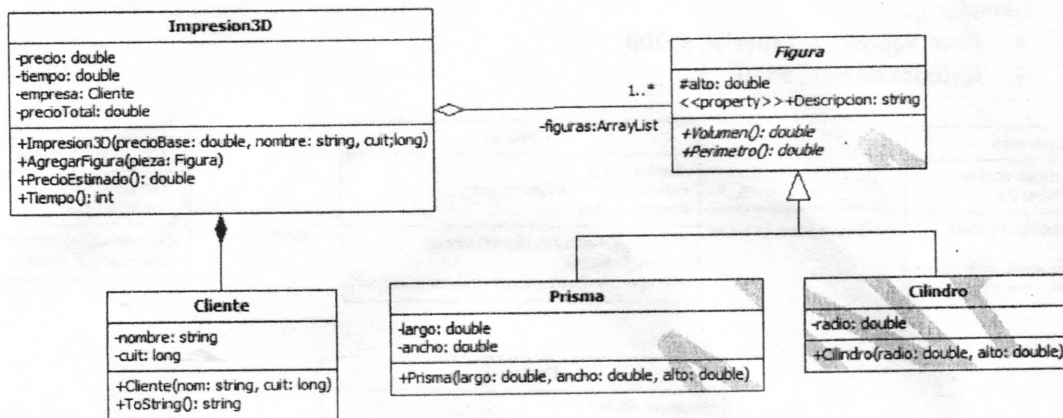


Parcial Programación II

Ejercicio 2

Una empresa de impresiones 3d necesita una aplicación para calcular costos y tiempos de producción. Para ello se diseñó el siguiente modelo:



- El tiempo se mide en segundos y sale del siguiente cálculo: **Perímetro x 1,75**
- El costo de la figura se calcula según: **Volumen*0,0035 del precio**
- El crear una figura, se indica en la Descripción (Prisma de [alto] o Cilindro de [alto])

Fórmulas a utilizar:

	Perímetro	Volumen
Cilindro	$(2 \times \pi \times \text{radio}) \times \text{alto}$	$\pi \times (\text{radio})^2 \times \text{alto}$
Prisma	$(2 \times \text{largo} + 2 \times \text{ancho}) \times \text{alto}$	$\text{largo} \times \text{ancho} \times \text{alto}$

Actividades:

1. Implemente las clases propuestas en el diagrama UML. 30pts
2. Codifique el botón Crear Impresión para tomar los valores del precio y cliente, creando un objeto Impresion3D. 10pts
3. Codifique el botón Agregar Figura para que abra la ventana secundaria Form2 y permita agregar un círculo o un rectángulo. 20ps
4. Implemente la interfaz IComparable para poder visualizar la lista de figuras (piezas) ordenadas por Descripción. 20ps
5. Codifique el botón Calcular para mostrar en el listBox los datos del cliente (utilice ToString), la lista ordenada de figuras (utilice la

Descripción) y el tiempo y precio total de la impresión 20pts

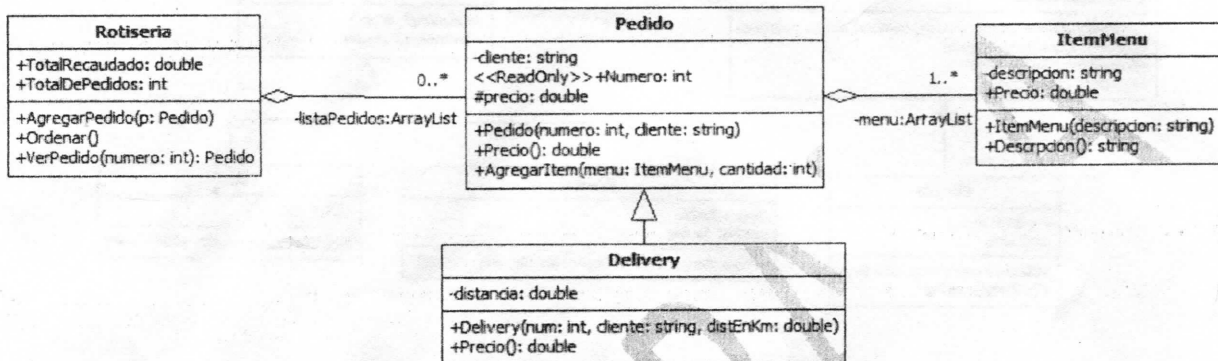
Parcial Programación II

Ejercicio 1

Una rotisería requiere de una aplicación para el registro de sus pedidos. Dispone de una lista de ítems de menú donde cada elemento se identifica por su descripción y precio.

Ejemplo:

- Pizza Napolitana Especial, \$1200
- Gaseosa de litro, \$850



Los pedidos pueden ser para retirar en el local, o por delivery. En caso de retirarlo en el local, el precio se calcula mediante la suma de los precios individuales (si supera los \$2000 tiene un descuento de 5%)

En caso de delivery no tiene descuento y se incrementa el precio con un valor fijo de \$150 más 0,8% por cada kilómetro de distancia a recorrer en el envío.

Ej: pedido \$3500 a entregar en una distancia de 3,5 Km: $3500 + 150 + (3500 * 3,5 * 0,08)$

El Form presenta un combobox1 con la lista de ItemMenu ordenados según su posición en un ArrayList llamado MenusActivos.

Actividades:

The screenshot shows the main window of the application, titled "Rotiseria". It contains a "Tomar pedido" button at the top. Below it is a dropdown menu for selecting a menu item. To the right of the dropdown is an "Agregar Menu" button. Below these is a text input field labeled "Cantidad:" followed by a "Borrar" button. Below the input field is a list box labeled "IBpedido". At the bottom of the window is a "Cerrar Pedido" button and a list box labeled "IBListaPedidos".

1. Implemente las clases propuestas en el diagrama UML. (El formulario puede ser utilizado como clase Rotisería). 30pts
2. Codifique el botón Tomar pedido que permita abrir la ventana secundaria e iniciar un nuevo pedido. 20pts
3. Codifique el botón Agregar Menú para cargar un ítemMenu y la cantidad de este al pedido, mostrando la descripción del itemMenu en el listBox IBpedido. 20pts
4. Codifique el botón cerrar pedido que permita informar el precio final en un messageBox, borrar el IBpedido y agregar el número de pedido y nombre de cliente en el listBox IBListaPedidos. 15pts
5. El método Ordenar de la clase Rotisería debe implementar el método QuickSort para ordenar los pedidos según su precio. 15pts

The screenshot shows a secondary window titled "Tomar pedido". It contains a text input field for "Nombre del cliente:" followed by a "Borrar" button. Below this is a checkbox labeled "Con envío?". To the right of the checkbox is a text input field for "Distancia en Km:" followed by a "Borrar" button. At the bottom of the window are "Aceptar" and "Cancelar" buttons.