

# **Proyecto 1 - Documento de Diseño Galería**

**Andrés Felipe Pereira**

**Santiago Vaquero**

**Juan David Roncancio**

**Diseño y Programación O.O.**

**Germán Romero**

**Universidad de los Andes**

**Ingeniería de Sistemas**

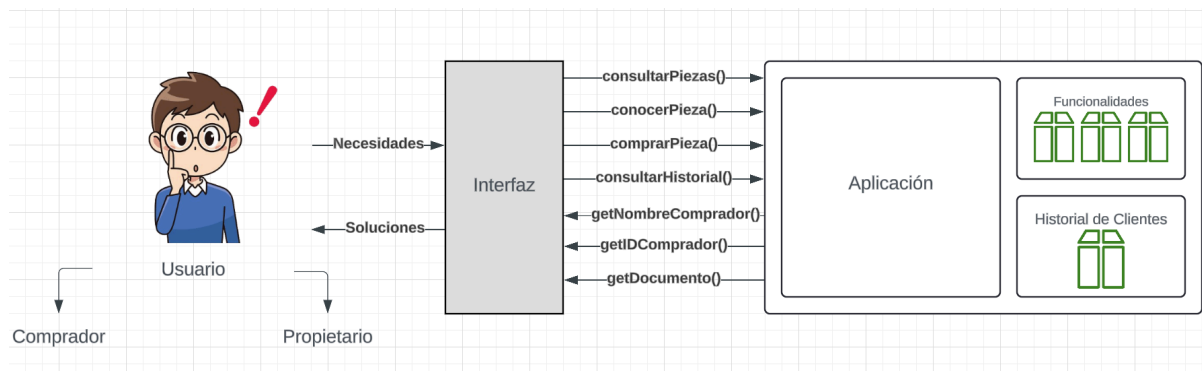
**Bogotá D.C.**

**2024**

# Proyecto 1 - Documento de Diseño de Galería

## 1. Contexto del problema

La aplicación que se va a presentar a continuación tiene el objetivo de administrar las diferentes funciones y servicios con los que cuenta una galería de arte, así como facilitar actividades de almacenamiento y manejo de información correspondiente a la variedad de piezas que allí se encuentran y de clientes como propietarios y posibles compradores. A continuación se presentará más gráficamente la distribución de la aplicación, con sus respectivas interfaces y funcionalidades, por supuesto.



Existen diferentes interfaces que funcionarán dentro de la aplicación, el usuario, el cual puede ser comprador o propietario y los trabajadores (de los cuales hablaremos más adelante) que pueden ser administradores o contadores.

Por un lado, la interfaz del usuario debe permitirle a este interactuar y ejecutar diferentes necesidades como consultar las piezas disponibles en el momento, con su respectiva información sobre características, historia y anteriores propietarios, adquirir una pieza que esté disponible para la venta y hasta participar en subastas de piezas. La aplicación por su parte debe pedirle al usuario información personal como el nombre, el documento de identidad, que después se almacenará dentro de la aplicación a modo de una base de datos.

## 2. Nivel 1

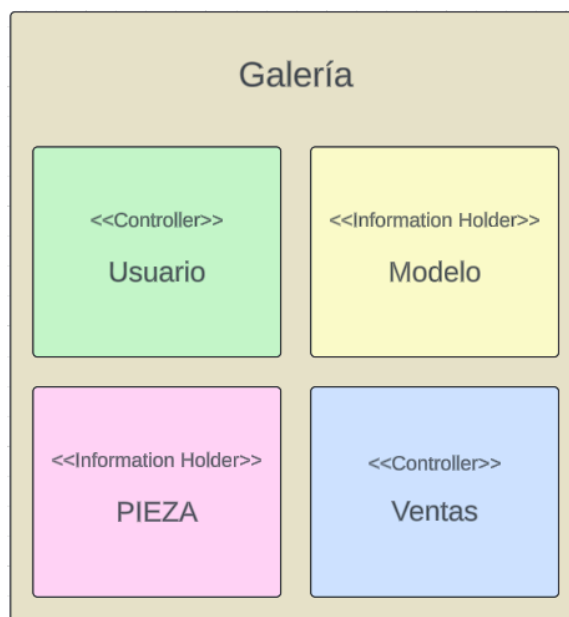
### 2. 1. Componentes candidatos y estereotipos

1. En primer lugar, uno de los roles más importantes dentro de la aplicación es el de los administradores, contadores, operarios, clientes, etc... pertenecientes

a la clase *Usuario*. Los primeros, encargados, entre otras cosas, de verificar información relevante sobre los clientes y funcionar como mediadores en los procesos de venta y adquisición de piezas entre los clientes y los controladores de subastas y ventas, a este se le asignará el estereotipo *controller*. Por su parte la clase de clientes aborda tanto a propietarios como a compradores.

