



UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

DISEÑO TÉCNICO DE LA APLICACIÓN

Programación Web II

Docente: Enrique Tolentino García

Equipo 4

Abraham Reyes Muñiz.....1803707
Samara Díaz Hirashi.....1662578
Edgar Alejandro Niño Sánchez.....1800066
Andrea Nahomi Rosas Sánchez1803204

Grupo: 01

Abril de 2021



TIENDA DE MÚJICA

DISEÑO DE BASE DE DATOS	3
DISEÑO DE DIAGRAMA TDM	5
ENDPOINTS	10
MODELOS	7



DISEÑO DE BASE DE DATOS

Definición: *Una base de datos es un programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.*

Usando: **MySQL**

USER
Username VARCHAR(15) NOT NULL, [Name] VARCHAR(50) NOT NULL, [Password] VARCHAR(15) NOT NULL, Email VARCHAR(30) NOT NULL, PhoneNumber VARCHAR(10) NULL, Active BIT NOT NULL, PRIMARY KEY(Username)

FORMAT
Id INT NOT NULL IDENTITY(1,1), TypeCode VARCHAR(3) NOT NULL, [Type] VARCHAR(15) NOT NULL, PRIMARY KEY(Id)

GENRE
Id INT NOT NULL IDENTITY(1,1), GenreName VARCHAR(30) NOT NULL, PRIMARY KEY(Id)

ARTIST
Id INT NOT NULL IDENTITY(1,1), StageName VARCHAR(50) NOT NULL, RealName VARCHAR(50) NOT NULL, [Description] VARCHAR(50) NOT NULL, Active BIT NOT NULL, PRIMARY KEY(Id)

PRODUCT
IId INT NOT NULL IDENTITY(1,1), [Name] VARCHAR(50) NOT NULL, Price DECIMAL NOT NULL, [Description] VARCHAR(MAX) NULL, Active BIT NOT NULL, IdGenre INT NOT NULL, CONSTRAINT FK_ProductGenre FOREIGN KEY (IdGenre) REFERENCES Genre(Id), PRIMARY KEY(Id)

PRODUCTFORMAT
IId INT NOT NULL IDENTITY(1,1), IdProduct INT NOT NULL, IdFormat INT NOT NULL, CONSTRAINT FK_ProductFormatProduct FOREIGN KEY(IdProduct) REFERENCES Product, CONSTRAINT FK_ProductFormatFormat FOREIGN KEY(IdFormat) REFERENCES [Format], PRIMARY KEY(Id)



DISEÑO DE BASE DE DATOS

Definición: *Una base de datos es un programa capaz de almacenar gran cantidad de datos, relacionados y estructurados, que pueden ser consultados rápidamente de acuerdo con las características selectivas que se deseen.*

Usando: **MySQL**

●	SHOPPING CART	●
	Id INT NOT NULL IDENTITY(1,1), IdProduct INT NOT NULL, Username VARCHAR(15) NOT NULL, CONSTRAINT FK_ShoppingCartProduct FOREIGN KEY (IdProduct) REFERENCES Product(Id), CONSTRAINT FK_ShoppingCartUser FOREIGN KEY (Username) REFERENCES [User](Username), PRIMARY KEY (Id),	

●	ARTISTPRODUCT	●
	Id INT NOT NULL IDENTITY(1,1), IdArtist INT NOT NULL, IdProduct INT NOT NULL, CONSTRAINT FK_ArtistProductArtist FOREIGN KEY (IdArtist) REFERENCES Artist(Id), CONSTRAINT FK_ArtistProductProduct FOREIGN KEY (IdProduct) REFERENCES Product(Id), PRIMARY KEY (Id)	

●	PHOTOS	●
	Id INT NOT NULL IDENTITY(1,1), [Image] VARBINARY(MAX) NOT NULL, IdProduct INT NOT NULL, CONSTRAINT FK_PhotosProduct FOREIGN KEY (IdProduct) REFERENCES Product(Id), PRIMARY KEY (Id),	

