



**Desenvolvendo o pensamento abstrato
através da aprendizagem baseada em jogos**

Rui Miguel Loureiro Baptista

UMinho | 2023

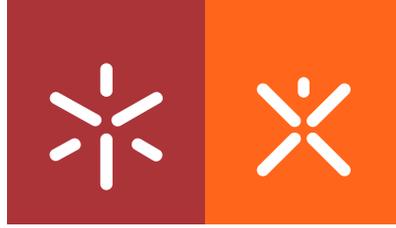


Universidade do Minho
Instituto de Educação

Rui Miguel Loureiro Baptista

**Desenvolvendo o pensamento abstrato
através da aprendizagem baseada em jogos**

outubro de 2023



Universidade do Minho
Instituto de Educação

Rui Miguel Loureiro Baptista

**Desenvolvendo o pensamento abstrato
através da aprendizagem baseada em jogos**

Relatório de Estágio
Mestrado em Ensino de Informática

Trabalho efetuado sob a orientação do
Doutor José Alberto Lencastre

DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

Este é um trabalho académico que pode ser utilizado por terceiros desde que respeitadas as regras e boas práticas internacionalmente aceites, no que concerne aos direitos de autor e direitos conexos.

Assim, o presente trabalho pode ser utilizado nos termos previstos na licença [abaixo](#) indicada.

Caso o utilizador necessite de permissão para poder fazer um uso do trabalho em condições não previstas no licenciamento indicado, deverá contactar o autor, através do RepositóriUM da Universidade do Minho.

Licença concedida aos utilizadores deste trabalho



Atribuição-NãoComercial

CC BY-NC

<https://creativecommons.org/licenses/by-nc/4.0/>

Agradecimentos

Os meus sinceros agradecimentos a todas as pessoas que desempenharam um papel fundamental na realização deste projeto de vida. Família, pelo apoio incondicional, compreensão e incentivo constante, a minha força para seguir em frente mesmo nos momentos mais difíceis. Aos meus amigos, que partilharam palavras e momentos de descontração que me permitiram recarregar energias. Aos meus colegas e professores, orientadores, que partilharam experiência e conhecimento, todos inestimáveis durante esta caminhada.

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

DESENVOLVENDO O PENSAMENTO ABSTRATO ATRAVÉS DA APRENDIZAGEM BASEADA EM JOGOS.

Resumo

A utilização de jogos educativos, através da ludificação, pode desempenhar um papel importante na motivação e envolvimento dos alunos no processo de aprendizagem, tornando-o mais cativante e estimulante. Através de mecânicas típicas de jogos, como por exemplo narrativas, desafios, recompensas, os alunos desenvolvem capacidades importantes para a resolução de problemas, colaboração e tomada de decisões. O objetivo da intervenção pedagógica relatada no presente relatório de estágio profissional do Mestrado em Ensino de Informática foi desenvolver o pensamento abstrato através da implementação de uma estratégia que convida os alunos a participarem ativamente no processo de aprendizagem através do uso de metodologias ativas e ludificação. Procurou-se promover o desenvolvimento do pensamento abstrato dos alunos e a compreensão dos conceitos associados ao paradigma de “Abstract Data Type”, bem como os mecanismos fundamentais da programação orientada a objetos. O referencial teórico apoia-se nas teorias construtivistas de Piaget e Vygotsky, que realçam a importância da interação do aluno com o ambiente e com os outros para a construção do conhecimento. Além disso, o Construcionismo de Seymour Papert oferece uma perspectiva que enfatiza a criação de artefactos, como jogos educativos, como instrumentos capazes de ativar a aprendizagem. Ao longo do relatório, serão detalhadas as metodologias utilizadas em sala de aula, assim como a sua aplicação prática no contexto do curso de Informática e Tecnologias Multimédia. Os resultados obtidos permitem demonstrar o impacto positivo da estratégia pedagógica na compreensão dos conceitos de programação orientada a objetos e no desenvolvimento do pensamento abstrato dos alunos. Espera-se que este relatório forneça perspectivas valiosas sobre a utilização de jogos e da ludificação na promoção de uma aprendizagem mais participativa e significativa. Para além disso, espera-se que o projeto sirva como base sólida para futuras intervenções pedagógicas, que procurem integrar de forma adequada a tecnologia e a ludificação no processo educativo e melhorar a qualidade de ensino e a aprendizagem dos alunos.

Palavras Chave: Construtivismo, Construcionismo, Programação Orientada a Objetos, Ludificação, *Abstract Data Type* (ADT).

DEVELOPING ABSTRACT THINKING THROUGH GAME-BASED LEARNING.

Abstract

The use of educational games, through gamification, can play a significant role in motivating and engaging students in the learning process, making it more captivating and stimulating. Through typical game mechanics such as narratives, challenges, and rewards, students develop important problem-solving, collaboration, and decision-making skills. The objective of the pedagogical intervention reported in this professional internship report for the master's in computer science education was to develop abstract thinking through the implementation of a strategy that invites students to actively participate in the learning process through the use of active methodologies and gamification. We aimed to promote the development of students' abstract thinking and their understanding of concepts associated with the 'Abstract Data Type' paradigm, as well as the fundamental mechanisms of object-oriented programming. The theoretical framework is based on the constructivist theories of Piaget and Vygotsky, emphasizing the importance of the student's interaction with the environment and others for knowledge construction. Furthermore, Seymour Papert's constructionism provides a perspective that emphasizes the creation of artifacts, such as educational games, as instruments capable of activating learning. Throughout the report, the methodologies used in the classroom will be detailed, as well as their practical application in the context of the Computer Science and Multimedia Technologies course. The results obtained demonstrate the positive impact of the pedagogical strategy on the understanding of object-oriented programming concepts and the development of students' abstract thinking. It is expected that this report will provide valuable insights into the use of games and gamification in promoting more participatory and meaningful learning. Furthermore, it is expected that the project will serve as a solid foundation for future pedagogical interventions that seek to integrate technology and gamification into the educational process properly and improve the quality of teaching and student learning.

Keywords: Constructivism, Constructionism, Object-Oriented Programming, Gamification, Abstract Data Type (ADT).

Índice

Introdução	1
Questão de investigação	2
Objetivos	2
Organização do relatório	2
1. Referencial Teórico	4
1.1. Construtivismo na Educação: Aprendizagem Significativa e Participativa	4
1.2. Jogos e Aprendizagem: Integração do Construcionismo na Educação.....	5
1.3. O Pensamento Abstrato - Abstract Data Type (ADT).....	6
1.3.1. Programação Orientada a Objetos	7
1.4. Ludificação	7
2. Caracterização e Contexto.....	8
2.1. O Meio.....	8
2.2. A Escola.....	9
2.3. Oferta Formativa	10
2.4. Caracterização da Turma	11
3. Enquadramento Curricular	11
3.1. O Curso	12
3.2. A Disciplina.....	13
3.2.1. Objetivos de Aprendizagem	14
3.2.2. Conteúdos Programáticos	15
3.3. Elenco temático da intervenção	16
3.4. Avaliação Curricular	16

4. Plano de intervenção	18
4.1. Ferramentas Utilizadas.....	18
4.2. Planificação	22
4.3. Restrições.....	22
4.4. Aulas Assistidas	23
4.5. Intervenção Pedagógica.....	29
4.5.1. Resumo do projeto	35
4.5.2. Atividades realizadas no âmbito da intervenção	41
4.5.2.1. Conversa com Especialistas	41
4.5.2.2. Envelope secreto.....	45
4.5.3. Projeto Final.....	51
4.6. Ludificação com estratégia motivacional	55
4.6.1. Quadro de progresso do projeto	56
4.6.2. Certificado de participação na Intervenção Pedagógica	63
4.7. Avaliação da intervenção	65
4.8. Inovação	68
4.9. Outras atividades e participação	69
5. Conclusão, limitações e recomendações	70
5.1. Conclusão.....	70
5.2. Limitações	71
5.2. Recomendações.....	72
Referências	73
Apêndices.....	75

Resumo do projeto	75
Planificação de aulas.....	80
Exemplo Ficha de trabalho	92
Projeto Final.....	95
Anexos.....	98
Anexo 1 – Autorização publicação do nome da instituição.	98
Anexo 2 – Manual Unity	99
Anexo 3 – Manual Git.....	116

Lista de figuras

FIGURA 1 - ESTRATÉGIA PEDAGÓGICA PROPOSTA - CRIAÇÃO DE JOGOS EDUCATIVOS POR ALUNOS (FONTE: MARTINS & OLIVEIRA, 2018, P. 4)	6
FIGURA 2 - MAPA VILA NOVA DE GAIA (FONTE: GOOGLE MAPS).....	9
FIGURA 3 - O CURSO: PLANO DE ESTUDOS (FONTE: COLGAIA.PT)	12
FIGURA 4 - FERRAMENTAS: UNITY (FONTE: UNITY3D.COM)	19
FIGURA 5 - FERRAMENTAS: MICROSOFT VISUAL STUDIO (FONTE: VISUALSTUDIO.MICROSOFT.COM)	19
FIGURA 6 - FERRAMENTAS: MICROSOFT TEAMS (FONTE: TEAMS.MICROSOFT.COM)	20
FIGURA 7 - FERRAMENTAS: MICROSOFT PLANNER (FONTE: TASKS.OFFICE.COM).....	20
FIGURA 8 - FERRAMENTAS: GITHUB (FONTE: GITHUB.COM)	20
FIGURA 9 - FERRAMENTAS: DB BROWSER (FONTE: GITHUB.COM/SQLITEBROWSER)	20
FIGURA 10 - FERRAMENTAS: SQLITE (FONTE: SQLITE.ORG)	21
FIGURA 11 - LINGUAGENS: C SHARP (FONTE: SQLITE.ORG).....	21
FIGURA 12 - LINGUAGENS: C SHARP (FONTE: STONEBRANCH.COM)	21
FIGURA 13 - AVALIAÇÃO PRÉVIA: GOSTAS DE JOGAR?.....	26
FIGURA 14 - AVALIAÇÃO PRÉVIA: QUANTAS HORAS POR DIA JOGAS, EM MÉDIA?.....	26
FIGURA 15 - AVALIAÇÃO PRÉVIA: OPORTUNIDADE DESENVOLVIMENTO JOGOS NO COLÉGIO	27
FIGURA 16 - AVALIAÇÃO PRÉVIA: MOTIVAÇÃO CURSO ITM.....	28
FIGURA 17 - AVIAÇÃO PRÉVIA: IMPORTÂNCIA DOS JOGOS NA APRENDIZAGEM DE CONCEITOS DE POO	28
FIGURA 18 - UNITY: CONCESSÃO DE LICENÇA EDUCACIONAL GRATUITA.	30
FIGURA 19 - UNITY: ABORDAGENS E METODOLOGIAS ATIVAS	30
FIGURA 20 - IMPLEMENTAÇÃO AGILE SCRUM: TAREFA	31
FIGURA 21 - IMPLEMENTAÇÃO AGILE SCRUM: QUADRO	32
FIGURA 22 -GESTÃO DOCUMENTAL DO PROJETO	33
FIGURA 23 - MANUAL DO PROJETO	33
FIGURA 24 - VERSIONAMENTO DE SOFTWARE.....	34
FIGURA 25 - VÍDEOS DEMONSTRATIVOS	35
FIGURA 26 - PROJETO: VISTA GERAL DA CENA.	36
FIGURA 27 - PROJETO: INTERAÇÃO DO UTILIZADOR.	38
FIGURA 28 - PROJETO: COR SELECIONADA	38
FIGURA 29 - PROJETO: UNIDADES E RECURSOS.....	39
FIGURA 30 - CONVERSA COM ESPECIALISTAS: VIDEOCONFERÊNCIA	42
FIGURA 31 - CONVERSA COM ESPECIALISTAS: MOTIVAÇÃO	42
FIGURA 32 - CONVERSA COM ESPECIALISTA: AVALIAÇÃO SESSÃO.....	43
FIGURA 33 - CONVERSA COM ESPECIALISTAS: INTERESSE DESPERTADO	43
FIGURA 34 - CONVERSA COM ESPECIALISTAS: INTERESSE NA UTILIZAÇÃO PESSOAL.....	44
FIGURA 35 - CONVERSA COM ESPECIALISTAS: INTERESSE NA UTILIZAÇÃO PROFISSIONAL	44
FIGURA 36 - ENVELOPE SECRETO: MENSAGEM SECRETA TEAM LEADER	45

FIGURA 37 - ENVELOPE SECRETO: TEAM LEADER.....	46
FIGURA 38 - ENVELOPE SECRETO: FORMAÇÃO DOS GRUPOS DE TRABALHO	47
FIGURA 39 - ESTATÍSTICA ENVELOPE SECRETO: GRAU DE MOTIVAÇÃO.....	47
FIGURA 40 - ESTATÍSTICA ENVELOPE SECRETO: AVALIAÇÃO DA AULA.....	48
FIGURA 41 - ESTATÍSTICA ENVELOPE SECRETO: SESSÃO NO ESCLARECIMENTO DE DÚVIDAS.....	48
FIGURA 42 - ESTATÍSTICA ENVELOPE SECRETO: COMPREENSÃO DOS PILARES ANTES DA SESSÃO.	49
FIGURA 43 - ESTATÍSTICA ENVELOPE SECRETO: AVALIAÇÃO DEPOIS DA SESSÃO	49
FIGURA 44 - ESTATÍSTICA ENVELOPE SECRETO: FOI POSSÍVEL AL APLICAÇÃO DOS CONCEITOS NA PRÁTICA.	50
FIGURA 45 - ESTATÍSTICA ENVELOPE SECRETO: APLICAÇÃO DOS CONCEITOS NA PRÁTICA.	50
FIGURA 46 - PROJETO FINAL: DIAGRAMA DO PROJETO.	51
FIGURA 47 - PROJETO FINAL: MENU LOGIN.....	52
FIGURA 48 – PROJETO FINAL: MENU DE REGISTO	53
FIGURA 49 - PROJETO FINAL: LISTA DE RESULTADOS	54
FIGURA 50 - PROJETO FINAL: CLASSE BD	55
FIGURA 51 - LUDIFICAÇÃO: CONQUISTADAS OBTIDAS POR UM ALUNO DA TURMA.....	59
FIGURA 52 - LUDIFICAÇÃO: QUADRO DE PROGRESSO.....	60
FIGURA 53 - LUDIFICAÇÃO: ANÁLISE E ESTATÍSTICA DE RESULTADOS OBTIDOS.....	61
FIGURA 54 - LUDIFICAÇÃO: CERTIFICADO DE PARTICIPAÇÃO NA INTERVENÇÃO PEDAGÓGICA	64
FIGURA 55 – LUDIFICAÇÃO: ENTREGA DE CERTIFICADOS AOS ALUNOS.....	64
FIGURA 56 - AVALIAÇÃO DA INTERVENÇÃO: OBJETIVOS.....	65
FIGURA 57 - AVALIAÇÃO DA INTERVENÇÃO: COMPREENSÃO DOS CONCEITOS	65
FIGURA 58 - AVALIAÇÃO DA INTERVENÇÃO: APLICAÇÃO DOS CONCEITOS APRENDIDOS.	66
FIGURA 59 - AVALIAÇÃO DA INTERVENÇÃO: ORGANIZAÇÃO, CLAREZA, RECURSOS E APOIO.	67
FIGURA 60 - PERSISTÊNCIA DE DADOS ENTRE CENAS.....	105
FIGURA 61 - UNITY PROCESSO DE SERIALIZAÇÃO E DESSERIALIZAÇÃO DE DADOS	108

Índice de tabelas

TABELA 1 - CONTEÚDOS PROGRAMÁTICOS: DOMÍNIOS E SUBDOMÍNIOS	15
TABELA 2 - RECOLHA DE DADOS REALIZADA EM DOCUMENTO PRÓPRIO	23
TABELA 3 - LUDIFICAÇÃO: PONTUAÇÃO QUADRO DE PROGRESSO	56
TABELA 4 - LUDIFICAÇÃO: ETIQUETAS E PONTOS POR CONQUISTA.....	56
TABELA 5 - LUDIFICAÇÃO: AVALIAÇÃO DAS FICHAS DE TRABALHO REALIZADAS PELOS ALUNOS.....	58
TABELA 6 - LUDIFICAÇÃO: REGISTO PONTUAÇÃO OBTIDA POR ALUNO.....	62

Lista de gráficos

GRÁFICO 1 - LUDIFICAÇÃO: INFORMAÇÃO GERAL RESULTADOS OBTIDOS.	63
--	----

Lista de Siglas e Abreviaturas

ADT - Abstract Data Type

ITM - Informática e Tecnologias Multimédia

POO - Programação Orientada a Objetos

TDM – Tecnologias e Desenvolvimento Multimédia

TP - Técnicas de Programação

IEBD - Implementação e Exploração de Base de Dados

NEE - Necessidades Educativas Especiais

SQL - Structured Query Language

SSAO - Screen Space Ambient Occlusion

IDE - Ambiente de Desenvolvimento Integrado

DB4S - DB Browser para SQLite

EAD - Ensino à Distância

PBL - Project Based Learning

Introdução

O plano de intervenção proposto é elaborado no âmbito do estágio profissional do Mestrado em Ensino de Informática, a realizar no Curso com Planos Próprios de Informática e Tecnologias Multimédia do Colégio de Gaia. O objetivo desta intervenção é ensinar conceitos de informática relacionados com a Programação Orientada a Objetos (POO), utilizando uma estratégia diferenciada para promover o pensamento abstrato em alunos do 11.º ano. Pretendeu-se envolver os alunos de forma mais significativa na construção do seu conhecimento ao longo do processo de aprendizagem, desenvolvendo o pensamento abstrato através da aprendizagem baseada em jogos (Gee, 2010; Osório, 2015).

O Pensamento Abstrato (Abstract Data Type), segundo Barradas et al. (2019), "é considerado uma aptidão fundamental no século XXI, pois permite aumentar a capacidade analítica das crianças nas diversas áreas do conhecimento, bem como promover outras competências, como o pensamento algorítmico, lógico e dimensionável (Phillips, 2009; Resnick, 2012; MIT, 2011)".

Jogos são aplicações utilizadas no ensino da programação de forma eficiente, aumentando o compromisso e a motivação dos alunos (Yildirim & Ozdener, 2021). Unity é um motor de jogo multiplataforma com um Ambiente de Desenvolvimento Integrado (IDE) desenvolvido pela Unity Technologies. É usado para desenvolver jogos multiplataformas (Peterpunk, 2013).

Colocar os alunos no papel de criadores é uma das possíveis abordagens à integração de jogos na educação, suportada pelo Construcionismo (Papert & Harel, 1991; Kafai & Burke, 2015), que defende que a construção de conhecimento é mais efetiva quando aliada à criação de artefactos (Martins & Oliveira, 2018). É com base na ideia do construcionismo de Papert que se pretende revestir esta intervenção, conduzindo os alunos através de um processo de "learning-by-making" (Papert & Harel, 1991; Osório, 2015).

O objetivo foi implementar uma estratégia pedagógica que convida à participação e envolvimento dos alunos no processo de ensino e de aprendizagem, permitindo-lhes deste modo a construção do seu próprio conhecimento. Com vista à promoção da autonomia, afigura-se interessante visitar a argumentação apresentada pelo filósofo francês Jacques Rancière (2004) no seu livro *Le Maître Ignorant*, onde conta a história (verdadeira) de Joseph Jacotot, um professor francês do início do século XIX que descobriu que é possível ensinar o que não se sabe, desde que se saiba criar desafios e inspirações que levem os alunos a aprender por si próprios (Figueiredo, 2016).

Com base nestes pressupostos, pretende-se demonstrar a capacidade de simulação de conceitos associados ao Paradigma Orientado a Objetos (POO) com recurso à aprendizagem baseada em jogos. Os conceitos do paradigma orientado a objetos (POO) devem ser representados graficamente, explicados

na interface, promovendo a abstração. Posteriormente, à medida que os conceitos são apresentados, os alunos são convidados a realizar atividades que permitirão interagir com a interface do protótipo e vincular os conceitos aprendidos às tarefas a serem desenvolvidas.

Questão de investigação

É possível desenvolver o pensamento abstrato dos alunos através de um ensino centrado na aprendizagem baseada em jogos?

Objetivos

- ✓ Compreender o paradigma de “Abstract Data Type” como introdução à metodologia de desenvolvimento orientada aos objetos.
- ✓ Compreender os conceitos e mecanismos básicos da programação orientada aos objetos.
- ✓ Desenvolver nos alunos competências de resolução de problemas na utilização do *Engine* de jogos *Unity* e ferramentas de programação associadas.
- ✓ Desenvolver nos alunos o pensamento abstrato através de uma aprendizagem baseada em jogos.
- ✓ Promover o desenvolvimento do conhecimento de forma partilhada e em equipa.
- ✓ Avaliar o impacto das estratégias no processo de aprendizagem dos alunos.

Organização do relatório

O relatório está estruturado em cinco capítulos, apresenta diferentes aspetos do projeto de intervenção pedagógica. A seguir, é apresentada uma breve descrição de cada capítulo:

- Referencial Teórico: Neste capítulo, fundamentam-se as decisões pedagógicas adotadas no projeto. São apresentadas as teorias construtivistas, construcionistas, pensamento abstrato e os conceitos de ludificação, que servem como base para a implementação da estratégia de ensino.
- Contexto Escolar: O segundo capítulo fornece informações sobre o contexto educacional em que a intervenção pedagógica foi realizada. São apresentados detalhes sobre a escola, os alunos envolvidos e os recursos disponíveis.
- Enquadramento Curricular: O terceiro capítulo explora o enquadramento curricular dos alunos, com foco nos conteúdos relacionados com a programação orientada a objetos e pensamento abstrato. São descritas as disciplinas e temas específicos abordados no curso.

- Plano de Intervenção: O quarto capítulo concentra-se na descrição detalhada da estratégia pedagógica implementada. São apresentadas as ferramentas utilizadas, atividades realizadas, materiais e recursos produzidos para promover a aprendizagem baseada em jogos e o desenvolvimento do pensamento abstrato.
- Conclusões: O quinto capítulo traz as conclusões tiradas da realização do projeto de intervenção pedagógica. São discutidos os resultados obtidos, o impacto da estratégia no processo de aprendizagem dos alunos e as reflexões pessoais sobre o projeto.

1. Referencial Teórico

No âmbito deste trabalho, com vista à realização de uma intervenção pedagógica eficaz foi colocada a seguinte questão: É possível desenvolver o pensamento abstrato dos alunos através de um ensino centrado na aprendizagem baseada em jogos? Com o objetivo de responder a esta questão, nesta secção do relatório são apresentadas teorias e metodologias relevantes para a compreensão do paradigma de "Abstract Data Type" (ADT) como introdução aos conceitos da programação orientada a objetos (POO). A teoria construtivista na Educação, enfatiza a construção ativa do conhecimento pelos alunos e valoriza a participação no processo de ensino e aprendizagem. A integração do Construcionismo na Educação é abordada como estratégia que coloca os alunos no papel de criadores, permitindo-lhes construir o seu próprio conhecimento ao interagir com o mundo. Esta abordagem, apoiada pelo Construcionismo de Seymour Papert, enfatiza a importância da criação de artefactos e utilização ativa dos conceitos estudados na aquisição de capacidades cognitivas fundamentais.

O Pensamento Abstrato promove a capacidade de análise dos alunos em diversas áreas do conhecimento. Neste contexto, a compreensão do paradigma de ADT e dos conceitos de POO requer o desenvolvimento do pensamento lógico e algorítmico.

A utilização de elementos e mecânicas de jogos em contextos educacionais, promovidos pela ludificação, aponta ser o caminho para a obtenção de resultados positivos na motivação e envolvimento dos alunos no projeto.

Através deste referencial teórico, procuro estabelecer uma base sólida para a proposta apresentada, a combinação de teorias como o Construtivismo, Construcionismo e Ludificação pode proporcionar uma experiência de aprendizagem significativa aos alunos, permitindo-lhes construir o seu próprio conhecimento, aprender a resolver problemas de forma ativa e criativa.

1.1. Construtivismo na Educação: Aprendizagem Significativa e Participativa

O Construtivismo tem sido grandemente influenciado pelas teorias e pesquisas no desenvolvimento humano, uma grande influência na ascensão do construtivismo foi a teoria e a pesquisa no desenvolvimento humano, especialmente as teorias de Piaget e Vygotsky (Schunk, 2012). Para os construtivistas, o conhecimento é obtido através da descoberta que o aluno/criança tem com os outros e com o mundo. Assim, ensinar é mais que a transmissão de conhecimentos (Costa Estrada, 2013).

Silva (citado por Estrada, 2013, p. 29) fundamentado em Piaget, destaca que:

Os indivíduos constroem o seu conhecimento do mundo, sendo esta a principal força motriz do movimento construtivista. Os psicólogos cognitivistas e desenvolvimentistas, Piaget em

particular, encaram a aprendizagem como um processo dinâmico no qual os alunos constroem o seu próprio conhecimento ao interagir com o mundo.

Fosnot (citado por Barbosa, 2006, p.1) ressalta que, embora o construtivismo não constitua uma teoria de ensino, aponta para a adoção de uma abordagem da educação radicalmente diferente daquela que normalmente é utilizada na maior parte das escolas. O construtivismo enfatiza currículos integrados e faz com que os professores usem os materiais de forma que os alunos se envolvam ativamente (Schunk, 2012, p. 235).

Estas perspectivas, reforçam a importância do Construtivismo como uma abordagem pedagógica que valoriza a construção do conhecimento pelos próprios alunos, estimulando a participação ativa e promovendo uma aprendizagem mais significativa e aliciente.

Além disso, estudos como os de (Oliveira L. , 2019) têm destacado o potencial dos jogos como veículos de aprendizagem, motivação e envolvimento. Uma abordagem que tem mostrado resultados positivos é colocar os alunos no papel de criadores de jogos, esta estratégia vai permitir aos alunos a construção do seu próprio conhecimento,

1.2. Jogos e Aprendizagem: Integração do Construcionismo na Educação

A utilização de Jogos no ensino não é novidade, mas a criação de jogos pelos próprios alunos é algo ainda em investigação (Cruz, Bento, & Lencastre, 2020) ainda os mesmos autores consideram que “os jogos além de constituírem uma fonte de prazer e entretenimento, podem apresentar um grande potencial didático e funcionar como estímulo para o desenvolvimento do pensamento matemático, nomeadamente o raciocínio lógico.”

Usar jogos na educação ajuda a superar preconceitos e vícios diários, estimulando vivências positivas. Através deles, ultrapassamos limitações pessoais ou sociais, descobrindo novos caminhos para melhorar (Gomes & Filho, 2008).

Construcionismo é uma reconstrução teórica a partir do construtivismo piagetiano, proposta por Seymour Papert (1994 e 1986), originalmente em 1980. Papert concorda com Piaget (1976), em que a criança é um “ser pensante” e construtora de suas próprias estruturas cognitivas, mesmo sem ser ensinada.

Papert & Harel e Kafai & Burke (citados por Cruz, Bento, & Lencastre, 2020, p. 3) consideram que colocar alunos no papel de criadores é uma das possíveis abordagens à integração de jogos na educação, suportada pelo Construcionismo.

Uma estratégia pedagógica suportada pelo Construcionismo, teoria de aprendizagem que descende do Construtivismo de Piaget é apresentada por (Martins & Oliveira, 2018). A estratégia alternativa, é

considerada indispensável para motivar e ensinar alunos em contexto de risco de insucesso escolar e foi utilizada para dar resposta ao problema em estudo que abordaram, colocando alunos no papel de criadores de jogos educativos. A figura 1 ilustra a estratégia pedagógica proposta pelas autoras.



Figura 1 - Estratégia pedagógica proposta - Criação de Jogos educativos por alunos (Fonte: Martins & Oliveira, 2018, p. 4)

De acordo com (Martins & Oliveira, 2018),

Papert (1971) reforça a importância da criação de artefactos, de objetos com os quais pensar, e a relevância de que as crianças possam manusear, alterar e criar para adquirirem a sensação de mestria, o sentido de conhecimento aplicado, e uma imagem autoconfiante e realista de si mesmas enquanto agentes intelectuais (idem).

Construcionismo visa maximizar a aprendizagem, com o mínimo de ensino. Fortalece a aprendizagem, valorizando construções mentais do sujeito apoiada em suas próprias construções no mundo. (Nunes & Santos, 2013).

1.3. O Pensamento Abstrato - Abstract Data Type (ADT)

O Pensamento Abstrato (Abstract Data Type), segundo (Barradas, Lencastre, Soares, & Valente, 2019), “é considerado uma aptidão fundamental no século XXI, pois permite aumentar a capacidade analítica das crianças nas diversas áreas do conhecimento”, bem como promover outras competências outras tais como o pensamento algorítmico, lógico e dimensionável (Phillips, 2009; Resnick, 2012; MIT, 2011)”. O objetivo dos Tipos de Dados Abstratos (Abstract Data Types - ADTs) é possibilitar a construção modular de sistemas de software através da composição de componentes cujo comportamento individual seja bem compreendido (Connor, 2003). Segundo o mesmo autor, esta abordagem para a construção de software tem uma série de vantagens. A especificação pode ser fornecida numa linguagem diferente da

linguagem de programação usada para implementá-la.

1.3.1. Programação Orientada a Objetos

A programação orientada a objetos pode ser considerada como um método declarativo de programação, no sentido de que fornece um modelo de computação (expresso pela metáfora de encapsulamento e passagem de mensagens) que é independente de um modelo de implementação específico (Eliens, 2000).

No paradigma POO, do ponto de vista dos programadores, uma linguagem orientada a objetos deve suportar três características explícitas muito importantes. Estes conceitos, são usados extensivamente para modelar os problemas do mundo real quando estamos a tentar resolvê-los em programas orientados a objetos (Tykhomyrov, 2002). Esses três conceitos são:

- encapsulamento,
- herança,
- polimorfismo.

O autor, considera como característica implícita a abstração. Usada para especificar novos tipos de dados abstratos, ou ADT para abreviar.

1.4. Ludificação

A ludificação, também conhecida como Gamificação, tradução do inglês “Gamification”, é um conceito amplamente conhecido, com diversos estudos e pesquisas académicas. Envolve a aplicação de elementos e mecânicas de jogos em contextos não relacionados a jogos, com o intuito de envolver, motivar e melhorar a experiência do utilizador (Deterding S. , Dixon, Khaled, & Nacke, 2011). Nos últimos anos, a ludificação tem ganho destaque em várias áreas, desde a educação aos negócios e até na saúde, proporcionando novas abordagens para a resolução de problemas e envolvimento das pessoas.

A origem do termo remonta a 1980, quando o professor da Universidade de Essex, Richard Bartle, o utilizou para definir o processo de “tornar algo que não é um jogo num jogo” (Neto, 2015). Atualmente, a ludificação abrange diversas técnicas presentes em videojogos (GLT - Game-like techniques) aplicadas em contextos não relacionados a jogos, como educação (GBL - Game-Based Learning), áreas sociais e de saúde, marketing, entre outras (Hakulinen & Zaric, citados por Costa, 2022, p. 24).

Para Kapp (citado por Costa, 2022, p. 24), Ludificação consiste na utilização da mecânica, estética e pensamento baseado em jogos, para aumentar o envolvimento entre pessoas e/ou área de intervenção.

Em contextos académicos, a motivação e interação entre formador e formando, promovendo assim a aprendizagem e resolvendo problemas da abordagem tradicional de ensino.

Além disso, Kapp (citado por Cabral, 2022, p. 33) considera que no fundo, a ludificação é um processo que permite a construção de ambientes de aprendizagens, já que ao aplicarmos elementos de jogo em ambiente de aprendizagem será possível envolver os alunos, fomentar a ação e promover a aprendizagem.

Na ludificação do ensino, à medida que os alunos vão cumprindo com os objetivos, são premiados. A utilização de elementos de ludificação, que é uma estratégia de envolvimento dos alunos nas atividades através de incentivos que estimulem o compromisso com a escola, de forma lúdica no processo de ensino-aprendizagem estará sempre presente (Carvalho, 2018).

