Documento de diseño Grupo 1

Juan David Ortiz, Juan Nieves y Sebastian Vargas

Proyecto 2 entrega 1

Historias de usuario

Administrador:

Como administrador de una galería de arte, quiero organizar y revisar mi inventario de piezas y mis clientes, para poder controlar el flujo de ventas de la empresa.

Entradas: login, password, acción que quiere realizar, filtro que quiere utilizar

Salidas: Acceso aceptado o denegado, Mostrar piezas, confirmar cambios

Como empleado de la galería quiero ingresar fácilmente al sistema y realizar mis tareas rápidamente para que no se me haga tedioso ni se me acumule el trabajo

Entradas: login, password, pieza a actualizar

Salidas: Acceso aceptado o denegado, confirmación de cambios

Como Cliente de la galería, quiero estar enterado de las piezas a la venta para decidir si las agrego a mi colección.

Entradas: Login, password, selección de consulta a realizar, selección de pieza a comprar

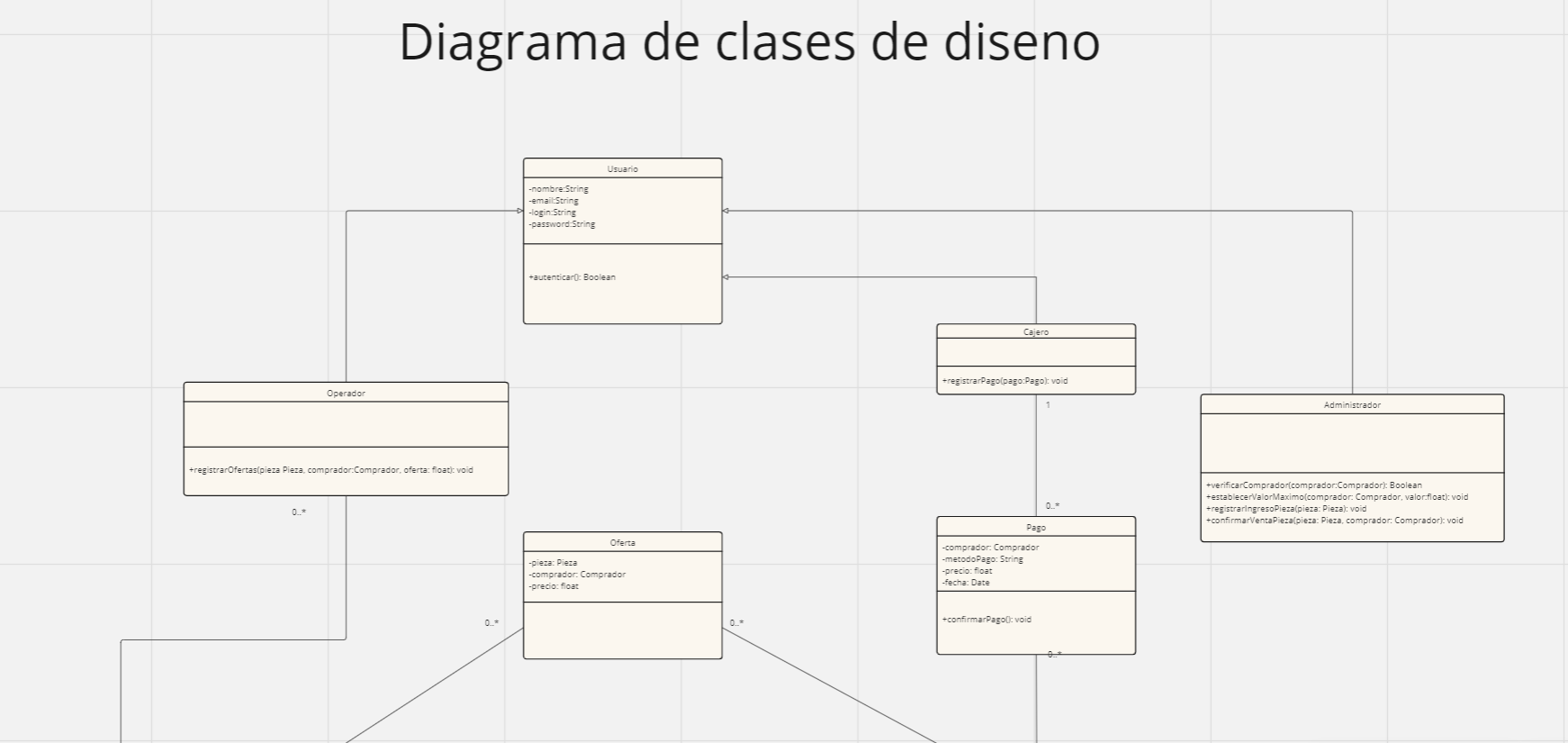
Salidas:Acceso aceptado o denegado, mostrar conjunto de piezas, confirmación de compra exitosa.

