

# **Proyecto 2 DPOO: Revisión del programa para el manejo de una galería de arte**

Adrian Velasquez 202222737

Jerónimo Vásquez Ponce 202223824

Jhonny Armando Hortua 202111749

## Tabla de contenido

Descripción del proyecto .....	4
Funciones del programa.....	4
Inventario de piezas .....	4
Compra y subasta de piezas.....	4
Propietarios .....	5
Compradores y pagos .....	5
Perfiles de usuario .....	5
Requerimientos del proyecto .....	6
Casos de uso base .....	6
Requerimientos funcionales .....	7
Requerimientos del inventario.....	7
Requerimientos de la compra y subasta de piezas.....	9
Requerimientos propietarios .....	10
Requerimientos compradores y pagos .....	12
Otros requerimientos funcionales .....	13
Requerimientos no funcionales .....	14
Diagramas, modelo y restricciones.....	15
Programas de prueba .....	16
Etapas de la solución .....	17
Etapa 1: Diseño inicial.....	17
Etapa 2: Responsabilidades del programa .....	20
Controlador .....	20
Persistencia .....	21

Galeria .....	21
Inventario .....	21
Usuarios .....	21
Piezas .....	22
PortalPagos .....	22
Etapas 3: Diseño final e implementación .....	22

## Descripción del proyecto

El proyecto busca integrar 3 funciones principales en un programa para el manejo y monitoreo de las operaciones de una galería de arte y una casa de subasta. A continuación, se describirán cada una de estas funciones y cómo se relacionan al programa principal.

## Funciones del programa

### Inventario de piezas

El inventario tiene como objetivo almacenar y manejar las piezas de la galería. Para esto, se busca guardar la información pertinente de cada pieza de la galería, incluyendo su tipo y demás atributos determinados en torno al tipo de la pieza. Hay atributos que comparten todas las piezas, siendo el título, el año, el lugar de creación y el/los autores. Los propietarios pueden consignarse a la galería y se retirarán cuando se acabe o algún comprador adquiera la pieza. El programa tendrá un historial de las piezas que hayan estado en la galería, y el único que puede registrar el ingreso, confirmar una venta y confirmar una devolución de una pieza es el administrador de la galería. Para información más detallada acerca de los requerimientos de esta función, diríjase a la sección de [requerimientos del inventario](#).

### Compra y subasta de piezas

La compra y subasta de piezas tiene como objetivo permitir que alguien compre una pieza si se encuentra disponible, esta debe ser verificada y si se encuentra con un valor fijo o si está en subasta, en cualquier caso, un usuario registrado en la casa usuario corriente y con los atributos de verificación, piezas actuales, piezas pasadas e historial de pagos, como comprador podrá realizar la adquisición de cualquiera de los objetos, en este caso la pieza quedara bloqueada para el resto de usuarios hasta que el administrador verifique que el usuario sea real y la validez del pago, en tal caso la piza quedara vendida de lo contrario seguirá en venta o subasta. También se cuenta con venta de piezas en bodega, no solo las que se encuentran en exhibición. En la subasta de piezas estas contarán con un valor inicial de venta, y para que esta sea realizada el comprador

debe ofrecer el valor mínimo estipulado para comprarla, el valor inicial de las piezas no será mencionado con anterioridad hasta el inicio de la subasta y solo podrán ser partícipes aquellos que estén verificados anteriormente por el administrador de la galería

## Propietarios

El propietario tiene como función gestionar la información y las interacciones relacionadas con los propietarios de las piezas de arte que se exhiben, venden o subastan en la galería. Esto incluye el registro de nuevos propietarios, la verificación de identidad, el seguimiento de estado de sus piezas y la comunicación entre galería y propietario. Este comparte atributos con acciones del usuario.

## Compradores y pagos

Esta función se encarga de gestionar las transacciones financieras realizadas por los compradores para adquirir las piezas de arte disponibles en la galería. Esto incluye el registro y la verificación de los compradores, la autorización, el registro de los pagos y entrega de las piezas. Comparte atributos de usuario, propietario y de inventario de las piezas.

## Perfiles de usuario

A continuación, se pueden consultar los diferentes tipos de usuarios del programa y sus roles correspondientes. Bajo estas definiciones se van a plantear los requerimientos del programa.

Rol del usuario	Administrador de la galería y de subasta
Descripción del usuario	El/los administradores de la galería son aquellos usuarios que tienen la autorización de registrar las piezas que entran y salen de la galería, al igual que autorizar, verificar y alterar los montos máximos de compra de los demás usuarios compradores y propietarios. También tiene autorización de realizar las demás funciones de los empleados.

Rol del usuario	Cajero de la galería
Descripción del usuario	El/los cajeros de la galería son aquellos usuarios que tiene la autorización de registrar las transacciones de la galería junto a las demás funciones de los empleados.

Rol del usuario	Empleado de la galería
-----------------	------------------------

<b>Descripción del usuario</b>	Los empleados de la galería son aquellos que tienen la autorización de llevar a cabo las funciones básicas del programa en cuanto al manejo de las piezas de la galería. El/los cajeros y el/los administradores son considerados como empleados, y pueden ejercer las funciones correspondientes.
--------------------------------	--

<b>Rol del usuario</b>	Propietario
<b>Descripción del usuario</b>	Un propietario tiene dos estados posibles: verificado y no-verificado. Un propietario no verificado no tiene autorización hasta que el administrador lo verifique. El propietario verificado de una pieza tiene la autorización de consignar una pieza de su colección a la galería.

<b>Rol del usuario</b>	Comprador
<b>Descripción del usuario</b>	Un comprador tiene dos estados posibles: verificado y no-verificado. Un comprador no verificado tiene la autorización de realizar una oferta para la compra de una pieza, pero no tiene autorización de participar en una subasta hasta que sea verificado por el administrador. Un comprador verificado tiene la autorización de comprar piezas de la galería y realizar ofertas durante la subasta de una pieza. Si la oferta de un comprador se sale del valor máximo propuesto por el administrador a la hora de haber sido verificado, el administrador podrá cambiar o no dicho valor máximo.

## Requerimientos del proyecto

El programa tiene una serie de requerimientos los cuales se busca sean cumplidos en su mayoría o, de ser posible, en su totalidad. Estos requerimientos corresponden a las necesidades de los usuarios, al igual que la descripción de las características propuestas en la sección de las [funciones del programa](#).

