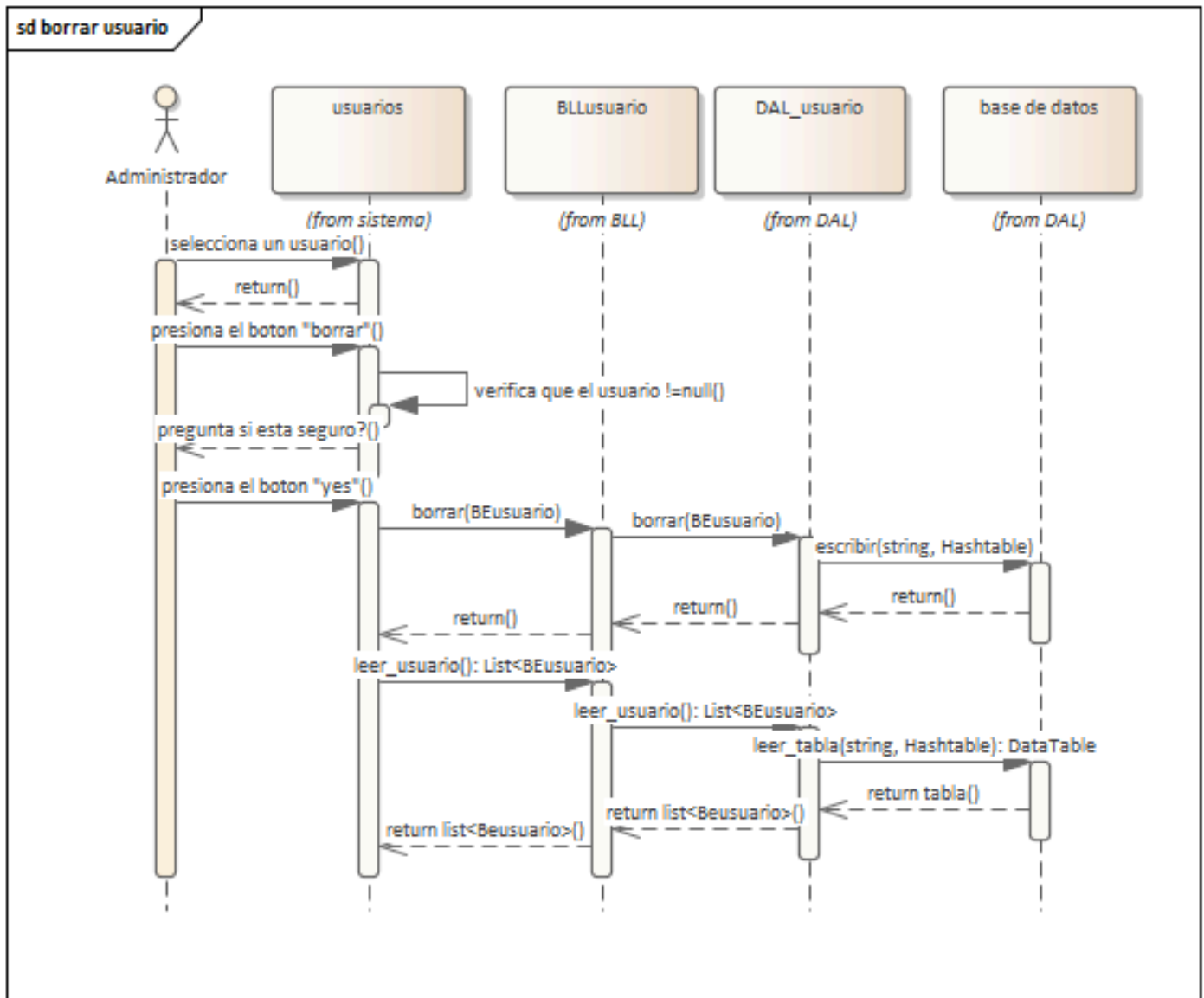
UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Precondiciones: El Administrador inició sesión previamente.
Postcondiciones: Se elimina un usuario del sistema y se persiste en la base de datos
Eventos Disparadores: El administrador presiona el botón “Borrar” después de seleccionar un usuario.
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 6. El actor selecciona un usuario de la lista. 7. El actor presiona el botón “Borrar”. 8. El sistema verifica que el usuario seleccionado no sea nulo. 9. El sistema solicita confirmación al actor. 10. El actor confirma la eliminación. 11. El sistema elimina el usuario y guarda los cambios en la base de datos. 12. El sistema actualiza la lista de usuarios. 13. El sistema muestra al actor la lista actualizada.
<p>Flujo Alternativo</p> <p>A1 – Usuario cancela la eliminación</p> <ol style="list-style-type: none"> 5A. El actor selecciona “No” o cierra el cuadro de confirmación. 6A. El sistema cancela la operación <p>A2 – No se ha seleccionado un usuario</p> <ol style="list-style-type: none"> 3A. El sistema detecta que no hay un usuario seleccionado. 4A. El sistema muestra un mensaje indicando que debe seleccionarse un

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

usuario.

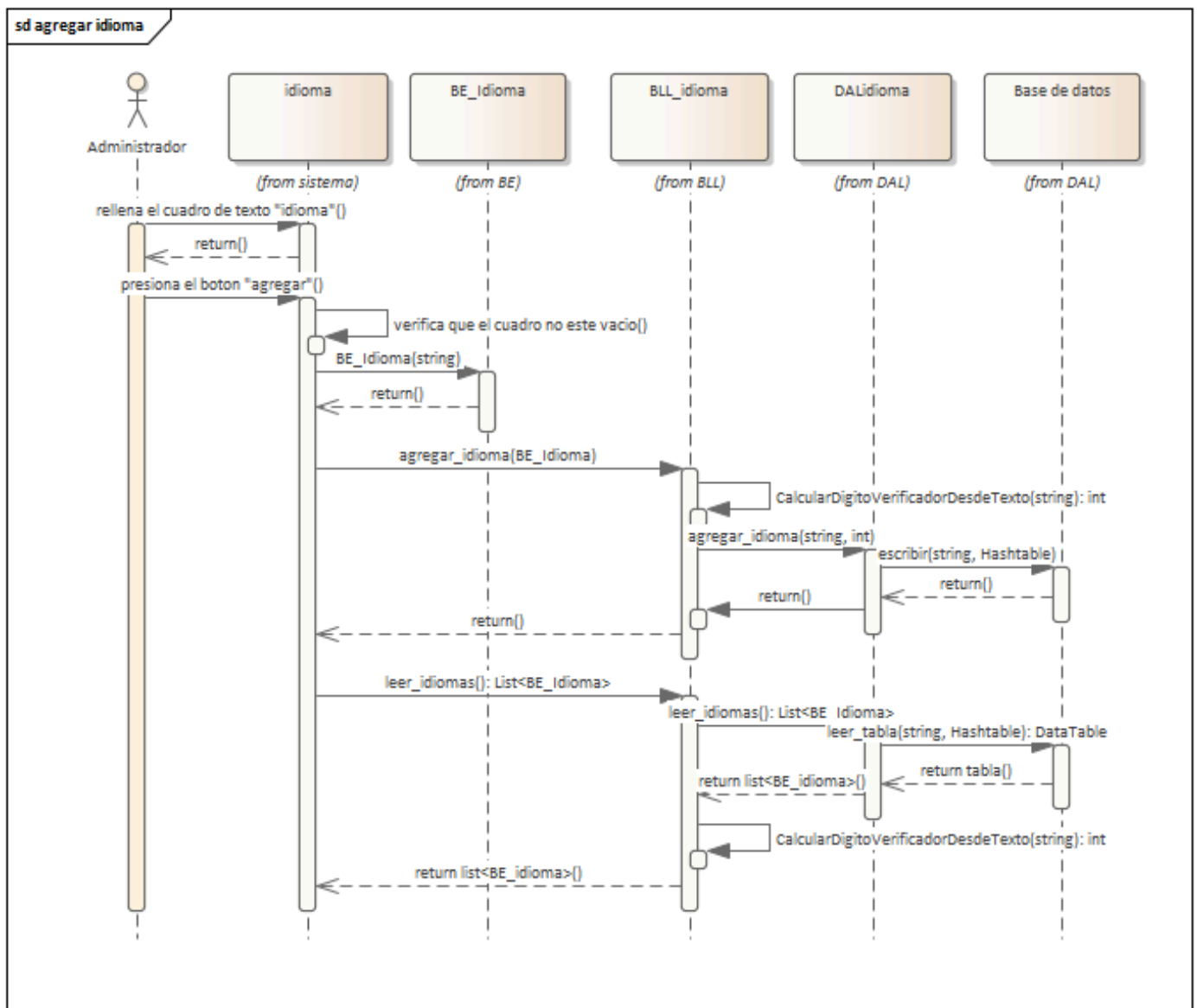
5A. El sistema cancela la operación. Fin del flujo alternativo.



<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 010-Agregar idioma
Actores: Administrador
Objetivo: Permitir que el administrador registre un nuevo idioma en el sistema.
Precondiciones: El administrador ha iniciado sesión. El administrador tiene permisos para gestionar idiomas.
Postcondiciones: El nuevo idioma queda registrado en la base de datos. La lista de idiomas se actualiza y muestra el nuevo idioma.
Eventos Disparadores: El administrador selecciona la opción “Agregar Idioma”.
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El administrador completa los cuadros de texto con los datos del nuevo idioma. 2. El administrador presiona el botón “Agregar”. 3. El sistema verifica que los cuadros de texto obligatorios no estén vacíos. 4. El sistema calcula el dígito verificador 5. El sistema guarda el nuevo idioma en la base de datos. 6. El sistema actualiza y muestra la lista de idiomas incluyendo el nuevo idioma.
<p>Flujo Alternativo</p> <p>A1 – Campos vacíos</p> <ol style="list-style-type: none"> 3A. El sistema detecta que faltan datos obligatorios. 4A. El sistema muestra un mensaje solicitando completar los campos. 5A. El sistema no agrega el idioma. Fin del flujo alternativo.

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

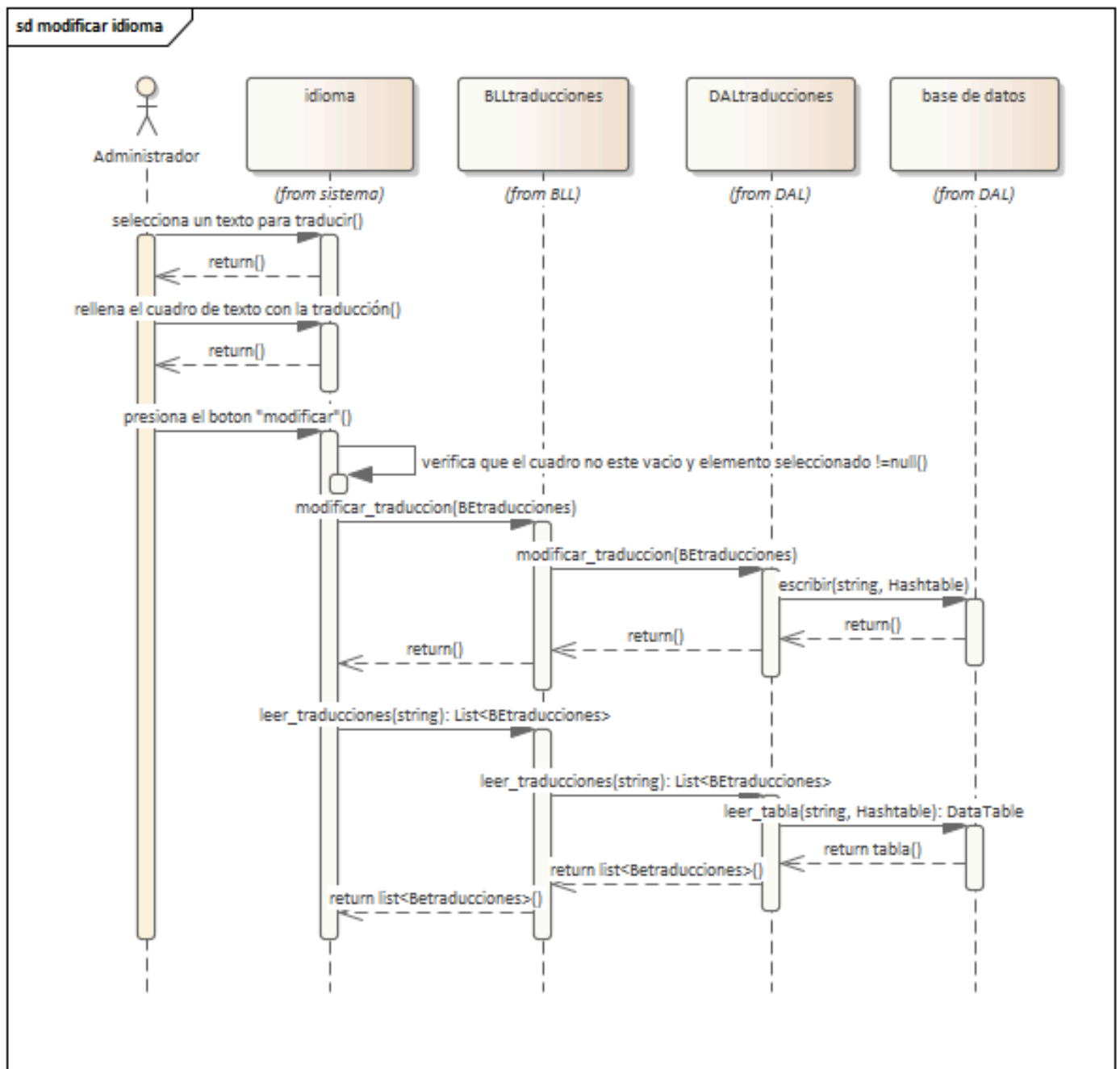


Nombre del Caso de uso: CU 011-Modificar Idioma
Actores: Administrador

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

Objetivo: Permitir que el administrador pueda modificar las traducciones del idioma
Precondiciones: El administrador inició sesión previamente. El administrador posee permisos para modificar las traducciones
Postcondiciones: Se carga en la base de datos la traducción nueva
Eventos Disparadores: El administrador ingresa traducción y presiona el botón “modificar”
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El administrador completa los cuadros de texto con los datos del nuevo idioma. 2. El administrador selecciona una traducción 3. El administrador presiona el botón “Modificar”. 4. El sistema verifica que los cuadros de texto obligatorios no estén vacíos. 5. El sistema modifica la nueva traducción en la base de datos. 6. El sistema actualiza y muestra la lista de idiomas incluyendo el nuevo idioma.
<p>Flujo Alternativo</p> <p>-cuadros de textos vacíos</p> <p>4A. El sistema detecta que los cuadros de textos están vacíos.</p> <p>5A. El sistema notifica al formulario que ocurrió un error.</p> <p>6A. El formulario muestra un mensaje informando que el administrador complete los cuadros de textos</p>

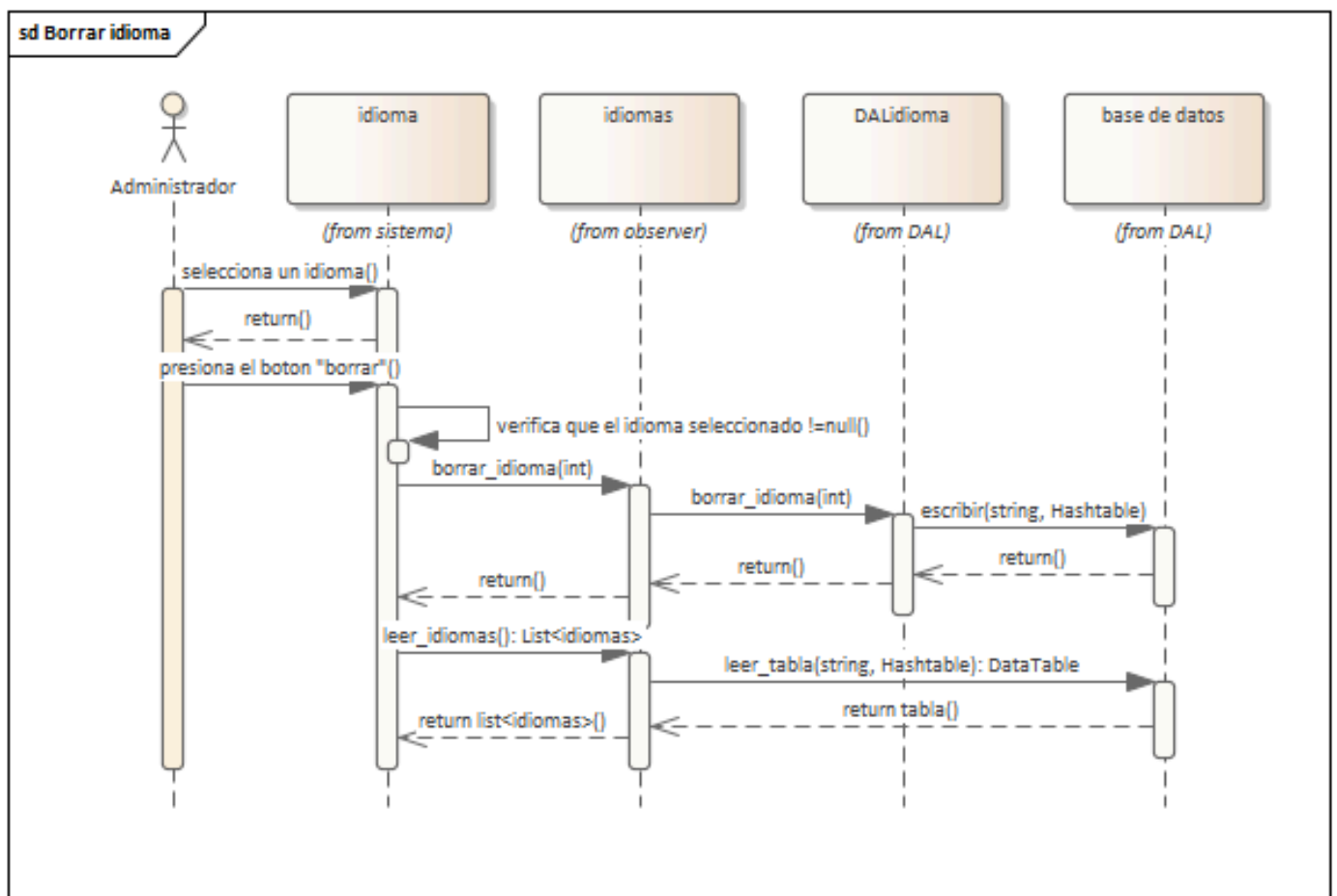
UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



<p>UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p>Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 012-Borrar Idioma
Actores: Administrador
Objetivo: Permitir al administrador eliminar un idioma y todas sus traducciones de la base de datos
Precondiciones: El administrador inició sesión previamente. El administrador posee permisos para borrar idioma
Postcondiciones: Se borra el idioma y todas sus traducciones de la base de datos
Eventos Disparadores: El administrador selecciona un idioma y presiona el botón "borrar"
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El administrador completa los cuadros de texto con los datos del nuevo idioma. 2. El administrador presiona el botón "borrar". 3. El sistema verifica que haya un idioma seleccionado. 4. El sistema elimina el idioma y las traducciones de la base de datos. 5. El sistema actualiza y muestra la lista de idiomas.
<p>Flujo Alternativo</p> <p>-idioma no seleccionado.</p> <p>3A el idioma seleccionado es igual a nulo</p> <p>4A el sistema muestra un mensaje de error</p>

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



Nombre del Caso de uso: CU 013-Agregar color/talle
Actores: administrador
Objetivo: Permitir que se agreguen colores y talles en la base de datos
Precondiciones: El administrador inició sesión previamente. El administrador posee permisos para agregar colores y talles.

