

Compiler

Jubilados

Fourth Stage

Project Manager

Ruiz Aguilar Eduardo

System Architect

Rodriguez Garcia Dulce Coral

System Tester

Hernández Escobar Oswaldo

System Integrator

Aguilera Ortiz Alfredo

Topics

- Code modifications.
 - Scanner.
 - Parser :
 - Grammar.
 - Code Generator.
 - Utils Module.
- Test
- New workflow :
 - Agile methodology.
 - Scrum.
 - Kanban.
- Organization :
 - Jira Software.



Modifications

Lexer

```
23
24 def find_tokens({program,line}) when program != "" do
25   line_keyword = line
26   IO.inspect({program,line})
27   {token, rest} =
28     case program do
29       "{" <> rest -> {:open_brace, line_keyword}, rest}
30       "}" <> rest -> {:close_brace, line_keyword}, rest}
31       "(" <> rest -> {:open_paren, line_keyword}, rest}
32       ")" <> rest -> {:close_paren, line_keyword}, rest}
33       ";" <> rest -> {:semicolon, line_keyword}, rest}
34       "-" <> rest -> {:neg_operator, line_keyword}, rest}
35       "&&" <> rest -> {:and_operator, line_keyword}, rest}
36       "||" <> rest -> {:or_operator, line_keyword}, rest}
37       "<=" <> rest -> {:less_than_or_equal_operator, line_keyword}, rest}
38       "<" <> rest -> {:less_than_operator, line_keyword}, rest}
39       ">=" <> rest -> {:greater_than_or_equal_operator, line_keyword}, rest}
40       ">" <> rest -> {:greater_than_operator, line_keyword}, rest}
41       "==" <> rest -> {:equal_operator, line_keyword}, rest}
42       "!=" <> rest -> {:not_equal_operator, line_keyword}, rest}
43       "!" <> rest -> {:logical_neg_operator, line_keyword}, rest}
44       "~" <> rest -> {:bitwise_operator, line_keyword}, rest}
45       "+" <> rest -> {:add_operator, line_keyword}, rest}
46       "*" <> rest -> {:mult_operator, line_keyword}, rest}
47       "/" <> rest -> {:div_operator, line_keyword}, rest}
48       "return" <> rest -> {:return_keyword, line_keyword}, rest}
49       "int" <> rest -> {:int_keyword, line_keyword}, rest}
50       "main" <> rest -> {:main_keyword, line_keyword}, rest}
51       :error -> {:error, nil}
52       rest -> get_constant(rest,line)
53     end
54     aux_token = {rest,line}
55     remaining_tokens = find_tokens(aux_token)
56     [token | remaining_tokens]
57   end
58 end
```

Code Generator

```
103 def emit_code(:equal, code_snippet, _) do
104   code_snippet <>
105     """
106     push %eax
107     pop %ecx
108     cmpl %eax, %ecx
109     movl #{code_snippet}, %eax
110     sete %al
111     """
112   end
113
114   def emit_code(:not_equal, code_snippet, _) do
115     code_snippet <>
116       """
117       push %eax
118       pop %ecx
119       cmpl %eax, %ecx
120       movl #{code_snippet}, %eax
121       setne %al
122       """
123     end
124
125     def emit_code(:less_than, code_snippet, _) do
126       code_snippet <>
127         """
128         push %eax
129         pop %ecx
130         cmpl %eax, %ecx
131         movl #{code_snippet}, %eax
132         setl %al
133         """
134       end
135
136       def emit_code(:less_than_or_equal, code_snippet, _) do
137         code_snippet <>
138           """
139           push %eax
140           pop %ecx
141           cmpl %eax, %ecx
142           movl #{code_snippet}, %eax
143           setle %al
144           """
145         end
```

```
147   def emit_code(:greater_than, code_snippet, _) do
148     code_snippet <>
149       """
150       push %eax
151       pop %ecx
152       cmpl %eax, %ecx
153       movl #{code_snippet}, %eax
154       setg %al
155       """
156     end
157
158     def emit_code(:greater_than_or_equal, code_snippet, _) do
159       code_snippet <>
160         """
161         push %eax
162         pop %ecx
163         cmpl %eax, %ecx
164         movl #{code_snippet}, %eax
165         setge %al
166         """
167       end
168
169       def emit_code(:or_op, code_snippet, _) do
170         code_snippet <>
171           """
172           cmpl $0, %eax
173           je _clause2
174           movl $1, %eax
175           jmp _end
176
177           _clause2:
178           cmpl $0, %eax
179           movl $0, %eax
180           setne %al
181
182           _end:
183         """
184       end
185
186       def emit_code(:and_op, code_snippet, _) do
187         code_snippet <>
188           """
189           cmpl $0, %eax
190           jne _clause2
191           jmp _end
192
193           _clause2:
194           cmpl $0, %eax
195           movl $0, %eax
196           setne %al
197
198           _end:
199         """
200       end
```


