

PROYECTO PRACTICO PROGRAMACIÓN ORIENTADA A OBJETOS

Food Court es un término que se puede traducir como “agrupación de restaurantes” en los que existe un espacio común para sentarse y poder consumir la comida comprada en los diferentes establecimientos.

Un ejemplo de ello se encuentra en los centros comerciales y aeropuertos. Este modelo gastronómico permite a los visitantes tener una gran variedad de restaurantes y estilos de comida, que van desde franquicias hasta cadenas de restaurantes especializadas. En la figura 1 se muestra un Food Court típico.



Figura 1. Food Court

La diferencia principal con el modelo de restaurante individual es que en éste solo se puede consumir comida en el interior del establecimiento, mientras en el modelo Food Court, los restaurantes operan uno al lado del otro, con un comedor común centralizado para acomodar a los clientes.

Otro modelo gastronómico es el **Food Halls** que se caracteriza porque los consumidores se interesan por experiencias del campo a la mesa, donde lo que se busca es poder disfrutar una buena comida en un espacio agradable, con diversidad de ofertas, no solo gastronómicas sino también de esparcimiento ya que muchos de estos se encuentran al aire libre, rodeados de naturaleza y espacios abiertos para el disfrute de la familia y las mascotas.

Este tipo de establecimientos, que son tendencia en Estados Unidos y Reino Unido desde hace varias décadas, están revolucionando la concepción de la plazuela de comida tradicional, pues proponen añadirle cierta sofisticación integrando restaurantes premium para alejarse de los espacios con los que ya cuentan los centros comerciales. En la figura 2 se muestra un Food Hall.



Figura 2. Food Hall

Actualmente en el oriente cercano de Antioquia se encuentran varios Food Halls, algunos de ellos ubicados en la vía Ilanogrande y la vía El Porvenir Aeropuerto zona Barro blanco. La aplicación de este modelo gastronómico en el oriente de Antioquia ha permitido que la atención se centre en la mesa del comensal y este puede hacer su pedido a los diferentes establecimientos que hacen parte del Food Hall, lo que ha presentado problemas en la forma que gestionan los pedidos y los pagos en cada establecimiento.

Algunos de las situaciones identificadas son:

- Luego que los usuarios o comensales realizan el pedido en la mesa, la persona encargada de la atención debe desplazarse por cada uno de los locales y entregar el pedido a cada uno de estos, y en muchos casos están retirados el uno del otro, lo que hace que el mesero se deba desplazar grandes distancias
- La forma en la que el mesero realiza el pedido en cada local es manual, es decir el pedido se toma en una agenda de papel y luego a cada establecimiento se le entrega su respectiva orden
- En caso de que se haga una modificación en el pedido, el mesero debe ir a la mesa, tomar el cambio y luego desplazarse hasta el local para hacer el ajuste
- Normalmente cada mesero atiende una mesa, pero en caso de que los usuarios deseen agregar más productos al pedido no lo pueden hacer con otro mesero, ya que este no conoce el estado previo del pedido, haciéndose necesario esperar a que el mesero inicial puede modificar el pedido
- Los meseros deben estar constantemente recorriendo los locales comerciales para preguntar si el pedido ya está listo y poder llevarlo a la mesa
- Al momento de pagar, al usuario final se le entrega una sola factura, pero previamente el mesero debe hacer el consolidado de cada uno de los locales, lo que implica que este debe tener muy claro lo que corresponde a cada uno de los establecimientos
- Luego que el Food Halls cierra y no atiende a más usuarios, el mesero debe generar el consolidado de las ventas del día, debe separarlo por establecimiento y entregar el informe final de estas

Por las anteriores situaciones descritas, **La empresa Food Alfa S.A.**, lo ha contratado a usted como desarrollador Junior para que solucione el problema

Los requisitos iniciales que tiene Food Alfa son:

- Se debe solucionar el problema utilizando Python o Java
- La solución se debe desarrollar utilizando el paradigma orientado a objetos
- Se debe tener por lo menos tres roles; administrador general, mesero y cocinero
- Debe existir una pantalla de visualización por cada rol asignado, donde se muestre las funcionalidades propias de cada uno
- El administrador general podrá crear, modificar y eliminar a los otros roles
- El administrador general podrá crear nuevos locales comerciales y los productos de estos
- Se podrá o no tener un administrador por cada local comercial. Este criterio será establecido en las reuniones posteriores de requisitos y deberá ser justificada por el equipo de desarrollo

- El administrador general será el encargado de realizar el cierre completo del día, el cual mostrará un resumen de las ventas generadas por local, mesero y mesa
- El mesero será el único que puede generar y modificar pedidos
- Cuando un pedido sea generado, se debe notificar de esté al cocinero de cada establecimiento. El cocinero podrá aceptar o rechazar el pedido
- Luego que un pedido está preparado y listo para ser entregado a la mesa, el cocinero deberá notificar al mesero
- En la pantalla de visualización del mesero deberá aparecer una notificación por parte del cocinero indicando que el pedido puede ser recogido y entregado a la mesa
- Cuando un cliente solicite la factura, el mesero deberá cerrar el servicio y el sistema deberá generar de forma automática el consolidado a cada establecimiento
- En la pantalla de visualización del administrador general (o por local) se podrá ver el estado de los pedidos. Se debe mostrar si está activo o cerrado, la mesa que lo realiza, el mesero asignado y el consolidado de pagos
- El administrador general podrá filtrar y visualizar en su pantalla las ventas por local, mesa, mesero y fecha

En caso de que se generen nuevos requisitos deberán ser documentados y justificados.

Para la solución del problema el equipo de desarrollo deberá entregar los siguientes elementos:

- El análisis general del problema, donde se identifiquen los actores, subproblemas, variables y otros elementos de interés
- Diagramas de contexto, de clases, y entidad-relación. Si es necesario se pueden tener diagramas individuales por funcionalidad que permitan la lectura final.
- La aplicación final implementada usando Python o Java donde se vea claramente el uso de clases y se aplique los conceptos del paradigma orientado a objetos, como herencia, cohesión y acoplamiento. **Esta aplicación se deberá entregar como un ejecutable de un solo archivo (.EXE), permitiendo la movilidad y eliminando la dependencia de un IDE.** El ejecutable debe tener su propio icono
- Un manual de usuario donde se explique a través de ejemplos, de pantallazos y otros elementos que usted considere necesarios, el funcionamiento completo de la solución implementada