



UNIVERSIDADE  
LUSÓFONA

# Plugin Drop Project para Professores

## Trabalho Final de curso

Relatório Final

Nome do Aluno: João Marques

Nome do Orientador: Bruno Cipriano

Nome do Orientador: Rodrigo Correia

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

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

## **Direitos de cópia**

Plugin DP para Professores, Copyright de João Marques, 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.

# Índice

Índice .....	iii
Lista de Figuras .....	iv
Lista de Tabelas .....	v
Resumo .....	vi
Abstract .....	vii
1 Identificação do Problema .....	1
2 Viabilidade e Pertinência .....	3
2.1 Inquérito .....	3
2.2 Principais conclusões do inquérito .....	4
3 Benchmarking .....	6
4 Engenharia .....	7
4.1 Levantamento e análise dos Requisitos .....	7
4.2 Diagramas de Casos de Uso ou descrição de cenários de aplicação .....	9
4.3 Diagramas de Sequência .....	10
4.4 Mockups .....	12
5 Solução Desenvolvida .....	14
5.1 Arquitetura .....	15
5.2 Implementação .....	16
5.3 Abrangência .....	16
6 Plano de testes e validação .....	17
6.1 Resultados .....	17
6.1.1 Bugs reportados .....	17
6.1.2 Problemas de usabilidade .....	17
7 Calendário .....	19
8 Resultados .....	20
9 Conclusão .....	21
Anexos .....	22
Progresso de trabalho .....	22
Bibliografia .....	23
Glossário .....	24

# Lista de Figuras

Figura 1 – Interface de listagem de exercícios do <i>plugin</i> anterior	2
Figura 2 - Interface de listagem de exercícios do <i>plugin</i> atual	2
Figura 3 – 1ª pergunta: “Utilizaria o <i>plugin</i> do Drop Project no IntelliJ?” Todos os docentes entrevistados mostraram interesse em utilizar o <i>plugin</i> para professores.	3
Figura 4 – 2ª pergunta: “Ao ter acesso aos alunos que já fizeram uma submissão, seria prático ter uma opção de fazer <i>download</i> da última submissão desse aluno, mesmo podendo fazer <i>download</i> dessa submissão quando abrir todas as submissões desse aluno e selecionar a última para <i>download</i> ?” A larga maioria (7 em 8) dos professores concorda que seria prático ter esta opção disponível, com apenas um professor a não ver esta funcionalidade como prática	3
Figura 5 - Diagrama de <i>Use Case</i>	9
Figura 6 - Diagrama de Sequência do <i>Login</i>	10
Figura 7 - Diagrama de Sequência da listagem de exercícios	10
Figura 8 - Diagrama de Sequência do <i>download</i> de uma submissão	11
Figura 9 - Diagrama de Sequência da visualização do <i>build report</i>	11
Figura 10 - Screenshot do <i>dashboard</i>	12
Figura 11 - Screenshot da lista de submissões de um exercício	12
Figura 12 - Screenshot da lista de exercícios	13
Figura 13 - Diagrama da arquitetura	15

# Lista de Tabelas

Tabela 1 - <i>Workflows</i>	1
Tabela 2 - Comparação das alternativas	6
Tabela 3 - Requisitos	7
Tabela 4 - Lista de <i>endpoints</i> implementados	8
Tabela 5 - <i>Bugs reportados</i>	17
Tabela 6 - Problemas de usabilidade reportados	17
Tabela 7 - Calendário	19
Tabela 8 - Tabela de progresso	22

## Resumo

Este projeto consiste no desenvolvimento de um *plugin* para a integração da funcionalidade do Drop Project (DP) [BCNP22] para professores com o ambiente de desenvolvimento (referido a partir de aqui como IDE) *IntelliJ IDEA*, de forma a facilitar e acelerar as tarefas realizadas com a interface web, sem ser necessário alternar entre o browser e o IDE. Este também deverá permitir uma melhor integração com as funcionalidades do IDE, como permitir descarregar e abrir os projetos dos alunos diretamente no IDE, facilitando desta forma as ajudas por parte dos professores aos alunos.

---

# Abstract

This project consists of developing a plugin to integrate Drop Project's (DP) [BCNP22] features for teachers with the integrated development environment (referred hereafter as IDE) *IntelliJ IDEA*, as to facilitate and speed up the tasks performed usually performed on the web interface, without having to switch between the browser and the IDE. This plugin shall also allow for a better integration with the IDE's features, such as downloading and opening the students' projects directly on the IDE, thus facilitating help from teachers to students.





# 1 Identificação do Problema

O Drop Project (DP) é uma ferramenta de avaliação automática de trabalhos de programação (*Automatic Assessment Tool* ou AAT) desenvolvida por docentes da Universidade Lusófona e está a ser usada na mesma desde 2018 para avaliar trabalhos, projetos e testes em cadeiras que usam Java e/ou Kotlin (atualmente Fundamentos de Programação, Algoritmia e Estruturas de Dados, Linguagens de Programação 2 e Computação Móvel).

Com o modo de funcionamento atual, o utilizador (um utente da universidade com acesso ao DP para professores) tem de realizar tarefas como o descarregamento dos trabalhos dos alunos e a consequente importação para o IDE *IntelliJ* manualmente, tal como teria de estar a trocar entre a janela do IDE e a janela do *browser* frequentemente, potencialmente reduzindo a produtividade. Com a criação de uma integração entre estes dois, esta redução de produtividade seria mitigada significativamente, visto que os processos manuais de download e importação para o IDE deixariam de ser necessários.

Exemplos de *workflows* comuns que sofrem do problema descrito anteriormente (Tabela 1):

**Tabela 1 - Workflows**

<b>Workflow A</b>
<ol style="list-style-type: none"> <li>1. Um aluno pede ajuda</li> <li>2. O professor vai ao DP, procura o exercício, procura a submissão do aluno, e abre o <i>build report</i> (este passo pode ser otimizado com a filtragem e ordenação de submissões e exercícios)</li> <li>3. O professor faz <i>download</i> da submissão do aluno, descompacta o ficheiro e abre-o no IDE (este passo pode ser otimizado com o <i>download</i> feito diretamente a partir do IDE)</li> </ol>
<b>Workflow B</b>
<ol style="list-style-type: none"> <li>1. Um aluno pede ajuda</li> <li>2. O professor pede o <i>URL</i> do <i>build report</i> ao aluno             <ol style="list-style-type: none"> <li>2.1. O professor espera pela resposta do aluno</li> </ol> </li> <li>3. O professor abre o relatório (este passo passaria a ser feito no IDE)</li> <li>4. O professor faz <i>download</i> da submissão, descomprime e abre-o no IDE (este passo também passaria a ser efetuado diretamente no IDE)</li> </ol>

Foi desenvolvido no ano letivo 2020/2021 um TFC (Trabalho Final de Curso) antecessor a este, porém o protótipo do plugin desenvolvido não utiliza a API (*Application Programming Interface*) atual do DP nem o sistema de login à base de *tokens*, tornando o seu uso atual impossível, além de nunca ter sido testado por professores em ambientes de trabalho real e de não possuir funcionalidades críticas à aceleração do *workflow*, como o “*Student History*” (uma página onde o professor tem acesso a todas as submissões de um aluno, em todos os exercícios, a partir do nome ou número de aluno), forçando o utilizador a navegar pelos exercícios.

