DevOps2022G-JJD

👨 **Members**

Aa Name	@ E-Mail
João Gomes	1200365@isep.ipp.pt
João Brito	1200363isep.ipp.pt
Dayana Moreira	1191809@isep.ipp.pt

Project Summary

- Project made for the DOS subject at the request of Professor Diogo Amaral to start the practice with .NET CORE and versioning projects.
- The project consists of building an **API** with the help of **.NET CORE**, a Microsoft framework.
- Basically, the API endpoints are related to **Buildings** and **Floors**, where the respective **CRUDs** will be performed

---

Project Phases

Phase 1 - Initiation

- At this stage, the project members were brought together and the problem was interpreted.
- The repository was also created and the entire development environment was configured.

Phase 2 - Planning

- At this stage, the entire code given by the teacher was read and interpreted.
- Regarding the philosophy of using git, it was at this stage that we decided to divide the tasks into their respective parts.

Branch structure

- We decided to split into branches and not just use the master as that way we use best practices and avoid conflicts and non-functional/non-tested code in "production".

  Master : In the master branches is all the code that would be in production, that is, tested and 100% functional

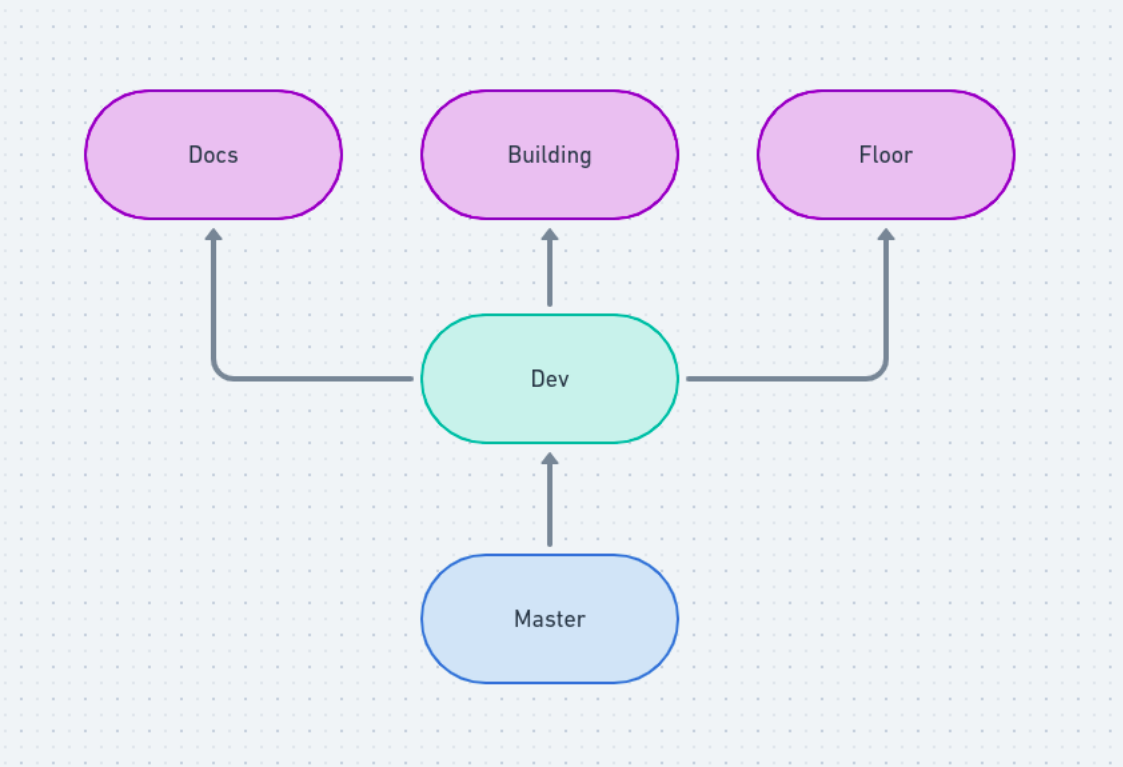
  Dev : The "Dev" branch represents the entire development process, and each new feature will be created on a branch from the dev branch

  Building: The "Building" branch is responsible for all the features responsible for the "Building" domain entity

  Building: The "Floor" branch is responsible for all the features responsible for the "Floor" domain entity

Docs: This branch is responsible for the documentation process of the project that will be linked to it, that is, all files related to documentation will be carried out and approved in this branch.

