

Controle de Versões			
Versão	Data	Autor	Notas da Revisão
1.0	19/09/2021	Greice Castilho	Inclusão de conteúdo / sprint 1
1.0	09/10/2021	Greice Castilho	Atualização de conteúdo / sprint 2
1.0	07/11/2021	Greice Castilho	Atualização de conteúdo / sprint 3
1.0	28/11/2021	Greice Castilho	Atualização de conteúdo / sprint 4

## Projeto de Testes de Softwares

### Introdução

O propósito deste projeto é a aplicação dos conhecimentos adquiridos durante a matéria de testes de softwares ministrada pelo professor Gerson da Penha Neto na Fatec-faculdade de tecnologia de São José dos Campos-SP no segundo semestre letivo de 2021.

A proposta do projeto é um dashboard de vendas onde o usuário conseguirá acompanhar o desempenho de vendas dos vendedores cadastrados. Nesta plataforma web, o usuário poderá visualizar em forma de lista todos os vendedores e suas atividades classificadas por: data, nome, clientes visitados, negócios fechados e o valor das suas vendas.

A plataforma ainda irá contar com os seguintes gráficos:

- Um gráfico de barras: onde apresentará as taxas de sucesso dos vendedores, ou seja, qual a porcentagem de vendas aquele vendedor realizou X a quantidade de clientes visitados.
- Um gráfico de rosca: onde apresentará as porcentagens de vendas de cada vendedor.

O projeto desenvolvido está documentado no github, este projeto é opensource e poderá ser acompanhado e consultado pelo link:  
<https://github.com/krusader1982/projetotsdashboard>

Este projeto está sendo desenvolvido em JAVA utilizando o Spring Boot no backend e React no frontend.

## Cronograma de entregas

As etapas de desenvolvimento utilizando a metodologia ágil será de um semestre e contará com 4 sprints obedecendo os seguintes prazos:

Cronograma	
30/08 a 19/09	Sprint 1
20/09 a 10/10	Sprint 2
18/10 a 07/11	Sprint 3
08/11 a 28/11	Sprint 4
29/11 a 05/12	Apresentação Final

Durante o desenvolvimento do projeto iremos realizar os testes pertinentes a cada fase, contando com a orientação e instruções do professor no decorrer deste processo.

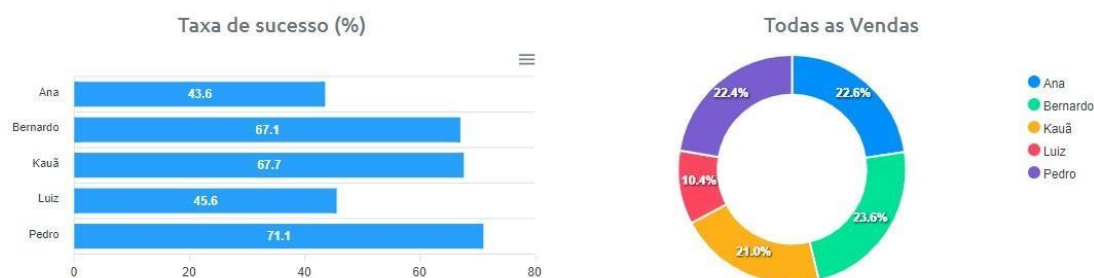
Previsão de entregas:



## Protótipo

Este é o protótipo do projeto de um dashboard de vendas.

### Dashboard de Vendas



**Figura 1:** Um gráfico de barras (taxa de sucesso) e um gráfico de rosca (todas as vendas por vendedor).

### Lista de Vendas

Data	Vendedor	Clientes visitados	Negócios fechados	Valor
22/04/2021	Ana	34	25	15017.00
22/04/2021	Ana	34	25	15017.00
22/04/2021	Ana	34	25	15017.00
22/04/2021	Bernardo	34	25	15017.00
22/04/2021	Luiz	34	25	15017.00
22/04/2021	Bernardo	34	25	15017.00
22/04/2021	Pedro	34	25	15017.00
22/04/2021	Kauã	34	25	15017.00

**Figura 2:** Apresentação do desempenho dos vendedores em forma de lista, apresentando dados como: data, nome do vendedor, clientes visitados, negócios fechados e valores das vendas realizadas.

<b>Fatec</b> São José dos Campos Prof. Jessen Vidal	<b>Matéria: Testes de Softwares</b>
	FATEC São José dos Campos - Prof. Jessen Vidal

## Sprint 2

### Modelo conceitual do banco de dados

#### Modelo conceitual



powered by Astah

No modelo de dados conceitual podemos visualizar que um vendedor possui várias vendas e que uma venda tem somente um vendedor.

O vendedor possui:

um id (identificador único) e um nome.


E vendas possui:

Id (identificador único), clientes visitados (visited), negócios fechados (deals), valor das vendas realizadas (amount) e a data da venda (date).

### Desenvolvimento

Nesta segunda sprint realizamos o desenvolvimento do frontend do projeto. O frontend foi desenvolvido em React e os dados foram utilizados de forma simulada (mockado).

