

CENTRO UNIVERSITÁRIO SENAI SANTA CATARINA
UNIDADE CURRICULAR: PROJETO APLICADO II
PROFESSOR: JANICE DETERS

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas



DOCUMENTO DO SISTEMA GLITCH

EQUIPE 15
ANA LÍGIA VINCENZI BORTOLOTTI
HENRY MAGNO COUTINHO FURTADO DE MENDONÇA
LETÍCIA ZALASIK
LUCAS DE ALMEIDA ALVES
SÁVIO EDUARDO ZOBOLI

Florianópolis/SC

Histórico de Versões

Data	Versão	Descrição	Autor	Revisor
08/08/2025	1.0	Elaboração inicial do documento	Sávio Eduardo Zoboli	Ana Lígia V.B.
08/08/2025	1.1	Preenchimento da seção Escopo, Descrição da solução e Recursos adicionais	Sávio Eduardo Zoboli	Ana Lígia V.B, Letícia Zalasik.
11/08/2025	1.2	Elencado requisitos funcionais, não funcionais e restrições	Sávio Eduardo Zoboli Ana Lígia V. B.	Ana Lígia V. B. Letícia Zalasik Sávio Eduardo Zoboli
29/08/2025	1.3	Produção do Modelo Entidade Relacionamento	Sávio Zoboli	

Objetivos de cada projeto aplicado:

Projeto aplicado I – protótipo da solução

Projeto Aplicado II – aplicação integrando com banco de dados

Projeto Aplicado III – aplicação CLIENTE /SERVIDOR (web)

Projeto Aplicado IV – aplicação CLIENTE /SERVIDOR ou mobile (web e mobile) com testes unitários

1. Escopo

Com o crescimento da indústria de eSports, um sistema de organização de eventos de games facilita o cadastro e gerenciamento de competições, jogadores e organizadores. O sistema permitirá a inscrição em eventos, o gerenciamento de torneios, a criação de equipes e a comunicação entre todos os participantes, proporcionando uma experiência estruturada e otimizada para competições online ou presenciais.

O sistema permitirá que organizadores criem e gerenciem eventos de jogos, como torneios e campeonatos, com opções de personalização de regras, formato e premiação. Os jogadores poderão se cadastrar, formar equipes e se inscrever em múltiplos eventos. O sistema atualizará automaticamente os resultados das partidas e exibirá as classificações ao vivo. Também será possível configurar notificações para manter todos os participantes informados sobre mudanças de horário, locais e resultados. Para os organizadores, o sistema proporcionará um painel de controle completo para visualizar inscrições, status de torneios e gerar relatórios pós-evento.

Link do SAGA SENAI:

<https://plataforma.gpinovacao.senai.br/plataforma/demandas-da-industria/interna/11035>

2. Descrição da Solução

2.1. Contextualização

Os jogos digitais estão deixando de ser apenas entretenimento e passando a ser considerados como meio de renda para a população no geral. Os *e-sports* estão cada vez mais presentes em nosso cotidiano, dando oportunidade para pessoas que querem trabalhar com o que gostam. A pesquisa de 2024 da PGB (Pesquisa Game Brasil) evidenciou que de sua amostra, 73,9% dos participantes jogam jogos digitais e destes 82,1% conhecem *e-sports* (TERRA, 2024).

Ainda falando de dados estatísticos, no ano de 2019, 450 milhões de pessoas assistiram a campeonatos de jogos eletrônicos segundo o levantamento da Newzoo (GLOBO ESPORTE, 2023), sendo que o Brasil ficou em terceiro no ranking de países com mais espectadores, com um total de 21,2 milhões de pessoas registradas.

Todos esses dados demonstram que o cenário dos e-sports tanto no Brasil quanto no mundo está em ascensão, garantindo mercado para diversas iniciativas como a criação de jogos competitivos, sistemas de gerenciamento e mercado nichado, aumentando a economia e circulação de capital.

