# Compiler

Jubilados
Second Stage

**Project Manager**
Ruiz Aguilar Eduardo

**System Architect**
Rodriguez Garcia Dulce Coral

**System Tester**
Hernández Escobar Oswaldo

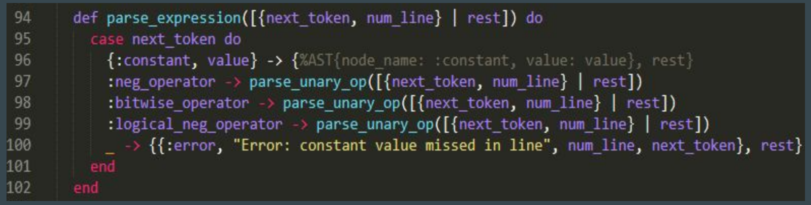
**System Integrator**
Aguilera Ortiz Alfredo

# Topics

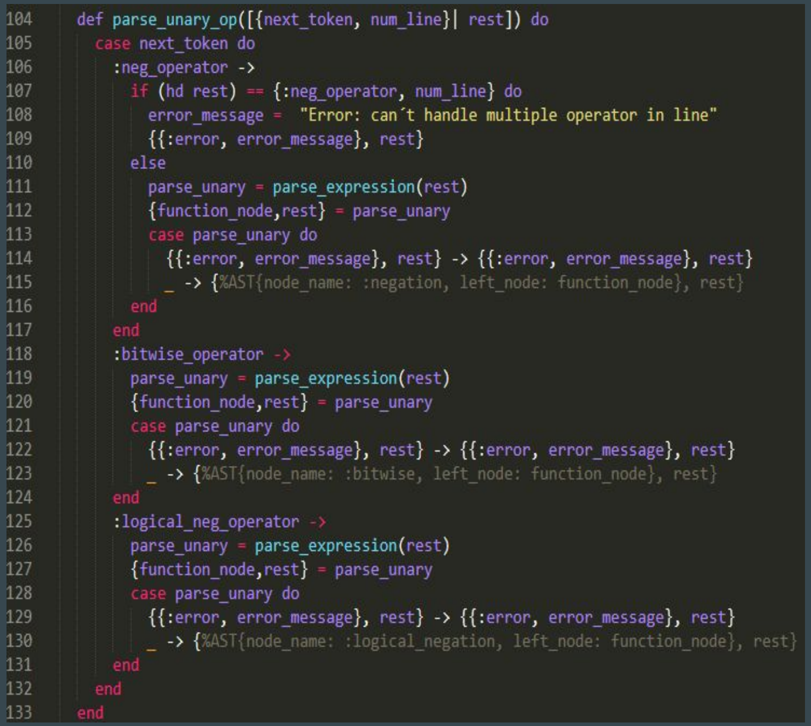
- Requirements
  - Unary operators
- Code Modifications
  - Code Generator
- Test
  - Before
  - After

# Modifications

## Parser

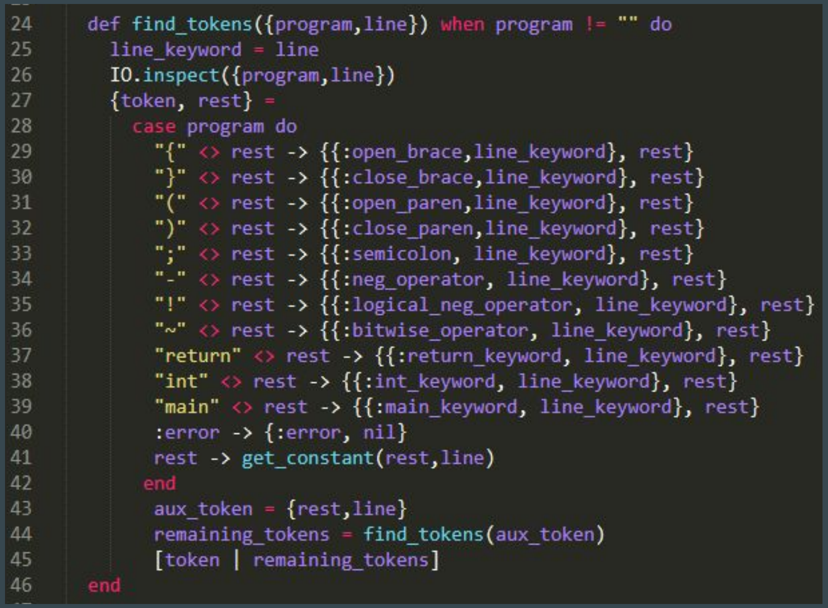


```
def parse_expression([{next_token, num_line} | rest]) do
  case next_token do
    {:constant, value} -> {%AST{node_name: :constant, value: value}, rest}
    :neg_operator -> parse_unary_op([{next_token, num_line} | rest])
    :bitwise_operator -> parse_unary_op([{next_token, num_line} | rest])
    :logical_neg_operator -> parse_unary_op([{next_token, num_line} | rest])
    _ -> {{:error, "Error: constant value missed in line", num_line, next_token}, rest}
  end
end
```



