

PYTHON

**ANÁLISE DE**

**DADOS**

**&**

**FUTEBOL DE ROBÔS**

# MÓDULOS



FUTEBOL DE ROBÔS

Módulo 1



ANÁLISE DE DADOS

Módulo 2



MÃO NA MASSA

Módulo 3



FUTEBOL DE ROBÔS

MÓDULO 1

# ROBOCUP

## BREVE HISTÓRIA

A RoboCup foi criada para promover a investigação em robótica e IA, oferecendo um desafio interessante para o público geral e, ainda assim, cientificamente produtivo. Uma das formas eficazes de promover a investigação científica e de engenharia é estabelecer um objetivo a longo prazo. Construir um robô que jogue futebol por si só não gerará impacto social e econômico significativo, mas a realização certamente será considerada uma grande conquista para a área.

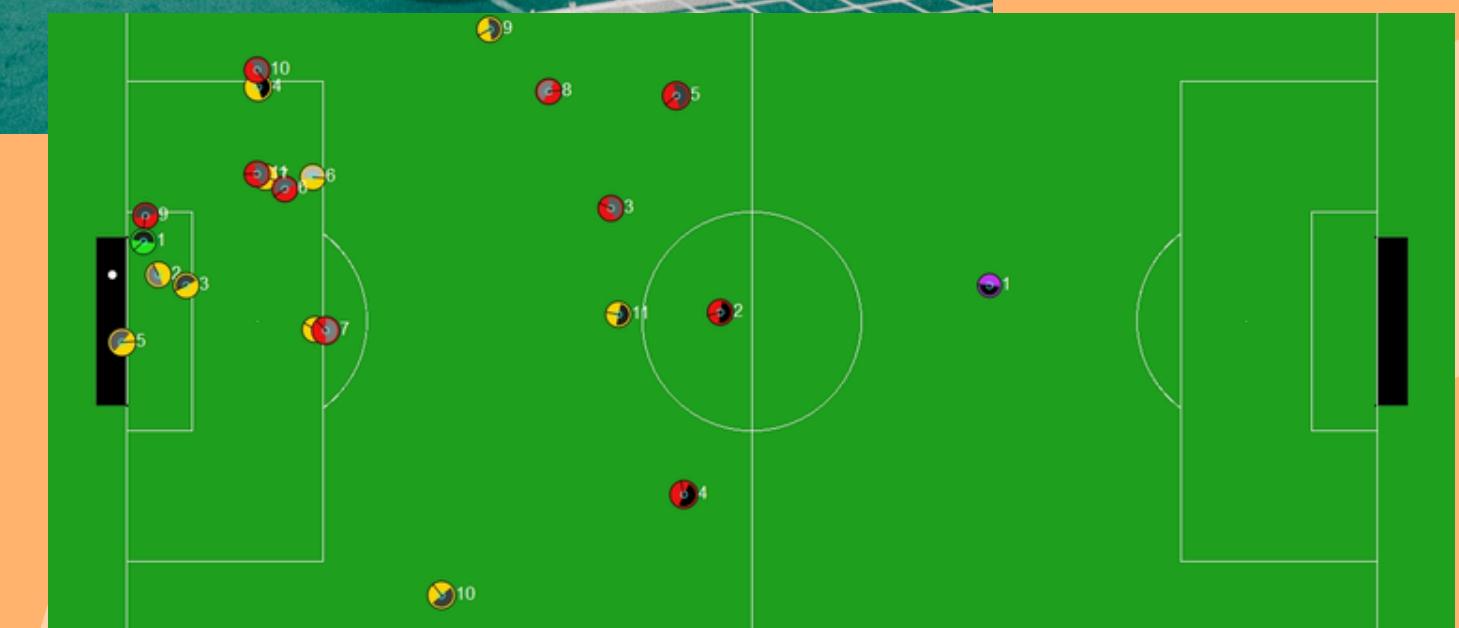
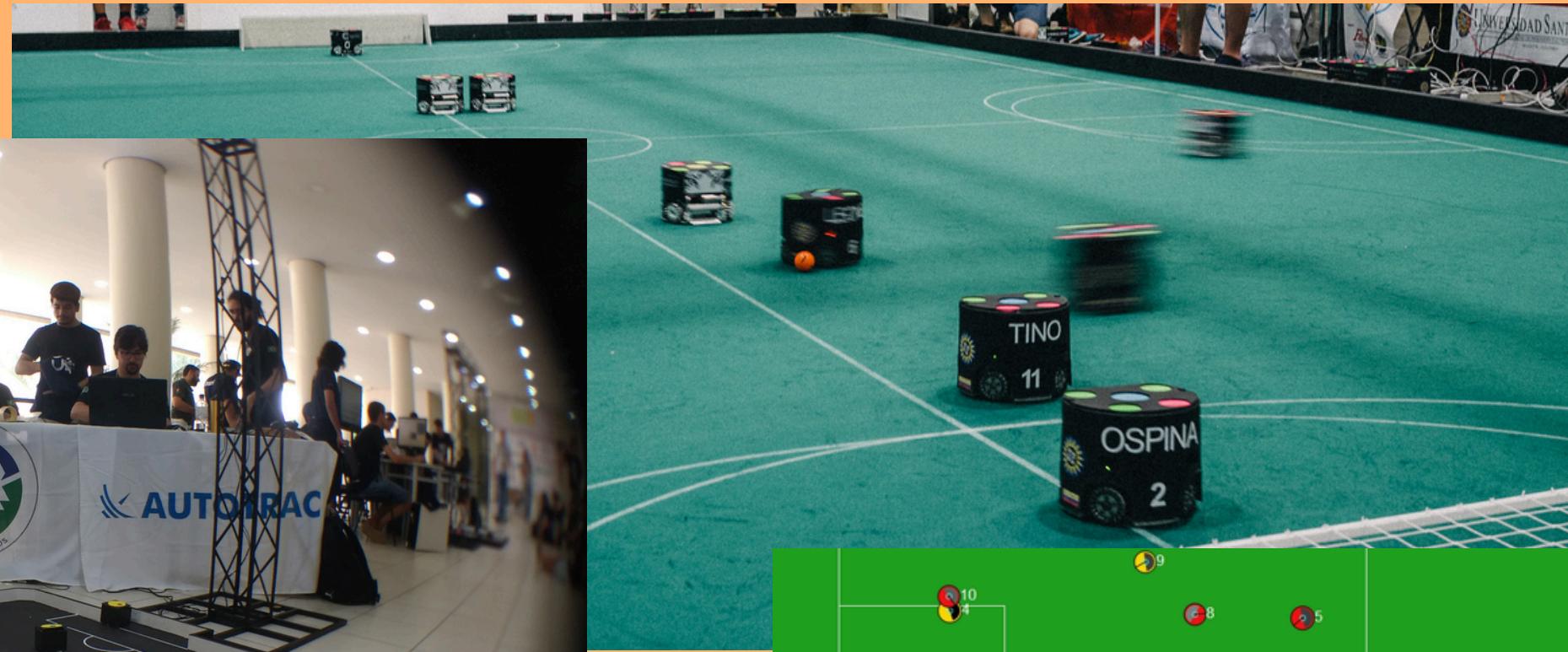
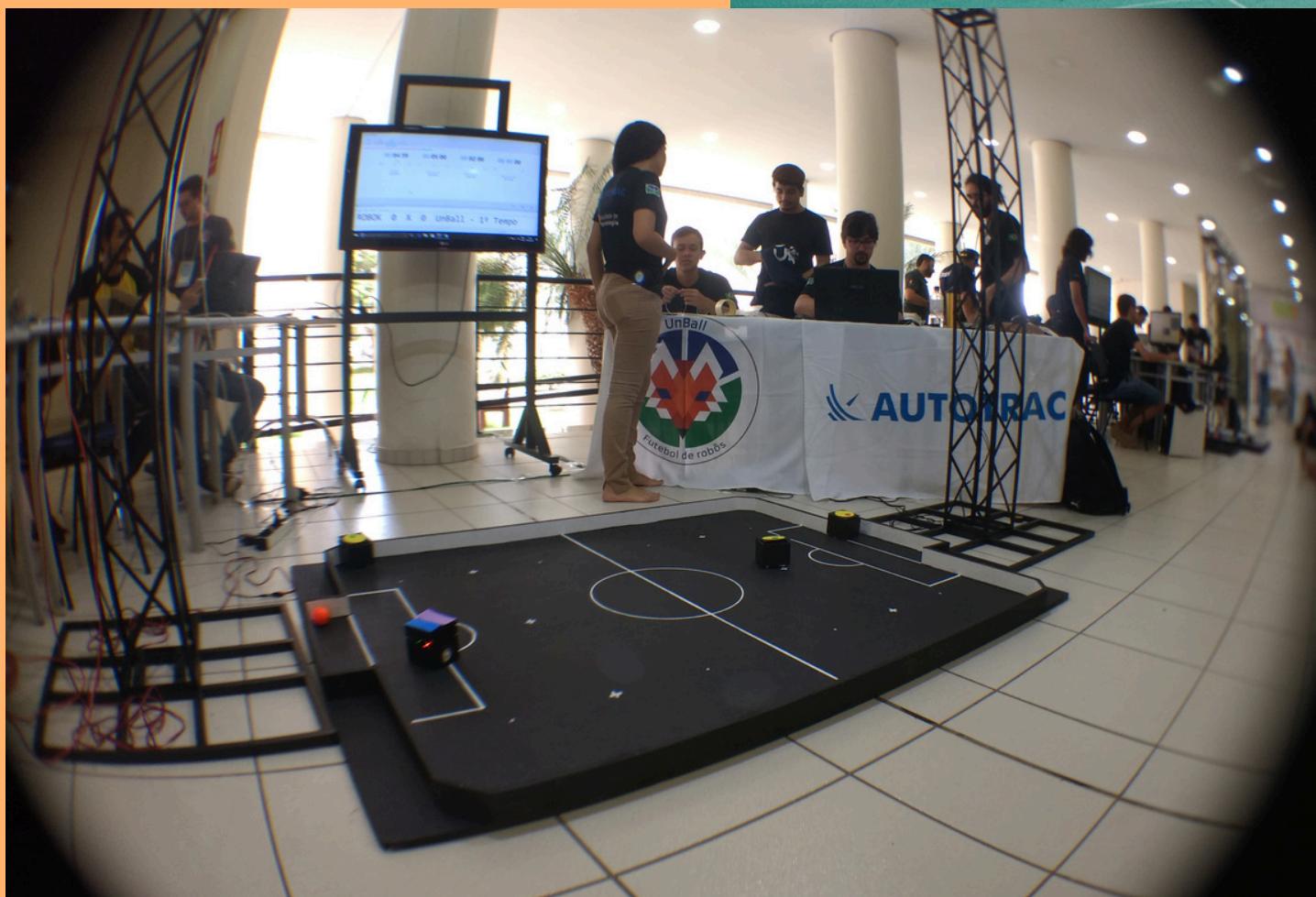
## ROBOCUP BRASIL