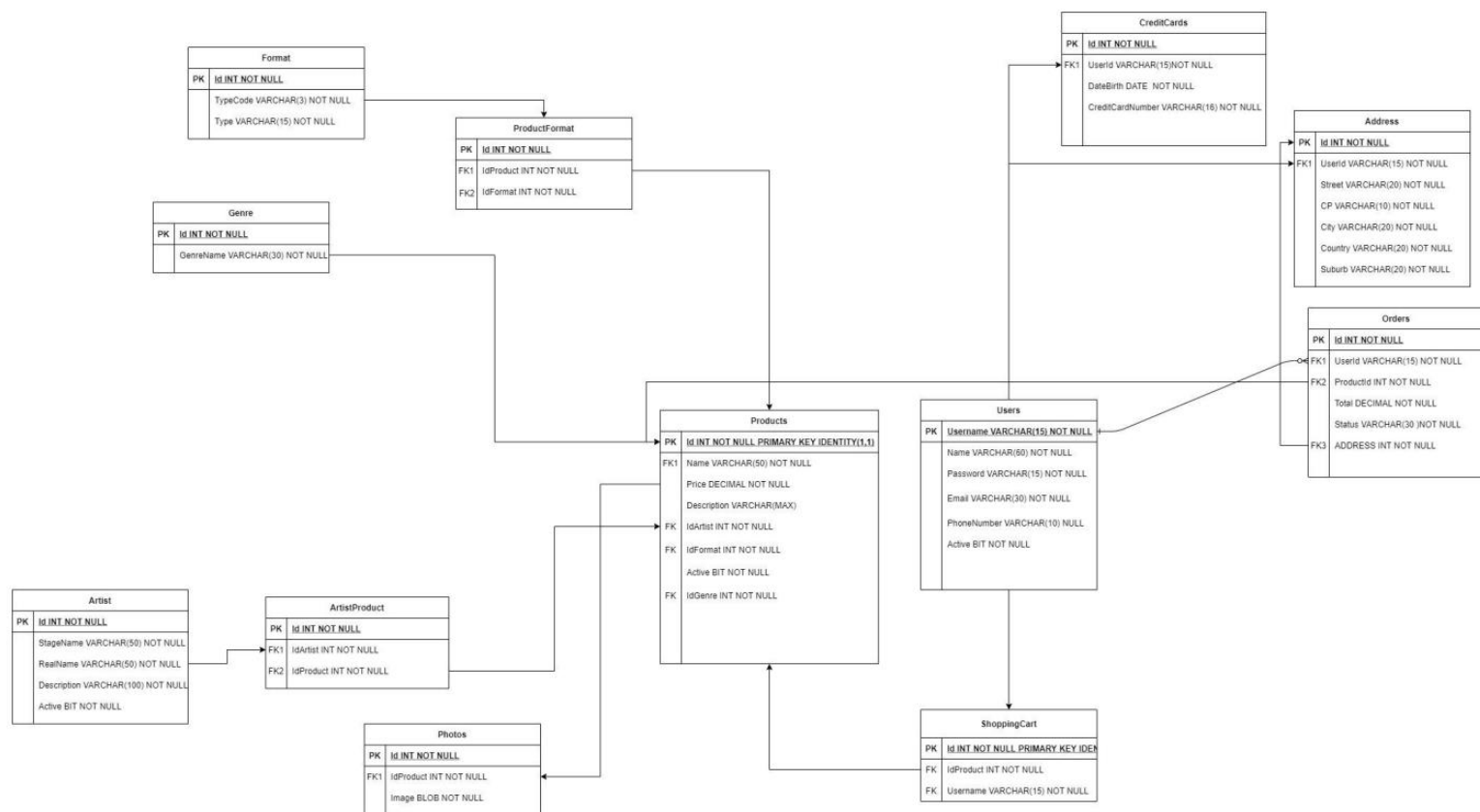
●	ADRESS	●
	Id INT NOT NULL, Street VARCHAR(20) NOT NULL, CP VARCHAR(10) NOT NULL, City VARCHAR(20) NOT NULL, Country VARCHAR(20) NOT NULL, Suburb VARCHAR(20) NOT NULL, Username VARCHAR(15) NOT NULL, CONSTRAINT FK_AddressUser FOREIGN KEY (Username) REFERENCES [ser](Username),	

●	ORDER	●
	Id INT NOT NULL, Total DECIMAL NOT NULL, [Status] VARCHAR(30) NOT NULL, IdAddress INT NOT NULL, Username VARCHAR(15) NOT NULL, IdProduct INT NOT NULL, CONSTRAINT FK_OrderUser FOREIGN KEY (Username) REFERENCES [User] (Username),	

●	ORDERPRODUCT	●
	Id INT NOT NULL, IdProduct INT NOT NULL, IdOrder INT NOT NULL, CONSTRAINT FK_OrderProductProduct FOREIGN KEY (IdProduct) REFERENCES Product(Id), CONSTRAINT FK_OrderProductOrder	



