

Backend

Abrir la solución en Visual Studio, y modificar el archivo de configuración “appsetting.json”

“Default”:

“server=(localdb)\\blazorcurso;database=VideoGames;user=root;password=19690149;TrustServerCertificate=Yes;”

```
5  Microsoft.AspNetCore.Hosting : warning
6  }
7  },
8  "ConnectionStrings": {
9    "Default": "server=(localdb)\\blazorcurso;database=VideoGames;user=root;password=19690149;TrustServerCertificate=Yes;"
10 },
11 "AllowedHosts": "*"
12 }
13 }
```

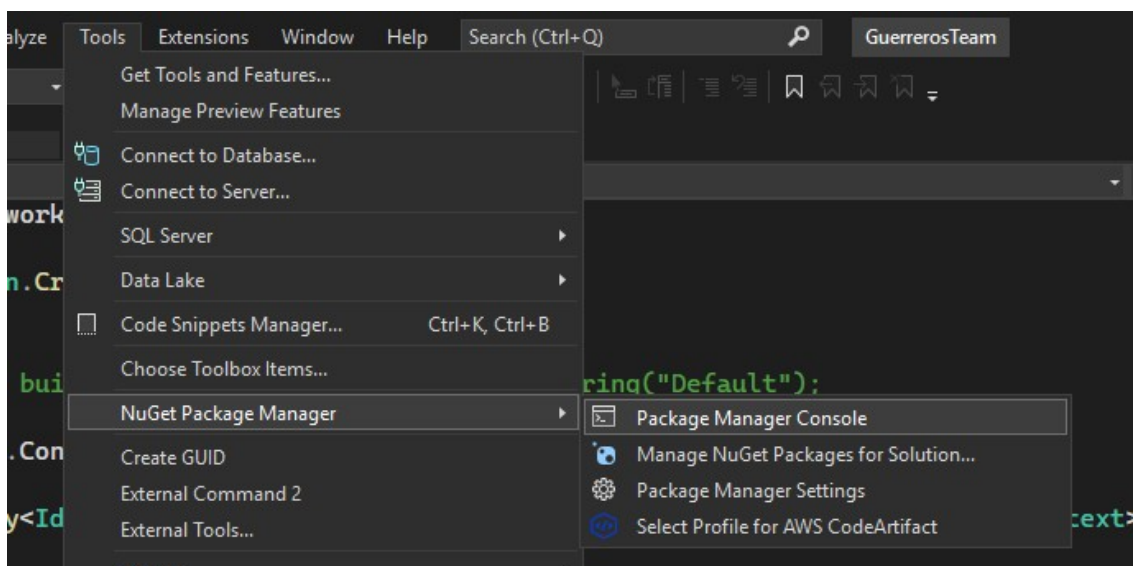
En la línea 9 se modifica la cadena de conexión para cambiar el nombre del servidor de la base de datos en SQL Server, la tabla, el usuario y la contraseña por el servidor local del equipo.

Nota: Si desea usar una base de datos MySQL, agregar el puerto de la base de datos.

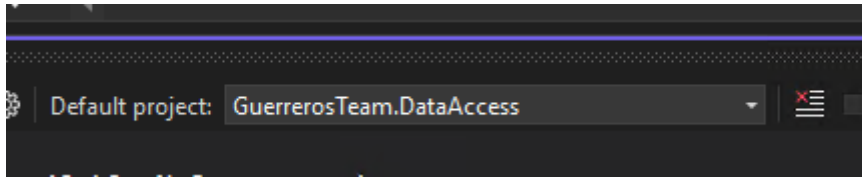
En program.cs, se ha configurado para usar SQL Server, si desea usar MySQL, se debe de cambiar la configuración:

```
appsettings.json Program.cs VideogameController.cs VideogameAppService.cs
GuerrerosTeam
7 using Microsoft.EntityFrameworkCore;
8
9 var builder = WebApplication.CreateBuilder(args);
10
11 //Connection String
12 //string connectionString = builder.Configuration.GetConnectionString("Default");
13
14 var configuration = builder.Configuration;
15
16 builder.Services.AddIdentity<IdentityUser, IdentityRole>().AddEntityFrameworkStores<VideogamesContext>();
17
18 builder.Services.AddDbContext<VideogamesContext>(options =>
19 {
20     options.UseSqlServer(configuration.GetConnectionString("Default"));
21 });
22
23 }
```

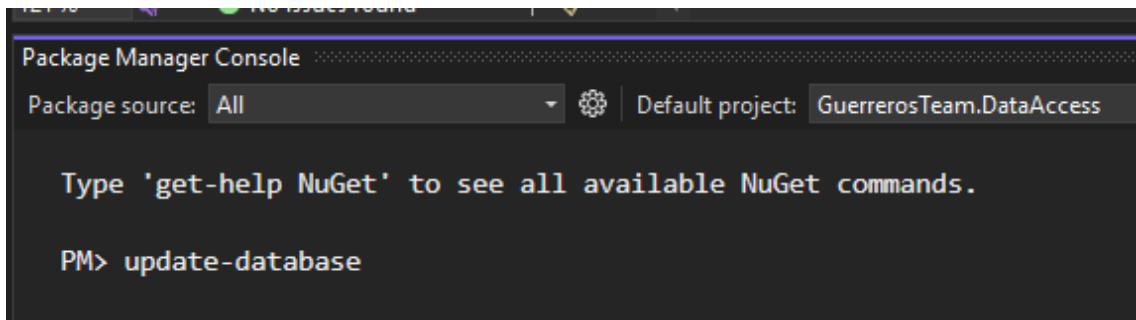
Realizar la migración a la base de datos usando Nuget package manager, para ello da clic en la sección “Tools”, seleccionas “Nuget package manager” y “Package manager console”.



