

# Palpiteiro, sistema gamificado de palpites esportivos<sup>1</sup>

## *Palpiteiro, gamified sports prediction system*

Guilherme Lopes Farto Fernandes<sup>2</sup>

Daniel Cardoso Moraes de Oliveira<sup>3</sup>

### Resumo

Nesse artigo será descrito um jogo *web* de palpites, baseado em pontos e não em dinheiro, no qual um mercado é aberto semanalmente com os jogos da época, neste caso jogos do campeonato brasileiro de futebol. O jogador deve escolher, para cada jogo da rodada, o resultado final, vitória do time da casa, empate ou vitória do time de fora, além de escolher um jogador para marcar um gol e um jogador para tomar cartão na partida. As pontuações atribuídas a cada resultado de cada jogo são calculadas de acordo com a colocação do time no campeonato e de acordo com os últimos cinco resultados de cada time. As pontuações dos atletas, tanto de gol como de cartão, são calculadas a partir da posição em que cada atleta joga e da recorrência de gols feitos e cartões tomados. Quanto mais alta a pontuação, menor é a probabilidade de ela ser ganha pelo jogador. Ao final de cada semana, o mercado é fechado e os resultados são computados, gerando um histórico e atribuindo a cada jogador a pontuação ganha na rodada. O jogo conta com sistema de ranking, por temporada e semanal, sistema de amigos, no qual o jogador adiciona outros usuários num ranking privado de amigos.

**Palavras-chaves:** Jogo web; Aplicação web; Palpites esportivos; Gamificação.

### Abstract

A sports prediction *web* game will be described in this article, the game is based on points, not on money, in which a weekly fixture is open for predictions, including matches from the brazilian soccer league. The user must choose, for each game, the final result, home, away or draw, one player to score and one player to receive a yellow or a red card. The points for home, away, draw and players are calculated based on the squad position in the league, last five results, player's position and probability to score or receive cards. Higher the points, less probable to happen.

<sup>1</sup> Trabalho de conclusão de curso apresentado ao curso de Bacharelado em Sistemas de Informação da Universidade Federal Fluminense como requisito parcial para conclusão do curso.

<sup>2</sup> Graduando do Curso de Sistemas de Informação - UFF, guilhermelff@id.uff.br.

<sup>3</sup> Orientador - Instituto de Computação - UFF, danielcmo@ic.uff.br.

By the end of each week, the fixture is closed and the predictions resolved, generating a record and adding the corresponding points for each user. The game comes with a ranking system, both for season and week, a friends system, in which the users add friends to a private rank.

**Keywords:** Web game; Web application; Sports prediction; Gamification.

**Aprovado em:** 12/12/2023. **Versão Final em:** 12/12/2023

## 1. INTRODUÇÃO

Atualmente existem diversos sites de apostas esportivas tradicionais, nos quais é implementado um sistema de pagamentos em dinheiro, sendo oferecido uma grande quantidade de esportes e mercados. Grande parte do público vê esses sites como diversão, outros como trabalho e alguns como puramente uma chance de ganhar dinheiro. O foco do jogo apresentado neste artigo é voltado para aqueles que veem como diversão, já que o processo de palpites é muito parecido com um site tradicional, sendo diferente na questão do dinheiro, que não é necessário para jogar.

O jogo é composto de um sistema de pontos, no qual cada jogo recebe três pontuações, vitória do time da casa, empate e vitória do time de fora. As pontuações são calculadas automaticamente de acordo com a posição de cada time na tabela e de acordo com os últimos cinco resultados da equipe. No começo do campeonato é montado um *power ranking* (WIKIPEDIA, 2023), tabela feita de acordo com os elencos dos times, já que não existe histórico dentro da competição. Para cada atleta inserido no jogo, é lhe atribuído uma pontuação em caso de gol feito ou cartão tomado, essas pontuações são calculadas de acordo com o valor esperado para cada evento. O valor esperado é calculado por bases de dados externas, que geram este valor de acordo com todos os chutes dados por certo jogador, no caso da pontuação para gol marcado, a partir da posição, ângulo, tipo de chute, distância do alvo e como o lance foi gerado, por um escanteio ou bola enfiada, por exemplo. Já o valor para cada atleta tomar cartão no jogo é calculado pelo histórico de cartões de cada atleta.

O jogo consiste de um mercado, que é aberto toda quarta-feira, no qual os jogos da rodada do brasileirão são adicionados para que os usuários possam fazer seus palpites. As apostas são adicionadas dinamicamente a uma ficha de apostas, que ao final de todas as seleções, permite que o jogador realize todos os palpites de uma só vez. Na quinta-feira o mercado é fechado, na terça-feira os palpites são resolvidos e as devidas pontuações atribuídas a cada jogador, mantendo um histórico de acertos e erros para consulta.

O jogo conta com um sistema de ranking, no qual todos os jogadores são ordenados, por um filtro semanal e um por temporada. Também conta com sistema

de amigos, no qual é possível enviar um convite de amizade a outro usuário que, após ser aceito, este usuário é adicionado num ranking pessoal com todos os seus amigos. A interface foi pensada para ser simples e intuitiva, e também contém uma aba Ajuda, para usuários inexperientes. O sistema foi feito em *JavaScript* (EICH, 1995), *Bootstrap* (OTTO, 2011), *CSS* (W3C, 1996), com dados sendo gravados no *Google Firebase* (TAMPLIN, 2011), e está hospedado no *Netlify* (BIILMANN, 2014), além de possuir uma *API* em *Node.js* (DAHL, 2009), que faz uma varredura nos sites do *Google* (PAGE, 1998) e do *FBref* (REFERENCE, 2000), salvando dados necessários para o funcionamento do jogo. As tecnologias foram escolhidas para prover performance e escalabilidade ao jogo.

## 2. TRABALHOS RELACIONADOS

### 2.1. CARTOLA FC

O *Cartola FC* (GLOBO, 2005), jogo *web* do Grupo Globo, é caracterizado como *fantasy sport game*, no qual atletas reais são inseridos na plataforma, e seus desempenhos na vida real refletem em pontuação no jogo. O usuário deve montar seu time com os jogadores que acha que terão bom desempenho na semana, enquanto o mercado está aberto, e ao final da semana as pontuações são computadas e atribuídas aos jogadores. O jogo conta com sistema de amigos e ligas, além de premiações para os melhores jogadores. Há também um serviço pago no qual são disponibilizadas dicas para montagem do time, além de abrir possibilidade para prêmios melhores.

### 2.2. BET365

A plataforma *bet365* (BET365, 2000), além das apostas tradicionais, também conta com um sistema de escolha de placares para jogos pré-definidos, em que o usuário deve escolher os placares, e de acordo com os acertos, ganha um prêmio em dinheiro. A interface deste sistema serviu de inspiração para o jogo.

### 2.3. GAMIFICAÇÃO

Há também um artigo sobre gamificação, *How to gamify?* (MORSCHHEUSER, 2017), que descreve com clareza as estratégias que geralmente são usadas para gamificar sistemas, estratégias essas que foram usadas também no processo de elaboração deste sistema. A gamificação tem como objetivo aumentar a motivação do usuário em relação as atividades propostas pelo sistema, e fazê-lo voltar a realizá-las. É considerado um processo constante, já que necessita de monitoramento e aperfeiçoamento, a fim de manter os usuários engajados com o jogo.

### 3. REQUISITOS

Algumas atividades foram realizadas na idealização do projeto: Identificação de possíveis necessidades do usuário e possíveis necessidades do sistema, para funcionar por completo; *Brainstorm* de ideias, a fim de escolher as que melhor se encaixavam com a proposta do jogo; Prototipação de telas e classes, a fim de visualizar um projeto inicial do jogo.

Foram levantados requisitos funcionais e não funcionais, que foram criados para que o sistema execute de acordo com o planejado. Os requisitos funcionais especificam o que o sistema faz, quais são suas funções. Os requisitos não funcionais descrevem os atributos de qualidade e limitações do sistema.

