

Technical Report - **Product specification**

Space301

Curso: IES - Introdução à Engenharia de Software

Data: Aveiro, 18/12

Estudantes: 110124: Edivaldo Gustavo Castro Luís Bonfim
115697: Shelton Lázio Agostinho
115665: Rafael Jorge Craveiro Miranda Semedo
115637: Giovanni Pereira Santos

Resumo do Projecto: Sistema de Administração de Lançamento de Foguetes

Table of contents:

[1 Introdução](#)

[2 Conceito de produto](#)

[Declaração de Visão](#)

[Personas e Cenários](#)

[Requisitos do Produto \(User Stories\)](#)

[3 Caderno de Arquitetura](#)

[Vista arquitetónica](#)

[Interação dos Módulos](#)

[API](#)

[4 Perspectiva da informação](#)

[5 Referencias e recursos](#)

1 Introdução

O objetivo deste projeto é propor, conceptualizar, e implementar uma aplicação multi-layer, usando uma arquitetura de software baseada em *enterprise frameworks*.

Dessa forma, O **Space301** vai ser desenvolvido para permitir aos engenheiros, operadores e o público em geral acompanhar e analisar em tempo real vários parâmetros do foguete, e também ser avisado quando houver alguma anormalidade nos mesmos parâmetros.

2 Conceito de produto

Declaração de Visão

O **Space301** é uma aplicação de monitoramento de lançamento e trajetória de foguetes que receberá informações do estado físico de foguetes e fornecerá uma interface para exibir ao utilizador em tempo real.

Será utilizado por empresas espaciais tanto em lançamento de foguetes de teste, para a recolha de dados necessários com intuito de facilitar a perceber as medidas corretas, quanto para as missões reais de lançamento para administrar a viagem.

Personas e Cenários

Para o design do desenvolvimento do Space301 foi baseado em sete Personas que representam membros de equipas contratadas por empresas espaciais e serão os principais utilizador do sistema:

1. Miguel Araújo, Engenheiro de Dados

Idade: 35 anos

Cargo: Engenheiro de Dados Aeroespaciais

Responsabilidades: Análise de dados históricos de lançamentos e identificação de padrões de desempenho de foguetes.

Objetivos:

- Ter acesso organizado aos dados históricos de lançamentos para identificar padrões de desempenho e melhorias.
- Obter informações que ajudem a prever possíveis falhas e aprimorar futuras missões.

Necessidades Principais:

- Acesso aos dados históricos com filtros detalhados.
- Ferramentas de análise visual, incluindo gráficos e tabelas comparativas.

2. Daniela, a Engenheira de Missão

Idade: 40 anos

Cargo: Engenheira de Missão

Responsabilidades: Coordenar e analisar o desempenho dos foguetes em tempo real.

Objetivos:

- Obter dados em tempo real para tomar decisões rápidas durante o lançamento e em operações críticas.
- Monitorar a integridade dos componentes e estados da tripulação para manutenção preditiva e ajustes imediatos.

Necessidades Principais:

- Interface de monitoramento em tempo real dos parâmetros dos foguetes.
- Alertas automáticos em casos de discrepâncias nos dados.

3. Eduardo, Administrador de Sistemas

Idade: 45 anos

Cargo: Administrador de Sistemas

Responsabilidades: Gerenciamento da segurança e acesso aos dados críticos de lançamento.

Objetivos:

- Garantir que os dados confidenciais estejam acessíveis apenas a usuários autorizados.
- Proteger as operações contra acessos indevidos e manipulação não autorizada.

Necessidades Principais:

- Ferramenta de gerenciamento de usuários com controle de permissões por nível.

4. Roberto, Gerente de Projetos

Idade: 50 anos

Cargo: Gerente de Projetos Espaciais

Responsabilidades: Análise financeira dos lançamentos e otimização dos custos.

Objetivos:

- Avaliar o desempenho financeiro dos lançamentos para identificar áreas de melhoria.

