

BankApp

[Tecnologías usadas](#)

[Implementación](#)

[.Net 6.0](#)

[Base de datos](#)

[Preparación](#)

[Restaurar la base de datos](#)

[Crear la base de datos mediante scripts](#)

[Código](#)

[Descripción del programa](#)

[Pantalla - Control](#)

[Pantalla - Customer](#)

[Parte superior - Búsqueda de cliente](#)

[Parte intermedia - Información del cliente](#)

[Parte inferior - Acciones sobre un cliente](#)

[Pantalla - Bank Account](#)

Tecnologías usadas

- .Net 6.0
- C#
- WPF (Windows Presentation Foundation)
- SQL server 2019
- T-SQL

Implementación

.Net 6.0

Este proyecto fue creado en .Net 6.0 por lo que es importante tener descargada e instalada la última versión que se puede encontrar de forma oficial en este [enlace](#).

Base de datos

Preparación

Los datos de la aplicación provienen de una base de datos alojada en SQL Server 2019, que se puede descargar desde este [enlace](#).

Una vez descargado es necesario instalarlo, para este proyecto se han utilizado las opciones por defecto que ofrece el asistente de instalación salvo en el apartado **“Configuración del Motor de base de datos- Configuración del servidor”**, en el cual se ha seleccionado el modo **“Autenticación de modo mixto”** lo que permite usar el usuario “sa”, usado en el proyecto para mayor comodidad.

Es importante recordar los datos introducidos y la contraseña asignada al usuario “sa”, pues serán usados en la cadena de conexión del programa.

Es recomendable seguir la [guía de instalación oficial de SQL Server](#) de Microsoft.

Tras instalar SQL Server 2019, debemos descargar e instalar [SQL Server Management Studio](#), que nos permitirá restaurar la copia de seguridad de la base de datos de ejemplo (archivo con extensión .bak), o usar los scripts proporcionados para crear la estructura de la base de datos y rellenarla con los datos de ejemplo (archivos con extensión .sql).

Restaurar la base de datos

Si elegimos esta opción, debemos abrir **SQL Server Management Studio** e iniciar sesión con el usuario “sa”.

Para restaurar la base de datos, seguiremos los pasos indicados en la [guía oficial de Microsoft](#).

Podremos encontrar el archivo con extensión .bak en la carpeta Script.

Crear la base de datos mediante scripts

La alternativa al método anterior es usar los scripts con extensión .sql que se encuentran en la carpeta Scripts.

Para ejecutar los scripts, abrimos **SQL Server Management Studio** e iniciar sesión con el usuario “sa”, tras ello pulsamos File > Open > File ... > Seleccionamos el archivo “create_structure.sql” y pulsamos “Execute”.

Tras ello, realizamos la misma operación con el script “fill_tables.sql”.

Código

Para asegurar la conexión con la base de datos debemos modificar la cadena de conexión en el archivo “App.Config” y reemplazar los datos por los introducidos cuando se instaló la base de datos.

Es importante que la cadena de conexión se siga llamando “bankapp”.

Una vez realizado este cambio ya es posible compilar el código, e implementar el programa en el equipo del cliente.

Descripción del programa

El programa está formado por 3 pantallas, cada pantalla a su vez está formada por una vista (archivo con extensión .xaml) y su correspondiente clase (archivo con extensión .cs) que controla el flujo de información y controla las acciones que realizan los botones.

Estas 3 pantallas son:

Control: Una pantalla de login que permite al cliente identificarse con su usuario y contraseña.

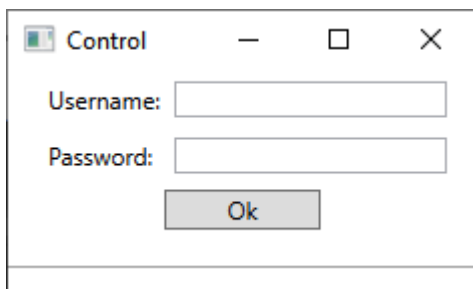
Customer: Pantalla principal que permite crear, leer, actualizar y borrar clientes, así como acceder a la pantalla Bank Account.

Bank Account: Muestra las cuentas bancarias del cliente seleccionado en la pantalla *Customer*.

Pantalla - Control

Esta pantalla está formada por dos label, con un TextBox para el campo del nombre de usuario y un PassBox para la contraseña.

También se incluye un control de tipo Button, que al ser presionado comprobará si la combinación introducida existe en base de datos, devolviendo un error de no ser así, o una confirmación y abriendo la pantalla **Customer si son correctos**.

