



Associação Propagadora Esdeva  
 Centro de Ensino Superior de Juiz de Fora – CES/JF  
 Curso de Bacharelado em Sistemas de Informação  
 Trabalho de Conclusão de Curso – Artigo

---

## CRIAÇÃO DE WEBSITE MEMORIAL ESTATÍSTICO DO TUPI FC, ALIMENTADO POR ROBÔ AUTÔNOMO

*Bruno Goulart Mosse*<sup>1</sup>

*Igor Costa de Oliveira Obeica*<sup>2</sup>

*Leonardo Mageste Da Cruz Herédia*<sup>3</sup>

*Matheus Medeiros Campos*<sup>4</sup>

*Othon Brück*<sup>5</sup>

*Centro de Ensino Superior de Juiz de Fora, Juiz de Fora, MG*

*Daves Martins*<sup>6</sup>

*Centro de Ensino Superior de Juiz de Fora, Juiz de Fora, MG*

### RESUMO

Hoje em dia, apesar de vivermos na era da informação é muito difícil conseguir dados futebolísticos e institucionais dos clubes de menor expressão. Tendo esta dificuldade em mente este projeto surge visando reviver o passado, trazendo os dados das temporadas e títulos para os torcedores do Tupi FC, que em sua grande maioria, são pessoas de uma idade mais avançada.

A ideia principal deste projeto é disponibilizar para os torcedores do Tupi FC um site iterativo, que centralize essas informações antes dispersas na web. Com isso toda a comunidade de torcedores e fãs poderão ter acesso a informações do memorial do clube, tais como estatísticas, datas importantes entre outras.

**Palavras-chave:** Crawler. Linguagem Natural. Chatobot.

---

<sup>1</sup> Discente do Curso de Bacharelado em Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora – CES/JF. Endereço: Rua Halfeld 1.179 – 36.016-000 – Juiz de Fora – MG - Brasil. E-mail: m.goulartbruno@gmail.com

<sup>2</sup> Discente do Curso de Bacharelado em Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora – CES/JF. Endereço: Rua Halfeld 1.179 – 36.016-000 – Juiz de Fora – MG - Brasil. E-mail: cristiano\_pascholaim@hotmail.com

<sup>3</sup> Discente do Curso de Bacharelado em Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora – CES/JF. Endereço: Rua Halfeld 1.179 – 36.016-000 – Juiz de Fora – MG - Brasil. E-mail: cristiano\_pascholaim@hotmail.com

<sup>4</sup> Discente do Curso de Bacharelado em Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora – CES/JF. Endereço: Rua Halfeld 1.179 – 36.016-000 – Juiz de Fora – MG - Brasil. E-mail: matheusmedeiroscampos@gmail.com

<sup>5</sup> Discente do Curso de Bacharelado em Engenharia de Software do Centro de Ensino Superior de Juiz de Fora – CES/JF. Endereço: Rua Halfeld 1.179 – 36.016-000 – Juiz de Fora – MG - Brasil. E-mail: othon.bruck@hotmail.com

<sup>6</sup> Docente do Curso de Bacharelado em Sistemas de Informação do Centro de Ensino Superior de Juiz de Fora. Orientador.

## 1 INTRODUÇÃO

O Tupi Football Club, ou simplesmente Tupi-MG, é uma agremiação esportiva da cidade de Juiz de Fora, no estado de Minas Gerais. É um dos clubes mais importantes do Interior de Minas. Fundado em 26 de maio de 1912, o clube mineiro manda suas partidas no Estádio Municipal Radialista Mário Helênio, que possui capacidade liberada para 31.863 pessoas.

Este projeto surge da necessidade que os clubes pequenos do interior têm em manter seu passado vivo e de fácil acesso para seus torcedores. Hoje em dia, para um torcedor, é difícil conseguir informações sobre os títulos obtidos pela equipe, quais seus maiores artilheiros, quais jogadores com maior projeção nacional.

Essas informações estão dispersas pela internet, dificultando e muito a obtenção da mesma.

