



*Centro Oficial de
Formación Profesional*

RRPP DE VIDEOJUEGOS

Joaquín Machuca Baena

tutor: Eduardo murillo Garrido

Sevilla 2022

1. Portada.
2. Índice
3. Resumen:
4. Introducción:
5. Objetivos y características del proyecto
- 6.-Desarrollo
- 7.-Medios materiales usados
8. Referencias bibliográficas
9. Conclusiones y trabajos futuros o posibles mejoras

Resumen:

Este trabajo de fin de grado superior se basa en la creación de un programa de administración de bases de datos relacionadas con el mundo de los videojuegos.

Este programa se encarga de visualizar datos introducidos previamente en la B.B.D.D y de ser capaz de extraer datos de ellos a la vez que introducirlo. Todo ello a través de una interfaz amigable para el usuario

Introducción

En este día definiremos de manera efectiva las partes del trabajo. Antes de empezar deberíamos definir qué es un videojuego:

Un videojuego es una aplicación interactiva orientada al entretenimiento que, a través de ciertos mandos o controles, permite simular experiencias en la pantalla de un televisor, una computadora u otro dispositivo electrónico.

Los videojuegos se diferencian de otras formas de entretenimiento, como ser las películas, en que deben ser interactivos; en otras palabras, los usuarios deben involucrarse activamente con el contenido. Para ello, es necesario utilizar un mando (también conocido como gamepad o joystick), mediante el cual se envían órdenes al dispositivo principal (un ordenador o una consola especializada) y estas se ven reflejadas en una pantalla con el movimiento y las acciones de los personajes.

Nuestro objetivo primordial en el trabajo que vamos a presentar es definir una base de datos de escritorio en la que tengamos la capacidad de crear consultas directas a una base de datos creada previamente o creada por el usuario.

Continuaremos especificando que es una base de datos:

Una base de datos es una recopilación organizada de información o datos estructurados, que normalmente se almacena de forma electrónica en un sistema informático. Normalmente, una base de datos está controlada por un sistema de gestión de bases de datos (DBMS). En conjunto, los datos y el DBMS, junto con las aplicaciones asociadas a ellos, reciben el nombre de sistema de bases de datos, abreviado normalmente a simplemente base de datos.

Los datos de los tipos más comunes de bases de datos en funcionamiento actualmente se suelen utilizar como estructuras de filas y columnas en una serie de tablas para aumentar la eficacia del procesamiento y la consulta de datos. Así, se puede acceder, gestionar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan un lenguaje de consulta estructurada (SQL) para escribir y consultar datos.

Source: <https://www.oracle.com/mx/database/what-is-database/>

Ya con nuestros dos principales temas definidos nos dispondremos a verificar que Herramientas, IDE y servidores de bases de datos usaremos:

Nuestra principal herramienta para trabajar será Visual Studio 2022 es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para Windows y macOS. Es compatible con múltiples lenguajes de programación, tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby y PHP, al igual que entornos de desarrollo web, como ASP.NET MVC, Django, etc., a lo cual hay que sumarle las nuevas capacidades en línea bajo Windows Azure en forma del editor Mónico.



Visual Studio permite a los desarrolladores crear sitios y aplicaciones web, así como servicios web en cualquier entorno compatible con la plataforma .NET (a partir de la versión .NET 2002). Así, se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos y videoconsolas, entre otros.

Visual Studio 2022 además posee una interfaz realmente intuitiva y es bastante sencillo tener organizadas todas las carpetas de trabajo para su desarrollo.



Nuestra siguiente herramienta de trabajo que usaremos será Microsoft SQL Server Studio que sin duda será el añadido perfecto para compaginar con Visual Studio 2022.

Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado (por línea de comandos o mediante la interfaz gráfica de Management Studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

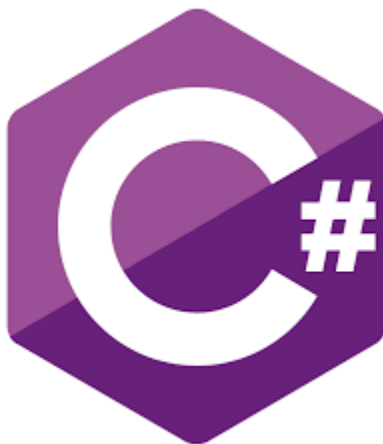
Dentro de los competidores más destacados de SQL Server están: Oracle, MariaDB, MySQL, PostgreSQL. SQL Server ha estado tradicionalmente disponible solo para sistemas operativos Windows de Microsoft, pero desde 2016 está disponible para GNU/Linux,²³ y a partir de 2017 para Docker también.

Puede ser configurado para utilizar varias instancias en el mismo servidor físico, la primera instalación lleva generalmente el nombre del servidor, y las siguientes - nombres específicos (con un guion invertido entre el nombre del servidor y el nombre de la instalación).

Continuare con herramientas secundarias dando el nombre y una breve descripción:

- Photopea: Herramienta online en sustitución a photoshop usada para pequeños arreglos de edición de imagen
-

Después ver los programas que he usado para el trabajo veremos los dos lenguajes elegidos para la creación del programa:



"C#" (pronunciado ci sharp en inglés) es un lenguaje de programación multiparadigma desarrollado y estandarizado por la empresa Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por el signo #, el cual se lee como sharp en inglés para notación musical. Es un juego de palabras, pues "C#" significa, musicalmente hablando, "do sostenido", donde el símbolo # indica que una nota (en este caso do, representada por C) debe ser un semitono más alta. Esto es una metáfora de la superioridad de C# sobre su antecesor C++ y a su vez hace alusión a la misma metáfora que se ideó para dar nombre a C++.1 Además, el símbolo # puede ser imaginado como la unión de cuatro símbolos +, continuando así con el sentido de progresión de los lenguajes C.

