

07

## Resolvendo duplicação de código

Você está desenvolvendo uma aplicação de geração de notas fiscais de produtos.

Você então cria o método `CalcularItem()`, para obter o valor total de um item da nota fiscal:

```
decimal CalcularItem()
{
    if (quantidade * precoItem > 1000) {
        return quantidade * precoItem * 0.95;
    }
    else {
        return quantidade * precoItem * 0.98;
    }
}
```

Em seguida, percebe que o código acima pode ser refatorado para:

```
decimal CalcularItem()
{
    var precoBase = quantidade * precoItem;

    if (precoBase > 1000) {
        return precoBase * 0.95;
    }
    else {
        return precoBase * 0.98;
    }
}
```

A seguir, você também percebe que existe duplicação da expressão `quantidade * precoItem`, pois essa mesma expressão já está sendo utilizada em outros métodos da classe.

Como você resolve essa duplicação de código?

*Selecione uma alternativa*

**A**

Extraindo a expressão `precoBase * 0.95` para um método `GetValorComDesconto()` e extraindo a expressão `precoBase * 0.98` para um método `GetValorNormal()`. A seguir, substituindo as expressões por chamadas a esses métodos.

**B**

Movendo a expressão `quantidade * precoItem` para um novo método separado `GetPrecoBase()`, e retorna o resultado a partir dessa expressão. Em seguida, substituindo às referências à variável `precoBase` e também à expressão `quantidade * precoItem` por chamadas ao método `GetPrecoBase()`.

**C**

Transformando a variável local `precoBase` num campo com escopo da classe. Em seguida, substituindo a expressão `quantidade * precoItem` por esse campo nos outros métodos da classe.

