

Programación II

Proyecto programado- I cuatrimestre de 2021

Profesoras: Laura Agüero Castro

Ana Cecilia Odio Ugalde

Porcentaje: 20%

Fecha de entrega y presentación:

1. **Diagrama de clases:** Semana del 15 al 20 de marzo durante el segundo día de clase (imagen).
2. **Proyecto completo:** Semana del 19 al 24 de abril durante el segundo día de clase.

I- Descripción del problema.

Una nueva plataforma de servicio de comidas, tipo Uber-Eats, lo ha contratado para diseñar su sistema de compra y cobro de servicios de entrega de productos.

1. El sistema abarcará la información de todos los servicios que ofrece.
2. Para hacer uso de la plataforma, es necesario registrar una cuenta de usuario de esta.
3. Para ingresar al sistema los usuarios se deben registrar y abrir una cuenta. Un usuario se debe registrar por medio de su correo electrónico y clave. Al mismo tiempo, debe registrar una tarjeta de crédito que puede ser Visa o Master Card. Existe usuario administrador, usuario regular y usuario gerente de restaurante o lo que es lo mismo usuario empresa.
4. Los usuarios regulares son las personas que compran alimentos y los gerentes son los encargados de los restaurantes (restaurantes que ofrecen productos) que se encargan de introducir en la plataforma, los productos que oferta dicho local por ejemplo los gerentes de los restaurantes.
5. De acuerdo con el tipo de usuario, se podrá acceder a diferentes opciones de la aplicación. El usuario regular no podrá acceder a las opciones de empresa ni a los reportes, solamente tendrá acceso a la sección de ventas. El usuario gerente no podrá acceder a las ventas, el administrador podrá ingresar a todo el sistema.
6. Cada restaurante o local se debe registrar en el sistema indicando su nombre, localización y tipo (Italiano, Hamburguesas, Chino, Repostería, entre otros). Utilice al menos 4 tipos de acuerdo a su elección.
7. Cada producto posee: un código que lo identifica, una descripción, precio y a qué negocio pertenece (es decir, por ejemplo la pizza margarita tiene un código diferente en cada restaurante que la ofrece).
8. Existen dos tipos de productos:
 - a. Preparados: para estos productos se debe registrar el tiempo mínimo de preparación (en minutos).
 - b. Empacados: Ejemplo: gaseosas, agua, tortillas, pan, entre otros. De estos se registra la cantidad que hay en el inventario del local y un porcentaje de impuesto ya que pueden ser diferentes de acuerdo al producto (exento 0%, 1%, 2%, entre otros).

9. Los usuarios regulares efectúan compras a través de la plataforma. Cada compra se auto numera consecutivamente a partir de 1 y puede contener uno o varios productos. Para los productos preparados se debe sumar el tiempo de preparación para indicar el tiempo de entrega; para los empacados es solo validar que exista la cantidad necesaria de productos en el archivo de productos en el restaurante correspondiente.

Los productos se cobran de la siguiente forma:

- Preparados: se debe cobrar el 13% de IVA más el 10% del servicio de transporte al precio del producto.
- Empacados: se debe cobrar según la lista de impuestos de tal manera que existirán algunos exentos, otros que deben pagar 1% o 2% de impuesto y así sucesivamente.

II – Requerimientos Técnicos

1. El programa debe estar estructurado en 3 capas:
 - **Capa Interfaz (capa gráfica, capa vista):** Diseño de ventanas a través de las cuales se establecerá la comunicación con el usuario
 - **Capa de Lógica de Restaurantes:** Diseño de las clases, su funcionalidad y sus relaciones.
 - **Capa de Acceso a datos:** Diseño de la(s) clases que dan persistencia a los objetos. Para efectos de este curso se utilizará una estructura de archivo para guardar los objetos Productos que posee la plataforma. También se utilizará un archivador en ArrayList que almacenará los objetos Usuario.
2. Cuando el usuario ingresa a la aplicación se le da la opción de registrarse o de ingresar si ya existe como usuario.
 - Para ingresar al sistema se solicita el usuario(correo) y clave, se consulta en el Archivador de Usuarios que dicho usuario existe y con esa clave. Se dan tres oportunidades y si no acierta se cierra la aplicación.
 - Si el usuario es nuevo y desea registrarse, se le solicita su información: correo electrónico, nombre, país de origen, clave (no se debe ver cuando la digita), tipo de usuario (regular, gerente) tipo de tarjeta (Visa o Master Card), número de tarjeta, fecha de vencimiento (mes y año).
 - De la tarjeta se debe revisar que no esté vencida de acuerdo con la fecha del sistema. Además, se debe validar que el número de tarjeta sea válido según los lineamientos que se describen más adelante.
 - El archivador de usuarios ya debe tener algunos usuarios registrados de los tres tipos que ingresan directamente a la aplicación, los que no se encuentren en el archivador se deberán registrar.
 - Si se ingresa con usuario **admin** y clave **123**, se identifica como usuario administrador y tendrá acceso a algunas funcionalidades que no tiene el usuario regular ni el gerente de empresa. Este usuario debe estar desde un inicio en el archivador con toda la información registrada previamente.
3. Si el usuario pudo acceder a la aplicación se le presentan sus perfiles registrados (tipos de usuario) y debe seleccionar el que corresponda que va a ingresar a la plataforma.
4. La aplicación debe presentar una ventana principal en la que el usuario por medio de un menú pueda acceder a las diferentes funcionalidades de la aplicación según su perfil (tipo de usuario) :Buscar productos, Realizar compras, Calificar productos, entre otros. Para los usuarios gerentes tendrán las

opciones de crear restaurantes y registrar productos para la venta, estos pueden estar creados con anticipación o ingresados en ese momento. Si es administrador además podrá ingresar todas las opciones de la aplicación.

