Monica Craveiro

Desenvolvedora Back-end

PHPeste #REC2024

PALESTRA

IA na Vida de Dev: Seu Novo Parça de Código!

Inteligência Artificial

IA Generativa

Desenvolvimento

https://speakerdeck.com/mocraveirodev

PHPeste
#PEC2024

Carioca da gema, ex-patinadora artística, gamer, #TechMaromba, Dev Emocionada, costumava fazer cálculos por aí mas hoje em dia fico “só nos compiuter”, amante de energético e aspirante a Influencer Tech.

Monica Craveiro de Menezes

@mocraveirode

AGENDA Palestra

# 1	Introdução ao Uso de IA no Desenvolvimento de Software
#2	Aplicação da IA no Dia a Dia do Desenvolvimento
#3	Ferramentas de IA e Suas Funcionalidades
#4	Como melhorar seu Prompt para Obter Respostas Mais Eficientes
#5	Riscos, Desafios e Cuidados no Uso da IA

PHPeste
#REC2024

Parte 1

Introdução ao Uso de IA

no Desenvolvimento de Software

a)	O que é Inteligência Artificial?
b)	Por que aprender IA?

O que é Inteligência Artificial?

Inteligência Artificial, ou **IA**, é um campo da ciência da computação que busca criar **sistemas capazes de realizar tarefas que normalmente requerem inteligência humana**. Isso inclui aprender com dados (aprendizado de máquina, algoritmos), entender e responder a linguagem natural (processamento de linguagem natural), e tomar decisões (visão computacional).

Ou seja, **IA funciona com base em algoritmos e modelos treinados** com grandes quantidades de dados.

Imagine uma assistente virtual que pode entender comandos de voz, texto escrito da forma como falamos ou uma aplicação que recomenda produtos com base no seu histórico de compras, seria incrível, né? **Tudo isso é possível graças à IA**.

Por que aprender IA?

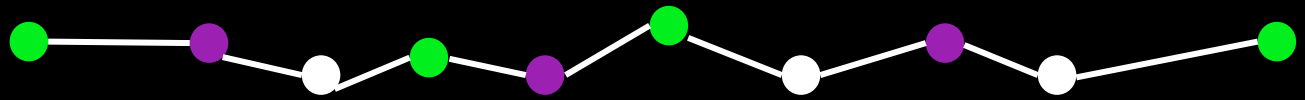
A IA de uma maneira geral, nos **ajuda** a realizar uma série de tarefas e nos ajuda a **economizar** inúmeras horas.

Se conseguirmos escrever um bom *prompt*, ou seja, descrever com detalhes, de forma **precisa e lógica**, é possível que a IA **consiga realizar a tarefa por você** ou até te ajudar a ter um ponto de partida robusto para **tarefas mais complexas**.

Parte 2

Aplicação da IA no
Dia a Dia do Desenvolvimento

a)	Melhorando a Produtividade da Pessoa Desenvolvedora de Software

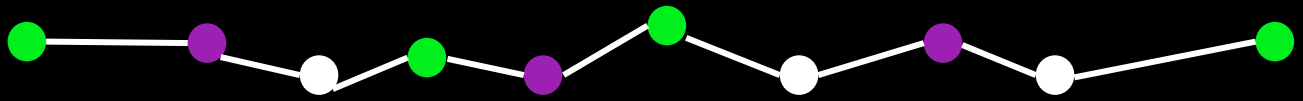


Melhorando a Produtividade da Pessoa Desenvolvedora de Software

A aplicação da IA no desenvolvimento de software tem um potencial incrível para transformar a forma como trabalhamos.

A Inteligência Artificial (IA) está transformando a forma como desenvolvemos software, agilizando processos e trabalhando em conjunto, as pessoas desenvolvedoras podem experimentar um aumento significativo na eficiência e na qualidade do software produzido.

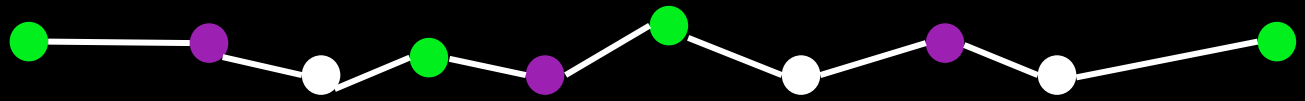
Ao adotar ferramentas baseadas em IA, as pessoas desenvolvedoras podem otimizar suas rotinas diárias, resolver problemas mais rapidamente e dedicar mais tempo à inovação.



Como podemos fazer isso?

IA como Assistente Pessoal:

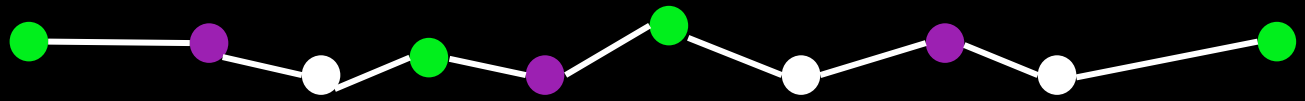
- **Geração de código**: Ferramentas como o Copilot podem sugerir linhas de código completas ou funções inteiras, acelerando significativamente o desenvolvimento.
- **Autocompletar código**: A IA prediz o próximo código que você irá digitar, reduzindo erros e economizando tempo.
- **Refatoração automática**: A IA pode sugerir melhorias na estrutura do código, tornando-o mais legível e manutenível.
- **Documentação automática**: A IA pode gerar comentários explicativos para o código, facilitando a compreensão por outros desenvolvedores.



Como podemos fazer isso?