Y cuando se abra la terminal, selecciona el proyecto "GuerrerosTeam.DataAccess":

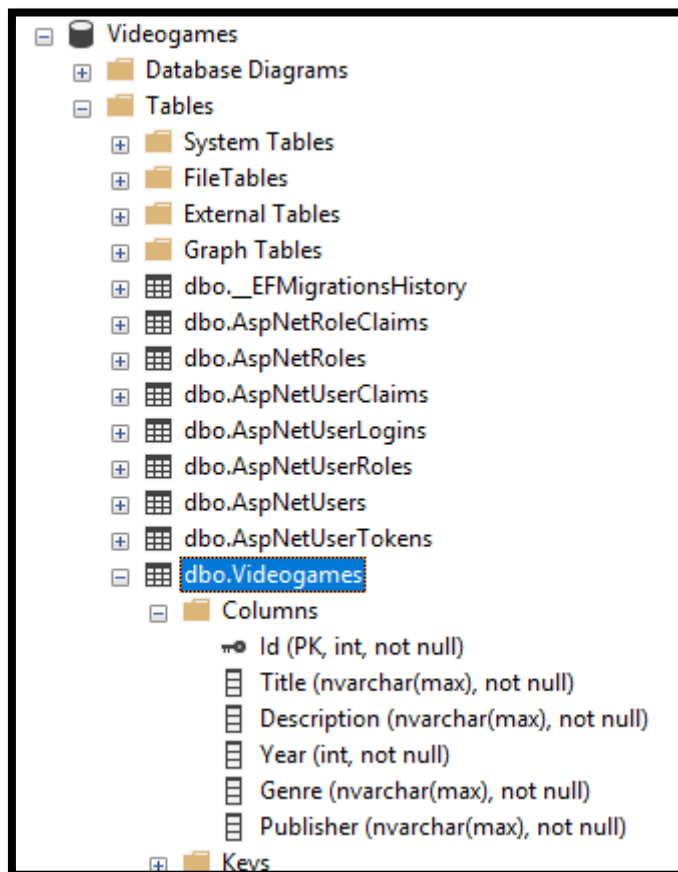


En la terminal escribe el siguiente comando y da enter: "update-database", para comenzar a migrar a la base de datos la entidad.

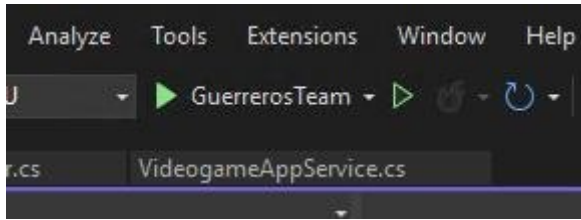


Nota: Si está usando SQL Server no debería de dar problema, si se configuro correctamente el nombre del servidor, el usuario, la contraseña y el nombre de la base de datos, si se usa MySQL, se debe de borrar la migración creada anteriormente y escribir en la terminal "add-migration Nombre_de_la_migracion" y después de que se genere el archivo de la migración escribe en la terminal "update-database".

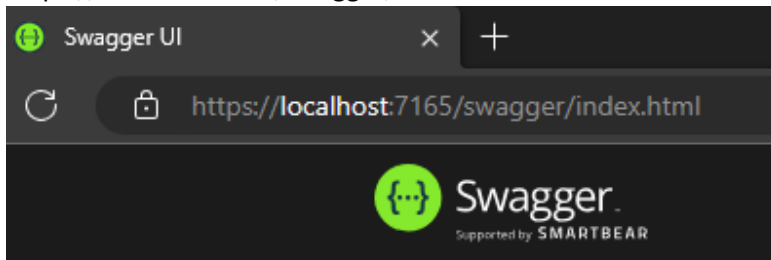
Si el proceso todo el proceso ha sido exitoso, podrá ver la estructura de la tabla en la base de datos:



Para ejecutar el proyecto da clic en el botón de run, el siguiente icono:

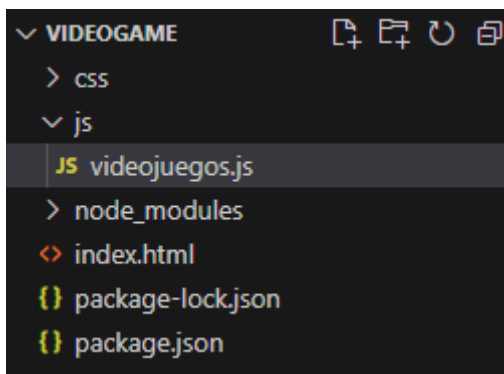


Una vez ejecutado se abre en el navegador para poder ser probado con [Swagger UI](https://localhost:7165/swagger/index.html)
<https://localhost:7165/swagger/index.html>

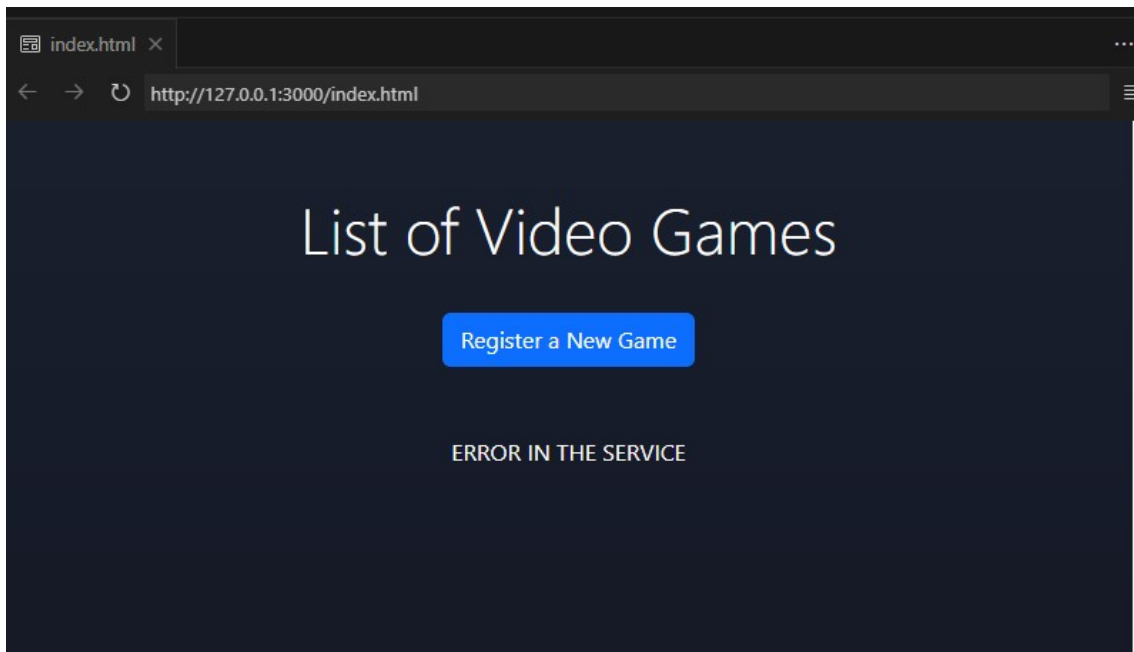


Frontend

El Front-end de la aplicación esta desarrollado en HTML CSS y JS, las peticiones son controladas por peticiones fetch de JavaScript



