

UnBuscas

Declaração de Escopo do Produto

Versão [\[1.0\]](#)

Histórico de Revisão

Data	Versão	Descrição	Autor(es)
23/10/2023	1		

Sumário

1	Problema / Sistema de software.	4
2	Backlog do produto.	5
2.1	Perfis.	5
2.2	Cenários.	5
2.3	Backlog do produto.	6
2.4	Sprints previstas.	7
3	Definição de Ready /Done.	7
4	US – User Histories.	7
5	Diagrama de casos de uso.	7
6	MVP.	7
7	Referências.	7

1 Problema/Sistema de Software

Definição da equipe:

Renan: Desenvolvimento Front-end.

Danielle: Desenvolvimento Front-end..

Felipe: Desenvolvimento Front-end.

Thomas: Desenvolvimento do motor de busca

Mariana: Desenvolvimento Fullstack

Pedro: Desenvolvimento Fullstack

André: Desenvolvimento Fullstack

João: Desenvolvimento do motor de busca

Resumo do problema:

Uma das dificuldades que os alunos da Universidade de Brasília enfrentam é a dificuldade de encontrar documentos importantes e de interesse (demora pela procura, documentos espalhados ou desatualizados), onde isso pode afetar nas oportunidades dos alunos podem ter, encontrando esses documentos, impactando negativamente os estudantes pela falta de informação.

Sistema de Software:

UnBuscas

Objetivos:

O objetivo do projeto é desenvolver um software que simplifique e padronize a busca por documentos da Universidade de Brasília (UnB). A intenção é criar uma experiência de busca mais intuitiva e amigável, diminuindo a quantidade de obstáculos que o usuário possa enfrentar devido à diversidade de formatos e estruturas de arquivos que a UnB apresenta. Ao proporcionar uma ferramenta que torna isso mais acessível para o público.

Resumo de tecnologias usadas:

Linguagens de Programação:

React: Utilizado para o Front-end da aplicação.

Rust: Utilizado para o Back-end do motor de busca da aplicação.

Node.js: Utilizado para o Back-end do Fluxo de usuários da aplicação.

API/Bancos de Dados:

MeiliSearch-SDK: Usado para o acesso ao banco de dados.

MySQL: Criar e gerenciar o banco de dados.

Ambiente:

Figma: Criação de designs da interface para o usuário(prototipação).

Github: Hospedar o projeto e salvar todo tipo de mudança do código-fonte.

Trello: Gerenciamento e acompanhamento do projeto.

Programas:

VsCode - utilizada para programar a aplicação.

Resumo da metodologia de desenvolvimento usada:

ScrumXP: O ScrumXP combina as metodologias Scrum e Extreme Programming(XP). Scrum é uma abordagem ágil de desenvolvimento de software que enfatiza a entrega em ciclos curtos de tempo, chamados de sprints. Valoriza também a colaboração e trabalho em equipe a partir de times que conseguem ser auto-organizáveis e adaptáveis. O XP (extreme programming) é uma metodologia com foco em agilidade de equipes e qualidade de projetos, apoiada em valores como simplicidade, comunicação, feedback e coragem. O XP é uma metodologia baseada em comportamentos e atitudes

Fases do ScrumXP:

Scrum:

Sprint

Sprint Planning

Daily

Retrospectiva da sprint

XP:

Projeto Simples

Integração contínua

Posse coletiva

Padronização de código

Small Releases (Pequenas versões)

Atividades:

Repetir até ficar correto

Métricas usadas no desenvolvimento:

Nenhuma métrica foi desenvolvida.

Testes de software:

Teste de Integração:

Teste de integração é quando os módulos (unidades de código) são testados em grupo para garantir que não haja nenhuma quebra naquilo que foi feito unitariamente e naquilo que está sendo integrado junto.

Teste Unitário:

Testes unitários, assim como qualquer teste automatizados não servem principalmente para verificar se uma função específica está funcionando, mas sim para garantir que sua aplicação continue funcionando após alguma alteração em sua base de código.

2 Backlog do produto

Forneça o backlog do produto, constituído pelos requisitos a serem contemplados no desenvolvimento:

- Forma de obtenção do requisito (elicitação): brainstorm, reuniões.

2.1 Perfis

Tabela 1: Perfis de acesso

#	Nome do perfil	Características do perfil	Permissões de acesso
1	Administrador	Responsável por manter os perfis de acesso da aplicação, criar novos usuários, alterar usuários já existentes, ou excluir usuários (Manter usuários)	Descrever sucintamente as permissões de acesso
2	Usuário	Responsável por consultar a plataforma, organizar seus documentos em uma sessão de favoritos	Pesquisar por documentos e criar conta no site para adicionar documentos aos favoritos

2.2 Cenários

Tabela 2: Cenários funcionais

Sistema: 1 – Cenários funcionais		
Numeração do cenário	Nome do cenário	Sprints
1	Pesquisar	1,2,3,4,5
2	Logar	4,5
3	Filtrar pesquisa	5

