

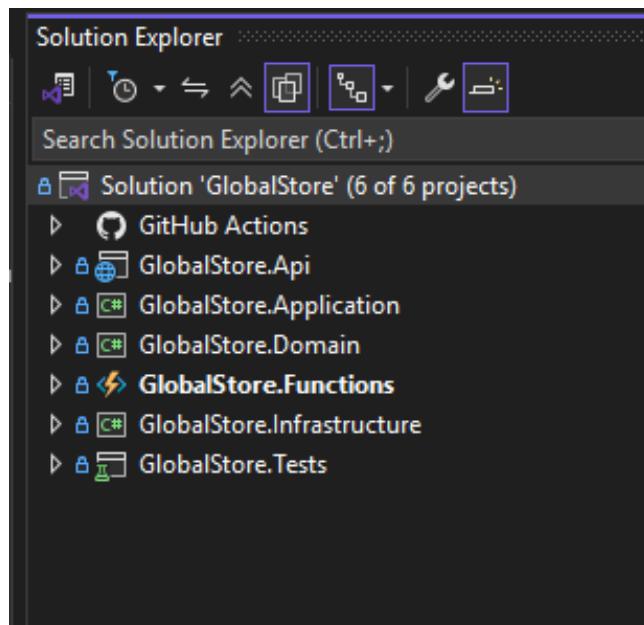
Technical Test - Quartile

Github: <https://github.com/viniciustg/global-store>

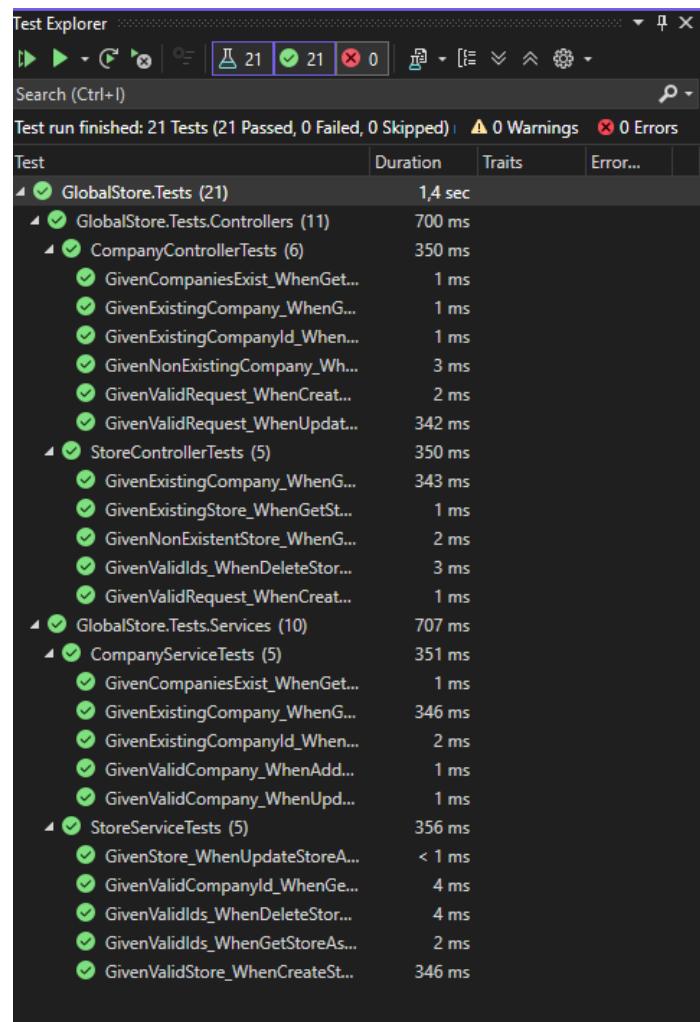
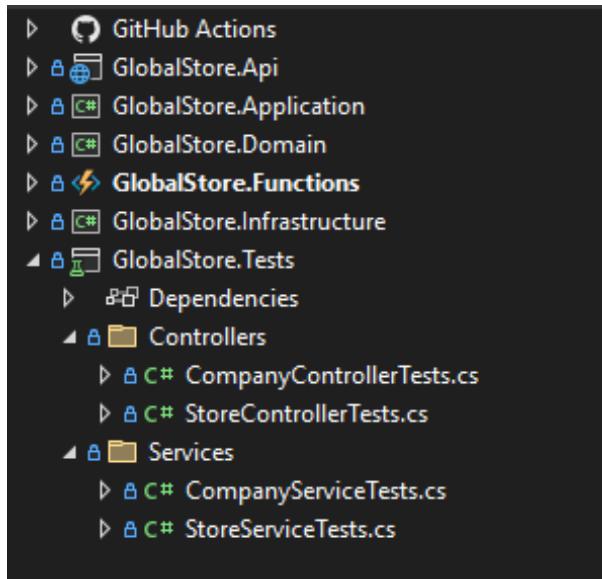
1. Technical Decisions