Análise e Melhoria do Código:

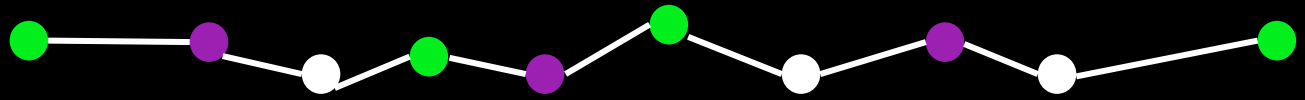
- **Detecção de bugs**: A IA pode identificar padrões de código que podem indicar a presença de bugs, ajudando a prevenir erros antes que eles causem problemas.
- **Sugestões de correção**: A IA pode oferecer soluções para os bugs detectados, economizando tempo na depuração.
- **Análise de código estático**: A IA pode verificar o código em busca de possíveis problemas de segurança, performance e outras métricas de qualidade.



Como podemos fazer isso?

Testes e Validação:

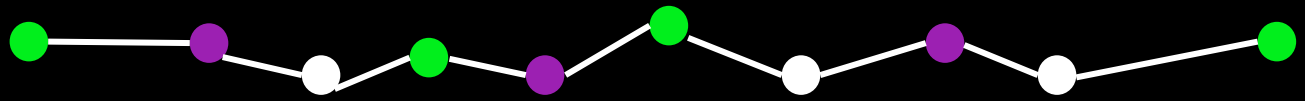
- **Geração de testes unitários**: A IA pode criar testes automatizados para garantir que o código funcione conforme o esperado.
- **Teste de integração**: A IA pode auxiliar na criação de testes que verifiquem a interação entre diferentes componentes do sistema.
- **Validação de resultados**: A IA pode comparar os resultados esperados com os resultados reais, ajudando a identificar discrepâncias e erros.



Como podemos fazer isso?

Aprendizagem e Resolução de Problemas:

- **Explicação de conceitos**: A IA pode explicar conceitos complexos de programação de forma clara e concisa.
- **Resolução de dúvidas**: A IA pode responder a perguntas sobre frameworks, bibliotecas e linguagens de programação, agilizando a busca por soluções.
- **Aprendizado contínuo**: A IA pode aprender com os seus hábitos de codificação e oferecer sugestões personalizadas.
- **Tradução de código**: A IA pode converter código de uma linguagem para outra, agilizando a adaptação de projetos para diferentes plataformas.



Como podemos fazer isso?

Automatização de Tarefas Repetitivas:

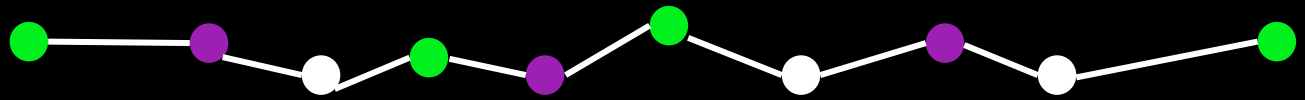
- **Build e deployment**: A IA pode automatizar o processo de construção e implantação de software.
- **Geração de relatórios**: A IA pode gerar relatórios sobre a qualidade do código, cobertura de testes e outras métricas.

Parte 3

Ferramentas de IA

e Suas Funcionalidades

a)	Ferramentas mais utilizadas

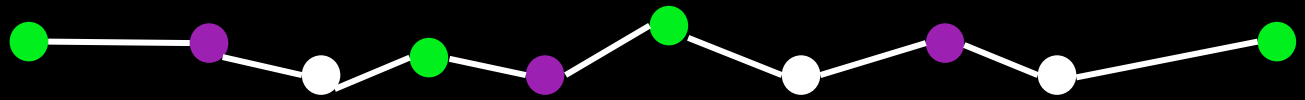


Ferramentas mais utilizadas

Existem várias ferramentas de IA que são amplamente utilizadas por pessoas desenvolvedoras para melhorar sua produtividade no dia a dia.

Cada uma delas possui funcionalidades específicas que podem ser integradas ao fluxo de trabalho de maneira a automatizar, melhorar, agilizar e auxiliar nessa jornada de desenvolvimento de software.

Vamos explorar algumas das mais populares e suas principais funcionalidades:

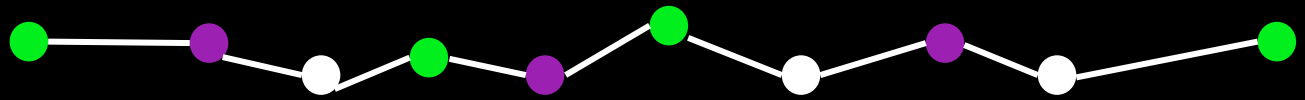


1. ChatGPT (OpenAI) - https://chatgpt.com/

Descrição: O ChatGPT é um modelo de linguagem desenvolvido pela OpenAI que pode realizar tarefas como responder perguntas, gerar explicações, escrever código e até mesmo sugerir melhorias em um projeto de software.
Casos de uso:

- **Geração de código**: Você pode pedir para o ChatGPT gerar trechos de código em várias linguagens.
- **Explicação de conceitos**: Pode ser usado para explicar conceitos complexos de programação ou técnicas de desenvolvimento.
- **Solução de bugs**: Ao descrever um erro, o ChatGPT pode sugerir correções ou explicar o motivo do problema.

Integração ao fluxo de trabalho: Pode ser integrado ao fluxo de trabalho como uma ferramenta de apoio para tirar dúvidas rápidas, gerar código de exemplo e até para refatoração de código. Ferramentas como o ChatGPT podem ser usadas como parte do processo de revisão de código e resolução de problemas.