### Casos de uso base

En la [ilustración 1](#) se pueden observar los casos de uso base que fueron identificados en el enunciado del problema. Con base en estos casos de uso y las restricciones del proyecto fueron planteados los requerimientos funcionales del programa. Asimismo, cabe resaltar que, tal como se describe en la sección de [perfiles de usuario](#), los empleados cuyo rol es de administrador o de cajero son capaces de realizar todas las tareas de un empleado corriente, y que a su vez los usuarios corrientes pueden ser compradores y vendedores a la vez (una vez sean verificados adecuadamente por el administrador de la galería).

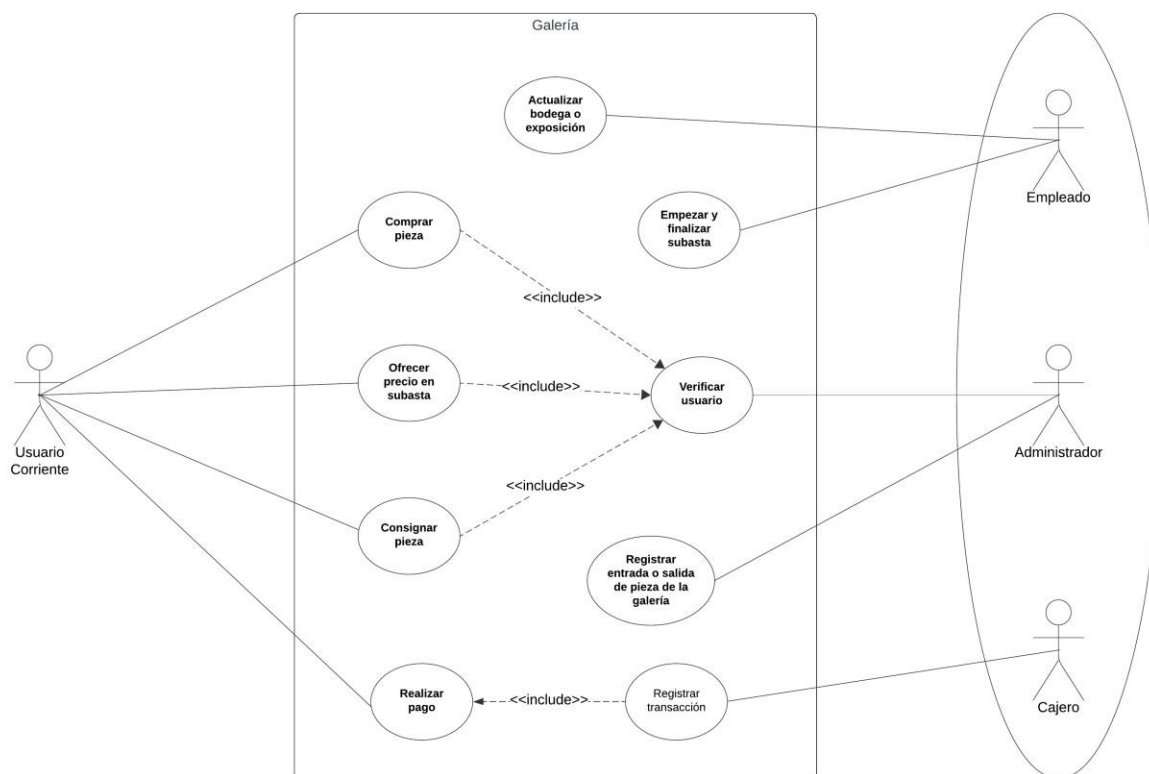


Ilustración 1: Casos de uso base de la aplicación

## Requerimientos funcionales

Los requerimientos funcionales son aquellos que son necesarios para el buen funcionamiento del programa, y aquellas características explícitas que permiten que los usuarios puedan usar el programa como se espera y como se describe en el [diagrama de caso de uso](#).

### Requerimientos del inventario

Req. Inventario	1
Usuario(s)	Propietario
Título	Consignar pieza
Descripción	Otrogarle a la galería una pieza por un tiempo específico.
Datos de entrada	Pieza para consignar, tiempo de consignación
Datos de salida	-

Req. Inventario	2
Usuario(s)	Administrador

<b>Título</b>	Registrar pieza
<b>Descripción</b>	Registrar una nueva pieza en el inventario de la galería
<b>Datos de entrada</b>	Pieza para registrar, tiempo de consignación
<b>Datos de salida</b>	-

<b>Req. Inventario</b>	3
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Exponer pieza
<b>Descripción</b>	Exponer una pieza en la galería
<b>Datos de entrada</b>	Pieza para exponer
<b>Datos de salida</b>	-

<b>Req. Inventario</b>	4
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Almacenar pieza
<b>Descripción</b>	Almacenar una pieza en la bodega de la galería
<b>Datos de entrada</b>	Pieza para almacenar
<b>Datos de salida</b>	-

<b>Req. Inventario</b>	5
<b>Usuario(s)</b>	Todos los usuarios
<b>Título</b>	Verificar estado de una pieza
<b>Descripción</b>	Verificar si una pieza está o no en el registro de la galería. Si lo está, verifica si fue vendida, si está expuesta, si está almacenada, si está bloqueada o si está en una subasta.
<b>Datos de entrada</b>	Nombre de la pieza
<b>Datos de salida</b>	Estado de la pieza buscada

<b>Req. Inventario</b>	6
<b>Usuario(s)</b>	Administrador
<b>Título</b>	Registrar venta
<b>Descripción</b>	Registra la venta de una pieza de la galería
<b>Datos de entrada</b>	Transacción de la venta
<b>Datos de salida</b>	-

<b>Req. Inventario</b>	7
------------------------	---



<b>Usuario(s)</b>	Administrador
<b>Título</b>	Registrar devolución
<b>Descripción</b>	Registra la devolución de una pieza de la galería
<b>Datos de entrada</b>	Nombre de la pieza
<b>Datos de salida</b>	-

<b>Req. Inventario</b>	8
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Actualizar estado de una pieza
<b>Descripción</b>	Actualiza el estado de una pieza, excepto si esta fue vendida o devuelta al propietario.
<b>Datos de entrada</b>	Nombre de la pieza, actualización
<b>Datos de salida</b>	-

<b>Req. Inventario</b>	9
<b>Usuario(s)</b>	Todos
<b>Título</b>	Consultar historial de una pieza y de un autor
<b>Descripción</b>	Consulta el historial de una pieza o un autor
<b>Datos de entrada</b>	Nombre de la pieza o nombre del autor
<b>Datos de salida</b>	Informacion de la pieza y el autor

