

AULA 01

{introcomp}

ALGORITMOS E COMPUTAÇÃO

Algoritmos

Imagine a seguinte situação:

Você está em casa e, com seu **carro**, gostaria de ir ao Mercado da Madalena. Como você faria isso?



Você está aqui!



R. Monsenor Silva



R. Ermani Braga



R. José Higino



R. Real da Torre



R. Dr. João Coimbra



R. Real da Torre



R. José Osório

Museu da Abolição



Av. Caxangá

Centro



Passo a passo

- 1 - Vire a esquerda;
- 2 - Siga em frente até o cruzamento;
- 3 - Pare. Se não vier carro, continue. Caso contrário, permaneça parado;
- 4 - Siga em frente até o cruzamento;
- 5 - Pare. Se não vier carro, continue. Caso contrário, permaneça parado;
- 6 - Vire a direita;
- 7- Estacione.



Algoritmos

Você é capaz de seguir essas instruções? Então você fez o que seu computador faz o tempo todo. Você seguiu um algoritmo!



Algoritmos

É a **descrição** de uma **sequência de passos** (ações/instruções) que devem ser seguidas para a **solução de um problema**.



Entrada e Saída

Tudo o que o seu computador faz é executar diversas operações baseadas em certas informações fornecidas a ele. Essas informações são as **entradas**. No problema de chegar ao mercado as entradas seriam as condições em que você estava: o mapa, a sua posição inicial e o fato de você estar de carro.

A **saída** do seu algoritmo é o seu objetivo, ou seja, chegar ao Mercado da Madalena!

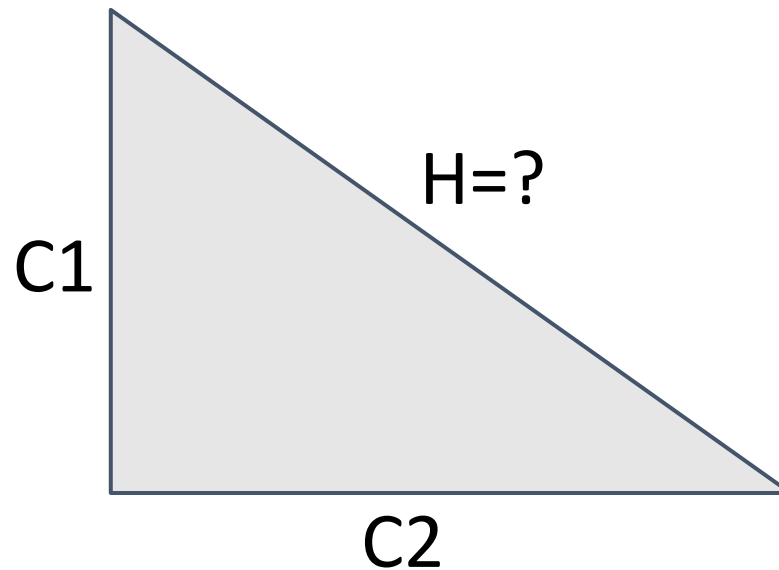


Algoritmos

Exemplo 2: descobrir a hipotenusa de um triângulo retângulo dado dois catetos.

Dados iniciais:

- Cateto 1
- Cateto 2



Algoritmos

Exemplo 2: descobrir a hipotenusa de um triângulo retângulo dado dois catetos.

Algoritmo:

1. Eleve o cateto 1 ao quadrado.
2. Escreva o resultado 1 da operação acima.
3. Eleve o cateto 2 ao quadrado.
4. Escreva o resultado 2 da operação acima.
5. Some o resultado 1 com o resultado 2.
6. Escreva o resultado 3 da operação acima.
7. Tire a raiz quadrada do resultado 3.
8. Escreva o resultado da operação acima que será a hipotenusa.

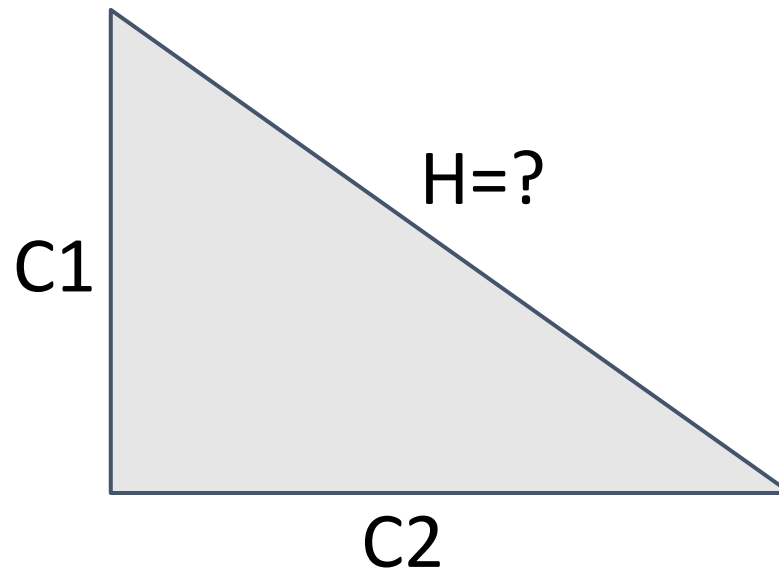


Algoritmos

Exemplo 2: descobrir a hipotenusa de um triângulo retângulo dado dois catetos.

Dados finais:

- Cateto 1
- Cateto 2
- Hipotenusa



Algoritmos

- **Variáveis:**

Conforme observamos, para executar um algoritmo é essencial a manipulação de dados, valores ou objetos. Por exemplo, no problema do Mercado, uma variável é a sua posição.

O mesmo vale para o computador.



Algoritmos

- **Variáveis:**
- Quando um computador executa um algoritmo, o mesmo precisa armazenar e manipular valores.
- Esses valores são armazenados e acessados da memória do computador.



Algoritmos

- **Variáveis:**
- As variáveis podem ser entendidas como “caixas” que podem guardar um valor
- Cada vez que guardamos um novo valor nela, apagamos o antigo
- Para guardarmos um novo valor em uma variável usamos o **comando de atribuição.**



Algoritmos

- **Variáveis:**
- Para a atribuição de valores iremos utilizar uma seta (\leftarrow)
- A variável sempre irá se encontrar à esquerda
- Na parte direita ficará uma expressão
- O resultado da expressão será armazenado pela variável



Algoritmos

- **Variáveis:**

- $x \leftarrow 5$

- $x \leftarrow 2 + 3 * 5$

- $x \leftarrow (2 + 3) * 5$



Algoritmos

- **Variáveis:**
- Exemplo: Determine o valor final da variável x ao final da sequência de passos:

```
↓  
x ← 5  
x ← 2 * x
```

- A. 4
- B. 7
- C. 10
- D. 25



Algoritmos

- **Variáveis:**
- Exemplo: Determine o valor final da variável x ao final da sequência de passos:

```
↓  
x ← 5  
x ← 2 * x
```


- A. 4
- B. 7
- C. 10
- D. 25

R: Alternativa C.