Aunque C# forma parte de la plataforma .NET, esta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows Microsoft, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

Este lenguaje es el apropiado para la utilización de herramientas .NET y creo que es el mas propio para un trabajo de esta índole

El segundo lenguaje que usaremos será de BBDD y será SQL que es el que trabaja SQL server



-SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje de dominio específico, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.² Una de sus principales características es el manejo del álgebra y el cálculo relacional para efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como realizar cambios en ellas.

Originalmente basado en el álgebra relacional y en el cálculo relacional, SQL consiste en un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de control de datos. El alcance de SQL incluye la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de esquemas y el control de acceso a los datos. También el SQL a veces se describe como un lenguaje declarativo, también incluye elementos procesales.

SQL fue uno de los primeros lenguajes comerciales para el modelo relacional de Edgar Frank Codd como se describió en su artículo de investigación de 1970 El modelo relacional de datos para grandes bancos de datos compartidos. A pesar de no adherirse

totalmente al modelo relacional descrito por Codd, pasó a ser el lenguaje de base de datos más usado.

SQL pasó a ser el estándar del Instituto Nacional Estadounidense de Estándares (ANSI) en 1986 y de la Organización Internacional de Normalización (ISO) en 1987. Desde entonces, el estándar ha sido revisado para incluir más características. A pesar de la existencia de ambos estándares, la mayoría de los códigos SQL no son completamente portables entre sistemas de bases de datos diferentes sin otros ajustes. SQL pasó a ser el estándar del Instituto Nacional Estadounidense de Estándares (ANSI) en 1986 y de la Organización Internacional de Normalización (ISO) en 1987. Desde entonces, el estándar ha sido revisado para incluir más características. A pesar de la existencia de ambos estándares, la mayoría de los códigos SQL no son completamente portables entre sistemas de bases de datos diferentes sin otros ajustes.

Este es el lenguaje estándar para las consultas y para la creación de bbdd

Source: <https://www.wikipedia.org/>

Objetivos y características del proyecto

Uno de los principales objetivos de este proyecto radica en la memoria y estructuración de juegos y su ordenación, ya sea para el jugador promedio que pueda tenerlo guardado y eficientemente ordenado, o aquella empresa de big data que quiera almacenar todos los datos referentes a los juegos creados por una compañía.

El programa posee una gran capacidad de búsqueda y de especificación, ya sea para puntuar, añadir datos o eliminarlos del mismo. La base de datos que se sustenta posee un numero de videojuegos exacerbada que ayuda a la extracción de un gran número de videojuegos

La página constará de una página de login con el logo del programa, la fuente del programa estará basado en la tipografía usada de SEGA.

Dentro del programa tendrá dos textinput, uno de usuario y otro de contraseña conectará con una base de datos y admitirá o no la entrada.

Una vez dentro del programa se presentará un menu conciso sobre las distintas opciones que se podrán realizar entre las que destaca: “insertar registro”, “buscar videojuego”, “eliminar videojuego” y “ranking de videojuegos”.

- Insertar registro: Es un apartado que te permite conectarte a una base de datos e introducir a las características propias de un videojuego. El botón de introducir crea un insert y mete los datos dentro de la bbdd
- Borrar videojuego: Busca en la bbdd y elimina uno de los registros
- Buscar videojuego: busca un titulo y te da todos los datos necesarios
- Ranking de videojuegos: Usa las mayores notas de los videojuegos de toda la bbdd para crear el orden de los mejores títulos y ordenarlos

Todas las paginas usan el mismo background utilizando tonos morados para resaltar los útiles de color azul y fondo blanco como colores secundarios.

