

Universidade do Minho  
Mestrado Integrado em Engenharia Informática  
Departamento de Informática

LABORATÓRIO EM ENGENHARIA INFORMÁTICA

---

## **Aplicação para a recolha de estatísticas em eventos desportivos**

---

Daniel Fernandes Veiga Maia A77531  
Maria de La Salete Dias Teixeira A75281  
Vitor Emanuel Carvalho Peixoto A79175

Ano Letivo de 2018/2019

## Resumo

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Planeamento</b>	<b>4</b>
<b>3</b>	<b>Recursos Necessários</b>	<b>5</b>
<b>4</b>	<b>Requisitos</b>	<b>6</b>
4.1	Objetivos do Projeto . . . . .	6
4.2	Requisitos Funcionais . . . . .	6
4.2.1	Funcionalidades genéricas . . . . .	6
4.2.2	Registo de eventos . . . . .	6
4.2.3	Utilizadores e permissões . . . . .	8
4.3	Requisitos de Usabilidade . . . . .	8
4.4	Requisitos de Desempenho . . . . .	9
4.5	Requisitos de Disponibilidade . . . . .	9
<b>5</b>	<b>Modelação</b>	<b>11</b>
5.1	Modelo de Domínio . . . . .	11
5.2	Diagrama de Casos de Uso . . . . .	13
5.2.1	Matriz de Funcionalidades . . . . .	14
5.3	Diagrama de Classes . . . . .	16
<b>6</b>	<b>Prototipagem</b>	<b>17</b>
<b>7</b>	<b>Implementação</b>	<b>23</b>
<b>8</b>	<b>Conclusão</b>	<b>24</b>

# 1 Introdução

Com a crescente acessibilidade de ferramentas de análise informáticas, tem vindo a ocorrer a difusão de aplicações analistas para uma variedade de modalidades desportivas. Ainda assim, a modalidade de hóquei em patins encontra-se carente de tais métodos desenhados para simplificar e agilizar o processo de análise de estatísticas desportivas.

É daqui que emerge o projeto de uma aplicação para a recolha de estatísticas em eventos desportivos, proposto pela Unidade Curricular de Laboratório de Engenharia Informática. Esta trata-se de uma aplicação que visa a permitir um utilizador introduzir dados relativos a jogos, em particular da modalidade de hóquei, e receber um conjunto de estatísticas relevantes relativos aos mesmos.

Ao longo deste relatório será explicitado o planeamento previsto da modelação e implementação da aplicação e os recursos necessários para cada estágio do projeto. Para além destes, serão levantados requisitos do sistema a implementar e destes será efetuada a modelação do mesmo. De seguida, serão elaborados um conjunto de *mockups* indicativos da interface visionada para a aplicação. Por fim, será efetuado um balanço do trabalho efetuado, dos resultados obtidos e do trabalho futuro na implementação do sistema.

## 2 Planeamento

O planeamento de um projeto é uma técnica muito importante e cada vez mais útil, sendo essencial em projetos de grandes dimensões. Com um bom planeamento a comunicação e organização entre a equipa é facilitada e o cliente tem uma melhor perceção dos passos necessários para chegar ao resultado final. Para além disso, ao criar o planeamento, fica-se com uma ideia mais clara de que etapas se devem tomar primeiro, quais as funcionalidades que dependem umas das outras e torna-se também possível avaliar o desempenho da equipa ao longo do desenvolvimento, sendo fácil identificar atrasos preocupantes.

Assim, com o intuito de otimizar o desenvolvimento do projeto, tentando maximizar o aproveitamento do tempo disponível, recorreu-se a um diagrama de *Gantt* com as etapas do projeto, as datas de início e a duração prevista para cada tarefa.