- Comparar custos entre lançamentos e analisar tendências financeiras ao longo do tempo.

Necessidades Principais:

- Relatórios financeiros com gráficos detalhados de gastos por missão.

5. Dr. Fernando, Médico da Missão

Idade: 38 anos

Cargo: Médico da Missão Espacial

Responsabilidades: Monitorar a saúde e segurança da tripulação durante o lançamento.

Objetivos:

- Acompanhar sinais vitais da tripulação para garantir condições seguras durante a missão.
- Agir rapidamente em caso de qualquer anomalia nos sinais vitais.
- Analisar as condições internas do foguete para preparar a condição física da tripulação.

Necessidades Principais:

- Monitoramento contínuo dos sinais vitais com alertas em casos de anomalias.

6. João, da Equipe de Comunicação

Idade: 33 anos

Cargo: Técnico de Comunicação Espacial

Responsabilidades: Manter a qualidade da comunicação entre foguete e base durante lançamentos e missões.

Objetivos:

- Garantir uma comunicação estável entre o foguete e a base.
- Acompanhar a qualidade do sinal para prevenção de quedas de comunicação.

Necessidades Principais:

- Exibição da qualidade de sinal com alertas em caso de quedas.
- Interface simples para visualização do estado do sinal em tempo real.

7. Mariana, da Equipe de Emergências

Idade: 42 anos

Cargo: Técnica de Emergências Espaciais

Responsabilidades: Monitorar e responder a situações de emergência durante lançamentos

e missões.

Objetivos:

- Receber avisos em tempo real sobre qualquer parâmetro fora do padrão.
- Acessar rapidamente detalhes sobre possíveis anomalias para elaborar planos de resposta.

Necessidades Principais:

- Definir no sistema quais valores dos parâmetros devem ser alertados.
- Sistema de alerta automática.
- Interface que forneça informações detalhadas sobre parâmetros anômalos.

Por último há uma Persona representando o público que acompanha os lançamentos de foguetes:

8. Persona: Manuela, o Público Entusiasta de Tecnologia Espacial

Idade: 25 anos

Cargo: Curiosa e Apreciadora de Lançamentos Espaciais.

Objetivos:

- Acompanhar lançamentos de forma divertida e educativa.
- Acessar informações visuais e atualizações em tempo real sobre os parâmetros dos foguetes

Requisitos do Produto (User Stories)

1. Acesso aos dados históricos dos lançamentos

User Story:

Como engenheiro de dados, quero acessar os dados do histórico dos lançamentos anteriores, para que possa analisar o desempenho dos foguetões e detectar padrões que possam melhorar próximos lançamentos.

Critérios de Aceitação:

Acesso aos dados em formato JSON.

Filtros por missão, tipo de foguete e parâmetros específicos.

Ferramentas de visualização de padrões em gráficos e tabelas.

2. Dados do foguete em tempo real

User Story:

Como engenheiro da missão, quero realizar análises do foguete em tempo real, como parâmetros, nível de integridade e estados da tripulação/carga, para que possa ser feita a análise do desempenho ou para uso na manutenção.

Critérios de Aceitação:

Visualização em tempo real dos principais parâmetros e estados.

Alertas automáticos em caso de discrepância em dados.

Acesso a dados históricos e em tempo real para comparação.

3. Gerenciamento de permissões dos usuários

User Story:

Como administrador de sistemas, quero gerenciar e definir permissões dos usuários, para que apenas as pessoas autorizadas possam ver e modificar os dados do lançamento.

Critérios de Aceitação:

Interface para gerenciamento de usuários e permissões.

Definição de permissões por grupo (Admin, Operador, Visualizador).

4. Acompanhamento de lançamentos para o público

User Story:

Como público, quero acompanhar os lançamentos dos foguetes através de gráficos que mostrem os parâmetros, para entender mais sobre os lançamentos e entretenimento.

Critérios de Aceitação:

Gráficos em tempo real com informações como velocidade, altitude e combustível.

Interface simples e acessível.