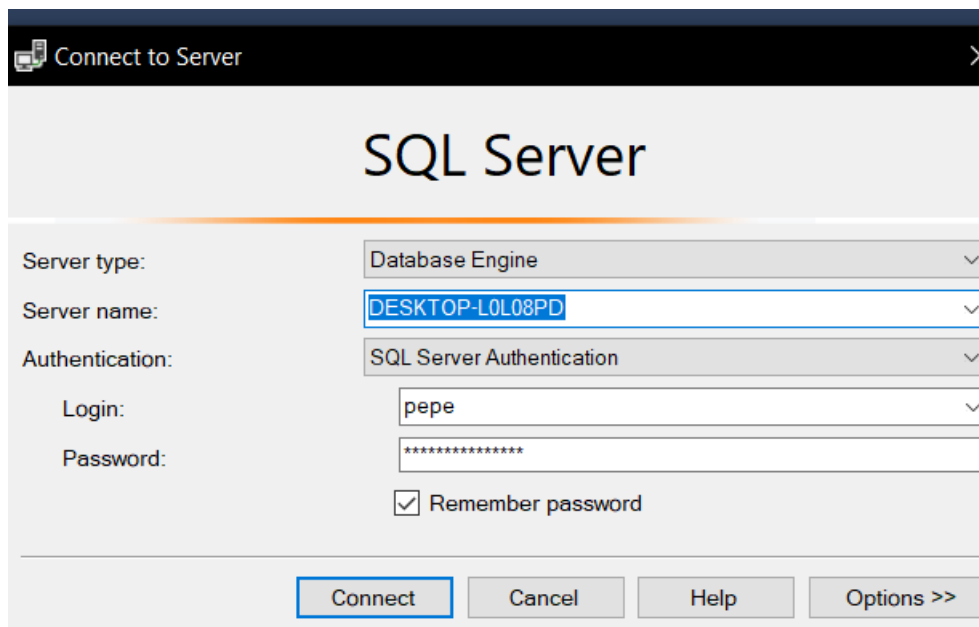


TEXTO DE PRUEBA

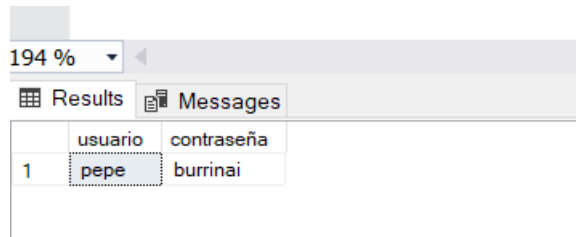
Desarrollo

Lo primero que me gustaría comentar es el uso que le he dado a las aplicaciones de Microsoft las cuales han sido casi en su totalidad, y sin duda, han sido mas que efectivas y no he tenido ningún problema de sincronización ni de versiones.

Empezando por sql server, he creado un usuario y le he dado todos los permisos necesarios para que pueda trabajar con facilidad con el programa.



A posteriori cree una tabla dedicada al futuro login que crearé introduciendo un usuario y un password:

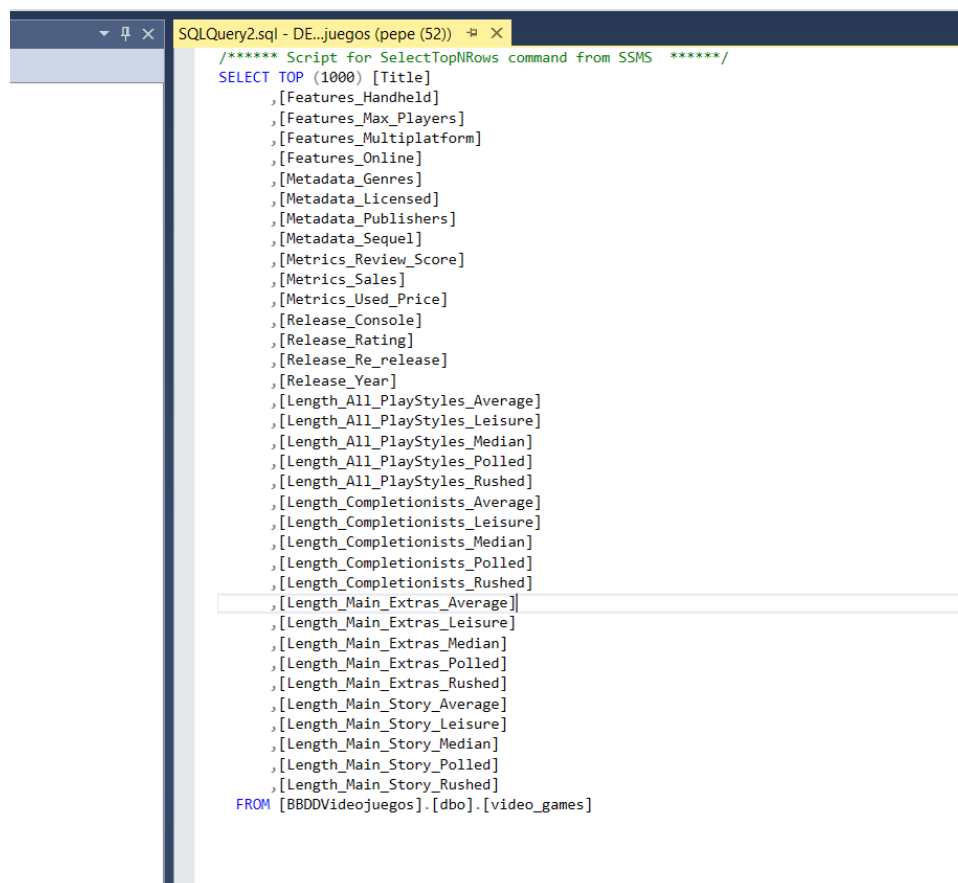


The screenshot shows a SQL Server Enterprise Manager interface. At the top, there's a zoom level of 194%. Below it, there are tabs for 'Results' and 'Messages'. The 'Results' tab is active, displaying a table with two columns: 'usuario' and 'contraseña'. The first row of the table contains the values 'pepe' and 'burrinai'.

	usuario	contraseña
1	pepe	burrinai

Como quería convertir esta base de datos en una dedicada a videojuegos, me fui a github y encontré diversas personas que habían subido a sus repositorios bases de datos relacionadas con juegos. Encontré una con mas de 1200 registros y con infinitud de datos que podría usar en mi programa así que la importe desde un archivo de Excel.

No fue difícil y no tardé encontrar un método sencillo para realizarlo.



