



Universidade do Minho
Instituto de Educação

Maurício Braga Morais

**Desenvolvimento de Apps Android e Utilização
de Teste de Pensamento Computacional num
Curso Profissional de Programador/a
de Informática**



Universidade do Minho
Instituto de Educação

Maurício Braga Morais

**Desenvolvimento de Apps Android e Utilização
de Teste de Pensamento Computacional num
Curso Profissional de Programador/a
de Informática**

Relatório de Estágio
Mestrado em Ensino de Informática

Trabalho efetuado sob a orientação da
Doutora Maria Altina Silva Ramos

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.



Atribuição-NãoComercial-Compartilha Igual CC BY-NC-SA

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

Agradecimentos

Agradeço à Universidade do Minho, aos meus Professores, aos meus Colegas do Mestrado, à Escola onde realizei a Intervenção Pedagógica, aos Alunos do 11ºTPI, à DGES, e a todos mais que participaram neste projeto de formação de um novo professor.

Em especial, agradeço,

Ao Professor Doutor António Osório,

À Professora Doutora Altina Ramos,

Ao Professor José Pereira,

À Professora Doutora Susana Caires,

Ao Professor Doutor Marcos Román–González,

Ao Sr. Rui Silva,

Ao Ricardo, Pedro Reis, Cátia, Isabel, Andreia, André, Rui, Nuno Ferreira e Marco,

À minha Família, tão pequena e tão querida, minha Mãe, meu Pai e minha Irmã,

E à Olivia, minha sobrinha, que, na sua alternância entre choros e sorrisos, lembra-me que não somos máquinas.

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.

Desenvolvimento de Apps Android e Utilização de Teste de Pensamento Computacional num Curso Profissional de Programador/a de Informática

Resumo

O presente relatório documenta a intervenção pedagógica supervisionada, realizada numa turma do 11º ano do Curso Programador/a de Informática, no âmbito da Unidade Curricular Estágio, do Mestrado de Ensino de Informática da Universidade do Minho.

O tema de ensino-aprendizagem foi a construção de interfaces gráficas e programação de interatividades, com o objetivo de criar programas para dispositivos móveis Android. A questão de investigação relacionou-se com o pensamento computacional, e com a utilização de ferramentas a ele associadas, no contexto do tema em estudo. Como parte da estratégia de motivação, segundo o modelo Atenção-Relevância-Confiança-Satisfação (ARCS), de Keller, utilizou-se o Android Studio, na perspetiva dos tópicos introdutórios da aprendizagem em espiral, de Bruner. As características dos alunos levaram-me à opção pela Aprendizagem Baseada em Casos (CBL), sobretudo porque (i) na CBL, os alunos já têm algum conhecimento prévio, receberam formação antecipada, e (ii) após uma sessão de CBL, há pouca, ou nenhuma, tarefa de autoestudo. A seleção das Apps, feita em conjunto com o Professor Cooperante, seguiu os seguintes critérios: (i) introduzir conceitos básicos Android num nível de complexidade adequado para o 11º ano de escolaridade, (ii) ser enquadrável no período previsto para a minha intervenção, (iii) possuir uma estrutura favorável ao método demonstrativo-interrogativo, na fase inicial da aula, e à atividade prática dos alunos, na fase seguinte, (iv) estar relacionada com o tema da educação física, referida positivamente pelos alunos no inquérito do início do ano.

Ao longo da minha intervenção, os alunos desenvolveram duas Apps: *Índice de Massa Corporal* (IMC) e *Índice Cintura-Quadril* (ICQ), e enviaram-me os seus trabalhos individuais. A avaliação dos trabalhos revelou que mais de 50% dos alunos cumpriram mais de 50% dos critérios de correção adotados nas duas Apps, sendo a segunda mais complexa que a primeira. A auto e heteroavaliação dos alunos foi favorável à intervenção realizada.

Ao início e ao final da intervenção pedagógica, foi aplicado o teste de Pensamento Computacional (CTt), de Román-González, adaptado por mim à linguagem Java. Os resultados médios da turma apontaram no mesmo sentido dos já descritos para alunos espanhóis, singapurenses e suíços, do ensino secundário. O resultado individual do CTt pode funcionar como um teste diagnóstico do estágio em lógica de programação, colaborando para delinear a zona de desenvolvimento proximal, de Vygotsky, e para os critérios da estratégia de Programação em Pares.

Palavras-chave: Android Studio, Aprendizagem Baseada em Casos, Criação de App, Programação para Dispositivos Móveis Android, Teste de Pensamento Computacional.

Android App Development and Computational Thinking Test in a Computer Programmer VET Course, ISCED-4

Abstract

The present report documents the supervised pedagogical intervention, carried out in an 11th year class of a Computer Programmer Course, within the scope of a Curricular Internship for the Master of Computer Science Teaching course at the University of Minho.

The teaching-learning process was the construction of graphical interfaces and programming of interactivity, with the objective of creating programs for Android mobile devices. The research question was related to computational thinking and the use of tools associated with it, in the context of the subject under study. As part of the motivation strategy, according to Keller's Attention-Relevance-Confidence-Satisfaction (ARCS) model, Android Studio was used, from the perspective of Bruner's introductory spiral learning topics. The characteristics of the students led me to opt for Case-Based Learning (CBL), mainly because (i) at CBL, students already have some prior knowledge, received early training, and (ii) after a CBL session, there is little, or no, self-study task. The selection of Apps, made together with Professor Cooperante, followed the following criteria: (i) introduce basic Android concepts at a level of complexity suitable for the 11th grade, (ii) be within the period foreseen for my intervention, (iii) have a structure favorable to the demonstrative-interrogative method, in the initial phase of the class, and to the practical activity of the students, in the following phase, (iv) to be related to the theme of physical education, mentioned positively by the students in the initial survey of the year.

During my intervention, students developed two Apps: Body Mass Index (BMI) and Waist-Hip Index (WCI), and sent me their individual works. The evaluation of the works revealed that more than 50% of the students fulfilled more than 50% of the correction criteria adopted in the two Apps, the second being more complex than the first. The students' self and hetero-assessment was favorable to the intervention performed.

At the beginning and at the end of the pedagogical intervention, the Computational Thinking Test (CTt), by Román-González, adapted by me to the Java language, was applied. The average results of the class pointed in the same direction as those already described for Spanish, Singaporean and Swiss secondary school students. The individual CTt result may work as a diagnostic test of the stage in programming logic, helping to delineate Vygotsky's zone of proximal development, and for the criteria of the Pair Programming strategy.

Keywords: Android Studio, App Creation, Case-Based Learning, Computational Thinking Test, Programming for Android Mobile Devices,

Índice Geral

Direitos de Autor e Condições de Utilização do Trabalho por Terceiros.....	ii
Agradecimentos.....	iii
Declaração de Integridade.....	iv
Resumo.....	v
Abstract.....	vi
Índice de Conteúdos.....	vii
Índice de Figuras.....	ix
Índice de Tabelas.....	xii
Lista de Abreviaturas, Acrónimos e Siglas.....	xiii

Índice de Conteúdos

1. Introdução.....	1
2. Revisão da Literatura.....	2
2.1. TIC e Programação Informática.....	2
2.2. Iniciativas de Promoção da Programação Informática.....	3
2.3. Apps Para Alunos & Apps De Alunos da Escolaridade Obrigatória.....	4
2.3.1. Apps para Alunos.....	5
2.3.2. Apps de Alunos.....	6
2.4. O Pensamento Computacional na Escolaridade Obrigatória em Portugal.....	7
2.5. Lecionação e Avaliação do Pensamento Computacional.....	9
2.6. O Teste de Pensamento Computacional de Román-González.....	10
3. Contextualização.....	12
3.1. Orientadores e Área da Intervenção Pedagógica.....	12
3.2. A Escola.....	12
3.3. A Disciplina, os Horários e a Sala de Aula.....	13
3.4. A Turma.....	14
3.5. Os Principais Acontecimentos que Impactaram a Escola.....	16
4. Plano de Intervenção Pedagógica Supervisionada.....	19
4.1. Tema.....	19
4.2. Objetivos.....	19
4.3. Estratégias de Ação da Intervenção.....	19
4.4. Estratégias de Avaliação da Intervenção.....	22
4.5. Justificação das Estratégias da Intervenção.....	23
5. Intervenção Pedagógica.....	27
5.1. Aulas.....	29
5.1.1. Sessão 1 – terça-feira 15 de março 2022, Turno 2.....	29
5.1.2. Sessão 2 – terça-feira 15 de março 2022, Turno 1.....	36
5.1.3. Sessão 3 – sexta-feira 18 de março 2022, Turno 1.....	39
5.1.4. Sessão 4 – sexta-feira 18 de março 2022, Turno 2.....	44
5.1.5. Sessão 5 – terça-feira 22 de março 2022, Turno 2.....	47
5.1.6. Sessão 6 – terça-feira 22 de março 2022, Turno 1.....	54

5.1.7.	Sessão 7 – sexta-feira 25 de março 2022, Turno 1	57
5.1.8.	Sessão 8 – sexta-feira 25 de março 2022, Turno 2	60
5.1.9.	Sessões 9 e 10 – terça-feira 29 de março 2022, Turnos 2 e 1	63
5.1.10.	Sessões 11 e 12 – sexta-feira 1 de abril 2022, Turnos 1 e 2	66
5.2.	Resultados e Análise	73
5.2.1.	Grelha de Observação dos Alunos	73
5.2.2.	Apps IMC e ICQ	75
5.2.3.	Testes de Pensamento Computacional	78
5.2.4.	Avaliação ao Professor Estagiário	84
5.2.5.	Autoavaliação dos Alunos.....	86
5.3.	Reflexões	89
5.3.1.	O meu Desempenho na Intervenção	89
5.3.2.	Limitações da Intervenção	90
5.3.3.	Aplicabilidade da Intervenção na Vida Profissional	91
5.4.	Considerações Finais sobre a Intervenção Pedagógica Supervisionada	92
6.	Reflexões sobre o Trabalho Futuro	93
	Referências Bibliográficas.....	95
	Apêndices.....	102
	I Teste de Pensamento Computacional (RG, 2016), convertido à Java	103
	II Declaração de Consentimento Informado, Livre e Esclarecido	117
	III Grelha de Observação dos Alunos	120
	IV Questionário de Avaliação ao Professor Estagiário	121
	V Grelha de Avaliação do Teste de Pensamento Computacional	122
	VI Excertos das 14 Entrevistas Individuais sobre o Teste de Pensamento Computacional	123
	VII Grelha de Avaliação das Apps IMC e ICQ	126
	VIII Excertos das 16 Entrevistas Individuais sobre as Apps IMC e ICQ	127
	IX Questionário de Autoavaliação do Aluno	130
	X Código-Fonte App IMC	131
	XI Código-Fonte App ICQ	134
	Anexos.....	138
	A. Definições Adotadas neste Relatório	139
	B. Autorização do Autor do Teste de Pensamento Computacional,	141
	C. Autorização do Autor da Tradução Portuguesa do Teste de Pensamento Computacional.....	142
	D. Autorização da Professora Orientadora Supervisora.....	143
	E. Autorização da Direção-Geral da Educação (DGE)	144
	F. Autorização do Professor Orientador Cooperante	146
	G. Tradução da lição 1.1 do Codelabs de Android Studio em Java para Português	147
	H. Correspondência entre Expressões Informáticas usadas em Portugal e no Brasil	158
	I. Medidas de Tendência Central em Distribuição Normal e Não Normal	159

Índice de Figuras

Figura 1. Esquema Resumido do Ensino de Informática na Escolaridade Obrigatória	2
Figura 2. Olimpíadas Nacionais de Informática, logotipo (ONI, 2017).....	3
Figura 3. Hora do Código, logotipo (ANPRI, 2021b).....	4
Figura 5. Projeto Programação e Robótica na Escola, logotipo (DGE, 2022)	4
Figura 4. PAPTICe, logotipo (ANPRI, 2022)	4
Figura 6. Níveis Hierárquicos das Linguagens de Programação (adapt. Null & Lobur, 2019)	5
Figura 7. Aplicação Grasshopper de Iniciação à Programação (adapt. Markezine, 2022).....	5
Figura 8. App Hábitos de Vida Saudável (Abreu, 2020).....	6
Figura 9. App Vencedora, PAPTICe 2021 (adapt. ANPRI, 2022)	7
Figura 10. App S-Notas, (adapt. Reis, 2015)	7
Figura 11. Semelhanças entre Pensamentos Computacional e Matemático (Shute et al., 2017)	8
Figura 12. Teste Original CTt, Resposta em Setas; Tarefa: Sequência, (Román-González, 2016)	10
Figura 13. Teste Original CTt, Resposta em Blocos; Tarefa: Depuração (Román-González, 2016).....	11
Figura 14. Distribuição da Sala de Aula 11ºTPI: Turno 1 (esquerda) e Turno 2 (direita).....	13
Figura 15. Tarefas do Ensino na Sala de Aula (Lopes 2003, citado por Caires, 2021).....	16
Figura 16. Modelo Ecológico de Bronfenbrenner do Desenv. Humano (citado por Dias, 2014, p.15) ..	16
Figura 17: Casos COVID-19, 03/2020 a 01/2022 (adapt. Inácio & Sequeira, 2022)	17
Figura 18. Demonstração em Braga, abril de 2022	18
Figura 19. Espiral de Aprendizagem de Bruner (adaptado de Vemuru et al., 2013).....	23
Figura 20. Pré-requisitos do Aprendiz para Beneficiar das Metodologias Ativas, num Contexto Educacional Formal de Ensino de Ciências (adapt. Brod, 2021)	25
Figura 21. Verbos & Atividades segundo a Taxonomia de Bloom Revisada (Hokkanen, 2015)	26
Figura 22. Cabeçalho do Teste de Pensamento Computacional	30
Figura 23. Apps <i>Hello Toast</i> (esquerda) e IMC (direita)	31
Figura 24. Excerto da Página da Disciplina na Plataforma Nónio.....	32
Figura 25. Atributo <i>onClick</i> e Manipulador de Eventos do Botão Limpar	32
Figura 26. Teste <i>Logcat</i> do Manipulador de Eventos Limpar	32
Figura 27. Mensagem <i>Logcat</i> do Botão Limpar	33
Figura 28. Renomeação dos IDs dos Elementos <i>View</i>	33
Figura 29. Declaração de Objetos <i>Widget</i>	33
Figura 30. Ligação Código-Tela com <i>findViewById</i>	34
Figura 31. Manipulador de Eventos Limpar com <i>setText</i>	34

Figura 32. Teste de Receção do Valor do Peso	35
Figura 33. Questão 10 do teste CTt; Opção C Corrigida.....	37
Figura 34. <i>Hints</i> nas Caixas de Texto e no Rótulo IMC.....	38
Figura 35. Teste <i>getText</i> para Peso	40
Figura 36. Conversão do Peso e Altura para <i>String</i>	40
Figura 37. Conversão de <i>Strings</i> para <i>Doubles</i> e Teste de Soma de Valores.....	40
Figura 38. Concisão Máxima do Manipulador Calcular	41
Figura 39. Classe IMC, sem o Método <i>Classifica</i>	41
Figura 40. Instanciação do Objeto IMC.....	41
Figura 41. Conversão com <i>Double.toString</i>	42
Figura 42. Método <i>calculaIMC</i> com <i>doubles</i>	42
Figura 43. Conversão com <i>String.format</i>	42
Figura 44. Verificação Errada de Valores Nulos com <i>String == null</i>	43
Figura 45. Verificação Correta de Valores Nulos com <i>String.isEmpty</i>	43
Figura 46. Classe IMC do 2º turno	45
Figura 47. Exemplo de Instanciação de Objeto IMC com Variáveis Erradas	45
Figura 48. Exemplo de Conversão Errada para <i>String</i>	45
Figura 49. Verificação Incorreta: Condicionais <i>if</i> sem retorno.....	46
Figura 50. Método <i>Classifica</i> da Classe IMC.....	46
Figura 51. Editor de Componentes Android SDK	48
Figura 52. App ICQ em Orientação Vertical (esquerda) e Horizontal (direita).	49
Figura 53. Construção do Cabeçalho: Modificação do rótulo <i>Hello World</i>	50
Figura 54. Construção do Cabeçalho: <i>Padding</i> e <i>Background Color</i>	50
Figura 55. Construção do Rodapé: <i>ConstraintLayout</i>	50
Figura 56. Construção do Rodapé: Botões	51
Figura 57. Corpo com <i>ScrollView</i>	51
Figura 58. Primeiro Elo da Cadeia Vertical	52
Figura 59. Cadeia Centrada Horizontalmente	53
Figura 60. Rótulos e Caixas de Texto com as suas <i>Baselines</i> Alinhadas	53
Figura 61. Verificação Correta: Condicionais <i>if</i> com retorno	54
Figura 62. <i>Layout</i> da App ICQ em Orientação Horizontal	56
Figura 63. Classe ICQ: Construtor com 2 parâmetros.....	58
Figura 64. Manipulador Calcular: Instanciação de Objeto <i>icq2</i>	59

Figura 65. Instanciação de Objeto icq3 com Variáveis de icq2	59
Figura 66. Atributos <i>Hint</i> e <i>InputType</i>	60
Figura 67. Prevenção de <i>inputs</i> Nulos com <i>requestFocus</i>	61
Figura 68. Manipulador Calcular: Instanciação de 3 Objetos ICQ	62
Figura 69. Classe ICQ: Método <i>Classifica</i>	64
Figura 70. Demonstração do Debugger.....	65
Figura 71. Diapositivo sobre <i>Activities</i>	67
Figura 72. App DuasAtividades_LevarDados	68
Figura 73. Instanciação de um Objeto <i>Intent</i> ; Método <i>startActivity</i>	69
Figura 74. Manipulador Voltar: Método <i>finish</i>	69
Figura 75. Declaração da Chave Extra_Dados com Domínio.....	69
Figura 76. Manipulador Enviar Com Dados: Método <i>putExtra</i>	70
Figura 77. Receção do <i>Intent</i> com Método <i>getIntent</i>	70
Figura 78. Método <i>getStringExtra</i>	70
Figura 79. Verificação de Extras Vazios Incompleto	71
Figura 80. Verificação de Extras Vazios com Mensagem <i>Toast</i>	71
Figura 81. Verificação de Extras Vazios com o Método <i>hasExtra</i>	72
Figura 82. Assiduidade e Pontualidade, Turnos 1 e 2	73
Figura 83. Comportamento, Turnos 1 e 2	73
Figura 84. Participação, Turnos 1 e 2	74
Figura 85. Empenho, Turnos 1 e 2	74
Figura 86. Autonomia, Turnos 1 e 2	74
Figura 87. App IMC: % de Acertos por Tópico Avaliado (n=13)	75
Figura 88. App ICQ: % de Acertos por Tópico Avaliado (n=15)	77
Figura 89. Notas das Apps IMC e ICQ - 11ºTPI.....	78
Figura 90. N° de Acertos por Aluno nos Testes de 15/03 e 01/04/2022.....	79
Figura 91. Frequências de Pontuação no teste CTt em 15/3 e 01/04.....	81
Figura 92. Índice de Dificuldade pi das questões do CTt, 11º TPI	83
Figura 93. Autoavaliação de Desempenho dos Alunos do 11º TPI sobre a Intervenção Pedagógica, em percentagem de respostas V/F	88
Figura 94: Operacionalização da Estratégia Pedagógica Pair Programming (Pedro et al., 2017)	93
Figura 95. Zona de Desenvolvimento Proximal (ZPD), (adapt. Basawapatna et al., 2013).....	94
Figura 96: Seis Conceitos Básicos do Pensamento Computacional (ACARA, 2020).....	139

Figura 97: Quatro Conceitos Básicos do Pensamento Computacional (Kaoneill, 2020)	139
--	-----

Índice de Tabelas

Tabela 1. Aplicações do CTt no Ensino Secundário.....	11
Tabela 2. Excerto do Questionário da Escola, aplicado no início de 21/22	15
Tabela 3. Calendário da Intervenção Pedagógica Sumariada	27
Tabela 4. Planificação dos Objetivos das Sessões.....	28
Tabela 5. Planificação da Sessão 1	29
Tabela 6. Planificação da Sessão 2	36
Tabela 7. Planificação da Sessão 3	39
Tabela 8. Planificação da Sessão 4	44
Tabela 9. Planificação da Sessão 5.....	47
Tabela 10. Planificação da Sessão 6.....	54
Tabela 11. Planificação da Sessão 7	57
Tabela 12. Planificação da Sessão 8.....	60
Tabela 13. Planificação das Sessões 9 e 10.....	63
Tabela 14. Planificação das Sessões 11 e 12.....	66
Tabela 15. Critérios de correção da App ICQ. Funcionalidade - Método/Atributo – Pontuação	76
Tabela 16. Pontuação dos Acertos Individuais e Análise Descritiva dos Resultados do 11ºTPI, em 15/03 e 01/04.....	80
Tabela 17. Teste CTt, Temas, Tarefas e Índice de Dificuldade, pi, do 11ºTPI	82
Tabela 18. Avaliação dos Alunos Relativamente à Ação do Professor em Sala de Aula	84
Tabela 19. Avaliação dos Alunos sobre a Planificação do Curso	85
Tabela 20. Avaliação Global Quantitativa sobre o Professor Estagiário.....	85
Tabela 21. Autoavaliação Qualitativa e Quantitativa dos Alunos.....	86
Tabela 22: Autoavaliação das Vertentes Atuação e Planificação da Intervenção.....	90

Lista de Abreviaturas, Acrónimos e Siglas

- 11ºTPI - 11º ano do Curso Profissional de Programador/a Informática
- ACMDI - Avaliação e Conceção de Materiais Didáticos de Informática
- AD - Atividades Desplugadas
- AEs - Aprendizagens Essenciais
- ANPRI – Associação Nacional dos Professores de Informática
- ANQEP - Agência Nacional para a Qualificação e Ensino Profissional
- Aplicação Web - *Web application*, website otimizado para dispositivo móvel, não é necessário instalar a aplicação no dispositivo móvel.
- CBL - *Case-Based Learning*; Aprendizagem Baseada em Casos
- CPPI - Curso Profissional de Programador/a de Informática
- CT - *Computational Thinking*, Pensamento Computacional
- CTt – *Computational Thinking test*; Teste de Pensamento Computacional de Román-González
- ENSICO - Associação para o Ensino da Computação
- GOA – Grelha de Observação dos Alunos
- iVET - *Initial Vocational Education and Training*, Educação e Formação Profissional Inicial (EFPI)
- IDE – *Integrated Development Environment*; Ambiente de Desenvolvimento Integrado
- PAP - Prova de Aptidão Profissional
- PAPTICe - Concurso Nacional de PAP de Informática, promovida pela ANPRI
- PBL - *Problem-Based Learning*, Aprendizagem Baseada em Problemas
- PIPS – Projeto de Intervenção Pedagógica Supervisionada
- PRIMM - *Predict-Run-Investigate-Modify-Make*; Prever-Executar-Investigar-Modificar-Criar
- Programação – Abreviatura de programação informática
- RTP – Rádio e Televisão de Portugal
- Software – Programa informático: ambas as palavras são usadas neste texto.
- SPM – Sociedade Portuguesa de Matemática
- SQL - *Structured Query Language*; Linguagem de Consulta Estruturada
- UFCDs - Unidades de Formação de Curta Duração
- VET - *Vocational Education and Training*; Educação e Formação Profissional (EFP)
- XML - *Extensible Markup Language*; Linguagem de Marcação/Notação

1. Introdução

O presente relatório, elaborado no âmbito da unidade curricular Estágio, do Mestrado em Ensino da Informática da Universidade do Minho, refere-se à Prática de Ensino Supervisionada desenvolvida numa Escola Secundária Pública de Braga. A intervenção realizou-se numa turma do 11º ano do Curso Profissional de Programador/a de Informática, no ano letivo 2021-22, especificamente na disciplina de Programação Móvel, módulo 0817.

Tendo como referência o currículo definido para a disciplina, o tema da intervenção incidiu sobre a construção e programação de interfaces gráficas, com vista à criação de aplicações para Android.

Organizei o relatório em seis capítulos: 1- Introdução; 2- Revisão da Literatura; 3- Contextualização; 4- Plano de Intervenção Pedagógica; 5- A Intervenção Pedagógica; e 6- Reflexões sobre o Trabalho Futuro. A seguir, as Referências Bibliográficas, os Apêndices e os Anexos. Os conteúdos dos Apêndices e Anexos foram agrupados preferencialmente em ordem cronológica de criação/utilização. No anexo A constam as definições adotadas neste relatório, para casos em que não há uma definição única, designadamente, o que é Informática, o que é Ensino de Informática e o que é Pensamento Computacional. As figuras e tabelas sem indicação de autor são de minha autoria.

2. Revisão da Literatura

2.1. TIC e Programação Informática

A definição do *Perfil do Aluno à Saída da Escolaridade Obrigatória* (PASEO) foi fundamental para clarificar os conhecimentos, capacidades e competências pretendidos para a Informática (Martins et al., 2017). Na sequência do PASEO, o Decreto-Lei 55/2018, de 6 de julho reviu o currículo dos ensinos básico e secundário e, entre outros aspetos, decidiu a expansão da lecionação das Tecnologias de Informação e Comunicação (TIC) a toda a escolaridade obrigatória (Figura 1). Ao mesmo tempo, o DL 55/2018 introduziu uma maior liberdade para que cada escola possa estabelecer estratégias, no sentido de os seus alunos desenvolverem adequadamente as competências previstas no PASEO, através do conhecimento do meio onde esteja inserida. Assim, no 2.º ciclo, os 325 minutos semanais da área Educação Artística e Tecnológica podem distribuir-se entre TIC e mais 3 disciplinas; no 3.º ciclo, os 175 minutos de Educação Artística e Tecnológica podem distribuir-se entre TIC e mais 2 disciplinas. Na vertente profissional do ensino secundário, está previsto todos os cursos terem a disciplina de TIC, numa carga horária total de 100 horas; não obstante, o DL 55/2018 também prevê que, em função da necessidade de reforço das aprendizagens, em vez de TIC, a escola pode optar por uma Oferta de Escola, de frequência obrigatória. Em relação ao ensino secundário, o currículo da vertente Científico-Humanista prevê apenas uma disciplina, opcional, *Aplicações Informáticas B*, no 12º ano. Com vista a colmatar o défice de competências e conhecimentos dos alunos desta vertente, a Associação Nacional dos Professores de Informática (ANPRI) entregou ao Ministério da Educação uma proposta de criação de uma disciplina bienal de Informática, para o curso Científico-Humanístico de Ciências e Tecnologias (ANPRI, 2021a).

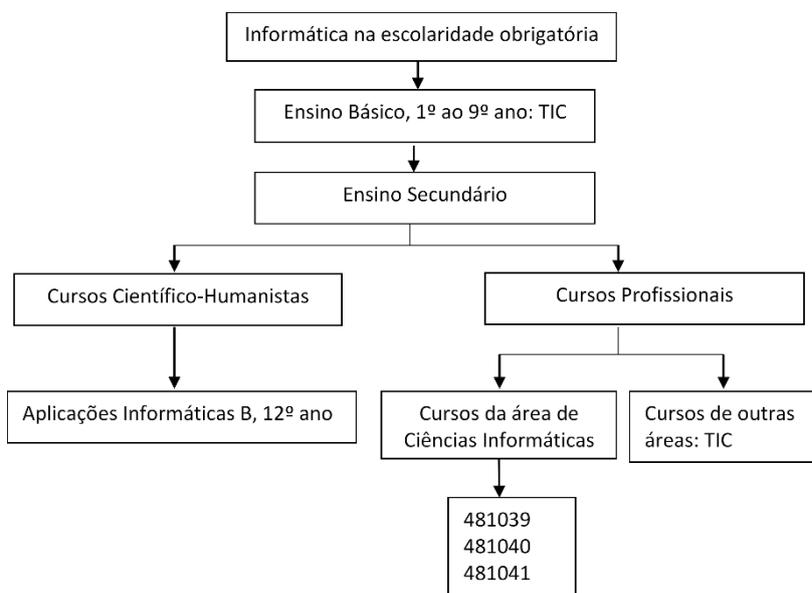


Figura 1. Esquema Resumido do Ensino de Informática na Escolaridade Obrigatória

Quanto ao 1º ciclo do Ensino Básico, não está prevista a lecionação especializada de TIC, uma vez que a revisão curricular consagra a monodocência, com exceção da disciplina de Inglês (Conselho de Ministros, 2018).

O que dizem as Aprendizagens Essenciais (AEs) de TIC, do 2º e 3º ciclos, relativamente à programação informática? Nas AEs do 5º, 6º e 9º anos, as atividades de programação indicadas são a introdução à programação por blocos, a programação para interagir com robots e outros artefactos tangíveis, e a exploração dos conceitos de programação para dispositivos móveis, respetivamente. No 7º e 8º anos, as AEs não referem atividades de programação (DGE, 2018a). No ensino secundário, vertente científico-humanista, as AEs de *Aplicações Informáticas B* apontam para a elaboração de programas simples, utilizando uma linguagem de programação imperativa (DGE, 2018b).

Nos cursos profissionais não pertencentes à área 481, Ciências Informáticas, a disciplina de TIC possui o módulo opcional 2, *Introdução à Programação*, que prevê a realização de atividades orientadas para a criação e análise de algoritmos e das diversas fases de programação de uma aplicação - decompor, testar, depurar e reformular. Nos cursos profissionais da área 481, as linguagens de programação estão presentes nos cursos 481040, *Programador/a de Informática* (C/C++, SQL, Java, C#, Asp.Net, Python), e 481039, *Técnico/a de Informática – Sistemas* (C/C++, Java). A lecionação de SQL também ocorre no curso *Técnico de Multimédia*, área 213, Audiovisuais e Produção dos Media (ANQEP, 2022).

2.2. Iniciativas de Promoção da Programação Informática

As iniciativas já consolidadas são as *Olimpíadas de Informática* (Figura 2), a *Hora do Código* (Figura 3) e o *PAPTICe* (Figura 5). A estas, soma-se o projeto *Programação e Robótica na Escola*, desde março de 2022 (Figura 4).



Figura 2. Olimpíadas Nacionais de Informática, logotipo (ONI, 2017)

As *Olimpíadas Nacionais de Informática* (ONI) são um concurso de programação destinado a alunos do ensino secundário, de carácter individual. Para resolver os problemas propostos, as linguagens de programação permitidas são o C, C++, Pascal, Java e Python. A organização da ONI está a cargo do Professor Pedro Ribeiro, da Universidade do Porto.

Os melhores classificados podem representar Portugal nas Olimpíadas Internacionais de Informática (IOI), que realiza-se anualmente desde 1989 (Azevedo, 2021). Em 2022, a IOI ocorreu na Indonésia, com mais de 300 participantes, sendo que dois alunos portugueses receberam menção honrosa (IOI, 2022).

Iniciado em 2013, nos Estados Unidos, a *Hora do Código* é um evento anual, geralmente em dezembro, que procura despertar o interesse de professores e alunos do ensino básico para a programação informática e a importância da literacia digital, através de atividades lúdicas. As escolas podem inscrever-se diretamente no site www.code.org ou através da ANPRI (ANPRI, 2021a).



Figura 3. Hora do Código, logotipo (ANPRI, 2021b)

O Concurso Nacional *PAPTICe* é uma iniciativa promovida pela ANPRI, com o apoio de múltiplos parceiros, que visa reconhecer o trabalho dos alunos dos cursos profissionais nos seus projetos de fim de curso, denominados PAP (Prova de Aptidão Profissional). As três áreas em concurso são Software, Artefactos (Hardware, Redes, Robótica e Eletrónica) e Multimédia. O concurso, cuja primeira edição foi em 2018, apoia-se em critérios de excelência, inovação, criatividade e empreendedorismo, sendo distinguido com o selo “uma ação INCoDe.2030”. Na edição de 2022, concorreram 34 projetos de 21 instituições de ensino (ANPRI, 2022).



Figura 5. PAPTICe, logotipo (ANPRI, 2022)

O projeto *Programação e Robótica na Escola* é promovido pela Direção Geral de Educação (DGE, 2022). O webinar de lançamento esteve a cargo do Professor João Piedade (Piedade & Dorotea, 2020).



Figura 4. Projeto Programação e Robótica na Escola, logotipo (DGE, 2022)

2.3. Apps Para Alunos & Apps De Alunos da Escolaridade Obrigatória

A expressão App, abreviatura de *Application*, corresponde a um programa de computador, ou parte de um software, projetado para uma finalidade específica, que se pode baixar num telemóvel ou outro dispositivo móvel (Cambridge Dictionary, 2022).

Além do código para o cálculo das variáveis, ou para a ação entre classes, ou ambos, a programação de uma App inclui a parte visual, que vai ser manuseada pelo utilizador final, a qual também tem de ser codificada.

O programador tem à sua disposição uma linguagem de 3ª geração para construir o ecrã do utilizador final, que vai funcionar em linguagem de 5ª geração, e conta com os software de apoio para fazer chegar as instruções à linguagem de máquina, 1ª geração (Figura 6).

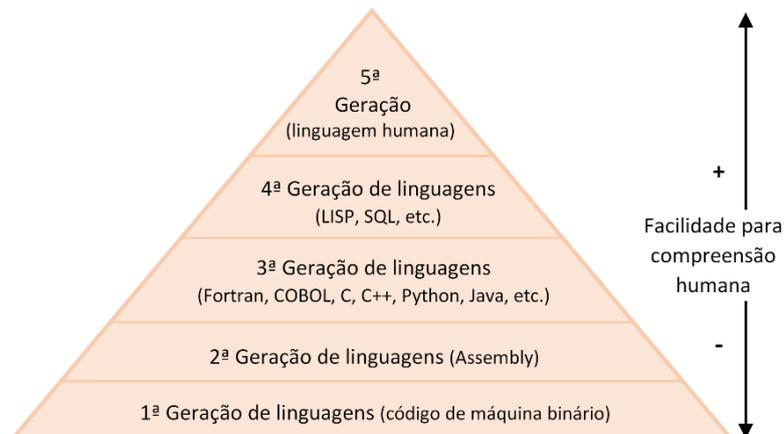


Figura 6. Níveis Hierárquicos das Linguagens de Programação (adapt. Null & Lobur, 2019)

2.3.1. Apps para Alunos

Nos dias de hoje, é ponto assente para os professores que o telemóvel pode ser usado em atividades escolares (Silva et al., 2020). Neste sentido, observa-se uma tendência mundial de criação de aplicações para dispositivos móveis, como recurso de ensino-aprendizagem (E-A), tanto da parte de grandes e pequenas empresas de software como de teses e artigos académicos.

Em junho de 2021, a Google lançou o Grasshopper com as instruções em Português (Figura 7). Trata-se de uma aplicação web gratuita, projetada como uma série de puzzles e questionários em complexidade crescente, nos quais os conceitos da lógica de programação vão sendo inseridos suavemente, em linguagem JavaScript. No âmbito da unidade curricular ACMDI, da Universidade do Minho, propus o Grasshopper como material didático para o 9º ano, numa disciplina de iniciação à programação (Morais, 2022). O site do Ministério da Educação de Portugal disponibiliza a App em: <https://appseducacao.rbe.mec.pt/2019/10/11/grasshopper/>



Figura 7. Aplicação Grasshopper de Iniciação à Programação (adapt. Markezine, 2022)

A monitorização da dieta e atividade física, adaptada de uma App Android, foi aplicada por Abreu (2020) em alunos do 3º ciclo e do Secundário, no Fundão e na Covilhã (Figura 8). A App foi planeada numa perspetiva aliciante e adaptada aos jovens. Na sua globalidade, a App possui funcionalidades de gamificação, informação, planos de dieta, treino, dicas e curiosidades, cálculo de passos e calorias. Os jovens envolvidos no estudo, na sua maioria, manifestaram aprovação à iniciativa.

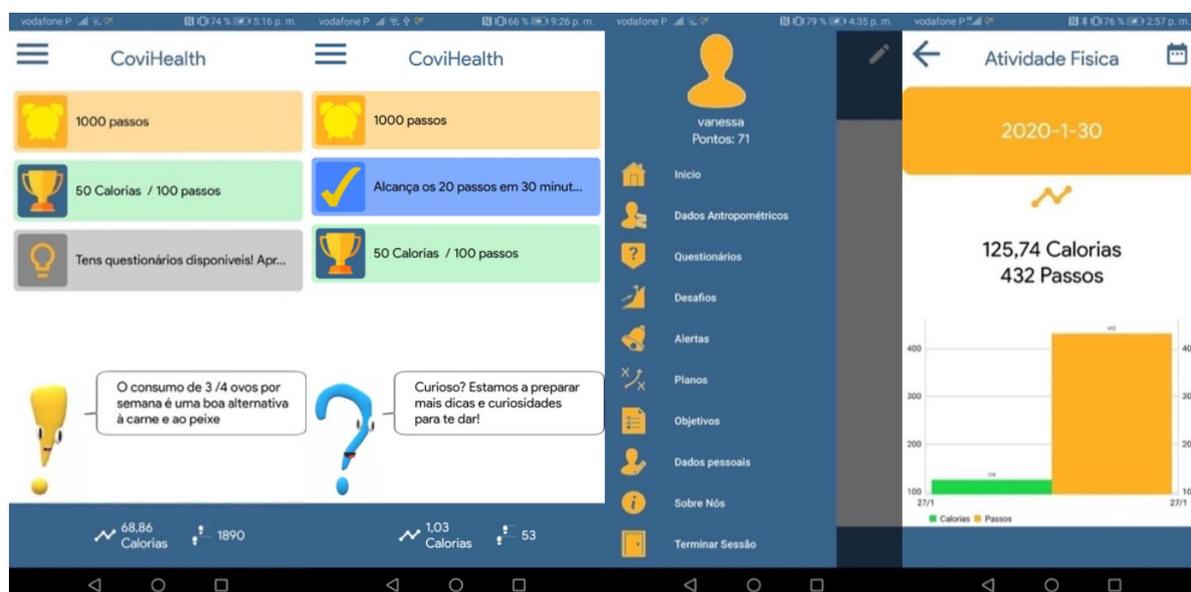


Figura 8. App Hábitos de Vida Saudável (Abreu, 2020)

A aprendizagem de conceitos e lógica de programação, em ambiente MIT App Inventor, foi descrita com alunos do 12º ano de Cursos Científico-Humanísticos, em Évora. A análise dos resultados indicou melhorias na autonomia e na consolidação dos conceitos (Mateus, 2021).

Nunes (2019) implementou a programação do teorema de Pitágoras, com alunos do 9º ano e MIT App Inventor, e testou a sua utilização com alunos do 8º ano, num colégio em Lisboa.

2.3.2. Apps de Alunos

As informações sobre trabalhos de programação móvel desenvolvidos por alunos da escolaridade obrigatória em Portugal foram obtidas, na sua maioria, no website da ANPRI, na seção referente ao concurso *PAPTICe*, que apresenta a lista dos trabalhos em concurso. De 2018 a 2021, pelo menos 9 projetos estiveram em concurso, sendo que, em 2021, o projeto *give IT easy*, relacionado com a doação de alimentos e bens de primeira necessidade, foi o vencedor na categoria software (Figura 9). O facto de este projeto ter três autores pode indicar uma elevada carga horária de dedicação ao tema, além do necessário acompanhamento dos professores orientadores.

No ano de 2022, o vencedor do PAPTICe na categoria software foi uma App, desenvolvida na Escola Secundária Henriques Nogueira, de Torres Vedras. O projeto denomina-se *Aplicação Móvel e Website para os Utilizadores das Agostinhas*, do aluno Sérgio Gribleac Silva (ANPRI, 2022).

Independentemente do concurso PAPTICe, encontrei dois alunos que divulgaram individualmente as suas Apps de PAPs: a App *S-Notas*, por Diogo Reis, na Escola Secundária de Tomaz Pelayo, em Santo Tirso, em 2015 (Figura 10); e a App *Rastreador de Localização GPS*, por Ricardo Gonçalves, na Escola Técnica Profissional de Mafra, em 2016 (Gonçalves, 2016). Também as escolas podem divulgar informação sobre Apps de alunos, nas suas páginas oficiais. Em 2020, o Agrupamento de Escolas de Canelas, em Vila Nova de Gaia, acolheu três PAPs com Apps: uma App de envio de mensagens, e duas Apps de encomendas de refeições – das quais uma incorpora realidade aumentada (AgrCanelas, 2020). Neste mesmo ano, a Escola Secundária José Saramago, de Mafra, divulgou 5 PAPs de jogos sendo 1 App, denominada *FunnyRunner*, de Bernardo Bento (ESJS-Mafra, 2020).



Figura 9. App Vencedora, PAPTICe 2021 (adapt. ANPRI, 2022)

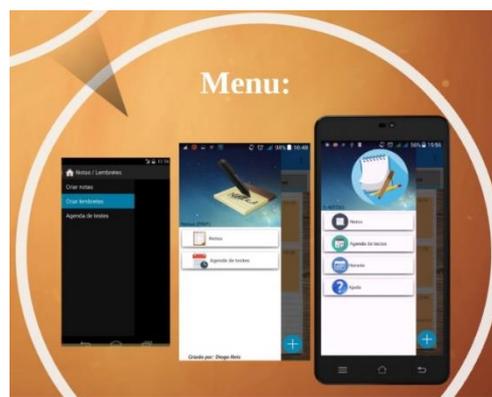


Figura 10. App S-Notas, (adapt. Reis, 2015)

2.4. O Pensamento Computacional na Escolaridade Obrigatória em Portugal

O ensino-aprendizagem de pensamento computacional consta formalmente nas AEs da disciplina de Matemática do 1.º, 3.º, 5.º e 7.º anos de escolaridade, a partir de 2022/23. Em 2023/24, deverá iniciar-se no 2.º, 4.º, 6.º e 8.º anos de escolaridade e, em 2024/25, no 9.º ano (MEC PT, 2021b). A Figura 11 mostra alguns dos elementos comuns aos pensamentos computacional e matemático, como resolução de problemas, modelação, análise de dados e interpretação (Shute et al., 2017).

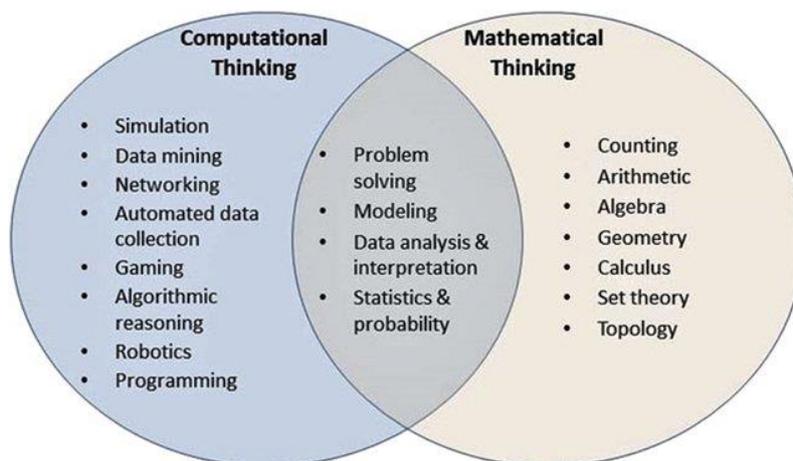


Figura 11. Semelhanças entre Pensamentos Computacional e Matemático (Shute et al., 2017)

A implementação das novas AEs de Matemática foi precedida de um estudo, realizado em 2020, com 593 docentes do ensino pré-escolar e do 1º ciclo de todo o país, no qual investigaram-se as necessidades de formação de educadores e professores. As principais recomendações do estudo apontam para (i) um plano organizacional de oferta formativa, que promova a proximidade de centros de formação e instituições de ensino superior, e (ii) um plano de formação profissional dos docentes, que assegure que possam participar de comunidades de aprendizagem prática (Ramos et al., 2022).

O modo de inserção do pensamento computacional no ensino secundário ainda não está definido. As novas AEs de Matemática no secundário entraram em discussão pública em junho de 2022, com final previsto para setembro de 2022. O grupo coordenador dos trabalhos, criado em 2018, preocupa-se em disponibilizar “ferramentas de análise dos processos sociais que estão na base do exercício de uma cidadania ativa”. “Valorizar o pensamento computacional” é uma das propostas em discussão (TERÃÇA, 2022). Neste sentido, é de referir que o pensamento computacional consta nas AEs da disciplina bienal de Informática, proposta pela ANPRI para o curso Científico-Humanístico de Ciências e Tecnologias (ANPRI, 2021a). Por sua vez, a ENSICO, que propõe o ensino das ciências da computação em toda a escolaridade obrigatória, também defende a ênfase em pensamento computacional neste percurso (ENSICO, 2022). Em 15/09/2022, a Sociedade Portuguesa de Matemática (SPM), emitiu parecer sobre a proposta das novas AEs de Matemática. Relativamente ao pensamento computacional, afirma “a SPM subscreve o parecer da ENSICO no que se refere ao Pensamento Computacional” (SPM, 2022).

2.5. Lecionação e Avaliação do Pensamento Computacional

Entre os vários desafios para a introdução curricular do pensamento computacional, na escolaridade obrigatória, podem citar-se a definição dos conteúdos e metodologias de ensino, assim como os critérios de avaliação da aprendizagem. Em relação à lecionação, um conteúdo consensual é a programação informática, porque o aprendiz interioriza os conceitos computacionais, como sequências, ciclos, condicionais, operadores, paralelismo, e os aplica na prática, ao experimentar e iterar, testar e depurar, reutilizar e remixar, abstrair e modularizar (Hsu et al., 2018; Román-González et al., 2017; Scherer et al., 2019).

Qual a relação entre pensamento computacional e programação informática? Uma analogia de fácil compreensão é proposta por Moreno-León et al. (2019): em resumo, o pensamento computacional é uma capacidade cognitiva humana que ganha vida aquando da programação de computadores, uma competência instrumental. Não obstante, o pensamento computacional pode ser projetado em problemas que não envolvam diretamente tarefas de programação. A relação assemelha-se à existente entre a aptidão verbal e a alfabetização. A aptidão verbal é considerada uma habilidade cognitiva enquanto a alfabetização (leitura e escrita) é classificada como competência instrumental, que requer um processo de ensino-aprendizagem. Um analfabeto pode ter aptidão verbal e usá-la em tarefas que não envolvam leitura nem escrita. Por outro lado, a aptidão verbal não é estanque, pode ser desenvolvida, aperfeiçoada, não somente através da leitura, mas também, por exemplo, pela audição de livros eletrónicos. Do mesmo modo, outras atividades podem contribuir para o pensamento computacional como, por exemplo, a aprendizagem de conceitos e práticas computacionais em atividades desplugadas (AD). Tais atividades não utilizam dispositivos digitais, implicando, geralmente, algum tipo de movimento físico dos participantes. Há evidências empíricas crescentes sobre a eficácia da contribuição de ADs no desenvolvimento do pensamento computacional na escolaridade obrigatória (Marques, 2021; Pereira A., 2021).

Relativamente à avaliação, na investigação de Bocconi et al. (2022), 10 em 24 países europeus declararam dificuldade em avaliar a aprendizagem do pensamento computacional no 1º e 2º ciclos (*primary school level*) e 13 países no 3º ciclo (*lower secondary level*). Neste sentido, as questões que vêm sendo formuladas para o projeto Bebras, em todos os países participantes, inclusive Portugal (Castor Informático, 2021), podem constituir uma base de dados que os professores poderão consultar para a elaboração de testes (Lockwood & Mooney, 2018; Román-González et al., 2019; Tang et al., 2020).

Cutumisu et al. (2019) efetuaram uma revisão sistemática, daí resultando 39 estudos empíricos, sobre pensamento computacional publicados entre 2014 e 2019. Segundo os autores, a maioria dos estudos teve como alvo os alunos da escolaridade obrigatória, usou a forma de teste e avaliou

principalmente o pensamento algorítmico, abstração, decomposição de problemas, pensamento lógico e dados. Não obstante, os autores destacam que poucos estudos foram devidamente validados em ambientes educacionais. O teste de pensamento computacional (CTt) de Román-González (2016), é um dos testes validados e foi adotado, com adaptações, na intervenção pedagógica descrita neste relatório.

2.6. O Teste de Pensamento Computacional de Román-González

Román-González (2016) desenvolveu o CTt aquando dos seus estudos de doutoramento. O teste foi pensado para aplicação na escolaridade obrigatória, sem a necessidade de conhecimento prévio de informática. Como base, para validar avaliações de conhecimento em informática, Román-González refere que adotou o guia prático de Buffum et al. (2015, referido por Román-González, 2016, p.529), estando este guia alinhado com os padrões internacionais para testes psicológicos e educacionais (AERA, APA, & NCME, 2014, referido por Román-González, 2016, p.526). O autor também refere que o CTt é consistente com outros testes de raciocínio computacional voltados para o ensino fundamental/médio, como o Teste de Medição de Habilidades Básicas de Programação (Mühling et al., 2015, referido por Román-González, 2016, p.544) e a Avaliação Comutativa (Weintrop & Wilensky, 2015a, referido por Román-González, 2016, p.556). O CTt foi inicialmente concebido com uma extensão de 40 questões, de escolha múltipla, tendo a versão final 28 questões, após um processo de validação realizado por 20 especialistas. As figuras 12 e 13 exemplificam o formato das perguntas e respostas do teste.

A íntegra do teste CTt encontra-se no anexo F da tese de Román-González, disponível em <https://redined.educacion.gob.es/xmlui/handle/11162/157859>

TPC (Versión 2.0) - ÍTEM 1

<p>¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?</p>	
Opción A	
Opción B	✓
Opción C	
Opción D	

Figura 12. Teste Original CTt, Resposta em Setas; Tarefa: Sequência, (Román-González, 2016)

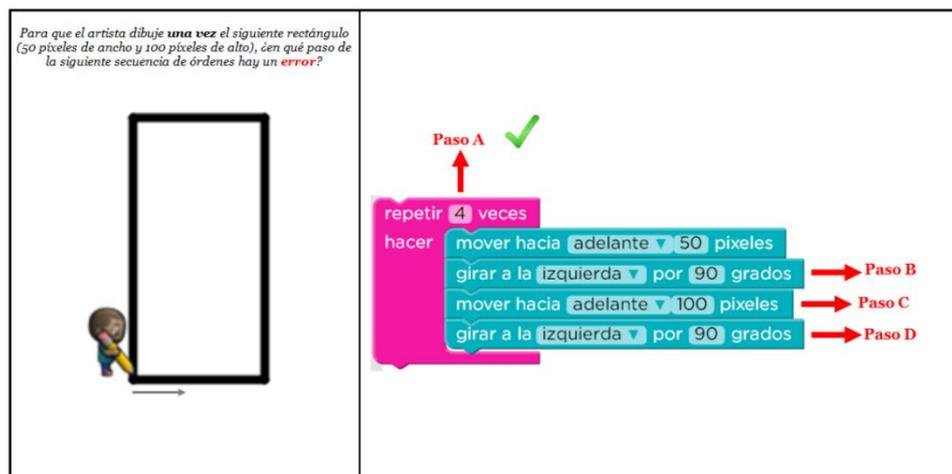


Figura 13. Teste Original CTt, Resposta em Blocos; Tarefa: Depuração (Román-González, 2016)

O CTt vem sendo aplicado em contextos variados, na sua forma original ou modificada, em aplicação única ou em conjunto com outro instrumento de avaliação, em grupos grandes e pequenos, em faixas etárias muito alargadas, podendo citar-se em alunos do 1º ciclo (Marques, 2021), do 2º e 3º ciclos (Çetin et al., 2020), do secundário (Chan et al., 2020; Guggemos et al., 2022) e do ensino superior (Peracaula-Boscha & González-Martínez, 2022), havendo, ainda, a intenção de incorporar o teste na formação de professores do primeiro ciclo do ensino básico (Collado-Sánchez et al., 2021).

Tabela 1. Aplicações do CTt no Ensino Secundário

Ref	País	Nº de Alunos	Ano de escolaridade	Obs.
Román-González, 2016	Espanha	141	10º	Teste CTt original
Chan et al, 2021	Singapura	153	Idades entre 15 e 16 anos	Teste CTt modificado*
Guggemos et al, 2022	Suíça	202	11º ano; idade média 17,23 anos; 53% mulheres	Teste CTt modificado**

* O teste original foi modificado, removeu-se a questão 1.

** O teste original foi modificado, substituíram-se as questões 1, 2, 3, 5, 6, 9 e 15 por outras não disponíveis no artigo.

Na presente intervenção pedagógica, o teste foi transformado, pela primeira vez, em teste misto: as 20 questões com blocos foram convertidas em Java (Apêndice I). Esta conversão elevou o nível de dificuldade do teste, porque linguagens de bloco são mais fáceis de interpretar do que linguagens textuais (Weintrop & Wilensky, 2015), e tornou-o mais especializado porque, para responder às questões, é necessário conhecer os conceitos básicos de Java.

3. Contextualização

3.1. Orientadores e Área da Intervenção Pedagógica

No início de julho de 2021, o Coordenador do Mestrado, Professor Doutor António Osório, informou-me que o meu estágio seria supervisionado pela Professora Doutora Altina Ramos e que o mesmo decorreria num Curso Profissional de Programador/a de Informática (CPPI), numa Escola Secundária Pública, em Braga, tendo como Orientador Cooperante o Professor José Pereira. O Pedro Sequeira, colega de turma de mestrado, seria o meu colega de estágio. Também em julho, eu e o Pedro Sequeira reunimos online com a Prof. Altina Ramos e recebemos as instruções iniciais. Ainda em julho, reunimos online com o Prof. José Pereira que mostrou-se disponível para voltar a reunir logo que soubesse as disciplinas que lecionaria em 2021-22.

