

# Unidade 0 - Nivelamento - Processo AEDs II de Exercícios



**PUC Minas**

Instituto de Ciências Exatas e Informática  
Departamento de Ciência da Computação

# Agenda

- Introdução
- Comparação de Arquivos com *DIFF*
- Redirecionamento de Entrada e Saída
- Plataforma Verde
- Exercícios

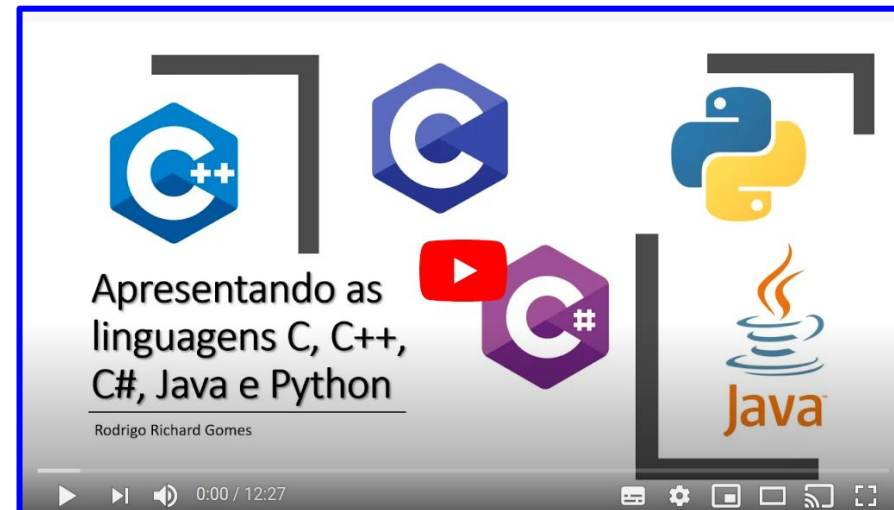
- **Introdução**
  - Comparação
  - Redirecionamento
  - Plataforma
  - Exercícios
- Contextualização: *for AEDs II*
  - Questões iniciais
  - Processo em um Exercício de Programação
  - Processo AEDs II de Exercícios

- **Introdução**
  - Comparação
  - Redirecionamento
  - Plataforma
  - Exercícios
- **Contextualização: *for AEDs II***
    - Questões iniciais
    - Processo em um Exercício de Programação
    - Processo AEDs II de Exercícios

# Linguagens C, Java, C++, C# e Python

- Veja os vídeos do Prof. Rodrigo Richard

- <https://youtu.be/-Z7E-TKFeas>
- <https://youtu.be/hyBXjChj4Pk>
- <https://youtu.be/MKItMDAGqPA>



# Sistema Operacional Linux



# Alguns Arquivos Interessantes no Linux

`~/.bash_history`      *// Lista de comandos executados*

`~/.vimrc`            *// Configurações do VIM*

`~/.profile`        *// Configurações do seu Usuário*

# Alguns Comandos Interessantes no Modo Shell

man	<i>// manual do linux</i>
exit	<i>// fechar o shell</i>
cd	<i>// mudar de pasta</i>
rm	<i>// apagar</i>
ls (e.g., ls -l -t -h)	<i>// listar pasta</i>
cp	<i>// copiar</i>
mkdir	<i>// criar pasta</i>
clear	<i>// limpar a tela</i>
chmod	<i>// alterar permissão</i>
grep (grep "PALAVRA" ARQUIVO)	<i>// procurar padrão</i>
wget	<i>// efetuar download</i>



# Alguns Comandos Interessantes no Modo Shell

<code>diff</code>	<i>(diff ARQUIVO1 ARQUIVO2)</i>	<i>// comparar arquivos</i>
<code>ssh</code>		<i>// protocolo ssh</i>
<code>sftp</code>		<i>// protocolo sftp</i>
<code>javac</code>		<i>// compilador Java</i>
<code>java</code>		<i>// interpretador Java</i>
<code>gcc</code>		<i>// compilador C</i>
<code>g++</code>		<i>// compilador C++</i>
<code>sudo</code>		<i>// executar comando como root</i>
<code>apt-get</code>		<i>// manipulação de pacotes apt</i>
<code>tar -zcvf file.tar.gz pasta</code>		<i>// compactar pasta</i>
<code>tar -zxvf file.tar.gz</code>		<i>// descompactar pasta</i>



# Alguns Comandos Interessantes no VIM

i	<i>// abrir modo de inserção</i>
ESC	<i>// fechar modo de inserção</i>
:w	<i>// salvar</i>
:q	<i>// sair</i>
:q!	<i>// forçar saída</i>
:wq!	<i>// salvar e forçar a saída</i>
? <u>PALAVRA</u>	<i>// procurar as ocorrências de <u>PALAVRA</u></i>
:%s/ <u>OLD</u> / <u>NEW</u> /gc	<i>// substituir as ocorrências de <u>OLD</u> por <u>NEW</u></i>

# Alguns Comandos Interessantes no VIM

<code>yy</code>	<i>// copiar uma linha</i>
<code><u>NÚMERO</u> yy</code>	<i>// copiar <u>NÚMERO</u> linhas</i>
<code>dd</code>	<i>// excluir uma linha</i>
<code><u>NÚMERO</u> dd</code>	<i>// excluir <u>NÚMERO</u> linhas</i>
<code>dw</code>	<i>// excluir palavra</i>
<code><u>NÚMERO</u> dw</code>	<i>// excluir <u>NÚMERO</u> palavras</i>
<code>p</code>	<i>// colar</i>
<code>:sp <u>ARQUIVO</u></code>	<i>// abrir o arquivo <u>ARQUIVO</u></i>
<code>ww</code>	<i>// alternar entre os arquivos abertos</i>
<code>CTRL+V</code>	<i>// abrir modo de visualização</i>

- **Introdução**
- Comparação
  - Contextualização: *for AEDs II*
  - **Questões iniciais**
  - Processo em um Exercício de Programação
  - Processo AEDs II de Exercícios
- Redirecionamento
- Plataforma
- Exercícios

# Questões Iniciais

- Quando temos um exercício de programação:
  - Como sabemos que nosso programa está correto?
  - Quanto tempo esperamos pela correção do programa?
  - Alguém já “bateu o olho” no seu programa e identificou um erro?

