

Restaurant (Reducido)

Un conocido restaurante nos pide realizar un programa para administrar su salón comedor. El mismo cuenta con mesas, de las cuales conocemos el número, la cantidad de personas que pueden sentarse y el estado (libre u ocupada).

También se nos informa que cuenta con un plantel de mozos. Cuenta además con varios cocineros, de los que nos informan la cantidad de años de experiencia que tienen y varios chefs, de los que conocemos el título que poseen (en caso de no poseer título oficial ese dato estará vacío).

De todos los empleados del restaurante mencionados anteriormente, nos informan el nombre, legajo y fecha de ingreso.

El restaurante cuenta con un menú en el cual figuran sus platos y bebidas. De cada uno de estos conocemos el nombre y el precio. De los platos además nos dan su descripción, un indicador que dice si es un plato vegano y otro que dice si es apto para celíacos. De las bebidas nos informan además su tamaño en centímetros cúbicos y un indicador que dice si tiene alcohol.

Cada vez que una mesa realiza un pedido al mozo, éste toma nota y lo ingresa en el sistema anotando los siguientes datos: mesa a la que pertenece el pedido, mozo que la atiende, cocinero que va a hacerse cargo del pedido, forma de pago (Efectivo, Crédito o Débito) y los productos pedidos.

En el caso que una mesa realice el pago en efectivo se le hará un descuento que se calculará de la siguiente forma: A los platos pedidos se les descontará un 5% salvo a los platos veganos que no llevan descuento y en el caso de los platos aptos para celíacos tendrán un 2% de descuento adicional. En el caso de las bebidas consumidas, el descuento es de 10% para las que no contiene alcohol y de 3% en para las que si contienen.

Basado en el enunciado descripto, realizar:

1. El diagrama de clases que lo modelice, con sus relaciones, atributos y métodos.
2. Crear un package de dominio con las clases que modelicen la solución y que tengan solo propiedades, constructores, getters, setters, toString y si es necesario equals y hashCode
3. Crear un package de servicio con las clases de negocio o servicio que sean necesarias para la ejecución de los casos de test que se plantean debajo.
4. Casos de Test:

1. Crear un Salon
2. Debe contener la siguiente cantidad de mesas.
 1. Mesa de 2 comensales x 2
 2. Mesa de 4 comensales x 2
 3. Mesa de 6 comensales x 2
3. Debe contener el siguiente menú:
 1. Bebidas
 1. Agua x 500
 2. Agua con Gas x 500
 3. Vino de la casa x 500
 4. Vino de la casa x 1000
 5. Cerveza
 6. Gaseosa
 2. Platos
 1. Hamburguesa, vegana: no, celíaco: no
 2. Hamburguesa Vegie, vegana: si, celíaco: no
 3. Hamburguesa Celiaco, vegana: no, celíaco: si
 4. Fideos con salsa, vegana: no, celíaco: no
 5. Fideos con salsa Vegie, vegana: si, celíaco: no
 6. Fideos con salsa Celiaco, vegana: no, celíaco: si
 7. Ensalada, vegana: si, celíaco: si
 8.
4. Debe contener los siguientes empleados
 1. Mozos x 2 del tipo que quieran
 2. Cocineros x 2
 3. Cheff x 1
5. Escenarios de Comanda
 1. Mesa de 2 con una 2 platos y bebidas sin alcohol, abono en Efectivo.

2. Mesa de 2 con una 2 platos veganos y bebidas sin alcohol, abono en Efectivo.
 3. Mesa de 4 con una 2 platos veganos, 1 celiaco, 1 comun y 2 bebidas sin alcohol y 2 con alcohol, abono Tarjeta
 4. Mesa de 4 con una 4 platos veganos y 10 bebidas con alcohol, abono en Efectivo
 5. Mesa de 6 con una 6 platos celiaco y 4 bebidas con alcohol y 2 sin alcohol, abono en Efectivo
 6. Mesa de 6 con una 6 platos celiaco y 4 bebidas con alcohol y 2 sin alcohol, abono en Tarjeta
 7. Con todas las mesas de 2 ocupadas, controlar que no se pueda crear otra mesa de 2, indicando el error, "no hay mesas disponibles".
 8. Liberar una mesa de 2
 9. Mesa de 2 con una 2 platos y bebidas sin alcohol, abono en Tarjeta.
 10. Cualquier otro que se les ocurra...
-
6. Cada escenario de comanda, hacer la liberación de la mesa y mostrar por pantalla el número de la mesa, el medio de pago y el resultado del total de la comanda, sin descuento y cuanto con el descuento, si aplica.
 7. Implementar el informe de:
 1. **pedidosPorCocinero** informe por pantalla el legajo, nombre y total de comandas atendidas.
 2. **pedidoConMasConsumibles** muestre por pantalla la comanda que contenga el mayor numero de consumibles, obtenido del listado de todas las comandas.
-
5. Para la entrega
 1. Crear un proyecto java en Eclipse, con el siguiente formato de nombre
 2. dv-pav-final-restaurant-[apellido]-[nombre]
 3. donde:
 1. [apellido] será reemplazado por cada uno de sus apellidos, en caso de tener más de uno, colocar solo el primero.
 2. [nombre] será reemplazado por cada uno de sus nombres, en caso de tener más de uno, colocar solo el primero.

3. IMPORTANTE: tanto el apellido y el nombre debe estar en minúscula.
4. Entregar el proyecto java de eclipse en formato zip, ejemplo:
dv-pav-final-restaurant-montero-francisco.zip