

JOÃO VICTOR OLIVEIRA
JOÃO VITOR VIEIRA
KEVIN WILLIAMS
MANUELA GONZALEZ
PAMELA CAMARGO
TAWANE DE SOUZA

GXSPORT: gerenciador de torneios

São Paulo - SP
2024

JOÃO VICTOR OLIVEIRA
JOÃO VITOR VIEIRA
KEVIN WILLIAMS
MANUELA GONZALEZ
PAMELA CAMARGO
TAWANE DE SOUZA

GXSPORT: gerenciador de torneios

Trabalho de Conclusão de Curso apresentado ao curso Técnico em Desenvolvimento de Sistemas da Etec Albert Einstein, orientado pela Prof^a Cristiane Tuji da Silva, como requisito parcial para obtenção do título de técnico em Desenvolvimento de Sistemas.

São Paulo - SP
2024

FOLHA DE APROVAÇÃO

X

Cristiane Tuji da Silva
Professor Orientador

X

X

Nome:
Professor Avaliador

X

Nome:
Professor Avaliador

X

Nome:
Professor Avaliador

RESUMO

O projeto tem como objetivo a criação de um aplicativo móvel voltado para a organização e gerenciamento de torneios esportivos amadores, respondendo à crescente demanda por soluções digitais eficientes nessa área. Atualmente, a gestão de campeonatos em pequenas e médias escalas, como jogos de futebol de bairro ou competições esportivas entre amigos, é frequentemente realizada de forma manual e desorganizada, por meio de planilhas, aplicativos de mensagens e redes sociais. Isso pode resultar em dificuldades na coordenação, erros no acompanhamento de jogos e falta de clareza nos resultados, afetando a experiência dos organizadores e participantes.

A solução proposta é um aplicativo que centraliza todo o processo de organização de torneios, automatizando tarefas essenciais como a inscrição de times e jogadores, o sorteio de grupos e a geração de tabelas de classificação. O app oferecerá uma interface simples e intuitiva, projetada para facilitar a criação de competições e o acompanhamento dos resultados em tempo real. Além disso, ele fornecerá notificações e atualizações constantes, garantindo que todos os envolvidos estejam sempre informados sobre o andamento do campeonato.

Entre as funcionalidades gratuitas, o aplicativo permitirá a escolha do tipo de campeonato (pontos corridos ou mata-mata), o sorteio de partidas, e o gerenciamento de categorias. Já na versão paga, estarão disponíveis funções mais avançadas, como estatísticas detalhadas de jogadores (gols, assistências, cartões), gestão de arbitragem, calendário de jogos e documentação de atletas com fotos, permitindo um nível maior de controle e profissionalismo para os organizadores.

Além de abranger esportes físicos, o aplicativo também terá suporte para competições de e-Sports, incluindo jogos eletrônicos como FIFA, PES e MOBAs, ampliando seu alcance e incentivando o desenvolvimento do esporte eletrônico. Com isso, a plataforma se posiciona como uma solução completa, que atende tanto ao

público amador quanto àqueles que organizam torneios eletrônicos, promovendo inclusão e diversidade esportiva.

Em resumo, o aplicativo visa modernizar a organização de campeonatos esportivos, simplificando e otimizando o gerenciamento das competições. A proposta busca fomentar o espírito competitivo e fortalecer as comunidades esportivas, ao mesmo tempo em que oferece uma plataforma segura, eficiente e acessível. Com isso, o projeto visa não apenas melhorar a experiência dos usuários, mas também contribuir para a popularização e crescimento do esporte amador e eletrônico.

Palavras-chave: Automatização. Organização. Esportes.

ABSTRACT

The project aims to create a mobile application aimed at organizing and managing amateur sports tournaments, responding to the growing demand for efficient digital solutions in this area. Currently, the management of championships on small and medium scales, such as neighborhood football games or sports competitions between friends, is often carried out in a manual and disorganized way, using spreadsheets, messaging applications and social networks. This can result in difficulties in coordination, errors in tracking games and a lack of clarity in results, affecting the experience of organizers and participants.

The proposed solution is an application that centralizes the entire tournament organization process, automating essential tasks such as registering teams and players, drawing groups and generating leaderboards. The app will offer a simple and intuitive interface, designed to facilitate the creation of competitions and monitoring results in real time. Additionally, it will provide constant notifications and updates, ensuring that everyone involved is always informed about the progress of the championship.

Among the free features, the application will allow you to choose the type of championship (points race or knockout stage), draw matches, and manage categories. In the paid version, more advanced functions will be available, such as detailed player statistics (goals, assists, cards), arbitration management, game calendar and athlete documentation with photos, allowing a greater level of control and professionalism for organizers.

In addition to covering physical sports, the application will also support e-Sports competitions, including electronic games such as FIFA, PES and MOBAs, expanding its reach and encouraging the development of electronic sports. With this, the platform positions itself as a complete solution, which serves both the amateur public and those who organize electronic tournaments, promoting inclusion and sporting diversity.

In short, the application aims to modernize the organization of sports championships, simplifying and optimizing the management of competitions. The proposal seeks to foster a competitive spirit and strengthen sports communities, while offering a safe, efficient and accessible platform. With this, the project aims not only to improve the user experience, but also to contribute to the popularization and growth of amateur and electronic sports.

Keywords: Automation. Organization. Sports.

SUMÁRIO

RESUMO	6
ABSTRACT	8
1 INTRODUÇÃO	14
2 PROBLEMATIZAÇÃO	16
3 OBJETIVO GERAL	19
4 OBJETIVO ESPECÍFICO	22
4.1 Desenvolver uma interface intuitiva e amigável para usuários	22
4.2 Proporcionar flexibilidade e personalização no gerenciamento de torneios	22
4.3 Automatizar sistemas de classificação e gerenciamento de resultados	23
4.4 Oferecer funcionalidades avançadas para usuários premium	23
4.5 Garantir a segurança e integridade das informações.....	24
4.6 Promover a inclusão e diversidade nas competições.....	24
4.7 Facilitar a comunicação entre organizadores, jogadores e espectadores	24
4.8 Incentivar a prática e o espírito competitivo nos esportes eletrônicos	25
4.9 Fortalecer o espírito comunitário através do esporte	25
4.10 Fornecer suporte técnico e evolução contínua do aplicativo.....	26
5 SOLUÇÃO PROPOSTA	27
5.1 Contextualização da Solução	27
5.2 Componentes Centrais da Solução.....	27
5.2.1 Interface Amigável e Intuitiva.....	28
5.2.2 Automatização de Processos Críticos	28
5.2.3 Personalização e Flexibilidade	28
5.2.4 Versão Gratuita e Premium.....	28
5.2.5 Comunicação Integrada	29
5.2.6 Segurança e Confiabilidade	29
5.2.7 Promoção da Inclusão e Diversidade	29
5.2.8 Foco em e-Sports e Competitividade	30
5.2.9 Crescimento Sustentável e Atualizações Contínuas.....	30
5.3 Impacto Esperado.....	30
CONCLUSÃO	79

REFERÊNCIAS	81
APÊNDICE A – TÍTULO	83

1 INTRODUÇÃO

A evolução das Tecnologias da Informação e Comunicação (TIC) ao longo das últimas décadas tem transformado profundamente o modo como as sociedades se organizam e interagem, especialmente no que tange à digitalização de processos tradicionais. Desde os anos 90, com a crescente informatização de setores como o bancário e o de transportes aéreos, passando pela popularização da internet e dos computadores pessoais nos anos 2000, até a explosão do uso de smartphones, a presença da tecnologia se expandiu em todas as esferas da vida cotidiana. Esse avanço tornou o acesso à informação quase onipresente, gerando novas dinâmicas sociais e comerciais.

Com esse cenário em mente, o trabalho proposto visa cobrir um nicho ainda pouco explorado no mercado digital: a sistematização e controle de campeonatos esportivos em pequena e média escala. Especificamente, busca oferecer uma solução prática e acessível para usuários amadores que organizam campeonatos esportivos locais, muitas vezes com comunicação e coordenação limitada a redes sociais e aplicativos de mensagens. Essa lacuna evidencia a necessidade de uma plataforma dedicada, que possa modernizar e otimizar a organização desses torneios, permitindo que os usuários acompanhem todas as etapas de um campeonato de maneira integrada e centralizada.

O desenvolvimento de um aplicativo com essa finalidade enfrenta a questão sobre qual abordagem tecnológica adotar: a escolha entre um desenvolvimento nativo, que aproveita ao máximo as capacidades de cada dispositivo, ou a adoção de um modelo multiplataforma, que, embora mais ágil e com menos exigências específicas, pode comprometer a performance. No entanto, como o aplicativo proposto não demanda grande capacidade de processamento ou uma interface complexa, o desenvolvimento multiplataforma se torna a escolha mais eficiente, simplificando a criação e manutenção do projeto.

O aplicativo se propõe a disponibilizar uma série de funções gratuitas, como a criação e gerenciamento de times, a definição de formatos de campeonatos e a automação de tabelas de classificação e sorteio de grupos. Além disso, oferece uma camada de funcionalidades pagas, como estatísticas detalhadas de jogadores e times,

inscrição de arbitragem e documentação dos atletas, o que traz um maior nível de detalhamento e controle para os organizadores.

A pesquisa de mercado realizada destaca que, embora já existam aplicativos que atendem ao nicho de organização de torneios esportivos, como o Ifut e o Copa Fácil, esses competem entre si com funcionalidades variadas e limitações em sua interface ou atualização de dados. O diferencial do projeto proposto está em sua simplicidade e robustez, apostando em uma estrutura enxuta e eficiente, que minimiza erros e proporciona uma experiência imersiva para seus usuários, tanto em jogos físicos quanto em e-Sports.

O objetivo central da plataforma é, portanto, facilitar a organização de campeonatos amadores, promovendo a inclusão, a diversidade, o espírito competitivo e o fortalecimento das comunidades locais. Ao aliar tecnologia, acessibilidade e usabilidade, o projeto busca incentivar a prática esportiva em suas diversas formas, oferecendo uma solução intuitiva e personalizada que atenda às necessidades dos organizadores e participantes de torneios, ao mesmo tempo em que fomenta o esporte eletrônico, um campo de crescente relevância na era digital.

2 PROBLEMATIZAÇÃO

A organização de torneios esportivos amadores, que tradicionalmente ocorre de forma manual, enfrenta uma série de desafios relacionados à falta de ferramentas digitais adequadas para atender às necessidades desses eventos. Nos campeonatos amadores de futebol de bairro, competições escolares ou mesmo eventos de e-Sports, o gerenciamento costuma depender de métodos improvisados, como planilhas de Excel, grupos em aplicativos de mensagens (WhatsApp, Telegram) e redes sociais para comunicação entre os participantes e organizadores. Embora esses métodos sejam amplamente utilizados por sua simplicidade e acessibilidade, eles têm limitações significativas, resultando em uma experiência desorganizada, demorada e suscetível a erros.

Entre os principais problemas está a dificuldade em manter o controle eficiente sobre a logística dos torneios, que envolve desde a inscrição de jogadores e formação de equipes até o sorteio de partidas, acompanhamento da classificação e comunicação de resultados. Por ser manual, esse processo frequentemente gera inconsistências, como falhas na distribuição de jogos, falta de atualizações precisas das tabelas e confusões nas regras ou definições de etapas, prejudicando o andamento e a experiência do torneio. Além disso, a comunicação entre participantes e organizadores tende a ser dispersa, ocorrendo em diferentes plataformas e canais, o que dificulta o acesso a informações essenciais, como horários de jogos, mudanças de última hora e pontuações atualizadas.

Outro ponto crítico é a ausência de uma estrutura padronizada para acompanhar e documentar o desempenho dos jogadores e equipes. Nos torneios amadores, frequentemente não há ferramentas para registrar estatísticas detalhadas, como gols, assistências, cartões, vitórias e derrotas, o que compromete a transparência e impede que os jogadores e equipes acompanhem seu progresso de forma clara e organizada. Esse tipo de informação, que é comum em torneios profissionais, agrega valor à competição e contribui para a motivação dos participantes, que buscam acompanhar seu desempenho ao longo do torneio.

No caso dos esportes eletrônicos (e-Sports), a situação é ainda mais desafiadora. Embora o mercado de e-Sports tenha crescido significativamente, muitos campeonatos amadores ainda são organizados de maneira rudimentar, sem o suporte de plataformas especializadas. A falta de ferramentas para gerenciar essas

competições eletrônicas resulta em desorganização, atrasos e, muitas vezes, dificuldades técnicas que comprometem a realização dos torneios.

Além dos desafios operacionais, os aplicativos já disponíveis no mercado, que se propõem a resolver esses problemas, apresentam suas próprias limitações. Muitos são focados apenas em esportes específicos, como futebol, ou possuem funcionalidades restritas, sem contemplar as necessidades de modalidades mais variadas, incluindo e-Sports. Isso cria uma lacuna para aqueles que desejam organizar campeonatos em diferentes esportes, mas não encontram ferramentas que atendam de maneira satisfatória essa diversidade. Adicionalmente, problemas técnicos, como bugs e falhas na interface, afetam a experiência dos usuários, além da falta de atualização frequente de informações nos aplicativos.

Por outro lado, o uso de soluções improvisadas e dispersas torna o processo de organização mais suscetível a erros humanos, o que pode prejudicar o andamento de um torneio. É comum, por exemplo, que as tabelas sejam montadas incorretamente ou que haja erros no sorteio de grupos, levando a confusões entre as equipes. A falta de um sistema automatizado para realizar essas tarefas aumenta as chances de problemas que, em uma competição, podem afetar a credibilidade do evento e frustrar os participantes.

Esses desafios também apontam para uma questão de inclusão. A maioria das plataformas existentes concentra-se em competições de médio e grande porte, ou atende a um público mais restrito, como jogadores de futebol ou competidores de e-Sports de nível avançado. Isso deixa de lado os organizadores de campeonatos menores, que buscam uma solução acessível e funcional para gerenciar torneios de escala reduzida, mas que ainda exigem organização, eficiência e transparência. Além disso, a falta de suporte adequado para competições mistas (físicas e eletrônicas) limita o potencial de crescimento de comunidades que desejam integrar diferentes tipos de eventos esportivos.

Diante disso, a problemática central deste projeto é clara: há uma lacuna significativa na oferta de ferramentas digitais eficientes e acessíveis para a organização de torneios esportivos amadores. O atual cenário carece de uma solução que centralize e automatize todos os aspectos da organização, desde a inscrição e gerenciamento de equipes até a geração de tabelas e acompanhamento de estatísticas detalhadas. Além disso, tal solução precisa ser adaptável a diferentes

modalidades esportivas, incluindo tanto competições físicas quanto eletrônicas, e fornecer uma interface intuitiva e amigável, capaz de atender tanto organizadores quanto participantes.

Essa falta de soluções digitais eficazes impacta negativamente a experiência dos organizadores, jogadores e até dos espectadores. Os torneios, que têm grande potencial para fortalecer comunidades e promover a prática esportiva de forma inclusiva e acessível, acabam sendo comprometidos por problemas de gestão e falhas de comunicação. Em um mundo onde a digitalização permeia todos os aspectos da vida cotidiana, a ausência de uma plataforma digital robusta para a gestão de torneios amadores representa uma barreira para o desenvolvimento e a profissionalização desse setor.

Portanto, o projeto propõe uma solução que visa modernizar e simplificar a organização desses eventos, oferecendo uma plataforma completa, acessível e funcional. A proposta é que o aplicativo permita automatizar processos essenciais, reduzir a margem de erro, centralizar a comunicação e proporcionar uma experiência de usuário eficiente e imersiva. Com isso, espera-se não apenas resolver os problemas enfrentados pelos organizadores de campeonatos amadores, mas também fomentar o crescimento das comunidades esportivas e incentivar a prática tanto de esportes físicos quanto eletrônicos.

3 OBJETIVO GERAL

O **objetivo geral** deste projeto é desenvolver uma plataforma digital, estruturada como um aplicativo móvel, que ofereça uma solução robusta, eficiente e automatizada para a organização de torneios esportivos amadores. A plataforma terá como foco a simplificação e otimização de todos os processos envolvidos na criação, gerenciamento e acompanhamento de competições, visando atender tanto esportes físicos tradicionais quanto o crescente segmento dos esportes eletrônicos (e-Sports). A ideia central é modernizar a forma como os torneios amadores são administrados, substituindo métodos manuais e fragmentados por um sistema digital integrado, que atenda às necessidades de organizadores, participantes e espectadores.

O projeto visa resolver problemas comuns enfrentados no gerenciamento de torneios amadores, como a falta de controle eficiente sobre a logística, a comunicação dispersa entre as partes envolvidas, o acompanhamento falho dos resultados e a dificuldade de organizar de maneira clara a distribuição de jogos e equipes. Muitos organizadores recorrem a planilhas, aplicativos de mensagens como WhatsApp, redes sociais e métodos manuais, o que pode resultar em erros de organização, como sorteios de jogos mal realizados, calendários confusos, e falta de transparência nas classificações e estatísticas dos jogadores. Ao automatizar essas tarefas, o aplicativo não só elimina essas falhas, mas também agiliza o processo, tornando-o mais prático e confiável.

A plataforma pretende oferecer uma experiência centralizada que permita, por meio de um único ambiente digital, gerenciar cada aspecto do torneio. Isso inclui a criação de torneios, a inscrição de equipes e jogadores, a geração automática de grupos e tabelas de classificação, o sorteio de partidas, e a atualização em tempo real de resultados e estatísticas. Tudo isso será feito de forma acessível, com uma interface intuitiva, pensada para usuários que muitas vezes não possuem grande familiaridade com ferramentas digitais complexas.

Além de cobrir as necessidades básicas de organização, o projeto também visa agregar valor ao permitir que tanto os organizadores quanto os jogadores e espectadores tenham acesso a uma série de funcionalidades avançadas. Essas funcionalidades incluem o acompanhamento detalhado do desempenho dos jogadores (gols, assistências, cartões, vitórias, etc.), a possibilidade de gestão de

arbitragem e documentação dos atletas (fotos, documentos, etc.), e a criação de calendários de jogos otimizados. Dessa forma, a plataforma se propõe a transformar o gerenciamento de torneios em uma experiência mais profissional e eficiente, que antes só era possível em grandes campeonatos organizados por entidades especializadas.

Uma das principais metas do projeto é também contemplar a diversidade de modalidades esportivas, atendendo desde campeonatos de futebol amador, futsal, basquete e outras modalidades físicas, até competições de e-Sports, que têm ganhado enorme destaque nos últimos anos. A inclusão de esportes eletrônicos amplia o escopo da plataforma, permitindo que ela se torne uma referência para diferentes tipos de competições, adaptando-se às necessidades específicas de cada modalidade. Essa flexibilidade é essencial para atender a um público cada vez mais diverso, que não se limita a esportes tradicionais, mas também explora novas formas de competição e interação digital.

Outro ponto central é a promoção da inclusão e acessibilidade. Ao desenvolver um aplicativo que possa ser utilizado por qualquer pessoa interessada em organizar ou participar de torneios, o projeto visa democratizar o acesso a ferramentas de gestão esportiva. Pequenos organizadores, que anteriormente dependiam de métodos arcaicos ou inacessíveis, terão à disposição uma plataforma que facilita o processo de organização de eventos, seja no âmbito de bairros, escolas, clubes, ou comunidades esportivas de pequeno e médio porte. Isso também estimula o crescimento de novas competições, criando um ambiente mais dinâmico e ativo para o esporte amador.

Em termos de usabilidade, o aplicativo será projetado para oferecer uma experiência amigável, priorizando a simplicidade e a facilidade de navegação. Mesmo os usuários com pouca experiência em aplicativos ou plataformas digitais poderão utilizar a ferramenta sem dificuldades, o que é essencial para garantir que o maior número possível de organizadores e participantes consiga se beneficiar da solução. A interface intuitiva permitirá que os usuários realizem todas as funções, desde o cadastro de equipes até o acompanhamento de estatísticas, sem a necessidade de processos complicados ou demorados.

O projeto também contempla uma estratégia de monetização inteligente, que visa equilibrar o acesso gratuito a funcionalidades básicas com a oferta de

pacotes pagos para usuários que necessitam de funcionalidades mais avançadas. A versão gratuita do aplicativo incluirá funções essenciais para a organização de torneios, como inscrição de equipes, geração de tabelas, sorteio de partidas e comunicação de resultados. Já a versão paga incluirá funcionalidades adicionais, como estatísticas detalhadas, gestão de arbitragem e documentação, além de outras opções que agreguem valor ao evento e proporcionem uma experiência mais completa e personalizada para os organizadores.

A longo prazo, a plataforma busca se consolidar como uma ferramenta de referência na organização de torneios esportivos amadores, promovendo a inclusão, a diversidade e o fortalecimento do espírito competitivo entre os participantes. Ao facilitar o processo de organização e proporcionar uma experiência de usuário moderna e funcional, o aplicativo terá o potencial de impulsionar o crescimento de comunidades esportivas locais e fomentar a prática de esportes de maneira geral. Além disso, ao integrar tanto modalidades físicas quanto eletrônicas, a plataforma se posiciona como um elemento unificador, capaz de conectar diferentes tipos de competições e esportistas em um único ecossistema digital.

Assim, o projeto tem como missão não apenas solucionar os problemas técnicos e operacionais enfrentados pelos organizadores de torneios amadores, mas também criar um ambiente que incentive a colaboração, o espírito esportivo e a inclusão. A plataforma visa impactar positivamente o cenário do esporte amador, oferecendo uma solução que transforma a maneira como os eventos esportivos são organizados, gerando benefícios tanto para organizadores quanto para participantes e espectadores, e contribuindo para o fortalecimento do esporte como um todo.

4 OBJETIVO ESPECÍFICO

Os **objetivos específicos** deste projeto são componentes críticos para garantir que a plataforma proposta atinja seu pleno potencial, oferecendo uma solução completa, funcional e eficiente para a organização de torneios esportivos amadores e e-Sports. A seguir, cada objetivo será expandido com maior nível de detalhamento, abordando suas implicações práticas e o impacto esperado sobre os usuários e o mercado.

