

Departamento de Ciências e Tecnologias de Informação

## A Robótica no Ensino e Aprendizagem de Programação

António Sérgio Gomes Pereira

Dissertação submetida como requisito parcial para obtenção do grau de

Mestre em Engenharia Informática

Orientador:

Doutor Carlos J. Costa, Professor Associado do ISEG (Lisbon School of Economics and Management), Universidade de Lisboa

Coorientadora:

Doutora Manuela Aparício, Professora Auxiliar Convidada do DCTI ISCTE-IUL

Outubro, 2017

# Agradecimentos

Gostaria de agradecer à minha família em especial à minha esposa e à minha irmã e aos meus amigos por toda a força e apoio incondicionais. Agradeço ao meu orientador Professor Doutor Carlos Costa e à minha coorientadora Professora Doutora Manuela Aparício, cuja sabedoria, amabilidade, dedicação e disponibilidade durante todo este processo representaram um suporte de valor incalculável, que me permitiram criar e desenvolver este projeto. Quero também agradecer ao Professor Doutor Fernando Duarte pelo apoio e contributo para a realização deste trabalho.

Muito obrigado por acreditarem em mim.

# Resumo

O presente estudo tem como objetivo verificar o real impacto que a utilização da robótica pedagógica no ensino-aprendizagem da programação pode ter na melhoria do desempenho dos alunos. A robótica promove no aluno, além da motivação, outros aspetos positivos para a sua aprendizagem como, raciocínio lógico, trabalho em equipa, interdisciplinaridade, autonomia. A proposta de intervenção envolveu a construção e programação de um *kit* de robótica e a utilização da plataforma Arduino. As estratégias da sua implementação assentaram na utilização da aprendizagem baseada em problemas, com recurso à robótica pedagógica e do trabalho colaborativo. Os resultados demonstram que a utilização da robótica pedagógica contribuiu para o cumprimento do objetivo proposto.

## Palavras-chave

Robótica pedagógica, ensino da programação, aprendizagem, *kits* didáticos, Arduino, modelo de adoção tecnológica.

# Abstract

The present study verifies the real impact that the use of pedagogic robotics in teaching-learning programming can have in improving student's performance. Robotics promotes in the student, besides motivation, other positive aspects for their learning as, logical reasoning, teamwork, interdisciplinarity, autonomy. The intervention proposal involved the construction and programming of a robotic kit and the use of the Arduino platform. The strategies of its implementation were based on the use of problem-based learning, using pedagogic robotics and collaborative work. The results demonstrate that the use of pedagogical robotics contributed to the accomplishment of the proposed objective.

## Keywords

Robotics pedagogic, programming teaching, learning, teaching kits, Arduino, technology adoption model.

# Índice

1 - Introdução.....	11
1.1 Motivação.....	11
1.2 Questão de investigação e Objetivos do trabalho .....	12
1.3 Abordagem metodológica.....	12
1.4 Estrutura da dissertação .....	13
2 – Revisão da Literatura.....	14
2.1 Ensino-Aprendizagem da Programação .....	17
2.3 Problemática do Ensino-Aprendizagem da Programação.....	18
2.4 Robótica Pedagógica. ....	18
2.4.1 Limitações e Desafios da Robótica Pedagógica.....	23
2.5 Aprendizagem por resolução de problemas (ABRP) .....	24
2.6 Projeto Arduino .....	26
2.6.1 <i>Hardware</i> .....	27
2.6.2 <i>Software</i> .....	30
2.7 <i>Kit de led’s e sensores</i> .....	31
2.8 Modelo de aceitação da tecnologia .....	31
3 – Trabalho Empírico .....	33
3.1 Introdução .....	33
3.2 Caracterização do estudo.....	34
3.3 Instrumentos .....	35
4 - Proposta de intervenção .....	38
4.1 Programa .....	38
4.2 Planificação da aula .....	38
4.3 Descrição da aula .....	39
4.4 Desafio.....	41
5 - Resultados.....	43
6 - Discussão.....	56
7 – Conclusões, Contribuições e Trabalhos Futuros .....	60
7.1 - Conclusões .....	60
7.2 - Contribuições da Pesquisa .....	61
7.3 - Trabalhos Futuros e Limitações .....	62
Referências.....	65
ANEXO A – Desenvolvimento do <i>kit de led’s e sensores</i> .....	71

ANEXO B – Software para teste do <i>kit</i> de <i>led</i> 's e sensores .....	85
ANEXO C – Planificação da aula .....	89
ANEXO D – Resultado do 1º desafio.....	92
ANEXO E – Resultado dos questionários.....	94
ANEXO F – Apresentação da aula.....	106
ANEXO G – Perguntas dos questionários .....	122
Questionário 1.....	122
Questionário 2.....	123
ANEXO H – Questionários <i>online</i> .....	124

# Índice de Tabelas

Tabela 1 – Quantidade de memória disponível em cada modelo de microcontrolador .....	28
Tabela 2 – Critérios de Qualidade.....	51
Tabela 3 – Resultado das Hipóteses .....	52
Tabela 4 –Correlação entre variáveis .....	52
Tabela 5 - Alpha de Cronbach das Escalas .....	52
Tabela 6 - AVE, Confiabilidade Compósita, Coeficiente de Determinação .....	53
Tabela 7 – Validade dos constructos.....	54
Tabela 8 – Análise Bootstrap .....	54
Tabela 9- Material para a montagem.....	78

# Índice de Figuras

Figura 1 - Uma das primeiras tartarugas criada por Seymour Papert (Walter, 2010) .....	20
Figura 2 - Uma tartaruga de água doce (Walter, 2010).....	20
Figura 3 - Etapas da ABRP e competências (Adaptado de Woods 1994).....	25
Figura 4 - Placa eletrônica Arduíno Uno.....	27
Figura 5 - Componentes do microcontrolador .....	27
Figura 6 - Blocos de um microcontrolador .....	28
Figura 7 - Sinal com <i>duty-cycle</i> variável .....	29
Figura 8 - IDE do Arduíno .....	30
Figura 9 - Ciclo de desenvolvimento .....	30
Figura 10 - <i>Kit de led's</i> e sensores .....	31
Figura 11 – Modelo de Aceitação de tecnologia -TAM.....	32
Figura 12 – Modelo de investigação de adoção de Arduíno na aprendizagem da programação. 36	
Figura 13 – Apresentação da aula .....	39
Figura 14 - Acompanhamento dos trabalhos .....	40
Figura 15 – Interação com o Kit.....	41
Figura 16 - Objetivo da tarefa .....	44
Figura 17 - Compreensão dos conceitos de programação .....	44
Figura 18 - Dificuldades na compreensão da tarefa proposta .....	45
Figura 19 - Dificuldades em compreender as respostas ( <i>feedback</i> ) das ações programadas.....	45
Figura 20 - A atividade prática contribuiu para uma melhor compreensão de novos conceitos de programação.....	45
Figura 21 - A Robótica Pedagógica é um instrumento interessante para ensino de programação .....	47
Figura 22 - utilização do “hardware” .....	48
Figura 23 - As respostas dadas pelo “hardware” .....	48
Figura 24 - Os recursos ( <i>led's</i> , sensores, etc.) foram percebidos de maneira clara.....	48
Figura 25 - A utilização do “ <i>IDE</i> ” ocorreu sem problemas .....	49
Figura 26 - Houve problema em compreender os objetivos da prática proposta .....	49
Figura 27 - Teve dificuldade em elaborar a solução da prática proposta.....	50
Figura 28 - Houve dificuldades em aplicar os conceitos de Lógica de Programação (exemplo: estruturas de decisão, estruturas de repetição) na solução .....	50
Figura 29 - Avaliação da Atividade .....	50
Figura 30 – Modelo de investigação .....	51
Figura 31 – Resultados do Modelo de adoção de Arduíno em contexto de aprendizagem de Programação.....	55
Figura 32 - Questões VS Ocorrências .....	57
Figura 33 - <i>Pinout</i> Arduíno UNO .....	71
Figura 34 - Resistências .....	72
Figura 35 - Código de Cores das Resistências .....	72
Figura 36 - LDR (Componentes) .....	73
Figura 37 - Gráfico de Resposta de uma LDR.....	73
Figura 38 - Valores máximo e mínimo da tensão de leitura da <i>LDR</i> .....	74
Figura 39 - Ligação do <i>LDR</i> .....	74
Figura 40 - Interior do Potenciômetro.....	74
Figura 41 - Resistência entre os terminais do Potenciômetro .....	75

Figura 42 - Ligação do Potenciômetro .....	75
Figura 43 - Sensor de temperatura .....	76
Figura 44 - Ligação do sensor de temperatura .....	76
Figura 45 - Esquema de ligação do <i>led</i> .....	77
Figura 46 - Corrente no led .....	77
Figura 47 - Implementação da ligação do <i>led</i> .....	77
Figura 48 - Montagem do protótipo .....	77
Figura 49 - <i>Kit de led's</i> e sensores.....	79
Figura 50 - Código em C para teste dos <i>led's</i> .....	85
Figura 51 - Código em C para teste do sensor de temperatura.....	86
Figura 52 - Código em C para o teste do sensor de luz.....	87
Figura 53 - Código em C para teste do potenciômetro.....	88
Figura 54 - Idade dos participantes .....	94
Figura 55 - Sexo dos participantes .....	94
Figura 56 - Dificuldades na compreensão da tarefa .....	95
Figura 57 - Objetivo da tarefa .....	95
Figura 58 - Compreensão dos conceitos de programação .....	95
Figura 59 - Compreensão das respostas das ações programadas .....	96
Figura 60 - Compreensão de novos conceitos de programação .....	96
Figura 61 - Adequação da tarefa .....	96
Figura 62 - Modelo interessante para a programação .....	97
Figura 63 - Dificuldade em utilizar o <i>hardware</i> .....	97
Figura 64 - Compreensão dos recursos disponibilizados .....	97
Figura 65 - respostas dadas pelo <i>hardware</i> .....	98
Figura 66 - Utilização da RP aumentou o conhecimento em programação .....	98
Figura 67 - Utilização do <i>IDE</i> .....	98
Figura 68 - Dificuldade nos conceitos de programação .....	99
Figura 69 - Compreensão dos objetivos da tarefa .....	99
Figura 70 - Dificuldade em elaborar a tarefa .....	99
Figura 71 - Dificuldade em aplicar os conceitos de lógica de programação.....	100
Figura 72 - Avaliação da atividade .....	100

# Abreviaturas

ABRP – Aprendizagem Baseada na Resolução de Problemas

EEPROM – Electrically-Erasable Programmable Read-Only Memory

IDE – Integrated Development Environment

LDR – Light Dependent Resistor

LED – Light Emitting Diode

SRAM – Static Random Access Memory

RP – Robótica Pedagógica

PWR – Power

PWM – Pulse Width Modulation

USB – Universal Serial Bus

TAM – Modelo de Aceitação da Tecnologia

PEOU – Facilidade de Uso Percebida

PU – Utilidade Percebida

IU – Intenção de Uso

AVE - Variância Média Extraída

# 1 - Introdução

*“Everybody should learn how to program a computer, because it teaches you how to think.” STEVE JOBS (1995)*

O presente trabalho, realizado no âmbito da tese em “A Robótica no Ensino e Aprendizagem de Programação”, foi desenvolvido no ISCTE, no curso de Mestrado em Engenharia Informática. Apresenta uma pesquisa realizada com a participação de estudantes, que possibilitou observar a aplicação da Robótica Pedagógica com forma de aprendizagem da programação e resolução de problemas.

O ensino e a aprendizagem da programação constituem um enorme desafio para professores e alunos. Os elevados níveis de insucesso nas disciplinas introdutórias de programação, independentemente do grau de ensino são alvo de preocupação constante, pelo que têm sido apresentadas várias propostas conducentes à melhoria dos resultados. A hipótese foi a introdução no ambiente da aula de um instrumento, “O Arduíno e o Kit didático”, que permita ao aluno observar o comportamento dos algoritmos (programas de computadores) desenvolvidos por ele e executados pelo Arduíno e as interações deste com o ambiente em tempo real, pudesse favorecer outras formas de perceção e de estímulo, contribuindo efetivamente para a aprendizagem de técnicas de programação de computadores.

## 1.1 Motivação

A emergente sociedade de informação, cada vez mais avançada em termos científicos, sustentada pelo uso generalizado de tecnologias e orientada no intuito de acompanhar os alunos nativos digitais (Prensky, 2001), tem em conta que o interesse dos alunos pelas novas tecnologias é cada vez maior e que a Robótica Pedagógica tem vindo a afirmar-se nos últimos anos como uma ferramenta extremamente vantajosa, ainda que emergente, em especial ao nível da educação científica e tecnológica (Ribeiro, Coutinho & Costa, 2009). Nesta perspetiva, utilizou-se a Robótica Pedagógica associada à estratégia de aprendizagem baseada em problemas e à aprendizagem colaborativa com o intuito de potenciar as aprendizagens dos alunos, levando-os a realizar as tarefas de aprendizagem

de forma lúdica e divertida (Kumar, 2001, citado por Koski, Kurhila & Pasanen, 2008) e levando-os a ser ativos na sua própria aprendizagem.

## 1.2 Questão de investigação e Objetivos do trabalho

No decorrer desse desafio vão ser propostas atividades diversificadas com ênfase na programação contextualizadas, e sempre que possível, apoiadas nos módulos de programação, para verificar a sua adequação às orientações e recomendações curriculares, recolher dados, analisar e avaliar o desempenho e as aprendizagens dos estudantes.

Definiu-se a seguinte questão de investigação:

- **Será que a utilização da Robótica Pedagógica contribui para a aprendizagem da programação?**

Para responder à questão de investigação foram definidos os seguintes objetivos:

### Objetivo geral

Explorar, por meio da percepção dos estudantes, o desenvolvimento de atividades de ensino-aprendizagem da programação, utilizando-se como ferramenta o Arduino.

### Objetivos específicos

- a) Elaborar, atividades que utilizem Robótica Pedagógica para o ensino-aprendizagem da programação, com base na literatura disponível respeitante à aplicação da Robótica Pedagógica para o ensino da programação e tendo como referencial a pedagogia de projetos, visando a escolha da metodologia de aplicação mais adequada aos participantes;
- b) Descrever a participação e o ponto de vista dos estudantes sobre a experiência vivenciada.

## 1.3 Abordagem metodológica

O presente estudo no tema “*A Robótica Pedagógica no ensino e aprendizagem da programação*” seguiu uma abordagem metodológica de design science research (Jarvinen, 2007; March & Smith, 1995), consistindo na proposta de uma solução que utiliza artefactos para solucionar um problema. A abordagem seguida consistiu na

apresentação de uma atividade prática a realizar em sala de aula em que os estudantes, em trabalho de grupos, recorrem à robótica para expressar raciocínios e ideias, através da resolução de tarefas de programação de *kits* didáticos, propositadamente criados para esta investigação. Baseia-se num trabalho orientado para a mudança pedagógica, na implementação de práticas interdisciplinares e na criação de ambientes de aprendizagem diversificados e motivadores.

Os resultados do presente estudo evidenciam que programar num sistema eletrónico, controlando-o a partir de um computador com *software* especificados, permite ao aluno assumir um papel ativo na construção dos seus próprios conhecimentos, reconhecendo a importância da reflexão sobre as decisões tomadas, aprendendo com os erros, tentando evitar repeti-los.

Verificou-se que a resolução de problemas que envolveram ações comandadas pelos alunos, despertou a atenção e a curiosidade dos grupos. Acresce ainda que durante a resolução dos desafios propostos, os estudantes demonstram maior interesse e envolvimento no estudo da programação.

Observou-se ainda que, dependendo da forma como as atividades envolvendo os *kits* didáticos são estruturadas, estas podem ser tratadas conceitualmente como desafios que incorporam o lúdico à aprendizagem de conteúdos escolares. Neste contexto espera-se que a utilização da Robótica Pedagógica venha a apoiar o processo de aprendizagem ao trazer implementar de mais recreativo e menos complexo, uma vez que os estudantes manifestam preferências por aulas com maior componente prática.

#### 1.4 Estrutura da dissertação

A presente tese está organizada em cinco capítulos, onde o primeiro apresenta o enquadramento, os objetivos, a abordagem metodológica e a estrutura da dissertação. O segundo capítulo faz uma abordagem da revisão da literatura com particular atenção para programação e descreve também os materiais usados. O terceiro capítulo descreve a metodologia usada neste trabalho. O quarto apresenta os resultados e a sua discussão. O quinto capítulo apresenta as conclusões deste trabalho, bem como as perspetivas futuras.

## 2 – Revisão da Literatura

O ensino da programação tem como propósito conduzir os alunos ao desenvolvimento de capacidades, que lhes permitam adquirir competências e conhecimentos para criarem programas capazes de resolver problemas reais. Gomes e Mendes (2007) referem que, muito embora existam uma série de instrumentos que apresentam resultados positivos, continuam a existir problemas que prevalecem no ensino da programação. Segundo Price, Hirst, Johnson, Petre, e Richards (2002); Santos, *et al.* (2009) os métodos tradicionais de ensino podem originar vários problemas. A insistência na imitação, obediência, repetição e controlo, muito frequentes nestes métodos, conduzem a uma negligência das capacidades criativas individuais em detrimento de competências que são puramente mecânicas e repetitivas. Chella (2002, citado por Santos *et al.*, 2009) refere que, esta não é uma realidade exclusiva dos alunos que revelam problemas de aprendizagem ao longo do seu percurso escolar, e que é notória a dificuldade que os alunos têm em aplicar os conhecimentos teóricos em problemas reais.

Do ponto de vista dos professores o ato de ensinar programação por si só não é fácil, uma vez que, devido a complexidades individuais, os vários alunos revelam diferentes dificuldades a nível da programação (Ala-Mutka, 2003). Destaca-se, por exemplo, a dificuldade em compreender a utilidade da programação e aplicar corretamente a sintaxe das linguagens de programação (Gomes *et al.*, 2008; Saeli *et al.*, 2011). Por outro lado, os alunos manifestam preferência por aulas com maior componente prática, em detrimento de uma exposição excessiva de conteúdos (Gomes *et al.*, 2012). Estes fatores induzem muitos deles a considerar a programação como difícil de aprender. Aprender a programar é muito mais do que “mera codificação”; implica que os alunos pensem de forma distinta (Gal-Ezer & Harel, 1998), o que exige um maior esforço, mesmo na resolução de problemas pouco complexos (Santos *et al.*, 2009). Porém, a experiência tem demonstrado que existe, em termos genéricos, uma grande dificuldade em compreender e aplicar certos conceitos abstratos de programação. Uma grande dificuldade resulta do facto de que os alunos não conseguem compreender e aplicar noções básicas de programação de modo a resolverem problemas concretos. Estas dificuldades são apresentadas independentemente do tipo de linguagem de programação usada.

Vários autores consideram que a programação é uma disciplina muito exigente e difícil de dominar, associando esta dificuldade à sua natureza abstrata. Por exemplo, Dunican (2002) afirma que as noções de variável, tipos de dados, memória, entre outros, não têm

correspondência concreta no dia-a-dia, pelo que se torna de difícil compreensão. Para Perkins *et al.*, (1988) a programação exige um esforço muito significativo envolvendo várias competências de diversas áreas. Para alguns autores, citado por Gomes & Mendes, (2007), Jenkins, (2002) existem algumas causas que podem estar na origem das dificuldades tidas ao nível da programação:

- Ausência de uma estratégia que abranja todos os alunos (Jenkins, 2002; Gomes & Mendes, 2007);
- Falta ou inadequação dos métodos de ensino (Olsson *et al.*, 2015; Pereira *et al.*, 2017);
- A dificuldade dos alunos em utilizar os conhecimentos para a resolução de um novo problema (Gomes *et al.*, 2008).
- A complexidade da sintaxe (Gomes & Mendes, 2007).
- Falta de conhecimento dos alunos de programação, nomeadamente na interpretação de erros (Costa, Aparício & Cordeiro, 2012a; Costa, Aparício & Cordeiro, 2012b; Gomes & Mendes, 2007).

Numa tentativa de superar estas dificuldades, alguns estudos apontam a Robótica Pedagógica como um bom aliado na resolução de alguns dos problemas identificados (Santos *et al.*, 2009).

Estudos realizados pela Associação Japonesa de Robótica [JPA], a Comissão Económica das Nações Unidas [UNEC] e da Federação Internacional de Robótica [IFR], indicam que o crescimento do mercado dos robôs pessoais, incluindo aqueles usados para fins educativos e de entretenimento, tem sido enorme e prevê-se um aumento nas próximas décadas (Kara, 2004, citado por Benitti, 2012). A revisão de literatura revela que no ensino básico a robótica tem sido utilizada como ferramenta para o ensino e aprendizagem de outras disciplinas escolares, com especial ênfase em projetos interdisciplinares envolvendo Ciência, Matemática, Informática e Tecnologia. No ensino secundário e superior a robótica é mais utilizada como objeto de aprendizagem, incidindo sobre disciplinas de programação, inteligência artificial e mecatrónica (Mitnik, Nussbaum & Soto, 2008; Benitti, 2012).

Com base nos estudos acima referidos que apontam para a utilização da Robótica Pedagógica como um incremento no ensino da programação, planeou-se a presente

intervenção. O estudo incidiu sobre a problemática do ensino da programação onde se procurou aferir, de que forma a utilização de robótica:

- Proporciona a aprendizagem da programação.
- Evidencia no aluno a compreensão dos problemas.

A introdução de projetos que aliam os recursos tecnológicos ao processo aprendizagem de conteúdos é uma metodologia para a promoção do pensamento crítico/reflexivo e a construção do conhecimento (Valente, 2005). No presente trabalho foram utilizadas tecnologias, tais como o uso de ferramentas da Robótica Pedagógica, nas aulas de programação. Através dessa ferramenta e do trabalho elaborado pelos professores os alunos podem construir o seu conhecimento através de um processo de investigação e de desafios. A Robótica Pedagógica, segundo os especialistas Benitti *et al.* (2009), Cabral (2010), Gomes (2007) e Ribeiro (2006) é uma forma de viabilizar o conhecimento científico-tecnológico e, ao mesmo tempo, estimular a criatividade e a experimentação de forma lúdica, aproximando os estudantes da complexidade tecnológica aliada à aplicação de conceitos relacionados a conteúdos curriculares da educação básica.

A manipulação das ferramentas da Robótica Pedagógica pode ser considerada um jogo, a partir do momento em que os alunos são convidados a resolver situações predefinidas, que fazem com que cada um atinja um resultado final a partir de uma série de regras. Essa manipulação e utilização podem proporcionar uma forma alternativa de aprender, dado que os alunos, ao invés de resolverem uma lista de exercícios, passam a solucionar problemas de investigação que contribuem para a construção de uma posição crítica em relação a esses problemas. Acresce ainda que esta situação também favorece o planeamento e o raciocínio para a construção de estratégias que beneficiam a formação em geral.

No presente trabalho objetiva-se explorar, por meio da percepção dos estudantes, as possibilidades de ensino-aprendizagem de conteúdos de programação, utilizando-se como ferramenta o Arduíno. Esta metodologia visa compreender o envolvimento dos alunos com a Robótica Pedagógica na aprendizagem da programação. A escolha do Arduíno, teve por base o facto de esta ser uma das ferramentas da Robótica Pedagógica disponível na atualidade, e ainda devido à sua forma de organização e diversidade de recursos oferecidos.

Considera-se que seria relevante, também, observar tais atividades sob a ótica e segundo a caracterização de um desafio, com a verificação das reações e resultados que pudessem evidenciar efetivas contribuições para a formação de conceitos de programação.

Propõe-se a resolução de tarefas de programação no âmbito do ensino, com a utilização da robótica, proporcionando, assim novas experiências de aprendizagem contextualizadas nesta área. A análise do trabalho realizado pelos alunos permitirá identificar as suas dificuldades na utilização destas tecnologias e na compreensão dos conceitos de programação.

## 2.1 Ensino-Aprendizagem da Programação

A Programação é considerada uma disciplina base para grande parte dos cursos de Ciências da Computação e obrigatória em muitos cursos de Engenharia. Esta disciplina aborda os princípios da lógica de programação, com o objetivo de dotar os alunos de conhecimentos que permitam analisar e resolver problemas por meio de algoritmos. Aprender a programar não se restringe apenas a escrever linhas de código numa dada linguagem é uma ciência, por ser constituída por um conjunto de regras orientadoras, onde é necessário usar lógica e onde existem métodos rigorosos para que se assegure a eficiência, segurança e utilidade dos programas gerados (Costa, Aparício & Caldeira, 2014; Costa *et al.*, 2012b; Costa, Aparício & Pierce, 2009; Ferreira, Costa, Aparício & Aparício, 2017; Pereira, Costa & Aparício, 2017; Pierce & Aparício, 2010; Piteira & Costa, 2012; Piteira & Costa, 2013; Piteira & Costa, 2017; Piteira, Costa & Haddad, 2012). A programação exige competências como o pensamento crítico, a abstração e a capacidade de analisar e modelar problemas (Costa *et al.*, 2012). Para muitos estudantes o problema surge numa fase inicial da aprendizagem, quando têm que compreender e aplicar conceitos abstratos de programação. Por outro lado, os que conseguem compreender os conceitos de programação melhoram a capacidade de escolha da linguagem e aumenta a possibilidade de aprender novas, uma vez que as linguagens de programação estão em constante evolução, e é requerido ao programador uma atualização e aprendizagem constante. Uma vez adquirida a compreensão completa dos conceitos fundamentais da linguagem, torna-se muito mais fácil ver como esses conceitos são incorporados.

## 2.3 Problemática do Ensino-Aprendizagem da Programação

No que tange à problemática do ensino da programação, alguns estudos refletem sobre os problemas inerentes ao ensino-aprendizagem. Contudo, consideramos crucial o contributo do livro “Computer Science Education Research”, editado por Sally Fincher & Marian Petre, que identificam algumas dessas áreas e asseguram que, apesar de existir uma série de instrumentos que apresentam resultados positivos no ensino da programação, continuam a persistir alguns problemas entre outros (Costa, *et al.*, 2012; Gomes & Mendes, 2007; Murphy *et al.*, 2006), tais como:

- falta de tempo,
- resolução de problemas práticos,
- inclusão de novos desenvolvimentos e tecnologias,
- transferências do ensino presencial para o ensino à distância,
- falta de raciocínio lógico.

O estudo empírico em Portugal sobre a problemática do ensino da programação relaciona-se com a dificuldade dos alunos na apreensão dos conceitos abstratos, por vezes relacionando com dificuldades na própria matemática (Mendes & Gomes, 2007). Outros fatores são apontados nessa problemática, tais como a ferramenta ou até mesmo o software e a forma como é abordado.

Alguns desses estudos relacionam a falta de competências para resolver problemas de conhecimentos matemáticos e lógicos, com as dificuldades na programação (Gomes, *et al.*, 2006), outros incidem na própria sintaxe da linguagem e no nível abstrato do pseudocódigo (Miliszewska & Tan, 2007; Lahtinen *et al.*, 2006). Alguns autores apontam ainda métodos de ensino desadequados à aprendizagem da programação e a conotação negativa associada a esta disciplina (Gomes *et al.*, 2008). Grande parte dos estudos neste âmbito foram realizados em países estrangeiros. No entanto, com base nesses estudos, em Portugal já foram implementadas novas abordagens no âmbito do ensino da programação, nomeadamente através da utilização de programas mais interativos (Almeida, s/data).

## 2.4 Robótica Pedagógica.

O interesse na Robótica Pedagógica como instrumento de aprendizagem tem vindo a crescer, desempenhando um papel cada vez mais ativo na construção da mesma. O crescimento da oferta de *kits* utilizados na Robótica Pedagógica, desde o ensino pré-

escolar até ao secundário, certifica a vantagem na sua utilização, não só nas disciplinas curriculares, mas também como forma de desenvolver competências técnicas e científicas dos alunos (Ferreira, Veruggio, Micheli & Operto, 2010). A Robótica Pedagógica surge como um instrumento educativo com maior incidência no ensino universitário. No entanto, de acordo com Erwin, (2000, citado por Ferreira *et al.*, 2010) esta pode ser utilizada desde o jardim-de-infância até ao ensino universitário. Em Portugal este tipo de trabalho tem sido desenvolvido em todos os níveis de ensino desde 2009.

A Robótica Pedagógica pode ser analisada como sendo uma ferramenta que permite ao professor demonstrar na prática muitos dos conceitos teóricos, muitas vezes de difícil compreensão, motivando o aluno, que a todo momento é desafiado a observar, abstrair e inventar. É uma proposta educacional, apoiada na experimentação e no erro e que cria uma nova relação aluno/professor no qual eles caminham juntos, inovando, errando e aprendendo (Costa *et al.*, 2104). Com este modelo o aluno passa a construir o seu modelo de conhecimento através das suas ações e desperta maior interesse quando consegue superar o problema. O ambiente em que este modelo se insere é definido como um ambiente em que são utilizados recursos de hardware e de software livre, como o projeto Arduíno (2017) como forma de se viabilizar projetos educacionais.

O projeto Arduíno prevê uma plataforma de hardware e de software de código aberto, de fácil utilização e acessível a qualquer pessoa podendo ser constituído por:

- Computador
- Componentes eletrónicos e mecânicos
- Software

A sua introdução na prática em sala de aula, como uma ferramenta de apoio, tem vindo a demonstrar maior eficiência, nomeadamente numa aprendizagem baseada em problemas concretos. Para Johnson (2003), os robôs tornam-se mais que um simples brinquedo, visto poderem realizar tarefas mais interessantes do que simplesmente alguns movimentos, podendo até ter um certo nível de autonomia. A sua área de intervenção é multidisciplinar e a sua principal característica é de ser um ambiente em que os alunos podem montar, programar e analisar o comportamento de um sistema robotizado, promovendo a interação e a autonomia do aluno. Dada a sua grande flexibilidade, pode ser aplicada nas mais diversas áreas de conhecimento e, permite aos professores apresentar de forma mais lúdica e atrativa os conceitos anteriormente tidos como teóricos e de difícil compreensão. É exigido ao aluno a organização de tarefas e pensamentos desde o planeamento até á

montagem e programação. A cada passo é necessário que o aluno consiga superar para que consiga solucionar o problema elevando gradualmente a complexidade do pensamento e por seguinte o grau de atenção do aluno. Mill e Cesar (2013) entendem a Robótica Pedagógica como uma proposta pedagógica. Trata-se de denominação para o conjunto de processos e procedimentos envolvidos em propostas de ensino e aprendizagem, que tomam os dispositivos robóticos como tecnologia de mediação para a construção do conhecimento (Mill; Cesar, 2013).

A Robótica Pedagógica teve início com as investigações desenvolvidas por Seymour Papert em 1970, quando criou uma tartaruga com recurso à linguagem de programação Logo. A tartaruga criada por Papert (2008), era constituída por um robô de grandes dimensões que ocupava quase toda a sala e funcionava ligada a vários fios e cabos (Figura 1). Movimentava-se com comandos simples inseridos no computador, como por exemplo: PARAFRENTE 50. A partir desse programa, a tartaruga movimentava-se 50 passos para frente. A Figura 2 ilustra um exemplo dessa tartaruga traçando um caminho.



FIGURA 1 - UMA DAS PRIMEIRAS TARTARUGAS CRIADA POR SEYMOUR PAPERT (WALTER, 2010)

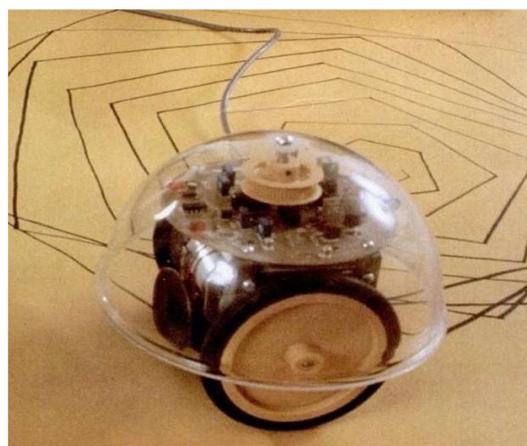


FIGURA 2 - UMA TARTARUGA DE ÁGUA DOCE (WALTER, 2010)

Com o desenvolvimento e aperfeiçoamento da programação em Logo, a grande tartaruga transformou-se num recurso gráfico que passou a movimentar-se no ecrã do computador. Papert (2008) procurava proporcionar aos alunos uma forma de comandar o computador com a programação, para que a tartaruga realizasse os movimentos. Com isso o equipamento passou a ser um instrumento que estimulava o pensar, o raciocínio para a

resolução de problemas, a consolidação de conceitos, favorecendo assim a construção do conhecimento.

