



RUST

PROGRAMMING LANGUAGE

Lucas Castro (PED) MC346 2021.2
lucas.castro@ic.unicamp.br

PED CLASSES

- 01 **RUST LANGUAGE**
09/11
- 02 GOLANG
11/11
- 03 CLOJURE
16/11
- 04 SCALA
18/11



RUST APPLICATIONS

WEB
ASSEMBLY

SYSTEM
PROGRAMMING

EMBEDDED
SYSTEMS



RUST TIMELINE

- Graydon Hoare - 2006
 - Mozilla
- Brendan Eich
 - JavaScript
 - NPM
 - Mozilla
- 2010 - Releasing
 - Mozilla Research Labs
- 2014 - Public



RUST FEATURES

NO
GARBAGE
COLLECTOR

NO
RUNTIME

YES
OWNERSHIP

Energy Efficiency across Programming Languages

How Does Energy, Time, and Memory Relate?

Rui Pereira
HASLab/INESC TEC
Universidade do Minho, Portugal
ruipereira@di.uminho.pt

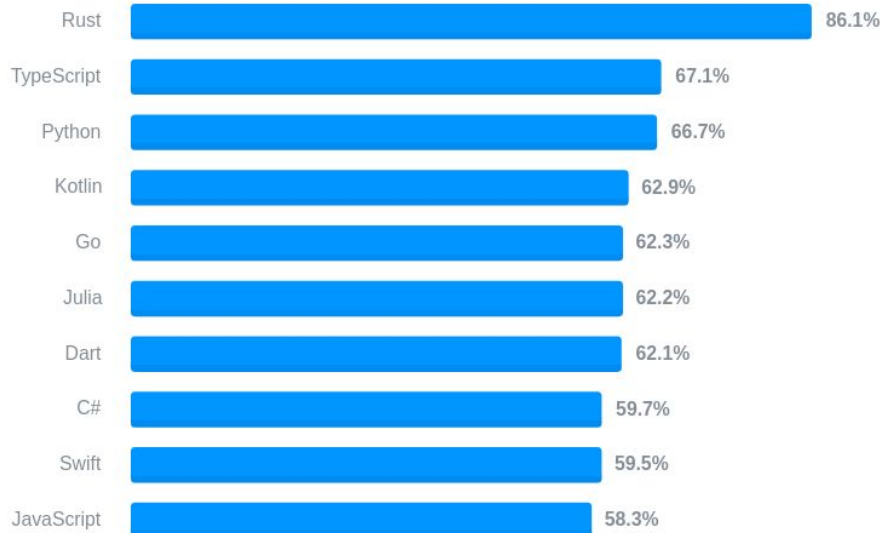
Marco Couto
HASLab/INESC TEC
Universidade do Minho, Portugal
marco.l.couto@inesctec.pt

Francisco Ribeiro, Rui Rua
HASLab/INESC TEC
Universidade do Minho, Portugal
fribeiro@di.uminho.pt
rrua@di.uminho.pt

Jácome Cunha
NOVA LINCS, DI, FCT
Univ. Nova de Lisboa, Portugal
jacome@fct.unl.pt

João Paulo Fernandes
Release/LISP, CISUC
Universidade de Coimbra, Portugal
jpf@dei.uc.pt

João Saraiva
HASLab/INESC TEC
Universidade do Minho, Portugal
saraiva@di.uminho.pt



% of developers who are developing with the language or technology and have expressed interest in continuing to develop with it

[LINK](#)