Em fins de agosto/início de setembro, o Prof. José Pereira informou-nos que as áreas disponíveis eram a Internet das Coisas (IOT), numa disciplina do 10º ano, e Programação Android, numa disciplina do 11º ano. O Pedro Sequeira interessou-se por IOT. Da minha parte, eu tinha feito um trabalho sobre este assunto, em Metodologia do Ensino da Informática II (Morais et al., 2021), e sinto-me confortável neste conteúdo; então, escolhi Programação Android, com a qual só tinha tido algum contacto durante a licenciatura: considerei uma boa oportunidade para relembrar e preparar-me como professor.

3.2. A Escola

A Escola do Estágio recebeu obras de melhoramento há poucos anos, contando atualmente com boas instalações ao nível de salas de aula, cantina e prática de desporto, entre outras. A Escola dispõe de condições adequadas para a lecionação de Informática e possui um espaço físico dedicado ao Clube de Robótica. A sua biblioteca promove ações de tutoria e literacia na escrita, com ênfase em alunos do 10º e 12º anos. No dia do patrono da Escola, realizam-se anualmente atividades culturais e científicas, de participação livre. Atendendo à demanda da comunidade, esta Escola oferece cursos noturnos de ensino secundário. Nos cursos profissionais de ensino secundário, os alunos recebem subsídio de alimentação e transporte e bolsa de material de estudo. Aquando da formação em contexto de trabalho, é frequente os alunos dos cursos profissionais desta Escola optarem pela modalidade Erasmus, realizando estágios no exterior.

O ano letivo começou com um rastreio à COVID-19, de forma faseada, a toda a comunidade escolar, na sequência do parecer da Direção-Geral da Saúde (DGS): Fase 1, Pessoal Docente e Não Docente, de 6 a 17/9; Fase 2, Alunos do ensino secundário, 20/9 a 1/10; Fase 3, Alunos do 3º ciclo,

4 a 15/10 (República Portuguesa, 2022). O uso obrigatório da máscara nas escolas manteve-se para docentes, funcionários e alunos com mais de 10 anos (ECO, 2021).

Neste início de segundo ano letivo sob as restrições para a não propagação da COVID-19, apesar da fadiga coletiva, o impacto psicológico entre os jovens foi menor que no pico da pandemia, pois não foram privados de suas conexões com colegas para obter apoio emocional (Magson et al., 2021).

3.3.A Disciplina, os Horários e a Sala de Aula

A partir de finais de setembro, passei a assistir, presencial e regularmente, às aulas da disciplina de Programação Móvel (PM), lecionadas pelo Prof. José Pereira às terças-feiras, no 11º ano do CPPI (11ºTPI), de nº 481040 no Catálogo Nacional de Qualificações (ANQEP, 2022), que a Escola oferece desde 2015-2016. Com duração anual, a PM é constituída pela UFCD 0816, *Programação de sistemas na linguagem Java*, seguida da UFCD 0817, *Programação de sistemas web em Java*, cada qual com carga horária de 50 horas. A disciplina PM tem 5 aulas semanais.

Horários: no ano letivo 2021/22 a turma foi dividida em 2 turnos, T1 com 9 alunos e T2 com 8. O horário anual foi organizado à 3ª e 6ªf, de manhã, sendo a distribuição horária em 1 bloco de 90 +45 minutos e outro de 90 minutos, para cada turno: T2, 3ª feira 8h20 às 10h45 e 6ª feira 11h40 às 13h10; T1, 3ª feira 10h45 às 12h25 e 6ª feira 9h05 às 11h30.

A Figura 14 mostra o esquema da sala onde decorreram as aulas. O professor localizava-se de frente para a turma e o estagiário ocupava a secretária próxima à do professor. Na altura da intervenção pedagógica, professor e estagiário inverteram as posições.

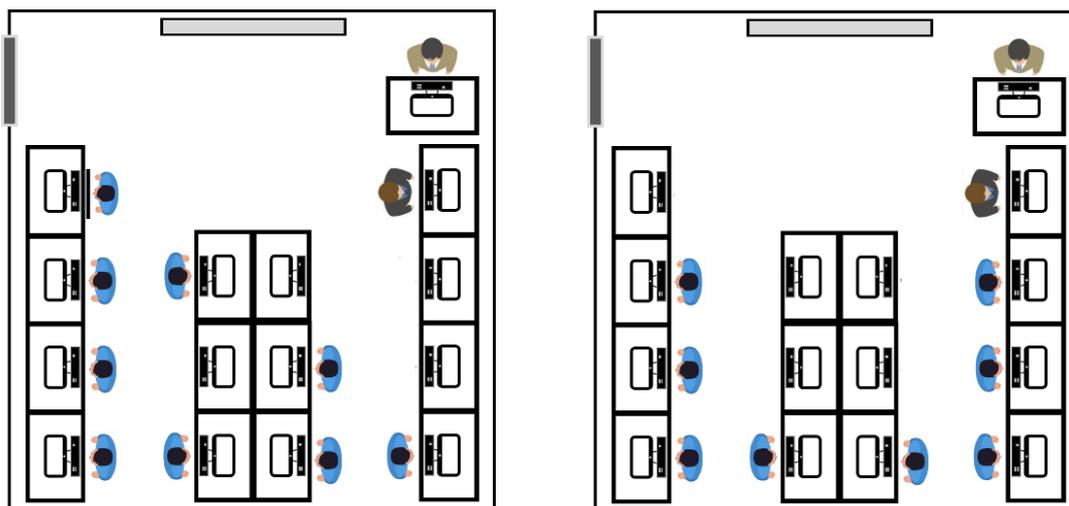


Figura 14. Distribuição da Sala de Aula 11ºTPI: Turno 1 (esquerda) e Turno 2 (direita)

3.4.A Turma

Os dados a seguir foram coletados pela escola, no início do ano letivo, e disponibilizados, a meu pedido, para a elaboração do *Projeto de Intervenção Pedagógica Supervisionada* (PIPS).

Em 2021-22, a turma do 11ºTPI foi formada por 18 rapazes de idades entre 15 e 18 anos, sendo 8 com 16 anos, 4 com 17, 4 com 15 e 2 com 18 anos. A turma tinha 15 nacionais portugueses e 3 brasileiros. Em linhas gerais, a situação sócio-económica da turma situava-se na classe média, 16 em 18 tinham internet em casa e 13 em 18 possuíam computador em casa. Os encarregados de educação, na maioria as mães (13 em 18), possuíam formação académica de nível secundário ou superior (11 em 18). A maior parte dos alunos indicou tomar o pequeno-almoço em casa (12 em 18) e estudar em casa (11 em 18), sugerindo baixos níveis de conflito familiar. Relativamente à saúde, 11 em 18 alunos responderam não ter problemas; os que mencionaram problemas, referiram asma (3 em 18), alergia (2 em 18), dificuldades visuais (1 em 18) e dificuldades motoras (1 em 18) - destes, 3 tomavam medicação regularmente. Sobre os hábitos de sono, 9 alunos indicaram dormir entre 7 e 8 horas, 5 referiram menos de 7 horas e 3 dormem entre 8 e 9 horas por noite. Por estarem no 11º ano, foi-lhes solicitada uma apreciação sobre as disciplinas que cursaram no 10º ano, do género “a tua disciplina favorita” e “a disciplina que menos gostaste”. *Engenharia de Software* foi a mais votada entre as favoritas (8 em 14), seguida de *Educação Física* (4 em 14). *Programação Móvel* recebeu mais votos entre as disciplinas que os alunos menos gostaram (5 em 15), seguida de *Português* (3 em 15) e *Área de Integração, Físico-Química, Inglês e Matemática*, cada uma com 2 votos.

A seguir, apresento as respostas da turma a perguntas sobre 9 temas: autoavaliação; situações favoráveis de aprendizagem; gosto pelo estudo; hábitos de estudo; perceção sobre insucesso escolar; atividades nos tempos livres; intenção de continuação dos estudos; profissão desejada; interação com os familiares sobre os assuntos escolares (Tabela 2).

Tabela 2. Excerto do Questionário da Escola, aplicado no início do ano 21/22

Temas	Perguntas e Respostas
Autoavaliação	1. Consideras-te um aluno...
	Médio: 14; Bom: 2; Fraco: 1
Situações favoráveis de aprendizagem	2. Em que situação aprendes melhor?
	Sozinho: 7; Nas aulas: 6; Em grupo: 4; Com um explicador: 0
Gosto pelo estudo	3. Gostas de estudar?
	Não: 11; Sim: 6
Hábitos de estudo	4. Quando estudas?
	Em véspera de teste: 9; Raramente: 5; Nunca: 3; Diariamente: 0
Perceção sobre insucesso escolar	5. O que contribui mais para o insucesso escolar dos alunos?
	Falta de estudo: 5; Falta de atenção/concentração: 5; Desinteresse pela disciplina: 4; Dificuldades em compreender o professor: 3; Indisciplina na sala de aula: 0; Conteúdos difíceis: 0
Atividades nos tempos livres	6. O que costumavas fazer nos tempos livres?
	Jogo no computador: 15; Vou à internet: 15; Encontro-me com amigos: 13 Oíço música: 13; Passeio: 8; Pratico desporto: 8; Vejo televisão: 5; Vou ao cinema: 4; Leio: 2
Intenção de continuação dos estudos	7. Até quando pensas estudar?
	Ensino superior: 11; 12ºano: 6;
Profissão desejada	8. Que profissão gostarias de ter?
	Programador: 5; Qualquer coisa envolvida com informática: 4; Engenheiro informático: 3; Não sei: 3; Mecânico: 1 Gostava de seguir box, ou algo ligado ao meu curso: 1
Interação com os familiares sobre os assuntos escolares	9. Em casa, falas sobre escola/estudo?
	Raramente: 12; Frequentemente: 4; Nunca: 1

Os alunos responderam às perguntas 1 a 6 baseados no tempo presente, enquanto as respostas à 7 e 8 indicam as suas expectativas sobre o futuro. Parece haver pouca reflexão sobre as consequências que as ações cotidianas podem ter na concretização das suas aspirações profissionais. O facto da maioria dos alunos raramente (12) ou nunca (1) conversar sobre a Escola com o seu Agregado Familiar/ Encarregado de Educação (pergunta 9), pode ser um dos factores que levam a esta inconsistência.

NOTA: Durante o primeiro período houve ligeiras alterações na composição da turma, designadamente saíram dois alunos e entrou uma aluna. A minha intervenção pedagógica, no segundo período, realizou-se com 17 alunos, dos quais 16 rapazes e 1 rapariga, sendo 13 nacionais portugueses e 4 brasileiros.

No período de setembro a março, em que estive a assistir às aulas, observei que o Professor Cooperante sabia implementar as tarefas do ensino na sala de aula (Figura 15), isto é, conseguia estabelecer a ordem coletiva de modo a promover a aprendizagem individual.

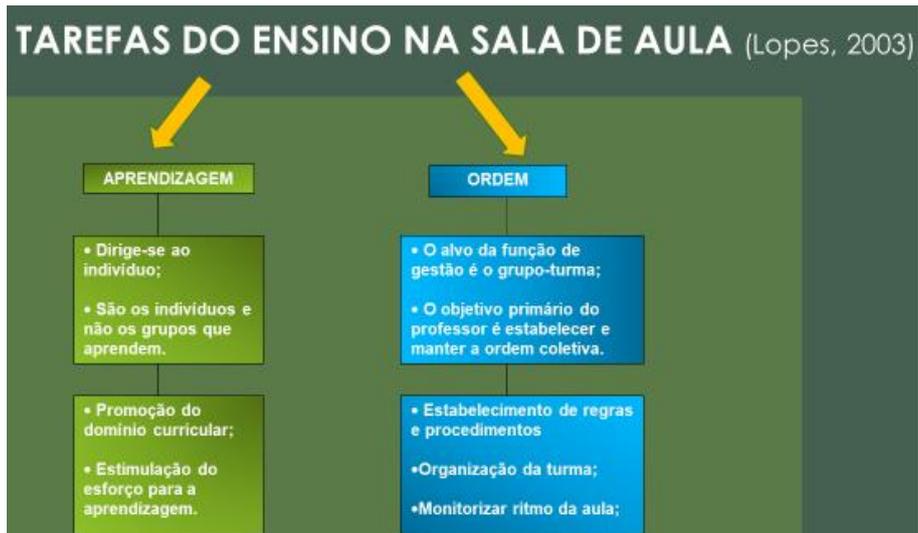


Figura 15. Tarefas do Ensino na Sala de Aula (Lopes 2003, citado por Caires, 2021)

3.5. Os Principais Acontecimentos que Impactaram a Escola

De acordo com o modelo de desenvolvimento humano de Bronfenbrenner (Figura 16), a escola não é um sistema isolado do resto do mundo; antes pelo contrário, os acontecimentos locais, nacionais e mundais, consoante a sua natureza e duração, influenciam as relações interpessoais na comunidade escolar e o ensino-aprendizagem. Tendo em conta um jovem que estivesse a frequentar o ensino secundário em Braga, no ano letivo 2021-22, considerei que os factores externos mais impactantes relacionaram-se com a pandemia de COVID-19 e com a guerra na Ucrânia.

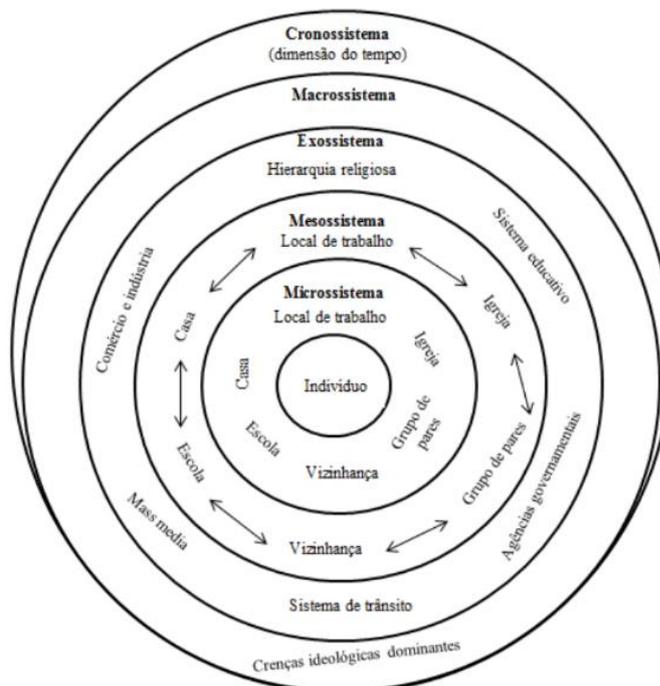


Figura 16. Modelo Ecológico de Bronfenbrenner do Desenv. Humano (citado por Dias, 2014, p.15)

Com vista a controlar a contaminação nas escolas após as férias de Natal e Ano Novo, o calendário escolar foi alterado e o regresso aconteceu a 10 de janeiro. Os cinco dias a menos de aulas foram compensados com menos dois dias nas férias de Carnaval e menos três dias na interrupção da Páscoa. Considero ter sido uma decisão correta, pois houve um grande aumento de casos de COVID entre 4 e 19 de janeiro de 2022, devido à nova variante Ómicron (Figura 17).

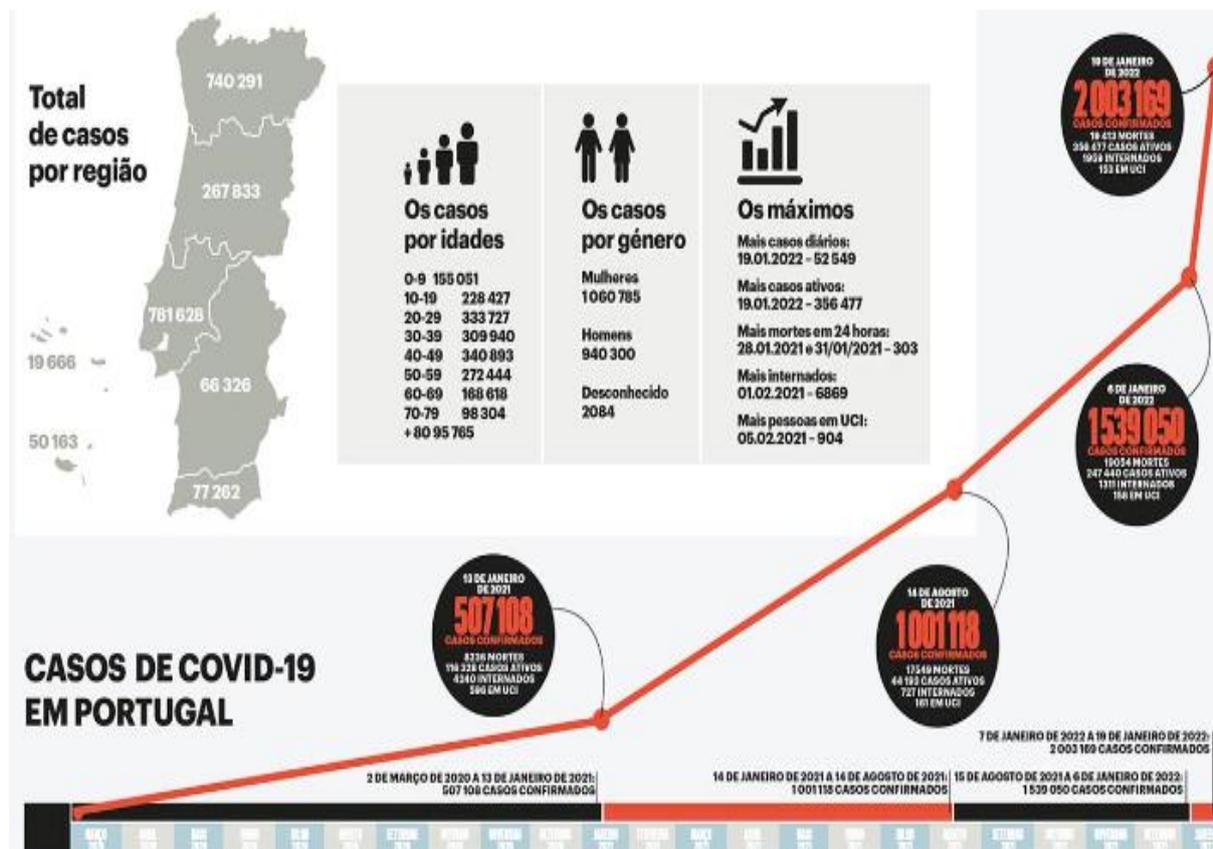


Figura 17: Casos COVID-19, 03/2020 a 01/2022 (adapt. Inácio & Sequeira, 2022)

Com o controlo da pandemia, a 21 de abril de 2022 terminou do uso obrigatório de máscara na maior parte dos espaços fechados, inclusive nas escolas (Godinho, 2022).

Ainda estava a comunidade escolar a usar máscara quando, em 24 de fevereiro, a RTP1 interrompeu a programação para anunciar a eclosão de uma guerra entre a Rússia e a Ucrânia (Portugal News, 2022). Nos dias que se seguiram, centenas de milhares de residentes na Ucrânia deixaram o país, em busca de refúgio; muitos vieram para Portugal. A 7 de março, Braga recebeu um autocarro com refugiados de guerra (RTP, 2022); inúmeras ações humanitárias e demonstrações de apoio foram realizadas (Figura 18).



Figura 18. Demonstração em Braga, abril de 2022

Não tenho conhecimento de estudos sobre os efeitos da guerra na Ucrânia na população adolescente das cidades portuguesas que receberam refugiados, mas é possível que tenham sido impactados. Num conflito armado, fortemente noticiado pelos media, mesmo as famílias e as comunidades não envolvidas diretamente são também perturbadas psicologicamente e muitos aspetos da vida normal são afetados, nomeadamente a produtividade do trabalho e da aprendizagem (Garry & Checchi, 2020). Até o momento em que escrevo este relatório, outubro de 2022, a guerra na Ucrânia continua.

4. Plano de Intervenção Pedagógica Supervisionada

4.1. Tema

O tema de ensino-aprendizagem (E-A) da intervenção foi a criação de programas para dispositivos móveis Android, doravante denominado “criação de Apps para Android”, inserido no módulo *Programação de Sistemas Web em Java*, da disciplina *Programação Móvel*.

A questão de investigação pedagógica (QIP) subjacente à intervenção foi:

“Como o pensamento computacional, e ferramentas a ele associadas, podem contribuir no E-A de Apps para Android?”

4.2. Objetivos

A nível geral, esta intervenção pedagógica visou contribuir para o desenvolvimento das competências associadas ao Perfil do Aluno à Saída da Escolaridade Obrigatória (PASEO) (Martins et al., 2017), através da promoção de um ambiente colaborativo nas atividades escolares, do estudo autónomo e da reflexão crítica, entre outras.

Os **objetivos específicos** que se pretenderam alcançar foram

- (i) Fomentar a motivação no estudo da programação
- (ii) Desenvolver a criação de Apps para Android
- (iii) Recolher e tratar dados relacionados com o pensamento computacional em programação
- (iv) Identificar as perceções dos alunos relativamente à qualidade da intervenção realizada e à sua aprendizagem.

4.3. Estratégias de Ação da Intervenção

Para acompanhar e registar o desenvolvimento das competências do PASEO durante a intervenção, elaborei a minha Grelha de Observação dos Alunos (GOA) com base em Cunha (2015), porque o seu modelo não só explicita e distingue as competências transversais (assiduidade, pontualidade, comportamento) das capacidades e atitudes (participação e empenho, demonstração de autonomia e iniciativa própria), como também apresenta indicadores coerentes para o seu preenchimento. As adaptações que fiz foram as seguintes:

- Agrupei os tópicos assiduidade e pontualidade, porque a plataforma Nónio permite distinguir a presença normal (P) da presença atrasada/falta de pontualidade (FP).

- Separei em dois o tópico “participação e empenho”, porque considerei que um aluno que faz perguntas é um aluno participativo, mas pode acontecer de fazer perguntas apenas por ter preguiça de

pensar, então, não é empenhado. Considerei que “empenho” assemelha-se a resiliência: o aluno continua a tentar resolver o problema mesmo quando não tem resultados favoráveis imediatos.

- Quanto ao tópico “demonstração de autonomia e iniciativa própria”, também me parece que são capacidades diferentes pois o aluno pode adquirir autonomia para realizar os trabalhos solicitados, sem apoio do professor nem dos colegas, ao longo das aulas, enquanto “iniciativa própria” é uma capacidade mais relacionada com a motivação, ou seja, o aluno está motivado e inicia um trabalho sem que o professor lho tenha solicitado. Assim, mantive “autonomia” e retirei “iniciativa própria” pois considerei que, no caso de observar uma situação deste tipo, a mesma deveria ser descrita à parte, com algum detalhe. De facto, ao longo da minha intervenção, só por uma vez presenciei um aluno a ter a iniciativa própria de tentar criar um jogo de computador, mas estava no seu computador pessoal, fora do contexto da aula.

Com isso, a GOA compôs-se de: Assiduidade e Pontualidade, Comportamento, Participação, Empenho, e Autonomia. Para o seu preenchimento, adotei os seguintes indicadores:

- Assiduidade e Pontualidade: Presente = P; Faltou = F; Falta de Pontualidade = FP; Falta Justificada = FJ; Falta injustificada = FI.
 - Comportamento: Adequado = A; Inadequado = I; Não consegui observar = ?.
 - Participação, Empenho, Autonomia: Sim =S; Não =N; Não se Aplica =N/A; Não consegui observar=?
- A GOA final consta no apêndice III.

Como referência para estimular a motivação no estudo da programação, objetivo (i), adotei o modelo ARCS (Atenção – Relevância – Confiança - Satisfação), de John Keller (1987; Moura & Carvalho, 2010). Tendo em conta as suas tabelas com sugestões de ação para as 4 estratégias, selecionei as propostas que considerei compatíveis com o meu modo de ser e que estavam ao meu alcance desempenhar (Tabela 22).

Para implementar a criação de Apps para Android, objetivo (ii), adotei a metodologia de Aprendizagem Baseada em Caso (CBL) (Merseth, 1991; Koslosky, 1999; Wijnia et al., 2017; Pereira, P. 2021). Os conteúdos teórico-práticos da aprendizagem basearam-se no manual online oficial da Google, *Codelabs for Android Developer Fundamentals*, o qual contém instruções passo a passo para a programação Android, na linguagem de programação Java e na linguagem de notação XML, utilizando o Android Studio. Desde setembro de 2022, está disponível a versão 2 do manual (Google Developers, 2022b).

O plano previa a criação de três Apps, selecionadas em conjunto com o Professor Cooperante. A escolha das Apps deveu-se a terem um nível de complexidade adequado para o 11º ano e serem enquadráveis na época do ano letivo prevista para a minha intervenção (2º período, após as férias de

carnaval). O facto de os alunos terem mencionado gosto e interesse pelas aulas de educação física, no questionário do início do ano, levou à suposição que a conversão para App Android dos programas Índice de Massa Corporal (IMC) e Índice Cintura-Quadril (ICQ), previamente estudados em Java, na unidade de formação de curta duração (UFCD) 0816, poderia ser bem recebida e despertar-lhes entusiasmo. A terceira App, Contato, seria precedida pela introdução dos conceitos sobre Atividades e Intentos, lecionados a seguir à segunda App. Na previsão inicial, a intervenção de 15 aulas em cada turno foi pensada da seguinte forma: 4 aulas por App (3 Apps = 12 aulas) + 1 aula para os conceitos de Atividades e Intentos + 2 aulas para as 2 aplicações do CTt. As aulas foram planeadas para terem uma parte inicial expositiva, em que o professor mostra o aspeto e as funcionalidades da App terminada. A seguir, o professor mostra as partes do código no IDE para completar a App, seguindo a lógica do pensamento computacional (decomposição, reconhecimento de padrões, abstração, algoritmia, modelação & simulação e avaliação). A exposição é intercalada com perguntas. A cada passo (decomposição), o professor faz menção a um problema que os alunos conhecem, seja da aula passada ou do período passado, e interroga sobre como o problema foi resolvido (reconhecimento de padrões). De seguida, o professor mostra a solução para o caso em questão (abstração, algoritmia, modelação & simulação e avaliação). As aulas estão descritas no tópico 5.1.

A estratégia para o objetivo (iii), recolher e tratar dados relacionados com o pensamento computacional em programação, foi a aplicação do teste de pensamento computacional de Román-González (2016), doravante designado como CTt, que adaptei à linguagem Java (Apêndice I). Para isso, solicitei e recebi autorização da Professora Supervisora. Os passos seguintes relacionaram-se com a autorização da Direção Geral da Educação (DGE), setor de Monitorização de Inquéritos em Meio Escolar (MIME). A submissão à DGE/MIME, através do site <http://mime.dgeec.mec.pt/> constou de um formulário e de anexos com o parecer da Professora Supervisora do Projeto (Anexo D), a autorização dos autores do teste/questionário original e da respetiva tradução (Anexos B e C) e o modelo da Declaração de Consentimento Informado, Livre e Esclarecido (Apêndice II). O pedido associado recebeu o n.º0807900001. Após a autorização (Anexo E), foi necessário dar o devido conhecimento ao Diretor da Escola e enviar aos Encarregados de Educação duas vias da Declaração de Consentimento Informado, Livre e Esclarecido, solicitando a devolução de uma via assinada pelo Encarregado e pelo Aluno. O CTt foi aplicado nos dias 15/3 e 1/4; a descrição da sua aplicação na intervenção consta nos tópicos 5.1.1., 5.1.2. e 5.1.10.

Para o objetivo (iv), identificar as perceções dos alunos relativamente à qualidade da intervenção realizada e à sua aprendizagem, a estratégia consistiu em quatro iniciativas, nomeadamente a aplicação de dois inquéritos em que, no primeiro, avaliaram sobretudo o professor e a intervenção e, no segundo,

autoavaliaram-se (Apêndices IV e IX respetivamente), e a realização de duas reuniões/entrevistas individuais (Apêndices VI e VIII). Os procedimentos sobre o objetivo (iv) estão descritos no texto introdutório do capítulo 5, intervenção não sumariada, e no final do tópico 5.1.10.

Para a QIP, *Como o pensamento computacional, e ferramentas a ele associadas, podem contribuir no E-A de Apps para Android?*, a estratégia de ação foi a consulta aos artigos de Marcos Román-González e Sue Sentance, por investigarem o pensamento computacional na escolaridade obrigatória, além de uma consulta intensiva ao *Google Scholar*, auxiliada pelo software *Publish or Perish* (Harzing, 2007), com uma grande variedade de palavras-chave, e suas combinações, em que as 5 mais frequentes foram *computational thinking skills*, *develop mobile apps*, *computer science education*, *code learning*, *vocational secondary school*, e respetivas correspondentes em português.

4.4. Estratégias de Avaliação da Intervenção

Os elementos utilizados na avaliação formativa foram (i) a GOA, (ii) a aplicação do CTt, (iii) as reuniões/entrevistas individuais e (iv) a autoavaliação dos alunos.

De referir que o PIPS previa a realização de entrevistas *focus group*. No entanto, no decorrer das aulas, pareceu-me que os resultados do CTt e das Apps IMC e ICQ deviam ser apresentados individualmente aos alunos, de modo que os pudessemos comentar livremente, sem constrangimentos e com respeito à privacidade, tal como recomendado nos tópicos S3.2 e S4.3, por Keller (1987). Assim, os encontros dos dias 19 e 26/4 tiveram um caráter duplo de reunião e entrevista curta: reunião porque mostrei-lhes os seus resultados (Apêndices V e VII), esclareci dúvidas e mostrei os gráficos dos resultados globais da turma; entrevista curta porque, no tempo restante, fiz perguntas mais ou menos padronizadas e gravei as suas respostas (Apêndices VI e VIII).

A avaliação sumativa correspondeu à correção dos trabalhos individuais sobre as Apps IMC e ICQ, enviados para o meu email institucional. Os critérios de correção valorizaram os acertos das tarefas que envolviam funções (denominadas “métodos”, em Java) porque, pela aplicação e correção do CTt no primeiro dia da minha intervenção, pude verificar que a maior parte dos erros da turma situava-se nas questões finais do teste, nomeadamente nos tópicos com condicional *while* (21 a 24) e com funções (25 a 28). Diante disso, procurei reforçar a lecionação dos métodos Java durante a intervenção (Ferreira, 2021, p.10). No tópico 5.2, apresento os resultados e respetiva análise. Os códigos-fonte que fiz para as Apps constam nos apêndices X e XI.

A estratégia de validação da QIP, no contexto específico da intervenção, baseou-se na dupla aplicação do CTt, sendo a interpretação dos resultados apoiada pelas observações e reflexões de autores

que o aplicaram em condições semelhantes, nomeadamente Guggemos et al.(2022), Chan et al.(2020) e Román-González (2016). Resultados e análise no tópico 5.2.

4.5. Justificação das Estratégias da Intervenção

A motivação é considerada um importante factor não cognitivo que afeta o desempenho académico dos alunos. Não há controvérsia sobre os benefícios da motivação intrínseca; em contrapartida, o reconhecimento da importância da motivação extrínseca é mais recente (Yunlong & LIU, 2022). Com vista a auxiliar os professores em táticas motivacionais, Keller (1987) desenvolveu o modelo ARCS que consiste em (i) atrair e manter a Atenção dos alunos, (ii) demonstrar a Relevância do conteúdo, ou ensinar o conteúdo de modo Relevante, (iii) gerar Confiança, transmitindo aos alunos uma expectativa positiva sobre os seus resultados de aprendizagem e (iv) recompensar os seus esforços, de modo a produzir Satisfação.

O desenvolvimento de Apps para dispositivos móveis, através do IDE Android Studio, é uma unidade curricular lecionada atualmente em Institutos politécnicos, por exemplo nos Institutos Politécnicos de Lisboa (ISEL), Tomar (IPT), e Cávado e Ave (IPCA). Esta técnica possui elevados graus de abstração e complexidade. Com efeito, o Android Studio é a ferramenta mais completa e atualizada para programação Android (Google Developers, 2022a). Contudo, o facto de um tema ser lecionado no ensino superior não impede que partes introdutórias do seu conteúdo sejam lecionadas previamente, na escolaridade obrigatória. A expansão da programação informática a níveis cada vez mais iniciais do ensino formal tem sido defendida por muitos autores e posta em prática por vários países. Através do artigo de Armoni (2014), a propósito do alargamento do ensino de informática à toda a escolaridade obrigatória em Israel, conheci o postulado que Bruner escreveu no início do capítulo 3 do seu livro *The Process of Education*, de 1960: “*We begin with the hypothesis that any subject can be taught effectively in some intellectually honest form to any child at any stage of development*”. Ou seja, segundo Bruner, em princípio, a faixa etária do aprendente não é causa de limitação de um dado conteúdo de aprendizagem, uma vez que, pela teoria da espiral da aprendizagem (de sua autoria), os níveis de complexidade dos conteúdos vão-se desenvolvendo ao longo do tempo (Figura 19).

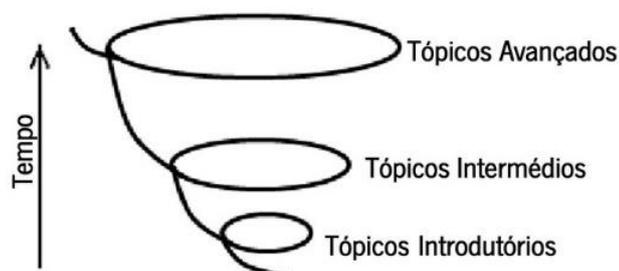


Figura 19. Espiral de Aprendizagem de Bruner (adaptado de Vemuru et al., 2013)

Assim, para além da não limitação *a priori*, a utilização do Android Studio pode contribuir para as boas práticas de informática (Quadros-Flores, 2011) e funcionar como um factor de motivação extrínseca, pois os alunos do ensino profissional sabem que existe a UC mais aprofundada no ensino superior. Segundo o modelo ARCS, trata-se de uma motivação do tipo “Relevância - Utilidade futura - R3.1 - Declarar explicitamente como a instrução se relaciona com as atividades futuras dos alunos” (Keller, 1987).

Do ponto de vista da lógica da sequência dos conteúdos curriculares, pode dizer-se que a programação para Android é lecionada em PM porque (i) a arquitetura do Android está intimamente ligada à linguagem Java, e aos seus recursos de desenvolvimento de software, tendo os alunos estudado Java no 1º período. Do mesmo modo, o Android Studio é o IDE adotado por ter sido desenvolvido a partir do IDE IntelliJ IDEA (Toporov, 2013), o qual os alunos utilizaram no 1º período, representando uma oportunidade de aplicação/consolidação de conhecimento.

No que respeita às metodologias de E-A, Freeman e co-autores (2014) publicaram uma meta-análise de 225 estudos, sobre alunos da área de ciências, com comparações entre os resultados de exames dos que tiveram formação em metodologia tradicional *versus* os resultados com formação em metodologias ativas. A comparação estatística mostrou que, no ensino superior, há diferenças significativas nos resultados, consoante a metodologia de ensino, sendo muito favorável às metodologias ativas. Contudo, no mesmo artigo, Freeman destacou que, na escolaridade obrigatória (K-12), a diferença a favor das metodologias ativas não foi tão significativa e recomendou uma melhor avaliação para alunos mais jovens.

Em 2021, Brod lembrou que o tipo de interação aluno-professor deve ter em consideração o estágio cognitivo do aluno, de modo que, nos estádios mais iniciais da aprendizagem, a intervenção do professor deve ser no sentido de garantir que o aluno tenha acesso aos conhecimentos fundamentais de um dado conteúdo de uma forma clara e por um período suficiente para o interiorizar. A Figura 20, adaptada por Brod, ilustra as muitas variáveis envolvidas numa metodologia ativa para aprendizagem de ciências. As metodologias ativas pressupõem que o aluno já possui, ou é capaz de desenvolver durante o processo, os principais pré-requisitos de sucesso da aprendizagem, nomeadamente o interesse pessoal, o conhecimento anterior, as capacidades cognitivas e a capacidade de monitorizar e autoregular os processos cognitivos. Num contexto educacional formal de ensino de ciências, a aprendizagem envolve a familiarização com as especificidades do setor em causa (por exemplo, o vocabulário), a manipulação matemática de dados, a simulação de modelos e a realização de experiências práticas.

As componentes das metodologias ativas são o próprio aluno, os seus pares e o professor; a qualidade e a quantidade de interação entre eles varia consoante a metodologia adotada.

Construção de um Eco-sistema de Compreensão: Aprendizagem Ativa com Pré-requisitos

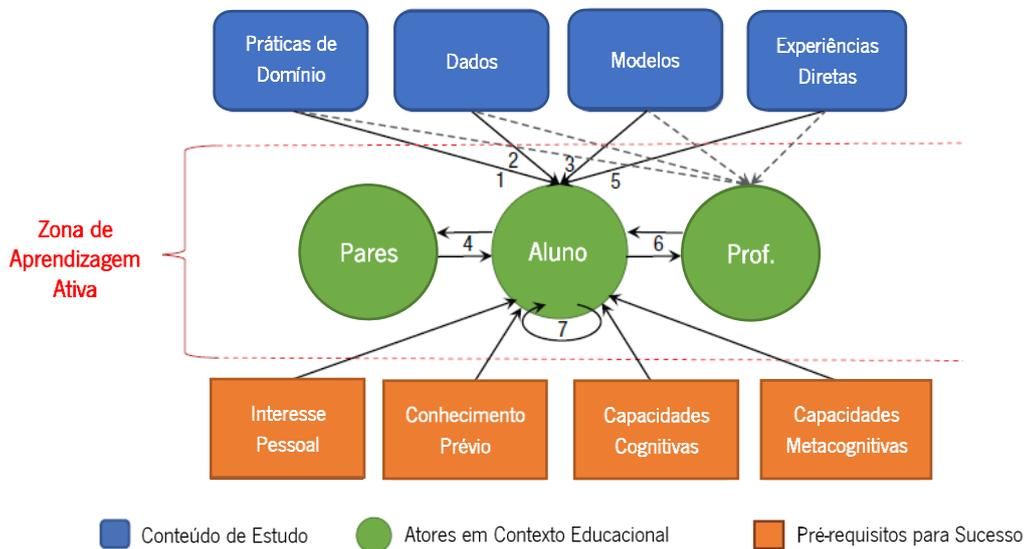


Figura 20. Pré-requisitos do Aprendiz para Beneficiar das Metodologias Ativas, num Contexto Educacional Formal de Ensino de Ciências (adapt. Brod, 2021)

Em resumo, a escolha da metodologia de ensino de um conteúdo deve ter em conta a idade e o contexto sociológico do aprendente e ser adequada ao seu estágio de autonomia e desenvolvimento metacognitivo. Em muitos casos, metodologias com maior intervenção do professor podem ser adequadas. McLean (2016) refere que a metodologia de Aprendizagem Baseada em Caso (Case-Based Learning, CBL) é uma ferramenta de ensino-aprendizagem usada na área da saúde, para dar ajudar na conexão da teoria à prática. Nos pressupostos da CBL, algum conhecimento básico é transmitido antes do período de aprendizagem - ao contrário da metodologia de Aprendizagem Baseada em Problema (Problem-Based Learning, PBL) em que muitas vezes a informação é pesquisada durante o período de aprendizagem, podendo não haver informação prévia (Wijnia et al., 2017). Deste ponto de vista, para alunos do ensino secundário, a CBL pode ser mais apropriada que a PBL.

No que respeita ao pensamento computacional na escolaridade obrigatória, e à utilização de ferramentas a ele associadas, a revisão de Bocconi et al.(2022), que abrangeu 30 países, relata 12 países que inserem o pensamento computacional ao nível da Educação e Formação Profissional Inicial (EFPI), ou seja, até o 3º ciclo, e 15 países onde é inserido ao nível do ensino secundário profissional (EFP). Em relação a Portugal, os autores mencionaram a realização de entrevista com os representantes oficiais, mas não incluem o país na lista dos que lecionam o pensamento computacional no ensino profissional (Bocconi et al., 2022, item 4.6, fig. 8).

A relevância da estratégia adotada na QIP pode ser demonstrada através de autores que tenham relatado resultados favoráveis em estratégias semelhantes. No seu artigo de revisão, Butler (2018) cita vários autores que consideram a realização de um teste não só como uma ferramenta de avaliação mas também como uma atividade que pode promover aprendizagem. Em relação à avaliação da aprendizagem cognitiva, Iz e Fok (2007) apresentaram exemplos de testes de escolha múltipla até ao nível Analisar, da taxonomia de Bloom (Figura 21), para conteúdos de geomática. Comparativamente, parece-me razoável supor que o teste de Román-González, adaptado para Java, também atinge o nível Analisar, nas questões finais do teste (21-28).

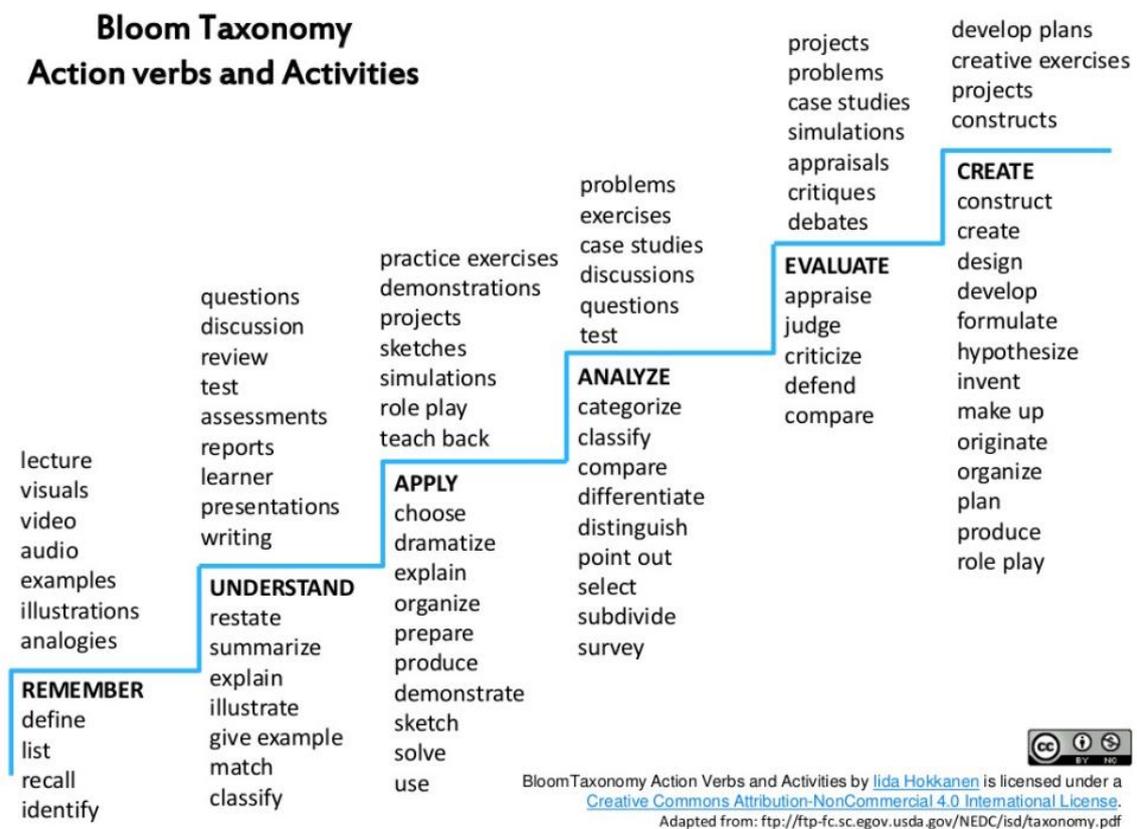


Figura 21. Verbos e Atividades segundo a Taxonomia de Bloom Revisada (Hokkanen, 2015)

A principal autocritica de Román-González à estrutura do seu teste CTt é o facto de ser focado em “conceitos computacionais” e apenas apresentar problemas lógicos e visual-espaciais, com labirintos e desenhos geométricos, ignorando outros aspetos do pensamento computacional, como, por exemplo, a narrativa interativa digital (Brennan & Resnick, 2012; Burke, 2012; Howland & Good, 2015; referidos por Román-González et al., 2017). No contexto da intervenção pedagógica descrita no presente relatório, a fragilidade do CTt apontada por Román-González foi, na verdade, uma mais-valia, pois os respondentes eram alunos de um curso profissional de programação informática e o que se pretendia verificar eram os seus estádios de conhecimento de conceitos de programação.

5. Intervenção Pedagógica

Previamente ao início das aulas de Android, o Professor Cooperante e eu procedemos à instalação do ambiente de desenvolvimento integrado (IDE) Android Studio, versão Bumblebee 2022, nos 14 computadores da sala de aula, e também realizámos testes de envio de questionários Google Forms.

A Intervenção Pedagógica, com lecionação sumariada, decorreu no período de 15 de março a 1 de abril de 2022, à 3ª e à 6ª feira, ininterruptamente, para os turnos 1 e 2 do 11º TPI, na sala e horário habitual da disciplina de Programação Móvel (PM), perfazendo 30 aulas (Tabela 3).

Tabela 3. Calendário da Intervenção Pedagógica Sumariada

15 março (3ªf)	18 março (6ªf)	22 março (3ªf)	25 março (6ªf)	29 março (3ªf)	1 abril (6ªf)
Sessão 1 – T2 8h20 – 10h45	Sessão 3 – T1 9h05 – 11h30	Sessão 5 - T2 8h20 – 10h45	Sessão 7 - T1 9h05 – 11h30	Sessão 9 - T2 8h20 – 10h45	Sessão 11 - T1 9h05 – 11h30
Sessão 2 – T1 10h45 – 12h25	Sessão 4 - T2 11h40 – 13h10	Sessão 6 - T1 10h45 – 12h25	Sessão 8 - T2 11h40 – 13h10	Sessão 10 - T1 10h45 – 12h25	Sessão 12 - T2 11h40 – 13h10

Na planificação original, seriam lecionadas 3 Apps. A intervenção de 15 aulas por turno previa 4 aulas por App, num total de 12 aulas, + 2 aulas para as 2 aplicações do CTt. + 1 aula para os conceitos de Atividades e Intentos. Na prática, sucedeu que as 2 primeiras Apps consumiram 12,5 aulas. Ao final da segunda App, reuni com o Prof. Cooperante e reorganizámos a planificação. Ficou decidido dividir a terceira App (Contacto) em três Apps menores, cada uma focada numa funcionalidade de Intentos, envolvendo duas *Activities*. Estas três Apps seguiriam uma estrutura muito parecida com a da Lição 2 do *CodeLabs Android Fundamentals*, na qual o exercício chama-se “*Two Activities*”. A primeira App foi denominada ‘DuasAtividades_LevarDados’; a segunda ‘DuasAtividades_TrazerDados’; e a terceira “DuasAtividades_LevarTrazerDados’. No meu tempo restante, eu lecionaria a primeira App, e as duas seguintes seriam lecionadas pelo Prof. José Pereira. Assim fizemos e correu conforme o planeado.

Mostram-se na tabela 4, os conteúdos principais da planificação ensino-aprendizagem efetivamente realizados na intervenção, sendo S1 a S12 as sessões de lecionação. Denominei “sessão” ao conjunto de aulas lecionadas a um turno no mesmo dia.

Tabela 4. Planificação dos Objetivos das Sessões

Objetivo de Lecionação	Turno 1	Turno 2
Demonstrar o funcionamento da App IMC	S2	S1
Efetuar a ligação da tela com o código	S2	S1
Programar o botão Limpar	S2	S1
Programar o botão Calcular sem a classe IMC	S3	S1
Programar o botão Calcular com a classe IMC	S3	S4
Prevenir o botão Calcular contra exceções nulas	S3	S4
Programar o botão Calcular com o método Classifica	S3	S4
Demonstrar o funcionamento da App ICQ	S6	S5
Construir as telas da App ICQ	S6	S5
Definir os tipos de valores de entrada	S7	S8
Efetuar a ligação dos botões Limpar e Calcular	S7	S8
Programar o botão Calcular com a classe ICQ	S7	S8
Programar o botão Calcular com o método Classifica	S10	S9
Demonstrar o funcionamento da App DuasAtividades_Levar	S11	S12
Programar o botão Enviar Sem Dados	S11	S12
Programar o botão Enviar Com Dados	S11	S12

A descrição das sessões começa com a tabela de planificação, a qual inclui, na escala temporal da aula, os objetivos, conteúdos, metodologias e atividades, e tipo de avaliação. A seguir à tabela vem o sumário, registado na plataforma Nónio da Escola. A seguir ao sumário, inicio a descrição da aula, com ênfase nos conteúdos de lecionação e na interação com os alunos.

Para além das aulas sumariadas, nos dias 19 e 26 de abril, durante o horário normal da aula de PM, 8h20 às 12h25, estive numa outra sala a reunir com os alunos, individualmente, por cerca de 10 minutos cada um. As sessões foram gravadas. Ainda em 26/04, à saída de cada tutoria/reunião/entrevista, entreguei aos alunos o inquérito nominal, em papel, *Questionário de Autoavaliação do Aluno* (Apêndice IX), que foi preenchido na sala de aula de PM e recolhido ao final da respetiva aula. Em 19/04 abordei com os alunos o teste CTt. Para isso, compus previamente uma grelha com os resultados individuais (Apêndice V). Expliquei os conceitos computacionais envolvidos nas questões do teste, com ênfase nos que tiveram mais erros. A seguir, perguntei-lhes a sua opinião sobre o teste. O apêndice VI contém os excertos das opiniões dos 14 alunos que compareceram. Em 26/04, abordei os trabalhos dos alunos sobre as Apps IMC e ICQ. Tal como para o CTt, construí uma grelha individual de avaliação (Apêndice VII) e, com ela, mostrei aos alunos os critérios que tiveram menos acertos e voltei a explicá-los. Os excertos dos comentários dos 16 alunos que compareceram constam no apêndice VIII.

Os resultados e análise da intervenção estão apresentados no tópico 5.2. A minha reflexão sobre a intervenção consta no tópico 5.3.

5.1. Aulas

5.1.1. Sessão 1 – terça-feira 15 de março 2022, Turno 2

Tabela 5. Planificação da Sessão 1

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento e informar os objetivos da aula	- Chamada de presença - Sumário da aula	Interrogativa e Expositiva.	45 min	Formativa
Recolher dados sobre raciocínio lógico e visão espacial dos alunos	- Explicação breve sobre o Pensamento Computacional - 1ª Aplicação do Teste CTt	Teste de Escolha Múltipla através do Google Forms		Sumativa
Demonstrar a App IMC terminada	- Revisão de conceitos anteriores - Elementos View - Funcionamento dos botões	Ativa. Aula prática demonstrativa e interrogativa. O professor mostra a App finalizada; cria uma sequência de passos para programar uma tarefa, e os alunos são solicitados a seguir esses passos para tarefas semelhantes. O professor circula pela sala, esclarece dúvidas e observa o progresso dos alunos.	10 min	Formativa
Ligação da tela com o código	- Atributo onClick - Manipuladores de Eventos - Teste Logcat		15 min	
Programar o botão Limpar	- Objetos Widget - Ligação código-tela findViewById - Método setText		20 min	
	Intervalo (9h50 – 10h00)		10 min	
Programar o botão Calcular sem a classe IMC	- Método getText - Conversão de Strings para Doubles - Conversão de Doubles para Strings	Ativa. Aula prática demonstrativa e interrogativa. (como antes do intervalo).	45 min	Formativa

Sumário: 1º Teste Pensamento Computacional. Início da App IMC

No primeiro dia, voltei a apresentar-me ao turno 2 (já o tinha feito meses antes) e disse que iria passar a dar aulas de Android durante as próximas três semanas.

Comecei com o teste de Pensamento Computacional. Após uma breve explicação sobre este conceito, informei que o teste seria aplicado duas vezes, ao início e ao final da minha intervenção. Informei que ao final da lecionação iria entrevistar os alunos, acerca dos resultados dos testes CTt e dos trabalhos. Enviei um convite por email aos alunos, para acederem ao questionário no Google Forms (Figura 22).



Pensamento Computacional

Este teste é composto por 28 questões.
Todas as questões têm 4 opções de resposta (A, B, C, D) das quais apenas uma é a correta.
Não é necessário responder a todas as questões.
Por favor, responda de acordo com o seu melhor entendimento, dentro do tempo determinado pelo professor.

Email *

O seu email

Figura 22. Cabeçalho do Teste de Pensamento Computacional

Um dos alunos notou que a questão nº10 tinha um erro de escrita: a opção certa (B) estava repetida na opção C. Pedi desculpas pelo erro e revelei a resposta certa a todos, desqualificando a questão. Um aluno chamou-me para tirar uma dúvida na questão nº3. Um aluno, com problemas psicológicos, terminou o teste em 10 minutos e saiu da sala; respondeu às questões sem qualquer cuidado.

O teste fechou automaticamente após 40 minutos, e todos conseguiram terminar dentro do tempo.

Iniciei a aula com uma revisão da App das aulas anteriores, *Hello Toast* - a continuação da App *Hello World* (tradução disponível no Anexo G) - na qual os alunos adicionaram um botão *Toast* e um botão *Count*. Tal revisão serviu para relembrar como aprenderam a criar rótulos, botões, e a fazer a sua ligação tela-código. Expliquei que o próximo passo seria adicionar caixas de texto – os *EditTexts*. Para isso, iriam criar uma App sobre um programa Java que fizeram no período anterior: o IMC. Isto foi a

Metodologia Baseada em Casos em ação, a reforçar as aprendizagens prévias adaptadas a novas situações.

Mostrei a App IMC terminada: abri o projeto no Android Studio, e expliquei o seu funcionamento no emulador (Figura 23).

