

# Cartolático: um sistema que utiliza a técnica de raciocínio baseado em casos para recomendação de escalações no aplicativo Cartola FC

Pedro Feltrin<sup>1</sup>, Gustavo Bathu Paulus<sup>1</sup>, Edimar Manica<sup>1</sup>

<sup>1</sup> Instituto Federal de Educação, Ciência e Tecnologia  
do Rio Grande do Sul (IFRS)  
Ibirubá – RS – Brasil

pedrofeltrin47@gmail.com,

{gustavo.paulus | edimar.manica}@ibiruba.ifrs.edu.br

**Abstract.** *Cartola FC is a fantasy-style game that allows users to create their own virtual teams and select players and coaches participating in the Brazilian Soccer Championship to represent them in each round. The high number of variables associated with a soccer match and the short period of time between two rounds makes the analysis of the available athletes complex. In this context, this paper proposes the Cartolático system, which uses the case-based reasoning technique to recommend players at the moment of their selection. This is an artificial intelligence technique that resembles the way human beings think when solving problems. To validate the system two metrics were proposed: The assertiveness of the system, calculated using a function that normalizes the recommendation of players, and the accuracy of the system, calculated based on the average difference between the predicted scores and the scores actually obtained. The system obtained accuracy averages above the median value for 5 of the 6 available positions.*

**Resumo.** *O Cartola FC é um jogo do estilo fantasy que permite que os usuários criem suas próprias equipes virtuais e escalem jogadores e técnicos participantes do Campeonato Brasileiro de Futebol para representá-los a cada rodada. O alto número de variáveis ligados a uma partida de futebol e o curto espaço de tempo entre duas rodadas torna complexa a análise dos atletas disponíveis para escalação. Neste contexto, o presente artigo propõe o sistema Cartolático, que utiliza a técnica de raciocínio baseado em casos para recomendar jogadores no momento da escalação. Essa é uma técnica de inteligência artificial que se assemelha a maneira como os seres humanos pensam na resolução de problemas. Para validar a o sistema foram propostas duas métricas: A assertividade do sistema, calculada por meio de uma função que normaliza a recomendação de jogadores, e a precisão do sistema, calculada com base na diferença média entre as pontuações previstas e as pontuações realmente obtidas. O sistema obteve médias de precisão acima do valor mediano para 5 das 6 posições disponíveis.*

## 1. Introdução

O jogo eletrônico Cartola FC é um exemplo de um *fantasy game*, um gênero de jogos virtuais que envolve eventos da vida real que impactam o contexto do jogo (LO-MAX, 2009). No Cartola FC, os eventos estão relacionados ao Campeonato Brasileiro

de Futebol. O jogo permite que os usuários, conhecidos como cartoleiros, criem equipes virtuais escolhendo entre os atletas das 20 equipes de futebol que compõem a primeira divisão do campeonato. Lançado em 2004, o jogo foi criado e é mantido pelo grupo Globo (GLOBO, 2005).

Em cada rodada, os cartoleiros devem submeter uma escalação até o fechamento do mercado, que, a partir de 2022, ocorre um minuto antes do início da primeira partida da rodada. O usuário seleciona uma formação entre um conjunto de 7 opções pré-definidas e escolhe 11 atletas e um treinador. A Cartoleta, unidade monetária do jogo, é outro fator importante a ser considerado na hora de montar uma equipe. Cada atleta e treinador tem um custo que varia de acordo com o desempenho recente e, nas primeiras rodadas, com o histórico. Todo usuário começa com 100 Cartoletas e a única forma de aumentar esse montante é escalando atletas que valorizem após a conclusão da rodada. Portanto, a valorização/desvalorização de um jogador influencia diretamente o patrimônio dos usuários que o escalaram.

O patrimônio, no entanto, é apenas um fator secundário no sistema de classificação de usuários; o principal é a pontuação. Durante a partida, os atletas são avaliados pela equipe de *scouting*<sup>1</sup> do Cartola FC e recebem pontos por ações positivas realizadas ao longo da partida, assim como perdem pontos por ações negativas. Alguns exemplos de ações positivas são gols, assistências, chutes para fora ou na trave, entre outros. Ações negativas incluem o recebimento de um cartão (sendo vermelho resulta em uma perda maior de pontos), um pênalti perdido e um passe errado.

A pontuação é um elemento fundamental no jogo Cartola FC, pois é através dela que os usuários são classificados em diferentes competições conhecidas como ligas. Além da liga Nacional, criada pelo próprio Cartola FC e a qual todo usuário participa assim que cria sua equipe no aplicativo, os usuários também podem criar suas próprias ligas, permitindo a formação de grupos de amigos ou de outros usuários com interesses em comum. O objetivo dos usuários é escalar uma equipe que alcance a maior pontuação possível, garantindo assim uma classificação elevada nas ligas em que participam.

As previsões no universo esportivo são complicadas, pois muitas variáveis afetam o contexto de um atleta, de uma equipe, ou de um campeonato (BUNKER; THABTAH, 2019). No Cartola FC, essa complexidade é ainda maior, pois o usuário deve levar em conta o orçamento e o impacto de diferentes resultados no seu capital. Além disso, a formação da equipe é um fator determinante e o curto espaço de tempo durante a abertura do mercado de escalações complica ainda mais o processo. Diante disso, uma recomendação automatizada de atletas e de técnico que leve em conta o orçamento do usuário pode facilitar e tornar mais ágil o processo de definição da equipe (FIALHO; MANHÃES; TEIXEIRA, 2019).

Nesse contexto, este artigo propõe o sistema Cartolático, que utiliza uma abordagem baseada em inteligência artificial para recomendação de jogadores e técnicos para o Cartola FC. Utilizou-se a técnica de raciocínio baseado em casos (*Case-based Reasoning* – CBR) como uma forma de simular o raciocínio humano na tomada de decisão de escalação. A utilização de CBR permite adaptar o sistema às características específicas de cada

---

<sup>1</sup>Processo de observação e análise que tem como principal objetivo a obtenção do máximo de informações sobre cada jogador

usuário, resultando em recomendações mais personalizadas e eficientes. Além disso, foram propostas duas métricas para avaliar as recomendações no contexto do Cartola FC: A assertividade do sistema na recomendação de jogadores e treinadores e a precisão dessas recomendações.

A assertividade do sistema é representada por um valor numérico entre 0 e 1 que é concedido à cada caso recomendado para uma consulta. Esse valor é calculado por meio de uma função normalizadora que leva em consideração a pontuação obtida por esse caso e o seu percentual de similaridade com o novo caso. A precisão é representada pela diferença absoluta entre a pontuação prevista e a pontuação realmente obtida. Os experimentos realizados mostram que o sistema foi eficiente na recomendação para 5 das 6 posições disponíveis, que apresentaram uma média de assertividade acima do valor mediano de 0,5.

O restante do artigo está organizado da seguinte forma: Na Seção 2, são apresentadas as tecnologias utilizadas. Na Seção 3, são comparados os trabalhos relacionados com o sistema Cartolático. Na Seção 4, é descrito o sistema Cartolático. Na Seção 5, são apresentadas detalhadas as duas métricas propostas. Na Seção 6, são apresentados os experimentos realizados e discutidos os resultados. Por fim, na Seção 7, são apresentadas as conclusões do trabalho e sugestões para trabalhos futuros.

## **2. Tecnologias Utilizadas**

Nesta seção, são apresentadas as tecnologias utilizadas para o desenvolvimento do trabalho, além dos conceitos teóricos necessários para a compreensão do contexto desta pesquisa.

