

APS Lógica da Computação

Luca Coutinho Melão

12/06/2023

```
location.search.substring(1)

arr = qp.split("&")
arr.forEach(function (v,i,arr) {
  v.replace('=','":"') + '"';))
qp = qp ? JSON.parse('{'+arr.join(',')+'}') : {}
function (key, value) {
  return key == "" ? value :
  decodeURIComponent(value)
}
```

MOTIVAÇÃO

Por que desenvolver essa linguagem?



3° Maior país com maior crescimento de desenvolvedores de software fora dos EUA, apenas atrás da Índia e da China.

O Brasil soma mais de **3 milhões** de desenvolvedores no GitHub, com crescimento de **39%** ao ano só em **2022** (924 mil pessoas).

Até **2025**, o Brasil pode enfrentar déficit anual de **159 mil desenvolvedores de software** e serviços de **TI**.

“É importante para nós garantir que todas as pessoas desenvolvedoras tenham oportunidades iguais, conectividade e a melhor experiência possível, independentemente de onde estejam localizadas”

- Julio Viana, gerente regional do GitHub no Brasil

MOTIVAÇÃO

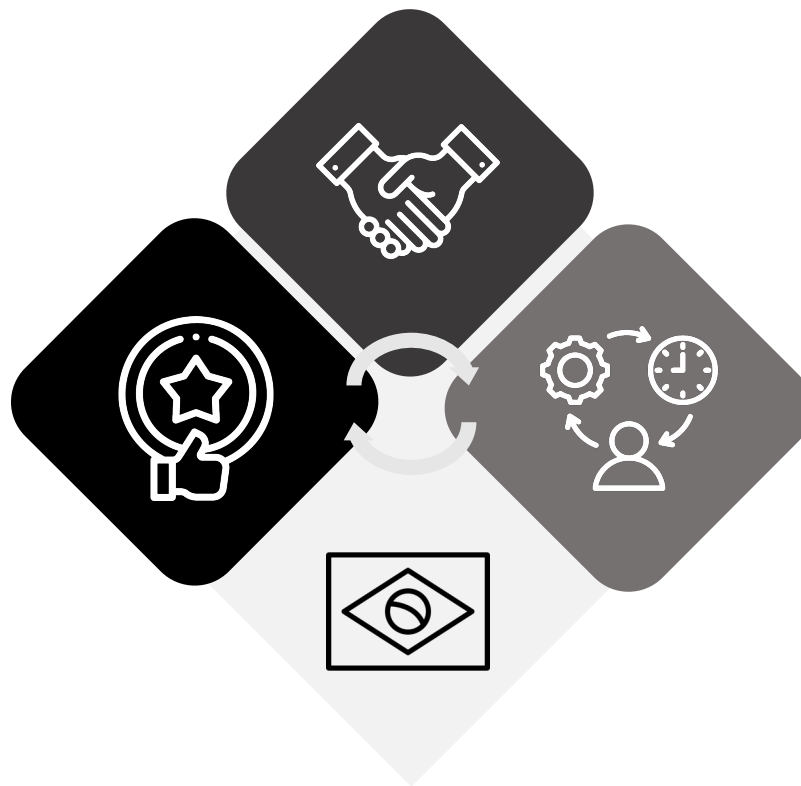
Pilares da Simple Language

INCLUSIVA

Permitir que mais pessoas façam uso de uma linguagem em seu idioma nativo.

DIDÁTICA

Criar uma linguagem para que novos desenvolvedores aprendam as noções de programação básicas.



FÁCIL

Viabilizar o desenvolvimento intuitivo de programas, de modo que a sintaxe não seja uma barreira.

