

## Documento de diseño de Galería

**Lina Muñoz - 202310172**

**Daniel Camilo Quimbay Velásquez - 202313861**

**Tomás Emilio Velásquez Delgado - 202311016**

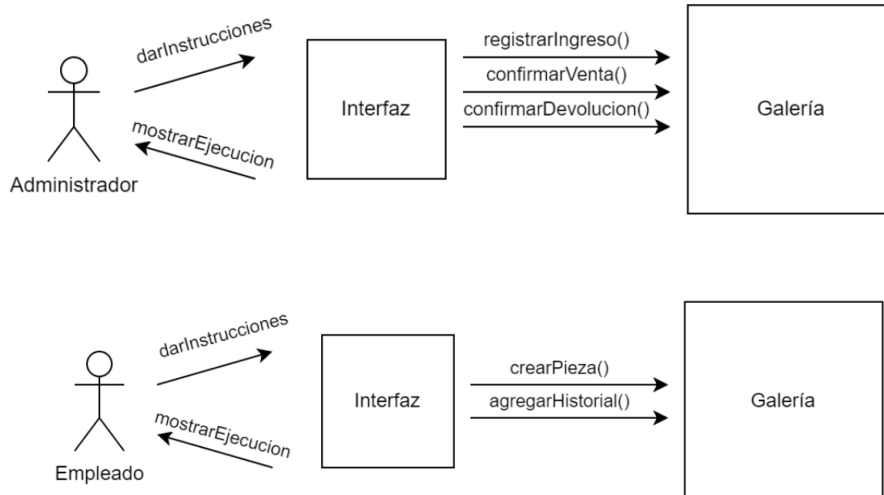
### 1. Contexto del problema

El sistema para apoyar las operaciones de la galería y casa de subastas debe encargarse de tres grandes funcionalidades:

- Modificar inventario de las piezas disponibles para ser expuestas o subastadas
- Gestionar el proceso de compra o subasta de una pieza o colección de piezas
- Manejar la información de los propietarios y compradores de las piezas

#### 1.1 Modificar Inventario

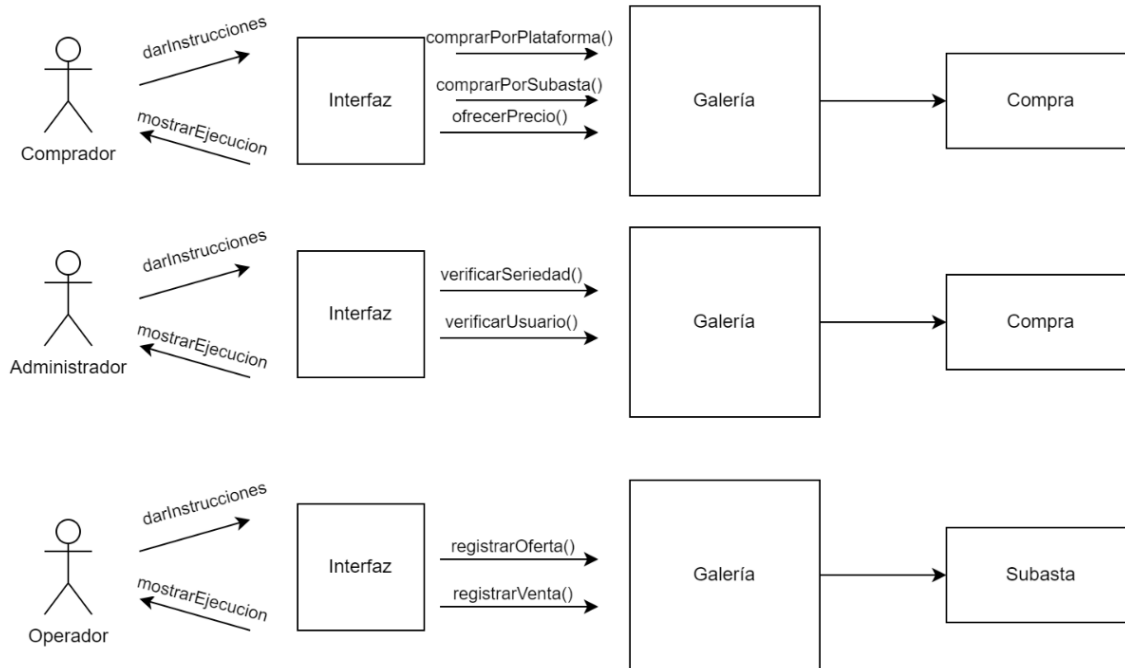
Entonces para manejar estas funcionalidades debemos dividir los roles que tomarán los usuarios para saber que pueden hacer. Primero, tenemos a los empleados que pueden ser divididos en dos: administrador de galería y empleado. El administrador de galería es el único que puede realizar acciones de registrar piezas y confirmar ventas o devoluciones, mientras que los demás empleados pueden realizar el resto de las acciones.



#### 1.2 Gestionar compras

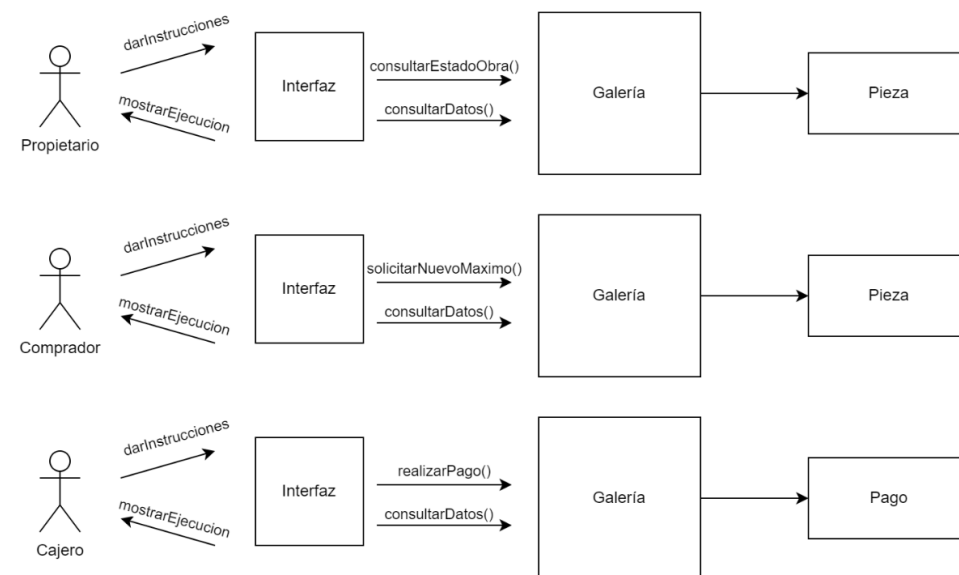
Después, tenemos la compra y subasta de piezas en la que participan el administrador de la subasta, el comprador y el operador. También tenemos compra por la plataforma y la compra por subasta. En la compra por plataforma el comprador se ofrece a realizar la compra, después se bloqueará la compra a otros usuarios hasta que el administrador revise la seriedad de la compra y después se venderá o quedará en su estado original.

