

Design Docs

PARA DEFINIR
PROJETOS DE
SOFTWARE





Cássio Botaro

@cassiobotaro
Engenheiro de Software @ PicPay

A photograph of a young man with short brown hair and glasses, wearing a bright yellow t-shirt, standing on a stage and gesturing with his right hand. He is positioned next to a white projector screen which is displaying a presentation slide. In the foreground, the back of a person's head with long brown hair is visible, suggesting they are in the audience. The background shows a brick wall.

Slides disponíveis em
acesse.one/QL9Ef

01

Definição

02

A estrutura

03

Ferramentas

04

O ciclo de vida

05

Quando não
escrever

Agenda

O que veremos hoje

06

Outros Documentos

07

Referências

Agenda

O que veremos hoje

Definição

Conceito e motivação

Documentos relativamente informais que o autor principal ou autores de um sistema de software ou aplicativo criam antes de embarcar no projeto de codificação.

O Documento de design documenta a estratégia de implementação de alto nível e as principais decisões de design com ênfase nas compensações que foram consideradas durante essas decisões.

**Documentos feito pelo time com o objetivo
de tornar claro e compartilhável todo o
processo de desenvolvimento de uma
solução.**

É documentado o **racional da solução
adotada, **alternativas**, **objetivos**,
arquitetura entre outros pontos.**

Objetivos

- • • • •
- • • • •
- • • • •

A identificação precoce de problemas de design ao fazer alterações ainda é barata.

Alcançar consenso em torno de um projeto na organização.

Assegurar a consideração de preocupações transversais.

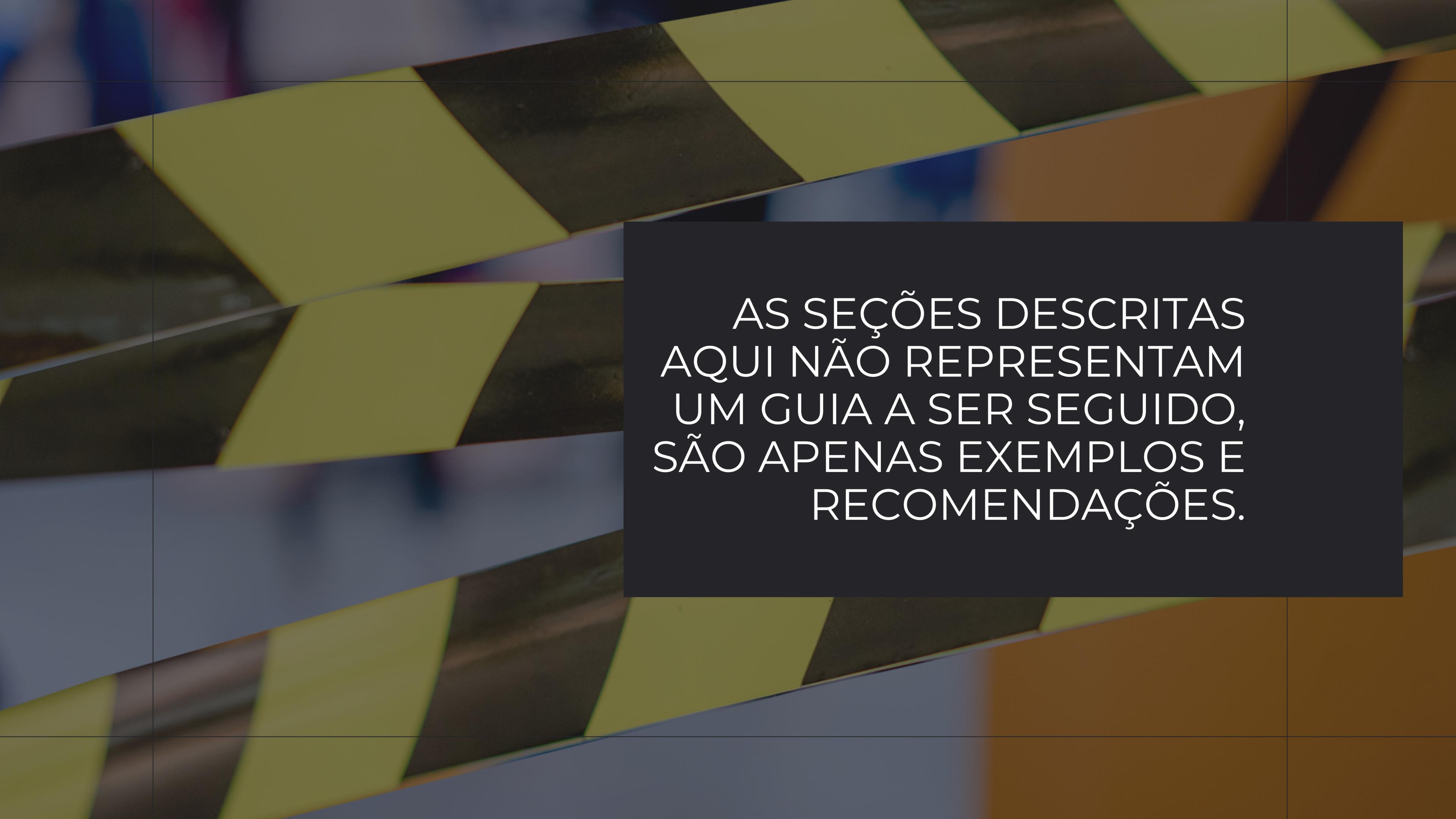
Escalar o conhecimento dos engenheiros seniores na organização.