4.1 Desenvolver uma interface intuitiva e amigável para usuários

O sucesso de qualquer aplicativo depende fortemente da facilidade de uso, especialmente em uma era onde a simplicidade e a eficiência são demandas fundamentais dos usuários. Um dos principais objetivos deste projeto é criar uma interface que possa ser utilizada por qualquer pessoa, independentemente de sua experiência tecnológica. Isso envolve a construção de uma interface gráfica (UI) que priorize a clareza e a navegabilidade. A arquitetura do aplicativo será desenhada com menus claros e funções facilmente acessíveis. O processo de cadastramento de equipes, jogadores, criação de campeonatos e sorteios será simplificado, minimizando a necessidade de explicações ou tutoriais complexos. A interface será projetada com princípios de design inclusivo, considerando as diversas necessidades de acessibilidade, garantindo que o aplicativo seja utilizável por pessoas com deficiências visuais ou motoras, por exemplo.

4.2 Proporcionar flexibilidade e personalização no gerenciamento de torneios

Um dos maiores diferenciais do aplicativo será a sua capacidade de se adaptar às necessidades específicas de diferentes tipos de torneios e de organizadores. O sistema permitirá que os usuários escolham o formato do torneio, como "pontos corridos" ou "mata-mata", e customizem o número de equipes participantes, categorias de idade, gênero e até o tipo de esporte, incluindo esportes eletrônicos. Essa flexibilidade é crucial, pois as necessidades de um torneio de bairro podem ser muito diferentes das de um campeonato regional ou de e-Sports. A

personalização vai permitir, por exemplo, que uma competição de futebol amador com times mistos e faixas etárias variadas seja gerida com a mesma facilidade que um torneio de e-Sports. A plataforma oferecerá opções para criar campeonatos para diferentes modalidades, incluindo futebol, futsal, vôlei, basquete e até competições virtuais de jogos populares como FIFA e PES. Essa flexibilidade será uma das chaves para a abrangência e adaptabilidade do aplicativo no mercado.

4.3 Automatizar sistemas de classificação e gerenciamento de resultados

A automatização dos processos de gerenciamento de resultados e criação de tabelas de classificação é outro pilar fundamental para a eficiência do aplicativo. Tradicionalmente, a organização de torneios amadores envolve uma quantidade significativa de trabalho manual, que inclui registrar resultados, atualizar tabelas e sortear partidas subsequentes. Esses processos são suscetíveis a erros, atrasos e inconsistências. Com a proposta deste aplicativo, esses procedimentos serão automatizados: os organizadores e árbitros poderão inserir os resultados diretamente no sistema, que automaticamente atualizará a tabela de classificação, calculará a média de pontos e gerará os próximos confrontos. Além disso, o aplicativo gerará estatísticas detalhadas dos jogadores (como gols, assistências, cartões, etc.), o que agregará valor à experiência dos participantes e espectadores, além de facilitar a transparência e o acompanhamento do campeonato.

4.4 Oferecer funcionalidades avançadas para usuários premium

Para viabilizar a sustentabilidade financeira do projeto, será oferecida uma versão premium do aplicativo, que incluirá funcionalidades avançadas. Essas funções adicionais permitirão aos organizadores um nível mais profundo de controle e profissionalismo. Entre essas funções estão a gestão de arbitragem, permitindo o cadastro e acompanhamento de árbitros; calendários detalhados de jogos, que notificam automaticamente jogadores e organizadores sobre os horários e locais das partidas; e a documentação e fotos dos atletas, o que poderá ser usado para comprovar identidades em competições mais formais. Essas funcionalidades avançadas são voltadas para organizadores que precisam de uma plataforma mais

robusta para campeonatos de maior escala e complexidade, proporcionando maior segurança e profissionalismo ao evento.

4.5 Garantir a segurança e integridade das informações

A segurança dos dados é uma questão de extrema importância em qualquer plataforma digital. O aplicativo terá que lidar com informações sensíveis, como dados pessoais dos usuários (jogadores, organizadores e árbitros) e estatísticas dos jogos. Assim, a proteção contra invasões, vazamento de informações e perda de dados será uma prioridade. O sistema contará com criptografia avançada, tanto para a transmissão quanto para o armazenamento de dados. Backup de informações será realizado regularmente para prevenir a perda de dados críticos em caso de falhas técnicas. A autenticação segura será implementada, garantindo que somente usuários autorizados possam acessar áreas sensíveis da plataforma. Essas medidas visam proteger a integridade e privacidade dos usuários, proporcionando uma experiência de uso confiável e segura.

4.6 Promover a inclusão e diversidade nas competições

A plataforma será projetada com um foco especial na promoção da diversidade e inclusão. A organização de torneios frequentemente enfrenta desafios para incluir participantes de diferentes gêneros, idades e habilidades. O aplicativo será flexível o suficiente para permitir a criação de campeonatos inclusivos, com categorias distintas, como campeonatos femininos, mistos, juvenis, master, e competições de e-Sports. Essa flexibilidade ajudará a criar um ambiente esportivo mais equitativo e acolhedor, onde qualquer pessoa, independentemente de suas características, poderá participar. Esse objetivo visa promover não apenas a participação de diferentes grupos, mas também contribuir para a formação de novas comunidades esportivas, fortalecendo a diversidade e a integração social por meio do esporte.

4.7 Facilitar a comunicação entre organizadores, jogadores e espectadores

A comunicação eficiente entre os diferentes participantes de um torneio é um fator essencial para o sucesso da organização. Este objetivo propõe a criação de

canais de comunicação integrados no aplicativo, permitindo que organizadores enviem notificações automáticas e mensagens diretas para as equipes e jogadores. Essas mensagens poderão alertar os jogadores sobre alterações nos horários de partidas, mudanças de regras, e outros detalhes importantes. Além disso, o aplicativo fornecerá aos espectadores atualizações em tempo real sobre os resultados dos jogos e a classificação do torneio. Com isso, todos os envolvidos estarão sempre informados, minimizando confusões e garantindo a eficiência e transparência do evento.

4.8 Incentivar a prática e o espírito competitivo nos esportes eletrônicos

O crescimento explosivo dos esportes eletrônicos (e-Sports) em escala global não pode ser ignorado. Este projeto visa não apenas focar em esportes tradicionais, mas também apoiar e incentivar a prática competitiva nos e-Sports. A plataforma será adaptada para permitir a organização de campeonatos de jogos eletrônicos, oferecendo as mesmas funcionalidades avançadas disponíveis para os esportes físicos, como gestão de times, criação de tabelas, sorteios automáticos e acompanhamento de estatísticas dos jogadores. Esse apoio ao e-Sports é uma estratégia para atingir um público mais jovem e tecnologicamente engajado, ampliando o alcance do aplicativo e fortalecendo sua presença no crescente mercado de competições digitais.

4.9 Fortalecer o espírito comunitário através do esporte

Um dos impactos sociais mais significativos deste projeto é o fortalecimento das comunidades locais por meio do esporte. A plataforma permitirá que pessoas organizem e participem de torneios amadores em seus bairros, cidades ou regiões, criando eventos que envolvem e integram a comunidade. Ao fornecer uma solução digital acessível e fácil de usar, o aplicativo estimulará a interação social, o trabalho em equipe e o espírito de competição saudável. Além disso, ele permitirá que comunidades mais afastadas ou menos atendidas tenham a oportunidade de organizar e participar de competições, promovendo uma maior inclusão social e contribuindo para a construção de um ambiente esportivo mais unido e colaborativo.

4.10 Fornecer suporte técnico e evolução contínua do aplicativo

O desenvolvimento de um aplicativo eficiente não termina com o lançamento inicial; é fundamental que ele seja continuamente atualizado e aprimorado para atender às novas necessidades dos usuários e às mudanças tecnológicas. O objetivo de fornecer suporte técnico contínuo garantirá que os usuários tenham assistência imediata caso encontrem problemas técnicos ou precisem de ajuda com alguma funcionalidade. Além disso, o feedback dos usuários será monitorado para identificar áreas de melhoria, garantindo que o aplicativo continue evoluindo ao longo do tempo, mantendo-se relevante e competitivo no mercado.

Esses objetivos específicos, detalhados e expandidos, proporcionam uma visão abrangente do que o projeto pretende alcançar, oferecendo uma solução moderna, eficaz e inclusiva para a organização de torneios esportivos e e-Sports. Através da combinação de tecnologia, flexibilidade e foco no usuário, o aplicativo não apenas simplificará a gestão de campeonatos amadores, mas também contribuirá para o crescimento do esporte e da competitividade em diversas comunidades.

5 SOLUÇÃO PROPOSTA

A **solução proposta** visa transformar a forma como torneios esportivos, tanto físicos quanto eletrônicos, são organizados e gerenciados, proporcionando uma plataforma digital intuitiva e eficaz. O desenvolvimento de um aplicativo móvel multiplataforma, destinado a pequenos e médios eventos, busca modernizar o processo, facilitando a vida de organizadores e participantes por meio de ferramentas automatizadas, personalizáveis e acessíveis. A proposta responde a lacunas evidentes na gestão manual de competições, como erros humanos, dificuldades logísticas, falta de centralização das informações e comunicação ineficiente.

5.1 Contextualização da Solução

O mundo contemporâneo testemunhou uma revolução no uso da tecnologia, especialmente na organização de eventos esportivos, com muitas atividades migrando para plataformas digitais em busca de maior eficiência e acessibilidade. No entanto, a gestão de torneios amadores, sejam esportes físicos ou e-Sports, ainda enfrenta desafios significativos devido à falta de soluções robustas e intuitivas, que se adequem às necessidades específicas de pequenos e médios organizadores.

Ao identificar essa demanda, a proposta de criar um aplicativo dedicado à **gestão automatizada de torneios** apresenta-se como uma solução oportuna. Atualmente, muitos campeonatos ainda dependem de métodos tradicionais e manuais de organização, como planilhas, papel e grupos de WhatsApp, o que acarreta em desorganização e confusão. A automação desses processos simplifica a logística, melhora a comunicação entre os envolvidos e elimina grande parte dos erros e atrasos comuns nesse contexto.

5.2 Componentes Centrais da Solução

5.2.1 Interface Amigável e Intuitiva

A experiência do usuário é um dos pilares centrais do aplicativo. O design será pensado para garantir que tanto organizadores quanto participantes possam utilizar o aplicativo com facilidade, independentemente de seu nível de familiaridade com tecnologias digitais. A interface intuitiva permitirá que usuários realizem tarefas complexas, como organizar tabelas de classificação ou configurar categorias de torneio, de maneira simples e rápida.

5.2.2 Automatização de Processos Críticos

O coração do aplicativo é a sua capacidade de automatizar todo o ciclo de vida de um torneio. Desde o cadastro de equipes e jogadores até o sorteio automático das rodadas, definição de tabelas de classificação, gerenciamento de calendário e envio de notificações, o aplicativo garante que todas as etapas sejam executadas com precisão. Isso reduz significativamente o trabalho manual dos organizadores e minimiza erros comuns, como o cálculo incorreto de pontuações ou a falta de comunicação sobre horários de jogos.

5.2.3 Personalização e Flexibilidade

Um dos diferenciais da solução é a flexibilidade oferecida aos organizadores. O aplicativo permitirá a criação de torneios para diversos esportes, como futebol, futsal, basquete, vôlei, além de incluir suporte para esportes eletrônicos (e-Sports). Dentro de cada categoria esportiva, será possível configurar formatos variados, como campeonatos de pontos corridos ou mata-mata, além de definir categorias de idade e gênero. A flexibilidade de personalização garante que o aplicativo atenda tanto torneios recreativos de bairro quanto competições semiprofissionais de maior escala.

5.2.4 Versão Gratuita e Premium

Para garantir acessibilidade a diferentes tipos de usuários, o aplicativo oferecerá uma versão gratuita que atenderá as principais demandas de organização

de torneios, como criação de campeonatos, sorteio de grupos e rodadas, e tabelas de classificação. No entanto, para usuários que buscam funcionalidades mais avançadas, haverá uma versão premium, que incluirá estatísticas detalhadas dos jogadores, gestão de arbitragem, controle de documentação e imagens dos atletas, além de calendários automáticos e completos. Isso criará um modelo de monetização sustentável, permitindo ao aplicativo crescer e oferecer mais recursos ao longo do tempo.

5.2.5 Comunicação Integrada

A comunicação é um dos principais desafios na gestão de campeonatos, especialmente amadores, onde muitas vezes os envolvidos dependem de grupos em redes sociais para compartilhar informações. A solução proposta inclui notificações automáticas sobre resultados, atualizações de tabela, mudanças de horário e outras informações essenciais. Além disso, a plataforma permitirá o envio de mensagens e o compartilhamento de documentos importantes, centralizando a comunicação dentro do aplicativo e eliminando a necessidade de usar várias ferramentas separadas.

5.2.6 Segurança e Confiabilidade

Um aspecto essencial do aplicativo é garantir a segurança dos dados dos usuários e a confiabilidade do sistema. A proposta inclui a implementação de criptografia para proteger informações pessoais e garantir que dados, como o histórico de torneios e resultados, sejam preservados. Além disso, o aplicativo contará com backups regulares e automáticos para evitar a perda de dados em caso de falhas.

5.2.7 Promoção da Inclusão e Diversidade

Um dos diferenciais da solução é seu compromisso com a inclusão de diferentes públicos. O aplicativo permitirá a criação de torneios para todas as categorias de gênero e idade, além de possibilitar competições mistas. Esse foco

inclusivo é essencial para promover a democratização do esporte, permitindo que torneios envolvam um público diverso, fomentando a participação de grupos que, tradicionalmente, poderiam estar à margem de tais competições.

5.2.8 Foco em e-Sports e Competitividade

Reconhecendo o crescimento dos esportes eletrônicos em escala global, a plataforma também estará preparada para organizar torneios de e-Sports, como competições de jogos populares. Isso amplia o escopo do aplicativo e atende a um público cada vez mais relevante. A solução busca fomentar o espírito competitivo, oferecendo suporte tanto para esportes físicos quanto para competições online, contribuindo para o fortalecimento das comunidades esportivas em diferentes frentes.

5.2.9 Crescimento Sustentável e Atualizações Contínuas

Para garantir que o aplicativo acompanhe as demandas crescentes dos usuários e as tendências do mercado, haverá um compromisso com o desenvolvimento contínuo da plataforma. Atualizações regulares serão implementadas com base no feedback dos usuários, e novas funcionalidades poderão ser integradas de acordo com as necessidades emergentes.

5.3 Impacto Esperado

Com a implementação desta solução, espera-se que a organização de torneios esportivos amadores e e-Sports seja significativamente facilitada e modernizada. O aplicativo digital eliminará a dependência de métodos manuais, como planilhas e papéis, permitindo que qualquer pessoa organize competições de maneira eficiente e profissional. Além disso, ao oferecer uma plataforma acessível e inclusiva, a solução contribuirá para a democratização do esporte, fortalecendo laços comunitários e promovendo a participação de grupos diversos em eventos esportivos.

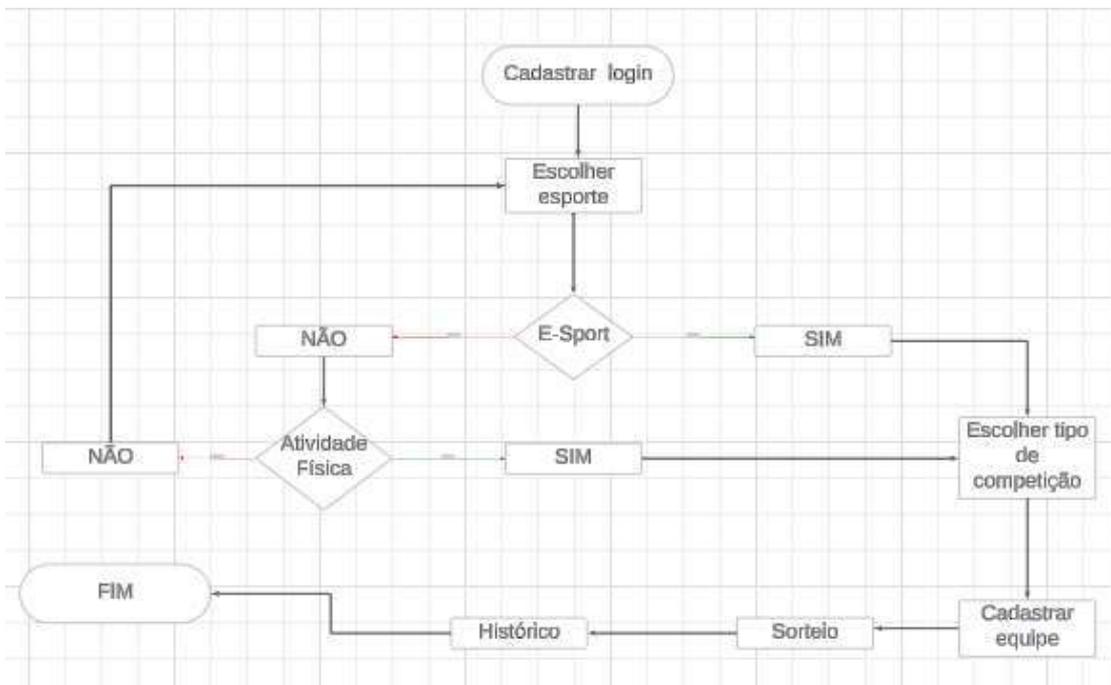
Ao tornar o gerenciamento de torneios mais acessível, prático e automatizado, a proposta tem o potencial de transformar o cenário esportivo amador, criando um ambiente mais competitivo e organizado. Além disso, a integração de e-Sports amplia ainda mais o alcance da solução, respondendo às demandas de um público jovem e

conectado, contribuindo assim para o desenvolvimento de competências em ambos os universos – físico e digital.



EAP

Fluxograma



Banco de dados

Trello



Protótipo

Página de login

Usuário/e-mail

Senha:

ENTRAR

Criar nova conta

Esqueci minha senha

Entrar com Google



Perfil

Nome:

Idade:

Usuário:

Telefone:

Adicione sua foto

SALVAR

Página Botão Torneio

Nome do torneio:

Selecione um formato

Ligo (Pontos corridos)

Eliminatória (Mata-mata)

Máximo de times: 4 - 8 - 16 - 20

Finalizar

Página Botão Times

Meu Time: Gaviões

Adicionar Jogador: João Vieira

Adicione o Escudo:

Finalizar

Menu Hamburguer

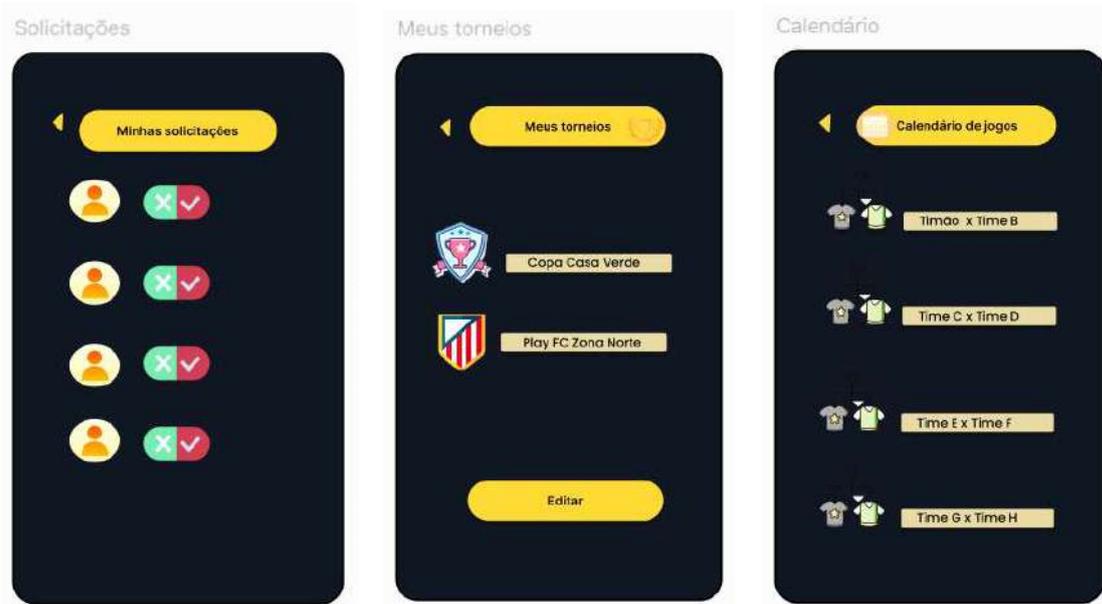
MENU

Meu perfil

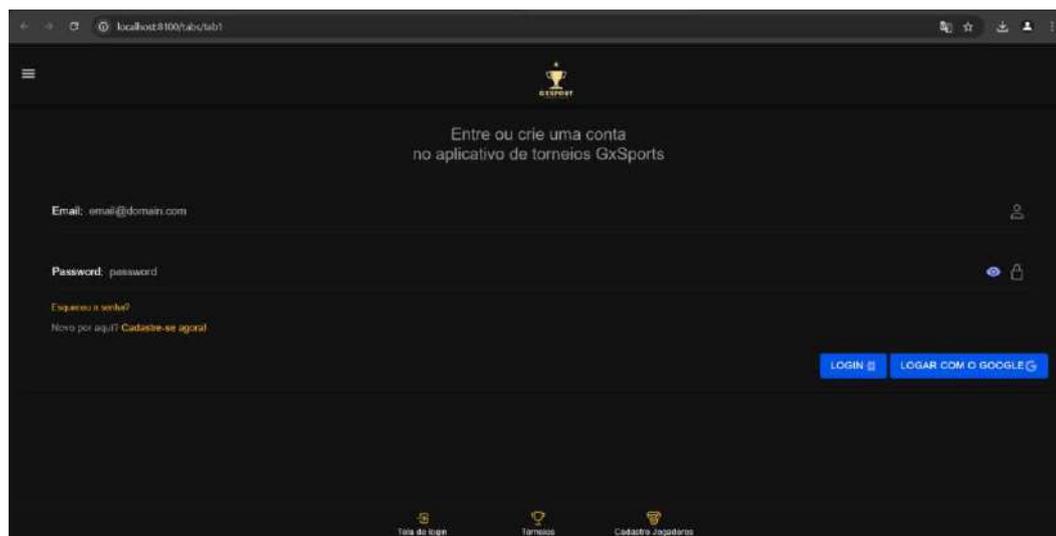
Minhas solicitações

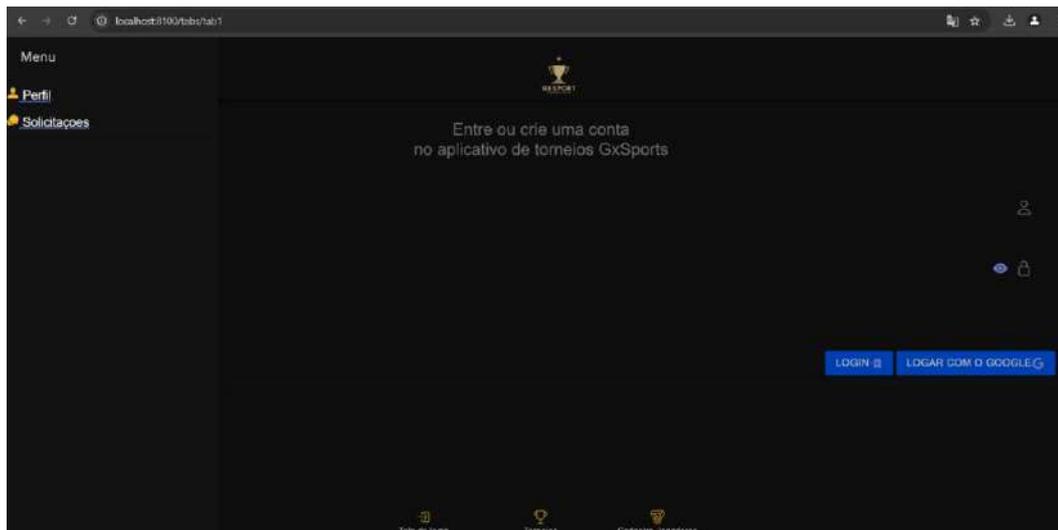
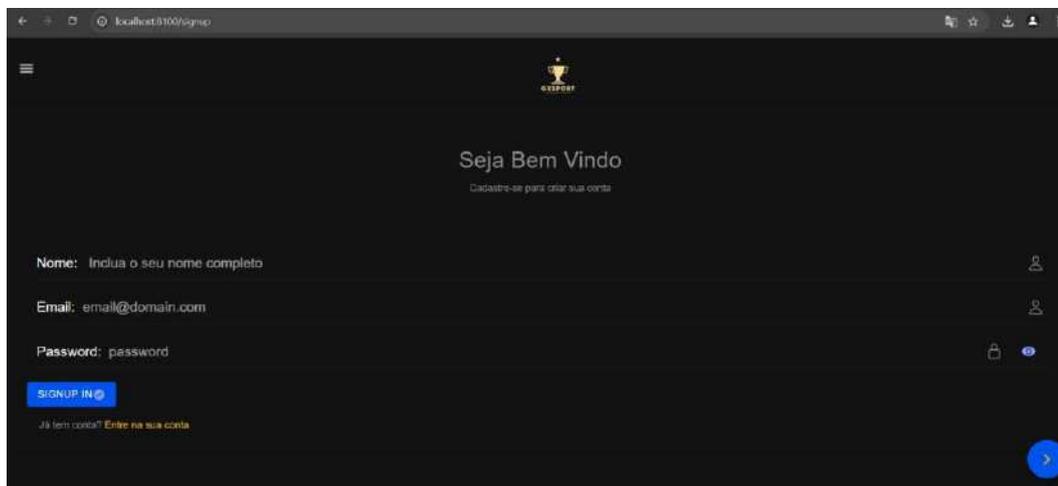
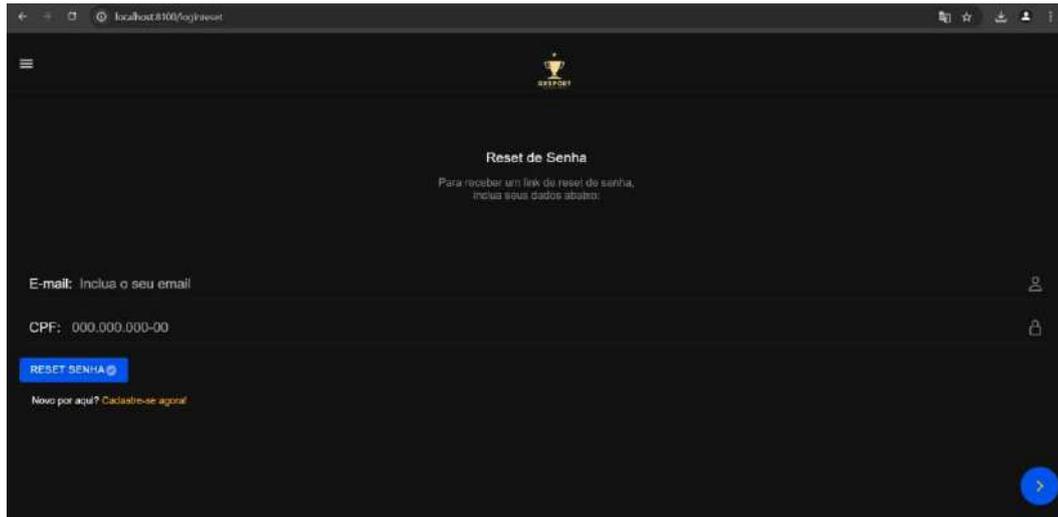
Meus torneios

Sair



Projeto (Telas)





localhost:8100/ver-6

Perfil



Nome
Digite seu nome

E-mail
Digite seu e-mail

Apelido
Digite seu apelido

Número da Camiseta
Digite o número da camiseta

Posição Selecione sua posição -

Dia
Dia

Mês
Mês

Ano
Ano

Idade

localhost:8100/solicitacoes

Minhas solicitações

	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<input checked="" type="checkbox"/>	<input type="checkbox"/>

localhost:8100/tabs/tab2



Meus torneios

CALENDÁRIO

Campeonato Municipal de Futebol Amador 2024

CALENDÁRIO

Torneio Amistoso de Verão 2024

CALENDÁRIO

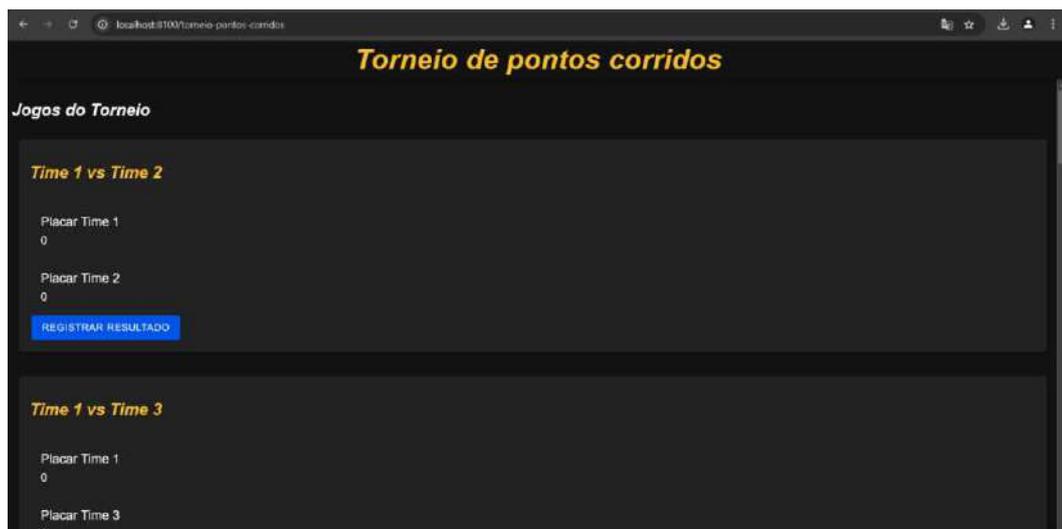
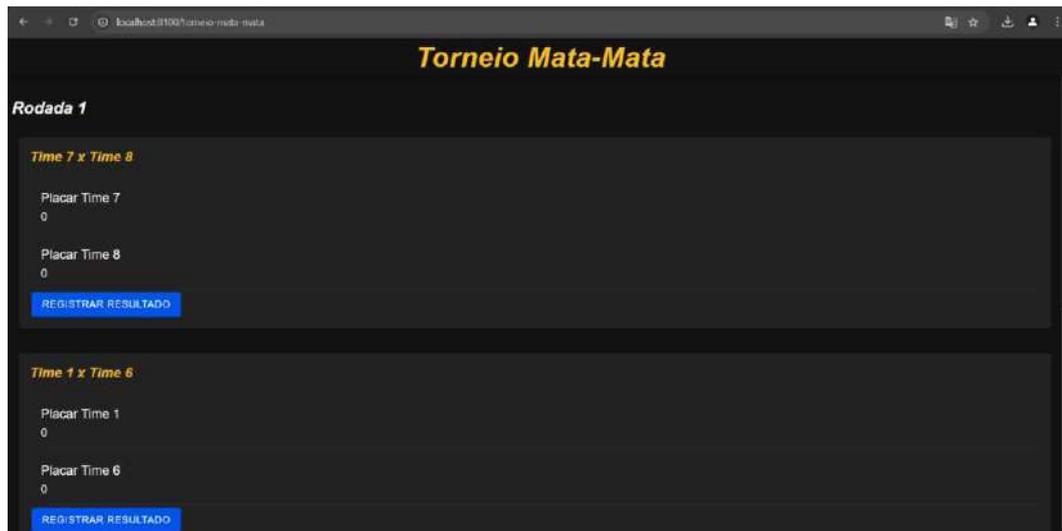
CALENDÁRIO

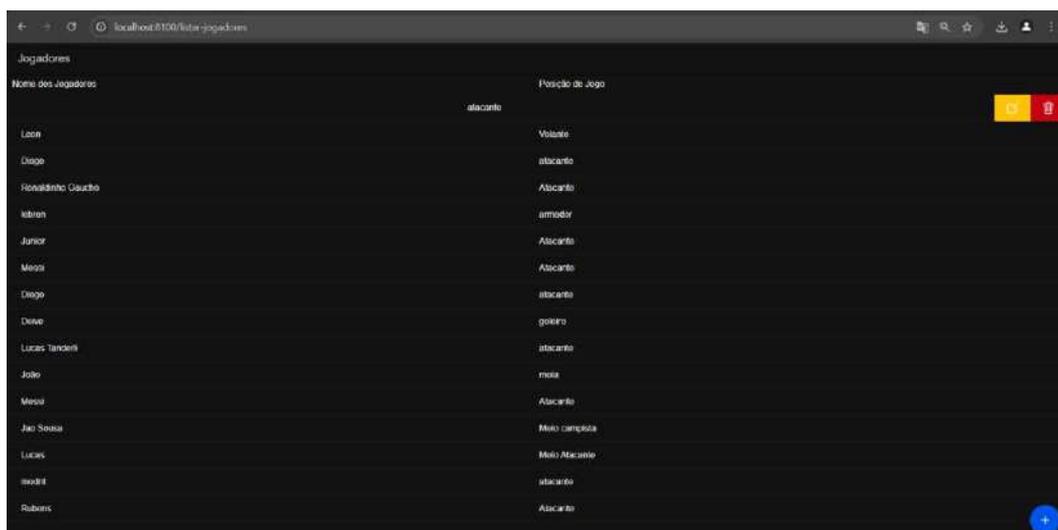
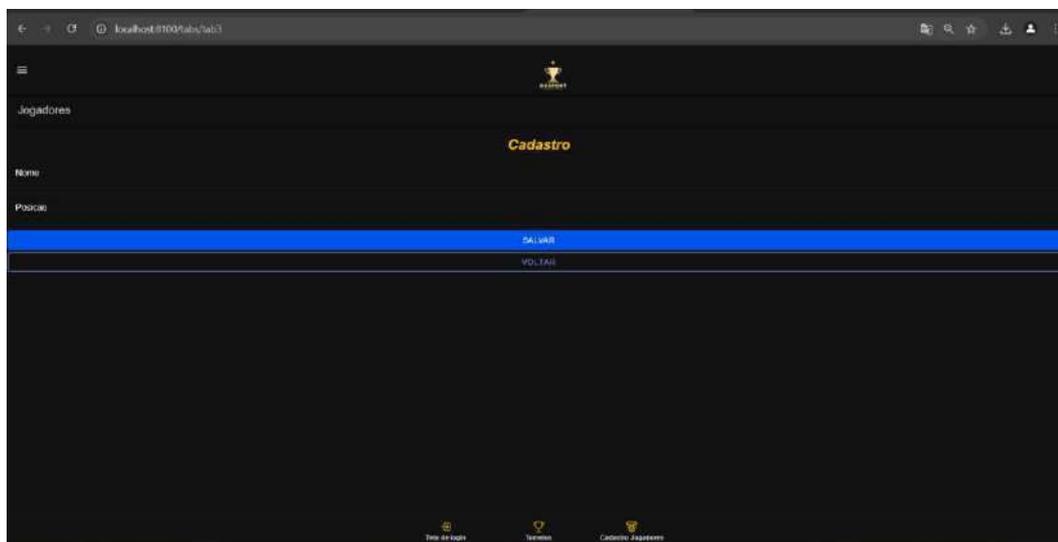
Copa de Futebol Interbairros 2024

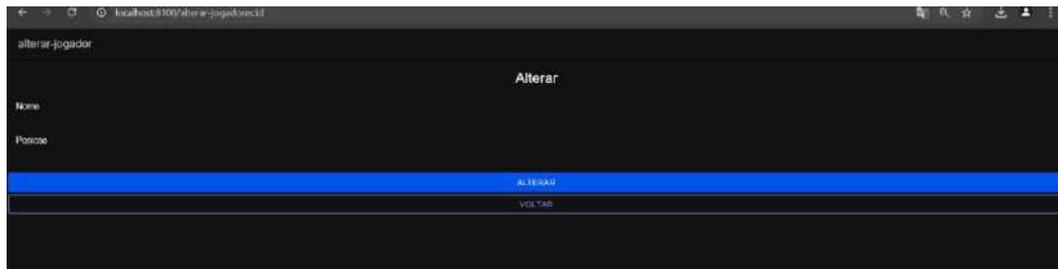
TOURNEIO MATA-MATA

TOURNEIO PONTOS CONSIDOS

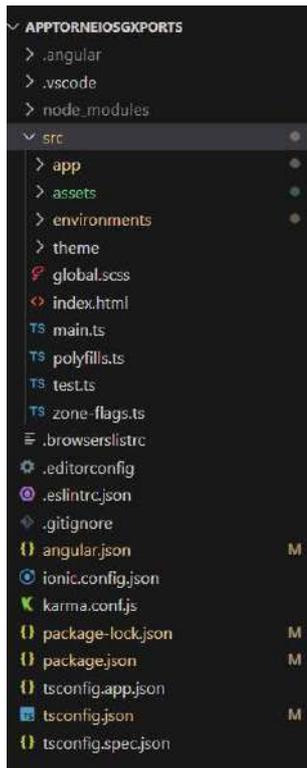
Tela de Login Torneios Cadastro Jogadores







Projeto Código:



App.Module. TS- Importações dos módulos para o funcionamento do projeto com a finalidade de integração com o banco de dados / páginas/ serviços

```

1 import { NgModule } from '@angular/core';
2 import { BrowserModule } from '@angular/platform-browser';
3 import { RouteReuseStrategy } from '@angular/router';
4
5 import { IonicModule, IonicRouteStrategy } from '@ionic/angular';
6
7 import { AppRoutingModule } from './app-routing.module';
8 import { AppComponent } from './app.component';
9
10 //importações dos módulos do Angular Fire e environments.ts
11
12 import { AngularFireModule } from '@angular/fire/compat';
13 import { AngularFireStoreModule } from '@angular/fire/compat/firestore';
14 import { environment } from 'src/environments/environment';
15
16
17 //importações dos módulos de autenticação
18 import { AngularFireAuth } from '@angular/fire/compat/auth';
19 import { AngularFireAuthModule } from '@angular/fire/compat/auth';
20 //import do serviço criado
21 import { AutenticacaoService } from './Services/autenticacao.service';
22
23
24 @NgModule({
25   declarations: [AppComponent],
26   imports: [BrowserModule, IonicModule.forRoot(), AppRoutingModule,
27     AngularFireModule.initializeApp(environment.firebaseconfig),
28     AngularFireAuthModule,AngularFirestoreModule],
29   providers: [{ provide: RouteReuseStrategy, useClass: IonicRouteStrategy }, AutenticacaoService],
30   bootstrap: [AppComponent],
31 })
32 export class AppModule {}
33

```

App-routing.module.ts- criação as rotas

```

1 import { NgModule } from '@angular/core';
2 import { PreloadAllModules, RouterModule, Routes } from '@angular/router';
3
4 const routes: Routes = [
5   {
6     path: '',
7     loadChildren: () => import('./tabs/tabs.module').then( m => m.TabsPageModule)
8   },
9   {
10    path: '',
11    redirectTo: 'login.page',
12    pathMatch: 'full'
13  },
14  {
15    path: 'home',
16    loadChildren: () => import('./home/home.module').then( m => m.HomePageModule)
17  },
18  {
19    path: 'login.page',
20    loadChildren: () => import('./login/login.module').then( m => m.LoginPageModule)
21  },
22  {
23    path: 'cadastro-jogadores',
24    loadChildren: () => import('./pages/cadastro-jogadores/cadastro-jogadores.module').then( m => m.CadastroJogadoresPageModule)
25  },
26  {
27    path: 'alterar-jogadores:id',
28    loadChildren: () => import('./pages/alterar-jogadores/alterar-jogadores.module').then( m => m.AlterarJogadoresPageModule)
29  },
30  {
31    path: 'listar-jogadores',
32    loadChildren: () => import('./pages/listar-jogadores/listar-jogadores.module').then( m => m.ListarJogadoresPageModule)
33  },
34 ]
35

```

```

EXPLORER
  APP/TORNEIOS/EXPORTS
  src > app > app-routing.module.ts > 90 routes
  4 const routes: Routes = [
  5
  6   path: 'reset',
  7   loadChildren: () => import('./reset/reset.module').then( m => m.ResetPageModule)
  8 },
  9
 10   path: 'loginreset',
 11   loadChildren: () => import('./loginreset/loginreset.module').then( m => m.LoginresetPageModule)
 12 },
 13
 14   path: 'signup',
 15   loadChildren: () => import('./signup/signup.module').then( m => m.SignupPageModule)
 16 },
 17
 18   path: 'perfil',
 19   loadChildren: () => import('./perfil/perfil.module').then( m => m.PerfilPageModule)
 20 },
 21
 22   path: 'torneo-pontos-corridos',
 23   loadChildren: () => import('./torneo-pontos-corridos/torneo-pontos-corridos.module').then( m => m.TorneioPontosCorridosPageModule)
 24 },
 25
 26   path: 'torneo-mata-mata',
 27   loadChildren: () => import('./torneo-mata-mata/torneo-mata-mata.module').then( m => m.TorneioMataMataPageModule)
 28 },
 29
 30   path: 'testetorneio',
 31   loadChildren: () => import('./testetorneio/testetorneio.module').then( m => m.TestetorneioPageModule)
 32 },
 33
 34   path: 'solicitacoes',
 35   loadChildren: () => import('./solicitacoes/solicitacoes.module').then( m => m.SolicitacoesPageModule)
 36 },
 37 ];
 38
 39 @NgModule({
 40   imports: [
 41     RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
 42   ]
 43 })
 44 export class AppRoutingModule {}
 45
 46
 47
 48
 49
 50
 51
 52
 53
 54
 55
 56
 57
 58
 59
 60
 61
 62
 63
 64
 65
 66
 67
 68
 69
 70
 71
 72
 73
 74
 75
 76
 77
 78
 79
 80
 81
 82
 83
 84
 85
 86
 87
 88
 89
 90
 91
 92
 93
 94
 95
 96
 97
 98
 99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Login page:

HTML

```

EXPLORE
APPTORNEIOSGXPOR
  > .vscode
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  TS login-routing.module.ts
  TS login.module.ts
  login.page.html
  login.page.scss
  TS login.page.spec.ts
  TS login.page.ts
  > loginreset
  > models
  > pages
  > perfil
  > reset
  > Services
  > signup
  > solicitacoes
  > tab1
  > tab2
  > tab3
  > tabs
  > testetorneio
  > torneio-mata-mata
  > torneio-pontos-corridos
  TS app-routing.module.ts
  app.component.html
  > OUTLINE
src > app > login > login.page.html > ion-content > ion-card
  1
  2
  3 <ion-content [fullscreen]="true">
  4
  5
  6 <ion-card class="card">
  7   <h2 class="GxSports">Entre ou crie uma conta <small> (seu aplicativo de torneos GxSports)</small>
  8
  9   <ion-list class="ion-padding">
 10
 11     <ion-item class="item">
 12       <ion-input placeholder=" email@domain.com " label=" Email: " label-placement="text" ion-focus="true" type="email" required=
 13     </ion-item>
 14     <ion-item>
 15       <ion-input placeholder=" password " label=" Password: " label-placement="password" type="password" required="true" class=
 16     </ion-item>
 17
 18     <div class="Esquec(Senha)">
 19       <a [routerLink]="['/loginreset']" style="color: #eeba2b">Esqueceu a senha?</a>
 20     </div>
 21
 22     <div class="signup">
 23       <p>Como por aqui? <a style="color: #eeba2b" [routerLink]="['/signup']">Cadastre-se agora!</a></p>
 24     </div>
 25
 26     <div class="btns">
 27
 28       <ion-button slot="end" size="default" (click)="login()"> login <ion-icon color="primary" name="log-in-sharp"></ion-icon></ion-
 29       <ion-button slot="end" size="default" (click)="loginwithgoogle()"> Logar com o Google <ion-icon color="primary" name="logo-goog
 30     </div>
 31
 32   </ion-list>
 33
 34 </ion-card>
 35
 36 </ion-content>

```

```

src > app > login > login.page.html > ion-content
  3 <ion-content [fullscreen]="true">
  37
  38
  39
  40   <div>
  41     <ion-fab vertical="bottom" horizontal="end" class="BtnPadrao">
  42       <ion-fab-button [routerLink]="['']">
  43         <ion-icon name="chevron-forward" style="color: #eeba2b"></ion-icon>
  44       </ion-fab-button>
  45     </ion-fab>
  46   </div>
  </ion-content>

```

CSS

