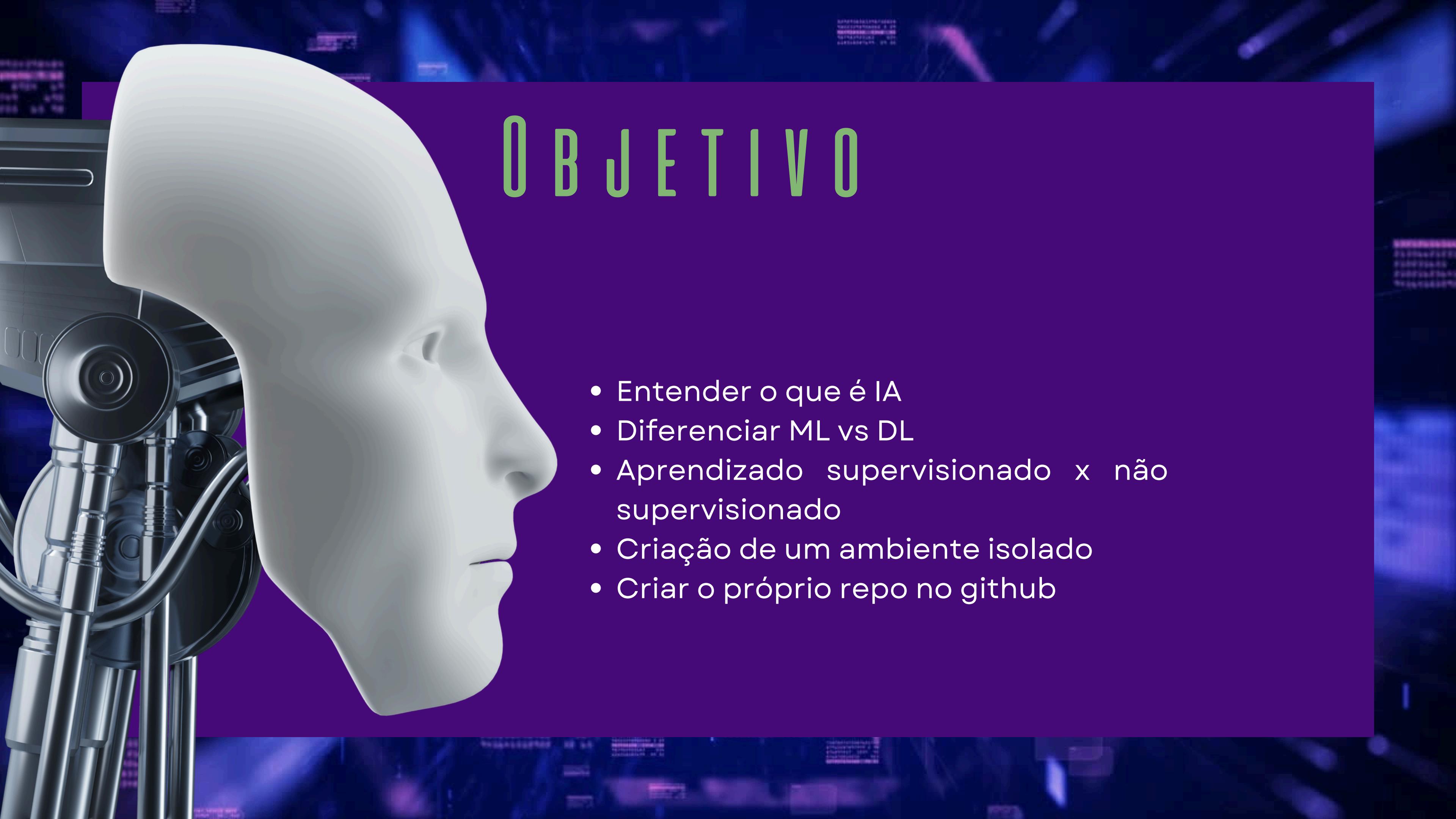


# AULA 1

## FUNDAMENTOS DE IA, PREPARAÇÃO DE AMBIENTE E VERSIONAMENTO





# OBJETIVO

- Entender o que é IA
- Diferenciar ML vs DL
- Aprendizado supervisionado x não supervisionado
- Criação de um ambiente isolado
- Criar o próprio repo no github



