

Proyecto 1

Reglas del negocio

Su equipo de desarrolladores ha sido seleccionado por una cadena de Pizzas para desarrollar su nuevo servicio de recepción y entrega de pedidos. En este proyecto debe desarrollar un sistema que sea capaz de leer los datos desde un archivo que le será proporcionado y configurar cada pizza del pedido, indicar el monto a cobrar. Al final del día el usuario puede obtener otro archivo con el resumen de las ventas del día, por tamaño de pizza, indicando los ingredientes solicitados de forma de mantener un inventario actualizado de tales elementos.

Las características del sistema a desarrollar son las siguientes:

- EL sistema contará con tres tamaños de pizza a seleccionar: personal, mediana y familiar
- La pizza básica será la margarita, la cual lleva salsa de tomate y queso, solamente. Los ingredientes adicionales se cobraran de acuerdo al tamaño de la pizza y precio del ingrediente adicional.
- La moneda a usar será una denominada Unidad Monetaria (UM).
- Los precios de la pizza básica (margarita), por tamaño se muestran a continuación:

Tamaño	Precio (en UMs)
Personal	10
Mediana	15
Familiar	20

- En la siguiente tabla se indican los ingredientes adicionales con su precio por tamaño de pizza:

Ingrediente	Personal	Mediana	Familiar
Jamón	1.5	1.75	2.00
Champiñones	1.75	2.05	2.50
Pimentón	1.5	1.75	2.00
Doble queso	0.80	1.30	1.70
Aceitunas	1.80	2.15	2.60
Pepperoni	1.25	1.70	1.90
Salchichón	1.60	1.85	2.10

- El archivo será un archivo de texto con extensión **.pz** . La estructura del archivo a leer con los datos del pedido será la siguiente:

```
COMIENZO_PEDIDO
Nombrecliente; fecha pedido
Tamaño pizza; ingrdiente adicional; ... ; ingrediente adicional
```

```

...
Tamaño pizza; ingrediente adicional; ... ; ingrediente adicional
FIN_PEDIDO
...
COMIENZO_PEDIDO
Nombrecliente; fecha pedido
Tamaño pizza; ingrdiiente adicional; ... ; ingrediente adicional
...
Tamaño pizza; ingrediente adicional; ... ; ingrediente adicional
FIN_PEDIDO

```

- Como se puede observar el archivo puede contener más de un cliente, además cada cliente puede tener más de una pizza.
- Tanto el tamaño de la pizza como los ingredientes estarán en letras minúsculas
- El sistema debe generar un archivo resumen de las operaciones con el siguiente formato:

Fecha: dd/MM/yyyy

Venta Total: 123456.78 UMs

Ventas por pizza (sin incluir adicionales):

Tamaño	Unidades	Monto UMs
Personal	1234	1234567.89
Mediana	1234	1234567.89
Familiar	1234	1234567.89

Ventas por Ingrediente:

Ingredientes	Unidades	Monto UMs
Jamón	1234	1234567.89
Champiñones	1234	1234567.89
Pimentón	1234	1234567.89
Doble Queso	1234	1234567.89
Aceitunas	1234	1234567.89
Pepperoni	1234	1234567.89
Salchichón	1234	1234567.89

(nota: las líneas verticales no forman parte del informe, son opcionales)

- Estos renglones se repiten por cada fecha de venta.
- Los nombres de los tamaños de pizza y de los ingredientes deben ser presentados como se ven en el ejemplo (tipo título)
- Los datos de todas las ventas realizadas deben quedar persistentes en una base de datos SQLite, para que así al abrir el sistema se puedan recuperar el histórico.

Condiciones

- El proyecto se podrá realizar en equipos de hasta un máximo de 4 personas.
- Cada equipo debe indicar un repositorio github donde colocará el código fuente, identificando en el mismo los integrantes del equipo. Si alguno uso un alias, se debe indicar el nombre que corresponde a ese alias. Este repositorio debe ser de acceso público

- Se debe notificar por email al profesor los integrantes del equipo y dirección del repositorio github, a más tardar el **viernes 12 de junio de 2020 a las 11:59 pm**, indicando quien será el capitán del equipo. **De no realizar este paso se considerará que no realizará el proyecto**
- En este repositorio debe haber clara evidencia del trabajo de todos los integrantes del equipo, el mismo se revisará de la cantidad de commits que realice cada integrante del equipo.
- Se suministrará dos archivos de pedidos para que realice sus pruebas, sin embargo la evaluación del funcionamiento se hará sobre otros archivos para comprobar total funcionamiento del sistema.
- De detectar alguna falla con respecto a alguno de los archivos usados para las pruebas, el sistema no debe dejar de funcionar, debe dar una indicación al usuario y solicitar la carga de un archivo correcto.
- En el proyecto se evaluará el uso de:
 - Programación Orientada a Objetos
 - Uso de la base de datos
 - Separación de responsabilidades en las partes del código (Definición de módulos y paquetes)
 - Detección de posibles errores (archivo no existente, incorrecto tipo de archivo, archivo mal formateado, datos incorrectos)
 - Claridad del código
 - Buen uso de la documentación de las funciones usando el doc-comments ("" doc "")
 - Por su puesto el correcto funcionamiento del proyecto en cada uno de los archivos usados para probarlo.
- El proyecto será presentado en la plataforma **jit.si** el **día viernes 26 de junio** en el orden en que fueron recibidas las notificaciones del repositorio github. La presentación se hará en la **hora de la clase (7:30 am a 10 am)** y **deben estar presentes todos los integrantes del equipo.**
- A más tardar el día **25 de junio de 2020** se debe entregar un pequeño informe que indique los integrantes del equipo, nombre dado al proyecto, dirección del repositorio github y una breve descripción del funcionamiento del proyecto. Este informe debe ser entregado a la plataforma M7 en la tarea que se creará para tal fin en formato PDF, **solo por el capitán del equipo.**

Punto extras de participación

- Se puede agregar funcionalidades adicionales, las cuales generarán puntos de participación
- Cada funcionalidad adicional, hasta un máximo de cuatro (4), le otorgará un punto de participación a cada uno de los integrantes del proyecto
- La funcionalidad adicional debe estar documentada en el informe final, explicando que hace y como se demuestra tal funcionalidad implementada
- No se considerará funcionalidad adicional, ningún tipo de formato especial al informe de salida generado por el sistema
- De requerir modificar los archivos de entrada al sistema para aprovechar la funcionalidad adicional, debe suministrar al menos un ejemplo para probar.

Evaluación

Objeto evaluado	Criterio	%
Funcionamiento del proyecto 100%	Completa funcionamiento con cada uno de los archivos que se le coloquen de prueba, los suministrados y los generados aparte por el profesor.	50
Notificación del repositorio de github.	Haber notificado por email los integrantes del equipo y dirección del repositorio de github y permitiendo el acceso público.	5
Limpieza y comentario del código	Revisión del código en github, correctamente indentado, buen uso de las reglas de codificación (pep08), comentado de las funciones del código	10
Uso de POO, base de datos y manejo de excepciones	Correcto uso de los conceptos de POO. Configuración de la base de datos (entidades) para la persistencia de los datos y manejo de excepciones para la detección de posibles errores	20
Prueba de trabajo en equipo	Medido sobre los commits realizados en github por cada integrante del equipo	10
Entrega del informe	Entrega antes de la fecha de vencimiento el informe que respaldará el proyecto, con indicación de la funcionalidad adicional.	5