

PROGRAMACIÓN Y LABORATORIO 3°

JavaCafé

DOCENTE : Benoffi, Gonzalo

ALUMNOS : Arias, Valentin
Falotiche, Federico
López, Agustin

2023

Resumen de la aplicación desarrollada

JavaCafé se utilizará en gastronomías para poder llevar un control sobre sus ventas, clientes y productos.

Posee un amplio menú donde se podrá ver, eliminar y editar dicha documentación.

Cuenta con dos menú de control tanto para empleados como administradores dando así diferentes funcionalidades para que sus datos sean privados al momento de quien los visualiza.

Proceso de desarrollo

La aplicación fue creada a través del **IDE IntelliJ** a base del lenguaje **Java**, se utilizó el software de controlador de versiones **Git Bash** para mantener un mismo código manejado por los participantes del grupo.

Se implementaron diferentes ideas para llegar al funcionamiento de esta aplicación utilizando una herramienta de diagramación **UML** para la gestión y diseño de clases, atributos y métodos.

La forma de implementación del código fue mayormente online a través de la plataforma **Discord**, y se realizaron reuniones para detallar algunos problemas que no se pudieron resolver virtualmente.

Se utilizó la herramienta de gestión de proyectos **TRELLO** para ir detallando las tareas realizadas, y por realizar.

Al finalizar cada día de desarrollo, se actualizaba la lista de tareas en este mismo, la matriz de errores y la planilla de tareas para una mayor organización.

Manual de usuario

En JavaCafé implementamos un menú amigable para cualquier tipo de usuario, se compone de dos entradas tanto para empleados de bajo

rango (quienes no necesitan obtener toda la información del negocio) y administradores (quienes controlan toda la información necesaria para la gestión del negocio).

```
BIENVENIDO.  
  
INGRESE FORMA DE OPERAR:  
  
1 - Empleado.  
2 - Administrador.  
  
0 - SALIR.  
  
Opcion:
```

Menu Empleado:

Dentro del menú empleado tendremos una gestión básica de información para la venta de productos, agregar clientes al sistema y mostrar todos los productos que hay en venta.

```
---OPCIONES EMPLEADO---  
  
1 - VENDER  
2 - AGREGAR CLIENTE  
3 - MOSTRAR PRODUCTOS EN SISTEMA  
  
0 - ATRAS  
  
Opcion:
```

- 1. Venta:** Al momento de ingresar una venta, el empleado tendrá a su disposición ingresar el DNI de un cliente existente, o una venta de sistema ingresando el DNI 0.

Si el DNI ingresado no existe, el programa le preguntará al usuario si desea agregar esa información al sistema, si acepta, cargará los datos del nuevo cliente, caso contrario, se creará la venta anónima con DNI 0.

Luego de la documentación del cliente, se empezará a cargar los productos que este pidió al mozo para armar su comanda y el lugar donde consumirá.

Al finalizar la comanda, automáticamente se le entregara el precio total de lo consumido.

- 2. Agregar Cliente:** Cuenta con la opción de poder cargar en sistema la información de un nuevo cliente si este mismo es alguien que concurre mucho al lugar.

El sistema tiene una pequeña opción de descuento si el sujeto agregado es VIP. Dicha opción se marcará al momento de agregarlo al sistema, y el descuento se hará al realizar la venta.

- 3. Mostrar Productos en Sistema:** Muestra todos los productos que están cargados en sistema.

Se dividen en sus tipos de comidas y bebidas, o pueden verse todos los productos al mismo tiempo.

```
PRODUCTOS EN SISTEMA:

1 - COMIDAS DULCES.
2 - COMIDAS SALADAS.
3 - BEBIDAS FRIAS.
4 - BEBIDAS CALIENTES.
5 - MOSTRAR TODOS LOS PRODUCTOS.

0 - ATRAS.

Opcion:
```

Menu Administrador:

Al momento de ingresar al menú de administración, se solicita un usuario y contraseña que, por defecto, iniciará siendo “admin” para ambos casos.

```
Usuario: admin
Contraseña: admin
```

una vez dentro del menú, el usuario podrá hacer cambios de esta clave ingresando en la opción 16 del menú.

```
---OPCIONES PARA CLIENTES---
1 - AGREGAR CLIENTE
2 - ELIMINAR CLIENTE
3 - MODIFICAR CLIENTE
4 - MOSTRAR TODOS LOS CLIENTES EN SISTEMA
5 - BUSCAR CLIENTE

---OPCIONES PARA VENTAS---

6 - VENDER
7 - ELIMINAR VENTA
8 - MODIFICAR VENTA
9 - MOSTRAR VENTAS DEL SISTEMA
10 - BUSCAR VENTA

---OPCIONES PARA PRODUCTOS---

11 - AGREGAR PRODUCTO
12 - ELIMINAR PRODUCTO
13 - MODIFICAR PRODUCTO
14 - BUSCAR PRODUCTO
15 - MOSTRAR TODOS LOS PRODUCTOS DEL SISTEMA

-----

16 - CAMBIAR CLAVE DE INGRESO.

0 - ATRAS

Opcion:
```

Opciones para clientes:

- 1. Agregar Cliente:** Funciona de igual manera que la opción 2 del menú empleado.
- 2. Eliminar Cliente:** Busca a un cliente por su número DNI y lo elimina del sistema, si el cliente no existe o se intenta eliminar al cliente con DNI 0, saltará un mensaje de error en pantalla.

