

Robótica Educacional

Módulo 3



Aula
13

Robótica com I.A. - II

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Ailton Lopes

Cleiton Rosa

Darice Alessandra Deckmann Zanardini

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Kellen Pricila dos Santos Cochinski

Projeto Gráfico, Diagramação e Geração de imagem por I.A.

Edna do Rocio Becker

2024

Sumário

Introdução	2
Objetivos desta aula	2
Roteiro da aula	3
1. Contextualização	3
2. Metodologia da Inteligência Artificial	4
Feedback e finalização	28
Referências	29

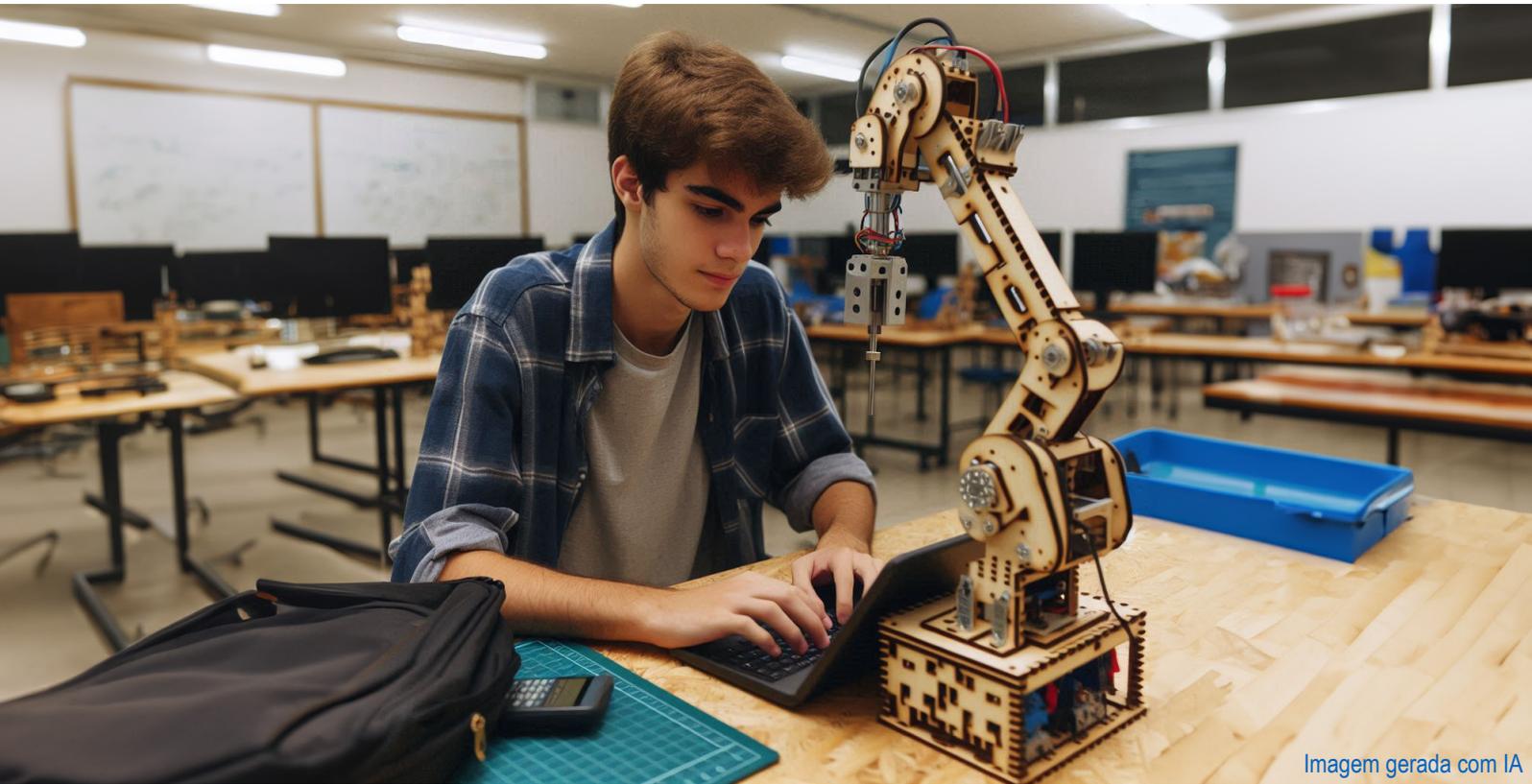


Imagem gerada com IA

Introdução

Na aula anterior, nos debruçamos sobre um novo e fascinante contexto: o da inteligência artificial (IA) e, ao final da aula, sobre as potencialidades da IA no campo educacional. Como nosso foco também é verificar as possibilidades de utilização de IA à Robótica, veremos, nesta aula, alguns exemplos de como a IA pode ser aplicada ao desenvolvimento dos nossos projetos, potencializando nossas experiências.

Porém, antes, vamos verificar como podemos utilizar a Inteligência Artificial para obter os melhores resultados. Desta forma, esse recurso vai se apresentar como um grande aliado em nosso dia a dia.

Objetivos desta aula

- Compreender IA como recurso para elaboração e refinamento de projetos;
- Utilizar IA para aprendizagem de programação e Arduino.

Lista de materiais

- Computador com acesso à internet.

Roteiro da aula

1. Contextualização

Após um fascinante mergulho pelas ideias, conceitos e desenvolvimento da IA, vamos utilizar esse recurso como auxílio para compreender, ajustar e desenvolver códigos de programação para nossos projetos de Robótica.

Você e seus colegas podem estar mais integrados ao assunto da IA e utilizando a ferramenta para auxiliar nos estudos, preparação de projetos e criações... ou pode ser também que tudo seja uma grande novidade e a curiosidade esteja a mil!

Nesta aula, nosso foco será analisar e refinar códigos de programação compartilhados pelo simulador online Wokwi. Cada exemplo que traremos mostrará a você e seus colegas possibilidades mais específicas de aplicação da IA na Robótica e, durante esse nosso percurso, você pode pensar em outras formas de utilização para ampliar as possibilidades de aplicação.



2. Metodologia de Inteligência Artificial

O ChatGPT ou Gemini, grandes modelos de LLM (Large Language Model), combinam grande quantidade de dados e Deep Learning para comunicar e gerar novos conceitos, trabalhando com redes neurais, etc. Para fornecer respostas, esses modelos utilizam-se de comandos chamados de prompts.

Prompt é um comando que você fornece à Inteligência Artificial para que ela gere respostas de acordo com o seu interesse. É como se você fizesse uma pergunta para um amigo e aguardasse a resposta. No contexto da Inteligência Artificial, os prompts são fundamentais para que se obtenha respostas úteis e relevantes.

Agora imagine que você pergunte para esse amigo: "vamos sair para fazer algo divertido?". Você possivelmente não vai receber uma resposta precisa. É provável que essa conversa continue com: "O que você deseja fazer?". Imagine agora se você pergunta: "Traz aquele negócio da última vez?". Seu amigo, no mínimo, vai perguntar: "Que negócio?" "Que última vez?". Para evitar esse tipo de problema, é preciso saber elaborar bem a pergunta.

Tenha clareza nas instruções
para obter as melhores
respostas!

A Inteligência Artificial Generativa é algo novo em termos de prática e ainda estamos aprendendo a lidar com esse recurso para obter os melhores resultados. Pesquisadores e até mesmo empresas como a Google vêm sondando formas de utilizar a Inteligência Artificial e alguns artigos já têm sido publicados em revistas de universidades como a Cornell University (Ithaca, NY - EUA). A seguir, veremos algumas técnicas e exemplos de comandos para a IA executar uma tarefa.

