



SÃO
PAULO
TECH
SCHOOL

Pesquisa e Inovação

Aula 08

Professor Gerson Santos

gerson.santos@sptech.school

Objetivo da Aula

- Feedback da Sprint
- Escopo
- Arquitetura

Bem vindos a Sprint 2

“Não bastar ser, precisa parecer.
Se parecer o que não é, aparece”

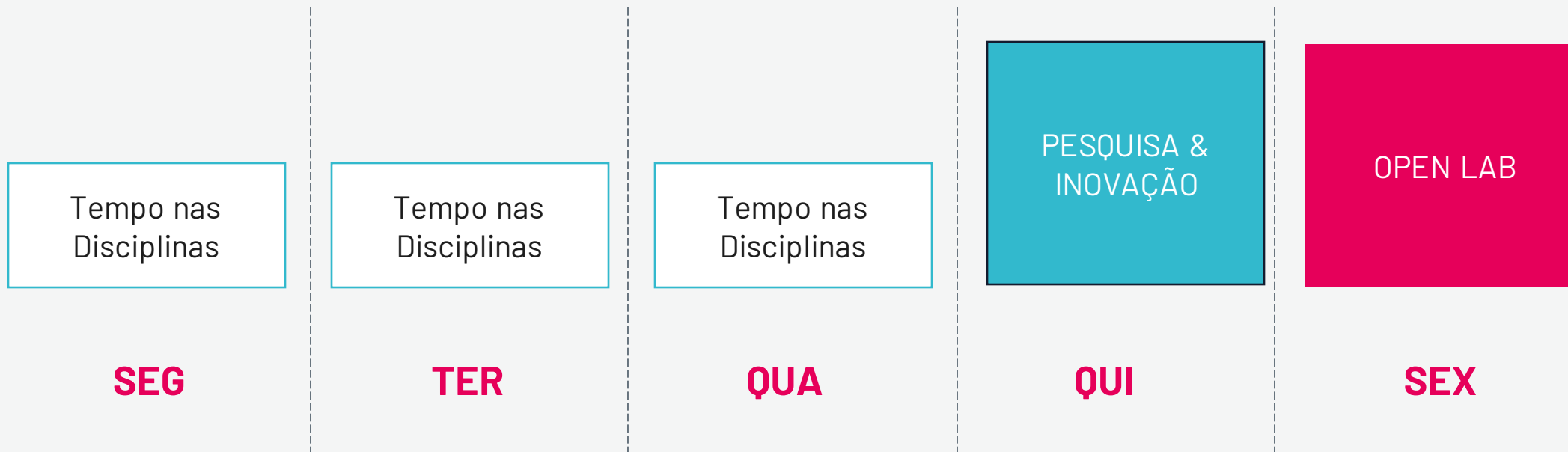
Regras de ouro

- Aluno que **desistir** do Grupo ou do Projeto estará automaticamente **reprovado**;
- O grupo que se **desintegrar ou dividir** estará automaticamente **reprovado**;
- **Se plágio** for detectado, o grupo estará automaticamente **reprovado**;
- **Se não entregar** o *minimum minimorum* do projeto, o grupo estará **reprovado**;

Processo de Trabalho - Sprints



Tempo para o Projeto – 2CCO



LIBERDADE COM RESPONSABILIDADE

Palavras-chave(s):

EXPERIMENTAR & PLANEJAR



S6

S3

DESENVOLVIMENTO

- **Sem spoilers**

- Analytics
- Back-end Finalizado
- Início do Dev. Front-end
- Qualidade

Entrega: 02/06/2022

S2

ESPECIFICAÇÃO & PROTÓTIPOS

- **Entregáveis no Próximo Slide**

- Espec. Funcional & Técnica
- Protótipos
- Métricas
- Back-end

Entrega: 28/04/2022

S1

IDEAÇÃO E ESPECIFICAÇÃO

- **Entregáveis no próximo slide**

- ✓ Definição do Negócio/Ideia
- ✓ Lições Aprendidas
- ✓ Organização & Planejamento
- ✓ Especificação Funcional

Entrega: 17/03/2022

Detalhamento da Sprint 1

SPRINT 1 - ITENS QUE VOCÊS DEVEM FAZER SOZINHOS

- ✓ Ambiente Azure + GitHub
- ✓ Site estático institucional
- ✓ Escopo, Requisitos e BackLog do Produto
- ✓ Diagrama de BD v1
- ✓ Desenho da Solução v1

ITENS QUE VOCÊS INICIAM E PROFESSORES TIRAM AS DÚVIDAS

- ✓ Planner: **Pesquisa e Inovação**
- **Visita (Virtual): Pesquisa e Inovação**
- **Definição da Inovação: Pesquisa e Inovação**

