Modificación Patrón Singleton

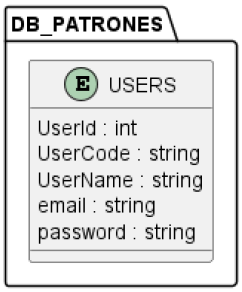
Objetivo

Crear aplicación que cargue la información de una base de datos. Si alguna propiedad es actualizada, debe reflejarse en la base de datos y al reiniciar el programa deberá leer el dato actualizado

Resolución

A continuación, se detallarán los pasos realizados para completar el objetivo de la tarea

i. Creación de la base de datos y tabla a ser utilizada



El proyecto se realizó en Punto Net (C#)

ii. Creación de la Entidad

```
class E_Users
{
    public int UserId { get; set; }
    public string UserCode { get; set; }
    public string UserName { get; set; }
    public string Email { get; set; }
    public string Password { get; set; }

}
```

iii. Creación del acceso a base de datos

Aquí es donde configuramos el patrón Singleton

```
public class HelperSQLServer
{
    private static HelperSQLServer HSQLServer { get; set; }
    private static SqlConnection oConnection { get; set; }

    private HelperSQLServer() { }

    public static HelperSQLServer getInstance() {
        if (HSQLServer == null) {
            HSQLServer = new HelperSQLServer();
        }
        return HSQLServer;
    }

    public SqlConnection getConnection() {
        string connectionString = "Data Source=192.168.1.37;Initial
Catalog=DB_PATRONES;User ID=sa; Password=******";

        if (oConnection == null) {
            oConnection = new SqlConnection(connectionString);
```

```
            oConnection.Open();
        }

        if(oConnection.State != System.Data.ConnectionState.Open) {
            while(true) {
                if(oConnection.State == System.Data.ConnectionState.Open) {
                    break; }
                if(oConnection.State == System.Data.ConnectionState.Broken) {
                    oConnection.Close(); oConnection.Open(); }
                if(oConnection.State == System.Data.ConnectionState.Closed) {
                    oConnection.Open(); }
            }
        }
        return oConnection;
    }
}
```

iv. Configuración de los DAO para el acceso (consulta/modificación) de la base de datos

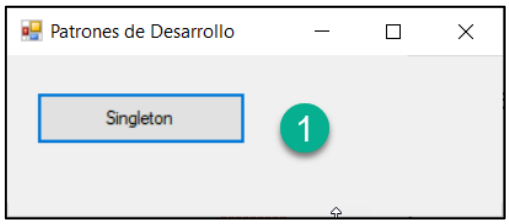
Aquí se utiliza la instancia del Singleton creado

```
class UsersDAO
{
    private SqlConnection oConnection;

    public UsersDAO() {
        oConnection = Singleton.HelperSQLServer.getInstance().getConnection();
    }

    public DataSet getListUsers() {
         DataSet oDataSet = new DataSet();
         var oAdapter = new SqlDataAdapter("SELECT * FROM USERS", oConnection);
         oAdapter.Fill(oDataSet, "USERS");
         oAdapter.Dispose();
         return oDataSet;
    }

    public int setUsers(Entities.E_Users user) {
         int iResp = 0;
         string SQL = $" UPDATE USERS " +
                      $"    SET UserName = '{user.UserName}', email =
'{user.Email}', [password] = '{user.Password}' " +
                      $" WHERE UserId = '{user.UserId}' ";
         var oCommand = new SqlCommand(SQL, oConnection);
         iResp = oCommand.ExecuteNonQuery();

         return iResp;
    }
}
```

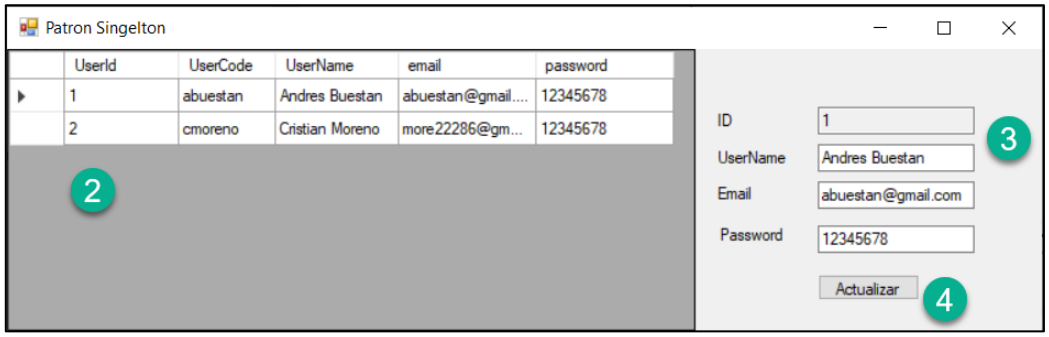
v. Creación de la capa lógica de negocio

```
class Users {
    DataAccess.UsersDAO UserDAO;
    public Users() {
        UserDAO = new DataAccess.UsersDAO();
    }
    public DataSet getListUsers()
    {
        return UserDAO.getListUsers();
    }
    public int setUsers(Entities.E_Users user) {
        return UserDAO.setUsers(user);
    }
}
```

vi. Creación del Front, para que el usuario interactúe



1. Abrir el formulario para el mostrar los datos por medio del patrón Singleton



2. En el Grid se recuperan los datos de la base de datos; y al dar doble clic sobre un registro se habilitará la opción de modificación (derecha del formulario)
3. Modificación de los datos
4. Enviar los datos a la base de datos