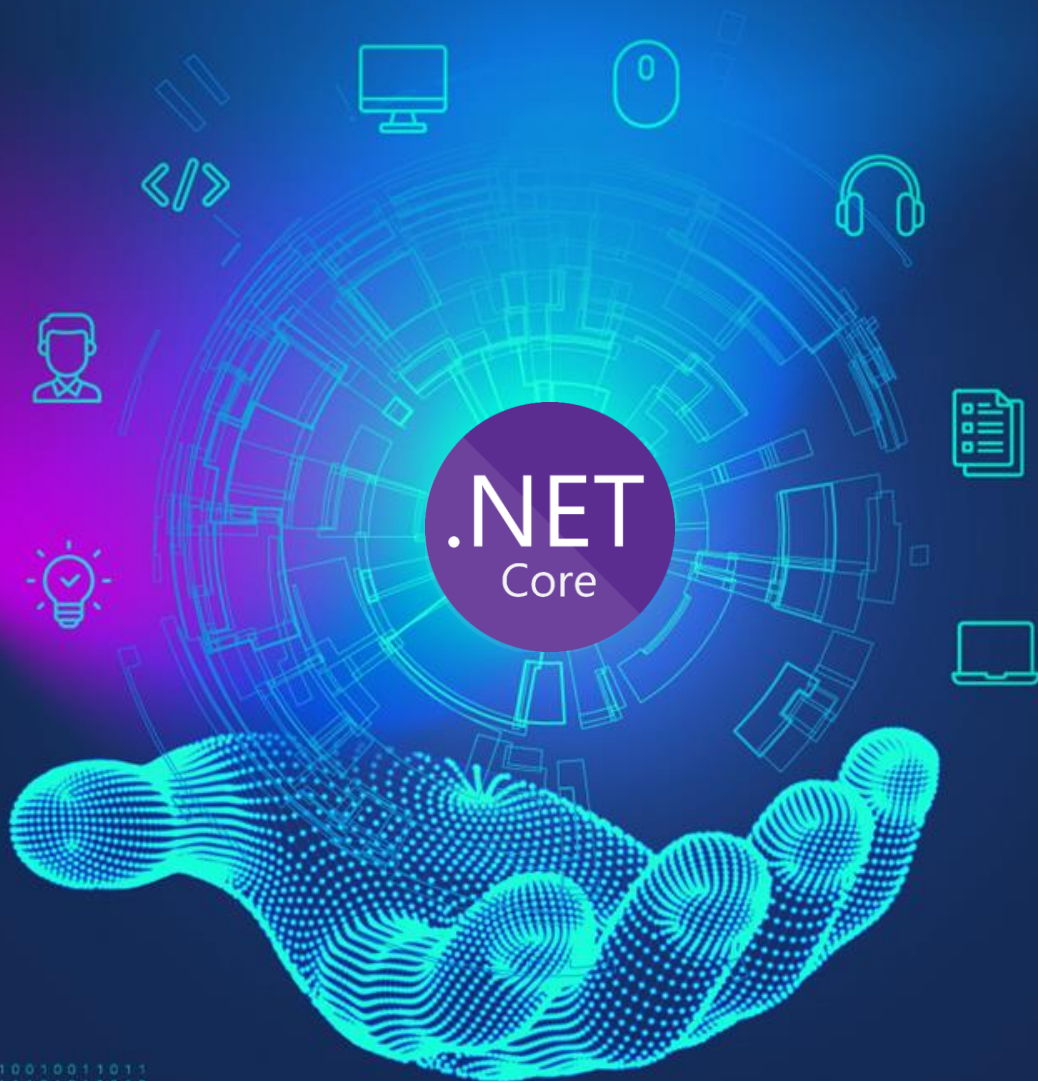




www.singulartech.ai



.Net Apis

.NET es una plataforma de desarrollo de código abierto creada y soportada por Microsoft utilizada para la construcción de diferentes tipos de aplicaciones de software.



Contenido del Curso

1. Introducción

1. Algo de Teoría
2. Algo de Historia
3. SOAP vs REST

2. Escenario

1. Herramientas de Desarrollo
2. Creación del Proyecto

3. Documentación

1. OpenAPI
2. Swagger
3. Documentación

4. Routing

5. Almacenamiento

1. Entity Framework
2. SQLite
3. SQL Server

6. Seguridad

1. Auth0
2. Apis
3. Permisos
4. Roles
5. Usuarios
6. Callbacks
7. Authorization

1. Herramientas de Desarrollo



Visual Studio

Link de Descarga:

<https://visualstudio.microsoft.com/downloads/>



Azure Data Studio

Link de Descarga:

<https://docs.microsoft.com/en-us/sql/azure-data-studio>



SQL Server

Link de Descarga:

<https://www.microsoft.com/es-es/sql-server/sql-server-downloads>



DB Browser for SQLite

Link de Descarga:

<https://sqlitebrowser.org/>

The background is a dark blue gradient with a pattern of small, light blue dots. Overlaid on this are several glowing, magenta-pink lines that form a complex, swirling pattern. A bright yellow and orange comet-like streak is visible in the lower right quadrant, moving towards the center. The text "1. INTRODUCCIÓN" is centered in a bold, white, sans-serif font.