ITENS QUE SERÃO INICIADOS NA SALA DE AULA COM O PROFESSOR

- ✓ Plano de resposta -Lições Aprendidas: **Pesquisa e Inovação**
- ✓ Definição do Negócio (Ideia aprovada): **Design de Interação + PI**
- ✓ Jornada do Usuário e Persona: **Design de Interação**
- ✓ Protótipo do Site Institucional – Alta Resolução: **Design de Interação**
- ✓ Protótipo do Sistema em Baixa Resolução: **Design de Interação**
- ✓ Mapa de Empatia: **Socioemocional**
- ✓ Estrutura do Projeto SpringBoot Funcionando: **Prog WEB**
- ✓ Back-end – Login e Logoff funcionando com Postman: **Prog WEB**
- ✓ Projeto Individual: **Prog WEB + Estrutura de Dados**
- ✓ Classe Abstrata ou Interface no Projeto: **Estrutura de Dados**
- ✓ Padrão Strategy no Projeto: **Estrutura de Dados**

Detalhamento da Sprint 2

SPRINT 1 - ITENS QUE VOCÊS DEVEM FAZER SOZINHOS

- Projetos Atualizados no GitHub
- Requisitos e BackLog do Produto Atualizados
- BPMN
- Diagrama de BD v2
- Planner Atualizado
- Controle da Execução
- Lições Aprendidas
- AWS Básico configurado (Back + BD)

ITENS QUE VOCÊS INICIAM E PROFESSORES TIRAM AS DÚVIDAS

- Pesquisa in Loco (PRESENCIAL): Pesquisa e Inovação
- Definição da Inovação: Pesquisa e Inovação

ITENS QUE SERÃO INICIADOS NA SALA DE AULA COM O PROFESSOR

- ✓ Especificação das Métricas e Dashboard: Pesquisa e Inovação & Design de Interação
- ✓ Desenho Técnico da Solução: Pesquisa e Inovação
- ✓ Funil de Vendas: Design de Interação + PI
- ✓ Jornada do Usuário e Persona: Design de Interação
- ✓ Protótipo do Sistema em Alta Resolução: Design de Interação
- ✓ Testes de Usabilidade: Design de Interação
- ✓ 2 APIS de CRUDS (SpringBoot + ORM): Prog WEB
- ✓ Login/Logoff com ORM: Prog WEB
- ✓ Documentação das APIs (Swagger) com tratativa de erro: Prog WEB
- ✓ Documento de Layout (Entrada/Saída) de Dados: Estrutura de Dados
- ✓ Padrão (Observer, Adapter ou Iterator) no Projeto: Estrutura de Dados
- ✓ Implementação de Lista no Projeto: Estrutura de Dados
- ✓ Exportação em CSV (sem usar componente) : Estrutura de Dados

Pesquisa e Inovação

Aula - Arquitetura

Professor Gerson Santos

gerson.santos@sptech.school

Conceitos que serão utilizados

Vamos pensar em containers (não é Docker), mas pensar que o container é conjunto que precisa estar funcionando ou rodando para um software funcionar.

Exemplos de Containers (Representados por grandes quadrados):

Server-side web application: Aplicação backend. Ex: Spring MVC, NodeJs, Asp.NET MVC, etc.

Client-side web application: A aplicação Javascript que roda no Web Browser. Ex: Angular, JQuery, React.

Client-side desktop application: A aplicação que roda local. Ex: Java JAR, .NET Windows, C++.

Mobile app: Ex: App IOS, App Android, App React Native.

Server-side console application: Ex: "public static void main" application, batch, script.

Microservice: Ex: Spring Boot.

Serverless function: Uma função que independe se servidor. Ex: Amazon Lambda, Azure Function.

Database: Um banco de dados relacional ou de objetos. Ex: MySQL, SQL Server, Oracle Database, MongoDB.C