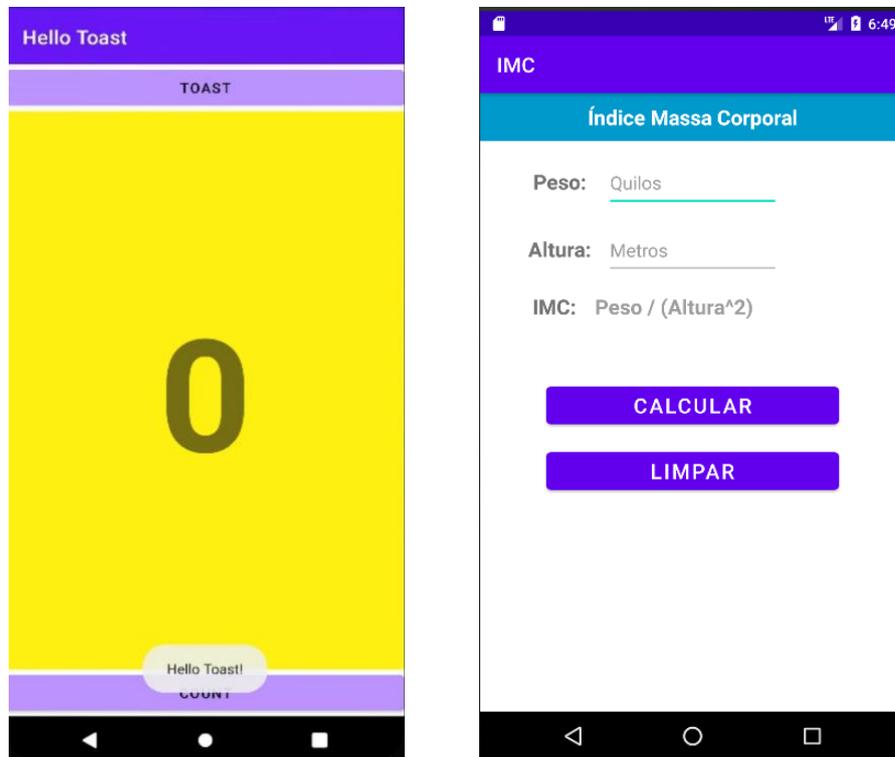


Figura 23. Apps *Hello Toast* (esquerda) e *IMC* (direita)

A App IMC consiste de dois botões, duas caixas de texto e dois rótulos (no total são seis rótulos, mas apenas dois importam para o código). O botão Calcular recebe os valores das caixas de texto e mostra os resultados IMC e a classificação de obesidade. O botão Limpar apaga todos os valores dos quatro elementos de texto. O botão Calcular deve também avisar se há caixas de texto vazias.

Pedi aos alunos para acederem à página do Nónio, a plataforma online oficial da escola (Figura 24) e baixarem o ficheiro *zip* no tópico '1.C', o qual continha o projeto com a tela da App IMC pré-feita. O foco deste projeto foi programar a App em código Java. A construção da tela no código XML ficaria para a próxima App, o ICQ.

Comecei com a ligação tela-código, isto é, associar elementos *View* do XML ao código Java. Nesta App, este tipo ligação só existe para os elementos *Button*, através do atributo *onClick*.

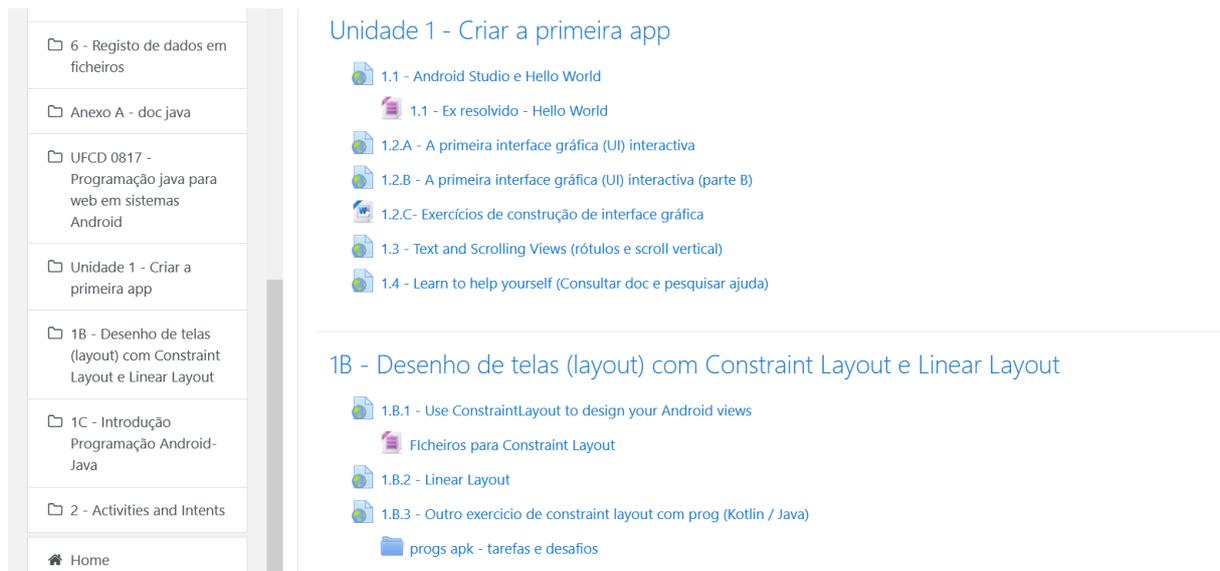


Figura 24. Excerto da Página da Disciplina na Plataforma Nónio

No ficheiro de layout *activity_main*, com a interface dividida entre código e *Design* no modo *Split*, dei instruções para adicionar o atributo *onClick* ao botão Limpar

Expliquei que o *onClick* está depreciado, mas que ainda pode ser utilizado. Este cria um manipulador de eventos (*Event Handler*) para o botão Limpar, que é um método Java específico para Apps Android. O manipulador tem como argumento um objeto *View*, que está associado ao elemento *View* correspondente no XML. O Android Studio abriu automaticamente o separador *MainActivity.java*.

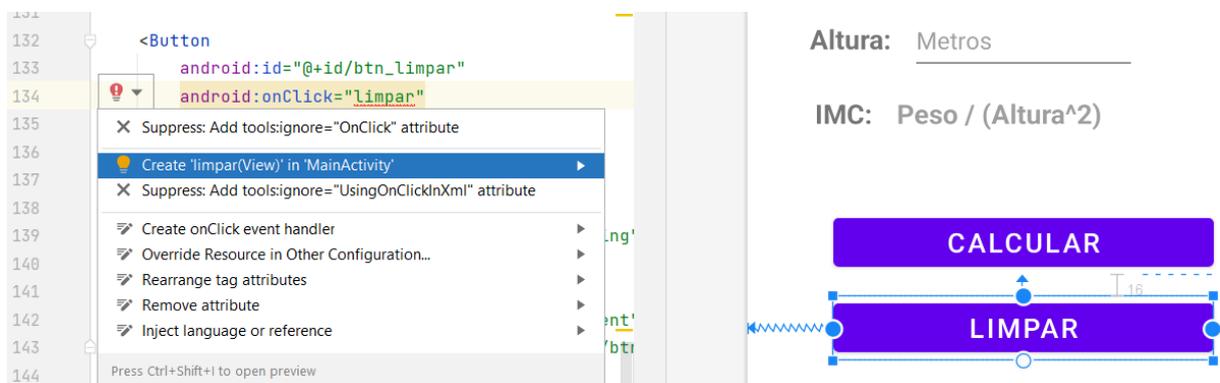


Figura 25. Atributo *onClick* e Manipulador de Eventos do Botão Limpar

Avisei que, antes de começarmos a programar, devíamos testar o *onClick* com o *Logcat* para certificar que está a funcionar. Criei um *Log* com uma mensagem "onClick funciona!" (Figura 26).

```

52 // manipulador de clique para botao limpar
53 public void limpar(View view) {
54     // testar o botão com Logcat
55     Log.i( tag: "btn_limpar", msg: "onClick funciona!");

```

Figura 26. Teste *Logcat* do Manipulador de Eventos Limpar

Corri a App no emulador e toquei no botão Limpar; no separador *Logcat*, surgiu uma mensagem de informação (*Info*), provando que a ligação tela-código funcionou (Figura 27).

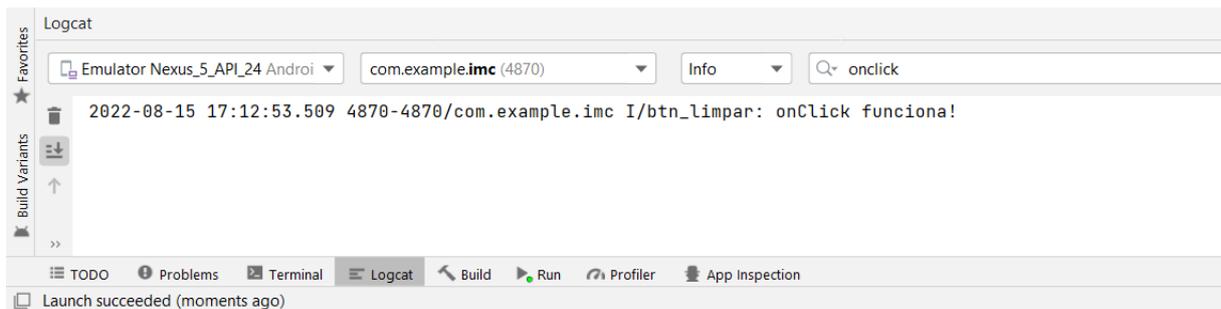


Figura 27. Mensagem *Logcat* do Botão Limpar

Passei à ligação código-tela. Expliquei que, para o código Java poder trabalhar com as telas Android, é preciso que o Java reconheça os elementos da tela – os *Views*, temos de especificar uma ligação entre os objetos Java e os elementos *View* do XML, os quais são reconhecidos através de uma chave identificadora - o *ID*.

No ficheiro *activity_main*, indiquei os atributos *ID*, e pedi aos alunos que renomeassem os *IDs* dos elementos usando uma convenção fácil de os identificar. Avisei que, sempre que fossem renomear *IDs*, que o fizessem através do comando *Refactor > Rename* (Figura 28).

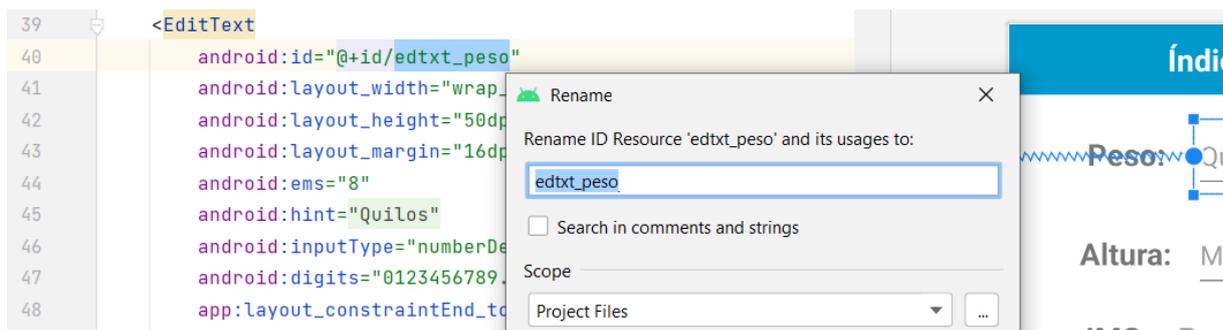


Figura 28. Renomeação dos IDs dos Elementos *View*

Avisei que o nome dos elementos *View* deve ser diferente dos objetos Java; enquanto o Java deve usar *CamelCase* e evitar *underscores*, o XML deve evitar maiúsculas e ligar palavras com *underscores*.

Na classe *MainActivity*, declarei todos objetos *Widget*: são um grupo de classes que representam os elementos *View* interativos, como os *TextViews* (Figura 29). Convencionei a nomenclatura conhecida como Notação Húngara: prefixo indicando o tipo (*txt* - *TextView*; *edtxt* - *EditText*), seguido pelo nome.

```

10 public class MainActivity extends AppCompatActivity {
11
12     public TextView txtIMC, txtMensagem;
13     public EditText edtxtPeso, edtxtAltura;

```

Figura 29. Declaração de Objetos *Widget*

Expliquei brevemente o método *onCreate*, o qual cria a tela quando a App é iniciada; tem um sub-método *setContentView*, o qual “inflaciona” o layout, isto é, preenche o ecrã da App com os elementos *View* localizados na pasta de recursos ‘res’(R), pelo do argumento *R.layout.activity_main*.

Expliquei sobre o método *findViewById* para ligar objetos *TextView* e *EditText* aos *Views*. Tal como o *setContentView*, também busca elementos na pasta de recursos, e encontra-os pelos seus *IDs*; o seu argumento é ‘*R.id.elemento_View*’ (Figura 30).

```
15      @Override
16      protected void onCreate(Bundle savedInstanceState) {
17          super.onCreate(savedInstanceState);
18          setContentView(R.layout.activity_main);
19
20          edtxtPeso = findViewById(R.id.edtxt_peso);
21          edtxtAltura = findViewById(R.id.edtxt_altura);
22          txtIMC = findViewById(R.id.txt_imc);
23          txtMensagem = findViewById(R.id.txt_mensagem);
24      }
25
26
27
```

txt_imc (= 1000286)	int
edtxt_altura (= 1000329)	int
edtxt_peso (= 1000009)	int
txt_mensagem (= 1000155)	int

Figura 30. Ligação Código-Tela com *findViewById*

Passei à programação do botão Limpar. Expliquei que “limpar” um texto não é apagar o que está escrito, mas “rescrever em branco”. Chamei o método *setText* do *edtxtPeso* com uma *String* vazia. Os alunos testaram a App no emulador, e o número do Peso foi “apagado” ao tocar no botão Limpar. Apliquei o mesmo método para todos os objetos *Widget* de texto no manipulador Limpar (Figura 31).

```
53      public void limpar(View view) {
54          edtxtPeso.setText("");
55          edtxtAltura.setText("");
56          txtIMC.setText("");
57          txtMensagem.setText("");
58      }
```

Figura 31. Manipulador de Eventos Limpar com *setText*

Até aqui, a leção sedimentou conceitos aprendidos nas aulas anteriores: *onClick* para botões, *findViewById* para ligar objetos, *setText* para rótulos. Doravante, os alunos iriam aprender novos conceitos: o método *getText*, e a conversão de valores com o método *toString*.

Passei à programação do botão Calcular; disse-lhes para fazerem a ligação tela-código e o teste *Logcat*. Expliquei como o método *getText* busca valores dos objetos *TextView* e *EditText*. No entanto, estes objetos armazenam os valores em variáveis que não são do tipo *String*; é preciso convertê-los primeiro, através do método *toString*. Testei no emulador este método conversor com uma variável local *strPeso*, mostrando-a no rótulo IMC (Figura 32).

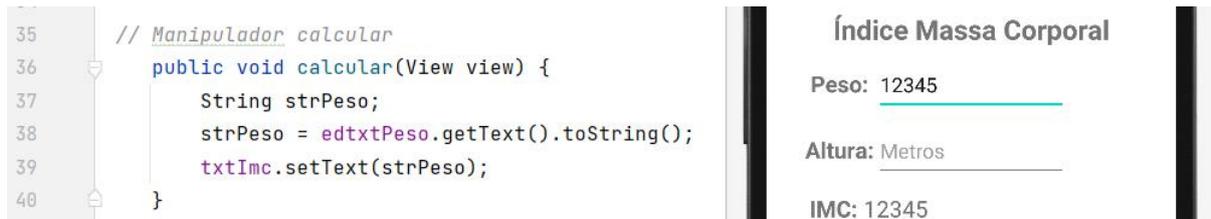


Figura 32. Teste de Recepção do Valor do Peso

Propus que os alunos fizessem o mesmo para a 'Altura', mostrando o valor no rótulo 'Mensagem'. Por esta altura, o tempo de sessão terminou, o Turno 2 saiu da sala, e entraram os alunos do Turno 1.

A aula correu bem: os alunos não dispersaram, não houve problemas técnicos com o questionário do Google Forms, nem com o Android Studio. A gestão de tempo podia ter sido melhor; não tive tempo de explicar a conversão das *Strings* para *Doubles*, prevista na planificação. Ao rever a sessão em casa, penso que foi desnecessário repetir o teste *Logcat* para o botão *Calcular*.

5.1.2. Sessão 2 – terça-feira 15 de março 2022, Turno 1

Tabela 6. Planificação da Sessão 2

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Ativa. Interrogativa.	45 min	Formativa
Recolher dados sobre raciocínio lógico e visão espacial	- Explicação breve sobre o Pensamento Computacional - 1ª Aplicação do Teste CTt	Teste de Escolha Múltipla através do Google Forms		Sumativa
Intervalo (11h30 – 11h40)			10 min	
Demonstrar a App IMC terminada	- Revisão de conceitos anteriores - Elementos View - Funcionamento dos botões	Ativa. Expositiva e Interrogativa. O professor mostra a App finalizada; cria uma sequência de passos para programar uma tarefa, e os alunos são solicitados a seguir esses passos para tarefas semelhantes. O professor circula pela sala, esclarece dúvidas e observa o progresso dos alunos.	5 min	Formativa
Ligação tela-código do botão Limpar	- Atributo Onclick - Manipuladores de Eventos - Teste com Logcat		20 min	
Programar o botão Limpar	- Objetos Widget - Ligação código-tela no método onCreate - Método setText		20 min	

Sumário: 1º Teste Pensamento Computacional. Início da app IMC.

A segunda sessão seguiu um percurso parecido ao da primeira, exceto que avançou apenas até à ligação *findViewById* no método *onCreate*.

Fiz uma exposição curta sobre o conceito de Pensamento Computacional; disse aos alunos que iria comparar a pontuação dos testes CTt e analisar se a aprendizagem de Android contribui para o desenvolvimento desta habilidade. Avisei que depois dos testes iria fazer entrevistas individuais aos alunos sobre os resultados obtidos. Quando o teste começou, avisei também que a questão nº10 tinha um erro (Figura 33). O teste terminou em 40 minutos e os alunos saíram para o intervalo.

Depois do intervalo, iniciei a aula de Android com uma curta revisão dos conceitos da App *Hello Toast*; as ligações tela-código para o botão *Count* e ligações código-tela para ambos o botão e o rótulo de contagem. O novo conceito a aprender nesta aula seria a caixa de texto; como fazer as suas ligações código-tela, e como manipular os valores escritos nelas.

10. Qual a linha que falta na sequência (__?) para que o Pac-Man chegue até 1 ponto ao Fantasma?

```

do{
  moveParaFrente();
  moveParaFrente();
  _____?
  viraDireita();
  moveParaFrente();
  viraEsquerda();
} while (fantasma = 1);

```

Opção A
viraEsquerda();

Opção B
viraDireita();

Opção C
moveParaFrente();

Opção D
Não falta nenhuma linha

Figura 33. Questão 10 do teste CTt; Opção C Corrigida

Demonstrei o funcionamento da App IMC no projetor; os alunos baixaram o projeto com a tela pré-feita na página do Nónio. Direcionei os alunos ao ficheiro *activity_main*, e dei instruções para adicionar o atributo *onClick* ao botão Limpar e criar o manipulador de eventos. Os alunos testaram o botão Limpar com o *Logcat*, confirmando que a ligação tela-código estava a funcionar.

Para programar o botão Limpar, expliquei como primeiro os elementos *View* têm de ter um *ID*. Indiquei os atributos de identificação no *activity_main*, e os alunos renomearam os IDs através do comando *Refactor > Rename*. Adverti que renomeassem com uma notação simples e que diferenciasses dos objetos Java. Declarei objetos *Widget* no *MainActivity* – os *TextViews* e os *EditTexts*. Expliquei o método *setContentview*, o qual “inflaciona” o *layout* com os recursos da pasta ‘res’. Os alunos fizeram a ligação código-tela dos objetos *Widget* com o método *findViewById*. Alguns alunos lembraram-se deste conceito, aprendido no “Hello Toast”.

Passei à programação do botão Limpar. Expliquei que “apagar” um texto é o mesmo que “sobrescrever” com uma *String* vazia. Fiz uma analogia das funções *Getter* e *Setter* do Java, com as funções *getText* e *setText*. Estas funções foram chamadas no manipulador Limpar para os quatro objetos de texto da App. Os alunos testaram o botão Limpar para apagar quaisquer textos digitados, confirmando que os objetos estão ligados aos elementos *View*.

Para programar o botão Calcular, comecei por mencionar as *Hints* “Quilos” e “Metros”, e a fórmula de cálculo no rótulo IMC. Expliquei que são atributos XML, que funcionam como rótulos temporários, e que servem para indicar o tipo de dado a ser inserido na caixa de texto (Figura 34).

The image shows a web form with a blue header containing the text "Índice Massa Corporal". Below the header, there are three rows of text. The first row is "Peso: Quilos" followed by a horizontal line representing an input field. The second row is "Altura: Metros" followed by a horizontal line representing an input field. The third row is "IMC: $\text{Peso} / (\text{Altura}^2)$ ".

Figura 34. *Hints* nas Caixas de Texto e no Rótulo IMC

Comecei a programar o botão Calcular através de um teste de receção de dados: o valor digitado na caixa 'Peso' seria copiado para o rótulo 'IMC', e o valor da 'Altura' para o rótulo 'Mensagem'.

Por esta altura, o tempo de sessão estava quase a terminar. Informei que na próxima aula iríamos aprender este novo método de buscar valores das caixas de texto, o *getText*.

Esta sessão correu melhor que a primeira. A sequência dos conteúdos fluiu mais facilmente; consegui cumprir a planificação deste turno, com tempo de sobra. Depois de lecionar o método *setText*, tentei avançar um pouco a matéria de modo a acompanhar no turno anterior.

Talvez não devesse ter mencionado os *Hints* nesta aula: este atributo estava planeado para a sessão do ICQ. Devia ter mantido o foco no *getText* das caixas de texto. Quanto à notação húngara: há quem discorde da sua utilidade, mas creio que serve de apoio para iniciantes de programação.

5.1.3. Sessão 3 – sexta-feira 18 de março 2022, Turno 1

Tabela 7. Planificação da Sessão 3

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento e informar os objetivos da aula	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	5 min	Formativa
Revisão de conceitos	- Atributo <code>onClick</code> - Ligação código-tela no método <code>onCreate</code>	Ativa. Aula prática demonstrativa e interrogativa. O professor demonstra uma sequência de passos para programar uma parte do código.		
Programar o botão Calcular sem a classe IMC	- Método <code>getText</code> - Conversão de String para Double - Conversão de Double para String	Os alunos são solicitados a seguir os passos para outras partes. O professor circula pela sala a esclarecer dúvidas e a observar como os alunos executam a tarefa.	40 min	
	Intervalo (9h50 – 10h00)		10 min	
Programar o botão Calcular com a classe IMC	- Importação/criação da classe IMC - Instanciação do objeto IMC - Conversão de Double para String	Ativa. Aula prática demonstrativa e interrogativa. (como antes do intervalo)	45 min	Formativa
Prevenir o botão Calcular contra exceções nulas	- Verificação de caixas de texto vazias		30 min	
Programar o botão Calcular com o método <code>classifica</code>	- Método <code>classifica</code> da classe IMC		15 min	

Sumário: Continuação da app IMC, função Calcular

Comecei por rever os conceitos: atributo `onClick`, manipulador de eventos, objetos `TextView` e `EditText`. Relembrei a ligação dos objetos às `Views` pelo `findViewById` e o método `setText`.

Repeti a programação botão Limpar. Chamei os métodos `setText` com `Strings` vazias para todos os objetos `Widget` de texto. Digitei o código enquanto perguntava aos alunos ‘o que é que este comando faz?’, ‘porque é que se faz isto?’, ‘o que vem a seguir?’, etc. Enquanto consolidava os conceitos da aula anterior, cometi um erro, e este mostrou-se ao testar no emulador: o botão Limpar não funcionava. Perguntei “alguém reparou o que fiz de errado?”, e depois encontrei o erro: o atributo `onClick` foi criado no botão Calcular ao invés do Limpar. Na sessão seguinte iria demonstrar o `Debugger` para resolver um problema intencional com exceções nulas.

Continuei a programação do botão Calcular. Expliquei como este manipulador deve receber os valores das caixas de texto, fazer um cálculo e escrever nos rótulos o resultado. Relembrei que a função que recebe valores é a *getText*, e fiz um teste de recepção do valor Peso (Figura 35)

```

36 // manipulador calcular
37 public void calcular(View view) {
38     txtImc.setText(edtxtPeso.getText());
39 }

```

Figura 35. Teste *getText* para Peso

Mostrei como fazer para o botão Calcular receber os valores Peso e Altura, e converter esses valores em *Strings*, com os métodos encadeados *getText* e *toString*. Depois, fiz o botão mostrar essas *Strings* nos rótulos IMC e Mensagem (Figura 36).

```

38 public void calcular(View view) {
39
40     String strPeso, strAltura;
41     strPeso = edtxtPeso.getText().toString();
42     strAltura = edtxtAltura.getText().toString();
43     txtImc.setText(strPeso);
44     txtMensagem.setText(strAltura);

```

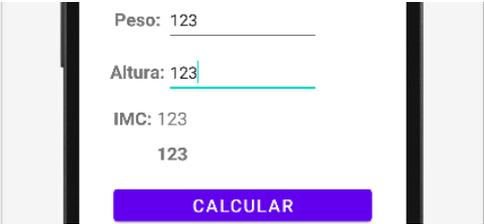


Figura 36. Conversão do Peso e Altura para *String*

Observei os alunos a codificarem. Perguntei ao turno “como fariam para somar o Peso e Altura?”. Um aluno disse que era preciso criar variáveis. Expliquei que as *Strings* têm de ser convertidas para *Double* para poder fazer cálculos; depois de calcular, o valor *Double* volta a ser convertido para *String*, para poder ser escrito no rótulo IMC. À medida que ia explicando, digitei comentários entre as linhas de código para ajudar a compreensão dos passos da conversão (Figura 37).

```

39 // declarar váriaveis String e Double
40 String strPeso, strAltura, strImc;
41 Double dblPeso, dblAltura, dblImc;
42 // receber os valores em Strings
43 strPeso = edtxtPeso.getText().toString();
44 strAltura = edtxtAltura.getText().toString();
45 // converter as duas Strings em Doubles
46 dblPeso = Double.parseDouble(strPeso);
47 dblAltura = Double.parseDouble(strAltura);
48 // somar os dois Doubles
49 dblImc = dblPeso + dblAltura;
50 // converter Double para String
51 strImc = Double.toString(dblImc);
52 // escrever rótulo imc com a String
53 txtImc.setText(strImc);

```



Figura 37. Conversão de *Strings* para *Doubles* e Teste de Soma de Valores

Expliquei como a variável *Double* (com maiúscula) é um objeto com métodos próprios de conversão: de *String* para *Double* utiliza-se o *parseDouble*, e *toString* para vice-versa. Disse aos alunos para aplicarem estes métodos para efetuarem a soma, e depois substituírem pelo cálculo do IMC.

Pu-los a programar e circulei pela sala a tirar dúvidas. Um aluno conseguiu concisar o código sem usar nenhuma variável auxiliar *Stringe Double*, de modo que o manipulador Calcular reduziu-se a apenas uma linha de comando (Figura 38)

```
36 // Manipulador calcular
37 public void calcular(View view) {
38     txtImc.setText(Double.toString(
39         d: Double.parseDouble(edtxtPeso.getText().toString()) /
40         Math.pow(Double.parseDouble(edtxtAltura.getText().toString()),2));
41
```



Figura 38. Concisão Máxima do Manipulador Calcular

Passei à importação da classe IMC. Fui ao painel de projeto e transferei o ficheiro de classe para a pasta Java de domínio (neste caso: *com.example.imc*), e adicionei o comando *package*. O professor José Pereira advertiu que esta operação devia ser feita de outro modo – criar uma nova classe Java, e colar o código IMC na classe criada. Esperei que os alunos criassem as novas classes; reparei que a maioria não tinha os trabalhos IMC com eles, pediram trabalhos emprestados a outros colegas.

Mostrei a classe IMC no projetor, com o construtor e a função *calculaIMC* (Figura 39).

```
1 package com.example.imc;
2
3 public class Imc{
4     public double peso, altura;
5
6     // construtor
7     public Imc(double peso, double altura){
8         this.peso = peso; this.altura = altura;
9     }
10
11     public Double calculaImc(){
12         return peso/(altura*altura);
13     }
}
```

Figura 39. Classe IMC, sem o Método Classifica

Expliquei a programação do método Calcular começando por instanciar um objeto IMC.

Os parâmetros do construtor IMC são o Peso e Altura; estes valores devem ser recebidos pelos métodos *getText* e pelos conversores *toString* e *parseDouble* (Figura 40).

```
37 Imc imc = new Imc(
38     Double.parseDouble(edtxtPeso.getText().toString()),
39     Double.parseDouble(edtxtAltura.getText().toString()));
```

Figura 40. Instanciação do Objeto IMC

Perguntei “qual o próximo passo?”, e um aluno respondeu “é chamar o calculador e pôr no *setText*”. Concordei, mas adverti que o valor de retorno do *calculaIMC* é um objeto *Double*, e o *setText* só aceita objetos *String*. Disse para converterem para *String* com o método *toString* (Figura 41).

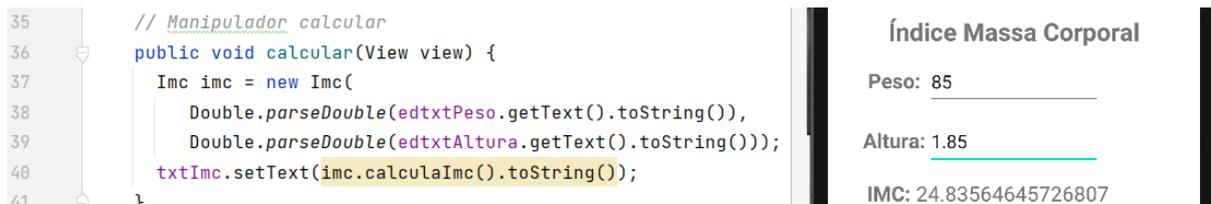


Figura 41. Conversão com *Double.toString*

Após testarem o código no emulador, aponte para o *txtImc.setText* e perguntei “porque é que o Android Studio fez este realçado a amarelo?”

Expliquei que é um aviso do analisador do compilador: apesar do código não ter erros de sintaxe, ainda pode induzir a erros (*bugs*) devido à não especificação ou incompatibilidade de versões. Neste caso, ao converter um objeto *Double*, com numeros reais e várias casas decimais, para um objeto *String*, é aconselhado especificar a localidade, porque, dependendo do país, a vírgula pode não ser reconhecida como separador decimal (por exemplo, nos EUA usa-se ponto).

Recomendei usarem a função *String.format*. O problema é que esta função não recebe objetos *Double*. Então modifiquei o método *calculaIMC* para retornar primitivas *double* (Figura 42)

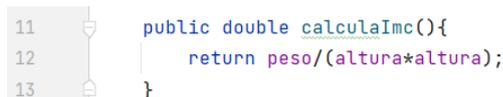


Figura 42. Método *calculaIMC* com *doubles*

A função *String.format* pode ter até três parâmetros – localizador, especificador de formato, e o valor a ser formatado. Para o localizador, usa-se *Locale.US*; para o especificador, *%.2f* (isto é, um *floating point* com duas casas decimais); e para o valor, o resultado do *calculaIMC* (Figura 43).



Figura 43. Conversão com *String.format*

Observei os alunos a fazerem a conversão. Reparei que alguns ainda estavam no construtor IMC, aparentemente confusos com o uso dos *Double.parseDouble*. Voltei a explicar que o *Double* é um objeto com funções próprias, e *double* é uma variável primitiva.

Passei à prevenção contra exceções nulas. Projetei o emulador no quadro e perguntei “O que acontece se eu clicar no ‘Calcular’ sem ter nada no Peso e na Altura?”, e demonstrei como o programa pára de funcionar, ou ‘estoura’. Expliquei que aconteceu um erro de exceção, no qual tentámos passar um valor nulo para o método *getText.toString* de um dos objetos *EditText*.

Expliquei que, para evitar estas exceções, deve-se criar uma condição *if* para verificar se as caixas de texto estão vazias. Se estiverem, o método *Calcular* deve terminar com *return*, e indicar no rótulo

'Mensagem' qual a caixa de texto vazia com *setText*. Se não estiverem, os valores podem continuar para serem introduzidos no construtor.

Digitei a condição *if*, e perguntei aos alunos se se lembravam de como se faz este tipo de verificação; alguns disseram que tinham usado o "duplo igual a nulo" (`== null`) para estes tipos de problemas. Adverti-os então que o "duplo igual" não iria funcionar para este caso. Testei no emulador e mostrei como a aplicação continua a 'estourar' (Figura 44).



Figura 44. Verificação Errada de Valores Nulos com *String == null*

Mostrei-lhes o aviso realçado a amarelo, em que a condição será sempre *false*. Expliquei que é preciso usar o método *isEmpty*, e demonstrei como a App não estoura deste modo (Figura 45).

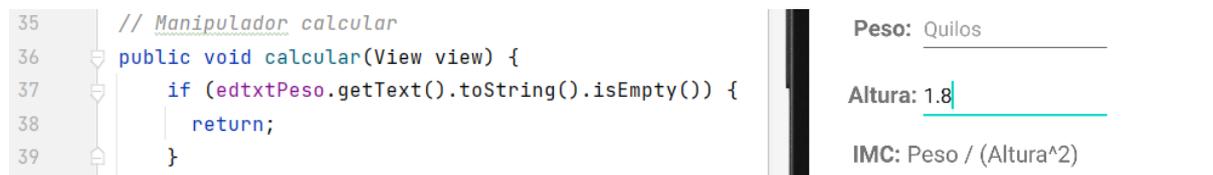


Figura 45. Verificação Correta de Valores Nulos com *String.isEmpty*

Por esta altura, o tempo de aula estava quase a terminar; não haveria tempo para lecionar o método *classificaPeso*. Orientei-os a aplicar as condições *ifs* com a função *isEmpty*, para cada caixa de texto, e também para aplicarem um *setText* a indicar o nome da caixa vazia.

Circulei pela sala a ver o código dos alunos. A maioria ainda estava na conversão dos *doubles*. Dediquei os minutos restantes a explicar individualmente como usar o *String.format*.

A aula terminou, e os alunos saíram. O próximo turno chegaria após o intervalo.

5.1.4. Sessão 4 – sexta-feira 18 de março 2022, Turno 2

Tabela 8. Planificação da Sessão 4

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento e informar os objetivos	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	5 min	Formativa
Revisão de conceitos da aula anterior	- Atributo <code>onClick</code> - Ligação <code>findViewById</code> - Método <code>getText</code>	Ativa. Aula prática demonstrativa e interrogativa. O professor demonstra uma sequência de passos para programar uma parte do código. Os alunos são solicitados a seguir os passos para as outras partes. O professor circula pela sala, esclarece dúvidas e observar a execução das tarefas.		
Programar o botão Calcular com classe IMC	- Importação/criação da classe IMC - Instanciação de um objeto IMC - Conversão Double-String - Conversão/formatação String-double		40 min	
Prevenir o botão Calcular contra exceções nulas	- Método <code>getText.toString.isEmpty</code>		30 min	
Programar o botão Calcular com o método <code>Classifica</code>	- Método <code>classifica</code>		15 min	

Sumário: Continuação da app IMC, função Calcular

Nesta sessão, a Professora Supervisora Altina Ramos esteve presente; o meu colega de estágio Pedro Sequeira não pôde assistir à aula, seis alunos chegaram atrasados.

Continuei a leção da App IMC a partir do trabalho da aula anterior, na qual começámos a programar o botão Calcular, sem uma classe IMC externa. Disse que iríamos recomeçar a programação do botão, e depois iríamos importar a classe IMC, contendo os métodos de cálculo e classificação.

Revi brevemente os conceitos anteriores: o atributo `onClick` e o método `findViewById` dentro do `onCreate`. Testei a receção de valores e chamei o método `getText` dos objetos `Peso` e `Altura`, dentro dos seus respetivos métodos `setText`, para que os valores das caixas de texto fossem mostrados nos rótulos ao clicar no botão Calcular.

Como este turno só teria 2 aulas de 45 minutos na sessão 4, calculei que se continuasse pelo trajeto de programar o botão com variáveis locais, para depois reprogramá-lo com a classe externa – como fiz com o turno anterior - não teria tempo de lecionar o método `Classifica`.

Decidi então passar direto para a criação da classe IMC e programar a recepção dos valores no construtor de objetos IMC. Ao contrário do turno anterior, a classe IMC não foi importada; foi criada a partir do zero, contendo duas variáveis *double*, um construtor, e um método *calculaIMC* que retorna uma primitiva *double* (Figura 46).

```
1 package com.example.imc;
2 public class Imc{
3     public double peso, altura;
4     public Imc(double peso, double altura) {
5         this.peso = peso; this.altura = altura;
6     }
7     public double calculaImc(){
8         return peso/(altura * altura);
9     }
```

Figura 46. Classe IMC do 2º turno

Passei para o manipulador *Calcular* na *MainActivity*, apaguei as variáveis auxiliares *strPeso* e *strAltura* da aula passada, e criei um objeto *Imc*. Copiei os métodos *getText.toString* feitos no teste de recepção de valores da a revisão, e coloquei-os como parâmetros do construtor *imc* (Figura 47)

```
36 // Manipulador calcular
37 public void calcular(View view) {
38     Imc imc = new Imc(
39         edtxtPeso.getText().toString(),
40         edtxtAltura.getText().toString());
41 }
```

Figura 47. Exemplo de Instanciação de Objeto IMC com Variáveis Erradas

Expliquei que os avisos de erro são devido aos valores no construtor serem *Strings*, e não *doubles* como na classe IMC. Completei os parâmetros com dois *parseDouble*, e disse aos alunos para chamarem o método *calculaIMC* para o rótulo IMC, e testarem no emulador.

Circulei pela sala a ver o progresso. Muitos alunos ainda estavam a fazer a classe IMC. Esclareci dúvidas dos alunos menos interessados sobre a instanciação do objeto *imc*. Alguns alunos estavam a tentar resolver o problema do rótulo IMC sem converter o resultado do método *imc.calcula* para *String*, que era o ponto onde eu queria chegar (Figura 48)

```
36 // Manipulador calcular
37 public void calcular(View view) {
38     Imc imc = new Imc(
39         Double.parseDouble(edtxtPeso.getText().toString()),
40         Double.parseDouble(edtxtAltura.getText().toString()));
41
42     txtImc.setText(imc.calculaImc().toString());
43     // txtMensagem.setText();
44 }
```

Figura 48. Exemplo de Conversão Errada para *String*

Digitei o código e perguntei “porque é que *toString* do IMC não funciona?”. Expliquei que é porque criámos um método *calculaIMC* que retorna primitivas *double* e não objetos *Double*. São os objetos *Double* que possuem funções conversoras do tipo *toString* e *parseDouble*. Disse que devemos utilizar

objetos *Doubles* apenas na classe *MainActivity* para fazer uso destas funções conversoras; mas na classe *IMC*, devemos usar primitivas *double*.

Para escrever o resultado nos rótulos com *setText*, fazemos a conversão *String.format*. Expliquei como utilizar a função, com o parâmetro de local *US*, o parâmetro de especificação *%.2f*, e o parâmetro do resultado *imc.calculaIMC.*, que é um *double*.

Passei à prevenção de exceções nulas. Perguntei ao turno o que aconteceria se mandasse calcular o IMC com o Peso e Altura vazios; e mostrei no emulador como a App ‘estoura’. Expliquei que se trata de um erro de exceção nula - acontece quando passamos um valor nulo para um método.

Implementei o *if- isEmpty* para o Peso e Altura, antes da instanciação do objeto *IMC*; para avisar o utilizador que uma das caixas está vazia, usei o *setText* do rótulo Mensagem. Digitei a mensagem “Insira o peso” para a verificação do Peso, depois copieei-a para a Altura, alterando os nomes correspondentes. Projetei o emulador no quadro, e testei o botão Calcular com as novas condicionais *if* - e deparei-me com o ecrã de erro (Figura 49).

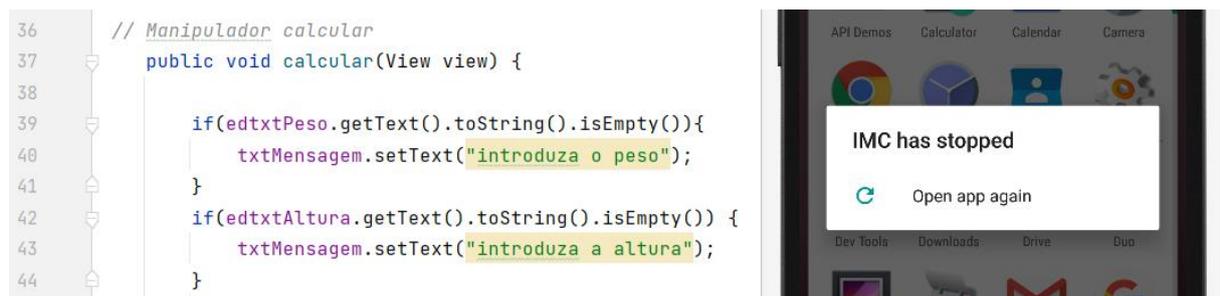


Figura 49. Verificação Incorreta: Condicionais *if* sem retorno

Corrigi o meu erro: esqueci de colocar o retorno (*return*) no final dos *ifs* de modo a terminar o método para a condição *isEmpty*. Pedi desculpa pelo lapso, e um dos alunos disse “Não faz mal.”

Passei para o método *classificaPeso*. Trata-se de uma série de condições *if* para várias gamas numéricas do IMC, e cada condição retorna uma *String* indicando o grau de obesidade (Figura 50).

```
12 public String classificaPeso(){
13     double imc = calculaImc();
14     if (imc < 18) { return "Peso baixo";}
15     if (imc >= 18 && imc < 25){ return "Peso normal";}
16     if (imc >= 25 && imc < 30){ return "Pré-obesidade";}
17     if (imc >= 30 && imc < 35){ return "Obesidade grau I";}
18     if (imc >= 35 && imc < 40){ return "Obesidade grau II";}
19     else { return "Obesidade grau III";}
20 }
```

Figura 50. Método Classifica da Classe *IMC*

Por esta altura, tinha terminado a parte expositiva da sessão, e passei o último quarto de hora a circular pela sala e a tirar dúvidas sobre os conceitos da sessão 4 – em particular a instanciação de objetos e o *String.format*.

5.1.5. Sessão 5 – terça-feira 22 de março 2022, Turno 2

Tabela 9. Planificação da Sessão 5

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	10 min	Formativa
Rever conceitos das aulas anteriores	- <code>findViewById</code> - <code>getText.toString.isEmpty</code> - <code>parseDouble</code> , <code>stringFormat</code>	Ativa. Aula prática demonstrativa e interrogativa. O professor demonstra uma sequência de passos para programar uma parte do código.		
Terminar a App IMC	- Método <code>classifica</code> - Entrega do trabalho: até 29/03	Os alunos são solicitados a seguir os passos para as outras partes. O professor circula pela sala, esclarece dúvidas e observa a execução das tarefas.	30 min	
Demonstrar a App ICQ terminada	- Elementos View - Funcionamento dos botões - Ecrã em orientação horizontal		5 min	
Construir as telas da App ICQ	- 3 <code>ConstraintLayouts</code> : cabeçalho (título), corpo (caixas de texto) e rodapé (botões)		45 min	
	Intervalo (9h50 – 10h00)		10 min	
Construir as telas da App ICQ	- <code>ScrollView</code> (rolagem) - <code>Constraint Baselines</code> (alinhamento)	Ativa. Aula prática demonstrativa e interrogativa. (como antes do intervalo)	45 min	Formativa

Sumário: Conclusão da app IMC com método `classifica`. Início da app ICQ com construção da tela.

Avisei aos alunos que depois de entregarem o IMC, iríamos começar a App ICQ, que é semelhante mas mais complexa. Quatro alunos chegaram atrasados.

Na aula anterior, o T2 ficou pelo método `Classifica`. Fiz uma revisão dos conceitos, inclusive a ligação código-tela pelo `findViewById` (alguns alunos não tinham aplicado este método no código). Lembrei que os valores das caixas de texto têm de ser recebidos por conversão do `getText.toString`, e que para construir o objeto IMC têm de ser convertidos para primitivas `double`, através do método conversor `parseDouble`. Também lembrei que a App ‘estoura’ se o conversor `getText.toString` receber valores nulos, por isso, temos de implementar as condicionais `if` para filtrar estas exceções, através do método `isEmpty`.

Para terminar o IMC, mostrei o método `classifica` e disse para o chamarem no rótulo Mensagem. Como este método retorna uma `String`, não necessita de conversão para o `setText`.

Depois de mostrar o método, um dos alunos (o mais atrasado na aprendizagem) chamou-me e disse que não conseguia “pôr o botão a funcionar”. Vi o código dele e notei que não tinha feito sequer a ligação código- tela com `findViewById`. Expliquei como se faz, depois dediquei cinco minutos a tirar dúvidas a outros alunos individualmente.

Muitos alunos tinham implementado o método Classificação, mas poucos tinham feito a verificação de caixas vazias antes de instanciar o objeto IMC.

Disse para enviarem os trabalhos IMC para o meu email institucional; e que quem não tivesse terminado, teria até segunda-feira da semana seguinte para entregar. Mostrei o meu email no projetor, copiei o projeto ICQ do meu portátil para o computador do docente, de modo a iniciar a leção com a demonstração da aplicação terminada.

No entanto, ao tentar abri-lo no Android Studio, ocorreu um problema técnico: o Android Studio do PC docente não reconheceu o endereço (*path*) da pasta SDK do Android. Tentei sincronizar o projeto com os ficheiros Gradle, e depois de duas tentativas e três minutos, o problema persistiu.

Fiquei perplexo; não sabia configurar o endereço do SDK (Software Development Kit), e calculei que iria levar alguns minutos para descobrir na internet como se faz. Considerei outras hipóteses de continuar a aula: conectar o meu PC portátil ao projetor e mostrar o meu projeto ICQ, ou copiar o código do ICQ e pastá-lo num novo projeto Android Studio no PC docente.

Disse aos alunos para esperarem um pouco devido a este contratempo, e eles distraíram-se na internet. O Professor José Pereira ajudou-me: abriu o painel *Tools > SDK Manager > Edit*, e editou o endereço SDK (Figura 51); este estava configurado para uma pasta com o nome do meu computador (C:\Users\ASUS\...). Bastou mudar o endereço para a pasta do Users\Administrador do PC docente.

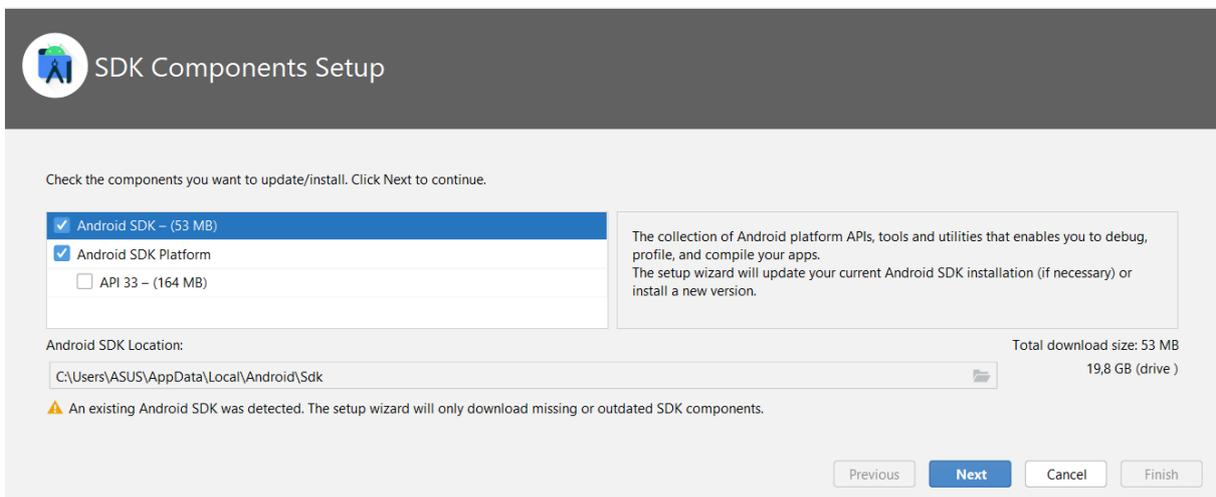


Figura 51. Editor de Componentes Android SDK

Depois deste contratempo, a sessão pôde continuar. Comecei por demonstrar a App ICQ; consiste de dois botões (Calcular e Limpar), cinco caixas de texto (Nome, Sexo, N° de Cidadão 'CC', Cintura e Quadril) e dois rótulos (ICQ e Mensagem) (Figura 52)

Ao tocar no botão Calcular, a App deve mostrar o resultado do cálculo ICQ no rótulo ICQ, e no rótulo Mensagem, um texto a indicar se o ICQ do utilizador está dentro ou acima do recomendado.

A App deve avisar se há alguma caixa de texto vazia.

O utilizador deve preencher pelo menos a Cintura e Quadril para a App poder calcular o ICQ.

O utilizador deve preencher pelo menos a Cintura, Quadril e o Sexo para a App poder calcular o ICQ, e a classificação do ICQ.

As caixas de texto Nome e CC são acessórias, mas o Nome pode ser chamado na mensagem de classificação (por exemplo, “João, o seu ICQ está dentro do recomendado”).

Ao tocar no botão Limpar, todos os rótulos e caixas de texto são apagados.

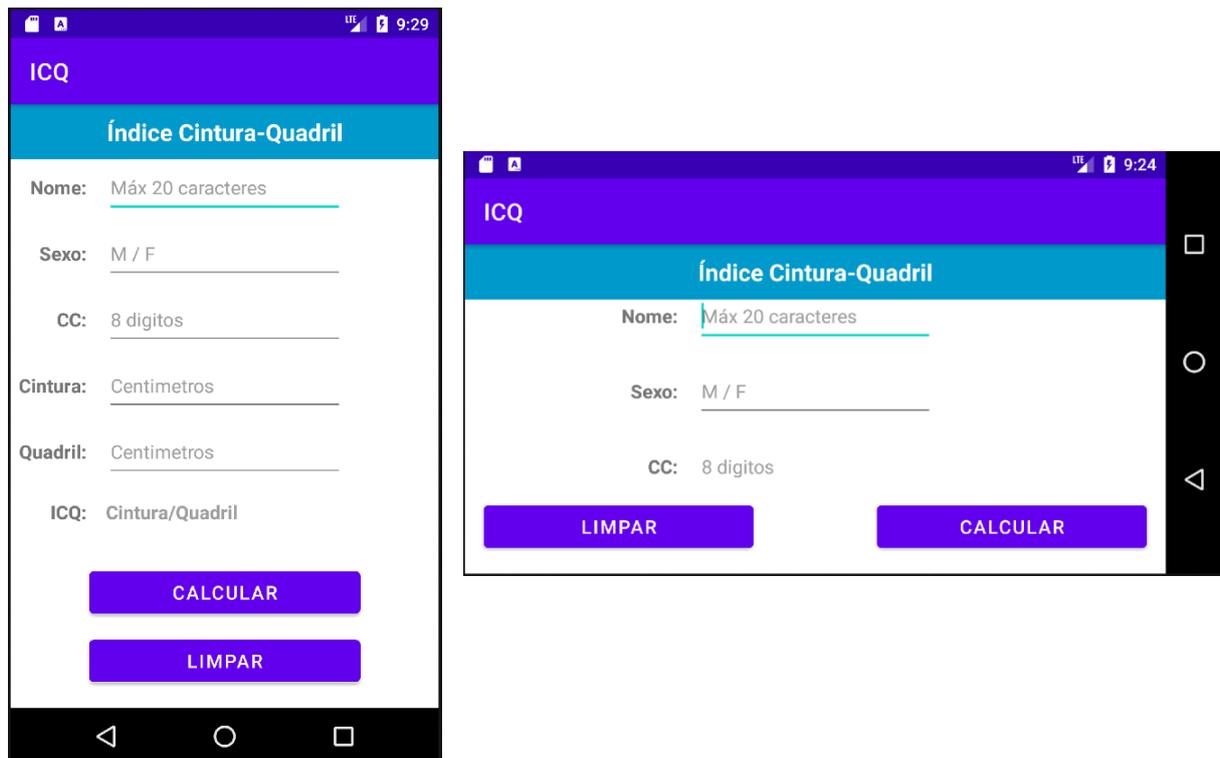


Figura 52. App ICQ em Orientação Vertical (esquerda) e Horizontal (direita).

O ecrã principal deve ser rolável, e também deve ter uma tela pré-configurada para a orientação horizontal, na qual o título e os botões ficam fixos no topo e no fundo.

Após demonstrar o funcionamento da App, os alunos iniciaram o Android Studio e criaram um novo projeto, do template *Empty Activity*, com o SDK mínimo de API 21: Android 5 (Lollipop).

Abri o painel *activity_main*, e expliquei-lhes a estrutura da App ICQ. A App deve ser organizada em três secções: cabeçalho, corpo e rodapé. O cabeçalho é composto por um rótulo com o nome da aplicação. O corpo é composto pelas várias caixas de texto, que servem para receber os dados a serem calculados. O rodapé é composto pelos dois botões que executam as funções da App. Os alunos devem organizar a tela com 3 *Viewgroups* do tipo *ConstraintLayout*, e 1 *ViewGroup* do subtipo *ScrollView*, do tipo *FrameLayout*. Os *ConstraintLayouts* servem para restringir (*constrain*) os elementos *View* dentro deles. O *ScrollView* permite ver estes *Views* na orientação horizontal pela rolagem da tela.

Transformei o rótulo *Hello World* no cabeçalho. Disse para abrirem o rótulo “*Hello World*” no painel dos Atributos; fui ao *Constraint Widget*, e removi a restrição de baixo (*bottom*). Isto fez com que o rótulo ficasse adjacente ao topo do seu layout-pai (Figura 53).