1. INTRODUCCIÓN

.Net

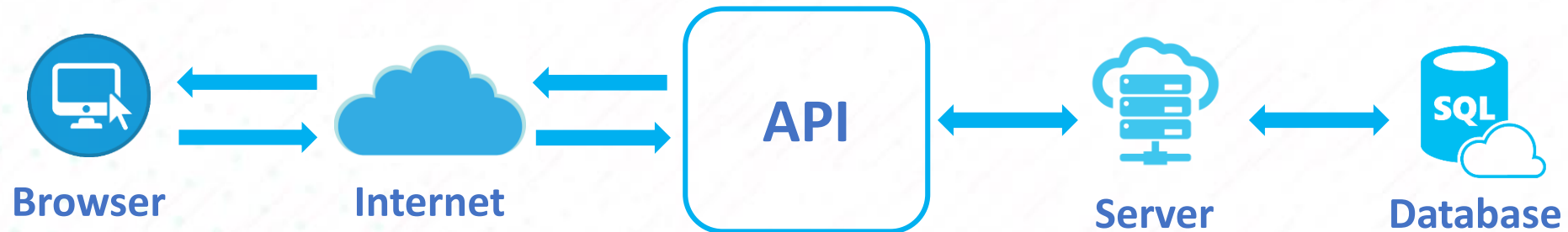
.NET Core es una plataforma de desarrollo de código abierto permite desarrollar aplicaciones Web, Desktop, Mobile y también Apis. En pocos pasos y con código simple se puede desarrollar un api completamente funcional que puede cubrir aspectos como Documentación, Routing, Lectura/Escritura de Datos y Autenticación que son los aspectos básicos que debe tener una aplicación back-end.

.Net Core es el corazón del ecosistema de desarrollo soportado por Microsoft

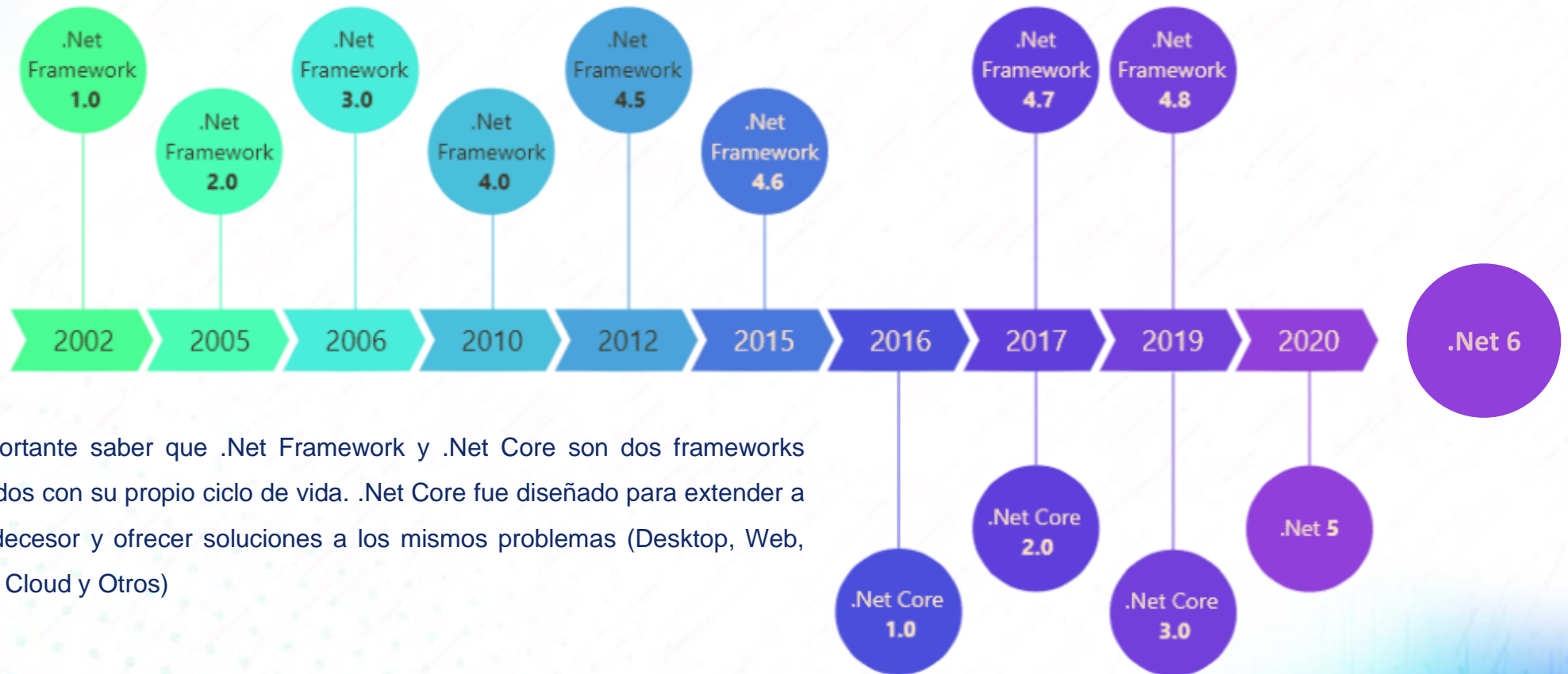


1. Algo de Teoría

Un API (Application Programming Interface) es un set de herramientas, definiciones y protocolos para integrar aplicaciones de software y servicios, estas definiciones nos permiten comunicar nuestros productos de software y servicios de software con otros productos y servicios sin la necesidad de modificar constantemente nuestra infraestructura.



