

# Artigo de Software Front-end

Daniella Ferreira Marques

<sup>1</sup>Instituto Federal do Piauí - IFPI - Campus Picos

isabelmarques902@gmail.com

**Abstract.** Tradução do texto original para o inglês.

**Keywords:**

**Resumo.** Para a elaboração do resumo e do abstract em um trabalho acadêmico, é recomendado utilizar fontes como Arial ou Times New Roman, tamanho 12, com espaçamento entre linhas simples de 1,0. É importante que o resumo contenha entre 150 a 250 palavras, incluindo informações sobre os objetivos do estudo, fundamentação teórica, metodologia, resultados e conclusões ou considerações. É fundamental que o texto não contenha citações ou siglas.

**Palavras - chave:** O documento requer a inclusão de um conjunto de descritores composto por um mínimo de três e um máximo de cinco termos, cujas letras iniciais devem ser grafadas em minúsculo, separados por ponto e vírgula, e terminados com um ponto final.

## 1. Introdução

Quando escrevemos a introdução de um artigo, é fundamental fornecer informações essenciais que contextualizem o leitor sobre o conteúdo da pesquisa. Existem três elementos-chave que devem ser incluídos: o tema, o problema e a hipótese.

O tema do artigo refere-se ao assunto principal que será abordado na pesquisa. Ele deve ser apresentado de forma clara e concisa, de modo que o leitor possa compreender imediatamente sobre o que o artigo se trata. Já o problema de pesquisa é a questão que impulsiona o estudo. É importante apresentar o problema de maneira precisa e objetiva, demonstrando qual é a questão que será investigada e por que é relevante pesquisá-la. Por fim, a hipótese é uma suposição inicial feita pelo pesquisador sobre a relação entre as variáveis em estudo. Ela deve ser fundamentada em evidências científicas e possuir uma conexão clara com o problema de pesquisa.

Além disso, a introdução deve oferecer uma visão geral da estrutura do artigo, destacando como o estudo foi organizado e quais são os principais pontos discutidos em cada seção. Isso auxilia o leitor a se orientar durante a leitura. É essencial que o tema, o problema e a hipótese sejam expostos de maneira concisa e compreensível, enfatizando a relevância do tema, delimitando o escopo da pesquisa e fornecendo uma visão geral da estrutura do artigo.

## 2. Metodologia

Ao planejar o desenvolvimento de uma plataforma, é essencial considerar diversos requisitos estratégicos, como tecnologias utilizadas, volume de acessos, usabilidade, escalabilidade e confiabilidade. Com o objetivo de criar uma plataforma simples e eficiente, o processo pode ser desenvolvido em três etapas distintas:

- Etapa inicial de desenvolvimento: Nessa fase, a plataforma é construída, levando em conta os requisitos e as tecnologias escolhidas. A equipe de desenvolvimento se concentra na implementação das funcionalidades principais e na criação da estrutura básica da plataforma.
- Testes: Após a conclusão da etapa de desenvolvimento, são realizados testes para identificar e corrigir possíveis erros de código. Esses testes visam garantir a qualidade e a estabilidade da plataforma, verificando se ela funciona conforme o esperado e atende aos requisitos definidos.
- Validação com usuários finais: Nessa etapa, a plataforma é submetida à validação por parte dos usuários finais. O foco é avaliar a usabilidade, confiabilidade e efetividade da plataforma. Os usuários são convidados a interagir com a plataforma, realizar tarefas específicas e fornecer feedback sobre sua experiência. Essa validação ajuda a identificar possíveis melhorias e a garantir que a plataforma atenda às necessidades dos usuários de forma eficiente.

### **3. Estado da Arte**

O estado da arte é basicamente o que há de mais avançado em uma determinada área de pesquisa. É importante para os pesquisadores entenderem o que já foi feito e quais questões ainda precisam ser respondidas. Isso ajuda a escolher a melhor metodologia e ferramentas, e também dá contexto aos resultados.