2. GitHub Copilot (Microsoft + OpenAI) - https://docs.github.com/pt/copilot

Descrição: O GitHub Copilot é um assistente de programação baseado em IA que sugere código e até mesmo funções inteiras enquanto você digita, com base em contextos anteriores. Ele utiliza um modelo de IA treinado em milhões de repositórios de código.
Casos de uso:

- **Autocompletar código**: Sugestões de linhas de código baseadas no que você já escreveu.
- **Criação de funções e módulos**: Pode sugerir funções inteiras ou trechos de código, poupando tempo.
- **Revisão de código**: Ajudar a melhorar e refatorar o código automaticamente.

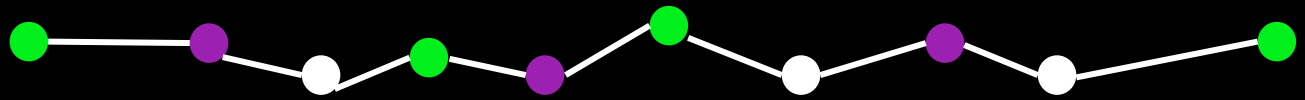
Integração ao fluxo de trabalho: Funciona como uma extensão para editores de código como VSCode, tornando o processo de codificação mais rápido e eficiente. Ele sugere melhorias em tempo real, funcionando como um par-programming automatizado.

3. Microsoft Copilot (Microsoft 365) - https://copilot.microsoft.com/

Descrição: Diferente do GitHub Copilot, o **Microsoft Copilot** é integrado aos aplicativos do Microsoft 365 (Word, Excel, PowerPoint, Outlook, etc.) e usa IA para melhorar a produtividade, gerando conteúdo, resumos e automatizando tarefas.
Casos de uso:

- **Geração de texto**: Ajudar a gerar e resumir documentos, relatórios ou e-mails.
- **Análise de dados**: Automatizar tarefas de análise de dados em planilhas e criar gráficos inteligentes.
- **Automatização de processos**: Usar IA para sugerir melhorias e automatizar tarefas repetitivas.

Integração ao fluxo de trabalho: Ideal para quem trabalha com análise de dados, documentos ou relatórios, automatizando grande parte do trabalho manual.

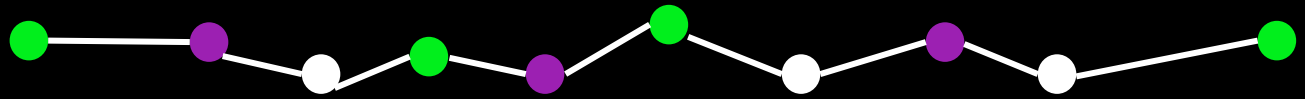


4. Perplexity AI - https://www.perplexity.ai/

Descrição: Perplexity é uma ferramenta de pesquisa assistida por IA que encontra respostas para perguntas complexas e pesquisa de maneira mais interativa.
Casos de uso:

- **Pesquisa de informações técnicas**: Pode ser útil para encontrar rapidamente informações sobre tecnologias, frameworks e práticas recomendadas.
- **Respostas rápidas**: Ao trabalhar em projetos, você pode usá-lo para encontrar rapidamente respostas para questões relacionadas ao desenvolvimento.

Integração ao fluxo de trabalho: Usado principalmente para pesquisa, pode ser integrado ao processo de busca de documentação e melhores práticas de desenvolvimento.

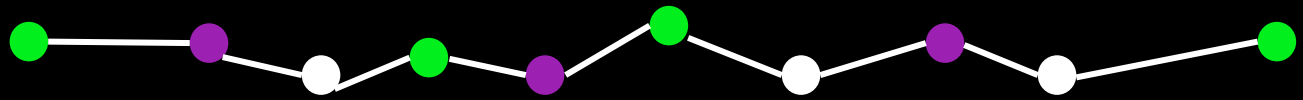


5. Qodo (Codium) - https://www.qodo.ai/

Descrição: Codium é uma plataforma de IA que ajuda a escrever código e otimizar o fluxo de desenvolvimento com sugestões inteligentes.
Casos de uso:

- **Gerar código baseado em requisitos**: Pode gerar código baseado em descrições em linguagem natural.
- **Refatoração e otimização**: Sugere melhorias e refatoração no código já existente.

Integração ao fluxo de trabalho: Codium pode ser integrado como uma extensão ao IDE, ajudando no fluxo de trabalho de escrita e refatoração de código.

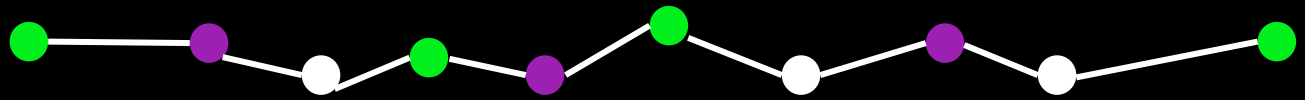


6. Tabnine - https://www.tabnine.com/

Descrição: Tabnine é uma ferramenta de completamento de código alimentada por IA, baseada no modelo GPT-3, que oferece sugestões inteligentes enquanto você codifica.
Casos de uso:

- **Autocompletar código**: Oferece sugestões para completar linhas de código.
- **Aprimorar a produtividade**: Ao sugerir código de maneira eficiente, melhora a produtividade da pessoa desenvolvedora.

Integração ao fluxo de trabalho: Tabnine pode ser integrado ao VSCode, JetBrains e outros editores de código, tornando o desenvolvimento mais rápido.