2. En segundo lugar, la información que se guardará sobre los diferentes tipos de piezas, pertenecientes a la clase padre *PIEZA*, con su respectiva historia, datos importantes y dueños pasados o actuales se le asignará el estereotipo *Information Holder*.
3. El modelo por su parte será una parte importante dentro del sistema al reunir información importante sobre la disponibilidad, historia y datos sobre las piezas que estén tanto exhibidas como en bodega dentro de la galería. Por tanto, el modelo contará con el estereotipo de *Information Holder*. Modelo agrupa a Galería e Inventario, los cuales entre otras cosas, manejan la información total de la galería y la almacenan en el inventario.
4. Finalmente, el corazón de la aplicación será la posibilidad de los compradores y propietarios de adquirir una pieza de la galería, además de poder participar en subastas de piezas que la galería realizará cada cierto tiempo. Por esta razón más que evidente, el sistemas de ventas y subastas *Ventas*, estará determinado por el estereotipo *Controller*, pues al tratarse de un proceso de interacción directa con el usuario, y requerir de servicios de administrador, se le está dando el control y manejo de las piezas de la galería.

La siguiente figura expone los posibles candidatos de componentes con sus respectivos estereotipos según lo que se explicó anteriormente.



## 2.2. Responsabilidades

En el siguiente cuadro se evidencian las diferentes responsabilidades de cada uno de los componentes de la aplicación.

#	Responsabilidad	Componente
1	Verificar comprador	Usuario
2	Registrar ingreso de pieza al inventario	
3	Confirmar venta o devolución de pieza	
4	Registrar venta de una pieza	
5	Comprar pieza	
6	Conocer piezas compradas por el cliente	
7	Añadir piezas nuevas a la galería	Modelo
8	Conocer las piezas pasadas del inventario	
9	Conocer las piezas actuales en exhibición	
10	Conocer las piezas actuales en bodega	
11	Conocer las piezas para subasta	
12	Conocer las piezas para venta fija	
13	Subastar una pieza	Ventas
14	Vender una pieza	
15	Conocer el precio inicial y final de las piezas	
16	Conocer los compradores o participantes a subasta	
17	Conocer las características de cada pieza	Pieza
18	Conocer el propietario de cada pieza	
19	Conocer la modalidad de adquisición de cada pieza	

## 2.3. Colaboraciones

Las responsabilidades que fueron descritas en el anterior cuadro llevan a cabo colaboraciones para así cumplir con las funcionalidades y requerimientos que necesita el usuario. A continuación se enuncian dichas colaboraciones.

- **Conocer una pieza:** Para que el usuario pueda conocer información relevante sobre la pieza debe recurrir directamente al inventario de piezas. El inventario deberá realizar las siguientes acciones:
  1. Recurrir a Pieza <<Information Holder>>.
  2. Conocer la información de cada pieza que ingresa a la galería.
  3. Almacenar información.
  4. Registrar piezas como disponibles o adquiridas.
- **Comprar una pieza:** Para que el usuario pueda comprar una pieza por la modalidad de venta fija. Venta Fija deberá realizar las siguientes acciones:

1. Preguntar al usuario la pieza que desea adquirir.
  2. Recurrir al inventario de piezas <<Information Holder>>.
  3. Conocer la modalidad de adquisición de la pieza dentro de su información.
  4. En caso de que la modalidad sea de venta fija, bloquear la pieza directamente en el inventario (cambiar su estado de “Disponible para venta” a “Reservada”).
  5. Recurrir a Trabajador -> Administrador.
  6. El administrador verifica al comprador.
  7. Una vez el comprador sea verificado, la pieza se transfiere al comprador y este pasa a ser propietario.
- **Subastar una pieza:** Para que el usuario pueda adquirir una pieza por medio de la modalidad de subasta. Subasta deberá realizar las siguientes acciones:
    1. Preguntar al usuario la pieza que desea adquirir.
    2. Recurrir al inventario de piezas <<Information Holder>>.
    3. Conocer la modalidad de adquisición de la pieza dentro de su información.
    4. Recurrir a Trabajador -> Administrador.
    5. El administrador verifica al comprador.
    6. En caso de que la modalidad sea de subasta, Subasta <<Controller>> deberá pedirle al usuario el valor mínimo que está dispuesto a ofrecer por la pieza.
    7. Subasta <<Controller>> deberá verificar que este valor mínimo sea mayor o igual al precio inicial de la pieza que se subasta.
    8. Después de ser verificado, Subasta <<Controller>> deberá conocer el valor máximo que el comprador está dispuesto a ofrecer por la pieza.
    9. En caso de que el comprador haya ofrecido un valor que no sea superado por ninguna de las demás ofertas por la pieza, el comprador habrá adquirido la pieza.
  - **Devolver una pieza por venta fija:** Para que el sistema pueda devolver una pieza al inventario después de un intento de compra, deberá realizar las siguientes acciones:
    1. Después de una venta fallida o de no considerar al comprador como verificado, la pieza permanecerá dentro del inventario de la galería.
    2. Se desbloqueará la pieza, es decir, su estado pasará de “Reservada” a “Disponible para venta”.

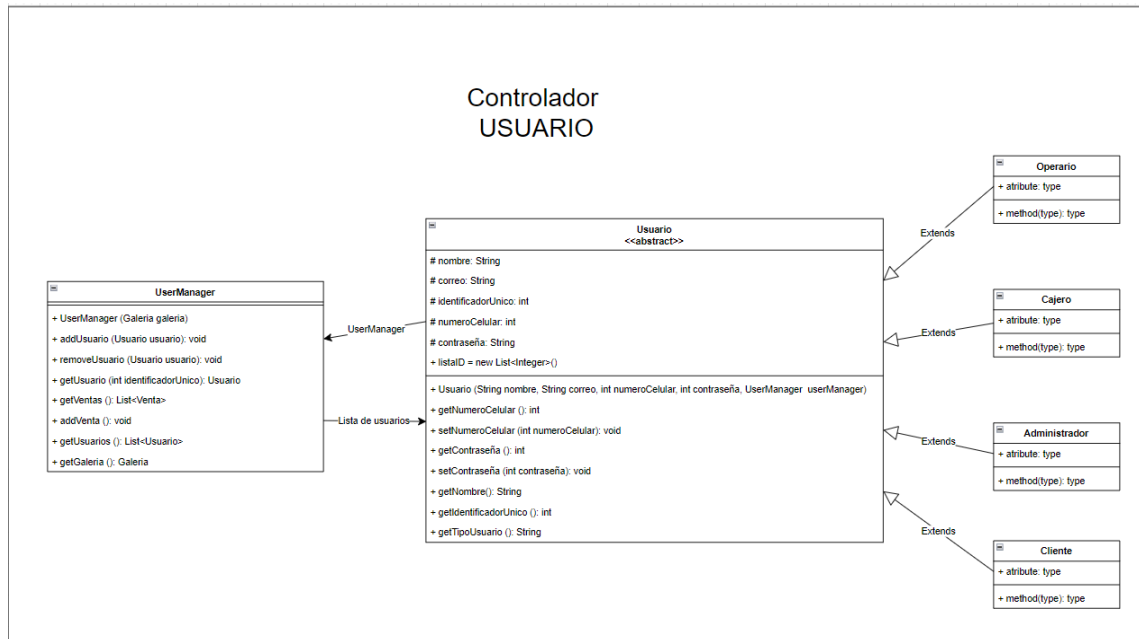
### 3. Nivel 2

Continuando con el proceso de diseño por niveles, se empezará con la descomposición de los componentes que fueron enunciados anteriormente.

#### 3.1. Usuario

### 3.1.1. Componentes candidatos y estereotipos

Teniendo en cuenta las responsabilidades que fueron evidenciadas del componente Usuario <<Controller>>, lo más conveniente en este caso es pensarlo como padre de dos roles importantes dentro del funcionamiento de la aplicación. Así pues, Administrador y Cajero, son dos clases que heredan sus atributos de la clase padre Usuario, la cual, cuenta con atributos generales como nombre, celular, ID y password correspondientes al operario. Para la clase padre Usuario, se manejan métodos como getNumeroCelular( ), getContraseña( ), getNombre( ), getIdentificacion( ). Por su parte, Cajero cuenta con el método de registrarVenta() y Administrador con confirmarConsignacion() y confirmarVenta().