Forme a base de uma memória organizacional em torno das decisões de design.

Atua como um artefato resumido no portfólio técnico do(s) designer(es) de software.

Estrutura

Como preencher o documento?



AS SEÇÕES DESCritAS
AQUI NÃO REPRESENTAM
UM GUIA A SER SEGUIDO,
SÃO APENAS EXEMPLOS E
RECOMENDAÇõES.

Cabeçalhos

Explicação: Contém o nome das pessoas autoras do documento, nome das pessoas que revisaram o documento, o estado atual do documento (rascunho, em revisão, proposto, aprovado, rejeitado), datas de criação.

Dicas:

- Não deixe de atualizar o estado do documento;
- Envolva e peça revisão para pessoas com maior senioridade de áreas relevantes (segurança, infraestrutura, plataforma etc) ou times envolvidos e impactados;
- Adicionar rótulos (tag) podem ajudar na busca e classificação.

Cabeçalhos

Exemplo:

Design Doc: Sistema de Gerenciamento de Tarefas

Autores: João Silva, Maria Souza

Revisores: Pedro Santos, Ana Oliveira

Data de criação: 2022-08-15

Status: Proposto

Tags: Design Doc, Gerenciamento de Tarefas

Visão Geral

Explicação: Texto breve e de alto nível sobre o que este documento se trata. Visa trazer o que esperar na leitura do documento.

Dicas:

- Tipicamente será um parágrafo ou no máximo dois;
- Não deve conter detalhes;
- Alguém sem contexto deve conseguir entender.

Visão Geral

Exemplo:

Este Design Doc apresenta a criação de um novo backend for frontend para um sistema de vendas online que agora irá oferecer um aplicativo móvel.

Escopo e Contexto

Explicação: Descreve o cenário o qual o sistema está inserido, os motivadores da escrita do documento e outras informações para entendimento do contexto onde o problema está inserido (Tipicamente contém informações de tecnologia atual, dívidas técnicas etc).

O objetivo é que os leitores sejam atualizados, mas algum conhecimento prévio pode ser assumido e informações detalhadas podem ser vinculadas.

Dicas:

- Seja sucinto!
- Não coloque objetivos e soluções propostas;
- Encare como um plano de fundo.