2.2. Descrição do Problema

Com o crescimento da indústria de *games*, a criação de sistemas e métodos de gerenciamento de torneios e seus participantes é muito importante, pois automatiza e padroniza o tratamento de dados, agendamentos e relatórios, além de automatizar processos de notificação e premiação.

Essa é a premissa da demanda encontrada na plataforma Saga SENAI, que é uma iniciativa do SENAI de Tocantins e visa desenvolver um sistema com diversos critérios, para auxiliar no controle e gerenciamento de eventos de *e-sports*.

Os eventos, torneios e equipes de e-sports por vezes podem ser não formalizadas, ou obter um grau de organização bem baixo. É isso que a demanda visa resolver. Desenvolver um sistema de gerenciamento de torneios e equipes de e-sports, garantindo a participação autônoma de empresas organizadoras, organizadores independentes, jogadores e equipes profissionais, é uma

ótima forma de inserção no mercado de desenvolvimento de sistemas, além de resolver um problema real da indústria de jogos digitais.

2.3. Impacto

As pessoas impactadas por essa solução seriam tanto os participantes dos campeonatos, quanto seus idealizadores e demais profissionais envolvidos na organização do evento. Pois isso traria mais agilidade e organização e aos processos, facilitando tanto para os participantes o acesso às competições quanto para os organizadores do evento que teriam essa ferramenta de auxílio. Como as competições podem se tornar mais acessíveis e organizadas, este mercado pode atrair ainda mais participantes aos campeonatos, patrocínios aos eventos, mais vendas de jogos e materiais relacionados a prática de *e-sports*. A pesquisa da PGB (TERRA, 2024) traz um dado relevante sobre isso mostrando que 54,5% dos participantes consideram importante comprar produtos de marcas que patrocinam o seu time de e-sports preferido.

2.4. Justificativa

A demanda foi escolhida pois consideramos que este é um mercado que possui uma forte tendência a crescimento e por alguns participantes do grupo participarem de jogos online tendo conhecimento do tema. Essa tendência de crescimento pode ser observada através dos dados da Newzoo (GLOBO ESPORTE, 2023): em 2019 os *e-sports* tiveram faturamento de US\$957,5 milhões, em 2020 por aspectos relacionados a pandemia do covid-19 tiveram pequena queda de 1,1%. Mas em 2021, voltou a crescer com mais de US\$1 bilhão de faturamento e o esperado para 2024 era de US\$1,6 bilhões de faturamento.

2.5. Solução

A solução será desenvolver um sistema web e mobile com base nas necessidades do cliente e que atendam aos padrões de qualidade e usabilidade. O software será desenvolvido em linguagens de programação atuais e consolidadas no mercado, como AngularJS (Framework TypeScript para desenvolvimento web), Node.JS (framework JavaScript para desenvolvimento backend) e PostgreSQL (banco de dados conhecido por ser eficiente e confiável).

A solução de interface será, para testes, um servidor local com docker, separando cada parte da aplicação (front-end, back-end e banco de dados) em containers separados. Para a infraestrutura de produção, serão utilizados serviços da AWS (Amazon Web Services), sendo para o front-end o Amazon Elastic Beanstalk (que é um PaaS para implantação de sites em ambientes

web), para o back-end o Amazon ECS (Elastic Container Services, que é um IaaS para containers de aplicações) e para o banco de dados o Amazon RDS (Relational Database Services, que é um PaaS que cria uma instância do banco de dados). Para armazenamento dos segredos (usuário e senha do banco de dados e outros dados sensíveis do sistema) será utilizado o serviço Amazon Secrets Manager.

3. Partes Interessadas

3.1. Coordenação do Curso de Análise de Sistemas

Aline Cristina Antoneli

3.2. Professor da UC Projeto Aplicado II

Janice Ines Deters

3.3. Cliente da aplicação

Demanda cadastrada pelo SENAI de Paraíso do Tocantins.

