



UNIVERSIDADE
LUSÓFONA

Plugin Drop Project + ChatGPT

Trabalho Final de curso

Relatório Final

Aluno:

Miguel Rodrigues Miranda Silva

a22103822

Orientadores:

Bruno Cipriano e Rodrigo Correia

Trabalho Final de Curso | LEI | 28/06/2024

www.ulusofona.pt

Direitos de cópia

Plugin Drop Project + ChatGPT, Copyright de Miguel Rodrigues Miranda Silva, Universidade Lusófona.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona (UL) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

Este trabalho final de curso propõe integrar o ChatGPT no plugin da plataforma Drop Project, uma ferramenta de avaliação automática utilizada na Universidade Lusófona em três cadeiras dos cursos de LEI e LIG, para o IntelliJ IDEA. A implementação da funcionalidade de interação com o ChatGPT através da mesma tem o objetivo de facilitar a interação de alunos com ferramentas de inteligência artificial no ambiente de desenvolvimento. A solução procura melhorar a eficiência do utilizador através da redução de trocas de contexto entre IDE e *browser*. A integração no plugin permite também recolher dados dos utilizadores que o aceitem, desta forma torna-se possível estudar a utilização dos modelos de linguagem no desenvolvimento de software.

Abstract

This final project proposes the integration of ChatGPT into the Drop Project plugin, a tool used for automatic assessment at Universidade Lusófona in three courses within the LEI and LIG programs, for IntelliJ IDEA. The implementation aims to facilitate student interaction with artificial intelligence tools within the development environment. The solution seeks to enhance user efficiency by minimizing context switches between the IDE and the browser. Additionally, the integration within the plugin allows for data collection from consenting users, making it possible to study how language models are utilized in software development.

Índice

Resumo.....	4
Abstract	5
Índice	6
Lista de Figuras	8
Lista de Tabelas	9
1 Identificação do Problema	1
2 Viabilidade e Pertinência.....	2
3 Benchmarking.....	3
3.1 GitHub Copilot	3
3.2 ChatGPT no website.....	3
3.3 Google Bard	4
3.4 Code Help.....	4
3.5 AskGPT para Visual Studio Code	5
4 Engenharia.....	6
4.1 Levantamento e análise dos Requisitos.....	6
4.1.1 Focus Group	6
4.1.2 Requisitos definidos pelos Orientadores	6
4.1.3 Requisitos propostos pelo aluno	7
4.1.4 Requisitos provenientes do Focus Group.....	7
4.2 Diagramas de Casos de Uso	8
4.3 Diagrama de Sequência.....	8
5 Solução Desenvolvida.....	9
5.1 Introdução	9
5.2 Arquitetura	9
5.2.1 OpenAI.....	10
5.2.1.1 ChatGPT API	10
5.2.2 Drop Project [8]	10
5.2.2.1 Drop Project API	10
5.2.3 IntelliJ IDEA.....	11
5.2.3.1 Plugin Drop Project	11
5.3 Tecnologias e Ferramentas Utilizadas.....	12

5.3.1 Kotlin	12
5.3.2 Java Swing	12
5.3.3 API da OpenAI e ChatGPT.....	12
5.3.4 IntelliJ	12
5.4 Implementação	13
5.4.1 Recursos Computacionais	13
5.4.2 Dependência da API OpenAI	13
5.5 Abrangência	13
6 Recolha de Dados.....	14
7 Testes e Validação.....	15
7.1 Testes Definidos	15
7.1.1 Teste de Instalação e Configuração	15
7.1.2 Teste de Interação com o ChatGPT	15
7.1.3 Teste de Envio de Texto do Editor para o ChatGPT	16
7.1.4 Teste de Envio de Texto da Consola para o ChatGPT.....	16
7.1.5 Teste de Frases Prefixas à <i>Prompt</i>	16
7.2 Testes de Utilização.....	17
8 Calendário	19
9 Resultados.....	20
10 Conclusão	21
11 Bibliografia	22
Glossário.....	23

Lista de Figuras

Figura 1 - Estatísticas do questionário do AskGPT	5
Figura 2 - Diagrama de Casos de Uso	8
Figura 3 - Diagrama de Sequência	8
Figura 4 - Arquitetura da Solução Implementada	9
Figura 5 - Arquitetura para guardar dados remotamente	14

Lista de Tabelas

Tabela 1 - Requisitos definidos pelos Orientadores	6
Tabela 2 - Requisitos propostos pelo aluno	7
Tabela 3 - Requisitos provenientes do Focus Group	7
Tabela 4 - Bugs Reportados/Feedback	17

1 Identificação do Problema

Large Language Models (LLMs) são modelos de inteligência artificial treinados com uma vasta quantidade de dados textuais. São capazes de gerar linguagem humana com elevada precisão. Com o crescimento e evolução da inteligência artificial (IA) e de LLMs como o ChatGPT, torna-se cada vez mais comum a sua utilização no desenvolvimento de software. Ferramentas deste tipo têm o potencial de acelerar o processo de desenvolvimento e permitir identificar e corrigir problemas difíceis de detetar por humanos [1].

No caso do ChatGPT, a sua utilização envolve várias mudanças de contexto: o utilizador precisa de estar sempre a passar do seu ambiente de desenvolvimento para o *browser* e de volta cada vez que precisa de enviar uma *prompt* ou ler alguma resposta dada pelo LLM. Estas mudanças de contexto tendem a fazer com que o utilizador perca a concentração, reduzindo a sua produtividade, especialmente quando está mais cansado, cada vez que troca de contexto precisa de se “orientar” novamente.

