

Proyecto 1 – Entrega 2

Diego Molano Roa – 202123015

Natalia Lemus - 202024977

Nicolás Castiblanco – 202224390

Diseño del programa

1. Contexto del problema

El problema se basa en construir un sistema para una galería, diseñando toda su estructura y funcionalidad para que los usuarios puedan interactuar con ella. La galería cuenta con piezas propias o piezas que son consignadas por clientes y estas que deben ser registradas en el inventario, ya sea en la bodega o en exhibición, para que puedan ser vendidas o subastadas a los compradores. Los procesos de venta y subasta deben permitir la interacción con uno o más compradores y llevar registro de los cambios en el estado de las piezas a medida que se realicen acciones sobre estas. Adicionalmente, la galería lleva registro de todos los usuarios, entre los cuales se tienen empleados, cada uno con distintas responsabilidades y funciones, y los clientes, los cuales pueden interactuar con la galería ya sea cómo compradores o cómo propietarios. Todo usuario debe estar registrado con un nombre y login únicos y con una contraseña privada.

2. Nivel 1

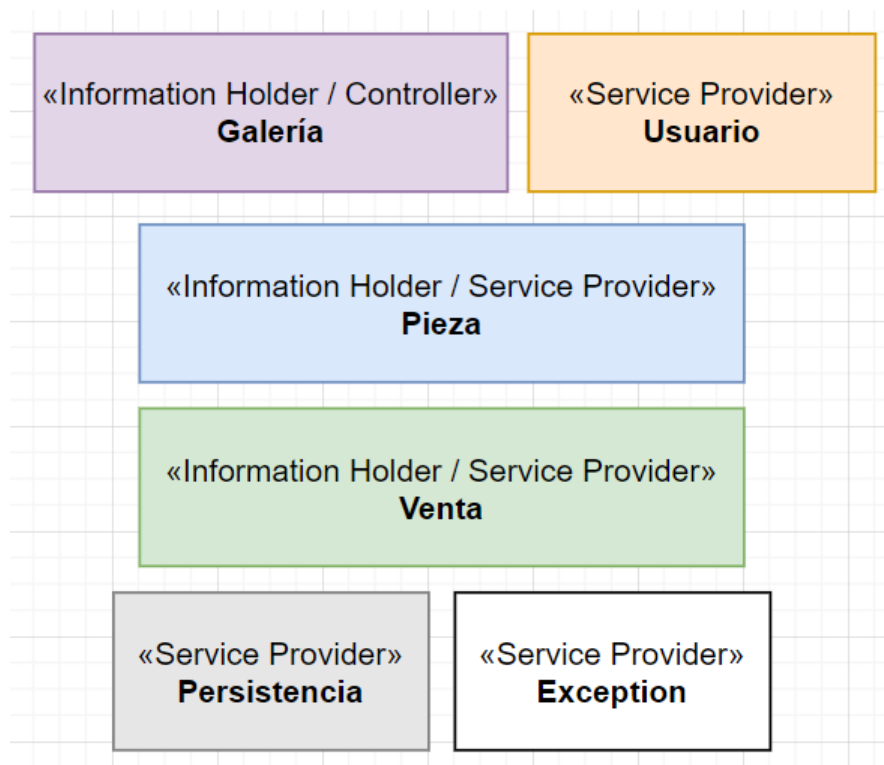
2.1. Componentes candidatos y estereotipos

1. Dependiendo del usuario que sea este tendrán diferentes acciones y responsabilidades dentro de la galería, que controlara la clase usuario. Teniendo en cuenta esto, la clase usuario ayuda a galería para cumplir los trabajos que esta necesite. Esta clase adopta el estereotipo de ***Service Provider***.

2. Respecto al manejo de información acerca de las listas de piezas y de todo el listado de usuarios se debe tener en cuenta una clase la cual guarde toda esa información y la mantenga, por eso la clase llamada Galería será la encargada de esto, siendo de tipo ***Information Holder***. La galería será la encargada de manejar las acciones de las otras clases, al recibir la información general de las demás clases. Por ende, el tipo será ***Controller***.

3. Puesto que las piezas deben mantener la información que contenga esta, como el estado, el autor, la fecha de creación, etc. Y permitir según el estado de éstas, interacciones con el usuario para que puedan adquirir una pieza o la venta de esta. Por ende, la clase pieza adoptara el estereotipo de ***Information Holder***. No obstante, al depender de la información que tenga esta para validar las compras de las piezas, ayuda a la parte de ventas y subastas de la galería. Por ende, se adapta al estereotipo ***Service provider***.

4. En la galería existen modalidades de venta para los usuarios para que puedan adquirir las piezas. Dentro de estas, podemos ver la información de la pieza dependiendo de la modalidad que se esté interesado. Por ende, lo podemos tomar como un ***Information Holder***. Cuando hay un movimiento relacionado con una pieza en específico, se tienen que comprobar varias condiciones para que se cambie el estado de la pieza. Y después de eso hacer el movimiento. Esto se acopla a la definición de ***Service Provider***.



