



# E-SPORTS

MIKEL GARRIDO NOVOA

# ÍNDICE

- INTRODUCCIÓN
- GLOSARIO
- APLICACIÓN

## MANUAL TÉCNICO

- ARQUITECTURA DE LA BBDD
- ANÁLISIS CASOS DE USO
- DIAGRAMA DE SECUENCIA



# INTRODUCCIÓN

Esta presentación recoge la documentación del proyecto de fin de curso para las asignaturas Bases de Datos y Entornos de Desarrollo.

El proyecto consiste en el diseño de una base de datos y la creación de una aplicación de gestión sobre esta base de datos.



# GLOSARIO

Acrónimos:

BDD: Bases de Datos.

Casos de Uso: La funcionalidad con la que un usuario interactúa con el sistema.

Modelo E-R: Modelo Entidad-Relación



# APLICACIÓN

- Elección del tema: Escogí el tema de los E-Sports ya que es algo que personalmente me produce mucho interés y además tengo conocimientos sobre el tema.
- La aplicación tiene varias funciones principales que son las de mostrar, añadir, eliminar o modificar datos de la tabla de la base de datos seleccionada.
- Esta aplicación va dirigida a los gestores de clubes de E-Sports, para que puedan acceder y modificar los datos de los clubes que gestionan.



# MANUAL TÉCNICO

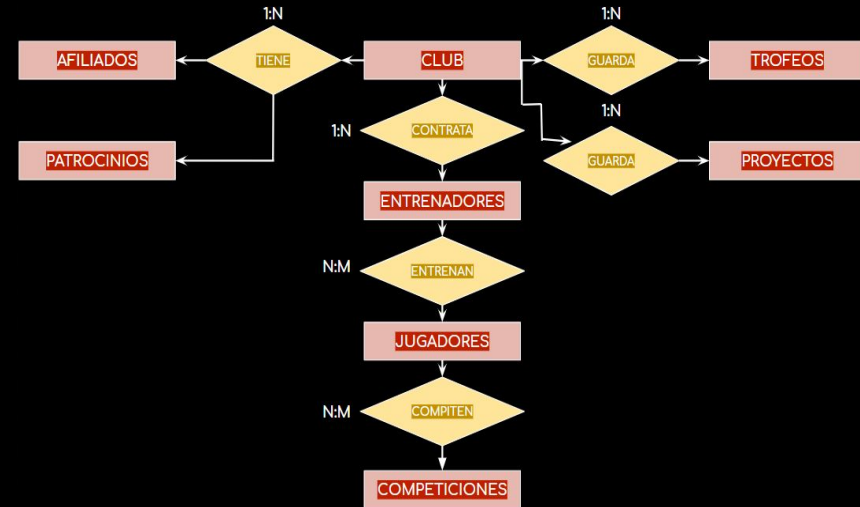
# ARQUITECTURA de la BBDD

La base de datos consta de 10 entidades, de las cuales 8 son entidades principales y 2 son entidades creadas por relaciones (N:M).

Entidades principales: Afiliados, Club, Competiciones, Entrenadores, Jugadores, Patrocinios, Proyectos, Trofeos.

Resto de entidades: Entrena, Compite.

Modelo E-R:



# ANÁLISIS CASOS DE USO

Los casos de uso de esta aplicación se basan en 4 funciones principales que se utilizan para cada entidad, estos son:

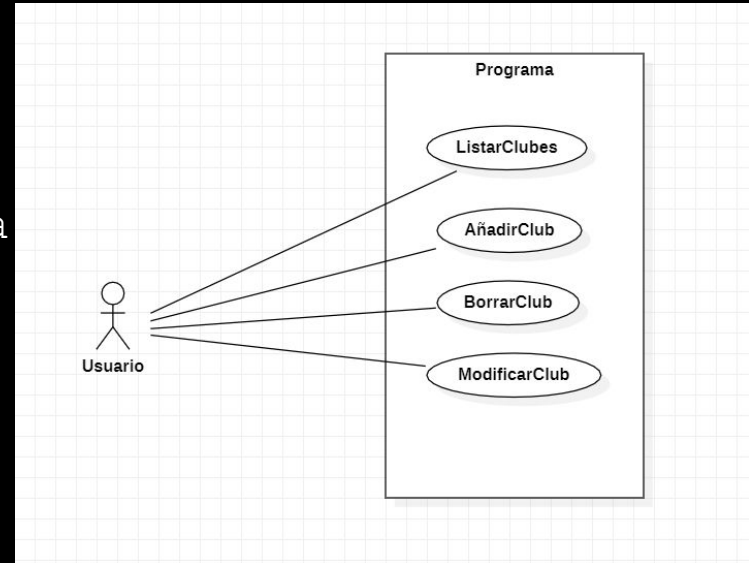
-VisualizarDatos

-AñadirDatos

-EliminarDatos

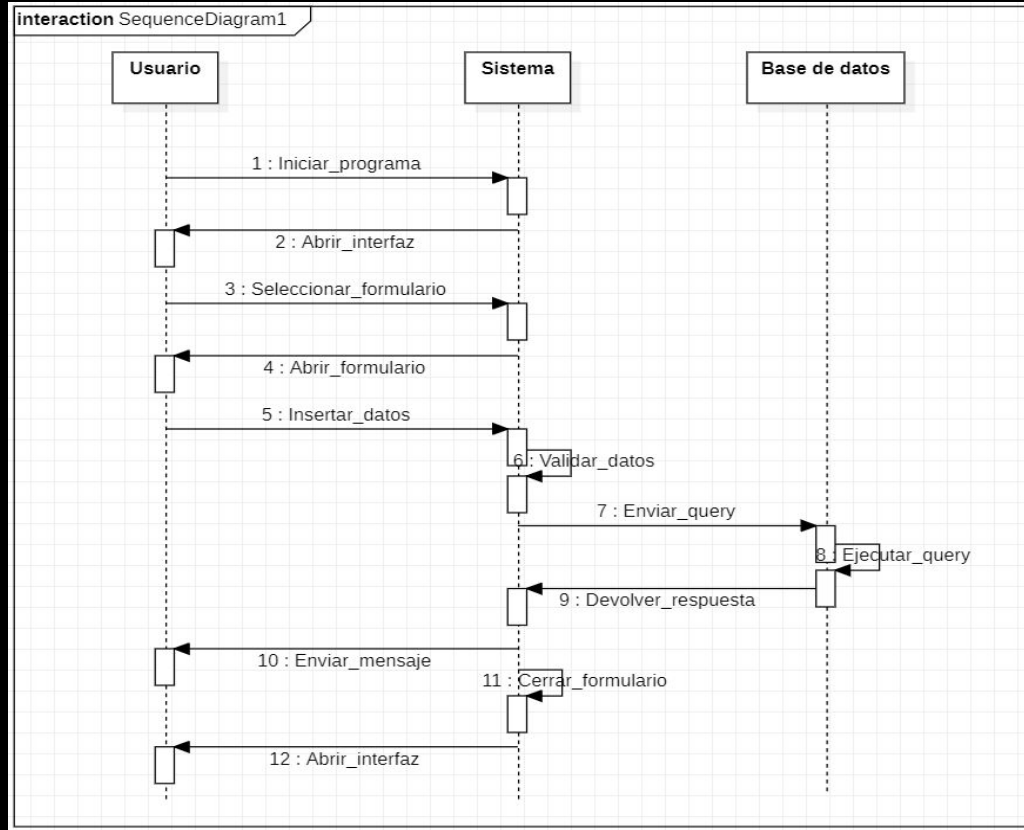
-ModificarDatos

Ejemplo diagrama  
casos de uso de  
la entidad Club:





# DIAGRAMA DE SECUENCIA



# PROGRAMACION/ DESARROLLO

# Herramientas y lenguaje

Herramientas:

- Visual Studio 2019

- Microsoft SQL Server Management Studio 18

Lenguajes:

- Visual Basic

- tSQL

# PROCEDIMIENTOS ALMACENADOS

Listar:

```
-- LISTAR
IF OBJECT_ID('SP_listarClubes') IS NOT NULL
BEGIN
    DROP PROC SP_listarClubes
END
GO

CREATE PROC SP_listarClubes
as
BEGIN
    SELECT IDCLUB AS ID
    ,NOMBRE
    FROM CLUB
END
GO
```

Crear:

```
-- CREAM
IF OBJECT_ID('SP_crearClub') IS NOT NULL
BEGIN
    DROP PROC SP_crearClub
END
GO

CREATE PROC SP_crearClub
@IDCLUB NVARCHAR(10),
@NOMBRE NVARCHAR(50)
as
BEGIN
    INSERT INTO CLUB([IDCLUB]
    ,[NOMBRE])
    VALUES(@IDCLUB, @NOMBRE)
END
GO
```

Modificar:

```
-- ACTUALIZAR
IF OBJECT_ID('SP_modificarClub') IS NOT NULL
BEGIN
    DROP PROC SP_modificarClub
END
GO

CREATE PROC SP_modificarClub
@IDCLUB NVARCHAR(10),
@NOMBRE NVARCHAR(50)
as
BEGIN
    UPDATE CLUB
    SET [NOMBRE] = @NOMBRE
    WHERE [IDCLUB] = @IDCLUB
END
GO
```

Eliminar:

```
-- BORRAR
IF OBJECT_ID('SP_borrarClub') IS NOT NULL
BEGIN
    DROP PROC SP_borrarClub
END
GO

CREATE PROC SP_borrarClub
@IDCLUB NVARCHAR(10)
as
BEGIN
    DELETE FROM CLUB WHERE IDCLUB = @IDCLUB
END
GO
```

# ARQUITECTURA EN CAPAS

La aplicación está basada en 4 capas distintas:

- capaDatos
- capaEntidad
- capaNegocio
- capaPresentacion

# capaDatos

Esta capa recoge las funciones y las subrutinas que llaman a los procedimientos almacenados de la base de datos de cada tabla.

Ejemplo con la tabla clubes:

```
Function listarClubes() As DataSet
    Try
        cn = objConexion.conectar
        da = New SqlDataAdapter("SP_listarClubes", cn)
        Dim ds As New DataSet
        da.Fill(ds, "Clubes")
        Return ds
    Catch ex As Exception
        Throw ex
    Finally
        cn.Dispose()
        da.Dispose()
    End Try
End Function
```

```
Sub borrarClubes(ByVal obj As CClub)
    Try
        cn = objConexion.conectar
        cn.Open()
        da = New SqlDataAdapter("SP_borrarClub", cn)
        da.SelectCommand.CommandType = CommandType.StoredProcedure
        With da.SelectCommand.Parameters
            .Add("@IDCLUB", SqlDbType.NVarChar).Value = obj.IDCLUB
        End With
        da.SelectCommand.ExecuteNonQuery()
        MsgBox("¡Club eliminado con éxito!", MsgBoxStyle.Information)
    Catch ex As Exception
        Throw ex
    Finally
        cn.Dispose()
        da.Dispose()
        cn.Close()
    End Try
End Sub
```

```
Sub registroClubes(ByVal obj As CClub)
    Try
        cn = objConexion.conectar
        cn.Open()
        da = New SqlDataAdapter("SP_crearClub", cn)
        da.SelectCommand.CommandType = CommandType.StoredProcedure
        With da.SelectCommand.Parameters
            .Add("@IDCLUB", SqlDbType.NVarChar).Value = obj.IDCLUB
            .Add("@NOMBRE", SqlDbType.NVarChar).Value = obj.NOMBRE
        End With
        da.SelectCommand.ExecuteNonQuery()
        MsgBox("¡Registro realizado con éxito!", MsgBoxStyle.Information)
    Catch ex As Exception
        Throw ex
    Finally
        cn.Dispose()
        da.Dispose()
        cn.Close()
    End Try
End Sub
```

```
Sub modificarClubes(ByVal obj As CClub)
    Try
        cn = objConexion.conectar
        cn.Open()
        da = New SqlDataAdapter("SP_modificarClub", cn)
        da.SelectCommand.CommandType = CommandType.StoredProcedure
        With da.SelectCommand.Parameters
            .Add("@IDCLUB", SqlDbType.NVarChar).Value = obj.IDCLUB
            .Add("@NOMBRE", SqlDbType.NVarChar).Value = obj.NOMBRE
        End With
        da.SelectCommand.ExecuteNonQuery()
        MsgBox("¡Club modificado con éxito!", MsgBoxStyle.Information)
    Catch ex As Exception
        Throw ex
    Finally
        cn.Dispose()
        da.Dispose()
        cn.Close()
    End Try
End Sub
```