2. Algo de Historia



Es importante saber que .Net Framework y .Net Core son dos frameworks separados con su propio ciclo de vida. .Net Core fue diseñado para extender a su predecesor y ofrecer soluciones a los mismos problemas (Desktop, Web, Mobile, Cloud y Otros)

3. SOAP vs REST

Simple Object Access Protocol (SOAP) y Representational State Transfer (REST) son dos tecnologías que han ayudado a que las API sean más simples en su diseño y más útiles en su implementación.

SOAP	REST
Xml-based message protocol	Architectural style protocol
Uses WSDL for communication between consumer and provider	Uses XML or JSON to send and receive data
Invokes services by calling RPC (Remote Procedure Calls) method	Simply calls services via URL path
Does not return human readable result	Result is readable which is just plain XML or JSON
Transfer is over HTTP, also uses other protocols such as SMTP, FTP, etc.	Transfer is over HTTP only
Javascript can call SOAP, but is difficult to implement	Easy to call from Javascript
Performance is not great compared to REST	Performance is much better compared to SOAP, less CPU intensive, lighter code, etc.

The background is a dark blue gradient. It features several glowing, overlapping pink or magenta elliptical orbits that resemble planetary paths. Scattered throughout the lower half of the image are numerous small, bright blue dots, giving the impression of a star field or a digital data space. A few larger, brighter blue and yellowish-white points are also visible, possibly representing specific celestial bodies or data points of interest.

2. ESCENARIO

Escenario

Eres un desarrollador web y estás a cargo del mantenimiento de Apis de tu compañía

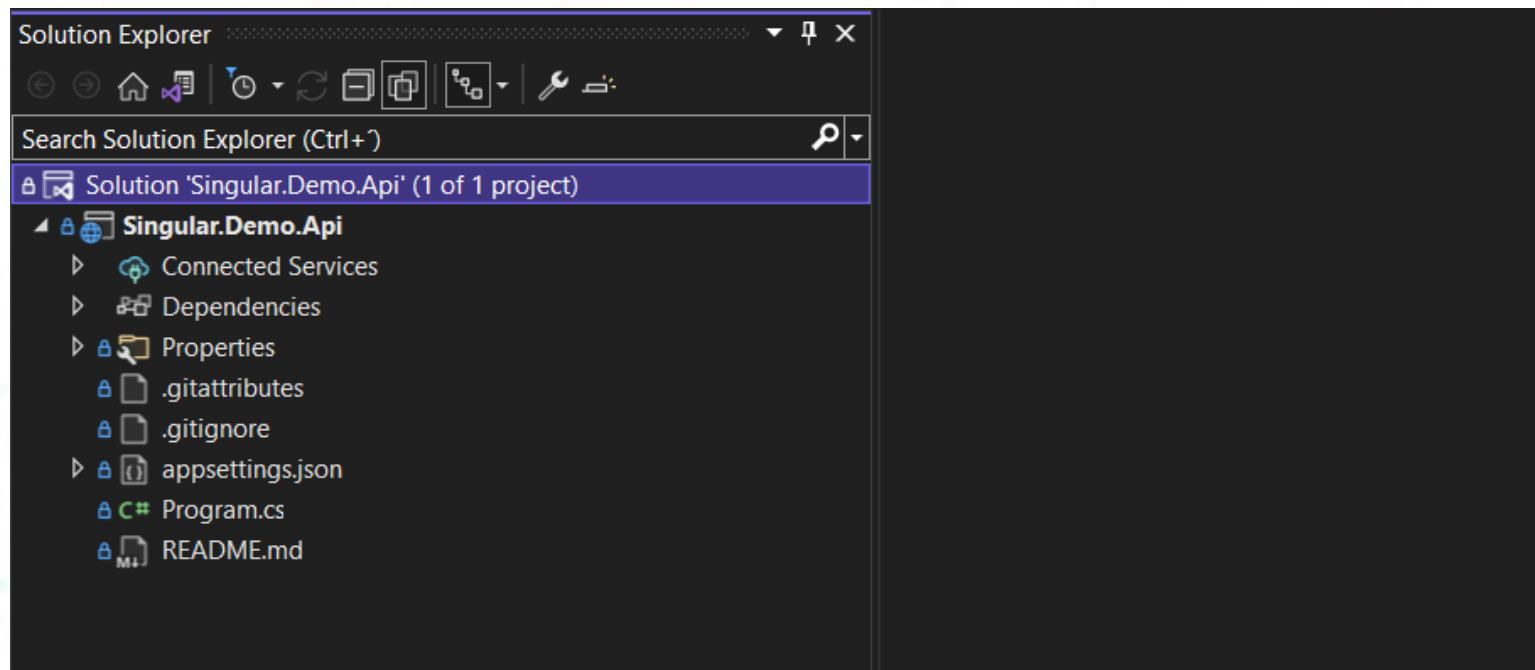
Para este ejercicio, necesitas crear un api que exponga y recolecte números de teléfonos para una aplicación cliente ya sea web, mobile o desktop. Debes asumir que el api que construyas sea simple y extensible a futuro.