A ideia principal desse projeto é disponibilizar para os torcedores desses clubes um site iterativo, que centralize essas informações antes dispersas na web. Com isso toda a comunidade de torcedores e fã dos clubes poderão ter acesso a informações do memorial do clube, informações estatísticas do clube, datas importantes entre outras. O mais interessante desse projeto é que o site será alimentado por robôs autônomos que ficarão buscando e descobrindo essas informações pela web.

Hoje em dia, diversos sites disponibilizam informações de jogos, gols, resultados de diversos tipos de jogos e clubes. Tornar esses dados visível para o torcedor é um grande desafio.

Diversas estratégias devem ser usadas para coletar esses dados e mantê-los atualizados sem a necessidade da intervenção humana, por isso faz-se necessário a adoção de robôs autônomos que se encarregam de buscar informações para alimentar o website do clube.

Os robôs autônomos utilizarão técnicas de machine learning para descoberta de conhecimento em base de dados não convencionais, esses dados servirão para alimentar o website.

O Website proporcionará aos torcedores do Tupi um meio de manter vivo sua paixão pelo clube, descobrindo curiosidades. Os torcedores irão interagir com o website podendo realizar perguntas a respeito do seu clube de coração.

## 2 REFERENCIAL TEÓRICO

### Arquitetura Rest

A aplicação desenvolvida foi separada em duas partes, Front-end que é responsável por toda interface e interação com usuário e o Back-end responsável pelas funcionalidades do sistema. No back-end foi adotada uma arquitetura REST na qual permite a comunicação entre dados de aplicações através de requisições HTTP.

O termo REST de acordo com Roy Fielding (Fielding, 2000) em sua dissertação de mestrado *Architectural Styles and the Design of Network-based Software* é um acrônimo para “Transferência de Estado Representacional” (Representational State Transfer). A arquitetura REST enxerga cada aplicação da web como um aglomerado de recursos que representam um estado particular de uma aplicação, quando se acessa este recurso é transferido o estado(conteúdo) e talvez alterado. A maioria das aplicações REST utilizam o HTTP como protocolo de comunicação padrão e os recursos da aplicação são indicados nas URI's.

O protocolo HTTP(HyperText Transfer Protocol) é um protocolo cliente-servidor e é o protocolo mais utilizado na web para transferência de dados e informações. O cliente faz uma requisição(request) e então o servidor interpreta a requisição e retorna uma resposta(response) a solicitação feita pelo cliente. Dentro das requisições feitas pelo cliente, o HTTP permite criar, atualizar, pesquisar, executar e remover determinados recursos.

O HTTP define um conjunto de métodos de requisições, também chamados de “verbos”, no qual indicam qual a ação está sendo requisitada pelo cliente. Dentre os verbos os principais são:

- GET - Busca o conteúdo específico do recurso solicitado. Deve retornar somente dados.
- POST - Solicita um recurso específico, porém alterando o estado do recurso.
- PUT - Substitui os valores da requisição do recurso.
- DELETE - Remove um recurso específico.



Imagem 1- Fonte: <https://informatica.mercadolivre.com.br/componentes-pc/#menu=categories>

Conforme mostrado na imagem 1 temos o exemplo da aplicação do Mercado Livre que utiliza a arquitetura REST, o website do mercado livre é famoso devido à possibilidade de qualquer pessoa comprar e vender produtos ou serviços. Como podemos ver os recursos foram especificados através do menu do site e da categoria “Componentes para PC” (Imagem 2).

