

Objetivo general del proyecto

El objetivo general de este proyecto es practicar varias etapas del desarrollo de una aplicación de software, desde el análisis hasta la construcción una aplicación funcional. A través del proyecto, los estudiantes pondrán en práctica todas las habilidades desarrolladas en el curso.

Objetivos específicos del proyecto

Durante el desarrollo de este proyecto se buscará el desarrollo de las siguientes habilidades:

1. Identificar y abstraer entidades en un dominio particular, así como sus características y relaciones.
2. Construir diagramas de clase UML para expresar modelos de dominio y modelos de diseño.
3. Diseñar una aplicación basada en el paradigma orientado a objetos y expresar el diseño, justificando las decisiones importantes que hayan tomado.
4. Implementar un diseño utilizando el lenguaje de programación Java, poniendo énfasis en la implementación de las funcionalidades requeridas.
5. Construir un conjunto de programas sencillos que demuestren que las funcionalidades están correctamente implementadas.

Instrucciones generales

A continuación, encontrará la definición inicial del proyecto, que será la base del trabajo de los 3 proyectos que se construirán durante todo el semestre.

El trabajo del proyecto #2 debe partir de la base del proyecto #1, y el trabajo del proyecto #2 será la base del proyecto #3. Si es necesario, podrán hacerse cambios en cada una de las etapas para mejorar o corregir decisiones que se hayan tomado en las etapas anteriores.

En el proyecto #1 no es necesario que haya una interfaz que le permita al usuario interactuar con la aplicación. Sin embargo, deben construirse programas que demuestren que la lógica de la aplicación y las historias de usuario quedaron debidamente implementadas. Estos programas (o consolas) deben solicitarle al usuario la menor cantidad posible de información para correr (por ejemplo, el nombre de un archivo para cargar los datos, o el nombre de una funcionalidad que se quiera demostrar)¹. Estos programas también tienen que mostrar, a través de la consola, el estado de la aplicación de una forma que sea fácil de entender y que permita apreciar que la funcionalidad es correcta.

En el proyecto #2, uno de los requerimientos será implementar un mecanismo de interacción con el usuario basada en consola. En este proyecto sí será necesario implementar una por una las historias de usuario y permitir toda la interacción con los usuarios para que entreguen la información que se requiere. El motivo de esta

¹ Estos programas también podrían verse como precursores de pruebas automatizadas. Si usted quiere, puede implementar pruebas usando Junit, pero eso se estudiará más adelante en el curso.

separación entre los requerimientos del proyecto #1 y el proyecto #2 es que en el primero el foco debería estar en el diseño y la implementación de la lógica de la aplicación y no en la lógica de la interacción con los usuarios, validación de las entradas y presentación de los resultados.

El proyecto debe desarrollarse en **grupos que podrán cambiar para el proyecto #2, pero no para el proyecto #3.**

Contexto del proyecto

Dentro del proyecto del curso vamos a construir un sistema para apoyar las operaciones de una galería y casa de subastas.

Este sistema se encargará de tres grandes funcionalidades. En primer lugar, tendrá un inventario de las piezas disponibles para ser expuestas o ser subastadas. En segundo lugar, tendrá un sistema que gestionará el proceso de compra o subasta de una pieza o de una colección de piezas. En tercer lugar, manejará la información de los propietarios y compradores de las piezas.

Inventario de piezas

El sistema que usted debe construir será capaz de manejar el inventario de las piezas de la galería. Las piezas pueden ser de diferente tipo (pinturas, esculturas, videos, fotografías, impresiones, etc.) y tendrán atributos diferentes según el tipo. Por ejemplo, para las esculturas es importante conocer las dimensiones (alto, ancho y profundidad), los materiales de construcción, el peso, si necesita electricidad, y si requiere algún otro detalle para su instalación. También hay atributos comunes, como el título, y el año y lugar de creación. Además, cada pieza debe tener un autor, aunque este podría ser anónimo. También hay piezas que son de la autoría de colectivos de artistas.

El inventario incluye información sobre piezas que pueden estar exhibidas o pueden estar en la bodega de la galería. En la mayoría de los casos, las piezas las obtiene la galería a través de la modalidad de “consignación”. Es decir que el dueño real de la pieza se la entrega temporalmente a la galería para que la exhiba, la venda o la subaste. Al finaliza el tiempo acordado, si la pieza no se ha vendido, debe ser regresada a su propietario.

La galería también mantiene la información de piezas que haya tenido en el pasado.

El administrador de la galería es el único que puede registrar el ingreso de una pieza al inventario, o confirmar que se realizó una venta o una devolución de una pieza.

Los otros **empleados** de la galería pueden realizar el resto de las acciones que sean necesarios para **administrar las piezas del inventario**.

Compra y subasta de piezas

El corazón del sistema será la parte que permitirá que alguien **compre una pieza pagando un valor fijo establecido**, o a través de **una subasta**.

Una característica muy importante de cualquier pieza es si está disponible para ser vendida por un valor fijo o no. Para aquellas piezas que estén disponibles, un **usuario registrado** como comprador en la plataforma **podrá ofrecerse a realizar la compra**. En ese momento la pieza quedará bloqueada (nadie más podrá comprarla) hasta que el administrador del sistema **verifique que el usuario sea real y la seriedad de la oferta**. En caso positivo, **la pieza quedará vendida**. En caso negativo, **volverá a su estado anterior**. Las piezas que estén en la bodega de la galería también pueden ser vendidas – no sólo las que estén exhibidas.

