



Imersão em GenAI para Desenvolvedores

Projeto 6: Assistente de IA para Desenvolvimento de Software

Preâmbulo: Neste projeto, você vai explorar o Continue.dev, criar testes unitários e analisar código, utilizando IA para aumentar sua produtividade no desenvolvimento de software.

Versão: 1.0

Sumário

I	Preâmbulo	2
II	Introdução	3
III	Instruções gerais	4
IV	Ex00 – Configuração do ambiente Continue.dev	6
V	Ex01 – Análise de codebase do ChromaDB	8
VI	Ex02 – Quebra estratégica de informações para uma busca ainda melhor	10
VI.1	Exercício	11
VII	Entrega e Avaliação entre pares	14
VII.1	Processo de Entrega	14
VII.2	Avaliação entre pares	14
VII.3	Dicas para uma avaliação bem-sucedida	14

Capítulo I

Preâmbulo

A integração de assistentes de IA no desenvolvimento de software promete uma transformação significativa na produtividade dos desenvolvedores. Estudos recentes mostram que essas ferramentas podem aumentar consideravelmente a eficiência, especialmente para desenvolvedores menos experientes.

Um estudo realizado com 4.867 desenvolvedores em três grandes empresas revelou que:

1. Aumento de produtividade:

- 26,08% de aumento no número de tarefas concluídas
- 13,55% de aumento no número de commits de código
- 38,38% de aumento no número de compilações de código

2. Impacto heterogêneo:

- Desenvolvedores juniores e com menor tempo de empresa apresentaram maiores ganhos de produtividade
- Maior taxa de adoção entre desenvolvedores menos experientes

Estes resultados indicam que:

Assistentes de IA, como o GitHub Copilot, têm o potencial de nivelar o campo de atuação, permitindo que desenvolvedores menos experientes alcancem níveis de produtividade mais próximos aos de seus colegas seniores.

Portanto, ao implementar assistentes de IA em ambientes de desenvolvimento, devemos:

Focar na integração efetiva e no treinamento, especialmente para desenvolvedores juniores e recém-contratados.

— Baseado no trabalho de Cui et al. (2024)¹

¹Cui, K. Z., Demirer, M., Jaffe, S., Musolff, L., Peng, S., & Salz, T. (2024). The Effects of Generative AI on High Skilled Work: Evidence from Three Field Experiments with Software Developers.

Capítulo II

Introdução

Os assistentes de IA para desenvolvimento de software estão se tornando rapidamente parte essencial do trabalho de muitos desenvolvedores. Eles oferecem sugestões de código, ajudam na manutenção e podem aumentar significativamente a produtividade. No entanto, é fundamental entender suas capacidades e limitações para usá-los de forma eficaz.

Nestes projetos, o uso de assistentes de IA não apenas é permitido, mas incentivado. Porém, é importante lembrar:

- A ferramenta deve complementar, e não substituir, seu raciocínio e compreensão.
- O aprendizado efetivo ainda depende da sua interação ativa com o conteúdo e da aplicação crítica do conhecimento.
- Sua capacidade de explicar e justificar as soluções geradas pela IA é fundamental.

Vale ressaltar que a configuração e a exploração inicial dessas ferramentas podem ser um processo trabalhoso. Paciência e atenção aos detalhes são essenciais nesta fase. O tempo investido agora pode resultar em ganhos significativos de produtividade no futuro.

Capítulo III


Instruções gerais

- Esta página é sua única referência oficial. Não confie em informações não verificadas.
- Os exercícios estão organizados em ordem crescente de complexidade. É essencial dominar cada exercício antes de prosseguir para o próximo.
- Preste atenção às permissões de seus arquivos e pastas.
- Siga rigorosamente o procedimento de entrega para todos os exercícios.
- Seus exercícios serão avaliados por colegas da Imersão.
- Para exercícios em Shell, utilize `/bin/zsh`.
- Mantenha em sua pasta apenas os arquivos explicitamente solicitados nos enunciados.
- Em caso de dúvidas, consulte seus colegas à direita ou à esquerda.
- Utilize recursos como Google, manuais online e a Internet como referência.
- Leia os exemplos com atenção. Eles podem conter requisitos não explicitamente mencionados no enunciado.
- Para exercícios em Python:
 - Use a versão do Python especificada no exercício de configuração do ambiente.
 - Utilize os modelos e provedores sugeridos para garantir tempos de resposta adequados e consistência nos testes.
- Esteja atento a erros em todos os exercícios. Eles raramente são tolerados durante a avaliação.
- **Aviso sobre o uso de ferramentas de AI (como ChatGPT):**
 - O uso de ferramentas como o ChatGPT não deve ser encarado como um substituto para seu próprio esforço e entendimento.
 - O aprendizado efetivo ocorre quando você interage ativamente com o conteúdo: pesquisando, refletindo e aplicando o que aprendeu.
 - Nas avaliações, serão feitas perguntas para avaliar sua compreensão real sobre o assunto.