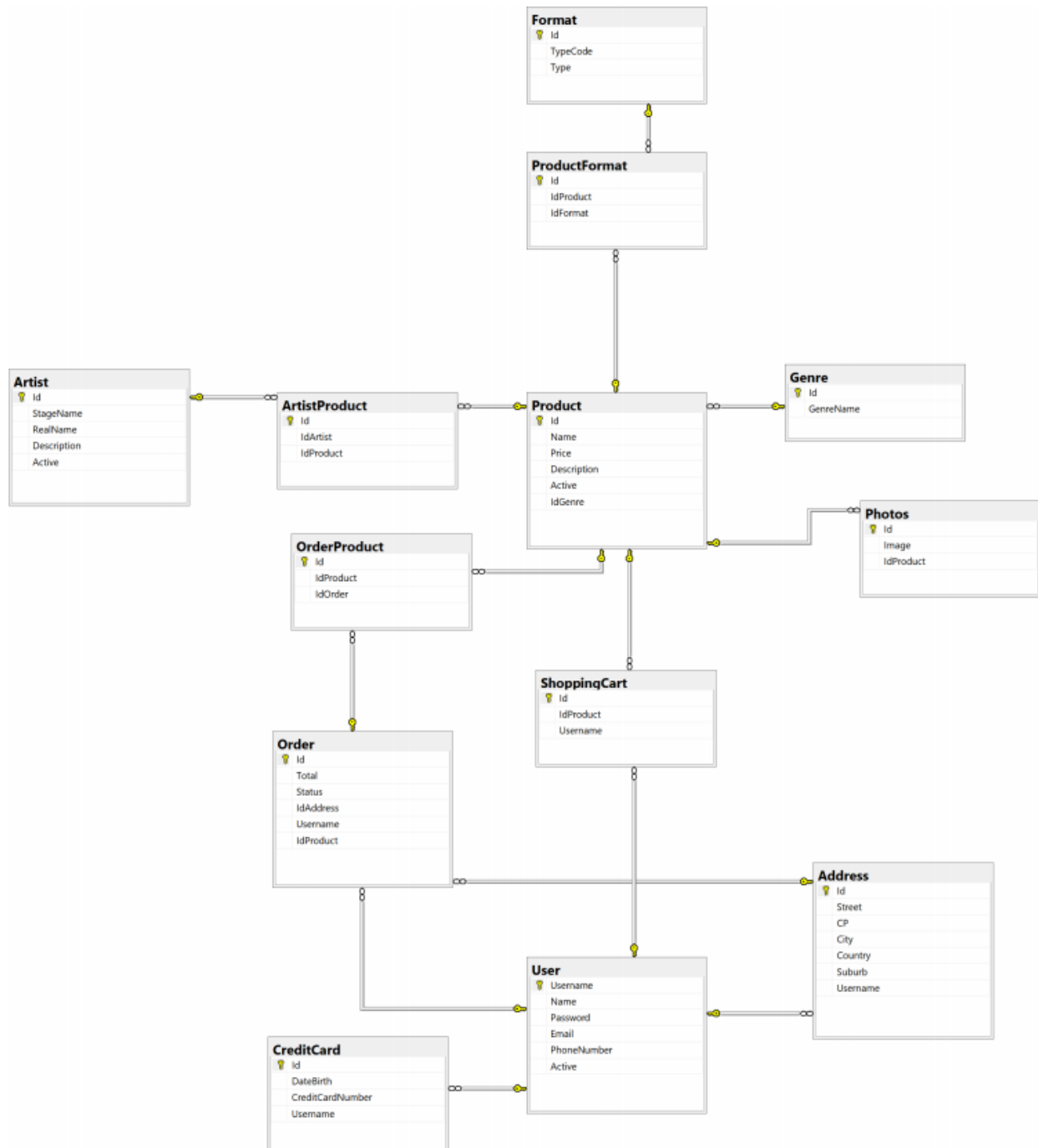
DISEÑO DE DIAGRAMA TDM





DISEÑO DE DIAGRAMA TDM

Usando: SQL





MODELOS

Usando: Visual Studio Code

Address

```
namespace TiendaDeMujica.Models
{
    3referencias
    public class Address
    {
        1referencia
        public int Id { get; set; }
        1referencia
        public string Street { get; set; }
        1referencia
        public string CP { get; set; }
        1referencia
        public string City { get; set; }
        1referencia
        public string Country { get; set; }
        1referencia
        public string Suburb { get; set; }
        1referencia
        public virtual User Username { get; set; }
        //address
    }
}
```

Artist

```
namespace TiendaDeMujica.Models
{
    2referencias
    public class Artist
    {
        1referencia
        public int Id { get; set; }
        1referencia
        public string StageName { get; set; }
        1referencia
        public string RealName { get; set; }
        1referencia
        public string Description { get; set; }
        1referencia
        public bool Active { get; set; }
    }
}
```

ArtistProduct

```
namespace TiendaDeMujica.Models
{
    2referencias
    public class ArtistProduct
    {
        1referencia
        public int Id { get; set; }
        1referencia
        public int IdArtist { get; set; }
        1referencia
        public int IdProduct { get; set; }
        1referencia
        public virtual ICollection<Artist> Artist { get; set; }
        1referencia
        public virtual ICollection<Product> Product { get; set; }
    }
}
```

CreditCard

```
namespace TiendaDeMujica.Models
{
    1referencia
    public class CreditCard
    {
        //credit
        0referencias
        public int Id { get; set; }

        0referencias
        public DateTime DateBirth { get; set; }

        0referencias
        public string CreditCardNumber { get; set; }

        0referencias
        public virtual User Username { get; set; }
    }
}
```

Format

```
namespace TiendaDeMujica.Models
{
    2referencias
    public class Format
    {
        0referencias
        public int Id { get; set; }
        0referencias
        public string TypeCode { get; set; }
        0referencias
        public string Type { get; set; }

        0referencias
        public virtual ICollection<ProductFormat> ProductFormat { get; set; }
    }
}
```

Genre

```
namespace TiendaDeMujica.Models
{
    5referencias
    public class Genre
    {
        1referencia
        public int Id { get; set; }
        0referencias
        public string GenreName { get; set; }
    }
}
```




MODELOS

Usando: Visual Studio Code

Order

```
namespace TiendaDeMujica.Models
{
    3 referencias
    public class Order
    {
        1 referencia
        public int Id { get; set; }

        1 referencia
        public decimal Total { get; set; }

        1 referencia
        public string Status { get; set; }

        0 referencias
        public virtual Address IdAddress { get; set; }

        1 referencia
        public virtual User Username { get; set; }

        //order
    }
}
```

OrderProduct

```
namespace TiendaDeMujica.Models
{
    1 referencia
    public class OrderProduct
    {
        0 referencias
        public int Id { get; set; }
        0 referencias
        public int IdProduct { get; set; }
        0 referencias
        public string IdOrder { get; set; }
        0 referencias
        public virtual Order Order { get; set; }
        0 referencias
        public virtual Product Product { get; set; }
    }
}
```

Photos

```
namespace TiendaDeMujica.Models
{
    2 referencias
    public class Photos
    {
        1 referencia
        public int Id { get; set; }
        1 referencia
        public int IdProduct { get; set; }
        1 referencia
        public string Image { get; set; }
        1 referencia
        public virtual Product Product { get; set; }
    }
}
```

