

PROYECTO 2 – DOCUMENTO DE DISEÑO

Integrantes: Juan Pablo Delgado, Andres Mateo Molano, Julián David Contreras

1. Contexto del proyecto:

Funcionalidades principales:

- Inventario de las piezas disponibles
- Sistema de gestión para subastas o compras de pieza(s)
- Manejar la información de los compradores y propietarios de piezas

Al iniciar el diseño es útil identificar las principales funcionalidades del proyecto a desarrollar, y tras identificarlas se quiere crear un sistema para una galería de piezas, con este sistema se busca cumplir todas las acciones pertinentes en esta galería, como tener un inventario, subastar una pieza o venderla entre otros. Al tener en cuenta que es un sistema de compras y ventas que maneja un inventario dinámico para manejar información de las piezas, empleados y compradores y propietarios, podemos formular el siguiente comportamiento esperado del sistema en la Figura 1:

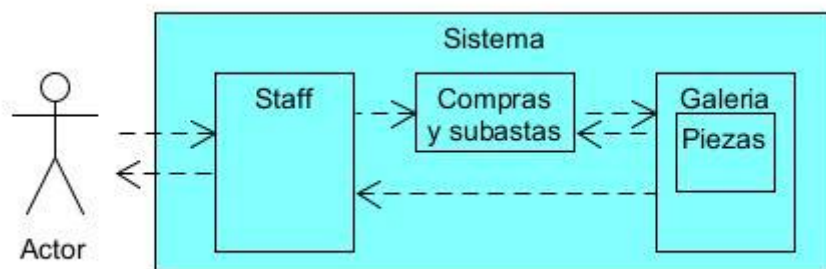


Figura 1

En este diagrama se muestra cómo se esperaría que fuese la interacción básica entre un actor y el sistema, cabe mencionar que los únicos actores que deberían tener acceso al sistema son los clientes (compradores y propietarios) y los empleados (staff) de la galería. En esta interacción El actor (cliente) se comunica con el sistema por medio del staff (todos los empleados que hay en la galería) y al hacer esto este último se comunica con compra y subastas donde se encuentra toda la lógica que opera el manejo de subastas y de piezas con precio fijo, así mismo este se encarga de hacer la comunicación con el inventario (galería) para obtener la información de las piezas.

2. Nivel 1

2.1 Componentes candidatos y estereotipos

1. La primera parte es el molde de las piezas, en donde se identifico que es posible tener dos tipos de piezas; piezas que se pueden exponer de manera normal como una pintura o escultura, y piezas que tienen un display diferente al ser video, fotos, proyecciones, etc. Con esto en cuenta se decide hacer una clase madre Pieza y dos hijas PiezaFisica y PiezaVirtual respectivamente, en este caso ambas serán <<Information holder>>.
2. Para manejar el inventario se creará la clase galeria en la cual estará la información de todas las piezas que están disponibles (piezas en subasta y piezas con precio fijo), asimismo aprovechamos esta clase para guardar la información de los compradores y propietarios. Al ser un estructurador de información su estereotipo es <<Structurer>>.
3. Los medios por los cuales se verifican las ofertas se hacen efectivas las compras, y se registra historial de una subasta, vienen por el paquete de Staff en el cual están todos los tipos de empleados; administrador, operador y cajero, estas verificaciones dependen de los roles por lo cual tienen el estereotipo <<Service Provider>>.
4. La lógica detrás de las subastas y compras se manejará a partir de las clases controladorSubasta y controladorOfertasFijas <<Controller>>, es aquí donde las ofertas llegan y se manejan, por ello es donde se revisan y verifican las ofertas (con el Staff), se escoge al ganador según la oferta más atractiva, y se hace efectiva la compra por medio del Staff, paquete que tiene diferentes responsabilidades.

La Figura 2 presenta los cuatro componentes candidatos identificados sin sus estereotipos ya que han sido mencionados anteriormente (en este diagrama de alto nivel no se tiene en cuenta la persistencia por motivos prácticos).

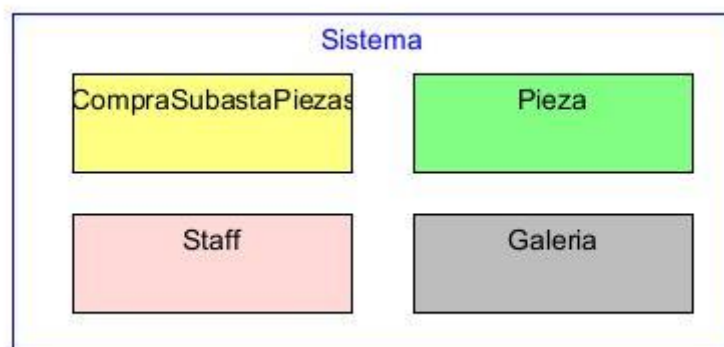


Figura 2 : Uml alto nivel sin estereotipos

Persistencia:

2.2 Responsabilidades

Algunas de las responsabilidades que maneja cada aspirante quedan implícitas al evaluarlos. Cada responsabilidad se enumera en su totalidad junto con el componente relacionado que se requiere para cumplir la siguiente tabla.

Tabla 1: Asignación de responsabilidades

#	Responsabilidades	Componente
1	Autenticar los usuarios	Staff
2	Gestionar las transacciones de la galería	Staff
3	Verificar a los compradores de la galería	Staff
4	Registrar el ingreso de nuevas piezas a la galería	Staff
5	Llevar la traza de las ofertas realizadas por cada pieza	Staff
6	Verificar la oferta de una pieza	Staff
7	Administrar el inventario de la galería	Galería
8	Conocer el estado de una pieza	Galería
9	Subastar pieza	Compras y subastas
10	Asignar oferta más alta de una subasta	Compras y subastas
11	Permitir a los compradores verificados hacer ofertas en subastas	Compras y subastas
12	Permitir a los compradores ofertar por las piezas	Compras y ventas
13	Mostrar a los compradores las piezas a subastar y con precio fijo	Compra y subastas
14	Cargar las piezas y usuarios de la galería	Persistencia
15	Almacenar la información del inventario, compradores y propietarios	Persistencia

Tabla 1: Responsabilidades

2.3 Colaboraciones

Después de definir todas las responsabilidades podemos pasar a analizar las posibles colaboraciones que se presentaran en el sistema:

- Colaboración entre CompraSubastaPiezas y staff

Esta colaboración quizá es la más importante del sistema, ya que permite que los empleados puedan manipular los estados de las ofertas y piezas para cumplir con la mayoría de las funcionalidades del proyecto. También se entiende que son paquetes complementarios ya que ninguno podría funcionar con el otro.

- Colaboración de galería:

Por otro lado, en cuanto a la galería este componente presenta colaboraciones con casi todos los demás componentes debido a la estructuración de la información para poder tener un inventario dinámico.