- E durante as avaliações, seus colegas também avaliarão seu nível de conhecimento.

Capítulo IV

Ex00 – Configuração do ambiente Continue.dev

	Exercício : 00
Instalar, configurar e explorar o Continue.dev para melhorar sua produtividade no desenvolvimento de software	
Pasta de entrega : <i>ex00/</i>	
Arquivos para entregar : Nenhum arquivo necessário	

Neste exercício, vamos configurar uma ferramenta para o uso da IA no desenvolvimento de software, o Continue.dev. Os assistentes de IA estão se tornando parte essencial do trabalho de muitos desenvolvedores, oferecendo sugestões de código, ajudando na manutenção e potencialmente aumentando a produtividade. É fundamental entender suas capacidades e limitações para usá-los de forma eficaz.

Durante este projeto, diferentemente de outros, encorajamos fortemente o uso de assistentes de IA. Você vai utilizar e experimentar com essas ferramentas na realização dos exercícios: este é um momento para explorar plenamente as capacidades desses assistentes e entender como eles podem ser integrados ao seu fluxo de trabalho.

No entanto, mesmo com esta liberdade, é crucial manter em mente:

- Embora o uso de IA seja incentivado, seu entendimento do processo e dos resultados continua sendo fundamental.
- O objetivo é aprender a trabalhar *com* a IA, não apenas deixá-la fazer todo o trabalho.
- Sua capacidade de explicar e justificar as soluções geradas ou sugeridas pela IA permanece fundamental para seu aprendizado.

Instruções

1. Instalação do Continue.dev:

- Instale a extensão Continue.dev no VSCode através do marketplace.
- Link: <https://marketplace.visualstudio.com/items?itemName=Continue.continue>

2. Configuração da API:

- Obtenha uma chave de API (Groq para o modelo llama3-8b-8192 ou gemini-1.5-flash da Google).
- Configure a extensão Continue.dev com a chave de API escolhida.

3. Exploração básica:

- Familiarize-se com a interface do Continue.dev no VSCode.
- Teste algumas funcionalidades básicas, como completar código ou gerar comentários.




Se encontrar dificuldades durante a instalação ou configuração, consulte a documentação oficial do Continue.dev ou peça ajuda a um colega.



A configuração e exploração inicial dessas ferramentas podem ser trabalhosas. Paciência e atenção aos detalhes são essenciais nesta fase. O tempo investido agora pode resultar em ganhos significativos de produtividade no futuro.

Capítulo V

Ex01 – Análise de codebase do ChromaDB

	Exercício : 01
Utilizar ferramentas de AI para analisar e compreender uma funcionalidade específica dentro da codebase do ChromaDB	
Pasta de entrega : <i>ex01/</i>	
Arquivos para entregar : <code>chromadb_analysis.md</code>	

Neste exercício, você vai exercitar suas habilidades de análise de código e compreensão de dependências em um projeto de código aberto. Você vai escolher uma funcionalidade específica dentro do ChromaDB e usar ferramentas de IA para analisar e documentar seu funcionamento.

Instruções

1. Seleção de funcionalidade:

- Escolha uma funcionalidade específica na codebase do [ChromaDB - \(documentação\)](#) para analisar (por exemplo, armazenamento de embeddings, indexação, busca ou persistência).
- Use a AI para identificar os arquivos e módulos relevantes para essa funcionalidade.