Product

```
namespace TiendaDeMujica.Models
{
    4 referencias
    public class Product
    {
        0 referencias
        public int Id { get; set; }
        0 referencias
        public string Name { get; set; }
        0 referencias
        public float Price { get; set; }
        0 referencias
        public string? Description { get; set; }
        0 referencias
        public bool Active { get; set; }

        0 referencias
        public int IdGenre { get; set; }

        0 referencias
        public virtual ICollection<ShoppingCart> ShoppingCart { get; set; }
        0 referencias
        public virtual ICollection<OrderProduct> OrderProduct { get; set; }
        0 referencias
        public virtual ICollection<ArtistProduct> ArtistProduct { get; set; }
        0 referencias
        public virtual ICollection<ProductFormat> ProductFormat { get; set; }
        1 referencia
        public virtual ICollection<Photos> Photos { get; set; }

        0 referencias
        public virtual Genre Genre { get; set; }
    }
}
```

ProductFormat

```
namespace TiendaDeMujica.Models
{
    public class ProductFormat
    {
        public int Id { get; set; }
        0 referencias
        public int IdProduct { get; set; }
        0 referencias
        public int IdFormat { get; set; }

        0 referencias
        public virtual Product Product { get; set; }
        0 referencias
        public virtual Format Format { get; set; }
    }
}
```

ShoppingCart

```
namespace TiendaDeMujica.Models
{
    2 referencias
    public class ShoppingCart
    {
        0 referencias
        public int Id { get; set; }
        0 referencias
        public int IdProduct { get; set; }
        0 referencias
        public string Username { get; set; }
        0 referencias
        public virtual User User { get; set; }
        0 referencias
        public virtual User Product { get; set; }
    }
}
```




MODELOS

Usando: Visual Studio Code

User

```
namespace TiendaDeMujica.Models
{
    public class User
    {
        public string Username { get; set; }

        public string Name { get; set; }
        0 referencias
        public string Password { get; set; }
        0 referencias
        public string Email { get; set; }
        0 referencias
        public string? PhoneNumber { get; set; }
        0 referencias
        public bool Active { get; set; }
        1 referencia
        public virtual ICollection<Address> Address { get; set; }
        1 referencia
        public virtual ICollection<Order> Order { get; set; }
        0 referencias
        public virtual ICollection<ShoppingCart> ShoppingCart { get; set; }
        0 referencias
        public virtual ICollection<CreditCard> CreditCard { get; set; }
    }
}
```

DbContext

```
6 referencias
public class TiendaDeMujicaDbContext : DbContext
{
    //public DbSet<> { get; set; }
    0 referencias
    public DbSet<Format> Format { get; set; }
    1 referencia
    public DbSet<Genre> Genre { get; set; }
    0 referencias
    public DbSet<ProductFormat> ProductFormat { get; set; }

    0 referencias
    public TiendaDeMujicaDbContext()
    {
    }

    0 referencias
    public TiendaDeMujicaDbContext(DbContextOptions<TiendaDeMujicaDbContext> options) : base(options) { }

    0 referencias
    protected override void OnModelCreating(ModelBuilder modelBuilder)
    {
        modelBuilder.Entity<Address>(entity =>{...});
        modelBuilder.Entity<Artist>(entity =>{...});
        modelBuilder.Entity<ArtistProduct>(entity =>{...});
        modelBuilder.Entity<CreditCard>(entity =>{...});
        modelBuilder.Entity<Format>(entity =>{...});
        modelBuilder.Entity<Genre>(entity =>{...});
        modelBuilder.Entity<Order>(entity =>{...});
        modelBuilder.Entity<OrderProduct>(entity =>{...});
        modelBuilder.Entity<Photos>(entity =>{...});
        modelBuilder.Entity<Product>(entity =>{...});
        modelBuilder.Entity<ProductFormat>(entity =>{...});
        modelBuilder.Entity<ShoppingCart>(entity =>{...});
        modelBuilder.Entity<User>(entity =>{...});
    }
}
```



ENDPOINTS

Definición

Usando: **Visual Studio Code**

Address

Ruta	<code>https://localhost:44306/api/Address/Get</code>
Descripción	Trae la dirección de todos los usuarios, Address es la tabla de direcciones de usuario.
Parámetros entrada	*No aplica*
Salida	<pre>{ "Id": "int", "Street": "string", "CP": "string", "City": "string", "Country": "string", "Suburb": "string", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/Address/Get/{id}</code>
Descripción	Trae la dirección de un usuario en específico, Address es la tabla de direcciones de usuario. El parámetro de entrada es el id del usuario.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "Street": "string", "CP": "string", "City": "string", "Country": "string", "Suburb": "string", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success"



Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"
----------------------------	--

Ruta	<i>https://localhost:44306/api/Address/Create</i>
Descripción	<i>Crea una nueva dirección(Address) para el usuario. Address es la tabla de direcciones de usuario.</i>
Parámetros entrada	<pre>{ "Street": "string", "CP": "string", "City": "string", "Country": "string", "Suburb": "string" }</pre>
Salida	<pre>{ "Id": "int", "Street": "string", "CP": "string", "City": "string", "Country": "string", "Suburb": "string", "Username": "string" }</pre>
Respuestas de éxito	200 - "Added new address successfully!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Ruta	<i>https://localhost:44306/api/Address/Update/{id}</i>
Descripción	<i>Actualiza la dirección de un usuario específico, Address es la tabla de direcciones de usuario. El parámetro de entrada es el id del usuario.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "Street": "string", "CP": "string", "City": "string", "Country": "string", "Suburb": "string", "Username": "string" }</pre>



