

**UNIVERSIDADE DE LISBOA**



**OS ROBÔS NO ENSINO DA PROGRAMAÇÃO**

**Humberto Jorge Duarte Ferreira**

**RELATÓRIO DA PRÁTICA DE ENSINO SUPERVISIONADA**

**MESTRADO EM ENSINO DE INFORMÁTICA**

**2013**



**UNIVERSIDADE DE LISBOA**



**OS ROBÔS NO ENSINO DA PROGRAMAÇÃO**

**Humberto Jorge Duarte Ferreira**

**RELATÓRIO DA PRÁTICA DE ENSINO SUPERVISIONADA ORIENTADO PELO PROFESSOR  
DOUTOR JOÃO FILIPE MATOS**

**MESTRADO EM ENSINO DE INFORMÁTICA**

**2013**



## **Agradecimentos**

Todas as pessoas ao longo deste processo foram bastante importantes tanto no apoio técnico ou moral. Agradeço em primeiro lugar à Professora Paula Abrantes pelo apoio prestado e ideias desenvolvidas, pela confiança que depositou em mim desde o início do curso e por alimentar o meu gosto pela programação e robótica. Agradeço ao Professor Doutor João Filipe Matos pelo apoio prestado e pela coordenação geral dos trabalhos. Agradeço à Escola Secundária de Camões pelo excelente acolhimento e nomeadamente ao professor cooperante, Alexandre Barão, pela disponibilidade prestada e aos alunos alvos da intervenção. Agradeço também aos meus colegas de curso José Coutinho, Elsa Almeida, Renato Santos e Henrique Correia pelo bom ambiente criado e excelente trabalho desenvolvido em conjunto. Agradeço ao Mauro Ferreira e ao Luís Carreira pelo apoio técnico prestado. Agradeço ao Tiago Ricardo pelas explicações de eletrónica. Agradeço à minha esposa Andreia Firmino por me apoiar e compreender neste duro percurso. Agradeço aos meus pais Manuel Ferreira e Margarida Duarte e ao meu irmão Marco Ferreira por me apoiarem ao longo deste processo. Agradeço ao meu primo Nicolau Ferreira por todas as pilhas disponibilizadas que foram necessárias para alimentar os robôs e também ao meu primo Cláudio Ferreira pela ajuda prestada.



## Resumo

Este relatório incide numa intervenção realizada na Escola Secundária de Camões em Lisboa e visa analisar em que medida os robôs podem potenciar a aprendizagem das estruturas de controlo, em programação, numa turma de 10º ano do Curso Profissional de Técnico de Informática de Gestão. O projeto realizado pelos alunos, em cinco sessões de noventa minutos, consistiu em elaborar algoritmos para movimentar os robôs sem que os mesmos colidissem com os objetos à sua volta. O trabalho foi inspirado em metodologias tais como Project Based Learning, Competition Based Learning e Cenários de Aprendizagem (Carroll, 2000). Para além dos instrumentos de avaliação dos alunos, foram realizados pré e pós-testes para verificar as aprendizagens ocorridas durante a intervenção. Os dados recolhidos revelam que a utilização de robôs como recurso educativo, nesta intervenção, revelou algumas evidências de aprendizagem dos conteúdos abordados. A utilização de robôs também demonstrou ser um fator de interesse e envolvimento por parte dos alunos.

**Palavras-chave:** robótica educativa, aprendizagem baseada na competição, programação.





## **Abstract**

Teaching and learning computer programming is quite difficult. Recently, the research identifies some artifacts that can contribute to student learning. The interaction with robots can have an important role in the computer science disciplines. This study tries to contribute with the possibility that robots can help secondary students to learn programming, in this case control structures. Before the robotics learning activity, the students took a pretest for analyzing their knowledge of control structures. After this test, students developed an algorithm to move the robots with no collisions in a small arena. The posttest was done after all robots activity. Both tests were used in this study to measure learning. To move robots correctly the teacher introduced basic instructions and the programming structures control. The robots were equipped with three simple ultrasonic sensors and two motors to move themselves. Coupled to all the hardware, the robot has an Arduino microcontroller which has been programmed by the students. The statistical result indicated that significant differences were found between pre and posttest. Like other authors mentioned in this document, this studied concludes that robots are not a guaranteed solution, but they can improve students programming learning's in computer science secondary courses.

**Keywords:** educative robotics, competition-based learning, computer programming.



## Índice Geral

Agradecimentos.....	i
Resumo.....	iii
Abstract .....	v
Índice Geral.....	vii
Índice de Figuras .....	viii
Índice de Gráficos .....	ix
Índice de Tabelas.....	ix
Introdução .....	1
1. Enquadramento Teórico .....	3
1.1. A Programação .....	3
1.2. A Problemática no Ensino da Programação .....	4
1.3. Possíveis Soluções na Literatura .....	5
2. Contexto Escolar .....	11
2.1. Caracterização da Turma.....	11
3. Unidade Didática e Enquadramento Curricular .....	13
3.1. A disciplina de Linguagens de Programação .....	13
3.2. Conceitos Fundamentais e Articulação com Experiências Prévias.....	13
4. Projeto de Intervenção.....	19
4.1. Organização das Sessões de Trabalho .....	21
4.2. Robô Adotado .....	22
5. Intervenção.....	23
6. Aprendizagens Ocorridas e Avaliação do Projeto .....	33
6.1. Instrumentos de Avaliação dos Alunos.....	33
6.2. Abordagem Metodológica.....	35
6.3. Avaliação do Projeto .....	36
6.4. Análise de Resultados .....	42

7. Reflexão Final .....	45
8. Referências Bibliográficas .....	47
9. Anexos.....	49
Anexo A – Questionário Online à Turma .....	50
Anexo B – Enunciado Projeto Um Robô em Marte.....	56
Anexo C – Linguagens de Programação: Elenco Modular .....	60
Anexo D – Cenário de Aprendizagem “Uma Experiência em Marte” .....	62
Anexo E – O Robô <i>Arduino</i> Montagem do Equipamento .....	64
Anexo F – Programação em <i>Arduino</i> vs. C/C++ .....	72
Anexo G – Project Based Learning.....	74
Anexo H – Etapas do Projeto .....	76
Anexo I – API ( <i>Application Programming Interface</i> ) do Robô.....	78
Anexo J – Planeamento das Sessões de Trabalho .....	82
Anexo K – Grelha de Observação Direta.....	90
Anexo L – Auto e Heteroavaliação .....	98
Anexo M – Diários de Bordo .....	104
Anexo N – Avaliação do Produto .....	108

### Índice de Figuras

Figura 1 - Acesso condicional <i>if</i> .....	14
Figura 2 - Acesso Condicional Composto <i>if..else</i> .....	15
Figura 3 - Ciclos de Repetição .....	16
Figura 4 - Arena .....	19
Figura 5 - Robôs.....	22
Figura 6 - Arena .....	24

### **Índice de Gráficos**

Gráfico 1 - Observação Direta Individual dos Alunos.....	34
Gráfico 2 - Avaliação das Soluções .....	35
Gráfico 3 - Percentagem da resposta mais correta .....	38
Gráfico 4 - Percentagem das Duas Respostas Mais Corretas .....	39
Gráfico 5 - Percentagem da Resposta Mais Correta por Estrutura de Controlo .....	39
Gráfico 6 - Percentagem das Duas Respostas Mais Corretas por Estrutura de Controlo .....	40
Gráfico 7 – Resumo do Questionário Geral.....	41

### **Índice de Tabelas**

Tabela 1 - Respostas do Pré-teste.....	37
Tabela 2 - Respostas do Pós-teste .....	38



## **Introdução**

Este relatório apresenta um projeto de plano de intervenção pedagógica no âmbito da Prática de Ensino Supervisionada do curso de Mestrado em Ensino de Informática do Instituto de Educação, da Universidade de Lisboa e apresenta a intervenção realizada no módulo 3, estruturas de controlo, da disciplina de Linguagens de Programação do Curso Profissional de 10º ano de Técnico de Informática de Gestão a decorrer na Escola Secundária de Camões em Lisboa. Este plano de intervenção teve uma duração de cinco sessões de noventa minutos sendo que o módulo em que se enquadra tem uma duração total de trinta e seis horas letivas.

No âmbito do ensino da programação, são diversos os autores que afirmam que o ensino e aprendizagem da programação continua a ser uma tarefa difícil. Segundo Gomes e Mendes (2007) citados por Gomes (2012), embora possam existir alguns instrumentos que podem facilitar este processo, ainda existem muitos problemas por solucionar. Este relatório tem como propósito apresentar uma proposta de plano de intervenção pedagógica e identificar possíveis evidências da utilização da robótica como instrumento facilitador da aprendizagem da programação informática, nomeadamente de estruturas de controlo. Para isso foram preparados robôs, controlados por um microprocessador Arduino, o qual foi programado ao longo das sessões pelos alunos. Apenas com a ajuda de sensores ultrassónicos ligados ao microcontrolador, os alunos tiveram que implementar algoritmos de modo a movimentar o robô evitando colisões com obstáculos à sua volta. O projeto desenvolvido foi inspirado num Cenário de Aprendizagem (Carroll, 2000), “Uma experiência em Marte” e o método de trabalho inspirado nas metodologias Project Based Learning e Competition Based Learning. A componente investigativa deste estudo procura identificar em que medidas a utilização de robôs pode potenciar a aprendizagem das estruturas de controlo em programação. Para verificar as aprendizagens ocorridas nos alunos, foram realizados dois inquéritos, um antes e outro depois da intervenção, pré e pós-teste respetivamente. Com base nos dados recolhidos as estatísticas mostram que os robôs, nesta situação específica, podem ter um papel importante no ensino da programação informática, nomeadamente nas estruturas de controlo.





## **1. Enquadramento Teórico**

Este capítulo procura, por um lado, enquadrar o leitor nas tecnologias abordadas ao longo do documento, e por outro, apresentar teoricamente, com base em literatura de referência, o conjunto de metodologias adotadas ou inspiradoras da intervenção pedagógica apresentada posteriormente no capítulo 3.

### **1.1. A Programação**

Geralmente os processos que envolvem a atividade de programação não são simples. Segundo Major (2010), a programação está muitas vezes associada a diversos estereótipos, tais como complicada ou confusa. Estas atividades exigem algum poder de concentração para, depois de identificar o problema, conceber e projetar mentalmente uma possível resolução. Estes processos terminam quando a própria solução é transposta para uma determinada linguagem de programação. Segundo Abrantes (2012)

Aprender a programar relaciona-se com vários campos da tecnologia e apesar do seu encanto poucos são os alunos que conseguem ter sucesso quando iniciam o estudo de uma linguagem de programação. Por vezes, o que começou por ser uma ambição facilmente se transforma uma tarefa árdua e desagradável. Alguns estudos realizados nesta área identificaram uma variedade de possíveis razões para esta perceção. (Abrantes, 2012, p.1)

Segundo Downey (2012) podemos definir a programação como o processo de conceptualizar, escrever, testar, depurar erros e até mesmo fazer a manutenção do código fonte das aplicações. Por outras palavras, tudo isto consiste em analisar um determinado problema, organizar os dados, gerir diferentes fluxos de informação e construir uma sequência de procedimentos que o computador terá de realizar de acordo com as necessidades identificadas. Para isso existem linguagens de programação que disponibilizam variáveis, estruturas de controlo, funções, objetos, procedimentos, operações, etc., para que o programador consiga, das mais variadas formas, implementar os requisitos desejados.

Para Blackwell (2002), referido por Bryant, Romero & Boulay (2008), a programação é uma tarefa difícil porque é necessário representar abstrações através de notações escritas. No mesmo sentido, Pennington, Lee & Rehder (1995), referidos

também por Bryant et al. (2008) afirmam que esta tarefa é difícil porque é necessário gerir vários níveis de informação em simultâneo.

As estruturas de controlo, também conhecidas por estruturas de controlo de fluxo, são uma das componentes importantes na programação. As mesmas estão ligadas ao controlo da lógica e sequência procedimental na execução de um programa. Nos seus estudos, Saeli, Perrenet, Jochems & Zwaneveld (2010) atribuem um papel muito importante às estruturas de controlo em programação. No mesmo sentido Govender (2006) identifica as estruturas de controlo como uma das 3 principais componentes dentro da programação. Estes conceitos estão de certa forma ligados às estruturas conceptuais cognitivas como podemos ver no estudo de Clancey (1985)

Conceptual Structures is really three books: an encyclopedic survey of philosophical and psychological foundations of AI theory (including an epilogue on the limits of formal reasoning); a mathematical text that develops a knowledge notation called a conceptual graph and reasoning operators for manipulating it; and examples of how this notation is useful for natural language processing, database inference, and knowledge engineering. (Clancey, 1985, p.1)

## **1.2. A Problemática no Ensino da Programação**

Considerando o que foi referido no ponto anterior, não sendo a aprendizagem da programação uma tarefa fácil, a tarefa de ensinar programação é também ela difícil. Segundo Saeli et al. (2010), existe uma grande diferença entre ter conhecimentos de programação (saber programar) e a aptidão para ensinar esses mesmos conhecimentos. O Conhecimento Pedagógico, cujo termo original *Pedagogical Content Knowledge (PCK)* introduzido por Shulman em 1986, referido por Saeli et al. (2010), é uma forma de representar e formular um conceito de modo a que fique compreensível aos outros. Neste contexto, também nos é dito que não existe qualquer forma poderosa para ensinar ou representar quaisquer conceitos, quem ensina é que deve possuir uma panóplia enorme de formas distintas e originais de representar determinado conhecimento de modo a ser compreendido pelo recetor, Shulman (1986). Diversos estudos já foram realizados nesta área, segundo Rovegno (1992), Grossman & Lynn (1990), Loughran, Milroy, Berry, Gunstone & Mulhall (2001),

Morine-Deshimer & Kent (1999), Van Driel, Verloop & de Vos (1998), Sanders, Borko & Lockard (1993) referidos por Saeli et al. (2010), o desenvolvimento destes Conhecimentos Pedagógicos (PCK) demoram anos a aperfeiçoar ao longo da carreira de um professor.

No contexto de programação Saeli et al. (2010) refere que

In the process of learning to program, Govender (2006) identifies, from a technical point of view, three main aspects students need to learn: data, instructions and syntax. Data refers to the concepts of variables and data types for procedural programming, and objects involving attributes and actions for OO programming. As for instructions, the needed understanding is about control structures and subroutines for the procedural programming, and interacting objects and methods in the case of OO programming. Syntax denotes the group of rules that determine what is allowed and what is not within a programming language. Syntax rules determine what it is called the vocabulary of the language, how programs can be constructed using techniques such as loops, branches and subroutines. (Saeli et al., 2010, p. 79)

Segundo Saeli et al (2010), os conceitos inerentes às estruturas de controlo, parte integrante de *Instructions* na citação anterior, são um exemplo de conteúdos difíceis de ensinar e aprender.

### **1.3. Possíveis Soluções na Literatura**

Neste subcapítulo são apresentados recursos educativos, tecnologias e metodologias que, segundo alguns autores, podem ajudar a colmatar os problemas identificados na aprendizagem da programação.

Tal como já foi evidenciado nos pontos anteriores, a programação não é uma tarefa fácil, por outro lado, a manipulação exclusiva de conceitos abstratos em problemas igualmente abstratos pode dificultar a sua aprendizagem (Santos, Fermé, & Fernandes, 2005). Os autores Gomes e Mendes (2007), citados por Gomes e Abrantes (2012), referem que embora existam bastantes instrumentos disponíveis para ensinar e aprender programação informática, ainda existem diversos problemas que dificultam o ensino da mesma; outros autores referem que os métodos

tradicionais de ensino da programação “tendem a enfatizar conceitos e procedimentos abstratos, o que faz com que os estudantes frequentemente não consigam extrapolar o comportamento dos programas a partir desses conceitos” (Santos, Fermé, & Fernandes, 2005, p.2). Isto traz mais dificuldade em entender o “comportamento” resultante das aplicações informáticas que os próprios alunos constroem (Price, Hirst, Johnson, Petre, & Richards, 2002; Santos, Fermé, Fernandes, 2005). Na sequência destes estudos e na tentativa de superar estas dificuldades, estudos feitos por alguns autores, Gomes (2012); Varanda (2012), Costa (2012) e Bernardo (2012), evidenciam o uso de robôs e as potencialidades dos mesmos no ensino da programação.

Edwards, Coddington & Caterina (1997), Hloupis, Stavrakas, Moutzouris, Alexandridis, & Triantis (2011) e Benitti (2012) referem que a utilização de robôs traz mais interesse e facilidade de aprendizagem. Carbonaro, Szafron, Cutumisu & Schaeffer (2010), referem que embora o grau de dificuldade possa ser o mesmo, poderão existir diferenças entre a observação de resultados num ecrã e a perceção de resultados num meio físico com um recurso educativo palpável, sendo que, este facto poderá fazer com que as aprendizagens sejam mais vividas pelos alunos. No mesmo sentido, Hloupis, et al. (2011) consideram que as teorias modernas (atuais) procuram utilizar métodos ou recursos que criem estímulos que despertem o interesse dos alunos. Neste caso, esse estímulo poderá ser provocado pelo robô. Os mesmos autores também afirmam que a utilização de equipamentos físicos, na aprendizagem, podem criar um ambiente informal que pode ajudar os alunos a aprender enquanto se divertem. O estudo de Gregorc (1982) referido por Lau e Yuen (2010) diz-nos que os alunos participantes conseguiram aprender com mais facilidade quando os seus sentidos, principalmente o tato, também foram estimulados ao longo do processo. O mesmo estudo também evidencia que este facto ajuda-os a entender melhor os algoritmos que estão a implementar, na medida em que ao longo do processo acabam por criar naturalmente diversas metáforas com aprendizagens previamente adquiridas.

A introdução da robótica no ensino da programação pressupõe a utilização de um microcontrolador como cérebro do robô. O *Arduíno*<sup>1</sup> é uma plataforma eletrónica

---

<sup>1</sup> [www.arduino.cc](http://www.arduino.cc)

open-source programável, através de uma porta USB ligada a um computador. Este microprocessador permite interligar diversos dispositivos tais como motores, leds, sensores de luz, sensores de som, etc., bem como qualquer outro equipamento eletrônico compatível. Após estar devidamente programado, o *Arduíno* pode, por exemplo, receber alguns *inputs* de sensores de luminosidade, sonar (para calcular distâncias) para depois refletir alguns comportamentos através de luzes, motores, ou mesmo comunicar através de rede *Ethernet* ou *sem fios* se tiver os devidos equipamentos necessários. O *Arduino* é assim introduzido neste contexto como um microprocessador capaz de controlar um equipamento mecânico, no presente contexto, um robô. Sendo uma tecnologia relativamente recente já se encontram alguns estudos sobre a sua utilização em contextos educativos. Cruz-Martín, Fernández-Madrigal, Galindo, González-Jiménez, Stockmans-Daou e Blanco-Claraco (2012) referem que a plataforma *Arduino* contém muitos componentes que permitem o controlo do sistema em tempo real e este fator pode fazer com que os alunos se concentrem mais na programação para perceber o que acontece em tempo real. Este facto deve-se aos alunos sentirem necessidade de perceber o que está a acontecer, ao nível da programação, para poderem controlar os diversos *inputs* externos do equipamento, Cruz-Martín, et al (2012). Esta tecnologia foi também utilizada na Escola Technical Institute Vallauri in Fossano (Cuneo), Piedmont, Itália onde foi objeto de estudo por parte de Barbero, Demo e Vaschetto (2011). Segundo estes autores, para além do interesse levantado pelos alunos, o mesmo estudo indica que esta plataforma pode potenciar a aprendizagem da programação.

Para programar o *Arduino* é disponibilizado um compilador no *site* do fabricante. Este compilador dispõe de uma linguagem de programação chamada *Arduino Language*<sup>2</sup> que é baseada em *Wiring* e muito semelhante à conhecida linguagem C/C++. O quadro apresentado no

---

<sup>2</sup> <http://www.arduino.cc/>

Anexo F – Programação em *Arduino* vs. C/C++ mostra as principais semelhanças entre a sintaxe das duas linguagens *Arduino* e C/C++. Como se pode observar, no mesmo Anexo, todas as estruturas de controlo apresentam-se com as mesmas palavras reservadas em ambas as linguagens.

Relativamente à utilização das linguagens de programação, segundo Xinogalos, Satratzemi, & Dagdilelis (2006), um fator importante na aprendizagem e na depuração de erros ao longo do desenvolvimento de uma aplicação consiste em fazer o chamado *debugging* em tempo real. Segundo os mesmos autores, depois da aplicação compilada, a monitorização da mesma, por parte dos alunos, ajuda-os nos seguintes pontos: compreender a dinâmica de execução de um programa; apoio substancial na compreensão dos conceitos e comportamentos das estruturas de controlo; localização e resolução de erros. Com esta tecnologia, a execução dos programas em *Arduino* é realizada no microcontrolador, e não no computador. No entanto, o *Arduino* disponibiliza uma comunicação por série que pode ficar ligada ao computador quando o mesmo está em plena execução, permitindo assim a monitorização do *Arduino* em tempo real através do *Serial Monitor* do compilador do *Arduino*. Para isso basta que o aluno adicione o envio das “mensagens” que pretende nos pontos do código que desejar.

Relativamente a metodologias de trabalho, nomeadamente em possíveis formas de introduzir projetos a realizar em sala de aula, Carroll (2000), apresenta o conceito de Cenário de Aprendizagem como possíveis histórias de atividades que poderiam acontecer. Estas mesmas histórias podem ser fonte inspiradora de um determinado tema ou contexto relevante em sala de aula. O mesmo autor refere também que os cenários são em simultâneo concretos, no sentido em que apresentam desafios focados e flexíveis, porque permitem visões diferentes do problema e consequentemente implementações distintas em casos concretos. O autor ainda apresenta os cenários de aprendizagem como desafios que podem facilitar o envolvimento dos alunos. Neste sentido os cenários de aprendizagem podem ser trazidos para as salas de aula e aplicados aos contextos de ensino-aprendizagem presentes nas mesmas.

