



FUTSIM

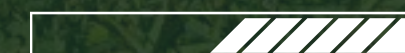
Simulador Mobile para Campeonato Futebolístico

Feito por:
Gabriel Lucas Silva Seabra,
Jussie Lopes Da Silva,
Luiz Alberto Cury Andalécio,
Vitor Gabriel Resende Lopes Oliveira

MOTIVAÇÃO

Atualmente, são poucas as soluções digitais, simples e acessíveis para a simulação personalizada de campeonatos de futebol. A maioria das ferramentas existentes, como o próprio papel, possui limitações quanto à praticidade na gerência dos torneios.

O projeto proposto busca preencher essa lacuna com um software que permita a simulação de campeonatos de futebol, permitindo ao usuário inserir seus próprios times em campeonatos de tipos variados e acompanhar resultados.



DESENVOLVIMENTO

- Plataforma-alvo: Aplicativos móveis.
- Linguagem de programação: Kotlin.
- Forma de desenvolvimento:
 - Remotamente via Discord.
 - Presencialmente no campus IFTM UPT.
- Ferramentas consideradas:
 - Visual Studio Code.
 - Android Studio.
- Método de acompanhamento do desenvolvimento:
 - Testes com usuários reais.
 - Coleta de feedbacks para ajustes iterativos.
 - Monitoramento de desempenho do software.
 - Atualizações periódicas com base nas demandas dos usuários.



IMPACTOS



Sobre os impactos, são dois tipos: social e financeiro

Socialmente, o projeto traz incentivo à gestão de campeonatos amadores, fomento ao entretenimento e interação entre amigos e comunidades esportivas.

Financeiramente, o consumidor tem a redução de trabalho e até custos para a organização dos torneios; já para os desenvolvedores, pode haver a possibilidade de monetização através de versões premium ou doações.

REQUISITOS

Brainstorm Inicial

Requisitos funcionais:

- Inserir times no simulador;
- Montar tabela;
- Automatizar a tabela a partir de resultados;
- Montar o chaveamento do torneio de forma aleatória;
- Automatizar a chave a partir de resultados;
- Alterar placares;
- Histórico de jogos;
- Mostrar próximo jogo;
- Design do aplicativo.

Requisitos não funcionais:

- Desenvolvimento para sistema android;
- Garantir que as transições sejam feitas simultaneamente com o decorrer dos jogos;
- Front-End intuitivo;
- O sistema deve garantir que somente administradores possam ter acesso a funcionalidades restritas como alteração de placares.



EXPERIÊNCIAS

EXPECTATIVAS VS. REALIDADE.

Objetivo do software:

- Proposta: Cumprida parcialmente conforme o planejado.
- Atendeu positivamente às expectativas iniciais de um software local.

Funcionalidades principais:

- Gestão básica de campeonatos nos modos:
 - Mata-mata.
 - Pontos corridos.



PROBLEMAS

Desafios enfrentados:

- Conciliar tempo:
 - Dificuldade em equilibrar o tempo entre o desenvolvimento do projeto e as demandas de outras disciplinas.
- Desafios no desenvolvimento:
 - Não houve grandes dificuldades no código em si.
- Desempenho da IDE:
 - Android Studio apresentou lentidão no processo de build e sincronização do Gradle.
 - A lentidão comprometeu a fluidez do trabalho.

CRONOGRAMA ESTIMADO VS. CRONOGRAMA EXECUTADO.

Cronograma: Atividades vs Implantação

Atividades	Implantação
Criar telas do app (UI)	Implantado
Criar funcionalidade de criar campeonato	Implantado
Criar funcionalidade de editar campeonato	Implantado
Criar funcionalidade de excluir campeonato	Implantado
Implementar persistência de dados	Implantado
Criar campeonato no formato pontos corridos	Implantado
Criar campeonato no formato mata-mata	Implantado
Criar campeonato no formato fase de grupos	Não implantado
Implementar edição para campeonato de pontos corridos	Implantado
Implementar edição para campeonato de mata-mata	Implantado
Implementar edição para campeonato de fase de grupos	Não implantado



ARQUITETURA E PRINCIPAIS COMPONENTES

O FutSim segue uma arquitetura MVVM (Model-View-ViewModel), separando claramente dados, lógica de negócio e interface.

O projeto está organizado em pacotes:

- model: modelos de dados.
- data: DAOs, repositórios e banco de dados.
- ui: telas (Compose), componentes visuais e temas.
- ui.viewmodel: ViewModel e CompositionLocal.

Foi utilizado Jetpack Compose para toda a interface, garantindo UI reativa e moderna.