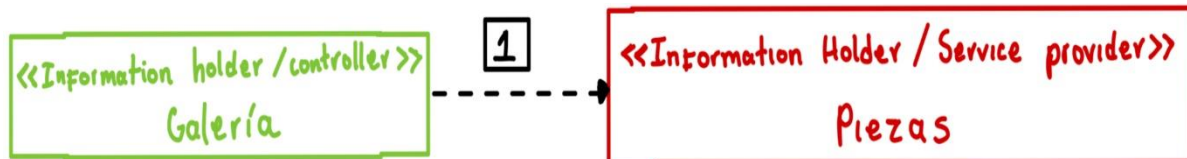
2.2. Responsabilidades

#	RESPONSABILIDADES	COMPONENTE
1	Dar lista de piezas en exhibición	Galería
2	Recibir y Devolver información (con ayuda de las otras secciones)	Galería
3	Conocer el listado de piezas de un propietario	Usuario
4	Iniciar Sesión	Usuario
5	Verificar Tipo de Usuario	Usuario
6	Registrar Ofertas	Usuario
7	Registrar Compras	Usuario
8	Dar información de pieza	Pieza
9	Registrar Pieza	Pieza
10	Ofertar Dentro de Subasta	Pieza
11	Conocer el estado de pieza	Pieza
12	Vender / Comprar Pieza	Pieza
13	Verificar Método De Pago	Venta

2.3. Colaboraciones

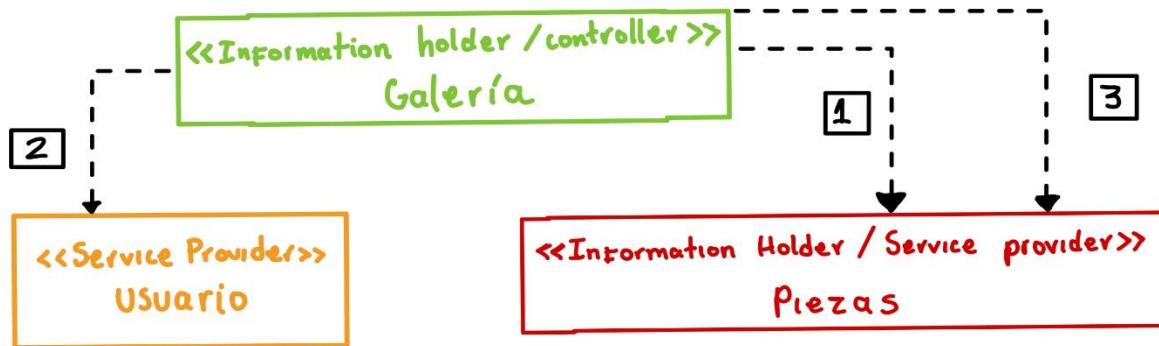
. **Dar lista de piezas en exhibición:** El controlador después de acceder a la lista de piezas

1. La clase galería accede a la lista de piezas. Al acceder, revisa las piezas que tengan como estado ‘exhibición’, y esta crea una lista con las piezas exhibidas, para que los usuarios puedan verla.



. **Agregar una nueva pieza al inventario:** El controlador al recibir esta tarea

1. La galería empieza comprobando si la pieza a agregar ya está en el inventario.
2. Le pregunta a la clase “usuario” de tipo service provider, si el usuario que está registrado tiene rol de administrador. Si es correcto, la acción se podrá realizar
3. Se añade la nueva pieza al inventario

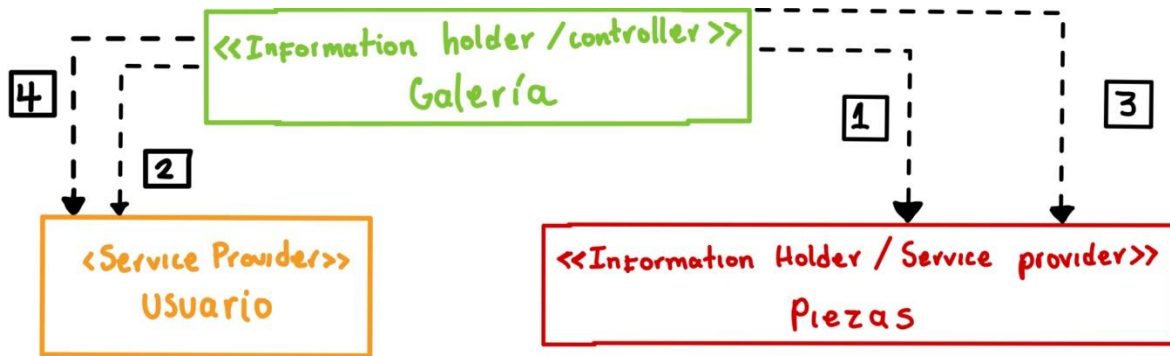


. **Ofertar dentro de una subasta**

1. Le pregunto a la clase usuario, si el comprador ya está verificado por el administrador.
2. Reviso si el comprador tiene los ingresos suficientes para participar en la subasta.