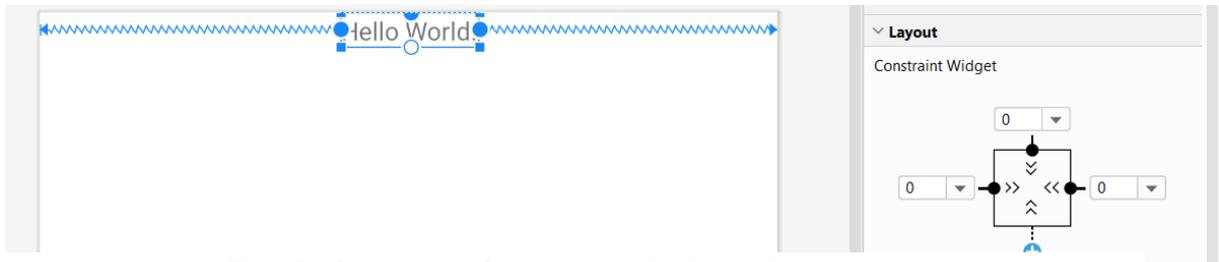


Figura 53. Construção do Cabeçalho: Modificação do rótulo *Hello World*

Os atributos devem ser: *id*“header_title”; *text*“Índice Cintura-Quadril”; *padding*“10dp”; *background* “color/holo_blue_dark”. Assim, o rótulo passou a ser o cabeçalho (Figura 54).

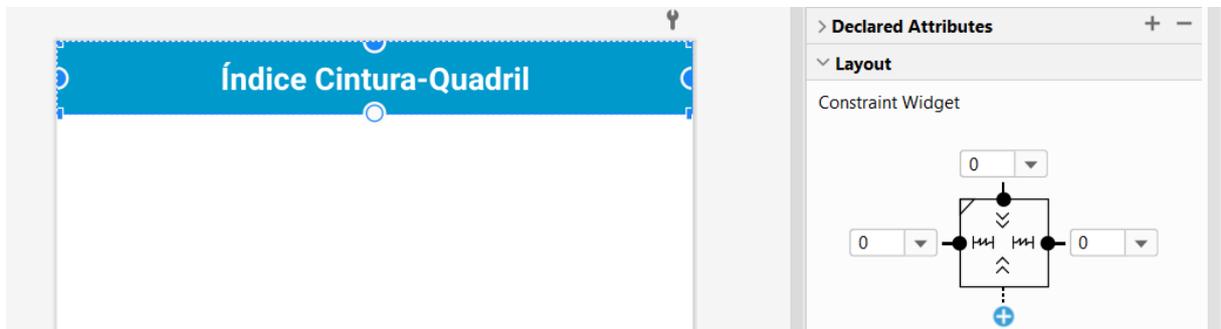


Figura 54. Construção do Cabeçalho: *Padding* e *Background Color*

Passsei para a construção do rodapé. Na paleta, selecionei um elemento *ConstraintLayout* na lista de *Layouts*, e arrastei-o para debaixo do cabeçalho. Arrastei os constraints do começo (lado esquerdo), fim (lado direito) e fundo (lado de baixo) às bordas da tela (Figura 55). Configurei os atributos do novo *ConstraintLayout*.

- *id*:"constraintlayout2";
- *layout_width* "match_parent";
- *layout_height* "200dp".

Adverti aos alunos que é importante atribuir uma chave de identificação aos layouts, assim será possível fazer *constraints* entre elementos mesmo quando os pontos de conexão não estão visíveis no editor de Design. Defini o id do *ConstraintLayout* original como "constraintlayout1".

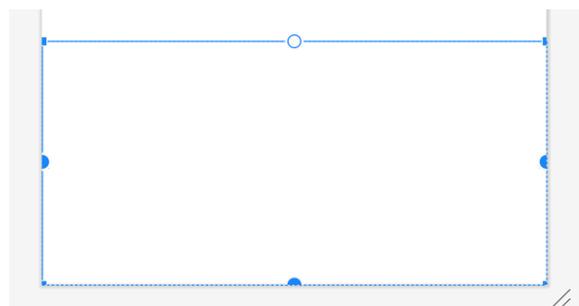


Figura 55. Construção do Rodapé: *ConstraintLayout*

Expliquei o porquê de a altura do rodapé estar excessivamente alta - devemos definir uma altura excessiva para permitir espaço suficiente para arrastar os botões para dentro deste layout. Quando estiver terminado, a altura do rodapé poderá ser redefinida como “*wrap_content*”.

Arrastei dois elementos *Button* da paleta para dentro do rodapé; depois arrastei os *constraints* de modo a que fiquem centrados e empilhados (Figura 56).

Assim, o rodapé consiste de um *ConstraintLayout* separado, com dois botões. Isto foi para que os botões estejam sempre visíveis na borda inferior quando o ecrã estiver em orientação horizontal; se os botões não fizerem parte do *ScrollView*, não serão rolados para baixo com o resto.

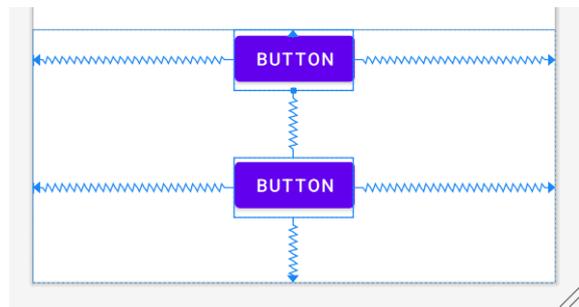


Figura 56. Construção do Rodapé: Botões

Circulei pela sala a observar o progresso dos alunos. Alguns estavam a passar à frente das instruções, já a construir as caixas de texto, enquanto outros estavam a tentar seguir a estrutura cabeçalho-corpo-rodapé, com dificuldades nos *constraints*.

Reparei que, da maneira que muitos estavam a fazer – apenas com um *layout* – quando fossem testar a App no emulador, todos elementos iriam ser rolados para cima junto com a tela.

Esta estrutura foi planeada com três *layouts* para que apenas o *ScrollView* no centro rolasse a tela, enquanto o cabeçalho e o rodapé se mantinham estáticos – mas os alunos estavam a ter dificuldades com esta implementação de três *layouts*.

Avisei que a estrutura de três *layouts* podia ser opcional. O importante era que conseguissem aplicar o *ScrollView*.

Por esta altura, os alunos saíram para o intervalo.

Quando voltaram, passei à construção do corpo da App. A partir da paleta, arrastei um elemento *ScrollView* para baixo do cabeçalho. Disse para arrastarem os *constraints* de modo a que o *ScrollView* ficasse estendido desde a margem de fundo do cabeçalho até à margem de topo do rodapé.

No painel dos atributos do *ScrollView*, configurei-os com:

```
layout_height"0dp" (match_constraint);  
fillViewport>true";  
layout_constraintTop_toBottomOf"@id/header_title";  
layout_constraintBottom_toBottomOf"@id/constraintLayout2".
```

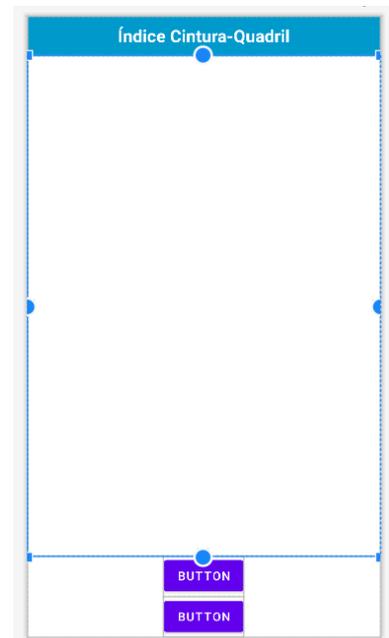


Figura 57. Corpo com *ScrollView*

Expliquei que, para o atributo *layout_height*"0dp", se a altura do *ScrollView* for definida como zero (ou *match_constraint*), ele adapta-se como se fosse *wrap_content*, mas não se sobrepõe a outros *layouts* externos a ele, e fica adjacente ao elemento restrito acima dele - o cabeçalho.

O atributo *fillViewport* “true” faz com que o *ScrollView* estenda a sua altura até ficar igual à altura do dispositivo; isto permite arrastar as caixas de texto para dentro do *constraintlayout2*.

Expliquei que o *ScrollView* não permite *constraints* dentro dele, porque é uma subclasse de *FrameLayout*. Como faz parte de um *ViewGroup* diferente, segue regras e métodos diferentes. Quando o *ScrollView* é arrastado da paleta para o editor de Design, traz com ele, por padrão, um *LinearLayout* com orientação vertical, por isso apaguei esta parte do código. Para poder usar atributos *constraint* nos elementos do *ScrollView*, é preciso inserir um *ConstraintLayout* dentro dele.

Inseri um terceiro *ConstraintLayout*, dentro do *ScrollView* no meio da tela. Configurei os seus atributos como: *id* “constraintlayout3”; *layout_width* “match_parent”; *layout_height* “0dp”. Assim, a tela passou a ter um corpo rolável capaz de ligar elementos com *constraints*. O próximo passo foi a adição destes elementos – as caixas de texto e os rótulos.

Passei a mostrar como o processo de adicionar vários elementos *View* alinhados pode ser feito de um modo eficiente e sistemático através de uma cadeia vertical. Expliquei que uma cadeia (*Chain*) é um grupo de elementos *View* vinculados uns nos outros com *constraints* bidirecionais, podendo ser distribuídos vertical ou horizontalmente.

Adicionei o primeiro elo (*Chain Head*) e arrastei uma caixa de texto para o corpo. Liguei os *constraints* às bordas laterais da tela e do cabeçalho().

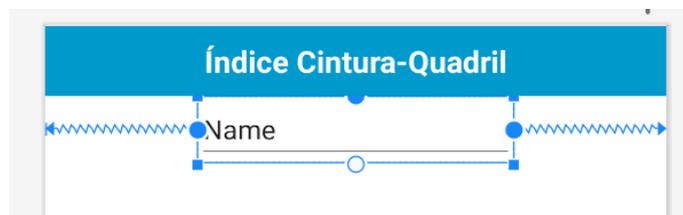


Figura 58. Primeiro Elo da Cadeia Vertical

Arrastei mais quatro caixas de texto e mais dois rótulos para o corpo, um em baixo do outro. Selecionei-os todos, cliquei com o botão direito no primeiro elo, e a partir da lista *drop-down*, selecionei *Chains > Create Vertical Chain*.

Depois, fiz o alinhamento da cadeia com *Center > Horizontally*. Criaram-se *constraints* laterais ligadas às bordas para todos os elos da cadeia (Figura 59). Defini um estilo para a cadeia (*Chain Style*), isto é, a distribuição de espaço entre elos encadeados. A partir do primeiro elo, selecionei *Chains > Vertical Chain Style > packed*.

A cadeia ficou com os elos centrados no *layout* (Figura 59). Expliquei que este estilo impede a variação do espaçamento entre os elos, quando a App for orientada horizontalmente, ou quando for utilizada em telemóveis com ecrãs muito compridos.

Defini uma margem para os espaçamentos: selecionei todos os elos, abri o separador dos atributos e fiz uma pesquisa (*Search*) para filtrar atributos com nome “margin”; expandi a lista *layout_margin*, e atribuí “16dp”.

Passei à orientação horizontal. No painel de Design, abri o seletor de orientação e fiz *Create Landscape Variation*. Automaticamente, o Android Studio criou um ficheiro de *layout* para orientação horizontal na pasta de projeto *res/layout/activity_main*; este novo ficheiro XML tem o sufixo (*_land*).

Circulei pela sala e vi os alunos fazerem as suas telas. Alguns conseguiram colocar e alinhar sem o uso do *Chains*, e alguns dos que usaram *Chains* estavam a experimentar com outros estilos de espaçamento além do *Packed*.

Voltei à secretária, e adicionei os rótulos permanentes; arrastei seis *Textviews* ao lado das caixas de texto, e liguei os *constraints* de fim e de começo.

Passei para a ligação dos *Baselines* dos rótulos. Os *constraints* do tipo *Baseline* fazem um alinhamento ao nível da linha do sublinhado dos textos. No primeiro rótulo, com o botão direito selecionei “*Show Baseline*”. O rótulo criou uma linha curva no nóculo do lado de baixo; arrastei essa linha para o *Baseline* da caixa de texto adjacente, e os elementos alinharam-se. Repeti este processo para os rótulos restantes ().

Disse aos alunos para tentarem fazer o mesmo processo. Dediquei o resto da aula a tirar dúvidas sobre *constraints* e *baselines*.

No final, avisei que a aula seguinte seria dedicada à programação Java da App ICQ.

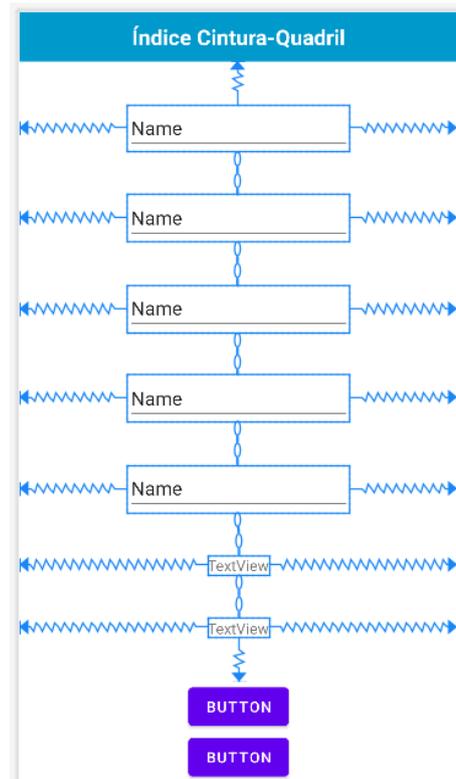


Figura 59. Cadeia Centrada Horizontalmente

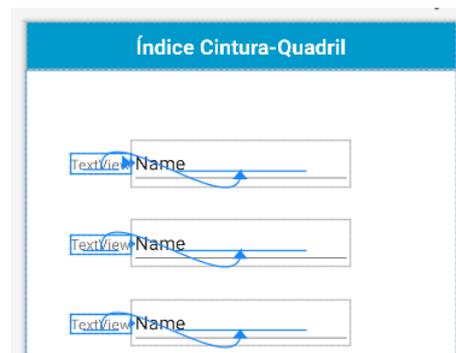


Figura 60. Rótulos e Caixas de Texto com as suas *Baselines* Alinhadas

5.1.6. Sessão 6 – terça-feira 22 de março 2022, Turno 1

Tabela 10. Planificação da Sessão 6

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Expositiva e Interrogativa.	10 min	Formativa
Rever conceitos das aulas anteriores	- findViewById - parseDouble, stringFormat - getText.toString.isEmpty	Ativa. Aula prática demonstrativa e interrogativa. O professor demonstra uma sequência de passos para programar uma parte do código. Os alunos são solicitados a seguir os passos para as outras partes.		
Terminar a App IMC	- Método classifica - Entrega do trabalho: até 29/03	O professor circula pela sala, esclarece dúvidas e observa a execução das tarefas.	35 min	
Intervalo (11h30 – 11h40)			10 min	
Demonstrar o funcionamento da App ICQ	- Elementos View - Funcionamento dos botões - Ecrã em orientação horizontal	Ativa. Aula Prática Demonstrativa e Interrogativa	5 min	Formativa
Construir as telas da App ICQ	- 3 ConstraintLayouts: cabeçalho (rótulo), corpo (caixas de texto) e rodapé (botões)	(como antes do intervalo)	40 min	

Sumário: Conclusão da app IMC com metodo classifica. Início da app ICQ com construção da tela.

A segunda sessão seguiu o mesmo percurso da primeira, exceto que esta avançou apenas até à orientação horizontal de *layout*.

Fiz uma revisão da aula anterior; perguntei sobre as ligações código-tela do *findViewById*, e tela-código do atributo *onClick*; perguntei sobre a conversão de *Strings* para *Doubles* através de método *parseDouble*, e a de *Double* para *String* com *String.format*.

Na aula anterior, o Turno 1 tinha ficado pela verificação de caixas de texto vazias; retomei a parte final deste tópico, na qual o rótulo Mensagem chama o método *setText* para escrever uma *String* a indicar que falta inserir o Peso, ou a Altura (Figura 61).

```

39  if(edtxtPeso.getText().toString().isEmpty()){
40      txtMensagem.setText("Peso?");
41      return;
42  }
43  if(edtxtAltura.getText().toString().isEmpty()) {
44      txtMensagem.setText("Altura?");
45      return;

```

Figura 61. Verificação Correta: Condicionais *if* com retorno

Passei para o método *classifica*. Esta função consiste de uma série de condições *if* que retornam *Strings* diferentes consoante a gama de valores do resultado do cálculo IMC; estas *Strings* aparecem na tela quando o método *classifica* é chamado pelo *setText* do rótulo Mensagem.

Circulei pela sala, observando o código dos alunos; todos tinham trazido o trabalho anterior do IMC em Java e estavam a copiar a função. Alguns ainda tinham dúvidas na parte de conversão de *Double* para *String*, e passei alguns minutos a esclarecer individualmente.

A primeira aula de 45 minutos terminou, e mostrei no projetor o meu email institucional para enviarem os trabalhos; avisei que, para os alunos que ainda não tinham terminado, a data limite de entrega seria até terça da semana seguinte, 29 de março. Também avisei os alunos que ainda não tinham entregue o consentimento para o teste CTt que o fizessem, caso contrário não divulgaria os resultados.

Comecei a leção da App ICQ. Para esta App, os alunos iriam começar a partir de um novo projeto Android do template *Empty Activity*, e construir a tela do zero.

Com o problema do endereço SDK resolvido, abri o emulador e demonstrei a estrutura da App: é composta por 3 layouts - 2 do tipo *constraint*, 1 do tipo *ScrollView*; o primeiro *layout* (cabeçalho) tem 1 rótulo, o segundo (corpo) tem 5 caixas de texto e 2 rótulos, e o terceiro (rodapé) tem 2 botões.

Para este turno, decidi adiar a explicação do funcionamento dos botões para a aula seguinte; não haveria tempo suficiente para começar a programá-los nesta aula.

Mostrei como construir o cabeçalho, alterando os atributos do rótulo Hello World inicial. Para o rodapé, mostrei como criar um novo *ConstraintLayout* e dimensioná-lo no canto inferior da tela, de modo a haver espaço para os dois botões.

Arrastei os dois botões da paleta para o rodapé, e disse-lhes para reconfigurarem o atributo de altura do rodapé para *wrap_content*, de modo a economizar espaço entre os botões e o corpo.

Construí o corpo, introduzindo um *ScrollView* com um *constraintLayout* dentro dele; expliquei que a rolagem do ecrã não deve incluir os botões – por isso usamos um *layout* do tipo *constraint* separado. Avisei que quem estiver com muita dificuldade em fazer os três layouts, pode construir apenas um, e que deve pôr nele o *ScrollView* de modo a possibilitar a rolagem do ecrã.

Para a orientação horizontal, expliquei como criar um ficheiro de *layout* ao seleccionar *Landscape* no botão de *Orientation Preview*; abri o separador *land/activity_main.xml*, e mostrei como economizar a altura do modo horizontal, e reposicionar os botões Calcular e Limpar lado-a-lado (Figura 62).

Avisei que este novo ficheiro de *layout* horizontal não é atualizado quando são feitas mudanças no ficheiro de *layout* retrato original.

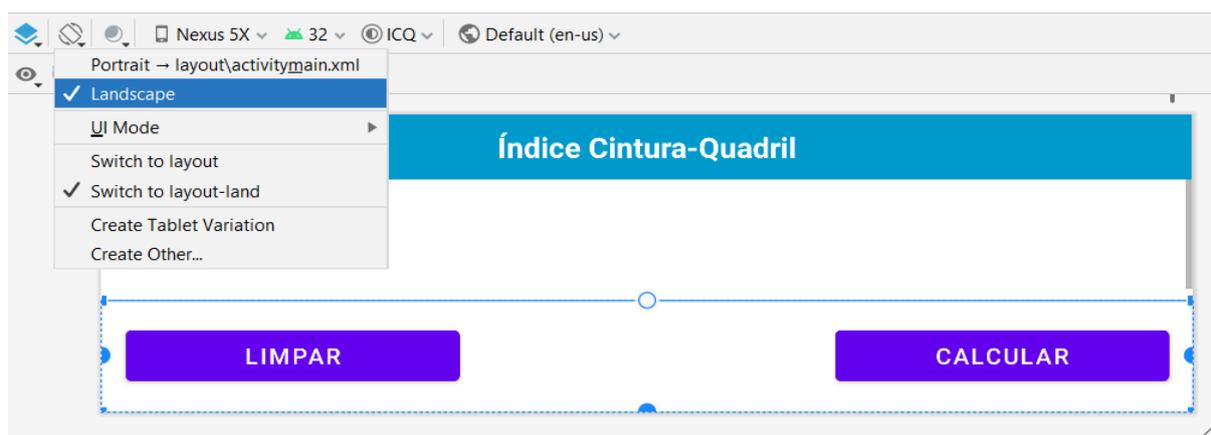


Figura 62. *Layout* da App ICQ em Orientação Horizontal

Por esta altura, a sessão estava quase a terminar, por isso dediquei-me a tirar dúvidas acerca dos *constraints* e do *ScrollView* até os alunos saírem.

Revido estas sessões, acho que não expliquei bem a estrutura de classes dos *ViewGroups*. Esta informação é desnecessária para um principiante. Basta saberem que o *ScrollView* não funciona com *Constraints*; o porquê disto é demasiado complexo. Também fiquei com a sensação de que o alinhamento das *Baselines* foi um passo puramente acessório; no entanto, é importante conhecer este atributo, pois se duas *Views* de texto adjacentes tiverem tamanhos de letra diferentes, sem este atributo elas ficam desalinhasadas.

No início da aula, um dos alunos não conseguiu baixar o ficheiro de projeto Android da página da disciplina; a internet estava muito lenta para baixar um ficheiro de 15 Mb. Transferi o projeto pessoalmente no PC do aluno através da minha *Pen-drive* USB.

Mais tarde, após pesquisar em casa, descobri que estava a disponibilizar os projetos Android para os alunos de maneira ineficiente: exportava o projeto, e fazia *upload* dessa pasta “zipada” na plataforma Nónio. Aprendi que existe um passo importante a ser feito antes de transferir um projeto: através do painel *Build > Clean Project*, o qual remove as pastas *Build*, reduzindo-o consideravelmente de tamanho (de 15Mb a aprox. 500Kb).

É possível que a causa do erro de localização do SDK tenha sido o facto de os ficheiros de projeto que eu disponibilizava não terem passado pelo *Clean Project*, e algumas das configurações mantiveram o endereço original do SDK do meu computador.

5.1.7. Sessão 7 – sexta-feira 25 de março 2022, Turno 1

Tabela 11. Planificação da Sessão 7

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	45 min	Formativa
Construir as telas da App ICQ	- ScrollView (rolagem) - Constraint Baselines (alinhamento)	Ativa. Aula prática demonstrativa e interrogativa. (como na aula seguinte)		
	Intervalo (9h50 – 10h00)		10 min	
Definir os tipos de valores de entrada	- Atributo InputType - Atributo Hint	Ativa. Aula prática demonstrativa e interrogativa. O professor demonstra uma sequência de passos para programar uma parte do código. Os alunos são solicitados a seguir os passos para as outras partes. O professor circula pela sala, esclarece dúvidas e observa a execução das tarefas.	15 min	Formativa
Preparar os botões Limpar e Calcular	- Ligações código-tela - Conversão de Doubles e Strings		30 min	
Programar o botão Calcular com a classe ICQ	- Instanciação de três objetos ICQ - Prevenção contra exceções - Método classificalCQ		45 min	

Sumário: Continuação da app ICQ. Ligações tela-código e classe-código.

Para esta sessão estava previsto completar a App ICQ de modo a poder começar na semana seguinte a leção de uma App com *Intents*. No entanto, muitos alunos não avançaram na construção das telas. Pensei em disponibilizar uma App ICQ com as telas já feitas, para aqueles que estavam a ter dificuldades; mas reconsiderarei, pois esta parte da disciplina contém critérios que contam para a avaliação final.

Retomei a leção do Turno 1 com a criação dos três layouts. Confesso que contava que alguns alunos tivessem terminado os elementos da tela em casa, mas nenhum fez qualquer avanço na App fora do tempo de aulas.

Expliquei a organização das caixas de texto em cadeia vertical. Arrastei cinco caixas *Plain Text*, selecionei-as e fiz *Create Vertical Chain*. As caixas de texto ligaram-se todas com *constraints* entre si; depois com *Center Horizontally*, ligaram-se entre as bordas da tela.

Defini o estilo para o espaçamento entre as caixas de texto como *Packed*, e a margem entre elas foi definida em grupo, com um filtro no painel Atributos para *layout_margin*, com o valor *16dp*.

Observei os alunos a construírem as telas. Reparei que poucos alunos implementaram os três layouts, e também poucos tinham implementado o *ScrollView*.

Passei à introdução dos novos atributos das caixas de texto - *InputType* e o *Hint*. Expliquei como os *Hints* são textos de indicação, que se apagam depois de digitar na caixa de texto. *Hints* poderiam até substituir o rótulos de *Nome*, *Sexo*, etc, mas, para esta App, servem como informação adicional.

Para a caixa de texto Nome, o *Hint* deve indicar que há um limite de 20 caracteres; para o Sexo, digitar apenas uma letra, M ou F, aceitando maiúscula ou minúscula; para o CC, um limite de 8 caracteres numéricos; para a Cintura e Quadril, deve indicar que a escala é em centímetros; e para o rótulo IMC, indica a fórmula "Cintura/Quadril", a qual será sobrescrita com o resultado.

Introduzi o atributo *InputType*. Mostrei que as caixas de texto podem ser configuradas para receberem diferentes tipos de dados. Para caracteres alfabéticos, inseri *inputType="text"* e deixei a lista de sugestões aparecer; dos vários tipos, selecionei "*textPersonName*". É possível concatenar configurações: por exemplo, com "*textPersonName/textCapWords*", testei no emulador e mostrei como a primeira letra de cada palavra é "capitalizada". Para limitar caracteres a 20, inseri *maxlength="20"*.

Parei a demonstração dos atributos e deixei os alunos configurarem as outras caixas de texto.

Passei à programação da *MainActivity*. Ao invés do IMC, comecei pelas ligações código-tela ao declarar os objetos *Widget*. Nesta parte, muitos alunos fizeram o esperado: copiaram o código do método *onCreate* do IMC e conformaram-no para o ICQ. Fizeram o mesmo para o método Limpar.

Passei para a programação do botão Calcular; comecei por importar a classe ICQ, e mostrei o código no projetor. Lembrei que a App ICQ não precisa de todos os valores de entrada (*inputs*) para efetuar o cálculo. Disse que devemos criar primeiro um construtor ICQ que receba apenas os *inputs* numéricos da Cintura e Quadril (Figura 63).

```
10 public class Icq {
11
12     public int cc;
13     public double cintura, quadril;
14     public String nome, sexo;
15
16     // construtor 2 parametros
17     public Icq(double cintura, double quadril){
18         this.cintura = cintura; this.quadril = quadril;
19     }
```

Figura 63. Classe ICQ: Construtor com 2 parâmetros

No manipulador Calcular, instanciei um objeto *icq2* com dois parâmetros (Figura 64); os valores do nome, sexo e CC são usados para outros objetos. Expliquei que este é o conceito de sobrecarga de construtores, e que para o ICQ utilizaríamos pelo menos dois construtores diferentes.

```

48 public void calcular(View view) {
49
50     Icq icq2 = new Icq(
51         Double.parseDouble(edtxtCintura.getText().toString()),
52         Double.parseDouble(edtxtQuadril.getText().toString()));
53
54     txtICQ.setText(String.format(Locale.US, format: "%.2f",
55         icq2.calculaIcq()));

```

Figura 64. Manipulador Calcular: Instanciação de Objeto icq2

Esprei os alunos programarem e fui circulando pela sala a tirar dúvidas.

Apercebi-me que a aula estava no fim e nem sequer tínhamos feito as verificações de caixas de texto vazias. Decidi então acelerar o passo para terminarmos pelo menos a instanciação do segundo objeto. Chamei a atenção para mostrar como instanciar *icq3*, o qual recebe os dois primeiros inputs do *icq2*, e um terceiro novo parâmetro, a variável String para o sexo ().

```

63     Icq icq3 = new Icq(
64         icq2.cintura, icq2.quadril,
65         edtxtSexo.getText().toString());

```

Figura 65. Instanciação de Objeto icq3 com Variáveis de icq2

Ao fazer as chamadas de variáveis do *icq2* como parâmetros do *icq3*, estamos a seguir um dos princípios do Java, conhecido com DRY (*Don't Repeat Yourself*). Disse aos alunos para programarem a o construtor de *icq3* adicionando à assinatura do *icq2*: *cintura, quadril, sexo*.

O tempo de sessão terminou e os alunos saíram.

5.1.8. Sessão 8 – sexta-feira 25 de março 2022, Turno 2.

Tabela 12. Planificação da Sessão 8

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	5 min	Formativa
Definir os tipos e valores de entrada	- Atributo Hint - Atributo InputType	Ativa. Aula prática demonstrativa e interrogativa. O professor demonstra uma sequência de passos para programar uma parte do código.	15 min	
Preparar os botões Limpar e Calcular	- Ligações código-tela - Prevenção contra exceções nulas	Os alunos são solicitados a seguir os passos para as outras partes.	25 min	
Programar o botão Calcular com a classe ICQ	- Instanciação de três objetos ICQ - Método <code>classicalCQ</code>	O professor circula pela sala, esclarece dúvidas e observa a execução das tarefas.	45 min	

Sumário: Continuação da app ICQ. Ligações tela-código e classe-código.

Nesta segunda sessão de sexta-feira, a Professora Altina Ramos e o colega Pedro Sequeira assistiram às minhas aulas.

Tal como na sessão anterior, a previsão era terminar a App ICQ, mas o ritmo dos alunos foi diferente do esperado, sendo que alguns estavam mais avançados que os outros, porém muitos ainda não tinham implementado o *ScrollView*. Esta parte da matéria não ficou bem consolidada.

Comecei por ensinar os novos atributos dos elementos *View* no código XML. Expliquei que o atributo *Hint* indica o tipo de dado a ser inserido na caixa de texto. Introduzi os atributos *inputType* para nomes dos utilizadores com *textPersonName* / *textCapWords* (Figura 66). Introduzi o atributo *maxLength*. Orientei os alunos a adicionarem um atributo para configurar as caixas de texto Cintura, Quadril e CC a aceitarem apenas caracteres numéricos – o *inputType="number"*.



Figura 66. Atributos *Hint* e *InputType*

Circulei pela sala a observar as telas. Houve um aluno com dúvidas sobre o espaçamento entre as caixas de texto. Depois de esclarecer, reparei que o aluno não tinha criado o rótulo Mensagem, por este não ser explicitamente visível na tela; lembrei-o deste elemento *View*, e quando tentou inseri-lo entre o

rótulo ICQ e o botão Calcular, a tela ficou com os *Constraints* desorganizados. Reparei também que o aluno não tinha implementado o *ScrollView*.

Os alunos copiaram as ligações código-tela do IMC para o projeto ICQ, e fizeram as alterações necessárias pôr código em conformidade.

Passei à programação do botão Calcular. O turno anterior serviu como indicação do nível de falta de compreensão, pelo que, decidi começar logo pela verificação de caixas vazias.

Comecei por explicar novamente este mecanismo de ‘filtro’ para prevenção de *inputs* nulos, antes de começar a explicar os construtores. O mecanismo consiste de uma série de *ifs* para cada objeto das caixas de texto, usando como condição o método *getText().toString().isEmpty()*, sempre que a condição for verdadeira (caixa vazia), três ações devem ser executadas: i) escrever mensagem a indicar o input nulo, ii) focar no input nulo, iii) terminar a função (Figura 67)

```
29 public void calcular(View view) {
30     if(edtxtCintura.getText().toString().isEmpty()){
31         txtMensagem.setText("cintura?");
32         edtxtCintura.requestFocus();
33         return;
34     }
35     if(edtxtQuadril.getText().toString().isEmpty()){
36         txtMensagem.setText("quadril?");
37         edtxtQuadril.requestFocus();
38         return;
39     }
40     if(edtxtSexo.getText().toString().isEmpty()){
41         txtMensagem.setText("sexo?");
42         edtxtSexo.requestFocus();
43         return;
44     }
}
```

Figura 67. Prevenção de *inputs* Nulos com *requestFocus*

Programei os *ifs* nos três inputs essenciais para um cálculo e uma classificação ICQ – cintura, quadril e sexo. Expliquei a utilidade do método *requestFocus*: o teclado virtual é aberto diretamente na caixa de texto indicada, pronto a digitar. Disse para aplicarem este mecanismo nas caixas de texto restantes; deste modo, nenhum valor nulo passa para a instanciação do objeto ICQ.

Passei à instanciação de três objetos ICQ: *icq2*, *icq3* e *icq5*. Disse aos alunos que basta o objeto *icq2* para chamar o método Calcular; para o método Classificar, é necessário o *icq3*. O terceiro objeto *icq5*, deverá utilizar um construtor com cinco *inputs*: os dois parâmetros vem do *icq2*, o terceiro parâmetro vem do *icq3*, e os quarto e quinto parâmetros vêm dos objetos *Nome* e *CC* (Figura 68).

```

55     Icq icq2 = new Icq(
56         Double.parseDouble(edtxtCintura.getText().toString()),
57         Double.parseDouble(edtxtQuadril.getText().toString()));
58
59     Icq icq3 = new Icq(
60         icq2.cintura, icq2.quadril,
61         edtxtSexo.getText().toString());
62
63     txtICQ.setText(String.format(Locale.US, format: "%.2f", icq3.icq));
64
65     Icq icq5 = new Icq(
66         icq2.cintura, icq2.quadril, icq3.sexo,
67         edtxtNome.getText().toString(),
68         Integer.parseInt(edtxtCC.getText().toString()));
69
70     txtMensagem.setText(icq5.classificaICQ());
71 }

```

Figura 68. Manipulador Calcular: Instanciação de 3 Objetos ICQ

Não haveria tempo para voltar a explicar o *String.format*. Circulei pela sala a ver os programas dos alunos; corriji alguns erros na parte de verificação de caixas vazias, mas muitos tiveram dificuldades na sobrecarga de construtores.

A sessão terminou e os alunos saíram.

A pouca compreensão sobre o código do ICQ surpreendeu-me. Julguei que os conceitos Java do IMC estavam consolidados e, pela semelhança das Apps, supus, incorretamente, que a progressão, de uma para outra, seria fácil e rápida.

5.1.9. Sessões 9 e 10 – terça-feira 29 de março 2022, Turnos 2 e 1

Tabela 13. Planificação das Sessões 9 e 10

Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
TURNO 2				
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	10 min	Formativa
Rever conceitos	- InputType - getText.toString.isEmpty	Ativa. Aula prática demonstrativa e interrogativa.		
Programar o botão Calcular para a classe ICQ	- Sobrecarga de construtores ICQ - Método calcular	O professor demonstra uma sequência de passos para programar uma parte do código.	40 min	
Programar o método classifica	- Método classifica - Conversão de variáveis String e Double	O professor circula pela sala, esclarece dúvidas e observa a execução das tarefas.	40 min	
	Intervalo (9h50 – 10h00)		10 min	
Testar a App ICQ	- Teste de inputs nas caixas de texto - Uso do Debugger (depurador) para um caso de erro.	Ativa. Aula prática demonstrativa e interrogativa. (como antes do intervalo)	40 min	Formativa
Terminar a App ICQ e receber os trabalhos	- Entrega dos trabalhos por email.		5 min	Sumativa
TURNO 1				
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	10 min	Formativa
Rever conceitos das aulas anteriores	- Atributo InputType - getText.toString.isEmpty	Ativa. Aula prática demonstrativa e interrogativa.		
Programar o botão Calcular para a classe ICQ	- Sobrecarga de construtores ICQ - Método calcular	(como no turno anterior)	35 min	
	Intervalo (11h30 – 11h40)		10 min	
Programar o método classifica	- Método classifica - Conversão de variáveis String e Double	Ativa. Aula prática demonstrativa e interrogativa. (como antes do intervalo)	30 min	Formativa
Testar a App ICQ	- Teste de inputs nas caixas de texto. - Uso do Debugger (depurador) para um caso de erro.		10 min	
Terminar a App ICQ e receber os trabalhos	- Entrega dos trabalhos por email.		5 min	Sumativa

A Sessão 9 foi composta por 3 aulas de 45 minutos: a 1ª aula consistiu na conclusão do botão Calcular; a 2ª aula abrangeu a conclusão do método Classificar; e a 3ª aula envolveu o teste da App e a sua entrega por email.

Sumário: Conclusão e teste da app ICQ. Entrega dos trabalhos IMC e ICQ.

As duas sessões deste dia decorreram com ritmo muito semelhante, com praticamente os mesmos conteúdos: tirar dúvidas sobre o projeto ICQ. No tempo extra do Turno 2 (45 minutos), a App ICQ foi testada e corrigimos vários erros de programação – desde a conversão de variáveis até aos construtores.

Para os dois turnos, comecei por avisar que alguns alunos ainda não tinham entregado o trabalho IMC, nem o consentimento para o teste CTt. Mostrei no projetor uma folha de cálculo listando os alunos cujos trabalhos e assinaturas faltava entregar.

Um dos alunos a quem faltava entregar reclamou que já tinha enviado o IMC, então mostrei no projetor o *Inbox* do meu email institucional, provando como não tinha recebido nada. O aluno disse que ‘achou estranho’; depois acessou o seu email e reenviou o trabalho IMC. Dei então início à lecionação, que ficou pela parte do método Classifica da classe ICQ.

Este método Classifica pode ser feito através de *ifs*, ou através da condicional *Switch* (Figura 69).

```
29 String classificaICQ() {
30     switch (sexo) {
31         case "M": case "m":
32             if (icq > 1) { return "Acima do recomendado. Masculino: 1,00"; }
33             else { return "Dentro do recomendado. Masculino: 1,00"; }
34         case "F": case "f":
35             if (icq > 0.8) { return "Acima do recomendado. Feminino: 0.80"; }
36             else { return "Dentro do recomendado. Feminino: 0.80"; }
```

Figura 69. Classe ICQ: Método Classifica

À medida que fui verificando os programas dos alunos, percebi que a maioria não tinha feito uma boa adaptação entre o código do botão Calcular e as classes importadas ICQ do período passado. Em geral, os alunos fizeram poucas mudanças necessárias ao código, e só alguns implementaram mais de um construtor ICQ. Isto levou a vários problemas, de modo que a App não desempenhou a função Classifica e, em alguns casos, nem sequer o método Calcula. Observei os programas dos alunos dos dois turnos, e, apesar das classes *MainActivity* estarem relativamente conformes – provavelmente devido aos alunos copiarem entre si – a maioria das classes estavam bastante diferentes.

Expliquei os erros de código individualmente a cada aluno; frequentemente, não efetuavam as verificações dos inputs antes de construir objetos ICQ. Os alunos não tinham uma noção clara de como os construtores funcionam.

A sessão prolongou-se com retrospectivas dos conceitos das aulas anteriores. Para o Turno 2, depois do intervalo, fiz uma demonstração de como fazer uma depuração de exceções nulas, através do *Debugger* (Figura 70). Não tive tempo de demonstrar o *Debugger* para o Turno 1.

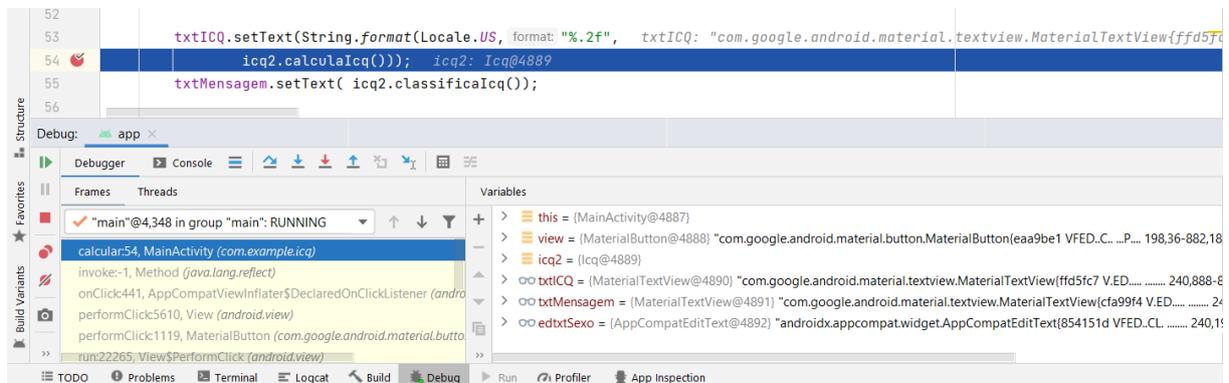


Figura 70. Demonstração do Debugger

Antes de a aula terminar, solicitei a todos que enviassem os trabalhos ICQ por email. Os alunos dos dois turnos enviaram os seus trabalhos e saíram. Mesmo assim, ao chegar em casa, verifiquei que faltavam os trabalhos de dois alunos.

5.1.10. Sessões 11 e 12 – sexta-feira 1 de abril 2022, Turnos 1 e 2

Tabela 14. Planificação das Sessões 11 e 12

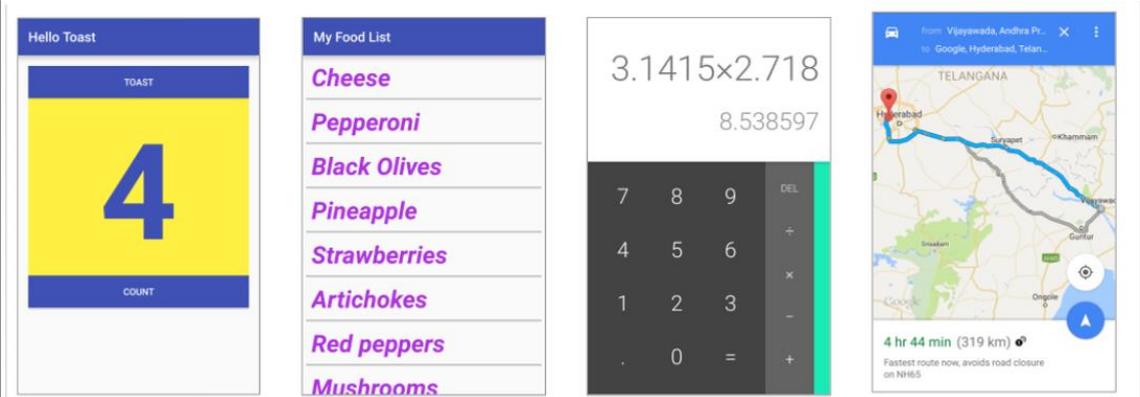
Objetivos	Conteúdos	Metodologia e Atividades	Tempo	Avaliação
TURNO 1				
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	10 min	Formativa
Expor Atividades e Intentos	- Classe Activity - Intentos Explícitos e Implícitos - Extras	Ativa. Aula prática demonstrativa e interrogativa. O professor demonstra uma sequência de passos para programar uma parte do código. O professor circula pela sala, esclarece dúvidas e observa a execução das tarefas.		
Demonstrar App DuasAtividades-Levar	- Elementos View - Funcionamento dos botões		5 min	
Programar o botão Enviar Sem Dados	- Instanciação de um Intent - Método startActivity		15 min	
Programar o botão Enviar Com Dados	- Método putExtra		15 min	
	Intervalo (9h50 – 10h00)		10 min	
Programar o botão Enviar Com Dados	- Método getIntent - Verificação de caixa de texto vazia	Ativa. Aula prática demonstrativa e interrogativa (como antes do intervalo)	45 min	Formativa
Compilar dados raciocínio lógico	- Aplicação do 2º Teste de Pensamento Computacional	Teste de Escolha Múltipla, através do Google Forms.	40 min	Sumativa
Obter avaliação da intervenção	- Aplicação do questionário de avaliação	Teste de Escolha Múltipla, anónimo, em papel.	5 min	
TURNO 2				
Controlar o comparecimento	- Chamada de presença - Sumário da aula	Interrogativa e expositiva.	10 min	Formativa
Expor Atividades e Intentos	- Classe Activity - Intentos Explícitos e Implícitos - Extras	Ativa. Aula prática demonstrativa e interrogativa (como no turno anterior).		
Demonstrar App DuasAtividades-Levar	- Elementos View - Funcionamento dos botões		5 min	
Programar os botões Enviar Sem Dados e Com Dados	- Instanciação de um Intent - Método startActivity - Método putExtra		30 min	
Compilar dados raciocínio lógico	- Aplicação do 2º Teste de Pensamento Computacional	Teste de Escolha Múltipla, através do Google Forms.	40 min	Sumativa
Obter avaliação da intervenção	- Aplicação do questionário de avaliação	Teste de Escolha Múltipla, anónimo, em papel.	5 min	

Sumário: Conceito de Intento (Intents). Início da app DuasAtividades. 2º Teste de Pensamento Computacional. Questionário de Avaliação do Prof. Estagiário.

No última dia, os conceitos a serem ensinados foram os *Intents* – que são métodos para comunicação entre duas ou mais *Activities*.

A Professora Supervisora Altina Ramos e o meu colega Pedro Sequeira estiveram presentes nas aulas do T2. Houve três faltas no T1, duas no T2; e um aluno do T2 chegou 15 minutos atrasado.

Comecei por recordar como aprendemos a criar Apps com apenas uma *Activity*. O próximo passo seria aprender a gerir a ligação entre duas *Activities* numa App (Figura 71).



Uma aplicação pode ter várias *Activities*.

Por exemplo., uma app de **Email** pode ter:

- uma *Activity* para mostrar a **lista** de novos emails;
- uma *Activity* para **escrever** um email;
- uma *Activity* para **ler** mensagens individuais;

Figura 71. Diapositivo sobre *Activities*

Expliquei como a *Activity* principal – *MainActivity* – é a que aparece na tela quando o utilizador abre uma App, e que, embora as *Activities* de uma App funcionem umas com as outras, cada *Activity* é independente das outras; pode-se iniciar uma *Activity* numa App “A” e permitir que outra App “B” inicie atividades usando os dados da App “A”.

Dei exemplos de Apps contendo várias *Activities*: uma para tirar fotos com a câmara do telemóvel, e outra para enviar as fotos e-mail; Disse que a comunicação entre *Activities* na mesma App é feita através dos *Intents*.

Expliquei a diferença entre *Intents*: no *Intent* implícito, a atividade de recebimento não é especificada; declara-se uma ação geral a ser executada e o sistema procura localizar uma atividade que consiga lidar com a ação solicitada; enquanto que, no *Intent* explícito, a atividade de recebimento é especificada, pelo nome da classe da atividade. Em geral, os intents explícitos são usados para iniciar componentes no próprio aplicativo, por exemplo, para mover entre telas na interface do utilizador.

Demonstrei no emulador a App *DuasAtividades_Levar* (Figura 72): consiste de duas telas, uma para a atividade principal *MainActivity*, e outra para uma segunda atividade, *SecondActivity*. O utilizador só visualiza uma tela de cada vez, ambas têm um cabeçalho de cor diferente.

No centro, a *MainActivity* tem uma caixa de texto, e a *SecondActivity* tem um rótulo.

No rodapé, a *MainActivity* tem dois botões: 'Enviar com Dados' e 'Enviar sem Dados'; ao tocar no 'Enviar com Dados', a App abre a tela da *SecondActivity*, e o conteúdo da caixa de texto é copiado para o rótulo da *SecondActivity*. Ao tocar no 'Enviar sem Dados', o rótulo da segunda tela mantém-se igual.



Figura 72. App *DuasAtividades_LevarDados*

A *SecondActivity* tem um botão no rodapé chamado 'Voltar'; ao ser tocado, tal como o botão 'Enviar sem Dados', apenas muda de tela.

Os alunos baixaram o projeto "DuasAtividades_layout", da página do Nónio; o projeto continha apenas as telas feitas. Tal como na primeira aula da App IMC, o objetivo era programar os botões da App com o código Java.

Comecei por programar o botão Enviar Sem Dados, cuja função serve como conceito de base para esta sessão: usar um *Intent* para iniciar (*start*) outra atividade. Expandi o painel do projeto, e mostrei que esta App tem dois ficheiros XML na pasta de recursos: *activity_main* e *activity_second*.

No ficheiro *activity_main*, criei o manipulador do botão Enviar Sem Dados com o atributo *onClick*. Mostrei como adicionar um *Intent* explícito à classe principal, ao instanciar um novo objeto *Intent* dentro do manipulador *enviarSemDados* (Figura 73)

```
24 public void enviarSemDados(View view) {
25     Intent irParaSecondAct = new Intent( packageContext: this, SecondActivity.class);
26     startActivity(irParaSecondAct);
27 }
```

Figura 73. Instanciação de um Objeto *Intent*; Método *startActivity*

Expliquei o construtor do *Intent*, tem dois parâmetros: um de Contexto, que permite acesso a outras classes no mesmo domínio, e outro de Classe, que explicita a classe com que o *Intent* vai trabalhar. Neste caso, o Contexto é o domínio do projeto (*this*), e a classe é a *SecondActivity*. Aconselhei nomear os *Intents* especificando o que eles fazem (*irParaSecondAct*).

Chamei o método *startActivity* para o *Intent* criado, e testei no emulador. Cliquei no botão Enviar Sem Dados, e o ecrã mudou para a tela da *SecondActivity*. Perguntei aos alunos “Como faço para voltar à *MainActivity*, sem ser pelo botão de navegação para voltar atrás?”. Disse que o “voltar atrás” trata-se de terminar a atividade *SecondActivity*. Expliquei que o *Intent* de *irParaSecondActivity* inicia uma nova atividade, mas a atividade inicial *MainActivity* ainda estava a correr na aplicação.

No ficheiro de *layout activity_second*, inseri o atributo *onClick* no botão Voltar, e automaticamente fui para o ficheiro de classe correspondente, *SecondActivity*. Inseri o método *finish* no manipulador Voltar (Figura 74). Testei o botão no emulador, “voltando” para a tela da *MainActivity*.

```
15 public void voltar(View view) {
16     finish();
17 }
```

Figura 74. Manipulador Voltar: Método *finish*

Disse a todos para programarem o *Intent* nos seus projetos. Dediquei alguns minutos a observar o código dos alunos.

Passei à programação dos *Extras*. Expliquei que a classe *Intent* inclui chaves *Extras* como constantes que começam com prefixo *EXTRA*; por exemplo, pode-se usar *Intent.EXTRA_EMAIL* para indicar endereços de e-mail como *Strings*. Disse que devemos definir as nossas próprias chaves, convencionalmente como variáveis estáticas, começando com o prefixo *EXTRA_*. Para garantir que a chave seja exclusiva, o valor da *String* da chave deve conter o domínio todo da aplicação (Figura 75).

```
10 public class MainActivity extends AppCompatActivity {
11
12     public static final String EXTRA_DADOS = "com.example.duasatividades_levardados.extra.DADOS";
```

Figura 75. Declaração da Chave *Extra_Dados* com Domínio

Criei um *EditText* chamado *edtxtMain* para a caixa de texto da *MainActivity*, e fiz a ligação código-tela no *onCreate*.

Com um objeto *EditText* para receber os dados e uma chave *Extra* para transferi-los, passei à programação do segundo botão, Enviar com Dados. Tal como no Enviar sem Dados: criei um *Intent* para iniciar a classe *SecondActivity*. Antes de a iniciar, chamei o método *putExtra* do *Intent* criado, com dois parâmetros: a chave *Extra*, que armazena os dados a serem transferidos, e a variável *String*, a ser armazenada (Figura 76).

```
28     public void enviarComDados(View view) {
29         Intent irParaSecondAct = new Intent( packageContext: this, SecondActivity.class);
30         irParaSecondAct.putExtra(EXTRA_DADOS, edtxtMain.getText().toString());
31         startActivity(irParaSecondAct);
32     }
```

Figura 76. Manipulador Enviar Com Dados: Método *putExtra*

Na classe *SecondActivity*, criei um rótulo (*TextView*) para mostrar a *String* da chave *Extra_Dados* e fiz a sua ligação com o elemento *View*.

Para receber um *Intent* com dados, é preciso criar um *Intent* que os receba; instanciei-o dentro do *onCreate* com o método *getIntent*, com o nome *intentMainAct*. (Figura 77)

```
15     TextView rtlDados;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_second);
21         rtlDados = findViewById(R.id.second_rtl_dados);
22
23         Intent intentMainAct = getIntent();
```

Figura 77. Receção do *Intent* com Método *getIntent*

Criei uma *String* auxiliar chamada *dadosMain* para buscar a *String* do *Extra_Dados*. Chamei o método *getStringExtra* do *intentMainAct*, e nele chamei a *Extra_Dados* da *MainActivity*. Finalmente, usei o método *setText* do rótulo para escrever a *String* *dadosMain* (Figura 78). Testei a App no emulador com uma mensagem “hello” na *MainActivity*, e esta apareceu na *SecondActivity*.

```
25     String dadosMain = intentMainAct.getStringExtra(MainActivity.EXTRA_DADOS);
26     rtlDados.setText(dadosMain);
```

Figura 78. Método *getStringExtra*

Disse para programarem os *Intents* e os *Extras*, e circulei pela sala a observar os programas.

Este foi o ponto final da leção em comum aos dois turnos. No dia 01/04, o T1 tinha mais tempo de aula, então, continuei a leção dos *Intents* por mais uma aula de 45 minutos:

A App deve verificar se a *Extra_Dados* está vazia. Estando, deve enviar uma mensagem para o utilizador a dizer “Sem Dados Recebidos”.

Desta vez, insisti que os alunos do Turno 1 revissem os conceitos anteriores para resolver as questões atuais; disse-lhes para usarem o método *isEmpty* para o caso da chave estar vazia, e um *else* em caso contrário. Sugeri que utilizassem a mesma lógica de verificação de caixas de texto vazias usada nas Apps IMC e ICQ.