```

APPTORNEIOSGEXPORTS
  > .vscode
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  > login-routing.module.ts
  > login.module.ts
  > login.page.html
  > login.page.scss
  > login.page.spec.ts
  > login.page.ts
  > loginreset
  > models
  > pags
  > perfil
  > reset
  > Services
  > signup
  > solicitacoes
  > tab1
  > tab2
  > tab3
  > tabs
  > testetorneio
  > torneio-mata-mata
  > torneio-portos-condos
  > app-routing.module.ts
  > app.component.html
  > OUTLINE
  > TIMELINE

src > app > login > login.page.scss > ...
1  .loginheader{
2    height: 100px;
3    margin-top: 5px;
4    margin-bottom: 5px;
5  }
6
7  .ton]barheader{
8    height: 90px;
9  }
10
11  .apresentacao{
12    height: fit-content;
13    width: fit-content;
14  }
15
16  .signup{
17    display: flex;
18    justify-content: flex-start;
19    margin-left: 35px;
20  }
21
22    p{
23      font-size: 0.9rem;
24    }
25    a{
26      text-decoration: none;
27    }
28
29  .EsqueciSenha{
30    display: flex;
31    justify-content: flex-start;
32    margin-left: 35px;
33    a{
34      font-weight: 400;
35      font-size: 0.8rem;
36    }
37  }

```

```

APPTORNEIOSGEXPORTS
  > .vscode
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  > login-routing.module.ts
  > login.module.ts
  > login.page.html
  > login.page.scss
  > login.page.spec.ts
  > login.page.ts
  > loginreset
  > models
  > pags
  > perfil
  > reset
  > Services
  > signup
  > solicitacoes
  > tab1
  > tab2
  > tab3
  > tabs
  > testetorneio
  > torneio-mata-mata

src > app > login > login.page.scss > ...
31  .EsqueciSenha{
32    display: flex;
33    justify-content: flex-start;
34    margin-left: 35px;
35    a{
36      font-weight: 400;
37      font-size: 0.8rem;
38      text-decoration: none;
39    }
40  }
41
42  .GxSports{
43    text-align: center;
44  }
45
46  .item{
47    padding: 20px;
48  }
49
50  .btns{
51    display: flex;
52    justify-content: end;
53    margin-top: 10px;
54  }
55
56  .card{
57    border-radius: 6px;
58  }

```

Typescript

```

src > app > login > TS login.page.ts > ...
1 import { Router } from '@angular/router';
2 import { Component, OnInit } from '@angular/core';
3 import { AutenticacaoService } from '../services/autenticacao.service';
4 import { AlertController, ToastController } from '@ionic/angular';
5 import { waitForAsync } from '@angular/core/testing';
6
7 @Component({
8   selector: 'app-login',
9   templateUrl: './login.page.html',
10  styleUrls: ['./login.page.scss'],
11 })
12 export class LoginPage implements OnInit {
13
14   email = '';
15   password = '';
16
17   constructor(public autenticacaoService: AutenticacaoService,
18     public router: Router, public alertController:AlertController, public toastController:ToastController) {
19     this.autenticacaoService.logged().subscribe(
20       data => {
21         if (data != null) this.router.navigate(['login']);
22       },
23       err => console.log(err)
24     )
25   }
26
27   ngOnInit() {
28
29   }
30
31   async login() {
32     try{
33       await this.autenticacaoService.login(this.email, this.password);
34       this.exibirAlertasEmailAutenticado();
35     } catch (err) {
36       console.log(err);
37     }
38   }
39 }

```

```

src > app > login > TS login.page.ts > ...
12 export class LoginPage implements OnInit {
32   async login() {
37
38   } catch (error) {
39
40     console.error('Erro de login: ', error);
41     this.exibirAlertasEmailNaoAutenticado();
42   }
43   } finally {
44     this.router.navigate(['perfil']);
45   }
46   }
47   }
48   }
49   }
50   }
51   }
52   }
53   }
54   }
55   }
56   }
57   }
58   }
59   }
60   }
61   }
62   }
63   }
64   }
65   }
66   }
67   }
68   }
69   }
70   }
71   }

```

```

APPTORNEIOSXPORTS
  > .vscode
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  TS login-routing.module.ts U 77
  TS login.module.ts U 78
  login.page.html U 80
  login.page.scss U 81
  TS login.page.spec.ts U 82
  TS login.page.ts U 83
  > loginreset 84
  > models 85
  > pags 86
  > perfil 88
  > reset 89
  > Services 90
  > signup 91
  > solicitacoes 92
  > tab1 93
  > tab2 94
  > tab3 95
  > tabs 96
  > testetorneio 97
  > torneio-mata-mata 98
  > torneio-pontos-corridos 99
  TS app-routing.module.ts M 100
  app.component.html 101
  > OUTLINE 102
  > TIMELINE 103
  104
  105

src > app > login > TS login.page.ts > ...
12 export class LoginPage implements OnInit {
70 }
71
72 // Funcoes de alertas tratamento
73
74 async exibirAlertasemailAutenticado(){
75
76   const toast=await this.toastController.create
77   (
78     message: 'Login realizado com sucesso',
79     duration:500
80   );
81   toast.present();
82 }
83
84
85 async exibirAlertasemailNaoAutenticado(){
86
87   const toast=await this.toastController.create
88   (
89     message: 'Nao autenticado',
90     duration:500
91   );
92   toast.present();
93 }
94
95
96
97
98 async exibirAlertasemailErroCadastro(){
99   const toast=await this.toastController.create
100  (
101    message: 'Erro ao se cadastrar',
102    duration:3000
103  );
104  toast.present();
105 }

```

Loginreset

```

APPTORNEIOSXPORTS
  > .vscode
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  > loginreset
  TS loginreset-routing.module.ts U 11
  TS loginreset.module.ts U 12
  loginreset.page.html U 13
  loginreset.page.scss U 14
  TS loginreset.page.spec.ts U 15
  TS loginreset.page.ts U 16
  > models 17
  > pags 18
  > perfil 19
  > reset 20
  > Services 21
  > signup 22
  > solicitacoes 23
  > tab1 24
  > tab2 25
  > tab3 26
  > tabs 27
  > testetorneio 28
  > torneio-mata-mata 29
  > torneio-pontos-corridos 30
  TS app-routing.module.ts M 31
  app.component.html 32
  > OUTLINE 33
  > TIMELINE 34
  35
  36
  37

src > app > loginreset > loginreset.page.html > @id#main-content:ion-page > @ion-header
1 <ion-menu contentId="main-content">
2   <ion-header>
3     <ion-toolbar>
4       <ion-title>Menu</ion-title>
5     </ion-toolbar>
6   </ion-header>
7 </ion-menu>
8 <div class="ion-page" id="main-content">
9   <ion-header>
10    <ion-toolbar>
11      <ion-buttons slot="start">
12        <ion-menu-button</ion-menu-button>
13      </ion-buttons>
14      <ion-toolbar class="toolbarheader">
15        <ion-img src="assets/img/logoporti.png" class="logoheader"></ion-img>
16      </ion-toolbar>
17    </ion-header>
18    <ion-content [fullscreen]="true" class="content">
19      <div class="header">
20        <ion-title class="title">Reset de Senha</ion-title>
21        <p class="subtitle">
22          <span>Para receber um link de reset de senha,</span>
23          <span>inclua seus dados abaixo:</span>
24        </p>
25        <ion-list class="ion-padding">
26          <ion-item class="item">
27            <ion-input placeholder="Inclua o seu email" label="E-mail" type="email" clear-input="true"
28            </ion-item>
29          <ion-item class="item">
30            <ion-input placeholder="000.000.000-00" label="CPF" type="number" clear-input="true" cl

```

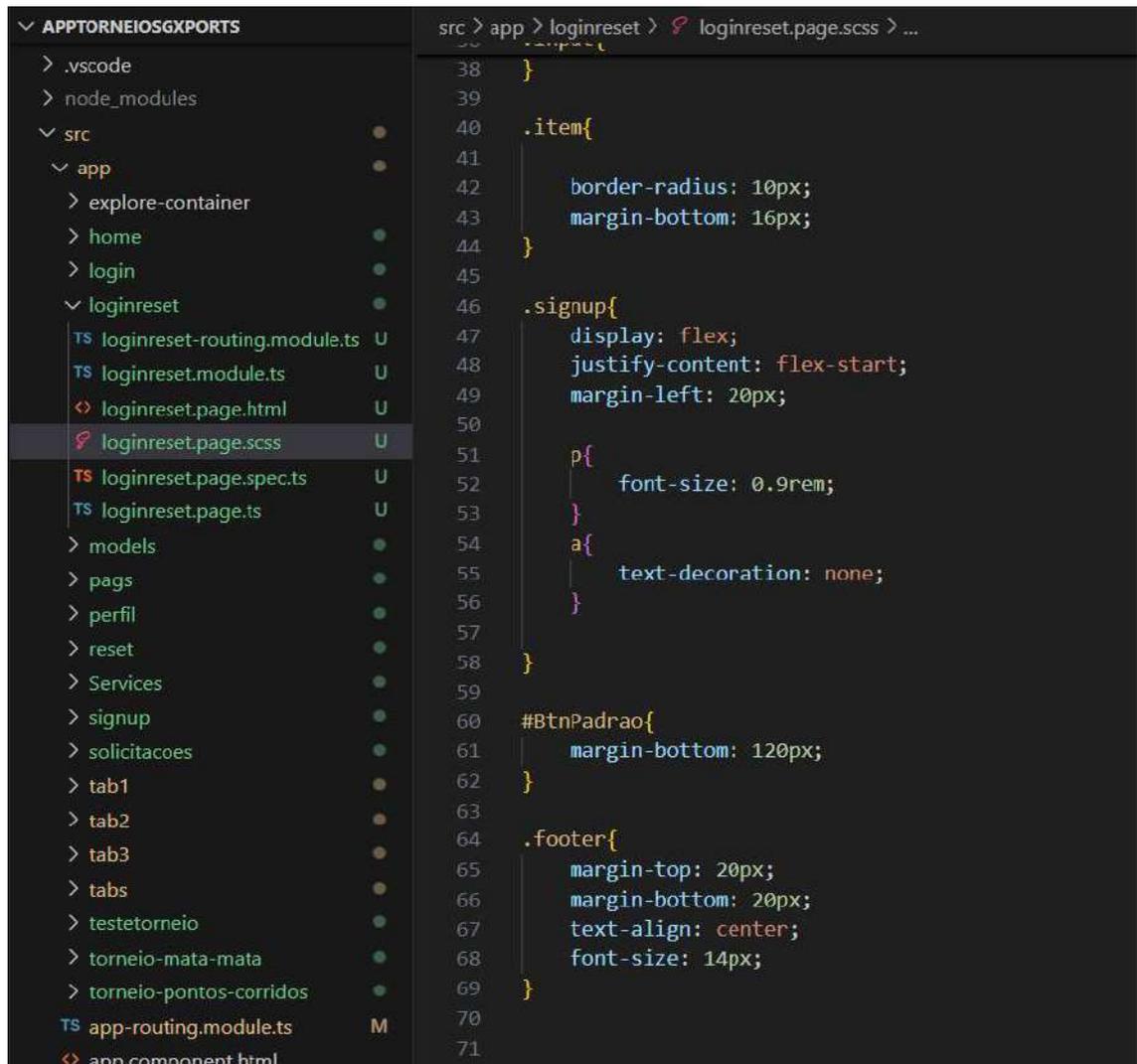
```

APPTORNEIOSGXPORTS
src > app > loginreset > loginreset.page.html > div/main-content/ion-page > ion-content/content > ion-list/ion-padding > ion-item/item
  > .vscode
  > node_modules
  > src
    > app
      > explore-container
      > home
      > login
      > loginreset
        TS loginreset-routing.module.ts U
        TS loginreset.module.ts U
        <> loginreset.page.html U
        <? loginreset.page.scss U
        TS loginreset.page.spec.ts U
        TS loginreset.page.ts U
      > models
      > pages
      > perfil
      > reset
      > Services
      > signup
      > solicitacoes
      > tab1
      > tab2
      > tab3
      > tabs
      > testetorneio
      > torneio-mata-mata
      > torneio-pontos-corridos
      TS app-routing.module.ts M
      <> app.component.html
    > OUTLINE
    > TIMELINE
  src > app > loginreset > loginreset.page.html > div/main-content/ion-page > ion-content/content > ion-list/ion-padding > ion-item/item
    8 <div class="ion-page" id="main-content">
    9 <ion-list class="ion-padding">
    10 <ion-item class="item">
    11 <ion-input placeholder="000.000.000-00" label="CPF; " label-placement="text" required="true" type="number" clear-inputs="true" class="ion-input">
    12 <ion-icon name="lock-closed-outline"></ion-icon>
    13 </ion-item>
    14 <ion-button slot="end" show="false" size="default" class="botao" (click)="reset()">Reset senha <ion-icon color="primary" name="chevron-forward"></ion-icon>
    15 </ion-button>
    16 </ion-list>
    17 <div class="signup">
    18 <p>Novo por aqui? <a style="color:#eeba2b" [routerLink]="['/signup']">Cadastre-se agora!</a></p>
    19 </div>
    20 </ion-content>
  
```

CSS

```

EXPLORER
APPTORNEIOSGXPORTS
  > .vscode
  > node_modules
  > src
    > app
      > explore-container
      > home
      > login
      > loginreset
        TS loginreset-routing.module.ts U
        TS loginreset.module.ts U
        <> loginreset.page.html U
        <? loginreset.page.scss U
        TS loginreset.page.spec.ts U
        TS loginreset.page.ts U
      > models
      > pages
      > perfil
      > reset
      > Services
      > signup
      > solicitacoes
      > tab1
      > tab2
      > tab3
      > tabs
      > testetorneio
      > torneio-mata-mata
      > torneio-pontos-corridos
      TS app-routing.module.ts M
      <> app.component.html
    > OUTLINE
    > TIMELINE
  src > app > loginreset > loginreset.page.scss > ...
    1
    2 .logoHeader{
    3   height: 100px;
    4   margin-top: 5px;
    5   margin-bottom: 5px;
    6 }
    7
    8 .toolbarHeader{
    9   height: 90px;
    10 }
    11
    12 .header{
    13   margin-top: 80px;
    14   margin-bottom: 80px;
    15   text-align: center;
    16   .subTitle{
    17     color: gray;
    18     font-size: 0.9 rem;
    19   }
    20
    21   title{
    22     font-size: 2rem;
    23   }
    24
    25   .termos{
    26     color: rgb(41, 40, 40);
    27     font-size: 0.9rem;
    28   }
    29 }
    30
    31 .form{
    32   margin-top: 150px;
    33 }
    34
    35
    36 .input{
    37   font-size: 20px;
  
```



```
src > app > loginreset > loginreset.page.scss > ...
38 }
39
40 .item{
41
42     border-radius: 10px;
43     margin-bottom: 16px;
44 }
45
46 .signup{
47     display: flex;
48     justify-content: flex-start;
49     margin-left: 20px;
50
51     p{
52         font-size: 0.9rem;
53     }
54     a{
55         text-decoration: none;
56     }
57 }
58
59 #BtnPadrao{
60     margin-bottom: 120px;
61 }
62
63
64 .footer{
65     margin-top: 20px;
66     margin-bottom: 20px;
67     text-align: center;
68     font-size: 14px;
69 }
70
71
```

Typescript:

```

src > app > loginreset > TS loginreset.page.ts > _
1 import { Router } from '@angular/router';
2 import { Component, OnInit } from '@angular/core';
3 import { AutenticacaoService } from '../services/autenticacao.service';
4 import { AlertController, ToastController } from '@ionic/angular';
5 import { waitForAsync } from '@angular/core/testing';
6
7 @Component({
8   selector: 'app-loginreset',
9   templateUrl: './loginreset.page.html',
10  styleUrls: ['./loginreset.page.scss'],
11 })
12 export class LoginresetPage implements OnInit {
13
14   email = '';
15   password = '';
16
17   constructor(public autenticacaoService: AutenticacaoService,
18     public router: Router, public alertController:AlertController, public toastController:ToastController) {}
19
20   ngOnInit() {
21   }
22
23   // Funcoes de alertas tratamento
24
25   async reset() {
26     try{
27       await this.autenticacaoService.reset(this.email);
28       this.exibirAlertasreset();
29       console.log("reset realizado com sucesso")
30       waitForAsync
31       this.router.navigate(['/login-page']);
32     }catch (error){
33
34       console.error("Nao existe esse e-mail na base de cadastro")
35       this.exibirAlertasresetfailed();
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

```

src > app > loginreset > TS loginreset.page.ts > LoginresetPage
12 export class LoginresetPage implements OnInit {
26   async reset() {
38   }
39 }
40
41   async exibirAlertasreset(){
42
43     const toast=await this.toastController.create
44     ({
45       message: 'reset realizado com sucesso',
46       duration:500
47     });
48     toast.present();
49   }
50
51   async exibirAlertasresetfailed(){
52
53     const toast=await this.toastController.create
54     ({
55       message: 'Nao existe esse e-mail na base de cadastro',
56       duration:500
57     });
58     toast.present();
59   }
60
61 }
62
63
64
65
66

```

Signup page

html

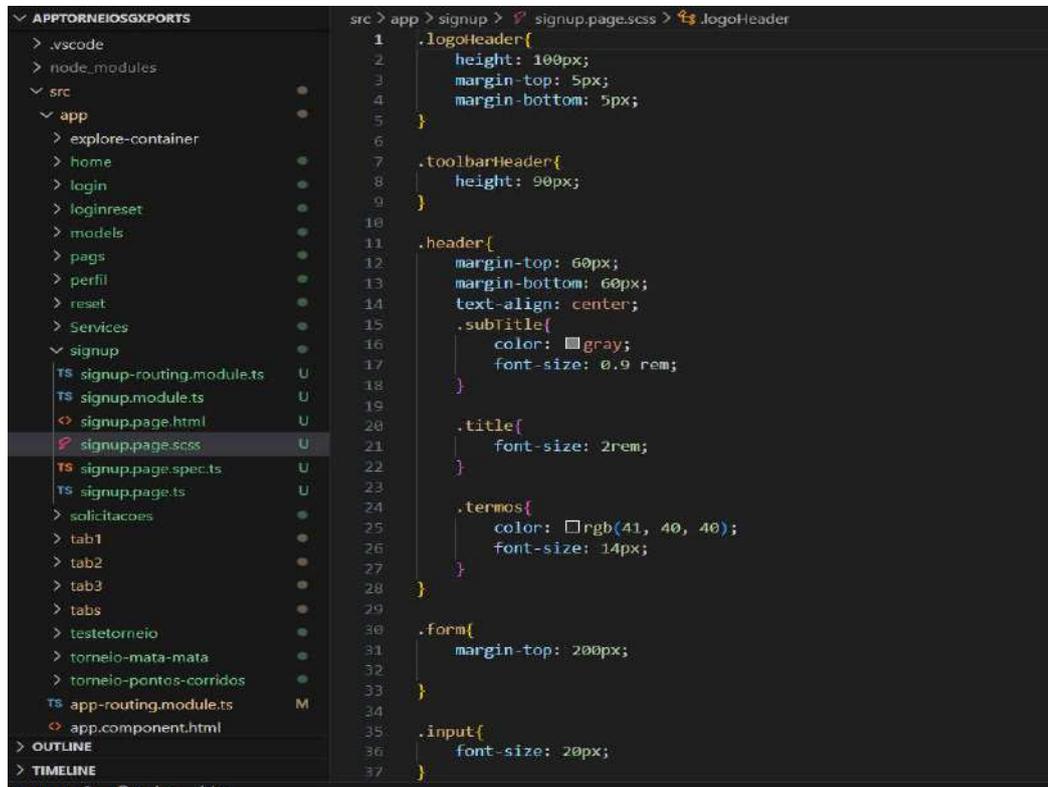
```

▼ APPFRAMEWORKS
  > encode
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  > loginreset
  > models
  > page
  > perfil
  > reset
  > Services
  > Signup
  > signup-routing.module.ts
  > signup.module.ts
  > signuppage.html
  > signuppage.scss
  > signuppage.spec.ts
  > signuppage.ts
  > solicitacoes
  > tab1
  > tab2
  > tab3
  > tabs
  > testformaria
  > testeio-mata-mata
  > testeio-puntio-convidos
  > app-routing.module.ts
  > app.component.html
  > login
  > loginreset
  > models
  > page
  > perfil
  > reset
  > Services
  > Signup
  > signup-routing.module.ts
  > signup.module.ts
  > signuppage.html
  > signuppage.scss
  > signuppage.spec.ts
  > signuppage.ts
  > solicitacoes
  > tab1
  > tab2
  > tab3
  > tabs
  > testeio-mata-mata
  > testeio-puntio-convidos
  > app-routing.module.ts
  > app.component.html

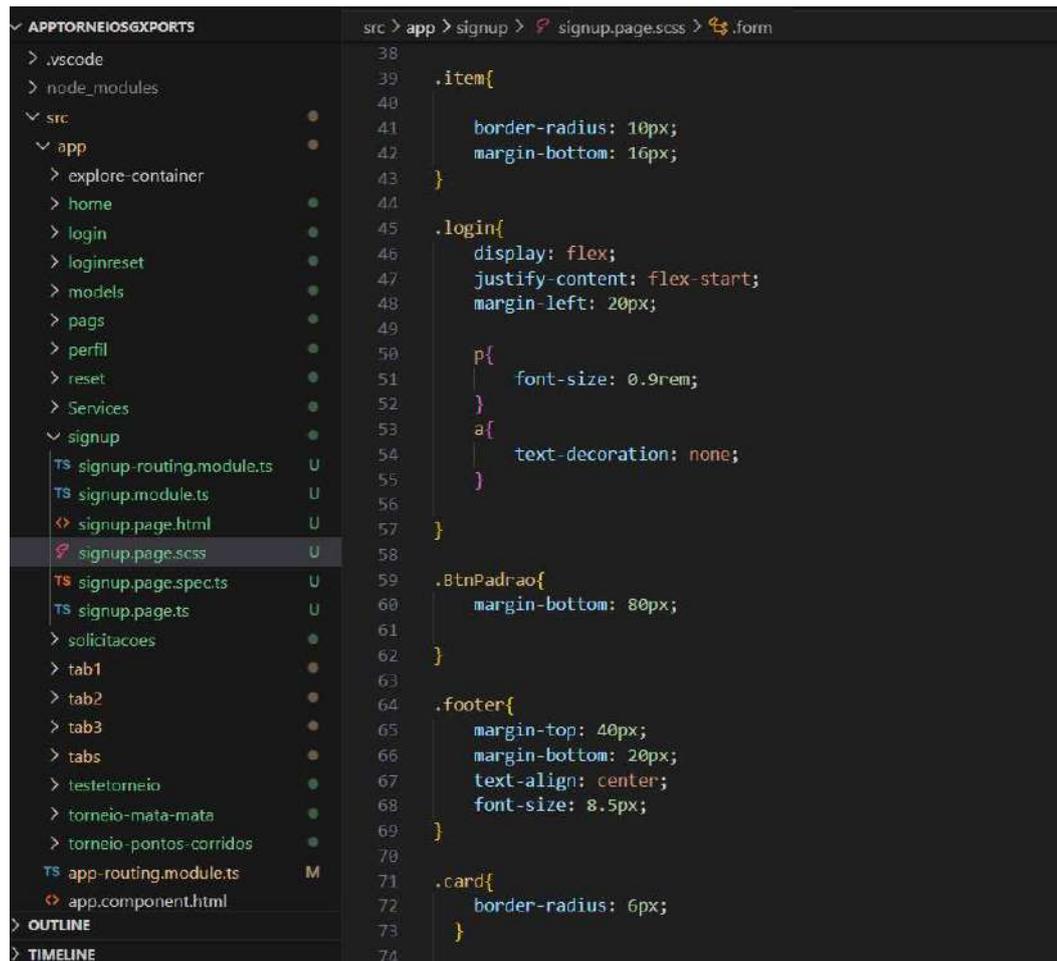
src/app/signup/signuppage.html | @ion-menu | @ion-header
1 <ion-menu contentId="main-content">
2 <ion-header>
3 <ion-toolbar>
4 <ion-title>Menu /ion-title
5 </ion-toolbar>
6 </ion-header>
7 </ion-menu>
8 <div class="ion-page" id="main-content">
9 <ion-header>
10 <ion-toolbar>
11 <ion-buttons slot="start">
12 <ion-menu-button>/ion-menu-button
13 </ion-buttons>
14 <ion-toolbar class="toolbarheader">
15 </ion-toolbar>
16 </ion-header>
17 </ion-toolbar>
18 </ion-header>
19 <ion-content [fullscreen]="true" class="content">
20 <ion-card class="card">
21 <div class="header">
22 <ion-title class="title">Seja Bem Vindo</ion-title>
23 <p class="subtitle">Cadastre-se para criar sua conta</p>
24 </div>
25 <ion-list class="ion-padding">
26 <ion-item class="item">
27 <input placeholder="Inclua o seu nome completo" label="Nome" type="text" req
28 <input type="password" label="Senha" type="password" required="true" clear-inp
29 </ion-item>
30 <ion-item class="item">
31 <input placeholder="Inclua o seu nome completo" label="Nome" type="text" req
32 <input type="password" label="Senha" type="password" required="true" clear-inp
33 </ion-item>
34 </ion-list>
35 <div class="login">
36 <p><a href="#">Já tem conta?</a> <a href="#">entre na sua conta.</a></p>
37 </div>
38 </ion-card>
39 <div>
40 <ion-fab vertical="bottom" horizontal="end" class="BtnPadrao">
41 <ion-fab-button [routerLink]="['/tabs/tab1']">
42 <ion-icon name="chevron-forward" style="color:#000000"/></ion-fab-button>
43 </div>
44 </ion-content>

