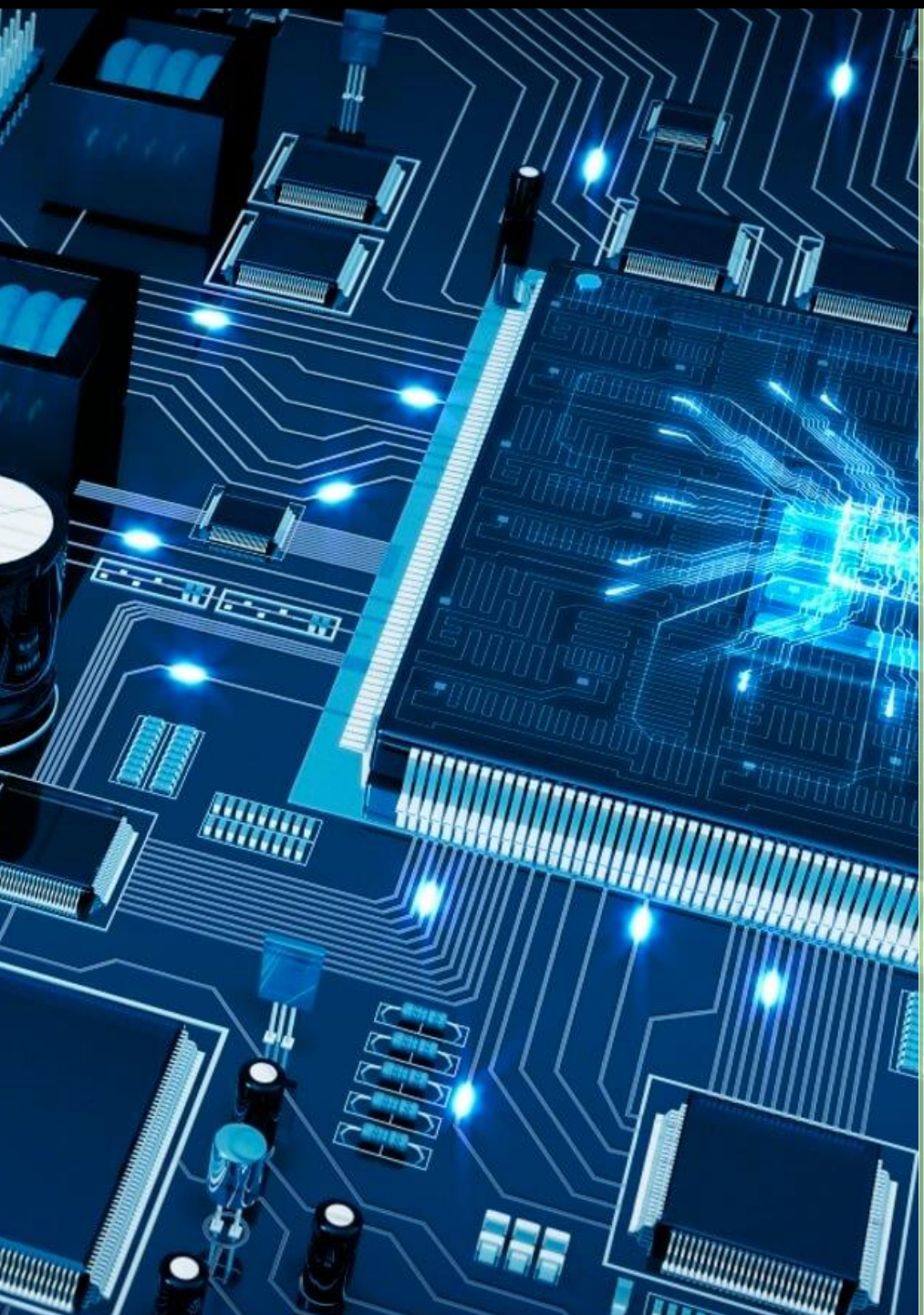
Compiler c202-jubilados

USER MANUAL



Project Manager
Ruiz Aguilar Eduardo

System Architect
Rodriguez Garcia Dulce Coral

System Tester
Hernández Escobar Oswaldo

System Integrator
Aguilera Ortiz Alfredo

INDEX

1. PURPOSE OF MANUAL 2

2. PARTICIPANTS 2

3. PURPOSE 2

4. USER MANUAL 2

4.1. INSTALLATION REQUIREMENTS 2

4.2. INSTALLATION 3

4.3. SETTING 4

4.4. EXECUTION 4

4.5. RELEASE DOWNLOAD..... 5

1. Purpose of manual

This document is intended as a user friendly, visual step-by-step guide to use our compiler, from finding it online, to downloading it and finally running it on your own PC without a problem. At this point, we find it important to mention that you need to have some basic background knowledge on compilers theory or at least some interested in this topic, be familiar with windows command line or terminal and finally but not less important, know how to use git software.

We pretend to display or present our information in a detailed, ordered and understandable way, so that any user interested in our project can achieve their goal whatever it is, either learning from it, running it or using it as a reference.