Parser

```
94 def parse_expression([next_token, num_line] | rest) do
95   logical_and_expression = parse_logical_and_expression([next_token, num_line] | rest)
96   {expression_node, logical_and_expression_rest} = logical_and_expression
97   [{next_token, num_line} | rest] = logical_and_expression_rest
98   case next_token do
99     :or_operator ->
100     subTree = %AST{node_name: :or_op}
101     parse_op = parse_logical_and_expression(rest)
102     {node, parse_rest} = parse_op
103     [{next_token, num_line} | rest_op] = parse_rest
104     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
105   ->
106   logical_and_expression
107 end
108 end
109
110 def parse_logical_and_expression([next_token, num_line] | rest) do
111   equality_expression = parse_equality_expression([next_token, num_line] | rest)
112   {expression_node, equality_expression_rest} = equality_expression
113   [{next_token, num_line} | rest] = equality_expression_rest
114   case next_token do
115     :and_operator ->
116     subTree = %AST{node_name: :and_op}
117     parse_op = parse_equality_expression(rest)
118     {node, parse_rest} = parse_op
119     [{next_token, num_line} | rest_op] = parse_rest
120     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
121   ->
122   equality_expression
123 end
124 end
125
126 def parse_equality_expression([next_token, num_line] | rest) do
127   relational_expression = parse_relational_expression([next_token, num_line] | rest)
128   {expression_node, relational_expression_rest} = relational_expression
129   [{next_token, num_line} | rest] = relational_expression_rest
130   case next_token do
131     :not_equal_operator ->
132     subTree = %AST{node_name: :not_equal}
133     parse_op = parse_relational_expression(rest)
134     {node, parse_rest} = parse_op
135     [{next_token, num_line} | rest_op] = parse_rest
136     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
137   :equal_operator ->
138     subTree = %AST{node_name: :equal}
139     parse_op = parse_relational_expression(rest)
140     {node, parse_rest} = parse_op
141     [{next_token, num_line} | rest_op] = parse_rest
142     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
143   ->
144   relational_expression
145 end
146 end
147
```

```
148 def parse_relational_expression([next_token, num_line] | rest) do
149   additive_expression = parse_additive_expression([next_token, num_line] | rest)
150   {expression_node, additive_expression_rest} = additive_expression
151   [{next_token, num_line} | rest] = additive_expression_rest
152   case next_token do
153     :less_than_operator ->
154     subTree = %AST{node_name: :less_than}
155     parse_op = parse_additive_expression(rest)
156     {node, parse_rest} = parse_op
157     [{next_token, num_line} | rest_op] = parse_rest
158     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
159   :greater_than_operator ->
160     subTree = %AST{node_name: :greater_than}
161     parse_op = parse_additive_expression(rest)
162     {node, parse_rest} = parse_op
163     [{next_token, num_line} | rest_op] = parse_rest
164     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
165   :less_than_or_equal_operator ->
166     subTree = %AST{node_name: :less_than_or_equal}
167     parse_op = parse_additive_expression(rest)
168     {node, parse_rest} = parse_op
169     [{next_token, num_line} | rest_op] = parse_rest
170     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
171   :greater_than_or_equal_operator ->
172     subTree = %AST{node_name: :greater_than_or_equal}
173     parse_op = parse_additive_expression(rest)
174     {node, parse_rest} = parse_op
175     [{next_token, num_line} | rest_op] = parse_rest
176     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
177   ->
178   additive_expression
179 end
180 end
181
182 def parse_additive_expression([next_token, num_line] | rest) do
183   term = parse_term([next_token, num_line] | rest)
184   {expression_node, term_rest} = term
185   [{next_token, num_line} | rest] = term_rest
186   case next_token do
187     :add_operator ->
188     subTree = %AST{node_name: :addition}
189     parse_op = parse_term(rest)
190     {node, parse_rest} = parse_op
191     [{next_token, num_line} | rest_op] = parse_rest
192     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
193   :neg_operator ->
194     subTree = %AST{node_name: :subtraction}
195     parse_op = parse_term(rest)
196     {node, parse_rest} = parse_op
197     [{next_token, num_line} | rest_op] = parse_rest
198     {%{subTree | left_node: expression_node, right_node: node}, parse_rest}
199   ->
200   term
201 end
202 end
203
```