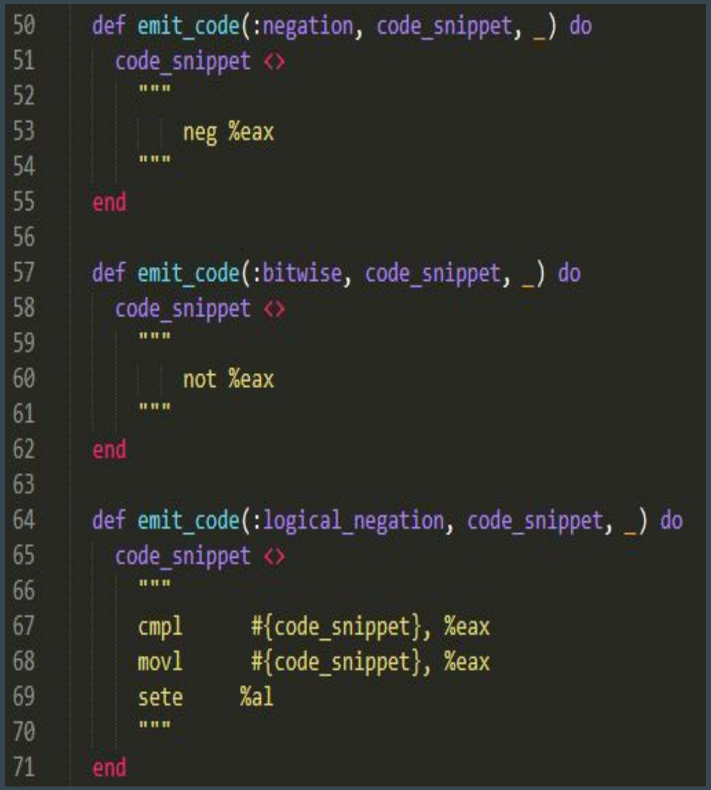
```
def parse_unary_op([{next_token, num_line}| rest]) do
  case next_token do
    :neg_operator ->
      if (hd rest) == {:neg_operator, num_line} do
        error_message =  "Error: can´t handle multiple operator in line"
        {{:error, error_message}, rest}
      else
        parse_unary = parse_expression(rest)
        {function_node,rest} = parse_unary
        case parse_unary do
          {{:error, error_message}, rest} -> {{:error, error_message}, rest}
          _ -> {%AST{node_name: :negation, left_node: function_node}, rest}
      end
    end
    :bitwise_operator ->
      parse_unary = parse_expression(rest)
      {function_node,rest} = parse_unary
      case parse_unary do
        {{:error, error_message}, rest} -> {{:error, error_message}, rest}
        _ -> {%AST{node_name: :bitwise, left_node: function_node}, rest}
    end
    :logical_neg_operator ->
      parse_unary = parse_expression(rest)
      {function_node,rest} = parse_unary
      case parse_unary do
        {{:error, error_message}, rest} -> {{:error, error_message}, rest}
        _ -> {%AST{node_name: :logical_negation, left_node: function_node}, rest}
    end
  end
end
```

## Lexer



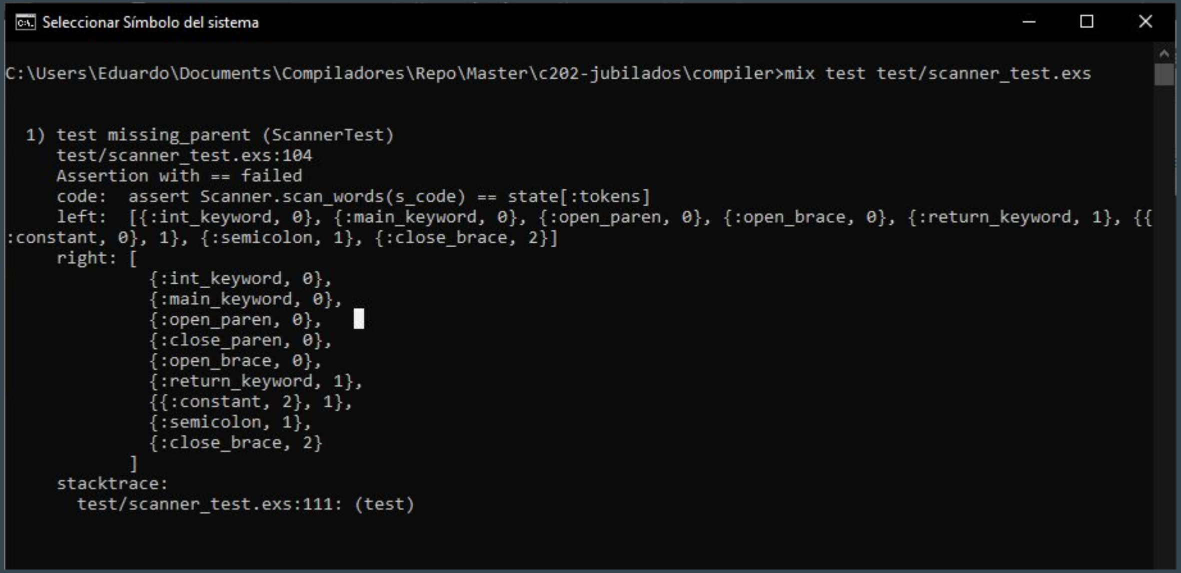
```
def find_tokens({program,line}) when program != "" do
  line_keyword = line
  IO.inspect({program,line})
  {token, rest} =
    case program do
      "{" <> rest -> {{:open_brace,line_keyword}, rest}
      "}" <> rest -> {{:close_brace,line_keyword}, rest}
      "(" <> rest -> {{:open_paren,line_keyword}, rest}
      ")" <> rest -> {{:close_paren,line_keyword}, rest}
      ";" <> rest -> {{:semicolon, line_keyword}, rest}
      "-" <> rest -> {{:neg_operator, line_keyword}, rest}
      "!" <> rest -> {{:logical_neg_operator, line_keyword}, rest}
      "~" <> rest -> {{:bitwise_operator, line_keyword}, rest}
      "return" <> rest -> {{:return_keyword, line_keyword}, rest}
      "int" <> rest -> {{:int_keyword, line_keyword}, rest}
      "main" <> rest -> {{:main_keyword, line_keyword}, rest}
      :error -> {:error, nil}
      rest -> get_constant(rest,line)
     end
      aux_token = {rest,line}
      remaining_tokens = find_tokens(aux_token)
      [token | remaining_tokens]
end
```