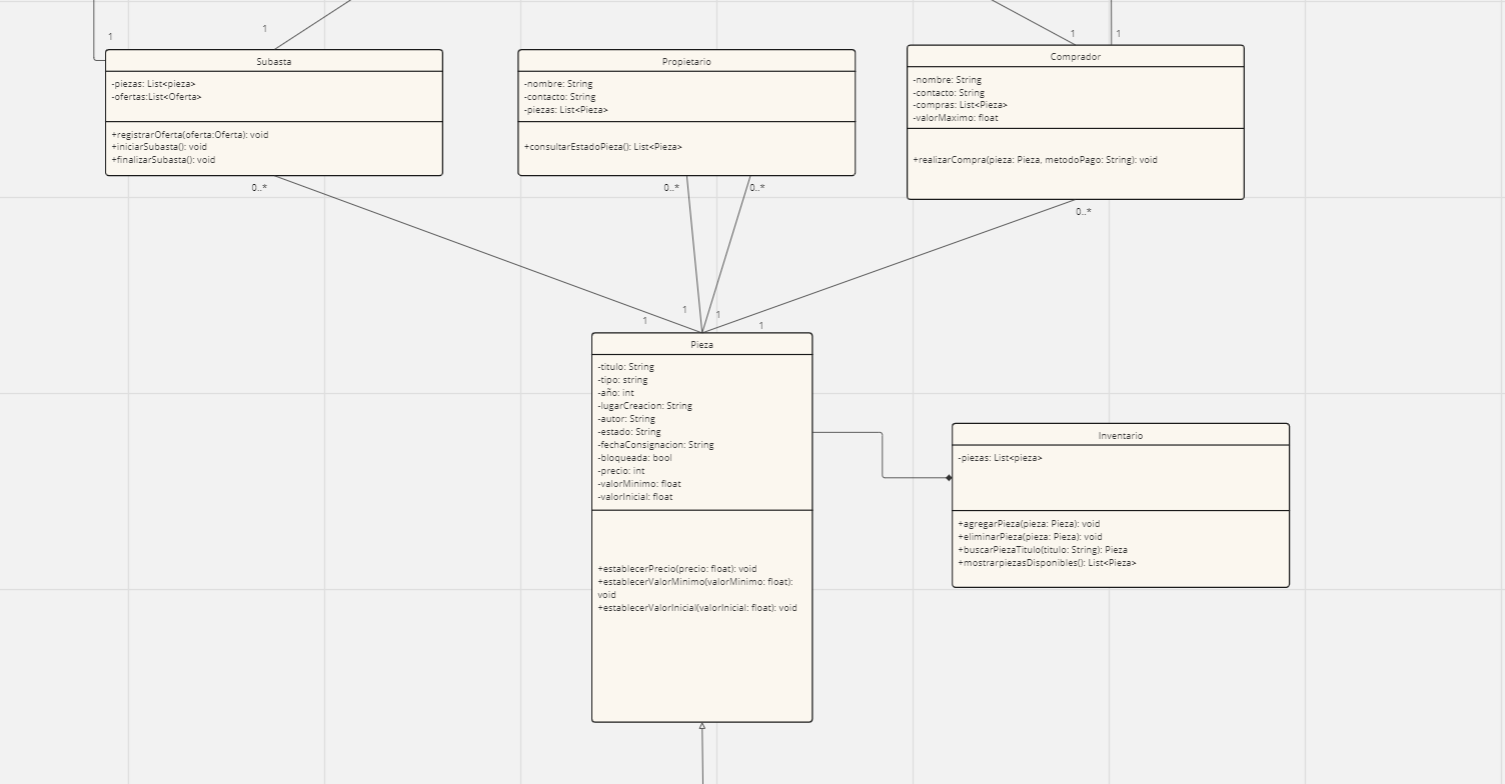
Cambios Introducidos

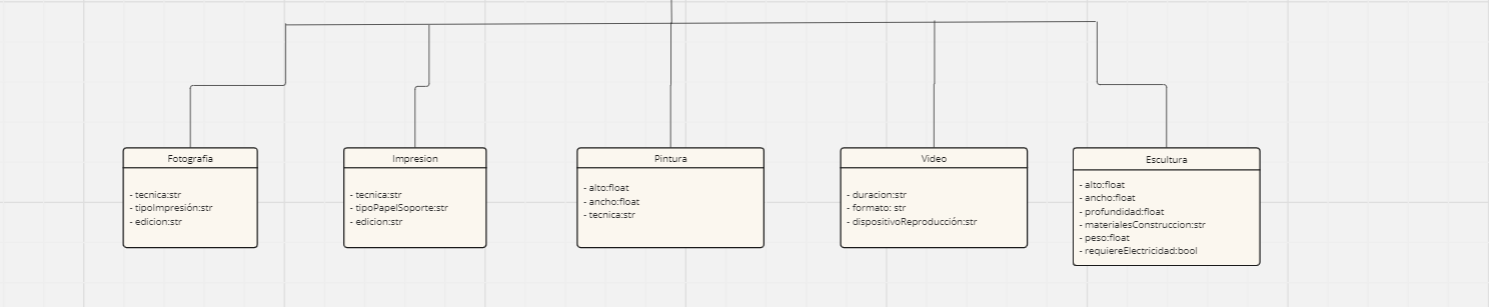
A partir de los comentarios del proyecto 1, decidimos estructurar el proyecto de la manera MVC para facilitar la forma de entenderlo y programarlo. En el modelo están las clases con las que se va a trabajar, siendo las piezas y los usuarios los más importantes y los roles para clasificarlos. En el view están las interfaces y a partir de una de inicio se divide en las funcionalidades de cada usuario tras aprobar su login. Se usa consultaUsuario para responder al inicio de sesión y consultaPieza para responder a todas las solicitudes de los usuarios. En el controller está la lógica del inicio de sesión y realizar las acciones que los usuarios esperan sobre el sistema. En data están los readers y writers para piezas y usuarios de los archivos csv que se encuentran en la carpeta ‘db’.