Também outros métodos de trabalho podem promover o envolvimento dos alunos em projetos, a Competition-Based Learning (CnBL), em português Aprendizagem

Baseada na Competição, pode ser uma opção. De acordo com o Dicionário Léxico<sup>3</sup>, Dicionário Português *Online*, a competição pode ser o ato de concorrer com alguém (rival) pelo mesmo objetivo procurando o melhor título. Baseando-se neste conceito, a CnBL, segundo Burguillo (2010, p.2) «is a methodology where learning is achieved through a competition, but the learning result is independent of the student's score in such competition». CnBL é assim uma metodologia que pode ser utilizada para promover a aprendizagem de algo. Chang, Yang, Yu, & Chan (2003), Siddiqui, Khan, & Akhtar (2008), Chung (2008) e novamente Burguillo (2010) consideram que este tipo de metodologia poderá aumentar o desejo de aprender dos estudantes. Os estudos realizados por Ebner & Holzinger (2007), Lawrence (2004), Ribeiro, Simões & Ferreira (2009) e Wallace & Margolis (2007) também referidos por Burguillo (2010), referem que a competição pode ser usada para promover a aprendizagem nas escolas secundárias e de ensino superior.

Desenvolvendo o seu estudo e observando os resultados, Burguillo (2010) refere que CnBL pode ser potenciado quando aliado a outras metodologias, tais como Project-Based Learning (PjBL), Problem Based Learning (PBL) e Cooperative Learning (CL). Neste sentido e como metodologia de trabalho geralmente ligada ao desenvolvimento ou construção de “produtos”, a aprendizagem baseada em projetos ou, em inglês, *Project-Based Learning* (PjBL), é uma metodologia de ensino e aprendizagem que se apoia nas correntes das teorias práticas e críticas. Estas assentam essencialmente na interação entre alunos e professores e na reflexão crítica das aprendizagens, distinguindo-se assim dos modelos da teoria técnica behaviouristas, (Pacheco, 1996). Não parece existir uma definição unanimemente aceite, contudo os conceitos ou componentes que envolve são consentidos pela generalidade dos autores. Helle, Tynjälä & Olkinuora (2006) com base numa revisão da literatura que envolveu artigos entre os anos 1960 a 2001, identificou um conjunto de características apresentadas no Anexo G – Project Based Learning. Estas características referem, por exemplo, que esta metodologia resulta frequentemente num produto final o qual pode ser um programa de computador.

Os mesmos autores referem que o PjBL pode ser implementado de diferentes formas, consoante orientações pedagógicas, éticas ou políticas (Helle, et all, 2006): Exercício

---

<sup>3</sup> <http://www.lexico.pt>

de Projeto; Componente de Projeto; e Orientação de Projeto. A descrição de cada uma também pode ser encontrada novamente no Anexo G – Project Based Learning.

O desenvolvimento de projetos envolve geralmente grupos de trabalho fomentando assim a interação aluno-aluno e o espírito de equipa apreciado na metodologia CnBL, Burguillo (2010). Diversos autores apontam algumas vantagens na realização de trabalhos em grupo. Segundo César (2000), onde são referidos trabalhos de Schubauer-Leoni (1989), existe uma melhor gestão de conflitos em situações de trabalhos em grupo. No mesmo sentido, o estudo de Bryant, Romero & Boulay (2008), refere que a atividade de programação realizada entre pares, “Pair Programming”, traz benefícios ao nível académico porque, por exemplo, as soluções são elaboradas e discutidas com mais detalhe. Por outro lado, as interações dentro dos grupos também fomentam os conflitos socio-cognitivos, conduzindo os sujeitos a “co-recontextualizar os seus saberes e competências” (César, 2000, p. 9). Ainda segundo César (2000), ao trabalharem em grupos, os alunos serão levados a argumentar e defender as suas opiniões sendo que desta forma as aprendizagens tornam-se mais efetivas.



## **2. Contexto Escolar**

A Escola Secundária de Camões está localizada na Praça José Fontana em Lisboa (ESC, 2010) e tem como objetivos ser um espaço de acessões artísticas, científicas e literárias promovendo um espaço de trabalho e experiências democráticas (Pires, 2010). Segundo os últimos registos de 2011/12, a escola tem mil setecentos e vinte e três alunos (ESC, 2012).

Como oferta formativa, inclui diversos cursos científico-humanísticos e cursos profissionais dos quais Técnico de Apoio à Gestão Desportiva, Técnico de Informática de Gestão e Técnico de Serviços Jurídicos.

A escola dispõe de infraestruturas antigas, contudo está equipada a nível tecnológico, nomeadamente acesso à internet e computadores recentes nas salas de informática. Cada uma destas salas possui cerca de dez computadores.

### **2.1. Caracterização da Turma**

A intervenção aqui descrita ocorreu numa turma do primeiro ano do Curso de Técnico de Informática de Gestão, na disciplina de Linguagens de Programação, a funcionar habitualmente em dois turnos. O primeiro turno tem 9 alunos, 7 rapazes e 2 raparigas; e o segundo tem 7 alunos, 6 rapazes e 1 rapariga.

Para verificar se os alunos tinham conhecimentos ou experiências prévias compatíveis com o trabalho a realizar realizou-se um inquérito (Anexo A – Questionário Online à Turma) com objetivo de analisar as seguintes dimensões: estado social (origem portuguesa ou estrangeira); ocupações extra escolares e tempos de viagem casa-escola (para verificar se os alunos têm o seu tempo muito ocupado para além das atividades letivas); compreensão da Língua Portuguesa (para verificar se não existe qualquer dificuldade acrescida de comunicação entre as interações aluno-professor e aluno-aluno); acesso fácil a equipamento informático (para verificar se os alunos poderão praticar algumas competências em casa); e, gosto e interesse pela disciplina de linguagens de programação. Os dados recolhidos podem ser consultados também no Anexo A e são apresentados de seguida de forma resumida:

- Os alunos têm idades entre os 14 e os 18 anos distribuídas da seguinte forma: 2 alunos com 14 anos; 2 alunos com 15 anos; 5 alunos com 16 anos; 3 alunos com 17 anos; 4 alunos com 18 anos.

- 9 Alunos são de nacionalidade portuguesa bem como os seus pais. Os restantes 7 alunos são de outras nacionalidades;
- Relativamente à expressão em língua portuguesa 2 alunos apresentam muitas dificuldades e 8 apresentam algumas dificuldades;
- Nenhum dos alunos é trabalhador estudante ou tem alguma atividade profissional durante o tempo de aulas;
- A maioria dos alunos não aparentam morar longe da escola, apenas 6 afirmam levar entre 30 e 60 minutos a deslocar-se entre a sua casa e a escola;
- Todos eles têm computador em casa com acesso à internet;
- Relativamente às disciplinas de informática, 2 alunos afirmam ter muitas dificuldades, 10 alunos afirmam ter apenas algumas dificuldades e 4 alunos afirmam não ter qualquer dificuldade neste tipo de disciplina, contudo, todos os alunos afirmam gostar da disciplina de Linguagens de Programação;
- Relativamente às duas últimas questões do inquérito, existem dados um pouco contraditórios já que 13 alunos afirmam querer seguir um curso de informática após o 12º ano mas apenas 11 querem seguir estudos após o 12º ano.

### **3. Unidade Didática e Enquadramento Curricular**

Este capítulo apresenta o contexto curricular em que este projeto se enquadra apresentando uma descrição da disciplina e os principais conteúdos abordados.

#### **3.1. A disciplina de Linguagens de Programação**

A disciplina de Linguagens de Programação enquadra-se na componente de formação técnica do Curso Profissional de Informática de Gestão sendo repartida em 17 módulos e perfaz um total de 459 horas ao longo de 3 anos letivos. Esta disciplina tem como finalidade promover no aluno a aquisição de conhecimentos, experiências e competências na área da programação e respetivas tecnologias de desenvolvimento associadas. Estando ligada aos sistemas de informação, também engloba a abordagem a diversas técnicas e paradigmas de programação, tal como a programação orientada a objetos em módulos posteriores. O programa indica que as metodologias adotadas devem ser eminentemente práticas. As experiências realizadas devem procurar dotar os alunos de mais capacidades de análise, raciocínio lógico e abstrato, estruturação do pensamento e subdivisão de problemas complexos por forma a serem capazes de conceptualizar e formalizar aplicações informáticas profissionais, Direção Geral da Formação Vocacional (2005).

#### **3.2. Conceitos Fundamentais e Articulação com Experiências Prévias**

O presente trabalho realizou-se ao longo do módulo 3, Estruturas de Controlo, que é apresentado no programa da disciplina com uma duração de 36 horas visando os seguintes objetivos:

- Utilizar e identificar instruções compostas;
- Identificar e utilizar estruturas de repetição num programa;
- Utilizar as várias estruturas de seleção;
- Desenvolver programas que utilizem combinações entre estruturas de repetição e de seleção;
- Criar menus de opções;
- Compreender e aplicar saltos incondicionais.

Esta intervenção incidu no segundo, terceiro e quarto objetivos enunciados no ponto anterior. Neste sentido, pretendeu-se dotar os alunos de capacidades para transpor explicitamente as suas resoluções mentais em suporte programável, neste caso em concreto, na linguagem de programação *Arduino Language*. Desta forma garantiu-se uma certa continuidade com as experiências já realizadas pelos alunos, já que esta linguagem tem uma sintaxe idêntica ao C/C++ até agora adotada pelo professor cooperante.

As estruturas de controlo a abordar na intervenção são de dois tipos: acesso condicional e ciclos de repetição.

Antes de abordar qualquer estrutura de controlo, é importante perceber que as estruturas de acesso condicional, bem como as estruturas de repetição dependem do resultado de uma expressão lógica com valor booleano, verdadeiro ou falso. Este tipo de conhecimento já foi abordado em módulos anteriores desta disciplina.

De seguida são apresentados as estruturas de controlo de fluxo a abordar na intervenção:

- Estrutura de decisão ou acesso condicional *if*: consiste em executar um determinado conjunto de procedimentos caso a condição lógica especificada na cláusula *if* seja verdadeira.

Conceptualmente esta estrutura pode-se representar com o seguinte diagrama:

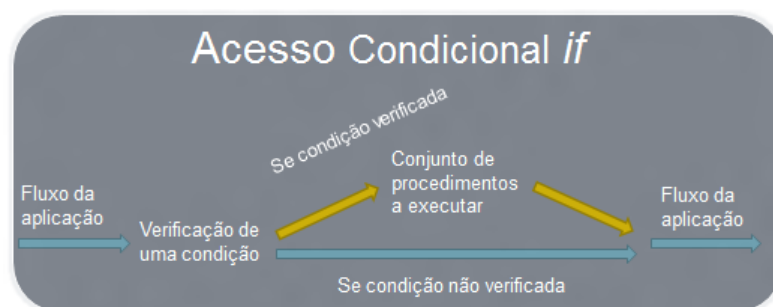


Figura 1 - Acesso condicional if

A sua representação na linguagem de programação *Arduino* pode ser da seguinte forma:

```
if (<condição lógica>){  
...  
}
```

- Estrutura de decisão composta ou acesso condicional composto *if..else*: Esta estrutura complementa a estrutura *if*, no sentido em que caso a condição lógica especificada tiver um valor de verdadeiro, é executado um conjunto de procedimentos, no caso contrário, se a condição lógica for falsa, é executado um conjunto de procedimentos distintos.

Conceptualmente esta estrutura pode-se representar com o seguinte diagrama:

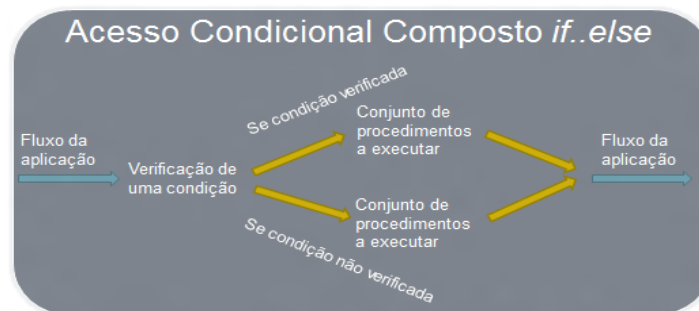


Figura 2 - Acesso Condicional Composto *if..else*

A sua representação em *Arduino Language* pode ser da seguinte forma:

```
if (<condição lógica>  
{  
...  
}  
else  
{  
...  
}
```

- Ciclo de repetição *for*: este ciclo de repetição é utilizado quando se conhece antecipadamente o número de iterações a executar pelo ciclo. A sua inicialização consiste em atribuir um valor inicial, geralmente 0, a uma variável de controlo,

definir a condição de paragem (geralmente o número de iterações desejadas) e definir o incremento à variável em cada iteração.

- A sintaxe para construção desta estrutura de controlo em *Arduino Language* é apresentada da seguinte forma:

```
for (inicialização; condição de paragem; incremento) {  
...  
}
```

- Ciclo de repetição *while* e *do..while*: este ciclo consiste em repetir um determinado conjunto de iterações até que a condição de paragem tenha o valor falso. Geralmente ao longo de cada iteração o valor de uma das variáveis da condição de paragem é de alguma forma manipulado. O *do..while* distingue-se do simples *while* porque o conjunto de instruções dentro do ciclo é sempre executado pelo menos uma vez, já que a condição de paragem é testada no final do ciclo.

Conceptualmente estas estruturas podem-se representar com um diagrama deste tipo:

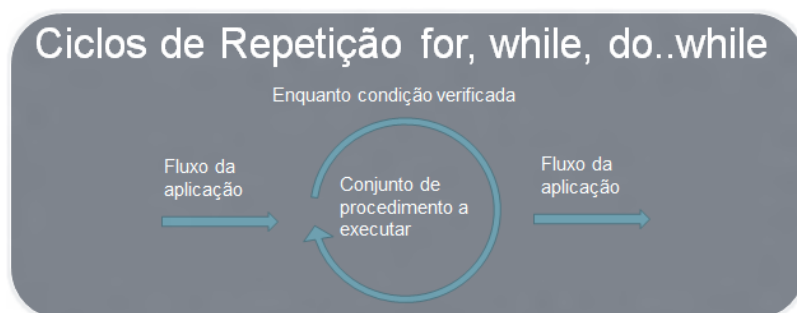


Figura 3 - Ciclos de Repetição

A sintaxe das estruturas *while* e *do..while* em *Arduino Language* são apresentadas de seguida respetivamente:

```
while(...){  
...  
}  
  
do{  
...  
} while (...);
```

Embora os saltos incondicionais, *continue*, *break*, *goto* façam parte deste módulo, os mesmos não farão parte dos conteúdos e objetivos desta intervenção. Sendo também uma estrutura de controlo, o salto incondicional *return* está ligado à implementação de funções, o que deverá apenas ser abordado num módulo posterior.

Por outras palavras e de modo a clarificar os objetivos gerais do módulo, acima referidos, é importante os alunos adquirirem competências/conteúdos:

- Apropriação do conceito inerente à estrutura de controlo Acesso Condicional Simples implementando a mesma, em situações distintas, com base em cláusulas lógicas que, se verificadas, determinam a execução de um determinado comportamento ou conjunto de operações.
- Apropriação do conceito inerente à estrutura de controlo de acesso condicional composto, aplicando a mesma, em situações distintas, com base em cláusulas lógicas que, se verificadas ou não, determinam a execução de um conjunto de procedimentos ou conjunto de procedimentos alternativos respetivamente.
- Apropriação do conceito inerente às estruturas de controlo de repetição com a aplicação de uma cláusula lógica condicionante à repetição de um conjunto de procedimentos desejados.
- Desenvolver algoritmos e programas que combinem a utilização dos 3 tipos de estruturas, enunciados nos pontos anteriores, para resolver problemas propostos em programação.





#### 4. Projeto de Intervenção

Este projeto é introduzido aos alunos como possível implementação do Cenário de Aprendizagem (Carroll, 2000) “Uma Experiência em Marte”, Anexo D – Cenário de Aprendizagem “Uma Experiência em Marte” deste documento. De forma geral, este Cenário de Aprendizagem apresenta um robô que tem por missão recolher amostras de Marte e enviar informação das mesmas para a Terra. Tendo como problemática principal a aprendizagem das estruturas de controlo acima apresentadas, operacionalizou-se o Cenário de Aprendizagem com a elaboração de um projeto com nome “Um Robô em Marte”, cujo enunciado também dado aos alunos pode ser consultado no Anexo B – Enunciado Projeto Um Robô em Marte. De uma visão

geral, este projeto consiste em programar um robô para, numa primeira fase fazê-lo, movimentar-se sem colidir com os obstáculos à sua volta, e por fim, encontrar um ponto de alto-relevo na maquete onde se encontra. Para detetar os obstáculos e o ponto de



Figura 4 - Arena

alto-relevo, os alunos deverão programar o robô para fazer a

leitura dos valores dos sensores ultrassónicos. Se esses valores forem demasiados pequenos, é porque o robô poderá estar prestes a colidir com um obstáculo, se se movimentar nessa direção. Se os valores subirem acima da dimensão da arena, os alunos poderão concluir que o robô encontrou o ponto de alto-relevo. Note-se que ao passar pelo ponto de alto-relevo, o robô fica com pelo menos um sensor apontado para cima sendo que a distância devolvida por esse mesmo sensor deixa de estar condicionada pelas dimensões da arena. Tal como nos inspira o Cenário de Aprendizagem, ao detetar o ponto de alto-relevo, o robô deverá parar e emitir um sinal sonoro, simulando o envio de comunicações para a Terra dando por terminada a sua missão.

Para criar condições propícias à competição, paralelamente à metodologia CnBL, a arena de jogo teve uma forma simétrica. Dentro da mesma, os dois robôs tiveram a possibilidade de se movimentar em espaços iguais e em simultâneo. Após contornar os obstáculos, os diversos robôs encontravam-se localizados num espaço comum, zona central da arena, onde tinham de encontrar o ponto de alto-relevo. Nesta zona não existiram obstáculos a não ser o próprio robô adversário. O primeiro robô a chegar ao ponto de alto-relevo tinha de enviar um sinal sonoro ganhando assim a partida. Ao longo das sessões, os confrontos de robôs foram-se realizando à medida que foram surgindo algoritmos/soluções tomadas como funcionais pelos alunos.

É de referir também que antes da realização deste projeto os alunos frequentaram sessões onde trabalharam os conteúdos aqui abordados. Esta intervenção realizou-se num período de cinco sessões de noventa minutos o que perfaz um tempo total relativamente reduzido para se poder implementar a metodologia PjBL na íntegra. Contudo, e dado que os alunos acabam por construir um “produto” final ao longo das sessões de trabalho, a metodologia adotada inspira-se na forma de Exercício de Projeto apresentada por Helle, Tynjälä & Olkinuora (2006).

Mantendo como objetivo principal promover a aprendizagem das estruturas de controlo em programação, este desafio procura levantar a necessidade de as utilizar para resolver os problemas levantados no projeto. De outro modo, sem a utilização das mesmas (estruturas de controlo), não seria possível chegar ao objetivo proposto no enunciado do projeto (Anexo B – Enunciado Projeto Um Robô em Marte). Foi importante os alunos perceberem o que era suposto o robô fazer para assim sentirem eles próprios a necessidade de implementarem determinados “comportamentos” no robô. Este “comportamentos” e tomadas de decisão puderam ser refletidos com a utilização das respetivas estruturas de controlo acima indicadas (*if*, *if..else*, *(do..)while e for*).

Para as sessões de trabalho, foi também criada uma página<sup>4</sup> em moodle onde foram disponibilizados os seguintes recursos: enunciado do projeto, interface de programação do robô, cenário de aprendizagem “Experiência em Marte”, fórum de discussão, critérios de avaliação do projeto e uma sala de *Chat*.

---

<sup>4</sup> <http://moodle2.agcolares.org/course/view.php?id=5>

Para realizar o projeto, sendo a turma de 9 alunos, ficou definido a mesma dividir-se em grupos de trabalho de 2 ou 3 alunos.

Como as atividades de programação, a realizar pelos alunos, não exigem o robô permanentemente ligado ao computador, os 3 robôs disponíveis são suficientes para todos os grupos.

Para tentar ainda promover a reflexão das tarefas realizadas ao longo das sessões, por parte dos alunos, foram desenvolvidas orientações para os mesmos apresentarem o ponto de situação do projeto no final de cada sessão num diário de bordo. Este diário teve como objetivo abordar os seguintes pontos: Tarefas superadas e quais as estruturas de controlo utilizadas; Dificuldades encontradas e resolução das mesmas; Algoritmo ou Fluxograma atualizado do projeto. O diário de bordo pode ser consultado no Anexo M – Diários de Bordo.

Mesmo não havendo alunos identificados como tendo necessidades educativas especiais, houve conhecimento por parte do professor cooperante que alguns alunos tinham muitas dificuldades na programação revelando possivelmente capacidades de aprendizagem muito abaixo da média. Para precaver a necessidade de orientar grupos de trabalho menos autónomos, tentou-se dividir o projeto em pequenas componentes ordenadas pelo seu grau de complexidade. Estas componentes procuram também subdividir o projeto a implementar em pequenas partes isoladas de menor dificuldade. Neste sentido, caso alguns alunos/grupos revelassem dificuldades acrescidas na resolução do projeto, os mesmos seriam encaminhados a resolver cada uma das etapas apresentadas até chegarem à solução final. As diferentes etapas podem ser consultadas no Anexo H – Etapas do Projeto.

#### **4.1. Organização das Sessões de Trabalho**

De uma visão geral, as diversas sessões de trabalho foram planeadas para inicialmente os alunos se concentrarem na definição e construção de algoritmos funcionais utilizando os recursos disponibilizados.

Embora o fator competição tivesse sido apresentado logo ao início do projeto, o mesmo foi realizado à medida que soluções funcionais foram surgindo. O planeamento das diversas sessões pode ser consultado no Anexo J – Planeamento das Sessões de Trabalho.

