



UNIVERSIDADE  
**LUSÓFONA**

## **Melhorar o Drop Project Student Plugin + GPT**

### **Trabalho Final de curso**

Relatório Intercalar 1º Semestre

**Afonso Castelhana, a22303665, Engenharia Informática**

**Orientador:** Bruno Cipriano

**Co-orientador:** Rodrigo Correia

Departamento de Engenharia Informática da Universidade Lusófona

Centro Universitário de Lisboa

14/11/2025

[www.ulusofona.pt](http://www.ulusofona.pt)

## **Direitos de cópia**

*(Melhorar o Drop Project Student Plugin + GPT)*, Copyright de (Afonso Castelhana), ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) 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.

# **Agradecimentos**

Gostaria de expressar o meu agradecimento aos meus orientadores, o Professor Bruno Cipriano e o Professor Rodrigo Correia. A sua orientação foi fundamental para o desenvolvimento deste projeto, não só pela partilha de conhecimento técnico essencial sobre o ecossistema do Drop Project e do IntelliJ SDK, mas também pela disponibilidade constante e pelo incentivo crítico que permitiu fazer este trabalho.

## Resumo

Este TFC tem como objetivo o desenvolvimento e melhoria de um plugin adicional para o **Drop Project**, destinado a integrar capacidades de assistência inteligente através de modelos de IA conversacional.

O plugin original permitia enviar para o ChatGPT o feedback gerado pelo Drop Project e receber respostas orientadas à correção de código, incluindo a disponibilização de um botão para copiar o código sugerido pela IA. Contudo, devido a alterações nas chaves e mecanismos de autenticação da API da OpenAI, o plugin encontra-se atualmente inoperacional.

No âmbito deste trabalho, propõe-se:

1. **Restabelecer o funcionamento do plugin**, atualizando a comunicação com a API da OpenAI e garantindo o correto envio e receção de mensagens;
2. **Implementar um conjunto de melhorias de usabilidade**, incluindo a reorganização das configurações do plugin e outras funcionalidades que tornem a interação mais simples e intuitiva;
3. **Melhorar a experiência de utilização durante a interação com respostas geradas pelos modelos de IA**, incluindo mecanismos mais claros e eficientes para lidar com diferentes tipos de conteúdo apresentados;
4. **Expandir o plugin** para suportar outros modelos de IA, possibilitando a comunicação com diferentes provedores para além do ChatGPT.

Com estas melhorias, o projeto visa tornar o plugin mais flexível, robusto e útil para estudantes e docentes, integrando a assistência por IA de forma mais eficiente no processo de desenvolvimento e correção automática de código.

### Palavras chave:

Drop Project; ChatGPT; API Integration

# Abstract

This project focuses on the development and improvement of an additional plugin for the **Drop Project**, aimed at integrating intelligent assistance through conversational AI models. The original plugin allowed users to send the feedback generated by the Drop Project to ChatGPT and receive code-oriented guidance, including a button to copy the code suggested by the AI. However, due to changes in the OpenAI API keys and authentication mechanisms, the plugin is currently non-functional.

The work proposed in this project includes:

1. **Restoring the plugin's functionality** by updating the communication with the OpenAI API and ensuring proper message exchange;
2. **Implement a set of usability improvements**, including reorganizing plugin settings and other functionalities that make interaction simpler and more intuitive;
3. **Improving the user experience during interaction with responses generated by AI models**, including clearer and more efficient mechanisms for handling different types of presented content;
4. **Extending the plugin to support additional AI models**, allowing communication with providers other than ChatGPT.

With these enhancements, the project aims to make the plugin more flexible, robust, and useful for students and instructors, integrating AI-based assistance more effectively into the process of developing and automatically evaluating programming assignments.

## Key-words:

Drop Project; ChatGPT; API Integration

# Índice

Agradecimentos .....	3
Resumo.....	4
Abstract .....	5
Índice .....	6
<b>1. Introdução .....</b>	<b>8</b>
<b>1.1. Enquadramento .....</b>	<b>8</b>
O que é este projeto? .....	8
O que é o Drop Project?.....	9
Qual é a utilidade dos plugins no ambiente de ensino? .....	9
Qual é o papel das IA conversacionais no apoio ao ensino/aprendizagem? .....	9
Como é que este projeto se insere no panorama de ferramentas educativas inteligentes? .....	9
<b>1.2. Motivação e Identificação do Problema.....</b>	<b>9</b>
O que motivou a criação deste TFC? .....	9
Porque é que isto é um problema para alunos e professores? .....	10
O que motivou a proposta? .....	10
<b>1.3. Objetivos.....</b>	<b>11</b>
Objetivo geral.....	11
Objetivos específicos.....	11
<b>1.4. Estrutura do Documento.....</b>	<b>11</b>
<b>2. Pertinência e Viabilidade.....</b>	<b>13</b>
<b>2.1. Pertinência.....</b>	<b>13</b>
Onde será usado .....	13
Por quem.....	13
Cenários de uso .....	13
<b>2.2. Viabilidade .....</b>	<b>13</b>
<b>2.3. Análise Comparativa.....</b>	<b>13</b>
<b>2.4. Proposta de inovação e mais valias .....</b>	<b>14</b>
<b>2.5. Identificação de oportunidade de negócio .....</b>	<b>15</b>
<b>3. Especificação e Modelação .....</b>	<b>15</b>

3.1.	Análise de requisitos .....	15
3.2.	Modelação .....	16
<b>4.</b>	<b>Implementação .....</b>	<b>19</b>
4.1.	Arquitetura do Plugin .....	19
4.2.	Detalhes de Implementação.....	19
4.2.1	Processamento de Respostas GenAI.....	19
4.2.2	Sistema de Cópia de Código .....	20
4.2.3	Configuração Dinâmica .....	21
<b>5.</b>	<b>Solução Proposta .....</b>	<b>21</b>
5.1.	Apresentação.....	21
5.2.	Tecnologias e Ferramentas.....	22
5.3.	Ambientes de Teste e de Produção .....	22
5.4.	Abrangência.....	22
5.5.	Componentes do sistema.....	23
5.6.	Interfaces.....	24
<b>6.</b>	<b>Testes e validação .....</b>	<b>24</b>
6.1.	Metodologia e Testes .....	24
6.2.	Casos de Teste e Resultados .....	24
6.3.	Análise de Resultados e Usabilidade .....	25
6.4.	Demonstração Visual .....	25
<b>7.</b>	<b>Método e Planeamento.....</b>	<b>26</b>
7.1.	Introdução .....	26
	Métodos técnicos.....	26
	Abordagem usada .....	26
7.2.	Planeamento Inicial.....	27
7.3.	Análise Crítica ao Planeamento.....	27
<b>8.</b>	<b>Planeamento .....</b>	<b>28</b>
	Plano de Trabalho .....	28
	Tarefas .....	28
	Riscos .....	29
	Mitigações.....	29
	Progresso já realizado .....	29

<b>9.</b>	<b>Conclusões e Trabalho Futuro</b> .....	29
9.1.	Conclusões atuais .....	29
9.2.	Trabalho Futuro.....	30
<b>10.</b>	<b>Elementos Técnicos</b> .....	30
	Modelo Conceptual.....	30
	Use Cases .....	30
	Planeamento.....	30
<b>11.</b>	<b>Github</b> .....	31
<b>12.</b>	<b>Inquérito sobre o uso de IA</b> .....	31
12.1.	Inquério Forms .....	32
12.2.	Inquérito em ficheiro Word .....	35
<b>13.</b>	<b>Bibliografia</b> .....	36
<b>14.</b>	<b>Glossário</b> .....	37

# 1.Introdução

## 1.1. Enquadramento

**O que é este projeto?** – Este projeto utiliza o plugin do Drop project como base e adiciona uma janela de conversa com o ChatGpt para se

poder enviar o feedback do Drop Project diretamente sem ter que se copiar.

