UNIVERSIDAD DE IBAGUÉ
FACULTAD DE INGENIERÍA
DESARROLLO DE APLICACIONES EMPRESARIALES
TERCER PROYECTO 2025B

OBJETIVO

Desarrollar una aplicación en Java (u otro lenguaje de programación de propósito general) que permita gestionar la información de dos objetos (maestro – detalle) y un objeto asociado a la ClaseA o ClaseB basados en el primer proyecto del curso realizando las operaciones de inserción, eliminación, actualización, búsqueda individual, búsqueda en lista (dos variantes) - CRUD y exponiendo estas funcionalidades a través de **servicios web REST**. La aplicación debe almacenar la información de los objetos (Ay B) en una base de datos relacional a través de dos tablas y los objetos seleccionados deben **gestionar la información de al menos 5 atributos (int, double, String, LocalDateTime y otro de su elección).** La Clase C debe ser un microservicio exponiendo las funcionalidades CRUD y tener su base de datos independiente**.** La aplicación tiene una arquitectura **web distribuida** donde varios *clientes* acceden a la lógica central de la aplicación ofrecida por un *servidor* utilizando la tecnología de **Servicios Web REST,** sin embargo, se deben implementar dos clientes utilizando una tecnología diferente (lenguaje diferente) a la de los microservicios (los dos clientes deben estar desarrollados en tecnologías diferentes).