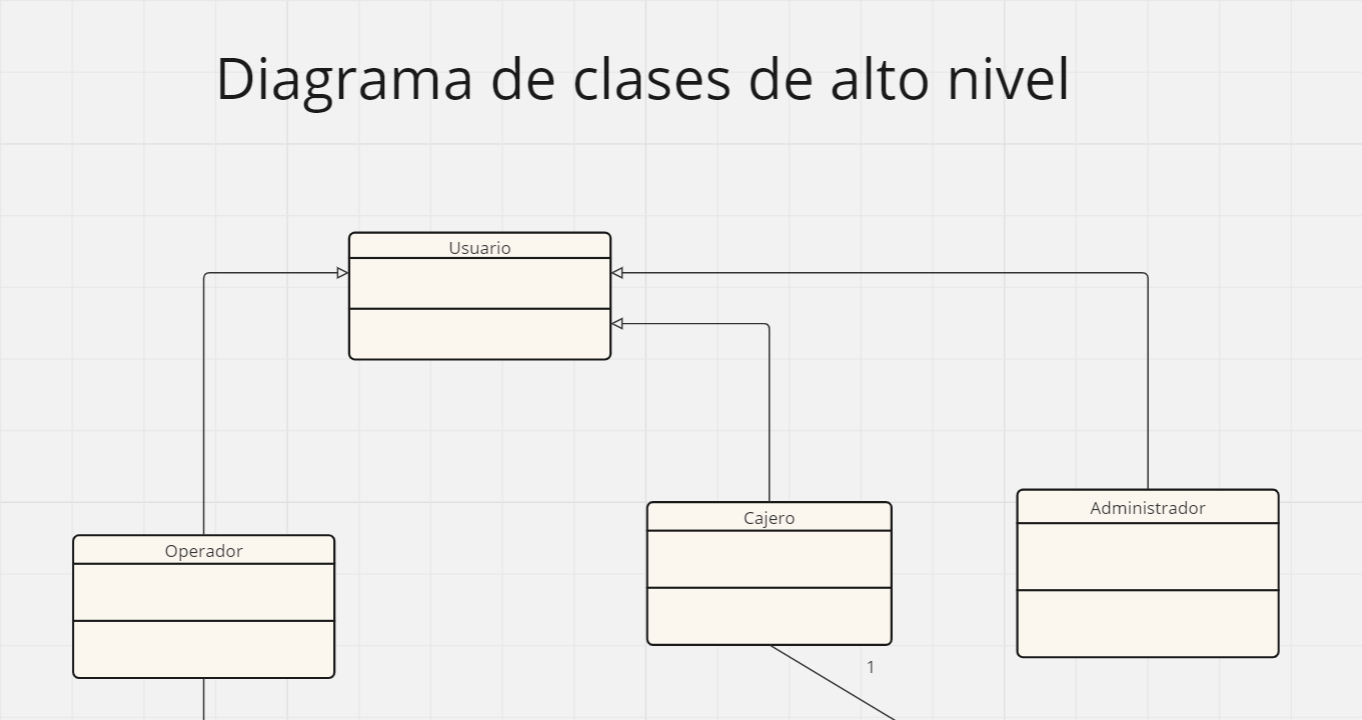
Ahora el programa se beneficia de las interfaces de cada usuario y del sistema simplificado de piezas para que las consultas y modificaciones sean mucho más sencillas de aplicar. También es un programa más sencillo de entender ya que al organizarlo en la forma MVC se agrupan varias clases y a su vez varios métodos, por ejemplo en las interfaces es más fácil que cada usuario llame a consulta de piezas para usar el método específico que necesite y esté a su vez use la lógica del pieces Controller.