2.3 Backlog do produto

Tabela 3: Backlog do produto

Sistema: 1 – Backlog do produto						
Numeração (Cenário / requisito)	Sprint	Nome do requisito	Tipo de requisito (Funcional / não funcional)	Priorização do requisito Must, Should, Could	Descrição sucinta do requisito	User stories (U.S.) associadas Identifique as U.S. associadas ao requisito
1	1	Popular banco de dados	Funcional	Must	Eu como desenvolvedor desejo popular meu banco de dados com dados fictícios para poder implementar a funcionalidade de motor de busca	1
1	2	Criar parser de textos	Funcional	Must	Eu como desenvolvedor desejo criar um parser de textos para editar documentos	2
1	2	Carregar dados do parser	Funcional	Must	Eu como desenvolvedor populou o banco de dados com dados extraídos do parser	3
1	3	Visualizar documentos na interface	Funcional	Should	Eu como desenvolvedor devo visualizar os dados carregados	4

					numa página para leitura	
1	3	Mostrar os dados de pesquisa	Funcional	Must	Eu como usuário, vejo os dados pesquisados para conferir aquilo que estava procurando	5
1	4	Visualizar erro	Funcional	Must	Eu como usuário, desejo visualizar um erro para saber o porquê de eu não conseguir acessar essa funcionalidade	6
1	4	Entrar pela minha conta na página	Funcional	Could	Eu como usuário desejo entrar na minha conta para acessar o site de busca	7
1, 2	5	Armazenar dados de usuário	Funcional	Should	Eu como desenvolvedor quero armazenar os dados de usuário para dedicar funções específicas	8
2	5	Filtrar dados de pesquisa	Funcional	Could	Eu como usuário desejo filtrar documentos para encontrar o que mais se adequa a minha pesquisa	9
1	4	Favoritar documentos	Funcional	Could	Eu, como usuário, quero favoritar os documentos para que eu os veja posteriormente com facilidade	10
3	6	Pré-visualizar pesquisa	Funcional	Could	Eu, como usuário, desejo pré-visualizar minha pesquisa enquanto digito para encontrar um documento de forma rápida	11

3	7	Visualizar miniatura dos documentos favoritos	Funcional	Could	Eu, como usuário, devo ver a miniatura dos documentos favoritos para reconhecê-los	12
3	8	Marcar documentos lidos	Funcional	Could	Eu, como usuário, desejo visualizar uma lista de documentos lidos para filtrar tudo que tenho favoritado	13

2.4 Sprints previstas

Tabela 3: Sprints previstas

Sistema: 1 – Sprints previstas			
# Sprint	Descrição	Objetivos	Composição de itens do backlog (Lista conforme tabela Backlog do produto)
1	Organização do projeto e documentação inicial	definir projeto a ser feito, definir a tecnologia a ser utilizada e definir o escopo do projeto	
2	Popular banco de dados e Carregar dados do parser	colocar documentos no banco de dados do motor de busca e filtrar ele via parser	US1 e US2
3	Carregar dados do parser Visualizar documentos em interface simples	os dados retirados do parser serão utilizados para popular novamente o banco de dados	US3 e US4

4	Conexão do back-end do motor de busca com o front-end	foi estabelecido a conexão entre back-end do motor de busca e do front-end	US 5
5	Visualizar erro, login de usuário e filtragem	Criação da página de erro, da página de usuário, fazer login pelo google e fazer filtragem de dados	US6 e US7
6	associar usuários com os documentos	Armazenar dados de usuário e conseguir favoritar documentos	US8 e US9

3 Definição de Ready /Done

Definition of Ready (DOR)

Definição de Preparado

- É um requisito é uma declaração funcional ou não-funcional?
- Sendo não-funcional o requisito se encaixa na matriz URPS+ ou funcional utiliza cartão, conversação e confirmação como história de usuário?
- O requisito é simples e curto?
- Tem tamanho ideal para uma sprint?
- Caso não, é possível separar em sprints?
- O requisito faz parte da história do usuário?
- O requisito possui critérios de aceite?

Definition of Done (DOD)

Definição de Pronto

- Os critérios de aceite foram cumpridos?

