

Aplicação web back-end de um fantasy game

Kaio de Melo Rodrigues^a e Celso Ap.de França^b

Departamento de Engenharia Elétrica da Universidade Federal de São Carlos

Resumo – No Brasil, o esporte de maior popularidade é o futebol e, portanto, é reconhecido em cenários que vão além dos gramados, como o caso dos *Fantasy Games*. Os usuários deste estilo de jogo criam e gerenciam suas próprias equipes online, usando jogadores reais de times de futebol de todo o mundo. A cada jogador é atribuído um valor de pontos com base em sua habilidade e histórico de desempenho. Há hoje distintos *Fantasy Games* de futebol, mas o de maior relevância no país é o Cartola FC. Neste trabalho foi desenvolvido uma aplicação *web Back-End* de um *Fantasy Game* de futebol similar ao Cartola da Rede Globo, chamado de InterRep Manager. Nele, os usuários gerenciam as pontuações de jogadores reais que participaram do InterReps, um torneio de futebol entre as repúblicas de São Carlos ocorrido no ano de 2022. As súmulas do torneio foram utilizadas na aplicação. Para avaliar a funcionalidade do aplicativo foi simulado o desempenho de três usuários e 15 jogadores em três partidas. Até o momento o software mostrou-se funcional, desde a etapa do *login*, até o cálculo da pontuação.

Palavras-Chave – Fantasy Game, Aplicação web .Net, ASP.NET RESTful Web API

1. INTRODUÇÃO

Constantes mudanças ocorrem no mundo e na sociedade como um todo devido a evoluções nas áreas da ciência, tecnologia, sociedade e ambiente. Um marco significativo nesse sentido foi o surgimento da internet [1], cujo impacto na vida cotidiana é direto e abrange áreas como economia, produtividade, redução de custos operacionais em diversos setores, como público, privado, acadêmico e esportivo [2].

Assim, com a ascensão das plataformas digitais, juntamente com a popularidade de diferentes esportes, o estilo de jogo conhecido como *Fantasy Game* ganhou espaço relevante no dia a dia das pessoas. Da mesma forma como os demais jogos, para os *fantasy games* se faz necessário a compreensão de símbolos, imagens e artefatos para a realização dessa atividade. Os *fantasy games* existem para os mais distintos esportes, tendo efetiva popularidade entre os países da América [3].

O futebol é o esporte com maior estima dentro dos *fantasy games*, influenciando torcedores assíduos a encontrarem formas de mostrar seus conhecimentos. Uma delas é através das plataformas de apostas. Espera-se que o mercado de *fantasy games*, avaliado em US\$ 26 bilhões cresça 120% até 2026, de acordo com uma pesquisa realizada pela TechNavio, empresa focada em mercados globais e novas tecnologias. No ano passado, só o mercado norte-americano movimentou o equivalente a 44 bilhões de reais. [4].

No Brasil, o *Fantasy Game* de maior sucesso é o Cartola FC, um jogo eletrônico de futebol. Assim, baseado nessa plataforma, o objetivo desse projeto foi propor a criação de uma aplicação *web Back-End*, de um *Fantasy Football* do torneio InterREP São carlense.

O torneio InterREP é um torneio de futebol *society* entre repúblicas estudantis da cidade de São Carlos. Atualmente o torneio conta com três modalidades: Série A e Série B apenas para atletas masculinos, e o Campeonato Feminino.

A série B possui 32 equipes participantes e seus jogos são eliminatórios em que os 4 com a melhor classificação recebem o direito de jogar na Série A. A Série A por sua vez também possui 32 equipes, sendo 4 provenientes da série B e 28 de jogos classificatórios das edições anteriores. A primeira etapa é dividida em 8 grupos de 4 equipes. As 2 melhores colocadas de cada grupo se classificam para a próxima fase, e desta fase em diante os jogos são no modelo eliminatórias. O Feminino possui 16 equipes e o modelo de disputa é eliminatório desde a primeira rodada.

O intuito da realização desse trabalho é melhorar a visibilidade do torneio, além de agregar valor e prestígio. Atualmente, a organização registra manualmente a performance

individual dos atletas envolvidos nas respectivas partidas no mesmo arquivo que a súmula. Este arquivo é do tipo Excel. Após a implementação do *fantasy game*, esses dados serão inseridos via arquivo csv na plataforma para a computação da performance dos jogadores na partida utilizando o *software* desenvolvido neste trabalho.

2. TRABALHOS RELACIONADOS

Não foi encontrado um estudo que cria um *fantasy game* com a utilização de uma aplicação web, entretanto foi possível dividir a aplicação em funcionalidades e pesquisar separadamente cada funcionalidade. Estas podem ser divididas em:

1. Busca, inserção, atualização e deleção de dados;
2. Busca e inserção de arquivos no Amazon S3.
3. Autenticação.

2.1 BUSCA, INSERÇÃO, ATUALIZAÇÃO E DELEÇÃO DE DADOS

Um desafio comum para desenvolvedores em todo o mundo é gerenciar os dados que são enviados repetidamente entre um aplicativo e o banco de dados. Para facilitar isso, é utilizado *Object-Relational Mapping* (ORM), em português, mapeamento objeto-relacional, cuja função é simplificar a persistência de dados, permitindo que os desenvolvedores trabalhem com objetos de linguagem de programação familiar em vez de ter que escrever manualmente o código SQL (*Structured Query Language*) para interagir com o banco de dados [5].

Dapper é uma micro-ORM para .NET criada pela *Stack Exchange* e projetada para ser rápida, leve e eficiente, além de fornecer uma API (*Application Programming Interface*) simples para consultar bancos de dados usando comandos SQL [5].

Foi escolhido utilizar Dapper, porque dentre as ferramentas de mapeamento objeto-relacional (ORM), esta foi a que se mostrou ser a de melhor performance mantendo simplicidade para a implementação do código [6]. Para a parte de acesso e manipulação do banco de dados, em Macoratti [6], tem-se orientações de como utilizar a biblioteca Dapper para executar consultas e inserções em banco de dados via .Net.

