



# IA para Devs

Minicurso de Inteligência Artificial aplicada ao desenvolvimento de software

Projeto: Sistema do Restaurante Universitário do IFMA

IFMA • ~40min teoria + prática ao vivo

# Quem sou eu?

## Wesley Eduardo de Oliveira Melo

Desenvolvedor Java Sênior • Consultor de TI • Entusiasta de IA

### Carreira

Desenvolvedor Java Sênior na CVC Corp  
12 anos de experiência em desenvolvimento  
Consultor de TI na Mentor

### Formação

Bacharel em Sistemas de Informação — IFMA  
Publicação acadêmica na área de  
Inteligência Artificial

### IA no dia a dia

Uso IA generativa diariamente para potencializar o desenvolvimento — tanto nos projetos profissionais quanto nos pessoais em andamento. O que vou mostrar neste curso é o mesmo fluxo que uso no meu trabalho real.

# O que vamos construir?

Sistema do Restaurante Universitário — do zero ao deploy



## Frontend

React + Vite  
TailwindCSS



## Backend

Spring Boot 4  
Java 21



## Banco

PostgreSQL 17  
via Docker



## Deploy

Vercel + Railway

### Página Pública

- Cardápio do dia (almoço e jantar)
- Navegação entre datas
- Calendário para escolher data

### Área Admin

- Login com autenticação
- CRUD de cardápios
- CRUD de administradores

# O que vamos aprender?

Primeiro a teoria, depois a prática — do conceito ao deploy.



## Base Teórica

~40 minutos

1. O que são LLMs
2. Modelos e Tokens
3. Ferramentas e Preços
4. Agentes vs. Chatbots
5. O Ciclo ReAct

## >\_ Prática ao Vivo

~3 horas

1. Setup + Primeira interação
2. Planejamento com o agente
3. Build da V1 completa
4. Deploy (Vercel + Railway)
5. Iterações e melhorias

# O que são LLMs?

Large Language Models — os "cérebros" por trás dos agentes



## Como funciona (de forma simples)

- Uma LLM é um programa treinado com bilhões de textos da internet
- Ela aprendeu padrões de linguagem, lógica e código
- Quando você pergunta, ela prevê a melhor resposta palavra por palavra
- Não "pensa" como humano — mas imita muito bem



## Treinamento

Aprendeu com código,  
livros, sites, artigos...



## Contexto

Quanto mais info você  
dá, melhor a resposta



## Geração

Cria texto novo baseado  
no que aprendeu

# Modelos e Tokens

Dois conceitos essenciais para entender antes de usar qualquer ferramenta



## Modelo

É o "cérebro" da IA. Existem vários tamanhos — modelos maiores são mais inteligentes, mas também mais caros e lentos.

Haiku

Rápido e barato

Sonnet

Equilíbrio ideal

Opus

Máxima qualidade



## Token

É a unidade de texto que a IA processa. Uma palavra tem de 1 a 3 tokens. Você paga pela quantidade de tokens usados.

"Olá mundo" = 2 tokens

💡 Quanto mais tokens uma tarefa usa, mais ela custa. Prompts claros = menos tokens gastos!

# Ferramentas de IA para Código

As principais opções disponíveis hoje e quanto custam

## > Claude Code

Anthropic

Pro US\$ 20/mês

Max 5x US\$ 100/mês

Max 20x US\$ 200/mês

API Por token

◆ Melhor para código complexo

Usado neste curso

Também existem:

GitHub Copilot

US\$ 10+/mês

## </> Gemini CLI

Google

Gratuito 1.000 req/dia

AI Pro US\$ 25/mês

AI Ultra US\$ 250/mês

API Por token

◆ Gratuito e open source

Ótimo pra praticar

## 🤖 Codex CLI

OpenAI

Gratuito Limitado

Plus US\$ 20/mês

Pro US\$ 200/mês

API Por token

◆ Integrado ao ChatGPT

Open source (Rust)

Cursor

US\$ 20+/mês

Windsurf

US\$ 15+/mês

Aider

Gratuito (OSS)

# Agentes vs. Chatbots

O que muda na prática?



