ISPGAYA
instituto superior politécnico

Apresentação da Unidade Curricular

Algoritmos e Modelos de Programação

1º ano / 1º semestre

22,5 horas teórico-práticas - TP (1,5h semanal)

22,5 horas horas práticas laboratoriais - PL (1,5h semanal)

Engenharia Informática (turma diurna e turma pós-laboral)

Professor: Jorge Simões (jsimoes@ispgaya.pt)

Conteúdos Inforestudante: https://inforestudante.ispgaya.pt

Apresentação da Unidade Curricular

Horas	Segunda	Terça	Quarta
14:30 - 15:00	Análise Matemática I [(JBP)] [4.1 ST]	Algoritmos e Modelos de Programação [(JMS)] [5.3 ST/Lab.Inf]	Instrumentação e Sistemas Digitais [(JML)] [5.3 ST/Lab.Inf; 5.4 ST/Lab.Inf]
15:00 - 15:30			
15:30 - 16:00			
16:00 - 16:30		Algoritmos e Modelos de Programação [(JMS)] [5.3 ST/Lab.Inf]	
16:30 - 17:00			
17:00 - 17:30			
17:30 - 18:00			
18:00 - 18:30	Análise Matemática I [(MFM)] [5.1 ST]	Instrumentação e Sistemas Digitais [(JML)] [4.3 ST/Lab.Inf]	Algoritmos e Modelos de Programação [(JMS)] [5.4 ST/Lab.Inf]
18:30 - 19:00			
19:00 - 19:30			
19:30 - 20:00			
20:00 - 20:30	Algoritmos e Modelos de Programação [(JMS)] [5.3 ST/Lab.Inf]	Instrumentação e Sistemas Digitais [(SRT)] [1.1/1.2 Lab.Electro]	Análise Matemática I [(MFM)] [5.1 ST]
20:30 - 21:00			
21:00 - 21:30			
21:30 - 22:00			
22:00 - 22:30			
22:30 - 23:00			

turma diurna

turma pós-laboral

Aula TP

Aula PL

Apresentação da Unidade Curricular

OBJETIVOS

No final da unidade curricular os alunos deverão estar aptos a desenvolver um raciocínio lógico, em termos de conceitos e técnicas de programação, de forma a serem capazes de conceber **algoritmos** para a resolução de problemas práticos de pequena e média complexidade. Os alunos deverão desenvolver as capacidades necessárias para a aplicação dos algoritmos concebidos em programas de computador concretos.

CONHECIMENTOS DE BASE

Conhecimentos de informática na ótica de utilizador. Conhecimentos de Matemática.

Apresentação da Unidade Curricular

PROGRAMA (Resumo)

- Introdução à Algoritmia
- Conceitos de Algoritmia e Programação (fluxogramas, linguagem algorítmica)
- Introdução à Programação com a linguagem Python

Apresentação da Unidade Curricular

BIBLIOGRAFIA ACONSELHADA

- Portela, F., Pereira, T. (2023). Introdução à Algoritmia e Programação com Python, FCA, ISBN: 978-972-722-931-4
- Sobral, S. (2024). Introdução à Programação usando Python, 2ª edição, Edições Sílabo, ISBN: 978-989-561-387-8
- Sobral, S. (2024). Introdução à Programação usando Python – Exercícios Resolvidos, Edições Sílabo, ISBN: 978-989-561-386-1
- Simões, J., Santos, M. (1996), Introdução à Programação, Ed. Catepse, ISBN: 972-97083-0-4, Cota: 004.43/SIMj/INT ISPGaya
- Carvalho, A. (2021). Práticas de Python: Algoritmia e Programação, FCA, 978-972-722-918-5
- Vasconcelos, J. (2015). Python: Algoritmia e Programação Web, FCA, ISBN: 978-972-722-813-3
- Costa, E. (2015). Programação em Python: Fundamentos e Resolução de Problemas, FCA, ISBN: 978-972-722-816-4

Apresentação da Unidade Curricular

LINKS ÚTEIS

- https://www.geeksforgeeks.org/dsa/introduction-to-algorithms/
- https://www.w3schools.com/python/
- https://docs.python.org/3/tutorial/index.html
- https://www.codecademy.com/learn/learn-python-3
- https://anandology.com/python-practice-book/
- outros ...