Respuestas de éxito	200 - "Address successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Ruta	<code>https://localhost:44306/api/Address/Delete/{id}</code>
Descripción	Elimina la dirección de un usuario específico, Address es la tabla de direcciones de usuario. El parámetro de entrada es el id del usuario.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ }</pre>
Respuestas de éxito	200 - "Address successfully deleted!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Artist

Ruta	<code>https://localhost:44306/api/Artist/GetAll</code>
Descripción	Trae la información de todos los artistas, Artist es la tabla que contiene la información de los artistas.
Parámetros entrada	*No aplica*
Salida	<pre>{ "Id": "int", "StageName": "string", "RealName": "string", "Description": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 - "Success"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"



Ruta	<code>https://localhost:44306/api/Artist/Get/{id}</code>
Descripción	Trae la información del artista que coincida con el id, Artist es la tabla que contiene la información de los artistas. El parámetro de entrada es el id del artista.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "StageName": "string", "RealName": "string", "Description": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/Artist/Create</code>
Descripción	Crea un nuevo artista (Artist). Artist es la tabla que contiene la información de los artistas.
Parámetros entrada	<pre>{ "StageName": "string", "RealName": "string", "Description": "string", "Active": "bool" }</pre>
Salida	<pre>{ "Id": "int", "StageName": "string", "RealName": "string", "Description": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 – "Added new artist successfully!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/Artist/Update/{id}</code>
-------------	---



Descripción	<i>Edita un artista que ya esté creado en la base de datos (Artist). Artist es la tabla que contiene la información de los artistas. El parámetro de entrada es el id del artista.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "StageName": "string", "RealName": "string", "Description": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 – "Artist successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/Artist/Disable/{id}</i>
Descripción	<i>Deshabilita la información de un artista específico, Artist es la tabla que contiene la información de los artistas. El parámetro de entrada es el id del artista.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ }</pre>
Respuestas de éxito	200 – "Artist successfully disabled!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

ArtistProduct

Ruta	<i>https://localhost:44306/api/ArtistProduct/GetAll</i>
Descripción	<i>Trae la información de todos los productos de los artistas, ArtistProduct es la tabla que contiene los productos de los artistas.</i>
Parámetros entrada	<pre>{</pre>



	<i>*no aplica*</i> }
Salida	{ "Id": "int", "IdArtist": "int", "IdProduct": "int" }
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/ArtistProduct/Get/{id}</i>
Descripción	<i>Trae la información de un producto en específico de un artista. ArtistProduct es la tabla que contiene los productos de los artistas. El parámetro de entrada es el id del producto del artista.</i>
Parámetros entrada	{ "Id": "int" }
Salida	{ "Id": "int", "IdArtist": "int", "IdProduct": "int" }
Respuestas de éxito	200 – "Success!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/ArtistProduct/Create</i>
Descripción	<i>Crea una nueva relación entre un producto y su artista, ArtistProduct es la tabla que contiene los productos de los artistas. El parámetro de entrada es el id del producto del artista y el id del producto.</i>
Parámetros entrada	{ "IdArtist": "int", "IdProduct": "int" }
Salida	{ "Id": "int",



	<pre>"IdArtist": "int", "IdProduct": "int" }</pre>
Respuestas de éxito	200 – "Added new artist-product successfully!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306 /api/ArtistProduct/Update/{id}
Descripción	Edita la relación entre un producto y su artista. ArtistProduct es la tabla que contiene los productos de los artistas. El parámetro de entrada es el id del producto del artista.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdArtist": "int", "IdProduct": "int" }</pre>
Respuestas de éxito	200 – "Artist-product successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/ArtistProduct/Delete/{id}
Descripción	Elimina la relación entre un artista y su producto, ArtistProduct es la tabla que contiene los productos de los artistas. El parámetro de entrada es el id del producto del artista.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ }</pre>
Respuestas de éxito	200 – "Artist-product successfully deleted!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"



CreditCard

Ruta	<i>https://localhost:44306/api/CreditCard/GetAll</i>
Descripción	<i>Trae todas las tarjetas de crédito de todos los usuarios. CreditCard es la tabla que contiene las tarjetas de crédito de cada usuario.</i>
Parámetros entrada	<pre>{ *No aplica* }</pre>
Salida	<pre>{ "Id": "int", "DateBirth": "DateTime", "CreditCardNumber": "string", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/CreditCard/Get/{id}</i>
Descripción	<i>Trae la información de las tarjetas de crédito de un usuario en específico. CreditCard es la tabla que contiene las tarjetas de crédito de cada usuario. El parámetro de entrada es el id del usuario.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "DateBirth": "DateTime", "CreditCardNumber": "string", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/CreditCard/Create</i>
-------------	--



Descripción	<i>Crea la información de una tarjeta de crédito de un usuario. CreditCard es la tabla que contiene las tarjetas de crédito de cada usuario.</i>
Parámetros entrada	<pre>{ "DateBirth": "DateTime", "CreditCardNumber": "string", "Username": "string" }</pre>
Salida	<pre>{ "Id": "int", "DateBirth": "DateTime", "CreditCardNumber": "string", "Username": "string" }</pre>
Respuestas de éxito	200 – "Added new credit card successfully!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/CreditCard/Update/{id}
Descripción	<i>Edita la información de la tarjeta de crédito del usuario. CreditCard es la tabla que contiene las tarjetas de crédito de cada usuario. El parámetro de entrada es el id del usuario.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "DateBirth": "DateTime", "CreditCardNumber": "string", "Username": "string" }</pre>
Respuestas de éxito	200 – "Credit card successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/CreditCard/Delete/{id}
Descripción	<i>Elimina la información de un usuario específico. CreditCard es la tabla que contiene las tarjetas de crédito de cada usuario. El parámetro de entrada es el id del usuario.</i>
Parámetros entrada	<pre>{</pre>