4. Atores

4.1. Jogador

Participantes dos torneios de e-sports.

4.2. Organizador

Usuários que organizam e gerenciam o torneio, aplicando regras personalizadas.

4.3. Espectador

São os usuários que acompanham resultados das partidas e desempenho dos jogadores.

5. Requisitos

5.1. Requisitos Funcionais

RF001: O sistema deve permitir que o usuário realize seu cadastro na plataforma utilizando nome completo, nome de usuário (opcional, caso não informado o nome completo será mostrado em dashboards), data de nascimento, CPF, e-mail e senha.
RF002: O sistema deve permitir que o usuário faça login no sistema, utilizando e-mail e senha.
RF003: O sistema deve permitir que o usuário redefina sua senha ao informar o e-mail cadastrado, enviando um link de recuperação para criação de uma nova senha.
RF004: O sistema deve permitir que o usuário faça login utilizando plataformas externas disponíveis (Google, Steam e Epic Games).
RF005: O sistema deve permitir que o usuário edite seu perfil, incluindo nome de usuário, avatar e vinculação/desvinculação de contas externas.
RF006: O sistema deve permitir acesso diferenciado com permissões específicas para usuários do tipo Organizador, Jogador e Administrador.
RF007: O sistema deve permitir que o usuário veja novos torneios disponíveis para inscrição, filtrando por jogo e data.
RF008: O sistema deve permitir que o usuário possa se inscrever em torneios.

RF009: O sistema deve permitir que o usuário crie novos torneios, definindo nome, local (online, presencial com endereço), jogo, público-alvo (amador, profissional), data de realização, data limite de inscrição, quantidade de jogadores, formação de grupos, quantidade de etapas, modo de classificação (chaves ou pontuação).
RF010: O sistema deve permitir que o administrador configure um limite máximo de eventos ativos por organizador.
RF011: O sistema deve permitir que o torneio possua uma taxa de inscrição.
RF012: O sistema deve permitir que o usuário realize pagamentos de inscrições por meio de integração com provedores de pagamento, registrando a confirmação ou falha da transação.
RF013: O sistema deve permitir que o usuário receba notificações de lembretes sobre torneios que está inscrito, seguindo ou organizando, em horários configuráveis (24h, 1h e 10min antes do início).
RF014: O sistema deve permitir que o usuário crie, gerencie e participe de equipes, incluindo envio e resposta a convites.
RF015: O sistema deve permitir que o organizador gerencie inscrições, aprovando ou recusando participantes e controlando fila de espera.
RF016: O sistema deve permitir que o organizador cadastre e edite patrocinadores, incluindo informações como nome, logo e link.
RF017: O sistema deve disponibilizar um dashboard de eventos para organizadores, apresentando informações como quantidade de inscritos, status das partidas, datas e resultados.
RF018: O sistema deve permitir que o organizador defina e gerencie as regras do torneio, incluindo formato, critérios de desempate, penalidades, restrições de entrada, classificação de jogadores (amador, semi-profissional, profissional) e outras especificações.
RF019: O sistema deve permitir que o usuário visualize transmissões ao vivo de eventos integrados via plataformas externas.
RF020: O sistema deve permitir que o usuário acompanhe chaves e brackets atualizados em tempo real.
RF021: O sistema deve permitir que o usuário visualize rankings filtrando por jogo, período e plataforma.
RF022: O sistema deve permitir que o usuário acesse seu painel pessoal, contendo próximos eventos, status das inscrições e histórico de participações.
RF023: O sistema deve permitir exportar compromissos para arquivo de calendário (.ics).
RF024: O sistema deve permitir troca de mensagens internas com destaque para comunicados de organizadores.
RF025: O sistema deve permitir que o usuário gerencie e personalize suas notificações in-app.
RF026: O sistema deve permitir que os participantes enviem avaliações e feedback sobre eventos concluídos, incluindo notas e comentários.
RF027: O sistema deve permitir que o usuário pesquise eventos, equipes, jogadores e organizadores utilizando filtros como nome, jogo, data e localização.
RF028: O sistema deve permitir que usuários e organizadores enviem imagens ou arquivos