Apresentação da Unidade Curricular

AVALIAÇÃO POR FREQUÊNCIA

- Trabalhos Práticos (TP1 e TP2), a realizar nas aulas práticas laboratoriais, com pesos de 20% e 30%, respetivamente.
- Prova Escrita (PE), com um peso de 40%.
- Assiduidade e participação (AP), com um peso de 10%.
- Classificação Final = 20% TP1 + 30% TP2 + 40% PE + 10% AP

Apresentação da Unidade Curricular

AVALIAÇÃO POR EXAME FINAL

- Trabalhos Práticos (TP1 e TP2), com pesos de 20% e 25%.
- Prova Escrita (PE), com um peso de 50%.
- Classificação Final = 20% TP1 + 30% TP2 + 50% PE
- **Cada aluno deve optar pela avaliação for frequência ou pela avaliação por exame, devendo comunicar a decisão ao regente da unidade curricular até ao fim da primeira semana de aulas. Na ausência de comunicação será entendido que o aluno optou pela avaliação por frequência.**

Apresentação da Unidade Curricular

AVALIAÇÃO EM RECURSO E ÉPOCAS ESPECIAIS

- Trabalhos Práticos (TP1 e TP2), com pesos de 20% e 30%.
- Prova Escrita (PE), com um peso de 50%.
- Classificação Final = 20% TP1 + 30% TP2 + 50% PE
- **Em todas as avaliações por exame, poderão ser considerados os trabalhos práticos da época de avaliação por frequência caso a classificação obtida nesses trabalhos tenha sido superior a 9,5 valores.**

Apresentação da Unidade Curricular

INSTRUMENTOS DE AVALIAÇÃO

- A Prova Escrita (PE), os Trabalhos Práticos (TP1 e TP2) e a componente de assiduidade e participação (AP) constituem os instrumentos de avaliação da unidade curricular.

- Na Prova Escrita e nos Trabalhos Práticos a **nota mínima é de 7,5** valores, para efeitos de cálculo da classificação final.

Regulamento de Avaliação de Conhecimentos e Competências
(disponível no Inforestudante)

Apresentação da Unidade Curricular

AVALIAÇÃO (datas previstas)

- **Trabalhos Práticos** (aula prática-laboratorial)

  Trabalho Prático 1:

  - Turma diurna: 11 de novembro de 2025
  - Turma pós-laboral: 12 de novembro de 2025

  Trabalho Prático 2:

  - Turma diurna: 13 de janeiro de 2026
  - Turma pós-laboral: 14 de janeiro de 2026

- **Prova Escrita** (teste de escolha múltipla feito na plataforma Moodle)

  A marcar pela Direção da Escola
  (meio de janeiro / início de fevereiro de 2026)

Apresentação da Unidade Curricular

FUNCIONAMENTO DAS AULAS

- Aulas Teórico-Práticas (TP) e Aulas Práticas Laboratoriais (PL): matéria teórica e exercícios

- Em todas as aulas: registo de presenças

- Os alunos podem resolver os trabalhos práticos usando os computadores do ISPGaya ou usando os seus computadores portáteis.

Apresentação da Unidade Curricular

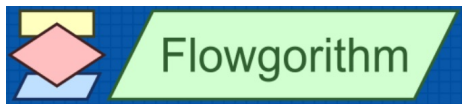
MATERIAL DE APOIO (a disponibilizar no Inforestudante)

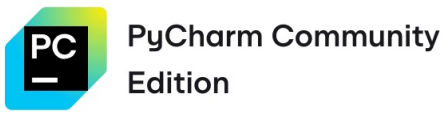
- **Slides das Aulas**
- **Fichas de Exercícios**
- **Livros e links indicados na bibliografia**
- **Software / plataformas de desenvolvimento**