## Chatbot

- Responde perguntas
- Gera trechos de código
- Você copia e cola
- Sem contexto do projeto
- Cada conversa é isolada



## Agente de IA

- Lê e edita seus arquivos
- Executa comandos no terminal
- Entende o contexto do projeto
- Trabalha junto com você
- Mantém memória (CLAUDE.md)

# O Ciclo ReAct

Como o agente "pensa" antes de agir



⟳ O ciclo se repete até a tarefa estar completa

# Janela de Contexto

O quanto o agente consegue "lembrar" de uma vez



## O que é?

A janela de contexto é a "memória de trabalho" do agente — tudo que ele consegue enxergar ao mesmo tempo: seu prompt, o código, as respostas anteriores. Quando essa janela enche, ele começa a "esquecer" o início da conversa.

Contexto usado: 70% ← conversa longa, muitos arquivos



### O problema

Em sessões longas, o contexto enche e o agente perde o fio da meada.



### /compact

Compacta a conversa, resuindo o que já foi feito. Libera espaço no contexto.



### CLAUDE.md

Memória persistente que sobrevive entre sessões. O agente sempre lê primeiro.

# Ecossistema do Agente

MCP, Skills e Subagentes — as peças que expandem o que o agente faz



# Como escrever bons prompts

A qualidade do que o agente entrega depende do que você pede

## Prompt vago

"faz uma API pra mim"

Sem contexto, sem stack, sem escopo.  
O agente vai chutar tudo.

## Prompt claro

"Crie um CRUD de cardápios com  
Spring Boot 4 e PostgreSQL 17"

Stack definida, escopo claro.  
O agente sabe o que construir.

## Os 4 ingredientes de um bom prompt:

1

### Contexto

O que o projeto é,  
qual o problema

2

### Stack

Tecnologias, versões,  
padrões de código

3

### Escopo

O que deve fazer  
(e o que NÃO deve)

4

### Critério

Como saber se  
ficou bom

# Quando usar (e quando não usar)

IA é extremamente poderosa — mas exige responsabilidade

## O que ela faz bem

- Criar projetos completos do zero ao deploy
- Acelerar o desenvolvimento (10x mais rápido)
- Gerar e executar testes automatizados
- Debugar erros com contexto do projeto
- Explorar stacks e tecnologias novas

## Cuidados importantes

- Sem revisar — sempre leia o código gerado
- Sem entender — saiba o que ele está fazendo
- Código sensível sem validação de segurança
- Como muleta — o objetivo é aprender junto
- Cegamente — ele erra, e você precisa pegar

Limitações reais que você vai encontrar:

### Alucinação

Inventa bibliotecas e APIs que não existem

### Código inseguro

Funciona, mas pode ter falhas de segurança

### Dependência

Use como ferramenta, não como muleta

### Custo

Projetos grandes consomem muitos tokens

# IA e Código Proprietário

Nem toda empresa permite usar agentes de IA no código



## Por que isso importa?

Quando você usa um agente de IA, o código que você envia pode ser processado em servidores externos. Para empresas com código proprietário, segredos industriais ou dados sensíveis, isso pode ser um problema de compliance, segurança e propriedade intelectual.

### Riscos comuns

- Código enviado para servidores externos
- Possível uso para treinar modelos
- Vazamento de segredos e tokens
- Violação de políticas internas

### Como lidar

- Consulte a política de IA da empresa
- Use em projetos pessoais e estudos
- Prefira planos com zero data retention
- Nunca envie credenciais ou dados reais



# Hora da prática!

Vamos abrir o terminal e construir tudo do zero com o Claude Code

1. Setup do ambiente

2. Planejar com o agente

3. Construir a V1

4. Deploy ao vivo