Algoritmos

- **Variáveis:**
- Exemplo: Determine o valor final da variável x ao final da sequência de passos:


 $y \leftarrow 6$
 $z \leftarrow 8$
 $x \leftarrow 2 * z + y$

A. 20

B. 22

C. 16

D. 24



Algoritmos

- **Variáveis:**
- Exemplo: Determine o valor final da variável x ao final da sequência de passos:

```
↓ y ← 6  
z ← 8  
x ← 2 * z + y
```

A. 20

B. 22

C. 16

D. 24

R: Alternativa B.



Algoritmos

- **Variáveis:**
- Exemplo: Determine o valor final da variável x ao final da sequência de passos:

↓	$x \leftarrow 6$
↓	$x \leftarrow 1$
	$x \leftarrow 3 * x$

A. 3

B. 6

C. 9

D. 21



Algoritmos

- **Variáveis:**
- Exemplo: Determine o valor final da variável x ao final da sequência de passos:

↓	x	\leftarrow	6
	x	\leftarrow	1
	x	\leftarrow	$3 * x$

A. 3

B. 6

C. 9

D. 21

R: Alternativa A.



Algoritmos

Um algoritmo pode ser constituído por 3 estruturas de fluxo:

- Estruturas Sequenciais
- Estruturas Condicionais
- Estruturas de Repetição



Algoritmos

- Estruturas Sequenciais

Estrutura sequencial é um conjunto de instruções no qual cada instrução será executada em sequência. **Por exemplo:**

- Siga em frente;
- Vire à esquerda;
- Vire à direita.



Algoritmos

- Estruturas Sequenciais
 - Exemplo: Algoritmo para calcular média de três notas.

```
a <- leiaUmNumeroDoTeclado()
```

```
b <- leiaUmNumeroDoTeclado()
```

```
c <- leiaUmNumeroDoTeclado()
```

```
media <- (a+b+c)/3
```

```
exiba media
```



Algoritmos

- Estruturas Sequenciais
 - Exemplo: Como vc faria um algoritmo que troca o valor de duas variáveis a e b?



Algoritmos

- Estruturas Sequenciais
 - Exemplo: Algoritmo que troca o valor de duas variáveis a e b.

```
auxiliar ← a
```

```
a ← b
```

```
b ← auxiliar
```



Algoritmos

- Estruturas Condicionais

Possibilita a escolha de um grupo de ações e estruturas a serem executadas quando determinadas condições são ou não satisfeitas.

Por exemplo:

- Se não vier carro, continue. Caso contrário, permaneça parado;



Algoritmos

- Estruturas Condicionais

- Acontecem apenas quando alguma condição é verdadeira
- Uma condição só pode ser **verdadeira** ou **falsa**

se (condição) **então**

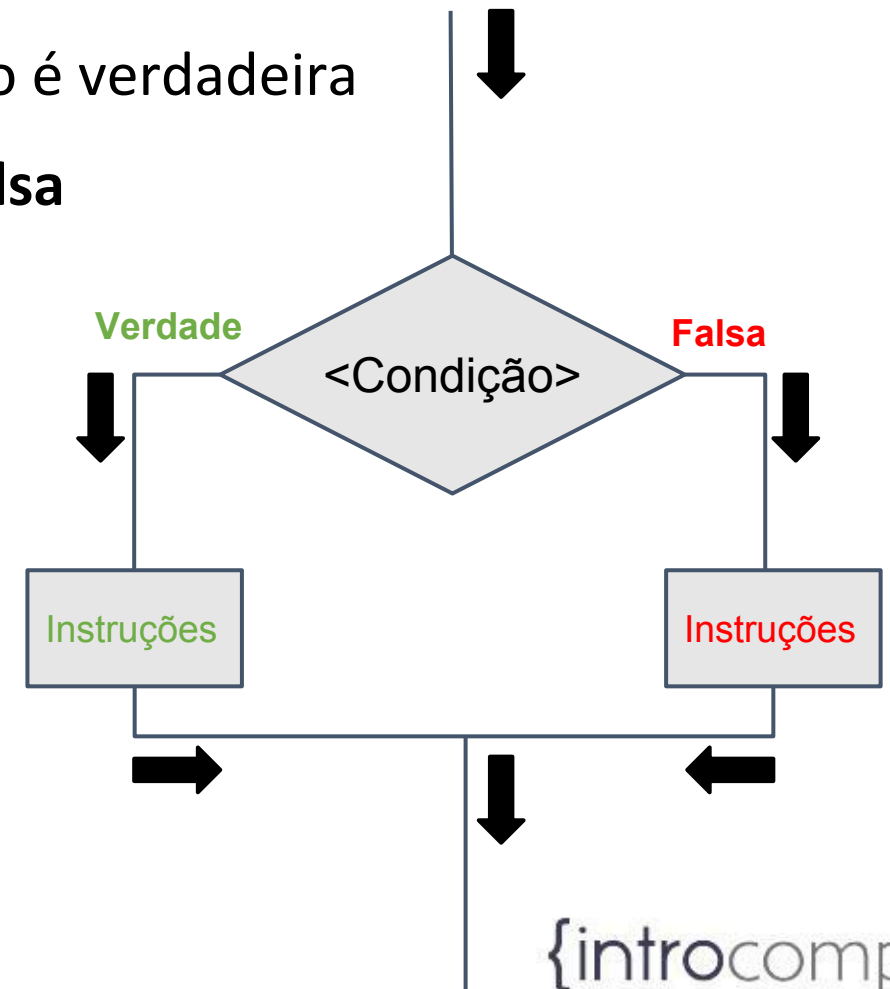
<instruções>

senão

<instruções>

fim-se

...



