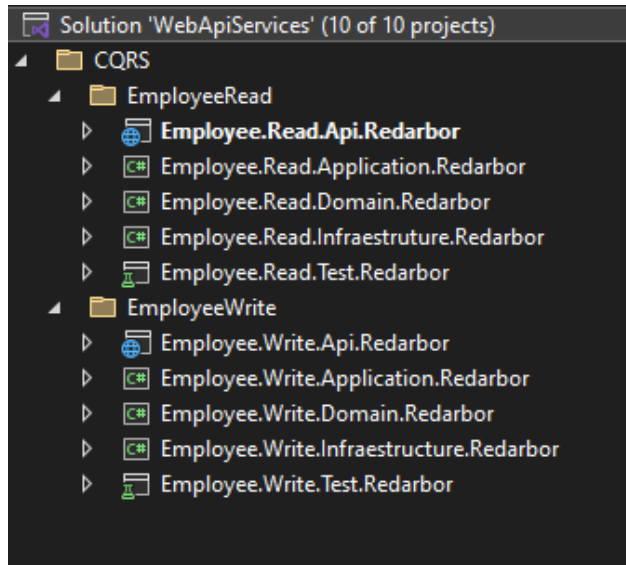


## Especificaciones de la Solución



### Detalle


- Tecnología: .Net 8, SQL Server, Dapepr, EntityFramework
- Patrón CQRS: Se crean dos servicios
  - Servicio de lectura con Dapper
  - Servicio de escritura con EntityFramework
- Se mantienen buenas prácticas de desarrollo: Principios SOLID, DDD, Pruebas unitarias.
- Se emplea autenticación JWT para el consumo de los servicios de escritura

## Pruebas Unitarias

Test	Duration	Traits	Error Message
Employee.Read.Test.Redarbor (2)	144 ms		
Employee.Read.Test.Redarbor (2)	144 ms		
EmployeeReadTests (2)	144 ms		
GetAllEmployees_ReturnsOk_WithListOfEmployees	135 ms		
GetEmployeeById_ReturnsOk_WithEmployee	9 ms		
Employee.Write.Test.Redarbor (6)	200 ms		
Employee.Write.Test.Redarbor (6)	200 ms		
EmployeeWriteTests (6)	200 ms		
CreateEmployee_ReturnsBadRequest_WhenEmployeeIsNull	143 ms		
CreateEmployee_ReturnsOk_WhenEmployeeIsCreated	48 ms		
DeleteEmployee_ReturnsBadRequest_WhenIdIsInvalid	1 ms		
DeleteEmployee_ReturnsOk_WhenEmployeeIsDeleted	4 ms		
UpdateEmployee_ReturnsBadRequest_WhenDataIsInvalid	1 ms		
UpdateEmployee_ReturnsOk_WhenEmployeeIsUpdated	3 ms		

## EmployeeServices

### Servicios de Lectura

 **Swagger**  
Supported by SMARTBEAR

Select a definition Employee.Read.Api.Redarbor v1

# Employee.Read.Api.Redarbor

1.0 OAS 3.0


<https://localhost:7272/swagger/v1/swagger.json>

## Employee

GET /api/Employee

GET /api/Employee/{id}

### Servicios de Escritura

 **Swagger**  
Supported by SMARTBEAR

Select a definition Employee.Write.Api.Redarbor v1

# Employee.Write.Api.Redarbor

1.0 OAS 3.0

<https://localhost:7149/swagger/v1/swagger.json>

## Employee

POST /api/Employee

PUT /api/Employee/{id}

DELETE /api/Employee/{id}

## Casos de Pruebas

### 1. Insertar un nuevo employee

Json utilizado:

```
{
  "CompanyId": 1,
  "CreatedOn": "2000-01-01 00:00:00",
  "DeletedOn": "2000-01-01 00:00:00",
  "Email": "test1@test.test.tmp",
  "Fax": "000.000.000",
  "Name": "test1",
  "Lastlogin": "2000-01-01 00:00:00",
  "Password": "test",
  "PortalId": 1,
  "RoleId": 1,
  "StatusId": 1,
  "Telephone": "000.000.000",
  "UpdatedOn": "2000-01-01 00:00:00",
  "Username": "test1"
}
```

