

# Code Challenge 5 Restaurante - 202030542

Para el Code Challenge 5 de la Administración de Restaurantes se utilizaron todos los patrones propuestos, en las siguientes secciones se explica en donde fueron utilizados y porque.

## Patrón Builder

El patrón builder fue el primer patrón utilizado para la creación de Pedidos, se utilizó un patrón builder debido a que es un patrón que permite la creación de clases complejas con muchas funciones con facilidad, para esto se implementó la clase PedidoBuilder que es la encargada de crear clases Pedido a medida.

Adicional a esto se creó un Director, se sabe que es necesaria la creación de combos, estos están predefinidos, por lo que una clase Director que conozca cómo están compuestos estos combos predefinidos es muy conveniente, por lo que el Director por medio de sus diferentes métodos make y un PedidoBuilder es capaz de crear diferentes combos si fuese necesario, así como el PedidoBuilder por sí mismo es capaz de crear pedidos que no son combos.

## Patrón Strategy

El patrón Strategy fue utilizado para la implementación de descuentos, esto porque los descuentos aunque comparten la lógica de que debe de reducir el total que el cliente paga, la forma y cantidad de descuento varía dependiendo del tipo de descuento, por lo que el patrón Strategy permite alterar la manera en que los descuentos son aplicados basados en diferentes estrategias, la forma común de aplicar descuento se encuentra en la interfaz DescuentoStrategy, pero esta es implementada por los dos descuentos identificados, DescuentoClienteFrecuente y DescuentoMartesPasta, cuya forma de aplicarse es diferente, y la cantidad de descuento es diferente.

## Patron Composite

El patrón Composite fue utilizado para la definición de productos que pueden ser parte de los pedidos, se utilizó composite porque este patrón permite la definición de clases de forma jerarquizada, siendo que las clases de abajo pueden consultar a clases de arriba en la jerarquía en busca de la obtención de respuestas, en este caso solo hay un nivel de la jerarquía, que es la clase que implementa a los Producto, que sabe como obtener el precio y nombre de cada cosa, entonces cada clase que implemente Producto, como los platos principales, bebidas, acompañamientos, y extras se consultan a sí mismos, esta

implementación permite la creación de clases más complejas si fuese necesario, y el agregado de más tipos de productos si el sistema lo requiriera en algún momento.

## Patrón Command

El patrón Command es un patrón que nos permite el encapsular métodos que se quisiera que fuesen comunes para varias clases, y que a través de una interfaz de pudiesen aplicar esos métodos sobre otras clases, en este caso se usó el patrón Command para la realización de acciones en la cocina, la cocina permite que se realicen acciones, estas son los Comandos, estos comandos pueden ejecutarse o cancelarse, entonces para que un método pueda ser aceptado por la cocina este debe de implementar la interfaz de Comando que es la que sabe que metodos utiliza la cocina para hacer sus acciones, que como se mencionó anteriormente es ejecutar y cancelar ese comando, actualmente hay una accion identificada y su cancelacion, que es el envio de un pedido, y su cancelacion si esta confirmado, entonces la clase EnviarPedidoCocina es un comando a traves del cual se puede hacer la accion de enviar un pedido a la cocina y cancelar el pedido ya que implementa la interfaz Comando, si en algun momento se quisiera agregar otras acciones sobre la cocina, bastaria con crear mas clases que implementen a Comando ya que este es el que sabe que se necesita para que la cocina acepte dicha accion.

## Patrón Adapter

El patrón Adapter es un patrón muy poderoso ya que la idea de este es servir de puente entre dos cosas, un adaptador permite el ingreso de una clase A por ejemplo para que la clase B pueda entender como funciona, por lo mismo el patrón Adapter fue usado para el manejo de XML y JSON, ya que la cocina maneja XML y la aplicación maneja JSON es necesaria alguna forma de hacer que ambos puedan entenderse, esta forma es por medio del XMLAdapter, un adaptador que hace que las instrucciones JSON puedan ser convertidas a XML y asi la cocina puede entender a nuestra aplicación.