## 4.2. Robô Adotado

Com base nos estudos apresentados acima, a plataforma *Arduino* apresentou-se como uma possível plataforma no ensino e aprendizagem da programação sendo que a mesma foi utilizada

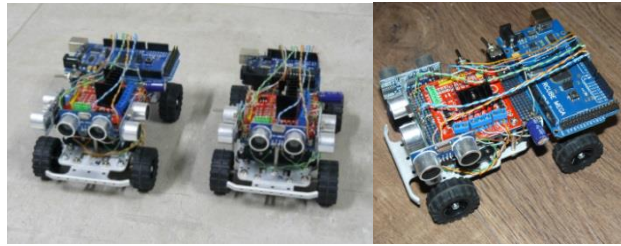


Figura 5 - Robôs

no presente projeto para controlar o robô. Toda a construção do robô foi feita pelo autor deste trabalho e pode ser consultada no Anexo E – O Robô *Arduino*. Para o efeito foram construídos 3 robôs idênticos.

Para a escolha da linguagem de programação do robô foram analisadas duas tecnologias, a plataforma nativa de programação *Arduino* e o *Scratch4Arduino*<sup>5</sup>. Optou-se por utilizar a primeira dado que disponibiliza a linguagem de programação *Arduino Language*<sup>6</sup> (baseada em Wiring), muito idêntica à linguagem de programação C/C++ até agora adotada pelo professor cooperante da turma neste projeto. Desta forma tentou-se assegurar uma certa continuidade partindo dos conhecimentos anteriormente adquiridos pelos estudantes.

De modo a harmonizar e compatibilizar todo o equipamento utilizado no robô, foi desenvolvido um conjunto de funções doravante chamada de API (*Application Programming Interface*) para que os alunos tivessem facilmente acesso às funcionalidades disponibilizadas pelo mesmo. Ou seja, dado que o *Arduino* é ligado aos outros equipamentos eletrônicos (motores, sensores e *beeper*), a manipulação direta dos mesmos poderia dificultar a tarefa dos alunos. Assim, a API especificamente desenvolvida para esta intervenção permite movimentar o robô, ler os valores dos sensores ultrassônicos e emitir um sinal sonoro invocando funções para o efeito. Todas as funções desta API estão descritas no Anexo I – API (*Application Programming Interface*) do Robô deste documento. Estas funções não inibem a necessidade de utilizar as estruturas de controlo abordadas no projeto, apenas facilitam a manipulação do robô.

<sup>5</sup> <http://seaside.citilab.eu/> (Plataforma de programação gráfica desenvolvida pela MIT)

<sup>6</sup> <http://arduino.cc/en/Reference/HomePage>

## 5. Intervenção

Este capítulo apresenta uma descrição sumária da intervenção realizada abordando diversas situações ocorridas em aula. Também se tenta apresentar as diversas práticas e interações professor-aluno aluno-professor, onde foram realizadas questões de reflexão revelando aos alunos possíveis problemas, soluções ou até mesmo novas questões.

Um pouco antes da primeira sessão, o professor cooperante informou o autor deste projeto que a intervenção não iria ser dada apenas a metade da turma, mas sim à turma inteira em simultâneo. Foi explicado ao professor cooperante que esta alteração chocava com o planeamento até agora definido, por outro lado, a sala tornava-se pequena para tantos alunos, os robôs poderiam ser insuficientes para todos os grupos interagirem e que a própria investigação associada à intervenção tinha como um dos seus propósitos identificar possíveis diferenças entre a aprendizagem da programação com e sem robôs. Foi explicado que para ser possível realizar uma experiência quasi-experimental, tal como tinha sido combinado, seriam necessários dois grupos sendo que um deles não poderia ter contacto com os robôs na programação. Também foi dito que todas as sessões de trabalho estavam preparadas para serem realizadas em 3 grupos de 2 alunos e 1 grupo de 3 sendo que esta escolha estava acompanhada de toda uma documentação teórica realizada no âmbito deste projeto o que não ficaria coerente com as novas condições apresentadas pelo professor que obrigavam a aumentar o número de alunos por grupo. Apesar desta explicação, o professor cooperante voltou a pedir que os dois turnos tivessem as sessões em conjunto justificando que, caso fosse realizado o contrário, o turno 2 poderia ficar melindrado de não poder trabalhar com robôs e que, por outro lado, teria que justificar este facto perante o conselho pedagógico.

Dadas as novas condições, foi necessário preparar os recursos, documentação em papel e grelhas de observação direta, para os restantes alunos. As cinco sessões acabaram por se realizar tal como o professor cooperante solicitou e estão descritas de seguida:

### **Sessão 1 - 18-02-2013:**

Deu-se início à sessão com os alunos a mostrarem curiosidade nos robôs logo à entrada em sala de aula. Tal como já tinham sido informados, foi-lhes dito que se

iria iniciar, naquele dia, um período de cinco sessões com programação em robôs. De forma a não colocar qualquer pressão sobre os alunos, derivada à mudança de condições inesperada, ou seja, envolvimento de todos os alunos, foi explicado aos mesmos que embora inicialmente se tivesse planejado este projeto para apenas os alunos do primeiro turno trabalharem com robôs, toda a turma acabou por ser envolvida em simultâneo.

Antes de se iniciar qualquer atividade, foi pedido aos alunos para preencherem um formulário (pré-teste) de diagnóstico, com o intuito de tentar perceber qual o nível de conhecimento dos alunos acerca das estruturas de controlo antes da intervenção.

De seguida a construção dos grupos foi então reformulada sendo que foram criados 4 grupos com 3 alunos e 1 grupo de 4 alunos.

Para envolver os alunos, foi apresentado o Cenário de Aprendizagem<sup>7</sup>, “convidando” os alunos a assumirem-se como os programadores responsáveis por esta missão em Marte. Para tornar possível esta implementação na sala de aula foi apresentada a arena que iria simular o local alienígena. Com a ajuda de um robô na arena, foi explicado os objetivos do projeto e, simultaneamente, também foi distribuído aos alunos o Cenário de Aprendizagem (Carroll, 2000) e o Enunciado do trabalho prático em papel. Surgiram algumas dúvidas as quais foram esclarecidas. Também foi dito aos alunos que toda a informação que estava a ser distribuída também se encontrava disponível numa página no *Moodle* criada para o efeito a qual os alunos tinham acesso.



Figura 6 - Arena

Após esta fase foi necessário mostrar aos alunos os procedimentos para programar os robôs. Para isso foi apresentado um pequeno exemplo que realizava um pequeno movimento para a frente do robô.

---

<sup>7</sup> Com base no conceito de Cenário de Aprendizagem de Carroll (2000) apresentado no Enquadramento Teórico.

Após dar-se o início dos trabalhos, por parte dos alunos, alguns grupos manifestaram dificuldades na manipulação do robô dizendo que não entendiam como poderiam fazer o mesmo movimentar-se ou efetuar a leitura de sensores. Aqui apresentou-se com mais detalhe o funcionamento do mesmo explicando calmamente da API disponibilizada. Os alunos prosseguiram com a implementação de pequenas operações as quais puderam ver realizadas pelos robôs.

Após cada grupo conseguir efetuar alguns movimentos nos robôs, tentou-se guiar os mesmos dando algumas orientações no sentido de dotarem os seus algoritmos da capacidade de detecção de objetos para assim o robô não colidir com objetos à sua volta. Alguns alunos perguntaram como poderiam relacionar os sensores com a movimentação do robô. Após questionados, os alunos manifestaram o conhecimento de algumas estruturas de controlo, contudo aparentavam não identificar a necessidade da utilização das mesmas perante uma determinada situação real, neste caso em concreto verificação da distância de objetos. De facto, nenhum aluno parecia estar a identificar a necessidade de utilizar, por exemplo, uma estrutura *if* para verificar se o robô poderia avançar, lendo para isso o sensor frontal. Para tentar promover a descoberta autónoma de soluções foi necessário mudar de estratégia. Foi então realizada uma discussão com os alunos onde foram colocadas e postas em discussão as seguintes questões à turma: quais são as estruturas de controlo que conhecem?; quais delas permitem executar um determinado conjunto de operações perante uma condição?; como fazem para testar o valor de uma variável?; que código podemos usar para ver se um sensor tem um valor superior a 30 centímetros?. Nesta fase alguns alunos apresentaram algumas ideias que pareciam estar corretas e fazer sentido. As equipas voltaram a testar e a alterar os seus algoritmos implementando alguns ciclos de controlo tais como *if* e *do..while*.

Chegando ao final da sessão, foi pedido para testarem os robôs uma última vez e para preencheram individualmente os diários de bordo. Estes registos, assim como as respostas ao questionário inicial foram analisados pelo autor do projeto para orientar as sessões seguintes.

## **Sessão 2 - 20-02-2013:**

A segunda sessão iniciou-se com a realização de um pequeno *briefing* no sentido de recapitular as tarefas realizadas, dificuldades e soluções encontradas pelos grupos na

sessão anterior. Em discussão com cada grupo foi realizado um ponto de situação do estado atual do projeto onde também foram dadas algumas orientações para prosseguirem.

Os diversos grupos começaram rapidamente a desenvolver os seus algoritmos e a realizar testes com os robôs na arena. Ao longo da sessão, com base nas informações retiradas dos diários de bordo realizados pelos alunos na primeira sessão, bem como do formulário pré-teste, foram realizadas algumas discussões com os mesmos de modo a tentar esclarecer e eliminar algumas dificuldades relacionadas com as estruturas de controlo transportando esses conceitos em possíveis funcionalidades a implementar no robô.

Os grupos foram implementando diferentes soluções, contudo foi revelado pelos mesmos que a dificuldade mais frequente estava na verificação da possibilidade de movimentar o robô fazendo a leitura dos sensores.

O grupo número 1 (4 alunos: A14, A7, A9 e A4), que inicialmente se apresentou bastante confiante, não estava a conseguir evitar colisões. Em discussão com o grupo promoveu-se alguma reflexão acerca das condições lógicas utilizadas. Após essa análise, os alunos verificaram que as mesmas não estavam construídas corretamente. O código desta solução foi então reformulado.

O grupo número 2 (3 alunos: A1, A8 e A10), apresentou uma estrutura condicional composta encadeada bem construída e aparentemente funcional que efetuando a leitura dos sensores dentro das condições das cláusulas *if*, despoletavam a movimentação do robô. Contudo, todas estas estruturas estavam dentro de um ciclo de repetição *do..while* cuja condição de paragem não se encontrava corretamente elaborada o que impedia o funcionamento desejado. Encorajou-se os alunos a continuarem o seu raciocínio lógico mas para terem cuidado com as condições de paragem no ciclo de repetição. Posteriormente verificou-se que os alunos acabaram por retirar o ciclo de repetição. Foi-lhes dito que esse ciclo poderia ser útil mas poderiam, numa fase inicial, testar apenas o funcionamento das estruturas condicionais na movimentação do robô e só depois acrescentar a condição de paragem final.



O grupo número 3 (3 alunos: A13, A11 e A15) continuou a apresentar algumas dificuldades na construção e utilização de estruturas *if*. Embora tenham sido um dos primeiros grupos a identificar a necessidade de utilizar estas estruturas, não estavam a conseguir utilizar as mesmas de forma eficaz. Após uma breve explicação da sintaxe das estruturas condicional e condicional composta, o grupo iniciou a implementação autónoma de um algoritmo mais elaborado.

O grupo número 4 (3 alunos: A5, A2 e A12) não optou por uma abordagem de tentativa e erro. Estes alunos passaram quase toda a aula a tentar organizar ideias implementando um algoritmo dentro das suas dificuldades ou limitações. Realizando uma pequena discussão, tentou-se orientar os trabalhos de modo a não confundirem as diversas ideias presentes. Era importante clarificarem as mesmas transformando-as em comportamentos do robô, para de seguida implementarem cada uma de forma clara no algoritmo. O grupo dividiu as diferentes ideias separando as mesmas em pequenos blocos de código delimitados por estruturas condicionais.

O grupo número 5 (3 alunos: A3, A6 e A16) estava um pouco atrasado nos trabalhos. O aluno A16 faltou a esta sessão. Estes alunos pareciam estar com muitas dificuldades na manipulação controlada do robô. Em discussão, os alunos afirmaram conhecer a estrutura de controlo *if*, mas não sabiam para que servia nem como se poderia utilizar. Com os restantes grupos a trabalhar de forma autónoma, optou-se, nesta sessão, por dar um pouco mais atenção e apoio individualizado aos alunos deste grupo. Foi pedido ao grupo que implementasse alguns movimentos básicos do robô para perceberem a sequência lógica da aplicação, ver Anexo H – Etapas do Projeto. Após esta fase os alunos evidenciaram que assim o robô andava, mas batia em todo lado e questionaram como poderiam utilizar os sensores para detetar obstáculos. Aqui, após os alunos identificarem o conceito necessário, foi-lhes apresentado a estrutura de controlo acesso condicional simples sendo que poderiam utilizá-la para relacionar um comportamento desejado de acordo com um teste lógico realizado aos sensores. Depois desta aplicação foi-lhes pedido que refletissem um pouco para conceptualizarem um possível algoritmo. Após alguns minutos o grupo apresentou uma solução funcional que apenas fazia avançar o robô se o sensor frontal não tivesse obstáculos a menos de 30 cm. Após esta tarefa superada solicitou-se que introduzissem movimentos laterais com a ajuda dos respetivos sensores.

Chegando-se ao final da sessão foi pedido a todos os alunos para preencherem o diário de bordo, mas desta vez de forma mais rigorosa para identificarem melhor o comportamento dos seus robôs de acordo com os algoritmos implementados.

### **Sessão 3 - 21-02-2013:**

No início da terceira sessão foi solicitado aos alunos para identificarem rapidamente as principais dificuldades encontradas até ao momento, para se realizarem pequenas discussões em grupo com o professor, autor deste projeto, com objetivo de resolver essas mesmas dificuldades. Foi realizada uma passagem por cada grupo onde se tentou promover a discussão interna de possíveis resoluções aplicáveis aos seus problemas. As diversas ideias dos alunos foram discutidas e testadas nos robôs. Apelou-se à turma para até ao final da aula tentarem desenvolver versões funcionais dos algoritmos para iniciarem o quanto antes pequenos duelos de robôs.

O grupo número 5, alvo de especial atenção, apresentou-se um pouco confuso relativamente à solução por eles implementada na sessão anterior. Também sem o elemento A16, por o mesmo se encontrar ainda a faltar, foi realizada uma discussão de modo a recapitular os conteúdos até agora abordados e aplicados no algoritmo. Nesta discussão confrontaram-se os alunos com as seguintes questões: qual era o comportamento que tinha o vosso robô?; Se o robô só andava para a frente quando tinha o caminho livre, o que é que foi utilizado?. Depois de algumas trocas de ideias o grupo voltou a sentir-se enquadrado no seu próprio trabalho continuando o mesmo.

O grupo número 2 que tinha inicialmente implementado uma solução clara e funcional estava agora com dificuldades porque o robô efetuava poucos movimentos. Perante esta situação apelou-se aos alunos para aumentar o tempo de movimentação do robô e verificar quais as alterações de comportamento do mesmo. Rapidamente concluíram que, por lapso, estavam a utilizar tempos de movimentação demasiados curtos o que resultava em movimentos pouco significativos.

O grupo número 3 apresentou um algoritmo com algumas falhas e com movimentos muito repetitivos. Em conversa com os alunos, verificou-se que o robô necessitava de se movimentar em mais direções, neste caso, para a esquerda. Prosseguiram de seguida com o melhoramento do mesmo.

Em discussão com o grupo número 4 verificou-se que a movimentação dada ao robô ainda não estava coerente com os testes realizados sobre os sensores através das estruturas *if..else*. Pediu-se ao grupo para organizarem as ideias e só posteriormente refletirem as mesmas no algoritmo.

O grupo número 1 tinha um algoritmo relativamente funcional mas afirmava estar a afinar os tempos de movimentação para otimizar o deslocamento do robô. Foi sugerido ao grupo a utilização de uma função para criar números aleatórios para assim movimentar o robô aleatoriamente no espaço. Os alunos não entendiam como poderiam usar números aleatórios para gerar conseqüentemente movimentos aleatórios, pelo que foi pedido aos alunos para refletirem sobre o assunto e tentarem utilizar as estruturas de controlo que já conheciam. Após alguns minutos foram surgindo algumas soluções com estruturas condicionais *if* e estruturas de repetição *while*.

Ao longo de toda a aula e com base nas informações obtidas da análise dos diários de bordo foi-se discutindo com todos os alunos a fim de esclarecer as dúvidas identificadas e ultrapassar algumas dificuldades reveladas nos seus textos e algoritmos.

Embora, nesta sessão, se tenham realizado alguns duelos, os alunos disseram que ainda não tinham algoritmos suficientemente eficientes para iniciar a competição, remetendo a mesma para a próxima aula.

Foi pedido aos alunos para enviarem as suas soluções para o autor deste projeto para se fazer uma avaliação intermédia dos trabalhos.

No final da aula foi pedido aos alunos para preencherem novamente o diário de bordo, mas desta vez debruçando-se com mais detalhe na explicação do algoritmo e estruturas de controlo utilizadas.

#### **Sessão 4 - 25-02-2013**

Esta sessão consistiu principalmente no melhoramento da eficiência das soluções encontradas. Embora ainda tenha havido alguns grupos a introduzir novas funcionalidades nos algoritmos, concluiu-se que a generalidade das tarefas realizadas

se resumiu a pequenas afinações nos movimentos previamente implementados. Em discussão em plenário com a turma também foi-lhes pedido para tentarem diversificar as movimentações até agora implementadas no sentido de tornar o comportamento do robô menos previsível.

Chamou-se a atenção aos alunos para lembrarem os critérios de avaliação disponibilizados na página de *Moodle* e para se prepararem para, na próxima aula, realizarem uma auto e heteroavaliação dentro do grupo sendo que as grelhas a preencher poderiam ser consultadas na mesma página.

Nesta sessão o aluno A16 continuava a faltar. Poderia ter-se realizado alterações nos grupos passando um elemento do grupo 1 para o grupo 5. Contudo pensou-se que este tipo de alteração poderia comprometer o bom funcionamento dos grupos e empenho dos alunos. Optou-se assim por não fazer alterações deste tipo. Para tentar precaver o possível desânimo do grupo 5 pela perda de um elemento, aumentaram-se os reforços positivos e apoio individualizado ao longo das sessões.

Ao longo de toda a sessão, com base na informação veiculada pelos diários de bordo bem como nas soluções enviadas pelos alunos, foram realizadas discussões acerca das dificuldades ainda persistentes no sentido de tentar colmatar as mesmas.

Nesta sessão, após pequenos ajustes, o grupo número 1 já apresentou um algoritmo totalmente funcional que após vários duelos com outros robôs acabou sempre por ganhar. Os algoritmos dos grupos 2 e 3 apresentavam-se funcionais contudo demoravam sempre demasiado tempo até alcançar o objetivo final, o ponto com alto-relevo na arena. Tentou-se reforçar positivamente os grupos para otimizarem os algoritmos.

Continuaram a realizar-se diversos duelos entre todos os grupos, verificando-se que, por vezes, alguns grupos adotaram uma estratégia de tentativa e erro para otimizar o robô. Em discussão com eles, tentou-se promover a compreensão efetiva dos comportamentos induzidos nos robôs, para assim desenvolverem soluções de forma mais controlada e consciente.

Chegando ao final da sessão solicitou-se aos alunos para preencherem o diário de bordo realçando o detalhe no algoritmo e principalmente as otimizações conseguidas.

## **Sessão 5 - 27-02-2013**

No início desta sessão o autor deste projeto pediu aos alunos para fazerem os últimos testes e prepararem as versões finais das suas soluções, para de seguida se realizar o campeonato decisivo.

Foi realizado o campeonato onde cada grupo defrontou todos os outros sendo que se obteve a seguinte classificação final: 1º Lugar- Grupo 1; 2º Lugar- Grupo 3; 3º Lugar- empate entre os grupos 4 e 2; e 4º Lugar- Grupo 5. Foram dados os parabéns aos grupos com mais sucesso, mas também foi explicado que os grupos com classificação mais baixa também tiveram boas prestações e deviam continuar a utilizar o que aprenderam nas próximas aulas.

Foi pedido aos alunos para preencherem pela última vez o diário de bordo explicando as versões finais dos algoritmos implementados. De seguida foi pedido aos alunos para preencherem o formulário (pós-teste) de verificação de conhecimentos no final desta intervenção.



## **6. Aprendizagens Ocorridas e Avaliação do Projeto**

Este capítulo procura apresentar as aprendizagens ocorridas nos alunos ao longo das cinco sessões, bem como apresentar os resultados dos pré e pós-testes.

Segundo Zabalza (1992) citado por Gaspar & Roldão (2007) e Gomes (2012), a avaliação centraliza-se nos resultados e nos processos que conduzem aos mesmos e ocorre durante o desenvolvimento curricular para regular o processo de ensino-aprendizagem.

Neste estudo, a avaliação divide-se em duas componentes: avaliação dos alunos e da solução/algoritmo desenvolvido; e avaliação das aprendizagens ocorridas ao longo do projeto. Para a avaliação dos alunos e do projeto foram elaborados e preenchidos os seguintes instrumentos: observação direta dos alunos; diário de bordo; avaliação da solução desenvolvida; e auto e heteroavaliação. Para a avaliação das aprendizagens ocorridas foram desenvolvidos formulários ligados à componente investigativa deste estudo e apresentados no ponto 6.3. Avaliação do Projeto. Adicionalmente os alunos também foram solicitados a responder a um questionário de avaliação do professor e a questões gerais acerca da utilização de robôs no ensino da programação. Para garantir a confidencialidade dos dados recolhidos todos os nomes dos alunos foram omitidos.