Figura UML de alto nivel: clases y relaciones

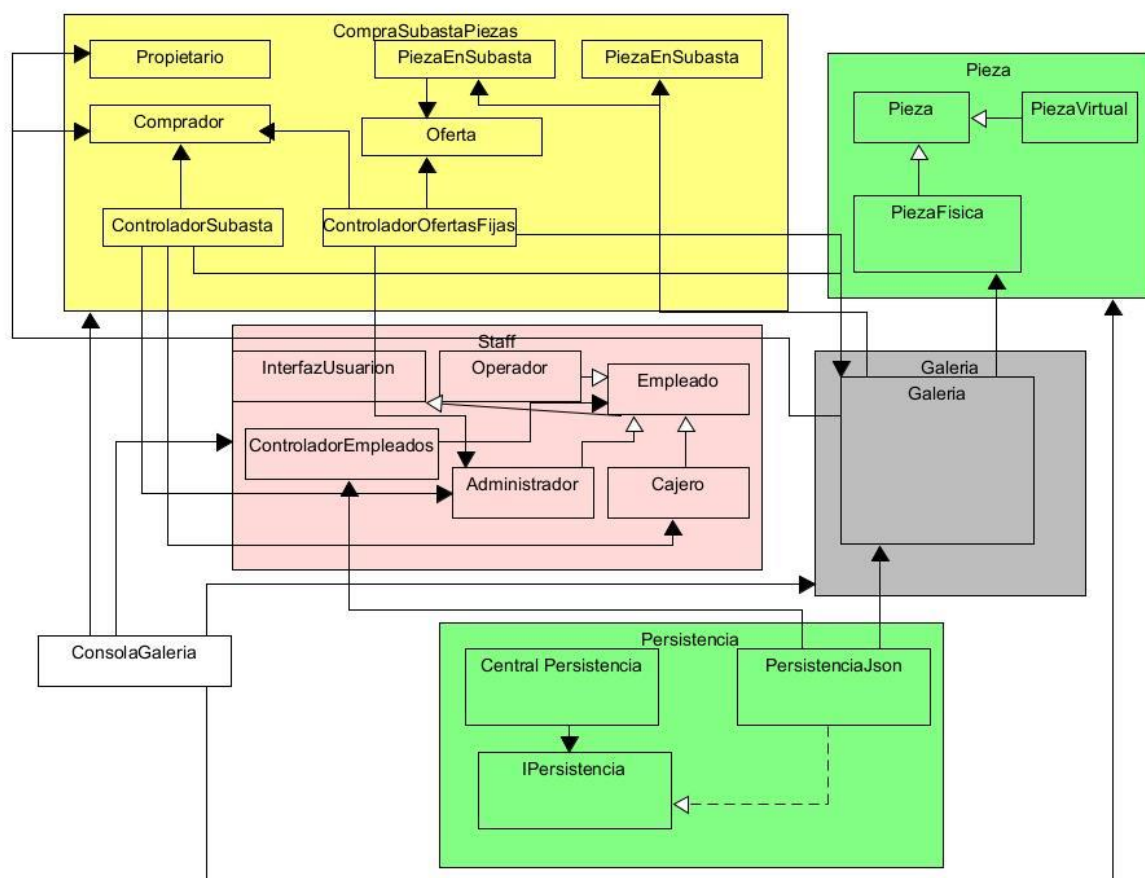
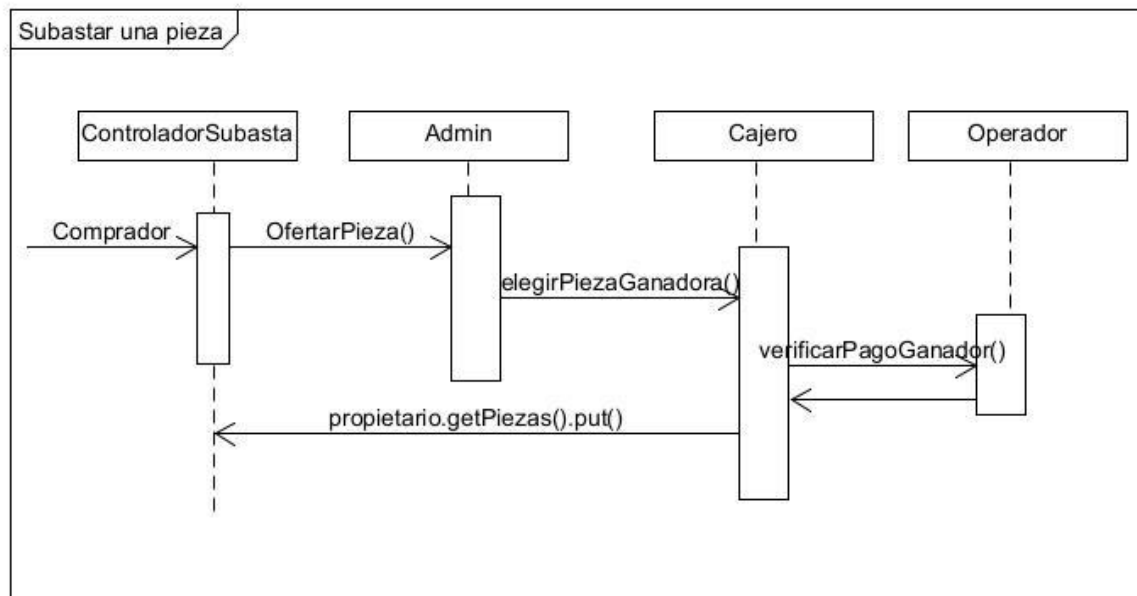


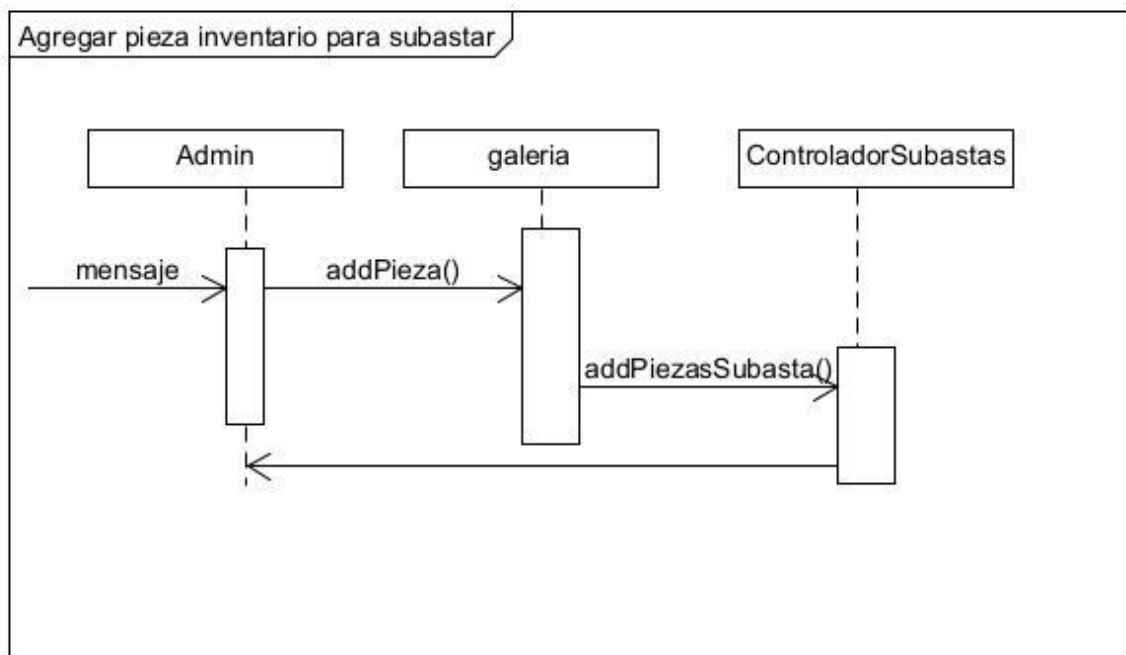
Figura 3

Diagramas de secuencia para funcionalidades críticas:

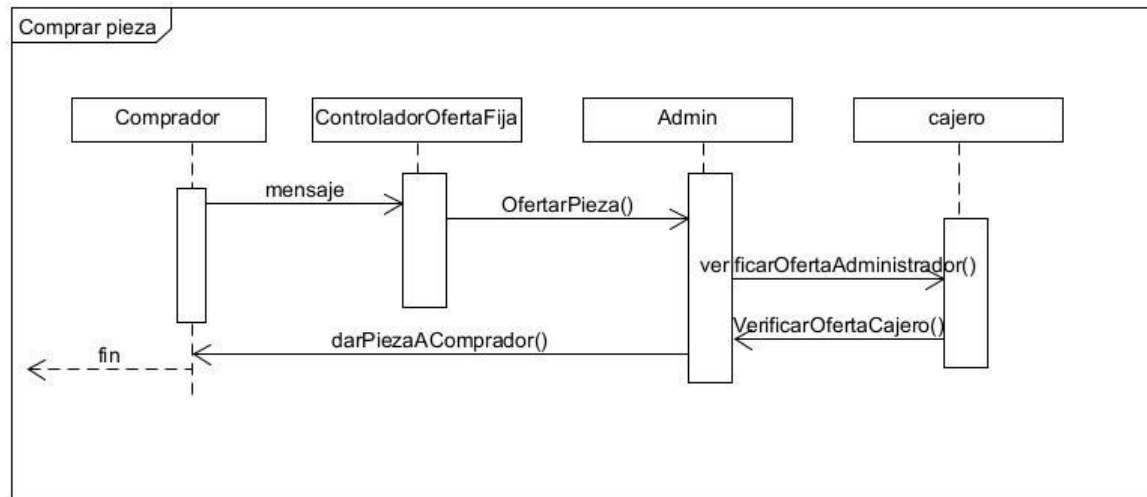
- Subastar una pieza:



- Agregar pieza al inventario



- Comprar pieza (precio fijo)



Uml clase alta (paquetes):

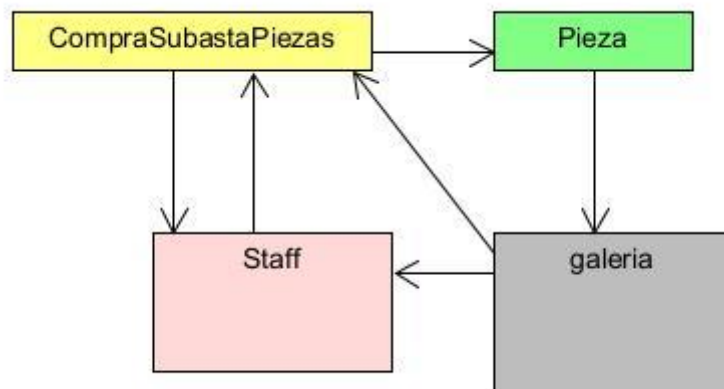


Figura 4

En la figura 4 se puede apreciar cómo se esperan que se relacionen los paquetes entre sí, en primera instancia se usan los moldes del paquete pieza para poder cargar la información a la galería en donde se estructuran todos estos para que luego puedan ser utilizados en la parte de comprasSubastaPieza en donde se relacionan con staff para poder avanzar hacia la compra y subasta de las piezas.

3. Nivel 2

Siguiendo el proceso de diseño por niveles, se procede a descomponer cada uno de los componentes perfilados en el nivel anterior: (pieza)

3.1 Pieza

Responsabilidades: La principal responsabilidad de este componente es ser el molde de la información de las piezas de la galería.

EL paquete designado como pieza funciona para poder expresar toda la información básica de las piezas como sus dimensiones o detalles importantes para su exposición en la subasta las clases de

este paquete son <<Information Holders>>, la información concreta de las clases se puede apreciar en la siguiente Figura:

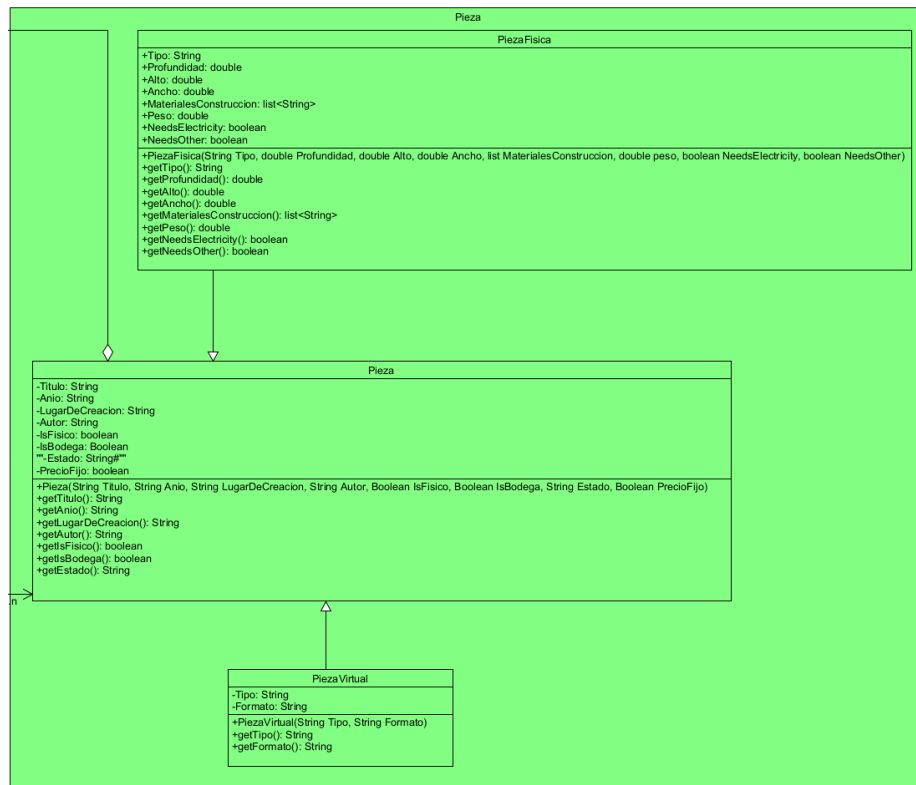


Figura 5: Paquete Pieza

3.2 Paquete Galería:

Responsabilidades: Administrar el inventario de la galería y Conocer el estado de una pieza.

El paquete galería cuenta con una única clase llamada “galería”, esta clase es el corazón de la aplicación ya que es aquí donde se estructura la información de muchos componentes, cumpliendo con las funciones de un inventario, en específico se guarda la información de todas las piezas, compradores y propietarios, por ello se le da el estereotipo de <<Structurer>>.

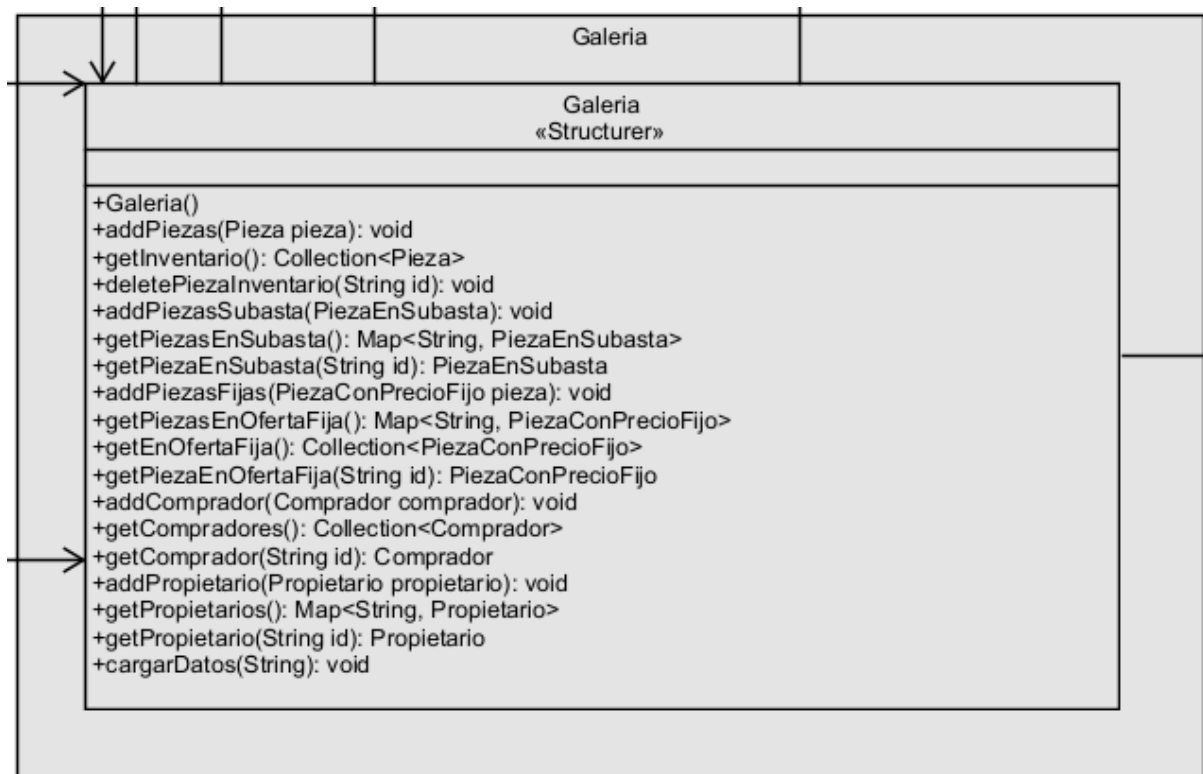


Figura 6

3.3 Paquete Staff:

Responsabilidades: Autenticar los usuarios, Gestionar las transacciones de la galería, Verificar a los compradores de la galería, Registrar el ingreso de nuevas piezas a la galería, Llevar la traza de las ofertas realizadas por cada pieza y Verificar la oferta de una pieza.