## Code Generator

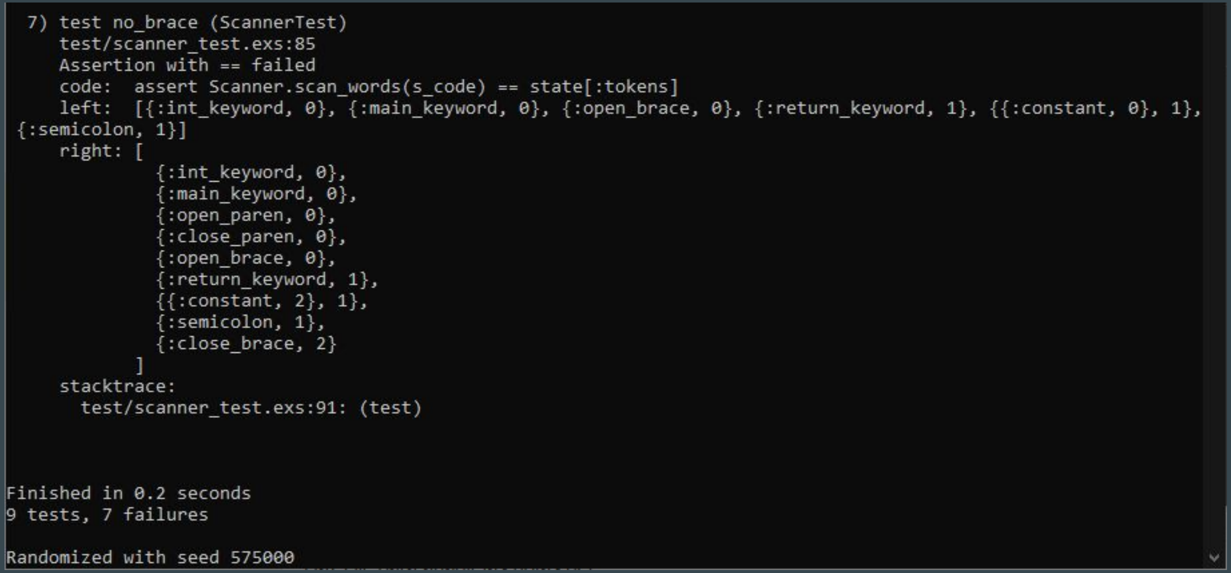


```
def emit_code(:negation, code_snippet, _) do
  code_snippet <>
    """

        neg %eax
    """
end

def emit_code(:bitwise, code_snippet, _) do
  code_snippet <>
    """

        not %eax
    """
end

def emit_code(:logical_negation, code_snippet, _) do
  code_snippet <>
    """

    cmpl     #{code_snippet}, %eax
    movl     #{code_snippet}, %eax
    sete     %al
    """
end
```

# Test

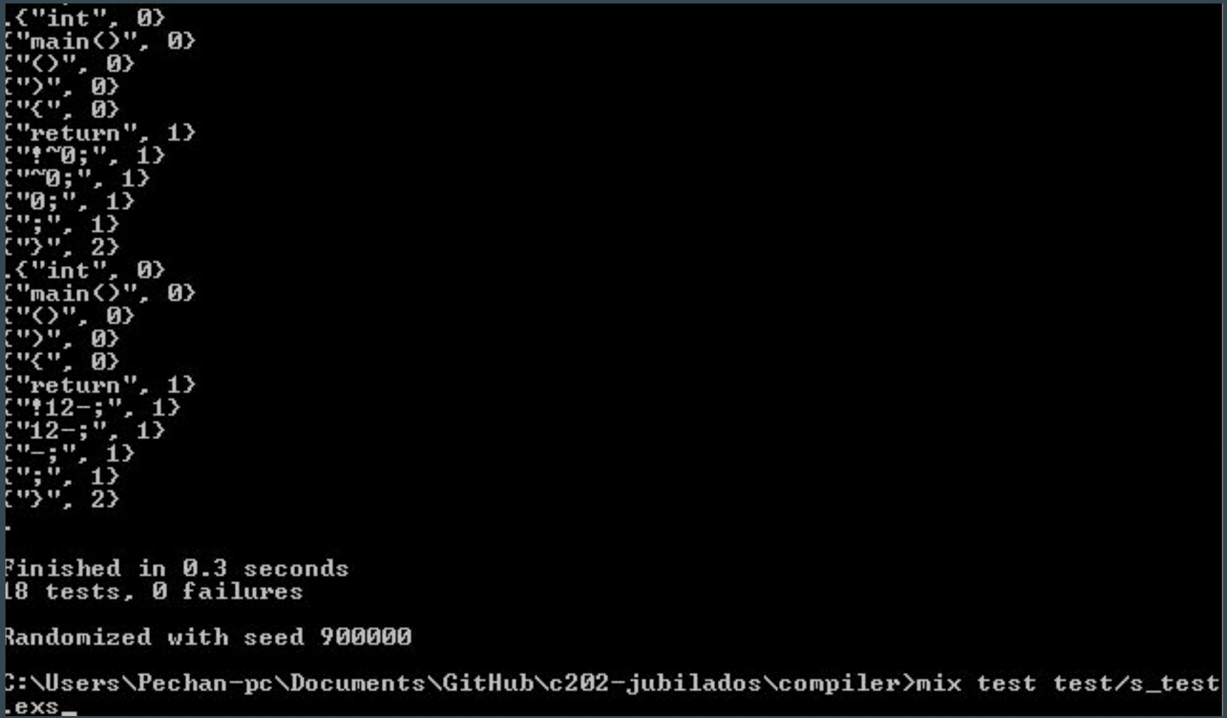


```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler>mix test test/scanner_test.exs


  1) test missing_parent (ScannerTest)
     test/scanner_test.exs:104
     Assertion with == failed
     code:  assert Scanner.scan_words(s_code) == state[:tokens]
     left:  [{:int_keyword, 0}, {:main_keyword, 0}, {:open_paren, 0}, {:open_brace, 0}, {:return_keyword, 1}, {{
:constant, 0}, 1}, {:semicolon, 1}, {:close_brace, 2}]
     right: [
              {:int_keyword, 0},
              {:main_keyword, 0},
              {:open_paren, 0},
              {:close_paren, 0},
              {:open_brace, 0},
              {:return_keyword, 1},
              {{:constant, 2}, 1},
              {:semicolon, 1},
              {:close_brace, 2}
            ]
     stacktrace:
       test/scanner_test.exs:111: (test)
```