2.2 BUSCA E INSERÇÃO DE ARQUIVOS NO AMAZON S3

A AWS (*Amazon Web Services*) é uma plataforma de serviços em nuvem da Amazon, que oferece recursos de computação, armazenamento, rede e banco de dados, entre outros, para empresas e indivíduos que desejam executar aplicativos e serviços na nuvem.

Um destes serviços é o Amazon S3 (*Simple Storage Service*/ Serviço de armazenamento simples da Amazon) que é o armazenamento de objetos em nuvem escalável, seguro e durável [7]. Este será utilizado para a parte de busca e inserção de arquivos da aplicação. O artigo criando aplicações .NET para AWS [8], explica como realizar a conexão a AWS, o upload e download de arquivos no S3.

O Amazon S3 foi escolhido por possuir alta durabilidade, disponibilidade e acesso rápido aos dados, além de tornar quase impossível problemas de acesso ou perda de arquivos [9], além de ser um serviço de armazenamento de baixo custo, com um modelo de cobrança simples baseado em “pagar conforme o uso” [10].

2.3 AUTENTICAÇÃO

A autenticação do usuário e o gerenciamento de acesso aos aplicativos de Internet são fatores importantes em termos de segurança de rede. Na comunicação cliente-servidor, os servidores mantêm a identidade dos clientes durante as interações e, se as medidas de segurança apropriadas não forem aplicadas para proteger dados do usuário, isso poderá levar ao uso indevido dessas informações [11]

O acesso aos dados torna-se mais complexo quando um mesmo usuário se comunica com o servidor de diferentes dispositivos e plataformas como iOS, Android, Windows e Linux, celular, computador pessoal, entre outros. Diante disso, o JWT (*Json Web Token*) foi utilizado neste trabalho como um *Bearer token* para a autenticação da aplicação. Na Figura 1, foi representada a sequência de aplicação do JWT. Como definição, o *Bearer token* é um tipo de *token* (símbolo) de autenticação utilizado em protocolos de segurança da API (Interface de Programação de Aplicativos). Ele é utilizado para permitir o acesso autorizado de um aplicativo a um servidor protegido por autenticação. O JWT é um padrão aberto que define um formato compacto e seguro para transmitir informações entre partes como um objeto JSON (*JavaScript Object Notation*) [12].

O JWT tem sido utilizado em vários estudos apresentados na literatura por manter a autenticidade dos clientes durante a interação com o servidor [11]. Esta é uma solução escalável com benefícios significativos de desempenho para controle de acesso do usuário em sistemas distribuídos descentralizados e de grande escala. Em Veras [13], tem-se a explicação de como implementar a etapa de autenticação utilizando *Bearer e Json Web Token* (JWT) [14].

Um esquema de autenticação baseado em JWT tem dois benefícios principais. O primeiro é que ele melhora o desempenho e o dimensionamento do sistema, porque as informações de autenticação são transmitidas pela comunicação em vez de serem recuperadas do servidor [15]. Isso significa que as informações do usuário são mantidas durante a sessão de comunicação. O segundo benefício é que ele simplifica a autorização em sistemas distribuídos, pois todas as informações necessárias estão contidas no próprio *token*, eliminando a necessidade de conexões adicionais entre os componentes protegidos. Isso é garantido por meio de criptografia e assinaturas digitais, que impedem que o cliente altere as informações no *token* [15].

3. FUNDAMENTAÇÃO TEÓRICA

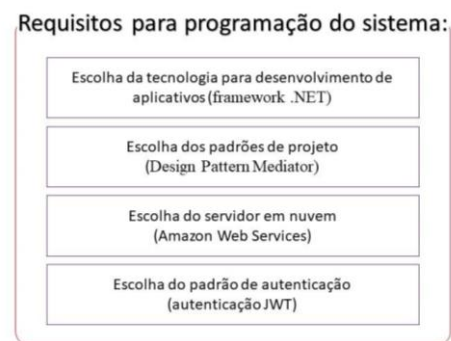
Com a ascensão das plataformas digitais nos últimos anos, há, disponível uma nova categorização de estilo de jogo, os *Fantasy Games* [3].

Fantasy games são jogos online no qual usuários montam e gerenciam equipes virtuais, baseado no desempenho dos jogadores e de partidas reais. Dentre estes, o esporte que mais se destaca é o futebol representando 54% [16]. A mais famosa plataforma do Brasil de *Fantasy Football* é conhecida como Cartola-FC 1, criada pela Sport TV no ano de 2005 [17].

Neste jogo, é possível que os usuários comandem seu próprio time, com jogadores e técnicos escalados de acordo com a sua preferência. O jogo é baseado em estatísticas de jogadores e técnicos da vida real que estão na Série A do Campeonato Brasileiro e nas súmulas da CBF, sendo que seus rendimentos influenciam diretamente na pontuação do jogo online [16].

Assim como as demais plataformas e aplicativos digitais, a programação dos *Fantasy Games* deve ser iniciada com levantamentos de requisitos que garantirão sua facilidade, segurança e estabilidade [18]. Na Figura 1, estão representados os principais requisitos para a programação de um sistema.

Figura 1: Requisitos para a programação de um sistema



Fonte: Feito pelo autor

Determinar a escolha das melhores tecnologias e arquiteturas é fundamental para o sucesso do projeto. Uma das tecnologias mais populares para o desenvolvimento de aplicativos de web é o *framework* .NET. Desenvolvida pela Microsoft esta plataforma oferece uma ampla variedade de recursos, como uma biblioteca de classes abrangente,

ferramentas de desenvolvimento preponderante, suporte a várias linguagens de programação, tanto em aplicações de pequeno a grande porte [19].

Para garantir a segurança do sistema desenvolvido e a facilidade em sua sustentação, o uso de padrões de projeto, como o *Design Pattern Mediator*, é amplamente utilizado [20]. O *Design Pattern Mediator* promove o desacoplamento entre os objetos do sistema, minimizando as interdependências de componentes, resultando em *designers* de *softwares* mais simples, escaláveis e menos propensos a erros. Além de uma maior capacidade de reutilização e testabilidade [20].

A partir deste momento, é necessário o desenvolvimento de sistemas na nuvem, como é o caso da *Amazon Web Services (AWS)*. Esta plataforma é uma das principais opções devido à sua confiabilidade, escalabilidade e segurança. Dentre eles, a *Amazon S3 (Amazon Simple Storage Service)* é um exemplo adequado do serviço da AWS por ser segura para armazenar e recuperar arquivos [21].