Apercebi-me que havia alunos com diferentes tipos de dificuldades. Alguns tinham compreendido que o mecanismo de verificação é basicamente *If – Intent – getString – isEmpty*, mas não sabiam que método certo para buscar *Strings* é *getStringExtra*. Outros não tinham percebido que a verificação deve ser feita na *SecondActivity*, e tentaram aplicar o método *getString* no *EditText* da *MainActivity*. Só os melhores alunos escreveram o código do mecanismo *if* de forma correta (Figura 79) depois de terem pesquisado no *Codelabs Android Fundamentals*.

```
27 if(intentMainAct.getStringExtra(MainActivity.EXTRA_DADOS).isEmpty()){
28     rtlDados.setText("Sem Dados Enviados");
29 } else {
30     String dadosMain = intentMainAct.getStringExtra(MainActivity.EXTRA_DADOS);
31     rtlDados.setText(dadosMain);
32 }
```

Figura 79. Verificação de Extras Vazios Incompleto

Houve até um aluno muito bom que foi criativo e aplicou um *Toast* em vez de *setText*! (Figura 80)

Depois de observar o progresso de alguns, revelei que o código *if* que eles tinham implementado tinha resolvido um problema, mas criado outro – a App passou a ‘estourar’ ao clicar-se no botão Enviar Sem Dados.

```
27 if(intentMainAct.getStringExtra(MainActivity.EXTRA_DADOS).isEmpty()){
28     Toast.makeText( context: this, text: "Sem dados enviados", Toast.LENGTH_LONG).show();
29 } else {
30     String dadosMain = intentMainAct.getStringExtra(MainActivity.EXTRA_DADOS);
31     rtlDados.setText(dadosMain);
32 }
```

Figura 80. Verificação de Extras Vazios com Mensagem *Toast*

Expliquei que é uma exceção: o método *OnCreate* está a verificar se a chave *Extra_Dados* tem a *String* vazia, antes de verificar se a própria chave existe. A linha do *else* declara e busca uma *String* de um *Extra*, mas quando clicamos no Enviar Sem Dados, o *Intent* cria a atividade *SecondActivity*, mas não cria uma chave estática com o *putExtra*.

Por isso, é preciso verificar primeiro se a chave Extra existe no *Intent*. Para isso, cria-se previamente um *if* que chama o sub-método *hasExtra* (Figura 81).

```
26  if(intentMainAct.hasExtra(MainActivity.EXTRA_DADOS)){
27      if(intentMainAct.getStringExtra(MainActivity.EXTRA_DADOS).isEmpty()){
28          rtlDados.setText("Sem Dados Enviados");
29      } else {
30          String dadosMain = intentMainAct.getStringExtra(MainActivity.EXTRA_DADOS);
31          rtlDados.setText(dadosMain);
32      }
33  }
```

Figura 81. Verificação de Extras Vazios com o Método *hasExtra*

Em 01/04, apliquei o segundo teste CTt na última aula de cada turno. Orientei os alunos para acederem aos emails e entrarem no Google Forms. O conteúdo foi o mesmo do primeiro teste, porém, troquei a ordem das questões e das escolhas múltiplas: as questões ímpares foram trocadas pelas questões pares, e as opções A e C foram trocadas pelas B e D. O teste realizou-se dentro da normalidade.

Terminado o CTt, entreguei a cada aluno o inquérito, anónimo, em papel, “Questionário de Avaliação da Qualidade do Professor Estagiário, do Curso e Autoavaliação do Estudante” (Apêndice IV). Recolhi os questionários dos dois turnos no mesmo envelope, mantendo o anonimato. Os resultados, compilados e analisados, apresento no tópico 5.2.4.

5.2. Resultados e Análise

A intervenção não sumariada que, no capítulo 4, classifiquei como “reuniões-entrevistas”, e, na introdução do capítulo 5, como tutoria/reunião/entrevista, no presente tópico é denominada apenas como “entrevistas”. Os formulários criados para a recolha de dados estão disponíveis nos Apêndices.

5.2.1. Grelha de Observação dos Alunos

Os resultados de observação dos alunos foram compilados por turno (Figuras 82-86; “NA” corresponde a “Não se Aplica”, e “?” indica “Não consegui observar”). Para os tópicos assiduidade & pontualidade, e comportamento, utilizei as observações das aulas sumariadas e das entrevistas. Nos tópicos participação, empenho, e autonomia, incluí apenas as aulas sumariadas.

No que respeita à assiduidade e pontualidade (Figura 82), considerando a totalidade dos alunos e as faltas totais, e não um aluno em particular, verifiquei, para o T1, um comparecimento médio a 87% das aulas, portanto, 13% de faltas (F), enquanto, para o T2, o destaque foram as faltas de pontualidade (FP), ou seja, os alunos chegaram atrasados, ou no início da aula ou depois dos intervalos, 16% das vezes, em média.

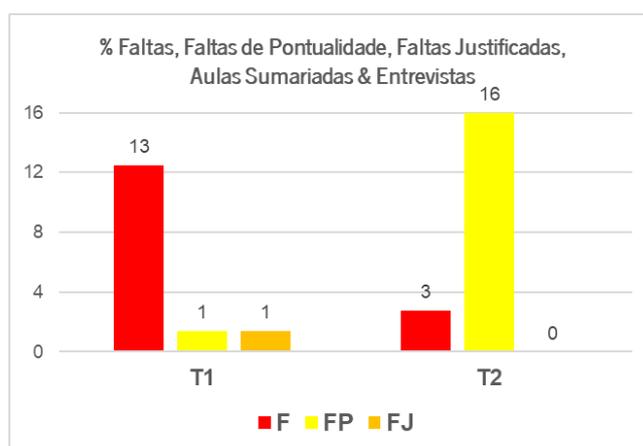


Figura 82. Assiduidade e Pontualidade, Turnos 1 e 2

Considereei que ambos os turnos tiveram um comportamento adequado nas aulas sumariadas e nas entrevistas (Figura 83), embora um dos turnos fosse menos disciplinado que o outro. Coincidentemente, ou não, no turno menos disciplinado, havia um ambiente de camaradagem e conversa e partilha de informação entre os alunos, durante as aulas; nos intervalos, formavam grupos de 4 a 6 alunos e às vezes reunia-se o turno todo. No turno mais disciplinado, os alunos interagiram mais com o professor; nos intervalos das aulas, formavam grupos pequenos, de 2 a 3 alunos, havendo também alunos solitários.

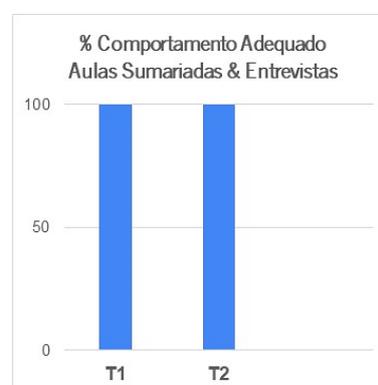


Figura 83. Comportamento, Turnos 1 e 2

A maioria dos alunos fez perguntas durante as aulas sumariadas; considerei que sim (S), isso é participar (Figura 84).



Figura 84. Participação, Turnos 1 e 2

Relativamente ao empenho, diante da grande quantidade de trabalhos ICQ incompletos, atribuí sim (S) apenas aos 2 alunos que assinalaram “Tentei fazer os trabalhos IMC e ICQ em casa”, na autoavaliação (tópico 5.2.5.), ambos do T1 (Figura 86). Relativamente à autonomia, todos os alunos assinalaram que precisaram de algum tipo de ajuda, do professor ou dos colegas, para realizar os trabalhos, na autoavaliação (5.2.5). Assim, considerei que “N” não desenvolveram autonomia (Figura 85).

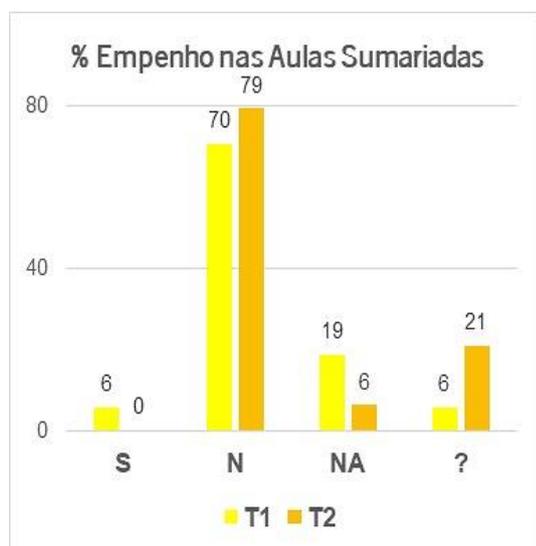


Figura 86. Empenho, Turnos 1 e 2

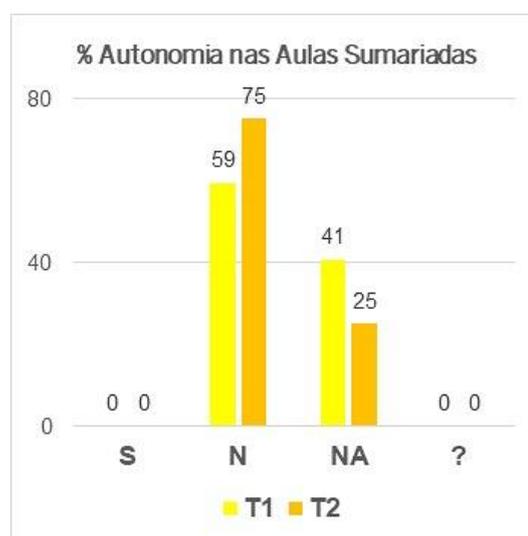


Figura 85. Autonomia, Turnos 1 e 2

Em resumo, pareceu-me que o conjunto da turma caminhou positivamente no sentido de atingir as competências associadas ao PASEO, mas as componentes de empenho e estudo autónomo precisam ser reforçadas.

5.2.2. Apps IMC e ICQ

IMC

A adesão à avaliação do trabalho IMC foi de 76%, ou seja, 13 dos 17 alunos enviaram-me as suas Apps. Os critérios de correção que adotei incidiram sobre os métodos Java das classes *TextView* e *EditText*, nomeadamente *setText*, *getText*, *toString*, *isEmpty*, *requestFocus*, e de conversão *Double.parseDouble*, *String.format*. Para além do Java, com 8 valores (v), atribuí pontuação à autoria do trabalho (correspondência da App com o trabalho Java do 1º período), 8v, e à participação (envio do trabalho), 4v. O gráfico dos resultados (Figura 87) foi construído em ordem decrescente de percentagem de acertos; a participação não consta no gráfico porque todos os que enviaram tiveram 4v. Para facilitar a leitura, denominei “Funciona”, aos métodos relacionados com escrita e conversão, e “Não estoura com inputs nulos” aos da verificação, em Android. Os resultados mostraram que menos de 50% dos trabalhos usaram os seus códigos do 1º período (*MainActivity* original e IMC original) e mais de 50% realizaram corretamente as tarefas para IMC em Android. Numa escala de 20v, as notas situaram-se entre 8 e 16v, sendo a média 11,38 e 12 a mediana e moda (n=13; os 4 alunos que não enviaram a App não entraram no cálculo).

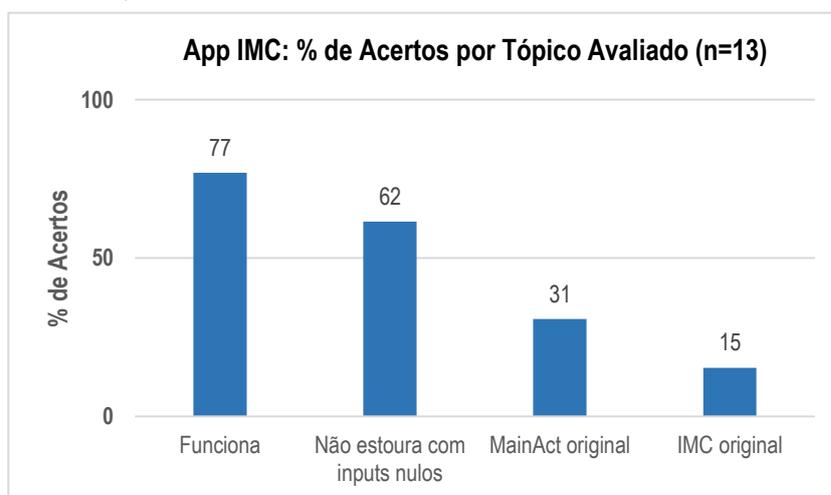


Figura 87. App IMC: % de Acertos por Tópico Avaliado (n=13)

Aquando das entrevistas, perguntei-lhes porque não tinham usado os códigos anteriores. As respostas foram diversas: uns disseram que foi por falta de atenção, outros que não sabiam que era preciso, outros que perderam o código, outros que naquela altura ainda estavam a aprender Java e os trabalhos não estavam corretos, etc. Argumentei que se, atempadamente, me tivessem dito que não tinham os seus códigos originais, eu poderia ter imaginado uma forma de dar a volta ao assunto.

ICQ

No trabalho ICQ, a adesão dos alunos subiu para 15 em 17 (88%), apesar de ser mais complexo. Para a correção adotei 12 tópicos, dos quais 8 sobre métodos Java e 4 sobre atributos XML, sendo a distribuição da pontuação 15v e 5v respectivamente. A tabela 15, organizada em ordem decrescente de acertos, mostra a relação entre a funcionalidade e os métodos/atributos, assim como a pontuação de cada tópico.

Tabela 15. Critérios de correção da App ICQ. Funcionalidade - Método/Atributo – Pontuação

Critério	Requisitos de Método() Java / <Atributo> XML	Pontuação
Caixas de texto: Hints	<EditText>: atributo <i>hint=""</i>	1
Botão Limpar: funciona	<Button>: atributo <i>onClick="limpar/calcular"</i> Limpar(): objetos <i>TextView</i> e <i>EditText</i> utilizam o método <i>setText</i>	1
Caixas de texto e rótulos: corretos	Tela contém todos os <i>TextViews</i> , <i>EditTexts</i> e <i>Buttons</i> necessários organizados com <i>ConstraintLayouts</i>	1
calculaCQ: não estoura com caixas vazias	Calcular(): objetos <i>EditText</i> utilizam o método <i>getText</i> , <i>toString</i> , <i>isEmpty</i> para proteger contra exceções nulas	2
calculaCQ: requestFocus	Proteções contra exceções nulas: utilizam o método <i>requestFocus</i> , o qual direciona o teclado virtual para o <i>EditText</i> indicado	1
calculaCQ: funciona com sexo	Calcular(): instancia objeto ICQ com pelo menos os valores de <i>Quadril</i> , <i>Cintura</i> e <i>Sexo</i> , chama o método de cálculo, e escreve o resultado no rótulo ICQ	5
classicalCQ: funciona corretamente	Calcular(): instancia objeto ICQ com pelo menos os valores de <i>Quadril</i> , <i>Cintura</i> e <i>Sexo</i> , chama o método de classificação, e escreve o resultado no rótulo Mensagem	2
CC, Cintura e Quadril: teclado numérico	<EditText>: atributo <i>inputType="number"</i>	1
Orientação horizontal: rolável	<ScrollView>: <i>ViewGroup</i> que contém um segundo <i>ConstraintLayout</i> Necessita de um ficheiro <i>activity_main.xml(land)</i>	1
classicalCQ: aceita sexo com minúsculas "m/f"	Classifica(): condicional <i>if / switch-case</i> com operador "ou" ()	2
calculaCQ: funciona sem sexo	Calcular(): instancia objeto ICQ com pelo menos os valores de <i>Quadril</i> e <i>Cintura</i> , e sem o valor do <i>Sexo</i> , chama o método de cálculo, e escreve o resultado no rótulo ICQ	1
classicalCQ: calcula Cintura em excesso	Classifica(): instancia objeto ICQ, chama o método de classificação, e retorna um número calculado da diferença entre a cintura atual e a cintura ideal, dependendo do valor da String do <i>Sexo</i>	2

O gráfico dos resultados também foi construído em percentagem decrescente de acertos relativamente à funcionalidade (Figura 88). Como pode ver-se no gráfico, 7 dos 12 critérios tiveram mais de 50% de acertos. Dos 5 critérios com menos de 50% de acertos, 4 eram métodos Java e 1 era atributo XML. Os tópicos com menos acertos eram, efetivamente, os mais complexos, ou seja, a ordem da percentagem de acertos foi coerente com a ordem de dificuldade do trabalho. Numa escala de 20 valores, o intervalo das notas ficou entre 3 e 14v, a moda foi 14 (n=4) e a mediana 12, mas a nota média baixou para 10,13 porque, dos 15 trabalhos avaliados, 3 Apps estavam muito incompletas, pontuaram apenas 3 valores, e baixaram a média da turma. Durante as entrevistas, estes 3 alunos disseram não ter tomado conhecimento de que era possível melhorar o trabalho e entregá-lo depois das férias da Páscoa (não têm o hábito de consultar o email institucional). De referir que os trabalhos com nota 14 estavam muito semelhantes, e os 4 alunos são colegas próximos. Ao longo das minhas aulas, pareceu-me que, em

geral, os alunos do T1 encontravam mais facilmente as soluções aos problemas de programação propostos; no entanto, no decorrer da correção, sucedeu que os trabalhos mais completos da App ICQ, foram os do T2 – e por isso tiveram melhores notas. Aquando das entrevistas individuais, mostrei a correção do ICQ e mencionei a existência de trabalhos MUITO semelhantes no T2, mas os alunos não deram seguimento aos meus comentários.

Os códigos-fonte das Apps IMC e ICQ estão disponíveis nos apêndices X e XI, respetivamente.

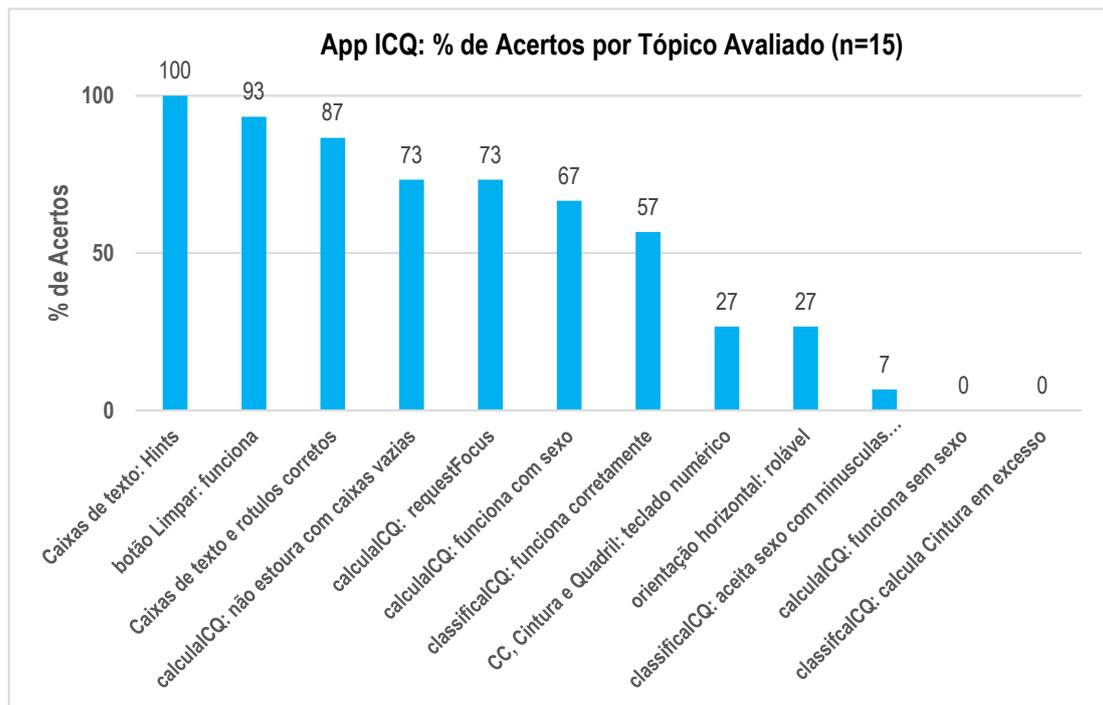


Figura 88. App ICQ: % de Acertos por Tópico Avaliado (n=15)

As notas individualizadas dos alunos do 11ºTPI nas Apps IMC e ICQ constam na Figura 89. Tendo em conta os 13 alunos que entregaram os 2 trabalhos, 10 tiveram nota média positiva, e 3 negativa. A nota média da turma foi 11,31 e a mediana 11,5; as modas foram 8,5; 11; 13, todas com n=2. Convertendo para percentagem, temos que 76,5% dos alunos do 11ºTPI entregaram os 2 trabalhos e, destes, 76,9% foram aprovados. Em resumo, 58,8% dos trabalhos do 11ºTPI foram aprovados. Uma vez que a % de aprovação foi superior a 50%, faço uma avaliação positiva da aprendizagem dos alunos nas Apps IMC e ICQ e considero que o objetivo (ii), Desenvolver a Criação de Apps para Android, foi alcançado.

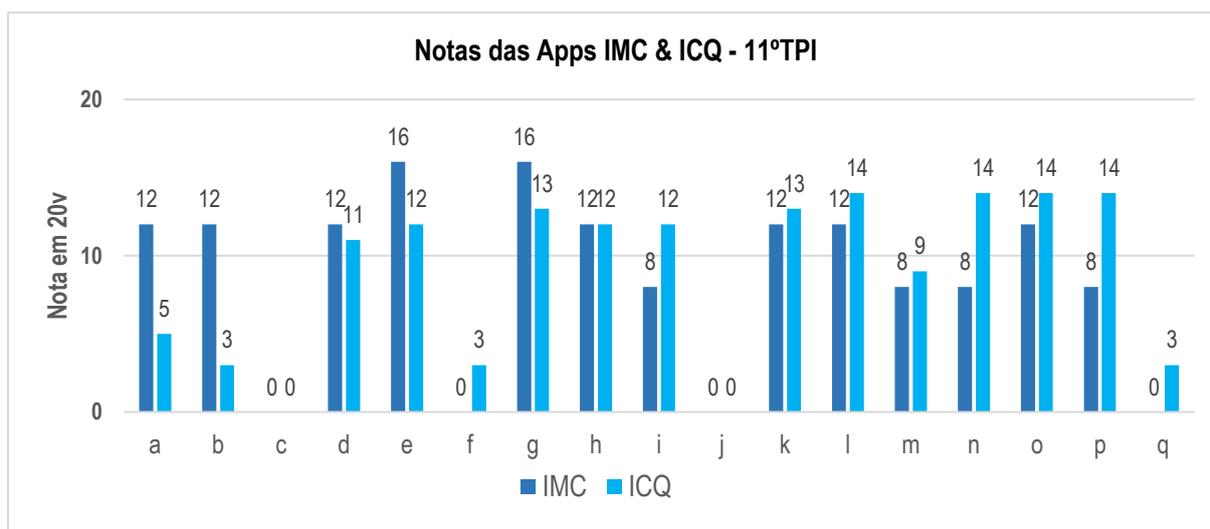


Figura 89. Notas das Apps IMC e ICQ - 11ºTPI

Nas entrevistas individuais, à pergunta *O que é mais difícil para ti no Android Studio?* Dos 16 alunos, 10 responderam, com as suas palavras, o código Java, 3 indicaram o XML, 2 não responderam claramente e 1 aluno referiu a complexidade do ambiente de desenvolvimento integrado Android Studio (Apêndice VIII). Perante as respostas dos alunos, considero que é correta a opção pelo Android Studio na leção de programação Android no CPPI, uma vez que é a ferramenta mais completa que se dispõe atualmente e os alunos não o mencionaram como um empecilho à aprendizagem.

5.2.3. Testes de Pensamento Computacional

A concretização do objetivo (iii) e da QIP estavam associadas ao CTt. Estive expetante quanto ao grau de adesão dos alunos ao teste, porque envolvia vários procedimentos ao longo de dois meses: era necessário que entregassem o Consentimento Informado aos seus Encarregados de Educação e mo devolvessem assinado; tinham de comparecer, com pontualidade, nos 2 dias de aplicação dos testes; tinham de responder a todas as perguntas dos testes; e por último, tinham de comparecer às entrevistas individuais. Felizmente, 88% (15 em 17) dos alunos aceitaram participar, e 73% (11 em 15) dos participantes cumpriram todos os procedimentos. Penso que esta atitude pode ser atribuída à motivação intrínseca dos alunos, à curiosidade de saberem o seu desempenho num teste de pensamento computacional. Contudo, ao longo das aulas, se o meu comportamento os tivesse levado a duvidar da validade do teste, talvez não teriam comparecido ao segundo teste, ou talvez teriam faltado à entrevista.

Em relação ao nível de proficiência dos alunos no teste CTt, eu não tinha uma ideia em mente, uma vez que os poucos estudos disponíveis não são diretamente comparáveis com o contexto da presente intervenção pedagógica. O teste CTt aplicado nos dias 15/03 e 01/04 foi o mesmo, porém, no segundo teste, troquei os números e/ou as alíneas das questões, relativamente ao primeiro. Por coincidência, ou não, sucedeu que, no T2, vários alunos tiveram menos acertos no segundo teste que no primeiro, enquanto, no T1, o número de acertos foi igual ou melhor no segundo teste.

A minha intervenção pedagógica teve, sim, intencionalidade no sentido de promover o desenvolvimento do pensamento computacional dos alunos. Não obstante, mesmo durante a fase de planeamento, já era facto assente que, num período tão curto, não era exetável conseguir detectar alterações no estágio de pensamento computacional. Assim, a primeira aplicação do CTt permitiu efetivamente recolher e tratar dados sobre o estágio de pensamento computacional dos alunos e, com isso, orientar as aulas no sentido de procurar colmatar as falhas detetadas (QIP). Por sua vez, a segunda aplicação do CTt, pelo pequeno intervalo de tempo decorrido desde a primeira, foi útil mais para verificar a repetibilidade, localizar falsos acertos (Barbosa, 2018, p.43). Não tenho conhecimento de estudos sobre o tempo necessário para alterar o nível de proficiência em pensamento computacional (CT) de alunos da escolaridade obrigatória. No caso de professores aprendizes deste conteúdo, Rich et al.(2021) consideram que *“it may require more than a year of teaching coding and CT for teachers to begin to really feel confident with their computational thinking abilities”*.

Dos 17 alunos do 11º TPI, 13 compareceram aos dois testes, 1 aluno faltou nos dois testes, 1 aluno desistiu no primeiro teste, e faltou no segundo, e 2 alunos fizeram só um dos testes. Em resumo, 14 alunos realizaram o teste nos dias 15/03 e 01/04, sendo que 13 foram os mesmos nos dois dias (Tabela 16). A aplicação do teste correu dentro da normalidade e a maior parte dos alunos não precisou de explicações detalhadas para entender os enunciados. À exceção do aluno que desistiu, todos os alunos que começaram o teste responderam às 28 questões e enviaram o formulário Google forms para o meu email dentro do prazo indicado, nos dois dias de teste. A figura 90 mostra os acertos por aluno nos dias 15/03 e 01/04.

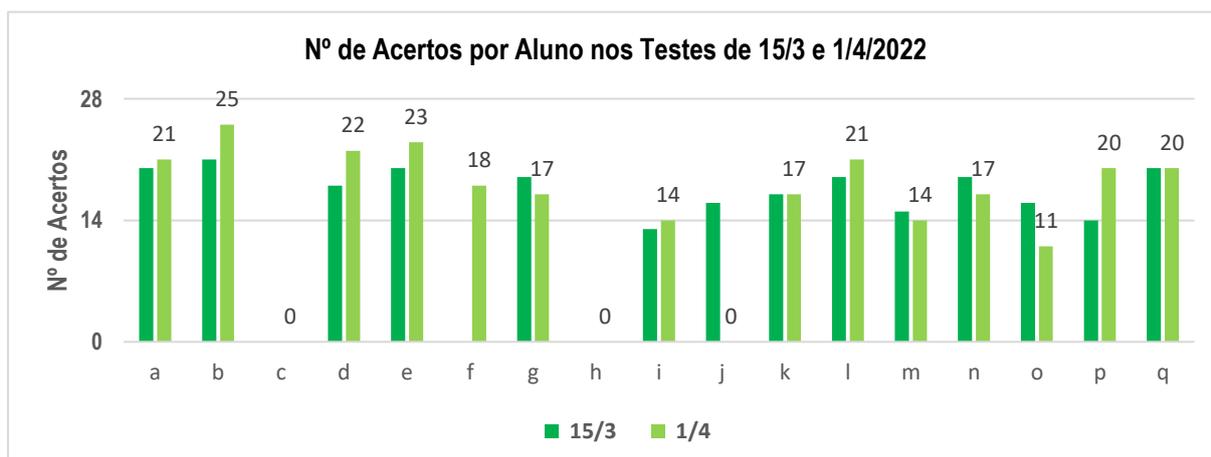


Figura 90. Nº de Acertos por Aluno nos Testes de 15/03 e 01/04/2022

A análise da estatística descritiva do 11ºTPI, calculada pelo suplemento do Excel 10, mostrou que, do dia 15/03 para o dia 01/04, houve uma variação positiva na média, a mediana manteve-se igual e a moda teve uma variação negativa (Tabela 16).

Tabela 16. Pontuação dos Acertos Individuais e Análise Descritiva dos Resultados do 11ºTPI, em 15/03 e 01/04

Aluno	15/03/2022	01/04/2022
	Nº de Acertos	
A	20	21
B	21	25
C	D	F
D	18	22
E	20	23
F	F	18
G	19	17
H	F	F
I	13	14
J	16	F
K	17	17
L	19	21
M	15	14
N	19	17
O	16	11
P	14	20
Q	20	20
Média	17,64	18,57
Erro-padrão	0,67	1,04
Mediana	19	19
Modas	19 e 20 (n=3)	17(n=3)
Desvio-padrão	2,50	3,88
Variância da amostra	6,25	15,03
Curtose	-0,91	-0,33
Assimetria	-0,53	-0,32
Intervalo	8	14
Mínimo	13	11
Máximo	21	25
Soma	247	260
Contagem	14	14
Média 15/03 e 01/04	18,11	

D = desistiu;

F = faltou

Ao mesmo tempo, a análise gráfica da frequência de pontuação nos dois dias de aplicação do teste revelou que os dados não seguiram uma distribuição normal (Figura 91).

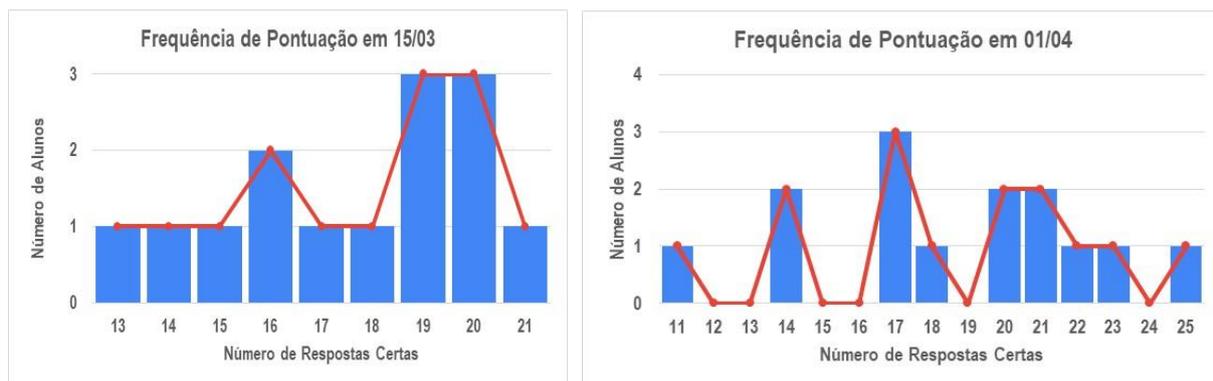


Figura 91. Frequências de Pontuação no teste CTt em 15/3 e 01/04

A comparação estatística de duas amostras emparelhadas é preferencialmente realizada pelo teste t de Student, porém, um dos requisitos é a simetria da distribuição (Anexo I). Quando os dados afastam-se da curva normal, realizam-se os testes não paramétricos correspondentes. No presente caso, a solução foi aplicar o teste não paramétrico de Wilcoxon para 2 amostras emparelhadas. Pela tabela de valores críticos de W, encontrou-se, em $n = 13$ e $\alpha = 0,05$, o valor crítico W igual a 17. Feitos os cálculos para os dados do 11ºTPI, os valores absolutos dos somatórios positivo e negativo foram 30 e 58, respetivamente. A comparação com o valor crítico W mostrou que o menor valor do somatório era maior que o valor crítico, pelo que, a hipótese nula não foi rejeitada, ou seja, os resultados de 15/03 e 01/04 não são significativamente diferentes (Grande, 2016; Yeager, 2022; Stat, 2022).

Tendo em conta os resultados médios do teste CTt dos alunos do 11ºTPI nos dois dias (18,11 em 28, Tabela 16), parece-me coerente deduzir que os acertos estão no mesmo patamar dos estudantes espanhóis (18,05 em 28, Román González, 2016, p. 389) e suíços (18,45 em 28, Guggemos et al., 2022, p.10), uma vez que as avaliações basearam-se no CTt e os alunos tinham um nível semelhante de idade e escolaridade (os resultados de Chan et al., 2021 não estão apresentados porque não os percebi).

A tabela 17 mostra os temas e as tarefas associadas às questões do CTt (Román-González, 2016, anexo F) assim como o índice de dificuldade, π , do 11ºTPI, médio dos dias 15/3 e 01/04, expresso pelo cociente entre o número de alunos que acertaram a questão e o número total de alunos que realizaram o teste (quanto maior o π , maior o número de acertos).

Tabela 17. Teste CTt, Temas, Tarefas e Índice de Dificuldade, pi , do 11ºTPI

Questão	Tema	Tarefa	Índice de dificuldade pi
1	Visão espacial, direções	Sequenciar	0,93
2		Completar	0,96
3		Depurar	0,82
4		Sequenciar	0,82
5	Ciclos (<i>Loop</i>), repetir vezes	Sequenciar	0,89
6		Completar	0,96
7		Depurar	0,82
8		Sequenciar	0,61
9	Ciclos (<i>Loop</i>), repetir até	Sequenciar	1,00
10		Completar	0,79
11		Depurar	0,96
12		Sequenciar	0,25
13	Condicional <i>if</i>	Sequenciar	0,93
14		Sequenciar	0,82
15		Completar	0,07
16		Depurar	0,68
17	Condicional <i>if/else</i>	Sequenciar	0,61
18		Sequenciar	0,93
19		Depurar	0,89
20		Completar	0,79
21	Condicional <i>while</i>	Sequenciar	0,25
22		Completar	0,32
23		Completar	0,25
24		Completar	0,29
25	Função simples	Sequenciar	0,32
26		Completar	0,57
27		Sequenciar	0,32
28		Completar	0,36

A Figura 92 mostra o índice de dificuldade por questão, pi , segundo a média dos testes de 15/03 e 01/04/2022. Como esperado, as questões entre a 21 e a 28, as mais complexas, foram as que tiveram mais frequência de erros. Os conceitos envolvidos são o uso da condicional *while* (21 a 24) e de função simples (25 a 28), pelo que, é preciso reforçá-los.

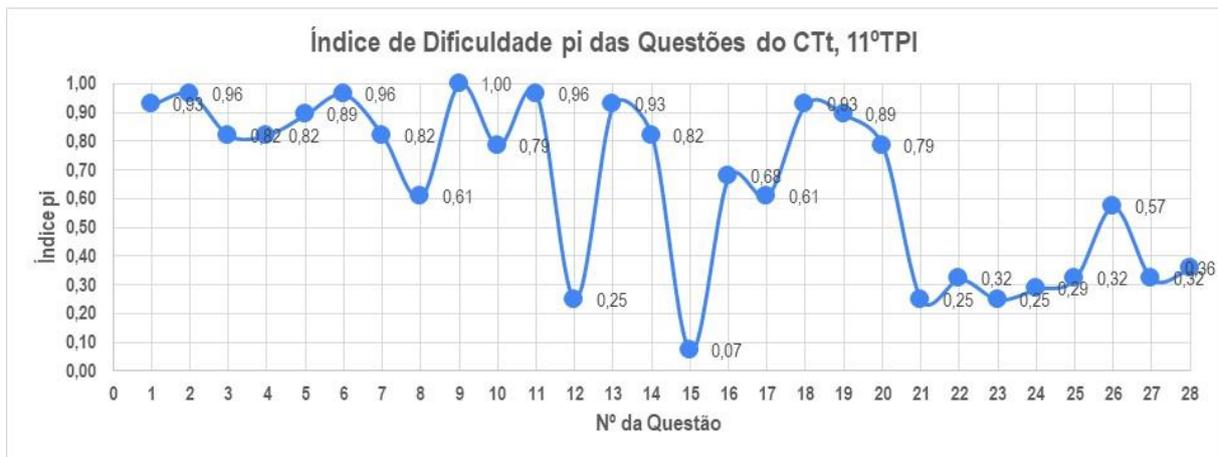


Figura 92. Índice de Dificuldade pi das questões do CTt, 11º TPI

Em relação às questões 12 e 15, que apresentaram um claro viés na ordem crescente de dificuldade para o 11º TPI, o autor do teste CTt também observou este facto aquando da aplicação com alunos espanhóis, considerando-as excessivamente difíceis (Román-González, 2016, p.439-440).

Na reflexão do autor do teste, a maior dificuldade da questão 12 deve-se à existência de um aninhamento simples: o ciclo (*loop*) definido, repetir vezes, está dentro de um ciclo (*loop*) indefinido, repetir até. Segundo Mühling e Hubwieser (2015, citado por Román-González, 2016, p.439), os alunos mais jovens têm dificuldade em resolver questões com *loop* indefinido em aninhamento. Na questão 15, há um duplo aninhamento: *loop* definido, repetir vezes, dentro de uma condicional simples ('*sim*'), e este, por sua vez, dentro de um *loop* indefinido, repetir até.

Na aplicação do CTt com alunos suíços, Guggemos et al. mantiveram a questão 12 e removeram a 15 do CTt, mas não apresentaram os acertos às questões referindo apenas que 67,8% dos alunos foram sinalizados com necessidades de apoio específico (2022, p.20).

Nas entrevistas individuais, perguntei aos alunos a sua opinião sobre a questão 15 (Apêndice VI). Das muitas respostas, uma que me pareceu interessante foi a do aluno Q: "... ah, sim, foi um caso de precipitação, como viram que a primeira era verdadeira, já ninguém olhou as outras [opções]. Talvez se começasse com a certa e depois as meio-certas...". De facto, nos dois testes a resposta certa era a alínea D, embora o problema constasse na questão nº 15, no dia 15/3, e tenha passado a questão nº 16, no dia 01/04. No dia 15/3, dos 13 alunos que erraram a resposta, 12 escolheram a alínea C; a 01/04, 11 alunos escolheram a alínea A, inclusive o que tinha acertado a pergunta no 1º teste. Dos 14 alunos entrevistados, apenas 1 sugeriu a eliminação da pergunta, pelo que, antes de eliminar a pergunta, numa próxima aplicação, pode valer a pena testar a hipótese do aluno Q e observar o número de acertos. É de referir que nenhum aluno mencionou que o teste CTt seria de difícil compreensão pelo facto de estar escrito em Java. Também considero relevante o facto de, para alunos da mesma faixa etária, as

perguntas com menos acertos serem as mesmas, independentemente da nacionalidade espanhola, portuguesa ou suíça.

Concluídas as entrevistas e o tratamento dos dados, apresentados neste tópico 5.2.3, considero que foi atingido o objetivo (iii) do plano de intervenção pedagógica, *Recolher e Tratar Dados Associáveis ao Pensamento Computacional em Programação*.

5.2.4. Avaliação ao Professor Estagiário

O Questionário de Avaliação da Qualidade do Professor Estagiário e do Curso foi aplicado anonimamente, em 01/04/2022, a seguir à segunda aplicação do CTt. Elaborei o questionário com base nos trabalhos de Pérez-Poch (2021) e Marsh & Roche(1997). O questionário é constituído por 11 perguntas, sendo 8 de resposta politómica, do tipo Likert, com 4 opções (Raramente/Às vezes/Muitas vezes/Quase sempre) e 3 de resposta dicotómica (Não/Sim). Das 11 perguntas, 5 dizem respeito à ação do professor em sala de aula (2, 3, 4, 5, 6), 3 abordam a planificação do curso (7, 8, 9), 1 quantifica a opinião sobre o professor, na globalidade (10) e 2 são de autoavaliação (1,11). Ao final, há uma 12ª questão, opcional, de resposta aberta, sobre sugestões de melhoria (Apêndice IV). Para facilitar a análise, os resultados foram convertidos em percentagem e agrupados por tipo de conteúdo.

Nas questões relativas à ação do professor em sala de aula (Tabela 18) e à planificação do curso (Tabela 19), havia duas respostas favoráveis ao professor, sendo uma muito favorável (quase sempre) e outra favorável com ressalvas (muitas vezes), e duas desfavoráveis, com a mesma nuance (às vezes – desfavorável com ressalvas; raramente – muito desfavorável).

Tabela 18. Avaliação dos Alunos Relativamente à Ação do Professor em Sala de Aula

Entusiasmo	2. O professor conseguiu que as suas aulas fossem agradáveis			
	Raramente: 0%	Às vezes: 0%	Muitas vezes: 43%	Quase sempre: 57%
Organização	3. As explicações do professor foram claras			
	Raramente: 0%	Às vezes: 0%	Muitas vezes: 57%	Quase sempre: 43%
Interação com o grupo	4. O professor estimulava as perguntas do grupo e dava respostas satisfatórias			
	Raramente: 0%	Às vezes: 0%	Muitas vezes: 71%	Quase sempre: 29%
Atitude pessoal	5. O professor mostrou-se acessível aos estudantes			
	Raramente: 0%	Às vezes: 0%	Muitas vezes: 36%	Quase sempre: 64%
Conteúdo	6. O professor apresentou vários pontos de vista quando foi necessário			
	Raramente: 0%	Às vezes: 14%	Muitas vezes: 50%	Quase sempre: 36%

Numa interpretação rápida, considerei que, nos tópicos entusiasmo e atitude pessoal, a maioria dos alunos (> 50%) fez uma avaliação muito favorável à minha atuação, ao indicarem que consegui que as aulas fossem agradáveis e que mostrei-me acessível aos estudantes, quase sempre. Quanto aos tópicos organização, interação com o grupo, a opinião da maioria foi favorável, com ressalvas, ao indicarem que, muitas vezes, dei explicações claras, estimei as perguntas do grupo e apresentei vários pontos de vista. O tópico conteúdo, foi o único que teve três tipos de opinião: a maioria expressou-se favorável, com ressalvas (50%), e 36% considerou muito favorável, mas 14% dos alunos (2 em 14) expressaram opinião desfavorável, com ressalvas, ao indicar “às vezes”.

Em relação à planificação dos curso (Tabela 19), 93% dos alunos consideraram que os métodos de avaliação foram adequados, 100% estiveram de acordo com a bibliografia disponibilizada, e 79% atribuíram ao curso um grau de dificuldade normal, comparativamente a outras disciplinas de informática, havendo também que o considerasse difícil (14%) e fácil (7%).

Tabela 19. Avaliação dos Alunos sobre a Planificação do Curso

Exames	7. Os métodos de avaliação do curso foram equitativos e adequados
	<input type="checkbox"/> Não: 1 (7%); <input type="checkbox"/> Sim:13 (93%)
Trabalhos do curso	8. A bibliografia e o material recomendado para este curso são corretos e adequados
	<input type="checkbox"/> Não; <input type="checkbox"/> Sim: 14 (100%)
Carga de trabalho e dificuldade	9. Este curso comparado com outros de informática, foi:
	<input type="checkbox"/> Muito fácil; <input type="checkbox"/> Fácil:1 (7%); <input type="checkbox"/> Normal: 11 (79%); <input type="checkbox"/> Difícil: 2 (14%); <input type="checkbox"/> Muito difícil

Quanto às avaliações quantitativas, 64% dos alunos atribuíram, ao professor e ao curso, uma nota entre 15 e 18, enquanto 36% outorgaram uma nota superior a 18 valores (Tabela 20). Na autoavaliação (Tabela 21), os alunos consideram a 100% que aprenderam coisas valiosas e foram mais exigentes com as suas próprias notas, tendo 71% revelado uma expectativa entre 11 e 14 valores e 29% entre 15 e 18 valores. Tais pedidos moderados de recompensa podem estar relacionados com a autoavaliação do seu empenho no curso, como mostra o questionário do tópico 5.2.5.

Tabela 20. Avaliação Global Quantitativa sobre o Professor Estagiário

Visão Geral	10. A nota geral que dás ao professor e ao curso é:
	<input type="checkbox"/> menor que 5
	<input type="checkbox"/> entre 6 e 10
	<input type="checkbox"/> entre 11 e 14
	<input type="checkbox"/> entre 15 e 18: 9 (64%)
<input type="checkbox"/> maior que 18: 5 (36%)	

Tabela 21. Autoavaliação Qualitativa e Quantitativa dos Alunos

Qualidade da aprendizagem	1. Aprendi coisas que considero valiosas <input type="checkbox"/> Não; <input checked="" type="checkbox"/> Sim: 14 (100%)
Autoavaliação	11. Pelo teu desempenho neste curso, a nota que esperas ter é: <input type="checkbox"/> menor que 5 <input type="checkbox"/> entre 6 e 10 <input checked="" type="checkbox"/> entre 11 e 14: 10 (71%) <input checked="" type="checkbox"/> entre 15 e 18: 4 (29%) <input type="checkbox"/> maior que 18

Na questão opcional de sugestão de melhoria, apenas três alunos responderam, designadamente: “As aulas foram bastante interessantes e gostei bastante das aulas que nos foram proporcionadas pelo forte do professor Mauricio Morais. Gostava de o ter por perto e que continuasse como professor na nossa turma!”; “Gostei, espero que volte, gostei das suas aulas”; “Forma de ensino calmo e bem explicado com disponibilidade para tirar dúvidas sempre que possível”. Do meu ponto de vista, estes três comentários elogiosos constituíram uma atitude de encorajamento ao meu trabalho, um “vá em frente” e não uma lisonja porque, ao observar as outras respostas, verifiquei que nenhum dos três deu-me pontuação máxima em todos os tópicos: na avaliação global, dois pontuaram-me entre 15 e 18 e, nos tópicos de atuação em sala de aula, o “muitas vezes” foi mais frequente que o “quase sempre”. Gostei da atitude destes três alunos pois souberam ser empáticos, sem deixar de afirmar a sua opinião crítica.

5.2.5. Autoavaliação dos Alunos

Na presente intervenção pedagógica, efetuei dois momentos de autoavaliação dos alunos: (i) em 01/04, no final da intervenção sumariada, no mesmo questionário em que avaliaram o professor, foi solicitada uma opinião sobre as suas impressões pessoais, sem que soubessem os seus resultados de avaliação; (ii) em 26/04, após mostrar os resultados sobre os testes CTt e das Apps, a autoavaliação incidiu sobre as suas atitudes em sala de aula, as dificuldades sentidas e as perspetivas futuras. O segundo questionário, composto por 14 perguntas dicotómicas (verdadeiro/falso), foi nominal e voluntário e feito na aula do Prof. José Pereira, depois da entrevista individual; recolhi-os no final da aula.

Todos os alunos que participaram das entrevistas de 26/04 entregaram o questionário preenchido e completo. O questionário é de minha autoria, específico sobre a intervenção e espelha as minhas interrogações e suposições (Apêndice IX). Analisando as respostas, compreendi melhor a perceção dos alunos, e a minha própria, sobre o tempo de E-A que vivenciámos. A figura 93 mostra os resultados globais, em percentagem, para facilitar a leitura. À afirmação “Compreendi o que era esperado fazer nos trabalhos IMC e ICQ”, 94% confirmaram saber o que era suposto fazerem. “Quando tive dúvidas perguntei ao professor”, 94% confirmaram. “Quando tive dúvidas perguntei aos colegas”, 88% o fizeram. “Fui capaz de resolver os trabalhos propostos sem ajuda do professor e dos colegas”, 100% dos alunos

rejeitaram a afirmação. “Precisei de apoio na construção das telas” foi confirmado por 56% e rejeitado por 44% dos alunos. “Precisei de apoio no código Java”, 88% certificaram. “O ambiente Android Studio é complicado”, 63% disseram que não. “Faltam-me bases de Java para bem compreender o Android”, 69% pensam que sim. “Um trabalho consegui terminar, mas o outro não”, foi o caso de 63% dos alunos. “Não consegui terminar os trabalhos IMC e ICQ na aula”, 63% não conseguiu. “Tentei fazer os trabalhos IMC e ICQ em casa”, 13% dos alunos tentaram completar os trabalhos em casa. “Interesso-me pela matéria e tenciono continuar a programar Apps para Android”, 63% tenciona continuar a programar Apps. “Interesso-me pela matéria, mas não tenciono continuar a programar Apps para Android”, 81% rejeitou abandonar a aprendizagem Android. “Não gosto de programação Android”, 75% dos alunos rejeitaram desgostar de Android.

Em geral, as respostas dos alunos seguiram o mesmo sentido das minhas observações de aula. As respostas que mais me impactaram foram (i) a confirmação de 63% que sabiam que não acabaram os trabalhos em sala de aula, conforme eu já tinha verificado durante a correção, e, mesmo assim, só 13% tentaram completar os trabalhos em casa, e (ii) apesar de tão pouco empenho, 63% declararam que tencionam continuar a programar Apps no futuro. Tal como no tópico 3.4, da contextualização, os alunos voltaram a evidenciar neste questionário a pouca perceção que têm do efeito dos seus comportamentos cotidianos na sua vida futura. No âmbito do Diretor de Turma, talvez alguma coisa pudesse ser feita, no sentido de conversar com os Encarregados de Educação. No âmbito dos Professores das disciplinas, talvez seja viável propor uma atividade interdisciplinar, individual ou em grupo, que os estimule a um estudo autónomo fora do horário da aula.

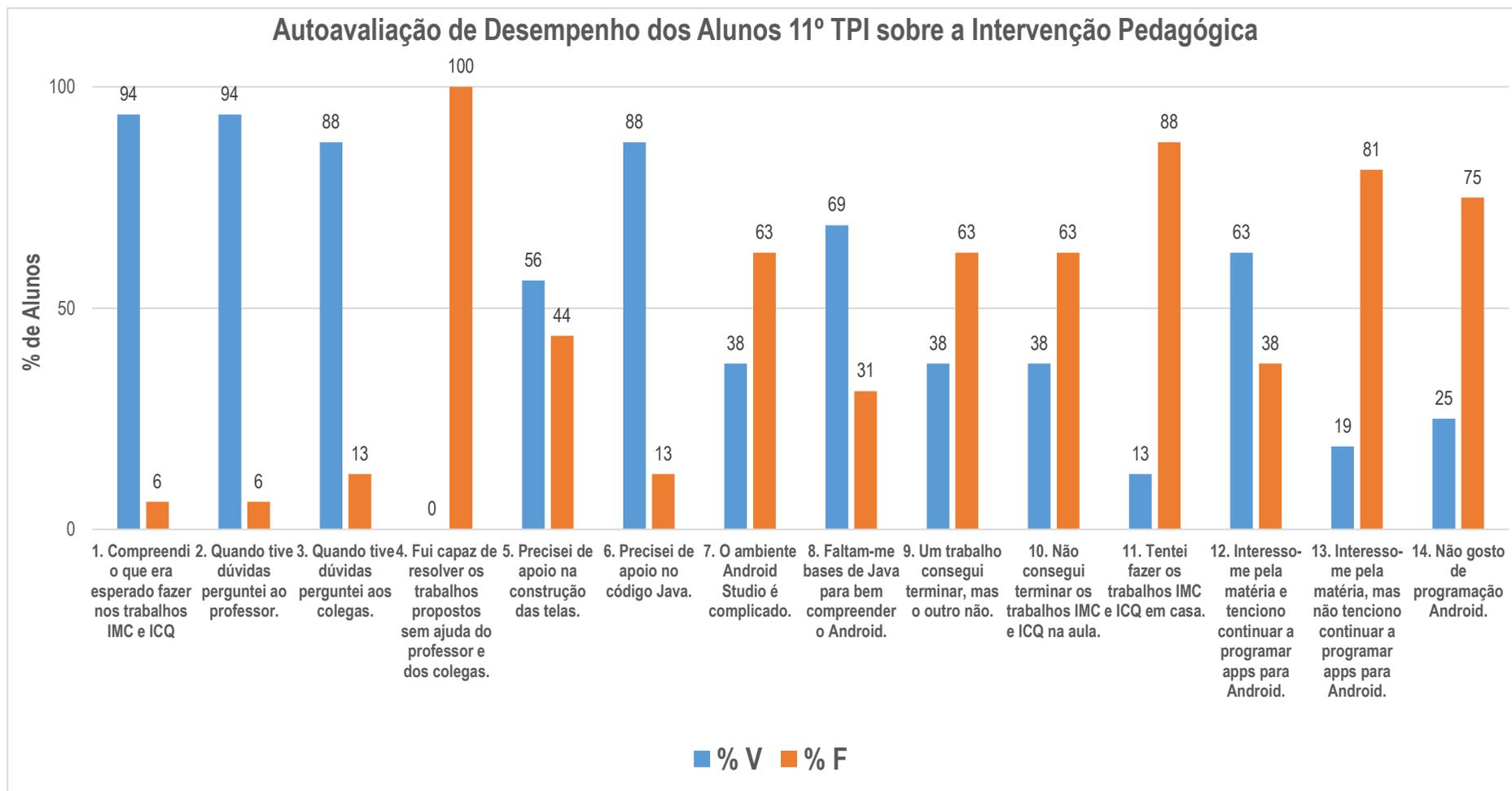


Figura 93. Autoavaliação de Desempenho dos Alunos do 11º TPI sobre a Intervenção Pedagógica, em percentagem de respostas V/F

5.3. Reflexões

5.3.1. O meu Desempenho na Intervenção

Para além da minha reflexão individual, considero que houve quatro períodos temporais de reflexão coletiva, descritos a seguir, que muito contribuíram para a minha perceção do meu desempenho na intervenção pedagógica.