Persistência de dados com Room, permitindo salvar campeonatos, times e partidas.

Navegação entre telas feita com Navigation Compose.

Models

- Campeonato: representa um campeonato, com id, nome e tipo (MATA_MATA, FASE_GRUPOS, PONTOS_CORRIDOS).
- Time: representa um time, com id, nome, id do campeonato e estatísticas (vitórias, empates, derrotas, gols).
- Partida: representa um confronto entre dois times, com gols e rodada.
- TipoCampeonato: enum para os tipos de campeonato.
- TimeTabela: modelo auxiliar para exibição de tabelas.

DAOs

- CampeonatoDao: CRUD para campeonatos.
- TimeDao: CRUD para times, busca por campeonato.
- PartidaDao: CRUD para partidas, busca por campeonato.
- Todos os DAOs usam suspend functions para integração com coroutines.

Banco de Dados

- FutSimDatabase: classe abstrata que define as entidades e DAOs.
- Usa @Database e @TypeConverters para suportar enums.
- Instanciado via Room.databaseBuilder na MainActivity.

Repositório

- FutSimRepository: centraliza o acesso aos DAOs.
- Fornece métodos suspensos para inserir, atualizar, deletar e listar campeonatos, times e partidas.
- Facilita a troca de fonte de dados e testes.

ViewModel

- FutSimViewModel: gerencia o estado da UI e executa operações assíncronas.
- Usa StateFlow para expor listas de campeonatos, times e partidas.
- Métodos para inserir, atualizar, deletar e carregar dados, sempre atualizando o estado.
- Integrado à UI via CompositionLocalProvider e collectAsState().

