

**MIEIC – December 2017**

**Software Development in Practice**

**ESOF**

**Class 1 - Group F**

**Diogo Peixoto Pereira – up201504326**

**Maria Eduarda Santos Cunha – up201506524**

**Pedro Miguel Ferraz Nogueira da Silva – up201505460**

Index

[1. Open Source Project 3](#_Toc500334821)

[1.1. Why did we choose this Project? 3](#_Toc500334822)

[1.2. How alive is this project? 3](#_Toc500334823)

[1.3. How important is it? 3](#_Toc500334824)

[1.4. What is it good for? 3](#_Toc500334825)

[1.5. What are the technologies involved? 3](#_Toc500334826)

[1.6. Are there many issues to fix? 3](#_Toc500334827)

[2. Issues 4](#_Toc500334828)

[2.1. Issue 1 4](#_Toc500334829)

[2.2. Issue 2 4](#_Toc500334830)

# Open Source Project

## Why did we choose this Project?

Visual Studio Code was launched on the 29th of April, in 2015 and was developed by Microsoft.

One of the reasons for the class to choose this project is because it was "alive" enough for us to get some feedback from the developers.

Also, it had a tremendous amount of issues to be fixed. That means that there are a lot of options for all groups to choose an issue and fix it.

## How alive is this project?

Visual Studio Code has almost daily updates and the community seems to be very active.

Since its creation it’s been famously growing in the developing community.

## How important is it?

Visual Studio is becoming more and more used and loved among programmers because it has a lot of important and useful tools such as Git, IntelliSense, auto-completion and many more that help the life of many programmers.

## What is it good for?

Editing, building and debugging with ease.

It is able to incorporate different languages in the same code editor, with multiple extensions.

IntelliSense TM Technology allows for easier coding and auto-completion.

## What are the technologies involved?

Visual Studio Code is developed using TypeScript as the programming language. It also uses a well-known library called Electron which allows the creation of a desktop application using web based programming languages such as JavaScript, HTML and CSS for all platforms (Windows, Linux and MacOS).

## Are there many issues to fix?

There are currently around 4000 open issues.

# Issues

## Issue 1

**Issue:** Allowing the use of ‘;’ as separator between files to include/exclude when searching.

#39549

<https://github.com/Microsoft/vscode/issues/39549>

study it, and describe it very shortly, compiling the most relevant information you found about it.

**Requirements:** if applicable, describe the requirements of the application (web, mobile, desktop) underlying the issue, in its context. You can describe it in the form of feature descriptions, user stories, use cases, and if applicable add screenshots.

**Source code files:** list the names of the files and code fragments directly and indirectly involved with the issue.

**Design of the fix:** do a very simple UML diagram that helps on illustrating how you plan to fix the issue. You may use one or all of the UML diagrams you learned already: classes, sequence, states, etc.

**Fix** **source** **code:** describe the code you wrote to fix the issue, packaging as a patch-file.

**Submit the fix:** after approval by your teacher, submit the fix to the project, as a pull-request, or a patch, depending on how it is supposed to submit fixes in your project.

## Issue 2

**Issue:** Html code completion for scripts

#39606

<https://github.com/Microsoft/vscode/issues/39606>

#39839

<https://github.com/Microsoft/vscode/pull/39839>

using the issue system of the project (Github, Bitbucket, Track, etc), study it, and describe it very shortly, compiling the most relevant information you foind about it.

**Requirements:** if applicable, describe the requirements of the application (web, mobile, desktop) underlying the issue, in its context. You can describe it in the form of feature descriptions, user stories, use cases, and if applicable add screenshots.

**Source code files:** list the names of the files and code fragments directly and indirectly involved with the issue.

**Design of the fix:** do a very simple UML diagram that helps on illustrating how you plan to fix the issue. You may use one or all of the UML diagrams you learned already: classes, sequence, states, etc.

**Fix** **source** **code:** describe the code you wrote to fix the issue, packaging as a patch-file.

**Submit the fix:** after approval by your teacher, submit the fix to the project, as a pull-request, or a patch, depending on how it is supposed to submit fixes in your project.