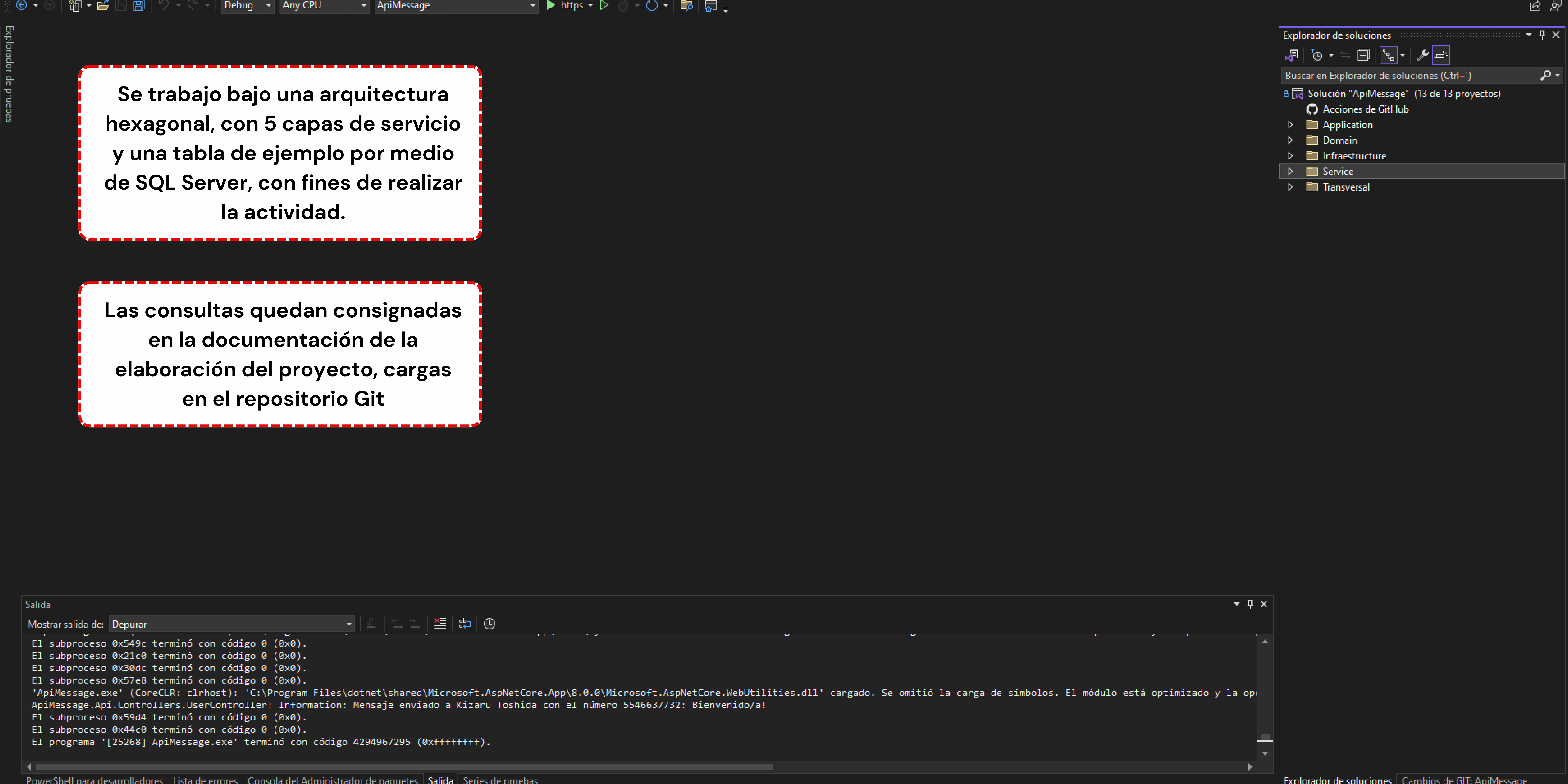


ULTIMA ACTUALIZACION: 14 DE DICIEMBRE DE 2024

INSTRUCTIVO API MESSAGE

VERSION 3.0

DESARROLLADOR:
JHONATTAN HALCON CASALLAS FELIPE



Se trabajo bajo una arquitectura hexagonal, con 5 capas de servicio y una tabla de ejemplo por medio de SQL Server, con fines de realizar la actividad.