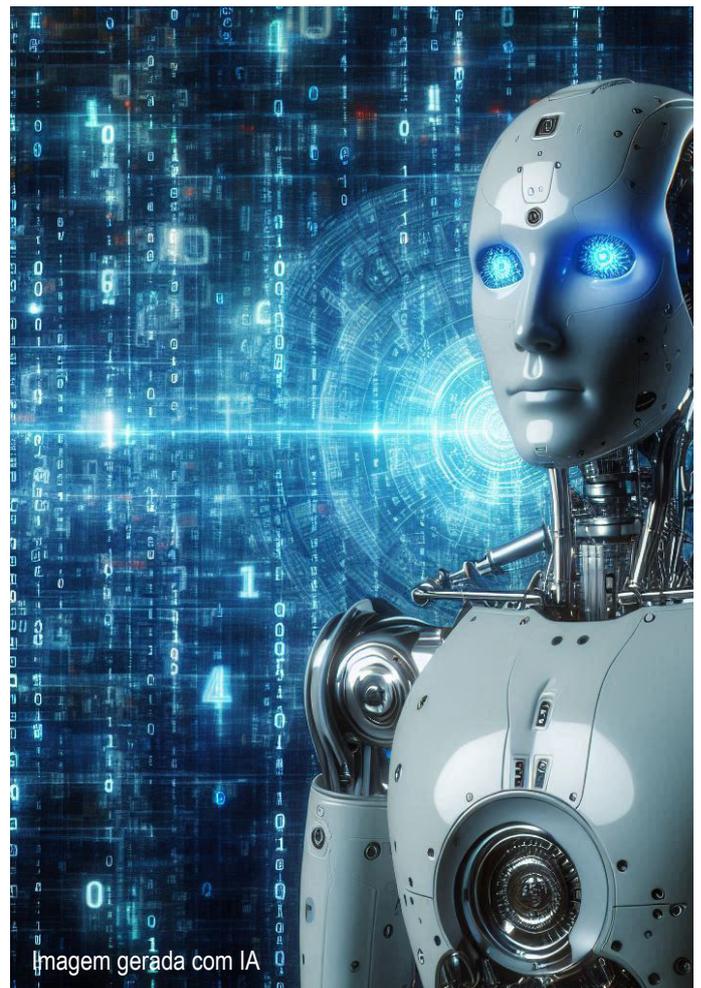


Imagem gerada com IA

■ Zero Shot Prompting

É a técnica de fornecer um comando ao modelo sem haver exemplos para a tarefa desejada: zero (0) shot (exemplo) prompting (ato de enviar um comando). Você faz uma solicitação diretamente ao modelo, confiando em seu conhecimento prévio para gerar a resposta adequada, sem a necessidade de explicação prévia.

Exemplo:

Tarefa: tradução de uma frase do inglês para o espanhol.

Prompt utilizando Zero Shot: "Traduza a seguinte frase para o espanhol: 'I like programming little robots'"

Resposta: "Me gusta programar pequeños robots."

Nesse caso, a solicitação é feita diretamente ao modelo sem fornecer exemplos de tradução, deduzindo que o seu conhecimento adquirido durante o treinamento do modelo (IA) seja o suficiente para dar a resposta mesmo sem exemplos no prompt.

■ One Shot Prompting

O One Shot Prompting é a técnica que dá o comando e fornece apenas um exemplo para o modelo (IA) executar a tarefa. Você vai compreender melhor no próximo item, pois a diferença é apenas na quantidade de exemplos.

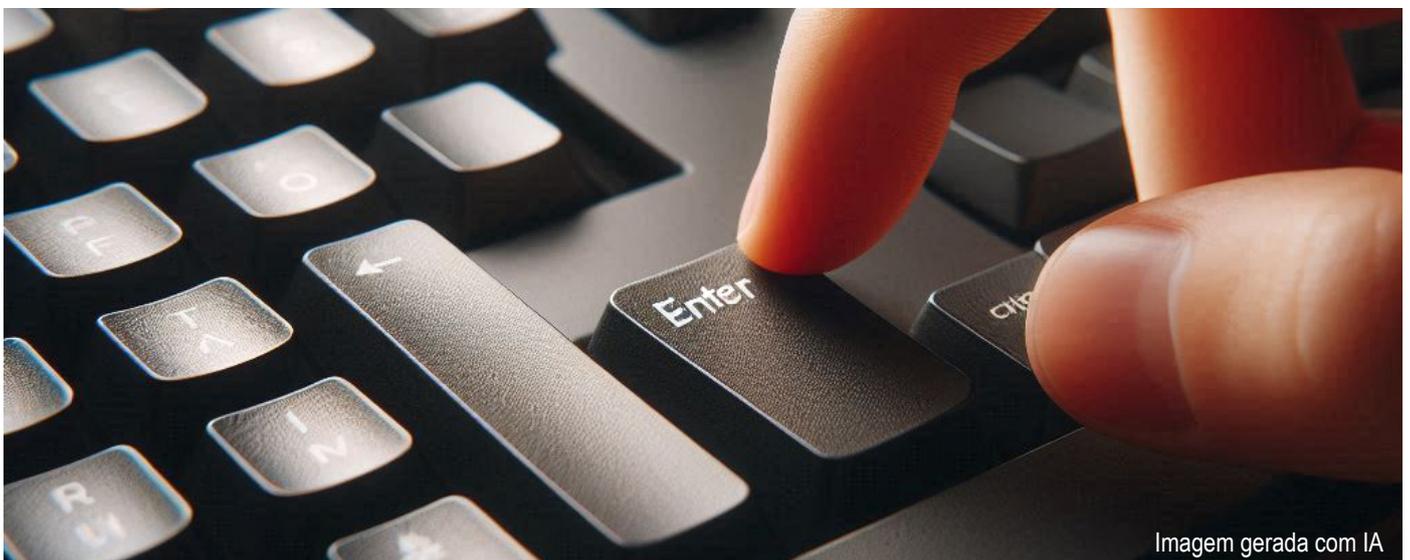


Imagem gerada com IA

■ Few Shot Prompting

É uma técnica de enviar comandos para o modelo de IA que fornece alguns exemplos ao solicitar a tarefa. Few (poucos) Shot (exemplos) Prompting (ato de enviar o comando). Com os exemplos fornecidos, o modelo é capaz de entender melhor a tarefa desejada, ajustando seu comportamento com base nos exemplos e estabelecendo um padrão de respostas.

Exemplo:

Prompt: I like apples.

Me gustan las manzanas.

She is reading a book.

Ella está leyendo un libro.

I like programming little robots.

Resposta: Me gusta programar pequeños robots.

* Prompt simulado no ChatGPT 3.5

O que aconteceu nesse caso?

O modelo (IA) entendeu que existia um padrão: frase em inglês em uma linha, tradução em espanhol na outra linha. Isso se repetiu por duas vezes. Na terceira, ele "entendeu" que deveria completar com a tradução em espanhol. Você pode pensar que isso foi inútil, pois poderia apenas ter optado pela técnica de Zero Shot Prompting pedindo logo para traduzir do inglês para o espanhol, mas imagine que você estivesse automatizando um processo de tradução de centenas de linhas ou quaisquer outros dados que pudessem ser interpretados.

Vamos para outro exemplo:

Prompt:

Texto original: O Brasil é o maior país da América do Sul, com uma área territorial de mais de 8 milhões de km². Sua população ultrapassa 210 milhões de habitantes, composta por pessoas de diversas origens étnicas e culturas. O país é conhecido por sua rica biodiversidade, com florestas tropicais, cerrado, pampas e outros biomas.

Resumo: O Brasil é um país extenso e populoso da América do Sul, com grande diversidade cultural e natural.

Texto original: A inteligência artificial (IA) é um campo da ciência da computação que se dedica à criação de agentes inteligentes, que são sistemas capazes de raciocinar, aprender e agir de forma autônoma. A IA tem um grande potencial para revolucionar diversos setores da sociedade, como saúde, educação, transporte e manufatura.

Resumo: A inteligência artificial é um campo da computação que busca criar sistemas inteligentes que podem aprender e agir de forma autônoma.

Texto original: A Guerra dos Farrapos (1835-1845) foi um conflito armado que ocorreu no sul do Brasil, envolvendo os caboclos da região contra o governo imperial. A guerra teve como principais causas o descontentamento com as altas taxas de impostos e a falta de representação política.

Texto original: A escravidão foi um dos pilares da economia brasileira durante o período colonial e imperial. Milhares de africanos foram sequestrados e trazidos para o Brasil para trabalhar em plantações e engenhos. A escravidão teve um impacto devastador na vida dos africanos e seus descendentes, deixando um legado de sofrimento e desigualdade que ainda persiste nos dias de hoje.

Texto original: A descoberta do Brasil pelos portugueses em 1500 marcou o início de um longo processo de colonização e exploração do território. Os colonizadores europeus trouxeram consigo sua cultura, religião e costumes, o que resultou em um processo de aculturação e miscigenação entre os povos indígenas, africanos e europeus. Ao longo dos séculos, o Brasil passou por diversas fases históricas, como o período colonial, o Império e a República. Cada um desses períodos foi marcado por eventos importantes que moldaram a identidade nacional brasileira.