5. Durante toda la aplicación en la barra de estado debe aparecer el usuario y perfil (tipo de usuario) conectado.
6. Ventana de Mantenimiento de Restaurantes (restaurantes que ofrecen productos). Se efectuará en un archivador que contenga un ArrayList de restaurantes.

Se debe diseñar una ventana en la que el usuario pueda ingresar locales, así como a los productos ofertados. La lista de productos se almacenará en un **Archivo** que se pueda acceder desde cualquier ventana de la aplicación. En esta ventana de mantenimiento de restaurantes, se debe permitir al usuario:

- a. Permite agregar toda la información del nuevo local (nombre, dirección y tipo). Se debe validar que el local no exista ya en el archivador. No se incluye las opciones de modificar ni eliminar.
7. Ventana de Mantenimiento de Productos. Se efectuará en un **archivo en disco**
 - a. La opción de agregar productos se puede manejar en la misma ventana del mantenimiento del negocio o diseñar una adicional para incorporar los productos que ofrece el local (el diseño será el elegido por usted).
 - b. También debe contener las operaciones de modificar, eliminar y consultar toda la información del producto.
 8. Ventana Orden de productos (Carrito de compras)
 - a. El usuario selecciona el tipo de negocio en el que desea comprar (Pizza, Repostería, etc.).
 - b. Dependiendo de su selección se le mostraran todos los restaurantes del tipo elegido junto a su calificación promedio de acuerdo a las calificaciones de los clientes según se indica más adelante.
 - c. El usuario selecciona el restaurante elegido.
 - d. Dependiendo de su selección se le mostraran todos los productos que se ofrecen en ese local (se busca en el archivo en disco el producto según el restaurante que lo ofrece)
 - e. Una vez seleccionado el restaurante, se puede buscar por nombre del producto.
 - f. Si el usuario busca por nombre de producto se deberá ir filtrando los productos de ese local según se va digitando. Ejemplo si el usuario digita pizza, deben aparecer todas las pizzas de dicho restaurante, pero si pone pizza de jamón, deben ir apareciendo todas las pizzas de jamón que hay en el archivo para ese restaurante y así sucesivamente se va actualizando a como se va digitando.
 - g. El numero de la orden se debe generar automáticamente. La fecha de la compra será obtenida a partir del sistema operativo.
 - h. Si el usuario realiza una nueva búsqueda se deberá actualizar la lista por cada tipo de búsqueda indicada, no se pueden buscar productos en diferentes restaurantes en una sola compra.
 - i. El usuario selecciona el producto que desea e indica la cantidad, esto debe ser almacenado en algún objeto grafico (tabla, lista, etc.). Una vez seleccionado un producto, el usuario podría

cambiar de opinión, eliminarlo y tomar otro antes de pagar la orden. Igualmente se debe ir mostrando el precio que debería pagar por esa orden.

- j. Una vez seleccionados todos los productos de la orden, el usuario efectúa un evento que indique “Pagar”. En este momento se debe mostrar la información completa de la orden incluyendo el precio, tiempo de preparación en un objeto JTextArea además de registrar todos los objetos relacionados con esta información. **El precio debe aparecer tanto en números como en letras.**
 - k. Por cada producto empacado se debe actualizar en el archivo la cantidad de productos en inventario.
 - l. Cada orden se almacena en una archivador de órdenes de compra(no requiere mantenimientos)
9. Ventana de Calificación de órdenes.
- a. En la ventana el usuario podrá calificar la calidad de la orden recibida.
 - b. Se ingresa y según el usuario podrá ver las compras realizadas con el detalle de los productos en él. El usuario selecciona de la lista uno de los productos y se le solicita la calificación de 1 a 5. Esta la puede realizar a cada uno de los productos por separado o al total de la orden.
 - c. La calificación deberá ser almacenada en el archivador de restaurantes en el restaurante correspondiente que emitió la orden. Los restaurantes deben tener un ArrayList de calificaciones
10. Reportes
- a. Reporte de todos los productos indicando código, descripción, restaurante que lo ofrece con su tipo.
 - b. Reporte de los tres mejores restaurantes según su promedio de calificaciones.
 - c. Reporte de compras realizadas por usuarios ordenados por restaurantes.