Para descobrir o estado da arte de um software, é preciso estudar a literatura existente, pesquisar as tecnologias mais novas e aprender com as boas práticas recomendadas na área de interesse. Isso permite ter uma visão geral das tendências e inovações no desenvolvimento de software e aplicar o conhecimento adquirido para criar softwares mais eficientes e de qualidade. O estado da arte de um determinado produto ou tecnologia pode ser dividido em dois tipos: concorrentes diretos e indiretos.

#### **3.1. Concorrentes Diretos**

Os concorrentes diretos são aqueles que oferecem produtos ou tecnologias que têm o mesmo objetivo ou função do produto em questão. Por exemplo, se o produto em questão é um smartphone, seus concorrentes diretos são outros smartphones no mercado.

#### **3.2. Concorrentes Indiretos**

Concorrentes indiretos são produtos ou soluções que oferecem funcionalidades ou benefícios semelhantes, mas que não competem diretamente com o software em questão. Eles podem estar em diferentes setores ou mercados, mas ainda assim podem impactar a adoção ou utilização do software em questão. Por exemplo, se o produto é um smartphone, seus concorrentes indiretos são o e-reader (leitor de livros digitais). Embora os smartphones possam oferecer aplicativos de leitura de livros, os e-readers são dispositivos eletrônicos dedicados para esse fim, oferecendo uma experiência de leitura mais confortável para os usuários.

### **4. Desenvolvimento do Software**

Aqui está um exemplo de um template de visão do produto que pode ser utilizado como referência.

Para (cliente) que (necessidade)  
 o (seu produto) é um (tipo de produto)  
 que (benefícios) diferente de (alternativas)  
 nosso produto (diferenciais)

**Figura 1. Visão do produto**

#### 4.1. Personas

Uma persona em UX (User Experience) é uma representação fictícia, porém baseada em dados reais, de um usuário ideal para um produto ou serviço. Ela é criada com o objetivo de ajudar a equipe de UX a compreender melhor as necessidades, objetivos e comportamentos dos usuários.

A criação de uma persona envolve a realização de pesquisas e análises de dados de usuários, como entrevistas, questionários e análise de métricas de uso e comportamento. A partir dessas informações, a equipe de UX pode criar uma persona que representa um usuário típico, retratando suas características demográficas, preferências, comportamentos, necessidades e desafios. A figura 2 ilustra uma persona para um aplicativo de música.

AS PRINCIPAIS PERSONAS DO PRODUTO	
<p><b>APELIDO</b> (AJUDA A LEMBRAR SEMPRE)</p> 	<p><b>PERFIL DA PERSONA</b></p> <ul style="list-style-type: none"> <li>- PERSONA TEM 30 ANOS</li> <li>- CASADA</li> <li>- TEM 3 FILHOS</li> <li>- ETC</li> </ul>
<p><b>CARACTERÍSTICAS COMPORTAMENTAIS DA PERSONA</b></p> <ul style="list-style-type: none"> <li>- GOSTA DE TECNOLOGIA</li> <li>- BAIXA MUITO APP NO CELULAR</li> </ul>	<p><b>NECESSIDADES DA PERSONA</b></p> <ul style="list-style-type: none"> <li>- PRECISA DE APP PARA OUVIR MUSICA</li> <li>- PRECISA DE VARIAS TIPOS DE PLAYLIST PARA CADA DIA DA SEMANA.</li> </ul>

**Figura 2. Persona**

#### 4.2. Mapa de Empatia

O mapa de empatia é uma ferramenta visual que destaca as necessidades, desejos, emoções e comportamentos dos clientes ou usuários de um produto ou serviço. Ele coloca o usuário no centro do processo de desenvolvimento, sendo usado no design thinking. O mapa de empatia é dividido em seis seções que abrangem diferentes aspectos do usuário:

o que ele vê, ouve, pensa e sente, fala e faz, suas dores e suas necessidades. Ao preencher essas seções, obtemos uma visão empática e abrangente do usuário, permitindo criar soluções mais eficazes e centradas nele. A figura 3 ilustra um exemplo do mapa de empatia.