- The choice for this approach to working with git was chosen because we think it is the simplest, but it maintains good practice and remains scalable.



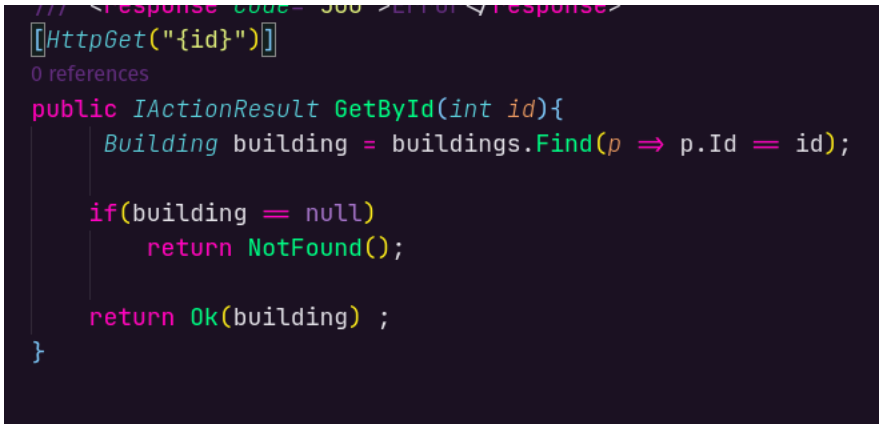
Git Branch Structure

Phase 3 - Execution

- At this stage, the planning was carried out, the entire development process was carried out at this stage.

Development process

- The development process aimed to complete all the methods proposed by the teacher. Basically all the CRUD of the project entities.
- The methods were written with the help of the microsoft documentation, we decided to do so because it is very well done in an objective way.
    - **Example**



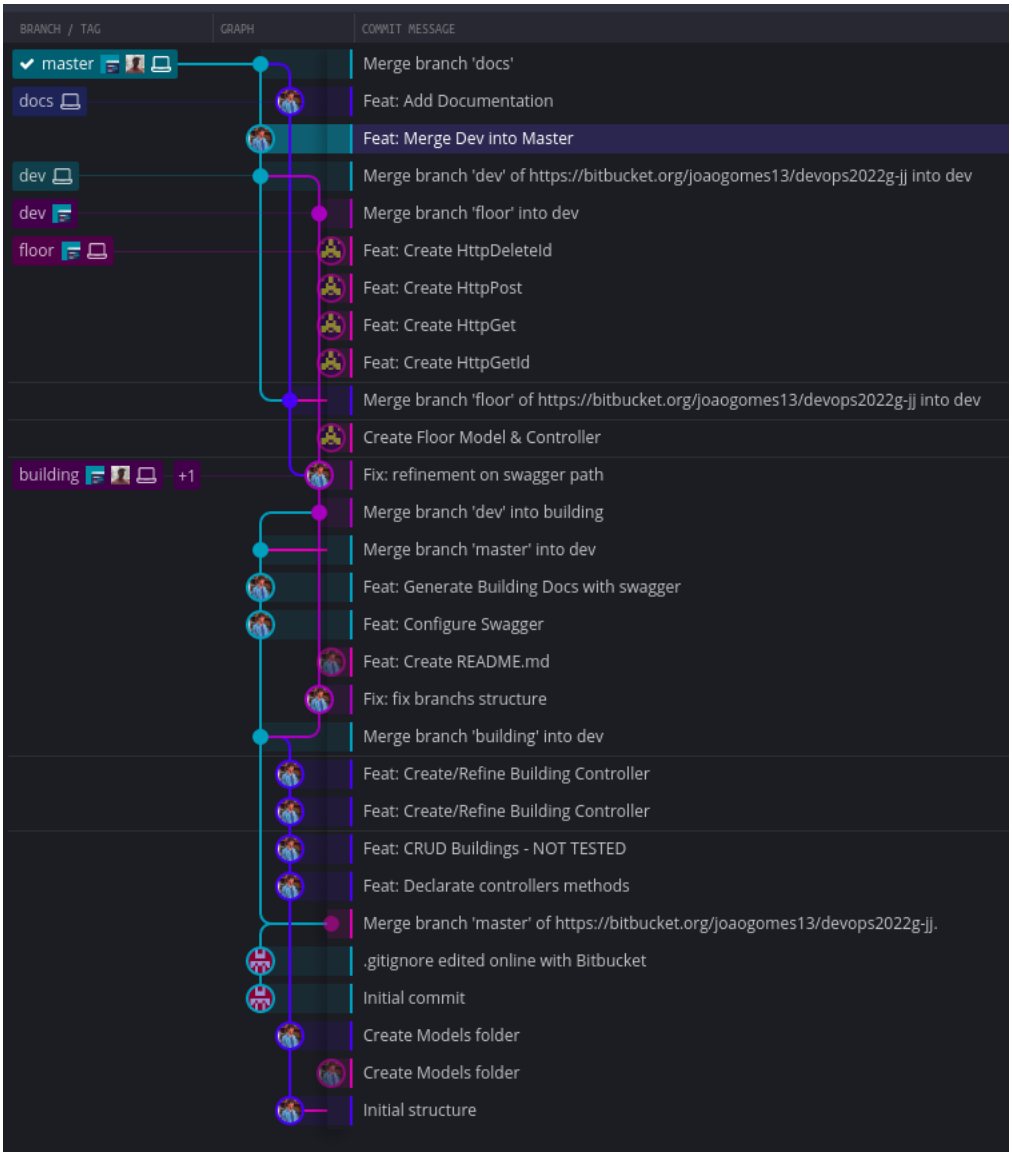
- The choice of the name of the variables was thought of in the understanding of the whole group, following good practices, that is, in English, and the variables written with the Camel Case pattern.

