



**Universidade do Minho**

UNIVERSIDADE DO MINHO

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA

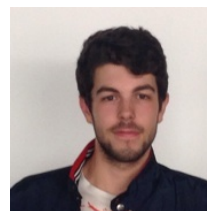
**Laboratório em Engenharia Informática**

**Tracking de jogadores de hóquei no ringue**

**Autores:**

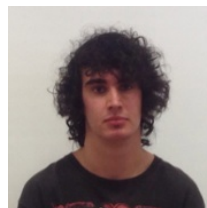
João Cabo

A75064



Diogo Soares

A74478



25 de Junho de 2019

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Contextualização</b>	<b>3</b>
<b>3</b>	<b>Recolha de Dados</b>	<b>3</b>
3.1	RTLS . . . . .	3
3.1.1	O que é? . . . . .	3
3.1.2	Como funciona? . . . . .	3
3.1.3	Implementação . . . . .	4
3.2	Gerador . . . . .	4
3.2.1	Estrutura do Ficheiro JSON . . . . .	4
<b>4</b>	<b>Tratamento de Dados</b>	<b>4</b>
<b>5</b>	<b>Apresentação dos Dados</b>	<b>5</b>
5.1	HeatMap . . . . .	5
5.2	Tabelas . . . . .	5
5.3	Representação do Jogo . . . . .	6
<b>6</b>	<b>Arquitectura da Aplicação</b>	<b>6</b>
<b>7</b>	<b>Manual de Utilização</b>	<b>7</b>
7.0.1	Dashboard . . . . .	8
7.0.2	Eventos . . . . .	8
7.0.3	Visualização de evento . . . . .	9
7.0.4	Registar Evento . . . . .	9
<b>8</b>	<b>Trabalho Futuro</b>	<b>9</b>
<b>9</b>	<b>Conclusão</b>	<b>10</b>

# 1 Introdução

No seguinte relatório apresentamos o projeto Tracking de Jogadores de Hóquei no Ringue, realizado no âmbito da unidade curricular Laboratório em Engenharia Informática. O projeto consiste na criação de uma aplicação que mostra dados relativos a jogos de hóquei em patins, dados esses que são recolhidos durante o jogo por sensores incorporados num sistema de *real time location* (RTLS). O foco deste projecto encontra-se assim, na movimentação dos jogadores e na construção de ferramentas que permitem analisar essa movimentação. A esta aplicação que desenvolvemos demos o nome de *Tactical Hockey*.

## 2 Contextualização

Cada vez mais se usa a tecnologia nas diferentes atividades do dia-a-dia. O desporto não é uma exceção, existindo cada vez mais casos da tecnologia a ser utilizada neste. Alguns exemplos das novas tecnologias usadas no desporto são: a linha de golo usada em vários desportos e, mais recentemente, a introdução do VAR no futebol.

Esta evolução não acontece só no jogo em si mas também no treino, onde cada vez mais se recolhe e analisa os dados do jogador melhorando a compreensão do seu desempenho e permitindo elevar as suas capacidades ao máximo. É neste campo em que nós entramos. É feita a recolha de dados dos jogadores intervenientes num jogo e são apresentados de forma agradável, permitindo ao treinador retirar mais conclusões sobre a *performance* de determinado jogador.

Como foi referido, este tipo de tecnologia já existe e já é aplicada em vários desportos *outdoor*, sendo o desafio aplicá-la também nos desportos *indoor* como é o caso do hóquei em patins.

Podemos dividir este projeto em três partes:

- **Recolha de dados**
- **Tratamento de dados**
- **Apresentação dos dados**

## 3 Recolha de Dados

Toda a aplicação gira em torno dos dados recolhidos durante um jogo de hóquei em patins. Nesta secção iremos descrever o processo pensado e idealizado para esta recolha de dados assim como a alternativa por nós encontrada.

### 3.1 RTLS

Numa fase inicial, pensou-se fazer a recolha de dados implementando um sistema de RTLS mas para isso seria necessário utilizar várias ferramentas que não tínhamos acesso, ainda assim vamos aqui apresentar como, numa primeira fase, pensamos tirar o melhor proveito deste sistema. Antes de passarmos à parte técnica convém esclarecer o que é realmente um RTLS.