Em meados da década de 1980, a parceria do pesquisador Papert com a empresa Lego, permitiu que os movimentos da tartaruga não se limitassem às abstrações no ecrã do computador. Esta trouxe para o mundo real as programações e construções, por meio de *kits* de robótica (Lego, 2017 a; Vimeo, 2017). O facto de os próprios alunos construírem o robô e a programação, segundo Papert (2008), pode favorecer o ensino, dado que a aprendizagem será facilitada se descobrirem por si mesmos. Este facto é denominado por Papert, como atitude construcionista. O construcionismo, conceito elaborado por esse autor, tem como base a possibilidade de construção do conhecimento pelo aluno. A forma construcionista de utilizar a tecnologia, tem como meta oferecer estratégias de estudo de forma a permitir uma maior aprendizagem possível, a partir do mínimo de ensino. “Evidentemente, não se pode atingir esse objetivo apenas reduzindo a quantidade do ensino, enquanto se deixa todo o resto inalterado” (Papert, 2008). Este investigador defende que a educação poderá ter um maior contributo para a formação dos alunos, se puder apoiá-los moral, psicológica, material e intelectualmente no processo de aprendizagem. Complementando a ideia de construcionismo, o autor relaciona a ideia de “conjuntos de peças para construção” com as ferramentas do Lego, e em seguida, alarga-as para as linguagens de programação, que podem ser consideradas conjuntos, a partir dos quais os programas podem ser elaborados.

A partir das primeiras manifestações sobre a construção de robôs como ferramenta auxiliar no ensino, pesquisadores tais como Benitti *et al.* (2009); Guedes e Kerber (2010); Mill e Cesar (2013) e Zanetti *et al.* (2012) têm-se dedicado a estudar os resultados da aplicação Robótica Pedagógica nas salas de aula e os seus possíveis benefícios para o processo de ensino-aprendizagem. Benitti *et al.* (2009) entendem a Robótica Pedagógica I como uma ferramenta que pode proporcionar ao aluno a criatividade, autonomia, o aperfeiçoamento do raciocínio, a capacidade de trabalhar em grupo visando o alcance de um objetivo comum. Complementando as ideias de Benitti *et al.*, (2012), Zanetti *et al.* (2012) consideram que a Robótica Pedagógica é um processo interativo que pode contribuir para que o aluno concilie o abstrato com o concreto. Para a construção de um robô com suas funções, o aluno passa pelos processos de conceção, implementação, construção e automação de um mecanismo, sendo essas atividades interdisciplinares que envolvem conhecimentos de várias áreas científicas.

A construção de robôs passa a contribuir para o processo de aprendizagem do aluno, dado que este aprende construindo, testando e pesquisando. Como observa Zanetti *et al.*, (2012), o objetivo que se deseja alcançar é, não apenas a construção de um robô em si, mas também a utilização da ferramenta que permite construir e manipular artefactos que oferecem experimentação e análise de conceitos teóricos, de maneira simples e objetiva. Além dessa possibilidade de construção, experimentação e análise, Mill e César (2013) destacam a aproximação que as aulas de robótica podem proporcionar entre professores e alunos, oferecendo possibilidade de trabalho cooperativo e solidário. Em consequência, o aluno sente-se mais participante, promovendo a autoestima e confiança nas suas capacidades.

Mill e César (2013) apresentam, resumidamente, as potencialidades que a introdução da robótica na educação pode proporcionar aos sujeitos envolvidos:

- a) Aprendizagem motivadora e divertida (potencializar as atividades lúdicas)
- b) Autonomia e responsabilidade pela aprendizagem
- c) Planeamento estratégico com base na aprendizagem
- d) Aprendizagem sociointeracionista
- e) Conceção de projetos de aprendizagem (domínio efetivo das tecnologias digitais)
- f) Aprendizagem digital e raciocínio abstrato
- g) Aprendizagem por tentativa-erro/ tentativa-acerto
- h) Aprendizagem pela pesquisa e pela multidisciplinaridade (Mill & César, 2013).

Atualmente os kits Lego (2017 b) possuem diversas peças e, a partir delas, os alunos são capazes de montar os seus robôs como se fossem um quebra-cabeças; as engrenagens e os motores permitem criar robôs que se podem movimentar sobre rodas, esteiras ou outros tipos de comandos; acionar uma alavanca ou movimentar um braço. Os sensores ajudam a “dar mais vida” aos robôs e a partir deles é possível detetar cores e a presença de outros objetos. Estes podem também ajudar no movimento, uma vez que a partir deles o robô pode “perceber” melhor o ambiente envolvente. A programação é construída com um programa específico, onde é possível controlar os sensores e a função de cada um dos motores do robô.

Concorda-se com o Mill & César (2013), quando caracterizam as atividades da Robótica Pedagógica como motivadoras, proporcionando ao aluno um ambiente propício para a descoberta, de forma lúdica e criativa. Em particular, os conteúdos da programação podem ser abordados pelas ferramentas da Robótica Pedagógica de forma que consigam

cativar os alunos, motivando-os e incentivando-os a produzirem de forma autônoma, artefactos e rotinas de programação dos robôs para resolver problemas, muitas vezes, de paciência, atenção concentrada e de dedicação (Costa *et al.*, 2012a; Costa *et al.*, 2012b; Costa *et al.*, 2014). Por meio dessas atividades os alunos podem compreender a programação não só como um conjunto de fórmulas e cálculos, mas também como conteúdos aplicados, inseridos em desafios que exigem mais do que cálculos para se encontrar uma resposta, podendo, na maioria das vezes, a resposta não ser única. Entretanto, não basta ter à disposição *kits* para que se promova a transformação da atividade de ensinar. Deve-se ter em conta os diversos dilemas da educação, que dificultam a introdução da robótica nas aulas. Desse modo não se pretende interpretar a Robótica Pedagógica como uma solução salvacionista que vai transformar a educação.

#### 2.4.1 Limitações e Desafios da Robótica Pedagógica

O trabalho de Benitti (2012) aborda três aspectos principais nas suas pesquisas sobre a Robótica Pedagógica:

1. Identificar a potencial contribuição da Robótica Pedagógica no ensino;
2. Apresentar uma síntese de avaliações empíricas do uso de Robótica Pedagógica;
3. Definir quais são as perspectivas futuras da Robótica Pedagógica.

Sobre as perspectivas de pesquisas futuras, através de sua análise literária, são colocadas três questões relacionadas com os desafios da utilização da Robótica Pedagógica:

1. Quais os temas aflorados através da Robótica Pedagógica nas escolas?
2. Como se processa a avaliação?
3. A Robótica Pedagógica é uma ferramenta de ensino? O que revelam estudos realizados?

Ainda há uma procura de estudos que possam implementar métodos de avaliação rigorosos e consistentes, para demonstrar a importância efetiva da Robótica Pedagógica como ferramenta de ensino.

Em relação aos métodos de avaliação, Benitti (2012) discute dois aspectos que podem ser verificados:

- Em que contexto são feitas avaliações - este aspecto questiona quais são os tipos de robôs utilizados, quais são os participantes, o tamanho da amostra e o contexto

educacional. Essa discussão sugere que a avaliação deve ter em consideração o público e os objetivos esperados. A aplicação da Robótica Pedagógica com crianças de diferentes níveis de ensino deve ter avaliações e objetivos diferentes de uma aplicação em estudantes de um curso superior.

- O método utilizado na experiência e na obtenção dos resultados - nesse aspecto, a discussão incide sobre a ausência de métodos relacionados com a elaboração e condução das atividades, as quais possam efetivamente utilizar a Robótica Pedagógica como uma ferramenta para auxiliar o ensino. Também ressalta que, através de um método formal, é possível criar meios para obter resultados de maneira mais fidedigna e coerente com a atividade proposta. Muitas vezes, práticas com recurso à Robótica Pedagógica são eventos extracurriculares ou complementares, não sendo o evento didático principal.

## 2.5 Aprendizagem por resolução de problemas (ABRP)

Aprendizagem por resolução de problemas (ABRP) é um modelo de ensino-aprendizagem que reconhece a necessidade de desenvolver competências de resolução de problemas e de ajudar os alunos na aquisição dos conhecimentos e competências essenciais. No processo de ABRP, o primeiro passo, numa sequência de passos inter-relacionados, é a apresentação do cenário (situações-problema) aos alunos, seguido de um brainstorming para promover o levantamento dos tópicos e questões emergentes associados ao tema apresentado e a procura de soluções através da promoção de atividades de investigação (Dahlgren & Oberg, 2001, Vasconcelos *et al.*, 2012, Torres, Preto & Vasconcelos, 2013). Este modelo recorre a problemas reais, não ao estudo de casos hipotéticos com resultados perfeitos e convergentes. É enfrentando esses problemas reais que os alunos aprendem conteúdos e desenvolvem competências de pensamento crítico (*critical thinking skills*). Nestes contextos, a aprendizagem resulta da participação ativa dos alunos no processo de aprendizagem, uma vez que os conduz à procura do conhecimento necessário à resolução dos problemas, sendo os próprios alunos a determinarem as suas necessidades de aprendizagem (Figura 3).

A ABRP apresenta algumas características enquanto modelo:

- Uma aprendizagem que é centrada no aluno

- Permite a aprendizagem de conteúdos uma vez que ao resolver problemas o aluno tenta ultrapassar as dificuldades que vão surgindo.
- Estimula a vontade de aprender, porque desenvolve no aluno o desejo pela procura do conhecimento, uma vez que é ele que tem de pesquisar e analisar para conseguir obter soluções;
- O trabalho é realizado em grupo

e baseia-se em quatro princípios:

1. Aprendizagem cumulativa - os conteúdos são introduzidos, sempre com um nível de complexidade crescente, de forma a contribuir para uma tomada de decisão devidamente fundamentada em relação à resolução de um problema;
2. Aprendizagem integrada - os conteúdos são introduzidos à medida que vão sendo necessários e que se relacionem com um determinado problema;
3. Progressão na aprendizagem - os vários aspetos do currículo devem sofrer alterações e prosperar paralelamente ao crescimento/desenvolvimento dos alunos;
4. Consistência na aprendizagem – os objetivos da ABRP definem-se tendo em conta os diversos aspetos do currículo, nomeadamente a avaliação a efetuar e os materiais didáticos a utilizar.

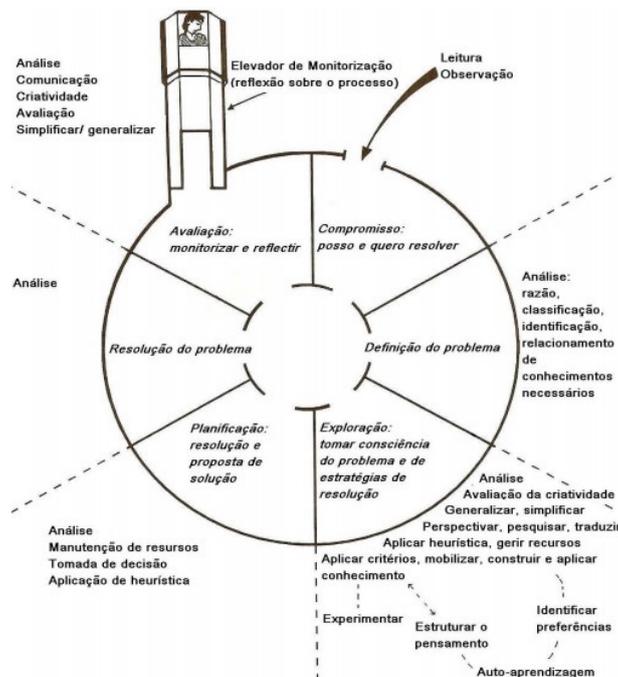


FIGURA 3 - ETAPAS DA ABRP E COMPETÊNCIAS (ADAPTADO DE WOODS 1994)

Em suma a ABRP é uma metodologia centrada no aluno, onde este desempenha um papel ativo e responsável na sua aprendizagem, fomenta a motivação, a autonomia, estimula o pensamento crítico, a criatividade e capacidades de análise e decisão, desperta a curiosidade em diferentes áreas do saber e desenvolve capacidades e competências para trabalhar em grupo.

## 2.6 Projeto Arduíno

O projeto Arduíno (2017), nascido na Itália em 2005, constitui uma plataforma de *hardware* e de *software* com o objetivo de possibilitar que indivíduos não especialistas em programação e/ou em eletrónica possam desenvolver aplicações de objetos e ambientes interativos. Como ferramenta, é habitualmente associado à filosofia de *Physical Computing*, ou seja, ao conceito que engloba a criação de sistemas físicos através do uso de *Software* e *Hardware* capazes de responder a *inputs* vindos do mundo real (Arduíno, 2017).

Para isso, a proposta do projeto visa, não só a criação de um *hardware* fácil de manusear e com os recursos necessários para trabalhar com os "mundos" digital e analógico, mas também de um *software* de desenvolvimento acessível para a programação dos projetos interativos.

O Arduíno (2017) pode ser designado simplesmente como uma peça de *Hardware* e/ou um *Software* de desenvolvimento, embora seja muito mais. Devido ao sucesso que tem vindo a alcançar ao longo do tempo, existe uma enorme comunidade de utilizadores/seguidores em todo o Mundo, podendo afirmar-se que o Arduíno representa também uma enorme comunidade. As razões para tal sucesso baseiam-se no seu baixo custo, - dadas as suas funcionalidades - a simplicidade na utilização e a possibilidade de utilização em vários sistemas operativos, como o *Windows*, *Macintosh OS* e *Linux* (Arduíno, 2017).

Uma vez programado, o Arduíno pode controlar uma gama de componentes eletrónicos como *led's*, motores, *displays*, (Figura 4).



FIGURA 4 - PLACA ELETRÔNICA ARDUÍNO UNO

Atualmente, com doze anos de vida, o Arduino é uma verdadeira "onda" mundial com aplicações em diversos segmentos (exemplo: música, artes plásticas, educação, meio ambiente) e com uma infinidade de comunidades espalhadas no planeta trocando experiências sobre os projetos.

## 2.6.1 Hardware

### 2.6.1.1 Microcontrolador

Um microcontrolador, é desenhado e construído de forma a integrar diversos componentes num único circuito integrado, evitando, assim, a necessidade de adicionar componentes externos ao microcontrolador, que permitirão implementar as suas funcionalidades.

Figura 5, apresenta alguns exemplos de componentes que se encontram disponíveis, conseguindo reunir uma grande quantidade de recursos num único chip integrado (Atmel, 2017).

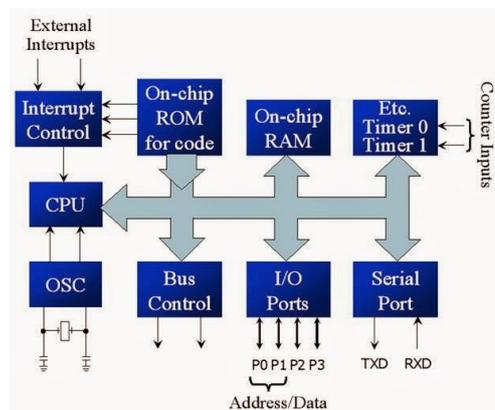


FIGURA 5 - COMPONENTES DO MICROCONTROLADOR

Na Figura 6, é apresentado um diagrama de blocos de um microcontrolador ATmega328, onde é possível identificar todos os seus constituintes.

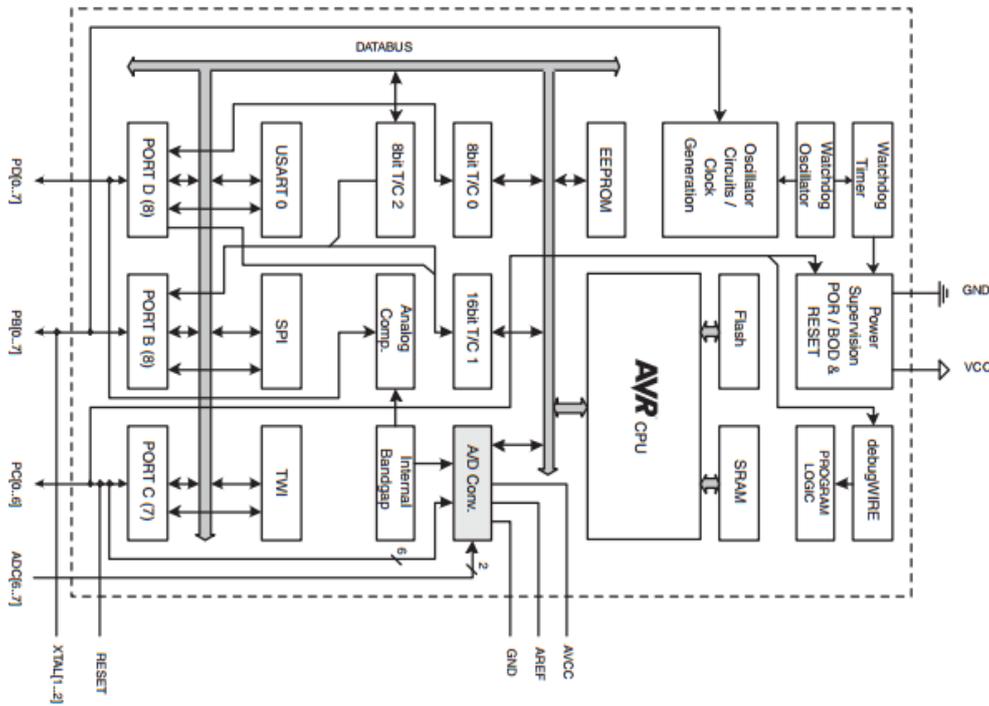


FIGURA 6 - BLOCOS DE UM MICROCONTROLADOR

### 2.6.1.2 Tipos e Quantidades de Memória Disponíveis

Uma das principais diferenças, em termos de capacidade, nos diferentes modelos disponíveis, reside na memória do microcontrolador utilizado. Essa diferença, representa um fator crucial no seu desempenho.

A Tabela 1 apresenta as quantidades de memória disponíveis nos diversos modelos de microcontroladores utilizados pelo Arduino (Atmel, 2011).

**Tabela 1 – Quantidade de memória disponível em cada modelo de microcontrolador** Fonte: Retirado do *datasheet* dos respectivos microcontroladores

	<b>ATMega168</b>	<b>ATMega328</b>	<b>ATMega1280</b>
<i>Flash</i>	16 KBytes (2 KBytes Bootloader)	32 KBytes (2 KBytes Bootloader)	128 KBytes (4 KBytes Bootloader)
<i>SRAM</i>	1024 Bytes	2048 Bytes	8192 Bytes
<i>EEPROM</i>	512 Bytes	1024 Bytes	4096 Bytes

Uma das memórias mais “importantes”, além da memória Flash que permite o armazenamento do *bootloader* (Arduino, 2017) e do programa – *sketch* - a ser executado,

é a memória SRAM (*Static Random Access Memory*), que se comporta de forma semelhante à RAM nos nossos computadores. É na SRAM que o programa, ao ser executado, cria e modifica todo o tipo de variáveis necessárias à sua execução. Este tipo de memória apenas mantém os dados enquanto se encontra alimentada, o mesmo não acontecendo com a memória EEPROM (*Electrically-Erasable Programmable Read-Only Memory*) e a memória Flash (Adafruit, 2016).

É ainda importante referir que a corrente máxima por cada pino analógico e digital é de 40 mA, à exceção da saída que providencia 3.3 V, que permite correntes máximas de 50 mA.

A capacidade de utilizar *Pulse Width Modulation (PWM)* (Arduíno, 2017)(Figura 7) é muito importante, uma vez que permite obter uma tensão analógica a partir de um sinal digital, ou seja, de um sinal que apenas pode assumir o estado lógico 0 (0V) ou 1 (5 V). O conceito de *PWM* é utilizado para referir um sinal que possua uma frequência constante e um *duty cycle* variável.

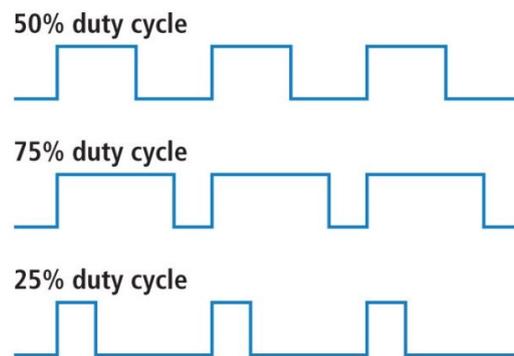


FIGURA 7 - SINAL COM *DUTY-CICLE* VARIÁVEL

Pela análise da Figura 7 pode-se concluir que a tensão média  $V_{dc}$  é diretamente proporcional ao *duty cycle* do sinal *PWM*. Esta característica permite-nos fazer variar a tensão, neste caso específico, entre 0 e 5 V (Arduíno UNO).

O Arduíno possui capacidade de operar alimentado, quer pela porta USB quer por uma entrada *Pwr* (do tipo *Power Jack*), sendo recomendada a sua alimentação (*Pwr*) entre os 7 e os 12V.

### 2.6.2 Software

A Figura 8 apresenta o IDE (*Integrated Development Environment*) do Arduino.

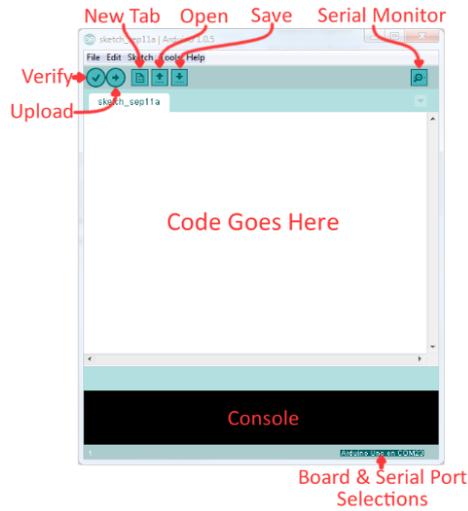


FIGURA 8 - IDE DO ARDUÍNO

### Ciclo de desenvolvimento

A Figura 9 apresenta um resumo dos passos necessários para elaborar uma aplicação (após ter feito a instalação do *Software*) de uma forma esquemática, a qual se pode designar por Ciclo de Desenvolvimento.

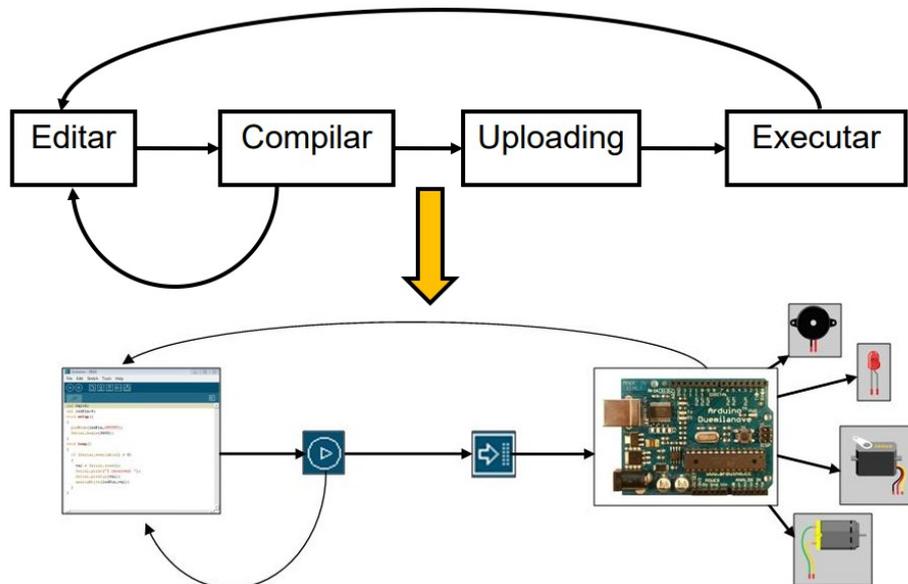


FIGURA 9 - CICLO DE DESENVOLVIMENTO

A análise do Ciclo de Desenvolvimento é possível fazer um resumo de todas as fases necessárias até à execução do programa criado, sendo muito importante a sua compreensão e apreensão.

### 2.7 Kit de led's e sensores

Para a implementação da atividade foi criado um *kit* de led's e sensores constituído por cinco led's, um sensor de luminosidade, um potenciómetro e um sensor de temperatura para que os alunos, através da programação do Arduino, pudessem manipulá-lo (Figura 10). Este *kit* estará ligado ao Arduino, o que permitira programá-lo de uma maneira rápida e com isso verificar de imediato, se o que foi programado estará a ser executado. (Anexo A).

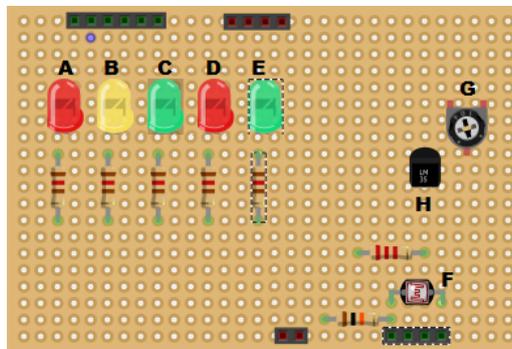


FIGURA 10 - KIT DE LED'S E SENSORES

### 2.8 Modelo de aceitação da tecnologia

Davis (1986) introduz o TAM (Technology Acceptance Model – Modelo de Aceitação da Tecnologia) com o objetivo de explicar a aceitação das tecnologias de informação. Baseou-se na Teoria de Ação Racional de Ajzen (1975), e centrou todo o seu modelo em torno de duas dimensões:

- a) Utilidade Percebida (PU) - grau em que uma pessoa acredita que o uso de um determinado sistema pode contribuir para melhorar o seu desempenho;
- b) Facilidade de Uso Percebida (PEOU) - grau em que um indivíduo acredita que o uso de um sistema de informação estará livre de esforço.

O principal objetivo da TAM é analisar o impacto das variáveis externas nas crenças internas, da atitude e na intenção do indivíduo (Davis, Bagozzi e Warshaw, 1989). É esperado que a utilidade e a facilidade de uso influenciem a atitude e a intenção de uso

do individuo. É esperado também uma ligação entre a facilidade de uso e a utilidade, propondo que a percepção da simplicidade ou dificuldade da utilização de um sistema, influencia a percepção de utilidade desse mesmo sistema (Vijayarathy, 2003).

Adam, Nelson e Todd (1992) reforçam essa perspectiva indicando que a facilidade de uso percebida tem um efeito direto e positivo na utilidade percebida e no uso do sistema. Um individuo irá usar um sistema se o considerar como conveniente e socialmente importante, apreciando o seu uso tecnológico (Saga e Zmud, 1994).

Segundo Davis (1989), as pessoas tendem a usar uma tecnologia com o objetivo de melhorar o seu desempenho – PU. No entanto, mesmo quando uma pessoa considera útil uma determinada tecnologia, a sua utilização poderá ser condicionada por algumas variáveis. Para Davis (1989), se o uso da tecnologia for bastante complexo, a pessoa tenderá a considerar que o esforço não justifica o uso – PEOU. Sendo assim, o TAM (Figura 11) está baseado fundamentalmente em duas dimensões: a PU e a PEOU, sendo que ambos são influenciados pelas variáveis externas, como características do sistema, processo de desenvolvimento, formação e intenção de uso (Davis, 1989). O objetivo deste modelo é representar o impacto de fatores externos relacionados ao sistema de informação, sobre os fatores internos do indivíduo, como as ‘atitudes’ e as ‘intenções de uso’ – BI (Davis, 1989; Davis *et al.*, 1989; Dillon & Morris, 1996; Venkatesh *et al.*, 2003).



FIGURA 11 –MODELO DE ACEITAÇÃO DE TECNOLOGIA -TAM

Fonte Davis et al., 1989, p. 985

O TAM é, assim, usado pelos investigadores de modo a fornecer explicações sobre o comportamento dos consumidores na utilização e adoção das tecnologias de informação. No TAM, as crenças dos consumidores determinam as atitudes de uso do sistema, e a intenção comportamental, por sua vez, é determinada por estas atitudes de uso do sistema.

## 3 – Trabalho Empírico

### 3.1 Introdução

No presente capítulo apresenta-se a metodologia selecionada para a elaboração do trabalho. Como já foi referido, o objetivo deste estudo prende-se com a perceção da forma como a Robótica Pedagógica poderá ser utilizada como ferramenta auxiliar na Programação. Como tal, inicialmente foi realizada uma revisão bibliográfica sobre o tema e o estudo em causa, através da lecionação de duas aulas e da aplicação de dois questionários. Este tipo de investigação privilegiou o recurso a material específico, que possibilitou a análise dos dados. Procedeu-se ainda à caracterização do contexto em que o trabalho se realizou, designadamente dos participantes no estudo em causa.

Quando se realiza uma investigação, utilizam-se formas de recolha de dados associados à natureza da metodologia utilizada. A metodologia qualitativa caracteriza-se por procurar a compreensão dos fenómenos no contexto em que estão inseridos e numa perspetiva indutiva, holística e ideográfica. É uma metodologia que estuda a realidade, sem a descontextualizar, a partir dos dados fornecidos pela realidade, enfatizando mais as particularidades e especificidades do que a procura de leis gerais.

As investigações podem ser de natureza exploratória, descritiva ou explicativa. A exploratória procura esclarecer os conceitos e ideias para melhorar a formulação dos problemas e das hipóteses de investigação de estudos posteriores. A descritiva tem por objetivo descrever as características da população ou esclarecer as relações existentes entre as variáveis. Finalmente a pesquisa explicativa, a mais complexa, procura isolar e explicar os fatores que estão na origem dos fenómenos (Bertuci, 2008).

De acordo com a técnica utilizada, as pesquisas podem ser documentais, de estudo de caso, ou de levantamento. A pesquisa documental é baseada na consulta de livros, artigos científicos, dissertações e outros documentos, sobre um determinado tema ou assunto. O estudo de caso compreende a exploração aprofundada de um número reduzido de objetos, sendo por vezes difícil a generalização das conclusões obtidas. No levantamento não se pretende o aprofundamento, estando este tipo de pesquisa orientada para a descrição dos aspetos gerais de um dado fenómeno ou situação (Bertuci, 2008).

Adotou-se o estudo de caso, para investigar a aplicação da Robótica no apoio à aprendizagem de programação, por parte de um grupo selecionado de alunos. Pretendeu-

se averiguar os eventuais benefícios do ensino e aprendizagem da programação, identificar os principais desafios, no âmbito das dificuldades de aprendizagem e do interesse e empenho pelas tarefas, e encontrar estratégias para os ultrapassar.

O estudo de caso é aplicado quando o investigador pretende avaliar, descrever o contexto, ou explicar relações entre intervenções dificilmente estudadas através de outras metodologias. A situação a estudar deverá estar bem delimitada, procurando retratar a realidade de forma aprofundada, recorrendo a uma grande variedade de fontes de informação e fazendo uma exaustiva descrição e interpretação do contexto do objeto de estudo. O estudo de caso é uma investigação empírica, baseado no raciocínio indutivo e em fontes diversificadas de dados (Coutinho & Chaves, 2002). Esta modalidade de investigação fornece informações relevantes para a tomada de decisão e é caracterizado por envolver uma descrição e análise detalhada, sistemática e aprofundada do objeto de estudo (Ponte, 1994).

### 3.2 Caracterização do estudo

O “Technology Acceptance Model (TAM)” – Modelo de Aceitação da Tecnologia sugere que quando os utilizadores são confrontados com uma nova tecnologia, uma série de fatores influenciam a sua decisão sobre como e quando a usar, nomeadamente: a ‘utilidade percebida’ (PU) – definida como "o grau em que uma pessoa acredita que utilizar um determinado sistema melhora o seu desempenho profissional" e ‘facilidade de uso percebida’ (PEOU) – definida como "o grau em que uma pessoa acredita que a utilização de um determinado sistema não implica qualquer esforço" (Davis, 1989).

A programação de computadores constitui um enorme desafio para professores e alunos. A tarefa de aprender e ensinar torna-se por isso uma tarefa difícil. Os motivos foram relatados em trabalhos anteriores que mencionam as principais dificuldades encontradas na programação. Têm-se apresentado várias soluções com o objetivo de minimizar essas dificuldades, e uma delas é a utilização da robótica, por esta torna-se atraente para os jovens, principalmente porque estes podem observar as reações do robô aos comandos dados pelos alunos. Neste contexto, é proposto uma tarefa de modo a oferecer aos alunos informações introdutórias, noções fundamentais de programação e experiência prática usando uma linguagem de programação. Aos alunos é disponibilizado um *kit* didático,

um Arduíno e um computador com um *IDE* instalado. A fim de ilustrar cada um dos itens do programa, vários desafios, correspondentes a um conjunto de exercícios foram desenvolvidos.