### **6.1. Instrumentos de Avaliação dos Alunos.**

A grelha de observação direta, para avaliação individual dos alunos, aborda 6 dimensões: análise, desenvolvimento do projeto, capacidade técnica e científica, planificação, auto e heteroavaliação e relacionamento interpessoal. Este instrumento foi preenchido para cada aluno com base nas diversas observações e interações professor-aluno e aluno-aluno realizadas ao longo das sessões de trabalho, e pode ser consultado no

Anexo K – Grelha de Observação Direta. De forma resumida apresenta-se um gráfico com as notas individuais:

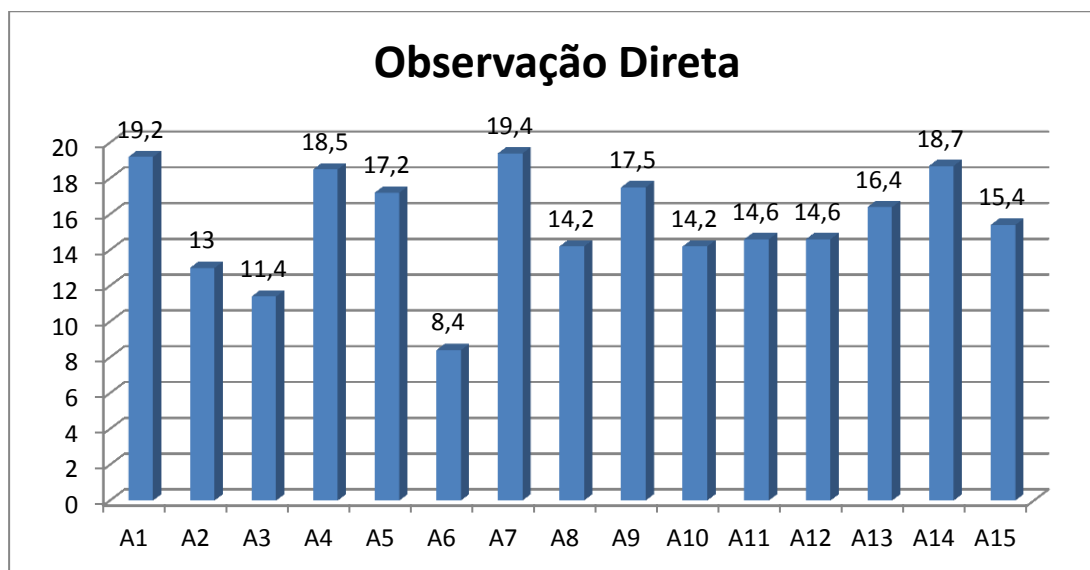


Gráfico 1 - Observação Direta Individual dos Alunos

Os diários de bordo tiveram um papel importante para se analisar a percepção dos alunos relativamente ao seu trabalho. Sendo preenchidos no final de cada sessão, os alunos foram solicitados a apresentar as soluções desenvolvidas, dificuldades encontradas e algoritmos implementados. Dado que cada exemplar foi realizado individualmente por sessão, apenas se apresentam 3 digitalizados disponíveis no Anexo M – Diários de Bordo.

Os critérios de avaliação da solução/algoritmo implementado pelos alunos analisam 3 dimensões: a organização do código, algoritmo e etapas do desafio. A organização do código refere-se ao sentido das ideias subjacentes ao longo do algoritmo e ao encadeamento lógico das mesmas. Os critérios de avaliação referentes ao algoritmo analisam a eficiência e eficácia da solução. Por fim, a componente de etapas do projeto refere-se à globalidade dos objetivos desejados, como por exemplo a sinalização de fim de missão com sinal sonoro. Os resultados das avaliações para cada grupo de alunos podem ser consultados no Anexo N – Avaliação do Produto e apresentam-se de forma resumida na seguinte tabela:



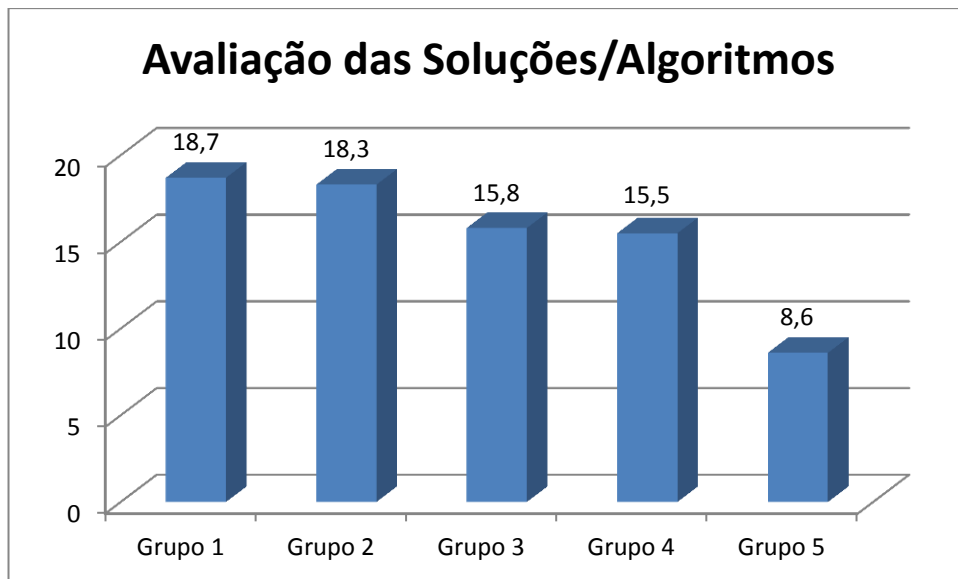


Gráfico 2 - Avaliação das Soluções

A componente de auto e heteroavaliação, preenchida pelos alunos, tem como propósito os mesmos apresentarem a sua avaliação acerca deles próprios e dos colegas de grupo. Este instrumento contempla duas dimensões: comportamentos e atitudes; e competências técnicas. Este instrumento, devidamente preenchido pelos alunos, pode ser consultado no Anexo L – Auto e Heteroavaliação.

## 6.2. Abordagem Metodológica

Procurou-se saber em que medida a utilização de robôs pode promover a aprendizagem de estruturas de controlo na programação informática em alunos de nível secundário. Inicialmente planeada para ser um estudo do tipo quasi-experimental, a turma foi dividida em dois grupos, turno 1, amostra do estudo a frequentar a intervenção, e turno 2, grupo de controlo não presente na mesma. Para verificar as diferenças de aprendizagem entre os dois turnos foram também elaborados um pré e um pós-teste, devidamente validados por 5 pessoas na área. Contudo, umas horas antes da intervenção, o professor cooperante solicitou que toda a turma frequentasse a intervenção pedagógica para que todos os alunos pudessem ter contacto com robôs. Assim, em detrimento da questão investigativa foi respeitada a componente pedagógica indo ao encontro da vontade do professor cooperante. Neste sentido, mantiveram-se os pré e pós-testes sendo que a amostra teve como elementos toda a turma, não existindo grupo de controlo.

Os inquéritos pré e pós-teste foram compostos pelas mesmas questões, sendo que apresentadas por uma ordem diferente. A temática das questões divide-se nas 3 estruturas de controlo utilizadas no projeto, *if/if..else*, *for* e *while/do..while*. De modo a aceitar respostas com algum grau de incerteza, por parte dos alunos, as mesmas foram dadas numa escala Likert de concordância de 5 pontos: concordo totalmente, concordo parcialmente, mediano, discordo parcialmente e discordo totalmente.

O pós-teste também incluiu questões de carácter geral para obter a opinião dos alunos acerca da utilização de robôs e da prestação do professor.

Para a concretização deste estudo foram realizados os respetivos pedidos de autorização aos elementos envolvidos, nomeadamente ao diretor da escola e encarregados de educação dos alunos.

### **6.3. Avaliação do Projeto**

Este capítulo apresenta os resultados dos instrumentos de avaliação do projeto comparando os valores pré e pós-teste, bem como os dados do inquérito geral da intervenção.

Os inquéritos pré e pós-teste procuraram identificar os conhecimentos dos alunos antes e depois da intervenção. Cada uma das questões, apresentadas de seguida, abordou uma das três estruturas de controlo:

1. A estrutura de controlo *if* permite testar uma condição para decidir se um determinado conjunto de operações são executadas ou não.
2. De acordo com a veracidade da condição introduzida, uma estrutura de controlo *if..else* permite executar um de dois conjuntos de operações distintos.
3. É possível criar um ciclo de repetição apenas utilizando uma estrutura *if..else*.
4. Uma estrutura de controlo *while* permite repetir um determinado conjunto de operações enquanto a condição especificada for falsa.
5. Uma estrutura de controlo *while* permite repetir um determinado conjunto de operações enquanto a condição especificada for verdadeira.

6. Imagina que estás a construir programa informático que acende uma luz sempre que alguém carregar num botão. Para construir este programa poderás utilizar uma das estruturas *while* ou *do..while*.

7. Para verificar se um número é maior do que 6 deve-se utilizar um ciclo *for*.

8. Se eu quiser construir um programa que acenda e apague uma lâmpada 20 vezes seguidas, posso utilizar um ciclo *for*.

9. Uma estrutura de controlo *for* permite, por exemplo, executar um determinado conjunto de operações, um determinado número de vezes.

10. Para verificar se um número é menor do que 5 pode-se utilizar uma estrutura de controlo *if*.

O resumo dos resultados das respostas pré-teste são apresentados na tabela abaixo, sendo que as respostas mais corretas estão assinaladas com “\*”:

Tabela 1 - Respostas do Pré-teste

Estrutura / Resposta	if	if.. else	if.. else	while	while	while/ do..while	for	for	for	if
Concordo totalmente	7*	4*	1	4	6*	5*	2	8*	6*	9*
Concordo Parcialmente	3	2	2	2	1	6	2	2	5	2
Mediano	2	5	3	2	2	1	4	1	1	1
Discordo Parcialmente	1	1	0	1	1	1	1	2	1	1
Discordo Totalmente	0	1	7*	4*	3	0	4*	0	0	0
% da resposta mais correta	54%	31%	54%	31%	46%	38%	31%	62%	46%	69%
% das duas respostas mais corretas	77%	46%	54%	38%	54%	85%	38%	77%	85%	85%

Apresenta-se de seguida uma tabela com o resumo dos resultados das respostas do pós-teste, sendo que as respostas mais corretas também se encontram assinaladas com “\*”:

Tabela 2 - Respostas do Pós-teste

Estrutura / Resposta	if	if.. else	if.. else	while	while	while/ do..while	for	for	for	if
Concordo totalmente	7*	5*	3	3	6*	8*	1	6*	6*	9*
Concordo Parcialmente	3	6	3	2	4	4	4	5	4	2
Mediano	2	2	2	2	1	0	1	0	1	2
Discordo Parcialmente	0	0	0	1	0	0	0	1	0	0
Discordo Totalmente	1	0	5*	5*	2	1	7*	1	2	0
% da resposta mais correta	54%	38%	38%	38%	46%	62%	54%	46%	46%	69%
% das duas respostas mais corretas	77%	85%	38%	46%	77%	92%	54%	85%	77%	85%

Confrontando a percentagem da resposta mais correta em cada questão nos dois momentos obtemos o seguinte gráfico:

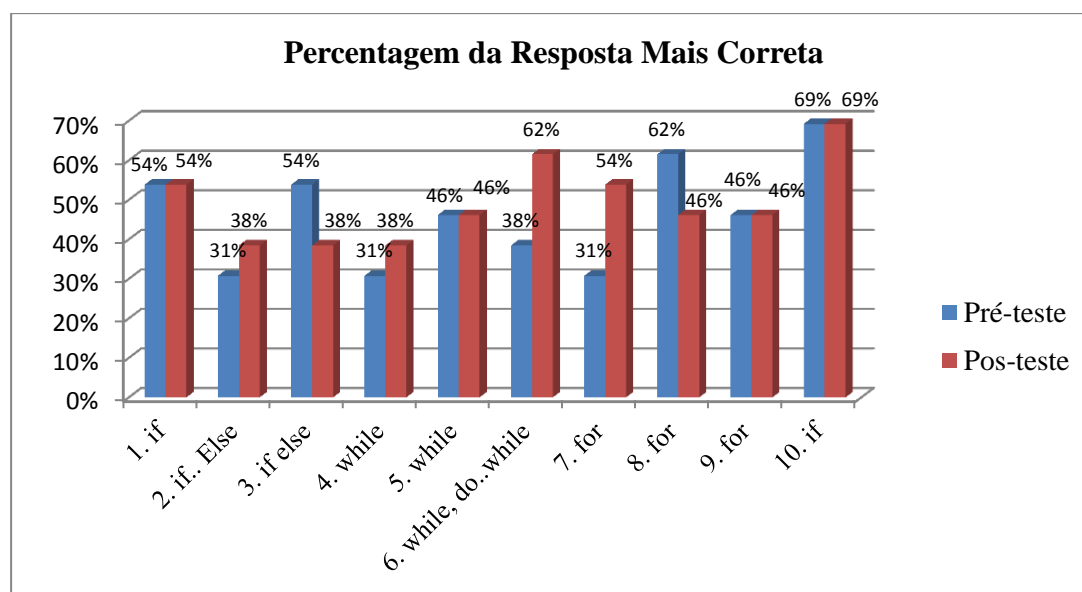


Gráfico 3 - Percentagem da resposta mais correta

Confrontando também a percentagem das duas respostas mais corretas em cada questão nos dois momentos obtemos o seguinte gráfico:

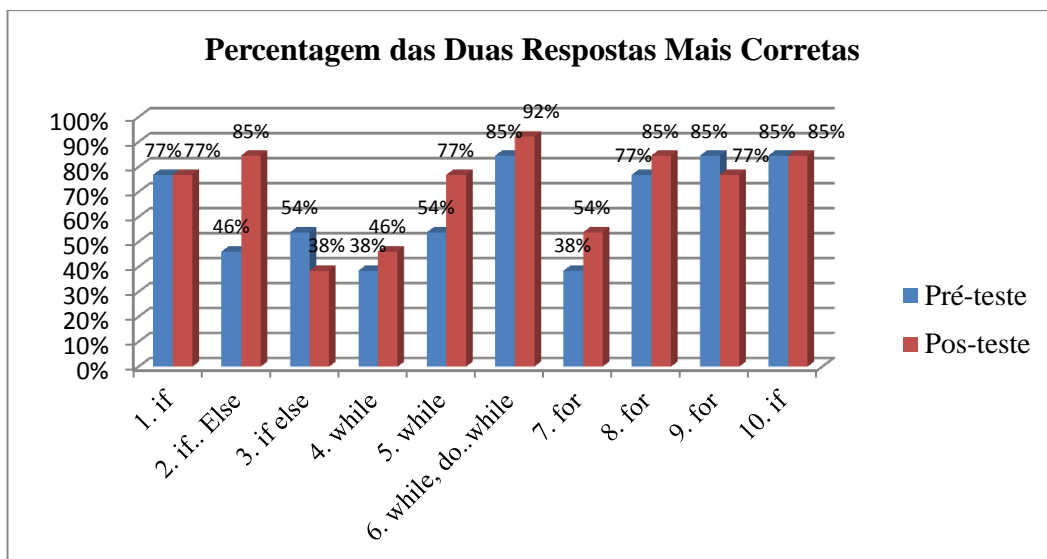


Gráfico 4 - Percentagem das Duas Respostas Mais Corretas

Os mesmos dados podem ser agrupados por tipo de estrutura de controlo sendo que se obtém a seguinte relação por uma ou duas respostas mais corretas, respetivamente:

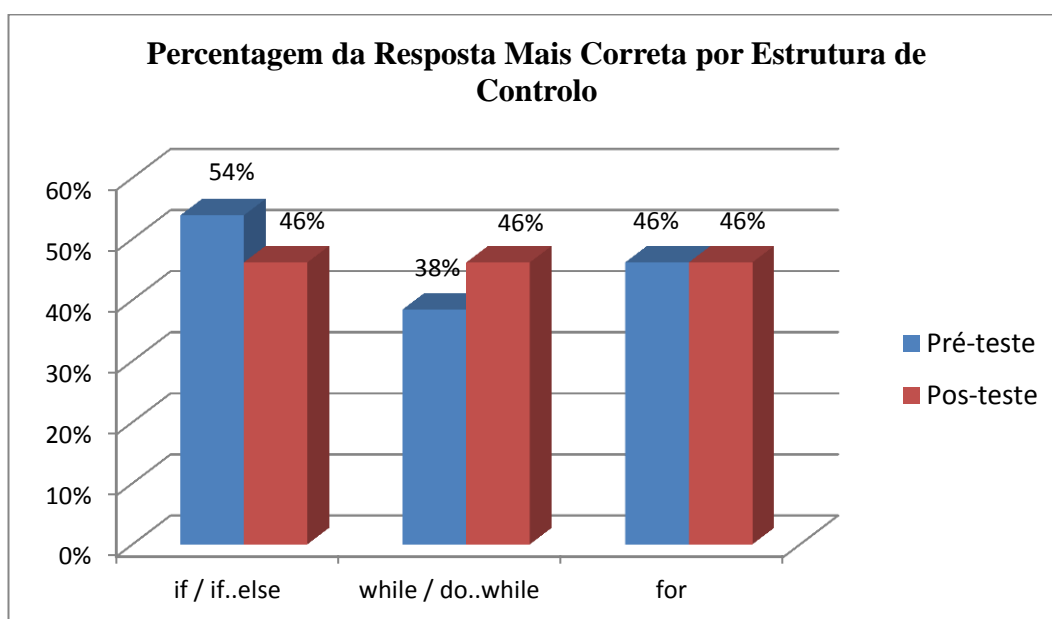


Gráfico 5 - Percentagem da Resposta Mais Correta por Estrutura de Controlo

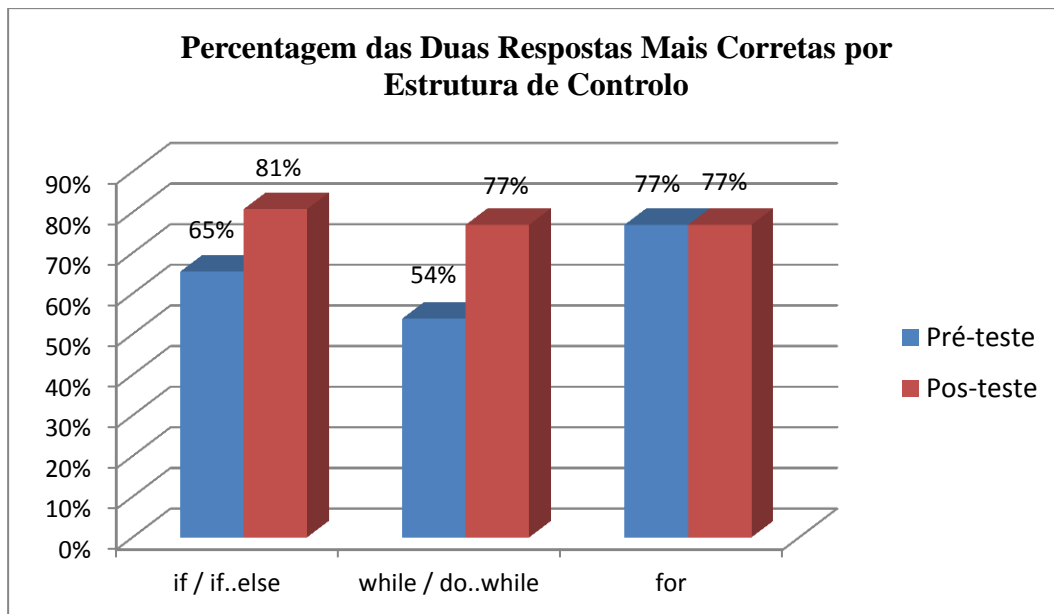


Gráfico 6 - Percentagem das Duas Respostas Mais Corretas por Estrutura de Controlo

O questionário geral e avaliação do professor foi diluído no pós-teste e tem como objetivo verificar qual a opinião dos alunos acerca da prestação professor, bem como da utilização de robôs no ensino da programação. Com a exceção da apresentação de sugestões, todas as questões também foram apresentadas sob a forma de afirmações as quais foram respondidas numa escala de Likert de concordância de 5 pontos. Apresentam-se de seguida as mesmas:

1. Esta intervenção contribuiu para a tua aprendizagem e compreensão das estruturas de controlo *if*, *if..else*, *for*, *while* e *do..while*.
2. A programação com robôs realizada nestas aulas ajudou-te a perceber melhor as estruturas de controlo *if*, *if..else*, *for*, *while* e *do..while*.
3. Sentiste-te mais envolvido e preferes a programação com robôs relativamente à programação tradicional.
4. O modo como o professor deu as aulas foi correto para aprender os conteúdos de programação aqui abordados.
5. O Professor explicou os conteúdos e tirou todas as dúvidas de forma clara.
6. O Professor teve um excelente relacionamento com os alunos.

7. Apresenta aqui qualquer sugestão que consideres importante relativamente à intervenção (professor, funcionamento das aulas, robôs,...).

Apresenta-se abaixo o resumo dos resultados a essas questões.