Objetivo

Construir un api con las características mínimas de Almacenamiento, Seguridad, Versionamiento y Documentación usando .Net Core.

2. Creación del Proyecto

```
dotnet new web -o Singular.Demo.Api -f net6.0
```



3. DOCUMENTACIÓN

The background is a dark blue gradient with a pattern of small, light blue dots. Overlaid on this are several glowing, abstract lines. A prominent pink line forms a large, sweeping loop that encircles the text. Other thinner, glowing lines in shades of pink and blue are scattered around, some ending in bright, out-of-focus light spots.

1. OpenAPI

La iniciativa OpenAPI (OAI) fue creada por un consorcio de expertos de la industria con visión de futuro que reconocen el inmenso valor de la estandarización de cómo las apis son descritas y documentadas. Como una estructura de gobernanza bajo la Fundación Linux (Linux Foundation), la OAI está centrada en crear, evolucionar y promover un formato de descripción neutral entre proveedores. La especificación OpenAPI fue originalmente basada en la especificación Swagger, donada por la compañía SmartBear Software.

OpenAPI Specification v3.1.0

Version 3.1.0

Published 15 February 2021

2. Swagger

La librería por excelencia para documentar apis en .Net Core es Swagger.

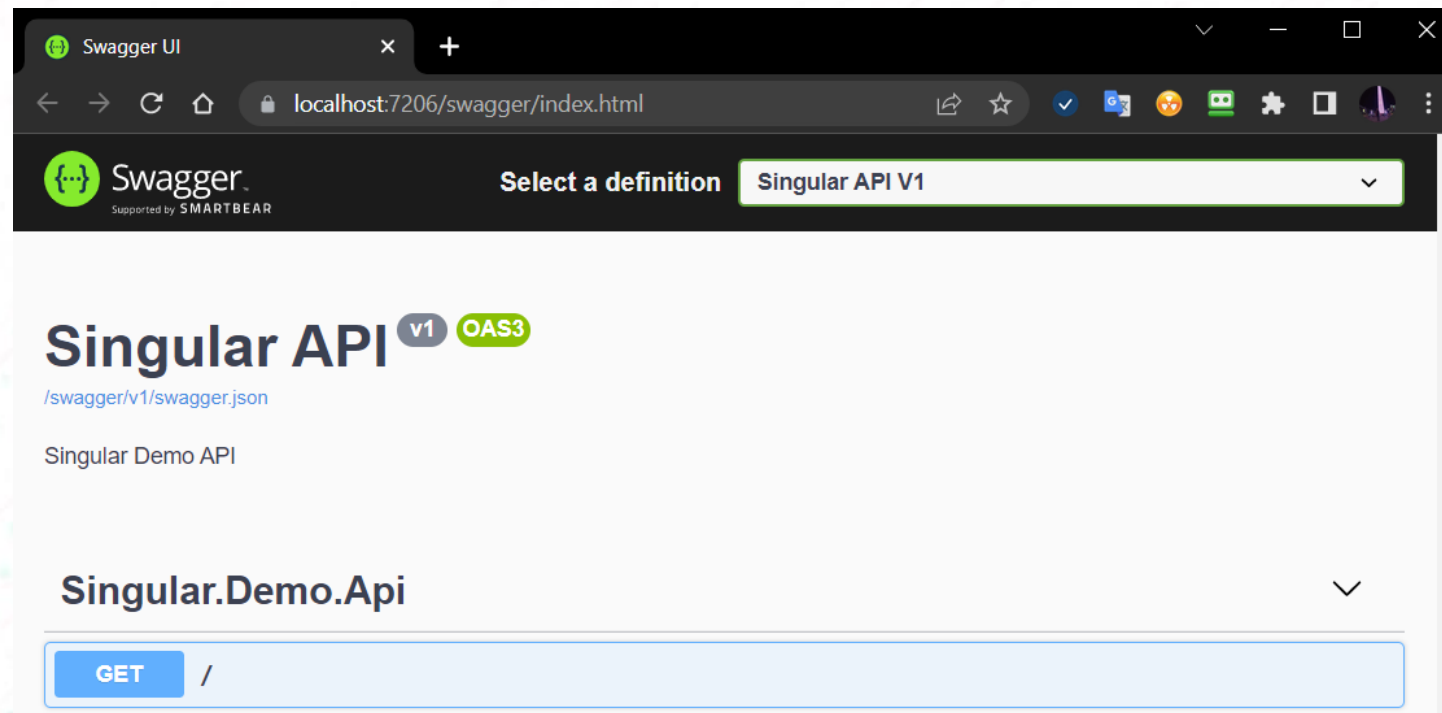
La “Especificación Swagger” fue creada por la compañía SmartBear Software con el fin de documentar apis de un a forma agnóstica durante el diseño de apis, esta especificación creació lo suficiente en el mercado al punto de convertirse en un estándar para la documentación y fue adoptada por el consorcio que ahora lo mantiene.