Las consultas quedan consignadas en la documentación de la elaboración del proyecto, cargas en el repositorio Git

Salida

Mostrar salida de: Depurar

El subproceso 0x549c terminó con código 0 (0x0).
El subproceso 0x21c0 terminó con código 0 (0x0).
El subproceso 0x30dc terminó con código 0 (0x0).
El subproceso 0x57e8 terminó con código 0 (0x0).
'ApiMessage.exe' (CoreCLR: clrhost): 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\8.0.0\Microsoft.AspNetCore.WebUtilities.dll' cargado. Se omitió la carga de símbolos. El módulo está optimizado y la op...
ApiMessage.Api.Controllers.UserController: Information: Mensaje enviado a Kizaru Toshida con el número 5546637732: Bienvenido/a!
El subproceso 0x59d4 terminó con código 0 (0x0).
El subproceso 0x44c0 terminó con código 0 (0x0).
El programa '[25268] ApiMessage.exe' terminó con código 4294967295 (0xffffffff).

Explorador de soluciones

Buscar en Explorador de soluciones (Ctrl+)

Solución "ApiMessage" (13 de 13 proyectos)

- Acciones de GitHub
- Application
- Domain
- Infraestructura
- Service
- Transversal

Explorador de soluciones Cambios de GIT: ApiMessage

Se trabaja con multiples bibliotecas, dentro de las cuales resalto Dapper, extensiones, AutoMapper, entre otras.