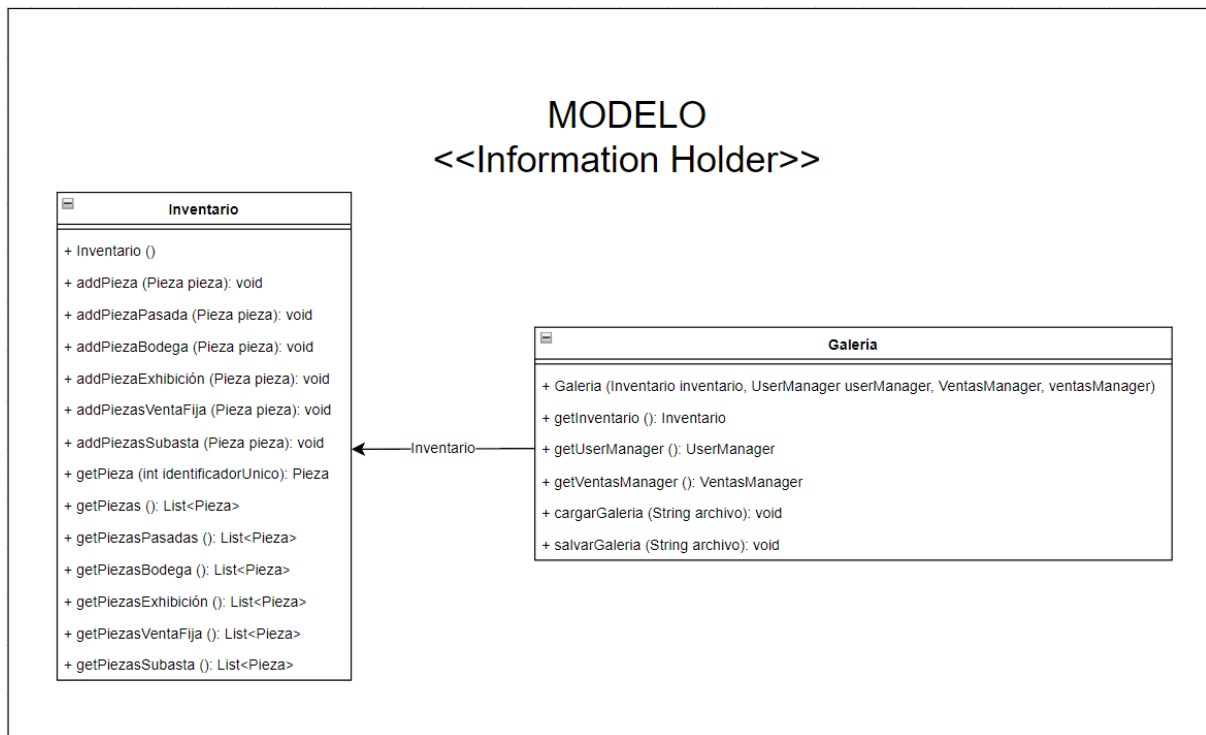
### 3.1.2 Responsabilidades y colaboraciones

El anterior componente cuenta con una clase padre y dos hijas, las cuales se complementan con métodos y atributos para llevar a cabo sus acciones y las colaboraciones fueron descritas anteriormente.

## 3.2. Modelo

### 3.2.1. Componentes candidatos y estereotipos

Inventario representa una sola clase condensada con sus respectivos atributos y un método importante para llevar a cabo los servicios que suplen las necesidades de los demás componentes dentro de la aplicación. Así pues, el <<Information Holder>> Modelo almacena información importante sobre cada una de las piezas que pertenecen al inventario tanto en exhibición como en bodega, así como las piezas que han sido parte de la galería en el pasado.