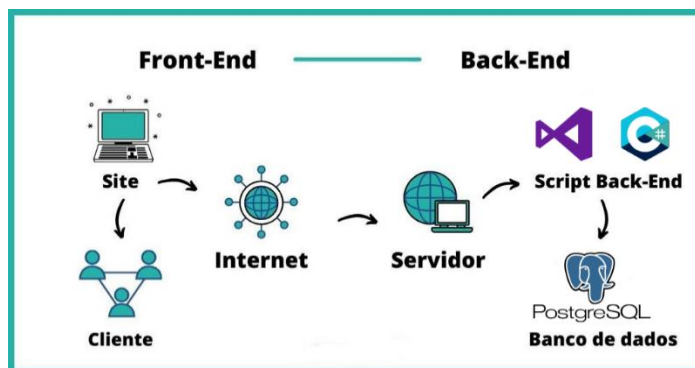
Como segurança para a autenticação, um sistema de software frequentemente utilizado é *JWT (JSON Web Token)*. O *JWT* tem sido utilizado em vários estudos apresentados na literatura para manter a autenticidade dos clientes durante a interação com o servidor. É um padrão aberto para representar as credenciais do usuário de forma segura, podendo ser utilizado para autenticação do usuário em sistemas de software, permitindo assim que apenas o usuário permitido acesse determinado recurso [22].

4. METODOLOGIA

A criação de um site é um processo que envolve o desenvolvimento de uma metodologia bem estruturada, garantindo eficiência e qualidade para o produto final. Por isso, é essencial contar com um conjunto de programas e tecnologias adequadas para alcançar os objetivos desejados. A programação em *front-end* e *back-end* são dois serviços distintos que compõem o desenvolvimento de um aplicativo web. Enquanto o *back-end* lida com a lógica do negócio, processamento e interação com o banco de dados, o *front-end*

do aplicativo gerencia a interação do usuário. Por meio de chamadas de API (*Application Programming Interface/ Interface de Programação de Aplicação*), esses dois podem se comunicar [23]. A Figura 2, exemplificada abaixo, representa esquematicamente a interação destas programações.

Figura 2. Esquema ilustrativo da interação dos modelos de programação *Front-End* e *Back-End*.



Fonte: Adaptado Antunes [24].

A implementação da API do software seguiu a abordagem *RESTful (Representational State Transfer)*, um estilo arquitetural utilizado para projetar serviços web que permite a comunicação de diferentes softwares. A comunicação destes dados, entre servidor e cliente, utilizando as tecnologias *JSON* e *RESTful*, resulta em um formato leve e de fácil leitura para a troca de dados [25].

Este trabalho utilizou o ambiente de programação "*Visual Studio*", uma *IDE (Integrated Development Environment)* desenvolvida pela Microsoft. Esta ferramenta é executada na área de trabalho e está disponível para diversos sistemas operacionais como *Windows*, *MacOS* e *Linux*. Desempenha um papel fundamental ao fornecer um ambiente integrado para desenvolvimento, depuração e gerenciamento de projetos, além de oferecer ferramentas para a escrita e teste dos códigos necessários como o suporte completo para o framework *.NET* e para a linguagem de programação *C#*, que foram empregados neste projeto [26].

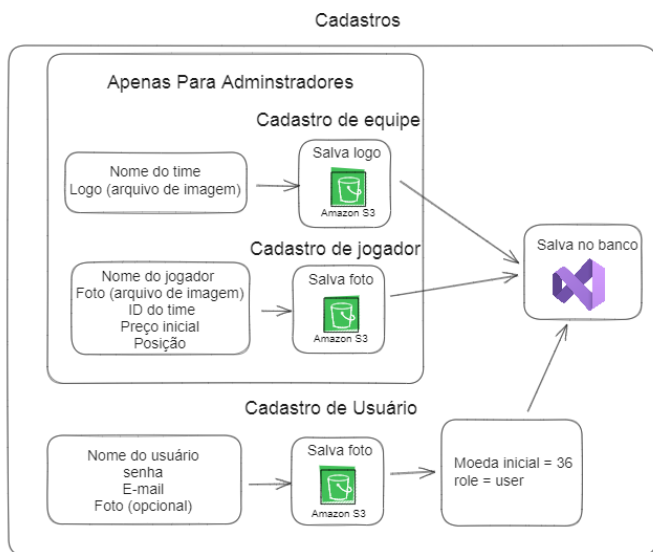
C# é uma linguagem de programação amplamente utilizada para desenvolvimento de aplicativos Web que são executados no *.NET*. Essa linguagem, orientada a objetos e orientada a componentes, oferece recursos avançados de

segurança e desempenho. Foi escolhida para o desenvolvimento *back-end*, pois oferece suporte a bancos de dados, gerenciamento de memória e recursos de rede [27].

O sistema de cadastro de usuários do aplicativo está representado no esquema abaixo (Figura 3). Esse sistema é dividido em duas partes, uma apenas para os administradores e a outra para os usuários. Na aba dos administradores será realizado o cadastro de todos os jogadores, com foto, preço, posição e time. O cadastro do usuário será realizado com um nome, senha e e-mail. Cada participante começará com 36 moedas para a compra dos jogadores.

Nesta etapa, como banco de dados, foi utilizado o PostgreSQL, um sistema de gerenciamento de banco de dados relacional de código aberto. O PostgreSQL oferece recursos avançados de armazenamento e recuperação de dados, garantindo a integridade e a segurança das informações do site [28].

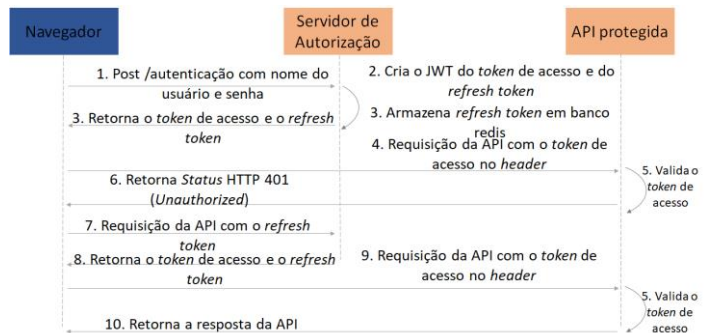
Figura 3. Esquema de cadastro dos usuários



Fonte: Feito pelo autor

O banco de dados Redis também foi utilizado no desenvolvimento, no entanto, apenas para salvar o *refresh token* utilizado no *login* do usuário. Na Figura 4 está o esquema de autenticação do usuário via *token*.