**O que é o Drop Project?** – O Drop Project é um site de testes automáticos à base de submissão do código fonte do utilizador. Testando quer seja funções ou classes inteiras com argumentos pré-definidos dando um feedback, não do teste em si, mas da resposta esperada pelo código e a resposta recebida, dizendo se são iguais ou não.

O Drop project também tem um plugin que pode ser utilizado no intellij em que em vez de se enviar a pasta do src dá-se diretamente login no plugin e apenas clica-se no botão de submissão e recebe-se o mesmo feedback.

**Qual é a utilidade dos plugins no ambiente de ensino?** – Plugins são uma forma mais rápida e eficaz de se enviar e receber de volta a informação que se pretende, em vez de mexer em pastas ou ter que copiar e colar informação, clica-se apenas em poucos botões e não tem que se sair do ambiente de programação.

**Qual é o papel das IA conversacionais no apoio ao ensino/aprendizagem?** – IA é bastante útil no apoio à aprendizagem, podendo indicar o que foi que se fez de errado e dar dicas para o que se deve fazer para corrigir os ditos erros e também ajuda com boas práticas, mesmo que não seja para programação.

**Como é que este projeto se insere no panorama de ferramentas educativas inteligentes?** – Este projeto inclui não só a componente de feedback e testes automáticos do Drop Project, como adiciona a componente de um LLM para resolver os erros dados no feedback apoiando o aluno na sua aprendizagem em linguagens de programação.

## **1.2. Motivação e Identificação do Problema**

**O que motivou a criação deste TFC?** – Apesar deste plugin ser ótimo por si mesmo, tem algumas funções em falta que dariam bastante

jeito ao utilizador, o que causou a criação deste TFC. As alterações mais significativas a serem feitas são a habilidade de alternar entre os LLMs de preferência e poder selecionar qual excerto de código dado pelo LLM se quer copiar.

**Porque é que isto é um problema para alunos e professores? - A**

indisponibilidade do plugin representa um problema significativo tanto para alunos como para professores.

Sem esta integração com modelos de IA, os estudantes deixam de receber feedback inteligente e orientado à correção do código, perdendo uma ferramenta que complementa o feedback automático do Drop Project com explicações detalhadas e sugestões de melhoria.

Esta limitação aumenta o esforço necessário por parte dos alunos, que têm de recorrer manualmente a plataformas externas como o ChatGPT, copiando e colando feedback e código entre diferentes ambientes de trabalho.

Para os docentes, a ausência de uma ferramenta integrada reduz a eficácia do apoio dado aos estudantes, dificulta a uniformização das explicações e compromete a utilização de tecnologias de IA como complemento ao ensino de programação.

**O que motivou a proposta? - A** proposta deste projeto surge da necessidade de aumentar a utilidade do plugin e melhorar a sua robustez.

Além de restaurar a integração com a API da OpenAI, pretende-se criar uma solução mais estável e flexível, eliminando dependências rígidas e problemas recorrentes de autenticação.

Outro fator motivador é a possibilidade de suportar diferentes modelos de IA, permitindo que alunos e docentes utilizem alternativas ao ChatGPT e garantindo maior adaptabilidade e longevidade à ferramenta.

### **1.3. Objetivos**

**Objetivo geral** – O objetivo geral deste projeto consiste em aperfeiçoar e expandir o plugin do Drop Project, reforçando a sua integração com sistemas de inteligência artificial conversacional.

Pretende-se disponibilizar uma ferramenta mais completa, estável e funcional, capaz de apoiar o processo de aprendizagem da programação através de feedback inteligente e contextualizado, diretamente acessível na plataforma utilizada pelos estudantes.

**Objetivos específicos**- Para concretizar este objetivo geral, definem-se os seguintes objetivos específicos:

1. Restabelecer o funcionamento do plugin, atualizando os mecanismos de comunicação com a API da OpenAI e garantindo um processo fiável de autenticação, envio e receção de mensagens.
2. Desenvolver uma nova secção de configurações, designada OpenAI Settings, que permita ao utilizador gerir credenciais, preferências e outros parâmetros associados aos modelos de IA utilizados.
3. Melhorar a funcionalidade de cópia de código, permitindo que, sempre que a IA devolva múltiplos blocos de código, o utilizador possa seleccionar de forma clara qual deles pretende copiar.
4. Expandir a compatibilidade do plugin, adicionando suporte para comunicação com diferentes modelos e fornecedores de IA, reduzindo a dependência exclusiva do ChatGPT e aumentando a flexibilidade e escalabilidade da solução.

### **1.4. Estrutura do Documento**

Este documento está organizado em vários capítulos que refletem as diferentes etapas de pesquisa, análise e planeamento do Trabalho Final de Curso.

**No Capítulo 1** é apresentado o enquadramento do projeto, a motivação, a identificação do problema, os objetivos definidos e a estrutura geral do relatório.

**O Capítulo 2** aprofunda a pertinência e viabilidade do trabalho, incluindo a análise comparativa com soluções existentes, a proposta de inovação e a identificação de potenciais mais-valias.

**O Capítulo 3** descreve a especificação do sistema, englobando a análise de requisitos, casos de uso, modelação e protótipos de interface.

**O Capítulo 4** demonstra como as maiores implementações do projeto foram introduzidas aprofundando mais acerca dos métodos usados.

**O Capítulo 5** apresenta a solução proposta, abordando a arquitetura, tecnologias, componentes e interfaces previstas.

**O Capítulo 6** encontra-se reservado para os testes e validação, que serão desenvolvidos em fases posteriores do projeto.

**O Capítulo 7** descreve o método e o planeamento da execução do trabalho.

**O Capítulo 8** será dedicado à apresentação dos resultados obtidos.

**O Capítulo 9** apresenta as conclusões atuais e possíveis avanços futuros.

Por fim, são incluídas as referências bibliográficas e anexos relevantes.

## 2. Pertinência e Viabilidade

### 2.1. Pertinência

**Onde será usado** – O plugin será utilizado em ambientes acadêmicos, tanto em aula como em estudo autônomo, para apoiar a realização dos laboratórios semanais das unidades curriculares que recorrem ao Drop Project.

**Por quem** – Embora possa ser utilizado por qualquer utilizador do Drop Project, o plugin é especialmente relevante para estudantes de cadeiras como Algoritmos e Estruturas de Dados e Linguagens de Programação II, que fazem uso intensivo da plataforma.

**Cenários de uso** – O plugin será mais frequentemente utilizado durante as aulas práticas e sessões de laboratório, para acelerar a experimentação e depuração de código, assim como em casa, como ferramenta complementar de estudo e prática de programação.