Este paquete tiene las clases correspondientes a los empleados de la galería, en donde de identifico diferentes tipos de empleados para ello creamos una clase madre(empleado) de la cual heredan operador, administrador y cajero los cuales tienen diferentes funciones a la hora de manipular las ofertas. Debido a que este paquete se encarga de manipular diferentes estados en el sistema se puede considerar que presta un servicio como tal por ello este paquete principalmente se identifica con el estereotipo de <<Service Provider>>.

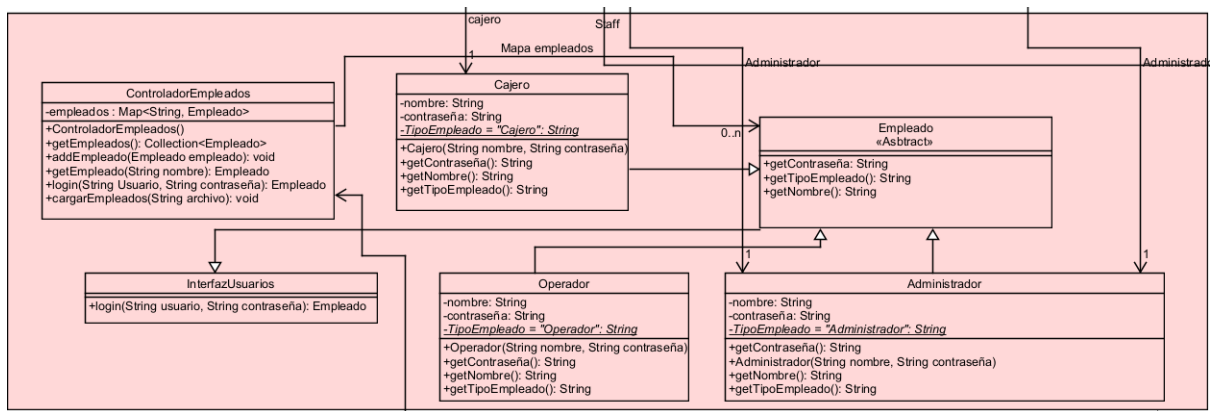


Figura 7

3.4 Paquete Compra y subastas:

Responsabilidades: Subastar pieza, Asignar oferta más alta de una subasta, Permitir a los compradores verificados hacer ofertas en subastas, Permitir a los compradores ofertar por las piezas, Mostrar a los compradores las piezas a subastar y con precio fijo.

En este paquete es donde se encuentra la mayor parte de la lógica del sistema, en primer lugar se tiene las clases de compradores, propietarios y ofertas los cuales funcionan como <<Information holders>>, por otro lado, tenemos las clases PiezaEnSubasta y PiezaConPrecioFijo las cuales son clases que se usan para dividir todas las piezas que se encuentran en la galería por ello también son <<Information holder>>. En las clases de controladorSubasta y controladorOfertasfijas se manejan las ventas de las piezas, aquí las piezas cambian de estado hasta ser compradas por un comprador, por ello tienen el estereotipo de <<Controller>>.

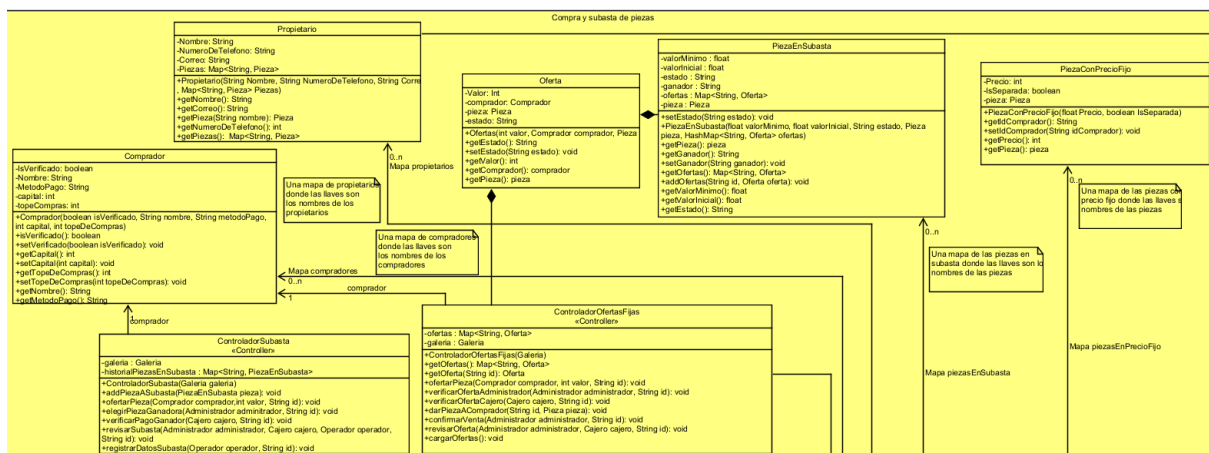


Figura 8

UML final del diseño (relaciones, atributos y métodos)

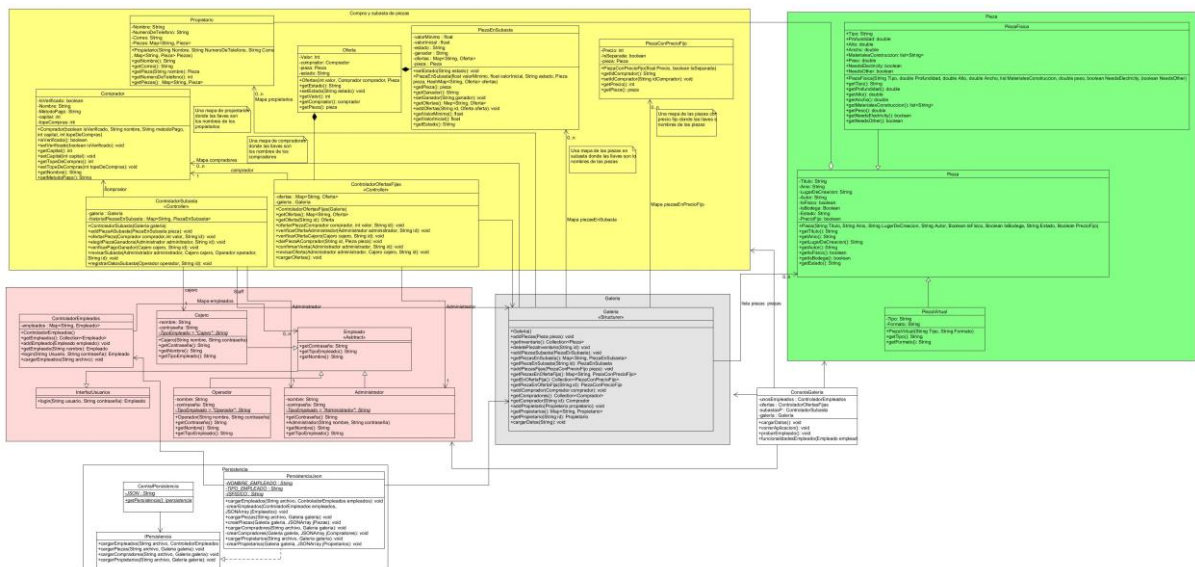


Figura 9

Restricción de proyecto:

Para el correcto funcionamiento del sistema es necesario descargar un componente extra a eclipse; un json.jar el cual se debe descargar y meter en la sección de propiedades del proyecto, en la parte de java build path>>libraries y en la sección de Classpath hay que quitar el que esta determinadamente y poner el archivo mencionado (json.jar) (add externa JARs) este archivo se adjuntará.