#### 3.1. REQUISITOS FUNCIONAIS

1. O jogador deve ser capaz de criar sua ficha de apostas;
2. O sistema deve apresentar os jogos da semana do brasileiro;
3. O jogador deve ser capaz de visualizar suas apostas feitas;
4. O sistema deve apresentar o ranking de jogadores;
5. O jogador deve ser capaz de adicionar amigos;
6. O sistema deve ser capaz de armazenar informações sobre palpites, usuários e amigos

#### 3.2. REQUISITOS NÃO FUNCIONAIS

1. O sistema deve mostrar os jogos da rodada em menos de 30 segundos;
2. O sistema deve ficar disponível 24 horas por dia;
3. O sistema não deve ter o seu funcionamento comprometido pelas atualizações;
4. O jogador deve ser capaz de consultar suas apostas em menos de 30 segundos

### 4. ABORDAGEM

Para implementar o jogo foram realizadas reuniões, que alinharam o que poderia ser feito no prazo de um ano, além de reuniões para aprovação da interface. Para ajudar no começo do desenvolvimento, foi criado um diagrama de classes e um diagrama de banco de dados, ambos modelos iniciais, que serviram de base e foram sendo modificados de acordo com as novas necessidades do sistema, até chegar no modelo final.

O projeto foi acompanhado pelo orientador em reuniões pontuais para acertar na arquitetura da plataforma. Houve definição de escopo, definição da arquitetura, desenvolvimento do código, testes e correções, no intervalo de um ano.

O jogo foi desenvolvido para ser usado em navegador *web*, tanto *desktop* quanto *mobile*. Houve controle de versão com *Git* (TORVALDS, 2005), hospedado no *GitHub* (WANSTRATH, 2008) em código aberto.

Para desenvolver o sistema, o design seguiu as heurísticas de Nielsen (NIELSEN, 1990) para que qualquer usuário que vá utilizar o sistema tenha facilidade em realizar as ações. Para isso, Nielsen (NIELSEN, 1990) propõe 10 (dez) heurísticas que possibilitam avaliar se o sistema projetado tem uma boa interface e uma boa experiência de uso: o usuário deve ser informado sobre o que acontece no sistema, a lógica do sistema deve ser compatível com a lógica do mundo real, deve dar liberdade ao usuário, consistente e padronizado, prevenção de erros, instruções para o bom funcionamento, flexibilidade para todos os tipos de usuários, eficiente e minimalista, ajuda aos usuários para se recuperarem de erros, informação facilmente encontrável e não extensa.

O projeto segue o padrão MVC (Model-View-Controller), criado por Trygve Reenskaug (REENSKAUG, 1979), que consiste em um modelo de dados que é acessado pelo controlador, que faz o tratamento dos dados vindos do modelo e passa para o visualizador, que por sua vez monta a interface e mostra ao usuário. O sistema conta com interfaces que fazem a comunicação entre as partes, garantindo integridade dos dados e facilitando a manutenção do código, possível troca de serviços de hospedagem e troca de tecnologias usadas. Em conjunto com o modelo de dados, foi criada uma API (Application Programming Interface) em *Node.js* (DAHL, 2009), para recuperar os dados necessários para a distribuição de pontos para cada time, além de acessar os últimos resultados de cada time e qual atleta fez gol ou tomou cartão. Os dados são extraídos do site da FBref (REFERENCE, 2000), renomado site de estatísticas esportivas, e da tabela do Brasileirão do Google (PAGE, 1998). O *scraper* funciona com um código que abre uma instância minimizada do navegador Chrome, e a partir dele acessa os sites e extrai o conteúdo *HTML* (BERNERS-LEE, 1993), trata os dados, que posteriormente são salvos num arquivo *JSON* (CROCKFORD, 2000). Este arquivo *JSON* (CROCKFORD, 2000) é lido pela plataforma e utilizado para determinar os pontos de cada time assim que um jogo é adicionado no mercado e para resolver as apostas feitas pelos jogadores. Cabe aos administradores inserir os jogos da rodada, inserir novos jogadores em cada time, abrir o mercado de apostas, fechar o mercado de apostas e acionar o comando que resolve as apostas feitas pelos jogadores. Essa decisão foi tomada devido a volatilidade das datas dos jogos, ocorrendo muitas remarcações, cancelamentos de jogos e transferências de jogadores. Um modelo automatizado destas tarefas seria muito complexo e estaria propenso a erros de datas de jogos, assim como jogadores que não atuam mais em

certo time. Em poucos cliques os administradores conseguem gerir o site de forma assertiva, o que garante simplicidade de manutenção do jogo e confiabilidade dos dados mostrados aos jogadores.

## 5. PONTUAÇÃO

As pontuações dos resultados de cada jogo são calculadas da seguinte forma:

### 5.1. POSIÇÃO NA TABELA

Cada time recebe um bônus de acordo com a sua posição na tabela, calculado de acordo com a função:

$$((totalDeTimes - posicaoTime) * 4 + totalDeTimes)$$

Essa função é inversamente proporcional à posição do time, quanto mais alto na tabela, menor será a pontuação para este time.

### 5.2. BÔNUS DE VITÓRIAS

Caso tenha três ou mais vitórias nos últimos cinco jogos:

*Mais 8 (oito) pontos*

### 5.3. PUNIÇÃO POR DERROTAS

Caso tenha quatro ou cinco derrotas nos últimos cinco jogos:

*Menos 8 (oito) pontos*

### 5.4. MANDANTE

Bônus para o time mandante:

*Mais 4 (quatro) pontos*

### 5.5. PONTUAÇÃO FINAL

A pontuação final de cada time é calculada a partir do total de pontos dos dois times dividido pelos pontos do time, multiplicado por 10 (dez):

$$((PontuaçãoTimeCasa + PontuaçãoTimeFora) / PontuaçãoTime) * 10$$

### 5.6. PONTUAÇÃO EMPATE

A pontuação do empate é calculada pela soma da pontuação dos dois times, dividido por 2 (dois):

$$(PontuaçãoFinalTimeCasa + PontuaçãoFinalTimeFora) / 2$$

## 5.7. BALANCEAMENTO

Dessa forma, as pontuações ficam balanceadas, destacando os acertos de cada jogador ao decorrer da competição.

## 6. IMPLEMENTAÇÃO

A plataforma *web* foi disponibilizada em um *host* online e gratuito, o *Netlify* (BIILMANN, 2014), disponível para todas as plataformas que tenham um *browser* com *JavaScript* (EICH, 1995). Os dados são armazenados em servidor nuvem do *Google* (PAGE, 1998), o *Firebase* (TAMPLIN, 2011), localizado em São Paulo, provendo agilidade na recuperação de dados, quando acessados do Brasil.

O *backend* foi desenvolvido em *JavaScript* (EICH, 1995) em conjunto com a API do *Firebase* (TAMPLIN, 2011), que dá suporte a autenticação e regras de segurança. O *frontend* foi desenvolvido com elementos do *framework Bootstrap* (OTTO, 2011), que propiciaram um padrão para a interface do sistema.

A hospedagem feita no *Netlify* (BIILMANN, 2014) provê agilidade ao desenvolvimento, e possui integração com o *GitHub* (WANSTRATH, 2008), onde cada versão submetida do código é automaticamente carregada no site.

O banco de dados é NoSQL, padrão criado por Carlo Strozzi (1998), foi escolhido por sua escalabilidade e mutabilidade.

## 7. ARQUITETURA

A arquitetura do sistema foi pensada de modo que fosse modularizada, com cada módulo independente, facilitando a implementação de novas funcionalidades e facilitando também a manutenção do sistema como um todo. As ações principais do sistema giram em torno dos dados coletados da *web*, via *web scraper*, e salvos num arquivo *JSON* (CROCKFORD, 2000). O sistema utiliza esse arquivo como base para fazer os cálculos e inserções.

Funções *JavaScript* (EICH, 1995) carregadas no *browser* do cliente montam os palpites e o mercado de jogos, levando performance ao jogo. Também existem funções no lado do cliente que fazem as tarefas de admin, que depois são salvas no banco de dados na nuvem.

A comunicação entre as funções *JavaScript* (EICH, 1995) e o banco de dados é feita através de uma interface personalizada, que permite que outros serviços de banco de dados sejam adicionados sem que o funcionamento da aplicação seja interrompido, permitindo assim a troca de serviços, caso um venha a falhar ou ficar caro com o tempo.

As funções, junto com o *CSS* (W3C, 1996), montam a página *HTML* (BERNERS-LEE, 1993) a ser exibida para o cliente. Todo o código do sistema está carregado no *GitHub* (WANSTRATH, 2008), e é acessado pelo servidor *web*.

O banco de dados foi montado de modo que cada documento possua uma chave estrangeira, a fim de diminuir a complexidade e o número de consultas feitas ao banco.