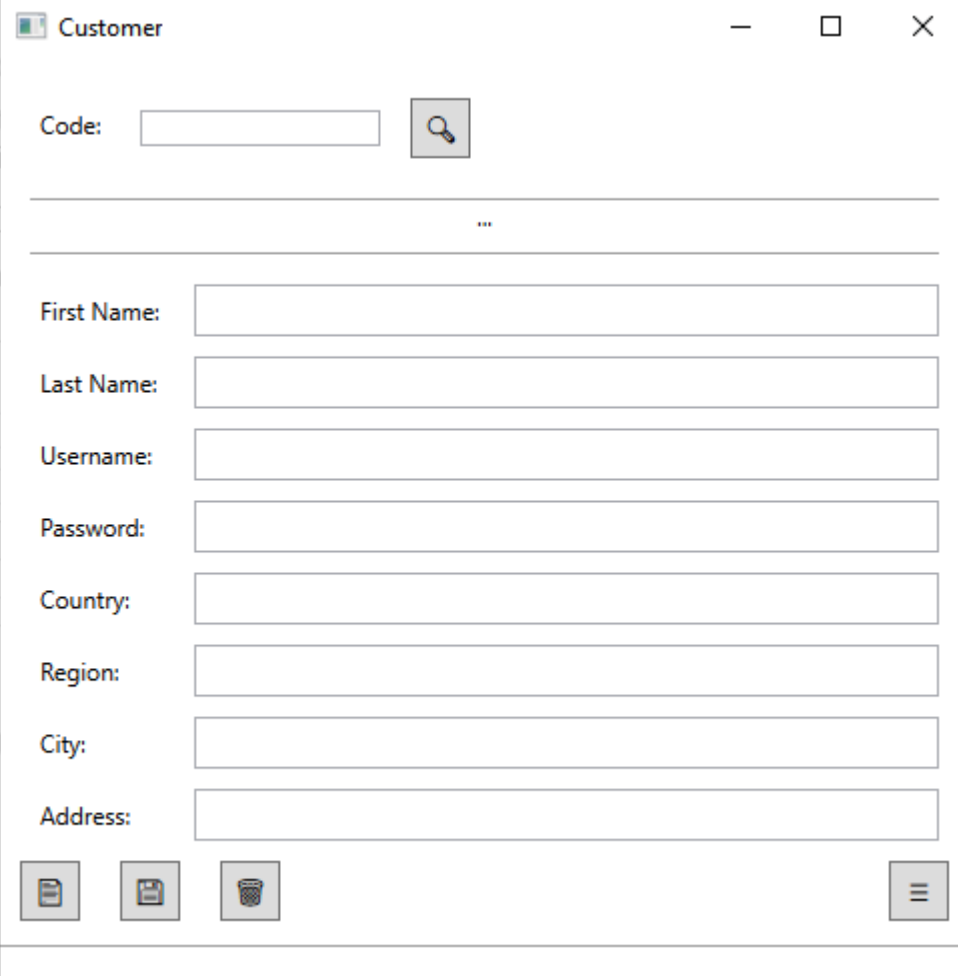


The screenshot shows a Windows-style window titled 'Control'. It contains two labels, 'Username:' and 'Password:', each followed by a text input field. Below these fields is a single button labeled 'Ok'. The window has standard minimize, maximize, and close buttons in the title bar.

Pantalla - Customer

Esta pantalla está dividida en 3 partes:

- La parte superior contiene la búsqueda del cliente.
- La parte intermedia contiene los campos que conforman un cliente.
- La parte inferior con iconos que permiten realizar acciones sobre un cliente.



The screenshot shows a window titled "Customer" with standard Windows window controls (minimize, maximize, close). The form is divided into three sections:

- Top Section:** A "Code:" label followed by a text input field and a magnifying glass icon button.
- Separator:** A horizontal line with three asterisks (***) in the center.
- Middle Section:** A vertical stack of labels and text input fields: "First Name:", "Last Name:", "Username:", "Password:", "Country:", "Region:", "City:", and "Address:".
- Bottom Section:** A horizontal row of four icons: a floppy disk (save), a document with a magnifying glass (search), a trash can (delete), and a hamburger menu icon.

Parte superior - Búsqueda de cliente

Esta parte está formada por una Label y una Textbox que permite al usuario introducir un código de cliente.

Junto a ello aparece un control Button, que al ser presionado buscará el código de cliente en la tabla Customer de la base de datos, de existir un cliente con ese ID, devuelve un MessageBox informando al usuario de que se ha cargado correctamente el cliente, y se rellenan los campos de la parte intermedia con su información.

De no encontrarse un cliente con ese ID, aparecerá un MessageBox informando al usuario de ello y se limpiarán los campos de la parte intermedia.

Parte intermedia - Información del cliente

En esta parte se muestra la información que conforman a un cliente.

Puede ser cubierta manualmente para crear un cliente, cubierta automáticamente al buscar un cliente, o modificada tras buscar un cliente.

Cada parte de la información del cliente está formada por una Label y un Textbox, salvo el campo "Password", que está formado por una Label y un PasswordBox, evitando así que se vea la contraseña en claro.

Parte inferior - Acciones sobre un cliente

Esta parte está dedicada a lanzar acciones sobre un cliente.

Para ello está formada de 4 controles Button, partiendo de la izquierda a la derecha son:

- **Nuevo usuario:** Toma los datos de los campos de la parte intermedia, realiza unas comprobaciones básicas en cada uno de los campos, de cumplirse se crea un nuevo cliente en la tabla **Customer** de la base de datos.

