

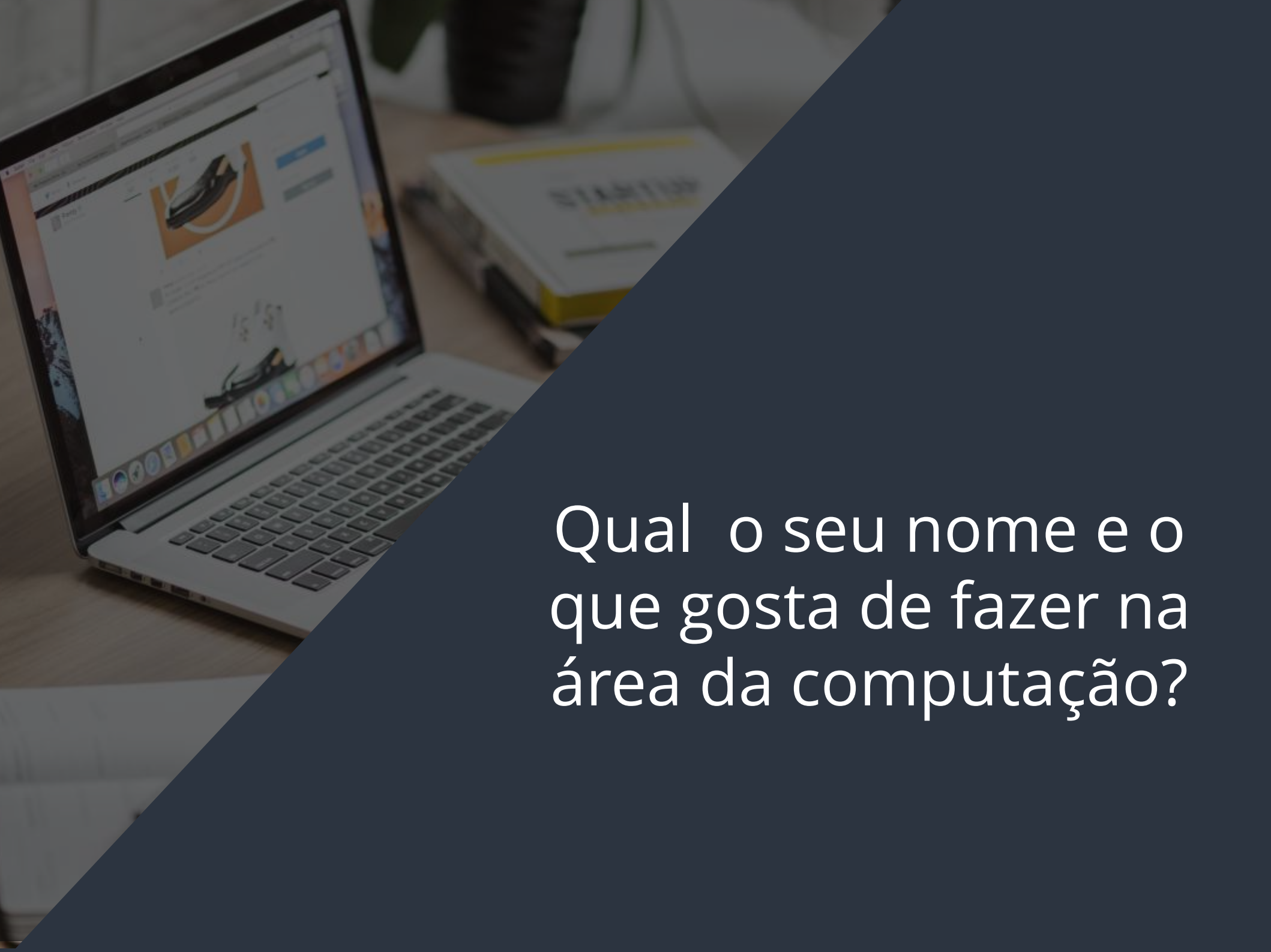


Bem vindos !!!