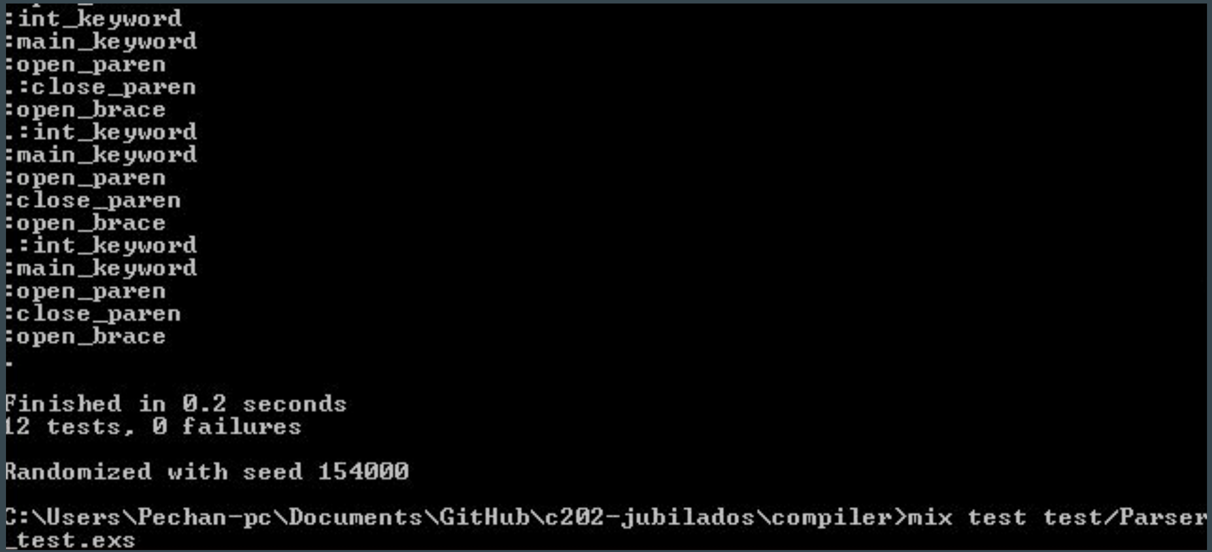


```
  7) test no_brace (ScannerTest)
     test/scanner_test.exs:85
     Assertion with == failed
     code:  assert Scanner.scan_words(s_code) == state[:tokens]
     left:  [{:int_keyword, 0}, {:main_keyword, 0}, {:open_brace, 0}, {:return_keyword, 1}, {{:constant, 0}, 1},
 {:semicolon, 1}]
     right: [
              {:int_keyword, 0},
              {:main_keyword, 0},
              {:open_paren, 0},
              {:close_paren, 0},
              {:open_brace, 0},
              {:return_keyword, 1},
              {{:constant, 2}, 1},
              {:semicolon, 1},
              {:close_brace, 2}
            ]
     stacktrace:
       test/scanner_test.exs:91: (test)



Finished in 0.2 seconds
9 tests, 7 failures

Randomized with seed 575000
```

