

TRABAJO PRÁCTICO

COMEDOR UNQ

INTEGRANTES:

- Lucas Lucarni
- Joaquín Castelanich

MATERIA: Programación con objetos 1

FECHA: 29/11/2024

Alcance

Implementamos todo lo que nos dictaminó el dominio del ejercicio. Realizamos la representación del comedor de la UNQ, la cual tiene su propio stock y además tiene una aplicación a su servicio que se encarga de manipular los datos (pedidos, compras, descuentos, verificaciones, estadísticas).

Junto a ello creamos herramientas que le sirven de ayuda al comedor y su sistema, como lo son los changos y los filtros. Los primeros servirían como enlace entre los comensales y el sistema, en cambio los filtros los utilizamos para traer información al comedor según una serie de criterios.

Sumado a todo esto se encuentra la representación de todo lo que conlleva que un comensal, que compre un producto, con un método de pago en específico, y las todas las combinaciones posibles entre los distintos tipos para sus descuentos o aumentos.

Adicionalmente creamos más filtros de los que pedía el dominio para poder tener un mejor control sobre estos datos. También se encuentran muchas más restricciones que no se aclaraban en el dominio pero las vimos correctas para el funcionamiento de la aplicación.

Modelo:

- Comedor: El comedor es nuestra representación del espacio de la vida real del comedor de la UNQ. Se encarga de registrar comensales universitarios y otorgarles su tarjeta al ser registrado, como también recargar las tarjetas con saldo en cualquier momento. Maneja su propio stock el cual puede agregar productos como también quitarlos al ser comprados. Adicionalmente recibe los pedidos de los comensales, que luego de unas verificaciones, se los delega al sistema que es único para este comedor. Por último el comedor puede filtrar los pedidos de su sistema en base a diversas restricciones para generar estadísticas.
- Sistema de comedor: Es la aplicación del comedor que se encarga de mantener y brindar la información tanto de los comensales registrados en el mismo como también los pedidos, ya sean pendientes de pago o pagos, de cualquier comensal al comedor. Además se encarga de calcular los costos en bruto y con los descuentos aplicados en función al tipo de comensal, la fecha en la que se hizo el pedido y el método de pago con el cual va a pagar ese pedido (que sería el costo real). En base a eso también puede cobrarle al comensal que hizo el pedido.
- Comensal: Representa un comensal en la vida real el cual puede recargar su saldo para que, con la ayuda de un chango, pueda realizar un pedido en el comedor. Existen diferentes tipos de comensal, entre ellos, los comensales universitarios y externos. Estos primeros tienen el beneficio de recibir una tarjeta como método de pago del comedor, en cambio los externos no lo

tienen. Cada comensal tiene diferentes beneficios en cuanto a los costos y productos que pueden pedir.

- Producto: Modela un producto del comedor, los cuales existen diferentes tipos. Pero todos tienen un precio establecido en el comedor. El comensal puede comprarlos siempre y cuando haya stock dentro del comedor.
- Método de pago: Simboliza los diferentes métodos de pago que puede tener un comensal. Estos pueden ser recargados cuando la persona lo requiera. Brindan diferentes descuentos o recargos según el día y el producto. Al pagar un pedido el sistema debita el respectivo saldo del método del comensal.
- Pedido: Plasma toda la información que requiere el sistema para poder calcular y cobrar los productos que requiere un comensal.

Progreso

Comenzamos creando la clase “Comedor” verificando para qué días se encontraba abierto o cerrado y para el registro de comensales universitarios. En los días que se encontraba cerrado nadie puede registrarse ni hacer pedidos. Para regular estas y muchas más excepciones creamos la clase, propia del comedor, “sistema de comedor”.

Implementamos la nueva clase “Comensal”, que luego iba a tener diferentes subclases dependiendo su tipo. Cada comensal realiza un pedido al comedor, para lo que conlleva algún producto/os, un método de pago con el cual va a pagar y la fecha del día del pedido. Acá nos dimos cuenta que para manejar toda esta información sería útil crear una clase “Pedido” que contenga todo lo dicho anteriormente acompañado del comensal el cual hizo el pedido y si fue pagado o no.