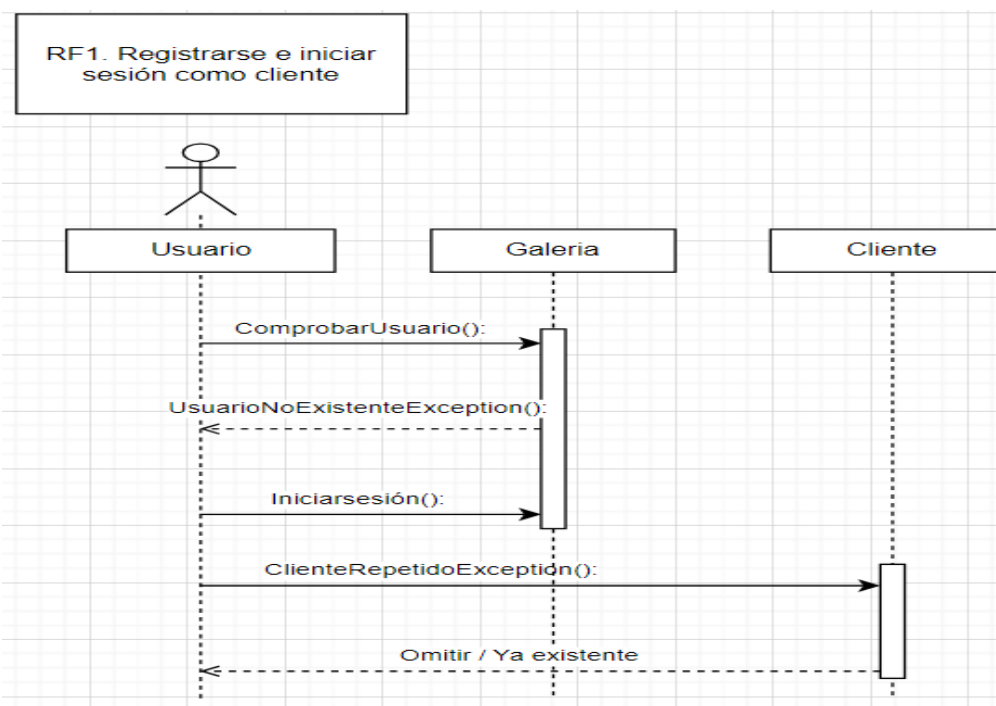
. **Vender / Comprar Pieza:** El controlador recibe la tarea

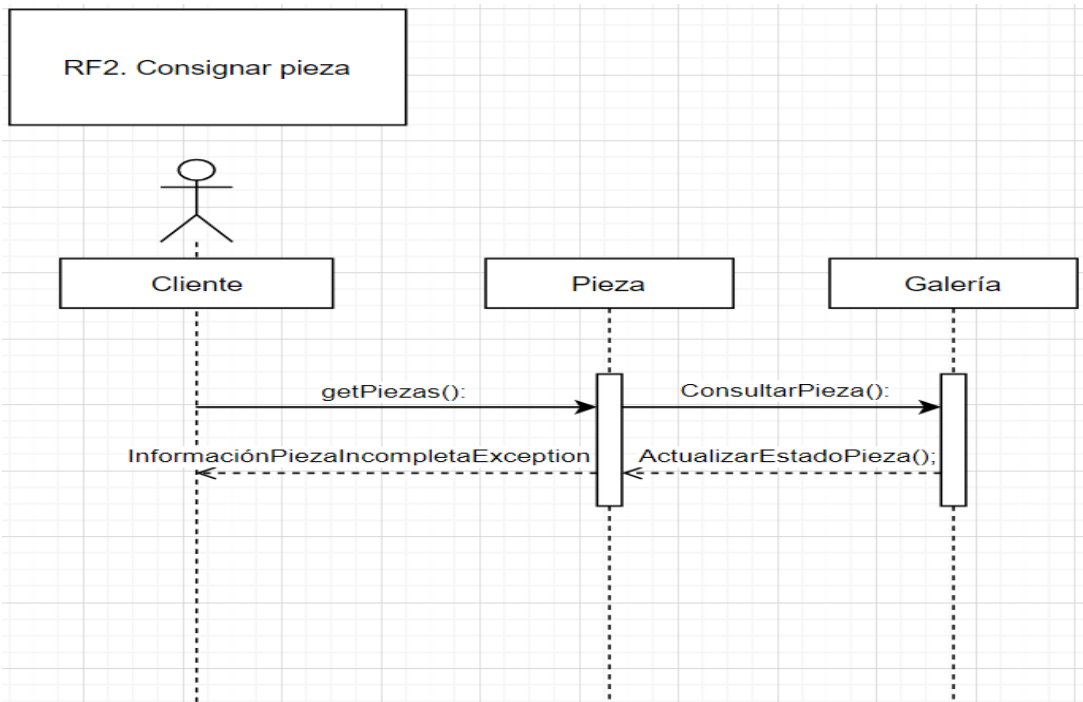
1. Ver si la pieza está disponible para hacer la compra
2. Busco al usuario interesado en la pieza, para comprobar si tiene los suficientes fondos para comprarla.
3. Una vez la pieza se venda, edito el estado de la pieza de disponible a vendida. Y al mismo tiempo ya no será parte de la galería.
4. Por último el usuario ahora será propietario de una pieza de arte.