A ludificação surge como uma alternativa promissora no campo da educação, pode permitir uma experiência de aprendizagem envolvente e impactante para os alunos, ao mesmo tempo, promove o desenvolvimento de competências fundamentais para o crescimento pessoal e profissional dos alunos.

2. Caracterização e Contexto

Este capítulo apresenta uma caracterização do contexto educativo envolvido no projeto, focando-se na escola onde a intervenção foi realizada, a oferta formativa e a descrição da turma do 11.º Ano. A escola, localizada em Vila Nova de Gaia, é de referência na região norte de Portugal. Com uma história rica e expansão significativa ao longo das décadas, possui um Projeto Educativo próprio, assente em valores e educação constante em diálogo com pais e encarregados de educação. A oferta formativa abrange desde o Ensino Pré-Escolar até o Ensino Secundário, com cursos aprovados por planos próprios e focados na formação integral dos alunos. A turma do 11.º Ano, que será objeto da intervenção, é composta por 28 alunos interessados na área da informática, empenhados e com desejo de prosseguir os estudos na área. A seguir, serão detalhados os aspetos relevantes deste contexto educativo, fornecendo um panorama completo para a compreensão do desenvolvimento do projeto.

2.1. O Meio

A escola está localizada na cidade de Vila Nova de Gaia, município pertencente à Área Metropolitana do Porto e ao distrito do Porto, localizado na Região Norte de Portugal.

Atualmente, é o terceiro município mais populoso do país, com mais de 300 000 habitante e o mais populoso da região norte. É conhecido internacionalmente pelas suas empresas de vinhos do Porto,

indústria automóvel, vidreira e de componentes eletrónicos e artistas culturais. A cidade acolhe milhares de turistas por ano.

A cidade disponibiliza uma vasta rede de transportes e vias da comunicação. Destacam-se as Autoestradas, Linhas do Caminho-de-Ferro, Metro, Eixos concelhios estruturantes.



Figura 2 - Mapa Vila Nova de Gaia (Fonte: Google Maps)

2.2. A Escola

A escola nasce em 1933, na “Quinta do Trancoso”, nascia o Colégio Externato de Gaia por iniciativa do então Bispo do Porto, D. António Meireles. Localizada na freguesia de Mafamude, Vila Nova de Gaia, a referida Quinta foi doada à Diocese do Porto por D. Maria Margarida Guimarães e Silva.

Em 1934 foi concedido alvará de funcionamento ao Colégio Externato de Gaia – hoje Colégio de Gaia – sendo nomeado para diretor o Sr. Pe. Nédio de Sousa.

Nas décadas de 1960 e 1970 assistiu-se a uma significativa expansão das instalações, transformando o Colégio de Gaia no maior Colégio do país ao nível de instalações escolares e desportivas e de espaços exteriores, situação esta que se mantém até aos dias de hoje.

Atualmente e sob a direção do Sr. Pe. António Manuel Barbosa Ferreira, o Colégio de Gaia possui um Projeto Educativo próprio, de forma a servir uma população escolar com cerca de 1500 alunos que frequenta níveis de ensino desde os 3 anos até ao 12.º ano de escolaridade.

A escola Tendo como entidade titular a Diocese do Porto e sendo uma instituição de ensino católica, o Colégio de Gaia assegura uma educação sólida e assente em valores que fomentam um diálogo

persistente e continuado com os pais e encarregados de educação centrando as nossas preocupações nos alunos e na sua formação e educação constante.

2.3. Oferta Formativa

A oferta formativa do colégio de Gaia¹ integra, acompanha e educa as crianças num projeto que valoriza a sua formação com respeito pela sua individualidade numa sociedade em permanente mudança, onde o seu, ritmo, percurso e conquistas são valorizados, oferecendo-lhes um ambiente de aprendizagens significativas que favoreça e potencie as suas inteligências múltiplas.

Ensino Pré-Escolar

Neste contexto, a Educação Pré-Escolar no Colégio de Gaia – Escola Católica reúne um conjunto de recursos humanos, físicos e pedagógicos que proporcionam condições para o desenvolvimento e a aprendizagem por parte das crianças.

Ensino Básico

Recorrendo a um quadro de docentes e auxiliares de ação educativa com elevada experiência na área, assim como a um conjunto de espaços educativos diversos (salas de aula interativas, laboratórios, biblioteca, pavilhões gimnodesportivos, campos desportivos ao ar livre e piscina), consegue garantir uma educação sólida, onde as experiências educativas integradas são privilegiadas diariamente.

Cursos com planos Próprios

O Colégio de Gaia é um estabelecimento de ensino particular e cooperativo que ministra cursos do ensino secundário, com planos próprios, aprovados pela Portaria n.º 272/2019, de 27 de agosto,

O ensino secundário do Colégio de Gaia, com uma oferta de 13 cursos com planos próprios, afirma-se como uma escolha de primeira qualidade nos universos educativos, quer português, quer europeu. São cursos com evidentes vantagens para quantos os frequentam, desde a integralidade, harmonia e solidez da sua formação geral, científica e técnica até à excelente articulação com o ensino superior seja pela pluralidade de opções que possibilitam, seja pela especificidade das disciplinas estudadas, passando pela capacidade adquirida de, aos dezoito anos, os alunos estarem aptos a desempenharem uma profissão.

¹ Informação disponível no site do Colégio de Gaia, <https://www.colgaia.pt/>

2.4. Caracterização da Turma

A turma do 11.º Ano onde o presente projeto se desenvolveu, faz parte deste curso e a intervenção foi realizada na disciplina de Técnicas de Programação (TP). É constituída por 28 alunos, sendo duas raparigas e 26 rapazes. A média de Idades da turma é de 16,5 anos. Estão matriculados à totalidade das disciplinas, apresentando 0 alunos com dupla matrícula e 2 alunos com Necessidades Educativas Especiais (NEE) diagnosticados com disléxica, usufruem de mediadas de apoio à disciplina de Português. No final do 10.º Ano, e segundo o relatório do diretor de turma, os alunos foram considerados muito interessados e empenhados a todas as disciplinas. Mostram grande interesse pela área da informática e pretende prosseguir estudos. A integração na instituição foi considerada fácil, apesar das condicionantes provocadas pela atual crise pandémica.

3. Enquadramento Curricular

O presente capítulo aborda o enquadramento curricular da disciplina de Técnicas de Programação no curso com plano próprio de Informática e Tecnologias Multimédia, ministrado no Colégio de Gaia. O curso visa fornecer formação qualificada e abrangente na área da informática e multimédia, preparando os alunos para desempenhar funções técnicas exigentes nesses campos. A disciplina de Técnicas de Programação, presente nos 10.º e 11.º anos, possui um carácter prático e aborda desde noções básicas de programação até à programação orientada a objetos e acesso a bases de dados.

Neste contexto, a disciplina tem como objetivo desenvolver competências essenciais para os futuros profissionais de Informática e Tecnologias Multimédia, abrangendo desde a resolução sistemática de problemas através de algoritmos até ao desenvolvimento de aplicativos complexos, incluindo a programação orientada ao objeto em ambiente gráfico e o acesso a bases de dados. Através de diversos projetos, os alunos são estimulados a aplicar conhecimentos e habilidades, aprimorando sua capacidade de resolução de problemas, trabalho em grupo e pensamento crítico e criativo.

A avaliação é realizada de forma globalizante, considerando os conhecimentos, capacidades e valores dos alunos. Além dos instrumentos de avaliação tradicionais, como testes escritos e práticos, fichas de trabalho e relatórios, são realizados projetos práticos que permitem aos alunos demonstrar suas competências de forma mais abrangente, incentivando a reflexão e o debate sobre os resultados obtidos. Dessa forma, este capítulo apresenta uma visão geral da disciplina de Técnicas de Programação no contexto do curso de Informática e Tecnologias Multimédia, bem como os objetivos de aprendizagem e conteúdos programáticos que norteiam o desenvolvimento da disciplina. Essa caracterização proporciona uma base sólida para a compreensão da intervenção pedagógica proposta, que será detalhada posteriormente.

3.1. O Curso

O curso com plano próprio de Informática e Tecnologias Multimédia tem como finalidade satisfazer a necessidade de formação qualificada, para exercício de funções de exigente valor técnico, no âmbito da informática e da multimédia.

Disciplinas	Carga Horária Semanal (minutos)			Carga Horária (horas) Ciclo de Formação (a)
	10.º Ano	11.º Ano	12.º Ano	
Componente de Formação Geral				
Português	180	180	240	339
Língua Estrangeira I, II ou III (c)	180	120	–	173
Filosofia	120	180	–	172
Educação Física	120	120	–	138
Subtotal	600	600	240	822
Componente de Formação Científica				
Matemática A	240	240	360	474
Física e Química A	240	360	–	344
Subtotal	480	600	360	818
Componente de Formação Tecnológica				
Moral, Ética e Deontologia	60	60	60	102
Aplicações Informáticas	120	–	–	70
Fundamentos e Arquitetura de Computadores	180	120	–	173
Técnicas de Programação	300	180	–	277
Implementação e Exploração de Bases de Dados	–	180	120	168
Programação Internet	–	–	300	165
Tecnologias e Desenvolvimento Multimédia	–	–	180	103
Projeto Tecnológico	–	60	60	67
Subtotal	660	600	720	1 125
Formação em Contexto de Trabalho (e)	–	105	295	400
Educação Moral e Religiosa (d) (e)	(27)	(27)	(27)	(81)
Total	1 740	1 800	1 320	3 165

(a) Carga horária do ciclo de formação a gerir pela escola, no âmbito da sua autonomia pedagógica.

(b) Componente desenvolvida com o contributo de disciplinas e componentes de formação.

(c) O(A) aluno(a), se tiver estudado apenas uma língua estrangeira no ensino básico (LE I), iniciará obrigatoriamente uma segunda língua estrangeira no ensino secundário (LE II). No caso de o(a) aluno(a) iniciar uma língua, tomando em conta as disponibilidades do estabelecimento de ensino, poderá cumulativamente dar continuidade à Língua Estrangeira I como disciplina facultativa, com aceitação expressa do acréscimo de carga horária. Aos alunos oriundos de sistemas educativos estrangeiros, aplica-se o disposto no artigo 11.º.

(d) Disciplina de frequência facultativa.

(e) Carga horária anual.

Figura 3 - O Curso: Plano de Estudos (Fonte: ColGaia.pt)

A formação pluridisciplinar destes alunos, nomeadamente, a formação tecnológica em laboratórios informáticos, permite-lhes que se adaptem facilmente a diversos cenários profissionais. De uma forma autónoma e de acordo com as especificações técnicas definidas, estão aptos a executar atividades de desenvolvimento, conceção, comercialização, manutenção de software; manutenção e comercialização

de equipamentos informáticos; instalação, reparação e manutenção de redes informáticas, introduzindo modificações e operações de acertos; criação e manutenção de bases de dados; desenvolvimento de produtos multimédia; criação de páginas Web e design e desenvolvimento de jogos informáticos; desenvolvimento de aplicações para dispositivos móveis, instalação e operação com sistemas operativos móveis. A figura 3 ilustra o plano de estudos do curso com plano próprio de ITM.

3.2. A Disciplina

A disciplina de Técnicas de Programação², insere-se nos 10.º e 11.º anos da componente de formação tecnológica do curso com Plano Próprio de Informática e Tecnologias Multimédia.

Esta disciplina, predominantemente prática, incide sobre as técnicas de programação destinadas ao desenvolvimento de aplicações informáticas. Neste sentido, no 10.º ano, é estudada a resolução de problemas de forma algorítmica, seguindo-se a abordagem da programação baseada numa linguagem estruturada, tendo como finalidade a implementação das soluções algorítmicas. No 11.º ano, evolui-se para a programação orientada ao objeto, integrando a interação com bases de dados.

Assim, a disciplina de Técnicas de Programação contribui diretamente para o desenvolvimento de competências necessárias ao desempenho da atividade profissional do técnico de Informática e Tecnologias Multimédia, sobretudo ao nível:

- do desenvolvimento de aplicações informáticas;
- da programação orientada ao objeto;
- da programação de aplicações cliente-servidor.

A disciplina de Técnicas de Programação constitui um espaço privilegiado de elaboração de relatórios escritos e apresentações orais; de realização de um conjunto de pesquisas sobre os conceitos associados à programação; de avaliação e validação de informação recolhida; de organização da informação recolhida; de elaboração e apresentação de conteúdos; de criação de conteúdos em ambiente digital; de desenvolvimento de diversos projetos e programas onde são convocados a apresentar soluções; de trabalho direto com vários equipamentos, ambientes de programação e plataformas de desenvolvimento computacional, contribuindo para o desenvolvimento das áreas de competências do Perfil dos Alunos à Saída da Escolaridade Obrigatória, nas mais diversas dimensões.

Além disso, os alunos consolidam hábitos de planeamento das etapas do trabalho, identificando os requisitos técnicos, condicionalismos e recursos para a concretização de experiências / projetos,

² Programa de Disciplina de Técnicas de Programação do Curso de Informática e Tecnologias Multimédia

utilizando para o efeito manuais das linguagens de programação, textos de apoio e a consulta dos ambientes de desenvolvimento integrado. Durante o desenvolvimento de programas em ambiente computacional, os discentes aplicam as metodologias de desenvolvimento de programas, para atingir um objetivo ou chegar a uma solução fundamentada, adequando a implementação à linguagem de programação em estudo.

Destacam-se o desenvolvimento de competências ao nível de linguagens e textos, com a utilização de linguagens verbais e não-verbais; ao nível da compreensão e interpretação de factos, opiniões, conceitos e pensamentos, quer oralmente, quer por escrito; de competências associadas à área de informação e comunicação, de competências associadas ao raciocínio e resolução de problemas; de competências associadas ao saber técnico e tecnológico e do pensamento crítico e criativo.

Procura ainda promover competências ao nível do relacionamento interpessoal, ao proporcionar a realização de trabalhos em grupo, nos projetos/experiências, onde os alunos juntam esforços para alcançar objetivos, valorizando a diversidade de perspetivas sobre as questões em causa, tanto lado a lado como através de meios digitais.

Destaca-se ainda o desenvolvimento de projetos, alguns deles inter e transdisciplinares, em particular no âmbito da componente de formação Cidadania e Desenvolvimento, definindo estratégias criativas e inovadoras, sendo os alunos encorajados a desenvolver e a pôr em prática os valores da responsabilidade e integridade; da excelência e exigência; da curiosidade, reflexão e inovação; e do cumprimento das normas de segurança, higiene, saúde e proteção ambiental.

3.2.1. Objetivos de Aprendizagem

No final da disciplina Técnicas de Programação, os alunos deverão saber e ser capazes de:

- aplicar as técnicas de programação;
- distinguir as diferentes etapas do processamento de dados;
- reconhecer o papel do programador no desenvolvimento de um programa informático;
- efetuar a análise sistemática de um problema;
- elaborar algoritmos;
- distinguir os tipos de dados com vista ao processamento computacional;
- realizar e interpretar fluxogramas;
- reconhecer as estruturas de programação de uma linguagem de alto nível;
- aplicar as diferentes estruturas de dados da linguagem de programação;
- operar com vetores, matrizes e cadeia de caracteres;

- estruturar um programa, adequando a solução ao paradigma da linguagem em estudo;
- reconhecer e aplicar funções pré-definidas da linguagem de programação;
- implementar rotinas e várias funções pré-definidas para acesso aos dispositivos de entrada e saída;
- desenvolver programas e rotinas para acesso e armazenamento de informação em ficheiros tradicionais; desenvolver programas e rotinas para acesso e armazenamento de informação em bases de dados;
- desenvolver aplicativos que integram vários programas com a finalidade de criar um “sistema de software”.

3.2.2. Conteúdos Programáticos

O conteúdo da disciplina de Técnicas de Programação é bienal, lecionado nos 10º e 11º ano e tem a Tabela 1 apresenta a distribuição de conteúdos por unidade curricular:

Tabela 1 - Conteúdos programáticos: Domínios e subdomínios

10º	Noções básicas de programação	Noção sobre técnicas de programação. Processamento de dados. Linguagens de programação, programador e programas. Análise sistemática de um problema. Algoritmos.
	Introdução à programação	Pseudocódigo. Fluxogramas. Teste do modelo de solução. Estrutura de dados simples - vetores e matrizes.
	Programação numa linguagem estruturada	Apresentação da linguagem. Elementos da linguagem. Estrutura da linguagem. Entrada e saída. Estruturas de decisão e seleção. Estruturas de controlo.
	Estruturas de dados simples	Vetores (arrays). Cadeia de caracteres (strings). Estruturas (structures).
	Procedimentos e funções	Estrutura. Parametrização. Funções predefinidas. Recursividade.
	Desenvolvimento em ambiente gráfico	Interface gráfica. Métodos e eventos. Objetos de controlo de ações. Objetos de manipulação de dados simples. Objetos de lista de valores.

11°	Programação orientada ao objeto	Conceitos básicos. Estrutura de dados avançados. Desenvolvimento de programas numa linguagem orientada ao objeto.
	Programação orientado ao objeto em ambiente gráfico	Ambiente de desenvolvimento. Desenvolvimento assistido. Objetos básicos. Construção de protótipos de interfaces, manipulando as propriedades dos elementos. Interação entre os elementos constituintes da interface através de eventos. Construção de aplicações simples.
	Acesso a base de dados	Tecnologias de acesso a bases de dados. Acesso e manipulação de dados. Desenvolvimento de aplicações de acesso a bases de dados.

O programa é constituído por 9 módulos, como se pode verificar na Tabela 1, com uma carga horária total de 277 horas. Os alunos têm 3 horas semanais no 10º Ano (175 Horas) e duas horas no 11º Ano (102 Horas).

3.3. Elenco temático da intervenção

Para a realização da intervenção pedagógica foram lecionadas duas unidades curriculares, ministradas no segundo e terceiro período do 11º ano, Unidade 2: Programação orientado ao objeto em ambiente gráfico com 18 horas letivas previstas e unidade 3: Acesso a bases de dados com 9 horas letivas.

3.4. Avaliação Curricular

A avaliação é globalizante, incidindo sobre os diferentes conteúdos da disciplina de Técnicas de Programação, dimensões de conhecimentos, capacidades e valores do aluno. Embora a avaliação formativa seja, essencialmente, um processo contínuo e interativo, por questões de operacionalização, ao nível da disciplina de Técnicas de Programação, distinguem-se vários momentos. Assim, durante o processo de ensino-aprendizagem, através de uma interação contínua, será possível clarificar com os alunos os níveis de exigência, definir e desenvolver medidas de reajustamento, com base na interpretação fundamentada das dificuldades e dos êxitos, permitindo, assim, uma maior diferenciação das aprendizagens. Finalmente, depois de um período mais longo, a avaliação formativa permitirá fazer um balanço das aprendizagens, possibilitando reorganizar as atividades conforme as diferentes necessidades dos alunos. Neste sentido, durante o período letivo, recorre-se ao desenvolvimento de projetos práticos, seguidos de apresentações orais, com debate e reflexão sobre o seu funcionamento e os resultados

obtidos. São, igualmente, fornecidos aos discentes um conjunto de tutoriais, para que possam, de forma autónoma, realizar os exercícios associados dos conteúdos abordados na disciplina. Após a realização dos exercícios, apresentam as suas soluções que são debatidas e analisadas.

O processo de avaliação, entre outros, inclui os seguintes instrumentos:

- grelha de registo de atividades individuais;
- grelhas de observação de atividades de grupo/projetos;
- grelhas de observação de apresentação de trabalhos;
- grelha de registo de realização de trabalhos de casa;
- testes de avaliação escritos e práticos;
- fichas de trabalho individuais ou em grupo;
- trabalhos práticos;
- relatórios.

4. Plano de intervenção

Percebemos muitas vezes, que a maioria dos professores encontram dificuldades em planejar as suas aulas, devido ao fato de muitas vezes não terem formação teórica metodológica necessária para compreender a verdadeira importância do ato de planejar na sua prática pedagógica (Santos & Perin, 2013). Para (Oliveira D. A., 1997) o ato de planejar exige aspetos básicos a serem considerados. Um primeiro aspeto é o conhecimento da realidade daquilo que se deseja planejar, quais as principais necessidades que precisam ser trabalhadas; para que quem as planeia evidencie.

Em síntese, fica evidente que o planeamento adequado das aulas é um fator determinante para o sucesso da prática pedagógica dos professores. A ausência de formação teórico-metodológica nessa área pode dificultar a conceção de aulas eficazes que atendam às necessidades dos alunos. Nesse sentido, torna-se fundamental adquirir competências sólidas de planeamento, compreendendo a realidade da sala de aula, identificando as principais áreas a serem trabalhadas.

Ao adotar uma abordagem estruturada e informada no planeamento das aulas, poderei otimizar o meu desempenho pedagógico e oferecer uma experiência de aprendizagem mais enriquecedora aos alunos. Além disso, ao estar ciente das necessidades e desafios específicos que os alunos enfrentam, posso adaptar estratégias de ensino de forma mais eficaz, garantindo uma aprendizagem significativa.

A valorização do ato de planejar, aliada ao compromisso de contínua atualização e melhoria profissional, possibilitará enfrentar os desafios da sala de aula de forma mais assertiva, proporcionando uma educação mais inclusiva, enriquecedora e transformadora para todos os envolvidos.

Assim, antes de iniciar a minha intervenção pedagógica, empenhei-me em realizar um profundo trabalho de investigação sobre as temáticas abordadas, especialmente no contexto da Programação Orientada a Objetos (POO). O objetivo principal desse esforço, foi capacitar-me para aplicar uma estratégia diferenciada com a colaboração da turma e professor cooperante, regente da disciplina, para a promoção do pensamento abstrato através da aprendizagem baseada em jogos.

4.1. Ferramentas Utilizadas

As ferramentas utilizadas no projeto tiveram duas funções em particular. As Ferramentas Unity, Microsoft Visual Studio, SQLite e DB Browser foram utilizadas no desenvolvimento do jogo, nas unidades curriculares lecionadas. Já as Ferramentas Microsoft Teams e Microsoft Planner foram utilizadas no âmbito da gestão de conteúdos e gestão do projeto. A utilização de ferramentas pagas neste projeto não foi uma opção, a escola o Colégio de Gaia é parceiro da Microsoft pelo que tem licenciamento e privilegia a utilização no âmbito escolar e profissional. Estas ferramentas podem ser substituídas por software de código aberto sem comprometer o desenvolvimento do plano de intervenção supervisionado.

Unity

Unity, também conhecido como Unity 3D, é um motor de jogo 3D / 2D e um Ambiente de Desenvolvimento Integrado (IDE) criado pela Unity Technologies. Unity é o software líder global da indústria de jogos.

A primeira versão foi anunciada na *Apple Worldwide Developers Conference* em 2005 e foi lançada a 8 de junho de 2005. A versão inicial foi limitada a Mac OS, tanto na criação como publicação de jogos. Incluía renderizador OpenGL orientado para *shader*, física *Novodex* (agora *PhysX*), suporte a áudio e linguagem C#. Nas versões 1.x foi adicionado suporte para permitir que os jogos fossem executados em ambiente Windows e “navegadores web” através de um plugin. (Unity, 2022)

Unity permite:

- Especificação de configurações de compactação e de resolução de textura para cada plataforma que o jogo suporta,
- Fornece suporte para mapeamento de colisões,
- Mapeamento de reflexão,
- Mapeamento parallax/NormalMap,
- Ecrã de oclusão espaço ambiente Screen Space Ambient Occlusion (SSAO),
- Sombras dinâmicas utilizando mapas de sombra,
- Programação nas linguagens C# e JavaScript;
- Capacidade de desenvolvimento de jogos para múltiplas plataformas;

Microsoft Visual Studio

Microsoft Visual Studio é um ambiente de desenvolvimento integrado (IDE) desenvolvido pela Microsoft e lançado pela primeira vez em 1997. Especialmente dedicado ao .NET Framework e às linguagens Visual Basic, C, C++, C# (o que possibilita a integração do Unity), e F#. É também um produto de desenvolvimento na área web através da plataforma ASP.NET. (Wikipédia, 2022)



Figura 4 - Ferramentas: Unity
(Fonte: unity3d.com)



Figura 5 - Ferramentas:
Microsoft Visual Studio
(Fonte: visualstudio.microsoft.com)

Microsoft Teams

Microsoft Teams é uma plataforma unificada de comunicação e colaboração que combina publicações, videoconferências, gestão documental (permite trabalho colaborativo) e integração de aplicações. O serviço integra do pacote de produtividade Office 365 e apresenta extensões para integração de produtos que não são da Microsoft (Wikipédia, 2022).



Figura 6 - Ferramentas: Microsoft Teams (Fonte: teams.microsoft.com)

Microsoft Planner

Microsoft Planner é uma aplicação leve, móvel e baseada na Web que incluída na maioria dos pacotes de produtividade Office 365 com subscrição empresarial. Com o Planner, equipas podem criar quadros, atribuir tarefas, conversar sobre tarefas e ver gráficos do progresso da equipa. O Planner pode ser utilizado a partir do Microsoft Teams e do Microsoft SharePoint. (Wikipédia, 2022)



Figura 7 - Ferramentas: Microsoft Planner (Fonte: tasks.office.com)

GitHub

GitHub é uma plataforma de hospedagem de ficheiros, especialmente código, online, que usa como base o mais expansivo Git, que lhe fornece funcionalidades de arquivamento e controlo de versões. É uma plataforma muito utilizada por programadores, pois a tecnologia Git incorporada permite uma sincronização do progresso de um projeto em diferentes máquinas muito fácil, e como tal permite uma maior versatilidade no desenvolvimento de um projeto como este. (Wikipédia, 2022)



Figura 8 - Ferramentas: GitHub (Fonte: github.com)

DB Browser for SQLite

O DB Browser para SQLite (DB4S) é uma ferramenta visual de código aberto de alta qualidade para criar, projetar e editar arquivos de base de dados compatíveis com SQLite.

O DB4S foi desenvolvido para utilizadores e programadores que desejam criar, pesquisar e editar bases de dados. (Wikipédia, 2022)



Figura 9 - Ferramentas: DB Browser (Fonte: github.com/sqlitebrowser)

SQLite

É um Motor de Base de Dados escrito na linguagem de programação C. Não é uma aplicação independente, em vez disso, é uma biblioteca que os programadores de software incorporam nas suas aplicativos. Pertence à família das Base de Dados embutidas. É o mecanismo de banco de dados mais amplamente utilizado, é usado por vários dos principais navegadores da Web, sistemas operativos, smartphones e outros sistemas integrados. (Wikipédia, 2022)



Figura 10 - Ferramentas: SQLite (Fonte: sqlite.org)

Linguagens abordadas

No desenvolvimento do projeto foram abordadas duas linguagens essenciais. C#, utilizada no desenvolvimento de scripts, linguagem lecionada na disciplina de Técnicas de Programação (TP), onde os alunos nesta fase do ano letivo já construíram aplicações em consola no paradigma imperativo Procedural e Orientado a Objetos. Outra linguagem abordada foi o SQL, faz parte do currículo dos alunos na disciplina de Implementação e Exploração de Base de Dados (IEBD), com o objetivo de trabalhar com os alunos a interdisciplinaridade escolar.

C Sharp

C Sharp também conhecida por C# é uma linguagem de programação de alto nível aplicada na generalidade das aplicações desenvolvidas na atualidade, multiparadigma, de tipagem forte foi desenvolvida pela Microsoft em 2000. Embora seja maioritariamente baseada em C++, é fortemente influenciada por linguagens como Object Pascal e, principalmente, Java. (Wikipédia, 2022)



Figura 11 - Linguagens: C Sharp (Fonte: sqlite.org)

SQL

Structured Query Language (SQL) em português Linguagem de Consulta Estruturada é uma linguagem específica de domínio usada em programação e projetada para gerir dados armazenados em Sistemas de Gestão de Base de Dados Relacionais (SGBDR) ou para processamento de fluxo em Sistemas de Gestão de Fluxo de Dados Relacional (SGFDR). É particularmente útil para lidar com dados estruturados, ou seja, dados que incorpora relações entre entidades e variáveis. (Wikipédia, 2022)



Figura 12 - Linguagens: C Sharp (Fonte: stonebranch.com)

4.2. Planificação

Para a realização da planificação das aulas, o professor cooperante forneceu a planificação anual da disciplina, que contém as orientações necessárias em termos de conteúdos, objetivos, estratégias adotadas, recursos utilizados e gestão de tempo. Os módulos de Desenvolvimento POO em ambiente Gráfico e Acesso a Base de Dados foram selecionados para a intervenção, totalizam 20 horas por turno, o que corresponde a uma carga horária de 3 horas semanais por turno, divididas em duas aulas, uma de 60 minutos teórico-prática e outra de 120 minutos maioritariamente prática.

A intervenção teve uma duração aproximada de 5 semanas, totaliza 40 horas, das quais, 9 horas foram assistidas pelo orientador, Doutor José Alberto Lencastre. A planificação foi elaborada com base nos pressupostos do documento fornecido pelo professor cooperante e adaptada à estratégia de intervenção. O documento resultante desse trabalho não foi totalmente finalizado antes do início da intervenção e sofreu alterações devido às restrições existentes durante a pandemia. A adaptação contínua foi necessária para garantir a eficácia e o sucesso da intervenção, considerando as limitações impostas pelo contexto pandémico.

4.3. Restrições

Devido ao período pandémico, alguns alunos foram obrigados a cumprir períodos de isolamento profilático de acordo com as leis em vigor. Para garantir a continuidade do ensino durante os períodos de isolamento, o colégio adotou o Ensino à Distância (EAD) através da aplicação Teams.

As restrições motivadas pela doença apresentaram desafios ao trabalho a realizar em sala de aula, uma vez que o professor teve de se dividir entre auxiliar os alunos em sala de aula e prestar assistência aos alunos que estavam a acompanhar remotamente. Como estratégia para superar a dificuldade, todos os alunos foram convidados a colaborar. Os alunos presentes foram incentivados a participar e apoiar os colegas que estavam em casa, desde que estes últimos estivessem em condições físicas para o fazer.

Outra estratégia adotada para enfrentar o desafio foi a disponibilização de recursos adicionais, como manuais, fichas de trabalho e vídeos explicativos, para acesso remoto. Esta estratégia permitiu aos alunos acompanhar as aulas de acordo com a sua disponibilidade, utilizando os recursos disponíveis (Computador, Smartphone) e aceder aos conteúdos no horário mais conveniente.

Estas medidas foram tomadas para garantir que todos os alunos tivessem acesso ao ensino, mesmo durante os períodos de isolamento ou em circunstâncias que os impediam de estar presencialmente na sala de aula. A flexibilidade e a colaboração de todos os envolvidos foi fundamentais para enfrentar os desafios impostos pelo contexto pandémico e assegurar a continuidade do processo de ensino-aprendizagem.

4.4. Aulas Assistidas

O período de aulas assistidas teve início a 28 de fevereiro de 2022 e foi concluído a 28 de abril de 2022. No total foram assistidas 28 horas letivas. Primeiramente, acompanhei o professor cooperante na unidade de POO. Conheci a turma e familiarizei-me com os alunos. Como estratégias adotadas, o professor cooperante utiliza o método expositivo para explicar sucintamente os conceitos envolvidos, comparando numa primeira fase a programação procedimental com a programação orientada a objetos, exibe pequenos exemplos demonstrativos, acompanhando-os de uma breve descrição dos vários conceitos envolvidos. Os elementos da linguagem são introduzidos gradualmente recorrendo sempre a exemplos ilustrativos da sua utilização. Na componente prática a introdução dos elementos é auxiliada pela realização de fichas de trabalho de complexidade crescente. O desenvolvimento nesta fase foi predominantemente realizado em modo consola com recurso ao Ambiente de Desenvolvimento Integrado (IDE) Microsoft Visual Studio. As fichas de trabalho propostas aos alunos têm objetivos e tempos de realização definidos.