- **Introdução**
  - Comparação
  - Redirecionamento
  - Plataforma
  - Exercícios
- Contextualização: *for AEDs II*
  - Questões iniciais
  - **Processo em um Exercício de Programação**
  - Processo AEDs II de Exercícios

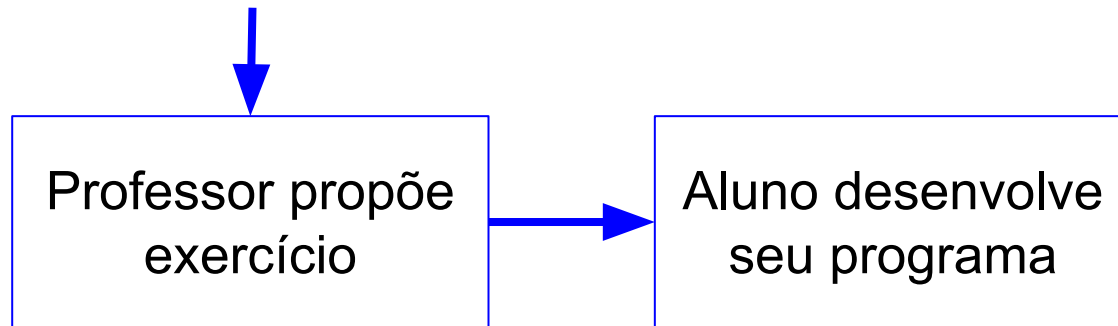
# Processo em um Exercício de Programação



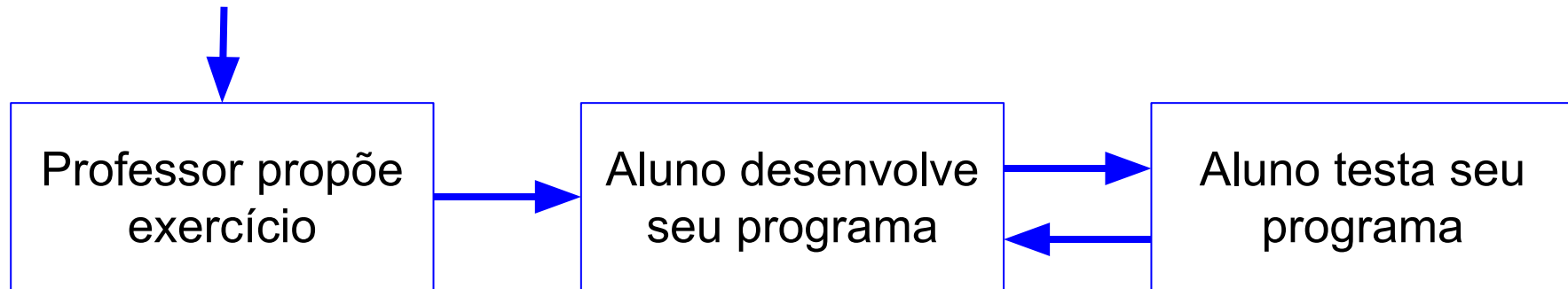
Professor propõe  
exercício



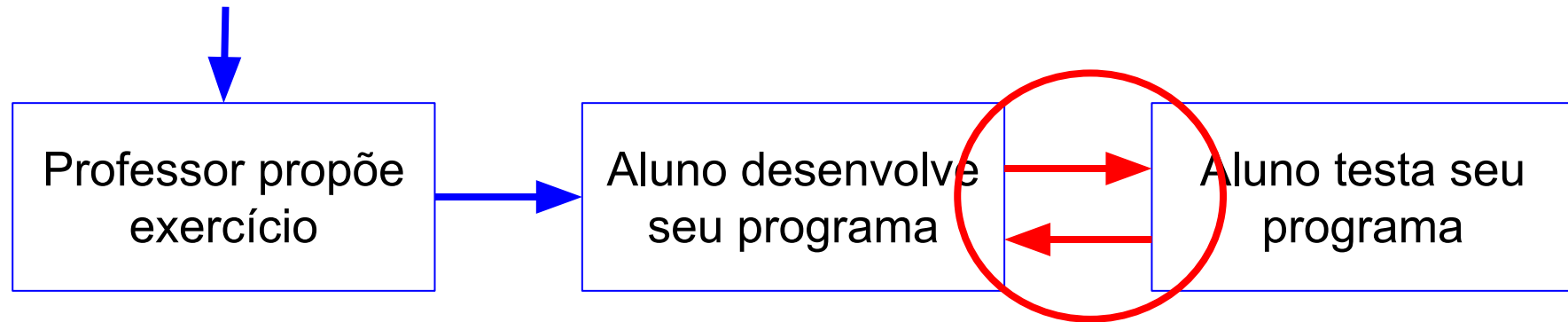
# Processo em um Exercício de Programação