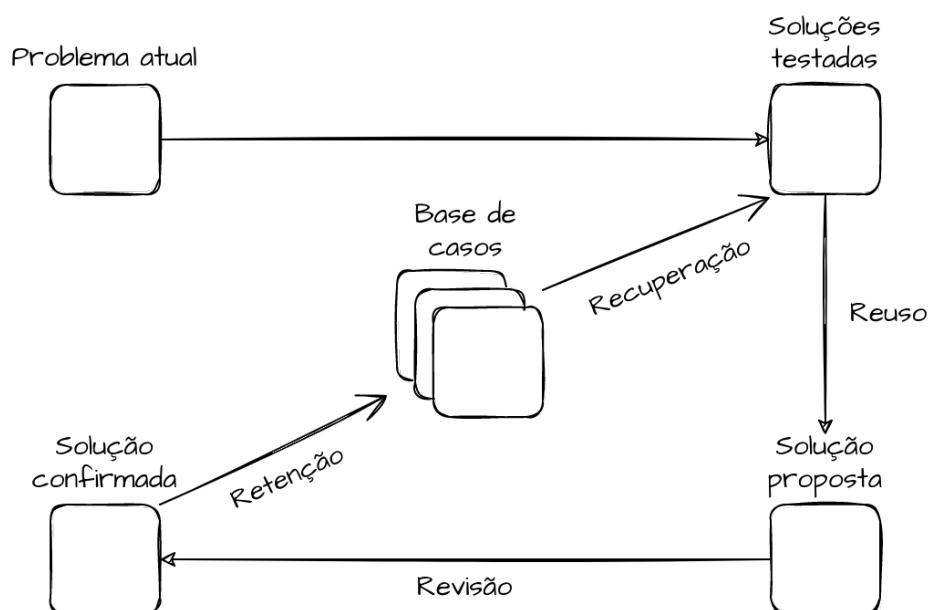
### **2.1. Raciocínio baseado em casos**

O CBR é um paradigma de inteligência artificial e ciência cognitiva que modela o processo de raciocínio de uma forma baseada primariamente em experiências passadas (LEAKE, 2015). O CBR imita o raciocínio de um humano e tenta encontrar respostas para novos problemas por meio da reutilização de soluções que foram aplicadas em problemas semelhantes no passado (AVRAMENKO; NYSTRÖM; KRASLAWSKI, 2004).

De acordo com López-Fernández et al. (2011), a técnica CBR possui um ciclo bem definido de funcionamento dividido em quatro etapas: recuperação, reuso, revisão e retenção. A Figura 1 ilustra as quatro etapas, os dados e as bases utilizadas no processo do CBR. Um novo problema é apresentado e o sistema busca na base de casos anteriores a soluções utilizadas para os problemas mais similares (recuperação). Essa solução é adaptada caso necessária e apresentada como possível solução para o novo problema (reuso). A solução é então analisada (revisão), podendo ser descartada, caso contrário é armazenada na base de casos (retenção) onde fica disponível para consultas futuras. As quatro etapas são detalhadas nas subseções seguintes.

#### **2.1.1. Recuperação**

O propósito dessa etapa é vasculhar a base de problemas passados e selecionar um ou mais que se assemelhem ao novo problema, juntamente com as soluções utilizadas para

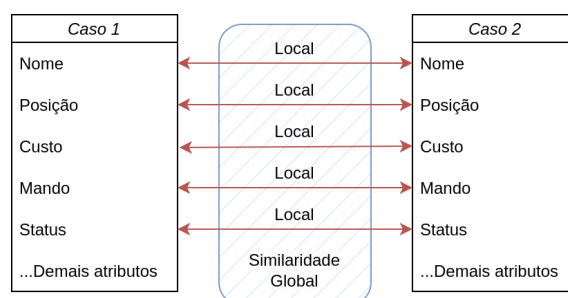


**Figura 1. Representação do ciclo de vida do CBR**

esses casos. Casos considerados bons são aqueles que possuem potencial para realizar previsões relevantes acerca do novo caso, ou seja, que tenham propriedades em comum, sem precisar de muita adaptação (KOLODNER, 2014).

Para tornar possível a etapa de recuperação, deve ser definido um conjunto de ações e de atributos respectivos a essas ações na hora de buscar na base de casos. As ações representam as buscas que devem ser feitas, e os atributos determinam o que deve ser comparado nessas buscas. Na hora de comparar esses atributos existem diferentes técnicas que determinam a similaridade entre os casos. A técnica dos vizinhos mais próximos é a mais tradicional para a recuperação de casos similares (PAULUS et al., 2020).

O processo de recuperação é dividido em duas etapas de computação de similaridade: local e global. Os objetivos desses dois cálculos de similaridade são, respectivamente: calcular a semelhança entre os atributos dos casos passados e do novo caso de forma individual, e calcular a semelhança geral entre os casos para determinar o quão útil a resolução passada é para o novo caso.



**Figura 2. Diferentes cálculos de similaridade**

Como pode ser observado na Figura 2, a similaridade local é calculada entre um mesmo atributo comum aos dois casos, já a similaridade global é calculada considerando

o resultado de todos os cálculos de similaridade local realizados.

A computação de similaridade local pode ser calculada por uma grande variedade de funções que atendem diferentes objetivos, dependendo da aplicação (PAULUS et al., 2020). As funções também podem variar conforme o domínio dos atributos. Atributos textuais requerem funções diferentes de atributos numéricos. A seguir são descritas duas funções de similaridade local: *equal* e *interval*.

$$equal_i(C_1, C_2) = \begin{cases} 1, & \text{if } C_1 = C_2 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

A função *equal* verifica se o valor de um determinado atributo  $i$  é igual em dois casos, conforme a Equação 1. Essa função recebe dois argumentos,  $C_1$  e  $C_2$ , que representam o valor de um mesmo atributo para os dois casos sendo comparados. O retorno dessa função é sempre 0 ou 1, onde 0 significa que os valores comparados são diferentes e 1 significa que são iguais.

$$interval_i(C_1, C_2) = 1 - \left( \frac{|C_1 - C_2|}{MAX\_VALUE} \right) \quad (2)$$

A função *interval* (RECIO-GARCÍA; GONZÁLEZ-CALERO; DÍAZ-AGUDO, 2014) calcula a similaridade entre atributos numéricos levando em consideração a proximidade do valor de um determinado atributo  $i$ , conforme a Equação 2. Essa função recebe dois argumentos,  $C_1$  e  $C_2$ , que representam o valor de 1 mesmo atributo para os dois casos sendo comparados. A função normaliza a diferença de dois valores pelo maior valor do atributo  $i$  considerando todos os casos da base ( $MAX\_VALUE$ ).

A computação de similaridade global visa agregar todas as similaridades locais computadas em um único valor de similaridade. A Equação 3 apresenta a função de similaridade global *average*, que é calculada a partir da média ponderada da similaridade local de cada atributo, onde  $q$  é um novo caso,  $c$  é um caso antigo sendo comparado,  $A$  é o conjunto de todos os atributos,  $i$  é o atributo sendo analisado,  $|A|$  é o número de atributos,  $L_i(q_i, c_i)$  é a similaridade local entre o caso novo e o caso antigo para o atributo  $i$ ,  $q_i$  é o valor do atributo  $i$  para o caso novo,  $c_i$  é o valor do atributo  $i$  para o caso antigo e  $W_i$  é o peso do atributo  $i$ . A função  $L_i$  pode ser *equal*, *interval*, entre outras. O valor retornado por  $L_i$  é um valor numérico entre 0 e 1, onde 0 significa 0% de similaridade e 1 significa 100%.

$$G(q, c) = \sum_{i \in A} \frac{L_i(q_i, c_i) \times W_i}{|A|} \quad (3)$$

A atribuição dos pesos é feita com base em análises experiências de usuários do jogo Cartola FC. Isso garante que os pesos reflitam de maneira precisa a importância relativa de cada atributo na previsão da pontuação dos jogadores. Além disso, os pesos também podem ser ajustados ao longo do tempo, de acordo com as mudanças nas condições do jogo e nas preferências dos usuários.

### 2.1.2. Reuso

Dos casos recuperados na etapa anterior, são extraídas as possíveis soluções para o novo problema. Diferentes políticas de reuso podem ser aplicadas para escolher a solução para o novo problema com base nos casos similares recuperados (PAULUS et al., 2020). Dentre as variadas políticas de reuso, em casos onde existe uma alta similaridade entre os problemas, é comum adaptar a solução utilizada pelo caso mais similar (PAULUS; ASSUNCAO; SILVA, 2019).

Reusar essas soluções pode ser relativamente simples se a solução para um caso bastante similar for recuperada. Caso contrário, podem ser aplicadas transformações aos casos a fim de torná-los mais parecidos ao novo caso em questão. Existem diferentes estratégias de adaptação de casos, em que é possível escolher quantos e quais atributos devem ser adaptados.

