

## **TRABAJO PRACTICO N°3**

**(AL FINAL EL AGREGADO DEL TP N°4)**

**FEDERICO LEONEL MARTINEZ 2°A**

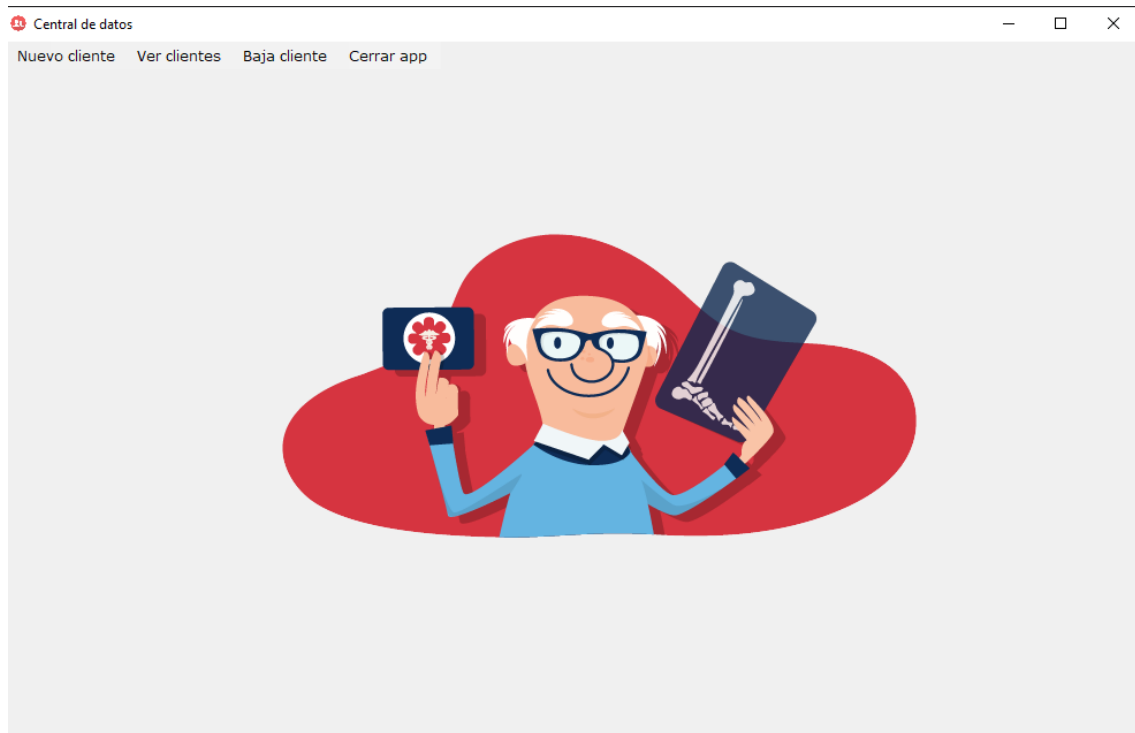
**Aplicación desarrollada para manejar un listado de clientes de una obra social**

- **AGREGAR**
- **ELIMINAR**
- **VER LISTADO**

**Se podrá agregar una cantidad ilimitada de clientes, todos los campos son obligatorios, y el DNI no se debe repetir ya que no se permitirá el alta.**

**El programa vendrá con una lista de clientes cargados, estos serán leídos de un archivo tipo XML, están también en un JSON.**

**Al cargar un nuevo cliente y al dar de baja uno existente, automáticamente se carga la modificación en los archivos XML y JSON.**



### Excepciones:

- **ArchivoNoEncontradoExcepcion:** Se dispara cuando no encuentra el archivo
- **CampoInvalidoExcepcion:** Se dispara cuando al dar de alta un cliente se ingresó mal un campo
- **DniExistenteExcepcion:** Se dispara cuando se intenta dar de alta un cliente con un DNI ya utilizado
- **NoHayClientesExcepcion:** Se dispara cuando se intenta dar de baja o ver la lista de clientes y la misma se encuentra vacía

### Pruebas Unitarias: Testing de 2 métodos de la clase CentralDatos

- **ValidarCantidadClientes\_DeberiaRetornarTrue\_CuandoHayClientes()**
- **ValidarCantidadClientes\_DeberiaRetornarFalse\_CuandoNoHayClientes()**
- **BajaCliente\_DeberiaEliminarClienteYRetornarTrue()**

### Tipos genéricos:

- Serializacion<T>
- SerializarJSON<T>
- IArchivos<T>

Donde T puede ser cualquier tipo, en este caso se usa como una lista de clientes

### Interfaces:

- IArchivos<T>: Declaramos los métodos LeerLista() y EscribirLista() que serán declarados por las clases Serializacion y SerializarJSON

### Archivos:

El programa viene con 2 archivos, uno de cada uno, donde en los mismos se encuentra la lista cargada de clientes.

Al agregar y dar de baja un cliente, los archivos se actualizan automáticamente.

## TRABAJO PRACTICO N°4

### Correcciones TP3:

- DNI negativo ya no se admite
- se implementó archivos(txt)

Se debe primero ejecutar el script adjuntado para crear la base de datos y llenarla con clientes

### **SQL:**

- clase Cliente SQL.
- El programa trae los clientes desde la base de datos previamente agregados con el script
- Tiene las funcionalidades de; Agregar, Eliminar y Leer

### **Delegados y Eventos:**

- Utilizo delegado para el evento que se encarga de simular la carga al principio del programa.
- Eventos para simular carga del programa, y habilitar los botones cuando el anterior finalice

### **Hilos/Tasks:**

- Se usa al momento de traer los clientes de la base de datos y escribirlos en listas. Clase MiEvento.