Al ejecutar el front end es probable que al inicio nos diga un error en el servicio, esto es debido a que el backend esta apagado o la dirección del entorno local es diferente a la que tiene por defecto

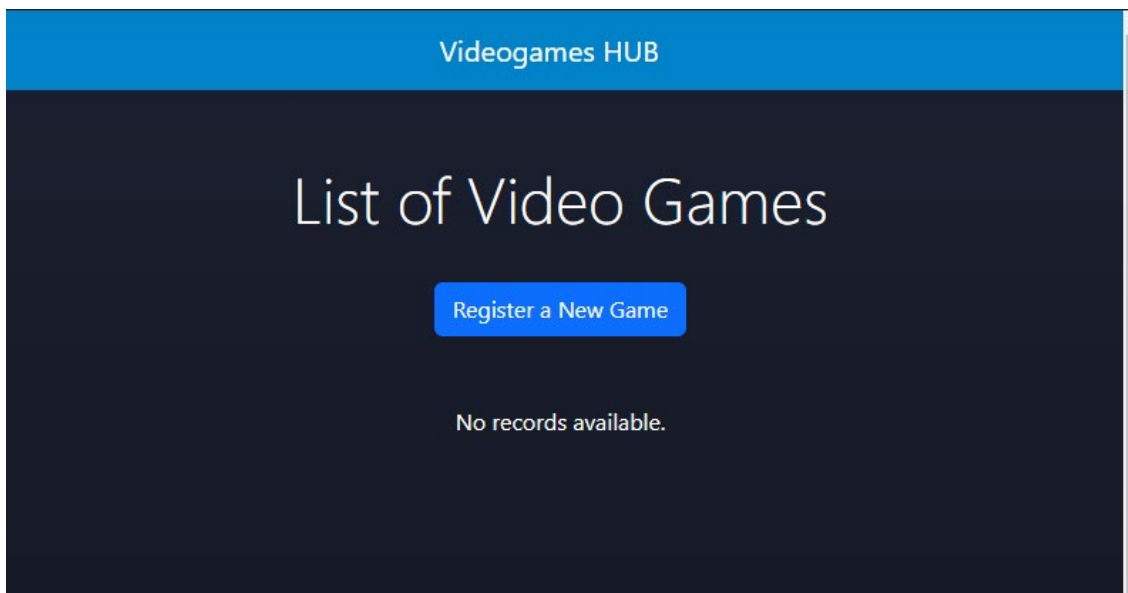


En dado caso que nos aparezca esta pantalla, lo que se tiene que hacer es cambiar la dirección que esta escrita al inicio del archivo JS

```
JS videojuegos.js X index.html
js > JS videojuegos.js > ...
1 const endpoint = "https://localhost:7165/api/videogame/";
2
```

Ya que se haya cambiado la dirección del Front-End por la correcta.

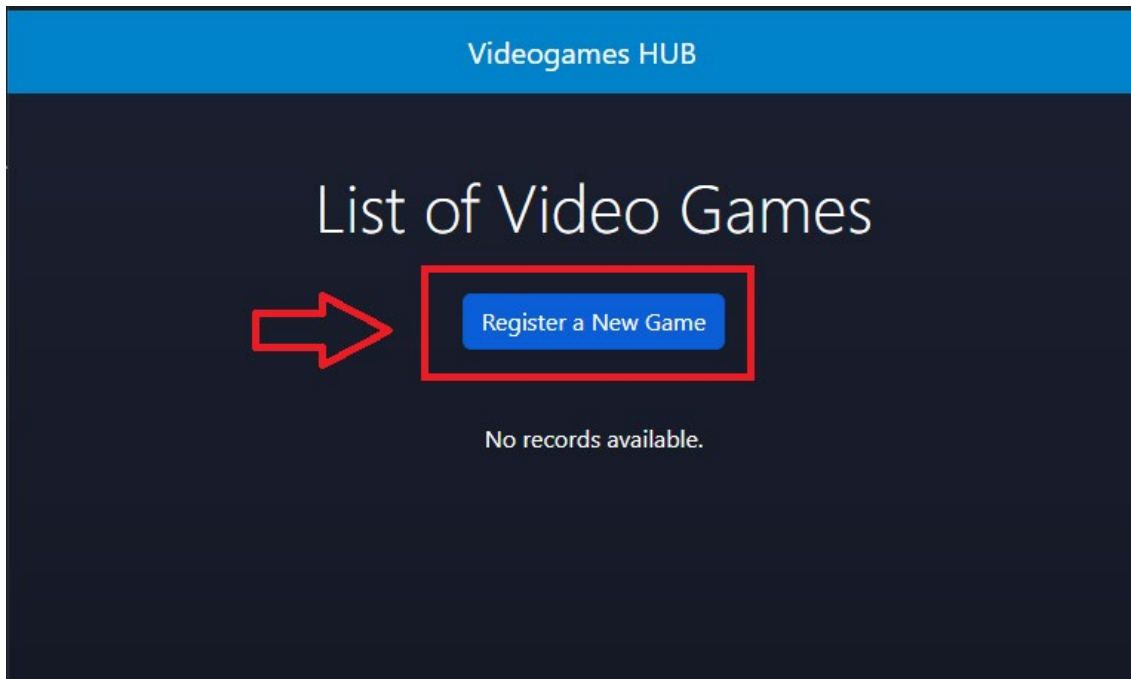
La pantalla cambiara de la siguiente forma.