Algoritmos

- Estruturas Condicionais

- Exemplo: Algoritmo que imprime qual o menor entre dois números

```
a <- leiaUmNumeroDoTeclado()
```

```
b <- leiaUmNumeroDoTeclado()
```

```
se (a < b) então
```

```
    menor <- a
```

```
senão
```

```
    menor <- b
```

```
fim-se
```

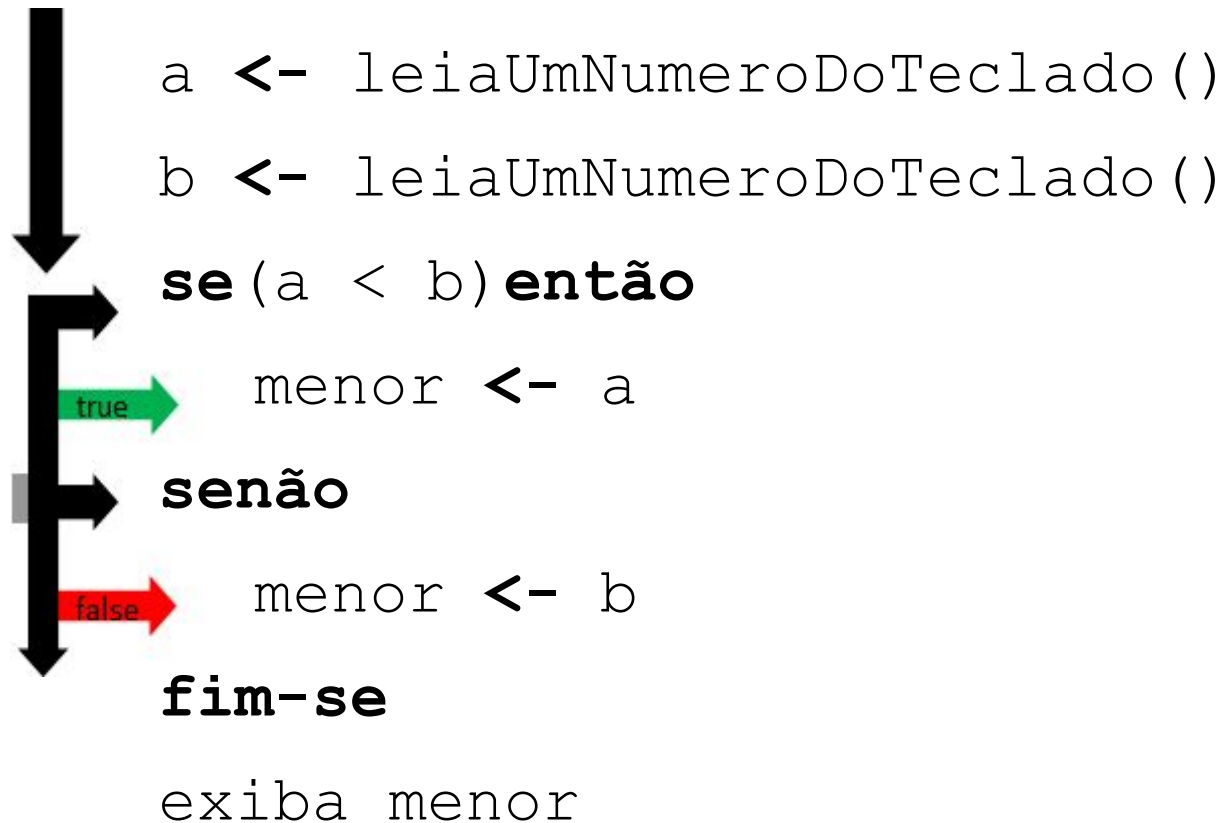
```
exiba menor
```



Algoritmos

- Estruturas Condicionais

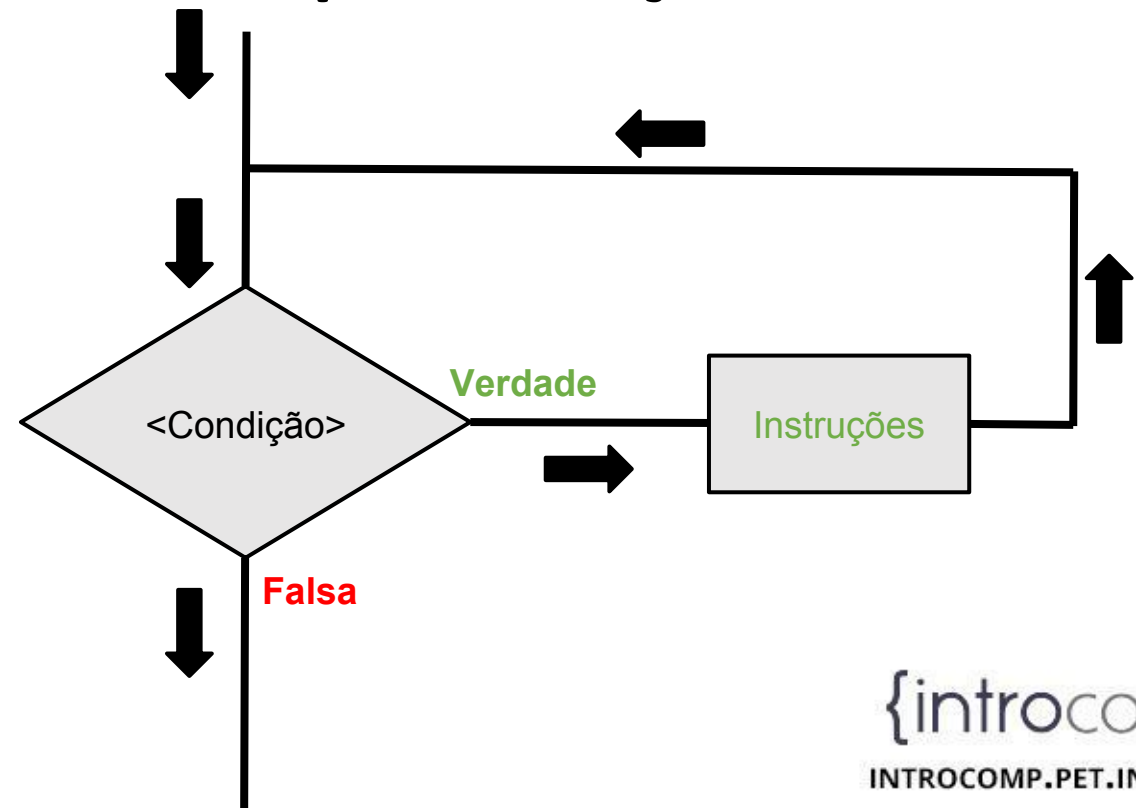
- Exemplo: Algoritmo que imprima qual o menor entre dois numeros



Algoritmos

- Estruturas de Repetição
 - Irão repetir enquanto uma condição for satisfeita
 - São usadas as palavras chave **enquanto-faça**

enquanto (<condição) **faça**
 <instruções>
fim-enquanto



Algoritmos

- Estruturas de Repetição
 - Exemplo: Faça um algoritmo que imprima todos os números de 1 a 20

```
num ← 1
```

```
enquanto (num ≤ 20) faça
```

```
    exiba num
```

```
    num ← num+1
```