Apresentação da Unidade Curricular

SOFTWARE

- Flowgorithm
  - Windows: http://www.flowgorithm.org
  - MacOS: https://github.com/jostasik/Flowgorithm-macOS
- Portugol Studio (Versão web: https://portugol-webstudio.cubos.io/)
- Python (https://www.python.org/downloads/)
- Pycharm (https://www.jetbrains.com/pycharm/)
- Plataformas low-code

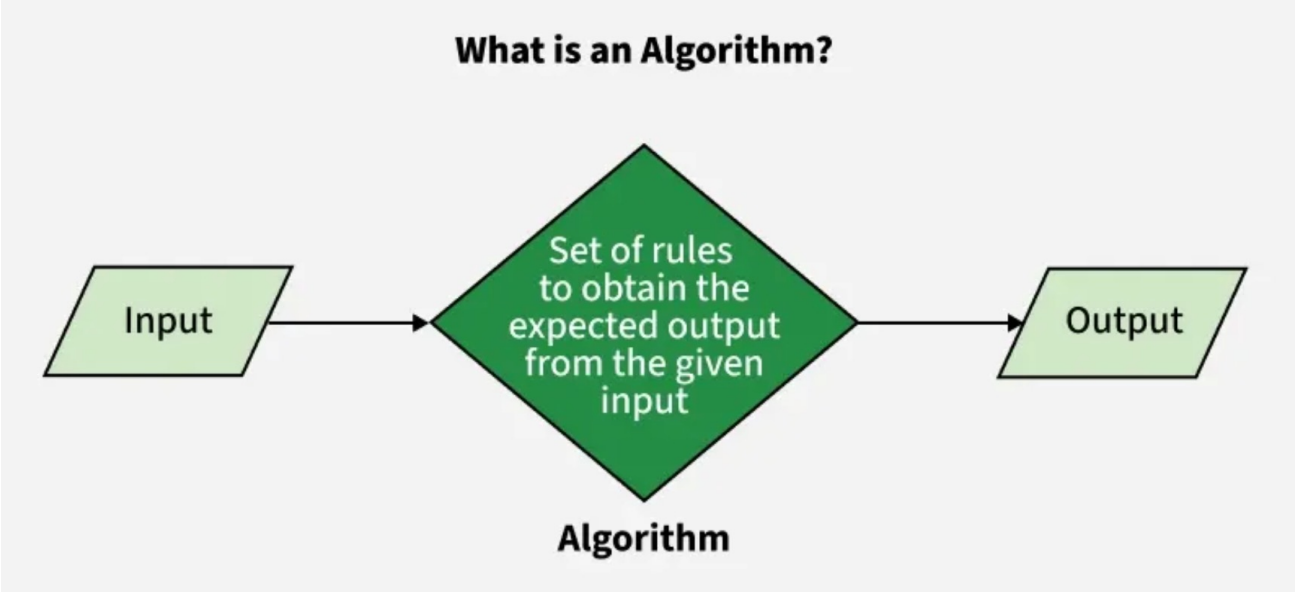




Algoritmos e Modelos de Programação ?

Algoritmo ?

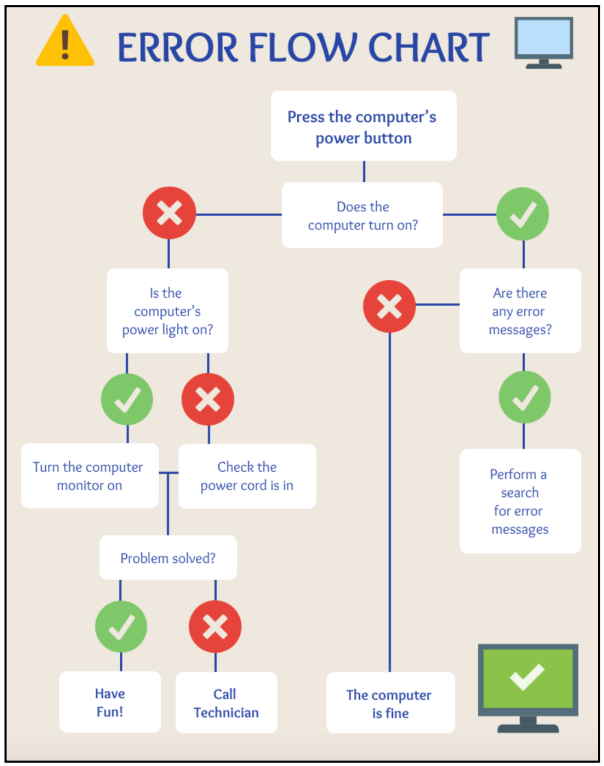
Conjunto finito de instruções, bem definidas e não ambíguas que, ao serem executadas numa determinada ordem, resolvem um problema.