Salida

Mostrar salida de: Depurar

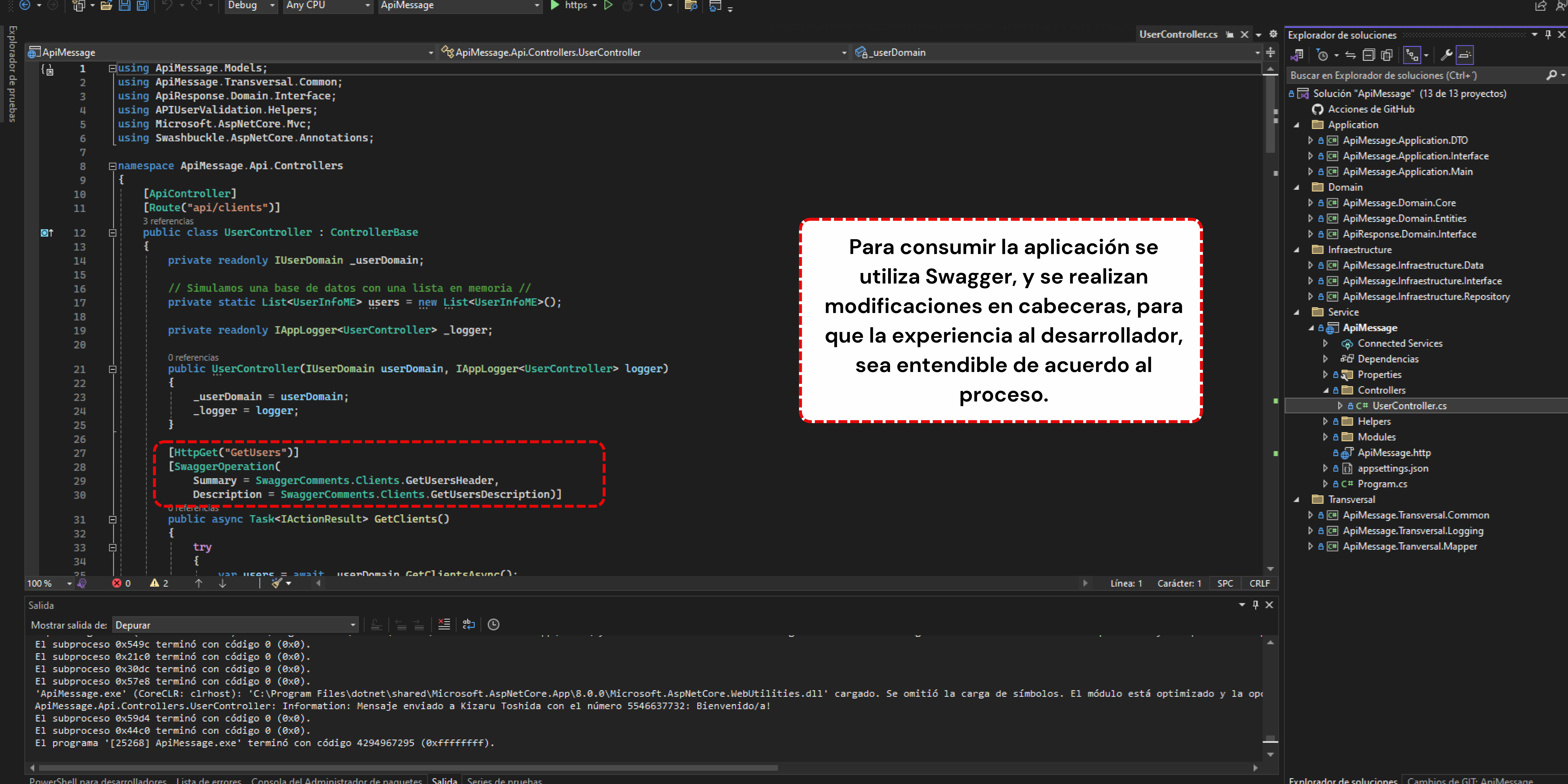
```
El subproceso 0x549c terminó con código 0 (0x0).
El subproceso 0x21c0 terminó con código 0 (0x0).
El subproceso 0x30dc terminó con código 0 (0x0).
El subproceso 0x57e8 terminó con código 0 (0x0).
'ApiMessage.exe' (CoreCLR: clrhost): 'C:\Program Files\dotnet\shared\Microsoft.AspNetCore.App\8.0.0\Microsoft.AspNetCore.WebUtilities.dll' cargado. Se omitió la carga de símbolos. El módulo está optimizado y la op
ApiMessage.Api.Controllers.UserController: Information: Mensaje enviado a Kizaru Toshida con el número 5546637732: Bienvenido/a!
El subproceso 0x59d4 terminó con código 0 (0x0).
El subproceso 0x44c0 terminó con código 0 (0x0).
El programa '[25268] ApiMessage.exe' terminó con código 4294967295 (0xffffffff).
```