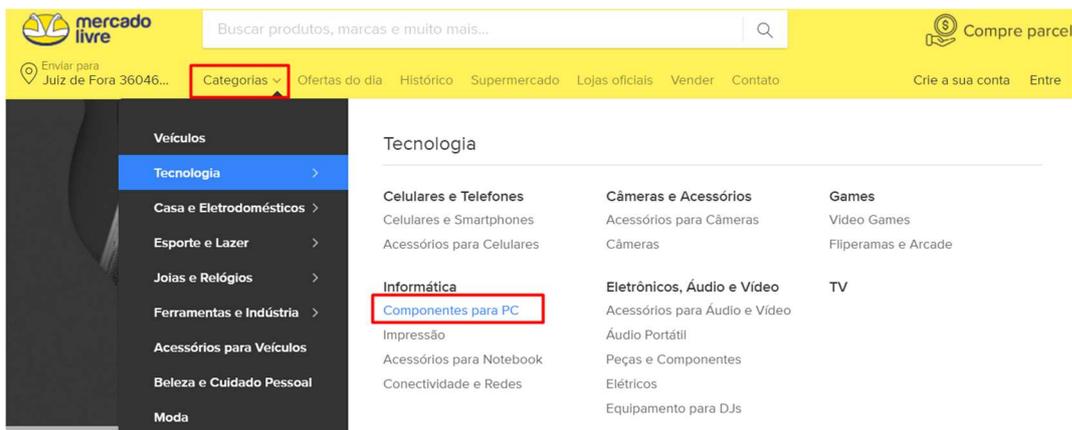


Imagem 2 - Fonte: <https://informatica.mercadolivre.com.br/componentes-pc/#menu=categories>

No projeto foram utilizados Web Crawlers (Augusto Verzbickas, 2013), ou simplesmente “Crawler”, robôs(algoritmos) que buscam dados de páginas na web através de requisições. Um exemplo claro de Web Crawlers são sites como Google e Bing, que a partir de um termo digitado realizam a busca em diversas páginas web procurando pelo termo e informando seu endereço.

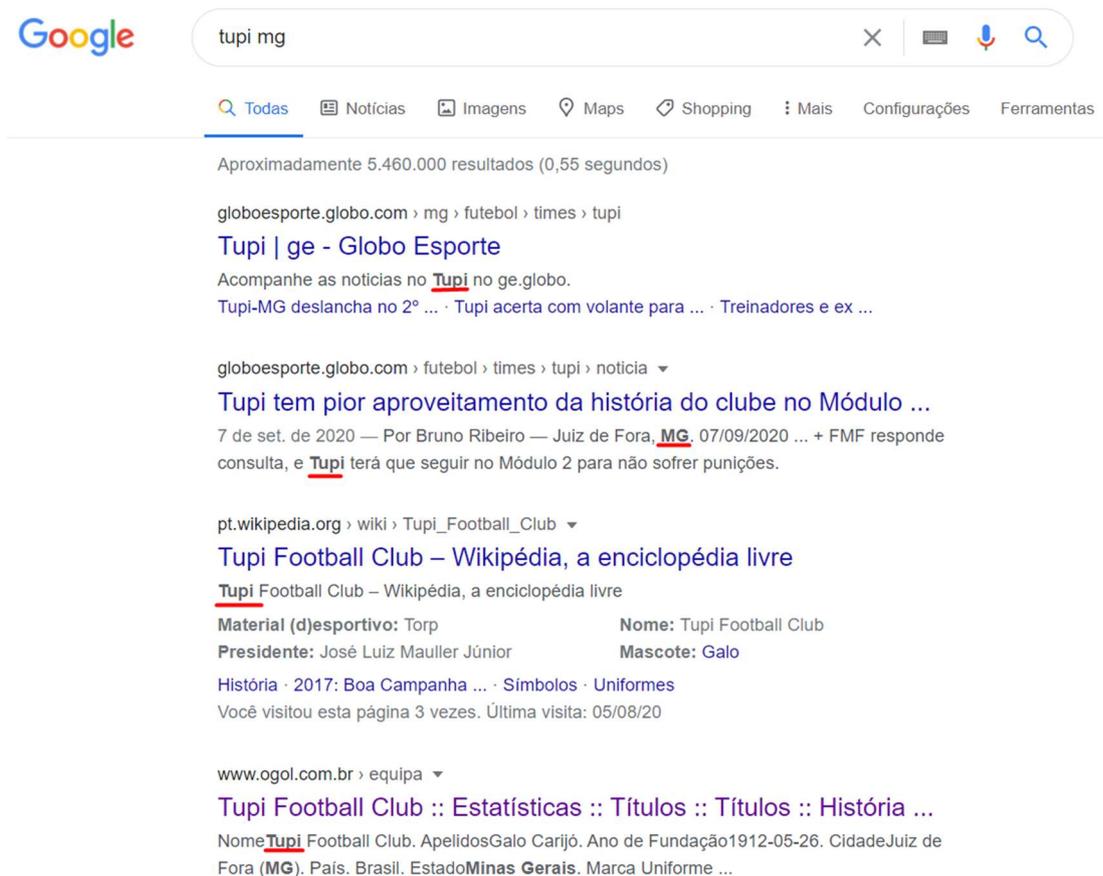


Imagem 3 - Fonte:

<https://www.google.com/search?q=tupi+mg&aq=tupi+mg&aqs=chrome.0.69i59j46j0l3j0i22i30l3.1098j0j4&sourceid=chrome&ie=UTF-8>

A partir dos dados coletados pelos pequenos robôs é possível tratá-los e classificar essas informações geradas. A coleta dos dados dos crawlers foi feita em diversas páginas, com essa técnica, conseguimos buscar informações tais como jogos futuros, jogos passados, posição em um determinado campeonato, maior artilheiro da história entre outras.



Imagem 4 referente ao serviço de jogos passados e futuros.

Fonte: <https://memorial-tupi.netlify.app/>



Imagem 5 referente ao serviço de classificação por campeonatos.

Fonte: <https://memorial-tupi.netlify.app/>

## Entendimento da Linguagem Natural

Durante o desenvolvimento da aplicação, sentimos a necessidade de uma maior interação do usuário, como por exemplo a possibilidade de o torcedor realizar uma pergunta.

Para isso, precisamos utilizar o Processamento de Linguagem Natural com intuito de implementar essa interatividade maior com o usuário durante a utilização do website.

O Processamento de Linguagem Natural é uma tecnologia que está em constante evolução, ela é responsável por criar uma comunicação entre computador e seres humanos, além da possibilidade de os computadores entenderem textos, ouvir e interpretar as falas e identificar sentimentos.

Suas principais utilidades são de plataformas de busca online que basicamente tem as funções de interpretar o que o usuário digitou e enviar para o usuário as melhores respostas, previsão de pesquisas em plataformas no caso mostrar sugestões sobre o que deseja pesquisar durante a digitação, assistentes virtuais que realizam tarefas solicitadas pelos usuários os exemplos mais comuns são Alexa, Siri e assistente do google.

A funcionalidade utilizada nesse projeto, foi criar um chatbots (Kuyven, 2018) para comunicação com o torcedor, imagem 6, com essa tecnologia podemos tornar a experiência com a aplicação mais agradável fazendo com que o usuário tenha maior facilidade e acesso a informações que possa ser útil ao mesmo.



Imagem 6 - do chatbot do Memorial Tupi

Fonte: <https://memorial-tupi.netlify.app/>

O DialogFlow da Google foi utilizado como mecanismo e processamento de linguagem natural, através deles podemos transformar a pergunta do usuário em uma ação a ser pesquisada pelos crawlers na internet.

## SPA

Atualmente, no mundo de desenvolvimento Front-end, novas tecnologias surgem para melhorar cada vez mais a agilidade e a qualidade das aplicações Web. Nesse contexto, a tecnologia SPA (Single Page Applications) foi criada para atender todas essas necessidades.

SPA funciona com o carregamento dos recursos (como CSS, JavaScript e HTML das páginas) apenas na primeira vez em que o usuário acessa a aplicação. Nesse primeiro acesso, todo o conteúdo do site é transferido para o cliente e a partir deste momento, quando o usuário transitar pelas páginas da aplicação, não será necessário mais fazer requisições para o servidor para a carga dessas novas páginas, pois o conteúdo relacionado a elas já foi baixado no primeiro acesso. O que acontece nesse momento é que o conteúdo da página é carregado via JavaScript, código este que é justamente gerado com base nos frameworks como Angular, React e Vue.js.

As aplicações SPA de maneira geral, nos permitem obter a vantagem de otimização em geral da performance da aplicação ao deslocar todo o esforço de renderização para o cliente e permitir um tráfego de dados mais leve entre cliente e servidor.

## 3 METODOLOGIA

### Arquitetura proposta

A arquitetura proposta para esse projeto, está descrita na imagem 7, nela podemos identificar duas estruturas bem divididas, o frontend e o backend. Por aí começamos o projeto, a equipe foi dividida em 2 grupos, e cada um responsável pelas funcionalidades. Essas duas estruturas serão detalhadas mais a frente bem como a comunicação entre eles.

vale ressaltar que toda a tecnologia e servidores utilizados são gratuitos.