Mini Curso de Programação

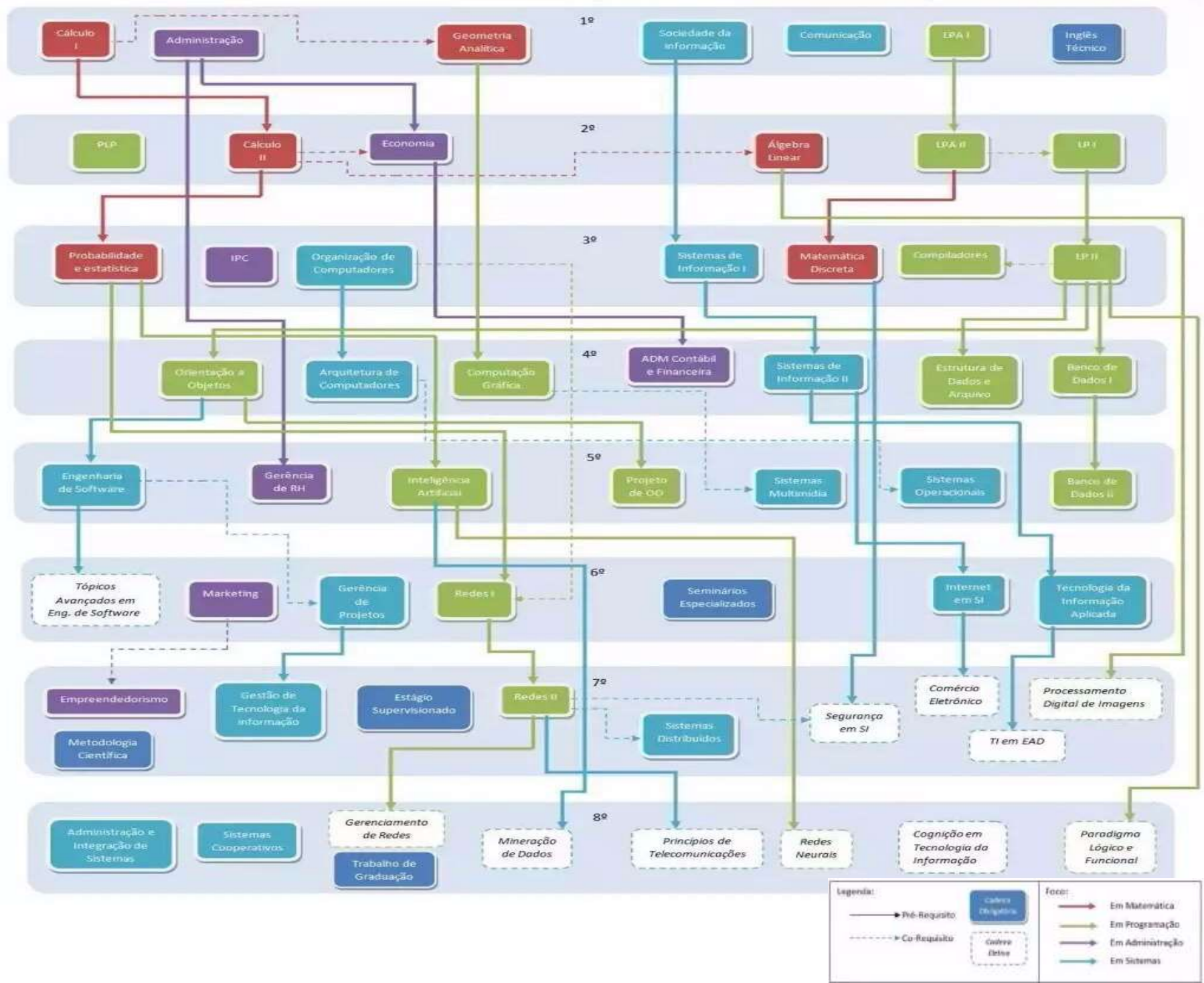
Denini Gabriel

Joédyson Bezerra

A photograph of a laptop on a desk, with a dark blue diagonal overlay on the right side. The laptop screen shows a webpage with a shoe image and some text. The keyboard is visible. In the background, there are some papers and a plant.

Qual o seu nome e o
que gosta de fazer na
área da computação?