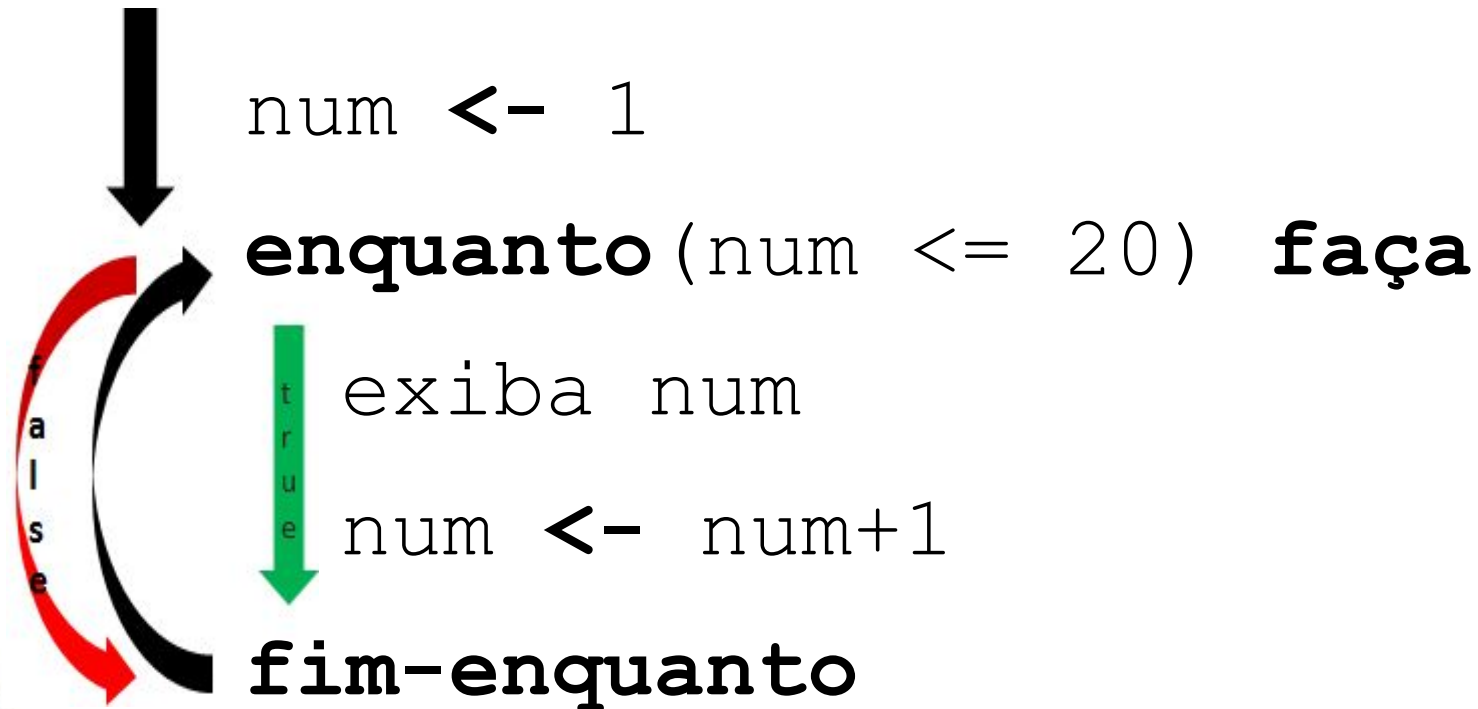
```
fim-enquanto
```



Algoritmos

- Estruturas de Repetição

- Exemplo: Faça um algoritmo que imprima todos os números de 1 a 20



Algoritmos

- Estruturas de Repetição
 - Exemplo: Faça um algoritmo que some todos os números entre 200 e 400

```
num <- 200
```

```
soma <- 0
```

```
enquanto (num <= 400) faça
```

```
    soma <- soma + num
```

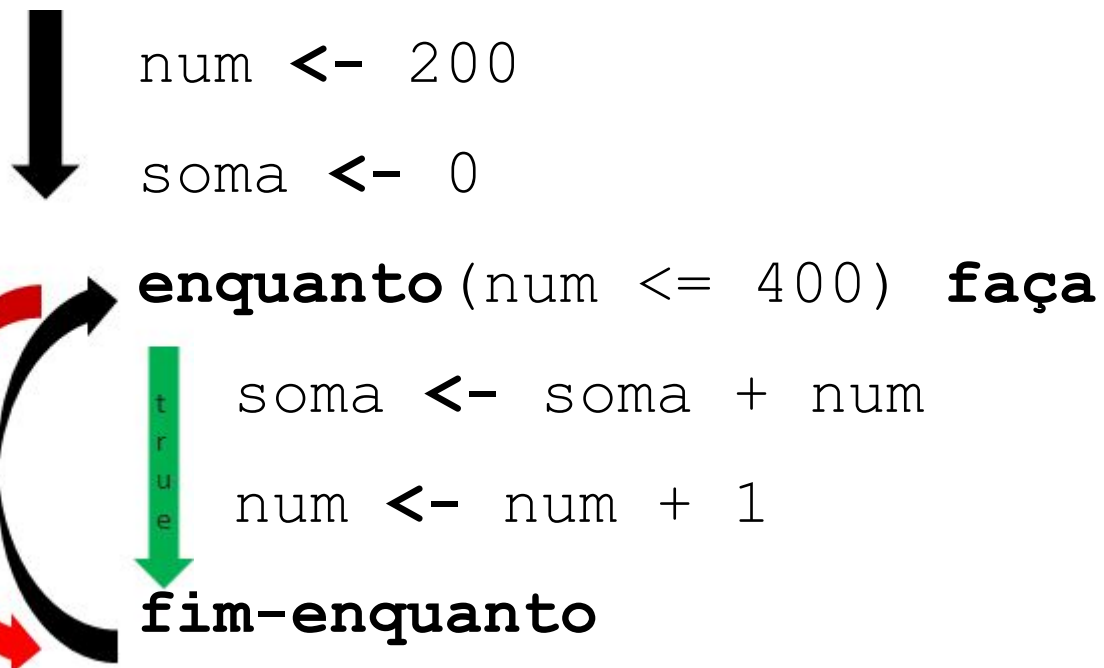
```
    num <- num + 1
```

```
fim-enquanto
```



Algoritmos

- Estruturas de Repetição
 - Exemplo: Faça um algoritmo que some todos os números entre 200 e 400



Algoritmos

- Por que estudar algoritmos?
 - O seu impacto é amplo e de longo alcance.

Internet. Busca na web, roteamento de pacote, compartilhamento de arquivos,...

Biologia. Projeto do genoma humano, enovelamento de proteínas,...

Computador. Layout de circuitos, sistema de arquivos, **compiladores (fique ligado!)...**

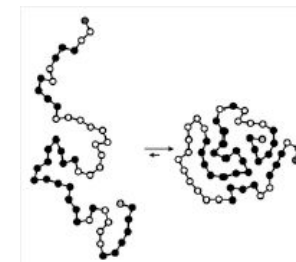
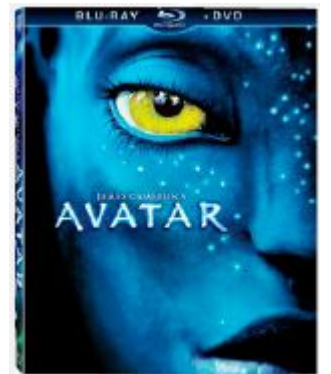
Computação gráfica. filmes, video games, realidade virtual...

