



“Rincón Floral”

- florería -

Nombre de la materia: Laboratorio III

Alumno/a: Mailen Cruz

Fecha de entrega: 30/06/2023

Resumen de la aplicación desarrollada y el proceso de desarrollo.

La aplicación desarrollada está pensada para la compra del cliente y para una configuración del sistema muy sencilla. En ella existe un menú principal el cual permite al usuario dar inicio de sesión en la aplicación o acceder a la configuración del sistema.

La opción que da lugar al inicio de sesión permite que el usuario ingrese o se registre, una vez dentro de la aplicación inicia su compra. En la misma comenzará agregando flores a su carrito y una vez que se encuentre conforme con su compra pasará a la elección del método de envío.

El método de envío le permite al usuario retirar su compra por el local o recibirla a domicilio, según la opción que ingrese son los datos que se le van a solicitar o a mostrar.

Una vez elegido el método de envío, se le muestra al usuario una serie de opciones para realizar el pago, entre ellas se encuentra el efectivo, la tarjeta o la cuenta DNI. Según la opción seleccionada son los datos a solicitar.

Cuando finaliza la compra se le imprime por pantalla al usuario un ticket con pantalla para que el mismo posea un registro de la compra. Este contiene las flores compradas, con el método de pago elegido y el valor total abonado.

Por otro lado, la opción configuración del sistema requiere de una contraseña para el acceso. En ella encontramos opciones para agregar, eliminar o modificar flores que se encuentren presentes en el catálogo vinculado a la aplicación, o para agregar, eliminar o modificar usuarios.

El proceso de desarrollo comenzó con la creación del catálogo en un archivo json, en el cual se investigó sobre las flores existentes en el mundo y se fueron cargando una por una. Luego se dió comienzo a la carga de un archivo.bin el cual contiene una base de usuarios ya existentes en la app.

Seguido de esto se comenzó la codificación de la parte que se encargaba de generar el alta, baja y modificación de los archivos previamente mencionados.

Una vez finalizado, se creó el boceto de clases e interacción entre las mismas para desarrollar todo lo relacionado a la compra de las flores. Las excepciones posibles a efectuar y las interfaces a implementar fueron pensadas en esta instancia del proceso.

Antes de comenzar con la codificación se hizo un listado de los temas obligatorios a incluir y se fue analizando si todos formaban parte del programa.

Cuando el boceto ya estaba ordenado y completo, se realizó el código correspondiente.



Informe Técnico de la aplicación, donde se deben describir el funcionamiento del sistema, las decisiones tomadas de diseño, las clases utilizadas y sus relaciones.

Como principal idea se decidió llevar a cabo las funciones del sistema dentro de una clase Menú, el cual contiene los diferentes menús que se van a ir ejecutando a medida que el usuario interactúe.

Configuración del sistema:

En él encontramos 2 archivos, uno .bin y otro .json.

El archivo.bin contiene información de usuarios cargados en el sistema, por lo que lo primero que se llevó a cabo fue la creación de la clase Usuario la cual contiene el nombre, apellido, mail y contraseña del mismo.

Seguido de esto se creó la clase ArchivoUtiles la cual consta de las funciones necesarias para leer y guardar datos de tipo Usuario en ese archivo.

Luego se creó una clase llamada ControladorUsuario la cual posee el desarrollo de las funciones de muestra, alta, baja y modificación de los mismos. Se tomó como elección el uso de un HashSet para guardar estos datos de tipo Usuario, debido a que esta colección no permite objetos repetidos.

El archivo.json contiene la información de un catálogo de flores, para el cual se realizó una investigación de las flores existentes en el mundo. Este se creó como un arreglo en el cual en cada posición había un objeto de tipo Flor el cual contenía los datos de la misma.

Para poder trabajar con este archivo en Java primero se implementó el uso de la librería JSON y una clase llamada JsonUtiles la cual contiene las funciones necesarias para poder pasar los datos del JSON al lenguaje de Java.