3. Documentación

Para agregar la librería de Swagger al proyecto ejecutar el siguiente comando

```
dotnet add package SwashBuckle.AspNetCore --version 6.1.4
```



The background is a dark blue gradient. It features several glowing, overlapping pinkish-purple elliptical orbits. In the lower-left corner, there is a dense field of small, light blue dots. Along the bottom of the orbits, there are several small, bright blue and yellow-white points of light, some with soft halos, suggesting stars or data points.

4. ROUTING

Routing

Para realizar las llamadas API RESTful, debes hacer uso de verbos HTTP

HTTP verb	Description
POST	Sends data that creates a resource
PUT	Sends data that updates a resource
DELETE	Removes a resource
GET	Returns data

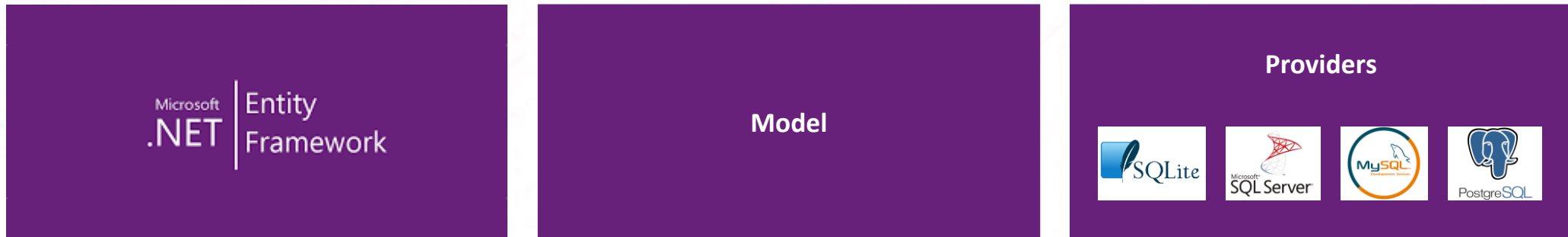
```
app.MapPost("/phones", (Phone phone) => PhonesDb.Create(phone));  
app.MapPut("/phones", (Phone phone) => PhonesDb.Update(phone));  
app.MapDelete("/phones/{id}", (int id) => PhonesDb.Delete(id));  
app.MapGet("/phones/{id}", (int id) => PhonesDb.Get(id));  
app.MapGet("/phones", () => PhonesDb.List());
```

The background is a dark blue gradient with a pattern of small, light blue dots. Overlaid on this are several glowing, abstract lines. A prominent, thick, pinkish-purple line forms a large, irregular loop that encircles the text. Other thinner, glowing lines in shades of pink and blue are scattered around, some ending in bright, out-of-focus light spots. The overall aesthetic is futuristic and digital.

5. ALMACENAMIENTO

1. Entity Framework

La herramienta más usada para el acceso a datos y almacenamiento en .NET Core es Entity Framework, este herramienta permite abstraer la forma en la que se acceden a diferentes tipos de bases de datos entre ellas SQL Server, MySQL, Postgres, etc. mediante “Providers”.



```
dotnet tool install --global dotnet-ef
```


2. SQLite

```
dotnet add package Microsoft.EntityFrameworkCore.Sqlite --version 6.0
```

```
dotnet add package Microsoft.EntityFrameworkCore.Design --version 6.0
```

```
dotnet ef migrations add SQLite
```

```
dotnet ef database update
```

Table: Phones				
	Id	Brand	Model	Number
	Filter	Filter	Filter	Filter
1	1	Samsung	S9	+51968133858
2	2	Apple	iPhone 11	+51994665973
3	3	Xiaomi	Mi 11	+51992778439

2. SQL Server

```
dotnet add package Microsoft.EntityFrameworkCore.SqlServer --version 6.0
```

```
dotnet add package Microsoft.EntityFrameworkCore.Design --version 6.0
```

```
dotnet ef migrations add SQLServer
```

```
dotnet ef database update
```