A RoboCup Brasil é uma associação civil sem fins lucrativos que tem como objetivo divulgar e aprimorar pesquisas sobre Inteligência Artificial e Robótica no país. Criada em 2007, é resultado de um esforço em conjunto de professores e pesquisadores que trabalham de forma voluntária para realizar, promover, apoiar e incentivar atividades que estimulem pesquisas científicas em todo o Brasil.

# UNBALL

## CATEGORIAS

- Atualmente competimos em VSSS (Very Small Size Soccer), SSL (Small Size League), e **Simulado 2D**





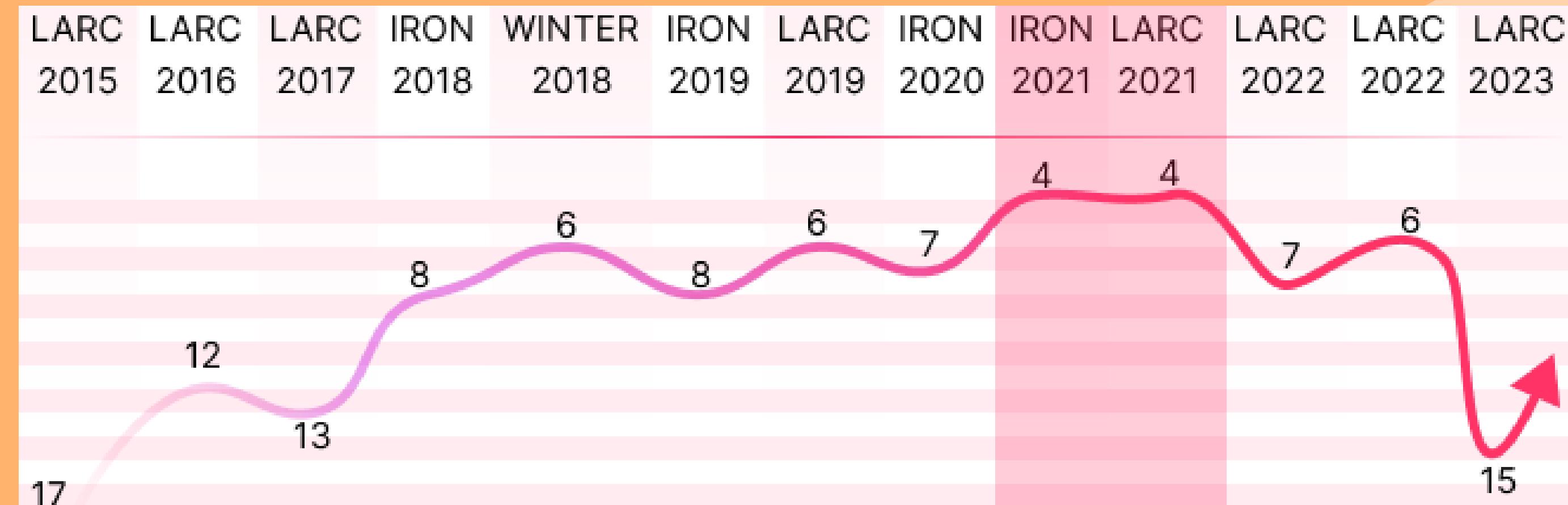
## MOTIVAÇÕES

- Colocar em prática o conhecimento teórico adquirido com as matérias do curso deparando-se com problemas reais
- Participar de competições nacionais e internacionais de robótica
- Oportunidade para aprender sobre assuntos novos
- Realizar pesquisas a respeito das áreas trabalhadas na equipe
- Fazer networking