### 2.1.3. Revisão

Se uma solução for falha, isto é, que não condiz com o novo problema, ela deve ser corrigida, ou descartada (SLADE, 1991). Nessa fase, pode ser obtido um relatório relativo à solução construída até o momento. Esse relatório pode ser dado na forma de uma avaliação de acerto referente ao resultado ou no formato de um caso revisado manualmente corrigido. O caso revisado entra na base de casos do sistema CBR para as próximas recomendações.

Caso necessário, a solução é revisada por um usuário, ou por um conjunto de regras construídas a partir do conhecimento do mesmo. Em aplicações comerciais que usam o CBR, esse usuário costuma ser um especialista (PAVÓN et al., 2001). O trabalho de Pavón et al. (2001) propõe um método para automatizar a etapa de revisão de sistemas CBR.

### 2.1.4. Retenção

A fase de retenção é a fase de aprendizado do sistema CBR. A forma típica de aprendizado que ocorre no sistema CBR é aprender ao adicionar um caso revisado à base de casos. Dessa forma, a nova solução torna-se disponível para reuso em futuros problemas. Esse aumento contínuo de casos na base, contudo, causa uma constante queda na eficiência do processo de recuperação. Existem trabalhos que estudam a retenção seletiva de casos, como apresentado em Moral et al. (2020).

O trabalho de Moral et al. (2020) aponta a retenção seletiva como uma possível solução para utilização em bases de casos não, ou parcialmente, organizadas previamente. No contexto do estudo de Moral et al. (2020) foi utilizada a estratégia ACE (POWELL; HAUFF; HASTINGS, 2005). A Elicitação Automática de Casos (ACE) é o processo de usar programas de computador para extrair informações de um texto e criar um caso que pode ser usado no CBR. O objetivo da ACE no CBR é automatizar o processo de criação de casos, reduzindo o tempo e o esforço necessários para revisão manual enquanto aumenta a precisão e a consistência das informações extraídas.

## 2.2. jCOLIBRI

O jCOLIBRI (UCM, 2004) é um *framework* orientado a objetos para o desenvolvimento de sistemas de CBR que foi criado pelo Departamento de Engenharia de Software e Inteligência Artificial da Universidade Complutense de Madri. Por se tratar de um *framework* consolidado para o desenvolvimento de aplicações de raciocínio baseado em casos (RECIO-GARCÍA; GONZÁLEZ-CALERO; DÍAZ-AGUDO, 2014), esse trabalho utilizou o *framework* jCOLIBRI.

A arquitetura do jCOLIBRI consiste em duas camadas: *design* e desenvolvimento. A primeira fornece ferramentas que guiam os usuários pelo processo de configuração e explicam o comportamento dos componentes dentro da aplicação. A camada de desenvolvimento fornece componentes Java básicos necessários para a criação de uma aplicação CBR da perspectiva de um desenvolvedor (BERGMANN; KOLODNER; PLAZA, 2005).

## 3. Trabalhos Relacionados

Nesta seção, são apresentados quatro trabalhos que visam o mesmo objetivo do sistema Cartolático, ou seja, recomendar jogadores para o Cartola FC através da aplicação de inteligência artificial. Porém, esses trabalhos utilizam diferentes técnicas e tecnologias. Esses trabalhos foram encontrados na plataforma Google Acadêmico através de uma busca por “Cartola FC” e foram selecionados com base na análise do título e resumo, verificando se possuíam relação com a aplicação de técnicas de inteligência artificial.

No trabalho de Viscondi, Justo e García (2017) foram utilizados os algoritmos *Extreme Gradient Boosting* (XGBoost) (CHEN et al., 2015), *Random Forest* (RF) (BIAU; SCORNET, 2016) e *Support Vector Machine* (SVM) (NOBLE, 2006) como métodos de predição de escalações. A técnica de aprendizado de máquina *k-means* foi escolhida para fazer o treinamento da inteligência artificial. A base de dados utilizada por Viscondi, Justo e García (2017) foi a mesma disponibilizada pelo próprio Cartola FC. Os dados referentes às 28 primeiras rodadas do Campeonato Brasileiro de Futebol do ano de 2017 foram utilizados para o treinamento da aplicação. Da 29ª a 35ª rodadas a aplicação foi testada e os resultados das previsões de escalações foram validados. Por fim, os resultados obtidos por Viscondi, Justo e García (2017) foram, segundo o autor, satisfatórios somente após a aplicação da técnica de clusterização.

O segundo trabalho analisado, de Ribeiro et al. (2019), foi fundamentado na utilização de uma rede neural e na classificação das entidades a serem consideradas. Além disso, utilizou uma base de dados contendo dados das rodadas da edição de 2014 até a edição de 2018 do campeonato brasileiro, além das 5 primeiras rodadas da edição de 2019, ano de realização do trabalho. Uma das escalações sugeridas pelo trabalho de Ribeiro et al. (2019) para a 7ª rodada conseguiu uma pontuação de 64,37 pontos, que foi aproximadamente 4% mais alta que a média de todos os cartoleiros para aquela rodada, que foi de 62,10, segundo o *blog* do Cartola FC.

O terceiro trabalho analisado, de Mota, Coimbra e Peixoto (2018), focou na utilização de estratégias de seleção por modelos aleatórios e estatísticos. Essas estratégias foram utilizadas para construir um modelo de análise e visualização dos dados baseado em simulação. A abordagem estatística adotada por Mota, Coimbra e Peixoto (2018) considerou os seguintes atributos na hora de escalar atletas: média de pontuação dos atletas,

posições mais valiosas, entre outros aspectos encontrados nos conjuntos de dados, a fim de tentar maximizar as chances de sucesso na escalação das equipes durante as rodadas. Os conjuntos de dados utilizados possuem representação dos anos de 2014 a 2017.

O quarto e último trabalho analisado, de Cruz, Sousa e Calcada (2022), teve como objetivo principal validar o uso de técnicas de extração de conhecimento em bases de dados para identificar padrões de bom desempenho no futebol brasileiro e criar um modelo do perfil do jogador ideal para escalar, tendo abrangido as posições de goleiro e atacante. Para isso, foi utilizada a técnica de Inteligência Artificial Explicável e uma base contendo dados referentes as edições de 2014 a 2021, exceto 2015. A técnica de Redes de Regras de Associação Filtradas foi utilizada como principal meio de estruturar e entender os padrões da base de dados e técnicas de *Knowledge-discovery in Databases* (KDD) (FRAWLEY; PIATETSKY-SHAPIO; MATHEUS, 1992) foram utilizadas para extrair conhecimento da base de dados que pudesse auxiliar na tomada de decisão na escalação de jogadores durante o jogo.

A Tabela 1 apresenta uma análise comparativa dos trabalhos relacionados, incluindo também o sistema Cartolático. Nessa tabela, observa-se que três dos quatro trabalhos relacionados, além do sistema Cartolático, consideraram o patrimônio do usuário para a recomendação. O trabalho de Cruz, Sousa e Calcada (2022) foi o único que não considerou todas as posições disponíveis, tendo utilizado somente as posições de goleiro e atacante, diferentemente do sistema Cartolático e dos outros trabalhos analisados, que analisaram todas as posições. Os trabalhos de Viscondi, Justo e García (2017) e Ribeiro et al. (2019) utilizaram a técnica de aprendizado de máquina no desenvolvimento de suas soluções.