# Test

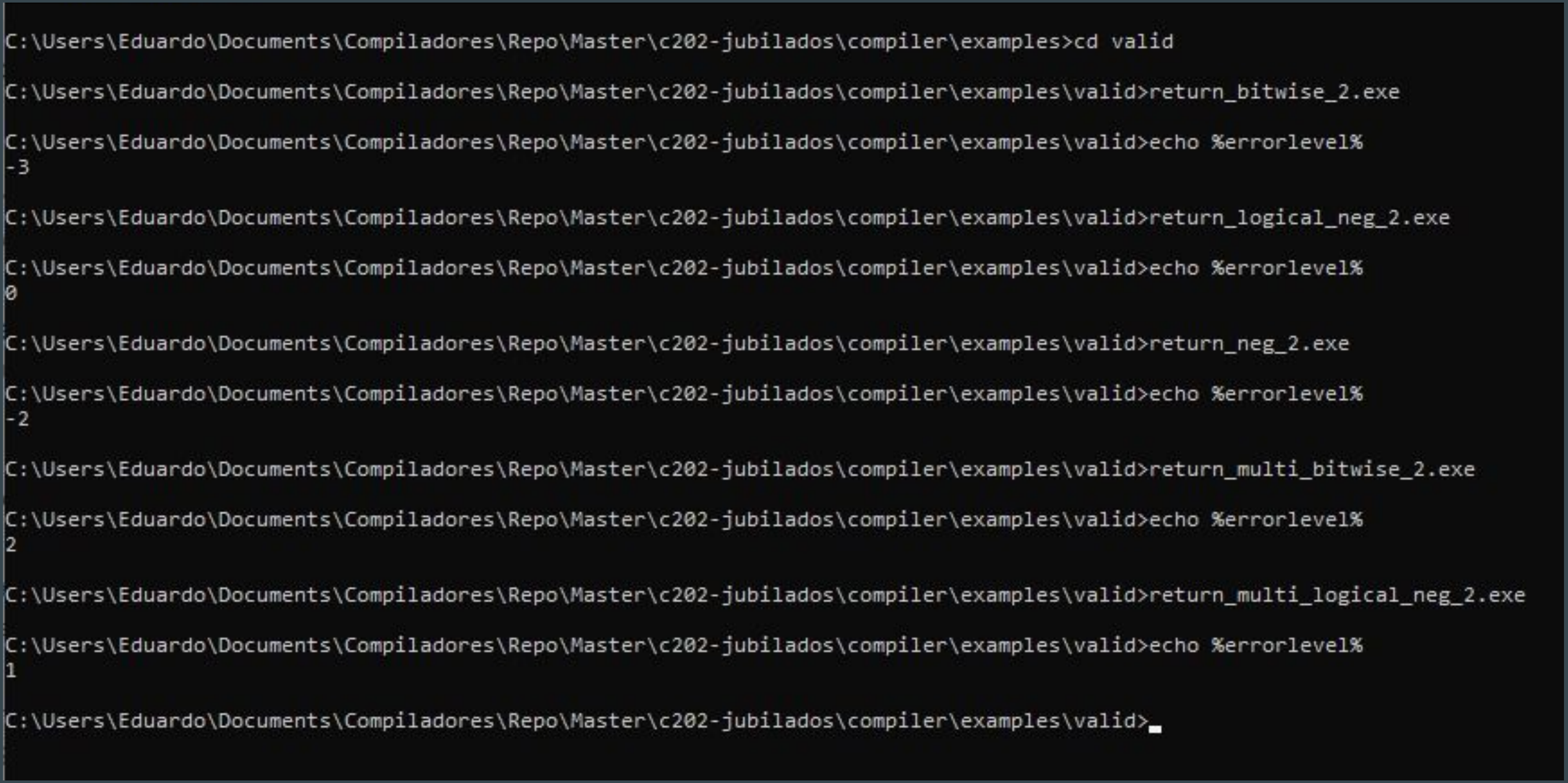


```
.{"int", 0}
{"main()", 0}
{"()", 0}
{")", 0}
{"{", 0}
{"return", 1}
{"!~0;", 1}
{"~0;", 1}
{"0;", 1}
{";", 1}
{"}", 2}
.{"int", 0}
{"main()", 0}
{"()", 0}
{")", 0}
{"{", 0}
{"return", 1}
{"!12-;", 1}
{"12-;", 1}
{"-;", 1}
{";", 1}
{"}", 2}
.

Finished in 0.3 seconds
18 tests, 0 failures

Randomized with seed 900000

C:\Users\Pechan-pc\Documents\GitHub\c202-jubilados\compiler>mix test test/s_test
.exs_
```



```
:int_keyword
:main_keyword
:open_paren
.:close_paren
:open_brace
.:int_keyword
:main_keyword
:open_paren
:close_paren
:open_brace
.:int_keyword
:main_keyword
:open_paren
:close_paren
:open_brace
.

Finished in 0.2 seconds
12 tests, 0 failures

Randomized with seed 154000

C:\Users\Pechan-pc\Documents\GitHub\c202-jubilados\compiler>mix test test/Parser
_test.exs
```

# Test

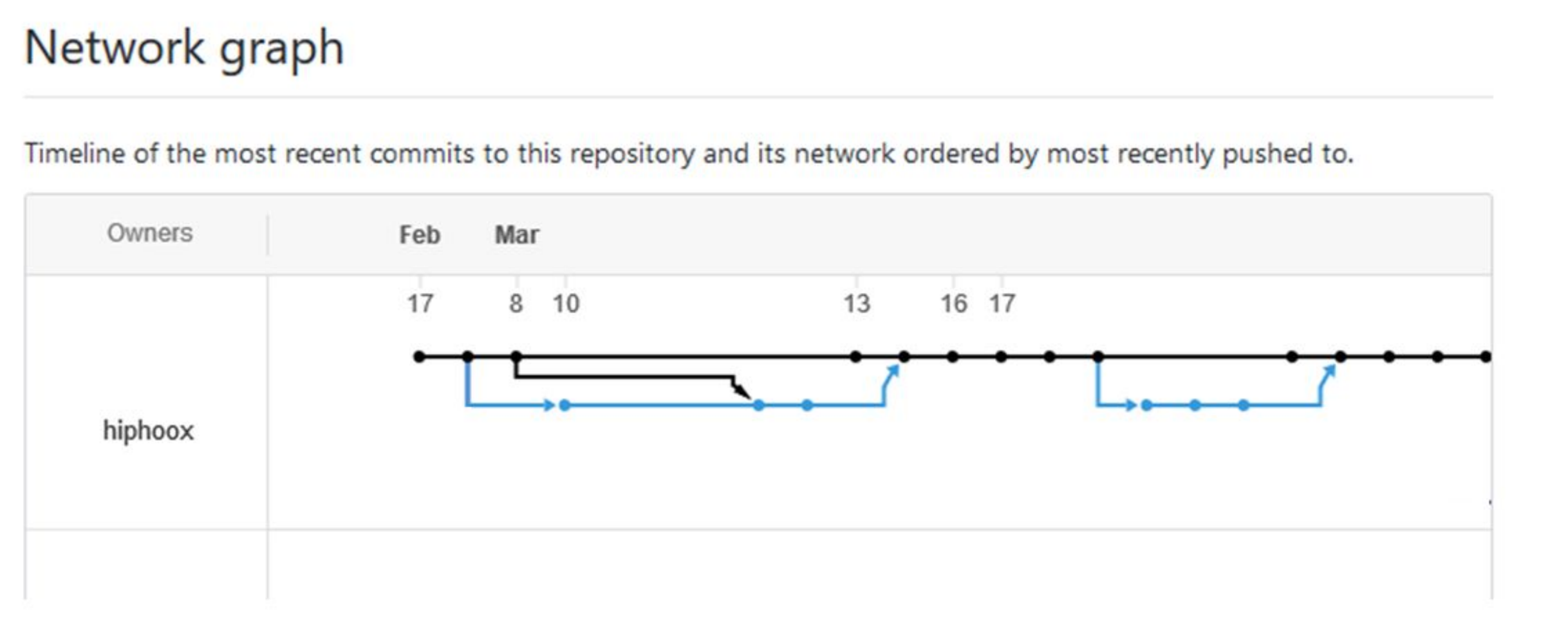


```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples>cd valid

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>return_bitwise_2.exe

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>echo %errorlevel%
-3

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>return_logical_neg_2.exe

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>echo %errorlevel%
0

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>return_neg_2.exe

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>echo %errorlevel%
-2

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>return_multi_bitwise_2.exe

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>echo %errorlevel%
2

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>return_multi_logical_neg_2.exe

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>echo %errorlevel%
1

C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados\compiler\examples\valid>
```