relevantes para eventos, equipes ou perfil, com validação de formato e tamanho.
RF029: O sistema deve enviar notificações por e-mail para ações importantes, como confirmação de inscrição, alterações no evento ou mensagens do organizador.
RF030: O sistema deve permitir que o organizador gere relatórios de participação, desempenho e engajamento dos eventos, exportáveis em formatos como PDF ou CSV.
RF031: O sistema deve permitir que sejam declarados pelo organizador os moderadores dos torneios. Esses moderadores, assim como organizadores, podem remover conteúdo inadequado enviado por usuários.
RF032: O sistema deve registrar o histórico de alterações feitas em eventos, equipes ou perfis, permitindo que organizadores e administradores acompanhem modificações.
RF033: O sistema deve permitir que o usuário compartilhe eventos, resultados e conquistas diretamente em redes sociais.
RF034: O sistema deve dispor de uma dashboard administrativa para o administrador do sistema, onde será possível visualizar os eventos do dia, quantidade de usuário esperados e valores arrecadados e retidos.
RF035: O sistema deve dispor de uma página de cadastro de jogos, onde serão cadastrados os jogos suportados pelo sistema, contendo nome, desenvolvedora e modo de conexão.
RF036: O sistema deve repassar o valor das inscrições para o organizador logo após a finalização do evento, caso não possua denúncias contra a organização.
RF037: O sistema deve incluir na taxa de inscrição ou repasse o valor da taxa da plataforma de pagamentos e taxa de uso do site, definido nas regras financeiras.
RF038: O sistema deve permitir que o usuário solicite a exclusão permanente da conta por meio da página de edição de perfil.
RF039: Ao selecionar o país da pessoa, o sistema deve mostrar dinamicamente os documentos obrigatórios para cadastro
RF040: O sistema deve permitir a atribuição de juízes, que devem ser outros usuários que não fazem parte dos jogadores do torneio. Caso não seja atribuído nenhum, o organizador do torneio será o juiz.
RF041: Ao remover ou desativar um torneio, o sistema deve automaticamente remover esse torneio dos favoritos do usuário, apresentando uma confirmação visual em tela sobre a operação.
RF042: Ao remover ou desativar um torneio, o sistema deve automaticamente remover esse torneio dos agendamentos do usuário, apresentando uma confirmação visual em tela sobre a operação.
RF043: Caso um torneio que já possua inscritos queira ser removido, o sistema não pode removê-lo, apenas desativá-lo.
RF044: Ao desativar um torneio, os jogadores inscritos devem ser desinscritos, notificados e o valor pago na inscrição (se aplicável) deve ser estornado por meio da API de pagamentos.
RF045: O sistema deve apresentar seus termos de uso de maneira clara para que o usuário aceite ou recuse ao se cadastrar na plataforma.
RF046: O sistema deve possuir uma tela de ajuda com dúvidas frequentes a respeito da

plataforma.
RF047: O sistema deve dispor de uma funcionalidade de tickets e denúncias, onde pode-se reportar resultados de partidas divergentes ou comportamentos incompatíveis com as normas da plataforma. Essas denúncias serão analisadas por um usuário administrador.
RF048: O sistema deve permitir que o administrador, ao resolver uma denúncia, possa: (a) editar o resultado de uma partida, (b) aplicar uma advertência ao usuário, (c) suspender temporariamente o usuário ou (d) banir permanentemente o usuário da plataforma.