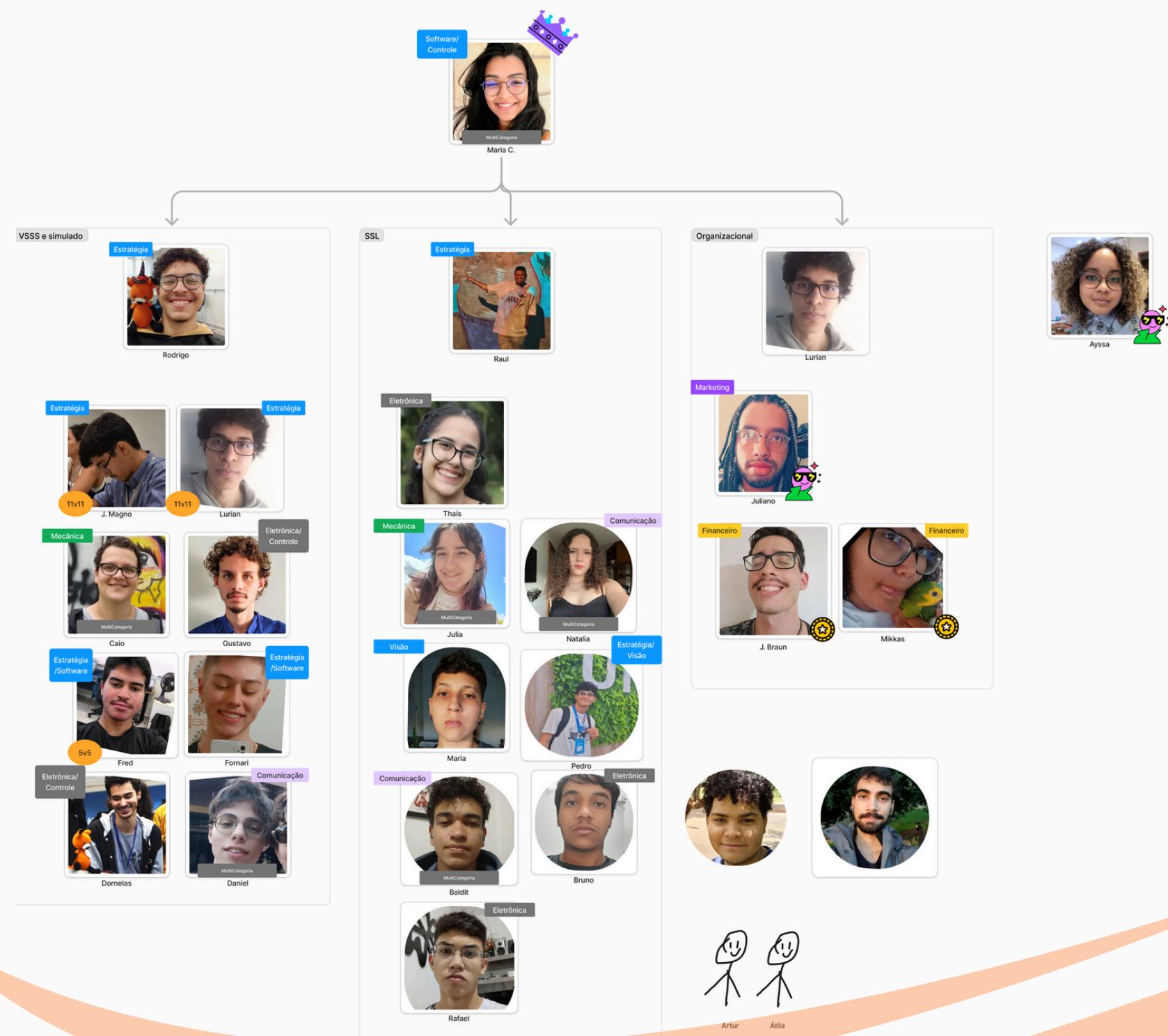
# UNBALL

## COMPETIÇÕES



# TREINAMENTOS UNBALL

# UNBALU



# SIMULADO

## ROBOCUP SOCCER - SIMULATION LEAGUE

É uma das ligas mais antigas da RoboCupSoccer. A Simulation League foca em inteligência artificial e estratégia de equipe. Jogadores (agentes) que se movem de forma independente jogam futebol em um campo virtual dentro de um computador. Existem 2 subligas: 2D e 3D.



# SIMULADO

## VISÃO GERAL

Duas equipes de onze jogadores (programas de software autônomos, chamados agentes) jogam futebol em um estádio de futebol virtual 2D representado por um servidor central, chamado SoccerServer.

## SOCER SERVER

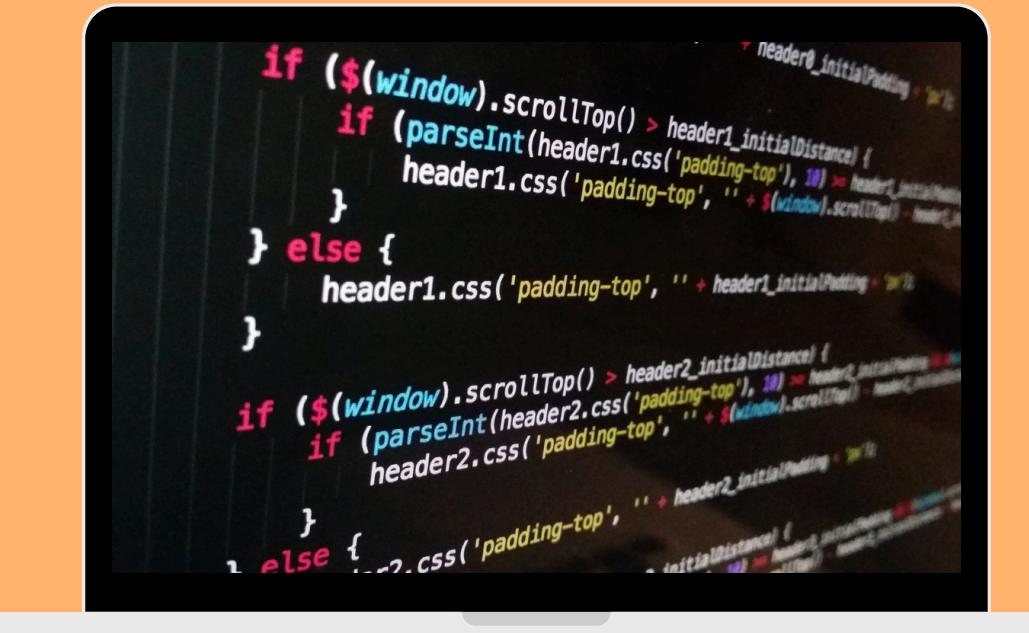
Este servidor sabe tudo sobre o jogo, ou seja, a posição atual de todos os jogadores e da bola, as condições “físicas” dos jogadores e do ambiente, e assim por diante. O jogo depende da comunicação entre o servidor e cada agente, e entre o servidor e o software de cada equipe.

# SIMULADO

rcssserver

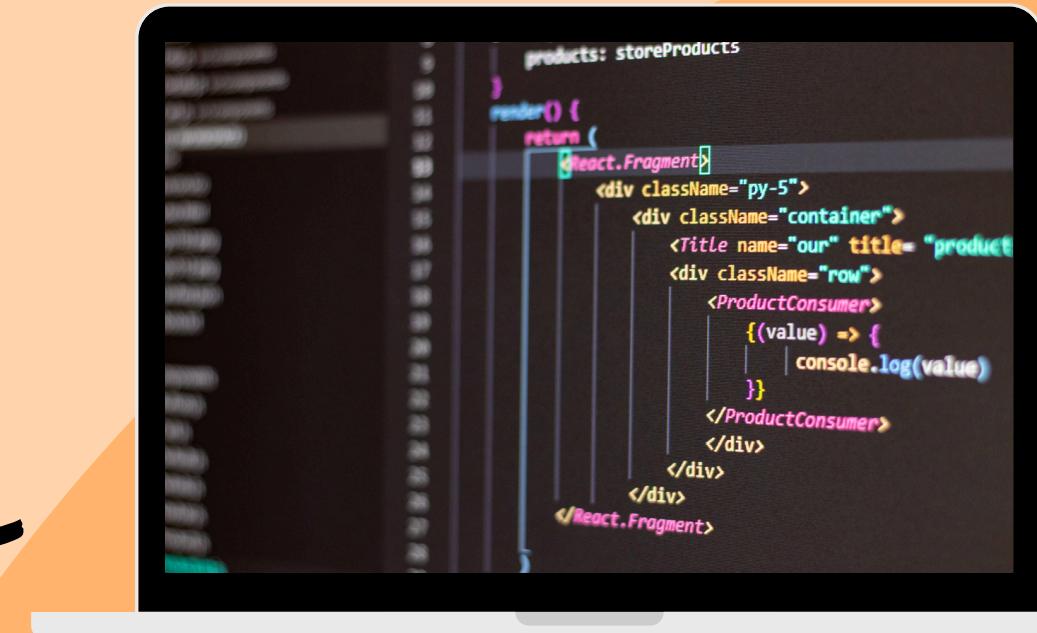
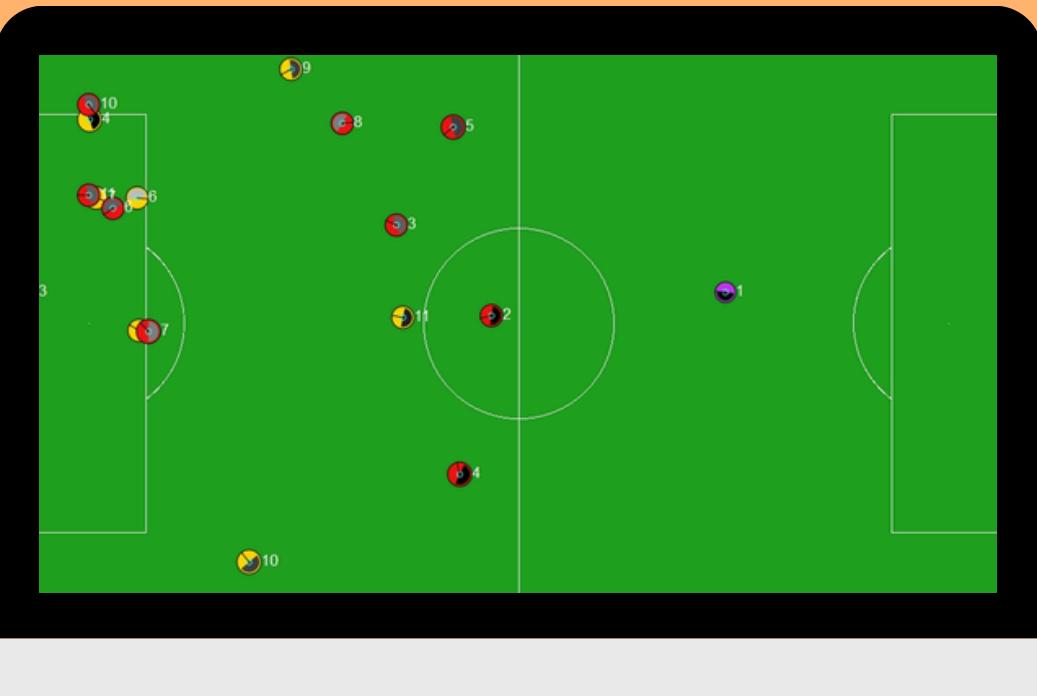
client