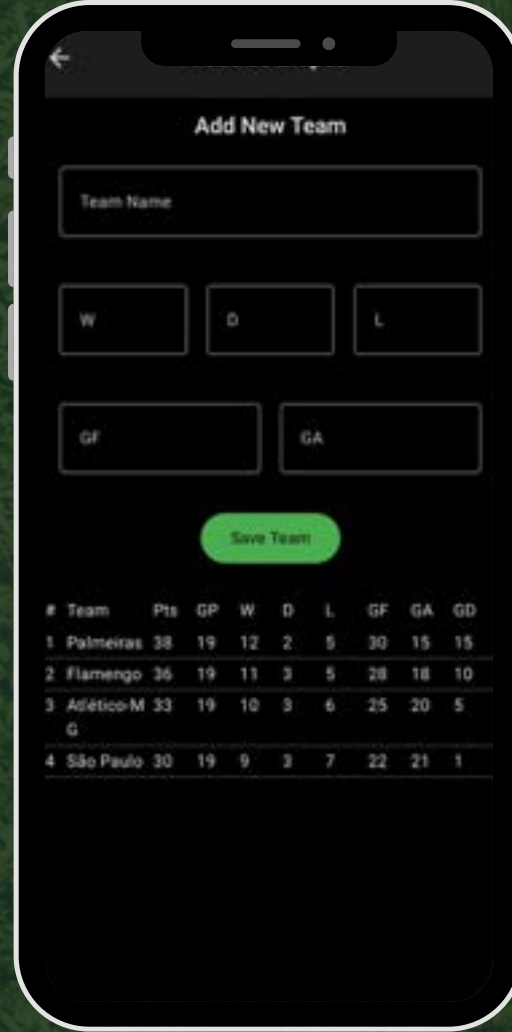
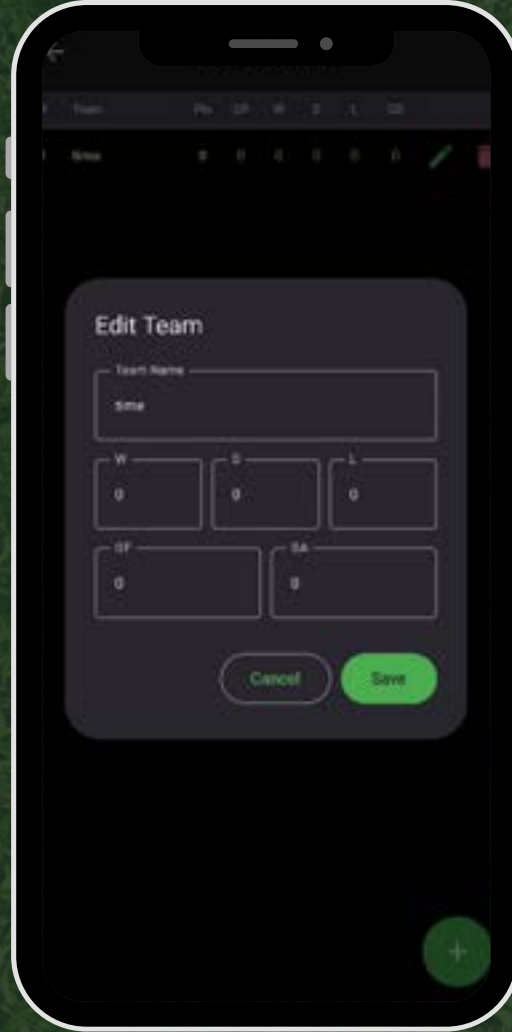
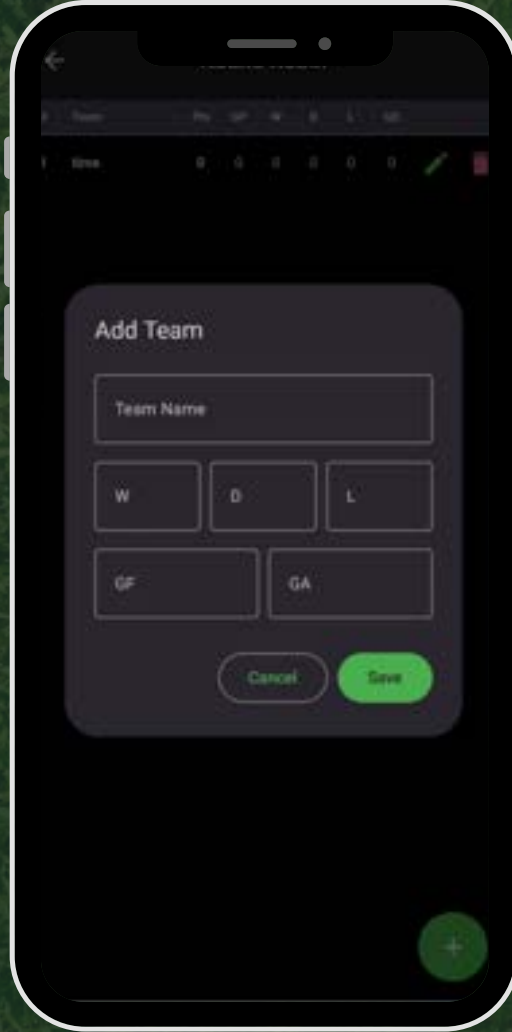
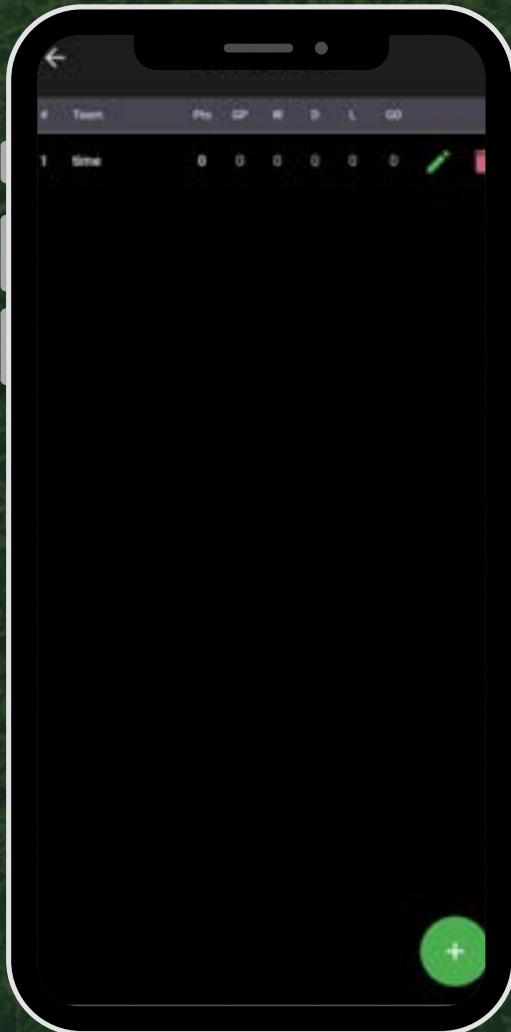
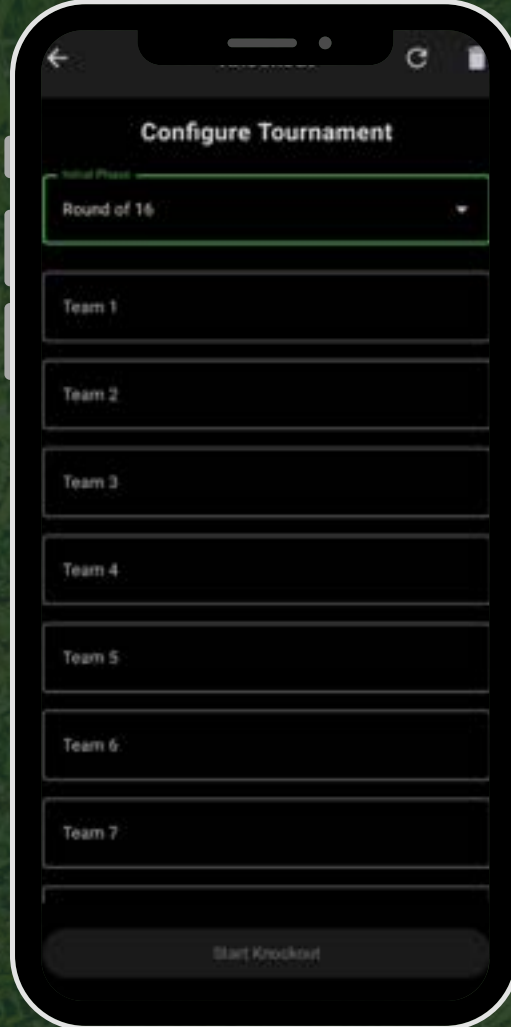
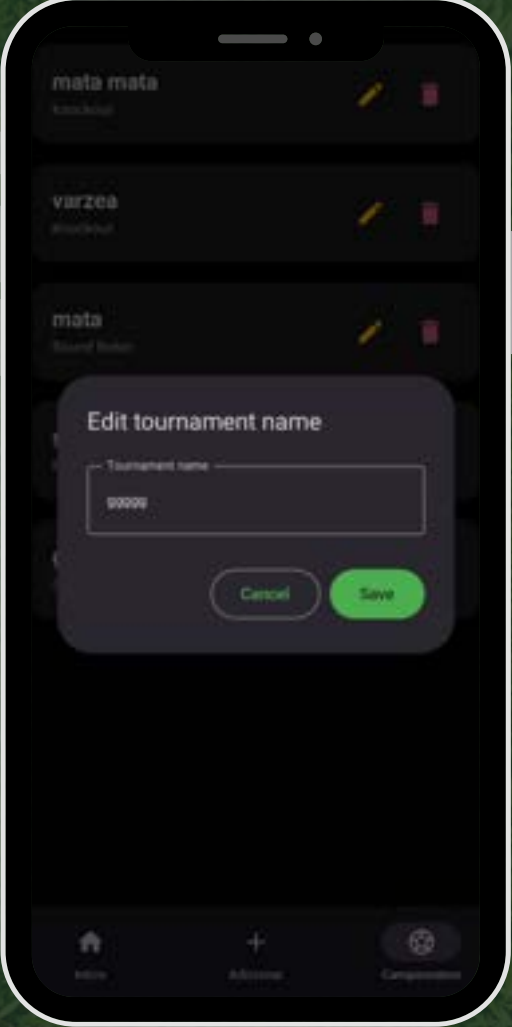
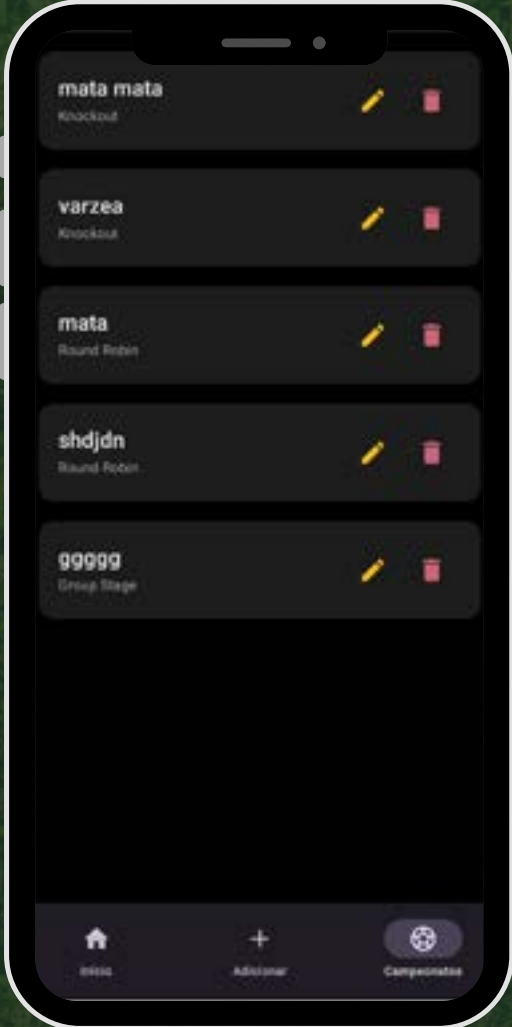