Llamado desde Postman

POST ▼ https://localhost:7149/api/Employee

Params Authorization Headers (8) **Body** ● Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "CompanyId": 1,
3   "CreatedOn": "2000-01-01 00:00:00",
4   "DeletedOn": "2000-01-01 00:00:00",
5   "Email": "test1@test.test.tmp",
6   "Fax": "000.000.000",
7   "Name": "test1",
8   "Lastlogin": "2000-01-01 00:00:00",
9   "Password": "test",
10  "PortalId": 1,
11  "RoleId": 1,
12  "StatusId": 1,
13  "Telephone": "000.000.000",
14  "UpdatedOn": "2000-01-01 00:00:00",
15  "Username": "test1"
16 }
```

Body Cookies Headers (3) Test Results ↺ **200 OK** = 6.54 s = 92 B

## Validar campos obligatorios (CompanyId, Email, Password, PortalId, RoleId, StatusId, Username.)

### Request:

POST ▼ | https://localhost:7149/api/Employee

Params Authorization Headers (8) **Body ●** Scripts Tests Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON** ▼

```
1 {
2   "CreatedOn": "2000-01-01 00:00:00",
3   "DeletedOn": "2000-01-01 00:00:00",
4   "Email": null,
5   "Fax": "000.000.000",
6   "Name": "test2",
7   "Lastlogin": "2000-01-01 00:00:00",
8   "Password": null,
9   "Telephone": "000.000.000",
10  "UpdatedOn": "2000-01-01 00:00:00",
11  "Username": null
12 }
```

Body Cookies Headers (4) Test Results ↺ **400 Bad Request** • 20 ms • 702 B

### Response:

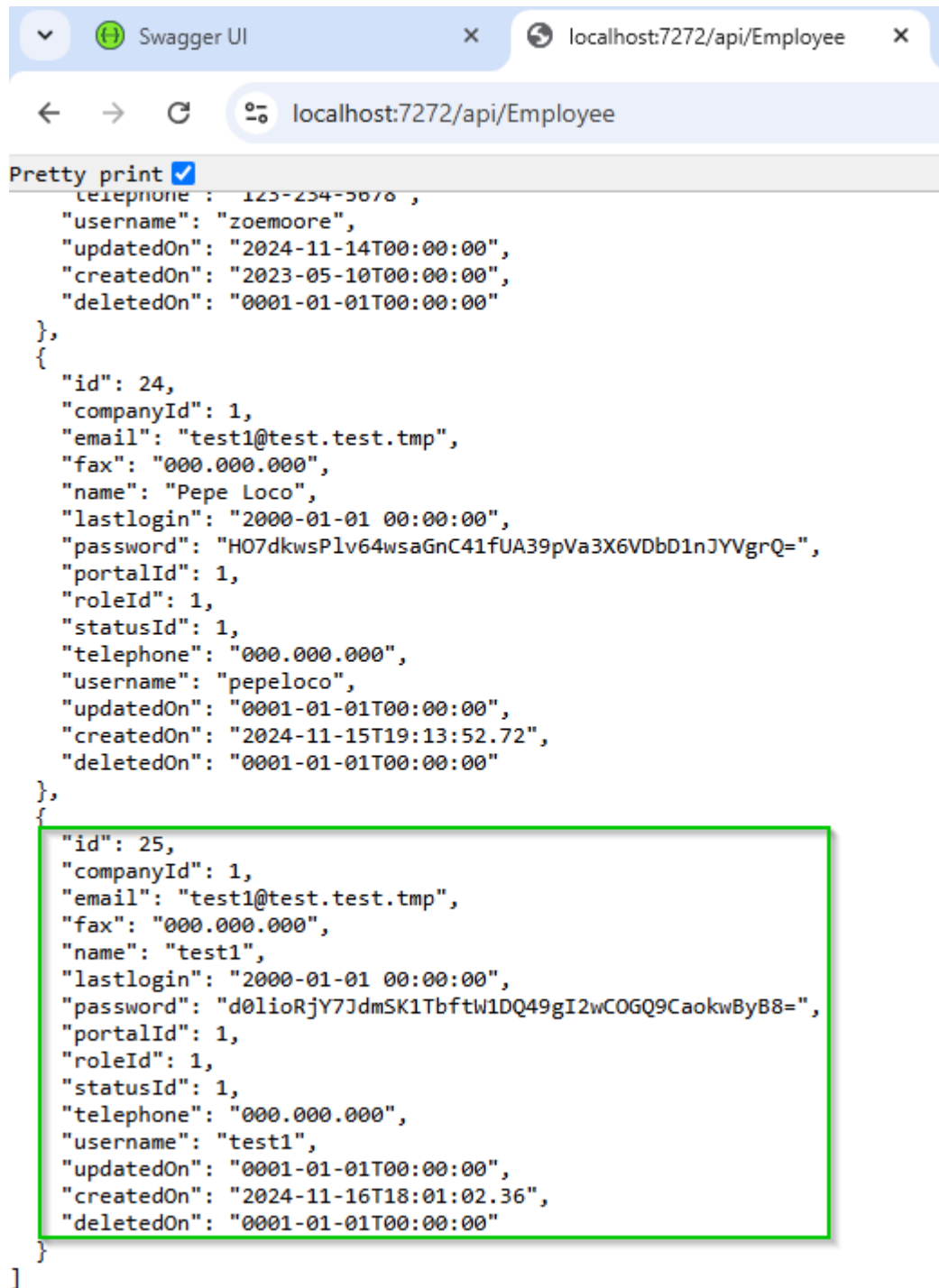
Body Cookies Headers (4) Test Results ↺ **400 Bad Request** • 20 ms

Pretty Raw Preview Visualize **JSON** ▼ ↺