The screenshot shows a SQL query script in SQL Server Enterprise Manager. The script is titled 'SQLQuery2.sql - DE...juegos (pepe (52))'. It contains a comment: '/****** Script for SelectTopNRows command from SSMS *****/'. The query is a SELECT statement with a TOP clause of 1000. It lists various columns related to video games, including features, metadata, metrics, release information, and lengths. The columns are: Title, Features_Handheld, Features_Max_Players, Features_Multiplatform, Features_Online, Metadata_Genres, Metadata_Licensed, Metadata_Publishers, Metadata_Sequel, Metrics_Review_Score, Metrics_Sales, Metrics_Used_Price, Release_Console, Release_Rating, Release_Re_release, Release_Year, Length_All_PlayStyles_Average, Length_All_PlayStyles_Leisure, Length_All_PlayStyles_Median, Length_All_PlayStyles_Polled, Length_All_PlayStyles_Rushed, Length_Completionists_Average, Length_Completionists_Leisure, Length_Completionists_Median, Length_Completionists_Polled, Length_Completionists_Rushed, Length_Main_Extras_Average, Length_Main_Extras_Leisure, Length_Main_Extras_Median, Length_Main_Extras_Polled, Length_Main_Extras_Rushed, Length_Main_Story_Average, Length_Main_Story_Leisure, Length_Main_Story_Median, Length_Main_Story_Polled, and Length_Main_Story_Rushed. The query ends with a FROM clause: 'FROM [BBDDVideojuegos].[dbo].[video_games]'.

```
/****** Script for SelectTopNRows command from SSMS *****/  
SELECT TOP (1000) [Title]  
    ,[Features_Handheld]  
    ,[Features_Max_Players]  
    ,[Features_Multiplatform]  
    ,[Features_Online]  
    ,[Metadata_Genres]  
    ,[Metadata_Licensed]  
    ,[Metadata_Publishers]  
    ,[Metadata_Sequel]  
    ,[Metrics_Review_Score]  
    ,[Metrics_Sales]  
    ,[Metrics_Used_Price]  
    ,[Release_Console]  
    ,[Release_Rating]  
    ,[Release_Re_release]  
    ,[Release_Year]  
    ,[Length_All_PlayStyles_Average]  
    ,[Length_All_PlayStyles_Leisure]  
    ,[Length_All_PlayStyles_Median]  
    ,[Length_All_PlayStyles_Polled]  
    ,[Length_All_PlayStyles_Rushed]  
    ,[Length_Completionists_Average]  
    ,[Length_Completionists_Leisure]  
    ,[Length_Completionists_Median]  
    ,[Length_Completionists_Polled]  
    ,[Length_Completionists_Rushed]  
    ,[Length_Main_Extras_Average]  
    ,[Length_Main_Extras_Leisure]  
    ,[Length_Main_Extras_Median]  
    ,[Length_Main_Extras_Polled]  
    ,[Length_Main_Extras_Rushed]  
    ,[Length_Main_Story_Average]  
    ,[Length_Main_Story_Leisure]  
    ,[Length_Main_Story_Median]  
    ,[Length_Main_Story_Polled]  
    ,[Length_Main_Story_Rushed]  
FROM [BBDDVideojuegos].[dbo].[video_games]
```

Como se muestra en la imagen de arriba, la tabla tenia infinitud de datos que se podrían llegar a usar en el programa, a su vez los registros databan de toda clase de juegos

	Title	Features_Handheld	Features_Max_Players	Features_Multiplatform	Features_Online	Metadata_Genres	Metadata_Licensed	Metadata_Publishers	Metadata_Sequel	Metrics_Review_Score	Metrics_Sales	Metrics_Used_Price
1	Super Mario 64 DS	1	1	1	1	Action	1	Nintendo	1	85	489	2495
2	Lumines Puzzle Fusion	1	1	1	1	Strategy	1	Ubisoft	1	89	56	1495
3	WaveWave Touch'd	1	2	1	1	Action,Racing / Driving,Sports	1	Nintendo	1	81	54	2295
4	Hot Shots Golf: Open Tee	1	1	1	1	Sports	1	Sony	1	81	49	1295
5	Spider-Man 2	1	1	1	1	Action	1	Activision	1	61	45	1495
6	The Urbz: Sims in the City	1	1	1	1	Simulation	1	EA	1	67	41	1295
7	Ridge Racer	1	1	1	1	Racing / Driving	1	Namco	1	88	36	1995
8	Metal Gear Acid	1	1	1	1	Strategy	1	Konami	1	75	34	1795
9	Madden NFL 2005	1	1	1	1	Sports	1	EA	1	68	25	895
10	Pokemon Dash	1	1	1	1	Racing / Driving	1	Nintendo	1	46	22	2495
11	Dynasty Warriors	1	1	1	1	Action,Adventure,Role-Playing (RPG)	1	NULL	1	62	2	1495
12	Feel the Magic XYXX	1	1	1	1	Action,Adventure,Racing / Driving,Sports	1	Sega	1	75	16	895
13	Ridge Racer DS	1	1	1	1	Racing / Driving	1	Namco	1	63	15	995
14	Darkstalkers Chronicle: The Chaos Tower	1	1	1	1	Action	1	Capcom	1	74	14	1495
15	Ape Escape Academy	1	4	1	1	Action,Sports	1	Sony	1	51	13	1495
16	Polarium	1	1	1	1	Strategy	1	Nintendo	1	73	12	895
17	Asphalt Urban GT	1	1	1	1	Racing / Driving,Simulation	1	Ubisoft	1	60	11	1495
18	Zoo Keeper	1	1	1	1	Action	1	NULL	1	74	1	1495
19	Mr. DRILLER: Drill Spirits	1	1	1	1	Action	1	Namco	1	72	9	1495
20	Spring	1	1	1	1	Adventure	1	Ubisoft	1	43	6	895
21	Armored Core: Formula Front - Extreme Battle	1	1	1	1	Action,Strategy	1	NULL	1	66	5	1695
22	Puyo Pop Fever	1	1	1	1	Action,Strategy	1	Sega	1	76	4	1995
23	Mario Kart DS	1	1	1	1	Racing / Driving	1	Nintendo	1	91	999	2995
24	Nintendogs	1	1	1	1	Simulation	1	Nintendo	1	83	972	2695
25	Brain Age: Train Your Brain in Minutes a Day!	1	1	1	1	Action	1	Nintendo	1	77	507	1795
26	Brain Age: More Training in Minutes a Day!	1	1	1	1	Action	1	Nintendo	1	77	365	1795
27	Grand Theft Auto: Liberty City Stories	1	1	1	1	Action,Racing / Driving	1	Capcom,Rockstar	1	88	297	1795
28	Animal Crossing: Wild World	1	1	1	1	Simulation	1	Nintendo	1	86	26	2495
29	Big Brain Academy	1	1	1	1	Action	1	Nintendo	1	74	177	1695
30	Call of Duty 2	1	4	1	1	Action	1	Activision,Konami	1	89	169	1795
31	Midnight Club 3: DUB Edition	1	1	1	1	Racing / Driving	1	Rockstar	1	74	169	1495
32	Need for Speed: Most Wanted S-1-0	1	1	1	1	Racing / Driving	1	EA	1	72	167	1495
33	Pokemon Mystery Dungeon: Blue Rescue Te...	1	1	1	1	Role-Playing (RPG)	1	Nintendo	1	62	122	1495
34	Sonic Rush	1	1	1	1	Action	1	Sega	1	82	12	1495
35	Star Wars: Battlefront II	1	1	1	1	Action	1	NULL	1	69	109	1695
36	SOCOM: U.S. Navy SEALs - Fireteam Bravo	1	1	1	1	Action	1	Sony	1	82	107	1495
37	Need for Speed: Most Wanted	1	2	1	1	Racing / Driving	1	EA	1	83	95	1795
38	Zoo Tycoon DS	1	1	1	1	Simulation,Strategy	1	THQ	1	44	92	1295
39	The Sims 2	1	1	1	1	Role-Playing (RPG),Simulation	1	EA	1	70	81	1795