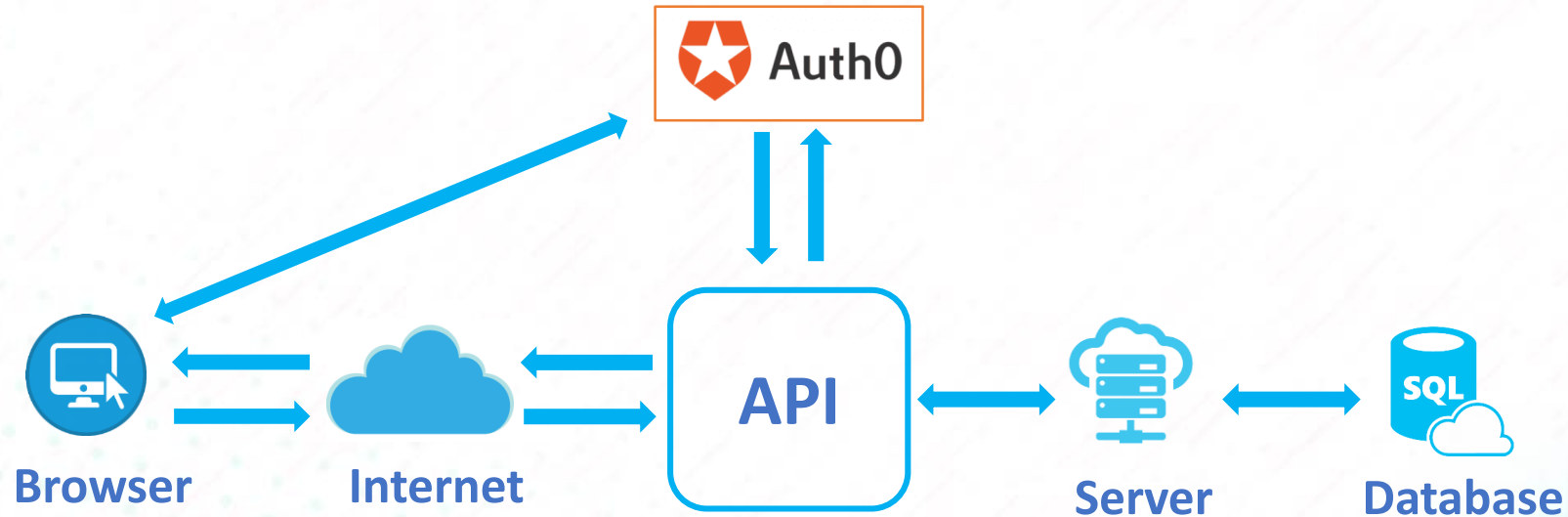
Table: Phones				
	Id	Brand	Model	Number
	Filter	Filter	Filter	Filter
1	1	Samsung	S9	+51968133858
2	2	Apple	iPhone 11	+51994665973
3	3	Xiaomi	Mi 11	+51992778439

The background is a dark blue gradient. It features several glowing, overlapping loops in a vibrant pink/magenta color. Scattered throughout are small, bright blue and white dots, some of which appear to be part of the glowing loops. The overall aesthetic is futuristic and digital.