# Processo em um Exercício de Programação

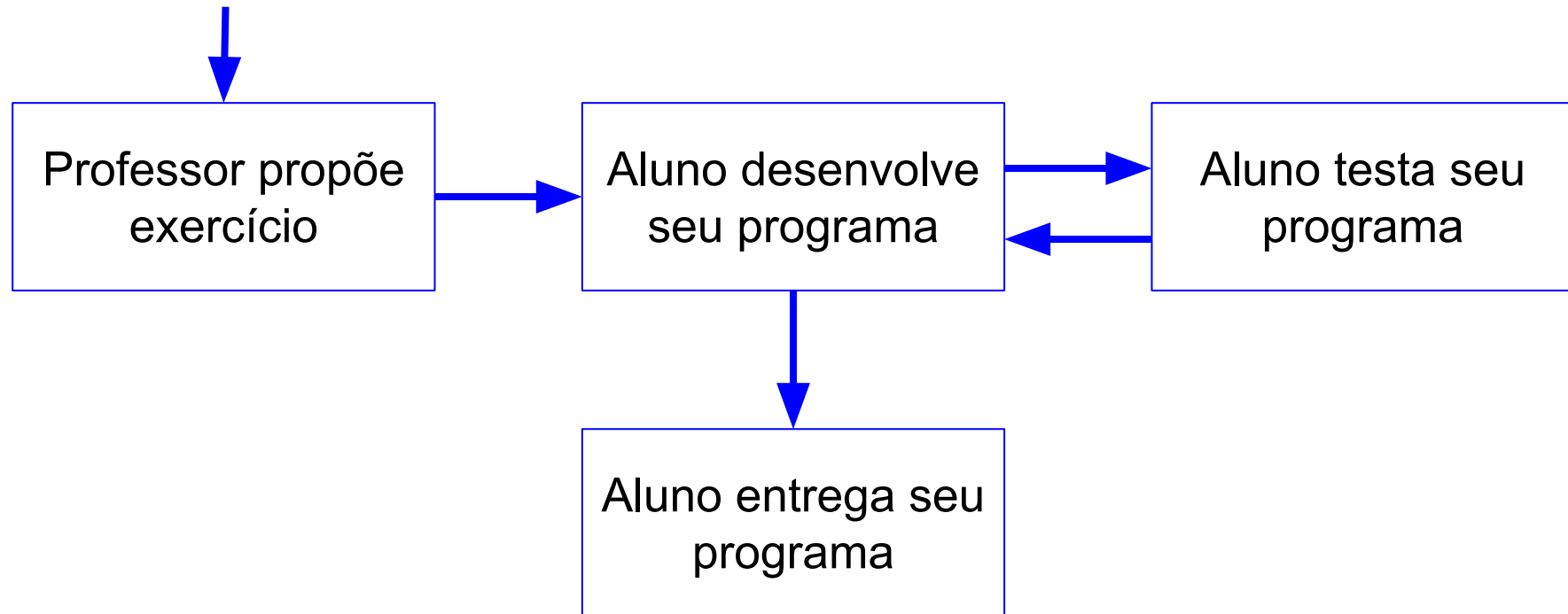


# Processo em um Exercício de Programação



**Efeito Lulu Santos:** A vida vem em ondas, como um mar / Num indo e vindo infinito

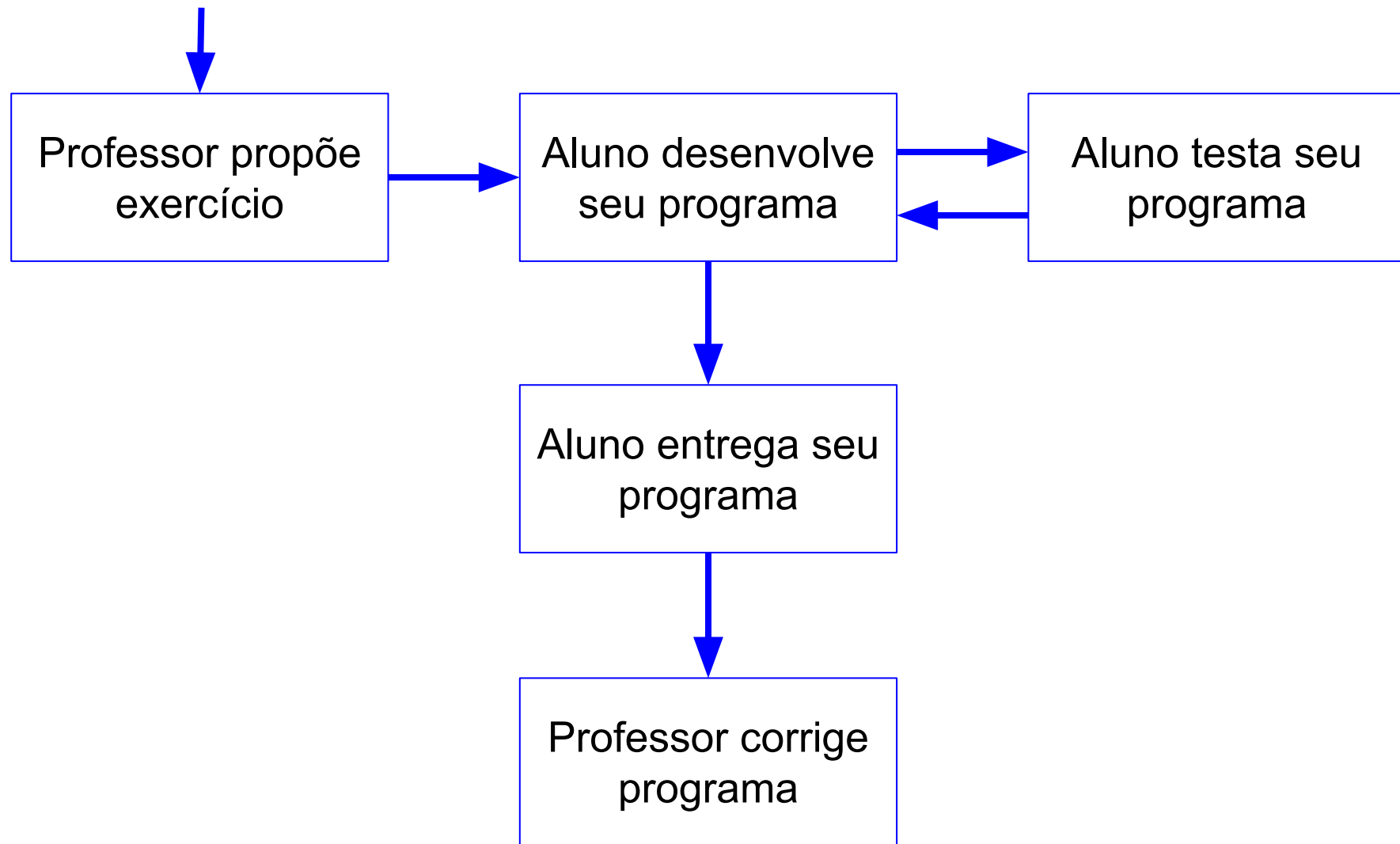
# Processo em um Exercício de Programação



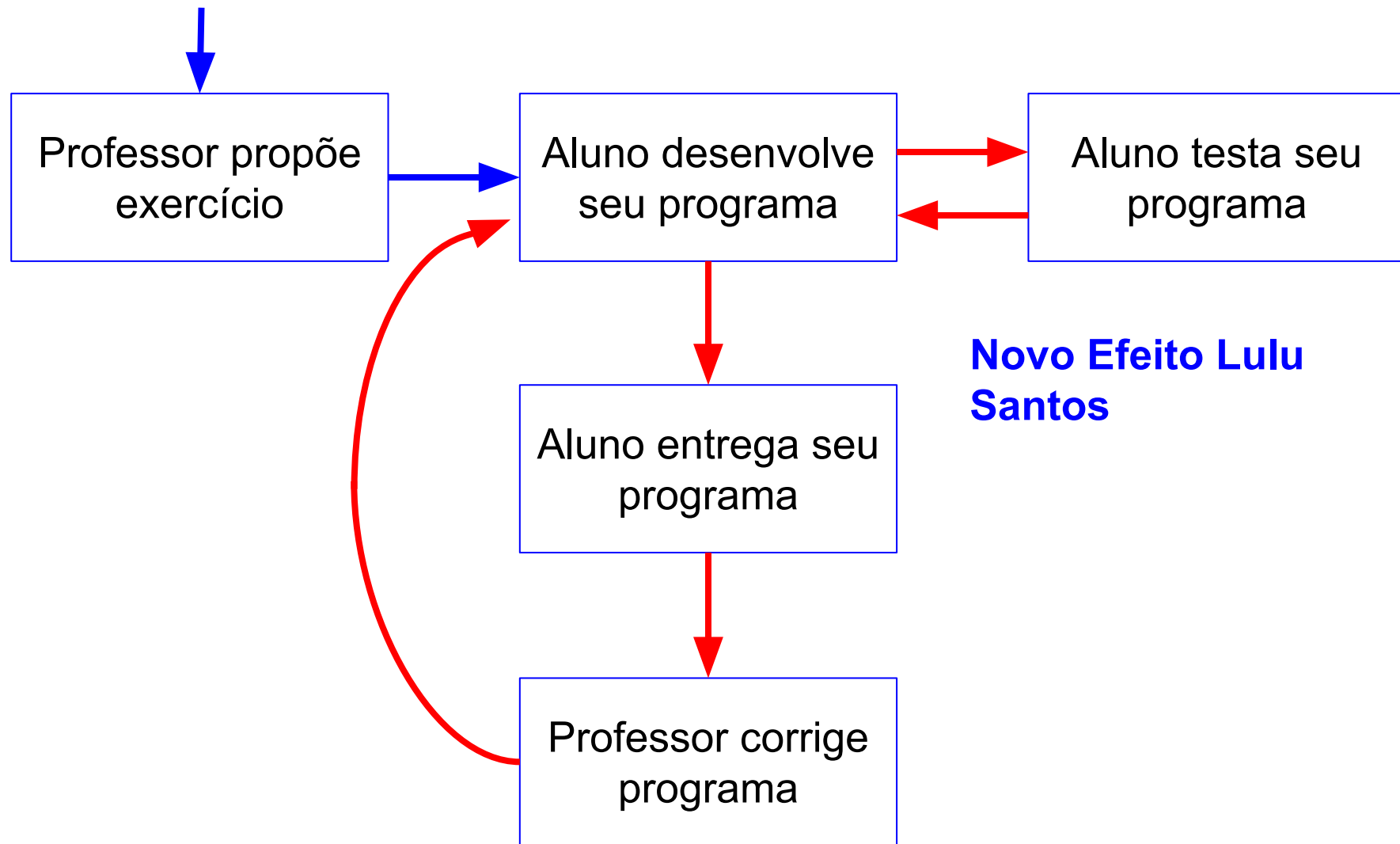
# Processo em um Exercício de Programação