Una vez finalizado la parte de la base de datos nos iremos a visual studio a formar y a definir la interfaz de usuario.


Lo primero que podemos observar es que el programa salta directamente en el centro de la pantalla para ello habremos de marcar la casilla centerTheMiddleScreen que permite Visual, a su vez tmb hemos anulado el: “maximizar pantalla” para así evitar errores de indentacion.

Decidí que un tamaño optimo para el programa debia ser de 1280:980 ya que era el tamaño común de las pantallas full hd y las mas adecuadas para un programa simplista como este.

Bajo un fondo de tonos morados nos encontramos con la pantalla de titulo, no quise ponerle un titulo complejo al programa ya que perdíamos demasiado tiempo para algo que es bastante simple su función. Usamos una fuente relacionado con la marca de videoconsolas y videojuegos SEGA para basarnos en la fuente.

BBDD DE VIDEOJUEGOS

El título ocupará la mitad de la pantalla de Login y la otra mitad estará ocupado por dos textinput el de login y el de password que verificarán si te has logeado correctamente



A login form on a purple background. It features the label "LOGIN:" in blue text above a white text input field. Below this, it features the label "PASSWORD:" in blue text above another white text input field.

La password tiene ocultado de caracteres para hacerlo lo mas real posible a un login.

En cuanto a los button que decidí usar me centré en la simpleza de dos tipos: los de dentro de la aplicación y los que te sacan fuera.



Two buttons on a purple background. The "ENTRAR" button is blue with white text and is located in the top left. The "SALIR" button is also blue with white text and is located in the bottom right.

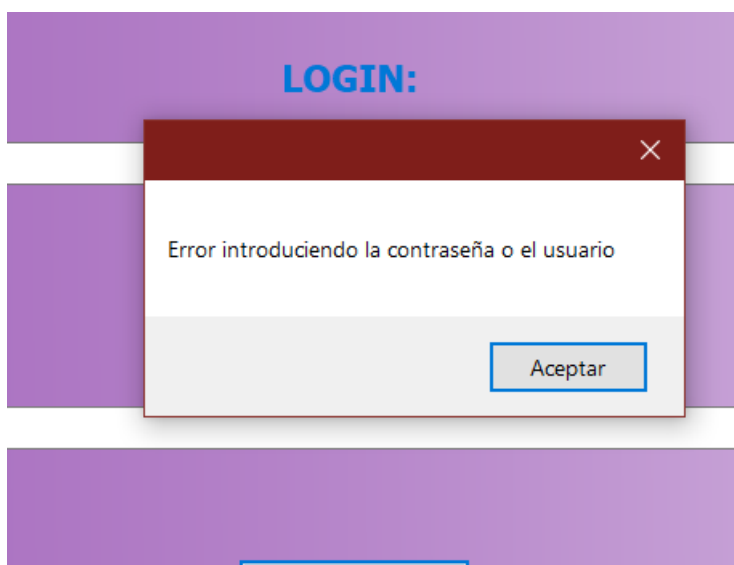
He usado el cambio de color de letra para diferenciarlo claramente

Introduce una característica especial en el código del login que impide que el usuario pueda meter números y símbolos en el nombre:

```
private void ControlBotones()
{
    if (NombreInput.Text.Trim() != String.Empty && NombreInput.Text.All(Char.IsLetter))
    {
        botonLogin.Enabled = true;
        errorProvider1.SetError(NombreInput, "");
    }
    else
    {
        if (!(NombreInput.Text.All(Char.IsLetter)))
        {
            errorProvider1.SetError(NombreInput, "El nombre solo debe contener letras");
        }
        else
        {
            errorProvider1.SetError(NombreInput, "Debe introducir su nombre");
        }
        botonLogin.Enabled = false;
        NombreInput.Focus();
    }
}
```

Esto facilitaría que el usuario introduzca bien el nombre de usuario.

