**Trabajo práctico I**

Michael Moreno Valoyes, Jane Juliana Gutiérrez, Santiago Salazar Ramírez

Ingeniería de Sistemas e Informática, Universidad Nacional de Colombia sede Medellín

Programación orientada a objetos

Jaime Alberto Guzmán

17 de diciembre de 2021

**Descripción general de la solución:**

**Análisis:**

Es pues importante diferenciar los tipos de usuarios en esta aplicación: el mesero y el administrador mismo. La diferencia propia de cada una es la función a desempeñar en la aplicación, dado que los dos poseen diferentes horizontes en el restaurante. El administrador deberá de digitar su usuario y contraseña en caso de que quiera usar sus funciones.

El administrador podrá:

1. Visualizar el estado del restaurante (Visualizar clientes, meseros y mesas disponibles del momento).
2. Observar las ganancias del día.
3. Poder observar qué tan frecuentes son los platillos que se tienen en el menú.
4. Cambiar la jornada (día o noche).
5. Crear, eliminar y visualizar meseros.
6. Reiniciar la caja del día (eliminar los pedidos que se generaron en todo el día).
7. Agregar platillos al menú.
8. Crear,eliminar y visualizar mesas.
9. Crear, eliminar y visualizar platillos.
10. Visualizar el menú.

El mesero podrá:

1.Crear un pedido (se crea un pedido con la mesa correspondiente del cliente, los datos del cliente y los platillos solicitados por el cliente).

2. Visualizar el pedido según la mesa seleccionada.

3. Crear la factura correspondiente a la mesa (en este también se liberará la mesa)

4.Agregar platillos a un pedido según la mesa.

**Diseño e implementación:**

De la UI se usaron métodos dentro de la misma clase UI, los cuales dependiendo de la entrada que otorga el usuario va a cambiar a cierta parte del programa. Ello se hizo con el uso de switch, para que dependiendo del valor vaya a otra parte del programa.

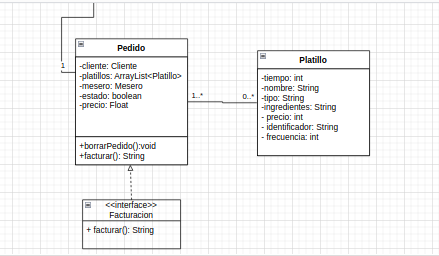
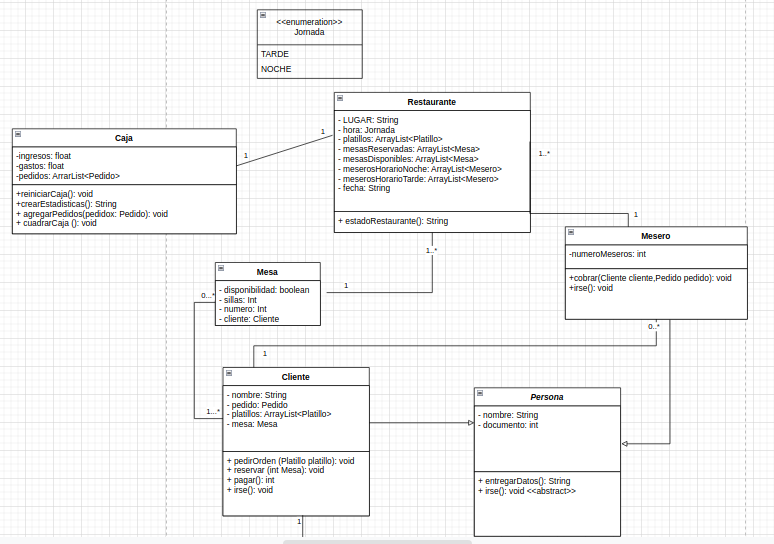
De las funciones del administrador:

1. El programa tiene una lista de mesas ocupadas (reservadas), las cuales están ligadas a un cliente que en su momento está presente. Entonces, para visualizar los clientes del momento se recorre por la lista de mesas ocupadas y se saca el nombre del cliente presente. Los meseros dependen de la jornada en la cual esté presente el restaurante, entonces según la jornada se imprimen los nombres de los meseros (tarde o noche).
2. Las ganancias del día se realizan desde el pedido, el cual está presente en cada cliente. Entonces, a la hora de facturar al cliente se realiza un almacenamiento de dicho pedido en una lista de pedidos. Cada pedido almacena los platillos que se solicitaron, los cuales tienen un precio fijado por el administrador. Ya cuando se elije esta función lo que hace es sumar los precios de los platillos por cada pedido, para luego sumar este valor con el de los demás pedidos.
3. El platillo tiene un atributo llamado frecuencia, el cual sirve para que cada vez que un cliente pide un platillo se le sume 1. Cuando se llama este método se itera por todos los platillos del menú, del cual se imprime la cantidad de veces que ha pedido ese platillo.
4. Función demasiada sencilla. El restaurante por defecto tiene como hora “Tarde”, cuando usted selecciona esta opción puede modificarlo a “Tarde” o “Noche”
5. Para crear a un mesero se pide su nombre, documento y la jornada en donde se encuentra. Dependiendo de la jornada seleccionada el programa guardará al mesero en una lista de meseros de la tarde o meseros de la noche. Para eliminar el mesero se adjunta su documento, lo que hará el programa es buscar el documento por las listas de las dos jornadas de meseros (en caso de que el documento sea igual eliminará al mesero de la lista). Para visualizar iterará por las dos listas de meseros e imprimirá el nombre.
6. Reiniciar caja también es sencillo: declara los ingresos y gastos como 0 (los reinicia) y declara la lista de pedidos como vacía. También reinicia la frecuencia de los platillos del menú.
7. Para adjuntar los platillos deseados debe de introducir el identificador del platillo, en caso de que sea más de uno debe de introducir un “;” entre cada platillo, sin espacio entre ellos. Ej: 1;2;3;4. Para implementarlo lo que se hace es hacer un split por cada “;”, para luego buscar el identificador en la lista de platillos del restaurante (no confundir con el menú, es totalmente diferente).
8. Crear la mesa pide 2 datos: el número de la mesa (lo que identifica la mesa de las demás) y el número de sillas que tiene. Cuando se crea se agrega a una lista de mesas disponibles. Para eliminar se digita el código y se busca en las mesas disponibles. Para visualizar se muestra el número de las mesas disponibles.
9. Para crear un platillo se solicita: el nombre del platillo, el tipo de platillo, los ingredientes del platillo,y el identificador del platillo (que se usará en la aplicación como tal), el precio del platillo y el tiempo de preparación del platillo. Ya luego se agrega a una lista llamada platillos (ahí se almacenan todos los platillos, sin importar si están o no están en el menú). Para eliminar se busca en la lista de Menú y de platillos, en caso de que tengan el mismo identificador pues se elimina de la lista (recuerde que siempre se ingresa el identificador, no el nombre del platillo). Para visualizar los platillos lo que se hace es mostrar los nombres de los platillos de la lista platillo.
10. Visualizar el menú: mostrar los nombres de los platillos de la lista menú.

De las funciones del mesero:

1. Cuando se crea un pedido se piden los datos del cliente(nombre, documento y mesa ). Con estos datos se crea un cliente, se le asigna la mesa correspondiente y ya se empieza con la solicitud de los platillos que quiere para su pedido. Primero se pregunta la cantidad de platillos DIFERENTES que tiene su pedido, para luego pedir el platillo (por su identificador) y el número de este mismo. Se comprueba si este platillo está en el menú y se agrega a los platillos que tiene el pedido (se introduce a la lista de platillos según el número introducido). Luego de ello el mesero debe añadir su documento (se busca el documento según la jornada en donde esté, para comprobar que sí sea un mesero y que está en la jornada correspondiente).
2. Busca la mesa introducida (en mesas reservadas), luego de ello busca el cliente que está relacionada con esa mesa y se busca el pedido que tiene el cliente. Del pedido se muestran los platillos y su respectiva cantidad (por esto es que se usa ocurrences).
3. Al crear la factura se va a mostrar: Nombre del restaurante + fecha del restaurante + la hora del restaurante + nombre del cliente + nombre del mesero + todos los platillos con su precio. Luego de mostrar esto en pantalla muestra el precio de todo el pedido y el número de contacto del restaurante. Además de todo este mensaje lo que se hace es adjuntar este pedido a la lista de pedidos que posee caja (con el precio total incluido).
4. Se podrá agregar platillos a un pedido según la mesa asignada. Se introduce la cantidad de platillos diferentes, para luego introducir el platillo por su cantidad deseada.