Figura 4. Esquema de autenticação do usuário via *refresh token*



Fonte: Feito pelo autor

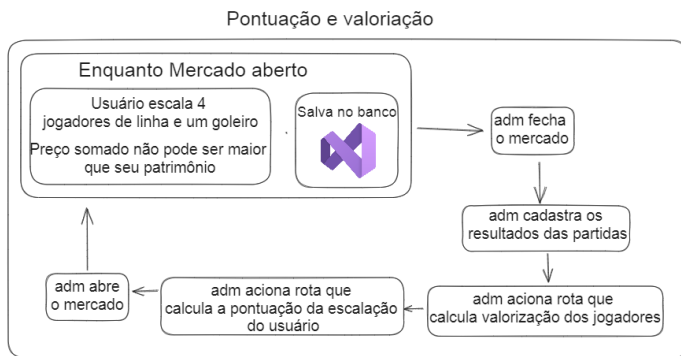
A autenticação via *refresh token* é um processo utilizado para permitir que um usuário mantenha a sessão ativa por um período prolongado sem precisar fornecer novamente suas credenciais de *login*. Assim que o usuário coloca seu nome e senha, o sistema autentica as credenciais e gera dois *tokens*: o *token* de acesso e o *refresh token*.

O *token* de acesso possui curta duração pois é usado para autenticar e autorizar as solicitações do usuário durante a sessão ativa. Já o *refresh token* é utilizado exclusivamente para solicitar um novo *token* de acesso quando o atual expira. Ele é geralmente válido por um período mais longo.

Este processo de renovação pode ser repetido várias vezes até que o *refresh token* expire ou seja revogado. Quando isso acontece, o usuário precisará fornecer novamente suas credenciais de *login* para obter um novo código.

Além disso, no desenvolvimento de um website no estilo *Fantasy Game*, é indispensável considerar a lógica de funcionamento do jogo e a pontuação dos jogadores. A Figura 5 exemplifica lógica de funcionamento do jogo enquanto que a Tabela 1 apresenta a pontuação dos jogadores.

Figura 5. Esquemática sobre a lógica de funcionamento do jogo em relação à sua pontuação e valorização.



Fonte: Feito pelo autor

A coleta dos dados de desempenho dos jogadores do torneio é realizada pela comissão organizadora após cada partida é registrada em uma planilha. Cada ação executada durante o jogo possui uma pontuação específica (Tabela 1), a qual é multiplicada pela sua ponderação correspondente. Após essa etapa, um arquivo CSV é gerado e salvo no banco de dados, contendo o nome do jogador e seus respectivos resultados.

Os usuários só conseguirão efetuar a escalação do seu time enquanto o mercado estiver aberto, sendo obrigatório escolher 4 jogadores de linha e um goleiro. O somatório dos preços dos 5 jogadores deve ser menor ou igual ao patrimônio do usuário.

Duas horas antes do jogo de cada rodada, o mercado será fechado por um administrador e reaberto somente após a inserção da nova pontuação. Neste *endpoint* será calculado novamente os valores de todos os jogadores de acordo com seu desempenho e os pontos dos usuários *online* que corresponde a soma dos jogadores escalados e seu patrimônio.

Tabela 1. Valor de pontos de cada ação desempenhada pelos jogadores.

Pontuações de defesa		Pontos	Pontuações de ataque		Pontos
Jogo sem sofrer gols	SG	5	Gol	GO	6
Jogo sem sofrer gols (goleiro)	SGG	6	Assistência	AS	4
Defesa de pênalti	DPG	6	Finalização no gol	FG	0,7
Defesa difícil	DDG	2,5	Finalização na trave	FT	1,5
Gol sofrido (goleiro)	GSG	-1,2	Falta sofrida	FS	0,5
Gol sofrido (time)	GST	-0,8	Pênalti perdido	PP	-3
Roubada de bola	RB	1,3	Pontuação de atuação	Pontos	
Gol contra	GC	-5	Entrou em campo	EC	1
Cartão Vermelho	CV	-4			
Cartão Azul	CAZ	-3			
Cartão Amarelo	CA	-2			
Falta cometida	FC	-0,7			

Fonte: Feito pelo autor

Algumas equações de ajuste para o cálculo do valor dos jogadores foram elaboradas e estão representadas abaixo, visando evitar valores excessivamente elevados, negativos ou significativamente inferiores a 1. Como determinado, caso o preço seja menor que R\$0,75 ele será mantido nesse mesmo valor. Por outro lado, se o preço for superior a R\$0,75, será utilizado um multiplicador K para evitar uma valorização excessiva.

$$X = (UP - PM)K \quad (1)$$

$$Y = \frac{(t-VA)}{t} \quad (2)$$

$$NV = VA + (XY) \quad (3)$$

O valor da constante K utilizada na fórmula apresentada é determinada pela comparação entre a última pontuação do jogador (UP) e sua pontuação média (PM). Dessa forma, se a última pontuação for inferior à média, K é igual a 0,15 para jogadores com valores menores que 6; 0,25 para jogadores com valores entre 6 e 9; e 0,5 para jogadores com valores iguais ou superiores a 10.

Por outro lado, se a última pontuação for maior ou igual à média, K é inicializado com 0,5 para jogadores com valores menores que 6; 0,25 para jogadores com valores entre 6 e 9; e 0,15 para jogadores com valores iguais ou superiores a 10.

Outras constantes e variáveis utilizadas nas fórmulas são: a constante t que corresponde a 20 (decimal); a variável X que é a diferença entre a última pontuação e a pontuação média multiplicada pela constante K (decimal) e a variável Y que é a função da diferença entre o valor do jogador e um valor constante de 20 (decimal). VA corresponde ao valor atual do jogador e NV ao novo valor do jogador também em decimal.