	<pre>"Id": "int" }</pre>
Salida	<pre>{ }</pre>
Respuestas de éxito	200 – "Credit card successfully deleted!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Format

Ruta	https://localhost:44306/api/Format/GetAll
Descripción	Trae todos los tipos de formatos de los productos. Format es la tabla que contiene la información del género del producto.
Parámetros entrada	<pre>{ *No aplica* }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Format/Get/{id}
Descripción	Trae la información de un formato de un producto en específico. Format es la tabla que contiene la información del género del producto. El parámetro de entrada es el id del formato.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 – "Success"



Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Ruta	https://localhost:44306/api/Format/Create
Descripción	Crea una nueva relación entre el producto y el tipo de formato. Format es la tabla que contiene la información del género del producto. El parámetro de entrada es el id del formato y el id del producto.
Parámetros entrada	<pre>{ "IdProduct": "int", "IdFormat": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 - "Added new format successfully!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Ruta	https://localhost:44306/api/Format/Update/{id}
Descripción	Edita la información del formato de un producto. Format es la tabla que contiene la información del género del producto. El parámetro de entrada es el id del formato.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 - "Format successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"



Ruta	<code>https://localhost:44306/api/Format/Disable/{id}</code>
Descripción	Elimina la información del formato de un producto. Format es la tabla que contiene la información del género del producto. El parámetro de entrada es el id del formato.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 – "Format successfully disabled!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Genre

Ruta	<code>https://localhost:44306/api/Genre/GetAll</code>
Descripción	Trae todos los tipos de géneros. Genre es la tabla que tiene los tipos de géneros del producto.
Parámetros entrada	<pre>{ *No aplica* }</pre>
Salida	<pre>{ "Id": "int", "GenreName": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/Genre/Get/{id}</code>
Descripción	Trae el género específico de los tipos de géneros. Genre es la tabla que tiene los tipos de géneros del producto. El parámetro de entrada es el id del género existente.



Parámetros entrada	{ "Id": "int" }
Salida	{ "Id": "int", "GenreName": "string", "Active": "bool" }
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Genre/Create
Descripción	Crea la relación entre un género y su producto. Genre es la tabla que tiene los tipos de géneros del producto.
Parámetros entrada	{ "Id": "int", "GenreName": "string", "Active": "bool" }
Salida	{ "Id": "int", "GenreName": "string", "Active": "true" }
Respuestas de éxito	200 – "Added new genre successfully!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Genre/Update/{id}
Descripción	Edita el tipo de género. Genre es la tabla que tiene los tipos de géneros del producto. El parámetro de entrada es el id del género existente.
Parámetros entrada	{ "Id": "int" }
Salida	{ "Id": "int", "GenreName": "string", "Active": "bool" }



Respuestas de éxito	200 – "Genre successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/Genre/Disable/{id}</code>
Descripción	Elimina el tipo de género. Genre es la tabla que tiene los tipos de géneros del producto. El parámetro de entrada es el id del género existente.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Active": "false" }</pre>
Respuestas de éxito	200 – "Genre successfully disabled!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Order

Ruta	<code>https://localhost:44306/api/Order/Get</code>
Descripción	Trae todas las órdenes. Order es la tabla que tiene todas las órdenes.
Parámetros entrada	*No aplica*
Salida	<pre>{ "Id": "int", "Total": "decimal", "Status": "string", "IdAdress": "int", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/Order/Get/{id}</code>
-------------	---



Descripción	<i>Trae las órdenes específicas que trae por id. Order es la tabla que tiene todas las órdenes. El parámetro de entrada es el id de la orden hecha.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "Total": "decimal", "Status": "string", "IdAdress": "int", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/Order/Create</i>
Descripción	<i>Crea una nueva entrada con información sobre una Orden. Order es la tabla que tiene todas las órdenes.</i>
Parámetros entrada	<pre>{ "Id": "int", "Total": "decimal", "Status": "string", "IdAdress": "int", "Username": "string" }</pre>
Salida	<pre>{ "Id": "int", "Total": "decimal", "Status": "string", "IdAdress": "int", "Username": "string" }</pre>
Respuestas de éxito	200 – "Added new order successfully!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/Order/Update/{id}</i>
Descripción	<i>Edita la información sobre una Orden. Order es la tabla que tiene todas las órdenes. El parámetro de entrada es el id de la orden hecha.</i>



Parámetros entrada	{ "Id": "int" }
Salida	{ "Id": "int", "Total": "decimal", "Status": "string", "IdAddress": "int", "Username": "string" }
Respuestas de éxito	200 – "Order successfully edited!!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Order/Disable/{id}
Descripción	Elimina la información sobre una Orden. Order es la tabla que tiene todas las órdenes. El parámetro de entrada es el id de la orden hecha.
Parámetros entrada	{ "Id": "int" }
Salida	{ "Status": "delivered" }
Respuestas de éxito	200 – "Order successfully disabled!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