En la subasta los compradores podrán ofrecer un precio superior al precio inicial y no se venderá si no se llega al precio mínimo. El operador será el que se encargue de llevar el control de todo lo que va sucediendo.



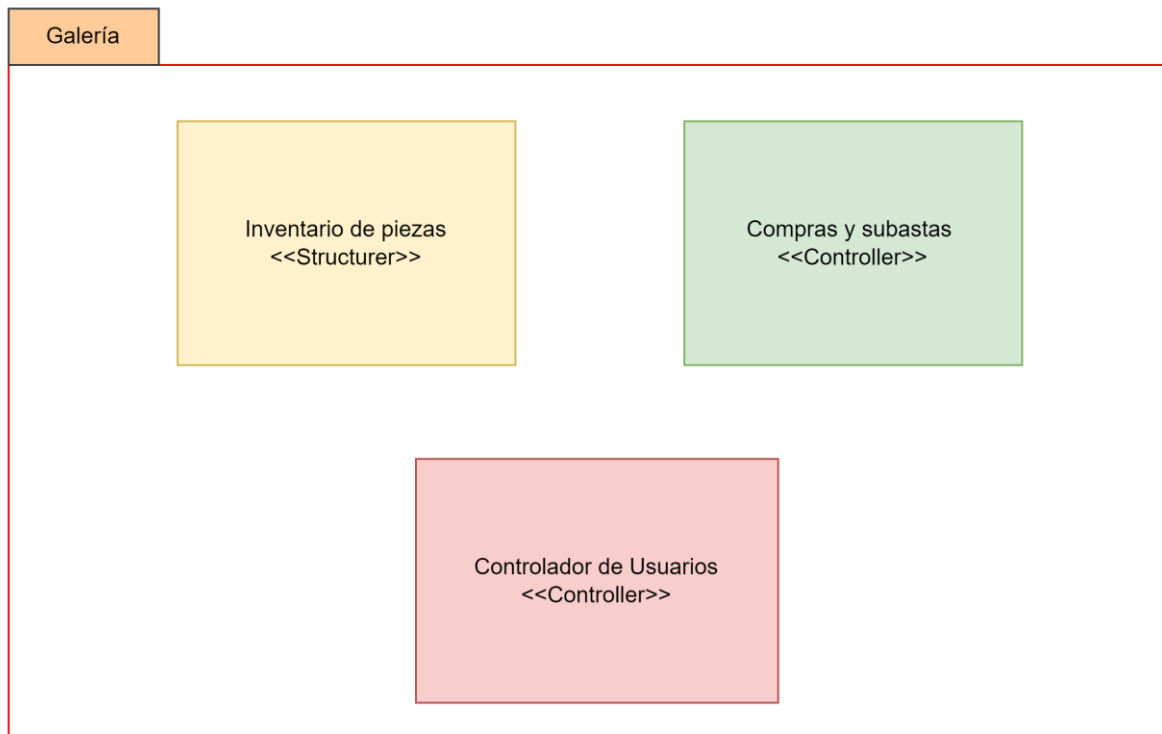
### 1.3 Manejar pagos, compradores y propietarios

Por último, tenemos el manejo de los propietarios y los pagos de las obras. Los roles que van a interactuar en estas funcionalidades son: comprador, propietario, administrador y cajero. Los propietarios podrán ver el estado de sus obras y de las que hayan sido suyas en el pasado. Los compradores podrán ver sus datos y solicitar un aumento de su valor máximo de compras el cuál se encargará de verificar un administrador. El cajero podrá realizar el registro de pago de un comprador a un propietario.



## 2. Primer nivel

Identificar los principales componentes que debe tener el sistema decidimos dividir la galería en los siguientes cuatro que se explorarán a profundidad después.



## 2.1 Objetos y roles

### 2.1.1 Inventario

Teniendo en cuenta que una gran parte del sistema es la enfocada en modificar y gestionar el inventario se ve la necesidad de tener un componente que se encargue de esto. En primer lugar, tendremos una clase que se llame inventario el cuál quedaría bien catalogado como *Information Holder* ya que su única función es almacenar el inventario y a través de métodos que ejecutarán los usuarios se podrá modificar. Este componente debe poder encargarse de agregar, cambiar y eliminar piezas del inventario. Además, debe dividirse en la colección de piezas que están en bodega, la colección en exhibición y el histórico de piezas que hayan sido parte del inventario.

### 2.1.2 Compra y subasta de piezas

A partir de que 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, se ve la necesidad de tener un componente que controle la compra y subasta de piezas, con el estereotipo controller. Este componente debe encargarse principalmente de crear subastas y compras, y asignárselas a sus compradores y a la galería.

### 2.1.3 Controlador Usuarios

El Controlador Usuarios, clasificado como un Controller en la arquitectura del sistema, es un componente crucial para la gestión de los diferentes roles de usuarios dentro de la galería. Este componente se encarga de agregar y modificar, así como de asignarles roles específicos dentro del sistema.

Para ello, el Controlador Usuarios debe contar con métodos para agregar Compradores, Propietarios, Administradores de Galería, Cajeros, y Operadores de Subastas. Además, debe permitir la asignación y modificación de roles a los usuarios existentes y la eliminación de usuarios del sistema si es necesario.

Este controlador es fundamental en la organización y administración de los usuarios de la galería, asegurando que cada uno tenga los permisos y funciones adecuadas según su rol dentro del sistema. Lo anterior quiere decir que, en cada clase de los roles, se van a tener los requerimientos específicos que cada rol puede realizar.

## 2.2 Responsabilidades

Teniendo en cuenta todos los requerimientos funcionales que debe tener el sistema se identificaron las principales responsabilidades del sistema. A continuación, enlistaremos estas responsabilidades y las asignaremos a uno de los componentes que se identificaron.