client



```
if ($window.scrollTop() > header1_initialDistance) {
    if (parseInt(header1.css('padding-top'), 10) > header1_initialPadding) {
        header1.css('padding-top', '' + $window.scrollTop() - header1_initialPadding);
    } else {
        header1.css('padding-top', '' + header1_initialPadding + 'px');
    }
}

if ($window.scrollTop() > header2_initialDistance) {
    if (parseInt(header2.css('padding-top'), 10) > header2_initialPadding) {
        header2.css('padding-top', '' + $window.scrollTop() - header2_initialPadding);
    } else {
        header2.css('padding-top', '' + header2_initialPadding + 'px');
    }
}
```

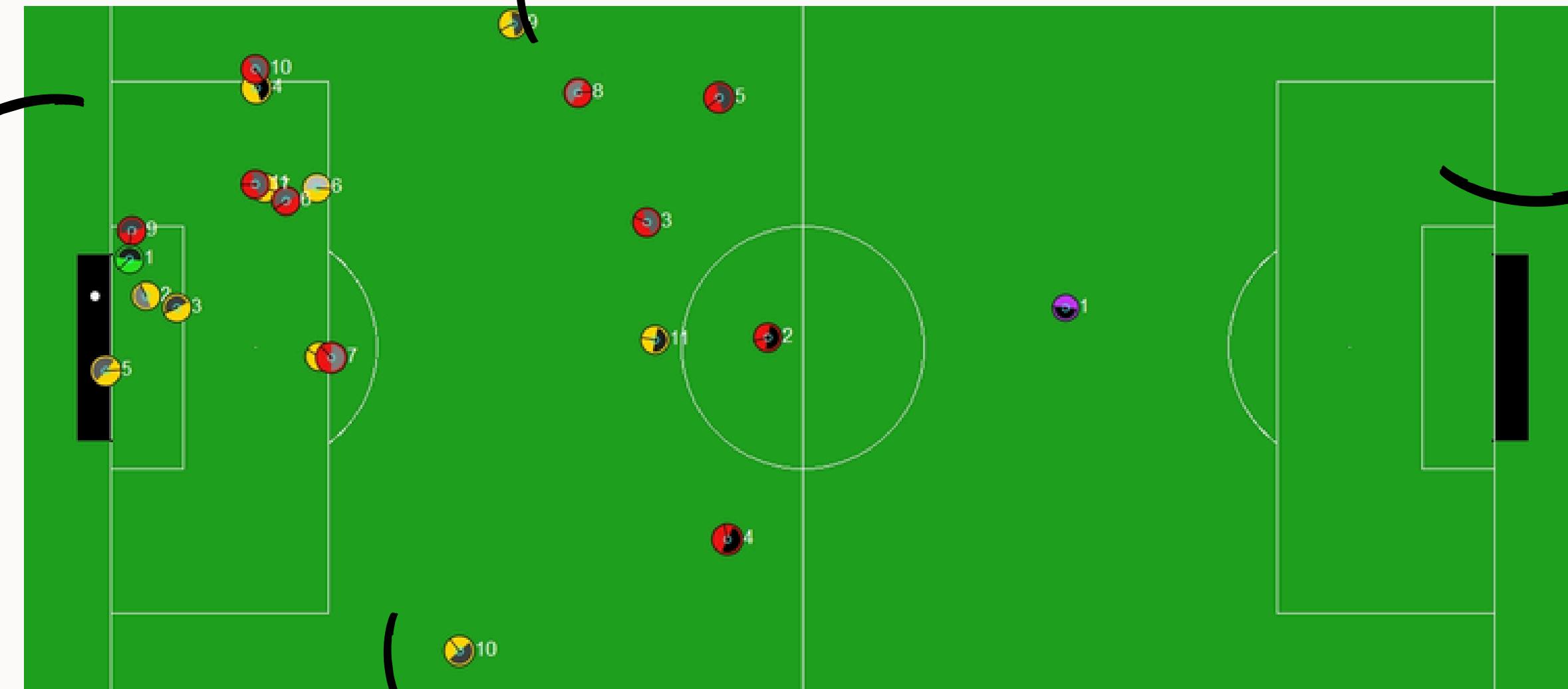


```
products: storeProducts

render() {
    return (
        <React.Fragment>
            <div className="py-5">
                <div className="container">
                    <Title name="our" title="product">
                        <div className="row">
                            <ProductConsumer>
                                {(value) => {
                                    console.log(value)
                                }}
                            </ProductConsumer>
                        </div>
                    </Title>
                </div>
            </div>
        <React.Fragment>
    )
}
```

# SIMULADO

Os jogadores têm um campo de visão de **45°, 90° ou 180°**



Os jogadores enviam ações para movimentação, direção e extensão do campo de visão a cada **100 ms**

Os jogadores recebem informação sobre energia, decisões do juiz e estado do jogo a cada **100 ms**

Os jogadores recebem informação a cada **75 ms, 150 ms ou 300 ms**

ANÁLISE DE DADOS

MÓDULO 2

# CONCEITOS

## DEFINIÇÃO

Análise de Dados é o nome do processo de coleta, limpeza, armazenamento e interpretação de dados com o objetivo de obter informação e tomar decisões com base nelas.

## COLETA DE DADOS

- Dados podem vir de diversas fontes
- Dados podem ser **estruturados** e **não estruturados**
- Dados precisam ser corretos e relevantes para análise

## LIMPEZA DE DADOS

Envolve remover valores duplicados, inconsistentes e errôneos

# CONCEITOS

## ARMAZENAMENTO E ORGANIZAÇÃO DE DADOS

Ordenar e agrupar dados é importante para revelar padrões e tendências nos dados.

## ANÁLISE E INTERPRETAÇÃO DE DADOS

Para entender os dados, utilizamos de métodos estatísticos e ferramentas matemáticas para extrair informação relevante e descobrir como os dados se relacionam entre si e com seu contexto.

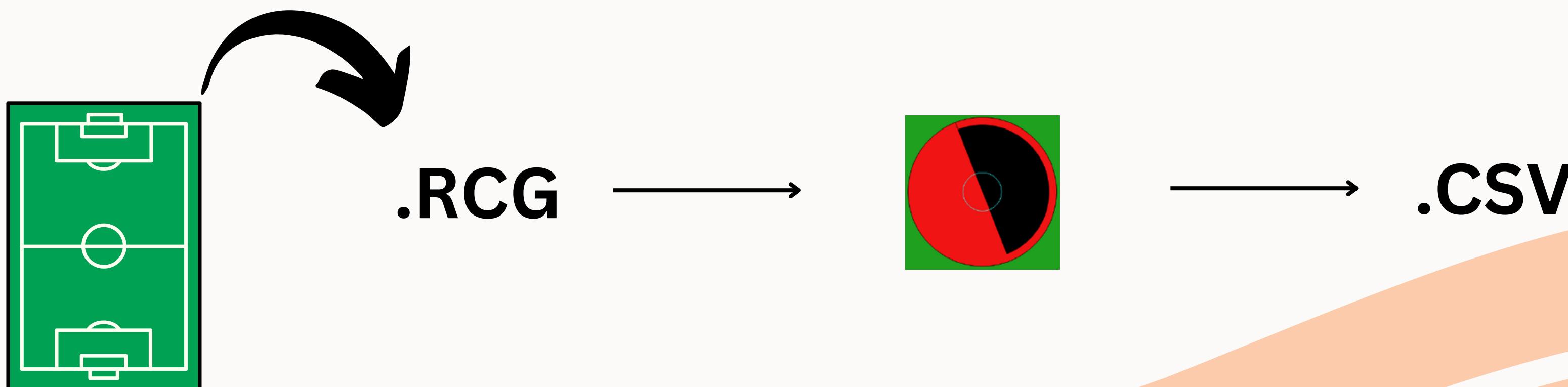
A partir de gráficos e outros recursos visuais, podemos apresentar os dados e o resultado das análises de forma clara e simples para

MÃO NA MASSA

**MÓDULO 3**

# DATASET

- Coletados a partir de partidas do simulado 2D e convertidos para o formato csv.
- Representam eventos que ocorrem durante a partida e contém a posição da bola e dos jogadores a cada ciclo do servidor, entre outras informações.