# Repo Modifications

- Improvements
  - Branches
    - Develop
    - Members branch
  - Version

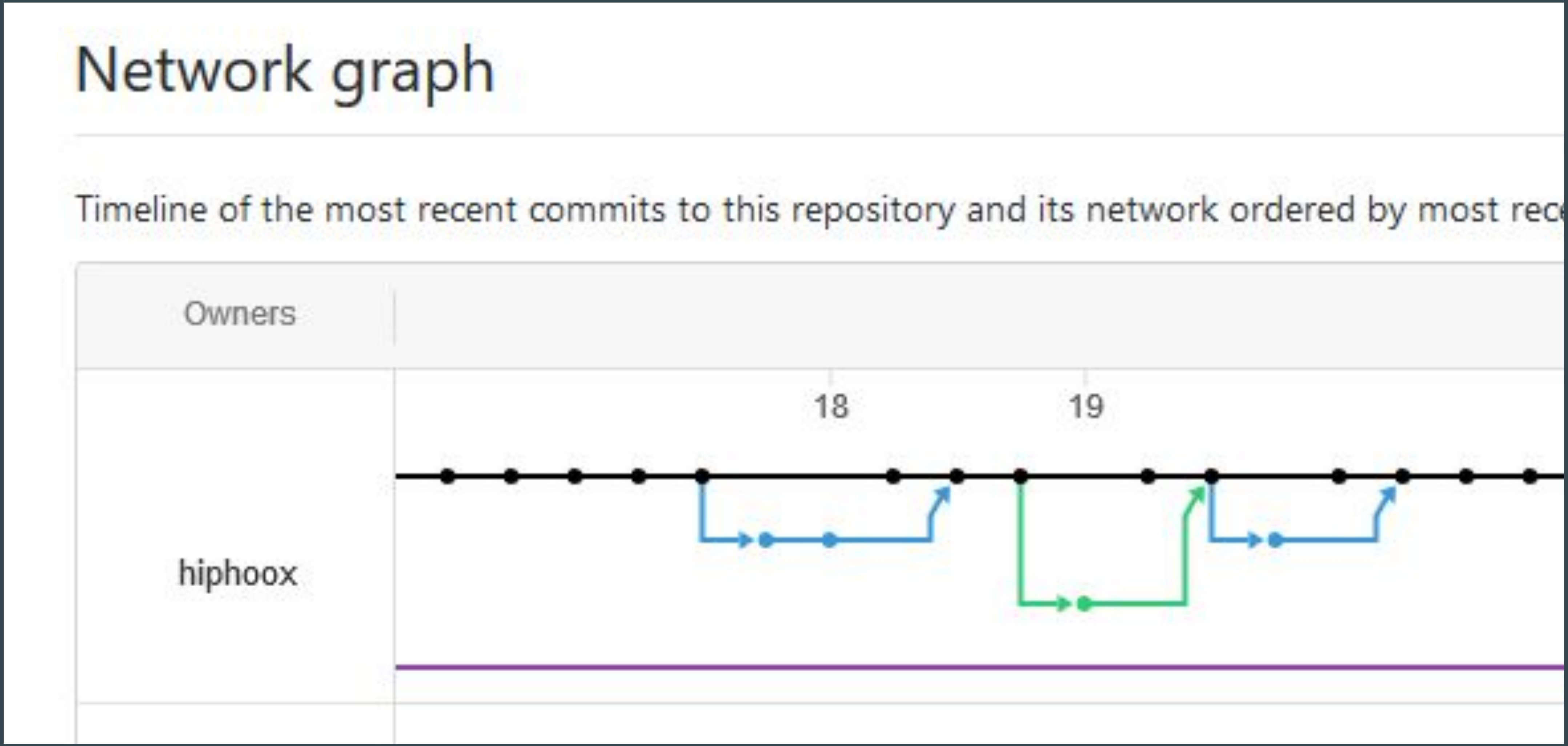
# First Project



Network graph

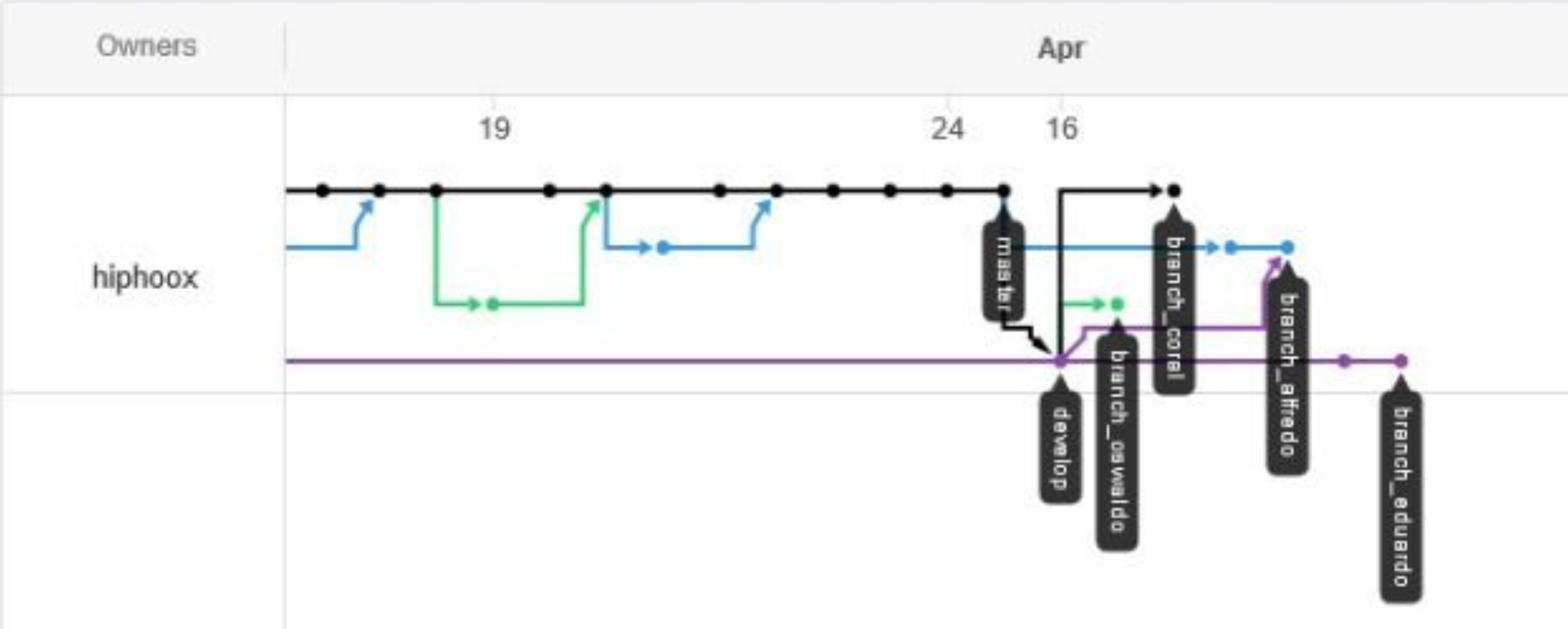
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

# Mistakes

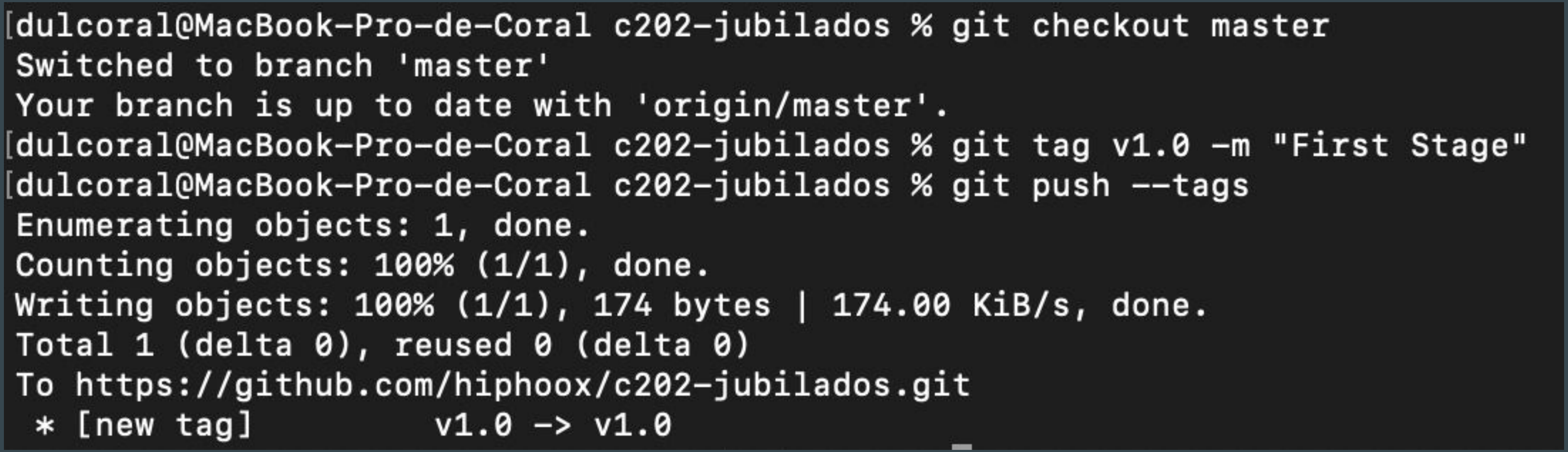


Network graph

Timeline of the most recent commits to this repository and its network ordered by most rec

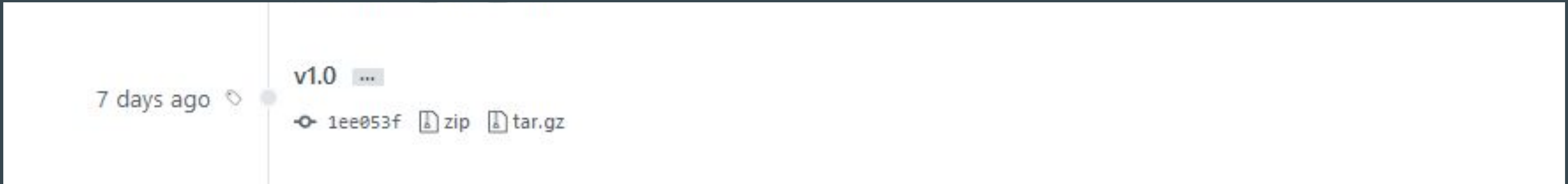

Owners
Apr
19
24
16
hiphoox
master
branch_coral
develop
branch_oswaldo
branch_alfredo
branch_eduardo

# Releases



```
[dulcoral@MacBook-Pro-de-Coral c202-jubilados % git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.
[dulcoral@MacBook-Pro-de-Coral c202-jubilados % git tag v1.0 -m "First Stage"
[dulcoral@MacBook-Pro-de-Coral c202-jubilados % git push --tags
Enumerating objects: 1, done.
Counting objects: 100% (1/1), done.
Writing objects: 100% (1/1), 174 bytes | 174.00 KiB/s, done.
Total 1 (delta 0), reused 0 (delta 0)
To https://github.com/hiphoox/c202-jubilados.git
 * [new tag]         v1.0 -> v1.0
```

