

UNIVERSIDADE DE LISBOA



RELATÓRIO DA PRÁTICA DE ENSINO SUPERVISIONADA

A CONSOLIDAÇÃO DE CONCEITOS DE PROGRAMAÇÃO
UTILIZANDO A ROBÓTICA EDUCATIVA

Rui Pedro Leal da Ressurreição

MESTRADO EM ENSINO DE INFORMÁTICA

2012

UNIVERSIDADE DE LISBOA



RELATÓRIO DA PRÁTICA DE ENSINO SUPERVISIONADA

A CONSOLIDAÇÃO DE CONCEITOS DE PROGRAMAÇÃO
UTILIZANDO A ROBÓTICA EDUCATIVA

Rui Pedro Leal da Ressurreição

MESTRADO EM ENSINO DE INFORMÁTICA

Trabalho orientado pelo Professor Doutor João Filipe Matos

2012

"As coisas, por si sós, não são interessantes, mas tornam-se interessantes apenas se nos interessamos por elas."

In Silvio Ceccato (1985) Ingegneria della felicità

Resumo

A dificuldade em compreender e aplicar conceitos abstratos de programação pelos alunos, foi identificada como um problema na aprendizagem da programação. O presente relatório descreve uma prática de ensino supervisionada, com o objetivo de consolidar alguns conceitos de programação numa turma do 12º do curso profissional de Técnico de Gestão de Informática. Foram lecionadas cinco aulas do módulo 17 da disciplina de Linguagens de Programação a um grupo de sete alunos.

Como estratégia de ensino, foi apresentado um problema aos alunos sobre o estacionamento automático de veículos. Os alunos trabalharam em grupo com o objetivo de encontrar soluções e programá-las nos robôs da Lego® Mindstorms® NXT, para três tipos de estacionamento: em espinha, em paralelo e em linha. A Robótica Educativa constituiu o instrumento principal para o ensino dos conceitos de programação, juntamente com um conjunto de princípios e estratégias pedagógicas.

A avaliação das aprendizagens incidiu principalmente na realização de um questionário de autodiagnóstico e de um teste de diagnóstico no início da intervenção, e na realização de um questionário de autoavaliação e de um teste de avaliação no final da intervenção. Após a intervenção foi realizada uma discussão de grupo para avaliar a perceção da turma em relação à intervenção.

Verificou-se que a Robótica Educativa, em associação com um conjunto de princípios e estratégias pedagógicas, poderá ter contribuído para a consolidação de conceitos de programação.

Palavras-chave: programação, aprendizagem, ensino, estacionamento automático de veículos, robótica educativa

Abstract

The difficulty in understanding and applying abstract concepts of computer programming by students was identified as a problem in learning programming.

This report describes a practice of supervised teaching in order to allow the consolidation of some programming concepts in a 12th grade class of a professional course of Management of Information Technology Technician. Five lessons were given to a group of seven students in the Unit 17 of the subject Programming Languages

As a teaching strategy, a problem was presented to the students about the automatic parking of vehicles. The students worked in groups with the purpose of finding solutions and programming them in Lego® Mindstorms® NXT robots, for three types of parking: angle, perpendicular and parallel. Educational Robotics using the Lego robots was the main tool for teaching programming concepts, along with a set of principles and teaching strategies.

The evaluation focused on a self-diagnostic questionnaire and a diagnostic test at the beginning of the intervention, as well as a self-assessment questionnaire and an evaluation test at the end of the intervention. A focus group was conducted after the intervention, to assess the perception of the class about the intervention.

It was found that Educational Robotics, in association with a set of principles and teaching strategies, might have contributed to the consolidation of programming concepts.

Keywords: computer programming, learning, teaching, automatic parking of vehicles, educational robotics

Índice

1. Introdução.....	11
2. Enquadramento.....	14
2.1. Programação.....	14
2.1.1. Problemas/dificuldades identificadas no ensino da programação.....	15
2.1.2. Soluções identificadas para o ensino da programação.....	17
2.1.2.1. Robótica educativa.....	20
2.1.2.2. Linguagem de programação visual.....	21
2.1.2.3. Programação em pares.....	22
2.2. Robótica	23
2.3. Estacionamento Automático de Veículos	24
3. Contexto Escolar	26
3.1. Caracterização da Escola	26
3.1.1. Ações educativas e pedagógicas.....	28
3.2. Caracterização da Disciplina	29
3.3. Caracterização da Unidade Didática.....	33
3.3.1. Conceitos envolvidos.....	35
3.4. Caracterização da Turma.....	37
4. Metodologia	46
4.1. Princípios e Estratégias Pedagógicas.....	47
4.1.1. Robótica educativa.....	47
4.1.2. Aprendizagem baseada em problemas.....	49
4.1.3. Princípios pedagógicos.....	51
4.2. Recursos	54
4.2.1. Lego® mindstorms® nxt 2.0.....	55
4.2.2. Instrumentos de recolha de dados.....	56
4.3. Avaliação da Intervenção	57
4.3.1. Avaliação das aprendizagens.....	59
5. Concretização das aulas.....	61
5.1. Primeira Aula	61
5.2. Segunda Aula.....	62

5.3. Terceira Aula	64
5.4. Quarta Aula	65
5.5. Quinta Aula.....	66
6. Resultados	67
6.1. Soluções Programadas	67
6.2. Páginas Criadas na Plataforma PBworks	68
6.3. Questionários de Autodiagnóstico e de Autoavaliação	69
6.4. Testes de Autodiagnóstico e de Autoavaliação	71
6.5. Registos de Aulas	75
6.6. Discussão de Grupo	79
7. Discussão dos Resultados.....	82
8. Reflexão Crítica	90
9. Referências	94
10. Anexos	100
Anexo A – Inquérito de Caracterização da Turma.....	101
Anexo B – Planos de Aula.....	105
Anexo C – Autorizações para a Recolha de Dados.....	117
Anexo D – Plataforma PBworks	121
Anexo E – Plataforma Moodle	122
Anexo F – Apresentação das Aulas Realizada aos Alunos	123
Anexo G – Vídeo Demonstrativo do Estacionamento Automático	124
Anexo H – Questionário.....	125
Anexo I – Teste de Diagnóstico	127
Anexo J – Teste de Avaliação	131
Anexo K – Grelha de Registos.....	135
Anexo L – Guião para a Discussão de Grupo	137
Anexo M – Critérios de Avaliação do Teste de Diagnóstico.....	139
Anexo N - Critérios de Avaliação do Teste de Avaliação	144
Anexo O – Relação Entre os Questionários e os Testes	149

Índice de quadros

Quadro 1	27
Quadro 2	30
Quadro 3	33
Quadro 4	39
Quadro 5	40
Quadro 6	41
Quadro 7	42
Quadro 8	43
Quadro 9	44
Quadro 10	45
Quadro 11	70
Quadro 12	76
Quadro 13	77
Quadro 14	78

Índice de figuras

<i>Figura 1.</i> Sala TIC na ESPAV.	26
<i>Figura 2.</i> Exemplificação da maquete apresentada aos alunos, com os três tipos de estacionamento, espinha, paralelo e linha.	49
<i>Figura 3.</i> Respostas sobre o nível de conhecimento para cada conceito indicado no questionário de autodiagnóstico.	70
<i>Figura 4.</i> Respostas sobre o nível de conhecimento para cada conceito indicado no questionário de autoavaliação.	71
<i>Figura 5.</i> Respostas por nível de conhecimento para cada conceito abordado no grupo I do teste de diagnóstico.	73
<i>Figura 6.</i> Respostas em que se manifestou reconhecer cada conceito abordado no grupo II do teste de diagnóstico e de avaliação.	73
<i>Figura 7.</i> Respostas por nível de conhecimento para cada conceito abordado no grupo I do teste de avaliação.	74
<i>Figura 8.</i> Percentagem de alunos que afirmaram, no questionário de autodiagnóstico, conhecer e saber aplicar um conceito e que reconheceram a aplicação desse conceito no grupo II do teste de diagnóstico.	82
<i>Figura 9.</i> Percentagem de alunos que afirmaram, no questionário de autoavaliação, conhecer e saber aplicar um conceito e que reconheceram a aplicação desse conceito no grupo II do teste de avaliação.	83
<i>Figura 10.</i> Percentagem de alunos que afirmaram no questionário de autodiagnóstico conhecer e saber aplicar um conceito e que aplicaram o conceito no grupo I do teste de diagnóstico.	84
<i>Figura 11.</i> Percentagem de alunos que afirmaram no questionário de autoavaliação conhecer e saber aplicar um conceito e que aplicaram o conceito no grupo I do teste de avaliação.	85

1. Introdução

Desde 2002, a utilização de computadores e Internet tem vindo a aumentar na população portuguesa, estendendo-se atualmente à maioria da população (Instituto Nacional de Estatística [INE], 2011). Esse uso é principalmente feito por indivíduos com o nível de escolaridade do ensino secundário e superior (Rodrigues & Mata, 2003). A tendência de aumento da utilização de computadores e Internet verifica-se também nos restantes países da Europa, mas é de realçar que Portugal continua abaixo da média europeia (Eurostat, INE & Ministérios, 2010). Provavelmente, esta tendência de aumento continuará a verificar-se.

Os programas utilizados nos computadores e as páginas de Internet são construídos usando linguagens de programação, podendo grande parte dos problemas do dia-a-dia, ao nível profissional ou pessoal, serem solucionados com recurso a estas ferramentas. No entanto, apenas uma parte da população, nomeadamente os Programadores ou Engenheiros de Software, sabem como fazê-lo, desenvolvendo a maioria dos programas que utilizamos e as páginas de Internet que consultamos.

A massificação do uso de computadores e Internet e a necessidade de criar programas aumenta a importância do ensino da informática, incluído o ensino da programação. O ensino da programação está praticamente restringido a cursos profissionais, equivalentes ao ensino secundário, e a cursos superiores. No entanto, em relação a outras aprendizagens, Gomes, Henriques e Mendes (2008) indicam que é na aprendizagem da programação, inclusive a um nível mais básico, onde os alunos sentem maiores dificuldades. Uma das causas para essas dificuldades é a desadequação dos métodos de ensino e de aprendizagem, nomeadamente no ensino da sintaxe de uma linguagem de programação e da resolução de problemas (Gomes et al., 2008; Jenkins, 2002). É então fundamental que os professores das disciplinas de programação investiguem novos métodos.

O professor tem o direito de participar no processo educativo, através de recomendações que possa dar em relação ao sistema educativo e de intervir na orientação pedagógica, através da escolha de métodos de ensino, tecnologias, meios auxiliares e técnicas de educação (Ministério da Educação, 1998). Assim, desempenha o papel de gestor do currículo, em que coloca hipóteses, analisa, integra, seleciona, organiza e decide, sobre o currículo e as estratégias adequadas a cada

situação concreta, tendo sempre como referência as orientações do currículo nacional (Roldão, 2009). É então exigida ao professor a capacidade de flexibilização nas suas metodologias e estratégias de ensino, não restringindo as suas ações a regras de atuação rígidas e uniformes (Guerreiro, 1986). As estratégias a utilizar em sala de aula devem ser concebidas de uma forma pensada, estruturada e organizada, respeitando os seus deveres profissionais para com os alunos. Outro papel desempenhado pelo professor é o de investigador das suas metodologias de ensino. Assumir o ensino como objecto de investigação permite aumentar o conhecimento do professor, diversificar os papéis que o professor desempenha e orientar a sua ação pedagógica para os alunos, tornando-o mais atento às situações em sala de aula.

O processo de investigação é um modo de aprendizagem de quem a realiza. A investigação realizada pelo professor incide sobre o planeamento e a execução da sua ação educativa. Segundo Moreira (2005), como professores aprendemos com aqueles que tentamos ensinar, cabendo-nos criar as condições ao desenvolvimento de conhecimentos, capacidades e atitudes, que beneficiarão os alunos a longo prazo. Temos que nos centrar na pessoa que é o aluno, incentivando-o a ser aquilo que é capaz de ser e permitindo que seja criativo, construtivo e transformador. A relação entre a teoria, a prática e a ética da prática, confere a integração do pensamento e da ação, do sentimento e da emoção, do crescimento pessoal e do desenvolvimento profissional.

O presente trabalho pretende contribuir para encontrar vias de solução para o ensino da programação. Para tal, realizou-se uma intervenção pedagógica com o objetivo de consolidar e/ou adquirir conceitos de programação utilizando-se como metodologia a Robótica Educativa e um conjunto de estratégias pedagógicas. A intervenção decorreu numa turma da disciplina de Linguagens de Programação do curso profissional de Técnico de Informática de Gestão. No final da intervenção, os alunos deveriam conseguir reconhecer a aplicação dos conceitos em programas já criados, saber aplicar os conceitos na programação e saber criar programas que resolvam um problema.

Este relatório encontra-se dividido em diversas secções. Na segunda secção é realizado um enquadramento da problemática, identificando as dificuldades no ensino e aprendizagem da programação e as soluções encontradas. A terceira secção, apresenta o contexto escolar e curricular em que decorreu a intervenção, através de uma caracterização da escola, da disciplina, do módulo e da turma. De seguida, na

quarta secção é descrita a metodologia aplicada, indicando os princípios e as estratégias pedagógicas seguidas, os recursos utilizados e o método na avaliação das aprendizagens e da intervenção. A quinta secção, é realizado uma descrição sumária das aulas em que decorreu a intervenção. Na sexta secção, são apresentados os resultados. De seguida, na sétima secção é realizada um discussão dos resultados obtidos. No final, ma oitava secção, é apresentada uma reflexão do trabalho desenvolvido e dos resultados obtidos.

2. Enquadramento

2.1. Programação

Um programador é aquele que é capaz de usar o computador eficientemente, sabendo criar um programa num ficheiro, compilá-lo e encontrar os resultados gerados (Sloane & Linn, 1988), bem como realizar testes ao programa desenvolvido de forma a detetar erros e corrigi-los (Gomes, Henriques & Mendes, 2008). Para além disso, um programador é aquele que analisa um problema, produz uma solução algorítmica e a traduz num código usando linguagens de programação (Saeli, Perrenet, Jochems & Zwaneveld, 2011). A nível profissional, poderá ser dividido em duas funções, o Engenheiro de Software e o Programador de Computadores. O Engenheiro de Software desenha e desenvolve programas, aplicando as teorias da ciência computacional e da matemática para criar, testar e avaliar programas. O Programador de Computadores coloca em prática o desenho de programas criado pelo Engenheiro de Software, convertendo-o num conjunto de instruções lógicas a seguir pelo computador (United States Department of Labor, 2009).

De acordo com Kanakadoss (2005), a programação é um processo de transformação de um plano mental, que se encontra estruturado de uma forma familiar para a pessoa, numa outra forma que é compatível com a linguagem compreendida pelo computador. Deste modo, para Mayer (citado por Kanakadoss, 2005), um aluno deve conhecer em detalhe os processos a decorrer no computador para se tornar um programador de sucesso. Sobral (2008) divide a programação em quatro fases:

1. Conceptualização e análise, em que os problemas são divididos em tarefas mais pequenas;
2. Resolução e criação de algoritmos usando pseudocódigo que traduza os problemas e as tarefas da fase anterior;
3. Utilização de variáveis e estruturas de dados para caracterizar o modelo de dados a utilizar;
4. Tradução do algoritmo numa linguagem de programação e a sua execução.

Para além destas fases, a autora menciona outras, como a verificação de erros e a comparação dos resultados obtidos com os esperados.

2.1.1. Problemas/dificuldades identificadas no ensino da programação.

Gomes, Henriques e Mendes (2008) mencionam que têm existido elevados níveis de insucesso e de desistência nas disciplinas em que é ensinada a programação ao nível básico, independentemente do grau e sistema de ensino. Byrne e Lyons (citado por Gomes et al., 2008) referem que não há nada inerentemente difícil no assunto, mas existem alunos que não têm as aptidões necessárias para programar, nomeadamente as capacidades de resolução de problemas e de raciocínio matemático.

Um dos fatores que tem levado ao insucesso e desistência é o tipo de ensino da programação realizado pelos professores (Gomes et al., 2008). Num estudo realizado no Reino Unido, foram identificadas lacunas no conhecimento fundamental de programação por parte dos professores de informática no ensino superior (Major, Kyriacou & Brereton, 2011). A maioria dos professores envolvidos no estudo não se mostraram confiantes no ensino da programação, podendo vir a resultar em métodos de ensino desadequados.

A falta de competências básicas dos alunos é outro fator de dificuldade na aprendizagem da programação. Quem inicia a aprendizagem da programação, enfrenta dificuldades em transformar as suas formas de raciocínio compatíveis com as do computador (Kanakadoss, 2005; Sobral, 2008), nomeadamente na combinação das várias instruções de código utilizadas no desenvolvimento de algoritmos (Kanakadoss, 2005). Para o desenvolvimento de um algoritmo é necessário que se possua conhecimento da sintaxe e da semântica, mas também capacidade de resolução de problemas. A combinação das duas permite a identificação e estruturação de várias instruções, de forma a formular o algoritmo que resolve um determinado problema (Coull & Duncan, 2011). Se o aluno não possuir capacidades de resolução de problemas, terá dificuldade em qualquer linguagem de programação, apesar do conhecimento que possa ter da sintaxe e da semântica dessa linguagem.

Também foi identificada nos alunos uma grande dificuldade em compreender e aplicar certos conceitos abstratos de programação, bem como na compreensão e aplicação de noções básicas, como por exemplo, as estruturas de controlo e a criação de algoritmos que resolvam problemas concretos (Gomes et al., 2008; Jenkins, 2002). Para além disso, a falta de ligação da programação realizada pelos alunos com a vida real é outro fator para o insucesso da aprendizagem da programação,

justificado pela natureza abstrata da tarefa de programar (Dunican, 2002). Noções como variáveis, tipos de dados, memória dinâmica, entre outros, não têm correspondência na vida do dia-a-dia dos alunos e a compreensão destes conceitos fundamentais de programação não é considerada simples. No entanto, Sobral (2008) afirma que mesmo que se estabeleça uma ligação do problema a situações da vida real, os alunos têm dificuldade em esquematizar o problema e realizar a sua programação.

As próprias linguagens de programação utilizadas são apontadas como fatores para o insucesso. As sintaxes usadas em linguagens de programação e o seu excesso de complexidade são em geral pouco adequadas para programadores inexperientes, apesar de não o serem ao nível profissional (Jenkins, 2002; Motil & Epstein, 2000). As exigências rígidas em termos de sintaxe, quando comparada com a natureza inexacta e livre da língua materna dos alunos, faz com que muitos não sejam capazes de escrever programas compiláveis com sucesso (Jenkins, 2001; Dunican, 2002). A aprendizagem da programação e a utilização de instruções estão intrinsecamente ligadas a diversas complexidades, uma vez que diversos problemas podem ser resolvidos de diferentes formas (Bossé & Nandakumar, 2000). Dessa forma, os alunos vêm-se com diversas opções, o que na maior parte das vezes, acaba por complicar, em vez de simplificar o processo de resolução de problemas.