Utils Module

```
defmodule Utils do
  @moduledoc """
  Documentation for `Utils`.
  """

  def clean_code(file_content) do
    divide_code = String.trim(file_content)
    clean_words = Regex.split(~r/\n/, divide_code)
    counter = 0
    get_tokens(clean_words, counter)
  end

  def get_tokens(program, line) when program != [] do
    lines = hd program
    extra_lines = tl program
    tokens = Regex.split(~r/\s+/, lines, trim: true)
    tokens_in_line = Enum.flat_map(tokens, fn token -> [{token, line}] end)
    line = line + 1
    others = get_tokens(extra_lines, line)
    tokens_in_line ++ others
  end

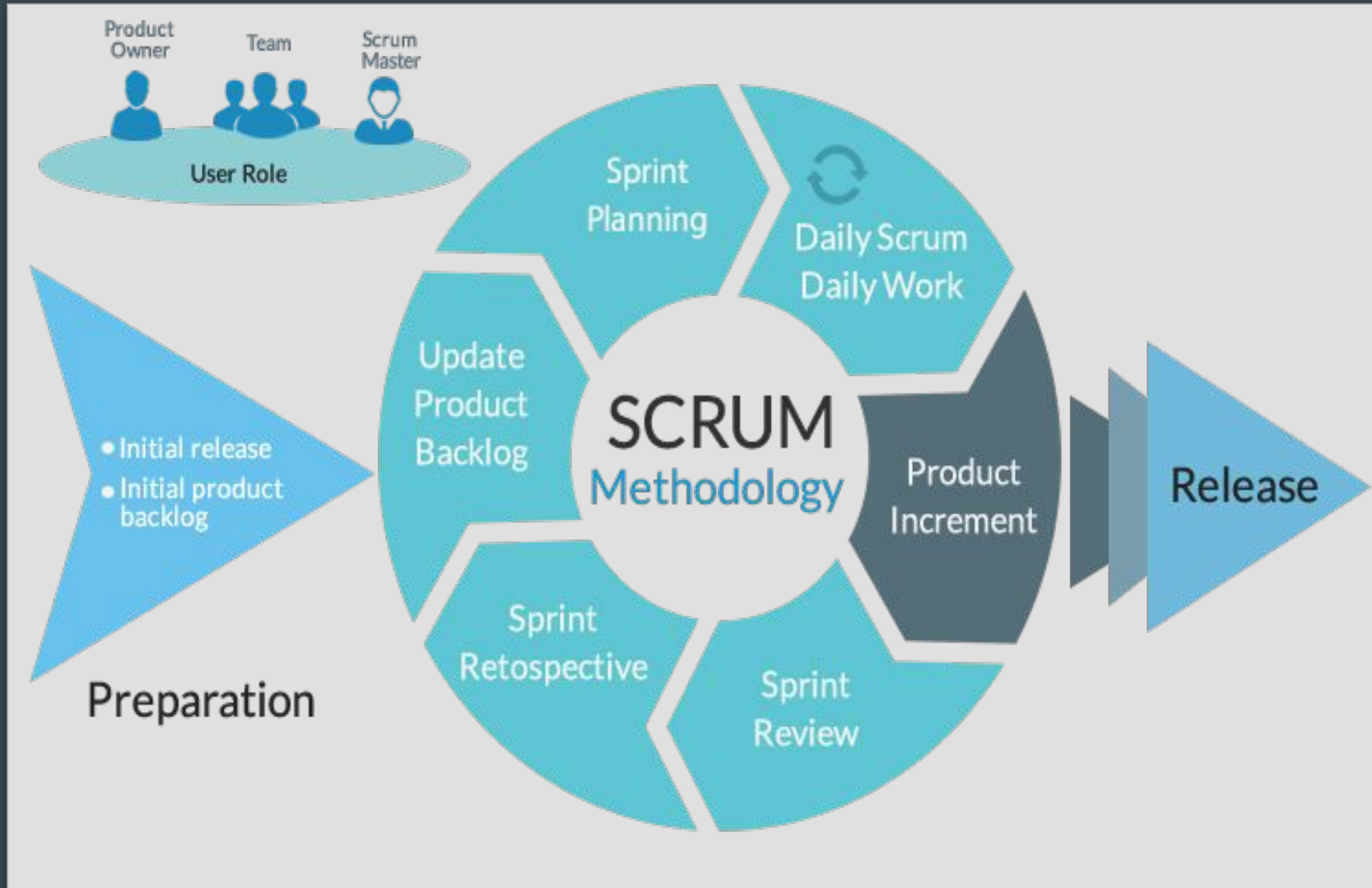
  def get_tokens(_program, _linea) do
    []
  end
end
```

Workflow

- ¿What is a methodology?
- ¿What is an agile methodology?
- ¿What are frameworks?