Tendo em conta a utilidade do ChatGPT no desenvolvimento dos trabalhos de programação por parte dos alunos, iremos alterar o plugin do Drop Project, uma ferramenta de avaliação automática de trabalhos de programação, desenvolvida e utilizada na universidade Lusófona, de forma a integrar interações com o GPT. Isto também permite estudar as interações dos alunos com a inteligência artificial (IA) caso estes optem por partilhar os seus dados. Para além disso facilita a comunicação com o modelo de linguagem já que o podem fazer sem ter de sair do ambiente de desenvolvimento, evitando trocas de contexto.

Alguns professores da Universidade Lusófona, nomeadamente os professores Pedro Alves, Bruno Cipriano e Rodrigo Correia, têm interesse em perceber a forma como os alunos interagem com o ChatGPT. Com o IntelliJ, o ambiente de desenvolvimento utilizado em três cadeiras (“Fundamentos de Programação”, “Algoritmia e Estruturas de Dados” e “Linguagens de Programação 2”) ao longo dos cursos de informática da Universidade Lusófona (LEI, LIG e LEIRT) esta integração pode fornecer informações valiosas sobre o uso de IA por parte dos alunos no desenvolvimento de projetos de programação.

2 Viabilidade e Pertinência

A boa utilização de ferramentas de inteligência artificial (IA) no desenvolvimento de software tem o potencial de acelerar o processo de desenvolvimento [1] e detecção de problemas. Para tal é necessário estudar soluções que facilitem o uso de IA, bem como ensinar a usar as ferramentas da forma mais eficiente.

Dada a tendência de crescimento deste tipo de ferramentas [6] é importante analisar a forma como são utilizadas, tanto para melhoria das ferramentas em si, como da forma como são utilizadas. O registo de interações com o ChatGPT (*logs*) por parte de alunos num contexto universitário é pertinente para esta análise, visando otimizar a sua utilização.

De acordo com um estudo publicado pela universidade de Zurich [7], mudanças de contexto mais demoradas, que requerem uma quebra no raciocínio, quando se sai do IDE para utilizar uma ferramenta de IA, por exemplo, reduzem a produtividade. Por este motivo devem ser estudadas soluções que reduzam a quantidade de trocas de contexto ou as tornem mais rápidas.

Perto do fim de novembro de 2023 foi realizado um *Focus Group* com alunos da universidade com o objetivo de recolher *feedback* e sugestões para o plugin, os alunos mostraram interesse em experimentar o plugin e sugeriram funcionalidades que agora já estão implementadas no mesmo.

3 Benchmarking

Existem algumas soluções cujo âmbito funcional intersecta, de alguma forma, com o que se pretende desenvolver neste TFC. De seguida vamos analisar brevemente cada uma delas, e discutir as principais diferenças relativamente à solução desenvolvida neste projeto.

3.1 GitHub Copilot

O GitHub Copilot é uma ferramenta que completa código automaticamente, desenvolvida pelo GitHub em colaboração com a OpenAI, este oferece sugestões de código enquanto o utilizador escreve, semelhante aos sistemas de *auto-complete*, mas com capacidade de produção de algoritmos completos.

Quer esta ferramenta, quer a que será desenvolvida neste TFC, pretendem auxiliar os programadores na produção de código, recorrendo à utilização de LLMs. No entanto, o GitHub Copilot é mais intrusivo, vai oferecendo sugestões de código à medida que o utilizador trabalha. No estudo ““It’s Weird That it Knows What I Want”: Usability and Interactions with Copilot for Novice Programmers” [2], concluiu-se que, muitas vezes, programadores poucos experientes não são capazes de compreender ou aproveitar as soluções propostas pela ferramenta. O plugin Drop Project + ChatGPT, pelo contrário, não é intrusivo, apenas oferece sugestões quando solicitadas pelo programador.

Contudo, a versão Enterprise, a mais cara do GitHub Copilot, tem a capacidade de utilizar vários repositórios de uma organização como contexto para as suas respostas, bem como a capacidade de analisar *pull requests*, funcionalidades que têm o potencial de acelerar o desenvolvimento em projetos de maior dimensão. Informação retirada do site do github copilot: <https://github.com/features/copilot>

3.2 ChatGPT no website

A utilização do ChatGPT no website, do ponto de vista do programador, é uma experiência muito semelhante à sua utilização através do plugin proposto neste TFC. No entanto, a utilização do GPT através do website implica ter o *browser* aberto e mudar de janela inúmeras vezes. Além disso, os *logs* de chat guardados no website têm um tempo limite, ou seja, eventualmente serão perdidos. O plugin Drop Project + ChatGPT irá permitir guardar os *logs* de chat (caso o utilizador o permita) por um período superior, pois os mesmos ficarão gravados na base de dados do próprio Drop Project.

3.3 Google Bard

O Google Bard é um LLM que funciona de forma semelhante ao ChatGPT, a sua utilização é feita através do *browser*, portanto tem a desvantagem de mudança de contexto (tal como a utilização do ChatGPT no *browser*). Para além disso, numa análise das capacidades de geração de código entre o Bard e o GPT-3.5 [3], o GPT-3.5 foi capaz de resolver corretamente 90.6% dos problemas propostos, enquanto o Bard só conseguiu resolver corretamente 50.1% dos problemas. Sendo assim, entre os dois referidos, o modelo desenvolvido pela OpenAI parece ser a melhor opção para geração de código Java, pelo menos neste momento, e para o tipo de exercícios testados neste *paper* (exercícios introdutórios, retirados do site codingbat.com).