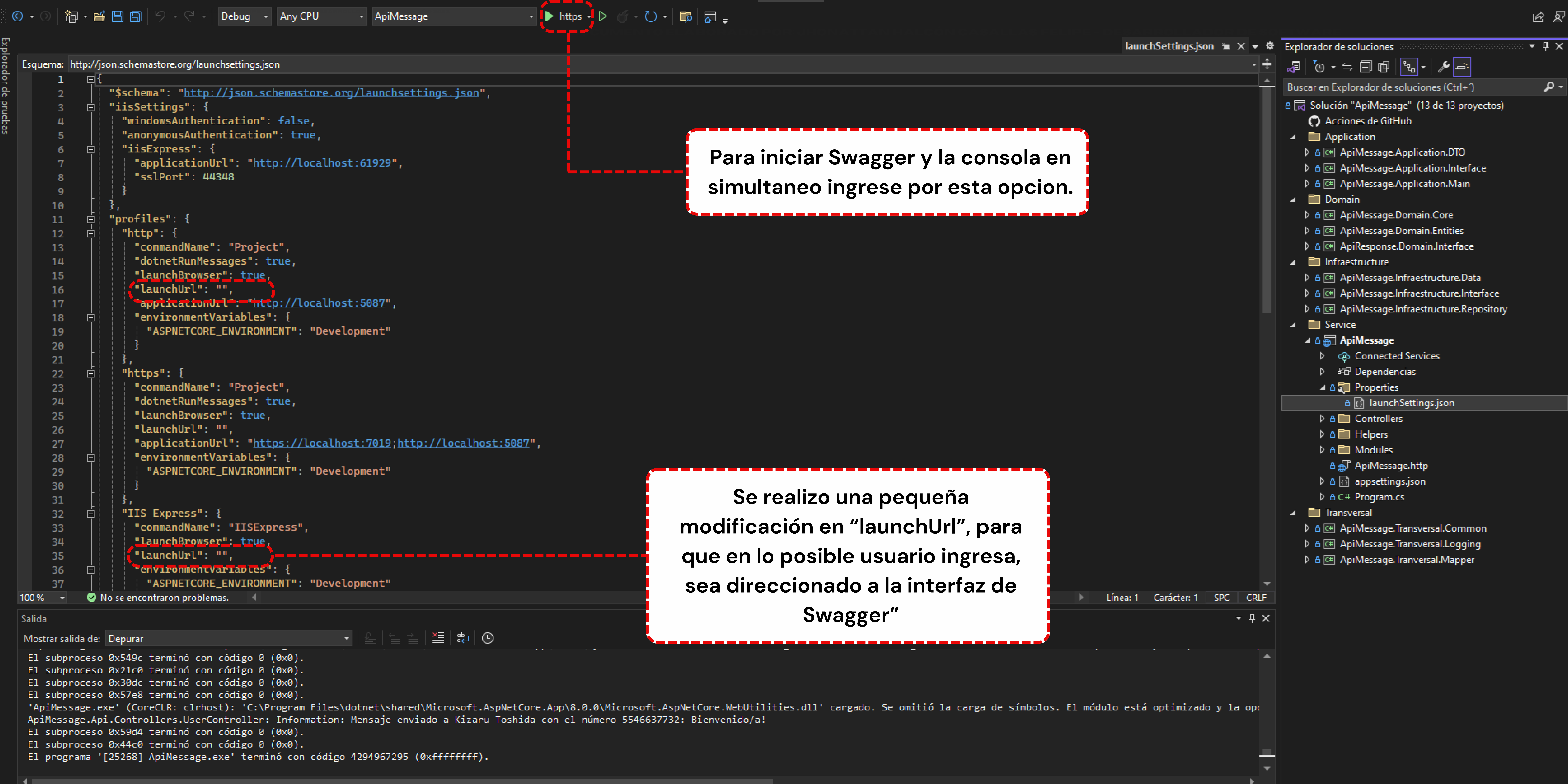
Explorador de soluciones

Buscar en Explorador de soluciones (Ctrl+)

- Solución "ApiMessage" (13 de 13 proyectos)
 - Acciones de GitHub
 - Application
 - ApiMessage.Application.DTO
 - ApiMessage.Application.Interface
 - ApiMessage.Application.Main
 - Domain
 - ApiMessage.Domain.Core
 - ApiMessage.Domain.Entities
 - ApiResponse.Domain.Interface
 - Infraestructure
 - ApiMessage.Infraestructure.Data
 - ApiMessage.Infraestructure.Interface
 - ApiMessage.Infraestructure.Repository
 - Service
 - ApiMessage**
 - Connected Services
 - Dependencias
 - Properties
 - Controllers
 - Helpers
 - Modules
 - ApiMessage.http
 - appsettings.json
 - Program.cs
 - Transversal
 - ApiMessage.Transversal.Common
 - ApiMessage.Transversal.Logging
 - ApiMessage.Tranversal.Mapper

Explorador de soluciones Cambios de GIT: ApiMessage





Para iniciar Swagger y la consola en simultaneo ingrese por esta opcion.