- .NET 8 for modernity, performance, and Azure compatibility.
- Clean architecture with separation of responsibilities.
- TDD to ensure testability from the start.
- GitHub Actions as CI/CD, free and native to GitHub.
- Automatic deployment to Staging, with the possibility of swapping to Production.
- Azure Functions.

2. Folder Structure



3. 🎉 TDD — Tests



5. 🌐 API RESTful - GlobalStore.Api

- Support for multiple businesses → Each business can have multiple stores.
- Endpoints:

The screenshot shows the Swagger UI for the Store Management API. It includes sections for Company and Store resources, each with a list of HTTP methods and their corresponding URLs and descriptions. The Company section includes endpoints for creating, reading, updating, and deleting companies. The Store section includes endpoints for creating, reading, updating, and deleting stores under a specific company. The UI uses color-coded buttons (blue, green, orange, red) to represent different HTTP methods.

Company	
GET	/api/companies Get all companies.
POST	/api/companies Create a new company.
GET	/api/companies/{id} Get a company by its ID.
PUT	/api/companies/{id} Update an existing company.
DELETE	/api/companies/{id} Delete a company by its ID.

Store	
GET	/api/companies/{companyId}/stores Get all stores for a specific company.
POST	/api/companies/{companyId}/stores Create a new store under a specific company.
GET	/api/companies/{companyId}/stores/{storeId} Get a specific store by its ID.
PUT	/api/companies/{companyId}/stores/{storeId} Update a store's information.
DELETE	/api/companies/{companyId}/stores/{storeId} Delete a store by its ID.

The screenshot shows the schema definitions for the GlobalStore.Api. It lists five schema types: CompanyRequest, CompanyResponse, ProblemDetails, StoreRequest, and StoreResponse, each with a corresponding icon and a 'View' link.

Schemas	
CompanyRequest	>
CompanyResponse	>
ProblemDetails	>
StoreRequest	>
StoreResponse	>