3.4 Code Help

O Code Help é uma ferramenta que atua como intermediário entre o utilizador e o modelo GPT [4]. A principal diferença entre a utilização do Code Help e a utilização normal do ChatGPT é que o Code Help não envia o pedido ao GPT tal como foi escrito e também faz alterações à resposta devolvida pelo modelo, com o objetivo de impedir a cópia de código diretamente das suas respostas, visto que a cópia tem o potencial de causar uma dependência de ferramentas de inteligência artificial especialmente para quem ainda está a aprender a programar.

De acordo com o artigo de Lau, S., & Guo, P. [5], no contexto de ensino há quem defenda que devem ser adotadas as ferramentas de IA, argumentando que é importante ensinar os alunos a utilizar corretamente estas ferramentas, de forma a estarem preparados para o futuro.

Por outro lado, segundo o artigo, alguns professores defendem que tais ferramentas devem ser banidas em contexto académico e que os princípios de programação devem continuar a ser lecionados como anteriormente.

Nas cadeiras dos professores orientadores (p.e. AED e LP2) não se pretende filtrar as respostas do GPT, mas sim ensinar os alunos a pensar criticamente sobre as mesmas. Por essa razão, a abordagem do Code Help, que filtra as respostas do GPT, não seria adequada.

3.5 AskGPT para Visual Studio Code

O AskGPT[9] é um plugin que foi desenvolvido para o Visual Studio Code. À semelhança do DropProject + ChatGPT também permite interagir com o modelo de linguagem através do IDE. No caso do AskGPT, os alunos usaram muito pouco o plugin, tendo sido feito um questionário para perceber o porquê.

Table 2: Reasons students did not use AskGPT.

Reason	Count	% Responses
May interfere with learning	25	57 %
Fear of breaking course rules	12	27 %
Awkward UI	9	20 %
Concerns about logging	7	16 %
Worse than alternatives	6	14 %
Responses were not useful	6	14 %
Did not want to use	4	9 %
Fear of overuse	4	9 %
Installation issues	2	5 %
Concerns about AI	2	5 %
Did not use VS Code	1	2 %

Figura 1 - Estatísticas do questionário do AskGPT

Analisando os resultados percebemos que 20% não utilizaram por causa de “Awkward UI”, talvez porque se afasta do modelo do ChatGPT no browser.

Outros 16 % apresentam preocupações relacionadas com o envio de *logs* para um servidor dos professores, é possível que o AskGPT não tenha tido opção de não enviar os dados.

Tendo em conta as respostas deste questionário, iremos evitar que o nosso plugin tenha os problemas que levaram os alunos a não utilizar o AskGPT.

4 Engenharia

4.1 Levantamento e análise dos Requisitos

Um dos principais objetivos deste projeto é facilitar a utilização da ferramenta de inteligência artificial ChatGPT, para tal visamos reduzir as trocas de contexto e tornar a interação com a LLM o mais fácil possível.

4.1.1 Focus Group

No final de novembro decorreu um focus group que reuniu alunos da universidade com o objetivo de discutir ideias relacionadas com o plugin. Tais como novas funcionalidades, proteção de dados e avaliação dos requisitos já definidos.

O focus group começou com uma demonstração do protótipo funcional, que consistia numa janela de Chat integrada no plugin já existente (Requisito com ID 1). Essa janela de Chat já permitia uma interação básica com o GPT. Os alunos foram então convidados a comentar essa funcionalidade, assim como outras que tinham sido pré-identificadas como potencialmente interessantes (p.e. Requisitos 2A e 4A).

Surgiram ideias consideradas muito úteis pelos alunos, como a funcionalidade de selecionar texto da consola e enviar diretamente para a caixa de texto do ChatGPT (Requisito 2B).

Para tal constam os seguintes requisitos.

4.1.2 Requisitos definidos pelos Orientadores

Tabela 1 - Requisitos definidos pelos Orientadores

ID	Descrição	Esforço (1-5)	Prioridade	Estado
1	Interação com o ChatGPT através da API - O plugin permite interagir com o ChatGPT através de uma aba no IntelliJ, reduzindo as trocas de contexto.	2	Must Have	Feito
2A	Send to ChatGPT - o plugin permite selecionar texto a partir do editor e enviá-lo para a caixa de texto do ChatGPT.	2	Must Have	Feito
4A	Lista de Frases Extra - O plugin permite editar uma lista de frases "comuns" que podem ser facilmente acrescentadas a uma prompt.	2	Must Have	Feito
4B	Lista de Frases Extra - O plugin permite escolher se a frase é adicionada no início ou fim do texto introduzido.	1	Should Have	Por Fazer
5	Persistência de logs de chat - As conversas dos utilizadores com o ChatGPT são guardadas de forma a que possam ser retomadas.	3	Should Have	Parcialmente Feito

7	Configurações de API Key - O plugin permite configurar a API Key utilizada para realizar os pedidos à OpenAI	2	Must Have	Feito
8	Associação dos <i>logs</i> ao respetivo <i>assignment</i> - O plugin associa os <i>logs</i> ao respetivo <i>assignment</i> de forma que seja possível estudar a interação feita para uma determinada tarefa.	3	Nice to Have	Por Fazer
9	Send to ChatGPT - o plugin permite enviar um erro para a caixa de texto do ChatGPT a partir do relatório de erros do assignment.	2	Must Have	Feito
10	Possibilidade de limpar a conversa – um botão na aba de <i>chat</i> limpa a conversa, removendo também o contexto.	2	Should Have	Feito
11	Botão Copy Code – Permite copiar o código da última resposta recebida no <i>chat</i> .	3	Should Have	Feito

4.1.3 Requisitos propostos pelo aluno

Tabela 2 - Requisitos propostos pelo aluno

ID	Descrição	Esforço (1-5)	Prioridade	Estado
3	Interface - O plugin dispõe de uma interface <i>user-friendly</i> semelhante à do ChatGPT no website.	3	Should Have	Feito
6	Persistência de <i>logs</i> de <i>chat</i> - O plugin permite que o utilizador organize os logs, dando-lhes nomes e podendo escolher qualquer um para retomar.	3	Nice to Have	Por Fazer

Estes requisitos foram propostos com o objetivo de garantir que não há nenhuma desvantagem da utilização do ChatGPT no plugin comparativamente à sua utilização no *browser*.

