

(Contexto) Sequenciador



Product Backlog

Projetos

MVP Sprint II - Sport Mais

...

Resumo

Backlog

Quadro

Código

Cronograma

Páginas

Formulários

Todos os tickets

Checklist

+

Pesquisar no backlog

Epic

SCRUM Sprint 1

Adicionar datas

(3 tickets)

21

0

0

Iniciar sprint

...

SCRUM-6

Cadastro na plataforma

CADASTRO E AUTENT...

BACKLOG

17 de out.

8

=

SCRUM-8

Criação de eventos esportivos

GESTÃO DE EVENTOS ...

BACKLOG

17 de out.

8

=

SCRUM-9

Cadastro em eventos esportivos

INSCRIÇÃO E PARTICI...

BACKLOG

17 de out.

5

=

+ Criar

3 tickets

Estimativa: 21

Backlog

(4 tickets)

31

0

0

Criar sprint

...

SCRUM-7

Calendário de eventos

GESTÃO DE EVENTOS ...

BACKLOG

31 de out.

8

=

SCRUM-11

Upload de fotos em massa

GESTÃO DE CONTEÚD...

BACKLOG

14 de nov.

13

=

SCRUM-12

Relatório de eventos

RELATÓRIOS E MONIT...

BACKLOG

31 de out.

5

=

SCRUM-10

Detalhes do evento

INSCRIÇÃO E PARTICI...

BACKLOG

31 de out.

5

=

+ Criar

Detalhamento do Sprint 1

Projetos

MVP Sprint II - Spor...

SCRUM-3

SCRUM-9

1

...

Cadastro em eventos esportivos

Descrição

COMO esportista QUERO visualizar eventos esportivos por categoria PARA decidir em qual vou inscrever-me.

COMO esportista QUERO selecionar um evento esportivo PARA inscrever-me.

COMO esportista QUERO selecionar uma forma de pagamento (boleto, pix ou cartão) PARA garantir minha vaga no evento.

Critérios de aceitação:

DADO QUE estou na tela listagem de eventos QUANDO seleciono uma categoria ENTÃO o sistema filtra os eventos da categoria escolhida

Requisitos não funcionais:

Apenas usuários autenticados poderão visualizar os detalhes completos dos eventos (localização, contato do organizador etc.).

A comunicação entre cliente e servidor deve ser criptografada (HTTPS/TLS).

A tela de listagem deve permitir filtros por categoria, data e localização de forma simples e intuitiva.

O tempo de carregamento da lista de eventos não deve exceder 2 segundos em conexões banda larga.

O sistema deve ser responsivo, garantindo boa visualização em smartphones, tablets e desktops.

Mensagens claras devem ser exibidas caso não haja eventos disponíveis na categoria escolhida.

O sistema deve estar disponível 99,5% do tempo mensal.

A listagem deve suportar pelo menos 5.000 eventos ativos sem degradação perceptível de desempenho.

O carregamento deve utilizar paginação ou scroll infinito, evitando sobrecarga ao carregar grandes quantidades de eventos.

Dados pessoais dos organizadores (telefone, e-mail direto) só devem ser exibidos se houver consentimento explícito, conforme a LGPD.

Os eventos visualizados pelo usuário podem ser registrados para fins estatísticos, respeitando a privacidade do mesmo.

O sistema deve suportar aumento progressivo do número de categorias de eventos sem necessidade de refatoração estrutural.

O mecanismo de busca e filtragem deve se adaptar a crescimento no volume de dados, mantendo performance aceitável.

Checklist

Definition of Done

OPEN

O código foi desenvolvido, revisado (code review) e integrado ao repositório principal.

OPEN

Os critérios de aceitação foram atendidos e testados (teste unitário, integração e/ou funcional).

OPEN

Os requisitos não funcionais foram respeitados (desempenho, segurança, usabilidade etc.).