Atualizações automáticas durante o lançamento.

5. Estatísticas financeiras de lançamentos

User Story:

Como gerente de projeto, quero traçar estatísticas de lançamentos com base nos gastos e perdas, para fornecer feedback financeiro.

Critérios de Aceitação:

Relatório de gastos por missão, com gráficos detalhados.

Comparação de custos entre lançamentos e identificação de tendências.

6. Monitoramento de sinais vitais em tempo real

User Story:

Como médico de missão, quero monitorar em tempo real os sinais vitais da tripulação, para ver se as condições atuais da nave encontram-se estáveis.

Critérios de Aceitação:

Exibição contínua dos sinais vitais (frequência cardíaca, pressão arterial, etc.).

Alertas automáticos em caso de sinais fora do padrão.

Histórico de sinais vitais com comparação a períodos anteriores da missão.

7. Monitoramento do estado de sinal entre foguete e base

User Story:

Como equipe de comunicação, quero monitorar o estado de sinal entre o foguete e a base, para garantir que a comunicação permaneça estável.

Critérios de Aceitação:

Exibição da qualidade do sinal em tempo real.

Alertas automáticos sobre possíveis quedas de comunicação.

8. Receber dados de avisos para emergências

User Story:

Como equipe de resposta a emergências, queremos receber os dados dos avisos quando os parâmetros estão fora do padrão, para poder elaborar as medidas necessárias e comunicar aos tripulantes.

Critérios de Aceitação:

Sistema de alerta que notifica automaticamente a equipe de emergência.

Detalhamento de parâmetros anômalos e causas potenciais.

Acesso rápido ao histórico de alertas para diagnóstico.

9. Monitoramento do primeiro estágio do foguete

User Story:

Como engenheiro da missão, quero monitorar o “primeiro estágio” que vai ser separado do

foguete principal, para ver se este pode ser reutilizado para futuros lançamentos.

Critérios de Aceitação:

Monitoramento da integridade do primeiro estágio após a separação.

Localização em tempo real e status do estágio.

Registro de histórico de utilizações.

3 Caderno de Arquitetura

Principais requisitos e restrições

A arquitetura definida é influenciada por alguns requisitos e restrições do sistema:

- A web-app deve estar sempre consistente e atualizada;
- Deverá ser possível, e em tempo real, observar os dados do foguete;
- Alguns dados do utilizador devem estar encriptados uma vez que são confidenciais;
- A base de dados deve permanecer acessível no intuito de receber pedidos de dados específicos a qualquer momento;
- O sistema deverá conseguir fornecer estatísticas sobre o lançamento/foguete;
- O sistema deve ser capaz de receber um grande fluxo de dados provenientes do MessageBroker;
- O sistema deve ser rápido a emitir notificações ao utilizador sobre eventuais parâmetros fora do padrão

Vista arquitetónica

O sistema é atualmente dividido em 5 componentes:

- **Gerador de dados:** Para criar os dados necessários para o sistema, um script Python para geração aleatória das diferentes informações. Este script irá simular o comportamento de um lançamento real:

- Velocidade (km/h)
- Aceleração (m/s^2)
- Quantidade de combustível (kg)
- Taxa de combustão (kg/s)
- Temperaturas dos motores ($^{\circ}C$)

- Níveis de oxigênio (combustão e tripulação) (mol/m^3)
- Energia elétrica (gerada e armazenada) (kWh)
- Mantimentos (para tripulação) (kg)
- Qualidade do sinal transmitido (dB)
- Temperatura interna e externa da nave ($^{\circ}\text{C}$)
- Pressão interna e externa da nave (atm)
- Altitude da nave (em relação ao nível do mar) (m)
- Dimensão da nave (m)
- Força G no interior da nave (G)
- Nível de radiação (Sv)
- Pressão arterial (mmHg)
- Níveis de oxigênio no sangue (mol/m^3)
- Frequência cardíaca (bpm)

- **Message Broker:** Irá desempenhar o papel de receber os dados provenientes do gerador de dados. Essas informações serão posteriormente consumidas pelo backend. Para o nosso sistema decidimos utilizar RabbitMQ para facilitar esse processo.