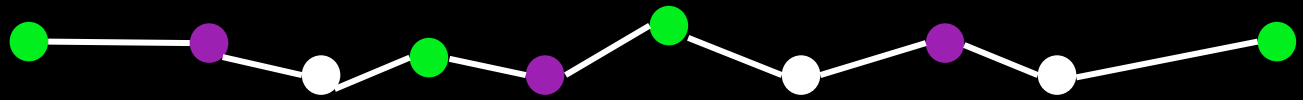


7. CodeWhisperer (Amazon) - https://docs.aws.amazon.com/codewhisperer/latest/userguide/what-is-cwspr.html

Descrição: CodeWhisperer é uma ferramenta da AWS (Amazon) que usa IA para sugerir código, detectar erros e otimizar funções enquanto você programa.
Casos de uso:

- **Sugestões de código em tempo real**: Semelhante ao GitHub Copilot, ajuda a completar código de forma inteligente.
- **Detecção de erros e sugestões de correção**: Pode detectar padrões errôneos e sugerir soluções.

Integração ao fluxo de trabalho: Usado no IDE, pode ajudar em várias linguagens de programação e fluxos de trabalho na AWS.

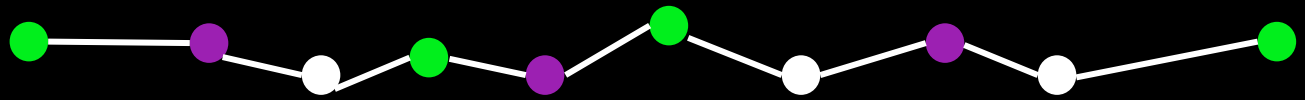


8. Llama (Meta) - https://www.llama.com/

Descrição: Llama é uma família de modelos de IA desenvolvida pela Meta para tarefas de processamento de linguagem natural.
Casos de uso:

- **Respostas baseadas em texto**: Pode ser usada para gerar explicações, resumos e até traduções de código.
- **Automatizar tarefas de NLP**: Pode ser usada para classificar dados, gerar relatórios ou analisar feedback de usuários.

Integração ao fluxo de trabalho: Pode ser útil para tarefas de análise de dados, assistentes de suporte ao cliente ou em integração com sistemas que necessitam de NLP.

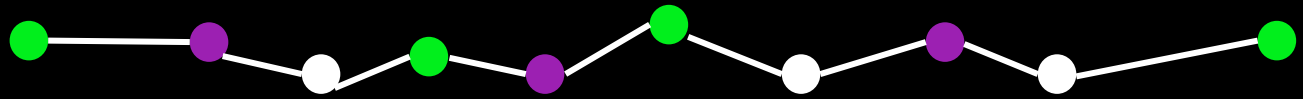


9. Codeium - https://codeium.com/

Descrição: Codeium é uma plataforma focada em gerar código e ajudar a escrever melhores trechos de código com IA, semelhante ao Copilot e Tabnine.
Casos de uso:

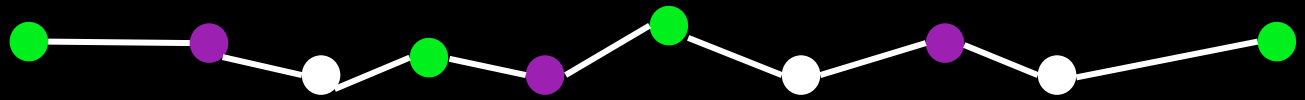
- **Geração de código inteligente**: Oferece sugestões rápidas com base no contexto do código.
- **Refatoração e otimização de código**: Ajuda a melhorar o código existente.

Integração ao fluxo de trabalho: Pode ser integrado ao VSCode e outros editores para um fluxo de trabalho mais eficiente.



10. Gemini (Google DeepMind) - https://gemini.google.com/

Descrição: Gemini é uma suíte de IA voltada para ajudar em várias áreas, inclusive no desenvolvimento de software, com foco em tarefas complexas e análises detalhadas.
Caso de uso: Ideal para tarefas avançadas como otimização de algoritmos e análise de desempenho.
Integração: Usado principalmente por meio de APIs do Google Cloud, permitindo integração com outros serviços e pipelines.



Existem várias outras....

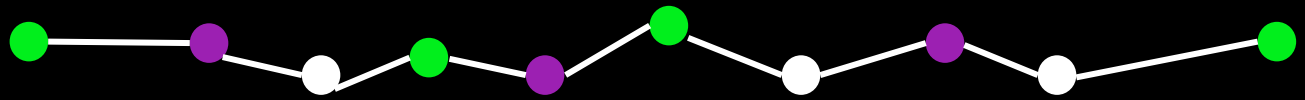
- Kite - https://www.kite.com/blog/product/kite-is-saying-farewell/
- Codex (OpenAI) - https://openai.com/index/openai-codex/
- Replit - https://replit.com/
- Sourcery - https://sourcery.ai/
- Polycoder - https://www.youtube.com/watch?v=8FY1ZPm6lQY
- Mintlify - https://mintlify.com/
- AskJarvis - https://jarvis.cx/
- SonarQube - https://www.sonarsource.com/products/sonarqube/
- CodeQL - https://codeql.github.com/
- DeepCode - https://www.deepcode.ca/
- Codiga - https://www.codiga.io/
- Snyk - https://snyk.io/pt-BR/platform/deepcode-ai/
- Poe - https://poe.com/

Parte 4

Como melhorar seu Prompt

para Obter Respostas Mais Eficientes

a)	O que é um “prompt”?
b)	Técnicas e estratégias para melhorar a eficácia dos prompts



O que é um “prompt”?

Um **prompt** é o termo usado para se referir à instrução, pergunta ou comando que você fornece a uma IA para receber uma resposta.

