

Programación II – Evaluación 2 (25% del ramo)

Problema

Estamos desarrollando una aplicación para un restaurante que permita gestionar carga de ingredientes, stock, carta restaurant, pedidos y generar boletas. La solución debe proporcionar una interfaz gráfica para los usuarios, permitir la gestión de pedidos con diferentes productos y generar boletas detalladas en formato PDF que puedan ser impresas o guardadas.

Se ha dispuesto un video (Falta el video aún) software en funcionamiento para que puedan revisar con más detalle.

Los menús para ofrecer están fijados por el cliente y son:

Menús	Precio de venta	Ingredientes
Papas fritas	500	5 x papas
Pepsi	1100	1 x Pepsi
Completo	1800	1x vienesa, 1x pan de completo, 1x tomate, 1x palta
Hamburguesa	3500	1x pan de hamburguesa, 1x lámina de queso, 1x churrasco de carne
Panqueques	2000	2x panqueques, 1x manjar, 1x azúcar flor
Pollo frito	2800	1x presa de pollo, 1x porción de harina, 1x porción de aceite
Ensalada mixta	1500	1x lechuga, 1x tomate, 1x zanahoria rallada

Descripción de la Pestaña 1: Carga de ingredientes

Propósito: La pestaña 1, titulada “Carga de ingredientes” está diseñada para permitir a los usuarios cargar los ingredientes desde un archivo .csv que será dado por el profesor. Debe contar con un botón de carga de archivo y un treeview para visualizar la carga del archivo. El archivo tiene 3 columnas (nombre, unidad, cantidad). Se debe agregar un botón llamado **Agregar al stock**, que agregue los ingredientes de la tabla treeview al stock.

Descripción de la Pestaña 2: Stock

Propósito: La pestaña 2, titulada “Stock” está diseñada para permitir a los usuarios gestionar el stock de ingredientes. Los usuarios pueden añadir nuevos ingredientes, eliminar ingredientes existentes, y visualizar la lista actual de ingredientes disponibles. Además, hay un botón llamado **Generar menú** que crea los menús prefijados a partir de la disponibilidad de ingredientes en el stock (ejemplo: si no hay palta, no se puede crear el menú completo).

Descripción de la Pestaña 3: Carta restaurant

Propósito: La pestaña 3, titulada “Carta restaurant” permite visualizar un archivo en formato PDF a partir de los menús disponibles. Se debe crear un botón llamado **Generar carta PDF**, que busque los menús disponibles en tiempo real y después los convierta a PDF.

Descripción de la Pestaña 4: Pedido

Propósito: La pestaña 4, titulada “Pedido” está diseñada para gestionar la selección de menús a partir de los ingredientes disponibles en el stock. Esta pestaña permite ver qué menús pueden prepararse, añadirlos a un pedido, gestionar las cantidades, y finalmente generar una boleta en formato PDF.

Archivos esperados del proyecto

Como orden básico de entrega de los archivos del proyecto se esperan mínimamente estos archivos (puede separar en más archivos si lo considera más claro):

- **Restaurante.py** → interfaz y funciones gráficas + carga de archivo csv con ingredientes (archivo principal).
- **Ingrediente.py** → Clase ingrediente considerando estructura del mismo.
- **Stock.py** → Clase stock con métodos de agregar, modificar y/o eliminar ingredientes al stock.
- **Pedido.py** → Clase pedido con métodos de agregar o quitar menús, modificar stock y calcular totales.
- **Menu.py** → Clase menú y métodos asociados + generación de menús por defecto.
- **Menupdf.py** → Funciones para generar el menú disponible respecto al stock en PDF.
- **Boleta.py** → Clase boleta con datos y métodos de generación de PDF.

Criterios de Evaluación

Criterio	Descripción	Excelente (10 pts)	Bueno (8 pts)	Suficiente (4 pts)	Insuficiente (1 pt)
Peso (%)					
Implementación de la Orientación a Objetos 30%	Uso correcto de clases, atributos, comportamiento de objetos	Utiliza todos los conceptos de OO correctamente	Utiliza la mayoría de los conceptos de OO	Utiliza algunos con errores	Uso incorrecto
Funcionalidad 15%	Cumplimiento de requisitos especificados	Cumple con todos sin errores	Cumple con la mayoría	Cumple algunos, con varios errores	No cumple
Presentación Oral 25%	Claridad, organización, diagramas, solución, manejo de preguntas	Excelente, clara, cumple tiempo	Buena, clara, algunos detalles	Suficiente, incompleta	Pobre, omite elementos
Implementación de Interfaz Gráfica 15%	Uso de GUI adecuada	Bien diseñada, intuitiva y funcional	Funcional con detalles menores	Básica, varios aspectos a mejorar	Pobre o inexistente
Evaluación del Diagrama de Clases 10%	Representación correcta de clases y relaciones	Claro, completo y correcto	Correcto, con detalles a mejorar	Con errores o ausencias	Incorrecto o incompleto
Repositorio GitHub y LaTeX 5%	Informe en L ^A T _E X incluido como archivo .tex en el repositorio. Todo el proyecto debe estar en GitHub siguiendo las buenas prácticas vistas en clases.	Informe completo en LaTeX, repositorio organizado con todas las entregas y siguiendo buenas prácticas	Informe en LaTeX con repositorio en GitHub, aunque con detalles menores en buenas prácticas o estructura	Informe parcial en LaTeX o repositorio incompleto o con pocas buenas prácticas	No se entrega en LaTeX o sin repositorio GitHub

¿Qué debo hacer?

Debes crear el diagrama de clases del problema propuesto y realizar la codificación usando POO e interfaces gráficas. Puedes basarte en el video de la aplicación dispuesto en la plataforma.

¿Qué debo entregar?

- Presentación PPT en grupos de 2–5 para el miércoles 15/10/2024 (sección 1) y 17/10/2024 (sección 2).
- Debe contener: título, integrantes, fecha y logo del departamento.
- El diagrama de clases con la solución propuesta.
- Presentación de la solución en código explicando el flujo principal.
- Presentación de la interfaz visual.
- Demostración de la funcionalidad total del programa.

La presentación tiene un tiempo máximo de 15 minutos, más una tanda de preguntas de 5 minutos.