Descripción parcial del funcionamiento de la aplicación:

Nota: La persistencia es en donde se guardará la información de la galería, en donde se ubican todas las piezas de la galería.

Piezas:

Las piezas pueden ser de dos tipos; físicas y virtuales, así se satisfacen las características de las posibles descritas en el contexto del proyecto, pero al agregar piezas a la galería se dividen en 2 categorías diferentes; en piezas con precio fijo o piezas en subasta las que pueden adquirir los compradores mediante ofertas.

Un propietario puede tener un número n de piezas, pero no porque todos tengan piezas significa que todas tengan un propietario asignado, las que no tienen se consideran de autor anónimo.

Cuando un comprador haga una oferta, este primero tiene que escoger una pieza de la galería y dependiendo del tipo de pieza (en subasta o con precio fijo) se dirige a diferentes componentes, si es una en subasta esta oferta va a clase subasta y en esta clase es donde la oferta más atractiva se elige, por otro lado, si es una pieza con precio fijo al consolidarse (la oferta), la pieza quedara bloqueada hasta que el administrador evalúe la oferta (si es rechazada, la pieza vuelve a su estado original).

Para ser comprada una pieza como se sabe primero debe tener una oferta, ya que esta oferta al ser evaluada por el administrador se puede efectuar, y el cajero es el que la hace efectiva con la información del comprador.

Persistencia: la persistencia de este proyecto se maneja con objetos json en donde hay información sobre los empleados de la galería, las piezas, compradores, propietarios, etc. En el paquete de la persistencia es donde se cargan y guardan los datos de la galería.

4. Nivel 3 : Requerimientos funcionales nuevos

- Historia de una pieza
- Historia de un artista
- Historia de un comprador

Para poder implementar las historias en el sistema fue necesario cambiar un par de cosas en el código fuente del sistema como parte de la persistencia, cambios en los compradores y en los controladores.

Cambios en el código:

Historias de usuario:

1. Historia de pieza

rol	Pieza (objeto principal en la subasta)(pintura de pedro)
objetivo	Poder mostrar la historia de una pieza
propósito	Registrar las transacciones a las cuales la pieza ha estado
Criterio de aceptación	-tener un historial por cada pieza -poder mostrar dicho historial a los compradores y empleados

- Tomando el rol de una pieza de la galería, debo ser comprada y para esto los compradores pueden hacer ofertas para comprarme, a medida que me compran y me vendan adentro de la galería, y se ha vuelto necesario tener un historial para que los compradores puedan verlo.

2. Historia de artista

- Como cada pieza tiene un autor en la galería se vuelve pertinente saber las piezas que ha hecho cada artista, por ello, llega a ser necesario poder desplegar una historia (piezas que ha hecho, cuándo, cuándo han sido vendidas y por cuánto) de las piezas por artista a los compradores y empleados.

Como	Comprador (también empleados)
quiero	Poder ver la historia de un artista (de sus piezas)
Para así	Poder ver toda la información sobre una pieza
Criterio de aceptación	- Tener una historia de las piezas por artista - Poder mostrar la historia a los compradores y empleados

3. Historia de un comprador

- Jorge es un administrador de la galería y se ha vuelto necesario conocer la información de los compradores para que así este pueda predecir cuales piezas son más atractivas, y para esto es necesario tener una historia por cada comprador de la galería.

rol	administrador
objetivo	Poder ver la historia del comprador
propósito	Registrar información del comprador
Criterio de aceptación	-tener un historial por comprador -mostrar dicho historial al administrador

Historias de usuario para los tres usuarios principales:

Administrador:

Rick es el administrador de la galería y como trabajo debe vender la mayor cantidad de piezas posibles en la galería y para ello se ha vuelto necesario conocer 3 informaciones esenciales para su objetivo; tener la historia de los compradores, la historia de las piezas y la historia de los artistas; y para poder ver esta información el administrador usa el sistema.

Entradas:

Para poder entrar al sistema, se deben ingresar las credenciales una vez hecho esto, el administrador podrá añadir una pieza, revisar las ofertas tanto de subastas como de ofertas fijas, consultar la historia de una pieza, de un artista y de un comprador.

Salidas:

Al momento de añadir una pieza el sistema agregara la pieza a la galería, y al momento de consulta alguna historia el sistema le arrojará una cadena de la historia, al revisar una oferta, si es exitosa cambiara de estado, si no se eliminara la oferta

Empleados: Juan es un empleado de la galería y ha trabajado por más de 20 años allá por lo que ha pasado por todos los roles que hay en esta, en su trabajo es necesario controlar la galería y las transacciones que hay acá, y con el propósito

de cumplir mejor su trabajo es de utilidad conocer la historia de las piezas que se venden y la historia de los artistas que hacen las piezas. Ahora con el sistema Juan puede acceder a dicha información por medio de solo su login en el sistema.

Entradas:

Para poder entrar al sistema, se deben ingresar las credenciales una vez hecho esto, el administrador podrá añadir una pieza, revisar las ofertas tanto de subastas como de ofertas fijas, consultar la historia de una pieza y de un artista.

Salidas:

Al momento de añadir una pieza el sistema agregara la pieza a la galeria, y al momento de consulta alguna historia el sistema le arrojara una cadena de la historia, al revisar una oferta, si es exitosa cambiara de estado, si no se eliminara la oferta

Comprador: Hugo es un millonario que se dedica a coleccionar piezas de una galería que conoce, para hacer una mejor predicción de cual pieza es más atractiva en terminos de valor monetario se ha vuelto necesario que Hugo pueda ver la historia de cada pieza para juzgar mejor. Con el sistema en el que Hugo esta registrado puede ver dicha historia por pieza.