5.2. Requisitos Não Funcionais

Requisito Não Funcional	Categoria
RNF001: O sistema deve ser compatível com diferentes plataformas, dispositivos e navegadores, restritos às últimas versões disponíveis em maio de 2023 dos navegadores Google Chrome, Microsoft Edge, Safari e Opera.	Compatibilidade
RNF002: O sistema deve implementar autenticação segura utilizando tokenização ou métodos equivalentes, garantindo proteção contra acessos não autorizados e prevenindo ataques.	Segurança
RNF003: O sistema deve garantir que todas as transações financeiras sejam processadas em canais criptografados (SSL/TLS) e por provedores certificados, assegurando a integridade e a confidencialidade das informações de pagamento.	Segurança
RNF004: O sistema deve garantir que a exportação de compromissos utilize o fuso horário exato do evento.	Confiabilidade
RNF005: O sistema deve armazenar e transmitir dados de usuários de forma criptografada, conforme a LGPD.	Segurança
RNF006: O sistema deve garantir que permissões sejam aplicadas corretamente de acordo com o tipo de usuário (Organizador ou Jogador).	Segurança
RNF007: O sistema deve carregar páginas e componentes em até 3 segundos.	Desempenho
RNF008: O sistema deve permitir atualização em tempo real. Refletir mudanças em placares, brackets e inscrições sem a necessidade de recarregar a página.	Desempenho
RNF009: Manter o sistema disponível 99,2% do tempo, exceto em períodos programados de manutenção.	Disponibilidade
RNF010: O sistema deve ajustar automaticamente a interface para diferentes tamanhos e resoluções de tela.	Usabilidade
RNF011: O sistema deve organizar menus e fluxos de forma que o usuário consiga concluir tarefas principais em até 3 cliques a partir da página inicial.	Usabilidade
RNF012: O sistema deve manter cores, fontes e espaçamentos consistentes em todas as páginas e componentes.	Usabilidade