A desmotivação dos alunos é outro fator que dificulta a aprendizagem da programação (Gomes et al., 2008; Jenkins, 2001). Segundo Dijkstra (citado por Gomes et al., 2008), a desmotivação pode ser causada pelo tipo de aprendizagem, que é um processo lento e gradual, observando-se falta de interesse por parte dos alunos. A desmotivação pode estar associada a uma forte carga de conceitos abstratos que intervêm em todo o conhecimento envolvido na programação onde as características próprias, cada vez mais sofisticadas, das linguagens, da máquina e do ambiente de programação, tendem a dificultar a programação (Almeida, Costa, Silva, Paes, Almeida & Braga, 2002; Benzeltout & Blanchfield, 2009; Gomes & Mendes, 2007). A imagem negativa dos programadores, como um “desadequado social”, apresenta-se como um fator desmotivador, uma vez que é pouco provável que os alunos aspirem a uma imagem deste tipo (Gomes et al., 2008; Jenkins, 2002).

2.1.2. Soluções identificadas para o ensino da programação.

Em relação aos métodos de ensino, este deveria ser mais personalizado, com respostas e supervisão adequadas às necessidades de cada aluno e com diversificação de estratégias, devendo ser contemplada a diversidade de pensamentos, compreensões, ritmos e estilos de aprendizagem (Gomes, Henriques & Mendes, 2008). A utilização de materiais de natureza estática, como apresentações projetadas, explicações verbais, diagramas, desenhos no quadro e textos, bem como o ensino dos detalhes de sintaxe de uma linguagem de programação antes que os alunos percebam qual a finalidade e utilidade de aprender programação, não promovem uma completa compreensão da dinâmica envolvida na programação. No caso da codificação, isto implica que os alunos aprendam a sintaxe básica e gradualmente aprendam a semântica, estrutura e finalmente o estilo, sendo afirmado por Gomes et al. (2008) que estas abordagens não têm resultado.

É referido por Gomes et al. (2008) que nas atividades de resolução de problemas, estas devem ser adequados ao nível cognitivo do aluno e aos seus estilos preferenciais de aprendizagem. O professor poderá aplicar um questionário aos alunos de forma a determinar o tipo de método que pode ser aplicado. No entanto, esse método deve focar na linguagem de programação como uma ferramenta para resolver problemas e devem ser feitos exercícios de forma gradual e progressiva. Numa primeira fase, os exercícios devem promover e captar a atenção do aluno, promovendo a sua motivação e posteriormente, deve-se ir avançando na dificuldade. Sendo assim, os mesmos autores referem que deve-se promover o estudo da programação, por meio de resoluções de problemas, verificando se o aluno percebeu o que lhe é pedido. Isto pode ser confirmado fazendo questões e propondo exercícios que tenham relação com outros anteriores. Se houver a indicação que o aluno não entendeu, então deve dividir-se o exercício em partes e relacioná-lo com outros já resolvidos pelo mesmo aluno e mostrar-lhe a relação.

Para se ter conhecimento e domínio na programação, é essencial adquirir-se bons conhecimentos matemáticos (Sloane & Linn, 1988) e de resolução de problemas (Gomes et al., 2008; Sloane & Linn, 1988). Gomes et al. (2008) afirmam que a linguagem de programação deve ser considerada como uma ferramenta para concretizar a solução do problema, devendo existir persistência na procura de soluções de problemas pelos alunos, que não se encontram normalmente de forma

simples e rápida. Os autores mencionam ainda que os alunos não fazem testes ao programa desenvolvido ou fazem-no superficialmente ou de uma forma reduzida, sem verificação de casos limite. Para além dos métodos de trabalho não serem os mais adequados, os autores referem que os métodos de estudo dos alunos também não o são, uma vez que são baseados em leituras sucessivas, memorização de fórmulas e mecanização de procedimentos. A programação exige uma compreensão dos assuntos e reflexão (Gomes et al., 2008; Sloane & Linn, 1988) e uma prática intensiva complementada com um intenso trabalho extra aulas (Gomes et al., 2008). Gomes et al. (2008) sugerem que deve-se permitir um treino intensivo de resolução de problemas de forma a que o aluno siga todas as etapas para uma correta solução. Essas etapas são:

- a compreensão do problema,
- a caracterização do problema,
- a representação do problema,
- a solução do problema e
- a reflexão sobre a solução obtida.

Os autores mencionam que a prática regular da reflexão acerca da forma de resolução de determinado problema, bem como o questionamento sobre novas propostas de solução em função de alterações no enunciado, constituiria uma mais-valia para a aprendizagem dos alunos. Gomes e Mendes (2007) afirmam que caso se verifique falta de conhecimento específico na resolução de um problema, podem ser dadas pistas de resolução e pedir-se aos alunos que iniciem o desenvolvimento de um programa a partir de um esboço. Podendo também ser pedido aos alunos que analisem e encontrem erros a partir de programas incompletos, em vez de iniciarem programas de raiz sem que as bases da programação estejam bem fundamentadas.

Para aumentar a motivação dos alunos, o professor tem um papel importante, devendo procurar criar um ambiente estimulante, mostrar-se paciente e bem disposto e tentar providenciar um acompanhamento permanente (Gomes & Mendes, 2007). De acordo com esse papel, o professor pode ainda incluir animações e modelos dinâmicos que melhor representem os vários conceitos de programação e apresentar atividades de acordo com o estilo de aprendizagem preferencial de cada aluno, ritmo e estado cognitivo, exibindo um ambiente estruturado, que assegure metodologias de estudo corretas, levando à reflexão e questionamento permanente.

Gomes et al. (2008) afirmam que o professor deve ainda oferecer uma abordagem gradual da programação. Em primeiro lugar, deverá desenvolver competências básicas de resolução de problemas, de planeamento de soluções e mostrando a utilidade da programação. Mais tarde, poderá abordar os detalhes sintáticos das linguagens de programação. De acordo com estes autores, existiria inicialmente uma minimização dos detalhes sintáticos de uma linguagem de programação.

Para diminuir a abstração associada à programação, Gomes e Mendes (2007) referem que deve-se utilizar metáforas e exemplos concretos conhecidos do aluno e permitir um treino intensivo de conhecimentos matemáticos e lógicos úteis à programação.

Gomes e Mendes (2007) referem ainda que, como forma de os alunos ganharem experiência em programação, o professor deve dar sugestões para atingir determinadas soluções, propondo atividades diversificadas, nomeadamente programas completos para os alunos analisarem, programas que contenham erros lógicos habitualmente cometidos e programas incompletos para completar. O professor poderá ainda apresentar modelos, para que os alunos adquiram as melhores práticas de programação. É referido pelos autores, que quando o aluno chega a um resultado deve levar-se o aluno a verificar se a solução é coerente, fazendo questões de modo a verificar a solução, permitindo aumentar e/ou manter a motivação.

É também importante explicar aos alunos que programar exige tempo e muita prática, por isso deve proporcionar-se tempo e um local para praticar, estimulando, ao mesmo tempo, a motivação. Uma outra solução para manter a motivação é a utilização da programação em pares, que é também apontada como estratégia para aumentar a produtividade e reduzir erros de programação (Carter & Jenkins, 2010; Coleman & Nichols, 2010).

Como apoio ao ensino e aprendizagem da programação, foi identificado o uso da Robótica Educativa e de Linguagens de Programação Visual (LPV), diminuindo o carácter abstrato da programação e aumentando a motivação e a capacidade de transformar algoritmos em programas (Benitti, Vahldick, Urban, Krueger & Halma, 2009; Benzeltout & Blanchfield, 2009; Goel & Kathuria, 2010; Páztor, Pap-Szigetu & Torok, 2010; Santos & Costa, 2006).

2.1.2.1. Robótica educativa.

A aplicação da robótica na educação, com o objetivo de aumentar o rendimento e o grau de aprendizagem dos alunos, designa-se Robótica Educativa. Esta consiste na aprendizagem por meio de montagem de mecanismos que apresentam alguma atividade física, como movimento mecânico, levantamento de objetos, entre outros (Bacaroglo, 2005).

Para além da construção do robô é exigida a criação de programas que permitem a realização de tarefas específicas. As tarefas programadas incidem principalmente na movimentação, no controlo, na definição de caminhos e na utilização dos sensores para interagir com o meio envolvente (Rocha, 2006). Assim, o aluno tem a possibilidade de interagir com o concreto, através do robô, e com o abstrato, através do programa desenvolvido, tendo a oportunidade de observar o comportamento do robô consequente da programação (Páztor, Pap-Szigetu & Torok, 2010). No ensino/aprendizagem da programação, a utilização da Robótica Educativa é focada no desenvolvimento de algoritmos e programas que controlam o robô construído pelos alunos de acordo com o problema colocado (Páztor et al., 2010; Rocha, 2006).

Em alguns estudos em que foi aplicada a Robótica Educativa usando os robôs da Lego® Mindstorms® NXT, foi concluído que permitiu melhorar as aprendizagens de conceitos de programação (Sartzemi, Dagdilelis & Kagani, 2008), ajudou a antecipar os comportamentos do robô de acordo com a programação efetuada (Wu, Tseng & Huang, 2008), ajudou a compreender os níveis abstratos da programação (Páztor et al., 2010) e melhorou a motivação para o estudo da programação (Páztor et al., 2010; Sartzemi et al., 2008). No entanto, nesses mesmos estudos é referido que os resultados obtidos que levaram a estas conclusões, não se deveram apenas à utilização da robótica educativa, mas à sua associação com princípios, métodos e estratégias de ensino usadas nas aulas. No estudo realizado por Páztor et al. (2010), um dos métodos referidos foi a Aprendizagem Baseada em Problemas, em que os alunos trabalharam em grupo, tiveram que apresentar e discutir as soluções com os colegas e o professor assumiu o papel de motivador e orientador. Apesar dos resultados destes estudos, os seus autores mencionam que é necessário desenvolver mais estudos sobre a utilização da robótica educativa no ensino da programação.

2.1.2.2. Linguagem de programação visual.

É fundamental analisar como os alunos pensam na resolução dos seus problemas do dia-a-dia, como programariam de acordo com esses problemas e manter os alunos conscientes do que acontece no sistema, fornecendo-lhes respostas ao seu trabalho (Kanakadoss, 2005). Assim, a utilização de sistemas informáticos de acompanhamento e visualização apresenta diversos benefícios, nomeadamente as possibilidades de incluir elementos numa animação, transmitindo informação mais detalhada ao aluno, de verificar como é que um conjunto de dados afeta o funcionamento do algoritmo desenvolvido, de executar e simular resoluções do problema em tempo real, traduzindo-se num melhor estudo e compreensão do comportamento do algoritmo e de motivar os alunos através de animações em vez da utilização de materiais estáticos (Mendes, 2002). No entanto, é importante de realçar que o valor real da utilização de animações é permitir que o aluno intervenha, construa, explore, modifique, experimente e teste as suas teorias, potenciando vários níveis de interação.

As Linguagens de Programação Visual (LPV) baseiam-se no princípio de que a utilização de uma representação gráfica é mais fácil de entender do que uma representação em texto e melhora a interface com o utilizador, diminuindo a dificuldade na programação (Gudwin, 1997; Kanakadoss, 2005). As LPV permitem criar programas usando unicamente a manipulação de gráficos, desenhos, animações, ícones e diagramas, como por exemplo o AgentSheets, Alice e o NXT-G. No entanto, há a necessidade de distinguir as LPV das ferramentas de programação visual, cujos gráficos e diagramas não especificam um programa ou algoritmo, como por exemplo as ferramentas para a criação de interfaces e as linguagens de programação visual que utilizam uma programação visual e de texto, das quais se destaca o Visual Basic (Gudwin, 1997).

De acordo com Myers (citado por Kanakadoss, 2005) as LPV são mais fáceis e naturais de ler, são autoexplicativas, reduzem a complexidade da sintaxe a utilizar e permitem uma melhor modelação do código. O mesmo autor menciona que o cérebro humano está otimizado para representações visuais, pelo que o uso de fluxogramas torna mais fácil a transposição de ideias do programador para o computador e o uso de imagens ajuda na compreensão de conceitos abstratos.

Segundo Mendes (2002) os iniciados em programação é que poderão beneficiar mais na utilização das LPV, uma vez que requer menos pensamento cognitivo e o processamento visual humano é otimizado para dados multidimensionais. De acordo com este autor, neste tipo de programação é realizado um mapeamento de representações computacionais em representações perceptuais ou visuais, permitindo “visualizar” o abstracto, escolhendo técnicas de codificação que possam maximizar a compreensão de quem aprende. Para além disso, permite ao aluno visualizar como o seu algoritmo se comporta, potenciar a deteção de eventuais erros, corrigi-los e através desse processo aprender a programar.

2.1.2.3. Programação em pares.

A Programação em Pares é um método recente e particular da aprendizagem colaborativa, que permite que um par de alunos trabalhe em conjunto para atingir um fim comum (Preston, 2006). Utilizando esta estratégia, dois alunos trabalham lado a lado no desenho e escrita de código que traduz um algoritmo (Goel & Kathuria, 2010). Dessa forma, permite uma aprendizagem mais eficaz, uma vez que os alunos podem resolver problemas com outros colegas colaborativamente. Para além disso, a programação em pares melhora aspetos concretos da programação, nomeadamente na redução de erros, reforço de competências técnicas e melhoria da comunicação entre os alunos (Coleman & Nichols, 2010).

Em empresas de Tecnologias de Informação, é comum a programação em pares ser realizada por programadores com diferentes conhecimentos e experiência profissional. O mesmo pode ser realizado na educação, juntando alunos com diferentes níveis de conhecimento sobre programação (Coleman & Nichols, 2010), o que diminui a dependência em relação ao professor, aumenta a satisfação de realizar a tarefa e aumenta a compreensão do processo (Preston, 2006).

Para além da tradicional forma de programação a pares, fisicamente no mesmo local, a programação em pares pode ser remota ou virtual, em que dois alunos em diferentes locais, trabalham através de um editor em tempo-real ou através da partilha do ambiente de trabalho do sistema operativo (Goel & Kathuria, 2010). Seja qual for o caso, normalmente cada aluno desempenha um de dois papéis: o de “condutor”, que controla as execuções no computador através, por exemplo, do teclado e do rato ou o de “navegador”, que verifica ou orienta o trabalho do

“condutor”, dando sugestões e correções a aplicar no desenho e codificação do algoritmo.

2.2. Robótica

A Robótica é a área da ciência responsável pelo estudo da construção e do funcionamento de robôs, na qual são utilizados, conjuntamente, conceitos de Inteligência Artificial, Mecânica, Cinemática, Informática e Hidráulica (Bacaroglo, 2005).

Robôs são dispositivos mecânicos equipados com sensores que atuam sob o controle de um sistema informático, executando movimentos limitados a um espaço físico (Angeles, 2007; Rocha, 2006). Estes, podem ser classificados por (Angeles, 2007; Bacaroglo, 2005):

- robôs inteligentes que são manipulados por sistemas multifuncionais controlados por computador, sendo capazes de interagir com o seu ambiente, através de sensores e de tomar decisões em tempo real;
- robôs com controle por computador que são semelhantes aos robôs inteligentes, mas não tem a capacidade de interagir com o ambiente;
- robôs de aprendizagem que se limitam a repetir uma sequência de movimentos, realizados com a intervenção de um operador ou memorizados;
- robôs manipuladores que consistem em sistemas mecânicos multifuncionais, cujo sistema de controle permite movimentar os seus membros de uma forma manual, quando o operador controla diretamente os movimentos, ou de uma forma de sequência variável, quando é possível alterar algumas das características do ciclo de trabalho.

O sistema mecânico usado na robótica é constituído por cinco subsistemas que comunicam entre si através de interfaces que codificam e decodificam dados transmitidos entre os diversos subsistemas (Angeles, 2007). Os cinco subsistemas existente são:

- o subsistema mecânico que é constituído pela parte física ou corpo do robô,
- o subsistema sensorial,

- o subsistema de ação ou de estímulo,
- o subsistema de controlo e
- o subsistema de processamento de informação.

De acordo com a sua aplicação e estrutura física, os robôs podem ser classificados em manipuladores, braços robóticos, mãos robóticas, movimento, locomoção, nadadores, voadores (Angeles, 2007). Os robôs de locomoção, incluem os robôs com pernas e com rodas.

2.3. Estacionamento Automático de Veículos

Uma possível definição para veículos inteligentes é a capacidade de tratar informação de sensores, planear e executar movimentos num veículo, de uma forma semelhante à de um condutor humano (Xu, Chen & Xie, 2000). O estacionamento é um dos movimentos executados por um condutor humano e um dos desafios apresentado aos veículos inteligentes. Através de simuladores usando robôs e veículos, têm sido desenvolvidos diversos estudos em que se aplicam modelos matemáticos dinâmicos, redes neuronais e “fuzzy logic” (Camus, Coelho & Quadrado, 2002; Ryu, Kim & Oh, 2008; Szádeczky-Kardoss, Kiss & Wahl, 2008; Xu et al., 2000; Zhao & Collins, 2005).

Uma preocupação no estacionamento automático de veículos inteligentes, consiste em tentar que seja o mais semelhante possível ao estacionamento efetuado por um condutor humano e de uma forma segura (Camus et al., 2002; Szádeczky-Kardoss et al., 2008). Desse modo, primeiro o veículo deve aperceber-se do que o rodeia de forma a encontrar um lugar de estacionamento vazio e detetar obstáculos num parque de estacionamento. Sendo que a necessidade evitar colisões com outros obstáculos, para além dos veículos estacionados originaria combinações infinitas de trajetórias que veículo poderia seguir (Camus et al., 2002; Ryu et al., 2008; Szádeczky-Kardoss et al., 2008; Xu et al., 2000; Zhao & Collins, 2005).

Para encontrar o espaço vazio e evitar obstáculos são utilizados sensores que recolhem informação do ambiente envolvente ao veículo, como os sensores ultrassónico e o visual ou de imagem (Ryu et al., 2008; Szádeczky-Kardoss et al., 2008; Xu et al., 2000; Zhao & Collins, 2005). O sensor visual ou de imagem permite a deteção de um lugar de estacionamento através da análise das cores dos traços que

delimitam um lugar de estacionamento e a detecção de obstáculos através da alteração de cor no campo de visão do veículo (Xu et al., 2000). É com base nestes sensores que um veículo inteligente trata a informação recebida, planeia e executa os movimentos necessários.

Um dos métodos de investigação estudados é a determinação previa dos movimentos que o veículo devia executar (Camus et al., 2002; Xu et al., 2000). Este método segue dois passos: a planificação do caminho, que consiste do desenho do caminho a seguir desde a posição inicial até à posição final e a planificação da trajetória, que consiste no desenho de uma sequência de configurações necessárias (orientação, posição e velocidade) do veículo para que este siga o caminho especificado.

Outro método, é transferir as capacidades de um condutor humano para o veículo, através da utilização de redes neuronais e “fuzzy logic” (Ryu et al., 2008; Szádeczky-Kardoss et al., 2008; Zhao & Collins, 2005). Desta forma, os movimentos são gerados em tempo-real de acordo com o estado atual do veículo.

Em alguns estudos, de forma a determinar o movimento e trajetória do veículo, foram utilizados sensores do Sistema Anti-Bloqueio (ABS) e do Sistema de Posicionamento Global (GPS) (Szádeczky-Kardoss et al., 2008). Em alguns tipos de estacionamento, como no estacionamento paralelo, é necessário que o veículo execute manobras de inversão ou marcha-a-trás, de forma a compensar limites físicos do ambiente envolvente, no que diz respeito ao comprimento ou largura do veículo, à posição e dimensão dos obstáculos e dimensão do local de estacionamento (Camus et al., 2002; Ryu et al., 2008; Szádeczky-Kardoss et al., 2008; Zhao & Collins, 2005). Essas manobras podem ser avaliadas e executadas por tentativa e erro, utilizando os diversos sensores para obter dados do ambiente, nomeadamente distâncias e localização de obstáculos (Camus et al., 2002; Szádeczky-Kardoss et al., 2008) ou pode ser utilizada a inteligência artificial através de algoritmos genéticos (Ryu et al., 2008; Zhao & Collins, 2005).