Figura 3. Mapa de empatia

### 4.3. Fluxo da Jornada do Usuário

O fluxo da jornada do usuário é uma representação visual das etapas que um usuário realiza para alcançar um objetivo em um produto ou serviço. É usado em UX para entender a interação do usuário ao longo do tempo, desde a primeira interação até a conclusão da tarefa. O fluxo da jornada do usuário é criado por meio de diagramas ou wireframes, destacando ações, decisões, feedback e emoções do usuário.

Na figura 4, temos um fluxo básico da jornada do usuário em um aplicativo de entrega de comida, abrangendo desde a pesquisa até a finalização da entrega. O objetivo é oferecer uma experiência simples e conveniente para o usuário, reduzindo possíveis obstáculos e aumentando a satisfação do usuário.

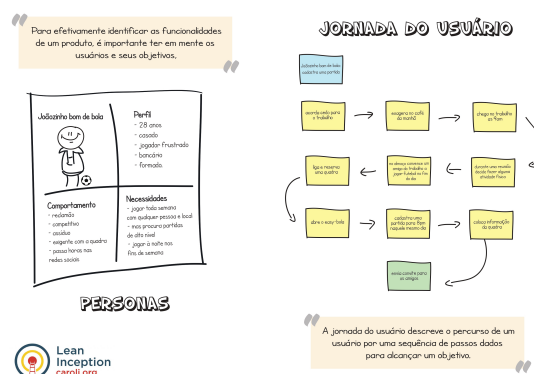


Figura 4. Fluxo da jornada do usuário

#### 4.4. Product Backlog

Nesta etapa é criado o product backlog que é “Uma sacolinha de funcionalidades do produto.” (Aislan Rafael, 2022). Através dele podemos ter uma visão mais focada a utilidade de cada funcionalidade e sua importância para o sistema possibilitando um desenvolvimento mais otimizado.

O Product Backlog é uma lista priorizada de requisitos, funcionalidades e melhorias necessárias para um produto ou projeto. Ele é constantemente atualizado pelo Product Owner com base no valor de negócio e feedback dos clientes. Durante as reuniões de planejamento do sprint, a equipe seleciona os itens do backlog para trabalhar no próximo ciclo de desenvolvimento. Isso permite um desenvolvimento incremental e adaptativo do produto. A figura 5 ilustra um template para listar o product backlog.

	A	B	C	D	E	F	G	H
1	PRODUCT BACKLOG							
2								
3	IBL	TEMA	DESCRIÇÃO	VALOR DE NEGÓCIO	ESTIMATIVA (STORY POINTS)	ROI (VN/SP)	SPRINT	
4	0	Tema (ex. Produtos)	Como <PAPEL> eu quero/gostaria/preciso de <FUNCIONALIDADE> para <VALOR DE NEGÓCIO>	1000	8	160,5	1	
5	1							
6	2							
7	3							
8	4							
9	5							
10	6							
11	7							
12	8							
13	9							
14	10							
15	11							
16	12							
17	13							
18	14							
19	15							
20	16							
21	17							
22	18							
23	19							
24	20							
25	21							
26	22							
27	23							
28	24							
29	25							
30	26							
31	27							
32	28							
33	29							
34	30							
35								
36								
37								

Figura 5. Template do product backlog

#### 4.5. Rabisco Frame

O wireframe de baixa fidelidade, também conhecido como rabisco frame, é uma representação visual simplificada de um site ou aplicativo. Ele é usado para planejar a estrutura e o layout do projeto antes do desenvolvimento. Os wireframes de baixa fidelidade são compostos por formas básicas e texto, enfocando a funcionalidade e a experiência do usuário. São rápidos de criar e permitem receber feedback dos stakeholders e da equipe de desenvolvimento em estágios iniciais do projeto. A figura 6 e 7 ilustram um exemplo do rabisco frame.