# capaEntidad

Esta capa recoge los getters y setters de los atributos de la tabla.

Ejemplo con la tabla clubes:

```
Public Class CEClub

    Private _IDCLUB As String
    Private _NOMBRE As String

    6 referencias
    Public Property IDCLUB As String
    |
    | Get
    |     Return _IDCLUB
    | End Get
    | Set(value As String)
    |     _IDCLUB = value
    | End Set
    End Property

    4 referencias
    Public Property NOMBRE As String
    |
    | Get
    |     Return _NOMBRE
    | End Get
    | Set(value As String)
    |     _NOMBRE = value
    | End Set
    End Property
End Class
```

# capaNegocio

Esta capa contiene funciones que llaman a las funciones de la capaDatos para que los datos vayan pasando entre las capas.

Ejemplo con la tabla clubes:

```
Imports capaDatos
Imports capaEntidad

4 referencias

Public Class CNClub
    Dim objCapaDatos As New CDClub

    2 referencias
    Function listarClubes() As DataSet
        Return objCapaDatos.listarClubes
    End Function

    1 referencia
    Sub registroClubes(ByVal obj As CEClub)
        objCapaDatos.registroClubes(obj)
    End Sub

    1 referencia
    Sub borrarClubes(ByVal obj As CEClub)
        objCapaDatos.borrarClubes(obj)
    End Sub

    1 referencia
    Sub modificarClubes(ByVal obj As CEClub)
        objCapaDatos.modificarClubes(obj)
    End Sub
End Class
```



# capaPresentacion

Esta capa contiene los formularios y su funcionalidad

Ejemplo con la tabla clubes:



```
Imports capaNegocio
```

```
1 referencia
```

```
Public Class ListadoClubes
```

```
0 referencias
```

```
Private Sub ListadoClubes_Load(sender As Object, e As EventArgs) Handles MyBase.Load
```

```
    listarClubes()
```

```
End Sub
```

```
1 referencia
```

```
Sub listarClubes()
```

```
    Dim objCapaNegocio As New CNCub
```

```
    Me.DataGridView1.DataSource = objCapaNegocio.listarClubes.Tables("CLUBES")
```

```
    Me.lblTotal.Text = objCapaNegocio.listarClubes.Tables("CLUBES").Rows.Count
```

```
End Sub
```

```
0 referencias
```

```
Private Sub btnBorrar_Click(sender As Object, e As EventArgs) Handles btnBorrar.Click
```

```
    Me.Close()
```