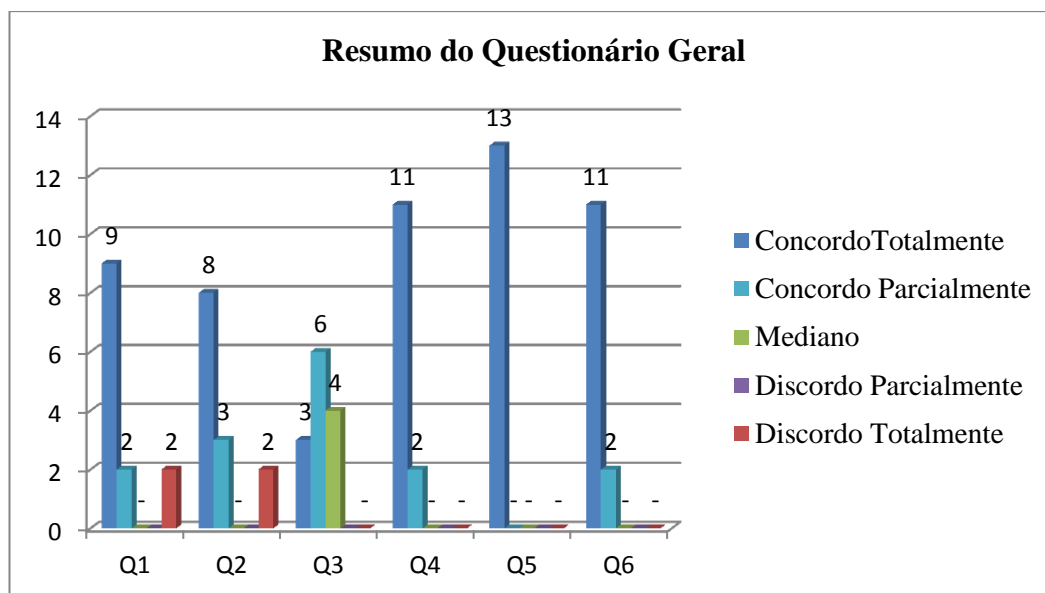


Gráfico 7 – Resumo do Questionário Geral

Dado que a questão número 7 é de resposta aberta não obrigatória apresentam-se as seguintes respostas transcrevendo o texto original dos alunos:

- “foi bom para aprender melhor programação, e a ficar mais a vontade com os vários estilos (*if...else,do,while etc*).”;
- “Gostei muito desta experiência, acho que foi útil até a nível de aplicar-mos os nossos conhecimentos num " projecto " real.”;
- “O professor explicou os conteúdos e tirou todas as dúvidas de forma clara, nas aulas foram "divertidas" e que foi mais fixe, foi que nos ganhamos”;
- “Gostei, porque serviu para testarmos o nosso conhecimento.”;
- “o professor teve um bom relacionamento com os alunos ... o funcionamento das aulas foi bom ... e os robôs também .”;
- “devia-mos mais aulas”;
- “As explicações do professor foram boas”;

#### **6.4. Análise de Resultados**

Este capítulo procura fazer uma análise dos resultados obtidos nos instrumentos de avaliação dos alunos e deste projeto. Embora se tenha procurado realizar uma avaliação correta e objetiva, poderá sempre existir uma componente subjetiva que reside na forma e capacidade de avaliar do Professor autor deste projeto. Os resultados dos pré e pós-teste

Relativamente à avaliação dos alunos, os resultados dos diversos instrumentos de avaliação parecem revelar algum sucesso, já que a maioria das notas obtidas se encontra acima de 14 valores, como podemos ver no Gráfico 1 – Observação Direta dos Alunos. Contrariamente à média geral, os alunos A2, A3 e A6 parecem necessitar de mais tempo para aperfeiçoarem os seus conhecimentos acerca dos conteúdos abordados. Estes alunos apresentaram também poucas bases de programação e dificuldades de aprendizagem acima da média.

Os alunos A14, A7, A9 e A4, do grupo 1, mostraram-se bastante confiantes e interessados. Por diversas vezes discutiram diferentes soluções as quais foram transpostas para o algoritmo. Inicialmente um pouco confusos na forma de programar o robô, apresentaram finalmente um algoritmo simples e eficaz. As ideias subjacentes no algoritmo foram apresentadas de forma clara e organizada.

No grupo 2 destaca-se o aluno A1 relativamente aos alunos A8 e A10. Extremamente organizado e autónomo, embora com autoestima aparentemente baixa, foi claramente o líder do grupo. Os alunos A8 e A10 apresentam conhecimentos técnicos muito semelhantes nomeadamente uma baixa autonomia. Os três alunos apresentaram um algoritmo simples e eficaz, contudo menos eficiente do que o grupo 1 uma vez que introduz movimentos laterais sempre que existe espaço para tal. Desta forma a translação do robô ao longo da arena realizava-se mais lentamente.

Os alunos A13, A11 e A15, do grupo 3 apresentam características semelhantes entre eles. O aluno A15 destaca-se pela sua iniciativa e interesse despertado no projeto. O algoritmo do grupo 3 faz o robô desviar-se da maioria dos obstáculos, contudo apresenta alguns movimentos que podem fazer colidir o robô. Contém também de certa forma a repetição de ideias que o tornam um pouco confuso. Apesar disto o robô obteve a segunda posição na classificação geral.



Os alunos do grupo 4 têm níveis de competência distintos. O aluno A5 tem alguma autonomia e boas capacidades de trabalho. O aluno A12 embora inicialmente com poucas bases mostrou uma grande evolução do longo das sessões. O aluno A2 apresentou-se repetidamente um pouco confuso com as soluções apresentadas pelos colegas o que despertou algum apoio individualizado por parte do professor. O grupo 4 apresentou um algoritmo que introduz movimentos laterais que podem fazer colidir o robô com objetos à sua volta.

Os alunos do grupo 5 apresentaram mais dificuldades que os outros. Inicialmente composto por 3 elementos, o aluno A16 apenas frequentou a primeira sessão. Pensou-se que o mesmo iria regressar e não foram realizadas alterações nos grupos. Contudo o mesmo não aconteceu mantendo-se o grupo com dois elementos. Estes alunos com pouca autonomia e baixos conhecimentos técnicos de programação, revelaram também alguma falta de bases, pelo que foram alvo de bastante reforço positivo e apoio individualizado que, apesar de tudo, se concretizou numa evolução bastante positiva. O aluno A3 mostrou sempre muito interesse o que o ajudou a superar algumas das suas dificuldades. A evolução demonstrada pelo mesmo demonstrou que apenas precisava de mais tempo para chegar ao nível dos outros. O aluno A6, menos interessado, apresentava-se pouco preocupado mas ia colaborando nos trabalhos. Embora a solução apresentada pelo grupo tenha sido a menos desenvolvida, acabou por ser uma vitória tendo em conta o ponto de partida dos alunos.

De forma quase generalizada, embora inicialmente alguns grupos possam ter utilizado o ciclo *for*, o mesmo não foi mantido nas versões finais dos algoritmos. Esta evidência pode dever-se ao facto deste ciclo de repetição provocar a reiteração de um conjunto de instruções um determinado número de vezes, inicialmente definido, sendo que o desafio em causa não contemplava forçosamente este tipo de necessidade. Desta forma os alunos puderam ter distinguido as diferentes utilizações dos dois tipos de ciclo *for* e *while*, contudo as estatísticas não apresentam a ocorrência de aprendizagem da estrutura *for*.

De acordo com as questões gerais acerca da intervenção, a maioria dos alunos afirmou que a intervenção, com robôs, contribuiu para a aprendizagem das estruturas de controlo em programação. No mesmo sentido, a maioria dos alunos também

admite estar mais envolvido nas aulas com robôs. A grande maioria dos alunos também considera que o professor teve uma prestação adequada no ensino da programação, esclarecendo todas as dúvidas de forma clara e tendo também um muito bom relacionamento com os alunos. Observando algumas sugestões deixadas pelos alunos no questionário, é manifestada a ideia que a utilização de robôs foi do agrado de muitos dos alunos da turma. O inquérito geral vai ao encontro das opiniões de Gomes (2012), Varanda (2012), Costa (2012) e Bernardo (2012) que referem diversas potencialidades no uso de robôs no ensino da programação.

Mesmo após serem realizados vários contactos, dois alunos não responderam ao pós-teste. Dado este facto, para garantir a fiabilidade do estudo, os dados estatísticos incluídos nos gráficos (referência cruzada) não contemplam as respostas dadas pelos mesmos no pré-teste.

Embora não existam resultados muito significativos, as estatísticas observadas nos pré e pós-teste, para cada questão, mostram que houve alguma evolução, que se percebe melhor quando observamos os gráficos estatísticos das duas respostas mais corretas. No mesmo sentido também se pode verificar alguma evolução nos gráficos 5 e 6 quando os dados são agrupados por estrutura de controlo. Aí podemos verificar que houve alguma aprendizagem das estruturas de controlo *if/if..else* e *while/do..while* não se verificando o mesmo para a estrutura *for*. A igualdade de resultados relativos à estrutura *for* pode dever-se à pouca utilização da mesma ao longo do desafio, já que as estruturas *if/if..else* e *while/do..while* foram mais adotadas nos algoritmos apresentados. Este projeto pode não ser adequado à aprendizagem do ciclo de repetição *for*.

As questões de resposta aberta não trazem novas informações significativas relativamente às questões de resposta fechada. Apenas podemos verificar que a maioria dos alunos apreciou as cinco sessões, pelo facto de terem interagido com robôs, sendo que alguns manifestaram também agrado na forma como decorreram as mesmas.

## 7. Reflexão Final

De forma geral o estudo aqui presente, relativamente à intervenção realizada, obteve resultados que revelam alguma evolução dos alunos na aprendizagem da programação, mesmo depois dos mesmos terem frequentado diversas sessões abordando os mesmos conceitos teóricos, com o Professor Cooperante. Os resultados obtidos neste estudo também confirmam as opiniões dos autores Edward et al. (1997), Hloupis et al. (2011) e Benitti (2012) que referem que a utilização de robôs traz mais interesse e facilidade de aprendizagem. Confirmando o que Carbonaro et al. (2010) referem, o questionário geral também revela que a maioria dos alunos se sentiu mais envolvido nas sessões de trabalho com robôs. Isto também pode ter potenciado as suas aprendizagens.

Este estudo também confirma alguns trabalhos realizados por Gomes (2012), Varanda (2012), Costa (2012) e Bernardo (2012), que evidenciam o uso de robôs e as potencialidades dos mesmos no ensino da programação. Contudo também podemos verificar algum paralelismo novamente com o estudo de Benitti (2012) onde refere que os robôs podem trazer algum potencial no ensino da programação, contudo os resultados não são garantidamente favoráveis e nem sempre significativos, como podemos concluir ao observarmos os dados estatísticos relativos às respostas mais corretas dos inquéritos ou mesmo relativamente à estrutura de controlo *for*.

Existe aqui também um fator que pode ter influenciado a fiabilidade dos dados relativamente aos conhecimentos dos alunos, a capacidade de interpretação das questões dos pré e pós-teste. Como podemos verificar no Anexo A – Questionário Online à Turma, cerca de 32% dos alunos afirmam ter algumas dificuldades na compreensão da língua portuguesa.

Não podemos esquecer também que tendo adotado uma amostra não probabilística por conveniência, os resultados aqui alcançados não podem ser extrapolados para outra população. Por outro lado, este estudo carece na quantidade de alunos o que pode aumentar o erro amostral e que, de acordo com Coutinho (2011), uma amostra inferior a trinta tem poucas probabilidades de refletir fielmente uma população. Neste sentido seria interessante aplicar este estudo a uma população mais alargada, como por exemplo a nível nacional.



## 8. Referências Bibliográficas

- Abrantes, P. Matos, J.F. (2012) Using Activity Theory to Understand How Using Robots Can Foster Learning of Basic Concepts of Programming, *Inted2012 Proceedings*, pp. 6483-6490.
- Bernardo, M., P. (2012). A robótica educativa aplicada na consolidação de conhecimentos na disciplina de Linguagens de Programação. Relatório Da Prática De Ensino Supervisionada de Mestrado apresentada à Universidade de Lisboa, Lisboa;
- Bryant, S., Romero, P., & Du Boulay, B. (2008). Pair programming and the mysterious role of the navigator. *International Journal of Human-Computer Studies*, 66(7), 519-529. Academic Press, Inc. Disponível em <http://linkinghub.elsevier.com/retrieve/pii/S1071581907000456>.
- Burguillo, J. C. (2010). Using game theory and Competition-based Learning to stimulate student motivation and performance. *Computers & Education*, 55(2), 566-575. Elsevier Ltd. Disponível em <http://linkinghub.elsevier.com/retrieve/pii/S0360131510000527>
- Carbonaro, M., Szafron, D., Cutumisu, M., & Schaeffer, J. (2010). Computer-game construction: A gender-neutral attractor to Computing Science. *Computers & Education*. Elsevier Ltd, 55, 1098-1111.
- Carroll, J.M. (2000). Five reasons for scenario-based design. Elsevier Science B.V. *Interacting with Computers*, 13, 43-60.
- César, M. (2000). Interações sociais e apreensão de conhecimentos matemáticos: A investigação contextualizada. In J. P. Ponte, & L. Serrazina (Eds.), *Educação matemática em Portugal, Espanha e Itália: Actas da escola de verão em educação matemática, 1999* (pp. 5-46). Lisboa: Sociedade Portuguesa de Ciências da Educação (SPCE) - Secção de Educação Matemática (SEM).
- Costa, J., M., A. (2012). A Robótica Educativa Na Aprendizagem Da Programação. Relatório Da Prática De Ensino Supervisionada de Mestrado apresentada à Universidade de Lisboa, Lisboa;
- Clancey, W. J., (1985), *Conceptual structures — Information processing in mind and machine*: J.F. Sowa, systems programming series. Stanford. CA.
- Coutinho, C. P. (2011). *Metodologia de Investigação em Ciências Sociais e Humana: Teoria e Prática*. Coimbra: Almedina.
- Direção Geral da Formação Vocacional (2005). *Referencial de Formação: Cursos Profissionais De Nível Secundário*. Ministério da Educação.
- Downey, A., B. (2012). *Think Python: How to Think Like a Computer Scientist*. Massachusetts. Green Tea Press. Disponível em <http://www.greenteapress.com/thinkpython/>.
- Escola Secundária Camões [ESC] (2010). Projeto Educativo 2010/2013. Retirado de [http://portal.escamoes.pt/images/docs/escola/PE\\_2010\\_2013.pdf](http://portal.escamoes.pt/images/docs/escola/PE_2010_2013.pdf).
- Escola Secundária Camões [ESC] (2012). *Relatório da Direção 1º Trimestre do Ano Letivo 2011/2012*. Retirado de <http://portal.escamoes.pt/images/docs/rel2012final.pdf>.
- Gaspar, I., & Roldão, M. (2007). *Elementos do Desenvolvimento Curricular*. Lisboa: UA Hammond, L.D.
- Gomes, A., Henriques, J., & Mendes, A. J. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias*, pp. 93-103.

- Gomes, G., & Abrantes, P. (2012). A Robótica Educativa no Ensino da Programação. II Congresso Internacional TIC e Educação. Disponível em <http://ticeduca.ie.ul.pt/atas/pdf/251.pdf>.
- Gomes, G. (2012). A Robótica Educativa No Ensino Da Programação. Relatório Da Prática De Ensino Supervisionada de Mestrado apresentada à Universidade de Lisboa, Lisboa;
- Govender, I. (2006). Learning to Program, Learning to Teach Programming: Pre- and In-service Teachers' Experiences of an Object-oriented Language. University of South Africa.
- Hloupis, G., Stavrakas, I., Moutzouris, K., Alexandridis, A., & Triantis, D. (2011). Procedia Engineering WSN Open Source Development Platform : Application to Green Learning. New York, 00, 0-3.
- Helle, L., Tynjälä, P. & Olkinuora, E. (2006). Project-Based Learning in Post-Secondary Education – Theory, Practice and Rubber Sling Shots. Higher Education. Vol. 51, No. 2 (Mar., 2006), pp. 287-314. Springer.
- Lau, W. W. F., Yuen, A. H. K. (2010). Promoting conceptual change of learning sorting algorithm through the diagnosis of mental models: The effects of gender and learning styles. Computers & Education, 54(1), 275-288.
- Major, L. (2010). Systematic Literature Review Protocol: Teaching Novices Programming Using Robots. Keele University, Staffordshire.
- Pacheco, J. 1996. Currículo: Teoria e Praxis, Porto Editora. Porto.
- Price, B. A., Hirst, A., Johnson, J. Petre, M., & Richards, M. (2002). Using robotics for teaching computing, science, and engineering at a distance. In Proceedings of the 5th IASTED International Conference on Computers and Advanced Technology in Education. Calgary, Canada, IASTED/ACTA Press;
- Saeli, M., Perrenet, J., Jochems, W. M. G., & Zwaneveld, B. (2010). Teaching Programming in Secondary School : A Pedagogical Content Knowledge Perspective. Informatics in Education, 10(1), 73-88. Retirado de <http://www.scopus.com/inward/record.url?eid=2-s2.0-79955374391&partnerID=40&md5=1ae3790ecea07635947faa7afd2a2edb>.
- Santos, E. A., Fermé, E., Fernandes, E. (2005), Utilização de Robots no Ensino de Programação: O Projeto *Droide*. Departamento de Matemática e Engenharias. Universidade da Madeira. Funchal;
- Shulman, L.S. (1986). Those who understand: knowledge growth in teaching. Educational Researcher, 15, 4–14.
- Varanda, F., M., S., P., D. (2012). Robótica Educativa No Ensino De Subprogramas. Relatório Da Prática De Ensino Supervisionada de Mestrado apresentada à Universidade de Lisboa, Lisboa;
- Xinogalos, S., Satratzemi, M., & Dagdilelis, V. (2006). An introduction to object-oriented programming with a didactic microworld: objectKarel. Computers & Education, 47(2), 148-171. Disponível em <http://linkinghub.elsevier.com/retrieve/pii/S0360131504001447>.

## **9. Anexos**

## **Anexo A – Questionário Online à Turma**

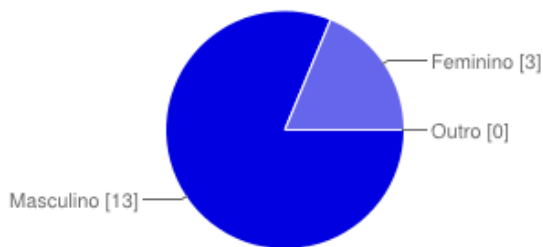


## Questionário Online à Turma

O inquérito realizado em Google Docs apresenta as seguintes questões e respetivos dados sumários. Os textos introduzidos são cópias integrais realizados pelos alunos e podem conter erros ortográficos.

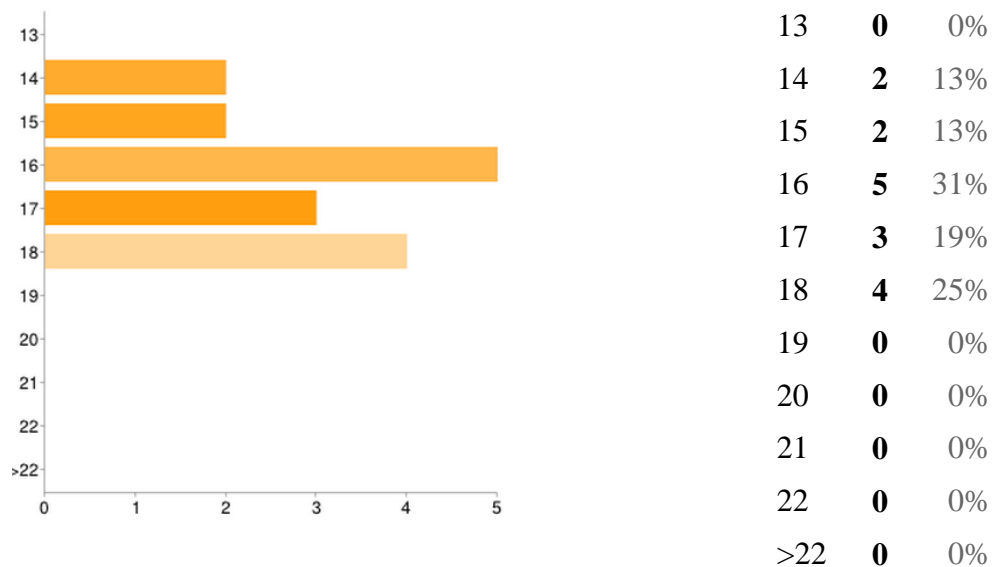
**Nome completo:** estes dados foram removidos para garantir a confidencialidade dos dados;

**Sexo:**



Masculino	<b>13</b>	81%
Feminino	<b>3</b>	19%
Outro	<b>0</b>	0%

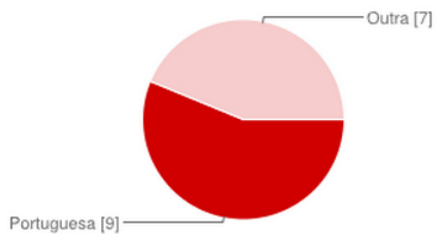
**Idade:**



13	<b>0</b>	0%
14	<b>2</b>	13%
15	<b>2</b>	13%
16	<b>5</b>	31%
17	<b>3</b>	19%
18	<b>4</b>	25%
19	<b>0</b>	0%
20	<b>0</b>	0%
21	<b>0</b>	0%
22	<b>0</b>	0%
>22	<b>0</b>	0%

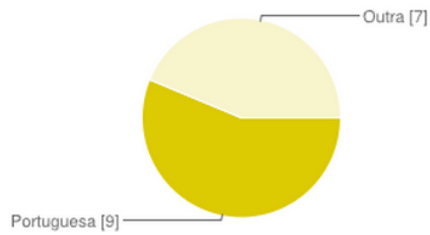
**e-Mail;**

**Qual a sua nacionalidade?**



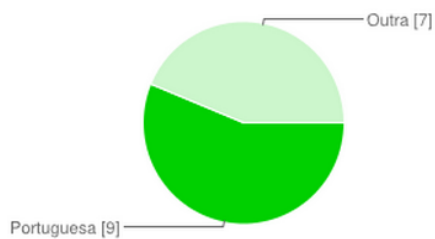
Portuguesa	<b>9</b>	56%
Outra	<b>7</b>	44%

### Qual a nacionalidade do pai?



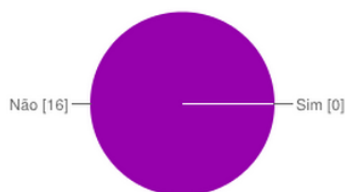
Portuguesa	<b>9</b>	56%
Outra	<b>7</b>	44%

### Qual a nacionalidade da mãe?



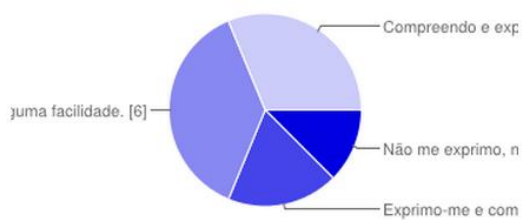
Portuguesa	<b>9</b>	56%
Outra	<b>7</b>	44%