Este plugin visa usar uma API mais atualizada, tal como uma interface gráfica melhorada e mais familiar para utilizadores da interface *web*.

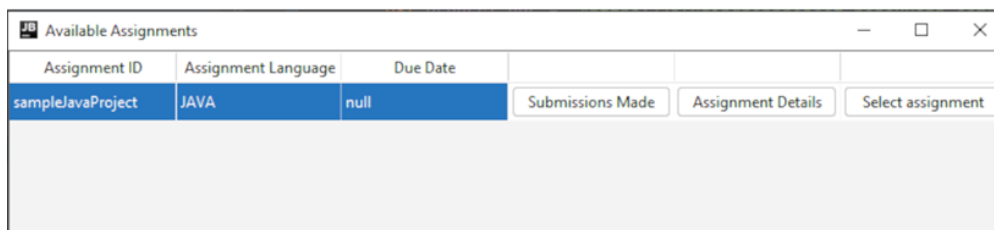


Figura 1 – Interface de listagem de exercícios do *plugin* anterior

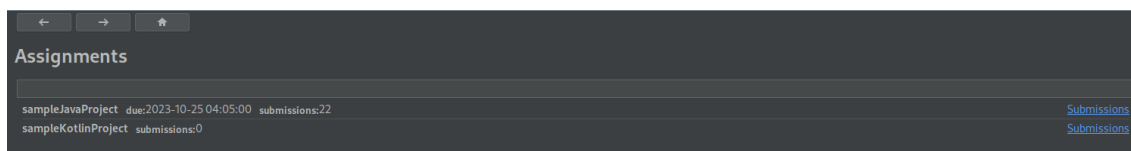


Figura 2 - Interface de listagem de exercícios do *plugin* atual

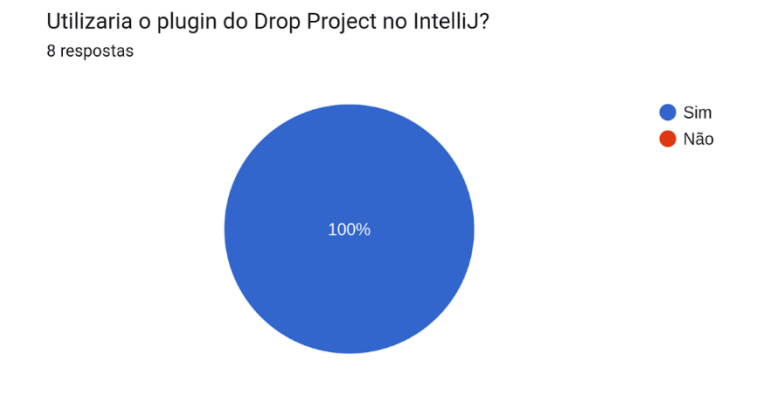
Em relação ao *plugin* anterior, o *plugin* atualmente a ser desenvolvido (Figura 2) possui pesquisa, navegação entre ecrãs (o *plugin* anterior usava janelas em vez de abas, Figura 1), e, no futuro, filtragem de entradas, para melhorar o *workflow* atual.

## 2 Viabilidade e Pertinência

Para averiguar o interesse de um plugin para professores, assim como as funcionalidades relevantes para os mesmos, foi feito um inquérito a 8 docentes pelo aluno responsável pela TFC predecessora desta [DIOG21], cujos resultados indicam que todos os professores teriam interesse em usar um plugin para o IDE *IntelliJ IDEA* que melhorasse o *workflow* do DP para professores, tal como o interesse em funcionalidades como o download de submissões, pesquisa, ordenação e filtragem de exercícios, acesso ao *build report* e criação de exercícios diretamente do IDE.

### 2.1 Inquérito

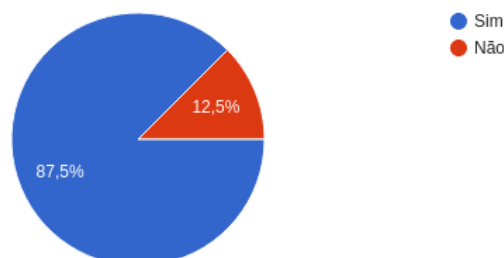
As perguntas e respostas relevantes a este projeto são as seguintes (Figura 3, Figura 4):



**Figura 3 – 1ª pergunta: “Utilizaria o plugin do Drop Project no IntelliJ?”**  
Todos os docentes entrevistados mostraram interesse em utilizar o plugin para professores.

Ao ter acesso aos alunos que já fizeram uma submissão, seria prático ter uma opção de fazer download da última submissão desse aluno, mesmo podendo fazer download dessa submissão quando abrir todas as submissões desse aluno e selecionar a última para download?

8 respostas



**Figura 4 – 2ª pergunta:**  
“Ao ter acesso aos alunos que já fizeram uma submissão, seria prático ter uma opção de fazer download da última submissão desse aluno, mesmo podendo fazer download dessa submissão quando abrir todas as submissões desse aluno e selecionar a última para download?”  
A larga maioria (7 em 8) dos professores concorda que seria prático ter esta opção disponível, com apenas um professor a não ver esta funcionalidade como prática

À pergunta “Que outras funcionalidades gostaria que existissem neste Plugin?”, os inquiridos responderam:

Professor 1:

- “Ver o último relatório para cada Submissão (sem ter que correr o maven/java/projecto/o\_que\_for)”
- “Talvez escolher as colunas que queria ver nas várias listagens (e que o IntelliJ guardasse as minhas preferências)”
- “Poder exportar para excel (com a mesma capacidade anterior - escolher o que exportar)”
- “Na exportação para excel, ter links directos para a submissão a que corresponde e ao código git (se for esse o caso)”
- “Ter funcionalidade para fazer download de todas as submissões marcadas como finais”
- “Ter funcionalidade para poder ver código das submissões (por exemplo, serem "módulos"/"projectos" no IntelliJ - devia ser possível navegar no código, correr, etc)”
- “Possivelmente podia querer, como professor, actualizar o código do Assignment sem ser preciso ir à página *web* do DP (possivelmente no futuro, vou querer todas as funcionalidades que há na vertente *web* do DP)”
- “Gostaria que houvesse uma ordenação por omissão, se calhar por número de aluno”
- “Ou talvez que o sistema guardasse as minhas preferências, ou a última "coisa" que fiz”
- “Ordenar por grupo”