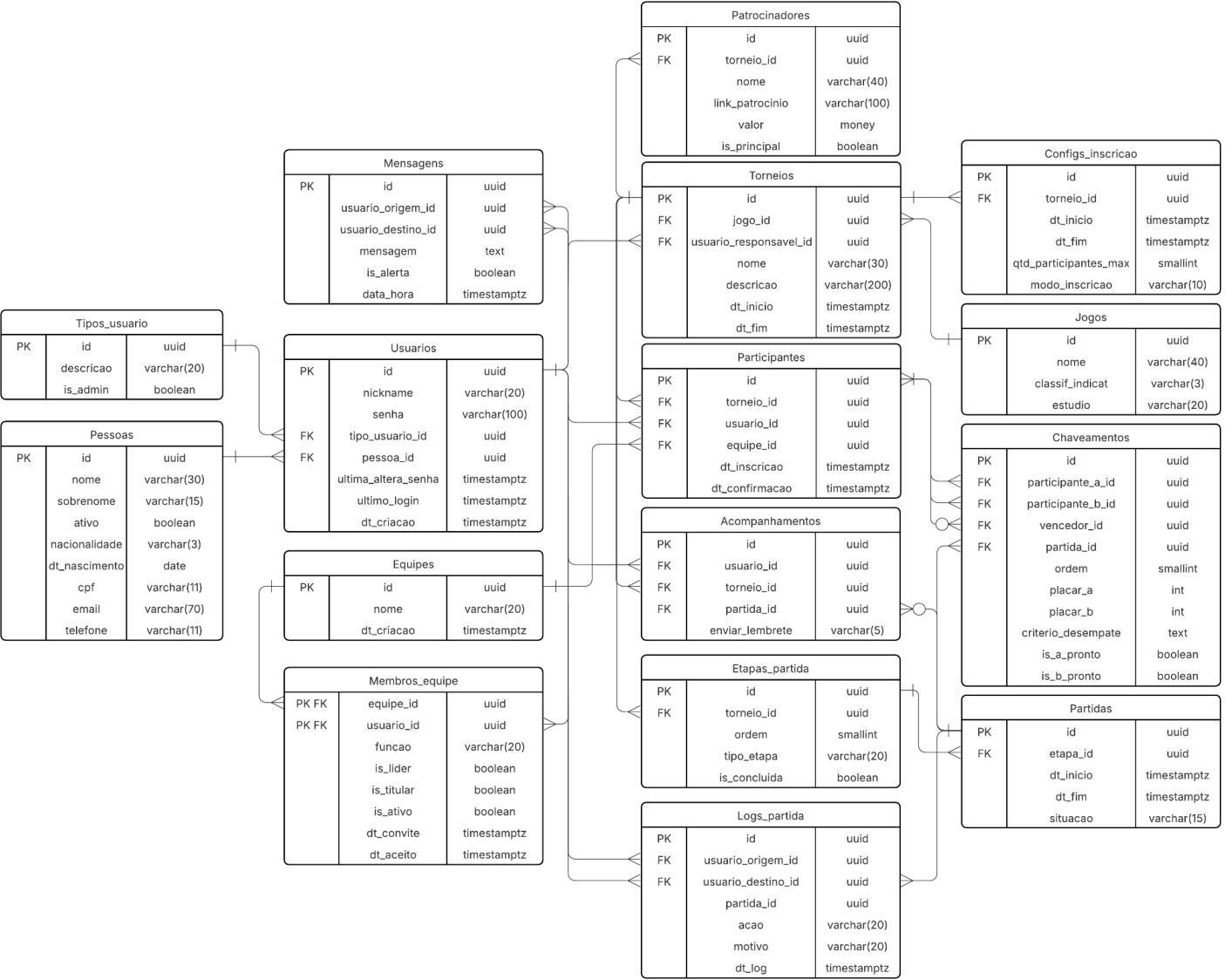
RNF013: Permitir aumento de capacidade do sistema sem interrupção de serviço.	Escalabilidade Manutenção	e
RNF014: Utilizar arquitetura que facilite manutenção e adição de novas funcionalidades sem impactar recursos existentes.	Escalabilidade Manutenção	e
RNF015: Interface adaptável a dispositivos móveis, tablets e desktops, com breakpoints para diferentes resoluções.	Usabilidade Acessibilidade	e
RNF016: Garantir contraste mínimo 4.5:1, navegação por teclado, foco visível e rótulos adequados para inputs.	Usabilidade Acessibilidade	e
RNF017: Mensagens de ajuda, ilustrações para estados vazios e tooltips explicativos.	Usabilidade Acessibilidade	e
RNF018: Indicadores claros para carregamento, sucesso, erro e salvamento.	Usabilidade Acessibilidade	e
RNF019: Tempo de carregamento da página principal $\leq 3s$ em redes 4G.	Desempenho	
RNF020: Utilizar cache para otimizar carregamento de imagens e scripts.	Desempenho	
RNF021: Mensagens amigáveis e opção de tentar novamente em caso de falha de conexão.	Confiabilidade	
RNF022: Reconexão automática para eventos em tempo real, com até 3 tentativas.	Confiabilidade	
RNF023: Uso consistente de cores, tipografia, espaçamento e componentes.	Consistência Visual	
RNF024: Alternância entre modos claro e escuro com preferência salva pelo usuário.	Consistência Visual	
RNF025: Uso de HttpOnly cookies ou memória protegida para tokens, prevenindo XSS (Cross-Site Scripting) e CSRF (Cross-Site Request Forgery).	Segurança	
RNF026: Limpeza total de dados sensíveis ao encerrar sessão e ocultar ações não permitidas.	Segurança	
RNF027: Interface funcional com até 200% de zoom sem quebra de layout.	Compatibilidade	
RNF028: Apenas cookies essenciais, sem coleta de dados de uso por padrão.	Privacidade	
RNF030: O banco de dados deve ter um uptime mínimo de 99,2%.	Disponibilidade	
RNF031: O banco de dados deve conseguir realizar uma ação de escrita (create/update) em menos de 100 milissegundos.	Desempenho	
RNF032: O banco de dados deve conseguir retornar dados de uma instrução select em menos de 1 segundo.	Desempenho	
RNF033: O acesso ao banco de dados deve ser restrito e seguir o princípio do mínimo privilégio.	Segurança	
RNF034: O usuário do banco de dados “usuario_api” terá permissões de INSERT, UPDATE, DELETE e SELECT.	Segurança	
RNF035: O usuário do banco de dados “admin_api” terá permissões de	Segurança	