Durante este período de análise foram utilizadas duas estratégias de recolha de dados. Inicialmente comecei por criar um documento e registar as principais dificuldades sentidas pelos alunos. A Tabela 2 exemplifica a recolha de dados realizada.

Tabela 2 - Recolha de dados realizada em documento próprio

Número	Nome	Turno	Aula 1 28/02/2022	Aula 2 07/03/2022
1	Aluno A	T1	Gestão de Tempo. Reconhece e Aplica Conceito de Override; Métodos e Atributos e acesso como get; set. Herança; Polimorfismo não sabe ou não deu. Em consola	
2	Aluno B	T2		
3	Aluno C	T1	Teve dificuldade inicial em perceber o conceito de class; Ainda não usou a implementar nos próximos exercícios; A interpretação do problema causa algum atraso na realização dos exercícios. Percebe e aplica o conceito de Herança	
4	Aluno D	T2		

5	Aluno E	T1	Não aplicou o Override, percebe e cria classes, propriedades e métodos. Polimorfismo ainda não usou. Objeto? Dificuldade em perceber conceitos, mas aplica. Tem dificuldade na interpretação de problemas.	
6	Aluno F	T2		
7	Aluno G	T1	Interpretação de problemas. Override? Apresentar caso prático. Paradigma Orientado a Objetos... Polimorfismo ainda não abordaram... aplica conceitos, mas não fala a linguagem em conceitos.	
8	Aluno H	T2		
9	Aluno I	T1	Override e overloading. Objeto, Herança a sintaxe da herança...	
10	Aluno J	T2		
11	Aluno L	T1		Dúvidas de conceitos não tem. Sintaxe, vem com a prática. Override, virtual
12	Aluno M	T2		
13	Aluno N	T1		
14	Aluno O	T2		
15	Aluno P	T1	Dificuldade em quase tudo. Não é capaz de interpretar problemas. Precisa de praticar.	
16	Aluno Q	T2		
17	Aluno R	T1		Conceitos: Percebe e aplica com algum tempo. Get Set. Virtual?
18	Aluno S	T2		
19	Aluno U	T1		Conceitos? Depende, Sabe para que servem, reconhece, mas não tem a certeza de quando usar.
20	Aluno V	T2		
21	Aluno X	T1		

22	Aluno Y	T2		
23	Aluno Z	T1		Conceitos? Percebe, mas tem dificuldades em aplicar.
24	Aluno AA	T2		
25	Aluno AB	T1		Relativamente fácil, mas a sintaxe da linguagem...
26	Aluno AC	T2		
27	Aluno AD	T1		Conceitos? Não sabe muito bem quando usar os conceitos.
28	Aluno AE	T2		

Com base nos dados registados, foi possível recolher informações relevantes relacionadas com a programação orientada a objetos:

- Conhecimento e dificuldades: A maioria dos alunos apresenta conhecimento básico dos conceitos de programação orientada a objetos, como criar classes, propriedades e métodos. No entanto, alguns alunos ainda enfrentam dificuldades em aplicar conceitos mais avançados, como polimorfismo, *override* e herança.
- Gestão do tempo: A turma evidencia dificuldades na gestão do tempo, com alguns alunos atrasados em relação ao planeamento, o que pode afetar o progresso e desempenho nas atividades propostas.
- Interpretação de problemas: Cerca de 10 alunos encontram dificuldades na interpretação de problemas relacionados com programação orientada a objetos. Essa capacidade é fundamental para resolver problemas e aplicar os conceitos lecionados nas aulas práticas.

A estratégia de recolha de dados foi adaptada para cumprir com as restrições atuais e garantir a participação de todos os alunos. Com o objetivo de obter uma visão abrangente, optei por criar um questionário para recolher informações sobre a motivação e conhecimento prévio dos alunos, considerando-o uma forma eficaz de compreender melhor as suas necessidades e interesses.

O questionário foi utilizado para avaliar a motivação dos alunos em relação à instituição de ensino e ao desenvolvimento de jogos. Procurei identificar os elementos do processo de ensino e aprendizagem baseado em jogos que despertam maior interesse entre os estudantes.

Esta abordagem permitiu recolher dados quantitativos e qualitativos, forneceu uma perspetiva mais abrangente da receptividade dos alunos em relação à estratégia pedagógica adotada. Com base nos resultados obtidos no questionário, foram realizadas análises e reflexões para melhorar a intervenção e proporcionar uma experiência de aprendizagem mais enriquecedora, alinhada com as necessidades dos alunos.

- **Gostas de jogar?**

A figura 13 ilustra a resposta dos alunos à primeira questão colocada.

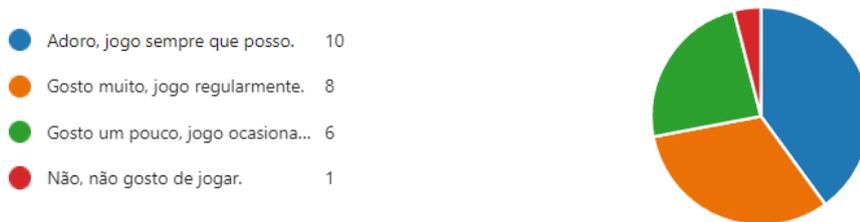


Figura 13 - Avaliação prévia: Gostas de Jogar?

40% dos alunos da turma adora jogar, 32% joga com regularidade, 24% joga ocasionalmente e apenas 4% não gosta ou não joga. A partir destes dados, pode-se inferir que a turma, em geral, tem uma atitude positiva em relação aos jogos. O facto de a grande maioria de alunos que adora jogar, jogar regularmente ou ocasionalmente sugere que os jogos são populares e interessantes para a maioria dos alunos. Sugere que com base nos dados recolhidos, é possível que a intervenção possa ser bem recebida e despertar o interesse dos alunos, considerando a atitude positiva da maioria em relação aos jogos. Pode criar uma atmosfera propícia para a participação e interesse dos alunos na intervenção.

- **Quantas horas jogas por dia, em média?**

A figura 14 ilustra os resultados obtidos na segunda questão:



Figura 14 - Avaliação prévia: Quantas horas por dia jogas, em média?

48% dos alunos inquiridos jogam 1 a 2 horas por dia. 16% 2 a 3 horas dia, 16% 3 a 4 a 6 horas diárias, 12% mais de 4 horas e apenas 8% dos alunos não joga. Com base nos dados fornecidos, podemos

concluir que a maioria dos alunos inquiridos joga uma quantidade moderada de horas por dia, com uma média de aproximadamente 2,28 horas de jogo dia. Para alunos do 11º ano esta média pode ter impacto na responsabilidade académica, equilíbrio entre vida académica e pessoal, efeitos na saúde e bem-estar assim como no controlo do tempo.

- **Sabias que no Colégio de Gaia, os alunos do curso de ITM têm a oportunidade de desenvolver jogos?**

A figura 15 apresenta os resultados obtidos na terceira questão.



Figura 15 - Avaliação prévia: Oportunidade desenvolvimento jogos no colégio

100% dos alunos responderam que sim, estão reconhecem a oportunidade. Com base nos dados fornecidos, posso concluir que todos dos alunos do curso de Informática e Tecnologias Multimédia (ITM) do Colégio de Gaia, estão informados da oportunidade. Este pode ser considerado um indicador positivo, pois mostra que a informação sobre a oportunidade de desenvolvimento de jogos foi bem comunicada e alcançou todos os alunos do curso.

A consciencialização dos alunos pode ter implicações importantes na motivação e valor do curso. Estarem conscientes da possibilidade, pode aumentar o interesse em aprender e aplicar conceitos relevantes aos desenvolvimentos de jogos.

Resumindo, a resposta positiva e unânime dos alunos indica que a oportunidade de desenvolver jogos é conhecida e valorizada pela turma, o que pode trazer benefícios significativos para o ensino e aprendizagem dos conceitos a lecionar.

- **Desenvolver jogos motivou a tua escolha pelo Colégio de Gaia e pelo curso de Informática e Tecnologias Multimédia (ITM) para este ciclo de estudos?**

A figura 16 ilustra os resultados obtidos na quarta questão.



Figura 16 - Avaliação prévia: Motivação curso ITM.

44% dos alunos escolheu o Colégio de Gaia e curso de Informática e Tecnologias Multimédia (ITM) para este ciclo de estudos motivados pela possibilidade de desenvolver jogos. Isso significa que quase metade dos alunos considerou a possibilidade de desenvolver jogos como um fator importante na sua decisão de ingressar no estabelecimento de ensino e curso em particular. 40% dos alunos responderam que não consideram esse tipo de experiência relevante para o seu ensino, indicando que para eles a oportunidade de fazer jogos não foi um fator determinante na escolha do Colégio de Gaia ou do curso. 16% dos alunos responderam "talvez", mostra que ainda têm dúvidas a respeito da relevância e oportunidade de desenvolver jogos. É interessante notar que nenhum aluno respondeu que nunca pensou nesse assunto e não tem uma opinião formada sobre a possibilidade de desenvolver jogos como parte de sua formação. Este indicador, sugere que todos os alunos têm alguma consideração ou conhecimento sobre a importância ou têm interesse em desenvolver jogos, mesmo que em diferentes graus.

- Na tua opinião, qual a importância dos jogos na aprendizagem de conceitos de Programação Orientada a Objetos?**

A figura 17 ilustra a quinta e última questão colocada.

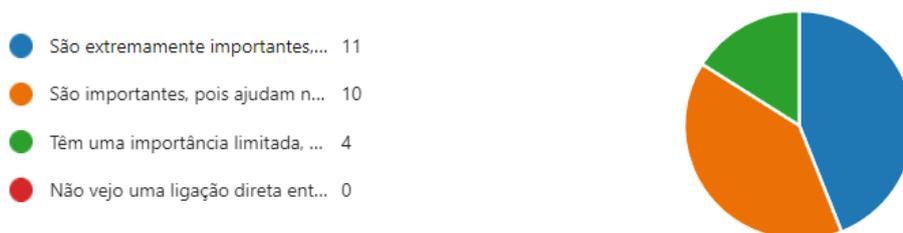


Figura 17 - Avaliação prévia: Importância dos jogos na aprendizagem de conceitos de POO

44% dos alunos acreditam que os jogos são extremamente importantes na aprendizagem dos conceitos, consideram que os jogos tornam os conceitos mais práticos e concretos. Sugere que para este grupo de alunos, os jogos desempenham um papel fundamental ao proporcionar uma experiência prática e concreta para aprender POO. A abordagem pode ajudar a consolidar o entendimento dos conceitos e

torná-los mais acessíveis e envolventes.

40% dos alunos consideram que os jogos são importantes, pois auxiliam a compreensão dos conceitos teóricos. A resposta indica que para alunos, os jogos podem ser uma ferramenta valiosa na ilustração e aplicação dos conceitos teóricos abordados em sala de aula, podem ser considerados facilitadores do processo de ensino e aprendizagem.

16% dos alunos reconhecem importância limitada dos jogos na aprendizagem de conceitos de POO, pois acreditam que os conceitos podem ser aprendidos de outras formas. Este grupo de alunos pode preferir outras abordagens educativas ou considerar que os jogos são apenas uma das várias maneiras de aprender os conceitos teóricos.

Nenhum aluno respondeu que não vê uma ligação direta entre jogos e a aprendizagem de conceitos de POO. Sugere que, entre os alunos que responderam à pesquisa, todos reconhecem de alguma forma a relevância dos jogos neste contexto.

No geral, os dados mostraram que os jogos são reconhecidos pelos alunos como uma ferramenta educacional relevante, especialmente na aprendizagem de conceitos de POO. No entanto, é importante considerar as diferentes perspectivas e necessidades dos alunos no planeamento da intervenção pedagógica, garantir que a abordagem é adequada e motivante para todos os alunos.

4.5. Intervenção Pedagógica

A intervenção pedagógica teve a duração de aproximadamente 5 semanas com início no dia 2 de maio de 2022 e conclusão a 7 de junho de 2022 num total de 40 horas letivas. Foi iniciada na unidade de Desenvolvimento POO em Ambiente Gráfico e terminou na unidade de Acesso a Base de Dados.

A base de trabalho selecionada para a intervenção foi o curso oferecido pela Unity, disponível na *Unity Learn*. Foi utilizado no desenvolvimento de conteúdos o curso *Junior Programmer: Manage scene flow and data*, unidades: *Principles of object-oriented programming*, *Implement data persistence between scenes* e *Implement data persistence between sessions*. A unity disponibiliza recursos para professores que podem ser adaptados às necessidades do curso, unidade curricular e utilizados em sala de aula. Unity é gratuito para professores e instituições de ensino acreditadas. Tem cursos desenhados para diferentes níveis de ensino e o Colégio de Gaia está registado como instituição acreditada desde 2020. Dispõe de 60 licenças da versão profissional, o meu interesse por esta plataforma motivou a adoção deste recurso educativo. A figura 18 confirma a veracidade da informação prestada e pode ser consultada no site da empresa.

^ Posso adquirir um plano Unity Educator para o meu curso?

O plano Unity Educator está disponível gratuitamente para educadores qualificados e não pode ser comprado. Consulte a [Concessão de Licença Educacional gratuita](#) para as necessidades escolares de sua instituição.

Figura 18 - Unity: Concessão de Licença Educacional gratuita.

Os conteúdos disponibilizados pela Unity foram concebidos com base em duas metodologias ativas acreditadas para o ensino, nomeadamente: Sala de Aula Invertida (Flipped Classroom) e Aprendizagem Baseada em Projetos (Project Based Learning - PBL). A figura 19 mostra o empenho da Unity no ensino e a disponibilidade de conteúdos educativos. Os materiais disponibilizados encontram-se redigidos em língua inglesa e espanhola.

Design your education experience

2a. Adapting Junior Programmer content for different teaching approaches and contexts

As an instructor/facilitator for a learning experience based around Junior Programmer, your most valuable contributions are likely to be:

- Basic scripting techniques and paradigms of C# development in Unity (this is especially the case for less technically literate cohorts)
- Facilitating discussion and exploration of creator skills and workplace industries
- Questioning to consolidate and deepen understanding
- Troubleshooting participant technical issues.

The following table offers some guidance on adapting this learning experience for your teaching approaches and circumstances:

Flipped Classroom / instruction	Pre-class work can be assigned by tutorial or Mission within the Junior Programmer Pathway. Research tasks for creator skills and real-time industry group discussions, presentations, or peer review feedback sessions are also ideal for the flipped classroom.
Project-based	The Junior Programmer Pathway is broken into Missions, with each Mission containing smaller projects. This can be used for project-based learning.

Figura 19 - Unity: Abordagens e metodologias ativas

Na minha intervenção foram desenvolvidas atividades numa sequência lógica de aprendizagem dos conceitos aqui abordados. Cada tarefa é disponibilizada aos alunos através da aplicação Microsoft Planner, num quadro Scrum.

Tarefas

○ **06FT_POOAG**
Alterada pela última vez a há 2 minutos por si

  Rui Miguel Loureiro Baptista

 Adicionar etiqueta

Grupo
BackLog do Projeto ▾

Progresso
○ Não iniciada ▾

Prioridade
● Médio ▾

Data de início
10/05/2022 

Data para conclusão
16/05/2022 

Repetir
 Não se repete ▾

Notas Mostrar no cartão

Objetivos:
-Trabalhar com Encapsulamento na programação orientada a objetos
-Proteger variáveis públicas
-Criar propriedade com campo de apoio

Lista de verificação 0/6 Mostrar no cartão

- 1. Na class GestorDoPrograma adicionou a seguinte variável publica:
- ⋮ ○ 1.1. Para efeitos experimentais, na classe MenuUIHandler adicione no final do ...  
- 1.1. Crie um get para aceder à variável GestorDoPrograma
- 1.2. Comente a linha de código criado no ponto 1.1
- 1.3. Adicione um setter privado à propriedade
- 2. Na class ResourcePile.cs, crie um campo de apoio para a variável productionSpeed.
- Adicionar um item

Anexos Mostrar no cartão

 06FT_POOAG.pdf 
<https://colegiodegaia.sharepoi>

Adicionar anexo

Figura 20 - Implementação Agile Scrum: Tarefa

Scrum é uma implementação da metodologia Agile, metodologia que revolucionou a forma como os projetos são desenvolvidos, proporciona flexibilidade, fomenta o trabalho colaborativo e pode ser adaptado às necessidades da equipa. Este é um dos *frameworks* mais populares no mercado na implementação da metodologia Agile. Agile é uma metodologia baseada em princípios e valores: promove a importância do indivíduo acima de processos e ferramentas enfatizando a importância da comunicação entre os membros da equipa; Entregas de software funcional e incremental; Colaboração com o cliente acima de documentação excessiva e a resposta a mudanças necessárias acima da necessidade de seguir a calendarização em caso de necessidade. Esta implementação foi utilizada para fomentar o trabalho em equipa e interdisciplinar, enquanto enriquece o currículo. A figura 20 ilustra a implementação.

No quadro apresentado na figura 21 o aluno tem acesso a toda a informação necessária ao desenvolvimento das tarefas. São informados do tempo estimado para a conclusão, com recurso a Data de Início e de Fim, tem acesso a notas do professor, lista de problemas a resolver e documentação necessária para consulta. A tarefa base é criada pelo professor, o aluno copia a tarefa, efetua a atribuição e si enquanto membro da equipa e avança nas colunas do quadro até à conclusão.

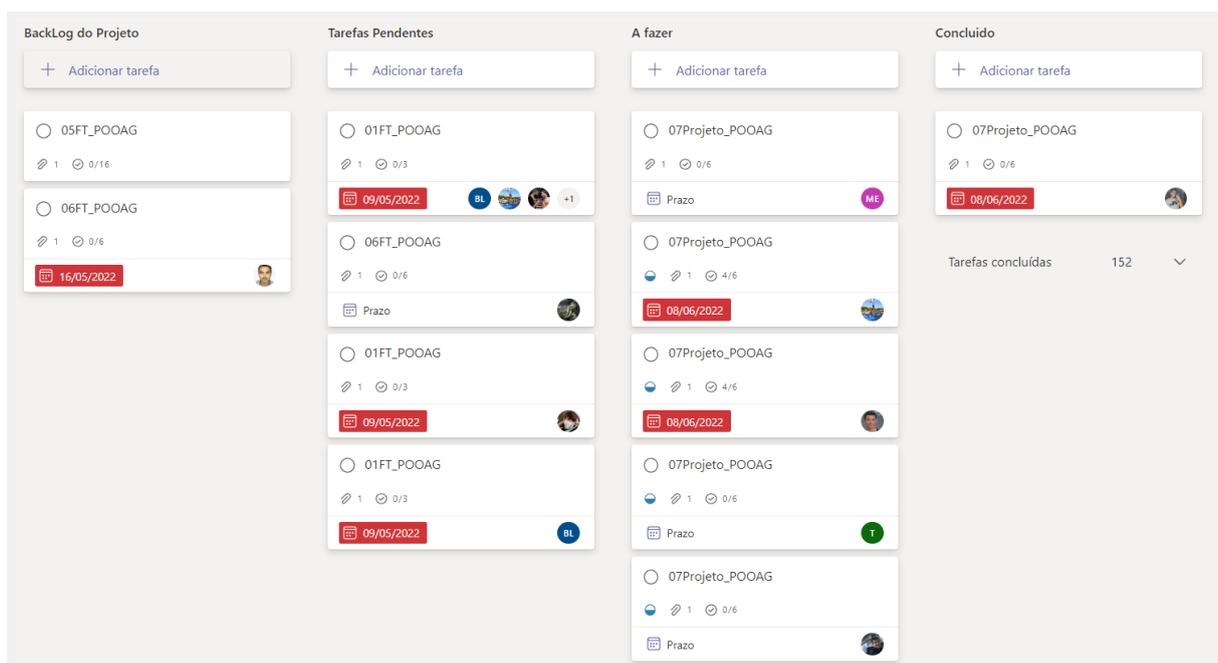


Figura 21 - Implementação Agile Scrum: Quadro

Toda a documentação do projeto foi centralizada, está disponível para consulta do aluno na Equipa do Projeto, criada com recurso à ferramenta integradora Microsoft Teams. A figura 22 ilustra a gestão documental da equipa. Na gestão documental, os alunos têm acesso a recursos diversos como código

fonte do projeto, fichas de trabalho e resolução, instrumentos de apoio à instalação de ferramentas, manual, informação do projeto, projetos (fichas de trabalho) e projeto final.



The screenshot shows a file management interface with the following structure:

- Documentos > General > 00_UNIDADE 2 POOAG
- Files/Folders:
 - APONTAMENTOS (Modified: 18/05/2022, Modified by: Rui Miguel Loureir...)
 - FICHAS DE TRABALHO (Modified: 18/05/2022, Modified by: Rui Miguel Loureir...)
 - GIT (Modified: 18/05/2022, Modified by: Rui Miguel Loureir...)
 - MANUAL (Modified: 18/05/2022, Modified by: Rui Miguel Loureir...)
 - PROJETO (Modified: 18/05/2022, Modified by: Rui Miguel Loureir...)
 - TRABALHO PRÁTICO (Modified: 26/05/2022, Modified by: Rui Miguel Loureir...)

Figura 22 -Gestão documental do projeto

Os alunos foram convidados a participar no desenvolvimento da documentação. Neste instrumento de apoio ao desenvolvimento foi disponibilizada informação relevante sobre as matérias abordadas que podem utilizar para facilitar a resolução dos problemas.

ABSTRAÇÃO E RESTRUTURAÇÃO DE SCRIPTS

Lembre-se de que uma parte importante da reestruturação de código é melhorar a sua funcionalidade sem alterar a maneira como outros programadores interagem com ela. A abstração desempenha um papel fundamental nisso! Contanto que a chamada e a saída do método não vão mudar, pode ajustar o conteúdo do método sem que o outro programador saiba. Isso significa que pode reestruturas o seu código com segurança, sem o risco de arranjar problemas no projeto.

Exemplo:

```
//Método que devolve o triplo do resultado obtido
1 referência
int ResultadoTriplo(int inputNumero)
{
    int outputNumero = inputNumero + inputNumero + inputNumero;
    return outputNumero;
}
```

Na função acima, o script de chamada passa um inteiro e obtém uma saída que é o triplo do valor recebido. A abordagem para obter o número de saída é um pouco ineficaz, então, pode reestruturar o código para que fique mais eficiente:

Figura 23 - Manual do projeto

O Manual disponibiliza ainda ao aluno alguns exercícios resolvidos que podem ajudar na interpretação. A figura 23 exibe informação facultada aos alunos para ajudar na interpretação de conceitos e resolução de problemas.

Outra ferramenta necessária ao desenvolvimento do projeto foi o GitHub Desktop. GitHub é uma implementação da plataforma *Git* que é um sistema de controlo e versionamento de código-fonte, amplifica a capacidade de equipas trabalharem no mesmo projeto em simultâneo. Permite o registo de alterações realizadas no projeto ao longo do tempo. Esta é mais uma ferramenta que promove o trabalho interdisciplinar e enriquece o currículo dos alunos.

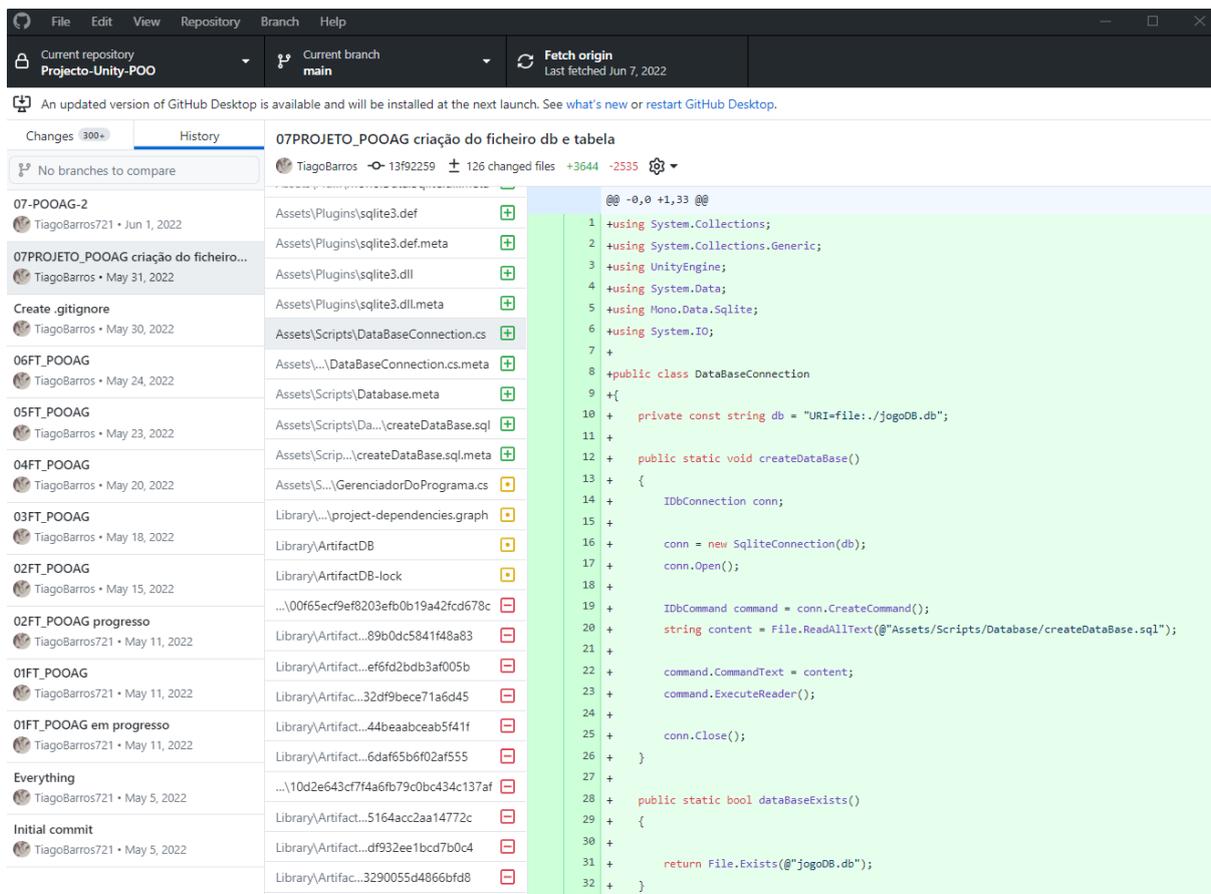


Figura 24 - Versionamento de software

Quando é finalizada uma tarefa o aluno entrega uma nova versão do projeto. O aluno tem a liberdade de no final da sessão submeter o estado atual do desenvolvimento e desta forma continuar o trabalho em casa se assim o entender. O processo é relativamente simples e rápido o que torna a sessão mais produtiva da mesma forma que promove o trabalho de casa.

Esta estratégia permite ainda ao professor a avaliação do trabalho desenvolvido com alguma facilidade.

Git permite comparar a versão atual com versões anteriores do código, o acesso às alterações e desenvolvimento atual do aluno é rápida e eficiente, este sistema de versionamento garante uma grande vantagem ao professor na avaliação do trabalho do aluno. A figura 24 ilustra o versionamento realizado por um aluno ao longo do desenvolvimento do projeto.

Para garantir o menor impacto possível, das restrições existentes, foram disponibilizados vídeos demonstrativos dos conteúdos lecionados, instalação e utilização das ferramentas. Os vídeos foram partilhados na team da equipa. Com esta utilidade, os alunos podem a partir de casa recuperar matérias essenciais ao desenvolvimento do projeto assim como auxiliam no esclarecimento de dúvidas. A figura 25 exhibe a organização e disponibilização dos artefactos. Saliento o elevado número de visualizações obtidas, permitem perceber de forma clara a adesão dos alunos a este tipo de conteúdo educativo.

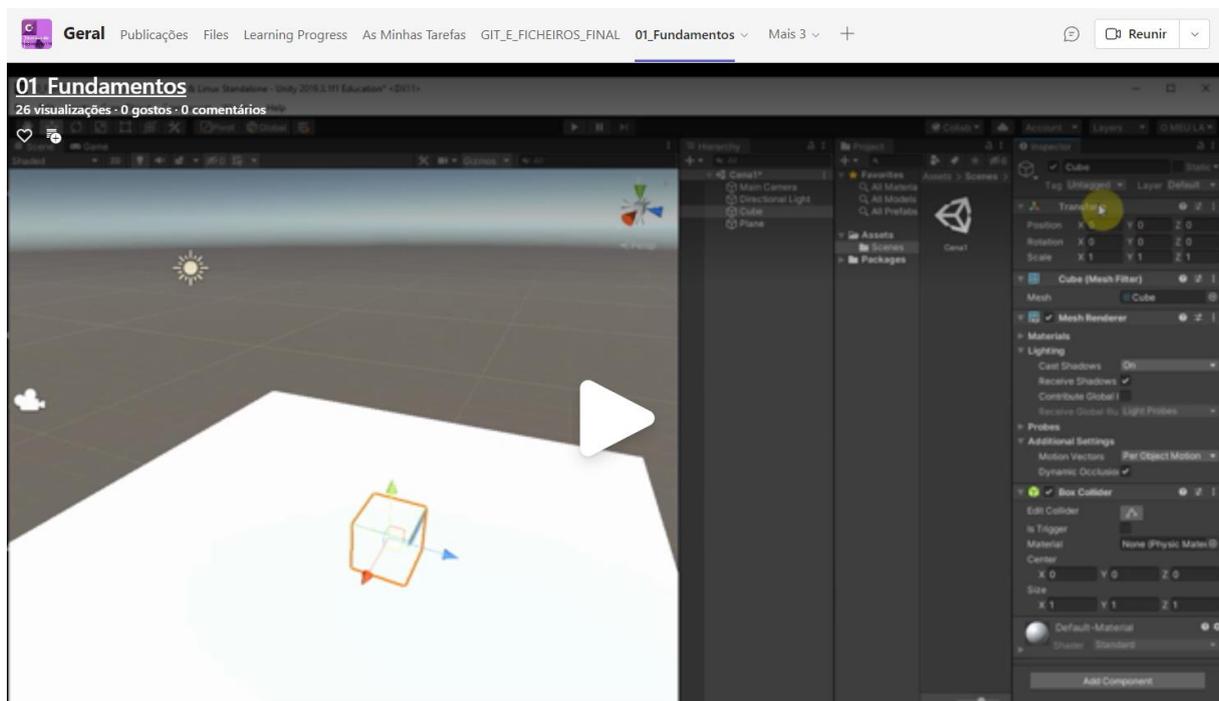


Figura 25 - Vídeos demonstrativos

4.5.1. Resumo do projeto

O projeto foi apresentado em forma de desafio. Procurei incutir a ideia de que professor e alunos representam como um todo uma equipa de desenvolvimento de software, o professor na qualidade de Team Leader e os alunos a equipa de desenvolvimento. O guião utilizado foi o seguinte:

“A empresa XPTO contratou uma reconhecida empresa de desenvolvimento de jogos que acabo por falir

sem terminar o trabalho produto. Eu, na qualidade de Team Líder desta equipa, fui contactado no sentido de apurar o nosso interesse enquanto interessados em dar continuidade ao projeto. Sem exitar, mostrei todo o meu interesse e recolhi a informação necessárias. A documentação foi disponibilizada na team da equipa, deve ser consultada imediatamente e a primeira tarefa deve ser iniciada na próxima aula. Bom trabalho a todos!”

O projeto a ser desenvolvido permitirá implementar funcionalidade numa aplicação que simula a gestão de recursos. A gestão de recursos está no centro de muitos géneros de jogos, mas as simulações são também comumente usadas em contextos educacionais (por exemplo, para explorar sustentabilidade e economia) assim como na indústria.



Figura 26 - Projeto: vista geral da cena.

Este tipo de projeto é particularmente adequado para ajudar a explorar sistemas de programação e arquitetura e muitas vezes incluem:

- Interação do utilizador, para permitir que influenciem a simulação;
- Transições entre cenas, para permitir maior personalização por parte do utilizador;
- Sistemas projetados para extensão, para aumentar a complexidade da simulação.