#### 3.1.1 O que é?

Um RTLS é um conjunto de ferramentas que permite identificar e rastrear automaticamente um objecto ou uma pessoa em tempo real. Este tipo de tecnologia já é usada em vários desportos ao ar livre, no entanto, existe uma maior dificuldade na sua implementação nos desportos de pavilhão pois estes não permite a recolha de dados como os desportos ao ar livre. Após um trabalho de pesquisa efetuado, descobrimos várias ferramentas que permitem a implementação de um RTLS graças ao uso de *ultra wide-band* (UWB) e com uma precisão de centímetros.

#### 3.1.2 Como funciona?

Cada objecto ou pessoa que queremos rastrear tem uma *tag* associada a si. A *tag* envia sinais UWB em intervalos regulares para as ancoras. Estas comunicam entre si através de *ethernet* ou *wifi*, cabendo depois ao servidor calcular a posição da *tag*.

### 3.1.3 Implementação

Para implementar este sistema no nosso projeto precisaríamos no mínimo de quatro ancoras de modo a cobrir um campo de hóquei com as maiores dimensões possíveis e ainda uma *tag* por jogador.

## 3.2 Gerador

Como alternativa ao RTLS desenvolvemos um gerador em *python* capaz de gerar posições para cada jogador durante um jogo inteiro de hóquei em patins. O nosso gerador está programado para gerar uma posição por segundo num jogo de 20 minutos por parte, o que faz com que cada jogador tenha 2400 posições. Além disso, usa as maiores dimensões possíveis de um campo de hóquei (22 metros por 44 metros) mas ignora possíveis substituições de jogadores. À medida que vai gerando posições também vai calculando a distância percorrida por cada jogador assim como a sua velocidade. Na implementação original, estes cálculos seriam feitos através de um *parser* que iria percorrer os dados gerados pelo RTLS.

O gerador retorna um ficheiro *json* que depois é inserido na nossa base de dados implementada em *mongoDB*.

Esta alternativa permite nos fazer a análise de dados e fazer uma apresentação do jogo em diferido, não sendo possível a transmissão do jogo em tempo real.

### 3.2.1 Estrutura do Ficheiro JSON

O ficheiro JSON que falamos na secção anterior possui a seguinte estrutura:

- teamA
  - name
  - players
    - \* name
    - \* positions
      - posX
      - posY
      - time
    - \* data
      - dist
      - vel\_max
      - vel\_media
- teamB (igual à teamA)

## 4 Tratamento de Dados

A secção que nos obrigou a tratar os dados foi a da representação do jogo onde tivemos de redimensionar as posições para o tamanho do campo usado. Além disso, para implementar a área de cobertura da equipa tivemos de fazer cálculos adicionais como calcular o centro dessa área e os ângulos deste com as posições dos jogadores.

## 5 Apresentação dos Dados

Os dados recolhidos são apresentados de três maneiras distintas:

- *Heatmap*
- Tabelas
- Representação do jogo

### 5.1 HeatMap

Esta forma de representação dos dados tanto pode ser relativa a uma equipa inteira ou apenas a um jogador. Podemos ver uma representação do campo de hóquei em patins dividido em quadrados que representam uma área de 2 metros quadrados. Cada quadrado tem uma cor associada mostrando a quantidade de tempo no jogo passado naquela zona do campo. Do menor tempo para o maior temos verde, amarelo e vermelho, sendo que em cada cor temos também uma intensidade associada. Para a criação dos *heatmaps* foi utilizada a biblioteca *apex-chart*.

