

UnBuscas

Documento de Arquitetura

Versão [1.0](#)

Histórico de Revisão

Data	Versão	Descrição	Autor(es)
03/11/2023	1.0	Finalização da Versão do Documento de Arquitetura	Grupo Centauri

Autores:

Matrícula	Nome	Descrição do papel assumido na equipe	% de contribuição ao trabalho (*) ¹
200059980	Pedro Miguel Martins de Oliveira dos Santos	Acompanhamento e desenvolvimento Frontend e Backend com foco maior nesse último.	20%
211062230	Mariana Letícia Santos da Cruz	Scrum Master e acompanhamento do desenvolvimento do Back-end	15%
211061402	André João Cordeiro Gomes	Acompanhamento no back e no front, com foco no que for necessário durante a sprint.	10%
211061574	Danielle Rodrigues Silva	Desenvolvimento Frontend e Prototipação	10%
211062446	Renan Araújo de Souza	Desenvolvimento Frontend e Prototipação	20%
211062526	Thomas Queiroz Souza Alves	Planejamento e codificação do backend de Busca de Documentos; Testes Unitários e de Integração; Otimização do backend de Busca para atender aos requisitos não funcionais.	3%
211041132	Felipe Guimarães Fernandes	Desenvolvimento Frontend	12%
	João		12%

¹ (*) — Para cada integrante da equipe, considere sua participação tanto no Documento de Arquitetura, quando nos demais documentos já entregues pela equipe (Visão do produto e do projeto; declaração de escopo) e atribua um, percentual. A soma dos percentuais de todos os integrantes devem fechar em 100%)

Sumário

1	Introdução.....	4
1.1	Propósito.....	4
1.2	Escopo.....	4
2	Representação Arquitetural.....	4
2.1	Definições.....	4
2.2	Justifique sua escolha.....	4
2.3	Detalhamento.....	4
2.4	Metas e restrições arquiteturais.....	4
2.5	Visão de Casos de uso (escopo do produto).....	4
2.6	Visão lógica.....	4
2.7	Visão de Implementação <relembra aula da Prof. ^a Milene Serrano>.....	5
2.8	Visão de Implantação <relembra aula da Prof. ^a Milene Serrano>.....	5
2.9	Restrições adicionais.....	5
3	Bibliografia.....	5

1 INTRODUÇÃO

1.1 PROPÓSITO

Este documento descreve a arquitetura do sistema sendo desenvolvido pelo grupo Centauri, na disciplina de MDS — Métodos de Desenvolvimento de Software — edição do segundo semestre de 2023 UnBuscas, a fim de fornecer uma visão abrangente do sistema para desenvolvedores, testadores e demais interessados.

1.2 ESCOPO

O detalhamento do escopo se encontra no documento ESCOPO — Centauri. Porém, em linhas gerais, o escopo do produto compreende o desenvolvimento de um software chamado UnBuscas, cujo objetivo é padronizar e simplificar a busca por documentos na Universidade de Brasília(UNB). O projeto visa melhorar a experiência de busca, tornando ela mais rápida e eficaz comparada a outros motores de busca, e reduzir obstáculos pelos alunos devido à diversidade de formatos e estruturas do documento da UnB.

2 REPRESENTAÇÃO ARQUITETURAL

2.1 DEFINIÇÕES

O sistema seguirá uma arquitetura *de Microserviços*.

2.2 JUSTIFIQUE SUA ESCOLHA.

Escolhemos a arquitetura de microserviços com base nos objetivos que desejamos alcançar durante o projeto, isto é: um back-end de motor de busca, um back-end de login de usuários e um front-end. Isso quer dizer que, ao modularizar essas seções e deixando-as independentes durante o processo, o desenvolvimento para essa proposta foi facilitada. Além disso, foi possível fazer o uso de diferentes linguagens em cada seção, o que facilitou também no desenvolvimento do projeto principalmente pela utilização de frameworks que facilitaram boa parte do projeto.