Si la conexión se hizo de manera exitosa, nos aparecerá un mensaje avisando que no hay registros disponibles, o harán aparición los registros que haya en el microservicio

FUNCIONES DE LA APLICACIÓN WEB

1 Añadir archivo



Para crear un nuevo archivo basta con presionar el botón “Register a New Game”, al hacer click en el aparecerá el siguiente dialogo.

Videogames Hub

Add Video Game

➡ ✕

Title

Description

Year

Genre

Publisher

Register

➡

➡

➡

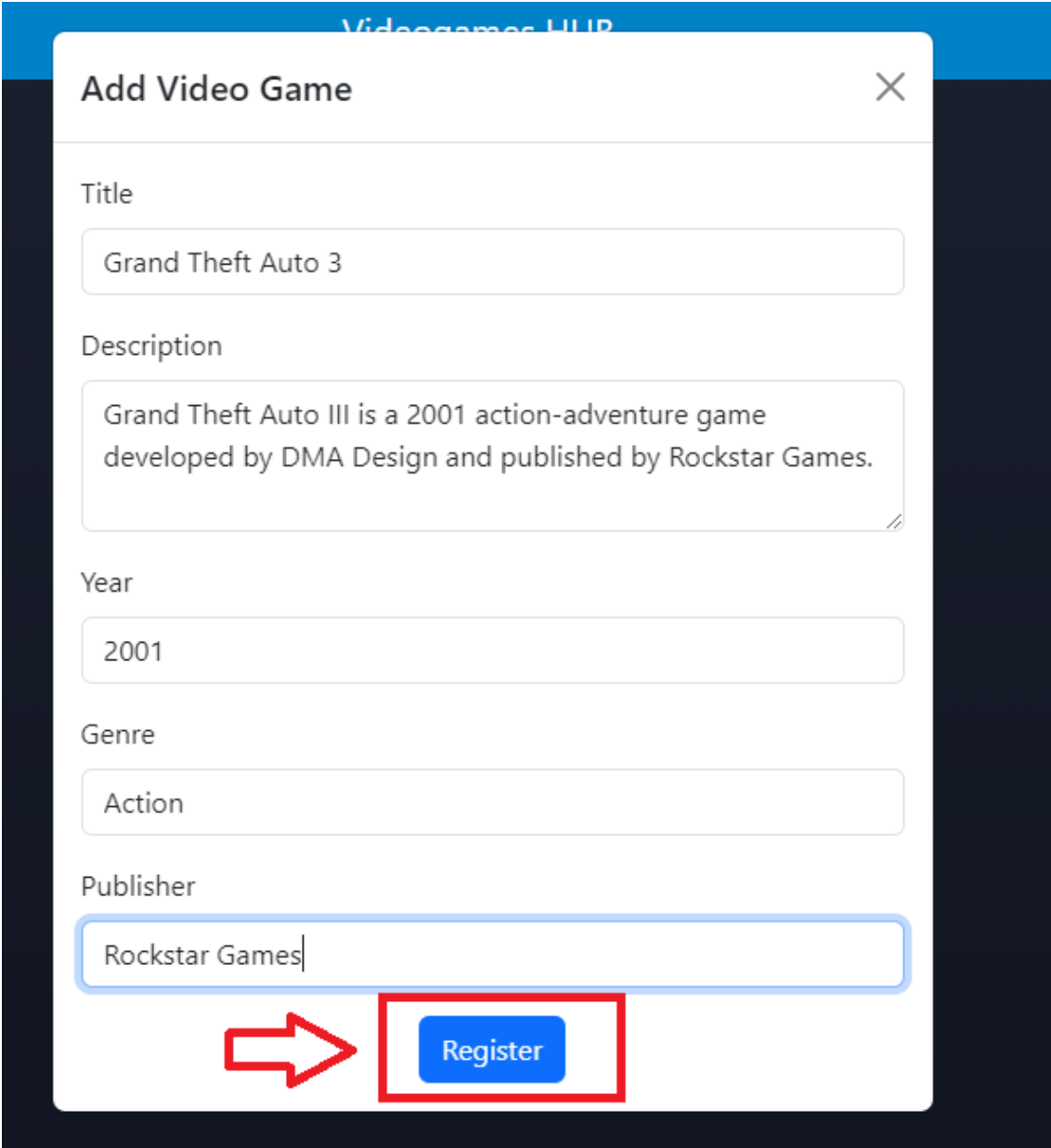
➡

➡

Aquí hay dos opciones, optar por cerrar simplemente el Dialogo, presionando la X, o
Rellenar el formulario y registrar un video juego

***NOTA: TODOS LOS CAMPOS SON REQUERIDOS**

2. Editar Registro



Add Video Game [X]


Title
Grand Theft Auto 3

Description
Grand Theft Auto III is a 2001 action-adventure game developed by DMA Design and published by Rockstar Games.