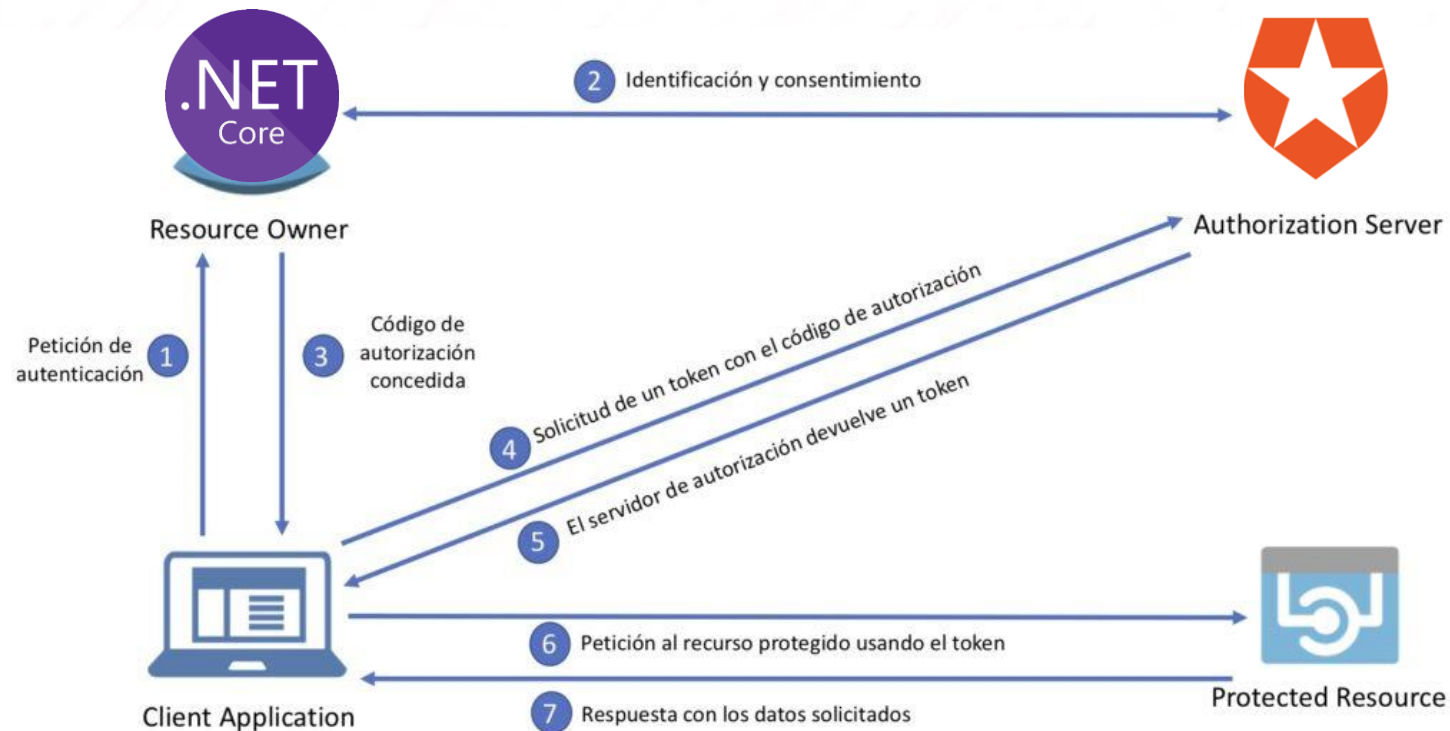
5. SEGURIDAD

1. Auth0

Para asegurar nuestras Apis vamos a usar un servicio de autorización y autenticación de terceros llamado Auth0. Este es un servicio que tiene implementados los estándares de seguridad OAuth2.0 y OpenIDConnect (OIDC) que nos ayudará a proteger nuestras Apis



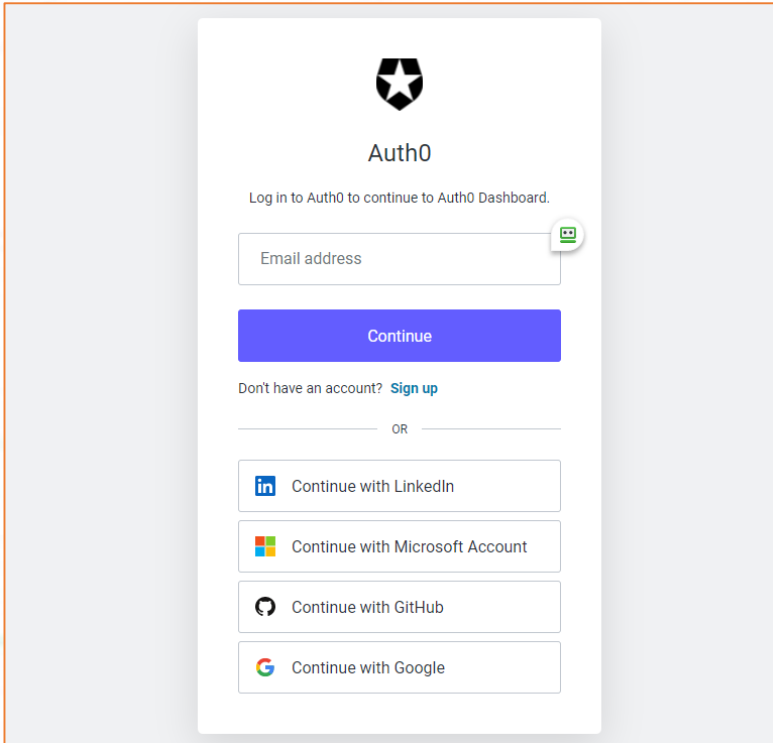
1. OAuth2.0 + OIDC(onnnect)