Os participantes do estudo são constituídos por alunos de três turmas do ensino universitário. Ao longo da execução do projeto de investigação houve, por parte do investigador, a preocupação de rigor, credibilidade das novas situações e descrição rigorosa de toda a investigação, analisando os dados obtidos através dos questionários e tentando que os pré-conceitos do investigador não influenciassem os resultados.

O estudo foi conduzido de modo a tentar responder à questão da investigação, a saber:

**Será que a utilização da Robótica Pedagógica contribui para a aprendizagem da programação?**

A validade deste estudo foi obtida internamente com a validação dos inquéritos propostos aos alunos. As conclusões apresentadas, correspondem à realidade reconhecida pelo investigador e pelos participantes. Partindo deste princípio, pode afirmar-se que existe validade na presente investigação, uma vez que o inquérito retrata a realidade, o que se pode confirmar através do cruzamento das respostas dos inquiridos no que respeita às capacidades que julgam adquiridas, com a literatura existente sobre esta temática.

### 3.3 Instrumentos

O TAM mostrou resultados coerentes, mantendo a sua eficácia na explicação da aceitação de tecnologia pelos utilizadores de sistemas de informação. Foi aplicado em diferentes tecnologias, bem como em diferentes situações (ao longo do tempo e culturas), com diferentes fatores de controlo (género, tipo e dimensão empresarial) e diferentes indivíduos (estudantes e profissionais), o que leva a crer na sua solidez e eficácia (Sá, 2006).

Contudo, posteriores ao modelo de Davis (1989), vários estudos foram desenvolvidos que tomaram como base o próprio TAM (Aparício & Costa, 2012; Costa & Aparício, 2013; Costa, Ferreira, Bento & Aparício, 2016; Pinheiro, Aparício & Costa, 2014; Sousa, Costa & Aparício, 2017). Estas novas extensões, vieram acrescentar novas dimensões de fatores ao modelo original de Davis, que provaram a sua influência sobre o comportamento de aceitação.

Uma nova tecnologia tem mais probabilidade de ser adotada quando é considerada útil e eficaz para se atingir um determinado objetivo. Estudos anteriores demonstraram que comparando os adultos jovens com os adultos mais velhos, os adultos jovens dão mais importância ao valor extrínseco da nova tecnologia e prestam mais atenção às vantagens associadas ao desempenho da tecnologia, enquanto os mais velhos dão mais importância à utilidade que irá ter a tecnologia (Hall e Khan, 2002).

No modelo proposto, considerou-se como base o modelo TAM (Davis, 1989) e procurou-se com apoio de outros modelos estudados evidenciar as variáveis externas consideradas pertinentes, tendo em especial atenção as características do Arduino. O principal objetivo deste modelo teórico é procurar explicar a relação entre os vários fatores que poderão influenciar a adoção do Arduino por parte do utilizador. Assim, foi introduzida dimensão, Enjoy. Na Figura 12 pode-se observar o modelo proposto que reflete as dimensões mencionadas, assim como as diferentes hipóteses sugeridas.

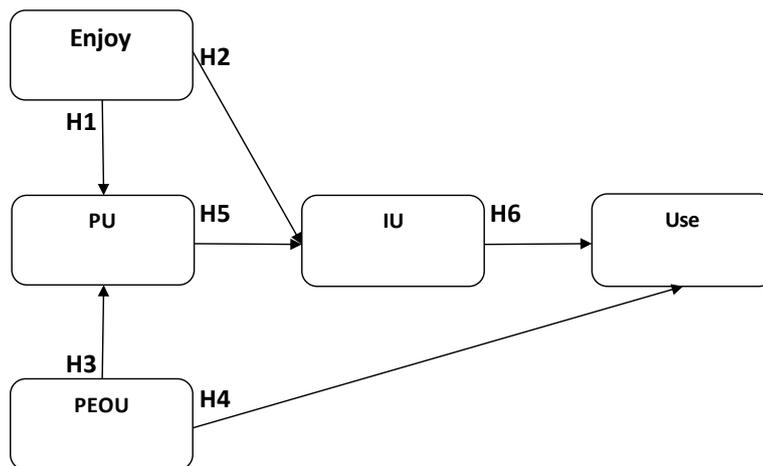


FIGURA 12 – MODELO DE INVESTIGAÇÃO DE ADOÇÃO DE ARDUÍNO NA APRENDIZAGEM DA PROGRAMAÇÃO

O modelo criado é assim composto por mais uma dimensão, para além dos já existentes no modelo de Davis (1989), que dão origem a seis hipóteses de trabalho. Com base nestas considerações, apresenta-se um modelo adaptado do TAM, considerando as seguintes hipóteses:

- ✓ H1 : O *Enjoy* influencia a perceção de utilidade do Arduino.
- ✓ H2 : O *Enjoy* influencia a intenção de uso do Arduino.
- ✓ H3 : A facilidade de uso percebida (PEOU) influencia a utilidade percebida (PU) no uso do Arduino.

- ✓ H4 : A facilidade de uso percebida (PEOU) influencia o uso do Arduino.
- ✓ H5 : A utilidade percebida (PU) influencia a intenção de uso do Arduino.
- ✓ H6 : A intenção de uso (IU) influencia a intenção de uso do Arduino.

O modelo de pesquisa foi validado através de métodos quantitativos usando uma escala previamente comprovada e testada. Considerando a revisão da literatura, foi selecionado um conjunto de itens para cada dimensão. O modelo escolhido utiliza uma escala de Likert de sete pontos (1- Discordo Totalmente, (...), 7- Concordo Totalmente).

A investigação incidiu apenas numa amostra de alunos de dois cursos superiores diferentes, pelo que não poderá ser generalizada para todos os estudantes.

É natural que se encontrem opiniões diversas por parte dos alunos ou mesmo contraditórias. Aliás, se as opiniões fossem indiscutíveis a ciência tornava-se desinteressante e imutável. Assim, assume-se que poderão existir dificuldades em manter o distanciamento necessário relativamente à problemática apresentada, podendo haver alguma influência involuntária nos resultados obtidos.

Embora se tenha previsto a recolha de dados online, por se considerar o método mais eficaz, esta não foi possível devido a dificuldades técnicas. Foi colocado um *link* no *e-learning* para cada inquérito e foi solicitado aos alunos o seu preenchimento. Para o grupo que teve dificuldades técnicas foi solicitado o preenchimento do inquérito em papel.

Na elaboração das questões, houve o cuidado de se utilizar uma linguagem clara e acessível, de forma a que, a cada questão estivesse associado um significado explícito.

O tipo de questões elaboradas partiu da informação reunida na revisão bibliográfica e das questões de investigação.

Optou-se por questões fechadas por apresentarem algumas vantagens em relação às abertas, nomeadamente por serem quantificáveis e de fácil resposta. Garante neste caso também uma maior fidelidade, uma vez que, todas as respostas estão sujeitas às mesmas opções, o que facilita a comparação dos resultados. Na elaboração do questionário, teve-se em conta a sua extensão, de modo a evitar cansaço e/ou a desmotivação aquando do seu preenchimento.

O primeiro questionário pretende fazer o levantamento de dados pertinentes ao estudo em causa. O segundo questionário foca essencialmente a utilização do Arduino como ferramenta de apoio.

## 4 - Proposta de intervenção

Na presente proposta didática, são apresentadas as orientações curriculares a serem seguidas na apresentação da atividade, designadamente o programa, a planificação, as tarefas e os recursos utilizados.

O programa proposto é relativamente exigente, uma vez que, para além dos tópicos de programação apresentados, inclui a aprendizagem de alguns conceitos de *hardware* utilizados no *kit* de *led's* e sensores. Juntamente com o programa, foram também elaborados a planificação e os planos de aula de todo o módulo. Finalmente apresenta-se ainda uma breve descrição das aulas do módulo, bem como um esboço geral da plataforma.

### 4.1 Programa

A gestão da atividade educativa deverá basear-se numa planificação de acordo com o nível de conhecimento dos alunos.

No que se refere às finalidades da atividade, são propostas as seguintes:

- Utilizar a Robótica como ferramenta de aprendizagem;
- Promover o interesse pela programação usando robótica educativa;
- Fomentar o trabalho experimental.

### 4.2 Planificação da aula

As atividades realizadas do decurso de uma aula com duração de cento e vinte minutos, tiveram a seguinte sequência:

- Apresentação do professor à turma e do motivo da sua presença na aula.
- Apresentação do desafio.
- Apresentação dos conceitos de programação necessários para a elaboração do desafio.
- Elaboração pelos dos alunos da 1ª parte do desafio (só para o 1º grupo de alunos).
- Início da 2ª parte do desafio.
- Teste com o *kit* de *led's* e sensores.
- Apresentação da 3ª parte do desafio.

- Teste com o *kit de led's* e sensores.
- Preenchimento dos questionários

### 4.3 Descrição da aula

O programa de aula foi realizado no contexto de uma aula de um curso de Informática no Instituto Universitário de Lisboa (ISCTE-IUL) em 2017, junto de alunos na fase de início de programação, cumprindo-se a planificação (Anexo C), deu-se início à aula com a apresentação da atividade (Figura 13).

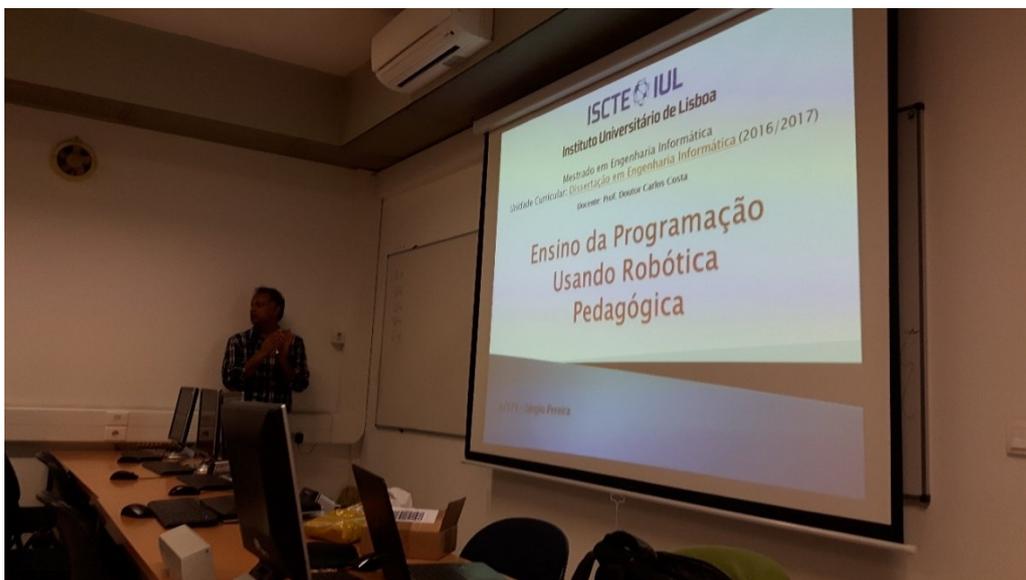


FIGURA 13 – APRESENTAÇÃO DA AULA

Posteriormente os alunos foram questionados, acerca dos seus conhecimentos relativos à temática em causa. Seguidamente efetuou-se uma apresentação, sobre o Arduino e a sua programação (Anexo F). Foi então solicitado a realização da resolução da primeira parte do desafio, tendo-se constatado que a maior parte dos alunos não realizou a tarefa proposta e seguiu para a parte dois do desafio.

Iniciou-se então o acompanhamento do desafio dois (Figura 14), tendo-se verificado que, com uma pequena orientação, os alunos conseguiram criar o programa necessário para “superar” o desafio. No entanto, quando se colocava o programa a correr no Arduino com o *kit de led's* e sensores, o resultado não era o esperado, mas com algumas correções quase todos os grupos conseguiram alcançar o objetivo pretendido. Notou-se ainda que os alunos sentiram alguma dificuldade na sintaxe

da linguagem utilizada, o que era expectável atendendo ao pouco tempo que tiveram para explorar os comandos disponíveis.



FIGURA 14 - ACOMPANHAMENTO DOS TRABALHOS

Relativamente à interação com o *kit* de *led's* e sensores, notou-se, por parte dos alunos um grande envolvimento, uma vez que ao poderem programar diretamente cada um dos *led's* (incluídos no *kit* de *led's* e sensores) conseguiriam de imediato observar o resultado (Figura 15).

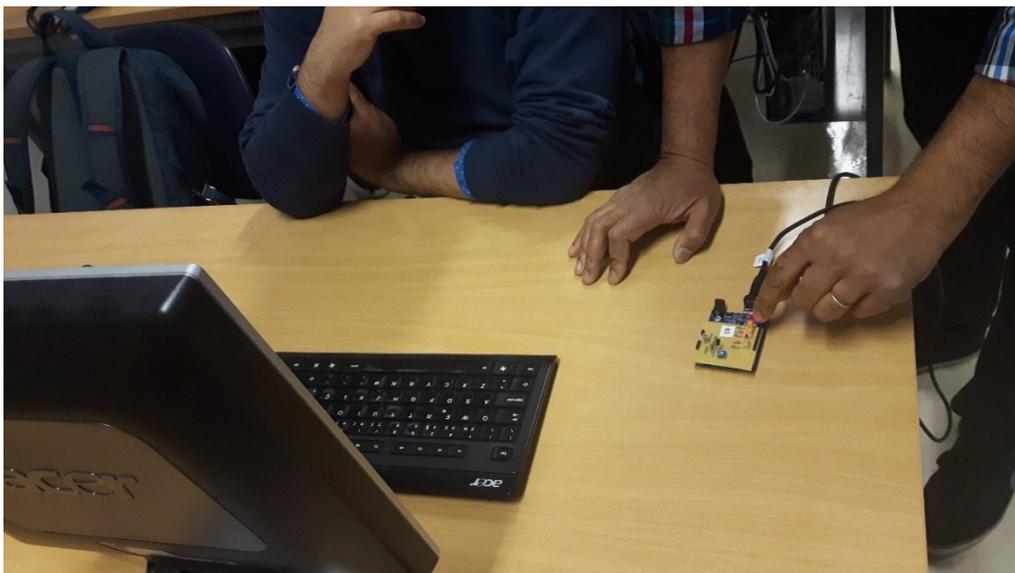


FIGURA 15 – INTERAÇÃO COM O KIT

A cada aluno/grupo que apresentasse a resolução da segunda parte do desafio concluída, era solicitado a resolução da terceira parte. Constatou-se uma grande dificuldade por parte dos alunos em conseguir superar esta tarefa, tendo sido necessário demonstrar-lhes a forma de ler o sensor de luz para poderem completar o desafio. Após esta explicação, os alunos retomaram a tarefa tendo todos conseguido terminá-la com sucesso. Apesar da programação ser individual, alguns alunos reuniram-se em grupo para discutir as ideias, apresentar sugestões e testar diferentes situações.

No decurso do trabalho foram sendo colocadas diversas questões, tendo-se verificado que os alunos expunham concretamente as suas dificuldades. Notou-se ainda que, quando os alunos terminaram o desafio e dispunham de algum tempo disponível, continuaram espontaneamente a explorar a programação do Arduino.

#### 4.4 Desafio

O desafio consistiu em simular um semáforo utilizando o *kit* de *led's* e sensores.

O kit era composto por:

- Dois *led's* vermelhos
- Dois *led's* verdes

- Um *led* amarelo
- Um sensor de luminosidade
- Um sensor de temperatura
- Um potenciômetro

#### Parte Um

Elaboração de um fluxograma / organograma / pseudo-linguagem em blocos, que pudesse avaliar o fluxo de informação do programa a ser executado.

#### Parte Dois

Programação do Arduino de modo a simular um semáforo (os tempos entre as mudanças dos *led's* foram definidos pelos alunos)

#### Parte Três

Utilização do sensor de luz que se encontra no *kit*, para simular as seguintes situações:

- Se não houver luz (sensor de luz tapado) os *led's* vermelhos e verdes deverão estar desligados e o led amarelo deve estar a piscar (o intervalo de tempo que o led pisca é atribuído pelo aluno)
- Se houver luz (sensor de luz descoberto) os *led's* simularão o semáforo (Parte dois)

## 5 - Resultados

Neste capítulo é feita a apresentação e a análise dos resultados obtidos. Através do questionário 1 apresentados aos alunos (Anexo G), nos quais foi utilizada a escala *Likert* para compor as opções de respostas de escolha múltipla, quer através das respostas do questionário 2 (Anexo G), relativamente à utilização do Arduino.

A amostra é composta por 43 alunos, sendo 37 homens e 6 mulheres, de 3 turmas do ensino universitário. Duas delas pertencentes ao 1º Ano do curso de Engenharia Eletrónica e uma terceira pertencente á pós-graduação em Engenharia Informática, no ano letivo de 2016/2017.

A realização de um exercício em papel (tarefa 1) para testar o funcionamento do fluxo de execução dos desafios, apresentou resultados muito negativos, uma vez que só dois alunos conseguiram concretizar, o que indica que a maioria dos alunos teve grande dificuldade em equacionar o problema apresentado Anexo D.

Relativamente ao questionário 1, estas estão organizadas por normas percetuais, cognitivas, avaliativas e denotativas. Esta análise baseia-se principalmente em respostas negativas, uma vez que estas apontam dificuldades e problemas encontrados pelos alunos. É possível perceber de maneira clara, os aspetos onde ocorreram maiores dificuldades, definindo assim, as informações e apontamentos que podem ser melhorados em práticas com robótica educativa.

As respostas negativas apresentadas estão relacionadas com dificuldades presentes na aplicação prática, assim como pontos a serem melhorados e corrigidos para o modelo.

Em relação às normas percetuais (questões 2 e 3), os alunos não apresentaram desconforto na identificação dos objetivos da prática (Figura 16) e também no conceito de programação, embora havendo a necessidade de melhorar a apresentação (Figura 17).

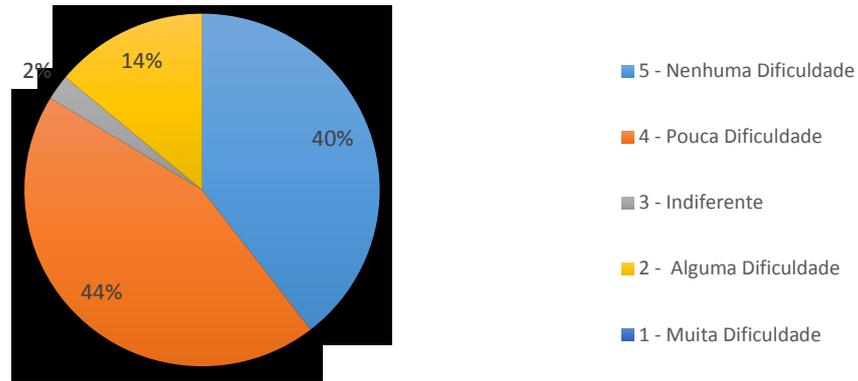


FIGURA 16 - OBJETIVO DA TAREFA

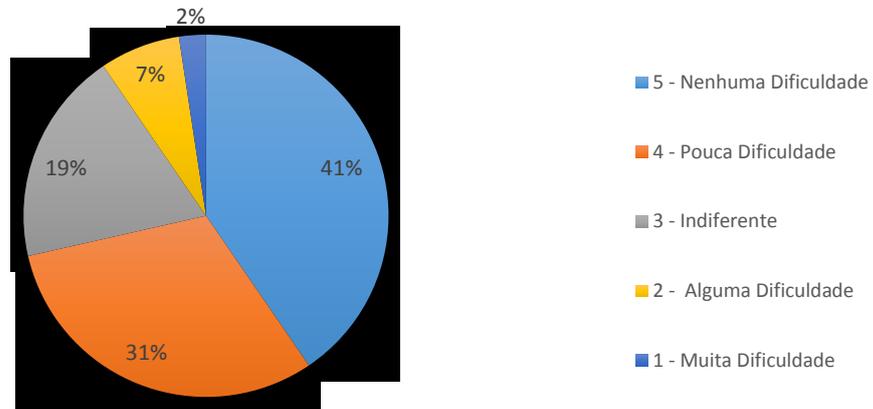


FIGURA 17 - COMPREENSÃO DOS CONCEITOS DE PROGRAMAÇÃO

Em relação às normas cognitivas, quando questionados sobre a compreensão da tarefa (i.e., o que devia fazer durante a prática) na questão 1 (Figura 18), foi possível identificar problemas relacionados com a descrição da tarefa.

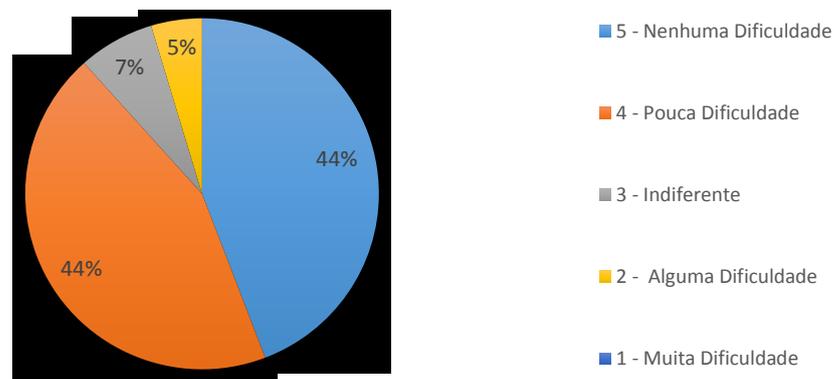


FIGURA 18 - DIFICULDADES NA COMPREENSÃO DA TAREFA PROPOSTA

Na questão 4, relacionada com as normas avaliativas, os alunos evidenciaram algumas lacunas. A questão refere as dificuldades de identificar o *feedback* do *kit de led's* e sensores em resposta ao que era programado (Figura 19), ou seja, se o mesmo respondia de acordo com a programação elaborada. Houve momentos de incertezas e a necessidade de uma vivência maior com a plataforma, para que esse *feedback* fosse compreendido da maneira mais adequada.

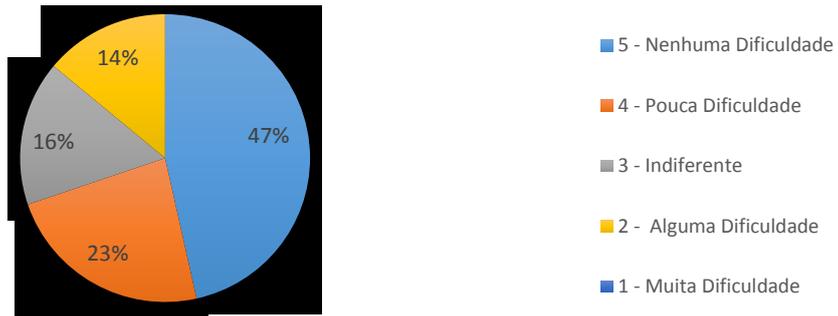


FIGURA 19 - DIFICULDADES EM COMPREENDER AS RESPOSTAS (*FEEDBACK*) DAS AÇÕES PROGRAMADAS

As respostas à questão 5 (Figura 20) foram quase todas positivas e afirmavam que a prática ajudou na concretização de novos conceitos de Programação.

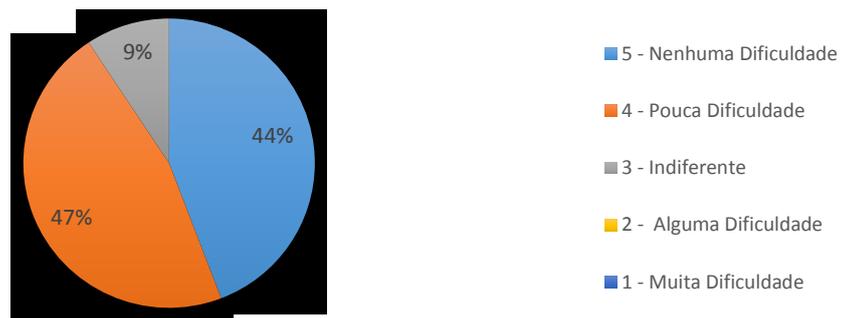


FIGURA 20 - A ATIVIDADE PRÁTICA CONTRIBUIU PARA UMA MELHOR COMPREENSÃO DE NOVOS CONCEITOS DE PROGRAMAÇÃO

Enquanto alguns alunos mostraram dificuldades em compreender o que deveriam realizar durante a tarefa e em compreender a aplicação dos conceitos de programação, outros não demonstraram quaisquer dificuldades a esse respeito.

Na questão 3 as respostas apontam uma melhoria na percepção dos alunos relativamente à aplicação dos conceitos e efeitos no *kit* de *led's* e sensores. Apenas um número reduzido de alunos apresentou resposta de caráter negativo.

A análise das respostas apresenta-se, de maneira geral, positiva em todas as questões, com exceção de algumas críticas e apontamentos negativos, principalmente relacionados com problemas no *kit* de *led's* e sensores. Toda a prática utilizando Robótica Pedagógica deve pressupor a ocorrência de incoerências em ações no *kit* de *led's* e sensores, já que este sofre a interferência de elementos externos à programação (e.g. variação da luminosidade do sensor de luz). Algumas das respostas negativas provêm desses problemas, os quais não foram tidos em conta na atividade. Também se deve considerar a experiência dos alunos e, conseqüentemente, a sua capacidade crítica.

Também na questão 3, relacionada com os aspetos cognitivos, houve apenas um número reduzido de respostas negativas, associados a alguma dificuldade do aluno em aplicar os conceitos de programação no *kit* de *led's* e sensores.

Frequentemente, algumas respostas apresentadas pelos alunos não possuem um detalhe suficiente que possibilite a identificação do problema ou das suas dificuldades, de uma maneira precisa. No entanto, mesmo com pequeno pormenor nas respostas, é possível notar um cenário predominantemente positivo nos alunos, mostrando as potencialidades de um processo formal para o desenvolvimento de práticas de programação com recurso à Robótica Pedagógica.

A análise acima referida ainda pode ser revista e melhorada, através de métodos que possam torná-la mais precisa na avaliação, em relação aos aspetos semânticos, cognitivos e comportamentais dos alunos, melhorando os resultados apresentados nessa seção.

As questões abaixo encontram-se dispostas na ordem que foram apresentadas no questionário:

1. A Robótica Pedagógica é um instrumento interessante para ensino de programação?
2. A utilização do “hardware” ocorreu sem problemas ou dificuldades?
3. Os recursos (*led's*, sensores, etc.) foram percebidos de maneira clara?
4. As respostas dadas pelo “hardware” à programação executada foram claras e coerentes?

5. A utilização da Robótica Pedagógica auxiliou na aprendizagem ou aumentou seu conhecimento nos conceitos de programação?
6. A utilização do “IDE” ocorreu sem problemas?
7. Houve dificuldades em aplicar os conceitos de Lógica de Programação?
8. Houve problema em compreender os objetivos da prática proposta?
9. Teve dificuldade em elaborar a solução da prática proposta?
10. Houve dificuldades em aplicar os conceitos de Lógica de Programação (estruturas de decisão, estruturas de repetição etc.) na solução?

A análise dos resultados demonstra que a Robótica Pedagógica pode ser utilizada e, conseqüentemente, a sua utilização para o ensino da programação é viável e construtiva.

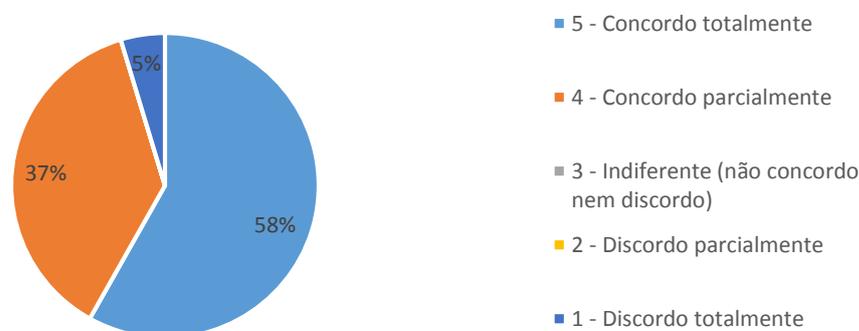


FIGURA 21 - A ROBÓTICA PEDAGÓGICA É UM INSTRUMENTO INTERESSANTE PARA ENSINO DE PROGRAMAÇÃO

Questões 2 e 4:

- “A utilização do “hardware” ocorreu sem problemas ou dificuldades (Figura 22)?”
- “As respostas dadas pelo “hardware” à programação executada foram claras e coerentes (Figura 23)?”

Estas questões estão diretamente relacionadas com aspectos tecnológicos particulares do *kit de led's* e sensores, como, por exemplo, comportamentos inesperados oriundos do funcionamento de sensores e dificuldades na comunicação física.

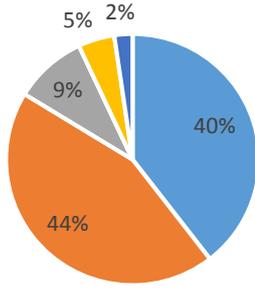


FIGURA 22 - UTILIZAÇÃO DO “HARDWARE”

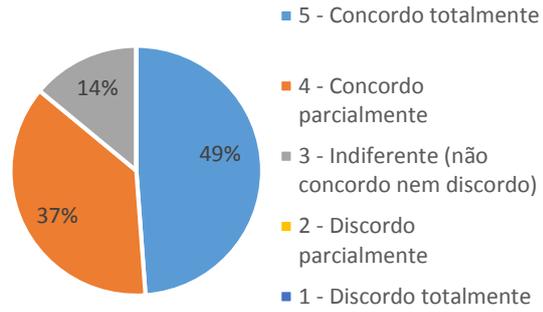


FIGURA 23 - AS RESPOSTAS DADAS PELO “HARDWARE”

Questão 3:

- “Os recursos (*led's*, sensores, etc.) foram percebidos de maneira clara (Figura 24)?”

Esta questão relaciona-se com a percepção dos alunos sobre a interpretação e o funcionamento do *kit*.

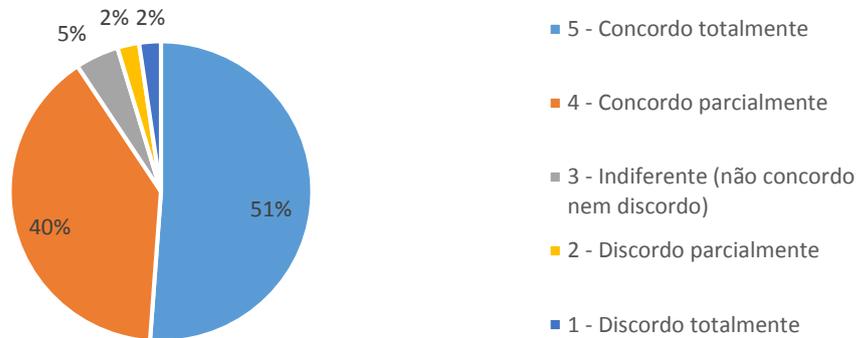


FIGURA 24 - OS RECURSOS (*LED'S*, SENSORES, ETC.) FORAM PERCEBIDOS DE MANEIRA CLARA

Questão 6:

- “A utilização do “*IDE*” ocorreu sem problemas (Figura 25)?”

Esta questão está relacionada com a utilização do *IDE* como interface de programação do Arduino. É de notar que a maioria dos alunos não teve dificuldade na sua utilização.

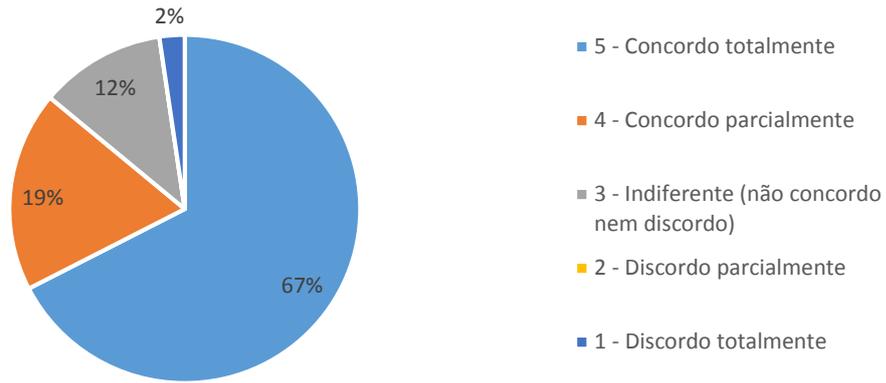


FIGURA 25 - A UTILIZAÇÃO DO “IDE” OCORREU SEM PROBLEMAS

Questões 8, 9 e 10:

- “Houve problema em compreender os objetivos da prática proposta (Figura 26)?”
- “Teve dificuldade em elaborar a solução da prática proposta (Figura 27)?”
- “Houve dificuldades em aplicar os conceitos de Lógica de Programação (estruturas de decisão, estruturas de repetição etc.) na solução (Figura 28)?”