Segurança. celular, e-commerce, urna eletrônica,...

Multimídia. MP3, JPG, DivX, HDTV, reconhecimento de face,

Redes Sociais. recomendações, feeds de notícia, propagandas,...

Física. simulação de partículas, simulação de colisão de partículas,...



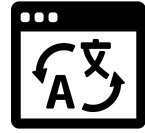
Linguagens de Programação

As pessoas se comunicam umas com as outras por meio de alguma linguagem. Podemos nos comunicar em português, inglês, francês, etc. Da mesma forma, precisamos de um jeito para nos comunicar com o computador. Para isso, utilizamos alguma linguagem de programação, como C, C++, Java, Python, PHP, Lua, etc.

Nesse curso, vamos aprender a linguagem C!



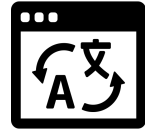
Compilador



As linguagens citadas anteriormente são chamadas de linguagem de alto nível, pois usam palavras-chave muito próximas da realidade. Por exemplo: if/else, while, etc.



Compilador

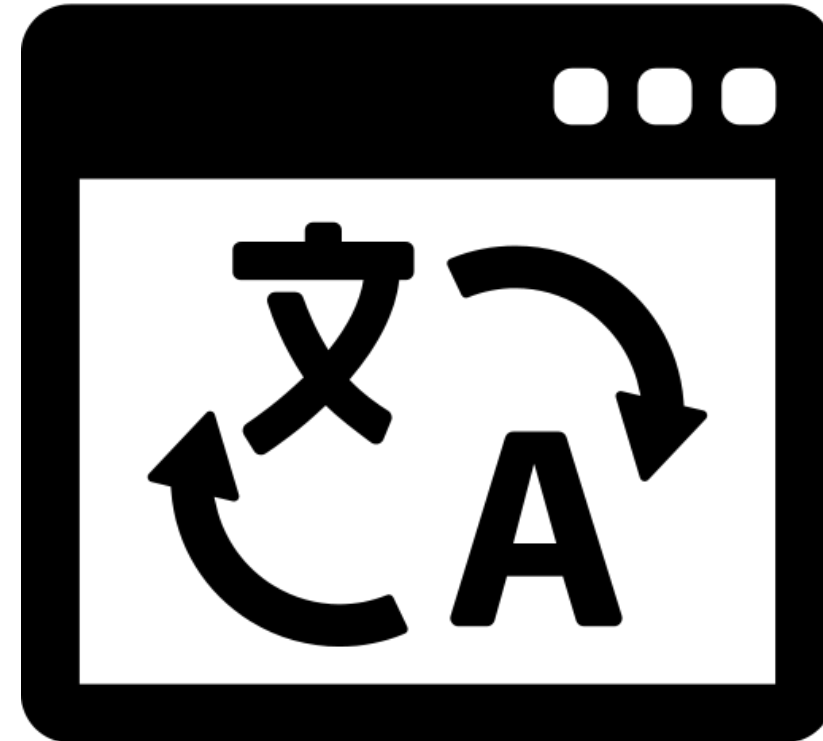


Entretanto, o computador precisa só entender linguagem de máquina. Dessa forma, precisamos de um “tradutor” que “traduza” nosso código em linguagem de alto nível para linguagem de máquina. O responsável por fazer isso é o compilador!



Linguagens de Programação

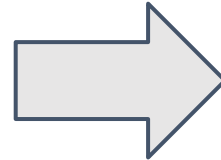
- Compilador
 - Traduz linguagem de alto nível (código fonte) em linguagem de de baixo nível (código objeto) na forma de um executável
 - Exemplos: gcc, Turbo C, Visual C, Visual Basic, etc
 - C é uma linguagem compilada!



Linguagens de Programação

```
#include <stdio.h>

int main()
{
    int a = 10, b = 5;
    printf ("%d\n", a+b);
    return 0;
}
```

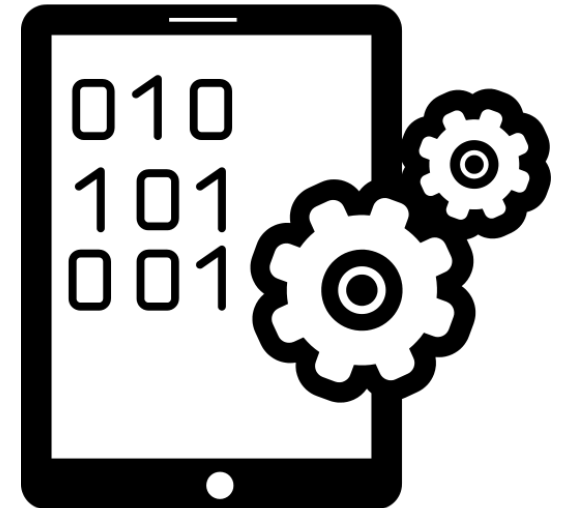
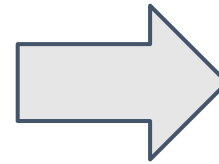


```
.data
x1: .word 10
x2: .word 5
x3: .word -1

.text
lw $t0, x1
lw $t1, x2
add $t2, $t0, $t1
sw $t2, x3

li $v0, 1
move $a0, $t2
syscall

li $v0, 10
syscall
```



[run.codes]



Sobre o Run.Codes

O Run.Codes é um site que irá analisar os exercícios passados em aula. Serão realizados vários casos testes para verificar se seu programa está bem codificado e otimizado.

Vale ressaltar que em alguns exercícios a nota do site não vai condizer 100% com sua nota real, pois seu código ainda será avaliado pelos monitores.

