



Pesquisa sobre o Algoritmo do **AVESTRUZ**.

Em, ciência da Computação, o **algoritmo do avestruz** é uma estratégia de ignorar problemas potenciais com base no fato de que eles podem ser extremamente raros — "enfie a cabeça na areia e finja que não há nenhum problema". Isso pressupõe é mais rentável permitir que o problema ocorra do que tentar a sua prevenção.

Esta abordagem pode ser usada no tratamento de impasses(*deadlocks*) em programação concorrente se os impasses são considerados muito raros, e se o custo de detecção ou prevenção é alto.

Ex Isso é um exemplo de Código Ignorando uma DeadLock entre o Recurso1 e um Recurso 2 que acaba activando O thread 3 que finalmente Finalisa o Algoritmos,

```
final static String RECURS01 = "Recurso 1";
final static String RECURS02 = "Recurso 2";

public static void main(String[] args) {
    try {
        Thread t1 = new Thread1();
        Thread t2 = new Thread2();
        t1.start(); //t1.join();
        t2.start(); //t2.join();
    } catch (Throwable e) {
        e.printStackTrace();
    }
    JOptionPane.showMessageDialog(null, "Programa entrou em DEADLOCK");
    Thread t3 = new Thread3();
    t3.start();
}

private static class Thread1 extends Thread {
    @Override
    public void run() {
        synchronized (RECURS01) {
            try {
                System.out.println("Thread #1: Bloqueou o recurso 1");
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("Thread #1: Tentando acesso ao recurso 2");
            // Aguarda ate obter o recurso 2.
            synchronized (RECURS02) {
                System.out.println("Thread #1: Bloqueou o recurso 2");
            }
        }
    }
}
```



UNIVERSIDADE FEDERAL DE RORAIMA
PROFESSOR (A) : [Herbert Oliveira Rocha](#)
DISCIPLINA DE SISTEMAS OPERACIONAIS.
ALUNO : Ibukun Chife didier Adjitche

```
private static class Thread2 extends Thread {
    @Override
    public void run() {
        synchronized (RECURS02) {
            try { System.out.println("Thread #2: Bloqueou o recurso 2");
                Thread.sleep(50);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
            System.out.println("Thread #2: Tentando o acesso ao recurso 1");
            synchronized (RECURS01) {
                System.out.println("Thread #2: Bloqueou o recurso 1");
            }
        }
    }
}

private static class Thread3 extends Thread {
    public void run() {
        System.out.println("Iniciou Thread 3");
        System.out.println("Thread #3: Entrando em ação");
        JOptionPane.showMessageDialog(null, "Thread #3 no comando");
        JOptionPane.showMessageDialog(null, "Algoritmo do Avestruz implementado com sucesso");
        System.out.println("Finalizou a Thread 3");
        System.exit(0);
    }
}
```