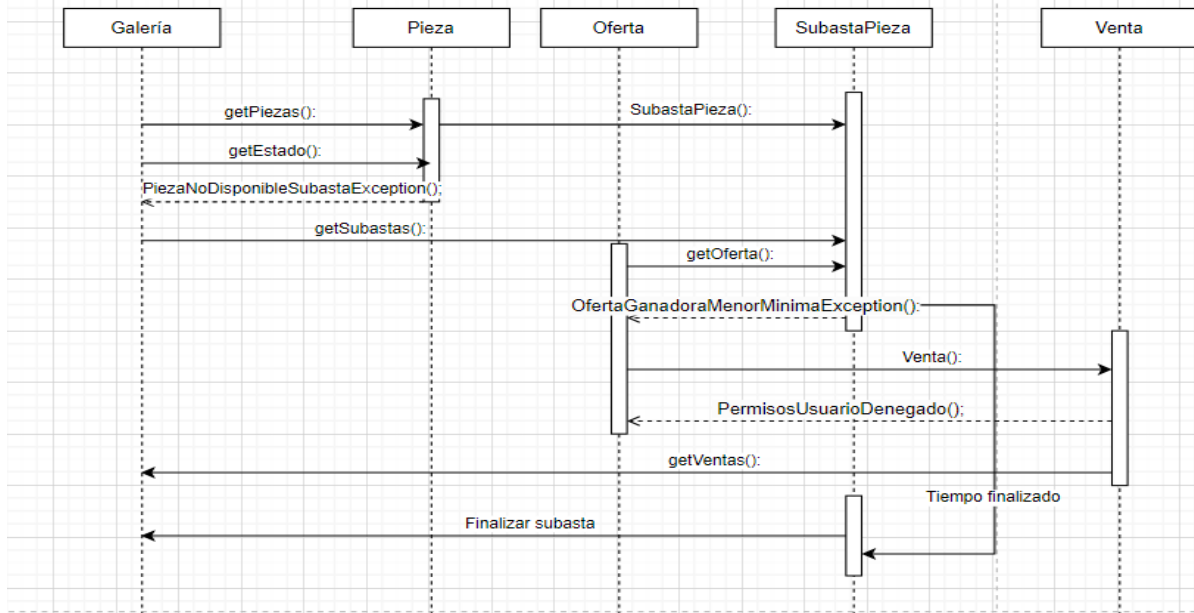
En las colaboraciones que se presentan en el documento, se puede evidenciar que la toma de decisiones y el control de acciones sobre las otras clases se concentra en el controlador, que en este caso representa Galería. Las colaboraciones necesitan la ayuda de diferentes clases para cumplir con su objetivo. En general se basa en la comunicación de estas. Un ejemplo de esto es el proceso de Vender o comprar una pieza. Debido que la clase controlador al recibir la instrucción se comunica con piezas para ver la disponibilidad de esta, después tendremos que asegurarnos que la clase Usuario este verificado para poder comprarla pieza.