Resposta:**Resumo dos textos originais:**

Texto original: A Guerra dos Farrapos (1835-1845) foi um conflito armado que ocorreu no sul do Brasil, envolvendo os caboclos da região contra o governo imperial. A guerra teve como principais causas o descontentamento com as altas taxas de impostos e a falta de representação política.

Resumo: A Guerra dos Farrapos (1835-1845) foi um conflito no sul do Brasil, motivado por altas taxas de impostos e falta de representação política.

Texto original: A escravidão foi um dos pilares da economia brasileira durante o período colonial e imperial. Milhares de africanos foram sequestrados e trazidos para o Brasil para trabalhar em plantações e engenhos. A escravidão teve um impacto devastador na vida dos africanos e seus descendentes, deixando um legado de sofrimento e desigualdade que ainda persiste nos dias de hoje.

Resumo: A escravidão foi central na economia brasileira colonial e imperial, resultando em um legado duradouro de sofrimento e desigualdade.

Texto original: A descoberta do Brasil pelos portugueses em 1500 marcou o início de um longo processo de colonização e exploração do território. Os colonizadores europeus trouxeram consigo sua cultura, religião e costumes, o que resultou em um processo de aculturação e miscigenação entre os povos indígenas, africanos e europeus. Ao longo dos séculos, o Brasil passou por diversas fases históricas, como o período colonial, o Império e a República. Cada um desses períodos foi marcado por eventos importantes que moldaram a identidade nacional brasileira.

Resumo: A colonização do Brasil pelos portugueses em 1500 iniciou um processo de aculturação e miscigenação, passando por períodos colonial, imperial e republicano, que moldaram a identidade nacional.

* prompt simulado no ChatGPT 3.5

■ Chain-of-Thought (CoT)

O Chain of Thought Prompting (CoT) - Cadeia de Pensamento é um método que encoraja o modelo (IA) a explicar passo a passo o porquê da resposta dada. Ele não entrega apenas uma resposta pronta e sim explica cada etapa.

Exemplo:

Few Shot Prompting	Chain of Thought Prompting
<p>Q: Edgar tem 8 bolas de tênis. Ele compra mais 3 latas de bolas de tênis. Cada lata tem 3 bolas de tênis. Quantas bolas de tênis Edgar tem agora?</p> <p>A: A resposta é 17.</p> <p>Q: O restaurante possui 45 tomates. Se eles utilizarem 20 tomates para fazerem almoço e comprarem mais 7, quantos tomates eles terão agora?</p>	<p>Q: Edgar tem 8 bolas de tênis. Ele compra mais 3 latas de bolas de tênis. Cada lata tem 3 bolas de tênis. Quantas bolas de tênis Edgar tem agora?</p> <p>A: Edgar começa com 8 bolas de tênis. 3 latas \times 3 bolas por lata = 9 bolas. 8 bolas+9 bolas = 17 bolas. Portanto, Edgar tem agora 17 bolas de tênis.</p> <p>Q: O restaurante possui 45 tomates. Se eles utilizarem 20 tomates para fazerem almoço e comprarem mais 7, quantos tomates eles terão agora?</p>
<p>Resposta do Modelo (IA):</p> <p>A resposta é 32.</p>	<p>Resposta do Modelo (IA):</p> <p>O restaurante começa com 45 tomates. Eles utilizam 20 tomates para fazer o almoço. Portanto, devemos subtrair esses 20 tomates dos 45 iniciais: $45 - 20 = 25$. O restaurante compra mais 7 tomates. Portanto, devemos adicionar esses 7 tomates aos 25 que restaram após o almoço: $25 + 7 = 32$. Assim, o número de tomates que o restaurante terá agora é 32.</p>

Como foi possível perceber, com o método Chain of Thought, o modelo (IA) fornece uma resposta mais completa, identificando e explicando cada etapa da resposta.

Esse tipo de método pode ser muito útil no aprendizado de programação e Robótica, uma vez que você poderá compreender etapas que está em dúvida. Você pode habilitar esse modo solicitando ao modelo: "mostre o passo a passo".

Segredos para elaborar um prompt imbatível

Imagem gerada com IA



Os seis componentes para a elaboração de um bom prompt em ordem de importância são:

- 1 Tarefa
- 2 Contexto
- 3 Exemplos
- 4 Persona
- 5 Formato
- 6 Tom

Vamos ver cada um deles a partir de agora com alguns exemplos.

1 Tarefa

No prompt, é o que você deseja que seja realizado. Portanto, sempre comece com um verbo de ação. Exemplo: gere, escreva, analise, etc.

Exemplos:

Tarefa única: Gere um programa de treinamento de 3 meses para eu seguir.

Multi-Tarefa: Explique o conceito de sustentabilidade, calcule a média anual de resíduos reciclados de 2019 a 2023 e desenvolva três medidas para uma escola se tornar mais sustentável.

2 Contexto

Para limitar as infinitas possibilidades de cenário, é necessário fornecer um contexto para o modelo (IA). Para isso, procure analisar as seguintes perguntas:

- Qual o histórico do usuário?
- Qual o objetivo que deseja alcançar?
- Em que ambiente ou condição o usuário se encontra?

Perceba a aplicação dessas perguntas no exemplo do prompt abaixo:

Eu sou um **homem de 76kg** com o objetivo de **ganhar 4kg de massa muscular nos próximos três meses**. Eu somente tenho **tempo de ir à academia duas vezes por semana e permanecer por uma hora cada vez**. Faça um programa de treinamento para eu seguir.

Lembre-se! Nem sempre há necessidade de inserir uma informação de contexto no prompt!

3 Exemplos

Ao elaborar o prompt e fornecer exemplos para o modelo (IA), é possível ensinar e estabelecer um padrão do que se deseja como resposta. Assim, o modelo identifica um padrão e elabora a resposta de acordo com o(s) exemplo(s) dado(s). É parecido com aquele exercício no livro didático que tem um exemplo para você seguir.

Exemplo de prompt:

Inglês: Hello! How are you?

Português: Olá, como vai?

Inglês: I'm learning to speak Portuguese.

Português: Eu estou aprendendo a falar Português.

Inglês: The weather is very nice today.

Português:

Resposta do modelo (IA): O tempo está muito bem hoje.

(Ele identificou um padrão de tradução).

4 Persona

Por vezes, vale a pena dar informações ao modelo (IA) indicando como ele deve se comportar no sentido de quem ele deve ser. Assim, as respostas serão produzidas a partir daquele olhar.

Exemplos:

"Aja como um técnico em eletrônica e..."

"Se você estivesse procurando trabalho e..."

"Eu sou um estudante universitário de 19 anos que está estudando inglês e..."

5 Formato

Esse item nada mais é do que o formato em si que pode ser um email, uma lista, blocos de programação, parágrafos, etc.

Exemplo:

Escreva um e-mail para o meu chefe, Carlos Mendes, compartilhando essas notícias positivas.

6 Tom

O tom é a forma com a qual você vai solicitar para que o modelo elabore a sua resposta. Pode não ter diferença quando o resultado é numérico, porém quando é textual, sim.

Exemplos de tom que podem ser utilizados: formal, informal, criativo, amigável, inteligente, confidente, sarcástico, engraçado, confuso, etc.

Essas são algumas dicas interessantes para criar alguns prompts, lembrando sempre que a clareza é sempre o ponto essencial.

Alucinação

Modelos de Inteligência Artificial podem oferecer respostas que se enquadram no que chamamos de “alucinação”. Isso quer dizer que oferecem respostas sem sentido ou com erros. Por isso, sempre esteja atento a essas situações. Para evitar esses problemas, tenha clareza na requisição no momento de elaborar o prompt e forneça o máximo de informações possíveis conforme as dicas dadas anteriormente.