Se realizo una pequeña modificación en "launchUrl", para que en lo posible usuario ingresa, sea direccionado a la interfaz de Swagger"

localhost:7019/index.html

Swagger
Supported by SMARTBEAR

Select a definition
Api-Message v1

Api-Message

v1.0 OAS 3.0

/swagger/v1/swagger.json

User

GET

/api/clients/GetUsers

Obtiene una lista de todos los usuarios de la base de datos. [SQL Vista]

GET

/api/clients/{id}

Obtiene un usuario por su ID. [SQL Function]

POST

/api/clients/CreateClient

Crea un nuevo cliente [SQL Stored Procedure]

PUT

/api/clients/UpdateClient

Actualiza un cliente existente [SQL Stored Procedure]

DELETE

/api/clients/DeleteClient

Elimina un cliente [SQL Stored Procedure]

POST

/api/clients/CreateMessage

Recibe los datos del usuario

Schemas

ProblemDetails

UserInfoME

De acuerdo a la solicitud, se agrega el requerimiento dado por correo electronico, el cual corresponde a este controller.

Parameters

Try it out

No parameters

Request body

application/json

Example Value | Schema

```
{
  "userId": 0,
  "fullName": "string",
  "phoneNumber": "string",
  "email": "string"
}
```

El cual al presionar "Try it out", va a a habilitar el textbox, para diligenciar el JSON.

Responses

| Code | Description | Links |
|------|-------------|----------|
| 200 | OK | No links |

Media type

text/plain

Controls Accept header.

Example Value | Schema

```
{
  "userId": 0,
  "fullName": "string",
  "phoneNumber": "string",
  "email": "string"
}
```

DELETE /api/clients/DeleteClient Elimina un cliente [SQL Stored Procedure] ▾

POST /api/clients/CreateMessage Recibe los datos del usuario ^

Este endpoint recibe una solicitud HTTP POST con el nombre y número de teléfono del usuario, los almacena y simula el envío de un mensaje de confirmación. Dichos registros quedan como mensaje a traves de los logs del sistema, Si alguno de los campos está vacío o nulo, se retornará un error.

Parameters Cancel Reset

No parameters

Request body application/json ▾

```
{
  "userId": 0,
  "fullName": "Jhonattan Halcon Casallas Felipe",
  "phoneNumber": "3144132541",
  "email": "jhonattanhalcon@gmail.com"
}
```

Execute

Responses

| Code | Description | Links |
|------------|-------------|----------|
| 200 | OK | No links |
| Media type | | |

Se deben diligenciar los datos en relacion al usuario, el cual en modo de simulación va a recibir la información”

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7019/api/clients/CreateMessage' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "userId": 0,
    "fullName": "Jhonattan Halcon Casallas Felipe",
    "phoneNumber": "3144132541",
    "email": "jhonattanhalcon@gmail.com"
  }'
```

Request URL

```
https://localhost:7019/api/clients/CreateMessage
```

Server response

| Code | Details |
|------|--|
| 200 | <div>Response body</div> <div><pre>{ "message": "Usuario creado con éxito", "user": { "userId": 0, "fullName": "Jhonattan Halcon Casallas Felipe", "phoneNumber": "3144132541", "email": "jhonattanhalcon@gmail.com" } }</pre></div> <div>Response headers</div> <div><pre>content-type: application/json; charset=utf-8 date: Sat, 14 Dec 2024 10:35:31 GMT server: Kestrel</pre></div> |

| Responses | |
|--|-------------|
| Code | Description |
| 200 | OK |
| <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> | |

Nos genera la respuesta deseada