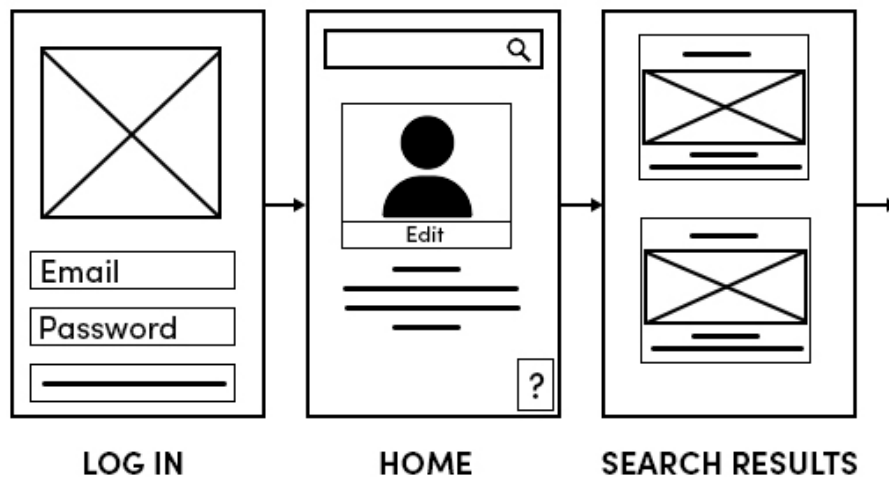


Figura 6. Exemplo de um rabisco frame

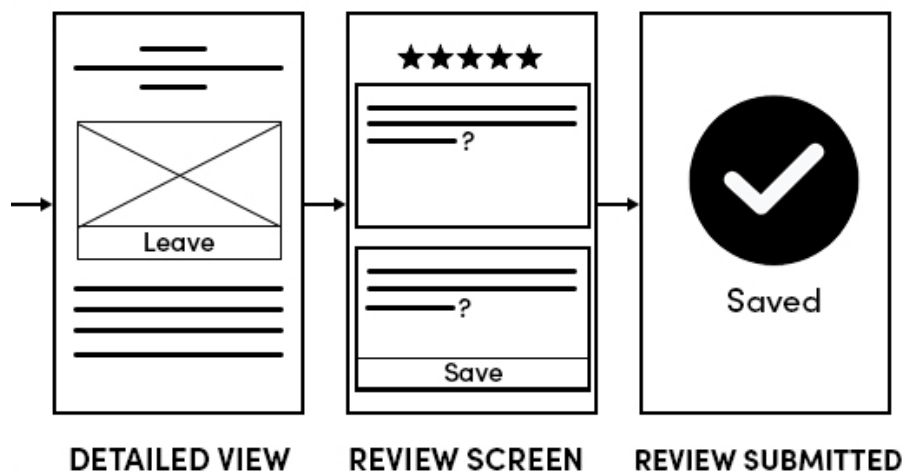


Figura 7. Exemplo de um rabisco frame

## 5. Conhecendo o sistema

Neste capítulo serão apresentados as telas e as funcionalidades do sistema web ou aplicativo.

### 5.1. Tecnologias utilizadas

Neste tópico, fornecemos informações relevantes sobre as ferramentas, linguagens de programação, bibliotecas e frameworks utilizados no desenvolvimento do software em questão. Essas escolhas tecnológicas desempenham um papel fundamental no funcionamento e desempenho do software, e sua compreensão é essencial para uma visão completa do projeto.

## **5.2. Protótipo de telas**

Nesta seção, incluímos imagens das telas do protótipo de software. Um protótipo de software é uma versão preliminar e simplificada do software em desenvolvimento. Ele é criado para testar e validar conceitos, funcionalidades e fluxos de interação antes do desenvolvimento completo. O objetivo principal é obter feedback dos usuários para melhorar o design e a usabilidade. Os protótipos podem variar em fidelidade, desde simples esquemas até representações visuais mais próximas do produto final. A escolha do tipo de protótipo depende dos objetivos e estágio do desenvolvimento do software.