A Figura 1 mostra a arquitetura do sistema e como os elementos conversam entre si:

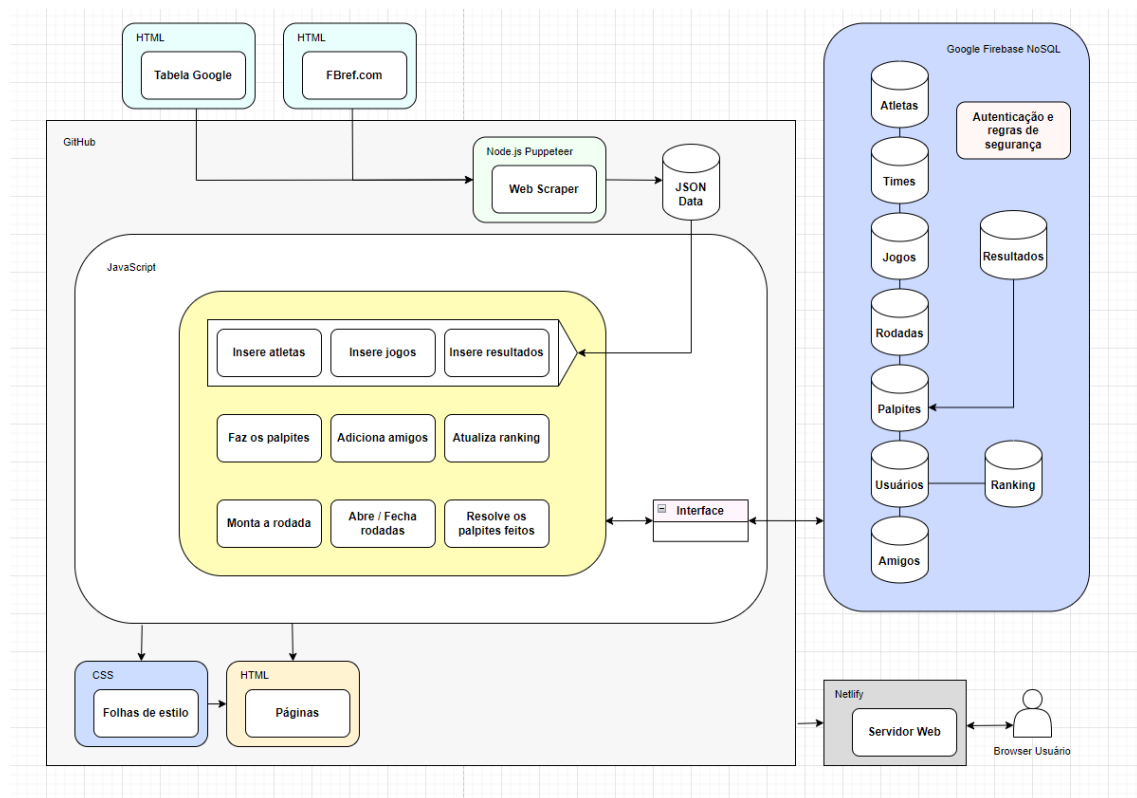


Figura 1 – Arquitetura do Sistema. Fonte: Autor.

## 8. CLASSES

### 8.1. CLASSE USUÁRIO

- ID: identificação do usuário
- email: email do usuário
- greens: acertos totais
- reds: erros totais
- imgExt: extensão do arquivo da imagem de perfil



- pontosSemana: pontos feitos na semana
- pontosTemporada: pontos feitos na temporada
- usuario: nome do usuário
- alertas: lista de notificações
- amigos: lista de amigos

## 8.2. CLASSE RANKING

- Período: ranking semanal ou de temporada
- método AtualizaRanking(): atualiza o ranking com as pontuações recentes

## 8.3. CLASSE RODADA

- Numero: número da rodada
- abertaFecha: indica se está aberta ou fechada
- método AbreRodada(): Abre a rodada para os usuários
- método FechaRodada(): Fecha a rodada para os usuários
- método ListaJogos(): lista os jogos da rodada na página inicial

## 8.4. CLASSE SCRAPER

- JSON: arquivo com os dados recuperados da web
- método BuscaTabela(): recupera a tabela do campeonato
- método BuscaUltimosResultados(): recupera os cinco últimos resultados de cada time

## 8.5. CLASSE TIME

- Nome: nome do time

## 8.6. CLASSE ATLETA

- Nome: nome do atleta
- Time: nome do time do atleta
- pontosCartao: pontos para receber cartão em um jogo
- pontosGol: pontos para fazer um gol no jogo

## 8.7. CLASSE APOSTA

- ID: identificação da aposta
- acertouErrou: indica se o usuário acertou ou errou o palpite
- casa: time da casa, do jogo correspondente a aposta
- fora: time de fora, do jogo correspondente a aposta
- jogo: jogo correspondente a aposta
- idApostador: identificação do usuário que fez a aposta
- casaEmpateFora: indicação do palpite feito para o resultado do jogo
- pontos: total de pontos a serem ganhos na aposta
- resolvida: indica se a aposta foi resolvida ou não
- resultado: indica se é uma aposta pro resultado da partida ou não
- rodada: indica a rodada em que a aposta foi feita
- cartao: indica se foi uma aposta de cartão ou não
- jogador: indica o nome do atleta da aposta

## 8.8. CLASSE JOGO

- TimeCasa: time que joga em casa
- TimeFora: time visitante
- Rodada: rodada correspondente do jogo
- Data: data da partida
- pontosCasa: pontos para o time da casa vencer
- pontosFora: pontos para o time de fora ganhar
- pontosEmpate: pontos para a partida acabar empatada

## 8.9. CLASSE ADMIN

- método insereJogo(): insere partida na rodada
- método insereAtleta(): insere atleta em um time

## 8.10. SERVIÇO - CLASSE RESOLVEAPOSTAS

- método `resolveApostas()`: compara as apostas feitas com os resultados, da um resultado para a aposta e atribui a pontuação ao usuario

## 8.11. SERVIÇO - CLASSE REGISTRAUSUÁRIO

- método `RegistraUsuário()`: registra o usuário no banco de dados

## 8.12. SERVIÇO - CLASSE CRIADORFICHAPOSTA

- método `CriaAposta()`: adiciona aposta à ficha de apostas
- método `DeletaAposta()`: deleta aposta da ficha de apostas
- método `FazAposta()`: faz as apostas da ficha de apostas

A Figura 2 mostra as classes do sistema:

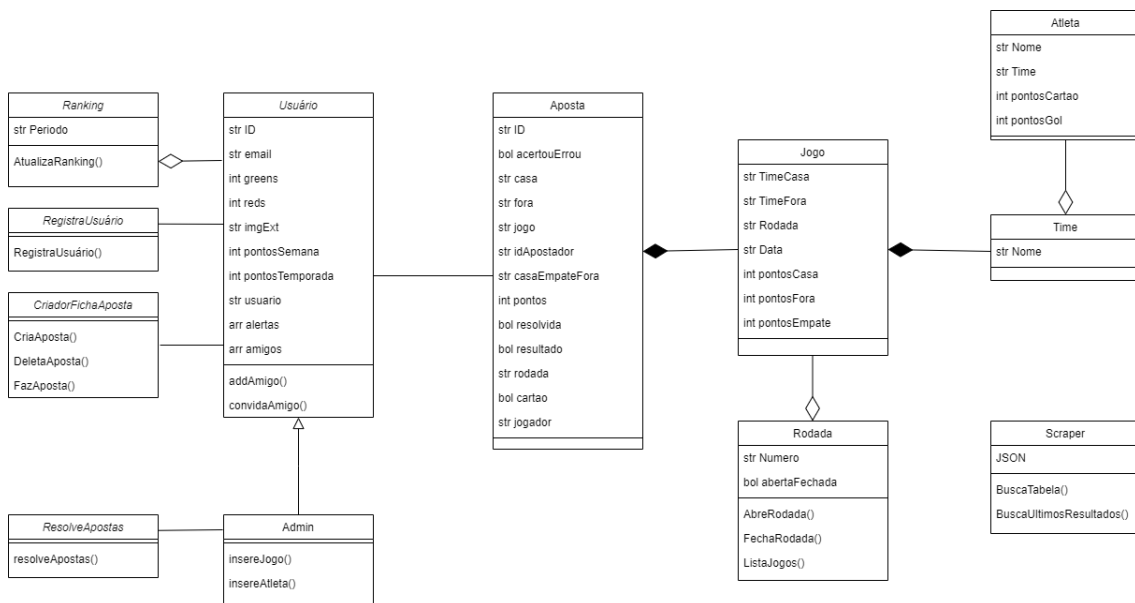


Figura 2 – Classes do Sistema. Fonte: Autor.

