

## Validando CPF

Considere uma aplicação com o seguinte método `CadastrarAluno`, que toma os parâmetros `cpf`, `nome` e `data de nascimento` para salvar as informações do aluno no banco de dados.

```
public void CadastrarAluno(string cpf, string nome, DateTime dataNascimento)
{
    dataService.SaveStudent(cpf, nome, dataNascimento);
}
```

Certo dia, você faz uma consulta ao banco de dados e descobre que vários CPFs estão com valores inválidos. Você decide reforçar as regras de entradas de dados, modificando o método `CadastrarAluno` para lançar uma exceção para o cliente caso o CPF seja inválido. Como você modificaria o código acima? Escolha a melhor opção.

*Selecione uma alternativa*

**A**

```
public void CadastrarAluno(string cpf, string nome, DateTime dataNascimento)
{
    if (new CPFValidator().IsValid(cpf))
    {
        dataService.SaveStudent(cpf, nome, dataNascimento);
    }
}
```

**B**

```
public void CadastrarAluno(string cpf, string nome, DateTime dataNascimento)
{
    new CPFValidator().IsValid(cpf);
    dataService.SaveStudent(cpf, nome, dataNascimento);
}
```

**C**

```
using Caelum.Stella.CSharp.Validation
.
.
.
public void CadastrarAluno(string cpf, string nome, DateTime dataNascimento)
{
    new CPFValidator().AssertValid(cpf);
    dataService.SaveStudent(cpf, nome, dataNascimento);
}
```