2. Análise de código:

- Utilize a ferramenta de IA para gerar uma explicação detalhada do fluxo de execução da funcionalidade escolhida.
- Peça à IA para identificar possíveis pontos de melhoria ou otimização no código.

3. Compreensão de dependências:

- Use a IA para mapear as dependências internas e externas da funcionalidade.
- Documente as bibliotecas e frameworks utilizados, com explicações geradas pela IA sobre seu propósito no projeto.

4. Relatório de análise:

- Crie um arquivo Markdown (chromadb_analysis.md) com o seguinte formato:

```
# Análise ChromaDB
```

```
## Funcionalidade Escolhida
```

```
[Nome da funcionalidade]
```

```
Breve descrição da funcionalidade e razão da escolha (6-10 frases)
```

```
## Análise do Código
```

- Principais arquivos/módulos envolvidos: [lista]
- Fluxo de execução resumido: [breve descrição]
- Pontos de melhoria identificados: [lista]

```
## Dependências
```


- Internas: [lista de módulos internos]
- Externas: [lista de bibliotecas/frameworks externos]
- Propósito principal de uma dependência chave: [breve explicação]



Utilize a IA para entender terminologias específicas relacionadas a sistemas de armazenamento vetorial e busca semântica.

Capítulo VI

Ex02 – Quebra estratégica de informações para uma busca ainda melhor

	Exercício : 02
Implementar um sistema de chunking para melhorar a granularidade e eficácia da busca semântica em documentos PDF, focando em currículos	
Pasta de entrega : <i>ex02/</i>	
Arquivos para entregar : <code>secure_resume_api.py</code>	

Chunking é o processo de dividir documentos longos em partes menores e mais gerenciáveis, chamadas de "chunks" ou fragmentos¹. No contexto de processamento de linguagem natural e sistemas de recuperação de informação, o *chunking* é uma etapa crucial por várias razões:

- **Gerenciamento de memória:** Documentos muito longos podem ser difíceis de processar de uma só vez devido a limitações de memória.
- **Granularidade na recuperação:** Permite recuperar trechos específicos e relevantes de um documento, em vez do documento inteiro.
- **Melhoria na assertividade:** Ajuda a focar a busca nas seções mais relevantes do texto.
- **Contextualização:** Mantém o contexto local dentro de cada trecho, o que é importante para entender o significado.

¹<https://www.pinecone.io/learn/chunking-strategies/>

Chunking simples vs. avançado

Chunking simples:

- Divide o texto em partes de tamanho fixo (por exemplo, a cada 1000 caracteres).
- Fácil de implementar, mas pode cortar frases ou parágrafos no meio.

Chunking avançado:

- Usa técnicas mais sofisticadas para dividir o texto de forma mais significativa.
- Pode considerar estruturas como parágrafos, frases ou até mesmo a semântica do conteúdo.

Como o chunking auxilia na busca semântica

A busca semântica visa entender a intenção e o contexto por trás de uma consulta, indo além da simples correspondência de palavras-chave. O chunking melhora a busca semântica de várias maneiras:

- **Relevância localizada:** Permite encontrar trechos específicos de um documento que são mais relevantes para uma consulta.
- **Contexto preservado:** Trechos bem definidos mantêm o contexto local, crucial para a compreensão do significado.
- **Eficiência:** Facilita a indexação e a busca em grandes volumes de texto.
- **Flexibilidade:** Permite ajustar o tamanho dos trechos para diferentes tipos de documentos ou necessidades de busca.

Desafios em realizar chunking

- **Tamanho do chunk:** Trechos muito pequenos podem perder o contexto, enquanto trechos muito grandes podem ser imprecisos.
- **Sobreposição:** Decidir quanta sobreposição ter entre trechos adjacentes para manter a continuidade.
- **Estrutura do documento:** Respeitar a estrutura natural do documento (parágrafos, seções) ao fazer a divisão em trechos.

VI.1 Exercício

Neste exercício, você implementará uma API segura usando Flask para processamento de currículos, utilizando chunking e incorporando medidas de segurança baseadas no OWASP API Security Top 10. Focaremos especificamente em:

- [API4:2023 - Unrestricted Resource Consumption](#)
- [API5:2023 - Broken Function Level Authorization](#)

As features implementadas devem levar em conta os cenários de teste dessas vulnerabilidades, conforme descritos pela OWASP, visando mitigar esses tipos de cenários na sua API.