```
5 errors : {
6   "Email": [
7     "The Email field is required."
8   ],
9   "RoleId": [
10    "Please enter a value bigger than 0"
11  ],
12  "Password": [
13    "The Password field is required."
14  ],
15  "PortalId": [
16    "Please enter a value bigger than 0"
17  ],
18  "StatusId": [
19    "Please enter a value bigger than 0"
20  ],
21  "Username": [
22    "The Username field is required."
23  ],
24  "CompanyId": [
25    "Please enter a value bigger than 0"
26  ]
27 }
```

## 2. Validar que está insertado correctamente

### Consulta Get



```
telephone : 123-234-5678 ,
"username": "zoemoore",
"updatedOn": "2024-11-14T00:00:00",
"createdOn": "2023-05-10T00:00:00",
"deletedOn": "0001-01-01T00:00:00"
},
{
  "id": 24,
  "companyId": 1,
  "email": "test1@test.test.tmp",
  "fax": "000.000.000",
  "name": "Pepe Loco",
  "lastlogin": "2000-01-01 00:00:00",
  "password": "H07dkwsPlv64wsaGnC41fUA39pVa3X6VDbD1nJYVgrQ=",
  "portalId": 1,
  "roleId": 1,
  "statusId": 1,
  "telephone": "000.000.000",
  "username": "pepeloco",
  "updatedOn": "0001-01-01T00:00:00",
  "createdOn": "2024-11-15T19:13:52.72",
  "deletedOn": "0001-01-01T00:00:00"
},
{
  "id": 25,
  "companyId": 1,
  "email": "test1@test.test.tmp",
  "fax": "000.000.000",
  "name": "test1",
  "lastlogin": "2000-01-01 00:00:00",
  "password": "d0lioRjY7JdmSK1TbftW1DQ49gI2wCOGQ9CaokwByB8=",
  "portalId": 1,
  "roleId": 1,
  "statusId": 1,
  "telephone": "000.000.000",
  "username": "test1",
  "updatedOn": "0001-01-01T00:00:00",
  "createdOn": "2024-11-16T18:01:02.36",
  "deletedOn": "0001-01-01T00:00:00"
}
]
```

### 3. Validar que está insertado correctamente, consulta por Id

#### Consulta Get/Id

GET

▼

https://localhost:7272/api/Employee/25

ParamsAuthorizationHeaders (6)BodyScriptsTestsSettings

Query Params

	Key	Value
--	-----	-------

bodyCookiesHeaders (4)Test Results

200 OK

PrettyRawPreviewVisualizeJSON▼

1

{

2

"id": 25,

3

"companyId": 1,

4

"email": "test1@test.test.tmp",

5

"fax": "000.000.000",

6

"name": "test1",

7

"lastlogin": "2000-01-01 00:00:00",

8

"password": "d0lioRjY7JdmSK1TbftW1DQ49gI2wCOGQ9CaokwByB8=",

9

"portalId": 1,

10

"roleId": 1,

11

"statusId": 1,

12

"telephone": "000.000.000",

13

"username": "test1",

14

"updatedOn": "0001-01-01T00:00:00",

15

"createdOn": "2024-11-16T18:01:02.36",

16

"deletedOn": "0001-01-01T00:00:00"

17

}

#### 4. Modificar el usuario del registro insertado anteriormente

Nuevo usuario : "test1updated"

PUT

https://localhost:7149/api/Employee/25

Params

Authorization

Headers (8)

Body

Scripts

Tests

Settings

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON

```
1  {
2    |    "Username": "test1updated"
3  }
```

Body

Cookies

Headers (3)

Test Results

200 OK

#### 5. Validar que el registro fue modificado

GET

https://localhost:7272/api/Employee/25

Params

Authorization

Headers (6)

Body

Scripts

Tests

Settings

Query Params

Key	Value
-----	-------

Body

Cookies

Headers (4)

Test Results

200 OK

Pretty

Raw


Preview

Visualize

JSON