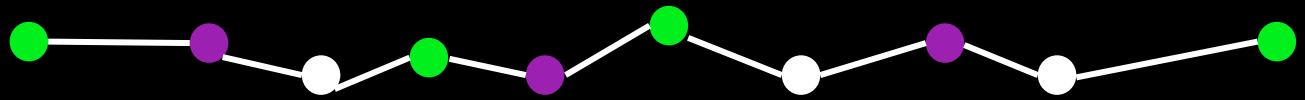
No contexto de IA, o prompt é a maneira como você se comunica com o modelo, dizendo exatamente o que deseja saber, qual tipo de resposta espera, ou até o formato que quer que a resposta tenha.

Exemplo básico de prompt:

- "Explique o que é um algoritmo."

Exemplo de prompt mais avançado e específico:

- "Explique o que é um algoritmo usando exemplos do dia a dia e evitando termos técnicos, para que uma pessoa leiga possa entender facilmente."

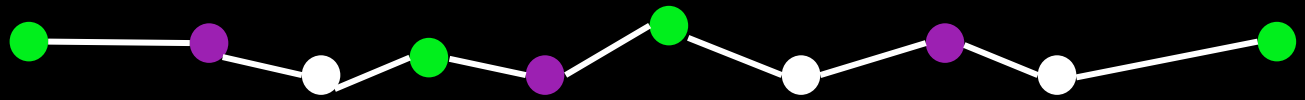


Técnicas e estratégias para melhorar a eficácia dos prompts

Para otimizar a formulação do prompt e obter respostas mais precisas e detalhadas de ferramentas de IA, é essencial seguir algumas técnicas que ajudam a IA a entender melhor o que você deseja.

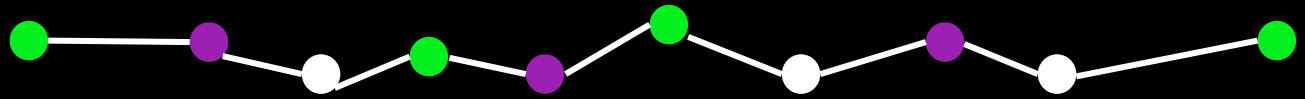
A “alucinação” (quando a IA inventa ou fornece respostas imprecisas) pode ser minimizada com prompts bem estruturados, claros e com o contexto necessário.

Vamos ver algumas estratégias para melhorar a eficácia dos prompts:



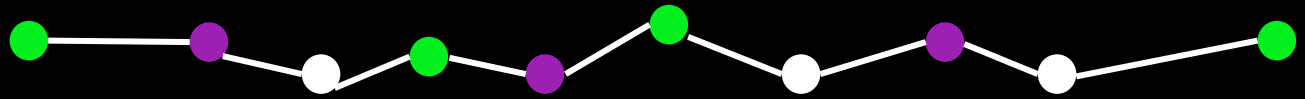
1. Especificidade e Clareza

Seja específico: Quanto mais específico você for, melhores serão as respostas. Evite perguntas genéricas e forneça detalhes contextuais sempre que possível.
Defina o formato de resposta: Se você deseja uma lista, um parágrafo explicativo ou um código, especifique isso no prompt.
Exemplo: Ao invés de "Explique machine learning", tente "Explique os conceitos básicos de machine learning com exemplos práticos em Python e foco em algoritmos supervisionados."



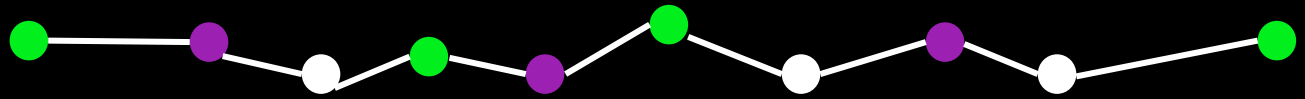
2. Divida a Pergunta em Partes

Evite pedir muita coisa de uma só vez: Quebre perguntas complexas em partes menores e faça uma pergunta de cada vez, se possível. Isso ajuda a IA a focar em uma tarefa específica.
Exemplo: Em vez de "Explique machine learning e sugira aplicações práticas", divida em "Explique os fundamentos de machine learning" e, depois, "Sugira algumas aplicações práticas de machine learning para iniciantes".



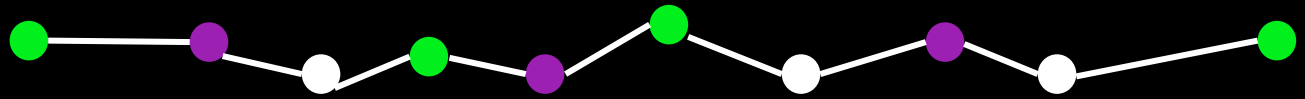
3. Ofereça Contexto e Informação Adicional

Contexto é essencial: Diga à IA qual é o propósito da resposta ou o público-alvo. Por exemplo, se é para iniciantes, intermediários ou avançados, ou se é para um tutorial ou apresentação.
Exemplo: "Explique como funcionam redes neurais para um público iniciante com exemplos visuais e evite jargões técnicos."



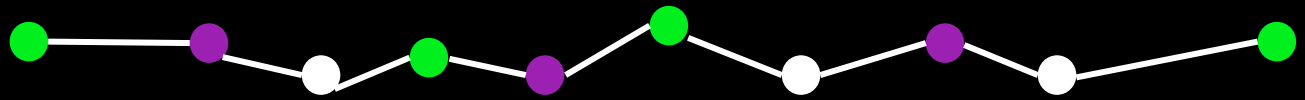
4. Use Perguntas de Acompanhamento (Follow-Up)

Construa sobre as respostas anteriores: Se a primeira resposta não foi exatamente o que você esperava, faça perguntas de acompanhamento específicas. Isso evita repetir o prompt e ajuda a IA a refinar a resposta com base no que já foi discutido.
Exemplo: “Gostei da explicação sobre redes neurais, mas poderia dar mais exemplos visuais? Pode explicar como esses conceitos se aplicam no reconhecimento de imagens?”