As questões acima referidas estão diretamente relacionadas com a compreensão do objetivo da atividade e desenvolvimento da mesma. Nota-se que quase a totalidade dos alunos conseguiu chegar à solução solicitada.

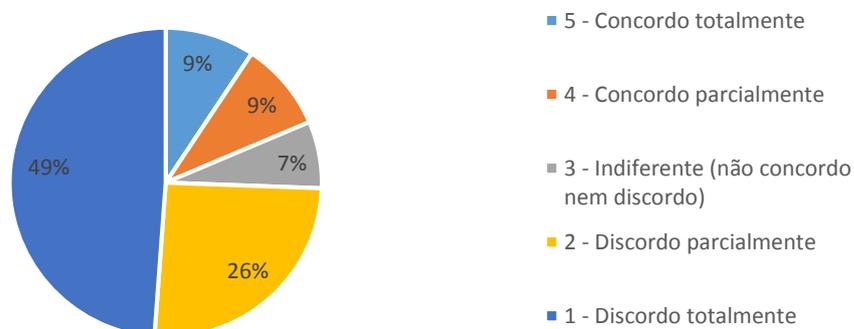


FIGURA 26 - HOUVE PROBLEMA EM COMPREENDER OS OBJETIVOS DA PRÁTICA PROPOSTA

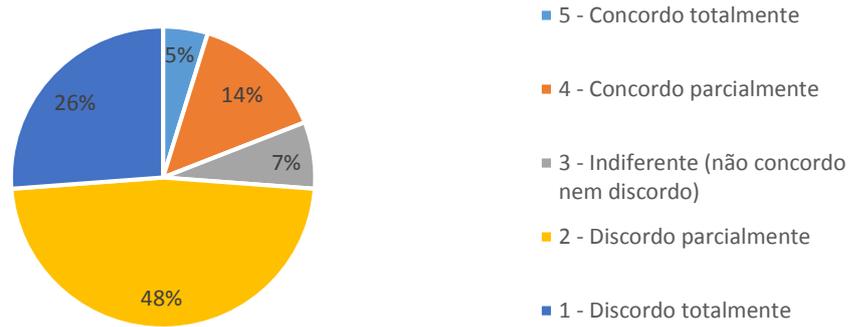


FIGURA 27 - TEVE DIFICULDADE EM ELABORAR A SOLUÇÃO DA PRÁTICA PROPOSTA

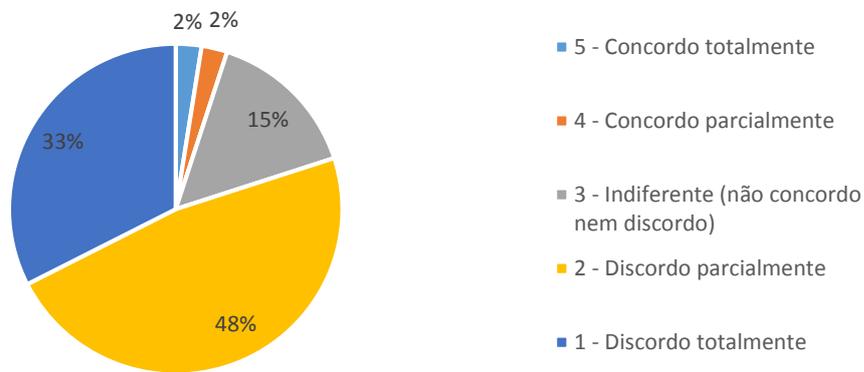


FIGURA 28 - HOVE DIFICULDADES EM APLICAR OS CONCEITOS DE LÓGICA DE PROGRAMAÇÃO (EXEMPLO: ESTRUTURAS DE DECISÃO, ESTRUTURAS DE REPETIÇÃO) NA SOLUÇÃO

Pela análise da Figura 29, que avalia a atividade realizada, pode-se verificar que esta teve bastante aceitação pela maioria dos alunos.

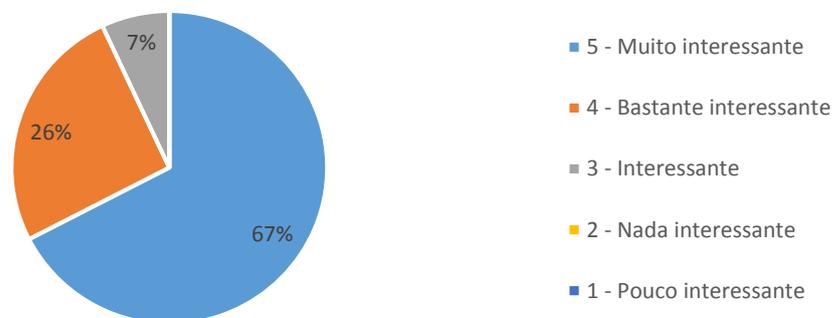


FIGURA 29 - AVALIAÇÃO DA ATIVIDADE

Em relação ao questionário 2 (Anexo G), este aborda fundamentalmente a utilização do Arduino, e a forma como esta ferramenta pode ser útil na utilização da programação. Para tal utilizou-se o modelo TAM (Technology Acceptance Model) de modo a auxiliar no entendimento do comportamento e intenção de uso desta tecnologia (Figura 30).

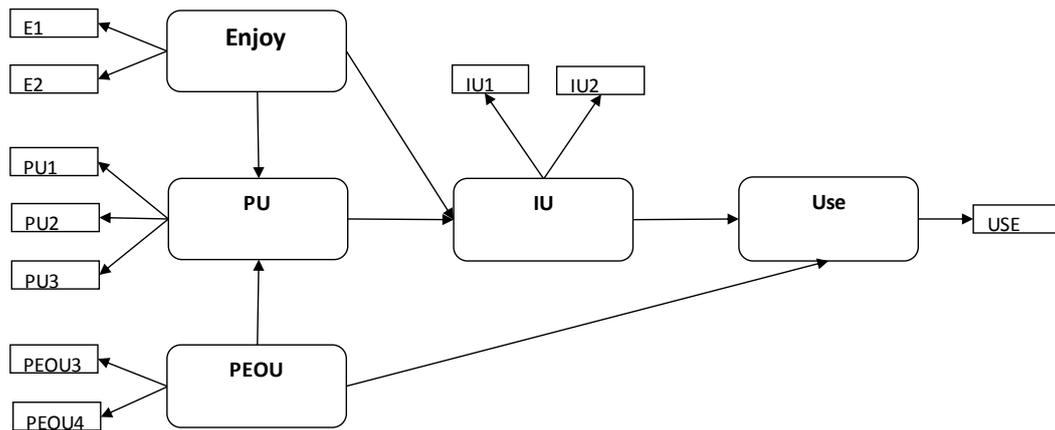


FIGURA 30 – MODELO DE INVESTIGAÇÃO

Na avaliação de fiabilidade dos resultados obtidos (Tabela 2), referente aos Critérios de Qualidade, as variáveis latentes apresentam um Alpha de Cronbach superior a 0.78, apresentando assim uma Fiabilidade de Consistência Interna (Hair *et al.*, 2014). A Variância Média Extraída (AVE), tendo em conta os “loadings” superiores a 0.817, permitiram assim analisar a Validade Convergente. Os valores AVE apresentam-se superiores a 0.5. Pode-se considerar a possibilidade do Alfa de Cronbach como sendo uma medida conservadora da Fiabilidade Consistência Interna (Hair *et al.*, 2014).

TABELA 2 – CRITÉRIOS DE QUALIDADE

	AVE	Composite Reliability	R <sup>2</sup>	Cronbachs Alpha	Communality	Redundancy
Enjoy	0.841841	0.914022		0.819461	0.841841	
IU	0.887418	0.940348	0.438144	0.873611	0.887418	0.255249
PEOU	0.820770	0.901563		0.781675	0.820770	
PU	0.817375	0.930655	0.509805	0.888092	0.817375	0.078269
Use	1.000000	1.000000	0.535327	1.000000	1.000000	0.224205

Com a validação do modelo estrutural, procedeu-se à avaliação dos caminhos estruturais para testar as hipóteses apresentadas neste estudo (Tabela 3).

TABELA 3 – RESULTADO DAS HIPÓTESES

Hipótese	Variável Independente		Variável Dependente	Beta	T-Value	Significância pValue f <sup>2</sup>
H1	Enjoy	->	IU	0.414630	3,2940	0,0010
H2	Enjoy	->	PU	0.179756	1,7879	0,0405
H3	PEOU	->	PU	0.643105	7,5987	0,0000
H4	PEOU	->	Use	0.557783	5,8956	0,0000
H5	PU	->	IU	0.388605	2,7621	0,0042
H6	IU	->	Use	0.285487	2,3674	0,0113

Nas várias correlações entre as variáveis (Tabela 4), verifica-se que existem correlações elevadas, como são o caso das correlações entre a Utilidade Percebida e a facilidade de uso percebida (0.693), entre o uso e a Facilidade de Uso Percebido (0.686) e entre o Uso Atual do Sistema e o Enjoy (0.611). A correlação mais baixa verifica-se entre a facilidade de uso percebida e o Enjoy (0.277).

TABELA 4 –CORRELAÇÃO ENTRE VARIÁVEIS

	Enjoy	IU	PEOU	PU
Enjoy	1.000000			
IU	0.553564	1.000000		
PEOU	0.276420	0.448073	1.000000	
PU	0.357522	0.536844	0.692793	1.000000
Use	0.061139	0.535415	0.685702	0.693030

O Alpha de Cronbach foi calculado com o objetivo de verificar as consistências internas existentes nas dimensões e aferir se estas são consistentes. Os valores de Alpha de Cronbach podem variar entre 0 e 1, sendo que um valor inferior a 0.6 é considerado inaceitável e valores acima de 0.9 considerados excelentes.

TABELA 5 - ALPHA DE CRONBACH DAS ESCALAS

Variável	Alfa de Cronbach
Enjoy	0.819461
IU	0.873611
PEOU	0.781675
PU	0.888092
Use	1.000000

As escalas das dimensões Utilidade Percebida, Intenção Comportamental, e Enjoy (Tabela 5) obtiveram valores bons (> 0.819) enquanto a dimensão Facilidade de Uso Percebido apenas por uma margem mínima não atinge esse valor, tendo obtido o valor de 0.782 apresentando assim uma boa fiabilidade.

A análise AVE (Variância média extraída) utiliza-se para verificar e analisar a validade discriminante das diferentes escalas, sendo que apenas valores superiores a 0.50 são considerados positivos (Chin, 2010). Se os valores se encontrarem acima de 0,50 significa que a variância explicada por esses indicadores excede a variância explicada pelo erro (Barroso, Carrion e Roldan, 2010)

TABELA 6 - AVE, CONFIABILIDADE COMPÓSITA, COEFICIENTE DE DETERMINAÇÃO

	AVE	Composite Reliability	R <sup>2</sup>
Enjoy	0.841841	0.914022	
IU	0.887418	0.940348	0.438144
PEOU	0.820770	0.901563	
PU	0.817375	0.930655	0.509805
Use	1.000000	1.000000	0.535327

A confiabilidade compósita das escalas serve para quantificar a consistência interna das dimensões, sendo que apenas valores superiores a 0.70 são considerados aceitáveis. Como se verifica na tabela anterior todos os valores estão acima de 0,90. Verifica-se que à semelhança dos valores da AVE, a Intenção Uso tem os valores mais elevados na confiabilidade compósita das escalas (0.940). Como se pode verificar na tabela anterior o Use apresenta os valores mais elevados (53%) sendo assim a variável que mais contribui para a utilização do Arduíno. A Intenção de Uso é a variável com menor valor (43%) sendo por isso aquela que menos contribui.

O R<sup>2</sup> (coeficiente de determinação) é uma medida de precisão explicativa da regressão, ou seja, permite medir as relações de causa e efeito entre duas ou mais variáveis dependentes e/ou um conjunto de variáveis independentes.

A avaliação da validade dos constructos permite medir todas as variáveis latentes. A validade dos constructos dá-se apenas quando a validade discriminante e a validade convergente são estabelecidas. Não obstante Chin e Newsted (1999) afirmam que para a validade convergente ser satisfatória terá de ter um valor superior a 0.5. Por outro lado, Fornell e Larker (1981) afirmam que apenas valores superiores a 0.70 asseguram a fiabilidade dos itens. Comrey e Lee (1992) consideram que valores acima de 0.71 devem ser considerados excelentes.

TABELA 7 – VALIDADE DOS CONSTRUCTOS

	Enjoy	IU	PEOU	PU	Use
E1	<b>0,883</b>	0,4267	0,088	0,195	-0,084
E2	<b>0,951</b>	0,567	0,366	0,419	0,149
IU1	0,516	<b>0,949</b>	0,427	0,586	0,525
IU2	0,529	<b>0,935</b>	0,417	0,415	0,482
PEOU3	0,194	0,325	<b>0,903</b>	0,629	0,602
PEOU4	0,305	0,484	<b>0,909</b>	0,627	0,640
PU1	0,388	0,510	0,565	<b>0,924</b>	0,625
PU2	0,232	0,483	0,619	<b>0,910</b>	0,668
PU3	0,346	0,463	0,689	<b>0,877</b>	0,588
USE	0,061	0,535	0,686	0,693	<b>1,000</b>

De acordo com a Tabela 7, verifica-se que todas as variáveis estão mais correlacionadas com as suas variáveis latentes da sua própria dimensão do que com as outras dimensões, validando-se assim as escalas utilizadas. Todos os valores estão acima de 0.7 o que mostra que os indicadores estão correlacionados com a sua própria dimensão.

A análise Bootstrap efectuada no PLS avalia a significância de cada coeficiente de trajetória e deste modo testa as hipóteses de investigação. Os valores são significativos a 90% quando são superiores a 1.64, são significativos a 95% quando superiores a 1.96, são significativos a 99% quando superiores a 2.57 e são significativos a 99.9% quando superiores a 3.29 (Chin e Newsted, 1999).

TABELA 8 – ANÁLISE TESTE T

	Enjoy	IU	PEOU	PU	Use
Enjoy		3.293984		1.787930	
IU					2.367440
PEOU				7.598668	5.895631
PU		2.762118			

Pela análise da Tabela 8 podemos afirmar que todas as relações são significativas a valores superiores a 90%.

Análise das hipóteses de investigação:

- ✓ H1 : O *Enjoy* influencia a perceção de utilidade do Arduino. ( $\beta=0,415$ ; **significativo para  $p<0,001$** ) - **Confirmado**
- ✓ H2 : O *Enjoy* influencia a intenção de uso do Arduino. ( $\beta=0,180$ ; **significativo para  $p<0,050$** ) - **Confirmado**
- ✓ H3 : A facilidade de uso percebida (PEOU) influencia a utilidade percebida (PU) no uso do Arduino. ( $\beta=0,643$ ; **significativo para  $p<0,001$** ) - **Confirmado**
- ✓ H4 : A facilidade de uso percebida (PEOU) influencia o uso do Arduino. ( $\beta=0,558$ ; **significativo para  $p<0,001$** ) - **Confirmado**
- ✓ H5 : A utilidade percebida (PU) influencia a intenção de uso do Arduino. ( $\beta=0,389$ ; **significativo para  $p<0,010$** ) - **Confirmado**

- ✓ H6 : A intenção de uso (IU) influencia a intenção dos utilizadores de continuarem a usar o Arduino. ( $\beta=0,285$ ; significativo para  $p<0,050$ ) – **Confirmado**

Apresenta-se na Figura 31 alguns valores do modelo adotado.

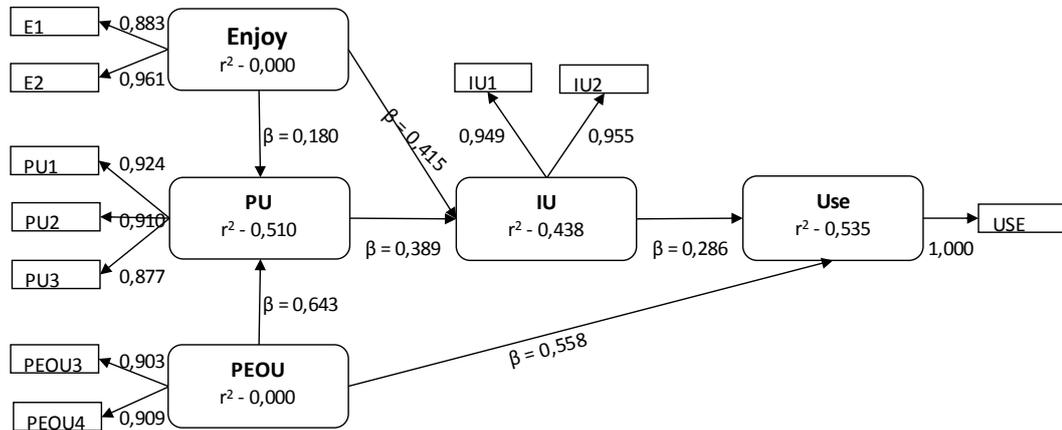


FIGURA 31 – RESULTADOS DO MODELO DE ADOÇÃO DE ARDUÍNO EM CONTEXTO DE APRENDIZAGEM DE PROGRAMAÇÃO

## 6 - Discussão

A análise dos resultados do presente trabalho evidencia que o método utilizado teve um impacto positivo nos alunos. Para a generalidade dos participantes o modelo apresentou boas perspectivas para a aprendizagem da programação dado que tornou mais motivante e atrativa. O método utilizado abrange o desenvolvimento e avaliação de práticas, através de um meio sistemático e formal, utilizando artefactos e bases teóricas que envolvem Robótica Pedagógica Eletrónica e Programação. Esta abordagem possui um carácter pouco explorado, tendo em vista pesquisas em trabalhos relacionados com a Robótica Pedagógica aplicada no ensino e aprendizagem da programação.

Pode ser uma alternativa interessante como proposta pedagógica para o ensino-aprendizagem da programação, uma vez que vem de encontro ao interesse dos alunos, tornando-o responsável pela sua aprendizagem e desenvolvimento. A Robótica é claramente uma área multidisciplinar, envolvendo um conjunto de disciplinas como a Informática ou a Eletrónica. Desta forma, a Robótica reúne todas as condições para proporcionar um conjunto de atividades interdisciplinares que promovam uma aprendizagem transversal dos diversos temas. As atividades envolvendo Robótica caracterizam-se por proporcionar uma quantidade quase infindável de problemas para os alunos resolverem, que têm como característica o serem inesperados, por vezes mesmo para o professor que coordena a atividade. O facto de estes problemas emergirem do mundo real e serem até por vezes impossíveis de resolver confere-lhes uma dimensão completamente distinta dos simples exercícios de sala de aula. Em quase todas as atividades de Robótica os alunos trabalham em equipas de forma colaborativa com um objetivo comum.

Analisando as questões do inquérito 1, é possível verificar que a predominância de respostas positivas é facilmente notada ao longo das questões 1 e 6. Entretanto, também é perceptível verificar em que questões ocorreram a resposta “indiferente” e onde ocorreram as respostas negativas. A partir da análise a (Figura 32), alguns aspetos podem ser evidenciados:

Para as perguntas diretas (1 a 6) as respostas situam-se entre o concordo totalmente e concordo parcialmente e para as perguntas indiretas (7 a 10) situam-se entre discordo parcialmente e discordo totalmente.

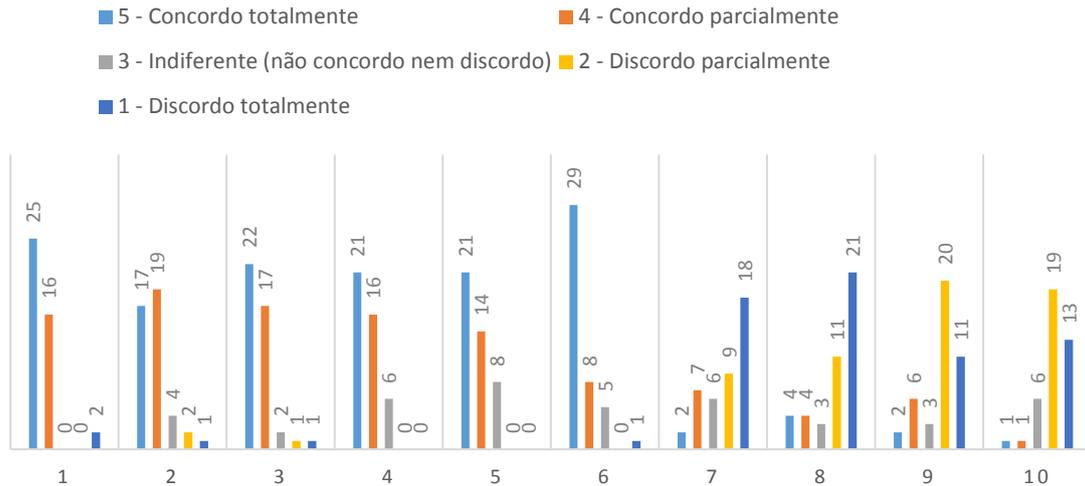


FIGURA 32 - QUESTÕES VS OCORRÊNCIAS

É possível notar que a utilização da Robótica Pedagógica é relevante para o ensino e aprendizagem da programação, como mostra a (Figura 21), quando se percebe que a maioria das respostas se situa entre 58% de “concordo totalmente” e 37% “concordo parcialmente”. Não há quase discordância na utilização da Robótica Pedagógica, quando se verifica que as respostas exclusivamente negativas de “discordo parcialmente” e “discordo totalmente” são apenas de 5%.

Em relação ao questionário 2 verificou-se que este estudo pretendeu medir a intenção comportamental dos alunos para a utilização do Arduino no ensino da programação ao avaliar de que forma o Modelo de Aceitação da Tecnologia permite compreender quais são as variáveis mais importantes na tomada de decisão para utilizar o Arduino como auxílio na programação.

Neste estudo foi utilizado o modelo proposto por Davis (1986) mas com algumas adaptações de modo a avaliar a intenção dos alunos na adoção do Arduino no ensino da programação. Os constructos são a Utilidade Percebida e a Facilidade de Uso Percebida e o Enjoy, a intenção é a Intenção Comportamental, e a resposta cognitiva é o Uso Atual do Sistema.

A confiabilidade das escalas foi verificada através do Alpha de Cronbach, onde se observou que a Facilidade de Uso Percebida (0,780), a Utilidade Percebida (0.889), a Intenção Comportamental (0.784) e o Enjoy (0.820) tinham uma excelente consistência

interna. A variância média (AVE) apresenta valores elevados e a confiabilidade composta das escalas confirma a consistência interna de todas as escalas.

O Enjoy, um fator humano relativamente novo neste campo de investigação, raramente foi examinado no que concerne ao seu efeito sobre o comportamento de utilização do Arduino. O Enjoy é um determinante importante para constatar se a interação com as interfaces tecnológicas irá ou não levar à utilização do Arduino.

Relativamente às hipóteses de estudo verifica-se que todas as 6 hipóteses formuladas foram confirmadas.

A primeira hipótese coloca em estudo a relação do Enjoy e a Utilidade Percebida, sendo que uma nova tecnologia tem maior probabilidade de ser adotada quando é considerada útil e eficaz para se atingir um objetivo.

A segunda hipótese coloca a relação do Enjoy com a Intenção de Usar o Arduino. Sendo um modelo com grande implementação poderá ser divertido o seu uso.

A terceira hipótese coloca a relação da facilidade de uso percebida e a utilidade percebida. É esperado que a facilidade de uso exerça uma influência direta e positiva sobre a utilidade percebida. Deste modo, e para confirmar o que estudos anteriores encontraram (Davis, 1986; Bagozzi e Warshaw, 1989; Park, 2010; Park, Lee e Cheong, 2007; Park et al., 2009, Venkatesh, 1999), a análise das significâncias das relações confirmou um fortíssimo nível de significância (superior a 90,0%).

A quarta hipótese assumia uma relação entre a facilidade de uso percebida relativamente ao uso do Arduino. Como já foi referido, e segundo Davis (1989) a facilidade de uso percebida é o grau pelo qual o individuo acredita que o uso do sistema é livre de esforço físico e mental. Deste modo, é esperado que a experiência tenha um efeito positivo. Através da análise das significâncias das relações verifica-se que a relação entre a experiência e a facilidade de uso percebida é significativa (superior a 90.0%) confirmando a hipótese.

A quinta hipótese assumia uma relação entre a Utilidade Percebida e a Intenção Comportamental relativamente ao uso do Arduino, sendo esperado que a Utilidade Percebida influencie positivamente a Intenção Comportamental. Diversos estudos (Agarwal e Prasad, 1999; Chau e Hu, 2002; Davis et al, 1989; Hu, Chau e Sheng, 1999; Igarria et al., 1999; Mathieson, 1991; Moon e Kim, 2001; Ramayah, Jantan e Ma'ruf,

2002; Venkatesk e Davis, 2000; Davis, 1989) confirmaram esta relação e demonstraram que quanto maior é a Utilidade Percebida maior é a Intenção Comportamental.

A última hipótese do modelo assumia uma relação entre a Intenção Comportamental e o Uso Atual do Sistema, demonstrando que quanto maior é a Intenção Comportamental maior é o Uso Atual do Sistema. Orbell, Hodgkins e Sheeran (1997) afirmaram que os indivíduos que demonstrem ou implementem intenções sobre um determinado ato têm maior probabilidade de realizar esse ato. Esta dissertação encontrou provas que confirmam a relação positiva e direta entre as variáveis, como se pode observar nos resultados obtidos. Na análise das significâncias das relações verificou-se um nível de significância superior a 90,0% (2.36).

## 7 – Conclusões, Contribuições e Trabalhos Futuros

Apresenta-se, em seguida, a conclusão desta dissertação, contendo as contribuições da pesquisa, trabalhos futuros e considerações finais. No início destaca-se as principais conclusões e de seguida as contribuições da pesquisa relativas às questões que delinearão todo o trabalho e os avanços resultantes deste estudo. Seguidamente é apresentada uma discussão sobre os trabalhos futuros, destacando as limitações da pesquisa e do método proposto, assim como a procura de novas pesquisas. Finalmente são apresentadas as considerações finais, com discussão de carácter pessoal e crítico-analítico.

### 7.1 - Conclusões

No presente estudo, começamos com a percepção de que a robótica pode ser uma ferramenta útil na abordagem do problema da aprendizagem de programação de computadores. De acordo com resultados preliminares, a atitude em relação à robótica é positiva. Os alunos demonstraram boa atitude em relação à robótica, pois a robótica é uma boa maneira de instanciar a programação de forma mais tangível. Com a resolução de exercícios mostraram os resultados de codificação imediatamente, os alunos conseguiram aprender de forma mais intuitiva os principais aspetos da programação.

A partir dos modelos analisados desenvolveu-se o nosso próprio modelo de aceitação da tecnologia, incorporando elementos que se consideram pertinentes tendo em atenção as características do Arduino. O modelo criado foi assim composto por mais um constructo, para além dos já existentes no modelo de Davis (1989) e que deram origem a seis (6) hipóteses de trabalho. O constructo acrescentado foi o Enjoy. Desta forma, além das influências existentes no modelo original, foram analisadas as respetivas influências que este constructo poderia eventualmente ter sobre a utilidade percebida do Arduino e sobre a facilidade de uso percebida.

Quanto às hipóteses de investigação formuladas, verificou-se através das análises estatísticas, que todas foram aceites para a amostra em estudo. Tendo em conta os objetivos propostos, pode-se concluir que os fatores presentes no modelo de aceitação da tecnologia, como sendo a utilidade percebida, a facilidade de uso percebida, o Enjoy, a intenção de uso e o uso efetivo, têm influência na utilização do Arduino.

- ✓ quanto maior a percepção de facilidade de uso do Arduino, maior será a percepção da utilidade do mesmo;
- ✓ quanto maior a percepção de facilidade de uso do Arduino, mais positiva será a atitude dos utilizadores em relação ao mesmo;
- ✓ quanto maior a percepção de utilidade do Arduino, maior será a intenção do seu uso;
- ✓ quanto maior a percepção de utilidade do Arduino, mais positiva será a atitude dos utilizadores em relação ao mesmo;
- ✓ quanto mais positiva for a atitude do utilizador em relação ao Arduino, maior será a sua intenção do seu uso;
- ✓ quanto maior a intenção de uso do Arduino, maior será o seu uso.

Finalmente, pode-se concluir que o objetivo geral desta investigação foi alcançado, tendo sido possível obter informações acerca de variáveis que influenciam a utilização do Arduino, por parte do aluno. Da mesma forma, concluí-se que foi dada resposta à questão de investigação definida para este trabalho, tendo sido verificado que a **“a utilização da Robótica Pedagógica contribui para a aprendizagem da programação”**, na perspetiva do aluno, é influenciada por variáveis que têm por base o modelo de aceitação da tecnologia.

## 7.2 - Contribuições da Pesquisa

Com o objetivo de responder á questão de pesquisa apresentadas no início dessa dissertação:

- Será que a utilização da robótica educativa contribui para a aprendizagem da programação?

O presente trabalho incidiu num estudo capaz de analisar as potencialidades da Robótica Pedagógica como ferramenta pedagógica, com a aplicação efetiva de um método num ambiente de ensino da Programação, tendo-se realizado a articulação entre ambos. Realizou-se a avaliação da atividade (i.e., objetivos gerais e específicos) e das práticas dinamizadas. Foi possível apresentar uma solução, que produz resultados concretos sobre as questões relacionadas com esse desafio.

Dos principais contributos deste trabalho, destacam-se os seguintes:

- Desenvolvimento de um método sistemático capaz de orientar a implementação e avaliação de práticas utilizando a Programação, por meio de uma proposta inovadora, utilizando técnicas provenientes da Robótica Pedagógica;
- Metodologia que pode ser aplicada através da visão de professor-pesquisador ou de um professor-aplicador. O método permite várias soluções, que podem ser exploradas e adaptadas em cenários específicos de ensino-aprendizagem da Programação, de acordo com o público-alvo. Para o professor-aplicador, apresenta uma proposta que encadeia uma série de passos para a formação, aplicação e avaliação de práticas utilizando a Robótica Pedagógica;
- Resultados do estudo do impacto da Robótica Pedagógica no ensino da Programação e a síntese desses resultados, tanto no caráter qualitativo como no quantitativo.

Os resultados obtidos através destas experiências mostraram que a Robótica Pedagógica tem potencial para o desenvolvimento da Programação. As limitações da aplicação deste processo, prendem-se com a utilização de artefactos para a realização prática.

### 7.3 - Trabalhos Futuros e Limitações

Entre as principais limitações da metodologia utilizada, pode-se citar as dificuldades em tratar as situações onde ocorrem respostas inesperadas ou inconsistentes em ações do *kit* de *led's* e sensores, sem que a programação fosse responsável por tais erros. Estudos futuros poderão agregar métodos de intervir em casos em que ocorram essas situações adversas, para que não haja a rutura no processo de aplicação ou que não interfira nos resultados esperados.

Outra limitação apresentada durante a pesquisa foi a dispersão obtida nos resultados das avaliações. Torna-se, assim, necessário realizar um reajustamento no desenvolvimento de formas de avaliação, as quais consigam demonstrar de forma mais clara as percepções dos alunos, de modo a identificar com maior precisão falhas no processo. Como exemplo, refere-se que, nas respostas dadas pelos alunos no questionário para a avaliação da atividade (questões fechadas), não é possível definir as causas das insatisfações ou dificuldades sentidas por eles.

Como próximo passo deste trabalho, propõe-se a realização de experiências com grupos maiores e mais heterogêneos em relação aos níveis de conhecimentos em Programação. Também se sugere a implementação das tarefas num público diferente para proporcionar uma nova visão relativamente à área em estudo.

Como em qualquer estudo, também este está limitado pelo tempo e pela disponibilidade para a sua realização.

A investigação incidiu apenas numa amostra de alunos de dois cursos superiores diferentes, pelo que não poderá ser generalizada para todos os estudantes.

É natural que se encontrem opiniões diversas por parte dos alunos ou mesmo contraditórias. Aliás, se as opiniões fossem indiscutíveis a ciência tornava-se desinteressante e imutável. Assim, assume-se que poderão existir dificuldades em manter o distanciamento necessário relativamente à problemática apresentada, podendo haver alguma influência involuntária nos resultados obtidos.

A Robótica Pedagógica para o ensino de programação, é um recurso pedagógico de grande potencial, tanto em questões ligadas à motivação dos alunos, quanto na diversidade de ferramentas e suporte que podem ser oferecidos aos professores. No entanto, a sua aplicação em sala de aula não é uma tarefa fácil, devendo ser acompanhada por métodos que possam conduzir à aprendizagem efetiva e, simultaneamente, fornecer instrumentos de avaliação eficazes.