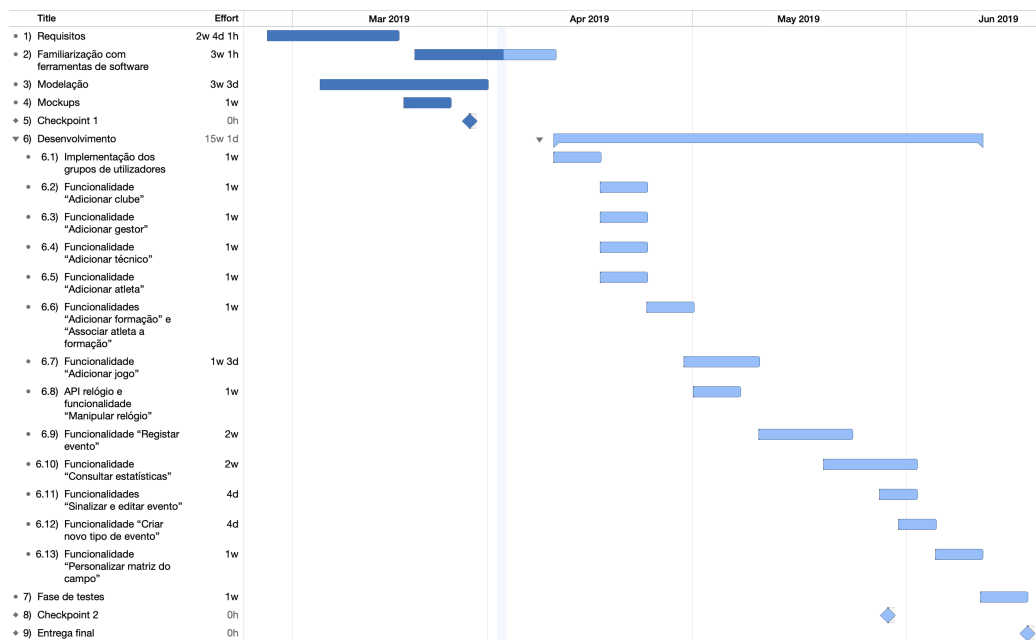


Figura 1: Diagrama de Gantt do projeto.

### 3 Recursos Necessários

Para o desenvolvimento desta aplicação foi necessário recorrer a várias ferramentas, tanto para o seu planeamento como para a implementação. Assim, para o planeamento utilizou-se o *Trello*, uma aplicação para organização de projetos, o *Lucidchart*, que permite desenvolver a modelação, e o *Inkscape*, uma ferramenta de criação de *mockups*.

Relativamente à implementação do projeto, esta teve que ser uma decisão mais ponderada, tendo-se tentado optar por ferramentas práticas, com uma boa curva de aprendizagem e ao mesmo tempo eficientes para os resultados pretendidas. Relativamente à base de dados, escolheu-se o *PostgreSQL* devido à sua notoriedade na comunidade e facilidade de utilização. Passando para a linguagem de programação, optou-se por codificar a aplicação em *Python*, pois esta é uma linguagem que engloba todas as características enunciadas, sendo bastante prática e intuitiva, ideal para um projeto desta dimensão e com uma boa curva de aprendizagem e uma biblioteca de funções diversa. Para o *back end*, optou-se pela *framework Django*, onde é possível aplicar a lógica de negócio e que lida automaticamente com a camada de dados, inclusive a criação das tabelas necessárias na base de dados. Por último, para o desenvolvimento do *front end* tem-se como características fundamentais uma interface rápida e responsiva, que seja capaz de se adaptar aos diferentes tamanhos de ecrã (computador, tablet e smartphone). Para tal, optou-se por utilizar a *framework* para *JavaScript* *Vue.js*, complementando esta com o *Bootstrap*, sendo este último o responsável pelo ajuste do *layout*.

É também importante referir que, numa fase mais avançada do projeto, será necessário eleger ferramentas adequadas para disponibilizar a aplicação ao público, nomeadamente um servidor online centralizado e o tipo de servidores locais para o registo de eventos no decorrer de um jogo.

## **4 Requisitos**

### **4.1 Objetivos do Projeto**

O projeto tem como objetivo desenvolver uma aplicação que facilite a análise e deteção de problemas que permitam auxiliar no treino e desenvolvimento de uma equipa de hóquei em patins.

O uso de uma aplicação num dispositivo móvel que permita o registo de eventos ocorridos numa partida, faculta a análise estatística de todos os aspetos táticos de uma equipa. Deste modo a equipa técnica pode consultar os pontos fortes e fracos da sua equipa e trabalhar neles, de modo a melhorar os resultados desportivos da sua equipa e a acelerar a evolução individual dos seus jogadores.

Este tipo de aplicações são bastante comuns no universo desportivo em Portugal, porém a modalidade de hóquei em patins encontra-se tecnologicamente atrasada. Assim, esta aplicação é vista como uma novidade que permitirá desenvolver a qualidade de jogo praticado em Portugal.