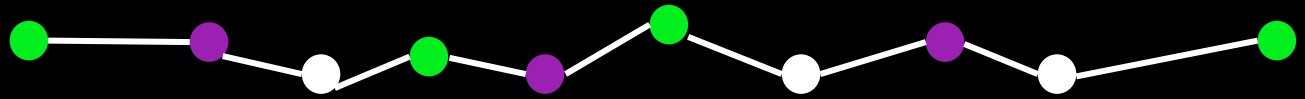
5. Defina Restrições e Limitações

Restringir o escopo evita alucinação: Especifique limites claros no prompt, como pedir que a IA só use fontes de código aberto ou que faça referência apenas a tecnologias específicas.
Exemplo: "Escreva uma função em Python usando apenas bibliotecas padrão para calcular a média de uma lista de números."



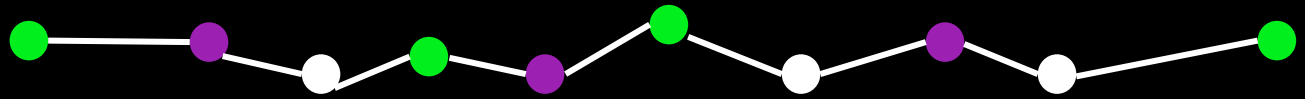
6. Experimente Várias Estruturas de Prompt

Teste diferentes maneiras de perguntar: Às vezes, reestruturar a forma de fazer a pergunta pode produzir respostas mais claras e precisas.
Exemplo: Se o prompt "Quais são os usos de IA no desenvolvimento de software?" não gerar a resposta desejada, tente "Liste e explique 5 maneiras práticas de usar IA para melhorar o desenvolvimento de software."



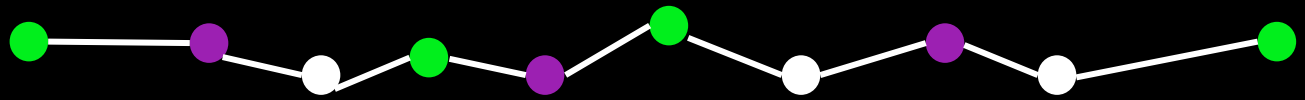
7. Dê Exemplos Concretos

Exemplos tornam o prompt mais claro: Mostrar exemplos de resposta ou do tipo de informação que você espera facilita para a IA seguir o mesmo padrão.
Exemplo: "Quero exemplos práticos de refatoração de código com IA. Exemplo: transformar funções aninhadas em funções independentes com melhores nomes."



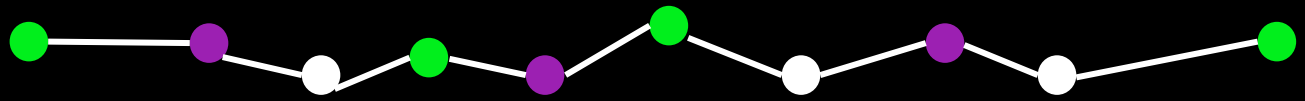
8. Peça para Confirmar ou Validar Fontes

Solicite referências ou fontes confiáveis: Especifique que a IA deve basear as respostas em dados conhecidos ou indicar fontes quando possível.
Exemplo: "Explique o que é uma árvore de decisão, citando artigos ou livros de referência."



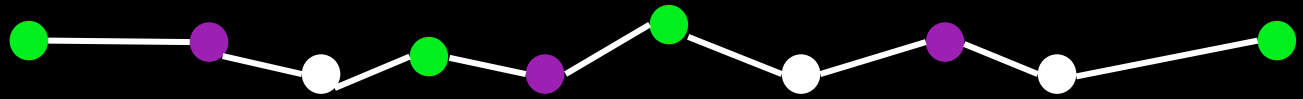
9. Use o Estilo de Conversa Esclarecedora

Dialogue com a IA como faria com uma pessoa: Expresse dúvidas ou expectativas de forma direta e peça uma “revisão” ou “explicação detalhada” quando necessário. Isso reduz a probabilidade de respostas vagas.
Exemplo: "Você poderia revisar este código que eu escrevi e sugerir melhorias de performance? Além disso, explique por que cada melhoria é recomendada."



10. Peça Alternativas ou Exemplos Comparativos

Prompts para gerar alternativas: Peça para a IA sugerir mais de uma solução ou fornecer exemplos de abordagens diferentes, o que enriquece o conteúdo e minimiza respostas unilaterais.
Exemplo: "Mostre duas alternativas para implementar uma função de autenticação em uma API RESTful usando Express.js, e explique as vantagens e desvantagens de cada uma."



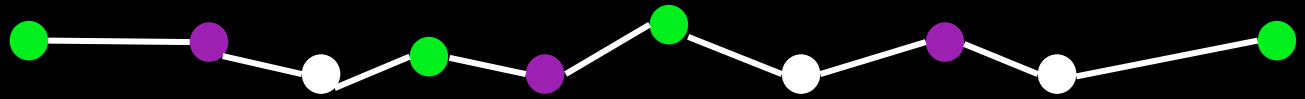
Técnicas Avançadas

Engenharia de Prompts: É a arte de criar prompts que maximizam a capacidade da IA de gerar respostas úteis e relevantes. Envolve técnicas como:

- **Few-shot learning:** Fornecer alguns exemplos de entrada e saída desejada para a IA.
- **Chain-of-thought prompting:** Quebrar um problema em passos menores e guiar a IA através do processo de raciocínio.