DIAGRAMAS DE SECUENCIA

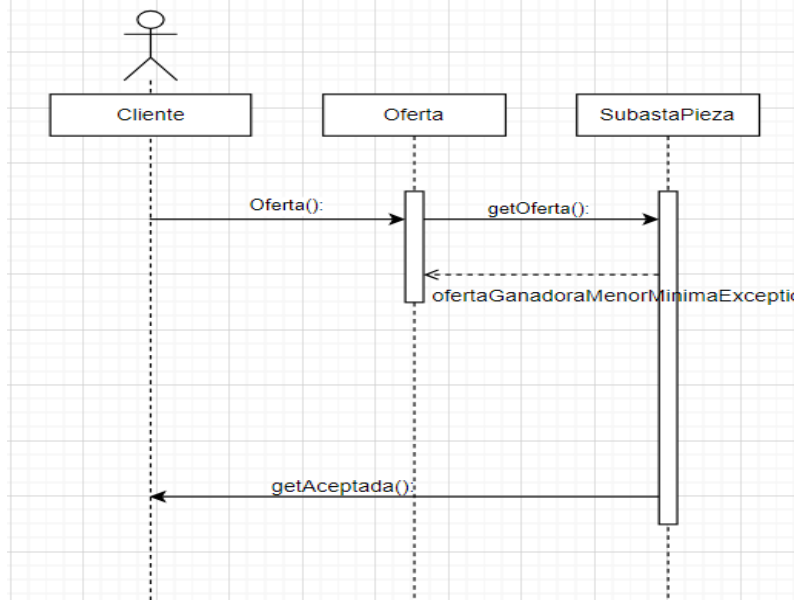


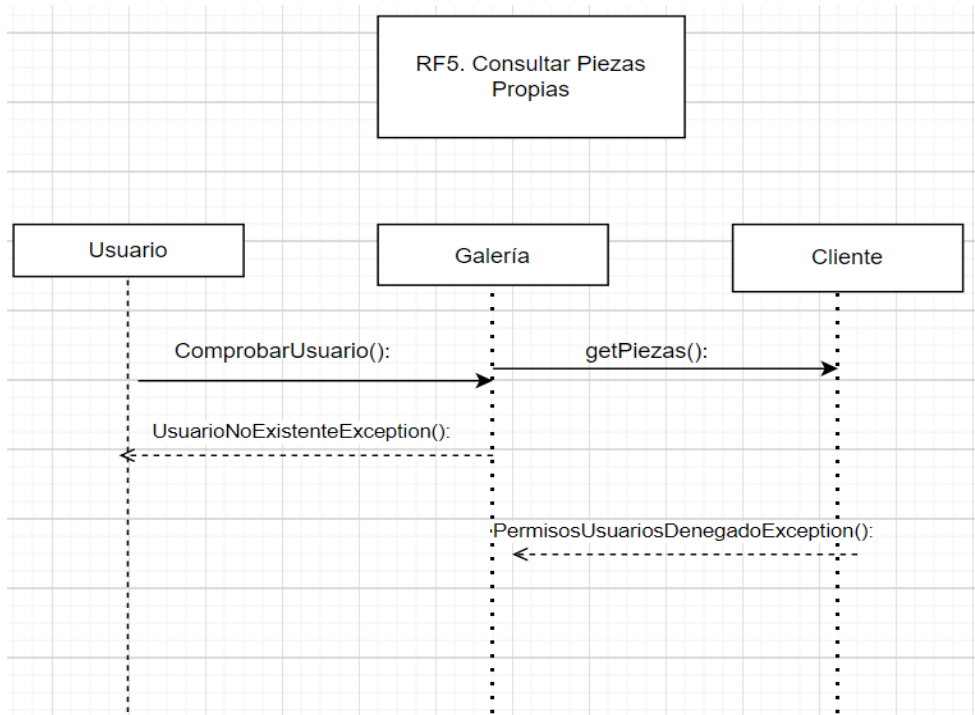


RF3. Realizar subasta



RF4. Realizar oferta en subasta





2.2 Nivel 2

2.3. Information Holder / Controller Galería

2.3.1. Componentes candidatos y estereotipos

Para hacer la carga de todos los usuarios y piezas dentro de la galería. Además de controlar las acciones que se hagan dentro de esta. Esta nos indica que el componente es un controlador. Este es el punto de arranque, debido que esta clase nos permitirá cargar todos los datos como los usuarios, las ventas, las piezas, etc.

2.3.2. Responsabilidades

Las responsabilidades de la clase galería se basan exclusivamente en recibir la información de las otras clases. Esto porque galería es el controlador del programa, lo que hace que tome decisiones importantes y controle las acciones de las demás clases con la información que le llega.

2.3.3. Colaboraciones

Debido a que galería corresponde solo a una clase, no existen colaboraciones dentro de esta.

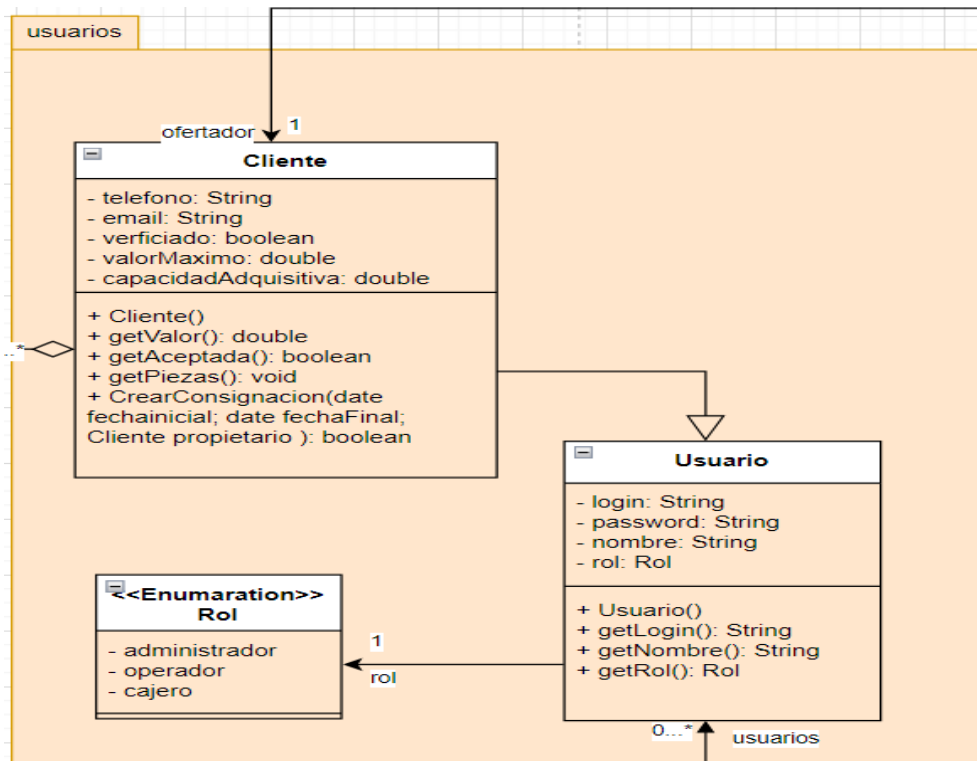
2.4. Usuario

2.4.1. Componentes candidatos y estereotipos

En esta clase estará la información de los usuarios registrados en la galería. Hay distintos tipos de usuarios, por eso es importante diferenciar a cada uno con un login y contraseña. Están los empleados de la galería y los clientes. Ambos tienen diferentes métodos según la necesidad que tengan. En el caso de los clientes, tenemos a los compradores y los propietarios. Además, esta clase se conecta con la parte de ventas para poder comprobar diferentes requisitos de los usuarios a la hora de adquirir una pieza. Este componente hace referencia a un *service provider*.

2.4.2. Responsabilidades

Las responsabilidades de este componente son poder registrar al usuario dentro de la galería con su usuario y su contraseña. Al igual que para los clientes tener la capacidad adquisitiva de este y las piezas de las cuales es propietario.



2.4.3. Colaboraciones

Como vemos dentro de las responsabilidades hay que efectuar diferentes tareas mediante la misma clase, por lo que se dividen diferentes clases dentro del componente usuario. También,

como en la galería existen diferentes tipos de usuarios, también hay que crear otra clase correspondiente a esto. Esta se llamará rol, será de tipo *enumeración* para facilitar esta acción. Sumado a esto, para que la galería obtenga las piezas se tiene que implementar una clase llamada consignación, que tenga como parámetros: la fecha inicial y la final, el propietario y por último si esta activa. De igual manera, habrá un apartado de usuarios los cuales sean propietarios o compradores. Por esta razón, se crea la clase cliente, que tenga toda la información de contacto de este.