### **4.2 Requisitos Funcionais**

#### **4.2.1 Funcionalidades genéricas**

1. A aplicação deve permitir o registo de ocorrências/eventos durante uma partida de hóquei em patins.
2. A aplicação deve gerar e permitir consultar as estatísticas durante e após uma partida, sendo que as estatísticas geradas devem ser apresentadas graficamente num campo do jogo.
3. A aplicação deve permitir a visualização das estatísticas relativas a um específico jogador, tipo de evento ou momento do jogo.
4. A aplicação deve apresentar um relógio semelhante ao utilizado na modalidade, permitindo que este seja alterado e pausado quando necessário.

#### **4.2.2 Registo de eventos**

5. O registo de um evento deve incluir a seguinte informação: tipo de evento; atleta e equipa; instante de tempo atual; instante de tempo

atualizado; zona de campo; zona da baliza e *timestamp* do horário de registo do evento.

6. Dependendo do tipo de evento registado, alguns dos detalhes do evento são dispensáveis, como atleta, instante de tempo atualizado, zona do campo ou zona da baliza.
7. Alguns tipos de evento existentes devem estar predefinidos na aplicação, podendo o utilizador registar novos tipos de eventos.
8. Os tipos de evento que a aplicação permite registar devem ser pelo menos os seguintes: ataque organizado; ataque rápido; contra-ataque; erro defensivo; remate à baliza; remate fora; remate interceptado; perda de bola; roubo de bola; recuperação de bola; golo; falta; falta de equipa; livre direto; penálti; bloqueio; golpe duplo; 5 segundos; 10 segundos; 45 segundos; cartão azul; cartão vermelho; substituição; *timeout*; *powerplay*; fim do *powerplay* e atualização do relógio.
9. O registo dos eventos deve ser efetuado através da introdução de códigos simples num teclado numérico.
10. Aquando o registo de um evento, o instante de tempo deverá ser obtido automaticamente, a partir de uma API.
11. No registo de eventos, para selecionar a zona do campo onde este ocorreu, deve ser utilizada uma matriz para dividir o terreno de jogo, sendo cada zona representada por um número. O utilizador deverá selecionar a zona do campo utilizando novamente o teclado numérico.
12. A matriz utilizada para dividir o campo deve ser, por omissão, 8x4 e deve poder ser personalizada pelo utilizador.
13. A matriz utilizada para dividir a baliza deve ser, por omissão, 3x3 e deve poder ser personalizada pelo utilizador.
14. A seleção do atleta responsável por um evento na partida deve ser feita pelo teclado numérico também, digitando o número da sua camisola.
15. A sinalização e alteração de eventos incorretamente registados deve ser facultada através de uma interface simplificada quando acedida durante o decorrer do jogo.



16. A sinalização e alteração de eventos incorretamente registados deverá ser facultada através de uma interface que inclui todas as opções de personalização do evento quando acedida após o término do respetivo jogo.

#### **4.2.3 Utilizadores e permissões**

17. A aplicação pode ser utilizada por um administrador principal, gestores de clubes e utilizadores comuns (técnicos de equipa).
18. O registo e *login* de qualquer tipo de utilizador deve incluir o email e password.
19. O administrador principal deve poder registar clubes e respetivo gestor.
20. Os gestores de clubes podem registar outros gestores, atletas, técnicos, formações (equipas) do clube e respetivos jogadores e jogos a realizar. Podem também consultar as estatísticas das partidas de todas as formações do clube.
21. Os técnicos de equipa podem criar novos tipos de eventos, registar, sinalizar e editar os eventos de uma partida e consultar as estatísticas da sua formação relativamente a eventos anteriores.
22. Os atletas devem ser identificados pelo seu número de licença.
23. Os atletas devem ter associado um número de camisola.
24. Os técnicos podem definir a numeração da equipa na secção de definições ou alterar a numeração antes de cada partida, mas nunca durante.
25. A criação de um jogo deve incluir a equipa adversária, os atletas inscritos por cada equipa, o dia, hora e local do jogo (casa ou fora) e o seu carácter (oficial ou amigável).

### **4.3 Requisitos de Usabilidade**

1. Um técnico com conhecimento da modalidade de hóquei em patins deverá conseguir utilizar 70% das funcionalidades da aplicação após 2

horas de utilização e 100% das funcionalidades básicas, como o registo de eventos.