2.3 DETALHAMENTO

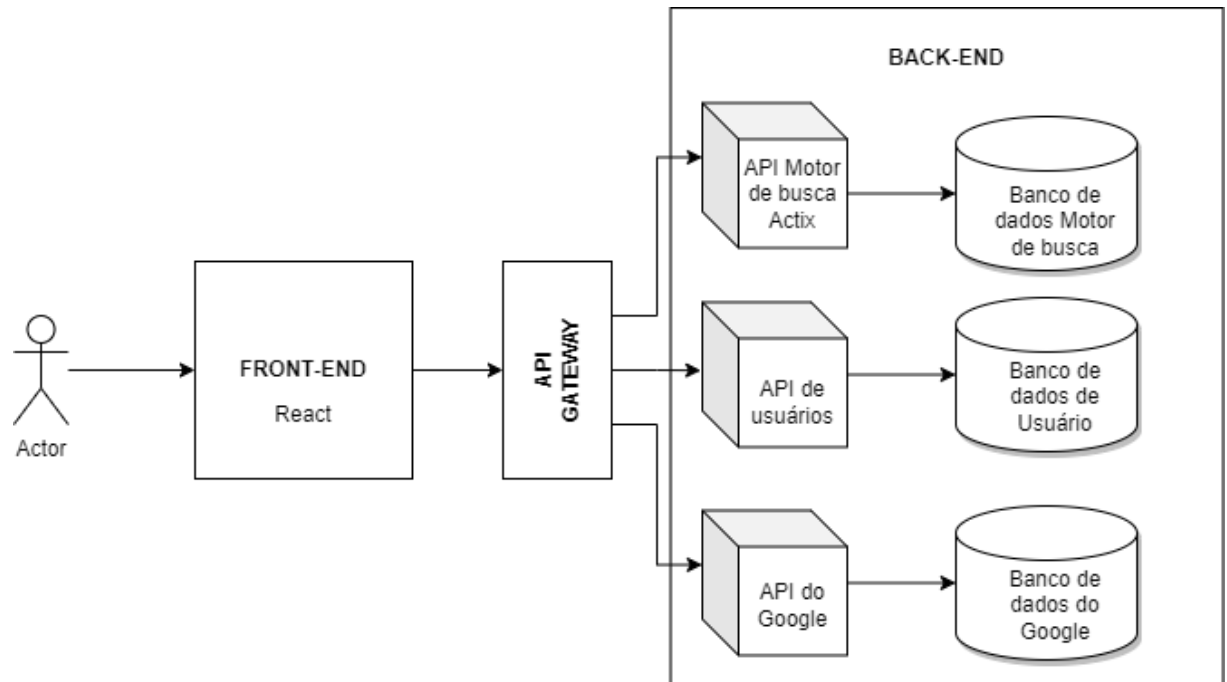


Figura 1 - Visão esquemática do estilo de arquitetura de micro serviços

2.4 METAS E RESTRIÇÕES ARQUITETURAIS

- O sistema deve atender às consultas em até 50 milissegundos, pelo tempo do servidor ser pequeno e já mostrar resultados enquanto o usuário estiver digitando.
- O sistema deve usar a linguagem RUST para o backend, MySQL para o banco de dados do motor de busca e do usuário, o React e Node.js para o front-end.
- Usar apenas tecnologias open-source e gratuitas, por ser um projeto sem objetivo de gerar lucro e ter o intuito de aprendizado.
- As requisições entre o frontend e o backend devem ser em HTTP com JSON em URI encoding
- O software deve ordenar os resultados e filtrar por categoria, para tornar o software mais fácil de entender e como usar (usabilidade)