Diagrama – Visão – Containers

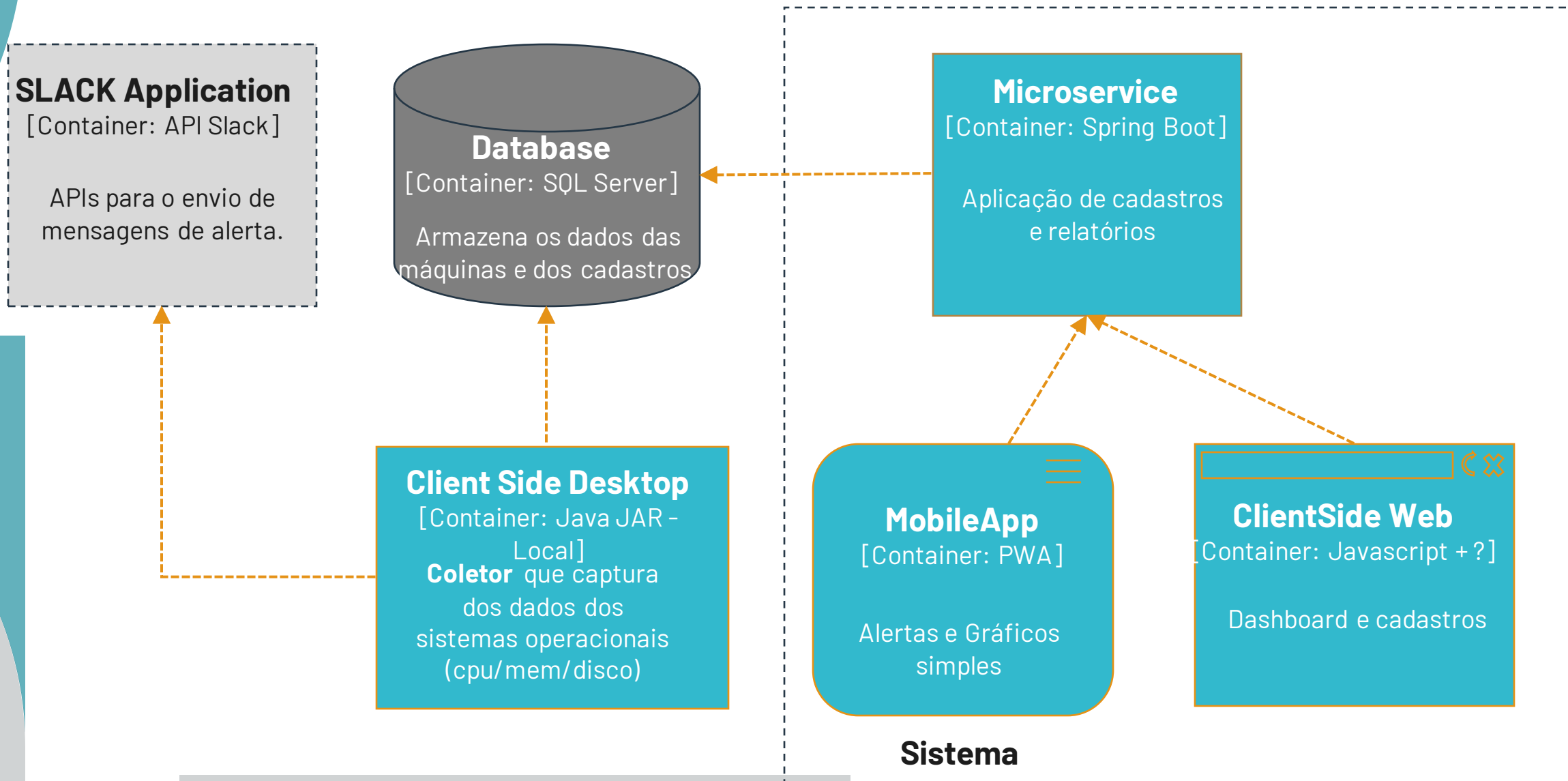


Diagrama – Visão – Containers

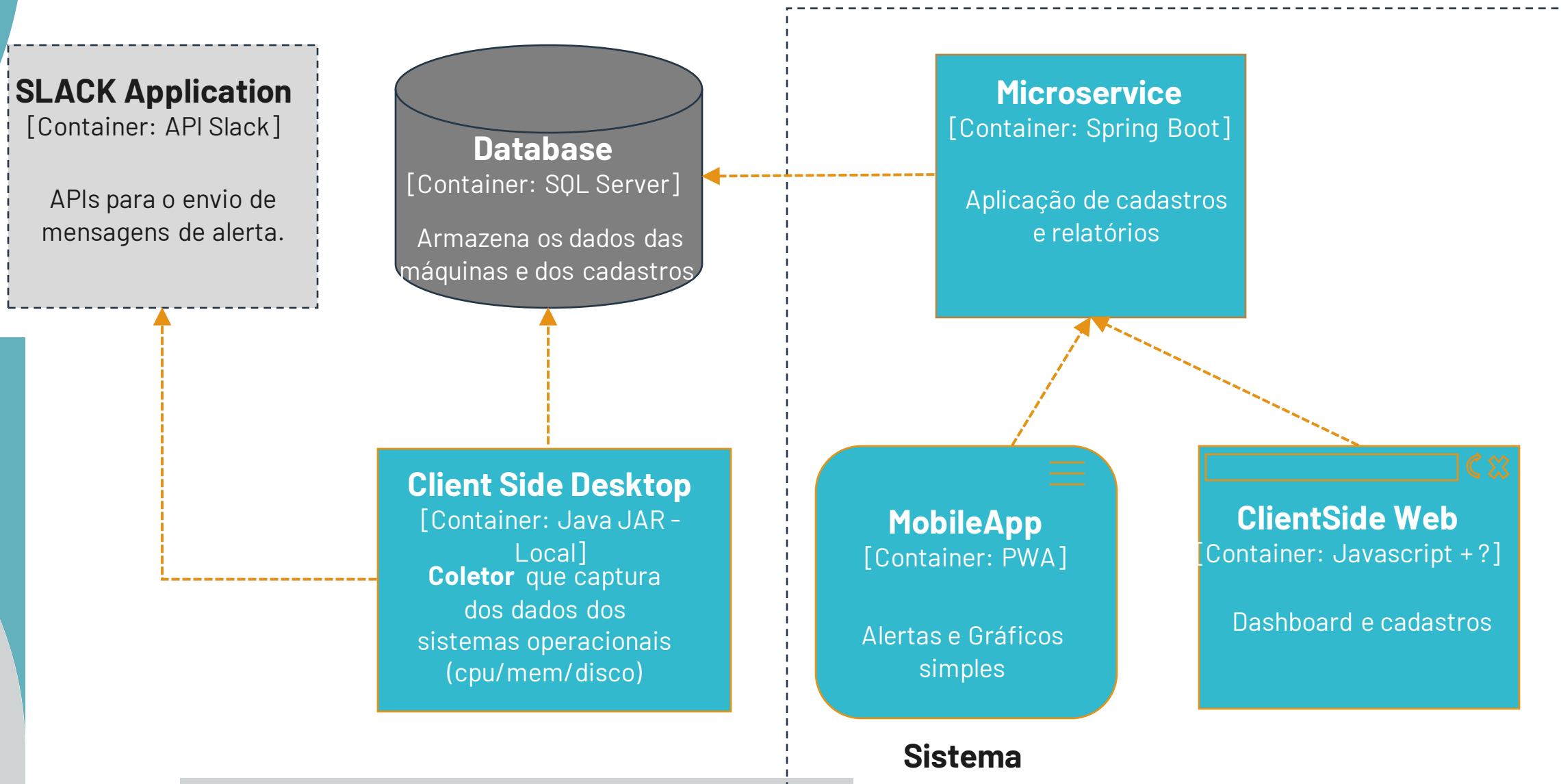


Diagrama – Visão – Containers