Escopo e Contexto

Exemplo:

O sistema de vendas online da empresa XYZ busca aprimorar a experiência do usuário por meio do lançamento de um novo aplicativo móvel. No entanto, a atual API REST do sistema possui um payload demasiado grande para ser processado em dispositivos móveis.

Objetivos e Fora de Escopo

Explicação: Enquanto os objetivos são requisitos de negócio ou técnicos que serão alcançados com a conclusão da atividade. Fora de escopo são pontos que não serão contemplados ou não são uma preocupação na atividade.
Ambos podem ser em formatos de listas e devem ser objetivos.

Dicas:

- Dê preferência a objetivos metrificáveis;
- Em Fora de escopo não negue apenas os objetivos;
- Seja sucinto, porém objetivo. Ao invés de escrever "Reducir custos" escreva "Reducir custos através da redução do tráfego"

Objetivos e Fora de Escopo

Exemplo:

Objetivos:

- Reduzir o tempo de resposta do sistema de vendas online para melhorar a experiência do usuário.
- Melhorar a escalabilidade do sistema de vendas online para atender a demanda crescente de usuários.
- Aumentar a taxa de conversão de vendas ao oferecer uma experiência de usuário mais rápida e intuitiva.
- Facilitar a manutenção e evolução do sistema de vendas online, isolando o código legado e fornecendo uma camada de abstração para as APIs do backend.

Objetivos e Fora de Escopo

Exemplo:

Fora de escopo:

- A criação de novas funcionalidades principais do sistema de vendas online, que já estão fora do escopo deste projeto.
- A implementação de segurança de alto nível no BFF, que será tratada em outro projeto específico.
- A criação de uma solução de cache, que não será abordada neste projeto e poderá ser implementada posteriormente.

O Design

Explicação:

Dado o contexto (fatos), objetivos e não objetivos (requisitos), mostre por que uma determinada solução atende melhor a esses objetivos.

Comece por dar uma visão geral e depois entre em detalhes.

Pode conter diagramas com contexto e aplicações, API's, os dados que serão manipulados e informações sobre sensibilidade dos mesmos, se necessário mas não comum, códigos e pseudo códigos, telas e payloads que serão trafegados.

Outros diagramas como os de sequência também podem ser úteis.

O Design

Explicação:

Dê foco nas compensações para produzir um documento útil com valor a longo prazo.

As soluções podem estar restritas a um sistema antigo ou biblioteca ou podem ocasionar em soluções não ótimas, nestes casos, foque na melhor solução considerando todas as compensações.

Dicas:

- Utilize **fatos** e **dados** para trazerem valor ao seu design;
- Evite copiar e colar todo um *schema* ou *API*;
- Elementos visuais são complementares e não substitutos de explicações.

O Design

Exemplo:

O objetivo deste projeto é melhorar a experiência do usuário e a escalabilidade do sistema de vendas online existente, adicionando um BFF (Backend For Frontend) para fornecer respostas mais leves e uma experiência mais fluida. Para atender a esses objetivos, propomos a implementação do BFF usando o framework Spring Boot em conjunto com o servidor de aplicativos Apache Tomcat. O BFF será responsável por fornecer uma camada de abstração para as APIs do backend e fornecer respostas mais rápidas para o frontend. Isso resultará em uma experiência de usuário mais rápida e intuitiva, aumentando assim a taxa de conversão de vendas.

...

O Design

Exemplo:

Prós:

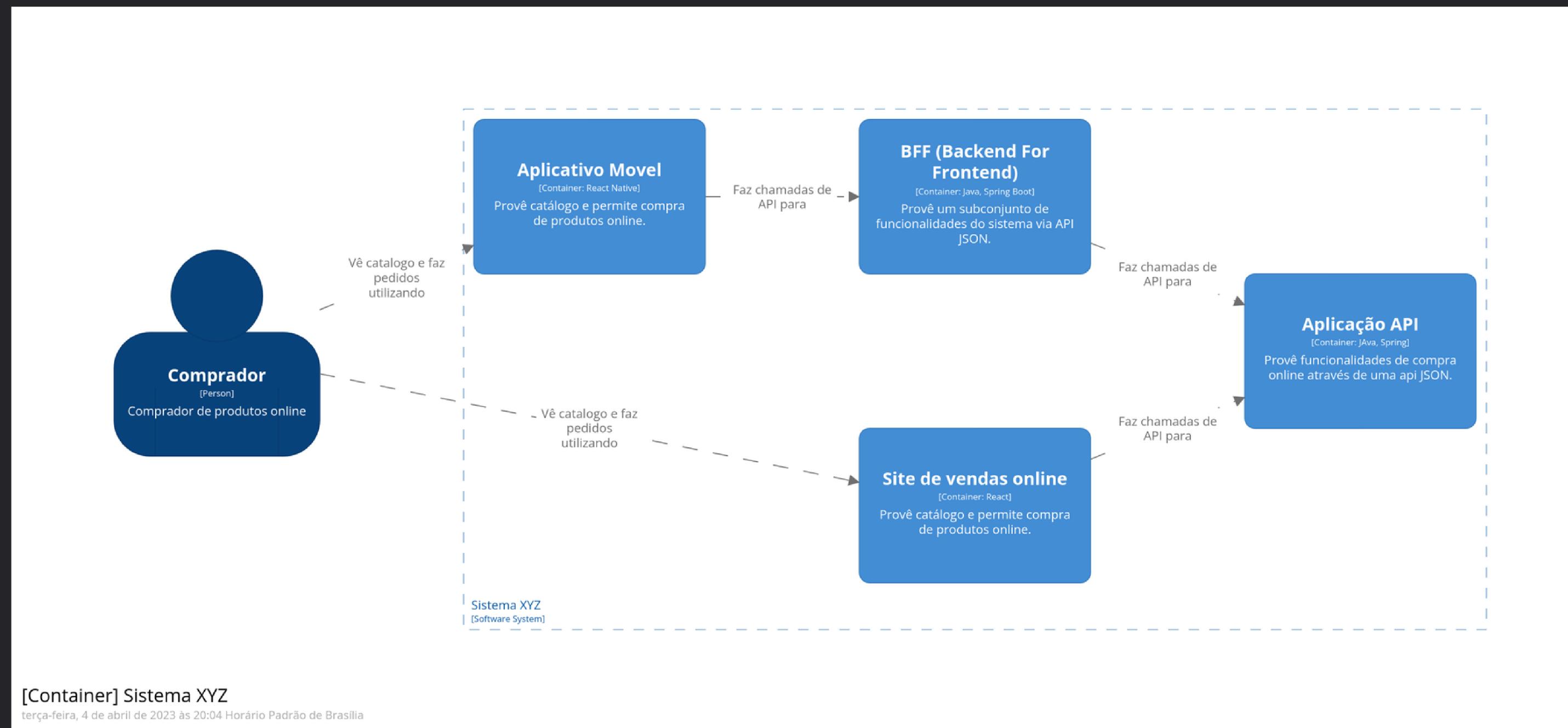
- Melhoria na experiência do usuário com respostas mais leves e uma experiência mais fluida;
- É possível criar APIs específicas para cada componente de frontend;
- Redução de 30% no payload trafegado em média nas requisições.

Contras:

- Maior complexidade na arquitetura do sistema, com a introdução de um novo componente;
- Possível aumento de tempo e custo de desenvolvimento;
- Necessidade de garantir que as APIs do backend estejam bem projetadas para suportar a orquestração do BFF.