**Tabela 1. Análise comparativa dos trabalhos relacionados**

Itens comparativos	Viscondi et al. 2017	Ribeiro et al. 2019	Mota et al. 2018	da Cruz et al. 2022	Cartolático
Técnica(s) utilizada(s)	Aprendizado de máquina	Aprendizado de máquina	Equações matemáticas	Inteligência Artificial Explicável	Raciocínio baseado em casos
Algoritmo(s) utilizado(s)	Extreme Gradient Boosting (XG), Random Forest (RF) e Support Vector Machine (SVM)	redes neurais e algoritmos genéticos	seleção aleatória e seleção estatística	Redes de Regras de Associação Filtradas	Nearest Neighbor (NN), funções de similaridade local (equal, interval) e global (average)
Tamanho da base	28 rodadas	190 rodadas	152 rodadas	304 rodadas (presumidamente)	38 rodadas
Posições abrangidas	todas	todas	todas	GOL e ATA	todas
Considerou o patrimônio	sim	sim	sim	não	sim
Temporada analisada	2017	2019	2018	2022	2022
Avaliou os resultados	parcialmente, comparando entre os próprios algoritmos testados	parcialmente	parcialmente, comparando a diferença obtida por diferentes valores utilizados para um mesmo atributo e a diferença entre os dois algoritmos utilizados	sim, com um objetivo diferente	sim

A maior diferença entre o CBR, cuja utilização é proposta no sistema Cartolático e técnicas como aprendizagem de máquina e redes neurais é que o CBR se baseia no uso de experiências passadas ou casos armazenados, enquanto outras técnicas se baseiam em modelos e padrões estatísticos para fazer previsões ou tomar decisões. A seleção estatística, utilizada por Mota, Coimbra e Peixoto (2018), por outro lado, está focada na identificação do melhor modelo estatístico a partir de um conjunto de dados. Além disso, o CBR é frequentemente usado para tarefas de solução de problemas, enquanto a aprendizagem de máquina e as redes neurais são técnicas usadas primariamente para tarefas de classificação.



Em comparação com as técnicas escolhidas por Viscondi, Justo e García (2017), Mota, Coimbra e Peixoto (2018), Ribeiro et al. (2019) e Cruz, Sousa e Calcada (2022) no desenvolvimento de seus trabalhos, o CBR se mostra uma abordagem diferente e que possibilita implementar agentes virtuais capazes de tomar decisões utilizando capacidades mais parecidas com as que um ser humano utilizaria. Também não necessita tantos dados em seu treinamento como técnicas de redes neurais para apresentar um resultado inicial satisfatório.

Percebe-se também que todos os autores validaram seus trabalhos, mesmo que de forma parcial, sendo que Ribeiro et al. (2019), por exemplo, avaliou o desempenho em uma única rodada. Outro diferencial no desenvolvimento do sistema Cartolático foi a proposta de duas diferentes métricas, capazes de validar a assertividade geral e pontual das recomendações, além de comparar com outras técnicas de IA. As métricas utilizadas foram: o *ranking* normalizado do jogador recomendado e a diferença entre a pontuação obtida e a pontuação esperada normalizada. Essas métricas são explicadas em maior detalhe na seção a seguir.

#### 4. Sistema Cartolático

O sistema Cartolático é uma ferramenta que utiliza técnicas de inteligência artificial para auxiliar usuários do jogo Cartola FC na montagem de suas equipes por meio da recomendação de jogadores e técnicos. A partir desse ponto passa a utilizar o termo “jogadores” para jogadores e técnicos, afim de facilitar a leitura. A Figura 3 mostra o fluxo de funcionamento do sistema, desde a entrada de dados por parte do usuário ao retorno dos jogadores recomendados.

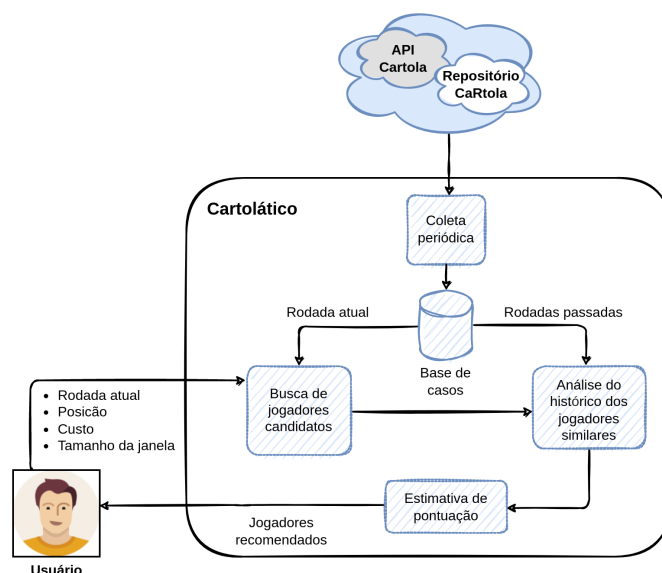


Figura 3. Sistema Cartolático

A cada rodada do campeonato que é finalizada, o sistema Cartolático busca, utilizando um programa semeador (*seeder*)<sup>2</sup>, os dados do mercado da rodada ativa e os salva

<sup>2</sup>Um semeador de banco de dados é uma ferramenta ou *script* que preenche um banco de dados com

na base de casos, além de atualizar os dados da rodada anterior, como a pontuação obtida por cada jogador e técnico. Esses dados são buscados diretamente da API oficial do Cartola FC (GLOBO, 2005).

O usuário fornece ao sistema um conjunto de informações que são usadas como parâmetros em diferentes momentos do ciclo de vida da aplicação. Dentre os parâmetros informados pelo usuário para consulta estão: a rodada desejada, a posição buscada, o orçamento aproximado disponível e o tamanho da janela de custo. O sistema então busca pelos jogadores de rodadas passadas que mais se assemelhem aos casos buscados na primeira etapa.

Na etapa de busca de jogadores candidatos o sistema vasculha a base de casos procurando por aqueles que sejam da rodada e posição informadas, e que estejam dentro do orçamento disponível considerando a janela informada. O sistema então busca os  $k$  casos mais similares na etapa de análise do histórico dos jogadores similares. Essas duas etapas buscam o caso diretamente da base de casos, que é atualizada periodicamente com os dados buscados pelo *seeder*. Na etapa de estimativa de pontuação, o sistema Cartolático faz o cálculo da pontuação prevista para cada jogador candidato. Esses jogadores são então retornados ao usuário por ordem de pontuação prevista.

#### 4.1. Busca de jogadores candidatos

O objetivo dessa etapa é buscar na base de casos todos os jogadores que satisfaçam os parâmetros fornecidos pelo usuário, ou seja, posição e custo. Esses são chamados de jogadores candidatos. O sistema busca na base de casos todos os jogadores disponíveis para a rodada informada e filtra os jogadores encontrados, mantendo apenas aqueles que possuam o status de “provável” para a rodada em questão.

Em seguida, o sistema filtra ainda mais a lista de jogadores, selecionando apenas aqueles que atuam na posição informada e que tenham um custo compatível com os parâmetros (custo e tamanho da janela) informados pelo usuário.

O Algoritmo 1 descreve a busca por jogadores e candidatos a recomendação. A entrada é composta por quatro parâmetros  $R$ ,  $P$ ,  $B$  e  $W$ , que representam, respectivamente: a rodada desejada, a posição desejada, o custo estimado e o tamanho da janela de custo. A rodada pode ser qualquer valor numérico inteiro de 1 à 38 (número de rodadas do campeonato), enquanto a posição pode ser uma das seis disponíveis no Cartola FC (goleiro, zagueiro, lateral, meia, atacante e técnico). O custo estimado é um número racional com no máximo uma casa decimal. Por fim, a janela é um número inteiro qualquer que representa a margem de erro tolerada para o custo.

As linhas 1 e 2 declaram as constantes e variáveis necessárias, que são: *cases*, que representa a lista de todos os casos (jogadores) na base; *candidates*, que representa a lista de casos candidatos à recomendação que é retornada ao final da função. A linha 3 percorre todos os casos na base e armazena o caso sendo iterado na variável  $C$ . A linha 4 verifica se o caso satisfaz a posição e rodada informadas. A linha 5 verifica se o caso possui *status* de “provável”. Nas linhas 6 e 7 o custo do caso  $C$  é comparado com o

---

um conjunto de dados predefinidos, normalmente usado para testar ou inicializar um banco de dados com alguns dados de base. Ele permite aos desenvolvedores inserir facilmente dados em um banco de dados, sem a necessidade de inseri-los manualmente.

