

Backlog do Produto

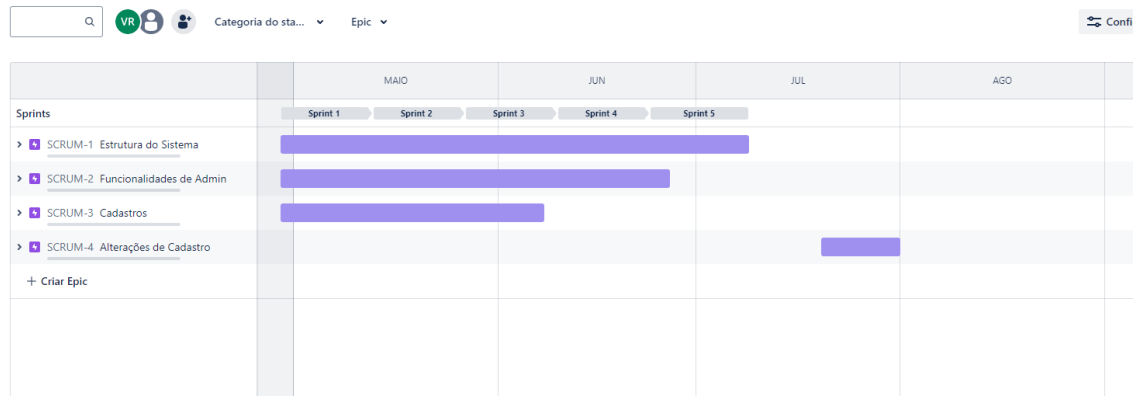
Épicos

Projetos / Jogos Internos

Linha do tempo

Enviar feedback

Compartilha



Épico – Estrutura do Sistema

O objetivo deste Épico é armazenar as histórias de Arquitetura e DBA (Enablers), que não representarão entregas que serão vistas pelo usuário, porém necessárias.

SCRUM-1 Estrutura do Sistema

SCRUM-5 Criação do projeto .Net

SCRUM-6 Criação do projeto FrontEnd

SCRUM-7 Criação das tabelas e seus relacionamentos

SCRUM-10 Implementação do Sonar

SCRUM-11 Implementação do Hotjar

SCRUM-12 Esteira CI/CD

SCRUM-40 Criação de pasta FTP para carregamento dos Atestados Médicos

SCRUM-41 Correções de BUGS encontrados na Homologação

TAREFAS PE...

TAREFAS PE...

TAREFAS PE...

TAREFAS PE...

TAREFAS PE...

TAREFAS PE...

TAREFAS PE...

TAREFAS PE...

Épico – Funcionalidades Admin

O objetivo deste Épico é armazenar as histórias referentes às funcionalidades das telas de Administrador do Sistema, e relatórios.

SCRUM-2 Funcionalidades de Admin		+
SCRUM-13	Front - Tela de Administrador - Categorias	TAREFAS PE...
SCRUM-20	API - Disponibilizar rotas CRUD para Categorias	TAREFAS PE...
SCRUM-14	Front - Tela de Administrador - Informações do Evento	TAREFAS PE...
SCRUM-21	API - Disponibilizar rotas CRUD para Informações do Evento	TAREFAS PE...
SCRUM-19	Front - Tela de Administrador - Tela de Aprovações	TAREFAS PE...
SCRUM-26	API - ROTA CRUD para aprovações	TAREFAS PE...
SCRUM-22	API - Disponibilizar rotas CRUD para cadastro de Administradores	TAREFAS PE...
SCRUM-15	Front - Tela de Administrador - Cadastro de Administradores	TAREFAS PE...
SCRUM-16	Front - Tela de Administrador - Relatórios - Torcedores	TAREFAS PE...
SCRUM-23	API - Rota Relatórios - Torcedores	TAREFAS PE...
SCRUM-17	Front - Tela de Administrador - Relatórios - Equipes	TAREFAS PE...
SCRUM-24	API - Rota Relatórios - Equipes	TAREFAS PE...
SCRUM-18	Front - Tela de Administrador - Relatórios - Atletas Individuais	TAREFAS PE...
SCRUM-25	API - Rota Relatórios - Atletas Individuais	TAREFAS PE...

Épico – Cadastros

Este épico tem como objetivo listar as histórias referentes às telas de Cadastros dos Colaboradores que desejam participar do evento Jogos Internos, contemplando os cadastros: Torcedor (colaborador que vai ao evento apenas para assistir), Atleta Individual (colaborador que participará de categorias individuais, como Natação e Corrida, por exemplo), Equipes (colaboradores que participarão de categorias coletivas, como Futebol e Basquete, por exemplo).

