

## Por que aparece o alerta “Resource leak: ‘scanner’ is never closed”?

Esse alerta aparece porque quando você faz:

```
Scanner scanner = new Scanner(System.in);
```

o **Scanner** **abre** um recurso de entrada (um *input stream*).

Qualquer recurso aberto deve ser **fechado**, caso contrário a IDE avisa que isso pode causar um *resource leak*.

---

## Quando esse alerta aparece?

Sempre que:

1. Você cria um **Scanner** (ou qualquer classe que implemente **AutoCloseable**),
2. Mas **não** chama **scanner.close()**.

Exemplo que gera o alerta:

```
Scanner scanner = new Scanner(System.in); // <-- recurso aberto
// ... mas não fecha
```

---

## MAS ATENÇÃO (importante!)

**NÃO** se deve fechar **Scanner(System.in)** na maioria dos casos

Isso está descrito na documentação do Java:

- Fechar um **Scanner** criado com **System.in** fecha também o **System.in**.
- Uma vez fechado, **System.in** **não** pode ser reaberto.
- Isso pode quebrar outras leituras no programa.

Por isso, muitos programadores **ignoram** esse **warning**.

---

## Três formas de resolver o alerta

### 1. Ignorar o alerta (recomendado para **System.in**)

Sim, isso é totalmente normal e seguro.

Muitas IDEs deixam isso como *warning leve* justamente porque **não é esperado que você feche o System.in**.

---

## 2. Usar try-with-resources (não recomendado com System.in)

```
try (Scanner scanner = new Scanner(System.in)) {  
    // uso  
}
```

Mas novamente: isso **fecha o System.in**, o que pode ser indesejado.

---

## 3. Fechar o scanner

Somente quando **NÃO** for System.in

Exemplo (para ler arquivos):

```
Scanner scanner = new Scanner(new File("dados.txt"));  
...  
scanner.close();
```

---

## Resumo simples

Situação	Deve fechar?	Por quê
new Scanner(System.in)	NÃO	Fechar System.in finaliza a entrada padrão permanentemente
new Scanner(File)	SIM	É um recurso real que precisa ser liberado
Warning da IDE	Normal	Apenas um aviso, não erro