Year
2001

Genre
Action

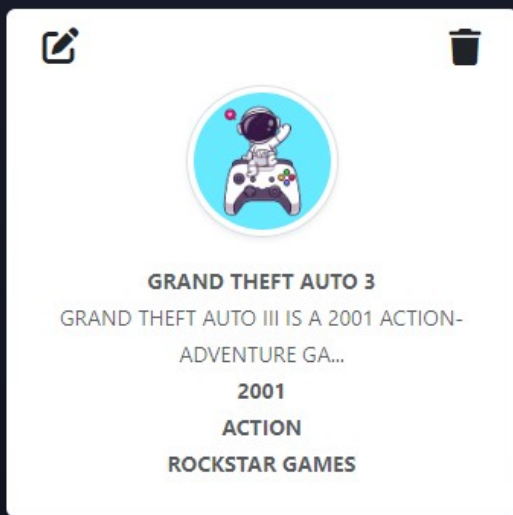
Publisher
Rockstar Games

 **Register**

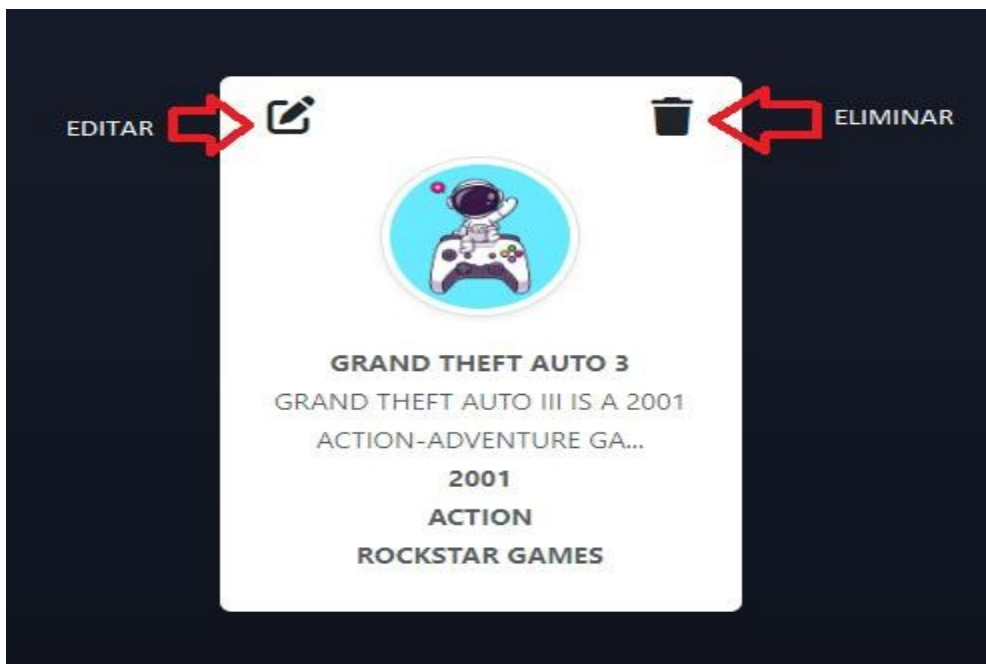
Al hacer click en el botón de register, se añadirá el nuevo registro y se mostrara en pantalla

List of Video Games

Register a New Game

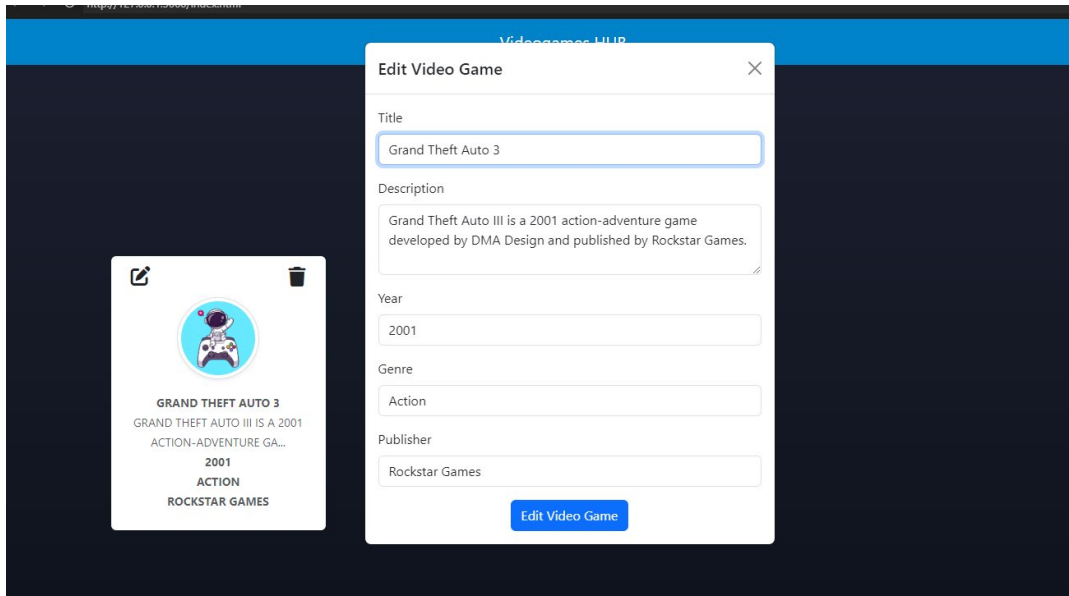


Funciones de la tarjeta de presentación.



La tarjeta de presentación cuenta con dos funciones, las cuales son Editar y Eliminar, al hacer uso de ellas se eliminará u editará el registro que aparece en la tarjeta.

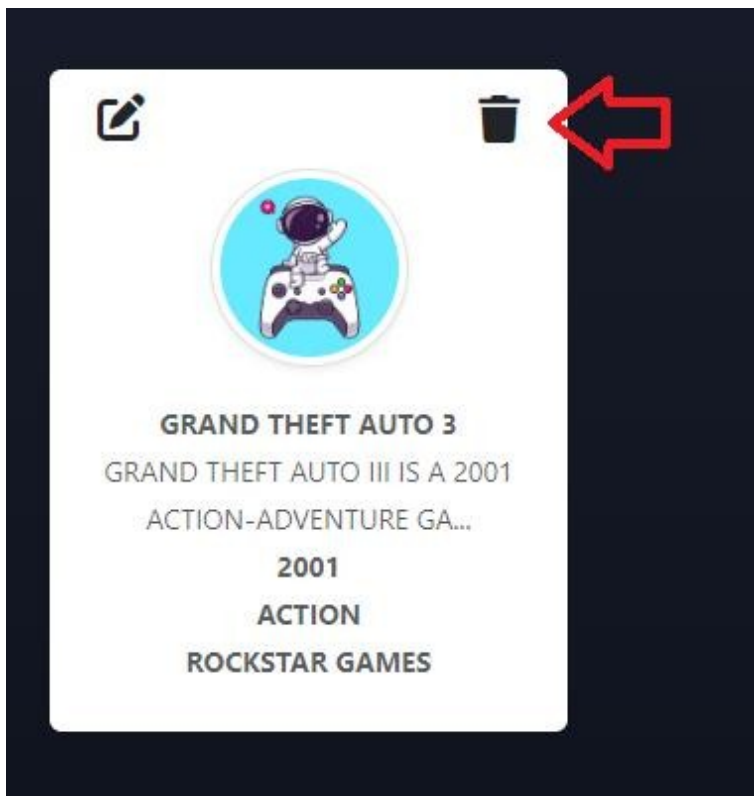
Al hacer click en Editar, haya aparición un dialogo con los datos del registro, el cual nos permitirá editarlos y registrar los cambios en el registro.