## 9. CASOS DE USO

Abaixo será demonstrado um caso de uso, de um jogador que entra pela primeira vez na plataforma:

O Jogador entra no site da plataforma, clica em registrar-se, preenche seus dados e clica em "Criar Conta", o sistema adiciona o usuário no banco de dados e loga retornando para a página inicial, onde está o mercado de jogos da rodada. O Jogador passa por todos os jogos da rodada, escolhendo entre "Casa", para vitória

do time da casa, "Empate"ou "Fora", para vitória do time que joga fora de casa. O Jogador clica no ícone de minimizar, na ficha de apostas, para que possa enxergar os jogos no mercado. O Jogador então seleciona todos os atletas que acha que vão fazer gol ou tomar cartão na rodada, de acordo com cada jogo da semana. Depois de fazer todas as seleções, o Jogador clica em "apostar"na ficha de apostas e recebe uma confirmação do sistema dizendo que as apostas foram computadas. A seguir, entra na área "Minhas Apostas"e confere as seleções feitas, fechando a plataforma em seguida.

## 9.1. FLUXOS DE ATIVIDADES

As figuras 3, 4, 5, 6 mostram fluxos de atividades do sistema.

A figura 3 descreve o ato de fazer um palpite dentro da plataforma. O Jogador acessa o sistema, se estiver logado segue para visualizar o mercado de jogos, caso contrário é preciso logar para fazer um palpite. Depois seleciona os resultados desejados e os atletas, precisando editar o Jogador exclui alguma opção na ficha e seleciona outro resultado, por fim confirma os palpites.

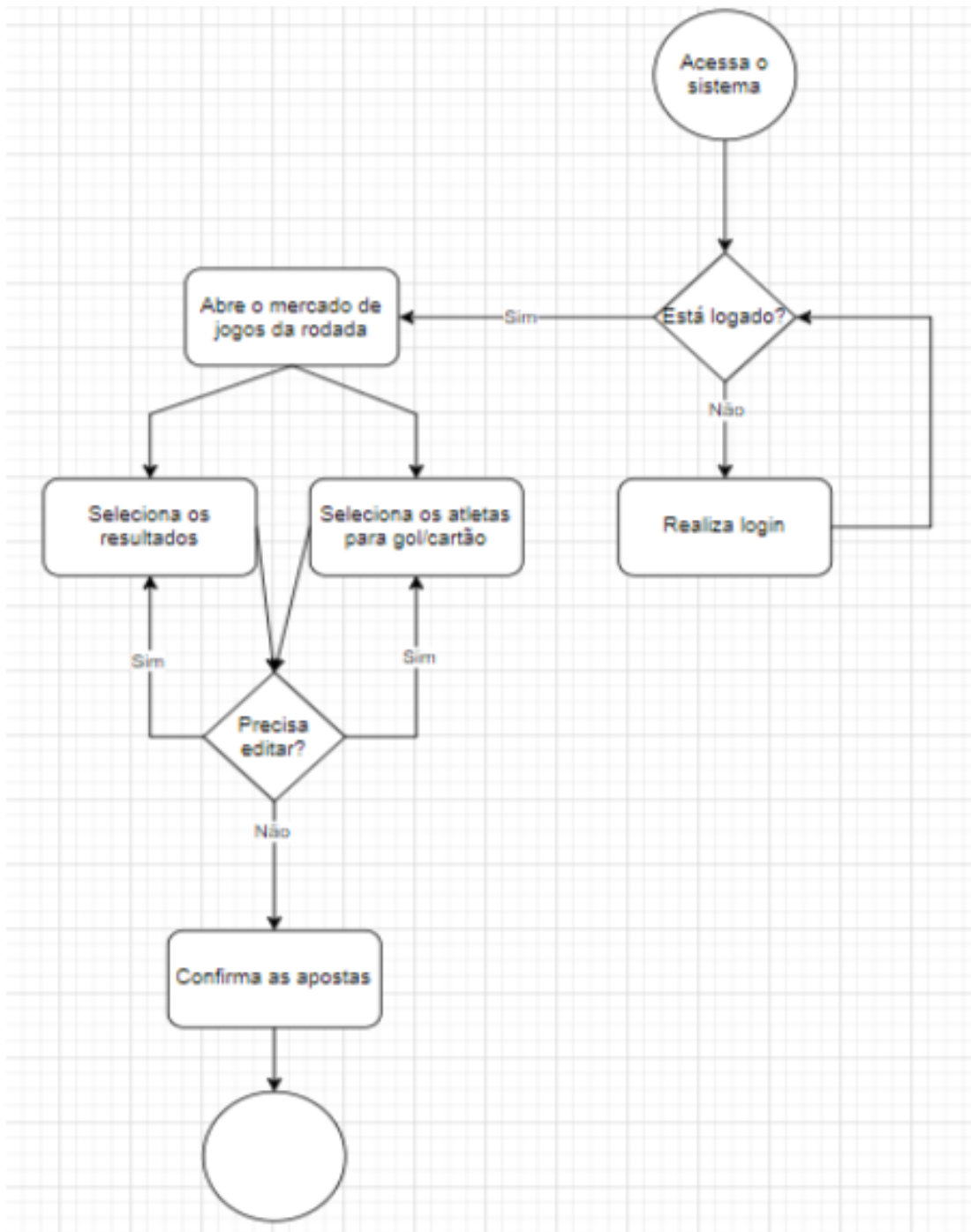


Figura 3 – Fluxo de Atividade: Fazer palpite. Fonte: Autor.

A figura 4 mostra como adicionar um amigo, é preciso estar logado, clicar na aba Amigos, depois em Adicionar, digitar o nome do usuário e clicar em pesquisar, caso o usuário exista, clique em enviar convite.

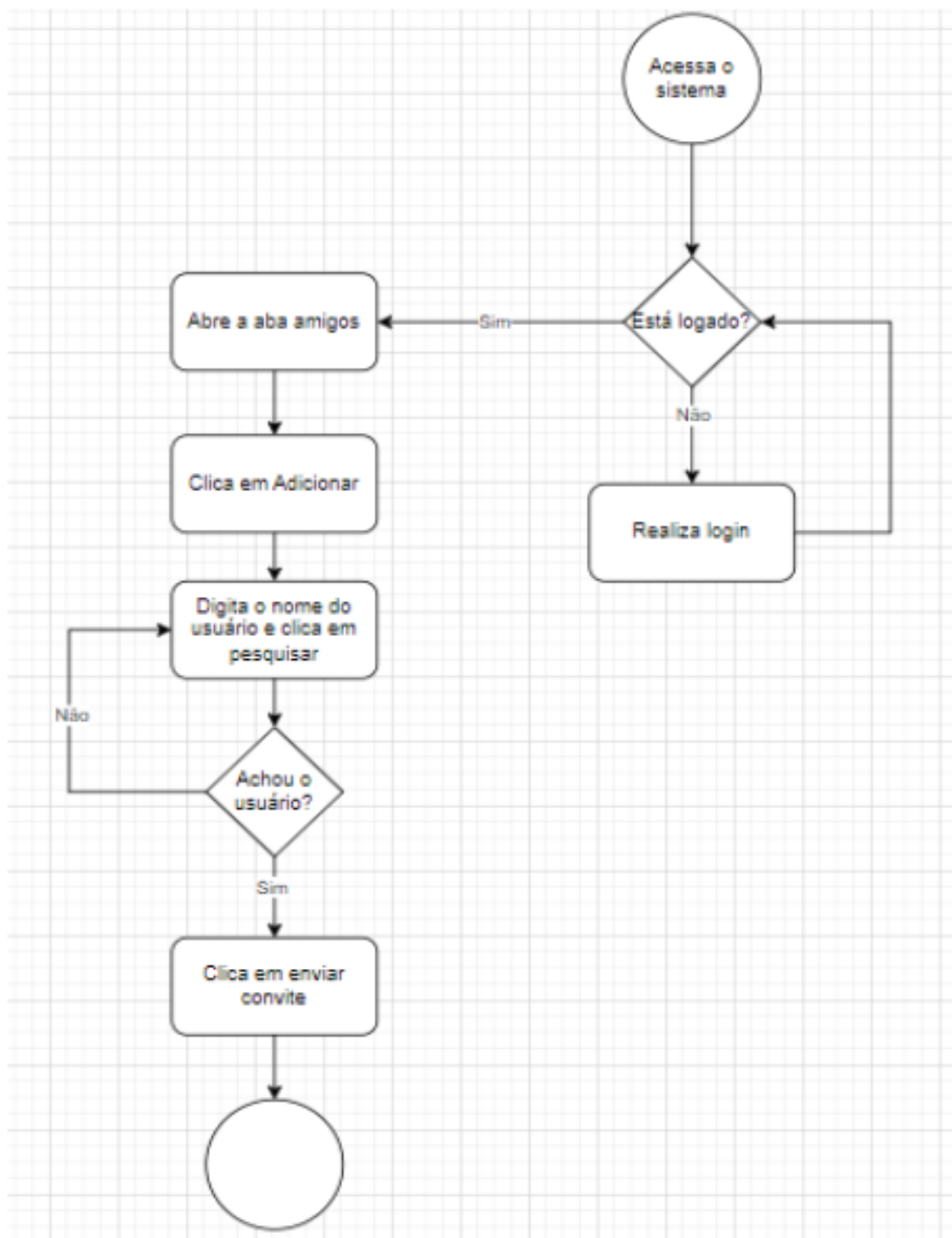


Figura 4 – Fluxo de Atividade: Adicionar amigo. Fonte: Autor.

