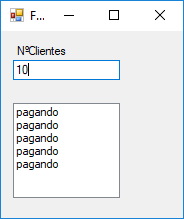
Vamos a simular un supermercado que tiene 5 cajas.

En ese supermercado van entrando clientes, cada cliente que entra hace la compra hasta que en un momento dado termina y va a pagar.

En el formulario principal visualizaremos cuántos clientes están comprando cuántos están pagando y además se creará un semáforo (“semaforocajas”) que controlará que no haya más de 5 clientes pagando al mismo tiempo.

El aspecto será el siguiente:



De manera que cada vez que un cliente entra a pagar, agrega una línea “pagando” a la listview y cada vez que termina de pagar quitará un “pagando” de la listbox.

El semáforo se creará en el formLoad, y cada vez que entra un cliente en el supermercado se recibirá un mensaje “WM\_ENTRAR”, cada vez que entra a pagar se recibirá un mensaje “WM\_PAGAR” y cada vez que genere la nota recibirá un “WM\_SALIR”.

Cuando se recibe “WM\_ENTRAR” contabiliza un cliente en Nº Clientes, cuando se recibe un “WM\_SALIR” resta un cliente en número de cliente y elimina un pagando de la listbox y cuando el cliente consigue caja para pagar agrega un pagando a la listbox.

# Los procesos cliente.

Cada proceso cliente tendrá una Lista de objetos de clase Detalle que crearemos a continuación.

## Clase Detalle.

Contiene los miembros Id y unidades, un constructor para recibir los valores para dichos miembros y los getter y setter correspondientes.

## Funcionamiento.

El proceso de cliente consistirá en una aplicación de consola que dará los siguientes pasos:

## Comprar

El cliente envía el mensaje WM\_ENTRAR al Formulario.

1. Generará un id de producto al azar entre 0 y el número de productos existentes (Los productos se almacenan en una tabla de una base de datos que describiremos más adelante).
2. Generará un número de unidades también al azar.
3. Con los dos números creará un objeto detalle y agregará el objeto a la lista.

El programa preguntará si se quiere comprar o no y la respuesta se generará al azar. Si la respuesta es afirmativa volverá al apartado 1) y si no pasará al apartado de Pagar. Para que haya más posibilidades de repetir que de terminar, se puede hacer, por ejemplo, que si el número generado al azar es 10 salga y si no repita o que genere códigos de caracteres y termine si se saca ‘n’ o ‘n’.

Meter retardos en cada vuelta para que puedan apreciarse los bloqueos y la ejecución concurrente.

## Pagar.

Solo puede haber 5 personas pagando a la vez. Aquí entra en juego el semáforo del formulario.

Si el proceso no se queda bloqueado el cliente envía el mensaje WM\_PAGANDO al formulario.

Pagar consiste en que para cada detalle de la lista de Detalles:

* Consulte la información sobre descripción y precio de la tabla de productos de la base de datos mencionada con anterioridad.
* Genere en un fichero de salida, y también en consola una línea de detalle como se muestra más adelante.

Cuando ha mostrado todas las líneas calculará el total, el IVA (21%) Y el total a pagar. En definitiva, mostrará una factura (tanto en pantalla, como en un fichero de texto) como la siguiente.

Producto unidades precio Importe

Pan 2 0.95 1.90

Leche 1 1.25 1.25

Total sin IVA: 3.15€

Total Pagar: 3.81€

Tras generarlo enviará el mensaje WM\_SALIR al formulario.

Meter retardos por cada vuelta para que se aprecien los bloqueos y la concurrencia.

**Nota:**

La tabla productos (la creáis en vuestra base de datos) tiene los siguientes campos, elegir los tipos.

**IdProducto:** un id autonumérico.

**Descripción:** el nombre del producto.

**Precio:** El precio unitario del producto.

Para probar la aplicación habrá que ejecutar el formulario e ir ejecutando los procesos clientes.