

## User Stories

**1a) Defina o que é uma User Story e qual é sua finalidade no desenvolvimento de software ágil.**

Uma User Story é uma técnica no desenvolvimento de software ágil que descreve requisitos do ponto de vista do usuário, com o objetivo de focar nas necessidades do usuário, comunicar de forma clara, permitir a priorização flexível, suportar a entrega contínua e garantir a testabilidade.

**1b) Dada a seguinte demanda do cliente: "Um cliente deseja pesquisar produtos no ecommerce por nome possibilitando encontrá-los rapidamente". Escreva uma User Story adequada para essa demanda, incluindo critérios de aceitação**

História do Usuário: Pesquisa Rápida de Produtos

Como um cliente que deseja fazer compras no e-commerce, eu gostaria de pesquisar produtos por nome para que eu possa encontrar rapidamente os itens que estou procurando.

Critérios de aceitação:

- Permitir pesquisa por nome de produto.
- Exibir resultados de pesquisa com base na correspondência com o nome do produto.
- Permitir correspondência parcial de nomes.
- Apresentar resultados em ordem de relevância.
- Permitir acesso às informações detalhadas do produto a partir dos resultados.
- Oferecer sugestões de pesquisa à medida que o usuário digita.
- Realizar testes de unidade e integração para garantir o funcionamento correto.
- Garantir desempenho eficiente, mesmo com um grande número de produtos.
- Adaptar-se a dispositivos móveis e desktops para uma experiência consistente.

**1c) Crie 2 histórias de usuário adicionais para este cenário do e-commerce, incluindo critérios de aceitação.**

História do Usuário 1: Adicionar Produto ao Carrinho

Como um cliente que deseja comprar produtos no e-commerce, eu gostaria de adicionar produtos ao carrinho de compras para que eu possa revisar e finalizar minha compra posteriormente.

Critérios de aceitação:

- Adição de produtos ao carrinho
- Quantidade ajustável para cada produto.
- Exibição clara do carrinho de compras.
- Resumo completo da compra no carrinho.
- Capacidade de remover produtos individuais do carrinho.
- Botão "Finalizar Compra" para concluir a compra.

## História do Usuário 2: Visualizar Histórico de Compras

Como um cliente que já fez compras no e-commerce, eu gostaria de visualizar meu histórico de compras anteriores para que eu possa rever os produtos que comprei no passado.

Critérios de aceitação:

- Acesso ao histórico de compras a partir da conta do usuário.
- Listagem de todas as compras anteriores com datas e números de pedido.
- Detalhes completos das compras, incluindo produtos, quantidades, preços e informações de entrega.
- Navegação intuitiva no histórico de compras.
- Compatibilidade com dispositivos móveis e desktops para uma experiência consistente.

**1d) Leia atentamente a seguinte User Story. Apresente possíveis critérios de aceitação:**

“Como gerente de vendas, desejo gerar relatórios mensais de vendas para analisar o desempenho dos produtos”

- Seleção de Período: O sistema deve permitir que o gerente de vendas selecione o período (mês e ano) para o qual deseja gerar o relatório.
- Filtros Opcionais: Deve haver opções de filtro adicionais, como categoria de produtos ou região geográfica, para permitir ao gerente de vendas refinar os resultados do relatório, se necessário.
- Dados de Vendas: O relatório deve incluir informações detalhadas sobre as vendas mensais, como o número de unidades vendidas, receita total e margem de lucro para cada produto.
- Gráficos e Visualizações: O sistema deve apresentar gráficos ou visualizações claras que mostrem o desempenho dos produtos de forma intuitiva.
- Exportação de Dados: Deve ser possível exportar o relatório em formatos comuns, como PDF ou Excel, para compartilhamento ou referência futura.
- Atualização Automática: O relatório deve ser gerado automaticamente no início de cada novo mês sem a necessidade de intervenção manual.
- Acesso Restrito: A funcionalidade de geração de relatórios deve estar restrita apenas ao gerente de vendas ou a usuários com as permissões apropriadas.
- Desempenho Eficiente: O processo de geração do relatório deve ser eficiente, mesmo quando há grandes volumes de dados de vendas.

## Test-Driven Development (TDD)

**2) Considerando uma função que inverte uma string nomeda `string_invert`. O pseudo código está abaixo**

```
FUNÇÃO string_invert(s) RETORNA string:
    DECLARE str_invertida = ""
    PARA CADA caractere c EM s DE TRÁS PARA FRENTE FAÇA:
        str_invertida += c
    FIM PARA
    RETORNE str_invertida
FIM FUNÇÃO
```

**Escolha entre JUnit, PyTest, Jest ou qualquer outra alternativa de teste unitários que preferir. Criar testes para:**

- 2a) strings vazias,**
- 2b) strings com um único caractere e**
- 2c) strings mais que um caractere.**

**2a) strings vazias.**

```
// Teste para strings vazias
test('Inverter uma string vazia deve retornar uma string vazia', () => {
    expect(string_invert('')).toBe('');
});
```

**2b) strings com um único caractere e.**

```
// Teste para strings com um único caractere
test('Inverter uma string com um único caractere deve retornar a mesma string', () => {
    expect(string_invert('a')).toBe('a');
    expect(string_invert('1')).toBe('1');
});
```

**2c) strings mais que um caractere.**

```
// Teste para strings com mais de um caractere
test('Inverter uma string com mais de um caractere deve retornar a string invertida', () => {
    expect(string_invert('abcdefg')).toBe('gfedcba');
    expect(string_invert('12345')).toBe('54321');
});
```

**3) Dado o seguinte teste unitário escrito em pseudocódigo:**

**Teste "verificarContagemDeCaracteres":**

**resultado = contarCaracteres("Live long & prosper")**

**verificarSe(resultado é igual a 18)**

**3a) Escreva uma possível função contarCaracteres (str) que atenda a esse teste. Use a linguagem que estiver mais habituado.**

```
function contarCaracteres(str) {  
  return str.length;  
}
```

**3b) Escreva pelo menos 1 outro teste unitário para a função contarCaracteres (str)**

// Teste para uma string vazia

```
test("verificarContagemDeCaracteres para string vazia", () => {  
  const resultado = contarCaracteres("");  
  expect(resultado).toBe(0);  
});
```

// Teste para uma string com espaços em branco

```
test("verificarContagemDeCaracteres para string com espaços em branco", () => {  
  const resultado = contarCaracteres(" Hello, World! ");  
  expect(resultado).toBe(15);  
});
```

// Teste para uma string com caracteres especiais

```
test("verificarContagemDeCaracteres para string com caracteres especiais", () => {  
  const resultado = contarCaracteres("123!@#$%^&*()_+");  
  expect(resultado).toBe(14);  
});
```