Grade Curricular do Curso de Sistema de Informação



Horário 2018.1

Horário/Dia	Segunda	Terça	Quarta	Quinta	Sexta
08:00 – 08:50		LPA1	Comunicação	Cálculo 1	G.A
08:50 – 09:40		LPA1	Comunicação	Cálculo 1	G.A
09:40 – 10:30		LPA1	Sociedade	Cálculo 1	ADM
10:30 – 11:20		LPA1	Sociedade	Cálculo 1	ADM
11:20 – 12:10		G.A	Sociedade	Inglês	ADM
12:10 – 13:00		G.A	Sociedade	Inglês	ADM

Momento de Perguntas

- Sobre o curso
- Sobre a UPE
- Sobre o que quiser





ica de Programa Algoritmos

Pensamento Computacional

Programação

Pensamento Computacional

Permite tomarmos um problema complexo, entender o que é o problema e desenvolver possíveis soluções.

Algoritmos

É um plano. Um conjunto de etapas (passo a passo) e instruções com o objetivo de descrever o caminho até a resolução do problema.

Pensamento Computacional

entradas → algoritmo → saídas



Pseudocódigo

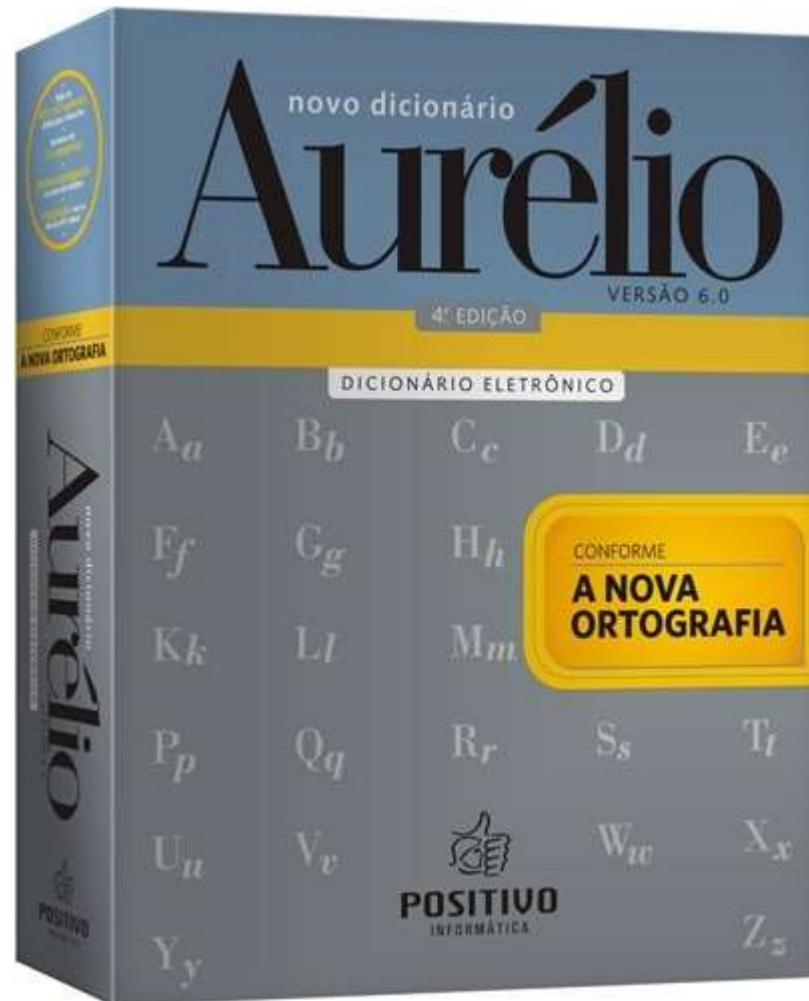
Quantas pessoas tem na sala?
(isso nunca funciona direito)

1.fique de pé

2.pense: **Eu sou número 1**

3.encontre outra pessoa que está em pé, some o seu número com o dela e adote a soma como o seu novo número

4.um de vocês devem se sentar; o outro deve voltar para o passo 3



Vamos criar um simples algoritmo para buscar uma palavra no dicionário:

- 1- Abra numa pagina aleatória;
- 2-procure a palavra na pagina;
- 3- se não encontrar a palavra comece novamente;

1-Procure pela primeira letra da palavra;

2-De acordo com a ordem alfabética vá procurando a palavra;

- 1- abra o dicionário na metade;
- 2- procure sabe se a palavra desejada está no lado direita ou na esquerda;
- 3- rasgue o lado que em que a palavra não está;
- 4- se não encontrar a palavra comece novamente;

Pseudocódigo de Trocar uma lâmpada queimada


















- 1 – Suba na escada
- 2 – Retire a lâmpada
- 3 – Coloque uma lâmpada nova da mesma potência
- 4 - Fim

Pseudocódigo de Trocar uma lâmpada queimada

- 1 - Acionar o interruptor
- 2 - Se a lâmpada não acender, então faça
 - 2.1 - Pegar uma escada
 - 2.2 - Posicionar a escada embaixo da lâmpada
 - 2.3 - Buscar uma lâmpada nova
 - 2.4 - Subir na escada
 - 2.5 - Retirar lâmpada queimada
 - 2.6 - Colocar lâmpada nova
 - 2.7 - Acionar o interruptor
- 3 - Enquanto lâmpada não acender , faça
 - 3.1 - Remover a lâmpada queimada
 - 3.2 - Colocar a lâmpada nova
- 4 - Fim

Crie um pseudocódigo para
fazer e comer um sanduiche

Linguagens 2017

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7


```
1  #include <stdio.h>
2
3  //Primeiro Programa em C
4  int main(){
5
6      printf("Hello world\n");
7
8      return 0;
9  }
```

Variáveis

Os Tipos mais utilizados

```
//Inteiro
```

```
int contador = 0;
```

```
//Float
```

```
float pi = 3.14;
```

```
//Double
```

```
double juro = 1.34567;
```



```
//Char
```

```
char letra = a;
```

```
//String
```

```
char letra[] = "Bem-Vindos";
```

Operações aritméticas

Adição +

Subtração -

Multiplicação *

Divisão /

Operações lógicas

Igual ==

Diferente !=

Maior que > (=)

Menor que < (=)

E &&

OU ||

Estruturas Condicionais

//Se

if (Condição) {

Sentença

//Senão

} **else** {

Sentença

}


```
1  #include <stdio.h>
2
3  //Exemplo de estrutura condicionais em C
4  int main(){
5
6      int a = 3;
7      int b = 2;
8      int soma = a+b;
9
10     if (soma == 5){
11         print("Resultado Correto!\n");
12     }else{
13         print("Resultado Errado!\n");
14     }
15
16     return 0;
17 }
```

Estruturas de Repetição

//Enquanto

```
while ( Expressão ){
```

Sentença

```
}
```

```
1  #include <stdio.h>
2
3  //Exemplo de estrutura repetição em C - While
4  int main(){
5
6      int i = 0;
7
8      while(i < 11){
9          print("Número: %d\n",i);
10         i++;
11     }
12
13     return 0;
14 }
```

//Semelhante ao while

```
for ( Inicializador; Condição; Incremento ){
```

```
    Sentença
```

```
};
```

```
1  #include <stdio.h>
2
3  //Exemplo de estrutura repetição em C - for
4  int main(){
5
6      int i;
7
8      for(i = 0; i < 11; i++){
9          print("Número: %d\n",i);
10     }
11
12     return 0;
13 }
```

Lista

scores									
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
3000	2500								

```
int lista[] = {5, 10, 15, 20, 25};
```




```
gcc -c arquivo.c
```

```
gcc arquivo.o -o programa
```





<https://sourceforge.net/projects/mingw-w64/>

Denini Gabriel

Desenvolvedor C/C++, Flex & Bison, Java

Pesquisador em EAD

github.com/denini08

Joédyson Bezerra

Desenvolvedor C, Javascript, Haskell.

github.com/joedysonbezerra

Monitor de Lógica de Programação e
Algoritmos 1

(LPA 1) Lógica de Programação e Algoritmos 2
(LPA 2).