OPEN

O código está coberto por testes automatizados com taxa mínima de cobertura definida (ex.: 70%).

OPEN

A funcionalidade foi testada em ambiente de homologação e validada pelo Product Owner.

OPEN

A documentação necessária foi atualizada (manual do usuário, changelog, README técnico etc.).

OPEN

Não há bugs críticos ou bloqueadores abertos relacionados à User Story.

OPEN

A funcionalidade está pronta para ser entregue em produção ou já implantada, dependendo da estratégia de release.

Definition of Ready

OPEN

Está claramente descrita no formato "Como [usuário] quero [ação] para [benefício]".

OPEN

Possui critérios de aceitação definidos (Given/When/Then ou equivalente).

OPEN

Tem os requisitos funcionais e não funcionais identificados.

OPEN

Dependências externas (ex.: integrações, aprovações) estão identificadas e documentadas.

OPEN

A equipe entende a prioridade da história no backlog.

OPEN

Wireframes/mockups ou descrições de interface estão disponíveis (quando aplicável).

SCRUM-2 / SCRUM-8

Criação de eventos esportivos

+@

Descrição

COMO organizador QUERO criar eventos esportivos PARA os esportistas

Critérios de aceitação:

DADO QUE

estou na tela de criação de eventos QUANDO preencher o formulário corretamente ENTÃO o cadastro do meu evento será realizado.

DADO QUE

estou na tela de criação de eventos QUANDO preencher o formulário incorretamente ENTÃO sistema irá retornar uma mensagem de erro.

Requisitos não funcionais:

Apenas usuários autenticados como **Organizadores** poderão acessar a funcionalidade de criação de eventos.

O envio dos dados deve ser feito obrigatoriamente via **HTTPS/TLS**.

Dados sensíveis do organizador (ex.: CPF/CNPJ, e-mail) devem ser armazenados em banco criptografado.

O formulário deve apresentar mensagens de erro claras e específicas para cada campo inválido.

Deve haver validação em tempo real para campos como data, local e categoria esportiva.

A interface deve ser responsiva, permitindo criação de eventos tanto em desktop quanto em dispositivos móveis.

O cadastro de eventos deve ser concluído em **até 3 segundos** em condições normais de rede.

O sistema deve suportar a criação simultânea de até **200 eventos por minuto** sem degradação perceptível.

A disponibilidade do serviço deve ser de **99,5% mensal**.

O sistema deve atender à **LGPD**, garantindo que os dados do evento (inclusive informações do organizador) só sejam usados para finalidades previstas em contrato.

Deve haver registro em **log de auditoria** com usuário, data e hora da criação/edição/exclusão de eventos.

A arquitetura deve permitir que novos tipos de eventos (ex.: corridas virtuais, torneios online) sejam adicionados sem alterações estruturais significativas.

O sistema deve comportar até **50.000 eventos cadastrados ativos** sem perda de performance.

Tickets filhos

...

...

...

0% concluído

07/19

+

-

...

Checklist

Definition of Done

☐

OPEN

O código foi desenvolvido, revisado (code review) e integrado ao repositório principal.

☐

OPEN

Os critérios de aceitação foram atendidos e testados (teste unitário, integração e/ou funcional).

☐

OPEN

Os requisitos não funcionais foram respeitados (desempenho, segurança, usabilidade etc.).

☐

OPEN

O código está coberto por testes automatizados com taxa mínima de cobertura definida (ex.: 70%).

☐

OPEN

A funcionalidade foi testada em ambiente de homologação e validada pelo Product Owner.

☐

OPEN

A documentação necessária foi atualizada (manual do usuário, changelog, README técnico etc.).

☐

OPEN

Não há bugs críticos ou bloqueadores abertos relacionados à User Story.

☐

OPEN

A funcionalidade está pronta para ser entregue em produção ou já implantada, dependendo da estratégia de release.