Professor 2:

- “Poder escolher 2 ou 3 submissões (do mesmo assignment) e abrir as mesmas no IntelliJ ao mesmo tempo (2 ou 3 janelas novas).”
- “Pesquisar Assignments por nome de aluno que submeteu”

Professor 6:

- “A mais importante de todas é conseguir "abrir" a submissão de um aluno com um click (equivalente a fazer download, descomprimir e importar como maven).”
- “"Nice to have" seria a possibilidade de refrescar uma submissão prévia de um aluno para a última versão (equivalente ao git pull). O meu objectivo principal é conseguir fazer debug das submissões dos alunos no IntelliJ.”
- “Também gostava de conseguir criar um assignment a partir de um projecto do IntelliJ que: (1) criasse um repo no github e colocasse lá o projeto e (2) criasse o assignment no DP automaticamente associado ao repo.”
- “Ordenação dos assignments por data da última submissão (do mais recente para o mais antigo)”

Professor 8:

- “Ativar/desativar assignments”
- “Não a última submissão mas acesso à submissão que passa em mais teste unitários”

## 2.2 Principais conclusões do inquérito

- Todos os professores inquiridos indicaram que utilizariam o *plugin*
- A maioria dos inquiridos vê interesse na funcionalidade de *download* da última submissão de um aluno

- Vários professores consideram importante que o *plugin* inclua opções para ordenação e pesquisa
- Alguns professores gostariam de ver funcionalidades relacionadas com a gestão de exercícios

### 3 Benchmarking

Atualmente, não existem plugins compatíveis com as funcionalidades de professor do DP, porém já existe um plugin para os alunos, e plugins para outras plataformas semelhantes, como o “Raccode” [AFMS18] (para o IDE *Eclipse*, e a plataforma *Mooshak*) e o plugin da plataforma *WEB-CAT* [WBCT23] (também para o IDE *Eclipse*)

**Tabela 2 - Comparação das alternativas**

Título	Descrição	Plataforma
Plugin para alunos do DP	<ul style="list-style-type: none"> <li>• Mostra os detalhes do exercício</li> <li>• Permite fazer submissões a partir do IDE <i>IntelliJ</i></li> <li>• Permite ver o resultado da submissão</li> </ul>	AAT: DP IDE: <i>IntelliJ</i>
<i>Raccode</i>	<ul style="list-style-type: none"> <li>• Permite fazer submissões diretamente do IDE <i>Eclipse</i></li> <li>• Mostra a classificação das equipes num formato tipo “<i>leaderboard</i>”</li> </ul>	AAT: <i>Mooshak</i> IDE: <i>Eclipse</i>
Plugin <i>Web-CAT</i>	<ul style="list-style-type: none"> <li>• Permite fazer submissões diretamente do IDE <i>Eclipse</i></li> </ul>	AAT: <i>Web-CAT</i> IDE: <i>Eclipse</i>

Em resumo, apesar de existirem ferramentas semelhantes, estas são feitas principalmente para os alunos, ou funcionam apenas com outros AATs, sendo assim este projeto o único a suportar um *workflow* para professores para o Drop Project.

## 4 Engenharia

### 4.1 Levantamento e análise dos Requisitos

Com os resultados do inquérito referido no capítulo *Viabilidade e Pertinência* e os objetivos já pré-estabelecidos, foram identificados os seguintes requisitos:

**Tabela 3 - Requisitos**

ID	Texto	Estado	Prioridade	Complexidade
1	Download de submissões - O plugin deverá permitir ao utilizador descarregar e abrir as submissões no IDE	Feito	Must have	3 pontos
2	Listagem de exercícios - O plugin deverá mostrar ao utilizador uma lista com os exercícios a que tem acesso	Feito	Must have	2 pontos
3	Listagem de submissões - O plugin deverá mostrar ao utilizador uma lista de todas as submissões feitas a um exercício	Feito	Must have	2 pontos
4	<i>Student History</i> - O plugin deverá mostrar ao utilizador uma lista das submissões feitas por um aluno em todos os exercícios	Feito	Should have	5 pontos
5	<i>Build Report</i> - O plugin deverá mostrar ao utilizador o <i>build report</i> de uma submissão	Feito	Must have	5 pontos
6	Pesquisa de submissões - O plugin deverá permitir ao utilizador pesquisar submissões por nome ou número do aluno que fez a submissão	Feito	Should have	1 ponto
7	Pesquisa de exercícios - O plugin deverá permitir ao utilizador pesquisar exercícios por nome ou <i>tag</i>	Feito	Should have	1 ponto
8	Criação/Atualização de exercícios - O plugin deverá permitir ao utilizador criar ou atualizar (código e/ou configurações) um exercício já existente	Por fazer	Nice to have	21 pontos
9	Ativação/Desativação de exercícios - O plugin deverá permitir ao utilizador	Feito	Should have	2 pontos

	ativar ou desativar um exercício existente			
10	<i>Dashboard</i> - O plugin deverá ter um menu que permita ao utilizador realizar pesquisas mais facilmente, melhorando assim o <i>workflow</i>	Feito	Should have	2 pontos
11	Marcação de submissões como “submissões finais” - O plugin deverá permitir ao utilizador marcar uma submissão como “final”	Feito	Should have	2 pontos

Dos requisitos elaborados, só ficou por implementar a criação e atualização de exercícios existentes, sendo que seria possível fazê-lo fora do âmbito de TFC.

Devido ao estado incompleto da API usada pelo *plugin*, foi necessário a implementação de certos *endpoints* para algumas das funcionalidades:

**Tabela 4 - Lista de *endpoints* implementados**

Nome do requisito	Endpoint
Listagem de submissões	<i>getAssignmentLatestSubmissions</i>
<i>Student History</i>	<i>getStudentHistory</i>
Download de submissões	<i>downloadProject</i>
<i>Build Report</i>	<i>getBuildReport</i>
<i>Dashboard</i>	<i>searchStudents</i> e <i>searchAssignments</i>
Ativação e desativação de exercícios	<i>toggleAssignmentState</i>
Marcação de submissões como “submissões finais”	<i>markAsFinal</i>



## 4.2 Diagramas de Casos de Uso ou descrição de cenários de aplicação

Os casos de uso identificados no projeto encontram-se no diagrama abaixo:

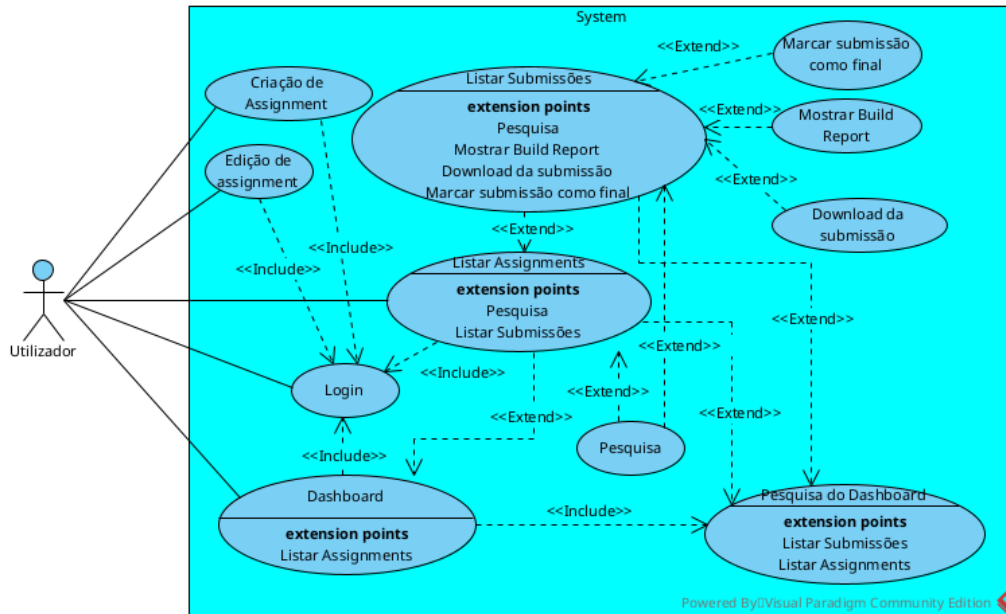


Figura 5 - Diagrama de Use Case

### 4.3 Diagramas de Sequência

Foram elaborados os seguintes diagramas de sequência para as operações principais (*Login*, listagem de exercícios, *download* e *build report*):

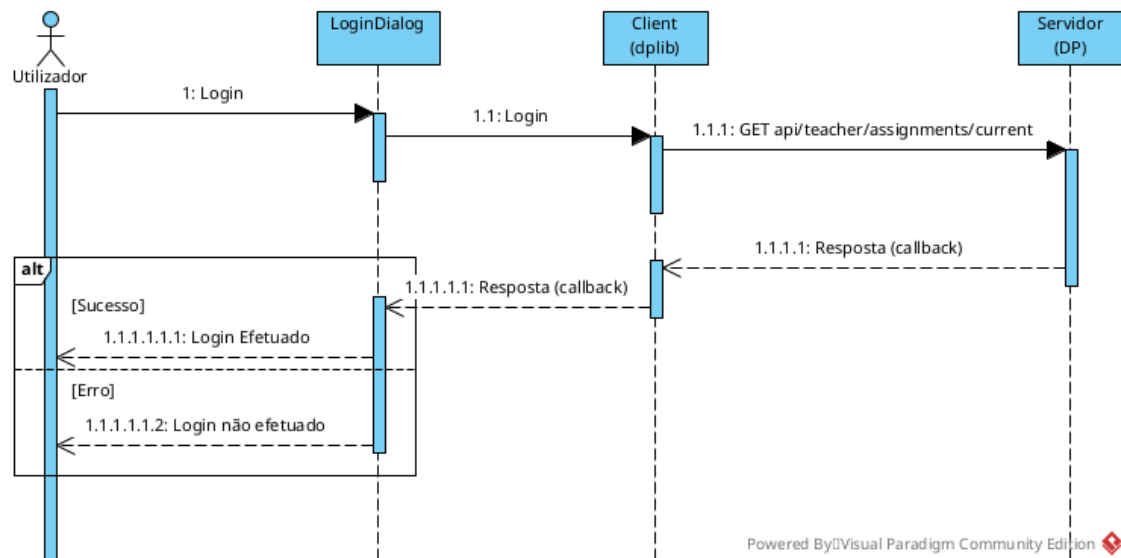


Figura 6 - Diagrama de Sequência do *Login*

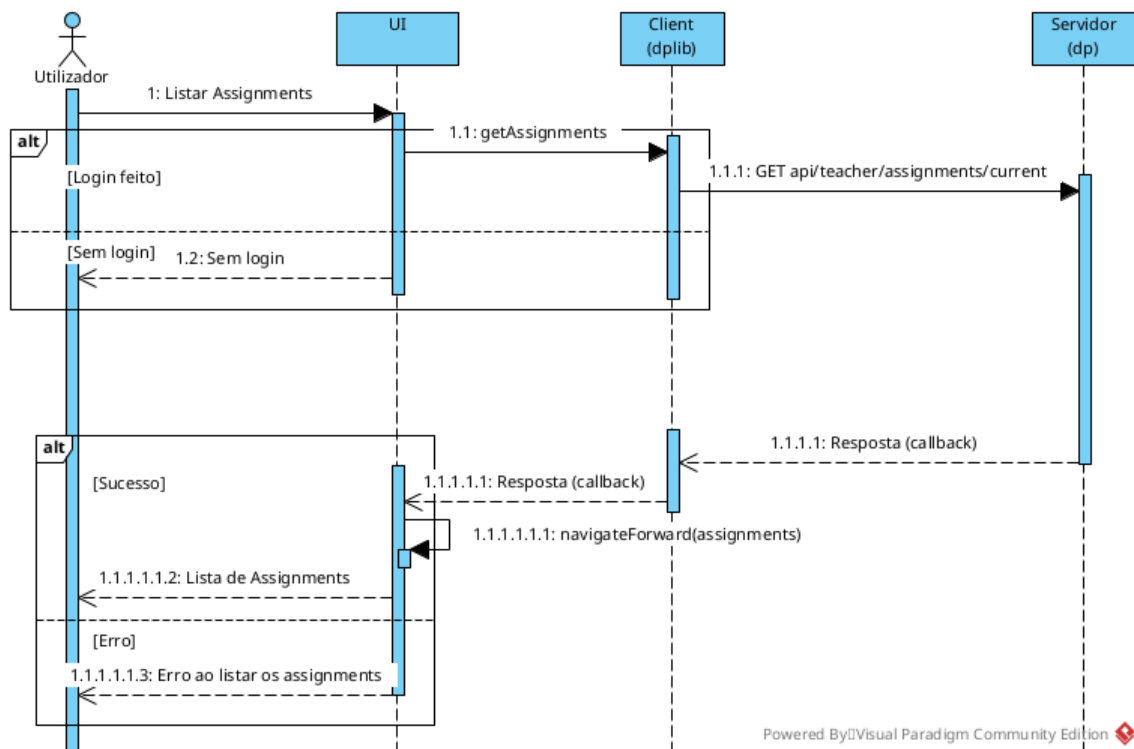


Figura 7 - Diagrama de Sequência da listagem de exercícios

Este diagrama (Figura 7), mesmo que seja específico para a listagem de exercícios, é muito próximo dos outros tipos de listagem. Caso o *login* esteja feito, o *plugin* faz um pedido ao DP para obter os dados referentes à listagem a ser efetuada, e mostra a página associada a esta.