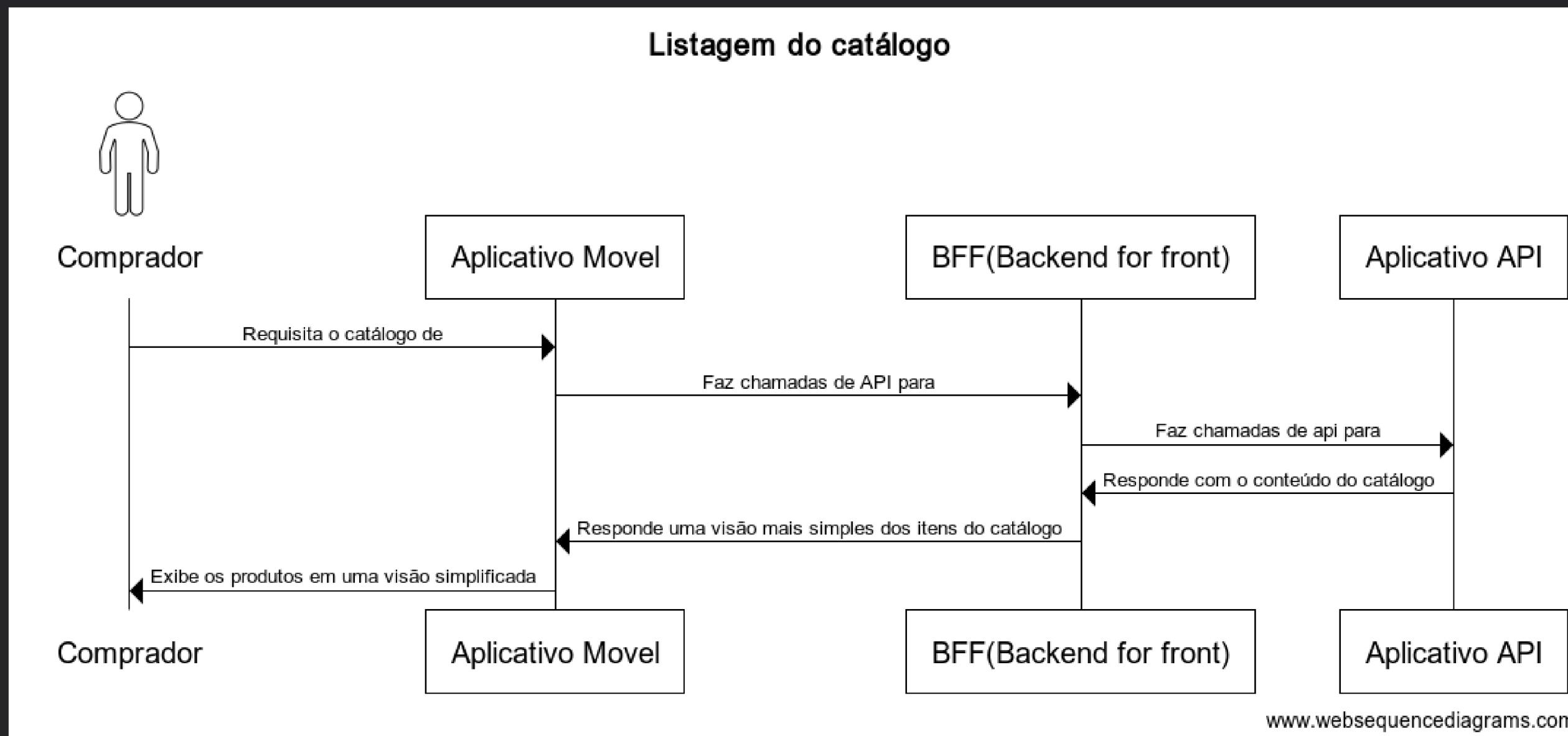
O Design

Exemplo:



O Design

Exemplo:



Alternativas consideradas

Explicação: Os sistemas ou soluções alternativas que foram consideradas para a resolução do problema, explicando compensações e motivações da escolha.

Dicas:

- Uma alternativa sempre a ser considerada é não fazer nada;
- O foco deve estar nas compensações que cada design faz e como isto levou ao tópico principal do documento;
- Seja sucinto, mas destaque o porque a solução escolhida é a melhor opção.

