

Ximena López – 202312848

Juliana Ferreira - 202312785

Sofia Losada – 202221008

Entrega 1/1

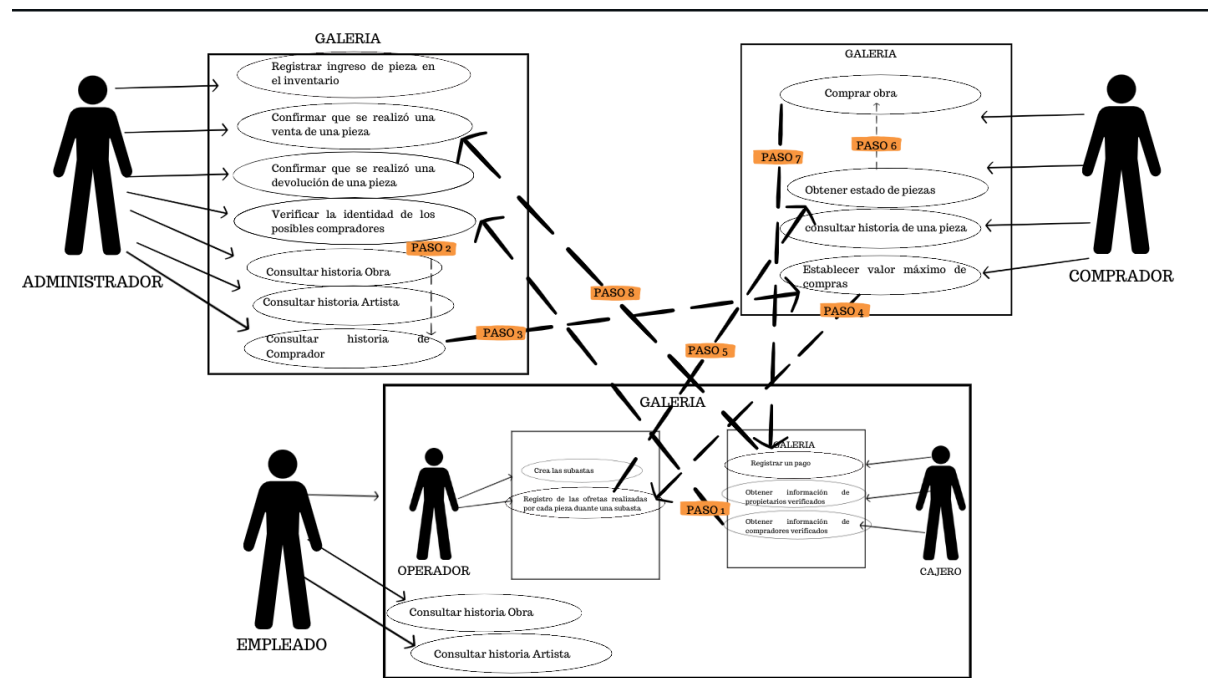
## Documento Diseño Proyecto 2

Se presentará un análisis de diseño detallado de las características, requerimientos y funcionalidades empleadas para el desarrollo de una aplicación de alta calidad que responde a las necesidades que surgen de las actividades de una casa de subasta.

En el diseño de solución presentado, tenemos como punto de partida tres grandes roles, entre los cuales se dividen las responsabilidades principales de: Administrador, Empleado constituido por Operador y Cajero, y Comprador. Para abordar todas las necesidades presentadas en el caso.

### Diagrama de Casos de Uso:

En el diagrama se presentan los tres roles principales y las respectivas funciones que ejecuta cada uno en la solución. Así mismo, se muestra mediante flechas, la relación entre las funciones de los diferentes roles, que constituyen un servicio. Se adaptó el modelo del Proyecto 1, mediante el agrupamiento de los roles operador y cajero, quienes ahora constituyen el rol Empleado. De esta manera se mantienen las responsabilidades de cada uno, pero comparten otras.



### Responsabilidades y Componente:

La relación que se establece entre las responsabilidades y componentes es dada el flujo de responsabilidades que se desarrolla para ejecutar la solución y responder a las necesidades de la galería.