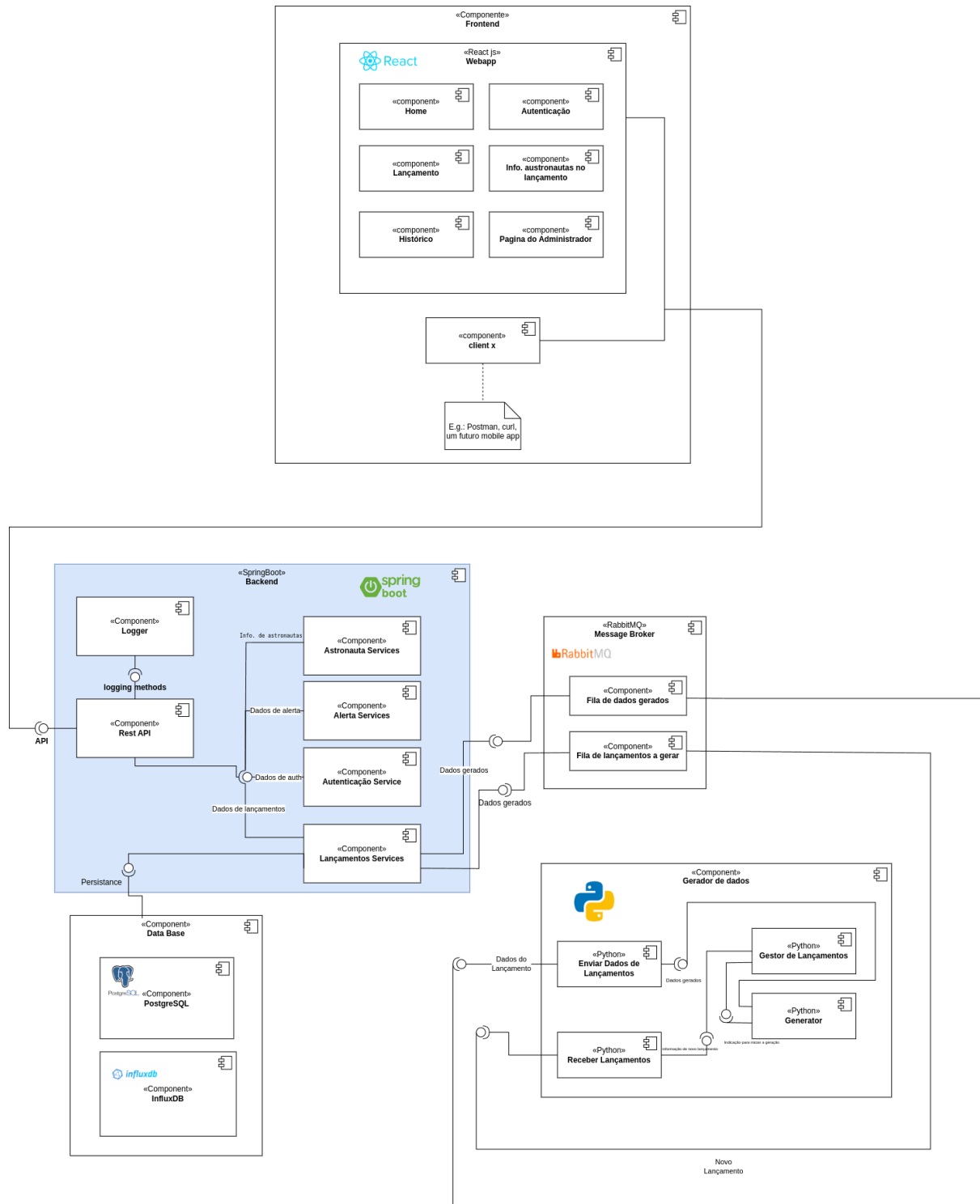
- **Base de dados:** Uma base de dados PostgreSQL para armazenar integralmente os dados estáticos do sistema. E uma base de dados InfluxDB para armazenar os dados dinâmicos do lançamento que serão acessados no histórico.