### Para além da escola, tem algum trabalho?



Não	<b>16</b>	100%
Sim	<b>0</b>	0%

### Competências em língua portuguesa:

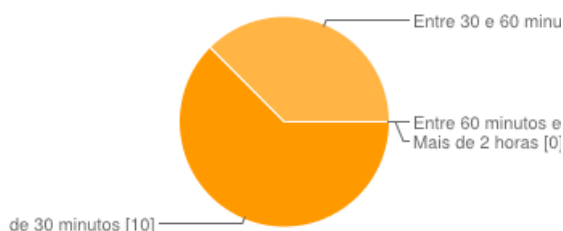


Não me expri, nem compreendo a língua Portuguesa. **2** 13%

Exprimo-me e compreendo a língua **3** 19%

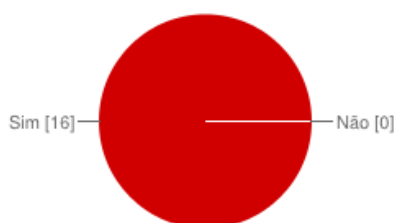
Portuguesa com alguma dificuldade.			
Exprimo-me compreendo a língua Portuguesa com alguma facilidade.	e	<b>6</b>	38%
Compreendo exprimo-me fluentemente em língua Portuguesa.	e	<b>5</b>	31%

**Quanto tempo demora na viagem casa-escola ou escola-casa?**



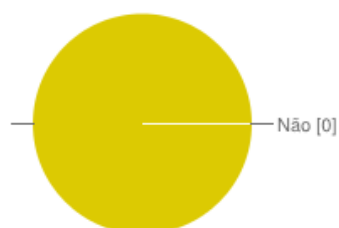
Menos de 30 minutos	<b>10</b>	63%
Entre 30 e 60 minutos	<b>6</b>	38%
Entre 60 minutos e 2 horas	<b>0</b>	0%
Mais de 2 horas	<b>0</b>	0%

**Tem computador em casa?**



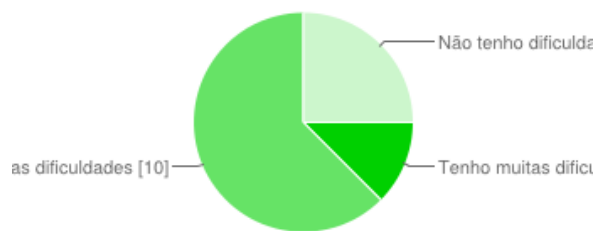
Sim	<b>16</b>	100%
Não	<b>0</b>	0%

**Tem acesso à Internet em casa através do seu computador?**



Sim	<b>16</b>	100%
Não	<b>0</b>	0%

**Costuma ter dificuldades nas disciplinas de informática?**



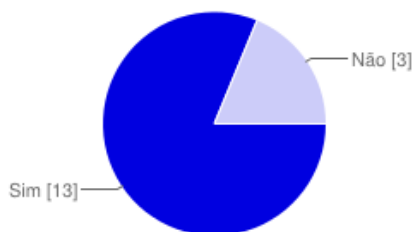
Tenho muitas dificuldades	<b>2</b>	13%
Tenho dificuldades	<b>10</b>	63%
Não tenho dificuldades nenhuma	<b>4</b>	25%

### Gosta da disciplina de Linguagens de Programação?



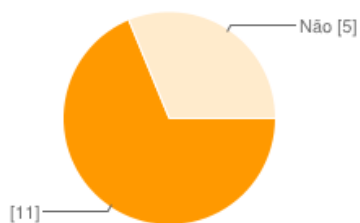
Sim	<b>16</b>	100%
Não	<b>0</b>	0%

### Quer seguir um curso de informática após o 12º ano?



Sim	<b>13</b>	81%
Não	<b>3</b>	19%

### Quer seguir estudos após o 12º ano?



Sim	<b>11</b>	69%
Não	<b>5</b>	31%

### Na sua opinião, para que é que serve a Disciplina de Linguagens de Programação?

1. Eu acho que é uma disciplina importante e que vai ajudar para o meu futuro no mercado de trabalho se estudar bem esta disciplina.

2. Para dar as bases de programação e para aprendermos diversas linguagens de código.
3. Penso que a disciplina serve para nos introduzir, as bases e as diferentes linguagens utilizadas para programar.
4. Para sabermos programar um computador e\ou programas.
5. Na Minha opinião a disciplina De Linguagem De programação serve para programar sites.
6. para aprender-mos a programar.
7. Na minha opinião, é a matéria mais importante neste curso, além de ser a mais divertida e satisfatória. Serve para ajudar-nos não apenas a programar, mas também prepara-nos para o mundo de trabalho.
8. na minha opiniao a disciplina linguagens de progrmação serve para fazer sites.
9. serve para ajudar em alguns trabalhos.
10. A disciplina Linguagens de Programação, serve para nos ensinar um dos componentes mais importantes de informática, a programação, as várias linguagens, a informática está também baseada nisso.
11. Nessa disciplina desenvolve-se uma grande capacidade de programação, é um disciplina fundamental.
12. Para nos aprender-mos a programar.
13. Para o curso actual, é uma disciplina muito fundamental, mesmo para o nosso futuro como informáticos. Através da mesma, podemos aprofundar mais os nossos conhecimentos no que diz respeito á informática, e á programação.
14. Através da mesma, podemos aprofundar mais os nossos conhecimentos no que diz respeito à informática, e à programação.
15. Para aprender programar atraves de varias Linguagens de Programação.
16. Para saber como programar, e para saber como montar coisas, como por exemplo momtar robôs, como montar computadores, como montar telemovéis, como monatr um carro telecomandado, como montar um carro com objectos automaticos(como por exemplo o Star/Stop nos carros eléctricos).

## **Anexo B – Enunciado Projeto Um Robô em Marte**

# Escola Secundária de Camões

Trabalho Prático:

## Um Robô em Marte

De acordo com o cenário apresentado e com o *robô* disponibilizado pelo professor, este enunciado apresenta um desafio a realizar em grupos de 2 (preferencialmente).

### Enquadramento

O NS4 é um robô que foi enviado para o planeta Marte para fazer a recolha de material alienígena. A informação recolhida da análise dos materiais deverá ser transmitida para o planeta Terra.

### Objetivo do projeto

Os alunos deverão programar o robô para que o mesmo percorra o espaço sem colidir com os obstáculos até encontrar o ponto de alto-relevo. Este ponto de alto-relevo será o pequeno monte na maquete. Depois do NS4 encontrar este ponto deverá emitir um sinal sonoro com o seu beeper. Este sinal sonoro determina o fim da missão.

Para conseguir alcançar este desafio o robô NS4 dispõe de:

- 3 Sensores ultrassónicos que permitem detetar a que distância ele está dos objetos que o rodeia;
- 2 Motores que trabalham em ambos os sentidos para o mesmo se conseguir movimentar.
- 1 Beeper para assinalar missão concluída.
- Um conjunto de funções que permitem controlar os diversos sensores, motores e beeper do robô.

### Torneio de Robôs

Para determinar qual o robô mais rápido será realizado um pequeno torneio. Para isso, os robôs poderão ser colocados 2 a 2 na maquete a distâncias equivalentes do ponto de alto-relevo a encontrar. O robô mais rápido ganhará o duelo indo confrontar-se com outro robô.

## API do Robô NS4

Para que os alunos possam manipular o robô são disponibilizadas as seguintes funções:

### Funções dos Sensores

LerSensorFrontal() – Faz a leitura do sensor de distância frontal e devolve os centímetros até ao obstáculo.

LerSensorEsquerdo() – Faz a leitura do sensor de distância esquerdo e devolve os centímetros até ao obstáculo.

LerSensorDireito() – Faz a leitura do sensor de distância direito e devolve os centímetros até ao obstáculo.

### Funções para Movimentação

AndarParaFrente() – Esta função inicia um movimento para a frente do robô.

AndarParaFrente(int miliSegundos) – Esta função faz andar o robô em frente durante um determinado tempo passado na variável *milissegundos*.

AndarParaTras() – Esta função inicia a marcha para traz.

AndarParaTras(int miliSegundos) – Esta função faz andar o robô para traz durante um determinado tempo passado na variável *milissegundos*.

VirarParaEsquerda(int miliSegundos) – Esta função pode ser chamada para fazer o robô virar para a esquerda durante o determinado tempo especificado no parâmetro.

VirarParaDireita(int miliSegundos) – Esta função pode ser chamada para fazer o robô virar para a direita durante o determinado tempo especificado no parâmetro.

Parar() – Esta função poderá ser invocada depois de chamar as funções *AndarParaFrente()* e *AndarParaTras()* imobilizando o robô.



### **Sinalização de Missão Concluída**

Beep(int miliSegundos) – Ao ser chamada, esta função faz o robô emitir um sinal sonoro com uma duração relativa à variável milissegundos introduzida.

## **Anexo C – Linguagens de Programação: Elenco Modular**

### Linguagens de Programação: elenco modular

<b>Número</b>	<b>Designação</b>	<b>Duração de Referência (horas)</b>
1	Algoritmia	24
2	Introdução à Linguagem de Programação	20
3	Estruturas de Controlo	36
4	Subprogramas (Procedimentos e Funções)	36
5	Tipo Estruturado – Tabelas	21
6	Tipo Estruturado – Registos	18
7	Estruturas Dinâmicas (Apontadores)	27
8	Ficheiros	18
9	Projeto	27
10	Introdução à Programação Orientada por Objetos	18
11	Introdução à Linguagem de Programação Orientada por Objetos	24
12	Fundamentos Avançados de Programação Orientada por Objetos	36
13	Gestão de Componentes	36
14	Acesso a Bases de Dados	36
15	Especificação e Documentação	27
16	Instalação e Apoio ao Utilizador	18
17	Desenvolvimento de uma Aplicação de Gestão	36

## **Anexo D – Cenário de Aprendizagem “Uma Experiência em Marte”**

## Uma Experiência em Marte



Adaptado de

[http://sol.sapo.pt/inicio/Internacional/Interior.aspx?content\\_id=56204](http://sol.sapo.pt/inicio/Internacional/Interior.aspx?content_id=56204)

**Inspiração:** A descoberta de ambientes alienígenas pode ser um recurso para a sobrevivência do ser humano.

**Objetivo Geral:** Participar numa equipa de desenvolvimento de algoritmos de deslocamento com base em sensores.

**Atividades:**

- Analisar o comportamento dos sensores ultrassónicos
- Definir estratégias de movimentação
- Implementação de algoritmos de deslocamento 2D
- Discussão e debate de soluções encontradas.

**Tecnologias e Recursos:** Os alunos irão dispor de veículos motorizados de tração integral controlados por microcontroladores *Arduino* acoplados a sensores ultrassónicos. A linguagem de programação será o *Arduino Language*, baseada em [Wiring](#), idêntica ao *C/C++*.

**Resumo da narrativa:**

A NASA está a desenvolver um projeto com estudantes de vários países que consiste em construir um veículo para se deslocar no planeta Marte, sem colidir com os artefactos à sua volta, com objetivo de recolher amostras dos diversos materiais encontrados. Uma estratégia de implementação é dividir os vários problemas pelas equipas de programadores. A Equipa portuguesa fica responsável por construir um algoritmo que permita deslocar o veículo apenas com a ajuda dos sensores ultrassónicos. Este tipo de sensor é muito importante porque apenas deteta materiais sólidos, não se deixando enganar por possíveis gases ou vapores marcianos.

**Palavras-chave:** Programação, Robôs, Ensino-Aprendizagem, Inteligência Artificial

**Espaços:** Os alunos poderão trabalhar no projeto dentro da sala de aula com maquetes reproduzindo um local alienígena, ou simplesmente na rua em locais de terra batida com os obstáculos naturais em redor.

**Papéis:** O professor deverá fazer o papel de orientador, guiando os diversos grupos de alunos nas suas soluções originais.

**Interações:** Os alunos poderão partilhar ideias discutindo as potencialidades e constrangimentos de modo a criar sinergias e encontrando também soluções mais eficientes. É importante o professor fomentar um espírito colaborativo e de partilha (sem plágio) entre todos levantando possíveis desafios nos contextos de cada grupo de trabalho valorizando as soluções originais e inovadoras.

## **Anexo E – O Robô *Arduino* Montagem do Equipamento**

# O Robô *Arduino*

(Montagem do Equipamento)

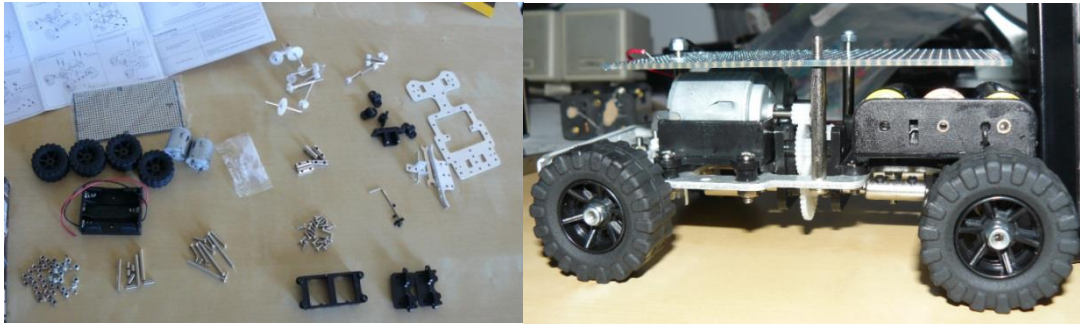
## 1. Robô *Arduino*

Os robôs utilizados nesta intervenção terão a forma de um pequeno veículo de tração integral movido com dois motores elétricos. Para a sua montagem foram necessárias as seguintes componentes:

- Chassis com dois motores de translação: automóvel e suporte a todo o equipamento;
- *Arduino*: cérebro do nosso robô;
- Controlador de motores: para permitir alimentar os motores, bem como direccionar os mesmos com rotação nos dois sentidos permitindo a marcha para a frente e retaguarda;
- Sensor ultrassónico: sensor que permite detetar obstáculos à sua frente dando assim uma certa capacidade de “visão” ao robô para não colidir com os prédios do nosso cenário.
- *Beeper*: pequeno equipamento capaz de emitir um ruído sonoro. Poderá, por exemplo, ser utilizado como output para *debugging* ou sinalizador de eventos.

## 2. Montagem do Chassis

São apresentadas algumas imagens das componentes de montagem do chassis:

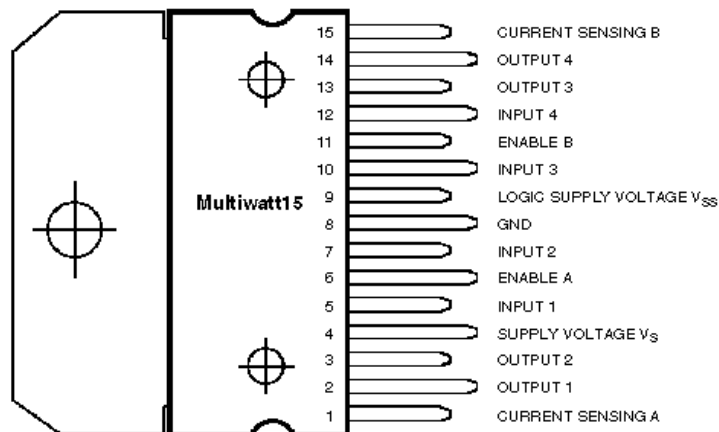


Componentes do chassis

Chassis

## 3. Controlador de Motores

O controlador de motores utiliza um chip L298N que para além de conseguir alimentar motores, tal como um transístor, permite inverter a polaridade das saídas de alimentação invertendo o sentido de rotação dos motores. Desta forma foi possível implementar a marcha atrás do robô. É apresentado abaixo o *Pinout* do circuito:



L298N - Pinout

Legenda:

- CURRENT SENSING A/B: pinos não utilizados;
- OUPUT 3 e 4 / 1 e 2: alimentação dos motores;



- ENABLE A e B: pino lógico para ativação dos motores;
- INPUT 1, 2, 3, 4: controlo de movimentação de motores;
- SUPPLY VOLTAGE  $V_s$  (0v~46v): alimentação principal de motores;
- LOGIC SUPPLY VOLTAGE  $V_{ss}$  (4.5v~7v): alimentação de controlos lógicos;
- GND: massa.

Admitindo que as alimentações do circuito estão ligadas a 5v, a movimentação de um dos motores, ligado aos Outputs 1 e 2, pode ser representada segundo a seguinte tabela de verdade:

ENABLE A	INPUT 1	INPUT 2	OUTPUT 1	OUTPUT 2	Rotação
5v	0v	0v	0v	0v	Nula
5v	5v	0v	5v	0v	<i>Clockwise</i>
5v	0v	5v	0v	5v	<i>Anticlockwise</i>
5v	5v	5v	5v	5v	Nula
0v	0v	0v	0v	0v	Nula
0v	5v	0v	0v	0v	Nula
0v	0v	5v	0v	0v	Nula
0v	5v	5v	0v	0v	Nula

Para ser possível ter controlo total na manipulação dos dois motores, nos dois sentidos, foram utilizadas as portas digitais 10, 11 e 13 do Arduino:

- Motor esquerdo: pino 13;
- Motor direito: pino 11;
- Motores direito e esquerdo em sentido inverso: pino 10.

O que nos permite efetuar os seguintes movimentos:

- Andar para a frente: pinos 11 e 13 ativos;
- Virar para a esquerda: pino 11 (motor direito);
- Virar para a direita: pino 13 (motor esquerdo);
- Andar para trás: pino 10 (os dois motores em sentido inverso).

#### 4. Sensor Ultrassónico ou Sonar

Este sensor permite medir distâncias entre 2 cm e 500cm (5m) entre ele próprio e um objeto à sua frente, tendo uma precisão de 3mm e a sua utilização com o compilador tradicional do Arduino é facilitada.

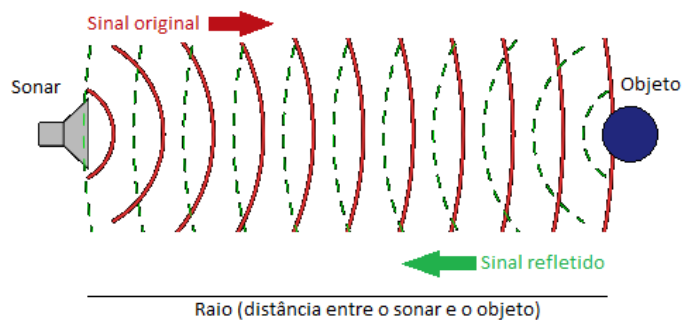


Algoritmo de funcionamento (versão simplificada):

1. O sonar (sensor ultrassónico) emite um sinal sonoro numa determinada direção.
2. Depois de emitir o sinal, o sonar aguarda escutando o seu próprio eco.
3. Ao atingir um objeto, o sinal sonoro é refletido em direção contrária voltando a colidir no próprio sonar.
4. Ao receber e detetar o seu eco, sabendo quanto tempo (T) decorreu entre a emissão e a receção do sinal, sabendo que a velocidade da propagação do som (VS) na atmosfera corresponde a cerca de 340,9ms (este valor pode variar derivado às condições atmosféricas), pode aplicar-se a seguinte fórmula para calcular a distância a que o sensor está do objeto à sua frente:

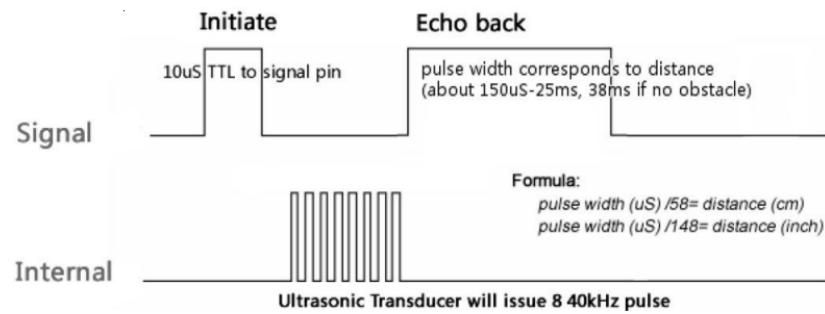
$$\text{Raio} = T \times VS / 2$$

O algoritmo pode também ser traduzido na seguinte figura:



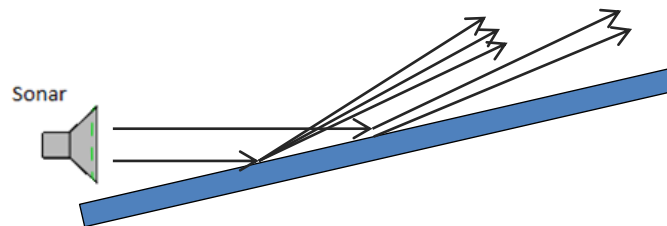
NOTA: Os alunos não terão de se preocupar com este tipo de cálculo porque o equipamento instalado no robô devolve já a distância medida em centímetros.

A implementação deste tipo de sensores requer no entanto mais algumas especificações técnicas. Neste caso, o funcionamento do HC-SR04, sensor ultrassónico por impulsos, pode também ser traduzido na seguinte figura:



Ciclo de atividade do sensor – (arduino.cc)

Após uma primeira montagem e alguns testes realizados verificou-se que um único sensor ultrassónico (fixo) é insuficiente para mover o robô em segurança sem colidir com objetos que se encontram ao seu lado. Em casos menos ideais o sonar pode ter alguma dificuldade em detetar objetos em ângulos oblíquos relativamente ao sentido de emissão do sinal sonoro. Vejamos a seguinte figura que apresenta as reflexões do sinal emitido:



Reflexões em superfícies com ângulos obtusos

Quando estamos perante um ângulo obtuso, o sinal sonoro não é refletido novamente para o emissor. Neste caso o sonar não consegue identificar nada à sua frente. Daí a necessidade de adicionar sensores laterais.