6. 🏠 Azure Function - GlobalStore.Functions

- Connected to the same SQL Server as the API.
- Endpoints via HTTP trigger:

```

Azure Functions Core Tools
Core Tools Version:      4.0.7512 Commit hash: N/A +8be8cc84f6ad64c784e083bf4da7fa381bdd3449 (64-bit)
Function Runtime Version: 4.1040.300.25317

[2025-07-13T21:34:28.321Z] Found C:\Projetos\global-store\GlobalStore.Functions\GlobalStore.Functions.csproj. Using for
user secrets file configuration.
[2025-07-13T21:34:30.501Z] Azure Functions .NET Worker (PID: 21840) initialized in debug mode. Waiting for debugger to a
ttach...
[2025-07-13T21:34:30.532Z] Worker process started and initialized.

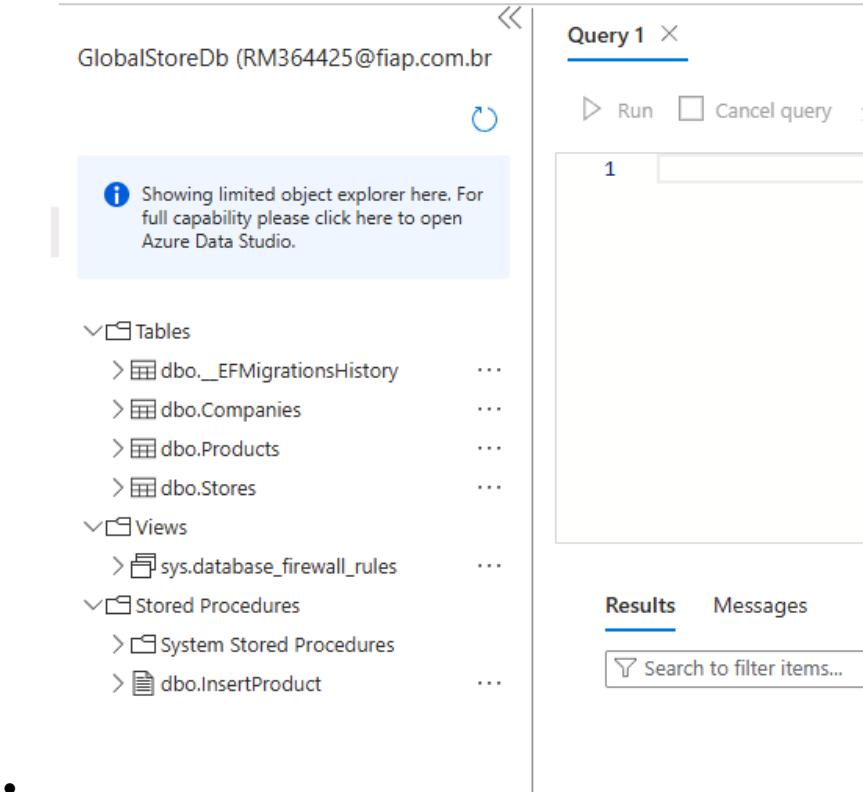
Functions:

    CreateProduct: [POST] http://localhost:7225/api/products
    DeleteProduct: [DELETE] http://localhost:7225/api/products/{id:int}
    GetProductById: [GET] http://localhost:7225/api/products/{id:int}
    GetProductsList: [GET] http://localhost:7225/api/products
    UpdateProduct: [PUT] http://localhost:7225/api/products/{id:int}

```

7. SQL Server: Tables / Function / Procedure

- Products table with fields ID, Name, Price, Stock, etc.
- Scalar Function: Returns JSON of a product by ID.
- Stored Procedure: Returns a list of products with optional filters.



The screenshot shows the Azure Data Studio interface with the following details:

- Top Bar:** Shows the connection name "GlobalStoreDb (RM364425@fiap.com.br)" and a "Run" button.
- Message Bar:** A tooltip message: "Showing limited object explorer here. For full capability please click here to open Azure Data Studio."
- Object Explorer:** A tree view of database objects:
 - Tables:** dbo._EFMigrationsHistory, dbo.Companies, dbo.Products, dbo.Stores
 - Views:** sys.database_firewall_rules
 - Stored Procedures:** System Stored Procedures, dbo.InsertProduct
- Query Editor:** Titled "Query 1", with a "Run" button and a results pane containing the number "1".
- Bottom Navigation:** Buttons for "Results" (selected), "Messages", and a search bar "Search to filter items...".