```

CSS



```
src > app > signup > signup.page.scss > .logoHeader
1  .logoHeader{
2      height: 100px;
3      margin-top: 5px;
4      margin-bottom: 5px;
5  }
6
7  .toolbarHeader{
8      height: 90px;
9  }
10
11 .header{
12     margin-top: 60px;
13     margin-bottom: 60px;
14     text-align: center;
15     .subtitle{
16         color: gray;
17         font-size: 0.9 rem;
18     }
19
20     .title{
21         font-size: 2rem;
22     }
23
24     .termos{
25         color: rgb(41, 40, 40);
26         font-size: 14px;
27     }
28 }
29
30 .form{
31     margin-top: 200px;
32 }
33
34 .input{
35     font-size: 20px;
36 }
37
```



```
APPTORNEIOSGEXPORTS
├── .vscode
├── node_modules
├── src
│   ├── app
│   │   ├── explore-container
│   │   ├── home
│   │   ├── login
│   │   ├── loginreset
│   │   ├── models
│   │   ├── pags
│   │   ├── perfil
│   │   ├── reset
│   │   ├── Services
│   │   └── signup
│   │       ├── signup-routing.module.ts
│   │       ├── signup.module.ts
│   │       ├── signup.page.html
│   │       ├── signup.page.scss
│   │       ├── signup.page.spec.ts
│   │       └── signup.page.ts
│   ├── solicitacoes
│   ├── tab1
│   ├── tab2
│   ├── tab3
│   ├── tabs
│   ├── testetorneio
│   ├── torneio-mata-mata
│   ├── torneio-pontos-corridos
│   ├── app-routing.module.ts
│   └── app.component.html
├── OUTLINE
└── TIMELINE

src > app > signup > signup.page.scss > form
38
39 .item{
40
41     border-radius: 10px;
42     margin-bottom: 16px;
43 }
44
45 .login{
46     display: flex;
47     justify-content: flex-start;
48     margin-left: 20px;
49
50     p{
51         font-size: 0.9rem;
52     }
53     a{
54         text-decoration: none;
55     }
56 }
57
58 .BtnPadrao{
59     margin-bottom: 80px;
60 }
61
62
63
64 .footer{
65     margin-top: 40px;
66     margin-bottom: 20px;
67     text-align: center;
68     font-size: 8.5px;
69 }
70
71 .card{
72     border-radius: 6px;
73 }
74
```

Typescript

```

src > app > signup > TS signup.page.ts > ...
1 import { Router } from '@angular/router';
2 import { Component, OnInit } from '@angular/core';
3 import { AutenticacaoService } from '../Services/autenticacao.service';
4 import { AlertController, ToastController } from '@ionic/angular';
5 import { waitForAsync } from '@angular/core/testing';
6
7 @Component({
8   selector: 'app-signup',
9   templateUrl: './signup.page.html',
10  styleUrls: ['./signup.page.scss'],
11 })
12 export class SignupPage implements OnInit {
13
14   email = '';
15   password = '';
16
17   constructor(public autenticacaoService: AutenticacaoService,
18     public router: Router, public alertController:Alertcontroller, public toastcontroller:toastcontroller) {}
19
20   ngOnInit() {
21
22
23
24
25   async signup() {
26     try{
27       await this.autenticacaoService.signup(this.email, this.password);
28       this.exibirAlertasemalSucessoCadastro();
29       waitForAsync
30       //
31     }catch (error){
32       console.error('Erro de cadastro: ', error)
33       this.exibirAlertasemalErroCadastro();
34     }finally{
35       this.router.navigate(['/perfil']);
36     }
37   }
38 }

```

```

src > app > signup > TS signup.page.ts > SignupPage > password
12 export class SignupPage implements OnInit {
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38   async exibirAlertasemalErroCadastro(){
39     const toast=await this.toastController.create
40     ({
41       message: 'Erro ao se cadastrar',
42       duration:3000
43     });
44     toast.present();
45   }
46
47   async exibirAlertasemalSucessoCadastro(){
48     const toast=await this.toastController.create
49     ({
50       message: 'Cadastro realizado com sucesso',
51       duration:3000
52     });
53     toast.present();
54   }
55 }
56
57

```

Autenticacao.Service.ts

```
APPTORNEIOSGXPORTS
├── node_modules
├── src
│   ├── app
│   │   ├── explore-container
│   │   ├── home
│   │   ├── login
│   │   ├── loginreset
│   │   ├── models
│   │   ├── pags
│   │   ├── perfil
│   │   ├── reset
│   │   └── Services
│   │       ├── autenticacao.service.spec.ts U
│   │       └── autenticacao.service.ts U
│   ├── jogador.service.spec.ts U
│   ├── jogador.service.ts U
│   ├── signup
│   ├── solicitacoes
│   ├── tab1
│   ├── tab2
│   ├── tab3
│   ├── tabs
│   ├── testetorneio
│   ├── torneio-mata-mata
│   ├── torneio-pontos-corridos
│   ├── app-routing.module.ts M
│   ├── app.component.html
│   ├── app.component.scss
│   └── app.component.spec.ts
└── src > app > Services > TS autenticacao.service.ts > AutenticacaoService
    1 import { Injectable } from '@angular/core';
    2 import { AngularFireAuth } from '@angular/fire/compat/auth';
    3
    4 import { Observable, from } from 'rxjs';
    5 import { GoogleAuthProvider } from 'firebase/auth';
    6
    7 @injectable({
    8   providedIn: 'root'
    9 })
   10 export class AutenticacaoService {
   11
   12   constructor(public afAuth: AngularFireAuth) { }
   13
   14   public login(email:any , password:any) {
   15     return this.afAuth.signInWithEmailAndPassword(email, password);
   16   }
   17
   18   public signup(email:any, password:any) {
   19     return this.afAuth.createUserWithEmailAndPassword(email, password);
   20   }
   21
   22   public reset(email:any) {
   23     return this.afAuth.sendPasswordResetEmail(email);
   24   }
   25
   26   public logged(){
   27     console.log(this.afAuth.user);
   28     return this.afAuth.user;
   29   }
   30
   31   public loginwithGoogle(){
   32     this.afAuth.signInWithPopup(new GoogleAuthProvider);
   33   }
   34 }
```

Perfil page

Html

```

APPTORNEIOSEXPORTS  src > app > perfil > perfil.page.html > ion-header
  > node_modules      1 <ion-header>
  > src                2 <ion-toolbar color="m">
  > app                3 <ion-title>Perfil</ion-title>
  > explore-container  4 </ion-toolbar>
  > home               5 </ion-header>
  > login              6
  > loginreset         7 <ion-content>
  > models             8 <ion-card>
  > pages              9 <ion-card-header>
  > perfil             10 <!-- Avatar com input de imagem -->
  > perfil-routing.module.ts U 11 <ion-avatar (click)="selectImage()">
  > perfil.module.ts   U 12 <img [src]="photo || 'assets/default-profile.png'" alt="Foto do Perfil" />
  > perfil.page.html   U 13 </ion-avatar>
  > perfil.page.scss   U 14 <input type="file" #fileInput accept="image/*" (change)="onFileSelected($event)" hidden />
  > perfil.page.spec.ts U 15 </ion-card-header>
  > reset              16 <ion-card-content>
  > Services           17 <!-- Nome -->
  > signup             18 <ion-item>
  > solicitacoes      19 <ion-label position="stacked">Nome</ion-label>
  > tab1               20 <ion-input [(ngModel)]="name" placeholder="Digite seu nome"></ion-input>
  > tab2               21 </ion-item>
  > tab3               22 <!-- E-mail -->
  > tabs               23 <ion-item>
  > testeTorneio      24 <ion-label position="stacked">E-mail</ion-label>
  > torneio-mata-mata  25 <ion-input [(ngModel)]="email" placeholder="Digite seu e-mail"></ion-input>
  > torneio-pontos-corridos  26 </ion-item>
  > app-routing.module.ts M 27 <!-- Apelido -->
  > app.component.html  28 <ion-item>
  > app.component.scss  29 <ion-label position="stacked">Apelido</ion-label>
  > OUTLINE            30 <ion-input [(ngModel)]="nickname" placeholder="Digite seu apelido"></ion-input>
  >                    31 </ion-item>
  >                    32
  >                    33
  >                    34
  >                    35
  >                    36
  >                    37
  >                    38
  >                    39
  >                    40
  >                    41
  >                    42
  >                    43
  >                    44
  >                    45
  >                    46
  >                    47
  >                    48
  >                    49
  >                    50
  >                    51
  >                    52
  >                    53
  >                    54
  >                    55
  >                    56
  >                    57
  >                    58
  >                    59
  >                    60
  >                    61
  >                    62
  >                    63
  >                    64
  >                    65
  >                    66
  >                    67
  >                    68
  >                    69
  >                    70
  >                    71
  >                    72
  >                    73
  >                    74
  >                    75
  >                    76
  >                    77
  >                    78
  >                    79
  >                    80
  >                    81
  >                    82
  >                    83
  >                    84
  >                    85
  >                    86
  >                    87
  >                    88
  >                    89
  >                    90
  >                    91
  >                    92
  >                    93
  >                    94
  >                    95
  >                    96
  >                    97
  >                    98
  >                    99
  >                    100
  >                    101
  >                    102
  >                    103
  >                    104
  >                    105
  >                    106
  >                    107
  >                    108
  >                    109
  >                    110
  >                    111
  >                    112
  >                    113
  >                    114
  >                    115
  >                    116
  >                    117
  >                    118
  >                    119
  >                    120
  >                    121
  >                    122
  >                    123
  >                    124
  >                    125
  >                    126
  >                    127
  >                    128
  >                    129
  >                    130
  >                    131
  >                    132
  >                    133
  >                    134
  >                    135
  >                    136
  >                    137
  >                    138
  >                    139
  >                    140
  >                    141
  >                    142
  >                    143
  >                    144
  >                    145
  >                    146
  >                    147
  >                    148
  >                    149
  >                    150
  >                    151
  >                    152
  >                    153
  >                    154
  >                    155
  >                    156
  >                    157
  >                    158
  >                    159
  >                    160
  >                    161
  >                    162
  >                    163
  >                    164
  >                    165
  >                    166
  >                    167
  >                    168
  >                    169
  >                    170
  >                    171
  >                    172
  >                    173
  >                    174
  >                    175
  >                    176
  >                    177
  >                    178
  >                    179
  >                    180
  >                    181
  >                    182
  >                    183
  >                    184
  >                    185
  >                    186
  >                    187
  >                    188
  >                    189
  >                    190
  >                    191
  >                    192
  >                    193
  >                    194
  >                    195
  >                    196
  >                    197
  >                    198
  >                    199
  >                    200
  >                    201
  >                    202
  >                    203
  >                    204
  >                    205
  >                    206
  >                    207
  >                    208
  >                    209
  >                    210
  >                    211
  >                    212
  >                    213
  >                    214
  >                    215
  >                    216
  >                    217
  >                    218
  >                    219
  >                    220
  >                    221
  >                    222
  >                    223
  >                    224
  >                    225
  >                    226
  >                    227
  >                    228
  >                    229
  >                    230
  >                    231
  >                    232
  >                    233
  >                    234
  >                    235
  >                    236
  >                    237
  >                    238
  >                    239
  >                    240
  >                    241
  >                    242
  >                    243
  >                    244
  >                    245
  >                    246
  >                    247
  >                    248
  >                    249
  >                    250
  >                    251
  >                    252
  >                    253
  >                    254
  >                    255
  >                    256
  >                    257
  >                    258
  >                    259
  >                    260
  >                    261
  >                    262
  >                    263
  >                    264
  >                    265
  >                    266
  >                    267
  >                    268
  >                    269
  >                    270
  >                    271
  >                    272
  >                    273
  >                    274
  >                    275
  >                    276
  >                    277
  >                    278
  >                    279
  >                    280
  >                    281
  >                    282
  >                    283
  >                    284
  >                    285
  >                    286
  >                    287
  >                    288
  >                    289
  >                    290
  >                    291
  >                    292
  >                    293
  >                    294
  >                    295
  >                    296
  >                    297
  >                    298
  >                    299
  >                    300
  >                    301
  >                    302
  >                    303
  >                    304
  >                    305
  >                    306
  >                    307
  >                    308
  >                    309
  >                    310
  >                    311
  >                    312
  >                    313
  >                    314
  >                    315
  >                    316
  >                    317
  >                    318
  >                    319
  >                    320
  >                    321
  >                    322
  >                    323
  >                    324
  >                    325
  >                    326
  >                    327
  >                    328
  >                    329
  >                    330
  >                    331
  >                    332
  >                    333
  >                    334
  >                    335
  >                    336
  >                    337
  >                    338
  >                    339
  >                    340
  >                    341
  >                    342
  >                    343
  >                    344
  >                    345
  >                    346
  >                    347
  >                    348
  >                    349
  >                    350
  >                    351
  >                    352
  >                    353
  >                    354
  >                    355
  >                    356
  >                    357
  >                    358
  >                    359
  >                    360
  >                    361
  >                    362
  >                    363
  >                    364
  >                    365
  >                    366
  >                    367
  >                    368
  >                    369
  >                    370
  >                    371
  >                    372
  >                    373
  >                    374
  >                    375
  >                    376
  >                    377
  >                    378
  >                    379
  >                    380
  >                    381
  >                    382
  >                    383
  >                    384
  >                    385
  >                    386
  >                    387
  >                    388
  >                    389
  >                    390
  >                    391
  >                    392
  >                    393
  >                    394
  >                    395
  >                    396
  >                    397
  >                    398
  >                    399
  >                    400
  >                    401
  >                    402
  >                    403
  >                    404
  >                    405
  >                    406
  >                    407
  >                    408
  >                    409
  >                    410
  >                    411
  >                    412
  >                    413
  >                    414
  >                    415
  >                    416
  >                    417
  >                    418
  >                    419
  >                    420
  >                    421
  >                    422
  >                    423
  >                    424
  >                    425
  >                    426
  >                    427
  >                    428
  >                    429
  >                    430
  >                    431
  >                    432
  >                    433
  >                    434
  >                    435
  >                    436
  >                    437
  >                    438
  >                    439
  >                    440
  >                    441
  >                    442
  >                    443
  >                    444
  >                    445
  >                    446
  >                    447
  >                    448
  >                    449
  >                    450
  >                    451
  >                    452
  >                    453
  >                    454
  >                    455
  >                    456
  >                    457
  >                    458
  >                    459
  >                    460
  >                    461
  >                    462
  >                    463
  >                    464
  >                    465
  >                    466
  >                    467
  >                    468
  >                    469
  >                    470
  >                    471
  >                    472
  >                    473
  >                    474
  >                    475
  >                    476
  >                    477
  >                    478
  >                    479
  >                    480
  >                    481
  >                    482
  >                    483
  >                    484
  >                    485
  >                    486
  >                    487
  >                    488
  >                    489
  >                    490
  >                    491
  >                    492
  >                    493
  >                    494
  >                    495
  >                    496
  >                    497
  >                    498
  >                    499
  >                    500
  >                    501
  >                    502
  >                    503
  >                    504
  >                    505
  >                    506
  >                    507
  >                    508
  >                    509
  >                    510
  >                    511
  >                    512
  >                    513
  >                    514
  >                    515
  >                    516
  >                    517
  >                    518
  >                    519
  >                    520
  >                    521
  >                    522
  >                    523
  >                    524
  >                    525
  >                    526
  >                    527
  >                    528
  >                    529
  >                    530
  >                    531
  >                    532
  >                    533
  >                    534
  >                    535
  >                    536
  >                    537
  >                    538
  >                    539
  >                    540
  >                    541
  >                    542
  >                    543
  >                    544
  >                    545
  >                    546
  >                    547
  >                    548
  >                    549
  >                    550
  >                    551
  >                    552
  >                    553
  >                    554
  >                    555
  >                    556
  >                    557
  >                    558
  >                    559
  >                    560
  >                    561
  >                    562
  >                    563
  >                    564
  >                    565
  >                    566
  >                    567
  >                    568
  >                    569
  >                    570
  >                    571
  >                    572
  >                    573
  >                    574
  >                    575
  >                    576
  >                    577
  >                    578
  >                    579
  >                    580
  >                    581
  >                    582
  >                    583
  >                    584
  >                    585
  >                    586
  >                    587
  >                    588
  >                    589
  >                    590
  >                    591
  >                    592
  >                    593
  >                    594
  >                    595
  >                    596
  >                    597
  >                    598
  >                    599
  >                    600
  >                    601
  >                    602
  >                    603
  >                    604
  >                    605
  >                    606
  >                    607
  >                    608
  >                    609
  >                    610
  >                    611
  >                    612
  >                    613
  >                    614
  >                    615
  >                    616
  >                    617
  >                    618
  >                    619
  >                    620
  >                    621
  >                    622
  >                    623
  >                    624
  >                    625
  >                    626
  >                    627
  >                    628
  >                    629
  >                    630
  >                    631
  >                    632
  >                    633
  >                    634
  >                    635
  >                    636
  >                    637
  >                    638
  >                    639
  >                    640
  >                    641
  >                    642
  >                    643
  >                    644
  >                    645
  >                    646
  >                    647
  >                    648
  >                    649
  >                    650
  >                    651
  >                    652
  >                    653
  >                    654
  >                    655
  >                    656
  >                    657
  >                    658
  >                    659
  >                    660
  >                    661
  >                    662
  >                    663
  >                    664
  >                    665
  >                    666
  >                    667
  >                    668
  >                    669
  >                    670
  >                    671
  >                    672
  >                    673
  >                    674
  >                    675
  >                    676
  >                    677
  >                    678
  >                    679
  >                    680
  >                    681
  >                    682
  >                    683
  >                    684
  >                    685
  >                    686
  >                    687
  >                    688
  >                    689
  >                    690
  >                    691
  >                    692
  >                    693
  >                    694
  >                    695
  >                    696
  >                    697
  >                    698
  >                    699
  >                    700
  >                    701
  >                    702
  >                    703
  >                    704
  >                    705
  >                    706
  >                    707
  >                    708
  >                    709
  >                    710
  >                    711
  >                    712
  >                    713
  >                    714
  >                    715
  >                    716
  >                    717
  >                    718
  >                    719
  >                    720
  >                    721
  >                    722
  >                    723
  >                    724
  >                    725
  >                    726
  >                    727
  >                    728
  >                    729
  >                    730
  >                    731
  >                    732
  >                    733
  >                    734
  >                    735
  >                    736
  >                    737
  >                    738
  >                    739
  >                    740
  >                    741
  >                    742
  >                    743
  >                    744
  >                    745
  >                    746
  >                    747
  >                    748
  >                    749
  >                    750
  >                    751
  >                    752
  >                    753
  >                    754
  >                    755
  >                    756
  >                    757
  >                    758
  >                    759
  >                    760
  >                    761
  >                    762
  >                    763
  >                    764
  >                    765
  >                    766
  >                    767
  >                    768
  >                    769
  >                    770
  >                    771
  >                    772
  >                    773
  >                    774
  >                    775
  >                    776
  >                    777
  >                    778
  >                    779
  >                    780
  >                    781
  >                    782
  >                    783
  >                    784
  >                    785
  >                    786
  >                    787
  >                    788
  >                    789
  >                    790
  >                    791
  >                    792
  >                    793
  >                    794
  >                    795
  >                    796
  >                    797
  >                    798
  >                    799
  >                    800
  >                    801
  >                    802
  >                    803
  >                    804
  >                    805
  >                    806
  >                    807
  >                    808
  >                    809
  >                    810
  >                    811
  >                    812
  >                    813
  >                    814
  >                    815
  >                    816
  >                    817
  >                    818
  >                    819
  >                    820
  >                    821
  >                    822
  >                    823
  >                    824
  >                    825
  >                    826
  >                    827
  >                    828
  >                    829
  >                    830
  >                    831
  >                    832
  >                    833
  >                    834
  >                    835
  >                    836
  >                    837
  >                    838
  >                    839
  >                    840
  >                    841
  >                    842
  >                    843
  >                    844
  >                    845
  >                    846
  >                    847
  >                    848
  >                    849
  >                    850
  >                    851
  >                    852
  >                    853
  >                    854
  >                    855
  >                    856
  >                    857
  >                    858
  >                    859
  >                    860
  >                    861
  >                    862
  >                    863
  >                    864
  >                    865
  >                    866
  >                    867
  >                    868
  >                    869
  >                    870
  >                    871
  >                    872
  >                    873
  >                    874
  >                    875
  >                    876
  >                    877
  >                    878
  >                    879
  >                    880
  >                    881
  >                    882
  >                    883
  >                    884
  >                    885
  >                    886
  >                    887
  >                    888
  >                    889
  >                    890
  >                    891
  >                    892
  >                    893
  >                    894
  >                    895
  >                    896
  >                    897
  >                    898
  >                    899
  >                    900
  >                    901
  >                    902
  >                    903
  >                    904
  >                    905
  >                    906
  >                    907
  >                    908
  >                    909
  >                    910
  >                    911
  >                    912
  >                    913
  >                    914
  >                    915
  >                    916
  >                    917
  >                    918
  >                    919
  >                    920
  >                    921
  >                    922
  >                    923
  >                    924
  >                    925
  >                    926
  >                    927
  >                    928
  >                    929
  >                    930
  >                    931
  >                    932
  >                    933
  >                    934
  >                    935
  >                    936
  >                    937
  >                    938
  >                    939
  >                    940
  >                    941
  >                    942
  >                    943
  >                    944
  >                    945
  >                    946
  >                    947
  >                    948
  >                    949
  >                    950
  >                    951
  >                    952
  >                    953
  >                    954
  >                    955
  >                    956
  >                    957
  >                    958
  >                    959
  >                    960
  >                    961
  >                    962
  >                    963
  >                    964
  >                    965
  >                    966
  >                    967
  >                    968
  >                    969
  >                    970
  >                    971
  >                    972
  >                    973
  >                    974
  >                    975
  >                    976
  >                    977
  >                    978
  >                    979
  >                    980
  >                    981
  >                    982
  >                    983
  >                    984
  >                    985
  >                    986
  >                    987
  >                    988
  >                    989
  >                    990
  >                    991
  >                    992
  >                    993
  >                    994
  >                    995
  >                    996
  >                    997
  >                    998
  >                    999
  >                    1000
  >                    1001
  >                    1002
  >                    1003
  >                    1004
  >                    1005
  >                    1006
  >                    1007
  >                    1008
  >                    1009
  >                    1010
  >                    1011
  >                    1012
  >                    1013
  >                    1014
  >                    1015
  >                    1016
  >                    1017
  >                    1018
  >                    1019
  >                    1020
  >                    1021
  >                    1022
  >                    1023
  >                    1024
  >                    1025
  >                    1026
  >                    1027
  >                    1028
  >                    1029
  >                    1030
  >                    1031
  >                    1032
  >                    1033
  >                    1034
  >                    1035
  >                    1036
  >                    1037
  >                    1038
  >                    1039
  >                    1040
  >                    1041
  >                    1042
  >                    1043
  >                    1044
  >                    1045
  >                    1046
  >                    1047
  >                    1048
  >                    1049
  >                    1050
  >                    1051
  >                    1052
  >                    1053
  >                    1054
  >                    1055
  >                    1056
  >                    1057
  >                    1058
  >                    1059
  >                    1060
  >                    1061
  >                    1062
  >                    1063
  >                    1064
  >                    1065
  >                    1066
  >                    1067
  >                    1068
  >                    1069
  >                    1070
  >                    1071
  >                    1072
  >                    1073
  >                    1074
  >                    1075
  >                    1076
  >                    1077
  >                    1078
  >                    1079
  >                    1080
  >                    1081
  >                    1082
  >                    1083
  >                    1084
  >                    1085
  >                    1086
  >                    1087
  >                    1088
  >                    1089
  >                    1090
  >                    1091
  >                    1092
  >                    1093
  >                    1094
  >                    1095
  >                    1096
  >                    1097
  >                    1098
  >                    1099
  >                    1100
  >                    1101
  >                    1102
  >                    1103
  >                    1104
  >                    1105
  >                    1106
  >                    1107
  >                    1108
  >                    1109
  >                    1110
  >                    1111
  >                    1112
  >                    1113
  >                    1114
  >                    1115
  >                    1116
  >                    1117
  >                    1118
  >                    1119
  >                    1120
  >                    1121
  >                    1122
  >                    1123
  >                    1124
  >                    1125
  >                    1126
  >                    1127
  >                    1128
  >                    1129
  >                    1130
  >                    1131
  >                    1132
  >                    1133
  >                    1134
  >                    1135
  >                    1136
  >                    1137
  >                    1138
  >                    1139
  >                    1140
  >                    1141
  >                    1142
  >                    1143
  >                    1144
  >                    1145
  >                    1146
  >                    1147
  >                    1148
  >                    1149
  >                    1150
  >                    1151
  >                    1152
  >                    1153
  >                    1154
  >                    1155
  >                    1156
  >                    1157
  >                    1158
  >                    1159
  >                    1160
  >                    1161
  >                    1162
  >                    1163
  >                    1164
  >                    1165
  >                    1166
  >                    1167
  >                    1168
  &gt
```

```

> home ● 71
> login ● 72
> loginreset ● 73
> models ● 74
> pags ● 75
> perfil ● 76
  TB perfil-routing.module.ts U 77
  TB perfil.module.ts U 78
  P perfil.page.html U 80
  P perfil.page.scss U 81
  TB perfil.page.spect.ts U 82
  TB perfil.page.ts U 83
> reset ● 84
> Services ● 85
> signup ● 86
> solicitacoes ● 87
> tab1 ● 88
> tab2 ● 89
> tab3 ● 90
> tabs ● 91
> testetorneio ● 92
> torneio-mata-mata ● 93
> torneio-pontos-corridas ● 94
TB app-routing.module.ts M 95
P app.component.html
P app.component.scss