3. Contexto Escolar

3.1. Caracterização da Escola

Fundada em 1965, a Escola Secundária com 3º ciclo Padre António Vieira (ESPAV) no ano letivo 2010/2011 sofreu uma intervenção de remodelação pela Parque Escolar e melhorias nos recursos digitais disponíveis (Conselho Geral, 2010; Parque Escolar, s.d.). Um dos objetivos foi repor a eficácia física e funcional, criando condições para a prática de um ensino moderno, adaptado aos conteúdos programáticos, às didáticas e às Tecnologias de Informação e Comunicação (TIC) (Parque Escolar, s.d.). De acordo com o Projeto Educativo (2010) da ESPAV, para os anos 2010 a 2013, com esta intervenção foi promovido o acesso e utilização das TIC, facilitando a divulgação e a comunicação de informação. Em termos de recursos informáticos, verificou-se que existem atualmente cinco salas de TIC (*Figura 1*), com 79 computadores e cinco quadros interativos. No total, em salas de aula e laboratórios, existiam um total de 130 computadores e 18 quadros interativos. As sala de aula e laboratórios que não possuíam quadro interativo, possuíam um projetor. Uma das salas de TIC foi criada devido a uma parceria entre a ESPAV e uma empresa de formação em informática, em que esta usufrui de uma sala para ações de formação, tendo em troca equipado a sala com computadores portáteis.



Figura 1. Sala TIC na ESPAV.

Para além dos recursos físicos, existe um conjunto de recursos digitais disponibilizados à comunidade e que se encontram referidos no Quadro 1, indicando-se o estado atual.

Quadro 1

Recursos Digitais Mencionados no Projeto Educativo de 2010 a 2013.

Descrição	Estado
Sítio oficial, com informações diversas sobre a escola.	Ativo e atualizado.
Plataforma <i>online</i> que permite consultar o saldo do cartão de aluno, ver os seus movimentos e comprar senhas de almoço.	Inativo.
Plataforma de <i>e-learning</i> com conteúdos de várias disciplinas e informações diversas.	Ativo e atualizado.
Contas pessoais de e-mail e espaço para armazenamento e partilha de ficheiros, para professores, assistentes e alunos.	Inativo mas todos os alunos e professores têm um endereço com um domínio próprio da escola.
Publicações periódicas com novidades e notícias da escola.	Última atualização em fevereiro de 2010.
Blogue que acompanha as atividades que decorrem na escola no âmbito dos Recursos Tecnológicos para o Ensino das Ciências (RTEC).	Ativo e atualizado.
Blogue que acompanha o projeto +Sucesso, do 7º ano.	Apenas teve atualizações nos meses de abril, maio e junho de 2010.
Blogue com informações e atividades da Mediateca.	Apenas com atualizações relativas ao segundo período.
Jornal <i>online</i> que acompanha as atividades que são desenvolvidas na escola.	Ativo com última atualização no mês de abril de 2011.

Relativamente ao acesso dos alunos às TIC é referido pelo Projeto Educativo (2010) que a maioria já possuíam computador próprio (83%) e acediam à Internet em casa (79%). No entanto, é de referir que as turmas dos cursos profissionais tinham um menor número de alunos com computadores pessoais (77%) e com Internet em casa (75%).

O Projeto Educativo (2010) menciona que em termos de idades, a maioria dos alunos encontrava-se nos escalões etários normalmente correspondentes ao ano de escolaridade que frequentam, excepto nas turmas de CEF e cursos profissionais. Este facto poderá evidenciar que os CEF e os cursos profissionais constituem uma via alternativa para os alunos que possuem um baixo aproveitamento escolar.

3.1.1. Ações educativas e pedagógicas.

Perante a realidade escolar, o Projeto Educativo (2010) de 2010 a 2013 apresenta como missão prestar um serviço público de qualidade num ambiente favorável à aprendizagem, de respeito mútuo e civismo, valorizando e otimizando os recursos humanos e físicos existentes, contribuindo para a formação de cidadãos livres, responsáveis, autónomos, críticos, criativos e intervenientes na construção de uma sociedade mais justa e solidária. No mesmo documento, é mencionado a necessidade de manter e promover os valores do respeito, da responsabilidade, da solidariedade, da disciplina e do trabalho. Em termos de valores, não possui nada de novo, sendo semelhante ao projeto educativo de outras organizações escolares e segue os valores gerais mencionados na Lei de Bases do Sistema Educativo. Como visão, a escola possui o lema “uma escola com todos e para todos”, devendo dar a oportunidade a todos e acolher as suas diferenças, independentemente da sua origem, permitindo e encorajando a partilha de experiências diversificadas.

De acordo com o definido pelo Projeto Educativo (2010) da ESPAV, a aprendizagem é considerada como um processo ativo que implica seleção, processamento e assimilação de informação, de acordo com a maturidade intelectual do indivíduo, dando-se ênfase à reflexão, abstração e à resolução de problemas, sendo o próprio aluno a construir o seu conhecimento. É então sugerida a realização de atividades interdisciplinares, trabalho de projetos, trabalho de pesquisa e investigação e trabalho colaborativo. A afirmação de que a aprendizagem não é apenas um processo de absorção passiva de informação mostra ser um passo para a ruptura de uma visão conservadora do ensino/aprendizagem.

Para os cursos profissionais são definidos objetivos específicos como a melhoria da taxa de conclusão, da pontualidade e assiduidade dos alunos e a implementação de medidas de combate ao abandono escolar. Como estratégias, para o cumprimento dos objetivos propostos, destacam-se as seguintes:

- promover o trabalho colaborativo entre docentes;
- implementar práticas de sala de aula adequadas e inovadoras;
- criar equipas educativas, com o perfil mais adequado às características das turmas, que acompanhem os alunos ao longo de cada ciclo;
- dedicar uma atenção especial ao diagnóstico das aprendizagens realizadas e das deficiências que ainda prevalecem, organizando reais oportunidades de recuperação para os alunos sinalizados;
- adequar a oferta formativa e modalidades de formação às necessidades dos alunos;
- solicitar aos alunos o preenchimento de uma ficha de registo de atraso, que o leve a refletir sobre as consequências desse atraso;
- entrevistar os alunos candidatos aos cursos profissionais, como forma de verificar a sua motivação para o curso ou aconselhar alternativas.

3.2. Caracterização da Disciplina

Os cursos profissionais são percursos que cumprem vários objetivos, sendo referidos pela Agência Nacional para a Qualificação (s.d.), os seguintes:

- contribuir para desenvolver competências pessoais e profissionais;
- preparar para o exercício de uma profissão;
- privilegiar as ofertas formativas que correspondem às necessidades de trabalho locais e regionais;
- preparar os alunos para o acesso a formações pós-secundárias ou ao ensino superior.

Criado pela Portaria nº. 913/2005, o curso profissional de Técnico de Informática de Gestão, equivalente ao nível secundário, está integrado na educação e formação de Ciências Informáticas (Ministério da Educação, 2005). O técnico de informática de gestão é o profissional qualificado que possui competências na gestão das organizações, nomeadamente, na construção de modelos de gestão de negócios/projetos, criando matrizes com recurso a aplicações informáticas para as micro, pequenas e médias empresas, com vista à eficácia de resultados. Para além disso, está apto a apoiar a coordenação de departamentos de informática e a proceder ao desenvolvimento, instalação e utilização de aplicações informáticas em qualquer

área funcional de uma organização/empresa (Agência Nacional para a Qualificação, s.d.).

Este curso profissional está repartido por três componentes de formação, apresentadas no Quadro 2. A cada uma das componentes, está associado um conjunto de disciplinas e um conjunto de horas (Agência Nacional para a Qualificação, s.d.).

Quadro 2

Componente de Formação do Curso Profissional de Técnico de Informática de Gestão

Componente de Formação	Horas
Componente de Formação Sociocultural	
Português	320
Língua Estrangeira I ou II	220
Área de Integração	220
Tecnologias de Informação e Comunicação	100
Educação Física	140
Componente de Formação Científica	
Matemática	300
Economia	200
Componente de Formação Técnica	
Linguagens de Programação	458
Organização de Empresas e Aplicações de Gestão	287
Sistemas de Informação	252
Aplicações Informáticas e Sistemas de Exploração	183
Formação em Contexto de Trabalho	420
Total de horas	3100

A disciplina onde será realizada a intervenção é Linguagens de Programação, pertencente à Componente de Formação Técnica do curso profissional de Técnico de Informática de Gestão. No total, devem ser lecionadas 458 horas em que estarão repartidas pelos três anos do curso e em blocos de 90 minutos. Segundo a Direcção-

Geral de Formação Vocacional (2005), esta disciplina tem como objetivo que o aluno adquira experiência e competências básicas na exploração e utilização de um conjunto alargado de ferramentas de programação. O aluno terá que adquirir a capacidade de analisar as linguagens de programação existentes, de compreender as técnicas básicas de implementação através das linguagens de programação, desenvolver a capacidade de aprender novas linguagens de programação e aumentar a capacidade de conceção e desenvolvimento de *software* (Direcção-Geral de Formação Vocacional, 2005). Tendo em conta esse objetivo, a disciplina encontra-se dividida em 17 módulos (Quadro 3).

Das competências a desenvolver pelos alunos, destacam-se as seguintes (Direcção-Geral de Formação Vocacional, 2005):

- efetuar a análise de sistemas de informação;
- conceber algoritmos através da divisão dos problemas em componentes;
- estimular o raciocínio lógico;
- saber gerir o tempo;
- estimular a reflexão, a observação e autonomia;
- respeitar os pontos de vistas dos outros;
- revelar espírito crítico e hábitos de tolerância e de cooperação;
- transformar a informação recolhida em conhecimento;
- demonstrar criatividade e abertura à inovação.

No programa da disciplina, são referenciadas algumas orientações e estratégias de ensino, que poderão ser adotadas pelos professores. Dessas orientações e estratégias, destaca-se as seguintes:

- introdução de uma linguagem estruturada, para que o aluno entenda a lógica da programação;
- abordagem de estruturas de controlo na linguagem leccionada, para que o aluno tenha a perceção de quando e como as deve utilizar;
- desenvolvimento de um projeto, em que serão utilizadas técnicas e conceitos da linguagem de programação leccionada;
- desenvolvimento de um projeto, enquadrando os alunos num ambiente criativo, aplicando os conhecimentos técnicos e científicos adquiridos ao longo dos três anos do ciclo de formação.

É ainda destacada a importância da interdisciplinaridade na realização do projeto, pois permite a interligação de conhecimentos, estimulando a capacidade de realização e inovação e inculcando o gosto pelo saber fazer. Para além disso, é considerada uma oportunidade para estabelecer uma ligação entre o aluno e a sociedade em geral, na resolução de problemas específicos que existam em empresas ou instituições.

Mencionada no programa da disciplina, outra metodologia e orientação de ensino é a necessidade de intercalar exposições teóricas com a resolução de exercícios práticos em computadores. Sempre que seja útil, deverá anteceder-se as exposições teóricas com a apresentação de situações problemáticas, que tornem possível aos alunos fazerem um enquadramento prático da situação, criando-lhes expectativas e motivações que os levem a participar na discussão do tema. Para além disso, os temas deverão ser introduzidos progressivamente, sempre contextualizados e ligando constantemente a teoria à sua aplicação. Dessa forma, deverá ser criado um ambiente de diálogo participado, em que os alunos vão sendo conduzidos na procura das soluções para os problemas apresentados, obtendo assim um maior interesse por parte dos alunos.

No programa da disciplina é mencionado que a resolução de exercícios deve ser de dificuldade progressiva, permitindo ao aluno o estudo dos diferentes temas de uma forma contínua. Em termos de avaliação dos conhecimentos adquiridos, deverão existir duas componentes obrigatórias: contínua e diagnóstica, através de observações pontuais feitas durante o desenvolvimento dos exercícios práticos e um teste sumativo.

Os critérios de avaliação de 2011/2012 definidos pelo grupo de professores de informática da escola para a disciplina de Linguagens de Programação, incidem nos seguintes domínios: competências, cognitivo, capacidades e atitudes (grupo 550, 2011). Para a avaliação dos domínios competências, cognitivos e capacidades, foram definidos os seguintes instrumentos: testes, trabalhos individuais ou de grupo, projetos, relatórios e apresentações orais. Relativamente avaliação do domínio atitudes, foi definido como instrumento a observação da conduta do aluno em sala de aula e o desempenho na realização das atividades propostas. Em termos de peso de cada domínio na avaliação final do aluno, os domínios competências e cognitivos possuem um maior peso, com 55%, o domínio capacidades tem um peso de 15% e o domínio das atitudes tem um peso de 30%.

Quadro 3

Módulos da Disciplina Linguagens de Programação

Número	Designação	Horas
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

3.3. Caracterização da Unidade Didática

O módulo 17 – Desenvolvimento de uma Aplicação de Gestão, tem como objetivo o desenvolvimento de um projeto pessoal, estruturante e centrado num ou vários temas e problemas perspetivados pelo aluno e em negociação com o professor. Nele devem ser investidos saberes e competências adquiridas em módulos já desenvolvidos nesta disciplina e, sempre que possível, ser um projeto

interdisciplinar, integrando saberes, capacidades e competências desenvolvidas ao longo do curso de formação. Desta forma, o projeto promove a consolidação de conhecimentos e competências, bem como uma preparação para a integração do aluno no mundo do trabalho.

Sendo assim, o módulo tem os seguintes objetivos:

- verificar a aplicação conjunta dos conceitos apreendidos;
- relacionar os conceitos apreendidos com os seus conhecimentos e/ou experiências pessoais;
- articular os conhecimentos adquiridos conjuntamente com novos conhecimentos, para fomentar um espírito crítico relativamente a formas de organização instituída;
- sensibilizar para a problemática da introdução dos sistemas de informação nas organizações.

Para os alunos atingirem esses objetivos, o conteúdo deste módulo é o desenvolvimento de uma solução informática completa, na área de gestão.

Como intervenientes no desenvolvimento deste módulo, para além do aluno, que é o principal responsável pelo projeto, do professor da disciplina, com o papel de acompanhamento e supervisão, podem existir outros professores, para prestar apoio técnico e específico, ao desenvolvimento dos projetos.

A avaliação deste módulo incidirá sobre:

- a conceção e o desenvolvimento do projeto;
- as experiências realizadas durante o desenvolvimento do projeto e a sua aplicação ou aplicabilidade prática;
- a complexidade do projeto;
- a atitude do aluno, em termos de motivação, empenho, criatividade e rigor na concretização das metas propostas, nomeadamente prazos;
- o grau de autonomia demonstrado pelo aluno;

a qualidade do relatório final e outra documentação entregue e a qualidade da defesa oral.

3.3.1. Conceitos envolvidos.

Os conceitos de Fluxogramas, Comentários, Variáveis, Operadores e Expressões Relacionais, Estruturas de Decisão e Repetição, Eventos e Instruções de Entrada e de Saída, são alguns dos conceitos envolvidos na programação e lecionados em disciplinas de programação, nomeadamente na disciplina de Linguagens de Programação (Direcção-Geral de Formação Vocacional, 2005).

Um fluxograma é uma representação visual, através de um diagrama e um conjunto de símbolos, que mostra as operações e a sua sequência de execução em sistemas de processamento de informação (IBM, 1970). Assim, permite ilustrar a sequência de operações a ser executada para obter a solução de um problema, sendo uma das fases iniciais na programação de soluções de problemas (EdrawSoft, 2012). Para além disso, a nível profissional, tem a vantagem de facilitar a comunicação dos programadores com os seus clientes ou com quem lhes pede uma determinada tarefa, ajudando a esclarecer quais os elementos principais do processo, como o sistema está atualmente a funcionar, como pode ser melhorado e documentar um programa complexo.

A utilização de comentários no código escrito, independentemente da linguagem de programação utilizada, é fundamental para a criação de documentação e compreensão dos processos envolvidos (Coelho, 2010). Estes comentários são anotações colocadas no código pelo programador e que são ignoradas pelo compilador. Desta forma, torna-se mais fácil entender o código escrito, o que será vantajoso, por exemplo, no caso de programação a pares ou na reutilização de código. A utilização de comentários para a criação de documentação pode ser realizada através da ferramenta *Javadoc* (Oracle, s.d.; Friendly, 1995). É então necessário que exista um conjunto de regras na escrita de comentários, que poderá depender da linguagem de programação e da ferramenta de conversão dos comentários em documentação. No caso da linguagem de programação Java os comentários são escritos em HTML e são constituídos por uma parte descritiva e outra com a descrição correspondente a um conjunto de etiquetas, por exemplo, *@param*, *@return* e *@see*. Os comentários devem ser colocados antes das declarações de classes, campos, construtores e métodos.

Uma variável poderá ser definida como uma entidade que possui um determinado valor num determinado instante, podendo existir quantas variáveis o

programador achar necessário (Coelho, 2010). O valor que a variável possui, pode ser alterado e utilizado ao longo do programa escrito. Assim, o conceito de variável é normalmente aplicado a linguagens de programação dinâmicas, não sendo aplicado, por exemplo, a linguagens de *Markup* como o HTML (W3C, 2012). Em algumas linguagens de programação, como a linguagem C, é necessário indicar o tipo de dados que a variável irá guardar e potencialmente o seu tamanho máximo, que corresponderá ao número de bytes que uma determinada variável ocupará na memória RAM do computador (Coelho, 2010).

Em linguagens de programação e na notação matemática são utilizados operadores, que são aplicados a um ou mais operandos numa expressão (Microsoft, s.d.). Sendo que uma expressão é uma sequência de um ou mais operandos e zero ou mais operadores. Assim, as expressões podem ter vários níveis de complexidade, podendo usar operadores que por sua vez, usem outras expressões como parâmetros ou chamadas de método. Em termos de operadores, estes podem ser de vários tipos de acordo com a linguagem de programação utilizada, traduzindo-se em vários tipos de expressões. Os tipos de operadores mais comuns são os operadores de atribuição, aritméticos, relacionais, lógicos e lógicos com bits (Coelho, 2010; Microsoft, s.d.).

As instruções podem ser programadas de forma a ter uma execução sequencial, mas pode-se pretender que exista uma execução alternativa. Uma das instruções, usada na maioria das linguagens de programação, para conseguir esse efeito é o *if* (Coelho, 2010; Microsoft, s.d.). De acordo com o resultado de uma expressão lógica, verdadeiro ou falso, é executado um bloco de instruções em detrimento de outro bloco de instruções. De realçar, que cada bloco de instruções pode possuir uma ou mais instruções. Estruturas de repetição são o equivalente a ciclos, que permitem a execução de um bloco de instruções zero ou mais vezes, de acordo com a expressão lógica utilizada (Coelho, 2010; Microsoft, s.d.). Sempre que a expressão lógica é verdadeira o bloco de instruções é executado, parando quando esta é falsa. No entanto, a expressão lógica pode ser verificada no início do ciclo ou no seu fim. Na maioria das linguagens de programação, as instruções mais comuns são o *while*, o *for* e o *foreach*.