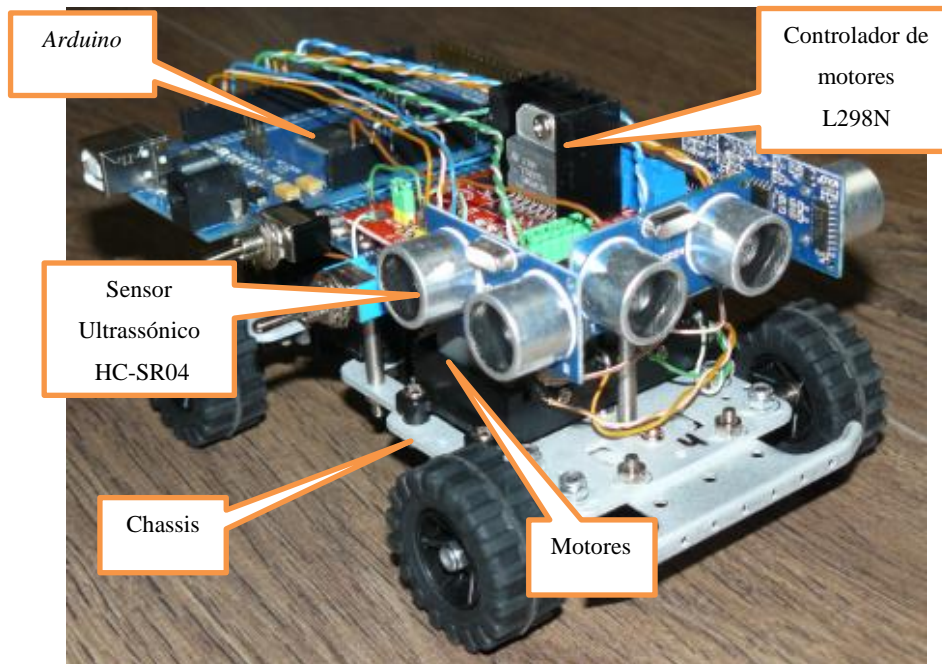
### Interface do sensor HC-SR04

O sensor HC-SR04 tem o seguinte pinout ou interface:

- VCC: alimentação 5v;
- Trigger: entrada de impulso;
- Eco: receção de impulso;
- GND: massa 0v.

## 5. Robô finalizado

A imagem seguinte ilustra o aspecto final do robô:



Robô



## **Anexo F – Programação em *Arduino* vs. C/C++**

# Programação em Arduino vs. C/C++

	Arduino	C/C++
<b>Acesso condicional simples - <i>if</i></b>	<code>if (...) { ... }</code>	<code>if (...) { ... }</code>
<b>Acesso condicional composto – <i>if ... else</i></b>	<code>if (...) { ... } else { ... }</code>	<code>if (...) { ... } else { ... }</code>
<b>Ciclo de repetição - <i>for</i></b>	<code>for (inicialização; condição de paragem; incremento) { ... }</code>	<code>for (inicialização; condição de paragem; incremento) { ... }</code>
<b>Escolha múltipla - <i>switch</i></b>	<code>switch (...) {   case opção1:     ...     break;   case opção2:     ...     break;   default:     ... }</code>	<code>switch (...) {   case opção1:     ...     break;   case opção2:     ...     break;   default:     ... }</code>
<b>Ciclo de repetição com condição de paragem no início - <i>while</i></b>	<code>while (...) { ... }</code>	<code>while (...) { ... }</code>
<b>Ciclo de repetição com condição de paragem no fim (o código é executado pelo menos uma vez) – <i>do ... while</i></b>	<code>do { ... } while (...);</code>	<code>do { ... } while (...);</code>
<b>Paragem de ciclo - <i>break</i></b>	<code><i>break</i>;</code>	<code><i>break</i>;</code>
<b>Reinício de ciclo – <i>continue</i></b>	<code><i>Continue</i>;</code>	<code><i>Continue</i>;</code>
<b>Retorno ou término do procedimento em execução - <i>return</i></b>	<code><i>return</i>; <i>return variável</i>;</code>	<code><i>return</i>; <i>return variável</i>;</code>
<b>Salto incondicional - <i>goto</i></b>	<code>goto pontoA; ... pontoA:</code>	<code>goto pontoA; ... pontoA:</code>

## **Anexo G – Project Based Learning**



## **Project Based Learning**

Na tentativa de caracterizar o PjBL genericamente, Helle, et all (2006), com base numa revisão sistemática da literatura que envolveu artigos entre os anos 1960 a 2001, identificou um conjunto de características referentes à metodologia as quais são de seguida descritas:

- i) Os problemas a resolver podem ser colocados pelos alunos ou outra entidade (e.g. professor);
- ii) O conjunto variado de atividades educativas implica a iniciativa de alunos ou grupos de alunos;
- iii) A metodologia frequentemente utilizada resulta num produto final que pode ser por exemplo uma tese, um relatório, planos de design, um programa de computador, um modelo ou objeto físico;
- iv) O trabalho decorre, frequentemente, por um período contínuo de trabalho;
- v) Em vez de ter um papel autoritário e de controlo, o corpo docente está envolvido por forma a aconselhar e orientar os alunos através das etapas do projeto.

Ainda segundo Helle, et all (2006), o PjBL pode ser implementado sob diferentes formas, consoante orientações pedagógicas, éticas ou políticas:

- i) Exercício de Projeto: Esta forma de projeto implica que os alunos apliquem conhecimentos previamente adquiridos sobre temas ou técnicas previamente conhecidas para chegar a um objetivo final. Este tipo de projeto é o mais comum e pressupõe alguma aprendizagem ao longo de algumas aulas independentes do exercício de projeto.
- ii) Componente de Projeto: Esta forma de projeto envolve geralmente objetivos mais abrangentes fomentando também possíveis interdisciplinaridades. A mesma pode também estar relacionada com aspetos do mundo real.
- iii) Orientação de Projeto: Este tipo de projeto é o mais abrangente dos três. Envolve a globalidade do currículo ou programa de estudos. Neste caso os alunos são envolvidos desde o início do curso que frequentam e todas as disciplinas são enquadradas com um determinado objetivo no projeto.

## **Anexo H – Etapas do Projeto**

## **Etapas do Projeto**

Caso o projeto dado aos alunos revele ser de dificuldade acrescida para os alunos com mais dificuldade, estes últimos serão aconselhados a desenvolver o projeto seguindo as seguintes etapas:

0. Andar para a frente 2 segundos e parar;
1. Andar para a frente durante 1 segundo, fazer uma pausa de 0.5 segundos e voltar para trás regressando à posição inicial;
2. Andar para a frente até encontrar um obstáculo, contornar o obstáculo e continuar a andar para a frente;
3. Andar em frente. Sempre que é identificado um obstáculo, o robô tem de mudar de direção.
4. Andar para a frente, se encontrar um obstáculo mudar de direção voltando ao início do algoritmo, se encontrar um ponto de alto-relevo emitir um sinal sonoro.

## **Anexo I – API (*Application Programming Interface*) do Robô**

## API (Application Programming Interface) do Robô

### Funções dos Sensores

Ler Sensor Frontal	<b>Especificação:</b> int LerSensorFrontal(void) <b>Descrição:</b> Faz a leitura do sensor de distância frontal e devolve o valor até ao obstáculo em centímetros.
Ler Sensor Esquerdo	<b>Especificação:</b> int LerSensorEsquerdo (void) <b>Descrição:</b> Faz a leitura do sensor de distância esquerdo e devolve o valor até ao obstáculo em centímetros.
Ler Sensor Direito	<b>Especificação:</b> int LerSensorDireito (void) <b>Descrição:</b> Faz a leitura do sensor de distância do lado direito e devolve o valor até ao obstáculo em centímetros.

### Funções de Movimentação

Andar Para Frente	<b>Especificação:</b> void AndarParaFrente (void) void AndarParaFrente(int tempo) <b>Parâmetros:</b> tempo: duração em milissegundos do tempo de marcha. <b>Descrição:</b> Esta função inicia um movimento para a
-------------------	---

	frente do robô. Caso seja especificado o parâmetro tempo, o robô deverá andar para a frente até que o tempo decorra.
Andar Para Trás	<p><b>Especificação:</b> void AndarParaTras (void) void AndarParaTras(int tempo)</p> <p><b>Parâmetros:</b> tempo: duração em milissegundos do tempo de marcha.</p> <p><b>Descrição:</b> Esta função inicia um movimento para trás do robô. Caso seja especificado o parâmetro tempo, o robô deverá andar para trás até que o tempo decorra.</p>
Virar Para Esquerda	<p><b>Especificação:</b> void VirarParaEsquerda(int tempo)</p> <p><b>Parâmetros:</b> tempo: duração em milissegundos do tempo de marcha.</p> <p><b>Descrição:</b> Esta função pode ser chamada para fazer o robô virar para a esquerda durante o determinado tempo especificado no parâmetro em milissegundos.</p>
Virar Para Direita	<p><b>Especificação:</b> void VirarParaDireita(int tempo)</p> <p><b>Parâmetros:</b> tempo: duração em milissegundos do tempo de marcha.</p>

	<p><b>Descrição:</b></p> <p>Esta função pode ser chamada para fazer o robô virar para a direita durante um determinado tempo especificado no parâmetro em milissegundos.</p>
Parar	<p><b>Especificação:</b></p> <p>void Parar()</p> <p><b>Descrição:</b></p> <p>Esta função poderá ser invocada depois de chamar as funções <i>AndarParaFrente()</i> e <i>AndarParaTraz()</i> imobilizando de seguida o robô.</p>

### Sinal Sonoro

Beep	<p><b>Especificação:</b></p> <p>void Parar()</p> <p><b>Parâmetros:</b></p> <p>tempo: duração do sinal sonoro em milissegundos.</p> <p><b>Descrição:</b></p> <p>Ao ser chamada, esta função faz o robô emitir um sinal sonoro com uma duração relativa à variável introduzida.</p>
------	---

## **Anexo J – Planeamento das Sessões de Trabalho**



## Planeamento das Sessões de Trabalho

### Sessão 1.

Nesta primeira sessão será apresentado aos alunos o cenário de aprendizagem “Uma experiência em Marte” onde o mesmo será discutido detalhadamente na tentativa de envolver os alunos nas atividades que se seguem. Com base no Cenário de Aprendizagem será introduzido um enunciado do trabalho a implementar, os critérios de avaliação e a metodologia de trabalho baseada na competição de equipas. Os alunos serão confrontados pela primeira vez com os robôs sendo que será feita uma primeira explicação da API disponibilizada. Se se manifestar necessário, o professor poderá exemplificar como programar o robô para o mesmo fazer uma operação básica. De seguida os alunos serão guiados para resolverem os desafios levantados pelo trabalho proposto.

### Recursos necessários.

- Cenário de Aprendizagem;
- Enunciado do Trabalho;
- Critérios de Avaliação;
- Computadores com Microsoft Windows XP ou superior, driver e compilador *Arduino*;
- Moodle;
- Robôs *Arduino*;
- API de controlo do robô;
- Cenário físico.

### Objetivos de preparação e enquadramento na tecnologia.

- Conhecer e compreender a forma de programar o *Arduino*;
- Conhecer e entender a API de controlo do robô;
- Conhecer e entender os *inputs* e *outputs* do robô: motores sensores e *beeper*.

### Objetivos específicos.

- Movimentar o robô nas várias direções possíveis;

- Fazer a leitura dos sensores ultrassónicos e compreender o significado dos valores recebidos;

### **Conteúdos.**

- Estruturas de controlo de acesso condicional *if* e *if..else*;

### **Atividades.**

- Introdução do trabalho a implementar;
- Programação básica do robô;
- Discussão de ideias a implementar;
- Reflexão e preenchimento do diário de bordo.

### **Sessão 2.**

Após ter um primeiro contacto com o robô os alunos deverão tentar implementar algoritmos para movimentar o robô sem o deixar colidir com outros obstáculos.

### **Recursos necessários.**

- Computadores com Microsoft Windows XP ou superior, driver e compilador *Arduino*;
- Robôs *Arduino*;
- API de controlo do robô;
- Cenário físico;
- Moodle.

### **Objetivos específicos.**

- Detetar obstáculos;
- Movimentar o robô sem colidir com obstáculos.

### **Conteúdos.**

- Estruturas de controlo de acesso condicional *if* e *if..else*;
- Estruturas de repetição *for*, *while* e *do..while*.

### **Atividades.**

- Criação de algoritmos e programação do robô;
- Discussão de ideias a implementar;
- Reflexão e preenchimento do diário de bordo.

### **Sessão 3.**

Esta sessão está orientada para os alunos formalizarem um algoritmo para o robô sair do labirinto onde se encontra inicialmente. O contorno de obstáculos será um ponto a focar nos seus algoritmos.

Os alunos que apresentarem mais dificuldades serão conduzidos a realizar o desafio por etapas, para assim conseguirem ultrapassar um problema de cada vez.

### **Recursos necessários.**

- Computadores com Microsoft Windows XP ou superior, driver e compilador *Arduino*;
- Robôs *Arduino*;
- API de controlo do robô;
- Cenário físico;
- Moodle.

### **Objetivos específicos.**

- Detetar obstáculos;
- Movimentar o robô sem colidir com obstáculos;
- Detetar alto-relevo.

### **Conteúdos.**

- Estruturas de controlo de acesso condicional *if* e *if..else*;
- Estruturas de repetição *for*, *while* e *do..while*.

### **Atividades.**

- Criação de algoritmos e programação do robô;
- Discussão e concetualização de algoritmos;
- Competição de robôs;
- Reflexão e preenchimento do diário de bordo.

#### **Sessão 4.**

Nesta sessão prevê-se que os alunos com menos dificuldades na programação já tenham um algoritmo completo e funcional. Estes alunos serão orientados no sentido de aplicarem otimizações no mesmo, de modo a tirarem o melhor partido possível das estruturas de controlo.

Os alunos que apresentem mais dificuldades continuarão a ser conduzidos a realizar o desafio por etapas até se mostrarem mais autónomos.

A realização do torneio deverá prosseguir entre todos os grupos.

#### **Recursos necessários.**

- Computadores com Microsoft Windows XP ou superior, driver e compilador *Arduino*;
- Robôs *Arduino*;
- API de controlo do robô;
- Cenário físico;
- Moodle.

#### **Objetivos específicos.**

- Detetar obstáculos;
- Movimentar o robô sem colidir com obstáculos;
- Detetar alto-relevo e emitir aviso sonoro;
- Otimizar as soluções até agora criadas.
- Conciliar os conhecimentos até agora adquiridos.

#### **Conteúdos.**

- Estruturas de controlo de acesso condicional *if* e *if..else*;
- Estruturas de repetição *for*, *while* e *do..while*.

#### **Atividades.**

- Otimização de algoritmos e programação do robô;

- Discussão e melhoramento de algoritmos;
- Competição de robôs;
- Reflexão e preenchimento do diário de bordo.

### **Sessão 5.**

Na última sessão poderão ser realizados os últimos ajustes nos algoritmos para finalmente realizar-se uma última competição com todos os grupos. Após esta atividade realizar-se-á uma discussão em plenário onde os alunos poderão apresentar as suas críticas e opiniões acerca da intervenção, realizações pessoais conseguidas ou não, entre outros aspetos que queiram realçar. O professor deverá também fazer uma apreciação global dos trabalhos e apontar os pontos fortes de cada grupo.

A auto e heteroavaliação a realizar tentará promover a reflexão e discussão dos alunos. Esta atividade será realizada através de uma discussão com os alunos para evitar o possível preenchimento aleatório ou sem reflexão dos mesmos. A respetiva ficha pode ser consultada no Anexo G que acompanha este documento.

#### **3.8.5.1. Recursos necessários.**

- Computadores com Microsoft Windows XP ou superior, driver e compilador *Arduino*;
- Robôs *Arduino*;
- API de controlo do robô;
- Cenário físico;
- Moodle;
- Ficha de auto e heteroavaliação.

#### **Objetivos específicos.**

- Movimentar o robô sem colidir com obstáculos;
- Detetar alto-relevo e emitir aviso sonoro;
- Otimizar as soluções até agora criadas.

#### **Conteúdos.**

- Estruturas de controlo de acesso condicional *if* e *if..else*;
- Estruturas de repetição *for*, *while* e *do..while*.

**Atividades.**

- Otimização de algoritmos e programação do robô;
- Discussão e melhoramento de algoritmos;
- Competição de robôs;
- Reflexão e preenchimento do diário de bordo.



## **Anexo K – Grelha de Observação Direta**



## Grelha de Observação Direta

<b>Grupo 1</b>					
<b>Observação</b>		<b>Nº de aluno</b>			
<b>Elementos</b>	<b>Cotação por elemento</b>	<b>A14</b>	<b>A7</b>	<b>A9</b>	<b>A4</b>
<b>1. Análise</b>					
1.1 Perceção da ideia central do projeto bem como os diversos objetivos a atingir	3	3	3	3	3
1.2 Distinção das várias etapas do projeto	3	3	3	3	3
1.3. Compreensão do método de avaliação do trabalho	3	3	3	3	3
<b>2. Desenvolvimento do Projeto</b>					
2.1. Capacidade para analisar criticamente o seu trabalho e resolver problemas que nele surjam, reformulando-o se necessário.	5	5	5	4	5
2.2. Correção ou reformulação do trabalho em função de recomendações feitas pelos acompanhantes (colegas de grupo/professor).	5	4,5	5	4	5
2.3. Capacidade de concetualizar e avaliar diversas hipóteses possíveis para atingir os diversos objetivos	5	4	4,5	3,5	3,5
<b>3. Capacidade Técnica e Científica</b>					
3.1. Seleção das operações e estruturas de controlo adequadas para o projeto	10	10	10	9	9
3.2. Capacidade de investigação nos diversos contextos abordados	5	4,5	4,5	4	4
3.3. Recolha de soluções fundamentadas na sua capacidade técnica e crítica	6	5	6	5	5
3.4. Realização de testes de robustez nas soluções desenvolvidas	5	5	5	4	5
3.5. Otimização e qualidade das soluções apresentadas	4	3	3,5	3	3
3.6. Autonomia	5	4	4	3,5	4
<b>4. Planificação</b>					
4.1. Planificação e orientação do trabalho	4	4	4	3	4
4.2. Cumprimento dos prazos definidos para as tarefas	4	4	4	4	4
<b>5. Auto e Heteroavaliação</b>					
5.1. Avalia corretamente o desenvolvimento das soluções	5	4,5	4,5	3,5	4
5.2. Avalia corretamente a sua participação nas soluções	5	5	5	5	5
5.3. Avalia corretamente a participação dos restantes elementos no projeto	5	5	5	5	5

<b>6. Relacionamento Interpessoal</b>					
6.1. Relacionamento com os elementos do grupo	4	3	4	4	4
6.2. Relacionamento com o professor	5	5	5	5	5
6.3. Aceitação de críticas construtivas	5	5	5	5	5
6.4. Trabalho colaborativo com os elementos do grupo	4	4	4	4	4
Total (0..100)	100	93,5	97	87,5	92,5
<b>Total (0..20)</b>	<b>20</b>	<b>18,7</b>	<b>19,4</b>	<b>17,5</b>	<b>18,5</b>

<b>Grupo 2</b>				
<b>Observação</b>		<b>Nº de aluno</b>		
<b>Elementos</b>	<b>Cotação por elemento</b>	<b>A1</b>	<b>A8</b>	<b>A10</b>
<b>1. Análise</b>				
1.1 Perceção da ideia central do projeto bem como os diversos objetivos a atingir	3	3	2	2
1.2 Distinção das várias etapas do projeto	3	3	3	3
1.3. Compreensão do método de avaliação do trabalho	3	3	3	3
<b>2. Desenvolvimento do Projeto</b>				
2.1. Capacidade para analisar criticamente o seu trabalho e resolver problemas que nele surjam, reformulando-o se necessário.	5	5	3	3
2.2. Correção ou reformulação do trabalho em função de recomendações feitas pelos acompanhantes (colegas de grupo/professor).	5	5	4	4
2.3. Capacidade de conceitualizar e avaliar diversas hipóteses possíveis para atingir os diversos objetivos	5	5	3	3
<b>3. Capacidade Técnica e Científica</b>				
3.1. Seleção das operações e estruturas de controlo adequadas e para o projeto	10	10	5	5
3.2. Capacidade de investigação nos diversos contextos abordados	5	4	3	3
3.3. Recolha de soluções fundamentadas na sua capacidade técnica e crítica	6	6	2	2
3.4. Realização de testes de robustez nas soluções desenvolvidas	5	5	2	2
3.5. Otimização e qualidade das soluções apresentadas	4	3	2	2
3.6. Autonomia	5	5	2	2

<b>4. Planificação</b>				
4.1. Planificação e orientação do trabalho	4	4	2	2
4.2. Cumprimento dos prazos definidos para as tarefas	4	4	4	4
<b>5. Auto e Heteroavaliação</b>				
5.1. Avalia corretamente o desenvolvimento das soluções	5	5	5	5
5.2. Avalia corretamente a sua participação nas soluções	5	5	4	4
5.3. Avalia corretamente a participação dos restantes elementos no projeto	5	5	4	4
<b>6. Relacionamento Interpessoal</b>				
6.1. Relacionamento com os elementos do grupo	4	3	4	4
6.2. Relacionamento com o professor	5	4	5	5
6.3. Aceitação de críticas construtivas	5	5	5	5
6.4. Trabalho colaborativo com os elementos do grupo	4	4	4	4
Total (0..100)	100	96	71	71
<b>Total (0..20)</b>	<b>20</b>	<b>19,2</b>	<b>14,2</b>	<b>14,2</b>

<b>Grupo 3</b>				
<b>Observação</b>		<b>Nº de aluno</b>		
<b>Elementos</b>	<b>Cotação por elemento</b>	<b>A13</b>	<b>A11</b>	<b>A15</b>
<b>1. Análise</b>				
1.1 Perceção da ideia central do projeto bem como os diversos objetivos a atingir	3	3	3	3
1.2 Distinção das várias etapas do projeto	3	3	3	3
1.3. Compreensão do método de avaliação do trabalho	3	3	3	3
<b>2. Desenvolvimento do Projeto</b>				
2.1. Capacidade para analisar criticamente o seu trabalho e resolver problemas que nele surjam, reformulando-o se necessário.	5	4	3	3,5
2.2. Correção ou reformulação do trabalho em função de recomendações feitas pelos acompanhantes (colegas de grupo/professor).	5	3,5	3	3,5
2.3. Capacidade de concetualizar e avaliar diversas hipóteses possíveis para atingir os diversos objetivos	5	3	3	3