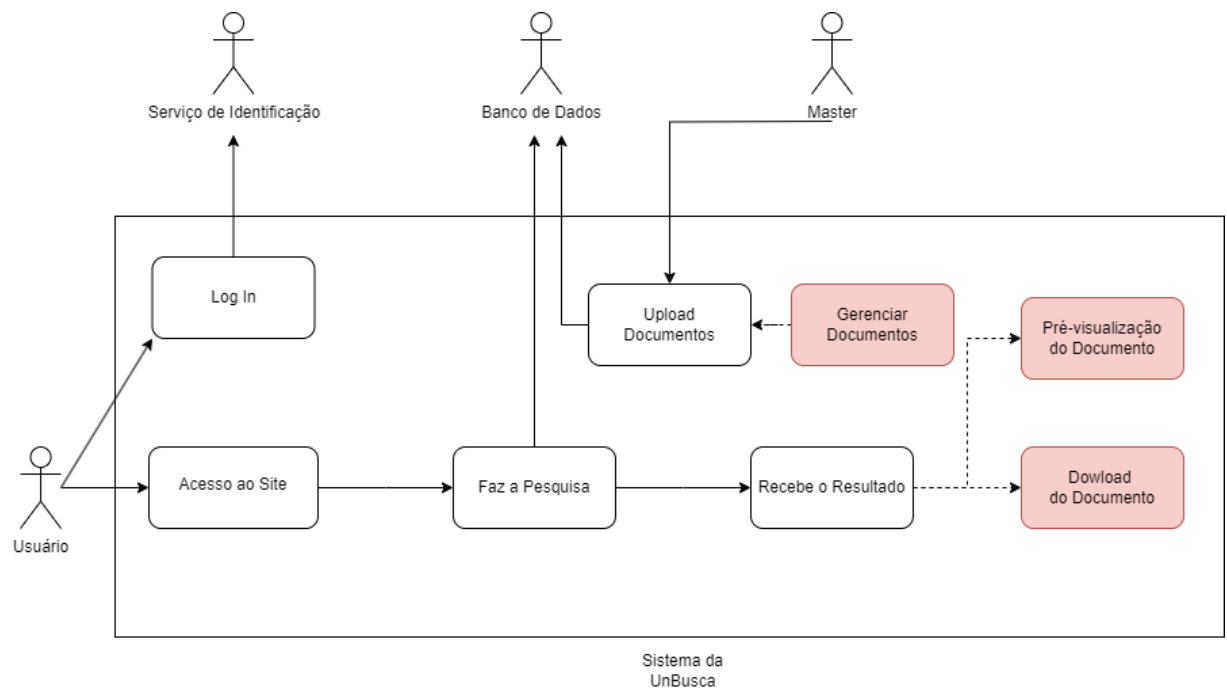
- Obedece às práticas de programação como clean code?
- Os testes foram realizados?
- Passou em todos os testes?
- Foi feita a documentação adequada?
- Houve design simples?
- Contém os padrões do cliente?

4 US – User Stories

Nº SPRINT	Nº US	Nome da US	TEMA	ÉPICO	Releases
1	-	Organização do projeto e documentação inicial			Release 1
2	1	Popular banco de dados	Pesquisa de documentos	Tratamento de documentos	
2	2	Criar parser de textos			
3	3	Carregar dados do parser			
3	4	Visualizar documentos na interface		Visualização dos documentos	
4	5	Mostrar os dados de pesquisa			
5	6	Visualizar erro			

5	7	<i>Entrar pela minha conta na página</i>	<i>Uso da interface</i>	<i>Integração com usuário</i>	<i>Release 2</i>
6	8	<i>Armazenar dados de usuário</i>			
4	9	<i>Filtrar dados de pesquisa</i>		<i>Usabilidade de pesquisa</i>	
6	10	<i>Favoritar documentos</i>			
7	11	<i>Pré-visualizar pesquisa</i>	<i>Funções de ajuda</i>	<i>Funções de ajuda</i>	<i>Release 3</i>
8	12	<i>Visualizar miniatura dos documentos favoritos</i>			
9	13	<i>Marcar documentos lidos</i>			

5 Diagrama de casos de uso



6

MVP

US 01	Popular banco de dados
US 02	Criar parser de dados
US 03	Carregar dados do parser
US 04	Visualizar documentos na interface
US 05	Mostrar os dados de pesquisa
US 06	Visualizar aviso de erro

O nosso primeiro MVP (*Minimum Product Viable* ou *Mínimo produto viável*) seria todo o tema 1 (Pesquisa de documentos) para o tratamento dos documentos (pelo desenvolvedor) e para a pesquisa dos documentos no banco, usando a barra de pesquisa da aplicação (pelo usuário). Dessa forma o produto consegue fazer minimamente sua função esperada, sem isso o produto não teria o funcionamento esperado e não resolveria o problema proposto.

Posteriormente novos requisitos podem ser implementados buscando deixar o produto mais completo e com mais funcionalidades, sendo um **MVP 2**:

US 07	Entrar pela minha conta na página
US 08	Armazenar dados do usuário
US 09	Filtrar dados de pesquisa
US 10	Favoritar documentos
US 11	Pré-visualizar pesquisa
US 12	Visualizar miniatura dos documentos favoritados
US 13	Marcar documentos lidos

Com o produto fazendo suas funcionalidades mínimas posteriormente pode ser implementado novas funcionalidades, como o login do usuário(US 07), permitindo que o usuário possa salvar como

favorito documentos para serem acessados em uma página específica. Outra funcionalidade a ser implementada será a filtragem dos documentos(US 09), como por exemplo a filtragem por datas para que o usuário delimite a época do documento que ele está procurando e possa ver um documento mais recente ou mais antigo para o problema a dúvida a ser resolvida.

7 Referências

LETÍCIA, M.; PEDRO MIGUEL. MDS Centauri - Backlog e Critérios de Aceitação. 2023. Disponível em: <https://docs.google.com/spreadsheets/d/18qN3aizmMQCIOUuCEOiuYvgYZNmI8NbCw8UrxjjkHCK/edit#gid=600013620>.