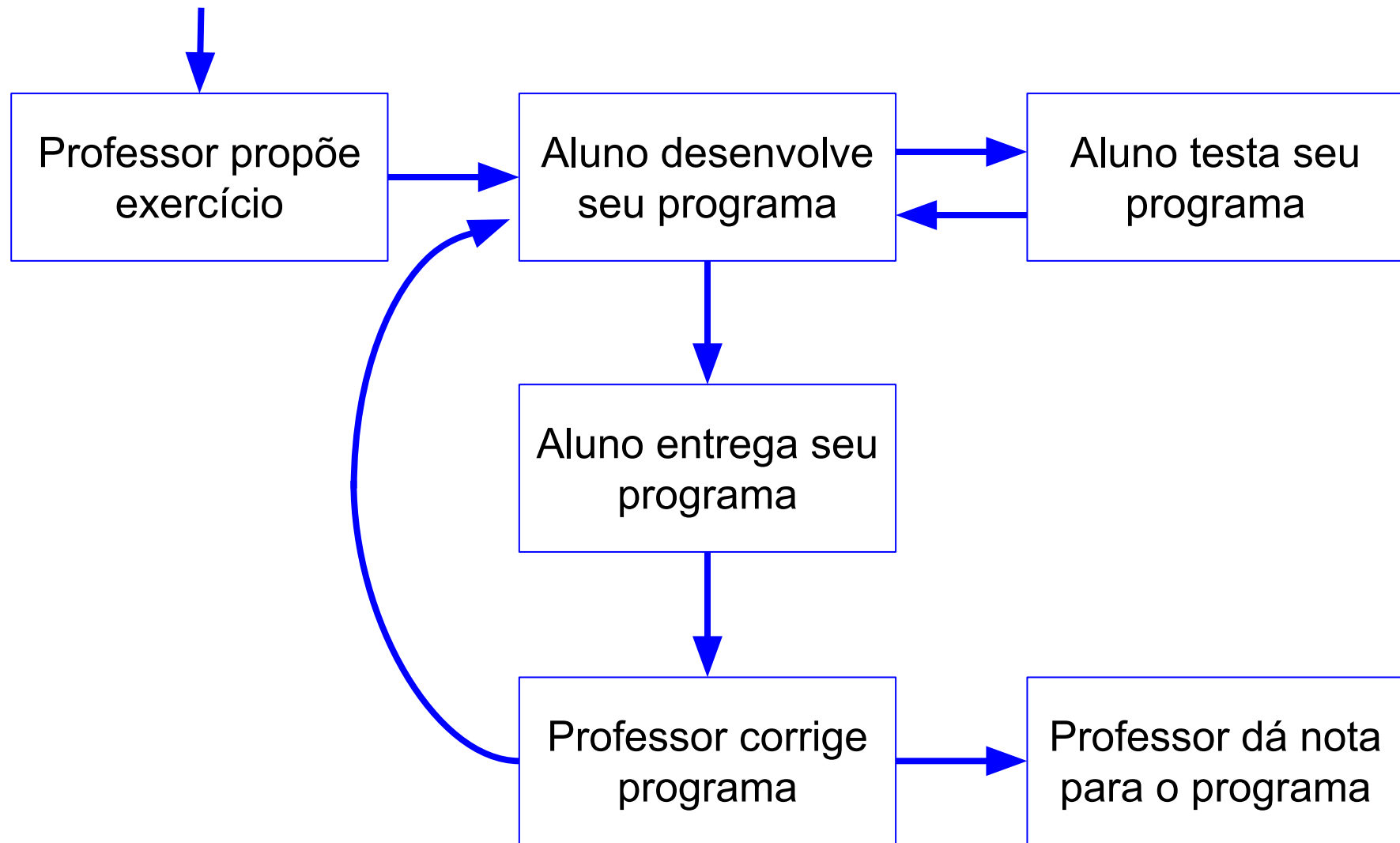
# Processo em um Exercício de Programação