Na literatura encontram-se diversos trabalhos que comprovam os benefícios educacionais e motivacionais que a Robótica Pedagógica prevê, embora ainda sejam escassos estudos de investigação que mostrem a sua eficiência de maneira sistemática e concreta, num cenário do ensino da programação.

Com os resultados obtidos a partir desta experiência, foi possível visualizar aspetos positivos na adoção da Robótica Pedagógica, assim como pontos a serem melhorados. Também ressaltaram aspetos relacionados com a Robótica Pedagógica que o método não abrange, não colabora ou é limitado. Esta pesquisa teve um caráter exploratório, tendo-se recorrido a uma amostra pequena para a experiência. No entanto, salienta-se, que permitiu a definição do problema e a formulação de uma proposta de solução, através de uma investigação sobre as possíveis dificuldades dos alunos em programação, aquando da utilização da Robótica Pedagógica.

O recurso a temas do quotidiano nas atividades dinamizadas com os alunos, facilitou a elaboração de hipóteses, a investigação de soluções através de testes no *kit* de *led's* e

sensores, e ao estabelecimento de relações com os seus conhecimentos prévios, conduzindo-os a conclusões acerca dos resultados obtidos.

Também a Direção-Geral de Educação (2017) está a promover neste ano letivo 2017/18, com o apoio da Universidade de Évora, do Instituto Politécnico de Setúbal, da Associação Nacional dos Professores de Informática e da Microsoft Portugal, a iniciativa “Programação e Robótica no Ensino Básico”. Esta iniciativa decorre da implementação do projeto-piloto Iniciação à Programação no 1.º Ciclo do Ensino Básico que, entre 2015 e 2017, envolveu mais de setenta mil alunos. Neste momento um novo ciclo de trabalho foi iniciado, onde se procura alargar as atividades desenvolvidas no âmbito da programação e robótica ao 2º e 3º ciclo do ensino básico.

Por fim, numa análise final e pessoal, estou convicto da importância de um trabalho colaborativo nas áreas que o mesmo abrange (Robótica Pedagógica e Programação). O modo como foi aplicado pode servir de incentivo para futuros trabalhos que possam explorar, de uma forma mais aprofundada, o problema principal que se propõe a resolver, o ensino da programação por meio da Robótica Pedagógica. Em relação aos estudos sobre programação utilizando a Robótica Pedagógica, este trabalho forneceu uma aplicação num cenário pouco explorado que é a Educação, podendo servir de incentivo à criação de outros métodos através de técnicas provenientes da Robótica Pedagógica, que possam ter um contributo relevante nessa área de estudo.

Para finalizar, saliento a importância da temática abordada na presente dissertação para a minha formação pessoal. No final do estudo, foi-me possível ter uma visão mais abrangente e clara da construção do conhecimento por parte dos alunos e da importância da inovação e da atualização num ambiente de ensino.

## Referências

- Adafruit (2016) Adafruit Learning Systems, Retrieved from: <https://cdn-learn.adafruit.com/downloads/pdf/memories-of-an-arduino.pdf> acessado a 24-01-2017
- Albuquerque, D., Bremgartner, V., Lima, Hillermann., Salgado, N. (2016). Uma experiência do Uso Do Hardware Livre Arduino no Ensino De Programação De Computadores. V Congresso Brasileiro de Informática na Educação (CBIE 2016), Anais do XXII Workshop de Informática na Escola (WIE 2016), 51-60.
- Alimisis, D. (2012). Robotics in Education & Education in Robotics: Shifting Focus from Technology to Pedagogy. 3 rd International Conference on Robotics in Education, RiE 2012, Prague, 7-14.
- Alimisis, D. (2013). Educational robotics: Open questions and new challenges. Themes in Science & Technology Education, 6(1), 63-71.
- Alves, R. M., Silva, A. L. C., Pinto, M. C., Sampaio, F. F., Elia, M. F. (2012). Uso do Hardware Livre Arduino em Ambientes de Ensino-Aprendizagem. Jornada de Atualização em Informática na Educação, JAIE 2012 162-187.
- Aparicio, M., & Costa, C. J. (2012, October). Collaborative systems: characteristics and features. In *Proceedings of the 30th ACM international conference on Design of communication* (pp. 141-146). ACM.
- Arduino (2017) Arduino Retrieved from: <http://www.arduino.org/> acessado a 28-04-2017; <https://create.arduino.cc/projecthub/mjrobot/physical-computing-scratch-for-arduino-72661> acessado a 14-04-2017; <https://www.arduino.cc/en/Guide/Introduction>; <https://www.arduino.cc/en/Main/Software> acessado a 16-03-2017; <https://www.arduino.cc/en/Hacking/Bootloader> acessado a 12-02-2017; <https://www.arduino.cc/en/Tutorial/PWM> acessado a 11-03-2017
- Atmel (2017) <http://www.atmel.com/products/microcontrollers/avr/default.aspx> acessado a 16-04-2017; [http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P\\_datasheet\\_Complete.pdf](http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf) acessado a 24-04-2017
- Beer, R. D., Chiel, H. J., Drushel, R. F. (1999). Using Autonomous Robotics to Teach Science and Engineering. *Communication of the ACM*, 42(6), 85-92
- Benitti, F. B. V. (2012). Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 53(3), 978-988.
- Bento, A. V. (2012). Investigação Quantitativa e Qualitativa: Dicotomia ou complementaridade? *Revista JA (Ass. Acad. Univ. Da Madeira)*, 64, 40 – 43.
- Bredendfeld, A., Hofmann, A., Steinbauer, G. (2010). Robotics in Education Initiatives in Europe - Status, Shortcomings and Open Questions. *Proceedings of*

SIMPAR 2010 Workshops Intl. Conf. On Simulation, Modeling And Programming For Autonomous Robots Darmstadt (Germany) November 15-16, 2010, 568-574.

- Cabral, C. P. Robótica educacional e resolução de problemas: uma abordagem microgenética da construção do conhecimento. 2011. Anais da Mostra Nacional de Robótica (MNR) 2011.
- Charlton, P., & Luckin, R. (2012). Time to reload? Computational thinking and computer science in schools. What researches says? Londres: Knowledge Lab - Institute of Education, University of London.  
<https://knowledgeillusion.wordpress.com/resources/27-april-what-the-research-says-computing-in-schools/> acessado a 25-02-2017
- Chella, M.T. (2002). Ambiente de Robótica Educacional com Logo. XXII Congresso da Sociedade Brasileira de Computação-SBC2002. Florianópolis, 2002.
- Computadores e Robótica Educativa na Escola: tendências evidenciadas nas produções
- Costa, C. J., & Aparício, M. (2013). Social networks: intentions and usage. In *Proceedings of the 2013 International Conference on Information Systems and Design of Communication* (pp. 101-107). ACM
- Costa, C. J., Aparício, M. & Cordeiro, C., (2012b). Web-Based graphic environment to support programming in the beginning learning process. In Proceedings of the 11th international conference on Entertainment Computing (ICEC'12), Marc Herrlich, Rainer Malaka, and Maic Masuch (Eds.). Springer-Verlag, Berlin, Heidelberg, 413-416.
- Costa, C. J., Aparício, M., & Cordeiro, C. (2012a). A solution to support student learning of programming. In Proceedings of the Workshop on Open Source and Design of Communication (pp. 25–29). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=2316942> acessado a 21-01-2017
- Costa, C. J., Aparício, M., & Pierce, R. (2009). Evaluating information sources for computer programming learning and problem solving. In Proceedings of the 9th WSEAS international conference on Applied computer science (ACS'09), Roberto Revetria, Valeri Mladenov, and Nikos Mastorakis (Eds.). World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, 218-223.
- Costa, C. J., Aparício, M., Caldeira, T. (2014). Teaching Programming through Open Source Robotics. <http://rcj.robocup.org/werob/werob2014/paper2.pdf>. Acessado a 15/01/2017.
- Costa, C. J., Ferreira, E., Bento, F., Aparício, M. (2016). Enterprise resource planning adoption and satisfaction determinants. *Computers in Human Behavior* 63 (2016) 659-671.
- Dargains, A. R. (2015). Estudo Exploratório Sobre o Uso da Robótica Educacional no Ensino de Programação Introdutória. (Dissertação de mestrado apresentada ao Programa

- DGE (2017) Direção Geral do Ensino; Acedido: <http://erte.dge.mec.pt/programacao-e-robotica-no-ensino-basico-0> acedido a 04-02-2017
- Dias, J., Teixeira, V. L. M. O. (2016). A Plataforma Arduino como Instrumento Metodológico no Curso de Tecnologia em Análise e Desenvolvimento de Sistemas. *Revista Eletrónica da UNIVAR*, 16, (2), 23-28.
- Dissertação em Engenharia Informática, Universidade da Madeira, Funchal – Portugal, 126 pág.
- Eguchi, Amy (2014). Robotics as a Learning Tool for Educational Transformation. *Proceedings of 4th International Workshop Teaching Robotics, Teaching with Robotics & 5th International Conference Robotics in Education Padova (Italy) July 18, 2014*, 27-34.
- Ferreira F., Costa C. J., Aparício M., & Aparício S. (2017). Learning programming: A continuance model, 2017 12th Iberian Conference on Information Systems and Technologies (CISTI), Lisbon, Portugal:IEEE
- Fritzing (2017) <http://fritzing.org/home/> acedido a 14-12-2016
- Gaspar, L. A. d. S. (2007). Os Robots nas Aulas de Informática: Plataformas e Problemas.
- Godoy, A. S. (1995). Pesquisa Qualitativa: tipos fundamentais. *Revista de Administração de Empresa*, 35(3), 20-29.
- Gomes, A. J. (2010). Dificuldades de aprendizagem de programação de computadores: contributos para a sua compreensão e resolução. Dissertação de Grau de Doutor em Engenharia Informática, Universidade de Coimbra, 468 pág.
- Gomes, A., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. *International Conference of Engineering Education – ICEE 2007*. September 3 -7, Coimbra, Portugal.
- Gomes, A., Henriques, J., & Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias*, 1(1), 93-103.
- Gomes, C. G., Silva, F. O., Botelho, J. C., Souza, A. R. (s/ data). A robótica como facilitadora do processo ensino-aprendizagem de matemática no ensino fundamental. In: PIROLA, NA. org. *Ensino de ciências e matemática, IV: temas de investigação [online]*. São Paulo: Editora UNESP; São Paulo: Cultura Acadêmica, 2010, 205-221.
- Gomes, G., Abrantes P. (2012). A Robótica Educativa no Ensino da Programação. II Congresso Internacional TIC e Educação – Universidade de Lisboa, 2039-2055.
- Gomes, G., Martinho, J., Bernardo M., Matos, F., Abrantes P. (2012). Dificuldades na Aprendizagem da Programação no Ensino Profissional. II Congresso Internacional TIC e Educação – Universidade de Lisboa, pp. 438-448.

- Guedes, A. L.; Kerber, F. M. (2010). Usando a robótica como meio educativo.
- Järvinen, P. (2007). Action research is similar to design science. *Quality & Quantity*, 41(1), 37-54.
- Lego (2017 a) Fundação Lego, Acedido: <http://www.legofoundation.com/tr-tr/newsroom/articles/2017/honouring-seymour-papert> acedido a 14-04-2017
- Lego (2017 b) Mindstorm systems, Acedido <https://www.lego.com/en-us/mindstorms> acedido a 15-02-2017
- Lessa, V. E., Forigo, F. M., Teixeira A. C., Licks, G. P. (2015). Programação de Lógica de Programação, DOI: 10.1109/CISTI.2014.6876894, 134-138, <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6876894>
- Margetson, D. (1997). Why is Problem – Based Learning a Challenge? In D. Boud & G. Feletti (Eds.), *The Challenge of Problem – Based Learning* 36-44.
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems*, 15(4), 251-266.
- Matos, F. J. P. (2012). Aprendizagem Baseada em Problemas com Recurso à Robótica na Programação. Dissertação de Mestrado em Ensino de Informática, Instituto de Educação da Universidade de Lisboa, 185 pág.
- Melo, R. M., Feliciano, F. D. O., Brito, J. A., Filho, I. J. M., Carvalho, R. S.; Gomes, A.
- Murphy, L., Richards, B., McCauley, R., Morrison, B. B., Westbrook, S., & Fossum, T. (2006). Women catch up: gender differences in learning programming concepts. In *ACM SIGCSE Bulletin* (Vol. 38, pp. 17–21). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=1121350> acedido a 15-01-2017
- Oliveira, D. F. R. (2013). A Robótica Educativa no Ensino e Aprendizagem de Conceitos de Programação e Algoritmos. (Relatório da Prática de Ensino Supervisionada Mestrado em Ensino de Informática), Universidade de Lisboa., 186 pág.
- Oliveira, D., Ferreira, S., Celestino, H., Ferreira, S., Abrantes, P. (2012). Uma Proposta de Ensino-Aprendizagem de Programação Utilizando Robótica Educativa e Storytelling. II Congresso Internacional TIC e Educação – Universidade de Lisboa, pp. 2567-2576.
- Oliveira, E., Finizola J., Junior, J., Genuino, R., Santos, R., Oliveira, A. L., Souza, F. (2016). Utilizando a robótica para o ensino e aprendizagem de conceitos de programação: um relato de experiência. Congresso Regional sobre Tecnologias da Educação, 232-243.
- Olsson M, Mozelius P and Collin J “Visualisation and Gamification of e-Learning and Programming Education” *The Electronic Journal of e-Learning* Volume 13 Issue 6 2015, (pp441-454).

- Patsko, L. F. (2006). Tutorial Aplicações, Funcionamento e Utilização de Sensores. [http://www.maxwellbohr.com.br/downloads/robotica/mec1000\\_kdr5000/tutorial\\_eletronica\\_-\\_aplicacoes\\_e\\_funcionamento\\_de\\_sensores.pdf](http://www.maxwellbohr.com.br/downloads/robotica/mec1000_kdr5000/tutorial_eletronica_-_aplicacoes_e_funcionamento_de_sensores.pdf) acessado a 02-03-2017
- Pedrosa, M. A. & João, P. (2013). Aprendizagem Baseada em Resolução de Problemas na Educação em Ciências para a sustentabilidade. Atas do Encontro sobre Educação em Ciências através da Aprendizagem Baseada na Resolução de Problemas, Braga: Universidade do Minho, 63-78.
- Pereira, R., Costa, C. J. & Aparicio, J. T. (2017) . Gamification to support programming learning. In 2017 12th Iberian Conference on Information Systems and Technologies (CISTI). Lisbon, Portugal: IEEE.
- Pierce, R. and Aparicio, M. “Resources to support computer programming learning and computer science problem solving,” in Proceedings of the Workshop on Open Source and Design of Communication, New York, NY, USA, (2010), pp. 35–40.
- Pinheiro, P., Aparicio, M., & Costa, C. (2014, May). Adoption of cloud computing systems. In *Proceedings of the International Conference on Information Systems and Design of Communication* (pp. 127-131). ACM.
- Pio, J. L. S., Castro, T. H. C., Júnior, A. N. C. (2006). A Robótica Móvel como Instrumento de Apoio à Aprendizagem de Computação. XVII simpósio Brasileiro de Informática na Educação 2006, 497-506.
- Piteira M., & Costa C. J. (2013). Learning computer programming: study of difficulties in learning programming. In Proceedings of the 2013 International Conference on Information Systems and Design of Communication (ISDOC '13). ACM, New York, NY, USA, 75-80.
- Piteira M., & Costa C. J. (2017). Gamificação: Framework Concetual para Cursos Online de Aprendizagem da Programação. In 2017 12th Iberian Conference on Information Systems and Technologies (CISTI). Lisbon, Portugal: IEEE.
- Piteira M., Costa C. J. & Haddad, S. R., (2012). Educational computer programming tools. In Proceedings of the Workshop on Open Source and Design of Communication (OSDOC '12). ACM, New York, NY, USA, 57-60.
- Piteira M., Costa C. J. (2012). Computer programming and novice programmers. In Proceedings of the Workshop on Information Systems and Design of Communication (ISDOC '12). ACM, New York, NY, USA, 51-53.
- programming course with physical computing modules. Frontiers in Education Conference (FIE), 2014 IEEE, 22-25 Oct. 2014.
- Ribeiro, C., (2006). Robô Carochinha um estudo sobre robótica educativa no 1º ciclo do ensino básico. Tese de Mestrado em Educação Tecnológica Educativa, Instituto de Educação e Psicologia, Universidade do Minho, 207 pág.
- Romero-Zaliz R., Rubio, M. A., Madrid, C. M. A. P. (2014). Enhancing an introductory

- S. (2014). Utilização do Arduíno como Estratégia Pedagógica na Aprendizagem da
- Sampaio, M. R. A. F. (2012). Avaliação da competência de resolução de problemas explorando as TIC. Dissertação de Mestrado em Didática, Universidade de Aveiro, 176 pág.
- Santos, E. A. A. d., Fermé, E. L., & Fernandes, E. M. d. S. (2006). Droide virtual -
- Santos, E. A., Fermé, F., Elsa, F. (2014). Utilização de Robots no Ensino de Programação: O Projecto Droid. Departamento de Matemática e Engenharia – Universidade da Madeira.  
[http://www.cee.uma.pt/people/faculty/elsa.fernandes/artigos/IXSPCE\\_Santos\\_etal.pdf](http://www.cee.uma.pt/people/faculty/elsa.fernandes/artigos/IXSPCE_Santos_etal.pdf) acedido a 20-12-2016
- Sousa, N., Costa, C. J., Aparício, M. (2017). Ba: Um Fator Determinante no Uso de Sistemas de Gestão do conhecimento. Revista Ibérica de Sistemas e Tecnologias de Informação. DOI: 10.17013/risti.22.1–19.
- Trindade, R. (2014). A autoaprendizagem no ensino superior e a aprendizagem baseada na resolução de problemas: perspetivas e questões. Revista Lusófona de Educação, 27 (27), 43-57.
- Unoesc e Ciências – ACET, Joaçaba, 1(2), 199-208.
- Vasconcelos, C., & Almeida, A. (2012). Aprendizagem Baseada na Resolução de Problemas no Ensino das Ciências: Propostas de Trabalho para Ciências, Biologia e Geologia. Porto: Porto Editora.
- Viegas, T. R., Okuyama, F. Y., Paravisi, M., Bertagnolli, S. C. (2015).
- Vimeo (2017) <https://vimeo.com/199805464> acedido a 14-04-2017
- Zanetti, H. A. P. (2014). Análise semiótica do uso de Robótica Pedagógica no ensino de Programação de Computadores. Dissertação de Mestrado em Ciências da Computação, Faculdade Campo Limpo Paulista, Brasil, 161

# ANEXO A – Desenvolvimento do *kit de led's e sensores*

## 1 - PROJETO

Elaboração de um módulo de *hardware* para proporcionar a utilização do Arduino (Figura 33) em sala de aula. Este módulo permitirá a ligação ao Arduino de modo a que aquando da programação, os resultados sejam observados no módulo.

## 2 - CIRCUITO

Vão ser construído os seguintes circuitos:

- *Led 1* - led vermelho ligado ao pino 13 do Arduino
- *Led 2* - led amarelo ligado ao pino 11 do Arduino
- *Led 3* - led verde ligado ao pino 9 do Arduino
- *Led 4* - led vermelho ligado ao pino 7 do Arduino
- *Led 5* - led verde ligado ao pino 5 do Arduino
- Potenciómetro ligado ao pino A3
- Sensor de Temperatura ligado ao pino A2
- *LDR* - Sensor de Luminosidade ligado ao pino A0

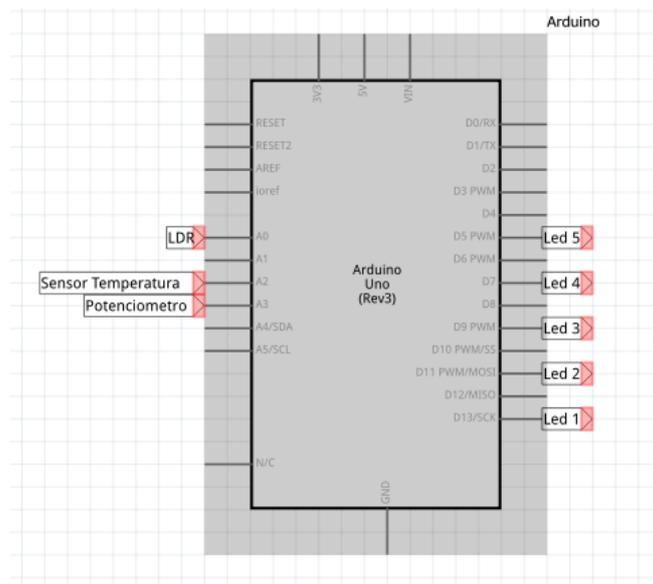


FIGURA 33 - PINOUT ARDUÍNO UNO

## Resistências

A Resistência é um dispositivo amplamente utilizado em equipamentos elétricos e circuitos eletrônicos, cujas aplicações principais são: geração de calor, limitação da corrente elétrica e divisor de tensão.

O seu funcionamento baseia-se na resistência à passagem da corrente elétrica, a qual gera calor por efeito Joule e uma queda de tensão nos seus terminais.

A unidade de medida da resistência elétrica é o *Ohm* ( $\Omega$ ), sendo muito empregados os seus múltiplos: *k $\Omega$*  e *M $\Omega$* , que correspondem, respetivamente, a mil ohms e um milhão de ohms.

As resistências podem ser construídos utilizando-se carvão, silício ou ligas metálicas (Figura 34).

O valor da resistência e a percentagem de tolerância de uma resistência de carvão são determinados por faixas coloridas com código de cores (Figura 35).

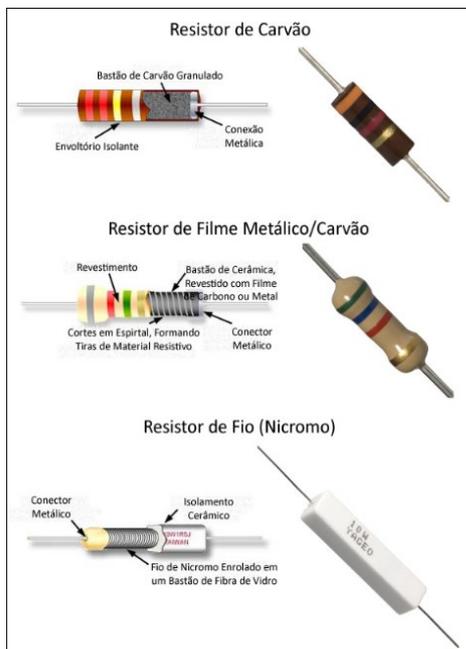


FIGURA 34 - RESISTÊNCIAS

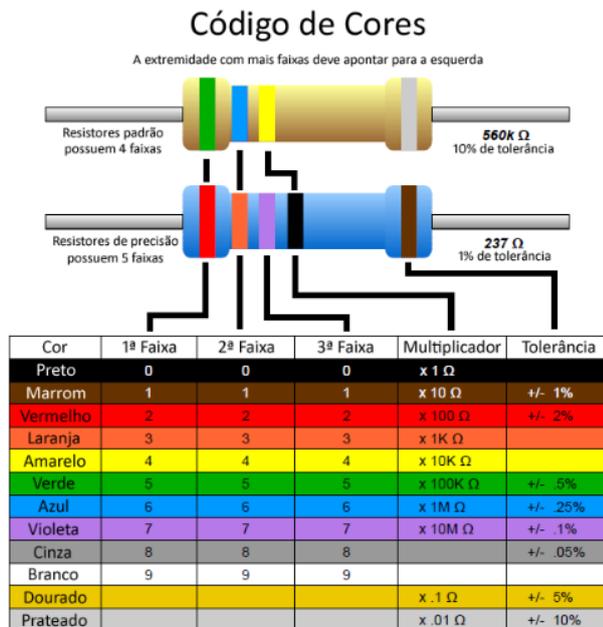


FIGURA 35 - CÓDIGO DE CORES DAS RESISTÊNCIAS

## Sensores Analógicos

Os sensores analógicos são os dispositivos comuns. Apresentam esta designação dado que se baseiam em sinais analógicos. Estes são os que, mesmo limitados entre dois valores de tensão, podem assumir infinitos valores intermédios. Isso significa que, pelo menos teoricamente, para cada nível da condição medida, haverá um nível de tensão correspondente.

### LDR (*Light Dependent Resistor*)

O LDR (Figura 36) é um dispositivo cuja resistência varia de acordo com a luminosidade. Ao ser submetido a uma luz mais intensa, verifica-se que a sua resistência diminui. Ao utilizar um circuito divisor de tensão, é possível variar a tensão através dessa variação da resistência (Figura 37)

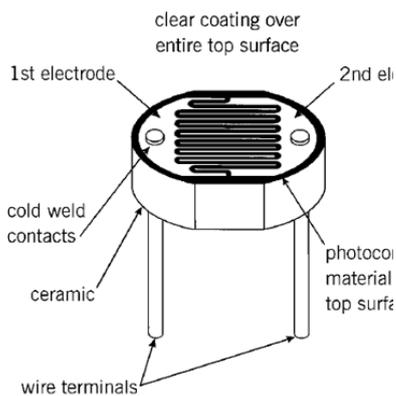


FIGURA 36 - LDR  
(COMPONENTES)

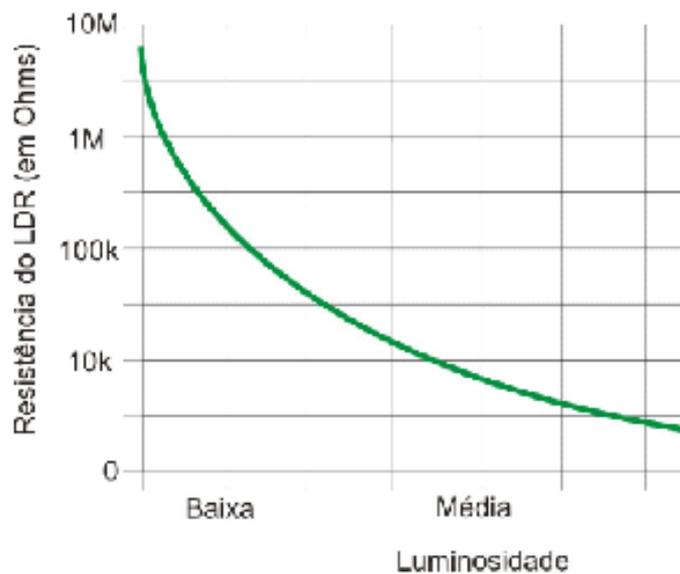
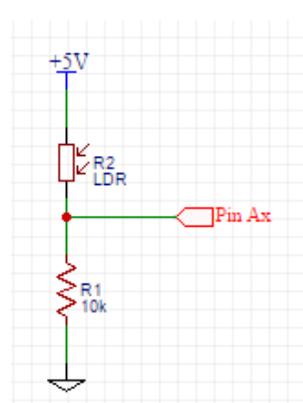


FIGURA 37 - GRÁFICO DE RESPOSTA DE UMA LDR

Para a ligação do LDR (5-10kΩ) ter-se-á que se colocar uma resistência (R1) no valor de 10kΩ para o divisor de tensão (Figura 39). Os valores de tensão máximo e mínimo são apresentados na (Figura 38).

$V_{\max} = V_{in} * R1 / (R1 + LDR_{\min})$ $V_{\max} = 5 * 10000 / 10005$ $V_{\max} = 4.9 \text{ (V)}$	
$V_{\min} = V_{in} * R1 / (R1 + LDR_{\max})$ $V_{\min} = 5 * 10000 / 20000$ $V_{\min} = 2.5 \text{ (V)}$	
<p>FIGURA 38 - VALORES MÁXIMO E MÍNIMO DA TENSÃO DE LEITURA DA <i>LDR</i></p>	<p>FIGURA 39 - LIGAÇÃO DO <i>LDR</i></p>

### Potenciômetro

O potenciômetro (Figura 40) pode ser mais claramente definido como sendo um divisor de tensão variável. É composto por uma faixa de material resistivo (geralmente grafite), ligado entre os seus dois terminais externos. Nesse material, desliza um cursor, ligado diretamente ao terminal central do potenciômetro. Esse cursor pode ser movimentado através de um eixo rotativo ou um pino de plástico ou metal. Quando se altera a posição do cursor, ocorre uma alteração da resistência entre o terminal central e os dois terminais externos do potenciômetro.

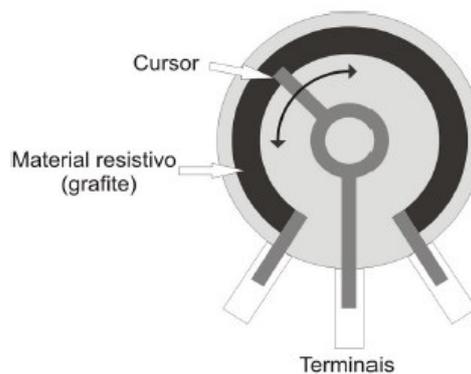


FIGURA 40 - INTERIOR DO POTENCIÔMETRO

O valor da resistência total do potenciômetro (medida entre seus terminais externos) é sempre constante, ou seja, o potenciômetro funciona como duas resistências em série,

onde é possível alterar simultaneamente os seus valores, desde que a soma das resistências seja constante (Figura 41). A ligação do potenciômetro é apresentada na (Figura 42).

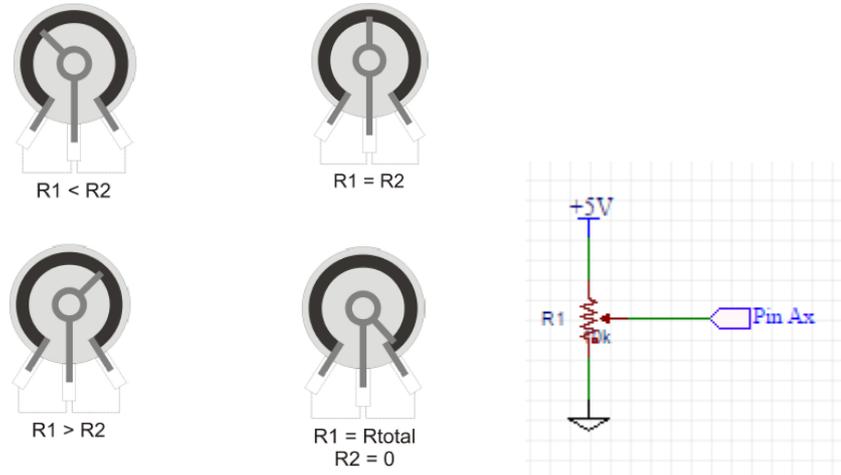


FIGURA 41 - RESISTÊNCIA ENTRE OS TERMINAIS DO POTENCIÓMETRO

FIGURA 42 - LIGAÇÃO DO POTENCIÓMETRO

### Sensor de Temperatura

Para a medição da temperatura é possível utilizar circuitos integrados dedicados (Figura 43). Dentre as vantagens desses componentes está a facilidade na sua aplicação, dado que todos os componentes necessários se dispõem numa única peça. O princípio de funcionamento baseia-se no fato de que o comportamento dos materiais semicondutores dependerem da temperatura. A principal classificação desses sensores é feita de acordo com o sinal de saída. Os modelos mais simples são os analógicos. Assim como qualquer sensor analógico, eles possuem um sinal de saída cuja tensão corresponde à temperatura do ambiente onde se encontra o sensor. A ligação para a medição da tensão é feito conforme indicado na (Figura 44).

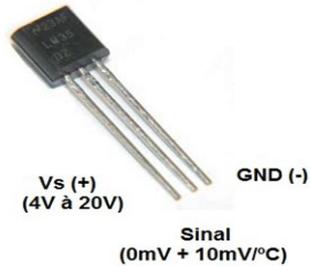


FIGURA 43 - SENSOR DE TEMPERATURA

Segundo os dados do sensor, este funciona para correntes de polarização compreendidas entre 500uA e 5mA. A fórmula de cálculo para a corrente é:  $I=(V_{cc}-V_{LM335})/R1$

Exemplo de valores para de temperatura de -10°C e de 40°C

$$I@ 40^{\circ}\text{C} = (5-(2.73+(40*0.01)))/2200 = 850\mu\text{A}$$

$$I@ -10^{\circ}\text{C} = (5-(2.73+(-10*0.01)))/2200 = 1.07\text{mA}$$

### Led's

O Led (díodo emissor de luz) é um díodo que quando polarizado diretamente emite luz (exemplo: verde, vermelho, azul, branco, infravermelho). A polarização é feita em série com uma resistência limitador de corrente Figura 46. Para a maioria dos LED's existentes no mercado, a tensão típica é entre 1.5V e 2.5V e a corrente no led é calculada de modo a não passar a corrente máxima suportada Figura 47. A ligação do led é feito segundo o esquema apresentado na Figura 48.