### 2.2. Viabilidade

O projeto é tecnicamente viável porque se baseia num plugin já existente e utiliza tecnologias estáveis e bem documentadas, como Java/Kotlin, Gradle e o IntelliJ SDK. As melhorias previstas — atualização da comunicação com a API da OpenAI, reorganização da arquitetura e criação de novas funcionalidades — são compatíveis com o nível esperado para um TFC e não exigem reconstrução completa da solução.

Além disso, os recursos necessários são mínimos: acesso ao IntelliJ IDEA, internet e uma chave de API. O tempo disponível também é suficiente para cumprir todas as fases do desenvolvimento, desde a análise até aos testes e relatório final, tornando o projeto totalmente exequível dentro do calendário definido.

### 2.3. Análise Comparativa

Atualmente, existem várias ferramentas que integram modelos de IA em ambientes de programação, como extensões do Visual Studio

Code (ex.: GitHub Copilot, Codeium) e plugins independentes que enviam código para modelos de IA. No entanto, nenhum destes sistemas se integra diretamente com o Drop Project, nem oferece uma ponte entre o feedback automático da plataforma e a análise contextualizada de um LLM.

O plugin original do Drop Project fornecia uma solução parcial, mas encontra-se desatualizado e limitado ao ChatGPT. Em comparação, ferramentas externas exigem copiar e colar feedback entre aplicações, o que aumenta o atrito na experiência do utilizador. Deste modo, o presente projeto distingue-se por:

- integrar IA diretamente no fluxo de submissão do Drop Project
- permitir a escolha de múltiplos modelos de IA, não apenas o ChatGPT
- melhorar a interação com respostas contendo múltiplos blocos de código
- eliminar dependências manuais e reduzir o tempo entre submissão, feedback e correção.

Assim, o plugin evoluído representa uma solução mais contextual, integrada e eficiente face às ferramentas alternativas.

#### **2.4. Proposta de inovação e mais valias**

A inovação principal do projeto reside na integração inteligente entre feedback automático e assistência generativa, criando um ciclo contínuo de análise, correção e aprendizagem. As mais-valias incluem:

- Suporte a múltiplos modelos de IA, aumentando a liberdade de escolha e reduzindo a dependência de um único fornecedor.
- Arquitetura modular baseada em providers, que facilita a adição futura de novos modelos.
- Melhoria significativa da usabilidade, incluindo a seleção de blocos de código gerados pela IA e uma interface de configurações mais clara.

- Aumento do desempenho pedagógico, ao permitir que os alunos compreendam melhor os erros e explorem soluções alternativas diretamente no ambiente de desenvolvimento.

Estas características tornam o plugin mais robusto, mais escalável e mais útil no contexto do ensino da programação.

## 2.5. Identificação de oportunidade de negócio

Embora o projeto tenha origem acadêmica, existe potencial para evoluir para um produto comercial, dado que muitas instituições procuram integrar inteligência artificial em sistemas de avaliação automática. Um plugin que combine feedback automático com análise gerada por modelos de IA pode ser interessante para universidades, bootcamps ou plataformas de ensino online.

A arquitetura modular permite também expandir a solução para outros IDE, como VSCode ou Eclipse, abrindo portas a versões premium ou serviços SaaS dedicados ao apoio à aprendizagem de programação.

# 3. Especificação e Modelação

## 3.1. Análise de requisitos

A definição de requisitos é essencial para orientar o desenvolvimento do plugin e garantir que a solução responde de forma adequada às necessidades identificadas no contexto de utilização real.

Uma parte dos objetivos do projeto é realizar melhorias de usabilidade. Algumas dessas melhorias já foram identificadas e transformadas em requisitos, mas este trabalho ainda está em progresso.

A tabela abaixo apresenta os requisitos funcionais já identificados.

Código	Categoria	Descrição	Importância	Sistema
RF1	Usabilidade	Deve existir um ecrã dedicado para Configurações da API, onde se possa definir: Provider, Modelo, API Key.	Obrigatório	Plugin

RF2	Usabilidade	Quando a resposta tem várias porções de código, permitir indicar qual se pretende copiar.	Obrigatório	Plugin
RF3	Generalização	Converter usos de “GPT” para “GenAI”	Obrigatório	Plugin
RF4	Usabilidade	Poder utilizar o plugin em várias línguas (Português e Inglês).	Opcional	Plugin
		Deve ser possível configurar um assignment do Drop Project como permitindo o uso do plugin.		
RF5	Configuração	O plugin deve respeitar essa definição. Para os assignments Drop Project que permitem uso do plugin, deve ser possível indicar se as respostas que contenham código devem ser filtradas de alguma forma (p.e. remover o código, substituir o código por pseudo-código).	Obrigatório	DP e Plugin
RF6	Configuração	O plugin deve respeitar essa definição. Para os assignments Drop Project que permitem uso do plugin, deve ser possível indicar se existe um máximo de vezes que se pode pedir ajuda ao LLM/GenAI.	Opcional	DP e Plugin
RF7	Configuração	O plugin deve respeitar essa definição.	Opcional	DP e Plugin

### 3.2. Modelação

A modelação envolve a criação de artefactos que descrevem o comportamento e a estrutura do sistema. Nesta fase, incluem-se:

Use Case Diagram (figura 2) que representa as interações entre o utilizador e o plugin;

Diagrama de classes (Figura 1) com entidades como Controlador do Plugin, IProvider, OpenAIProvider e Painel de Chat;

Modelo conceptual que descreve as principais entidades e relações;

Fluxo de comunicação entre o plugin e a API da OpenAI, ilustrando os passos de envio e recepção de mensagens.

Estes modelos fornecem uma visão clara da solução e servem de base para a implementação.