<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

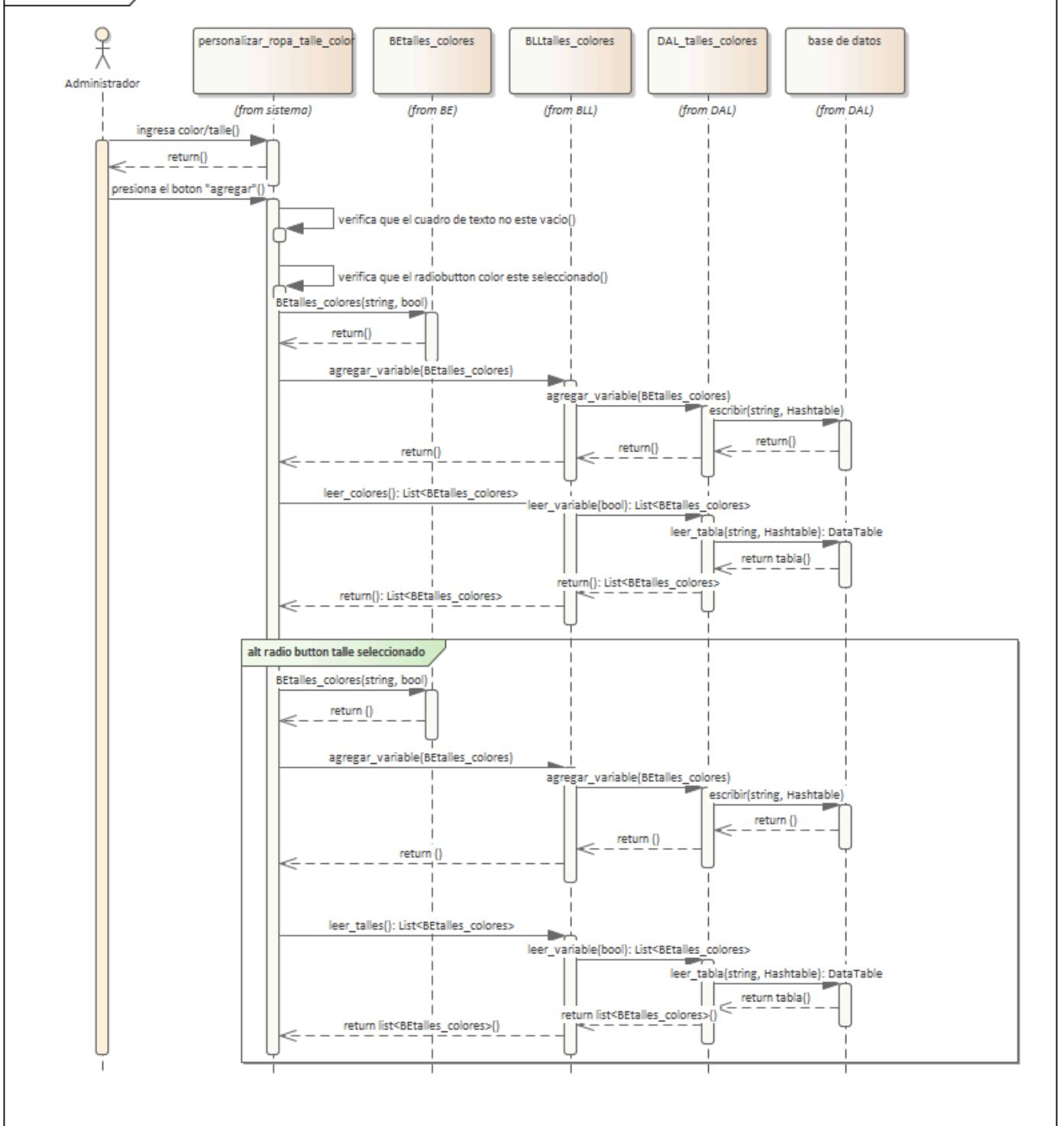
Postcondiciones: se agregaron colores y talles en la base de datos
Eventos Disparadores: El administrador completa los cuadros y presiona el botón "agregar".
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El administrador rellena el cuadro de texto talle/color. 2. El administrador presiona el botón "agregar". 3. El sistema verifica que los cuadros de textos no estén vacíos. 4. El sistema verifica que color esté seleccionado. 5. El sistema guarda el color en la base de datos. 6. El sistema muestra la lista actualizada de los colores.
<p>Flujo Alternativo</p> <p>-cuadros vacíos.</p> <p>3A. El sistema detecta cuadros de textos vacíos.</p> <p>4A. El sistema muestra un mensaje de error al administrador.</p> <p>-Talle seleccionado</p> <p>4A. El sistema verifica que talle esté seleccionado.</p> <p>5A. El sistema guarda el talle en la base de datos.</p> <p>6A. El sistema muestra la lista actualizada de los talles.</p>

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				
	<Nombre de Sección / Etapa>				Versión: 1.0

sd agregar color/talle



<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

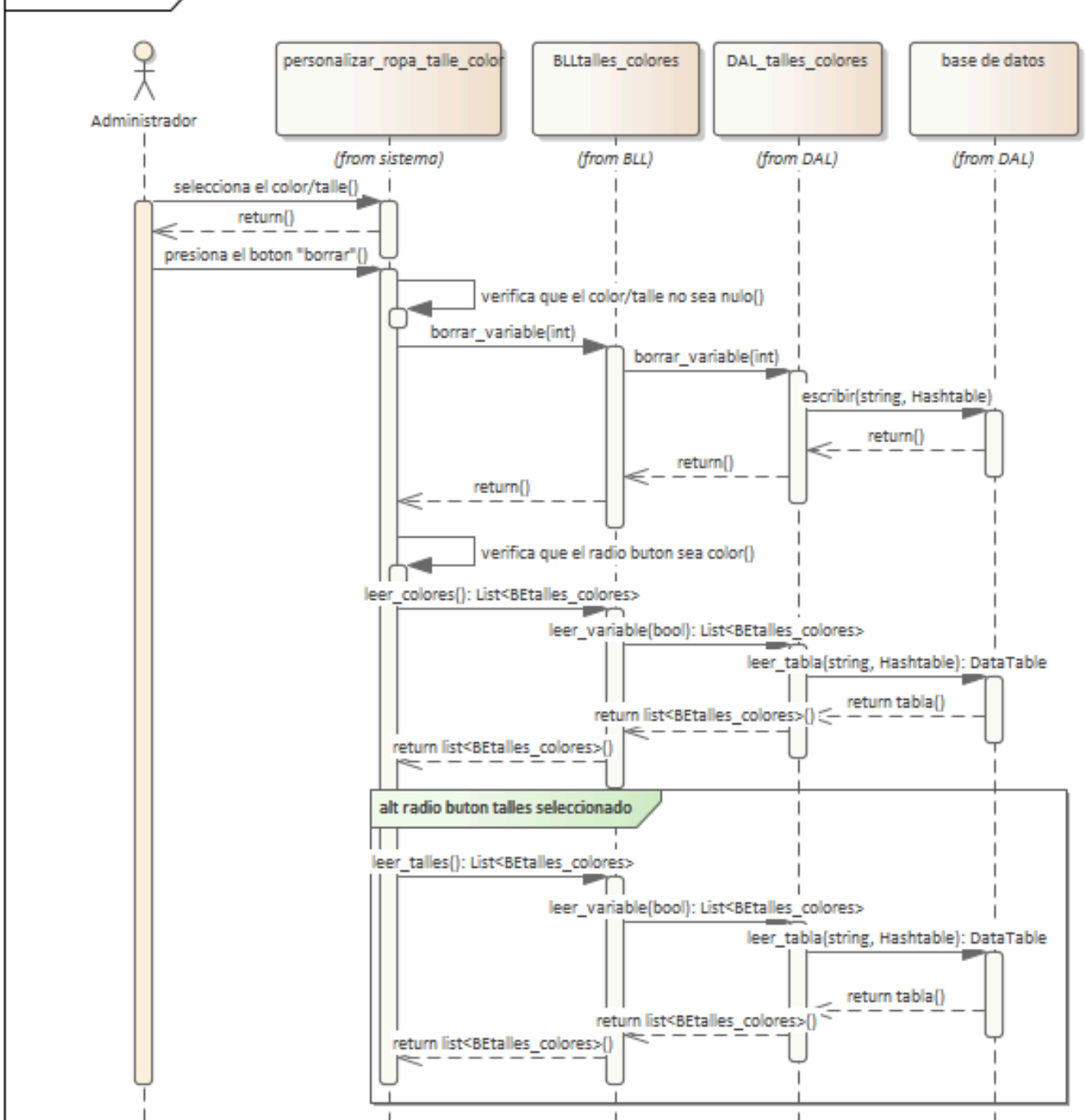
Nombre del Caso de uso: CU 014-Borrar color/talle
Actores: administrador
Objetivo: Permitir que se borren colores y talles de la base de datos
<p>Precondiciones: El administrador inició sesión previamente.</p> <p>El administrador posee permisos para borrar talles/colores</p> <p>Existe una lista de talles o colores en la base de datos.</p>
Postcondiciones: Se borran colores y talles de la base de datos
Eventos Disparadores: El administrador selecciona un talle/color y presiona el botón "borrar".
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El administrador selecciona un talla/color. 2. El administrador presiona el botón "borrar". 3. El sistema verifica que haya un talle/color seleccionado. 4. El sistema borra el color/talle de la base de datos. 5. El sistema verifica que el radiobutton color esté seleccionado. 6. El sistema muestra la lista actualizada de los colores.
<p>Flujo Alternativo</p> <p>-Radiobutton talle seleccionado.</p> <p>5A. El sistema verifica que el radiobutton talle esté seleccionado.</p> <p>6A. El sistema muestra la lista actualizada de los talles.</p>

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				
	<Nombre de Sección / Etapa>				Versión: 1.0

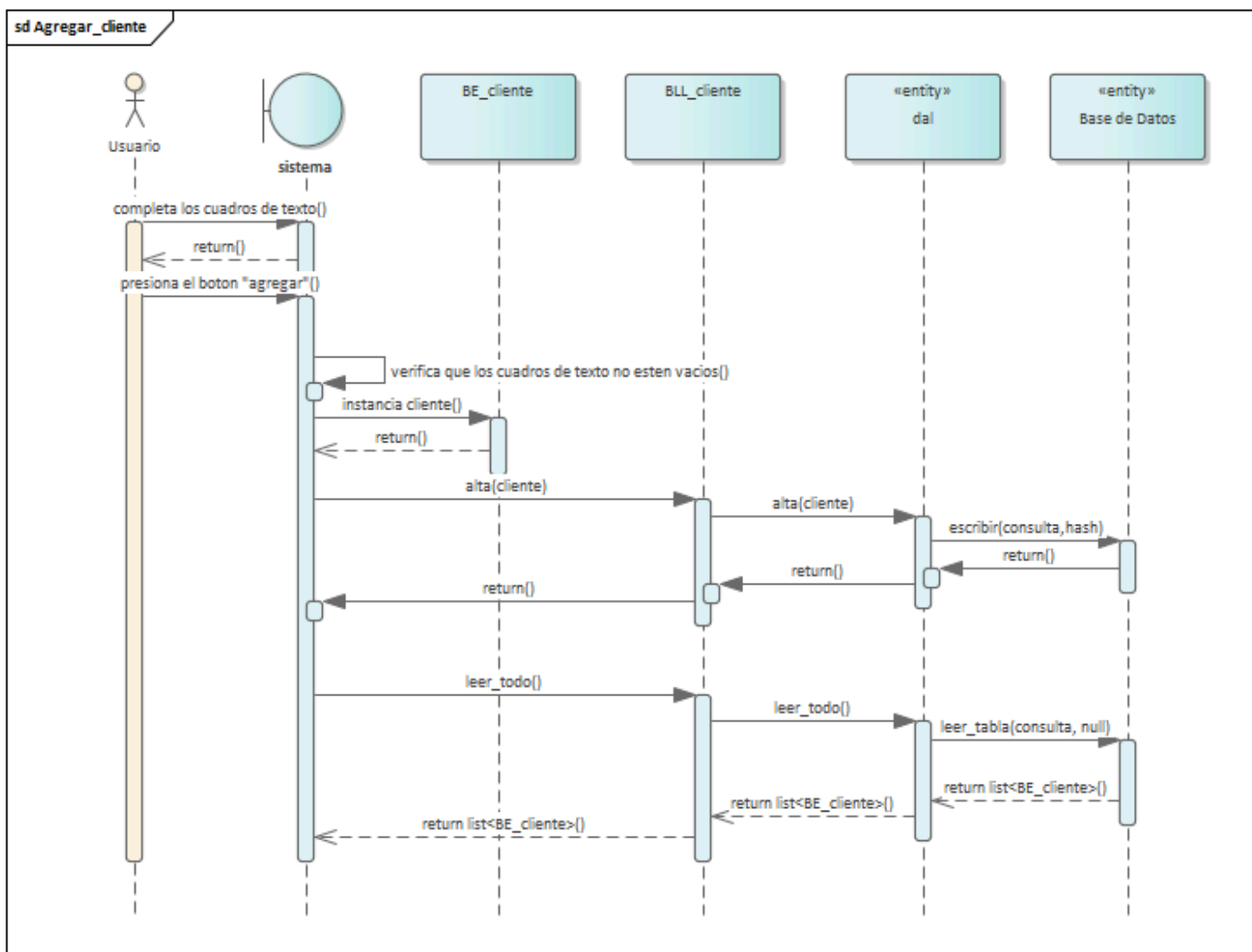
sd borrar color/talle



<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 015-Agregar Cliente
Actores: Usuario
Objetivo: Permitir que el usuario pueda agregar clientes nuevos al sistema y base de datos
Precondiciones: El usuario inició sesión previamente. El usuario posee permisos para agregar cliente
Postcondiciones: se agrega un cliente nuevo al sistema y base de datos
Eventos Disparadores: El usuario rellena los datos y presiona el botón “agregar”
Puntos de extensión: Ninguno
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El usuario rellena los cuadros de textos. 2. El usuario presiona el botón “agregar”. 3. El sistema verifica que los cuadros de textos no estén vacíos. 4. El sistema agrega al cliente al sistema y a la base de datos. 5. El sistema muestra lista actualizada de clientes
<p>Flujo Alternativo</p> <p>-cuadros vacíos.</p> <p>3A. El sistema detecta cuadros de textos vacíos.</p> <p>4A. El sistema muestra un mensaje de error al usuario..</p>