Este dialogo podrá desaparecer simplemente haciendo click en la x.

3. Eliminar registro

Si se hace click en el icono de eliminar, se recargará la página y se eliminara el registro.



Integración de OIDC(OpenId Connected Discovery) con Azure Active Directory

Se ingresa a Azure con la cuenta de Microsoft para poder registrar una aplicación que servirá para poder autenticar a los usuarios que accedan a la API, se pone un nombre a la aplicación y en url se selecciona la plataforma web y se le puede poner una url como <http://localhost:5000>, por ejemplo.

[Inicio](#) > [Directorio predeterminado](#) | [Registros de aplicaciones](#) >

Registrar una aplicación

* Nombre

Nombre para mostrar accesible por los usuarios de esta aplicación. Se puede cambiar posteriormente.

Tipos de cuenta compatibles

¿Quién puede usar esta aplicación o acceder a esta API?

- ☒ Solo cuentas de este directorio organizativo (solo de Directorio predeterminado: inquilino único)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)
- ☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant) and personal Microsoft accounts (e.g. Skype, Xbox)
- ☐ Solo cuentas personales de Microsoft

[Ayudarme a elegir...](#)

URI de redirección (opcional)

Devolveremos la respuesta de autenticación a esta dirección URI después de autenticar correctamente al usuario. Este dato es opcional y se puede cambiar más tarde, pero se necesita un valor para la mayoría de los escenarios de autenticación.

Web

▼

por ejemplo, <https://example.com/auth> ✓

Una vez creado el registro en el panel general se pueden ver los datos que se va a ocupar para poder autenticar un usuario

Microsoft Azure

Actualización

Buscar recursos, servicios y más

[Inicio](#) > [Directorio predeterminado](#) | [Registros de aplicaciones](#) >

openid-test

🔗 ...

🔍 Buscar

«

🗑️ Eliminar

🌐 Puntos de conexión

📄 Características en versión preliminar

🏠 Información general

🚀 Inicio rápido

🛠️ Asistente para integración

Administrar

🏷️ Personalización de marca y propiedades

🔄 Autenticación

🔑 Certificados y secretos

🔧 Configuración de token

👁️ Permisos de API

🔗 Exponer una API

^ Información esencial

Nombre para mostrar : [openid-test](#)

Id. de aplicación (cliente) : 953827bc-9226-4a93-8e91-96c4092a9c07

Identificador de objeto : 10883164-db34-451f-8a15-2952cc751aff

Id. de directorio (inquilino) : 9309e85f-f7dc-4cd2-8e0e-721456d8b292

Tipos de cuenta compati... : [Solo mi organización](#)

📘

Starting June 30th, 2020 we will no longer add any new features to Azure Active Directory. All new features will need to be upgraded to Microsoft Authentication Library (MSAL) and Microsoft Graph.




Introducción

Documentación


Después se selecciona Autenticación para marcar al casilla de Tokens de acceso, para poder


autenticar el acceso y guardar cambios.


[Inicio](#) > [Directorio predeterminado](#) | [Registros de aplicaciones](#) > [openid-test](#)


 **openid-test** | Autenticación  

Buscar


 ¿Tiene algún comentario?


 Información general


 Inicio rápido


 Asistente para integración


Administrar


 Personalización de marca y propiedades


 Autenticación

 Certificados y secretos

 Configuración de token

 Permisos de API

 Exponer una API

 Roles de aplicación

Flujos de concesión implícita e híbridos

Solicite un token directamente desde el punto de conexión de autorización. Si la aplicación tiene una arquitectura de página única (SPA) y no usa el flujo de código de autorización, o si invoca una API web mediante JavaScript, seleccione los tokens de acceso y los tokens de id. Para aplicaciones web de ASP.NET Core y otras aplicaciones web que usen la autenticación híbrida, seleccione solo los tokens de id. [Obtenga más información sobre los tokens.](#)

Seleccione los tokens que quiera que emita el punto de conexión de autorización:

☒ Tokens de acceso (usados para flujos implícitos)

☐ Tokens de id. (usados para flujos híbridos e implícitos)

Tipos de cuenta compatibles

¿Quién puede usar esta aplicación o acceder a esta API?

☒ Solo cuentas de este directorio organizativo (solo de Directorio predeterminado: inquilino único)

☐ Accounts in any organizational directory (Any Microsoft Entra ID tenant - Multitenant)

[Ayudarme a decidir...](#)

En el proyecto Web API se configura el servicio de autenticación para la web API por medio de Microsoft Identity, para proteger los endpoints.

```
//OIDC Azure
builder.Services.AddMicrosoftIdentityWebApiAuthentication(configuration);
```

En appsettings.json se agrega una configuración para especificar la ruta de autenticación estándar de Azure, se establece el ClientId de la aplicación registrada en Azure, el TenantId de la aplicación, y el recurso que va a acceder por medio del token.

```
{
  "AzureAd": {
    "Instance": "https://login.microsoftonline.com/",
    "ClientId": "953827bc-9226-4a93-8e91-96c4092a9c07",
    "TenantId": "9309e85f-f7dc-4cd2-8e0e-721456d8b292",
    "Audience": "953827bc-9226-4a93-8e91-96c4092a9c07"
  },
  "AllowedHosts": "*"
}
```