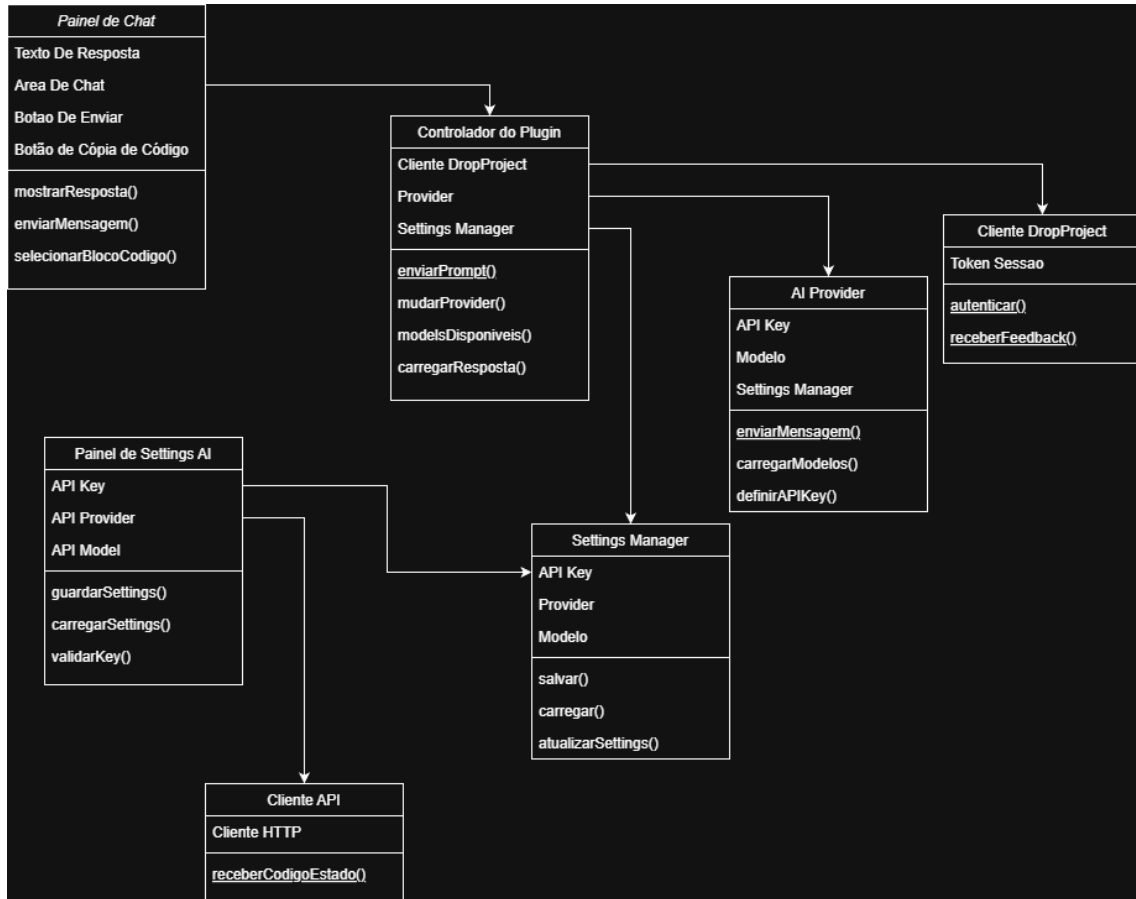


Figura 1 - Modelo Conceptual de classes

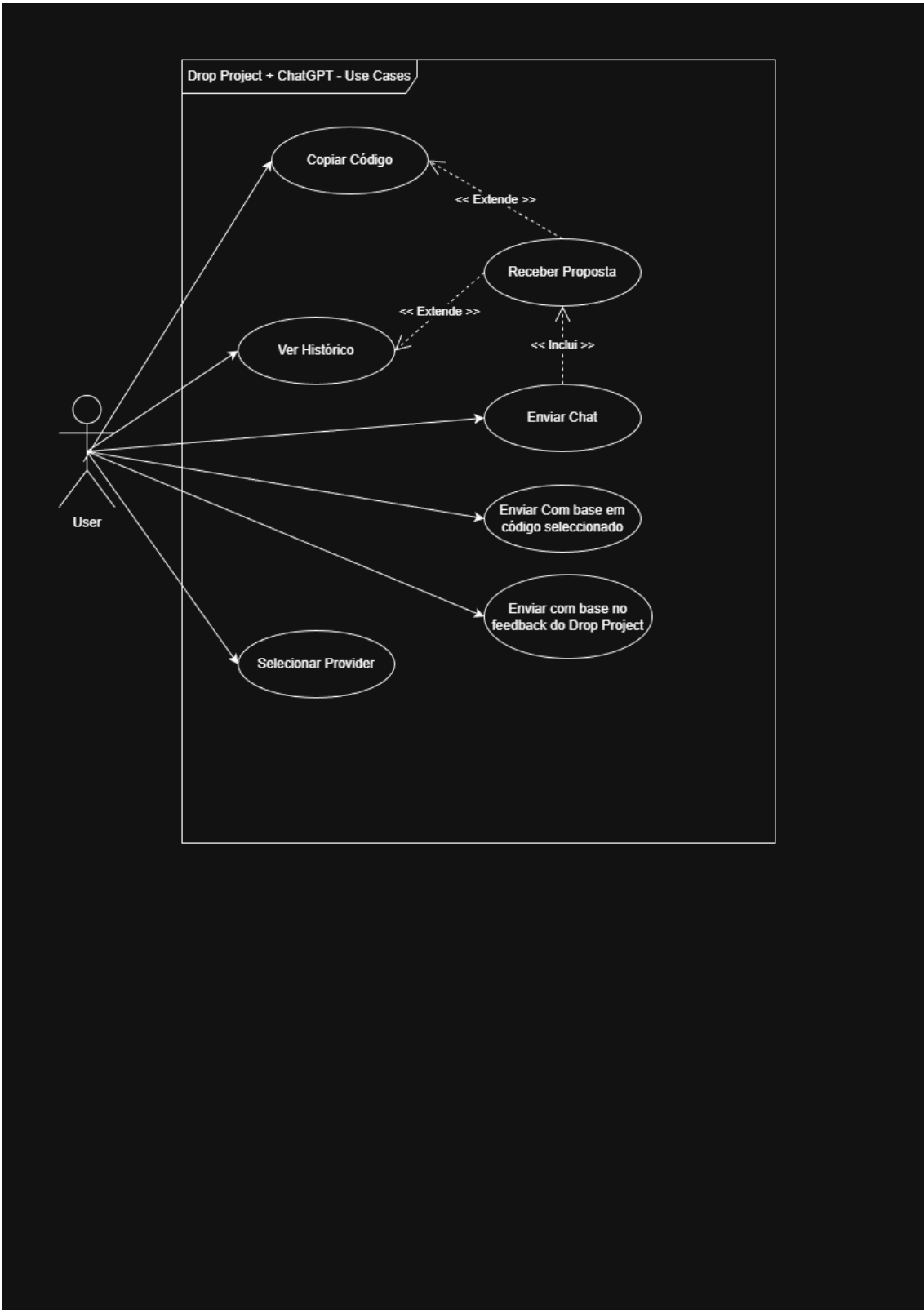


Figura 2 - Use Cases de interação com o utilizador

# 4. Implementação

## 4.1. Arquitetura do Plugin

A arquitetura do Plugin do Drop Project foi expandida para suportar a integração de inteligência Artificial Generativa. A estrutura baseia-se no framework do IntelliJ SDK, utilizando Kotlin como linguagem principal. A interface é construída sobre a biblioteca Swing, utilizando um modelo de ToolWindow.

Um ponto central da arquitetura é a separação entre a visualização das mensagens e a lógica de processamento de texto. Enquanto a classe `UIGpt` gere o layout e os eventos de utilizador, a classe `ChatHtmlBuilder` isola toda a responsabilidade de formatação visual, permitindo uma manutenção mais simples do estilo CSS e da renderização de componentes dinâmicos, como os botões de cópia de código.

A figura 3 demonstra como é atualmente dividida cada classe pela sua função no programa em si.