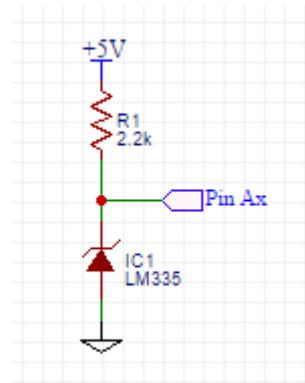


FIGURA 44 - LIGAÇÃO DO SENSOR DE TEMPERATURA

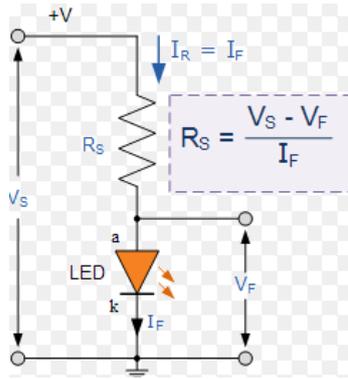


FIGURA 45 - ESQUEMA DE LIGAÇÃO DO LED

A corrente que circula no LED é:

$$I_D = \frac{V_s - V_D}{R}$$

Em que:

$V_s$ - tensão na fonte

$V_f$ - tensão no diodo

$R$  – Resistência de carga

FIGURA 46 - CORRENTE NO LED

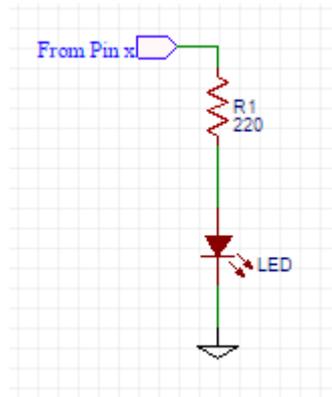


FIGURA 47 - IMPLEMENTAÇÃO DA LIGAÇÃO DO LED

## MONTAGEM

Implementação (Figura 48) e teste de um protótipo (Fritzing, 2017)

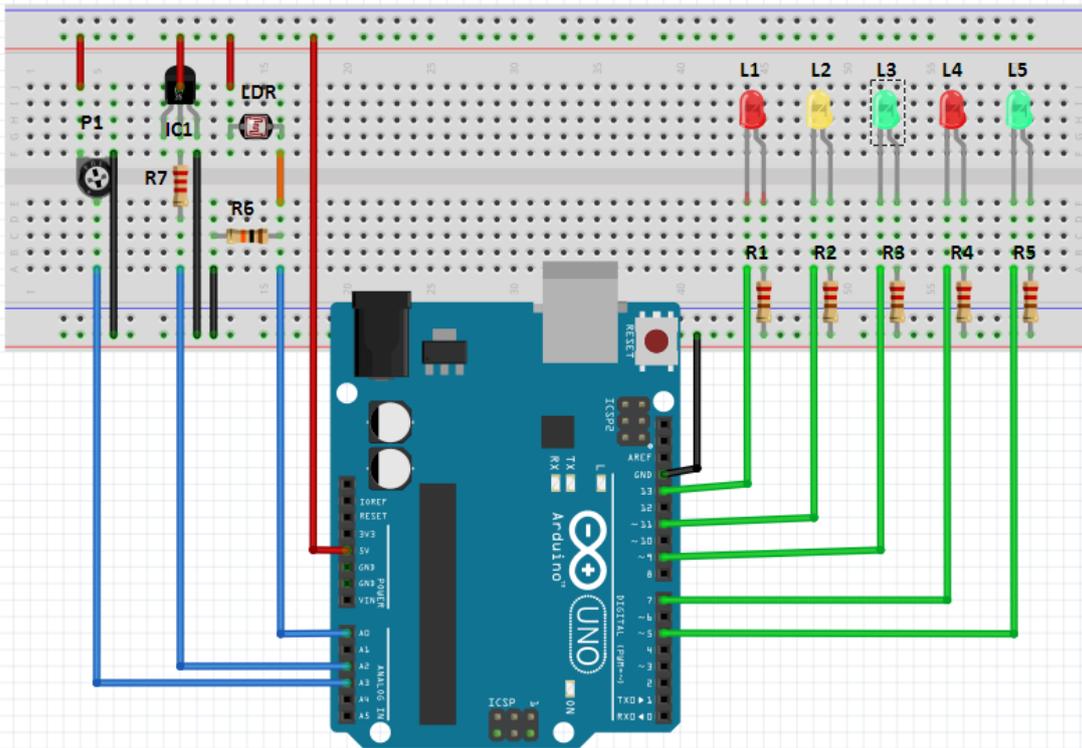


FIGURA 48 - MONTAGEM DO PROTÓTIPO

Após a montagem do circuito num *bread-board* procedeu-se ao teste para a verificação do *software* (Anexo B) e do *hardware*. A ligação do Arduino ao Computador é feita via *USB*.

Utilizou-se os seguintes módulos para o teste:

- a) Piscar *led's* intermitentes (“intermitente”) – Faz a verificação de cada um dos *led's* individual
- b) Leitor da temperatura (“temperatura”) – Lê a temperatura do LM335
- c) Leitor do sensor de luminosidade (“luminosidade”) – Lê o sensor de luminosidade
- d) Leitura do potenciômetro (“potenciômetro”) – Lê o potenciômetro

### 3.2 – Produção

Para a produção de 10 protótipos foi necessário o seguinte material constante na Tabela 9

TABELA 9- MATERIAL PARA A MONTAGEM

Material	Quantidade
<i>Led's</i> vermelhos	20
<i>Led's</i> amarelos	10
<i>Led's</i> verdes	20
Resistências 220 $\Omega$	50
Resistências 10k $\Omega$	10
Resistências 2k2 $\Omega$	10
<i>LDR</i> (5-10k $\Omega$ )	10
IC LM335	10
Placa circuito impresso (5x5)	10
Pinos (em régua)	160

Equipamento e material necessário:

Ferro de soldar

X-ato

Solda

Alicate corte

Multímetro

Fonte de alimentação 5V

Fios e *jumpers*

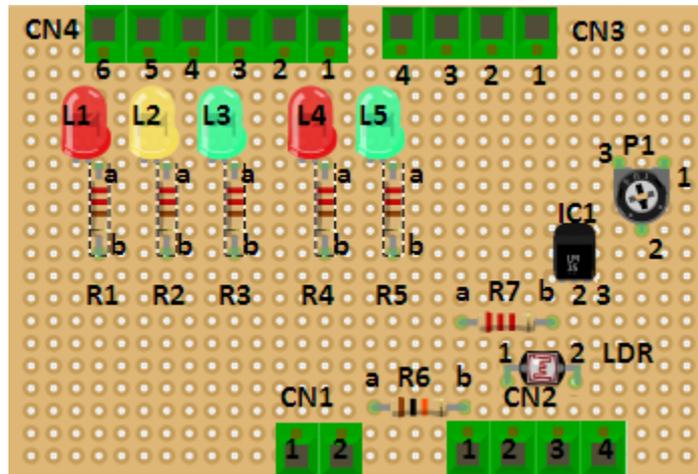


Figura 49 - Kit de led's e sensores

Após a montagem dos circuitos procedeu-se ao teste das mesmas.

1º - Verificar o valor de resistência entre:

- a) R1a e o pino CN1-1 (*Gnd*)  
 Valor entre 200  $\Omega$  e 240  $\Omega$  - OK  
 Fora destes valores – Falha 1
- b) R2a e o pino CN1-1 (*Gnd*)  
 Valor entre 200  $\Omega$  e 240  $\Omega$  - OK  
 Fora destes valores – Falha 2
- c) R3a e o pino CN1-1 (*Gnd*)  
 Valor entre 200  $\Omega$  e 240  $\Omega$  - OK  
 Fora destes valores – Falha 3
- d) R4a e o pino CN1-1 (*Gnd*)  
 Valor entre 200  $\Omega$  e 240  $\Omega$  - OK  
 Fora destes valores – Falha 4
- e) R5a e o pino CN1-1 (*Gnd*)  
 Valor entre 200  $\Omega$  e 240  $\Omega$  - OK  
 Fora destes valores – Falha 5

2º - Ligação da fonte de alimentação nos pinos do CN1 -1 (+5V), CN1 -2 (*Gnd*)

3º - Teste dos *led's*

Ao colocar no ânodo do *led 1* a tensão de 5V este:

Acende – OK

Não acende – Falha 11

Ao colocar no ânodo do *led 2* a tensão de 5V este:

Acende – OK

Não acende – Falha 12

Ao colocar no ânodo do *led 3* a tensão de 5V este:

Acende – OK

Não acende – Falha 13

Ao colocar no ânodo do *led 4* a tensão de 5V este:

Acende – OK

Não acende – Falha 14

Ao colocar no ânodo do *led 5* a tensão de 5V este:

Acende – OK

Não acende – Falha 15

4º - Teste aos pinos do conector CN1

Ao colocar no pino 2 do CN1 a tensão de 5V o *led 1*:

Acende – OK

Não acende – Falha 21

Ao colocar no pino 3 do CN1 a tensão de 5V o *led 2*:

Acende – OK

Não acende – Falha 22

Ao colocar no pino 4 do CN1 a tensão de 5V o *led 3*:

Acende – OK

Não acende – Falha 23

Ao colocar no pino 5 do CN1 a tensão de 5V o *led* 4:

Acende – OK

Não acende – Falha 24

Ao colocar no pino 6 do CN1 a tensão de 5V o *led* 5:

Acende – OK

Não acende – Falha 25

#### 5º - Teste do potenciômetro

Medir a tensão entre o pino P1-1 e P1-2. Ao rodar o botão central os valores a medir estarão entre 0V e 5V.

Valores entre 0V e 5V – OK

Valor sempre constante – Falha 31

Verificar se há continuidade entre o P1-2 e o CN2-3

Se houver – OK

Se não houver – Falha 32

#### 6º - Teste do LDR

Medir a tensão entre os pinos 1 e 2 do LDR

Com o LDR tapado o valor é na ordem dos 0V – OK

Outros valores – Falha 41

Com o LDR aberto o valor é na ordem de 2.5V – OK

Outros valores – Falha 42

#### 7º - Teste ao sensor de temperatura – IC1 (LM335)

Medir a continuidade entre o pino 3 do LM335 e o CN1-1

Valor 0 – OK

Valor diferente – Falha 51

Medir a tensão do pino2 do LM335

Valor menor que 1.87V – OK

Aquecer o LM339

Medir a tensão do pino2 do LM335

Valor superior ao anterior > 1.87V– OK

Não alteração do valor – Falha 52

Resolução das falhas:

Falha 1:

- a) Verificar se o pino R1b está ligado ao CN1-1 (*Gnd*)
- b) Verificar o valor de R1 (220Ω).

Falha 2:

- a) Verificar se o pino R2b está ligado ao CN1-1 (*Gnd*)
- b) Verificar o valor de R2 (220Ω).

Falha 3:

- a) Verificar se o pino R3b está ligado ao CN1-1 (*Gnd*)
- b) Verificar o valor de R3 (220Ω).

Falha 4:

- a) Verificar se o pino R4b está ligado ao CN1-1 (*Gnd*)
- b) Verificar o valor de R4 (220Ω).

Falha 5:

- a) Verificar se o pino R5b está ligado ao CN1-1 (*Gnd*)
- b) Verificar o valor de R5 (220Ω).

Falha 11:

- a) Verificar se o *led* está corretamente montado.
- b) Verificar se existe ligação entre o cátodo do L1 e a R1a
- c) Verificar se a R1b está ligada ao CN1-1
- d) Verificar se o L1 está em condições de funcionamento
- e) Verificar o valor de R1

Falha 12:

- a) Verificar se o *led* está corretamente montado.
- b) Verificar se existe ligação entre o cátodo do L2 e a R2a
- c) Verificar se a R2b está ligada ao CN1-1
- d) Verificar se o L2 está em condições de funcionamento
- e) Verificar o valor de R2

Falha 13:

- f) Verificar se o *led* está corretamente montado.
- g) Verificar se existe ligação entre o cátodo do L3 e a R3a
- h) Verificar se a R3b está ligada ao CN1-1
- i) Verificar se o L3 está em condições de funcionamento
- j) Verificar o valor de R3

Falha 14:

- f) Verificar se o *led* está corretamente montado.
- g) Verificar se existe ligação entre o cátodo do L4 e a R4a
- h) Verificar se a R4b está ligada ao CN1-1
- i) Verificar se o L4 está em condições de funcionamento
- j) Verificar o valor de R4

Falha 15:

- a) Verificar se o *led* está corretamente montado.
- b) Verificar se existe ligação entre o cátodo do L5 e a R5a
- c) Verificar se a R5b está ligada ao CN1-1
- d) Verificar se o L5 está em condições de funcionamento

- e) Verificar o valor de R5

Falha 21:

- a) Verificar se existe ligação entre o ânodo do L1 e o CN4-6

Falha 22:

- a) Verificar se existe ligação entre o ânodo do L2 e o CN4-4

Falha 23:

- a) Verificar se existe ligação entre o ânodo do L3 e o CN4-2

Falha 24:

- a) Verificar se existe ligação entre o ânodo do L4 e o CN3-4

Falha 25:

- a) Verificar se existe ligação entre o ânodo do L5 e o CN3-2

Falha 31:

- a) Verificar se o P1-1 está ligado ao CN1-1
- b) Verificar se o P1-3 está ligado ao CN1-2
- c) Verificar se o potenciômetro está em condições

Falha 32:

- a) Verificar a continuidade entre o P1-2 e o CN2-3

## ANEXO B – Software para teste do *kit de led's* e sensores

a) Piscar *led's* intermitente Figura 52

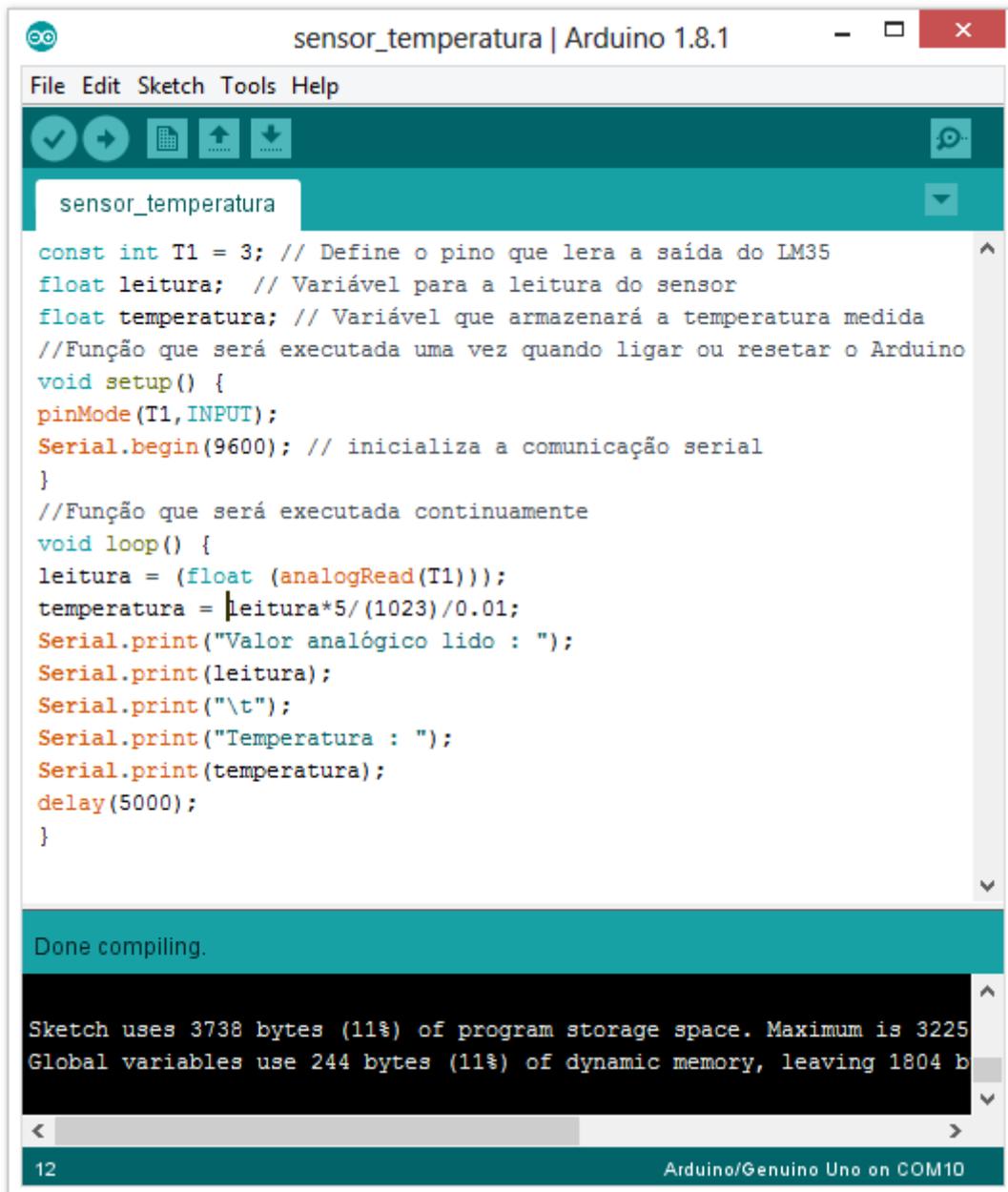
```

SEQUENCIA_V1_5_ALTERNADOS - sequencia_V1_5_alternados.ino...
File Edit Sketch Tools Help
sequencia_V1_5_alternados
int leds[]={13, 11, 9, 7, 5, 7, 9, 11}; //leds ligados aos pinos 8,9,10 e 11
int contador = 0; //numero de leds
int espera = 150; //tempo de espera
int numero_leds = 5; //numero de leds ligados ao arduino
void setup() {
  Serial.begin(9600);
  Serial.println("Valor do N_led ");
  // inicializa os pinos como output
  for(contador=0; contador < numero_leds; contador++){
    pinMode(leds[contador], OUTPUT);
  }
}
// the loop function runs over and over again forever
void loop() {
  int n_led;
  for(contador=0; contador < sizeof(leds)/sizeof(int); contador++){
    digitalWrite(leds[contador], HIGH); // liga o LED
    delay(espera);
    digitalWrite(leds[contador], LOW); // desliga o LED
    delay(espera);
  }
}
Done compiling.
Sketch uses 2082 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 220 bytes (10%) of dynamic memory, leaving 1828 bytes for
16 Arduino/Genuino Uno on COM10

```

FIGURA 50 - CÓDIGO EM C PARA TESTE DOS LED'S

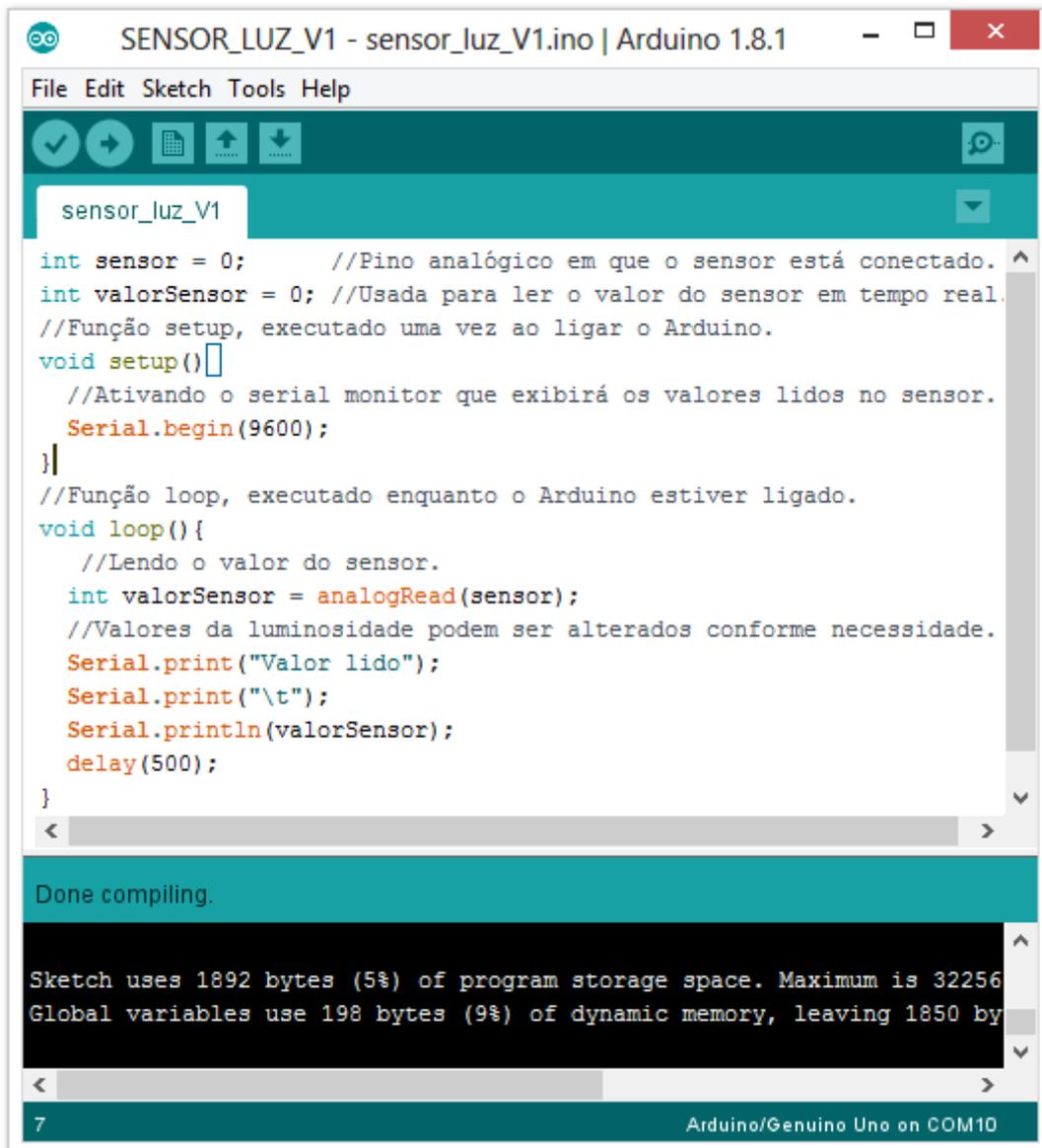
## b) Leitor da temperatura (Figura 53)



```
sensor_temperatura | Arduino 1.8.1
File Edit Sketch Tools Help
sensor_temperatura
const int T1 = 3; // Define o pino que lera a saída do LM35
float leitura; // Variável para a leitura do sensor
float temperatura; // Variável que armazenará a temperatura medida
//Função que será executada uma vez quando ligar ou resetar o Arduino
void setup() {
  pinMode(T1, INPUT);
  Serial.begin(9600); // inicializa a comunicação serial
}
//Função que será executada continuamente
void loop() {
  leitura = (float (analogRead(T1)));
  temperatura = leitura*5/(1023)/0.01;
  Serial.print("Valor analógico lido : ");
  Serial.print(leitura);
  Serial.print("\t");
  Serial.print("Temperatura : ");
  Serial.print(temperatura);
  delay(5000);
}
Done compiling.
Sketch uses 3738 bytes (11%) of program storage space. Maximum is 3225
Global variables use 244 bytes (11%) of dynamic memory, leaving 1804 b
12 Arduino/Genuino Uno on COM10
```

FIGURA 51 - CÓDIGO EM C PARA TESTE DO SENSOR DE TEMPERATURA

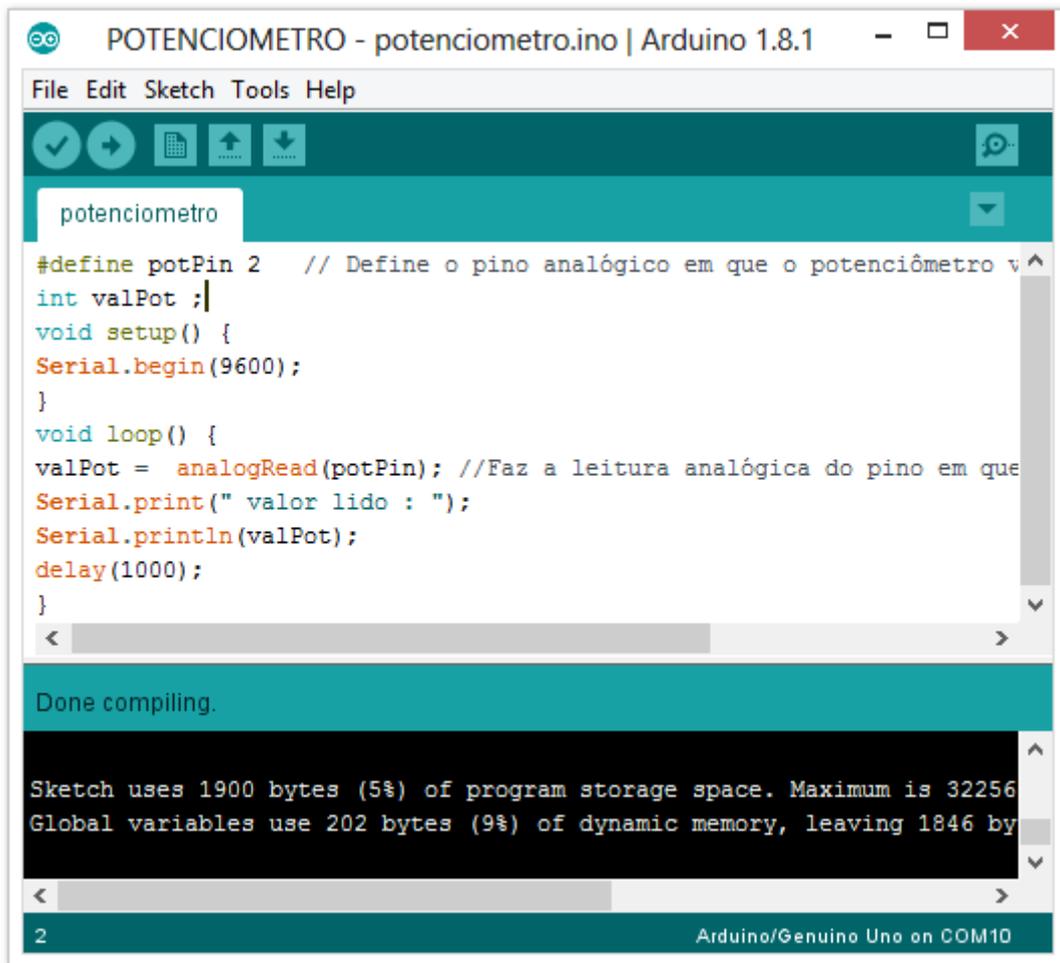
c) Leitor do sensor de luminosidade (Figura 54)



```
SENSOR_LUZ_V1 - sensor_luz_V1.ino | Arduino 1.8.1
File Edit Sketch Tools Help
sensor_luz_V1
int sensor = 0; //Pino analógico em que o sensor está conectado.
int valorSensor = 0; //Usada para ler o valor do sensor em tempo real.
//Função setup, executado uma vez ao ligar o Arduino.
void setup()
  //Ativando o serial monitor que exibirá os valores lidos no sensor.
  Serial.begin(9600);
}
//Função loop, executado enquanto o Arduino estiver ligado.
void loop(){
  //Lendo o valor do sensor.
  int valorSensor = analogRead(sensor);
  //Valores da luminosidade podem ser alterados conforme necessidade.
  Serial.print("Valor lido");
  Serial.print("\t");
  Serial.println(valorSensor);
  delay(500);
}
Done compiling.
Sketch uses 1892 bytes (5%) of program storage space. Maximum is 32256
Global variables use 198 bytes (9%) of dynamic memory, leaving 1850 by
7 Arduino/Genuino Uno on COM10
```

FIGURA 52 - CÓDIGO EM C PARA O TESTE DO SENSOR DE LUZ

d) Leitura do potenciômetro (Figura 55)



```
POTENCIOMETRO - potenciometro.ino | Arduino 1.8.1
File Edit Sketch Tools Help
potenciometro
#define potPin 2 // Define o pino analógico em que o potenciômetro v
int valPot ;|
void setup() {
Serial.begin(9600);
}
void loop() {
valPot = analogRead(potPin); //Faz a leitura analógica do pino em que
Serial.print(" valor lido : ");
Serial.println(valPot);
delay(1000);
}
Done compiling.
Sketch uses 1900 bytes (5%) of program storage space. Maximum is 32256
Global variables use 202 bytes (9%) of dynamic memory, leaving 1846 by
2 Arduino/Genuino Uno on COM10
```

FIGURA 53 - CÓDIGO EM C PARA TESTE DO POTENCIÔMETRO

## ANEXO C – Planificação da aula

Aula Prática de Programação usando Robótica Pedagógica				Tempo: 120 minutos
Objetivos de aprendizagem:				Extra:
<ul style="list-style-type: none"> <li>• Elaborar pseudo-código / fluxograma</li> <li>• Compreender conceitos de programação:             <ol style="list-style-type: none"> <li>1. Listar os comandos disponíveis</li> <li>2. Identificar sensores</li> </ol> </li> <li>• Desenvolver programas</li> </ul>				
<b>Duração</b>	<b>Atividades</b>	<b>Papel do Professor / Papel do Aluno</b>	<b>Avaliação / Produtos a Desenvolver</b>	<b>Recursos</b>
5 min	Organização Dos Alunos	Cada aluno deve ter um computador para trabalhar		Computador Internet
3 min	Apresentação do Professor	O professor apresenta-se à turma e relembra o motivo da sua presença.		
5 min	Apresentação do desafio 1	O professor apresenta o primeiro desafio.		
15 min	Apresentação da temática	O professor projeta um <i>power point</i> sobre o Arduino e o modo como se programa.		Computador Vídeo Projetor
10 min	Desafio 1 – parte 1	O professor fornecerá uma folha de papel em que os alunos	A nível de conhecimentos e competências: avaliar o	Papel

		elaboram um pseudo-código / fluxograma.	conteúdo das folhas	
2 min	Recolha do desafio 1 – parte 1	O professor recolhe as folhas de papel		
20 min	Desafio 1 – parte 2	Os alunos devem iniciar a programação do Arduino de modo a que o desafio 1, parte 2 seja cumprido.		Computador
25 min	Experimentação do desafio 1 – parte 2	Será fornecido a cada um dos alunos um módulo para testar a sua programação. O professor apoia os alunos de modo a que consigam superar o desafio 1 – parte 2	A nível de conhecimentos e competências: avaliar se o desafio foi superado	Computador Arduino <i>Kit de led's</i> e sensores
10 min	Desafio 1 – parte 3	Para cada um dos alunos que superaram o desafio 1 – parte 2 é apresentado a parte 3 do desafio		
20 min	Experimentação do desafio 1 – parte 3	Apresentação por parte dos alunos da 3ª parte do desafio 1	A nível de conhecimentos e competências: avaliar se o	Computador Arduino <i>Kit de led's</i> e sensores

			desafio foi superado	
2 min	Questionário sobre o Desafio 1	Disponibilização no <i>e-learning(moodle)</i> do questionário 1		Questionário 1 Desafio 2 <i>online</i> Computador Internet
3 min	Questionário de cariz investigativo	Disponibilização no <i>e-learning(moodle)</i> do questionário 2		Questionário 2 <i>online</i> Computador Internet

## ANEXO D – Resultado do 1º desafio

Carros { Red ①  
Yellow ②  
Green ③

Pessoas { Red ④  
Green ⑤

3 Leds 2 Leds  
Carros Pessoas

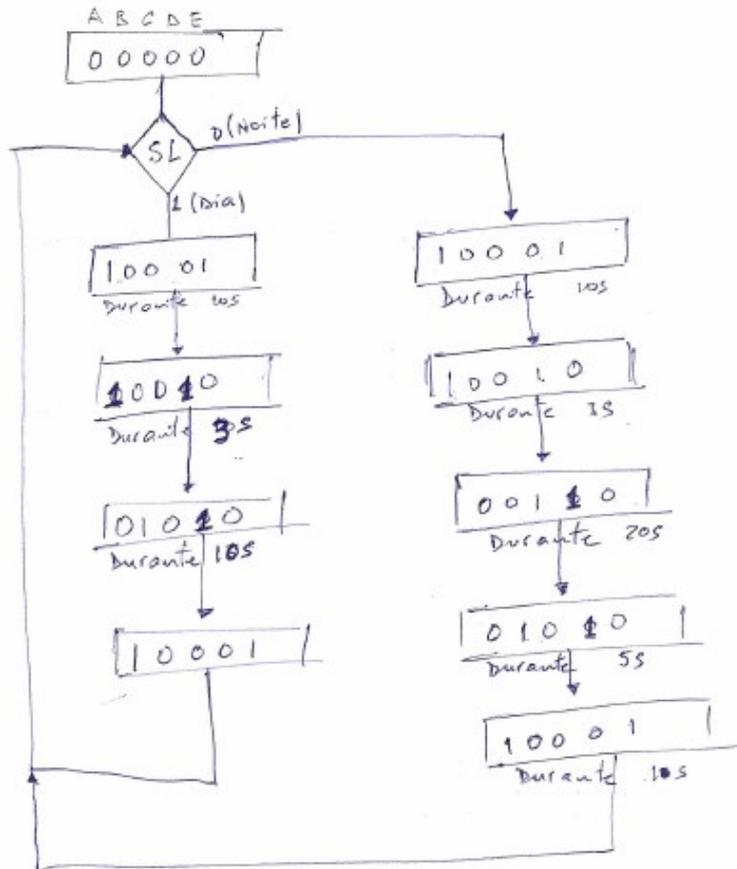
Sensor { Luz ⑥

Se o ① estiver ON, ② e ③ estão OFF e ⑤ está ON, ~~④~~ e ④ Off. Se ③ estiver ON, ⑤ está off e ④ está ON, ① e ② estão Off. Se ⑥ estiver ON, ③ ficará ligado 2x (Time quando ⑥ estiver Off.) antes de ③ ficar ON, ① fica OFF e ② fica ON

```
int ledVermelho = 13; ledAmarelo = 11; ledVerde = 9; ledVermelhoP = 7;
    ledVerdeP = 5; sensorLuz = 0;
    setup {
```

Entradas

- + Sensor luminosidade
- + Potenciômetro
- + sensor de temperatura



## ANEXO E – Resultado dos questionários

### Questionário 1

#### Questão 1 – Idade dos Participantes (Figura 54)

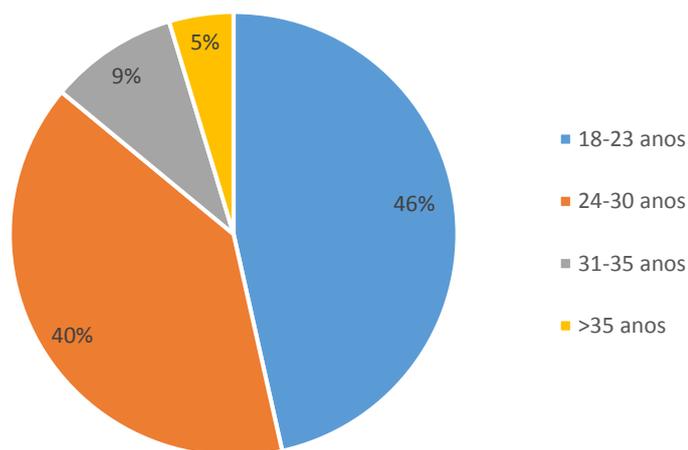


FIGURA 54 - IDADE DOS PARTICIPANTES

#### Questão 2 – Sexo (Figura 55)

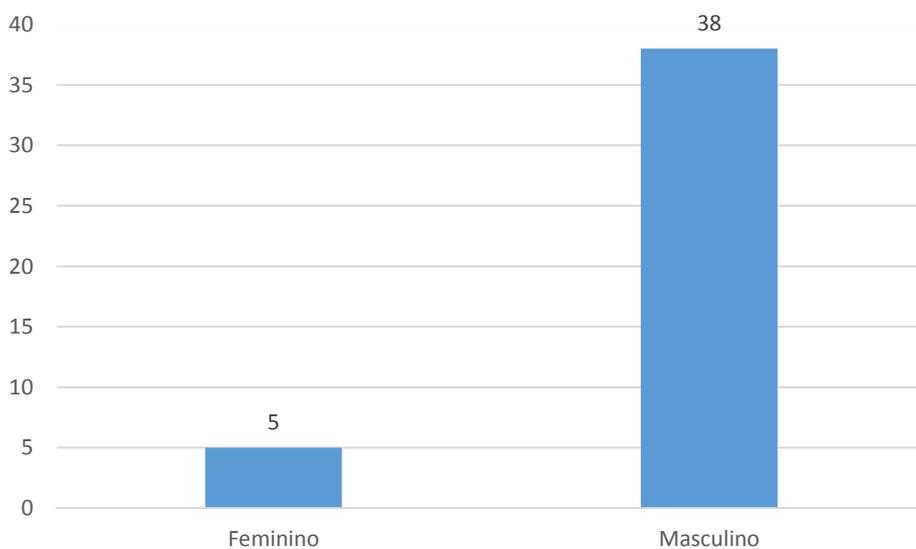


FIGURA 55 - SEXO DOS PARTICIPANTES

Questão 3 - Sentiu dificuldades na compreensão da tarefa proposta (Figura 56)?

