



PROJETO DESKDATA

Arquitetura e Desenvolvimento do Sistema

AUGUSTO BUIN

CAIO VITOR DIAS

FELIPE AUGUSTO GRACIANO

JÚLIO DE PAULA MACHADO

VALDERI DOUGLAS CAMARGO

Fatec Jessen Vidal

Grupo DeskData - 6º Semestre

Desenvolvimento de Software e Multiplataforma

2025

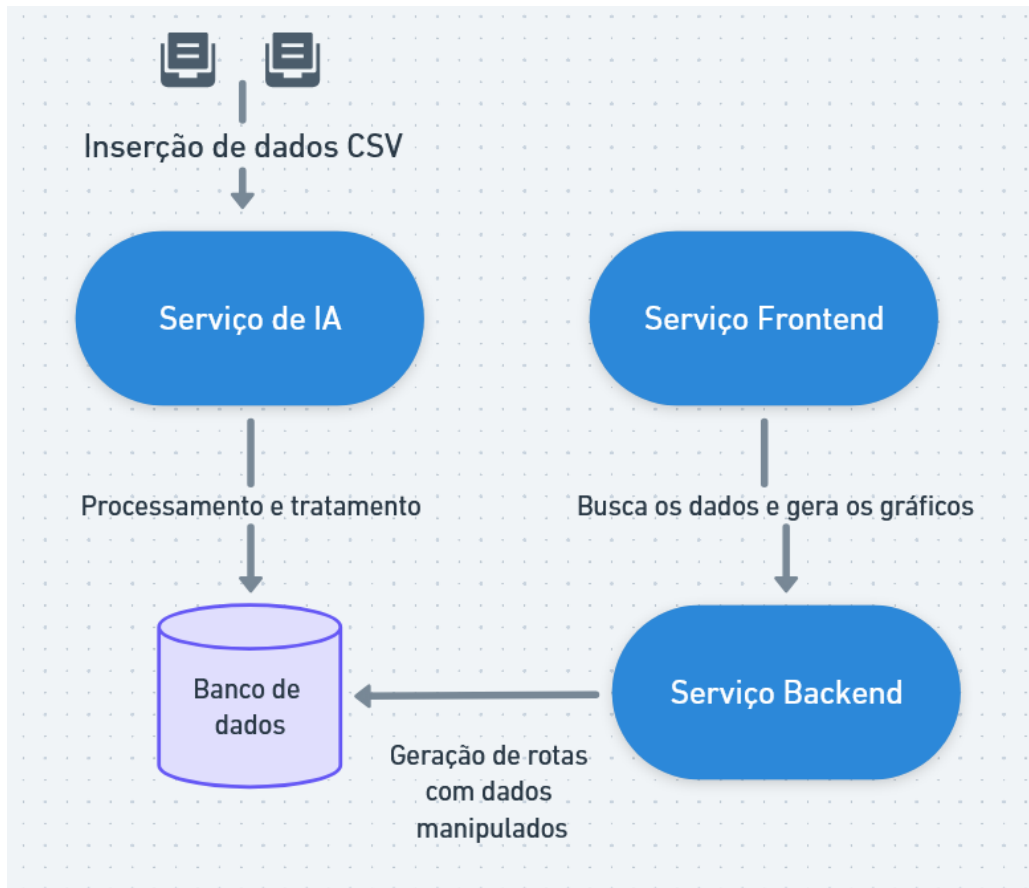
ARQUITETURA E FUNCIONAMENTO DOS SISTEMAS

I. Sobre o Sistema

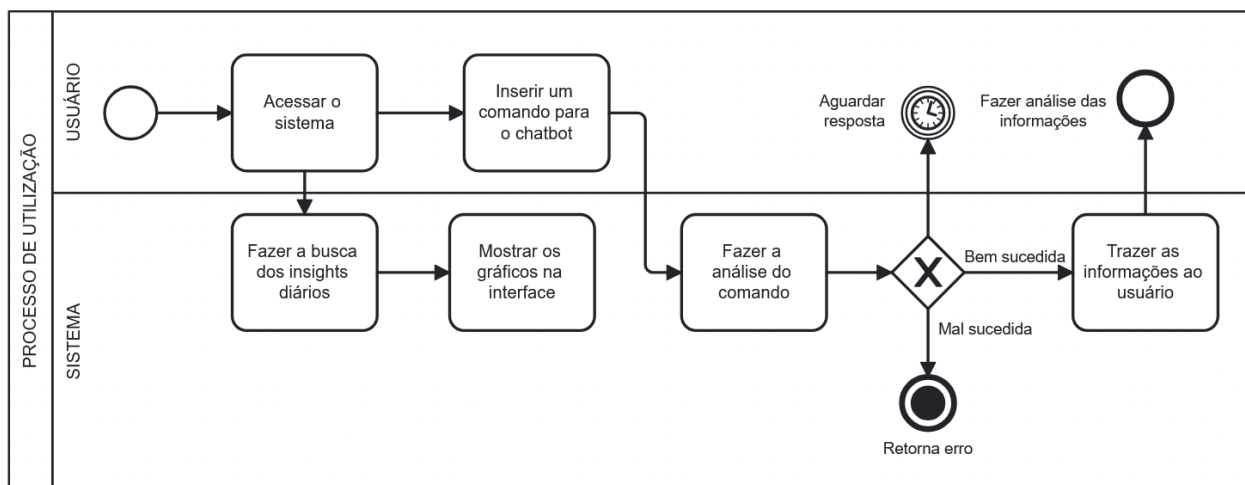
Esse sistema foi elaborado com base na problemática de, na presença de grandes volumes de dados gerado por interações de usuários tendo como fonte os sistemas de atendimento, se faz necessária uma análise de dados eficiente e automatizada para que se possa extrair informações relevantes destes dados e formar insights de interesse ao negócio. Tendo isso em vista, o sistema foi construído para receber um conjunto de dados no forma de CSV, fazer seu processamento, tratamento e manipulação para que possam ser então mostrados graficamente e intuitivamente.