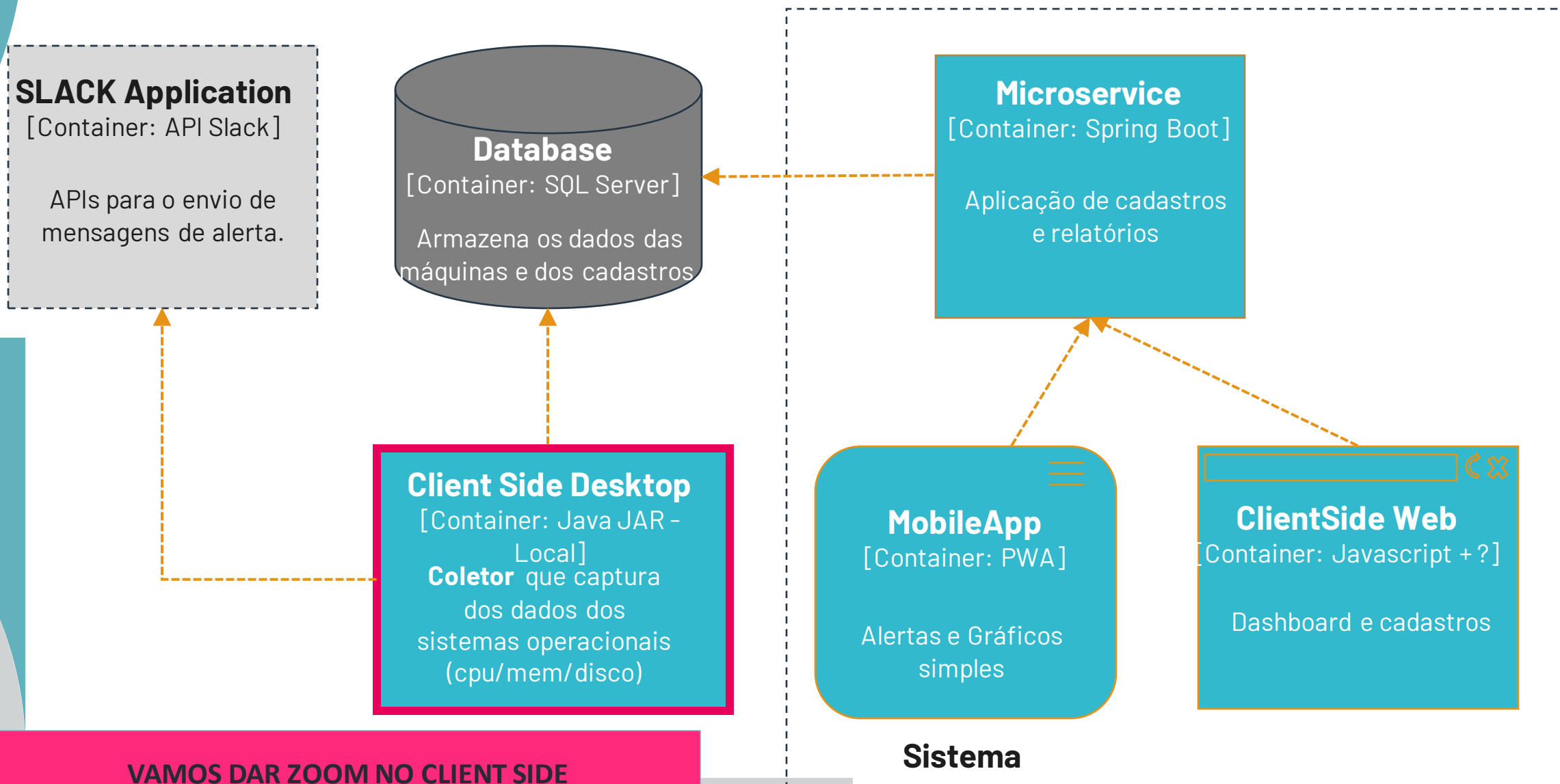
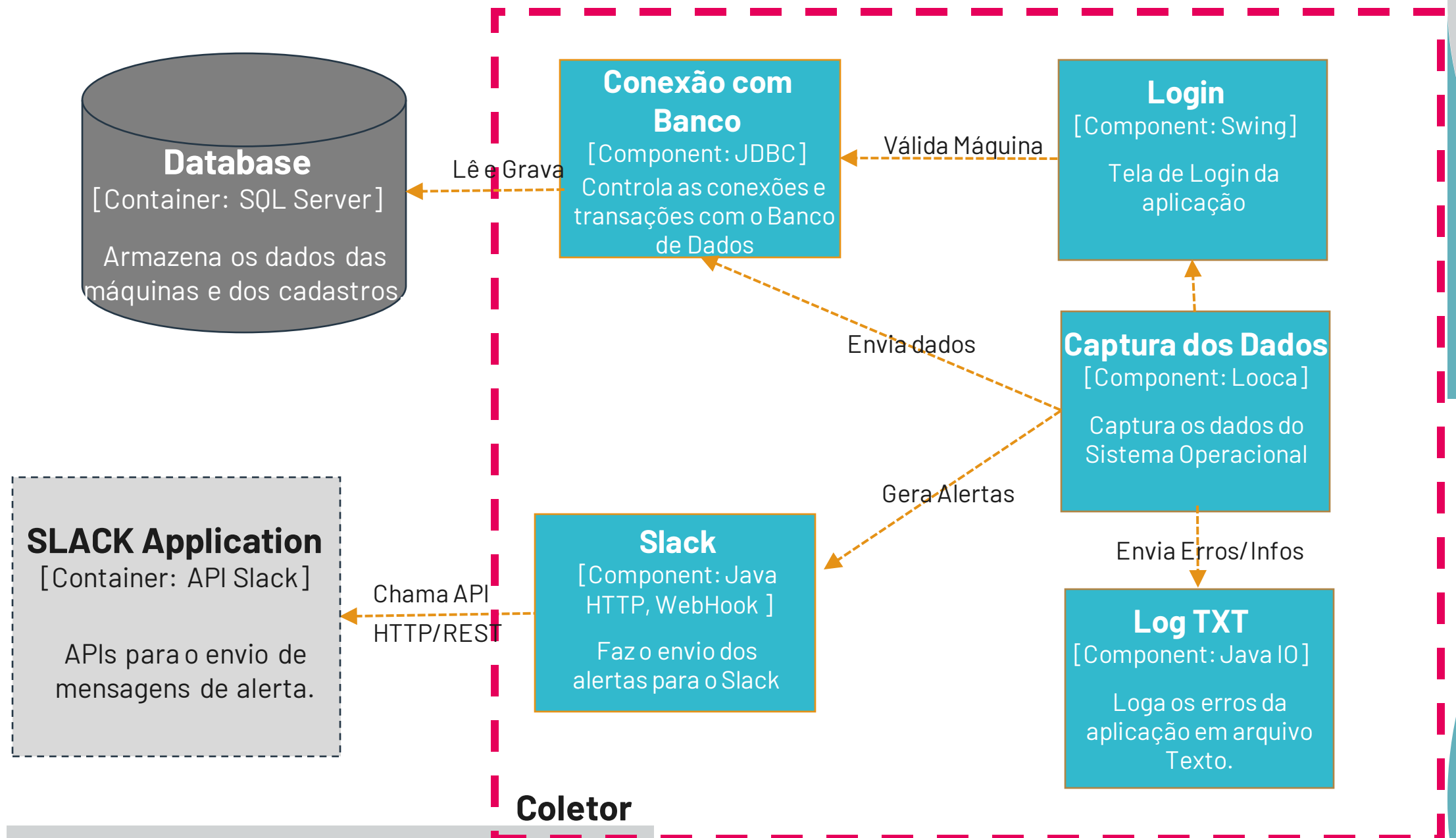