Responsabilidad	Componente
Agregar pieza al inventario	Inventario
Buscar pieza	
Confirmar venta en inventario	
Devolver pieza al propietario	
Poner pieza en exhibición	
Guardar pieza en bodega	
Bloquear una pieza	
Desbloquear una pieza	
Añadir al historial de inventario	
Crear Subasta	Compra y subasta de piezas
Crear Compra	
Recibir una Oferta de subasta	
Evaluar que una oferta de subasta sea superior al valor inicial y al limite	
Vender pieza	
Verificar si una pieza puede ser vendida por un valor fijo	
Crear los diferentes tipos de roles (Comprador, propietario, cajero, administrador, operador Subasta)	Controlador usuarios
Generar Id únicos	
Los requerimientos que le correspondan a cada rol (esta más a fondo en nivel 2)	
Verificar login y password, lo que implica la existencia del usuario o cliente	

Estas responsabilidades se catalogaron teniendo en cuenta cuál componente debe encargarse de implementarla en vez de qué usuario tiene que realizarla, ya que se manejará este tema más a profundidad más adelante.

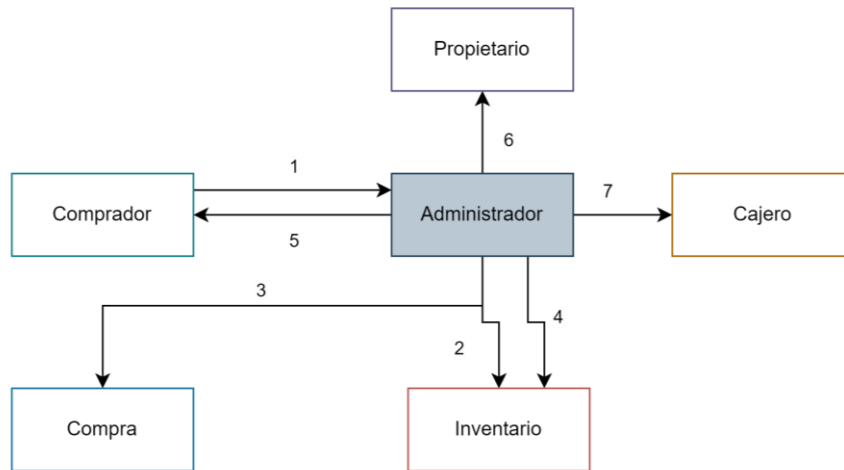
## 2.3 Colaboraciones

### 2.3.1 Confirmar Venta

Permite que, tras un intento de compra fija de cierta pieza, el cajero pueda registrar la compra tras analizarla el administrador de la galería y hacer los cambios necesarios.

- 1) El Comprador realiza una tentativa de compra de cierta pieza.
- 2) El administrador bloquea la pieza.
- 3) El administrador verifica seriedad compra.

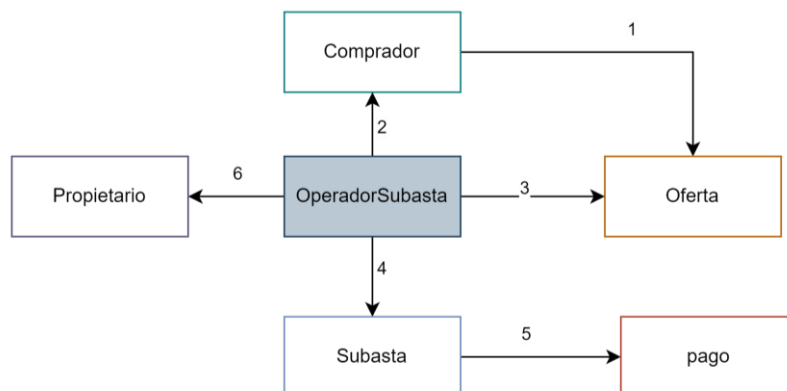
- 4) El administrador quita de piezas disponibles y las agrega a pasadas.
- 5) Se le agrega a el comprador las piezas y una compra.
- 6) Se le asigna al propietario las piezas actuales.
- 7) El cajero después de la confirmación del Administrador de galería registra el pago.



### 2.3.2 Realizar Subasta

Es el proceso en el cual los compradores realizan ofertas y el operador evalúa las ofertas y al terminar hace las decisiones de como finalizar la subasta

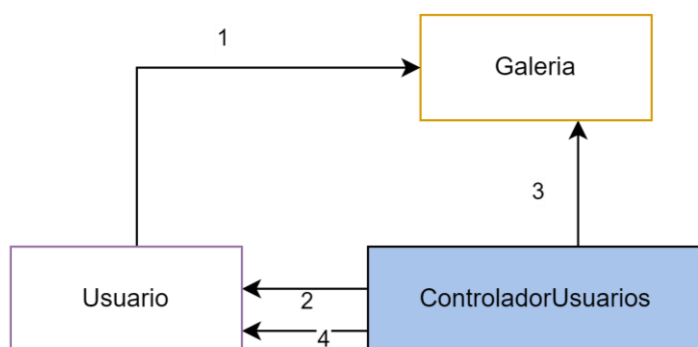
- 1) El comprador realiza una oferta.
- 2) El Operador verifica que existe el comprador
- 3) El Operador evalúa la oferta.
- 4) El Operador Subasta registra la oferta en la Subasta.
- 5) Se decide un ganador de la subasta y se realiza el pago.
- 6) La pieza se guarda en las piezas del comprador



### 2.3.3 Crear usuarios

En este proceso se asigna al usuario nuevo, su rol adentro de la aplicación y según esta asignación, tiene acceso a cierta información o procesos del programa.

- 1) Un usuario se registra en el sistema
- 2) El controlador verifica al usuario
- 3) El controlador lo ubica en su respectiva colección
- 4) Según la colección, se le asigna información y funcionalidades a los que puede acceder (empleados a galería, compradores a piezas Disponibles y propietarios a sus propias piezas)