Copilot | Designer

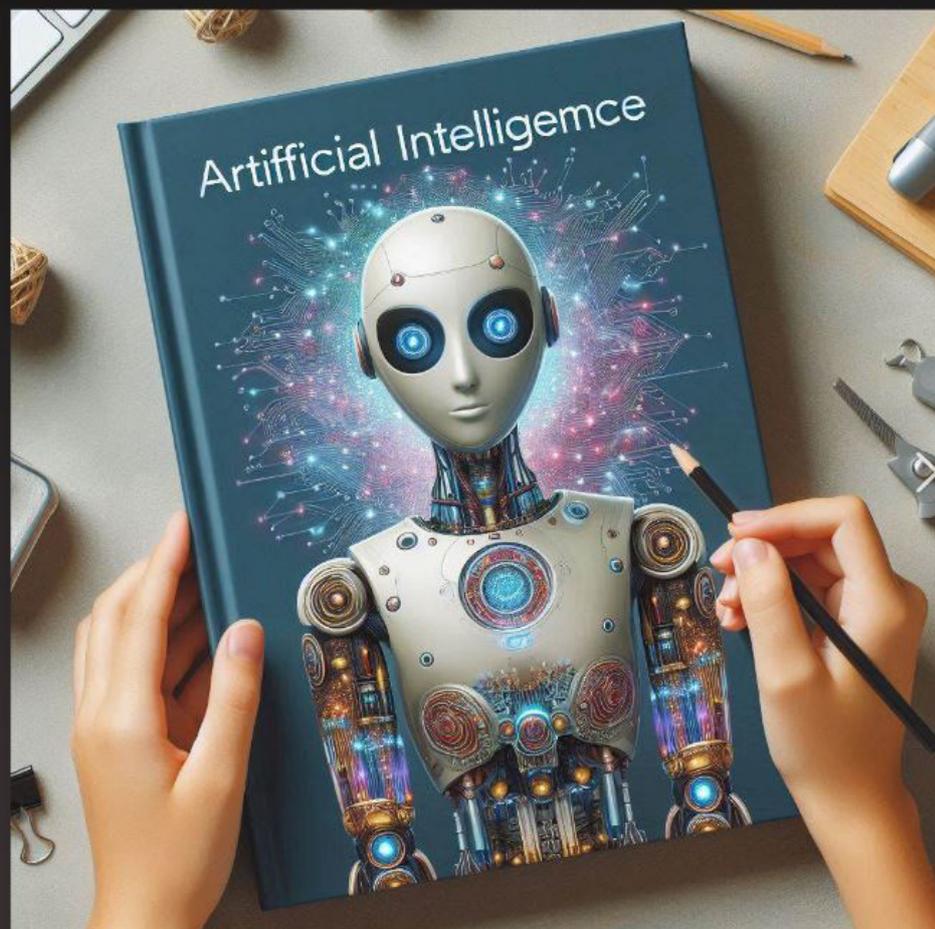


Imagem para capa de uma apostila destinada a alunos de robótica, que resuma o que é inteligência artificial, estilo fotografia

Designer | 1024 x 1024 jpeg | Há 19 min

[Compartilhar](#) [Salvar](#) [Baixar](#) [Comentários](#)

Credenciais de conteúdo
Gerado com IA · 26 de julho de 2024 às 9:11 AM

Agora, vamos programar com IA!

Vamos à nossa primeira aplicação? Utilizar a inteligência artificial para criar códigos.

Ao utilizar a IA para gerar códigos, é fundamental que a gente forneça uma descrição clara e detalhada do problema a ser resolvido ou os requisitos desejados para a programação, utilizando uma linguagem natural e, sempre que possível, incluindo exemplos adicionais ou o contexto de aplicação do projeto.

Figura 1 – Exemplo de prompt para projeto de uma calculadora com teclado e display OLED



Você

Crie um código em Arduino para implementar uma calculadora simples utilizando um teclado matricial de membrana 4x4 e um display OLED 128x64. As conexões do teclado são: colunas 1 a 4 conectadas nas portas 7 a 4 do Arduino, e linhas 1 a 4 conectadas nas portas 11 a 8. O código deve permitir a inserção de números e operadores através do teclado, processar as operações matemáticas básicas e exibir o resultado no display OLED.

Fonte: Copilot

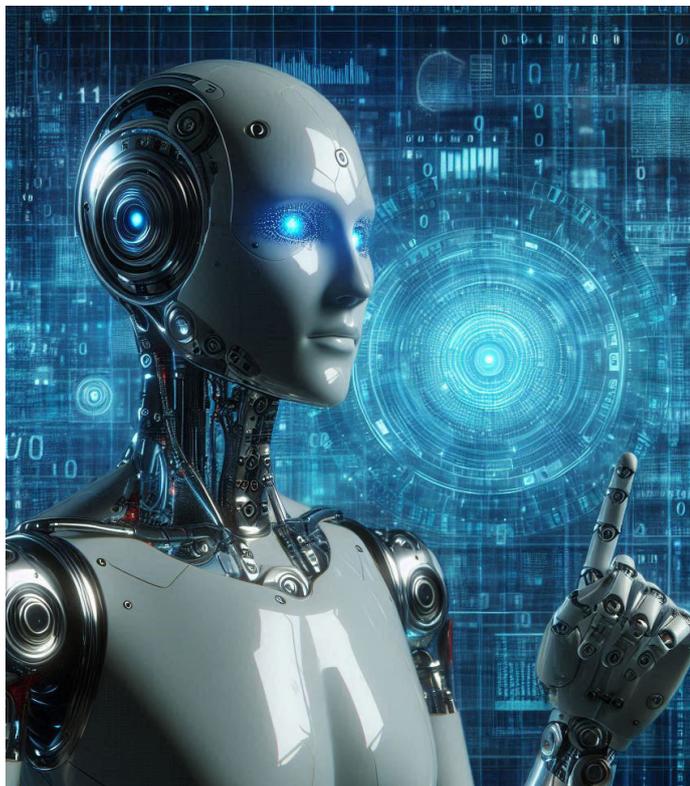
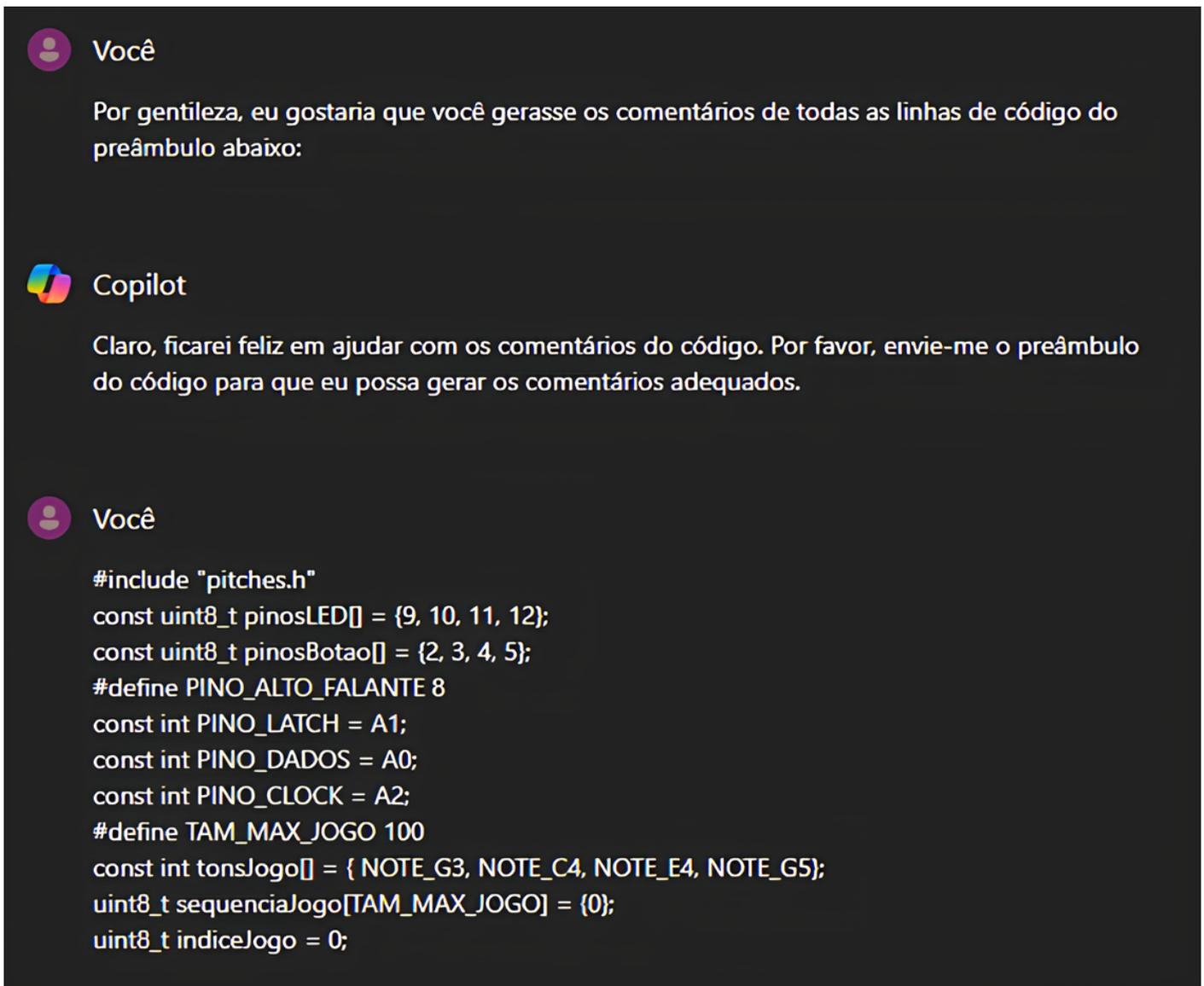


Imagem gerada com IA

Teste o código em um [protótipo montado](#) e observe que, assim como em demais situações de pesquisa ou análise de dados e resultados, é muito importante, ao receber o código gerado pela IA, **revisá-lo cuidadosamente, fazer ajustes conforme o necessário e iterar o processo**, se preciso, com base em feedbacks. O bacana desse processo todo é que a utilização da IA para geração de códigos também nos proporciona **oportunidades de aprendizado** contínuo, permitindo a descoberta de novas técnicas e padrões de codificação e, com o tempo, você **aprimorará suas habilidades**, o que trará mais **eficiência** na obtenção de códigos.