2.5 VISÃO DE CASOS DE USO (ESCOPO DO PRODUTO)

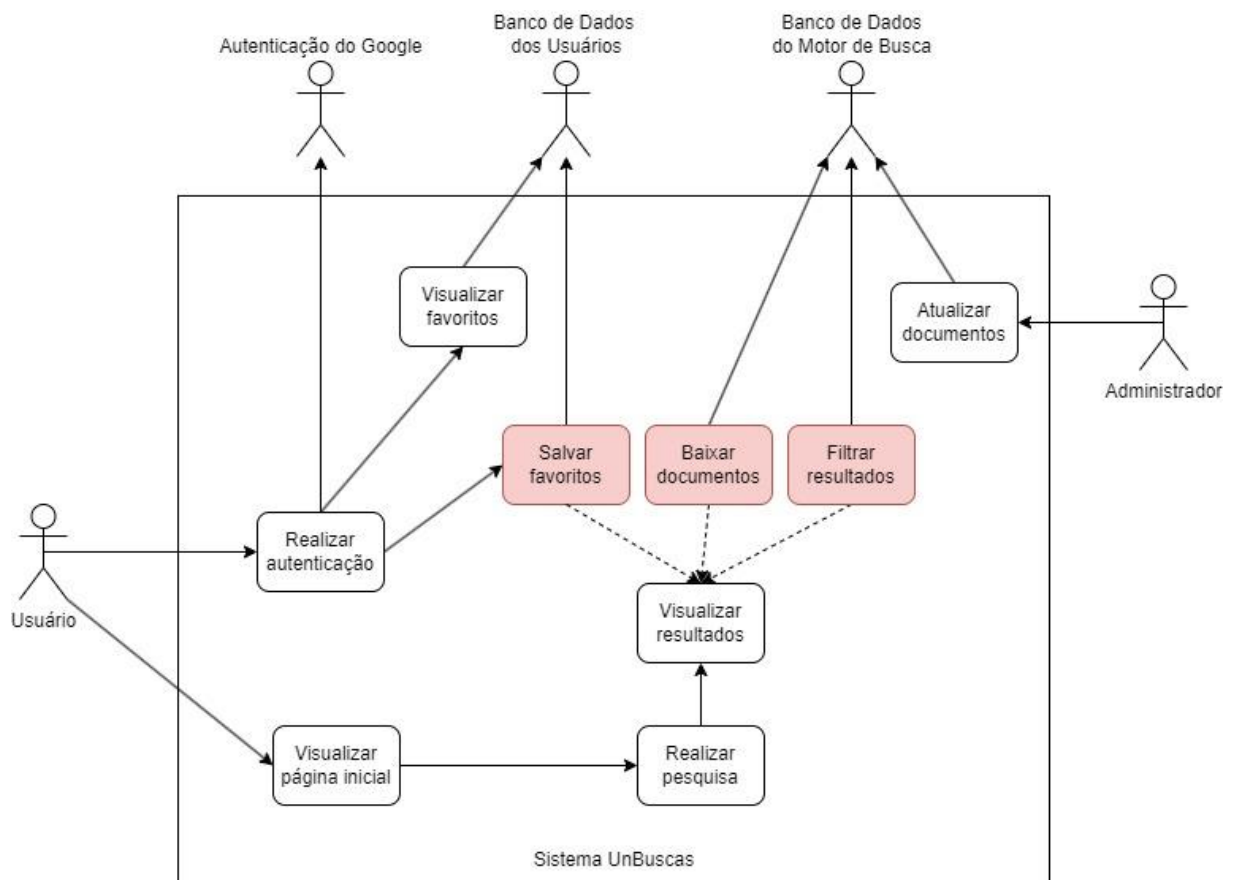


Figura 2 - Diagrama de caso de uso

O escopo dos casos de uso é bem tranquilo por se tratar de um fluxo simples para realizar pesquisas de documentos da UnB e possivelmente baixar, filtrar ou salvar em seus favoritos. Para este último é necessário realizar a autenticação pelo sistema externo integrado do Google que terá no UnBuscas.

2.6 VISÃO LÓGICA

2.6.1 Módulos

- back-end Motor de busca
- back-end Usuários
- front-end

2.6.2 Como as interfaces se comunicam

Da interface da página inicial, se pode ir para duas outras interfaces: a página de login, e a página de pesquisa.

Da página de login, se pode voltar para a página home e ir para a página de usuários(onde os documentos favoritos são armazenados).

Da página de pesquisa, se pode ir para a para a página de erro(quando um documento não foi encontrado).