Tools

- The tools that were used for the development of the project were:
  - **VS Code with pack extensions for .Net CORE**
    - We chose VS Code because the operating system that most members of the group use is Linux, and the main IDE for .NET (Visuial Studio) is not available for it.
  - **Dotnet CLI**
    - The use of the dotnet CLI was fundamental for the project's productivity, and with simple commands we can give the project **build** and **run**

  This development environment pleased us immensely, as we were able to implement the code and execute it very easily. Without a doubt a very good long-term environment.

Commit tree



Phase 4 - Documentation

- For the construction of the project documentation, we used the Notion tool, a very practical application of notes in Markdown, with support for PDF export.
- Endpoint documentation with Swagger.
  - We decided to implement the swagger as it is an excellent way to test and document the built endpoints.
  - To access it you should:
    - Clone the repository
    - Build and Run the project
    - Access the http://localhost:5000 and you be redirected to Swagger page

---

Remote Repositories

Github

GitHub - JoaoGomes5/DevOps2022G-JJD: 🚧 Project carried out for the Software Development and Operations subject to improve the practice of versio

🚧 Project carried out for the Software Development and Operations subject to improve the practice of version control and development in .NET Core 🐙 - GitHub - JoaoGom
for the Software Development and Operations subject to improve the practice of version control and development in .NET Core 🐙

https://github.com/JoaoGomes5/DevOps2022G-JJD

Bit Bucket

Bitbucket

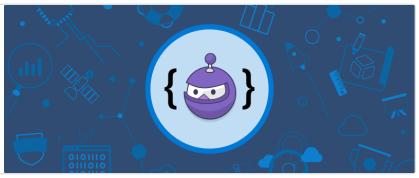
https://bitbucket.org/joaogomes13/devops2022g-jjd

Sources cited

Criar uma API Web com o ASP.NET Core - Learn

Crie um serviço RESTful com ASP.NET Core que suporte operações CRUD (Create, Read, Update, Delete). Neste módulo, vai: Criar um projeto de API Web com o ASP.NET Core. Criar uma base de dados dentro da memória para manter produtos. Adicionar suporte para operações CRUD.

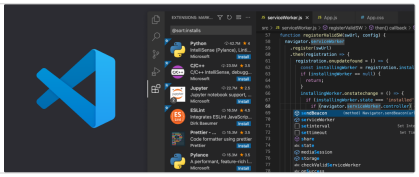
https://docs.microsoft.com/pt-pt/learn/modules/build-web-api-aspnet-core/



Visual Studio Code - Code Editing. Redefined

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favorite platform - Linux, macOS, and Windows.

https://code.visualstudio.com/



.NET Core Extension Pack - Visual Studio Marketplace

This extension pack packages some of the most popular (and some of my favorite) .NET Core related extensions. If you like it, please leave your Rating & Review and share with your friends. If you have any idea on how to improve this extension pack, I'm looking forwarded to hear from you.

https://marketplace.visualstudio.com/items?itemName=doggy8088.netcore-extension-pack

Documentation

https://swagger.io/docs/