2. A aplicação deverá ser construída para que o registo de um evento ocorrido durante a partida não dure mais que 5 segundos.
3. O registo de eventos através de códigos deve ser preciso, de modo a razão de erro no registo de eventos seja inferior a 5%.
4. A interface deverá ser intuitiva, de modo a que um utilizador com periodicidade de utilização semanal, seja capaz de manusear a aplicação com a mesma facilidade na semana seguinte.
5. A aplicação deve suportar a correção de eventos incorretamente registados em menos de 10 segundos.
6. O acerto no tempo do relógio deverá demorar, no máximo, 3 segundos.

#### **4.4 Requisitos de Desempenho**

1. A aplicação deve conseguir apresentar estatísticas num tempo de resposta máximo de 2 segundos após a inserção de um evento.
2. O registo de uma nova ocorrência deverá ser processado de imediato, de modo a que seja possível inserir um novo evento imediatamente a seguir.
3. A submissão das estatísticas de um jogo, de uma base de dados local para a base de dados geral da aplicação deve ser efetuada com uma latência nunca superior a 10 segundos.

#### **4.5 Requisitos de Disponibilidade**

1. O produto deverá estar disponível para utilização um mínimo de 360 dias por ano.
2. O produto deverá estar disponível durante um período mínimo 240 minutos ininterruptos de cada vez.

3. Os dados e estatísticas relativos a um evento a decorrer devem estar disponíveis de imediato (armazenadas localmente), mesmo sem ligação ao servidor central.
4. Os dados e estatísticas de eventos passados deverão estar disponíveis para consulta e edição mediante uma conexão ao servidor central da aplicação.



administradores, gestores e técnicos trata-se dos utilizadores do sistema, e, como tal, têm associados um email único e uma palavra passe com os quais acedem ao sistema.

Para além disto, cada técnico é responsável por treinar uma formação de atletas do clube e gerir os eventos e relógio de quaisquer jogos em que a sua formação participe. Estes jogos são utilizados para gerar dados estatísticos para análise posterior.

Por fim, cada atleta tem um número de licença único e imutável, bem como um número de camisola variável entre jogos. Este número de camisola pode ser alterado pelo técnico da formação respetivo do atleta.

## 5.2 Diagrama de Casos de Uso

A análise dos requisitos apresentados levou à criação de um diagrama de casos de uso, onde se definiu as funcionalidades para as três entidades referidas nos requisitos: administrador, gestor de clube e técnico.