Regresando al program.cs, se agrega una verificación simple para saber si el usuario que realiza la solicitud esta autenticado, si el usuario no esta autenticado, se establece el código 401 para indicar que no esta autorizado, en caso contrario que el usuario si esta autenticado, puede acceder al endpoint de la solicitud.

```
app.UseAuthentication();

app.Use(async (context, next) =>
{
    //verifica si el usuario que ha realizado la solicitud no está autenticado
    if (!context.User.Identity?.IsAuthenticated ?? false)
    {
        // Esto establece el código de estado de respuesta HTTP en 401, que indica "No autorizado"
        context.Response.StatusCode = 401;
        await context.Response.WriteAsync("Not Authenticated!!");
    }
    // pasar la solicitud al siguiente
    else await next();
});

//app.UseAuthorization();
```

Para obtener la autorización de la API, es necesario iniciar sesión con una cuenta de Microsoft,

para ello se debe de mandar en el navegador los siguientes parámetros:

```
https://login.microsoftonline.com/9309e85f-f7dc-4cd2-8e0e-721456d8b292/oauth2/authorize?
client_id=953827bc-9226-4a93-8e91-96c4092a9c07
&response_type=token
&redirect_uri=http%3A%2F%2Flocalhost%3A5000%2F
&resource=953827bc-9226-4a93-8e91-96c4092a9c07
&response_mode=fragment
&state=12345
&nonce=678910
```

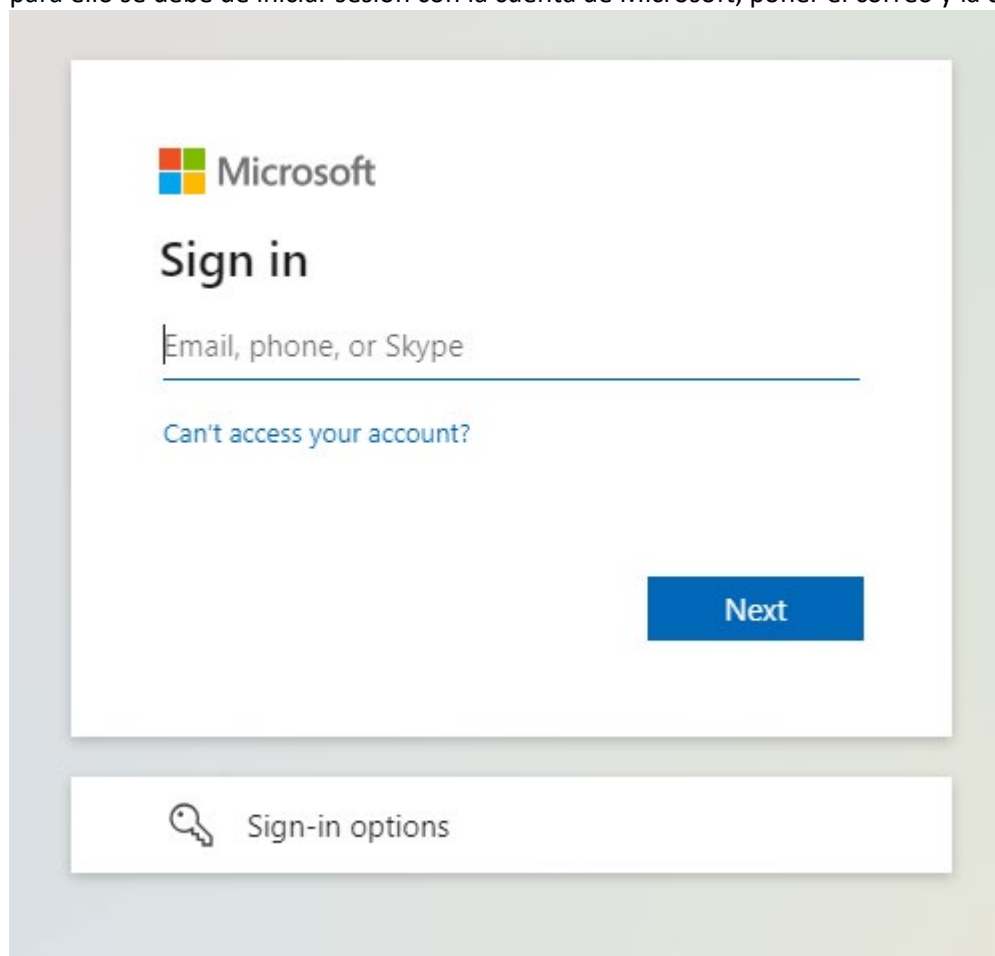
TenantId

Donde Client_id: va a llevar el id de la aplicación registrada en Azure AD y será el mismo que le pasamos a resource, en response_type, esperamos un token, y ese token, es el que se va a ocupar para poder autenticar al momento de hacer la solicitud en la API, en redirect_uri, se pone la URL de la aplicación registrada en Azure, y lo demás queda igual.

La url completa seria :

https://login.microsoftonline.com/9309e85f-f7dc-4cd2-8e0e-721456d8b292/oauth2/authorize?client_id=953827bc-9226-4a93-8e91-96c4092a9c07&response_type=token&redirect_uri=http%3A%2F%2Flocalhost%3A5000%2F&resource=953827bc-9226-4a93-8e91-96c4092a9c07&response_mode=fragment&state=12345&nonce=678910

