

06

As várias formas de declarar um pipe

Vimos durante a aula uma forma de declaração de pipes na aplicação, mas existem outras.

Marque as opções com as afirmações verdadeiras.

DICA: Dê uma olhada na [documentação do NestJS \(https://docs.nestjs.com/pipes\)](https://docs.nestjs.com/pipes) relativa a pipes para aprofundar ainda mais seus conhecimentos!

Selecionar 4 alternativas

A

```
const app = await NestFactory.create(AppModule);
app.useGlobalPipes(new ValidationPipe());
```

B

```
@Post()
async cria(
  @Body(new ValidationPipe()) livro: Livro) {
  this.livroService.cria(livro);
}
```

C

```
import { Module } from '@nestjs/common';
import { APP_PIPE } from '@nestjs/core';

@Module({
  providers: [
```

```
{  
  provide: APP_PIPE,  
  useClass: ValidationPipe,  
},  
],  
})  
export class AppModule {}
```

D

Numa aplicação NestJS, pipes só podem ser declarados com o método `useGlobalPipes()` fazendo deles pipes globais. Ou através do decorator `@UsePipes()` que deve ser utilizado apenas em controladores, definindo assim que determinado pipe será utilizado para todos os endpoints desse controlador.

E

```
@Post()  
@UsePipes(new ValidationPipe())  
async cria(@Body() livro: Livro) {  
  this.livroService.cria(livro);  
}
```