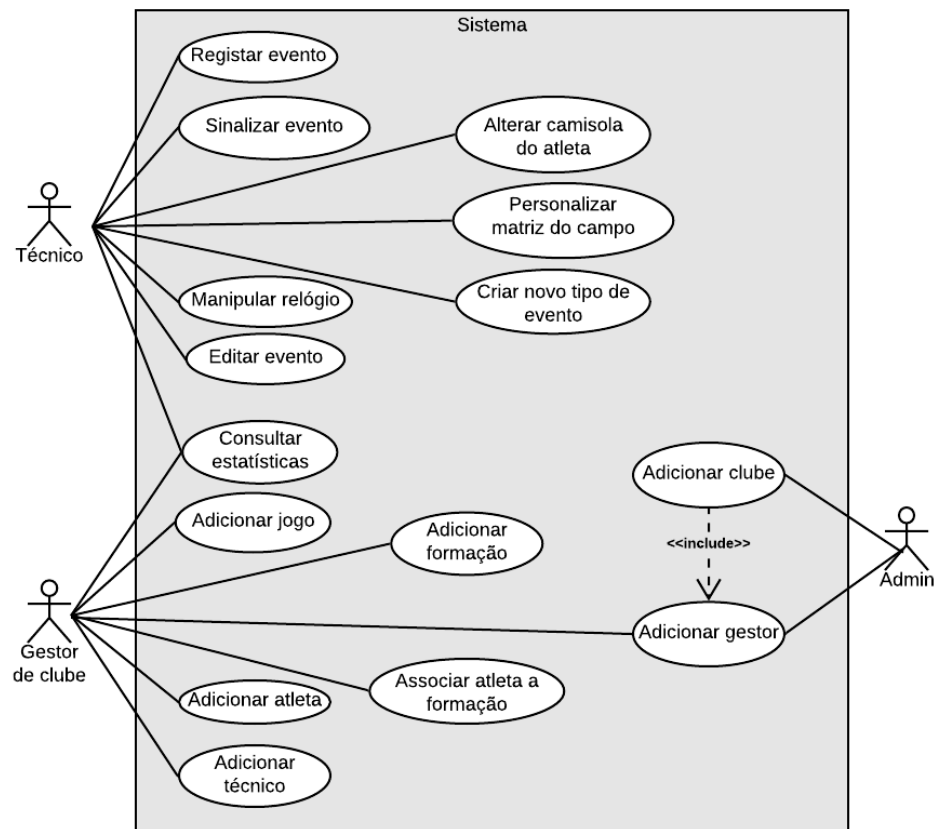


Figura 3: Diagrama de Casos de Uso

O Administrador pode também realizar as ações de adicionar clube e, consequentemente, adicionar um gestor associado ao mesmo clube.

O Gestor de Clube partilha com o Administrador a ação de adicionar gestor de clube, sendo que este apenas o pode fazer para o seu próprio clube. Este tem também associado a si os *use cases* de adicionar um novo jogo, adicionar um técnico, adicionar uma nova formação, adicionar um atleta,

associar um atleta a uma formação e consultar estatísticas referentes ao seu clube.

Por último, o Técnico partilha com o Gestor de Clube o caso de uso de consultar estatísticas, podendo também realizar todo o tipo de ações relacionadas com a criação de eventos para os jogos que desejar. Estas ações passam por criar um novo tipo de evento, personalizar a matriz do campo, alterar camisola de um atleta, manipular relógio, ou seja, pausar, incrementar ou decrementar o relógio em segundos, registar, sinalizar e editar um evento.

### 5.2.1 Matriz de Funcionalidades

A partir do Diagrama de Casos de Uso, desenvolveu-se uma matriz de funcionalidades, que permite uma melhor visualização das funcionalidades por tipo de utilizador. Assim, torna-se mais fácil identificar quais os utilizadores que partilham funcionalidades e a melhor ordem para o desenvolvimento das mesmas, conseguindo assim identificar que ações dependem de outras.

	Administrador	Gestor	Técnico
Adicionar clube			
Adicionar gestor			
Adicionar técnico			
Adicionar formação			
Adicionar atleta			
Associar atleta a formação			
Adicionar jogo			
Consultar estatísticas			
Alterar camisola de atleta			
Personalizar matriz do campo			
Manipular relógio			
Criar novo tipo de evento			
Registar evento			
Sinalizar evento			
Editar evento			

Tabela 1: Matriz de Funcionalidades

Com os requisitos especificados e as funcionalidades identificadas, é possível associar quais os requisitos que cada funcionalidade deve respeitar.

- **Adicionar clube:** deve estar de acordo com o requisito 18;
- **Adicionar gestor:** deve estar de acordo com os requisitos 18 e 19;
- **Adicionar técnico:** deve estar de acordo com o requisito 19;
- **Adicionar formação:** deve estar de acordo com o requisito 19;
- **Adicionar atleta:** deve estar de acordo com os requisitos 19 e 21;
- **Associar atleta a formação:** deve estar de acordo com os requisitos 19 e 22;
- **Adicionar jogo:** deve estar de acordo com os requisitos 19 e 24;
- **Consultar estatísticas:** deve estar de acordo com os requisitos 2, 3, 19 e 20;
- **Alterar camisola de atleta:** deve estar de acordo com os requisitos 22 e 23;
- **Personalizar matriz do campo:** deve estar de acordo com o requisito 12;
- **Manipular relógio:** deve estar de acordo com o requisito 4;
- **Criar novo tipo de evento:** deve estar de acordo com os requisitos 5, 6 e 8;
- **Registar evento:** deve estar de acordo com os requisitos 1, 5, 7, 8, 9, 10, 11, 13 e 20;
- **Sinalizar evento:** deve estar de acordo com os requisitos 14, 15 e 20;
- **Editar evento:** deve estar de acordo com os requisitos 14, 15 e 20;



### 5.3 Diagrama de Classes

O diagrama de classes é uma das partes mais importantes da modelação, sendo o que fornece mais informações e o que mais se aproxima da implementação do projeto. Neste definem-se as classes, atributos e tipos de dados que se deve ter presente na aplicação.