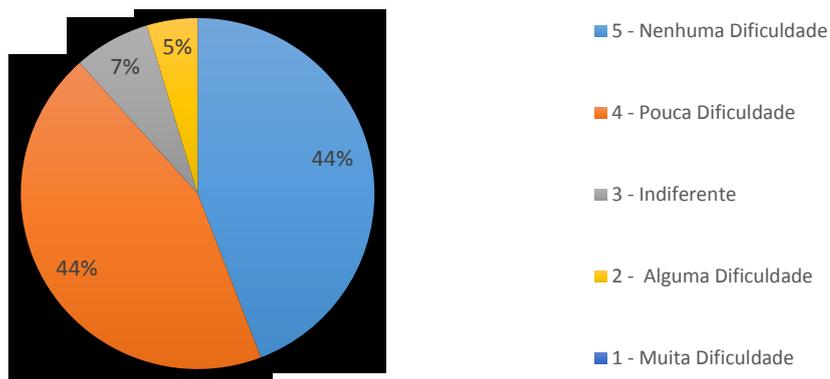


FIGURA 56 - DIFICULDADES NA COMPREENSÃO DA TAREFA

Questão 4 – O objetivo da tarefa foi entendido (Figura 57)?

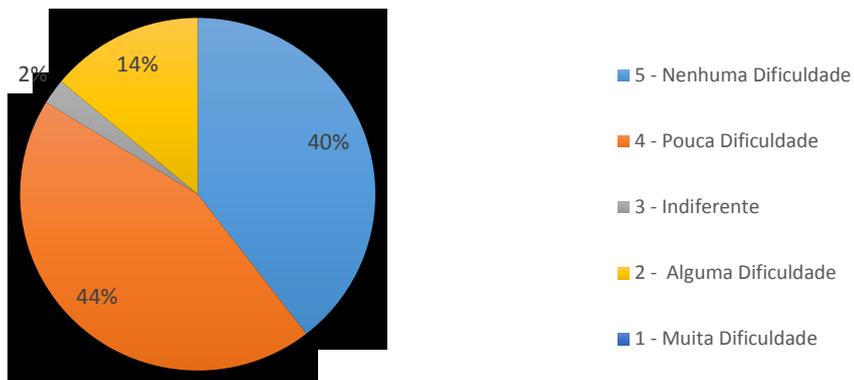


FIGURA 57 - OBJETIVO DA TAREFA

Questão 5 – Conseguiu compreender os conceitos de programação (Figura 58)?

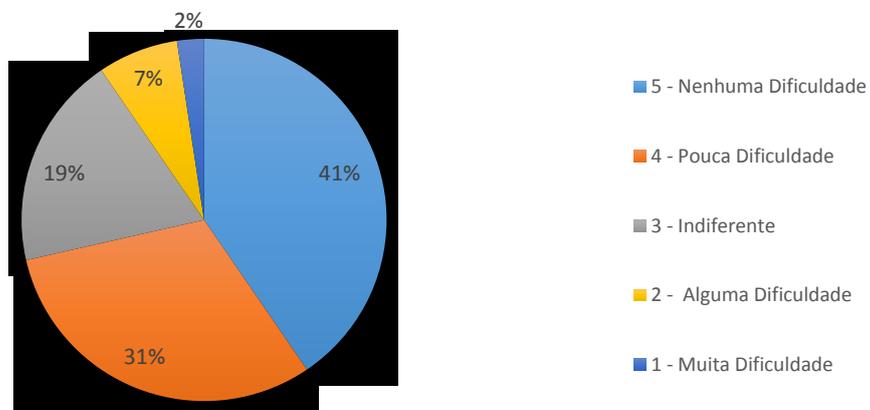


FIGURA 58 - COMPREENSÃO DOS CONCEITOS DE PROGRAMAÇÃO

Questão 6 – Sentiu dificuldades em compreender as respostas (feedback) das ações programadas (Figura 59)?

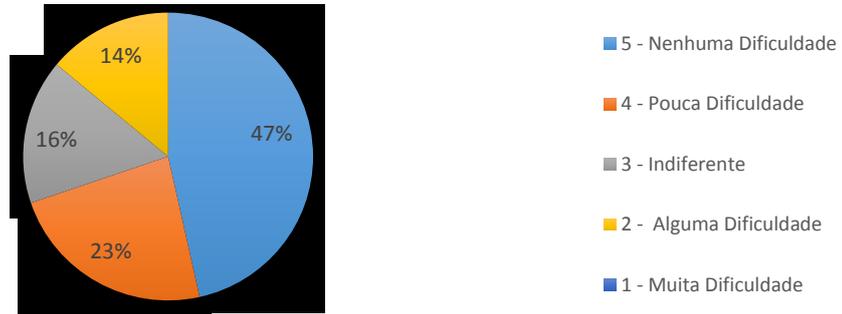


FIGURA 59 - COMPREENSÃO DAS RESPOSTAS DAS AÇÕES PROGRAMADAS

Questão 7 – A atividade prática contribuiu para uma melhor compreensão de novos conceitos de programação (Figura 60)?

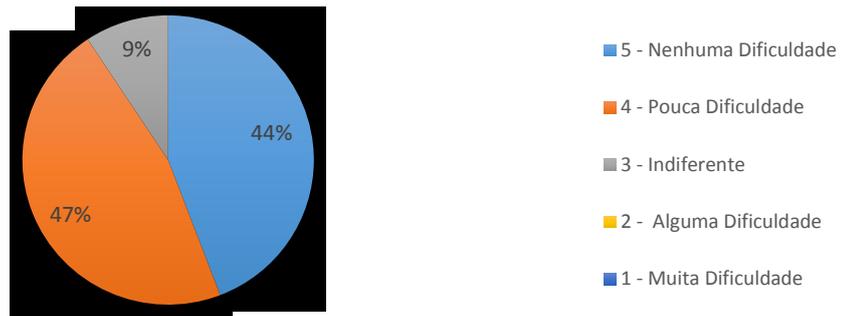


FIGURA 60 - COMPREENSÃO DE NOVOS CONCEITOS DE PROGRAMAÇÃO

Questão 8 – Os conceitos de programação foram adequados à categoria/tipo de tarefa (Figura 61)?

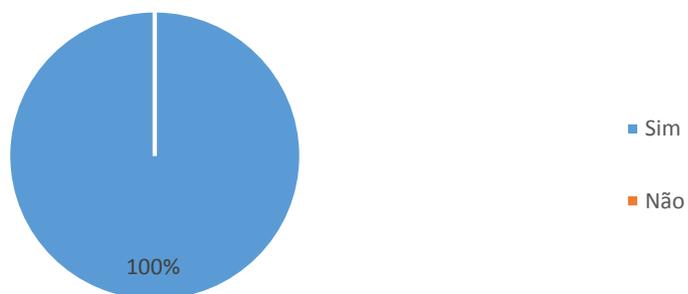


Figura 61 - Adequação da tarefa

Questão 9 – A Robótica Pedagógica é um instrumento interessante para ensino de programação (Figura 62)?

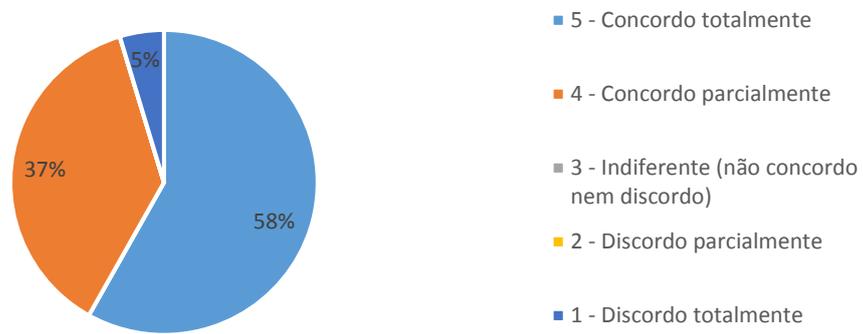


FIGURA 62 - MODELO INTERESSANTE PARA A PROGRAMAÇÃO

Questão 10 – A utilização do *hardware* ocorreu sem problemas ou dificuldades (Figura 63)?

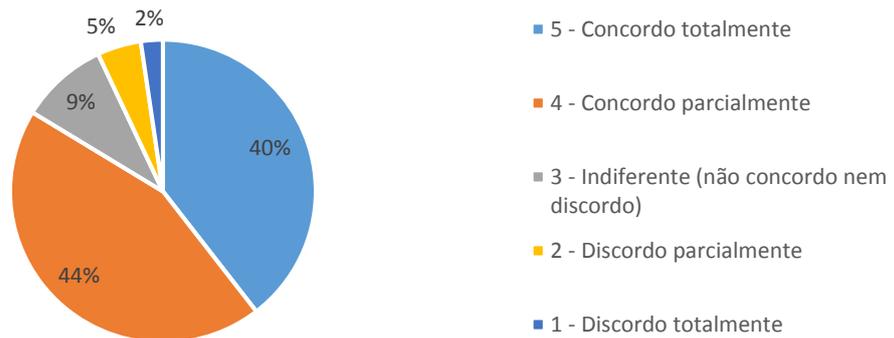


FIGURA 63 - DIFICULDADE EM UTILIZAR O *HARDWARE*

Questão 11 – Os recursos (*led's*, sensores, etc.) foram percebidos de maneira clara (Figura 64)?

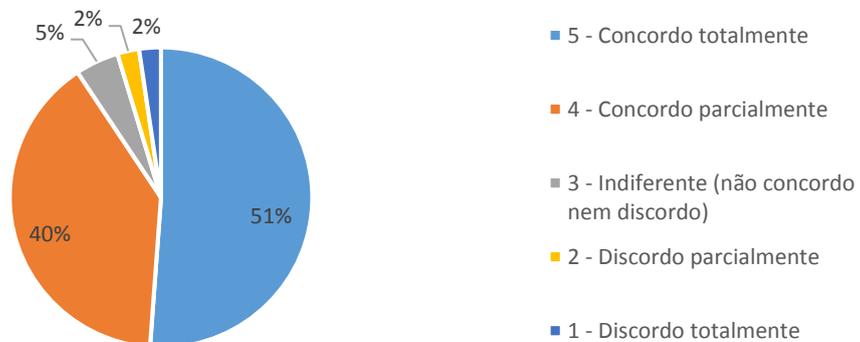


FIGURA 64 - COMPREENSÃO DOS RECURSOS DISPONIBILIZADOS

Questão 12 – As respostas dadas pelo *hardware* à programação executada foram claras e coerentes (Figura 65)?

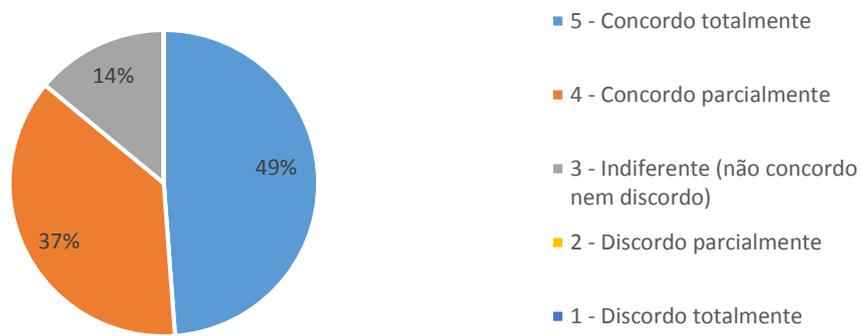


FIGURA 65 - RESPOSTAS DADAS PELO *HARDWARE*

Questão 13 – A utilização da Robótica Pedagógica auxiliou na aprendizagem ou aumentou seu conhecimento nos conceitos de programação (Figura 66)?

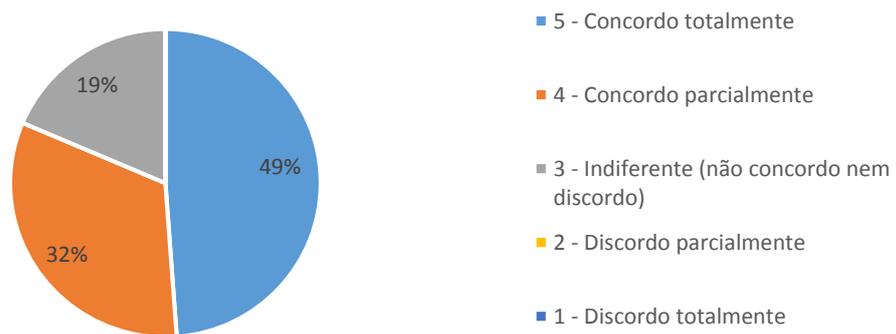


FIGURA 66 - UTILIZAÇÃO DA RP AUMENTOU O CONHECIMENTO EM PROGRAMAÇÃO

Questão 14 – A utilização do “IDE” ocorreu sem problemas ( Figura 67)?

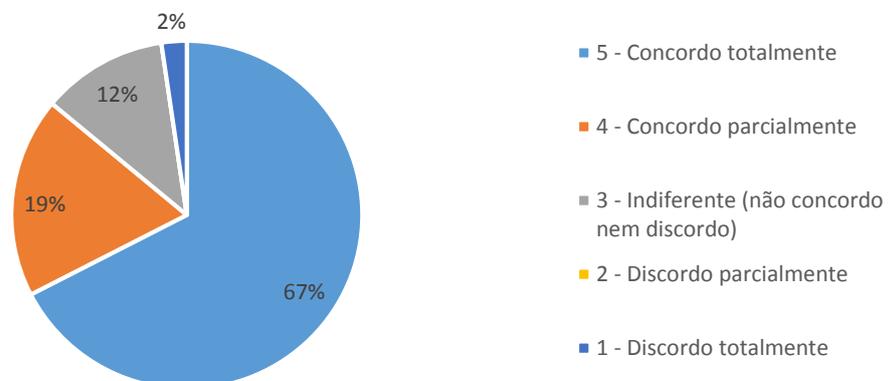


FIGURA 67 - UTILIZAÇÃO DO *IDE*

Questão 15 – Houve dificuldades em aplicar os conceitos de Lógica de Programação (Figura 68)?

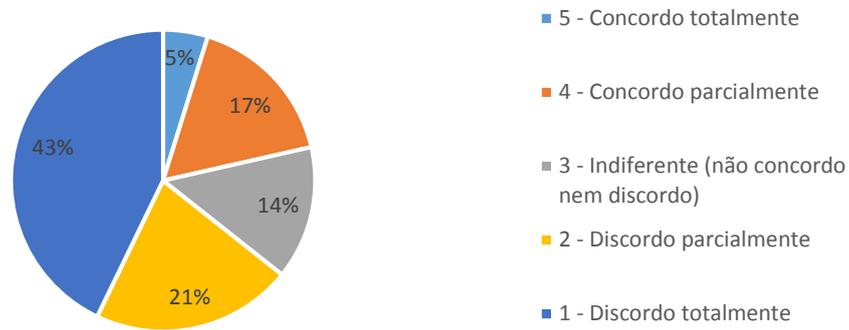


FIGURA 68 - DIFICULDADE NOS CONCEITOS DE PROGRAMAÇÃO

Questão 16 – Houve problema em compreender os objetivos da prática proposta (Figura 69)?

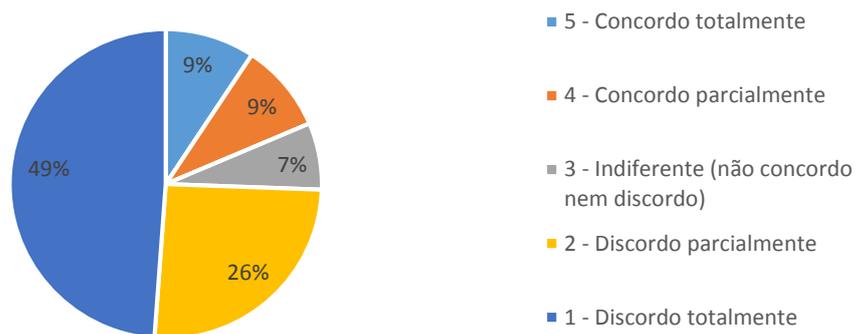


FIGURA 69 - COMPREENSÃO DOS OBJETIVOS DA TAREFA

Questão 17 – Teve dificuldade em elaborar a solução da prática proposta (Figura 70)?

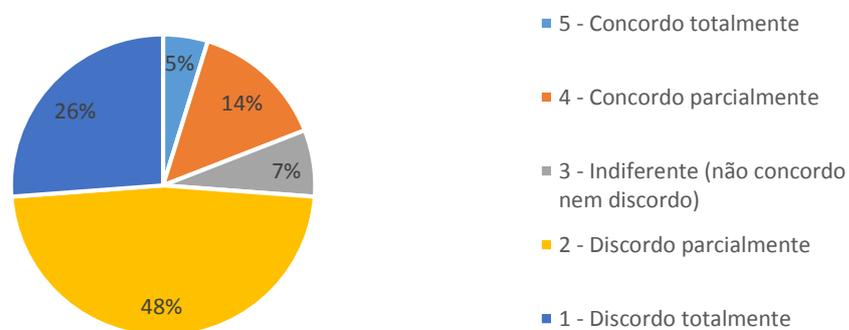


FIGURA 70 - DIFICULDADE EM ELABORAR A TAREFA

Questão 18 – Houve dificuldades em aplicar os conceitos de Lógica de Programação (estruturas de decisão, estruturas de repetição etc.) na solução (Figura 71)?

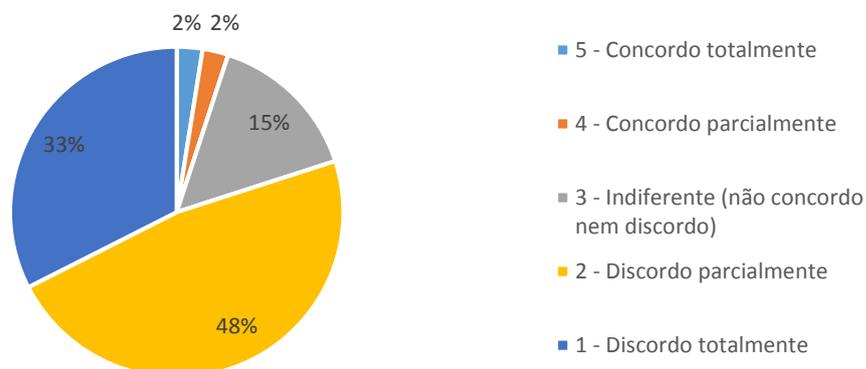


FIGURA 71 - DIFICULDADE EM APLICAR OS CONCEITOS DE LÓGICA DE PROGRAMAÇÃO

Questão 19 - Como avalia a atividade realizada (Figura 72)?

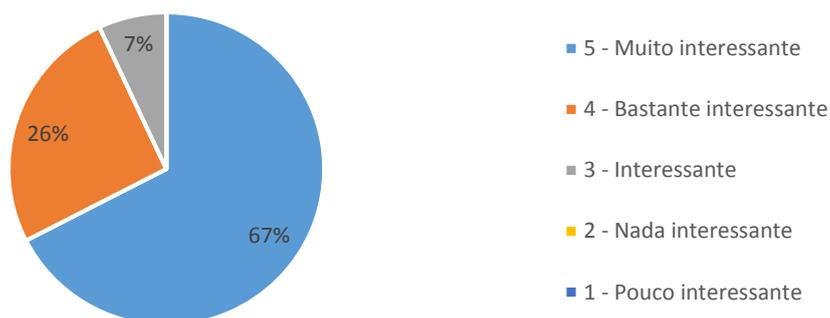


FIGURA 72 - AVALIAÇÃO DA ATIVIDADE

Resultados por inquérito do questionário 2

Abv.	Perguntas	Respostas: 1 - Discordo Totalmente ... 7 - Concordo Totalmente																																										
E1	Considero o uso do Arduino para a programação divertido.	7	6	4	7	7	7	7	7	7	7	6	7	7	7	7	6	4	7	7	7	7	5	7	6	7	7	7	5	7	6	7	7	7	5	4	7	7	6	7	5	7	5	
E2	Considero o uso de Arduino na programação agradável	7	5	4	7	7	7	7	7	7	6	7	7	6	7	5	4	7	7	7	7	5	7	6	7	7	6	7	7	6	7	7	6	7	4	7	7	5	6	7	7	7		
IU1	Tenciono continuar a programar usando Arduino	5	1	2	7	4	7	5	7	7	6	4	6	2	4	5	5	2	7	4	7	5	7	7	6	4	6	5	6	7	6	6	6	2	5	5	2	7	4	1	4	7	7	1
IU2	Dado que tenho acesso ao Arduino, planeio usá-lo	6	1	1	7	4	7	6	7	7	6	4	7	2	4	6	1	1	7	4	7	6	2	7	6	4	2	2	4	7	6	4	7	2	4	1	1	7	4	1	1	7	6	1
VSE	Atualmente considero-me um utilizador de Arduino frequente	1	1	1	7	1	1	1	2	7	1	4	4	1	2	1	1	1	7	1	1	1	2	7	1	4	4	1	2	7	1	4	4	1	2	1	1	2	1	1	2	7	7	2
PU1	O uso de Arduino melhora o meu desempenho na programação	4	1	1	7	2	2	3	7	6	2	4	7	3	5	4	1	1	6	2	2	3	7	6	2	2	6	3	5	6	2	3	7	3	6	6	1	6	2	1	5	6	6	6
PU2	O uso do Arduino melhora a minha produtividade	4	1	1	7	2	2	4	7	7	2	3	7	3	3	4	4	1	7	2	2	4	6	7	2	3	7	5	3	7	2	5	7	3	5	5	4	7	2	5	3	6	5	5
PU3	O uso de Arduino torna-me eficaz na programação	4	5	1	7	2	2	4	5	7	2	3	7	3	5	4	6	1	7	2	2	4	6	7	6	3	7	3	5	7	6	3	7	3	4	6	1	7	7	4	6	7	7	5
PU4	Considero útil o uso de Arduino na programação	7	4	4	7	5	3	7	7	7	2	5	7	3	6	3	4	4	3	5	3	2	1	1	2	5	2	5	5	7	2	5	7	3	5	2	4	2	5	4	5	1	7	2
EOV1	Aprender programação é fácil	5	4	3	6	5	6	4	5	5	1	4	5	7	2	7	4	6	6	7	6	6	7	7	1	4	7	7	2	7	1	4	7	7	2	7	7	6	7	6	2	7	7	7
EOV2	Considero que é fácil fazer o que pretendo com o uso do Arduino em programação	4	4	3	7	5	6	6	3	7	5	5	7	4	2	1	1	3	7	5	6	6	3	7	5	5	3	5	2	1	5	5	5	3	2	5	3	6	5	5	2	5	6	5
EOV3	Para mim será fácil ser um utilizador avançado em Arduino	4	4	2	7	6	5	4	5	6	2	5	6	3	4	4	4	3	7	6	5	5	5	6	3	7	6	3	5	7	4	5	6	3	5	7	4	7	6	7	5	7	7	7
EOV4	É fácil usar o Arduino	4	5	3	7	6	4	5	5	7	6	5	7	3	4	4	5	3	7	6	4	2	5	7	6	5	7	3	4	7	6	5	7	3	4	4	2	7	6	6	4	7	7	7

## Resultados totais do questionário 2

Abv.	Pergunta	1-Discordo Totalmente	2	3	4	5	6	7-Concordo Totalmente
E1	Considero o uso do Arduino para a programação divertido	0	0	0	3	5	6	29
E2	Considero o uso de Arduino na programação agradável	0	0	0	3	4	7	29
IU1	Tenciono continuar a programar usando Arduino	3	5	0	7	8	8	12
IU2	Dado que tenho acesso ao Arduino, planeio usá-lo	9	5	0	9	0	8	12
VSE	Atualmente considero-me um utilizador de Arduino frequente	22	8	0	6	0	0	7
PU1	O uso de Arduino melhora o meu desempenho na programação	6	9	6	3	3	11	5
PU2	O uso do Arduino melhora a minha produtividade	3	8	7	6	7	2	10
PU3	O uso do Arduino torna-me eficaz na programação	3	5	6	6	5	6	12
PU4	Considero útil o uso de Arduino na programação	3	8	6	6	10	1	9
EOV1	Aprender programação é fácil	3	4	1	6	5	8	16
EOV2	Considero que é fácil fazer o que pretendo com o uso do Arduino em programação	3	4	7	3	15	6	5
EOV3	Para mim será fácil ser um utilizador avançado em Arduino	0	2	5	8	10	8	10
EOV4	É fácil usar o Arduino	0	2	5	9	8	7	12

<b>PLS - Quality Criteria</b>						
	<b>AVE</b>	<b>Composite Reliability</b>	<b>R Square</b>	<b>Cronbachs Alpha</b>	<b>Communality</b>	<b>Redundancy</b>
<b>Enjoy</b>	0.841841	0.914022		0.819461	0.841841	
<b>IU</b>	0.887418	0.940348	0.438144	0.873611	0.887418	0.255249
<b>PEOU</b>	0.820770	0.901563		0.781675	0.820770	
<b>PU</b>	0.817375	0.930655	0.509805	0.888092	0.817375	0.078269
<b>Use</b>	1.000000	1.000000	0.535327	1.000000	1.000000	0.224205

	<b>redundancy</b>	<b>R Square</b>	<b>Cronbachs Alpha</b>
<b>Enjoy</b>		<b>Enjoy</b>	<b>Enjoy</b>
<b>IU</b>	0.255249	<b>IU</b>	<b>IU</b>
<b>PEOU</b>		<b>PEOU</b>	<b>PEOU</b>
<b>PU</b>	0.078269	<b>PU</b>	<b>PU</b>
<b>Use</b>	0.224205	<b>Use</b>	<b>Use</b>

<b>Latent Variable Correlations</b>					
	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>Enjoy</b>	1.000000				
<b>IU</b>	0.553564	1.000000			
<b>PEOU</b>	0.276420	0.448073	1.000000		
<b>PU</b>	0.357522	0.536844	0.692793	1.000000	
<b>Use</b>	0.061139	0.535415	0.685702	0.6930	1.000000

<b>Cross Loadings</b>					
	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>E1</b>	0.882946	0.426700	0.088102	0.195356	-0.083768
<b>E2</b>	0.950836	0.567292	0.365712	0.419314	0.149003
<b>IU1</b>	0.515649	0.949362	0.427288	0.586125	0.524836
<b>IU2</b>	0.528570	0.934638	0.416630	0.415210	0.481706
<b>PEOU3</b>	0.194314	0.325435	0.903116	0.628782	0.601820
<b>PEOU4</b>	0.304967	0.484190	0.908802	0.626590	0.640122
<b>PU1</b>	0.388179	0.510283	0.565141	0.924196	0.625295
<b>PU2</b>	0.232382	0.482940	0.619129	0.910451	0.667706
<b>PU3</b>	0.345734	0.462940	0.688837	0.876963	0.587944
<b>USE</b>	0.061139	0.535415	0.685702	0.693030	1.000000

	<b>AVE</b>	<b>Communality</b>	<b>Composite Reliability</b>
	<b>AVE</b>	<b>Communality</b>	<b>Composite Reliability</b>
<b>Enjoy</b>	0.841841	<b>Enjoy</b>	<b>Enjoy</b>
<b>IU</b>	0.887418	<b>IU</b>	<b>IU</b>
<b>PEOU</b>	0.820770	<b>PEOU</b>	<b>PEOU</b>
<b>PU</b>	0.817375	<b>PU</b>	<b>PU</b>
<b>Use</b>	1.000000	<b>Use</b>	<b>Use</b>

<b>Total Effects</b>					
	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>Enjoy</b>		0.484483		0.179756	0.138314
<b>IU</b>					0.285487

<b>PEOU</b>	0.249913	0.643105	0.629130
<b>PU</b>	0.388605		0.110942
<b>Use</b>			

**Outer Loadings (Weights or Loadings)**

	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>E1</b>	0.882946				
<b>E2</b>	0.950836				
<b>IU1</b>		0.949362			
<b>IU2</b>		0.934638			
<b>PEOU3</b>			0.903116		
<b>PEOU4</b>			0.908802		
<b>PU1</b>				0.924196	
<b>PU2</b>				0.910451	
<b>PU3</b>				0.876963	
<b>USE</b>					1.000000

**Path Coefficients**

	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>Enjoy</b>		0.414630		0.179756	
<b>IU</b>					0.285487
<b>PEOU</b>				0.643105	0.557783
<b>PU</b>		0.388605			
<b>Use</b>					

**Outer Weights**

	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>E1</b>	0.430230				
<b>E2</b>	0.652195				
<b>IU1</b>		0.563333			
<b>IU2</b>		0.497726			
<b>PEOU3</b>			0.543943		
<b>PEOU4</b>			0.559810		
<b>PU1</b>				0.362447	
<b>PU2</b>				0.358365	
<b>PU3</b>				0.386282	
<b>USE</b>					1.000000

**Index Values**

**Measurement Model (restandardised)**

	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>E1</b>	0.933544				
<b>E2</b>	1.030556				
<b>IU1</b>		0.498578			
<b>IU2</b>		0.399150			
<b>PEOU3</b>			0.612033		
<b>PEOU4</b>			0.592506		
<b>PU1</b>				0.442107	
<b>PU2</b>				0.459090	
<b>PU3</b>				0.438778	
<b>USE</b>					0.456363

<b>Path Coefficients</b>					
	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>Enjoy</b>		0.907150		0.263481	
<b>IU</b>					0.561592
<b>PEOU</b>				0.578104	1.472230
<b>PU</b>		0.580043			
<b>Use</b>					

<b>Measurement Model</b>					
	<b>Enjoy</b>	<b>IU</b>	<b>PEOU</b>	<b>PU</b>	<b>Use</b>
<b>E1</b>	0.475304				
<b>E2</b>	0.524696				
<b>IU1</b>		0.555378			
<b>IU2</b>		0.444622			
<b>PEOU3</b>			0.508106		
<b>PEOU4</b>			0.491894		
<b>PU1</b>				0.329936	
<b>PU2</b>				0.342611	
<b>PU3</b>				0.327452	
<b>USE</b>					1.000000

### Index Values for Latent Variables

	<b>LV Index Values</b>
<b>Enjoy</b>	6.430807
<b>IU</b>	4.684647
<b>PEOU</b>	5.115902
<b>PU</b>	4.269588
<b>Use</b>	2.581395

# ANEXO F – Apresentação da aula



Instituto Universitário de Lisboa

Mestrado em Engenharia Informática

Unidade Curricular: Dissertação em Engenharia Informática (2016/2017)

Docente: Prof. Doutor Carlos Costa

## Ensino da Programação Usando Robótica Pedagógica

67573 – Sérgio Pereira



Instituto Universitário de Lisboa

### Introdução

- ▶ Entre os grandes desafios no ensino de programação de computadores estão a complexidade dos conceitos e o alto nível de abstração.
- ▶ As metodologias tradicionais utilizadas para ensinar/aprender revelam-se insuficientes, pelo que é apresentada uma nova proposta, centrada na atividade de resolução de problemas.