A figura 5 mostra a ação de inserir jogos na plataforma, é preciso estar logado e ser admin, depois abrir a aba Admin, inserir os dados do jogo e depois clicar em inserir jogo, então o sistema calcula as pontuações de cada time de acordo com o jogo e adiciona o jogo à rodada.

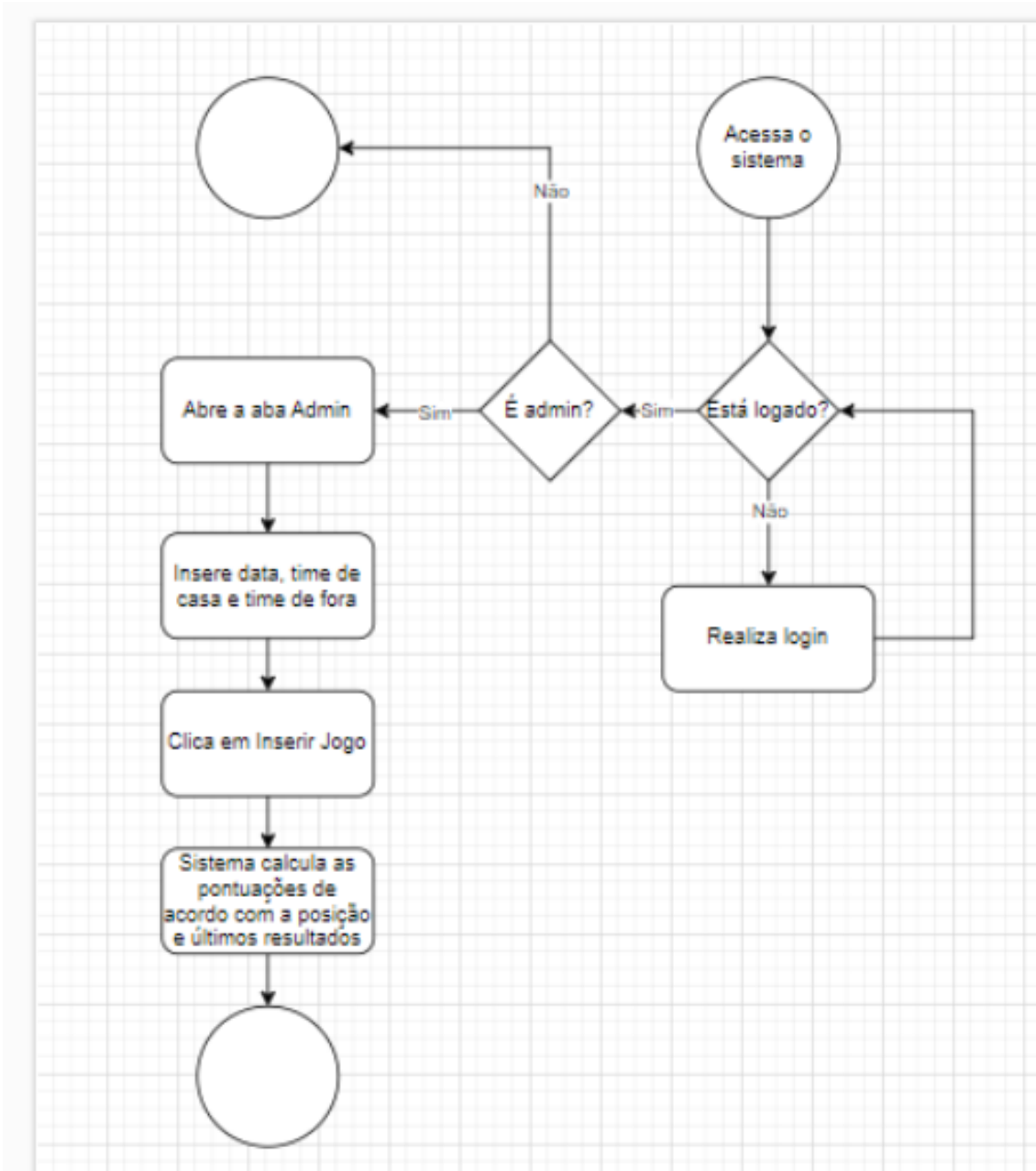


Figura 5 – Fluxo de Atividade: Inserir Jogos. Fonte: Autor.

A figura 6 mostra como resolver as apostas dos jogadores, é preciso estar logado e ser admin, abrir a aba Admin e clicar em resolver as apostas, o sistema então verifica os resultados, resolve as apostas e atribui as pontuações aos jogadores.

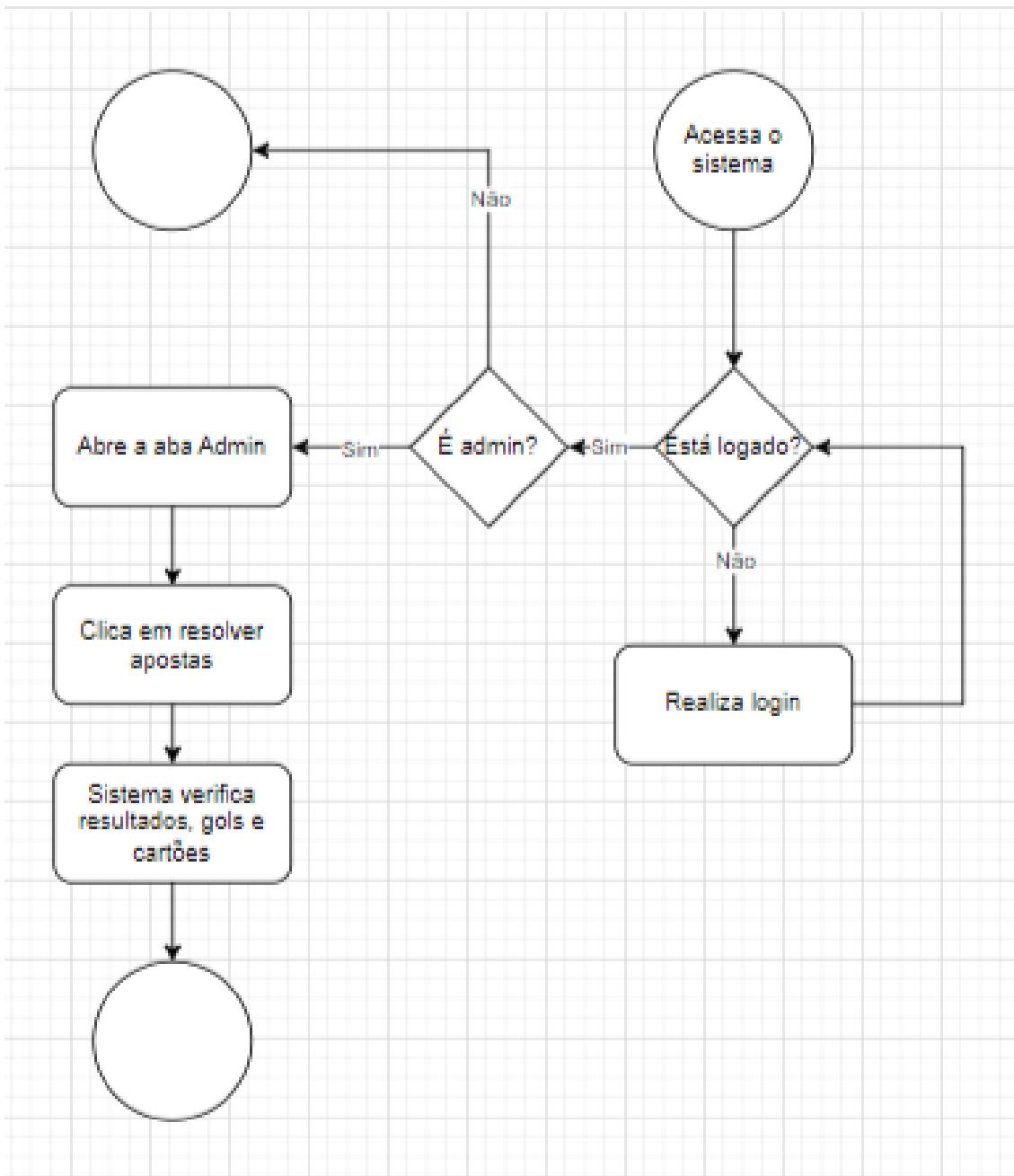


Figura 6 – Fluxo de Atividade: Resolver Palpites. Fonte: Autor.