III- Observaciones

- 1. El programa debe estar estructurado en 3 capas:
 - **Capa vista o interfaz de usuario (UI):** Diseño de ventanas a través de las cuales se comunicará con el usuario
 - **Capa de Lógica de Restaurantes:** Diseño de las clases y sus relaciones.
 - **Capa de Persistencia:** Diseño de la(s) clases que dan persistencia a los objetos. Para efectos de este curso se utilizará una simulación de la capa de acceso a datos, utilizando archivos.
- 2. El lenguaje de desarrollo es Java utilizando el IDE Netbeans 8.2.
- 3. Debe ser desarrollado utilizando el ambiente gráfico de Java, específicamente las librerías gráficas Swing.
- 4. La interfaz con el usuario debe ser amigable y permitir que éste digite únicamente lo necesario, es decir, debe seleccionar las cosas por medio de objetos gráficos como el combo box, listas, botones de opción, entre otros.
- 5. Utilice las bondades de la interfaz gráfica para que el programa sea vistoso. Cada control gráfico tiene una funcionalidad que se ajusta a una situación específica, por ejemplo, para el tipo de usuario (regular o gerencial) utilice por ejemplo botones de radio. El usuario no debe conocer códigos y con el fin de evitar inconsistencias debe digitar únicamente lo estrictamente necesario.
- 6. No delegue al usuario el ejecutar acciones que no le corresponden y deben ser ejecutadas por el sistema, tales como: actualizar cantidad de ventas de cada producto.

7. Utilice comentarios en el código
8. Es importante recalcar como último punto, que se trata de un proyecto que incluye investigación por lo que el profesor brindará ayuda técnica y aclarará dudas con respecto al contexto del proyecto únicamente, la información adicional debe ser de investigación por parte del estudiante.
9. Debe crear el JAR para realizar su ejecución de forma más simple
10. El proyecto es estrictamente individual. Proyectos iguales serán anulados.
11. El día de la defensa del proyecto debe planear una serie de casos ya listos para presentar al profesor su proyecto y tanto el archivo como los archivadores implementados mediante ArrayList deben traer datos previos.

NOTAS:

1. Puntos adicionales

- ✓ Desarrolla la aplicación utilizando una Base de Datos Sql Server para el mantenimiento de productos y las compras (10 pts).
- ✓ Diseño de un archivo de usuarios y que cuando se equivoca tres veces en la clave lo marque como inactivo y que luego el administrador lo vuelva a activar para que pueda volver a ingresar (4 pts).

Validación de la tarjeta de crédito

- Se debe solicitar el número de tarjeta y código de seguridad.
- Los números de tarjeta deben estar formados por 16 dígitos para Visa y Master Card.
- El código de seguridad está formado por 4 dígitos.
 - Si la tarjeta es Visa debe iniciar con 4, si es Master Card con los números que se encuentran entre 51 y 55 inclusive.
- Se debe validar el código de seguridad de la tarjeta, el cual debe ser de 4 dígitos.
- Una vez validado que el número de tarjeta inicie de acuerdo a las reglas de las compañías, se debe validar que dicho número también cumpla con las especificaciones de validez de acuerdo a los siguientes lineamientos.

Para que un número de tarjeta sea válido debe cumplir lo siguiente:

- El último dígito del número de tarjeta es el dígito de verificación (en el caso del siguiente ejemplo corresponde al dígito 7). Éste dígito debe ser el resultado del siguiente algoritmo.
 - ✓ **Paso 1:** Se recorre el número de tarjeta de derecha a izquierda a partir del penúltimo dígito, y en cada posición de por medio, se multiplica cada dígito por 2. Se van almacenando los resultados en otra estructura (arreglo, hilera, etc.).
 - ✓ **Paso 2:** Cada número almacenado a partir del paso uno, se convierte en un solo dígito sumando sus 2 dígitos.
 - ✓ **Paso 3:** Se suman todos los dígitos resultantes
 - ✓ **Paso 4:** La suma de los dígitos (73) se multiplica por 9
 - ✓ **Paso 5:** Si el dígito de las unidades del resultado del paso anterior es igual al dígito de verificación el número de tarjeta es correcto.

Ejemplo:

Validar número de tarjeta: 4 4 8 5 7 3 6 1 9 0 4 6 8 1 3 7

	No. Tarjeta	4	4	8	5	7	3	6	1	9	0	4	6	8	1	3	7
Paso 1	Duplicar dígitos cada 2 posiciones	8	4	16	5	14	3	12	1	18	0	8	6	16	1	6	
Paso 2	Convertir a un solo dígito	8	4	7	5	5	3	3	1	9	0	8	6	7	1	6	
Paso 3	Sumar los dígitos																Suma = 73
Paso 4	Multiplicar suma por 9 (R)																R= 73*9 = 657
Paso 5	Validar dígito de verificación																Las unidades de 657 es 7. 7 es igual al dígito de verificación => Número de tarjeta correcto

Ejemplos de números de tarjeta válidos:

Visa:

4485736190468137
4539930121856504
4929053880070445
4024007106215774

Master Card:

5390395583964086
5517858154140558
5547444953800421
5371235376835429