Para acessar o site: <https://run.codes/>



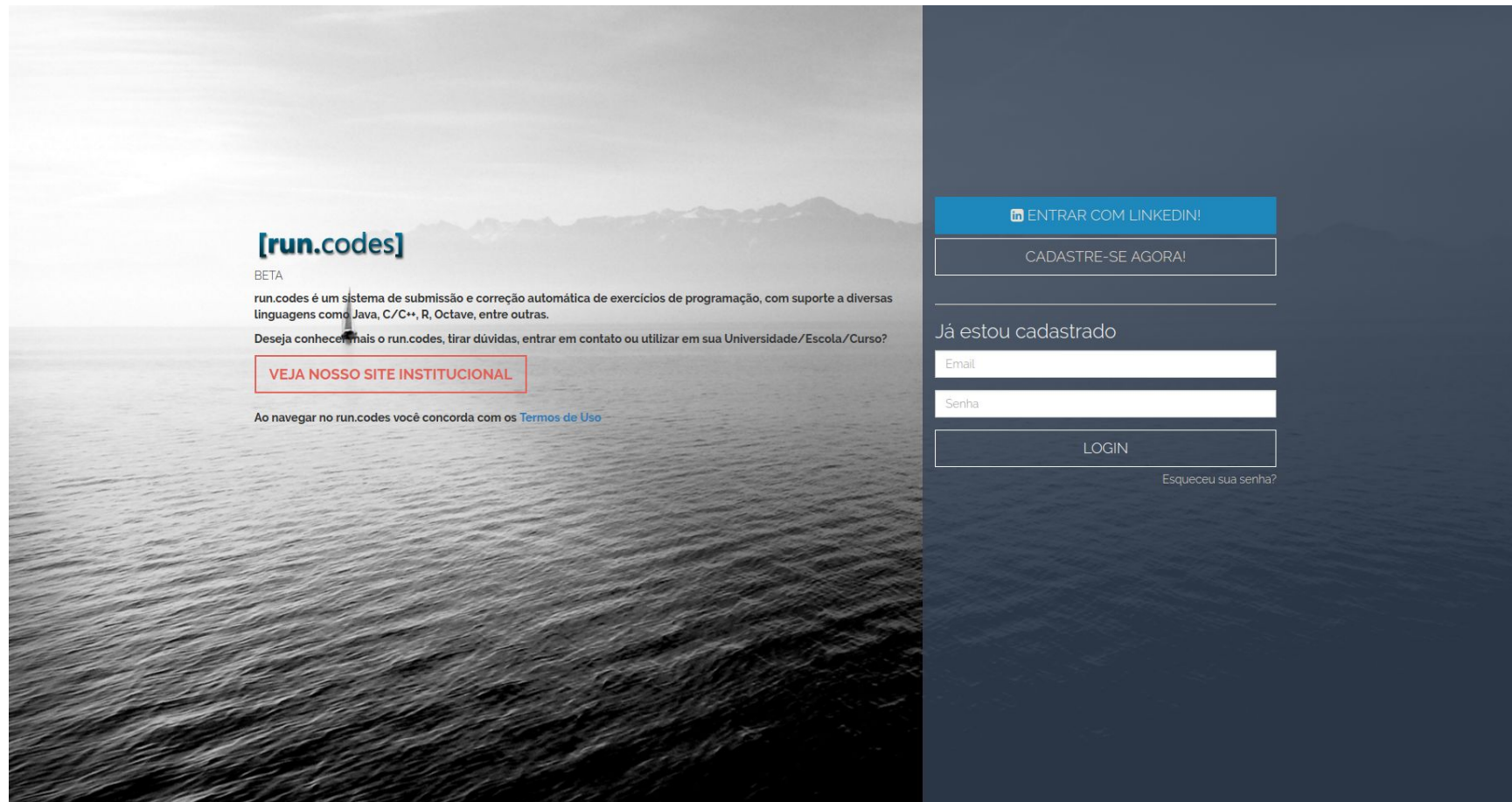
Sobre o Run.Codes

Para fazer o cadastro no site é muito simples, só seguir os passos que serão mencionados.



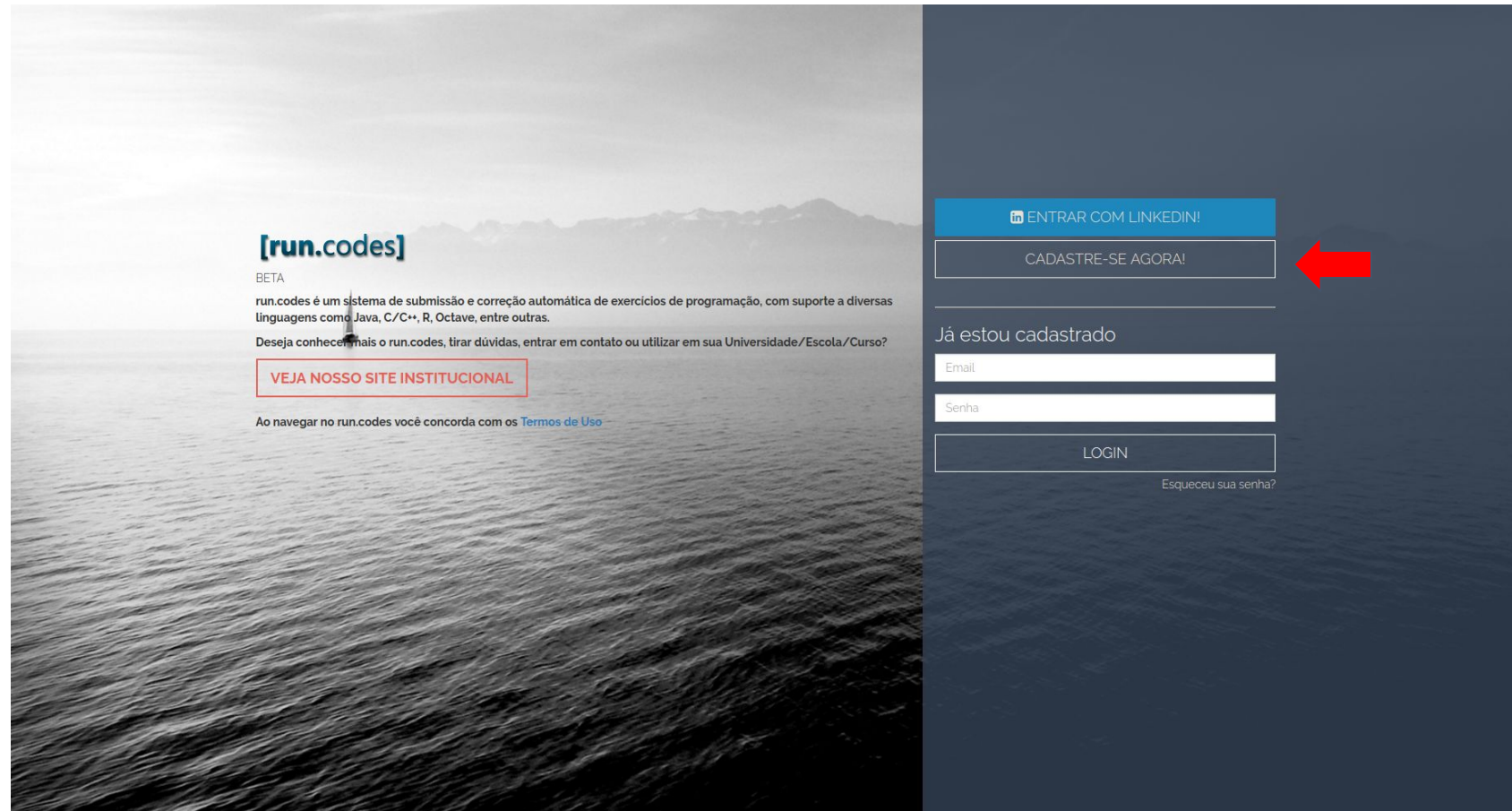
Sobre o Run.Codes

1) Logo após entrar no site você vai se deparar com essa tela:



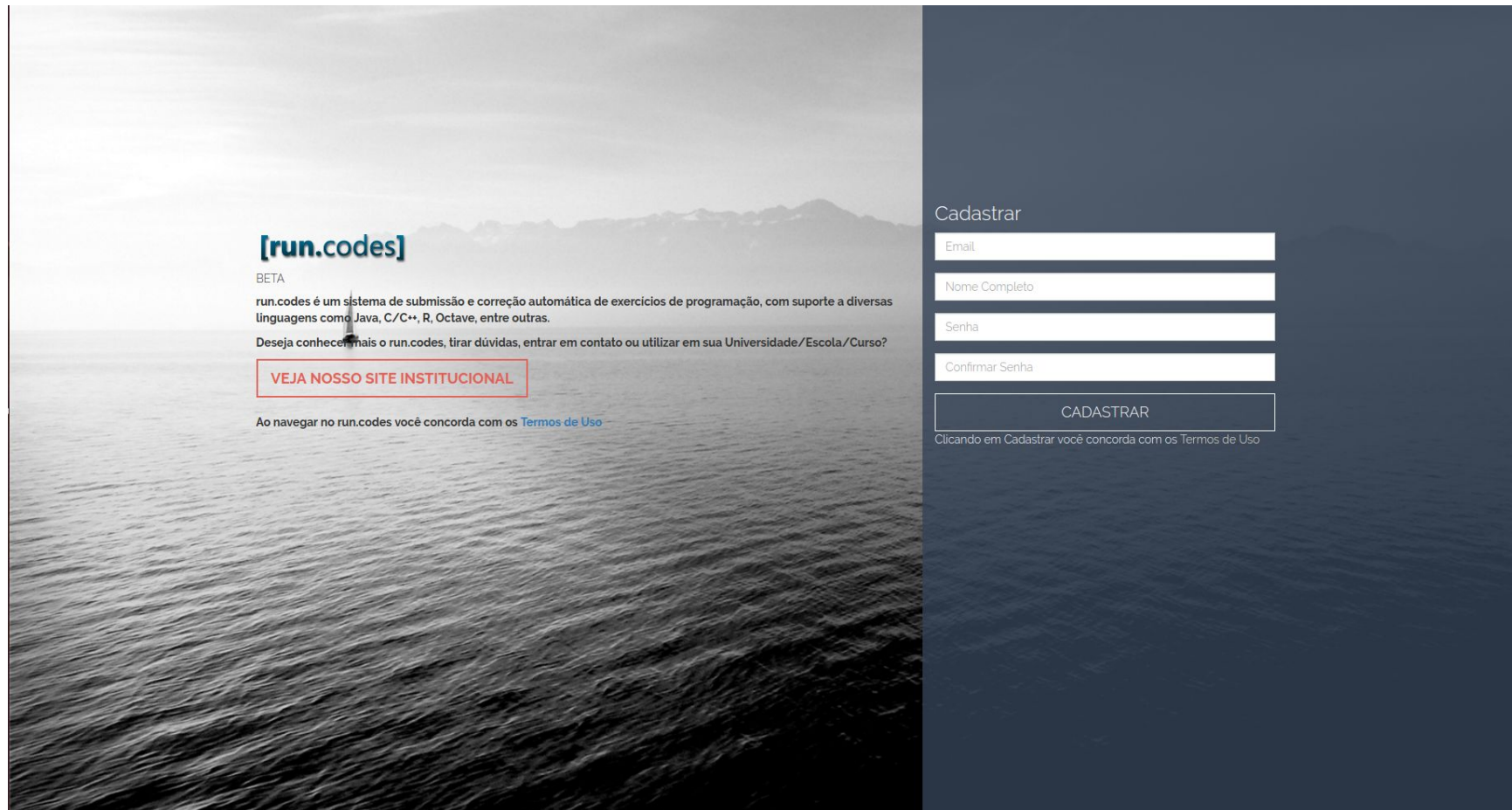
Sobre o Run.Codes

- Clique em “CADASTRE-SE AGORA”



Sobre o Run.Codes

- Preencha seus dados e clique em cadastrar.



[run.codes]
BETA
run.codes é um sistema de submissão e correção automática de exercícios de programação, com suporte a diversas linguagens como Java, C/C++, R, Octave, entre outras.
Deseja conhecer mais o run.codes, tirar dúvidas, entrar em contato ou utilizar em sua Universidade/Escola/Curso?
[VEJA NOSSO SITE INSTITUCIONAL](#)
Ao navegar no run.codes você concorda com os [Termos de Uso](#)

Cadastrar

Email

Nome Completo

Senha

Confirmar Senha

CADASTRAR


Clicando em Cadastrar você concorda com os [Termos de Uso](#)