<b>3. Capacidade Técnica e Científica</b>				
3.1. Seleção das operações e estruturas de controlo adequadas para o projeto	10	8	7	7
3.2. Capacidade de investigação nos diversos contextos abordados	5	3,5	3	3
3.3. Recolha de soluções fundamentadas na sua capacidade técnica e crítica	6	4	4	4
3.4. Realização de testes de robustez nas soluções desenvolvidas	5	4	3	3,5
3.5. Otimização e qualidade das soluções apresentadas	4	3	2,5	2,5
3.6. Autonomia	5	4	3	3
<b>4. Planificação</b>				
4.1. Planificação e orientação do trabalho	4	3	2	2
4.2. Cumprimento dos prazos definidos para as tarefas	4	4	4	4
<b>5. Auto e Heteroavaliação</b>				
5.1. Avalia corretamente o desenvolvimento das soluções	5	3	3	3
5.2. Avalia corretamente a sua participação nas soluções	5	4	3,5	4
5.3. Avalia corretamente a participação dos restantes elementos no projeto	5	4	4	4
<b>6. Relacionamento Interpessoal</b>				
6.1. Relacionamento com os elementos do grupo	4	4	4	4
6.2. Relacionamento com o professor	5	5	4	5
6.3. Aceitação de críticas construtivas	5	5	4	5
6.4. Trabalho colaborativo com os elementos do grupo	4	4	4	4
Total (0..100)	100	82	73	77
<b>Total (0..20)</b>	<b>20</b>	<b>16,4</b>	<b>14,6</b>	<b>15,4</b>

<b>Grupo 4</b>				
<b>Observação</b>		<b>Nº de aluno</b>		
<b>Elementos</b>	<b>Cotação por elemento</b>	<b>A5</b>	<b>A2</b>	<b>A12</b>
<b>1. Análise</b>				
1.1 Perceção da ideia central do projeto bem como os diversos objetivos a atingir	3	3	3	3
1.2 Distinção das várias etapas do projeto	3	3	2	3
1.3. Compreensão do método de avaliação do trabalho	3	3	3	3

<b>2. Desenvolvimento do Projeto</b>				
2.1. Capacidade para analisar criticamente o seu trabalho e resolver problemas que nele surjam, reformulando-o se necessário.	5	4	2	3
2.2. Correção ou reformulação do trabalho em função de recomendações feitas pelos acompanhantes (colegas de grupo/professor).	5	4	3	3
2.3. Capacidade de concetualizar e avaliar diversas hipóteses possíveis para atingir os diversos objetivos	5	4	2	3
<b>3. Capacidade Técnica e Científica</b>				
3.1. Seleção das operações e estruturas de controlo adequadas para o projeto	10	7	5	6
3.2. Capacidade de investigação nos diversos contextos abordados	5	4	2	3
3.3. Recolha de soluções fundamentadas na sua capacidade técnica e crítica	6	5	2	3
3.4. Realização de testes de robustez nas soluções desenvolvidas	5	4	3	3
3.5. Otimização e qualidade das soluções apresentadas	4	3	2	2
3.6. Autonomia	5	4	3	3
<b>4. Planificação</b>				
4.1. Planificação e orientação do trabalho	4	3	2	2
4.2. Cumprimento dos prazos definidos para as tarefas	4	4	4	4
<b>5. Auto e Heteroavaliação</b>				
5.1. Avalia corretamente o desenvolvimento das soluções	5	4	3	3
5.2. Avalia corretamente a sua participação nas soluções	5	4	4	4
5.3. Avalia corretamente a participação dos restantes elementos no projeto	5	5	4	4
<b>6. Relacionamento Interpessoal</b>				
6.1. Relacionamento com os elementos do grupo	4	4	4	4
6.2. Relacionamento com o professor	5	5	4	5
6.3. Aceitação de críticas construtivas	5	5	5	5
6.4. Trabalho colaborativo com os elementos do grupo	4	4	3	4
Total (0..100)	100	86	65	73
<b>Total (0..20)</b>	<b>20</b>	<b>17,2</b>	<b>13</b>	<b>14,6</b>

<b>Grupo 5</b>			
<b>Observação</b>	Cotação por elemento	<b>Nº de aluno</b>	
		<b>A3</b>	<b>A6</b>
<b>Elementos</b>			
<b>1. Análise</b>			
1.1 Perceção da ideia central do projeto bem como os diversos objetivos a atingir	3	2	2
1.2 Distinção das várias etapas do projeto	3	2	2
1.3. Compreensão do método de avaliação do trabalho	3	2	1
<b>2. Desenvolvimento do Projeto</b>			
2.1. Capacidade para analisar criticamente o seu trabalho e resolver problemas que nele surjam, reformulando-o se necessário.	5	3	2
2.2. Correção ou reformulação do trabalho em função de recomendações feitas pelos acompanhantes (colegas de grupo/professor).	5	3	1
2.3. Capacidade de concetualizar e avaliar diversas hipóteses possíveis para atingir os diversos objetivos	5	3	1
<b>3. Capacidade Técnica e Científica</b>			
3.1. Seleção das operações e estruturas de controlo adequadas e para o projeto	10	5	4
3.2. Capacidade de investigação nos diversos contextos abordados	5	2	2
3.3. Recolha de soluções fundamentadas na sua capacidade técnica e crítica	6	3	1
3.4. Realização de testes de robustez nas soluções desenvolvidas	5	3	1
3.5. Otimização e qualidade das soluções apresentadas	4	1	1
3.6. Autonomia	5	2	1
<b>4. Planificação</b>			
4.1. Planificação e orientação do trabalho	4	1	1
4.2. Cumprimento dos prazos definidos para as tarefas	4	2	2
<b>5. Auto e Heteroavaliação</b>			
5.1. Avalia corretamente o desenvolvimento das soluções	5	3	2
5.2. Avalia corretamente a sua participação nas soluções	5	3	3
5.3. Avalia corretamente a participação dos restantes elementos no projeto	5	3	3

6. Relacionamento Interpessoal			
6.1. Relacionamento com os elementos do grupo	4	3	3
6.2. Relacionamento com o professor	5	4	3
6.3. Aceitação de críticas construtivas	5	4	4
6.4. Trabalho colaborativo com os elementos do grupo	4	3	2
Total (0..100)	100	57	42
<b>Total (0..20)</b>	<b>20</b>	<b>11,4</b>	<b>8,4</b>

## **Anexo L – Auto e Heteroavaliação**



## Auto e Heteroavaliação

		Grupo 1																				
		A14				A7				A9				A4								
		Nunca	Poucas vezes	Algumas vezes	Muitas vezes	Sempre	Nunca	Poucas vezes	Algumas vezes	Muitas vezes	Sempre	Nunca	Poucas vezes	Algumas vezes	Muitas vezes	Sempre	Nunca	Poucas vezes	Algumas vezes	Muitas vezes	Sempre	
		<b>1. Comportamentos/Atitudes</b>																				
Assinala com uma cruz (x) a opção que melhor se adequa à situação.																						
Foi assíduo e Pontual.						X																X
Participação de forma útil na elaboração do projeto.						X																X
Foi organização e trouxe o material necessário.						X																X
Empenhamento nas tarefas propostas e esforço por ultrapassar as suas dificuldades e as do grupo.						X																X
Respeito pelo professor e os colegas.						X																X
Ajudou os colegas com mais dificuldades.						X																X
Autonomia.						X																X
Cumprimento dos prazos estabelecidos.						X																X
Discussão das ideias com clareza.						X																X
Utilização de uma linguagem adequada às diversas situações.						X																X
<b>2. Competências Técnicas</b>																						
Com base nas competências definidas para o módulo, classifica o desempenho assinalando com uma cruz (x) a opção que melhor se adequa.																						
Compreensão do fluxo de execução de uma aplicação.		Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom	
Compreensão e utilização adequada das estruturas de acesso condicional simples ou compostas (if, if..else) numa aplicação.						X					X											X
Compreensão e utilização adequada das estruturas de repetição (for, while, do..while) com ou sem condição de paragem numa aplicação.						X					X											X
Construção de algoritmos eficazes e eficientes.						X					X											X
<b>3. Nota final (0..20)</b>		17 valor				18 valor				16 valor				17 valor								
<b>4. Observações e sugestões</b>																						

Grupo 2

1. Comportamentos/Atitudes

	Grupo 2														
	A1				A8				A10						
	Nunca	Poucas vezes	Alguma s vezes	Muita s vezes	Sempre	Nunca	Pouca s vezes	Alguma s vezes	Muitas vezes	Sempre	Nunca	Poucas vezes	Alguma s vezes	Muitas vezes	Sempre
Assinala com uma cruz (x) a opção que melhor se adequa à situação.															
Foi assíduo e Pontual.					X					X					X
Participação de forma útil na elaboração do projeto.					X					X					X
Foi organização e trouxe o material necessário.					X					X					X
Empenhamento nas tarefas propostas e esforço por ultrapassar as suas dificuldades e as do grupo.				X					X					X	
Respeito pelo professor e os colegas.					X					X					X
Ajudou os colegas com mais dificuldades.					X					X					X
Autonomia.					X					X					X
Cumprimento dos prazos estabelecidos.					X					X					X
Discussão das ideias com clareza.					X					X					X
Utilização de uma linguagem adequada às diversas situações.				X					X					X	
<b>2. Competências Técnicas</b>															
Com base nas competências definidas para o módulo, classifica o desempenho assinalando com uma cruz (x) a opção que melhor se adequa.															
Compreensão do fluxo de execução de uma aplicação.	Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom
Compreensão e utilização adequada das estruturas de acesso condicional simples ou compostas (if, if..else) numa aplicação.				X					X					X	
Compreensão e utilização adequada das estruturas de repetição (for, while, do..while) com ou sem condição de paragem numa aplicação.				X					X					X	
Construção de algoritmos eficazes e eficientes.			X						X				X		
<b>3. Nota final (0..20)</b>			<b>18</b>									<b>14</b>			
			valor									valor			
<b>4. Observações e sugestões</b>															

		Grupo 3														
		A13					A11					A15				
1. Comportamentos/Atitudes		Nunca	Poucas vezes	Alguma vez	Muitas vezes	Sempre	Nunca	Poucas vezes	Alguma vez	Muitas vezes	Sempre	Nunca	Poucas vezes	Alguma vez	Muitas vezes	Sempre
Assinala com uma cruz (x) a opção que melhor se adequa à situação.																
Foi assíduo e Pontual.						X					X					X
Participação de forma útil na elaboração do projeto.						X					X			X		
Foi organização e trouxe o material necessário.						X				X					X	
Empenhamento nas tarefas propostas e esforço por ultrapassar as suas dificuldades e as do grupo.						X				X					X	
Respeito pelo professor e os colegas.						X				X					X	
Ajudou os colegas com mais dificuldades.						X				X				X		
Autonomia.											X				X	
Cumprimento dos prazos estabelecidos.						X				X					X	
Discussão das ideias com clareza.						X				X				X		
Utilização de uma linguagem adequada às diversas situações.						X				X					X	
2. Competências Técnicas																
Com base nas competências definidas para o módulo, classifica o desempenho assinalando com uma cruz (x) a opção que melhor se adequa.																
Compreensão do fluxo de execução de uma aplicação.		Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom
Compreensão e utilização adequada das estruturas de acesso condicional simples ou compostas (if, if..else) numa aplicação.						X					X					
Compreensão e utilização adequada das estruturas de repetição (for, while, do..while) com ou sem condição de paragem numa aplicação.											X					X
Construção de algoritmos eficazes e eficientes.						X				X					X	
3. Nota final (0..20)			18										17			
			valor										valor			
4. Observações e sugestões																

1. Comportamentos/Atitudes	Grupo 4																						
	A5					A2					A12					valor							
	Munca	Poucas vezes	Algumas vezes	Muitas vezes	Sempre	Munca	Poucas vezes	Algumas vezes	Muitas vezes	Sempre	Munca	Poucas vezes	Algumas vezes	Muitas vezes	Sempre								
Assinala com uma cruz (x) a opção que melhor se adequa à situação.																							
Foi assíduo e Pontual.				X						X													X
Participação de forma útil na elaboração do projeto.				X						X													X
Foi organização e trouxe o material necessário.					X					X													X
Empenhamento nas tarefas propostas e esforço por ultrapassar as suas dificuldades e as do grupo.					X					X													X
Respeito pelo professor e os colegas.					X					X													X
Ajudou os colegas com mais dificuldades.					X					X													X
Autonomia.					X					X													X
Cumprimento dos prazos estabelecidos.					X					X													X
Discussão das ideias com clareza.					X					X													X
Utilização de uma linguagem adequada às diversas situações.					X					X													X
<b>2. Competências Técnicas</b>																							
Com base nas competências definidas para o módulo, classifica o desempenho assinalando com uma cruz (x) a opção que melhor se adequa.																							
Compreensão do fluxo de execução de uma aplicação.						Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom			
Compreensão e utilização adequada das estruturas de acesso condicional simples ou compostas (if..else) numa aplicação.										X													X
Compreensão e utilização adequada das estruturas de repetição (for, while, do..while) com ou sem condição de paragem numa aplicação.										X				X									X
Construção de algoritmos eficazes e eficientes.									X					X									X
<b>3. Nota final (0..20)</b>						16																15	
						valor																valor	
<b>4. Observações e sugestões</b>																							

1. Comportamentos/Atitudes		Grupo 5																		
		A3					A6													
		Nunca	Poucos vezes	Alguma s vezes	Muita s vezes	Sempre	Nunca	Pouca s vezes	Alguma s vezes	Muitas vezes	Sempre									
Assinala com uma cruz (x) a opção que melhor se adequa à situação.																				
Foi assíduo e Pontual.					X					X										X
Participação de forma útil na elaboração do projeto.						X														X
Foi organização e trouxe o material necessário.									X											
Empenhamento nas tarefas propostas e esforço por ultrapassar as suas dificuldades e as do grupo.								X											X	
Respeito pelo professor e os colegas.										X										X
Ajudou os colegas com mais dificuldades.		X										X								
Autonomia.						X														
Cumprimento dos prazos estabelecidos.									X											X
Discussão das ideias com clareza.										X										X
Utilização de uma linguagem adequada às diversas situações.															X					
<b>2. Competências Técnicas</b>																				
Com base nas competências de finidas para o módulo, classifica o desempenho assinalando com uma cruz (x) a opção que melhor se adequa.																				
		Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom	Mau	Insuf.	Suf.	Bom	Bom				
Compreensão do fluxo de execução de uma aplicação.				X									X							
Compreensão e utilização adequada das estruturas de acesso condicional simples ou compostas (if..else) numa aplicação.					X									X						
Compreensão e utilização adequada das estruturas de repetição (for, while, do...while) com ou sem condição de paragem numa aplicação.					X								X							
Construção de algoritmos eficazes e eficientes.					X									X						
<b>3. Nota final (0..20)</b>						11											10			
			valor										valor							
<b>4. Observações e sugestões</b>																				

## **Anexo M – Diários de Bordo**

# Diários de Bordo



## Um Robô em Marte - Diário de Bordo Individual

Nome:


Grupo n.º: 1

Data: 21-2-13

Tarefas superadas com sucesso (explicar como foram realizadas e com que estruturas de controlo: *if, if..else, for, while, do..while, etc*): Conseguimos  
(com vários ajustes) pôr o Robot a chegar  
ao alvo, quase sem falhas.

Dificuldades encontradas e limitações da solução: Para a proxima  
avla vamos tentar utilizar o Random  
e fazer ajustes a Nivel do delay e os  
cm do sensor.

Algoritmo ou Fluxograma:

  
- se (Ler sensor frente (1)) > 20  
Andar Para frente  
- senão { Andar Para frente  
virar Para Direita  
se (Ler Sensor Esq < 15  
Andar Para frente  
virar Para Direita  
se Ler sensor Direito < 15  
Andar Para frente  
virar Para Esquerda.  
se Ler sensor Esquerda < 15  
Andar Para frente  
virar Para Direita





Um Robô em Marte - Diário de Bordo Individual

Nome:

Grupo n.º: 1

Data: 21/02/2013

Tarefas superadas com sucesso (explicar como foram realizadas e com que estruturas de controlo: *if, if..else, for, while, do..while, etc*):

*utilizámos if e if else, de modo a conseguirmos que o robô detetasse o sensor*

Dificuldades encontradas e limitações da solução: *Coz quando tentamos fazer uma mala função chamada round.*

Algoritmo ou Fluxograma:

- se (ler sensor frontal)  $( ) > 20$   
= Andar para trás  
=

- se não { Andar para trás  
virar para a direita

se (ler sensor esquerda)  
virar para a direita

se (ler sensor direita)  
virar para a esquerda





Um Robô em Marte - Diário de Bordo Individual

Nome:

Grupo n.º: 2

Data: 25.02.13

Tarefas superadas com sucesso (explicar como foram realizadas e com que estruturas de controlo: *if, if..else, for, while, do..while, etc*):

As mesmas que antes, mas desta vez em vez de andar só para a frente, <sup>quando pode</sup> também o metemos a andar virar para um lado ou para o outro com um do.

Dificuldades encontradas e limitações da solução: Contar as voltas com o for, depois desistimos com o for, porque só dava voltinhas no lugar, e metemos um do.

Algoritmo ou Fluxograma:

metemos uma variável para contar as voltas, nome ctd.  
Metemos o ctd = 0 e lêmos o sensor frontal.  
Se sensor frontal for maior que 20 anda para frente e faz um do para fazer primeiro 2 voltas para a esquerda e fazemos outro do para fazer 2 voltas para direita, senão anda para trás e ver se se pode virar para a direita se sim vira para direita, senão lê o sensor esquerdo e vê se se pode virar para a esquerda se sim vira para esquerda.

## **Anexo N – Avaliação do Produto**

## Avaliação do Produto

Grupo 1		
Avaliação da Solução Apresentada	Alunos	
Membros	Nome: A14	
	Nome: A7	
	Nome: A9	
	Nome: A4	
Componente	Cotação por elemento	Valor
Código		
Estruturação e organização do código	10	10
Adequação das estruturas de controlo às necessidades	15	15
Combinação das estruturas de controlo	13	13
Qualidade Geral	10	10
Algoritmo		
Eficácia	5	5
Eficiência	7	6,5
Otimizações	5	3
Qualidade Geral	5	4,5
Etapas do desafio		
Deteção de Obstáculos Fixos	10	10
Deteção de Obstáculos Móveis	5	2
Contorno de Obstáculos	5	5
Deteção de Alto Relevo	5	4,5
Gestão dos 3 Sensores	5	5
Total (0..100)	100	93,5
<b>Total (0..20)</b>	<b>20</b>	<b>18,7</b>

Grupo 2		
Avaliação da Solução Apresentada	Alunos	
Membros	Nome: A1	
	Nome: A8	
	Nome: A10	
Componente	Cotação por elemento	Valor
Código		
Estruturação e organização do código	10	10
Adequação das estruturas de controlo às necessidades	15	15
Combinação das estruturas de controlo	13	13
Qualidade Geral	10	10

<b>Algoritmo</b>		
Eficácia	5	5
Eficiência	7	5
Otimizações	5	3
Qualidade Geral	5	4
<b>Etapas do desafio</b>		
Deteção de Obstáculos Fixos	10	10
Deteção de Obstáculos Móveis	5	3
Contorno de Obstáculos	5	4
Deteção de Alto Relevio	5	4,5
Gestão dos 3 Sensores	5	5
Total (0..100)	100	91,5
<b>Total (0..20)</b>	<b>20</b>	<b>18,3</b>

<b>Grupo 3</b>		
<b>Avaliação da Solução Apresentada</b>	<b>Alunos</b>	
Membros	Nome: A13	
	Nome: A11	
	Nome: A15	
<b>Componente</b>	<b>Cotação por elemento</b>	<b>Valor</b>
<b>Código</b>		
Estruturação e organização do código	10	8
Adequação das estruturas de controlo às necessidades	15	14
Combinação das estruturas de controlo	13	11
Qualidade Geral	10	8
<b>Algoritmo</b>		
Eficácia	5	4,5
Eficiência	7	5,5
Otimizações	5	1
Qualidade Geral	5	4
<b>Etapas do desafio</b>		
Deteção de Obstáculos Fixos	10	8
Deteção de Obstáculos Móveis	5	2
Contorno de Obstáculos	5	4
Deteção de Alto Relevio	5	4,5
Gestão dos 3 Sensores	5	4,5
Total (0..100)	100	79
<b>Total (0..20)</b>	<b>20</b>	<b>15,8</b>

Grupo 4		
Avaliação da Solução Apresentada	Alunos	
Membros	Nome: A5	
	Nome: A2	
	Nome: A12	
Componente	Cotação por elemento	Valor
Código		
Estruturação e organização do código	10	8
Adequação das estruturas de controlo às necessidades	15	12
Combinação das estruturas de controlo	13	11
Qualidade Geral	10	8
Algoritmo		
Eficácia	5	4,5
Eficiência	7	5
Otimizações	5	2
Qualidade Geral	5	4
Etapas do desafio		
Deteção de Obstáculos Fixos	10	9
Deteção de Obstáculos Móveis	5	3
Contorno de Obstáculos	5	4
Deteção de Alto Relevo	5	3
Gestão dos 3 Sensores	5	4
	Total (0..100)	77,5
	<b>Total (0..20)</b>	<b>15,5</b>

Grupo 5		
Avaliação da Solução Apresentada	Alunos	
Membros	Nome: A3	
	Nome: A6	
Componente	Cotação por elemento	Valor
Código		
Estruturação e organização do código	10	3
Adequação das estruturas de controlo às necessidades	15	10
Combinação das estruturas de controlo	13	8
Qualidade Geral	10	4
Algoritmo		
Eficácia	5	2
Eficiência	7	3
Otimizações	5	1
Qualidade Geral	5	2

<b>Etapas do desafio</b>		
Deteção de Obstáculos Fixos	10	5
Deteção de Obstáculos Móveis	5	1
Contorno de Obstáculos	5	1
Deteção de Alto Relevo	5	0
Gestão dos 3 Sensores	5	3
Total (0..100)	100	43
<b>Total (0..20)</b>	<b>20</b>	<b>8,6</b>