ESTRUTURA DA SIMPLE LANGUAGE

Segundo o padrão EBNF

```
BLOCO = {COMANDO} ;

COMANDO = ( λ | ATRIBUICAO | IMPRESSAO | ENQUANTO | SE | FUNCAO |
RETORNA | CHAMADA), ";", "\n" ;

ATRIBUICAO = IDENTIFICADOR, (["=", RELEXPR] | "=", RELEXPR ) ;

IMPRESSAO = "imprime", "(" RELEXPR ")"' ;

SE = "se", "(", RELEXPR, ")", "{", COMANDO, "}", ["senao", "{",
COMANDO, "}"] ;

ENQUANTO = "enquanto", "(", RELEXPR, ")", "{", COMANDO, "}"' ;

FUNCAO = "funcao", IDENTIFICADOR, "(", [PARAMETRO], ")", "{",
"\n", {BLOCO}, "}"' ;

PARAMETRO = IDENTIFICADOR, {",", IDENTIFICADOR} ;

CHAMADA = IDENTIFICADOR, "(", [RELEXPR, {",", RELEXPR}] , ")";

FATOR = (NÚMERO | STRING | IDENTIFICADOR, ["(", [RELEXPR, {",", RELEXPR}] , ")"] | ("+" | "-" | "!"), FACTOR) | "(", RELEXPR, ")" ;

RELEXPR = EXPRESSAO, { ("<" | ">" | "==" ), EXPRESSAO } ;

EXPRESSAO = TERMO { ("+" | "-", "ou") TERMO } ;

TERMO = FATOR { ("*" | "/" | "e"), FATOR } ;

CHAMADAFUNCAO = IDENTIFICADOR "(" [EXPRESSAO {",", EXPRESSAO}] ")"' ;

IDENTIFICADOR = LETRA, {LETRA | DIGITO | "_"} ;

NÚMERO = DIGITO, { DIGITO } ;

STRING = '"' {LETRA} '"';

LETRA = ( a | ... | z | A | ... | Z ) ;

DIGITO = ( 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 ) ;

RETORNA = "retorna", RELEXPR;
```

CARACTERÍSTICAS

Quais recursos a linguagem oferece? Quais as suas peculiaridades?



COMENTÁRIOS

Para realizar comentários no código, basta adicionar `///`.



VARIÁVEIS

Armazena variáveis do tipo String ou Inteiro, e não exige tipagem na declaração.



FUNÇÕES E LOOPS CONDICIONAIS

A linguagem oferece suporte a funções e loops de maneira intuitiva, usando `{ }` como delimitador.



PONTO E VÍRGULA E CHAVES

A linguagem exige `{ }` e `;` com o intuito de deixar mais claro o que cada trecho de código está executando.

***Observação:** A linguagem não permite múltiplos retornos em funções.

EXEMPLOS

Como a linguagem pode ser aplicada em programas na prática?

Condicional com múltiplas condições lógicas

```
nota1 = 7;
nota2 = 9;
nota3 = 6;
media = (nota1 + nota2 + nota3) / 3;

se (media > 5 e (nota1 > 5 e nota2 > 5 e nota3 > 5)) {
    imprime("Aprovado");
} senao {
    imprime("Reprovado");
}
```

```
idade = 25;
temCarteira = 1;

se ((idade > 17 e idade < 66) e temCarteira == 1) {
    imprime("A pessoa pode dirigir");
} senao {
    imprime("A pessoa não pode dirigir");
}
```

* Mais exemplos no arquivo testes.txt

Funções

```
funcao subtracao(a, b) {
    retorna a - b;
}

x = 5;
y = 3;

resultado = subtracao(x, y);
imprime(resultado);
```

```
funcao media(a, b, c) {
    retorna (a + b + c) / 3;
}

x = 5;
y = 10;
z = 15;

resultado = media(x, y, z);
imprime(resultado);
```

```
funcao fatorial(n) {
    se ((n == 0) ou (n == 1)) {
        resultado = 1;
    } senao {
        resultado = n * fatorial(n - 1);
    }
    retorna resultado;
}
```

```
x = 5;
resultado = fatorial(x);
imprime(resultado);
```

```
funcao ePar(n) {
    se (n / 2 * 2 == n) {
        resultado = 1;
    } senao {
        resultado = 0;
    }
    retorna resultado;
}

imprime("Digite um número: ");
x = entrada();
se (ePar(x) == 1) {
    imprime("O número digitado é par");
} senao {
    imprime("O número digitado é ímpar");
}
```

Muito Obrigado!

Luca Coutinho Melão | lucacm1@al.insper.edu.br

<https://github.com/lucamelao/APS-LogComp-Luca.git>

12/06/2023

```
location.search.substring(1)

arr = qp.split("&")
arr.forEach(function (v,i,arr) {
  v.replace('=','":"') + '"';))
qp = qp ? JSON.parse('{'+arr.join(',')+'}') : {}
function (key, value) {
  return key == "" ? value :
  decodeURIComponent(value)
}
```

RERÊNCIAS UTILIZADAS

Fontes dos dados apresentados nos slides

1. <https://www.convergenciadigital.com.br/Carreira/Brasil-soma-mais-de-3-milhoes-de-desenvolvedores-no-GitHub-61917.html?UserActiveTemplate=mobile>
2. <https://inforchannel.com.br/2022/03/11/cresce-o-numero-de-desenvolvedores-de-software-no-brasil/>
3. <https://tecnoblog.net/noticias/2022/11/09/brasil-se-destaca-no-github-com-mais-de-3-milhoes-de-desenvolvedores-ativos/>
4. <https://exame.com/carreira/razoes-faltam-desenvolvedores-sofware-brasil/>