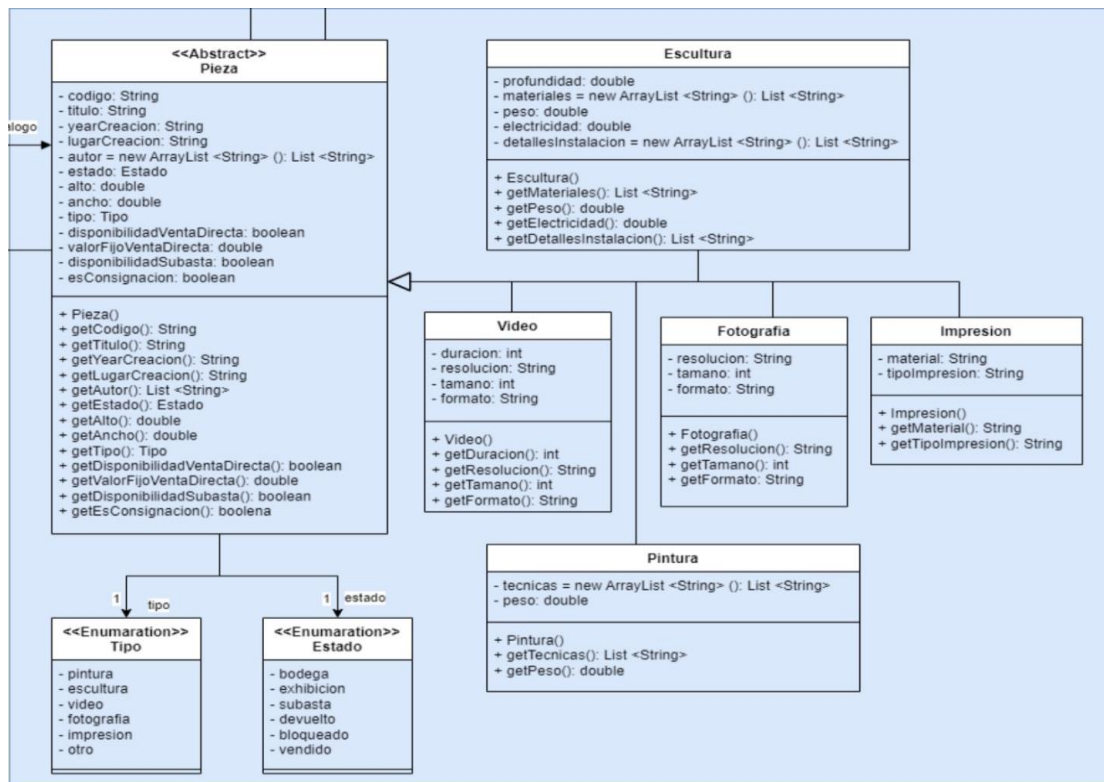
2.5. Information Holder / Service Provider Pieza

2.5.1. Componentes candidatos y estereotipos

Esta clase almacena toda la información de las piezas como su autor, estado, etc. Por otro lado, está cuenta con otros atributos que contribuyen en otros aspectos de la galería, como el de la venta y subasta de piezas. Se espera que el usuario pueda acceder a esta, para ver las piezas que estén disponibles y poder interactuar con estas. Por ende, esta representaría Information Holder y Service Provider.

2.5.2. Responsabilidades

Las responsabilidades de este componente se basan en obtener toda la información de la pieza. Para que dependiendo de la acción que se quiera realizar, se tenga la información de las piezas. Algunas responsabilidades están ligadas a la parte de ventas para ver la disponibilidad de una pieza en una subasta, comprobar si está disponible para comprar en venta directa y si tiene un valor fijo.



2.5.3. Colaboraciones

Este componente cuenta con atributos de información de la pieza, lo cual nos hace referencia a que es de tipo Information Holder. Al igual que service provider al tener diferentes interacciones con estas.

Por otra parte, las colaboraciones dentro de este componente son dos. Estos refiriéndose a los distintos métodos de pago y tipos que las piezas puedan adquirir, así que se implementan otras clases de tipo *enumeración* para que estos procesos se faciliten. La clase tipo nos indica como el nombre lo dice, el tipo de la pieza. Y la clase estado corresponde al estado que tenga una pieza en específico, esto es importante debido que dependiendo del estado de la pieza se podrán o no hacer diferentes acciones.

