

Projeto CodeHub

Atualmente existe uma constante preocupação em relação a otimização de tempo e recurso na execução de projetos, necessitando de um gerenciamento estratégico e utilização de metodologias que auxiliem na dinâmica da equipe e relacionamento com o cliente.

Diante desse cenário, a documentação retrata uma descrição sucinta do projeto GitHub, ao qual, estará fornecendo uma oportunidade de interação e desenvolvimento de habilidades de alunos e/ou entusiastas na área de programação. Por fim, o presente projeto se desenvolve com a metodologia Scrum, utilizando de metodologias ágeis, que serão de auxílio no gerenciamento dos processos e etapas do projeto.

1. Product Backlog

CODEHUB BACKLOG							
ID	Nível hierárquico	Funcionalidade	Objetivo	Prioridade	Sprint	Status	Legenda
1	Visitante	Criar uma conta	Usufruir do sistema	Alta	1	R	Concluído
2	Usuário	Fazer Login	Autenticação	Alta	1	J	Aguardando aprovação
3	Usuário	Fazer Logout	Sair do sistema	Alta	1	M	Bloqueada
4	Usuário	Iniciar novo repositório	Iniciar área de trabalho	Alta	1	I	Pendente
5	Visitante	Exibir comando de ajuda	Orientar usuários quanto as funcionalidades	Alta	1	P	Em Desenvolvimento
6	Usuário	Adicionar ao Container	Permitir confirmações e versionamento	Alta	2		Em teste
7	Usuário	Remover do Container	Não efetuar versionamento	Alta	2		
8	Usuário	Exibir histórico de versões	Lista histórico de comentários e versões	Alta	2		
9	Usuário	Criar nova versão	Versionar o código, evidenciar mudanças	Alta	2		
10	Usuário	Voltar para versão	Retornar a versão anterior, recuperar	Alta	2		
11	Usuário	Remover versão	Cancelar versionamento realizado	Alta			
12	Usuário	Exibir Container	Listar itens contidos no container	Média			
13	Usuário	Backup versão removida	Permitir acesso a contextos apagados	Média			
14	Usuário	Restaurar versão removida	Recupera a última versão disponível	Média			
15	Criador	Adicionar outros usuários	Permitir trabalho colaborativo	Baixa			
16	Administrador	Enviar para nuvem	Salvar conteúdo em um Repósitorio Remoto	Baixa			
17	Usuário	Baixar da nuvem	Get dos dados para o repositório local	Baixa			



2. Sprint Backlog

Durante o planejamento, definiu-se uma sequência de ações a serem consideradas, já cientes da existência de possibilidade de serem alteradas durante o desenvolvimento do projeto.

ANÁLISE DE REQUISITOS SPRINT 1	
Funcionais	Não funcionais
Criar uma conta	A aplicação deve funcionar em sistemas operacionais Linux e Windows
Fazer Login	O sistema deve ser implementado na linguagem de programação java
Fazer Logout	É recomendado 2GB de RAM, 10GB de HD e processador de 1GHz
Iniciar novo repositório	Os dados de login devem ser autenticados
Exibir comando de ajuda	O sistema deve responder imediatamente aos comandos do usuário
	O sistema deve ter um banco de dados local

Sprint 1:

Tempo estimado: 15 dias

Tempo decorrido: 10 dias

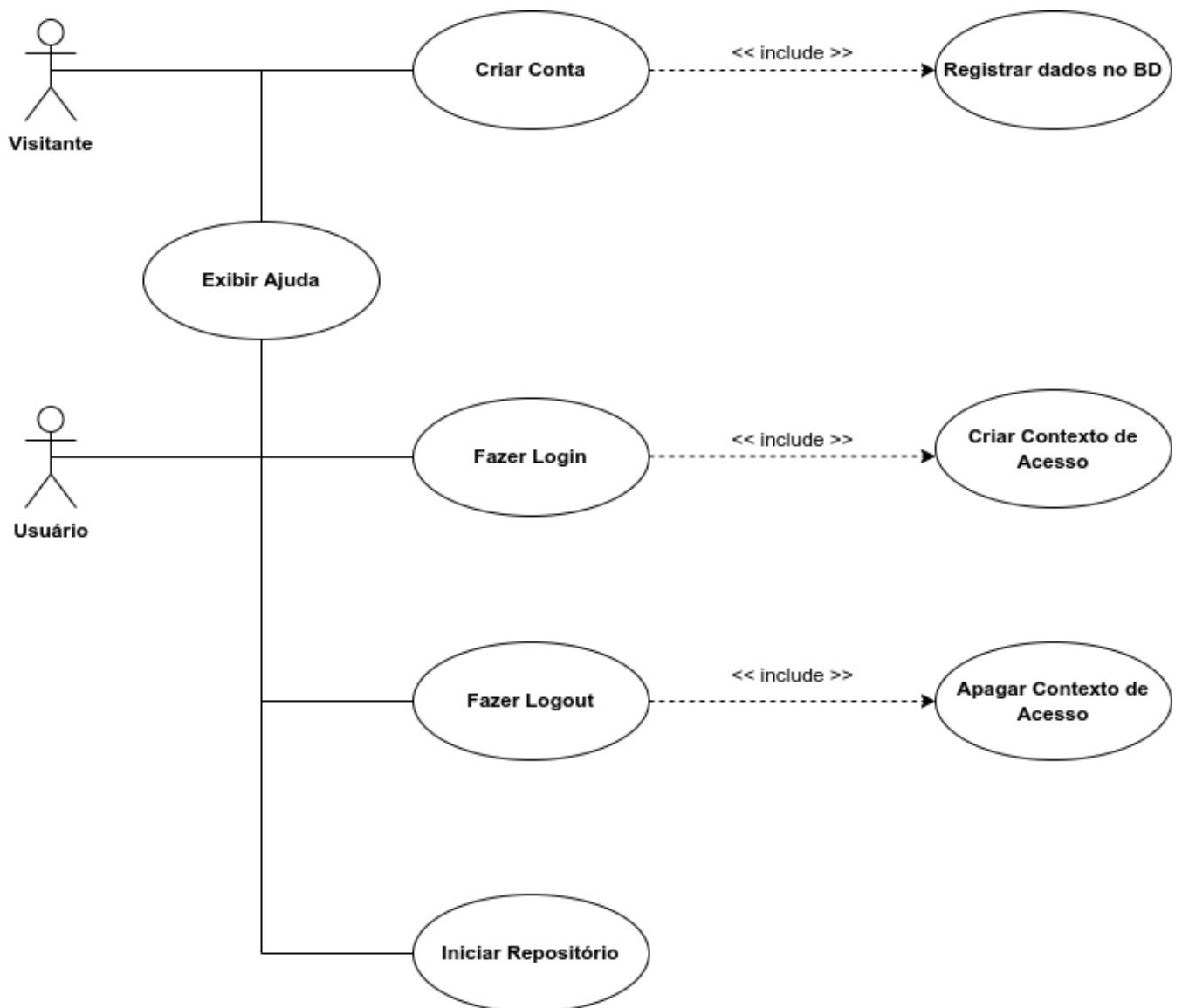
Entrega: 16/05/2022

Relator: Professora Luciana

Funcionalidade	Prioridade	Responsável
Criar conta	Alta	Riquelme
Fazer login	Alta	João Gabriel
Fazer logout	Alta	Marcus Vinícius
Exibir ajuda	Alta	Paloma
Iniciar repositório	Alta	Isaac Santiago

3. Caso de uso e cenário de caso de uso

3.1. Diagrama de casos de uso:



3.2. Cenários de caso de uso:

3.2.1. Sprint 1

Cenário 01:

Nome do Cenário: Criar conta

Ator: Visitante

Pré-condição: nome, email e senha válidas

Fluxo normal:

1. Entrar com dados de registro;
2. Confirmar dados de registro;
3. Caso de uso “Registrar dados no BD”.

Fluxos alternativos: Dados de registro fora do padrão pedido

1. Solicitar repetição de campo

Pós-condição: Dados de login são registrados no Banco de Dados

Cenário 02:

Nome do Cenário: Fazer login

Ator: Usuário

Pré-condição: Já existir contas no sistema

Fluxo normal:

1. Inserir o email
2. Inserir a senha
3. Entrar no sistema

Fluxos alternativos: Email ou senha errada => Apresentar novamente a opção de entrar no sistema.

Pós-condição: Autenticado no sistema.

Cenário 03:

Nome do Cenário: Fazer Logout

Ator: Usuário

Pré-condição: Ter um registro no sistema, e, conseqüentemente, estar logado

Fluxo normal:

1. Exibir mensagem de sair com sucesso.

Fluxos alternativos:

1. Usuário não logado.
2. Exibir mensagem de aviso da necessidade de logar

Pós-condição: Encerrar o processo.

Cenário 04:

Nome do Cenário: Iniciar repositório

Ator: Usuário

Pré-condição: Usuário cadastrado e repositório não ter sido iniciado anteriormente.

Fluxo normal: 1. Inserir comando de iniciar;
2. Checar se o repositório não foi iniciado anteriormente;
3. Gerar repositório .CodeHub.

Fluxos alternativos: Repositório já iniciado

1. Exibir erro ao usuário

Pós-condição: Repositório .CodeHub é criado dentro da pasta do projeto

Cenário 05:

Nome do Cenário: Exibir ajuda

Ator: Usuário, Visitante

Pré-condição: Não possui.

Fluxo normal:

1. Inserir comando de ajuda;
2. Verificar qual comando é de interesse;

Fluxos alternativos: Não existe necessidade de exibir ajuda.

Pós-condição: Não possui.