(i) Ao longo da intervenção, o exercício de autoavaliação foi auxiliado pelos emails semanais partilhados por nós três: eu, o Professor Cooperante e a Professora Supervisora, antes da primeira intervenção da semana. Nos dias de intervenção, o Professor Cooperante sempre esteve presente na aula e, nos intervalos, entre a saída de um turno e a entrada do seguinte, fazia os seus comentários e chamadas de atenção. Além disso, no decorrer das três semanas de intervenção, a Professora Supervisora assistiu à intervenção em um turno, todas as semanas, e, ao final das intervenções, conversávamos sobre as minhas impressões, se eu considerava que tinha atingido os objetivos a que me tinha proposto, e ela fazia comentários diretos e objetivos; não obstante, só ao final da intervenção apercebi-me da profundidade das suas palavras.

(ii) Nas intervenções não sumariadas de 19 e 26/04, pude inteirar-me e refletir sobre as perceções dos alunos em relação aos tópicos de ensino-aprendizagem e de investigação da intervenção.

(iii) A 27 de abril, houve a reunião preliminar de avaliação de intervenção pedagógica, 15 às 17h, comigo, o Professor Cooperante e a Professora Supervisora, na Escola Secundária. Neste dia fizemos um análise aprofundada da minha intervenção. Apercebi-me que a grande maioria das observações dos Orientadores eram pertinentes e que devia refletir sobre elas.

(iv) A 22 de julho os meus Orientadores e eu voltámos a reunir, para a sessão final de avaliação e comentários sobre a Grelha de Avaliação da Prática Profissional do Estagiário (GAPPE), desta vez com a presença do meu colega de estágio, Pedro Sequeira, que também foi avaliado pelos Orientadores. Neste dia, eu e ele realizámos a auto e heteroavaliação dos tópicos B e C do GAPPE.

Para fins de apresentação no presente relatório, organizei a lista dos meus 5 principais pontos, fracos e fortes, nas vertentes atuação e planificação da intervenção, segundo a minha autoavaliação, mas tendo em conta os comentários dos Orientadores e dos Alunos (Tabela 22).

Tabela 22: Autoavaliação das Vertentes Atuação e Planificação da Intervenção.

Atuação na Intervenção	
Pontos Fracos	Pontos Fortes
<ul style="list-style-type: none"> - Muitas vezes, ter estado centrado em mim mesmo. - Às vezes, não ter implementado um ritmo de demonstração adequado para a média dos alunos. - Ter sido lento a propor uma atividade alternativa aos alunos, quando a atividade prevista apresentou falhas. - Não ter convertido uma situação embaraçosa num momento de descontração. - Não ter dado tempo suficiente para o aluno conjecturar/desenvolver a sua resposta à uma pergunta (nas entrevistas individuais). 	<ul style="list-style-type: none"> - Ter atendido atempadamente, e com boa vontade, sempre que os alunos solicitaram esclarecimentos. - Ter aplicado a proxémica, quase sempre. - Ter aplicado a paralinguagem, muitas vezes. - Ter respeitado a privacidade dos alunos, ao mostrar individualmente os seus resultados no CTt e nas Apps. - Ter desenvolvido uma relação de confiança que motivou os alunos (16 em 17) a aderirem ao inquérito final, voluntário e identificado, de autoavaliação de desempenho na intervenção e o responderem na totalidade.
Planificação da Intervenção	
Pontos Fracos	Pontos Fortes
<ul style="list-style-type: none"> - Não ter elaborado uma pergunta para o início de cada aula. Por exemplo, sobre a visualização de algum conteúdo enviado aos emails institucionais dos alunos, de modo a cumprir o 1º passo de Gagné (IEFP, 2004) e a estimular o hábito de consulta do email institucional. - Não ter elaborado puzzles de Parson, como atividade alternativa, no caso de algum imprevisto (Sentance et al, 2019). - Não ter elaborado uma pergunta para o final de cada aula, para além de apenas informar as atividades da aula seguinte. Por exemplo, solicitar que os alunos referissem os conceitos aprendidos naquele dia, cumprindo o 9º passo de Gagné (IEFP, 2004). - Não ter elaborado um debate rápido sobre o que os alunos pretendiam fazer com o material que produziram, após a entrega da segunda App, cumprindo o tópico Satisfação, S1.1, da estratégia ARCS (Keller, 1987). - Não ter elaborado uma estratégia que permitisse a heteroavaliação das Apps pelos alunos (Sitthiworachart & Joy, 2008). 	<ul style="list-style-type: none"> - Ter escolhido uma metodologia que aplica conhecimentos e dados prévios, a CBL, de modo a poder focar-me nos conteúdos Android. - Ter elaborado uma apresentação com complexidade adequada ao 11º ano, em sequência lógica, no modo demonstrativo-interrogativo, e tendo em conta as debilidades detetadas pelo CTt. - Ter selecionado os critérios de avaliação das Apps como uma combinação dos conhecimentos novos com os tópicos de baixo desempenho no CTt. - Ter elevado a complexidade do teste CTt, e dirigido a alunos do 11º ano de programação, ao converter a linguagem original de blocos para Java. - Ter elaborado o inquérito final de autoavaliação de desempenho de um modo que ajudou a colmatar algumas das minhas falhas de observação da turma.

5.3.2. Limitações da Intervenção

A experiência prática que a Intervenção Pedagógica Supervisionada proporcionou-me fez toda a diferença na minha aprendizagem como professor. A convivência com os alunos, nas aulas e nas entrevistas, com os seus trabalhos de programação, e com os testes CTt, e as suas avaliações sobre a minha atuação, são inesquecíveis.

Não obstante, pelo facto de a intervenção ter ocorrido num intervalo de tempo muito compacto, penso que a minha falta de experiência atrapalhou o meu desempenho. Sim, é um facto que foi possível

realizar trabalho, e os resultados mostraram que houve aprendizagem, no entanto, se houvesse mais tempo para refletir, entre uma aula e outra, talvez eu fosse mais eficaz. Por outro lado, um professor tem de estar apto para todas as situações. Neste sentido, ter lecionado 5 aulas consecutivas, em contexto real e presencial, contribuiu para a minha formação docente, porque agora sinto-me mais preparado para atuar em situações de estresse.

5.3.3. Aplicabilidade da Intervenção na Vida Profissional

Considero que poderei aproveitar o conjunto da minha intervenção pedagógica, de uma maneira geral, e, em especial, se vier a lecionar num Curso Profissional de Programador/a de Informática. Os conhecimentos sobre programação Android lecionados ao 11ºTPI, embora situados no nível introdutório da espiral de aprendizagem, têm várias possibilidades de utilização, inclusivamente ao nível das atividades interdisciplinares. Por exemplo, as Apps de Índice de Massa Corporal (IMC) e Índice Cintura-Quadril (ICQ) podem ser utilizadas nas atividades da disciplina de Educação Física, funcionando a parceria como um reforço motivacional para alunos e professores de ambas as disciplinas. Esta sugestão vai no mesmo sentido da proposta de Moura e Carvalho (2010), que referiram o uso do telemóvel como uma potencial fonte de motivação dos alunos pela escola e constataram que alunos do ensino secundário (n=68) aderiram à proposta de integrar o telemóvel na sua aprendizagem. As mesmas Apps podem colaborar na prevenção e combate à obesidade de adolescentes, juntamente com outras medidas antropométricas, uma vez inseridas num programa do âmbito das Ciências da Nutrição (Tafur-Morales et al., 2021; Tenenaula & Buñay, 2020).

Para além das Apps IMC e ICQ, os conteúdos globais lecionados ao 11ºTPI, se devidamente acompanhados por um docente de informática, seriam suficientes para a realização de uma ação interdisciplinar, por exemplo com a disciplina de Cidadania e Desenvolvimento (DACP0081000), em moldes semelhantes à realizada com a disciplina de Estudos Sociais, do 10º ano (Freire, 2021).

No caso de me ser atribuída a leção em cursos secundários científico-humanistas, penso que a programação de Apps continua a ser um conteúdo possível e desejável mas, neste caso, devido a não formação prévia dos alunos, poderá ser mais adequado recorrer a um IDE de programação visual por blocos, como o MIT App Inventor (Liang et al., 2021; Tóth & Lovászová, 2021).

5.4. Considerações Finais sobre a Intervenção Pedagógica Supervisionada

Recapitulando os objetivos gerais da planificação, penso que a relação de E-A promovida na intervenção foi favorável ao desenvolvimento das competências do PASEO, com destaque para o tópico *Participação*, maior que 70% nos dois turnos. Quanto aos objetivos específicos, considero que a adesão inicial dos alunos à aplicação do CTt e, principalmente, o cumprimento de todos procedimentos por 73% dos participantes, é um indicador positivo de motivação no estudo da programação, objetivo (i), pois, entre um teste e outro, os alunos estudaram a programação de duas Apps. Convertendo para percentagem, temos que 59% dos trabalhos de Apps do 11ºTPI foram aprovados, o que parece-me positivo para o objetivo (ii). Através dos dados obtidos no CTt, foi possível verificar uma similaridade no padrão de respostas, e de dificuldades, apresentadas pelos alunos do 11º TPI e os seus colegas de mesma idade em Espanha e Suíça; pondero que foi cumprido o objetivo (iii). Relativamente ao objetivo (iv), *identificar as percepções dos alunos relativamente à qualidade da intervenção realizada e à sua aprendizagem*, as respostas dos alunos aos questionários e às entrevistas foram muito esclarecedoras, aprendi a interpretá-las e pretendo estar mais atento às minhas falhas nas próximas intervenções, portanto, julgo que o objetivo (iv) também cumpriu-se.

A aplicação prática do pensamento computacional é um tema interessante e vasto. Para focar a questão de investigação pedagógica no contexto da intervenção - *Como o pensamento computacional, e ferramentas a ele associadas, podem contribuir no E-A de Apps para Android?* - foi preciso encontrar uma ferramenta com elevada especificidade. Felizmente, o teste CTt, criado por Román-González, baseia-se em conceitos de programação. Por tê-lo aplicado no primeiro dia de aula, e corrigido logo a seguir, foi possível detetar as lacunas dos alunos e, a partir daí, sempre que o conteúdo da aula permitiu, procurei reforçar a aprendizagem daquelas falhas, ou seja, a primeira aplicação do CTt funcionou como um teste de diagnóstico. Também aquando da elaboração dos critérios de correção das Apps, foi útil a reflexão sobre os conceitos das questões finais do CTt. Em jeito de conclusão, penso que as duas utilizações referidas constituem respostas viáveis para a QIP.

É de mencionar que a repetição da aplicação do CTt pode auxiliar a monitorizar a evolução do pensamento computacional de alunos de programação informática, no entanto, o espaço de tempo entre as aplicações deve ser superior ao da presente intervenção (talvez 1 ano letivo?). No capítulo 6, sugiro outras aplicações práticas do CTt, em outros contextos.

6. Reflexões sobre o Trabalho Futuro

Num primeiro momento, a disponibilidade material de um computador por aluno parece o cenário ideal para a aprendizagem de programação. No entanto, há autores que recomendam a Programação em Pares ou uma alternância entre o estudo individual e o estudo aos pares (Brown & Wilson, 2018; Sentance et al., 2019; Sentance & Waite, 2021).

No estudo individual, por um lado, o aluno tem liberdade para trabalhar ao seu ritmo, testar as suas hipóteses e exercer a sua criatividade. Por outro lado, o aluno que trabalha sozinho frequentemente solicita opinião, comenta observações e pede esclarecimentos, aos colegas e ao professor, de maneira aleatória. Esta ação, descompromissada e imediatista, pode levar a que o solicitante interrompa o colega do lado num momento inconveniente, por exemplo, se o colega estiver a interiorizar uma instrução ou a elaborar um algoritmo de resolução de problema. A alternativa de recorrer ao professor também pode dar-se numa conjuntura inadequada, por exemplo, se o professor já está a esclarecer outro aluno ou está a refletir sobre a continuação da aula. O somatório destas ações casuais, por vários alunos, pode levar a ruído elevado, desconforto, e à quebra de atenção do conjunto da turma.

Na Programação em Pares a existência de ruído na sala de aula é um pressuposto, pois os pares conversam entre si (Figura 94). De facto, na abordagem PRIMM (Predict-Run-Investigate-Modify-Make), que adota a Programação em Pares, a oralidade é vista como um factor facilitador da aprendizagem de programação porque “... considerando que a compreensão do que um programa faz é uma tarefa desafiadora, ser capaz de explicar verbalmente o que o programa faz será mais fácil do que expressá-lo por escrito (Sentance et al., 2019)”. No PRIMM, a aprendizagem inicia-se com a leitura de um trecho, simples e pré-construído, de um programa na mesma linguagem. A seguir, o programa lido é executado e as suas características são investigadas e debatidas. O professor propõe uma modificação simples e correta do programa e, por fim, lança um desafio aos alunos, no sentido de avançarem na complexidade do programa (Sentance & Waite, 2017). Do ponto de vista da gestão da sala de aula, a Programação em Pares traz desafios adicionais ao professor, como a eclosão de conflitos entre os alunos e a dispersão do tema da aula, pelo que, a lecionação em pares de professores poderá ser a estratégia mais eficaz

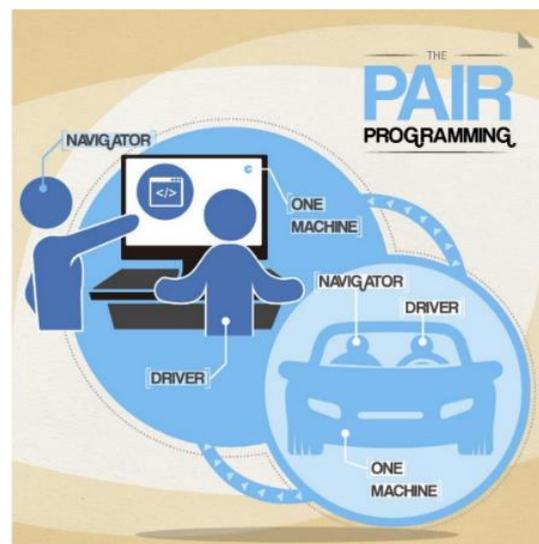


Figura 94: Operacionalização da Estratégia Pedagógica Pair Programming (Pedro et al., 2017)

neste tipo de aula. Quanto à formação dos pares, além de terem um perfil comportamental compatível, é recomendado agrupar alunos com habilidades semelhantes, para evitar que um aluno domine a colaboração (Lewis et al., 2018).

O teste CTt tem-se mostrado uma ferramenta adequada para a medição de habilidades de raciocínio lógico e visão espacial (Román-González et al., 2017; Çetin et al., 2020; Guggemos et al., 2022), pelo que, pode ser aproveitado na seleção dos alunos para a formação dos pares. Além disso, por refletir o estágio de conhecimento de lógica de programação de cada aluno, o CTt pode ajudar o professor a distinguir as tarefas de programação que o aluno está pronto a realizar, aquelas que poderá vir a realizar, com a devida instrução, e as que ainda estão além das suas capacidades cognitivas atuais, ou seja, o CTt pode ser um auxiliar na determinação da Zona de Desenvolvimento Proximal (ZPD) (Figura 95), proposta por Vygotsky (Sentance & Waite, 2021).



Figura 95. Zona de Desenvolvimento Proximal (ZPD), (adapt. Basawapatna et al., 2013).

No que toca aos benefícios da Programação em Pares, Pedro et al.(2017) argumentam que contribui para: “(i) melhorar a qualidade dos programas e soluções, reduzindo os erros e melhorando a sua eficiência; (ii) incrementar a partilha de conhecimento entre pares; (iii) promover o desenvolvimento de competências; (iv) promover a capacidade de resiliência; e (v) facilitar o acompanhamento por parte do professor.”

Em Portugal, a estratégia da Programação em Pares conta com relatos de intervenções bem sucedidas em vários anos da escolaridade obrigatória, nomeadamente no 11º ano de um curso profissional de informática (Sousa, 2016), no 8º ano, e no ensino profissional de eletrotecnia e gestão informática (Rosado, 2020).

Nas minhas futuras aulas, penso incluir a Programação em Pares e a abordagem PRIMM, no leque de possíveis estratégias a utilizar.

Referências Bibliográficas

- Abreu, M. V. (2020). CoviHealth: Aplicação móvel para monitorização de hábitos de vida saudáveis. Dissertação de Mestrado. Universidade da Beira Interior.
- Agrupamento de Escolas de Vila Nova de Gaia, (AgrCanelas). (2020). *Prova de Aptidão Profissional*. Obtido de <https://agrcanelas.edu.pt/2020/07/16/pap-informatica-1a-sessao/> e <https://agrcanelas.edu.pt/2020/07/17/pap-informatica-2a-sessao/>
- Agência Nacional para a Qualificação e o Ensino Profissional (ANQEP). (2022). *Pesquisa de Qualificações, Qualificações de Dupla Certificação*. Obtido de <https://catalogo.anqep.gov.pt/qualificacoesPesquisa>
- Armoni, M. (2014). Spiral thinking: K-12 computer science education as part of holistic computing education. *ACM Inroads*, 5(2), 31-33.
- Associação Nacional dos Professores de Informática (ANPRI). (2021a). *Proposta de uma Disciplina de Computação, como Opção Bial para o Curso Científico-Humanístico de Ciências e Tecnologias*. Obtido de <https://www.anpri.pt/mod/forum/discuss.php?d=13502>
- Associação Nacional dos Professores de Informática (ANPRI). (2021b). Webinar: A Hora do Código, 09/12/2021. Obtido de <https://www.youtube.com/watch?v=NPGmqtLk6n4>
- Associação Nacional dos Professores de Informática (ANPRI). (2022). *O Concurso Nacional PAPTICe*. Obtido de <https://anpri.pt/pap/>
- Australian Curriculum, Assessment and Reporting Authority (ACARA). (2020). *Computational thinking poster*. Obtido de <https://www.digitaltechnologieshub.edu.au/search/computational-thinking-poster/?id=4ee74b98-09f9-6792-a599-ff0000f327dd>
- Azevedo, D. (2021). *Já são conhecidos os 4 alunos que vão representar Portugal nas Olimpíadas Internacionais de Informática 2021*. Obtido de <https://www.maistecnologia.com/ja-sao-conhecidos-os-4-alunos-que-vaio-representar-portugal-nas-olimpiadas-internacionais-de-informatica-2021/>
- Barbosa, A. C. (2018). *Aprendizagem significativa do conceito de polígono: uma sequência didática para o sexto ano do ensino fundamental*. Dissertação de Mestrado. Universidade Federal de Uberlândia, Brasil.
- Basawapatna, A. R. (2013, August). The zones of proximal flow: guiding students through a space of computational thinking skills and challenges. In *Proceedings of the ninth annual international ACM conference on International computing education research* (pp. 67-74).
- Benczúr, A. & Molnár B. (2018, August). On the Notion of Information–Info-sphere, the World of Formations. In *2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)* (pp. 000033-000038). IEEE.
- Bocconi, S., Chiocciariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., & Stupurienė, G. (2022). *Reviewing Computational Thinking in Compulsory Education (No. JRC128347)*. Joint Research Centre (Seville site). Obtido de <https://erte.dge.mec.pt/noticias/pensamento-computacional-na-escolaridade-obrigatoria-relatorio>
- Brod, G. (2021). How Can We Make Active Learning Work in K–12 Education? Considering Prerequisites for a Successful Construction of Understanding. *Psychological Science in the Public Interest*, 22(1) (pp. 1-7).
- Caires, S. (2021). *Psicologia da Educação, Apontamentos de Aula. Gestão de Sala de Aula*. Universidade do Minho.
- Cambridge Dictionary. (2022). Obtido de App. University of Cambridge, UK: <https://dictionary.cambridge.org/us/dictionary/english/app>

- Castor Informático. (2021). Obtido de Edição 2021. DCC.FC.UP: <http://bebras.dcc.fc.up.pt/2021.html>
- Çetin, İ., Otu, T., & Oktaç, A. (2020). Adaption of the computational thinking test into Turkish. *Turkish Journal of Computer and Mathematics Education* 11 (pp. 343-360). Obtido de <https://dergipark.org.tr/en/download/article-file/1079231>
- Chan, S.-W., Looi, C.-K., & Sumintono, B. (2020). Assessing computational thinking abilities among Singapore secondary students: A Rasch model measurement analysis. *Journal of Computers in Education*, 8 (pp. 213–236). doi:10.1007/s40692-020-00177-2
- Collado-Sánchez, M., García-Peñalvo, F. J., & Pinto-Llorente, A. M. (2021, October). Computational thinking competences training for primary education teachers. In *Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21)* (pp. 758-762). doi:abs/10.1145/3486011.3486544
- Conselho de Ministros. (2018). Decreto-Lei n.º 55/2018, de 6 de julho. *Diário da República n.º 129/2018, Série I de 2018-07-06*. Obtido de <https://data.dre.pt/eli/dec-lei/55/2018/p/cons/20210803/pt/html>
- Cunha, M. E. D. S. (2015). *Aulas interativas com recurso à exposição: uma metodologia para o ensino da linguagem de programação Python*. Dissertação de Mestrado. Universidade do Minho.
- Cutumisu, M., Adams, C., & Lu, C. (2019). A scoping review of empirical research on recent computational thinking assessments. *Journal of Science Education and Technology*, 28(6) (pp. 651-676). Obtido de <https://link.springer.com/article/10.1007/s10956-019-09799-3>
- Dias, A. (2014). *Apoiando uma criança em dificuldades no Jardim de Infância*. Dissertação de Mestrado. Universidade de Aveiro.
- Direção Geral da Educação (DGE). (2018a). *Aprendizagens Essenciais - Ensino Básico*. Obtido de <https://www.dge.mec.pt/aprendizagens-essenciais-ensino-basico>
- Direção Geral da Educação (DGE). (2018b). *Aprendizagens Essenciais - Ensino Secundário*. Obtido de <https://www.dge.mec.pt/aprendizagens-essenciais-ensino-secundario>
- Direção Geral da Educação (DGE). (2022). Projeto "Programação e Robótica na Escola". Obtido de <https://robotica.dge.mec.pt/> e <https://www.youtube.com/watch?v=8Yj4hyfnebl>
- ECO. (2021). Alunos com mais de 10 anos devem usar máscara nas escolas, ECO, 31/08/2021. Obtido de <https://eco.sapo.pt/2021/08/31/alunos-com-mais-de-10-anos-devem-usar-mascara-nas-escolas/>
- ENSICO. (2022). *Plano de Ação*. Obtido de <https://ensico.pt/#services-section>
- ESJS-Mafra, E. S. (2020). *Prova de Aptidão Profissional 2019/20*. Obtido de https://esjs-mafra.net/escola/cursosprofissionais/PAP/sinopse_GPSI_2020.pdf
- European Commission's Staff Working Document (EC-SWD). (2020). Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions Digital Education action Plan 2021-2027. Resetting education and training for the digital age. Obtido de https://education.ec.europa.eu/sites/default/files/document-library-docs/deap-swd-sept2020_en.pdf
- Ferreira, M. (2021). Intencionalidade Educativa-Diferenciação, Planificação e Avaliação, capítulo 3. *Educomunicação Parento-Filial inclusiva: Ciência, Cultura e Cidadania*. 1ª ed. Edições Universitárias Lusófonas. Obtido de https://www.researchgate.net/publication/355394669_Intencionalidade_Educativa_-_Diferenciacao_Planificacao_e_Avaliacao

- Freeman, S., Eddy, S. L., McDonough, M., Smith, M. K., Okoroafor, N., & Jordt, H. &. (2014). Active learning increases student performance in science, engineering, and mathematics. In *Proceedings of the national academy of sciences*, 111(23) (pp. 8410-8415).
- Garry, S., & Checchi, F. (2020). Armed conflict and public health: into the 21st century. *Journal of Public Health*, 42(3) (pp. 287-298).
- Godinho, R. M. (2022). Marcelo promulga fim do uso obrigatório de máscara. *Diário de Notícias* 21/04/2022. Obtido de <https://www.dn.pt/sociedade/mascaras-deixam-de-ser-obrigatorias-nas-escolas-14788325.html>
- Gonçalves, R. (2016). PAP, Rastreador de Localização GPS. Escola Técnica Profissional de Mafra. Obtido de <https://pt.slideshare.net/RicardoGonalves123/gps-location-tracker-apresentao-de-pap-61254165>
- Google Developers. (2022a). *Conheça o Android Studio*. Obtido de <https://developer.android.com/studio/intro?hl=pt-br>
- Google Developers. (2022b). *Codelabs for Android Developer Fundamentals*. Obtido de Google: <https://developer.android.com/courses/fundamentals-training/toc-v2>
- Grande, T. (2016). *Wilcoxon Signed-Rank Test in Excel*. Obtido de <https://www.youtube.com/watch?v=mJtbhGETU88>
- Guggemos, J., Seufert, S., & Román-González, M. (2022). Computational Thinking Assessment–Towards More Vivid Interpretations. *Technology, Knowledge and Learning* (pp. 1-30).
- Harzing, A.W. (2007) Publish or Perish. Obtido de <https://harzing.com/resources/publish-or-perish>
- Hokkanen, I. (2015). *Bloom Taxonomy Verbs and Activities*. Obtido de <https://www.slideshare.net/lidaHokkanen/bloom-taxonomy-action-verbs-and-activities> e <https://www.itcilo.org/es/node/1768>
- Hsu, T. C., Chang, S. C., & Hung, Y. T. (2018). How to learn and how to teach computational thinking: Suggestions based on a review of the literature. *Computers & Education*, 126 (pp. 296-310).
- Inácio, A. M., & Sequeira, P. (2022). Abertura das escolas faz disparar casos. 500 mil em apenas duas semanas. Diário de Notícias. Obtido de Diário de Notícias 20/01/2022: <https://www.dn.pt/sociedade/abertura-das-escolas-faz-disparar-casos-500-mil-em-apenas-duas-semanas-14508000.html>
- IOI, I. O. (2022). IOI Results 2022. Indonesia. Obtido de <https://stats.ioinformatics.org/delegations/2022>
- Iz, H. B., & Fok, H. S. (2007). Use of Bloom's taxonomic complexity in online multiple choice tests in Geomatics education. In *Survey Review*, 39(305) (pp. 226-237).
- Jain, R. (2017). Summarizing Measured Data. Washington University, St Louis, USA. Obtido de https://www.cse.wustl.edu/~jain/cse567-17/ftp/k_12smd.pdf
- Kaoneill. (2020). Computational Thinking. The-elements-of-computational-thinking. Teacher in Training. North Carolina State University, USA. Obtido de <https://koneilleci201.wordpress.ncsu.edu/2020/01/28/computational-thinking/>
- Keller, J. M. (1987). Development and use of the ARCS model of instructional design. *Journal of instructional development*, 10(3) (pp. 2-10). Obtido de <https://ocw.tudelft.nl/wp-content/uploads/Development-and-Use-of-the-ARCS-Model-of-Instructional-Design.pdf>
- Koslosky, M. A. (1999). Aprendizagem Baseada em Casos – Um ambiente para ensino de lógica de programação. Dissertação de Doutorado. Universidade Federal de Santa Catarina, Brasil. Obtido de <https://repositorio.ufsc.br/bitstream/handle/123456789/81012/146842.pdf?sequence=1>
- Liang, Z., Nishi, M., & Kishida, I. (2021, December). Teaching Android App Development to First Year Undergraduates: Textual Programming or Visual Programming? In *2021 IEEE International Conference on Engineering, Technology & Education (TALE)* (pp. 01-08). Obtido de

- https://ieeexplore.ieee.org/abstract/document/9678602?casa_token=_zlpp4I9GIQAAAAA:_ZlwczLtHwXzfpzaVdX0DfmasP-BRK2qMesuw6nOcJ6oejz0P4rsPzPHF9sAgHV04_m-JlPtMk
- Lockwood, J., & Mooney, A. (2018). Developing a computational thinking test using Bebras problems. Obtido em <https://mural.maynoothuniversity.ie/10316/1/AM-Developing-2017.pdf>
- Lopes, J. A. (2003). Problemas de Comportamento, Problemas de Aprendizagem e Problemas de Ensinar. ISBN: 972-8535-92-9. Adaptado por Caires, S. (2021). In *notas de aula da UC Gestão da Sala de Aula* (pp. 107-168). Universidade do Minho.
- Magson, N. R., Freeman, J. Y., Rapee, R. M., Richardson, C. E., Oar, E. L., & Fardouly, J. (2021). Risk and protective factors for prospective changes in adolescent mental health during the COVID-19 pandemic. *Journal of youth and adolescence, 50(1)* (pp. 44-57).
- Markezine. (2022). Grasshopper, o App de Programação para Iniciantes. *Educação Informática*. Obtido de <https://www.markezine.com.br/educacao/grasshopper-o-app-de-programacao-para-iniciantes/>
- Marques, A. S. (2021). Iniciação à Programação no 1.º Ciclo do Ensino Básico: Análise da implementação da medida a alunos do 3.º ano. Dissertação de Doutoramento. Universidade da Beira Interior. Obtido de <https://ubibliorum.ubi.pt/bitstream/10400.6/11156/1/8%20-%20>
- Marsh, H. W., & Roche, L. A. (1997). Making students' evaluations of teaching effectiveness effective: The critical issues of validity, bias, and utility. *American psychologist, 52(11)* (p. 1187). Obtido de <https://psycnet.apa.org/record/1997-43129-003>
- Martins, G. D., Gomes, C. A., Brocardo, J., Pedroso, J. V., Camilo, J. L., Silva, L. M., & Rodrigues, S. M. (2017). Perfil dos alunos à saída da escolaridade obrigatória. Ministério da Educação/DGE. Portugal. Obtido de https://comum.rcaap.pt/bitstream/10400.26/22377/1/perfil_dos_alunos.pdf
- Mateus, P. I. (2021). Aprendizagem baseada em projetos e ensino da programação com recurso ao App Inventor. Dissertação de Mestrado. Universidade de Évora.
- McLean, S. F. (2016). Case-based learning and its application in medical and health-care fields: a review of worldwide literature. *Journal of medical education and curricular development, 3, JMECD-S20377*.
- MEC PT, M. d. (2021). *Despacho n.º 8209/2021 de 19 de agosto. D. R. 2ª série C. 161 p. 115-116*. Obtido de <https://dre.pt/dre/detalhe/despacho/8209-2021-169831748>
- Merseth, K. K. (1991). The early history of case-based instruction: Insights for teacher education today. *Journal of Teacher Education, 42(4)* (pp. 243-249).
- Morais, M. (2022). Proposta de Utilização da App Grasshopper na Escolaridade Obrigatório no Âmbito da Lógica de Programação – Iniciação à Programação de Computadores. UC ACMDI. Universidade do Minho. Obtido de <https://acmdi-grasshopper.blogspot.com/2022/02/>
- Morais, M., Gonçalves, C., & Reis, P. (2021). Aplicação da Metodologia PRIMM em Trabalhos de Computação Física. UC Metodologia de Ensino de Informática II. Universidade do Minho. Obtido de <http://www.mei2.pt/>
- Moreno-León, J., Robles, G., Román-González, M., & Rodríguez García, J. D. (2019). Not the same: a text network analysis on computational thinking definitions to study its relationship with computer programming. *Revista Interuniversitaria de Investigación en Tecnología Educativa (RIITE), 7* (pp. 26-35)
- Moura, A., & Carvalho, A. A. (2010). Enquadramento teórico para integração de tecnologias móveis em contexto educativo. *ticEDUCA2010 – I Encontro Internacional TIC e Educação.*, 19-20 Novembro, IE-UL, Lisboa. Obtido de <https://>

- <http://repositorio.uportu.pt:8080/jspui/bitstream/11328/454/2/Enquadramento%20te%C3%B3rico%20para%20a%20%20integra%C3%A7%C3%A3o.2010.pdf>
- Neto, L. D. (2020). Quem ou o que pensa? Uma busca de aportes para questões filosóficas suscitadas pela revolução informática atual. Universidade Federal de São Paulo, Brasil. Obtido de <https://philpapers.org/archive/DEHQ00.pdf>
- Null, L., & Lobur, J. ... (2019). *Essentials of Computer Organization and Architecture. 5th edition*. Jones & Bartlett Publishers. ISBN: 9781284136852.
- Nunes, J. M. (2019). Mobile learning e pensamento computacional: contributos para o desenvolvimento de aplicações em contextos educativos. Dissertação de Doutoramento. Universidade Aberta.
- Olimpiadas Nacionais de Informática (ONI). (2017). Obtido de <https://oni.dcc.fc.up.pt/2017/> e <https://apdsi.pt/2022/olimpiadas-de-informatica-2022-ja-sao-conhecidos-os-quatro-alunos-que-irao-representar-portugal-nas-olimpiadas-internacionais-de-informatica/>
- Pedro, A., Mato, J. F., Piedade, J., & Dorotea, N. (2017). Probótica, Programação e Robótica no Ensino Básico. . Instituto de Educação, Universidade de Lisboa. Obtido de https://www.dge.mec.pt/sites/default/files/ERTE/origramacao_robotica/probotica_-_linhas_orientadoras_2017.pdf
- Peracaula-Boscha, M., & González-Martínez, J. (2022). Developing Computational Thinking Among Preservice Teachers. *ATEE* (pp. 57-60). Obtido de <https://library.oapen.org/bitstream/handle/20.500.12657/55676/9788855184120.pdf?sequence=1#page=59>
- Pereira, A. (2021). Eles não sabem nem sonham, que a Educação comanda a Vida! Dissertação de Mestrado. Instituto Politécnico do Porto. Obtido de <https://recipp.ipp.pt/handle/10400.22/19582>
- Pereira, P. A. (2021). Programação de dispositivos móveis. Programa de UC, ISEL-PT. Obtido de <https://www.isel.pt/disciplinas/programacao-de-dispositivos-moveis-leirt>
- Pérez-Poch, A. (2021). Programación de Dispositivos Móviles: una propuesta docente adaptada a un entorno híbrido. Asociación de Enseñantes Universitarios de la Info (AENUI). In *Actas de las XXVII Jornadas sobre la Enseñanza Universitaria de la Informática* (pp. 211-218). Obtido de <https://upcommons.upc.edu/handle/2117/350168>
- Piedade, J., & Dorotea, N. A. (2020). A Robótica Educacional como Recurso Pedagógico para Aprender Programação e Desenvolver Competências de Pensamento Computacional: Práticas de Futuros de Informática. In *Formação no Contexto do Pensamento Computacional, da Robótica e da Inteligência Artificial na Educação, 89*. Obtido de https://www.researchgate.net/profile/Joao-Piedade-4/publication/347489511_Formacao_no_contexto_do_pensamento_computacional_da_robotica_e_da_inteligencia_artificial_na_educacao/links/5fddf3c092851c13fe9d4426/
- Portugal News. (24 de Fevereiro de 2022). *RTP interrompe programação devido à guerra na Ucrânia*. Obtido de <https://portugalnews.online/rtp/rtp-interrompe-programacao-devido-a-guerra-na-ucrania/>
- Quadros-Flores, P. (2011). Os Dez Princípios de uma Boa Prática com TIC. A par dos tempos que correm, as TIC e o centenário da República. Obtido de https://recipp.ipp.pt/bitstream/10400.22/6333/1/ART_PaulaFlores_2011.pdf
- Queirós, R. (2016). *Android. Desenvolvimento de Aplicações com Android Studio* (p. 291). FCA. Lisboa. ISBN 978-972-722-819-5.

- Rádio e Televisão de Portugal (RTP). (2022). Refugiados Ucrânicos, chegaram dois autocarros a Portugal. Obtido de https://www.rtp.pt/noticias/pais/refugiados-ucranianos-chegaram-dois-autocarros-a-portugal_v1389562
- Ramos, J. L., Espadeiro, R. G., & Monginho, R. (2022). Introdução à programação, robótica e ao pensamento computacional na educação pré-escolar e 1.º ciclo do ensino básico. Necessidades de formação de educadores e professores. Universidade de Évora: ISBN 978-972-778-252-9. Obtido de <https://digital.dge.mec.pt/sites/default/files/documents/2022/177-14551b4676f7711407147b60af2f9a3f.pdf>
- Reis, D. R. (2015). Prova de Aptidão Profissional. Tema S-Notas. Escola Secundária de Tomaz Pelayo, Santo Tirso. Obtido de <https://prezi.com/l3c3kssmqfnp/prova-de-aptidao-profissional/>
- República Portuguesa. (2022). *Testes à Covid-19 nas escolas para o ano letivo 2021/2022*. Obtido de <https://www.portugal.gov.pt/pt/gc22/comunicacao/noticia?i=testes-a-covid-19-nas-escolas-para-o-ano-letivo-20212022>
- Rich, P. J., Mason, S. L., & O'Leary, J. (2021). Measuring the effect of continuous professional development on elementary teachers' self-efficacy to teach coding and computational thinking. *Computers & Education, 168*, 104196.
- Román González, M. (2016). Codigoalfabetización y pensamiento computacional en educación primaria y secundaria: validación de un instrumento y evaluación de programas. Dissertação de Doutoramento. UNED, Espanha. Obtido de <https://redined.educacion.gob.es/xmlui/hand>
- Román-González, M., Moreno-León, J. & Robles, G. (2019). Combining assessment tools for a comprehensive evaluation of computational thinking interventions. In *Computational Thinking Education* (pp. 79-98). Springer, Singapore.
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in human behavior, 72*, 678-691.
- Rosado, N. F. (2020). Estratégias de aprendizagem da programação na educação básica e secundária: um estudo exploratório com recurso ao método de "pair programming". Dissertação de Mestrado. Universidade de Évora.
- Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology, 111*(5), 764.
- School of Informatics. (2016). What is informatics? University of Edinburg. Obtido de <https://www.ed.ac.uk/sites/default/files/atoms/files//what20is20informatics.pdf>
- Sentance, S., & Waite, J. (2017, November). PRIMM: Exploring pedagogical approaches for teaching text-based programming in school. In *Proceedings of the 12th Workshop on Primary and Secondary Computing Education* (pp. 113-114).
- Sentance, S., & Waite, J. (2021, August). Teachers' Perspectives on Talk in the Programming Classroom: Language as a Mediator. In *Proceedings of the 17th ACM Conference on International Computing Education Research* (pp. 266-280).
- Sentance, S., Waite, J., & Kallia, M. (2019). Teaching computer programming with PRIMM: a sociocultural perspective. *Computer Science Education, 29*(2-3) (pp. 136-176).
- Shute, V. J., Sun, C., & Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review, 22* (pp. 142-158).
- Silva, B. D., Ramos, M. A., Lencastre, J. A., & Bento, M. (2020). Utilização inovadora de dispositivos móveis no processo educativo. In *Patrícia Lupion Torres (org.) Ciência, Inovação e Ética: tecendo redes e conexões para a produção do conhecimento* (pp. 331-357). Curitiba: SENAR-PR.

- Sitthiworachart, J., & Joy, M. (2008). Computer support of effective peer assessment in an undergraduate programming class. In *Journal of computer assisted learning*, 24(3) (pp. 217-231).
- Sousa, S. I. (2016). Utilização da estratégia Pair Programming no ensino da programação de sistemas de comunicação com recurso a sockets. Dissertação de Mestrado. Universidade de Lisboa. Obtido de <https://repositorio.ul.pt/bitstream/10451/24621/1/ulfpie051099>
- SPM, S. P. (2022). Parecer da Sociedade Portuguesa de Matemática à Proposta de Revisão Curricular das Aprendizagens Essenciais de Matemática A para o Ensino Secundário. 15/07/2022. Obtido de <https://www.spm.pt/files/images/documentos/pareceres/SPM%20Parecer%20Aes%20Ensino%20Secundario%20%202022%20%20.pdf>
- Stat. (2022). Critical Values of the Wilcoxon Signed Ranks Test. University of Florida, USA. Obtido de https://users.stat.ufl.edu/~winner/tables/wilcox_signrank.pdf
- Tafur-Morales, C., Gabriel-Cheng, D., & Cabanillas-Carbonell, M. (2021, November). Mobile App to Make Better the Obesity Prevention Process in Adolescents. In *2021 International Conference on e-Health and Bioengineering (EHB)* (pp. 1-4). IEEE.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148, 103798.
- Tenenaula, J. A., & Buñay, A. M. (2020). Correlación entre perfil lipídico y medidas antropométricas en adolescentes de cuatro Unidades Educativas de Riobamba. Bachelor's thesis. Universidad Nacional de Chimborazo, Ecuador. Obtido de <http://dspace.unach.edu.ec/handle/51000/6664>
- TERÃŠA. (2022). Mudanças a Matemática do Ensino Secundário entram em Discussão Pública. In *Diário de Aveiro*. 07/06/2022. Obtido de <https://www.diarioaveiro.pt/noticia/83511>
- Toporov, E. (2013). *IntelliJ IDEA is the base for Android Studio, the new IDE for Android developers*. Obtido de JetBrains: <https://blog.jetbrains.com/blog/2013/05/15/intellij-idea-is-the-base-for-android-studio-the-new-ide-for-android-developers/>
- Tóth, T., & Lovászová, G. (2021). Visual Vs. Textual Programming: A Case Study on Mobile Application Programming by Teenagers. In *Ad Alta: Journal of Interdisciplinary Research*.
- Weintrop, D., & Wilensky, U. (2015, August). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. In *Proceedings of the eleventh annual international conference on international computing education research* (pp. 101-110).
- Wijnia, L., Loyens, S. M., Noordzij, G., Arends, L. R., & Rikers, R. M. (2017). The effects of problem-based, project-based, and case-based learning on students' motivation: A meta-analysis. *Eindrapport NRO-project* (pp. 405-15).
- Wing, J. (2017). Computational thinking's influence on research and education for all. In *Italian Journal of Educational Technology*, 25(2) (pp. 7-14). Ortona, Italy: Edizioni Menabò. Obtido de <https://www.learntechlib.org/p/183466/>
- Yeager, K. (2022). Spss Tutorials: Paired Samples T Test. Kent State University, USA. Obtido de <https://libguides.library.kent.edu/spss/pairedsamplestest>
- Yunlong, J., & LIU, Y. (2022). Structure and effects of motivation: From the perspective of the motivation continuum. In *Advances in Psychological Science*, 30: 1589-1603 doi: 10.3724/SP.J.1042.2022.01589. Obtido de <https://journal.psych.ac.cn/xlkxjz/EN/10.3724/SP.J.1042.2022.01589>

Apêndices

Apêndice I

Teste de Pensamento Computacional (Román-González, 2016)

- O teste é composto por 28 perguntas, distribuídas por 16 páginas com 2 perguntas cada.
- Todas as perguntas têm 4 opções de resposta (A, B, C ou D) **das quais apenas uma é a correta.**
- Não é necessário responder a todas as perguntas.

Número Institucional de aluno: _____

Questão 1:

Qual a seqüência de passos que leva o Pac-Man pelo caminho até ao Fantasma?	
	Opção A
	Opção B
	Opção C
	Opção D

Questão 1: A B C D

Questão 2:

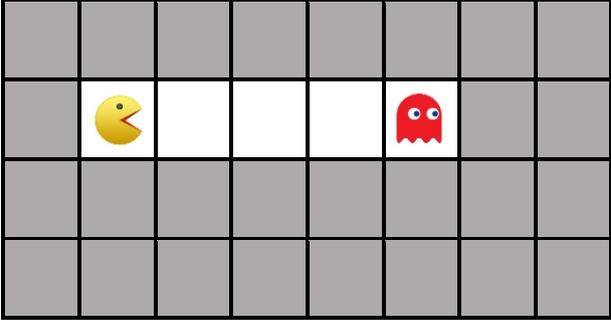
Que passo falta na seqüência para levar o Pac-Man pelo caminho até ao Fantasma?	
	Opção A
	Opção B
	Opção C
	Opção D

Questão 2: A B C D

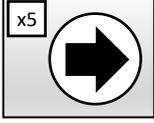
Questão 5:

Qual a sequência que leva o Pac-Man pelo caminho até ao Fantasma?

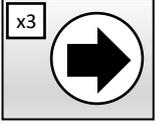
Número de repetições



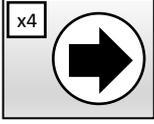
Opção A



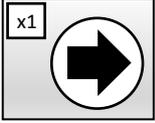
Opção B



Opção C



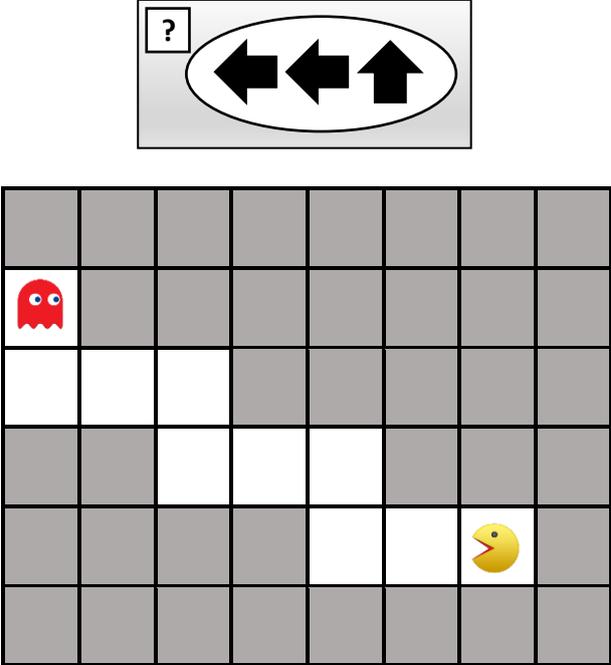
Opção D



Questão 5: A B C D

Questão 6:

Quantas vezes se deve repetir esta sequência para levar o Pac-Man pelo caminho até ao Fantasma?



Opção A

x 2

Opção B

x 1

Opção C

x 4

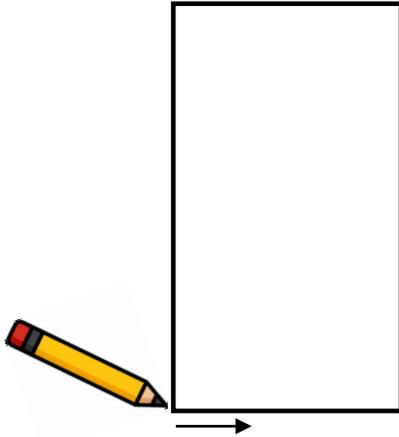
Opção D

x 3

Questão 6: A B C D

Questão 7:

Para que o Lápis desenhe o seguinte retângulo (50 pixéis de largura, 100 pixéis de altura), qual a linha da sequência que está errada?



```
for (i=0; i < 4; i++)      // Linha A
{
    moveParaFrente(50);
    viraEsquerda(90);      // Linha B
    moveParaFrente(100);   // Linha C
    viraEsquerda(90);     // Linha D
}
```

Questão 7:

 A

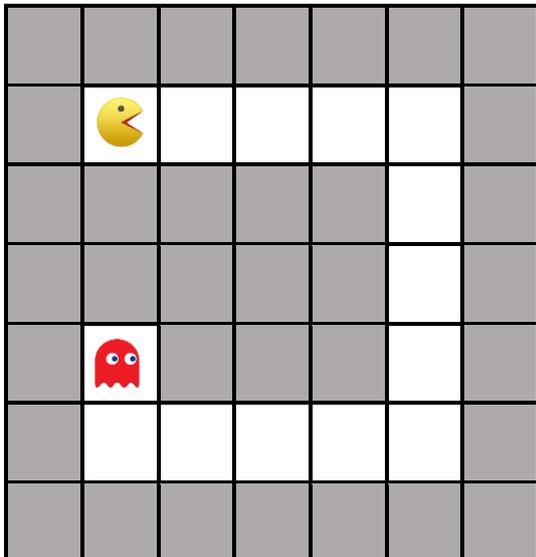
 B

 C

 D

Questão 8:

Qual a sequência que leva o Pac-Man pelo caminho até ao Fantasma?



Opção A

```
for (i=0; i < 4; i++)
{
    for (i=0; i < 3; i++)
    {
        moveParaFrente();
    }
    viraDireita();
}
moveParaFrente();
```

Opção B

```
for (i=0; i < 3; i++)
{
    for (i=0; i < 4; i++)
    {
        moveParaFrente();
    }
    viraDireita();
}
moveParaFrente();
```

Opção C

```
for (i=0; i < 3; i++)
{
    for (i=0; i < 4; i++)
    {
        moveParaFrente();
        viraDireita();
    }
}
moveParaFrente();
```

Opção D

```
for (i=0; i < 4; i++)
{
    moveParaFrente();
}
for (i=0; i < 3; i++)
{
    viraDireita();
}
moveParaFrente();
```

Questão 8:

 A

 B

 C

 D

Questão 9:

Qual a sequência que leva o Pac-Man pelo caminho até ao Fantasma?

	<p>Opção A</p> <p>Repetir até </p>	<p>Opção B</p> <p>Repetir até </p>
	<p>Opção C</p> <p>Repetir até </p>	<p>Opção D</p> <p>Repetir até </p>

Questão 9: A B C D

Questão 10:

Qual a linha que falta na sequência para que o Pac-Man chegue até ao Fantasma?

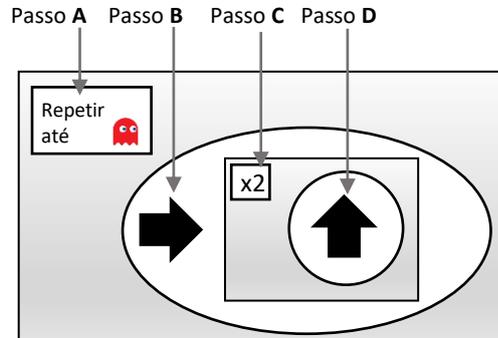
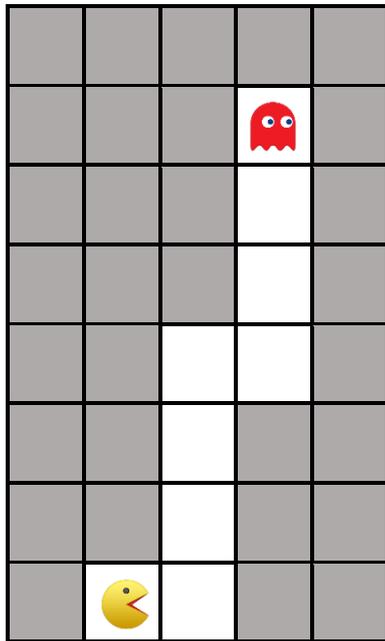
```
do{
  moveParaFrente();
  moveParaFrente();
  _____?
  viraDireita();
  moveParaFrente();
  viraEsquerda();
} while (fantasma = 1);
```

	<p>Opção A</p> <p>viraEsquerda();</p>
	<p>Opção B</p> <p>viraDireita();</p>
	<p>Opção C</p> <p>moveParaFrente();</p>
	<p>Opção D</p> <p>Não falta nenhuma linha</p>

Questão 10: A B C D

Questão 11:

Para que o Pac-Man chegue até ao Fantasma, qual o passo da sequência que está **errado**?



Questão 11:

A

B

C

D

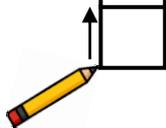
Questão 12:

Que sequência deve o lápis seguir para desenhar a escada até perfazer um quadrado na Lâmpada? Cada degrau sobe 30 pixéis. Cada ângulo tem 90 graus.

moveParaFrente(60);



moveParaFrente(30);
saltaParaFrente(30);



Opção A

```
do {
  for (i=0; i < 4; i++) {
    moveParaFrente(30);
    viraDireita(90);
  }
  saltaParaFrente(30);
} while (lampada = 0);
```

Opção B

```
do {
  for (i=0; i < 4; i++) {
    moveParaFrente(120);
    viraDireita(90);
  }
  saltaParaFrente(30);
} while (lampada = 0);
```

Opção C

```
do {
  for (i=0; i < 4; i++) {
    moveParaFrente(30);
    viraDireita(90);
  }
  saltaParaFrente(210);
} while (lampada = 0);
```

Opção D

```
do {
  for (i=0; i < 7; i++) {
    moveParaFrente(30);
    viraDireita(90);
  }
  saltaParaFrente(30);
} while (lampada = 0);
```

Questão 12:

A

B

C

D

Questão 13:

Qual a sequência que leva o Pac-Man pelo caminho até ao Fantasma?

	<p>Opção A</p>	<p>Opção B</p>
	<p>Opção C</p>	<p>Opção D</p>

Questão 13: A B C D

Questão 14:

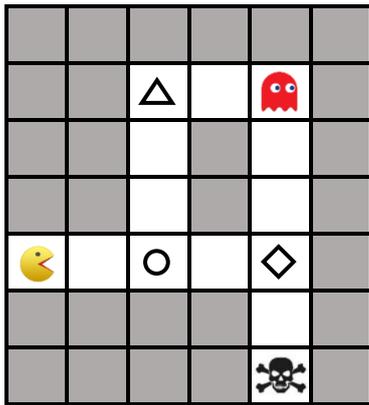
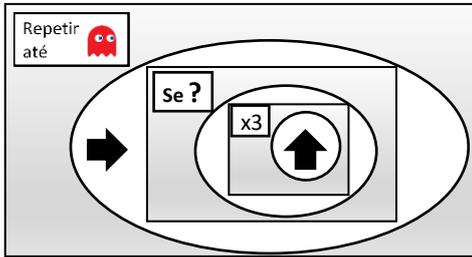
Qual a sequência que leva o Pac-Man pelo caminho até ao Fantasma?