Alternativas consideradas

Exemplo:

Utilização de cabeçalhos HTTP

Uma dessas soluções seria o uso de cabeçalhos HTTP para restringir o conteúdo da resposta enviada pela API, permitindo que apenas as informações necessárias para o funcionamento do aplicativo móvel sejam enviadas. Sua implementação e manutenção é prejudicada pois as alterações futuras seriam complexas.

Compressão de dados

Outra opção seria o uso de técnicas de compressão de dados para reduzir o tamanho do payload enviado pela API. No entanto, essa solução não se mostrou eficaz em testes, reduzindo em apenas 10% o tamanho do payload.

Preocupações transversais

Explicação: Descreva o que pode impactar outras pessoas. Pode ser uma preocupação de segurança, quebra de compatibilidade de API, aumento de fluxo em outros sistemas ou questões relacionadas a infraestrutura.

Está relacionado com a colaboração com outros times que estejam envolvidos ou sejam impactados.

Dicas:

- Envolva os times o mais cedo possível;
- Peça revisões aos times;

Preocupações transversais

Exemplo:

É importante garantir que o BFF seja projetado e implementado de forma segura, com protocolos de criptografia e autenticação adequados para proteger os dados sensíveis dos usuários. Além disso, é necessário garantir a compatibilidade das APIs do backend com o BFF, para que não haja problemas de integração. Por fim, é importante monitorar o aumento no fluxo de dados que o BFF pode gerar, para garantir que a infraestrutura suporte essa carga adicional e que o sistema continue operando de maneira adequada.

Outros

Testabilidade e Observabilidade

Descreva como a solução será testada e como métricas serão criadas para garantir o atingimento do sucesso. Inclua como a solução será observada.

Plano de implantação

Segmentação das entregas e passos necessários para entrega de valor da solução.

Perguntas em aberto

Pontos que ainda não foram definidos ou ainda são desconhecidos.

Ferramentas

Tenha em mão boas escolhas

Ferramentas

Confluence

Wiki corporativo, é uma ferramenta de colaboração da Atlassian que permite a criação de documentos, incluindo Design Docs. É especialmente útil para equipes de desenvolvimento de software, pois oferece integração com outras ferramentas da Atlassian, como Jira e Bitbucket.

[Site](#)

Google Docs

O Google Docs é uma ferramenta online e gratuita que permite a edição colaborativa em tempo real e o compartilhamento fácil do documento com outras pessoas.

[Site](#)

GitHub

Uma plataforma de hospedagem de código fonte que também pode ser utilizada para a escrita de documentos, incluindo Design Docs. O GitHub permite a edição colaborativa de documentos em formato Markdown.

[Site](#)

Ferramentas

PlantUML

Ferramenta que permite criar diagramas de forma textual. Ele permite a criação de diagramas de sequência, casos de uso, C4 entre outros.

[Plant UML Server](#)

Web Sequence

Ferramenta para criação de diagramas de sequência de maneira textual.

[websequencediagrams.com](#)

Structurizr

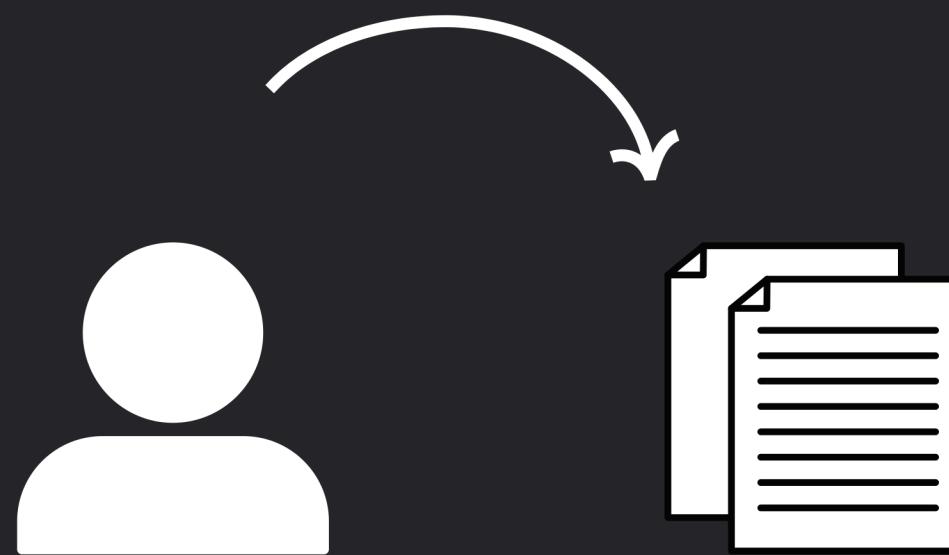
O Structurizr se baseia em "diagramas como código", permitindo que você crie vários diagramas de arquitetura de software a partir de um único modelo.