Após a conclusão do desenvolvimento utilizamos o Netlify (uma plataforma de hospedagem em nuvem) para colocar o projeto, desta forma ficou mais cômodo e fácil visualizar o resultado obtido e traçar a próxima fase do desenvolvimento que será a parte de backend e persistência de dados.

	<b>Matéria: Testes de Softwares</b>
	FATEC São José dos Campos - Prof. Jessen Vidal

## Sprint 3

### Desenvolvimento

Nesta terceira sprint nos concentramos em desenvolver o backend e na implantação do banco de dados da aplicação. Durante o desenvolvimento já fizemos a conexão entre o frontend e o backend.

### Estruturar o projeto em camadas

Padrão de camadas adotado:

- Criar repositories
- Criar DTO's
- Criar service
- Criar controller


Ao final foi feito um deploy do projeto no Heroku.

## Sprint 4

Nesta quarta e última sprint do projeto realizamos os testes da aplicação.

Os testes desenvolvidos foram testes do tipo caixa branca, ou seja, neste teste temos acesso ao código fonte e conseguimos analisar por qual caminho ocorre o fluxo de dados e conseguimos verificar se há a passagem correta em todas as condições esperadas.

Desenvolvemos alguns testes unitários, esse tipo de teste de software analisa um componente isolado ou classe do sistema.

	<b>Matéria: Testes de Softwares</b>
	FATEC São José dos Campos - Prof. Jessen Vidal

Exemplos dos testes desenvolvidos:

### 1º Teste

```
@Test //Teste para encontrar todos Vendedores cadastrados
void testFindAll() {
    List<SellerDTO> sellers = seller.findAll();
    assertEquals(sellers.size(), 5);
}
```

Realizamos o teste para encontrar todos os vendedores cadastrados na plataforma, neste caso se faz uma busca na lista SellerDTO dos vendedores usando o método findAll e depois através do método de comparação assertEquals passamos os parâmetros que estamos buscando e o parâmetro esperado. Neste caso acima temos 5 vendedores cadastrados na plataforma então colocamos o valor de 5 como resultado esperado.

### 2º Teste


```
@Test//Verifica se o retorno da lista de vendedores esta vazia
void testFindAllTest1(){
    List<SellerDTO> sellers = seller.findAll();
    assertEquals(sellers.isEmpty(), false);
}
```

Neste segundo teste os métodos utilizados foram os mesmos, porém aqui queríamos verificar se a lista de vendedores cadastrados estava vazia, então passamos o parâmetro de lista vazia e o argumento de falso, já que a lista não deve estar vazia.

### 3º Teste

```
@Test //Teste para verificar se esta retornando um numero errado de Vendedores cadastrados
void testFindAllTest() {
    List<SellerDTO> sellers = seller.findAll();
    assertNotEquals(sellers.size(), 6);
}
```

Neste teste foi verificado se a quantidade de vendedores cadastrados estava correta, observe que neste caso usou o método de comparação assertNotEquals e como parâmetros de consulta a quantidade de vendedores cadastrados e uma quantidade de

	<b>Matéria: Testes de Softwares</b>
	FATEC São José dos Campos - Prof. Jessen Vidal

6 vendedores. Sabendo que na plataforma não consta 6 vendedores cadastrados o teste deve ser validado já que o método de comparação utilizado foi o assertEquals.

#### 4º Teste

```


@Test //Teste para verificar se esta salvando o vendedor
void sellerSaveTestOK(){
    Seller sel = new Seller();
    sel.setName("Samuel");
    selRepo.save(sel);
    assertNotNull(sel.getId());
}

@Test
void saleSaveTestOK(){
    Sale sal = new Sale();
    sal.setId((long) 1);
    sal.setVisited(null);
    sal.setDeals(null);
    sal.setAmount(null);
    sal.setDate(null);
    //sal.setSeller(5);
    salRepo.save(sal);
    assertNotNull(sal.getId());
}

@Test
void saleServiceCadastrarSaleTest(){
    sale.cadastrarSale((long)170,45,25,5561.0,null,null);
}

```

Neste teste simulamos o cadastro de um novo vendedor.

	<b>Matéria: Testes de Softwares</b>
	FATEC São José dos Campos - Prof. Jessen Vidal

### 5° Teste

```
@Test //Teste para verificar o service
void sellerServiceCadastrarSellerTest() {
    seller.cadastrarSeller("Samuel");
    assertNotNull(selRepo.findByName("Samuel"));
}
```

Realizamos um teste para verificar o service e se o vendedor recém cadastrado estava constando no banco. Para isso usamos o método findByName e o nome do vendedor recém cadastrado como parâmetro.

### 6° Teste

```
@Test //Teste para verificar se encontra um vendedor pelo nome
void findByNameTest() {
    List<SellerDTO> sellers = seller.findAll();
    assertNotNull(selRepo.findByName("Barry Allen"));
}
```

E neste último teste verificamos se é possível encontrar um vendedor pelo nome, também usando o método findByName.

## Conclusão

Através deste projeto conseguimos desenvolver vários testes unitários e aplicar o conhecimento adquirido durante o semestre na matéria de testes de software, também conseguimos entender a importância dos testes de software para uma aplicação e a valorizar a sua utilidade.