4.1.4 Requisitos provenientes do Focus Group

Tabela 3 - Requisitos provenientes do Focus Group

ID	Descrição	Esforço (1-5)	Prioridade	Estado
2B	Send to ChatGPT - o plugin permite selecionar texto a partir da consola e enviá-lo para a caixa de texto do ChatGPT.	2	Must Have	Feito

4.2 Diagramas de Casos de Uso

Segue-se o use case da interação do utilizador com o plugin quando comunica com o ChatGPT.

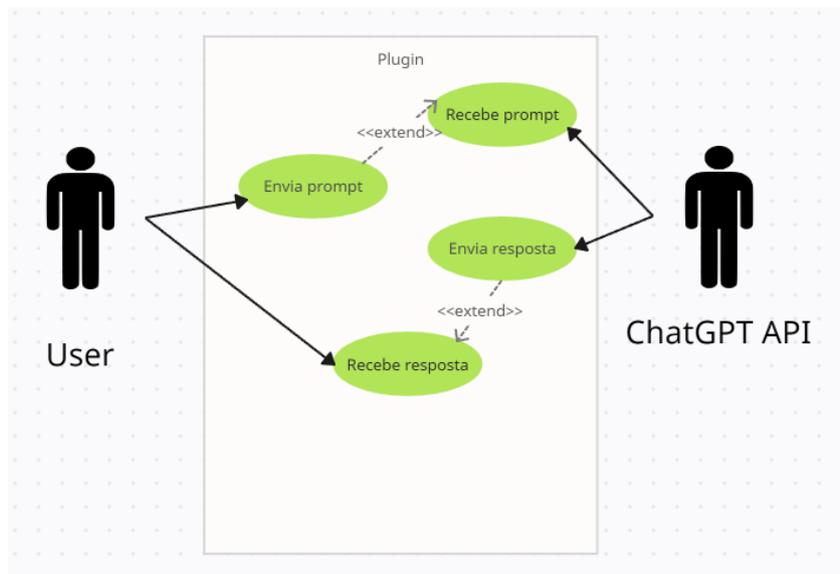


Figura 2 - Diagrama de Casos de Uso

4.3 Diagrama de Sequência

Segue-se um diagrama representativo da interação do plugin com a API da OpenAI que permite a interação com o ChatGPT.

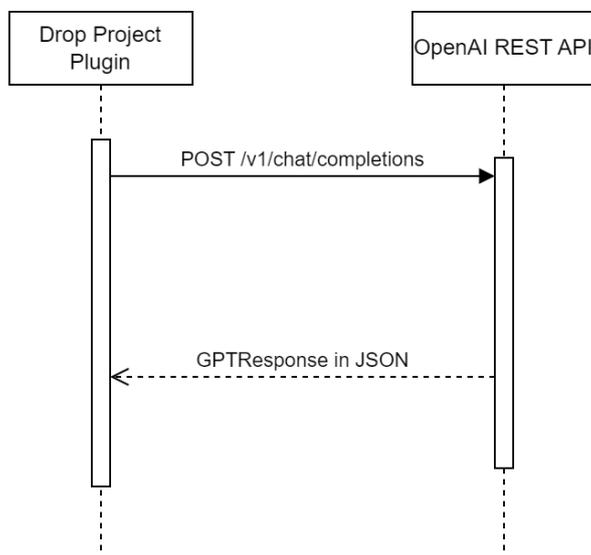


Figura 3 - Diagrama de Sequência

O plugin envia o pedido https incluindo a chave da API definida nas configurações, após receber a resposta da API processa o JSON recebido e adiciona a mensagem à janela de *chat*. Caso não obtenha a resposta da API apresenta uma notificação a avisar o utilizador.

5 Solução Desenvolvida

5.1 Introdução

A solução proposta atualiza o plugin Drop Project para IntelliJ, desenvolvido como trabalho final de curso por outro aluno da universidade, para permitir a interação com o ChatGPT no ambiente de desenvolvimento. Tal permite que os alunos interajam com a ferramenta de inteligência artificial sem a necessidade de trocas frequentes entre o IntelliJ e o *browser*, oferecendo também novas funcionalidades como o envio de texto diretamente do editor para o GPT.

A solução já foi implementada e testada por alunos, a maior parte dos problemas detectados foram resolvidos, mais detalhes na secção [Testes e validação](#). Aqui está um vídeo que demonstra o funcionamento do plugin: <https://youtu.be/3IFYuxgOveo?si=Ap0Zn2feNNbSEfAk>

5.2 Arquitetura

Na figura seguinte apresenta-se uma visão geral da arquitetura que está a ser implementada e que serve de base para este projeto.