- 3. Modificar Cliente:** Se puede modificar la membresía VIP de un cliente existente. No se puede modificar el cliente DNI 0.
- 4. Mostrar Todos Los Clientes En Pantalla:** Se mostrará en forma de lista todos los clientes que están guardados en el sistema de gestión.
- 5. Buscar Cliente:** Busca a un determinado cliente a través de su DNI. Si el cliente no existe, da opción a cargarlo en el sistema.

Opciones para ventas:

- 6. Vender:** Funciona de igual manera que la opción 1 del menú empleado.
- 7. Eliminar Venta:** Se listarán todas las ventas que se generaron en el sistema hasta el día, cada venta posee un **número de ticket** por el cual se gestionará su eliminación.

Si la venta no existe, se mostrará un mensaje de error en pantalla.
- 8. Modificar Venta:** Se puede modificar el precio total de una venta en específico buscándola por su número de ticket.
- 9. Mostrar Ventas Del Sistema:** Muestra todas las ventas existentes en una lista.
- 10. Buscar Venta:** Se busca una venta por número de ticket y si esta existe, se mostrará todos los detalles de esa venta en pantalla.

Opciones para productos:

- 11. Agregar Producto:** Agrega un nuevo producto para la venta al sistema.

Se consulta qué tipo de producto se desea agregar.

```
Ingrese que producto desea agregar:
```

```
1 - Comida.
```

```
2 - Bebida.
```

```
0 - Atras.
```

```
Opcion:
```

Si el producto es comida, pregunta cuál es su categoría.

```
Que tipo de comida desea agregar:
```

```
1 - Comida dulce.
```

```
2 - Comida salada.
```

```
0 - Atras.
```

```
Opcion:
```

Al igual que los productos de bebidas.

```
Que tipo de comida desea agregar:
```

```
1 - Bebida fria.
```

```
2 - Bebida caliente.
```

```
0 - Atras.
```

```
Opcion:
```

luego se le añade el nombre del producto, una descripción del mismo y su precio.

12. Eliminar Producto: Elimina del sistema un producto en específico, se mostrará la lista de productos y se solicitará que ingrese el ID del producto a eliminar.

Si este no existe, mostrará un error en pantalla.

13. Modificar Producto: Modificara el precio de un producto ingresando su ID.

14. Buscar Producto: Se busca por medio de su ID y se muestra la información completa del mismo.

15. Mostrar Todos Los Productos Del Sistema: Funciona igual que la opción 3 del menú empleado.

En cada SubMenú existe la opción de volver atrás presionando 0.

Matriz de soluciones y diario de trabajo

MATRIZ DE ERRORES	
ERROR	SOLUCION
No se supo donde settear el descuento si un cliente tenia acceso VIP.	Se consulto al docente, la condicion iba en el setter del atributo VIP del cliente y este es llamado en su propio constructor.
Problema de implementacion de clase Enum.	Se encontro la forma de aplicar dicho Enum en la clase correspondiente.
Creamos clases equivocadas de comidas especificas.	Se utilizaron Enum que reemplazarian a estas clases, reduciendo la jerarquia de las mismas.
Los servidores de las APIs implementadas dejaron de responder, nos quedamos sin productos de bebidas frias y calientes.	Se crearon dos archivos JSON con dichos productos, para esto la clase envoltoriaProducto tiene sus dos metodos de consumo JSON archivo local.
Problema de la interfaz IABM en los parametros de las funciones.	Se ajustaron los parametros de las funciones para las tres clases que la implementan.
Al momento de eliminar y modificar un dato, sigue persistiendo en su respectiva lista por su archivo local.	Se implemento en los metodos eliminar y modificar, que re-escriba el archivo con la lista actualizada.
Error de interpretación en cuales colecciones teniamos que usar en el proyecto. Falta implementar Map.	Se cambio la coleccion de clientes de un LinkedHashSet por un HashMap.
Persistencia de JSON se duplicaba al momento de agregar un objeto.	Se cambio de lugar el metodo consumoJSON y se creo un metodo de vaciarListaProductos
Se listaban los JSON comidas dulces y saladas, ambas listas mostraban lo mismo	Habia un grabado repetida de Comidas Saladas en el metodo agregarComidaAJson
Error de persistencia de datos al momento de crear los metodos de crear archivos Ventas y Clientes	No se habia implementado la interfaz Serializable, se aplico y funciona.
Errores de printeos y strings nulos en clase menu	Se rastreo y modifiko las variables agregando condiciones.

Aplicaciones utilizadas para el desarrollo del sistema.

REPOSITORIO GIT:

[AriasValentin/AppGastronomica: Trabajo final de programación en java. \(github.com\)](#)

UML:

[herramienta de diagramación UML](#)

TRELLO:

[TRABAJO FINAL PROG Y LAB 3 | Trello](#)

DIARIO DE TRABAJO:

[Lista de tareas - Hojas de cálculo de Google](#)

Fuentes de información consultadas

GIT TUTORIAL:

<https://www.youtube.com/watch?v=VdGzPZ31ts8>

COMANDOS GIT BASICOS:

https://apuntes-snicoper.readthedocs.io/es/latest/git/comandos_basicos.html