### Requerimientos de la compra y subasta de piezas

<b>Req. Compra y subasta</b>	1
<b>Usuario(s)</b>	Comprador
<b>Título</b>	Ofrecer por una pieza
<b>Descripción</b>	Hacer una oferta de dinero por una pieza en una subasta
<b>Datos de entrada</b>	Valor de la oferta
<b>Datos de salida</b>	-

<b>Req. Compra y subasta</b>	2
<b>Usuario(s)</b>	Comprador
<b>Título</b>	Realizar el pago de una pieza
<b>Descripción</b>	Realizar el pago de la pieza seleccionada
<b>Datos de entrada</b>	Valor de la oferta

<b>Datos de salida</b>	-
------------------------	---

<b>Req. Compra y subasta</b>	3
<b>Usuario(s)</b>	Administrador
<b>Título</b>	Confirmar autenticidad del pago y la cuenta
<b>Descripción</b>	Verificar la severidad de la oferta y la autenticidad de la cuenta
<b>Datos de entrada</b>	Valor de la oferta
<b>Datos de salida</b>	-

<b>Req. Compra y subasta</b>	4
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Bloquear pieza
<b>Descripción</b>	Bloquear venta de pieza mientras se verifican datos de transacción
<b>Datos de entrada</b>	Nombre de la pieza
<b>Datos de salida</b>	-

<b>Req. Compra y subasta</b>	5
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Descontinuar pieza
<b>Descripción</b>	Descontinuar pieza después de una venta
<b>Datos de entrada</b>	Nombre de la pieza
<b>Datos de salida</b>	-

<b>Req. Compra y subasta</b>	6
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Activar subasta
<b>Descripción</b>	Activar la subasta de la pieza con el valor mínimo
<b>Datos de entrada</b>	Nombre de la pieza, Valor de la pieza, Fecha y hora de la subasta
<b>Datos de salida</b>	-

## Requerimientos propietarios

<b>Req. Propietarios</b>	1
<b>Usuario(s)</b>	Administrador
<b>Título</b>	Registrar propietario

<b>Descripción</b>	Permite al administrador registrar un nuevo propietario al sistema
<b>Datos de entrada</b>	Información del propietario
<b>Datos de salida</b>	Confirmación de registro propietario e identificador propietario

<b>Req. Propietarios</b>	2
<b>Usuario(s)</b>	Administrador
<b>Título</b>	Verificar propietario
<b>Descripción</b>	Permite al administrador verificar la identidad del propietario registrado
<b>Datos de entrada</b>	Identificación del propietario
<b>Datos de salida</b>	Confirmación verificación exitosa propietario

<b>Req. Propietarios</b>	3
<b>Usuario(s)</b>	Propietario
<b>Título</b>	Información propietario
<b>Descripción</b>	Permite al propietario consultar su información en sistema
<b>Datos de entrada</b>	Identificación del propietario
<b>Datos de salida</b>	Información del propietario

<b>Req. Propietarios</b>	4
<b>Usuario(s)</b>	Propietario
<b>Título</b>	Actualizar información propietario
<b>Descripción</b>	Permite al propietario actualizar su información registrada en el sistema
<b>Datos de entrada</b>	Nueva información del propietario
<b>Datos de salida</b>	Confirmación actualización exitosa de información del propietario

<b>Req. Propietarios</b>	5
<b>Usuario(s)</b>	Administrador
<b>Título</b>	Eliminar propietario
<b>Descripción</b>	Permite al administrador eliminar un propietario registrado en el sistema
<b>Datos de entrada</b>	Identificación del propietario a eliminar
<b>Datos de salida</b>	Confirmación eliminación exitosa del propietario

<b>Req. Propietarios</b>	6
<b>Usuario(s)</b>	Propietario
<b>Título</b>	Consultar estado de la pieza del propietario

<b>Descripción</b>	Permite al propietario consultar el estado de las piezas que tiene en la galería
<b>Datos de entrada</b>	Identificación del propietario
<b>Datos de salida</b>	Estado de las piezas del propietario

## Requerimientos compradores y pagos

<b>Req. Compradores y pagos</b>	1
<b>Usuario(s)</b>	administrador
<b>Título</b>	Registrar comprador
<b>Descripción</b>	Registrar un nuevo comprador al sistema
<b>Datos de entrada</b>	Datos comprador
<b>Datos de salida</b>	Confirmación registro exitoso e identificador comprador

<b>Req. Compradores y pagos</b>	2
<b>Usuario(s)</b>	administrador
<b>Título</b>	Verificar comprador
<b>Descripción</b>	Verifica la identidad del comprador antes de que realice compras o participe en subastas
<b>Datos de entrada</b>	Datos comprador
<b>Datos de salida</b>	Confirmación verificación exitosa y actualización estado comprador

<b>Req. Compradores y pagos</b>	3
<b>Usuario(s)</b>	cajero
<b>Título</b>	Registrar pago
<b>Descripción</b>	Registrar los pagos realizados por los compradores para las piezas adquiridas
<b>Datos de entrada</b>	Identificador pieza vendida, monto pago y método de pago
<b>Datos de salida</b>	Confirmación registro exitoso pago y actualización estado de la pieza

<b>Req. Compradores y pagos</b>	4
<b>Usuario(s)</b>	comprador
<b>Título</b>	Consultar historial de compras

<b>Descripción</b>	El comprador puede acceder a su historial de compras para ver las piezas que han adquirido en la galería
<b>Datos de entrada</b>	Identificador comprador
<b>Datos de salida</b>	Lista de piezas compradas

<b>Req. Compradores y pagos</b>	5
<b>Usuario(s)</b>	comprador
<b>Título</b>	Registrar estado pieza
<b>Descripción</b>	Permite al comprador ver el estado en el cual se encuentra la pieza (pago aceptado, en envío, entregada)
<b>Datos de entrada</b>	Datos comprador
<b>Datos de salida</b>	Estado de la pieza

<b>Req. Compradores y pagos</b>	6
<b>Usuario(s)</b>	cajero
<b>Título</b>	Devolución pago
<b>Descripción</b>	Registra la devolución de una pieza por parte del comprador
<b>Datos de entrada</b>	Identificador pieza, comprador y motivo devolución
<b>Datos de salida</b>	Confirmación registro exitoso de la devolución y actualización estado pieza

### Otros requerimientos funcionales

<b>Req. Otros</b>	1
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Salvar cambios
<b>Descripción</b>	Salvar los cambios realizados en el programa
<b>Datos de entrada</b>	-
<b>Datos de salida</b>	-