#	Responsabilidad	Componente
1	Registrar ingreso de una pieza al inventario	Administrador
2*	Obtener información de una pieza	Empleado, Administrador o Comprador
3	Recibir oferta de compra	Empleado - Operador
4	Bloquear pieza	Administrador
5	Obtener información del comprador	Empleado-Cajero
6	Verificar comprador	Administrador
7	Acceder al historial del comprador	Administrador
8	Comprar pieza	Comprador
9	Registrar pago	Empleado-Cajero
10	Confirmar venta de la pieza	Administrado

2*	Obtener información de una pieza	Empleado, Administrador o Comprador
3	Crear subasta	Empleado-Operador
4	Recibir oferta de compra	Empleado- Operador
5	Verificar comprador	Administrador
6	Acceder al historial del comprador	Administrador
7	Registrar oferta	Empleado- Operador
8	Comprar pieza	Comprador
9	Registrar pago	Empleado-Cajero
10	Confirmar la venta de la pieza	Administrador

El asterisco denota el cambio en el proceso si es una subasta.

#### **Justificación de decisiones tomadas en el diseño de la aplicación:**

Se implementan tres roles principales, entre los cuales se distribuyen las responsabilidades y funciones que la solución debe abordar. El administrador es el único que puede realizar el registro de una pieza de nueva de arte en la galería, pero permite que los empleados y compradores tengan acceso a los atributos e información específica de cada pieza, para poder realizar tareas generales de inventario.

Así mismo, la solución aborda la compra y venta de obras, de manera que el operador crea una subasta y para validar la compra de alguna pieza el administrador valida la identidad del comprador, accede a su historial, para verificar si el comprador cumple las condiciones de compra basándose en sus antecedentes, revisa que el monto inicial de compra se cumpla, de manera que, si es validada, el operador registra la oferta, el cajero registra el pago y el administrador confirma que la pieza fue vendida.

Por otro lado, si la venta no ocurre en una subasta, el cajero es quien inicializa el proceso, valida que el pago se realice y que se guarde la transacción correctamente. De esta manera, el administrador congela la pieza mientras valida la identidad del comprador, para esto accede al historial del comprador y al ser verificada el cajero registra el pago. Finalmente, el administrador confirma la

venta de la pieza. Como aspecto transversal a los procesos de compra e inventario en la galería, los empleados, el administrador y los compradores tienen acceso al historial de la obra y los artistas.

Esta distribución se definió, teniendo en cuenta aspectos de efectividad y recursividad, ya que esta distribución de roles y funciones no solo aborda todas las necesidades del cliente, sino también permite establecer relaciones entre los roles, lo que permite un desarrollo completo y coherente de la solución.

Por otra parte, la persistencia se hizo inicializando desde la Galería. Sin embargo, los archivos se leen en las clases de los principales roles. En los roles asociados a Usuario, ya que respectivamente se cumplen los requerimientos. De esta manera, Administrador, Cajero, Operador, Propietario y Comprador, tiene su propia persistencia para que sus respectivos requerimientos independientes y relacionados se ejecuten de manera correcta. Lo anterior se hace con el fin de que la carga de esos datos sea de una vez en las clases que los usan, facilitando la ejecución. Por lo mismo, decidimos que la galería tuviera un inventario diverso.

### **Requerimientos de la Aplicación:**

#### **Requerimientos funcionales:**

##### **Administrador:**

- Debe poder cargar piezas desde un archivo de texto plano
- Debe poder crear Piezas y añadirlas al inventario
- Debe poder mostrar los **títulos** que se encuentran en el inventario
- Debe poder acceder a la historia de una pieza
- Debe poder acceder a la historia de un artista
- Debe poder acceder a la historia de un comprador:

**Empleado:** Como se mencionó anteriormente, se busca relacionar Operador y Cajero bajo el rol de Empleado, ya que en la solución desarrollada comparten funcionalidades, mientras que conservan las individuales.