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

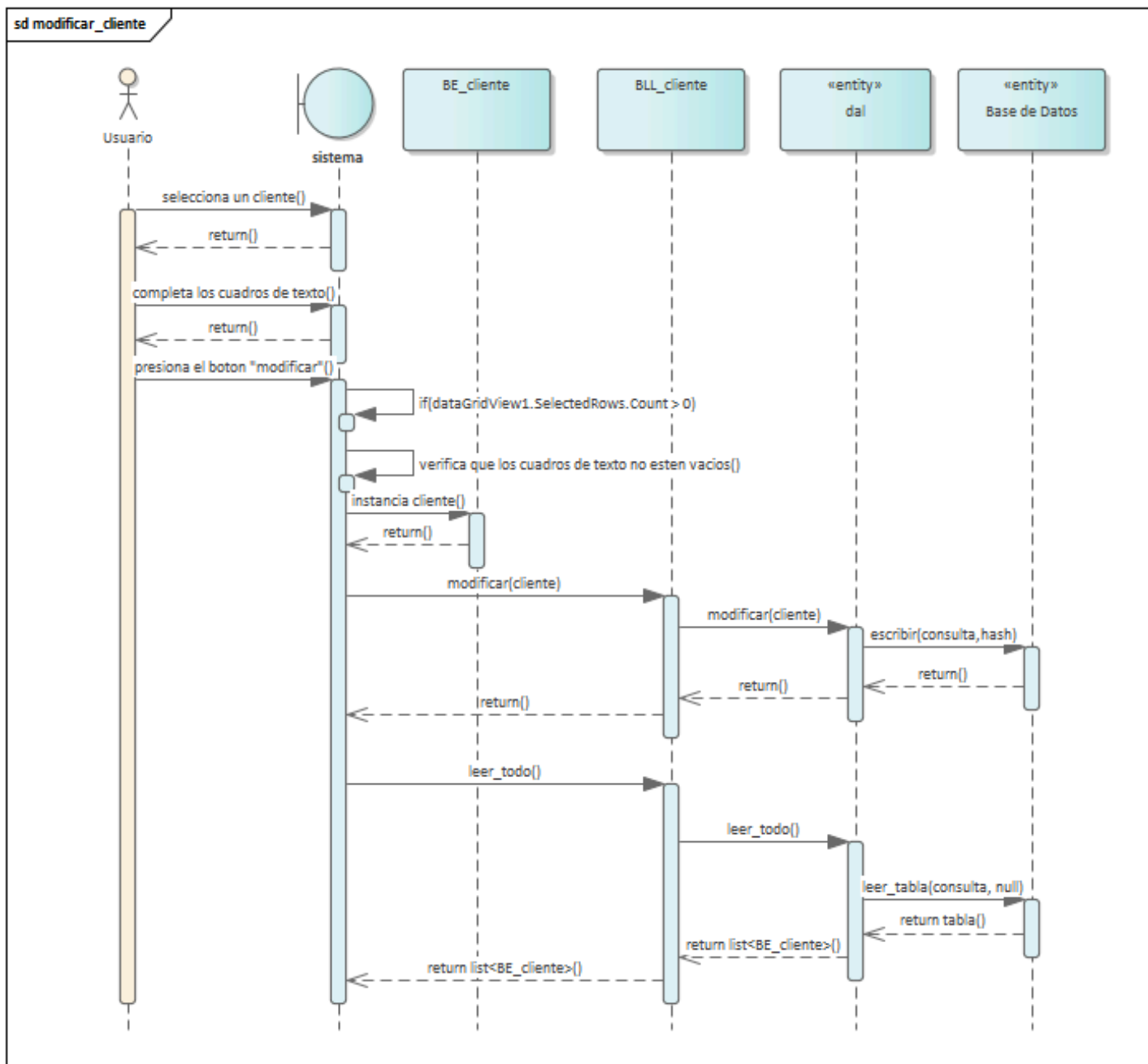


Nombre del Caso de uso: CU 016-Modificar Cliente
Actores: Usuario
Objetivo: Permitir que el usuario pueda modificar clientes al sistema y base de datos

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Precondiciones: El usuario inició sesión previamente.
El usuario posee permisos para modificar cliente
Postcondiciones:
se modifica un cliente existente en el sistema y base de datos
Eventos Disparadores: El usuario rellena los datos y presiona el botón “modificar”
Puntos de extensión: Ninguno
Flujo Principal <ol style="list-style-type: none"> 1. El usuario selecciona un cliente. 2. El usuario rellena los cuadros de textos. 3. El usuario presiona el botón “modificar”. 4. El sistema verifica que haya un cliente seleccionado. 5. El sistema verifica que los cuadros de textos no estén vacíos. 6. El sistema modifica la base de datos. 7. El sistema muestra lista actualizada de clientes
Flujo Alternativo -cuadros vacíos. 5A. El sistema detecta cuadros de textos vacíos. 6A. El sistema muestra un mensaje de error al usuario. -cliente no seleccionado. 4A. El sistema detecta que no hay un cliente seleccionado. 5A. El sistema muestra un mensaje de error al usuario.

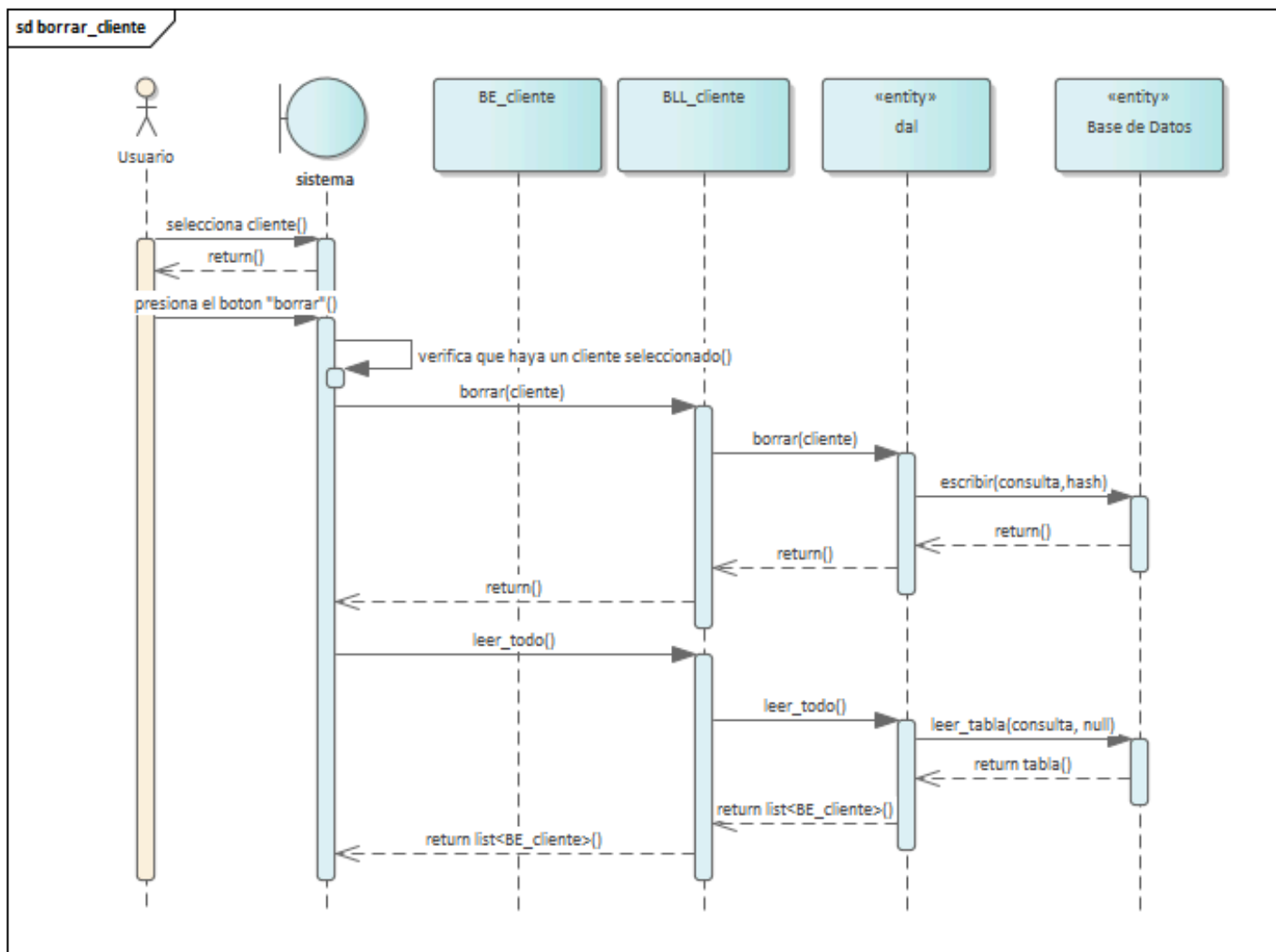
UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 017-Borrar cliente
Actores: Usuario
Objetivo: Permitir que el usuario pueda borrar clientes del sistema y la base de datos
Precondiciones: El usuario inició sesión previamente. El usuario posee permisos para borrar cliente
Postcondiciones: Se borra un cliente existente del sistema y base de datos
Eventos Disparadores: El usuario selecciona un cliente y presiona el botón “borrar”
Puntos de extensión: Ninguno
Flujo Principal <ol style="list-style-type: none"> 1. El usuario selecciona un cliente. 2. El usuario presiona el botón “borrar”. 3. El sistema verifica que haya un cliente seleccionado. 4. El sistema borra al cliente de la base de datos. 5. El sistema muestra lista actualizada de clientes
Flujo Alternativo -cliente no seleccionado. 3A. El sistema detecta que no hay un cliente seleccionado. 4A. El sistema muestra un mensaje de error al usuario.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



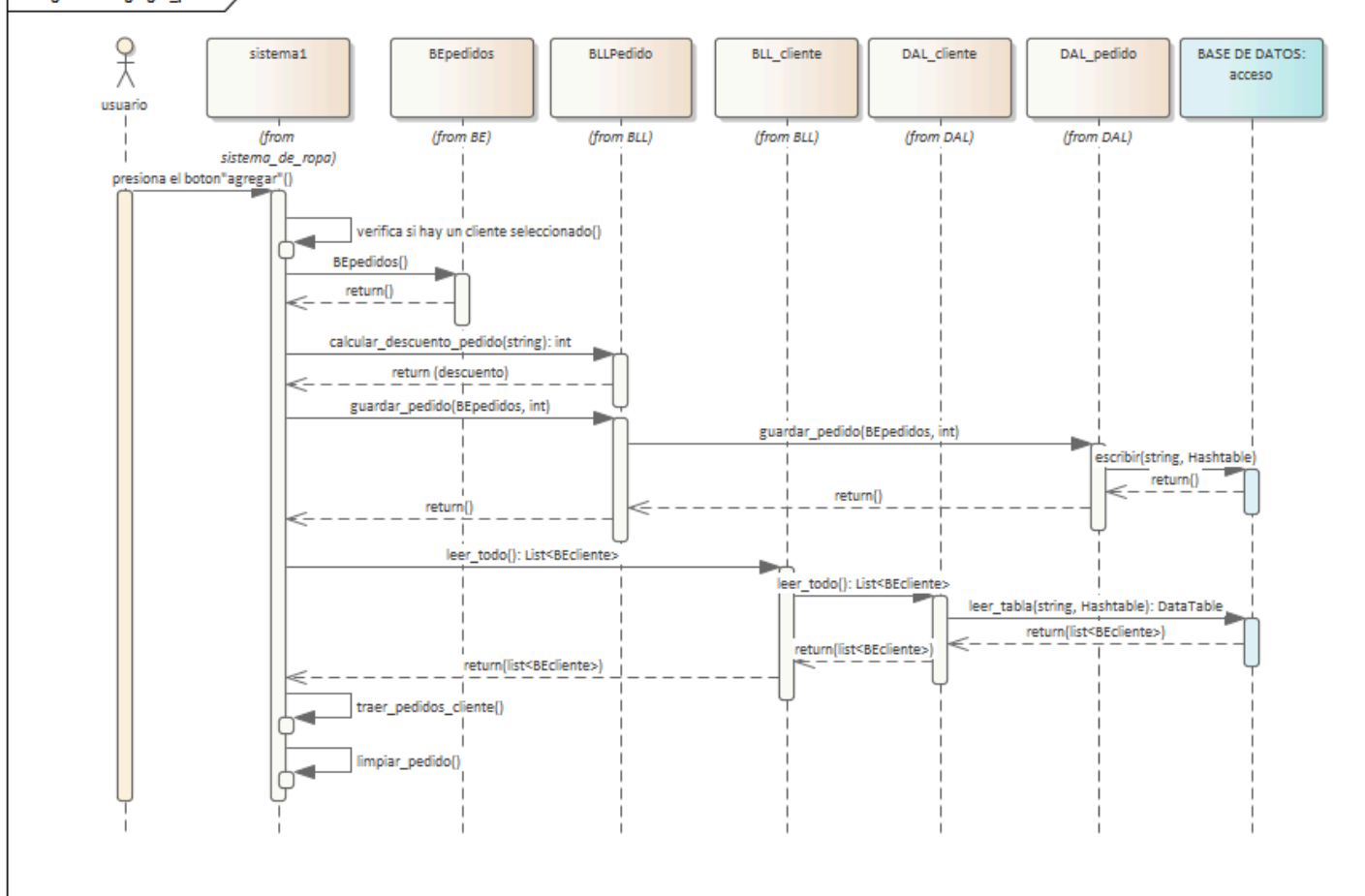
Nombre del Caso de uso: CU 018-Agregar Pedido
Actores: usuario
Objetivo: Permitir que el usuario pueda agregar un o varios pedidos por cada cliente y aplicarles un descuento especial según el tipo de cliente.

<p>UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p>Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

<p>Precondiciones: El usuario inició sesión previamente.</p> <p>El usuario posee permisos para agregar pedidos</p> <p>El usuario selecciona un cliente.</p>
<p>Postcondiciones:</p> <p>Se agrega un pedido relacionado a un cliente</p>
<p>Eventos Disparadores: El usuario presiona el botón “agregar pedido” después de seleccionar un cliente.</p>
<p>Puntos de extensión: Ninguno</p>
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El usuario presiona el botón “agregar pedido” 2. El sistema verifica que haya un cliente seleccionado. 3. El sistema calcula el descuento del pedido. 4. El sistema guarda el pedido en la base de datos. 5. El sistema actualiza la lista de clientes en el sistema y sus pedidos.
<p>Flujo Alternativo</p> <p>-Cliente no seleccionado.</p> <p>2A. El sistema detecta que no existe un cliente seleccionado.</p> <p>3A. El sistema envía un mensaje de error al usuario y vuelve al inicio del proceso.</p>

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

sd diagrama de agregar_pedido

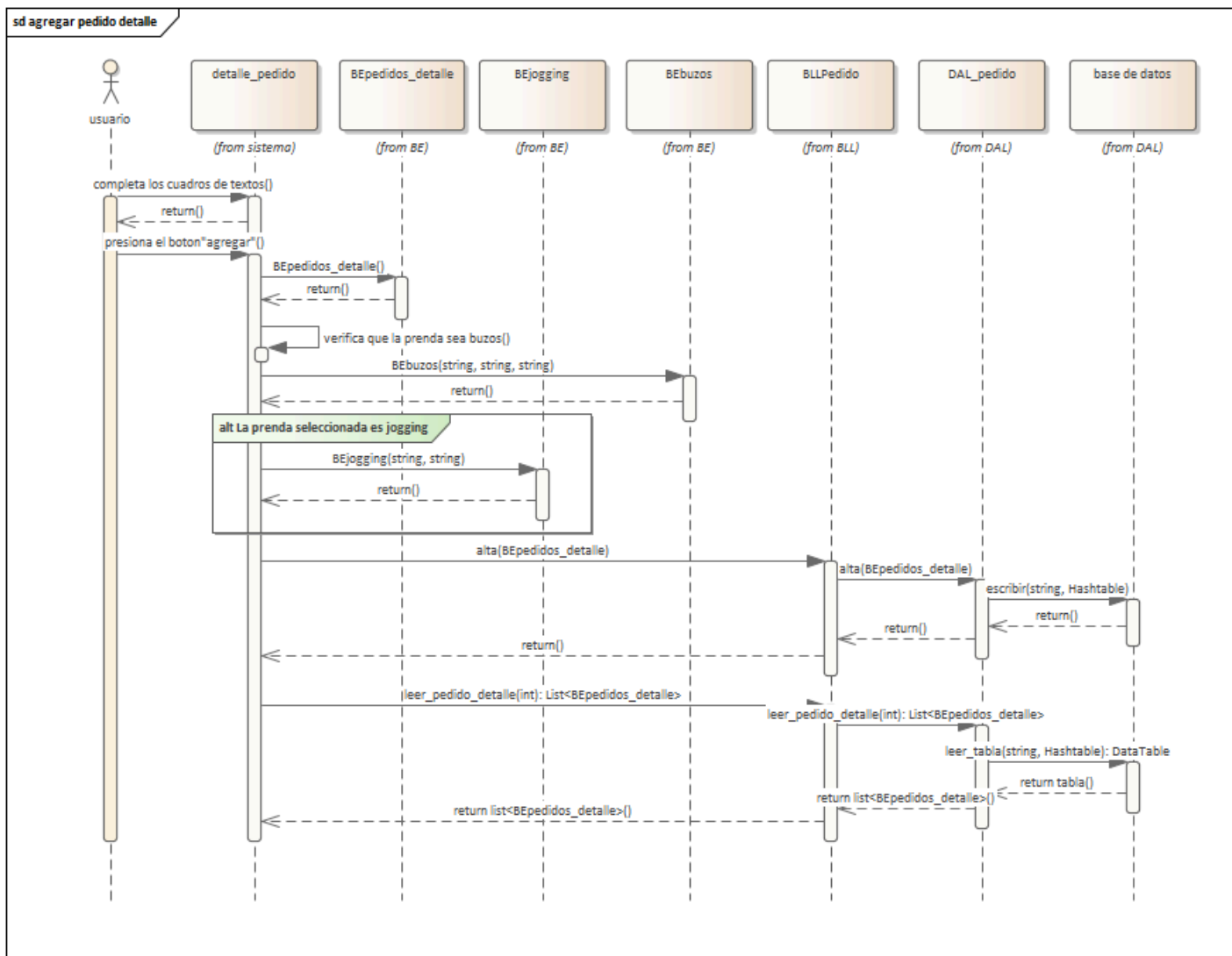


Nombre del Caso de uso: CU019a-Agregar pedido detalle
Actores: Usuario
Objetivo: Permitir que el usuario pueda agregar detalles al pedido del cliente
Precondiciones: El usuario inició sesión previamente.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