**Descripción del diseño estático del sistema en la especificación UML:**



Clases:

Caja: (Se relaciona con el restaurante) (sus métodos y atributos son estáticos)

* Atributos:
  + Ingresos: Las ganancias que obtuvo el restaurante en el día.
  + Gastos: Los gastos que obtuvo el restaurante en el día.
  + Pedidos: Los pedidos que obtuvo el restaurante en el día. (una lista)
* Métodos:
  + reiniciarCaja: Reiniciar los valores de la caja.
  + CrearEstadisticas: Las estadísticas de los pedidos (muestra la frecuencia de los platillos en los pedidos.
  + AgregarPedidos: Lo que dice el nombre, agrega pedidos a la lista de pedidos.
  + CuadrarCaja: Para definir los ingresos del restaurante según los pedidos

Restaurante: (Se relaciona con la caja, mesa y mesero) (sus métodos y atributos son estáticos)

* Atributos:
  + LUGAR: Una constante que define el lugar del restaurante.
  + Hora: La jornada del restaurante (Tarde o Noche).
  + Platillos: Los platillos que tiene el restaurante (una lista), diferente al menú.
  + MesasReservadas: Las mesas en las que ahora mismo están ocupadas por algún cliente.
  + MesasDisponibles: Las mesas en las que no hay cliente.
  + MeserosHorarioNoche: Los meseros que trabajan por la noche.
  + MeserosHorarioTarde: Los meseros que trabajan por la tarde.
  + Fecha: La fecha en la que está el restaurante.
* Métodos:
  + estadoRestaurante:Muestra los clientes actuales del restaurante, meseros y mesas disponibles.

Mesa: (Se relaciona con el restaurante y cliente)

* Atributos:
  + Disponibilidad: Si la mesa está disponible o no.
  + Sillas: El número de sillas
  + Número: El número identificador de la mesa.
  + Cliente: El cliente que posee actualmente.

Mesero: (Se relaciona con el restaurante y el cliente. Hereda de persona)

* Atributos:
  + numeroMeseros: Lo que se dice, el número de meseros total. (estático)
* Métodos:
  + Cobrar: Le cobra a un cliente lo de la factura.
  + Irse: Pues eso, que se va del trabajo.

Cliente(Se relaciona con mesa, mesero y pedido. Hereda de persona)

* Atributos:
  + Nombre: El nombre del cliente.
  + Pedido: El pedido que pidió.
  + Platillos: Los platillos que pidió, se le pasa al pedido.
  + Mesa: La mesa en donde está
* Métodos:
  + pedirOrden: Eso, pide la orden
  + reservar: reserva la mesa.
  + Pagar: Paga el pedido.
  + Irse: Pues eso, se va del restaurante.

Persona: (abstracta)

* Atributos:
  + Nombre: nombre.
  + Documento: El documento con el cual se identifica el Estado.
* Métodos:
  + entregarDatos: Entrega su nombre y documento.
  + Irse: Se va…

Pedido: (Se relaciona con platillo, facturación y cliente)

* Atributos:
  + Cliente: El cliente que posee el pedido.
  + Platillos: Los platillos que tiene el pedido.
  + Mesero: El mesero con el cual está relacionado.
  + Estado: Si está efectuado o no.
  + Precio: El precio total del pedido.
* Métodos:
  + borrarPedido: Se borra el pedido.
  + Facturar: Se factura el pedido.

Platillo: (se relaciona con pedido)

* Atributos:
  + Tiempo: El tiempo de preparación necesario para hacer un platillo.
  + Nombre: El nombre del platillo.
  + Tipo: El tipo de platillo (postre, entrada, fuerte).
  + Precio: El precio del platillo.
  + Identificador: Se usa para poder digitar el platillo desde la plataforma
  + Frecuencia: Las veces que se ha pedido este platillo.

Facturación: (Se relaciona con Pedido) (interfaz)

* Métodos:
  + facturar: Para facturar el pedido.