8. 🌐 CI/CD with GitHub Actions

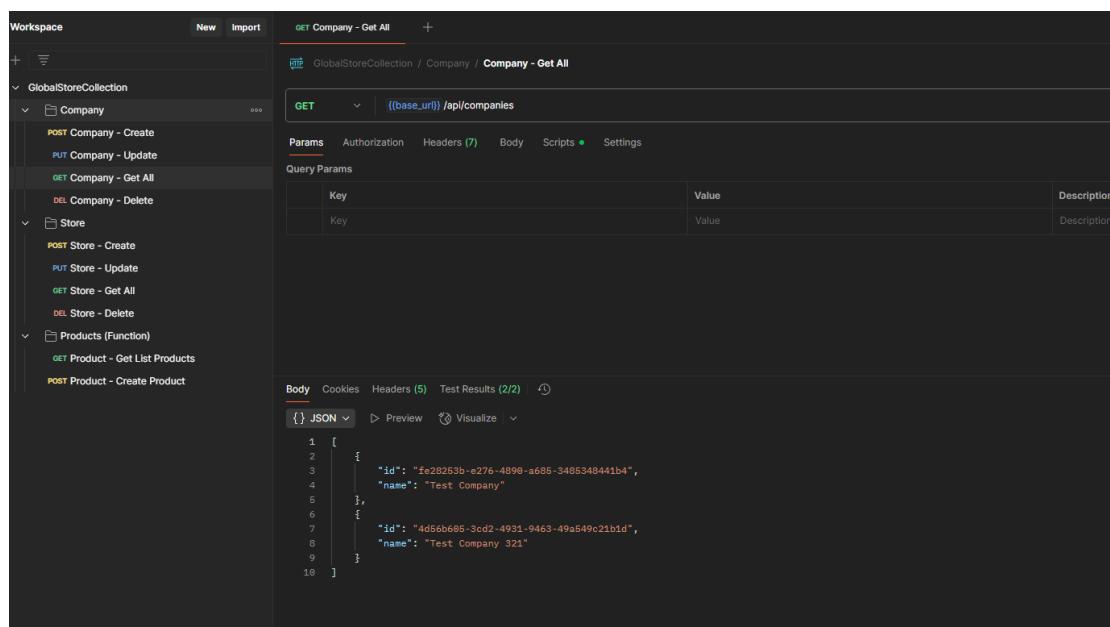
- Workflow em .github/workflows/deploy.yml and deploy-function.yml

The screenshot shows the GitHub Actions interface. On the left, there's a sidebar with options like 'Actions', 'New workflow', 'All workflows' (which is selected), 'Build and deploy ASP.NET Core app to ...', 'Deploy Azure Function - GlobalStore.Fu...', 'Management', 'Caches', 'Deployments', 'Attestations', 'Runners', 'Usage metrics', and 'Performance metrics'. The main area is titled 'All workflows' and shows '23 workflow runs'. A search bar at the top right says 'Filter workflow runs'. Below it, there are five workflow run cards, each with a green checkmark icon, the name of the workflow (e.g., 'feat: removing AuthorizationLevel to help tests'), a brief description, the branch ('main'), and the time it was run (e.g., '21 minutes ago').

This screenshot shows a detailed view of a GitHub Action run. At the top, it says 'Summary' and lists 'Jobs' (with 'build' and 'deploy' checked) and 'Run details' (with 'Usage' and 'Workflow file' listed). To the right, it provides summary statistics: 'Triggered via push 21 minutes ago', 'Status Success', 'Total duration 1m 17s', and 'Artifacts 1'. Below this, it shows the 'deploy.yml' configuration: 'on: push'. It then displays a timeline of the steps: a 'build' step (42s) followed by a 'deploy' step (28s), both of which are marked with green checkmarks.