- **Backend:** Será composto por uma RESTapi com vários endpoints para acesso de dados como: dados gerais do foguetão, dados da tripulação e entre outras informações que serão disponibilizadas, autenticação do staff, websockets e a parte da lógica de negócios e processamento dos dados que virão do MessageBroker. A tecnologia que iremos utilizar é Springboot.

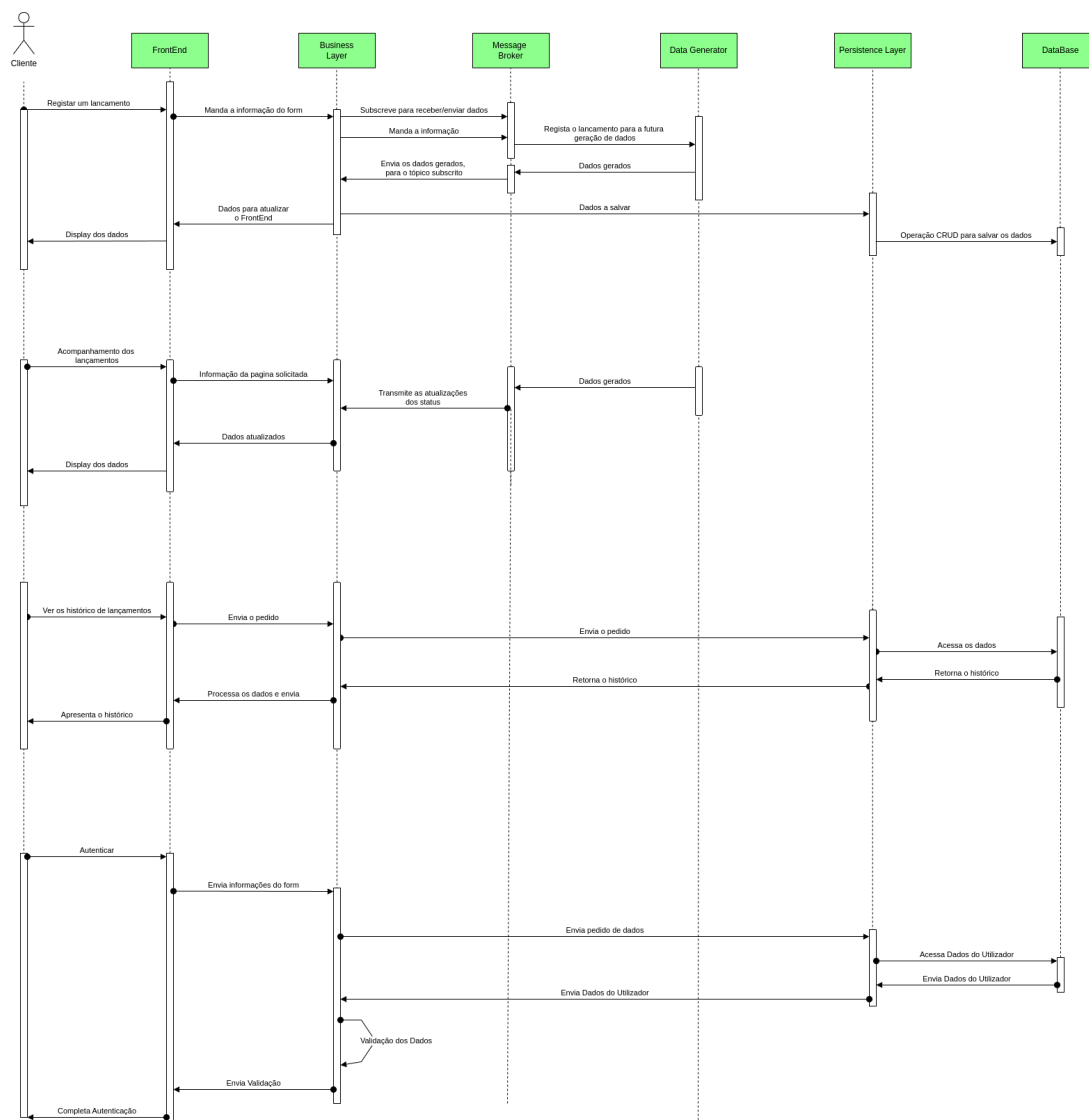
- **Persistência:** Um driver Spring Boot JPA que é usado para se conectar com a base de dados PostgreSQL com o propósito de oferecer uma interface que cria um mapeamento para objectos em Java.