---

**Algorithm 1** Algoritmo de busca de jogadores candidatos
 

---

**Input:** R, P, B, W**Output:** candidates

```

1: cases ← all cases retrieved from the casebase
2: candidates ← []
3: for C in cases do
4:   if  $C[\textit{position}] = P$  and  $C[\textit{round}] = R$  then
5:     if  $C[\textit{status}] = \textit{provavel}$  then
6:       if  $C[\textit{price}] \geq B - W$  and  $C[\textit{price}] \leq B + W$  then
7:         candidates ← [...candidates, C]
8:       end if
9:     end if
10:  end if
11: end for
12: return candidates

```

---

custo estimado informado pelo usuário e, caso se encontre dentro do intervalo da janela, é adicionado à lista de jogadores candidatos denominada *candidates*. Por fim, a linha 12 retorna a lista *candidates*.

#### 4.2. Análise do histórico de jogadores similares

O objetivo dessa etapa é iterar sobre a lista de jogadores candidatos e aplicar a técnica de raciocínio baseado em casos para obter os  $k$  casos mais similares, ou seja, os  $k$  jogadores mais similares a cada um deles. Essa similaridade é calculada a partir da comparação de um conjunto pré-definido de características comuns aos dois casos, como *status* do jogador, adversários, e outros. A similaridade entre o mesmo atributo de dois casos diferentes é definida por um valor entre 0 e 1, onde 0 significa nenhuma semelhança e 1 significa que os valores são idênticos.

Cada um dos atributos comparados no raciocínio baseado em casos (CBR) tem um peso específico que é definido no momento de execução e que sempre pode ser alterado. Isso é feito para refletir a importância relativa de cada atributo na previsão da pontuação dos jogadores. Por exemplo, um atributo como o *status* do jogador pode ter um peso maior, pois é um fator importante que influencia diretamente sua pontuação.

O Algoritmo 2 descreve o a busca por jogadores similares. A entrada *candidate\_case* é um dos casos candidatos retornados no Algoritmo 1. A linha 1 define a constante  $k$  que determina o número de casos similares retornados. A linha dois inicializa uma lista *cases* com todos os casos na base. As linhas 3 e 4 iniciam as listas de casos passados e casos similares (*previous\_cases* e *similar\_cases*). No bloco que vai da linha 5 à linha 9 são incluídos na lista *previous\_cases* apenas os casos de rodadas passadas que possuem a mesma posição do caso candidato. Em seguida, nas linhas 10 à 12 é calculada a similaridade entre cada caso prévio e o caso candidato armazenando em *similar\_cases*

#### 4.3. Política de reuso baseada na estimativa de pontuação

Nesta etapa é apresentada a política de seleção de casos utilizada na etapa de reuso do no trabalho proposto. Esta política está baseada em uma análise que tem como

**Algorithm 2** Algoritmo de busca de jogadores similares**Input:** candidate\_case**Output:** similar\_cases

---

```

1:  $k \leftarrow 20$ 
2:  $cases \leftarrow$  all cases retrieved from the casebase
3:  $previous\_cases \leftarrow []$ 
4:  $similar\_cases \leftarrow []$ 
5: for  $C$  in  $cases$  do
6:   if  $C[position] = candidate\_case[position]$  and  $C[round] <$ 
      $candidate\_case[round]$  then
7:      $previous\_cases \leftarrow [...previous\_cases, C]$ 
8:   end if
9: end for
10: for  $P$  in  $previous\_cases$  do
11:    $similar\_cases \leftarrow G(candidate\_case, similar\_case)$   $\triangleright$  Conforme Equação 3
12: end for
13:  $similar\_cases \leftarrow OrdenaLista(similar\_cases)$ 
14: return  $similar\_cases$  limit  $k$ 

```

---

objetivo estimar a pontuação de um jogador candidato a partir da pontuação prévia dos casos mais similares e da similaridade desses casos com o novo caso. Essa estimativa é feita conforme a Equação 4, onde  $j$  é um caso (jogador) candidato,  $S$  é uma lista com os casos mais similares ao jogador candidato,  $G(i, j)$  é a similaridade global entre o jogador candidato e o caso similar  $i$  calculada conforme 3,  $\|S\|$  é a quantidade de casos similares e  $P_i$  é a pontuação que o caso similar obteve.

$$P(j, S) = \sum_{i \in S} \frac{G(i, j) \times P_i}{|S|} \quad (4)$$

Para exemplificar a estimativa, considere a Tabela 2, que apresenta 4 casos sendo que o caso 4 é um caso candidato e os casos 1-3 são os casos mais similares. Neste caso, a pontuação estimada do caso 4 seria 7,22. Destaca-se que essa pontuação é calculada a partir da pontuação dos casos 1, 2 e 3, sendo que a pontuação do caso 1 tem mais influência porque tem maior similaridade com o caso candidato.

**Tabela 2. Exemplo de casos**

Caso	Nome	Rodada	Adversário	Mando	Custo	Similaridade	Pontuação Prevista	Pontuação Obtida
1	Terans	17	Internacional	C	9,57	98%	-	14,50
2	Eduardo	32	Internacional	C	10,81	92%	-	8,20
3	Jair	1	Internacional	C	9,00	91%	-	-0,10
4	Alê	35	Internacional	C	9,69	-	7,22	4,70

## 5. Métricas Propostas

Para avaliar a eficácia das recomendações do sistema Cartolático, foram propostas duas métricas: assertividade, que analisa a posição do jogador sugerido dentre todos os candidatos para uma determinada consulta e a precisão, que analisa a diferença entre a pontuação realmente obtida e a pontuação prevista pelo sistema Cartolático. Essas métricas são detalhadas nas subseções seguintes.

### 5.1. Assertividade

A primeira métrica utilizada foi a assertividade, que consiste na posição do jogador no grupo ordenado por pontuação obtida. No entanto, como algumas posições possuem mais jogadores disponíveis do que outras, foi necessário realizar um processo de normalização do ranking para tornar a comparação mais precisa. Para isso, foi utilizada a Equação (5).

$$\text{Assertividade} = \frac{(n - r + 1)}{n} \quad (5)$$

onde  $n$  é o número de jogadores da mesma posição e com custo dentro da janela para a rodada informada e  $r$  é a posição que o jogador recomendado ficou em um *ranking* ordenado de forma decrescente pela posição realmente obtida. Jogador recomendado é o jogador de mesma posição que obteve a maior pontuação estimada normalizada na rodada. Dessa forma, foi possível avaliar o quanto o jogador sugerido pelo sistema foi assertivo em relação aos demais jogadores sugeridos. Essa equação sempre resulta em um valor que vai de 0 à 1, sendo que quanto mais próximo o resultado for de 1 maior a assertividade, pois 1 significa que ficou no topo do *ranking*.

A Tabela 3 apresenta os 4 primeiros casos retornados para uma consulta por jogadores de meio-campo com orçamento de 8 Cartoletas para a rodada 35. Os casos da tabela estão ordenados pela pontuação que obtiveram ao final da rodada. A assertividade obtida pelo jogador recomendado utilizando a Equação 5 foi de 0,75.

**Tabela 3. Exemplo de casos com assertividade normalizada**

Caso	Nome	Rodada	Adversário	Mando	Custo	Pontuação Prevista	Pontuação Obtida	Assertividade
1	André	35	Ceará	F	7,57	3,32	6,70	-
2	Alê	35	Internacional	C	9,69	7,22	4,70	0,75
3	Lucas Evangelista	35	Avaí	F	9,69	3,29	2,80	-
4	Lucas Sasha	35	Palmeiras	F	7,64	4,49	0,20	-

### 5.2. Precisão

A segunda métrica, precisão, é calculada pela diferença entre a pontuação obtida e a prevista pelo sistema Cartolático, é calculada com base na média ponderada dos casos mais similares. Essa métrica permite avaliar o quanto a pontuação obtida pelo sistema foi próxima da pontuação prevista, o que indica a precisão das recomendações do sistema.

Essa diferença é representada por um valor racional qualquer onde valores mais próximos de 0 indicam previsões melhores.

$$\text{Precisão} = |R - E| \quad (6)$$

onde  $R$  é a pontuação realmente obtida pelo jogador recomendado e  $E$  é a pontuação estimada pelo sistema Cartolático para o jogador na rodada.

Na Tabela 3 podem ser observadas as pontuações previstas e as pontuações realmente obtidas para cada caso. Os casos 1 e 4 obtiveram uma diferença entre pontuação prevista e pontuação obtida de menos de 0,5, enquanto o caso 3 teve uma diferença de 3,38.