Em programação, um evento é o resultado de uma ação com diferentes origens, nomeadamente, de *hardware*, como por exemplo a falta de memória, de *software*, como a ocorrência de um erro ou de interface, como o clicar de uma tecla do teclado (Burnett & Ambler, 1992; Keckler, Dally, Chang, Lee & Chatterjee,

1999; Samek, 2009). Os eventos permitem que uma classe ou objeto, chamados de *publisher*, notifique outras classes ou objetos, chamados de *subscribers*, quando algo ocorre (Microsoft, s.d.). Dessa forma, consegue-se que apenas uma parte do código seja executado, quando algo ocorre, interrompendo um estado de “hibernação” do programa ou a execução normal dos processos de um programa. Esta possibilidade, permitiu que os computadores possuíssem um sistema orientado a eventos, sendo que desta forma os computadores podem esperar continuamente a ocorrência de um evento interno ou externo.

Instruções de entrada e de saída são principalmente utilizadas quando se pretende receber ou enviar dados, através de instruções de acordo com a linguagem de programação utilizada (Microsoft, s.d.; Hennessy & Patterson, 2012). As Instruções de Entrada e de Saída são normalmente funções já existentes em bibliotecas de cada linguagem de programação. Esta é uma forma simples de se obter dados do utilizador, que podem ser manipulados ou guardados no programa e devolver dados resultantes da execução de um programa. Esses dados podem ser de vários tipos, como por exemplo, texto, imagens ou som.

3.4. Caracterização da Turma

No 1º ano do curso profissional de Técnico de Informática de Gestão, correspondente ao ano letivo 2009/2010, a turma era constituída por 25 alunos. No 2º período do 3º ano, do ano letivo 2011/2012, a turma era constituída por 14 alunos. Devido à falta de documentação escrita que permita saber os motivos da redução de alunos na turma, foi feita essa questão ao professor cooperante. Este informou que a maioria dos casos se deveu a anulações de matrícula, a transferências para outras escolas e a exclusões por não terem aprovação ou não comparecerem nas provas de recuperação para quem atinge o limite de faltas previsto no Despacho de 27 de Setembro de 2008 proferido pelo Secretário de Estado da Educação. Na disciplina de Linguagens de Programação, a turma encontrava-se dividida em dois turnos, com sete alunos cada um.

A turma no 2º período do ano letivo 2011/2012 era totalmente constituída por alunos do sexo masculino e com uma média de idade 18,5 anos. A média de idades da turma, poderá confirmar o afirmado no Projeto Educativo da ESPAV em que a faixa etária nos cursos profissionais é superior à faixa etária no ensino secundário.

Tendo como referência o final do ano de 2009, a média de idades no 12º ano do ensino secundário é de 17,6 anos (Conselho Geral, 2010).

Com o objetivo de identificar qual o interesse e/ou motivação dos alunos no curso, qual o interesse e uso que os alunos fazem das tecnologias, quais as possíveis dificuldades na informática e/ou programação e quais os hábitos de trabalho e de estudo dos alunos, foi aplicado um questionário anónimo a todos os alunos da turma (Anexo A). Os resultados mais significativos desse questionário são indicados a seguir.

Em relação ao curso, todos os alunos consideram que o gosto de usar e querer conhecer novas tecnologias e a utilidade do curso foram importantes ou muito importantes para se matricularem no curso profissional de Técnico de Informática de Gestão (Quadro 4). O gosto de jogar jogos de computador e pela informática foi considerado por 12 (86%) alunos como factores importantes ou muito importantes para a escolha do curso. Realça-se que apesar de o gosto pela informática ser referido como um fator importante na escolha do curso, nas fichas informativas preenchidas no ano letivo de 2009/2010, apenas três alunos referiram querer trabalhar na área da informática. De realçar que alguns alunos mencionaram que o gosto pela gestão e contabilidade (50%) e o gosto pela programação (42%) como os fatores nada ou pouco importantes para a escolha do curso.

Quadro 4

Respostas Obtidas Sobre os Motivos que Levaram os Alunos a Escolher o Curso.

Motivo	Nada importante		Pouco importante		Importante		Muito importante	
	nº	perc.	nº	perc.	nº	perc.	nº	perc.
Gosto pela Informática	0	0%	2	14%	7	50%	5	36%
Gosto em jogar em PCs e Consolas	1	7%	1	7%	5	36%	7	50%
Gosto em usar e conhecer novas tecnologias	0	0%	0	0%	9	64%	5	36%
Será útil no meu futuro	0	0%	0	0%	7	50%	7	50%
Gosto pela programação	3	21%	3	21%	5	36%	3	21%
Quero trabalhar em informática	1	7%	2	14%	7	50%	4	29%
Gosto por gestão e contabilidade	2	14%	5	36%	3	21%	4	29%

No que diz respeito aos fatores positivos do curso, 11 (79%) alunos mencionam gostar muito ou bastante da localização da escola e 10 (71%) alunos referiram o mesmo em relação aos colegas da turma (Quadro 5). O horário das aulas foi considerado como um fator negativo do curso, com 12 (85%) alunos referirem que não gostam ou gostam pouco. Os professores e os colegas da escola também foram referidos como fatores negativos, tendo 8 (57%) aluno referido que não gostam ou gostam pouco.

Quadro 5

Respostas Obtidas Sobre os Gostos dos Alunos em Relação ao Curso.

Gosto	Não Gosto		Gosto Pouco		Gosto Muito		Gosto Bastante	
	nº	perc.	nº	perc.	nº	perc.	nº	perc.
Da escola	0	0%	6	43%	5	36%	3	21%
Dos professores	0	0%	8	57%	5	36%	1	7%
Dos colegas da turma	0	0%	4	29%	8	57%	2	14%
Dos colegas da escola	0	0%	8	57%	3	21%	3	21%
Das disciplinas de Informática	1	7%	5	36%	6	43%	2	14%
Das disciplinas que não são de Informática	1	7%	6	43%	5	36%	2	14%
Localização da escola	2	14%	1	7%	7	50%	4	29%
Do horário das aulas	9	64%	3	21%	1	7%	1	7%

Em relação à satisfação dos alunos em relação às disciplinas do curso, 12 (86%) alunos indicaram estarem satisfeitos ou muito satisfeitos com as disciplinas Área de Integração, Educação Física e Aplicações Informáticas e Sistemas de Educação (Quadro 6). A disciplina de Matemática é a detêm uma maior insatisfação, tendo 9 (64%) indicado que estão nada ou pouco satisfeitos. No que diz respeito à disciplina de Linguagens de Programação, em que decorrerá a intervenção, 8 (57%) indicaram estarem satisfeitos ou muito satisfeitos com a disciplina.

Quadro 6

Respostas Obtidas Sobre os Níveis de Satisfação dos Alunos em Relação às Disciplinas do Curso.

Disciplina	Nada Satisfeito		Pouco Satisfeito		Satisfeito		Muito Satisfeito	
	nº	perc.	nº	perc.	nº	perc.	nº	perc.
Linguagem de Programação	1	7%	5	36%	5	36%	3	21%
Sistema de Informação	1	7%	6	43%	6	43%	1	7%
Organização de Empresas e Aplicações de Gestão	1	7%	3	21%	4	29%	6	43%
Aplicações Informáticas e Sistemas de Exploração	0	0%	2	14%	6	43%	6	43%
Economia	4	29%	2	14%	6	43%	2	14%
Matemática	4	29%	5	36%	5	36%	0	0%
Português	2	14%	3	21%	7	50%	2	14%
Língua Estrangeira (Inglês)	0	0%	5	36%	4	29%	5	36%
Área de Integração	0	0%	2	14%	5	36%	7	50%
Educação Física	1	7%	1	7%	3	21%	9	64%
Tecnologias de Informação e Comunicação	1	7%	4	29%	4	29%	5	36%

No uso de tecnologias, apenas um aluno mencionou não ter computador em casa, apesar de todos referirem ter acesso à Internet. Em comparação com os dados mencionados no Projeto Educativo da ESPAV referentes aos alunos do ensino secundário e dos cursos profissionais, os alunos desta turma apresentam uma percentagem maior. De salientar que todos os alunos da turma utilizam o computador todos os dias e o uso do telemóvel é frequente ou todos os dias por 12 (85%) alunos (Quadro 7). Os blogs e os robôs, semelhantes aos da Lego® Mindstorms® NXT 2.0, são os menos utilizados, tendo 11 (79%) alunos mencionado que nunca ou raramente

utilizaram. Apesar de 9 (64%) alunos mencionarem que nunca utilizaram os robôs da Lego® Mindstorms® NXT 2.0, foi referido pelo professor cooperante e pelos alunos da turma que já os utilizaram no ano letivo anterior.

Quadro 7

Respostas Obtidas Sobre a Utilização das TIC e de Plataformas.

TIC/Plataforma	Nunca		Raramente		Algumas vezes		Frequentemente		Todos os dias	
	nº	perc.	nº	perc.	nº	perc.	nº	perc.	nº	perc.
Televisão	0	0%	1	7%	2	14%	0	0%	11	79%
Telemóvel	0	0%	1	7%	1	7%	2	14%	10	71%
Computador	0	0%	0	0%	0	0%	0	0%	14	100%
Consolas de jogos	4	29%	3	21%	3	21%	2	14%	2	14%
Robô NXT da Lego	9	64%	2	14%	2	14%	0	0%	1	7%
Moodle	0	0%	2	14%	0	0%	8	57%	4	29%
Redes sociais, como o Facebook	1	7%	1	7%	1	7%	1	7%	10	71%
Email	0	0%	0	0%	4	29%	3	21%	7	50%
Blog	7	50%	4	29%	2	14%	0	0%	1	7%
Wiki	7	50%	2	14%	3	21%	1	7%	1	7%
Foruns	4	29%	5	36%	3	21%	0	0%	2	14%
Chat, como o Messenger	2	14%	1	7%	2	14%	2	14%	7	50%
Rádio/Leitor de MP3 ou MP4	2	14%	1	7%	2	14%	1	7%	8	57%

Os conhecimentos em linguagens de programação dos alunos são baixos, tendo a maioria dos alunos afirmado que não conhece ou conhece o básico das seguintes linguagens: Visual Basic, Pascal, C, C++, Java, SQL, JavaScript, HTML e PHP. As linguagens C e C++ foram indicadas por 13 (93%) alunos onde o seu conhecimento é nenhum ou básico. No entanto, 5 (35%) alunos consideram os seus conhecimentos de Visual Basic e Pascal como Bom e Muito Bom (Quadro 8). A linguagem Visual Basic é a linguagem de programação que se encontrava a ser utilizada antes da intervenção.

Quadro 8

Respostas Obtidas Sobre os Níveis de Conhecimento dos Alunos em Relação a Linguagens de Programação.

Linguagens	Nenhum		Básico		Bom		Muito Bom	
	nº	perc.	nº	perc.	nº	perc.	nº	perc.
Visual Basic	1	7%	8	57%	3	21%	2	14%
Pascal	1	7%	8	57%	3	21%	2	14%
C	12	86%	1	7%	0	0%	1	7%
C++	11	79%	2	14%	0	0%	1	7%
Java	6	43%	6	43%	1	7%	1	7%
SQL	3	21%	9	64%	1	7%	1	7%
JavaScript	3	21%	9	64%	1	7%	1	7%
HTML	1	7%	9	64%	3	21%	1	7%
PHP	1	7%	10	71%	2	14%	1	7%

Os alunos da turma ao serem confrontados com dificuldades nas aulas, 9 (64%) alunos mencionam que em primeiro lugar pedem ajuda ao professor e em segundo lugar os colegas (Quadro 9). Apenas um aluno referiu não pedir ajuda e apenas dois utilizam em primeiro lugar os manuais disponibilizados pelo professor.

Quadro 9

Respostas Obtidas Sobre a Atitude dos Alunos Quando Sentem Dificuldades Durante as Aulas.

Atitude	1º Lugar		2º Lugar		3º Lugar		4º Lugar		5º Lugar	
	nº	perc.	nº	perc.	nº	perc.	nº	perc.	nº	perc.
Peço ajuda ao professor	9	64%	2	14%	2	14%	0	0%	1	7%
Peço ajuda aos colegas	2	14%	9	64%	0	0%	1	7%	2	14%
Vou à procura na Internet	1	7%	3	21%	7	50%	2	14%	1	7%
Vou ver os manuais dados pelo professor	2	14%	0	0%	2	14%	5	36%	5	36%
Não peço ajuda	1	7%	0	0%	2	14%	3	21%	8	57%

Em relação ao modo como os alunos são avaliados, 12 (86%) alunos afirmaram preferirem serem avaliados pelas suas capacidade e 11 (79%) alunos pelas suas atitudes (Quadro 10). Os instrumentos de avaliação que 10 (71%) alunos preferem que sejam utilizados na sua avaliação são: as fichas formativas, os testes práticos e os trabalhos de grupo.

Quadro 10

Respostas Obtidas Sobre as Preferências dos Alunos em Relação à Sua Avaliação.

Tipo de Avaliação	Não Gosto		Gosto	
	nº	perc.	nº	perc.
Fichas formativas	4	29%	10	71%
Teste teórico	7	50%	7	50%
Teste prático	4	29%	10	71%
Teste teórico-prático	7	50%	7	50%
Trabalhos de grupo	4	29%	10	71%
Atitudes (assiduidade, pontualidade, autonomia, responsabilidade, ...)	3	21%	11	79%
Capacidades (destreza na utilização do equipamento e do Software, utilização do vocabulário técnico, resolução de problemas, ...)	2	14%	12	86%

4. Metodologia

A intervenção pedagógica teve uma duração total de cinco aulas de 90 minutos, no módulo 17 - Desenvolvimento de uma Aplicação de Gestão da disciplina de Linguagens de Programação. Em concordância com o professor cooperante que é simultaneamente professor da disciplina de Linguagens de Programação, a intervenção decorreu a meio do módulo. Portanto, quando se deu o início da intervenção, os alunos já tinham iniciado o módulo com o desenvolvimento de um projeto, a criação de uma aplicação de gestão. Assim, a intervenção tem como objetivo a consolidação e/ou aquisição de alguns conceitos de programação. Estabelece-se assim, uma relação com o objetivo principal do módulo, promover a consolidação de conhecimentos e competências (Direcção-Geral de Formação Vocacional, 2005). Como instrumento principal de ensino, foi utilizada a robótica educativa associada a um conjunto de princípios e estratégias pedagógicas. Os robôs utilizados são os da marca Lego® Mindstorms® NXT 2.0.

Os conceitos abordados durante a intervenção, foram seleccionados com base no número de aulas da intervenção, os conceitos abordados na disciplina e os que poderiam ser abordados usando os robôs da Lego (Direcção-Geral de Formação Vocacional, 2005; Kelly, 2007). Esse conceitos foram:

- Fluxogramas;
- Comentários;
- Variáveis;
- Operadores e Expressões Relacionais;
- Estruturas de Decisão e Repetição;
- Eventos;
- Instruções de Entrada e de Saída.

Como objetivos específicos que os alunos deveriam atingir no final da intervenção, foram definidos os seguintes:

- Reconhecer a aplicação dos conceitos em programas já criados;
- Saber aplicar os conceitos na programação;
- Saber criar programas que resolvam um problema.

Na disciplina de Linguagens de Programação, a turma encontrava-se dividida em dois turnos, com sete alunos em cada um. Para além da minha intervenção, a mesma turma iria participar, na mesma disciplina, numa outra intervenção elaborada

por uma colega do Mestrado em Ensino da Informática da Universidade de Lisboa. Assim, em concordância com o professor cooperante, a minha intervenção foi realizada com um dos turnos.

Antes de iniciar a intervenção, foi desenvolvido um plano de aula para cada uma das aulas, cinco no total (Anexo B). A elaboração dos planos de aula teve como objetivo ter consciência das atividades que se pretendia realizar, questionar os motivos para a sua realização e antecipar possíveis respostas dos alunos, ressaltando o caráter pedagógico que se pretendia com essas mesmas atividades (Gabini & Diniz, 2009; Serrazina, 2009). Assim, nesses planos de aula, foram indicados os objetivos, os conteúdos temáticos presentes na aula e as atividades a realizar. Para cada uma das atividades, especificou-se a duração, a metodologia, os instrumentos de avaliação dos alunos e os recursos a utilizar, quer por mim como pelos alunos. Servindo como orientação para a concretização das aulas, os planos de aula não foram utilizados de forma rígida, podendo ser alterados durante o decorrer das aulas ou após reflexão sobre cada aula.

Como forma de os alunos se habituarem à minha presença em sala de aula, assisti a cinco aulas da disciplina de Linguagens de Programação, lecionadas pelo professor cooperante. A minha presença em sala de aula com os alunos, também serviu para conhecermos mutuamente, através do apoio ao professor cooperante no esclarecimento de dúvidas aos alunos.

Para a realização da recolha de dados de cada aluno, foi feito um pedido de autorização à diretora do Conselho Pedagógico da ESPAV, ao diretor de turma, ao diretor de curso e ao encarregado de educação do aluno (Anexo C).

4.1. Princípios e Estratégias Pedagógicas

4.1.1. Robótica educativa.

Na aplicação da robótica educativa, os alunos reuniram-se em pequenos grupos de trabalho com dois e três alunos, onde teriam que desenvolver soluções para um conjunto de problemas. Durante esse momento, os grupos puderam consultar motores de pesquisa na Internet. Os robôs utilizados foram os da Lego® Mindstorms® NXT 2.0, que utilizam uma Linguagem de Programação Visual (LPV) com o nome NXT-G, baseada na manipulação de ícones. Desta forma, pretendia-se

que os alunos focassem nos conceitos de programação aplicados e não na sintaxe a utilizar, uma vez que as LPV são autoexplicativas, ao serem mais fáceis e naturais de ler (Kanakadoss, 2005; Mendes, 2002).

Cada grupo tinha à sua disposição um robô previamente construído, com dois motores associados a duas rodas na dianteira e uma roda rotativa na traseira. A cada grupo, coube a tarefa de adaptar e montar os necessários para cada solução. Deste modo, pretendia-se que os alunos se concentrassem nas soluções que deveriam programar nos robôs.

Como apoio à programação a realizar nos robôs da Lego, foi pedido aos alunos, antes de iniciarem a programação, que descrevessem a sequência de passos para a solução de cada problema. Deste modo, os grupos poderiam consultar estas descrições durante a programação dos robôs, auxiliando-os na reflexão e na construção da sequência lógica de programação e de algoritmos.

Durante a criação de um robô autônomo que permitisse a resolução de um problema, os grupos seguiram os seguintes passos (Sartatzemi, Dagdilelis & Kagani, 2008):

1. montagem de sensores no robô;
2. desenvolvimento da programação;
3. carregamento no robô do programa criado;
4. executar o programa através do robô.

No entanto, para que o robô pudesse manifestar com exatidão um determinado comportamento, é necessário que os alunos repitam os passos dois a quatro várias vezes. A repetição desses passos apenas ocorre se, como resultado do *feedback* do robô, na execução do passo quatro o grupo não obtém o comportamento resultante da programação desenvolvida. Assim, cada grupo de alunos tem a possibilidade de interagir com o concreto, através do robô, e com o abstrato, através do programa desenvolvido, tendo a oportunidade de observar o comportamento do robô consequente da programação (Páztor, Pap-Szigetu & Torok, 2010).