Esses valores são definidos empiricamente pela equipe responsável com base na experiência adquirida ao longo do tempo e podem variar dependendo do contexto em que a função é utilizada. A funcionalidade do jogo depende diretamente dessa determinação já que esses valores têm um impacto direto no cálculo do preço do jogador.

5. RESULTADOS E DISCUSSÕES

Como mencionado anteriormente, a etapa inicial de um desenvolvimento *back-end* consiste em realizar o levantamento de todos os requisitos necessários e a escolha das melhores tecnologias para o desenvolvimento do software, sendo necessário definir o comportamento e as restrições operacionais que o produto deve atender. Este por sua vez deve possuir quatro atributos essenciais: facilidade de manutenção caso seja necessário fazer alguma alteração ou atualização; confiança, ou seja, precisa ser seguro protegendo dados sensíveis e não deve ocasionar danos físicos ou econômicos; precisa ser eficiente, possuindo rapidez de resposta e no tempo de processamento, além de otimizar a utilização de memória; e por fim facilidade de uso pelo usuário [29].

Isso foi alcançado por meio de escolhas estratégicas, como a tecnologia *Framework .NET* para o desenvolvimento de aplicativos, a adoção do *Design Pattern Mediator* para padrões do projeto, a utilização do *Amazon Web Services* como servidor em nuvem e a escolha do padrão de autenticação *JWT*.

Para testar as funcionalidades do sistema, a primeira etapa foi a verificação do *login*. Na figura 6 é possível ver o *JWT bearer token* gerado pelo *login* de uma aplicação. Sua composição é dividida em três partes separadas por pontos

("."): o *header* (cabeçalho), o *payload* (carga útil) e a assinatura [30].

Figura 6. Codificação do *token JWT Bearer*.

O diagrama mostra a estrutura de um token JWT Bearer dividido em três partes principais:

- CABEÇALHO (HEADER):** Um objeto JSON contendo:
 - "alg": "HS256"
 - "typ": "JWT"
- CARGA ÚTIL (Payload):** Um objeto JSON contendo:
 - "unique_name": "kaiodemr@gmail.com"
 - "role": "admin"
 - "nbf": 1687957093
 - "exp": 1687960693
 - "iat": 1687957093
- ASSINATURA:** Formada pela expressão:


```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
)
```

Fonte: Feito pelo autor

A parte em vermelho corresponde ao *Header*. O cabeçalho é um objeto JSON que contém informações sobre o tipo de *token* e o algoritmo de assinatura utilizado. É codificado em Base64Url e nessa estrutura o termo "alg" especifica o algoritmo de assinatura, sendo HMAC com SHA-256 (HS256), e o termo "typ" define o tipo de *token* como JWT [30].

A região roxa é a carga útil (*Payload*) que contém as informações adicionais do *token*, como os dados do usuário, permissões ou quaisquer outros dados relevantes para a aplicação. O *payload* também é um objeto JSON codificado em Base64Url. Pode ter qualquer estrutura definida pela aplicação, mas existem alguns campos comuns predefinidos. Neste exemplo, observa-se o campo "unique" que representa o identificador do usuário (*e-mail*), "role" para o tipo do usuário, "iat" para o momento em que o *token* foi emitido (em formato de *timestamp*) e "exp" para o momento em que o *token* expira [30].

A região azul é a assinatura, a parte final do *token JWT Bearer*. Ela é usada para verificar a integridade do *token* e garantir que não tenha sido alterado durante a transmissão. A assinatura é calculada usando o cabeçalho codificado, a carga útil codificada, uma chave secreta conhecida apenas pelo servidor e o algoritmo de assinatura especificado no cabeçalho.

O resultado é uma sequência de caracteres que pode ser usado para verificar a autenticidade [30].

Ao receber um *token JWT Bearer*, o servidor pode decodificá-lo, verificar a assinatura e obter as informações necessárias do *payload* para autenticar o usuário e autorizar as solicitações. Esse tipo de *token* é uma forma de autenticação usada em muitas aplicações web e APIs sendo baseado no padrão JSON, que é compacto, autocontido e seguro [31]. Ele é usado para transmitir informações de forma segura entre o cliente e o servidor. Também é importante ressaltar que o *token JWT Bearer* é apenas uma forma de autenticação e não deve ser usado como um substituto para a autorização adequada em nível de aplicação. O servidor ainda deve verificar se o usuário tem permissão para realizar determinadas ações antes de respondê-las [32].

Um critério importante de ressaltar é a definição do desempenho dos jogadores. Quando um jogador é desvalorizado, significa que sua pontuação em uma rodada foi menor que a sua média. Por outro lado, quando o jogador pontua acima da sua média, o seu valor aumenta por conta do seu desempenho. Dessa forma, os jogadores foram pontuados com base em ações executadas nas partidas reais, podendo haver variação no seu preço ao longo de todo o campeonato.

A segunda etapa para analisar a funcionalidade do jogo foi a avaliação do desempenho de três usuários distintos ao longo de três rodadas (tabela 2). É válido lembrar que cada participante começa com 36 moedas e com a pontuação zerada. Os valores iniciais de cada jogador foram definidos pela banca, sendo que calouros custam 5 moedas e veteranos 8.

Tabela 2. Desempenho dos usuários a cada rodada.

Rodadas	0		1		2		3		Pontuação Final
	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	
Usuário 1	0	R\$ 36,00	95,7	R\$ 52,97	122,4	R\$ 56,09	90,7	R\$ 51,79	308,8
Usuário 2	0	R\$ 36,00	28,8	R\$ 37,13	22,9	R\$ 31,34	25,6	R\$ 31,93	77,3
Usuário 3	0	R\$ 36,00	-3,9	R\$ 32,48	-20,9	R\$ 30,43	38	R\$ 47,19	13,2

Fonte: Feito pelo autor

O usuário número 1 apresentou o melhor desempenho em todas as disputas totalizando 308,8 pontos. Seu melhor resultado pode ser observado na rodada 2 onde ele atingiu a

pontuação mais alta (112,4), além de acumular a maior quantidade de moedas de todas as rodadas (56,09).

