

≡ 05  
**Gerador**

Em Ruby, o que acontece ao tentar executar o código abaixo:

```
module Gerador
  private
  def gera_codigo
    "gerador original"
  end
end

class Cliente

  def gera_codigo
    "novo gerador"
  end

  include Gerador
end

puts Cliente.new.gera_codigo
```

*Selecione uma alternativa*

- A** O interpretador do Ruby detecta ambiguidade ao tentar incluir o módulo
- B** Roda e gera erro de acesso a método inexistente
- C** Roda e imprime "novo gerador"
- D** Roda e imprime "gerador antigo"
- E** Gera erro de acesso a método privado