2.6.3 Diagrama de estados

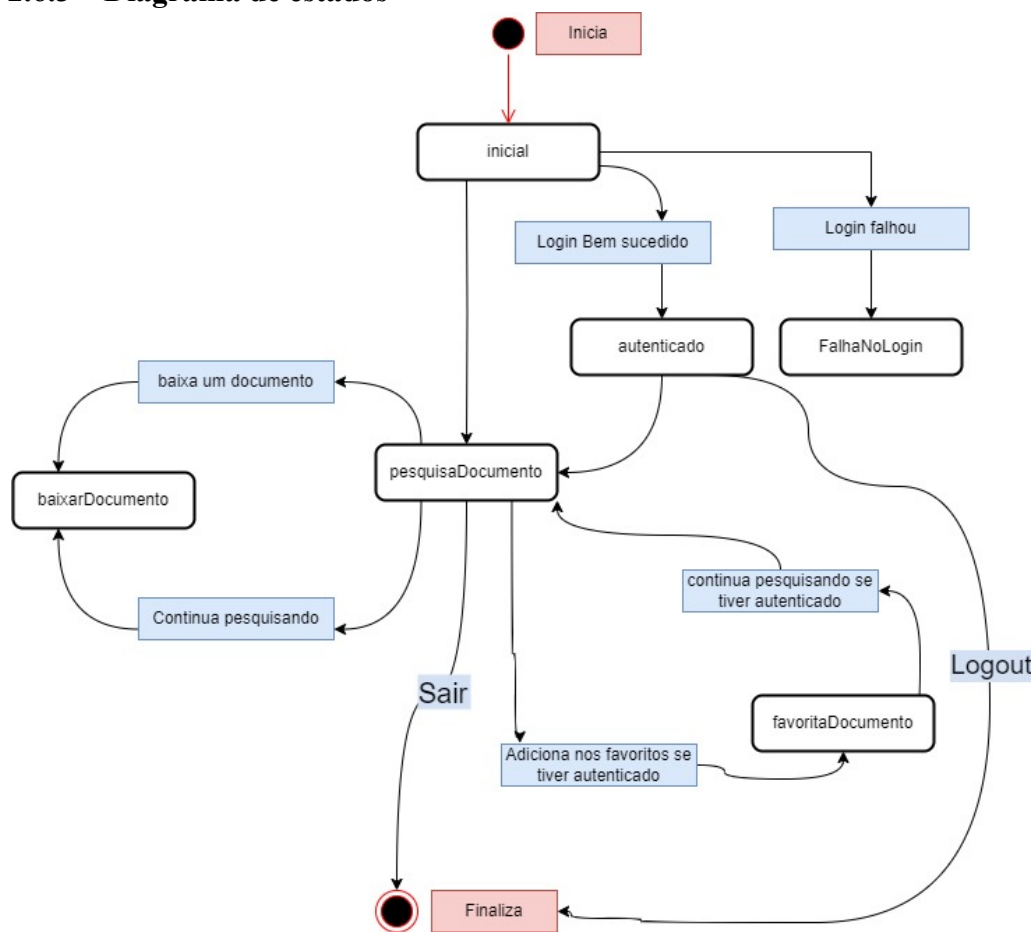


Figura 3 - Diagrama de estados

2.6.4 Diagrama de Atividades

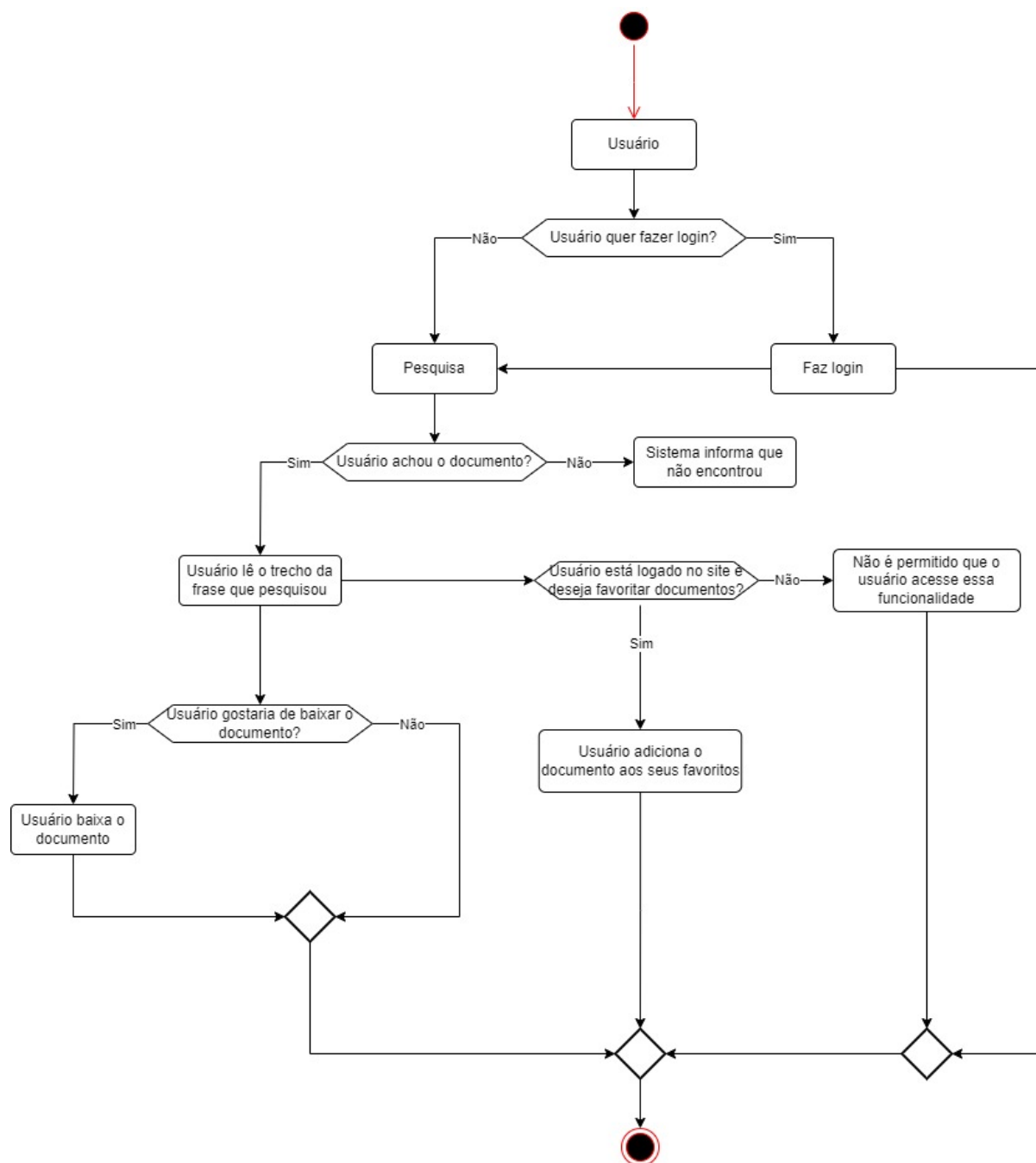