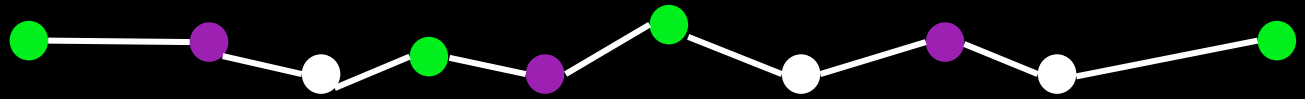
Prompt Chaining: Conectar múltiplos prompts para criar um diálogo mais complexo e obter resultados mais sofisticados.

Role Playing: Definir um papel para a IA (por exemplo, uma professora, uma cientista de dados) para moldar suas respostas.



Evite:

- **Perguntas abertas demais:** "O que você acha da inteligência artificial?"
- **Instruções contraditórias:** "Diga-me tudo sobre gatos, mas não me diga nada sobre gatos."
- **Informações falsas ou enganosas:** A IA aprenderá com os dados que você fornecer.



Ferramentas Úteis:

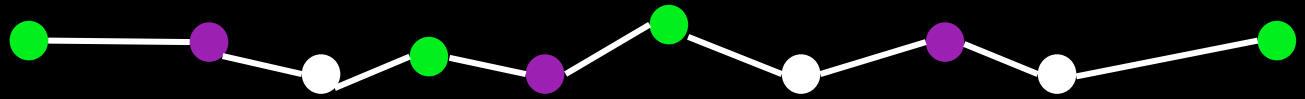
PromptHero: Uma plataforma que permite criar, compartilhar e avaliar prompts.
Hugging Face: Uma comunidade e plataforma de código aberto para machine learning, com muitos recursos para engenharia de prompts.

Parte 4

Riscos, Desafios e Cuidados

no Uso da IA

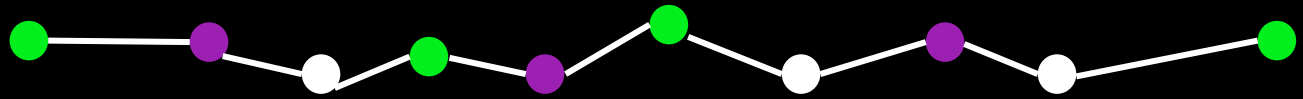
a)	Riscos, Desafios e Cuidados



Riscos, Desafios e Cuidados

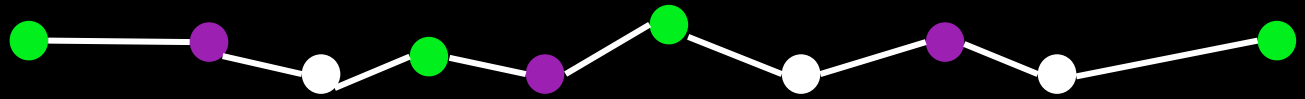
O uso de IA no desenvolvimento de software traz muitas vantagens, mas também exige cautela, pois existem riscos e desafios que precisam ser compreendidos e gerenciados.

A seguir estão os principais pontos a serem considerados para que o uso de IA seja feito de forma segura e eficaz, preservando a qualidade do trabalho e a segurança das informações.



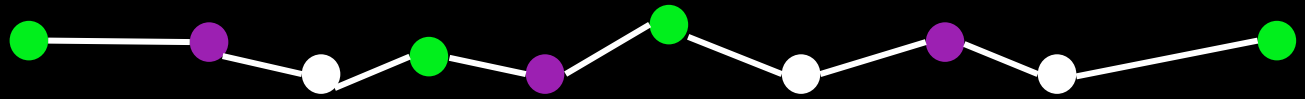
1. Limitações das Ferramentas de IA

Falta de Compreensão Completa: A IA não entende conceitos como um ser humano. Ela gera respostas com base em padrões de dados, mas pode fornecer informações incorretas ou incompletas, especialmente em questões complexas ou que exigem raciocínio crítico.
Alucinações (Informações Inventadas): IAs podem gerar respostas inventadas que parecem confiáveis. Elas podem sugerir soluções ou explicações erradas, que, se não forem validadas, podem levar a problemas no código.
Dependência de Dados Treinados: A IA responde com base em dados já disponíveis e treinados, então pode estar desatualizada e não conhecer tecnologias, padrões ou práticas recentes.



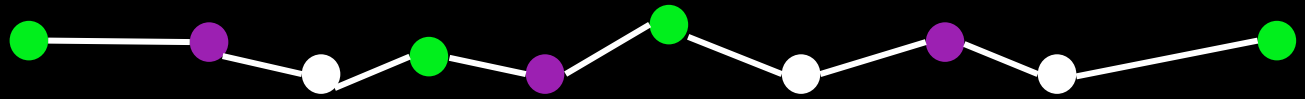
2. Risco de Dependência Excessiva

Perda de Habilidade Crítica: Confiar demais na IA pode reduzir a capacidade de análise crítica e resolução de problemas. Pessoas desenvolvedoras podem perder a habilidade de identificar e resolver bugs por conta própria, prejudicando a autonomia.
Redução do Aprendizado: Dependência da IA pode fazer com que pessoas desenvolvedoras menos experientes deixem de aprender habilidades essenciais, confiando apenas na IA para resolver problemas ou tomar decisões de design.
Impacto na Criatividade: Quando usada excessivamente, a IA pode limitar o desenvolvimento de soluções criativas. As respostas de IA seguem padrões que podem restringir a inovação, tornando o código e as soluções menos originais.