- **Frontend:** Uma interface intuitiva destinada aos utilizadores, proporcionando a criação de lançamentos e a visualização abrangente de informações cruciais. Que por sua vez inclui o acompanhamento de todo o estado do foguete desde o seu lançamento e apresentação de estatísticas detalhadas, como a velocidade e a temperatura. Optamos por implementar o frontend com React usando Vite com TypeScript e css e TailwindCss.

Arquitetura:



Interação dos Módulos



Os dados gerados no gerador de dados serão enviados para um tópico no broker, e o backend por sua vez recebe esses dados, processa, armazena o que for necessário armazenar e enviar para o frontend.

O frontend interage com o backend principalmente pelas APIs, por exemplo quando o registo de um novo lançamento é feito no frontend os dados são enviados para o backend

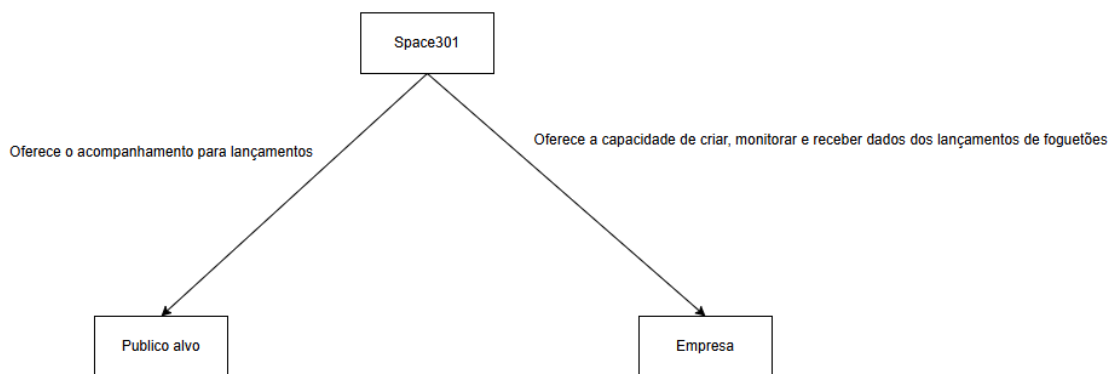
para processamento e depois serem colocados em um tópico no message broker que posteriormente serão utilizados pelo gerador de dados, quando um staff pretende fazer autenticação os dados são introduzidos no frontend e passados para o backend e passam pela autenticação em que os dados serão comparados com os que estão na base de dados.

Todas as operações CRUD na base de dados serão feitas pela camada de persistência, e não serão feitas diretamente pela business layer.

API

- A documentação da api está no **documentation/documentacao.html**

4 Perspectiva da informação



5 Referencias e recursos

- <https://www.codecademy.com/>
- <https://chatgpt.com/>
- <https://spring.io/>
- <https://www.baeldung.com/>
- <https://www.thymeleaf.org/>
- <https://lucide.dev/>
- <https://www.figma.com/>
- <https://dev-mus.medium.com/how-to-deploy-a-vite-react-app-using-nginx-server-d7190a29d8cd>
- <https://app.diagrams.net/>
- <https://code.visualstudio.com/>
- <https://spring.io/projects/spring-boot>