

Documento de diseño del Proyecto 1

Por David Calderón

PD: Este trabajo lo desarrolle sin tener más integrantes de mi grupo solicito no ser tan duros con la nota, principalmente en la implementación

Introducción y Justificación del método de control:

En términos generales, los requerimientos funcionales son traducibles en manejar (crear, acceder, editar y mantener) tipos abstractos de datos. Se propuso una división primaria en tres partes view, controller, model. En primera instancia, un *modelo* compuesto solo de *data Holders* con únicamente getters y setters representando los tipos abstractos de datos que se deben manejar (ej Pieza y Venta) en Pieza se permitió una división de herencia dependiendo del tipo de pieza tal como se pide en el proyecto, es decir una clase si la pieza es una pintura, una escultura, fotografía, etc que tienen como super clase Pieza, esto ayuda a manejar la complejidad del contenido que puede tener una pieza. En segunda instancia, un *controller* compuesto en principio de 6 clases, 3 de ellas representando cada una a un usuario (administrado, cliente y empleado) con estereotipo de *controllers*, en el sentido de que deben definir cómo manejar los datos según lo que quiera hacer cada usuario. Además, en este nivel se incluye varias clases con denominadas; BasesDatosGaleria, BasesDatosInventario y BaseDatosEmpresa, compuesta de mapas de objetos, informationHolder y metodos delegadores, que debe cargar, descargar y mantener los datos, como los métodos writer (que convierten de objeto a string las instancias de los objetos), los métodos Reader hacen lo opuesto. La idea era que cada miembro del equipo se encarga de un usuario y desarrolla las clases que puedan apoyar a su controller o dejar toda la implementación en esta dicha clase únicamente, a criterio de cada uno. Por último, el programa cuenta con una consola general y 4 auxiliares, una por cada usuario, que regulan la interacción con los usuarios mediante login y menús de acciones.

Persistencia:

La persistencia se maneja conjuntamente entre las tres clases BasesDeDatos y a través de archivos .txt. Hay un archivo txt por cada tipo abstracto de datos definido en el modelo, (ej pieza.txt, venta.txt). Cada vez que un usuario cierra sesión, se reescriben los archivos que enlistan las instancias de los objetos a través de Strings que permitan su posterior descarga y conversión en objetos. Por ejemplo, para cargar un Cliente, se transforman todos sus atributos individuales en string y se juntan en un orden determinado separados por punto y coma.

Cada vez que se inicializa la aplicación se descargan los datos, se crean los objetos y se añaden estos a los mapas. Se descargan todas las instancias de algún objeto y se almacenan en el map correspondiente antes de pasar al siguiente. Para esto se utiliza los métodos reader hallados en cada clase BaseDeDatos, que a partir de las String devuelve instancias de los objetos. Por ejemplo, al descargar el cliente, primero se crea este a partir de sus atributos primitivos, con el nombre de usuario en el mapaClientes y se agrega. Para que funcione correctamente es necesario que los constructores funcionen solo con datos primitivos y que los demás se puedan agregar con sets.

La interacción con el usuario se maneja a través de la consola. La implementación de estas interfaces se divide en 4 partes principales. Para ingresar se le pedirán al usuario los datos que aparecen y se buscará si coinciden, si no, no dejara ingresar. Cada una de estas tiene los requerimientos de cada usuario; Administrador , Cliente, Empleado. Al administrador se le pedirá que ingrese una nueva pieza o que registre una compra, confirme una venta, verificar el comprador y crear una nueva subasta;

El cliente tiene la opción de comprar ingresando el titulo de la pieza , el administrador también tiene de ser un cajero y recibir las compras que se hagan , muchos de estos métodos quedaron incompletos por obvias razones, O incluso no se implementaron.