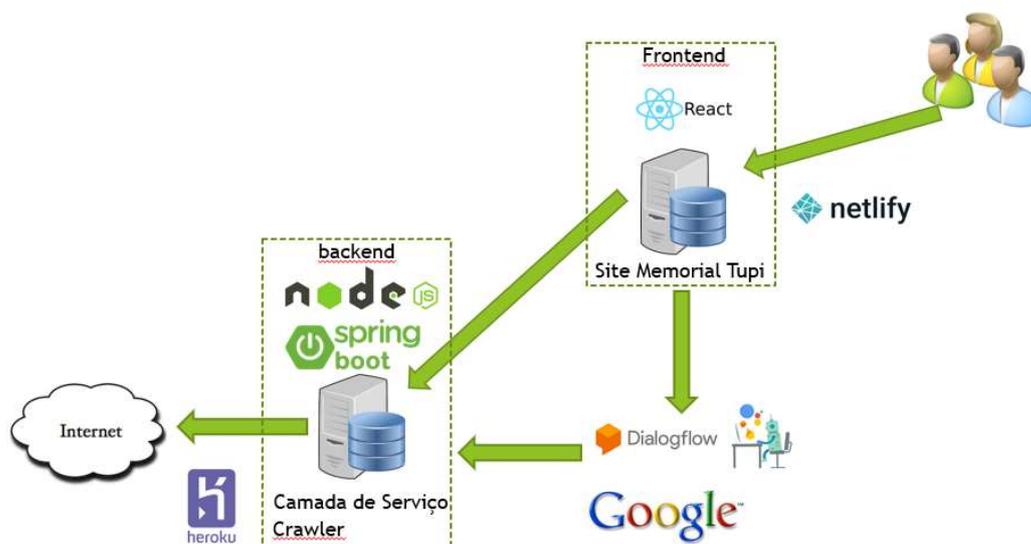


Imagem 7 – Arquitetura do sistema. Fonte: Do autor

## Backend

No Backend foi utilizado a tecnologia Java, Spring Boot, e NodeJS que são duas linguagens mais utilizadas nos dias atuais.

No backend estão localizados os serviços REST, toda solicitação, do usuário vinda do frontend, encontrará um serviço que responderá a solicitação do usuário, por exemplo, se o torcedor solicitar a classificação do Tupi no Campeonato Mineiro de 2005, terá um serviço que saberá responder a essa solicitação.

Algumas das respostas do serviço devem ser pesquisadas na internet, antes de ser respondidas, essa parte, é o papel do crawler, e foi desenvolvido em NodeJS.

Cabe ressaltar essas duas partes do sistema, estão hospedadas na heroku (Heroku, s.d.), A heroku é uma plataforma na nuvem que faz deploy de várias aplicações backend seja para hospedagem, testes em produção ou escalar as suas aplicações.

## Frontend

A tecnologia escolhida para desenvolvimento do Front-end foi React. Esta foi selecionada por ser a biblioteca mais popular de JavaScript do momento e por ser utilizado por grandes companhias ao redor do mundo, como: Netflix, Airbnb, American Express, Facebook, WhatsApp, eBay e Instagram.

O React foi lançado em 2013 como uma ferramenta JavaScript de código aberto para construir interfaces de usuário. Uma das grandes vantagens que o React traz é a possibilidade de aproveitar grande parte da sintaxe HTML e acoplar no código JavaScript. Com isso, os desenvolvedores podem aproveitar o seu conhecimento em HTML para desenvolver com esta biblioteca, porém utilizando a sintaxe JSX (JavaScript + sintaxe XML).

Outra vantagem é que a comunidade está crescendo muito em fóruns, novos recursos, implementações e correções para o React estão sendo feitas para facilitar cada vez mais o desenvolvimento dos sites. Além disso, o React permite que você crie uma interface de usuário que pode ser encontrada e acessa em diversos motores de busca. Esse recurso é uma vantagem sensacional porque nem todos os frameworks de JavaScript são amigáveis a SEO.