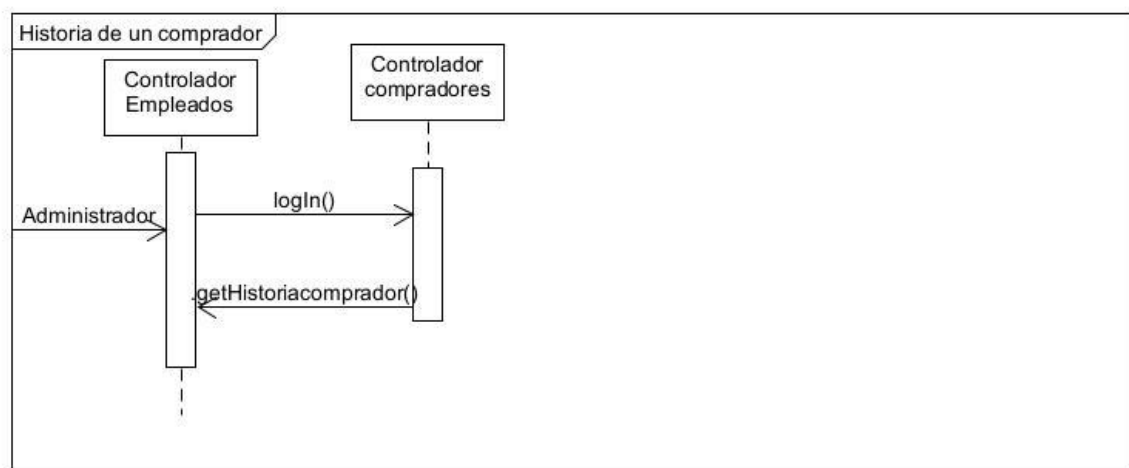
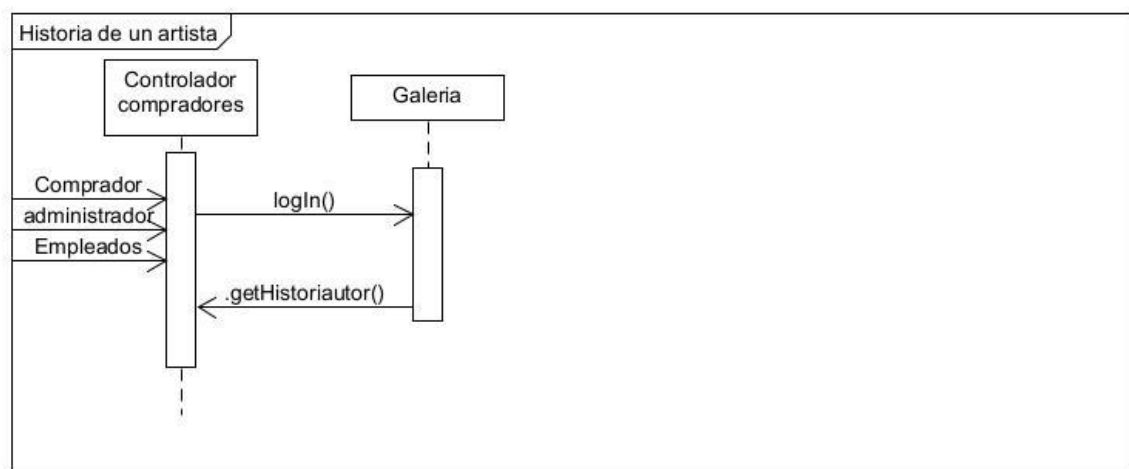
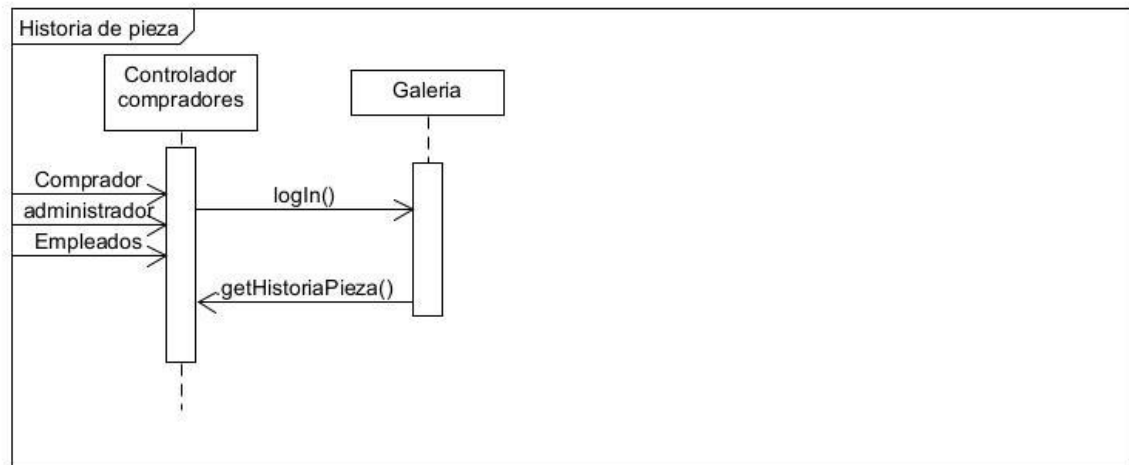
Entradas:

Para poder entrar al sistema, se deben ingresar las credenciales una vez hecho esto, el comprador podrá hacer una oferta a una pieza o a una subasta el cual quedará en cola para su revisión, el momento de que le validen la oferta se le entregará la respectiva pieza a su inventario, para consultar la historia de una pieza y de un artista, simplemente tendrá que logearse y podrá consultar la historia.

Salidas:

Al momento de hacer una oferta se retornará la oferta para su revisión, en las consultas el sistema le arrojara una cadena con las historias.

Modificación del uml :



Interfaz:

Hay tres interfaces implementadas para cada tipo de usuario, están implementadas dentro del paquete de modelo, y cada una prueba el funcionamiento de la aplicación para cada caso de usuario, eje: un administrador tendrá que indicar su usuario y contraseña para acceder a sus funciones el cual son revisar ofertas o consultar historias, no se tendrá que introducir el nombre del archivo y para guardar el usuario tendrá que seleccionar la opción guardar en el primer conjunto de opciones.

Nivel 4

Nuevos requerimientos:

- Interfaz grafica
- Pago Tarjeta crédito

En este nuevo nivel hay dos requerimientos nuevos principales, hacer que los pagos en el sistema también se puedan hacer (modelado) con tarjetas de crédito añadiendo funcionalidad a los pagos de las piezas de la galería y por su otra parte, hacer una interfaz gráfica para el usuario en vez de manejarla por consola.

Pagos Tarjeta de crédito:

Para este requerimiento implementamos un nuevo paquete que contiene la lógica y el código, el funcionamiento es simple una vez el comprador este en el sistema podrá cambiar su método de pago, el cual se utilizara hasta que decida cambiarlo, se incluyó una nueva opción en la interfaz que es “método de pago” en cual esta opción, establece si va a pagar en efectivo o si va a utilizar la tarjeta, la información de las tarjetas se almacenan dentro del archivo datos, el cual contiene los datos necesarios y el saldo, en el momento en que el cajero revise la oferta y el comprador va a pagar con tarjeta tendrá que seleccionar una de las pasarelas disponibles y ingresar los datos del comprador, el sistema descontara el dinero de la tarjeta y registrara la transacción en el archivo logTransaccion.txt, si la tarjeta no contiene el saldo suficiente para la oferta, simplemente esta se cancelara, la configuración de las pasarelas está contenida en el archivo metodos.config el cual se pueden cambiar las pasarelas disponibles.

Modificacion uml:

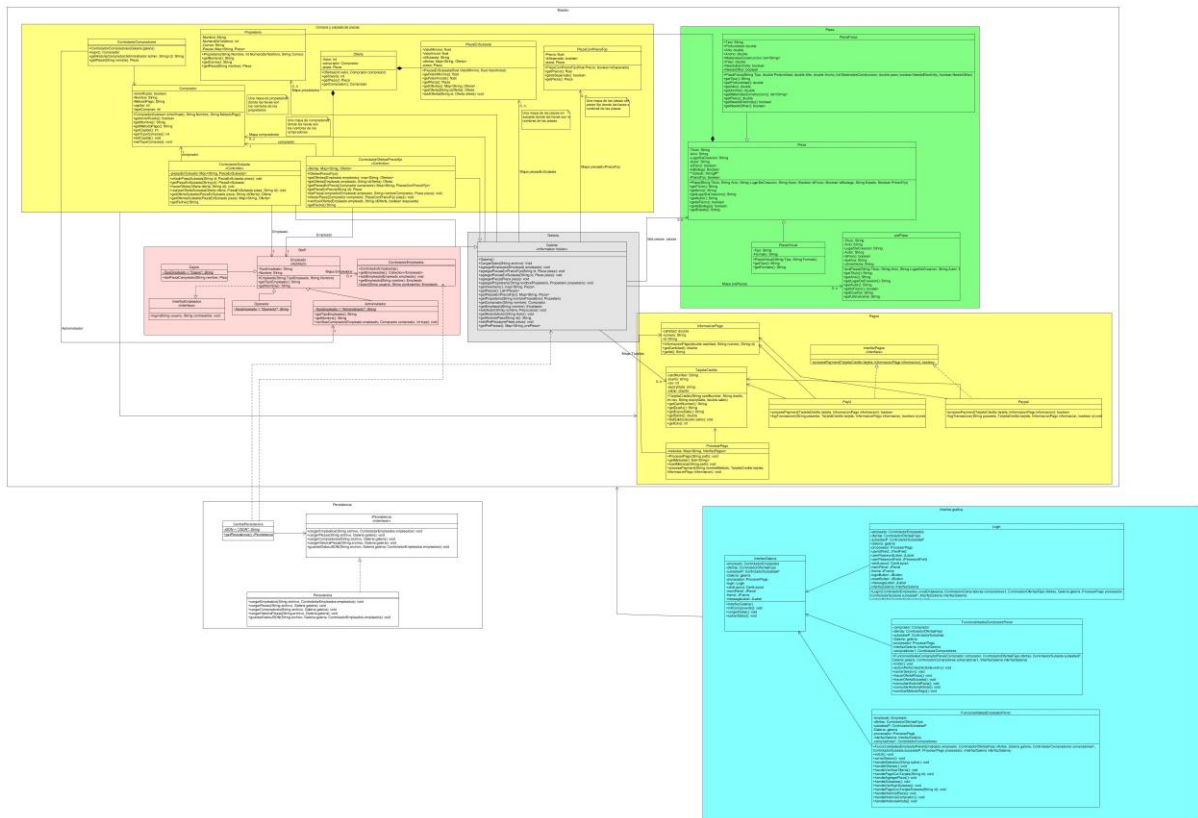
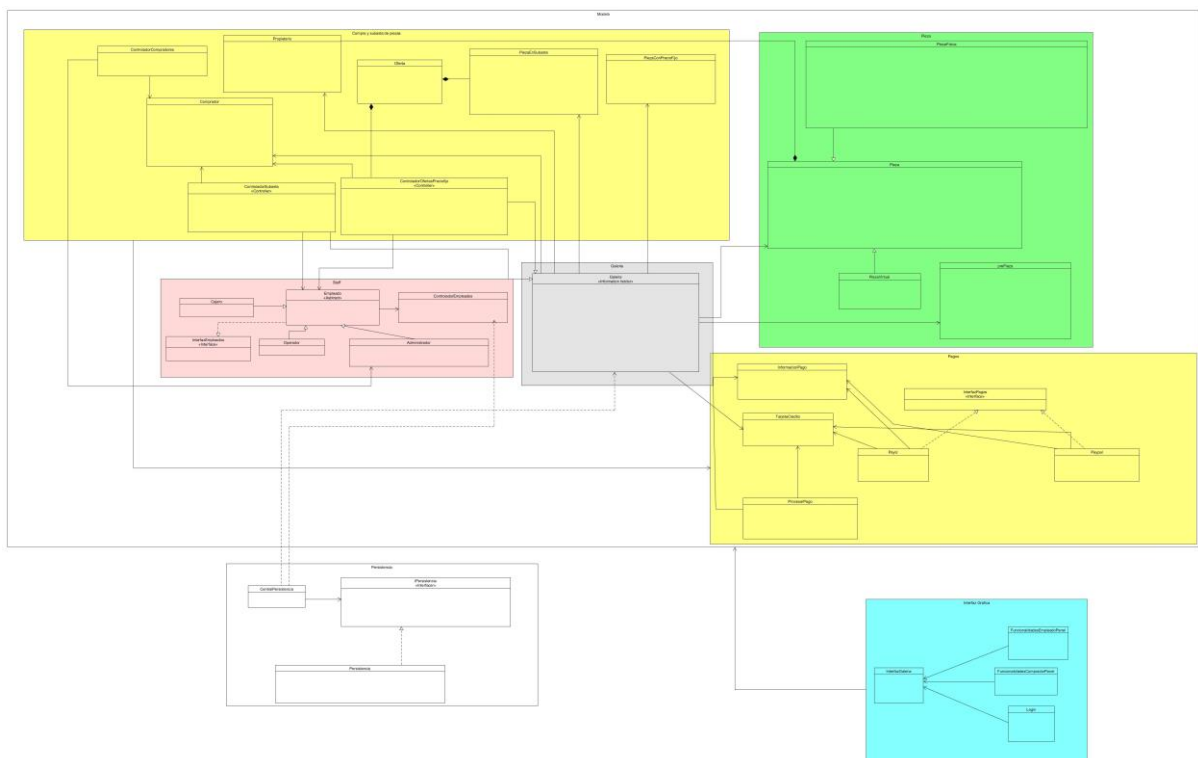


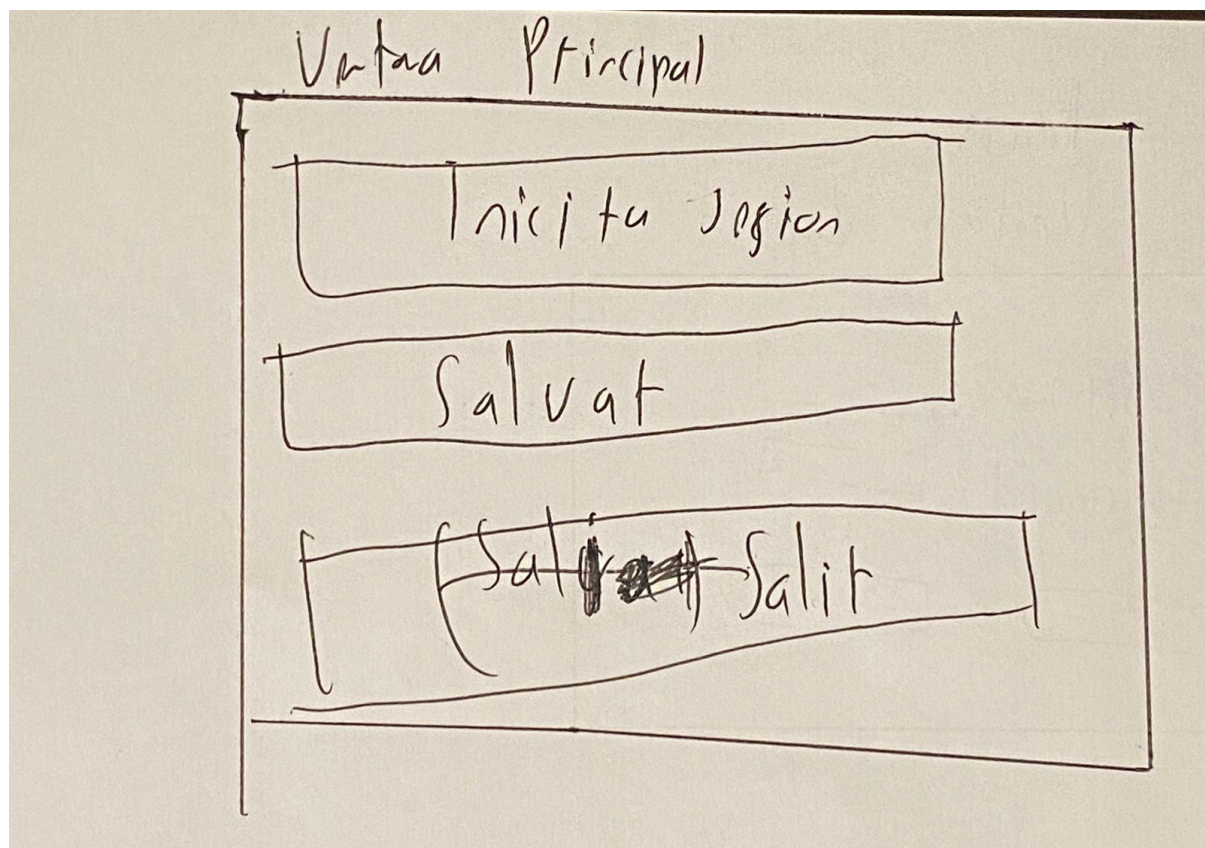
Diagrama de alto nivel



Interfaz Gráfica (GUI):

Para hacer la interfaz usando java swing se necesita crear una parte principal de la pantalla donde se hará display de la información, para ello se hicieron los bosquejos de cómo sería una manera de mostrar la información de la aplicación con los lugares de inputs correspondientes para el funcionamiento completo.

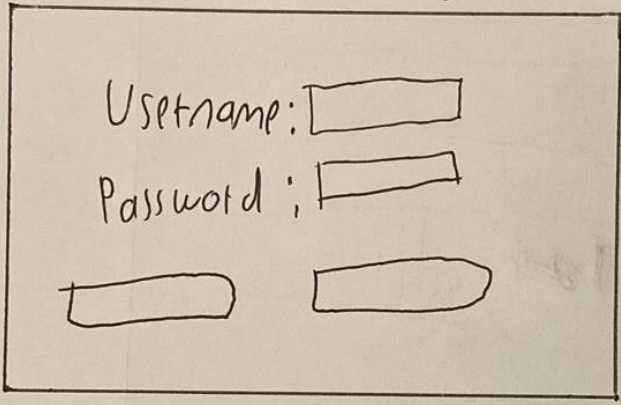
Ventana Principal:



Al momento de iniciar la aplicación se mostrará algo como la figura anterior en donde solo se encuentran las opciones principales del menú "Iniciar sesión", "salvar" (información), "salir", esta ventana es bastante sencilla.