2. Apis

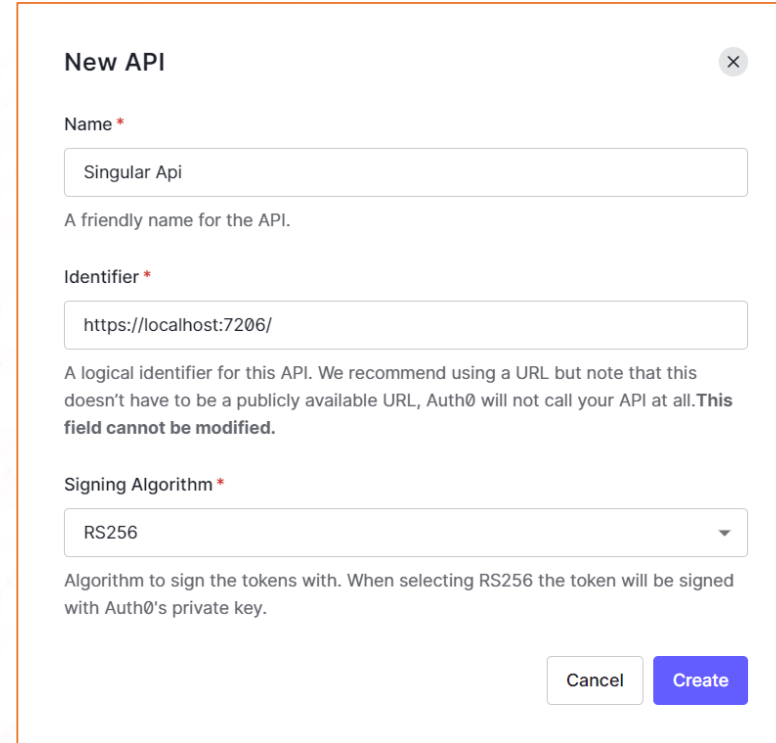
Ingresar  <https://manage.auth0.com/>

Crear Cuenta



The image shows the Auth0 login and signup interface. At the top is the Auth0 logo (a shield with a star) and the text 'Auth0'. Below this is the instruction 'Log in to Auth0 to continue to Auth0 Dashboard.' There is a text input field for 'Email address' with a green eye icon for toggling visibility. Below the input is a blue 'Continue' button. Underneath the button is a link: 'Don't have an account? [Sign up](#)'. A horizontal separator with the word 'OR' in the center follows. Below the separator are five social login buttons: 'Continue with LinkedIn' (with LinkedIn icon), 'Continue with Microsoft Account' (with Microsoft icon), 'Continue with GitHub' (with GitHub icon), and 'Continue with Google' (with Google icon).

Crear Api



The image shows the 'New API' form in the Auth0 dashboard. The form has a title 'New API' with a close button (X) in the top right corner. It contains three main sections: 1. 'Name *' with a text input field containing 'Singular Api' and a description 'A friendly name for the API.' 2. 'Identifier *' with a text input field containing 'https://localhost:7206/' and a detailed description: 'A logical identifier for this API. We recommend using a URL but note that this doesn't have to be a publicly available URL, Auth0 will not call your API at all. This field cannot be modified.' 3. 'Signing Algorithm *' with a dropdown menu currently set to 'RS256' and a description: 'Algorithm to sign the tokens with. When selecting RS256 the token will be signed with Auth0's private key.' At the bottom right of the form are two buttons: 'Cancel' and 'Create'.

3. Permisos

Ingresar  <https://manage.auth0.com/>

Crear Permisos

Add a Permission (Scope)

Define the permissions (scopes) that this API uses.

Permission (Scope) *

read:appointments

Description *

Read your appointments

+ Add

List of Permissions (Scopes)

These are all the permissions (scopes) that this API uses.

Permission	Description	
read:phone	Obtener un número telefónico	
create:phone	Crear un número telefónico	
update:phone	Actualizar un número telefónico	
delete:phone	Eliminar un número telefónico	
read:phones	Listar números telefónicos	

4. Roles

Ingresar  <https://manage.auth0.com/>

Crear Rol y Agregar Permisos

New Role

Name *

Singular Admin

Description *

Administrador

Cancel

Create

Add Permissions

Select permissions from existing APIs

Singular App

Permissions Select: All None

☒ read:phone

☒ create:phone

☒ update:phone

☒ delete:phone

☒ read:phones

Add Permissions

5. Usuarios

Ingresar  <https://manage.auth0.com/>

Crear Usuario

Create user

✕

Email *

Password *

Repeat Password *

Connection *

Username-Password-Authentication

▼

Cancel

Create

Add Roles

✕

Select roles to assign to this user. You may assign up to 50 roles per user.

Singular Admin ✕

Select Roles

✕ ▼

Assign

6. Callbacks

Ingresar  <https://manage.auth0.com/>

Configurar App

Allowed Callback URLs

```
https://localhost:7206/swagger/oauth2-redirect.html
```

After the user authenticates we will only call back to any of these URLs. You can specify multiple valid URLs by comma-separating them (typically to handle different environments like QA or testing). Make sure to specify the protocol (`https://`) otherwise the callback may fail in some cases. With the exception of custom URI schemes for native clients, all callbacks should use protocol `https://` . You can use [Organization URL](#) parameters in these URLs.

Allowed Web Origins

```
https://localhost:7206
```

Comma-separated list of allowed origins for use with [Cross-Origin Authentication](#), [Device Flow](#), and [web message response mode](#), in the form of `<scheme> "://" <host> [":" <port>]` , such as `https://login.mydomain.com` or `http://localhost:3000` . You can use wildcards at the subdomain level (e.g.: `https://*.contoso.com`). Query strings and hash information are not taken into account when validating these URLs.

7. Authorization

```
dotnet add package Microsoft.AspNetCore.Authentication.JwtBearer --version 6.0.3
```

```
var clientId = "*****";
var clientSecret = "*****";
var authority = "https://dev-0000-0000000000.auth0.com/";
var audience = "https://localhost:7206";

var scopes = new Dictionary<string, string> {
    { "create:phone", "Crear número telefónico" },
    { "update:phone", "Actualizar número telefónico" },
    { "delete:phone", "Eliminar número telefónico" },
    { "read:phone", "Obtener número telefónico" },
    { "read:phones", "Listar números telefónicos" }
};
```

A decorative graphic consisting of two overlapping, glowing purple elliptical orbits. A bright blue and white light source is positioned at the bottom right of the orbits, with a trail of light extending upwards and to the left, passing through the center of the orbits.

GRACIAS

LIMA - PERÚ

Calle Monte Rosa 270 Of. 502, Santiago de Surco

contacto@singulartech.ai / +51 992778439 / +51 999015208

www.singulartech.ai