OrderProduct

Ruta	https://localhost:44306/api/OrderProduct/Get
Descripción	Trae toda la información de la relación entre la orden y el producto. OrderProduct es la tabla que tiene todas los productos de una orden.
Parámetros entrada	*No aplica*
Salida	{ "Id": "int", "IdProduct": "int", "IdOrder": "int" }



Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/OrderProduct/Get/{id}</code>
Descripción	<i>Trae la información en específico de una relación entre la orden y el producto. OrderProduct es la tabla que tiene todas los productos de una orden. El parámetro de entrada es el id del producto dentro de la orden.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdOrder": "int" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/OrderProduct/Create</code>
Descripción	<i>Crea una nueva información en específico de una relación entre la orden y el producto. OrderProduct es la tabla que tiene todas los productos de una orden. El parámetro de entrada es el id del producto dentro de la orden y el id de orden.</i>
Parámetros entrada	<pre>{ "IdProduct": "int", "IdOrder": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdOrder": "int" }</pre>
Respuestas de éxito	200 – "Added new orderProduct successfully"



Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"
Ruta	<i>https://localhost:44306/api/OrderProduct/Update/{id}</i>
Descripción	<i>Edita la información en específico de una relación entre la orden y el producto. OrderProduct es la tabla que tiene todas los productos de una orden. El parámetro de entrada es el id del producto dentro de la orden.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdOrder": "int" }</pre>
Respuestas de éxito	200 - "OrderProduct successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Ruta	<i>https://localhost:44306/api/OrderProduct/Delete/{id}</i>
Descripción	<i>Elimina la información en específico de una relación entre la orden y el producto. OrderProduct es la tabla que tiene todas los productos de una orden. El parámetro de entrada es el id de la orden.</i>
Parámetros entrada	<pre>{ "IdOrder": "int" }</pre>
Salida	<pre>{ }</pre>
Respuestas de éxito	200 - "OrderProduct successfully deleted!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Product

Ruta	<i>https://localhost:44306/api/Product/Get</i>
-------------	--



Descripción	<i>Trae toda la información de los productos. Product es la tabla que tiene todos los productos de la tienda.</i>
Parámetros entrada	*No aplica*
Salida	<pre>{ "Id": "int", "Name": "string", "Price": "decimal", "Description": "string", "Active": "bool", "IdGenre": "int" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Product/Get/{id}
Descripción	<i>Trae toda la información de los productos. Product es la tabla que tiene todos los productos de la tienda. El parámetro de entrada es el id del producto.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "Name": "string", "Price": "decimal", "Description": "string", "Active": bool, "IdGenre": int }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Product/GetProductFormats/{id}
Descripción	<i>Trae toda la información de los productos con sus formatos. Product es la tabla que tiene todos los productos de la tienda. El parámetro de entrada es el id del producto.</i>
Parámetros entrada	<pre>{</pre>



	<pre>"Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "Name": "string", "Price": "decimal", "Description": "string", "Active": bool, "IdGenre": int }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Product/Create
Descripción	Crea la información de un nuevo producto. Product es la tabla que tiene todos los productos de la tienda.
Parámetros entrada	<pre>{ "Id": "int", "Name": "string", "Price": "decimal", "Description": "string", "Active": bool, "IdGenre": int }</pre>
Salida	<pre>{ "Id": "int", "Name": "string", "Price": "decimal", "Description": "string", "Active": bool, "IdGenre": int }</pre>
Respuestas de éxito	200 – "Added new product successfully!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/Product/Update/{id}
Descripción	Edita la información de un producto. Product es la tabla que tiene todos los productos de la tienda. El parámetro de entrada es el id del producto.



Parámetros entrada	{ "Id": "int" }
Salida	{ "Id": "int", "Name": "string", "Price": "decimal", "Description": "string", "Active": bool, "IdGenre": int }
Respuestas de éxito	200 – "Product successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<i>https://localhost:44306/api/Product/Disable/{id}</i>
Descripción	<i>Elimina la información de un producto. Product es la tabla que tiene todos los productos de la tienda. El parámetro de entrada es el id del producto.</i>
Parámetros entrada	{ "Id": "int", "Active": "false" }
Salida	{ "Id": "int", "Name": "string", "Price": "decimal", "Description": "string", "Active": "false", "IdGenre": "int" }
Respuestas de éxito	200 – "Product successfully disabled!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

ProductFormat

Ruta	<i>https://localhost:44306/api/ProductFormat/GetAll</i>
Descripción	<i>Trae la información entre la relación de un producto y su formato. ProductFormat es la tabla que tiene la información del formato de cada producto.</i>
Parámetros entrada	*No aplica*



Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/ProductFormat/Get/{id}
Descripción	<i>Trae la información la relación de un producto en específico y su formato. ProductFormat es la tabla que tiene la información del formato de cada producto. El parámetro de entrada es el id del formato del producto.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/ProductFormat/Create
Descripción	<i>Crea una nueva relación de un producto en específico y su formato. ProductFormat es la tabla que tiene la información del formato de cada producto.</i>
Parámetros entrada	<pre>{ "IdProduct": "int", "IdFormat": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 – "Added new orderProduct successfully!"



Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"
----------------------------	--

Ruta	<code>https://localhost:44306/api/ProductFormat/Update/{id}</code>
Descripción	Edita la información la relación de un producto en específico y su formato. ProductFormat es la tabla que tiene la información del formato de cada producto. El parámetro de entrada es el id del formato del producto.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "IdFormat": "int" }</pre>
Respuestas de éxito	200 - "Product-format successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Ruta	<code>https://localhost:44306/api/ProductFormat/Delete/{id}</code>
Descripción	Elimina la información la relación de un producto en específico y su formato. ProductFormat es la tabla que tiene la información del formato de cada producto. El parámetro de entrada es el id del formato del producto.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ }</pre>
Respuestas de éxito	200 - "Product-format successfully deleted!"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Security

Ruta	<code>https://localhost:44306/api/Security/CreateUser</code>
-------------	--



Descripción	<i>Crea la seguridad del usuario.</i>
Parámetros entrada	<pre>{ "name": "string", "userName": "string", "phoneNumber": "string", "email": "string", "password": "string" }</pre>
Salida	<pre>{ "name": "string", "userName": "string", "phoneNumber": "string", "email": "string", "password": "string" }</pre>
Respuestas de éxito	200 – "Usuario creado exitosamente"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message" 500 – "Llena todos los campos" 500 – "Ingresa un email válido" 500 – "Ingresa una contraseña mayor a 4 caracteres" 500 – "Ingresa un número telefónico de 10 caracteres"

Ruta	<i>https://localhost:44306/api/Security/Login</i>
Descripción	<i>Permite al usuario entrar a la aplicación web. El parámetro de entrada es el username y password del usuario.</i>
Parámetros entrada	<pre>{ "userName": "string", "password": "string" }</pre>
Salida	<pre>{ "userName": "string", "password": "string" }</pre>
Respuestas de éxito	200 – "Ok"
Respuestas de Error	400 - "Invalid Action" 500 – "Exception Message" 500 – "Usuario o contraseña inválidos" 500 – "El usuario no existe"

ShoppingCart

Ruta	<i>https://localhost:44306/api/ShoppingCart/Get</i>
-------------	--



Descripción	<i>Trae la información de los elementos del carrito de compras. ShoppingCart es la tabla que tiene la información del carrito de compras.</i>
Parámetros entrada	*No aplica*
Salida	<pre>{ "Id": "int", "IdProduct": "int", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/ShoppingCart/Get/{id}
Descripción	<i>Trae la información de elementos específicos del carrito de compras. ShoppingCart es la tabla que tiene la información del carrito de compras. El parámetro de entrada es el id del carrito de compras.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "Username": "string" }</pre>
Respuestas de éxito	200 – "Success"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/ShoppingCart/Create
Descripción	<i>Crea la información de un nuevo elemento del carrito de compras. ShoppingCart es la tabla que tiene la información del carrito de compras. El parámetro de entrada es el id del carrito de compras, id del producto y el id del username.</i>
Parámetros entrada	<pre>{ "Id": "int", "IdProduct": "int", "Username": "string" }</pre>



Salida	<pre>{ "Id": "int", "IdProduct": "int", "Username": "string" }</pre>
Respuestas de éxito	200 – "Added new shoppingCart successfully!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/ShoppingCart/Update/{id}
Descripción	<i>Edita la información de un elemento del carrito de compras. ShoppingCart es la tabla que tiene la información del carrito de compras. El parámetro de entrada es el id del carrito de compras.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Id": "int", "IdProduct": "int", "Username": "string" }</pre>
Respuestas de éxito	200 – "ShoppingCart successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	https://localhost:44306/api/ShoppingCart/Delete/{id}
Descripción	<i>Elimina la información de un elemento del carrito de compras. ShoppingCart es la tabla que tiene la información del carrito de compras. El parámetro de entrada es el id del carrito de compras.</i>
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ }</pre>
Respuestas de éxito	200 – "ShoppingCart successfully deleted!"



Respuestas de Error

400 - "Invalid Action"
500 - "Exeption Message"

User

Ruta	<code>https://localhost:44306/api/User/Get</code>
Descripción	Trae la información de todos los elementos de los usuarios. User es la tabla que tiene la información de los usuarios.
Parámetros entrada	*No aplica*
Salida	<pre>{ "Username": "string", "Name": "string", "Password": "string", "Email": "string", "PhoneNumber": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 - "Success"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"

Ruta	<code>https://localhost:44306/api/User/Get/{id}</code>
Descripción	Trae la información de un elemento en específico de los usuarios. User es la tabla que tiene la información de los usuarios. El parámetro de entrada es el id del usuario.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Username": "string", "Name": "string", "Password": "string", "Email": "string", "PhoneNumber": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 - "Success"
Respuestas de Error	400 - "Invalid Action" 500 - "Exeption Message"



Ruta	<code>https://localhost:44306/api/User/Update/{id}</code>
Descripción	Edita la información de un elemento en específico de los usuarios. User es la tabla que tiene la información de los usuarios. El parámetro de entrada es el id del usuario.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Username": "string", "Name": "string", "Password": "string", "Email": "string", "PhoneNumber": "string", "Active": "bool" }</pre>
Respuestas de éxito	200 – "User successfully edited!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"

Ruta	<code>https://localhost:44306/api/User/Disable/{id}</code>
Descripción	Elimina la información de un elemento en específico de los usuarios. User es la tabla que tiene la información de los usuarios. El parámetro de entrada es el id del usuario.
Parámetros entrada	<pre>{ "Id": "int" }</pre>
Salida	<pre>{ "Active": "false" }</pre>
Respuestas de éxito	200 – "User successfully disabled!"
Respuestas de Error	400 - "Invalid Action" 500 – "Exeption Message"