# O QUE É INTELIGÊNCIA ARTIFICIAL?

Inteligência Artificial é uma ciência que visa realizar tarefas que normalmente exigiriam inteligência humana, como percepção, raciocínio, aprendizagem e tomada de decisão.

Objetivo da IA: Criar sistemas capazes de pensar, aprender, adaptar-se e agir de maneira inteligente.

# SUBCAMPOS DA IA

**Redes Neurais Artificiais:** Sistemas que simulam o cérebro humano para aprender com grandes volumes de dados.

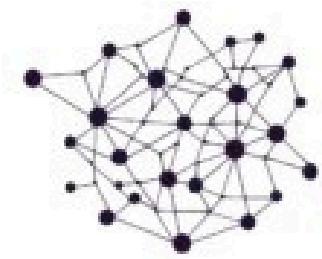
**Processamento de Linguagem Natural (PLN):** Permite que máquinas compreendam e interajam com a linguagem humana.

**Visão Computacional:** Faz com que máquinas interpretem imagens e vídeos do mundo real.

**Machine Learning (Aprendizado de Máquina):** Sistemas que aprendem e melhoram automaticamente a partir de dados.

**Robótica Autônoma:** Robôs que executam tarefas complexas de forma independente, usando IA.

**Processamento de Áudio e Voz:** Tecnologias que entendem e interpretam sons e fala humana.



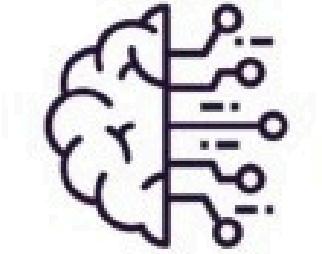
Redes Neurais  
Artificiais



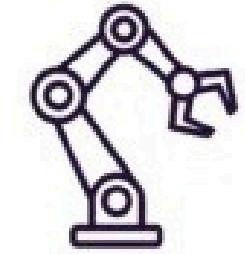
Processamento  
de linguagem  
natural



Visão  