CREATE, ALTER, INSERT, UPDATE, DELETE, DROP e SELECT, e deverá ser utilizado apenas pelo ORM (Sequelize)	
RNF036: O sistema deve garantir que apenas usuários autorizados possam acessar o banco de dados, respeitando perfis de permissão distintos (ex.: usuários de aplicação e administradores).	Segurança
RNF037: Todos os dados considerados sensíveis segundo a LGPD devem ser armazenados com criptografia enquanto estiverem em repouso (at rest) no banco de dados.	Conformidade
RNF038: Os dados devem permitir anonimização ou exclusão definitiva conforme solicitação do titular, de acordo com as restrições legais da LGPD.	Conformidade
RNF039: O banco de dados não deve possuir colunas ou campos que armazenem dados pessoais sensíveis sem a devida finalidade.	Conformidade
RNF040: Os logs de transações do banco de dados devem ser armazenados pelo período mínimo de 6 meses, em conformidade com o Marco Civil da Internet.	Conformidade
RNF041: O banco de dados deve apenas aceitar requisições na sua porta padrão (5432) por meio da rede privada da aplicação. A porta não deve ser exposta na Internet.	Segurança
RNF042: Configurações de acesso ao banco de dados (usuário, senha, host e porta) devem ser gerenciados por um Sistema Gerenciador de Segredos.	Segurança
RNF043: As senhas devem seguir a política de no mínimo 8 caracteres, contendo ao menos uma letra maiúscula, uma letra minúscula, um número e um caractere especial.	Segurança
RNF044: Deve haver backup diário dos dados do sistema.	Manutenção
RNF045: A API de produção deve estar acessível por meio de um proxy reverso.	Segurança
RNF046: O banco de dados deve armazenar o CPF com a restrição UNIQUE.	Integridade
RNF047: Ao adicionar um novo usuário, o banco de dados deve armazenar o timestamp da criação.	Integridade
RNF048: Ao alterar um usuário, o banco de dados deve armazenar o timestamp da última alteração.	Integridade
RNF049: Antes de inserir um novo usuário o sistema deve verificar se já há um usuário com aquele CPF.	Integridade
RNF050: Antes de alterar o nome de usuário, o sistema deve verificar se já não há um usuário com aquele nome.	Integridade
RNF051: O sistema deve possuir proteção contra acesso à rotas restritas no sistema, validando se o usuário está autenticado. Caso não esteja autenticado, ele deve ser automaticamente redirecionado à página de login.	Segurança

5.3. Restrições

<i>Restrição</i>
<i>RST001: Não há garantia de funcionamento para versões defasadas de browsers (anteriores a 2020)</i>
<i>RST002: Uploads de imagens serão limitados a 8MB e a tipos permitidos .jpg, .png. Arquivos serão aceitos apenas .pdf de no máximo 12MB.</i>
<i>RST003: A plataforma não tolera conteúdo ilegal. Organizadores e moderadores são responsáveis pela remoção e denúncia.</i>
<i>RST004: Após 5 tentativas de login incorretas consecutivas a plataforma bloqueia novas tentativas por 10 minutos ou até o usuário seguir o processo de recuperação de senha.</i>
<i>RST005: As sessões de usuários expiram em 60 minutos de inatividade.</i>
<i>RST006: A plataforma deve suportar até 1.000 usuários simultâneos sem degradação de performance na primeira versão.</i>
<i>RST007: Para cadastrar uma taxa de inscrição, o organizador deve possuir CNPJ ativo e ter cadastrado os dados de pagamento (banco, agência e conta). A conta deve estar em titularidade da empresa organizadora.</i>
<i>RST008: O sistema só permitirá o cancelamento de eventos com até 10 dias de antecedência caso possua taxa de inscrição e com 5 dias de antecedência caso seja gratuito.</i>
<i>RST009: Denúncias recebidas devem ser respondidas em até 2 dias úteis.</i>
<i>RST010: Resultados de partidas denunciadas serão congelados até resposta da administração.</i>