Scrum



Kanban



Test

```
{"return", 1}
{"2", 1}
{"*", 1}
{"3;", 1}
{";", 1}
{"", 2}
{"int", 0}
{"main()", 0}
{"<", 0}
{"", 0}
{"{", 0}
{"return", 1}
{"0", 1}
{"<=", 1}
{"2;", 1}
{";", 1}
{"", 2}
.
```

Finished in 1.7 seconds
59 tests, 0 failures

Randomized with seed 574000

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

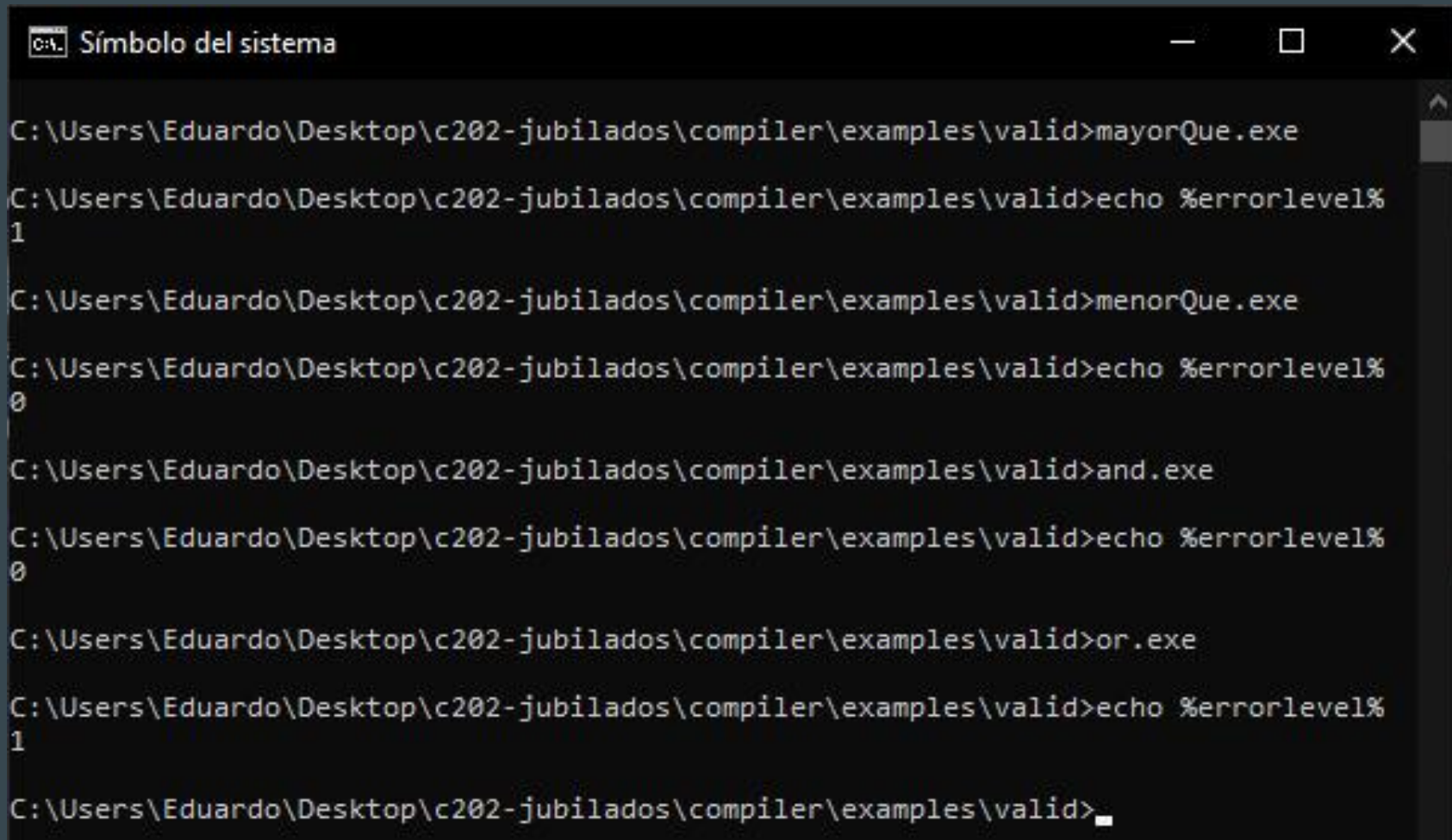
```
:open_brace
.:int_keyword
:main_keyword
:open_paren
:close_paren
:open_brace
.:int_keyword
:main_keyword
:open_paren
:close_paren
:open_brace
.:int_keyword
:main_keyword
:open_paren
:close_paren
:open_brace
.
```

Finished in 1.1 seconds
48 tests, 0 failures

Randomized with seed 83000

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

Test



```
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>mayorQue.exe  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>echo %errorlevel%  
1  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>menorQue.exe  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>echo %errorlevel%  
0  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>and.exe  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>echo %errorlevel%  
0  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>or.exe  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>echo %errorlevel%  
1  
C:\Users\Eduardo\Desktop\c202-jubilados\compiler\examples\valid>_
```

GitHub


```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git checkout branch_eduardo
Switched to branch 'branch_eduardo'
Your branch is up to date with 'origin/branch_eduardo'.
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git status
On branch branch_eduardo
Your branch is up to date with 'origin/branch_eduardo'.

nothing to commit, working tree clean
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git merge origin develop
git: 'merge' is not a git command. See 'git --help'.
```