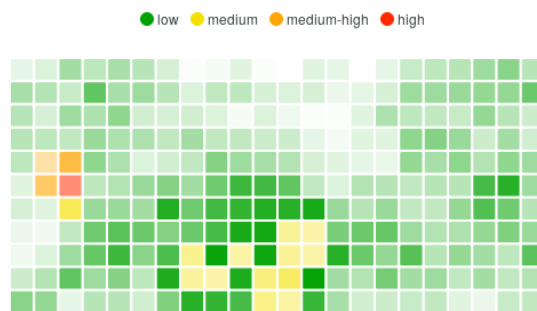


Figura 1: Heatmap de uma Equipa

### 5.2 Tabelas

Usamos tabelas para mostrar os dados estatísticos de cada jogador, existindo tabelas individuais onde só consta os dados de determinado jogador e tabelas da equipa onde podemos ver os dados de todos jogadores e o total da equipa. Nesta tabela podemos observar a distância percorrida, a velocidade média e a velocidade máxima de um jogador.

UD Oliveirense	Maximum Velocity (m/s)	Average Velocity (m/s)	Distance (m)
Tulipa Freitas	0.28	0.15	362.57
Arquimínio Coelho	2.79	1.51	3617.2
Lisandra Campos	2.79	1.53	3671.92
Adonilo Jesus	2.77	1.51	3628.26
Cristene Sousa	2.77	1.53	3678.51
<b>TOTAL</b>	<b>2.79</b>	<b>1.25</b>	<b>14958.46</b>

Figura 2: Tabela de Estatística dos Jogadores

### 5.3 Representação do Jogo

Será por aqui que o utilizador poderá ver o jogo depois de concluído, ou pelo menos, acompanhar o movimentos dos jogadores ao longo do encontro. É uma representação do jogo em 2D onde cada círculo representa um jogador, os de cor verde serão da equipa da casa e os da cor amarela da equipa visitante. Além disso podemos visualizar a área coberta pelos jogadores de campo de cada equipa. Este componente foi realizado através da biblioteca *p5.js*.

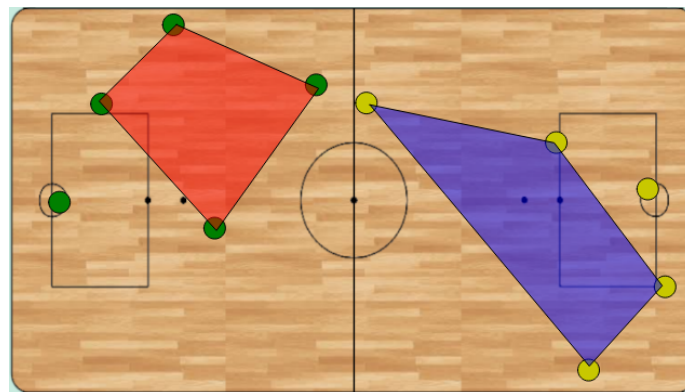


Figura 3: Representação do Jogo

## 6 Arquitectura da Aplicação

Iremos agora apresentar a arquitectura da nossa aplicação.

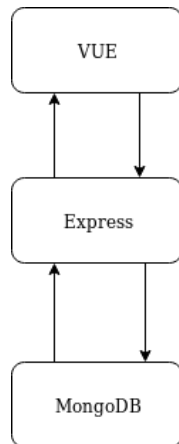


Figura 4: Arquitectura da Aplicação

Para a base de dados decidimos usar *mongoDB* por ser uma base de dados escalável que nos permite guardar diretamente os documentos JSON que contêm os dados dos jogos, permitindo aceder facilmente aos dados que desejamos.

No *backend* decidimos usar uma *framework* do *Node.js*, o *Express* que possui várias características bastante úteis quando trabalhamos com aplicações web. Por fim, no *frontend*, optamos pela utilização do *vue.js*, uma framework de Javascript que permite construir interfaces de utilizador de uma forma fácil e rápida.

## 7 Manual de Utilização

Na seguinte secção, encontra-se um guia de utilização para a nossa aplicação.