No frontend estão listadas as funcionalidades acessadas pelo cliente, nela é feita a interação com o dialogflow.

O frontend está hospedado na nelify, também fornece hospedagem gratuita de aplicações frontend.

## Dialogflow

O Dialogflow é uma ferramenta utilizada durante no desenvolvimento que facilitou a criação do chatbot no Memorial Tupi, a plataforma é usada para fazer projetos e integrar uma interface de usuário conversacional, com ela é possível controlar

dispositivos IoT, automatizar os atendimentos e a própria criação do chatbot. O motivo do seu uso é pela questão de ser um sistema que tem implementado o entendimento da linguagem natural sendo assim possível facilitar design e criar um chatbot sem ter que implementar um processamento de linguagem natural do zero, além de oferecer uma interface prática no desenvolvimento dos projetos e fornece novas maneiras para os usuários interagirem com a aplicação.

O consumo dos serviços que correspondiam ao dialogflow era utilizado por conta dos webhooks as perguntas personalizadas que eram atendidas pela plataforma são enviadas para o backend sendo assim possível criar métodos que façam consultas a banco de dados ou chamadas de API externas, sendo assim o serviço envia a mensagem para o Dialogflow e disponibilizando para o usuário sua resposta.

## **4 RESULTADOS E DISCUSSÃO**

O site foi desenvolvido para atender a demanda de resgatar os dados de anos anteriores do Tupi FC, como forma de facilitar o acesso a esses dados foi criado um chatbot interativo com a mascote do Tupi FC em diferentes expressões, artes feitas por Paola Fidélis Freitas, que pode, através de perguntas simples, tirar dúvidas dos usuários e informar sobre diferentes temas como, dados da fundação do clube, sobre a mascote, artilheiro do ano selecionado e sobre o estádio atual.

Outras funcionalidades foram implementadas em dois cards, o primeiro com as datas e placares de jogos anteriores, que podem ser filtrados de acordo com o ano selecionado, e sua colocação em cada campeonato atualmente. também mostrando a colocação final em anos anteriores, e o segundo card com informações do estádio, lotação e como chegar até lá.

Além disso existem botões que levam as redes sociais do time, como Facebook, Instagram e Youtube, além do contato via Whatsapp.

Para a hospedagem do site utilizamos a plataforma Netlify. Esse sistema oferece a funcionalidade de automatizar o sistema de deploy da aplicação no servidor, também disponibilizado por eles, ao sincronizar o repositório do projeto, seja no GitHub, Bitbucket ou GitLab. Assim, a cada atualização no repositório, é realizada uma nova rotina de build automaticamente do projeto e em poucos minutos a nova versão do site está disponível.

### **Problemas encontrados**

Uma das dificuldades encontradas pelo grupo foi com relação às requisições feitas aos “endpoints” da aplicação que tinha a funcionalidade de crawler, pois alguns sites que eram pesquisados com maior frequência, entendiam que essas requisições constantes e seguidas ao site deles poderia ser uma tentativa de ataque para derrubar o servidor. Assim o crawler era bloqueado. Uma maneira de tentar contornar essa situação, porém sem sucesso foi a implementação de um proxy, termo utilizado para definir intermediários entre o usuário e servidor. O proxy então esconde o endereço do computador fazendo com que a rota da requisição seja desviada, desta forma, evitando o bloqueio.

Por fim uma solução paliativa foi a coleta de dados em uma API e armazenamento em uma base dados MongoDB, evitando assim que as muitas das consultas fossem repetidas.

## **5 CONSIDERAÇÕES FINAIS**

Apesar das dificuldades encontradas, principalmente se o uso do website for feito em larga escala, o projeto atende aos torcedores do Tupi. Nesse o Website estão disponíveis dados estatísticos e históricos do clube carijó, os torcedores poderão interagir com o site realizando perguntas para um Chatbot que responderá as dúvidas dos torcedores.

## **REFERÊNCIAS**

Augusto Verzbickas, E. F. (2013). RELATÓRIO WEB CRAWLERS.

Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures.

*Heroku*. (s.d.). Fonte: [www.heroku.com](http://www.heroku.com).