## 10. LAYOUTS

As figuras 7 a 15 mostram os layouts mais importantes do sistema.

A figura 7 mostra a tela de admin, com os botões para recuperar resultados, resolver as apostas, fechar a rodada, abrir a rodada e abrir o mercado, um campo para inserir jogos e outro para inserir atletas.





Figura 7 – Tela do admin. Fonte: Autor.

A figura 8 mostra os campos para registro do usuário, inserir imagem de perfil, usuário e senha.

Preencha os dados

Procurar... Nenhum a...cionado.

Imagem de perfil

Nome de usuário

E-mail

Senha

Repita a senha

Criar conta

Figura 8 – Registro Usuário. Fonte: Autor.

A figura 9 possui os campos de login, com email e senha.

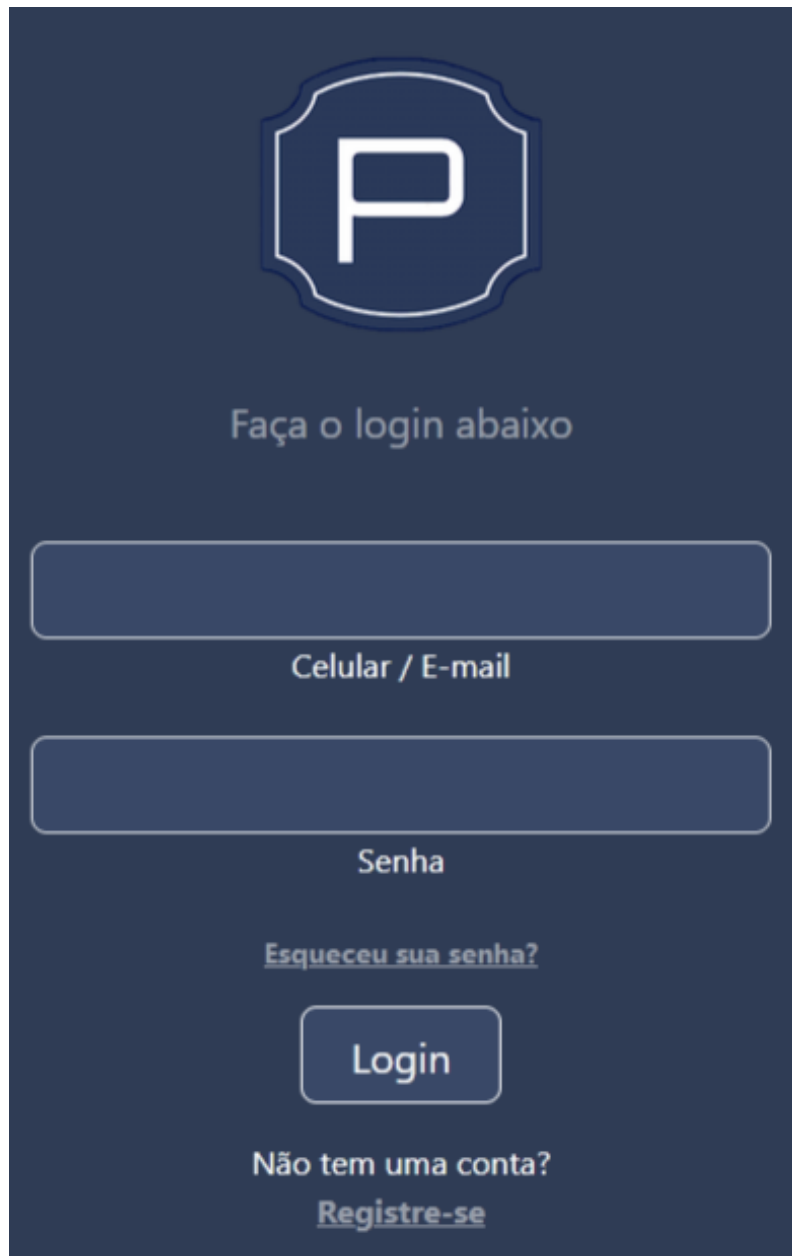


Figura 9 – Login. Fonte: Autor.

A figura 10 mostra o mercado de jogos, com cada card de cada jogo, que possuem os botões para selecionar os resultados e os campos para selecionar os atletas, e a data do jogo.

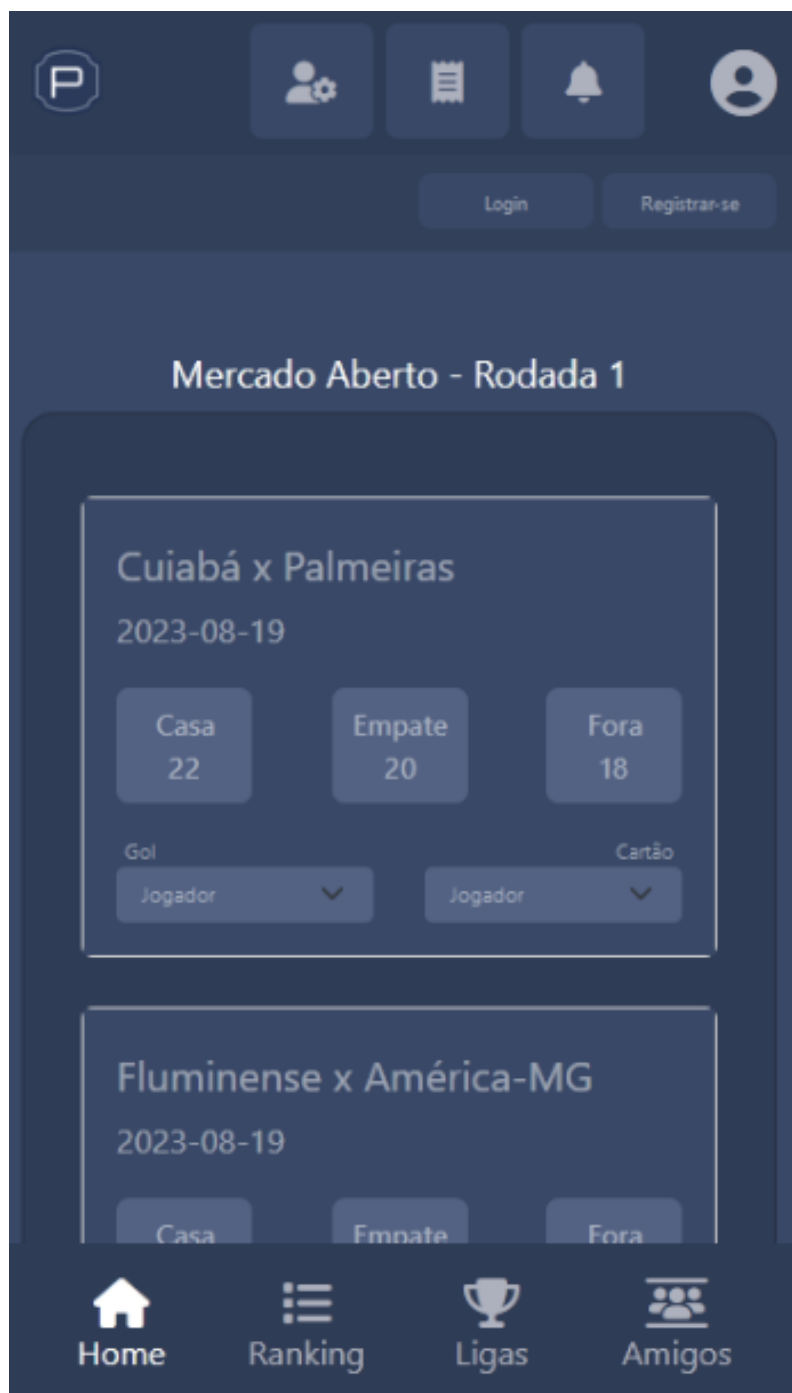


Figura 10 – Mercado de jogos. Fonte: Autor.