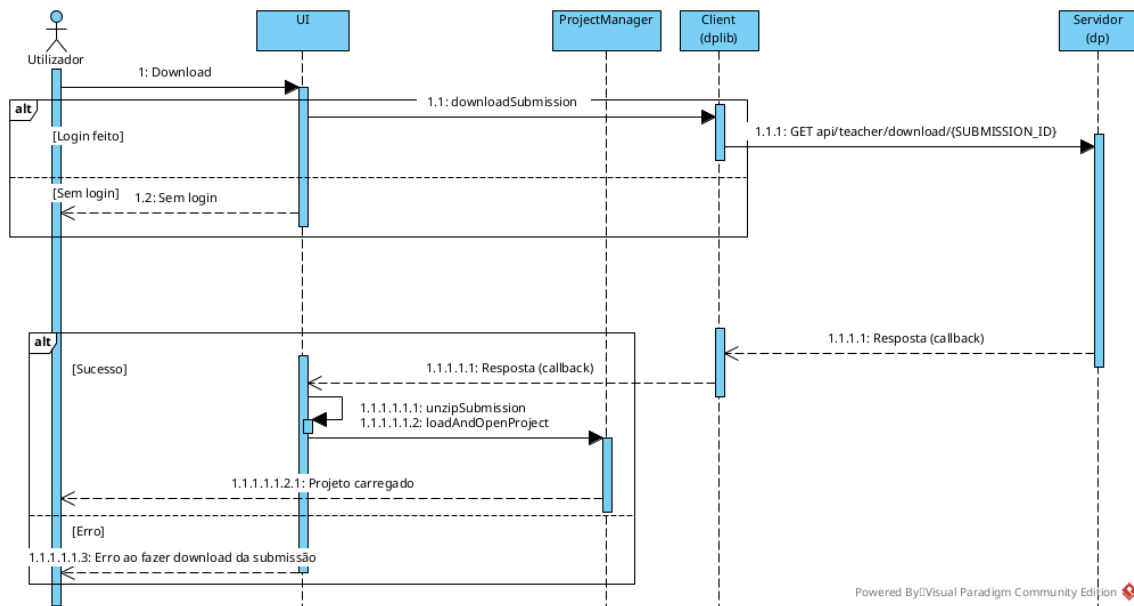


Figura 8 - Diagrama de Sequência do *download* de uma submissão

Após o *download* ter sido feito, o *plugin* descompacta o ficheiro para a pasta temporária do sistema, e carrega o projeto através da *API* do *IntelliJ*.

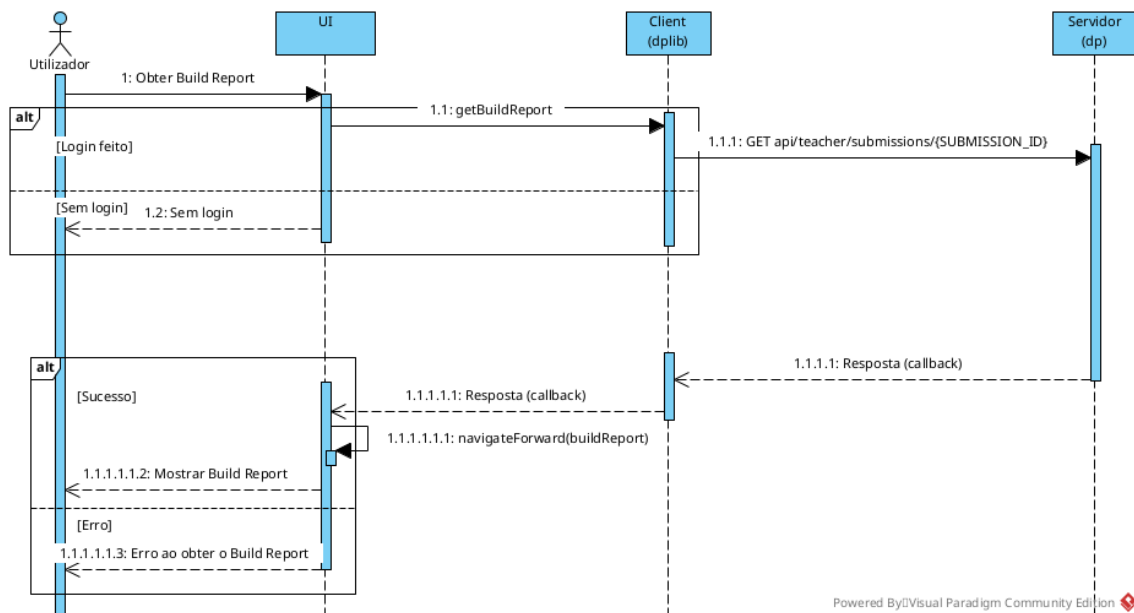


Figura 9 - Diagrama de Sequência da visualização do *build report*

## 4.4 Mockups

As seguintes figuras 10 e 11 são *screenshots* de uma versão anterior do *plugin*:

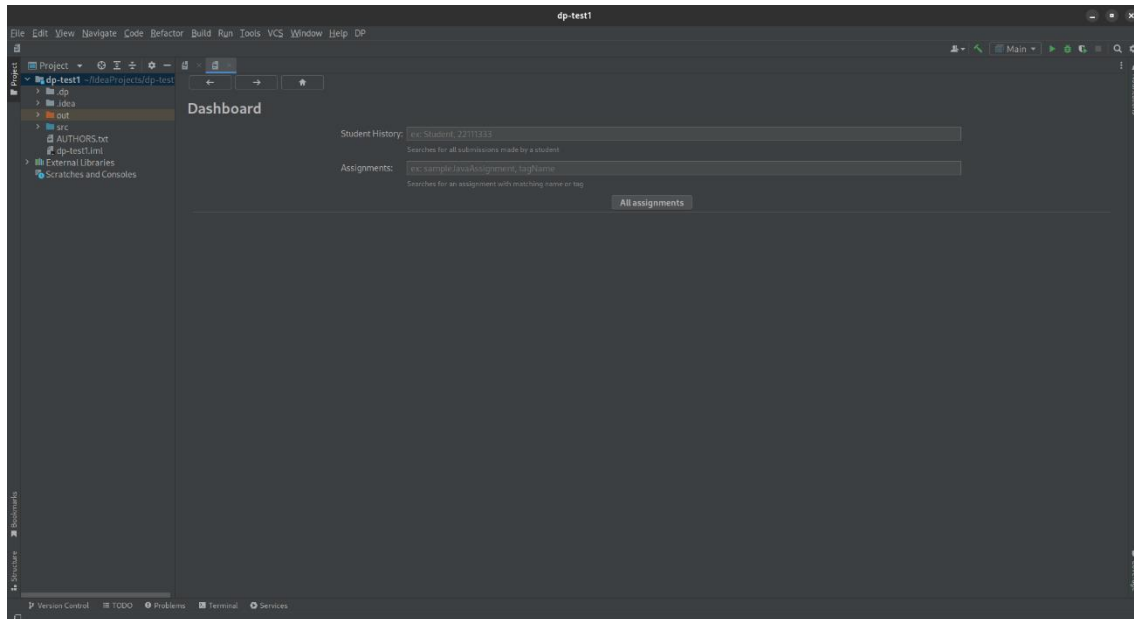


Figura 10 - Screenshot do *dashboard*

Este é o *dashboard*, uma página que vai dar acesso às principais funções de pesquisa, com o objetivo de permitir que o utilizador pesquise mais rapidamente.

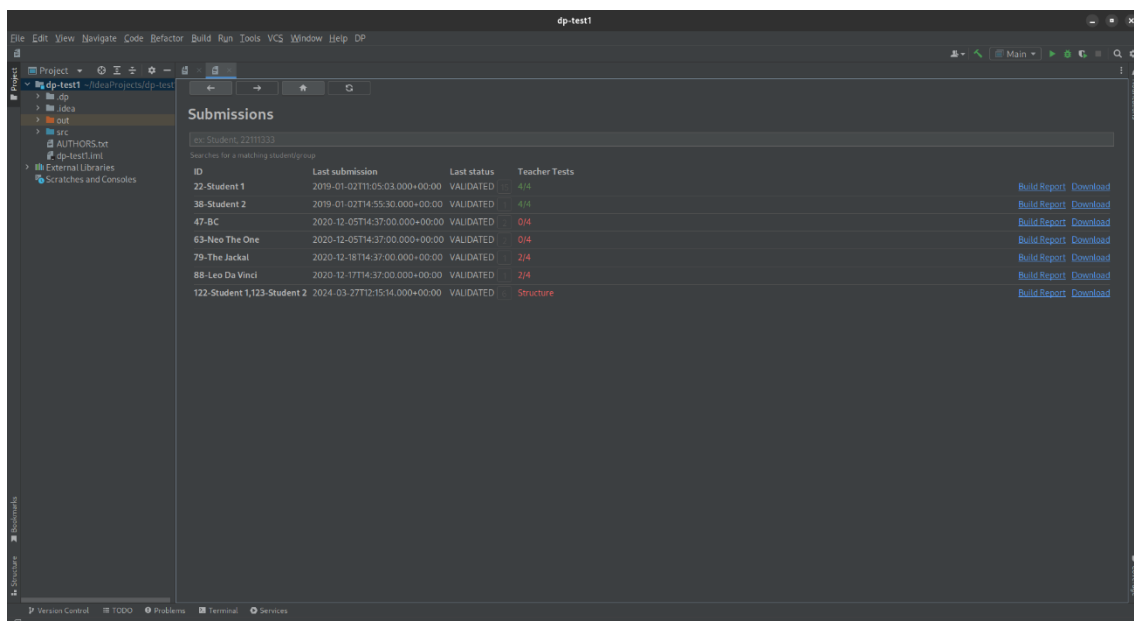
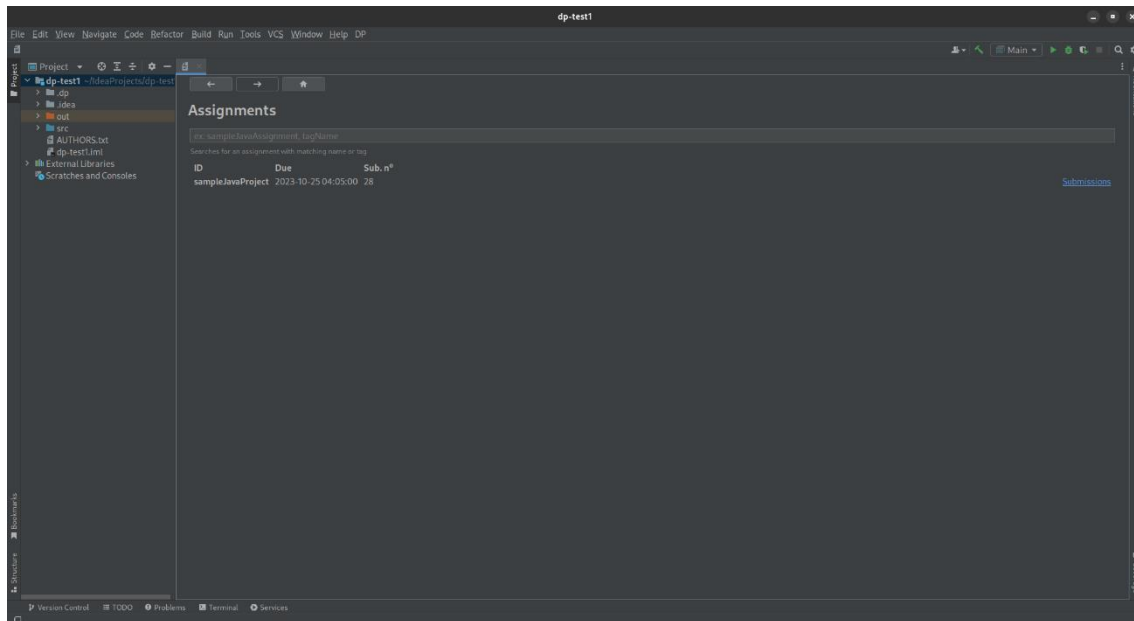


Figura 11 - Screenshot da lista de submissões de um exercício

A lista de submissões de um exercício contém as informações básicas sobre as últimas submissões realizadas por cada aluno, além do botão de download desta mesma.



**Figura 12 - Screenshot da lista de exercícios**

A lista de exercícios contém apenas os *IDs* das submissões, a data de criação da mesma, e o número de submissões feitas.

## 5 Solução Desenvolvida

O plugin foi implementado em *Kotlin*, usando a API do *IntelliJ IDEA* [JBII23] e o *Swing*, para a interface gráfica.

Para interagir com o Drop Project, foi usado o *OkHttp* [BLCK23] da *Block*<sup>1</sup>, antiga *Square*, em conjunto com a biblioteca de interpretação de JSON do *Kotlin* para processar os dados recebidos.

O objetivo do trabalho é a implementação tanto das funções já disponíveis na interface *web* como de funcionalidades extra que potencialmente iriam melhorar o *workflow*.