Uma segunda aplicação pode ser a de solicitarmos à IA **comentários em um código**, como ao [jogo Simon, desenvolvido por Uri Shaked](#).

Figura 2 – Exemplo de prompt para geração de comentários a um código de programação



Você

Por gentileza, eu gostaria que você gerasse os comentários de todas as linhas de código do preâmbulo abaixo:

Copilot

Claro, ficarei feliz em ajudar com os comentários do código. Por favor, envie-me o preâmbulo do código para que eu possa gerar os comentários adequados.

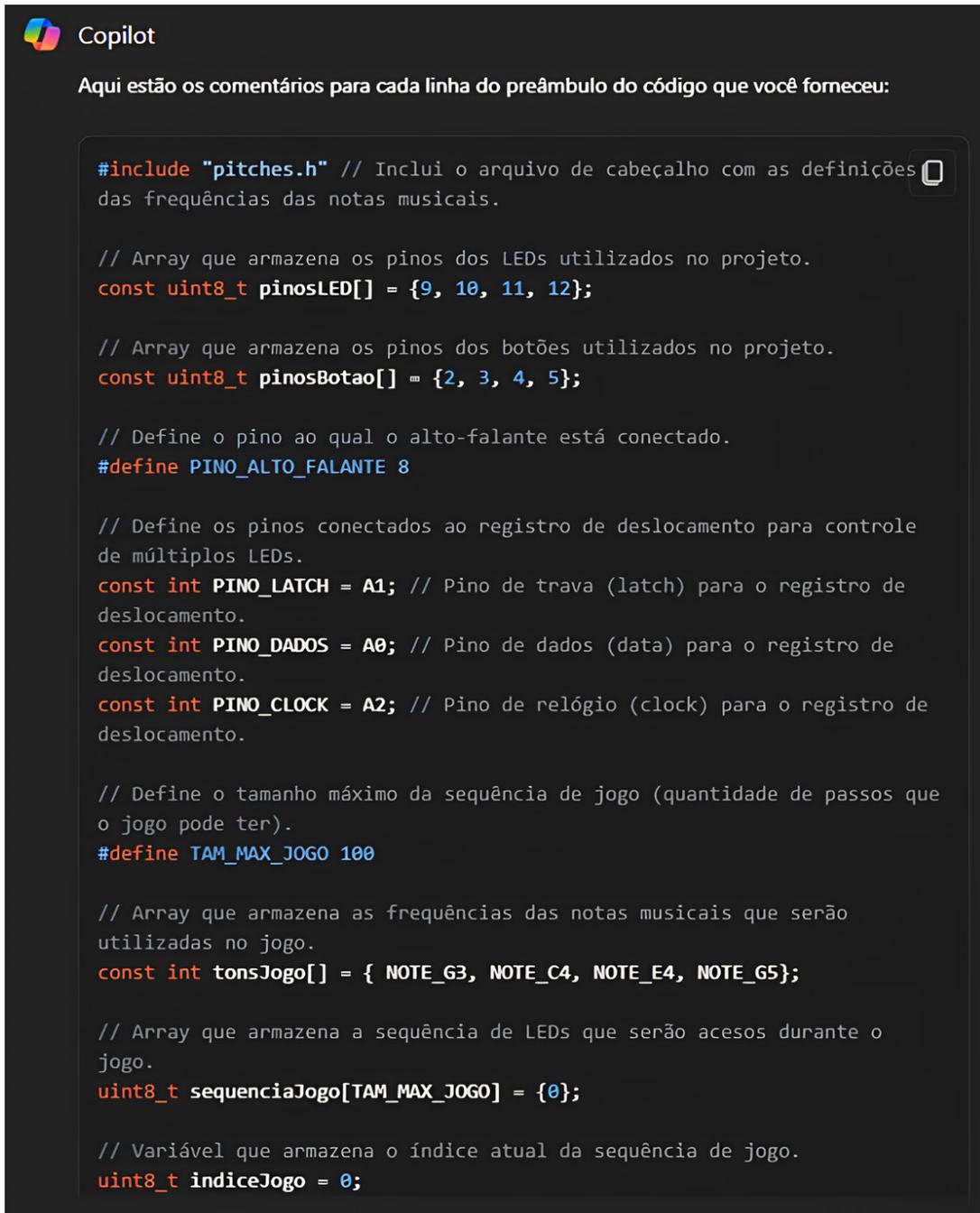
Você

```
#include "pitches.h"
const uint8_t pinosLED[] = {9, 10, 11, 12};
const uint8_t pinosBotao[] = {2, 3, 4, 5};
#define PINO_ALTO_FALANTE 8
const int PINO_LATCH = A1;
const int PINO_DADOS = A0;
const int PINO_CLOCK = A2;
#define TAM_MAX_JOGO 100
const int tonsJogo[] = { NOTE_G3, NOTE_C4, NOTE_E4, NOTE_G5};
uint8_t sequenciaJogo[TAM_MAX_JOGO] = {0};
uint8_t indiceJogo = 0;
```

Fonte: Copilot

Conforme a programação que queremos comentar, é bacana inserir as **partes** do código no prompt, dividindo **preâmbulo**, **void setup()** e **void loop()** ou demais partes do código, por exemplo, para que a IA possa nos retornar os comentários de forma organizada, considerando eventuais limitações de caracteres tanto para o prompt quanto para as respostas geradas.

Figura 3 – Exemplo de retorno dos comentários gerados pela IA



Copilot

Aqui estão os comentários para cada linha do preâmbulo do código que você forneceu:

```
#include "pitches.h" // Inclui o arquivo de cabeçalho com as definições
das frequências das notas musicais.

// Array que armazena os pinos dos LEDs utilizados no projeto.
const uint8_t pinosLED[] = {9, 10, 11, 12};

// Array que armazena os pinos dos botões utilizados no projeto.
const uint8_t pinosBotao[] = {2, 3, 4, 5};

// Define o pino ao qual o alto-falante está conectado.
#define PINO_ALTO_FALANTE 8

// Define os pinos conectados ao registro de deslocamento para controle
de múltiplos LEDs.
const int PINO_LATCH = A1; // Pino de trava (latch) para o registro de
deslocamento.
const int PINO_DADOS = A0; // Pino de dados (data) para o registro de
deslocamento.
const int PINO_CLOCK = A2; // Pino de relógio (clock) para o registro de
deslocamento.

// Define o tamanho máximo da sequência de jogo (quantidade de passos que
o jogo pode ter).
#define TAM_MAX_JOGO 100

// Array que armazena as frequências das notas musicais que serão
utilizadas no jogo.
const int tonsJogo[] = { NOTE_G3, NOTE_C4, NOTE_E4, NOTE_G5};

// Array que armazena a sequência de LEDs que serão acesos durante o
jogo.
uint8_t sequenciaJogo[TAM_MAX_JOGO] = {0};

// Variável que armazena o índice atual da sequência de jogo.
uint8_t indiceJogo = 0;
```

Fonte: Copilot



Conforme o retorno da IA, você pode agradecer ou solicitar mais ajuda para os comentários gerados inicialmente ou mesmo continuar com o restante do código da sua programação.

A utilização da IA para gerar comentários em algum código de programação **simplifica a compreensão e a manutenção do código**, explicando a função de linhas específicas, a lógica por trás dos algoritmos e a finalidade de variáveis e métodos.