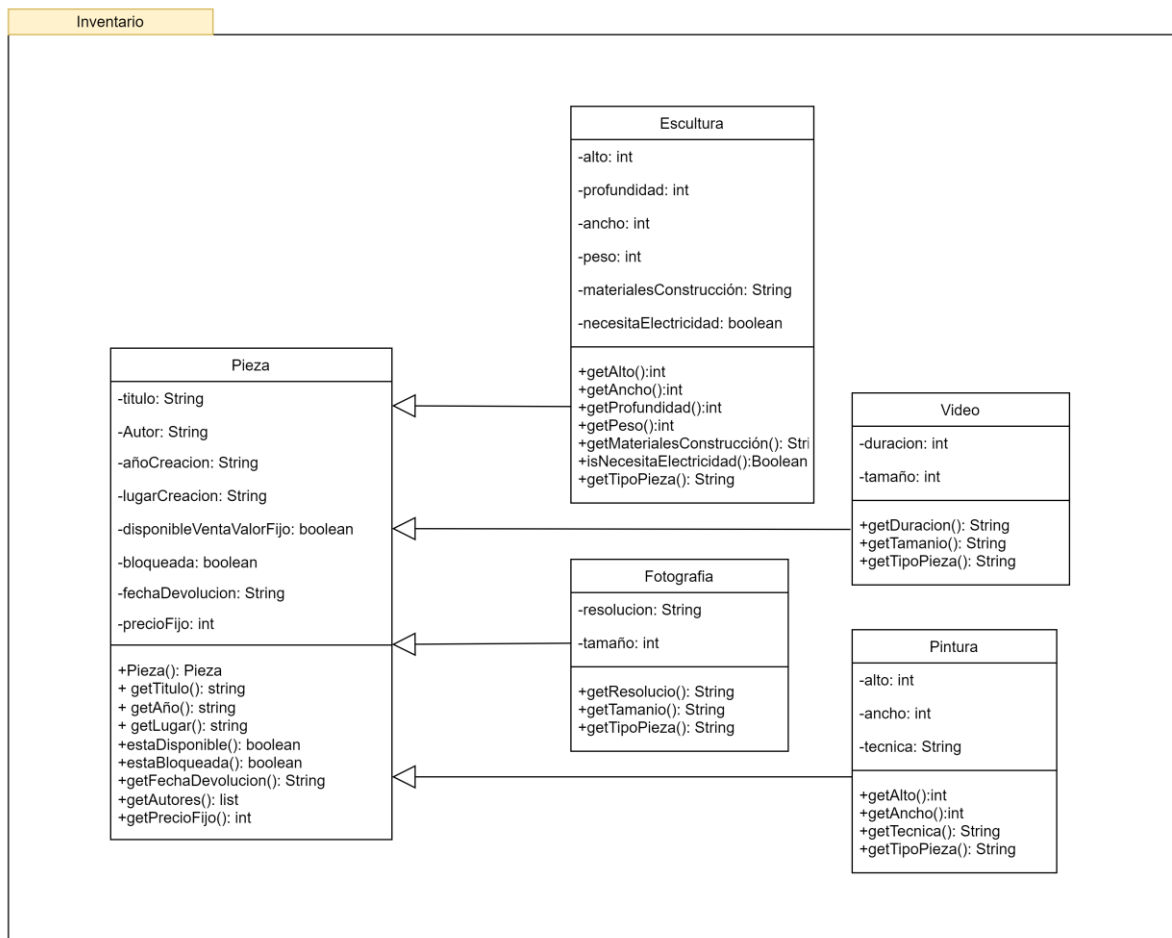
## 3. Segundo nivel

El primer nivel incluyó diagramas de alto nivel por lo que hace falta expandir los componentes para que las responsabilidades se puedan asignar a clases más específicas y se pueda tener mayor encapsulamiento y que el sistema sea más escalable.

### 3.1 Objetos y roles

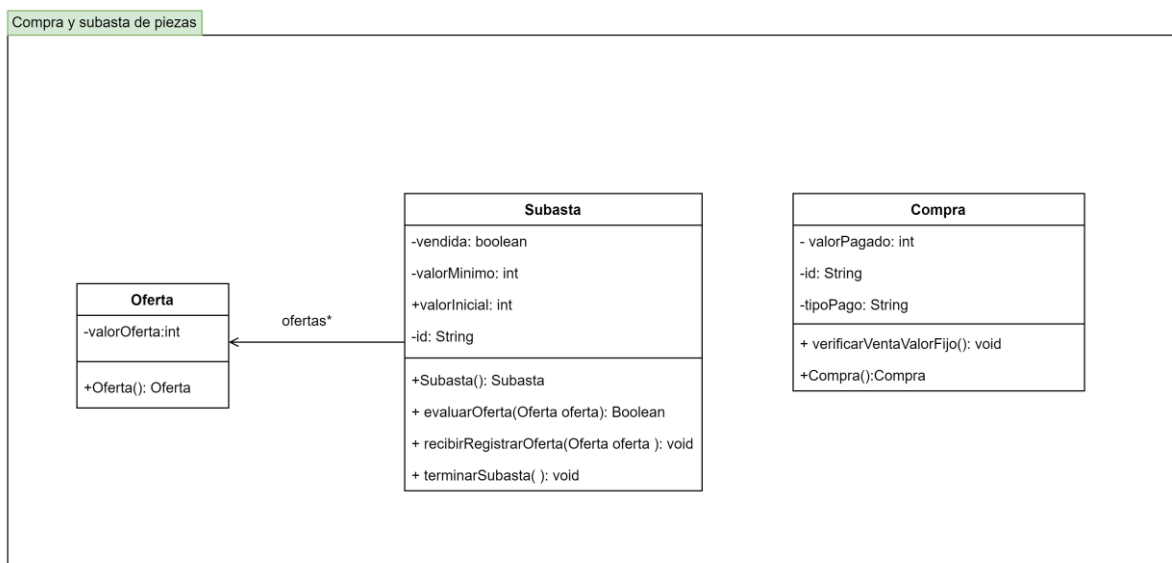
#### 3.1.1 Componente de Inventario

Con una sola clase de Inventario no es suficiente para cumplir con todos los requerimientos, por lo que se necesita crear más clases para poder satisfacer las necesidades del inventario. Es por esto por lo que se decidió dividir el inventario en más clases. Primero, tendremos la clase inventario que seguirá siendo un Information Holder de las colecciones de piezas que hacen parte de la bodega, la exhibición y el historial. Segundo, identificamos que necesitamos una clase “Pieza” donde podamos crear todos los atributos necesarios de las piezas, esta clase será abstracta ya que hay distintos tipos de piezas (pinturas, esculturas, videos, fotografías, etc.) que tienen varios atributos en común. Además, dado que una pieza puede tener referencia a más de un autor debemos crear una clase “Autor” que tenga esta información y que sea parte de una colección de autores en la clase de “Pieza”.