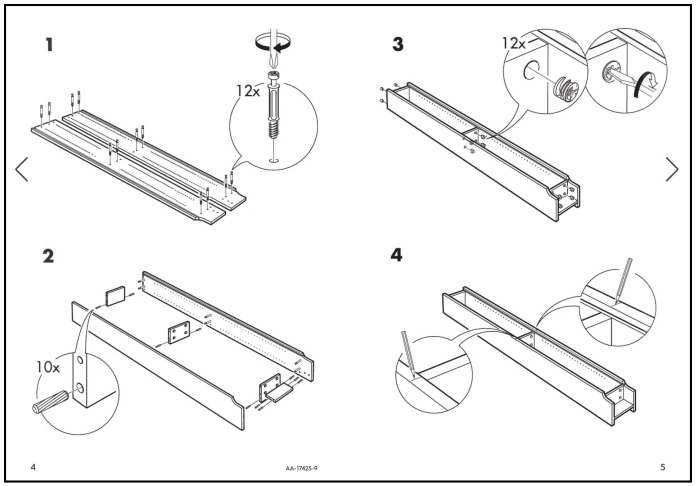


https://www.geeksforgeeks.org/dsa/introduction-to-algorithms/

Apresentação da Unidade Curricular

Exemplos de algoritmos?





QUICK TOMATO MOLD
(Pictured below)

For an unusual appetizer, salad, or relish—a tangy tomato juice mixture that's molded in the juice can.

1 can (1 pt. 2 oz.) tomato juice
1 package (3 oz.) Jell-O Lemon or Mixed Fruit Gelatin
1½ tablespoons lemon juice or vinegar
½ teaspoon salt
Dash of pepper

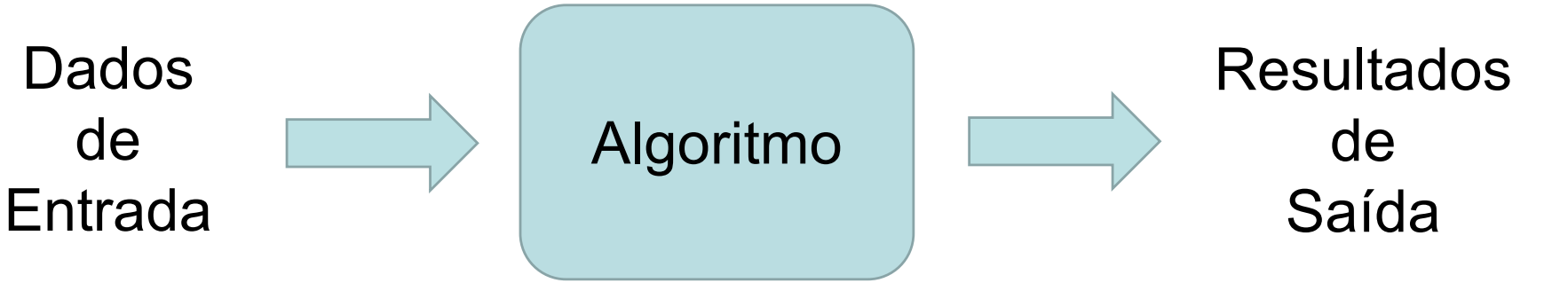
Bring 1 cup tomato juice to a boil. Stir in Jell-O Gelatin until dissolved. Then pour into juice can, blending with remaining juice. Add seasonings. Chill until firm. To unmold, puncture bottom of can before dipping in warm water -see Tips on Unmolding Jell-O Gelatin on page 82. Slice or serve from a relish dish. Makes 2¾ cups, or 4 to 6 side salads or 8 to 10 relish servings.

