



DAVID SERRANO
JOSE VIERA

INTERFAZ APLICACIÓN.

ÍNDICE.

1. Nombre de los integrantes del equipo.....	3
2. Descripción general de la solución aportada: Detallar en esta sección todas aquellas cuestiones importantes asumidas a la hora de resolver el proyecto.....	3
3. Implementación sobre C# de la aplicación:.....	3
3.1. Formularios creados:	3
o Nombre del formulario y descripción de las tareas que realiza.	
o Controles principales que contiene (Controles que contienen otros controles)	
o Descripción de los métodos principales.	
4. Consideraciones finales y conclusiones.....	12

1. Nombre de los integrantes del equipo.

El equipo lo componen los siguientes integrantes :

- José Viera
- David Serrano

2. Descripción general de la solución aportada: Detallar en esta sección todas aquellas cuestiones importantes asumidas a la hora de resolver el proyecto.

Una biblioteca desea organizar su actividad diaria de forma sencilla, organizando los usuarios, empleados, libros y préstamos de la misma.

De manera que, de los usuarios sepan sus datos personales y ver en todo momento que libros han sido prestados, cuales les faltan por prestar, cuantos tiene ahora mismo.

De los empleados se desea conocer sus datos personales y conocer que prestamos realiza a que cliente.

De los libros se desea llevar de forma ordenada sus títulos, a que género pertenecen, quien es el/los autor/es, número de páginas que tienen, etc.

Por lo tanto, de los datos recibidos se asume que:

- El usuario, si está bloqueado, no puede volver a hacer un préstamo, hasta que haya sido desbloqueado.
- Un usuario estará desbloqueado cuando su valor sea 0, distinto de 0 será que está bloqueado.
- Cada préstamo tendrá un identificador, con información precisa para distinguir entre los distintos préstamos.
- Un usuario tendrá 14 días para entregar un libro, pudiendo hacerlo antes de dicho plazo.
- De los autores, se deseará conocer su nombre y apellidos. Así como, de su estilo el nombre.

3. Implementación sobre C# de la aplicación :

3.1. Formularios creados :

La aplicación muestra un sistema de botones que facilita la tarea al usuario para su mejor manejo, eso hace que tengamos diferentes formularios :

- Form1.cs : Donde se encuentra la pantalla principal del programa.
- FormAcercaDe.cs : En este esta una pequeña información de los creadores de la interfaz.
- FormEmpleados.cs : La creación de este formulario hace que sea más fácil para los empleados acceder a una zona más segura del programa.
- FormUsuario.cs : Formulario para el acceso de los usuarios, pudiendo crear, cargar, modificar o eliminar usuarios.

- FormLibreria.cs : Obviamente la gran cantidad de datos que hay que introducir al respecto hace necesario la creación de este formulario, que en su interior contiene otro formulario más.
- FormNuevoLibro.cs : Dentro del formulario anteriormente descrito esta este nuevo formulario, que hace mucho más legible la introducción de datos.
- FormPrestamos.cs : Es la zona para ver los prestamos de los libros a los diferentes usuarios.

Form1.cs :

```

1 reference
private void buttonUsuario_Click(object sender, EventArgs e) {...}

1 reference
private void buttonEmpleados_Click(object sender, EventArgs e) {...}

1 reference
private void buttonLibreria_Click(object sender, EventArgs e) {...}

1 reference
private void buttonPrestamos_Click(object sender, EventArgs e) {...}

1 reference
private void buttonAcercaDe_Click(object sender, EventArgs e) {...}

```

Contiene los diferentes métodos onClick(); para acceder a los demás formularios.

FormAcercaDe.cs :

Este formulario solo contiene labels.

FormEmpleados.cs :

```
private void cargarDatosEmpleado();
```

Este método esta relacionado a los demás, ya que los datos son comunes para el formulario.

```

try
{
    OracleConnection conexion;
    OracleCommand comando;
    OracleDataReader reader;
    .....
    reader.Close();
    conexion.Close();

    MessageBox.Show("Datos Cargados con éxito", "Aplicación Base de Datos",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}
catch (OracleException err)
{
    ....
}

```

Código ejemplo que se va a utilizar a lo largo del programa para conectarse a la BBDD.

Abre una nueva conexión a la base de datos para cargar en los métodos subyacentes los datos llamando a esta función.

```
private void usuariosBloqueados();
```

Como en la anterior, abre una nueva conexión con la base de datos para realizar las consultas.