Esses comentários nos auxiliam tanto a entender, quanto a modificar o código de modo mais simplificado e intuitivo, aumentando sua **legibilidade** e facilitando a **colaboração** entre os desenvolvedores. Caso algum trecho não tenha ficado claro ainda, solicite que a IA escreva mais detalhes sobre.

Outra aplicação da IA à Robótica refere-se à **correção de erros de compilação**.

Ao compilar um código de programação pelo Arduino IDE, caso existam erros na programação, como variáveis não declaradas, bibliotecas não incluídas, funções não criadas ou demais erros de sintaxes, o Arduino IDE os identificará e você poderá **copiar a mensagem de erro de compilação** para solicitar à IA a **correção**.

Com foco no exemplo desta aula, alteramos a programação do [protótipo de mini piano, desenvolvido por Uri Shaked](#), para, ao iniciarmos a simulação, o simulador indicar os erros de compilação, os quais solicitaremos à IA corrigir.

Figura 4 – Exemplo de mensagem de erro de compilação ao compilar o código pelo Wokwi

Build failed!

```

mini-piano.ino:17:1: error: expected ',', or ';' before 'const'
const int buttonTones[] = {
  ^~~~~~
mini-piano.ino: In function 'void setup()':
mini-piano.ino:25:24: error: 'j' was not declared in this scope
  pinMode(buttonPins[j], INPUT_PULLUP);
                        ^
mini-piano.ino: In function 'void loop()':
mini-piano.ino:33:9: error: 'digitalRea' was not declared in this scope
  if (digitalRea(buttonPins[i]) == LOW) {
      ^~~~~~
mini-piano.ino:33:9: note: suggested alternative: 'digitalRead'
  if (digitalRea(buttonPins[i]) == LOW) {
      ^~~~~~
mini-piano.ino:34:15: error: 'buttonTones' was not declared in this scope
  pitch = buttonTones[i];
             ^~~~~~
mini-piano.ino:34:15: note: suggested alternative: 'buttonPins'
  pitch = buttonTones[i];
             ^~~~~~
mini-piano.ino:40:5: error: 'notone' was not declared in this scope
  notone(SPEAKER_PIN);
  ^~~~~~
mini-piano.ino:40:5: note: suggested alternative: 'noTone'
  notone
  ^~~~~~
Error during build: exit status 1
  
```

GET HELP CLOBB

Fonte: Wokwi

Durante o desenvolvimento dos nossos projetos, podemos cometer equívocos simples, como esquecer um ; ou { } (quem nunca?) e, conforme a extensão da programação, pode ser difícil identificarmos o que precisa ser corrigido tão rapidamente.

Figura 5 – Exemplo de prompt com solicitação de identificação e correção de erros pela IA

```

 Você

Olá! Você pode me ajudar com a correção destes erros? mini-piano.ino:17:1: error: expected ';'
or ';' before 'const'
const int buttonTones[] = (
^~~~~
mini-piano.ino: In function 'void setup()':
mini-piano.ino:25:24: error: 'j' was not declared in this scope
  pinMode(buttonPins[j], INPUT_PULLUP);
                    ^
mini-piano.ino: In function 'void loop()':
mini-piano.ino:33:9: error: 'digitalRea' was not declared in this scope
  if (digitalRea(buttonPins[i]) == LOW) (
    ^~~~~~
mini-piano.ino:33:9: note: suggested alternative: 'digitalRead'
  if (digitalRea(buttonPins[i]) == LOW) (
    ^~~~~~
    digitalRead
mini-piano.ino:34:15: error: 'buttonTones' was not declared in this scope
  pitch = buttonTones[i];
            ^~~~~~
mini-piano.ino:34:15: note: suggested alternative: 'buttonPins'
  pitch = buttonTones[i];
            ^~~~~~
            buttonPins
mini-piano.ino:40:5: error: 'notone' was not declared in this scope
  notone(SPEAKER_PIN);
  ^~~~~~
mini-piano.ino:40:5: note: suggested alternative: 'noTone'
  notone(SPEAKER_PIN);
  ^~~~~~
  noTone

Error during build: exit status 1

```

Fonte: Copilot

Como a ferramenta de IA tem uma alta capacidade de processamento, seu uso tanto facilita para nós encontrarmos o erro, quanto fomenta uma solução para o erro encontrado. E, assim como nas outras finalidades da IA, ao obter os resultados, sempre verifique tudo e **confira a solução apresentada, revisando e compilando novamente o código.**

Figura 6 – Exemplo de solução apresentada pela IA



Copilot

Claro, posso ajudar com a correção dos erros no seu código Arduino. Aqui estão as correções necessárias:

1. No erro `expected ',' or ';' before 'const'`, parece que falta um ponto e vírgula no final da declaração anterior à linha 17. Certifique-se de que cada instrução termine com um ponto e vírgula.
2. O erro `'j' was not declared in this scope` sugere que a variável `j` não foi declarada. Você precisa declarar `j` dentro do escopo da função `setup()` ou usá-la como índice em um loop `for`.
3. O erro `'digitalRea' was not declared in this scope` é um erro de digitação. Deve ser `digitalRead`.
4. O erro `'buttonTones' was not declared in this scope` indica que a matriz `buttonTones` não foi declarada corretamente. Certifique-se de que ela esteja declarada globalmente ou dentro do escopo onde está sendo usada.
5. O erro `'notone' was not declared in this scope` é outro erro de digitação. A função correta é `noTone`.

Fonte: Copilot

Figura 7 – Exemplo de código corrigido pela IA

Aqui está o código corrigido com base nos erros fornecidos:

```
// Certifique-se de que todas as declarações anteriores a esta linha
terminem com ponto e vírgula
const int buttonTones[] = { /* valores dos tons */ };

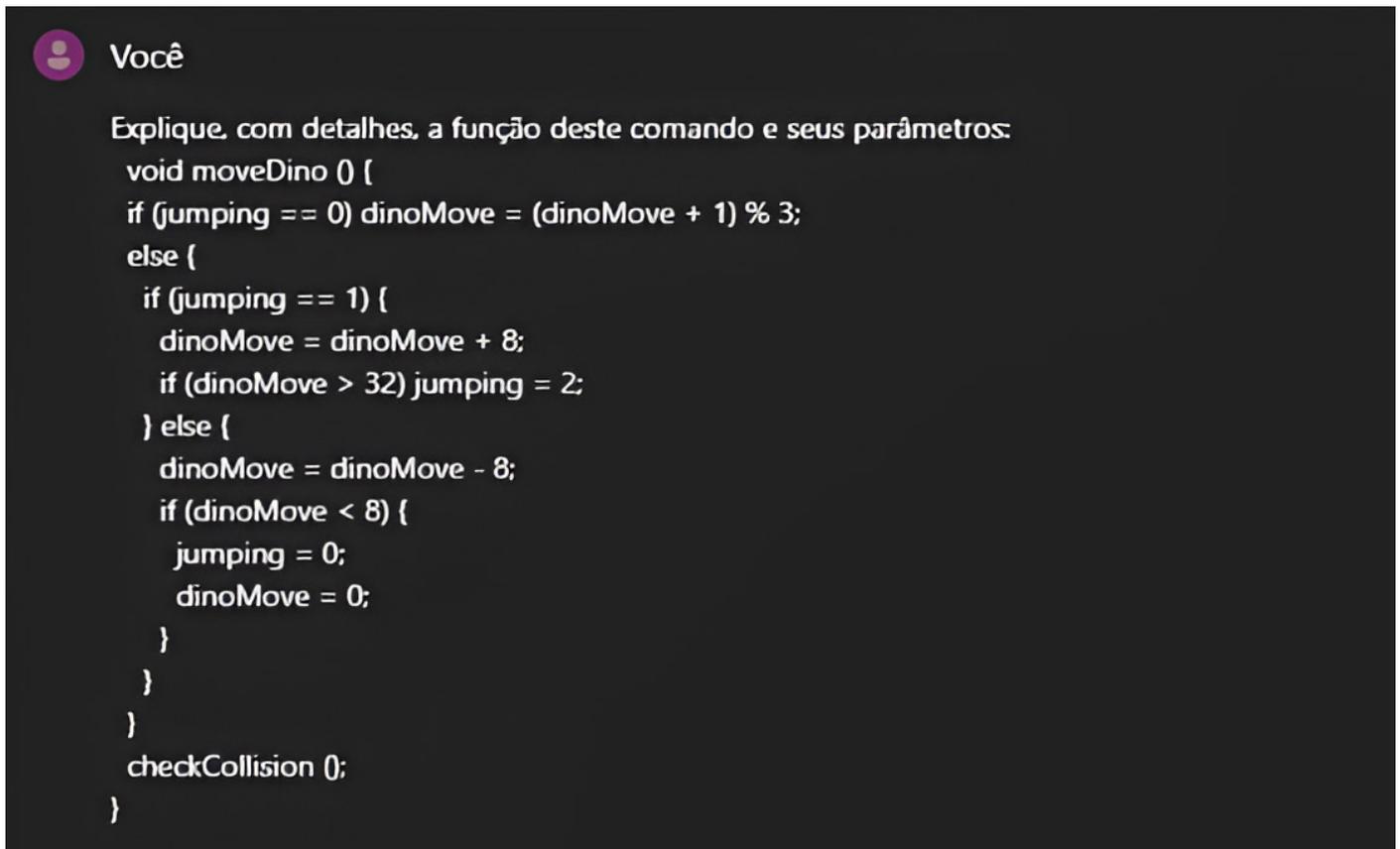
void setup() {
  // Substitua 'j' por 'i' ou declare 'j' antes de usá-lo
  for (int i = 0; i < /* tamanho do array buttonPins */; i++) {
    pinMode(buttonPins[i], INPUT_PULLUP);
  }
}

void loop() {
  for (int i = 0; i < /* tamanho do array buttonPins */; i++) {
    // Corrija para 'digitalRead'
    if (digitalRead(buttonPins[i]) == LOW) {
      // Certifique-se de que 'buttonTones' esteja declarado corretamente
      int pitch = buttonTones[i];
      // Código para tocar o tom aqui
    } else {
      // Corrija para 'noTone'
      noTone(SPEAKER_PIN);
    }
  }
}
```

