

Modelado y Programación

Proyecto 2

17 de Enero de 2021

Integrantes:

- | | |
|------------------------------------|--------------------------|
| • Gutiérrez Sánchez Claudia Itzel. | No. de Cuenta: 112002433 |
| • Montiel Manriquez Ricardo. | No. de Cuenta: 314332662 |

Problemática: Negocio de tortas

El dueño de un negocio de tortas quiere hacer una app donde los clientes puedan hacer sus pedidos y automáticamente se les cobre a una cuenta bancaria, para poder hacer un pedido deben estar suscritos al servicio con un ID único. Quiere que se guarde la información del cliente (Nombre, domicilio y su cuenta).

Ya que muchos clientes se encuentran reacios a entregar la información de su cuenta bancaria a la sucursal, el dueño les ha prometido que la información de sus cuentas será protegida.

Cada pedido que haga el cliente tendrá un número de pedido, la información del cliente que hizo el pedido y su orden.

Ya que el cliente cuenta con una suscripción tendrá notificaciones nuevas todos los días (descuentos y/o información de sus pedidos). El negocio cuenta con distintos descuentos dependiendo el día (La tortería solo abre entre semana):

Día Lunes: 10% de descuento en tortas sencillas.

Día Miércoles: 5x3 tortas.

Día Viernes: 10% de descuento en tortas especiales.

Los demás días se cobrarán las tortas a su precio de lista.

Los clientes podrán pedir escogiendo sólo el número de torta que quieren, tomando en cuenta que el menú de la tortería es el siguiente:

1. Pierna Roja con Quesillo \$40
2. Milanesa con Quesillo \$40
3. Chorizo con Quesillo \$40
4. Salchicha con Quesillo \$40
5. Jamón con Quesillo \$40
6. Huevo con Quesillo \$40
7. Pastor con Quesillo \$40
8. Rusa (Pierna Roja, Milanesa y Quesillo) \$60
9. Milchori (Milanesa, Chorizo y Quesillo) \$60
10. Especial (Pierna Roja, Jamón y Quesillo) \$60
11. Hawaiana (Jamón, Piña y Quesillo) \$60
12. Texana (Milanesa, Jamón y Quesillo) \$60
13. Suiza (Quesillo, Queso Blanco y Queso Amarillo) \$60
14. Toluqueña (Salchicha, Chorizo y Quesillo) \$60
15. Pachuqueña (Milanesa, Piña y Quesillo) \$60
16. Norteña (Chorizo, Jamón y Huevo) \$60
17. Lambada (Pierna Roja, Chorizo y Huevo) \$60
18. Negra (Milanesa, Salchicha y Quesillo) \$60

Se trata de un negocio nuevo e innovador por lo que solo se cuenta con una sucursal y el personal a cargo serán 2 robots que se encargarán de cocinar y entregar la comida, cada uno de ellos contará con distintas características dependiendo de la tarea para que se hayan creado. Los robots permanecerán suspendidos cuando no tengan un pedido que realizar o entregar, y se activarán hasta que se les de un pedido, una vez que acaben de procesarlo se volverán a suspender. El robot cocinero estará a cargo de cocinar y empaquetar la comida, mientras que el robot repartidor deberá transportar y entregar la comida.

Nuestro programa funciona como una simulación de dicha tortería durante 5 días. Incluye la creación inicial de los dos robots que trabajan en la sucursal y la creación automática de 20 clientes con sus ID únicos, domicilios y una cantidad de dinero asignada al azar.

Todos los días cinco clientes hacen un pedido. La sucursal procesa el pedido y solo si el cobro al cliente es exitoso un robot procede a preparar cada torta para que después el robot repartidor la “entregue”.

Patrones y Clases en las que se ocupan:

Proxy:

Clases: *Payment, Cuenta, CuentaProxy.*

Se utiliza para proteger la información de la cuenta bancaria del cliente con la que se lleva a cabo el pago. La sucursal le cobra a la CuentaProxy en lugar de tener acceso a la Cuenta original del cliente.

State:

Clases: *EstadoRobot, Cocinando, Empaquetando, Suspendido, Transportandose, Entregando.*

Para representar los cambios de estado por los que pasan ambos robots, de suspenderse a cocinar o transportarse para entregar una orden. Los robots seguirán los siguientes pasos:

Robot Cocinero: *Suspendido → Cocinar → Empaquetar → Suspendido.*

Robot Repartidor: *Suspendido → Transportar → Entregar → Transportar → Suspendido.*

Abstract Factory:

Clases: *RobotFabrica, RobotCocineroFabrica, RobotRepartidorFabrica, Llantas, Bateria, Herramientas, LlantasAntiDerrape, LlantasTodoTerreno, Bateria12H, Bateria24H, KitCocina, GPS.*

Para construir a los dos tipos necesarios de robot. El que cocina tendrá herramientas que el que hace entregas no y viceversa.

Template:

Clases: Tortas, Pachuquenia, Nortenia, Toluquenia, Lambada, Negra, PRQ, MQ, CQ, SQ, Suiza, Texana, Hawaiana, Especial, PQ, Milchori, Rusa, HQ, JQ.

Aunque se trate de 18 tortas distintas, comparten varios procesos a la hora de cocinarlas, usamos Template para poder definir aquellos pasos que cambian dependiendo de la torta que se esté cocinando, como cocinar la carne o agregar extras a la torta.

Observer:

Clases: *Observer, Cliente, SucursalTortas, Sujeto.*

Debido a que se trata de un servicio al que te suscribes nos sirve para llevar un control de los clientes suscritos y poder mandar todas las notificaciones de la sucursal al cliente, como los descuentos o la información de su compra.

Strategy:

Clases: *Cobro, CobroViernes, CobroLunes, CobroMiercoles, CobroNormal.*

Usado para poder calcular el monto que se le va a cobrar al cliente dependiendo del día, ya que se tienen distintos descuentos en los días LUNES, MIÉRCOLES y VIERNES. Cada cobro funciona de una forma distinta y calcula automáticamente cuánto se le debe cobrar al cliente.

Ejecución del Programa:

Estando dentro de la carpeta ' src ' abriremos la terminal y se ejecutará el comando ' javac Main.java ' para compilar el programa. Después ejecutaremos el comando ' java Main ' y con esto se mostrará la simulación del proyecto.

Consideraciones:

Para probar que Proxy está funcionando, todos los días se crea un txt (que se guarda en la carpeta ' src ' junto con los ' .class ') que muestra el nombre, saldo inicial y saldo final de los que compraron tortas ese día.

En nuestro proyecto no ocupamos MVC, debido a que acabamos el código antes de que su uso se volviera obligatorio para el proyecto. Le enviamos un correo a la profesora explicando nuestra situación y dijo que no había problema si **no** agregamos MVC.