Figura 4 - Diagrama de atividades

2.6.5 Diagrama de Classes

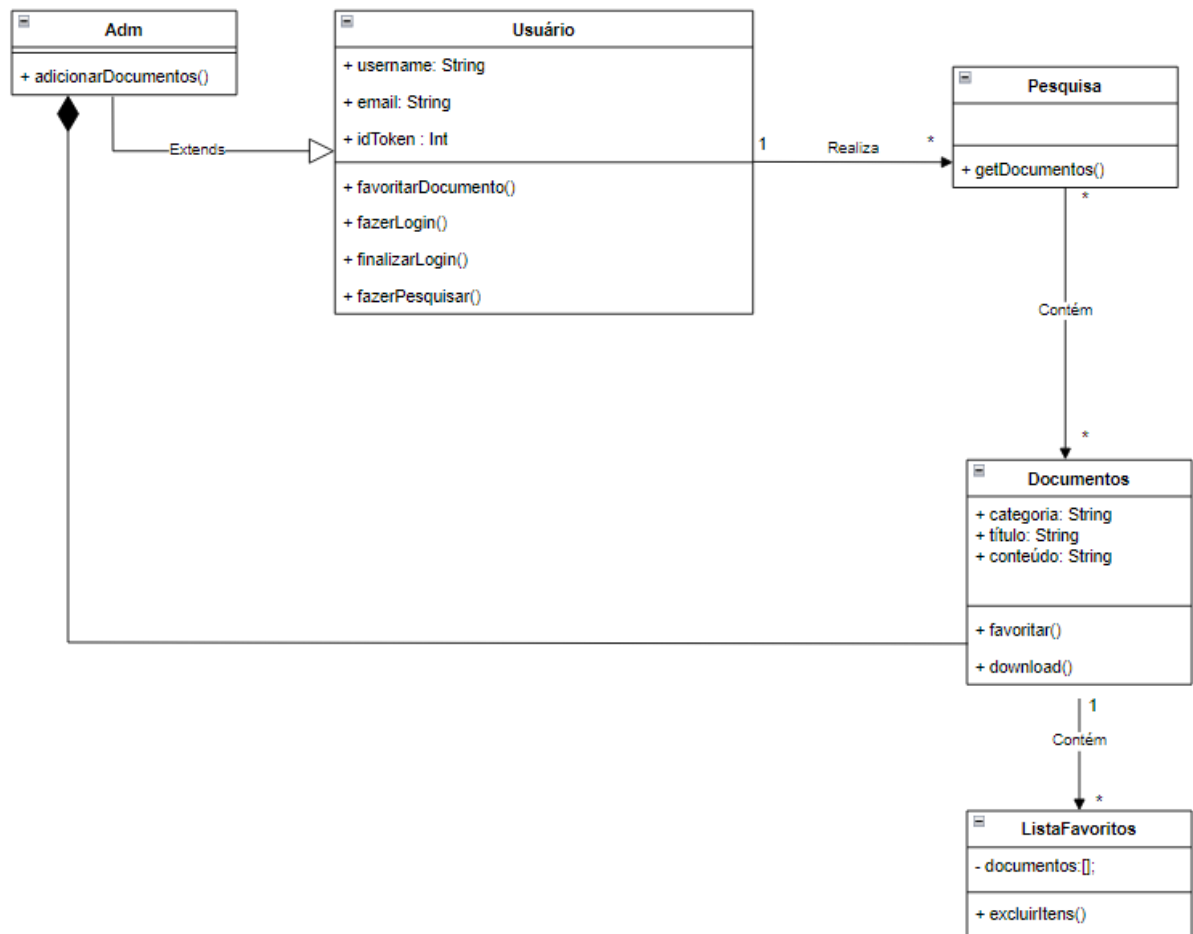


Figura 5 - Diagrama de classes

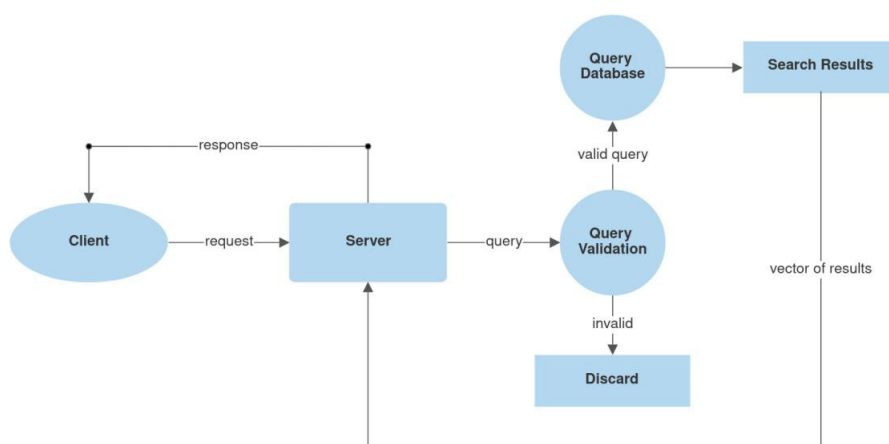