- Debe poder acceder a la historia de una pieza
- Debe poder acceder la historia de un artista
- **Operador:**
  - Debe poder cargar subastas
  - Debe poder crear subastas
  - Debe poder cargar registros
  - Debe poder crear registros
  - Debe poder cerrar subasta y mostrar el nombre del nuevo dueño
- **Cajero:**
  - Debe poder crear venta
  - Debe poder cargar venta
  - Debe poder confirmar pago y mostrar el número de factura

**Registro y verificación de usuarios:** La aplicación requiere la aprobación del registro de usuarios y su posterior verificación por parte del administrador. Debe haber diferentes roles como vendedor y

gerente.

### **Requerimientos No funcionales:**

**Seguridad:** Se implementa medidas de seguridad para proteger su información personal y sus transacciones, incluido el uso de cifrado para almacenar contraseñas y realizar transacciones seguras.

**Persistencia:** Se implementa la función de almacenar en archivos de texto plano donde solo la aplicación puede escribir y leer de la carpeta donde se guarden.

**Portabilidad:** Su aplicación debe poder ser usable cuando se implemente la interfaz y que a su vez pueda manejar cargas de usuarios y transacciones para responder rápidamente a los tiempos de tráfico y al número de participantes.

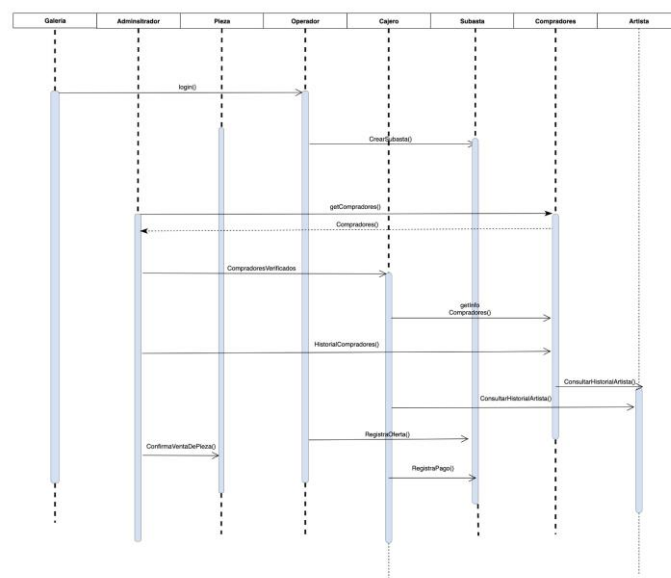
**Escalabilidad:** la arquitectura de la aplicación debe ser escalable, lo que permite agregar servidores o recursos adicionales según sea necesario para manejar el crecimiento en el número de usuarios y componentes administrativos.

**Compatibilidad:** Compatible con varios dispositivos y navegadores web para brindar una experiencia de usuario consistente.

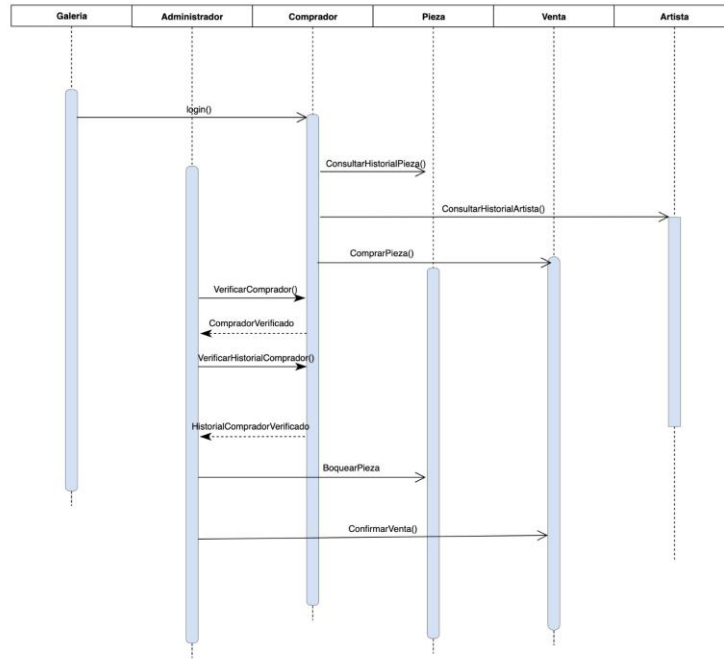
### **Diagramas:**

Como grupo decidimos que las funcionalidades críticas que se presentan en nuestra aplicación Galería y por lo cual creemos pertinente realizar los siguientes diagramas de secuencia:

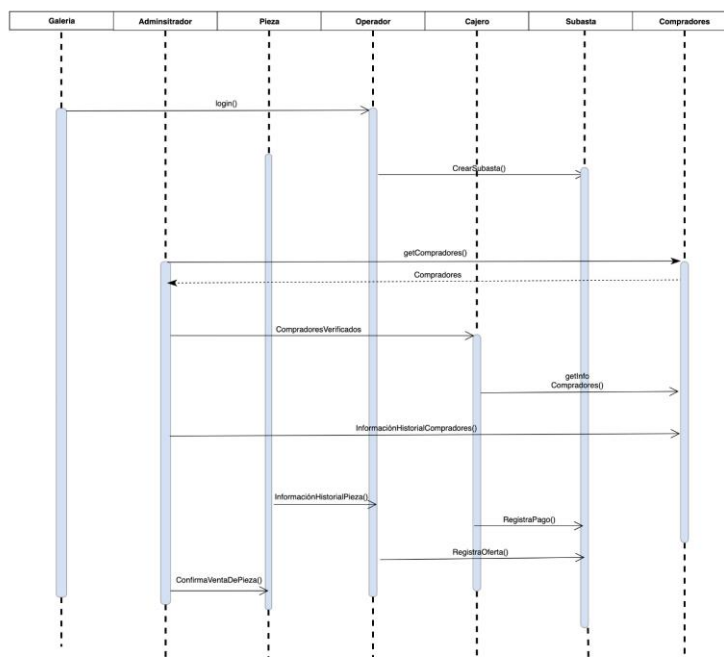
#### **1. Diagrama de Secuencia de Subasta:**