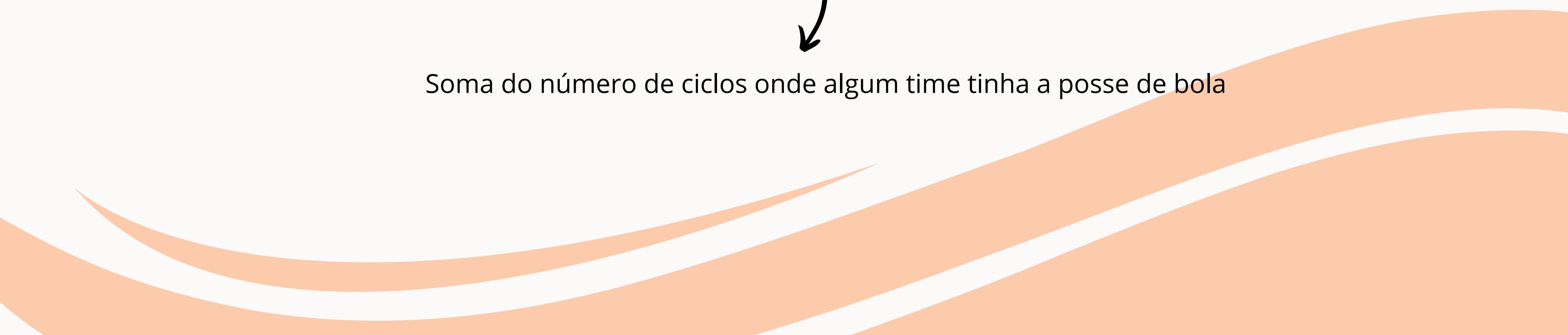
# POSSE DE BOLA

$$P_t = \frac{T_t}{T_{Total}} * 100$$

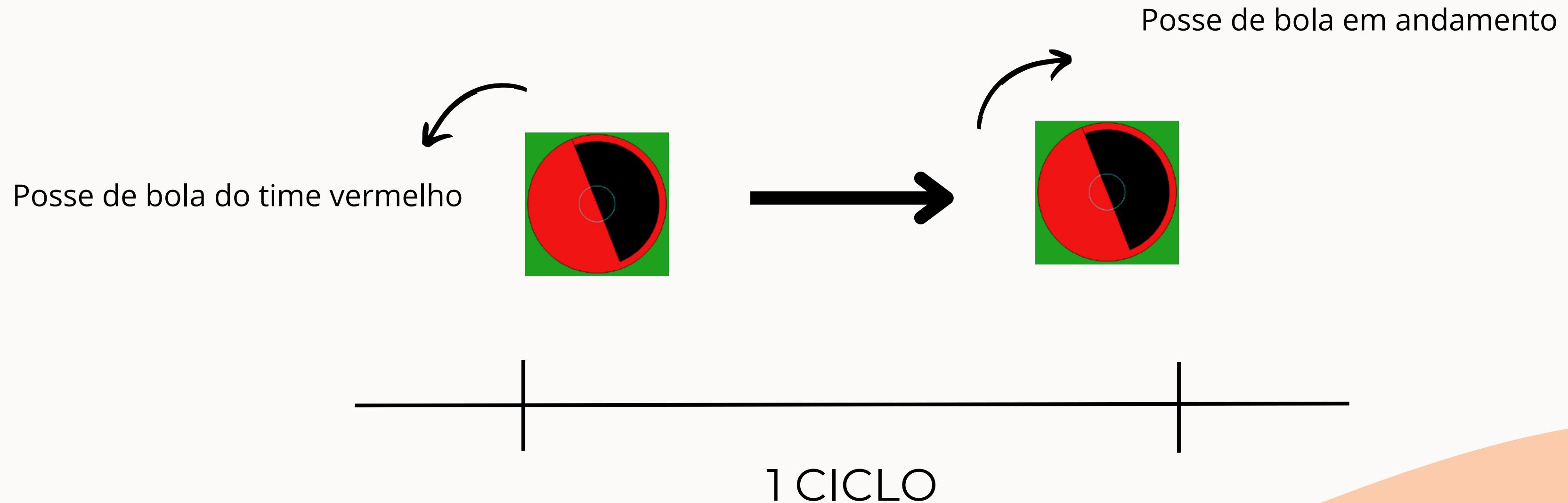
Posse de bola de um time em uma partida

Número de ciclos onde o time tinha a posse de bola

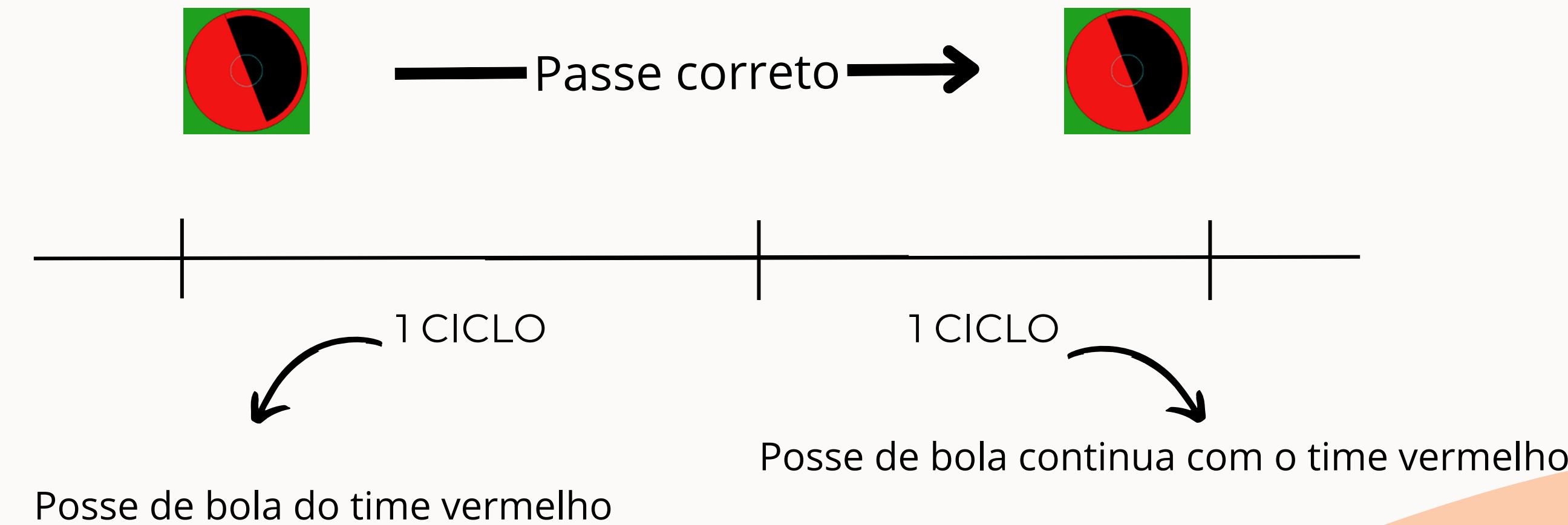
Soma do número de ciclos onde algum time tinha a posse de bola



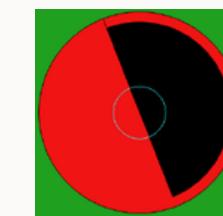
# PASSES



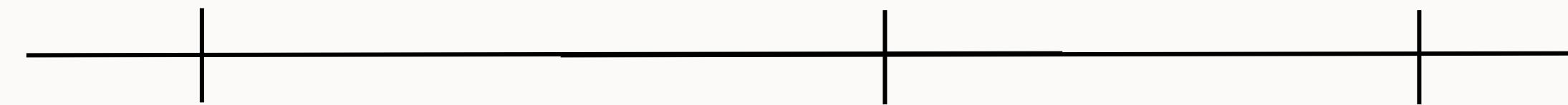
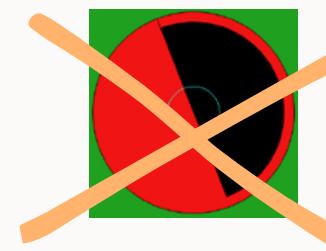
# PASSES