	<p>Opção A</p> <pre>do { moveParaFrente(); if (haCaminhoDireita()) { viraDireita(); } } while (fantasma = 1);</pre>	<p>Opção B</p> <pre>do { viraDireita(); if (haCaminhoDireita()) { moveParaFrente(); } } while (fantasma = 1);</pre>
	<p>Opção C</p> <pre>do { moveParaFrente(); if (haCaminhoDireita()) { viraEsquerda(); } } while (fantasma = 1);</pre>	<p>Opção D</p> <pre>do { moveParaFrente(); if(haCaminhoEsquerda()) { viraEsquerda(); } } while (fantasma = 1);</pre>

Questão 14: A B C D

Questão 15:

Que condição falta na sequência (Se _?) para levar o Pac-Man pelo caminho até ao Fantasma?



Opção A



Opção B



Opção C



Opção D

Tanto a opção A como a opção C estão corretas

Questão 15:

 A

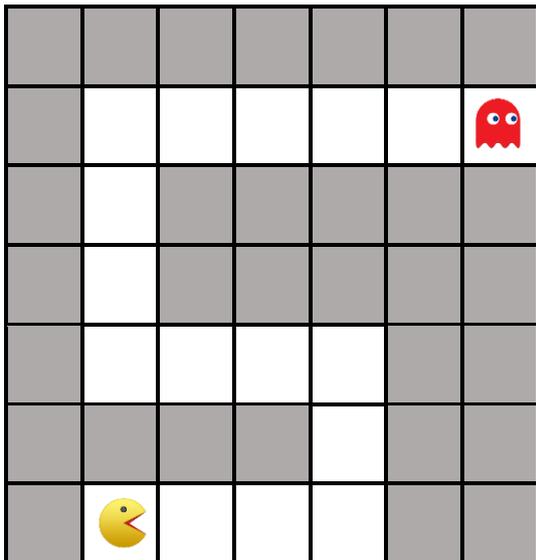
 B

 C

 D

Questão 16:

Para que o Pac-Man chegue até ao Fantasma, qual a linha da sequência que está **errada**?



```
do {
  moveParaFrente();
  if (haCaminhoEsquerda()) // Linha A
  {
    viraEsquerda(); // Linha B
  }
  if (haCaminhoDireita()) // Linha C
  {
    moveParaFrente(); // Linha D
  }
}
```

Questão 16:

 A

 B

 C

 D

Questão 17:

Qual a sequência que leva o Pac-Man pelo caminho até ao Fantasma?						
	<p>Opção A</p> <pre>do { if(haCaminhoFrente()) { moveParaFrente(); } else { viraEsquerda(); } } while (fantasma=1);</pre>			<p>Opção B</p> <pre>do { if(haCaminhoFrente()) { moveParaFrente(); } else { viraDireita(); } } while (fantasma=1);</pre>		
	<p>Opção C</p> <pre>do { if(haCaminhoDireita()) { viraDireita(); } else { moveParaFrente(); } } while (fantasma=1);</pre>			<p>Opção D</p> <pre>do { if(haCaminhoEsquerda()) { viraEsquerda(); } else { moveParaFrente(); } } while (fantasma=1);</pre>		

Questão 17: A B C D

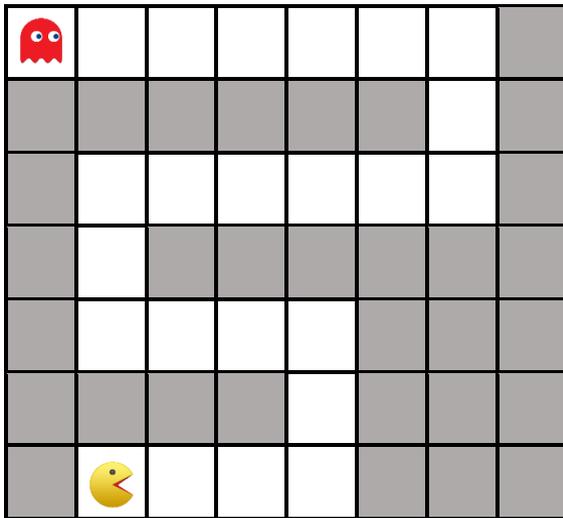
Questão 18:

Qual a sequência que leva o Pac-Man pelo caminho até ao Fantasma?						
	<p>Opção A</p> <pre>do { if(haCaminhoFrente()) { moveParaFrente(); } else { viraEsquerda(); } } while (fantasma=1);</pre>			<p>Opção B</p> <pre>do { if(haCaminhoFrente()) { moveParaFrente(); } else { viraDireita(); } } while (fantasma=1);</pre>		
	<p>Opção C</p> <pre>do { if(haCaminhoDireita()) { viraDireita(); } else { moveParaFrente(); } } while (fantasma=1);</pre>			<p>Opção D</p> <pre>do { if(haCaminhoEsquerda()) { viraEsquerda(); } else { moveParaFrente(); } } while (fantasma=1);</pre>		

Questão 18: A B C D

Questão 19:

Para que o Pac-Man chegue até ao Fantasma, qual a linha da sequência que está **errada**?



```
do {
  if (haCaminhoFrente())
  {
    moveParaFrente();      // Linha A
  } else {
    if (haCaminhoDireita()) // Linha B
    {
      viraEsquerda();      // Linha C
    } else {
      viraDireita();       // Linha D
    }
  }
} while (fantasma = 1)
```

Questão 19:

 A

 B

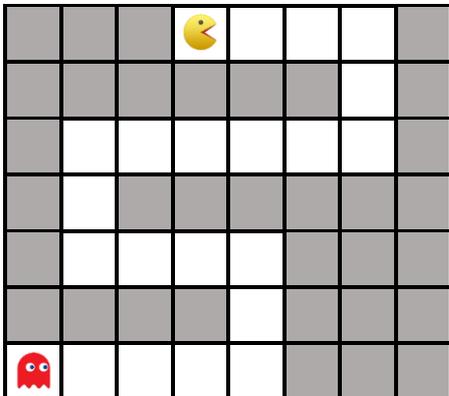
 C

 D

Questão 20:

Qual a linha que falta na sequência para levar o Pac-Man pelo caminho até ao Fantasma?

```
do {
  if (haCaminhoFrente())
  {
    moveParaFrente();
  } else {
    if (haCaminhoDireita())
    {
      viraDireita();
    } else {
      _____?
    }
  }
} while (fantasma = 1)
```



Opção A

moveParaFrente();

Opção B

viraDireita();

Opção C

viraEsquerda();

Opção D

Não falta nenhuma linha

Questão 20:

 A

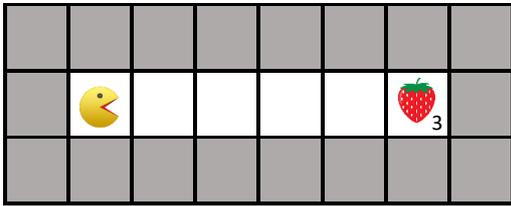
 B

 C

 D

Questão 21:

Qual a sequência que leva o Pac-Man pelo caminho até aos Morangos, e faz com que ele coma o número de Morangos indicado?



Opção A

```
while
(haCaminhoFrente())
{
  moveParaFrente();
}
for (i=0; i<3; i++)
{
  comeMorango();
}
```

Opção B

```
while
(haCaminhoFrente())
{
  moveParaFrente();
}
for (i=0; i<4; i++)
{
  comeMorango();
}
```

Opção C

```
while
(haCaminhoFrente())
{
  moveParaFrente();
}
for (i=0; i<5; i++)
{
  comeMorango();
}
```

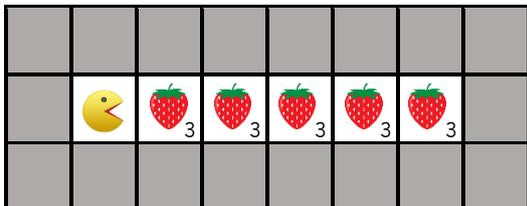
Opção D

```
while
(haCaminhoFrente())
{
  moveParaFrente();
  for (i=0; i<3; i++)
  {
    comeMorango();
  }
}
```

Questão 21: A B C D

Questão 22:

Qual a sequência que leva o Pac-Man pelo caminho até aos Morangos, e faz com que ele coma o número de Morangos indicado?



Opção A

```
while
(haCaminhoFrente())
{
  for (i=0; i<5; i++)
  {
    moveParaFrente();
  }
  for (i=0; i<3; i++)
  {
    comeMorango();
  }
}}
```

Opção B

```
while
(haCaminhoFrente())
{
  moveParaFrente();
  for (i=0; i<3; i++)
  {
    comeMorango();
  }
}
```

Opção C

```
while
(haCaminhoFrente())
{
  for (i=0; i<3; i++)
  {
    moveParaFrente();
  }
  for (i=0; i<5; i++)
  {
    comeMorango();
  }
}}
```

Opção D

```
while
(haCaminhoFrente())
{
  moveParaFrente();
}
for (i=0; i<3; i++)
{
  comeMorango();
}
```

Questão 22: A B C D

Questão 23:

O que falta na sequência para o levar Pac-Man a comer os Morangos pelo caminho?	
<pre>while (haCaminhoFrente()) { for (i=0; ____? ; i++) { moveParaFrente(); } if (haMorango()) { comeMorango(); } }</pre>	<p>Opção A</p> <p>i < 1</p>
	<p>Opção B</p> <p>i < 2</p>
	<p>Opção C</p> <p>i < 3</p>
	<p>Opção D</p> <p>i < 5</p>

Questão 23: A B C D

Questão 24:

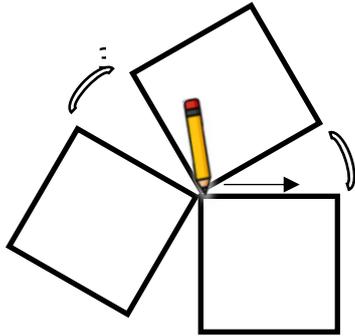
O que falta na sequência (____?) para levar o Pac-Man a comer os Morangos pelo caminho?	
<pre>while (haCaminhoFrente()) { moveParaFrente(); if (haMorango()) { _____? { comeMorango(); } } }</pre>	<p>Opção A</p> <p>while (haCaminhoFrente())</p>
	<p>Opção B</p> <p>while (haCaminhoFrente()==false)</p>
	<p>Opção C</p> <p>while (haMorango())</p>
	<p>Opção D</p> <p>while (haMorango()==false)</p>

Questão 24: A B C D

Questão 25:

Se tivermos uma função que desenha um quadrado (100 pixéis de lado), qual a sequência que o Lápis deve executar para desenhara seguinte figura?

```
desenhaQuadrado(){
  for (i=0; i<4; i++)
  {
    moveParaFrente(100);
    viraDireita(90);
  }
}
```



Opção A

```
for (i=0; i<3; i++)
{
  desenhaQuadrado();
}
viraDireita(120);
```

Opção B

```
for (i=0; i<3; i++)
{
  desenhaQuadrado();
  viraDireita(120);
}
```

Opção C

```
for (i=0; i<3; i++)
{
  desenhaQuadrado();
}
viraDireita(90);
```

Opção D

```
for (i=0; i<3; i++)
{
  desenhaQuadrado();
  viraDireita(90);
}
```

Questão 25:

 A

 B

 C

 D

Questão 26:

Se tivermos uma função que desenha um triângulo equilátero (50 pixéis de lado), o que falta na sequência para que o Lápis desenhe a seguinte figura?

```
desenhaTriangulo(){
  for (i=0; i<3; i++)
  {
    moveParaFrente(50);
    viraEsquerda(120);
  }
}

for (i=0; ____? ; i++)
{
  desenhaTriangulo();
  saltaParaFrente(50);
}
```



Opção A

```
i < 15
```

Opção B

```
i < 5
```

Opção C

```
i < 4
```

Opção D

```
i < 3
```

Questão 26:

 A

 B

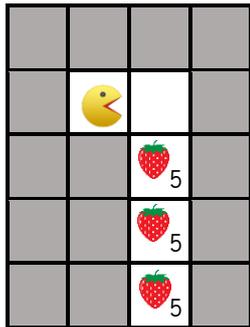
 C

 D

Questão 27:

Se tivermos a função “come5Morangos()”, qual a sequência que leva o Pac-Man pelo caminho e faz com que ele coma o número de Morangos indicado?

```
come5Morangos(){
  for (i=0; i<5 ; i++)
  {
    comeMorango();
  }
}
```



Opção A

```
moveParaFrente();
viraDireita();
for (i=0; i<3 ; i++)
{
  moveParaFrente();
  come5Morangos();
}
```

Opção B

```
moveParaFrente();
viraDireita();
for (i=0; i<3 ; i++)
{
  come5Morangos();
}
moveParaFrente();
```

Opção C

```
moveParaFrente();
viraDireita();
for (i=0; i<5 ; i++)
{
  moveParaFrente();
  come5Morangos();
}
```

Opção D

```
moveParaFrente();
viraDireita();
for (i=0; i<5 ; i++)
{
  come5Morangos();
}
moveParaFrente();
```

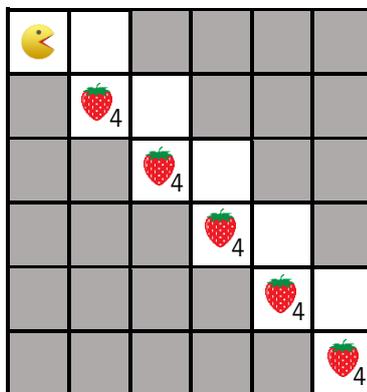
Questão 27: A B C D

Questão 28:

Se tivermos a função “andaCome4Morangos”, o que falta na sequência (___?) para fazer com que o Pac-Man siga pelo caminho e coma todos os Morangos?

```
andaCome4Morangos(){
  moveParaFrente();
  viraDireita();
  moveParaFrente();
  for (i=0; i<4 ; i++)
  {
    comeMorango();
  }
  viraEsquerda();
}
```

```
for(i=0; ___?; i++)
{
  andaCome4Morangos();
}
```



Opção A

```
i < 3
```

Opção B

```
i < 4
```

Opção C

```
i < 5
```

Opção D

```
i = 5
```

Questão 28: A B C D

Apêndice II



Declaração de Consentimento Informado, Livre e Esclarecido

Exm.º Senhor (a) Encarregado (a) de Educação,

O meu nome é Maurício Braga Morais e frequento o Estágio Profissional do Mestrado em Ensino de Informática, da Universidade do Minho, sob a orientação dos Professores Altina Ramos, da Universidade do Minho, e José Pereira, da Escola Secundária [REDACTED].

Neste contexto, estou a desenvolver, integrado na disciplina de Programação de Dispositivos Móveis, um projeto de investigação intitulado “Em que medida o Android Studio e a metodologia baseada em casos contribuem para a aprendizagem de programação e para o desenvolvimento do pensamento computacional?”

Esta investigação tem os seguintes objetivos:

- Promover o uso consciente, pelos alunos, dos princípios e procedimentos do pensamento computacional para que estejam aptos a acompanhar os avanços tecnológicos que virão.
- Contribuir para o aumento da motivação no estudo da programação.
- Identificar a contribuição da intervenção pedagógica no desenvolvimento do pensamento computacional dos alunos

Para realizar esta investigação, há a necessidade de efetuar uma recolha de dados através da aplicação de um questionário individual e anónimo aos alunos. O questionário é de natureza puramente técnica, não contém perguntas pessoais. Em geral, as perguntas do questionário têm um enunciado com uma imagem e são oferecidas quatro hipóteses de solução, das quais o aluno deve escolher uma.

Além do questionário, vai decorrer uma entrevista do tipo *focus group*, ou seja, uma entrevista curta, sobre a temática do estudo, em um pequeno grupo, com recurso a gravação áudio.

Consoante o andamento do projeto, poderá ser necessário fazer gravações áudio e vídeo de algumas aulas nomeadamente durante a realização das atividades que serão propostas aos alunos, para facilitar a posterior análise dos dados recolhidos.

Esclareço desde já que estes recursos audiovisuais têm apenas um interesse documental para efeitos de investigação e para que eu possa, posteriormente, analisar as situações de aula. Não serão divulgados por meio

nenhum, manter-se-ão confidenciais. Se necessário, poderão ser consultados pelos orientadores, sempre em total sigilo investigativo.

Todos os dados audiovisuais serão guardados apenas durante o tempo em que se mantiverem necessários para a publicação da tese e de trabalhos relacionados com este estudo, sendo destruídos após esse tempo. Além disso, a identidade dos alunos, bem como os seus rostos serão sempre preservados, já que nunca serão referidos os seus nomes, mostrados os seus rostos nem será identificada a escola onde o trabalho será realizado.

Por fim, uma vez terminado o meu trabalho pedagógico, preciso que os alunos avaliem os vários aspetos da minha atuação e avaliem a eles mesmos. Este questionamento vai ocorrer por meio de um inquérito individual e anónimo, no qual os alunos selecionam as opções que lhes parecerem mais adequadas.

De modo a autorizar esta recolha de dados, peço que assine o seguinte termo de consentimento informado. O seu educando também deve aceitar a participação neste estudo, pelo que deve também ele assinar este termo de consentimento informado. **A participação neste estudo é completamente voluntária, e qualquer participante pode abandonar o estudo a qualquer momento, sem qualquer tipo de problema associado.**

Grato pela atenção,

Com os melhores cumprimentos,

O investigador,

.....
Maurício Braga Morais

Data: 14/02/2022

Para mais esclarecimentos, poderá contactar-me por:

Contacto telefónico: ■■■ ■■■ ■■■

Email:pg40625@alunos.uminho.pt

Declaração de Consentimento Informado, Livre e Esclarecido

Eu,, encarregado(a) de educação do(a) aluno(a), da turma, declaro ter lido e compreendido este documento, e autorizo/não autorizo (sublinhar a sua opção) a recolha de dados presentes nesta declaração de consentimento informado, livre e esclarecido, nos moldes que foram aqui apresentados, permitindo a utilização dos dados de forma voluntária, confiando em que apenas serão utilizados para esta investigação e nas garantias de confidencialidade e anonimato que me são dadas pelo/a investigador/a.

....., de de 2022

Assinatura do Encarregado de Educação

.....

Eu,, aluno(a) da turma..., aceito participar neste estudo e contribuir para a recolha de dados que me for solicitada, de acordo com o que está afirmado nesta declaração de consentimento informado, livre e esclarecido.

....., de de 2022

Assinatura do Aluno

.....

ESTE DOCUMENTO É COMPOSTO DE 3 PÁGINA/S E FEITO EM DUPLICADO:
UMA VIA PARA O/A INVESTIGADOR/A, OUTRA PARA A PESSOA QUE CONSENTE

Apêndice III

Grelha de Observação dos Alunos

Curso 481040, UFCD 0817, 11º ano TPI, Escola [REDACTED]

Data: _____ Aula: _____ Turno: _____

Nome do Aluno	Assiduidade e Pontualidade	Comportamento	Participação	Empenho	Autonomia

Preenchimento da grelha:

- Assiduidade e Pontualidade:

P = Presente; F = Faltou; FP = Falta de Pontualidade; FJ = Falta Justificada; FI = Falta injustificada.

- Comportamento: A= Adequado; I = Inadequado; ? = Não consegui observar.

- Participação, Empenho, Autonomia: S = Sim; N/A = Não se Aplica; ? = Não consegui observar; N = Não.

Observações

Apêndice IV

**Questionário de Avaliação
da Qualidade do Professor Estagiário, do Curso e Autoavaliação do Estudante**

MBM (2022), inspirado em Pérez Poch (2021) e Marsh & Roche(1997)

Curso 481040, UFCD 0817, 11º ano TPI, Escola [REDACTED]

Âmbito de avaliação	A tua opinião
Qualidade da aprendizagem	Aprendi coisas que considero valiosas <input type="checkbox"/> Não; <input type="checkbox"/> Sim
Entusiasmo	O professor conseguiu que as suas aulas fossem agradáveis <input type="checkbox"/> Raramente; <input type="checkbox"/> Às vezes; <input type="checkbox"/> Muitas vezes; <input type="checkbox"/> Quase sempre
Organização	As explicações do professor foram claras <input type="checkbox"/> Raramente; <input type="checkbox"/> Às vezes; <input type="checkbox"/> Muitas vezes; <input type="checkbox"/> Quase sempre
Interação com o grupo	O professor estimulava as perguntas do grupo e dava respostas satisfatórias <input type="checkbox"/> Raramente; <input type="checkbox"/> Às vezes; <input type="checkbox"/> Muitas vezes; <input type="checkbox"/> Quase sempre
Atitude pessoal	O professor mostrou-se acessível aos estudantes <input type="checkbox"/> Raramente; <input type="checkbox"/> Às vezes; <input type="checkbox"/> Muitas vezes; <input type="checkbox"/> Quase sempre
Conteúdo	O professor apresentou vários pontos de vista quando foi necessário <input type="checkbox"/> Raramente; <input type="checkbox"/> Às vezes; <input type="checkbox"/> Muitas vezes; <input type="checkbox"/> Quase sempre
Exames	Os métodos de avaliação do curso foram equitativos e adequados <input type="checkbox"/> Não; <input type="checkbox"/> Sim
Trabalhos do curso	A bibliografia e o material recomendado para este curso são corretos e adequados <input type="checkbox"/> Não; <input type="checkbox"/> Sim
Carga de trabalho e dificuldade	Este curso comparado com outros de informática, foi: <input type="checkbox"/> Muito fácil; <input type="checkbox"/> Fácil; <input type="checkbox"/> Normal; <input type="checkbox"/> Difícil; <input type="checkbox"/> Muito difícil
Visão Geral	A nota geral que dás ao professor e ao curso é: <input type="checkbox"/> Menor que 5; <input type="checkbox"/> entre 6 e 10; <input type="checkbox"/> entre 11 e 14; <input type="checkbox"/> entre 15 e 18; <input type="checkbox"/> maior que 18
Autoavaliação	Pelo teu desempenho neste curso, a nota que esperas ter é: <input type="checkbox"/> Menor que 5; <input type="checkbox"/> entre 6 e 10; <input type="checkbox"/> entre 11 e 14; <input type="checkbox"/> entre 15 e 18; <input type="checkbox"/> maior que 18

Tens alguma sugestão de melhoria? (opcional)

Apêndice V

Grelha de Avaliação do Teste de Pensamento Computacional

Curso 481040, UFCD 0817, 11º ano TPI, Escola [REDACTED]

Nome: _____; N°: _____

Turno: _____

Consentimento Informado? _____

15/03, _____ certas em 28, Nota em 20v _____

01/04, _____ certas em 28, Nota em 20v _____

Os números das questões e a posição da alternativa correta não foram os mesmos em 15/3 e 01/04

Questão n°, Conceito	15/03 N° das questões erradas	01/04 N° das questões erradas (N° e alinea reconvertidos à 15/3)	Comparação
1 a 4, Sequências			
5 a 8, Loop <i>repeat times</i>			
9 a 12, Loop <i>repeat until</i>			
13 a 16, Condicional <i>if</i>			
17 a 20, Condicional <i>if/else</i>			
21 a 24, Condicional <i>while</i>			
25 a 28, Funções			

Comentário geral

Apêndice VI

Excertos das 14 Entrevistas Individuais sobre o Teste de Pensamento Computacional, CTt, em 19/4/2022, após ter apresentado a cada um o seu resultado e o resultado global da turma Curso 481040, UFCD 0817, 11º ano TPI, Escola [REDACTED]

Aluno - A

A questão 15 foi removida de outros testes por ser considerada confusa. Qual a tua opinião?

"...esta questão [15] era um pouco difícil, eu pensei que ele [Pac-man] ia para cima e não acontecia nada. Fiquei na dúvida entre as duas escolhas."

Achaste o teste muito comprido, ou muito difícil?

"Não foi muito difícil, foi diferente do que fazemos na escola. Não foi muito comprido, não estava muito cansado, mas já começava a baralhar as idéias..."

Achas que o teste devia começar logo com as questões mais difíceis?

"...acho que foi preciso essa progressão do fácil para o difícil."

Qual a tua opinião sobre questões com linhas [20] e blocos de código [21]?

"É melhor ter linhas separadas [20] do que vários blocos [21] para analisar na mesma pergunta."

Aluno - B

Achaste o teste muito comprido, ou muito difícil?

"O teste é compreensível, mas tem um padrão de dificuldade, é preciso ter atenção. Não achei muito comprido, os 45 minutos cumpriram bem."

Compreendeste a linguagem visual convertida do Scratch?

"Por acaso, compreendi bem, eu usava o Scratch e depois comecei a construir robôs."

Achas que o teste devia começar logo com as questões mais difíceis?

"É bom ter perguntas fáceis porque vai dando um início que é para a pessoa ir compreendendo as próximas. Para melhorar o teste, acho que podia diminuir o nº de questões fáceis, por exemplo, 3 ou 4 fáceis, e depois umas medianas, e depois subir o nível."

Aluno - C

Aluno não compareceu.

Aluno - D

Qual a tua opinião da questão 15?

"...eu estava indeciso, mas depois fui para o que me parecia que tinha mais sentido. Agora que explicou já percebi que foi falta de atenção..."

Qual a tua opinião da questão 21?

"...While há morango - era uma solução alternativa... (durante a entrevista, o aluno propôs uma alternativa correta que poderia ser introduzida no teste)"

Achaste o teste muito comprido, ou muito difícil?

"Não era comprido, fazia-se rápido, mas tinha-se de pensar muito."

Achas que o teste devia começar logo com as questões mais difíceis?

"Se talvez apanhasse uma questão muito difícil logo no início, talvez prestasse mais atenção ao teste."

Aluno - E

Aluno não compareceu.

Aluno - F

Qual a tua opinião da questão 15? -

"...eu achei que quando encontrasse o triângulo ele [Pac-man] ia entrar numa variável que o podia impedir de seguir..."

Achaste o teste muito comprido, ou muito difícil?

"Não achei o teste comprido... inicialmente tive dificuldade de perceber a movimentação do personagem Pac-Man, mas depois notei que quando ele vira, a frente dele muda..."

Compreendeste a linguagem visual convertida do Scratch? - "A linguagem visual foi fácil de entender. Não achei que foi brusca a passagem para Java. A minha dificuldade foi perceber a movimentação do objeto... Por isso atrasei-me e fui um dos últimos a entregar, tive de voltar ao início do teste e rever as questões, pois, no início, tinha entendido errado, mas, de resto, entendi sim.

Achas que o teste devia começar logo com as questões mais difíceis?

"Eu achava interessante se fosse primeiro o Java, e depois passasse para as setas. As questões do fim são mais difíceis, então o pessoal erra porque não tem tempo para analisar...."

Aluno - G

Em geral, que achaste do teste?

"Achei interessante, gostei de fazer o teste. Faltou atenção. Para melhorar é preciso ler as perguntas com mais atenção."

Aluno - H

Aluno não compareceu.

Aluno - I

Achaste o teste muito difícil, ou que tinha demasiadas perguntas?

"Eu antes não olhava para programação, e agora estou prestando mais atenção, já está na hora. Não é questão de ser muitas perguntas, depende do tipo de perguntas, depende se são fáceis ou difíceis..."

Aluno - J

Compreendeste a linguagem visual convertida do Scratch?

"...é bom para poder testar a lógica, porque ajuda a entender um pouco melhor. Porque são muito básicos, mas é bom para ver o erro e passar a entender o código."

Achaste o teste muito comprido, ou muito difícil?

"...acho que 28 [perguntas] está bom, porque testa bastantes coisas, é uma quantidade boa.

Achas que o teste devia começar logo com as questões mais difíceis?

"Se começasse logo com a difícil, ia-me sentir intimidada. É bom alternar uma fácil e uma difícil para não desanimar."

Aluno - K

Achaste o teste muito difícil?

"...talvez errei porque já estava a ficar cansado quando cheguei nas perguntas do final."

Aluno - L

Qual a tua opinião da questão 15?

" À primeira estava confuso, mas agora já percebi como é que era [questão 15]."

Em geral, que achaste do teste?

"Está bom o teste. Podia talvez ficar mais pequeno e tirar a 15 [questão]."

Aluno - M

Qual a tua opinião da questão 15?

"...baralhei um bocado.

Achaste o teste muito comprido, ou muito difícil?

"Não foi muito comprido, foi normal."

Achas que o teste devia começar logo com as questões mais difíceis?

"Acho que havia de ser logo de início, sem aquecimento."

Aluno - N

Achaste o teste muito exaustivo?

"No primeiro dia não, foi logo no início da aula. A segunda vez foi no final, já estava cansado."

Qual a tua opinião da questão 21? "Não sei, sinceramente, estas dos morangos não estava a perceber."

Achaste o teste muito comprido, ou muito difícil?

"Até se faz bem o teste, não é muito comprido, temos é de ter mais atenção."

Achas que o teste devia começar logo com as questões mais difíceis?

"Se fosse ao contrário, se começasse com as mais difíceis, acho que estaríamos mais atentos."

Aluno - O

Em geral, que achaste do teste?

"Até achei interessante, nunca tinha feito um teste deste. Foi a primeira vez. Não é daqueles testes que se chega a meio e se está farto. Agora que estou no 11º, já estou um bocado habituado."

Qual a tua opinião sobre questões com linhas [20] e blocos de código [21]?

"Não acho mais complicado resolver as questões com [blocos de] código, porque é uma coisa fácil de se entender. Acho mais fácil escolher um código do que escolher uma só linha, por exemplo.

Qual a tua opinião da questão 15?

Na questão 15 não sabia qual escolher, fui mais pela sorte.

Aluno - P

A questão 15 foi removida de outros testes por ser considerada confusa. Qual a tua opinião?

"Agora que vi a resposta, percebi que não era difícil, não vi bem as outras opções."

Achaste o teste muito difícil?

"Achei o teste médio, havia perguntas difíceis e havia perguntas fáceis."

Aluno - Q

Muita gente errou na questão 15. Qual a tua opinião desta questão?

"Ah, sim, foi um caso de precipitação, como viram que a primeira era verdadeira, já ninguém olhou as outras [opções]. Talvez se começasse com a certa e depois as meio-certas..."

Qual a tua opinião sobre questões com linhas [20] e blocos de código [21]?

"Ver dois [blocos de] código já dá trabalho, agora, ver quatro..."

Achaste o teste muito comprido, ou muito difícil?

"...no final, nas últimas, já estávamos atrofiados."

Achas que o teste devia começar logo com as questões mais difíceis?

"Devia-se fazer uma mistura das questões, tipo fácil-média-difícil, fácil-média-difícil, acho que dava para o cérebro descansar, entrar numa rotina."

Apêndice VII

Grelha de Avaliação das Apps IMC e ICQ

Curso 481040, UFCD 0817, 11º ano TPI, Escola XXXXXXXXXX

Nome: _____; N.º: _____

Turno: _____

 **App Índice de Massa Corporal (IMC)** _____ certas, _____ erradas, ver tabela. Nota: _____

Tópicos de correção em ordem decrescente de acertos

Critério IMC	Funcionalidade	Pontuação	Aluno
Entregou	Participação	4	
Funciona	Escrita e Conversão: métodos de objeto <i>Widget: setText()</i> , <i>Double.parseDouble()</i> , <i>String.format()</i>	4	
Não estoura	Verificação: métodos de objeto <i>Widget: getText()</i> , <i>toString()</i> , <i>isEmpty()</i> , <i>requestFocus()</i>	4	
<i>MainAct</i> original	Ver se foi escrito pelo aluno ou se é cópia de outro aluno ou do Professor	4	
IMC original	Ver se foi escrito pelo aluno ou se é cópia de outro aluno ou do Professor	4	

 **App Índice Cintura-Quadril (ICQ)** _____ certas, _____ erradas, ver tabela. Nota: _____

Tópicos de correção em ordem decrescente de acertos

Critério	Requisitos de Método() Java / <Atributo> XML	Pontuação	Aluno
Caixas de texto: Hints	<EditText>: atributo <i>hint=""</i>	1	
Botão Limpar: funciona	<Button>: atributo <i>onClick=""limpar/calcular"</i> Limpar(): objetos <i>TextView</i> e <i>EditText</i> utilizam o método <i>setText</i>	1	
Caixas de texto e rótulos: corretos	Tela contém todos os <i>TextViews</i> , <i>EditTexts</i> e <i>Buttons</i> necessários organizados com <i>ConstraintLayouts</i>	1	
calculaICQ: não estoura com caixas vazias	Calcular(): objetos <i>EditText</i> utilizam o método <i>getText</i> , <i>toString</i> , <i>isEmpty</i> para proteger contra exceções nulas	2	
calculaICQ: requestFocus	Proteções contra exceções nulas: utilizam o método <i>requestFocus</i> , o qual direciona o teclado virtual para o <i>EditText</i> indicado	1	
calculaICQ: funciona com sexo	Calcular(): instancia objeto ICQ com pelo menos os valores de Quadril, Cintura e Sexo, chama o método de cálculo, e escreve o resultado no rótulo ICQ	5	
classicalICQ: funciona corretamente	Calcular(): instancia objeto ICQ com pelo menos os valores de Quadril, Cintura e Sexo, chama o método de classificação, e escreve o resultado no rótulo Mensagem	2	
CC, Cintura e Quadril: teclado numérico	<EditText>: atributo <i>inputType=""number"</i>	1	
Orientação horizontal: rolável	<ScrollView>: <i>ViewGroup</i> que contém um segundo <i>ConstraintLayout</i> . Necessita de um ficheiro <i>activity_main.xml(land)</i>	1	
classicalICQ: aceita sexo com minúsculas "m/f"	Classifica(): condicional <i>if / switch-case</i> com operador "ou" (<code> </code>)	2	
calculaICQ: funciona sem sexo	Calcular(): instancia objeto ICQ com pelo menos os valores de Quadril e Cintura, e sem o valor do Sexo, chama o método de cálculo, e escreve o resultado no rótulo ICQ	1	
classicalICQ: calcula Cintura em excesso	Classifica(): instancia objeto ICQ, chama o método de classificação, e retorna um número calculado da diferença entre a cintura atual e a cintura ideal, dependendo do valor da <i>String</i> do Sexo	2	

Apêndice VIII

Excertos das 16 Entrevistas Individuais sobre as Apps IMC e ICQ, em 26/4/2022, após ter apresentado a cada um o seu resultado e o resultado global da turma.

Curso 481040, UFCD 0817, 11º ano TPI, Escola [REDACTED]

Aluno - A

O que é mais difícil para ti no Android Studio?

"Acho que é um bocado o código [Java], mas difícil também é meter o Design. No Visual Studio era mais fácil. Não era preciso usar as setinhas [Constraints] para deixar a tela equilibrada."

Tens mais dificuldade com o XML do que com o Java?

"Sim, acho mais difícil, preferia o IntelliJ."

Achas que o ritmo do curso foi bom?

"Sim."

Aluno - B

O que é mais difícil para ti no Android Studio?

"É mais a diferença dos comandos dessa plataforma para os outros que usávamos [IntelliJ]. Começámos o ano com o IntelliJ e dois meses depois partimos para o Android Studio... Devíamos ter começado com o A.S."

Tens mais dificuldade com o Java ou com o XML?

"Java. Com o XML tive dificuldade no começo, mas depois comecei a entender e já não tive problemas."

Achas que as aulas estavam a ir num bom ritmo, ou estavam a ir muito depressa?

"Acho que estava a ir um bocado depressa mas depois todos se adaptaram."

Aluno - C

Aluno não compareceu.

Aluno - D

O que é mais difícil para ti no Android Studio?

"Não sei... o código é mais difícil. As partes principais eu já aprendi direito. Onclick, já sei fazer. findViewById, já sei fazer. Se for buscar à net coisas que não sei, já sei fazer isso. Há partes que é preciso andar à procura, e isso já não sei fazer muito bem."

Tens mais dificuldade com o Java ou com o XML?

"Java. As telas demoram tempo, mas fazem-se. O Java tem-se que puxar mais pela cabeça."

Que achaste do Android Studio?

"É interessante. Dá gosto aprender."

Aluno - E

O que achas do ritmo das aulas Android?

"Acho que está bom."

O que achas do Android Studio?

"Acho que é acessível."

Tens mais dificuldade com o Java ou com o XML?

"Com o XML, acho eu. Às vezes a organizar as coisas, é difícil, como hei-de dizer... XML para mim é novo, parece um dicionário de atributos. Mesmo com a paleta e com o designer, parece matéria excessiva. Mas eu acho acessível. É ir fazendo e aprendendo os novos termos."

Aluno - F

Que achaste do Android Studio?

"Achei mais fácil do que Visual Studio, mas o meu problema foi que eu tive uma grande quantidade de faltas, por isso errei muitos destes exercícios. Também, quando fui fazer o trabalho estava com pouco tempo. O IMC e o ICQ eu já tinha por causa do Visual Studio. Por isso tentei converter para o Android Studio. Aí até facilitou; o problema mesmo foram as faltas. Por exemplo, não sabia que tinha o Scroll(View)."

Tens mais dificuldade com o Java ou com o XML?

"A tela [XML] é fácil, é intuitivo aprender e colocar. Em dois minutos eu consigo entender. O código [Java] acho que é mais difícil de entender."

Achas que as aulas estavam a ir num bom ritmo, ou estavam a ir muito depressa?

"Sim, estavam [num bom ritmo]. Eu é que tive de me acelerar porque faltei demais."

Aluno - G

O que achaste do processo de fazer Apps?

"Gostei, foi interessante fazê-lo. Algumas partes são mais complexas que outras, mas no geral, acho que é uma coisa que se faz bem."

O que é mais difícil para ti no Android Studio?

"Programar as partes de cálculo no IMC e ICQ. Tive mais dificuldade nessa parte."

Aluno - H

Que achaste do Android Studio?

"Eu não gosto muito de aplicações Android, eu gosto mais de fazer as aplicações no computador."

Tens mais dificuldade com o Java ou com o XML?

"A parte do *layout* [XML] é o que eu mais me atrapalho. Confunde um bocado, às vezes. É diferente dos outros programas que a gente está habituado. O código não é difícil. A lógica é a mesma, só muda as palavras."

Achas que as aulas estavam a ir num bom ritmo, ou estavam a ir muito depressa?

"Não, foi um ritmo acessível."

Aluno - I

Achas que as aulas estavam a ir num bom ritmo, ou estavam a ir muito depressa?

"Acho que não, tava até que num ritmo bom."

Tens mais dificuldade com o Java ou com o XML?

"Eu diria que no XML."

Aluno - J

Achas que as aulas estavam a ir num bom ritmo, ou estavam a ir muito depressa?

"Não, tava bom, eu é que não presto atenção na aula mesmo..."

Aluno - K

Tens mais dificuldade com o Java ou com o XML?

"Não sei..."

O que é mais difícil para ti no Android Studio?

"Programar é o mais difícil. Fazer o *layout* é mais fácil"

Que achaste do Android Studio?

"É bom. Não é muito difícil, mas também não é muito fácil."

Aluno - L

O que achaste do trabalho, de converter IMC e ICQ para uma aplicação Android?

"Foi bom para aprender um bocado."

Gostas de programação em Android?

"Sim, não é mau."

O que é mais difícil para ti no Android Studio?

"É o código. A parte de programar as aplicações."

Aluno - M

O que é mais difícil para ti no Android Studio?

"O Java."

Achas a interface do Android Studio confusa?

"Não, dá para entender mais ou menos. Só que não sou muito bom em inglês."

Aluno - N

Achaste o Android Studio difícil?

"Não acho muito difícil, mas ao mesmo tempo é um bocado complexo. São muitas coisas."

O que é mais difícil para ti no Android Studio?

"O código Java."

Gostaste do Android Studio?

"Sim, gostei. Até mais que o Visual Studio. Gostei da parte do *layout*- do Design."

Aluno - O

Que achas do Android Studio?

"Não estou a desgostar, mas baralho-me um pouco. Não estou muito à vontade para trabalhar."

Tens mais dificuldade com o Java ou com o XML?

"Com o Java. Com as telas também baralho-me um pouco, mas com tempo chego lá."

Sentes que estás a aprender?

"Sim. No início custou um pouco, mas já estou a aprender."

Aluno - P

Tens mais dificuldade com o Java ou com o XML?

"O código [Java] é mais difícil."

Achas que as aulas estavam a ir num bom ritmo?

"Acho que sim. Mais ou menos."

Aluno - Q

O que é mais difícil para ti no Android Studio?

"Acho que é o código [Java]..."

Achas que estás a conseguir acompanhar?

"Mais ou menos."

Apêndice IX

**Autoavaliação do Aluno sobre o seu desempenho
nas aulas lecionadas pelo Prof. Estagiário Maurício Morais
no período de 15/03 a 01/04/2022
Curso 481040, UFCD 0817, 11º ano TPI, Escola [REDACTED]**

Nome:

Nº.....

Indique se as afirmações são verdadeiras V, ou falsas, F, de acordo com o seu perfil.

- 1) Compreendi o que era esperado fazer nos trabalhos IMC e ICQ
- 2) Quando tive dúvidas perguntei ao professor.
- 3) Quando tive dúvidas perguntei aos colegas.
- 4) Fui capaz de resolver os trabalhos propostos sem ajuda do professor e dos colegas.
- 5) Precisei de apoio na construção das telas.
- 6) Precisei de apoio no código Java.
- 7) O ambiente Android Studio é complicado.
- 8) Faltam-me bases de Java para bem compreender o Android.
- 9) Um trabalho consegui terminar, mas o outro não.
- 10) Não consegui terminar os trabalhos IMC e ICQ na aula.
- 11) Tentei fazer os trabalhos IMC e ICQ em casa.
- 12) Interesse-me pela matéria e tenciono continuar a programar apps para Android.
- 13) Interesse-me pela matéria, mas não tenciono continuar a programar apps para Android.
- 14) Não gosto de programação Android.

Apêndice X

Código-Fonte App IMC

IMC: activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

<TextView
android:id="@+id/header_title"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:fontFamily="sans-serif"
android:text="@string/main_header_title"
android:textSize="24sp"
android:textStyle="bold"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"/>

<TextView
android:id="@+id/rtl_peso"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="32dp"
android:text="@string/main_rtl_peso"
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintBaseline_toBaselineOf="@+id/edtxt_peso"
app:layout_constraintEnd_toStartOf="@+id/edtxt_peso"
app:layout_constraintHorizontal_bias="0.5"
app:layout_constraintHorizontal_chainStyle="packed"
app:layout_constraintStart_toStartOf="parent" />

<EditText
android:id="@+id/edtxt_peso"
android:layout_width="wrap_content"
android:layout_height="50dp"
android:layout_marginTop="8dp"
android:ems="8"
android:hint="@string/main_hint_edtxt_peso"
android:inputType="numberDecimal"
android:digits="0123456789."
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/header_title" />

<TextView
android:id="@+id/rtl_altura"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="32dp"
android:text="@string/main_rtl_altura"
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintBaseline_toBaselineOf="@+id/edtxt_altura"
app:layout_constraintEnd_toStartOf="@+id/edtxt_altura"
app:layout_constraintHorizontal_bias="0.5"
app:layout_constraintHorizontal_chainStyle="packed"
app:layout_constraintStart_toStartOf="parent" />

<EditText
android:id="@+id/edtxt_altura"
android:layout_width="wrap_content"
android:layout_height="50dp"
android:layout_marginTop="8dp"
android:ems="8"
android:hint="@string/main_hint_edtxt_altura"
android:inputType="numberDecimal"
android:digits="0123456789."
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/edtxt_peso" />

<TextView
android:id="@+id/rtl_imc"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="32dp"
android:text="@string/main_rtl_imc"
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintBaseline_toBaselineOf="@+id/txt_imc"
app:layout_constraintEnd_toStartOf="@+id/txt_imc"
app:layout_constraintHorizontal_bias="0.5"
app:layout_constraintHorizontal_chainStyle="packed"
app:layout_constraintStart_toStartOf="parent" />

<TextView
android:id="@+id/txt_imc"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="12dp"
android:ems="8"
android:text="@string/main_txt_imc"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/edtxt_altura"/>

<TextView
android:id="@+id/txt_msg"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="12dp"
android:ems="8"
android:text=""
android:textSize="20sp"
android:textStyle="bold"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txt_imc" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

```

<Button
android:id="@+id/btn_calcular"
android:onClick="calcular"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:ems="12"
android:text="@string/main_btn_calcular"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf=
"parent"
app:layout_constraintTop_toBottomOf=
"@+id/txt_msg" />

```

```

<Button
android:id="@+id/btn_limpar"
android:onClick="limpar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="12dp"
android:ems="12"
android:text="@string/main_btn_limpar"
android:textSize="20sp"
app:layout_constraintEnd_toEndOf=
"parent"
app:layout_constraintStart_toStartOf=
"parent"
app:layout_constraintTop_toBottomOf=
"@+id/btn_calcular" />

```

```

</androidx.constraintlayout.widget.ConstraintLa
yout>

```

IMC: Imc.java

```
package com.example.imc;
```

```
public class Imc{
```

```
    public double peso, altura, imc;
    public String strClassifica;
```

```
    // getters
```

```
    double getPeso(){return peso;}
    double getAltura(){return altura;}
    double getImc(){return imc;}

```

```
    // setters
```

```
    void setPeso(double peso){this.peso=peso;}
    void setAltura(double altura){ this.altura=altura;}

```

```
    void setImc(){imc = peso/(altura*altura);}
```

```
    void setClassifica(){
```

```
        setImc();
        if (imc < 18)                { strClassifica = "Peso baixo";}
        if (imc >= 18 && imc < 24.9){ strClassifica = "Peso normal";}
        if (imc >= 25 && imc < 29.9){ strClassifica = "Pré-obesidade";}
        if (imc >= 30 && imc < 34.9){ strClassifica = "Obesidade grau I";}
        if (imc >= 35 && imc < 39.9){ strClassifica = "Obesidade grau II";}
        if (imc >= 40)                { strClassifica = "Obesidade grau III";}
    }

```

```
    String getClassifica(){
```

```
        setClassifica();
        return strClassifica;
    }

```

```
}
```

IMC: MainActivity.java

```
package com.example.imc;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    public TextView txtIMC, txtMsg;
    public EditText edtxtPeso, edtxtAltura;

    Imc imc = new Imc();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtxtPeso = findViewById(R.id.edtxt_peso);
        edtxtAltura = findViewById(R.id.edtxt_altura);
        txtIMC = findViewById(R.id.txt_imc);
        txtMsg = findViewById(R.id.txt_msg);
    }

    public void calcular(View view) {
        Log.i("btn_calcular", "onClick funciona!");
        // verificar se os edtxts estão vazios
        if (edtxtPeso.getText().toString().isEmpty()) {
            txtMsg.setText("Peso?");
            edtxtPeso.requestFocus();
            return;
        }
        if (edtxtAltura.getText().toString().isEmpty()) {
            txtMsg.setText("Altura?");
            edtxtAltura.requestFocus();
            return;
        }
        // converter inputs para Double e guardar variaveis no objeto imc
        imc.setPeso(Double.parseDouble(edtxtPeso.getText().toString()));
        imc.setAltura(Double.parseDouble(edtxtAltura.getText().toString()));
        // chamar metodo de calculo
        imc.setImc();
        // converter resultado Double para String
        // String strImc = String.valueOf(imc.getImc()); // sem formatação
        String strImc = String.format("%.2f", imc.getImc()); // 2 casas decimais
        // escrever resultado no textview Imc
        txtIMC.setText(strImc);
        // escrever classificação no textview msg
        txtMsg.setText(imc.getClassifica());
    }

    public void limpar(View view) {
        Log.i("btn_limpar", "onClick funciona!");
        edtxtPeso.setText("");
        edtxtAltura.setText("");
        txtIMC.setText("");
        txtMsg.setText("");
        edtxtPeso.requestFocus();
    }
}
```



```

android:textStyle="bold"
app:layout_constraintBaseline_toBaselineOf=
"@+id/edtxt_cintura"
app:layout_constraintEnd_toStartOf=
"@+id/edtxt_cintura"
app:layout_constraintHorizontal_chainStyle=
"packed" />

```

```

<EditText
android:id="@+id/edtxt_cintura"
android:layout_width="200dp"
android:layout_height="48dp"
android:layout_margin="16dp"
android:ems="10"
android:hint="@string/hint_cintura"
android:inputType="number"
android:digits="0123456789"
android:textSize="16sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf=
"@+id/edtxt_cc" />

```

```

<TextView
android:id="@+id/rtl_quadril"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="16dp"
android:text="@string/rtl_quadril"
android:textSize="16sp"
android:textStyle="bold"
app:layout_constraintBaseline_toBaselineOf=
"@+id/edtxt_quadril"
app:layout_constraintEnd_toStartOf=
"@+id/edtxt_quadril"
app:layout_constraintHorizontal_chainStyle=
"packed" />

```

```

<EditText
android:id="@+id/edtxt_quadril"
android:layout_width="200dp"
android:layout_height="48dp"
android:layout_margin="16dp"
android:ems="10"
android:hint="@string/hint_quadril"
android:inputType="number"
android:digits="0123456789"
android:textSize="16sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf=
"parent"
app:layout_constraintTop_toBottomOf=
"@+id/edtxt_cintura" />

```

```

<TextView
android:id="@+id/rtl_icq"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_margin="16dp"
android:text="@string/rtl_icq"
android:textSize="16sp"
android:textStyle="bold"
app:layout_constraintBaseline_toBaselineOf=
"@+id/txt_icq"
app:layout_constraintEnd_toStartOf=
"@+id/txt_icq"
app:layout_constraintHorizontal_chainStyle=
"packed" />

```

```

<TextView
android:id="@+id/txt_icq"
android:layout_width="200dp"
android:layout_height="wrap_content"
android:layout_margin="16dp"
android:hint="@string/hint_icq"
android:textSize="16sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf=
"@+id/edtxt_quadril" />

```

```

<TextView
android:id="@+id/txt_mensagem"
android:layout_width="200dp"
android:layout_height="wrap_content"
android:layout_margin="16dp"
android:text="@string/rtl_mensagem"
android:textSize="16sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf=
"@+id/txt_icq"
app:layout_constraintBottom_toBottomOf=
"parent" />
</androidx.constraintlayout.widget.
ConstraintLayout>
</ScrollView>

```

```

<androidx.constraintlayout.widget.
ConstraintLayout
android:id="@+id/Constraintlayout_3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textAlignment="gravity"
app:layout_constraintBottom_toBottomOf=
"@id/ConstraintLayout_1"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent">
<Button
android:id="@+id/btn_calcular"
android:onClick="calcular"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="12dp"
android:ems="12"
android:text="@string/btn_calcular"
android:textSize="16sp"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintBottom_toTopOf=
"@id/btn_limpar" />
<Button
android:id="@+id/btn_limpar"
android:onClick="limpar"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="10dp"
android:layout_marginBottom="16dp"
android:ems="12"
android:text="@string/btn_limpar"
android:textSize="16sp"
app:layout_constraintTop_toBottomOf=
"@id/btn_calcular"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
/>
</androidx.constraintlayout.widget.
ConstraintLayout>
</androidx.constraintlayout.widget.
ConstraintLayout>

```

ICQ: Icq.java

```
package com.example.icq;
public class Icq {

    public int cc;
    public double cintura, quadril, icq;
    public String nome, sexo, classifica = "";

    // construtor 2 parametros
    public Icq(double cintura, double quadril){
        this.cintura = cintura; this.quadril = quadril;
        setIcq();
    }
    // construtor 5 parametros
    public Icq(String nome, String sexo, int cc, double cintura, double quadril){
        this.nome = nome; this.sexo = sexo; // strings
        this.cc = cc; // int
        this.cintura = cintura; this.quadril = quadril; // doubles
        setIcq();
        setClassifica();
    }
    void setIcq(){
        this.icq = (double) cintura / (double) quadril; // (casting)
    }

    void setClassifica() {
        switch (sexo) {
            case "": classifica = "Sexo?";
                break;
            case "M": case "m":
                if (icq > 1) { classifica = "Acima do recomendado. Masculino <= 1.00"; }
                else { classifica = "Dentro do recomendado. Masculino <= 1.00"; }
                break;
            case "F": case "f":
                if (icq > 0.8){ classifica = "Acima do recomendado. Feminino <= 0.80"; }
                else { classifica = "Dentro do recomendado. Feminino <= 0.80"; }
                break;
            default: // digitar outro sexo para listar valores das variaveis
                classifica = "nome= " + nome + " cc= " + cc +
                    " cint= " + cintura + " quad= " + quadril;
                break;
        }
    }
}
```

ICQ: MainActivity.java

```
package com.example.icq;

import androidx.appcompat.app.AppCompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.widget.EditText;
import android.os.Bundle;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    public TextView txtICQ, txtMensagem;
    public EditText edtxtNome, edtxtSexo, edtxtCC, edtxtCintura, edtxtQuadril;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtxtNome = findViewById(R.id.edtxt_nome);
        edtxtSexo = findViewById(R.id.edtxt_sexo);
        edtxtCC = findViewById(R.id.edtxt_cc);
        edtxtCintura = findViewById(R.id.edtxt_cintura);
        edtxtQuadril = findViewById(R.id.edtxt_quadril);
    }
}
```

```

txtICQ = findViewById(R.id.txt_icq);
txtMensagem = findViewById(R.id.txt_mensagem);
// CC é input type password. usar este metodo para não ocultar numeros
edttxCC.setTransformationMethod(null);
}

public void calcular(View view) {

    if(edtxtNome.getText().toString().isEmpty()){
        txtMensagem.setText("Nome?");
        edtxtNome.requestFocus();
        return;
    }

    if(edtxtSexo.getText().toString().isEmpty()){
        txtMensagem.setText("Sexo?");
        edtxtSexo.requestFocus();
        return;
    }

    if(edtxtCC.getText().toString().isEmpty()){
        txtMensagem.setText("CC?");
        edtxtCC.requestFocus();
        return;
    }
    if(edtxtCintura.getText().toString().isEmpty()){
        txtMensagem.setText("Cintura?");
        edtxtCintura.requestFocus(); return;
    }
    if(edtxtQuadril.getText().toString().isEmpty()){
        txtMensagem.setText("Quadril?");
        edtxtQuadril.requestFocus();
        return;
    }

    // 2 parametros mínimos para chamar setIcq sem saber o sexo
    Icq icq2 = new Icq(
        Double.parseDouble(edtxtCintura.getText().toString()),
        Double.parseDouble(edtxtQuadril.getText().toString()));

    // 5 parametros para chamar setIcq e classificar segundo o sexo
    Icq icq5 = new Icq(
        edtxtNome.getText().toString(),
        edtxtSexo.getText().toString(),
        Integer.parseInt(edtxtCC.getText().toString()),
        icq2.cintura,
        icq2.quadril);

    String strIcq = String.format(Locale.US, "%.2f", icq2.icq);
    txtICQ.setText(strIcq);
    txtMensagem.setText(icq5.classifica);
}

public void limpar(View view) {
    edtxtCC.setText("");
    edtxtNome.setText("");
    edtxtSexo.setText("");
    edtxtCintura.setText("");
    edtxtQuadril.setText("");
    txtICQ.setText("");
    txtMensagem.setText("");
    edtxtNome.requestFocus();
}
}

```

Anexos

Anexo A

Definições Adotadas neste Relatório

- O que é Informática, etimologia e significado.