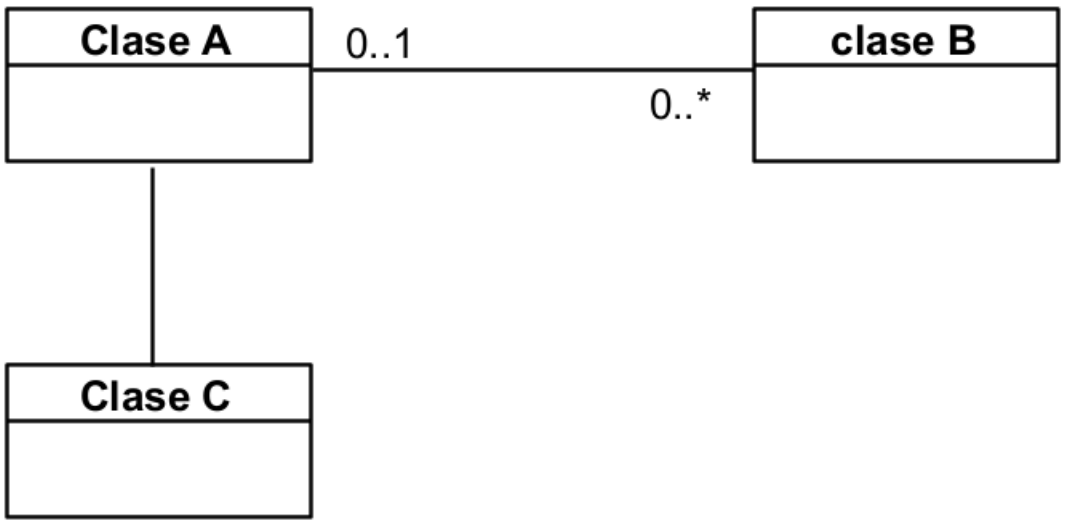

Figura 1. Diagrama de clases de los dos objetos del estudio de caso

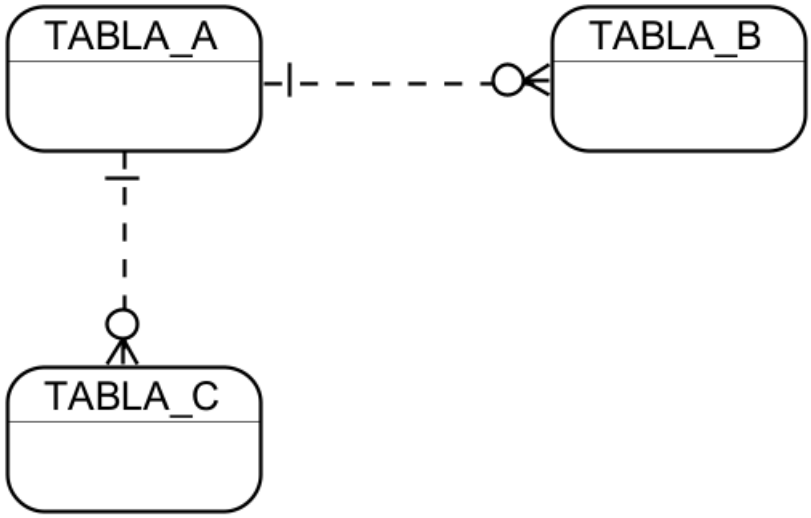

Figura 2. Diagrama relacional de las dos tablas del estudio de caso

CASO DE ESTUDIO

Considere que requiere gestionar la información de **dos de los objetos (maestro – detalle)** más relevantes para el caso de estudio seleccionado como proyecto del curso junto a un tercero relacionado a la Clase A o Clase B. Las operaciones básicas que se pueden realizar con los tres objetos están definidas en CRUD (Create, Read, Update, Delete) siendo la relación Clase A/B – Clase C la más relevante para este caso de estudio. En la figura 3 se describen las funcionalidades de la aplicación con cada uno de los objetos. Estas funcionalidades debes estar expuestas a través de microservicios REST y ser consumidas por dos clientes desarrollados en otro lenguaje de programación con el fin de garantizar la interoperabilidad. Los clientes deben ser operativos y funcionales, no se espera una interfaz gráfica demasiado elaborada, pero si amigable con el usuario.

Los objetos seleccionados deben tener una llave primaria, atributo(s) obligatorios y atributo(s) opcionales, así como criterios de validación a cada atributo como ser valores positivos o tener

cierto formato. La relación entre las dos tablas debe manejarse a través de la etiqueta One-Many o Many-One. El uso de una lista está prohibido en este proyecto.

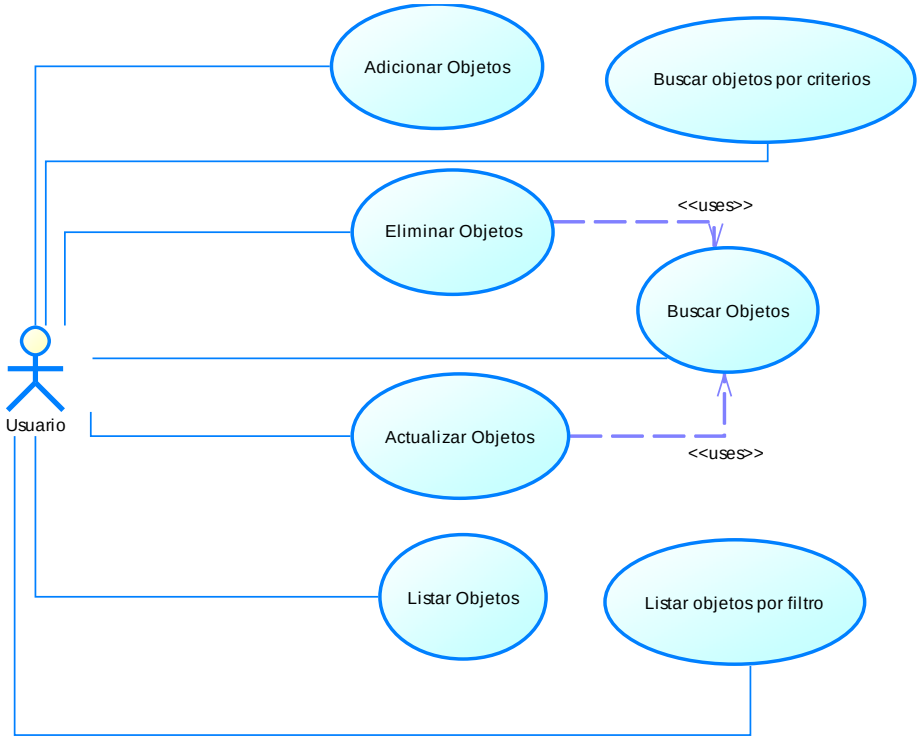

Figura 3. Diagrama de casos de uso de la aplicación para cada objeto

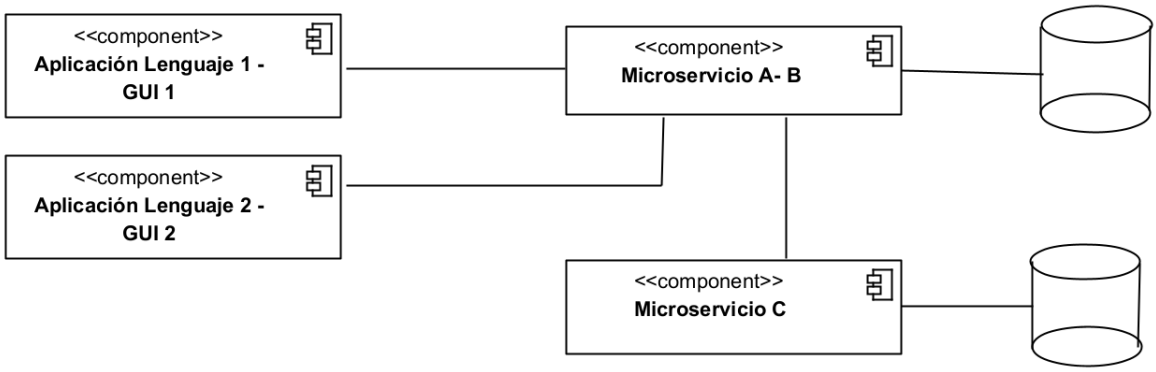

Figura 4. Diagrama de arquitectura

ACTIVIDADES A REALIZAR

Desarrolle una aplicación en Java (u otro lenguaje de programación de propósito general) que permita gestionar dos objetos (CRUD) y exponer la funcionalidad a través de servicios web REST para el primer microservicio, el segundo microservicio debe exponer la funcionalidad del tercer objeto (CRUD). La aplicación debe funcionar en una red (varios clientes hacen peticiones al servidor) y cumplir con las siguientes características:

1. Se debe utilizar la tecnología de **Servicios Web REST**.
2. En el primer microservicio, debe existir al menos dos clases estructurales que representen los objetos seleccionados. **Los objetos deben ser gestionados por tablas en una base de datos relacional**.
3. En el segundo microservicio, debe existir una tercera clase estructural y ser gestionado por una base de datos relacional.
4. El primer microservicio debe gestionar la invocación de las funcionalidades del segundo de forma transparente para los clientes GUI.
5. Desarrollar dos **clientes** en un lenguaje de programación diferente al utilizado en el servidor. La interfaz gráfica debe cumplir con los requerimientos de usabilidad. Cada ventana (equivalente a JFrame) debe implementar un solo caso de uso.

6. Para los casos de uso de eliminar y actualizar se debe realizar la previa búsqueda del objeto. **NO SE DEBE UTILIZAR LA MISMA VENTANA PARA REALIZAR MÁS DE UNA FUNCIONALIDAD**.
7. La ventana principal debe tener un menú que permita al usuario ir a la función deseada. Además, debe existir un menú "ayuda" con "Acerca de…" donde aparezca el nombre de los integrantes del equipo y la versión de su aplicación.
8. La aplicación debe ser "subida" a GitHub o equivalente y compartida con el correo carlos.lugo@unibague.edu.co
9. El caso de uso Listar permite realizar la consulta de todos los registros y filtrar al menos por un parámetro.

CONDICIONES PARA LA ENTREGA

1. La actividad será desarrollada de forma grupal (3-4). Cualquier taller que sea encontrado como copia de otro, será reportado al comité disciplinario según el reglamento estudiantil.
2. La implementación desarrollada deberá presentarse en la clase, en el tiempo estipulado para tal fin.
3. La calificación del aplicativo desarrollado se basará en la rúbrica especificada para tal fin.
4. La implementación deberá entregarse a través de la plataforma en el enlace dispuesto para tal fin.
5. La actividad tiene un valor proporcional correspondiente al segundo taller del curso.

RÚBRICA DE CALIFICACIÓN

UNIVERSIDAD DE IBAGUÉ
FACULTAD DE INGENIERÍA
DESARROLLO DE APLICACIONES EMPRESARIALES
TERCER PROYECTO 2025B

Arquitectura	***Implementación de PD***	***Insertar C***	***Eliminar C***	***Listar C con parámetro***	***Actualizar C***	***Consulta C con parámetro***
La aplicación debe cumplir con las especificaciones de diseño, las clases GUI (vistas), Model y estructurales deben existir así como la clase XXX en sus respectivos paquetes. Las operaciones deben ser coherentes con la naturaleza de la clase, por ej.: la clase GUI debe realizar procesos asociados a interfaz gráfica.	Debe implementar la tecnología de Servicios Web REST **en los dos microservicios**. El clientes debe estar desarrollado en un lenguaje de programación diferente al del servidor. Cada caso de uso se califica de acuerdo a su funcionalidad en **LAS DOS** tecnologías. La aplicación debe almacenar la información en una Base de Datos relacional.	Operación que permite adicionar un nuevo objeto de la clase C de acuerdo a sus características (atributos). Esta adición debe asegurar de alguna forma la integridad referencial.	Operación que permite eliminar un objeto previamente **buscado** por el usuario. Antes de ser eliminado, se debe mostrar al usuario **TODA** la información del objeto.	Operación que permite Listar todos los objetos en la Clase C y dos de la Clase A a través de una grilla. Es posible filtrar la información con un parámetro.	Operación que permite actualizar un o varios atributos del objeto previamente **buscado** por el usuario. Antes de ser actualizado se debe mostrar al usuario TODA la información del objeto.	Operación que permite consultar UN SOLO objeto mostrando sus características (atributos) a partir de un parámetro ingresado por el usuario.
REQUERIMIENTO	**REQUERIMIENTO**	20%	20%	20%	20%	20%

REQUERIMIENTO: En caso de NO cumplir con el requerimiento descrito, este producto NO será aceptado y tendrá una nota de 0.00