[structurizr.com](#)

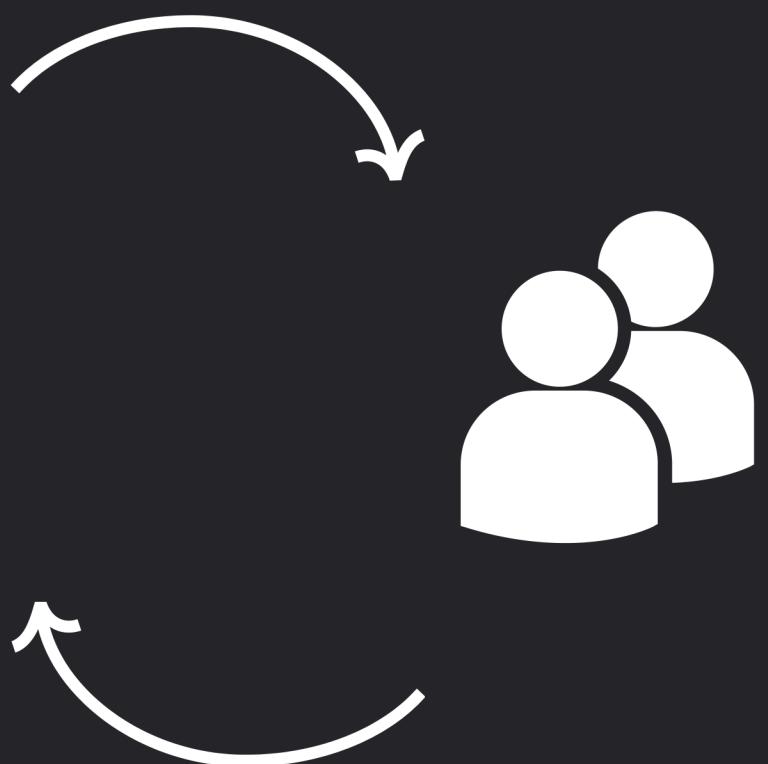
O Ciclo de vida

A documentação viva

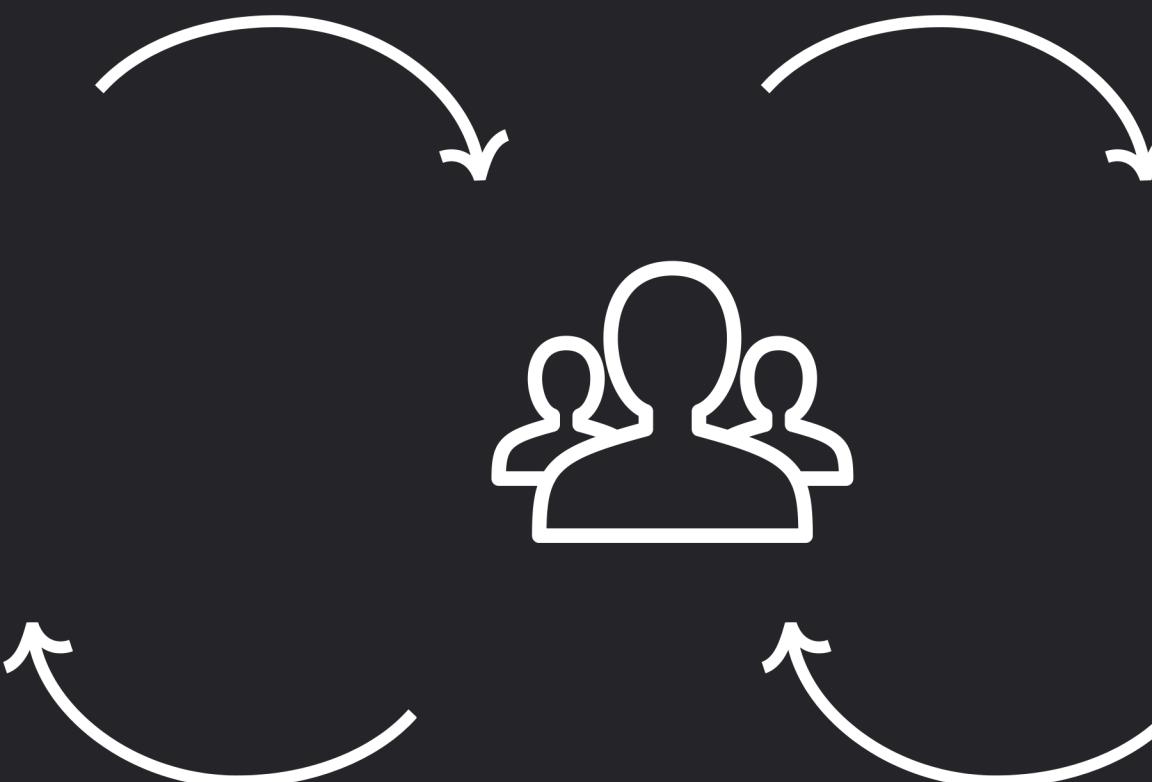
**Escreve um documento sozinho
ou com co-autores**



**Compartilha com os colegas que têm mais
conhecimento sobre o espaço do problema**



**Compartilha com um
público mais amplo.**



Quando confiante, inicie a implementação



**Receba críticas e sugestões
para enriquecer o documento**

**Receba críticas e sugestões
para enriquecer o documento**

**Mantenha o documento vivo
adicionando manutenções e aprendizados**

Quando não escrever

O custo da escrita

Não escreva se...

A cultura organizacional é patologica

Há pouco benefício em escrever documentos que dizem "é assim que vamos implementar", sem alternativas, compensações e explicações.

A sobrecarga no processo é um problema

Escrever documentos geram alguma sobrecarga e sem o entendimento dos benefícios, pode gerar ruído.

Quando não há complexidade

Quando a complexidade da tarefa é inferior a compensação da escrita de um documento.

Outros Documentos

ADRs, RFCs, Tutoriais etc

Alem de Design Docs existem...

ADR

Um registro de decisão de arquitetura (ADR) é um documento que captura uma importante decisão de arquitetura feita junto com seu contexto e consequências.

[ADR - ADR-tool](#)

RFC

Pedidos de comentários (RFC) são documentos técnicos criados por indivíduos e organizações que lidam com tecnologia, com destaque para a Internet Engineering Task Force (IETF).

[RFC - Post Minetto](#)

Tutorial

É um conteúdo baseado em informações iniciais ou primárias sobre determinado assunto, que é ensinado por um tutor ou alguém que possui tutoria ("poder sobre") o assunto.

[Tutorial](#)

Referências

Mais materiais sobre o assunto

Referências

- Design Docs at Google
- Uma introdução aos Design Docs
- How to write an effective design document
- Botaro Cássio no Youtube - Design Docs
- Companies Using RFCs or Design Docs and Examples of These

Obrigado!

\$ Cássio Botaro_

 @cassiobotaro

 @cassiobotaro

 in/cassiobotaro

 CássioBotaro