SCRUM-3 Cadastros	
SCRUM-27	Front-Tela Inicial com informações básicas do Evento
SCRUM-29	API - Disponibilizar rota com informações básicas do Evento
SCRUM-28	Front-Tela de Cadastro de Torcedores
SCRUM-30	API - Disponibilizar rota para carregamento de informações do colaborador
SCRUM-31	API - Disponibilizar rota para cadastro do Torcedor
SCRUM-34	Front - Tela para cadastro de Atletas Individuais
SCRUM-35	API - Disponibilizar rotas CRUD para Atletas Individuais
SCRUM-32	Front - Tela para cadastro de Equipes
SCRUM-33	API - Disponibilizar rotas CRUD para Equipes

Épico – Alterações de Cadastro

O objetivo deste épico é listar histórias referentes às telas de Edição dos Cadastros, que possibilitará que um colaborador mude seu cadastro dentro de um prazo estipulado pelo Administrador do sistema. Ex: Mudar a categoria de um esporte que deseja praticar, enviar outro atestado médico, excluir a inscrição, mudar a opção de transporte. Este épico será desenvolvido em uma Fase 2 do sistema.

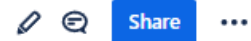
▼	✚	SCRUM-4 Alterações de Cadastro	
📌	SCRUM-36	Front - Tela de login para colaboradores cadastrados	TAREFAS PE...
📌	SCRUM-37	API - Login de usuários Cadastrados	TAREFAS PE...
📌	SCRUM-38	Front - Tela para edição do cadastro de Colaboradores	TAREFAS PE...
📌	SCRUM-39	API - Edição de Cadastro de Colaboradores	TAREFAS PE...

Obs: As informações de prioridades estarão no Backlog da Sprint.

Definition of Ready

Documento foi desenvolvido sendo dividido por equipe:

Definition of Ready



Owned by Vitor Henrique Vianna Reis

Last updated: cerca de uma hora atrás • 1 min read • See how many people viewed this page

Este documento descreve os critérios de Definition of Ready.

1. Equipe de Front-End

- Protótipo de baixa fidelidade pronto e validado com o usuário;
- Protótipo de baixa fidelidade pronto e entregue para os DEVs;
- História refinada pelo time de Desenvolvimento;
- História estimada pelo time de Desenvolvimento.

2. Equipe de Back-End

- História escrita utilizando modelo INVEST;
- História refinada pelo time de Desenvolvimento;
- História estimada pelo time de Desenvolvimento;
- Arquitetura desenhada;
- Ambiente de desenvolvimento pronto e DEVs com acesso.

3. Equipe de DBA

- História refinada pelo time de Desenvolvimento;
- História estimada pelo time de Desenvolvimento;
- Servidor online e DEVs com acesso;
- Diagrama de Classes desenhado.

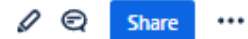
4. Equipe de Arquitetura

- História refinada pelo time de Desenvolvimento;
- História estimada pelo time de Desenvolvimento;
- Ferramenta de DevOps configurada e DEVs com acesso.

Definition of Done

Documento foi desenvolvido sendo dividido por equipe:

Definition of Done



Owned by Vitor Henrique Vianna Reis · ...

Last updated: 36 minutos atrás · 1 min read · See how many people viewed this page

Este documento descreve os critérios de Definition of Done.

1. Equipe de Front-End

- Testes Unitários validados;
- Critérios de aceite contemplados;
- Nenhum erro ao inspecionar pelo navegador;
- Funcionar devidamente nos navegadores Chrome, Firefox e Edge;
- Utilizar Angular, Bootstrap, e HTML5;
- Códigos revisados;
- P.O aceita história.

2. Equipe de Back-End

- Testes Unitários validados;
- Funcionar devidamente nos servidores das ferramentas DevOps;
- Estar escrito na plataforma .Net, versão 8;
- Utilizar ORM Entity Framework;
- Swagger funcional, Rotas e Atributos devidamente descritos;
- Critérios de aceite contemplados;
- Códigos revisados;
- P.O aceita a história;
- 90% mínimo de cobertura de código pelo SONAR (após o mesmo ser implementado).

3. Equipe de DBA

- Servidor online;
- Servidor de Banco de Dados a ser utilizado SQL Server;
- Ambientes DEV, QA, HMG e PRD equalizados;
- P.O aceita a história.

4. Equipe de Arquitetura

- Ferramenta DevOps devidamente configurada;
- Utilizar ferramenta Azure DevOps;
- P.O aceita a história.