# Processo em um Exercício de Programação



# Processo em um Exercício de Programação





# Alguns Desafios neste Processo

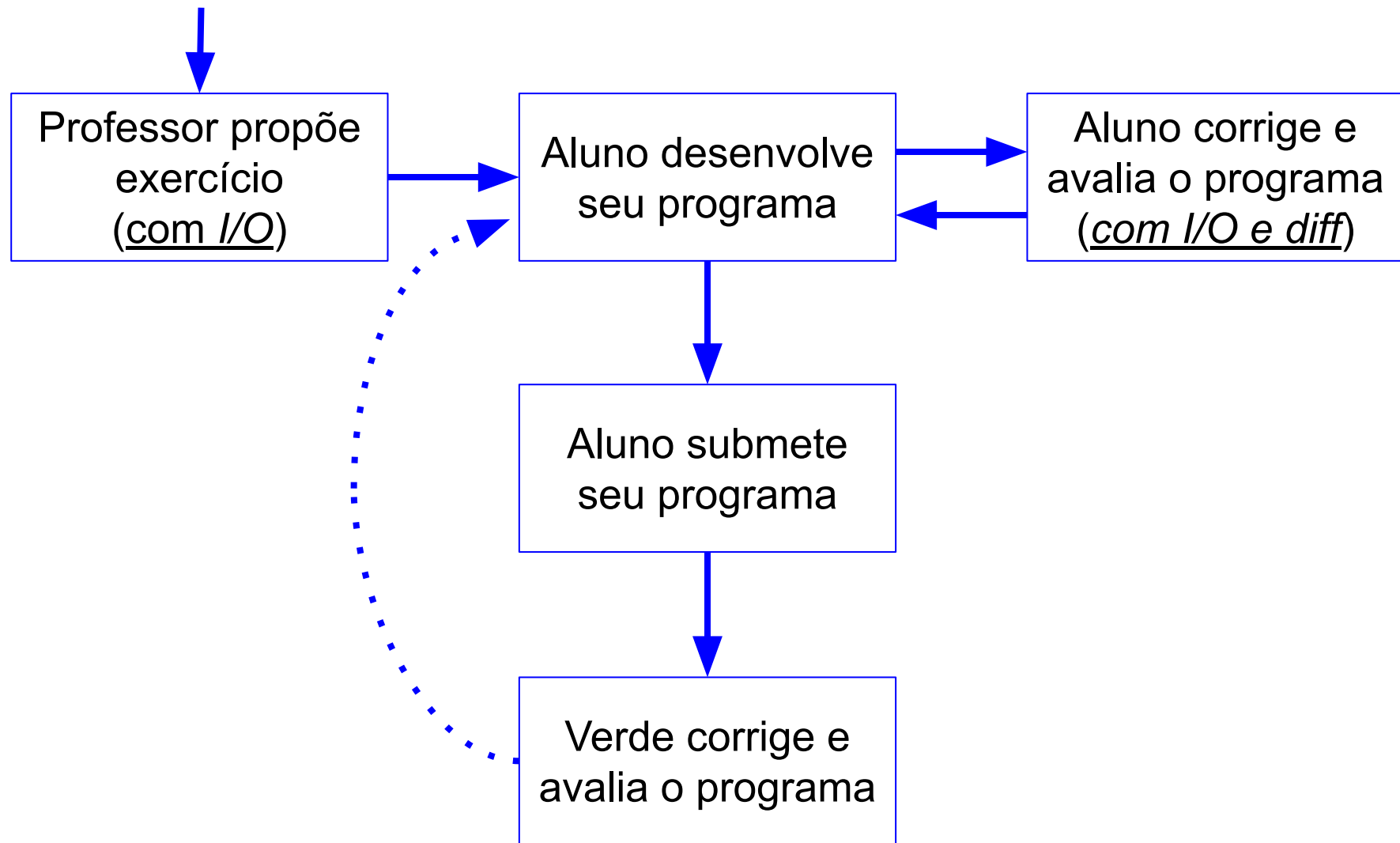
- Aluno precisa testar seus programas
- Aluno depende dos testes realizados pelo professor
- Sobrecarga / ineficiência do professor para realizar correções
- Tempo de resposta da correção não é adequado
- Escalabilidade: Número de testes vs. de alunos

- **Introdução**
  - Comparação
  - Redirecionamento
  - Plataforma
  - Exercícios
- Contextualização: *for AEDs II*
  - Questões iniciais
  - Processo em um Exercício de Programação
  - **Processo AEDs II de Exercícios**

# Objetivos

- Eliminar o tempo de resposta da correção
- Eliminar a tarefa de correção, otimizando o tempo do professor para atividades mais nobres (e.g., comentários e qualidade de código)
- Aumentar o número de testes

# Processo AEDs II de Exercícios



- Introdução
- **Comparação de Arquivos com *DIFF***
- Redirecionamento de Entrada e Saída
- Plataforma Verde
- Exercícios

# Comando *DIFF*

- Recebe dois arquivos texto e verifica se são iguais



ComputerHope.com

## Exercício Resolvido (1): Usando o *DIFF*

- (1) Leia os arquivos `matriz.c` e `matriz.cc` na pasta 01
- (2) No terminal, compare os arquivos `matriz.c` e `matriz.cc`  
comando: *diff -wB matriz.c e matriz.cc*
- (3) No terminal, faça uma cópia do `matriz.c`  
comando: `cp matriz.c matriz2.c`
- (4) No terminal, compare os arquivos `matriz.c` e `matriz2.c`

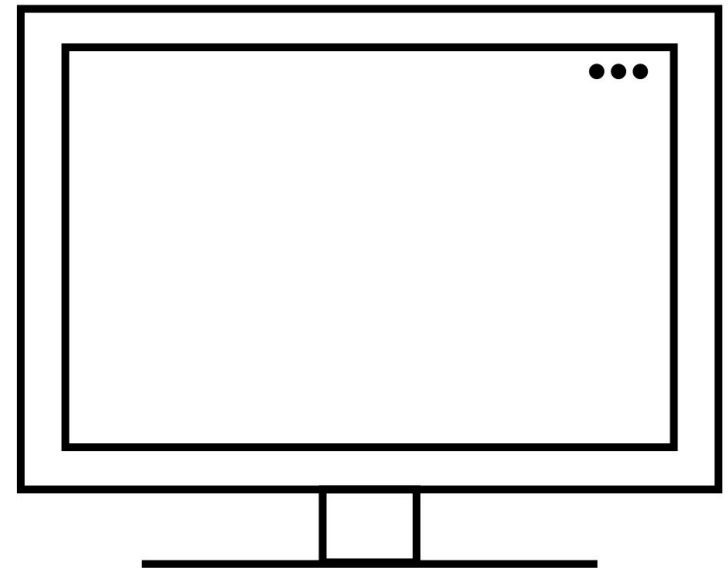
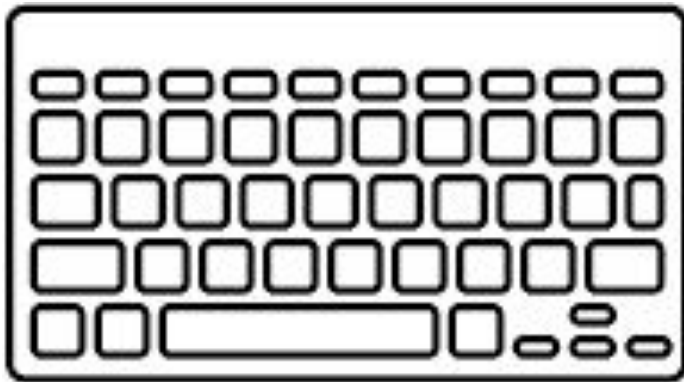
# Agenda

- Introdução
- Comparação de Arquivos com *DIFF*
- **Redirecionamento de Entrada e Saída**
- Plataforma Verde
- Exercícios





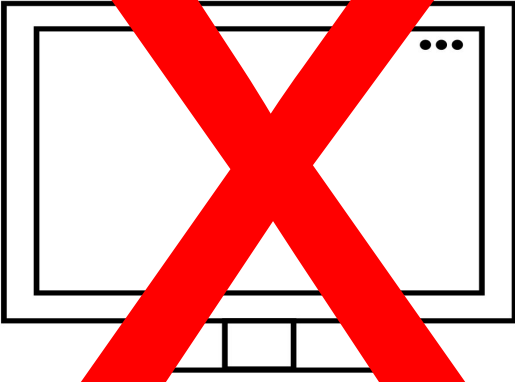

# Redirecionamento de Entrada e Saída

- Quando executamos um programa o dispositivo padrão de entrada é o teclado e o de saída, a tela