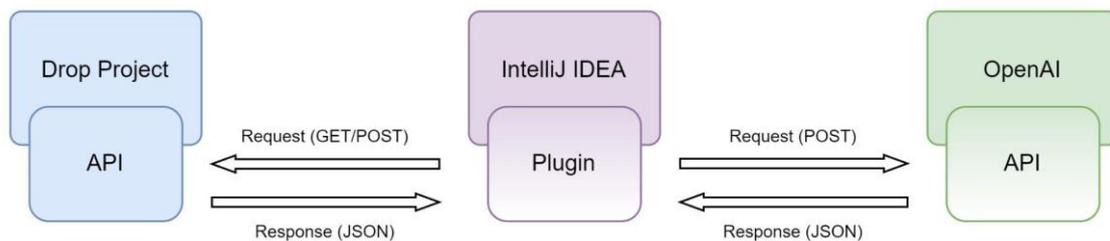


Figura 4 - Arquitetura da Solução Implementada

Segue-se a descrição de cada uma das partes envolvidas na arquitetura.

5.2.1 OpenAI

A OpenAI é uma organização com o objetivo de promover e desenvolver modelos de inteligência artificial. Fundada em 2015, a OpenAI desenvolveu modelos como o GPT-3 e o GPT-4, que são utilizados em diversas aplicações. A organização também colabora com outras instituições para desenvolver e explorar o potencial da inteligência artificial.

5.2.1.1 ChatGPT API

Através da API disponibilizada pela OpenAI, é possível comunicar com os vários modelos de linguagem da organização. No caso do nosso plugin utilizamos os modelos mais avançados atualmente, o GPT-3.5 ou o GPT-4o. O pedido enviado a esta API requer uma API key que pode ser criada através do site da OpenAI, é necessário que esta tenha plafond disponível para que a API responda aos pedidos. O pedido inclui também as mensagens anteriores, para que o modelo responda tendo em conta o contexto da conversa.

5.2.2 Drop Project [8]

Uma *web app open source* onde professores podem criar exercícios (*assignments*) com testes unitários associados, permitindo que os alunos possam submeter a sua solução do exercício e obter feedback sobre se esta cumpre os requisitos definidos pelo professor. É utilizada em três cadeiras (“Fundamentos de Programação”, “Algoritmia e Estruturas de Dados” e “Linguagens de Programação 2”) ao longo dos cursos de informática da Universidade Lusófona (LEI, LIG e LEIRT).

5.2.2.1 Drop Project API

A API do Drop Project permite submeter trabalhos e visualizar relatórios ou submissões, sem a necessidade de aceder ao website. Esta é utilizada pelo plugin para submeter trabalhos e obter as suas classificações.

5.2.3 IntelliJ IDEA

O IntelliJ IDEA é um ambiente de desenvolvimento produzido pela JetBrains, muito utilizado tanto em contexto académico como profissional. Os estudantes têm acesso à sua versão paga gratuitamente, mas esta não é necessária para utilizar o plugin.

5.2.3.1 Plugin Drop Project

O *plugin* desenvolvido como trabalho final de curso por um outro aluno da universidade no ano letivo 22/23, inicialmente desenvolvido para facilitar a submissão por parte dos alunos do seu trabalho para a plataforma Drop Project, permite que submetam o seu trabalho e observem a sua avaliação a partir do IntelliJ IDEA.

Nesta fase o plugin é atualizado de forma a integrar a funcionalidade de comunicação com o ChatGPT através da API da OpenAI, tendo de ser capaz de enviar o pedido à API, receber a resposta e apresentá-la no ecrã, bem como de garantir condições para recolha de *logs*. Dispõe de uma aba de chat em que o utilizador pode interagir com o modelo de linguagem, tal como no *browser*, mas sem sair do ambiente de desenvolvimento.

O facto de a comunicação estar integrada no *plugin* permite realizar recolha de dados, caso o utilizador aceite, valiosos para o estudo do modo como os alunos utilizam ferramentas de inteligência artificial.

5.3 Tecnologias e Ferramentas Utilizadas

5.3.1 Kotlin

Kotlin é uma linguagem de programação multiplataforma, orientada a objetos e funcional, desenvolvida pela JetBrains em 2011. Esta linguagem suporta o plugin. A sua compatibilidade com Java permite a utilização de Java Swing juntamente com bibliotecas de interface da JetBrains para desenvolver a interface do plugin.

5.3.2 Java Swing

Java Swing é uma biblioteca de interface gráfica (GUI) um conjunto de componentes para aplicações Java. Desenvolvido para ser um toolkit de GUI mais sofisticado e poderoso do que o anterior, Abstract Window Toolkit (AWT), o Swing oferece um conjunto rico de componentes, layouts e funcionalidades que permitem criar interfaces gráficas para aplicações Java ou Kotlin.

5.3.3 API da OpenAI e ChatGPT

A API da OpenAI oferece um meio para integrar modelos de linguagem avançados, como o ChatGPT, numa aplicação. A OpenAI disponibiliza *endpoints* que permitem enviar *prompts* e receber respostas geradas pelos modelos de linguagem.

No contexto deste projeto a API da OpenAI é utilizada para permitir a interação com o modelo do ChatGPT através do plugin Drop Project.

5.3.4 IntelliJ

O IntelliJ, um IDE desenvolvido pela JetBrains, é o ambiente de desenvolvimento onde vai correr o plugin, e também o ambiente utilizado para desenvolver o mesmo. O IntelliJ possui uma série de funcionalidades que o distinguem de outros ambientes de desenvolvimento que suportam Java/Kotlin, nomeadamente, o suporte de plugins, ferramentas que facilitam o uso de GitHub e uma interface intuitiva.

5.4 Implementação

Nesta seção, são descritos os requisitos necessários para a solução, incluindo recursos computacionais, de armazenamento e de rede.