# PASSES



— Passe errado →



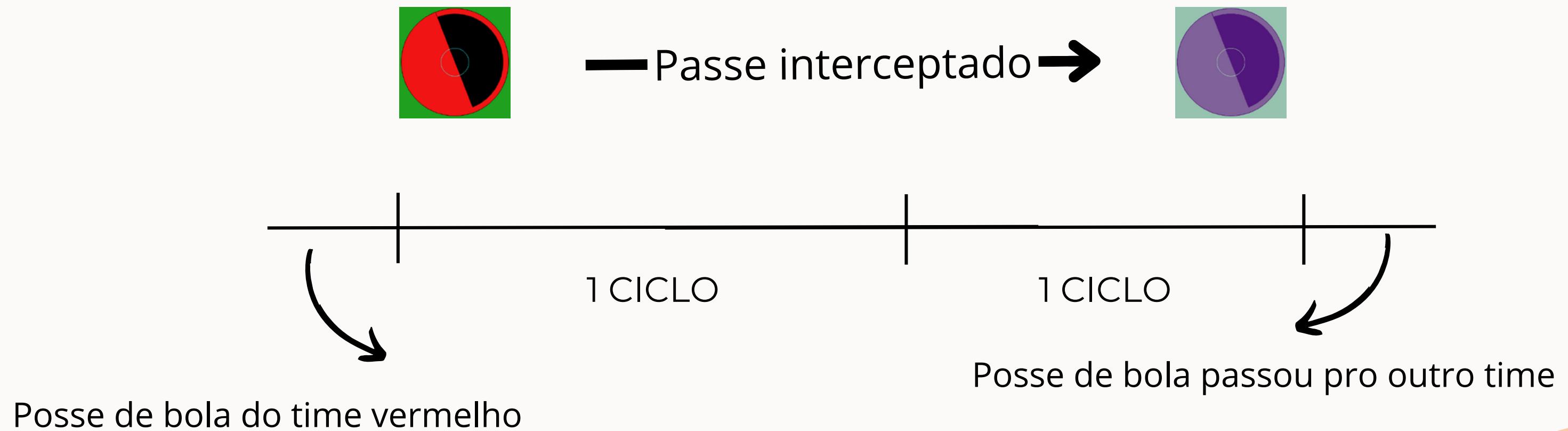
Posse de bola do time vermelho

1 CICLO

1 CICLO

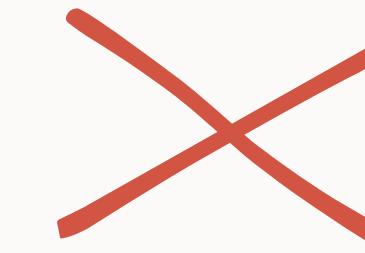
Posse de bola com nenhum dos times

# PASSES



# GOLS

Gol concedido



Gol defendido



# CHUTE A GOL

contagem do número de chutes sofreu alteração

- ✓ Um jogador foi identificado como atacante

- ✓ O jogador pertence ao time vermelho

- ✓ O jogador está pelo menos na metade do time do adversário

- ✓ A bola está em movimento em direção ao gol

componente horizontal  
não é nula

# STAMINA

## DEFINIÇÃO

Define o vigor ou energia de cada jogador durante a partida. Todo jogador começa com uma quantidade máxima de stamina na partida.

## CONSUMO

- É consumida a cada ação que envolve acelerar o robô em determinada direção.
- Acelerar pra trás gera maior consumo de stamina.
- Existe um número mínimo de stamina para um jogador poder agir.
- Caso a stamina de um jogador esteja abaixo do mínimo, pode ser usada uma quantidade extra de stamina disponível.

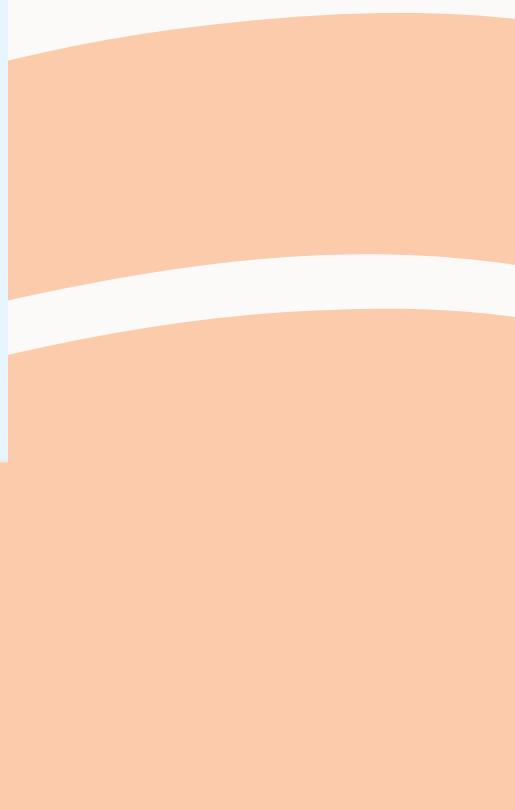
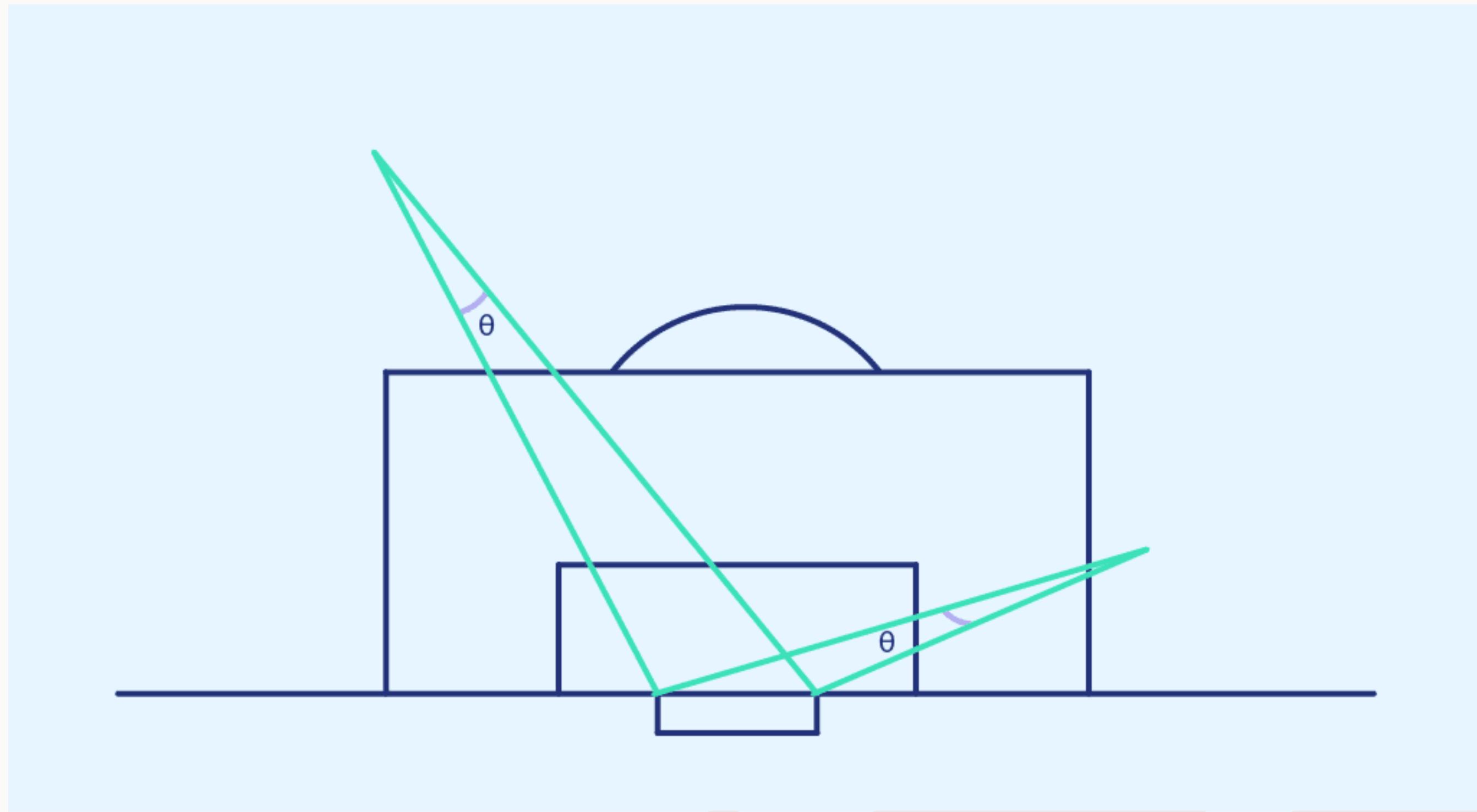
# STAMINA