A necessidade de refletirem, alterarem a programação no robô e testarem, poderá funcionar como uma aprendizagem por reflexão e com o erro cometido (Pinheiro, Sarrico & Santiago, 2010; Santos L. , 2002). Para que essa aprendizagem possa ser feita, procurei orientar o aluno, não identificando ou corrigindo o erro, mas sim questionando ou apresentando caminhos de orientação da ação a desenvolver, que o levasse à identificação e correção do erro (Santos L. , 2002).

4.1.2. Aprendizagem baseada em problemas.

A resolução de problemas é uma das dificuldades identificadas nos alunos para a aprendizagem da programação (Coull & Duncan, 2011; Gomes, Henriques & Mendes, 2008; Sloane & Linn, 1988), no entanto um programador deve possuir essa capacidade (Saeli, Perrenet, Jochems & Zwaneveld, 2011). Deste modo, foi apresentado o seguinte problema aos alunos: quais os passos que um veículo deve executar para que este possa efetuar o estacionamento automático em três situações de estacionamento, em espinha, em paralelo e em linha? Os alunos, organizados em grupos de trabalho, procuraram solucionar o problema do estacionamento automático fazendo uso de robôs da Lego® Mindstorms® NXT 2.0 e de uma maquete com os três tipos de estacionamento (*Figura 2*).

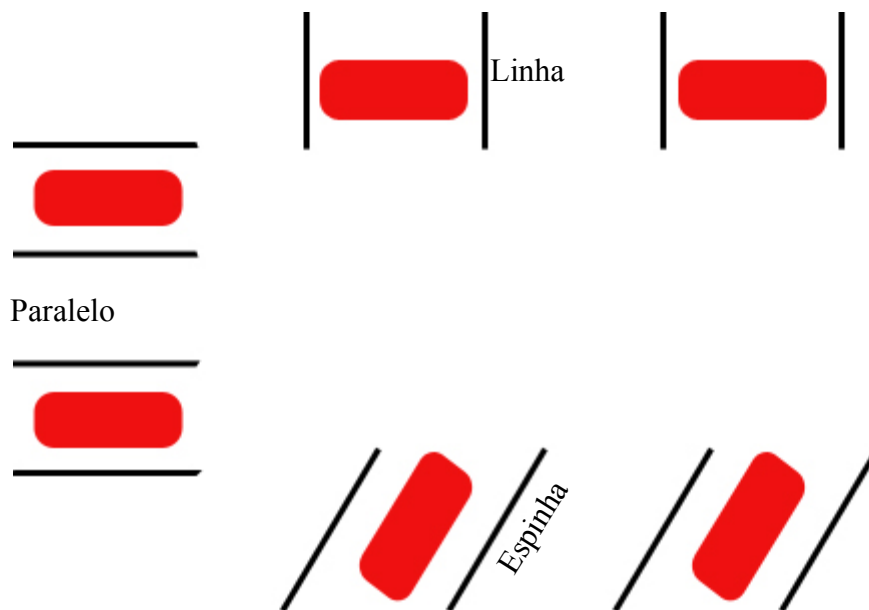


Figura 2. Exemplificação da maquete apresentada aos alunos, com os três tipos de estacionamento, espinha, paralelo e linha.

O problema proposto aproxima-se da vida real dos alunos, sendo considerado um factor importante na conceptualização do problema de forma a torná-lo relevante e realista para o aluno (Sockalingam et al., 2011). Esta aproximação deve-se ao facto de na turma existirem alunos com carta de condução ou a frequentar aulas de código/condução. Para além disso, procurou-se contextualizar o problema com

possíveis problemas que podem ser colocados aos alunos durante o estágio de formação profissional que vão realizar no início do 3º período, dependendo da área de negócio da instituição de acolhimento. Como reforço à contextualização do problema, foi exibido um vídeo em que é mostrado o estacionamento automático em paralelo de um veículo de um fabricante de automóveis, demonstrando o problema e parte da solução encontrada. Assim, permitiu aproximar o problema com problemas reais e que atualmente ainda se procura possíveis soluções. Para além disso, o vídeo juntamente com a existência de uma maquete com os tipos de estacionamento, permitiu aos alunos refletirem sobre o problema e possíveis soluções, através de uma discussão com todos os alunos.

Os três tipos de estacionamento, correspondem a três níveis de dificuldade, com base na complexidade requerida na programação da solução de cada um dos tipos. Deste modo, foi sugerido aos alunos que iniciassem pelo estacionamento em espinha, seguido do estacionamento em paralelo e terminassem com o estacionamento em linha, para que os exercícios fossem feitos de forma gradual. Assim, procurou-se criar um desafio aos alunos e aumentar a sua motivação na programação a desenvolver (Gomes et al., 2008).

Os alunos organizados em grupos de trabalho, procuraram solucionar o problema, passando por quatro fases, que foram identificados na aprendizagem baseada em problemas (Gomes et al., 2008; Kolmos, et al., 2007; Sockalingam, Rotgans & Schmidt, 2011):

1. identificar e definir o problema, em que os alunos confrontados com um cenário do mundo real, colocam questões;
2. aceder, avaliar e utilizar informação de diversas fontes e tipos;
3. construção de uma solução para o problema;
4. reflexão sobre a solução obtida.

Na primeira fase, após a apresentação do problema e dos recursos disponíveis aos alunos, foi realizado uma discussão entre todos os alunos. Nessa discussão os alunos colocaram questões e apresentaram sugestões de resolução do problema, no que diz respeito aos passos que o veículo deveria executar para cada um dos tipos de estacionamento. Essa discussão foi continuada, mas desta vez entre os elementos de cada grupo. Como forma a fomentar essa discussão, foi pedido que descrevessem os passos a executar pelo robô, que representa o veículo. A descrição das soluções foi

escrita em linguagem natural numa página na plataforma PBworks, criada por cada grupo.

A segunda e terceira fase, foram realizadas em conjunto. Os grupos iniciaram a programação do robô da Lego, utilizando um programa próprio para o efeito, e durante esse processo puderam realizar pesquisas na Internet sobre o problema colocado e possíveis soluções. Para além disso, puderam refletir e testar a programação realizada, utilizando a maquete com os três tipos de estacionamento.

Na quarta fase, os grupos realizaram uma demonstração das soluções encontradas. Durante a demonstração, ocorreu uma pequena discussão sobre as soluções encontradas, permitindo a reflexão por parte dos alunos.

Para além de apresentar o problema aos alunos, o professor assumiu o papel de orientador, dando *feedback* do trabalho desenvolvido e questionando ou apresentando sugestões de orientação ao trabalho a desenvolver. Durante as discussões de grupo ou entre todos os alunos, o professor assumiu também o papel de moderador, valorizando e dando oportunidades de participação aos alunos que o pretendessem fazer.

4.1.3. Princípios pedagógicos.

Os princípios pedagógicos adoptados serviram como guião para alcançar os objetivos da intervenção. Estes princípios foram definidos de forma que fossem suficientemente flexíveis, podendo assim serem adaptados para focarem aspetos particulares (Department of Education and Early Childhood Development, 2004). A utilização dos princípios proporciona uma visão mais clara das aprendizagens e ensino, constituindo um conjunto de linhas orientadoras das estratégias e ações do professor e servindo como base para avaliar a sua prática. Assim, foram adotados seis princípios pedagógicos principais (Department of Education and Early Childhood Development, 2004; Moreira, 2005):

- Autonomia,
- Desafio/Motivação,
- Democracia,
- Relevância,
- Reflexão e

- Responsabilidade.

Os princípios mencionados estão de acordo com os papéis dos intervenientes definidos para o módulo 17. Não existindo o apoio técnico e específico de outros professores durante a intervenção, temos como intervenientes: o aluno, que é o principal responsável pelo projeto, e o professor da intervenção, com o papel de acompanhamento e supervisão.

Na autonomia, pretendeu-se promover as atitudes e capacidades de autogestão das aprendizagens e possibilitar a negociação das decisões e dos papéis pedagógicos. Os grupos de trabalho tiveram autonomia de decisão sobre os passos a seguir no desenvolvimento do seu trabalho. O professor desempenhou um papel de orientador de uma forma recorrente e contínua, fazendo propostas, questões e dando *feedback* sobre as decisões técnicas de programação necessárias para obter uma solução, bem como sobre a metodologia de trabalho em grupo (Luz, 2011). Para além disso, procurei construir um conjunto diversificado de contextos facilitadores para o desenvolvimento da autoavaliação dos alunos, de modo a torná-los cada vez mais autónomos (Santos L. , 2002). Por exemplo, coloquei questões de uma forma continuada para que o aluno aprendesse a colocar-se essas questões de uma forma autónoma. As questões colocadas ocorreram oralmente na sala de aula, enquanto os alunos realizavam as tarefas propostas e por escrito, num questionário de autoavaliação.

Como desafio/motivação, usou-se estratégias para promover a confiança do aluno e o desejo de correr riscos na aplicação dos seus conhecimentos, através da discussão de ideias e de uma sequência de atividades que promovessem uma aprendizagem gradual. A existência de três tipos de estacionamento, representando três níveis de dificuldade, permitiu criar um desafio e competitividade entre grupos. Nas atividades propostas, procurou-se privilegiar o relacionamento de ideias, conhecimentos, imaginação e criatividade.

No que diz respeito ao princípio que se refere ao posicionamento democrático, procurou-se desenvolver comportamentos de cooperação e promover a liberdade de pensamento, a liberdade de expressão, o sentido de justiça e de respeito pelos outros. Em atividades de discussão de ideias e quando existia cooperação entre grupos de trabalho na troca de ideias e de propostas de soluções, procurei estar atento aos alunos que pretendiam participar e na qual, os colegas poderiam não estar a dar essa oportunidade. A oportunidade de participação desses alunos, deverá ser no final

da intervenção dos colegas, questionando se têm alguma sugestão ou algo a acrescentar à discussão. No caso de existirem conflitos entre alunos, procurei chamar à atenção para o comportamento demonstrado e questionar o motivo daquele comportamento. Independentemente do motivo, tentei que os alunos refletissem e chegassem a um acordo sobre o comportamento que cada um deveria ter para evitar futuros conflitos.

Quanto ao princípio da relevância, pretendeu-se relacionar as atividades com situações da realidade profissional de um técnico de Informática de Gestão. Dessa forma, procurou-se utilizar e valorizar as experiências, conhecimentos e capacidades dos alunos, integrando conhecimentos prévios, motivações, expectativas e interesses dos alunos, valorizando a sua experiência na utilização das tecnologias. Uma vez que a turma iria realizar um estágio no final do terceiro período, as atividades foram relacionadas com o que poderiam encontrar na instituição de acolhimento.

Em relação à reflexão, utilizei estratégias como a autoavaliação, que consiste num processo mental interno através do qual o próprio aluno toma consciência dos diferentes momentos e aspetos da sua atividade cognitiva (Santos L. , 2002). Essa autoavaliação foi realizada através de um questionário escrito e de questões orais colocadas por mim, durante o desenvolvimento das atividades. Para além disso, procurei dar *feedback* das aprendizagens e do trabalho desenvolvido, de forma continuada, para que o aluno aprendesse a colocar-se questões. Assim, procurei desenvolver no aluno a capacidade de autoquestionamento.

No que diz respeito à responsabilidade, procurei que os alunos se responsabilizassem pelas suas aprendizagens e do trabalho desenvolvido colaborativamente. O papel orientador do professor, dando de uma forma continuada *feedback* das tarefas realizadas e colocando questões, juntamente com a autonomia na realização das tarefas e a realização de momentos de reflexão, poderá permitir aos alunos responsabilizarem-se pelas suas aprendizagens. Assim, ocorreram momentos em que alguns alunos realizaram pesquisas e analisaram os resultados dessas pesquisas como forma de ultrapassar algumas dificuldades, experimentaram as soluções que programaram e corrigiram erros que detetaram nessas mesmas soluções, dividiram tarefas pelos elementos do grupo e geriram o tempo na realização dessas tarefas.

4.2. Recursos

Na sala de aula já existia um conjunto de recursos que foram utilizados como apoio à realização das tarefas dos alunos e do professor. A sala de aula possuía um videoprojetor, dez computadores com ligação à Internet e quatro *kits* de robôs Lego® Mindstorms® NXT 2.0. Dos computadores existentes, um deles era para uso do professor, mas com características iguais aos restantes. Nos computadores encontravam-se instalados e foram utilizados os programas Microsoft Visual Basic 6.0, Microsoft Word 2010 e diversos *browsers*, dando possibilidade de escolha entre o Windows Internet Explorer 9, Mozilla Firefox 11 e Google Chrome. Para a programação dos robôs Lego foi necessário instalar em todos os computadores o programa Lego® Mindstorms® NXT v2.0.

Na utilização da Internet, os alunos tiveram a liberdade de utilizar as ferramentas e consultar os sítios que acharam serem mais convenientes para a realização das diversas tarefas desenvolvidas em sala de aula. No entanto, foi criado e gerido por mim, um espaço de trabalho na plataforma PBworks (Anexo D) e uma disciplina na plataforma Moodle da escola (Anexo E) a serem utilizados pelos alunos e professor em sala de aula. Estas duas plataformas serviam de base à interação com os alunos, apoio à realização das tarefas e entrega dos trabalhos desenvolvidos. Na plataforma PBworks, foi ainda criada por mim uma página de apresentação dos objetivos da utilização da plataforma. Na plataforma Moodle, as diversas secções foram criadas por mim, onde foram incluídos os materiais em formato electrónico para a realização das tarefas pelos alunos. Para além destas plataformas, foi utilizada a ferramenta Prezi, onde foi construída uma apresentação a utilizar nas aulas (Anexo F) e a ferramenta YouTube para a exibição de um vídeo informativo sobre o estacionamento automático de veículos (Anexo G).

Outro recurso utilizado para a realização das tarefas em sala de aula, foi de uma maquete do parque de estacionamento com os três tipos de estacionamento, em paralelo, em espinha e em linha. Esta maquete possuía duas caixas que permitiam simular a existência de veículos estacionados ao lado do lugar vazio de estacionamento.

4.2.1. Lego® mindstorms® nxt 2.0.

A Lego® Mindstorms® NXT 2.0 é uma marca de robôs programáveis criados pela empresa Lego. Estes robôs são constituídos por diversos tipos de sensores, motores e um “tijolo” programável (The LEGO Group, 2012; Sartatzemi, Dagdilelis & Kagani, 2008). Em termos de classificação, os robôs da Lego são classificados como robôs inteligentes, pois podem ser programados de modo a utilizarem os sensores para obter dados do ambiente que os rodeia e decidir quais as ações a executar (Angeles, 2007; Rocha, 2006).

A componente principal é o “tijolo” programável, que consiste num computador em forma de tijolo, à qual se ligam os sensores, num máximo de quatro, e os motores, num máximo de três, através de cabos. Este computador possui um ecrã LCD, quatro botões e altifalantes que permitem a interação entre o utilizador e o computador.

Para o robô conseguir mover-se é necessária a utilização de motores (The LEGO Group, 2012). Estes motores são constituídos por um sensor de rotação que mede as rotações do motor em graus ou rotações completas, permitindo um maior controlo dos movimentos do robô, incluindo a velocidade. Cada rotação corresponde a 360°.

O robô recebe informação do meio ambiente através de vários sensores, nomeadamente de cor, toque, luz, som e ultrassónica (The LEGO Group, 2012). O sensor de cor permite ao robô distinguir e detetar seis cores e discriminar entre tonalidades de claro ou escuro, através da medição da intensidade de luz no ambiente e em superfícies coloridas. Esta distinção entre tonalidades claras e escuras também é realizada pelo sensor de luz, sendo que neste é a única função que desempenha. O sensor de toque funciona como um botão, em que o sensor deteta quando é pressionado por algo e quando é libertado. O robô consegue também detetar sons através do sensor de som. Este sensor consegue detetar decibéis (dB), incluindo os sons audíveis e não audíveis pelo ouvido humano, e decibéis ajustados (dBA), detetando apenas os sons audíveis pelo ouvido humano. O sensor ultrassónico, permite detetar objetos, podendo ser usado para evitar obstáculos, medir distâncias e detetar movimentos. A distância é medida em centímetros ou polegadas, com uma margem de erro de cerca de três centímetros. O seu funcionamento é semelhante aos radares, mede as distâncias calculando o tempo que uma onda de som demora a

atingir um objeto e a voltar. Este sensor funciona melhor em objetos grandes e superfícies lisas, tendo maior dificuldades em detetar objetos pequenos, finos ou arredondados.

4.2.2. Instrumentos de recolha de dados.

Como instrumentos de recolha de dados, foram utilizados os seguintes:

- câmara de filmar;
- gravador áudio;
- questionário de autodiagnóstico;
- questionário de autoavaliação;
- teste de diagnóstico;
- teste de avaliação;
- grelha de registos;
- páginas na plataforma PBworks;
- programa Lego® Mindstorms® NXT v2.0.

Os questionários de autodiagnóstico e de autoavaliação eram constituídos por dois grupos de perguntas (Anexo H). O primeiro grupo tinha como objetivo classificar os conhecimentos de cada aluno em relação a um conjunto de conceitos, nomeadamente: Fluxogramas, Comentários, Variáveis, Operadores e Expressões Relacionais, Estruturas de Decisão e Repetição, Eventos e Instruções de Entrada e de Saída. Esta classificação baseou-se em três níveis: 1 – não conheço o conceito; 2 - conheço o conceito, mas não sei aplicar na programação e 3 – conheço o conceito e sei aplicar na programação. No segundo grupo, foi pedido ao aluno que indicasse com que frequência ocorrem determinadas situações, no que diz respeito ao trabalho desenvolvido durante a programação, nomeadamente em relação ao código escrito, à reutilização e adaptação de código e as soluções programadas. São apresentadas seis situações, constituídas por três situações positivas e três situações negativas. A indicação é feita de acordo com quatro níveis: 1 – nunca; 2 – raramente; 3 – a maior parte das vezes e 4 – sempre.

Os testes de diagnóstico (Anexo I) e de avaliação (Anexo J) eram constituídos por um total de 20 questões, divididos por dois grupos. No primeiro grupo, constituído por 15 questões, foi apresentado um cenário, em que os alunos teriam

que realizar a programação necessária de forma a responder a cada uma das questões. A programação realizou-se no computador, usando o programa Microsoft Visual Basic 6.0 e a linguagem de programação Visual Basic 6.0, uma vez que esta foi a última linguagem de programação utilizada pelos alunos nas aulas da disciplina de Linguagens de Programação. No entanto, existiam duas questões em que os alunos tiveram que desenhar fluxogramas, podendo usar um programa à sua escolha, mas sendo sugerido o Microsoft Word, uma vez que os alunos estavam mais familiarizados com este programa. No segundo grupo, constituído por cinco questões, foi apresentado um código escrito na linguagem de programação Visual Basic 6.0 e os alunos tiveram que responder às questões num editor de texto à sua escolha.

A grelha de registos foi criada com base nos critérios de avaliação estabelecidos para a disciplina de Linguagens de Programação, para a avaliação do domínio das atitudes. No entanto, os indicadores nos critérios de avaliação que não poderiam ser verificados nas aulas da intervenção, não foram utilizados, como o *Cumprimento dos prazos estabelecidos para os projetos*. Noutros casos, os indicadores foram substituídos por outros como por exemplo, *Relações interpessoais* por *Respeita os direitos, opiniões e capacidades dos outros*. Assim, a grelha possuía quatro dimensões: Responsabilidade, Autonomia, Solidariedade e Comunicação/Expressão, totalizando 16 indicadores (Anexo K).