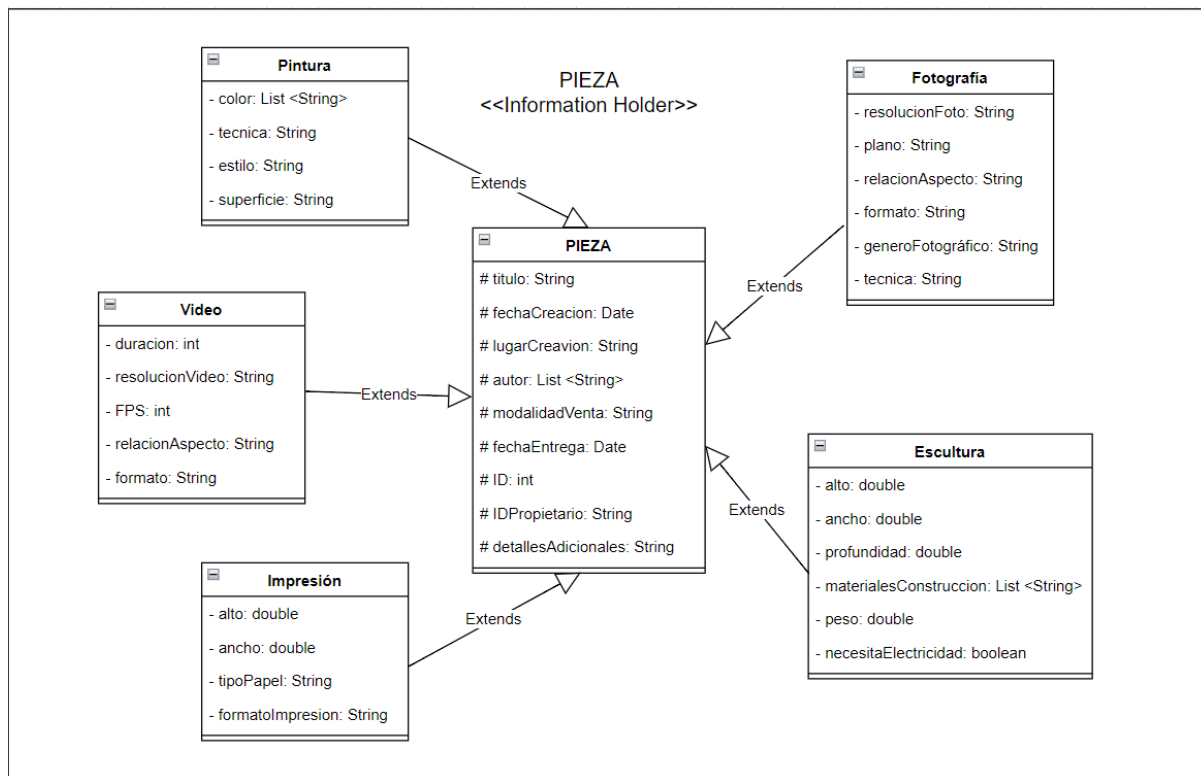
### 3.2.2. Responsabilidades y colaboraciones

Teniendo en cuenta que Modelo representa un *<<Information Holder>>* condensado en dos clases, las responsabilidades ya expuestas en la sección 2.2. irán de la mano con las colaboraciones entre las diferentes clases que recurren a inventario para realizar sus funciones. Dentro de los métodos de Inventario se encuentran diferentes acciones para clasificar las piezas que son recibidas mediante estructuras de datos desde la clase pieza, es decir, una pieza puede clasificarse en piezas pasadas, piezas exhibidas o piezas en bodega. Esto resulta de gran ayuda en la colaboración con otros componentes, pues el tener clasificadas las piezas por su estado facilita el proceso de subasta y venta, esto, teniendo en cuenta que para la modalidad de venta por precio fijo, solo pueden ser adquiridas piezas que estén disponibles para la venta, por lo cual, no pueden ser propiedad de un comprador conocido como propietario.

## 3.3 Pieza

### 3.3.1. Componentes candidatos y estereotipos

Pieza es una clase padre que maneja atributos comunes para 5 diferentes tipos de piezas, de la cuales, cada una cuenta con diferentes atributos únicos dada la composición y las características importantes para cada tipo de pieza. Pieza es un *<<Information Holder>>* dada la funcionalidad que cumple dentro de la aplicación, conoce cada una de las características que posee cada pieza dentro de la galería, así como la información de todos sus propietarios, si es que los ha tenido, por supuesto.



