

GamaHub

Declaração de Escopo do Produto

Versão 1.0

Histórico de Revisão

Data	Versão	Descrição	Autor(es)
25/10/2023	1.0	Primeiras definições de escopo	Rodrigo, João Pedro, Manoel

Sumário

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

1 Problema / Sistema de software

- **Definição da equipe:** Seus integrantes e papéis exercidos no desenvolvimento (todos da equipe devem estar citados – [lista]. Não citados serão considerados como não participantes do projeto Manoel: desenvolvedor front-end, representante do Product Owner; Lara: desenvolvedora front-end; João Pedro: desenvolvedor back-end; Mateus Santos: desenvolvedor back-end; Rodrigo: analista de bancos de dados, scrum master; Eduardo: testador, prototipador, desenvolvedor.
- **Resumo do problema:** conforme documento *Visão do Produto e do Projeto* Falta de um espaço/meio de comunicação para a promoção de eventos de interação entre discentes no campus UnB Gama.
- **Sistema de Software.** Nome do sistema e objetivos, também conforme documento *visão do Produto e do Projeto*. O nome do projeto é GamaHub e seu objetivo é ser uma plataforma para facilitar o encontro de pessoas com interesses em comum e aproximar os discentes.
- **Resumo de tecnologias usadas:** Resuma as tecnologias usadas no desenvolvimento, justificando seus usos Serão utilizados a linguagem de programação JavaScript, por afinidade dos membros do grupo; o JS runtime Node.js, pois é o runtime de maior afinidade do grupo; o framework React, para desenvolver o front-end com facilidade e com as conveniências do framework; o software de bancos de dados MongoDB, por afinidade da equipe; a IDE VSCode, por afinidade da equipe e o software de organização Notion, para organizar as sprints, documentos e arquivos do projeto.
- **Resumo da metodologia de desenvolvimento usada:** Resuma o a filosofia de desenvolvimento seu ciclo de vida (fases e atividades). Considerando o contexto do projeto que é de uma aplicação web com vários riscos envolvidos e prazo bastante limitado, foi definido um ciclo de vida ágil, assim como os processos do Scrum XP, visando um produto de software que agrada os usuários e aceite mudanças de requisitos.
- **Outras informações sobre o ciclo de vida:**
 - o **Métodos, técnicas:** Pair Programming, Code Review, Refactoring, Padrões de Codificação, Integração Contínua, Propriedade Coletiva, Projeto Simples e Metáforas.
 - o **Métricas usadas no desenvolvimento:** Tempo de capacitação da equipe, tempo em dias e quantidade de desenvolvedores para completar uma sprint, grau de satisfação do PO
 - o **Testes de software:** Os níveis de testes usados serão unitários e integração.

2 Backlog do produto

Obtenção de requisitos é realizada por meio de Brainstorm, e pesquisa com alunos;

2.1 Perfis

Tabela 1: Perfis de acesso

#	Nome do perfil	Características do perfil	Permissões de acesso
1	Administrador	Responsável por realizar a moderação dos usuários e eventos cadastrados no sistema.	Permite ver, e deletar usuários e eventos cadastrados. Não tem acesso direto às senhas.
2	Usuário	Pode cadastrar informações pessoais para contato, registrar eventos e se inscrever em eventos criados por outros usuários.	Permite gerenciar seus próprios eventos e dados cadastrados.
3	Convidado	Pode ver todos os eventos cadastrados	Não pode se inscrever

		no sistema.	ou criar eventos, apenas visualizar.
--	--	-------------	---

2.2 Cenários

Tabela 2: Cenários funcionais

Sistema: GamaHub – Cenários funcionais		
Numeração do cenário	Nome do cenário	Sprints

2.3 Backlog do produto

Tabela 3: Backlog do produto

Sistema: GamaHub – 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	Criar Conta	Funcional	Must	Permite ao usuário criar conta para se registrar nos eventos.	Eu como visitante quero criar uma conta.
	3	Visualizar eventos	Não funcional	Must	O visitante pode ver os eventos antes de se cadastrar; Gerar interesse a aplicação	Eu como visitante quero que o sistema me mostre os eventos cadastrados.
		Gerenciar dados	Funcional	Must	Após cadastrado, o usuário deve informar dados para contato;	Eu como usuário quero informar meus dados de contato.
		Criar eventos	Funcional	Must	O usuário pode registrar um ou mais eventos.	Eu como usuário quero cadastrar um evento.
		Editar evento	Funcional	Must	O usuário pode editar seu evento após o cadastro.	Eu como usuário quero gerenciar um evento criado por mim.
		Excluir evento	Funcional	Must	O usuário pode excluir eventos criados por ele anteriormente.	Eu como usuário quero excluir um evento que eu tenha criado
		Visualizar eventos de usuários	Funcional	Must	O administrador pode visualizar todos os eventos criados.	Eu como administrador quero ver os eventos cadastrados por usuários.
		Apagar eventos de usuários	Funcional	Must	O administrador pode apagar um evento, caso esteja inativo por muito tempo ou seja ofensivo de alguma forma, por exemplo.	Eu como administrador quero apagar eventos criados por usuários
		Visualizar usuários	Funcional	Must	O administrador pode visualizar a lista de usuários cadastrados no sistema;	Eu como administrador quero visualizar todos os usuários cadastrados.
		Banir usuário	Funcional	Must	O administrador pode banir um usuário que tenha uma má conduta.	Eu como administrador quero poder banir um usuário.

2.4 Sprints previstas

Tabela 3: Sprints previstas

Sistema: GamaHub – Sprints previstas			
# Sprint	Descrição	Objetivos	Composição de itens do backlog (Lista conforme tabela Backlog do produto)
1		Prototipação do projeto	
2		Criar sistema de login de usuário	Criar Conta
3		Criar home	Visualizar eventos
4			
5			
6			
7			
8			

Fonte: (Desenvolvedores, 2023)

3 Definição de Ready / Done

- Ready: Quando a equipe está apta a realizar a tarefa / Não necessita de um tempo para capacitação em questão, sua descrição está clara e alinhada com quem irá executar;
- Done: Quando a tarefa foi finalizada, testada e validada.

4 US – User Stories

- *Forneça uma previa das histórias de usuários previstas.*
- *Pesquisem a melhor forma de tabela-las, e indiquem em qual sprint ela será desenvolvida, o perfil de acesso, a técnica de elicitação, e quais critérios de Ready/done elas estão sujeitas.*

5 Diagrama de casos de uso

- *Forneça a visão de diagrama de casos de uso, contendo: os atores envolvidos, os casos de uso (grosso modo considerem casos de uso como as histórias de usuários que vcs definiram), os relacionamentos entre casos de uso e seus tipos (ex.: Include/extend) etc.*

6 MVP

- *Em todo o contexto do desenvolvimento definam o MVP – Minimun Product Viable proposto como entrega final do software pretendido, justificando suas escolhas. Caso existam U.S. excluídas do MVP, justifiquem e indiquem em quais manutenções evolutivas elas deveriam existir para a completude total do software proposto.*
- *O MVP será a entrega final de cada grupo e ele é o compromisso assumido pela equipe. Portanto, sejam coerentes e conscientes ao defini-lo.*

7 Referências