El programa también está habilitado para realizar una pantalla de error si no coincide el usuario y la contraseña con la base de datos



La consulta para realizar la comprobación es esta

```
List<string> listaLogin = new List<string>();
listaLogin= Conexion.queryF("SELECT * FROM usuario").Select(i => i.ToString()).ToList();
if (listaLogin[0] == NombreInput.Text && listaLogin[1] == passwordInput.Text)
{
```

Que se aplica siempre que se carga la pagina.

La pagina del menú consta de 4 botones que te dirigen a las distintas partes del menú, son botones qamplios que ocupan casi la totalidad de la pantalla. Es la pagina principal y todas las pantallas de esta tienen la posibilidad de volver al menú original



Aquí podemos observar como nos introducimos en la pantalla apropiada y escondemos la pantalla actual para que no se acoplen

```
1 referencia
private void button1_Click(object sender, EventArgs e)
{
    InsertarVideojuego insertarVideojuego = new InsertarVideojuego();
    this.Hide();
    insertarVideojuego.ShowDialog();
}

1 referencia
private void button2_Click(object sender, EventArgs e)
{
    EliminarMenu eliminarMenu = new EliminarMenu();
    this.Hide();
    eliminarMenu.ShowDialog();
}
```

Una de los primeras partes del programa se llama insertar y manda directamente datos a la base de datos y esta los almacena.

Le activamos una característica que anulaba la posibilidad de insertar ningún dato sino habías rellenado todos los input puestos previamente en la pantalla, con idea de que no se inserten valores nulos

The screenshot shows a web form titled "Insertar Videojuego" with a purple background. The form contains the following fields and controls:

- Titulo:** A text input field.
- Empresa:** A text input field.
- Año de Creacion:** A date picker showing "miércoles, 1 de junio de 2022".
- Genero:** A text input field.
- Plataforma:** A text input field.
- Multijugador:** A radio button group with options "SI" and "NO".
- INSERTAR:** A large blue button to submit the form.
- VOLVER:** A smaller blue button in the bottom left corner.

Aquí vemos la query usada para la inserción:

```
void insertButton_Click(object sender, EventArgs e)
{
    InsertTitulo.Text.Length > 0 && InsertEmpresa.Text.Length > 0 && InsertFecha.Value.Year > 0 && InsertPlataforma.Text.Length > 0
    onexion.queryF("INSERT INTO[pepe].[video_games]([Titulo],[Empresa],[Anyo],[Genero],[Plataforma],[Multijugador],[Puntuacion])" +
        " VALUES('"+InsertTitulo.Text+"','"+InsertEmpresa.Text+"','"+InsertFecha.Value.Year.ToString()+"','"+InsertGenero.Text+"',"+
        "+InsertPlataforma.Text+"+',true',0); ");
    MessageBox.Show("Insertado con éxito");
}
```


Aquí vemos una prueba de inserción:

Título: Ataque de los quesos

Empresa: QuesoSL

Año de Creacion: jueves , 17 de marzo de 2022

Genero: queseria

Plataforma: playsqueso

Multijugador: ☒ SI ☐ NO

Y aquí vemos el resultado:

34	Assassin's Creed: Altar's Chronicles	Ubisoft	2006	Action	Nintendo DS	0	0.3770
35	Astonishia Story	Ubisoft	2006	Role-Playing (RPG),Strategy	Sony PSP	0	8.4893
36	Ataque de los quesos	QuesoSL	2022	queseria	playsqueso	1	0.0000
37	ATV Offroad Fury: Blazin' Trails	Sony	2005	Racing / Driving,Sports	Sony PSP	1	0.9955
38	ATV Quad Frenzy	Ni II I	2005	Racing / Driving,Sports	Nintendo DS	0	4.0033

Los datos relacionados con las notas tuve problemas al introducirlo y al cabo del tiempo decidí que lo mas propio era quitarlo.

La pagina de Buscar tiene una interfaz mucho mas simple para que quepan de manera holgada los datos que uno busca

aquí vemos los datos sin buscar

Busca la Informacion de un juego:

Título:

Empresa:

Año de Creacion:

Genero:

Plataforma:

Puntuacion:

Y aquí con los resultados introducidos:

Busca la Informacion de un juego:

Título: *Age of Empires: Mythologies*

Empresa: *THQ*

Año de Creacion: *2008*

Genero: *Strategy*

Plataforma: *Nintendo DS*

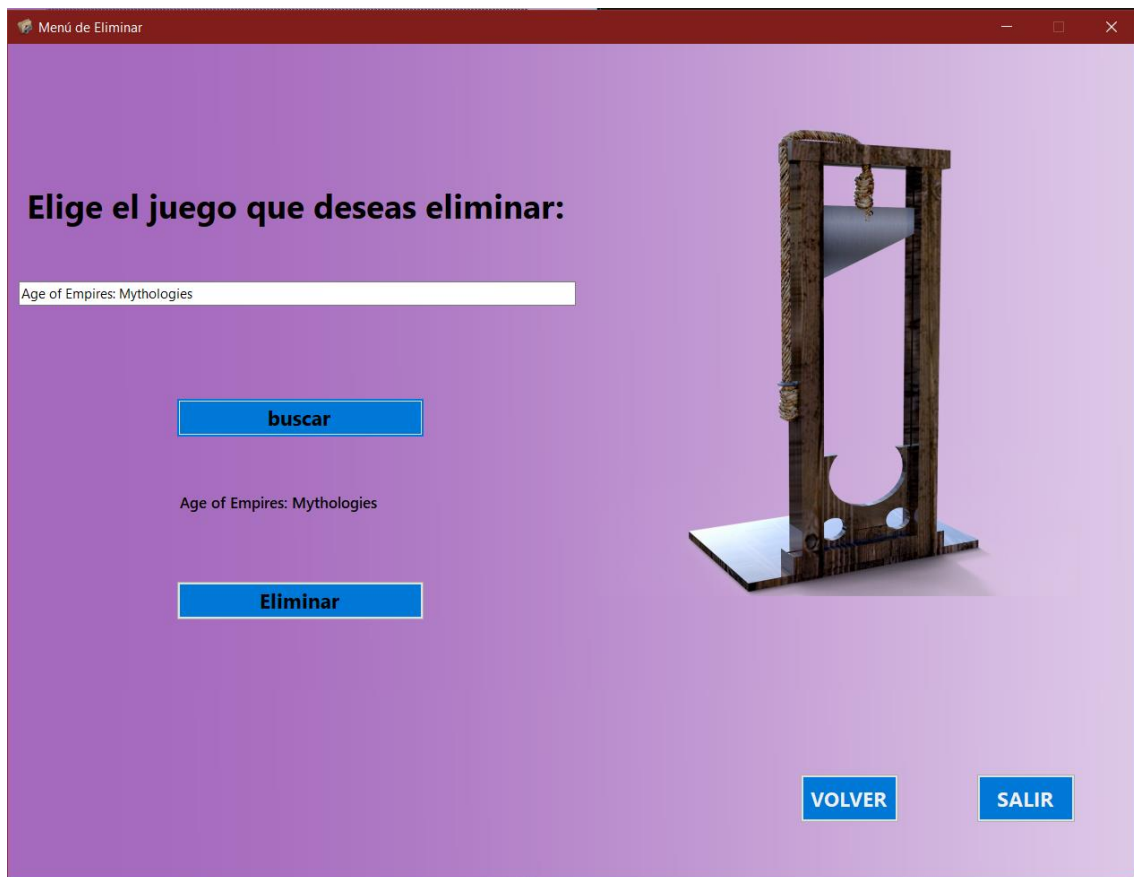
Puntuacion: *2,0782*

Se efectúan distintas queries que hacen que se definan los campos a buscar, además es robusto con distintos mensajes de error

```
if (BuscarJuegoText.Text != null)
{
    try
    {
        //for (int i = 0; i < 5; i++)
        //{
        //    Label Titulo = this.Controls.Find("info" + (i + 1), true).FirstOrDefault() as Label;
        //    Titulo.Text = (Conexion.queryF("SELECT TOP (1) [Titulo] FROM[BDDVideojuegos].[pepe].[video_games] orde; ")[i].ToString());
        //}

        info1.Text = (Conexion.queryF("SELECT TOP(1) * FROM[BDDVideojuegos].[pepe].[video_games] where Titulo = '" + BuscarJuegoText.Text + "'"")[0].ToString());
        info2.Text = (Conexion.queryF("SELECT TOP(1) * FROM[BDDVideojuegos].[pepe].[video_games] where Titulo = '" + BuscarJuegoText.Text + "'"")[1].ToString());
        info3.Text = (Conexion.queryF("SELECT TOP(1) * FROM[BDDVideojuegos].[pepe].[video_games] where Titulo = '" + BuscarJuegoText.Text + "'"")[2].ToString());
        info4.Text = (Conexion.queryF("SELECT TOP(1) * FROM[BDDVideojuegos].[pepe].[video_games] where Titulo = '" + BuscarJuegoText.Text + "'"")[3].ToString());
        info5.Text = (Conexion.queryF("SELECT TOP(1) * FROM[BDDVideojuegos].[pepe].[video_games] where Titulo = '" + BuscarJuegoText.Text + "'"")[4].ToString());
        info7.Text = (Conexion.queryF("SELECT TOP(1) * FROM[BDDVideojuegos].[pepe].[video_games] where Titulo = '" + BuscarJuegoText.Text + "'"")[6].ToString());
    }
    catch (Exception ex)
    {
        MessageBox.Show("Ese juego no existe");
    }
}
else
{
    MessageBox.Show("Introduce un valor valido");
}
```

La siguiente pagina es la pantalla de eliminar esta dispone de una imagen png de una guadaña para representar la eliminación del registro, posee un label y un botón para asegurarse que el titulo existe.