### 3.3.2. Responsabilidades y colaboraciones

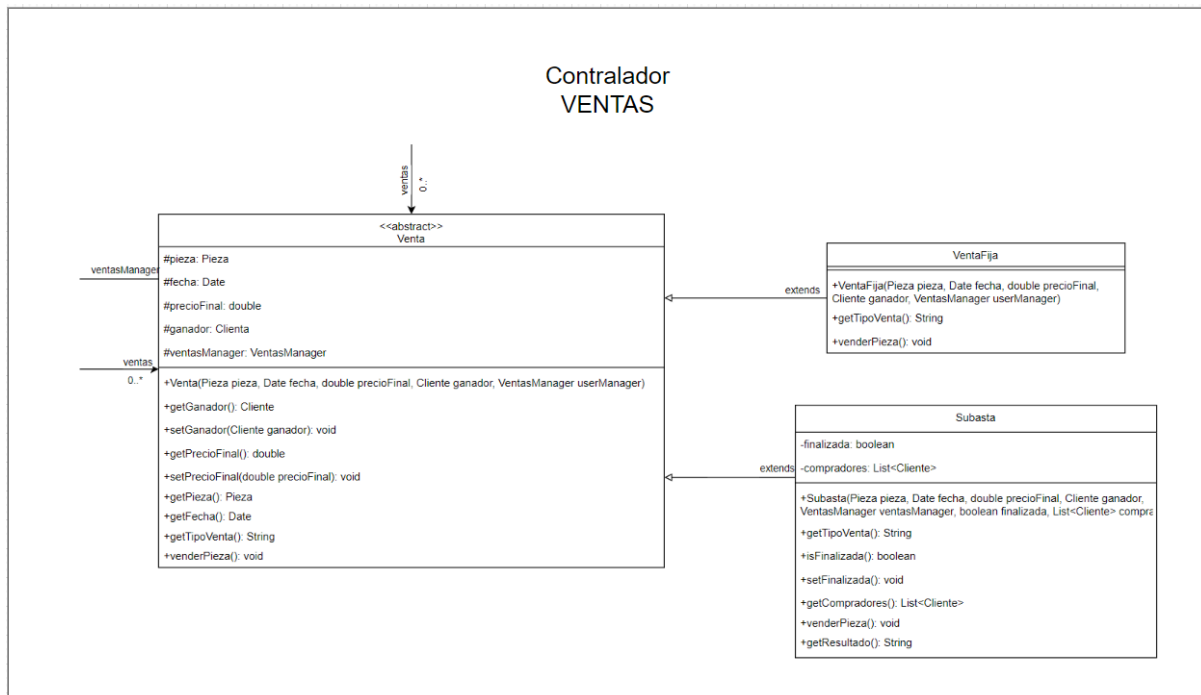
Las responsabilidades de PIEZA ya fueron tratadas en el material 2.2. Sin embargo, las colaboraciones de este componente, a pesar de no ser las más influyentes, resultan una herramienta clave para el relacionamiento de los componentes dentro de la aplicación. Así las cosas, el componente directo de relacionamiento con PIEZA es el *<<Information Holder>>* Inventario, pues este, requiere directamente del correcto funcionamiento de pieza para guardar la información importante sobre cada una de las piezas e indirectamente, se relaciona con el proceso de venta de las piezas; del correcto proceder de PIEZA depende que solo sean subastadas y vendidas, piezas que realmente estén disponibles, así como de la verificación de comprador acertada por parte del administrador.

## 3.4 Venta

### 3.4.1. Componentes candidatos y estereotipos

Venta es una clase padre con dos clases hijas, las cuales heredan de ella, atributos como la pieza, la fecha de la venta o la subasta, el precio final, el ganador de la subasta o de la venta, y el manager de dicha venta. Al ser una composición de clases que trabajan directamente con la información de sus clientes y la de sus piezas para poder llevar a cabo un proceso de adquisición de piezas, Venta se catalogó con el estereotipo *<<Controller>>*, pues lleva a cabo un debido proceso para llegar a la mejor solución posible.





### 3.4.2. Responsabilidades y colaboraciones

Las responsabilidades de Venta ya fueron enunciadas en la sección 2.2. Ahora bien, Venta representa el corazón de la aplicación al ser el proceso más requerido y en el que más hay interacción entre la galería y sus clientes. Venta trabaja de la mano con clases como Usuario y Modelo y resulta de gran importancia para llevar las funciones cotidianas de la Galería, como lo pueden ser, vender una pieza, subastar una pieza, brindarle información al comprador sobre la pieza en la que esté interesado, etc...

## 4. Diseño Final

Finalmente, después del proceso de análisis y diseño correspondiente, resolviendo problemas, inconsistencias y cuestiones inútiles en el diseño inicial, se obtuvo el siguiente diagrama como la versión final de la aplicación.