Exemplos de referência:

- **Fishbanks:** Uma simulação multijogador de gestão de recursos renováveis para uso educacional

- **Age of Empires** ou **Civilization**: jogos de estratégia clássicos que exigem gestão de recursos.

Lista de tarefas:

Gestão de Cenas

- Crie transições entre duas cenas.
- Configure botões para que o utilizador possa controlar as transições criadas.
- Configure um botão para sair da aplicação (ou sair do modo Play, no Unity Editor).

Persistência de dados

- Configure botões numa cena para aplicar uma cor escolhida a objetos de uma segunda cena.
- Salve a última cor escolhida pelo utilizador e garanta que fica pré-selecionada na próxima vez que a aplicação for executada.

Herança e Polimorfismo

- Crie um tipo de objeto na simulação com uma variação de comportamento derivada de uma classe base.

Abstração

- Analise o código para reduzir código duplicado e melhorar a reutilização.

Encapsulamento

- Utilize *getters* e *setters* para proteger os dados contra uso indevido.

Otimize o código

- Analise o código exemplo para identificar problemas básicos de otimização.

Visão geral do projeto

Importante: Ao abrir o projeto pela primeira vez a simulação do armazém terá funcionalidades básicas, mas a aplicação não funcionará!

Cenas no projeto

Há duas cenas no projeto Unity:

1. Um menu iniciar (Menu) onde, os utilizadores podem iniciar a simulação e fechar a aplicação.
2. Uma cena de simulação (Principal) que é estilizada como um armazém.

Nota: Uma cena adicional chamada Otimização — caso de estudo de otimização que será usada para traçar o perfil do código e identificar problemas.

Interação do utilizador

Irá implementar transições controladas pelo utilizador:

- Entre as duas cenas;
- Para sair da aplicação, apenas a partir da cena Menu.

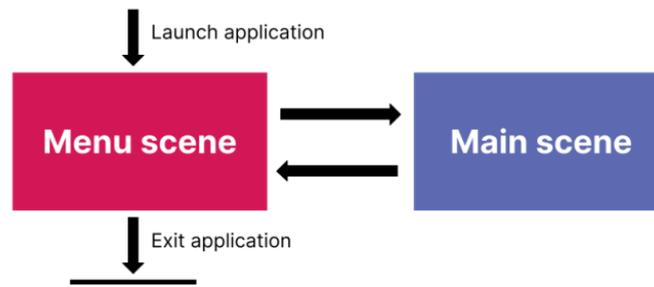


Figura 27 - Projeto: Interação do utilizador.

Interações do utilizador

O utilizador precisa ser capaz de:

- Iniciar a cena de simulação a partir do menu iniciar;
- Voltar ao menu iniciar da cena de simulação;
- Sair da aplicação (ou sair do modo Play, de testes no Editor);
- Selecionar uma cor para aplicar às Unidades Transportadoras (empilhadores) na simulação.

NOTA: A última cor escolhida pelo utilizador também deve ser pré-selecionada na próxima vez que iniciar a aplicação. A figura 28 ilustra a cor selecionada.

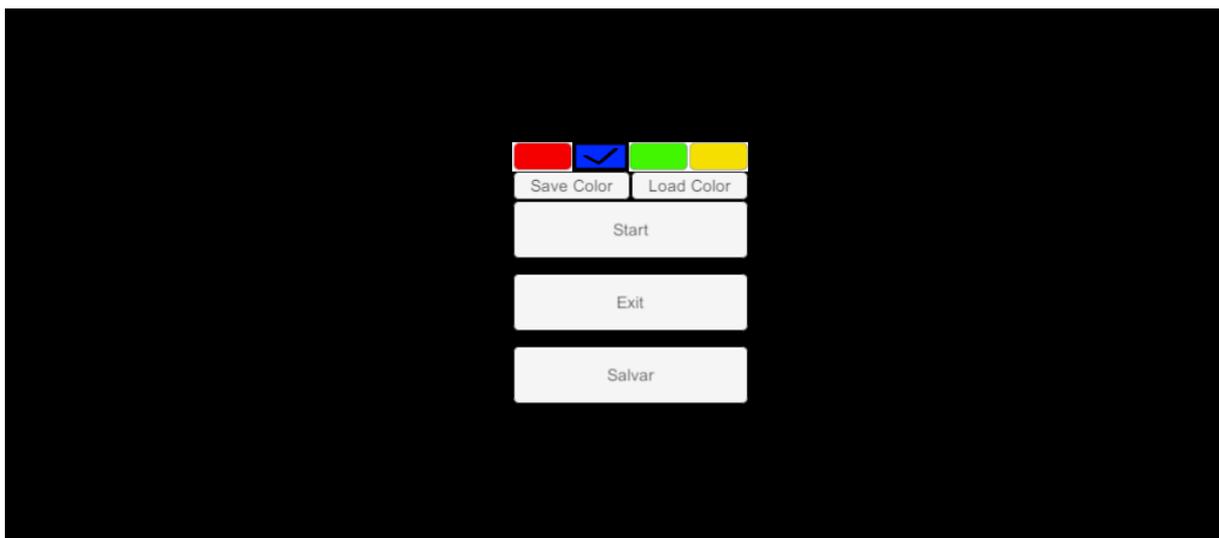


Figura 28 - Projeto: Cor Selecionada

Simulação do armazém

A simulação básica (na cena principal) que contém:

- Dois tipos diferentes de Pilha de Recursos (objetos numa palete) que, produzem recursos “itens” a uma velocidade de 0,5 por segundo (/s)
- Dois tipos de Unidade:

- Uma Unidade (trabalhador) que se pode mover, mas, não tem mais funcionalidades.
- Uma unidade de transporte (empilhador), que o utilizador pode definir para transportar “itens” para uma base predefinida (o espaço na frente do camião marcado com um círculo vermelho).

A figura 29 exhibe a cena, unidades e recursos.



Figura 29 - Projeto: Unidades e Recursos

Funcionalidades básicas da simulação

Na simulação básica, o utilizador pode:

- Usar as teclas de seta (ou WASD) para mover a câmara pelo espaço do armazém.
- Clique com o botão esquerdo em qualquer objeto definido para selecioná-lo e abrir uma sobreposição de IU com os seus detalhes.
- Quando uma unidade é selecionada, clique com o botão direito do rato:
 - Em qualquer local, para mover o utilizador para essa localização;
 - Numa Pilha de Recursos, para configurá-la para transportar itens de recursos do objeto para a Base.

Requisitos de simulação adicionais

Ao trabalhar neste projeto vai:

- Criar uma Unidade de Produtividade que aumenta a produção de itens de recurso numa base e aplicá-lo ao trabalhador no armazém.
- Refinar e otimizar o código criado, aplicando princípios de Programação Orientada a objetos.

Estilo do projeto

A temática deste projeto é um armazém, num estilo *low poly*. Se quiser personalizar o tema do projeto enquanto trabalha nesta funcionalidade, pode importar os seus próprios ativos e trocá-los.

Scripts fornecidos

O projeto vem com sete scripts principais que se encontram parcial ou completamente escritos. Existem alguns scripts adicionais em subpastas que pode visualizar, mas não estão incluídos aqui.

Base.cs

Esta é uma subclasse da classe *Building*. Não altera o comportamento da classe, mas armazena uma referência (função *Awake*). Significa que pode extrair a base de qualquer lugar. É usada pelas Unidades quando elas voltam à base.

Building.cs

Esta é uma classe base abstrata de duas subclasses: *ResourcePile* e *Base*. Mantém um inventário do que os membros da classe contêm e também suporta a adição e remoção de “itens” desse inventário. Também implementa *IUIInfoContent*, a sobreposição básica da interface do utilizador na simulação.

ColorHandler.cs

Script que permite criar botões de cores para que os utilizadores possam selecionar uma cor e aplicar às Unidades de Transporte a partir do menu iniciar.

ResourcePile.cs

Esta é uma subclasse da classe *Building*. Leva uma referência a um *ResourceItem* que irá produzir e armazenar as seguintes informações sobre ele numa classe *Serializable*:

- nome
- ID exclusivo
- *sprite* associado

Define uma função de atualização e incrementa um contador de produção pela velocidade/s de produção a cada atualização. Uma vez acima de 1, cria recursos e diminui o contador.

Nota: também substitui a função *GetData* para o *IUIInfoContent* de *Building* para dar ao *InfoPanel* a sua velocidade de produção como informação.

TransporterUnit.cs

Esta é uma subclasse da **Unit.cs**. Contém:

- O destino (Pilha de Recursos) para o qual uma Unidade de Transporte está atualmente definida;
- A quantidade de recursos que pode transportar de uma só vez;
- O que está a ser transportado atualmente pela Unidade Transportadora (Empilhador).

A subclasse *TransporterUnit* substitui três funções de sua classe base:

- Comportamento à volta dos edifícios nas proximidades;
- Duas funções *GoTo*, para reagir quando o alvo da unidade de transporte muda **Unit.cs**.

Esta é uma classe base, abstrata para todas as Unidades. Controla Unidade:

- Movimento, incluindo um ponto de destino para o qual se mover (que pode ser um Edifício ou uma posição);
- Cor, quando configurado;
- Comportamento quando estiver no alcance de seu edifício alvo.

UserControl.cs

Esta classe lida com os inputs do utilizador. Move a Câmara no plano xz ao ler os valores vertical e horizontalmente. Também controla a funcionalidade de clique com o botão esquerdo e direito.

O resumo do projeto foi entregue aos alunos na primeira aula juntamente com o código fonte e manual com instruções de utilização. O projeto incremental prevê seis fase que serão entregues na aplicação de projetos durante o decorrer da intervenção. Cabe aos alunos analisar os problemas e implementar a solução. É esperado que esta estratégia de desenvolvimento valorize a construção do conhecimento pelos próprios alunos, estimule a participação ativa e promova uma aprendizagem mais significativa e aliciante, colocar alunos no papel de criadores é uma das possíveis abordagens à integração de jogos na educação, suportada pelo Construcionismo.

4.5.2. Atividades realizadas no âmbito da intervenção

No decorrer da intervenção pedagógica, foram desenvolvidas duas atividades com objetivos distintos. A primeira teve como propósito enfatizar a importância e a necessidade da utilização das ferramentas introduzidas, enquanto a segunda teve como objetivo reforçar e consolidar os conhecimentos dos alunos em relação às problemáticas abordadas durante a intervenção supervisionada.

4.5.2.1. Conversa com Especialistas

Para promover as ferramentas introduzidas foram convidados dois especialistas na área do desenvolvimento de software que trabalham em empresas de renome nacional e internacional. O objetivo da sessão foi promover o contacto com profissionais, utilizadores de ferramentas de versionamento de software, Gestão de Projetos e Gestão documental, utilizadores de metodologia *Agile* no âmbito do trabalho e *Frameworks* assentes na metodologia como o *scrum*. A sessão decorreu em regime não presencial, com recurso a videoconferência, a figura 30 ilustra a sessão.



Figura 30 - Conversa com especialistas: Videoconferência

Os convidados falaram das suas experiências e utilização das ferramentas em contexto de trabalho. A importância da utilização destas ferramentas em equipa e da forma como facilitam o trabalho seja este presencial ou remoto. Colocaram o trabalho em equipa acima do trabalho individual como forma de desenvolvimento e criação de conhecimento.

Salientaram ainda a forma como esta abordagem permitiu suavizar os impactos das restrições em vigor, por se tratar de estratégias implementadas antes da pandemia suavizaram o efeito, tanto a nível pessoal como empresarial. Mostraram desta forma a importância das ferramentas e metodologia para o setor, considerando o impacto no trabalho deles como quase nulo.

Esta atividade foi alvo de avaliação com recurso a formulário onde foram colocadas as seguintes questões:

Grau de motivação para a sessão?

A figura 31 ilustra a resposta dos alunos à primeira questão colocada.

Muito Bom	4
Bom	19
Suficiente	2
Insuficiente	0



Figura 31 - Conversa com especialistas: Motivação

16% dos alunos consideraram o grau de motivação foi Muito Bom. 76% consideraram Bom, 8%

suficiente e nenhum aluno considerou estar desmotivado. A reação positiva dos alunos à atividade indica que estavam interessados e motivados para as temáticas abordadas.

Como avalia a sessão?

A figura 32 ilustra a resposta dos alunos à questão número dois.

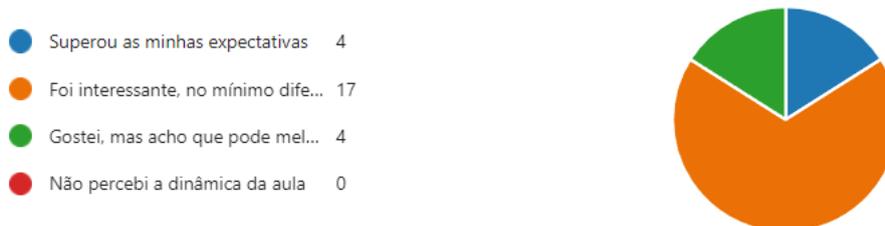


Figura 32 - Conversa com especialista: Avaliação sessão

16% superou as minhas expectativas, 68% consideraram interessante, no mínimo diferente, 16% gostaram, mas consideram que pode melhorar e nenhum aluno considerou não ter percebido a dinâmica da aula. Com base nos resultados posso concluir que a sessão foi na generalidade bem avaliada pelos participantes. A maioria dos alunos ficou satisfeita com o conteúdo e demonstrou interesse nas temáticas discutidas. O facto de que nenhum aluno relatar problemas de compreensão da dinâmica da aula sugere que a sessão foi bem estruturada e acessível.

Qual o nível de interesse despertado pela apresentação?

A figura 33 ilustra a resposta dos alunos à terceira questão.



Figura 33 - Conversa com especialistas: Interesse Despertado

20% dos alunos considerou estar muito interessado e que despertou grande curiosidade. 72% interessados, acharam relevante e útil, 8% neutrais, nem muito interessados nem desinteressados e 0% estiveram pouco interessados, não se sentiram envolvidos pelo tema. A maioria dos alunos demonstrou um nível significativo de interesse pela apresentação. A combinação dos alunos que se mostraram muito interessados e neutrais sugere que a sessão conseguiu envolver a grande maioria dos participantes, despertar interesse e relevância em relação ao tema abordado. A ausência de alunos que se sentiram

pouco interessados ou desinteressados reforça a percepção positiva da atividade por parte da turma como um todo.

Considera importante a utilização destas ferramentas a nível pessoal?

A figura 34 ilustra a resposta dos alunos à questão.

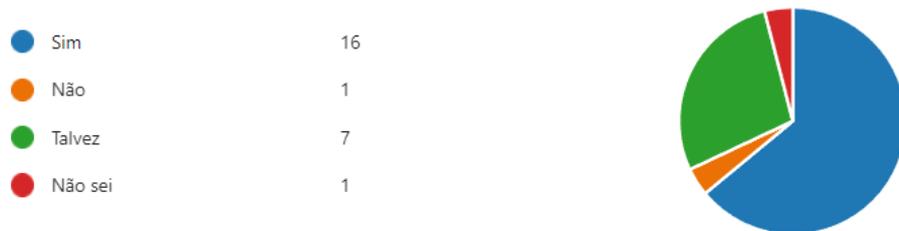


Figura 34 - Conversa com especialistas: Interesse na utilização pessoal

64% consideram que sim, 4% consideram que não, 28% talvez e 4% não sabe. No geral, a maioria dos alunos considera importante a utilização das ferramentas a nível pessoal, o que destaca o reconhecimento da relevância das práticas e metodologias de desenvolvimento de software não apenas em contextos profissionais, mas também para aprimorar as suas atividades e projetos pessoais. A presença de uma pequena percentagem de desinteresse e indecisão sugere que alguns alunos podem precisar de mais informações ou esclarecimentos sobre o funcionamento das ferramentas ou não estão interessados em progredir estudos nesta área.

Considera importante a utilização destas ferramentas a nível profissional?

A figura 35 ilustra a resposta dos alunos à última questão colocada.

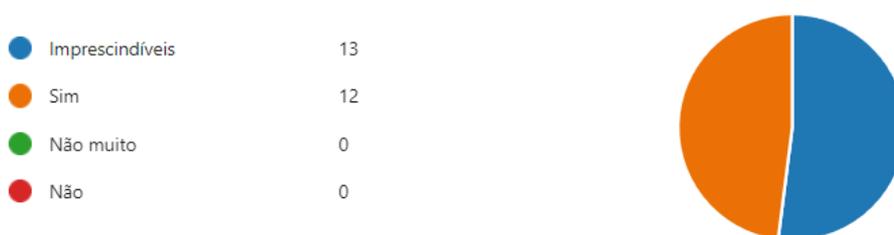


Figura 35 - Conversa com especialistas: Interesse na utilização profissional

52% consideram imprescindível, 48% responderam que sim, 0% não muito e 0% não. No geral, a maioria dos alunos considera a utilização das ferramentas importante e até mesmo imprescindível em contextos de trabalho. A percepção positiva reflete o papel fundamental que as ferramentas têm no desenvolvimento de software e na gestão de projetos em ambientes profissionais. Nenhum aluno respondeu que as ferramentas não são importantes a nível profissional sugere que reconhecem serem valiosas e essenciais para o sucesso no mercado de trabalho na área de desenvolvimento de software.

Com base nos dados recolhidos, podemos concluir que a sessão foi bem avaliada pelos participantes e despertou um nível significativo de interesse e motivação. A maioria dos alunos demonstrou satisfação com o conteúdo apresentado e considerou a sessão interessante, relevante e útil.

Além disso, a importância da utilização das ferramentas e metodologias abordadas foi reconhecida tanto a nível pessoal quanto profissional. A maioria dos alunos considera a utilização das ferramentas importante em ambas as esferas, seja na utilização em atividades pessoais ou profissionais.

Houve uma reação positiva, no geral, e uma minoria expressa alguma indecisão ou neutralidade em relação ao nível de interesse despertado pela apresentação. No entanto, a inexistência de alunos desmotivados ou pouco interessados sugere que a atividade foi bem estruturada e conseguiu envolver a grande maioria dos participantes.

A avaliação positiva demonstrada pelos alunos indica que a sessão foi eficaz e atingiu os objetivos de promover o contato com profissionais experientes, apresentar as ferramentas e metodologias relevantes ao desenvolvimento de software e destacar a sua importância tanto para o trabalho em equipa.

No geral, os resultados sugerem que a sessão foi bem-sucedida em proporcionar aos alunos uma experiência enriquecedora e informativa, contribuiu para o enriquecimento do conhecimento e saber no contexto do desenvolvimento de software e práticas relacionadas.

4.5.2.2. Envelope secreto

Foi ainda desenvolvida uma dinâmica com os alunos em sala de aula para apurar o conhecimento adquirido sobre os conceitos estudados durante o desenvolvimento do jogo.

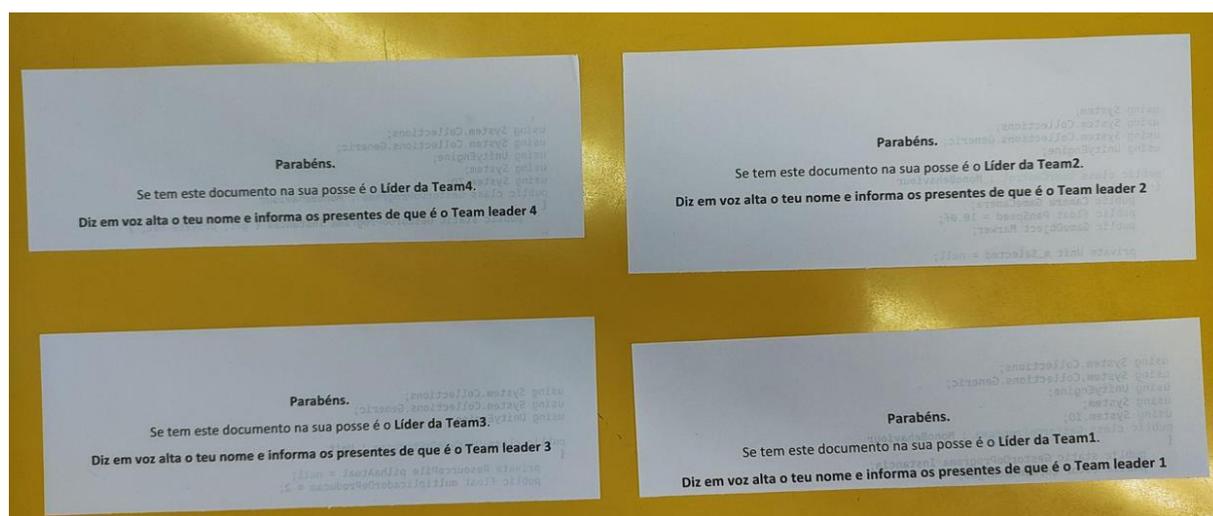


Figura 36 - Envelope Secreto: Mensagem secreta Team Leader

Consiste num envelope que é distribuído a cada um dos alunos com uma parte do código de uma classe que tiveram de criar para o projeto, quatro envelopes continham uma mensagem adicional. A figura 36 ilustra a mensagem.

Os pilares da programação a objetos foram praticados pelos alunos em quatro classes distintas, não em exclusividade, mas com maior destaque. A mensagem tinha o propósito de informar o aluno que retirou ao acaso o envelope com a mensagem de que a partir daquele momento ele se tinha tornado um líder de uma das quatro equipas a formar. A figura 37 ilustra a mensagem encontrada no envelope, no verso da primeira parte do código.

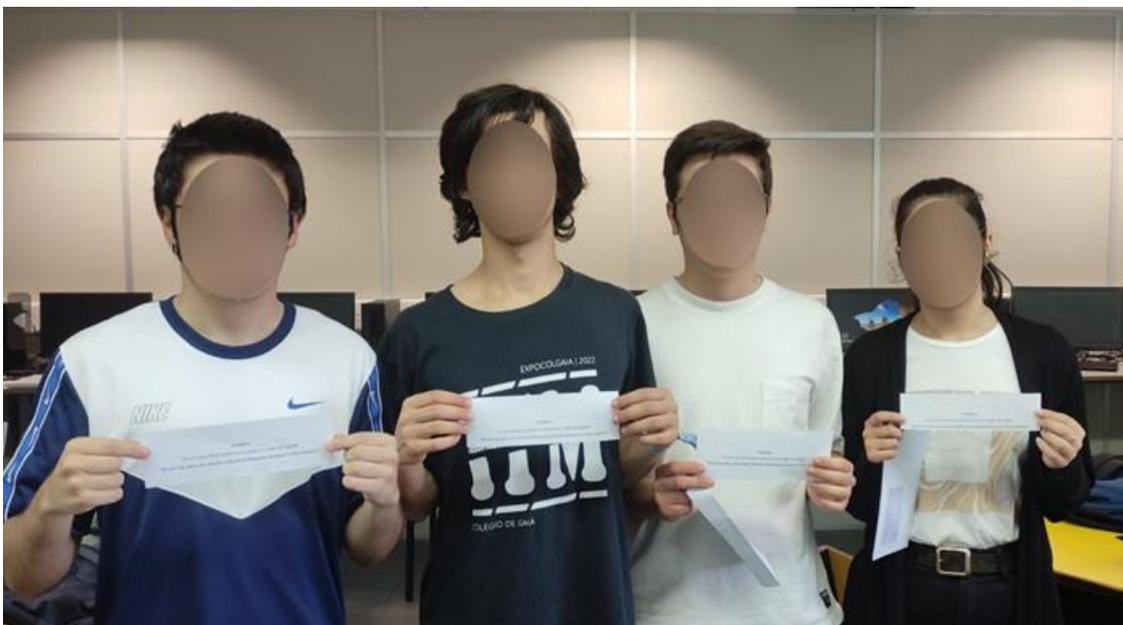


Figura 37 - Envelope Secreto: Team Leader

Caberia ao líder e aos restantes colegas encontrarem entre si as restantes partes do código da classe para identificar a equipa correspondente. Após a formação das equipas, o desafio proposto consistia em prepararem uma breve apresentação sobre o conceito estudado. No final de cada apresentação, eram esclarecidas dúvidas e promovia-se a discussão para aprofundar a compreensão do tema. A figura 38 ilustra a criação dos grupos de trabalho e organização da sala de aula.



Figura 38 - Envelope Secreto: Formação dos grupos de trabalho

A avaliação da sessão foi realizada com recurso a formulário. As questões colocadas foram as seguintes:

Grau de motivação para a sessão?

A figura 39 ilustra a resposta dos alunos à questão.

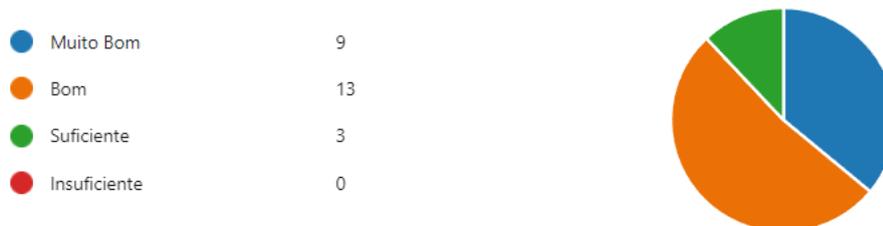


Figura 39 - Estatística Envelope Secreto: Grau de motivação

36% reponderam Muito Bom, 52% responderam bom, 3% suficiente e 0% responderam insuficiente. A análise dos dados revela que a maioria dos alunos estava motivada para a sessão, com uma percentagem significativa classificando-a com "Muito Bom" a maioria considerando-a "Bom". A ausência de alunos a indicar desinteresse sugere que a abordagem da atividade, o conteúdo apresentado e dinâmica da sessão foram eficazes a manter o interesse e envolvimento dos alunos durante a sessão.

Como avalia a dinâmica desenvolvida na aula de hoje?

A figura 40 ilustra a resposta dos alunos à segunda questão.



Figura 40 - Estatística Envelope Secreto: Avaliação da aula

40% respondeu superou as minhas expectativas, 56% que foi interessante, no mínimo diferente, 4% Gostei, mas acho que pode melhorar e 0% Não percebi a dinâmica da aula. A análise dos dados revela uma reação geralmente positiva dos alunos em relação à dinâmica desenvolvida na aula. A maioria dos alunos achou a atividade interessante e diferente, enquanto um número considerável ficou positivamente surpreendido ao ver que a dinâmica superou as suas expectativas. O feedback construtivo de alguns alunos também pode ajudar a melhorar a dinâmica no futuro. No geral, os resultados indicam que a dinâmica foi bem-sucedida, envolveu e foi interessante para os alunos, proporcionou uma experiência de ensino positiva.

Esta sessão permitiu esclarecer dúvidas relacionadas com os temas abordados ao longo desta unidade curricular?

A figura 41 ilustra a resposta dos alunos à terceira questão.



Figura 41 - Estatística Envelope Secreto: Sessão no esclarecimento de dúvidas.

44% respondeu sim, todas as minhas dúvidas foram esclarecidas, 48% respondeu, sim, a maioria das minhas dúvidas foram esclarecidas, 8% sim, algumas dúvidas foram esclarecidas e 0% não, as minhas dúvidas não foram esclarecidas. A análise sugere que a sessão foi eficaz no esclarecer dúvidas relacionadas com as temáticas abordadas ao longo da unidade curricular. A grande maioria dos alunos esclareceu as suas dúvidas, o que indica que a atividade foi bem-sucedida, valorizou a compreensão dos conceitos estudados. Dado que nenhum aluno considerou que as suas dúvidas não foram esclarecidas reforça a percepção positiva da sessão como uma fonte efetiva de aprendizagem.

Qual era a tua compreensão geral dos pilares da Programação Orientada a Objetos (POO) antes desta sessão?

A figura 42 ilustra a resposta dos alunos à questão.

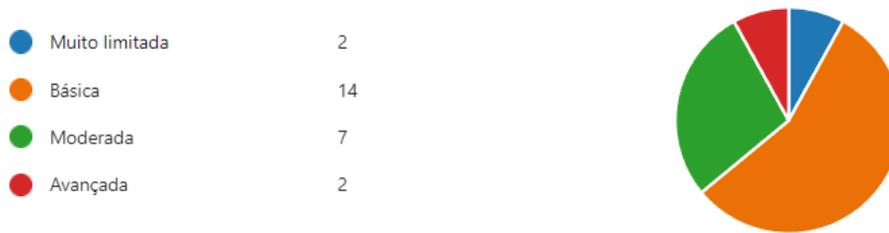


Figura 42 - Estatística Envelope Secreto: Compreensão dos pilares antes da sessão.

8% respondeu muito limitada, 56% respondeu básica, 28% moderada e 8% avançada. A compreensão geral dos alunos dos pilares da programação orientada a objetos antes da sessão variava de muito limitada a avançada, compreende diferentes níveis de conhecimento sobre o assunto. A sessão pode ter sido útil para melhorar, independentemente do nível inicial de conhecimento.

Após esta sessão, como avalias a tua compreensão dos pilares da POO?

A figura 43 ilustra a resposta dos alunos à questão.

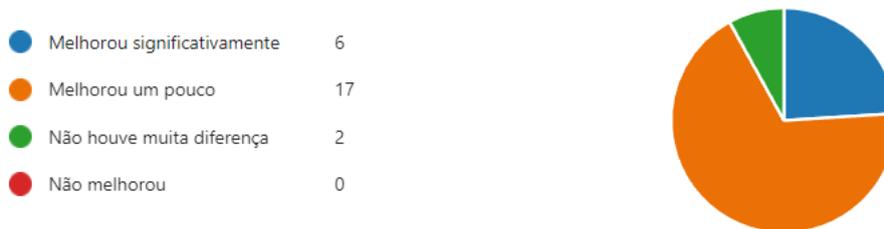


Figura 43 - Estatística Envelope Secreto: Avaliação depois da sessão

24% respondeu que melhorou significativamente, 68% melhorou um pouco, 8% não houve muita diferença e 0% não melhorou. A grande maioria revela uma melhoria na compreensão, com alguns alunos a registar um progresso significativo, enquanto outros tiveram melhorias moderadas. Os resultados indicam que a atividade foi bem-sucedida e contribuiu para a melhoria do conhecimento dos alunos sobre os conceitos da POO. No entanto, é importante considerar as sugestões dos alunos que sentiram pouca diferença na compreensão dos conceitos, futuramente, tentar encontrar formas de personalizar a abordagem de ensino para atender às necessidades de todos os estudantes é uma opção.

Durante a unidade curricular, foi possível aplicar os conceitos dos pilares da P00 na prática? Se sim, como avalia essa experiência?

A figura 44 ilustra a resposta dos alunos à questão.

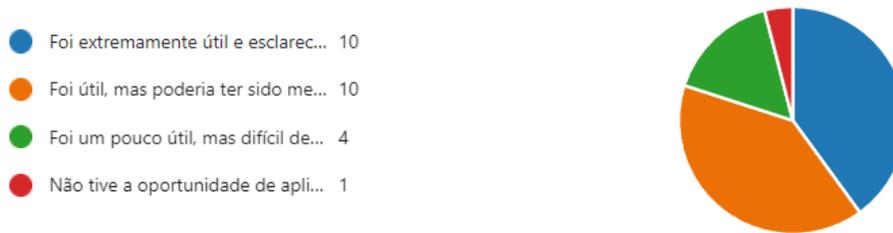


Figura 44 - Estatística Envelope Secreto: Foi possível a aplicação dos conceitos na prática.

40% considerou extremamente útil e esclarecedora, 40% útil, mas poderia ser mais explorada, 16% considerou pouco útil e 4% não teve a oportunidade de aplicar os conceitos na prática. Com base nos resultados, a maioria dos alunos teve uma experiência positiva ao aplicar os conceitos dos pilares da P00 na prática. É importante observar que uma pequena percentagem (16%) considerou a experiência pouco útil, o que pode ser uma oportunidade para avaliar e melhorar a abordagem de aplicação prática dos conceitos.