fasta				
	Energy	Time	Ratio	Mb
(c) Rust ↓ ₉	26.15	931	0.028	16
(c) Fortran ↓ ₆	27.62	1661	0.017	1
(c) C ↑ ₁ ↓ ₁	27.64	973	0.028	3
(c) C++ ↑ ₁ ↓ ₂	34.88	1164	0.030	4
(v) Java ↑ ₁ ↓ ₁₂	35.86	1249	0.029	41
(c) Swift ↓ ₉	37.06	1405	0.026	31
(c) Go ↓ ₂	40.45	1838	0.022	4
(c) Ada ↓ ₂ ↑ ₃	40.45	2765	0.015	3
(c) Ocaml ↓ ₂ ↓ ₁₅	40.78	3171	0.013	201
(c) Chapel ↑ ₅ ↓ ₁₀	40.88	1379	0.030	53
(v) C# ↑ ₄ ↓ ₅	45.35	1549	0.029	35
(i) Dart ↓ ₆	63.61	4787	0.013	49
(i) JavaScript ↓ ₁	64.84	5098	0.013	30
(c) Pascal ↓ ₁ ↑ ₁₃	68.63	5478	0.013	0
(i) TypeScript ↓ ₂ ↓ ₁₀	82.72	6909	0.012	271
(v) F# ↑ ₂ ↑ ₃	93.11	5360	0.017	27
(v) Racket ↓ ₁ ↑ ₅	120.90	8255	0.015	21
(c) Haskell ↑ ₂ ↓ ₈	205.52	5728	0.036	446
(v) Lisp ↓ ₂	231.49	15763	0.015	75
(i) Hack ↓ ₃	237.70	17203	0.014	120
(i) Lua ↑ ₁₈	347.37	24617	0.014	3
(i) PHP ↓ ₁ ↑ ₁₃	430.73	29508	0.015	14
(v) Erlang ↑ ₁ ↑ ₁₂	477.81	27852	0.017	18
(i) Ruby ↓ ₁ ↑ ₂	852.30	61216	0.014	104
(i) JRuby ↑ ₁ ↓ ₂	912.93	49509	0.018	705
(i) Python ↓ ₁ ↑ ₁₈	1,061.41	74111	0.014	9
(i) Perl ↑ ₁ ↑ ₈	2,684.33	61463	0.044	53

[Reference](#)



RUST NA SUA MÁQUINA

- Official guide
 - Linux and MAC*
 - `$ curl --proto '=https' --tlsv1.2 https://sh.rustup.rs -sSf | sh`
 - `rustup update`
 - `code hello.rs`
 - `rustc hello.rs`
 - `./hello`

```
home > lcastro > Documents >  hello.rs
1  fn main(){
2      println!("Hello World!");
3  }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[lcastro@lucaslaptop Documents]$ rustc hello.rs
[lcastro@lucaslaptop Documents]$ ./hello
Hello World!
[lcastro@lucaslaptop Documents]$ 
```

*Windows - Follow the Official guide



RUST – CARGO

- Official documentation
 - cargo new <projectName>
 - cd <projectName>
 - cargo build

```
1 fn main() {
2     println!("Hello, world!");
3 }
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[lcastro@lucaslaptop person]$ cargo build
    Finished dev [unoptimized + debuginfo] target(s) in 0.02s
[lcastro@lucaslaptop person]$ ls
Cargo.lock Cargo.toml src target
[lcastro@lucaslaptop person]$ ./target/debug/
build/     deps/     examples/ .fingerprint/ incremental/ person
[lcastro@lucaslaptop person]$ ./target/debug/person
Hello, world!
[lcastro@lucaslaptop person]$
```





RUST ONLINE

- <https://play.rust-lang.org/>

The screenshot shows the Rust Online playground interface. At the top, there is a browser address bar with the URL `play.rust-lang.org`. Below the address bar, there is a navigation bar with buttons for `RUN`, `DEBUG`, and `STABLE`. The main area is a code editor with the following Rust code:

```
1 fn main() {  
2     println!("Hello, world!");  
3 }
```

At the bottom of the interface, there is a small notification box that reads: "As of Rust 1.56, the default edition of Rust is now Rust 2021. Learn more about editions in the [Edition Guide](#). To specify which edition to use, use the advanced compilation options menu." The notification box has a close button (X) on the right.



RUST DATATYPES

- Rust datatypes
 - Inere os tipos
 - Tipagem estática

Length	Signed	Unsigned
8-bit	i8	u8
16-bit	i16	u16
32-bit	i32	u32
64-bit	i64	u64
128-bit	i128	u128
arch	isize	usize

Tuple Type

```
home > lcastro > examples.rs
1  fn main() {
2      let tup = (500, 6.4, 1);
3
4      let (x, y, z) = tup;
5
6      println!("The value of y is: {}", y);
7  }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

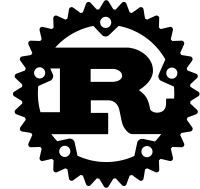
```
[lcastro@lucaslaptop ~]$ ./examples
The value of y is: 6.4
[lcastro@lucaslaptop ~]$
```




RUST DATATYPES

- Array type

```
home > lcastro > examples.rs
1  fn main() {
2      let a = [1, 2, 3, 4, 5];
3      let months = ["January", "February", "March", "April", "May", "June", "July",
4          "August", "September", "October", "November", "December"];
5      let b: [i32; 5] = [1, 2, 3, 4, 5];
6
7      let first = a[0];
8      let second = a[1];
9
10     println!("{}", second);
11
12
13
14 }
```



RUST IF STATEMENT