As a complement at the end of this file we added a table with some of the commands that we used in git. As well as another table that contains the commands that we used for the manipulation of the project.

2.Participants

Project Manager
Ruiz Aguilar Eduardo

System Architect
Rodriguez Garcia Dulce Coral

System Tester
Hernández Escobar Oswaldo

System Integrator
Aguilera Ortiz Alfredo

3.Purpose

Learn and analyze the basic stages of operation of a C compiler, as well as a quick review of abstract syntax trees (AST), assembly language, in order to develop our own compiler, using the elixir programming language.

4. User manual

- **Installation requirements**

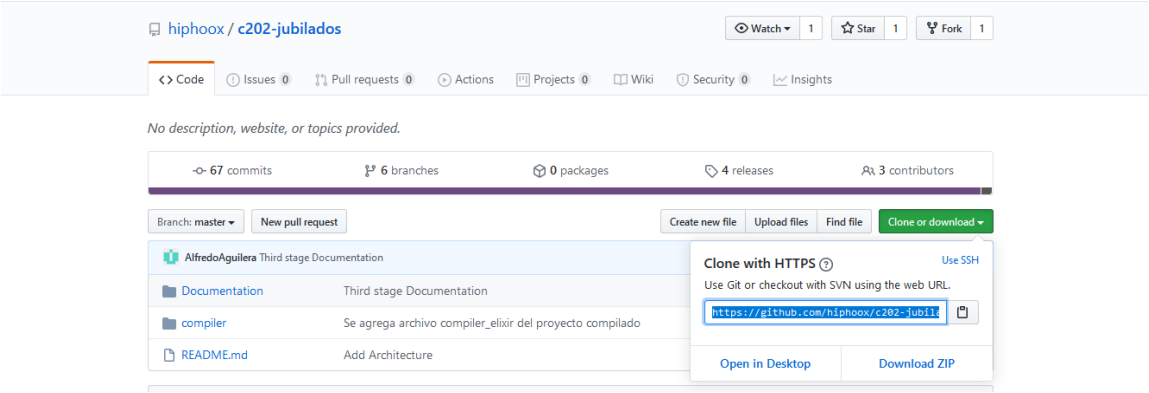
  -Have a text editor for code (Optional)
  -Have a github account https://github.com/join?source=login
  -Have git installed https://git-scm.com/download/win

-Have Elixir installed https://elixir-lang.org/install.html

- **Installation**

The code of our compiler is currently in the remote repository of the GitHub platform, for this reason we need to have installed git in our pc to download the repository. We must open our terminal or console, move to the path where we we want to manage our files, we type the *git clone [path]* command (being *path* the address below) in our terminal and we bring the contents of the repository as shown below:

https://github.com/hiphoox/c202-jubilados.git



Once these steps have been elaborated, we can see that I have already created a directory with the name of our repository.

Now to enter the local repository from terminal or console, type “cd” followed by the path where the repository folder is found, this way the contents of the folder can be visualize.

When opening our folder you can display the content, which is:

```
 Directorio de C:\Users\alfre\Documents\GitHub\c202-jubilados

17/06/2020  18:08    <DIR>          .
17/06/2020  18:08    <DIR>          ..
17/06/2020  18:08    <DIR>          compiler
17/06/2020  18:08    <DIR>          Documentation
17/06/2020  18:08               356 README.md
               1 archivos            356 bytes
               4 dirs  313.543.208.960 bytes libres

C:\Users\alfre\Documents\GitHub\c202-jubilados>
```

Documentation: This folder contains the project work planning, scopes, presentations and user manual.

Compiler: Inside this folder you can find the latest version of our compiler code, we can see that this is where the program executable will be generated. Inside of it is:

- **lib: This folder has the code, contains all the modules written in elixir programming language**
- **test:** Here we have the Unit tests for the compiler
- **examples:** Contains valid and invalid c programs for our compiler

This time we will move to the compiler folder

- **Setting**

  Once we are inside the folder we have to generate the executable, we will do this with the mix tool, we will execute it as shown below:

```
Símbolo del sistema

C:\Users\alfre\Documents\GitHub\c202-jubilados\compiler>mix escript.build
Generated escript compiler_elixir with MIX_ENV=dev

C:\Users\alfre\Documents\GitHub\c202-jubilados\compiler>
```

- **Execution**

After using the mix command, a new file with **.out** termination will be generated, which we will have to execute as follows