4.3. Avaliação da Intervenção

O objetivo principal da intervenção pedagógica consistiu em desenvolver a consolidação e/ou aquisição de alguns conceitos de programação pelos alunos. Em particular, procurou-se responder às seguintes questões:

- A robótica educativa, juntamente com um conjunto de estratégias pedagógicas, contribui para a consolidação de alguns conceitos de programação?
- A realização de um questionário seguido de um teste autoavaliação, promove a reflexão dos alunos sobre os seus conhecimentos reais?
- A descrição escrita dos passos a seguir na solução de um problema contribui positivamente para a programação dessa solução?

Para avaliar se os objetivos foram alcançados, realizou-se uma avaliação das aprendizagens dos alunos, a recolha vídeo e áudio das aulas, o preenchimento de grelhas de registos de cada aula e a recolha áudio de uma discussão de grupo.

As grelhas de registos de cada aula e os registos vídeo e áudio, permitiram avaliar os comportamentos demonstrados pelos alunos durante a intervenção, em relação a cada uma das atividades propostas. Os registos vídeo e áudio das aulas também permitiram analisar o comportamento do professor em sala de aula. Para além destas dimensões referentes aos registos de aula, tentou-se avaliar o interesse dos alunos nas atividades com os robôs da Lego® Mindstorms® NXT 2.0 nas aulas dois, três e quatro, através dos indicadores:

- está atento;
- realiza as tarefas propostas;
- mostra-se responsável pelo progresso do seu grupo de trabalho;
- participa espontaneamente;
- tenta ultrapassar as dificuldades;
- preocupa-se com a qualidade do trabalho.

A discussão de grupo foi realizada no dia seguinte à última aula da intervenção com a presença de todos os alunos. Como apoio à análise das respostas dos alunos, foi realizado um registo áudio da discussão. A discussão foi moderada por mim, seguindo um guião de questões (Anexo L) que serviram de orientação. Apesar de ser utilizado um guião, foram realizadas outras questões, de acordo com as respostas dadas pelos alunos durante a discussão. As questões colocadas focaram-se nos seguintes aspetos:

- realização dos questionários e dos testes como elementos de autoavaliação;
- trabalho desenvolvido em grupo;
- utilização da robótica educativa na aprendizagem da programação;
- ensino/aprendizagem da programação;
- comunicação entre professor e aluno.

4.3.1. Avaliação das aprendizagens.

A avaliação deve contribuir para as decisões que regulam o ensino e as aprendizagens (Santos L. , 2002). Deste modo, a intervenção foi iniciada com a realização de um questionário de autodiagnóstico e um teste de diagnóstico, com o objetivo de obter informação sobre os conhecimentos do aluno em relação aos conceitos que iriam ser abordados na intervenção e promover a reflexão do aluno sobre os seus conhecimentos e a regulação das suas aprendizagens futuras. Este tipo de avaliação, permite ao professor adotar estratégias que possibilitem a recuperação dos alunos com problemas de aprendizagem (Pimentel, França & Omar, 2003).

Com o objetivo de avaliar as aprendizagens realizadas pelos alunos durante a intervenção, aplicou-se um questionário de autoavaliação seguido de um teste de avaliação no final da intervenção, comparando os resultados obtidos com os resultados do questionário e do teste de diagnóstico. O questionário de autoavaliação e o teste de avaliação permitiram ainda a reflexão do aluno sobre os seus conhecimentos.

Os questionários eram iguais, de forma a comparar os resultados antes e depois da intervenção, sobre a perceção que cada aluno tinha dos seus conhecimentos. No entanto, os questionários possuíam nomes diferentes de forma a distinguir os dois momentos em que foram aplicados. O aluno deveria registar o seu grau de conhecimento e a frequência com que se depara com determinadas situações.

Os testes não eram iguais, apesar de serem semelhantes, variando na contextualização do problema e das questões colocadas. Estabeleceu-se um conjunto de problemas, com dois níveis de dificuldade, que o aluno deveria resolver individualmente com o uso do computador. Durante a realização dos testes, o aluno podia realizar pesquisas na Internet e em manuais disponibilizados pelo professor cooperante.

Na avaliação dos testes de diagnóstico (Anexo M) e de avaliação (Anexo N), não se pretendia atribuir uma avaliação de acordo com o número total de respostas certas ou erradas, mas sim com base no nível de conhecimento que se alcançou em cada conceito. De acordo com os objetivos de aprendizagem dos alunos, os testes eram constituídos por dois grupos de questões. No primeiro grupo, o aluno teria que aplicar conceitos de programação e no segundo grupo, teria que reconhecer alguns conceitos num excerto de código escrito na linguagem de programação Visual Basic

6.0. Deste modo, no primeiro grupo existem dois níveis de avaliação: N1, sabe aplicar o conceito na sua programação quando é pedido diretamente para o fazer; N2, sabe aplicar o conceito na sua programação, sem ser pedido diretamente para o fazer. O segundo grupo permite verificar se o aluno reconhece a aplicação de conceitos. Se o aluno soube aplicar um conceito no primeiro grupo e reconheceu-o no segundo grupo, considerou-se que ele conhecia o conceito. Se não respondeu às questões nos dois grupos, considerou-se que não conhecia o conceito.

De forma a relacionar as respostas dadas pelos alunos no primeiro grupo dos questionários e nos testes, foi criada uma relação entre o nível de conhecimento que o aluno afirma possuir sobre um determinado conceito e o nível de conhecimento que manifestou no teste (Anexo O).

Um objetivo de aprendizagem dos alunos é saber criar programas que resolvam um problema. Para determinar se este objetivo foi atingido, os grupos de trabalho foram avaliados com base nas páginas criadas na plataforma PBworks, em que descreveram em linguagem natural as soluções encontradas para o problema do estacionamento automático, indicando os passos que o robô da Lego deveria executar. Deste modo, para cada tipo de estacionamento, foi verificado se o grupo de trabalho descreveu uma solução ou não e se para cada solução descrita, esta tinha uma sequência de passos desde a posição inicial do robô até à posição final.

Em relação às soluções programadas na linguagem NXT-G, os grupos de trabalho foram avaliados pela resolução do problema do estacionamento automático de veículos. Assim, para cada tipo de estacionamento, os alunos foram avaliados positivamente se as suas soluções resolviam o problema do estacionamento de veículos. Com este método de avaliação das soluções, procurou-se determinar se o aluno sabia criar programas para resolver um problema, tendo em conta que existem muitas maneiras, mais ou menos eficientes, de se chegar a um algoritmo que resolva um problema (Pimentel, França & Omar, 2003).

5. Concretização das aulas

Em relação aos planos de aula, ocorreu uma alteração em relação ao plano da segunda aula, em que estava previsto que fosse realizado um exercício prático como introdução ao programa Lego® Mindstorms® NXT v2.0, à linguagem de programação NXT-G e aos passos necessários para realizar testes de execução das soluções, nos robôs da Lego. No entanto, uma vez que os alunos já tinham algum conhecimento sobre os robôs e do programa para a programação das soluções, devido a uma experiência anterior, optei por deixar os alunos explorarem o programa. Assim, os alunos iniciaram de imediato a programação das soluções encontradas. Explicações sobre o funcionamento do programa foi dado individualmente a cada grupo, quando era solicitado e quando verificava que os alunos estariam com dificuldades.

Na primeira e quinta aula, uma vez que o que se pretendia que os alunos conseguissem estabelecer uma relação entre as respostas no questionário e o que posteriormente responderam no teste, não existiu uma duração fixa para responder ao questionário e ao teste como se encontrava mencionado no plano de aula. Os tempos referidos no plano de aula serviram de referência ao professor, tendo-se verificado que os alunos demoraram cerca de 60 minutos a responder ao questionário e ao teste.

5.1. Primeira Aula

Na primeira aula, todos os alunos estiveram presentes e chegaram à sala de aula com pontualidade.

Com todos os alunos sentados à mesa no centro da sala, foi feita uma apresentação das aulas, indicando-se os objetivos gerais e específicos de cada aula e como alcançá-los. De seguida, foi permitido aos alunos realizarem questões sobre a intervenção. Após alguns esclarecimentos, cada aluno ocupou um computador, acedeu à plataforma Moodle da escola, onde foi criada uma disciplina com o nome da intervenção e acedeu a essa disciplina.

Depois de todos os alunos terem acedido à disciplina, indiquei que se encontrava disponível a apresentação das aulas, presente e futuras, um questionário de autodiagnóstico e um teste de diagnóstico. De seguida, foi pedido que cada aluno preenchesse o questionário e realizasse o teste de diagnóstico, com permissão para

efetuar pesquisas na Internet e utilizar materiais das aulas. Foi ainda indicado que o teste de diagnóstico seria realizado em Visual Basic, uma linguagem de programação utilizada nas aulas da disciplina com o professor cooperante. Reforçou-se os objetivos da realização do questionário e do teste de diagnóstico, para que os alunos não os associassem a momentos de avaliação com significativo peso na nota final do módulo. De destacar que os alunos questionaram se as aulas da intervenção contavam para a nota final e se o teste de diagnóstico contava para a avaliação. Devido à disposição dos computadores na sala de aula, os alunos encontravam-se próximos uns dos outros, o que originou várias conversas entre eles, tendo sido necessário relembrar-lhes os objetivos desta tarefa individual. Durante a realização do questionário e do teste, foi necessário transmitir algumas orientações e esclarecer dúvidas dos alunos relativamente à interpretação das perguntas.

No final do teste de diagnóstico, os alunos submeteram os ficheiros criados para a plataforma Moodle, como era habitual fazerem nas aulas com o professor cooperante. No entanto, verificou-se que quatro alunos não submeteram todos os ficheiros, tendo sido informados desse facto antes do início da segunda aula. Quando os alunos foram à procura dos ficheiros em falta, de forma a poderem entregar, não os conseguiram localizar.

5.2. Segunda Aula

No início da segunda aula, voltou-se a verificar a pontualidade e a presença de todos os alunos.

Com todos os alunos sentados à mesa no centro da sala, fez-se uma revisão dos objetivos da aula anterior, apresentou-se os objetivos da presente aula e os meios para os atingir. Para este efeito, utilizou-se a mesma apresentação da aula anterior, dando um sentido de continuidade e ligação entre as diversas aulas.

De seguida, foi apresentado o problema do estacionamento automático de automóveis, contextualizando-o como um possível problema que podem encontrar em empresas. Para esse efeito, foi exibido um vídeo real de um fabricante de automóveis, que demonstra o estacionamento automático num estacionamento paralelo e parte da solução encontrada através da indicação dos sensores utilizados na deteção de um lugar vazio. Depois de os alunos levantarem algumas questões e realizarem um debate sobre o vídeo observado e os possíveis passos que um veículo

deve executar para estacionar em paralelo, foi mostrado um esquema dos três tipos de estacionamento e pedido aos alunos para programarem as soluções para o estacionamento automático dos robôs da Lego® Mindstorms® NXT 2.0. Foi então colocado à disposição dos alunos um papel de cenário com os três tipos de estacionamento e um robô. Os alunos foram incentivados a levantarem questões e a refletirem sobre o problema, ocorrendo uma discussão entre todos os alunos da turma, em que puderam exemplificar e experimentar possíveis soluções utilizando o robô e o cenário.

A visualização do vídeo e a discussão em grupo foi importante para os alunos refletirem e indicarem de que forma o condutor sabe que existe um lugar vazio, como sabe que colidiu com outro veículo estacionado e como sabe que está dentro dos limites do lugar de estacionamento. Desta forma, os alunos tiveram uma melhor percepção de quais os sensores que poderiam utilizar

Após essa reflexão, os alunos juntaram-se em grupos. Inicialmente, deveriam estar formados três grupos de dois alunos e um dos alunos deveria trabalhar individualmente, como acontecia na maioria das aulas da disciplina. No entanto, a pedido do aluno e com concordância desse grupo, este juntou-se a um dos grupos já formados.

Trabalhando em grupo, foi-lhes atribuída a tarefa de descreverem e colocarem na plataforma PBworks uma solução para cada tipo de estacionamento. Para esse efeito, foi distribuído os dados de acesso a cada aluno, estando cada um deles previamente registado, pelo professor. Foi-lhes ainda mostrado como criar e editar uma página. Durante a realização desta tarefa, cada grupo teve à sua disposição um cenário e um robô para poder experimentar e refletir sobre os passos que o robô necessitaria de executar para cada solução.

Durante esta tarefa existiu interação e cooperação entre grupos, com trocas de ideias e reflexão conjunta. Como professor, fui colocando algumas questões de orientação sobre as soluções encontradas por cada grupo, com o objetivo de levar os alunos a refletir sobre essas soluções e sobre a existência de possíveis alternativas mais adequadas. Todos os grupos criaram uma página na plataforma PBworks, com as soluções encontradas.

5.3. Terceira Aula

Dada a ausência de dois alunos do mesmo grupo, a terceira aula foi iniciada com dois grupos. Apesar da falta de assiduidade, os presentes foram pontuais.

Como forma de inculcar um maior dinamismo, a aula foi iniciada com a programação dos robôs de acordo com as soluções escritas por cada grupo. Os grupos começaram por aceder à plataforma PBworks. Com base nas soluções escritas na aula anterior e utilizando o cenário e um robô, os grupos iniciaram a programação dos robôs. Os grupos efetuaram algumas alterações na estrutura física dos robôs, colocando os sensores que seriam necessários para que o robô realiza-se o estacionamento de acordo com as soluções escritas.

Durante esta tarefa, como orientação, insisti que os grupos seguissem os passos descritos nas soluções que apresentaram e que os alterassem, caso verificassem que existiria uma solução melhor. No entanto, verificou-se que um dos grupos não seguia os passos descritos. Esse mesmo grupo ocupou mais de metade da aula a tentar alterar o robô com a colocação de novos sensores que não estavam descritos nas soluções apresentadas. De forma a que o grupo refletisse sobre a metodologia de trabalho, uma vez que corria o risco de não conseguir programar as soluções, questionei várias vezes o que estavam a fazer e porquê, bem como os motivos para os outros elementos do grupo não estarem a adiantar o trabalho. Inicialmente, o grupo manteve o mesmo método de trabalho, mas perto do fim da aula iniciaram a programação das soluções com algum sucesso, devido à experiência anterior de um dos alunos com a programação dos robôs Lego.

O outro grupo não funcionou como grupo devido à falta de cooperação entre os elementos. Apenas um aluno trabalhava para a tarefa, enquanto o outro ia falar com o outro grupo e ver o que estavam a fazer. Foi então necessário dar atenção a este grupo, devido à falta de cooperação entre os elementos e às dificuldades que sentiam na programação e na utilização do sensor ultrassónico. Sempre que me dirigia ao elemento que se encontrava a trabalhar, perguntava o que o colega estaria a fazer, chamava-o e perguntava sobre o trabalho que estaria a desenvolver. Também colocava questões, de forma a incentivar a sua participação e atribuía tarefas a esse elemento. Verificou-se que os elementos passaram a estar mais tempo juntos e quando um deles se afastava, era chamado à atenção pelo outro de forma a realizarem as tarefas necessárias.

Continuou a existir colaboração entre os grupos como na aula anterior, mas alguns alunos mostraram não valorizar a opinião dos outros.

Todos os grupos começaram por tentar implementar a solução para o estacionamento em espinha. Devido à experiência anterior dos alunos na programação dos robôs, foi apenas necessário recordar algumas funcionalidades. Durante a realização desta tarefa, foi dado *feedback* aos grupos, sobre a forma como estavam a trabalhar em grupo e como estava a progredir o seu trabalho. Foram também feitas algumas questões sobre alternativas de implementação e de soluções, como forma a suscitar a reflexão dos alunos.

5.4. Quarta Aula

Na quarta aula, verificou-se a ausência de três alunos de grupos diferentes. Os restantes, que estiveram presentes, foram pontuais.

Assim, em dois grupos, as tarefas foram realizadas individualmente, o que me obrigou a dar mais atenção a estes alunos, apoiando-os na programação dos robôs com propostas e orientações mais específicas. No grupo que possuía três alunos, as tarefas foram desenvolvidas por dois deles.

Como aconteceu na aula anterior, iniciou-se com a programação dos robôs. Mais uma vez, verificou-se a colaboração entre os grupos e nesta aula não ocorreram conflitos entre alunos. Essa colaboração foi muito vantajosa, tendo dois dos grupos conseguido implementar as soluções para os três tipos de estacionamento. Apenas um grupo não o conseguiu, uma vez que não esteve presente na aula anterior e apenas um dos elementos desse grupo esteve presente nesta aula.

No final da aula, os grupos colocaram os ficheiros com as soluções implementadas nas páginas criadas na plataforma PBworks. No entanto, um dos grupos não colocou os ficheiros na plataforma e foi pedido que o fizesse. O grupo não encontrou os ficheiros com as soluções programadas, pelo que os ficheiros submetidos foram os existentes no robô que o grupo utilizou. Contudo, estes ficheiros não permitiam visualizar a programação feita através do programa Lego® Mindstorms® NXT v2.0.

5.5. Quinta Aula

Na quinta e última aula da intervenção, todos os alunos estiveram presentes e foram pontuais.

A aula foi iniciada com a demonstração das soluções programadas para o estacionamento automático do robô. Para isso, cada grupo utilizou um cenário e um robô da Lego® Mindstorms® NXT 2.0. Dois dos grupos implementaram as três soluções com os movimentos necessários, mas ainda era necessário realizar alguns ajustes nos valores correspondentes às distâncias e rotação das rodas. Um dos grupos conseguiu implementar duas das soluções, o estacionamento em paralelo e em espinha. Durante as demonstrações, alguns alunos de outros grupos participaram propondo soluções alternativas e correções aos colegas.

Após as demonstrações, cada aluno aceitou a disciplina da intervenção na plataforma Moodle, onde respondeu a um questionário de autoavaliação e a um teste de avaliação. Como aconteceu na primeira aula, devido à proximidade dos computadores, os alunos encontravam-se muito próximos uns dos outros, o que originou várias conversas durante a realização desta tarefa individual.

Existiu orientação da minha parte, no esclarecimento de dúvidas em termos de interpretação daquilo que era pedido. Verificou-se uma postura diferente de alguns alunos em relação à primeira aula em que também foi realizado um questionário e um teste. Alguns alunos estavam mais distraídos, por exemplo, ouvindo música, iniciando conversas com os colegas do lado e fingindo que faziam o questionário e/ou teste. Sendo assim, fui chamando a atenção a esses alunos de forma a concentrarem-se no que estavam a fazer e tentando acompanhar o desenvolvimento das tarefas. A atitude de alguns deles mudou, mas tal não aconteceu com outros.

No final do teste de avaliação, os alunos submeteram os ficheiros criados para a plataforma Moodle, como fizeram na primeira aula e nas aulas com o professor cooperante. No entanto, apesar de ter chamado a atenção dos alunos para submeterem todos os ficheiros que tinham criado, quatro alunos não o fizeram.

6. Resultados

6.1. Soluções Programadas

Dois dos grupos de trabalho programaram os robôs de forma a realizarem o estacionamento automático nos três tipos de estacionamento, em espinha, em paralelo e em linha. O terceiro grupo, conseguiu programar para dois tipos de estacionamento, em espinha e em paralelo. No entanto, já tinham iniciado parte da programação da solução para o estacionamento em linha.