Qual foi o impacto da atividade em grupo na tua compreensão e aplicação dos pilares da P00?

A figura 45 ilustra a resposta dos alunos à última questão.

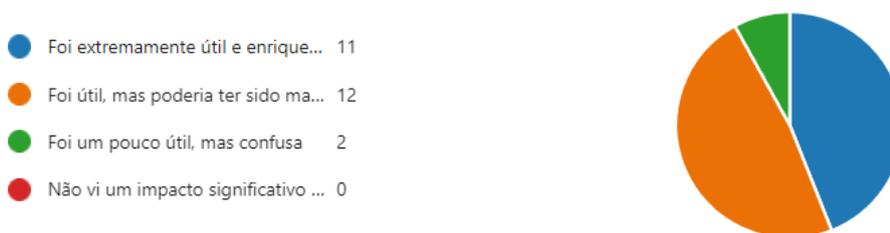


Figura 45 - Estatística Envelope Secreto: Aplicação dos conceitos na prática.

44% consideraram extremamente útil e enriquecedora, 48% útil, mas poderia ter sido mais direcionada, 8% um pouco útil, mas confusa e 0% não viu um impacto não significativo da atividade de grupo. A análise dos dados sugere que a atividade em grupo teve um impacto geralmente positivo na compreensão e aplicação dos pilares da P00. Para a maioria dos alunos, a atividade foi útil e enriquecedora, permitindo-lhes explorar e aprender de forma mais efetiva os conceitos através do trabalho colaborativo. No entanto, algumas sugestões de melhoria foram mencionadas: tornar a atividade mais direcionada ou evitar confusões. Estas informações podem ser úteis para ajustar e melhorar futuras atividades em grupo com

o objetivo de maximizar a aprendizagem e a experiência dos alunos.

Resumindo, a sessão parece ter sido bem-sucedida no que respeita à motivação dos alunos, foi envolvente, dinâmica e interessante. Ajudou a esclarecer as dúvidas e a melhorar a compreensão dos pilares da POO. A aplicação prática dos conceitos, tanto na atividade em grupo como individualmente, também teve impactos positivos para a maioria dos alunos, embora, consideram que há espaço para melhorias. Os resultados sugerem que a abordagem e estratégias utilizadas foram eficazes ao proporcionar uma experiência de aprendizagem valiosa e significativa para a maioria dos alunos na unidade curricular.

4.5.3. Projeto Final

Para concluir a intervenção pedagógica, foi realizado um trabalho prático em consonância com os conteúdos programáticos da unidade de "Acesso a Base de Dados". Esse trabalho consistiu na continuidade do desenvolvimento do jogo realizado pelos alunos, com o objetivo de criar uma Base de Dados que permitisse o registo de utilizadores, a edição de dados, a leitura e validação do utilizador ao iniciar sessão, o armazenamento dos dados da sessão quando o utilizador conclui o jogo, e a disponibilização de uma lista de resultados.

Os recursos necessários para a realização do projeto são:

- Base de Dados SQLite com o mínimo de 3 tabelas.
- Classe para gestão de operações na Base de Dados, como INSERT, CREATE, UPDATE, SELECT.
- Classe para gestão de dados relacionados com o utilizador e sessão.
- Formulários para Registrar, Login, Editar utilizador e apresentar resultados. Outros que considerar necessários.
- Validação de dados.

O projeto pode ser desenvolvido segundo diagrama apresentado na figura 46.

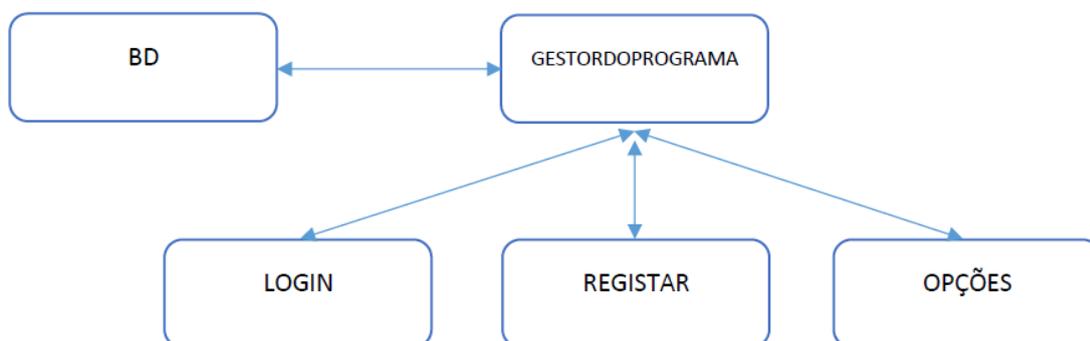


Figura 46 - Projeto final: Diagrama do projeto.

A avaliação do projeto foi realizada com base nos seguintes parâmetros:

- Validação de entradas
- Criação de classes
- Abstração
- Herança
- Polimorfismo
- Encapsulamento
- Qualidade Global

A figura 47 apresenta uma possível solução para o formulário de início de sessão. Os campos "Nome de utilizador" e "Password" são de preenchimento obrigatório. O aluno foi desafiado a criar o formulário, de modo a permitir que o utilizador insira os dados necessários para efetuar o início da sessão na aplicação. Deve ainda realizar a validação dos dados inseridos e exibir uma mensagem de erro caso a inserção não esteja correta ou caso a conta não esteja registada.

Para proporcionar uma experiência amigável, o formulário de início de sessão também deve oferecer ao utilizador a opção de registo. Na figura, é possível observar a presença de um botão para o login. Todos os elementos de interação presentes no formulário devem estar devidamente identificados, e a sua funcionalidade deve ser intuitiva.

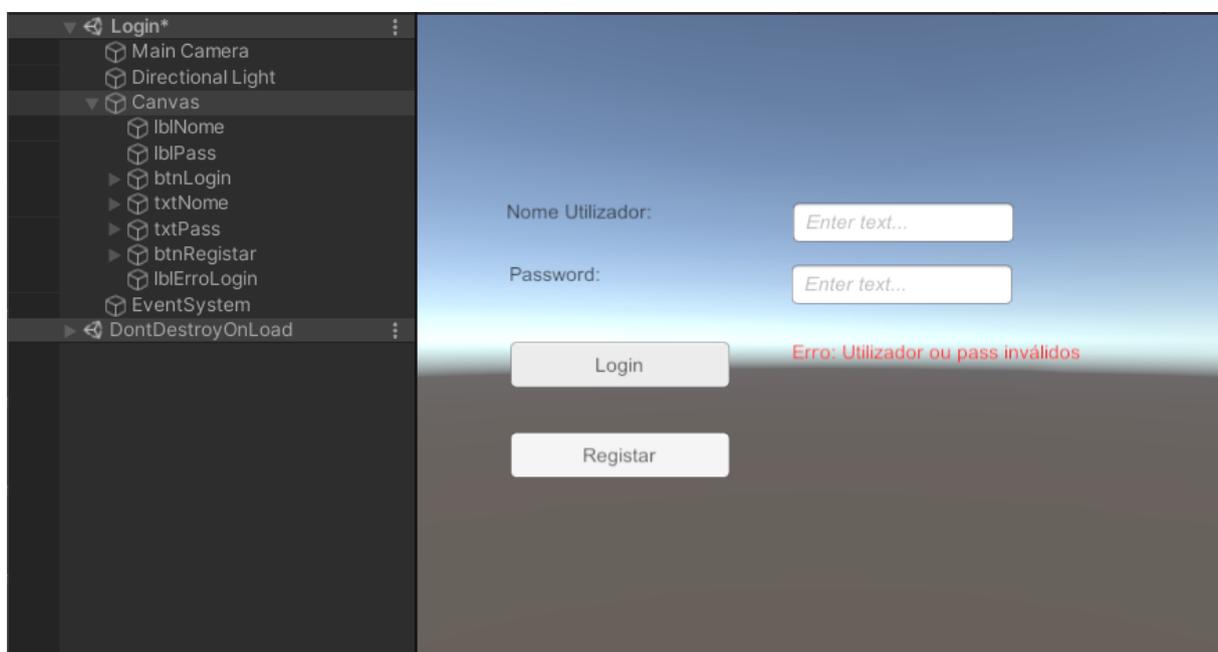


Figura 47 - Projeto final: Menu Login

Na figura 48 é apresentado um possível menu de registo, que exhibe os elementos de interação obrigatórios e outros facultativos para o registo do utilizador. No menu apresentado, optei por incluir um botão para a criação da base de dados, mas os alunos têm a liberdade de melhorar a experiência, caso assim o desejem. A base de dados pode ser criada no script desenvolvido para lidar com as interações com a base de dados, caso ela ainda não exista.

Durante o registo, os alunos devem assegurar que o utilizador confirma a sua password, verificar também se o utilizador já foi registado anteriormente. Além disso, o sistema deve fornecer mensagens informativas ao, orientando-o durante o processo de registo. Por fim, é importante que o menu de registo permita ao utilizador regressar ao menu de início de sessão, garantindo uma experiência fluída e intuitiva.



Figura 48 – Projeto final: Menu de registo

Para concluir, é necessário apresentar os resultados obtidos pelos utilizadores através de uma lista de resultados. A figura 49 ilustra um exemplo de como essa lista poderia ser estruturada. É responsabilidade do aluno realizar a formatação adequada da lista e garantir que os dados são apresentados por ordem de classificação, de forma organizada e compreensível. A lista deve fornecer uma visão clara e objetiva das pontuações alcançadas pelos utilizadores, possibilitando a identificação dos melhores resultados de forma fácil e intuitiva.



Figura 49 - Projeto final: Lista de resultados

A figura 50 apresenta uma parte da solução para a classe BD que deve ser desenvolvida pelo aluno. É fundamental que o aluno assegure que a classe seja capaz de lidar com todas as interações necessárias com a base de dados, garantir a integridade dos dados de acordo com os objetivos da intervenção e os conceitos da Programação Orientada a Objetos. A classe deve ser cuidadosamente projetada e implementada para permitir a criação, leitura, atualização e exclusão de dados na base de dados, bem como a realização de validações necessárias para manter a consistência dos dados armazenados. O aluno deve aplicar os princípios da POO, como encapsulamento, herança e polimorfismo, para criar uma solução robusta e eficiente que atenda aos requisitos da intervenção. Além disso, é essencial que a classe seja adequadamente documentada e testada para garantir o correto funcionamento e facilidade de manutenção no futuro.

O Projeto Final foi uma etapa crucial da intervenção pedagógica, permitiu que os alunos aplicassem de forma prática os conhecimentos adquiridos. Ao desenvolver a aplicação com uma base de dados funcional e utilizando conceitos de POO, os alunos tiveram a oportunidade de melhorar a destreza técnica, para além de consolidar o entendimento dos conteúdos teóricos da unidade curricular "Acesso a Base de Dados".

```

8 referências
public class BD
{
    //Criar ligação a base de Dados
    private string bd = "URI=file:./jogoDB.db";
    IDbConnection ligacaoBD;
6 referências
    public void Ligacao()
    {
        //Abrir ligação
        ligacaoBD = new SqlConnection(bd);
        ligacaoBD.Open();
    }
1 referência
    public void Criar()
    {
        try
        {
            //Abrir ligação
            Ligacao();

            // Criar BD
            IDbCommand dbcmd;
            dbcmd = ligacaoBD.CreateCommand();

            //Criar tabelas
            string q_CriarTabelas = " CREATE TABLE IF NOT EXISTS jogador (" +
                " id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT," +
                " nome VARCHAR(30)," +
                " nomeUtil VARCHAR(10) NOT NULL," +
                " email VARCHAR(50) NOT NULL," +
                " pass VARCHAR(50) NOT NULL);" +
                " CREATE TABLE IF NOT EXISTS dados_jogo (" +
                " id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT," +
                " nomeUtil VARCHAR(10) NOT NULL," +
                " unidlocalizacao VARCHAR(20) NOT NULL," +
                " unidCor VARCHAR(40) NOT NULL );";

            //Criar tabelas
            dbcmd.CommandText = q_CriarTabelas;
            dbcmd.ExecuteReader();

            //Fechar ligação
            FecharLigacao();
        }
    }
}

```

Figura 50 - Projeto final: Classe BD

4.6. Ludificação com estratégia motivacional

Tem-se verificado na última década um impulso para implementar e operacionalizar a utilização de estratégias que habitualmente associamos a jogos – como a atribuição de prémios, acumulação de pontos, e estabelecimento de tabelas de pontuação (leaderboards) (Mesquita, 2021).

Neste projeto, a ludificação foi adotada como uma estratégia motivacional, utilizando a aplicação de Gestão de Projetos Planner. Os alunos eram recompensados com base nos objetivos alcançados durante o desenvolvimento do jogo e adquiriam competências à medida que acumulavam pontos. Esta abordagem permitiu envolver os alunos de forma mais ativa e estimulante, incentivando-os a participar e progredir no projeto de maneira entusiástica.

4.6.1. Quadro de progresso do projeto

Dentro do quadro de progresso do projeto, pontos são atribuídos a cada tarefa concluída. Durante o desenvolvimento, o aluno deve ter em consideração o tempo de resolução, a eficiência do código produzido e a aplicação dos princípios fundamentais da POO. Na tabela 3, são apresentados os diferentes estados e a pontuação necessária para progredir. Esta abordagem de pontuação e progressão permite incentivar os alunos a dedicar mais tempo ao projeto e a aplicar os conceitos aprendidos de forma prática e eficiente, estimular o desenvolvimento acadêmico e técnico.

Tabela 3 - Ludificação: Pontuação quadro de progresso

Principiante	Explorador	Conhecedor	Junior Developer
0 - 3499	3500 - 6999	7000 – 9999	10000 ou mais

Os pontos são atribuídos através da funcionalidade etiquetas, que foram criadas com o propósito de valorizar e informar os alunos sobre os objetivos conquistados. As etiquetas permitem premiar os resultados obtidos nas tarefas, domínio dos conceitos e partilha de conhecimento, o que motiva os alunos a colaborarem na resolução dos desafios propostos. A tabela 4 lista todas as etiquetas criadas para esta finalidade, é um sistema claro de reconhecimento e incentivo aos alunos à medida que avançam no projeto e demonstram as suas competências e conhecimentos adquiridos.

Tabela 4 - Ludificação: Etiquetas e pontos por conquista

ETIQUETAS	PONTOS	SIMULADOR
01FT_POOAG	500	500
01FT_POOAG MASTER	650	
02FT_POOAG	650	500
02FT_POOAG MASTER	800	
03FT_POOAG	800	
03FT_POOAG MASTER	950	950
04FT_POOAG	950	
04FT_POOAG MASTER	1100	1100
05FT_POOAG	1100	
05FT_POOAG MASTER	1250	1250

06FT_POOAG	1250	1250
06FT_POOAG MASTER	1450	
PROJECT	1450	
PROJECT MASTER	1600	1600
ESTOU MOTIVADO	500	500
SEI INVESTIGAR	500	500
PARTILHO CONHECIMENTOS	500	
SEI FAZER REVISÃO DE CÓDIGO	500	500
POO MASTER	500	
DOMINO CONCEITO DE POLIMORFISMO	500	500
DOMINO CONCEITO DE HERANÇA	500	500
DOMINO CONCEITO DE ENCAPSULAMENTO	500	500
DOMINO O CONCEITO DE ABTRAÇÃO	500	
SEI CRIAR TAREFAS	500	
TRABALHO EM CASA	650	
TOTAL		8150 PTS

A avaliação e atribuição de pontuação aos alunos foi realizada através de uma grelha de avaliação e observação direta do desempenho em sala de aula. A tabela 5 apresenta a estrutura e os parâmetros utilizados na avaliação do desempenho dos alunos em relação a um dos objetivos propostos. A grelha permitiu uma avaliação criteriosa e detalhada, considerando diversos critérios relevantes para aferir o progresso e a qualidade do trabalho dos alunos, proporcionando um feedback claro e objetivo sobre o seu desempenho ao longo do projeto.

Tabela 5 - Ludificação: Avaliação das fichas de trabalho realizadas pelos alunos.

AVALIACAO TRABALHO INDIVIDUAL

T1					
Critérios		Aluno A	Aluno B	Aluno C	Aluno D
1. Crie o método CenaPrincipal() depois do método Start() da classe MenuUIHandler.cs adicione o código necessário para mudar de cena.	35%	0	20	20	20
2. Crie o método Sair() para sair da aplicação.					
2.1. Altere o método de forma a permitir sair do modo de reprodução (Play) no Unity Editor.	35%	0	20	10	10
3. Configure a transição de volta para a cena Menu no botão Back to menu da cena Principal (Main).	30%	0	20	20	20
	0%	20	20	20	20
	0%	20	20	20	20
	0%	20	20	20	20
	0%	20	20	20	20
Autoavaliação	0%	20	20	20	20
	100%	0%	100%	83%	83%
		0	20	16,5	16,5

Após a avaliação de cada objetivo, as etiquetas correspondentes são atribuídas ao cartão do aluno, fornecem informações sobre o objetivo alcançado e a pontuação obtida. A figura 51 ilustra as etiquetas conquistadas por um aluno da turma. O cartão de informação do aluno é composto pelas etiquetas obtidas, o grupo ao qual pertence com base na pontuação alcançada até ao momento da avaliação, o

estado do progresso, a prioridade atribuída ao desafio, a data de início e conclusão do projeto, notas informativas e uma lista de verificação de atividades realizadas.

The image shows a gamification interface with the following elements:

- Badges:** Four colored badges with a close button (X):
 - 01FT_POOAG +500PTS (pink)
 - 02FT_POOAG +650PTS (yellow)
 - 03FT_POOAG +800PTS (blue)
 - ESTOU MOTIVADO +500PTS (orange)
- Filters:** A grid of filter buttons:
 - Grupo:** Principiante (dropdown)
 - Progresso:** Em curso (dropdown)
 - Prioridade:** Urgente (dropdown)
 - Data de início:** 02/05/2022 (calendar icon)
 - Data para conclusão:** 31/05/2022 (calendar icon)
 - Repetir:** Não se repete (dropdown)
- Notas:** A text box containing the note: "Leading Progress - Quadro de progresso do projeto. Por cada tarefa concluída são atribuídos pontos. No desenvolvimento de cada tarefa deve ter em conta o tempo de resolução, eficiência do código produzido e os princípios fundamentais da POO." A checkbox "Mostrar no cartão" is present.
- Lista de verificação 3/7:** A progress bar and a checkbox "Mostrar no cartão". The list includes:
 - 01FT_POOAG: Entender funcionamento de classes, names... (checked)
 - 02FT_POOAG: Persistência da dados entre cenas (checked)
 - 03FT_POOAG: Persistência da dados entre cenas (checked)
 - 04FT_POOAG: Abstração (unchecked)

Figura 51 - Ludificação: Conquistadas obtidas por um aluno da turma.

Somente o docente da disciplina tem a permissão para efetuar a atribuição de etiquetas, bem como realizar alterações na lista de verificação e no grupo do aluno, baseando-se na pontuação obtida até o momento da avaliação. A estratégia permite ao professor manter o controlo e a precisão na avaliação dos alunos, garantir uma abordagem justa e coerente ao longo do projeto.

Através desta estrutura de cartão e etiquetas, os alunos recebem um feedback claro sobre o seu desempenho e progresso, o que os motiva a alcançar novos objetivos e a melhorar o seu desempenho. Além disso, a abordagem proporciona uma gestão mais eficiente da turma e dos resultados obtidos por aluno, permite ao professor acompanhar de perto o desenvolvimento dos estudantes e ajustar as estratégias de ensino conforme necessário.

O progresso e desempenho dos alunos é apresentado de forma global, através de um quadro desenvolvido e disponibilizado na Team da equipa. Neste quadro, todos os alunos têm acesso aos resultados e posicionamento obtido por cada um dos colegas. A figura 52 exibe o quadro na sua totalidade, permitindo uma visão abrangente e transparente do desempenho de todos os participantes.

O quadro proporciona um ambiente de competição saudável e motivador, os alunos podem acompanhar o seu progresso em relação aos colegas e visualizar as suas conquistas em relação aos objetivos propostos. Incentiva à colaboração e à troca de conhecimentos entre os alunos, uma vez que todos têm a oportunidade de aprender com as estratégias e abordagens dos colegas melhor posicionados.

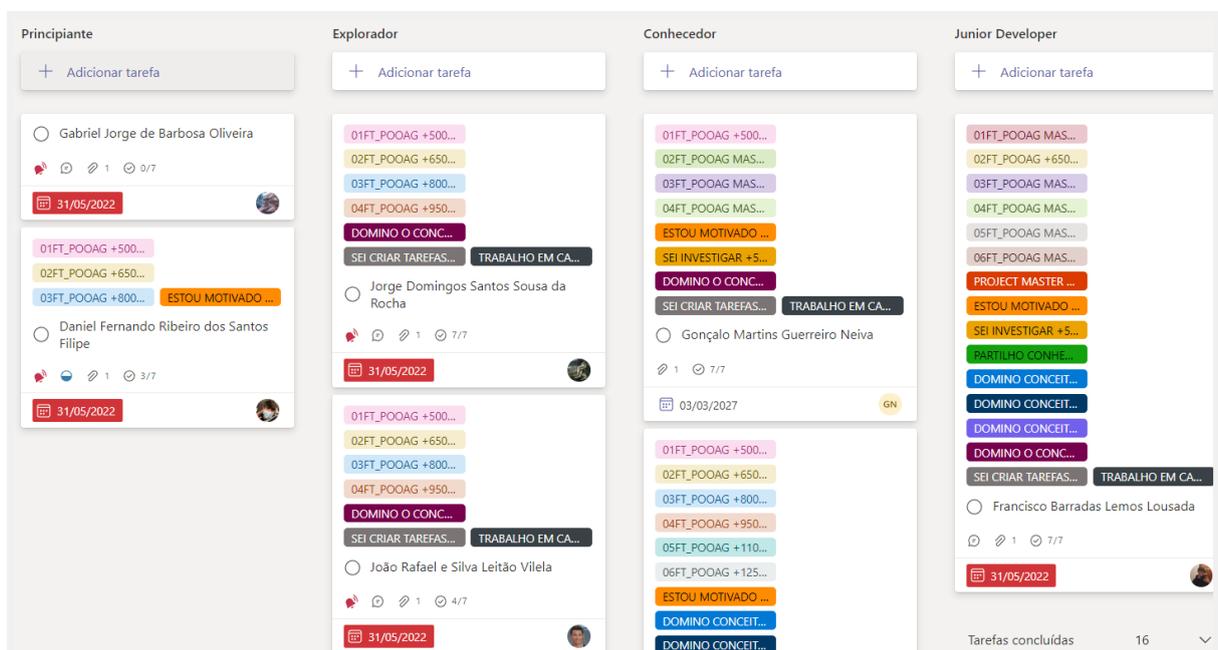


Figura 52 - Ludificação: Quadro de progresso

Além das funcionalidades mencionadas anteriormente, a aplicação Planner oferece recursos adicionais de análise e estatística que permitem ao docente uma visão aprofundada do desempenho dos alunos e facilitam a tomada de medidas para prevenir atrasos em relação à calendarização das atividades propostas. Através destes dados, o professor pode ter uma compreensão mais abrangente do progresso da turma e identificar possíveis áreas que requerem atenção especial.

A figura 53 ilustra a análise estatística que é possível obter através da ferramenta. As funcionalidades de análise e estatística oferecidas pela aplicação Planner auxiliam o docente no acompanhamento do desempenho dos alunos e na gestão eficiente do projeto.



Figura 53 - Ludificação: Análise e estatística de resultados obtidos.

Como parte da estratégia motivacional e informativa, no final de cada atividade, os alunos recebiam feedback detalhado sobre o seu desempenho, apresentado de forma visual através de gráficos estatísticos. Os gráficos forneciam informações essenciais sobre os pontos obtidos e o progresso alcançado até ao momento da avaliação, permitindo que os alunos acompanhassem o seu próprio desenvolvimento ao longo do projeto.

Para gerar o gráfico, utilizei uma tabela com os objetivos conquistados por cada aluno durante o desenvolvimento do projeto. Através desta tabela, é possível obter uma visão geral do desempenho individual de cada estudante, incluindo o número de pontos acumulados ao longo das atividades.

A tabela 6 exemplifica os dados registados por aluno que são utilizados para gerar do gráfico informativo.

Tabela 6 - Ludificação: Registo pontuação obtida por aluno.

LUDIFICAÇÃO: REGISTO PROGRESSO ALUNOS						
ETIQUETAS	PONTOS	Aluno A	Aluno B	Aluno C	Aluno D	Aluno E
01FT_POOAG	500	500	500	500	500	500
01FT_POOAG MASTER	650					
02FT_POOAG	650	650	650			650
02FT_POOAG MASTER	800			800	800	
03FT_POOAG	800	800	800			
03FT_POOAG MASTER	950			950	950	950
04FT_POOAG	950	950				
04FT_POOAG MASTER	1100			1100	1100	1100
05FT_POOAG	1100	1100				
05FT_POOAG MASTER	1250			1250	1250	1250
06FT_POOAG	1250	1250				
06FT_POOAG MASTER	1450			1450	1450	1450
PROJECT	1450	1450	0	1450	1450	0
PROJECT MASTER	1600				1600	1600
ESTOU MOTIVADO	500	500	500	500	500	500
SEI INVESTIGAR	500	500		500	500	500
PARTILHO CONHECIMENTOS	500					500
SEI FAZER REVISÃO DE CÓDIGO	500					500
POO MASTER	750					750
DOMINO CONCEITO DE POLIMORFISMO	500	500		500	500	500
DOMINO CONCEITO DE HERANÇA	500	500		500	500	500
DOMINO CONCEITO DE ENCAPSULAMENTO	500	500		500	500	500
DOMINO O CONCEITO DE ABTRAÇÃO	500	500		500	500	500
SEI CRIAR TAREFAS	500	500		500	500	500
TRABALHO EM CASA	650	650		650	650	500
		10850	2450	11650	13250	13250

O gráfico gerado é partilhado com os alunos através de mensagem privada na team da equipa.

A estratégia de ludificação motivacional apresentada mostrou-se bem-sucedida ao transformar o ambiente de ensino aprendizagem mais estimulante e produtivo. Combinar elementos de jogos, recompensas, feedback personalizado e acompanhamento detalhado, os alunos foram motivados a terem mais empenho nas tarefas, melhorar as suas competências e alcançar melhores resultados no desenvolvimento do projeto. A adoção desta estratégia pode ter um impacto importante na experiência dos alunos, estimular o interesse e a participação nas atividades.

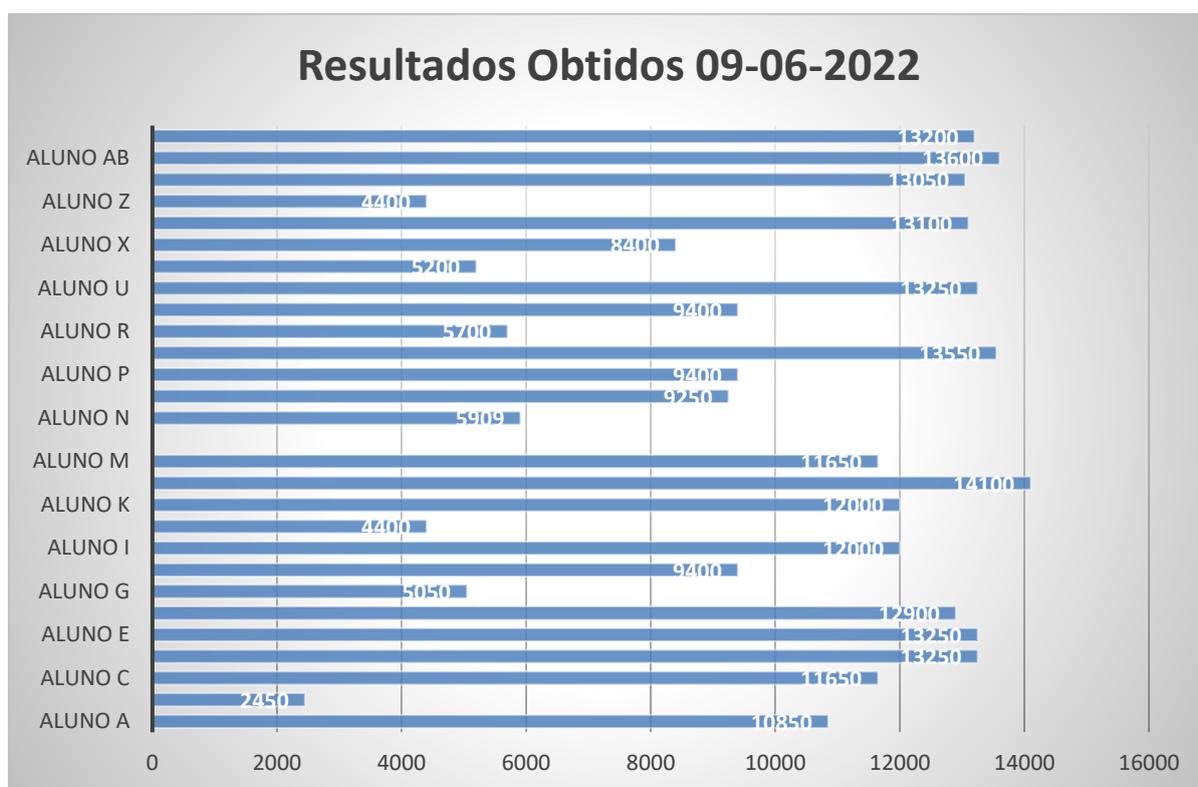


Gráfico 1 - Ludificação: Informação geral resultados obtidos.

4.6.2. Certificado de participação na Intervenção Pedagógica

Citando (Carvalho, 2018) “Na ludificação do ensino, à medida que os alunos vão cumprindo com os objetivos, são premiados.” Neste projeto como recompensa de conclusão foi atribuído aos alunos um certificado com pontuação, nível e conquistas obtidas. O objetivo deste certificado é recompensar os alunos pelo trabalho e dedicação ao projeto e motivá-los a conseguir o melhor resultado possível. A figura 54 ilustra o certificado entregue aos alunos no final da intervenção.



Figura 54 - Ludificação: Certificado de participação na intervenção Pedagógica

Após a entrega dos certificados os alunos mostraram contentamento e apreço pelo trabalho realizado. A figura 55 ilustra o momento da entrega dos certificados, esta etapa encerrou o projeto e celebrou as conquistas e esforços dos alunos ao longo deste projeto.



Figura 55 – Ludificação: entrega de certificados aos alunos.

4.7. Avaliação da intervenção

Foi realizada uma avaliação da experiência dos alunos no final da intervenção com recurso a formulário com as seguintes questões:

Considerando o objetivo do projeto de ensinar conceitos relacionados com Programação Orientada a Objetos e a sua aplicação em qualquer linguagem orientada a objetos, conseguiu atingir o objetivo?

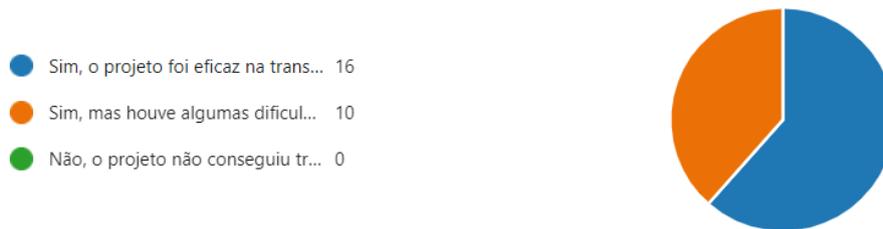


Figura 56 - Avaliação da intervenção: Objetivos

61,5% respondeu sim, o projeto foi eficaz na transmissão dos conceitos e sua aplicação, 38,5% considera que sim, mas houve algumas dificuldades na compreensão e aplicação dos conceitos, 0% Não, o projeto não conseguiu transmitir adequadamente os conceitos e sua aplicação.