## 6. Validação Experimental

Para avaliar a eficácia do sistema Cartolático para recomendação de jogadores no Cartola FC, foi realizada uma série de experimentos em diferentes condições. O sistema foi desenvolvido e testado em um ambiente de configuração específico, utilizando bases de dados de reais.

Nas subseções a seguir, são apresentados os detalhes dos experimentos realizados, assim como os resultados obtidos em relação à posição no ranking e ao ganho de pontos. A subseção 6.1 descreve a base de dados utilizada. A subseção 6.2 especifica a configuração do ambiente utilizada para o desenvolvimento e os testes. Por fim, na subseção 6.3 são apresentados os experimentos realizados e discutidos os resultados.

### 6.1. Bases de Dados

A base de dados utilizada para o desenvolvimento e testes do sistema Cartolático foi obtida a partir de duas fontes: a API oficial do Cartola FC (GLOBO, 2005) e o repositório *git* intitulado “caRtola” (HENRIQUEPGOMIDE, 2022), disponível no *GitHub*. Os dados da API do Cartola FC são obtidos em formato JSON, enquanto os dados do repositório são obtidos em formato CSV.

Os dados buscados no repositório *git* caRtola (HENRIQUEPGOMIDE, 2022) incluem registros de jogadores em mercados de rodadas passadas, que já não são mais disponibilizados pela API do Cartola FC (GLOBO, 2005). Esses dados incluem informações sobre todos os jogadores aptos para escalação naquela rodada. Na API do Cartola FC (GLOBO, 2005), por outro lado, também são disponibilizados dados sobre as 20 equipes que participam do campeonato e sobre as partidas realizadas em cada rodada. Os registros de partidas, diferentemente dos dados de mercado, são disponibilizados para todas as rodadas.

Para garantir a integridade e a consistência dos dados, foi necessário realizar um processo de normalização, que envolveu a padronização dos campos e a conversão dos dados para o formato JSON. O processo de normalização foi realizado de forma automatizada, utilizando scripts em JavaScript, para garantir a eficiência e a rapidez do processo. Os dados normalizados foram então armazenados em um banco de dados MySQL. O processo se deu nas seguintes etapas:

1. Ler os dados das duas fontes (API do Cartola FC (GLOBO, 2005) e repositório caRtola (HENRIQUEPGOMIDE, 2022))

2. Converter os dados para o mesmo formato JSON
3. Padronizar os campos de acordo com o formato escolhido
4. Armazenar os dados normalizados no banco de dados

Na realização dos experimentos foram utilizados dois conjuntos de dados: os dados das partidas disputadas em cada rodada e os dados do mercado de escalação. Esses dados foram fundamentais para que o sistema pudesse avaliar o desempenho das equipes e dos atletas, bem como o potencial de pontuação de cada jogador sugerido. Os dados das partidas incluem informações sobre o time mandante e o time visitante, o local de disputa e o desempenho recente das equipes.

Os dados do mercado de escalação incluem informações sobre os atletas, como, por exemplo: nome, clube, preço, entre outras. Esses dados são importantes para que o sistema possa avaliar o potencial de pontuação de cada atleta e sugerir jogadores mais assertivos. A pontuação média e a pontuação na última rodada são especialmente relevantes, pois indicam o desempenho recente dos atletas e podem indicar um possível aumento ou diminuição de pontuação na rodada seguinte.

## 6.2. Ambiente de Testes

Para desenvolver e testar o sistema Cartolático, foi necessário criar um ambiente de trabalho específico. Para isso, foi utilizada a ferramenta Java de código aberto OpenJDK na versão 11.0.17, que forneceu a linguagem de programação Java e as ferramentas necessárias para o desenvolvimento. Além disso, foi utilizado o framework jColibri na versão 3.2, que forneceu os recursos para a implementação da técnica de raciocínio baseado em casos.

Para armazenar e gerenciar os dados utilizados pelo sistema, foi necessário utilizar um sistema de gerenciamento de banco de dados. Neste caso, foi escolhido o MySQL 8, que foi instalado em um ambiente Docker isolado para garantir a estabilidade e segurança do sistema.

Os experimentos foram executados em um laptop Asus ROG Strix G731GU-BI7N9 17.3", com um processador Intel Core i7 9750H, 16GB de memória RAM, placa de vídeo NVIDIA GeForce GTX 1660 Ti, SSD NVme de 512GB, tendo como sistema operacional principal o Zorin OS 16 Kernel 5.15.0-53-generic, uma distribuição Linux baseada em Ubuntu.

A fim de efetuar os testes, foram designados os pesos e as funções de similaridade local apresentados na Tabela 4.

A Tabela 4 apresenta as funções de similaridade local utilizadas para cada atributo, bem como o peso de cada atributo para o cálculo da função de similaridade global. Na Tabela 4, é possível perceber foram utilizados diferentes funções de similaridade local para diferentes atributos. Os atributos posição, *status*, custo e oponente são comparados utilizando a função *equal* (Equação 1). Por outro lado, para comparar a similaridade entre o custo dos casos foi utilizada a função *interval* (Equação 2), que calcula a proximidade entre os dois em valores em um intervalo  $n$ , uma vez que os jogadores têm custos variados e raramente serão iguais. Para o cálculo da similaridade global foi utilizada a função *average* (Equação 3).

**Tabela 4. Funções de similaridade local e pesos utilizados para cada atributo**

Atributo	Função de similaridade local	Peso
Posição	<i>Equal</i>	1,00
Status	<i>Equal</i>	1,00
Custo	<i>Interval</i>	1,00
Mando de campo	<i>Equal</i>	1,00
Oponente	<i>Equal</i>	1,00

### 6.3. Experimentos

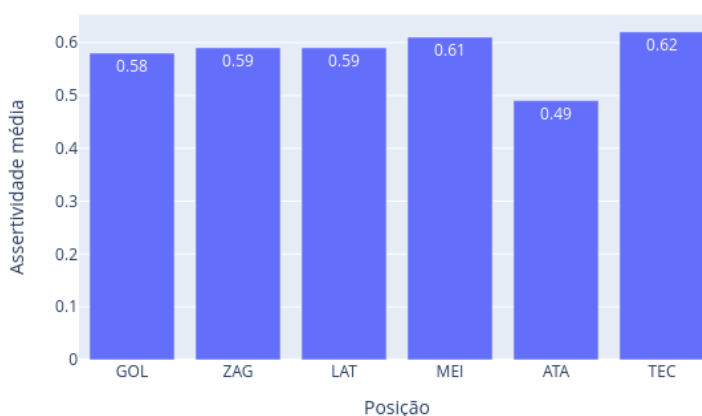
O objetivo desta subseção é avaliar a eficácia do CBR para recomendar jogadores para o jogo virtual Cartola FC. Para isso, foram realizadas consultas teste para todas as 38 rodadas do Campeonato Brasileiro de 2022. Foram consideradas todas as posições possíveis, que são: goleiro (GOL), zagueiro (ZAG), lateral (LAT), meia (MEI), atacante (ATA) e técnico (TEC). Foram testados três valores de orçamento diferentes: 4, 8 e 12 Cartoletas com tamanho de janela 2. Com esses três diferentes orçamentos e uma janela de tamanho dois foi possível cobrir todos os jogadores com custo de 2 a 14 Cartoletas, o que representa 78,94% de todos os jogadores disponíveis na base de casos. As consultas realizadas resultaram em um total de 7546 recomendações de jogadores, que foram avaliadas utilizando as métricas propostas neste artigo.

#### 6.3.1. Análise de assertividade

A Figura 4 apresenta a assertividade média para cada posição disponível (GOL, ZAG, LAT, MEI, ATA, TEC). A assertividade média foi calculada para todos os jogadores recomendados para os 3 orçamentos ao longo das 38 rodadas. Observa-se que 5 das 6 as posições obtiveram um ranking médio que ultrapassa o valor mediano de 0,5, exceto para os atacantes, cujo assertividade média foi de 0,49. Um valor de 0,5 significa que em um ranking de  $n$  elementos, o jogador recomendado ficou acima da metade.

A única consulta cuja assertividade média ficou abaixo de 0,5 foi a de atacantes. Tal fato pode ser explicado pelo alto número de jogadores disponíveis para essa posição, o que dificulta acertar a posição exata do jogador recomendado, mas não significa que o jogador recomendado não possa ter sido eficaz.