```
home > lcastro >  if.rs
1  fn main() {
2
3      let n1 = 3;
4      if n1 < 5 {
5          println!("yes");
6      } else {
7          println!("no");
8      }
9
10     let condition = true;
11     let n2 = if condition { 5 } else { 6 };
12     println!("The value of number is: {}", n2);
13 }
```



RUST IMMUTABLE – SECURITY

```
home > lcastro > examples.rs
1  fn main() {
2      let name = "Maria";
3      name = "Joana";
4
5      println!("Nome: {}",name);
6
7  }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

~~~~~

= **note:** `#[warn(unused\_assignments)]` on by default  
= **help:** maybe it is overwritten before being read?

**error[E0384]:** cannot assign twice to immutable variable `name`  
--> examples.rs:3:5

```
2  |     let name = "Maria";
    |     ----
    |     |
    |     first assignment to `name`
    |     help: consider making this binding mutable: `mut name`
3  |     name = "Joana";
    |     ~~~~~~ cannot assign twice to immutable variable
```

**error:** aborting due to previous error; 1 warning emitted

```
home > lcastro > examples.rs
1  fn main() {
2      let mut name = "Maria";
3      name = "Joana";
4
5      println!("Nome: {}",name);
6
7  }
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
[lcastro@lucaslaptop ~]$ ./examples
Nome: Joana
[lcastro@lucaslaptop ~]$
```

# RUST O.O

- Structs
- Impl (key-feature)

```
home > lcastro > examples.rs
1  struct Brand{
2      name: String
3  }
4  struct Car{
5      name: String,
6      brand: Brand,
7      plate: u64
8  }
9
10
11
12 fn main(){
13     let fiat = Brand {name: "Fiat".to_string()};
14     let uno = Car {name: String::from("Uno"), brand: fiat, plate: 999222};
15
16     println!("Eu tenho um {} - placa: {}",uno.name,uno.plate);
17
18 }
19
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

```
[lcastro@lucaslaptop ~]$ ./examples
Eu tenho um Uno - placa: 999222
[lcastro@lucaslaptop ~]$
```

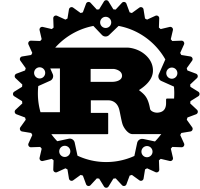
# RUST O.O

- Structs
- Impl (key-feature)

```
home > lcastro > examples.rs
1  struct Brand{
2  |     name: String
3  | }
4  struct Car{
5  |     name: String,
6  |     brand: Brand,
7  |     plate: u64
8  | }
9
10 impl Car{
11 |     fn saymodel(&self) -> String{
12 |         format!("Eu tenho um {} - placa: {}",self.name,self.plate)
13 |     }
14 | }
15
16
17 fn main(){
18 |     let fiat = Brand {name: "Fiat".to_string()};
19 |     let uno = Car {name: String::from("Uno"), brand: fiat, plate: 999222};
20
21 |     println!("{}",uno.saymodel());
22 |
23 | }
24
25
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[lcastro@lucaslaptop ~]$ ./examples
Eu tenho um Uno - placa: 999222
```



# RUST - OWNERSHIP

- Habilita o Rust a não precisar do Garbage Collector
  - Borrowing
  - Slices
- 3 regras básicas
  - Toda variável possui um dono
  - Só existe um dono por vez
  - Quando o dono sai do escopo do valor, o valor é descartado
    - string.rs
    - Scope.rs
- Copy, move, &reference

**ME: I NEED 64 BYTES OF MEMORY**

**RUST**

**YEAH, UMM, IF YOU COULD JUST GO AHEAD AND MAKE SURE THAT YOU'RE BORROWING THIS MEMORY REGION CORRECTLY, THAT'D BE GREAT**

**C**

**HERE'S SOME BYTES FOR YOU MAN**



# RUST - TRAITS



- Semelhante ao comportamento de interface
- Comportamento compartilhado
- Generic datatypes
  - `trait.rs`

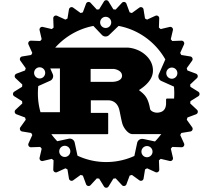
college prof: you learn one programming language, you've learned them all

**Rust:**



Cannot borrow foo as mutable

# RUST - THREADS - CHANNEL



```
home > lcastro > threads.rs
1  use std::sync::mpsc;
2  use std::thread;
3
4  fn main() {
5      let (tx, rx) = mpsc::channel();
6
7      thread::spawn(move || {
8          let val = String::from("hi");
9          tx.send(val).unwrap();
10     });
11
12     let received = rx.recv().unwrap();
13     println!("Got: {}", received);
14 }
```