A maioria dos alunos considerou o projeto eficaz na transmissão dos conceitos relacionados com a POO e aplicação em qualquer linguagem. Por outro lado, uma percentagem significativa considerou eficaz, mas tiveram alguma dificuldade na compreensão e aplicação dos conceitos. Nenhum aluno considerou a transmissão inadequada dos conceitos e aplicação prática, no geral julgo que o projeto obteve resultados positivos em relação ao objetivo proposto.

Acredita que o projeto contribuiu para melhorar a compreensão dos conceitos de Programação Orientada a Objetos?

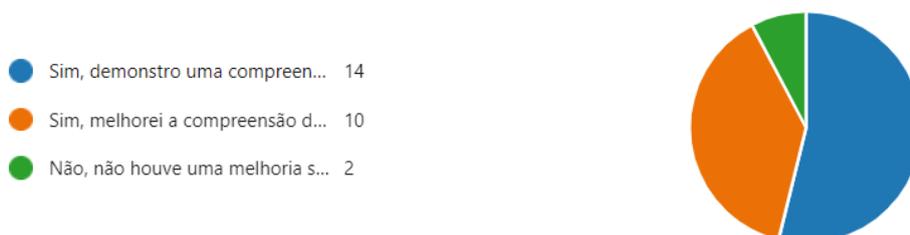


Figura 57 - Avaliação da intervenção: Compreensão dos conceitos

54% dos alunos respondeu sim, demonstro uma compreensão sólida dos conceitos após o projeto. 38% sim, melhorei a compreensão dos conceitos após o projeto e 8% não, não houve uma melhoria

significativa na compreensão dos conceitos.

De acordo com as respostas obtidas, a maioria dos alunos considera que o projeto contribuiu para uma compreensão sólida dos conceitos de POO. Um percentagem significativa dos alunos afirmam que o projeto os ajudou a melhorar a compreensão dos conceitos. No entanto, dois alunos não sentiram uma melhoria significativa na compreensão dos conceitos após o projeto, o que não compromete os resultados sabendo que no questionário realizado na atividade Envelope Secreto, dois alunos responderam ter níveis avançados de compreensão dos pilares. Posso assim concluir que, os resultados indicam que o projeto teve um impacto muito positivo na compreensão dos conceitos pelos alunos.

Aplicaste os conceitos aprendidos no projeto em projetos individuais ou em grupo?

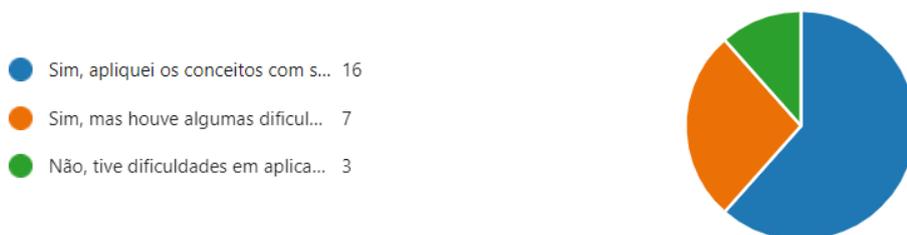


Figura 58 - Avaliação da intervenção: Aplicação dos conceitos aprendidos.

61,5% respondeu sim, apliquei os conceitos com sucesso, Já 27% dos alunos considerou que sim, mas houve algumas dificuldades na aplicação dos conceitos e 11,5% respondeu não, tive dificuldades em aplicar os conceitos em projetos adicionais.

De acordo com as respostas obtidas, a maioria dos alunos consegue aplicar os conceitos aprendidos no projeto com sucesso em projetos individuais ou em grupo. 27% dos alunos aplica os conceitos, mas encontra alguma dificuldade na aplicação. Uma pequena percentagem de alunos teve dificuldades na aplicação dos conceitos. Posso concluir que a maioria dos alunos foi capaz de aplicar os conceitos aprendidos no projeto em outros contextos de trabalho, o que sugere que o projeto teve um impacto positivo na sua capacidade de aplicar conhecimentos.

Em geral, qual a tua avaliação do projeto em termos de organização, clareza das explicações, recursos utilizados e apoio fornecido?



Figura 59 - Avaliação da intervenção: Organização, Clareza, Recursos e Apoio.

42,3% dos alunos considerou excelente, o projeto foi muito bem organizado e as explicações foram claras, 46,2% considerou bom, o projeto teve uma organização adequada e as explicações foram compreensíveis, 11,5% achou regular, houve algumas falhas na organização e/ou nas explicações fornecidas. Para 0% o projeto foi fraco, o projeto apresentou sérias deficiências na organização e/ou nas explicações.

Analisando os resultados, a maioria dos alunos considerou o projeto como "Bom", indicando que a organização foi adequada e as explicações foram compreensíveis. uma grande percentagem de alunos avaliou o projeto como "Excelente", indica que a organização foi muito bem feita e as explicações foram claras. Três dos alunos classificaram o projeto como "Regular", o que indica que houve algumas falhas na organização e/ou nas explicações fornecidas ou a falta de conhecimento prévio condicionou a interpretação. Nenhum aluno classificou o projeto como "Fraco". Posso concluir que os resultados mostram que a grande maioria dos alunos teve uma avaliação positiva, destacando a sua organização, clareza das explicações e recursos utilizados, o que é um indicador de que o projeto foi bem-sucedido na perspetiva dos alunos.

No geral, os resultados do questionário indicam que o projeto de intervenção pedagógica foi bem-sucedido na medida em que a transmissão dos conceitos de POO foi interessante para os alunos aprenderam de forma motivante e sentiram-se envolvidos. A combinação das abordagens inovadoras, construcionismo e ludificação, parece ter sido eficaz na promoção da compreensão dos conceitos e na aplicação prática dos conhecimentos adquiridos pelos alunos. Os resultados também sugerem que a maioria dos alunos ficou satisfeita com a organização do projeto e a clareza das explicações fornecidas, o que é fundamental para o sucesso de qualquer intervenção pedagógica.

4.8. Inovação

O discurso sobre o novo é algo que se identifica com a palavra inovação, explorar, cada vez mais, nos espaços formais e informais de educação, como uma mudança que traduz resultados (Dussel, 2013).

A abordagem tradicional, dá importância às situações de sala de aula, onde os alunos são “instruídos” e “ensinados” pelo professor. É um ensino centrado no professor, que vê o aluno sobretudo como um recetor de conhecimentos (Estrada, 2013).

O meu projeto de intervenção alterou completamente a abordagem utilizada em sala de aula pelo professor cooperante. Foi um grande desafio, mas apoiado desde o início, sendo a única restrição a metodologia de avaliação das unidades curriculares. A avaliação foi realizada através do projeto de consolidação de conhecimentos, projeto que foi na integra desenvolvido por mim com a colaboração do professor cooperante e pode ser consultado no ponto anterior.

Os resultados obtidos nos questionários realizados, indicam que a aplicação da teoria construtivista através da integração do Construcionismo foi bem recebida pelos alunos, e teve um impacto significativo no ensino e aprendizagem dos conceitos, desenvolveu o pensamento crítico, ajudou na resolução de problemas, trabalho colaborativo e criatividade. Considero que as sociedades estão a evoluir rapidamente, a educação precisa de acompanhar as mudanças para preparar os alunos para o futuro, esta pode ser uma estratégia de consolidação do ensino tradicional com as necessidades das empresas do setor visto que foi estimulada a utilização de metodologias e ferramentas utilizadas na atualidade por grande parte dessas empresas.

A Ludificação da ação pode ter contribuído positivamente na construção de conhecimento e partilha, a integração de elementos de jogos na educação pode aumentar a motivação dos alunos e tornar o processo de aprendizagem mais envolvente e divertido. Os alunos acompanharam e discutiram resultados obtidos, procuraram melhorar e recuperar posições mesmo quando já se encontravam atrasados relativamente ao calendário proposto.

A abordagem construcionista e a ludificação podem contribuir para a transformação do sistema educacional, promovem uma educação mais centrada no aluno, alinhada com as necessidades do mundo contemporâneo.

Considero que este projeto foi sem sombra de dúvida uma mais-valia para o curso, disciplina e alunos, pode ser aplicado noutras disciplinas e contextos educativos. O trabalho interdisciplinar realizado pode fomentar o rápido desenvolvimento de projetos e prepará-los para enfrentar desafios mais complexos no futuro.

4.9. Outras atividades e participação

Durante a realização deste projeto tive a oportunidade de participar ativamente na organização da exposição anual intitulada EXPOCOLGAIA 2022. Esta exposição, realizada nas instalações do Colégio de Gaia, tem como objetivo divulgar a oferta formativa da instituição e apresentar os trabalhos realizados nos cursos com planos próprios.

Na qualidade de formador há 7 anos, promovo os trabalhos desenvolvidos em diversas disciplinas, no curso de Eletrónica Computadores e Telecomunicações, os alunos apresentaram aplicações Microsoft Windows criadas para controlar projetos desenvolvidos noutras disciplinas, utilizando a plataforma Arduino. Já na disciplina de Tecnologias e Desenvolvimento Multimédia, do 12.º ano de Informática e Tecnologias Multimédia, os alunos representaram a disciplina apresentando os jogos desenvolvidos ao longo do ano letivo. Na área de Computação Gráfica e Animação 3D, do curso de Comunicação Multimédia, promovemos demonstrações de modelação 3D durante a exposição e apresentamos alguns trabalhos desenvolvidos ao longo do ano letivo. Foi uma oportunidade emocionante para os alunos mostrarem os conhecimentos e criatividade no campo da multimédia.

Além disso, no ano letivo 21/22, assumi o cargo de diretor de turma do 12.º ano do curso de Informática e Tecnologias Multimédia, pelo 4º ano consecutivo. A função de gestão tem sido muito gratificante para mim, pois conduzo os alunos com muito orgulho no cumprimento de suas obrigações e acompanhamento de perto o seu crescimento e desenvolvimento ao longo do percurso académico.

Estas experiências permitiram-me não só contribuir para a divulgação dos trabalhos dos alunos e das disciplinas que leciono, mas também desempenhar um papel fundamental na orientação e apoio aos estudantes como diretor de turma. Estou extremamente satisfeito com o resultado alcançado e com o envolvimento dos alunos em todas as atividades promovidas durante a EXPOCOLGAIA 2022.

5. Conclusão, limitações e recomendações

5.1. Conclusão

Quando me lancei neste novo desafio académico não tinha a menor ideia das mudanças que poderiam advir desta caminhada. Foram anos difíceis, atormentados por uma crise pandémica sem igual, problemas familiares que causaram enorme desconforto. Procurei ser sempre cumpridor das minhas obrigações, não comprometi e prova disso é estar a fechar este capítulo, espero, da melhor forma possível. O curso de Mestrado em Ensino de Informática na Universidade do Minho foi muito trabalhoso, com uma componente prática muito exigente, promoveu na generalidade das disciplinas o trabalho em equipa o que na minha opinião é muito enriquecedor e motivante. Ser trabalhador-estudante com cargos e família foi muito desafiante. Estar comprometido com 24 horas letivas mais 6 horas de estágio e aulas na universidade do Minho roubaram espaço à família e amigos que espero que esta valiosa perda seja recompensada de futuro.

Da minha experiência de ensino, já tinha tido a oportunidade de trabalhar com alunos das turmas de ITM no 11.º Ano em ciclos anteriores. Sou professor do curso há 6 anos e trabalho maioritariamente ao nível do 12.º ano, neste curso, na disciplina de Técnicas de Desenvolvimento Multimédia. Na disciplina ensino os alunos a trabalhar com ferramentas de gestão de projetos, interfaces, áudio, vídeo e projeto. Introduzi na escola o Unity como ferramenta de desenvolvimento e ensino os alunos a desenvolver jogos desde então. A escolha deste trabalho não acontece por acaso, a disciplina que leciono perdeu duas horas letivas, na última revisão, o que torna quase impraticável a construção de jogos a partir do zero. Esta foi a oportunidade para fazer a ligação e voltar ao caminho. Esta oportunidade de introduzir as ferramentas de gestão e desenvolvimento no 11.º ano pode ser eficaz a colmatar a necessidade premente de horas no 12.º. Poderia retomar a normalidade do meu projeto de ensino e garantir que os alunos no final do ano letivo entregam trabalhos de elevada qualidade.

Sinto ter cumprido com os objetivos propostos. As aulas foram cuidadosamente estruturadas e a melhoria continua do desenvolvimento vai permitir resolver problemas sentidos pelos alunos. Reconheço que trabalhar com alunos com um elevado nível de conhecimento facilitou a minha abordagem, estudei o meu público-alvo e apresentei uma proposta à medida das capacidades demonstradas previamente pelos alunos.

Em resposta à minha questão de investigação - É possível desenvolver o pensamento abstrato dos alunos através de um ensino centrado na aprendizagem baseada em jogos? - considero que o projeto, baseado na abordagem construcionista e na ludificação como estratégia motivacional, demonstrou ser eficaz. A

integração de elementos de jogos, como narrativas envolventes, desafios estimulantes e recompensas atrativas, revelou-se um meio valioso para alcançar o objetivo. Além de estimular o interesse e a motivação dos alunos, tornou o processo de aprendizagem mais envolvente e divertido.

Para verificar o desenvolvimento do pensamento abstrato, utilizei a aplicação dos sete conceitos (sequências, ciclos, eventos, paralelismo, condições, operadores e dados) definidos por (Brennan & Resnick, 2012), que, em conjunto, representam a aplicação do pensamento abstrato.

Os resultados dos questionários revelaram que a maioria dos alunos considerou que o projeto foi bem-sucedido na transmissão dos conceitos de POO e na sua aplicação em projetos individuais e em grupo. Os alunos consideram ainda que tiveram uma melhoria significativa na compreensão dos conceitos após a realização do projeto, estes fatores indicam que o ensino centrado na aprendizagem baseada em jogos teve um impacto positivo no desenvolvimento do pensamento abstrato dos estudantes.

A abordagem construcionista, que incentiva os alunos a construir o seu conhecimento através da criação de projetos práticos, e a ludificação, que utiliza elementos de jogos para motivar a participação ativa dos alunos, mostraram-se complementares e eficazes na promoção do pensamento abstrato. Através do trabalho colaborativo, da resolução de problemas e da aplicação prática dos conceitos, os alunos desenvolveram habilidades de pensamento crítico e analítico, essenciais para o pensamento abstrato.

Portanto, o projeto de intervenção demonstrou que é possível desenvolver o pensamento abstrato dos alunos através de um ensino centrado na aprendizagem baseada em jogos. A combinação de abordagens inovadoras e estratégias motivacionais pode transformar o processo educativo, tornando-o mais centrado no aluno e alinhado com as necessidades do mundo contemporâneo. O incentivo à criatividade, colaboração e autonomia na construção do conhecimento proporcionou aos alunos uma experiência de aprendizagem enriquecedora, que contribuiu para o seu desenvolvimento pessoal e enriquecedor a nível curricular.

5.2. Limitações

Uma das principais limitações sentidas com este projeto foi a gestão do tempo. Como mencionei anteriormente, a minha carga de trabalho era substancial. Equilibrar 24 horas letivas semanais com as 6 horas de estágio e as aulas na Universidade do Minho representou um desafio considerável. Esta limitação não só afetou a minha disponibilidade para a família e amigos, mas também restringiu a minha capacidade de dedicar tempo adicional ao projeto e aos alunos. Embora tenha demonstrado resiliência ao enfrentar este desafio, não posso ignorar os efeitos que a sobrecarga de trabalho teve no desenvolvimento do projeto e nas atividades planeadas.

5.2. Recomendações

O projeto educativo é uma tarefa desafiante que requer grande dedicação e esforço por parte do professor, além de um público-alvo motivado e bem informado. Com o intuito de otimizar o impacto do trabalho docente, recomendo a implementação de contínuos processos de avaliação, validação e controle que devem incluir Avaliação Contínua: Práticas de avaliação constantes para medir o progresso dos alunos e identificar áreas que necessitam ser melhoradas, o que permitirá que os professores façam ajustes ao longo do projeto. Feedback Qualitativo: Garantir feedback construtivo aos alunos regularmente, destacando os seus pontos fortes e áreas a desenvolver, desta forma contribuir para o seu envolvimento e crescimento. A Personalização do Ensino: adaptando as estratégias de acordo com as necessidades e níveis de conhecimento dos alunos, um ensino mais personalizado é mais eficaz. Avaliação de Resultados: Medir o sucesso do projeto com base em objetivos claros e mensuráveis, analisar os resultados para entender o impacto do projeto na aprendizagem dos alunos. Formação Contínua: Para os professores, de forma a garantir que estão atualizados com as melhores práticas e estratégias educacionais no sentido de ajuda a manter a qualidade do ensino.

Referências

- Barbosa, A. (2006). *Conhecimento tácito substantivo histórico sobre o encontro entre povos e culturas : um estudo com alunos dos 7.º e 10.º anos de escolaridade*. Universidade do Minho, Instituto de Educação e Psicologia. Braga: Universidade do Minho.
- Barradas, R., Lencastre, J. A., Soares, S., & Valente, A. (2019). Desenvolvimento do pensamento computacional em idades precoces usando a plataforma code.org.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. p. 25.
- Cabral, M. F. (2022). *Mais que um simples jogo: as potencialidades da gamificação no ensino da História*. Porto: Faculdade de Letras, Universidade do Porto.
- Carvalho, G. M. (2018). *"Gamificação" no ensino de programação: estudo de uma estratégia pedagógica para sucesso na aprendizagem*. Braga: Universidade do Minho.
- Connor, R. (Janeiro de 2003). Abstract data type. *Encyclopedia of Computer Science*, p. 5.
- Costa Estrada, C. M. (2013). *A compreensão Evolutiva do Ensino e da Aprendizagem como Promoção da Melhoria da Prática Docente*. Porto: Universidade Católica Portuguesa.
- Costa, R. F. (2022). *Gamificação em Ensino - Estudo preparatório em duas turmas de Ensino Profissional*. Lisboa: ISTECS.
- Cruz, S., Bento, M., & Lencastre, J. A. (2020). Construção de jogos em Scratch sobre conceitos da Matemática. *5º Encontro sobre jogos e mobile learning* (p. 10). Centro de Estudos Interdisciplinares do Século 20 (CEIS20) & Universidade de Coimbra.
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining Gamification. *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*, 8. doi:10.1145/2181037.2181040
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). Gamification: Using game design elements in non-gaming contexts. *Proceedings of the International Conference on Human Factors in Computing Systems*. CHI Extended Abstracts. doi:10.1145/1979742.1979575
- Dussel, I. (2013). Currículo e Inovações Tecnológicas no Contexto Escolar. (R. Linhas, Entrevistador) Vera Lucia Gaspar da Silva.
- Eliens, A. (2000). *Principles of Object-Oriented Software Development*. Boston: Publishing Co., Inc.
- Estrada, C. M. (2013). *A compreensão Evolutiva do Ensino e da Aprendizagem como Promoção da Melhoria da Prática Docente*. Porto: Universidade Católica Portuguesa.

- Figueiredo, A. D. (Julho/Setembro de 2016). A Pedagogia dos Contextos de Aprendizagem.
- Gomes, J. L., & Filho, N. A. (2008). Jogos: a importância no processo educacional. *Programa de Desenvolvimento Educacional – PDE - 2008, da Secretaria de Estado da Educação*, p. 32.
- Martins, A. R., & Oliveira, L. R. (2018). *Motivação e aprendizagem através da criação de jogos educativo*. Indagatio Didactica.
- Mesquita, I. (2021). Educação, Território e desenvolvimento Humano. (U. Católica, Ed.) *Atas do IV seminário Internacional*, p. 20.
- Neto, H. R. (2015). *Gamificação: engajando pessoas de maneira lúdica*. São Paulo: FIAP.
- Nunes, S. d., & Santos, R. P. (Novembro de 2013). O Construcionismo de Papert na criação de um objeto de aprendizagem e sua avaliação segundo a taxionomia de Bloom. *Conference: IX ENPEC - Encontro Nacional de Pesquisa em Educação de Ciências*, p. 9.
- Oliveira, D. A. (1997). *Gestão Democrática Da Educação. Desafios Contemporâneos*. Brasil: Vozes.
- Oliveira, L. (2019). *Motivação e Aprendizagem através da Criação de Jogos Educativos*. Universidade do Minho, Ciências da Educação. Braga: Universidade do Minho. Obtido em 2023, de <https://hdl.handle.net/1822/76831>
- Osório, C. M. (2015). *O desenvolvimento do pensamento abstrato através de game-based learning e do Scratch*.
- Peterpunk, P. (2013). *Design A Game*. Obtido de <https://designagame.eu/2013/12/unity-popular-videogame-development/>.
- Santos, M. L., & Perin, C. S. (2013). Os Desafios da Escola Pública Paranaense na Perspectiva do Professor PDE. *Governo do Estado*, p. 24. Obtido de http://www.diaadiaeducacao.pr.gov.br/portals/cadernospede/pdebusca/producoes_pde/2013/2013_fafipa_ped_artigo_maria_lucia_dos_santos.pdf
- Schunk, D. H. (2012). *Learning Theories An Educational Perspective*. Boston: Pearson Education, Inc.
- Tykhomyrov, O. (22 de Novembro de 2002). Introducing to Object-Oriented Programming. *Seventh College on Microprocessor-Based Real-Time Systems in Physics*, p. 33.
- Unity. (2022). <https://unity.com/pt/learn>. Obtido de unity.com/pt: <https://unity.com/pt>
- Wikipédia. (2022). pt.wikipedia.org/wiki/. Obtido de Wikipedia: <https://pt.wikipedia.org/wiki/>
- Yildirim, O. G., & Ozdener, N. (2021). An Action Research Study on the Development of Object-Oriented. *International Journal of Technology in Education and Science*, 29.

Apêndices

Resumo do projeto

Resumo do projeto

Conceito da aplicação

Resumo

O projeto a ser desenvolvido permitirá implementar funcionalidade numa aplicação que simula a gestão de recursos. A gestão de recursos está no centro de muitos géneros de jogos mas as simulações são também comumente usadas em contextos educacionais (por exemplo, para explorar sustentabilidade e economia) assim como na indústria.



Este tipo de projeto é particularmente adequado para ajudá-lo a explorar sistemas de programação e arquitetura e muitas vezes incluem:

- Interação do utilizador, para permitir que influenciem a simulação;
- Transições entre cenas, para permitir maior personalização por parte do utilizador;
- Sistemas projetados para extensão, para aumentar a complexidade da simulação.

Exemplos de referência:

Fishbanks: Uma simulação multijogador de gestão de recursos renováveis para uso educacional

Age of Empires ou **Civilization:** jogos de estratégia clássicos que exigem gestão de recursos.

Lista de tarefas:

Gestão de Cenas

- Crie transições entre duas cenas.
- Configure botões para que o utilizador possa controlar as transições criadas.

Configure um botão para sair da aplicação (ou sair do modo Play, no Unity Editor).

Persistência de dados

- Configure botões numa cena para aplicar uma cor escolhida a objetos de uma segunda cena.
- Salve a última cor escolhida pelo utilizador e garanta que fica pré-selecionada na próxima vez que a aplicação for executada.

Herança e Polimorfismo

- Crie um tipo de objeto na simulação com uma variação de comportamento derivada de uma classe base.

Abstração

- Analise o código para reduzir duplicados e melhorar a reutilização.

Encapsulamento

- Utilize *getters* e *setters* para proteger os dados contra uso indevido.

Otimize o código

- Analise o código exemplo para identificar problemas básicos de otimização.

Visão geral do projeto

Importante: Ao abrir o projeto pela primeira vez a simulação do armazém terá funcionalidades básicas, mas a aplicação não funcionará!

A aplicação

Cenas no projeto

Há duas cenas no projeto Unity:

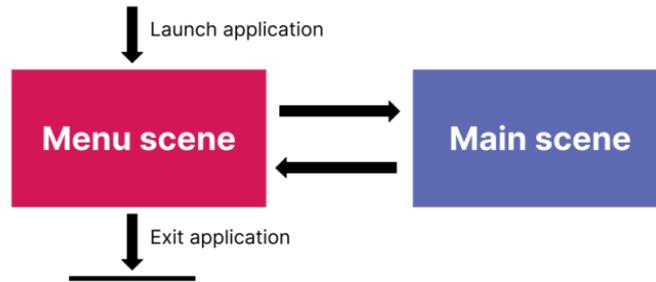
1. Um menu iniciar (Menu) onde, os utilizadores podem iniciar a simulação e fechar a aplicação.
2. Uma cena de simulação (Principal) que é estilizada como um armazém.

Nota: Uma cena adicional chamada Otimização — caso de estudo de otimização que será usada para traçar o perfil do código e identificar problemas.

Interação do utilizador

Irá implementar transições controladas pelo utilizador:

- Entre as duas cenas;
- Para sair da aplicação, apenas a partir da cena Menu.



Interações do utilizador

O utilizador precisa ser capaz de:

- Iniciar a cena de simulação a partir do menu iniciar;
- Voltar ao menu iniciar da cena de simulação;
- Sair da aplicação (ou sair do modo Play, de testes no Editor);
- Selecionar uma cor para aplicar às Unidades Transportadoras (empilhadores) na simulação.

NOTA: A última cor escolhida pelo utilizador também deve ser pré-selecionada na próxima vez que iniciar a aplicação.

Simulação do armazém

Simulação básica

A simulação básica (na cena principal) que contém:

- Dois tipos diferentes de Pilha de Recursos (objetos numa palete) que, produzem recursos “itens” a uma velocidade de 0,5 por segundo (/s)
- Dois tipos de Unidade:
 - Uma Unidade (trabalhador) que se pode mover mas, não tem mais funcionalidades.
 - Uma unidade de transporte (empilhador), que o utilizador pode definir para transportar “itens” para uma base predefinida (o espaço na frente do camião marcado com um círculo vermelho)

Funcionalidades básicas da simulação

Na simulação básica, o utilizador pode:

- Usar as teclas de seta (ou WASD) para mover a câmara pelo espaço do armazém.
- Clique com o botão esquerdo em qualquer objeto definido para selecioná-lo e abrir uma sobreposição de IU com os seus detalhes.
- Quando uma unidade é selecionada, clique com o botão direito do rato:
 - Em qualquer local, para mover o utilizador para essa localização;
 - Numa Pilha de Recursos, para configurá-la para transportar itens de recursos do objeto para a Base.

Requisitos de simulação adicionais

Ao trabalhar neste projeto vai:

- Criar uma Unidade de Produtividade que aumenta a produção de itens de recurso numa base e aplicá-lo ao trabalhador no armazém.
- Refinar e otimizar o código criado, aplicando princípios de Programação Orientada a objetos.

Estilo do projeto

A temática deste projeto é um armazém, num estilo low poly. Se quiser personalizar o tema do projeto enquanto trabalha nesta funcionalidade, pode importar os seus próprios ativos e trocá-los.

Scripts fornecidos

O projeto vem com sete scripts principais que foram parcial ou completamente escritos. Existem alguns scripts adicionais em subpastas que pode visualizar, mas não estão incluídos aqui.

Base.cs

Esta é uma subclasse da classe Building. Não altera o comportamento da classe, mas armazena uma referência (função Awake). Significa que pode extrair a base de qualquer lugar. É usada pelas Unidades quando elas voltam à base.

Building.cs

Esta é uma classe base abstrata de duas subclasses: ResourcePile e Base. Mantém um inventário do que os membros da classe contêm e também suporta a adição e remoção de “itens” desse inventário. Também implementa IUIInfoContent, a sobreposição básica da interface do utilizador na simulação.

ColorHandler.cs

Script que permite criar botões de cores para que os utilizadores possam selecionar uma cor e aplicar às Unidades de Transporte a partir do menu iniciar.

ResourcePile.cs

Esta é uma subclasse da classe Building. Leva uma referência a um ResourceItem que irá produzir e armazenar as seguintes informações sobre ele numa classe Serializable:

- nome
- ID exclusivo
- sprite associado

Define uma função de atualização e incrementa um contador de produção pela velocidade/s de produção a cada atualização. Uma vez acima de 1, cria recursos e diminui o contador.

Nota: também substitui a função GetData para o IUIInfoContent de Building para dar ao InfoPanel a sua velocidade de produção como informação.

TransporterUnit.cs

Esta é uma subclasse da Unit.cs. Contém:

- O destino (Pilha de Recursos) para o qual uma Unidade de Transporte está atualmente definida;
- A quantidade de recursos que pode transportar de uma só vez;
- O que está a ser transportado atualmente pela Unidade Transportadora (Empilhador).

A subclasse TransporterUnit substitui três funções de sua classe base:

- Comportamento à volta dos edifícios nas proximidades;
- Duas funções GoTo, para reagir quando o alvo da unidade de transporte muda **Unit.cs**.

Esta é uma classe base abstrata para todas as Unidades. Controla Unidade:

- Movimento, incluindo um ponto de destino para o qual se mover (que pode ser um Edifício ou uma posição);
- Cor, quando configurado;
- Comportamento quando estiver no alcance de seu edifício alvo.

UserControl.cs

Esta classe lida com os inputs do utilizador. Move a Câmara no plano xz ao ler os valores vertical e horizontalmente. Também controla a funcionalidade de clique com o botão esquerdo e direito.