Instruções

1. Implementar a função `chunk_text_recursive` usando `RecursiveCharacterTextSplitter`.
2. Criar uma API usando Flask com os seguintes endpoints:
 - `POST /upload_pdf`: para receber e processar um arquivo PDF de currículo
 - `GET /search`: para realizar buscas semânticas nos currículos processados
 - `DELETE /curriculum/<id>`: para excluir um currículo
3. Implementar um sistema de autenticação e autorização com três níveis de acesso:
 - Candidato: pode fazer upload de seu próprio currículo e buscar em currículos públicos
 - Recrutador: pode buscar em todos os currículos
 - Administrador: pode fazer tudo, incluindo excluir currículos
4. Implementar limites de taxa (rate limiting) para uploads de PDF e buscas, considerando os cenários de teste da OWASP para API4:2023.
5. Implementar controles de acesso baseados em função, levando em conta os cenários de teste da OWASP para API5:2023.
6. Integrar o armazenamento dos chunks processados no ChromaDB.
7. Implementar uma função de busca semântica que aproveite os chunks mais granulares.

Código base:

```
def chunk_text_recursive(text, chunk_size, chunk_overlap, separators):  
    pass  
  
def upload_pdf():  
    pass  
  
def search():  
    pass  
  
def delete_curriculum(id):  
    pass  
  
if __name__ == '__main__':  
    pass
```

Saída esperada:

```
POST /upload_pdf  
Response: {"message": "PDF processado com sucesso", "chunks_created": 5}  
  
GET /search?query=python  
Response: [  
  {  
    "document": "curriculo_12.pdf",  
    "chunk": 2,  
    "content": "Experiência profissional: Desenvolvedor Python Sênior na TechSolutions (2018-2022)..."  
  },  
  {  
    "document": "curriculo_35.pdf",  
    "chunk": 1,  
    "content": "Resumo Profissional: Engenheiro de Software com 6 anos de experiência em desenvolvimento Python..."  
  }  
]  
  
DELETE /curriculum/123 (com token de administrador)  
Response: {"message": "Currículo excluído com sucesso"}  
  
DELETE /curriculum/123 (com token de candidato)  
Response: {"error": "Acesso não autorizado"}  
  
POST /upload_pdf (excedendo o limite de taxa)  
Response: {"error": "Limite de taxa excedido. Tente novamente mais tarde."}
```



Considere implementar testes que simulem os cenários de ataque descritos pela OWASP para garantir que suas medidas de segurança sejam eficazes.



Ao implementar os controles de segurança, certifique-se de que eles não possam ser contornados por manipulação de parâmetros de solicitação ou cabeçalhos HTTP.

Capítulo VII

Entrega e Avaliação entre pares

VII.1 Processo de Entrega

- Submeta seu trabalho no repositório Git gerado na página principal do projeto.
- Certifique-se de que todos os arquivos necessários estejam incluídos e organizados conforme as instruções do projeto.
- Respeite o prazo de entrega estabelecido.

VII.2 Avaliação entre pares

- Seu projeto será avaliado por um dos seus colegas.
- A avaliação focará na qualidade do seu código e na aderência aos requisitos do projeto.
- Critérios de avaliação podem incluir:
 1. Funcionalidade: O código atende a todos os requisitos especificados?
 2. Legibilidade: O código é claro e bem estruturado?
 3. Eficiência: As soluções implementadas são otimizadas e seguem boas práticas?
 4. Organização: Os arquivos e estrutura do projeto estão bem organizados?
- Feedback detalhado é esperado, mas pode variar em extensão e detalhamento.

VII.3 Dicas para uma avaliação bem-sucedida

- Revise seu código antes da submissão final.

- Teste exaustivamente todas as funcionalidades implementadas.
- Se entender necessário para clareza, documente claramente qualquer decisão ou suposição feita.
- Esteja preparado para explicar suas escolhas de implementação.



A avaliação entre pares é uma oportunidade para aprendizado e crescimento pessoal e profissional. Esteja aberto ao feedback recebido e use-o para aprimorar suas habilidades.