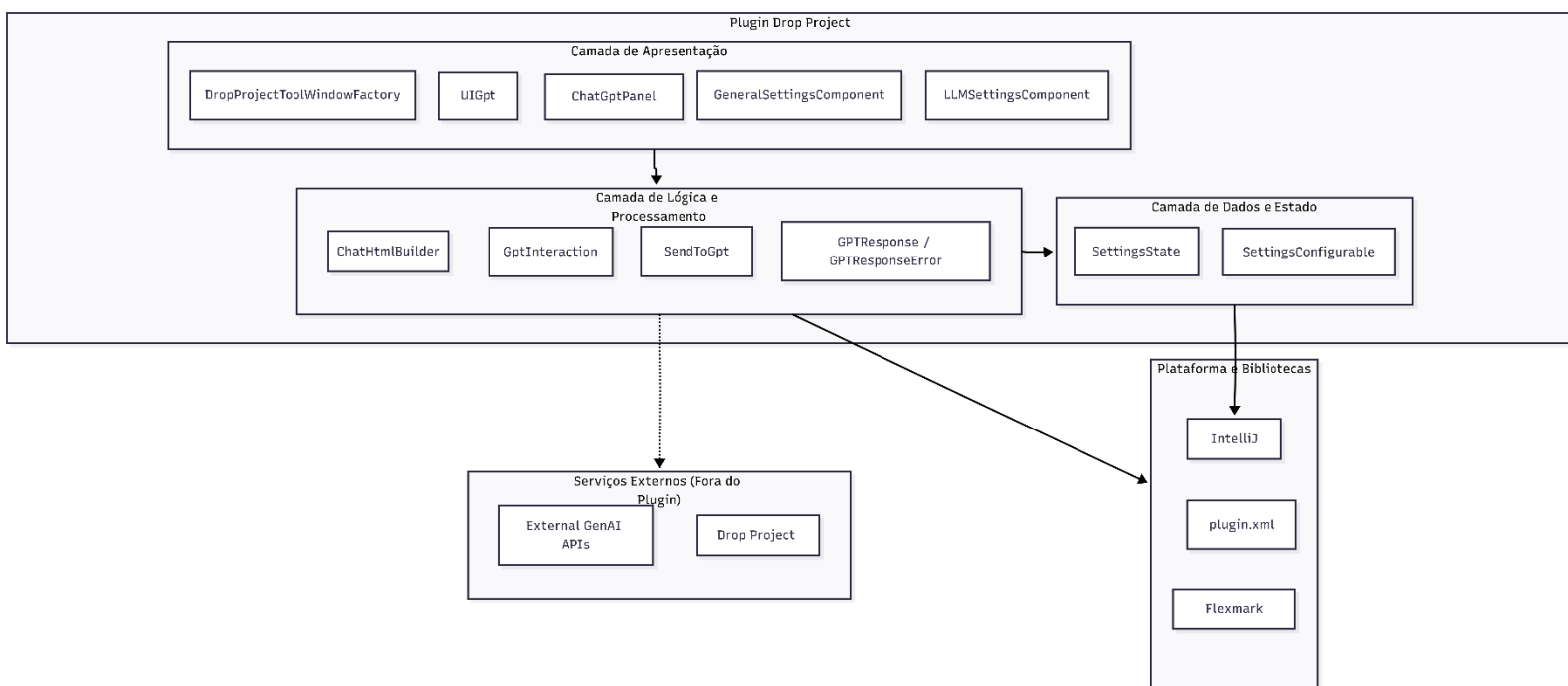


Figura 3- Diagrama de Camadas da arquitetura real do projeto

## 4.2. Detalhes de Implementação

### 4.2.1 Processamento de Respostas GenAI

Uma componente que o projeto já integrava antes da minha participação foi a maneira como se processava o output gerado pela IA.

A visualização das respostas geradas pelos modelos de linguagem é um dos componentes críticos para o plugin ser utilizável.

Para garantir que o feedback da IA seja apresentado de forma estruturada é necessário uma conversão de texto do formato Markdown para HTML.

#### Porquê o uso do flexmark?

Para processar as respostas da API, o projeto inicial integrava a biblioteca Flexmark-java, que, simplificando, permite converter o texto bruto para HTML.

Isto é necessário para se poder formatar o texto de uma maneira que se torne funcional dentro da chatbox do plugin.

Para assegurar a legibilidade e a integridade estética com a IDE, está implementado um estilo CSS que define a tipografia do texto, a adaptação dos temas e finalmente as margens de texto dentro do chat.

### **4.2.2 Sistema de Cópia de Código**

A funcionalidade de cópia de código foi redesenhada ao longo do projeto.

Inicialmente foi alterada para não desaparecer quando se dá scroll ao longo do texto, posicionando-se o botão debaixo da input box para se manter sempre visível. Mas isto tinha um erro, apenas copiando o código da última mensagem enviada com código pela IA, causando um problema de usabilidade no botão.

Então o botão foi redesenhado uma segunda vez para permitir que o utilizador selecione de forma clara qual o bloco de código é que se pretende copiar, especialmente quando a IA devolve mais do que um bloco de código de uma vez.

#### Como é que se alcançou este objetivo?

Na classe *'ChatHtmlBuilder.kt'* o HTML gerado pela IA é processado por uma Regex configurada para detetar o padrão `'<pre><code.*?>'`. Este padrão é o que isola excertos de código do resto do texto gerado.

Após isso, sempre que um bloco é detetado o sistema injeta um elemento *'div'*, que contém uma hiperligação (Hyperlink) com o texto "Copy Code". Esta hiperligação é o que age como o novo botão de cópia de código, transportando o conteúdo sob a forma de um URL, garantindo que caracteres especiais não quebrem a estrutura.

Finalmente a interação com o botão é gerida pela função *'HyperLinkListener'* na classe *'UIGpt.kt'*, a classe que trata de todas as interações do utilizador com a interface do plugin. Ao detetar o protocolo de Cópia, o sistema executa duas operações.

- 1- A Decodificação do URL: Utiliza-se a o URLDecoder para restaurar o texto original.
- 2- Limpeza de Entidades: Devido à transferência de texto sobre diferentes formatos alguns caracteres ficam como entidades HTML, as mais significantes sendo as aspas (“) ficarem como ‘&quot;’ e o símbolo de maior (<) ser ‘&gt;’

### **4.2.3 Configuração Dinâmica**

A flexibilidade do plugin assenta numa arquitetura de configurações que tem como foco no futuro permitir ao utilizador personalizar a experiência de uso e alternar entre diferentes provedores de IA de forma dinâmica.

Esta Componente, mesmo ainda não ter sido implementada, já está a ser preparada.

Na janela de Settings do Plugin foram separadas as definições do DropProject e as da LLM, tornando-se mais apelativo visualmente e organizando-se melhor as diferentes opções.

A separação foi feita nas classes “GeneralSettingsComponent.kt” e “LLMSettingsComponent.kt”, que são classes de componentes visuais novas criadas tendo por base o código da antiga classe “SettingsComponent.kt”. Tendo também a classe “SettingsConfigurable.kt” que gere a lógica das definições.

Cada ecrã de configuração valida as entradas do utilizador, como a presença de um token válido, antes de permitir a gravação das alterações.

## **5. Solução Proposta**

### **5.1. Apresentação**

A solução proposta é um plugin atualizado e melhorado para o IntelliJ IDEA, baseado no Drop Project Plugin original, mas expandido para oferecer uma experiência melhorada de integração com inteligência artificial.

O plugin permitirá ao utilizador configurar modelos de IA, enviar feedback automaticamente para esses modelos e interagir com as respostas de forma mais conveniente, especialmente no que diz respeito à manipulação de blocos de código.