Para o desenvolvimento deste diagrama, teve-se em conta a *framework* utilizada para o *back end* desta aplicação, nomeadamente o *Django*, uma *framework* de *Python*. Depois de uma breve pesquisa, o grupo aprendeu como gerar automaticamente o diagrama de classes a partir dos *models* do projeto. Assim, começou-se por definir os *models* necessários e só depois se criou o diagrama, que se apresenta de seguida.

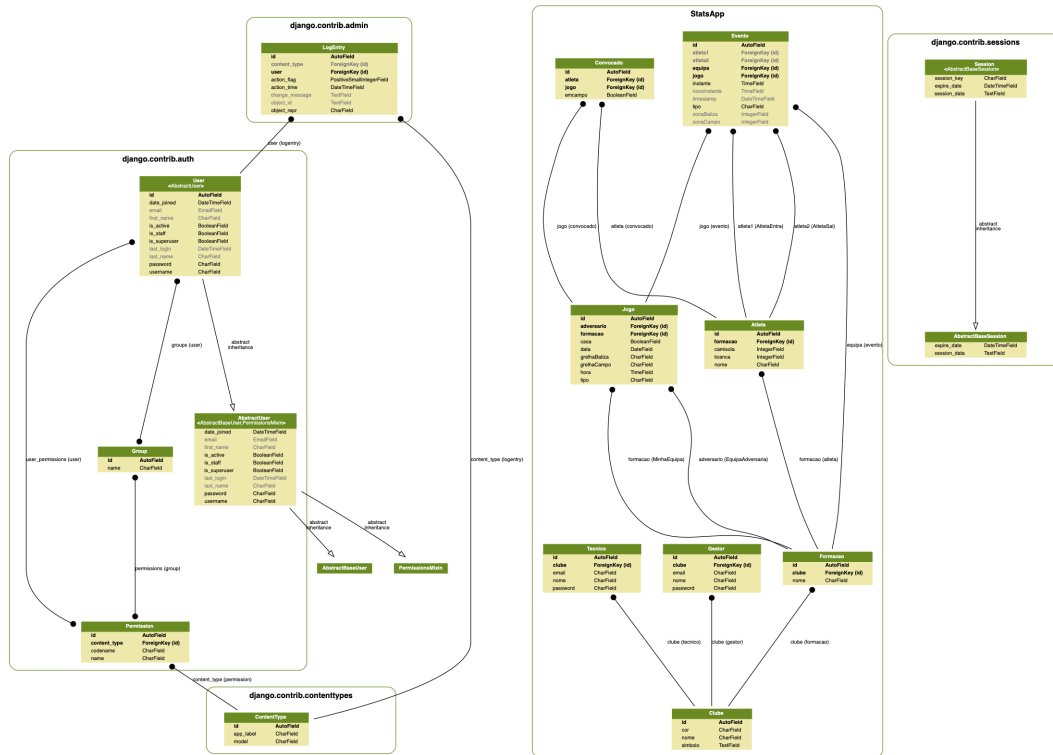


Figura 4: Diagrama de Classes

## 6 Prototipagem

Tendo em conta o objetivo principal desta aplicação, o registo de eventos, rapidamente se percebe que a interface é uma das partes mais importantes do projeto. Esta deve ser bastante intuitiva, simples e prática, permitindo que o registo de um evento seja uma ação rápida de concluir.

Assim, para se concluir a aplicação com uma interface que seja capaz de cumprir todos os pontos necessários, optou-se por desenvolver protótipos para a mesma, tendo estes sido validados pelo cliente. De seguida, apresentam-se os *mockups* desenvolvidos.



Figura 5: Página inicial com a listagem completa de jogos.



Figura 6: Página representativa do resumo de um jogo.

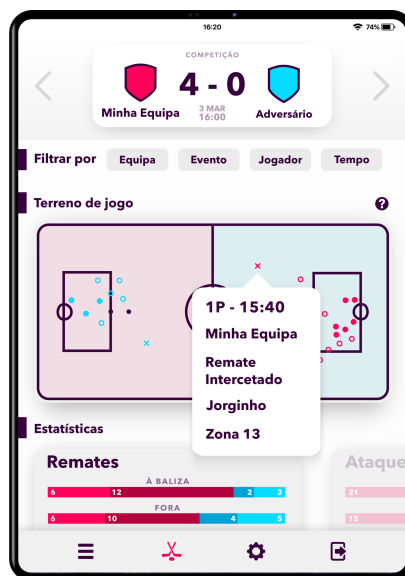


Figura 7: Exemplo da representação detalhada de um determinado evento de um jogo.