```
1  {
2    |    "id": 25,
3    |    "companyId": 1,
4    |    "email": "test1@test.test.tmp",
5    |    "fax": "000.000.000",
6    |    "name": "test1",
7    |    "lastlogin": "2000-01-01 00:00:00",
8    |    "password": "d0lioRjY7JdmSK1TbftW1DQ49gI2wCOGQ9CaokwByB8=",
9    |    "portalId": 1,
10   |    "roleId": 1,
11   |    "statusId": 1,
12   |    "telephone": "000.000.000",
13   |    "username": "test1updated",
14   |    "updatedOn": "2024-11-16T18:29:01.2",
15   |    "createdOn": "2024-11-16T18:01:02.36",
16   |    "deletedOn": "0001-01-01T00:00:00"
17 }
```


## 6. Borrar el nuevo registro

**DELETE**  <https://localhost:7149/api/Employee/25>


Params Authorization Headers (6) Body Scripts Tests Settings

Query Params

	Key	Value
	Key	Value

Body Cookies Headers (3) Test Results  200 OK


## 7. Validar que el registro ya no existe



**GET**  <https://localhost:7272/api/Employee/25>

Params Authorization Headers (6) Body Scripts Tests Settings

Query Params

	Key	Value
--	-----	-------

Body Cookies Headers (4) Test Results  404 Not Found

Pretty Raw Preview Visualize JSON  

```
1 {
2   "type": "https://tools.ietf.org/html/rfc9110#section-15.5.5",
3   "title": "Not Found",
4   "status": 404,
5   "traceId": "00-35b3af2aacea05a8b6642a5d474ccc38-8c1b15b773b6fb69-00"
6 }
```



## Seguridad de los Servicios

Se agrega un medio de autenticación a través de un token de acceso (JWT) para los servicios de escritura.

Llamada a servicio de crear Employee sin enviar token de acceso

The screenshot shows a REST client interface with a POST request to `https://localhost:7149/api/Employee`. The request body is a JSON object with the following fields: `CompanyId`, `CreatedOn`, `DeletedOn`, `Email`, `Fax`, `Name`, `Lastlogin`, `Password`, `PortalId`, and `RoleId`. The response status is `401 Unauthorized`. The response body is a JSON object with a `message` field set to `"Unauthorized"`.

```
1 {
2   "CompanyId": 1,
3   "CreatedOn": "2000-01-01 00:00:00",
4   "DeletedOn": "2000-01-01 00:00:00",
5   "Email": "test3@test.test.tmp",
6   "Fax": "000.000.000",
7   "Name": "test3",
8   "Lastlogin": "2000-01-01 00:00:00",
9   "Password": "test",
10  "PortalId": 1,
11  "RoleId": 1,
12 }
```

```
1 {
2   "message": "Unauthorized"
3 }
```

## Generando token de acceso

The screenshot shows a REST client interface with a POST request to `https://localhost:7149/api/Authentication/authenticate`. The request body is a JSON object with `username` set to `"System"` and `password` set to `"System"`. The response status is `200 OK`. The response body is a JSON object with `name`, `userName`, and `token` fields. The `token` field contains a long JWT string.

```
1 {
2   "username": "System",
3   "password": "System"
4 }
```

```
1 {
2   "name": "System",
3   "userName": "System",
4   "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEiLCJuYmYiOiJlMzZlE4MDQ0NzQsImV4cCI6MTczMjQwOTI3NCwiaWF0IjoxNzIxODc0NDc0fQ.uKV7LQHubm8oxnT5YD-MuWKY0LLAFwnMV0-aUYghK2g"
5 }
```

## Haciendo la llamada con el token

POST

https://localhost:7149/api/Employee

Send

ParamsAuthorizationHeaders (9)BodyScriptsTestsSettingsCookies

Auth Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborative environment, we recommend using variables. Learn more about [variables](#).

Token

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ...

POST

https://localhost:7149/api/Employee

ParamsAuthorizationHeaders (9)BodyScriptsTestsSettings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

```
1  {
2    "CompanyId": 1,
3    "CreatedOn": "2000-01-01 00:00:00",
4    "DeletedOn": "2000-01-01 00:00:00",
5    "Email": "test3@test.test.tmp",
6    "Fax": "000.000.000",
7    "Name": "test3",
8    "Lastlogin": "2000-01-01 00:00:00",
9    "Password": "test",
10   "PortalId": 1,
11   "RoleId": 1,
```

BodyCookiesHeaders (3)Test Results

200 OK