Em 1957, Karl Steinbuch publicou o artigo “Informatik: Automatische Informationsverarbeitung”, criando, em alemão, o acrónimo informatik, resultado de informação automática. Em 1962, Philippe Dreyfus usou a expressão “informatique”, com intenção de indicar o processamento automático da informação por meio de equipamentos electrónicos e sistemas computacionais. Outras línguas românicas, como português, espanhol e italiano, incorporaram o acrónimo como “informática” (Neto, 2020 – p.24).

“Apesar da sua intensa utilização, ainda não há uma definição consensual de informática na comunidade científica. A informática, como terminologia aceite e utilizada na Europa continental, é uma disciplina emergente que abrange uma ampla gama de áreas científicas e implicacionais. Uma das noções centrais em Informática é a informação; a computação é a nova possibilidade de trabalhar com formas representacionais de informação. Nos Estados Unidos da América, a Ciência da Computação (Computer Science) e a Tecnologia da Informação (Information Technology), juntas, abrangem uma área semelhante” (Benczúr & Molnár, 2018).

A Escola de Informática da Universidade de Edimburgo apresenta a seguinte definição no seu sítio da internet:

“A informática é o estudo da estrutura, comportamento e interações de sistemas computacionais naturais e de engenharia. O foco central da Informática é a transformação da informação - seja por computação ou comunicação, seja por organismos ou artefactos. Compreender fenómenos informacionais - como computação, cognição e comunicação - possibilita avanços tecnológicos” (School of Informatics, 2016).

-- O que é o Ensino de Informática

O documento de trabalho que acompanha o Plano de Ação de Educação Digital 2021-2027, da Comissão Europeia, fornece a seguinte indicação sobre o ensino de Informática:

“O ensino de computação e informática, também conhecido como ciência da computação em muitos países, é uma disciplina científica distinta, caracterizada por seus próprios conceitos, métodos, corpo de conhecimento e questões em aberto. Abrange os fundamentos de estruturas computacionais, processos, artefatos e sistemas, e seus projetos de software, suas aplicações e seu impacto na sociedade” (EC-SWD, 2020-p.95).

- O que é Pensamento Computacional

“O pensamento computacional é o processo de pensamento envolvido na formulação de um problema e na expressão de sua(s) solução(ões) de tal forma que um computador – humano ou máquina – possa efetivamente realizar” (Wing, 2017).

A figura 96 ilustra seis pilares básicos do pensamento computacional, com utilização de computador-máquina.

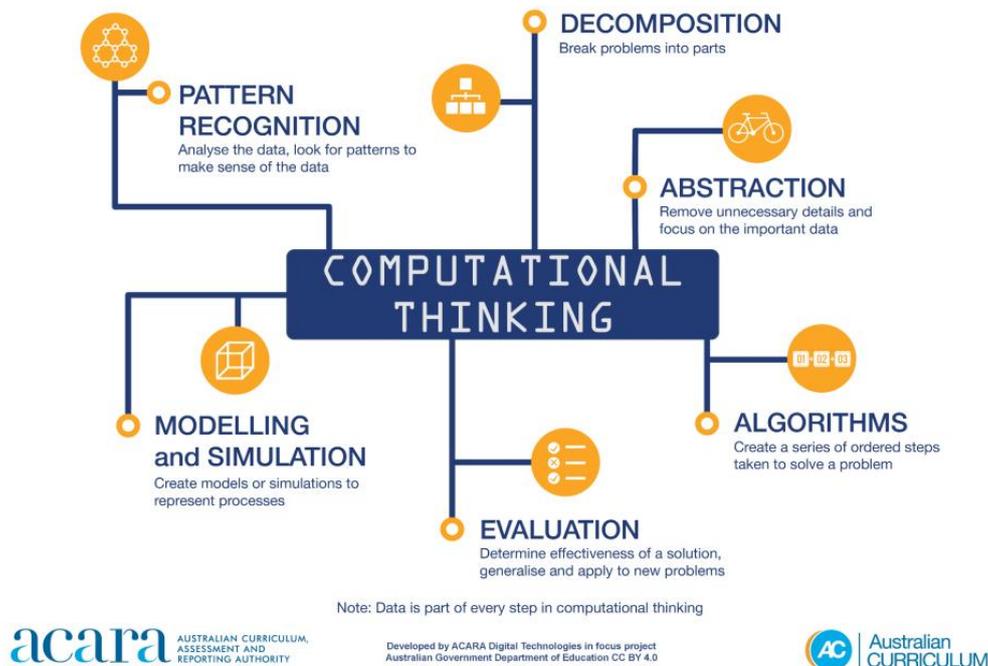


Figura 96: Seis Conceitos Básicos do Pensamento Computacional (ACARA, 2020)

Para fins de expansão do pensamento computacional a outras disciplinas, podem remover-se os conceitos diretamente relacionados com a interação com o computador, e destacar os quatro conceitos fundamentais do computador-humano: decomposição, reconhecimento de padrões, abstração e algoritmia (Figura 97).

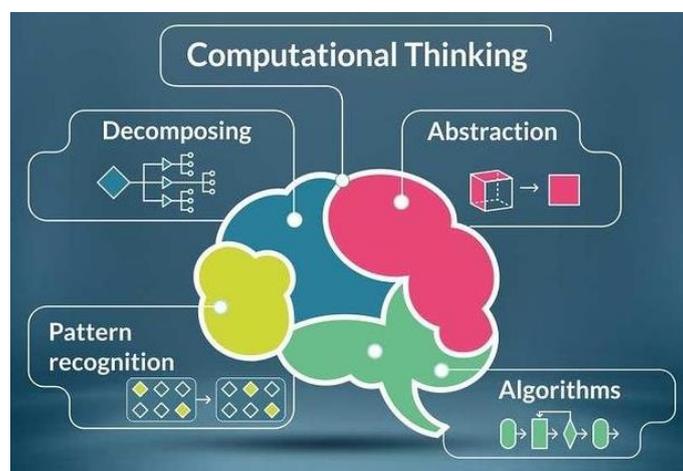


Figura 97: Quatro Conceitos Básicos do Pensamento Computacional (Kaoneill, 2020)

Anexo B

Autorização do Autor do Teste Original de Pensamento Computacional, Prof. Dr. Marcos Román González. Disponível em <https://redined.educacion.gob.es/xmlui/handle/11162/157859>

— Mensagem encaminhada de MARCOS ROMAN GONZALEZ <mroman@edu.uned.es> —
Data: Mon, 6 Dec 2021 13:01:15 +0000
De: MARCOS ROMAN GONZALEZ <mroman@edu.uned.es>
Assunto: RE: Solicitud de autorización para usar la prueba
Para: Mauricio Braga Morais <pg40625@alunos.uminho.pt>
Cc: "mb.morais" <mb.morais@sapo.pt>

Estimado Mauricio,

Tiene usted mi autorización.

Enhorabuena por la adaptación y mucho ánimo con la investigación.

Un abrazo fraternal desde España.

Marcos Román González

Coordinador de Comunicación, Virtualización y Tecnologías

Facultad de Educación, Departamento MIDE I
C/ Juan del Rosal, 14, Despacho 2.18. 28040 Madrid
T. +34 91 398 90 37

Anexo C

Autorização do Autor da Tradução para o Português do Teste Original de Pensamento Computacional, Prof. Dr. António Soares Marques. Disponível em <https://ubibliorum.ubi.pt/handle/10400.6/11156>

De: António Marques <prof.informatica.marques@gmail.com>

Enviado: 9 de dezembro de 2021 20:56

Para: Mauricio Braga Morais <pg40625@alunos.uminho.pt>

Assunto: Re: Pedido de autorização para usar a sua tradução

Boa noite,

Antes de mais, desejo uma ótima investigação.

Relativamente à autorização solicitada, da minha parte, tem autorização para poder utilizar a tradução/adaptação que fiz do questionário sobre pensamento computacional de Marcos Román-González.

Com os melhores cumprimentos,

.:António Marques

Anexo D

Autorização do Orientador Supervisor, Prof. Dr. Maria Altina Silva Ramos.



Universidade do Minho
Instituto de Educação

PARECER

Para os devidos efeitos, declaro que: a) oriento o Estágio integrado do Mestrado em Ensino de Informática do estudante Maurício Braga Morais; b) concordo com a investigação que propõe tanto quanto em relação ao conteúdo quanto em relação à metodologia utilizada para atingir os objetivos propostos. Declaro ainda que os instrumentos a aplicar na investigação foram discutidos em conjunto e por mim aprovados.

Braga e Instituto de Educação, 06 de janeiro de 2022

A Orientadora



Maria Altina Silva Ramos
(Prof. Auxiliar)

Anexo E

Autorização da Direção-Geral da Educação (DGE).

De: mime.noreply@min-educ.pt <mime.noreply@min-educ.pt>

Enviado: 23 de fevereiro de 2022 08:25

Para: Mauricio Braga Morais <pg40625@alunos.uminho.pt>; Mauricio Braga Morais <pg40625@alunos.uminho.pt>

Assunto: Monotorização de Inquéritos em Meio Escolar: Inquérito nº 0807900001

Exmo(a)s. Sr(a)s.

O pedido de autorização do inquérito n.º 0807900001, com a designação *A utilização da ferramenta “Android Studio” contribui para o desenvolvimento do pensamento computacional de estudantes do curso profissional de informática? Estudo de caso numa turma do 11º ano.*, registado em 06-01-2022, foi aprovado.

Avaliação do inquérito:

Exmo. Senhor Maurício Braga Morais

Venho por este meio informar que o pedido de realização de inquérito em meio escolar é autorizado uma vez que, submetido a análise, cumpre os requisitos, devendo atender-se às observações aduzidas.

Com os melhores cumprimentos

José Vitor Pedroso

Diretor-Geral

DGE

Observações:

a) A aplicação dos instrumentos de inquirição (Teste de Pensamento Computacional, questionário de avaliação da qualidade do professor estagiário, autoavaliação do estudante e entrevista do tipo focus group) fica sujeita à autorização da Direção da Escola Secundária [REDACTED], em Braga, a contactar para a realização do estudo. Merece especial atenção o modo, o momento e condições de aplicação dos instrumentos de recolha de dados em meio escolar, porque onerosos, devendo fazer-se em estreita articulação com a Direção da Escola e e com os encarregados de educação ou quem tutele os menores.

b) Deve considerar-se o disposto legal em matéria de garantia de anonimato dos sujeitos, confidencialidade, proteção e segurança dos dados pessoais a recolher e tratar no presente estudo, devendo prever-se medidas adequadas e específicas para a defesa dos direitos fundamentais e dos interesses dos titulares dos dados. Deste modo, procura-se garantir o tratamento lícito dos mesmos e a conformidade com os termos procedimentais indicados e legislação em vigor. Considerados os documentos que foram anexados e para efeitos da proteção de dados pessoais a recolher junto dos inquiridos resultam obrigações que o responsável se propõe cumprir. Destas deve dar conhecimento a todos os inquiridos e a quem intervenha na recolha e tratamento de dados. É obrigatório recolher previamente as declarações de consentimento inequívoco, informado e esclarecido, junto dos inquiridos, titulares dos dados, no caso de menores, junto dos seus representantes legais sobre a sua anuência no preenchimento dos instrumentos de inquirição. As autorizações assinadas devem ficar em poder da Escola Secundária [REDACTED]. Recomenda-se que para efeitos de redação do consentimento supracitado, proteção de dados e cumprimento do disposto legal nesta matéria, o/a Encarregado/a de Proteção de Dados da entidade responsável pela investigação académica possa apoiar todo o processo.

c) Ao ser utilizada uma plataforma tecnológica para registo de dados junto dos inquiridos, (não devem ser recolhidos emails pessoais de alunos, salvo se houver consentimento inequívoco e informado dos seus representantes legais)

deve-se acautelar que as questões colocadas pelos instrumentos de inquirição/registo devem ser respondidas apenas pelo destinatário pretendido

(proceder-se à inquirição através de um único acesso - link da plataforma a utilizar - utilizando-se um ou mais computadores a disponibilizar para o efeito na escola, ou outra forma considerada adequada àquele propósito).

Em caso de ser instrumento de livre acesso, não é da competência da Direção-Geral da Educação (DGE) autorizar a sua aplicação, uma vez que qualquer pessoa pode responder.

Pode consultar na Internet toda a informação referente a este pedido no endereço <http://mime.gepe.min-edu.pt>. Para tal terá de se autenticar fornecendo os dados de acesso da entidade.

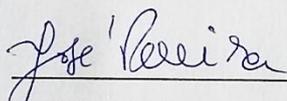
Anexo F

Autorização do Orientador Cooperante, Professor José Manuel Almeida Pereira.

DECLARAÇÃO

Serve a presente declaração para autorizar o estudante de Mestrado de Ensino de Informática da Universidade do Minho, Maurício Braga Morais, PG 40625, portador do cartão de cidadão número 12651890-4ZY7, com data de validade 01/06/2028, a publicar o meu nome como Professor Cooperante no seu relatório final de conclusão do mesmo.

Braga, 1 de junho de 2022.



José Manuel Almeida Pereira, c.cidadão nº 8544205-4ZX2

Anexo G

Tradução da lição 1.1 do *Codelabs (Hello World)* do Android Studio, em Java, para o Português de Portugal, pelo autor deste relatório. Original disponível em:

<https://developer.android.com/codelabs/android-training-hello-world?index=..%2F..%2Fandroid-training#0>

Aula 1 – Hello World

Introdução

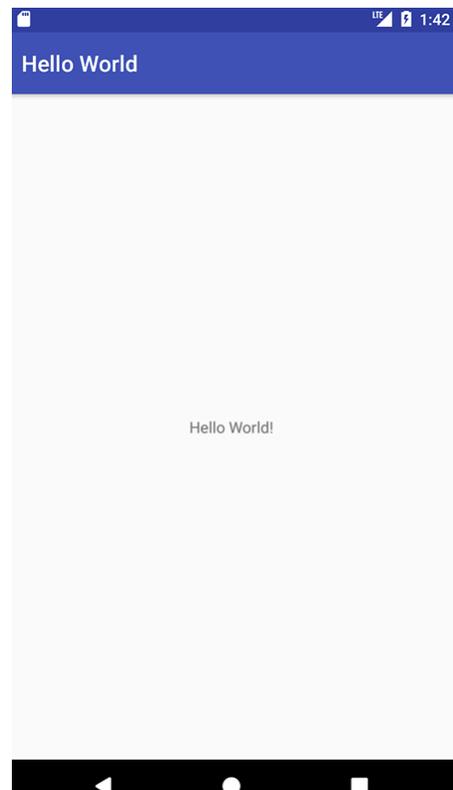
Nesta aula vamos aprender os conceitos básicos do Android Studio - o ambiente de desenvolvimento integrado (IDE) do Android. Vamos aprender a estrutura básica dos projetos Android enquanto criamos uma aplicação *Hello World*, e testá-la num dispositivo físico e num dispositivo virtual.

O que já deve saber:

- O processo de desenvolvimento de software para aplicações usando um IDE.
- Programação orientada a objetos, particularmente voltada para a linguagem de programação Java.

O que vamos fazer:

1. Criar a aplicação Hello World
 - 1.1. Criar um Novo Projeto
 - 1.2. Explorar o painel de Projeto
 - 1.3. Explorar a pasta de scripts Gradle
 - 1.4. Explorar as pastas App e Res
 - 1.5. Explorar a pasta Manifests
2. Criar um dispositivo virtual (Emulador)
3. Testar a aplicação num dispositivo físico
4. Alterar configurações do Gradle
5. Adicionar mensagens de registro (Log)



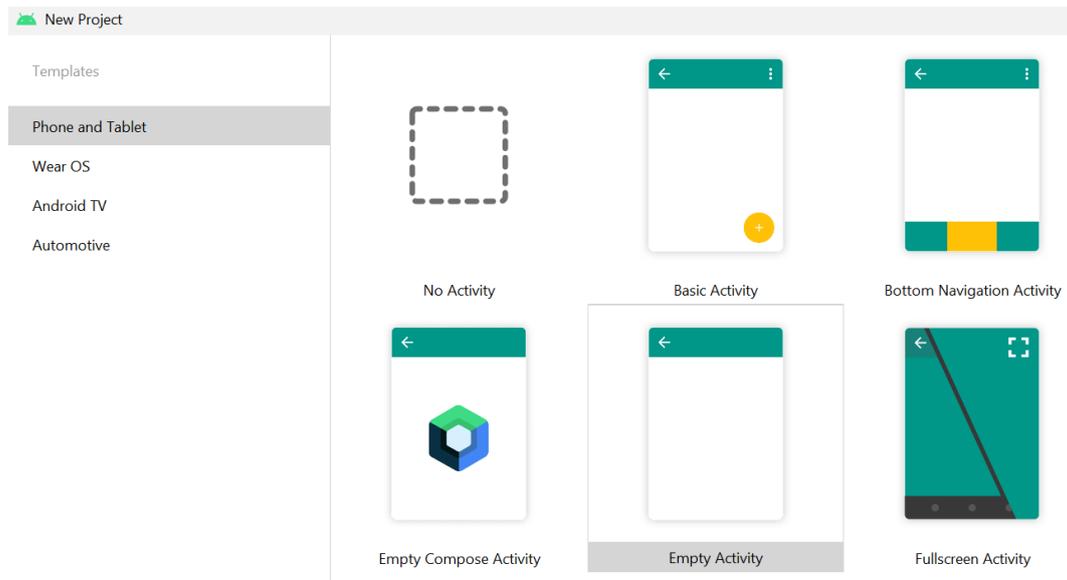
Captura de tela da app

Criar a aplicação Hello World

Criar um Novo Projeto

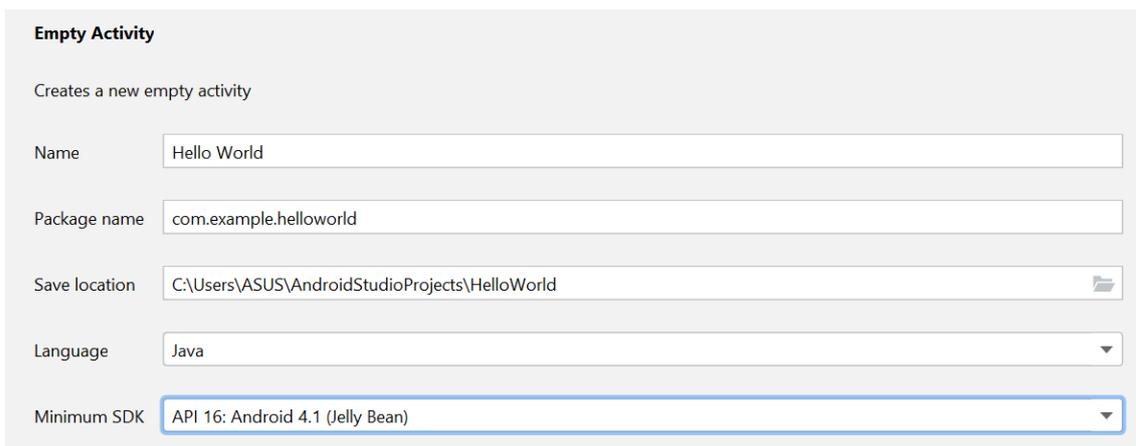
Abra o Android Studio se ainda não estiver aberto. Na janela principal “**Bem-vindo ao Android Studio**”, clique em **Novo projeto**.

Escolha **Phone and Tablet** como o dispositivo para o projeto. Escolha uma **Activity**. O Android Studio fornece estes modelos (templates) com layouts pré-feitos, de modo a dar um avanço na criação de novos projetos. Para este simples projeto Hello World, escolha **Empty Activity**.



Na janela seguinte, defina o **nome** da aplicação como Hello World. Mantenha **com.example.helloworld** para **Package name** (também conhecido como **Domínio**).

O local do projeto (**Save Location**) é na pasta padrão `\Utilizador\AndroidStudioProjects`:

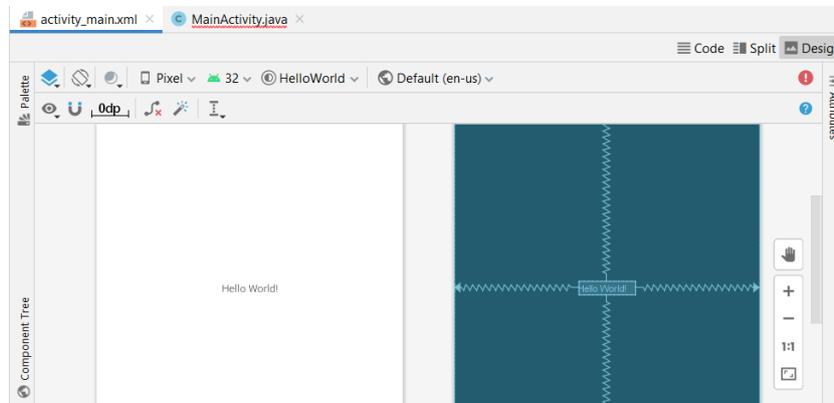
The image shows the 'Empty Activity' configuration screen. It has a title 'Empty Activity' and a subtitle 'Creates a new empty activity'. There are five input fields: 'Name' with 'Hello World', 'Package name' with 'com.example.helloworld', 'Save location' with 'C:\Users\ASUS\AndroidStudioProjects\HelloWorld', 'Language' with 'Java', and 'Minimum SDK' with 'API 16: Android 4.1 (Jelly Bean)'. Each field has a small icon to its right (a folder icon for 'Save location' and a dropdown arrow for 'Language' and 'Minimum SDK').

Selecione o kit de software (**SDK**) mínimo: **API 16: Android 4.1 Jelly Bean**.

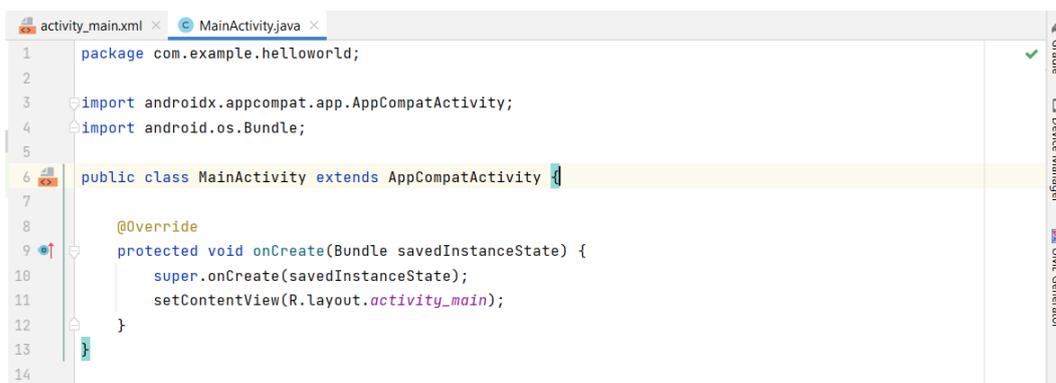
Esta é a interface integrada (**API**) mais antiga ainda retro-compatível em 2022. Funciona em quase todos (99%) os dispositivos Android.

Explorar o painel de Projeto

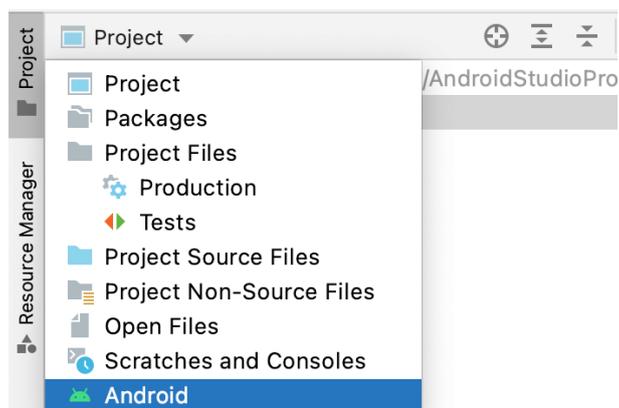
Após criar o projeto, o editor do Android Studio é exibido. Clique no separador “**activity_main.xml**” para ver o editor de layout. Clique na guia **Design** do editor de *layout* no canto direito do painel para mostrar uma representação gráfica do layout:



Clique no separador **MainActivity.java** para ver o editor de código:

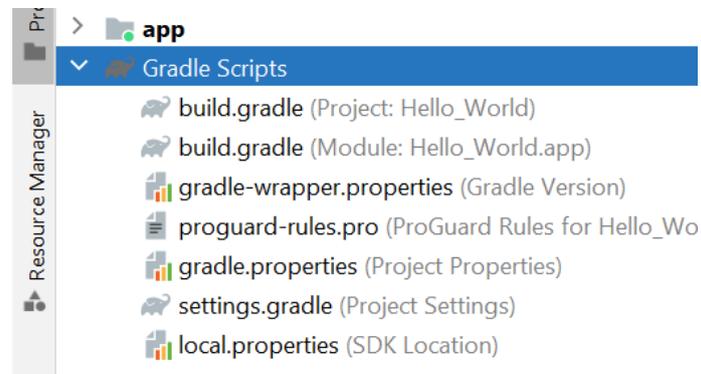


Clique no separador **Projeto** na coluna de separadores verticais no lado esquerdo do Android Studio. Para visualizar o projeto na hierarquia-padrão do Android, escolha **Android** no menu da parte superior do painel Projeto:



Explorar a pasta de scripts Gradle

O sistema de compilação do **Gradle** facilita a inclusão de binários externos ou outros módulos de biblioteca como dependências. A pasta **Gradle** contém todos os ficheiros necessários para o sistema de compilação. Expanda a pasta **Gradle Scripts** do painel de projeto:



Abra o ficheiro **build.gradle (Project: Hello_World)**:

```
1 // Top-level build file where you can add configuration options
2 buildscript {
3     repositories {
4         google()
5         mavenCentral()
6     }
7     dependencies {
8         classpath "com.android.tools.build:gradle:7.0.3"
9     }
}
```

Este ficheiro contém os blocos de configuração comuns a todos os módulos que compõem o projeto. Todos os projetos Android Studio tem este ficheiro único de compilação de alto nível (top-level build file). Na maioria das vezes, não é preciso fazer alterações a este ficheiro. Por padrão, o **build.gradle** usa a secção **buildscript** para definir os repositórios e dependências do Gradle que são comuns a todos os módulos do projeto. Quando a dependência é diferente de uma biblioteca local, o Gradle irá procurar ficheiros nos repositórios online, especificados na secção **repositories**. O Android Studio declara o Google e o [Maven](#) como os locais padrão de repositório.

Abra o ficheiro **settings.gradle (Project Settings)**:

```
1 dependencyResolutionManagement { DependencyResolutionManagement it ->
2     repositories { RepositoryHandler it ->
3         google()
4         mavenCentral()
5         jcenter() // Warning: this repository is going to shut down soon
6     }
7 }
```

Aqui estão as configurações do repositório a nível do projeto e os módulos a serem incluídos ao criar a aplicação. Por padrão, este ficheiro de configurações (settings) usa o bloco **pluginManagement** para configurar os repositórios que o Gradle usa para pesquisar ou baixar os plugins do Gradle e suas dependências transitivas. O bloco **dependencyResolutionManagement** configura os repositórios e dependências usados por todos os módulos do projeto.

Abra o ficheiro **build.gradle (Module: Hello_World.app)**. Além do ficheiro de projeto **build.gradle**, cada módulo possui um **build.gradle** próprio, que permite definir as configurações de compilação para cada módulo específico. Esta aplicação Hello World tem apenas um módulo.



```
1 plugins {id 'com.android.application'}
4 android {
5     compileSdk 31
6     defaultConfig {...}
14    buildTypes {...}
20    compileOptions {...}
24 }
25 dependencies {
26     implementation 'androidx.appcompat:appcompat:1.3.1'
27     implementation 'com.google.android.material:material:1.4.0'
28     implementation 'androidx.constraintlayout:constraintlayout:2.1.1'
29     testImplementation 'junit:junit:4.+'
30     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
31     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
32 }
```

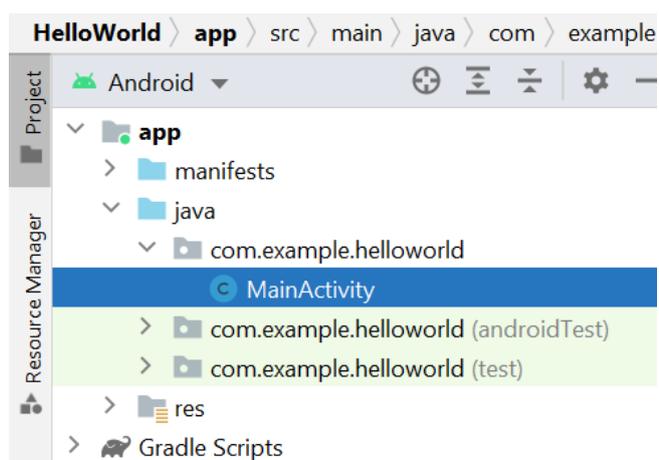
Ao definir essas configurações de compilação (build settings) fornecemos opções de empacotamento customizado (custom packaging options), tais como tipos adicionais de compilação e variações de produto. Podemos substituir estas configurações no ficheiro **AndroidManifest.xml** ou no **build.gradle** de alto nível. Este **build.gradle** é frequentemente o ficheiro a ser editado para alterar configurações de alto nível, tais como declarar dependências no bloco **dependencies**.

Quando fazemos alterações nos ficheiros **build** de um projeto, o Android Studio requer que façamos **sincronização** para poder importar as mudanças na configuração do projeto, e executar verificações para garantir que a nova configuração não crie erros de compilação.

Explorar as pastas App e Res

No painel **Project**, o Android Studio organiza as pastas e ficheiros de maneira diferente do que no Explorador de Ficheiros do Windows. Todos os ficheiros de fonte (source files) da aplicação estão localizados na pasta **\app**.

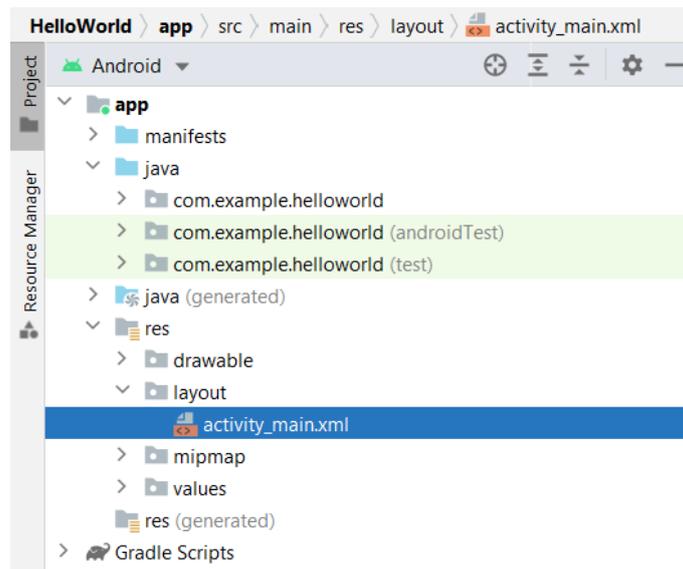
Expanda a pasta **\java** e a pasta **\com.example.helloworld** para ver o ficheiro **MainActivity.java**:



A pasta **\app** contém a subpasta **\java**, que inclui ficheiros de classe Java em três subpastas.

A subpasta **\com\example\helloworld** contém todos os ficheiros de domínio (Package) da aplicação. As outras duas subpastas (**\androidTest** e **\test**) são usadas para teste, e serão explicadas noutra aula.

Expanda a pasta **res** e a pasta de *layout* e clique duas vezes no ficheiro **activity_main.xml** para abri-lo no editor de layout.



A pasta **res** contém recursos, como layouts, strings e imagens. Uma **Activity** geralmente está associada a um *layout* da interface de utilizador definido como um ficheiro XML. Este ficheiro geralmente é nomeado consoante a sua **Activity** correspondente.

Explorar a pasta Manifests

A pasta **manifests** contém ficheiros com informações essenciais sobre a aplicação para o sistema Android, as quais o sistema tem de obter antes de executar qualquer código da aplicação.

Expanda a pasta **manifests** e abra o ficheiro **AndroidManifest.xml**

O ficheiro **AndroidManifest.xml** descreve todos os componentes da aplicação.

Todos os componentes de uma aplicação, como as **Activity**, devem ser declarados neste ficheiro.

Noutras aulas deste curso, iremos modificar este ficheiro para adicionar recursos e permissões.

Para obter uma introdução, consulte [Visão geral do manifesto da aplicação](#).

```
activity_main.xml x AndroidManifest.xml x MainActivity.java x
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.helloworld">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Hello World"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/Theme.HelloWorld">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

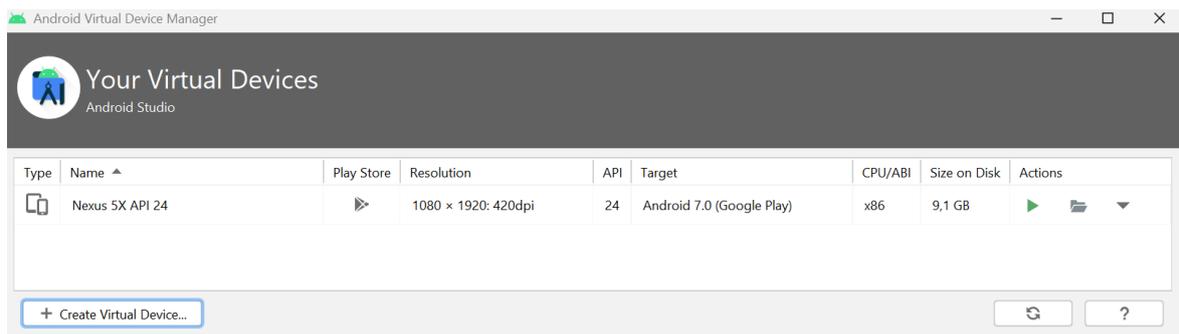
</manifest>
```

Criar um dispositivo virtual (Emulador)

Para criar um dispositivo virtual (emulador) usaremos o gestor [Android Virtual Device \(AVD\)](#) que simula a configuração de um tipo específico de dispositivo Android. Note que este emulador tem [requisitos adicionais](#) além dos requisitos básicos para o Android Studio; em particular, 8 Gb de RAM são recomendados. Com o AVD Manager, podemos definir as características de hardware de um dispositivo, o nível de API, armazenamento, aparência e outras propriedades, e o gravamos como um dispositivo virtual.

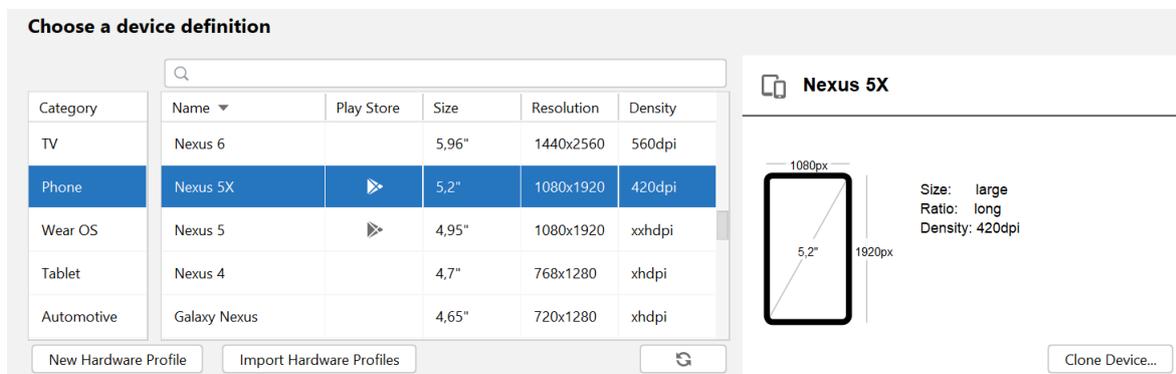
Selecione **Tools > AVD Manager** ou clique no ícone AVD Manager  na barra de ferramentas.

A tela **Your Virtual Devices** aparece:

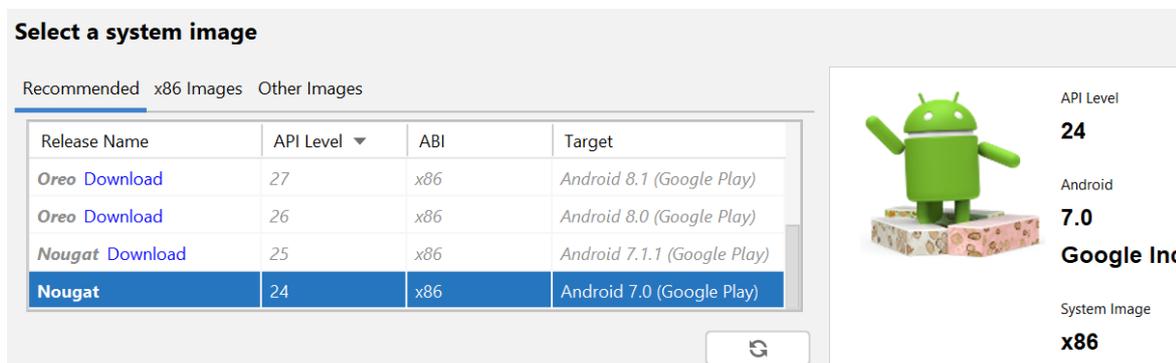


Clique em **+ Create Virtual Device**. A janela **Select Hardware** aparece com uma lista de categorias e dispositivos pré-configurados. Para cada dispositivo, a tabela mostra colunas para tamanho de exibição diagonal (**size**), resolução (**resolution**), e densidade de pixels (**density**).

Escolha um dispositivo, como o **Nexus 5X**.



A tela **System Image** (imagem do sistema) aparece.



No separador **Recommended**, escolha a versão **Nougat** do sistema Android 7.0 para download. Após baixar a imagem do sistema, dê um nome ao seu dispositivo virtual (**Nexus 5X API 24**) e mantenha as configurações padrão.

No Android Studio, clique em **Run > Run > 1. App**, ou clique no ícone  na barra de ferramentas.

A aplicação é compilada com o **Gradle**.

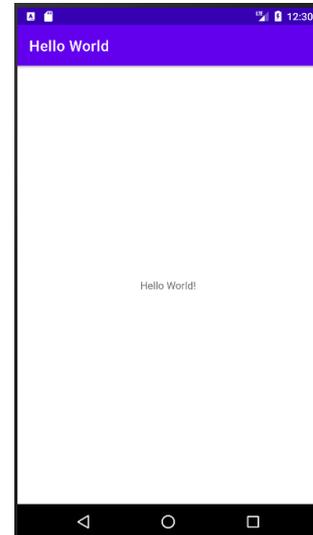
Clique no **AVD Manager** para abrir a tela **Your Virtual Devices**, e clique no ícone da coluna **Actions** –  **Run** para abrir o emulador.

O Android Studio fará o upload da aplicação para o emulador e a executará automaticamente.

Dica: Ao testar num dispositivo virtual, é uma boa prática iniciá-lo uma vez, no início da sessão.

Não se deve fechar o emulador, ao invés, deve-se antes parar a aplicação com o botão , de modo a que ele não precise passar pelo processo de inicialização do dispositivo novamente.

Um atalho para fechar o dispositivo virtual são as teclas **Control + Q**.



Testar a aplicação num dispositivo físico

Nesta tarefa, vamos executar a aplicação num dispositivo móvel físico, como um telefone ou tablet. É preciso um cabo de dados para conectar seu dispositivo físico ao computador através da porta USB.

Se estiver a usar um sistema Linux ou Windows, poderá ser necessário executar etapas adicionais para executar em um dispositivo de hardware. Verifique a documentação [Usando dispositivos de hardware](#).

Também poderá ser necessário instalar o driver USB apropriado para o seu dispositivo.

Para drivers USB baseados em Windows, consulte [OEM USB Drivers](#).

Para permitir que o Android Studio comunique com o dispositivo, devemos ativar a **depuração USB** no dispositivo, através da tela de **Opções de Desenvolvedor** (Developer Options). Nas versões Android 4.2 e superior, a tela de opções do desenvolvedor fica oculta por padrão.

No seu dispositivo, abra **Configurações** (Settings), procure e toque no **Sobre o telefone** (About phone), e toque no **Número de compilação** (Build number) sete vezes.

Retorne à tela anterior (**Configurações/Sistema**). As **opções de desenvolvedor** irão aparecer na lista. Toque nas Opções do desenvolvedor e selecione **Depuração USB** (USB Debugging).

Conecte o dispositivo ao computador com um cabo USB.

No Android Studio, escolha **Executar (Run)**, depois clique em **Selecionar dispositivo (Select Device)**. De seguida, clique no botão **Executar**  na barra de ferramentas. O Android Studio automaticamente instala e executa a aplicação no dispositivo físico.

Alterar configurações do Gradle

Nesta tarefa, iremos alterar algo no ficheiro **build.gradle** para aprender como fazer alterações e como sincronizá-las no projeto Android Studio.

Expanda a pasta **Gradle Scripts** e clique duas vezes no ficheiro **build.gradle (Module:Hello_World.app)**.



```
build.gradle (:app) x
You can use the Project Structure dialog to view and edit your project configuration
1  plugins {id 'com.android.application'}
4  android {
5      compileSdk 31
6      defaultConfig {...}
14     buildTypes {...}
20     compileOptions {...}
24 }
25 dependencies {
26     implementation 'androidx.appcompat:appcompat:1.3.1'
27     implementation 'com.google.android.material:material:1.4.0'
28     implementation 'androidx.constraintlayout:constraintlayout:2.1.1'
29     testImplementation 'junit:junit:4.+'
30     androidTestImplementation 'androidx.test.ext:junit:1.1.3'
31     androidTestImplementation 'androidx.test.espresso:espresso-core:3.4.0'
32 }
```

No bloco **defaultConfig{...}**, altere o valor de **minSdk** 16 para 23.



```
6  defaultConfig {
7      applicationId "com.example.helloworld"
8      minSdk 16
9      targetSdk 31
10     versionCode 1
11     versionName "1.0"
12     testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
13 }
```

O editor irá mostrar na parte superior uma barra de notificação **Sincronizar agora (Sync Now)**.

Quando fazemos alterações nos ficheiros **build** de um projeto, o Android Studio requer que façamos **sincronização** para poder importar as mudanças na configuração do projeto, e executar verificações para garantir que a nova configuração não crie erros de compilação.

Para sincronizar os ficheiros do projeto, clique em **Sincronizar agora** na barra de notificação que aparece ao fazer uma alteração ou clique no ícone **Sincronizar projeto com ficheiros Gradle**  na barra de ferramentas.

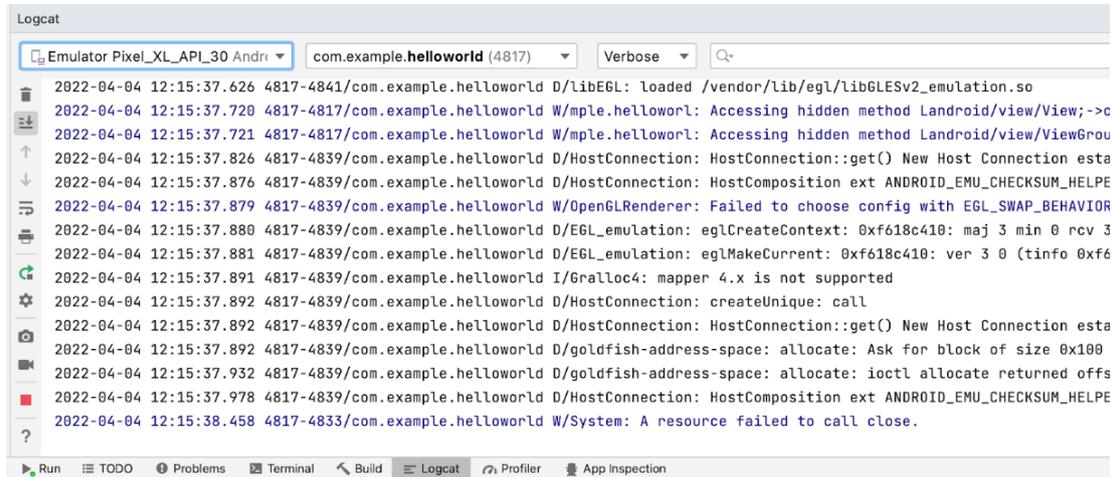
Quando a sincronização do Gradle for concluída, a mensagem **'Gradle build finished'** aparecerá no canto inferior esquerdo da janela do Android Studio.

Para uma visão mais detalhada do Gradle, confira a documentação [Build System Overview](#) e [Configuring Gradle Builds](#).

Adicionar mensagens de registro Log

Agora iremos adicionar instruções **Log** à aplicação, que exibem mensagens no painel **Logcat**. Mensagens **Log** são uma ferramenta de depuração que pode ser usada para verificar variáveis, caminhos de execução (paths) e relatar exceções (exceptions).

Clique no separador **Logcat** na parte inferior da janela do Android Studio:



O separador **Logcat** exibe informações sobre a aplicação durante a execução. Se adicionarmos mensagens **Log** à aplicação, as mensagens **Log** aparecerão neste painel.

O menu de nível Log está definido como **Verbose** (palavroso), o qual mostra todas as mensagens Log. Outras configurações incluem **Debug**, **Error**, **Info**, **Warn** e **Assert**.

As mensagens Log são escritas no código Java e são compostas por três partes. Por exemplo:

```
Log.d("MainActivity", "Hello World");
```

Log: instanciação da classe **Log** para enviar mensagens de log para o painel **Logcat**.

d: configuração do nível de **depuração (debug)** para filtrar a mensagem que exibida no Logcat.

"MainActivity": o primeiro argumento é um marcador (**tag**) que pode ser usado para identificar mensagens no Logcat. Geralmente usa-se o nome da Activity da qual a mensagem originou.

No entanto, podemos escrever qualquer coisa que seja útil para identificar durante a depuração.

"Hello World": o segundo argumento é a mensagem final.

Por convenção, as **tags** são definidas como constantes na Activity do seguinte modo:

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

Vamos seguir os seguintes passos:

Abra a aplicação Hello World no Android studio, e abrir a classe **MainActivity** no editor de código.

Para adicionar importações inequívocas automaticamente ao código (como **android.util.Log**), vá para **File > Settings**, escolha **Editor > General > Auto Import**. Marque todas as caixas de seleção e defina **Insert Imports on Paste** como **Always**. Clique em **Apply**, e em seguida clique em **OK**.

No método `onCreate()` da `MainActivity`, adicione a linha `Log` usada como exemplo no final do método:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.d("MainActivity", "Hello World");
}
```

Abra o painel `Logcat` e verifique se o nome do destino e de domínio (`package`) da aplicação estão corretos. Mude o nível de `Log` no painel `Logcat` para `“Debug”` (ou deixe como `“Verbose”`, visto que só há uma mensagem `Log`), e corra a App.

A seguinte mensagem deverá aparecer no painel `Logcat`:

```
11-24 14:06:59.001 4696-4696/? D/MainActivity: Hello World
```

Dica: Se ver demasiadas mensagens de log extras do seu emulador, poderá usar o `filtro` (lupa) para ver apenas as mensagens de `Log` contidas na `MainActivity`.

Anexo H

Correspondência entre Expressões Informáticas usadas em Portugal e no Brasil (Queirós, 2016, p.293).

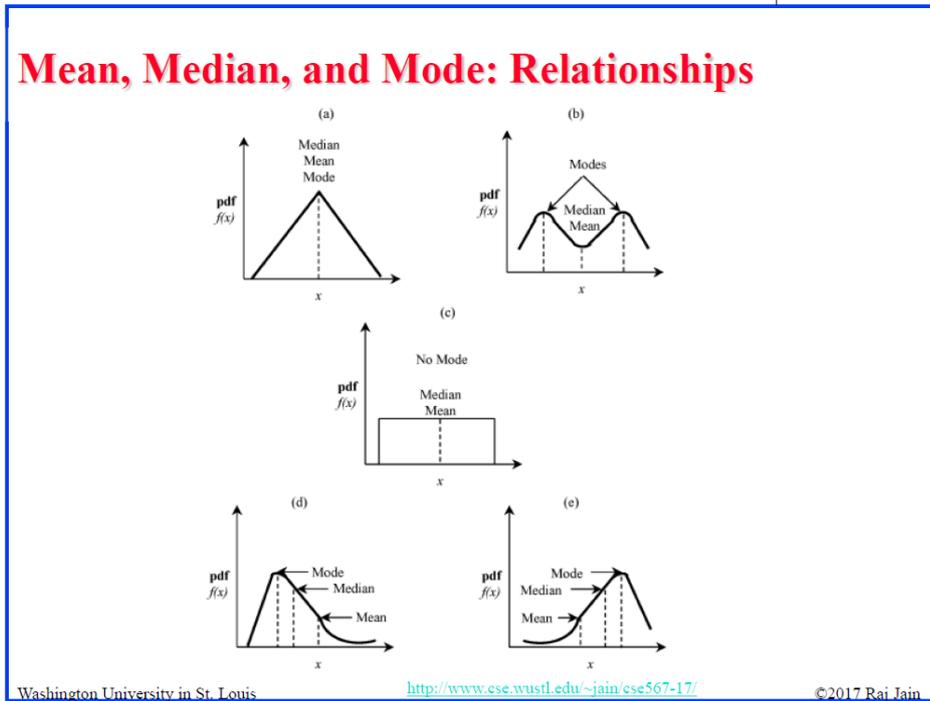
GLOSSÁRIO DE TERMOS PORTUGUÊS EUROPEU* / PORTUGUÊS DO BRASIL

Para facilitar a utilização deste livro pelos leitores brasileiros, incluímos este glossário de termos.

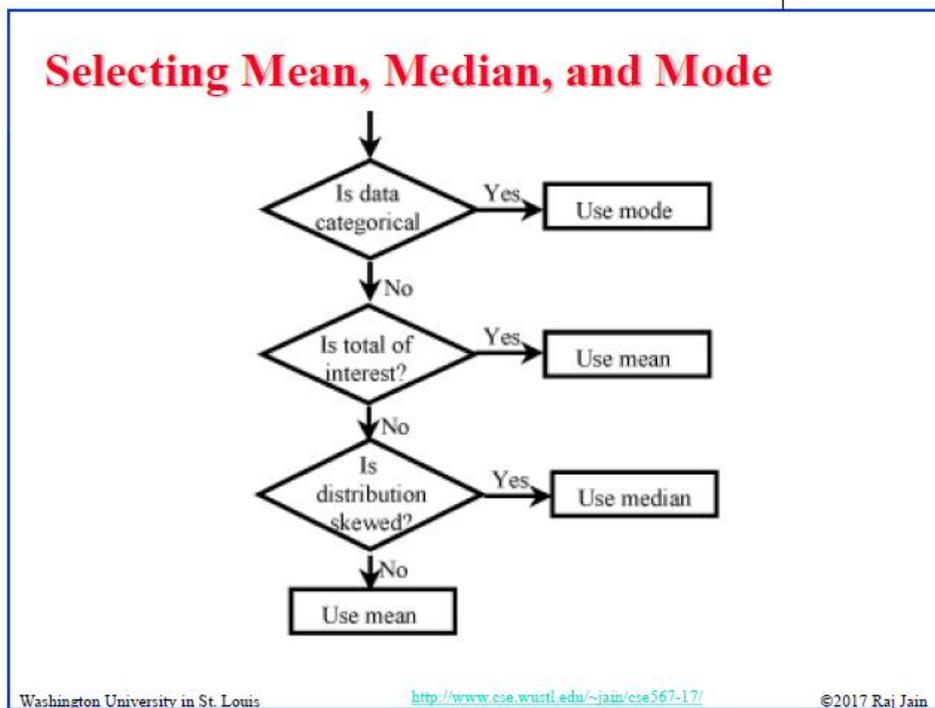
PORTUGUÊS EUROPEU*	PORTUGUÊS DO BRASIL
Aceder	Acessar
Aplicação	Aplicativo
Arranque	Inicialização
Barra de estado	Barra de status
Base de dados	Banco de dados
Diretório	Diretoria
Ecrã	Tela
Ficheiro	Arquivo
Fornecedor	Provedor
Gestão	Gerenciamento
Guardar/Gravar	Salvar
Ligação	Conexão
Ligar	Conectar
Modelo	Padrão
Monitorizar	Monitorar
Normalização	Estandardização, padronização
Palavra-passe	Senha
Personalizar	Customizar
Rato	Mouse
Registo	Registro
Separador	Aba
Sistema operativo	Sistema operacional
Utilizador	Usuário

* Designa-se por Português Europeu a variante da língua falada em Angola, Cabo Verde, Guiné-Bissau, Moçambique, Portugal, São Tomé e Príncipe e Timor-Leste.

Medidas de Tendência Central em Distribuição Normal e Não Normal



12-19



12-20

Jain, R. (2017). Summarizing Measured Data. Washington University, St Louis, USA. Disponível em https://www.cse.wustl.edu/~jain/cse567-17/ftp/k_12smd.pdf