## Desafio

- ▶ **Parte Um**
- ▶ Elaboração de um fluxograma / organograma / pseudo-linguagem em blocos
- ▶ **Parte Dois**
- ▶ Programação do Arduino de modo a simularem um semáforo (os tempos entre as mudanças dos led's são atribuídas pelos alunos)
- ▶ **Parte Três**
- ▶ Utilizando o sensor de luz que se encontra no kit deverá simular o seguinte:
- ▶ Se não houver luz (sensor de luz tapado) os led's vermelhos e verdes deverão estar desligados e o led amarelo deve estar a piscar (o intervalo de tempo que o led pisca é atribuído pelo aluno)
- ▶ Se houver luz (sensor de luz descoberto) os led's simularão o semáforo (Parte dois)

3

## Proposta: Robótica Pedagógica

- ▶ Está presente em diversas instituições de ensino e pesquisa, sendo usada como método para criar um ambiente de ensino mais atraente e diversificado.
- ▶ Proporciona montagem, automação e controle de dispositivos mecânicos.
- ▶ Assim, os alunos podem trabalhar a ligação de peças mecânicas e de componentes.
- ▶ É possível observar o impacto de instruções num objeto concreto e associar os comandos abstratos aos movimentos e ações do mesmo.

4

## Algumas soluções existentes



5

## Arduino : O que é

- ▶ É uma plataforma de desenvolvimento open source que utiliza microcontroladores e é composta por:
  - Placa de circuito impresso com um microcontrolador
  - IDE de desenvolvimento multi-plataforma.



O código do ambiente Java (IDE) é disponibilizado sob a licença GPL, as bibliotecas microcontroladoras C/C++ sob LGPL, e os esquemas e arquivos CAD sob Creative Commons Attribution Share-Alike.

6

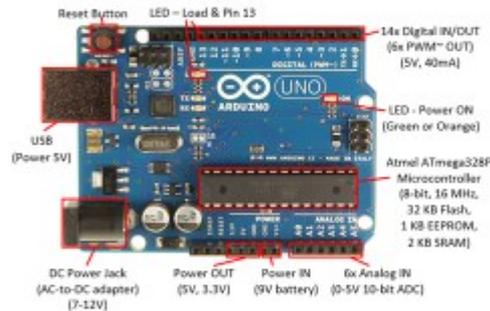
## Arduino : Versões



7

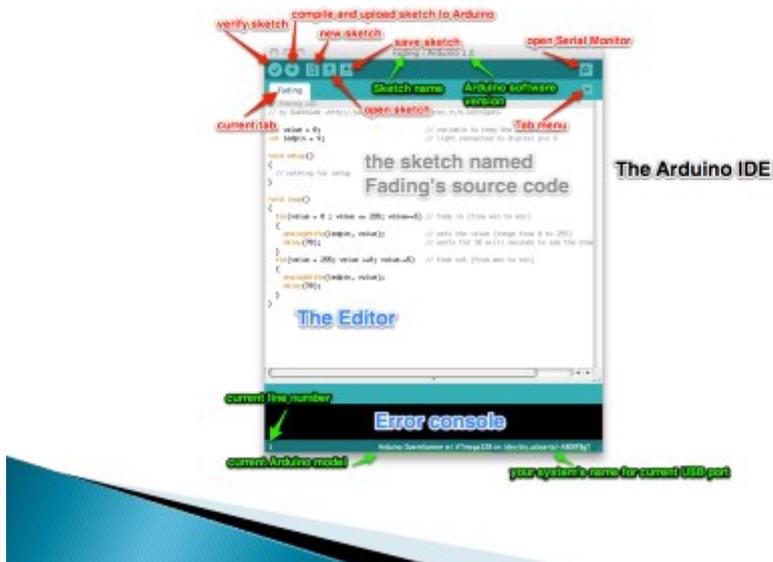
## Arduino : Características Principais

- Microcontrolador : Amtel AVR
- Tensão de operação : 5V
- Tensão de Entrada : 7-12V
- Entradas/Saídas Digitais : 14 (Entradas/Saídas PWM : 6)
- Entradas Analógicas : 6
- Botão de Reset
- Porta de Ligação USB (para Comunicação com PC)
- Monitor Serial



8

## Arduino : IDE



9

## Arduino : Software

- ▶ O software inclui um editor de código, sendo capaz de compilar e carregar programas para a placa com um único clique. Com isso não há a necessidade de editar Makefiles ou executar programas em ambientes de linha de comando.
- ▶ Possui a capacidade de programar em C/C++, assim permite criar com facilidade muitas operações de entrada e saída, tendo que definir apenas duas funções no pedido para fazer um programa funcional:
- ▶ **setup()** – Inserida no início, na qual pode ser usada para inicializar configuração;
- ▶ **loop()** – Chamada para repetir um bloco de comandos ou esperar até que seja desligada.

10

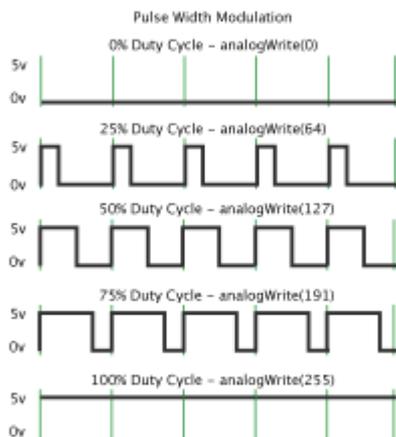
## Arduino : Portas Digitais

- › Nas portas digitais (leitura e/ou escrita) os valores são HIGH e LOW
- › Nas portas digitais que permitem PWM (**Pulse Width Modulation**) pode-se simular diferentes valores para se comportarem como portas analógicas.

PWM, é uma técnica utilizada para a variação do valor médio de uma forma de onda periódica. A técnica consiste em manter a frequência de uma onda quadrada fixa e variar o tempo que o sinal fica em nível lógico alto. Esse tempo é chamado de duty cycle, ou seja, o ciclo ativo da forma de onda.

11

## Arduino : Portas PWM



A frequência da onda tem o mesmo valor e o que varia é duty cycle. Quando o duty cycle está em 0% o valor médio da saída encontra-se em 0 V e para um duty cycle de 100% a saída assume seu valor máximo, que no caso é 5V. Para um duty cycle de 50% a saída assumirá 50% do valor da tensão, 2,5 V e assim sucessivamente para cada variação no duty cycle. Para calcular o valor médio da tensão de saída de um sinal PWM pode-se utilizar a seguinte equação:

$$V_{out} = (\text{duty cycle}/100) * V_{cc}$$

Onde:

- Vout - tensão de saída em V;
- duty cycle - valor do ciclo ativo do PWM em %;
- Vcc - tensão de alimentação em V.

12

## Arduino : Componentes do código

O código no Arduino é constituído por 5 elementos principais:

- ▶ –As **bibliotecas**, que vêm no início do programa, e onde são incluídas todas as bibliotecas as quais são normalmente constituídas por funções pré-feitas por outros programadores ou até mesmo pelos criadores da plataforma Arduino.
- ▶ –As **Variáveis**, as quais são definidas após as livrarias, servindo para guardas informações, como por exemplo o valor de uma velocidade, o resultado de um cálculo, etc...
- ▶ –As **funções**, não sendo completamente necessário que sejam colocadas abaixo das variáveis, mas para seguir um raciocínio linear vamos admitir que são colocadas abaixo das variáveis. Estas funções já dizem respeito a funções criadas por nós e não às incluídas nas bibliotecas.
- ▶ –void **setup()**, este é normalmente usado para inicializar variáveis, modos de pinos, definição a velocidade de transmissão serial, etc...
- ▶ –void **loop()**, aqui é a parte do código em que se faz um loop, ou seja, cria-se um ciclo de repetição para que as instruções dentro do mesmo sejam repetidas.

13

## Componentes do código : Biblioteca

- ▶ Quando se cria um código/programa usando uma linguagem de programação, neste caso C/C++, existe a possibilidade de usar um conjunto de funções pré-criadas. A esse conjunto de funções damos o nome de bibliotecas.
- ▶ – Wifi: Esta é utilizada quando temos um placa wireless ligada ao Arduino, assim de uma forma intuitiva, consegue-se ligar a uma rede wireless sabendo o seu SSID.
  - Na prática, para se utilizar uma biblioteca basta inclui-la no início do código, por exemplo, para usar a biblioteca servo deve ter uma linha no início do código escrita com o formato apresentado na imagem à direita:



```

include_Wifi | Arduino
File Edit Sketch Tools Help
include_Wifi
#include <WiFi.h>

char ssid[] = "MinhaRede";
int status = WL_IDLE_STATUS; // the
void setup() {
  // put your setup code here, to run once
  Serial.begin(9600); //verifica a existencia
  status = WiFi.begin(ssid); //liga à rede
}

void loop() {
  // put your main code here, to run repeatedly
}
  
```

14

## Componentes do código : Variáveis

- ▶ Uma variável é uma forma de dar um nome e guardar um valor para usar ao longo do programa, como por exemplo, dados de um sensor ou um valor intermediário de um cálculo.
- ▶ Antes de serem utilizadas, as variáveis devem ser declaradas. Declarar uma variável significa definir o seu tipo, e opcionalmente, configurar um valor inicial (iniciar a variável). As variáveis não precisam ser iniciadas quando são declaradas.
- ▶ Existem diversos tipos de variáveis, umas onde apenas se podem guardar números e outras onde pode ser guardado qualquer tipo de carater, seja ele um número, uma letra ou um símbolo (int,double,float, char).



```

sketch
File Edit Sketch Tools Help
sketch_oct24a$
int variavel1;
int variavel12 = 0;
float var2 = -0.0489;
double var3 = 0.478009;
char var5[20] = "ISCTE";

void setup() {

}

void loop() {
    }
    
```

15

## Componentes do código : Funções

- ▶ As funções são usadas para criar pequenos pedaços de códigos separados do programa principal. No caso do Arduino estas são colocadas fora do loop().
- ▶ Qual a vantagem de fazermos uma função?
- ▶ As funções são importantes pois devolvem valores, ajudam a fragmentar o código em partes menores e podem ser utilizadas mais de uma vez no mesmo programa, poupando tempo de programação e inúmeras linhas de código que fazem o mesmo.
- ▶ Na imagem à direita podemos ver como se cria uma função e como é utilizada. A função em questão soma dois valores e depois devolve o resultado ao programa principal.



```

sketch_oct28a | Arduino 1.7.
File Edit Sketch Tools Help
sketch_oct28a$
int a, b, c;
int soma (int x, int y) {
    return(x+y);
}
void setup() {
    // put your setup code here, to
    ...
}
void loop() {
    // put your main code here, to
    ...
    a=2;
    b=7;
    c = soma(a,b);
    Serial.print(c);
    ...
}
    
```

16

## Componentes do código : setup e loop

- ▶ O "void setup ()" é normalmente usado para inicializar variáveis , modos de pinos , definição a velocidade de transmissão serial, etc.
- ▶ O "void loop ()" é a parte do código em que se faz um loop, ou seja, cria-se um ciclo de repetição para que as instruções dentro do mesmo sejam repetidas.

```
sketch_oct28a | Arduino 1.7.11
File Edit Sketch Tools Help
sketch_oct28a $
int a, b, c;
int soma (int x, int y){
  return(x+y);
}
void setup() {
  // put your setup code here, to run
  pinMode (13, OUTPUT);
  pinMode (12, INPUT);
}
void loop() {
  // put your main code here, to run
  ...
  a=2;
  b=7;
  c = soma(a,b);
  int t;
  Serial.print(c);
  digitalWrite (13, HIGH);
  digitalWrite (13, x);
  ...
}
```

17

## Condição : if then else

- ▶ O comando if é utilizado para executar um comando ou bloco de comandos no caso de uma determinada condição ser avaliada como verdadeira. Caso seja necessário executar um comando se a condição for falsa usa a clausula else com o comando if
  - if (condicao){
    - comandoA;
    - if (condicao2){
      - comandoB;
    - else {
      - comandoC;
  - else{
    - comandoD;

18

## Condição : while e do-while

O loop while executa-se enquanto se cumprir a condição, enquanto que o loop do-while executa-se ao menos uma vez e depois confirma se a condição se cumpre.

```

sketch_oct24a | Arduino
File Edit Sketch Tools Help
sketch_oct24a $
void setup() {
}
void loop() {
do{
digitalWrite(2,HIGH);
delay(1000);
digitalWrite(2,LOW);
delay(1000);
}while(vaz<200);
...
while(vaz < 200){
digitalWrite(2,HIGH);
delay(1000);
digitalWrite(2,LOW);
delay(1000);
...
}
}
19
    
```



## Condição : for

A instrução for é usada para repetir um bloco de instruções definido entre as chaves. Geralmente utiliza-se um contador para incrementar e finalizar o loop.

O header para o loop com for é composto por três partes:  
for (inicialização; condição;  
incremento)  
{ //instrução(ou instruções); }

```

dimmer_led | Arduino 1.7.11
File Edit Sketch Tools Help
dimmer_led
#define pinoPWM = 10 // define pino 10 como PWM
void setup()
{
// não é necessário setup
}
void loop()
{
int i;
int x = 1;
for (int i = 0; i < 10; i++){
digitalWrite(pinoPWM, i);
if (i==255) i=-1;
delay(50);
}
}
20
    
```

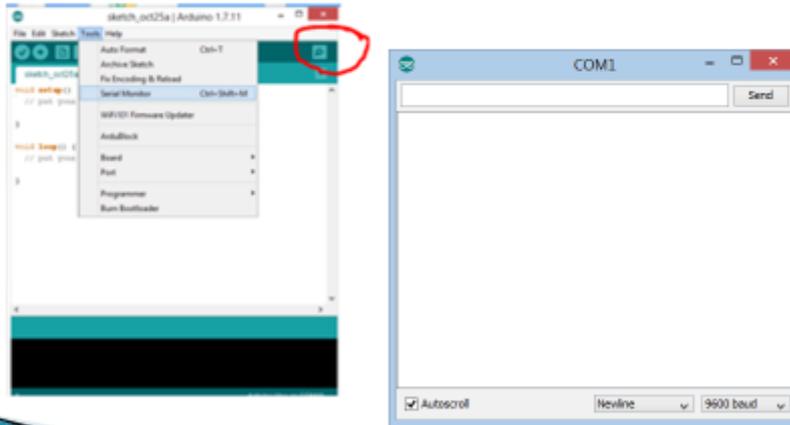
```

sketch_for | Arduino 1.7.11
File Edit Sketch Tools Help
sketch_for
void setup() {
int i;
Serial.println(0);
for (int i = 0; i < 10; i++){
Serial.println(i);
}
}
void loop() {
}
20
    
```



## Terminal Serial

O IDE tem um terminal serial que auxilia no envio e receção de dados para a placa.



21

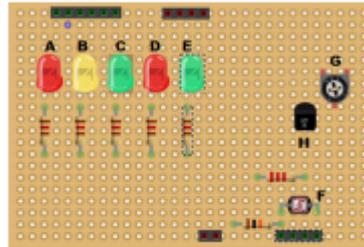
## Arduino : Ligação ao PC



22

## Hardware de Apoio

- ▶ Para o presente trabalho acrescentou-se um módulo de hardware de modo a facilitar a programação do Arduino, Este módulo é constituído por 5 led's, 1 sensor de luminosidade, 1 sensor de temperatura e 1 potenciómetro.
- ▶ As ligações ao Arduino são as seguintes:
  - A) led vermelho á porta digita 13
  - B) led amarelo á porta digital 11
  - C) led verde á porta digita 9
  - D) led vermelho á porta digital 7
  - E) led verde á porta digital 5
  - F) sensor de luz á porta analógica 0
  - G) potenciómetro á porta analógica 2
  - H) sensor de temperatura á porta analógica 3



23

## Primeiro programa

- ▶ Por a piscar o led ligado ao pino 13

```

Blink | Arduino 1.7.11
File Edit Sketch Tools Help
Sketch
void setup() {
  // inicializa o pino 13 como output
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED, HIGH); // liga o LED
  digitalWrite(LED, LOW);  // desliga o LED
}
    
```

```

sketch_oct28a | Arduino 1.7.11
File Edit Sketch Tools Help
sketch_oct28a$
void setup() {
  // put your setup code here, to run once
  pinMode (11, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly
  for(int i=0; i < 255; i++){
    analogWrite (11, i);
    delay(20);
  }
}
    
```

24

## Terminal Serial

Exibir números sequenciais no Monitor Serial:

```
sketch_oct25a | Ardu
File Edit Sketch Tools Help
sketch_oct25a $
void setup() {
  Serial.begin(9600); //Envia e recebe
  Serial.println("Escrever valores ");
}
int numero = 0;
void loop() {
  Serial.print("Valor: ");
  Serial.println(numero);
  delay(500);
  numero++;
}
```

Exibindo valores da 1 a 255 em vários sistemas:

```
sketch_oct25a | Arduino 1.7.11
File Edit Sketch Tools Help
sketch_oct25a $
Serial.begin(9600);
Serial.println("Exibindo valores em sistemas diversos");
}
int valor;
void loop() {
  for (valor = 0; valor <= 255; valor++){
    Serial.print("DEC: ");
    Serial.print(valor);
    Serial.print(" HEX: ");
    Serial.print(valor, HEX);
    Serial.print(" BIN: ");
    Serial.println(valor, BIN);
    if(valor == 255) {
      while(true) {
        continue;
      }
    }
  }
}
```

25

## Terminal Serial

Exibir números sequenciais no Monitor Serial:

```
sketch_oct25a | Ardu
File Edit Sketch Tools Help
sketch_oct25a $
void setup() {
  Serial.begin(9600); //Envia e recebe
  Serial.println("Escrever valores ");
}
int numero = 0;
void loop() {
  Serial.print("Valor: ");
  Serial.println(numero);
  delay(500);
  numero++;
}
```

Exibindo valores da 1 a 255 em vários sistemas:

```
sketch_oct25a | Arduino 1.7.11
File Edit Sketch Tools Help
sketch_oct25a $
Serial.begin(9600);
Serial.println("Exibindo valores em sistemas diversos");
}
int valor;
void loop() {
  for (valor = 0; valor <= 255; valor++){
    Serial.print("DEC: ");
    Serial.print(valor);
    Serial.print(" HEX: ");
    Serial.print(valor, HEX);
    Serial.print(" BIN: ");
    Serial.println(valor, BIN);
    if(valor == 255) {
      while(true) {
        continue;
      }
    }
  }
}
```

25

## Funções disponíveis :

```
void Serial.begin(baude_rate);
    baude_rate - velocidade de comunicação na porta Série (9600)
long Serial.print("texto");
    Apresenta "texto" na consola
long Serial.print(valor);
    Apresenta o valor na consola
long Serial.println("texto");
    Apresenta "texto" na consola e muda de linha
char Serial.read();
    lê um caracter da consola
boolean Serial.available();
    verifica se há um caracter na consola
Boolean isDigit(caracter);
    verifica se o caracter é um dígito (0..9)
```

27

## Desafio 1

- ▶ Simular um semáforo
  - Utilizar 2 led's vermelhos (ligados aos pinos 13 e 10)
  - Utilizar 1 led amarelo (ligado ao pino 12)
  - Utilizar 2 led's verdes (ligados aos pinos 11 e 9)
- - led\_Vermelho\_Carro
- - led\_Amarelo\_Carro
- - led\_Verde\_Carro
- - led\_Vermelho\_Peao
- - led\_Vermelho\_Peao

Funcionamento:

Com o led vermelho do carro acesso, temos também acesso o led verde do peão.

Com o led verde de carro acesso o vermelho do peão está também acesso.

Com o led amarelo do carro o led vermelho do peão está acesso

O tempo é:

O led vermelho do carro durante 12 segundos acesso, o led verde do carro durante 12 segundos acesso e o led amarelo durante 4 segundos .



28

## Desafio 1 – programa do semáforo

```

int timer_1 = 500;
int ledVermelhoCarro = 13;
int ledAmareloCarro = 11;
int ledVerdeCarro = 9;
int ledVermelhoPeao = 7;
int ledVerdePeao = 4;
void setup() {
  pinMode(ledVerdeCarro, OUTPUT);
  pinMode(ledVermelhoCarro, OUTPUT);
  pinMode(ledAmareloCarro, OUTPUT);
  pinMode(ledVermelhoPeao, OUTPUT);
  pinMode(ledVerdePeao, OUTPUT);
  for (int i=0; i<4; i++){
    digitalWrite(ledVerdeCarro, HIGH);
    digitalWrite(ledVermelhoCarro, HIGH);
    digitalWrite(ledAmareloCarro, HIGH);
    digitalWrite(ledVermelhoPeao, HIGH);
    digitalWrite(ledVerdePeao, HIGH);
    delay(timer_1);
    digitalWrite(ledVerdeCarro, LOW);
    digitalWrite(ledVermelhoCarro, LOW);
    digitalWrite(ledAmareloCarro, LOW);
    digitalWrite(ledVermelhoPeao, LOW);
    digitalWrite(ledVerdePeao, LOW);
    delay(timer_1);
  }
}

void loop() {
  digitalWrite(ledVerdeCarro, HIGH);
  digitalWrite(ledVermelhoCarro, LOW);
  digitalWrite(ledAmareloCarro, LOW);
  digitalWrite(ledVermelhoPeao, HIGH);
  digitalWrite(ledVerdePeao, LOW);
  delay(20*timer_1);
  digitalWrite(ledVerdeCarro, LOW);
  digitalWrite(ledVermelhoCarro, LOW);
  digitalWrite(ledAmareloCarro, HIGH);
  digitalWrite(ledVermelhoPeao, HIGH);
  digitalWrite(ledVerdePeao, LOW);
  delay(4*timer_1);
  digitalWrite(ledVerdeCarro, LOW);
  digitalWrite(ledVermelhoCarro, HIGH);
  digitalWrite(ledAmareloCarro, LOW);
  digitalWrite(ledVermelhoPeao, LOW);
  digitalWrite(ledVerdePeao, HIGH);
  delay(16*timer_1);
  digitalWrite(ledVerdePeao, LOW);
  delay(timer_1);
  digitalWrite(ledVerdePeao, HIGH);
  delay(timer_1);
  digitalWrite(ledVerdePeao, LOW);
  delay(timer_1);
  digitalWrite(ledVerdePeao, HIGH);
  delay(timer_1);
  digitalWrite(ledVerdePeao, LOW);
  delay(timer_1);
  digitalWrite(ledVerdePeao, HIGH);
  delay(timer_1);
}

```

29

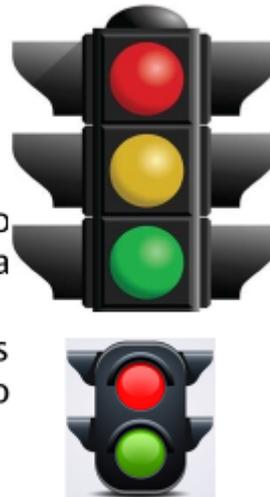
## Desafio 2

- Utilizando o LDR para simular o dia e a noite

Funcionamento:

Durante o dia – LDR destapado – o funcionamento do semáforo fica como no desafio 1

Durante a noite – LDR tapado – os led's do semáforo apagam e só o led amarelo é que fica a piscar



30

# Questões?

Obrigado pela vossa atenção.

31

## Bibliografia

- ▶ <http://www.arduino.org>
- ▶ <https://www.arduino.cc>
- ▶ <http://www.infowester.com>
- ▶ <http://escoladerobotica.ipcb.pt>
- ▶ <http://www.arduinoportugal.pt>
- ▶ <http://www.cirtuitar.com.br>

32

## ANEXO G – Perguntas dos questionários

### Questionário 1

- Sentiu dificuldades na compreensão da tarefa proposta?
- O objetivo da tarefa foi entendido?
- Conseguiu compreender os conceitos de programação?
- Sentiu dificuldades em compreender as respostas (feedback) das ações programadas?
- A atividade prática contribuiu para uma melhor compreensão de novos conceitos de programação?
- Os conceitos de programação foram adequados à categoria/tipo de tarefa?
- A Robótica Pedagógica é um instrumento interessante para ensino de programação?
- A utilização do “*hardware*” ocorreu sem problemas ou dificuldades?
- Os recursos (*led's*, sensores) foram percebidos de maneira clara?
- As respostas dadas pelo “*hardware*” à programação executada foram claras e coerentes?
- A utilização da Robótica Pedagógica auxiliou na aprendizagem ou aumentou seu conhecimento nos conceitos de programação?
- A utilização do “*IDE*” ocorreu sem problemas?
- Houve dificuldades em aplicar os conceitos de Lógica de Programação?
- Houve problema em compreender os objetivos da prática proposta?
- Teve dificuldade em elaborar a solução da prática proposta?
- Houve dificuldades em aplicar os conceitos de Lógica de Programação (estruturas de decisão, estruturas de repetição etc.) na solução?
- Como avalia a atividade realizada?

## Questionário 2

- Considero o uso do Arduino para a programação divertido?
- Considero o uso de Arduino na programação agradável?
- Tenciono continuar a programar usando Arduino?
- Dado que tenho acesso ao Arduino, planeio usá-lo?
- Atualmente considero-me um utilizador de Arduino frequente?
- O uso de Arduino melhora o meu desempenho na programação?
- O uso do Arduino melhora a minha produtividade?
- O uso de Arduino torna-me eficaz na programação?
- Considero útil o uso de Arduino na programação?
- Aprender programação é fácil?
- Considero que é fácil fazer o que pretendo com o uso do Arduino em programação?
- Para mim será fácil ser um utilizador avançado em Arduino?
- É fácil usar o Arduino?
- Descreva por palavras suas, os pontos fortes e os pontos fracos do uso da robótica para a aprendizagem da programação:

# ANEXO H – Questionários *online*

## Questionário 1

Práticas com Robótica Pedagógica

**\*Obrigatório**

**Iniciais do nome \***

**Idade \***

**Sexo? \***

*Marcar apenas uma oval.*

- Feminino
- Masculino

**Responda a cada afirmação usando a escala de 1- Muita Dificuldade ... 5 - Nenhuma Dificuldade \***

*Marcar apenas uma oval por linha.*

	1 - Muita Dificuldade	2	3	4	5 - Nenhuma Dificuldade
Sentiu dificuldades na compreensão da tarefa proposta?	<input type="radio"/>				
O objetivo da tarefa foi entendido?	<input type="radio"/>				
Conseguiu compreender os conceitos de programação?	<input type="radio"/>				
Sentiu dificuldades em compreender as respostas (feedback) das	<input type="radio"/>				

	1 - Muita Dificuldade	2	3	4	5 - Nenhuma Dificuldade
ações programadas?					
A atividade prática contribuiu para uma melhor compreensão de novos conceitos de programação?	<input type="radio"/>				

**Os conceitos de programação foram adequados à categoria/tipo de tarefa? \***

*Marcar apenas uma oval.*

- Sim
- Não

**Prática com Robótica Pedagógica. Responda a cada afirmação usando a escala de 1- Discordo Totalmente ... 5 - Concordo totalmente \***

*Marcar apenas uma oval por linha.*

	1 - Discordo Totalmente	2 - Discordo parcialmente	3 - Indiferente (não concordo nem discordo)	4 - Concordo parcialmente	5 - Concordo totalmente
A Robótica Pedagógica é um instrumento interessante para ensino de programação?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A utilização do "hardware" ocorreu sem problemas ou dificuldades?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Os recursos (led's, sensores, etc.) foram	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	1 - Discordo Totalmente	2 - Discordo parcialmente	3 - Indiferente (não concordo nem discordo)	4 - Concordo parcialmente	5 - Concordo totalmente
percebidos de maneira clara?					
As respostas dadas pelo "hardware" à programação executada foram claras e coerentes?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A utilização da Robótica Pedagógica auxiliou na aprendizagem ou aumentou seu conhecimento nos conceitos de programação?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A utilização do "IDE" ocorreu sem problemas?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Houve dificuldades em aplicar os conceitos de Lógica de Programação?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Houve problema em compreender os objetivos da prática proposta?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Teve dificuldade em elaborar a solução da prática proposta?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Houve dificuldades	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1 - Discordo Totalmente	2 - Discordo parcialmente	3 - Indiferente (não concordo nem discordo)	4 - Concordo parcialmente	5 - Concordo totalmente
-------------------------------	------------------------------	--	---------------------------------	-------------------------------

---

em aplicar os conceitos de Lógica de Programação (estruturas de decisão, estruturas de repetição etc.) na solução?

---

**Responda a cada afirmação usando a escala de 1- Nada interessante ... 5 - Muito interessante \***

*Marcar apenas uma oval por linha.*

1 - Nada interessante	2 - Pouco interessante	3 - Interessante	4 - Bastante interessante	5 - Muito interessante
--------------------------	---------------------------	---------------------	------------------------------	---------------------------

---

Como avalia a atividade realizada?



Com tecnologia



## Questionário 2

\*Obrigatório

Responda a cada afirmação usando a escala de 1- Discordo Totalmente .....7-  
Concordo Totalmente \*

Marcar apenas uma oval por linha.

	1- Discordo Totalmente	2	3	4	5	6	7- Concordo Totalmente
Considero o uso do Arduino para a programação divertido.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Considero o uso de Arduino na programação agradável	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tenciono continuar a programar usando Arduino	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dado que tenho acesso ao Arduino, planeio usá-lo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Atualmente considero-me um utilizador de Arduino frequente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O uso de Arduino melhora o meu desempenho na programação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
O uso do Arduino melhora a minha produtividade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

	1- Discordo Totalmente	2	3	4	5	6	7- Concordo Totalmente
O uso de Arduino torna-me eficaz na programação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Considero útil o uso de Arduino na programação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aprender programação é fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Considero que é fácil fazer o que pretendo com o uso do Arduino em programação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Para mim será fácil ser um utilizador avançado em Arduino	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
É fácil usar o Arduino	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Descreva por palavras suas, os pontos fortes e os pontos fracos do uso da robótica para a aprendizagem da programação:**

Com tecnologia