## 5.2. Tecnologias e Ferramentas

As principais ferramentas incluem: IntelliJ IDEA, Java/Kotlin, Gradle, APIs e JUnit para testes para diagramas. Serão também usadas bibliotecas para comunicação HTTP e processamento JSON.

**Justificação de escolhas** - As tecnologias selecionadas garantem compatibilidade com o IntelliJ e facilitam a evolução do plugin.

Java/Kotlin são adequados ao ecossistema JetBrains; Gradle simplifica o build; a arquitetura modular permite suportar vários fornecedores de IA; e o uso de testes assegura estabilidade. Estas escolhas visam tornar o plugin mais robusto, flexível e sustentável.

## 5.3. Ambientes de Teste e de Produção

Computador com acesso à internet, IntelliJ instalado como ambiente de testes, acesso a uma instalação do Drop project, local ou remota, Acesso a uma AI generativa a partir de uma API Key.

## 5.4. Abrangência

A implementação deste projeto envolve diversas áreas de conhecimento adquiridas ao longo da licenciatura, refletindo a abrangência disciplinar necessária para desenvolver um plugin funcional, integrado e orientado à interação com serviços.

Unidade Curricular (UC)	Conteúdo relevante aplicado ao projeto
Interação Humano-Máquina (IHM)	Princípios de usabilidade aplicados à interação do plugin e melhoria da experiência do utilizador.
LP2 – Linguagens de Programação II	Desenvolvimento orientado a objetos em Java, estruturação do projeto e manipulação de respostas.
Sistemas Operativos	Processamento de pedidos HTTP, comunicação com APIs externas (OpenAI), gestão de tempo de resposta e estrutura de protocolos.
Desenvolvimento de Interfaces Web	Manipulação de dados em formato JSON, tratamento de respostas estruturadas e integração com componentes visuais do plugin.

Link do repositório GitHub do projeto:

<https://github.com/AfonsoCastelhano/Drop-Project-for-IntelliJ-Idea-a22303665>

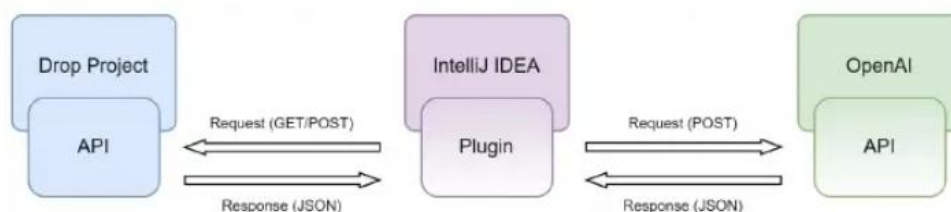
## 5.5. Componentes do sistema

O sistema global associado ao projeto envolve três componentes principais que interagem entre si para permitir a utilização do plugin no contexto do Drop Project:

1. Drop Project – Plataforma académica responsável pela gestão de assignments, submissões de código e avaliação automática.
2. Student Plugin – Plugin para IntelliJ IDEA que estende as funcionalidades do Drop Project, permitindo ao aluno obter assistência da ferramenta de IA diretamente no ambiente de desenvolvimento.
3. LLM (Large Language Model) – Serviço externo de IA generativa responsável por processar os pedidos do utilizador e devolver respostas estruturadas, código ou explicações.

O trabalho desenvolvido neste projeto irá alterar diretamente as duas primeiras componentes — Drop Project e Student Plugin — através da adição de novas opções de configuração, funcionalidades de controlo e melhorias de usabilidade. A comunicação com o LLM será feita de forma indireta, através de pedidos HTTP para a API correspondente, sem modificações diretas sobre esse serviço.

Para representar visualmente a relação entre os três componentes deste sistema, apresenta-se de seguida um diagrama da arquitetura global, onde se observa a interação entre o Drop Project, o Student Plugin e o LLM.



*Figura 4 - Diagrama de arquitetura global*

Este diagrama foi adaptado da Figura 1 do artigo do professor Bruno Cipriano, apresentado na conferência Koli Calling.

<https://dl.acm.org/doi/abs/10.1145/3699538.3699588>

## 5.6. Interfaces

Este projeto envolve comunicação entre três componentes principais, tal como descrito na secção anterior. A interação com o Drop Project será realizada através da sua API oficial, cuja documentação se encontra disponível em <https://playground.dropproject.org/dp/swagger-ui/>.

Esta API permitirá estabelecer a ligação necessária para o envio e receção de informação entre o plugin e a plataforma.

Para além disso, a comunicação com os modelos de linguagem (LLMs) será efetuada através das APIs fornecidas pelos respetivos serviços. Como exemplo, destaca-se a API da OpenAI, cuja documentação pode ser consultada em <https://platform.openai.com/docs/api-reference/introduction>.

Estas interfaces permitirão enviar o feedback do Drop Project para os modelos de IA e receber respostas orientadas à correção de código.

# 6. Testes e validação

## 6.1. Metodologia e Testes

Nesta fase final, a validação do plugin seguiu uma abordagem de **Testes de Aceitação do Utilizador** e **Testes Funcionais**.

- 1- Testes Manuais:** Cada funcionalidade de interface, como os botões, foi testada para garantir a reatividade da UI.
- 2- Testes de Integração:** Validou-se a comunicação “ponta-a-ponta” entre o IntelliJ IDEA e a API da GenAI.
- 3- Ambiente de Teste:** os testes foram realizados em ambiente sandbox do IntelliJ.

## 6.2. Casos de Teste e Resultados

ID	Requisito	Descrição do Teste	Resultado	Estado
T01	RF1	Resposta com 2 blocos de código distintos	Foram gerados 2 botões "Copy Code" independentes.	Sucesso
T02	RF2	Cópia de código com aspas (") e símbolos (<, >).	O código foi colado no editor sem entidades HTML.	Sucesso
T03	RF3	Envio de código selecionado via menu de contexto.	A ToolWindow enviou logo a mensagem copiada para a IA.	Sucesso

### 6.3. Análise de Resultados e Usabilidade

A implementação do sistema de cópia seletiva e a integração direta no editor através do ficheiro 'SendToGpt.kt' demonstraram uma melhoria significativa na usabilidade face à versão original do plugin.

### 6.4. Demonstração Visual

#### T01-

Como é demonstrado nas figuras 6 e 7, numa só conversa, com 2 prompts diferentes ambos os outputs da IA contêm um bloco de texto com o botão de cópia de código.