No panorama geral, o usuário 3 apresentou o pior resultado, começando com uma pontuação negativa logo na primeira disputa e finalizando com 13,2 pontos. Entretanto, na 3ª rodada o usuário 3 teve sua melhor pontuação (38,0) o que valorizou seu patrimônio o fazendo chegar a 47,19 moedas. Com isso este consegue escalar jogadores mais caros em busca de uma melhor pontuação, fazendo com que o mesmo não perca interesse no jogo.

Por outro lado, o usuário 2 apresentou o segundo melhor desempenho com uma pontuação final de 77,3, mantendo uma média entre cada rodada.

Em súmula foram obtidos as seguintes exemplificações: usuários com alto desempenho e, portanto, alta pontuação, sendo o caso do Usuário 1 que, apesar da queda na última rodada, obteve uma pontuação final alta e, por isso, suas moedas aumentaram, assim como descrito pela fórmula utilizada; usuários com médio desempenho, como é o caso do Usuário 2, visto que os valores de sua pontuação foram similares entre todas as rodadas; e, para concluir, o usuário com baixo desempenho, que é o caso do Usuário 3, exemplificando o resultado final de um participante abaixo da média e, portanto, com uma desvalorização.

Ao final de cada partida, todas as estatísticas individuais dos atletas foram contabilizadas. As pontuações de cada ação no jogo estão listadas na tabela 1 da metodologia e contemplam tanto o setor defensivo quanto ofensivo.

De acordo com Reis [33], os valores de cada jogador são flutuantes e podem ser comparados a bolsa de valores: comprar pelo menor preço possível, analisar o potencial de valorização de um bem, vender no momento certo e sair com lucro dessas operações.

Para a avaliação dessa funcionalidade, foram aferidos os resultados dos jogadores de quatro times durante três rodadas e estes dados estão demonstrados na tabela 3. Pode-se verificar condições em que o jogador é valorizado de acordo com o aumento do seu desempenho ao longo dos jogos, jogadores que

perdem seu valor por apresentar baixo rendimento e também aqueles que mantiveram uma pontuação estável.

Tabela 3. Desempenho dos jogadores a cada rodada.

Equipe 1									
Rodadas	0		1		2		3		Pontuação Final
	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	
Jogador 1	0	R\$ 5,00	19,4	R\$ 10,40	11,7	R\$ 8,55	15,2	R\$ 8,50	46,3
Jogador 2	0	R\$ 8,00	28,2	R\$ 11,48	26,2	R\$ 11,05	18,5	R\$ 9,11	72,9
Jogador 3	0	R\$ 8,00	24,2	R\$ 10,88	22,7	R\$ 10,54	15,5	R\$ 8,66	62,4
Jogador 4	0	R\$ 5,00	12,9	R\$ 7,96	47,4	R\$ 13,15	18,7	R\$ 11,19	79
Jogador 5	0	R\$ 5,00	11	R\$ 7,25	14,4	R\$ 7,79	22,8	R\$ 9,33	48,2
Equipe 2									
Rodadas	0		1		2		3		Pontuação Final
	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	
Jogador 1	0	R\$ 5,00	6	R\$ 5,38	3	R\$ 5,05	5,2	R\$ 5,31	14,2
Jogador 2	0	R\$ 5,00	5,4	R\$ 5,15	6	R\$ 5,37	5,4	R\$ 5,34	16,8
Jogador 3	0	R\$ 5,00	4,9	R\$ 4,99	6,3	R\$ 5,51	6	R\$ 5,66	17,2
Jogador 4	0	R\$ 5,00	6,4	R\$ 5,53	4,3	R\$ 5,30	3	R\$ 5,04	13,7
Jogador 5	0	R\$ 5,00	6,1	R\$ 5,41	3,3	R\$ 5,11	6	R\$ 5,59	15,4
Equipe 3									
Rodadas	0		1		2		3		Pontuação Final
	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	
Jogador 1	0	R\$ 8,00	6,1	R\$ 8,17	7	R\$ 8,30	-2	R\$ 7,05	11,1
Jogador 2	0	R\$ 8,00	5,3	R\$ 8,05	4,9	R\$ 7,99	12	R\$ 9,02	22,2
Jogador 3	0	R\$ 5,00	4,8	R\$ 4,98	4,2	R\$ 4,91	5	R\$ 5,10	14
Jogador 4	0	R\$ 5,00	6,2	R\$ 5,45	3,7	R\$ 5,18	4,8	R\$ 5,16	14,7
Jogador 5	0	R\$ 5,00	6,3	R\$ 5,49	3,9	R\$ 5,23	3,9	R\$ 5,09	14,1
Equipe 4									
Rodadas	0		1		2		3		Pontuação Final
	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	Pontuação	Moedas	
Jogador 1	0	R\$ 5,00	-1	R\$ 4,33	-7,2	R\$ 3,60	5	R\$ 7,33	-3,2
Jogador 2	0	R\$ 8,00	-2,2	R\$ 6,92	-3,4	R\$ 6,72	12	R\$ 9,18	6,4
Jogador 3	0	R\$ 5,00	-1,2	R\$ 4,30	-4,4	R\$ 3,93	6	R\$ 7,46	0,4
Jogador 4	0	R\$ 5,00	0,2	R\$ 4,46	-2,4	R\$ 4,16	10	R\$ 8,55	7,8
Jogador 5	0	R\$ 5,00	0,3	R\$ 4,47	-3,5	R\$ 4,03	5	R\$ 6,66	1,8

Fonte: Feito pelo autor

O jogador com melhor desempenho foi o Jogador 4 da equipe 1, alcançando uma pontuação final de 79. Este começou com um valor inicial de 5 moedas por se tratar de um calouro e na primeira rodada atingiu uma pontuação de 12,9. Ao avançar para a segunda rodada quase quadruplicou seus pontos em relação a partida anterior, somando 47,4, mas reduziu a 18,7 na última disputa. Mesmo assim, seu valor aumentou para 11,19 moedas, seguindo as fórmulas apresentadas na metodologia.

Isso acontece, pois, este jogador alcançou entre a primeira e a segunda rodada, um desempenho muito alto e, como descrito anteriormente, não conseguiu permanecer neste patamar. Desta forma, ao chegar na terceira partida, o jogador teve o decréscimo do seu desempenho e, portanto, de sua pontuação. Mesmo assim, ao final da última competição pode-se observar que sua pontuação final foi a maior quando comparada com todos os outros jogadores.

