

## TP N.º 4 - Laboratorio de computación II.

El trabajo presentado funciona como boceto a un posible programa para controlar los datos de una colonia de vacaciones. Me he inclinado por el mismo ya que durante las temporadas de verano, coordino una colonia y me sería necesario continuar el mismo, probarlo, testearlo y finalizarlo para utilizarlo durante la próxima temporada 2021. En el programa presentado, existen partes harcodeadas ya que realizarlo me llevó más tiempo del que esperaba. Continuaré el mismo para tenerlo listo antes del 15 de diciembre 2020 y darle el 100% de utilidad a este material didáctico.

### **Contenidos implementados:**

#### **EXCEPTION**

Un ejemplo de manejo de excepciones se desarrolla en la clase Grupo en la siguiente firma:

```
namespace Entidades
{
    public class Grupo
    {
        public static Grupo operator +(Grupo g1, Colono c1)
        {
            ///Codigo...
        }
    }
}
```

Durante la misma se intenta agregar un colono a un grupo. En caso de que el niño ya se encuentre en la colonia, lanzará una excepción [ColonoRepetidoException](#).

#### **TEST UNITARIOS**

Se implementan para:

- Testear que si se intenta agregar un alumno repetido, este sea rechazado ya que su dni es igual al alumno existente en la colonia.
- Testear que si se intenta agregar un colono a la colonia, esta actualice su lista de colonos conteniendo al alumno ingresado.

#### **TIPOS GENÉRICOS**

Ha sido utilizado como clase contenedora de dos tipos diferentes de productos a vender: “Gorritas” y “Antiparras”. Se implementa en el namespace STOCK en la clase ControlStock. Esta clase contiene una List<T> que puede ser cargada por cualquiera de los productos vendidos.

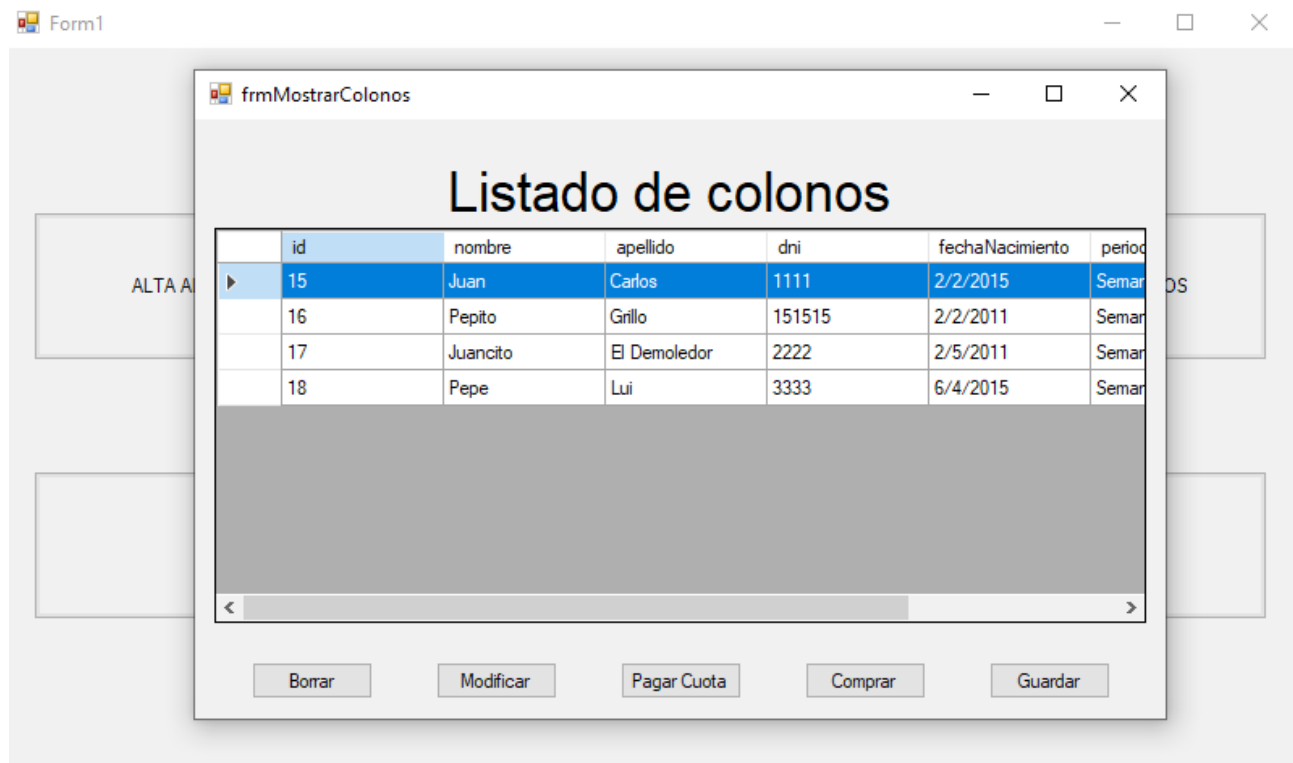
```
namespace Stock
{
    public class ControlStock<T>
    {
        ///Codigo...
    }
}
```

## INTERFACES

Se encuentra desarrollada dentro del namespace Entidades dentro de la interfaz “Icomprar”. Es la encargada de desarrollar las ventas de los productos que la colonia tiene en stock.

```
namespace Entidades
{
    public interface IComprar
    {
        void RealizaVenta(Colonia colonia, Producto p1, Colono c1);
    }
}
```

Mediante el método implementado, se elimina el producto de la lista de posibles ventas y se carga el importe del producto a las deudas del colono. Mediante un botón “Comprar” del frmMostrarColonos, se conseguirá realizar lo anteriormente mencionado.



The screenshot shows a Windows application window titled "frmMostrarColonos". Inside the window, there is a title "Listado de colonos" above a table. The table has columns: id, nombre, apellido, dni, fechaNacimiento, and periodo. The first row is highlighted in blue. Below the table, there are five buttons: "Borrar", "Modificar", "Pagar Cuota", "Comprar", and "Guardar".

|   | id | nombre   | apellido     | dni    | fechaNacimiento | periodo |
|---|----|----------|--------------|--------|-----------------|---------|
| ▶ | 15 | Juan     | Carlos       | 1111   | 2/2/2015        | Semar   |
|   | 16 | Pepito   | Grillo       | 151515 | 2/2/2011        | Semar   |
|   | 17 | Juancito | El Demoledor | 2222   | 2/5/2011        | Semar   |
|   | 18 | Pepe     | Lui          | 3333   | 6/4/2015        | Semar   |

Ejemplo comprar.

## ARCHIVOS O SERIALIZACION XML.

Se ha elegido la serializacion Xml como la adecuada para lograr guardar los datos de la colonia en el escritorio del cliente.

El método Serializacion se encuentra dentro de la clase Colonia.

```

namespace Entidades
{
    public class Colonia : IComprar
    {
        public bool SerializacionXml()
        {
            ///Codigo...
        }
    }
}

```

## **Bases de datos / SQL**

A fines didácticos solamente se ha desarrollado la carga y el guardado de los Colonos en DB. El alumno que se dá de alta, pasa directamente a la DB. La misma es mostrada en un datagridview para visualizar todos los colonos dentro de una instancia del objeto colonia. Los profesores y algunos productos de venta, tales como “Gorritos” y “Antiparras” son hardcodeados y viven en memoria. Por lo tanto si se desarrolla el “Pago de la cuota” del colono (desde el formulario frmMostrarColonos) habrá que guardar los cambios para que si se cierra el formulario, siga ese colono con el saldo en cero, sino, al traer los datos desde el objeto actual de colonia, se perderán los pagos realizados. Esta es una de las tantas modificaciones a realizar antes de finalizar el proyecto Colonia.

## **THREADS**

Ha de utilizarse para modificar el aspecto visual del formulario apenas arranca el formulario principal.

## **EVENTOS**

```

namespace Formularios
{
    public partial class frmAltaColono : Form
    {
        private void bntAceptar_Click(object sender, EventArgs e)
        {
            ///codigo...
        }
    }
}

```

Su desarrollo se utiliza para cargar en la base de datos cada colono cargado en DataGridView.

## **MÉTODO DE EXTENSIÓN**

```
namespace Entidades
{
    public static class ExtensionColonoPagaCuota
    {
        public static double PagarCuota(this Colono c1, Colono colono, Colonia colonia)
        {
            ///Codigo...
        }
    }
}
```

El mismo realiza una extensión de la Clase Colono. Ha sido implementado para abonar la cuota (y elementos que compre el colono) bajando su deudo a cero y aumentando el saldo de la colonia.

## **FUNCIONALIDADES PARA UN SISTEMA DE VENTA**

Durante el trabajo práctico se ha codificado el alta de productos principales: Colonos y Profesores (algunos de estos se encuentran hordcodeados ya que no están alojados en base de datos). Cada Colono que ingresa a la colonia va a pertenecer a un grupo específico según su edad:

- Entre 3 y 6 años : “Peques”.
- Entre 7 y 10 años: “Medianos”
- Entre 11 y 13 años “Grandes”.

Si no existe ningun colono en la colonia, el primer grupo que se creará será el de el colono que primero ingrese. Si el segundo colono que ingresa, pertenece al grupo del primer colono, este se agregará a su grupo, sino se creará uno nuevo, etc. También cada alumno pagará una cuota según el período en el que se anote a la colonia (este se encuentra hardcodeado al momento de crearse, de modo que siempre un nuevo colono pertenecerá al periodo “mes”).

Además se ha creado otro sistema de venta en el que efectivamente se pueden vender “Gorritas” y “Antiparras”.