# First
# Version

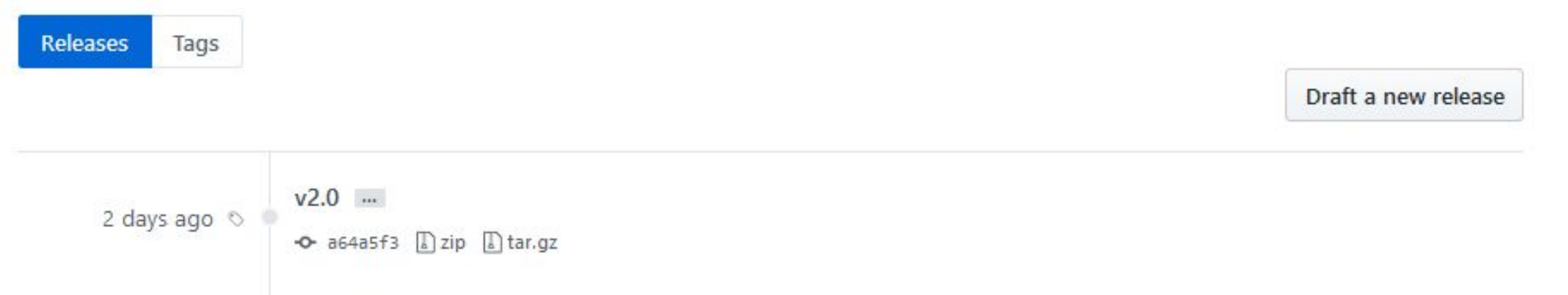


7 days ago

v1.0 ...

1ee053f zip tar.gz

# Second Version

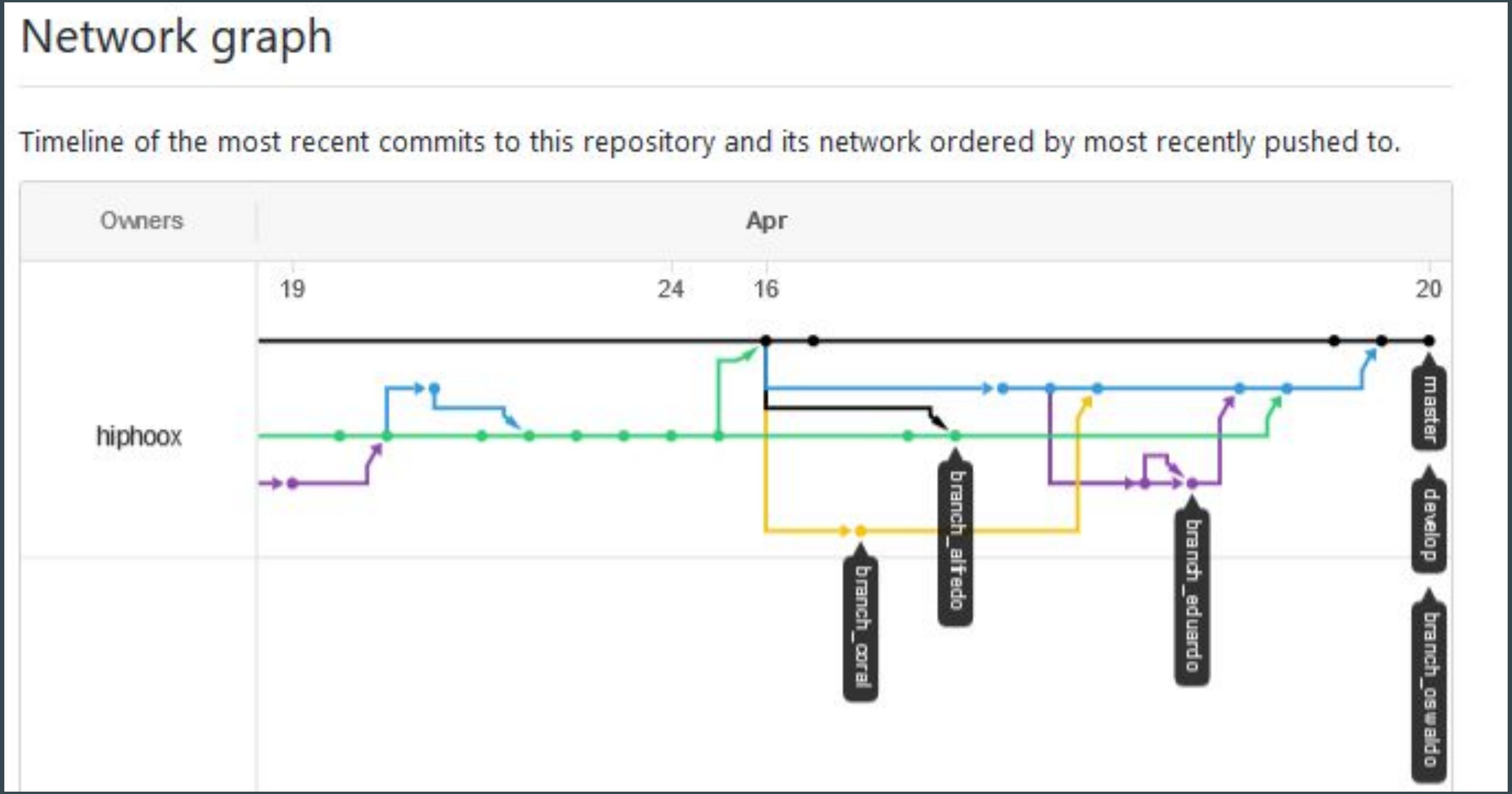


2 days ago

v2.0 ...

a64a5f3 zip tar.gz

# Final Result



Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.

# Conclusions