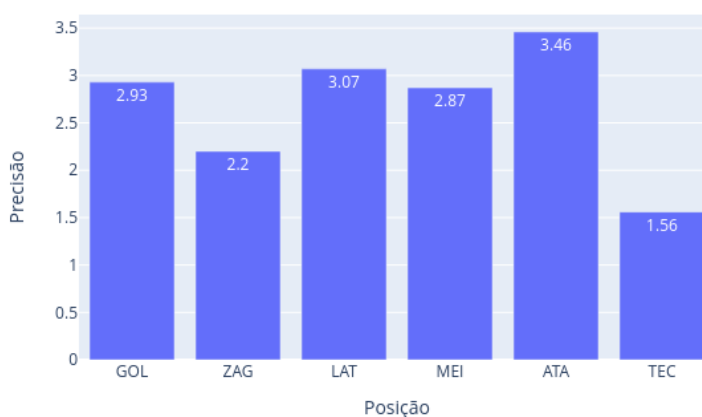




**Figura 4. Assertividade por posição.**

### 6.3.2. Análise da precisão

A Figura 5 apresenta a precisão média para cada posição disponível. A precisão média foi calculada para todos os jogadores recomendados para os 3 orçamentos testados durante as 38 rodadas. Neste gráfico, valores mais próximos a 0 indicam índices mais altos de precisão. Dessa forma, é possível perceber que as recomendações de técnicos se destacaram por terem previsto pontuações muito próximas as pontuações realmente obtidas, com uma diferença de 1,56. Por outro lado, nota-se que os atacantes apresentaram um desempenho inferior, com uma média de diferença de 3,46.



**Figura 5. Precisão média para cada posição**

### 6.3.3. Análise do comportamento para diferentes custos

A Figura 6, apresenta a precisão média para cada posição disponível e cada valor de orçamento (4, 8 e 12 cartoletas) testado. A média foi calculada considerando todas as rodadas. Das 18 combinações testadas, somente 2 obtiveram diferenças médias menores ou maiores que 1, são elas: GOL com orçamento de 4 Cartoletas e MEI com orçamento de 12 Cartoletas. Isso significa que a diferença entre a pontuação prevista e a pontuação obtida pode ser considerada boa em mais de 85% das consultas.

Não parece haver um padrão claro entre a diferença média e o orçamento para nenhuma posição. Contudo, é possível perceber que há maior facilidade do sistema Cartolático em fazer recomendações para algumas posições, como ZAG e TEC, que tiveram valores próximos a 1 para todos os orçamentos testados. As previsões para LAT e ATA também se mostraram homogêneas para todos os orçamentos, porém com uma diferença média de pontuação maior. Por outro lado, as previsões realizadas para as posições de GOL e MEI mostram as maiores diferenças entre um orçamento e outro, sendo que o valor absoluto da diferença entre a média das previsões com 12 cartoletas e 4 cartoletas foi, para cada posição, respectivamente, 1,28 e 1,01.

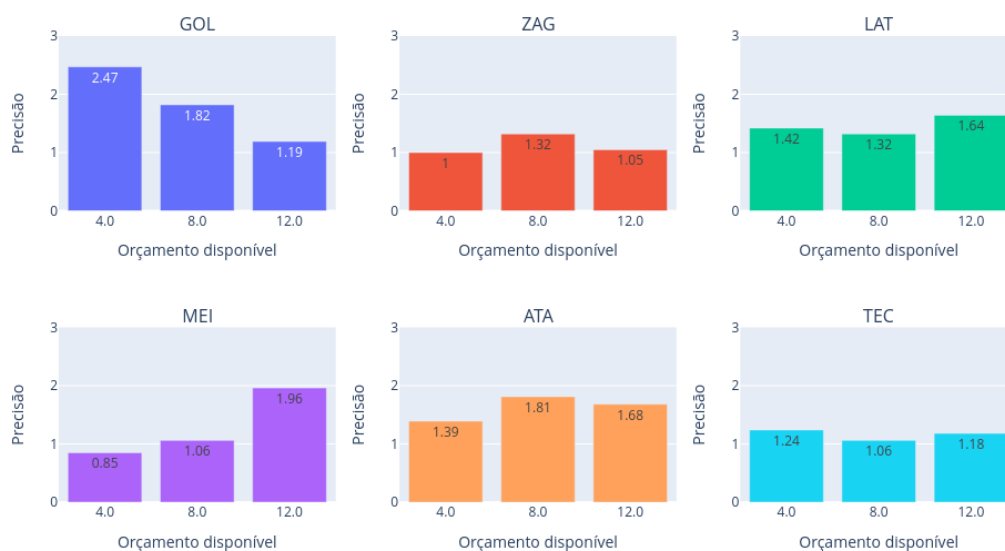
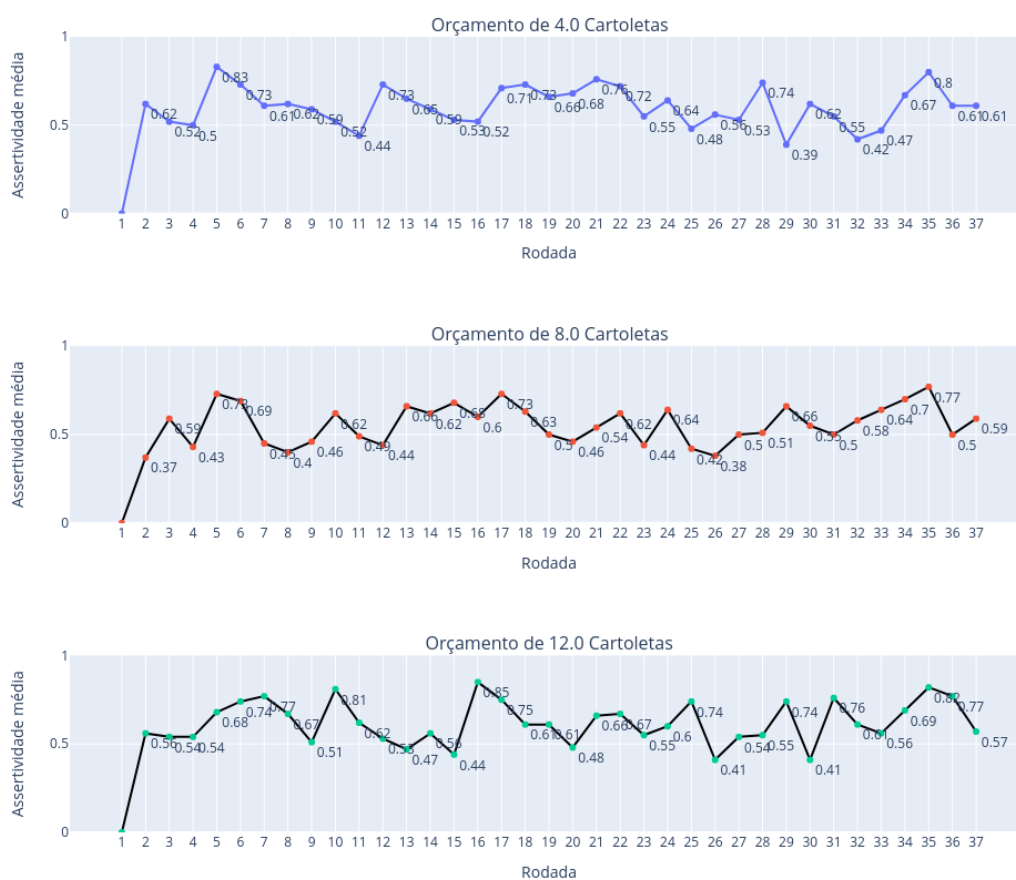


Figura 6. Precisão média

### 6.3.4. Análise do comportamento a cada rodada

Por fim, a Figura 7 mostra a assertividade média em cada rodada para cada orçamento. A média foi calculada incluindo todas as posições testadas. É possível perceber que a utilização da técnica de CBR foi mais eficaz na recomendação de jogadores para orçamentos inferiores a 12 cartoletas durante o campeonato. Para cada jogador recomendado, foi calculado seu ranking entre todos os jogadores candidatos para a rodada dentro

do orçamento fornecido. Este ranking foi normalizado utilizando a posição do jogador recomendado em relação a todos os demais, que foram retornados em uma lista ordenada pela pontuação obtida. Para os orçamentos de 4 Cartoletas foram realizadas somente 5 recomendações que tiveram assertividade média inferior a 0,5, enquanto para os orçamentos de 8 e 12 Cartoletas foram, respectivamente, 11 e 5. Isso sugere que a técnica de CBR foi capaz de produzir recomendações de jogadores mais bem-sucedidas com orçamentos abaixo de 6 e acima de 12 Cartoletas.