Fonte: Copilot

Podemos utilizar a IA também para a **explicação de determinados comandos, funções ou linhas de código**, como no caso do projeto do [jogo Dino](#), desenvolvido por Ishani and Arijit Sengupta. Ao analisar as linhas de programação desse código, você pode se deparar com trechos mais complexos e solicitar auxílio à IA.

Figura 8 – Exemplo de prompt para explicação de trechos de programação



The image shows a chat interface with a dark background. On the left, there is a purple circular icon with a white person silhouette, followed by the text 'Você'. To the right of this is a white text prompt: 'Explique, com detalhes, a função deste comando e seus parâmetros:'. Below the prompt is a code block in white text on a black background. The code is a C++ function named 'void moveDino ()' which contains several conditional statements and arithmetic operations on variables 'dinoMove' and 'jumping'. The code ends with a call to 'checkCollision ()'.

```
void moveDino () {
  if (jumping == 0) dinoMove = (dinoMove + 1) % 3;
  else {
    if (jumping == 1) {
      dinoMove = dinoMove + 8;
      if (dinoMove > 32) jumping = 2;
    } else {
      dinoMove = dinoMove - 8;
      if (dinoMove < 8) {
        jumping = 0;
        dinoMove = 0;
      }
    }
  }
  checkCollision ();
}
```

Fonte: Copilot

A IA pode ter o comportamento de explicar direto o código ou trazer comentários às linhas de programação com o complemento da explicação dos componentes.

Figura 9 – Comentários aos trechos da programação



Copilot

A função `moveDino()` parece controlar o movimento de um dinossauro em um jogo, possivelmente para um jogo de corrida ou obstáculos. Vamos analisar o código passo a passo:

```
void moveDino() {
    // Se o dinossauro não estiver pulando, atualiza a animação do
    movimento.
    if (jumping == 0) dinoMove = (dinoMove + 1) % 3;
    else {
        // Se o dinossauro estiver subindo no pulo, aumenta a posição
        vertical.
        if (jumping == 1) {
            dinoMove = dinoMove + 8;
            // Se o dinossauro atingir a altura máxima do pulo, começa a
            descer.
            if (dinoMove > 32) jumping = 2;
        } else {
            // Se o dinossauro estiver descendo no pulo, diminui a posição
            vertical.
            dinoMove = dinoMove - 8;
            // Se o dinossauro voltar ao chão, reseta o estado de pulo e a
            posição vertical.
            if (dinoMove < 8) {
                jumping = 0;
                dinoMove = 0;
            }
        }
    }
    // Verifica se houve colisão com algum obstáculo.
    checkCollision();
}
```

Fonte: Copilot

Conforme sua utilização, a IA pode trazer mais elementos à explicação.

Figura 10 – Explicação dos componentes presentes na função

Explicação dos componentes da função:

- `void moveDino()` : Declaração da função `moveDino` que não retorna nenhum valor (`void`).
- `if (jumping == 0)` : Verifica se a variável `jumping` é igual a 0, o que indica que o dinossauro não está pulando.
- `dinoMove = (dinoMove + 1) % 3;` : Atualiza a variável `dinoMove` para mudar a animação do dinossauro. O operador `%` garante que o valor de `dinoMove` fique entre 0 e 2.
- `if (jumping == 1)` : Verifica se o dinossauro está na fase de subida do pulo.
- `dinoMove = dinoMove + 8;` : Aumenta a posição vertical do dinossauro durante o pulo.
- `if (dinoMove > 32)` : Verifica se o dinossauro atingiu a altura máxima do pulo.
- `jumping = 2;` : Muda o estado de `jumping` para 2, indicando que o dinossauro está descendo.
- `dinoMove = dinoMove - 8;` : Diminui a posição vertical do dinossauro durante a descida do pulo.
- `if (dinoMove < 8)` : Verifica se o dinossauro voltou ao nível do chão.
- `jumping = 0;` : Reseta o estado de `jumping` para 0, indicando que o dinossauro não está mais pulando.
- `dinoMove = 0;` : Reseta a posição vertical do dinossauro.
- `checkCollision();` : Chama a função `checkCollision` para verificar se o dinossauro colidiu com algum obstáculo.

Essa função é responsável por atualizar a posição vertical do dinossauro durante o pulo e verificar colisões. Os parâmetros `jumping` e `dinoMove` são provavelmente variáveis globais ou de classe que mantêm o estado do dinossauro no jogo. A variável `jumping` parece ter três estados: 0 para não pulando, 1 para subindo no pulo, e 2 para descendo do pulo. A variável `dinoMove` representa a posição vertical do dinossauro e é usada para calcular a animação do movimento e a altura do pulo. A função `checkCollision()` é chamada no final para verificar se o dinossauro colidiu com algum obstáculo após o movimento.

Fonte: Copilot

Assim como nas demais utilizações da IA, sempre cheque os comentários e explicações e, caso algo não pareça tão claro ou correto, solicite uma nova explicação à IA.

Na **Aula 07 - Recursos das bibliotecas**, aprofundamos nosso conhecimento referente aos recursos oferecidos pelas bibliotecas e, na **Aula 08 - Técnicas de programação: sketch com abas**, aprimoramos a organização dos nossos códigos e a criação de funções. A IA pode nos auxiliar também para a **criação de novas bibliotecas**, facilitando assim o projeto em que estamos trabalhando por melhorar a escrita e a própria manutenção do projeto.

Por exemplo, em um novo projeto, ao invés de utilizarmos as bibliotecas já existentes, como ocorre em aulas anteriores que envolveram prototipagem com chassis, motores e sensores, podemos utilizar vários comandos já usados para a elaboração de uma biblioteca destinada ao controle da ponte H e do sensor ultrassônico.

Nesse processo, é importante passar à IA a descrição dos componentes e pinos utilizados em sua montagem, podendo especificar também os parâmetros iniciais que deseja para sua biblioteca.

Figura 11 – Exemplo de prompt para início de criação de biblioteca



Você

Olá! Eu gostaria de criar uma biblioteca para controle de robô ultrassônico, utilizando um sensor ultrassônico SR-04; drive de motores L298N; controle de direção e velocidade, através dos pinos IN1, IN2, IN3 e IN4. Crie também as seguintes funções:

Frente(velocidade),

Re(velocidade),

vireDireita(velocidade),

vireEsquerda(velocidade);

lerSensor();

Como ficaria o arquivo .h desta biblioteca chamada "RoboUltrasonico"?

Fonte: Copilot

Como o processo da geração de bibliotecas pode ser mais extenso e detalhado, opte em solicitar por partes a geração dos arquivos básicos da biblioteca, com foco em suas funcionalidades.