The most similar command is
merge

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git merge origin develop
Updating 17134df..3e0a41a
Fast-forward
 compiler/lib/code_generator.ex | 69 ++++++
 compiler/lib/parser.ex         | 22 +++++
 compiler/lib/scanner.ex        | 30 +++++
 3 files changed, 89 insertions(+), 32 deletions(-)
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git status
On branch branch_eduardo
Your branch is ahead of 'origin/branch_eduardo' by 6 commits.
  (use "git push" to publish your local commits)
```

nothing to commit, working tree clean

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git status
On branch branch_eduardo
Your branch is ahead of 'origin/branch_eduardo' by 6 commits.
  (use "git push" to publish your local commits)
```

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   compiler/lib/parser.ex
```

no changes added to commit (use "git add" and/or "git commit -a")

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git commit -m "Se agregan las funciones para
las expresiones AND, EQUALITY, RELATIONAL y ADDITIVE de la gramática"
[branch_eduardo c301203] Se agregan las funciones para las expresiones AND, EQUALITY, RELATIONAL y ADDITIVE de
la gramática
1 file changed, 95 insertions(+), 1 deletion(-)
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git push origin branch_eduardo
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 864 bytes | 216.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/hiphoox/c202-jubilados
 17134df..c301203 branch_eduardo -> branch_eduardo
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git status
On branch branch_eduardo
Your branch is up to date with 'origin/branch_eduardo'.
```

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   compiler/lib/code_generator.ex
```

no changes added to commit (use "git add" and/or "git commit -a")

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git add compiler/lib/code_generator.ex
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git commit -m "Se agregan las funciones para
tokens mayor que, menor que, mayor o igual que y menor o igual que del generador de código"
[branch_eduardo 5d454c7] Se agregan las funciones para tokens mayor que, menor que, mayor o igual que y menor o
igual que del generador de código
1 file changed, 44 insertions(+)
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git push origin branch_eduardo
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 560 bytes | 140.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0)
```

```
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/hiphoox/c202-jubilados
 c301203..5d454c7 branch_eduardo -> branch_eduardo
```

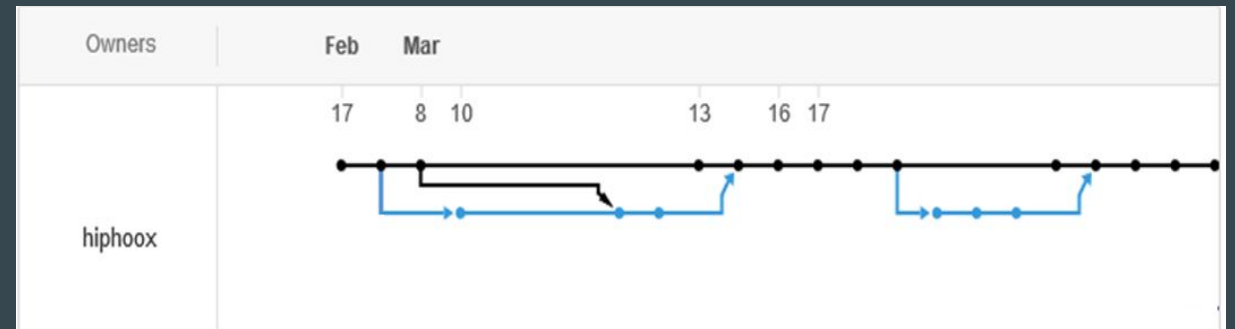
```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git checkout develop
Switched to branch 'develop'
Your branch is up to date with 'origin/develop'.
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git merge branch_eduardo
Updating 3e0a41a..5d454c7
Fast-forward
 compiler/lib/code_generator.ex | 44 ++++++
 compiler/lib/parser.ex         | 96 ++++++
 2 files changed, 139 insertions(+), 1 deletion(-)
```