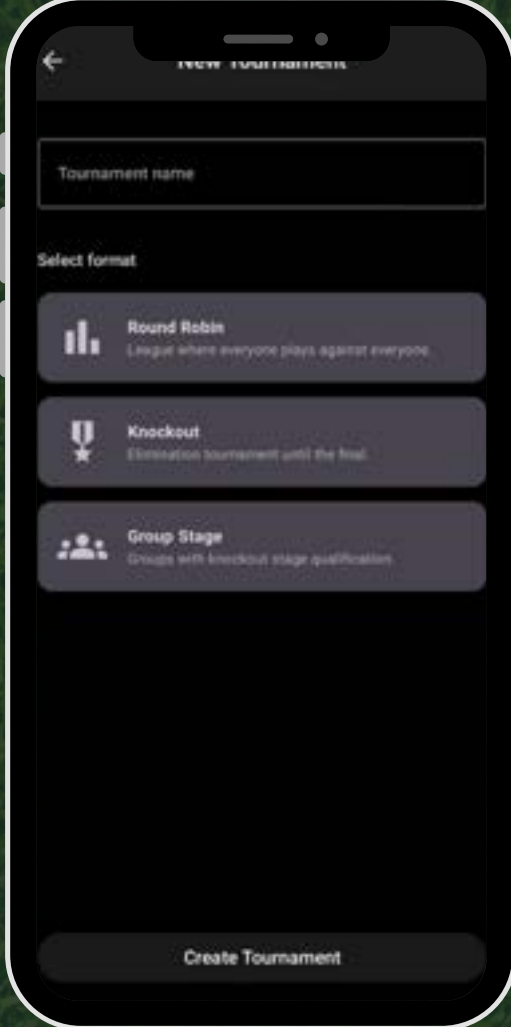
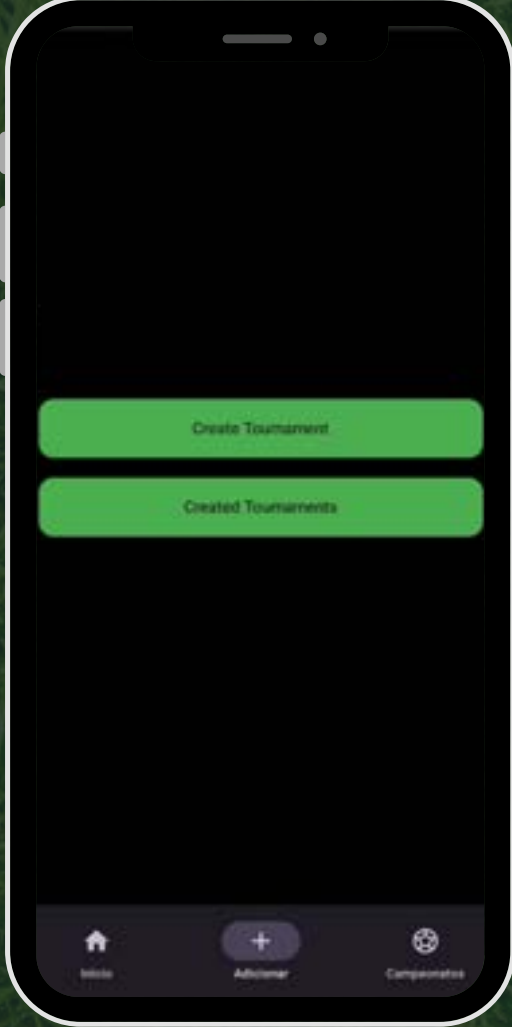
MainActivity e Navegação

- MainActivity: inicializa banco de dados, repositório e ViewModel.
- Define a navegação entre telas com NavHost e rotas nomeadas.
- Bottom navigation bar para acesso rápido às principais telas.

Componentes de UI

- ButtonUniversal: botão customizado reutilizável.
- TextFields: campos de texto para entrada de dados, com validação.
- Cards: usados para exibir campeonatos e confrontos.
- LazyColumn: listas performáticas para exibir times, campeonatos e partidas.

APLICATIVO



FUTURO DO APP...



OBRIGADO