```
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: https://localhost:7019
info: Microsoft.Hosting.Lifetime[14]
      Now listening on: http://localhost:5087
info: Microsoft.Hosting.Lifetime[0]
      Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
      Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
      Content root path: C:\Users\halco\Desktop\ApiMessage\ApiMessage\ApiMessage
info: ApiMessage.Api.Controllers.UserController[0]
      Mensaje enviado a Jhonattan Halcon Casallas Felipe con el número 3144132541: Bienvenido/a!
```

A su vez, que en consola, nos muestra el nombre completo de la persona que ingreso su información en el controller CreateMessage.

localhost:7019/index.html

Swagger
Supported by SMARTBEAR

Select a definition
Api-Message v1

Api-Message

v1.0OAS 3.0

/swagger/v1/swagger.json

Vista organizada

User

GET

/api/clients/GetUsers

Obtiene una lista de todos los usuarios de la base de datos. [SQL Vista]

Este endpoint permite consultar todos los usuarios registrados en el sistema.La respuesta incluirá el ID del usuario, nombre completo, número de teléfono y correo electrónico.

Si no hay usuarios registrados, se retornará una lista vacía.

Parameters

No parameters

Try it out

Responses

Curl

curl -X 'GET' \n'https://localhost:7019/api/clients/GetUsers' \n-H 'accept: */*'

Request URL

https://localhost:7019/api/clients/GetUsers

Server response

Code

Details

200

Response body

Para complementar la presentación del proyecto, se adiciona el proceso CRUD, para registro de una persona, el cual se encuentra totalmente funcional, y esta vinculado a SQL Server, mediante distintas tecnologías nativas de este manejador de base de datos.

Swagger UI

localhost:7019/index.html

curl

```
curl -X 'GET' \
  'https://localhost:7019/api/clients/GetUsers' \
  -H 'accept: */*'

```

Request URL

https://localhost:7019/api/clients/GetUsers

Server response

Code

Details

200

Response body

```
{
  "userId": 6,
  "fullName": "David Green",
  "phoneNumber": "555-4567",
  "email": "david.green@example.com"
},
{
  "userId": 9,
  "fullName": "Grace Pink",
  "phoneNumber": "555-8901",
  "email": "grace.pink@example.com"
},
{
  "userId": 10,
  "fullName": "Hank Yellow",
  "phoneNumber": "555-9012",
  "email": "hank.yellow@example.com"
},
{
  "userId": 11,
  "fullName": "Ivy Red",
  "phoneNumber": "555-0123",
  "email": "ivy.red@example.com"
},
{
  "userId": 12,
  "fullName": "Jack Purple",
  "phoneNumber": "555-1234",
  "email": "jack.purple@example.com"
}

```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 14 Dec 2024 10:39:36 GMT
server: Kestrel

```


Api-Message

v1.0OAS 3.0

/swagger/v1/swagger.json

Función SQL

User

GET

/api/clients/GetUsers

Obtiene una lista de todos los usuarios de la base de datos. [SQL Vista]

GET

/api/clients/{id}

Obtiene un usuario por su ID. [SQL Function]

Este endpoint permite consultar los detalles de un usuario específico utilizando su ID.

Parameters

Cancel

| Name | Description |
|---------------------------------------|-------------|
| userId integer(\$int32) (query) | userId |
| id * required string (path) | id |

Execute

Responses

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

| Name | Description |
|------------------|-------------|
| userId | |
| integer(\$int32) | 12 |
| (query) | |
| id * required | |
| string | 12 |
| (path) | |

Execute

Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7019/api/clients/12?userId=12' \
  -H 'accept: text/plain'
```

Request URL

```
https://localhost:7019/api/clients/12?userId=12
```

Server response

| Code | Details |
|------|--|
| 200 | <div><div>Response body</div><div><pre>{ "userId": 12, "fullName": "Jack Purple", "phoneNumber": "555-1234", "email": "jack.purple@example.com" }</pre></div><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-type: application/json; charset=utf-8 date: Sat, 14 Dec 2024 10:40:49 GMT server: Kestrel</pre></div></div> |

Responses

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

Procedimientos almacenados

User

GET

/api/clients/GetUsers

Obtiene una lista de todos los usuarios de la base de datos. [SQL Vista]

GET

/api/clients/{id}

Obtiene un usuario por su ID. [SQL Function]

POST

/api/clients/CreateClient

Crea un nuevo cliente [SQL Stored Procedure]

Este endpoint permite crear un nuevo cliente con los datos proporcionados, como nombre, teléfono y correo electrónico. Si alguno de los campos está vacío o nulo, se retornará un error.

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{  "userId": 0,  "fullName": "Miguel Angel Duarte Rivas",  "phoneNumber": "39887477473",  "email": "rivasmigue@gmail.com"}
```