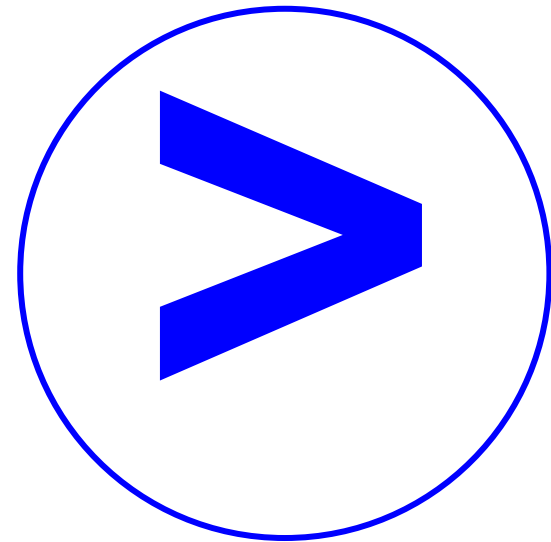
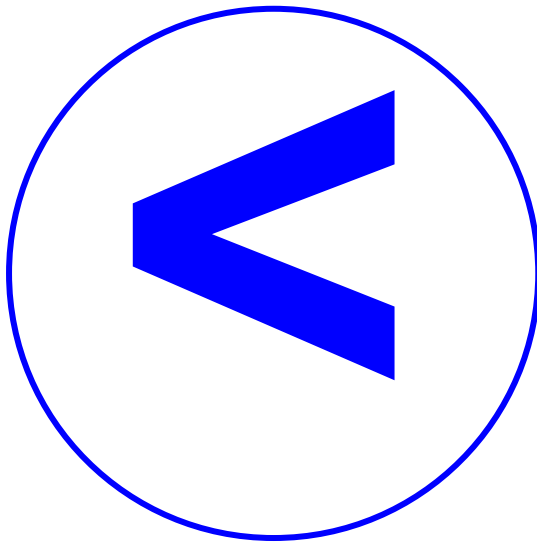


# Redirecionamento de Entrada e Saída

- Podemos modificar os dispositivos padrão de entrada e de saída para serem arquivos texto

<b>Entrada</b>	 
<b>Saída</b>	 

# Comandos para Redirecionamento de E/S



## Exercício Resolvido (2): Redirecionamento E/S

- Faça um programa em C, C++ e Java para ler e mostrar uma *string*, *int*, *double* e *char*. Execute redirecionando E/S



# Exercício Resolvido (2): Redirecionamento E/S

- Faça um programa em C, C++ e Java para ler e mostrar uma *string*, *int*, *double* e *char*. Execute redirecionando E/S



```
char str[100];
int inteiro;
double real;
char caractere;

printf("\nEntre com uma palavra: ");
scanf ("%s", str);
printf("\nEntre com um inteiro: ");
scanf ("%i", &inteiro);
printf("\nEntre com um real: ");
scanf ("%lf", &real);
scanf ("%*c"); //limpar o buffer
printf("\nEntre com um caractere: ");
scanf ("%c", &caractere);

printf("\nSua string: %s", str);
printf("\nSeu inteiro: %i", inteiro);
printf("\nSeu real: %0.1f", real);
printf("\nSeu caractere: %c\n\n", caractere);
```

# Exercício Resolvido (2): Redirecionamento E/S

- Faça um programa em C, C++ e Java para ler e mostrar uma *string*, *int*, *double* e *char*. Execute redirecionando E/S



```
char str[100];
int inteiro;
double real;
char caractere;

cout << "\nEntre com uma palavra: ";
cin >> str;
cout << "\nEntre com um inteiro: ";
cin >> inteiro;
cout << "\nEntre com um real: ";
cin >> real;
cout << "\nEntre com um caractere: ";
cin >> caractere;

cout << "\nSua string: " << str;
cout << "\nSeu inteiro: " << inteiro;
cout << "\nSeu real: " << real;
cout << "\nSeu caractere: " << caractere << "\n\n";
```



# Exercício Resolvido (2): Redirecionamento E/S

- Faça um programa em C, C++ e Java para ler e mostrar uma *string*, *int*, *double* e *char*. Execute redirecionando E/S



```
String str;  
int inteiro;  
double real;  
char caractere;  
  
System.out.println("Entre com uma palavra: ");  
str = MyIO.readString();  
System.out.println("Entre com um inteiro: ");  
inteiro = MyIO.readInt();  
System.out.println("Entre com um real: ");  
real = MyIO.readDouble();  
System.out.println("Entre com um caractere: ");  
caractere = MyIO.readChar();  
  
System.out.println("Sua string: " + str);  
System.out.println("Seu inteiro: " + inteiro);  
System.out.println("Seu real: " + real);  
System.out.println("Seu caractere: " + caractere);
```

# Exercício Resolvido (2): Redirecionamento E/S


- Compile o programa anterior

 *g++ exemplocpp.c -o exec*

 *gcc exemploc.c -o exec*

 *javac ExemploMyIO.java*

- Executar o programa

 *./exec*

 *./exec*

 *java ExemploMyIO*



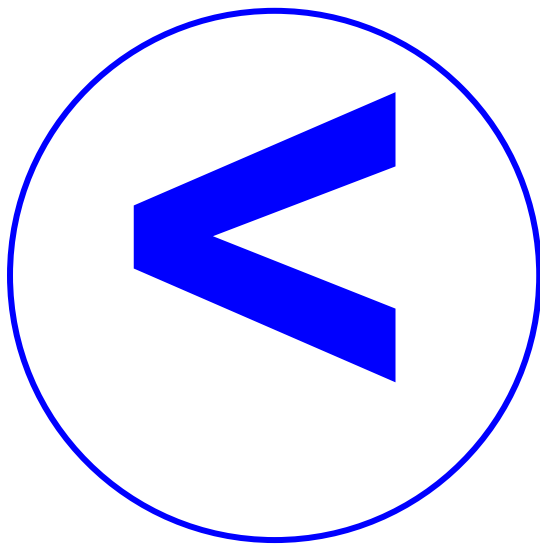
# Exercício Resolvido (2): Redirecionamento E/S