5.4.1 Recursos Computacionais

O IntelliJ IDEA não é um ambiente de desenvolvimento muito pesado, ainda assim, recomenda-se a utilização de hardware competente para que o utilizador tenha uma boa experiência. Seguem-se os requisitos de hardware para correr o IntelliJ IDEA:

Sistema Operativo: Windows, macOS ou Linux.

Memória: Pelo menos 2GB de RAM livres, mas idealmente 4GB para melhor desempenho.

Espaço em Disco: Pelo menos 3.5GB de espaço livre.

Esses recursos garantem que os alunos possam utilizar o plugin de forma eficiente e sem interrupções significativas no ambiente de desenvolvimento.

5.4.2 Dependência da API OpenAI

A solução desenvolvida depende do acesso à API da OpenAI para gerar respostas usando o modelo ChatGPT. É necessário um acesso estável à internet para comunicação com a API.

5.5 Abrangência

A solução proposta abrange os conteúdos de várias disciplinas presentes no curso de licenciatura em engenharia informática, aplicando conceitos aprendidos em Linguagens de Programação 2, Algoritmia e Estruturas de Dados, e Computação Distribuída. Nestas cadeiras foram lecionados conceitos tal como programação orientada a objetos e utilização de *Web Services* que são utilizados no contexto deste projeto. A aplicação destes conhecimentos é essencial para o desenvolvimento do trabalho.

6 Recolha de Dados

Os *logs* de chat são guardados localmente em ficheiros com formato JSON, de forma a facilitar o seu processamento, numa pasta criada na diretoria do projeto em que se está a trabalhar, cada nova conversa é guardada num ficheiro separado.

Futuramente irá-se analisar se será melhor guardar os dados remotamente, possivelmente associando os mesmos respectivo exercício (*assignment*) e aluno que comunicou com o ChatGPT. Apenas serão recolhidos dados dos alunos que o autorizem.

Através dos dados recolhidos pretende-se estudar a forma como a ferramenta está a ser utilizada. Sendo um dos objetivos do estudo dos dados ensinar os alunos a tirar melhor proveito de ferramentas de inteligência artificial.

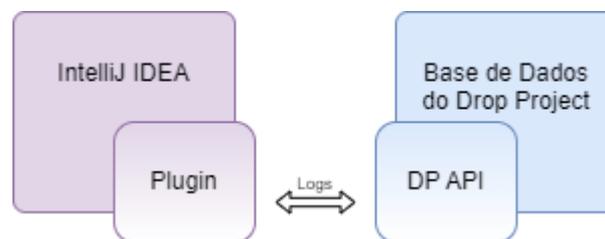


Figura 5 - Arquitetura para guardar dados remotamente

7 Testes e Validação

Para validar o bom funcionamento do plugin iremos distribuir uma versão do plugin pelos alunos que estão de momento inscritos em cadeiras que utilizam o Drop Project.

Com este objetivo em mente definimos alguns testes base para garantir que as funcionalidades base do plugin estão disponíveis e de seguida damos liberdade aos alunos para que usem o plugin da forma que lhes for mais conveniente na resolução das fichas das cadeiras que utilizam o Drop Project.

7.1 Testes Definidos

Nesta secção estão definidos os testes das funcionalidades base do plugin para garantir que está a funcionar corretamente, iremos pedir aos alunos que concluam os mesmos antes de prosseguirem com a utilização normal.

7.1.1 Teste de Instalação e Configuração

Com este teste pretende-se validar que os utilizadores são capazes de instalar o plugin e preencher os dados necessários para o seu funcionamento.

- Abrir o IntelliJ IDEA
- Instalar o plugin a partir do ficheiro zip disponibilizado
- Nas configurações do plugin preencher os dados pedidos

Resultado esperado: O utilizador instala e configura o plugin sem dificuldades.

7.1.2 Teste de Interação com o ChatGPT

Com este teste pretende-se validar se o plugin é capaz de comunicar corretamente com a API da OpenAI.

- Na aba do ChatGPT introduzir uma mensagem na caixa de texto de *input*
- Clicar no botão “*Send Message*”
- Verificar se o ChatGPT responde sem nenhum erro

Resultado esperado: O ChatGPT responde à mensagem enviada sem qualquer tipo de erro.

7.1.3 Teste de Envio de Texto do Editor para o ChatGPT

Com este teste pretende-se validar a funcionalidade de selecionar texto a partir do editor e enviar o mesmo para a caixa de texto da *prompt* para o ChatGPT.

- Selecionar código no editor
- A partir do menu de contexto selecionar a opção “Send to ChatGPT”
- Verificar que o texto selecionado está na caixa de texto da *prompt*

Resultado esperado: O texto selecionado é enviado corretamente para a caixa de texto da *prompt* a enviar ao ChatGPT.

7.1.4 Teste de Envio de Texto da Consola para o ChatGPT

Com este teste pretende-se validar a funcionalidade de selecionar texto a partir do editor e enviar o mesmo para a caixa de texto da *prompt* para o ChatGPT.

- Selecionar código na consola do IntelliJ
- A partir do menu de contexto selecionar a opção “Send to ChatGPT”
- Verificar que o texto selecionado está na caixa de texto da *prompt*

Resultado esperado: O texto selecionado é enviado corretamente para a caixa de texto da *prompt* a enviar ao ChatGPT.

7.1.5 Teste de Frases Prefixas à *Prompt*

Com este teste pretende-se validar o funcionamento da lista de frases possíveis de acrescentar rapidamente a qualquer *prompt*, para tal pedimos ao utilizador que adicione uma nova frase à lista e de seguida a utilize numa *prompt*.