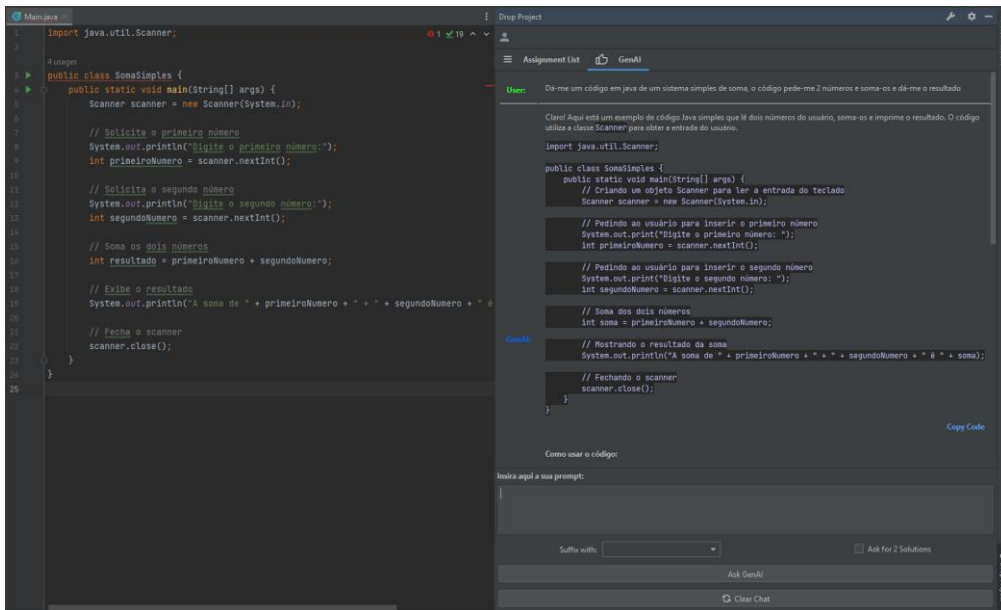


Figura 5 - 1º Bloco de Texto

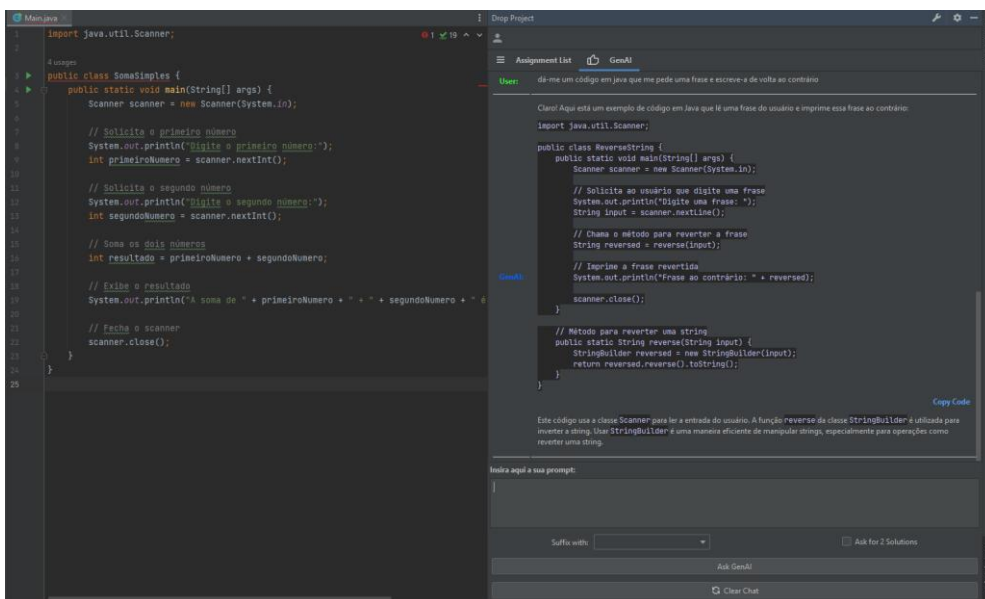


Figura 6 - 2º Bloco de Texto

## **T02-**

Aqui está anexado o link para o segundo teste, devido à necessidade de se visualizar o mesmo enquanto corre:

<https://www.youtube.com/watch?v=ei1MyqQdh9A>

## **T03-**

Junta-se em anexo o link para o terceiro caso de teste:

<https://www.youtube.com/watch?v=yi4xa3W3vfQ>

# **7. Método e Planeamento**

## **7.1. Introdução**

O desenvolvimento será feito de forma incremental, dividido em quatro etapas principais, estas sendo, a análise do plugin atual e levantamento de requisitos, a implementação das melhorias essenciais como a comunicação com API e uma secção separada nas configurações, a extensão do plugin para suportar múltiplos modelos de IA e testes e validação final no IntelliJ.

O trabalho será acompanhado regularmente pelo orientador para validar decisões e garantir o cumprimento do planeamento.

**Métodos técnicos** - Serão utilizados métodos de engenharia de software, incluindo análise do código existente, programação orientada a objetos em Java e Kotlin, desenvolvimento orientado a APIs, criação de arquitetura modular e realização de testes unitários e exploratórios.

Diagramas e modelos serão usados para documentar a estrutura e o comportamento do sistema.

**Abordagem usada** - Será utilizada uma abordagem centrada no utilizador e baseada em desenvolvimento incremental.

Pretende-se criar uma solução modular, fácil de manter e extensível, garantindo que novas funcionalidades possam ser adicionadas sem alterar a estrutura principal.

**A prioridade será a fiabilidade e a facilidade de utilização do plugin.**

## **7.2. Planeamento Inicial**

O planeamento inicial prevê quatro fases principais:

1. Fase de análise e levantamento de requisitos. Envolve a revisão do plugin existente, a identificação das limitações e a definição detalhada das funcionalidades a implementar.
2. Fase de implementação das melhorias essenciais. Inclui o restauro da comunicação com a API da OpenAI, a criação da secção de configurações e a melhoria da funcionalidade de cópia de código.
3. Fase de extensão da arquitetura para múltiplos providers. Nesta fase será criada a interface IProvider e implementado pelo menos um provider adicional.
4. Fase de testes, validação e refinamento final. Envolve testes no IntelliJ, validação das interações com a IA e ajustes antes da entrega final.

## **7.3. Análise Crítica ao Planeamento**

Embora o planeamento inicial esteja estruturado e alinhado com os objetivos do projeto, existem alguns aspetos a considerar criticamente:

A integração com APIs externas pode introduzir atrasos devido a limitações de quota, tempo de resposta e alterações inesperadas na documentação;

A arquitetura do plugin original pode conter dependências ocultas que exijam mais tempo para compreender e adaptar;

A implementação de múltiplos providers poderá revelar-se mais complexa do que previsto, exigindo maior abstração na interface IAuthProvider;

O tempo disponível poderá ser afetado pela carga de trabalho de outras unidades curriculares.

No entanto, o planeamento é suficientemente flexível para permitir ajustes ao longo do desenvolvimento, e o acompanhamento regular com o orientador mitiga riscos e facilita decisões técnicas.

## **8. Planeamento**

**Plano de Trabalho** – O projeto deverá ser feito em fases progressivas, começando numa fase de análise e terminando numa fase de testes. A ordem prevista para o trabalho é, restaurar o plugin a um estado funcional, de seguida melhorar as funcionalidades existentes e, finalmente, estender a arquitetura para suportar vários modelos de AI.

### **Tarefas –**

- Análise do plugin existente
- Levantamento e detalhamento de requisitos
- Atualização da comunicação com a API da OpenAI
- Criação da secção “OpenAI Settings”
- Melhoria da funcionalidade de cópia de código
- Implementação de suporte para múltiplos provedores de IA
- Testes unitários e exploratórios
- Documentação e preparação dos artefactos técnicos



## 9.2. Trabalho Futuro

Como perspectiva de evolução, planeia-se melhorar a integração do plugin diretamente no editor através de um menu de contexto ativado pelo botão direito do rato. Esta funcionalidade permitirá selecionar um trecho de código em vez de somente escolher a opção "Ask GenAI to...", isto irá apresentar uma lista dinâmica de prefixos configurados, como instruções para encontrar erros ou otimizar a performance.