Se creó la clase llamada Flor para poder cargar cada elemento del arreglo la cual contiene los atributos nombre, código, color, origen y precio por unidad.

Luego se tomó la decisión de poner las flores en un Mapa específicamente un HashMap para así poder tener acceso a ellas mediante una clave, que en este caso podía ser tanto el nombre, el código, el color o el origen.

Por último se creó la clase ControladorFlores la cual posee el desarrollo de las funciones de muestra, alta, baja y modificación del mismo.



Ingreso a la APP:

El ingreso a la app lo primero que tiene en cuenta es el inicio de sesión del usuario o el registro del mismo. Para ello se realizan una serie de comprobaciones, entre ellas que el mail ya se encuentre en uso, el lanzamiento de una excepción que se genera cuando se ingresa una contraseña menor a 8 dígitos, entre otras.

Una vez que ingresa la primera clase que entra en funcionamiento es la del Carrito, que posee como atributos un arreglo de Elecciones y un total. La clase Elecciones es creada con el fin de que el usuario además de seleccionar la flores que quiere pueda poner la cantidad deseada de la misma, por ello la clase Elecciones contiene como atributo una flor de tipo Flor y una cantidad.

Carrito posee los métodos “agregaralcarrito”, el cual agrega la flor elegida por el usuario junto con su cantidad en el arreglo, “eliminar del carrito” la cual elimina por cantidad deseada las flores que se encuentren en el arreglo,” calcular total” la cual calcula el total de la compra y “mostrar carrito” que nos devuelve una visión de la flor seleccionada junto con la cantidad elegida de la misma.

En el menú encontramos las funciones “cargaCarrito” en el cual se le pide al usuario que ingrese el código de la flor y la cantidad deseada de la misma y “modificarCarrito” que se le da una serie de opciones al usuario entre ellas agregar o eliminar flor que se encargará de llevarlas a cabo. Estos métodos son usados cuando el usuario selecciona la opción correspondiente al mismo.

Por otro lado se crea la clase abstracta MetodoDeEntrega, en ella se encuentra el método “calcularTotalDelEnvio” que es implementado en las dos clases que heredan de él.

Por un lado tenemos la clase Local la cual tiene como atributo una calle, altura, días que se encuentra abierto y el horario del mismo, la implementación del método la lleva a cabo aplicando un 10% de descuento sobre el total de la compra por realizar el retiro del pedido en el local.

Luego está la clase Domicilio que tiene como atributos calle, altura y código postal e implementa el método generando un aumento en el precio total correspondiente al precio del envío. El precio del envío se calcula según el código postal ingresado.

Luego en el menú según la opción ingresada se solicitan los datos del domicilio o se muestra la info del local, ambos con los precios totales actualizados. Finalizada la carga o muestra de datos se corrobora que el usuario esté conforme con la elección de entrega y de ser así se lo envía al menú de Método de pago.

Para la elección del método de pago se crearon 3 clases: Efectivo, Tarjeta y CuentaDNI, éstas implementan una interfaz llamada I_ProcesarPago la cual se encarga justamente de procesar el pago o cancelarlo.

La clase Efectivo no contiene atributo pero si las implementaciones de los métodos, “procesarPago” mostrará un mensaje indicando que el pago de la compra se hará en el local sino no se entregará el producto y “cancelarPago” el cual muestra un mensaje de que el

pago ha sido cancelado y a nivel de código lo que se genera es un reseteo del programa borrando todos los datos previamente cargados.

La clase Tarjeta contiene los atributos nombre del titular, apellido del titular, número de la tarjeta, código de seguridad y el tipo (débito o crédito). La implementación de los métodos la realiza mostrando por pantalla una vez acreditado el pago el valor total con el número de tarjeta correspondiente y la cancelación del pago la implementa de la misma forma que Efectivo.

En el menú podemos encontrar el menú de método de pago el cual usa los métodos “cargaTarjeta” y “elecciónDeCuotas”.