**Figura 7. Assertividade em cada rodada.**

## 7. Conclusão

Neste artigo, foi proposto o sistema Cartolático que utiliza a técnica CBR para recomendar jogadores para o jogo virtual Cartola FC. O sistema busca comparar dados de jogadores disponíveis em rodadas anteriores à rodada atual para buscar e fazer recomendações de jogadores e treinadores com bom potencial de pontuação na nova rodada. Esses dados são tratados como problemas e o sistema Cartolático tenta resolvê-los buscando as soluções utilizadas para problemas passados que sejam similares, e o faz de uma maneira que se assemelha muito à forma humana de se resolver problemas. O trabalho também propôs uma política de reuso para a etapa de recuperação, que utiliza a similaridade e a

pontuação obtida pelos casos similares para selecioná-los. Além disso, foram propostas duas métricas para avaliação dos experimentos, com fim de avaliar a assertividade e a precisão das recomendações.

Os experimentos realizados neste estudo demonstraram que a utilização da técnica de CBR foi eficaz na identificação de um conjunto de jogadores com potencial promissor para a formação de escalões. O sistema Cartolático conseguiu atingir médias positivas de recomendações assertivas para mais de 80% das situações, tendo se sobressaído em consultas realizadas para rodadas mais próximas do fim do campeonato. O sistema Cartolático também conseguiu realizar previsões precisas de pontuação em 88% das situações testadas.

Como trabalhos futuros, sugere-se a expansão dos experimentos realizados e a comparação com os feitos no presente artigo. Além disso, sugere-se o teste com as seguintes modificações e inclusões:

- Incluir dados de edições passadas do Cartola FC na base de casos utilizados pelo CBR para aumentar o número de casos disponíveis e refinar as semelhanças entre os casos, levando a recomendações mais precisas e personalizadas para cada jogador.
- Ajustar os pesos atribuídos aos atributos utilizados.
- Incluir outros atributos internos e externos como informações sobre as condições da partida, o retrospecto recente de cada equipe e posição das equipes no ranking da CBF.
- Utilizar outros tamanhos de janela.
- Considerar o uso de métricas diferentes para avaliar a precisão das recomendações do CBR em diferentes situações.
- comparar a política de reuso proposta nesse trabalho com políticas de reuso convencionais como a recomendação do jogador ou técnico mais similar dentre os recuperados e a recomendação do caso similar que retornou maior pontuação.
- Utilizar as métricas de avaliação criadas para escolher quando persistir um novo caso e comparar o resultado utilizando as bases geradas.

## Referências

AVRAMENKO, Y.; NYSTRÖM, L.; KRASLAWSKI, A. Selection of internals for reactive distillation column—case-based reasoning approach. *Computers Chemical Engineering*, v. 28, n. 1, 2004. Escape 12.

BERGMANN, R.; KOLODNER, J.; PLAZA, E. Representation in case-based reasoning. *The Knowledge Engineering Review*, v. 20, n. 3, 2005.

BIAU, G.; SCORNET, E. A random forest guided tour. *Test*, v. 25, n. 2, p. 197–227, 2016.

BUNKER, R. P.; THABTAH, F. A machine learning framework for sport result prediction. *Applied Computing and Informatics*, v. 15, n. 1, 2019.

CHEN, T. et al. Xgboost: extreme gradient boosting. *R package version 0.4-2*, v. 1, n. 4, p. 1–4, 2015.

- CRUZ, D. L. da; SOUSA, J. V. M.; CALCADA, D. Utilizando inteligencia artificial explicável para formação de perfil de jogadores no cartola fc. *Conjecturas*, v. 22, n. 12, p. 968–985, 2022.
- FIALHO, G.; MANHÃES, A.; TEIXEIRA, J. P. Predicting sports results with artificial intelligence – a proposal framework for soccer games. *Procedia Computer Science*, v. 164, 2019.
- FRAWLEY, W. J.; PIATETSKY-SHAPIRO, G.; MATHEUS, C. J. Knowledge discovery in databases: An overview. *AI magazine*, v. 13, n. 3, p. 57–57, 1992.
- GLOBO. *Cartola FC*. 2005. Disponível em <<https://ge.globo.com/cartola/>>. Último acesso em 08/06/2022.
- HENRIQUEPGOMIDE. *CaRtola*. 2022. Disponível em <<https://github.com/henriquepgomide/caRtola/>>. Último acesso em 15/01/2023.
- KOLODNER, J. *Case-based reasoning*. [S.l.]: Morgan Kaufmann, 2014.
- LEAKE, D. Problem solving and reasoning: Case-based. *International Encyclopedia of the Social Behavioral Sciences: Second Edition*, p. 56–60, 2015.
- LOMAX, R. G. Fantasy sports: History, game types, and research. In: *Handbook of sports and media*. [S.l.]: Routledge, 2009. p. 416–425.
- LÓPEZ-FERNÁNDEZ, H. et al. *Using CBR as design methodology for developing adaptable decision support systems*. [S.l.]: USA: Intech Publishers, 2011.
- MORAL, R. C. et al. Investigating case learning techniques for agents to play the card game of truco. In: IEEE. *2020 19th Brazilian Symposium on Computer Games and Digital Entertainment (SBGames)*. [S.l.], 2020. p. 107–116.
- MOTA, E.; COIMBRA, D.; PEIXOTO, M. Cartola fc data analysis: Uma ferramenta para simulação, análise e visualização de dados para o fantasy game cartola-fc. In: ANAIS DO XIV SIMPÓSIO BRASILEIRO DE SISTEMAS DE INFORMAÇÃO, 2018, Caxias do Sul, Brasil. *Anais...* Porto Alegre, RS, Brasil: SBC, 2018.
- NOBLE, W. S. What is a support vector machine? *Nature biotechnology*, v. 24, n. 12, p. 1565–1567, 2006.
- PAULUS, G. B.; ASSUNCAO, J. V. C.; SILVA, L. A. L. Cases and clusters in reuse policies for decision-making in card games. In: IEEE. *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. [S.l.], 2019. p. 1361–1365.
- PAULUS, G. B. et al. *Casos e clusters no desenvolvimento de políticas de reuso para tomada de decisão em jogos de cartas*. Dissertação (Mestrado) — Universidade Federal de Santa Maria, 2020.
- PAVÓN, R. et al. Improving the revision stage of a cbr system with belief revision techniques. *Comput Inf Syst J*, v. 8, 2001.
- POWELL, J. H.; HAUFF, B. M.; HASTINGS, J. D. Evaluating the effectiveness of exploration and accumulated experience in automatic case elicitation. In: *Case-Based Reasoning Research and Development*. [S.l.: s.n.], 2005. p. 397–407.

- RECIO-GARCÍA, J. A.; GONZÁLEZ-CALERO, P. A.; DÍAZ-AGUDO, B. jcolibri2: A framework for building case-based reasoning systems. *Science of Computer Programming*, v. 79, p. 126–145, 2014.
- RIBEIRO, L. E. d. S. et al. Predição de escalações para o jogo cartola fc utilizando aprendizado de máquina e otimização. 2019.
- SLADE, S. Case-based reasoning: A research paradigm. *AI Magazine*, v. 12, n. 1, p. 42, 1991.
- UCM. *jCOLIBRI*. 2004. Disponível em <<https://gaia.fdi.ucm.es/research/colibri/jcolibri/>>. Último acesso em 14/08/2022.
- VISCONDI, G.; JUSTO, D.; GARCÍA, N. Aplicação de aprendizado de máquina para otimização da escalação de time no jogo cartola fc. 2017.