- Executar o programa redirecionando **entrada**

 `./exec < pub.in`

 `./exec < pub.in`

 `java ExemploMyIO < pub.in`



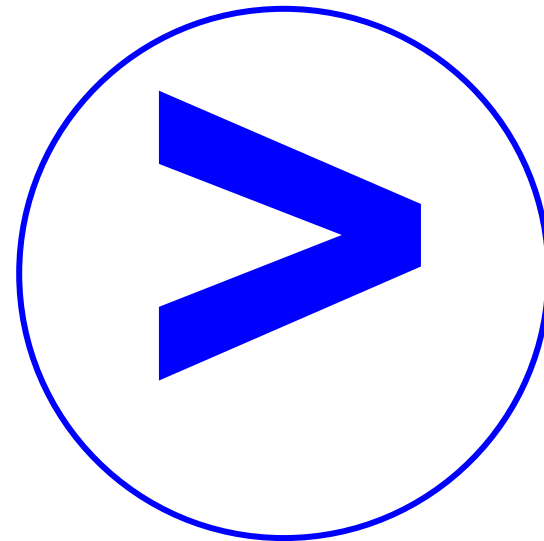
# Exercício Resolvido (2): Redirecionamento E/S

- Executar o programa redirecionando entrada e **saída**

 `./exec < pub.in > aluno.out`

 `./exec < pub.in > aluno.out`

 `java ExemploMyIO < pub.in > aluno.out`



## Exercício Resolvido (3): Matriz

- Faça um programa que leia uma matriz quadrada de inteiros e mostre na tela a soma dos elementos de cada coluna

**Entrada:** Leia um inteiro  $n < 10$  (número de linhas e colunas) e os  $n^2$  elementos da matriz

**Saída:**  $n$  números inteiros indicando as somas das colunas

Execute seu programa com a entrada pub.in e compare sua saída com a pub.out

## Exercício Resolvido (3): Matriz

- Faça um programa que leia uma matriz quadrada de inteiros e mostre na tela o maior e o menor elemento de cada coluna.
- Entrada:** Um inteiro  $n < 10$  (número de linhas e colunas) e os elementos da matriz.
- Saída:** Para cada coluna, dois inteiros indicando o menor e o maior elemento das colunas.
- Execute seu programa com a entrada `pub.in` e compare sua saída com a saída `pub.out`.

4  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16

pub.out  
28 32 36 40

## Exercício Resolvido (3): Matriz

• Faça um programa que leia uma matriz quadrada de inteiros e mostre na tela o maior elemento de cada coluna.

Entrada: **pub.in**

colunas

Saída: **pub.out**

Execute seu programa com a entrada pub.in e compare sua saída com a saída do compilador.

teiro  $n < 10$  (número de linhas e elementos da matriz)

teiros indicando o maior elemento de cada coluna

32 36 40

28

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

## Exercício Resolvido (3): Matriz

• Faça um programa que leia uma matriz quadrada de inteiros e mostre na tela o maior elemento de cada coluna.

Entrada: o primeiro inteiro  $n < 10$  (número de linhas e colunas) e os elementos da matriz.

Saída:  $n$  inteiros indicando o maior elemento de cada coluna.

Execute seu programa com a entrada pub.in e compare sua saída com a saída pub.out.

The input file 'pub.in' contains the following data:

```

4
1 2 3 4 5 6 7 8 9 10
11 12 13 14 15 16 17 18 19 20
21 22 23 24 25 26 27 28 29 30
31 32 33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60
61 62 63 64 65 66 67 68 69 70
71 72 73 74 75 76 77 78 79 80
81 82 83 84 85 86 87 88 89 90
91 92 93 94 95 96 97 98 99 100

```

The output file 'pub.out' contains the following data:

```

32 36 40

```

## Exercício Resolvido (3): Matriz

• Faça um programa que leia uma matriz quadrada de inteiros e mostre na tela o maior e o menor elemento de cada coluna.

Entrada: **3** (número  $n < 10$  de linhas e colunas)

Saída: **7** (maior elemento da primeira coluna), **36** (menor elemento da terceira coluna), **15** (menor elemento da quinta coluna)

Execute seu programa com a entrada pub.in e compare sua saída com a saída do programa de referência.

## Exercício Resolvido (3): Matriz

• Faça um programa que leia uma matriz quadrada de inteiros e mostre na tela o maior elemento de cada coluna.

**Entrada:** Primeiro inteiro  $n < 10$  (número de linhas e colunas da matriz).

**Saída:** n inteiros indicando o maior elemento de cada coluna.

Execute seu programa com a entrada `pub.in` e compare sua saída com a saída `pub.out`.

**pub.in**

4  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16

**pub.out**

28 32 36 40



## Exercício Resolvido (4): Letras Maiúsculas

- Faça um programa que leia várias frases e mostre o número de letras maiúsculas de cada frase

**Entrada:** composta por várias linhas sendo que a última contém a palavra FIM

**Saída:** contém um inteiro para cada linha da entrada

Execute seu programa com a entrada pub.in e compare sua saída com a pub.out

# Exercício Resolvido (4): Letras Maiúsculas

- Faça um programa que leia várias frases e mostre o número de letras maiúsculas em cada uma.

Algoritmos e Estruturas de Dados I	4
Computadores e Sociedade	2
Geometria Analítica	2
Introdução à Ciência da Computação	3
Lab. de Programação Visual	3
Seminários I	2
Algoritmos e Estruturas de Dados II	5
Arquitetura de Computadores I	3
Cálculo I	2
Cultura Religiosa: Fenômeno Religioso (Cultura I)	6
Matemática Discreta	2
Seminários II	3
FIM	

# Agenda

- Introdução
- Comparação de Arquivos com *DIFF*
- Redirecionamento de Entrada e Saída
- **Plataforma Verde**
- Exercícios

# Plataforma Verde



# Plataforma Verde

- **Cadastro de exercício:** nome, pub.in, pub.out, pri.in, pri.out e linguagem



## Cadastrar Novo Problema

**Nome do Problema:**

Entrada Publica:  Nenhum arquivo selecionado

Saida Publica:  Nenhum arquivo selecionado

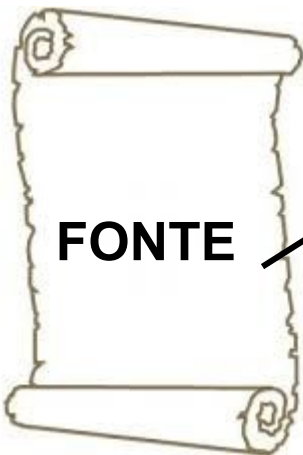
Entrada Privada:  Nenhum arquivo selecionado

Saida Privada:  Nenhum arquivo selecionado

☒ Java  
☐ C++

# Plataforma Verde

- Submissão de exercício

A screenshot of the Verde platform submission interface. The header is teal with the 'Verde' logo, a dropdown menu showing 'AEDS II/2020/1', and a download icon. Below the header are three tabs: 'Submissões', 'Turmas', and 'Comentários'. The main content area is titled 'Nova Submissão' and contains a dropdown menu with 'TP03Q01 - Lista com Alocação'. Below this is a button labeled 'Escolher arquivo' and the text 'Nenhum arquivo selecionado'. An arrow points from the 'FONTE' scroll to the 'Escolher arquivo' button.