As funcionalidades implementadas na solução foram referidas nos requisitos anteriormente definidos, dando prioridade aos requisitos com prioridades *Must have* e *Should have*.

Como a API para professores do DP só possuía um *endpoint* para obter a lista de exercícios, teve de haver alterações a esta, para suportar as funcionalidades descritas acima, estando disponível uma lista completa destes na secção “Levantamento e análise dos Requisitos”.

Os repositórios *Git* da solução desenvolvida podem ser acedidos através dos seguintes *URLs*:

- Plugin: [github.com/DEISI-ULHT-TFC-2023-24/TFC-DEISI37-Plugin-DP-Professores](https://github.com/DEISI-ULHT-TFC-2023-24/TFC-DEISI37-Plugin-DP-Professores)
- Drop Project: [github.com/joao-marques-a22108693/drop-project](https://github.com/joao-marques-a22108693/drop-project)

---

<sup>1</sup> <https://block.xyz>

## 5.1 Arquitetura

As principais componentes presentes são as seguintes:

- *UI*: Interface constituída pelo DPTabProvider, pelos TabProviders e pelas Tabs em si, usa um padrão inspirado no “view locator” em que o DPTabProvider usa o Tab Provider mais apropriado para o tipo de dados que se pretende mostrar na interface, facilitando assim a adição e alteração de tabs
- *Client API (dplib)*: Biblioteca que comunica com a API do DP, tem um conjunto de funções assíncronas à base de *callbacks*, para garantir que nenhuma operação é necessariamente bloqueante, e um conjunto de funções síncronas
- *TeacherAPIController*: API para professores do DP que está a ser desenvolvida em paralelo com este projeto, usa *dependency injection* para obter os componentes necessários para processar os pedidos
- *ProjectManager*: Componente do *IntelliJ* usada pela *UI* para abrir os projetos descarregados pelo utilizador

A arquitetura completa tem o seguinte formato (Figura 13 - Diagrama da arquitetura):

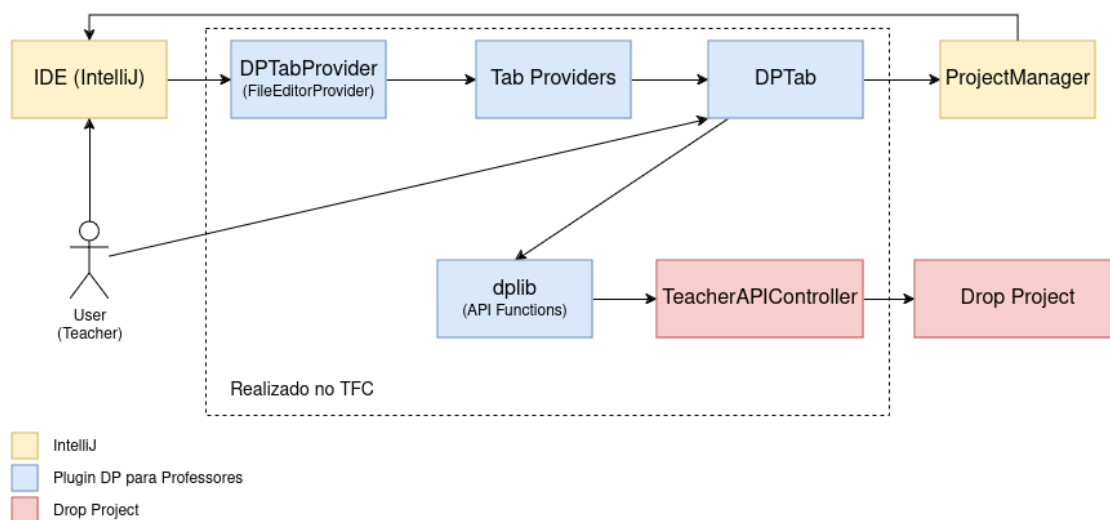


Figura 13 - Diagrama da arquitetura

## 5.2 Implementação

Uma demonstração do projeto desenvolvido neste TFC pode ser vista através dos seguintes URLs:

Versão	URL vídeo
Final	<a href="https://youtu.be/dm1HgdDNjtQ">https://youtu.be/dm1HgdDNjtQ</a>
Beta 2	<a href="https://youtu.be/Ay6lkdUf89o">https://youtu.be/Ay6lkdUf89o</a>
Beta 1	<a href="https://youtu.be/-sdRW8G6EbU">https://youtu.be/-sdRW8G6EbU</a>

## 5.3 Abrangência

A implementação da solução proposta abrangeu conhecimentos de diversas cadeiras do curso de Engenharia Informática:

- Interação Humano-Máquina: design da interface de utilizador, para que seja acessível ao maior número de utilizadores possível
- Computação Distribuída: design dos *endpoints* da API
- Linguagens de Programação 2: conhecimentos de programação orientada a objetos (OOP), para garantir uma maior flexibilidade do código fonte



## 6 Plano de testes e validação

Como o público-alvo deste *plugin* são utilizadores com conhecimentos avançados de tecnologia e com conhecimento já existente do *workflow* da interface web do DP, os testes que irão ser realizados serão de natureza exploratórios, em que caso o utilizador encontre *bugs* ou características não desejadas durante o seu uso normal, este deverá reportá-los, e estes irão ser considerados durante a fase de correção de *bugs* e de definição de requisitos.

### 6.1 Resultados

Legenda:

- PA – Pedro Alves
- BC – Bruno Cipriano
- RC – Rodrigo Correia

#### 6.1.1 Bugs reportados

Tabela 5 - Bugs reportados

Bug	Reportado por	Data	Corrigido?
Login não funciona com o servidor oficial	PA, BC, RC	11/06/2024	Sim
API responde com demasiada informação	PA	11/06/2024	Sim
O campo do <i>token</i> não o esconde (falha de segurança)	BC	15/06/2024	Sim
<i>Student History</i> não mostra erro quando há uma falha no pedido	RC	17/06/2024	Sim
O <i>Student History</i> não mostra os exercícios de forma cronológica, baseado na última submissão feita a este	PA	27/06/2024	Sim

#### 6.1.2 Problemas de usabilidade

Tabela 6 - Problemas de usabilidade reportados

Problema	Reportado por	Data	Corrigido?
URL do servidor do servidor oficial não cabe na caixa de login	BC	15/06/2024	Sim
Devia de ter um <i>link</i> para obter o <i>token</i> de autenticação	BC	15/06/2024	Sim
Login falhado não mostra a razão da falha	BC	15/06/2024	Sim
Título do menu do <i>plugin</i> (i.e. “DP”) é demasiado ambíguo	BC	15/06/2024	Sim
Título da janela de login (i.e. “DP - Login”) é demasiado ambíguo	BC	15/06/2024	Sim

