Summary

1 Introduction

This document is intended to standardize operations carried out in the project's back/front-end development.

2 GitHub

2.1 Commits

Language

"Commits" must be made in English.

Message

The "commits" messages must be made according to the "feature" in question and must be well described (for the evaluator's understanding). Furthermore, messages must be written in the present verb. Below is the list of message prefixes.

- feat: 'message' *{*When it is a new "feature" use the prefix feat*}*
- fix: 'message' *{*When correcting a functionality*}*
- docs: 'message' *{*When adding or updating documentation*}*
- refactor: 'message' *{*When refactoring some functionality *}*
- test: 'message' *{* When creating/editing related to tests *}*
- hotfix: 'message' *{* When it is a fix in production*}*

One example: git commit -m 'feat: add new endpoint for list all zones"

2.2 Branches

Branch names must be obtained directly from the GitHub card and formatted in lowercase. You must prefix the branch name with the modification type, separated by a forward slash ('/'). If the card title is extensive, abbreviate it while preserving the main subject. Here is an example of how to generate the branch name:

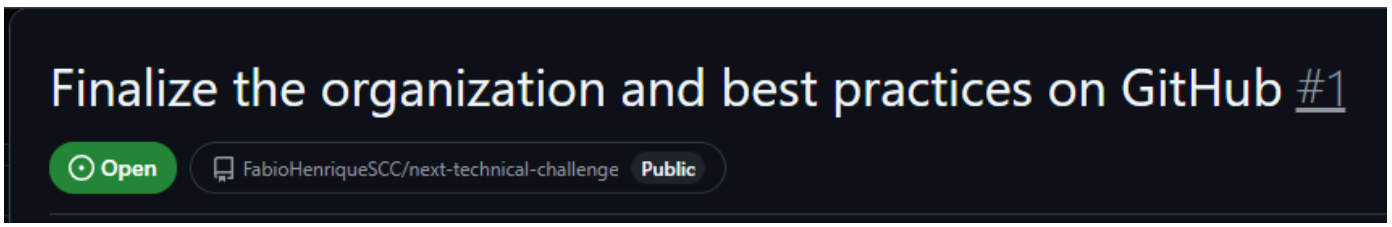

Figura 1: Example of a card on Github

In the example above the name of branch would be "docs/finalize-the-organization-github".

2.3 Pull Request

Language

The “Pull requests” must be made in English.

Message

The “pull request” must be made based on the template available below:

Pull Request Template

Description

This section provides a brief overview of the changes introduced in the PR. It should be concise yet informative enough to give reviewers an idea of what has been done.

Example:

- **Description**: "Created the front-end project using React."

Card/Issue Link

Include a link to the related card or issue in your project management tool. This ensures that the PR is linked to a tracked item and gives context to the work done.

Example:

- **Card/Issue**: "link_from_your_card"

Type of Change

This section is a checklist where the developer indicates the type of change introduced by the PR. Only one option should be selected:

- **Bug fix**: A change that addresses a bug or issue.
- **New feature**: Implementation of a new feature or functionality.
- **Performance improvement**: Enhancements aimed at optimizing the performance of the application.
- **Code refactor**: Changes made to improve the structure of the code without altering its functionality.
- **Documentation update**: Updates to documentation related to the project.

Example:

- **Type of change**: "[x] New feature"

How to Test

In this section, provide instructions on how to test the changes introduced by the PR. If no testing is required, state "None."

Example:

- **How to test**: "None."

Checklist

A checklist to ensure that all necessary steps have been taken before submitting the PR. Each point should be checked off to confirm completion:

- **Code follows the project standards and conventions**: Ensure that the code adheres to the project's coding standards and guidelines.
- **I made all tests**: Confirm that all relevant tests have been created and passed successfully.
- **Documentation has been updated if necessary**: Verify that any required updates to documentation have been made.
- **Code review completed**: Ensure that the code has been reviewed by another team member.

Example:

- **Checklist**:
    - Code follows the project standards and conventions.
    - I made all tests.
    - Documentation has been updated if necessary.
    - Code review completed.

Additional Information

Any extra information relevant to the PR can be included in this section. This might include known issues, limitations, or future considerations.

Example:

- **Additional information**: "None."

Flow

The Pull Request must be done from the **target branch** to the **Dev** branch, once the PR is approved by a reviewer, the Merge will be performed. After that, a PR will be opened from the **Dev** branch to the **Main** branch once by week, preferably on Monday.

Hotfix

When there is a serious error in production, a branch must be opened in this 'hotfix/ feature-in-question' format. And the Pull Request must be done directly in the 'main' branch. After opening the PR, request review from a developer.

Note: update the DEV branch after this procedure.

2.4 Reviewers

Reviewers must always pay attention to reviews, observing code readability, improvements, security, among others. Since the reviewer's idea is to help deliver optimized code.

2.5 Deploy

To deploy, you must first analyze the changes that are coming in. If these are changes that do not introduce new functionalities:

- Create a PR from the 'dev' branch to the 'main' branch;
- Send the PR to another developer for approval;
- Merge.

e.g.: bug fixes, tests and changes to endpoints that do not affect usability.