Figura 12 – Exemplo de do início da conversa com IA para criação de biblioteca



Copilot

Claro, posso ajudá-lo a criar um arquivo de cabeçalho `.h` para uma biblioteca chamada "RoboUltrasonico" que controlará um robô ultrassônico. Aqui está um exemplo de como o arquivo `RoboUltrasonico.h` pode ser estruturado:

```
#ifndef RoboUltrasonico_h
#define RoboUltrasonico_h

#include "Arduino.h"

class RoboUltrasonico {
public:
    RoboUltrasonico(int pinIN1, int pinIN2, int pinIN3, int pinIN4, int
pinTrig, int pinEcho);
    void frente(int velocidade);
    void re(int velocidade);
    void vireDireita(int velocidade);
    void vireEsquerda(int velocidade);
    long lerSensor();

private:
    int _pinIN1, _pinIN2, _pinIN3, _pinIN4, _pinTrig, _pinEcho;
    void acionarMotor(int motor1, int motor2, int velocidade);
};

#endif
```

Neste arquivo de cabeçalho, definimos a classe `RoboUltrasonico` e declaramos os métodos que você especificou. Note que também inclui um método privado `acionarMotor` que é utilizado internamente para controlar os motores. Você precisará implementar a lógica dessas funções no arquivo `.cpp` correspondente.

Lembre-se de ajustar os pinos no construtor para os pinos que você está usando em seu projeto e implementar as funções no arquivo `.cpp` para controlar o hardware adequadamente. Se precisar de ajuda com a implementação ou tiver outras perguntas, fique à vontade para perguntar!

Fonte: Copilot

Além do arquivo **.h** e como implementar as funções no arquivo **.cpp**, os quais conhecemos um pouco mais na **Aula 07 - Recursos das bibliotecas** e na **Aula 08 - Técnicas de programação: sketch com abas**, você também pode solicitar depois, conforme vai refinando e testando a biblioteca, os arquivos de exemplos de programação. No decorrer do processo, lembre-se de abrir uma pasta para a biblioteca em seu computador ou drive para salvar cada um dos arquivos gerados.

Sugestão de etapas para criação de biblioteca com o auxílio de IA, revisando todas as etapas com foco nas melhores funções, parâmetros, indicação de pinos e demais ajustes:

- Criação do arquivo **.h**;
- Criação do arquivo **.cpp**;
- Gerar arquivos de **exemplos** para a biblioteca;
- Como opcional, gerar também os arquivos **keywords.txt** e **library.properties**;
- Criar a pasta do projeto, contendo seus arquivos principais e subpasta **example**.

Depois de tudo certo, como utilizar a pasta da biblioteca? Compactar tudo como arquivo **.zip** e instalar pelo Arduino IDE.

Para mais detalhes sobre criação de bibliotecas para Arduino, retorne ao apêndice “*Quer se aprofundar mais?*” presente na **Aula 07 - Recursos das bibliotecas** e amplie seu diálogo com colegas e IA.

Como vimos nesta sequência de aulas e dicas sobre a utilização de IA em projetos aplicados à Robótica, muitas vezes a tomada de decisão em relação aos nossos projetos pode não ser tão simples e a IA nos auxilia de modo mais eficiente. Na programação de nossos protótipos, você e seus colegas podem ter percebido que as tarefas não são tão simples e muitas vezes acabamos “esquentando a cabeça” sobre como melhorar códigos, tomar decisões, aplicar recursos, etc. Pensando nesses momentos, a IA pode ser uma aliada e apoiar o desenvolvimento de projetos.

Desafios:

Que tal gerar ao menos um prompt para cada possibilidade de utilização da IA em códigos de programação?

Converse com seus colegas... vocês conseguem identificar outras utilizações da IA para auxiliar no desenvolvimento ou aprimoramento dos projetos de Robótica em relação à montagem e programação?

E se...

O projeto não funcionar?

- Aprimore os prompts, experimentando outras formas e trazendo mais detalhes acerca do motivo pelo qual você está utilizando o recurso da IA em seus projetos de Robótica. Na geração de um código de programação, por exemplo, o prompt pode ser refinado e melhorado para fazer mais alguns processos e especificar as portas e conexões utilizadas no projeto facilita porque a IA pode gerar quaisquer outras, o que implicaria em fazermos as correções posteriores na programação ou no próprio protótipo já montado.
- Revise os códigos apresentados, consultando até outros exemplos ou mesmo a Documentação de Referência do Arduino como uma forma de ampliar seus estudos.
- Conforme o software de IA que você estiver utilizando, pode ser que ele apresente o resultado da conversa em mais de uma parte, observando se há alguma menção como "continue gerando" em sua resposta.

3. Feedback e finalização

Nesta aula, trouxemos exemplos iniciais de prompts com foco na Robótica. A linguagem utilizada com a IA precisa ser clara e detalhada do problema, o que amplia nossa habilidade de explanação e comunicação. Além disso, ao recebermos qualquer resultado, é preciso revisar, testar e experimentar várias vezes até alcançarmos o resultado desejado.

Como você pode observar, o processo todo é circular por conversarmos com a IA sobre o que queremos, analisarmos os resultados devolvidos e ir ajustando, revisando e refinando a ação toda em um processo amplo de ensino e aprendizagem.

REFERÊNCIAS

- 4 METHODS OF PROMPT ENGINEERING. Disponível em: <https://www.youtube.com/watch?v=1c9iyo-VlwDs> Acesso em 14/06/2024.
- AI TIPPING POINT. In: Curiosity Channel. Disponível em: <https://youtu.be/1cKE12LK4Eo?si=5DuAJzvhW-8F7FsOQ> Acesso em 14/06/2024.
- ALUCINAÇÃO NA INTENLIGÊNCIA ARTIFICIAL. In: Jornal O Globo. Disponível em: <https://www.youtube.com/watch?v=ITFRfB2Pov4> Acesso em 20/06/2024.
- A PRACTICAL SURVEY ON ZERO-SHOT PROMPT DESIGN FOR IN-CONTEXT LEARNING. In: Cornell University. Disponível em: https://doi.org/10.26615/978-954-452-092-2_069 Acesso em 17/06/2024.
- ARDUINO. Documentação de Referência da Linguagem Arduino. Disponível em: <https://www.arduino.cc/reference/pt/>. Acesso em: 27 mai. 2024.
- O QUE É MACHINE LEARNING? Disponível em: <https://www.youtube.com/watch?v=Z1YHbl0lh88> Acesso em 12/06/2024.
- CHATGPT E ENGENHARIA DE PROMPT: TÉCNICAS PARA O PROMPT PERFEITO. Disponível em: <https://www.youtube.com/watch?v=NsXfldreSPQ> Acesso em 12/06/2024.
- CHATGPT: O QUE É, COMO USAR E DICAS DE COMANDOS PARA O DIA A DIA. In: Alura. Disponível em: <https://www.alura.com.br/artigos/chatgpt> Acesso em 12/06/2024.
- GUIA COMPLETO DE PROMPTS PARA IA: POTENCIALIZE SUA CRIAÇÃO DE CONTEÚDO. In: Rockcontent. Disponível em: <https://rockcontent.com/br/blog/guia-de-prompts-de-inteligencia-artificial-/p/ara-/p/roducao-de-conteudo/> Acesso em 17/06/2024.
- LANGUAGE MODELS ARE FEW-SHOT LEARNERS. In: Cornell University. Disponível em: <https://doi.org/10.48550/arXiv.2005.14165> Acesso em 17/06/2024.
- LANGUAGE MODELS PERFORM REASONING VIA CHAIN OF THOUGHT. In: Google Research. Disponível em: <https://research.google/blog/language-models-perform-reasoning-via-chain-of-thought/> Acesso em 11/06/2024.
- LARGE LANGUAGE MODELS ARE ZERO-SHOT REASONERS. In Cornell University. Disponível em: <https://arxiv.org/abs/2205.11916> Acesso em 11/06/2024.
- LEARN PROMPTING. Disponível em: <https://learnprompting.org/> Acesso em 13/06/2024.
- TECHNIQUES TO IMPROVE RELIABILITY. In: OpenAI Cookbook. Disponível em: https://cookbook.openai.com/articles/techniques_to_improve_reliability Acesso em 11/06/2024.

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

- Adilson Carlos Batista
- Ailton Lopes
- Andrea da Silva Castagini Padilha
- Cleiton Rosa
- Darice Alessandra Deckmann Zanardini
- Edna do Rocio Becker
- Kellen Pricila dos Santos Cochinski
- Marcelo Gasparin
- Michele Serpe Fernandes
- Michelle dos Santos
- Roberto Carlos Rodrigues
- Sandra Aguera Alcova Silva

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação do Paraná (SEED), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica. Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