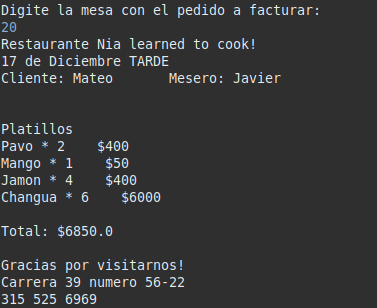
**Implementación de características de la programación orientada a objetos en la implementación:**

* Clases abstractas y métodos abstractos: se realizó este en la clase persona (la cual es abstracta), de las cuales hereda cliente y mesero. El método abstracto es irse(), el cual es para que se retire la persona (dependiendo del cargo en el cual está presente)
* Interfaces: La interfaz empleada se llama facturación, la cual se emplea para que la clase Pedido herede el método facturar.
* Herencia: Ya como se dijo del primero se dice que la herencia se denota en Persona con cliente y mesero. ¿Por qué se hizo así? Porque los dos son personas.
* Ligadura dinámica: se emplea en el método irse(), el cual cambia en las dos clases: mesero y cliente.
* Atributos de clase y métodos de clase: Todos los métodos y atributos que están en restaurante son de clase. El método de clase: estadoRestaurante() (Ya se ha dicho con anterioridad su función, la cual es mostrar los clientes del momento, meseros y mesas disponibles). El atributo de clase: Hora.(La jornada en la cual se encuentra inmerso el restaurante)
* Uso de constante: LUGAR, se encuentra en restaurante. (Para decir cual es el lugar del restaurante).
* Encapsulamiento: En todo el trabajo se mostró este uso del encapsulamiento, desde el uso de private para los atributos hasta el uso de public para los métodos. Un ejemplo es la clase Caja.
* Sobrecarga de métodos y constructores: La sobrecarga de métodos se empleó en Persona con el método irse() (el método se sobrecarga en Cliente). Constructores se emplearon en cliente, pedido y mesa (en la propia UI muestra que cuando se crea uno de estos se hace constructores, también se muestran los constructores en las correspondientes clases).
* Manejo de referencias this para desambiguar y this() entre otras: Se emplea en Pedido, al final del método facturar se llama otro método usando this como argumento.
* Implementación de un caso de enumeración: Se emplea en jornada, para cambiar la hora del restaurante.

**Descripción de cada una de las 5 funcionalidades implementadas:**

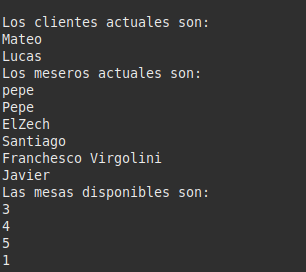
1. Crear factura:

* Descripción de la funcionalidad: El mesero introduce la mesa a la cual quiere crear la factura. De ahí el mismo programa busca el cliente asignado que tiene la mesa, lo cual luego va a buscar el pedido asignado del cliente. Va a imprimir: Nombre del restaurante + fecha del restaurante + hora del restaurante (jornada) + nombre del cliente + nombre del mesero + nombre del platillo con la cantidad que se pidió del platillo. Al final imprime el precio total del pedido, con el lugar del restaurante y el número de contacto del restaurante. Este pedido se agrega a la clase caja (para luego poder cuadrar caja, recordar que se incluye el precio total del pedido con la suma de los platillos). También se vacía la mesa en donde antes estaba el cliente.
* Objetos que intervienen en la implementación: Restaurante, Mesa, Pedido, Cliente y Mesero.
* Secuencia del proceso: Introducción de la mesa a la cual se quiere crear factura → Búsqueda del cliente por la mesa digitada → Búsqueda del pedido por el cliente → Impresión de la factura → Cálculo del coste total del pedido → Agregar el pedido a la caja → Eliminación del cliente de la mesa.
* Captura de pantalla de la creación de factura:

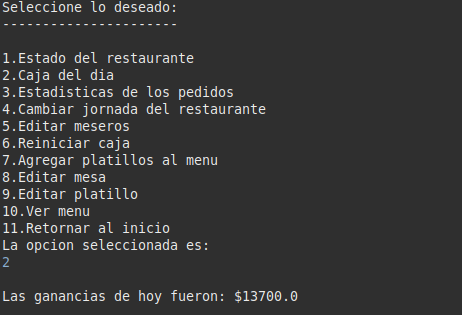


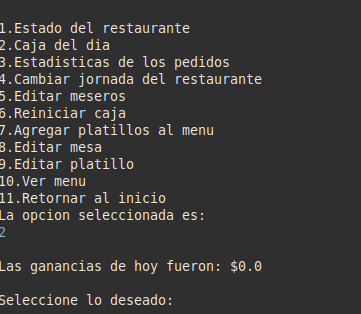
1. Mostrar el estado del restaurante:

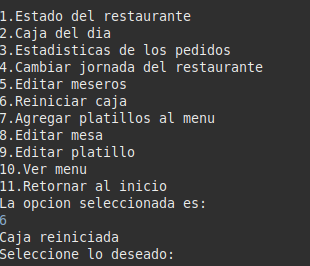
* Descripción de la funcionalidad: Va a mostrar los clientes que ahora mismo están en el restaurante, los meseros que se encuentran ahora mismo (por la jornada que se está haciendo) y las mesas disponibles del restaurante.
* Objetos que intervienen: Restaurante, Mesa, Cliente y Mesero.
* Secuencia del proceso: Buscar las mesas reservadas → buscar los clientes por cada mesa reservada (agrega los nombres de estos clientes al mensaje) → Buscar los meseros según la jornada → Agregar estos meseros al mensaje → Buscar las mesas libres → Agregar el identificador de cada mesa libre al mensaje.
* Captura de pantalla:



1. Ganancias del día (caja del día) con su respectivo reinicio:

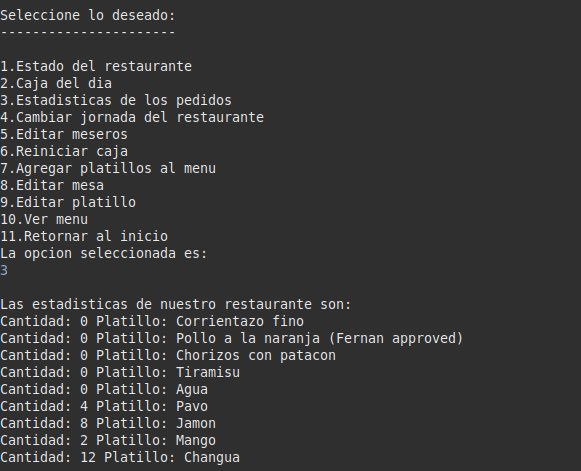
* Descripción de la funcionalidad:El principio de este mismo es que cuando se crea un pedido a un cliente se lleva el pedido del cliente a una lista de pedidos de caja. Ya cuando usted entra a “Caja del día” va a mostrar la suma de los precios de los pedidos que está en la lista de pedidos. Si se fijó de la funcionalidad anterior recordará el cómo se realizó la obtención del precio de cada pedido. Para reiniciar el día es desde otra parte de la interfaz (reiniciar caja), lo que hará es reiniciar los valores de los ingresos y la lista de platillos. (También se reinicia la frecuencia de los platillos)
* Objetos que intervienen: Caja, Pedido y Restaurante (de fondo intervienen Mesa y Cliente).
* Secuencia del proceso: Se agrega el pedido del cliente a la lista de pedidos en caja → Se suma el precio de cada pedido → Se imprime el resultado. Opcional: Reiniciar la caja y los valores que intervienen en él.
* Capturas de pantalla:



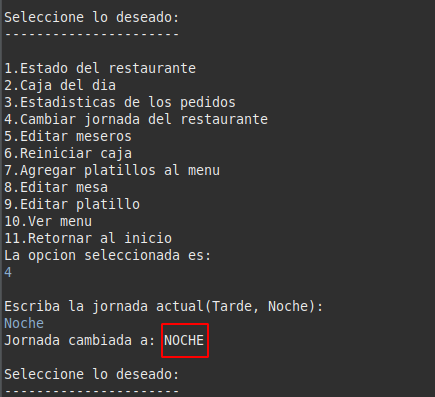


1. Estadísticas de los pedidos:

* Descripción de la funcionalidad: Se almacena en cada platillo un atributo llamado frecuencia, el cual aumenta en 1 según la cantidad de veces que lo hayan pedido (esto se hace cuando se crea el pedido). Ya cuando usted realiza la función de estadísticas de pedidos va a mostrar la frecuencia de todos los platillos del MENÚ (el cual es diferente a la lista de platillos de restaurante, puesto aunque platillos estará en menú este mismo no estará en platillos necesariamente).
* Objetos que intervienen: Caja, Restaurante y Platillo.
* Secuencia del proceso: Se realiza un pedido por parte del cliente → Se incrementa la frecuencia según el platillo y la cantidad de veces que se haya introducido → Se llama a la funcionalidad → Se imprime los platillos con su debida frecuencia.
* Captura de pantalla:

1. Cambiar la jornada:

* Descripción de la funcionalidad: El administrador puede cambiar de jornada de una manera sencilla: seleccionando la opción llamada “Cambiar jornada del restaurante”. Cuando se cambia la jornada del restaurante se cambian los meseros que operan en el restaurante. ¿En qué afecta esto? Que si usted adjunta su documento (como mesero) cuando no está en la jornada en donde trabaja no le va a funcionar la aplicación, haciendo que necesariamente deba de estar en la lista de la jornada especifica.
* Objetos que intervienen: Restaurante y Mesero.
* Secuencia del proceso: Se selecciona la opción de cambiar jornada → Cambia la jornada a la deseada → Restringir a los meseros según la jornada.
* Captura de pantalla:



**Manual de usuario:**

El usuario podrá seleccionar entre 3 opciones al inicio de su sesión: (La selección de la opción se basa en introducir el entero inicial de cada opción).

1. Administrador (el usuario debe de digitar el usuario: admin y contraseña: 1918).
2. Mesero.
3. Apagar el programa.

Administrador:

1. Estado del restaurante: Ya se habló de lo que hace esta funcionalidad.
2. Caja del día: Ya se habló lo que hace esta funcionalidad.
3. Estadísticas de los pedidos: Ya se habló de esta funcionalidad.
4. Cambiar jornada del restaurante: Introduce la jornada deseada (Tarde, Noche).
5. Editar meseros:
   1. Crear mesero: Usted debe de digitar el nombre, documento y la jornada en la que está presente.
   2. Eliminar mesero: Usted debe digitar el documento del mesero al cual quiere eliminar
   3. Visualizar meseros: Muestra los meseros según la jornada
6. Reiniciar caja: Ya se habló lo que hace reiniciar caja.
7. Agregar platillos al menú: Esto es totalmente relevante. Una cosa es crear un platillo y otra cosa totalmente distinta es agregar este platillo al menú. ¿Cómo se hace para agregar platillos al menú? Debe de introducir su identificador (es un número), en caso de que sea más de un platillo a identificar debe de introducir un “;” entre los identificadores. Ej: 13;14;15;16
8. Editar mesa:
   1. Crear mesa: Debe de introducir el número de la mesa (hará de identificador) y la cantidad de sillas que esta posee.
   2. Eliminar mesa: Elimina la mesa con el identificar (número de la mesa).
   3. Ver mesas: visualiza las mesas disponibles.
9. Editar platillo:
   1. Crear platillo: Debe de introducir el nombre del platillo, el tipo de platillo, ingredientes, identificador (este distingue a los platillos de los demás), precio del platillo y tiempo de preparación
   2. Eliminar platillo: Debe de digitar el identificador del platillo para eliminarlo.
   3. Ver platillos: Visualizar todos los platillos (diferente al ver todo el menú, lo visualiza con sus diferentes identificadores).
10. Ver menú: Visualiza los platillos que tiene el restaurante.

Mesero:

1. Asignar mesa y hacer pedido: Debe de introducir el nombre del cliente, documento del cliente, la mesa en donde se va a sentar el cliente, la cantidad de platillos diferentes que quiere introducir. El programa pedirá x (cantidad de platillos diferentes) veces el identificador del platillo con su debida cantidad. Luego de ello se solicita el documento del mesero.
2. Ver pedido: Digita la mesa, de ahí puede ver el pedido que se está ejecutando en el momento.
3. Crear factura a una mesa: Digita la mesa en donde quiere crear la factura (claramente donde un cliente esté presente). Ya luego se creará la factura (tal como se dijo en las funcionalidades).
4. Agregar platillos a una mesa: digita la mesa a la cual quiere agregar los platillos. El programa pedirá los platillos y su cantidad de la misma forma que cuando se crea un pedido.

Notas:

1. Se puede retornar en todas las opciones del menú, es solo seleccionar el número que dice en pantalla.