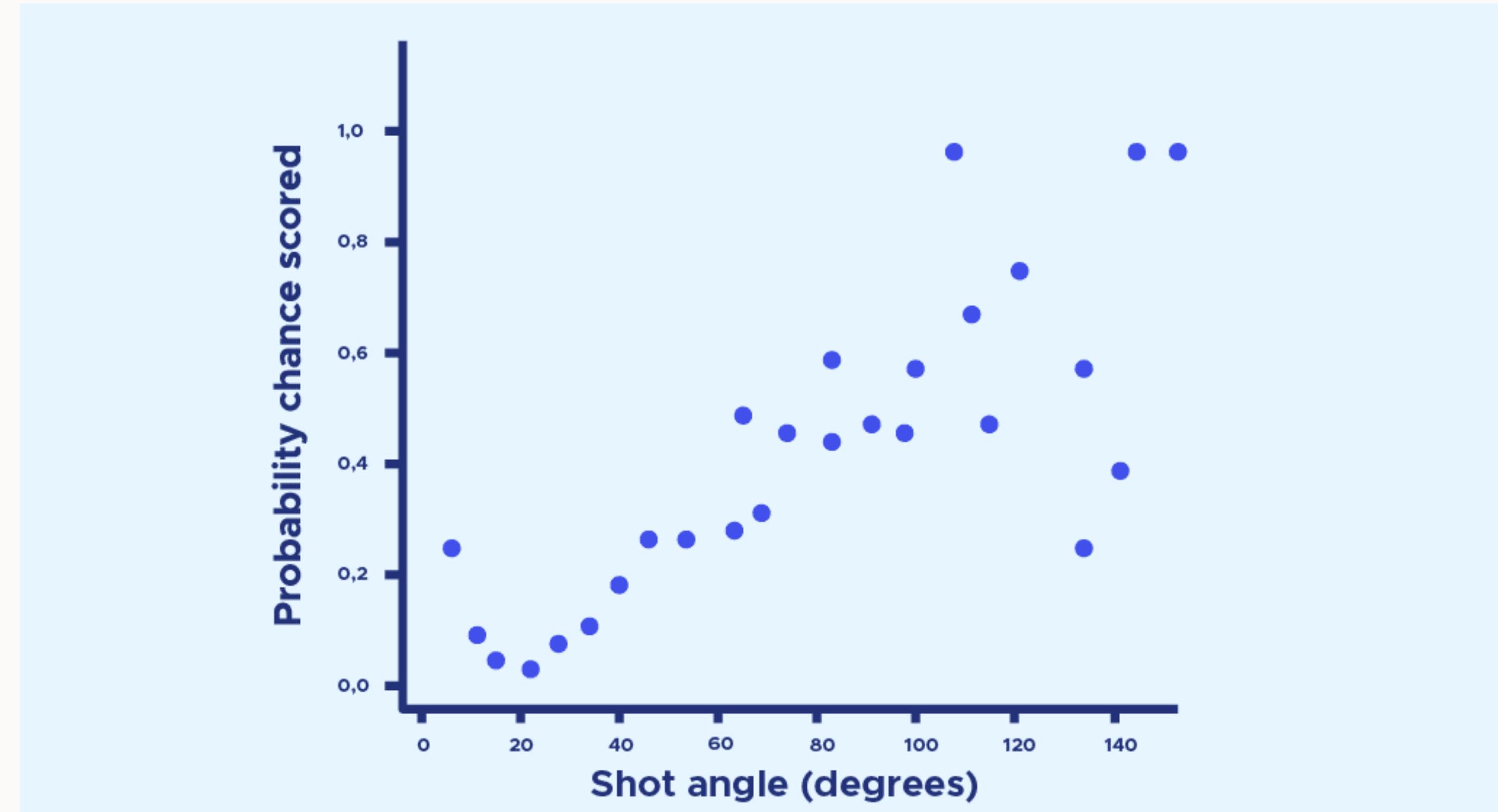
## RECUPERAÇÃO

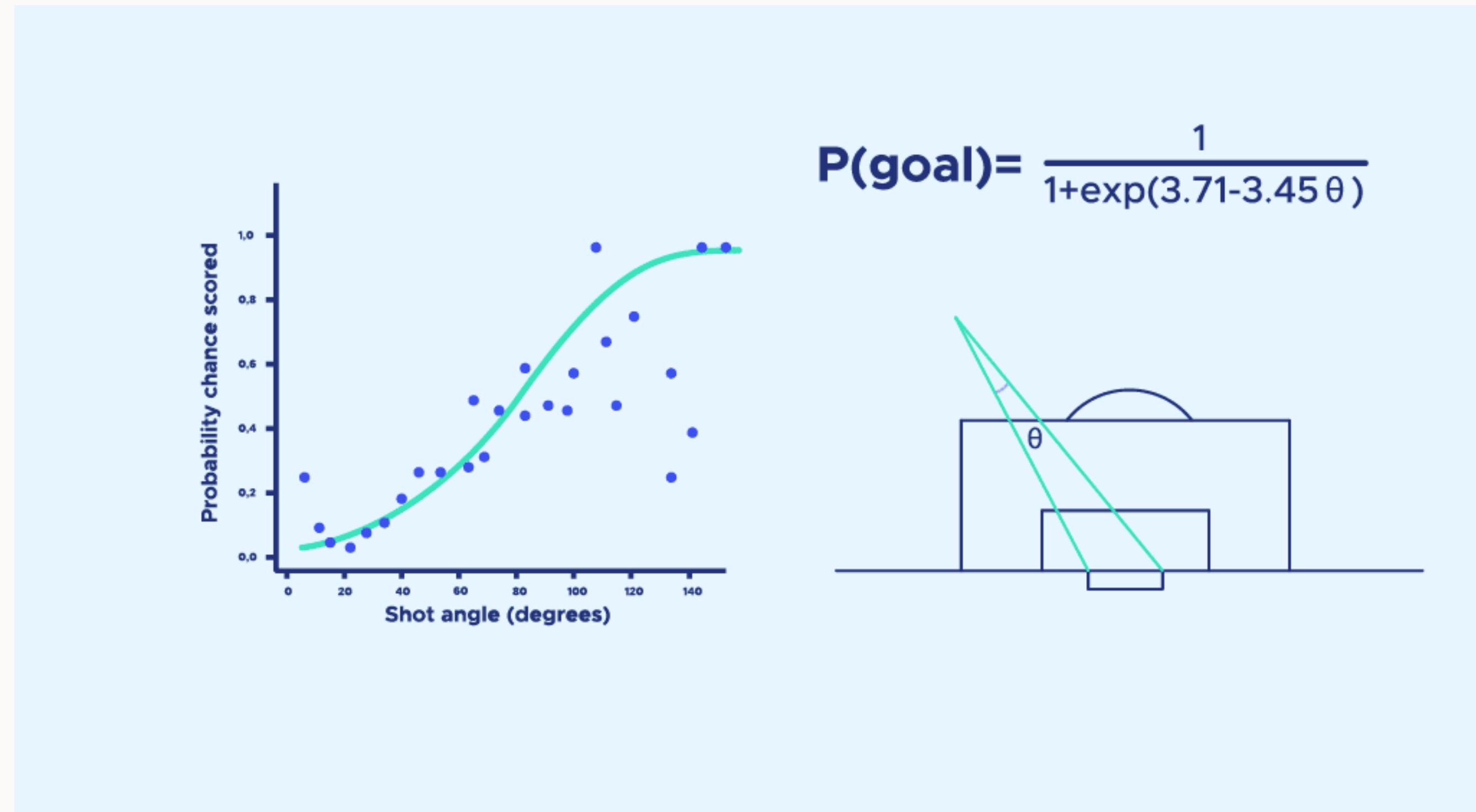
- É recuperada uma taxa de stamina em cada ciclo da partida.
- Durante o intervalo, essa é resetada para o valor inicial.
- O valor de stamina que é recuperada a cada ciclo vai diminuindo ao longo da partida até atingir um valor mínimo.
- Quando o nível de stamina de um chegador está abaixo do valor mínimo, o valor de esforço para agir é alterado para o valor máximo.