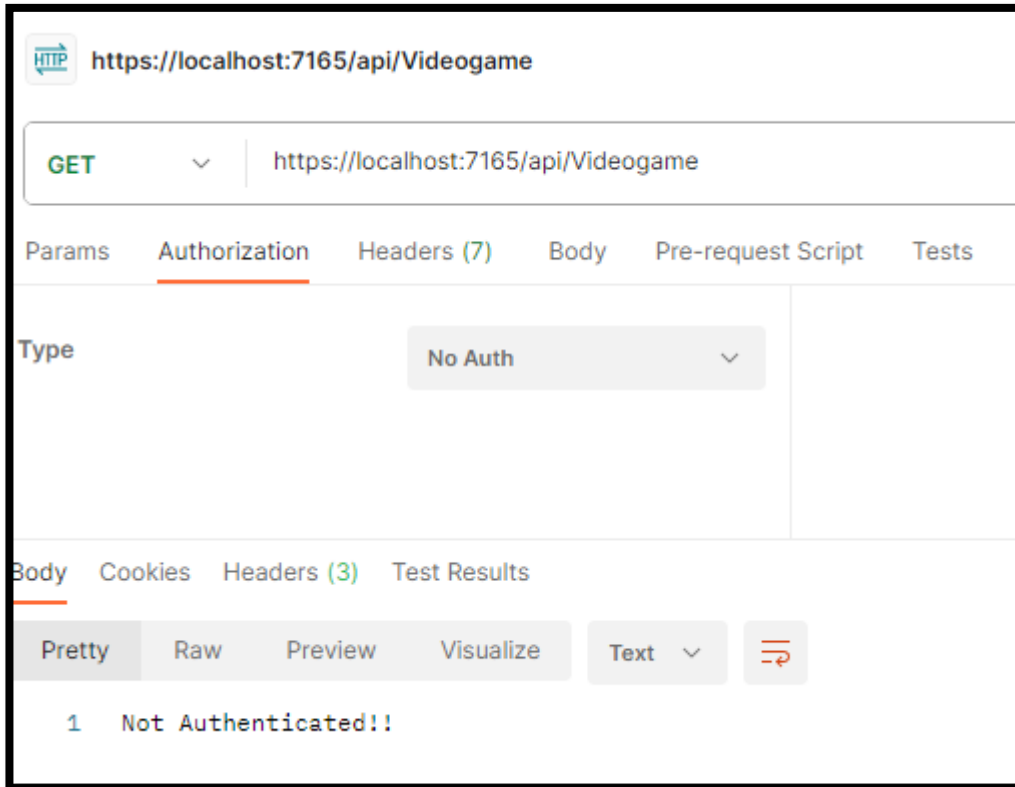
para ello se debe de iniciar sesión con la cuenta de Microsoft, poner el correo y la contraseña



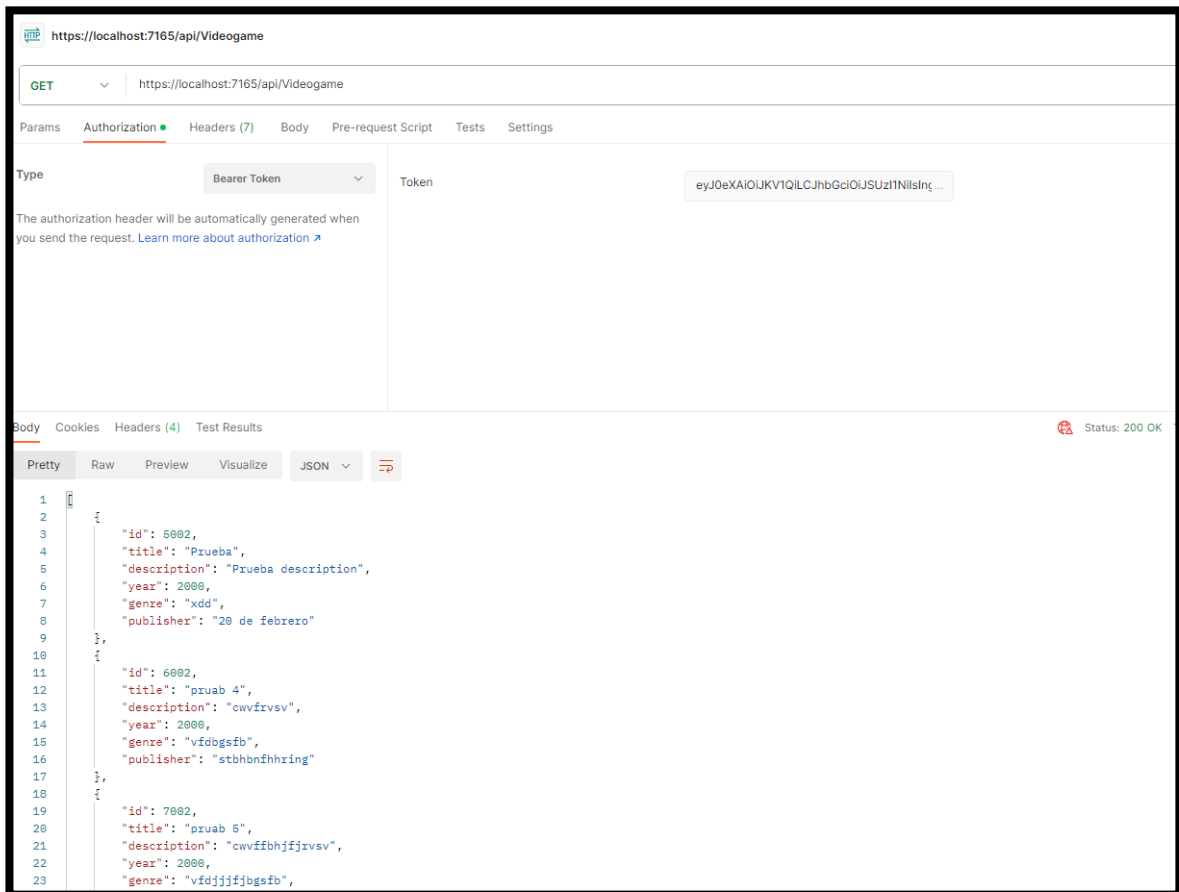
Una vez iniciado sesión, se redirecciona a la pagina que definimos en el registro de la aplicación y se obtiene el token de autorización

localhost:5000/#access_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6Ii1LSTNROW5OUjdiUm9meG1lWm9YcW

En postman accedemos a la API, pero si no hemos añadido el token, nos devuelve una respuesta de que no está autenticado



Para poder acceder a la Api se debe de seleccionar Bearer Token y pegar el token obtenido después de iniciar sesión y se envia la solicitud para que el servidor pueda verificar y acceder a la API, si es exitoso devuelve los datos que trae la API.



De esta manera se puede autenticar la identidad del usuario y dar el acceso al endpoint a través de un inicio de sesión con Microsoft.

PROYECTO ELABORADO POR EL EQUIPO GUERREROS TEAM

Susana Esparza Ramírez

Jose Angel Vazquez Torres

Joel Julián Ramírez

Cristóbal Flores

Abelardo Rodríguez Lucero

José Andrés Mata Olivares