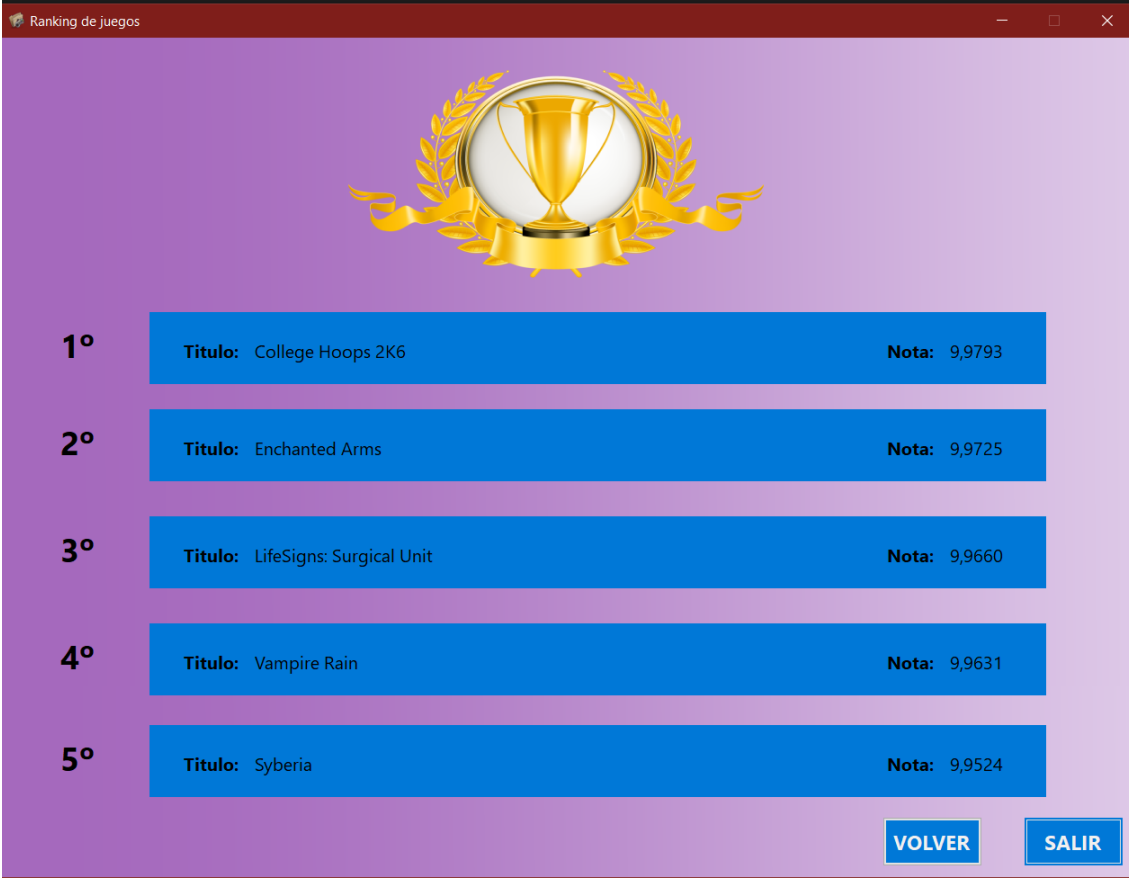


Aquí vemos la consulta de la eliminación de datos

```
1 referencia
private void eliminarJuegoText_TextChanged(object sender, EventArgs e)
{
}

1 referencia
private void button1_Click(object sender, EventArgs e)
{
    if (eliminarJuegoText.Text != null)
    {
        Conexion.queryF("DELETE FROM[pepe].[video_games] WHERE titulo = '" + eliminarJuegoText.Text + "';");
        MessageBox.Show("eliminado con exito");
    }
    else
    {
        MessageBox.Show("Escribe algo en el texto");
    }
}
```

La ultima pagina de la base de datos de videojuegos fue el ranking, cree un insert para generar números aleatorios a todos los videojuegos y crear la columna puntuación, esto genero 1200 puntuaciones entre 1 y 10, estas puntuaciones fueron ordenadas para generar un top 5 con las mejores puntuaciones de todas dando como resultado algo así:



The screenshot shows a window titled 'Ranking de juegos' with a purple background. At the top center is a gold trophy icon surrounded by laurel leaves. Below it is a table with 5 rows, each representing a game's rank. The table has two columns: 'Titulo' and 'Nota'. At the bottom right are two buttons: 'VOLVER' and 'SALIR'.

1°	Titulo: College Hoops 2K6	Nota: 9,9793
2°	Titulo: Enchanted Arms	Nota: 9,9725
3°	Titulo: LifeSigns: Surgical Unit	Nota: 9,9660
4°	Titulo: Vampire Rain	Nota: 9,9631
5°	Titulo: Syberia	Nota: 9,9524

Vemos que los juegos puestos en este top 5 no son de los mejores, pero esto es debido al mundo del azar.

A continuación presentaré el código usado para la generación de este programa

Una de las cualidades del código fue el descubrimiento de una función nueva (find) que buscaba directamente en la vista algo con el nombre parecido. Gracias a eso pude hacer un for que generase un top 5 sin tener que sobrecargar el código con muchas queries:

```
form.ShowDialog();
}

1 referencia
private void Ranking_Load(object sender, EventArgs e)
{
    for (int i = 0; i < 5; i++)
    {
        Label Titulo = this.Controls.Find("Puesto" + (i + 1), true).FirstOrDefault() as Label;
        Titulo.Text = (Conexion.queryF("SELECT TOP (10) [Titulo] FROM [BBDDVideojuegos].[pepe].[video_games]" +
            " order by Puntuacion desc; ")[i].ToString());
        Label TituloN = this.Controls.Find("Nota" + (i + 1), true).FirstOrDefault() as Label;
        TituloN.Text = (Conexion.queryF("SELECT TOP (10) [Puntuacion] FROM [BBDDVideojuegos].[pepe].[video_games]" +
            " order by Puntuacion desc; ")[i].ToString());
    }
}
```

Con esto finaliza el desarrollo del programa.

Medios materiales usados

- PC con Windows 10 de sobremesa
- PC con Windows 10 portatil
- Visual Studio 2020
- SQL Server 2019
- Photopea
- Telegram
- Github
- Chocolately
- Discord
- Microsoft Word
- Microsoft Excel
- freepng.es

Bibliografía

- www.wikipedia.com
- https://www.youtube.com/watch?v=IY1Z0Hgo288&list=LL&index=1&ab_channel=Programaci%C3%B3nDesdeCero
- www.github.com
- www.photopea.com

Apéndice

Para la instalación de este programa es necesario la instalación de visual studio y la instalación de SqlServer se debería importar la solución del programa y la inserción del archivo Excel que se encuentras en el a carpeta del programa. Una vez realizado eso no debería haber problema para su ejecución.

Quiero dar las gracias a mi familia y amigos por el apoyo brindado a la hora de realizar este trabajo.