

OBJETIVO

Desarrollar una aplicación en Java (u otro lenguaje de programación de propósito general) que permita gestionar la información de un objeto XXX realizando las operaciones de inserción, eliminación, actualización, búsqueda individual, búsqueda en lista (CRUD) y representación de un valor en una gráfica de línea (u otro tipo). **La aplicación debe permitir la conexión a una base de datos Oracle donde reside la información de forma persistente.** La aplicación tiene una arquitectura **distribuida** donde varios *clientes* acceden a la lógica central de la aplicación ofrecida por un *servidor* utilizando la tecnología **RMI o equivalente**.

CASO DE ESTUDIO

Considere que requiere gestionar la información del objeto más relevante para el caso de estudio seleccionado como proyecto del curso (acordado con el profesor en clase). Para hacerlo, requiere que la aplicación cumpla con el patrón de diseño MVC (Modelo Vista Controlador). Las operaciones básicas que se pueden realizar con el objeto están definidas en CRUD (Create, Read, Update, Delete) + representación en una gráfica; estas funcionalidades deben ser implementadas a través de una aplicación cliente con interfaz gráfica de usuario sencilla pero funcional. En la figura 1 se describen las funcionalidades de la aplicación. En la figura 2 se describe la arquitectura general de la solución informática.

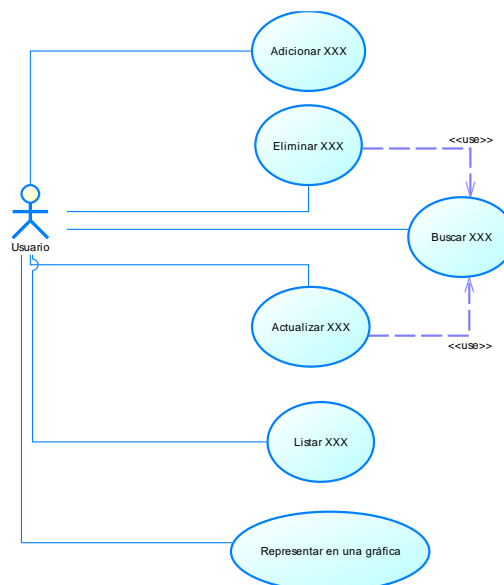


Figura 1. Diagrama de casos de uso para la aplicación XXX

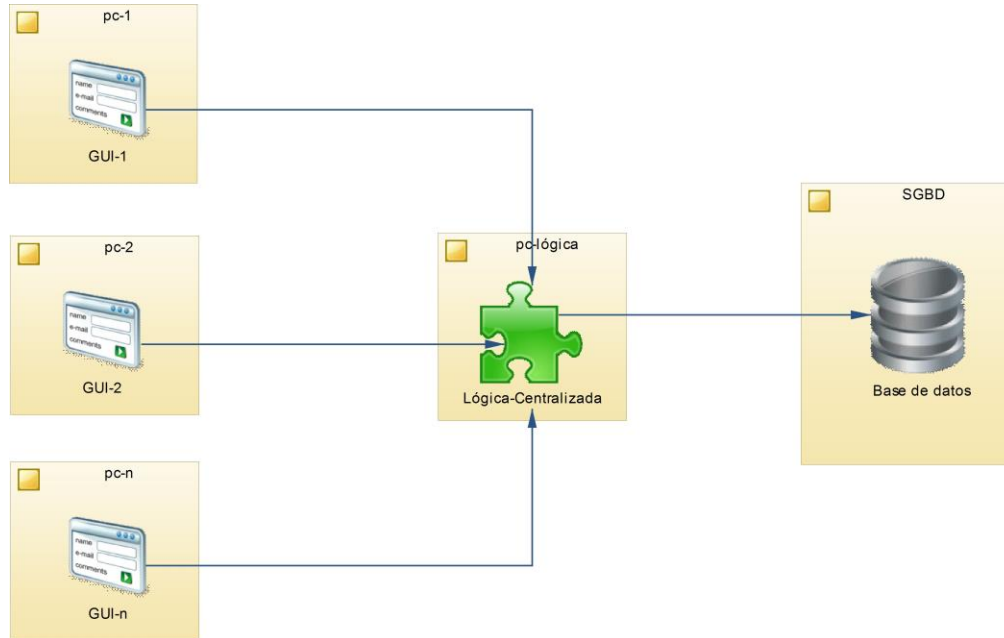


Figura 1. Arquitectura general de la solución

ACTIVIDADES A REALIZAR

Desarrolle una aplicación en Java (u otro lenguaje de programación de propósito general) que permita gestionar un objeto XXX (CRUD + representación en una gráfica) utilizando el patrón de diseño MVC a través de una interfaz gráfica de usuario sencilla y funcional. La aplicación debe funcionar en una red local (varios clientes hacen peticiones al servidor) y cumplir con las siguientes características:

1. Se debe implementar el patrón de diseño **MVC**, es decir, se deben separar las capas en la aplicación.
2. Se debe utilizar la tecnología **RMI**, Remoting o similar.
3. Debe existir una clase estructural XXX que represente el objeto central estructural de la aplicación y objeto que será almacenado en una base de datos Oracle.

4. La interfaz gráfica debe cumplir con los requerimientos de usabilidad. Cada ventana (JFrame o equivalente) debe **implementar un solo caso de uso**.
5. 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.**
6. 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.
7. La aplicación debe ser “subida” a GitHub o equivalente y compartida con el correo carlos.lugo@unibague.edu.co

CONDICIONES PARA LA ENTREGA

1. La actividad será desarrollada de forma grupal (3). 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 primer taller del curso.

RÚBRICA DE CALIFICACIÓN

**UNIVERSIDAD DE IBAGUÉ
FACULTAD DE INGENIERÍA
ARQUITECTURA ORIENTADA A SERVICIOS
PRIMER TALLER 2021B**

Ítem	Arquitectura	Implementación de PD	Insertar XXX	Eliminar XXX	Listar XXX	Consultar XXX	Actualizar XXX	Representar gráficamente XXX
Descripción	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.	El patrón de diseño MVC deben estar implementados en la solución. La aplicación debe almacenar la información en una base de datos Oracle.	Operación que permite adicionar un nuevo objeto de la clase XXX de acuerdo a sus características (atributos).	Operación que permite eliminar un objeto previamente buscado por el usuario.	Operación que permite Listar todos los objetos en la Aplicación a través de una grilla.	Operación que permite consultar UN SOLO objeto mostrando sus características (atributos).	Operación que permite actualizar un o varios atributos del objeto previamente buscado por el usuario.	Operación que permite representar gráficamente, utilizando una librería como JFreeChart, la información de uno o varios atributos del objeto XXX.
Peso	10%	REQUERIMIENTO	10%	10%	20%	10%	20%	20%

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