2.6. Modalidades de Venta

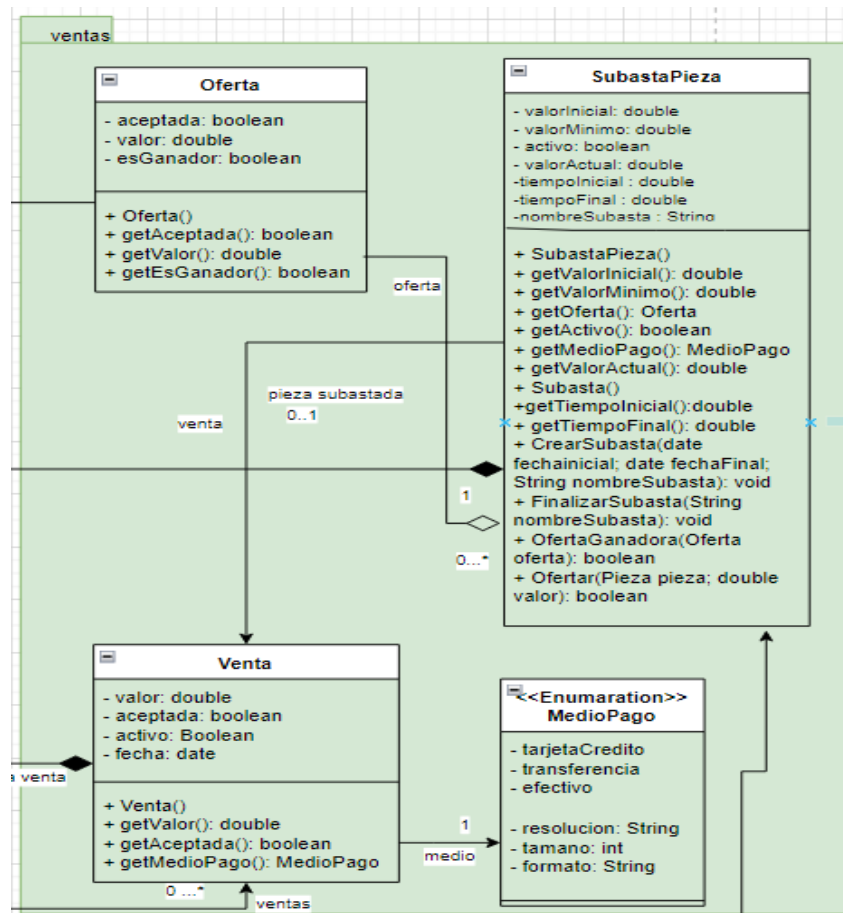
2.6.1. Componentes candidatos y estereotipos

Mediante este componente las piezas se podrán vender o subastar. Al igual que los usuarios podrán interactuar haciendo ofertas y pagando las piezas. Este componente al tener toda la información de las ofertas y ventas que se realizan por cada subasta tendría el tipo de *information*

Holder. Además, se encarga de realizar los pagos por cada pieza, debido a esto representaría de igual manera un *service provider*.

2.6.2. Responsabilidades

Dentro de las responsabilidades de este componente esta poder registrar toda la información de las ofertas. Desde el ganador de esta, el valor que se oferto y si esta es aceptada. Como la parte de venta, que tiene como responsabilidad ver si se aceptó, ver el valor de la venta y obtener el medio de pago por el que el usuario adquiere la pieza. Por último, la parte de la subasta la cual se encarga del valor inicial de la pieza y todas las ofertas registradas por la subasta.