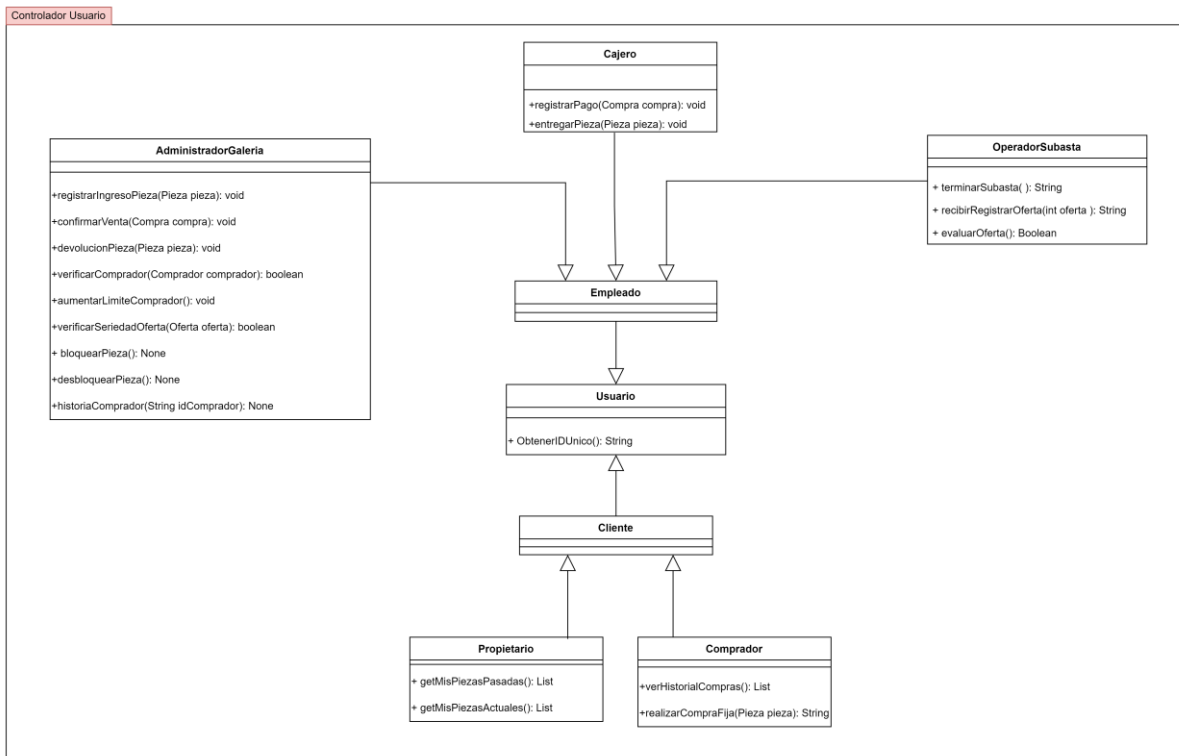
### 3.1.2 Controlador de compra y subasta de piezas

Por las diferencias y similitudes entre una compra y una subasta, se vio necesario crear dos clases aparte (compra y subasta) porque tomamos la decisión de manejarlas por aparte, ya que una subasta solo se convierte en una compra si se logra termina la subasta, es decir, no toda subasta es una compra. A continuación, se ve la descomposición del componente inicial con las responsabilidades divididas entre subasta y compra.



### 3.1.3 Controlador Usuarios

Considerando que tanto el vendedor como el comprador pueden suplir esta funcionalidad de ser clientes, no basta sencillamente con una separación de las clases sin mantener relación entre ellas. Por eso hay que desglosar la herencia que se puede generar a una misma clase de clientes y de empleados. Dividimos en esta iteración una superclase que se llama usuario de la cual tanto cliente como empleado heredan características comunes como lo son el login y el password, así mismo como la funcionalidad de conseguir sus propios identificadores únicos. De estas subclases de usuario esta cliente de la cual heredan los dos roles de propietario y comprador cada uno con sus respectivos métodos y datos importantes; y de la subclase de empleado se despliegan (extend) los tres roles de Administrador, Operador y Cajero los cuales tienen en su mayoría los métodos que permiten funcionar la aplicación.



## 4. Diseño Final

Finalmente, según las relaciones identificadas al definir las colaboraciones en el nivel 1 y contemplar la descomposición en clases efectuada por componente, se obtiene el siguiente diseño final para implementar la solución a la galería y casa de subastas. En este caso, uno de los grandes cambios fue la creación de una clase como Controlador Usuarios para permitir la construcción de usuarios y el manejo de la información a la que solo puede acceder una clase específica. Asimismo, se puede observar el diseño de alto nivel para entender mejor el relacionamiento entre clases. Para mejor calidad en las imágenes verlas en el formato pdf que se encuentra dentro de la carpeta en GitHub.

Diagrama de clases detallado:



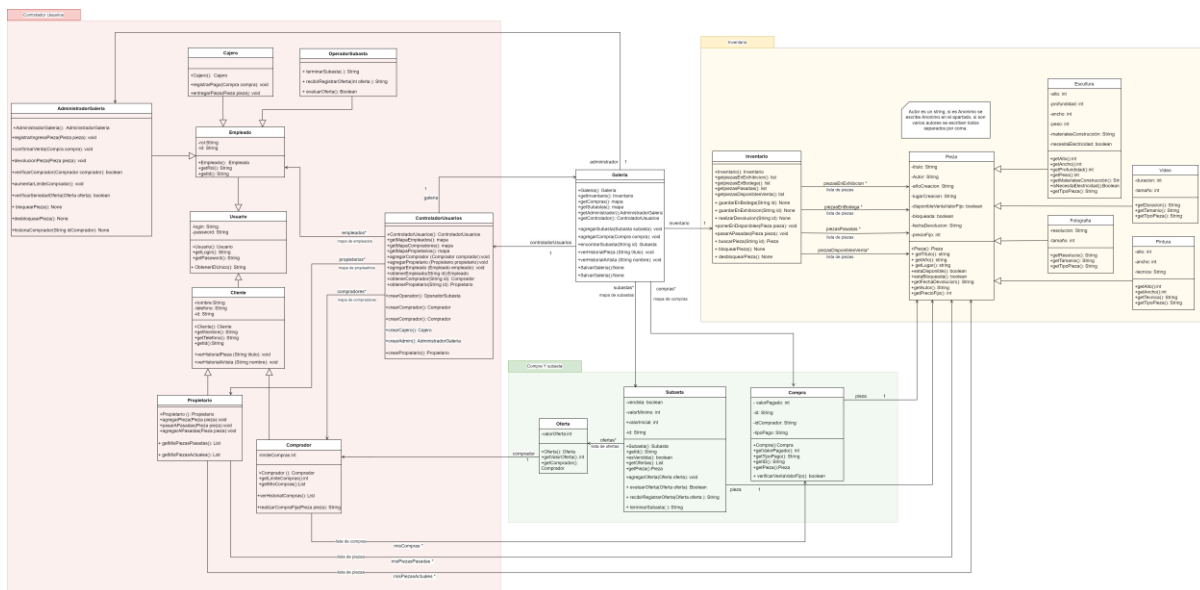
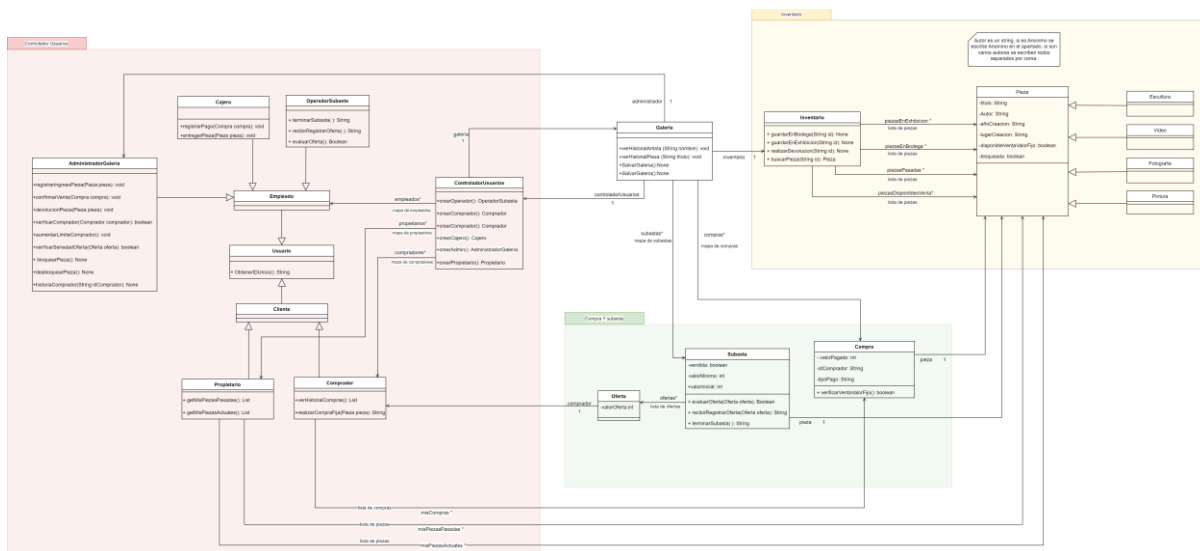
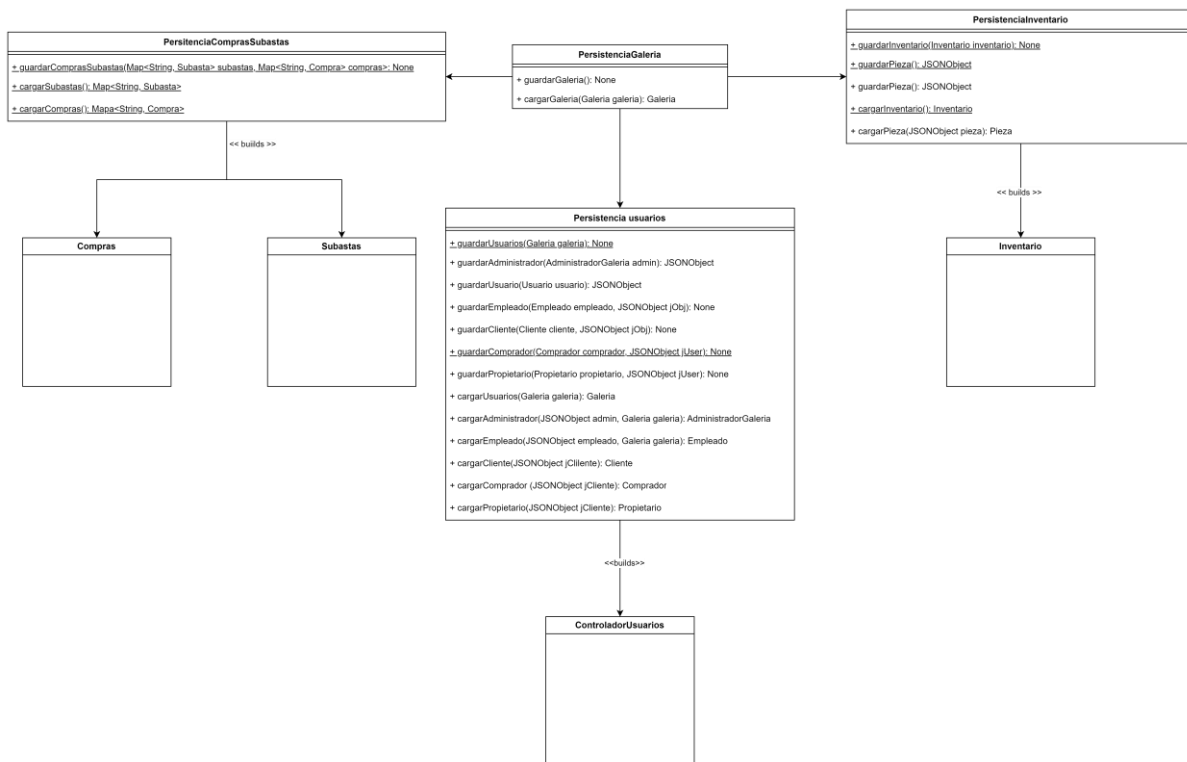


Diagrama de clases de alto nivel:



## 5. Persistencia

Para la persistencia de la galería se decidió dividirla en tres partes: un archivo con el inventario, otro con los usuarios y un último archivo con las compras y subastas. En inventario se guarda la información de todas las piezas y luego en el resto de las clases que tengan referencia a una pieza se guarda solo el inventario. Para cargar los archivos se hace el proceso inverso, se empieza por inventario para tener la información de las piezas y luego se empiezan a cargar los demás archivos.



## 6. Historias de Usuario

A continuación, se presentan una serie de historias de usuario para el desarrollo de un sistema de gestión de una galería y casa de subastas. Estas historias de usuario están diseñadas para ver las funcionalidades y características que deben implementarse en el sistema, desde la perspectiva de los diferentes tipos de usuarios involucrados, es decir, el administrador del sistema, los empleados de la galería y los compradores.