Figura 6 - Diagrama de Data-flow do back-end do motor de busca

2.7 VISÃO DE IMPLEMENTAÇÃO

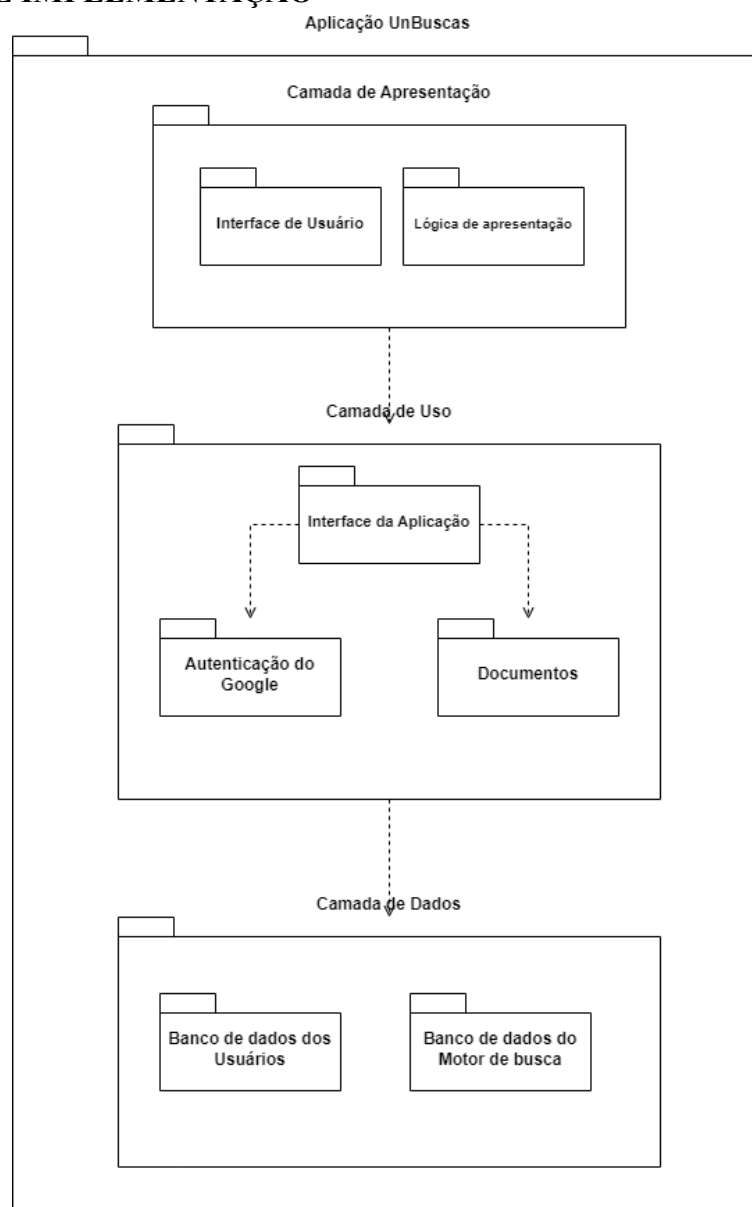


Figura 7 - Diagrama de Pacotes

Quando o usuário visualiza a interface de aplicação ele pode realizar a autenticação do Google ou seguir para a pesquisa, filtro ou download de documentos que é auxiliada pelo banco de dados.