Diagrama – Visão – Componentes – Coletor



Desenho da Solução – Containers

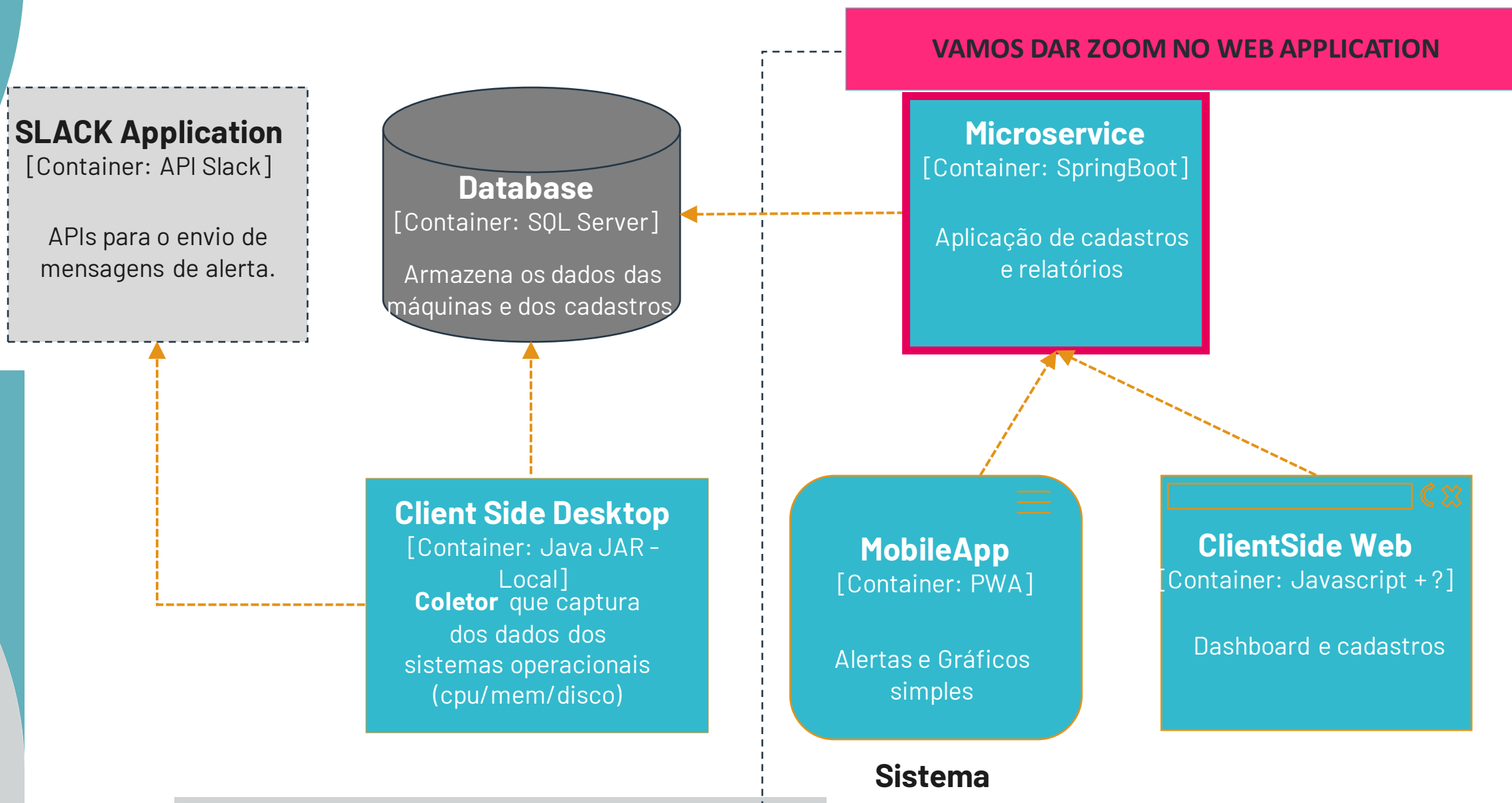


Diagrama – Visão – Componentes – Web Application

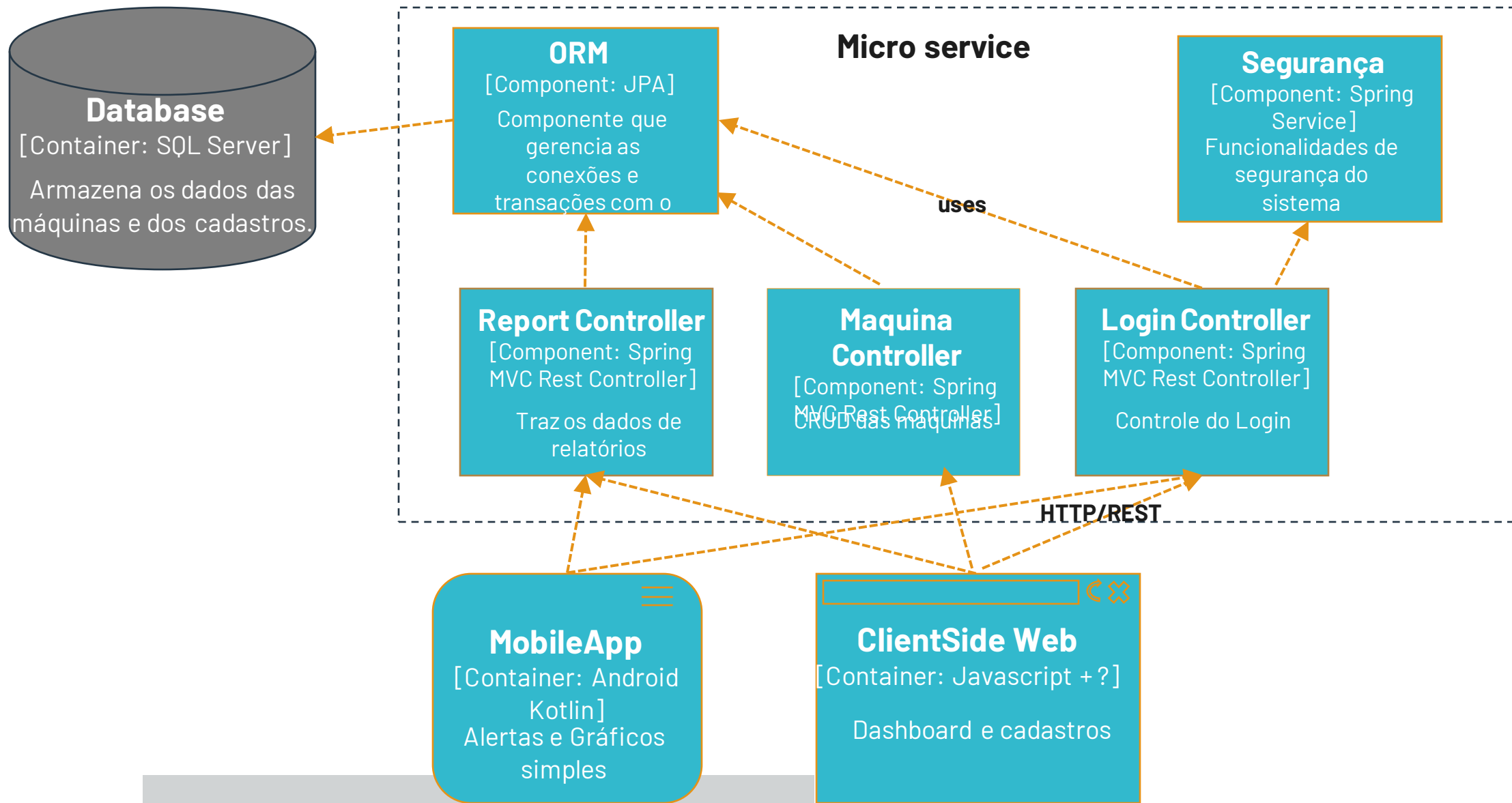


Diagrama – Visão – Componentes – Web Application

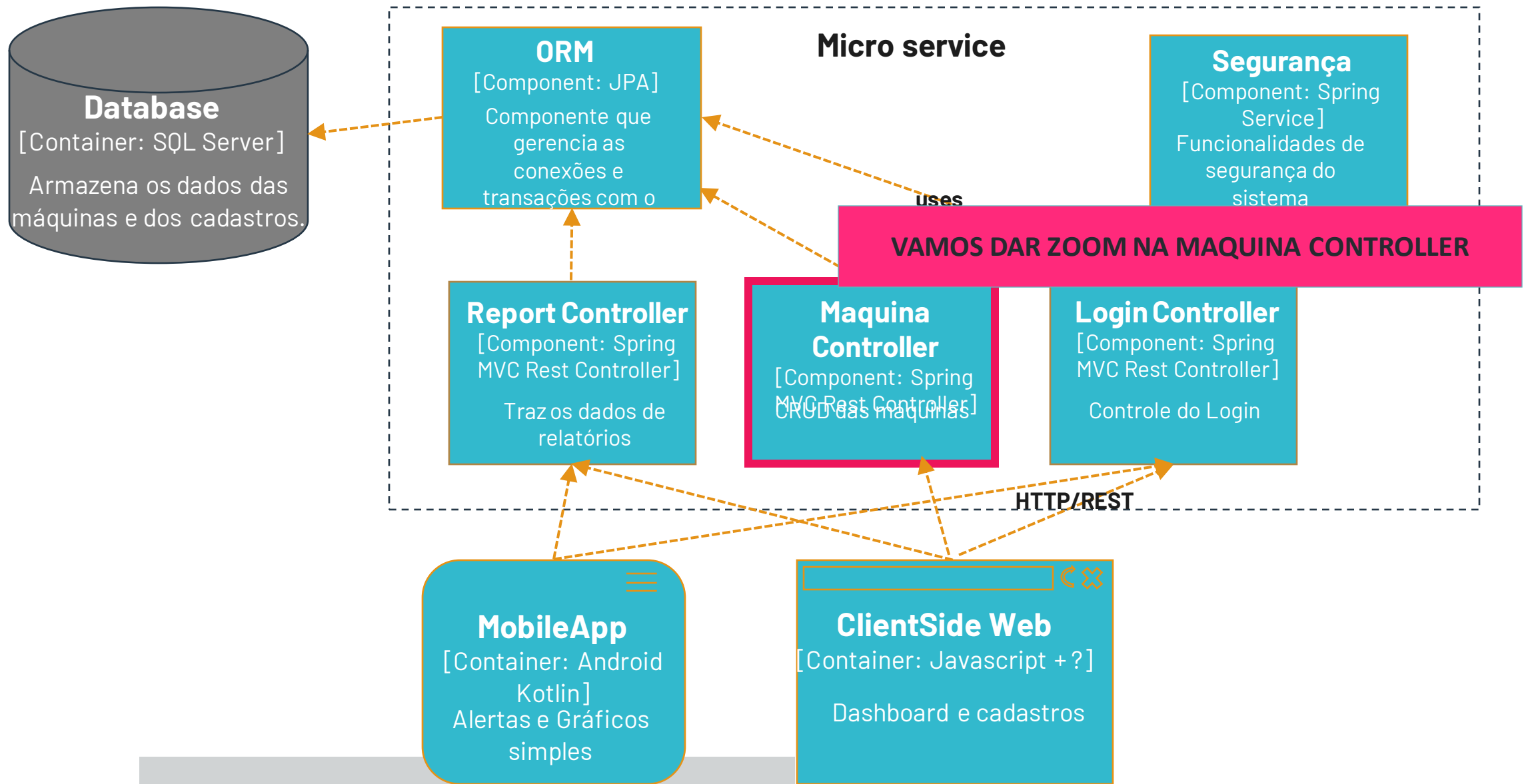
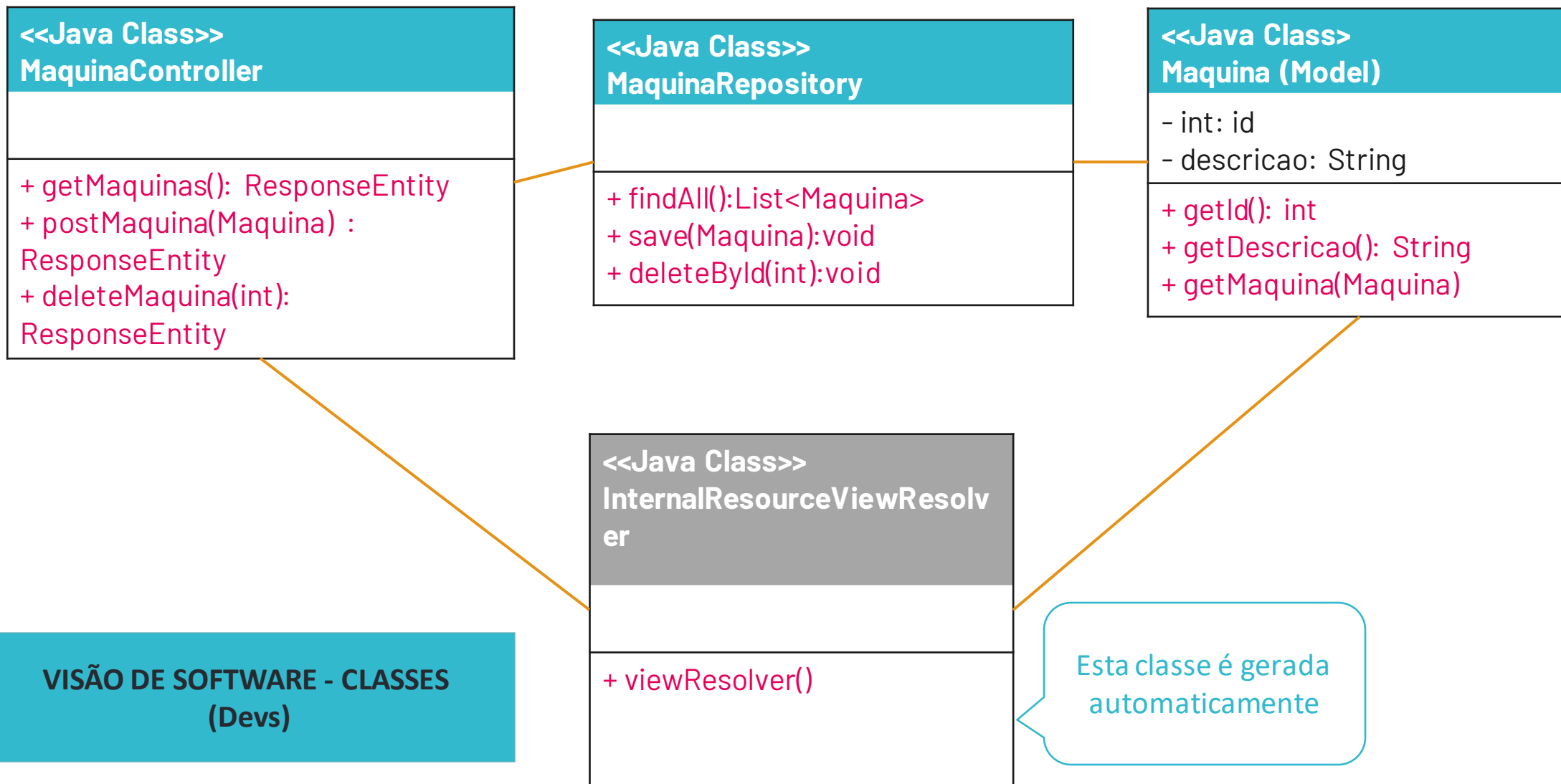


Diagrama de Classes – Maquina Controller



1. Identificar os Objetivos das Arquitetura
2. Cenários Chave
 1. O que é crítico para o negócio?
 2. O que gera alto impacto?
3. Fazer a visão global (overview) da Aplicação
 1. Determinar o tipo da sua aplicação (WEB, Mobile, etc)
 2. Identificar as restrições no desenvolvimento (Rede, Segurança, Sistema Operacional)
 3. Identificar estilos importantes de arquitetura (Camadas, SOA) – Vamos ver mais a frente.
 4. Determinar as tecnologias relevantes (Spring, Node.JS)
4. Desenhar no quadro ou folha de papel
5. Identificar os assuntos chaves (Key Issues: Qualidade, Deploy, Execução, Usabilidade)
6. Cuidar dos itens Transversais (Caching, Comunicação, Autenticação, etc).