- Aceder às configurações do plugin
- Na lista de frases a adicionar à *prompt* acrescentar uma frase
- Selecionar a frase adicionada na *dropdown* junto à caixa da *prompt*
- Escrever qualquer mensagem na caixa de texto da *prompt*
- Enviar a mensagem
- Verificar se a frase foi adicionada à mensagem

Resultado esperado: A frase adicionada à lista é corretamente concatenada com a mensagem escrita no campo de *input*.

7.2 Testes de Utilização

Para validar o bom funcionamento do plugin este passou por três fases de teste, a grande parte dos bugs reportados (mais detalhes na tabela) em cada fase foram corrigidos antes do início da fase seguinte, ainda assim, algum *feedback* foi recebido entre as fases de teste.

Inicialmente o plugin foi testado pelos alunos que participaram no *Focus Group*. As duas fases de teste seguintes deram-se principalmente nas aulas da cadeira de Algoritmia e Estruturas de Dados, nas quais os alunos foram incentivados a utilizar a ferramenta na resolução das fichas, que são avaliadas através do Drop Project.

Ao longo das fases de teste foi recolhido o *feedback* dos alunos, tanto os bugs reportados como sugestões de melhoria ou de novas funcionalidades.

Seguem-se os bugs reportados ao longo das três fases de teste.

Tabela 4 - Bugs Reportados/Feedback

Descrição	Reportado por	Data	Estado
Tamanho da caixa do chat demasiado pequena	Bruno Rodrigues	14/06/2024	Resolvido
Erro ao adicionar assignment por ID	Cátia Coelho	27/05/2024	Resolvido
ZIP gerado excede tamanho máximo para submeter para o Drop Project	Cátia Coelho	03/06/2024	Resolvido
O botão copy code causa um <i>crash</i> quando não há código na resposta	Bruno Cipriano (Orientador)	15/05/2024	Resolvido
Sugestão de funcionalidade - Recomeçar conversa	Bruno Cipriano (Orientador)	15/05/2024	Resolvido
Quando se envia uma mensagem a barra de scroll do chat volta ao topo	Bruno Cipriano (Orientador)	15/05/2024	Resolvido
Quando se clica mais do que uma vez no botão "Send" aparecem várias respostas à mensagem	Bruno Rodrigues	29/04/2024	Resolvido
É difícil distinguir as nossas mensagens das mensagens do ChatGPT	Bruno Rodrigues	29/04/2024	Resolvido
Não há mensagem de erro quando não há tokens disponíveis	Sebastião Coelho	24/03/2024	Resolvido

Caso haja uma linha longa sem espaços pode sair da janela de chat	Miguel Silva (Aluno)	05/07/2024	Por Resolver
---	-------------------------	------------	--------------

8 Calendário

Nesta seção vamos apresentar a calendarização das diferentes fases e tarefas do projeto.

Fase	Data de início	Data de fim
Componente Gráfica - Análise Inicial	29 / 09 / 2023	02 / 10 / 2023
Componente Gráfica - Implementação	02 / 10 / 2023	15 / 10 / 2023
Componente de integração com o GPT	29 / 09 / 2023	30 / 09 / 2023
Integrar protótipos <i>Backend</i> e <i>Frontend</i>	15 / 10 / 2023	03 / 11 / 2023
Modelação da Base de Dados relativa ao <i>LOG</i>	03 / 11 / 2023	10 / 11 / 2023
Relatório Parte 1	29 / 09 / 2023	19 / 11 / 2023
<i>Focus Group</i> - Realização	20 / 11 / 2023	20 / 11 / 2023
<i>Focus Group</i> - Análise de Resultados	03 / 12 / 2023	05 / 12 / 2023
Definição de Requisitos	03 / 12 / 2023	10 / 12 / 2023
Implementação dos Requisitos Definidos	10 / 12 / 2023	10 / 04 / 2024
1º Fase de Testes do Plugin com alunos do <i>Focus Group</i>	22 / 02 / 2024	20 / 03 / 2024
Correção de bugs reportados pelos alunos do <i>Focus Group</i>	20 / 03 / 2024	15 / 04 / 2024
2º Fase de Testes do Plugin com alunos do <i>Focus Group</i>	15 / 04 / 2024	21 / 04 / 2024
Correção de bugs reportados pelos alunos do <i>Focus Group</i>	24 / 04 / 2024	17 / 05 / 2024
Testes do Plugin em Aula	20 / 05 / 2024	09 / 06 / 2024
Correções de bugs reportados pelos alunos em aula	09 / 06 / 2024	20 / 06 / 2024
Oficialização do Plugin no repositório GitHub do DP	Após entrega do TFC (será realizado pelos professores orientadores)	
Publicação do Plugin no Marketplace do IntelliJ	Após entrega do TFC (será realizado pelos professores orientadores)	

Tabela 4 - Tabela de Calendarização

9 Resultados

Ao longo do desenvolvimento do trabalho, foram implementados todos os requisitos inicialmente definidos como obrigatórios, foram também acrescentados novos requisitos através de sugestões feitas por professores e alunos (detalhes na secção [Requisitos provenientes do Focus Group](#)).