Los siguientes métodos llaman a los métodos anteriores para realizar la conexión de cargar los datos de la base de datos.

```
private void buttonCargarUsuarios_Click();
```

```
private void buttonCargarDatosEmpleado_Click();
```

```
private void buttonCrearEmpleado_Click();
```

```
private void buttonModificarEmpleado_Click();
```

```
private void buttonEliminarEmpleado_Click();
```

```
private void listViewEmpleados_SelectedIndexChanged();
```

Esta es la zona de usuarios bloqueados, que al igual que el resto solo contienen las conexiones y las consultas.

```
private void buttonUsuariosBloqueados_Click();
```

```
private void buttonDesbloquearUsuario_Click();
```

```
private void listViewUsuariosBloqueados_SelectedIndexChanged();
```

FormEmpleados

ZONA DEL EMPLEADO

Empleado Acceso

Contraseña

Cargar Datos

Empleados Zona de Control

Cargar Usuarios

Desbloquear Usuario

Usuarios Bloqueados

USUARIOS

Usuario	Título	Fecha Inicial	Fecha Devol...	Fecha Limite
---------	--------	---------------	----------------	--------------

Usuario	Días Bloqueado
---------	----------------

FormEmpleados

ZONA DEL EMPLEADO

Cargar Datos

Empleado Acceso

Contraseña

Empleados | Zona de Control

ID Empleado

Nombre

Apellidos Empleado

Turno

Crear Empleado

Modificar Datos

Eliminar Empleado

ID	Nombre	Apellidos	Turno
----	--------	-----------	-------

Como se observan en las anteriores imágenes, se utilizan botones, labelText, tabControl (para cambiar de pestaña entre una zona y otra) y listView (para almacenar los datos cargados).

Aparte hemos implementado un sistema de contraseña con dos textBoxAccesoEmpleado y textBoxContraseña.

El tabControl nos ayuda como se ve a acceder a la sección de bloqueo de usuarios y la de creación de nuevos empleados.

US#	Nombre	Apellidos	Direccion	Días Bloqueo
-----	--------	-----------	-----------	--------------

Método de donde cargan los datos los demás métodos onClick();

```
private void cargarDatosUsuario();
```

Método principal del formulario, de donde acceden los demás métodos para cargar los datos.

```
private void buttonCargarDatosUsuario_Click();
```

```
private void listViewUsuario_SelectedIndexChanged();
```

```
private void buttonNuevoUsuario_Click();
```

```
private void buttonModificarDatosUsuario_Click();
```

```
private void buttonDarBajaUsuario_Click();
```

Este formulario, simplemente tiene textView para recoger los datos del listView, y así poder guardarlos para posteriormente poder tener una referencia y poder utilizarlos dentro de nuestro código.

La librería como se observa tiene únicamente 2 botones, para cargar datos y para acceder al formulario de Nuevo Libro y los listView para ver los datos.

Lo interesante de este form son el uso de los métodos que reciben las variables de otros lugares, siendo el caso de los listView (categoriaOrganizada y estiloOrganizado) que utilizan métodos con uso de variables referenciadas a los listView Categoria y Estilo con un selectIndexChanged.

```
private void cargarLibrosPorCantidadCategorias();
private void cargarLibrosPorCategoria(string cat);
private void cargarLibrosPorCantidadEstilos();
private void cargarLibrosPorEstilo(string estilo);
private void listViewCategoria_SelectedIndexChanged();
private void listViewEstilo_SelectedIndexChanged();
```

Estos dos métodos siguientes son los utilizados para cargar los datos, y para acceder al siguiente formulario.

```
private void buttonNuevoLibro_Click();
private void buttonCargarLibreria_Click();
```


FormNuevoLibro

ZONA DE CREACIÓN DE LIBROS

Cargar Datos

AU#

Nobre Autor

Apellidos Autor

LI#

Título del Libro

Número de Páginas

Número de Edición

Año del Libro

Estilo Literario

ES#

Tipo de Categoría

CA#

TABLAS DATOS

DETALLES LIBROS

Nuevo Libro

Modificar Datos

Eliminar Libros

AU#	Nombre Autor	Apellidos Autor	LI#	Título Libro	N° Páginas	N° Edición	Año	Estilo Literario	Ca
-----	--------------	-----------------	-----	--------------	------------	------------	-----	------------------	----