Planificação de aulas

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	Configuração do Ambiente de desenvolvimento; <ul style="list-style-type: none"> Unity: Interface e Primeiros Scripts. GitHub: Configuração de conta e Projeto Apresentação do projeto a desenvolver	Aula N°	1	
		Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> Explicar o ambiente de desenvolvimento mostrando os seus elementos. Aprender a trabalhar com ferramenta de versionamento de projeto. 		•		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> Efetuar a chamada e registo de faltas; Registar o sumário e apresentar os objetivos da aula; Ferramentas adotadas no módulo. <ul style="list-style-type: none"> GitHub Desktop – Apresentação da ferramenta, instruções obrigatórias; (Convidar os alunos a participar) <ul style="list-style-type: none"> Configuração de conta e primeiro projeto. Unity – Apresentação da ferramenta, instruções Obrigatórias; <ul style="list-style-type: none"> Canvas Teams integração de Agile Application para Gamificação; Objetivos próxima aula Avaliação da sessão 	<ul style="list-style-type: none"> Computador Videoprojector Plataforma escolar Ficha de trabalho GitHub Desktop Unity Engine Visual Studio 	<ul style="list-style-type: none"> Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> 2min 5min 40min 5min 8min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento POO em ambiente gráfico

Sumário:	<ul style="list-style-type: none"> • Ferramenta de Gestão de Projetos • Funcionamento de recompensas • Unity: Interface e Primeiros Scripts. 	Aula Nº	2	
		Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> • Explicar metodologia de avaliação. • Criação de classes, instâncias, métodos. • Chamar métodos, Criação de objetos. 		•		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registrar o sumário e apresentar os objetivos da aula; • Rever os conteúdos da aula anterior através de debate e exposição teórica. Os quatro pilares da POO; • Criação de classe, propriedades e métodos • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • Ficha de trabalho • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 5 min • 35min • 5min • 8min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento POO em ambiente gráfico

Sumário:	Ficha de Trabalho N01: <ul style="list-style-type: none"> • Entender o funcionamento de classes; • Entender o funcionamento de Bibliotecas de Classes; • Criação de Métodos. • Eventos 	Aula Nº	3	
		Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> • Conceitos básicos: Classes, Objetos, Atributos, Métodos, Abstração, Herança, Polimorfismo e Encapsulamento. • Apresentar e explicar o ambiente de desenvolvimento mostrando os seus elementos. 		<ul style="list-style-type: none"> • Ferramentas de trabalho 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registar o sumário e apresentar os objetivos da aula; • Rever os conteúdos da aula anterior através de debate e exposição teórica. Os quatro pilares da POO;(Mentimeter) • Resolução da ficha de trabalho N01 • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 5min • 40min • 5min • 3min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	Continuação da Ficha de Trabalho N01: <ul style="list-style-type: none"> • Entender o funcionamento de classes; • Entender o funcionamento de Bibliotecas de Classes; • Criação de Métodos. • Eventos 	Aula Nº	4	
		Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> • Conceitos básicos: Classes, Objetos, Atributos, Métodos, Abstração, Herança, Polimorfismo e Encapsulamento. • Apresentar e explicar o ambiente de desenvolvimento mostrando os seus elementos. 		<ul style="list-style-type: none"> • Objetos • Métodos 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registrar o sumário e apresentar os objetivos da aula; • Continuação da resolução da ficha de trabalho N01; • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 45min • 5min • 8min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	<ul style="list-style-type: none"> • Conclusão da ficha de trabalho N01. • Ficha de trabalho N02: <ul style="list-style-type: none"> ○ Implementar persistência de dados entre cenas; ○ Criação de classes: Instâncias de classes. ○ Singleton 	Aula N°	5	
		Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<p>Aprender a usar persistência de dados para preservar informações em diferentes cenas, selecionando uma cor que o utilizador seleciona na cena Menu e aplicá-la às unidades de transporte na cena Principal. No final deverá ser capaz de:</p> <ul style="list-style-type: none"> • Garantir que os dados são preservados durante uma sessão da aplicação utilizando o método Unity DontDestroyOnLoad; • Reconhecer quando usar membros de classes estáticos e singletons para implementar persistência de dados; • Modificar um GameObject com um script que contém as variáveis a serem guardadas entre Cenas. 		<ul style="list-style-type: none"> • Classes • Objetos • Atributos • Métodos 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registrar o sumário e apresentar os objetivos da aula; • Correção da ficha de trabalho N01 • Apresentação dos objetivos da aula, resolução da ficha de trabalho N02; • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 5min • 40min • 5min • 8min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	Conclusão e correção da ficha de trabalho N02	Aula Nº	6	
	Ficha de trabalho N03: <ul style="list-style-type: none"> • Trabalhar com Classes, bibliotecas de classes. Atributos, Métodos e Objetos • Implementar persistência de dados entre sessões; • Trabalhar com ficheiros: Escrever e Ler dados com JSON. 	Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> • Chamar os métodos de inicialização apropriados na sequência de inicialização da aplicação; • Armazenar e organizar dados aplicando estruturas de dados, como listas e dicionários; • Guarda e Carregar dados do utilizador entre sessões. 		<ul style="list-style-type: none"> • Métodos • Estruturas de dados 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registar o sumário e apresentar os objetivos da aula; • Correção da ficha de trabalho N03; • Resolução da ficha de trabalho N04 • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • Ficha de trabalho • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 5min • 40min • 5min • 8min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	<p>À conversa com dois especialistas: Metodologias de trabalho adotadas pelas empresas de desenvolvimentos de software com os convidados</p> <ul style="list-style-type: none"> Manuel Marinho - Ubiquity Technology Daniel Duarte - Natixis Groupe BPCE's Global Financial <p>Continuação da ficha de trabalho N03:</p> <ul style="list-style-type: none"> Trabalhar com Classes, bibliotecas de classes. Atributos, Métodos e Objetos Implementar persistência de dados entre sessões; Trabalhar com ficheiros: Escrever e Ler dados com JSON. 	Aula Nº	7	
		Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> Entender a importância da utilização das ferramentas de apoio ao projeto 		<ul style="list-style-type: none"> Gir Agile 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> Efetuar a chamada e registo de faltas; Registar o sumário e apresentar os objetivos da aula; Apresentação dos objetivos da aula: <ul style="list-style-type: none"> Conversa com especialistas Continuação da ficha de trabalho N03 Objetivos próxima aula Avaliação da sessão 	<ul style="list-style-type: none"> Computador Videoprojector Plataforma escolar Ficha de trabalho GitHub Desktop Unity Engine Visual Studio 	<ul style="list-style-type: none"> Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> 2min 5min 30min 20min 2min 3min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	Conclusão e correção da ficha de trabalho N03.			Aula Nº	7
	Ficha de trabalho N04: <ul style="list-style-type: none"> • Abstração; • Restruturação de scripts; 			Duração	60 minutos
Objetivos de Aprendizagem:			Conteúdos:		
<ul style="list-style-type: none"> • Explicar como a abstração é utilizada para expor apenas os componentes do script necessários. • Expor apenas os detalhes importantes de um objeto reconhecendo corretamente as oportunidades para implementar a abstração. 			<ul style="list-style-type: none"> • Abstração 		
Estratégias	Atividades	Recursos	Avaliação	Gestão	
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registar o sumário e apresentar os objetivos da aula; • Correção da ficha de trabalho N03 • Apresentação dos objetivos da aula, resolução da ficha de trabalho N04; • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • Ficha de trabalho • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 5min • 40min • 5min • 8min 	

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	Conclusão e correção da ficha de trabalho N04.			Aula Nº	8
	Ficha de trabalho N05			Duração	60 minutos
Objetivos de Aprendizagem:			Conteúdos:		
<ul style="list-style-type: none"> • Criar um novo tipo de objeto (unit type); • Trabalhar Herança; • Trabalhar Polimorfismo. • Trabalhar com notação de substituição (Override) • Trabalhar com sobrecargas (Overloads) 			<ul style="list-style-type: none"> • Criar objetos • Herança • Polimorfismo 		
Estratégias	Atividades	Recursos	Avaliação	Gestão	
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registar o sumário e apresentar os objetivos da aula; • Correção da ficha de trabalho n04 • Apresentação dos objetivos da aula, resolução da ficha de trabalho N05; • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • Ficha de trabalho • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 5min • 40min • 5min • 8min 	

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	Conclusão e correção da ficha de trabalho N05.	Aula Nº	9	
	Ficha de trabalho N06 <ul style="list-style-type: none"> Trabalhar com Encapsulamento na programação orientada a objetos Proteger variáveis públicas Criar propriedade com campo de apoio 	Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> Explicar como o encapsulamento é usado para escrever código que só pode ser usado conforme pretendido pelo programador. Controlar o acesso aos dados dentro de uma classe aplicando encapsulamento com getters e setters Diferenciar entre variáveis públicas (propriedades), variáveis privadas (campos) e variáveis locais. Trabalhar a legibilidade e a depuração organizando corretamente as classes para incluir apenas o código de propósito singular. 		<ul style="list-style-type: none"> Encapsulamento Legibilidade e Depuração 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> Efetuar a chamada e registo de faltas; Registar o sumário e apresentar os objetivos da aula; Correção da ficha de trabalho N05 Apresentação dos objetivos da aula, resolução da ficha de trabalho N06; Objetivos próxima aula Avaliação da sessão 	<ul style="list-style-type: none"> Computador Videoprojector Plataforma escolar Ficha de trabalho GitHub Desktop Unity Engine Visual Studio 	<ul style="list-style-type: none"> Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> 2min 5min 5min 40min 5min 8min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento POO em ambiente gráfico

Sumário:	Continuação da ficha de trabalho N06 <ul style="list-style-type: none"> • Trabalhar com Encapsulamento na programação orientada a objetos • Proteger variáveis públicas • Criar propriedade com campo de apoio 	Aula Nº	10	
		Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> • Entender o conceito de classe como um novo tipo de dados que agrega dados e mecanismos de manipulação dos mesmos • Entender a estrutura de um programa como uma troca de mensagens entre objetos • Ler e Escrever dados de ficheiros 		<ul style="list-style-type: none"> • Conceção de Classes • Alojamento de Objetos 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registar o sumário e apresentar os objetivos da aula; • Continuação da resolução da ficha de trabalho N06; • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • Ficha de trabalho • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 40min • 5min • 5min • 8min

PLANIFICAÇÃO DE AULA

Unidade 2 – Desenvolvimento P00 em ambiente gráfico

Sumário:	Conclusão e correção da ficha de trabalho N06.	Aula Nº	11	
	Proposta de trabalho prático. Implementar os conhecimentos adquiridos nas aulas anteriores: <ul style="list-style-type: none"> • Criação de formulários (Menus) • Criação e utilização de base de dados. 	Duração	60 minutos	
Objetivos de Aprendizagem:		Conteúdos:		
<ul style="list-style-type: none"> • Entender o conceito de classe como um novo tipo de dados que agrega dados e mecanismos de manipulação dos mesmos • Entender a estrutura de um programa como uma troca de mensagens entre objetos • Ler e Escrever dados de ficheiros 		<ul style="list-style-type: none"> • Conceção de Classes • Alojamento de Objetos 		
Estratégias	Atividades	Recursos	Avaliação	Gestão
<ul style="list-style-type: none"> • Promover uma metodologia ativa, PBL – aprendizagem baseada em problemas; • Fomentar dinâmicas de grupo que promovam o diálogo, debate, troca de ideias e reflexão; • Incentivar à autonomia e capacidade de pesquisa; 	<ul style="list-style-type: none"> • Efetuar a chamada e registo de faltas; • Registar o sumário e apresentar os objetivos da aula; • Correção da ficha de trabalho N06 • Apresentação dos objetivos da aula, Trabalho prático; • Objetivos próxima aula • Avaliação da sessão 	<ul style="list-style-type: none"> • Computador • Videoprojector • Plataforma escolar • Ficha de trabalho • GitHub Desktop • Unity Engine • Visual Studio 	<ul style="list-style-type: none"> • Avaliação contínua, do tipo formativa, tendo em conta as questões e o diálogo professor-aluno e aluno-aluno existentes durante a aula. 	<ul style="list-style-type: none"> • 2min • 5min • 5min • 40min • 5min • 5min

Exemplo Ficha de trabalho

Objetivos:

- Trabalhar com Encapsulamento na programação orientada a objetos
- Proteger variáveis públicas
- Criar propriedade com campo de apoio

Recursos:

- Unity
 - Visual Studio
 - GitHub
 - Internet
 - Manual
-

Conteúdos/Exercícios:

Encapsulamento na Programação Orientada a Objetos

Assim como a abstração, o encapsulamento concentra-se fortemente em manter um nível de separação entre a complexidade subjacente do código e o outro código que o utiliza. Devido à sua semelhança, por vezes verá alguns seguidores de abstração e encapsulamento de grupos de práticas OOP num único pilar, geralmente sob o cabeçalho de encapsulamento. No entanto, estamos fazendo uma distinção que achamos importante: enquanto a abstração trata de resumir o código para torná-lo mais simples para outros programadores, o encapsulamento trata da proteção de valores e dados, como se estivessem dentro de uma cápsula, para que seja possível controlar o que os outros fazem e não têm acesso.

Objetivo

Neste exercício, vai trabalhar o quarto pilar da programação orientada a objetos: encapsulamento.

- Explicar como o encapsulamento é usado para escrever código que só pode ser usado conforme pretendido pelo programador.
- Controlar o acesso aos dados dentro de uma classe aplicando encapsulamento com getters e setters
- Diferenciar entre variáveis públicas (propriedades), variáveis privadas (campos) e variáveis locais.
- Trabalhar a legibilidade e a depuração organizando corretamente as classes para incluir apenas o código de propósito singular.

Agora que Já sabe como trabalhar com abstração, herança e polimorfismo:

- ✓ Criar um novo tipo de objeto (unit type);
- ✓ Trabalhar Herança;
- ✓ Trabalhar Polimorfismo.
- ✓ Trabalhar com notação de substituição (Override)
- ✓ Trabalhar com sobrecargas (Overloads)

Vai agora aplicar o conceito de encapsulamento.

Revisão do projeto

No resumo do projeto, verá que um dos itens a serem construídos. É uma unidade de produção. Esta unidade deve aumentar a produtividade de qualquer tipo de recurso que o utilizador tenha selecionado na cena. O utilizador selecionará o recurso da mesma forma que a empilhadora: clique com o botão esquerdo para selecionar a unidade e clique com o botão direito para selecionar o recurso para o mover. A produtividade de um recurso só deve ser aumentada enquanto a unidade de produtividade estiver a trabalhar ativamente nele e, se a unidade deixar o recurso, deve voltar a uma taxa de produção normal.

Recursos necessários:

- Encapsulamento na programação orientada a objetos**
- Proteger variáveis públicas
- Criar propriedade com campo de apoio**

Exercício:

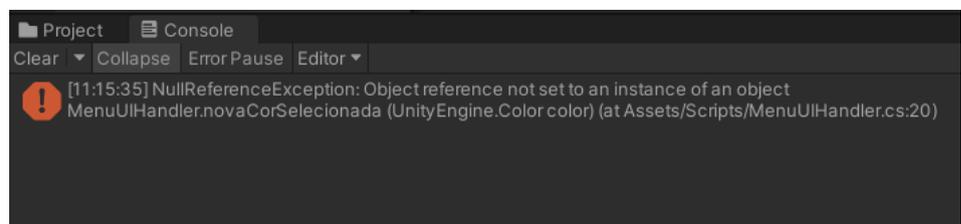
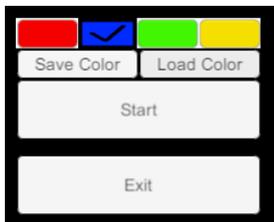
1. Na class GestorDoPrograma adicionou a seguinte variável pública:

```
//Declaração de membro da class como estática  
public static GestorDoPrograma Instancia;
```

Esta variável não pode ser privada porque causaria erros nos outros scripts que precisam de aceder-lhe, mas deixá-la pública significa que está vulnerável a um possível uso indevido.

- 1.1. Para efeitos experimentais, na classe MenuUIHandler adicione no final do método Start() uma linha de código que permita colocar a instancia de MainManager = null.

Teste a aplicação. Quando selecionar uma cor deve receber o seguinte erro:



Precisamos de uma maneira de evitar o uso indevido e ao mesmo tempo conceder acesso público à variável.

- 1.1. Crie um get para aceder à variável GestorDoPrograma

Assim que adiciona um get ou set a uma variável, essa variável transforma-se numa propriedade. Um tipo especial de variável que fornece acesso a dados internos através desses métodos (Get/Set) especializados.

Sem um set, a propriedade é estritamente de leitura o seu valor não pode ser definido em nenhum lugar

do programa. Agora, há um erro na linha de código maliciosa que adicionou anteriormente na class MenuUIHandler.cs. O que é bom, não queremos que nenhuma outra class possa redefini-la.

1.2. Comente a linha de código criado no ponto 1.1.

Repare na existência de um erro na class Gestor do programa. Isto significa que nem na class é possível modificar o valor da propriedade.

CS0200 Property or indexer 'GestorDoPrograma.Instancia' cannot be assigned to -- it is read only Assembly-CSharp MenuUIHandler.cs

1.3. Adicione um setter privado à propriedade.

Pode agora definir o valor da propriedade na class, mas apenas consegue obtê-lo quando fora da classe. Está encapsulado para aceitar apenas modificações da sua própria classe, protegido contra uso indevido e corrupção do mundo exterior!

2. Na class ResourcePile.cs, crie um campo de apoio para a variável `productionSpeed`.

Observações:

- Duração: 15 minutos
- Avaliação: Será tido em conta na avaliação do trabalho, quer o conteúdo, quer a eficiência.
- Entrega do Trabalho:
 - ✓ **Se** terminar dentro da duração solicitada **Commit:** `06FT_POOAG`
 - ✓ **Se não** terminar na duração solicitada, **Commit:** `06FT_POOAG_INPROGRESS`

*Nota: Não esquecer de fazer **Push***

Projeto Final

Objetivos:

- Criação de Base de Dados
- Desenvolvimentos de Interfaces
- Criação de classes segundo conceitos estudados.
- Criação de Base de Dados

Recursos:

- Unity
 - Visual Studio
 - GitHub
 - Internet
 - Manual
 - SQLite
-

Conteúdos/Exercícios:

Trabalho prático.

Agora que já sabe utilizar os conceitos de POO Programação Orientada a Objetos, utilizar persistência de dados entre cenas e sessões e tornar o código mais eficiente e seguro. Vamos melhorar a aplicação para que permita a criação de conta, login, armazenamento e tratamento de dados do utilizador.

Objetivo

Neste projeto vamos criar e gerir dados com recurso a uma base de dados SQLite.

- Criar uma Base de Dados
- Registo de dados do utilizador
- Editar dados do utilizador
- Carregar os dados do utilizador quando inicia a sessão
- Guardar os dados do utilizador quando termina a sessão
- Disponibilizar lista de resultados

Na ficha de trabalho anterior trabalhámos o encapsulamento:

- ✓ Explicar como o encapsulamento é usado para escrever código que só pode ser usado conforme pretendido pelo programador.
- ✓ Controlar o acesso aos dados dentro de uma classe aplicando encapsulamento com getters e setters
- ✓ Diferenciar entre variáveis públicas (propriedades), variáveis privadas (campos) e variáveis locais.
- ✓ Trabalhar a legibilidade e a depuração organizando corretamente as classes para incluir apenas o código de propósito singular.

Vai agora aplicar todos os conceitos estudados.

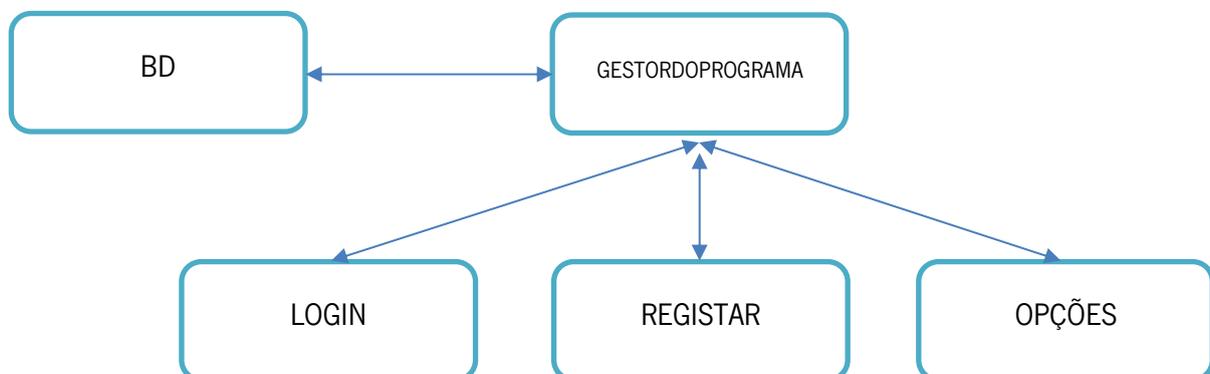
Revisão do projeto

O projeto até este momento permite a produção, transporte e entrega de produtos. É possível aumentar a produção sempre que a nova unidade se aproxima de uma pilha de recursos e transportar os produtos até ao ponto de entrega. A cor é persistente entre cenas e é possível guardas os dados da cor num ficheiro.

Recursos necessários:

- Base de Dados SQLite com no mínimo 3 tabelas.**
- Class para gestão de operações na Base de Dados, como INSERT, CREATE, UPDATE, SELECT.
- Class para gestão de dados relacionados com o utilizador e sessão.**
- Formulários para Registrar, Login, Editar utilizador e apresentar resultados. Outros que considerar necessários.**
- Validação de dados.**

Diagrama (exemplo):



Login, Registrar e Opções: são classes associadas a cenas. Permitem receber dados dos objetos da cena e enviar e receber dados da class GestorDoPrograma.

GestorDoPrograma: implementa persistência de dados, serve para armazenar dados do jogo e utilizador.

BD: Executa o tratamento de dados

Parâmetros de avaliação

<i>Validação de entradas</i>	
<i>Criação de classes</i>	
<i>Abstração</i>	
<i>Herança</i>	
<i>Polimorfismo</i>	
<i>Encapsulamento</i>	
<i>Qualidade Global</i>	

Observações:

- Duração: 3 horas letivas + trabalho de casa
- Avaliação: Sumativa.
- Entrega do Trabalho:

✓ **Commit:** *07PROJETO_POOAG*

*Nota: Não esquecer de fazer **Push***

Anexos

Anexo 1 – Autorização publicação do nome da instituição.

**COLÉGIO
DE GAIA**

Só com a luz do saber
se alcança a vitória.

DECLARAÇÃO

Serve a presente declaração para autorizar o aluno de Mestrado de Ensino de Informática, Rui Miguel Loureiro Baptista, portador do cartão de cidadão nº11600817 2 ZX1 com data de validade 12-02-2030, a publicar o nome: "Colégio de Gaia – Escola Católica", no seu relatório final para conclusão do mesmo.***

Por ser verdade e ter sido pedida, se passa a presente declaração que vai assinada e autenticada de acordo com a lei.***

Vila Nova de Gaia em 27 de Julho de 2023

Pe António Manuel Barbosa Ferreira
Diretor do Colégio de Gaia - Escola Católica

COLÉGIO DE GAIA
Escola Católica
Rua de Pádua Correia, 166
4400-238 Vila Nova de Gaia
NIPC: 501 066 144

Colégio de Gaia / NIF: 502060144

Rua de Pádua Correia, 166
4400-238 Vila Nova de Gaia
Portugal

Tel 223 754 007
Fax 223 705 477

geral@colgaia.pt
www.colgaia.pt

REPÚBLICA
PORTUGUESA

Colaborado por

PPCM

2020

EUROPEAN
COMMISSION

98

Anexo 2 – Manual Unity



**COLÉGIO
DE GAIA**

Objetivo:

Serve o presente manual como guia de apoio na aprendizagem dos alunos.

Ferramentas:

Unity

Autor:

Rui Baptista

Contribuidores:

Conteúdo

<i>Licença concedida aos utilizadores deste trabalho</i>	2
Questão de investigação	2
Objetivos	2
Organização do relatório	2
1.1. Construtivismo na Educação: Aprendizagem Significativa e Participativa	4
1.2. Jogos e Aprendizagem: Integração do Construcionismo na Educação.....	5
1.2. O Pensamento Abstrato - Abstract Data Type (ADT)	6
1.3.1. Programação Orientada a Objetos	7
1.4. Ludificação	7
2.1. O Meio.....	8
2.2. A Escola.....	9
2.3. Oferta Formativa	10
2.4. Caracterização da Turma	11
3.1. O Curso	12
3.2. A Disciplina.....	13
3.2.1. Objetivos de Aprendizagem	14
3.2.2. Conteúdos Programáticos	15
3.3. Elenco temático da intervenção	16
3.4. Avaliação Curricular	16
4.1. Ferramentas Utilizadas.....	18
4.2. Planificação	22
4.3. Restrições.....	22

4.4. Aulas Assistidas	23
4.5. Intervenção Pedagógica.....	29
4.5.1. Resumo do projeto	35
4.5.2. Atividades realizadas no âmbito da intervenção	41
4.5.2.1. Conversa com Especialistas	41
4.5.2.2. Envelope secreto.....	45
4.5.3. Projeto Final.....	51
4.6. Ludificação com estratégia motivacional.....	55
4.6.1. Quadro de progresso do projeto.	56
4.6.2. Certificado de participação na Intervenção Pedagógica	63
4.7. Avaliação da intervenção	65
4.7. Inovação.....	68
4.9. Outras atividades e participação	69
5.1. Conclusão.....	70
5.2. Limitações	71
5.2. Recomendações.....	72
1. Referências	73
Resumo do projeto	75
Planificação de aulas.....	80
Exemplo Ficha de trabalho	92
Projeto Final.....	95
Anexo 1 – Autorização publicação do nome da instituição.	98
Anexo 2 – Manual Unity	99

Índice de Figuras

FIGURA 1 - ESTRATÉGIA PEDAGÓGICA PROPOSTA - CRIAÇÃO DE JOGOS EDUCATIVOS POR ALUNOS (FONTE: MARTINS & OLIVEIRA, 2018, P. 4)	6
FIGURA 2 - MAPA VILA NOVA DE GAIA (FONTE: GOOGLE MAPS).....	9
FIGURA 3 - O CURSO: PLANO DE ESTUDOS (FONTE: COLGAIA.PT)	12
FIGURA 4 - FERRAMENTAS: UNITY (FONTE: UNITY3D.COM)	19
FIGURA 5 - FERRAMENTAS: MICROSOFT VISUAL STUDIO (FONTE: VISUALSTUDIO.MICROSOFT.COM)	19
FIGURA 6 - FERRAMENTAS: MICROSOFT TEAMS (FONTE: TEAMS.MICROSOFT.COM)	20
FIGURA 7 - FERRAMENTAS: MICROSOFT PLANNER (FONTE: TASKS.OFFICE.COM).....	20
FIGURA 8 - FERRAMENTAS: GITHUB (FONTE: GITHUB.COM)	20
FIGURA 9 - FERRAMENTAS: DB BROWSER (FONTE: GITHUB.COM/SQLITEBROWSER)	20
FIGURA 10 - FERRAMENTAS: SQLITE (FONTE: SQLITE.ORG)	21
FIGURA 11 - LINGUAGENS: C SHARP (FONTE: SQLITE.ORG).....	21
FIGURA 12 - LINGUAGENS: C SHARP (FONTE: STONEBRANCH.COM)	21
FIGURA 13 - AVALIAÇÃO PRÉVIA: GOSTAS DE JOGAR?.....	26
FIGURA 14 - AVALIAÇÃO PRÉVIA: QUANTAS HORAS POR DIA JOGAS, EM MÉDIA?.....	26
FIGURA 15 - AVALIAÇÃO PRÉVIA: OPORTUNIDADE DESENVOLVIMENTO JOGOS NO COLÉGIO	27
FIGURA 16 - AVALIAÇÃO PRÉVIA: MOTIVAÇÃO CURSO ITM.....	28
FIGURA 17 - AVIAÇÃO PRÉVIA: IMPORTÂNCIA DOS JOGOS NA APRENDIZAGEM DE CONCEITOS DE POO	28
FIGURA 18 - UNITY: CONCESSÃO DE LICENÇA EDUCACIONAL GRATUITA.	30
FIGURA 19 - UNITY: ABORDAGENS E METODOLOGIAS ATIVAS	30
FIGURA 20 - IMPLEMENTAÇÃO AGILE SCRUM: TAREFA	31
FIGURA 21 - IMPLEMENTAÇÃO AGILE SCRUM: QUADRO	32
FIGURA 22 - GESTÃO DOCUMENTAL DO PROJETO	33
FIGURA 23 - MANUAL DO PROJETO	33
FIGURA 24 - VERSIONAMENTO DE SOFTWARE	34
FIGURA 25 - VÍDEOS DEMONSTRATIVOS	35
FIGURA 26 - PROJETO: VISTA GERAL DA CENA.	36
FIGURA 27 - PROJETO: INTERAÇÃO DO UTILIZADOR.	38
FIGURA 28 - PROJETO: COR SELECIONADA	38
FIGURA 29 - PROJETO: UNIDADES E RECURSOS.....	39
FIGURA 30 - CONVERSA COM ESPECIALISTAS: VIDEOCONFERÊNCIA	42
FIGURA 31 - CONVERSA COM ESPECIALISTAS: MOTIVAÇÃO	42
FIGURA 32 - CONVERSA COM ESPECIALISTA: AVALIAÇÃO SESSÃO.....	43

FIGURA 33 - CONVERSA COM ESPECIALISTAS: INTERESSE DESPERTADO	43
FIGURA 34 - CONVERSA COM ESPECIALISTAS: INTERESSE NA UTILIZAÇÃO PESSOAL.....	44
FIGURA 35 - CONVERSA COM ESPECIALISTAS: INTERESSE NA UTILIZAÇÃO PROFISSIONAL	44
FIGURA 36 - ENVELOPE SECRETO: MENSAGEM SECRETA TEAM LEADER	45
FIGURA 37 - ENVELOPE SECRETO: TEAM LEADER	46
FIGURA 38 - ENVELOPE SECRETO: FORMAÇÃO DOS GRUPOS DE TRABALHO	47
FIGURA 39 - ESTATÍSTICA ENVELOPE SECRETO: GRAU DE MOTIVAÇÃO.....	47
FIGURA 40 - ESTATÍSTICA ENVELOPE SECRETO: AVALIAÇÃO DA AULA.....	48
FIGURA 41 - ESTATÍSTICA ENVELOPE SECRETO: SESSÃO NO ESCLARECIMENTO DE DÚVIDAS.	48
FIGURA 42 - ESTATÍSTICA ENVELOPE SECRETO: COMPREENSÃO DOS PILARES ANTES DA SESSÃO.	49
FIGURA 43 - ESTATÍSTICA ENVELOPE SECRETO: AVALIAÇÃO DEPOIS DA SESSÃO	49
FIGURA 44 - ESTATÍSTICA ENVELOPE SECRETO: FOI POSSÍVEL AL APLICAÇÃO DOS CONCEITOS NA PRÁTICA.	50
FIGURA 45 - ESTATÍSTICA ENVELOPE SECRETO: APLICAÇÃO DOS CONCEITOS NA PRÁTICA.	50
FIGURA 46 - PROJETO FINAL: DIAGRAMA DO PROJETO.	51
FIGURA 47 - PROJETO FINAL: MENU LOGIN.....	52
FIGURA 48 – PROJETO FINAL: MENU DE REGISTO	53
FIGURA 49 - PROJETO FINAL: LISTA DE RESULTADOS	54
FIGURA 50 - PROJETO FINAL: CLASSE BD	55
FIGURA 51 - LUDIFICAÇÃO: CONQUISTADAS OBTIDAS POR UM ALUNO DA TURMA.	59
FIGURA 52 - LUDIFICAÇÃO: QUADRO DE PROGRESSO.....	60
FIGURA 53 - LUDIFICAÇÃO: ANÁLISE E ESTATÍSTICA DE RESULTADOS OBTIDOS.	61
FIGURA 54 - LUDIFICAÇÃO: CERTIFICADO DE PARTICIPAÇÃO NA INTERVENÇÃO PEDAGÓGICA	64
FIGURA 55 – LUDIFICAÇÃO: ENTREGA DE CERTIFICADOS AOS ALUNOS.....	64
FIGURA 56 - AVALIAÇÃO DA INTERVENÇÃO: OBJETIVOS.....	65
FIGURA 57 - AVALIAÇÃO DA INTERVENÇÃO: COMPREENSÃO DOS CONCEITOS	65
FIGURA 58 - AVALIAÇÃO DA INTERVENÇÃO: APLICAÇÃO DOS CONCEITOS APRENDIDOS.	66
FIGURA 59 - AVALIAÇÃO DA INTERVENÇÃO: ORGANIZAÇÃO, CLAREZA, RECURSOS E APOIO.	67
FIGURA 60 - PERSISTÊNCIA DE DADOS ENTRE CENAS.....	105
FIGURA 61 - UNITY PROCESSO DE SERIALIZAÇÃO E DESSERIALIZAÇÃO DE DADOS	108

Tipos de dados

enum

Um tipo enumeração (ou tipo enum) é um tipo de valor definido por um conjunto de constantes nomeadas do tipo numérico integral subjacente. Para definir um tipo de enumeração, use a palavra-chave enum e especifique os nomes dos membros enum:

Tipo	Exemplo / Representação
enum	<code>enum coelho {velocidade, estado};</code>

Exemplo:

```
public enum Coelho
{
    Lento,
    Rápido
}

static void Main(string[] args)
{
    Coelho a = Coelho.Lento;
    Console.WriteLine($"Valor integral de {a} é {(int)a}");// output: 0

    var b = (Coelho)1;
    Console.WriteLine(b); // output: rápido

    var c = (Coelho)4;
    Console.WriteLine(c); // output: 4

    Console.ReadKey();
}
```

namespace

namespaces são usados para organizar de forma lógica classes, Estruturas, interfaces, enums e delegar. Os namespaces em C# podem ser aninhados. Significa que um namespace pode conter outros namespaces.

Exemplo de namespace:

```
using UnityEngine;

Exemplo de namespace aninhado
using UnityEngine.SceneManagement;
```

Persistência de Dados

Persistência de dados permite preservar informação entre cenas. Persistência de dados significa fazer com que os dados durem mais do que o processo usado para criá-los.

Exemplos:

- O ícone específico de cada jogador, que seleciona num jogo multijogador (Ex: um jogo de perguntas e respostas)
- O nome inserido no início de uma aplicação e apresentado durante toda a sessão (ex: numa ferramenta de pesquisa)

- O progresso de um jogo em andamento (por exemplo, um jogo de RPG)
- O documento que está a criar numa aplicação de processamento de texto.

