

Suponha que você tenha um módulo chamado `com.vet`. Onde você poderia colocar o seguinte arquivo `module-info.java` para criar um módulo válido?

```
public module com.vet {  
    exports com.vet;  
}
```

- A. No mesmo nível que a pasta `com`
- B. No mesmo nível que a pasta `vet`
- C. Dentro da pasta `vet`
- D. Nenhuma das opções acima

ANÁLISE DA QUESTÃO

Vamos entender a estrutura de módulos no Java para poder responder a essa pergunta. Um **módulo no Java** é uma forma de **organizar e encapsular código em pacotes**, proporcionando melhor controle sobre a visibilidade e a acessibilidade do código.

Quando falamos sobre o arquivo `module-info.java`, ele deve estar **localizado no nível do módulo**. Isso significa que ele precisa estar no mesmo nível que a pasta raiz do módulo.

Para ilustrar isso com o exemplo da pergunta:

- O nome do módulo é `com.vet`.
- A estrutura de pastas correspondente a esse módulo seria algo como:

```
src/  
  com/  
    vet/  
      <código fonte>
```

- O arquivo `module-info.java` precisa estar no mesmo nível que a pasta `com`, porque a pasta `com` é a raiz do módulo `com.vet`.

Portanto, a resposta correta poderia ser :

- A. No mesmo nível da pasta `com`

Mas porém contudo, todavia, vamos olhar mais atentamente ao código, a questão está construída de tal modo a enganar um olho menos atento. Veja a linha abaixo:

```
public module com.vet
```

A linha de código na pergunta não está correta! 🤖

A palavra-chave `"public"` não é usada na **declaração de módulos**. A forma correta de declarar um módulo no arquivo `module-info.java` é simplesmente usando a palavra-chave `module` seguida do nome do módulo.

Logo o código em questão vai apresentar um erro de compilação e portanto a alternativa correta será

D. Nenhuma das opções acima