II. Visão Geral

A seguir, uma imagem contendo a arquitetura e fluxo do sistema:



A seguir, uma imagem contendo o BPMN do processo de utilização do sistema:



III. Funcionamento do Serviço de IA

O serviço de IA foi desenvolvido em python e a principal biblioteca é o pandas para manipulação dos dados.

A primeira etapa ocorre com a inserção dos arquivos CSV dentro da pasta destinada. Atualmente o sistema trabalha com arquivos de chamados e do software jira.

O intuito desse serviço é processar os dados dos arquivos e logo em seguida fazer o tratamento.

Durante o tratamento os dados passam por uma pipeline de adaptações, fazendo normalização, limpeza de conjuntos irrelevantes, padronização e etc.

Finalizada a etapa de tratamento o serviço então armazena os dados obtidos no banco de dados.

IV. Funcionamento do Serviço Banco de Dados

O banco de dados escolhido foi o postgresql no formato de container do docker.

Ele possui duas tabelas, uma para administrar os dados de chamados, outra para os dados do jira.

A seguir, uma imagem contendo as tabelas e seus atributos.

Tabela Chamados	Tabela Jira
id	id_da_item
titulo	resumo
entidade	tipo_do_ticket
categoria	status
localizacao	nome_do_projeto
elementos_associados	responsavel
data_abertura	criado
data_fechamento	atualizado
tempo_interno_excedido	resolvido
tempo_resposta	descricao
tempo_interno_resposta	anexos
descricao	
plugins	
solucao	
status	
tipo	
tecnico_atribuido	
fornecedor_atribuido	
ultima_atualizacao	

v. Funcionamento do Serviço Backend

O serviço backend foi construído em typescript.

Assim que os dados são guardados no banco de dados, o backend estará habilitado a coletá-los a partir de suas consultas.

Com os dados, o back utiliza suas várias funções para enquadrá-los nas funcionalidades do sistema, fazendo filtragens, categorizações e aplicando-os com base nas regras de negócio.

Feita essa afunilação de informações, o back então libera esses dados manipulados em rotas http.

VI. Funcionamento do Serviço Frontend

O frontend foi elaborado em typescript com react e vite.

Essa é a última etapa do sistema. O front disponibiliza uma interface simples, clara e intuitiva, para que o usuário tenha facilidade ao procurar as informações geradas.

Após a manipulação do back, o front carrega os dados, fazendo buscas nas rotas disponíveis. Esses dados são então aplicados e mostrados em gráficos para formalizar os insights de interesse.

DESENVOLVIMENTO DO PROJETO

VII. Metodologias e técnicas

O projeto engloba os princípios e práticas da metodologia ágil scrum.

O grupo se organizou utilizando o discord, como principal meio de interação, o figma para prototipação, o whinsical para documentação, o jira para planejamento de escopo e tarefas, o slack para comunicação com o cliente e principalmente o git/github para documentação e repositório do projeto.

Para desenvolvimento prático do projeto, foi utilizado o docker para containerização, aws para infraestrutura em rede, visual studio code para criação

de código e postgresql como banco de dados. As principais linguagens e tecnologias relacionadas utilizadas em serviços foram: python, pandas, typescript, express, prisma, react, e vite.

VIII. Padronização de branches e commits

O planejamento do projeto foi elaborado na ferramenta jira, desta forma, optamos por seguir como o padrão de branches, cada funcionalidade adicionada ou retrabalhada, de acordo com as tarefas descritas no jira, ou seja, o nome das branches será dado pelo nome da tarefa, pois cada tarefa representa uma funcionalidade.

Para o padrão de commits, estabelecemos como regra uma mensagem curta e direta, começando pela palavra identificadora de trabalho, em seguida os “:”, e então a mensagem que descreve resumidamente a atividade realizada.

As palavras identificadoras de trabalho são: “feat” - para novas features que serão implementadas, “fix” - para correção crítica em features que já foram implementadas, “refactor” - para alteração ou melhorias em componentes que não necessariamente representam um problema, e “docs” - para descrever alguma documentação de caráter específico.

IX. Andamento do projeto

Esse projeto teve um começo (primeira sprint) bastante instável no que se refere a planejamento de escopo. O planejamento inicial passou por diversas alterações até alcançar sua versão sólida, isso significou, inclusive, certa quantidade de esforço e tempo desperdiçados. Felizmente porém, esses obstáculos já estão previstos na metodologia ágil e é com ela que é feito o contorno dessas situações.

Inicialmente, foi proposto um escopo muito complexo e longo, mas que rapidamente recebeu alterações, algumas delas foram: o abandono de um serviço em plataforma mobile que já havia sido iniciado, funcionalidades relacionadas à login (não deve ser feito por enquanto), dentre outras.

Devido à constante comunicação com o cliente, os professores e com o próprio grupo, foi possível contornar esses obstáculos e realizar a entrega.

F

Funcionamento do Serviço Backend

Funcionamento do Serviço frontend

Estilo commits e branches

Primeiro ocorre processamento dos dois csv

Depois ocorre o tratamento limpeza dos dados

Campo descrição é o mais importante

Retorna uma lista de chamados

Aplica essa lista ao banco

Backend pega os dados e faz o processamento

Aplica regras de negócio para os insights

Aponta os dados para uma rota

Frontend busca a rota e mostra o gráfico