```

```

71 <ion-item>
72   <ion-label position="stacked">Ano</ion-label>
73   <ion-input [(ngModel)]="year" type="number" placeholder="Ano" (ionChange)="calculateAge()"></ion-input>
74 </ion-item>
75 </div>
76
77 <!-- Idade -->
78 <ion-item>
79   <ion-label>Idade: </ion-label>
80   <ion-input [value]="age" readonly></ion-input>
81 </ion-item>
82
83 <!-- ID do Perfil -->
84 <ion-item>
85   <ion-label position="stacked">ID do Perfil</ion-label>
86   <ion-input [(ngModel)]="profileId" readonly></ion-input>
87 </ion-item>
88
89 <!-- Botão Salvar Perfil -->
90 <ion-button expand="block" (click)="saveProfile()">SALVAR PERFIL</ion-button>
91
92 </ion-card-content>
93 </ion-card>
94 </ion-content>
95

```

Typescript

```

APPTORNEIOSPORTS
  > node_modules
  > src
  > app
    > explore-container
    > home
    > login
    > loginreset
    > models
    > pags
    > perfil
      TB perfil-routing.module.ts U
      TB perfil.module.ts U
      P perfil.page.html U
      P perfil.page.scss U
      TB perfil.page.spect.ts U
      TB perfil.page.ts U
    > reset
    > Services
    > signup
    > solicitacoes
    > tab1
    > tab2
    > tab3
    > tabs
    > testetorneio
    > torneio-mata-mata
    > torneio-pontos-corridas
  TB app-routing.module.ts M
  P app.component.html
  P app.component.scss

```

```

src > app > perfil > TB perfil.page.ts > ...
1
2 import { Component, ViewChild, ElementRef } from '@angular/core';
3 import { Router } from '@angular/router';
4
5 @component({
6   selector: 'app-perfil',
7   templateUrl: './perfil.page.html',
8   styleUrls: ['./perfil.page.scss'],
9 })
10 export class PerfilPage {
11
12   @ViewChild('fileInput', { static: false }) fileInput!: ElementRef; // Referência ao input de arquivo
13
14   // Variáveis para armazenar os dados do formulário
15   name!: string;
16   email!: string;
17   nickname!: string;
18   shirtNumber!: number;
19   position!: string;
20   day!: number;
21   month!: number;
22   year!: number;
23   age!: number;
24   profileId!: number = 12345; // Exemplo de ID fixo
25   photo!: string | ArrayBuffer = ''; // Variável para armazenar a URL da foto
26
27   constructor(public router: Router) {}
28
29   // Função para abrir o seletor de arquivo de avatar
30   selectImage() {
31     if (this.fileInput && this.fileInput.nativeElement) {
32       this.fileInput.nativeElement.click(); // Aciona o clique no input de arquivo oculto
33     }
34   }
35

```

```

APPTORNEIOSXPORTS
  src
    app
      perfil
        perfil-routing.module.ts U
        perfil.module.ts U
        perfil.page.html U
        perfil.page.scss U
        perfil.page.spec.ts U
        perfil.page.ts U
        reset
        Services
        signup
        solicitacoes
        tab1
        tab2
        tab3
        tabs
        testetorneio
        torneio-mata-mata
        torneio-pontos-corridos
  src > app > perfil > TS perfil.page.ts > PerfilPage > email
  10 export class PerfilPage {
    // Função para tratar o upload de arquivo
    onFileSelected(event: any) {
      const file = event.target.files[0];
      if (file) {
        const reader = new FileReader();
        reader.onload = () => {
          this.photo = reader.result || ''; // garante que 'null' não seja atribuído
        };
        reader.readAsDataURL(file);
      }
    }
    // Função para calcular a idade com base na data de nascimento
    calculateAge() {
      if (this.day && this.month && this.year) {
        const birthDate = new Date(this.year, this.month - 1, this.day);
        const ageDiffMs = Date.now() - birthDate.getTime();
        const ageDate = new Date(ageDiffMs);
        this.age = Math.abs(ageDate.getUTCFullYear() - 1970);
      }
    }
  }
  src > app > perfil > TS perfil.page.ts > PerfilPage > calculateAge
  10 export class PerfilPage {
    // Função para salvar o perfil
    saveProfile() {
      // lógica de salvamento - por exemplo, enviar os dados para uma API ou localStorage
      console.log('Perfil salvo', {
        name: this.name,
        email: this.email,
        nickname: this.nickname,
        shirtnumber: this.shirtnumber,
        position: this.position,
        age: this.age,
        profileid: this.profileid,
      });
    }
  }

```

Solicitacoes page

html

```

src > app > solicitacoes > solicitacoes.page.html > ion-header
1 <ion-header>
2 <ion-toolbar>
3   <ion-buttons slot="start">
4     <ion-back-button class="botao-sair" defaultHref="/tabs/tab1"></ion-back-button> <!-- Botão de sair -->
5   </ion-buttons>
6   <ion-title>
7     <div class="titulo">
8       minhas solicitações
9     </div>
10  </ion-title>
11 </ion-toolbar>
12 </ion-header>
13
14 <ion-content>
15 <ion-list>
16 <ion-item *ngor="let solicitacao of solicitacoes">
17   <ion-avatar slot="start">
18     <img [src]="solicitacao.imagem" alt="Avatar">
19   </ion-avatar>
20   <ion-buttons slot="end">
21     <ion-button color="success" (click)="aprovarSolicitacao(solicitacao)">
22       <ion-icon slot="icon-only" name="checkmark-outline"></ion-icon>
23     </ion-button>
24     <ion-button color="danger" (click)="rejeitarSolicitacao(solicitacao)">
25       <ion-icon slot="icon-only" name="close-outline"></ion-icon>
26     </ion-button>
27   </ion-buttons>
28 </ion-item>
29 </ion-list>
30 </ion-content>
31
32

```

CSS

```

src > app > solicitacoes > solicitacoes.page.scss > ion-content
1 ion-content {
2   --background: #f1e2d;
3 }
4
5 ion-toolbar {
6   --background: transparent;
7 }
8
9 .titulo {
10  background-color: #dabb54;
11  color: #000000;
12  font-weight: bold;
13  padding: 10px 20px;
14  border-radius: 20px;
15  display: inline-block;
16 }
17
18 .botao-sair {
19  background-color: transparent; /* Removemos a cor de fundo do botão */
20  border: none; /* Remove borda padrão */
21  border-radius: 50%; /* Borda arredondada */
22 }
23
24 .icone-sair {
25  color: #dabb54; /* Cor do ícone de sair */
26 }
27
28 ion-item {
29  background-color: #26263a;
30  border-radius: 12px;
31  margin: 10px;
32  padding: 10px;
33  --color: #fff;
34 }
35

```

```

APPTORNEIOSGXPORTS
├── src
│   ├── app
│   │   ├── solicitacoes
│   │   │   ├── solicitacoes-routing.module.ts U 36
│   │   │   ├── solicitacoes.module.ts U 37
│   │   │   ├── solicitacoes.page.html U 38
│   │   │   ├── solicitacoes.page.scss U 39
│   │   │   ├── solicitacoes.page.spec.ts U 40
│   │   │   ├── solicitacoes.page.ts U 41
│   │   │   ├── tab1 U 42
│   │   │   ├── tab2 U 43
│   │   │   ├── tab3 U 44
│   │   │   ├── tabs U 45
│   │   │   ├── testetorneio U 46
│   │   │   ├── torneio-mata-mata U 47
│   │   │   ├── torneio-pontos-corridos U 48
│   │   │   ├── app-routing.module.ts M 49
│   │   │   ├── app.component.html U 50
│   │   │   ├── app.component.scss U 51
│   │   │   ├── app.component.spec.ts U 52
│   │   │   ├── app.component.ts U 53
│   │   │   ├── app.module.ts M 54
│   │   │   ├── assets U 55
│   │   │   ├── environments U 56
│   │   │   ├── theme U 57
│   │   │   ├── global.scss U 58
│   │   │   ├── index.html U 59
│   │   │   ├── main.ts U 60
│   │   │   ├── polyfills.ts U 61
│   │   │   └── test.ts U 62
├── ...
└── ...

```

```

src > app > solicitacoes > solicitacoes.page.scss > ion-item
36 ion-avatar {
37   width: 60px;
38   height: 60px;
39   background-color: #ffff;
40   border-radius: 50%;
41   padding: 5px;
42
43   img {
44     width: 100%;
45     height: 100%;
46     object-fit: cover; /* Garante que a imagem preencha o avatar */
47   }
48 }
49
50 ion-button {
51   border-radius: 50%;
52   width: 45px;
53   height: 45px;
54   margin: 0 5px;
55 }
56
57 ion-button[color="success"] {
58   --background: #4CAF50;
59 }
60
61 ion-button[color="danger"] {
62   --background: #F44336;
63 }

```

Typescript

```

APPTORNEIOSGXPORTS
├── src
│   ├── app
│   │   ├── solicitacoes
│   │   │   ├── solicitacoes-routing.module.ts U 1
│   │   │   ├── solicitacoes.module.ts U 2
│   │   │   ├── solicitacoes.page.html U 3
│   │   │   ├── solicitacoes.page.scss U 4
│   │   │   ├── solicitacoes.page.spec.ts U 5
│   │   │   ├── solicitacoes.page.ts U 6
│   │   │   ├── tab1 U 7
│   │   │   ├── tab2 U 8
│   │   │   ├── tab3 U 9
│   │   │   ├── tabs U 10
│   │   │   ├── testetorneio U 11
│   │   │   ├── torneio-mata-mata U 12
│   │   │   ├── torneio-pontos-corridos U 13
│   │   │   ├── app-routing.module.ts M 14
│   │   │   ├── app.component.html U 15
│   │   │   ├── app.component.scss U 16
│   │   │   ├── app.component.spec.ts U 17
│   │   │   ├── app.component.ts U 18
│   │   │   ├── app.module.ts M 19
│   │   │   ├── assets U 20
│   │   │   ├── environments U 21
│   │   │   ├── theme U 22
│   │   │   ├── global.scss U 23
│   │   │   ├── index.html U 24
│   │   │   ├── main.ts U 25
│   │   │   ├── polyfills.ts U 26
│   │   │   └── test.ts U 27
├── ...
└── ...

```

```

src > app > solicitacoes > TS solicitacoes.page.ts > SolicitacoesPage
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-solicitacoes',
5   templateUrl: './solicitacoes.page.html',
6   styleUrls: ['./solicitacoes.page.scss'],
7 })
8 export class SolicitacoesPage {
9
10   solicitacoes = [
11     {
12       id: 1,
13       nome: 'Solicitação 1',
14       imagem: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANM9GobhuwTP1225h30yY1130k2P_VY1HrxcwLFIW6AS3CPI1TSVFrFzR14V1VtasMh0071k'
15     },
16     {
17       id: 2,
18       nome: 'Solicitação 2',
19       imagem: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANM9ScQwM726Fsrcw0P02nBlux051ky6dVA6s0A9HeBRT05eFFkoV13E0_V3u4eY188N6'
20     },
21     {
22       id: 3,
23       nome: 'Solicitação 3',
24       imagem: 'https://img.mylove.com.br/fotomurais/vektor-futebol-tim-futebol-clube-icone-400-114633272.jpg'
25     },
26     {
27       id: 4,
28       nome: 'Solicitação 4',
29       imagem: 'https://encrypted-tbn0.gstatic.com/images?q=tbn:ANM9GcW0G2ctUeM1J9EzAf7eK0uKs066bz_C0U7B9F0k0A0L9G1M60A06eW0TPK1f1ctG'
30     }
31   ];
32 }

```

```

src > app > solicitacoes > TS solicitacoes.page.ts > SolicitacoesPage
33 export class SolicitacoesPage {
34
35   constructor() {}
36
37   aprovarSolicitacao(solicitacao: any) {
38     console.log('solicitação aprovada:', solicitacao.nome);
39     // Lógica para aprovar solicitação
40   }
41
42   rejeitarSolicitacao(solicitacao: any) {
43     console.log('solicitação rejeitada:', solicitacao.nome);
44     // Lógica para rejeitar solicitação
45   }
46 }

```

Model- Jogador

```

APPTORNEIOSGXPORTS
├── node_modules
├── src
│   ├── app
│   │   ├── explore-container
│   │   ├── home
│   │   ├── login
│   │   ├── loginreset
│   │   └── models
│   │       └── TS jogador.ts
└── TS jogador.ts

```

```

src > app > models > TS jogador.ts > Jogador
1 export class Jogador {
2     id= '';
3     nome= '';
4     posicao='';
5 }

```

Jogadores page

Cadastrar-jogadores page

HTML

```

APPTORNEIOSGXPORTS
├── node_modules
├── src
│   ├── app
│   │   ├── explore-container
│   │   ├── home
│   │   ├── login
│   │   ├── loginreset
│   │   ├── models
│   │   └── pags
│   │       ├── alterar-jogadores
│   │       ├── cadastro-jogadores
│   │       ├── TS cadastro-jogadores-routi...
│   │       ├── TS cadastro-jogadores.modu...
│   │       ├── cadastro-jogadores.page...
│   │       ├── cadastro-jogadores.page...
│   │       ├── TS cadastro-jogadores.page...
│   │       ├── TS cadastro-jogadores.page.ts
│   │       └── listar-jogadores
└── TS cadastro-jogadores.page...

```

```

src > app > pags > cadastro-jogadores > cadastro-jogadores.page.html > ion-content > h1.ion-text-center
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title>Jogadores</ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content >
8   <h1 class="ion-text-center">Cadastro</h1>
9
10  <ion-item lines="full">
11    <ion-label position="floating">Nome</ion-label>
12    <ion-input [(ngModel)]="jogador.nome"></ion-input>
13  </ion-item>
14  <ion-item lines="full">
15    <ion-label position="floating">Posicao</ion-label>
16    <ion-input [(ngModel)]="jogador.posicao"></ion-input>
17  </ion-item>
18
19  <ion-button expand="block" (click)="salvar()">SALVAR</ion-button>
20  <!--<ion-button expand="block" (click)="deletar()">DELETAR</ion-button-->
21  <ion-button expand="block" fill="outline" routerLink="/">Voltar</ion-button>

```

CSS

```

APPTORNEIOSGXPORTS
├── node_modules
├── src
│   ├── app
│   │   ├── explore-container
│   │   ├── home
│   │   ├── login
│   │   ├── loginreset
│   │   ├── models
│   │   └── pags
│   │       ├── alterar-jogadores
│   │       ├── cadastro-jogadores
│   │       ├── TS cadastro-jogadores-routi...
│   │       ├── TS cadastro-jogadores.modu...
│   │       ├── cadastro-jogadores.page...
│   │       ├── cadastro-jogadores.page...
│   │       ├── TS cadastro-jogadores.page...
│   │       ├── TS cadastro-jogadores.page.ts
│   │       └── listar-jogadores
└── TS cadastro-jogadores.page...

