

Italo Sousa de Alencar - 495638
Luis Antonio da Silva Maia - 493458
José Portela Soares Neto - 377181

TB5 - Testes Unitários

Verificação, Validação e Teste de Software (CK0241)

O que é Jest?



Jest é um poderoso Framework de Testes em JavaScript com um foco na simplicidade.

Funciona com projetos usando: TypeScript, Node, React, Angular, Vue, etc.



Suíte de testes

Servem para definir o escopo do que está sendo testado.

No Jest, a suíte é uma função global JavaScript chamada 'describe', que possui dois parâmetros, que seriam sua descrição e os testes (specs).

```
describe("Operação de Adição", () => { });
```



Specs (Testes)

Specs são os testes que validam uma suíte de testes.

Assim como as suítes, ela é uma função global Javascript chamada 'it', que contém dois parâmetros: uma descrição e uma função, respectivamente. Dentro do segundo parâmetro, é onde adicionamos as verificações (expectations).

```
it("deve garantir que 1 + 9 = 10", () => { });
```

Verificações (Expectations)

Verificações servem para validar um resultado de um teste.

O Jest possui uma função global Javascript chamada 'expect', que recebe um parâmetro como argumento, que é o resultado a ser verificado.

```
expect(Calculadora.adicionar(1, 9)).toBe(10);
```



Método Escolhido - Criação de Produto

O seguinte método tem como objetivo inserir um novo produto no sistema GoShop.

```
export const createProduct = async (req: Request, res: Response, next: NextFunction) => {
  try {
    const image = req.image || "";
    const stripeProduct = await stripe.products.create({
      name: req.body.name,
      description: req.body.description,
      default_price_data: {
        currency: "usd",
        unit_amount: req.body.price * 100
      },
      // images: [image]
    });
  }
};
```

```
const newProduct = await prisma.product.create({
  data: {
    stockQuantity: Number(req.body.stockQuantity),
    price: Number(req.body.price),
    id: stripeProduct.id,
    priceId: stripeProduct.default_price as string,
    image,
    name: req.body.name,
    description: req.body.description,
    category: {
      connectOrCreate: {
        where: {
          name: req.body.category
        },
        create: {
          name: req.body.category
        }
      }
    }
  }
});
```

```
    res.status(201).json(newProduct);  
  } catch (error) {  
    console.log(error)  
    next({ message: "Unable to create the product with given details", error });  
  }  
};
```



Teste para o método escolhido

```
describe('Product', () => {
  let productId : any;

  it('should be able to create a product', async () => {

    const body = {
      name: "Teclado RGB",
      description: "Teclado para gamers",
      price : "750",
      stockQuantity: "3",
      category: "Teclados",
    };

    const mockReq = { body } as Request;

    const mockRes: any = { json: jest.fn(), status: jest.fn().mockReturnThis() };

    const mockNext = jest.fn();

    await createProduct(mockReq, mockRes, mockNext)

    const jsonCall = mockRes.json.mock.calls[0];
    productId = jsonCall[0].id

    expect(mockRes.json).toHaveBeenCalledWith(expect.objectContaining({ name: body.name, description: body.description }));
  });
});
```

Execução de testes

```
PASS test/testes.spec.ts
```

```
Product
```

```
✓ should be able to create a product (733 ms)  
✓ should be able to update a product (681 ms)  
✓ Should be able to find an product (4 ms)  
✓ Should not be able to find an product (2 ms)  
✓ Should be able to delete an product (372 ms)
```

```
□
```

```
Test Suites: 1 passed, 1 total
```

```
Tests: 5 passed, 5 total
```

```
Snapshots: 0 total
```

```
Time: 3.865 s, estimated 4 s
```

```
Ran all test suites.
```

Referências

<https://jestjs.io/pt-BR/>

<https://blog.brq.com/testes-unitarios-com-jest/>