A figura 11 mostra a ficha de apostas, com as seleções feitas e um botão de exclusão em cada uma delas, além de o total de possíveis pontos a serem ganhos e o botão para realizar as apostas.

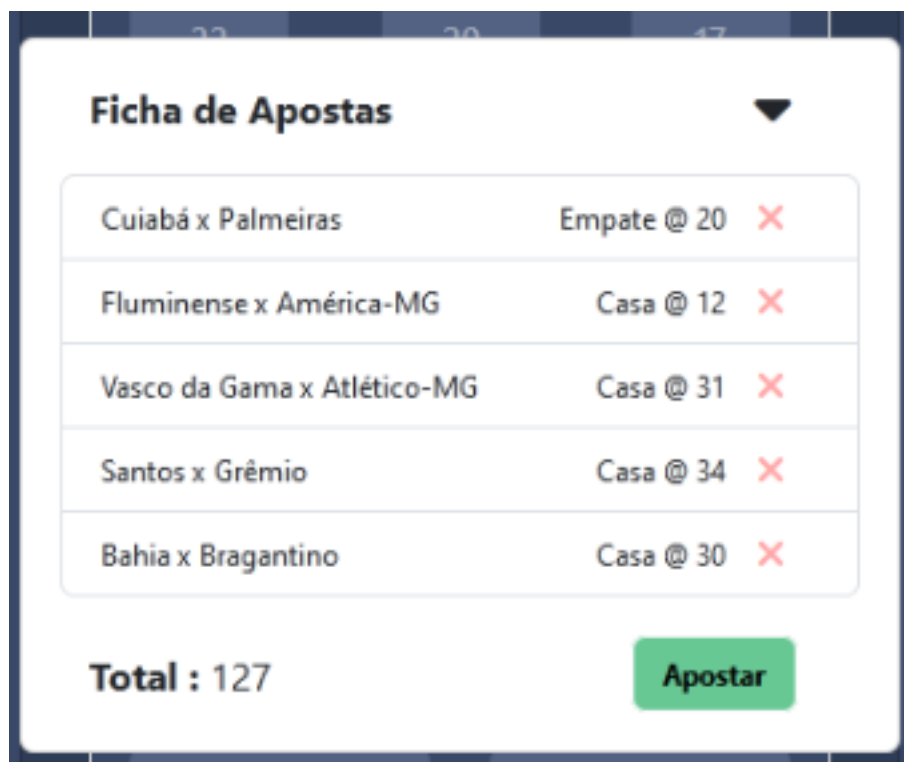


Figura 11 – Ficha de apostas. Fonte: Autor.

A figura 12 mostra o layout do ranking, com a posição atual do jogador e um campo para selecionar ranking semanal ou de temporada.



Figura 12 – Ranking Geral Temporada. Fonte: Autor.

A figura 13 mostra o layout das apostas feitas, com a descrição do palpite, junto com os possíveis pontos e um ícone mostrando se foi resolvida ou não, e se acertou o palpite ou não.



Figura 13 – Apostas Feitas. Fonte: Autor.

A figura 14 mostra a tela de notificações, com um pedido de amizade, junto com os botões de aceitar ou recusar.



Figura 14 – Notificações. Fonte: Autor.

A figura 15 mostra o campo para adicionar amigos, um campo para pesquisa e um campo para os resultados da pesquisa, junto com o botão de enviar convite.



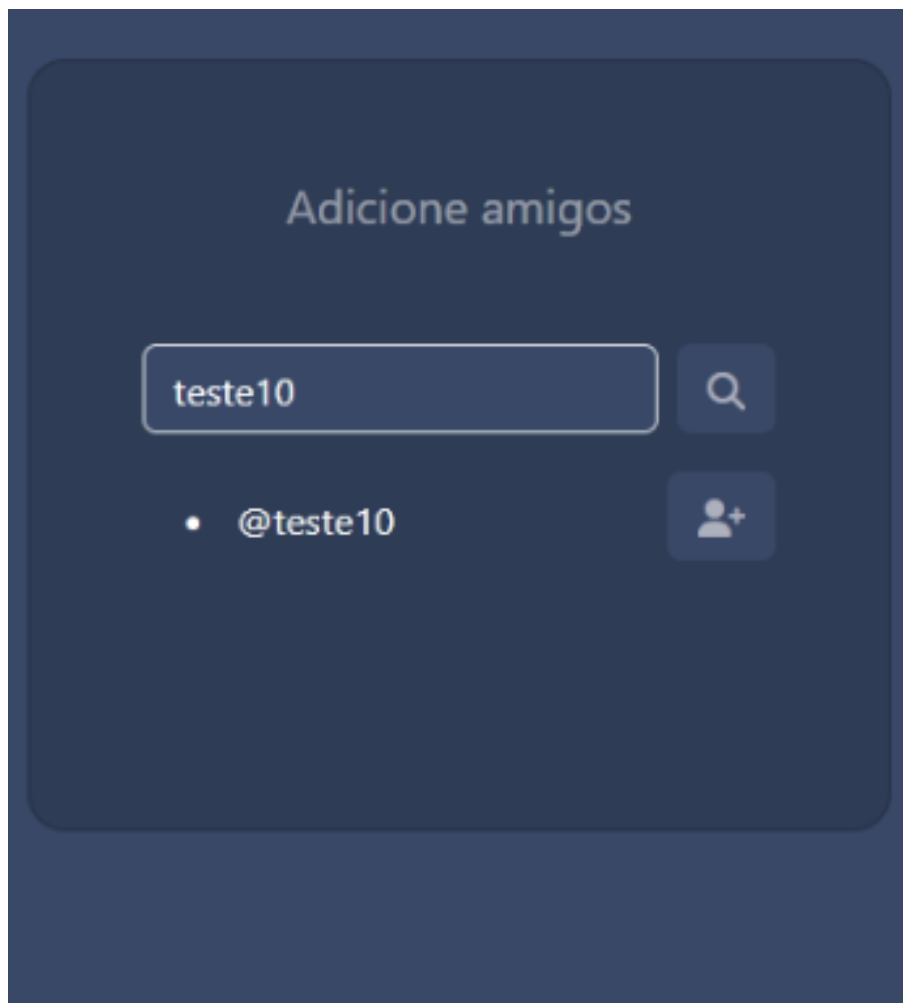


Figura 15 – Adicionar amigo. Fonte: Autor.

## 11. RESULTADOS

Para avaliar os resultados do projeto, foi criado um formulário dentro da plataforma, com uma pergunta sobre a facilidade de uso da plataforma, uma avaliação de um a dez sobre a experiência do usuário com o jogo, além de um espaço para comentários e sugestões. Os resultados se mostraram satisfatórios, comprovando a análise prévia que foi feita. Houve comentários sobre as cores da interface, sugerindo que o contraste entre os elementos poderia ser maior, além da falta de um tutorial explicando como funciona o jogo. Em relação a esse último tópico, foi criada uma aba "Ajuda" descrevendo brevemente o funcionamento do jogo para novos usuários.

Um detalhe que vale ser mencionado é que a maioria dos usuários não fez todas as apostas possíveis, o que não estava previsto. A ideia era que todos fizessem todas as apostas possíveis na rodada.

Uma evolução do jogo será: ao invés de um mercado com todos os jogos, um botão de "Começar" será adicionado à página inicial, que, ao ser clicado abre um carrossel, com cada jogo da rodada, obrigando os usuários a escolherem todos os

palpites possíveis, um jogo por vez.

A figura 16 mostra os resultados da pesquisa de experiência ao jogar, com um voto nota oito, um voto nota nove e três votos nota dez.

A figura 17 mostra os comentários e sugestões: "muito bom, excelente site muito bom o layout e é bem intuitivo", "não entendi o funcionamento do jogo logo de cara demorei um pouco para perceber como se joga", "o contraste das cores poderia ser melhor".