<b>Req. Otros</b>	2
<b>Usuario(s)</b>	Empleado
<b>Título</b>	Cargar datos
<b>Descripción</b>	Cargar los datos más recientes del programa
<b>Datos de entrada</b>	-
<b>Datos de salida</b>	-

<b>Req. Otros</b>	3
<b>Usuario(s)</b>	Todos los usuarios
<b>Título</b>	Iniciar sesión
<b>Descripción</b>	Inicia sesión al programa y muestra opciones de acuerdo con el tipo y rol del usuario
<b>Datos de entrada</b>	Login y contraseña
<b>Datos de salida</b>	-

<b>Req. Otros</b>	4
<b>Usuario(s)</b>	Todos los usuarios
<b>Título</b>	Crear cuenta
<b>Descripción</b>	Crea un nuevo usuario no verificado
<b>Datos de entrada</b>	Login y contraseña
<b>Datos de salida</b>	-

## Requerimientos no funcionales

Por otro lado, en cuanto a los requerimientos no funcionales, el programa debería tener en cuenta los siguientes atributos de calidad. En cuanto a estos requerimientos, se podría afirmar que es necesario que:

- El mismo cliente o los empleados de la galería pueden acceder a la información de los usuarios. Todas las consultas sobre las piezas deben hacerse directamente en la galería.
  - Atributos de calidad:
    - ☐ Seguridad
    - ☐ Privacidad
    - ☐ Eficiencia
- La aplicación sea persistente, y solamente los empleados pueden acceder y modificar la información guardada.
  - Atributos de calidad:
    - ☐ Seguridad
    - ☐ Fiabilidad
    - ☐ Buen rendimiento

- Debe haber un registro de todas las transacciones realizadas, al igual que los clientes, dueños, destinatarios, y propietarios de las piezas con el fin de asegurar un buen servicio.
  - Atributos de calidad
    - Integridad
    - Auditoría
    - Disponibilidad
- Solamente los usuarios verificados podrán vender y comprar piezas de arte en la galería. EL administrador de esta es el encargado de verificar a los usuarios y asegurarse de que estos puedan participar en la compra, venta y subasta de piezas.
  - Atributos de calidad:
    - Seguridad
    - Transparencia

## Diagramas, modelo y restricciones

En la [figura 1](#) se puede observar una representación general de la estructura del programa de la galería donde se evidencian las características y las relaciones que existen entre las entidades propuestas.

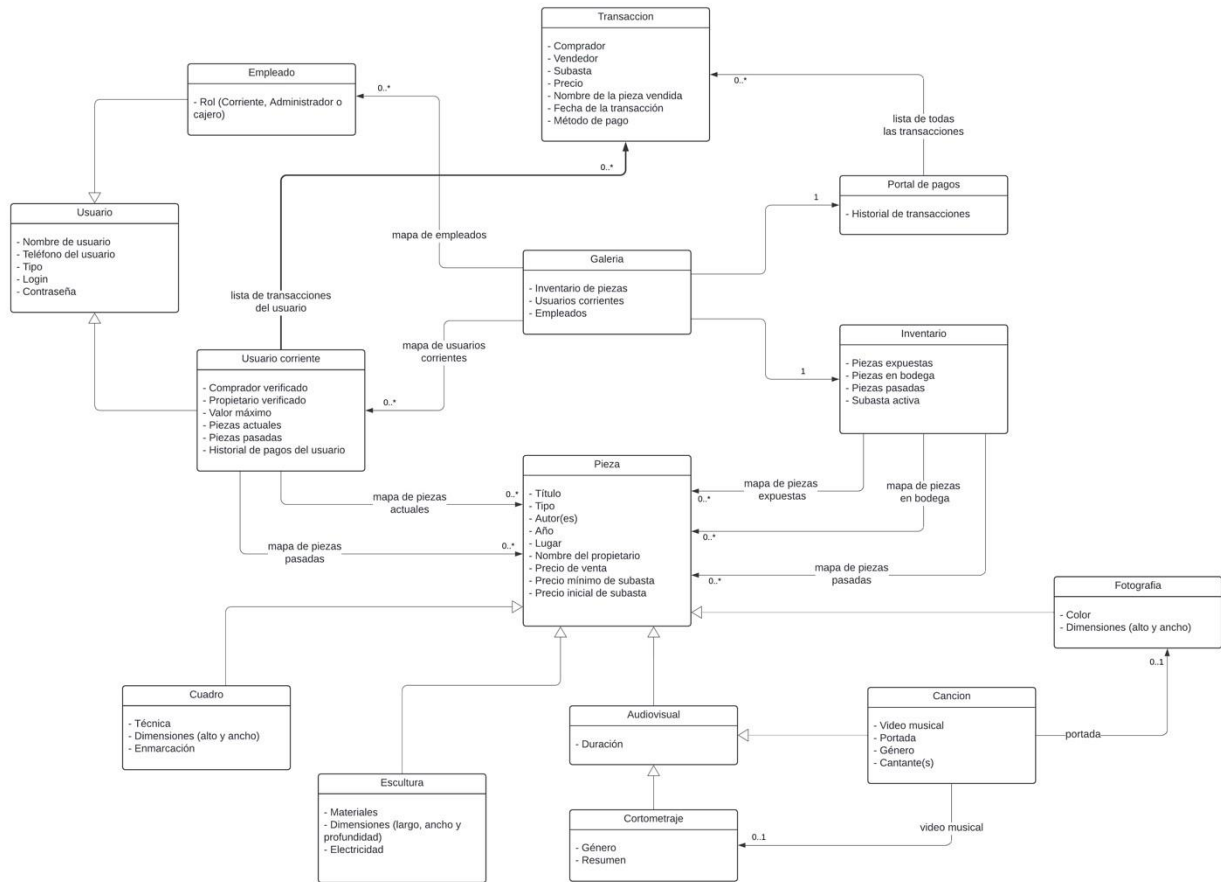


Figura 1: Diagrama de modelo del programa de la galería

En este planteamiento inicial, se tiene un par de restricciones en cuanto a la escalabilidad del programa. Esto se debe a la plataforma de desarrollo y la falta de experiencia en cuanto a las interfaces gráficas para los usuarios. Asimismo, se espera que haya cambios a la hora de realizar el diseño concreto del programa y su implementación.