Por outro lado, foi também observado jogadores que perderam rendimento ao longo das rodadas. Este é o caso do Jogador 1 da Equipe 3. Este iniciou com um total de 8 moedas.

Entretanto, o jogador teve um desempenho abaixo do que já havia apresentado, decaindo seu preço para 7,05. Por fim, ao finalizar a terceira rodada, o jogador permaneceu nessa mesma pontuação.

Além disso, foi simulado também jogadores com desempenho mediano, sendo o caso por exemplo do Jogador 3 da Equipe 2. Seu desempenho e, portanto, sua pontuação, foram similares entre todas as rodadas, mantendo um padrão e uma baixa oscilação no número total de moedas final.

Como um demonstrativo para jogadores que finalizam as jogadas com pontuações negativas, temos o Jogador 1 da Equipe 4. Isso acontece quando o jogador tem um desempenho abaixo do esperado em uma partida, devido a ações cometidas ao longo da disputa, como por exemplo gol contra, cartão vermelho, cartão amarelo, gol sofrido (exemplificados na Tabela 1). Neste caso, o Jogador 1 obteve duas pontuações negativas em rodadas consecutivas, melhorando seu rendimento apenas na última partida, mas ainda assim, não sendo o suficiente para obter uma pontuação final positiva. Esta pontuação negativa do jogador atinge o rendimento do usuário como um todo, enfatizando a importância de escolhas estratégicas e táticas ao longo de todo o campeonato.

Porém, apesar do jogador 1 ter finalizado com uma pontuação negativa ele valorizou ao fim da terceira rodada. Isto se deve ao fato de na rodada 3 sua pontuação ser consideravelmente superior à sua média, resultando assim em uma valorização do atleta.

Analisando esses resultados foi verificado que as fórmulas de valorização estão funcionando conforme o esperado, uma vez que mesmo com a diferença de desempenho dos usuários, não foram observados valores muito elevados ou muito abaixo na quantidade de moedas evitando que ocorra disputas desequilibradas entre os participantes.

Além disso, outras rodadas com outras equipes de diferentes jogadores estão sendo simuladas, afim de possuir um N amostral maior e, portanto, conseguir mais comprovações referentes à credibilidade do programa.

Também foi possível observar que para o usuário aumentar sua pontuação e seu número de moedas, além do conhecimento técnico pelo esporte, no caso o futebol, é necessário conhecer estratégias de escalação e de formação tática com base em análises estatísticas e acompanhamento das rodadas. Por se tratar de jogos dinâmicos com jogadores reais, há oscilações não somente dentro do campo, mas de acordo com as vivências de cada jogador [17].

Além de toda a parte tática, jogos de *Fantasy Games* podem ser relacionados com a sorte, visto que, como já descrito, é necessário considerar a vida do jogador dentro e fora das quadras, fazendo com que a pontuação seja influenciada de acordo com os momentos de inspiração individual, decisões da arbitragem, lesões, entre outros [34].

Assim, para se obter sucesso no jogo, além do desempenho individual de cada jogador, é essencial que o usuário acompanhe todo o campeonato e todos os demais jogadores, desenvolvendo estratégias e acompanhando toda a estatística anunciada.

6. CONCLUSÕES

Este trabalho visou desenvolver um *fantasy game* de futebol baseado no torneio Intereps da cidade de São Carlos com intuito de melhorar a visibilidade da competição. Até o momento o software mostrou-se funcional, desde a etapa do *login*, até o cálculo da pontuação dos usuários e jogadores.

Os jogos estilo *Fantasy Game* promovem experiências interativas e significativas para o seu público alvo, permitindo a interação desse com a realidade e o virtual. A criação do site e o funcionamento do programa são essenciais para o alto alcance de usuários e, portanto, sua popularidade.

Para a criação do site desse jogo, foi necessário um trabalho meticuloso e cauteloso, de forma a desenvolver uma plataforma que permita a experiência completa ao usuário, sem contratempos e interferências, além de proporcionar uma prática de fácil uso e agradável.

Como perspectivas futuras é necessário a finalização da etapa do *front-end* possibilitando o acesso e a utilização dessa plataforma pelo público.

7 REFERÊNCIAS

- [1] FREIRE, Gabriel Gonçalves; GUERRINI, Daniel. Os Jogos na Sociedade Contemporânea: as Influências dos Avanços Tecnológicos. **Revista de Ensino, Educação e Ciências Humanas**, v. 17, n. 5, p. 463, 2016.
- [2] VISCONDI, G.; JUSTO, D.; GARCÍA, N. Aplicação de aprendizado de máquina para otimização da escalação de time no jogo cartola FC, 2017.
- [3] CASAGRANDE, M. C.; LAVARDA, S. D. L.; SILVEIRA, A. C. M. Fantasy game e reconfiguração da cultura desportiva. *Intexto*, n. 52, p. 95133, 14 maio 2021.
- [4] Febre dos fantasy games impulsiona crescimento do Arena 22. Disponível em: <<https://exame.com/bussola/febre-dos-fantasy-games-impulsiona-crescimento-do-arena-22/>>. Acesso em: 22 dez. 2022.
- [5] FORSBERG, M. M. An evaluation of .NET Object-Relational Mappers in relational databases Entity Framework Core and Dapper. Disponível em: <<https://www.diva-portal.org/smash/get/diva2:1687513/FULLTEXT01.pdf>>. Acesso em: 4 mai. 2023..
- [6] MACORATTI. 2020. **ADO .NET - Usando o Micro ORM Dapper .net**. Disponível em: https://www.macoratti.net/15/12/adn_dapper1.htm. Acesso em: 22 set. 2022.
- [7] PERSICO, V., et al. "On the Network Performance of Amazon S3 Cloud-Storage Service". 2016 5th IEEE International Conference on Cloud Networking (Cloudnet), IEEE, 2016, p. 113–18.
- [8] JONES, M.; HARDT, D. "RFC 6750: The OAuth 2.0 Authorization Framework: Bearer Token Usage.". Outubro de 2012
- [9] JONES, M., BRADLEY, J., SAKIMURA, N. "RFC 7519: JSON Web Token (JWT)". Maio de 2015
- [10] PALANKAR, M. R. et al. Amazon S3 for science grids: a viable solution? **Proceedings of the 2008 international workshop on Data-aware distributed computing**. Anais...Boston MA USA: ACM, 24 jun. 2008
- [11] AHMED, Salman; MAHMOOD, Qamar. An authentication based scheme for applications using JSON web token. In: **2019 22nd International Multitopic Conference (INMIC)**. Islamabad, Pakistan: IEEE, 2019, p. 1–6.

- [12] BALTA. 2019. **ASP.NET 5 – Autenticação e Autorização com Bearer e JWT**. Disponível em: <https://balta.io/blog/aspnet-5-autenticacao-autorizacao-bearer-jwt>. Acesso em: 23 set. 2022.
- [13] **Industry Demographics - Fantasy Sports & Gaming Association**. Fantasy Sports & Gaming Association - Connecting Fans to the Sports They Love. Disponível em: <https://thefsga.org/industry-demographics/>. Acesso em: 17 abr. 2023.
- [13] VERAS, Manoel. **Arquitetura corporativa de nuvem: amazon web services (AWS)**. 1. Ed. Rio de Janeiro: Brasport, 2013. Capítulo 1 e Capítulo 6.
- [14] INFOQ. 2020. **Criando aplicações .NET para AWS**. Disponível em: <https://www.infoq.com/br/articles/cloud-development-aws-sdk/>. Acesso em: 22 set. 2022.
- [15] JÁNOKY, László Viktor; LEVENDOVSKY, János; EKLER, Péter. An analysis on the revoking mechanisms for JSON Web Tokens. **International Journal of Distributed Sensor Networks**, v. 14, n. 9, p. 155014771880153, 2018.
- [16] **Industry Demographics - Fantasy Sports & Gaming Association**. Fantasy Sports & Gaming Association - Connecting Fans to the Sports They Love. Disponível em: <https://thefsga.org/industry-demographics/>. Acesso em: 17 abr. 2023.
- [17] SILVA, Marina Aragão. Cartola FC: fantasy game, mídia e engajamento do torcedor no futebol nacional. 2019. Disponível em: <https://repositorio.ufba.br/handle/ri/28294>. Acesso em: 5 jul. 2023.
- [18] GARETT, Renee; CHIU, Jason; ZHANG, Ly; *et al.* A Literature Review: Website Design and User Engagement. **Online journal of communication and media technologies**, v. 6, n. 3, p. 1–14, 2016
- [19] THAI, Thuan L.; LAM, Hoang Q. **NET framework essentials**. 3rd ed. Beijing ; Sebastopol, CA: O'Reilly, 2003.
- [20] ASCHE, Peter; THEIS, Sabine; BRÖHL, Christina; *et al.* Building and Exchanging Competence Interdisciplinarily: Design Patterns as Domain Mediator. **Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care**, v. 5, n. 1, p. 19–24, 2016
- [21] CUNNINGHAM, Kevin. Use of the Mediator Design Pattern in the LaSRS++ framework. *In: Modeling and Simulation Technologies Conference and Exhibit*. Portland,OR,U.S.A.: American Institute of Aeronautics and Astronautics, 1999.
- [22] MATHEW, Sajee; VARIA, J. Overview of amazon web services. **Amazon Whitepapers**, v. 105, p. 1-22, 2014.
- [23] MADURAPPERUMA, I. H., et al. State-of-Art Frameworks for Front-End and Back-End Web Development. agosto de 2022. ir.lib.seu.ac.lk, <http://ir.lib.seu.ac.lk/handle/123456789/6339>.
- [24] ANTUNES, Daniel. **DESENVOLVIMENTO BACK-END: UM GUIA PARA EMPREENDEDORES**. 2020. Disponível em: <https://www.gobacklog.com/blog/back-end-guia-para-empreendedores/>. Acesso em: 06 jun. 2023.
- [25] LANTHALER, Markus; GÜTL, Christian. On using JSON-LD to create evolvable RESTful services. *In: Proceedings of the third international workshop on RESTful design*. 2012. p. 25-32.
- [26] **Visual Studio: IDE e Editor de Código para Desenvolvedores de Software e Teams**. Visual Studio. Disponível em: <https://visualstudio.microsoft.com/pt-br/> Acesso em: 17 de maio de 2023.
- [27] BILLWAGNER. **Um tour por C# – Visão geral**. Disponível em: <https://learn.microsoft.com/pt-br/dotnet/csharp/tour-of-csharp/>. Acesso em: 5 jun. 2023.
- [28] GROUP, PostgreSQL Global Development. **PostgreSQL**. PostgreSQL. Disponível em: <https://www.postgresql.org/>. Acesso em: 17 de maio de 2023.
- [29] ALMEIDA, Leonardo. Desenvolvimento de um aplicativo WEB para o gerenciamento de redes de telecomunicações utilizando o Rational Unified Process como processo de engenharia de software. 2009.
- [30] RAHMATULLOH, A; GUNAWAN, R; NURSUWARS, F M S. Performance comparison of signed algorithms on JSON Web Token. **IOP Conference Series: Materials Science and Engineering**, v. 550, n. 1, p. 012023, 2019.
- [31] PEIXOTO, Rui Filipe Vilaça Oliveira. **Gestao de Acesso a Dados no Back-end: Controle de Acesso Baseado em Funções e Geração Automática de Relatórios**. 2020. Tese de Doutorado. Universidade de Lisboa (Portugal).
- [32] JONES, Michael B.; BRADLEY, John; SAKIMURA, Nat. **JSON Web Token (JWT)**. [s.l.]: Internet Engineering Task Force, 2015. Disponível em: <https://datatracker.ietf.org/doc/rfc7519/>. Acesso em: 25 jun. 2023
- [33] REIS, Rafael. Entre o físico e o digital: uma análise do game Cartola FC. Dissertação (Mestrado em Comunicação). Faculdade Cásper Líbero, Programa de Mestrado em Comunicação, – São Paulo, 2015
- [34] BATISTA, A. R. DE C. A gamificação como recurso estratégico de marketing de conteúdo: estudo de caso do fantasy game Cartola FC. repositorio.uft.edu.br, 25 abr. 2018