Seguem os resultados do formulário:

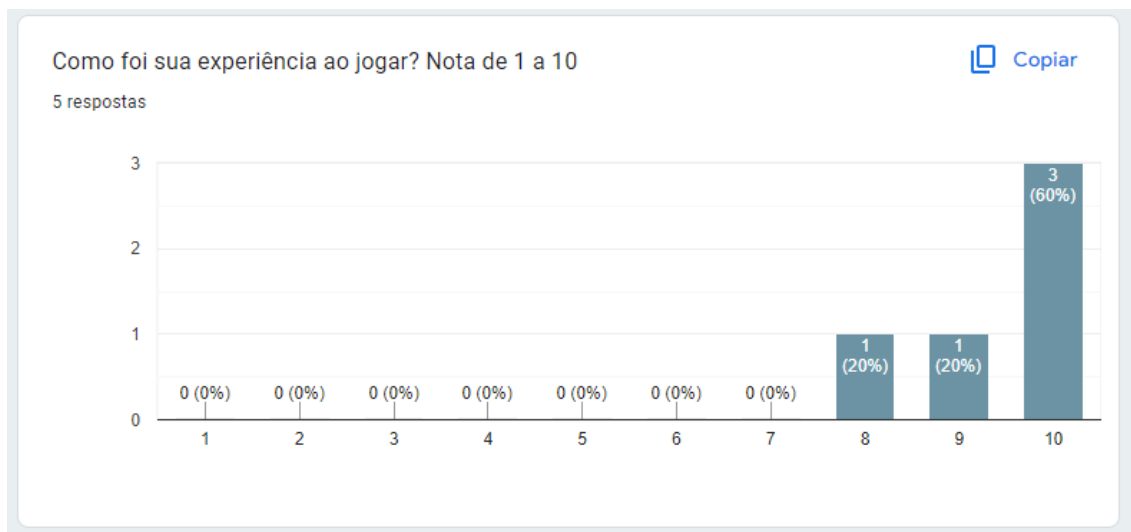


Figura 16 – Experiência do usuário: Nota de 1 a 10. Fonte: Autor.

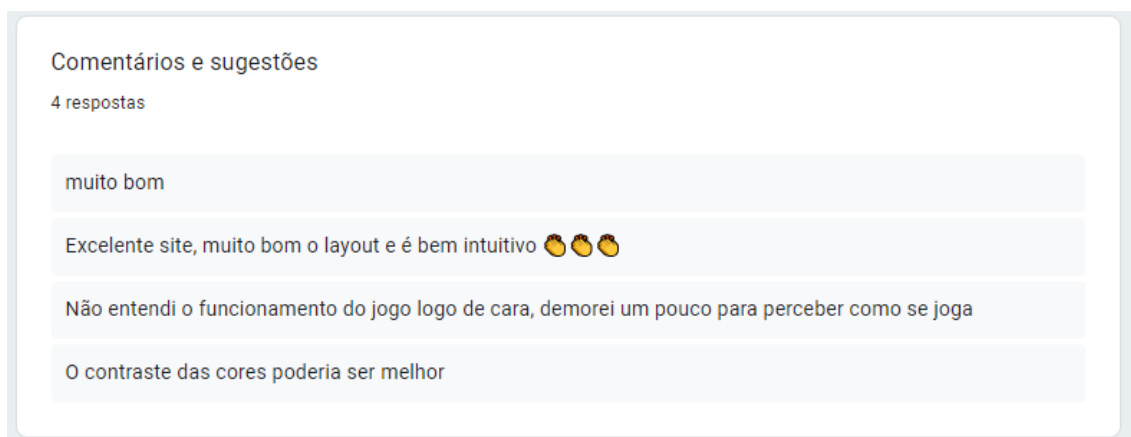


Figura 17 – Comentários e sugestões. Fonte: Autor.

## 12. CONCLUSÃO

O propósito deste artigo é demonstrar o processo de criação de um jogo *web*, desde o esboço inicial até a fase de testes e lançamento. Para isso, foram realizadas

reuniões iniciais de alinhamento, seguidas de meses de desenvolvimento, testes e correções, até chegar no produto final. Foi aplicado um formulário, para identificar como foi a experiência dos usuários com o jogo. Para o futuro, será avaliado uma série de melhorias e novas funcionalidades para o jogo, seguindo o mesmo padrão apresentado neste artigo, até chegar no lançamento final.

Link para a plataforma: <https://palpiteirofc.netlify.app/>

Link para o repositório no GitHub: <https://github.com/guilhermelff/front-palpiteiro>

## Referências

- BERNERS-LEE, T. *HTML*. 1993. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 7 nov. 2023.
- BET365. *bet365*. 2000. Disponível em: <https://www.bet365.com//HO/>. Acesso em: 7 nov. 2023.
- BIILMANN, M. *Netlify*. 2014. Disponível em: <https://www.netlify.com/>. Acesso em: 7 nov. 2023.
- CROCKFORD, D. *JSON*. 2000. Disponível em: <https://json.org/json-pt.html>. Acesso em: 7 nov. 2023.
- DAHL, R. *Node.js*. 2009. Disponível em: <https://nodejs.org/en>. Acesso em: 7 nov. 2023.
- EICH, B. *JavaScript*. 1995. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 7 nov. 2023.
- GLOBO, G. *Cartola FC*. 2005. Disponível em: <https://cartola.globo.com/>. Acesso em: 7 nov. 2023.
- MORSCHHEUSER, B. How to gamify? a method for designing gamification. *HICSS*, 2017.
- NIELSEN, J. Heuristic evaluation of user interfaces. *Proc. ACM CHI'90 Conf, Seattle, WA, p. 249-256*, 1990.
- OTTO, M. *Bootstrap*. 2011. Disponível em: <https://getbootstrap.com/>. Acesso em: 7 nov. 2023.
- PAGE, L. *Google*. 1998. Disponível em: <https://www.google.com.br/?hl=pt-BR>. Acesso em: 7 nov. 2023.
- REENSKAUG, T. *MVC*. 1979. Disponível em: <https://folk.universitetetioslo.no/trygver/themes/mvc/mvc-index.html>. Acesso em: 7 nov. 2023.
- REFERENCE, S. *FBRef*. 2000. Disponível em: <https://fbref.com/pt/>. Acesso em: 7 nov. 2023.
- TAMPLIN, J. *Firebase*. 2011. Disponível em: <https://firebase.google.com/?hl=pt-br>. Acesso em: 7 nov. 2023.
- TORVALDS, L. *Git*. 2005. Disponível em: <https://git-scm.com/>. Acesso em: 7 nov. 2023.
- W3C. *CSS*. 1996. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 7 nov. 2023.
- WANSTRATH, C. *GitHub*. 2008. Disponível em: <https://github.com/>. Acesso em: 7 nov. 2023.

WIKIPEDIA. *Sports Rating System*. 2023. Disponível em: <https://t.ly/QgPDU>.  
Acesso em: 7 nov. 2023.

## APÊNDICE A - Lista de ilustrações

|   |    |
|---|----|
| Figura 1 – Arquitetura do Sistema. Fonte: Autor. . . . .                  | 8  |
| Figura 2 – Classes do Sistema. Fonte: Autor. . . . .                      | 11 |
| Figura 3 – Fluxo de Atividade: Fazer palpite. Fonte: Autor. . . . .       | 13 |
| Figura 4 – Fluxo de Atividade: Adicionar amigo. Fonte: Autor. . . . .     | 14 |
| Figura 5 – Fluxo de Atividade: Inserir Jogos. Fonte: Autor. . . . .       | 15 |
| Figura 6 – Fluxo de Atividade: Resolver Palpites. Fonte: Autor. . . . .   | 16 |
| Figura 7 – Tela do admin. Fonte: Autor. . . . .                           | 17 |
| Figura 8 – Registro Usuário. Fonte: Autor. . . . .                        | 18 |
| Figura 9 – Login. Fonte: Autor. . . . .                                   | 19 |
| Figura 10 – Mercado de jogos. Fonte: Autor. . . . .                       | 20 |
| Figura 11 – Ficha de apostas. Fonte: Autor. . . . .                       | 21 |
| Figura 12 – Ranking Geral Temporada. Fonte: Autor. . . . .                | 22 |
| Figura 13 – Apostas Feitas. Fonte: Autor. . . . .                         | 23 |
| Figura 14 – Notificações. Fonte: Autor. . . . .                           | 24 |
| Figura 15 – Adicionar amigo. Fonte: Autor. . . . .                        | 25 |
| Figura 16 – Experiência do usuário: Nota de 1 a 10. Fonte: Autor. . . . . | 26 |
| Figura 17 – Comentários e sugestões. Fonte: Autor. . . . .                | 26 |

## Agradecimentos

Queria agradecer a minha família pelo suporte, a todos os meus amigos de curso, que fizeram dessa caminhada mais leve e fácil, e a todos os professores e funcionários da UFF, que fazem tudo acontecer.