Os detalhes de <i>login</i> não são guardados e apresentados na próxima vez que se faz <i>login</i>	PA	27/06/2024	Sim
O campo para o URL do servidor podia vir pré-preenchido com o URL do servidor do DEISI	PA	27/06/2024	Sim
A janela de <i>login</i> devia de fechar automaticamente assim que se tenta fazer <i>login</i>	PA	27/06/2024	Não
O <i>dashboard</i> tem demasiado conteúdo, tornando-o confuso	PA	27/06/2024	Não
Não há <i>feedback</i> entre o clique num aluno no <i>dashboard</i> e o carregamento do <i>Student History</i>	PA	27/06/2024	Sim
Os botões de navegação são facilmente confundíveis com os botões de avançar e retroceder de página, no ecrã do <i>Student History</i>	PA	27/06/2024	Não
As páginas deviam de caber no ecrã, sem a necessidade de fazer <i>scroll</i>	PA	27/06/2024	Não
Devia de haver um indicador da página onde se está atualmente	PA	27/06/2024	Sim
Os resultados dos testes no <i>build report</i> não são selecionáveis	PA	27/06/2024	Sim

## 7 Calendário

Nesta secção apresenta-se o calendário definido para a realização do trabalho. (Tabela 7)

A fase 1 corresponde à fase da criação de um protótipo funcional que permite aos professores executar o *workflow* descrito anteriormente no relatório.

A fase 2 corresponde à “afinação” e criação de funcionalidades resultantes do *feedback* dado pelos professores depois do uso do *plugin* num ambiente de trabalho real.

Tabela 7 - Calendário

Tópico	Data de início	Data de fim
Clonar repositórios	15/09/23	15/09/23
Protótipo de <i>login</i>	16/09/23	24/09/23
Protótipo do download de submissões	24/09/23	24/09/23
Protótipo da lista de exercícios	26/09/23	29/09/23
Protótipo da lista de submissões	29/09/23	30/09/23
Protótipo do <i>dashboard</i>	12/10/23	12/10/23
Protótipo do <i>build report</i>	31/10/23	08/11/23
Definição de requisitos - Fase 1	20/11/23	31/11/23
Implementação dos requisitos - Fase 1	1/12/23	05/02/24
Testes com professores	05/02/24	08/06/24
Definição de requisitos – Fase 2	23/04/24	29/04/24
Implementação dos requisitos - Fase 2	30/04/24	25/06/24
Correção de bugs - Fase 2	08/06/24	28/06/24
Oficialização no <i>GitHub</i> do Drop Project	A fazer mais tarde*	
Publicação no <i>Marketplace</i> do <i>IntelliJ</i>	A fazer mais tarde*	

\* A ser feita pelos professores, depois do TFC.

## 8 Resultados

A implementação final do trabalho encontra-se funcional, e cumpre a maioria dos requisitos elaborados.

As funcionalidades disponíveis nesta versão são as seguintes:

- *Download* de submissões
- Listagem, pesquisa, filtragem (por *tag* e nome) e ativação/desativação de exercícios
- Listagem, ordenação e pesquisa de submissões por grupo
- Listagem de submissões
- *Student History*
- Visualização do *build report*
- Marcação de submissões como “submissões finais”

## 9 Conclusão

Este trabalho permitiu o desenvolvimento de diversas competências, entre elas as capacidades de criação de *plugins* para o IDE *IntelliJ*, de criação de *endpoints* para um servidor *Spring Boot* e as capacidades de testagem com utilizadores e resposta a *feedback*.

Mesmo estando maioritariamente completo, seria possível no futuro a melhoria da experiência de utilizador, como evidenciado na tabela 6, e a implementação de funcionalidades não consideradas para o projeto.

## Anexos

### Progresso de trabalho

Tabela 8 - Tabela de progresso

Tópico	Data de início	Data de fim
Clonar repositórios	15/09/23	15/09/23
Protótipo de <i>login</i>	16/09/23	24/09/23
Protótipo do download de submissões	24/09/23	24/09/23
Protótipo da lista de exercícios	26/09/23	29/09/23
Protótipo da lista de submissões	29/09/23	30/09/23
Protótipo do <i>dashboard</i>	12/10/23	12/10/23
Protótipo do <i>build report</i>	31/10/23	08/11/23
Definição de requisitos - Fase 1	20/11/23	31/11/23
Implementação dos requisitos - Fase 1	1/12/23	05/02/24
Testes com professores*	15/04/24	22/04/24
Correção de bugs - Fase 1**	15/04/24	A definir mais tarde
Definição de requisitos – Fase 2*	23/04/24	29/04/24
Implementação dos requisitos - Fase 2*	30/04/24	A definir mais tarde
Correção de bugs - Fase 2*	A definir mais tarde	A definir mais tarde
Oficialização no <i>GitHub</i> do Drop Project	A definir mais tarde	
Publicação no <i>Marketplace</i> do <i>IntelliJ</i>	A definir mais tarde	

**Legenda:**

- \* - Concluído
- \*\* - Não realizado

Em comparação com a tabela definida no relatório intercalar do 2º semestre, o prazo dos testes com os utilizadores foi alargado e a fase de correção de problemas ocorreu numa única fase.

## Bibliografia

- [BCNP22] Bruno Pereira Cipriano, Nuno Fachada, Pedro Alves, Drop Project: An automatic assessment tool for programming assignments, 2022
- [JBIJ23] JetBrains s.r.o., IntelliJ Platform SDK, <https://plugins.jetbrains.com/docs/intellij>, acedido em Nov. 2023.
- [BLCK23] Block Inc., OkHttp, <https://square.github.io/okhttp>, acedido em Nov.2023
- [DIOG21] Diogo Casaca, Trabalho Final de Curso, [https://informatica.ulusofona.pt/wp-content/uploads/sites/57/2022/10/TFC\\_21\\_DEISI189.pdf](https://informatica.ulusofona.pt/wp-content/uploads/sites/57/2022/10/TFC_21_DEISI189.pdf), 2021.
- [AFMS18] André Filipe Monteiro da Silva, Raccode: Um Plugin de Avaliação Automática de Programas para o Eclipse, 2018
- [WBCT23] IDEAS Research Group Virginia Tech, <https://web-cat.github.io/projects/Web-CAT/EclipsePlugins/>, acedido em Nov. 2023.
- [AVLN24] AvaloniaUI OÜ, *The View Locator*, <https://docs.avaloniaui.net/docs/concepts/view-locator>, acedido em Jan. 2024.

## **Glossário**

LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso
IDE	<i>Integrated Development Environment</i> (Ambiente de desenvolvimento)
DP	Drop Project
AAT	<i>Automatic Assessment Tool</i>
API	<i>Application Programming Interface</i>