Login de la aplicación:

Ventana Login



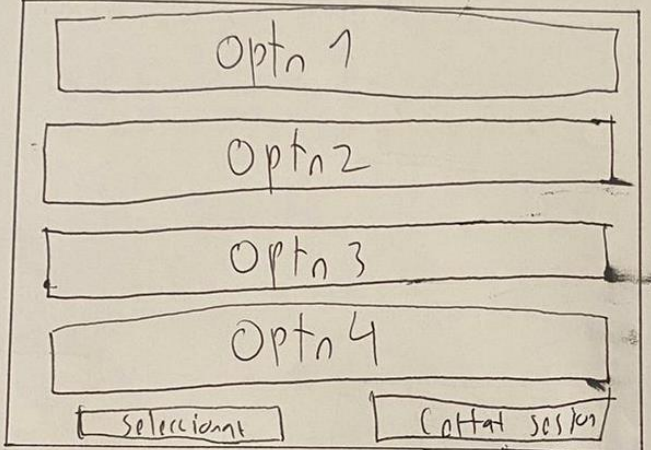
Username:

Password:

Al momento de hacer login para hacer uso de todas las funcionalidades del sistema se deberá haber seleccionado la opción indicada en la ventana principal y como se ve en la figura anterior aparecerá una ventana donde hay un lugar donde poner el username y la contraseña indicados en la ventana y para hacer la acción de loggear solo se debe oprimir uno de dos botones que estarán abajo (el otro será para salir/devolverse) .

Ventana Usuario:

Ventana Empleado, admin, operador, Comptodo, propietario.



Optn 1

Optn 2

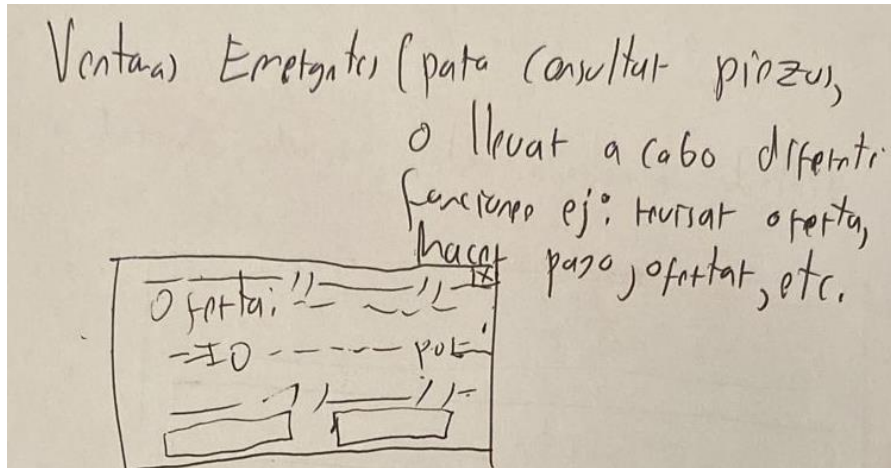
Optn 3

Optn 4

seleccionar Cerrar sesion

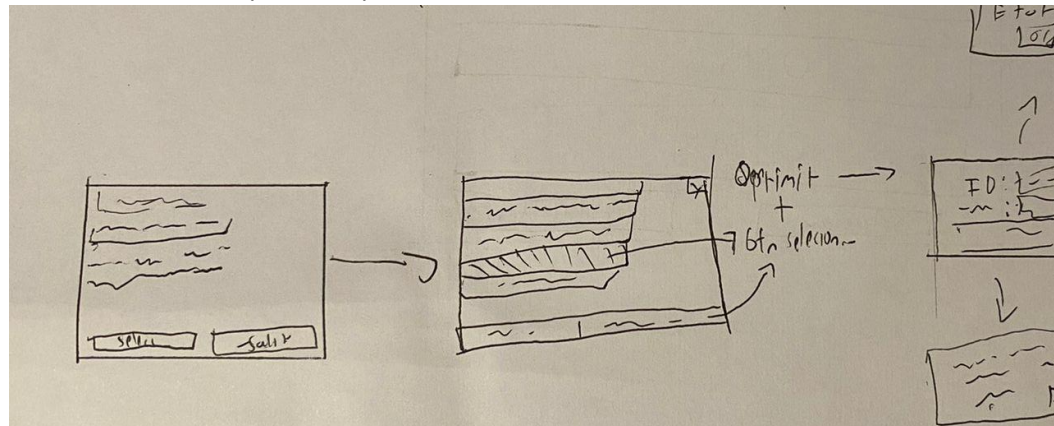
dependiente del tipo de user.

Ventanas Emergentes de la aplicación:



En este bosquejo se construye el esqueleto de las ventanas emergentes que planeamos implementar para estos decidimos hacerlo de la forma más breve posible solo mostrando el mensaje que indica en que parte del proceso se encuentra y botones para aceptar o seguir adelante con algún proceso.

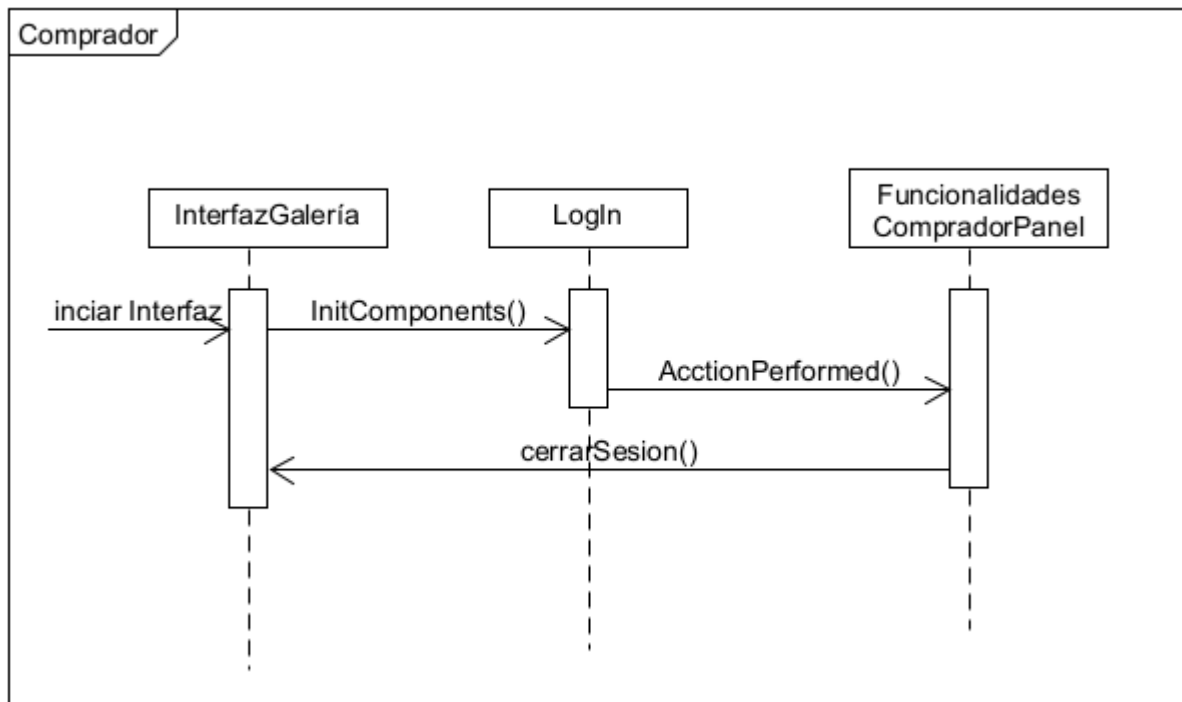
Proceso de uso de la aplicación por la interfaz:



En la figura anterior es posible ver como después de haber iniciado sesión correspondiente es posible empezar a seleccionar diferentes opciones ya dependiendo del usuario ya sea administrador, cajero, empleado, etc. Para hacer esto se necesita seleccionar una opción y después pulsar el botón para seguir adelante mediante diferentes ventanas emergentes que saldrán para seguir con el proceso y al terminar solo será necesario cerrar estas.

Diagramas de secuencias para :

Compradores:



Para empleados:

