

Arquitetura de Soluções para Desenvolvedores Full Stack No- Code/Low-Code

Prof. Bruno Villanova

Arquitetura de Soluções No-Code e Low-Code

- Nesta jornada, vamos explorar como desenhar soluções robustas e escaláveis utilizando ferramentas No-Code e Low-Code, sem abrir mão de um design estratégico e eficiente.

O que é Arquitetura de Soluções?

- A Arquitetura de Soluções é o processo de projetar e definir a estrutura de um sistema para atender aos requisitos de um negócio.
- Envolve: Análise de requisitos, escolha de tecnologias, mapeamento de componentes e definição da integração entre os componentes.

Porque Arquitetura de Soluções é fundamental?

- Organização e Eficiência;
- Escalabilidade;
- Segurança;
- Reutilização;
- Comunicação e Colaboração.

Elementos da Arquitetura de Soluções

- Requisitos de negócio;
- Requisitos Técnicos;
- Componentes do Sistema;
- Dados;
- Segurança;
- Desempenho;
- Escalabilidade;
- Manutenção e Evolução.

Arquitetura de Soluções no Mundo No-Code/Low-Code

- A grande vantagem de usar plataformas No-Code/Low-Code é a rápida criação de protótipos e a implementação de soluções. No entanto, uma arquitetura sólida é crucial para garantir que as soluções construídas sejam escaláveis, seguras e resilientes.

Princípios de Arquitetura para Soluções No-Code/Low-Code

- Modularidade: Divida sua solução em módulos independentes e reutilizáveis. Isso facilita a manutenção, o teste e a atualização.
- Abstração: Utilize ferramentas No-Code/Low-Code para abstrair complexidades e criar interfaces intuitivas.
- Escalabilidade: Planeje sua solução para lidar com o crescimento do volume de dados e usuários.
- Segurança: Implemente medidas de segurança robustas, como autenticação e autorização, para proteger seus dados.
- Manutenção: Facilite a manutenção e atualização da sua solução com uma arquitetura modular e documentação clara.

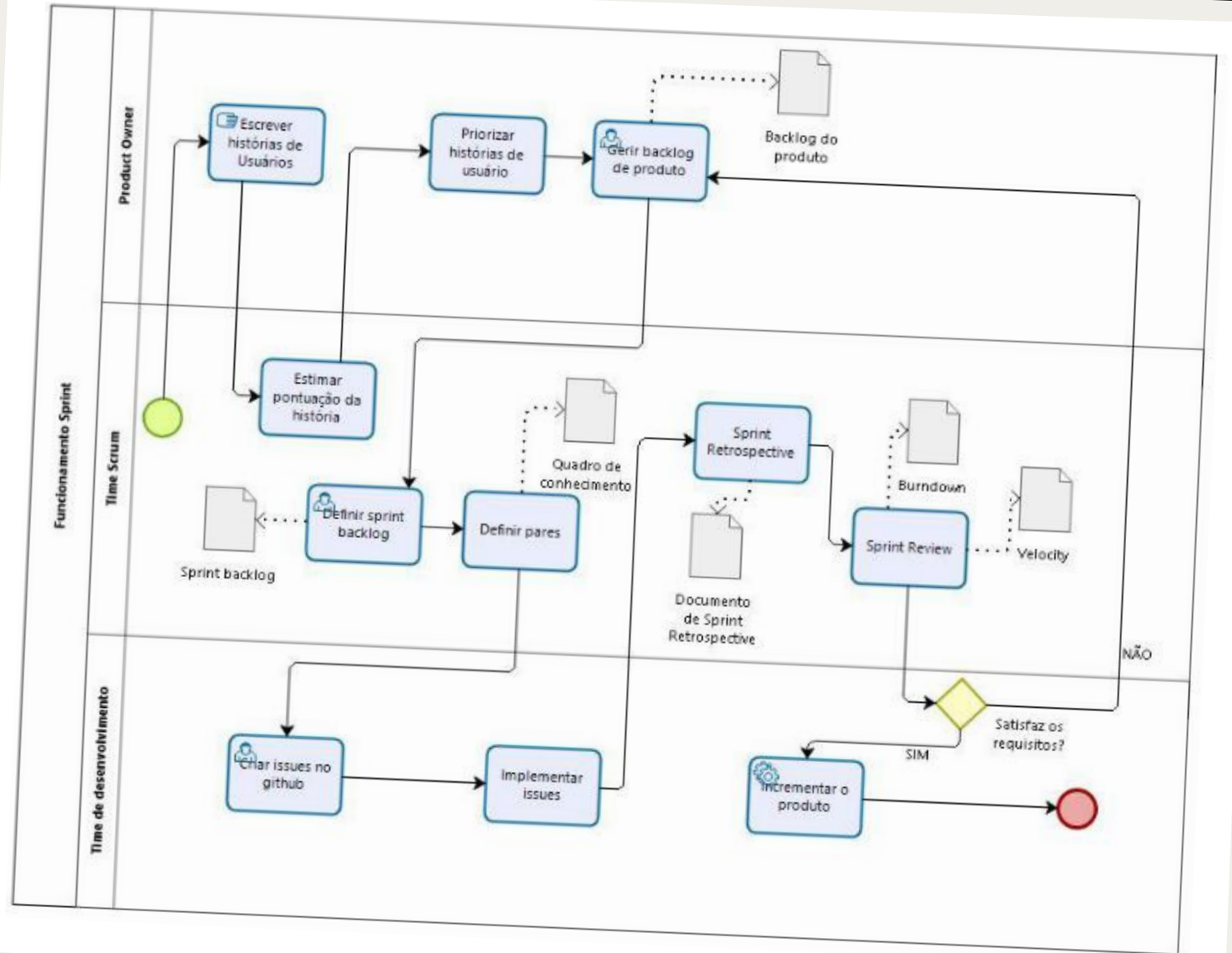
Ferramentas e Modelos

- Diagramas de Arquitetura: Utilize diagramas UML, BPMN ou C4 Model para visualizar a estrutura da sua solução.
- Modelagem de Dados: Utilize ferramentas de modelagem de dados para definir relacionamentos entre entidades e garantir a integridade dos dados.
- Orquestração de Processos: Utilize ferramentas No-Code/Low-Code para automatizar processos e integrar diferentes sistemas.
- API Design: Defina APIs bem documentadas para facilitar a integração com outros sistemas.

Ferramentas de Modelagem - Conceito

- UML (Unified Modeling Language): Permitem representar visualmente a estrutura do sistema, os componentes, as relações e o fluxo de dados.
- BPMn (Business Process Model and Notation): Para modelar os processos de negócio e a interação entre os diferentes componentes do sistema.
- C4 Model: Linguagem visual para descrever a estrutura de software de forma clara e concisa, ideal para comunicar a arquitetura para diferentes stakeholders.
- Modelagem de dados: Utilizando ferramentas como ERDs (Entity-Relationship Diagrams) para representar a estrutura dos dados e suas relações.

Exemplo de Arquitetura de Soluções com fluxo BPMn



Passo a Passo BPMn

- Comece com um modelo;
- Defina o escopo (processo);
- Adicione as atividades (sequencial);
- Projete as categorias(departamento, atores, atividade);
- Use os conectores;
- Legenda.

Exemplo de Arquitetura de Soluções em C4 Model

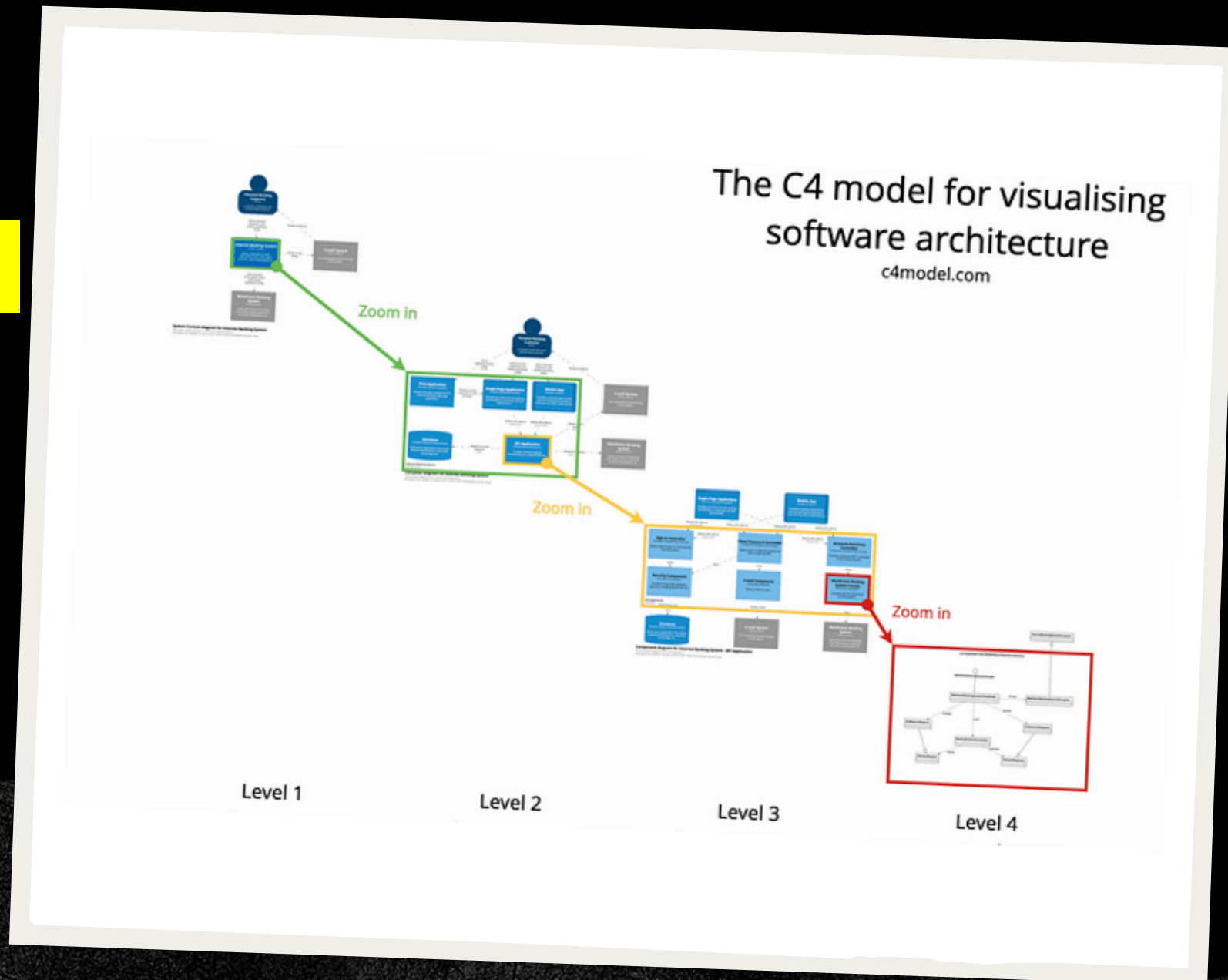
O C4 Model é indicado quando o seu diagrama tem mais de 20 elementos, dividindo-os em camadas C4 para melhor entendimento.

C1 – Contexto | Overview | Continente

C2 – Containers | Overview | Estado

C3 – Componentes | Zoom | Cidade

C4 – Código | Detalhes | Foto da rua



Exemplo de Arquitetura com utilização de ferramentas Low- Code (modelo cloud)

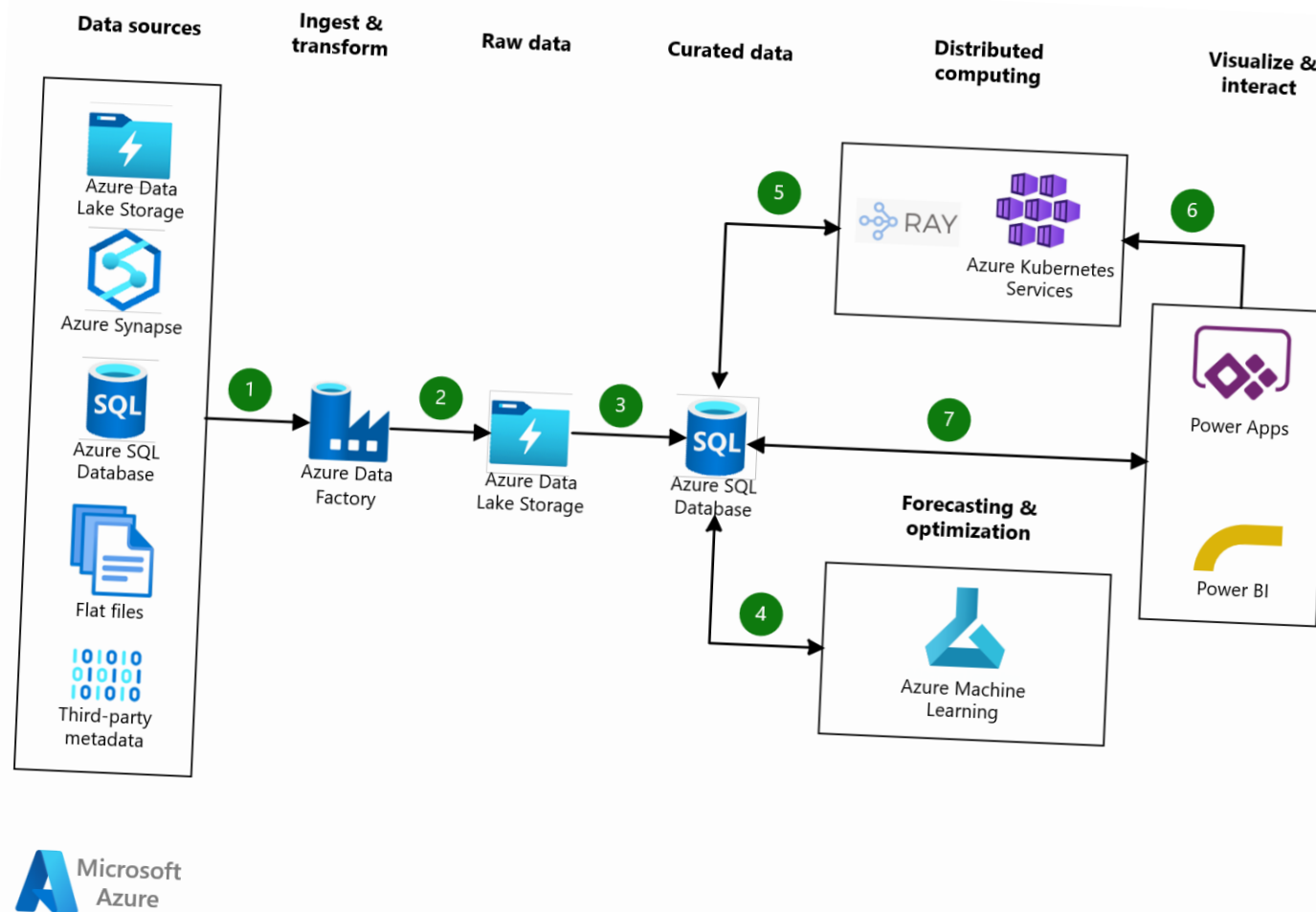
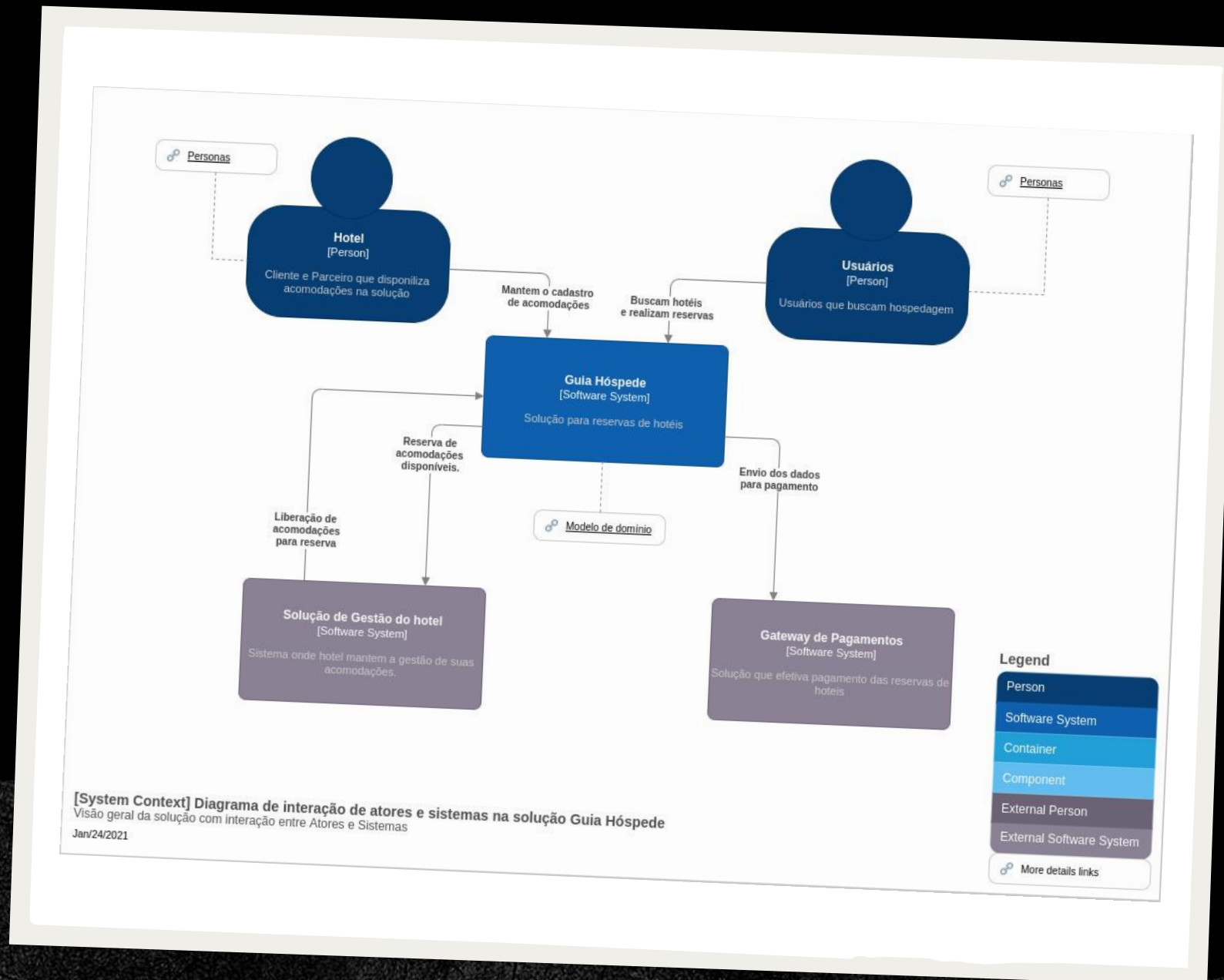


Diagrama de interação



Algumas Ferramentas Interessantes Para Desenho de Arquitetura

Vamos dar uma olhada nas ferramentas mais utilizadas para um Diagrama de Arquitetura de Soluções:

Ferramentas No-Code:

- Miro: Ideal para diagramas colaborativos e brainstorming, ótimo para o início do projeto.
- Lucidchart: Ferramenta completa para diagramas UML, BPMN, fluxogramas e muito mais, permitindo exportar para diversos formatos.
- Draw.io: Integração com Google Drive e Confluence, ideal para diagramas simples e compartilhamento com a equipe.
- Bizagi: O Bizagi é uma plataforma completa de software para automatizar, modelar, executar, monitorar e otimizar seus processos de negócio, aumentando a eficiência, reduzindo custos e melhorando a qualidade.

Ferramentas Low-Code:

- Microsoft Visio: Ferramenta clássica para diagramas de vários tipos, incluindo arquitetura de sistemas.
- Adobe XD: Ferramenta de prototipagem com recursos para diagramas de fluxo de usuários e arquitetura básica.
- Figma: Plataforma de design colaborativa com recursos para diagramas e Wire frames, ideal para visualizar a estrutura do app.

Exemplos Práticos

- Aplicativo de Gerenciamento de Projetos: Modularize o aplicativo em módulos de tarefas, usuários, relatórios e integrações.
- Plataforma de E-commerce: Crie uma arquitetura com módulos para catálogo de produtos, carrinho de compras, checkout e integração com gateways de pagamento.
- Sistema de Automação de Marketing: Use ferramentas No-Code/Low-Code para orquestrar fluxos de trabalho e integrar com plataformas de e-mail marketing e CRM.

Exemplos práticos – Escolha de tecnologia

Ferramenta Low-Code escolhida:

FlutterFlow é uma plataforma low-code que permite criar aplicativos móveis e web sem escrever muito código.

Com o FlutterFlow, você pode criar interfaces de usuário visualmente, conectar-se a bancos de dados e adicionar lógica de negócios.

Ele gera código Flutter nativo, o que é ótimo para desenvolvimento rápido e manutenção futura.

Banco de dados:

Firebase é uma plataforma de desenvolvimento de aplicativos da Google.

Ele oferece serviços como autenticação de usuários, armazenamento em nuvem, banco de dados em tempo real (Firestore) e hospedagem de aplicativos.

Você pode usar o Firebase para gerenciar dados do catálogo de produtos, autenticar usuários e muito mais.

Gateway de pagamentos:

Stripe é um gateway de pagamento popular.

Ele permite que você processe pagamentos com cartões de crédito, débito e outros métodos.

Integre o Stripe ao seu aplicativo para lidar com transações de checkout e pagamentos.

Desafio

- Escolher uma plataforma No-Code/Low-Code e criar um diagrama de arquitetura para um aplicativo simples. Ex: Miro, Draw.io, Lucid, Bizagi, Power Point....
- Desenvolver um sistema de autenticação para um aplicativo usando ferramentas No-Code/Low-Code e descrever os desafios e as melhores práticas de segurança. Ex: Outsystems, Flutterflow, WeWeb....
- Criar um protótipo de um sistema de gerenciamento de estoque para uma loja online, utilizando ferramentas No-Code/Low-Code, e descrever a arquitetura da solução. Ex: Outsystems, Flutterflow, WeWeb...