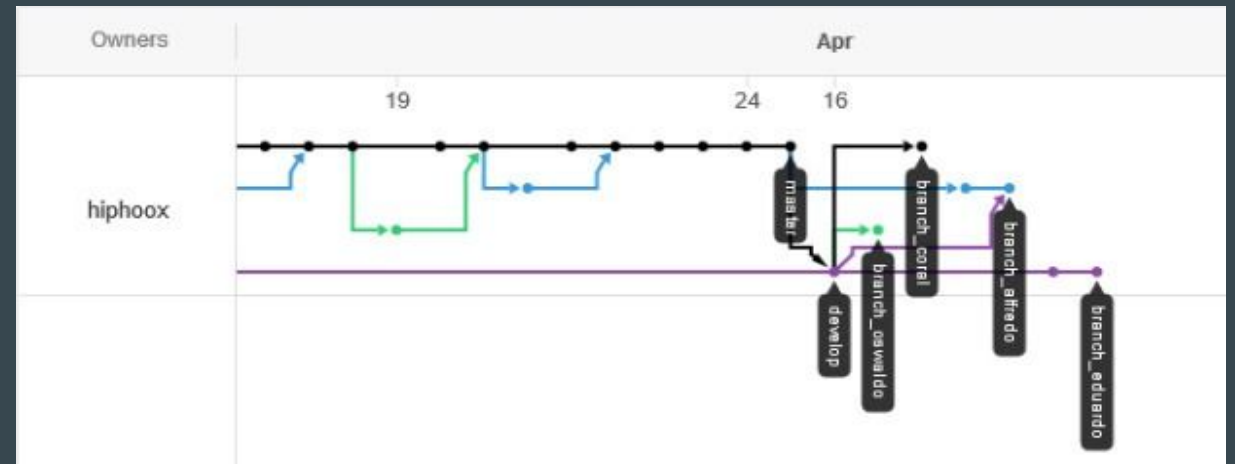
```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>git push origin develop
Total 0 (delta 0), reused 0 (delta 0)
To https://github.com/hiphoox/c202-jubilados
 3e0a41a..5d454c7 develop -> develop
```

```
C:\Users\Eduardo\Documents\Compiladores\Repo\Master\c202-jubilados>_
```