Requisitos implementados na solução desenvolvida:

- Requisito 1 - Janela de conversa com o ChatGPT, com uma caixa de texto, apresenta uma interface semelhante à do site.
- Requisitos 2A, 2B e 9 - “Send to ChatGPT” – permite enviar texto/código a partir do editor, da consola ou de um erro do relatório de *assignment* diretamente para a caixa de texto da conversa.
- Requisito 3 – A interface tem um funcionamento semelhante ao do site oficial, facilitando a sua utilização por parte de utilizadores que já estão familiarizados com o site.
- Requisito 4A - Funcionalidade de adicionar um sufixo de uma lista a qualquer *prompt*, a lista de sufixos pode ser editada nas definições do plugin.
- Requisito 5 – Os *logs* são guardados localmente, mas ainda não foi implementada a funcionalidade de continuar uma conversa.
- Requisito 7 – É possível definir a API Key a ser utilizada através das definições do plugin.
- Requisito 10 - Opção de recomeçar a conversa, limpando o contexto enviado ao LLM.
- Requisito 11 - Botão “Copy Code” permite copiar o código de uma resposta do ChatGPT.

Os requisitos 4B, 5, 6 e 8 não foram totalmente implementados. Estes requisitos estão relacionados com os *logs*, sendo que inicialmente o objetivo era enviar os mesmos diretamente para a base de dados do próprio Drop Project. Para tal, seria necessário realizar atualizações na base de dados. Estas modificações acabaram por não ser feitas e então os *logs* são guardados apenas localmente. No entanto, caso se pretenda fazer esta adaptação, não serão precisas grandes alterações no plugin.

10 Conclusão

O objetivo principal deste trabalho foi desenvolver uma solução que permite melhorar a experiência de utilização do ChatGPT, por parte dos alunos, na resolução de fichas ou desenvolvimento de projetos de cadeiras de programação dos cursos de informática da Universidade Lusófona. Desse modo permitindo também a recolha de *logs*, dos alunos que assim autorizem, com o objetivo de estudar a forma como os alunos utilizam inteligência artificial para que os professores os possam guiar no melhor caminho. Para tal foi atualizado o plugin do Drop Project já existente, que por sua vez também estava envolvido nas fichas e projetos, com o objetivo de facilitar a submissão dos mesmos para a plataforma de avaliação automática utilizada, o Drop Project.

A solução desenvolvida, não estando completamente livre de bugs, atendeu às expectativas, cumprindo todos os requisitos definidos como obrigatórios e implementando também sugestões dadas por alunos e professores.

O desenvolvimento deste projeto foi um pouco desafiante ao início, envolveu bastante trabalho de pesquisa devido a novas ferramentas pouco conhecidas, sendo que o desenvolvimento do plugin envolve as bibliotecas disponibilizadas pelos desenvolvedores do IDE, e, devido à necessidade de aplicação de alguns conceitos de programação assíncrona, pouco trabalhados antes do 3º ano do curso de Licenciatura em Engenharia Informática.

11 Bibliografia

- [1] Feldt, R., de Oliveira Neto, F. G., & Torkar, R. (2018, May). Ways of applying artificial intelligence in software engineering. In Proceedings of the 6th International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (pp. 35-41). <https://arxiv.org/pdf/2308.06921.pdf>
- [2] Prather, J., Reeves, B. N., Denny, P., Becker, B. A., Leinonen, J., Luxton-Reilly, A., ... & Santos, E. A. (2023). " It's Weird That it Knows What I Want": Usability and Interactions with Copilot for Novice Programmers. arXiv preprint arXiv:2304.02491. <https://arxiv.org/pdf/2304.02491.pdf>
- [3] Destefanis, G., Bartolucci, S., & Ortu, M. (2023). A Preliminary Analysis on the Code Generation Capabilities of GPT-3.5 and Bard AI Models for Java Functions. arXiv preprint arXiv:2305.09402. <https://arxiv.org/pdf/2305.09402.pdf>
- [4] Liffiton, M., Sheese, B., Savelka, J., & Denny, P. (2023). Codehelp: Using large language models with guardrails for scalable support in programming classes. arXiv preprint arXiv:2308.06921. <https://arxiv.org/pdf/2308.06921.pdf>
- [5] Lau, S., & Guo, P. (2023, August). From " Ban it till we understand it" to " Resistance is futile": How university programming instructors plan to adapt as more students use AI code generation and explanation tools such as ChatGPT and GitHub Copilot. In Proceedings of the 2023 ACM Conference on International Computing Education Research-Volume 1 (pp. 106-121). <https://dl.acm.org/doi/pdf/10.1145/3568813.3600138>
- [6] Alves, P., & Cipriano, B. P. (2023). The centaur programmer--How Kasparov's Advanced Chess spans over to the software development of the future. arXiv preprint arXiv:2304.11172. <https://arxiv.org/pdf/2304.11172.pdf>
- [7] Meyer, A. N., Fritz, T., Murphy, G. C., & Zimmermann, T. (2014, November). Software developers' perceptions of productivity. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering (pp. 19-29). <https://www.zora.uzh.ch/id/eprint/98324/1/productivity.pdf>
- [8] Cipriano, B. P., Fachada, N., & Alves, P. (2022). Drop Project: An automatic assessment tool for programming assignments. SoftwareX, 18, 101079. <https://doi.org/10.1016/j.softx.2022.101079>
- [9] Prasad, S., Greenman, B., Nelson, T., & Krishnamurthi, S. (2023, December). Generating Programs Trivially: Student Use of Large Language Models. In Proceedings of the ACM Conference on Global Computing Education Vol 1 (pp. 126-132). <https://dl.acm.org/doi/abs/10.1145/3576882.3617921>

Glossário

LEI	Licenciatura em Engenharia Informática
LIG	Licenciatura em Informática de Gestão
TFC	Trabalho Final de Curso
LLM	<i>Large Language Model</i>
IA	Inteligência Artificial
DP	Drop Project
API	<i>Application Programming Interface</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
GUI	<i>Graphical User Interface</i>
AWT	<i>Abstract Window Toolkit</i>