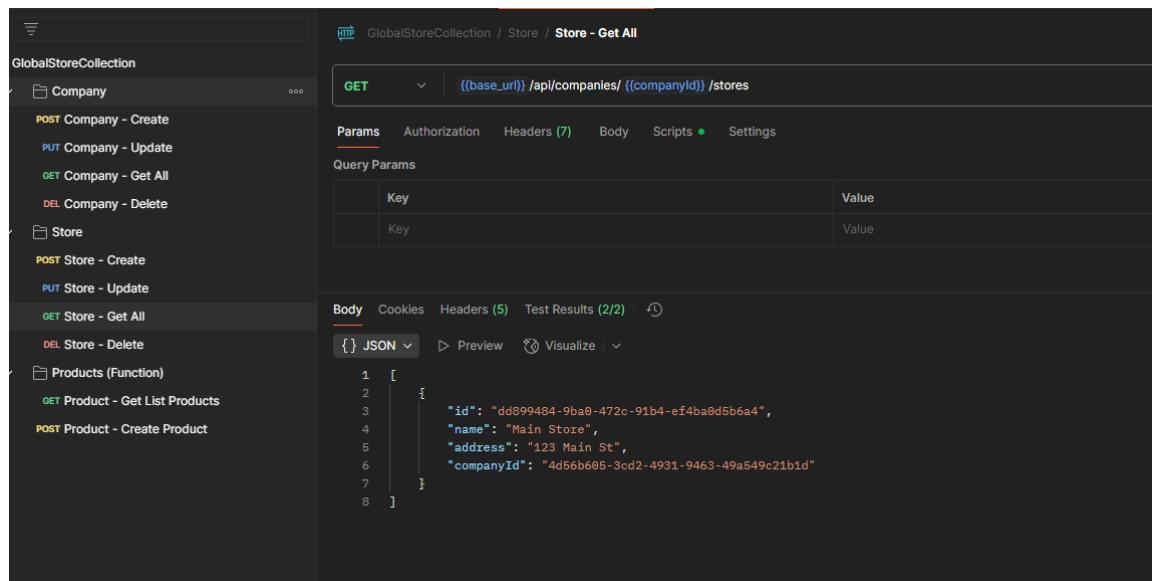
9. Evidences (prints)

Postman – Get Companies



The screenshot shows the Postman interface with a 'GET Company - Get All' request selected. The URL is `http://GlobalStoreCollection/Company/Company - Get All`. The 'Headers' tab is active, showing a single header `Content-Type: application/json`. The 'Body' tab shows a JSON response with two company objects:

```
[{"id": "fe28253b-e276-4890-a685-3485348441b4", "name": "Test Company"}, {"id": "4d56b685-3cd2-4931-9463-49a549c21b1d", "name": "Test Company 321"}]
```



The screenshot shows the Postman interface with a 'GET Store - Get All' request selected. The URL is `http://GlobalStoreCollection/Store/Store - Get All`. The 'Headers' tab is active, showing a single header `Content-Type: application/json`. The 'Body' tab shows a JSON response with one store object:

```
[{"id": "dd899484-9ba0-472c-91b4-ef4ba0d5b6a4", "name": "Main Store", "address": "123 Main St", "companyId": "4d56b685-3cd2-4931-9463-49a549c21b1d"}]
```

The screenshot shows a REST API testing interface with the following details:

- URL:** GlobalStoreCollection / Products (Function) / Product - Get List Products
- Method:** GET
- Auth Type:** No Auth
- Body:** JSON (Preview and Visualize tabs available)
- Response Headers:** 200 OK, 2.48 s, 703 B
- Response Body (JSON):**

```
1 [  
2 [  
3 {  
4     "id": 1,  
5     "companyId": "f1a8d8e7-0a0b-4d6e-b2c4-1e8a6fcbb5ff",  
6     "storeId": "c7b2b4fc-3ef9-4ea8-bcd8-28d9f78ae4f5",  
7     "name": "Teclado Mecânico RGB",  
8     "description": "Inclui com switches azuis e iluminação RGB customizável",  
9     "price": 349.99,  
10    "createdAt": "2025-07-13T21:28:54.627",  
11    "updatedAt": null  
12 },  
13 {  
14     "id": 2,  
15     "companyId": "f1a8d8e7-0a0b-4d6e-b2c4-1e8a6fcbb5ff",  
16     "storeId": "c7b2b4fc-3ef9-4ea8-bcd8-28d9f78ae4f5",  
17     "name": "Teclado Mecânico",  
18     "description": "Inclui com switches RGB customizável",  
19     "price": 349.99,  
20     "createdAt": "2025-07-13T21:32:55.227",  
21     "updatedAt": null  
22 ]]
```