Execute

Clear

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7019/api/clients/CreateClient' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "userId": 50,
    "fullName": "Miguel Angel Duarte Rivas",
    "phoneNumber": "39887477473",
    "email": "rivasmiguel@gmail.com"
  }'
```

Request URL

https://localhost:7019/api/clients/CreateClient

Server response

| Code | Details |
|------|---------|
|------|---------|

201
Undocumented

Response body

```
{
  "userId": 50,
  "fullName": "Miguel Angel Duarte Rivas",
  "phoneNumber": "39887477473",
  "email": "rivasmiguel@gmail.com"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 14 Dec 2024 10:43:06 GMT
location: https://localhost:7019/api/clients/50
server: Kestrel
```

Responses

| Code | Description | Links |
|------|-------------|-------|
|------|-------------|-------|

| | | |
|-----|----|----------|
| 200 | OK | No links |
|-----|----|----------|

Procedimientos almacenados

GET

/api/clients/{id}

Obtiene un usuario por su ID. [SQL Function]

POST

/api/clients/CreateClient

Crea un nuevo cliente [SQL Stored Procedure]

PUT

/api/clients/UpdateClient

Actualiza un cliente existente [SQL Stored Procedure]

DELETE

/api/clients/DeleteClient

Elimina un cliente [SQL Stored Procedure]

Este endpoint permite eliminar un cliente específico utilizando su ID. Si el cliente no existe, se retornará un error.

Parameters

Try it out

| Name | Description |
|---------------------------------------|-------------------------------------|
| userId integer(\$int32) (query) | <input type="text" value="userId"/> |

Responses

| Code | Description | Links |
|------------|-------------|-----------------------|
| 200 | OK | No links |
| Media type | | <div>text/plain</div> |

DELETE `/api/clients/DeleteClient` Elimina un cliente [SQL Stored Procedure]

Este endpoint permite eliminar un cliente específico utilizando su ID. Si el cliente no existe, se retornará un error.

Parameters

Cancel

| Name | Description |
|------------------|-------------|
| userId | |
| integer(\$int32) | 13 |
| (query) | |

Execute

Clear

Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7019/api/clients/DeleteClient?userId=13' \
  -H 'accept: text/plain'
```