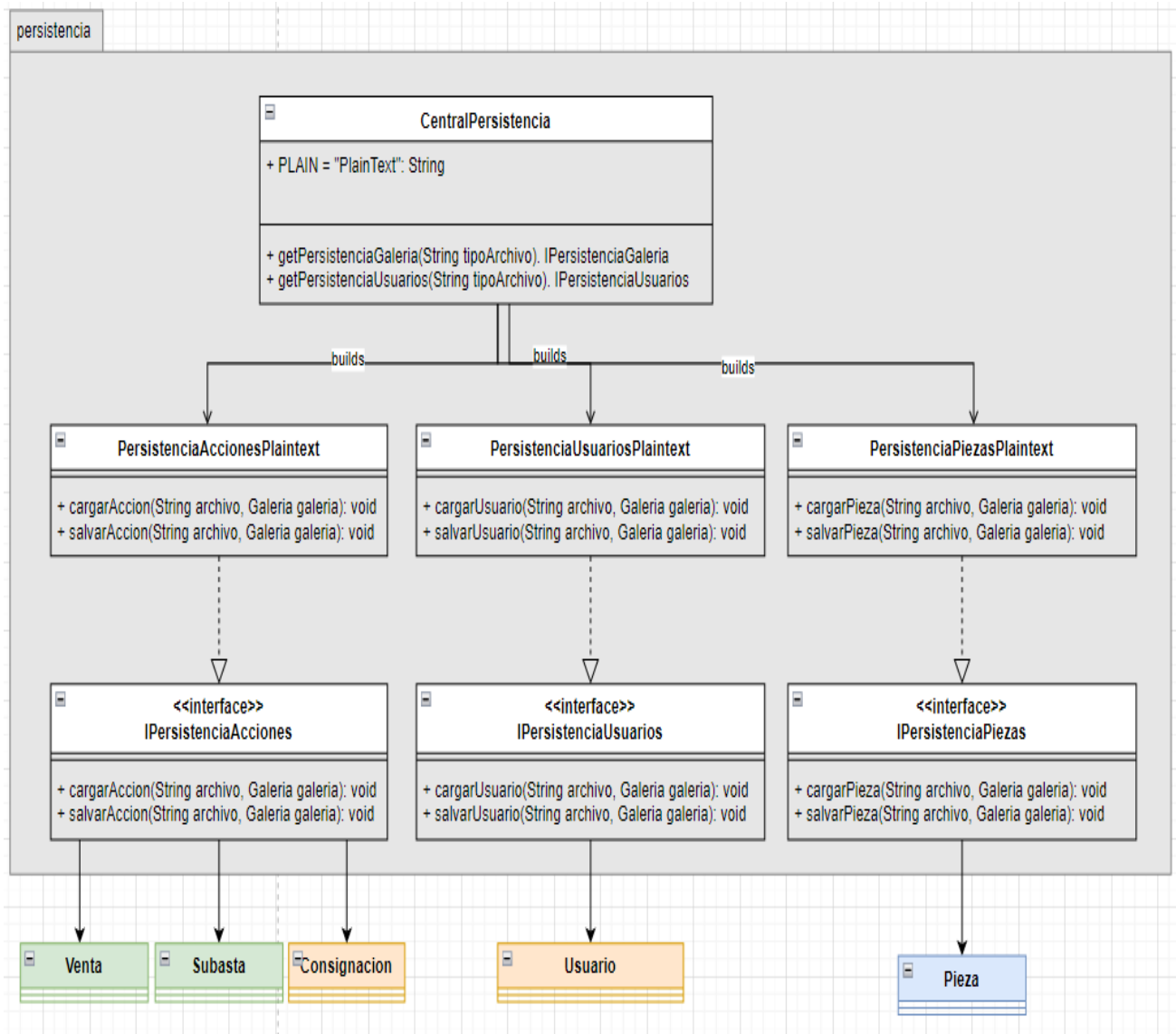


2.6.3. Colaboraciones

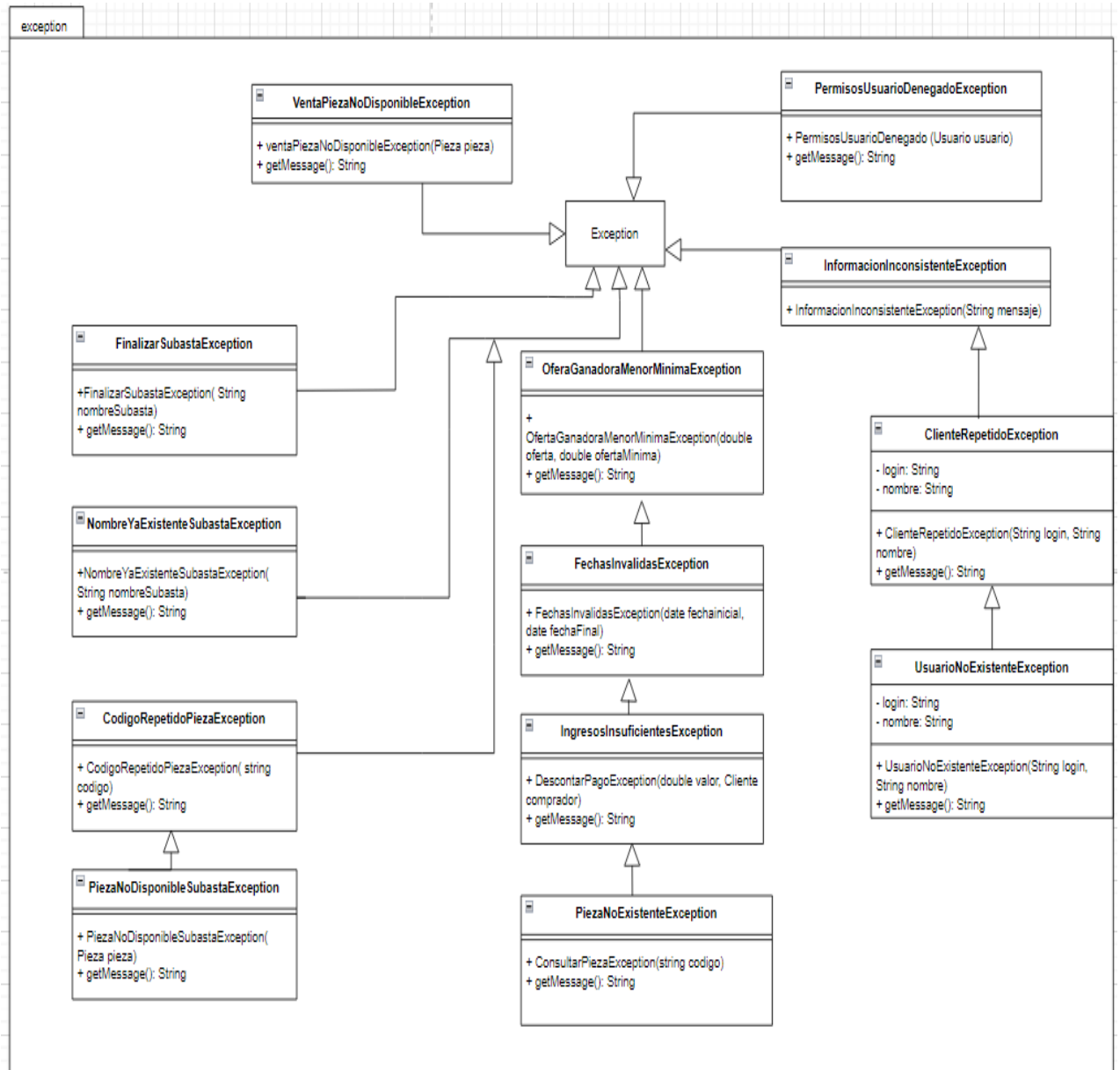
Dentro del componente de venta se tienen que cumplir diferentes tareas. Es por esto, que esta clase se divide en 3 clases que nos permitan cumplir todas las tareas. La primera clase, Venta,

nos permite acceder a la información de la venta de una pieza, igual que el método de pago que se efectuará, para que los compradores se conviertan en propietarios. La segunda clase será la de subasta que llevará el registro de todas las ofertas que se hagan dentro un tiempo acordado por la galería. Por último, será necesario crear otra clase para que se hagan las ofertas y poder comprobar si son aceptadas o no.

2.7. Persistencia



2.8. Excepciones



3. Diseño final