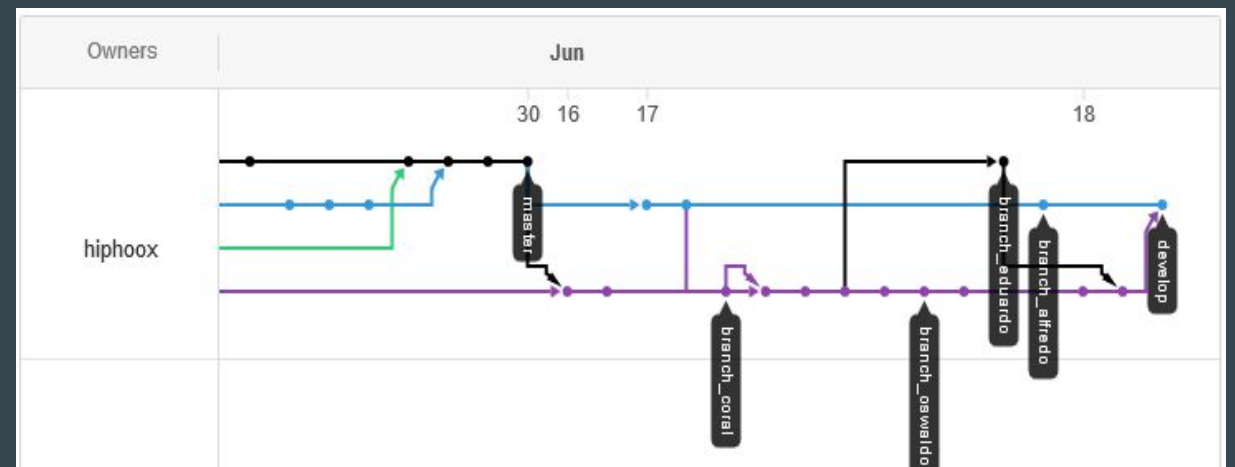
First Stage



Second Stage



Third Stage



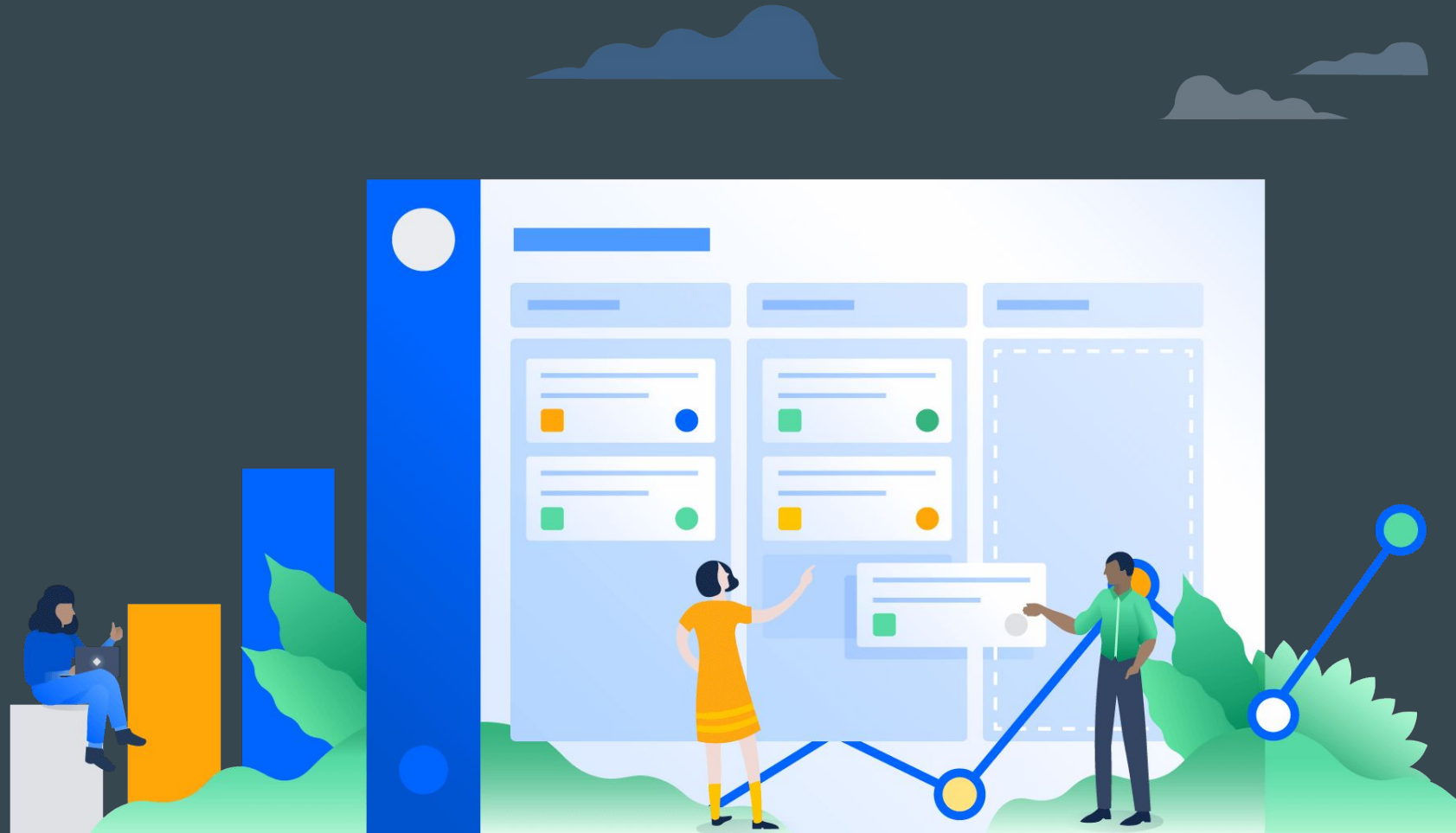
Final Release

The screenshot shows the GitHub interface for the repository `hiphoox / c202-jubilados`. The repository has 1 watch, 1 star, and 1 fork. The navigation bar includes links for Code, Issues (0), Pull requests (0), Actions, Projects (0), Wiki, Security (0), and Insights. The 'Releases' tab is selected, and a 'Draft a new release' button is visible. The releases section displays a vertical timeline of four releases:

Release Name	Commit Hash	Assets	Timestamp
v3.0	17134df	zip, tar.gz	19 days ago
v2.0	a64a5f3	zip, tar.gz	on 21 Apr
v1.0	1ee053f	zip, tar.gz	on 16 Apr
v0.0.1	488622d	zip, tar.gz	on 16 Apr

Jira Software

Development tool used
by agile teams





Compiladores

Proyecto de software clásico



Tablero COM

Tablero ▾



Backlog



Sprints activos



Informes



Lanzamientos



Incidencias y filtros



Páginas



Componentes



c202-jubilados



Añadir elemento



Configuración de p...

Proyectos / Compiladores / Tablero COM

Entrega final 4



1 día restante

Terminar sprint



Solo Mis Incidencias

Recientemente Actualizadas

POR HACER

Generar una clase Utils



5

COM-1



Realizar documentación trabajo escrito



5

COM-11



Hacer presentacion



3

COM-13



Realizar manual de usuario



3

COM-12



Modificar el generador de código 2



3

COM-15



EN CURSO

Modificar el parser 1



3

COM-7



Modificar el generador de código 1



3

COM-8



Test para el scanner



3

COM-9



Modificar el parser 2



3

COM-14



LISTO



Compiladores
Proyecto de software clásico



Tablero COM
Tablero ▾



Backlog



Sprints activos



Informes



Lanzamientos



Incidencias y filtros



Páginas



Componentes



c202-jubilados



Añadir elemento



Configuración de p...

Proyectos / Compiladores / Tablero COM

Entrega final 4



1 día restante

Terminar sprint



Solo Mis Incidencias

Recientemente Actualizadas

Borrar todo

POR HACER

Realizar manual de usuario



COM-12 AA

EN CURSO

Modificar el generador de código 1



COM-8 AA

LISTO

Proyectos / Compiladores / Tablero COM

Entrega final 4

☆ ⌚ 0 días restantes Terminar sprint

Q OE AA CR ER Solo Mis Incidencias Recientemente Actualizadas

POR HACER

EN CURSO

LISTO

Generar una clase Utils

🟢 ↑ 5

COM-1 CR

Modificar el parser 1

🟢 ↑ 3

COM-7 CR

Modificar el generador de código 2

🟢 ↑ 3

COM-15 ER

Modificar scanner 2

🟢 ↑ -

COM-19 CR

Test para el scanner

🟢 ↑ 3

COM-9 OE

Modificar el parser 2

🟢 ↑ 3


COM-14 ER

Crear incidencia

Importar incidencias

 Configurar campos ▾

Proyecto*

 Compiladores (COM) ▾

Tipo de Incidencia*

 Historia ▾



Algunos tipos de incidencia no están disponibles debido a alguna incompatibilidad en la configuración de campos o a asociaciones con flujos de trabajo.