Uma vez que os robôs entregues aos alunos já se encontravam construídos, os alunos focaram-se nos sensores que seriam necessários montar nos robôs para programarem as soluções que encontraram e descreveram nas páginas criadas na plataforma PBworks. Apesar de estarem disponíveis vários tipos de sensores, todos os grupos nas soluções programadas, só utilizaram um sensor ultrassónico para detetar um lugar vazio.

Nos tipos de estacionamento em espinha e em paralelo as soluções programadas foram semelhantes, seguindo a mesma sequência de passos, variando nos valores das variáveis velocidade de rotação das rodas, sentido de deslocamento das rodas, ângulo da direção das rodas, número de rotações das rodas e a distância entre o robô e um obstáculo. Assim, os passos seguidos pelo robô foram:

1. andar em frente até detetar um lugar vazio através do sensor ultrassónico;
2. quando deteta um lugar vazio o robô para;
3. andar uma determinada distância em frente no sentido do lugar vazio através do ajustamento do valor do ângulo da direção das rodas.

No tipo de estacionamento em linha os dois grupos as soluções programadas executam passos diferentes. Num dos grupos, os passos seguidos pelo robô foram:

1. andar em frente até detetar um lugar vazio através do sensor ultrassónico;
2. quando deteta um lugar vazio o robô para;
3. andar em frente até detetar um lugar ocupado através do sensor ultrassónico;
4. andar uma determinada distância em marcha-a-trás e para;
5. andar uma determinada distância para trás no sentido do lugar vazio através do ajustamento do valor do ângulo da direção das rodas e para;

6. andar uma determinada distância em frente de forma a ajustar o robô ao lugar através da alteração do valor do ângulo da direção das rodas.

No outro grupo, os passos seguidos pelo robô foram:

1. andar em frente até detetar um lugar vazio através do sensor ultrassónico;
2. quando deteta um lugar vazio o robô para;
3. andar uma determinada distância em frente no sentido do lugar vazio através do ajustamento do valor do ângulo da direção das rodas;
4. andar uma determinada distância em marcha-a-trás de forma a ajustar o robô ao lugar através da alteração do valor do ângulo da direção das rodas.

6.2. Páginas Criadas na Plataforma PBworks

Cada grupo criou uma página plataforma PBworks, com uma descrição da solução encontrada para cada tipo de estacionamento. As páginas descreviam, em linguagem natural, a sequência de passos que o robô deveria executar nas soluções a programar, fazendo referência aos sensores a utilizar.

Essas páginas foram criadas na segunda aula e foi pedido que realizassem as alterações necessárias se, no decorrer da intervenção, verificassem que a solução inicialmente encontrada tinha sofrido alterações. No entanto, dois dos grupos não o fizeram, apesar de terem existido alterações em todas as soluções que inicialmente descreveram. O outro grupo, alterou a sua página no decorrer da intervenção, mas apesar de ir reforçar o pedido, o grupo deixou-o de o fazer. Aparentemente, deveu-se ao facto de o grupo ter-se apercebido que não teriam tempo para programar toda a solução dentro do número de aulas estabelecidos para essa atividade. Essa falta de tempo deveu-se a algumas dificuldades que o grupo sentiu em programar alguns passos das suas soluções e na quarta aula, um dos elementos esteve ausente.

Dois dos grupos consultaram as páginas para montar os sensores que seriam necessários e para programarem os robôs. No entanto, como não efetuaram alterações nas soluções descritas nas páginas, programaram os robôs com passos que não tinham sido inicialmente descritos, o que levou os grupos a deixarem de consultar as descrições na quarta aula.

Um dos grupos nunca consultou as descrições, apesar de ter reforçado a importância de o fazerem. Quando o grupo sentiu dificuldades na programação da

sequência de passos, não sabendo quais os passos que deveriam programar a seguir, questionei o grupo se a descrição realizada não indicava a sequência que devia ser programada. No entanto, o grupo comunicou-me que não era necessário e que se lembravam dos passos que o robô deveria executar. Por vezes, o grupo indicava-me que a solução que estavam a programar tinha passos diferentes do que foi descrito e como não alteraram essa descrição, não era útil consultá-la.

6.3. Questionários de Autodiagnóstico e de Autoavaliação

No questionário de autodiagnóstico, em relação ao primeiro grupo (*Figura 3*), mais de metade dos alunos indicaram que não sabem aplicar os conceitos Fluxogramas, Comentários, Operadores e Expressões Relacionais, Estruturas de Decisão e Repetição, Eventos e Instruções de Entrada e de Saída. Apesar disso, responderam que conhecem os conceitos Comentários, Estruturas de Decisão e Repetição, Eventos e Instruções de Entrada e Saída. A maioria dos alunos referiu conhecer e saber aplicar o conceito de Variáveis.

Comparando as respostas dos alunos no primeiro grupo do questionário de autoavaliação (Quadro 11) com as respostas no questionário de autodiagnóstico, verifica-se que mais alunos indicaram que conhecem os conceitos (94%) e que sabem aplicá-los (37%).

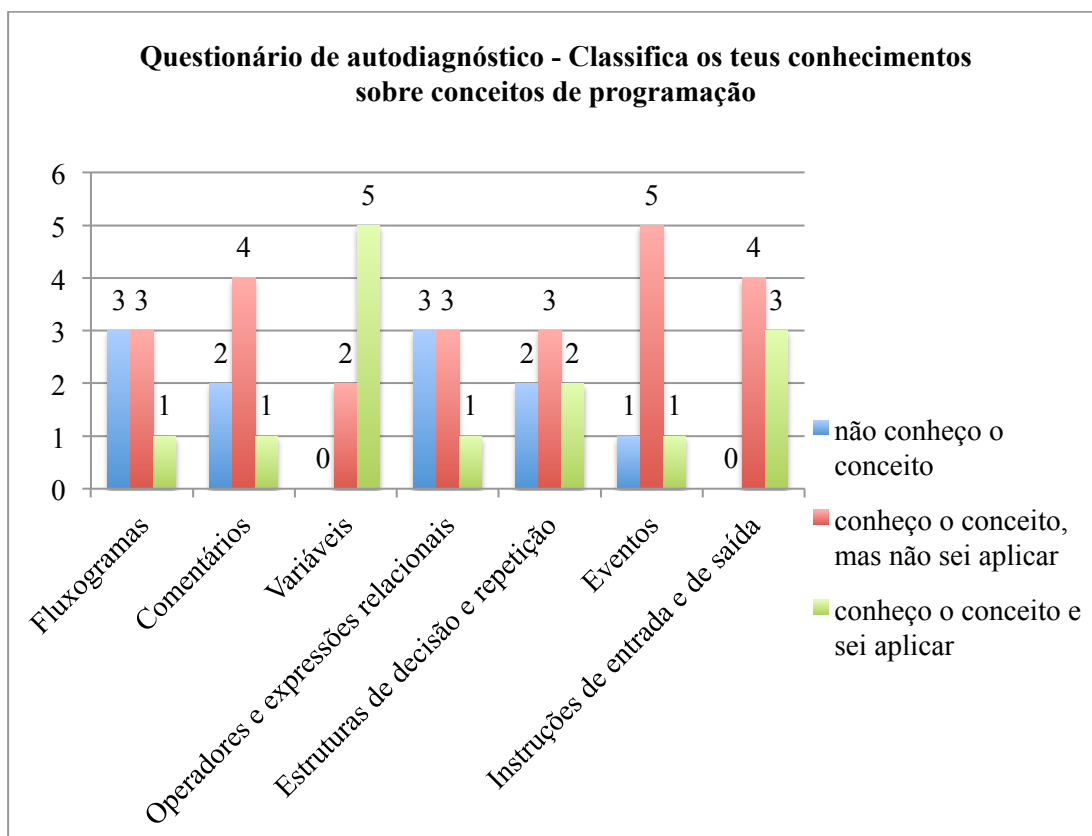


Figura 3. Respostas sobre o nível de conhecimento para cada conceito indicado no questionário de autodiagnóstico.

Quadro 11

Comparação entre Questionários, Indicando-se a Percentagem de Respostas para cada Nível de Conhecimento.

	Não conheço o conceito	Conheço o conceito, mas não sei aplicar	Conheço o conceito e sei aplicar
Autodiagnóstico	22%	49%	29%
Autoavaliação	6%	57%	37%

No questionário de autoavaliação (Figura 4), mais de metade dos alunos, indicado que conhecem e sabem aplicar o conceito de Comentário e Variáveis. Em comparação com o questionário de autodiagnóstico, obteve-se mais respostas a indicar que conhecem e sabem aplicar os seguintes conceitos:

- Comentários;
- Operadores e Expressões Relacionais;

- Estruturas de Decisão e Repetição;
- Eventos.

No entanto, menos alunos indicaram que conhecem e sabem aplicar os conceitos de Fluxogramas e de Instruções de Entrada e Saída. Tendo a totalidade dos alunos indicado conhecer o conceito de Fluxogramas, apesar de não o saber aplicar.

Ainda comparando os resultados dos dois questionários, autodiagnóstico e autoavaliação, apenas o conceito de Variáveis não sofreu alterações.

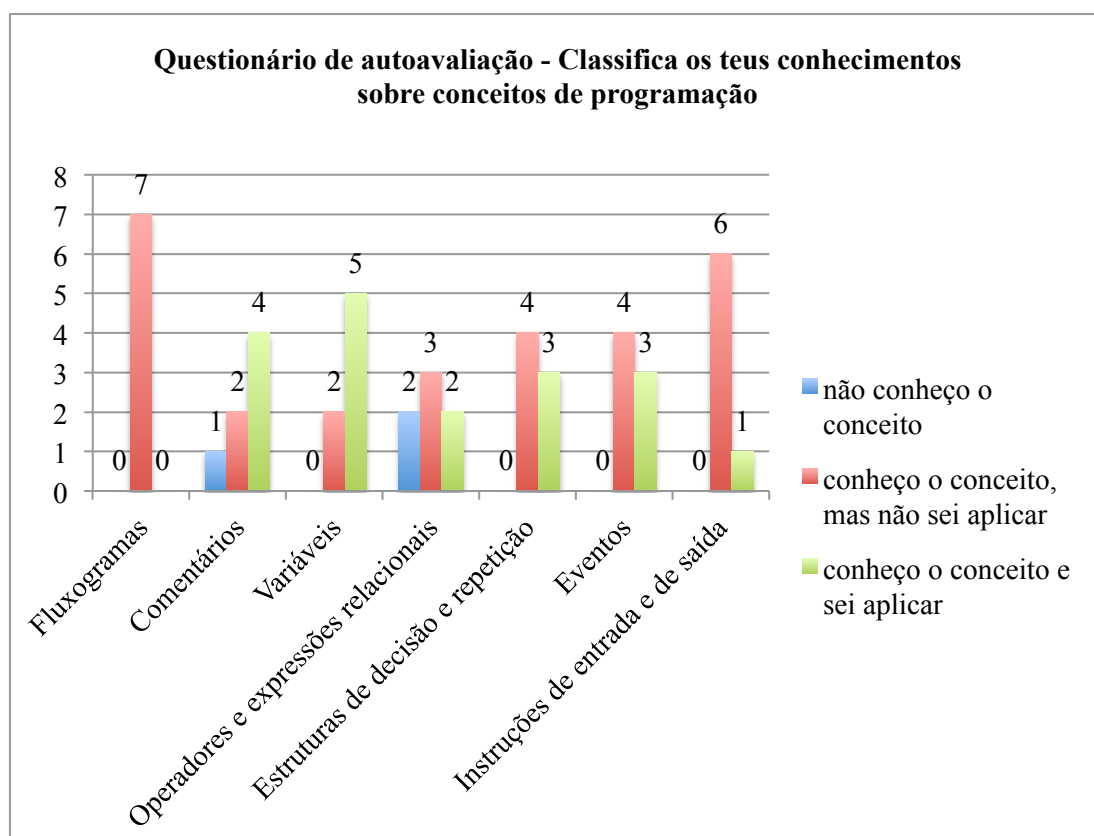


Figura 4. Respostas sobre o nível de conhecimento para cada conceito indicado no questionário de autoavaliação.

6.4. Testes de Autodiagnóstico e de Autoavaliação

Nos testes de diagnóstico e de avaliação, dois alunos não entregaram os ficheiros correspondentes ao grupo I e outros dois alunos não entregaram os do grupo II, pelo que foram analisadas as respostas de cinco alunos.

Em relação ao grupo I do teste de diagnóstico (*Figura 5*) o conceito de Comentário foi aquele em que os alunos tiveram maior dificuldade, tendo havido dois alunos que obtiveram o nível N1, sabem aplicar o conceito quando pedido diretamente para o fazer, e três alunos não responderam às questões relacionadas com este conceito. O conceito de Eventos foi aquele em que os alunos apresentaram menores dificuldades, tendo sido obtido o nível N1 por todos os alunos, dos quais três também obtiveram o nível N2, sabem aplicar o conceito apesar de não ser pedido diretamente. Sobre os conceitos Operadores e Expressões Relacionais e Estruturas de Decisão e Repetição, três alunos obtiveram o nível N1 e também o nível N2, tendo os restantes dois alunos não respondido às questões que abordam estes conceitos. Relativamente ao conceito de Fluxogramas, nenhum dos alunos obteve o nível N2, três alunos obtiveram o nível N1 e dois alunos não responderam às questões sobre este conceito.

No grupo II do teste de diagnóstico (*Figura 6*), todos os alunos reconhecerem a aplicação do conceito de Variáveis, mas não reconheceram a aplicação do conceito Instruções de Entrada e Saída. Em relação ao conceito de Comentários, quatro dos cinco alunos reconheceram a sua aplicação. No entanto, note-se que apenas um aluno reconheceu o conceito de Operadores e Expressões Relacionais e Estruturas de Decisão e Repetição.

Apenas considerando os alunos que entregaram os ficheiros do grupo I e do grupo II do teste de diagnóstico e considerando que apenas os alunos que reconheceram o conceito no grupo II e que souberam aplicar o conceito no grupo I é que conhecem o conceito, apenas um aluno conhece, reconhece e sabe aplicar o conceito Comentários e Eventos. Dois alunos conhecem, reconhecem e sabem aplicar o conceito Variáveis.

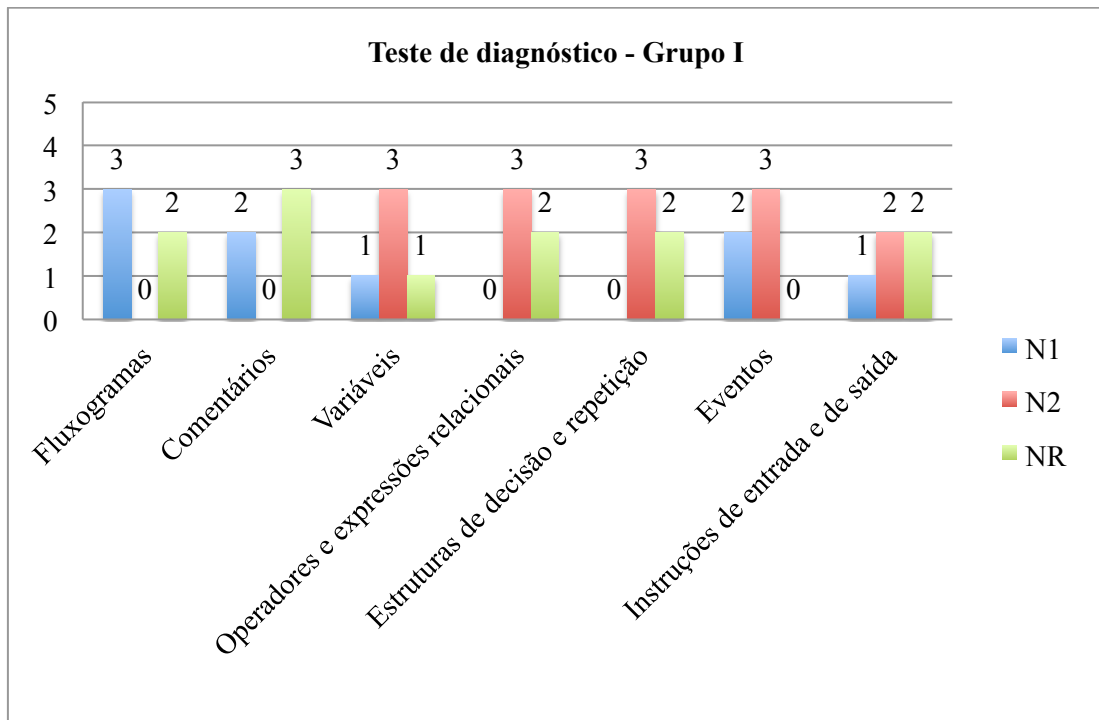


Figura 5. Respostas por nível de conhecimento para cada conceito abordado no grupo I do teste de diagnóstico.

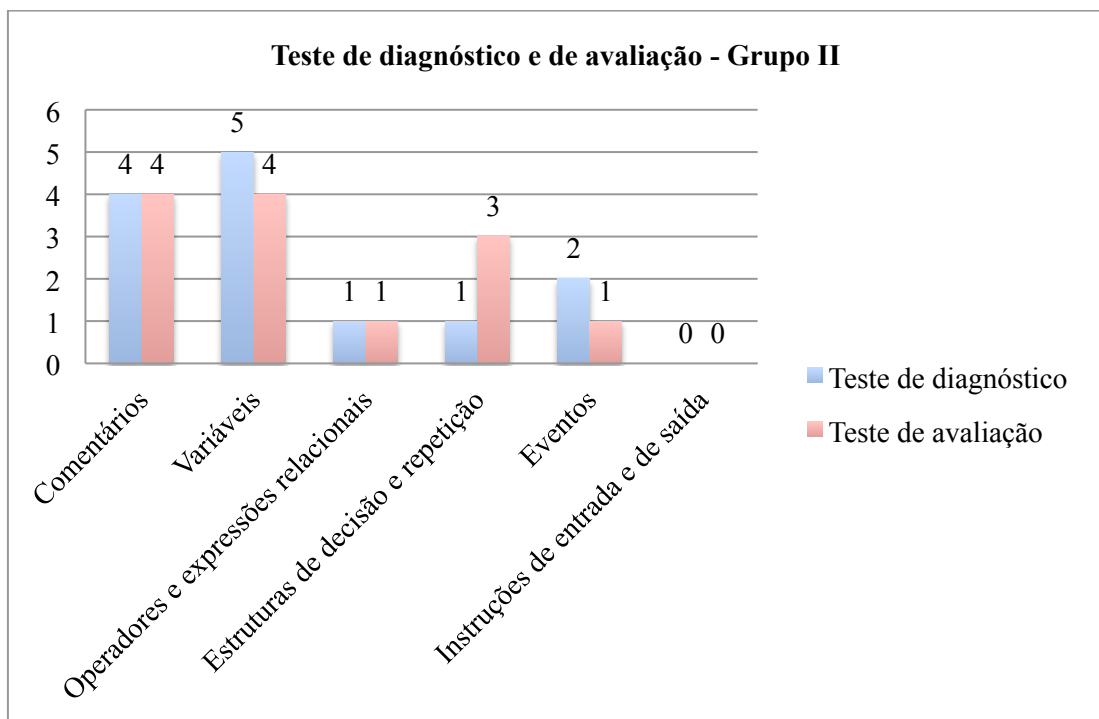


Figura 6. Respostas em que se manifestou reconhecer cada conceito abordado no grupo II do teste de diagnóstico e de avaliação.