FormNuevoLibro

ZONA DE CREACIÓN DE LIBROS

Cargar Datos

AU#

Nobre Autor

Apellidos Autor

LI#

Título del Libro

Número de Páginas

Número de Edición

Año del Libro

Estilo Literario

ES#

Tipo de Categoría

CA#

TABLAS DATOS

DETALLES LIBROS

AUTORES

AU#	Nombre Autor	Apellidos Autores
-----	--------------	-------------------

LIBROS

LI#	Título Libro	N° Páginas	N° Edición	Año
-----	--------------	------------	------------	-----

9

Las imágenes reflejan la interfaz del formulario para la creación de un nuevo libro, la primera es la sección para ver los libros completos en el detalle, y la segunda es para poder tener separadas las listas de libro y autor, ya que puede haber más libros que autores.

Las herramientas utilizadas son las mismas que en los demás formularios, con la nueva característica que en este hemos optado por añadir 2 comboBox para estilo y categoría, ya que eso hace que los libros tengan una categoría predefinida, uniéndolo con el label de su lado y así poder usar ese dato cargado con las referencias en el código.

Los métodos son los siguientes :

```
private void cargarLibros();
```

Siendo el método general para acceder a las conexiones.

```
private void buttonCargarLibros_Click();
```

```
private void listViewAutores_SelectedIndexChanged();
```

```
private void listViewLibros_SelectedIndexChanged();
```

```
private void listViewLibrosCompleto_SelectedIndexChanged();
```

```
private void buttonNuevoLibro_Click();
```

```
private void buttonModificarLibro_Click();
```

```
private void buttonEliminarLibros_Click();
```

```
private void comboBoxEstilo_SelectedIndexChanged();
```

```
private void comboBoxCategoria_SelectedIndexChanged();
```

FormPrestamos.cs :

Este formulario nos permite crear nuevos prestamos, y ver los prestamos ya creados.

Usando 2 listView : listViewUsuarios y listViewPrestamos.

Y cargando los datos y crearlos con los botones, buttonCargarDatos y buttonCrearPrestamo.

FormPrestamos

ZONA PRESTAMOS

US#	Nombre	Apellidos	Direccion	Dias bloqueado
-----	--------	-----------	-----------	----------------

PR#	Fecha	Fecha devolucion	Fecha limite	Libro
-----	-------	------------------	--------------	-------

Cargar datos

Nuevo

Fecha dd/MM/yyyy

EM#

PR#

US#

Los métodos creados en este form son los siguientes :

```
private void cargarUsuarios();
```

Usamos este método para cargar los datos de manera global y lo llamamos en los métodos de cargar y crear.

```
private void cargarUsuariosConPrestamos(string US);
```

```
private void buttonCargarDatos_Click();
```

```
private void buttonCrearPrestamo_Click();
```

```
private void listViewUsuarios_SelectedIndexChanged();
```

Aquí recojo el dato para su posterior uso en `private void cargarUsuariosConPrestamos (string US);` , clicando en el listView.

4. Consideraciones finales y conclusiones.

La aplicación visual es básica para un uso concreto de base de datos para un cliente en concreto, por lo que las conclusiones sacadas de este proyecto son :

1ª) Esta experiencia ha mostrado cómo es posible diseñar y aplicar un aprendizaje basado en BBDD y lenguaje de programación, en una aplicación visual manejable para su uso práctico.

2ª) También ha mostrado que las aplicaciones generadas son útiles y valiosas para producir mejoras apreciables y significativas, siendo un método automático para introducir datos a tu BBDD virtual.

3ª) Sin embargo, la experiencia pone en evidencia la cantidad de acciones y funciones que se pueden desarrollar y por falta de tiempo y conocimientos no son posibles hacer.

4ª) Se observa a lo largo del proceso, que los datos exigidos en el proyecto son la espina que tienen los programadores para hacer el correcto programa que el usuario necesita.

5ª) La mejor manera de aprender el uso práctico de la conexión entre diferentes dispositivos.

6ª) Y propongo como última conclusión la observación de los 3 días iniciales de la práctica para crear el mejor diagrama, ya que ha sido un gran alivio que todas las consultas y los datos estuviesen correctamente dispuestos y no obtener ningún fallo ya que eso hubiese hecho que las consultas adscritas en la aplicación diesen muchos fallos.