### 7.0.1 Dashboard

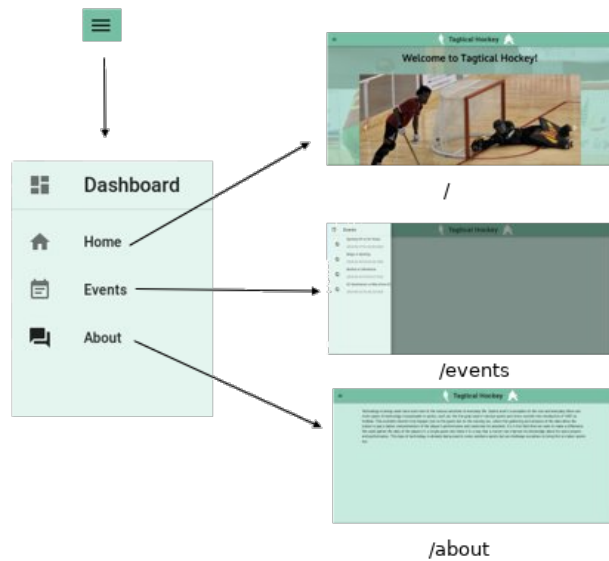


Figura 5: Funcionamento da Dashboard

### 7.0.2 Eventos



Figura 6: Seleccionar Eventos

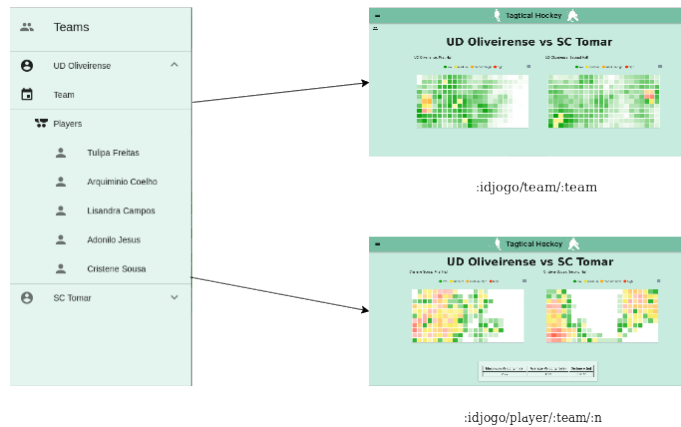


Figura 7: Visualizar dados da equipa ou de um jogador

### 7.0.3 Visualização de evento

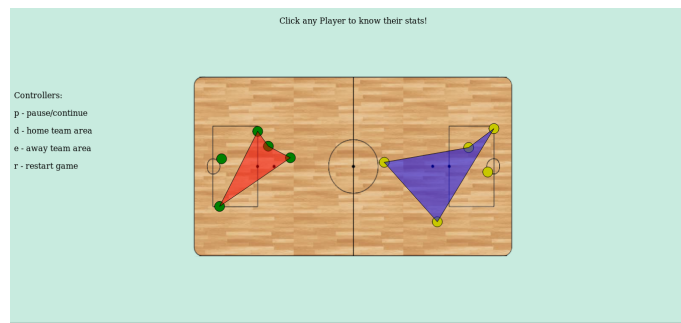


Figura 8: Visualização do evento

### 7.0.4 Registar Evento

Só os *admins* podem registar eventos, assim sendo, o utilizador que queira usar a nossa aplicação para ver as informações dos seus jogos particulares deve contactar os administradores, ficando estes encarregues da distribuição das *tags* e do registo das informações das equipas/jogadores.

## 8 Trabalho Futuro

Futuramente gostaríamos de introduzir o RTLS como proposto inicialmente. Achamos que seria interessante tanto a parte dos testes com as *tags* como o tratamento dos dados. Acreditamos que seriam preciso poucas mudanças na implementação realizada na adaptação à nova recolha dos dados, no entanto a parte mais interessante e provavelmente mais desafiante seria conseguir transmitir as movimentações dos jogadores em tempo real. Também gostaríamos de controlar a que minuto queremos ver o jogo, alterar a velocidade deste e voltar atrás a um momento específico do jogo.

## 9 Conclusão

Concluindo, estamos bastante satisfeitos com o resultado final visto termos cumprido os objetivos possíveis a que nos propusemos inicialmente. Além disso, este projeto tornou-se bastante importante porque nos permitiu familiarizar com ferramentas novas, principalmente relacionadas com o *frontend*, que são bastante utilizadas nos dias de hoje.