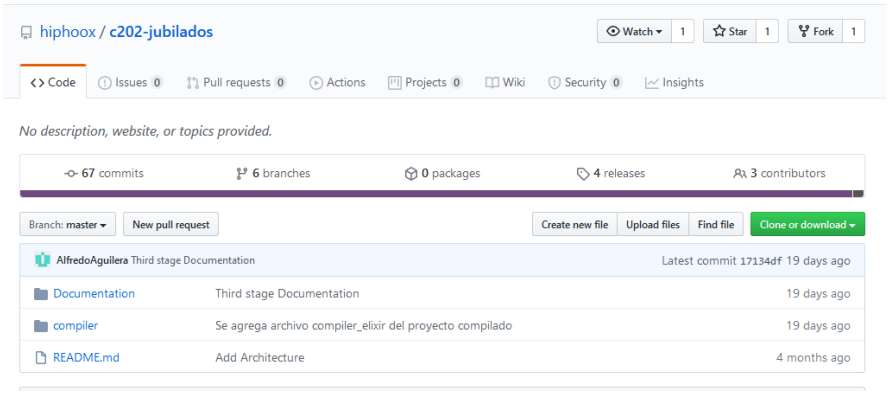
```
Símbolo del sistema                                          Tiempo restante de la

:\Users\alfre\Documents\GitHub\c202-jubilados\compiler>mix escript.build
enerated escript compiler_elixir with MIX_ENV=dev

:\Users\alfre\Documents\GitHub\c202-jubilados\compiler>escript compiler_elixir examples/valid/all_op.c
```

- **Release download**

In this section we will download a specific version of the jubilados project.Which is in the repository

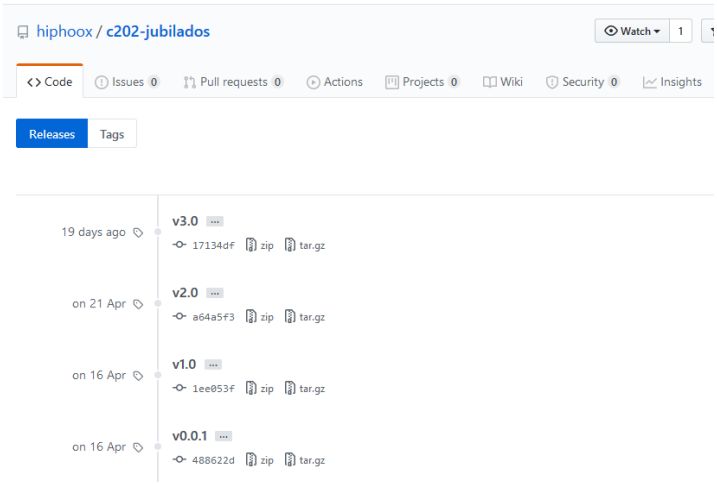
We enter the repository that is on github

https://github.com/hiphoox/c202-jubilados

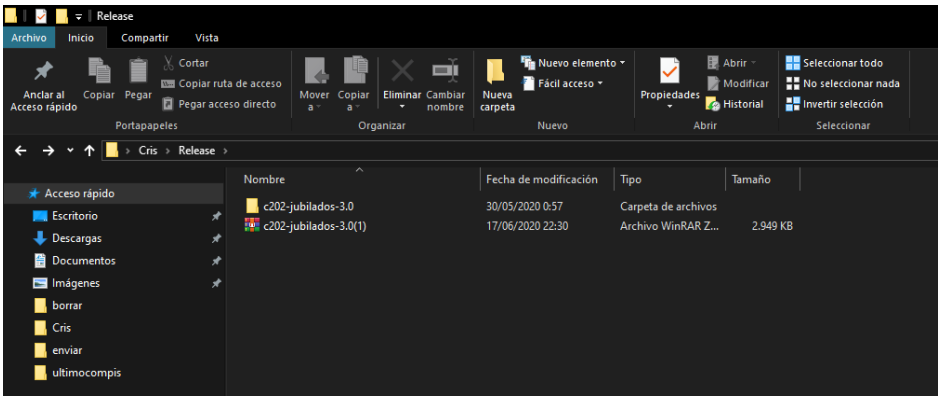


We select the releases section.

We download the version we are interested in.



Once the version is downloaded we place it in a folder and extract the file



Git command table

git init	Create empty Git repo in specified directory. Run with no arguments to initialize the current directory as a git repository.
git clone	Clone repo located at <repo> into local machine. Original repo can be located on the local filesystem or on a remote machine.
git add .	Stage all changes in <directory> for the next commit. Replace <directory> with a <file> to change a specific file.

git commit -m "Descriptive text"	Commit the staged snapshot, but instead of launching a text editor, use <message> as the commit message.
git push	Push the branch to <remote>, along with necessary commits and objects. Creates named branch in the remote repo if it doesn't exist.
git pull	Fetch the specified remote's copy of current branch and immediately merge it into the local copy.
git branch <branchName>	List all of the branches in your repo. Add a <branch> argument to create a new branch with the name <branch>.
git status	List which files are staged, unstaged, and untracked.

Other commands used

- mix escript.build

  Builds an escript for the project.

  An escript is an executable that can be invoked from the command line. An escript can run on any machine that has Erlang/OTP installed and by default does not require Elixir to be installed, as Elixir is embedded as part of the escript.

- ls command

  Type DIR to show the folders and files in command prompt

- Cd command

  The cd command, also known as chdir (change directory), is a command-line shell command used to change the current working directory in various operating systems