**Atenção:** Arquivos em C/C++ devem ter a extensão cpp

# Plataforma Verde

- Verde efetua a correção



- (1) Compilar
- (2) Executar redirecionando entrada (pub.in) e saída (aluno.out)
- (3) Comparar os arquivos aluno.out com (pub.out)
- (4) Executar redirecionando entrada (pri.in) e saída (aluno.out)
- (5) Comparar os arquivos aluno.out com (pri.out)

# Plataforma Verde

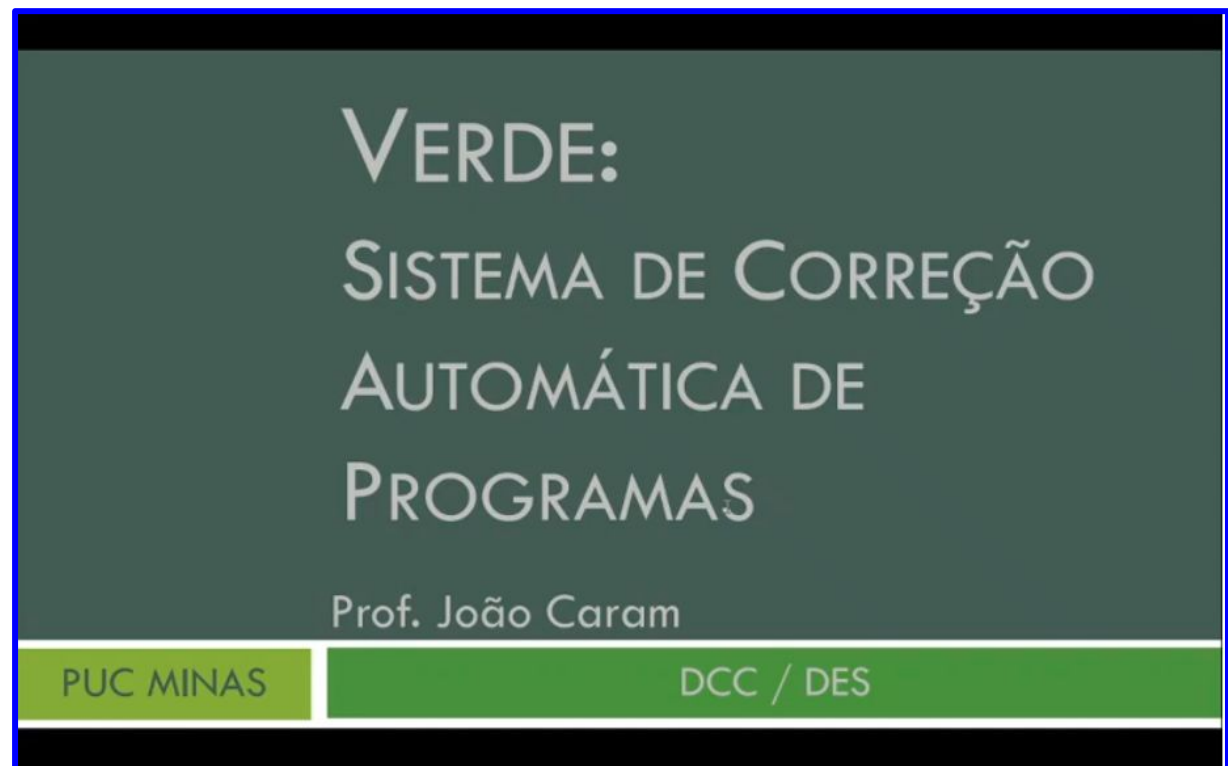
- Submetidos

Arquivos Submetidos								
#	Problema	Data	Arquivo	Publico	Privado	Detalhes	Fonte	Remover



# Verde - Treinamento Alunos

- Veja os vídeos do Prof. João Caram
- <https://www.youtube.com/playlist?list=PLD8tg9ADnIVjk0Y15Lm-rINrVUTq9IVxr>



## Cadastro no Verde

- Todos os alunos de IC (CC/ES) estão cadastrados na turma de IC do Verde
- maratona.crc.pucminas.br (login *matrícula* e senha 1234)
- Entre no Verde agora e ganhe seus primeiros totais!!!

# Agenda

- Introdução
- Comparação de Arquivos com *DIFF*
- Redirecionamento de Entrada e Saída
- Plataforma Verde
- **Exercícios**

# Submeta no Verde

- 1) **Matriz em Java**
- 2) **Matriz em C/C++** (extensão .cpp)
- 3) **Letras Maiúsculas em Java**
- 4) **Letras Maiúsculas em C/C++** (extensão .cpp)

5) **Palíndromo em Java** - Faça um programa recebe várias frases e identifica se elas são palíndromos

**Entrada:** composta por várias linhas sendo que a última contém a palavra FIM

**Saída:** para cada linha de entrada, escreva SIM / NÃO, indicando se a linha é um palíndromo

6) **Palíndromo em C/C++** - Refaça a questão anterior

**7) Dona Mônica em Java -** Dona Mônica é mãe de três filhos que têm idades diferentes. Ela notou que, neste ano, a soma das idades dos seus três filhos é igual à idade dela. Neste problema, dada a idade de dona Mônica e as idades de dois dos filhos, seu programa deve computar e imprimir a idade do filho mais velho. Por exemplo, se sabemos que dona Mônica tem 52 anos e as idades conhecidas de dois dos filhos são 14 e 18 anos, então a idade do outro filho, que não era conhecida, tem que ser 20 anos, pois a soma das três idades tem que ser 52. Portanto, a idade do filho mais velho é 20. Em mais um exemplo, se dona Mônica tem 47 anos e as idades de dois dos filhos são 21 e 9 anos, então o outro filho tem que ter 17 anos e, portanto, a idade do filho mais velho é 21.

**Entrada:** composta por várias linhas, em cada uma temos 3 inteiros sendo a idade de Dona Mônica e dos seus dois filhos mais novos. A última linha contém apenas um zero

**Saída:** para cada linha de entrada, escreva um número inteiro indicando a idade do filho mais velho

**8) Dona Mônica em C/C++**

9) **Cometa em Java** - O cometa Halley é um dos cometas de menor período do Sistema Solar, completando uma volta em torno do Sol a cada 76 anos. Na última ocasião em que ele tornou-se visível do planeta Terra, em 1986, várias agências espaciais enviaram sondas para coletar amostras de sua cauda e assim confirmar teorias sobre suas composições químicas.

Escreva um programa que, dado o ano atual, determina qual o próximo ano em que o cometa Halley será visível novamente do planeta Terra. Se o ano atual é um ano de passagem do cometa, considere que ele já passou neste ano, logo, nesse caso, considere sempre o próximo ano de passagem, não considerando o ano atual.

**Entrada:** composta por várias linhas, em cada uma temos um ano entre 2020 e 3000. A última linha contém apenas um zero

**Saída:** para cada linha de entrada, escreva um número inteiro indicando o ano da próxima passagem

10) **Cometa em C/C++**