

# GliottiCom

## TP 3

Mi Trabajo Práctico consiste en un sistema ABM de clientes para la “empresa” de Telefonía GliottiCom, con la particularidad de que además de vender un Plan de dicha compañía, también tiene convenio con las demás compañías para poder vender sus servicios.

En cuanto a los temas aplicados de la clase esta incorporado:

- **Excepciones:** Hice cuatro clases de excepciones propias. La ClienteException que será dada en el FrmAltaCliente (cuando se ingresa un dni existente), FrmModificacionCliente y FrmBajaCliente  
La DniException se lanzará en las mismas tres clases Form en el momento en el que se ingrese un dni que no cumpla con los requisitos (no contener espacios, ser nulo, contener letras, contener menos o más de ocho dígitos).  
La GuardarException se lanzará en las mismas tres clases Form cuando no pueda Serializar.  
La NumeroCelularException se lanzará cuando el número de celular ingresado no cumpla con los requisitos (no contener espacios, ser nulo, contener letras, contener menos o más de 10 dígitos).
- **Test Unitarios:** Pruebo instanciar un cliente con datos erróneos para corroborar que se lancen las excepciones, luego en hacerlo con datos correctos para comprobar que se instancie correctamente y luego para comprobar la sobrecarga del operador + y - para corroborar que me agregue o quite un cliente de la lista, pruebo haciendolo agregando uno que ya exista, que eso me va a lanzar un ClienteException y luego un agregado normal.
- **Genericos:** Lo utilizo en la clase GestorArchivo, para poder Guardar/Serializar un archivo con cualquier tipo de dato, en este caso en concreto una lista de clientes.
- **Interfaces:** La utilizo simplemente en las tres clases Form mencionadas para recibir un cliente, appendear su información y agregarle la fecha en la que se haya hecho cada acción (ABM) y generar un ticket y acá mismo implemento **Archivos**.
- **Archivos:** Genero un archivo txt después de realizar ABM para que me genere un ticket con la información del cliente y la fecha en la que se realizó determinado movimiento.
- **Serialización:** Después de cada ABM me serializo en XML la lista de Clientes.

Algo a comentar: testeando la sobrecarga del operador +, pasa las pruebas, pero a la hora de utilizarlo por algún motivo que desconozco y no encuentro solución, no funciona, el problema que detecte viene de la lista que recibo por parámetro, no entra al foreach, estando vacia, siendo que la lista en teoría debería estar cargada. Haciendo un debug veo que está vacia, pero no le encuentro explicación, por eso en el código va a estar comentada una línea if (clientes + clienteNuevo). Las comparaciones tampoco son el problema.

Segundo tema a comentar: Los tickets se generan en la ruta Tp3\GliottiCom\bin\Debug\net5.0-windows\HistorialCliente. No agregue un ver archivo, ya que en mi lógica de negocio es un ticket que sirve de recibo para el cliente, por lo que no necesito verlo.

## **TP 4**

En esta etapa del Trabajo pPactico, lo que agregué fue un botón llamar, que lo que hace es traer números de teléfonos aleatorios de la base de datos, que son números de NO-Clientes y los guardo en una lista para simular una llamada con ese mismo, al ser una llamada a un número aleatorio, genero un random de 0 a la cantidad total que hay en la lista, de esta forma, ese random me devuelve un índice aleatorio para traer ese número. A todo esto, creo un hilo secundario donde a través de un evento y un delegado imprimo por pantalla el tiempo de transcurso de la llamada.

En el supuesto caso que el cliente acepte lo que ofrecemos, vamos a darle de alta, al momento de darle de alta, mando la query para que se elimine el número del que ahora ES cliente, de la base de datos de NO-Clientes para que no vuelva a ser llamado. En caso de que “no quiera”, simplemente finalizamos la llamada con el botón cortar y eso cancela el hilo y cierra el formulario (no lo borra de la base de datos de No-Clientes, así de esta manera puede volver a ser llamado). Y por último, el botón copiar del Formulario de llamada, lo único que hace es copiar el número de teléfono del label para poder dar de alta fácilmente sin tener que tipear el número.

Así que de esta forma quedan aplicados los últimos temas:

- **SQL:** Para traer leer o borrar datos (en este caso números de teléfono) de la base de datos.
- **Delegados/Hilos/Eventos:** Para generar la simulación de llamada.
- **Método de extensión:** Para devolver la cantidad total de la lista de clientes a través del método de extensión del tipo int.

**Algo a comentar:** Quedó solucionado la sobrecarga del + y -, ya que genere otra sobrecarga del == / != donde compare con una lista, este error que tenia en el anterior trabajo practico, se daba a que si recorría una lista vacía, nunca iba a entrar, por lo que jamás se llegaba a preguntar si el cliente existe en determinada lista. Por eso quedaron dos sobrecargas del == y del !=.

También están controladas las excepciones marcadas como error en el anterior trabajo practico. Dicho esto, dejo confirmado la corrección del anterior trabajo practico sumándole los temas nuevos del nuevo trabajo practico.