Figura 8: Exemplo de seleção de eventos que serão monitorizados no decorrer de um determinado jogo.



Figura 9: Página de um jogo a realizar e ligação para encontros anteriores.



Figura 10: Página de seleção dos convocados de uma formação para um determinado jogo.

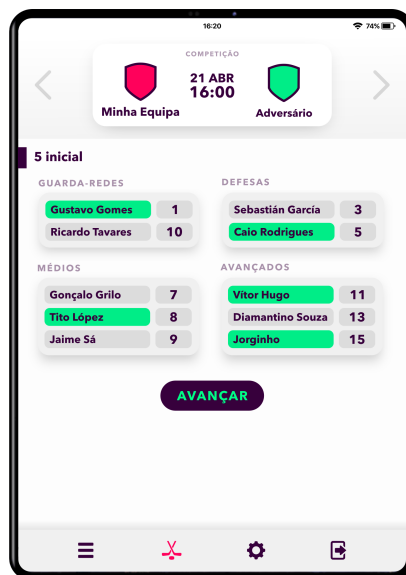


Figura 11: Página de seleção do 5 inicial de uma formação para um determinado jogo.

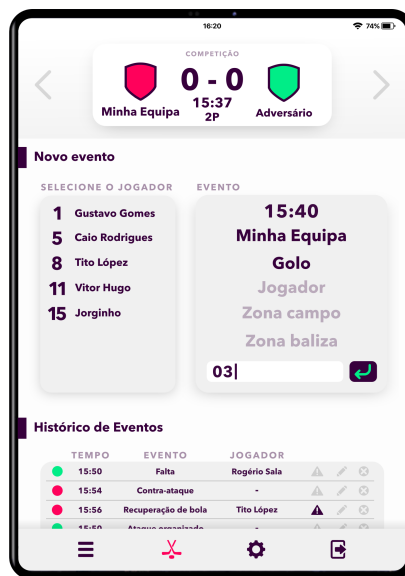


Figura 12: Exemplo demonstrativo do ecrã de marcação de eventos.

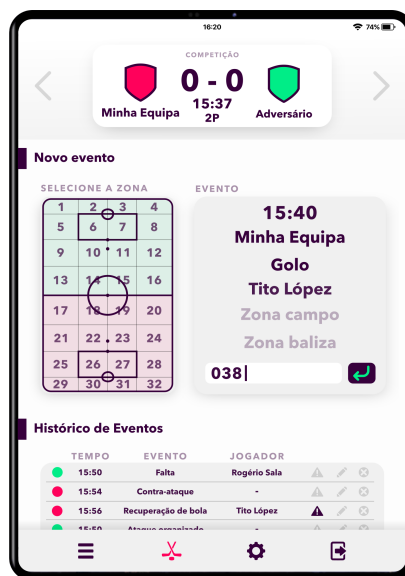


Figura 13: Exemplo demonstrativo do mapa do campo durante a marcação de eventos.

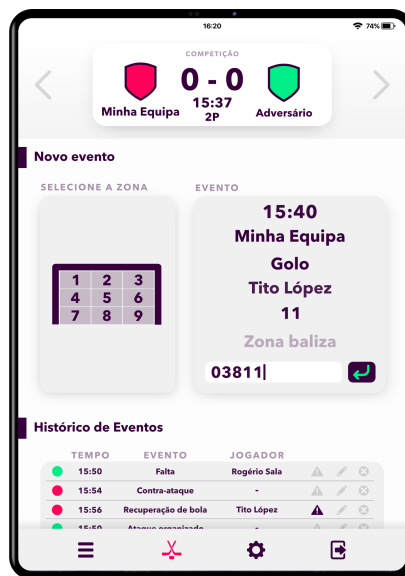


Figura 14: Exemplo demonstrativo da grelha da baliza durante a marcação de eventos.



Figura 15: Página de visualização e alteração do relógio do jogo.

## 7 Implementação



## 8 Conclusão