Cada historia de usuario describe una tarea o funcionalidad específica que un usuario desea realizar en el sistema, seguida de los criterios de entrada y salida/resultados esperados. Estas historias de usuario se han creado con el objetivo de guiar el proceso de desarrollo del sistema, asegurando que se cumplan las necesidades y requisitos de los usuarios finales.

### 6.1 Administrador de la Galería

A continuación, las historias de usuario para los 11 requerimientos del Administrador:

<b>1</b>	<b>COMO...</b>	Administrador de la galería
	<b>QUIERO...</b>	poder registrar nuevas piezas al inventario de la galería
	<b>PARA...</b>	mantener actualizado el catálogo de la galería y ofrecer una variedad de obras a los compradores.
<b>CONDICIONES</b>		
	<b>ENTRADA:</b>	Una pieza, con título, autor, año creación, lugar creación, si está disponible, fecha devolución y si está bloqueada
	<b>SALIDA:</b>	True si se agregó al inventario, False en caso contrario

2	COMO...	Administrador de la galería
	QUIERO...	poder confirmar la realización de una venta
	PARA...	mantener un registro preciso de las transacciones, el estado del inventario y así entregar la pieza al comprador.
CONDICIONES		
	ENTRADA:	Una compra, una pieza, y el id del comprador
	SALIDA:	"Venta confirmada" si se logró la venta, "Venta no confirmada" en caso contrario.

3	COMO...	Administrador de la galería
	QUIERO...	poder realizar devoluciones de piezas
	PARA...	mantener el inventario actualizado.
CONDICIONES		
	ENTRADA:	Una pieza, id del propietario que va a entregar la pieza
	SALIDA:	True si se confirmó la devolución, False en caso contrario

4	COMO...	Administrador de la galería
	QUIERO...	quiero poder verificar a nuevos compradores
	PARA...	para la existencia de la persona y prevenir fraudes.
CONDICIONES		
	ENTRADA:	id del comprador
	SALIDA:	True si el comprador existe, False en caso contrario

5	COMO...	Administrador de la galería
	QUIERO...	poder aumentar el límite de compra de un comprador
	PARA...	para permitir que el comprador pueda realizar más compras si tiene como pagarlas.
CONDICIONES		
	ENTRADA:	id del comprador y un número que represente en cuanto se quiere aumentar el limite

<b>SALIDA:</b>	True si se realizó el aumento, False en caso contrario
----------------	--

6	<b>COMO...</b>	Administrador de la galería
	<b>QUIERO...</b>	poder verificar la seriedad de una oferta
	<b>PARA...</b>	para garantizar que la oferta por valor fijo no supere el crédito del comprador.
<b>CONDICIONES</b>		
	<b>ENTRADA:</b>	id del comprador y el valor de la oferta a realizar
	<b>SALIDA:</b>	"El comprador no está registrado" o "El comprador no tiene suficiente crédito" o "Oferta seria"

7	<b>COMO...</b>	Administrador de la galería
	<b>QUIERO...</b>	poder bloquear una pieza
	<b>PARA...</b>	para asegurar que nadie más pueda comprarla mientras se verifica el proceso de compra.
<b>CONDICIONES</b>		
	<b>ENTRADA:</b>	título de la pieza
	<b>SALIDA:</b>	True si se realizó el bloqueo, False en caso contrario

8	<b>COMO...</b>	Administrador de la galería
	<b>QUIERO...</b>	poder desbloquear una pieza
	<b>PARA...</b>	que en caso de que no se confirme la venta esta esté de nuevo disponible para su venta.
<b>CONDICIONES</b>		
	<b>ENTRADA:</b>	título de la pieza
	<b>SALIDA:</b>	True si se realizó el desbloqueo, False en caso contrario

9	<b>COMO...</b>	Administrador de la galería
	<b>QUIERO...</b>	poder ver la historia de una pieza
	<b>PARA...</b>	resolver disputas o responder preguntas de los propietarios o compradores sobre el historial de la obra.

CONDICIONES	
<b>ENTRADA:</b>	título de la pieza
<b>SALIDA:</b>	Imprime el historial de la pieza si la pieza existe, "Pieza no existe" en caso contrario

10	<b>COMO...</b>	Administrador de la galería
	<b>QUIERO...</b>	poder ver la historia de un artista (autor)
	<b>PARA...</b>	evaluar la popularidad y la demanda del trabajo de ese artista en la galería.
CONDICIONES		
	<b>ENTRADA:</b>	Nombre del artista
	<b>SALIDA:</b>	Imprime el historial del artista si el artista existe, "No hay piezas con ese autor en la galería" en caso contrario

11	<b>COMO...</b>	Administrador de la galería
	<b>QUIERO...</b>	poder ver la historia de un comprador
	<b>PARA...</b>	comprender el comportamiento de compra y las preferencias de los clientes.
CONDICIONES		
	<b>ENTRADA:</b>	id del comprador
	<b>SALIDA:</b>	Imprime el historial del comprador si el comprador existe, "Comprador no existe" en caso contrario

## 6.2 Cliente

A continuación, las historias de usuario para los 6 requerimientos de los clientes:

1	<b>COMO...</b>	Comprador/Propietario
	<b>QUIERO...</b>	ver el historial de mis compras
	<b>PARA...</b>	para realizar un seguimiento de mis adquisiciones y planificar futuras compras.
CONDICIONES		
	<b>ENTRADA:</b>	Nada
	<b>SALIDA:</b>	Imprime todos los títulos de las piezas y el valor que se pagó por cada una.