Aquí se puede observar consola cuando se ejecuta desde eclipse y se escoge la opción del Login:

```
Console X
Interfaz [Java Application] C:\Users\Dell\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0
Casa de Ventas y Subastas

## Bienvenido a la Casa de Subastas ##

1. Ingresar como Cliente
2. Ingresar como Empleado
3. Ingresar como Administrador
4. Salir de la Aplicacion
Por favor seleccione una opción: 3

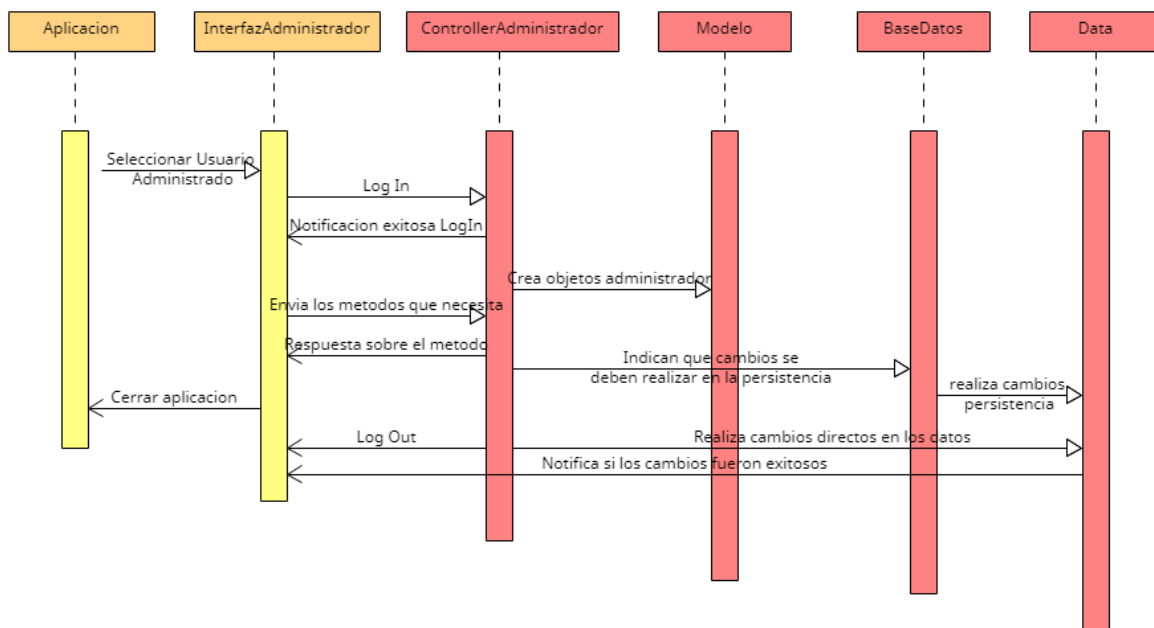
Console X
Interfaz [Java Application] C:\Users\Dell\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0
Bienvenido Administrador

Opciones de la aplicación

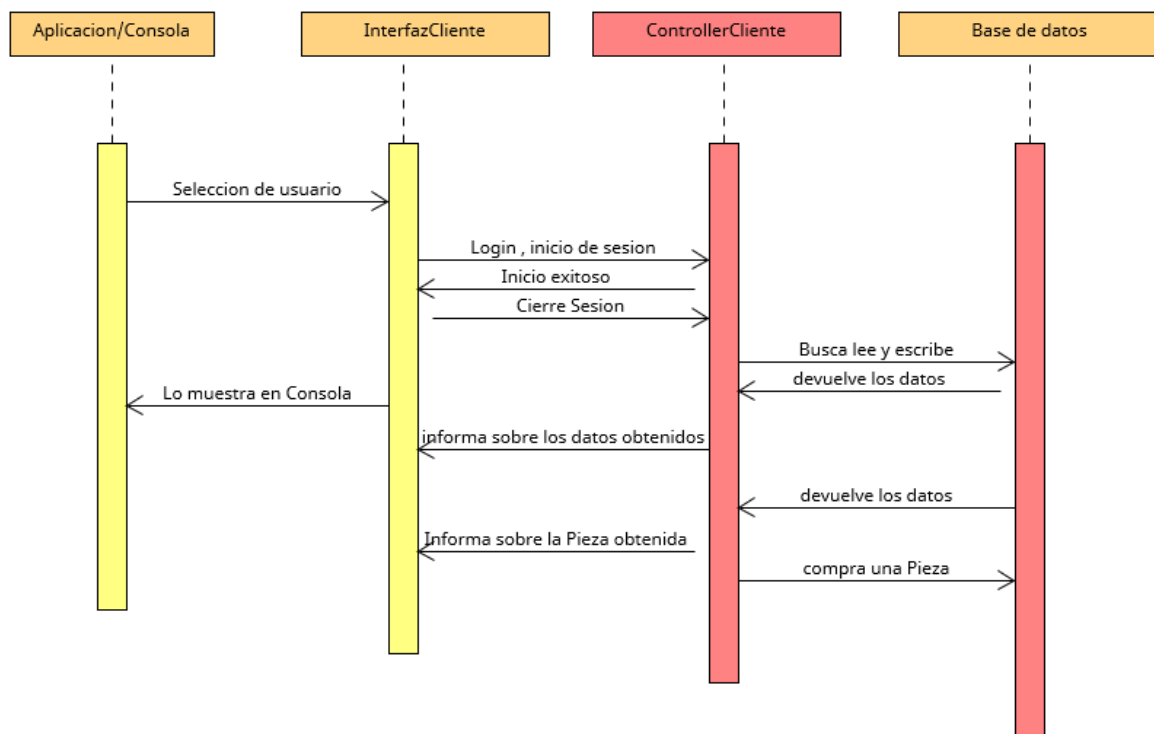
1. LogIn
2. Salir de la aplicacion
Por favor seleccione una opción: 1
Usuario: : dav
contraseña: : 1234
Ingresado correctamente
```

Historia de Usuario

Administrador:



Comprador, Cliente



Historia Empleado

[illegible]