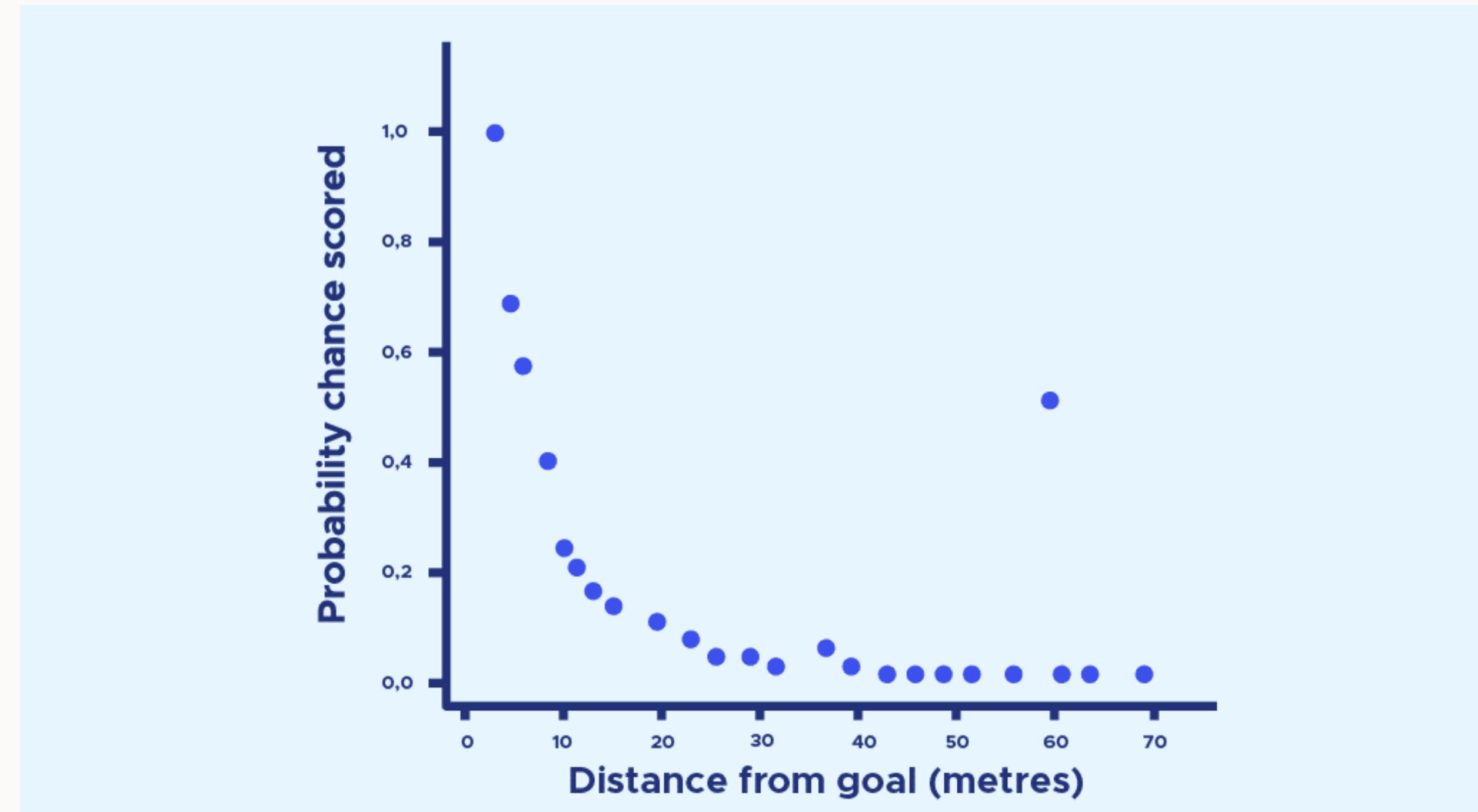


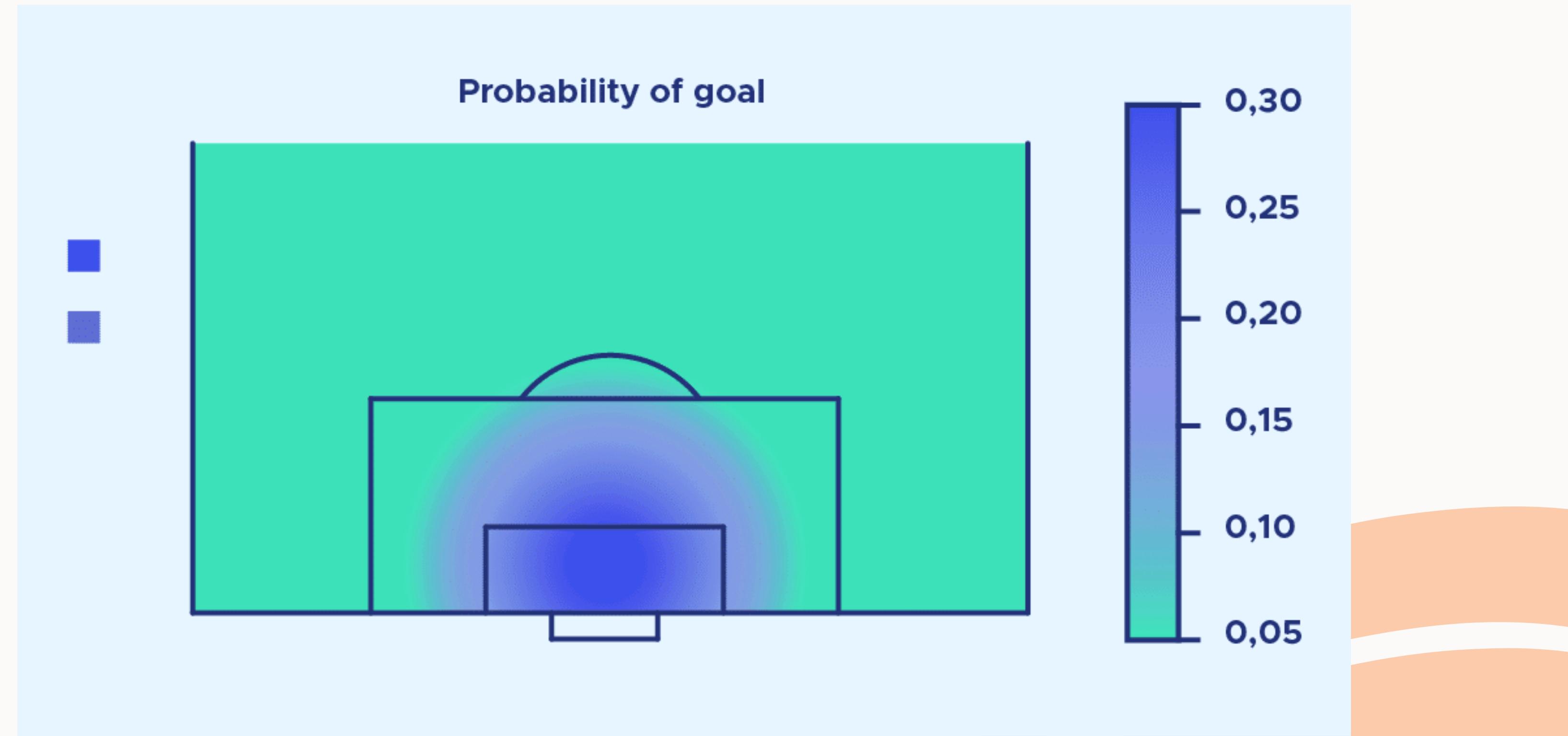
# MATERIAL COMPLEMENTAR











# REFERÊNCIAS

- F. N. A. Pereira, M. F. B. Soares and E. N. S. Barros, "A Data Analysis Graphical user Interface for RoboCup 2D Soccer Simulation League," 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE), Natal, Brazil, 2020, pp. 1-6, doi: 10.1109/LARS/SBR/WRE51543.2020.9307104.  [> Link <](#)
- RobôCIn, "RobôCIn Soccer Analyzer," v2.0.0, 2021, GitHub Repository.  [> Link <](#)
- J. VanderPlas, "Python Data Science Handbook: Essential Tools for Working with Data," O'Reilly Media, Inc., 2016.  [> Link <](#)
- M. Krause, "Expected goals: Understand the math behind it", DataScientest, 2024.  [> Link <](#)
- M. CrabTree, A. Nehme, "What is Data Analysis? An Expert Guide With Examples," Datacamp, 2024.  [> Link <](#)