Las comprobaciones son las siguientes:

- Campos "First Name", "Last Name": Longitud mínima 2, longitud máxima 50 y no puede contener dígitos.
- Campo "Username": Longitud mínima 3 y longitud máxima 50.
- Campo "Password": Longitud mínima 4, longitud máxima 100 y debe contener al menos un dígito.
- Campos "Country", "Region", "City": Longitud mínima 1, longitud máxima 100 y no puede contener dígitos.
- Campo "Address": Longitud mínima 1 y longitud máxima 250.

De no cumplirse alguno de los requisitos, se le mostrará al usuario un MessageBox con los requisitos del campo que provocó el error.

Tras crear el usuario se limpian los campos de la parte intermedia.

- **Guardar cambios:** Comprueba que se haya cargado un usuario previamente, de no haberlo hecho muestra un MessageBox informando de ello al usuario.

De tener un cliente cargado, se toman los datos de los campos de la parte intermedia y se realizan las mismas comprobaciones que en el caso de crear un nuevo cliente.

De no cumplirse alguno de los requisitos, se le mostrará al usuario un MessageBox con los requisitos del campo que provocó el error.

De cumplirse todos los requisitos, se actualiza la información en base de datos si la fecha en la que se recuperó la información del cliente es inferior a cuando se actualizó por última vez su información en la tabla *Customer* de la base de datos, mostrando un MessageBox con el resultado de la operación para informar al usuario y limpiando los campos de la parte intermedia.

- **Eliminar usuario:** Comprueba que se haya cargado un usuario previamente, de no haberlo hecho muestra un MessageBox informando de ello al usuario.
De tener un cliente cargado, se intenta borrar de la tabla *Customer* de base de datos el cliente que tenga el mismo id que el cliente cargado, mostrando un MessageBox con el resultado de la operación para informar al usuario y limpiando los campos de la parte intermedia.

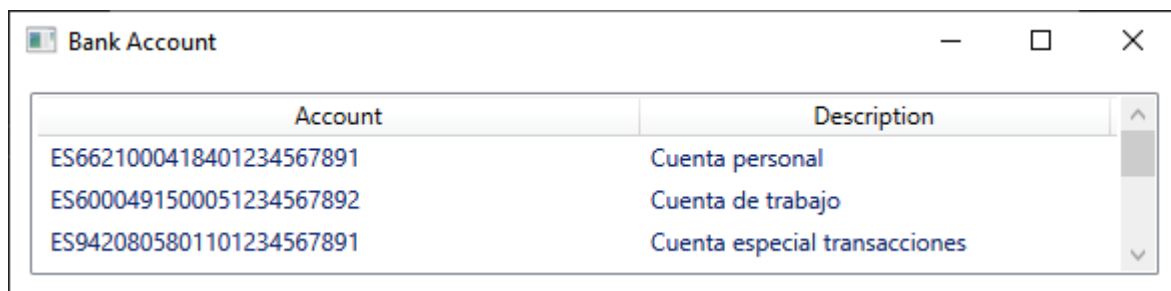
- **Mostrar cuentas:** Comprueba que se haya cargado un usuario previamente y si el cliente tiene cuentas que mostrar, de no haberlo hecho o no tener cuentas muestra un MessageBox informando de ello al usuario.
Abre la pantalla **Bank Account**.

Pantalla - Bank Account

Esta pantalla muestra las cuentas del cliente y su descripción.

Está formada por una ListView, con dos GridViewColumn para representar el número de cuenta y su descripción.

No es posible modificar los datos, tampoco es posible usar la pantalla "Customer" mientras esta ventana no se cierre.



Base de datos

La base de datos está formada por 2 tablas, una almacena la información que define al cliente (Customer) y la otra las cuentas bancarias y descripciones.

Ambas tablas están relacionadas para permitir el cruce de datos.

Tabla - Customer

Contiene toda la información que define a un cliente, como su nombre, apellido, etc.

A mayores incluye el campo id, que aumenta cada vez que se crea un cliente, y el campo last_update, que nos permitirá tener un control de cuando se modifica la información, y permitir concurrencia e impedir que los usuarios actualicen datos de un mismo cliente sin haber obtenido la última versión de los datos antes.

Estas son las características de las columnas en base de datos

- id: int NOT NULL Autoincremental Primary Key
- first_name: nvarchar(50)
- last_name: nvarchar(50)
- username: nvarchar(50)
- password: nvarchar(100)
- country: nvarchar(100)
- region: nvarchar(100)
- city: nvarchar(250)
- last_update: datetime

Tabla - Account

Contiene la información de las cuentas bancarias, su descripción y cómo relacionarla con los clientes.

La columna `customer_id` está relacionada con la columna `id` de la tabla *Customer* para permitir identificar su propietario.

Estas son las características de las columnas en base de datos

- `id`: int NOT NULL Autoincremental Primary Key
- `customer_id`: int NOT NULL Foreign key (Customer.id)
- `account_number`: nvarchar(50)
- `description`: nvarchar(max)

Diagrama