## **6. Avaliação**

Nesta seção, serão apresentados os testes de infraestrutura e usabilidade realizados, bem como os resultados obtidos, visando assegurar a confiabilidade, usabilidade e integridade do sistema.

### **6.1. Teste de desempenho**

O teste de desempenho avalia o desempenho, estabilidade e escalabilidade de um sistema em condições de carga e estresse. Simula situações reais com muitos usuários ou carga intensiva, analisando o tempo de resposta, taxa de transferência, capacidade de processamento e utilização de recursos. Identificando gargalos e pontos fracos, o teste permite otimizar o sistema, ajustar configurações e melhorar a experiência do usuário. Existem várias ferramentas gratuitas disponíveis para realizar testes de desempenho. Aqui estão alguns exemplos:

- Page Speed Insights (O Google PageSpeed: É uma família de ferramentas do Google Inc, projetada para ajudar na otimização do desempenho de um site), com esta ferramenta é possível identificar problemas de codificação, usabilidade, carregamento de páginas e renderização das páginas para dispositivos de telas diferentes.
- Apache JMeter: É uma ferramenta de teste de carga e desempenho amplamente utilizada. Permite simular diferentes cargas de trabalho, medir o desempenho de servidores web, aplicativos e serviços, além de oferecer recursos avançados de relatórios e análise

### **6.2. Teste de usabilidade**

O teste de usabilidade avalia a facilidade de uso e a experiência do usuário em um produto. Os usuários realizam tarefas enquanto são observados por especialistas. Os resultados identificam problemas e ajudam a melhorar o design e a interação. O teste de usabilidade pode ser realizado seguindo algumas etapas básicas:

- Defina os objetivos: Determine quais aspectos do produto você deseja avaliar e quais perguntas deseja responder por meio do teste de usabilidade.
- Identifique o público-alvo: Selecione os usuários que representam o público-alvo do produto. Eles devem ter características que reflitam os usuários reais do produto.
- Crie cenários de teste: Desenvolva tarefas ou cenários realistas que os usuários devem realizar durante o teste. Essas tarefas devem abranger os principais recursos e funcionalidades do produto.

- Execute o teste: Peça aos usuários selecionados que realizem as tarefas definidas enquanto são observados. Encoraje-os a pensar em voz alta e expressar suas opiniões e dificuldades durante o processo.
- Colete feedback e observe: Registre os comentários, observações e comportamentos dos usuários durante o teste. Observe suas ações, reações e expressões faciais para obter insights adicionais.
- Analise os resultados: Revise os dados coletados e identifique os padrões, problemas de usabilidade e oportunidades de melhoria. Priorize as áreas que necessitam de ajustes.

### **6.3. Resultados**

Ao apresentar os resultados obtidos com os testes de usabilidade, é importante destacar os principais insights e descobertas que foram identificados. Aqui estão algumas informações que podem ser abordadas:

- Problemas de usabilidade: Descreva os principais problemas ou dificuldades encontrados pelos usuários durante o teste. Isso pode incluir problemas de navegação, falta de clareza nas instruções, confusão de layout, dificuldade em encontrar informações ou executar tarefas específicas. Forneça exemplos concretos e ilustrativos desses problemas.
- Feedback dos usuários: Apresente os comentários, opiniões e sugestões dos usuários durante o teste.
- Ações corretivas tomadas: Descreva as medidas corretivas que foram adotadas com base nos resultados dos testes de usabilidade.

## **7. Considerações Finais**

Nesta seção, apresentaremos um resumo dos principais pontos do relatório, reafirmando o problema identificado, a solução proposta e discutindo as implicações do trabalho. Além disso, sugeriremos possíveis direções para futuros desenvolvimentos.

## **Referências**