También creamos las clases “Producto del comedor” y “Método de pago” para determinar los precios de los productos con sus respectivos descuentos según el tipo de comensal. No todos los productos del comedor pueden ser pedidos por todos los tipos de comensal, por lo tanto, creamos varias restricciones para ello.

En el medio de esto nos dimos cuenta que necesitábamos una representación de agrupar y llevar los productos a la caja para realizar el pedido. Para ello creamos la clase “Chango”.

Al realizar un pedido teníamos que verificar si había stock dentro del comedor como para que ese pedido pueda ser válido. Por lo tanto creamos una instancia dentro del comedor llamada “stock” para representar el stock del comedor y que en cualquier momento pueda agregar productos a su stock. Si el pedido es válido, se le sacaría del stock del comedor los productos del mismo.

Dentro del sistema del comedor creamos dos instancias para los pedidos. Esta primera instancia, llamada “pedidos pendientes”, la representamos con un diccionario ya que el enunciado no lo especificaba pero nos resultó conveniente y

lo más evidente a la hora de asociar un comensal con su pedido. El diccionario nos brindaba que un comensal no pueda tener dos pedidos pendientes de pago y que sea de fácil acceso el pedido por cada comensal único. La otra instancia son los “pedidos pagos” los cuales luego al implementar la realización del pago de un pedido nos iba a ser de mucha ayuda.

Antes de que un comensal realice un pago, hicimos que nuestra clase “Pedido” pueda saber su costo en bruto y además su costo real (con los descuentos aplicados).

Asimismo, al comensal pagar su pedido pendiente en el comedor, se le debita el dinero del mismo al método de pago asignado al pedido del comensal. Allí creamos una excepción para cuando el comensal no tenga el suficiente dinero como para costear el pedido. Este costo real es calculado a través del sistema que a su vez utiliza “double dispatch” para todas las combinaciones posibles de descuentos y además un “Handler de descuento” que efectúa los descuentos según el día de la semana.

Creamos un mensaje del comedor para que en cualquier momento pueda cerciorarse de que tenga más plata o igual que la suma de dinero entre los pedidos pagos.

A la hora de crear los filtros para la información que requería manejar el comedor sobre ciertos pedidos con características específicas, creamos una nueva clase “Filtros”. Adicionamos más filtros de los requeridos y además es posible combinarlos entre ellos de infinitas formas.

Para terminar con el dominio del comedor implementamos unos últimos mensajes del mismo que nos iban a brindar información sobre estadísticas de sus pedidos, ganancias, stock y productos.

Al terminar con el dominio, pulimos toda la implementación refactorizando todos los mensajes posibles para que este bien encapsulado. Y corroboramos que todas las excepciones estén bien lanzadas.

DIFICULTADES

El trabajo en sí, no nos resultó muy complejo si no que al ser muy extenso en cuanto requerimientos del dominio, tuvimos que tener en cuenta muchas partes en forma conjunta para no perder la formalidad de lo que conlleva un comedor.

Sobre todo lo más enrevesado lo tuvimos a la hora de implementar los “test” con todas las combinaciones posibles entre comensales, métodos de pago, productos, y los días de la semana.

Para hacer el “handler de descuento” tuvimos la idea de crearlo con un diccionario, pero luego comprendimos que no era lo más comprensible para el que revise la implementación. Por lo tanto introducimos nuevas clases que se ayudan entre sí para aplicar el descuento según ciertas condiciones.

También a la hora de hacer los filtros nos supuso un grado de complejidad mayor pero con la ayuda del TDD nos fue mucho más cómodo y rápido para implementarlo.

CONCLUSIONES

Para concluir, creemos que utilizamos mayormente todos los conceptos aprendidos durante las clases de la materia.

Además lo que nos llevamos de la realización del trabajo son las habilidades y conceptos que aprendimos de manera avanzada sobre el paradigma de la programación con objetos.