Resumen*

Ejemplo de creacion de incidencia

Componentes

Ninguno

Adjunto



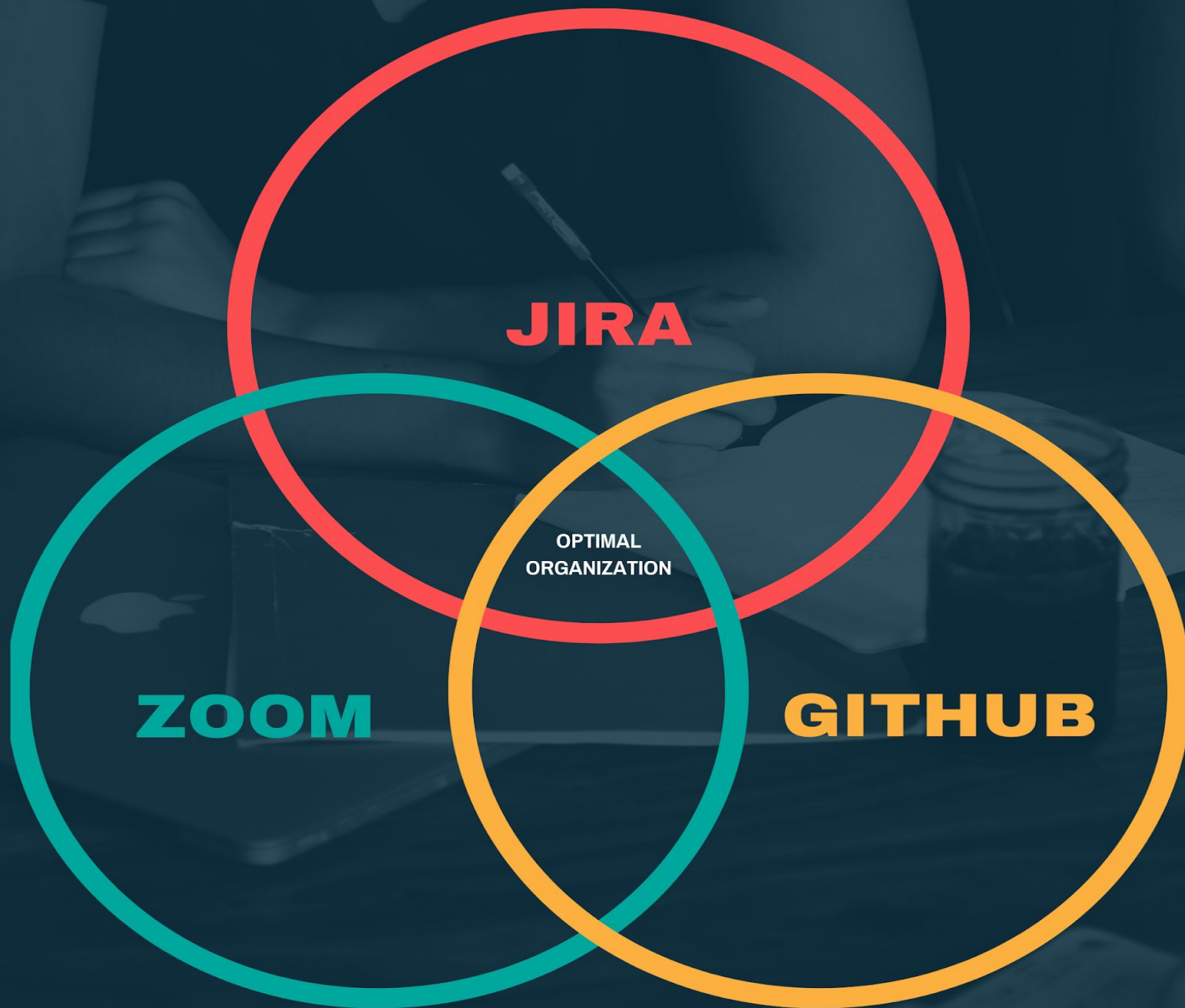
Suelta los archivos para adjuntarlos o [explorar](#).

☐ Crear otra

Crear

Cancelar

Advantages



JIRA

**OPTIMAL
ORGANIZATION**

ZOOM

GITHUB