#### **2. Diagrama de secuencia de una Venta:**

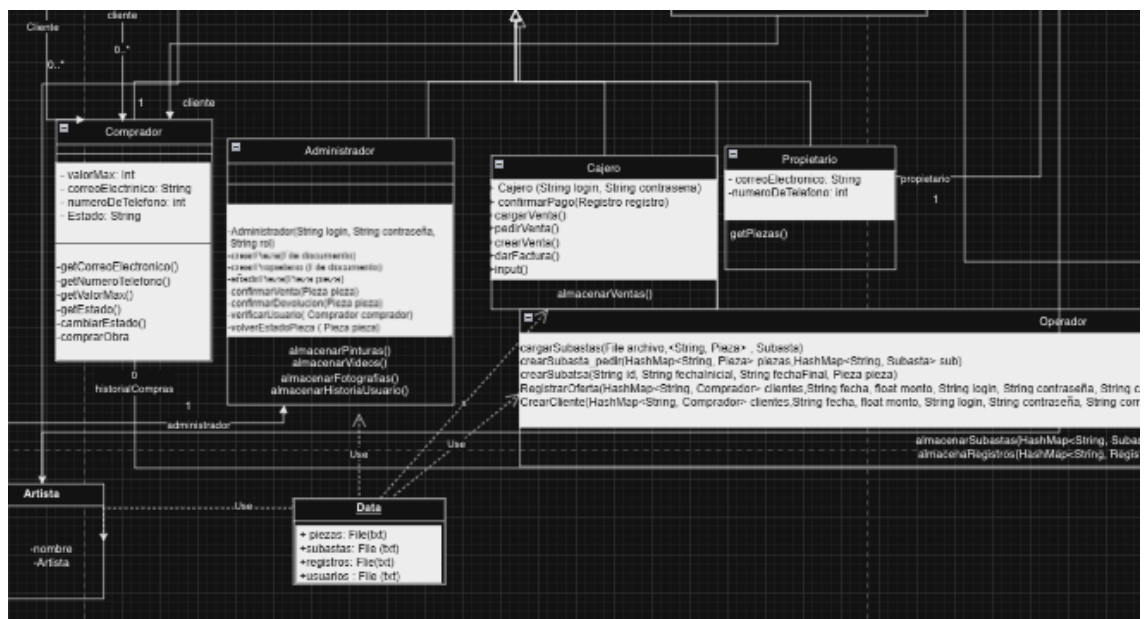
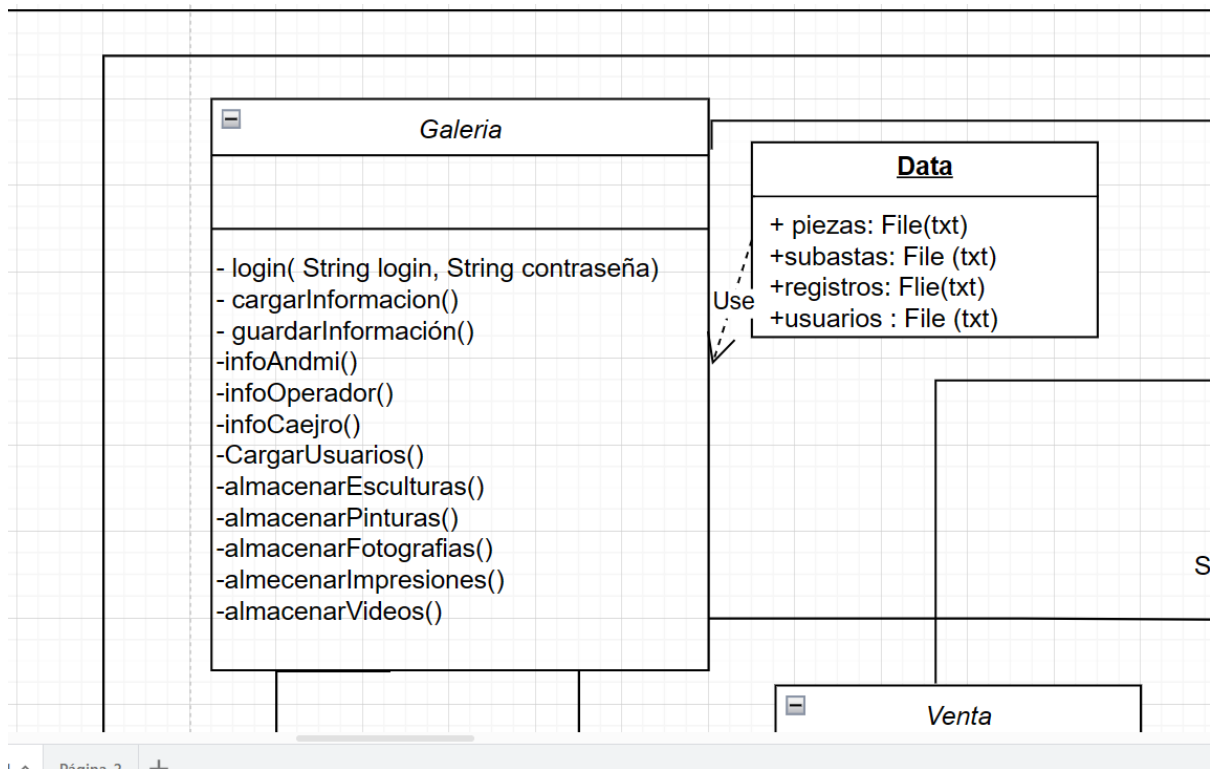


### 3. Diagrama de secuencia de una gestión en inventario:



### Diagrama de persistencia:

El siguiente diagrama es la persistencia para desde la aplicación poder ser almacenada. Esta se hace principalmente desde la Galería y como se mencionó las clases de usuario cuentan con su propia persistencia para realizar sus requerimientos.



### Diagrama de clases completo:

En el siguiente diagrama de clases UML se encuentran todas las relaciones de las clases con sus respectivos métodos y atributos: