

# Trabajo Práctico N°4

## TP Final

*Apellido: Tejeda*

*Nombre: Roberto*

*División: 2C*

### **ATENCIÓN:**

**PARA QUE LA APLICACIÓN FUNCIONE DEBE PEGAR LA CARPETA DENOMINADA “TP4”, QUE ESTA EN LA RAÍZ DEL REPOSITORIO, EN EL ESCRITORIO DE SU COMPUTADORA.**

**La base de datos se crea con el script que está en la raíz de la carpeta “TP4” denominado “SQL-TEJEDA2C-TPF”. En el anexo, al final del presente documento, se encuentra el script extendido.**

Esta se trata de una aplicación pensada para quien trabaja en la empresa Direct TV (DTV), en el sector de “ATENCIÓN AL CLIENTE”, por lo tanto, se inicia solicitando el Loguin del Usuario de la aplicación el cual se puede hacer desde una base de datos SQL o un archivo XML.

En adelante usaremos la siguiente nomenclatura:

**Usuario:** Es el trabajador de DTV con nombre de usuario y password habilitado para iniciar la aplicación, viene codeado para ingresar con el usuario “rtejeda” y la password “1234”. Otros usuarios habilitados son 'atejeda' password '5678' y 'ntejeda' password '9012'.

Esta aplicación no permite el agregado de Usuarios ya que esa es responsabilidad de otra instancia.

**Cliente:** Es la persona, titular del servicio de DTV, que se contacta para realizar alguna gestión sobre su cuenta. El numero de cliente es su DNI.

### **PERSISTENCIA DE DATOS**

La persistencia de los datos está dada por los archivos “clientesDTV.xml”, “usuariosDTV.xml” y la base de datos [TEJEDA2C-TPF] que es creada con el script provisto. Toda la información se encuentra en la raíz del proyecto, en la carpeta denominada “TP4”.

Al ser una aplicación pensada para usuarios de atención al cliente los mismos no pueden ser creados, borrado ni editados ya que esa tarea forma parte de otra instancia de la aplicación.

### **LÓGICA de uso**

La aplicación inicia en el formulario donde el Usuario debe colocar nombre de usuario y password.

Una vez el usuario se identifica satisfactoriamente se accede al formulario de identificación de cliente que se realiza a través de su DNI. En esta instancia podemos seleccionar al cliente o darlo de alta. Además, podemos salir de la aplicación.

El formulario de alta ingresa al sistema solo los datos del Cliente.

Si seleccionamos el Cliente accedemos al formulario de sus servicios donde una vez realizados los cambios podemos guardar la nueva situación del cliente.

La Solución "Tejeda.Roberto.2C.TPFinal" cuenta con los siguientes proyectos:

- Formularios: Todo los formulario Windows
  - FormUsuario : Es el formulario de inicio, el proyecto lleva por default el nombre de usuario y clave válidos para iniciar sesión.
  - FormValidarCliente : Es el formulario que se instancia una vez que el usuario se conecta satisfactoriamente, el mismo lleva por default un Cliente valido.
  - FormAltaCliente: Es el formulario que se instancia para dar de alta un nuevo cliente, en esta instancia solo se hace con los datos personales.
  - FormServiciosCliente. Es el formulario que se instancia para modificar los servicios del Cliente (esta incompleto)
- Entidades es una biblioteca de clase con las siguientes entidades:
  - Persona
  - UsuariosDTV : Hereda de Persona y es el trabajador de atención al cliente.
  - ClienteDTV: Hereda de Persona, es el destinatario del servicio y quien se contacta con la empresa solicitando Alta, Baja o Modificación del servicio.
  - UsuarioDTVDAOs: Manejo de base de datos SQL.
  - Servicio: Contiene todos los atributos propios del negocio.
- 
- PersistirDatos : Es una biblioteca de clases que contiene las clases necesaria para el manejo de archivos.
  - IResguardarDatos: Es un interface que asegura la existencia de los métodos mínimos para el guardo de información.
  - ManejarArchivos: Es una clase que se encarga de el manejo de los archivos, los path más presesamente
  - Serializar: Es la clase que se encarga de serializar y deserealizar los archivos tipo json y xml.
  - Logger; Es una clase que se encarga de crear archivos de texto
- Test Unitario: Implementa dos TEST
  - IdentificarCliente\_CuandoRecibeUnDNIQueEstaEnLaBaseDeDatos\_DebeRetornarClienteConMismoDni
  - IdentificarUsuario\_CuandoRecibeUnPasswordYNombreDeUsuarioQueEstaEnLaBaseDeDatos\_DebeRetornarTrue

### **GENÉRICOS e INTERFACES**

En la clase SerealizarXML del proyecto Serealizar

### **EXCEPCIONES**

En todo el manejo de archivos

### **SQL**

En el log de Usuario

### **SERIALIZACION XML**

Persistencia y lectura de datos de ClientesDTV.

### **EXPRESIONES LAMBDA**

Declaración de propiedades

### **CREACION DE ARCHIVOS TXT**

Logs de Clientes ya creados e intentados dar de alta nuevamente, "UsuarioExistente.txt" y los de Usuarios a la plataforma en "UsuariosLoguadosSQL.txt", "NuevoUsuario.txt" y "UsuariosLoguadosXML.txt"

### **TEST UNITARIO**

IdentificarCliente\_CuandoRecibeUnDNIQueEstaEnLaBaseDeDatos\_DebeRetornarCliente

IdentificarUsuario\_CuandoRecibeUnPasswordYNombreDeUsuarioQueEstaEnLaBaseDeDatos\_DebeRetornarTrue

# ANEXO

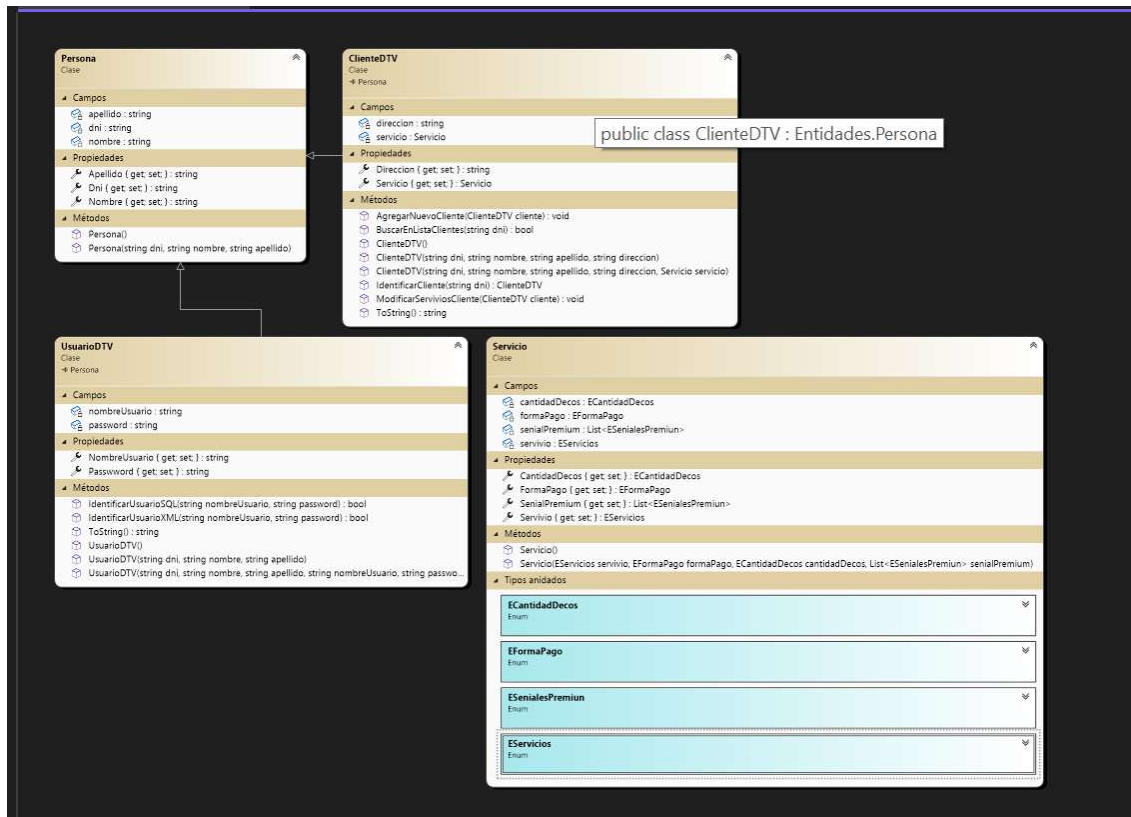
## A-

Script para la creación de la base de datos de Usuarios habilitados a operar la aplicación:

```
CREATE DATABASE [TEJEDA2C-TPF];
GO
USE [TEJEDA2C-TPF];
GO
CREATE TABLE [TEJEDA2C-TPF].dbo.Usuarios
(
    Dni INT PRIMARY KEY,
    Nombre VARCHAR(255) NOT NULL,
    Apellido VARCHAR(255) NOT NULL,
    NombreUsuario VARCHAR(255) NOT NULL,
    Password VARCHAR(10) NOT NULL,
);
GO
INSERT INTO Usuarios VALUES ('18223125', 'Roberto', 'Tejeda', 'rtejeda', '1234')
INSERT INTO Usuarios VALUES ('18456789', 'Alberto', 'Tejeda', 'atejeda', '5678')
INSERT INTO Usuarios VALUES ('18987654', 'Norberto', 'Tejeda', 'ntejeda', '9012')
```

## B-

### DIAGRAMAS DE CLASE ENTIDADES



C-

## DIAGRAMAS DE CLASE FORMULARIOS