El método “cargaTarjeta” se encarga de solicitarle al usuario la carga de los datos de la misma y usa dos métodos que lanzan dos excepciones, la primera es lanzada cuando los números de la tarjeta son distintos a una cantidad de 15 o 16 y la segunda es lanzada cuando el código de seguridad no es de 3 dígitos.

El método “elecciónDeCuotas” le da a elegir al usuario si quiere pagar en 3, 6 o 12 cuotas, los intereses de las mismas son calculados en otro método. Una vez que el usuario elige se le muestra el valor de cada cuota y el total de la compra, se le pregunta si está conforme con la elección y de no ser así se lo hace volver a elegir las cuotas.

La clase cuentaDNI tiene como atributos un dni, una clave, y el valor a pagar, implementa el método de procesar pago mostrando el total de la compra y el dni asociado a la misma y la cancelación del mismo lo implementa de la misma forma que la clase Efectivo y Tarjeta.

Cuando el usuario selecciona esta opción el sistema trata de imitar una compra por cuenta dni, primero le pide que ingrese el dni, luego que ingrese el monto de la compra en la parte de Generar Clave en la app de cuenta DNI y que genere la misma y la ingrese. Una vez hecho todo esto si el usuario lo desea se procesa el pago.

Cuando el usuario ingresa la clave, se lanza una excepción la cual se genera cuando la clave no posee 6 dígitos y le solicita al usuario que la vuelva a generar e ingresar.

Una vez que el usuario confirma el pago se genera un ticket el cual es creado con una clase genérica Ticket la cual posee como atributo 3 datos para mostrar en él, se pensó que en un futuro la empresa podría querer mostrar otros datos en el ticket, por eso se hizo genérica.

La implementación de la misma en el programa se hace mostrando los productos seleccionados, la forma en la que se pagó y el total de compra abonado.



Matriz de soluciones y diario de trabajo.

MATRIZ DE SOLUCIONES	
PROBLEMAS	SOLUCION
mal funcionamiento del bucle de modificacion	agregue un bucle while y en lugar de usar dos funciones use 1
funciona las funciones de ABM pero no me guarda los cambios en el json	cree una variable en el Menu que reciba el mapa con los datos json
cuando el usuario agrega o elimina flores de su carrito, modifica mal el precio	se agrego una linea de codigo la cual sumaba o restaba la cantidad de flores que queria agregar o eliminar con el precio actual
lanzaba excepcion cuando el usuario queria volver a cargar el numero de la tarjeta si lo ingresaba mal	estaba lanzando la excepcion en el metodo y en la declaracion del mismo por lo que al momento de llamarlo de nuevo me volvia a lanzar la excepcion, opte por retirar el lanzamiento de la excepcion de la declaracion del metodo
Desea cancelar el pago? pongo que no y me muestra el menu principal	inicializacion en "s" a la variable que hacia que el bucle se mantenga en funcionamiento

DIARIO DE TRABAJO	
FECHA	TAREA REALIZADA
1/6/2023	Planteo de idea del Tp - Ideas de como iba a funcionar el mismo
5/6/2023	Investigacion de flores - creacion de un listado de las mismas - investigacion del origen de cada una
9/6/2023	Creacion del archivo json - investigacion del formato en el que devian estar los datos - planeacion de resultado que queria que tenga el archivo en Java
13/6/2023	Pasaje de datos json a Java - generacion del archivo de usuarios
14/6/2023	Comienzo del codigo utilizado para la configuracion del sistema
19/6/2023	Planteo de diagrama de clases utilizadas para hacer la compra del usuarios
20/6/2023	Comienzo del codigo utilizado para la compra del usuario
23/6/2023	codigo enfocado en el carrito y la modificacion del mismo
26/6/2023	codigo enfocado en el metodo del pago y el metodo de entrega
28/6/2023	creacion del ticket para finalizar con la compra del usuario