El usuario posee permisos para agregar pedido detalle
Postcondiciones:
Se carga en la base de datos el nuevo pedido detalle
Eventos Disparadores: El usuario completa los cuadros de textos y presiona el botón “agregar”
Puntos de extensión: Ninguno
Flujo Principal <ol style="list-style-type: none"> 1. El usuario completa los cuadros y combobox del pedido detalle. 2. El usuario presiona el botón “agregar”. 3. El sistema verifica que el tipo de prenda sea buzos. 4. El sistema guarda el pedido detalle en la base de datos. 5. El sistema muestra la lista actualizada de los pedidos detalles
Flujo Alternativo <p>-El tipo de prenda es Jogging.</p> <p>3A.El sistema verifica que el tipo de prenda sea buzos.</p> <p>4A. El sistema guarda el pedido detalle en la base de datos.</p> <p>5A. El sistema muestra la lista actualizada de los pedidos detalles</p>

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
	Versión: 1.0			



Nombre del Caso de uso: CU019b-Modificar pedido detalle
Actores: Usuario

<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Objetivo: Permitir que el usuario pueda modificar detalles al pedido del cliente
Precondiciones: El usuario inició sesión previamente. El usuario posee permisos para modificar pedido detalle
Postcondiciones: Se carga en la base de datos el pedido detalle modificado
Eventos Disparadores: El usuario selecciona un pedido detalle, completa los cuadros de textos y presiona el botón “modificar”
Puntos de extensión: Ninguno
Flujo Principal <ol style="list-style-type: none"> 1. El usuario selecciona un pedido detalle. 2. El usuario completa los cuadros y combobox del pedido detalle. 3. El usuario presiona el botón “modificar”. 4. El sistema verifica que el tipo de prenda seleccionada sea buzos 5. El sistema verifica que el tipo de prenda del nodo seleccionado y el pedido detalle que va a tomar su lugar sea también buzo 6. El sistema guarda la modificación del pedido detalle en la base de datos. 7. El sistema muestra la lista actualizada de los pedidos detalles
Flujo Alternativo <p>-Pedido detalle no seleccionado</p> <p>4A. El sistema detecta que no hay un pedido detalle seleccionado.</p> <p>5A El sistema le manda al usuario un mensaje de error.</p> <p>-El tipo de prenda no coincide</p> <p>5A. El sistema detecta que el tipo de prenda del pedido detalle seleccionado y el nuevo para modificar no son los mismos</p> <p>6A El sistema le manda al usuario un mensaje de error.</p>

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

-La prenda seleccionada es jogging.

4A. El sistema verifica que el tipo de prenda seleccionada sea buzos

5A. El sistema verifica que el tipo de prenda del nodo seleccionado y el pedido detalle que va a tomar su lugar sea también buzo

6A. El sistema guarda la modificación del pedido detalle en la base de datos.

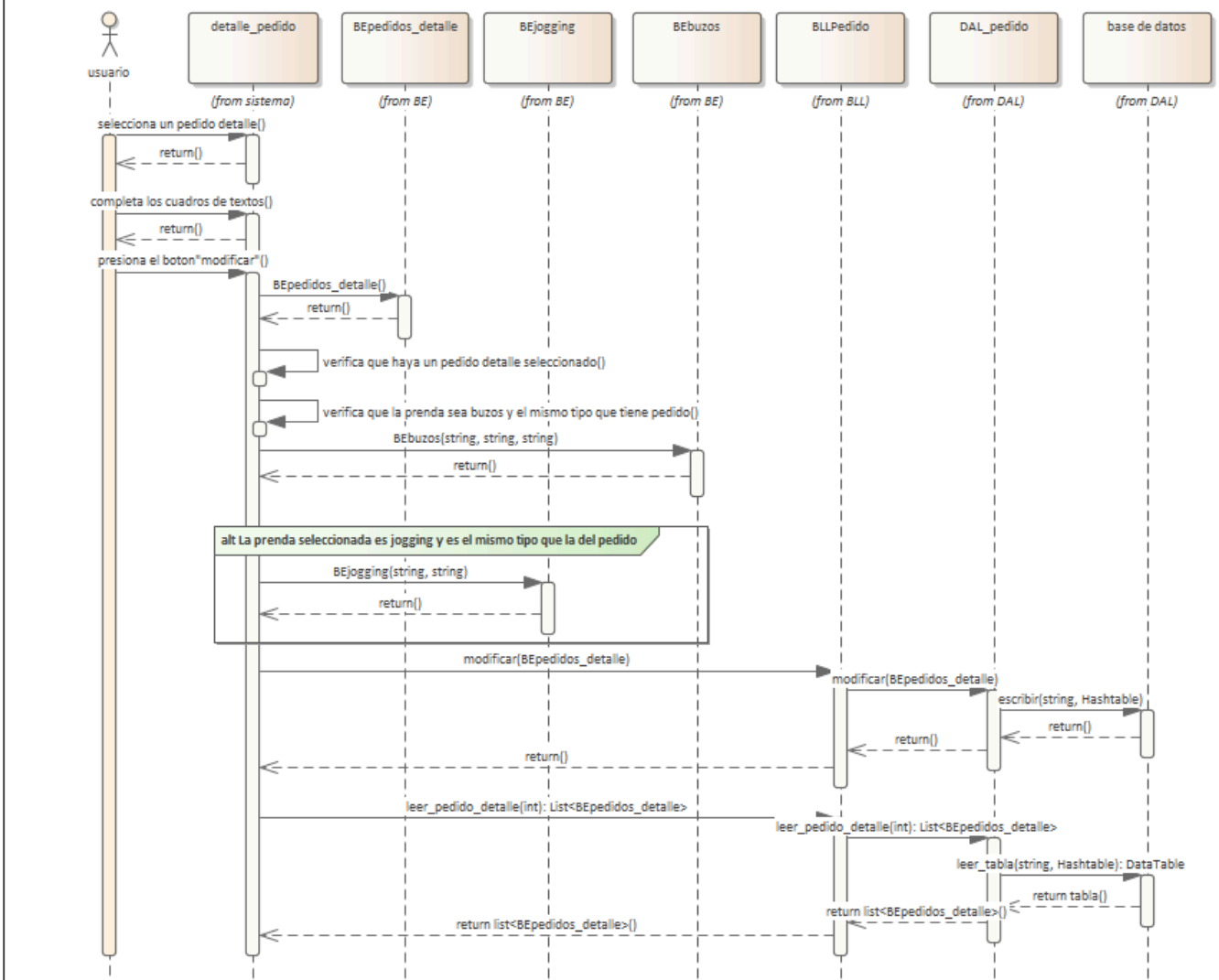
7A. El sistema muestra la lista actualizada de los pedidos detalles

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				
	<Nombre de Sección / Etapa>				Versión: 1.0

sd modificar pedido detalle

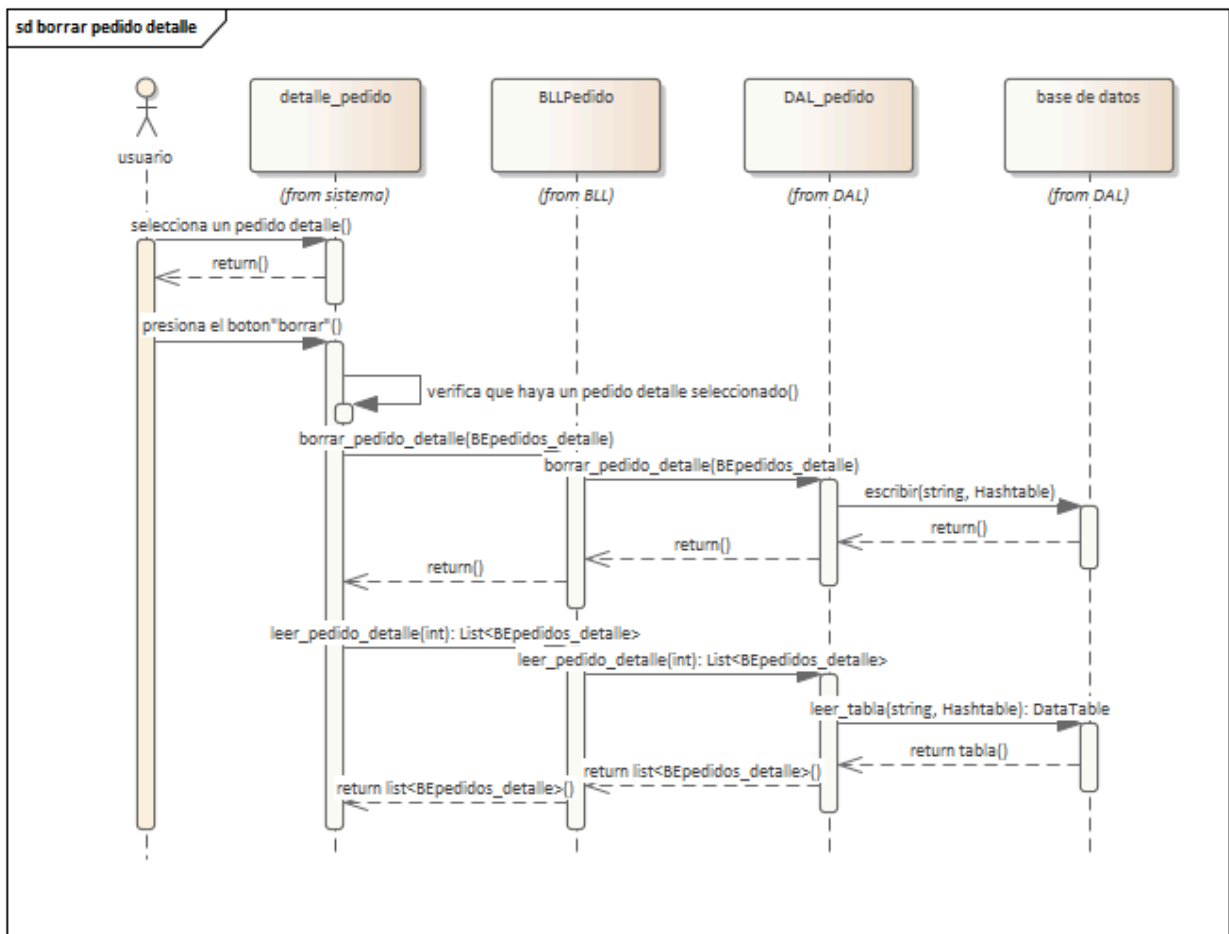


Nombre del Caso de uso: CU019b-Borrar pedido detalle

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Actores: Usuario
Objetivo: Permitir que el usuario pueda borrar detalles al pedido del cliente
Precondiciones: El usuario inició sesión previamente. El usuario posee permisos para borrar pedido detalle
Postcondiciones: Se borra el pedido detalle en la base de datos
Eventos Disparadores: El usuario selecciona un pedido detalle y presiona el botón "borrar"
Puntos de extensión: Ninguno
Flujo Principal <ol style="list-style-type: none"> 1. El usuario selecciona un pedido detalle. 2. El usuario presiona el botón "borrar". 3. El sistema verifica que haya un pedido detalle seleccionado. 4. El sistema elimina el pedido detalle de la base de datos. 5. El sistema muestra la lista actualizada de los pedidos detalles
Flujo Alternativo -Pedido detalle no seleccionado 3A. El sistema detecta que no hay un pedido detalle seleccionado. 4A El sistema le manda al usuario un mensaje de error.

<p>UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p>Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



Nombre del Caso de uso: CU 020-Borrar Pedido
Actores: usuario
Objetivo: Permitir que el usuario pueda borrar un pedido de un cliente
Precondiciones: El usuario inició sesión previamente.

<p style="text-align: center;">UNIVERSIDAD ABIERTA INTERAMERICANA</p> <p style="text-align: center;">Facultad de Tecnología Informática</p>				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

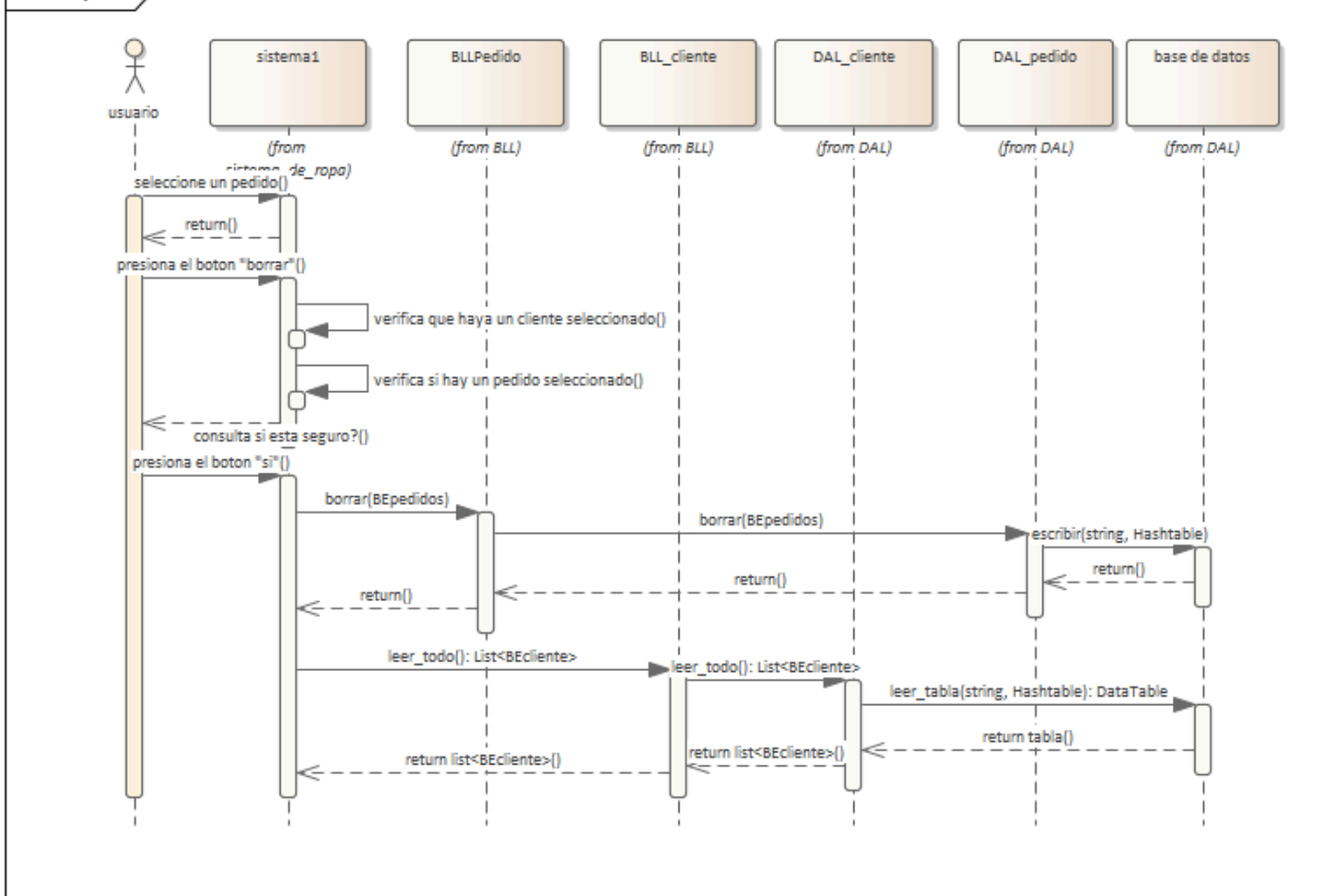
<p>El usuario posee permisos para borrar pedidos</p> <p>El usuario selecciona un cliente.</p>
<p>Postcondiciones:</p> <p>Se borra un pedido relacionado al cliente seleccionado</p>
<p>Eventos Disparadores: El usuario presiona el botón “borrar pedido” después de seleccionar un cliente.</p>
<p>Puntos de extensión: Ninguno</p>
<p>Flujo Principal</p> <ol style="list-style-type: none"> 1. El usuario selecciona un pedido. 2. El usuario presiona el botón “borrar pedido” 3. El sistema verifica que haya un cliente seleccionado. 4. El sistema verifica que haya un pedido seleccionado. 5. El sistema le manda al usuario un diálogo para verificar que está seguro 6. El usuario presiona “sí”. 7. El sistema borra el pedido de la base de datos. 8. El sistema actualiza la lista de clientes en el sistema y sus pedidos.
<p>Flujo Alternativo</p> <p>-Cliente no seleccionado.</p> <p>3A. El sistema detecta que no existe un cliente seleccionado.</p> <p>4A. El sistema envía un mensaje de error al usuario y vuelve al inicio del proceso.</p> <p>-Pedido no seleccionado.</p> <p>4B. El sistema detecta que no existe un pedido seleccionado.</p> <p>5B. El sistema envía un mensaje de error al usuario y vuelve al inicio del proceso.</p> <p>-El usuario presiona el botón “NO”</p> <p>6C. El usuario presiona el botón “NO”.</p> <p>7C El proceso termina</p>

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

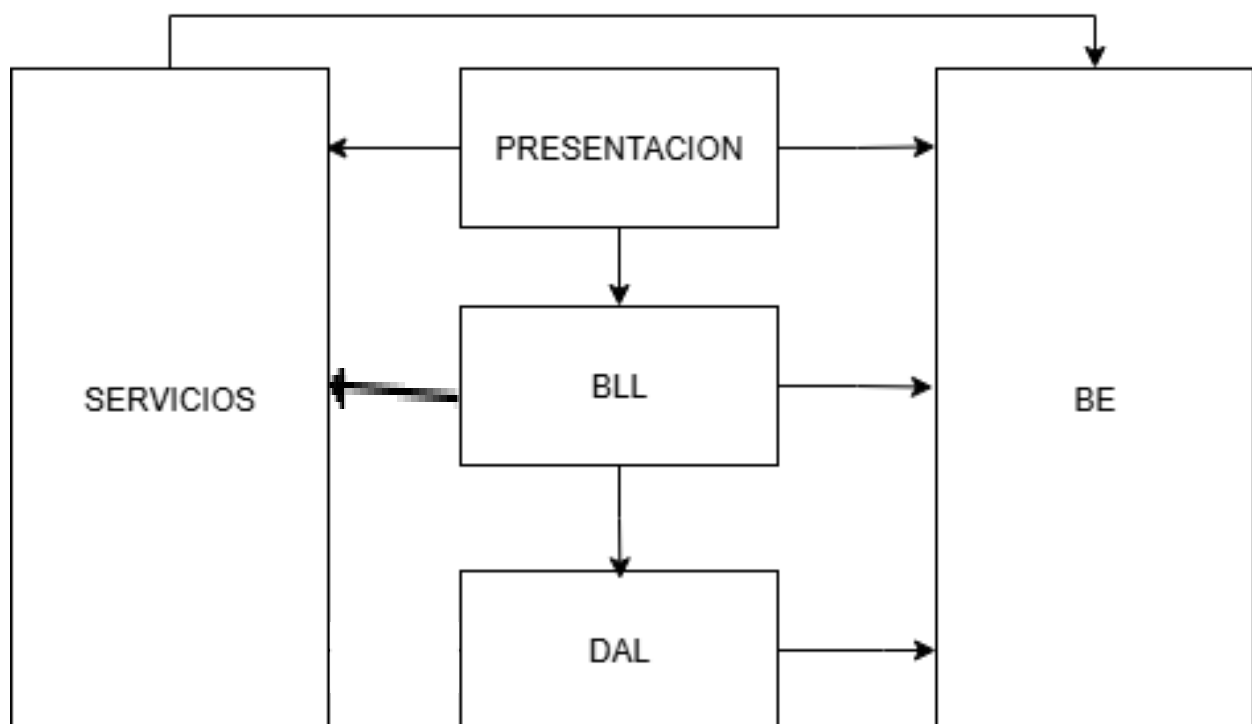
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				
	<Nombre de Sección / Etapa>				Versión: 1.0

sd borrar pedido



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Fecha: 15/4/25
	Legajo:			
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
<Nombre de Sección / Etapa>				Versión: 1.0

T01. Arquitectura Base



Descripción

El sistema está desarrollado bajo un modelo de cinco capas, elegido por su claridad, orden y facilidad de mantenimiento. Las capas utilizadas son: Presentación (UI), Servicios, BLL, DAL y BE.

Este modelo fue seleccionado porque:

- Resulta adecuado para el tamaño y la complejidad del sistema; no fue necesario agregar más capas como mappers u otros componentes.
- Permite organizar el código de forma intuitiva: UI para formularios, BE para entidades, DAL para base de datos, BLL para reglas de negocio y Servicios para funcionalidades específicas.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Descripción de cada capa

1. Capa Presentación (UI)

Es la capa donde se encuentran todos los formularios y pantallas del sistema.

Esta capa solo accede a: Servicios, BLL y BE

Su función es mostrar la interfaz al usuario y enviar las acciones hacia las capas inferiores.

2. Capa BLL

Contiene la lógica de negocio del sistema.

Aquí se implementan las funciones y procesos necesarios para que las entidades funcionen correctamente.

La BLL puede acceder a: Servicios, DAL y BE

3. Capa DAL

Es la capa encargada del acceso a la base de datos.

Incluye:

- Una clase Acceso, con métodos genéricos y reutilizables para conectarse a la base.
- Una clase DAL por cada entidad, donde se implementan todos los métodos necesarios para obtener, insertar, modificar y eliminar datos.

Estas clases utilizan internamente la clase Acceso.

La DAL solo puede ver:

- BE

4. Capa BE

Es la capa que posee las propiedades y constructores de las clases

Esta capa no ve a ninguna otra, ya que debe estar completamente aislada.

5. Capa Servicios

Incluye servicios generales que pueden ser utilizados por varias capas.

Contiene:

- El servicio de idiomas, implementado con Observer.
- El servicio de encriptación, tanto Hash como encriptación simétrica.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

- Implementaciones de patrones comunes (por ejemplo, Singleton).

La capa Servicios puede ver: DAL y BE

T02. Gestión de Log In / Log Out del Sistema

Descripción

El sistema incorpora una funcionalidad de **inicio y cierre de sesión** que controla el acceso de los usuarios mediante autenticación.

Esta funcionalidad está implementada en la **capa Servicios**, a través de una clase que utiliza el **patrón Singleton**, lo que garantiza que exista una única instancia activa que represente la sesión del usuario durante toda la ejecución del sistema.

Inicio de Sesión

Cuando un usuario ingresa sus credenciales, la capa de Servicios valida los datos y, si son correctos, la instancia Singleton se inicializa cargando en memoria:

- Los **datos del usuario**
- Sus **roles y permisos asociados**
- Y su **estado de sesión activa**.

Este proceso es indispensable para habilitar el acceso al sistema, ya que los permisos del usuario se utilizan luego para habilitar o restringir las distintas funcionalidades disponibles en la interfaz.

Además, cada inicio de sesión queda registrado en la **bitácora**, donde se almacena el usuario, la fecha y la acción realizada.

Cierre de Sesión

Cuando el usuario selecciona la opción de cerrar sesión, la misma instancia Singleton es limpiada, eliminando de memoria los datos del usuario autenticado y retornando el sistema a un estado seguro.

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

El cierre de sesión también queda registrado en la bitácora con fecha, usuario y acción correspondiente.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			Versión: 1.0
	<Nombre de Sección / Etapa>			

Diagrama de secuencia y especificación (Log in)

UNIVERSIDAD ABIERTA INTERAMERICANA

Facultad de Tecnología Informática

Materia: Ingeniería de software

Docente: Julián Martín Rodríguez

Fecha: 15/4/25

Alumno: Tomas Meneces Fernandez Benjamin

Legajo:

Localización: Castelar

Comisión: 2-A

Turno: Noche

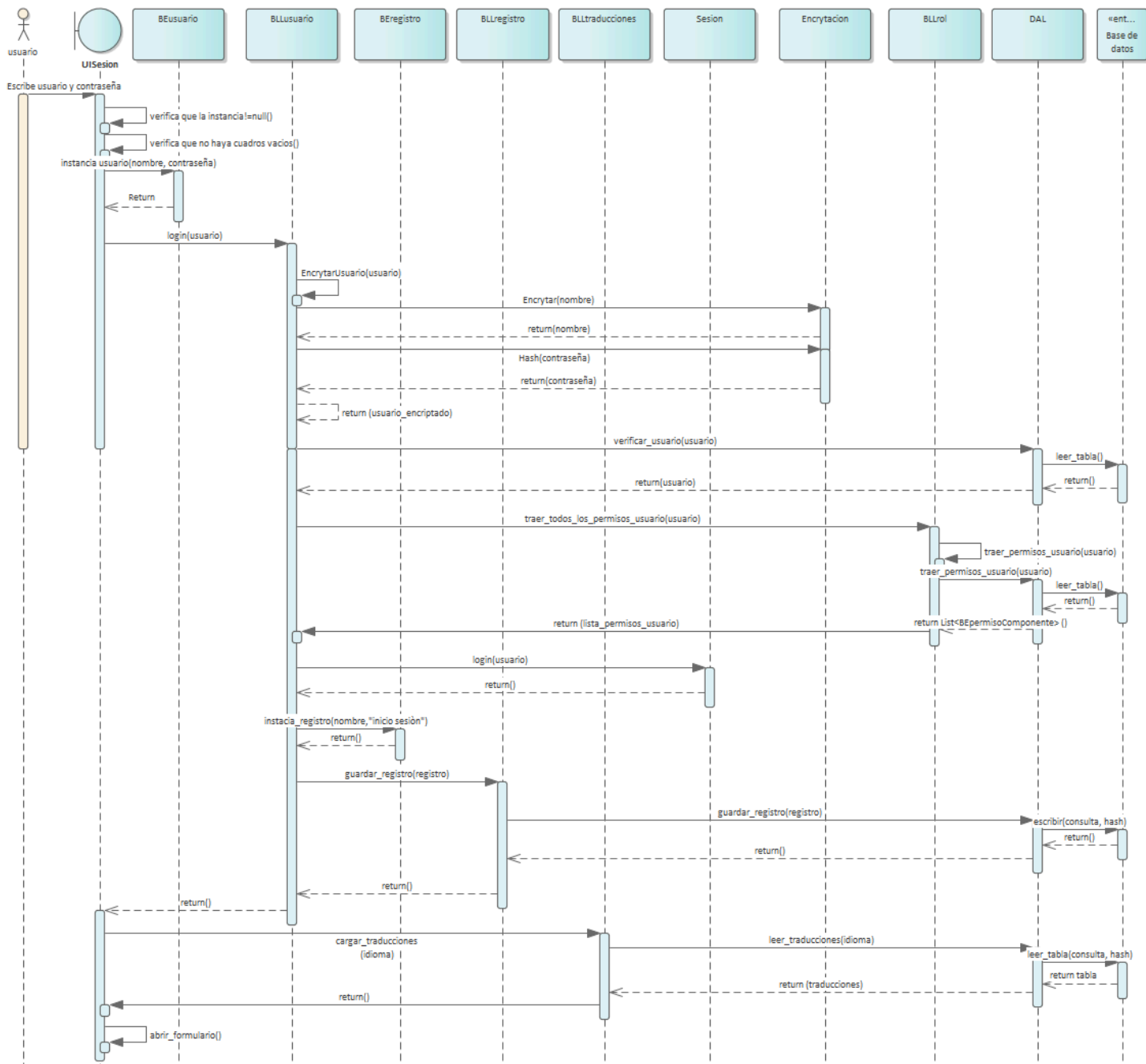
Año: 3

sistema de ropa

Versión: 1.0

<Nombre de Sección / Etapa>

sd diagramas de inicio sesion



UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Nombre del Caso de uso: CU 000-Inicio de Sesión
Actores: Usuario, Administrador y Gerente
Objetivo: Permitir que el usuario ingrese sus credenciales y acceder al sistema validando su identidad y cargando sus permisos.
Precondiciones: El usuario debe estar registrado en el sistema.
Postcondiciones: La sesión del usuario queda iniciada. Se cargan los permisos asignados al usuario. Se registra la acción en la bitácora.
Eventos Disparadores: El usuario introduce sus credenciales y selecciona la opción “Ingresar”
Puntos de extensión: Ninguno

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Flujo Principal:

1. El usuario ingresa sus credenciales.
2. El usuario presiona el botón “iniciar sesión”
3. El sistema verifica que no exista una instancia usuario.
4. El sistema verifica que no haya cuadros de textos vacíos.
5. El sistema encripta el usuario.
6. El sistema verifica que el usuario exista en la base de datos.
7. El sistema trae todos los permisos del usuario
8. El sistema inicia sesión en el sistema con el usuario
9. El sistema guarda un registro en la bitácora.
10. El sistema carga las traducciones en el sistema según el idioma
11. El sistema abre el formulario “menu”

Flujo Alternativo:

-existe una instancia del usuario.

3A. El sistema detecta que la instancia del usuario existe.

4A. El sistema le manda un mensaje de error al usuario.

-hay cuadros vacios

4A El sistema detecta que hay cuadros vacíos.

5A. El sistema le manda un mensaje de error al usuario.

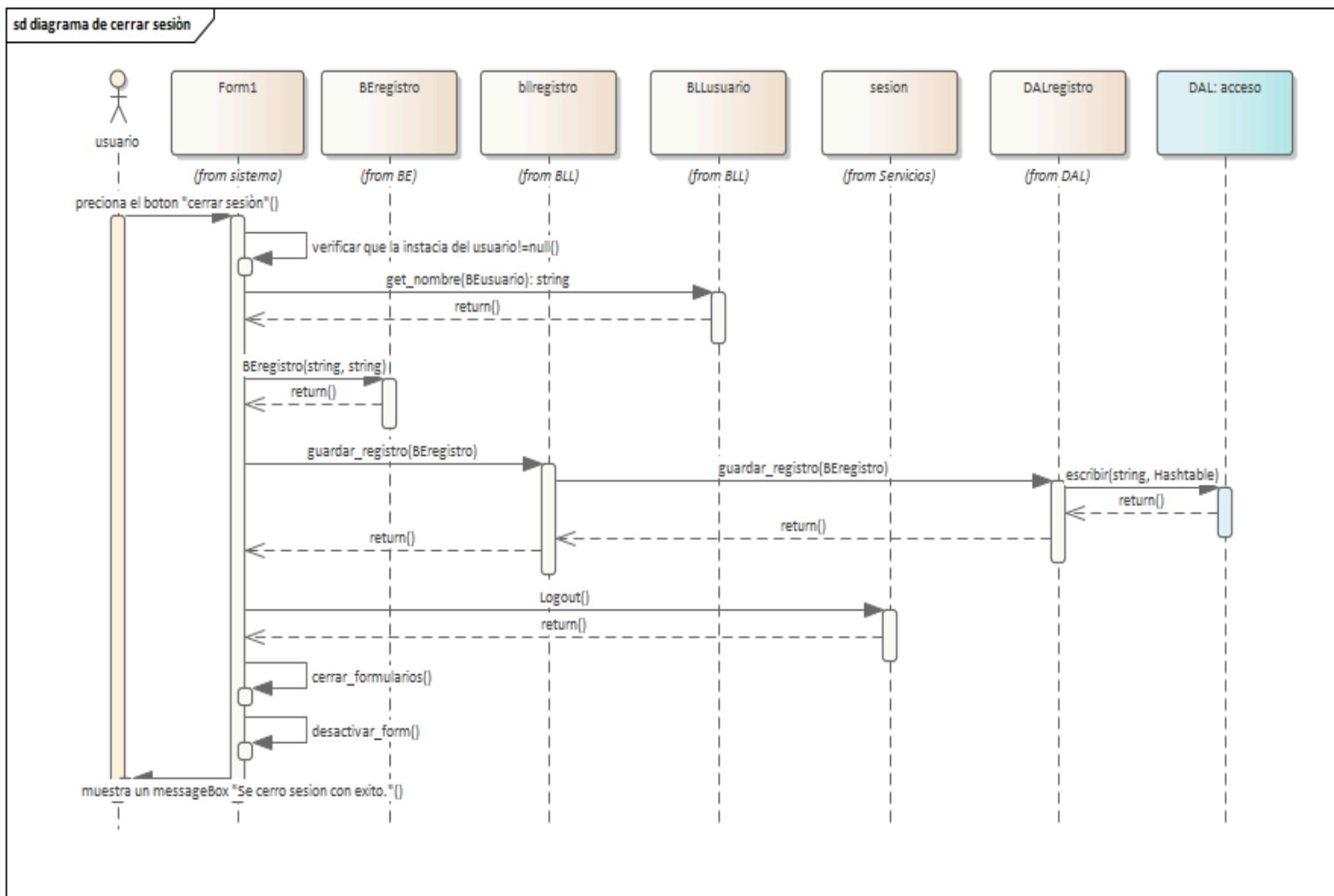
-no existe un usuario con esas credenciales.

6A. El sistema detecta que el usuario encriptado no existe en la base de datos

7A. El sistema le manda un mensaje de error al usuario.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Diagrama de secuencia y especificación (Log out)



Nombre del Caso de uso: CU 001-Cerrar Sesión

Actores: Usuario, Administrador y Gerente

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Objetivo: Permitir que el usuario cierre su sesión activa en el sistema de forma segura, registrando la acción en la bitácora y liberando la instancia actual de sesión.
Precondiciones: El usuario debe tener una sesión iniciada.
<p>Postcondiciones:</p> <p>La sesión del usuario queda cerrada.</p> <p>Se registra la acción “Cerrar sesión” en la bitácora.</p> <p>El sistema solo permite el inicio de sesión en el formulario.</p>
Eventos Disparadores: El usuario presiona el botón “Cerrar sesión” en el formulario principal.
Puntos de extensión: Ninguno
<p>Flujo Principal:</p> <ol style="list-style-type: none"> 1. El usuario hace clic en el botón “Cerrar sesión” desde la pantalla principal. 2. El sistema verifica que exista una sesión activa (instancia del Singleton no nula). 3. El sistema obtiene el nombre del usuario logueado para registrarlo en la bitácora. 4. Se crea un nuevo registro de bitácora con la acción “Cerrar sesión” y la fecha actual. 5. El sistema guarda el registro de la bitácora en la base de datos. 6. El sistema ejecuta la función Log Out() del Singleton de Sesión, eliminando la instancia activa. 7. El sistema cierra todos los formularios abiertos. 8. El sistema vuelve a mostrar el formulario de inicio de sesión. 9. Se muestra un mensaje confirmando: “Se cerró sesión con éxito.”

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Flujo Alternativo:

3.a Si alguno de los campos (usuario/contraseña) está vacío:

- El sistema muestra un mensaje indicando que los campos deben completarse.
- El proceso finaliza sin iniciar sesión..

6.a Si las credenciales no coinciden con ningún usuario registrado:

- El sistema muestra el mensaje *"El usuario o contraseña incorrecta, vuelva a intentarlo."*
- El proceso finaliza sin iniciar sesión.

10.a Si el sistema detecta un problema con los dígitos verificadores del usuario:

- Se bloquea el sistema entero.
- Se informa al usuario que hubo un error de integridad en los datos.

T03. Gestión de Encriptado

Descripción

La encriptación del sistema se encuentra desarrollada dentro de la **capa Servicios**, y combina dos mecanismos distintos según la necesidad.

UNIVERSIDAD ABIERTA INTERAMERICANA Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Servicios::encryptacion	
-	<u>key: string = "kjfsdhjklfdshjkl" {readOnly}</u>
-	<u>iv: string = "jdkshfhjkhfsdajk" {readOnly}</u>
+	<u>Hash(string): string</u>
+	<u>encriptar(string): string</u>
+	<u>desencriptar(string): string</u>

Hash (encriptación no reversible – MD5)

El sistema utiliza un método de **Hash MD5**, el cual es un tipo de encriptación **no reversible**.

Este mecanismo se aplica principalmente a información que no debe descifrarse nuevamente, como por ejemplo:

- **Las contraseñas de los usuarios**, que se almacenan únicamente en formato hash para evitar que puedan recuperarse en texto plano.
- **El cálculo del dígito verificador**, que utiliza también este Hash como parte del proceso de generación del código de integridad.

Al ser un método no reversible, garantiza que estos datos no puedan reconstruirse aun cuando alguien acceda a la base de datos.

Encriptación simétrica (reversible)

El sistema implementa un mecanismo de **encriptación simétrica**, el cual permite **encriptar y desencriptar** información según sea necesario.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

La encriptación simétrica permite asegurar la confidencialidad de los datos sensibles manteniendo la posibilidad de leerlos cuando el sistema lo requiera.

ejemplo:

- **Los nombres de los usuarios**, que se almacenan en la base de datos en forma encriptada y se desencripta al momento de mostrarse en el sistema.

T04. Gestión de Perfiles de Usuario

Descripción

La gestión de perfiles de usuario del sistema fue implementada utilizando el **Patrón Composite**, permitiendo manejar tanto permisos simples como permisos compuestos (roles).

Permisos simples

Los permisos simples representan acciones que permiten usar funciones dentro del sistema. Estos permisos son fijos y no pueden ser agregados ni eliminados. Los permisos simples definidos son:

- ABM Permisos
- Ver Bitácora
- Ver Reportes
- ABM Idioma
- ABM Usuario
- Personalizar Ropa
- ABM Clientes
- Entrar al Sistema

Permisos compuestos (Roles)

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Versión: 1.0

Los permisos compuestos funcionan como roles, y están formados por un conjunto de permisos simples y/o de otros permisos compuestos. El sistema permite agregar, modificar y eliminar roles.

Los roles implementados en el sistema son:

Usuario

Incluye:

- ABM Clientes
- Entrar al Sistema

Administrador

Incluye:

- ABM Idioma
- ABM Usuario
- Personalizar Ropa
- Además, incorpora todos los permisos del rol Usuario.

Gerente

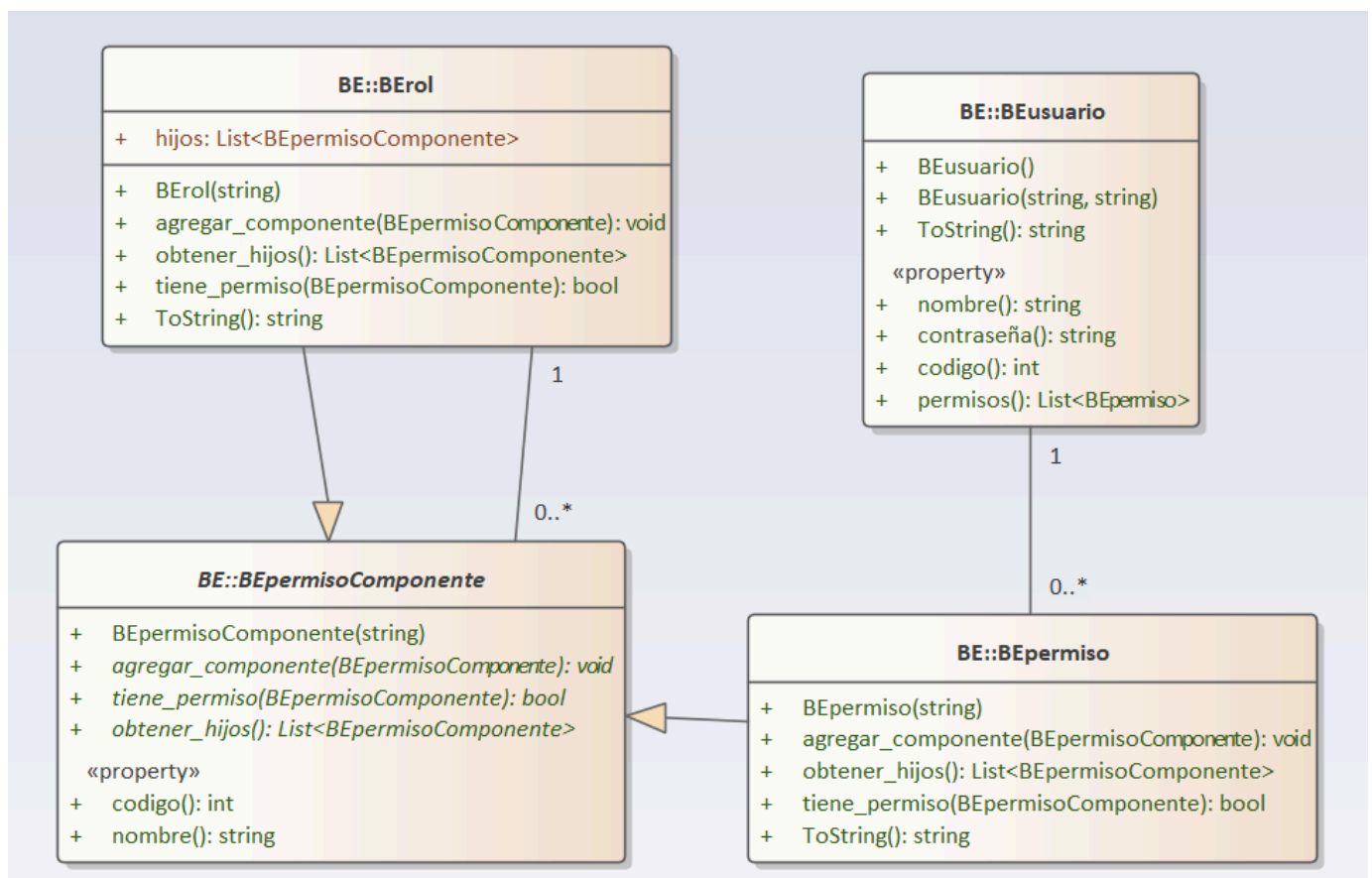
Incluye:

- Ver Bitácora
- ABM Permisos
- Ver Reportes
- Además, incorpora todos los permisos del rol Administrador y del rol Usuario.

Gracias al Patrón Composite, la verificación de permisos es recursiva: un usuario que posea un rol compuesto automáticamente hereda todos los permisos contenidos dentro de dicho rol, ya sean directos o indirectos.

Diagramas de clases:

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				



T05. Gestión de Múltiples Idiomas

Descripción

La gestión de idiomas del sistema se implementa utilizando el **patrón de diseño Observer**, permitiendo que todos los formularios actualicen automáticamente sus textos cuando el usuario selecciona un idioma diferente.

Estructura general

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

- **Interfaz IIdioma:** define el comportamiento básico de un idioma, el cual posee 2 funciones, que le permite guardar observadores y notificarles a esos observadores el cambio del idioma.
- **Clase Idioma:** implementa IIdioma e incluye una lista de observer(formularios que deben actualizarse).
- **Interfaz IObserverTraduccion:** implementada por todos los formularios del sistema.
 - Contiene el proceso ActualizarIdioma(), obligatoria para cada formulario.
- **Clase Traducciones:** gestiona las traducciones para cada idioma.

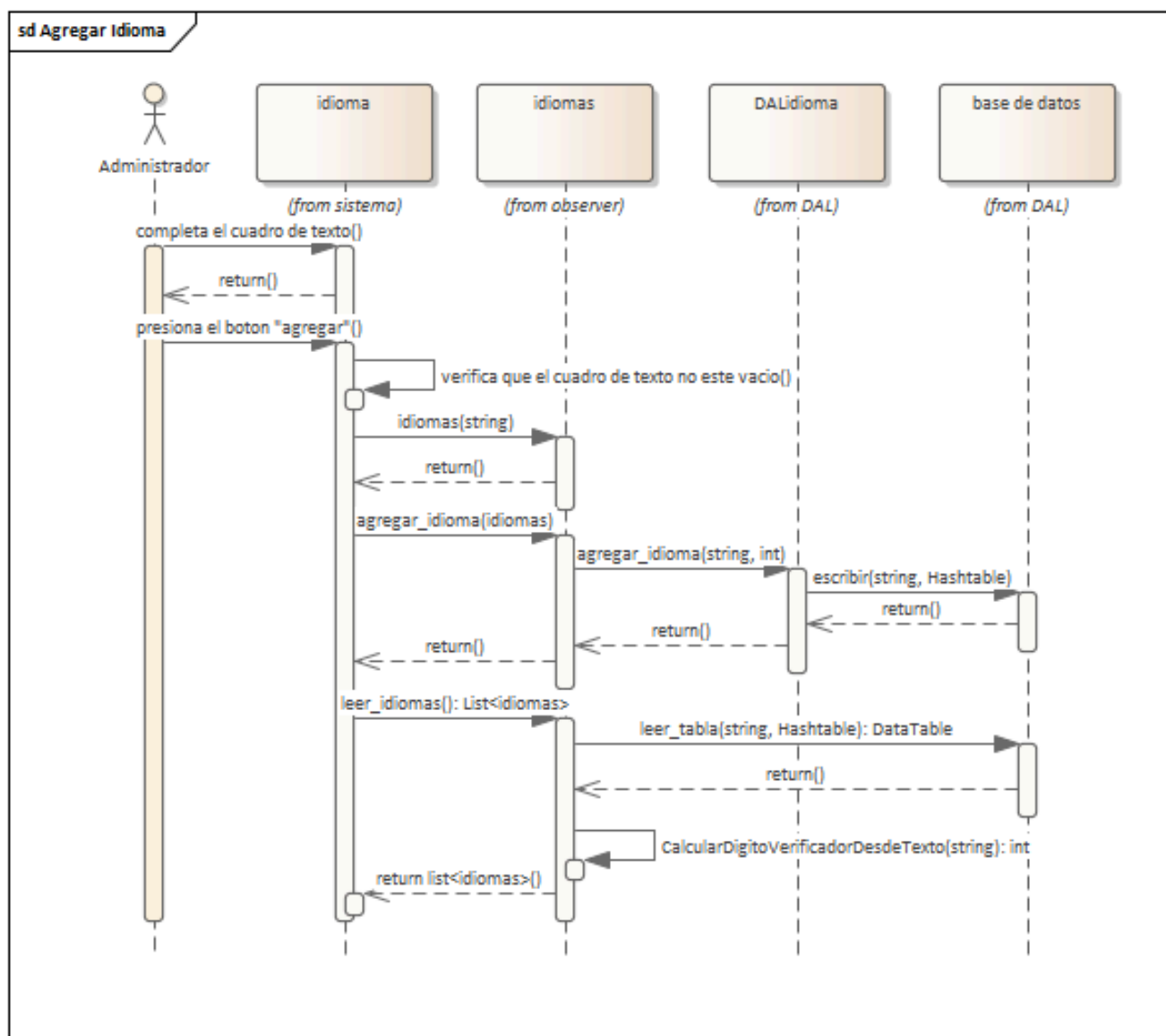
Funcionamiento

1. Cada formulario implementa la interfaz IObserverTraduccion, lo que lo obliga a definir el método ActualizarIdioma(), donde actualiza todos sus textos usando las traducciones del idioma seleccionado.
2. Al crear un formulario, el sistema lo agrega a la lista de observadores del idioma actual.
3. Cuando el usuario cambia de idioma, la clase Idioma ejecuta su método Notificar_a_todos(), que recorre la lista de observadores.
4. Cada observador recibe la notificación y ejecuta ActualizarIdioma(), actualizando automáticamente los textos visibles en pantalla.
5. Esto permite que todo el sistema cambie de idioma en tiempo real sin necesidad de reabrir formularios.

El sistema permite:

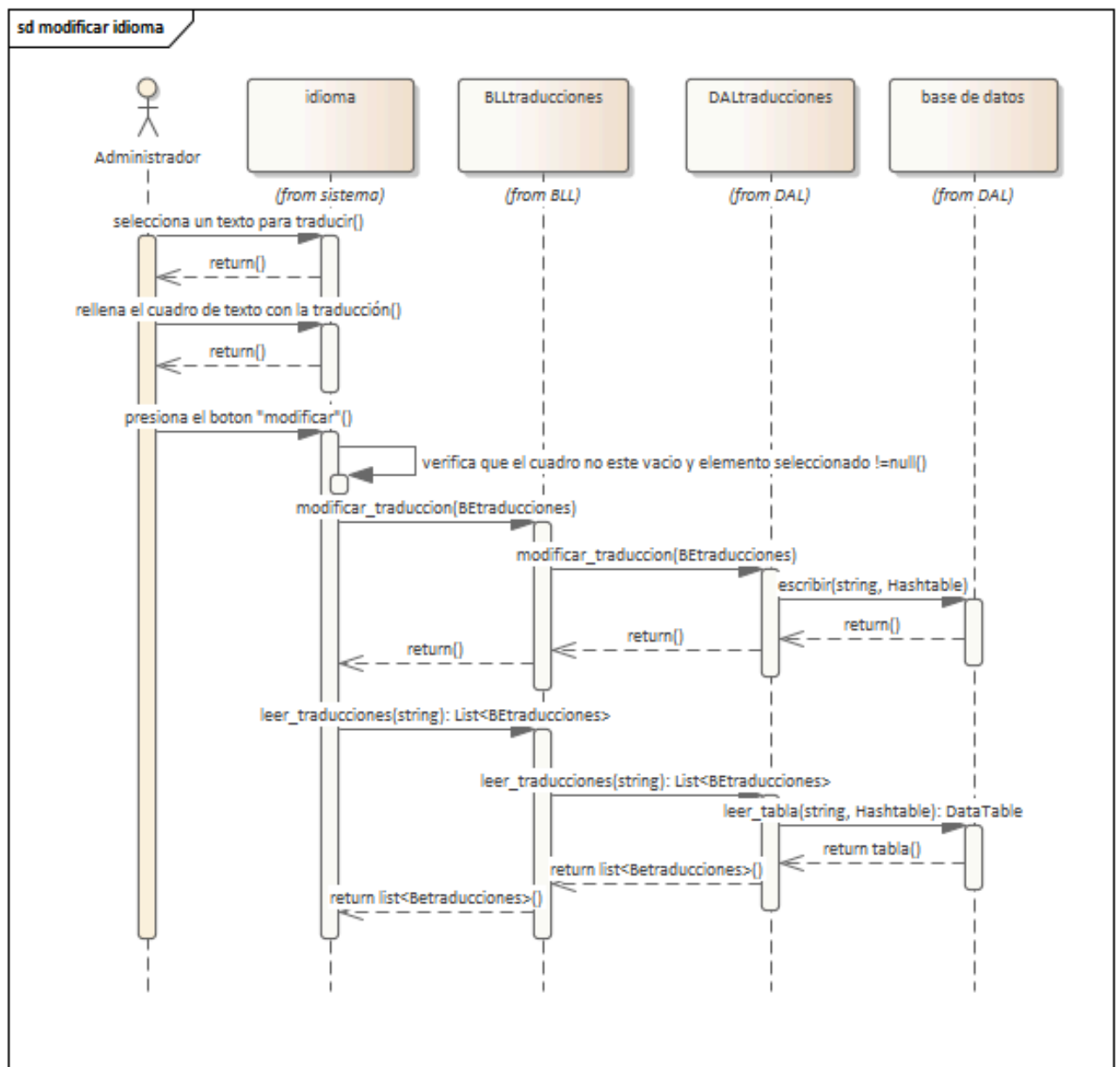
- **Agregar** nuevos idiomas

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



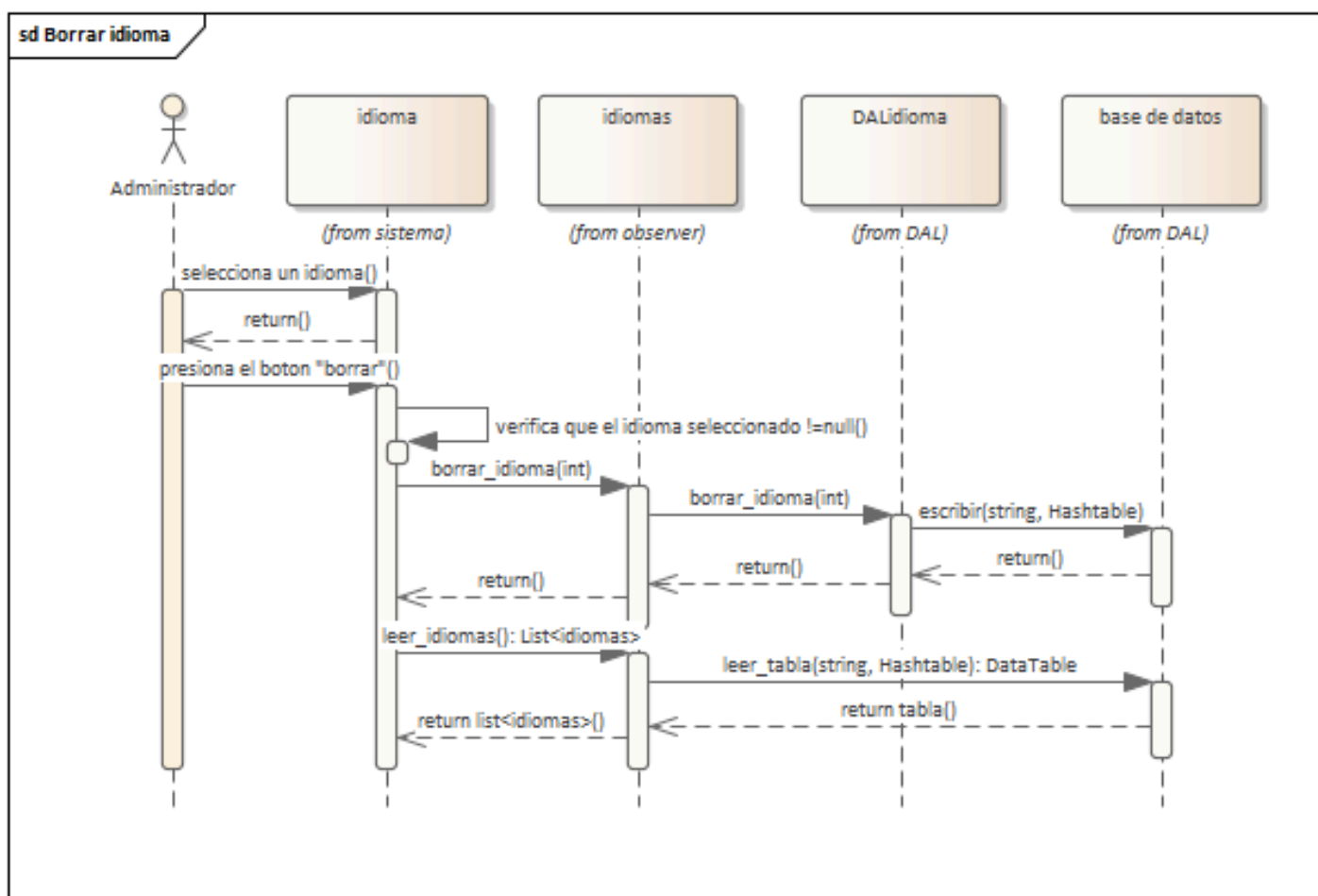
- **Modificar** las traducciones existentes

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



- **Eliminar idiomas**

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



T06a. Gestión de bitácora

Descripción general

La bitácora es un registro interno del sistema que almacena las acciones críticas realizadas por los usuarios. Su finalidad es permitir un control histórico sobre el inicio y cierre de sesión en el sistema.

Datos registrados

Cada entrada de la bitácora guarda la siguiente información:

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

- **Código de bitácora.**
- **Nombre del usuario** que realizó la acción.
- **Acción realizada** (Inicio de Sesión / Cierre de Sesión).
- **Fecha** en que ocurrió la acción.

Funcionamiento

Cada vez que un usuario inicia o cierra sesión, el sistema genera automáticamente un registro en la bitácora.

No puede ser modificada ni eliminada solo el sistema hace ese registro.

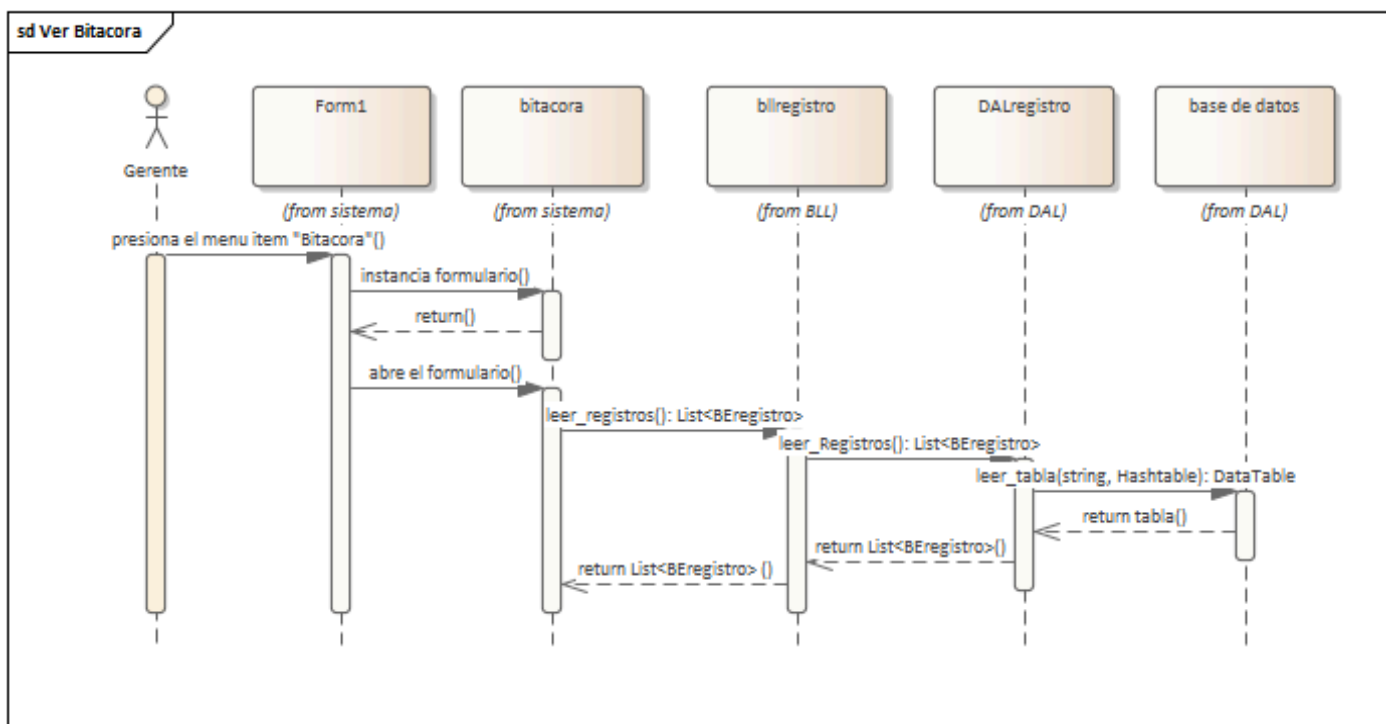
Acceso a la bitácora

Únicamente los usuarios con **rol Gerente** tienen acceso al módulo “Bitácora”.

Dentro de este módulo pueden realizar:

- Búsqueda por **nombre de usuario**
- Búsqueda por **fechas**

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa				
	<Nombre de Sección / Etapa>				Versión: 1.0



T06b. Control de cambios

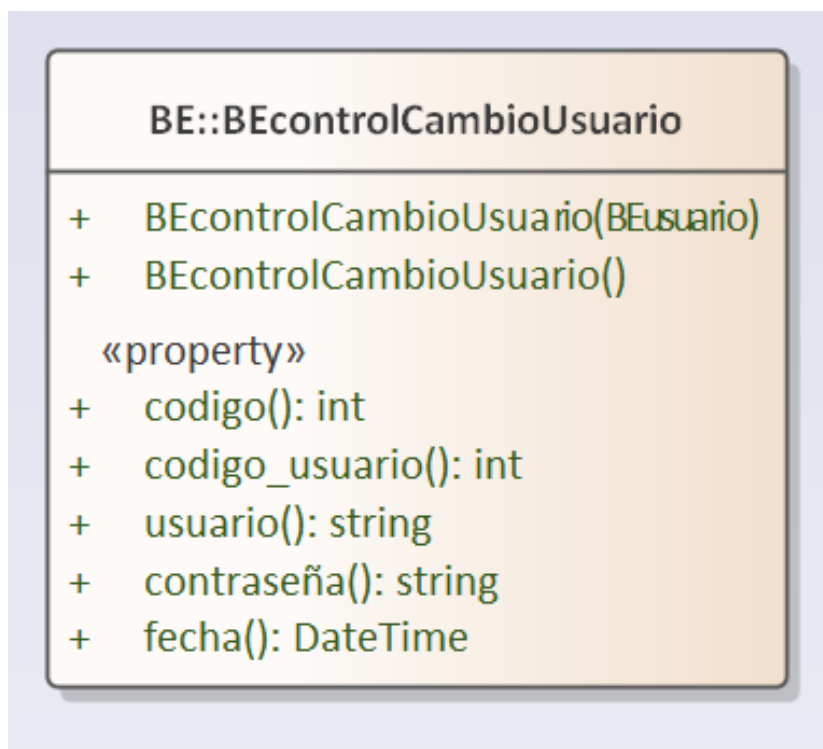
Objetivo

Registrar todas las modificaciones realizadas sobre los usuarios para permitir al que tenga permisos de administrador restaurar una versión anterior de ser necesario.

Alcance

Se aplica únicamente al módulo de Usuarios. Solo los usuarios con permisos de administrador pueden ver y usar el Control de Cambios.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



Datos registrados

Cada cambio guarda:

- Código del registro
- Código del usuario afectado
- Nombre del usuario
- Contraseña del usuario
- Fecha

Funciones

- Creación: al crear un usuario se registra la versión inicial.
- Modificación: después de que modifica un usuario, paso siguiente el sistema guarda el registro

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

- Borrado: al eliminar un usuario, se elimina también todo su historial de control de cambios
- Restauración: el administrador puede elegir una versión anterior de un usuario y restaurarla.

T07. Gestión de Dígitos Verificadores

El sistema implementa un mecanismo de Dígito Verificador (DV) para garantizar la integridad de los datos del módulo de Idiomas, evitando modificaciones directas en la base de datos que el sistema no pueda detectar.

Método utilizado

Para calcular el dígito verificador, el sistema aplica la siguiente lógica:

1. Entrada: recibe el nombre del idioma (por ejemplo, "Español" o "Inglés").
2. Hash: el texto se transforma mediante una función Hash MD5, generando una cadena encriptada.
3. Recorrido de caracteres: se recorre cada carácter del texto encriptado.
4. Operación por carácter:

Se incrementa un contador "v" para cada letra.

Se suma a la variable "suma" la multiplicación entre el carácter y el contador.

5. Cálculo final:

La suma total se divide usando el módulo 197317.

El resto resultante es el Dígito Verificador.

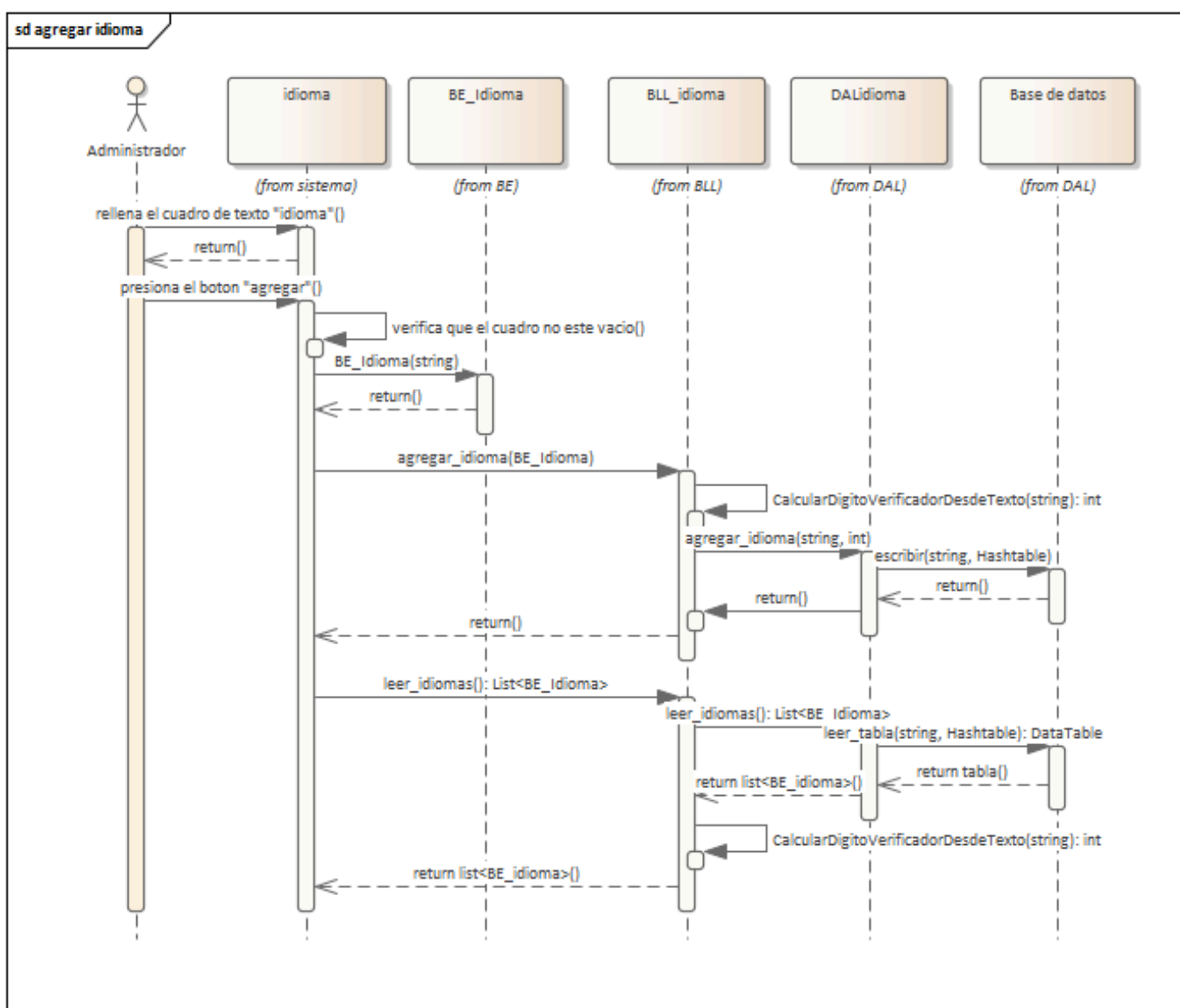
UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

```

52 public static int CalcularDigitoVerificadorDesdeTexto(string texto)
53 {
54     texto = encryptacion.Hash(texto);
55     int suma = 0;
56     int v = 0;
57     foreach (char c in texto)
58     {
59         v++;
60         suma += c*v;
61     }
62     suma = suma % 197317;
63     return suma;
64 }

```

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0



Uso dentro del sistema

- Cada idioma almacenado en la base de datos posee un dígito verificador asociado.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

- Al iniciar el sistema, antes de mostrar la pantalla de login:
 1. Se cargan los idiomas desde la base de datos.
 2. Para cada idioma, se vuelve a calcular el DV.
 3. Se compara contra el DV guardado para confirmar que los datos no han sido modificados

Detección de manipulación

Si alguno de los idiomas fue alterado manualmente en la base de datos, los dígitos no coinciden.

En ese caso, el sistema:

- Bloquea el inicio de sesión.
- Muestra un mensaje de error indicando una posible manipulación externa y se cierra el programa.

Alcance

El dígito verificador está aplicado únicamente al módulo de Idiomas, ya que se considera información sensible para la traducción y funcionamiento global del programa.

D03. Material de apoyo al usuario final

MANUAL DE USUARIO – SISTEMA LOOK IDEAL

1. Introducción

El sistema “Look Ideal” es una aplicación destinada a la gestión completa de usuarios, clientes, pedidos, idiomas, permisos, bitácora, personalización de ropa y generación de reportes.

Este manual describe cada formulario, cada función y cada operación disponible para el usuario final.

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

2. Inicio de Sesión

 sesion
 —
□
×

USUARIO

IDIOMA

Español
 

CONTRASEÑA

INGRESAR

Al iniciar la aplicación se muestra el formulario de inicio de sesión.

El usuario debe:

1. Ingresar su **nombre de usuario**
2. Ingresar su **contraseña**
3. Seleccionar el **idioma** (Español, Inglés o el que esté disponible)
4. Presionar **Ingresar**

Si las credenciales son válidas, se accede al menú principal del sistema.

3. Menú Principal

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			Versión: 1.0
	<Nombre de Sección / Etapa>			

The screenshot shows a web application interface. At the top, there is a navigation menu with the following items: usuario, ingresar, reportes, idioma, Clientes, sistema, Bitacora, ABM_Perminos, permisos, personalizar ropa, CERRAR_SESION. On the right side of the menu, there is a language dropdown menu set to 'Español' and a user profile section labeled 'Usuario: tomi'.

Desde este formulario se puede acceder a todos los módulos del sistema:

- Usuario (gestión de usuarios)
- Reportes
- Idioma (gestión de idiomas)
- Clientes(gestión de clientes)
- Bitácora
- Sistema (Pedidos)
- ABM Permisos (gestión de permisos)
- Permisos (permisos que tiene el usuario)
- Personalizar Ropa (gestión de variables de ropa)
- Cerrar sesión

En la parte superior derecha se muestra:

- **Idioma actual**
- **Nombre del usuario logueado**

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

4. Gestión de Usuarios (ABM Usuarios)

USUARIO

CONTRASEÑA

AGREGAR

MODIFICAR

BORRAR

Ir al Historial de cambios

	nombre	contraseña	codigo
	aylen1	????R?M? 6?...	21
	tomi	????R?M? 6?...	24
▶	tomas	????R?M? 6?...	30
	alejandro	????R?M? 6?...	1031

☒ Rol ☐ Permiso

gerente

AGREGAR

BORRAR

gerente

- administrador
 - usuario
 - ABM_clientes
 - Entrar_Sistema
 - ABM_Idioma
 - ABM_usuario
 - Personalizar_ropa
 - ABM_permisos
 - Ver_bitacora
 - Ver_reportes

Desde este módulo el administrador puede:

4.1 Agregar Usuario

- Completar campos: Usuario, Contraseña
- Presionar **Agregar**
- El sistema valida campos vacíos y guarda el nuevo usuario
- Se actualiza la lista

4.2 Modificar Usuario

- Seleccionar un usuario de la lista
- Cambiar contraseña o datos permitidos
- Presionar **Modificar**

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

- Se actualiza la lista

4.3 Borrar Usuario

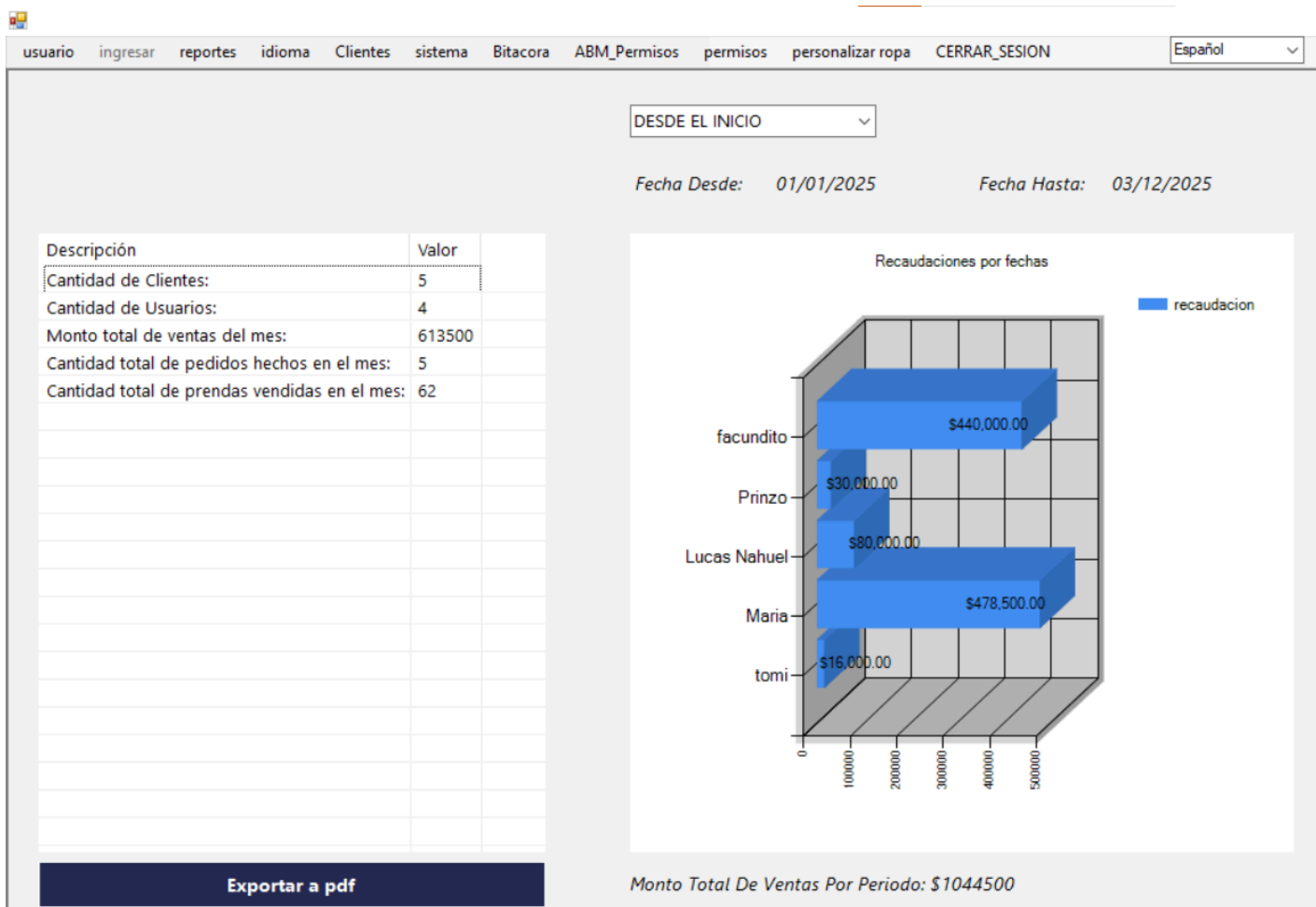
- Seleccionar el usuario
- Presionar **Borrar**
- Confirmar la acción
- Se actualiza la lista

4.4 Historial de Cambios (Control de Cambios)

Muestra todas las modificaciones hechas a los usuarios.

5. Reportes

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Versión: 1.0



El módulo de reportes muestra:

- Cantidad de clientes
- Cantidad de usuarios
- Total de ventas del mes
- Cantidad total de pedidos hechos en el mes
- Cantidad total de prendas vendidas en el mes

Los reportes pueden exportarse a **PDF**, generando un archivo con toda la información.

6. Gestión de Idiomas

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

usuario

ingresar

reportes

idioma

Cientes

sistema

Bitacora

ABM_Perminos

permisos

personalizar ropa

CERRAR_SESION

Español

	codigo	idioma	digito_verificad
▶	1	Español	8207
	2	English	8587
	1017	JAPONES	9608
	1065	eeee	6842
	1067	qqq	9451

idioma

AGREGAR

BORRAR

traducción

MODIFICAR

El usuario administrador puede:

6.1 Agregar Idioma

- Completar nombre del idioma
- Presionar botón **Agregar**
- Se actualiza la lista

6.2 Borrar Idioma

- Seleccionar idioma
- Presionar botón **Borrar**

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			Versión: 1.0
	<Nombre de Sección / Etapa>			

- Se actualiza la lista

6.3 Modificar Idioma / Traducciones

- Seleccionar idioma
- Seleccionar la traducción
- Escribir en el cuadro de texto la traducción
- Presionar el botón Modificar
- Se actualiza la lista

El sistema soporta múltiples idiomas configurados dinámicamente.

7. Gestión de Clientes

Cientes

Nombre_Completo

DNI

Provincia

Buenos_Aires

Localidad

1 Merlo

AGREGAR

MODIFICAR

BORRAR

Ir al formulario localidad

	codigo	nombre_compl	DNI	provincia	tipo	localidad
▶	2	tomi	1234	Cordoba	Basico	1 Merlo
	3	Maria	555	Buenos_Aires	Plata	3 Castelar
	4	Lucas Nahuel	45175043	La_Pampa	Basico	5 Liniers
	11	Prinzo	35546573	Chubut	Basico	4 Padua
	13	facundito	2312312	Salta	Basico	8 Hurlingan

Permite administrar los clientes registrados.

7.1 Agregar Cliente

Completar:

- Nombre completo
- DNI

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			Versión: 1.0
	<Nombre de Sección / Etapa>			

- Provincia
- Localidad

Y luego presionar **Agregar**.

7.2 Modificar Cliente

Seleccionar cliente → editar campos → Presionar el botón **Modificar**

7.3 Borrar Cliente

Seleccionar cliente → Presionar el botón **Borrar** → Confirmar

7.4 Gestión de Localidades

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

localidad

CODIGO

LOCALIDAD

AGREGAR

ELIMINAR

	localidad	codigo
▶	Merlo	1
	Castelar	3
	Padua	4
	Liniers	5
	San Martín	6
	Libertad	7
	Hurlingham	8

Desde este botón se abre el formulario para:

7.4.1 Agregar Localidad

- Completar: Localidad
- Presionar **Agregar**.

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin		Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
<Nombre de Sección / Etapa>				Versión: 1.0

- Se actualiza la lista

7.4.2 Modificar Localidad

Seleccionar localidad → editar campo → Presionar el botón **Modificar** → Se actualiza la lista

7.4.3 Borrar Cliente

Seleccionar localidad → Presionar el botón **Borrar** → Se actualiza la lista

8. Sistema / Pedidos

usuario	ingresar	reportes	idioma	Cientes	sistema	Bitacora	ABM_Permisos	permisos	personalizar ropa	CERRAR_SESION	Español
---------	----------	----------	--------	---------	---------	----------	--------------	----------	-------------------	---------------	---------

CLIENTE

Buscar Cliente:

Filtro

	codigo	nombre_compl	DNI	provincia	tipo	localidad
▶	2	tomi	1234	Cordoba	Basico	1 Merlo
	3	Maria	555	Buenos_Aires	Plata	3 Castelar
	4	Lucas Nahuel	45175043	La_Pampa	Basico	5 Liniers
	11	Prinzo	35546573	Chubut	Basico	4 Padua
	13	facundito	2312312	Salta	Basico	8 Hurlingan

Pedidos

Total

Fecha

Descuento

Cantidad De Ropa

Total

Fecha

Descuento

Este módulo permite gestionar los pedidos de los clientes.

UNIVERSIDAD ABIERTA INTERAMERICANA					
Facultad de Tecnología Informática					
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez		Fecha: 15/4/25
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3	
	sistema de ropa			Versión: 1.0	
	<Nombre de Sección / Etapa>				

Funciones principales:

8.1 Buscar Cliente

Permite buscar un cliente por nombre o DNI.

8.2 Agregar Pedido

- Seleccionar cliente
- Presionar **Agregar Pedido**
- El pedido se crea inicialmente con total = 0
- Se actualiza la lista

8.3 Borrar Pedido

- Seleccionar pedido
- presionar **Borrar**
- Se actualiza la lista

8.4 Personalizar Pedido (Detalles)

pedidos

Buzos

COLORES

Rojo

TALLES

S

CANTIDAD

CAPUCHA

SI

AGREGAR

MODIFICAR

BORRAR

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

Dentro de un pedido, el usuario puede agregar, modificar o borrar detalles de personalización de la prenda. Cada detalle contiene:

- **Tipo de prenda** (ej.: Buzo, Campera, Remera)
- **Color**
- **Talle**
- **Cantidad**
- **Capucha** (Sí / No)

La pantalla muestra un listado de los detalles cargados y los campos para editarlos.

8.5.1 Agregar Detalle

1. El usuario completa los campos:
 - Tipo de prenda
 - Color
 - Talle
 - Cantidad
 - Capucha
2. Presiona **Agregar Detalle**
3. Se actualiza la lista de detalles en pantalla

8.5.2 Modificar Detalle

1. El usuario selecciona un detalle existente
2. Edita los campos
3. Presiona **Modificar**
4. Se muestra la lista actualizada

8.5.3 Borrar Detalle

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Versión: 1.0

☐ Permiso
☒ Rol

gerente

▼

AGREGAR

MODIFICAR

BORRAR

CREAR NUEVO ROL

NODO SELECCIONADC gerente

gerente

administrador

usuario

ABM_clientes

Entrar_Sistema

ABM_Idioma

ABM_usuario

Personalizar_ropa

ABM_permisos

Ver_bitacora

Ver_reportes

usuario

ABM_clientes

Entrar_Sistema

hola1

gerente

administrador

usuario

ABM_clientes

Entrar_Sistema

ABM_Idioma

ABM_usuario

Personalizar_ropa

ABM_permisos

Este módulo administra los permisos del sistema utilizando una estructura **Composite**, es decir:

- Roles
- Permisos individuales
- Permisos hijos
- Jerarquías completas

Se puede:

10.1.1 Agregar Permiso o Rol

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			
	<Nombre de Sección / Etapa>			
				Fecha: 15/4/25
				Versión: 1.0

1. El usuario selecciona en el ComboBox si quiere agregar un **Permiso o Rol**
2. Completa el nombre del nuevo permiso/rol
3. Presiona **Agregar**
4. La lista del ComboBox se actualiza

10.1.2 Modificar

1. El usuario selecciona un permiso/rol existente del ComboBox
2. Cambia el nombre
3. Presiona **Modificar**
4. El ComboBox se refresca

10.1.3 Borrar

1. Seleccionar el permiso o rol a eliminar
2. Presionar **Borrar**
3. El ComboBox se actualiza

10.1.4 Crear Nuevo Rol

El usuario puede generar un rol vacío sin seleccionar nada del ComboBox:

1. Presiona **Crear Nuevo Rol**
2. Completa el nombre del nuevo rol
3. El sistema valida que no exista un rol con ese nombre
4. Se agrega el nuevo rol a la lista
5. El ComboBox se actualiza

11. Personalizar Ropa

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin		Legajo:	
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			Versión: 1.0
	<Nombre de Sección / Etapa>			

ADMINISTRADOR DE VARIABLES

☐ COLORES
 ☒ TALLES

AGREGAR

BORRAR

	codigo	variable
▶	2	S
	3	M
	4	L
	5	XL
	6	XXL
	7	XXXL

VINCULACIÓN ROPA

TIPO DE PRENDA

Buzos

☐ COLORES
 ☐ TALLES

AGREGAR

BORRAR

Este formulario permite configurar:

11.1 Colores

- Agregar color
- Borrar color

11.2 Talles

- Agregar talle

UNIVERSIDAD ABIERTA INTERAMERICANA				
Facultad de Tecnología Informática				
	Materia: Ingeniería de software		Docente: Julián Martín Rodríguez	
	Alumno: Tomas Meneces Fernandez Benjamin			Legajo:
	Localización: Castelar	Comisión: 2-A	Turno: Noche	Año: 3
	sistema de ropa			Versión: 1.0
	<Nombre de Sección / Etapa>			

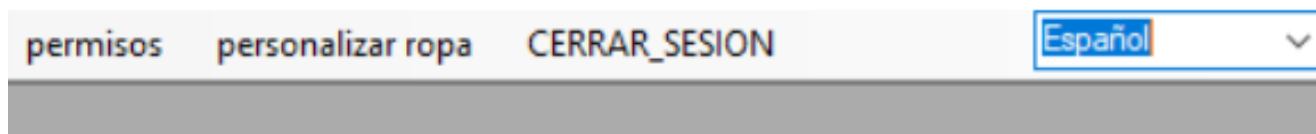
- Borrar talla

11.3 Vinculación con ropa

- Asociar color/talle a un tipo de prenda
- Borrar asociaciones

Esto permite que en los pedidos el usuario tenga disponible ropa personalizable.

12. Cerrar Sesión



El usuario presiona **Cerrar sesión**, confirma la acción y vuelve al formulario de inicio.

13. Conclusión

Este manual abarca todas las funciones del sistema Look Ideal, detallando cada formulario, operación, gestión y módulo disponible.