```

```

src > app > pags > cadastro-jogadores > cadastro-jogadores.page.scss > .ion-text-center
1
2 .ion-text-center{
3   color: #eeba2b;
4   font-style: oblique;
5   font-weight: bolder;
6 }

```

Typescript

```

APPTORNEIOSEXPORTS
├── node_modules
├── src
│   ├── app
│   │   ├── explore-container
│   │   ├── home
│   │   ├── login
│   │   ├── loginreset
│   │   ├── models
│   │   └── pags
│   │       ├── alterar-jogadores
│   │       ├── cadastro-jogadores
│   │       │   ├── cadastro-jogadores-routing.module.ts
│   │       │   ├── cadastro-jogadores.module.ts
│   │       │   ├── cadastro-jogadores.page.html
│   │       │   ├── cadastro-jogadores.page.scss
│   │       │   ├── cadastro-jogadores.page.spec.ts
│   │       │   └── cadastro-jogadores.page.ts
│   │       ├── listar-jogadores
│   │       ├── perfil
│   │       ├── reset
│   │       ├── Services
│   │       ├── signup
│   │       ├── solidtacoes
│   │       ├── tab1
│   │       ├── tab2
│   │       ├── tab3
│   │       ├── tabs
│   │       ├── testetorneio
│   │       ├── torneio-mata-mata
│   │       └── torneio-pontos-corridos
└── ...

```

```

src > app > pags > cadastro-jogadores > TS cadastro-jogadores.page.ts > ...
1  import { Component, OnInit } from '@angular/core';
2
3  import { JogadorService } from 'src/app/Services/jogador.service';
4  import { Jogador } from 'src/app/models/jogador';
5  import { Router, Route } from '@angular/router';
6
7  @Component({
8      selector: 'app-cadastro-jogadores',
9      templateUrl: './cadastro-jogadores.page.html',
10     styleUrls: ['./cadastro-jogadores.page.scss'],
11 })
12 export class CadastroJogadoresPage implements OnInit {
13
14     jogador: Jogador = new Jogador();
15     constructor(
16         private jogadorService: JogadorService,
17         private route: Router
18     ) {}
19
20     ngOnInit() {
21     }
22
23     salvar(){
24         this.jogadorService.salvar(this.jogador);
25         this.jogador.nome= '',this.jogador.posicao='';
26         this.route.navigate(['/listar-jogadores']);
27     }
28
29 }
30
31

```

Alterar-jogadores page

HTML

```

APPTORNEIOSEXPORTS
├── node_modules
├── src
│   ├── app
│   │   ├── explore-container
│   │   ├── home
│   │   ├── login
│   │   ├── loginreset
│   │   ├── models
│   │   └── pags
│   │       ├── alterar-jogadores
│   │       ├── cadastro-jogadores
│   │       │   ├── cadastro-jogadores-routing.module.ts
│   │       │   ├── cadastro-jogadores.module.ts
│   │       │   ├── cadastro-jogadores.page.html
│   │       │   ├── cadastro-jogadores.page.scss
│   │       │   ├── cadastro-jogadores.page.spec.ts
│   │       │   └── cadastro-jogadores.page.ts
│   │       ├── listar-jogadores
│   │       ├── perfil
│   │       ├── reset
│   │       ├── Services
│   │       ├── signup
│   │       ├── solidtacoes
│   │       ├── tab1
│   │       ├── tab2
│   │       ├── tab3
│   │       ├── tabs
│   │       ├── testetorneio
│   │       ├── torneio-mata-mata
│   │       └── torneio-pontos-corridos
└── ...

```

```

src > app > pags > alterar-jogadores > alterar-jogadores.page.html > ion-content > ion-button
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title>alterar-jogador</ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <div class="ion-text-center">Alterar</div>
9
10  <ion-item lines="full">
11    <ion-label position="floating">Nome</ion-label>
12    <ion-input [(ngModel)]="jogador.nome"></ion-input>
13  </ion-item>
14
15  <ion-item lines="full">
16    <ion-label position="floating">Posicao</ion-label>
17    <ion-input [(ngModel)]="jogador.posicao"></ion-input>
18  </ion-item>
19  <div>
20    <ion-button expand="block" (click)="salvar()">Alterar</ion-button>
21    <ion-button expand="block" fill="outline" routerLink="/cadastro-jogadores">Voltar</ion-button>
22  </div>

```

Typescript

```

APPTORNEIOSEXPORTS src > app > pags > alterar-jogadores > alterar-jogadores.page.ts > AlterarJogadoresPage > constructor
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  > loginreset
  > models
  > pags
  > alterar-jogadores
  TS alterar-jogadores-routing... U
  TS alterar-jogadores.module.ts U
  alterar-jogadores.page.html U
  alterar-jogadores.page.scss U
  TS alterar-jogadores.page.sp... U
  TS alterar-jogadores.page.ts U
  > cadastro-jogadores
  > listar-jogadores
  > perfil
  > reset
  > Services
  > signup
  > solicitacoes
  > tab1
  > tab2
  > tab3
  > tabs
  > testetorneio
  > torneio-mata-mata
  > torneio-pontos-corridos
  > OUTLINE
  > TIMELINE

11  })
12  export class AlterarJogadoresPage implements OnInit {
13
14  //instancia uma classe para o Jogador
15  jogador:Jogador = new Jogador();
16
17  constructor(
18    private jogadorService: JogadorService,
19    private route:Router,
20    private rotaAtiva: ActivatedRoute
21  ) {
22
23    // Pega o ID que veio pela rota e salva dentro da propriedade id do produto
24    this.jogador.id = this.rotaAtiva.snapshot.params['id'];
25  }
26
27  ngOnInit() {
28    //Busca os dados dos atributos do jogador com o id que retornou da rota
29    this.jogadorService.buscarporId(this.jogador.id).subscribe((dados:any)->{
30      this.jogador.nome = dados['nome'];
31      this.jogador.posicao = dados['posicao'];
32    });
33  }
34
35  // metodo que pega os dados do jogador para realizar alteracao
36  salvar(){
37    // Envia os dados do jogador para alteraçao
38    this.jogadorService.alterar(this.jogador);
39    this.route.navigateByUrl('/')
40  }
41
42
43
44

```

Listar-jogadores page

HTML

```

APPTORNEIOSEXPORTS src > app > pags > listar-jogadores > listar-jogadores.page.html > ion-content > ion-fab > ion-fab-button
  > node_modules
  > src
  > app
  > explore-container
  > home
  > login
  > loginreset
  > models
  > pags
  > alterar-jogadores
  > cadastro-jogadores
  > listar-jogadores
  TS listar-jogadores-routing... U
  TS listar-jogadores.module.ts U
  listar-jogadores.page.html U
  listar-jogadores.page.scss U
  TS listar-jogadores.page.spec... U
  TS listar-jogadores.page.ts U
  > perfil
  > reset
  > Services
  > signup
  > solicitacoes
  > tab1
  > tab2
  > tab3
  > tabs
  > testetorneio
  > torneio-mata-mata
  > torneio-pontos-corridos
  > OUTLINE
  > TIMELINE

9  <ion-content [fullscreen]="true">
10
11
12
13  <ion-grid>
14    <ion-row>
15      <ion-col>Nome dos Jogadores</ion-col>
16      <ion-col>Posição de Jogo</ion-col>
17    </ion-row>
18  </ion-grid>
19  <ion-item-sliding *ngfor="let jogadores of listarJogadores">
20    <ion-item lines="full">
21      <ion-grid>
22        <ion-row>
23          <ion-col>{{jogadores.nome}}</ion-col>
24          <ion-col>{{jogadores.posicao}}</ion-col>
25        </ion-row>
26      </ion-grid>
27      <!-- <ion-label>{{jogadores.nome}}</ion-label>
28      <ion-label>{{jogadores.posicao}}</ion-label -->
29    </ion-item>
30    <ion-item-options side="end">
31      <ion-item-option color="warning" routerLink= "/alterar-jogadores:{{jogadores.id}}">
32        <!-- "/pags/alterar-jogadores/{{jogadores.id}} -->
33        <ion-icon slot="icon-only" name="create-outline"></ion-icon>
34      </ion-item-option>
35      <ion-item-option color="danger" (click)="deletarJogador(jogadores.id)">
36        <ion-icon slot="icon-only" name="trash-outline"></ion-icon>
37      </ion-item-option>
38    </ion-item-options>
39  </ion-item-sliding>
40
41  <ion-fab vertical="bottom" horizontal="end" slot="fixed">
42    <ion-fab-button routerLink= "/cadastro-jogadores">
43      <ion-icon name="add"></ion-icon>
44    </ion-fab-button>
45  </ion-fab>
46 </ion-content>
47

```

Typescript

```

src > app > pags > listar-jogadores > TS listar-jogadores.page.ts > ListarJogadoresPage
1 import { Component, OnInit } from '@angular/core';
2 import { JogadorService } from 'src/app/Services/jogador.service';
3 import { Jogador } from 'src/app/models/jogador';
4 import { AlertController } from '@ionic/angular';
5
6 @Component({
7   selector: 'app-listar-jogadores',
8   templateUrl: './listar-jogadores.page.html',
9   styleUrls: ['./listar-jogadores.page.scss'],
10 })
11 export class ListarJogadoresPage implements OnInit {
12
13   public listarJogadores: Jogador[] = [];
14   constructor(private JogadorService: JogadorService,
15     private alertController: AlertController) {}
16
17   ngOnInit() {
18
19     this.buscarJogadores();
20   }
21
22   buscarJogadores(){
23     this.JogadorService.buscarJogadores().subscribe(dadosRetorno =>{
24       this.listarJogadores = dadosRetorno.map((registro:any)=>{
25         {
26           id: registro.payload.doc.id,
27           nome:registro.payload.doc.data()['nome'],
28           posicao:registro.payload.doc.data()['posicao'],
29         }
30       });
31     });
32   }
33 }

```

```

src > app > pags > listar-jogadores > TS listar-jogadores.page.ts > ListarJogadoresPage > ngOnInit
11 export class ListarJogadoresPage implements OnInit {
12   buscarJogadores(){
13     ""
14   }
15
16   async deletarJogador(id:string){
17     const alert = await this.alertController.create({
18       header: 'confirma exclusao desse jogador?',
19       buttons: [
20         {
21           text:'Mao',
22           role:'cancel',
23           handler:()->{
24
25         },
26       ],
27         {
28           text:'Sim',
29           role:'confirm',
30           handler:()->{
31             this.JogadorService.deletar(id);
32           },
33       },
34     ]);
35     await alert.present();
36     this.buscarJogadores();
37   }
38 }
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61

```

Jogador.service

```

APPTORNEIOSGXPORTS
├── src
│   ├── app
│   │   ├── pags
│   │   │   ├── listar-jogadores
│   │   │   ├── perfil
│   │   │   ├── reset
│   │   │   └── Services
│   │   │       ├── autenticao.service.spec.ts U
│   │   │       ├── autenticao.service.ts U
│   │   │       ├── jogador.service.spec.ts U
│   │   │       └── jogador.service.ts U
│   │   ├── signup
│   │   ├── solicitacoes
│   │   ├── tab1
│   │   ├── tab2
│   │   ├── tab3
│   │   ├── tabs
│   │   ├── testetorneio
│   │   ├── torneio-mata-mata
│   │   ├── torneio-pontos-corridos
│   │   ├── app-routing.module.ts M
│   │   ├── app.component.html
│   │   ├── app.component.scss
│   │   ├── app.component.spec.ts
│   │   ├── app.component.ts
│   │   ├── app.module.ts M
│   │   ├── assets
│   │   ├── environments
│   │   └── theme
└── src > app > Services > TS jogador.service.ts > JogadorService
    1  import { Injectable } from '@angular/core';
    2
    3  //Importa o Firestore
    4  import { AngularFireStore } from '@angular/fire/compat/firestore'
    5  //Importa a classe Jogador/ Tab Jogador
    6  import { Jogador } from 'src/app/models/jogador';
    7
    8  @Injectable({
    9  |   providedIn: 'root'
   10 | })
   11  export class JogadorService {
   12
   13  //instancia na variavel cap de captura o Firestore
   14  constructor(private cap:AngularFirestore) { }
   15
   16  //metodo salvar novos jogadores
   17  salvar(jogador: Jogador){
   18  |   |   //collection é como se fosse a tabela jogador no Firestore
   19  |   |   return this.cap.collection('jogador').add({...jogador});
   20  |   }
   21  }
   22
   23  //listar os jogadores salvos
   24  buscarJogadores(){
   25
   26  |   |   return this.cap.collection('jogador').snapshotChanges();
   27  |   }
   28  }
   29  // buscar um jogador pelo ID
   30  buscarPorId(id: string){
   31
   32  |   |   return this.cap.collection('jogador').doc(id).valueChanges();
   33  |   }
   34  }
  
```

```

APPTORNEIOSGXPORTS
├── src
│   ├── app
│   │   ├── pags
│   │   │   ├── listar-jogadores
│   │   │   ├── perfil
│   │   │   ├── reset
│   │   │   └── Services
│   │   │       ├── autenticao.service.spec.ts U
│   │   │       ├── autenticao.service.ts U
│   │   │       ├── jogador.service.spec.ts U
│   │   │       └── jogador.service.ts U
│   │   ├── signup
│   │   ├── solicitacoes
│   │   ├── tab1
│   │   ├── tab2
│   │   ├── tab3
│   │   ├── tabs
│   │   ├── testetorneio
│   │   ├── torneio-mata-mata
│   │   ├── torneio-pontos-corridos
│   │   ├── app-routing.module.ts M
│   │   ├── app.component.html
│   │   ├── app.component.scss
│   │   ├── app.component.spec.ts
│   │   ├── app.component.ts
│   │   ├── app.module.ts M
│   │   ├── assets
│   │   ├── environments
│   │   └── theme
└── src > app > Services > TS jogador.service.ts > JogadorService > salvar
    11  export class JogadorService {
    30  buscarPorId(id: string){
    34  }
    35  // altera um registro no banco
    36  alterar(jogador: Jogador){
    37
    38  |   |   return this.cap.collection('jogador').doc(jogador.id).update({...jogador});
    39  |   }
    40  }
    41
    42  // deleta um registro no banco por id
    43  deletar(id:string){
    44  |   |   return this.cap.doc('jogador/' + id).delete();
    45  |   }
    46  }
    47
    48
    49
  
```

Torneios

Tab2

Html

```

EXPLORER
  APPTORNEIOSGXPORTS
    src
      app
        tab2
          tab2-routing.module.ts
          tab2.module.ts
          tab2.page.html
          tab2.page.scss
          tab2.page.spec.ts
          tab2.page.ts
        tab3
        tabs
        testetorneio
        torneio-mata-mata
        torneio-pontos-corridos
        app-routing.module.ts
        app.component.html
        app.component.scss
        app.component.spec.ts
        app.component.ts
        app.module.ts
        assets
        environments
        theme
        global.scss
        index.html
        main.ts
        polyfills.ts
        test.ts
        zone-flags.ts
        browserlistrc
    OUTLINE
    TIMELINE

src > app > tab2 > tab2.page.html > ion-content > div.top-buttons-container > div.meus-torneios > ion-icon.trophy-icon
1 <ion-content>
2
3
4 <div class="top-buttons-container">
5   <!-- Botão de voltar -->
6   <ion-button fill="clear" class="back-button" (click)="goBack()">
7     <ion-icon name="arrow-back-outline"></ion-icon>
8   </ion-button>
9
10  <!-- Div para Meus Torneios -->
11  <div class="meus-torneios" (click)="openMyTournaments()">
12    Meus torneios
13    <ion-icon name="trophy-outline" class="trophy-icon"></ion-icon>
14  </div>
15
16  <div class="torneios-container">
17    <!-- Primeiro Card -->
18    <div class="tournament-card">
19      <div class="overlay">
20        <h2 class="tournament-title">Campeonato Municipal de Futebol Amador 2024</h2>
21        <ion-button class="calendar-button" fill="solid" (click)="openCalendar()">
22          Calendário
23        </ion-button>
24      </div>
25    </div>
26
27    <!-- Segundo Card -->
28    <div class="tournament-card2">
29      <div class="overlay">
30        <h2 class="tournament-title">Torneio Amistoso de Verão 2024</h2>
31        <ion-button class="calendar-button" fill="solid" (click)="openCalendar()">
32          Calendário
33        </ion-button>
34      </div>
35    </div>
36
37    <!-- Terceiro Card -->
38    <div class="tournament-card3">
39      <div class="overlay">
40        <h2 class="tournament-title">Copa de Futebol Interbairros 2024</h2>
41        <ion-button class="calendar-button" fill="solid" (click)="openCalendar()">
42          Calendário
43        </ion-button>
44      </div>
45    </div>
46  </div>
47
48  <div class="div-button">
49    <ion-button class="torneios-button" fill="solid" (click)="TorneioMataMata()">
50      Torneio Mata-Mata
51    </ion-button>
52    <ion-button class="torneios-button" fill="solid" (click)="TorneioPontosCorridos()">
53      Torneio Pontos Corridos
54    </ion-button>
55    <ion-button class="torneios-button" fill="solid" (click)="TorneioAmistosoVerao()">
56      Torneio Amistoso de Verão
57    </ion-button>
58  </div>
59
60 </ion-content>
61
62
  
```

CSS

```

APPTORNEIOSGXPORTS src > app > tab2 > tab2.page.scss > ...
  1 .top-buttons-container {
  2   display: flex;
  3   justify-content: center;
  4   align-items: center;
  5   gap: 10px;
  6   padding: 30px;
  7 }
  8
  9 .meus-torneios {
 10  background-color: #ffc02d; /* Cor de fundo amarela */
 11  color: black; /* Texto preto */
 12  border-radius: 50px; /* Borda arredondada */
 13  padding: 10px 20px; /* Espaçamento interno */
 14  display: flex; /* Flexbox para alinhar ícone e texto */
 15  align-items: center; /* Centraliza verticalmente */
 16  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* Sombra para dar destaque */
 17  font-style: oblique;
 18  font-weight: bolder;
 19 }
 20
 21 .trophy-icon {
 22  margin-left: 10px; /* Espaço entre o texto e o ícone */
 23 }
 24
 25 .torneios-container {
 26  display: flex;
 27  flex-direction: column;
 28  align-items: center;
 29  justify-content: center;
 30  height: 65%;
 31  gap: 20px;
 32  position: relative;
 33 }
 34
  35 .back-button {
  36   position: absolute;
  37   top: 16px;
  38   left: 16px;
  39   z-index: 10;
  40   --ion-color-primary: #adab54;
  41 }
  42
  43 .tournament-card {
  44   position: relative;
  45   background-image: url('https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9Gc79F6E9C0W6:Fc09z2z64fM91nH7Q7w0&img=');
  46   background-size: cover;
  47   background-position: center;
  48   height: 250px;
  49   width: 90%;
  50   border-radius: 8px;
  51   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  52 }
  53
  54 .tournament-card2 {
  55   position: relative;
  56   background-image: url('https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTq6l7196t8a991vzFARqj-0rcvnl0MlqAS&img=');
  57   background-size: cover;
  58   background-position: center;
  59   height: 250px;
  60   width: 90%;
  61   border-radius: 8px;
  62   box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
  63 }
  64
  65
  66
  67
  68
  69
  70
  71
  72
  73
  74
  75
  76
  77
  78
  79
  80
  81
  82
  83
  84
  85
  86
  87
  88
  89
  90
  91
  92
  93
  94
  95
  96
  97
  98
  99
  100
  101
  102
  103
  104
  105
  106
  107
  108
  109
  110
  111
  112
  113
  114
  115
  116
  117
  118
  119
  120
  121
  122
  123
  124
  125
  126
  127
  128
  129
  130
  131
  132
  133
  134
  135
  136
  137
  138
  139
  140
  141
  142
  143
  144
  145
  146
  147
  148
  149
  150
  151
  152
  153
  154
  155
  156
  157
  158
  159
  160
  161
  162
  163
  164
  165
  166
  167
  168
  169
  170
  171
  172
  173
  174
  175
  176
  177
  178
  179
  180
  181
  182
  183
  184
  185
  186
  187
  188
  189
  190
  191
  192
  193
  194
  195
  196
  197
  198
  199
  200
  201
  202
  203
  204
  205
  206
  207
  208
  209
  210
  211
  212
  213
  214
  215
  216
  217
  218
  219
  220
  221
  222
  223
  224
  225
  226
  227
  228
  229
  230
  231
  232
  233
  234
  235
  236
  237
  238
  239
  240
  241
  242
  243
  244
  245
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  1000

```

```

src > app > tab2 > tab2.page.scss > .back-button
  65 .tournament-card3 {
  66   position: relative;
  67   background-image: url('https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GctmXSTjEiffab_0u7pq_#FA7E09R1P1G82&e');
  68   background-size: cover;
  69   background-position: center;
  70   height: 250px;
  71   width: 90%;
  72   border-radius: 8px;
  73   box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);
  74 }
  75
  76
  77 .overlay {
  78   position: absolute;
  79   top: 0;
  80   left: 0;
  81   width: 100%;
  82   height: 100%;
  83   display: flex;
  84   flex-direction: column;
  85   justify-content: space-between;
  86   padding: 16px;
  87   background-color: #rgba(0, 0, 0, 0.4);
  88   border-radius: 8px;
  89 }
  90
  91 .tournament-title {
  92   color: white;
  93   font-size: 1.5em;
  94   text-align: center;
  95   margin: 0;
  96   margin-top: 16px;
  97 }

```

```

src > app > tab2 > tab2.page.scss > .tournament-card3
  98
  99 .calendar-button {
100   align-self: flex-end;
101   --background: #fbc02d; /* Cor do botão */
102   --color: black; /* Texto preto para contraste */
103   font-style: oblique;
104   font-weight: bolder;
105 }
106
107 .div-button {
108   display: flex;
109   justify-content: center; /* Centraliza horizontalmente */
110   align-items: center; /* Centraliza verticalmente */
111   font-style: oblique;
112   font-weight: bolder;
113 }
114
115 .torneios-button {
116   margin: 0 16px; /* Adiciona margem entre os botões */
117   --background: #fbc02d; /* Cor do botão */
118   color: black; /* Texto preto para contraste */
119   padding-top: 40px;
120   font-style: oblique;
121   font-weight: bolder;
122 }
123
124