Computacional



Machine  
Learning

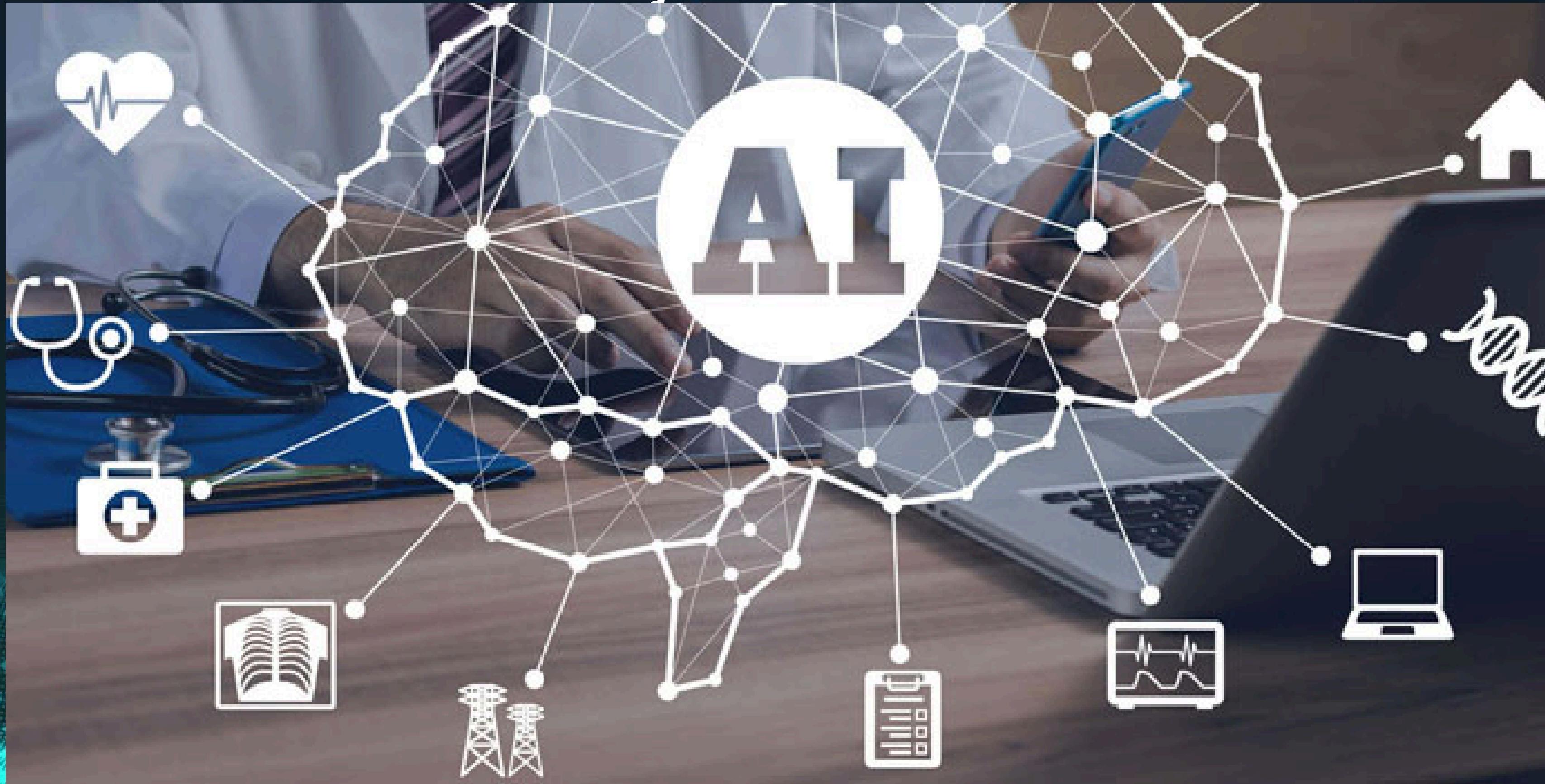


Robótica  
Autônoma



Processamento  
de áudio e voz

# APLICAÇÕES DA IA



# APLICAÇÕES DA IA

- Cidades Inteligentes: Gestão de tráfego, coleta de lixo com a IA planejando a rota mais eficiente para os caminhões de lixo, garantindo que as lixeiras cheias sejam esvaziadas sem que os caminhões percam tempo em lixeiras vazias.
- Agricultura: Monitoramento de plantações, automação de equipamentos...
- Automação: Controle de qualidade → Câmeras em linhas de produção identificando defeito em peças.
- Saúde: Monitoramentos de pacientes com IA ajudando a reconhecer problemas

# Tipos de aprendizado

Dentro da Inteligência Artificial, existem conceitos fundamentais que precisamos entender antes de iniciar qualquer projeto.

Entre eles, destacam-se:

- ◆ Aprendizado Supervisionado
- ◆ Aprendizado Não Supervisionado

Ao iniciar um projeto, precisamos responder:

- Temos dados com resposta correta?
- Qual é o objetivo do problema?
- Queremos prever algo ou descobrir padrões?

# Tipos de aprendizado

## Aprendizado Supervisionado:

O modelo aprende com dados que já possuem a resposta correta.