Periódicamente la galería **realizará subastas**, en las que estarán disponibles una o varias piezas. Durante una subasta, cada pieza **tendrá un valor mínimo y un valor inicial**. Ningún comprador podrá ofrecer menos del valor inicial, el cual será conocido, y la pieza no se venderá a menos que se llegue al valor mínimo, el cual no se conocerá

con anterioridad. En una subasta sólo pueden participar aquellos compradores que **hayan sido verificados con anterioridad por el administrador de la galería**.

Durante una subasta, **el sistema ayudará a llevar la traza de las ofertas** realizadas por cada pieza: un usuario con **el rol 'operador'** se encargará de ir registrando todo lo que vaya pasando.

Propietarios, compradores y pagos

La tercera parte del sistema es la que tiene que ver con el manejo los propietarios de las piezas, los compradores, y los pagos.

De los propietarios de las piezas se tendrá información básica de contacto, y ellos podrán entrar a **consultar tanto el estado de sus piezas**, como **la historia de las piezas que hayan sido de ellos en el pasado**.

De los compradores se tendrá también información básica de contacto e información de todas sus compras. Cuando un administrador verifique a un comprador, establecerá un valor máximo de sus compras; si el comprador supera ese valor, **tendrá que demostrarle al administrador que puede pagarlo** y el **administrador aumentará el límite**.

Los propietarios de piezas pueden ser también compradores y viceversa.

Los pagos en la galería pueden hacerse con tarjeta de crédito, transferencia electrónica o en efectivo. **El usuario con el rol de cajero de la galería podrá registrar cuando se haya hecho un pago**, lo cual permitirá que **la pieza involucrada sea entregada a su nuevo propietario**.

Aspectos técnicos y restricciones

Toda la información debe ser persistente. La información debe almacenarse en archivos (pueden ser planos, en el formato que ustedes definan, o binarios), dentro de una carpeta y se puede suponer que sólo la aplicación va a escribir y leer de esa carpeta (ningún usuario malicioso va a modificar los archivos que ahí se encuentren sin utilizar la aplicación). La carpeta no puede ser la misma carpeta donde se encuentre el código fuente de la aplicación.

La persistencia no necesariamente debe hacerse en un solo archivo: diseñe con cuidado cuántos archivos habrá y cómo van a estar estructurados.

Todos los usuarios del sistema deben tener un **login y un password**.

La aplicación debe estar hecha en Java.

El hecho de que en esta entrega no se tenga que construir una interfaz para los usuarios (sólo programas que demuestren que la lógica quedó correctamente implementada), no significa que no deban estar implementadas todas las funcionalidades necesarias para que en el proyecto #2 se puedan completar las historias de usuario.

No es necesario que se soporten funcionalidades que no hayan sido mencionadas, pero se pueden incluir funcionalidades adicionales si facilitan el trabajo.

Entrega 1: Análisis del proyecto

La primera etapa del proyecto consiste en realizar el análisis del sistema que debe construir.

Actividades

1. Construya un modelo de dominio (diagrama de clases) a partir de la información del caso:
 - a. Identifique las entidades que aparecen dentro del caso.
 - b. Identifique las características (atributos) de esas entidades.
 - c. Establezca las relaciones entre las entidades, incluyendo asociaciones y relaciones de herencia.

2. Construya un documento de análisis para su proyecto. El documento debe incluir:
 - a. El modelo de dominio que construyó en el paso anterior.
 - b. Una descripción con las restricciones del proyecto.
 - c. Una descripción de lo que demostrará cada uno de los programas de prueba.

Entrega

1. El proyecto debe entregarse en una carpeta dentro del repositorio GIT del grupo con el nombre **“Proyecto 1”**. Dentro de esta carpeta debe existir una carpeta con el nombre **“Entrega 1”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo tanto los archivos fuente de los diagramas como imágenes que se puedan leer con facilidad.
2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **“Proyecto 1 - Entrega 1”**.

Entrega 2: Diseño e implementación

Teniendo en cuenta el análisis realizado en la primera entrega del proyecto, para la segunda entrega debe realizar el diseño detallado del sistema y construir la implementación del sistema.

Actividades

1. Realice el diseño y construya un documento de diseño donde presente el diseño, haciendo especial énfasis en justificar las decisiones clave que haya tomado. El documento debe incluir, como mínimo, los siguientes elementos:
 - a. Un diagrama de clases de diseño que incluya todas las clases, incluyendo sus relaciones, atributos y métodos.
 - b. Un diagrama de clases de alto nivel, que incluya todas las clases y sus relaciones, pero no todos los métodos ni atributos. Este diagrama facilitará entender las relaciones entre clases.
 - c. Diagramas de secuencia para las funcionalidades que usted considere críticas:Estos elementos NO son los únicos que debe incluir su documento: piense en qué otros diagramas, descripciones y justificaciones pueden ser necesarias para poder entender totalmente el diseño. Si su documento incluye únicamente los tres elementos mencionados antes, con seguridad será considerado insuficiente.
2. Implemente el sistema que diseñó. Tenga en cuenta que los detalles de la implementación deben ser coherentes tanto con el modelo de clases, como con los diagramas de secuencia.
3. Asegúrese de que la documentación del diseño sea consistente con la implementación.

Entrega

1. Dentro de la carpeta del proyecto debe crear una carpeta con el nombre **“Entrega 2”** donde deben quedar todos los elementos correspondientes a esta entrega, incluyendo los archivos fuente de los diagramas, el documento de diseño, y el proyecto Eclipse con instrucciones para su ejecución. Incluya también archivos de prueba para poder correr las aplicaciones y tener datos con los que se pueda probar con facilidad.

2. Entregue un enlace al repositorio a través de Bloque Neón en la actividad designada como **“Proyecto 1 - Entrega 2”**.