```

Typescript

```

src > app > tab2 > TS tab2.page.ts > ...
1 import { Component } from '@angular/core';
2 import { NavController } from '@ionic/angular';
3 import { Router } from '@angular/router';
4 @component({
5   selector: 'app-tab2',
6   templateUrl: 'tab2.page.html',
7   styleUrls: ['tab2.page.scss']
8 })
9 export class Tab2Page {
10
11   openMyTournaments() {
12     throw new Error('Method not implemented.');
```

```

src > app > tab2 > TS tab2.page.ts > ...
9   export class Tab2Page {
33
34     //Método para torneio Mata a Mata
35
36     TorneioPontosCorridos(){
37       this.router.navigate(['/torneio-pontos-corridos']);
38     }
39   }
40
41
42
43

```

Torneio-mata-mata page

Html

```

src > app > torneio-mata-mata > torneio-mata-mata.page.html > ion-content
1 <ion-header [translucent]="true">
2   <ion-toolbar>
3     <ion-title class="dados-posicionados">Torneio Mata-Mata</ion-title>
4   </ion-toolbar>
5 </ion-header>
6
7 <ion-content>
8   <!-- Exibe os confrontos de rodada atual -->
9   <ion-list *ngIf="!isTournamentOver()">
10    <ion-grid>
11      <h2 class="titulo">Rodada {{ currentRound }}</h2>
12      <ion-list *ngFor="let match of matches; let i = index">
13        <ion-card>
14          <ion-card-header>
15            <span-card-title class="subTitulo">{{ match.team1 }} x {{ match.team2 }}</span-card-title>
16          </ion-card-header>
17          <ion-card-content>
18            <ion-item>
19              <ion-label position="floating" class="dados-posicionados-novos">Placar: {{ match.team1 }}</ion-label>
20              <ion-input type="number" [(ngModel)]="match.score1"></ion-input>
21            </ion-item>
22            <ion-item>
23              <ion-label position="floating" class="dados-posicionados-novos">Placar: {{ match.team2 }}</ion-label>
24              <ion-input type="number" [(ngModel)]="match.score2"></ion-input>
25            </ion-item>
26            <ion-button (click)="recordMatchResult(i, match.score1, match.score2)">
27              Registrar Resultado
28            </ion-button>
29          </ion-card-content>
30        </ion-card>
31      </ion-list>
32    </ion-grid>
33  </div>
34  <ion-fab vertical="bottom" horizontal="end" class="BtnPadrao">
35    <ion-fab-button [routerLink]="['./tabs/tab2']">
36      <ion-icon name="chevron-forward" style="color: #eeba2b"></ion-icon>
37    </ion-fab-button>
38  </ion-fab>
39 </div>
40 </ion-list>
41 </div>
42
43 <ion-list>
44
45 <!-- Exibe o vencedor do torneio -->
46 <ion-list *ngIf="isTournamentOver()">
47   <h2>Campeão: {{ getWinner() }}</h2>
48   <ion-card>
49     <ion-img src="assets/img/mascote.jpg" class="imagem-responsive"></ion-img>
50   </ion-card>
51
52   <div>
53     <ion-fab vertical="bottom" horizontal="end" class="BtnPadrao">
54       <ion-fab-button [routerLink]="['./tabs/tab2']">
55         <ion-icon name="chevron-forward" style="color: #eeba2b"></ion-icon>
56       </ion-fab-button>
57     </ion-fab>
58   </div>
59 </ion-list>
60
61 </ion-list>
62
63 </ion-content>
64

```

Css

```

src > app > torneio-mata-mata > torneio-mata-mata.page.scss > .subtitulo
1  ion-card {
2    --ion-background-color: #f5f5f5;
3    --ion-border-radius: 10px;
4    margin-bottom: 10px;
5  }
6
7  ion-input {
8    --background: transparent;
9    --placeholder-color: #ccc;
10 }
11
12 .title{
13   text-align: center;
14 }
15
16 .imagem-responsive{
17   width: 500px;
18   height: 450px;
19   object-fit: cover;
20   margin: 20px auto;
21   display: block;
22 }
23
24 .dados-posicionados{
25   text-align: center;
26   font-size: 40px;
27   font-style: oblique;
28   font-weight: bolder;
29   color: #fbc02d;
30 }
31
32
33 .dados-posicionados-novos{
34   text-align: justify;
35   font-size: 25px;
36 }
37
38 .subtitulo{
39   color: #eeba2b;
40   font-style: oblique;
41   font-weight: bolder;
42 }
43
44 .titulo{
45   font-style: oblique;
46   font-weight: bolder;
47 }
48
49 #BtnPadrao{
50   margin-bottom: 120px;
51 }

```

Typescript

```

vscode > app > torneo-mata-mata > |> torneo-mata-mata/pages > -
1  import { Component, OnInit } from '@angular/core';
2
3  @Component({
4    selector: 'app-torneio-mata-mata',
5    templateUrl: './torneo-mata-mata.page.html',
6    styleUrls: ['./torneo-mata-mata.page.scss'],
7  })
8  export class TorneioMataMataPage implements OnInit {
9
10   // Lista inicial de times
11   times: string[] = [
12     'Time 1', 'Time 2', 'Time 3', 'Time 4', 'Time 5',
13     'Time 6', 'Time 7', 'Time 8'
14   ];
15
16   // confrontos sorteados
17   matches: { team1: string, team2: string, score1: number, score2: number, winner: string | null }[] = [];
18   currentRound: number = 1; // Para controlar a rodada atual
19
20   // Guardar os times da próxima fase
21   nextRoundTimes: string[] = [];
22
23   constructor() {
24     this.shuffleTimes(); // embaralha os times para o sorteio inicial
25     this.generateMatches(); // Gera os confrontos iniciais
26   }
27
28   ngOnInit() {
29
30
31

```

```

APPTORNEIOSGXPORTS
  src
    app
      tab3
      tabs
      testetorneio
      torneio-mata-mata
        torneio-mata-mata-routing... U
        torneio-mata-mata.module... U
        torneio-mata-mata.page.ht... U
        torneio-mata-mata.page.scss U
        torneio-mata-mata.page.sp... U
        torneio-mata-mata.page.ts U
      torneio-pontos-corridos
      app-routing.module.ts M
      app.component.html
      app.component.scss
      app.component.spec.ts
      app.component.ts
      app.module.ts M
      assets
      environments
      theme
      global.scss
      index.html
      main.ts
      polyfills.ts
      test.ts
      zone-flag.ts
      .browserslistrc
      .editorconfig
    OUTLINE
  src > app > torneio-mata-mata > TS torneio-mata-mata.page.ts > ...
    8 export class TorneioMataMataPage implements OnInit {
    32 // Embaralha os times para o sorteio aleatório
    33 shuffleTeams() {
    34   const shuffled = [...this.teams];
    35   for (let i = shuffled.length - 1; i > 0; i--) {
    36     const j = Math.floor(Math.random() * (i + 1));
    37     [shuffled[i], shuffled[j]] = [shuffled[j], shuffled[i]];
    38   }
    39   this.teams = shuffled;
    40 }
    41 // Gera os confrontos para a rodada inicial
    42 generateMatches() {
    43   this.matches = [];
    44   for (let i = 0; i < this.teams.length; i+=2) {
    45     this.matches.push({
    46       team1: this.teams[i],
    47       team2: this.teams[i + 1],
    48       score1: 0,
    49       score2: 0,
    50       winner: null
    51     });
    52   }
    53 }
    54 // Registra o resultado de um jogo e decide o vencedor
    55 recordMatchResult(matchIndex: number, score1: number, score2: number) {
    56   const match = this.matches[matchIndex];
    57   match.score1 = score1;
    58   match.score2 = score2;
    59   // Define o vencedor
    60   if (score1 > score2) {
    61     match.winner = match.team1;
    62   } else if (score1 < score2) {
    63     match.winner = match.team2;
    64   } else {
    65   }
    66 }
  }
  src > app > torneio-mata-mata > TS torneio-mata-mata.page.ts > TorneioMataMataPage > generateMatches
    57 recordMatchResult(matchIndex: number, score1: number, score2: number) {
    67   // Em caso de empate, você pode decidir uma regra (por exemplo, pênaltis)
    68   match.winner = Math.random() < 0.5 ? match.team1 : match.team2;
    69 }
    70 // Adiciona o vencedor à próxima fase
    71 this.nextRoundTeams.push(match.winner);
    72 // Verifica se todos os jogos da rodada foram registrados
    73 if (this.matches.every(m => m.winner !== null)) {
    74   this.advanceToNextRound();
    75 }
    76 }
    77 }
    78 // Avança para a próxima rodada
    79 advanceToNextRound() {
    80   // Atualiza a rodada atual
    81   this.currentRound++;
    82   // Se houver mais de um time, gera os confrontos da próxima rodada
    83   if (this.nextRoundTeams.length > 1) {
    84     this.teams = [...this.nextRoundTeams]; // Atualiza os times para a próxima fase
    85     this.nextRoundTeams = [];
    86     this.generateMatches(); // Gera novos confrontos
    87   } else {
    88     // Se sobrar apenas um time, ele é o vencedor do torneio
    89     this.teams = [this.nextRoundTeams[0]];
    90     this.nextRoundTeams = [];
    91   }
    92 }
    93 // Verifica se o torneio já acabou
    94 isTournamentOver() {
    95   return this.teams.length === 1;
    96 }
    97 }
    98 }
    99 }
    100 }
  OUTLINE
  TIMELINE

```

```

src > app > torneio-mata-mata > TS torneio-mata-mata.page.ts > TorneioMataMataPage > advanceToNextRound
8   export class TorneioMataMataPage implements OnInit {
98   isTournamentOver() {
100  }
101
102  // Retorna o nome do vencedor
103  getWinner() {
104    return this.teams[0];
105  }
106  }
107
108
109
110

```

Torneio-pontos-corridos page

Html

```

src > app > torneio-pontos-corridos > torneio-pontos-corridos.page.html > ion-content > ion-list > ion-grid.classificao > div
1   <ion-header>
2   <ion-toolbar>
3     <ion-title class="dados-posicionados">
4       Torneio de pontos corridos
5     </ion-title>
6   </ion-toolbar>
7   <ion-header>
8   <ion-content>
9     <ion-list>
10    <ion-grid>
11    <h2 class="titulo">Jogos do Torneio</h2>
12    <ion-list *ngFor="let match of matches; let i = index">
13      <ion-card >
14        <ion-card-header>
15          <ion-card-title><h2 class="subtitulo">{{ match.team1 }} vs {{ match.team2 }}</h2></ion-card-title>
16        </ion-card-header>
17        <ion-card-content >
18          <ion-item >
19            <ion-label position="floating" class=" dados-posicionados-novos">Placar {{ match.team1 }}</ion-label>
20            <ion-input type="number" [(ngModel)]="match.score1"></ion-input>
21          </ion-item >
22          <ion-item >
23            <ion-label position="floating" class=" dados-posicionados-novos">Placar {{ match.team2 }}</ion-label>
24            <ion-input type="number" [(ngModel)]="match.score2"></ion-input>
25          </ion-item >
26          <ion-button (click)="reconMatchResult(i, match.score1, match.score2)">
27            Registrar Resultado
28          </ion-button >
29        </ion-card-content >
30      </ion-card >
31    </ion-list >
32  </ion-grid >
33 </ion-list >

```

```

src > app > torneio-pontos-corridos > torneio-pontos-corridos.page.html > ion-content > ion-list
35 <ion-content>
36 <ion-list>
37 <ion-grid class="classificao">
38 <ion-card>
39 <ion-card-header>
40 <ion-card-title ><h2 class="dados-posicionados">Classificação: </h2></ion-card-title>
41 <ion-lng src="assets/img/mascote.jpg" class="imagem-responsive"></ion-lng >
42 </ion-card-header >
43 <ion-item *ngFor="let team of getSortedTeams()">
44 <ion-label class="dados-posicionados-novos">
45   {{ team.team }} - {{ team.score }} pontos
46 </ion-label >
47 </ion-item >
48 </ion-card >
49 <div >
50 <ion-fab vertical="bottom" horizontal="end" class="BtnPadrao">
51 <ion-fab-button [routerLink]="['../tabs/tab2']">
52 <ion-icon name="chevron-forward" style="color: #000000;"></ion-icon >
53 </ion-fab-button >
54 </div >
55 </ion-grid >
56 </ion-list >
57 </ion-content >

```

Css

```

src > app > torneio-pontos-corridos > torneio-pontos-corridos.page.scss > ...
1  ion-card {
2    --ion-background-color: #f5f5f5;
3    --ion-border-radius: 10px;
4    margin-bottom: 10px;
5  }
6
7  ion-input {
8    --background: transparent;
9    --placeholder-color: #ccc;
10 }
11
12 .title{
13   text-align: center;
14 }
15
16
17 .dados-posicionados{
18   text-align: center;
19   font-size: 40px;
20   font-style: oblique;
21   font-weight: bolder;
22   color: #fbc02d;
23 }
24
25
26 ion-img{
27   border-radius: 50px;
28 }
29
30 .imagem-responsive{
31   height: 80px;
32
33   // margin-top: 5px;
34   // margin-bottom: 5px;
35   border-radius: 50px;
36 }
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

```

src > app > torneio-pontos-corridos > torneio-pontos-corridos.page.scss > .title
37
38 .dados-posicionados-novos{
39   text-align: justify;
40   font-size: 25px;
41 }
42
43 .subtitulo{
44   color: #fbc02d;
45   font-style: oblique;
46   font-weight: bolder;
47 }
48
49 .titulo{
50   font-style: oblique;
51   font-weight: bolder;
52 }
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Typescript

```

src > app > torneio-pontos-corridos > TS torneio-pontos-corridos.page.ts > TorneioPontosCorridosPage > getSortedTeams
1 import { Component, OnInit } from '@angular/core';
2
3 @component({
4   selector: 'app-torneio-pontos-corridos',
5   templateUrl: './torneio-pontos-corridos.page.html',
6   styleUrls: ['./torneio-pontos-corridos.page.scss'],
7 })
8 export class TorneioPontosCorridosPage {
9
10  // Lista de times
11  teams: string[] = [
12    'Time 1', 'Time 2', 'Time 3', 'Time 4', 'Time 5'
13  ];
14
15  // Pontuação inicial dos times
16  scores: { [key: string]: number } = {};
17
18  // Jogos do torneio
19  matches: { team1: string, team2: string, score1: number, score2: number }[] = [];
20
21  constructor() {
22    // Inicializar pontuação de cada time
23    this.teams.forEach(team => {
24      this.scores[team] = 0;
25    });
26
27    // Gerar jogos (round-robin)
28    this.generateMatches();
29  }
30
31
32  // Gerar os jogos no formato round-robin
33  generateMatches() {
34    for (let i = 0; i < this.teams.length; i++) {
35      for (let j = i + 1; j < this.teams.length; j++) {
36        this.matches.push({
37          team1: this.teams[i],
38          team2: this.teams[j],
39          score1: 0,
40          score2: 0
41        });
42      }
43    }
44  }
45
46  // Função para registrar o resultado de um jogo
47  recordMatchResult(matchIndex: number, score1: number, score2: number) {
48    const match = this.matches[matchIndex];
49    match.score1 = score1;
50    match.score2 = score2;
51
52    // Atualizar pontuação
53    if (score1 > score2) {
54      this.scores[match.team1] += 3;
55    } else if (score1 < score2) {
56      this.scores[match.team2] += 3;
57    } else {
58      this.scores[match.team1] += 1;
59      this.scores[match.team2] += 1;
60    }
61  }
62
63  // Ordenar times pela pontuação
64  getSortedTeams() {
65    return Object.entries(this.scores)
66      .sort((a, b) => b[1] - a[1])
67      .map(([team, score]) => ({ team, score }));
68  }
69
70 }
71

```

Rotas dos Tabs1,2,3 em Tabs

```

APPTORNDOSKIPORTS
  src
    app
      tabs
        tabs-routing.module.ts
        tabs.module.ts
        tabs-page.html
        tabs-page.scss
        tabs-page.spec.ts
        tabs-page.ts
        testatomoio
        terraria-mods-mat
        terraria-pontos-corridos
        app-routing.module.ts
        app.component.html
        app.component.scss
        app.component.spec.ts
        app.component.ts
        app.module.ts
        assets
        environments
        theme
        global.scss
        index.html
        main.ts
        polyfills.ts
        test.ts
        zone-flag.ts
    OUTLINE
    TIMELINE
  src > app > tabs > tabs-routing.module.ts > tabs > routes > children
  1 import { NgModule } from '@angular/core';
  2 import { RouterModule, Routes } from '@angular/router';
  3 import { TabsPage } from './tabs.page';
  4
  5
  6
  7 const routes: Routes = [
  8   {
  9     path: 'tabs',
10    component: TabsPage,
11    children: [
12      {
13        path: 'tab1',
14        loadChildren: () => import('../login/login.module').then( m => m.LoginPageModule)
15      },
16      {
17        path: 'tab2',
18        loadChildren: () => import('../tab2/tab2.module').then( m => m.Tab2PageModule)
19      },
20      {
21        path: 'tab3',
22        loadChildren: () => import('../paga/cadastro-jogadores/cadastro-jogadores.module').then( m => m.CadastroJogadoresPageModule)
23      },
24    ],
25  },
26  {
27    path: '',
28    redirectTo: '/tabs/tab1',
29    pathMatch: 'full'
30  }
31 ],
32
33
34
35
36 path: '',
37 redirectTo: '/tabs/tab1',
38 pathMatch: 'full'
39 ]

```

CONCLUSÃO

A proposta de criação de um aplicativo para a **gestão automatizada de torneios esportivos**, tanto físicos quanto eletrônicos, apresenta-se como uma solução robusta e inovadora para um problema amplamente observado no contexto atual: a dificuldade de organizar competições de forma eficiente, acessível e sem erros. Com a crescente digitalização da sociedade e o uso massivo de dispositivos móveis, a criação de uma plataforma dedicada a pequenos e médios eventos esportivos é uma resposta prática e oportuna à necessidade de modernização do setor, além de uma oportunidade de promover a inclusão e fortalecer comunidades esportivas locais.

Ao automatizar os processos mais críticos, como a criação de tabelas, sorteio de grupos e rodadas, e o cálculo de classificações, o aplicativo elimina o desgaste e a complexidade do gerenciamento manual, que frequentemente resulta em erros e inconsistências. A possibilidade de personalização da plataforma para diferentes tipos de esportes, incluindo e-Sports, torna a solução flexível e adaptável a um público diverso, garantindo sua relevância em diferentes contextos. Com uma interface intuitiva e funcionalidades distribuídas entre uma versão gratuita e uma premium, o aplicativo consegue ser acessível a um público amplo, ao mesmo tempo que oferece recursos mais avançados para organizadores que desejam uma experiência completa e detalhada.

Além disso, a solução proposta enfatiza a **comunicação eficiente** entre os envolvidos nos torneios, algo crucial para o sucesso de qualquer competição, eliminando a fragmentação de informações e centralizando a gestão em uma única plataforma. A segurança dos dados e a confiabilidade do sistema também são garantidas, com o uso de criptografia e backups automáticos, garantindo que os organizadores e participantes possam confiar no aplicativo em todas as etapas do evento.

O impacto esperado da implementação dessa solução é significativo: torneios amadores, que antes dependiam de métodos rudimentares e desorganizados, poderão ser gerenciados de maneira profissional, promovendo a competitividade, a inclusão e o crescimento do esporte, tanto físico quanto eletrônico.

Essa transformação não só melhora a experiência dos organizadores e jogadores, mas também eleva o nível das competições, fortalecendo laços comunitários e incentivando a prática esportiva em suas diversas formas.

Em suma, a **solução proposta** representa um avanço importante na digitalização e profissionalização de torneios amadores e de e-Sports, promovendo eficiência, flexibilidade e inclusão. Com essa plataforma, espera-se que o processo de organização de competições se torne mais ágil, seguro e acessível, beneficiando tanto os esportistas quanto as comunidades envolvidas, e contribuindo para o fortalecimento do esporte como uma ferramenta de integração e desenvolvimento social.

REFERÊNCIAS

- ALMEIDA, L. **Aplicativo para sorteio de jogadores na formação de equipes em esportes coletivos**. Relatório de Estágio—UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO: [s.n.]. – Acesso em: 06/05/2024.
- BIØRN-HANSEN, A. et al. An empirical investigation of performance overhead in cross-platform mobile development frameworks. **Empirical Software Engineer**, v. 25, n. 4, p. 2997–3040, 2020. – Acesso em: 06/05/2024.
- DEDRICK, J.; KRAEMER, K. L. The impacts of IT on firm and industry structure: The personal computer industry. **California management review**, v. 47, n. 3, p. 122–142, 2005. – Acesso em: 06/05/2024.
- LAGORIO, A. et al. A systematic literature review of innovative technologies adopted in logistics management. **International Journal of Logistics Research and Applications**, v. 25, n. 7, p. 1043–1066, 2022. – Acesso em: 06/05/2024.
- LEE, S. Quantifying the benefits of smartphone adoption: Digital device substitution and digital consumption expansion. **SSRN Electronic Journal**, 2017. – Acesso em: 06/05/2024.
- LI, T. et al. Understanding the long-term evolution of mobile app usage. **IEEE transactions on mobile computing**, v. 22, n. 2, p. 1213–1230, 2023. – Acesso em: 06/05/2024.
- PARK, Y. J.; OH, Y. W. Effects of smartphones on economic and subjective quality of life. **First Monday**, 2021. – Acesso em: 06/05/2024.
- Waves of change: Business evolution through information technology. **Long range planning**, v. 29, n. 2, p. 265, 1996. – Acesso em: 06/05/2024.
- WILLOCX, M.; VOSSAERT, J.; NAESSENS, V. **Comparing performance parameters of mobile app development strategies**. Proceedings of the International Conference on Mobile Software Engineering and Systems. **Anais...**New York, NY, USA: ACM, 2016. – Acesso em: 06/05/2024.

Aplicativo para sorteio de jogadores na formação de equipes em esportes coletivos. Escrito por Luiz Alberes Bispo de Almeida, 2013 - Disponível em: https://repository.ufrpe.br/bitstream/123456789/1131/1/tcc_eso_luizalberesbispodealmeida.pdf – Acesso em: 02/10/2024.

POIT, D. R. Organização de Eventos Esportivos. 5ª ed. São Paulo: Phorte Editora, 2013. – Acesso em: 02/10/2024.

APÊNDICE A – TÍTULO

Material complementar ao texto, elaborado pelo autor