Entrada (X) + Saída (y conhecida)

Usado para:  
Classificação  
Regressão  
Previsões

## Aprendizado Não Supervisionado

O modelo recebe apenas os dados, sem respostas.  
Entrada (X)

Usado para:  
Agrupamento  
Clusterização  
Descoberta de padrões

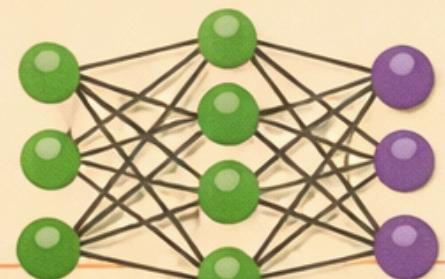
# Tipos de aprendizado

## Aprendizado Supervisionado

### Dados de Treinamento



### Modelo de Treinamento



Treinamento Com Rótulos

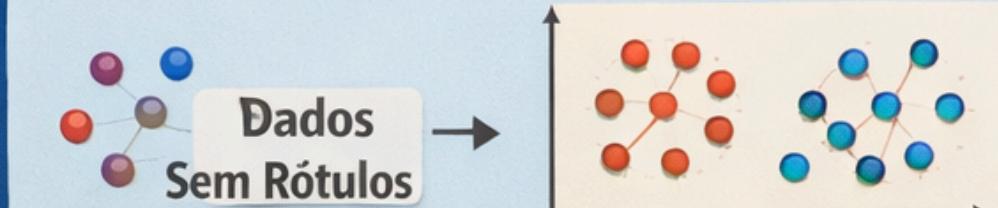
### Classificação ou Previsão



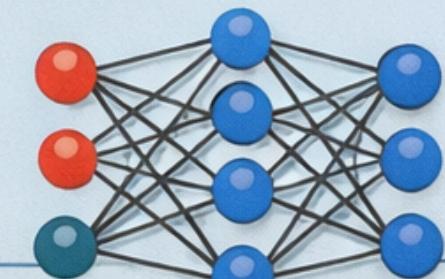
Com Rótulos

## Aprendizado Não Supervisionado

### Dados de Treinamento



### Modelo de Treinamento



Treinamento Sem Rótulos

### Agrupamento ou Padrões



Grupo A      Grupo B  
Encontrando Estruturas

Sem Rótulos

# Deep Learning(DL) x Machine learning(ML)

ML:

**Algoritmos que aprendem padrões a partir dos dados.**

**Normalmente precisam que o humano selecione as características (features).**

**Ex: Árvore de Decisão, Regressão, SVM.**

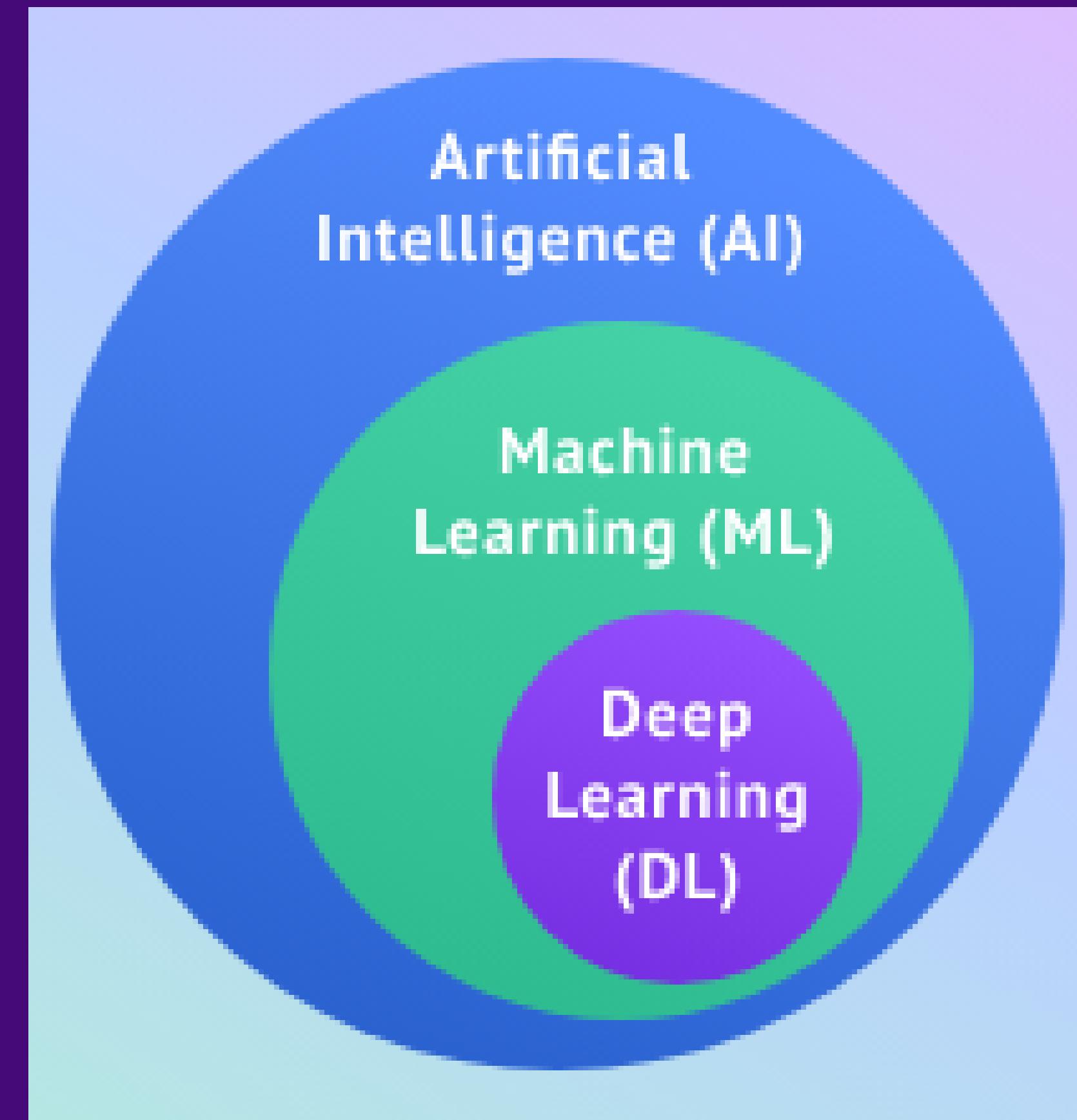
Deep Learning (DL)

**Subárea do Machine Learning baseada em Redes Neurais Profundas.**

**O próprio modelo aprende automaticamente as características.**

**Ex: CNN, RNN, Transformers.**

# Deep Learning(DL) x Machine learning(ML)



# Aplicações DL X ML

## Machine Learning (ML)

**Previsão de preço de imóveis**

**Entrada:**

- **Tamanho da casa**
- **Número de quartos**
- **Localização**

**Saída:**

- **Valor estimado**

## Deep Learning (DL)

**Reconhecimento de imagens  
(ex: detectar rosto)**

**Entrada:**

- **Imagen bruta (pixels)**

**Saída:**

- **Identificação do objeto ou pessoa**

**A rede aprende automaticamente quais padrões da imagem são importantes.**

# AMBIENTES VIRTUAIS

Um ambiente virtual é um espaço isolado dentro do seu computador

onde instalamos as bibliotecas de um projeto sem afetar outros projetos.

Ele funciona como uma “caixinha separada” para cada aplicação.

Dentro dessa caixinha:

- Instalamos apenas as dependências necessárias
- Controlamos as versões das bibliotecas
- Mantemos o projeto independente do sistema

# AMBIENTES VIRTUAIS

Por que devemos usar o ambiente virtual?