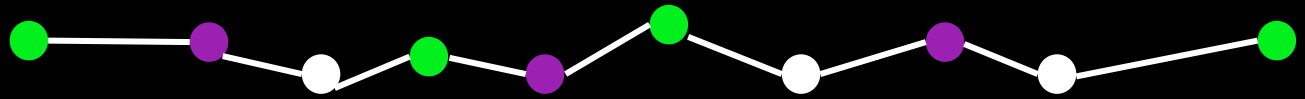
3. Questões Éticas e Viés

Viés nos Modelos de IA: A IA é treinada com dados que podem conter vieses e preconceitos. Isso pode levar a decisões e respostas enviesadas, impactando negativamente a diversidade, a inclusão e a justiça em projetos de software.
Preocupações com Direitos Autorais: Muitos modelos de IA foram treinados com dados que incluem material com direitos autorais. Isso pode gerar questões legais e éticas sobre o uso de código ou conteúdo gerado pela IA.
Responsabilidade pelas Decisões: Em casos de falhas ou problemas no software, o uso de IA pode complicar a identificação de responsabilidades, pois parte do código ou das decisões pode ter sido gerada automaticamente.



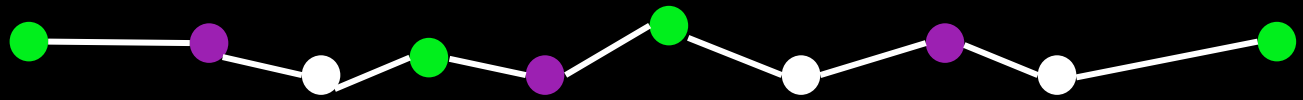
4. Segurança da Informação

Risco de Exposição de Dados: Usar IA com informações confidenciais ou sensíveis pode representar riscos. Dados enviados para ferramentas de IA podem ser armazenados e potencialmente vazados ou utilizados de forma indevida.
Exposição de Código Proprietário: Usar IA para resolver problemas ou otimizar código pode resultar na exposição de código proprietário ou sensível. Ferramentas de IA externas, especialmente, podem coletar informações do código e usá-las para melhorar seus próprios modelos.
Ataques Baseados em IA: Hackers podem utilizar ferramentas de IA para automatizar ataques ou explorar vulnerabilidades. Isso pode ser especialmente perigoso se um desenvolvedor usa a IA para gerar código sem fazer revisões e testes de segurança adequados.



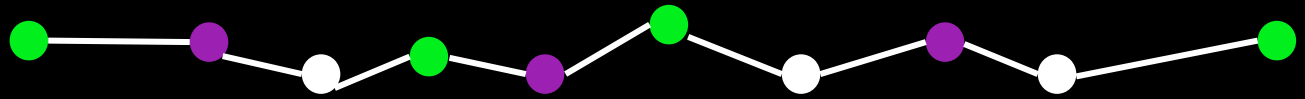
5. Desafios de Qualidade e Confiabilidade

Respostas Superficiais: Ferramentas de IA muitas vezes oferecem respostas "genéricas", que podem não ser adequadas para problemas específicos ou complexos. É essencial validar essas respostas para garantir que estejam corretas.
Incertezas em Problemas Complexos: A IA pode não ser confiável em problemas que exigem conhecimento profundo ou análise de contexto. Nessas situações, é importante fazer uma revisão detalhada e complemente o que foi gerado pela IA com conhecimento humano.



6. Cuidados no Uso da IA no Desenvolvimento de Software

Use a IA Como Ferramenta Complementar, Não Substitutiva: A IA deve ser vista como um suporte que ajuda a automatizar tarefas e a inspirar soluções, mas nunca como um substituto para o conhecimento e a criatividade da pessoa desenvolvedora.
Valide e Revise Sempre as Respostas: Em tarefas críticas, como a solução de bugs ou a refatoração de código, é fundamental revisar o que a IA sugere e fazer testes para verificar a eficácia e a segurança da solução.
Desenvolva a Habilidade Crítica e a Intuição Técnica: Trabalhe para desenvolver suas habilidades e conhecimento, e use a IA para complementar, não substituir, essa expertise. A IA pode sugerir soluções, mas a pessoa desenvolvedora deve ser capaz de avaliar a viabilidade e adequação dessas sugestões.
Priorize a Privacidade e a Segurança dos Dados: Nunca insira dados confidenciais ou sensíveis em ferramentas de IA sem verificar as políticas de privacidade e segurança. Se necessário, use uma versão local de IA ou evite usar IA para essas tarefas.



7. Quando Confiar nas Respostas e Quando Revisar?

Confiança Parcial para Tarefas Simples: Em tarefas repetitivas ou simples, como sugestões de documentação, pequenos exemplos de código ou auxílio em dúvidas básicas, a IA pode ser confiável.
Revisão Rigorosa em Tarefas Críticas e Sensíveis: Sempre que a IA gera código ou soluções para funções sensíveis, que envolvem dados importantes ou exigem alta performance, é essencial validar cada detalhe e fazer testes rigorosos.
Validação Constante em Questões de Segurança: Qualquer sugestão relacionada a segurança, criptografia, ou manipulação de dados sensíveis deve ser revisada por especialistas para garantir que está de acordo com as melhores práticas.

PHPeste
#PEC2024

INFORMAÇÃO IMPORTANTE

Essa palestra foi gerada

com auxílio da IA

PHPeste
#PEC2024

Obrigada!!!

@mocraveirodev