Ao selecionar uma destas opções, a janela de chat abrir-se-á automaticamente com o prefixo e o código selecionado já preenchidos, agilizando drasticamente a interação.

Complementarmente, o suporte a modelos de linguagem locais e a criação de um histórico de mensagens persistente são vistos como passos fundamentais para tornar a ferramenta ainda mais privada, robusta e eficiente para o ensino de programação.

## 10. Elementos Técnicos

**Modelo Conceptual** – Representa as principais entidades e relações do sistema, permitindo compreender a estrutura lógica do plugin e a forma como interage com o Drop Project e com o serviço de IA.

**Use Cases** - Identificam os atores envolvidos e os principais casos de uso do plugin, descrevendo de forma clara as operações que o utilizador pode realizar e o comportamento esperado do sistema.

**Planeamento**- O cronograma do projeto está representado no diagrama de Gantt seguinte, evidenciando a distribuição das tarefas ao longo do ano letivo.

## **11. Github**

Aqui encontra-se o link para o repositório Github do Projeto na organização dos TFC's:

<https://github.com/DEISI-ULHT-TFC-2025-26/Drop-Project-Student-Plugin>

Para estar funcional, está em falta somente o URL da IA e a API key, mas estes devem estar guardados para motivos de privacidade.

## **12. Inquérito sobre o uso de IA**

Foi-nos pedido aos alunos para responder a um questionário sobre o uso de IA no âmbito do nosso trabalho, aqui encontram-se as resposta ao inquérito do forms e o ficheiro do questionário word.

## 12.1. Inquério Forms

### Uso de IA Generativa no Trabalho de Fim de Curso (TFC) — Questionário Inicial

Este questionário é anónimo e serve para investigação pedagógica sobre como estudantes usam IA generativa (ex: ChatGPT, Copilot, Gemini) em projetos longos. Não tem impacto na avaliação. Responde com base na tua prática real.

1. Estás a desenvolver o TFC em:

- Grupo
- Individual

2. Em que fase do TFC estás neste momento?

- Planeamento / requisitos
- Arquitetura / design
- Arquitetura / design
- Implementação avançada
- Testes / validação
- Escrita do relatório
- Outra

3. Área principal do teu TFC (escolhe a mais próxima):

- Web / backend
- Mobile
- Sistemas / redes
- IA / data
- Segurança
- Jogos / multimédia
- Outra

Figura 8 - Respostas ao inquérito

4. Que ferramentas de IA generativa usas (ou tencionas usar) no TFC?

- ChatGPT
- GitHub Copilot
- Gemini
- Claude
- Perplexity
- Outro

5. Em média, quantas interações (prompts) fazes num dia em que usas IA?

- Não uso IA
- 1-2
- 3-5
- 6-10
- 11-20
- >20

6. Para que tarefas usas (ou tencionas usar) IA no TFC?\*

- Compreender conceitos / tirar dúvidas
- Gerar código (funções, módulos)
- Refatorar / melhorar código existente
- Debugging (encontrar causas, sugerir correções)
- Escrever ou melhorar testes
- Revisão de código (code review)
- Documentação / relatório
- Design/arquitetura (alternativas, trade-offs)
- Segurança (ameaças, validações, checks)
- Outro

Figura 9 - Respostas ao inquérito

7. Quando pedes ajuda à IA, o que fazes mais frequentemente?

- Peço uma solução completa e adapto
- Peço explicação e depois implemento eu
- Peço hipóteses / alternativas e escolho
- Peço validação de algo que eu já fiz
- Outro

8. Quando a IA sugere código, o que fazes normalmente antes de aceitar? (escolhe a opção mais frequente)

- Copio e executo sem validar
- Executo com 1-2 testes simples
- Leio o código e tento entender linha a linha
- Comparo com documentação/teoria/referências
- Escrevo testes e tento cobrir casos limite
- Outro

9. Quando algo falha, qual é a tua estratégia mais comum?

- Peço nova solução à IA imediatamente
- Tento isolar o erro manualmente e depois peço ajuda
- Uso logs/prints/debugger primeiro
- Releio requisitos/teoria/documentação primeiro
- Outro

10. Com que frequência verificas se a resposta da IA tem suposições erradas (ex.: detalhes técnicos incorretos, confundir com outro termo semelhante)? (1 estrela não acontece, 10 estrelas acontece sempre).

★ ★ ★ ★ ★ ☆ ☆ ☆ ☆

Figura 10 - Respostas ao inquérito

11. Nestas frases indica o teu nível concordância.

	Strongly agree	Agree	Neither agree nor disagree	Disagree	Strongly disagree
IA torna o meu trabalho no TFC mais rápido	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
IA melhora a minha compreensão do que estou a fazer.	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sinto-me confiante a avaliar se uma resposta da IA está correta	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tenho um método claro de validação (testes, invariantes, checks) antes de aceitar resultados	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Em tarefas complexas, sinto que a IA pode "parecer certa" mesmo estando errada	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

12. Em que tipo de tarefa consideras que a IA é menos fiável no teu TFC?

- Decisões de arquitetura / trade-offs
- Concurrency / sincronização
- Debugging de erros subtis
- Segurança
- Integração entre componentes
- Otimização/performance
- Especificações/requisitos ambíguos
- Outro

Figura 11 - Respostas ao inquérito

13. Qual destas afirmações descreve melhor a tua visão atual

- A IA já resolve grande parte do trabalho técnico por mim
- A IA ajuda muito, mas preciso validar bastante
- A IA ajuda pouco; é mais confusa do que útil
- Ainda não tenho opinião / não usei o suficiente

14. Opcional: Escreve um bom exemplo de prompt que usaste recentemente no TFC

Figura 12 - Respostas ao inquérito

## 12.2. Inquérito em ficheiro Word



DEISI\_TFC\_FormUso  
IA.docx

## 13. Bibliografia

- [DEISI24] DEISI, Regulamento de Trabalho Final de Curso, Out. 2024.
- [DEISI24b] DEISI, [www.deisi.ulusofona.pt](http://www.deisi.ulusofona.pt), Out. 2024.
- [TaWe20] Tanenbaum,A. e Wetherall,D., *Computer Networks*, 6ª Edição, Prentice Hall, 2020.
- [ULHT21] Universidade Lusófona de Humanidades e Tecnologia, [www.ulusofona.pt](http://www.ulusofona.pt),  
acedido em Out. 2024.

## 14. Glossário

AI- Artificial Intelligence

IA- Inteligência artificial

API- Application programming interface

LLM- Large Language Models

LEI- Licenciatura em Engenharia Informática

DP – Drop Project

GenAI- Inteligência artificial generativa

IDE (Integrated Development Environment): Ambiente de desenvolvimento integrado

SDK – kit de desenvolvimento do Software

JSON (JavaScript Object Notation) – formato leve de intercâmbio de dados

Plugin – extensão do software que adiciona funcionalidades específicas

Regex – Linguagem de marcação leve

Swing - Biblioteca gráfica

TollWindow - Componente de interface do IntelliJ que representa as janelas ancoradas nas bordas da IDE