Definition of Ready

☐

OPEN

Está claramente descrita no formato "Como [usuário] quero [ação] para [benefício]".

☐

OPEN

Possui critérios de aceitação definidos (Given/When/Then ou equivalente).

☐

OPEN

Tem os requisitos funcionais e não funcionais identificados.

☐

OPEN

Dependências externas (ex.: integrações, aprovações) estão identificadas e documentadas.

Projetos / MVP Sprint II - Spor... / SCRUM-1 / SCRUM-6

Cadastro na plataforma

+@

Descrição

COMO usuário QUERO me cadastrar PARA ter acesso à plataforma.

Critérios de aceitação:

DADO QUE

estou na tela de cadastro QUANDO preencher todos os dados corretamente ENTÃO meu cadastro será realizado

DADO QUE

estou na tela de cadastro QUANDO preencher a senha num formato inválido ENTÃO uma mensagem de erro é retornada

DADO QUE

estou na tela de cadastro QUANDO preencher com um e-mail duplicado ENTÃO uma mensagem de erro é retornada

DADO QUE

estou na tela de cadastro QUANDO preencher com um documento ENTÃO minha identidade será verificada

Requisitos não funcionais:

O sistema deve armazenar senhas utilizando algoritmo de hash seguro (ex.: bcrypt, Argon2).

A comunicação entre cliente e servidor deve ser criptografada por HTTPS/TLS.

O sistema deve bloquear tentativas de cadastro após **5 falhas consecutivas** de preenchimento suspeito, exibindo mensagem apropriada.

Mensagens de erro devem ser claras, em português, e indicar qual campo precisa de correção.

O formulário deve validar os campos em tempo real (ex.: e-mail com formato inválido, senha fraca).

O processo de cadastro deve ser concluído em **no máximo 3 etapas**.

O tempo de resposta do sistema, após envio do formulário, não deve exceder **2 segundos** em condições normais de rede.

O serviço de cadastro deve estar disponível **99,5% do tempo mensal**.

O sistema deve seguir a **LGPD (Lei Geral de Proteção de Dados)**, garantindo consentimento explícito do usuário para armazenamento e uso de seus dados.

Os dados sensíveis (documentos, e-mails, senhas) devem ser armazenados em conformidade com normas de segurança de dados.

A solução deve permitir o cadastro simultâneo de até **500 usuários por minuto** sem degradação perceptível da performance.

...

...

...

07/19

+

-

...

Checklist

Definition of Done

☐

OPEN

O código foi desenvolvido, revisado (code review) e integrado ao repositório principal.

☐

OPEN

Os critérios de aceitação foram atendidos e testados (teste unitário, integração e/ou funcional).

☐

OPEN

Os requisitos não funcionais foram respeitados (desempenho, segurança, usabilidade etc.).

☐

OPEN

O código está coberto por testes automatizados com taxa mínima de cobertura definida (ex.: 70%).

☐

OPEN

A funcionalidade foi testada em ambiente de homologação e validada pelo Product Owner.

☐

OPEN

A documentação necessária foi atualizada (manual do usuário, changelog, README técnico etc.).

☐

OPEN

Não há bugs críticos ou bloqueadores abertos relacionados à User Story.

☐

OPEN

A funcionalidade está pronta para ser entregue em produção ou já implantada, dependendo da estratégia de release.

Definition of Ready

☐

OPEN

Está claramente descrita no formato "Como [usuário] quero [ação] para [benefício]".

☐

OPEN

Possui critérios de aceitação definidos (Given/When/Then ou equivalente).

☐

OPEN

Tem os requisitos funcionais e não funcionais identificados.

☐

OPEN

Dependências externas (ex.: integrações, aprovações) estão identificadas e documentadas.

☐

OPEN

A equipe entende a prioridade da história no backlog.

☐

OPEN

Wireframes/mockups ou descrições de interface estão disponíveis (quando aplicável).

(Extra) Timeline - MVP + Incrementos