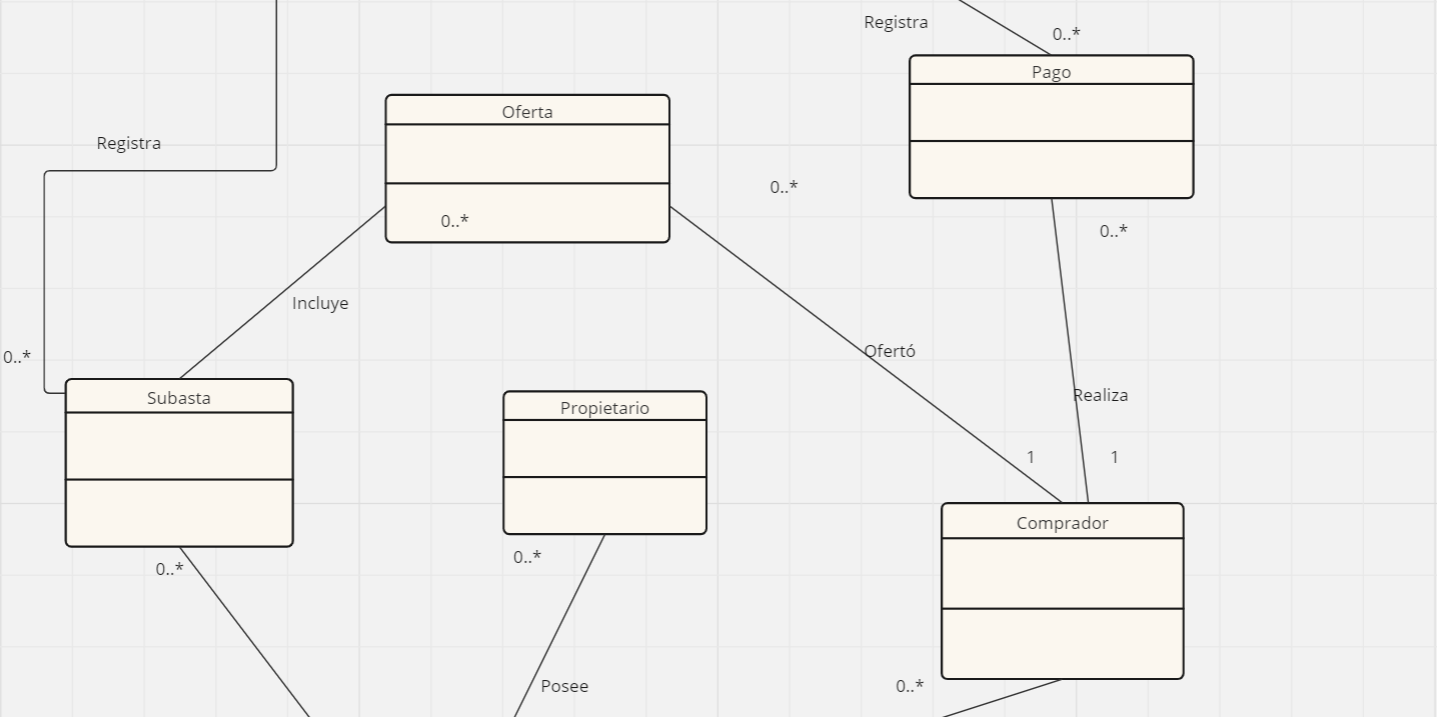
En la interfaz de inicio se selecciona por cual tipo de usuario quieres ingresar para así ofrecerle sus métodos individuales, a continuación una interfaz pide el login o correo y password los cuales se encuentran en el csv de users junto con sus roles. Una vez se comprueba el correcto login pueden acceder a ver las piezas y ver las de un artista concreto. El comprador puede consultar sus propias piezas ya sea que estén a la venta o las que tiene adquiridas, esto se comprueba con su login ya que en teoría debe ser único y está ligado a una propiedad de la pieza, además cuando ingrese el nombre de una pieza para comprarla esta cambiará su estado de bloqueada a true. El administrador además de consultar todos los usuarios también puede ver que piezas están bloqueadas para poder aprobar o denegar esa venta cambiando el estado de aprobado a true y una vez estén ambos estados como verdaderos la caja confirma el pago y actualiza el estado de pago a true para que el administrador confirme la venta de la pieza. Por último el administrador podrá ver las piezas que tenga un comprador.











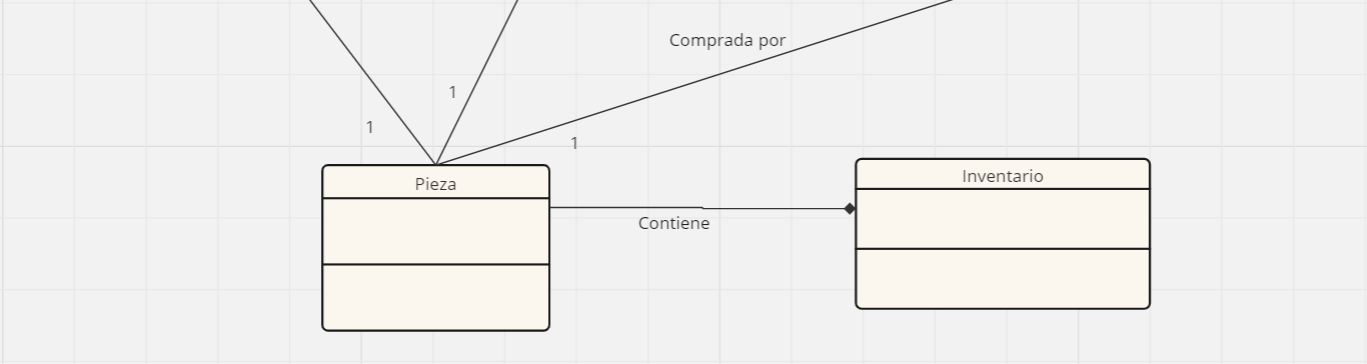


Diagrama de secuencia