## Programas de prueba

Como se ha mencionado anteriormente, en este proyecto fueron implementados los módulos de pruebas y de consola. En cuanto a las pruebas, estas están divididas en 4 archivos principales. El primero es una recopilación de todas las pruebas de los requerimientos del proyecto anterior implementada en Junit, mientras que las otras 3 están diseñadas para probar los nuevos requerimientos de este proyecto, los cuales se basan en la consulta de la información de las piezas y usuarios. Para la correcta implementación de estas pruebas, se tendrán en cuenta los [perfiles de usuario](#), [los casos de uso](#), al igual que las demás características y restricciones del



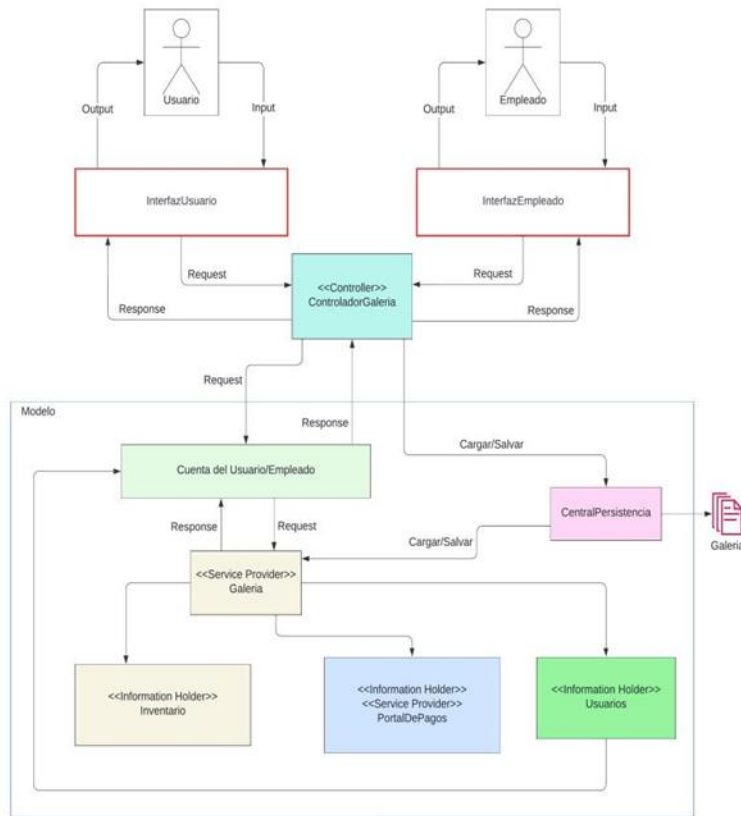
programa. Estas pruebas se encuentran en el paquete tests.pruebaUsuariosP2, y fueron implementados como Junit.

## Etapas de la solución

El diseño de la solución se divide en 3 etapas principales.

### Etapas 1: Diseño inicial

En primera instancia, una vez finalizado el análisis del problema, empezamos el diseño de los componentes de la aplicación. De ahí surgió el diagrama de la figura 1, en el cual se puede ver la forma en la que se espera que funcione el programa. En esencia, se espera que, por medio de las interfaces, tanto los usuarios como los empleados interactúen con el controlador de la galería. Las interfaces aseguran que los inputs de los usuarios sean apropiados y los envía en forma de requests al controlador. Una vez el request llega al controlador, se hace provecho del vínculo que existe con la galería y el usuario actual con el fin de llevar a cabo las tareas necesarias. Estas tareas se realizan a lo largo del modelo para cumplir los requerimientos del programa. En este proyecto se busca implementar el modelo y el controlador del programa, razón por la que las interfaces se reducirán a los programas de prueba para la aplicación y los requerimientos. Asimismo, los colores utilizados en este diseño inicial serán utilizados a lo largo de las demás etapas con el fin de conectar cada uno de los diseños propuestos.



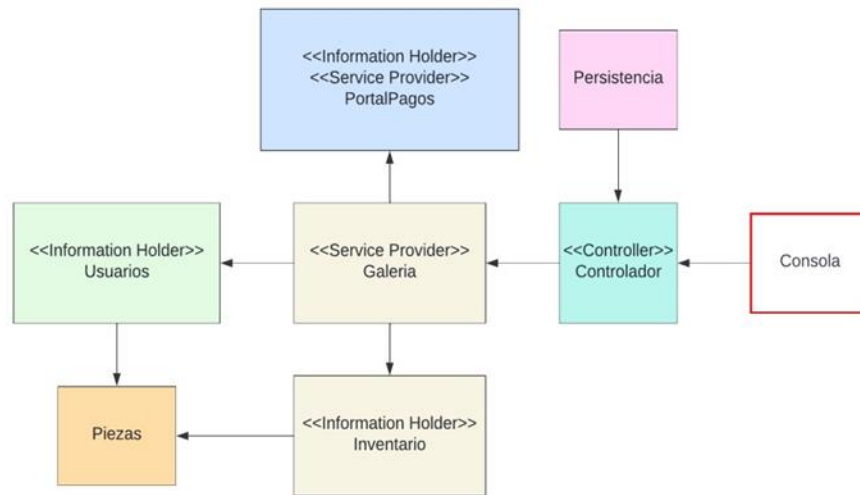
Como descrito anteriormente, los requests representan las instrucciones que el programa recibe para realizar cada uno de los requerimientos. En el análisis previo al diseño, se identificaron una serie de requerimientos funcionales y no funcionales, algunos de los cuales corresponden a los usuarios corrientes, y otros de los cuales corresponden a los empleados (muchos de estos solamente corresponden a un rol específico de empleado). Estos requerimientos generales se detallan a continuación:

Requerimiento	Descripción	Encargado
<b>Consignar pieza</b>	Consigna una pieza a la galería, debe ser aprobado por el administrador	Usuario corriente
<b>Comprar pieza</b>	Envía una solicitud de compra a la galería, debe ser aprobado por el administrador	Usuario corriente

<b>Realizar oferta</b>	Envía una nueva oferta a la subasta, debe estar verificado por el administrador	Usuario corriente
<b>Verificar usuario</b>	Verifica un usuario, ya sea para comprar, vender o subastar y ofrecer precios para la subasta de piezas	Administrador
<b>Registrar ingreso o salida de pieza</b>	Registra el ingreso o salida de una pieza a la galería	Administrador
<b>Cambiar monto máximo</b>	Modificar el monto máximo de un usuario	Administrador
<b>Registrar transacción</b>	Registrar una nueva transacción	Cajero
<b>Consultar historial pieza</b>	Consulta la información acerca de una pieza	Todos
<b>Consultar historial autor</b>	Consulta la información acerca de un autor	Todos
<b>Consultar historial vendedor/comprador</b>	Consulta la información de un vendedor/comprador	Administrador
<b>Empezar/finalizar subasta</b>	Empieza o finaliza una subasta de una pieza	Operador
<b>Registrar nueva oferta</b>	Registra una nueva oferta de un usuario	Operador
<b>Salvar/cargar datos</b>	Salva y carga los datos de la galería	-

## Etapa 2: Responsabilidades del programa

En la [figura 1](#) se pueden ver algunas de las responsabilidades definidas para el programa, a pesar de que todavía no se habían planteado las responsabilidades específicas de cada módulo. En esta segunda etapa del diseño se pretenden definir las relaciones puntuales que existen entre los módulos de la aplicación, al igual que las responsabilidades de cada uno. En la [figura 2](#) se muestra un diagrama UML de alto nivel con base en el cual se van a desarrollar las clases del programa, y en el cual se basará el diseño final de la aplicación.