2	<b>COMO...</b>	Comprador/Propietario
	<b>QUIERO...</b>	realizar la compra de una pieza que se pueda vender por un valor fijo

<b>PARA...</b>	adquirir obras de arte que me interesen.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	Una pieza
<b>SALIDA:</b>	"Compra realizada" o "Compra no realizada pues la pieza no estaba disponible para ser vendida por compra fija o el precio de la pieza supera tu límite de compras"

<b>3 COMO...</b>	Comprador/Propietario
<b>QUIERO...</b>	ver las piezas que poseo actualmente
<b>PARA...</b>	para estar al tanto de su ubicación y estado en la galería.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	Nada
<b>SALIDA:</b>	El nombre de todas las piezas que posee el cliente actualmente

<b>4 COMO...</b>	Comprador/Propietario
<b>QUIERO...</b>	ver las piezas que tuve en el pasado
<b>PARA...</b>	para estar al tanto de su ubicación y estado en la galería.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	Nada
<b>SALIDA:</b>	El nombre de todas las piezas que fueron del cliente

<b>5 COMO...</b>	Comprador/Propietario
<b>QUIERO...</b>	poder ver la historia de una pieza
<b>PARA...</b>	para evaluar su historial de ventas y determinar su demanda y popularidad entre otros coleccionistas.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	título de la pieza
<b>SALIDA:</b>	Imprime el historial de la pieza si la pieza existe, "Pieza no existe" en caso contrario

<b>6 COMO...</b>	Comprador/Propietario
<b>QUIERO...</b>	poder ver la historia de un artista (autor)

<b>PARA...</b>	para conocer su estilo y trayectoria, y tomar decisiones informadas sobre qué obras adquirir.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	Nombre del artista
<b>SALIDA:</b>	Imprime el historial del artista si el artista existe, "No hay piezas con ese autor en la galería" en caso contrario

### 6.3 Empleado

A continuación, las historias de usuario para los 7 requerimientos del Empleado:

<b>1</b>	<b>COMO...</b>	Empleado
	<b>QUIERO...</b>	poder ver la historia de un artista (autor)
	<b>PARA...</b>	evaluar la popularidad y la demanda del trabajo de ese artista en la galería.
	<b>CONDICIONES</b>	
	<b>ENTRADA:</b>	Nombre del artista
	<b>SALIDA:</b>	Imprime el historial del artista si el artista existe, "No hay piezas con ese autor en la galería" en caso contrario

<b>2</b>	<b>COMO...</b>	Empleado
	<b>QUIERO...</b>	poder ver la historia de una pieza
	<b>PARA...</b>	resolver disputas o responder preguntas de los propietarios o compradores sobre el historial de la obra.
	<b>CONDICIONES</b>	
	<b>ENTRADA:</b>	título de la pieza
	<b>SALIDA:</b>	Imprime el historial de la pieza si la pieza existe, "Pieza no existe" en caso contrario

<b>3</b>	<b>COMO...</b>	Empleado
	<b>QUIERO...</b>	poder registrar un pago
	<b>PARA...</b>	para confirmar que se ha completado una transacción y actualizar el estado de la pieza en el inventario.
	<b>CONDICIONES</b>	
	<b>ENTRADA:</b>	Una compra, una pieza, y el id del comprador
	<b>SALIDA:</b>	"Pago registrado"

<b>4</b>	<b>COMO...</b>	Empleado
----------	----------------	----------

<b>QUIERO...</b>	poder entregar una pieza a su nuevo propietario
<b>PARA...</b>	garantizar una entrega segura que actualice el estado del inventario.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	Una pieza, y el id del comprador
<b>SALIDA:</b>	True si se entregó la pieza al, comprador False en caso contrario

<b>5 COMO...</b>	Empleado
<b>QUIERO...</b>	poder recibir y registrar una oferta de subasta
<b>PARA...</b>	para iniciar el proceso de puja y mantener un registro de las ofertas recibidas.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	Oferta, id del comprador
<b>SALIDA:</b>	"La oferta se agregó correctamente a la subasta" o "La oferta no supero el valor inicial y no pudo ser registrada"

<b>6 COMO...</b>	Empleado
<b>QUIERO...</b>	poder evaluar una oferta de subasta
<b>PARA...</b>	determinar si cumple con los requisitos establecidos y si es válida para continuar con el proceso de subasta.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	Oferta, id del comprador
<b>SALIDA:</b>	True si la oferta supera el valor inicial, False en caso contrario

<b>7 COMO...</b>	Empleado
<b>QUIERO...</b>	poder terminar una subasta
<b>PARA...</b>	para cerrar la transacción y proceder con la entrega de la pieza al comprador ganador.
<b>CONDICIONES</b>	
<b>ENTRADA:</b>	id de la subasta
<b>SALIDA:</b>	"Se termino la subasta con id:" o "La subasta no se terminó dado que no había ofertas que cumplieran con el valor mínimo"

7.

**Pruebas**

**Unitarias:**

Implementación:



## Organización de Pruebas:

Las pruebas unitarias se han organizado en una única clase nombrada "Pruebas.java". Cada archivo contiene múltiples clases de prueba, cada una se centra en probar parte específica del sistema, como la gestión de usuarios, la galería, las subastas, las compras, entre otros.

## Estructura de Clases de Prueba:

### Importaciones:

Se importan las clases necesarias para realizar las pruebas, donde consideramos JUnit5 la mejor opción y la utilización de los distintos assert e importaciones dentro del mismo proyecto.

### Variables Estáticas:

Se definen variables estáticas que representan instancias de objetos utilizados en las pruebas, para inicializar un funcionamiento completo y correcto de las responsabilidades de la galería.

### SetUp:

Se define un método setUp() que se ejecuta antes de cada prueba y se encarga de inicializar el estado necesario para las pruebas, es decir condensa todas las variables estáticas en una sola galería

### Métodos de Prueba:

Se definen métodos de prueba precedidos por la anotación @Test, los cuales contienen la lógica de las pruebas. Cada método de prueba se enfoca en verificar un comportamiento específico del sistema y una sola Responsabilidad.

### TearDown:

Al final de todas las pruebas, se define un método tearDown() que se encarga de limpiar el estado del sistema, liberando los recursos utilizados durante las pruebas. Se hace global para que, si se desean asignar nuevas variables estáticas, sea sencillo.

### Selección de Asertos:

Para verificar el comportamiento del sistema, se utilizan diferentes métodos de aserción proporcionados por la clase org.junit.jupiter.api.Assertions. Estos métodos incluyen assertEquals, assertTrue, y assertFalse, entre otros, y se utilizan según la necesidad de cada prueba para comparar valores esperados con valores obtenidos.

### Uso de Anotaciones:

Se utilizan anotaciones proporcionadas por JUnit, como @BeforeAll y @AfterAll, para definir métodos que se ejecutan antes y después de todas las pruebas en la clase de prueba, respectivamente. También se utilizan anotaciones @Test para identificar los métodos que contienen pruebas unitarias.

### Conclusiones:

La implementación de pruebas unitarias utilizando el framework JUnit en el sistema de gestión de una galería de arte proporciona una sólida base para garantizar la calidad y fiabilidad del software. Estas pruebas permiten identificar y corregir errores de manera temprana en el ciclo de desarrollo, lo que contribuye a la creación de un sistema robusto y estable. Y usando el método TTD pudimos implementar con facilidad los nuevos requerimientos pedidos en esta entrega.