Persistência de dados entre cenas

No Unity, os dados criados numa cena estão sempre disponíveis nessa cena. Mas quando um utilizador chama outra cena, o que acontece? Normalmente, os dados são perdidos. A persistência de dados entre cenas é o processo de transferência de dados de cena para cena para dar ao utilizador uma experiência consistente à medida que avança na aplicação. Os dois primeiros exemplos acima, exige persistência de dados entre cenas.

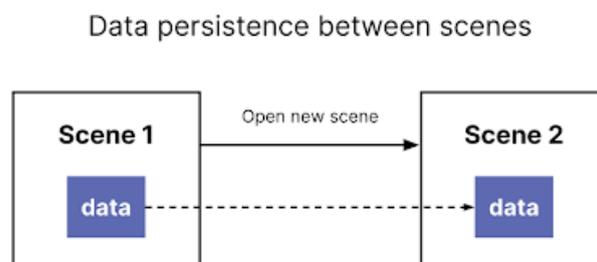
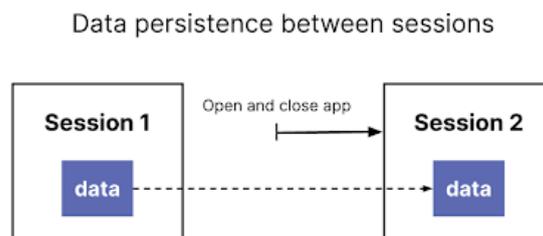


Figura 60 - persistência de dados entre cenas

Persistência de dados entre sessões

Os últimos dois exemplos (jogos em andamento e aplicação de processamento de texto) são tipicamente experiências de várias sessões. Os utilizadores desejam guardar o progresso feito durante uma sessão e restaurá-la para continuar onde pararam. Estes são exemplos de persistência de dados entre sessões.



Encontrará frequentemente os dois tipos de persistência de dados em aplicações mais complexas — dados que precisam “seguir” o utilizador conforme este se move entre cenas e dados que precisam de ser guardados e restaurados em várias sessões.

Exemplo:

```

public class GerenciadorDoPrograma : MonoBehaviour
{
    //Este código permite o acesso ao GerenciadorDoPrograma de qualquer outro script

    //Declaração de membro da classe como estática
    public static GerenciadorDoPrograma Instancia;

    //Awake(Despertador) é chamado quando a instancia do script for carregada
    📧 Mensagem do Unity | 0 referências
    private void Awake()
    {
        //Informa o programa que é desta instância que estamos a tratar
        Instancia = this;
        //Método que informa a aplicação para não destruir o objeto marcado quando é carregada uma cena.
        DontDestroyOnLoad(gameObject);
    }
}

```

Declaração `public static GerenciadorDoPrograma Instancia;`

Esta é a declaração de um membro de classe estática. Observe a palavra-chave `static`, depois da palavra-chave `public`. Esta palavra-chave significa que os valores armazenados neste membro de classe serão partilhados com todas as instâncias da classe.

Exemplo: se existirem cinco instâncias de `GerenciadorDoPrograma` na cena, todas elas partilham o valor armazenado em `Instancia`. Se qualquer uma das 5 instâncias de `GerenciadorDoPrograma` alterar o valor, também será alterado o valor das restantes.

Método `Awake()`

O método `Awake()` é chamado quando a instância da class for chamada. Neste exemplo o seguinte código é executado assim que o método é chamado:

```

Instancia = this;
DontDestroyOnLoad(gameObject);

```

A primeira linha de código armazena “`this`” (Instância) no membro da classe `Instancia` – a instância atual de `MainManager`. A instância pode agora ser chamada da seguinte forma: `GerenciadorDoPrograma.Instancia` de qualquer outro script (ex o script `Unit`) e obter um link para a instância específica sem a necessidade de criar uma referência como se faz quando atribui `GameObjects` a propriedades de script no Inspector.

A segunda linha de código `DontDestroyOnLoad(gameObject);` informa que o `GameObject` anexado a este script não deve ser destruído quando mudar de cena.

Destruir `GameObjects`

Nem sempre é necessária a existência de Objetos de Jogo duplicados. Um jogo pode ter múltiplos

inimigos do mesmo tipo o que exige a criação de várias instâncias do mesmo GameObject. Já para criar persistência de dados essa necessidade não se verifica, basta uma instância ser criada para obter o resultado pretendido. Em ambos os casos a criação e destruição necessita de ser controlada.

Singleton

Singleton é um padrão utilizado para garantir que apenas uma instância de um determinado GameObject existe durante a execução de um programa.

Exemplo:

```
//Se instância não for nula então
if (Instancia != null)
{
    Destroy(gameObject);
    return;
}
```

Armazenar e passar dados selecionados

Passar dados entre cenas é agora mais fácil com a implementação de persistência:

Se quiser alterar a cor de um objeto noutra cena é agora possível com recurso a Instância. Basta para o efeito declarar uma variável do tipo Cor na Classe GestorDoPrograma e passar a cor que o utilizador selecionar:

```
//Variável do tipo cor
public Color corMontacargas;
```

Para armazenar a cor selecionada a atribuição é realizada da seguinte forma:

```
// adicione código aqui para lidar com a cor selecionada
GerenciadorDoPrograma.Instancia.corMontacargas = color;
```

A cor é atribuída ao objeto sempre que ele é chamado pela primeira vez (Método Start())

```
private void Start()
{
    //Atribuição da cor selecionada às unidades montacargas
    if (GerenciadorDoPrograma.Instancia != null)
    {
        SetColor(GerenciadorDoPrograma.Instancia.corMontacargas);
    }
}
```

Persistência de dados entre sessões

Para que os dados persistam entre as sessões, precisam de ser armazenados de alguma forma. É necessário converter os dados selecionados pelo utilizador num formato que possa ser armazenado e lido quando carregar novamente a aplicação.

O processo de conversão de dados complexos num formato no qual possam ser armazenados é

chamado de **serialização (serialization)**. Quando acede os dados novamente, o processo de convertê-los de volta é chamado de **desserialização (deserialization)**.

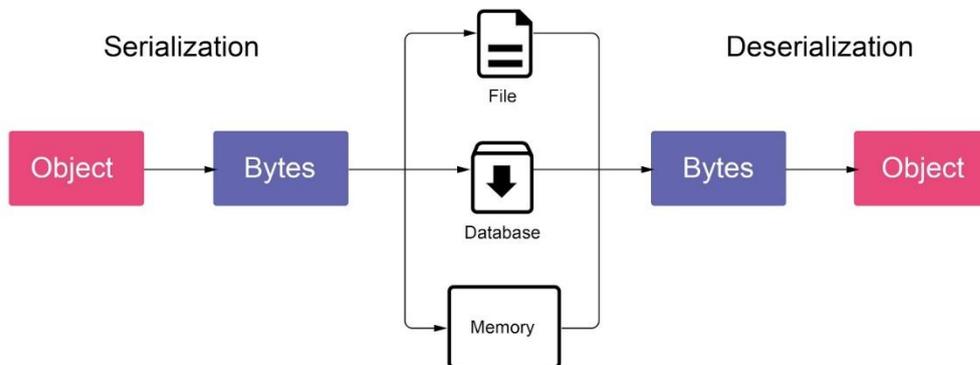


Figura 61 - Unity processo de serialização e desserialização de dados

Json no processo de armazenamento de dados

JSON é um formato de texto usado para armazenar dados e trocá-los entre plataformas. Foi desenvolvido pela primeira vez para a web e o seu nome completo é **JavaScript Object Notation (JSON)**. É baseado na linguagem JavaScript, mas é independente da linguagem — pode usá-lo se estiver a programar em C# ou em qualquer outra linguagem de programação.

O formato JSON armazena dados na forma de um par **chave:valor**. A chave é uma string e o valor pode ser:

- um número
- uma linha
- um booleano (verdadeiro/falso)
- uma matriz de dados
- Outro objeto JSON

A seguinte string JSON codifica um objeto que armazena informações básicas sobre uma pessoa:

```
{  
  "nome": "Rui",  
  "idade": 23,  
  "morada": {  
    "rua": "21 da dezembro",  
    "cidade": "Porto"  
  }  
  "animal": ["cão", "gato"]  
}
```

- Cada **objeto** está compreendido entre chavetas (`{ }`).
- Cada **entrada** é representada na forma de um par chave:valor, separado por uma vírgula.
- A entrada “animal” representa um array de strings — os valores do array são listados entre parêntesis retos (`[]`).
- O valor associado à entrada “morada” também é representado entre chavetas — esta é a forma de representar um **novo objeto JSON**.

JSON foi desenvolvido para troca de dado entre sistemas muito diferentes, o que o torna uma boa opção para guardar dados nas aplicações a desenvolver. O Unity tem uma classe auxiliar chamada **JsonUtility**, que permite aceder a uma classe **serializável** e transformá-la numa representação JSON. Vamos ver o funcionamento para **serializar** e **desserializar** dados.

Serializar dados

Considere a seguinte classe **Jogador**:

```
[Serializable]
public class Jogador
{
    public int nivel;
    public Vector3 posicao;
    public string nomeJogador;
}
```

NOTA: o atributo `[Serializable]` pertence ao namespace `System` (`using System;`) é necessário para transformar dados em Json (`JsonUtility`)

Imagine que quer passar os dados da classe para **JsonUtility** com os seguintes valores:

```
PlayerData dadosJogador = new PlayerData();
dadosJogador.nivel = 1;
dadosJogador.posicao = new Vector3(3.0f, 4.4f, 2.3f);
dadosJogador.nomeJogador = "Pedro";
```

faz a seguinte chamada: `string json = JsonUtility.ToJson(dadosJogador)`; a chamada resulta na seguinte string JSON:

```
{
  "nivel": 1,
  "posicao": {
    "x" : 3.0,
    "y" : 4.4,
    "z" : 2.3 },
  "nomeJogador": "Pedro"
}
```

Observe que, como **posicao** é um **vetor3**, é codificado como um **novo objeto JSON** com três chaves: **x**, **y** e **z**. Isto ocorre porque a classe **Vector3** (extremamente simplificada) é representada da seguinte forma:

```
[Serializable]
public class Vector3
{
    public float x;
    public float y;
    public float z;
}
```

Desserialização de dados:

JsonUtility implementa um método que faz o inverso do processo que serialização: **FromJson<T>**.

Este método lê uma string de dados JSON e cria uma instância do objeto com os dados preenchidos.

Utilização do modelo — como deve ter reparado no exemplo **string JSON dadosJogador**, o tipo original de dados não é armazenado no arquivo JSON. O Unity precisa do argumento do modelo para ler o valor correto no campo correto.

Para chamar os dados do jogador escrevemos:

```
Jogador dadosJogador = JsonUtility.FromJson<Jogador>(json);
```

Desta forma preenche os **dadosJogador** com os valores da string JSON.

Limitações do JsonUtility

Existem algumas limitações na classe **JsonUtility** do Unity, que foi projetada para desempenho e simplicidade. **JsonUtility** não funciona em tipos primitivos, arrays, listas ou dicionários.

Escrever e Ler dados .json

Escrever dados

Para escrever dados para um ficheiro .Json é necessário a utilização da seguinte expressão:

```
File.WriteAllText(Application.persistentDataPath + "/guardardados.json", json);
```

NOTA: a Class **File** pertence ao namespace **System.IO** (**using System.IO;**) é necessária para trabalhar com ficheiros.

Método **WriteAllText**: Recebe dois parâmetros, (caminho + tipo de dados) do ficheiro **Application.persistentDataPath** (caminho por defeito), o nome do ficheiro **"/guardardados.json"** e o tipo de dados é **json**.

NOTA: lista do caminho do ficheiro de dados por defeito para cada plataforma: [lista de caminhos por plataforma](#).

Exemplo Escrever Dados:

```

public void Escrever()
{
    //Criar uma nova instância da class Jogador
    Jogador dados = new Jogador();
    //Atribuição do valor a armazenar ao membro da class Jogador
    dados.nivel = nivel;
    //transformar a instância em JSON
    string json = JsonUtility.ToJson(dados);
    //escrever dados para um ficheiro
    File.WriteAllText(Application.persistentDataPath + "/guardardados.json", json);
}

```

Ler dados

Para ler os dados de um ficheiro .Json é necessária a seguinte expressão:

```
string path = Application.persistentDataPath + "/guardardados.json";
```

Para verificar se o ficheiro existe utilizamos o método: `File.Exists(path)` se o ficheiro existir o método lê o ficheiro com o método: `File.ReadAllText(path)`.

Exemplo ler dados:

```

public void Ler()
{
    //Armazenar caminho para o ficheiro:
    string path = Application.persistentDataPath + "/guardardados.json";
    //verificar se o ficheiro existe
    if (File.Exists(path))
    {
        //Ler ficheiro
        string json = File.ReadAllText(path);
        //Armazenar os dados na instancia
        Jogador dados = JsonUtility.FromJson<Jogador>(json);
        //armazenar os dados no membro da class nivel
        nivel = dados.nivel;
    }
}

```

Carregar e Guardar dados na aplicação

Para carregar nível se este existir, no projeto em desenvolvimento, procure o método `Awake()` na class `GestorDoPrograma` e adicione o seguinte código:

```
//Ler nível guardado
Ler();
```

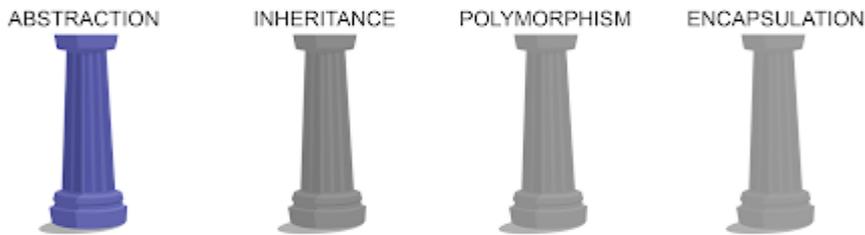
Para pré-selecionar o nível guardado, no método `Start()` da class `MenuUIHandler.cs`, adicione o seguinte código:

```
//Pré-selecionar nível guardado entre sessões
Utilizador.Nivel = GestorDoPrograma.Instancia.nivel;
```

Para Guardar o nível selecionada quando o utilizador sair da aplicação, no inicio do método `Sair()` da class `MenuUIHandler.cs`, adicione o seguinte código:

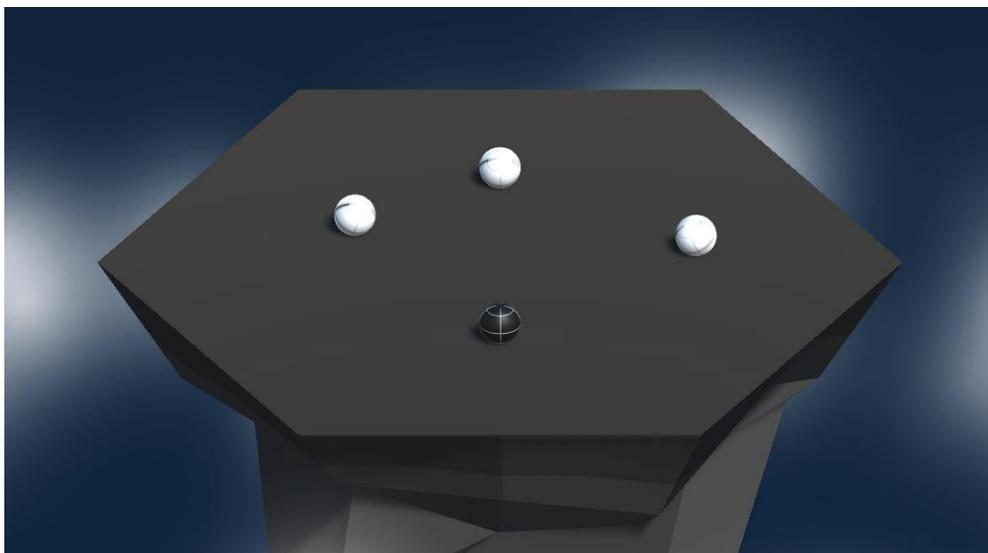
```
//Guardar o último nível
GestorDoPrograma.Instancia.Escrever();
```

Abstração



O primeiro pilar da POO é a **Abstração**. Abstração permite manter o código limpo e simples de interpretar para o programador que o usa. Sejas tu ou outra pessoa. Abstração é o processo de remoção de código complexo dos scripts onde outros programadores o verão, e apenas expor a funcionalidade que outros programadores realmente precisam. Ao “abstrair” os detalhes, reduz o código duplicado e fornece acesso fácil às funções mais úteis. Na verdade, já está bastante familiarizado com este pilar, porque já o utilizou com alguma frequência na unidade anterior. Sempre que chama um método para executar uma tarefa em vez de escrever o código todo novamente, está a utilizar a abstração!

Vamos imaginar que pretendemos criar inimigos num jogo. Para o Efeito foi criado um inimigo padrão (Prefab), que deve ser invocado assim que o jogo começa. Neste exemplo o inimigo padrão tem o nome **inimigoPrefab**.



Para instanciar inimigos podemos escrever o seguinte código no método **Start()**:

```
void Start()
{
    for (int i = 0; i < 3; i++)
    {
        Instantiate(inimigoPrefab, PosicaoCriacaoPadrao(),
            inimigoPrefab.transform.rotation);
    }
}
```

Como o método **Start()** é executado apenas uma vez quando o jogo é iniciado, se quisesse gerar outra onda de inimigos, teria de colar as instruções noutra lugar do script para que isso fosse possível ocorrer. Ter de reescrever constantemente (ou copiar e colar) as mesmas linhas de código repetidamente é ineficiente e propenso a erros. Também significa que a reestruturação de scripts, posteriormente, torna-se difícil pois precisa pesquisar em todos os seus scripts e reescrever todas as referências novamente.

Para evitar esse trabalho pode escrever um método para lidar com a criação de inimigos:

```
void CriarOndaDeInimigos() // Criar um novo método de nível superior
{
    for (int i = 0; i < 3; i++)
    {
        Instantiate(inimigoPrefab, PosicaoCriacaoPadrao(),
            inimigoPrefab.transform.rotation);
    }
}

void Start() // Chamar método de nível superior
{
    CriarOndaDeInimigos();
}
```

A experiência com a abstração não se limita a métodos criados por si. Na verdade, tem beneficiado da abstração quando invoca métodos incorporados no .Net Framework e mais recentemente no Unity.

```
rb.AddForce(Vector3.up * Random.Range(12, 16), ForceMode.Impulse);
```

Adicionar força a um Rigidbody envolve muita matemática de interação com o sistema físico. No entanto, como o Unity forneceu essa função de forma simples “abstrata” de alto nível (AddForce), precisa apenas chamar o método AddForce e fornecer os parâmetros relevantes às suas necessidades. Nem precisa entender a física do Unity para adicionar força com sucesso ao objeto!

Abstração e reestruturação de scripts

Lembre-se de que uma parte importante da reestruturação de código é melhorar a sua funcionalidade sem alterar a maneira como outros programadores interagem com ela. A abstração desempenha um papel fundamental nisso! Contanto que a chamada e a saída do método não vão mudar, pode ajustar o conteúdo do método sem que o outro programador saiba. Isso significa que pode reestruturas o seu código com segurança, sem o risco de arranjar problemas no projeto.

Exemplo:

```
//Método que devolve o triplo do resultado obtido
1 referência
int ResultadoTriplo(int inputNumero)
{
    int outputNumero = inputNumero + inputNumero + inputNumero;
    return outputNumero;
}
```

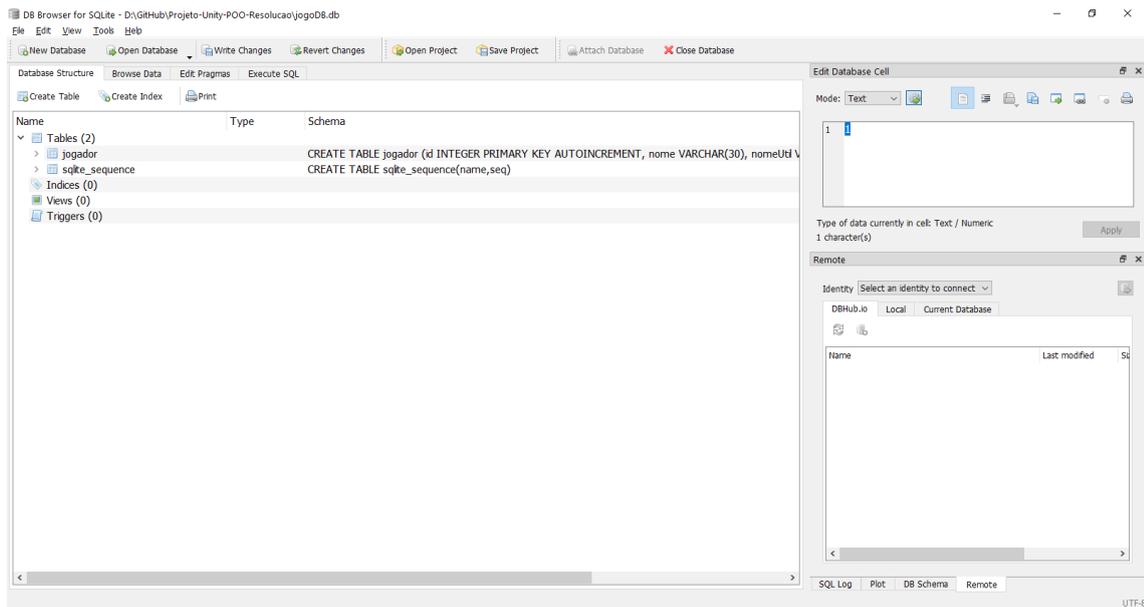
Na função acima, o script de chamada passa um inteiro e obtém uma saída que é o triplo do valor recebido. A abordagem para obter o número de saída é um pouco ineficaz, então, pode reestruturar o código para que fique mais eficiente:

```
//Método que devolve o triplo do resultado obtido -> Reestruturado
1 referência
int ResultadoTriplo(int inputNumero)
{
    int outputNumero = inputNumero * 3;
    return outputNumero;
}
```

Mesmo que o conteúdo do método tenha sido reescrito, a forma como o método é chamado permanece exatamente a mesma, assim como a forma como é devolvido. Portanto, qualquer chamada da função em qualquer parte do projeto continuará funcional.

DB Browser (SQLite)

DB Browser é um interface gráfico que permite o acesso à base de dados. Repleto de funcionalidades, permite de forma intuitiva a criação e edição de bases de dados tabelas e dados. Pode fazer o download [aqui](#):



System.Globalization

Contém classes que definem informações relacionadas à cultura, incluindo idioma, país/região, calendários em uso, padrões de formato para datas, moeda e números e ordem de classificação para strings.

```
using System.Globalization;
```

Converter string para float

Para converter dados armazenados como texto num sistema de dados de um país/região sem correr o risco de obter erros de caracteres utilizamos o namespace `System.Globalization`. Exemplo de conversão de texto para float:

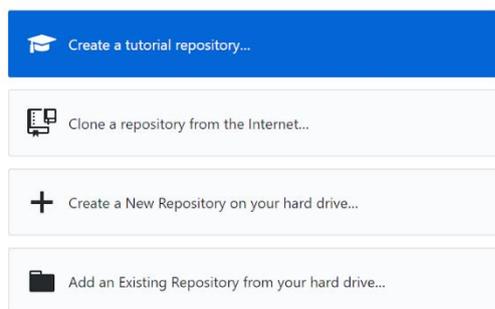
Anexo 3 – Manual Git

Criar Repo GitHub

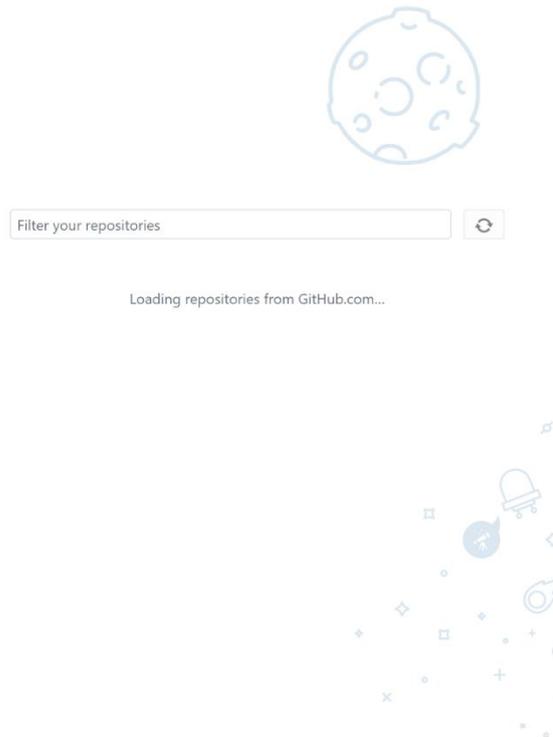
1. Se ainda não tiver uma conta, crie uma conta GitHub gratuita em github.com.
2. Faça download e instale o [GitHub Desktop](#).
3. Se ainda não tiver um repositório associado à sua conta deve visualizar o seguinte menu de boas-vindas:

Let's get started!

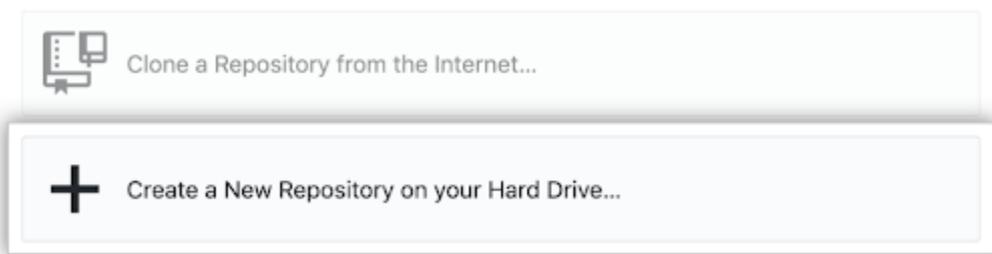
Add a repository to GitHub Desktop to start collaborating



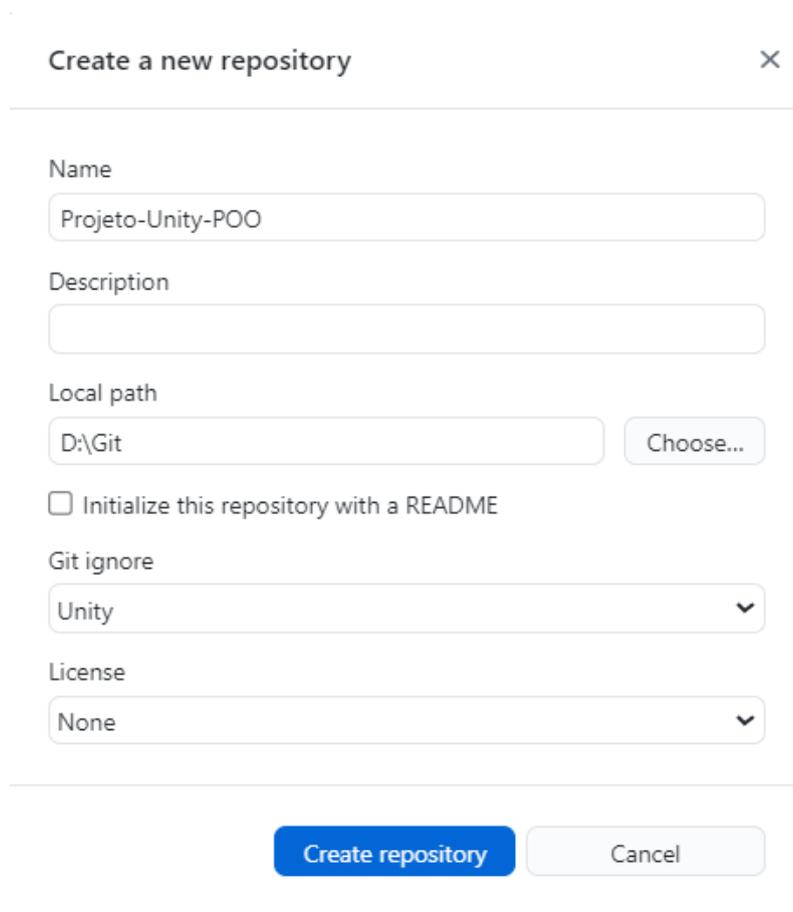
ProTip! You can drag & drop an existing repository folder here to add it to Desktop



4. Crie um repositório no computador:



Para criar um repositório clique em **File > New Repository**



Create a new repository

Name
Projeto-Unity-POO

Description

Local path
D:\Git Choose...

Initialize this repository with a README

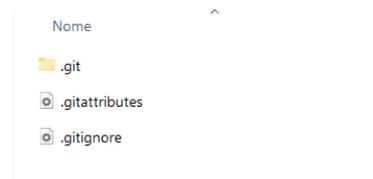
Git ignore
Unity

License
None

Create repository Cancel

5. Clique em Create repository
6. Abra a pasta que acabou de ser criada no seu computador no local que especificou. Se o seu computador estiver configurado para visualizar arquivos ocultos, verá três itens:

- a. .git folder
- b. .gitattributes
- c. .gitignore



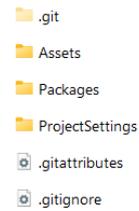
7. A pasta .git informa ao GitHub que este é, na verdade, um repositório do GitHub. Os atributos .gitignore e .gitattributes especificam a configuração do seu repositório. Não é necessário, mas se quiser ver esses arquivos ocultos, pode pesquisar no Google como fazer para Windows ou Mac.

Para dar ao GitHub algo para monitorizar, colocará um projeto real do Unity nesta pasta na próxima etapa.

8. Faça download do projeto aqui:
9. Adicione o conteúdo da pasta ao repositório criado

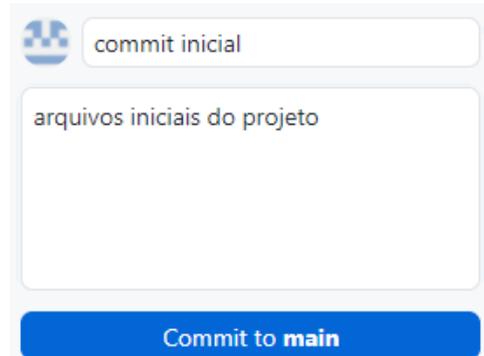
10. Adicione a pasta de repositório do projeto ao Unity Hub e abra o projeto no Editor do Unity.

11. Faça um commit inicial e publique o seu repositório.

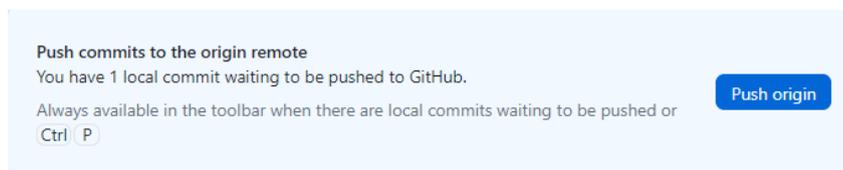


O primeiro conjunto de arquivos que confirmar no seu repositório será incluído no commit inicial. Normalmente faria um commit inicial depois de configurar o seu projeto pela primeira vez, então vamos fazer isso agora.

No campo Resumo abaixo das alterações listadas no seu repositório, digite “commit inicial” e preencha a Descrição (por exemplo, “arquivos iniciais do projeto”). Em seguida, selecione Commit to main para salvar as suas alterações na ramificação “principal” do projeto. Se não sabe o que são ramificações, não se preocupe - abordaremos isso num tutorial posterior.



12. Clique Push origin.



O comando git push permite que envie (ou em tradução literal, empurre) os commits do repositório Git local para o repositório remoto.

Para poder fazer um git push para o repositório remoto, deve garantir que todas as alterações no repositório local foram confirmadas.

13. Pull origin.

O comando git pull é usado para procurar e descarregar conteúdo de repositórios remotos e fazer a atualização imediata ao repositório local para que os conteúdos sejam iguais.