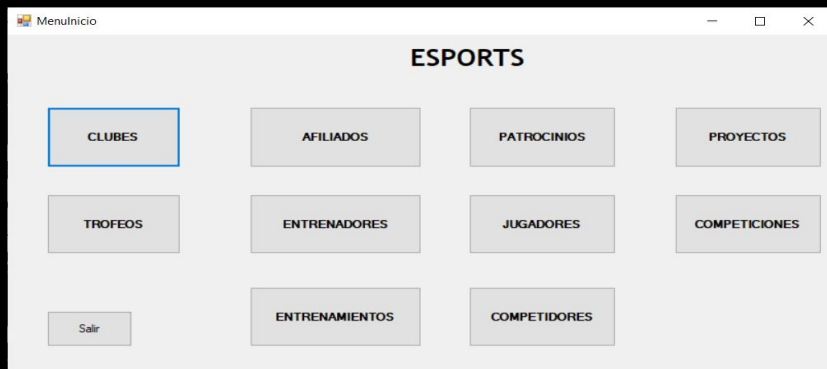
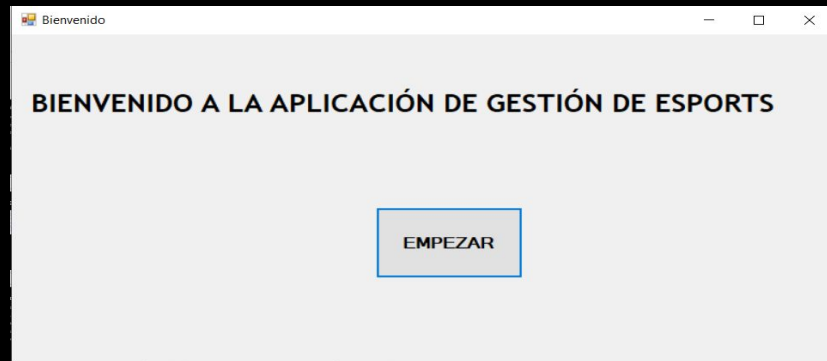
```
End Sub
```

```
End Class
```

# MANUAL DE USUARIO

# Inicio

Al iniciar la aplicación aparece esta ventana:



Si pulsamos en "Empezar" se abrirá este otro formulario:

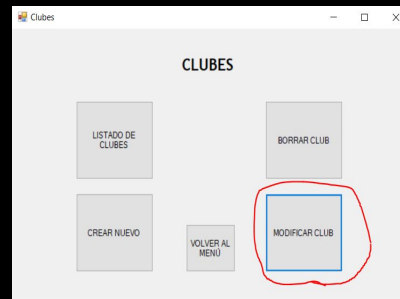
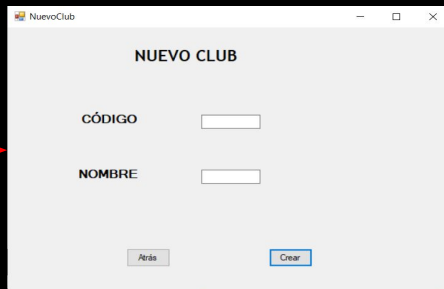
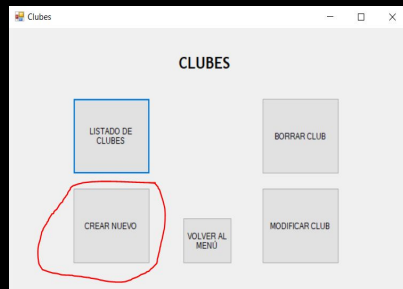
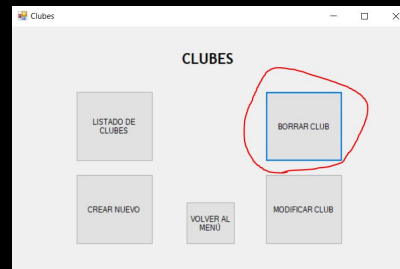
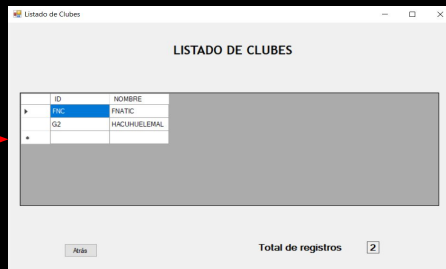
# Menú y formularios

Si en el menú principal pulsamos en una de las pestañas se nos abrirá otro formulario con las funciones sobre esa tabla:



# Funcionalidades formulario

El formulario que se nos ha abierto tiene varias funcionalidades:



# Conclusiones

Esta ha sido mi primera aplicación en visual basic y con esta estructura de capas, y con ella he aprendido muchas cosas sobre las diferentes formas de desarrollar una aplicación.