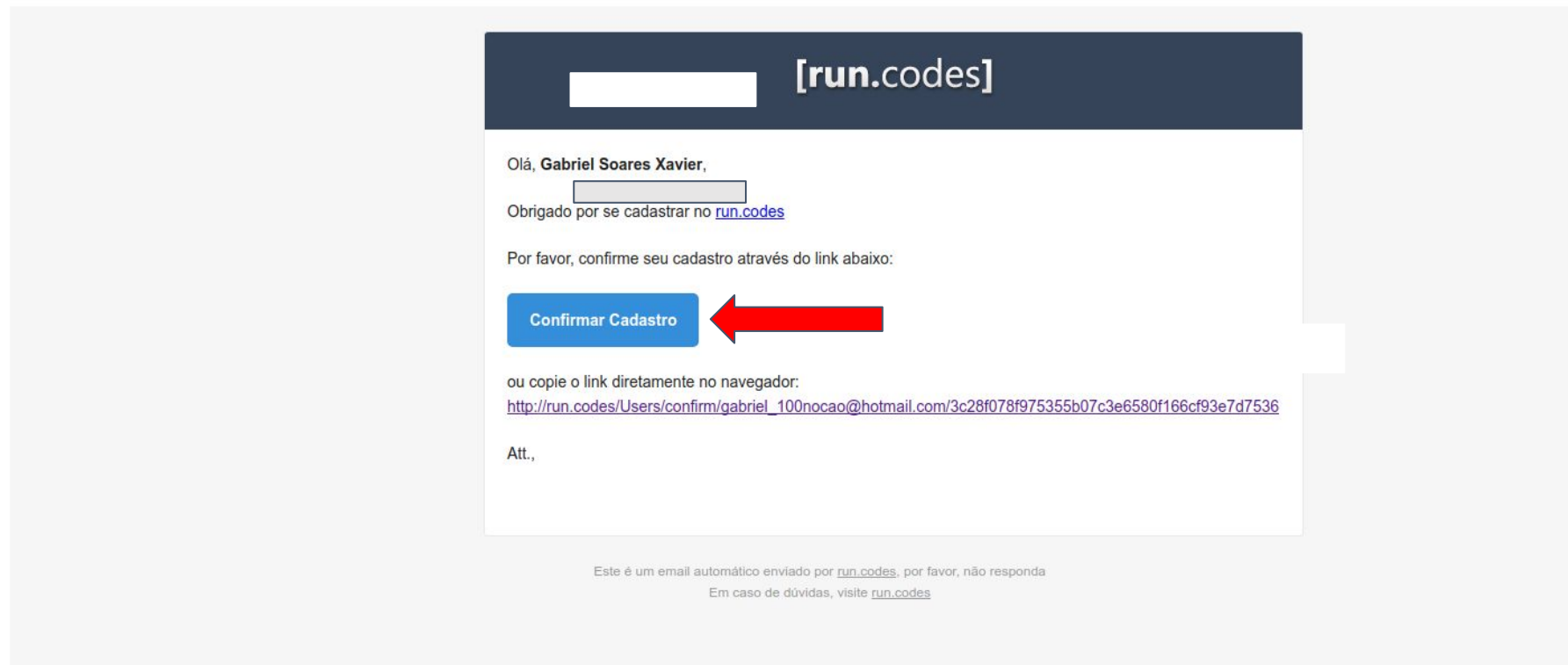
Sobre o Run.Codes

- Abra seu email e confirme seu cadastro.

Confirme seu cadastro no run.codes

 run.codes <no-reply@run.codes>
Hoje, 13:18
Você ↕

Responder | ▾



Sobre o Run.Codes

Por favor, complete seu cadastro com as informações sobre sua universidade

Dados Pessoais

Nome Completo

Universidade

Identificação de Aluno

Confirmar Alterações

Trocar Senha

Senha Atual

Nova Senha

Confirmar Senha

Confirmar Alterações

No campo “Universidade” coloque:
“PET-UFES-PET-Universidade Federal do Espírito Santo”.

No campo “Num.Matrícula” coloque seu
número de inscrição do introcomp.



Sobre o Run.Codes

- Aqui nessa etapa, em “Código de Matrícula”, coloque a matrícula que foi passada em aula.

Nova Matrícula

Código de Matrícula:

Se você não sabe o código de matrícula da sua turma, contate o seu professor

Matricular












Sobre o Run.Codes

Clicando em ver “Página da Disciplina” você será redirecionado a sua turma e poderá começar a enviar seus códigos para avaliações.

Minhas Disciplinas	Nova Matrícula
PETUFES01 Introcomp - Turma 2017	Código de Matrícula:
Ver Página da Disciplina	<input type="text"/>
Ver Turmas Antigas	Se você não sabe o código de matrícula da sua turma, contate o seu professor
	Matricular



Sobre o Run.Codes

56	Desafio - Hacking Day de Bancos de Dados	Não Entregue	0/0	0	2	2/99	18/11/2017 23:59:59 	Ver Detalhes Remover Exercício 
57	Trabalho Computacional	Não Entregue	0/0	0	38	23/99	11/12/2017 04:00:00 	Ver Detalhes Remover Exercício 
58	Working 1 - Praticando 3	Não Entregue	0/0	0	96	61/99	01/01/2019 16:31:19 	Ver Detalhes Remover Exercício 
59	Working 1 - Praticando 1	Não Entregue	0/0	0	91	67/99	01/01/2019 16:32:45 	 Ver Detalhes Remover Exercício 

- O cadeado verde significa que a questão ainda está em aberto.

- Clique em Ver Detalhes para enviar o exercício.




Sobre o Run.Codes



Novo Envio

Você pode submeter um arquivo até
01/01/2019 16:32:45

 [Selecionar Arquivo](#)

Ao realizar esta submissão você garante que é o autor do(s) arquivos enviados ou detêm autorização para envio de partes da sua submissão pertencentes a terceiros. Além disso, autoriza o sistema run.codes a realizar a correção automática e a verificação de eventuais plágios. Também afirma que não está violando os Termos de Uso

- Clique em “Selecionar Arquivo” para submeter a questão.




Sobre o Run.Codes

Novo Envio G

Você pode submeter um arquivo até
01/01/2019 16:32:45

[Selecionar Arquivo](#)

150.txt - 268 bytes

 [Confirmar](#)

Ao realizar esta submissão você garante que é o autor do(s) arquivos enviados ou detêm autorização para envio de partes da sua submissão pertencentes a terceiros. Além disso, autoriza o sistema run.codes a realizar a correção automática e a verificação de eventuais plágios. Também afirma que não está violando os Termos de Uso

- Clique em “Confirmar” para enviar o arquivo.



Sobre o Run.Codes

IP: 200.137.66.1	2017100551	25/04/2018 17:01:19	Submetida	0	0.00	Detalhes
------------------	------------	---------------------	-----------	---	------	--------------------------



- Clique em “Detalhes” para acessar os detalhes da sua submissão.

Informação: [Baixar Arquivo](#)

Submetido por: Gabriel Soares Xavier (gabriel_100nocaio@hotmail.com)
Núm. Matrícula: 2017100551
Data: 25/04/2018 17:01:19

status Submetida	compilado Não	casos corretos 0/0	pontuação 0.00
----------------------------	-------------------------	------------------------------	--------------------------

Submetida: Sua entrega não será processada automaticamente, ela está aguardando uma resposta manual dos responsáveis pela turma

- Aguarde a correção. Pode durar alguns segundos



Sobre o Run.Codes

Informações Gerais: [Baixar Arquivo](#)

Submetido por:
Núm. Matrícula: 150
Data: 01/09/2017 20:27:14

Finalizado status **Sim** compilado **1/1** casos corretos **10.00** pontuação

Finalizado: Sua entrega obteve resposta correta em todos os casos
Mensagem da Compilação:
Compilado com Sucesso

Editar Nota: [Recompilar](#) [Ver Arquivo](#)

Nova Nota

Status
Finalizado

[Confirmar Mudança](#)

Resultado dos Casos:

Caso	Status	Tempo de CPU	Mensagem
Caso 1	Correto	0.0000 s	Resposta Correta

Detalhes dos Casos de Teste

✓ Caso 1

- Essa será sua tela de detalhes da questão. Aqui você verá em quais casos o seu programa não rodou.



Sobre o Run.Codes

Considerações finais:

- Você poderá enviar quantas vezes quiser a mesma questão, mas só será levada em consideração a última enviada.
- Qualquer dúvida podem procurar seus professores.



AULA 01

{introcomp}

ALGORITMOS E COMPUTAÇÃO