6. Diagrama de Modelo Entidade e Relacionamento



7. Diagrama de UML

<serão informados de acordo a evolução dos projetos aplicados>

7.1 Diagrama de Use-Case

<essa seção deverá ser preenchido nos **Projeto Aplicado II e III e IV**>

<inserir o diagrama de Use-case da solução conforme orientação da UC Arquitetura e Modelagem de Sistemas . Caso a apresentação não fique legível mudar a orientação da página para paisagem>

7.2 Diagrama de Sequência

<essa seção deverá ser preenchido nos **Projeto Aplicado III e IV**>

<inserir o diagrama de sequência da solução conforme orientação da UC Arquitetura e Modelagem de Sistemas . Caso a apresentação não fique legível mudar a orientação da página para paisagem>

7.3 Diagrama de Atividades

<essa seção deverá ser preenchido nos **Projeto Aplicado III e IV**>

<inserir o diagrama de atividades da solução proposta conforme orientação da UC Arquitetura e Modelagem de Sistemas . Caso a apresentação não fique legível mudar a orientação da página para paisagem>

7.3 Diagrama de Classes

<essa seção deverá ser preenchido nos **Projeto Aplicado IV**>

<inserir o diagrama de classes da solução proposta conforme orientação da UC Arquitetura e Modelagem de Sistema

7.4 Diagrama de Componentes

<essa seção deverá ser preenchido nos **Projeto Aplicado III e IV**>

<inserir o diagrama de componentes da solução proposta conforme orientação da UC Arquitetura e Modelagem de Sistema

7.4 Diagrama de Implantação

<essa seção deverá ser preenchido nos **Projeto Aplicado III e IV**>

<inserir o diagrama de implantação da solução proposta conforme orientação da UC Arquitetura e Modelagem de Sistema

8. Proposta de Solução Tecnológica Escolhida

< Essa seção deverá ser preenchido nos **Projeto Aplicado II, III e IV**

Descreva a solução tecnológica adotada, a razão pela escolha desta ferramenta>

Exemplo: Como banco de dados da aplicação foi utilizado PostgreSQL pelo fato de ...>

8.1 Topologia da Solução

Essa seção é **optativo** inserir o diagrama da topologia da arquitetura da solução.

9. Recursos Adicionais

9.1. Planejamento do Projeto Aplicado

Visão geral do gerenciamento, compartilhado com o e-mail institucional da professora mentora.

<https://app.clickup.com/9014963053/v/b/7-9014963053-2>

Visão de cada view, com permissão de apenas visualização sobre cada uma delas

Tarefas de Documentação:

<https://sharing.clickup.com/9014963053/l/h/6-901411890825-1/13aef038fd938db>

Tarefas do FrontEnd:

<https://sharing.clickup.com/9014963053/l/h/6-901411914508-1/3015ab810cc3e5d>

Tarefas de BackEnd:

<https://sharing.clickup.com/9014963053/l/h/6-901411914593-1/b3cd7bbda7f0b8e>

Tarefas de Banco de Dados:

<https://sharing.clickup.com/9014963053/l/h/6-901411914610-1/c782ae39333198b>

Tarefas de infraestrutura:

<https://sharing.clickup.com/9014963053/l/h/6-901411946277-1/ff44b7d648a1a43>

9.2. Repositório da Aplicação

github.com/saviozoboli/glitch