Figura 8 - Página Inicial



Figura 9 - Página de Resultado de Pesquisa

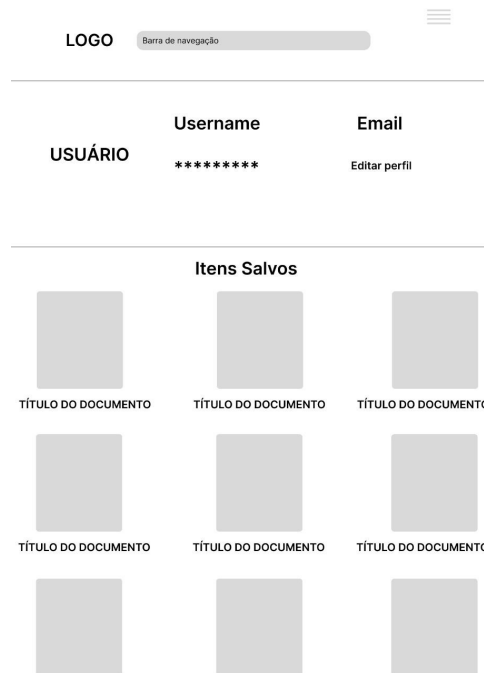


Figura 10 - Página de Favoritos

2.8 VISÃO DE IMPLANTAÇÃO

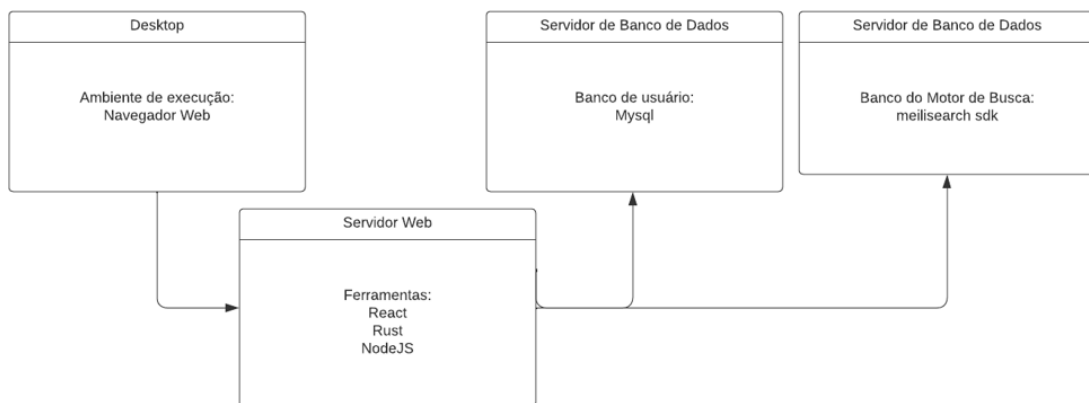


Figura 11 - Diagrama da visão de implantação

O software será implantado em um Desktop para ser executado em um servidor web como a maioria dos motores de busca da atualidade, segundo as tecnologias Rust, React e NodeJs. O Rust foi escolhido devido à sua capacidade de oferecer abstrações de baixo custo e segurança de memória sem a necessidade de um coletor de lixo. Seu compilador detecta erros durante a compilação e evita a necessidade de alguns tipos de testes unitários. O Node JS foi

selecionado para o BackEnd do fluxo de usuários, enquanto o React, escolhido para o front-end, permite a criação de interfaces de usuário divididas em componentes reutilizáveis, tornando o trabalho eficiente e ágil para a equipe do UnBuscas. Assim como o banco de dados MySQL para o banco de usuário e Meiliseach para o banco do motor de busca, respectivamente. O MySQL como o sistema de gerenciamento de banco de dados para o projeto devido ao conhecimento prévio da maioria dos membros do grupo sobre o sistema, e também devido à capacidade de implementar um banco de dados relacional. Já a seleção do MeiliSearch como base de dados para documentos, é fundamental considerar as alternativas disponíveis. No cenário atual, para a realização de pesquisas textuais, há duas opções viáveis: Elasticsearch e MeiliSearch. O Elasticsearch é reconhecido por sua robustez, porém é direcionado principalmente para Analytics e Data Science, o que o torna complexo para ser configurado de acordo com nossos propósitos, além de não oferecer um SDK satisfatório para facilitar o desenvolvimento. Por outro lado, o MeiliSearch dispõe de um SDK versátil, que pode ser facilmente adaptado para as exigências de diversos projetos.

2.9 RESTRIÇÕES ADICIONAIS

- Exigir que os usuários façam login para acessar os serviços do software, o motivo para isso é a confiabilidade e portabilidade do software, criando um ambiente seguro e manter o histórico do usuário caso utilize em diferentes dispositivos.

3 BIBLIOGRAFIA

Grupo Alpha Centauri. Miro Mind Maps MDS. Disponível em: https://miro.com/app/board/uXjVMILh5MM=/?share_link_id=69700570299. Acesso em: 05 de outubro de 2023

LETÍCIA, M.; PEDRO MIGUEL. MDS Centauri - Backlog e Critérios de Aceitação. 2023. Disponível em: <https://docs.google.com/spreadsheets/d/18qN3aizmMQCIOUuCEOiuYvgYZNmI8NbCw8UrxjkhCk/edit#gid=600013620>.

Documentação Backend MDS Centauri, disponível em:

<https://proptest-rs.github.io/proptest/intro.html>,
https://docs.rs/meilisearch-sdk/latest/meilisearch_sdk,
<https://actix.rs/docs/>.