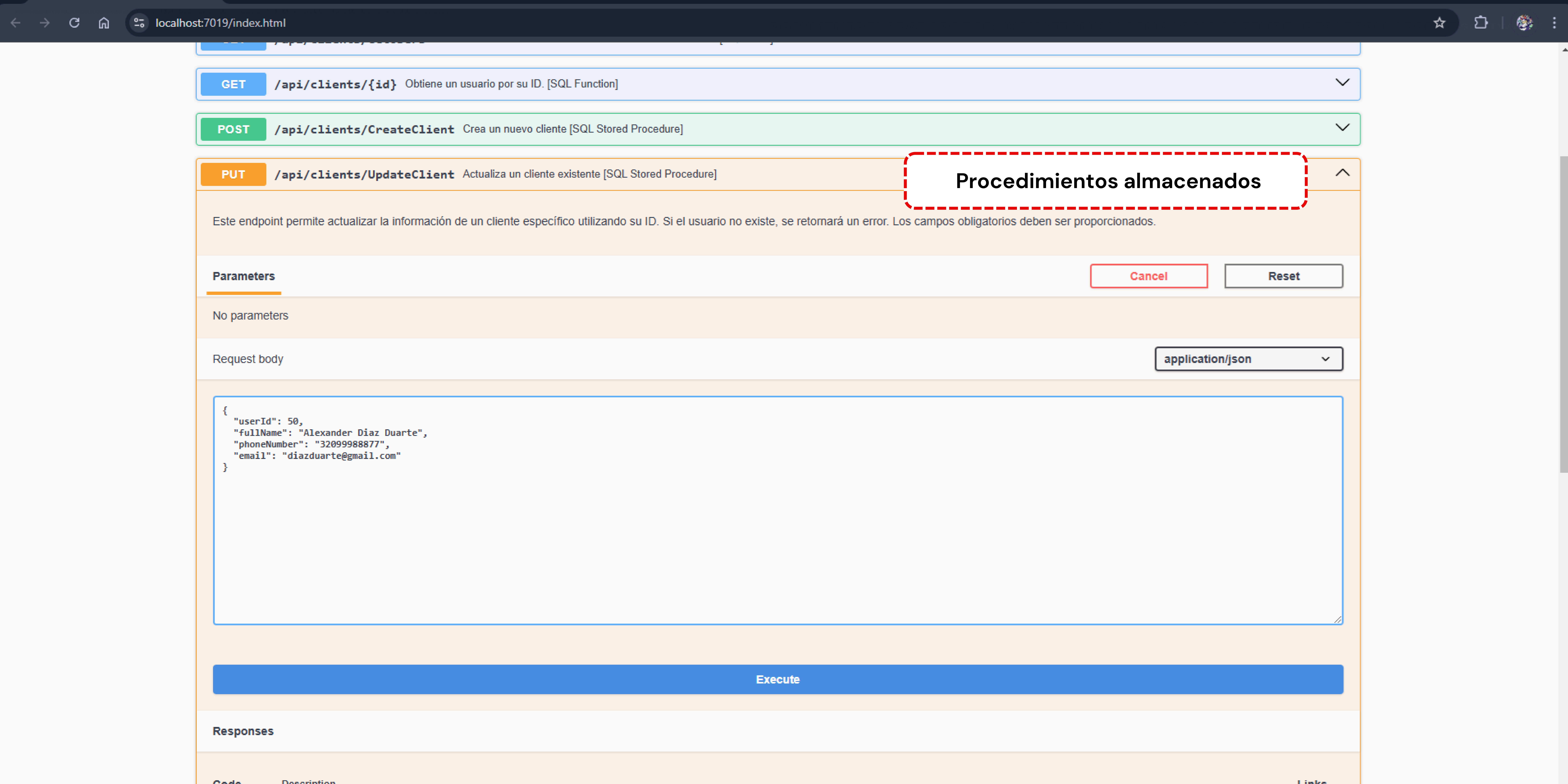


Request URL

<https://localhost:7019/api/clients/DeleteClient?userId=13>

Server response

| Code | Details |
|------|---|
| 200 | <div><div>Response body</div><div><pre>true</pre></div><div><div></div><div>Download</div></div></div> <div><div>Response headers</div><div><pre>content-type: application/json; charset=utf-8 date: Sat, 14 Dec 2024 10:45:04 GMT server: Kestrel</pre></div></div> |



GET

/api/clients/{id} Obtiene un usuario por su ID. [SQL Function]

POST

/api/clients/CreateClient Crea un nuevo cliente [SQL Stored Procedure]

PUT

/api/clients/UpdateClient Actualiza un cliente existente [SQL Stored Procedure]

Procedimientos almacenados

Este endpoint permite actualizar la información de un cliente específico utilizando su ID. Si el usuario no existe, se retornará un error. Los campos obligatorios deben ser proporcionados.

Parameters

Cancel

Reset

No parameters

Request body

application/json

```
{
  "userId": 50,
  "fullName": "Alexander Diaz Duarte",
  "phoneNumber": "32099988877",
  "email": "diazduarte@gmail.com"
}
```

Execute

Responses

Code

Description

Links

2. Borre la cadena de conexion existente de SQL e incorpore, su cadena de conexión, recuerde ejecutar consultas enviadas, como demas funcionalidades SQL, que se encuentran en el repositorio de GitHub, en la carpeta recursos.

1. Ingrese a esta clase appsettings.json