## Sem ambiente virtual:

- Conflitos entre versões de bibliotecas
- Projetos quebrando inesperadamente
- Instalações desorganizadas no sistema

## Com ambiente virtual:

- Cada projeto fica isolado
  - Podemos usar versões diferentes da mesma biblioteca
  - Garantimos reprodutibilidade
  - Evitamos que um projeto quebre o outro
- 📌 Em projetos reais de IA, isso não é opcional – é obrigatório.

# GIT AND GITHUB

## O que é Git?

O Git é um sistema de controle de versão.

Ele permite:

Salvar versões do código

---

Acompanhar mudanças

Voltar para versões anteriores

Trabalhar em equipe

📌 É como um “histórico inteligente” do projeto.

## O que é GitHub?

O GitHub é uma plataforma online onde armazenamos projetos que usam Git.

Ele permite:

---

Compartilhar código

- Colaborar com outras pessoas
- Criar portfólio profissional
- Armazenar projetos na nuvem

# G I T A N D G I T H U B

## Por que usar Git e GitHub?

### Sem Git:

- Perdemos versões antigas
- Não sabemos o que foi alterado
- Trabalhar em equipe vira bagunça

### Com Git:

---

- Cada alteração fica registrada
  - Podemos voltar atrás se algo der errado
  - Organizamos o desenvolvimento
  - Seguimos padrão profissional
- 📌 Em projetos de IA e tecnologia, versionamento é essencial.

# BOAS PRÁTICAS

## Estratégias de Gerenciamento de Repositório Git

### Arquivo .gitignore



Especifica arquivos a serem ignorados pelo Git

### Mensagens de Commit

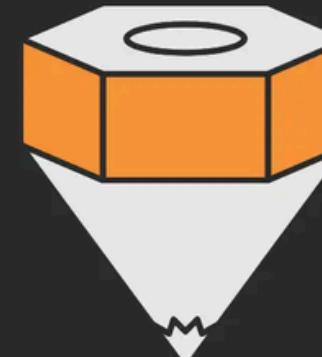
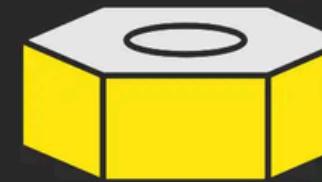
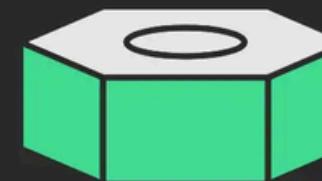


Fornece histórico claro de mudanças

### Tags



Marca pontos de referência importantes na história do repositório



### Estratégias de Branch



Organiza o fluxo de desenvolvimento

### Documentação



Oferece visão geral e instruções do projeto

# COMANDOS BÁSICOS



## Comandos Básicos do Git e GitHub

Comando	Para que serve
<b>git clone URL</b>	Baixar um repositório do GitHub para sua máquina
<b>git add .</b>	Preparar alterações feitas no projeto
<b>git commit -m "mensagem"</b>	Salvar a alteração no histórico
<b>git push</b>	Enviar as mudanças para o GitHub
<b>git pull</b>	Atualizar seu projeto com o que está no GitHub
<b>git branch</b>	Criar/listar ramificações de desenvolvimento
<b>git checkout</b>	Trocar de branch
<b>git merge</b>	Unir mudanças de uma branch
<b>git fork (no GitHub)</b>	Criar uma cópia do projeto para contribuir
<b>git fork (no GitHub)</b>	Criar uma cópia do projeto para contribuir
<b>git fork (no GitHub)</b>	Criar uma cópia do projeto para contribuir
<b>Pull Request (no GitHub)</b>	Solicitar que suas alterações sejam integradas



```
# The First 10 GIT COMMANDS I Type in Every New Project

git init
git status
git add .
git commit -m "Initial commit"
git branch -M main
git remote add origin https://github.com/username/repository.git
git push -u origin main
git log
```