Como se muestra en la [figura 2](#), los módulos principales del programa son los módulos de Controlador, Persistencia, Galeria, Inventario, PortalPagos, Usuarios y Piezas. A continuación, se explicarán las responsabilidades de cada módulo en detalle.

### Controlador

El controlador del programa es el encargado de manejar toda la información del programa y comunicarla a la consola. Por lo tanto, la responsabilidad principal de este módulo es cargar, salvar, y ser el puente entre el usuario y la lógica del programa. Los dos atributos principales del controlador serán la galería y la persistencia, a su vez apoyándose del usuario actual para desarrollar las tareas solicitadas.

## Persistencia

La única responsabilidad del módulo de persistencia es almacenar y acceder a la información guardada de la galería. Para cumplir esta tarea, se tuvieron 2 opciones en mente. La primera opción propuesta gira entorno a una serie de archivos JSON en los cuales se guarda la información del inventario, los usuarios corrientes, los empleados, las transacciones y demás atributos puntuales de la galería. La segunda opción es la serialización de las clases del programa con el fin de facilitar el almacenamiento y carga de datos. Para este proyecto decidimos utilizar la serialización y deserialización para la persistencia del programa con el fin de poder realizar la implementación en un tiempo oportuno<sup>[1]</sup>. Sin embargo, se espera que la transición entre la serialización y el uso de JSON sea sencilla para evitar posibles problemas a futuro.

## Galería

La galería es el módulo principal del programa, y es donde se encuentran todas las demás partes importantes como el inventario, los usuarios y las transacciones. La responsabilidad de la galería, por lo tanto, es conectar todos los demás módulos para garantizar cohesión en las tareas realizadas.

## Inventario

EL inventario es el módulo en el cual se tendrá acceso a todas las piezas registradas en la galería, lo cual es evidente al ser un Information Holder, y es el encargado de mantener la información al día. Esto se lleva a cabo por medio de los empleados, sin embargo, todas las consultas acerca de las piezas no se realizan en el inventario directamente, si no que primero se valida en la galería.

## Usuarios

El módulo de usuarios es el encargado de crear y mantener la información de los usuarios, al igual que ejecutar iniciar muchas de las tareas relacionadas a los requerimientos del programa. Asimismo, el programa debe tener por lo menos 4 tipos de usuarios distintos, pero en este diseño hay 5.

- UsuarioCorriente: Son los usuarios corrientes de la galería, los cuales pueden ser vistos como clientes. Como se describe en la Etapa 1, estos usuarios son capaces de realizar

pagos, ofrecer montos durante subastas, y consignar piezas a la galería. Para realizar estas tareas, el programa debe garantizar que el usuario esté verificado.

- Empleado: Son los usuarios capaces de realizar cambios en el inventario, al igual que realizar transacciones (estas transacciones no están registradas en el sistema, simplemente son realizadas). Las demás tareas son realizadas por usuarios con roles específicos.
  - a. Administrador: Es el usuario encargado de verificar a los usuarios corrientes, tanto para la compra y venta de piezas, como para la subasta.
  - b. Operador: Es el usuario encargado de manejar las subastas. Pueden empezarlas, terminarla y registrar las ofertas de los usuarios.
  - c. Cajero: Es el usuario encargado de registrar las transacciones realizadas en el sistema.

## Piezas

El módulo de piezas se encarga de definir las piezas y los diferentes tipos de piezas. Cada pieza tendrá información acerca de su propietario, al igual que una serie de atributos generales y específicos.

## PortalPagos

El portal de pagos es el módulo encargado de la generación, guardado y consulta de transacciones. En general, estas transacciones van a ser realizadas por los empleados, y solamente van a poder ser consultadas por ellos. Sin embargo, los usuarios también tendrán acceso a sus propias transacciones. Por esta razón, el portal de pagos es tanto un Information Holder como un Service Provider.

## Etapa 3: Diseño final e implementación

Una vez planteados los requerimientos y las responsabilidades de cada módulo, se planteó el diseño de la [figura 3](#). Como se puede observar, este diseño es acorde a lo planteado en las diferentes etapas del diseño, y está listo para la implementación. A pesar de que no hace parte del diagrama, en la implementación fueron realizadas una serie de pruebas cuyo objetivo es verificar el correcto funcionamiento de los componentes del programa. En primera instancia, se prueba la

**Objetivo do Sistema:** O sistema tem como objetivo gerenciar o ciclo de vida dos recursos humanos, desde a contratação até a aposentadoria, incluindo a gestão de salários, benefícios e desempenho.

**Atores e Permissões:**

- Administrador:** Possui todas as permissões do sistema.
- Gestor:** Possui permissões para gerenciar a equipe sob sua responsabilidade.
- Funcionário:** Possui permissões para consultar suas informações pessoais e profissionais.
- Recrutador:** Possui permissões para gerenciar o processo de recrutamento.

**Processos:**

1. **Contratação:** O recrutador cria uma vaga, publica a vaga, recebe currículos e realiza entrevistas.
2. **Seleção:** O gestor seleciona o candidato mais adequado para a vaga.
3. **Contratação:** O administrador aprova a contratação e emite o contrato de trabalho.
4. **Onboarding:** O novo funcionário é integrado à equipe e recebe treinamento.
5. **Gestão de Salário e Benefícios:** O sistema calcula o salário e os benefícios do funcionário.
6. **Avaliação de Desempenho:** O gestor avalia o desempenho do funcionário.
7. **Desligamento:** O funcionário é desligado do sistema e o processo de rescisão é iniciado.

**Diagrama de Fluxo de Processo:**

```

graph TD
    A[Contratação] --> B[Seleção]
    B --> C[Contratação]
    C --> D[Onboarding]
    D --> E[Gestão de Salário e Benefícios]
    E --> F[Avaliação de Desempenho]
    F --> G[Desligamento]
  
```

**Diagrama de Casos de Uso:**

```

graph TD
    UC1[Contratar Funcionário] --> UC2[Selecionar Funcionário]
    UC2 --> UC3[Contratar Funcionário]
    UC3 --> UC4[Onboarding]
    UC4 --> UC5[Gestão de Salário e Benefícios]
    UC5 --> UC6[Avaliação de Desempenho]
    UC6 --> UC7[Desligamento]
  
```

**Diagrama de Classes:**

```

classDiagram
    class Administrador
    class Gestor
    class Funcionario
    class Recrutador
    class Vaga
    class Salario
    class Beneficio
    class Avaliacao
    class Desligamento
  
```

**Diagrama de Atividade:**

```

graph TD
    A[Contratar Funcionário] --> B[Selecionar Funcionário]
    B --> C[Contratar Funcionário]
    C --> D[Onboarding]
    D --> E[Gestão de Salário e Benefícios]
    E --> F[Avaliação de Desempenho]
    F --> G[Desligamento]
  
```

**Diagrama de Sequência:**

```

sequenceDiagram
    participant A as Administrador
    participant B as Gestor
    participant C as Funcionario
    participant D as Recrutador
    A->>D: Criar Vaga
    D->>A: Publicar Vaga
    A->>B: Selecionar Funcionário
    B->>A: Contratar Funcionário
    A->>C: Onboarding
    C->>A: Gestão de Salário e Benefícios
    A->>B: Avaliação de Desempenho
    B->>A: Desligamento
  
```

**Diagrama de Estado:**

```

stateDiagram-v2
    [*] --> EstadoInativo
    EstadoInativo --> EstadoAtivo
    EstadoAtivo --> EstadoInativo
    EstadoAtivo --> EstadoDesligado
    EstadoDesligado --> EstadoInativo
  
```

**Diagrama de Componentes:**

```

graph TD
    C1[Componente 1] --> C2[Componente 2]
    C2 --> C3[Componente 3]
    C3 --> C4[Componente 4]
  
```

**Diagrama de Pacotes:**

```

graph TD
    P1[Package 1] --> P2[Package 2]
    P2 --> P3[Package 3]
    P3 --> P4[Package 4]
  
```

**Diagrama de Tarefas:**

```

graph TD
    T1[Tarefa 1] --> T2[Tarefa 2]
    T2 --> T3[Tarefa 3]
    T3 --> T4[Tarefa 4]
  
```

**Diagrama de Fluxo de Dados:**

```

graph TD
    F1[Fluxo 1] --> F2[Fluxo 2]
    F2 --> F3[Fluxo 3]
    F3 --> F4[Fluxo 4]
  
```

**Diagrama de Relações:**

```

graph TD
    R1[Relação 1] --> R2[Relação 2]
    R2 --> R3[Relação 3]
    R3 --> R4[Relação 4]
  
```

**Diagrama de Dependências:**

```

graph TD
    D1[Dependência 1] --> D2[Dependência 2]
    D2 --> D3[Dependência 3]
    D3 --> D4[Dependência 4]
  
```

**Diagrama de Hierarquia:**

```

graph TD
    H1[Hierarquia 1] --> H2[Hierarquia 2]
    H2 --> H3[Hierarquia 3]
    H3 --> H4[Hierarquia 4]
  
```

**Diagrama de Comunicação:**

```

graph TD
    C1[Comunicação 1] --> C2[Comunicação 2]
    C2 --> C3[Comunicação 3]
    C3 --> C4[Comunicação 4]
  
```

**Diagrama de Integração:**

```

graph TD
    I1[Integração 1] --> I2[Integração 2]
    I2 --> I3[Integração 3]
    I3 --> I4[Integração 4]
  
```

**Diagrama de Configuração:**

```

graph TD
    C1[Configuração 1] --> C2[Configuração 2]
    C2 --> C3[Configuração 3]
    C3 --> C4[Configuração 4]
  
```

**Diagrama de Monitoramento:**

```

graph TD
    M1[Monitoramento 1] --> M2[Monitoramento 2]
    M2 --> M3[Monitoramento 3]
    M3 --> M4[Monitoramento 4]
  
```

**Diagrama de Segurança:**

```

graph TD
    S1[Segurança 1] --> S2[Segurança 2]
    S2 --> S3[Segurança 3]
    S3 --> S4[Segurança 4]
  
```

**Diagrama de Performance:**

```

graph TD
    P1[Performance 1] --> P2[Performance 2]
    P2 --> P3[Performance 3]
    P3 --> P4[Performance 4]
  
```

**Diagrama de Usabilidade:**

```

graph TD
    U1[Usabilidade 1] --> U2[Usabilidade 2]
    U2 --> U3[Usabilidade 3]
    U3 --> U4[Usabilidade 4]
  
```

**Diagrama de Acessibilidade:**

```

graph TD
    A1[Acessibilidade 1] --> A2[Acessibilidade 2]
    A2 --> A3[Acessibilidade 3]
    A3 --> A4[Acessibilidade 4]
  
```

**Diagrama de Interoperabilidade:**

```

graph TD
    I1[Interoperabilidade 1] --> I2[Interoperabilidade 2]
    I2 --> I3[Interoperabilidade 3]
    I3 --> I4[Interoperabilidade 4]
  
```

**Diagrama de Compatibilidade:**

```

graph TD
    C1[Compatibilidade 1] --> C2[Compatibilidade 2]
    C2 --> C3[Compatibilidade 3]
    C3 --> C4[Compatibilidade 4]
  
```

**Diagrama de Portabilidade:**

```

graph TD
    P1[Portabilidade 1] --> P2[Portabilidade 2]
    P2 --> P3[Portabilidade 3]
    P3 --> P4[Portabilidade 4]
  
```

**Diagrama de Escalabilidade:**

```

graph TD
    E1[Escalabilidade 1] --> E2[Escalabilidade 2]
    E2 --> E3[Escalabilidade 3]
    E3 --> E4[Escalabilidade 4]
  
```

**Diagrama de Flexibilidade:**

```

graph TD
    F1[Flexibilidade 1] --> F2[Flexibilidade 2]
    F2 --> F3[Flexibilidade 3]
    F3 --> F4[Flexibilidade 4]
  
```

**Diagrama de Robustez:**

```

graph TD
    R1[Robustez 1] --> R2[Robustez 2]
    R2 --> R3[Robustez 3]
    R3 --> R4[Robustez 4]
  
```

**Diagrama de Confiabilidade:**

```

graph TD
    C1[Confiabilidade 1] --> C2[Confiabilidade 2]
    C2 --> C3[Confiabilidade 3]
    C3 --> C4[Confiabilidade 4]
  
```

**Diagrama de Disponibilidade:**

```

graph TD
    D1[Disponibilidade 1] --> D2[Disponibilidade 2]
    D2 --> D3[Disponibilidade 3]
    D3 --> D4[Disponibilidade 4]
  
```

**Diagrama de Segurança da Informação:**

```

graph TD
    S1[Segurança da Informação 1] --> S2[Segurança da Informação 2]
    S2 --> S3[Segurança da Informação 3]
    S3 --> S4[Segurança da Informação 4]
  
```

**Diagrama de Privacidade:**

```

graph TD
    P1[Privacidade 1] --> P2[Privacidade 2]
    P2 --> P3[Privacidade 3]
    P3 --> P4[Privacidade 4]
  
```

**Diagrama de Integridade:**

```

graph TD
    I1[Integridade 1] --> I2[Integridade 2]
    I2 --> I3[Integridade 3]
    I3 --> I4[Integridade 4]
  
```

**Diagrama de Autenticidade:**

```

graph TD
    A1[Autenticidade 1] --> A2[Autenticidade 2]
    A2 --> A3[Autenticidade 3]
    A3 --> A4[Autenticidade 4]
  
```

**Diagrama de Não Repúdio:**

```

graph TD
    N1[Não Repúdio 1] --> N2[Não Repúdio 2]
    N2 --> N3[Não Repúdio 3]
    N3 --> N4[Não Repúdio 4]
  
```

**Diagrama de Aceitação:**

```

graph TD
    A1[Aceitação 1] --> A2[Aceitação 2]
    A2 --> A3[Aceitação 3]
    A3 --> A4[Aceitação 4]
  
```

**Diagrama de Satisfação:**

```

graph TD
    S1[Satisfação 1] --> S2[Satisfação 2]
    S2 --> S3[Satisfação 3]
    S3 --> S4[Satisfação 4]
  
```

**Diagrama de Engajamento:**

```

graph TD
    E1[Engajamento 1] --> E2[Engajamento 2]
    E2 --> E3[Engajamento 3]
    E3 --> E4[Engajamento 4]
  
```

**Diagrama de Retenção:**

```

graph TD
    R1[Retenção 1] --> R2[Retenção 2]
    R2 --> R3[Retenção 3]
    R3 --> R4[Retenção 4]
  
```

**Diagrama de Crescimento:**

```

graph TD
    C1[Crescimento 1] --> C2[Crescimento 2]
    C2 --> C3[Crescimento 3]
    C3 --> C4[Crescimento 4]
  
```

**Diagrama de Inovação:**

```

graph TD
    I1[Inovação 1] --> I2[Inovação 2]
    I2 --> I3[Inovação 3]
    I3 --> I4[Inovação 4]
  
```

**Diagrama de Sustentabilidade:**

```

graph TD
    S1[Sustentabilidade 1] --> S2[Sustentabilidade 2]
    S2 --> S3[Sustentabilidade 3]
    S3 --> S4[Sustentabilidade 4]
  
```

**Diagrama de Resiliência:**

```

graph TD
    R1[Resiliência 1] --
```

23

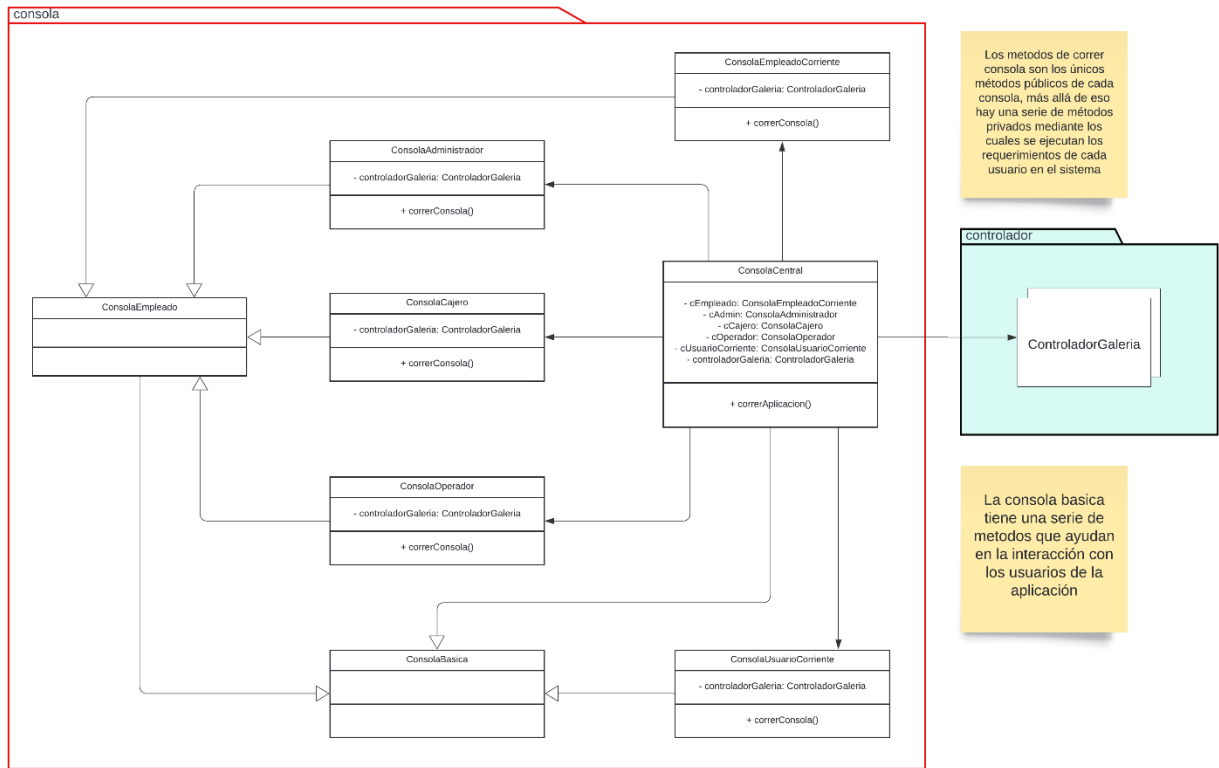


Figura 4: Diagrama UML Consola