Apresentação da Unidade Curricular

Um algoritmo deve ser:

- **Genérico**: deve considerar todos os casos possíveis do problema;
- **Determinístico**: deve ter um comportamento bem definido, não aleatório e sem ambiguidades (para a mesma entrada deve apresentar sempre a mesma saída);
- **Finito**: deve atingir uma solução em tempo finito (em tempo útil) e de forma eficiente;
- **Eficaz**: deve ser capaz de resolver o problema, sem falhas.
- **Capaz de comunicar**: deve ter capacidade de comunicação, recebendo dados (entradas – 0 ou mais) e apresentando os resultados que obtém (saídas – pelo menos uma).

Apresentação da Unidade Curricular



0 ou mais dados de entrada

os algoritmos são independentes das linguagens de programação

pelo menos um resultado de saída

Apresentação da Unidade Curricular

Escrever algoritmos para resolver os problemas seguintes:

- Mudar um pneu de um automóvel;
- Substituir uma lâmpada fundida de um candeeiro de teto;
- Calcular o máximo divisor comum de dois números inteiros;
- **Verificar se um número inteiro é par ou ímpar;**
- **Verificar se um número inteiro é positivo, negativo ou zero;**
- Verificar se um número inteiro é um número primo;
- Obter o produto de dois números inteiros recorrendo apenas à operação adição;
- **Ler uma sequência de números inteiros que termina com o número 0 e contar quantos dos números lidos são positivos e quantos são negativos**.

Apresentação da Unidade Curricular

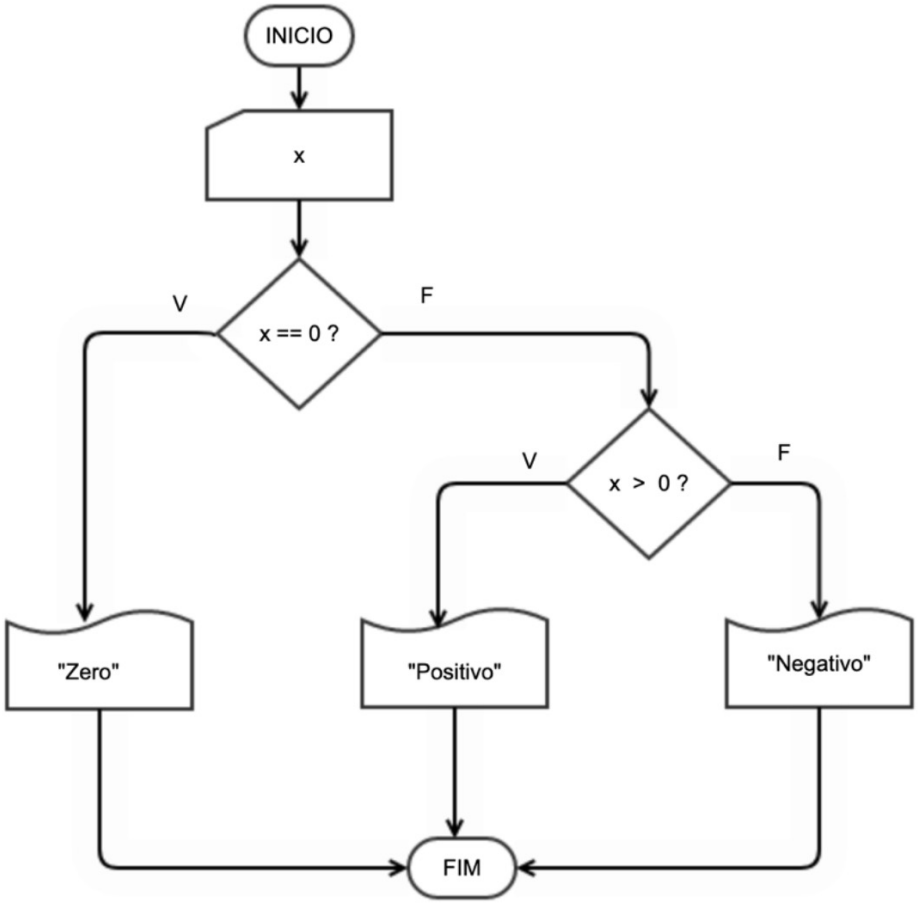
Verificar se um número inteiro é par ou ímpar (em linguagem corrente)

1. Ler um número inteiro
2. Dividir o número lido por 2 e guardar o resto dessa divisão
3. Se o resto obtido for igual a zero, então escrever "o número é par"
4. Senão, escrever "o número é ímpar"

(e se o número lido for 0 ?)

Apresentação da Unidade Curricular

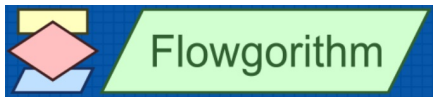
Verificar se um número inteiro é positivo, negativo ou zero (em fluxograma)

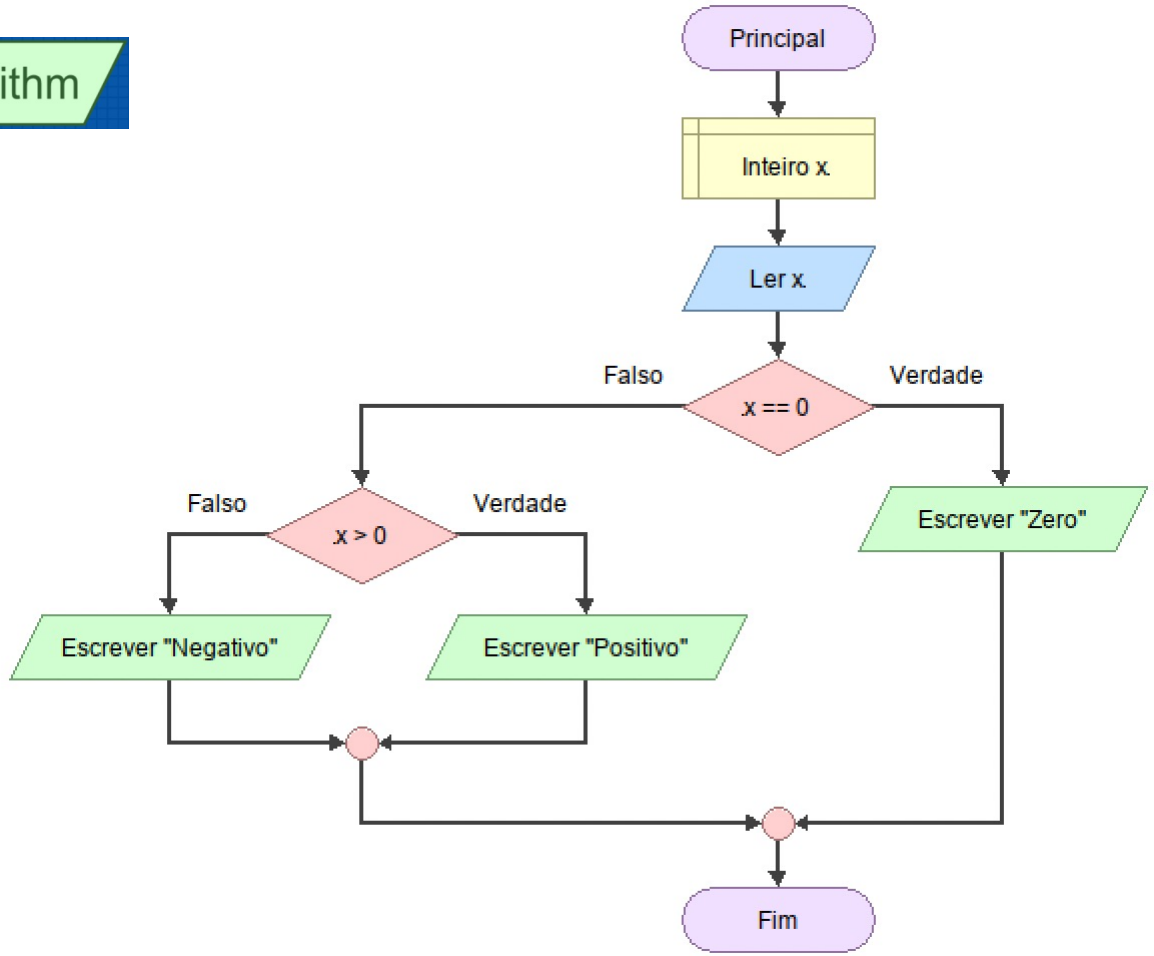


- A aplicação **Flowgorithm** permite desenhar fluxogramas (mas com uma notação diferente).
- O **Flowgorithm** permite "executar" o fluxograma e converte-o em instruções numa linguagem de programação à escolha.

Apresentação da Unidade Curricular

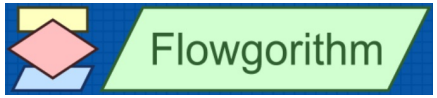
Verificar se um número inteiro é positivo, negativo ou zero (em fluxograma)





Apresentação da Unidade Curricular

Verificar se um número inteiro é positivo, negativo ou zero



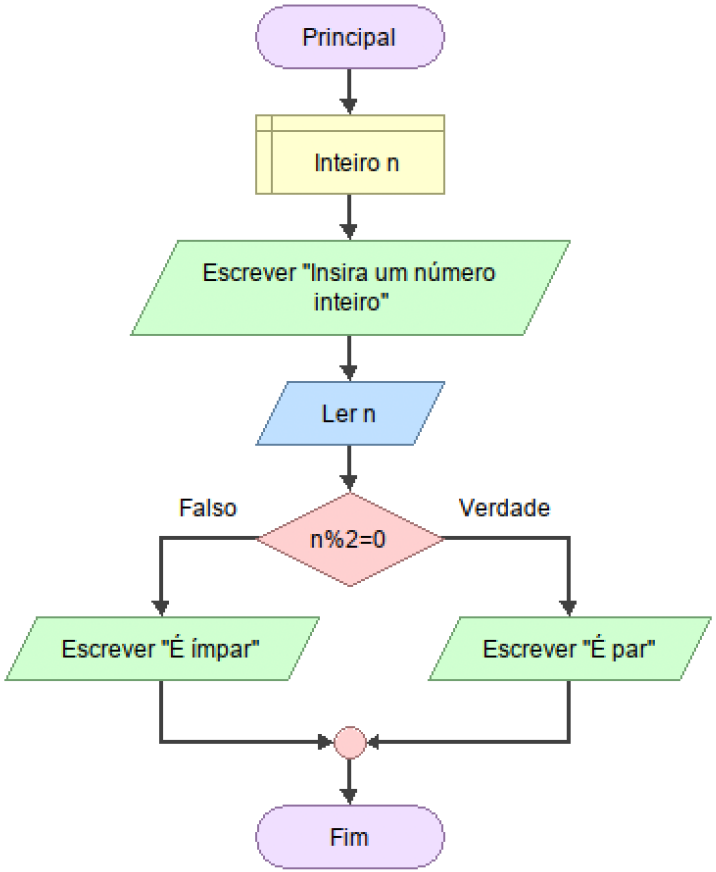

```
x = int(input())
if x == 0:
    print("Zero")
else:
    if x > 0:
        print("Positivo")
    else:
        print("Negativo")
```


```
public static void main(String[] args) {
    int x;

    x = input.nextInt();
    if (x == 0) {
        System.out.println("Zero");
    } else {
        if (x > 0) {
            System.out.println("Positivo");
        } else {
            System.out.println("Negativo");
        }
    }
}
```

Apresentação da Unidade Curricular

Verificar se um número inteiro é par ou ímpar



Flowgorithm

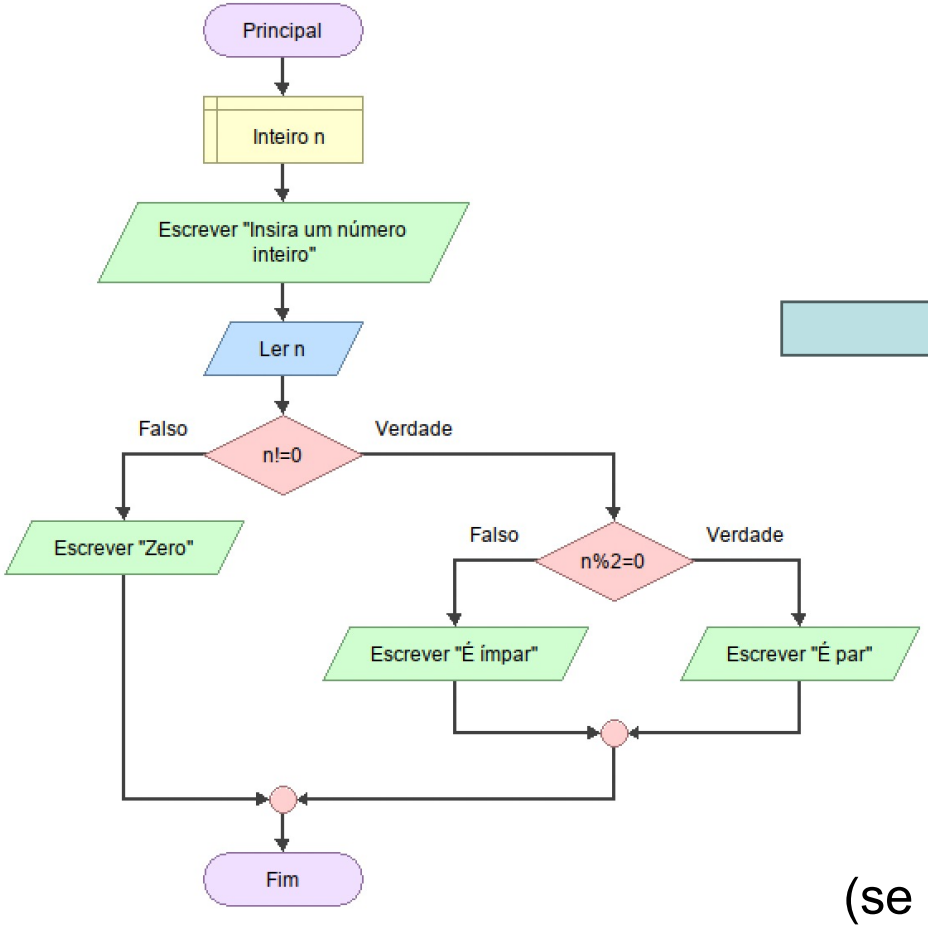
```
print("Insira um número inteiro")
n = int(input())
if n % 2 == 0:
    print("É par")
else:
    print("É ímpar")
```

python

(e se o número lido for 0 ?)

Apresentação da Unidade Curricular

Verificar se um número inteiro é par ou ímpar



Flowgorithm

```
print("Insira um número inteiro")
n = int(input())
if n != 0:
    if n % 2 == 0:
        print("É par")
    else:
        print("É ímpar")
else:
    print("Zero")
```

python

(se o número lido for 0 escreve ”Zero”)

Apresentação da Unidade Curricular

Ler uma sequência de números inteiros que termina com o número 0 e contar quantos dos números lidos são positivos e quantos são negativos.

1. Início
2. Iniciar a contagem de positivos em 0 e a contagem de negativos em 0.
3. Ler um número inteiro.
4. Se o número for igual a 0 vai para o passo 8.
5. Senão, se o número for maior que 0 escreva "positivo" e aumenta 1 à contagem de positivos.
6. Senão, se o número for menor que 0 escreva "negativo" aumenta 1 à contagem de negativos.
7. Volta para o passo 3.
8. Escreve a contagem de positivos e a contagem de negativos.
9. Fim

Apresentação da Unidade Curricular

Problema

De um conjunto de 25 moedas de ouro, sabe-se que uma é falsa e pesa menos do que as restantes. Existe disponível uma balança de pratos que, no entanto, apenas pode efetuar um máximo de 3 pesagens. Descubra um algoritmo que permita encontrar a moeda falsa usando a balança de pratos.