No que diz respeito ao grupo I do teste de avaliação (*Figura 7*), os resultados foram semelhantes aos obtidos no teste de diagnóstico. No entanto, destacam-se algumas diferenças. Obteve-se mais uma resposta a questões relacionadas com os conceitos de Comentários e Operadores e Expressões Relacionais, tendo sido obtido o nível N1. Nos conceitos de Variáveis e Instruções de entrada e de saída, todos os alunos que responderam obtiveram o nível N2.

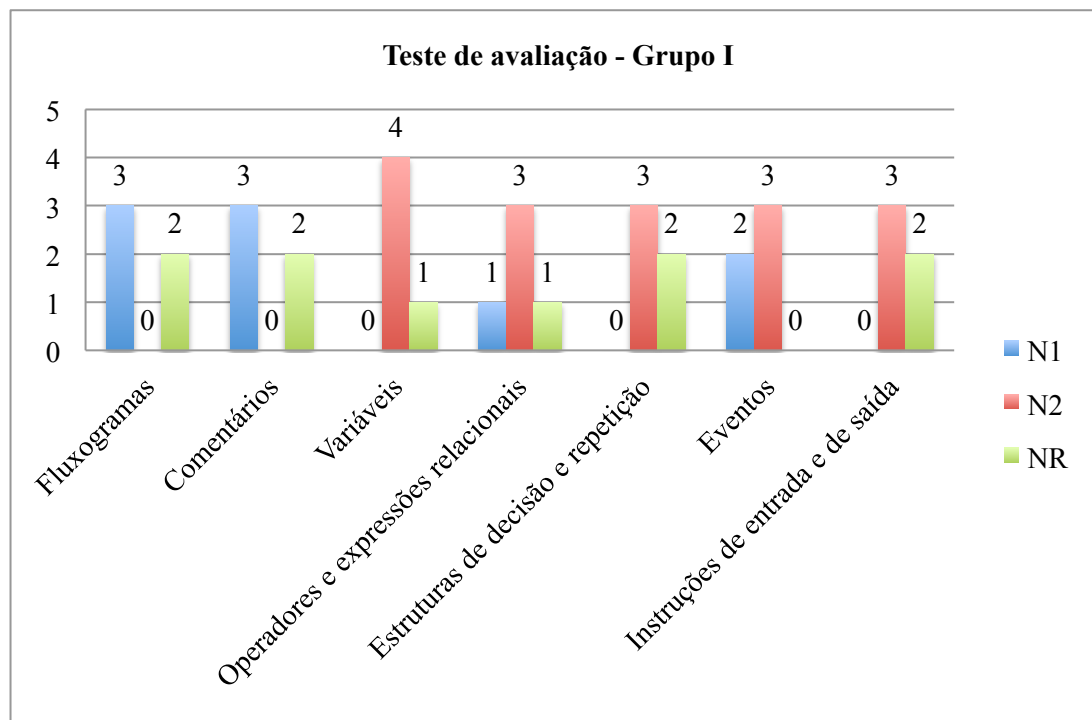


Figura 7. Respostas por nível de conhecimento para cada conceito abordado no grupo I do teste de avaliação.

No que diz respeito ao grupo II do teste de avaliação (*Figura 6*), também existem semelhanças com os resultados obtidos no teste de diagnóstico. Em relação aos conceitos Variáveis e Eventos, houve menos uma resposta correta na identificação da aplicação dos conceitos. No conceito de Estruturas de Decisão e Repetição, houve mais dois alunos que reconheceram a aplicação do conceito.

Apenas considerando os alunos que entregaram os ficheiros do grupo I e do grupo II do teste de avaliação e que apenas os alunos que reconheceram o conceito no grupo II e que souberam aplicar o conceito no grupo I é que conhecem o conceito, apenas um aluno da turma conhece, reconhece e sabe aplicar o conceito

Comentários, Operadores e Expressões Relacionais e Estruturas de Decisão e Repetição. Dois alunos conhecem, reconhecem e sabem aplicar o conceito Variáveis.

6.5. Registos de Aulas

Os indicadores que indicam que o aluno mostrou-se responsável pelo progresso do seu grupo de trabalho e coopera nos trabalhos de grupo, não foram manifestados em todas as aulas, nomeadamente na primeira e quinta aula em que foram realizados os questionários de autodiagnóstico e de autoavaliação e os testes de diagnóstico e de avaliação.

Uma vez que alguns alunos não estiveram presentes em todas as aulas da intervenção, o número de registos efetuados não foi igual para todas as aulas. Assim, as comparações entre aulas só foi realizada quando o número de alunos presente eram iguais. Na terceira aula não estiveram presentes dois alunos e na quarta aula não estiveram presentes três alunos.

De acordo com o registo de observações das aulas em que foram realizadas atividades utilizando os robôs da Lego® Mindstorms® NXT 2.0, totalizando três aulas, os alunos presentes mostraram estar interessados nas atividades (Quadro 12). O interesse dos alunos foi medido com base nos seguintes indicadores:

- está atento;
- realiza as tarefas propostas;
- mostra-se responsável pelo progresso do seu grupo de trabalho;
- participa espontaneamente;
- tenta ultrapassar as dificuldades;
- preocupa-se com a qualidade do trabalho.

Assim, verificou-se que a maioria dos alunos mostrou um comportamento sistemático nos indicadores referidos.

Quadro 12

Porcentagem de Alunos que Manifestaram Sistemáticamente um Determinado Comportamento, para cada Indicador de Interesse nas Aulas Dois, Três e Quatro.

Indicador	perc.
Está atento	89%
Realiza as tarefas propostas	79%
Mostra-se responsável pelo progresso do seu grupo de trabalho	62%
Participa espontaneamente	71%
Tenta ultrapassar as dificuldades	55%
Preocupa-se com a qualidade do trabalho	60%

Comparando o interesse dos alunos na primeira aula, onde foi realizado o questionário de autodiagnóstico e o teste de diagnóstico, com a quinta aula, onde foi realizado o questionário de autoavaliação e o teste de avaliação, existe a percepção de que o interesse diminuiu (Quadro 13). O interesse dos alunos foi medido com base nos seguintes indicadores:

- está atento;
- realiza as tarefas propostas;
- participa espontaneamente;
- tenta ultrapassar as dificuldades;
- preocupa-se com a qualidade do trabalho.

Verificou-se que na quinta aula, aumentou o número de alunos que mostraram comportamentos ocasionais em relação aos indicadores mencionados.

Quadro 13

Número de Alunos que Manifestaram um Determinado Comportamento, para cada Indicador de Interesse na Primeira e Quinta Aula.

Indicador	Primeira aula			Quinta aula		
	A	B	C	A	B	C
Está atento	4	3	0	2	1	4
Realiza as tarefas propostas	3	3	1	1	2	4
Participa espontaneamente	3	1	3	1	2	4
Tenta ultrapassar as dificuldades	0	4	3	1	2	4
Preocupa-se com a qualidade do trabalho	1	3	3	3	1	3

Nota: Para cada indicador da grelha de registos, é indicada o número de alunos que mostrou um comportamento sistematicamente (A), dominante (B) e ocasionalmente (C).

No total das cinco aulas da intervenção (Quadro 14), verificou-se que a maioria dos alunos manifestou sistematicamente comportamentos responsáveis, principalmente no que diz respeito aos bens pessoais dos colegas e da escola, por 93% dos alunos. Na dimensão autonomia, a utilização do tempo de forma organizada foi manifestada predominantemente por 69% dos alunos. A cooperação nos trabalhos de grupo foi manifestado sistematicamente por 86% dos alunos e o respeito pelos direitos, opiniões e capacidades dos outros foi manifestado sistematicamente por 53%. Na dimensão da comunicação e expressão, a maioria dos alunos expressou-se sistematicamente de uma forma clara, 70% dos alunos, com correção linguística, 63% dos alunos, e com vocabulário específico, 57% dos alunos.

Quadro 14

Percentagem de Alunos que Manifestou um Determinado Comportamento em Todas as Aulas.

Indicador	A	B	C
Responsabilidade			
Cumprir as regras estabelecidas e as diretrizes que lhe são dadas	60%	27%	13%
É cuidadoso com os bens pessoais e da escola	93%	7%	0%
É atento	67%	20%	13%
Realiza as tarefas propostas	57%	27%	17%
Mostra-se responsável pelo progresso do seu grupo de trabalho	63%	31%	6%
Participa espontaneamente	50%	23%	27%
Autonomia			
Tenta ultrapassar as dificuldades	30%	43%	27%
Preocupa-se com a qualidade do trabalho	47%	30%	23%
Utiliza o tempo de uma forma organizada	14%	69%	17%
Identifica as suas dificuldades	47%	43%	10%
Solidariedade			
Entreajuda na realização de tarefas	33%	43%	23%
Coopera nos trabalhos de grupo	86%	14%	0%
Respeita os direitos, opiniões e capacidades dos outros	53%	43%	3%
Comunicação/Expressão			
Exprime-se de forma clara	70%	30%	0%
Exprime-se com correção linguística	63%	37%	0%
Utiliza vocabulário específico	57%	37%	7%

Nota: Para cada indicador da grelha de registos, é indicada a percentagem de alunos que mostrou um comportamento sistematicamente (A), dominante (B) e ocasionalmente (C).

6.6. Discussão de Grupo

Durante a discussão de grupo, foi indicado por todos os alunos que, em comparação com outras linguagens de programação, a linguagem NXT-G é mais prática, bem como na utilização do programa Lego® Mindstorms® NXT v2.0. Um dos motivos apontado pelos alunos foi facto de utilizar uma linguagem de programação visual, em que basta arrastar ícones para programar o robô. A interação foi também considerado como um motivo para tornar a programação e a utilização dos robôs mais prática, uma vez os alunos não interagiam apenas com o programa onde desenvolviam o código, mas também com o robô, observando diretamente as consequências da programação. Um dos exemplos, dados pelos alunos dessa interação, foi o facto de conseguirem visualizar os passos do robô consequentes à programação que tinham realizado.

Apesar de o estacionamento não funcionar nas primeiras vezes e terem de corrigir a programação realizada, todos os alunos consideraram que esta linguagem de programação era mais simples e prática. No entanto, a maior parte dos alunos manifestou não gostar de programação, independentemente da linguagem utilizada.

Devido à necessidade de os alunos obterem aprovação nos diversos módulos das disciplinas de programação, a maioria vê a aprendizagem da programação e dos seus conceitos como uma obrigação e necessidade, não sendo do seu interesse pessoal e não considerando que possa vir a ser útil no futuro. Outro factor apontado por alguns alunos como justificação para a sua falta de interesse pela programação consiste nos objetivos propostos nos problemas, trabalhos e exercícios colocados em sala de aula, considerando-os pouco estimulantes e relevantes de acordo com os seus interesses e dia-a-dia.

Apenas um aluno afirmou gostar de programar, apresentando como motivo o facto de ter facilidade em programar, uma vez que consegue acompanhar os conteúdos envolvidos e compreender a lógica da programação.

A falta de conhecimentos básicos de programação e a falta de compreensão dos conceitos envolvidos é apresentada, por alguns alunos, como um motivo para considerarem desinteressante a programação e estarem desmotivados para a sua aprendizagem. Estes alunos afirmaram que não conseguem acompanhar os conteúdos envolvidos, conforme estes se vão tornando progressivamente mais complexos. Apesar de poderem utilizar a Internet na pesquisa sobre os problemas que encontram

na programação, a maioria dos alunos não consegue entender o código e as explicações que encontram, acontecendo o mesmo na consulta de manuais ou nas explicações orais que são dadas em sala de aula. No entanto, quando a explicação oral é reformulada, como aconteceu durante a intervenção, a maioria dos alunos afirmaram terem compreendido melhor essa explicação. A percepção de alguns alunos é que o problema está neles e não no professor e na linguagem que é utilizada por este, uma vez que afirmam tratar-se de uma linguagem adequada para ser usada em sala de aula.

Em relação aos conceitos de programação envolvidos na programação no Lego® Mindstorms® NXT 2.0, a maior parte dos alunos afirmou que no início não tinha a noção de estar a aplicar esses conceitos. A maioria dos alunos indicou que começou a aperceber-se de que estava a aplicar os conceitos e a relacioná-los com o que era realizado noutras linguagens de programação, quando isso lhes era indicado pelo professor através de sugestões e questões que incluíam os nomes dos conceitos. No entanto, afirmaram que a programação no Lego não os ajudaria a compreender outras linguagens de programação, apesar de alguns conceitos aplicados serem semelhantes, uma vez que é muito diferente das outras linguagens de programação que conhecem.

Alguns alunos indicaram que uma das dificuldades sentidas durante a intervenção foi programar a sequência de execução das instruções, como por exemplo, ter que programar o robô para encontrar um espaço vazio. Para tal, seria necessário estabelecer uma sequência que permitisse ao robô andar em frente enquanto não encontra um espaço vazio, através do uso do sensor ultrassónico, e fazê-lo parar quando encontrasse esse espaço. Alguns alunos também sentiram dificuldades em ajustar os valores da velocidade, do grau da direção das rodas e de distância do sensor ultrassónico, uma vez que era necessário realizar vários testes e ajustes desses valores, para que o estacionamento fosse ocorresse de acordo com o que esses alunos tinham pensado. Para um dos alunos, os testes e ajustes não representaram a maior dificuldade, mas sim compreender como funcionava o sensor ultrassónico e a relação entre os valores e as consequências nos movimentos do robô. No entanto, estas dificuldades foram superadas com as indicações visuais do programa Lego® Mindstorms® NXT v2.0, como por exemplo, serem mostradas setas que davam indicação da direção das rodas quando era ajustado o valor do grau da direção das rodas.

A maioria dos alunos sentiu a necessidade de as aulas da disciplina serem mais dinâmicas e de haver a necessidade de realizar revisões dos conceitos de programação, como ocorreu durante a intervenção na utilização dos robôs. Os alunos admitiram que a necessidade de revisões se deve ao facto de não se terem interessado pelas aulas no início do curso, sentindo dificuldades em acompanhar a matéria ao longo do ano letivo.

A necessidade de existirem aulas mais dinâmicas, exercícios com objetivos do seu interesse e a utilização de robôs como os da Lego, foi considerado pela maioria dos alunos com fatores que tornariam as aulas e a aprendizagem da programação mais interessantes.

Sobre o efeito da realização de um questionário em que teriam que avaliar os conhecimentos que possuíam sobre um determinado conceito, seguido de um teste em que tinham que aplicar os conhecimentos, alguns alunos afirmaram que tinham indicado que sabiam determinados conceitos, mas com a realização do teste verificaram que não sabiam. Alguns alunos afirmaram que as diferenças entre as respostas dadas, no questionário de autodiagnóstico e no teste de diagnóstico, foram maiores nestes do que quando responderam ao questionário de autoavaliação e ao teste de avaliação. A maioria dos alunos afirmou que a diferença não se deveu a não reconhecerem o nome dado aos conceitos, mas sim por pensarem que sabiam o conceito e aplicá-lo, quando afinal alguns conceitos só ouviram falar dele e não sabiam aplicar. Alguns alunos manifestaram não gostarem de realizar testes. No entanto, outros alunos afirmaram que é indiferente se realizam um teste ou não, o que não gostam é terem que programar como aconteceu no teste de diagnóstico e de avaliação.

7. Discussão dos Resultados

Em relação ao objetivo de aprendizagem do aluno em reconhecer a aplicação de conceitos em programas já criados, verifica-se que o total de respostas corretas no grupo II do teste de avaliação foi igual que no grupo II do teste de diagnóstico. Apenas se verificou diferenças nas respostas em alguns conceitos, nomeadamente, Variáveis, Estruturas de Decisão e Repetição e Eventos.

Comparando os resultados do grupo II do teste de diagnóstico com o questionário de autodiagnóstico, os resultados mostram que a percentagem de alunos que consegue reconhecer os conceitos num excerto de código é maior do que a percentagem de alunos que indicaram conhecer e saber aplicar no questionário (Figura 8). Os conceitos onde se verificou que a percentagem de alunos que tinham indicado no questionário que sabiam o conceito e aplicar na programação era maior do que a percentagem de alunos que reconheceram esses conceitos no grupo II do teste de diagnóstico, foi em relação ao conceito Estruturas de Decisão e Repetição e Instruções de Entrada e de Saída.

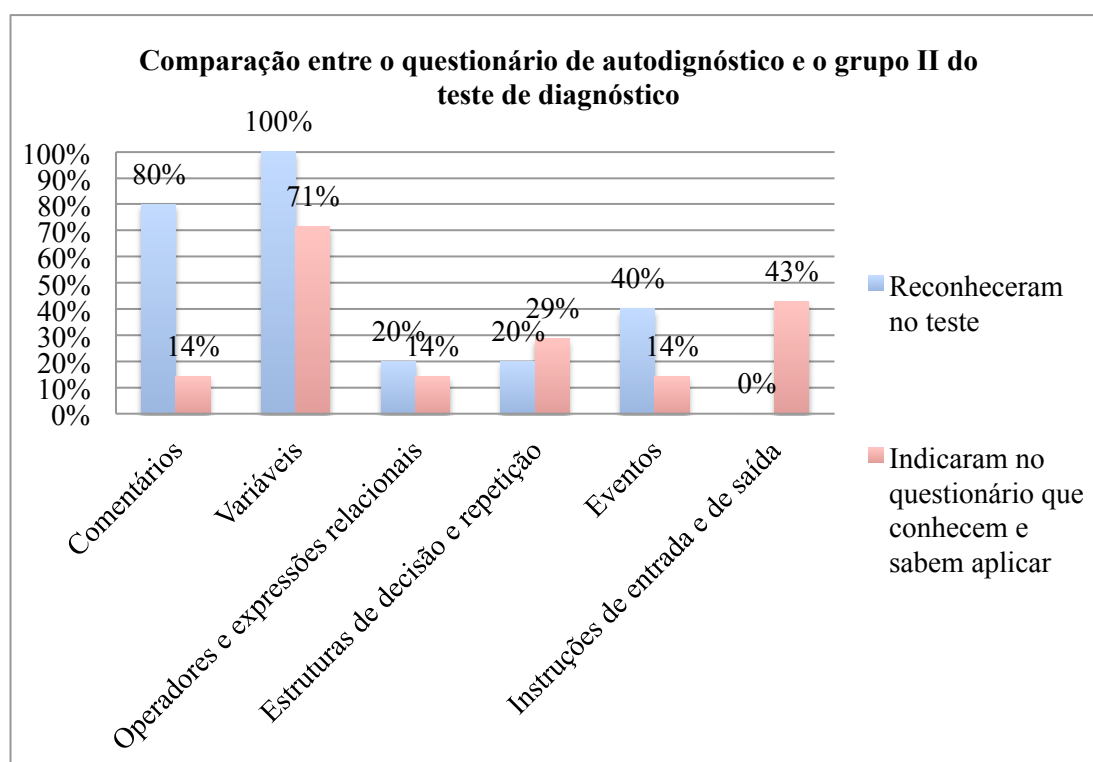


Figura 8. Percentagem de alunos que afirmaram, no questionário de autodiagnóstico, conhecer e saber aplicar um conceito e que reconheceram a aplicação desse conceito no grupo II do teste de diagnóstico.

Para os conceitos Comentários, Variáveis e Estruturas de Decisão e Repetição, a percentagem de alunos que reconheceram o conceito no grupo II do teste de avaliação é maior do que a percentagem de alunos que afirmou conhecer e saber aplicar o conceito no questionário de autoavaliação (*Figura 9*).

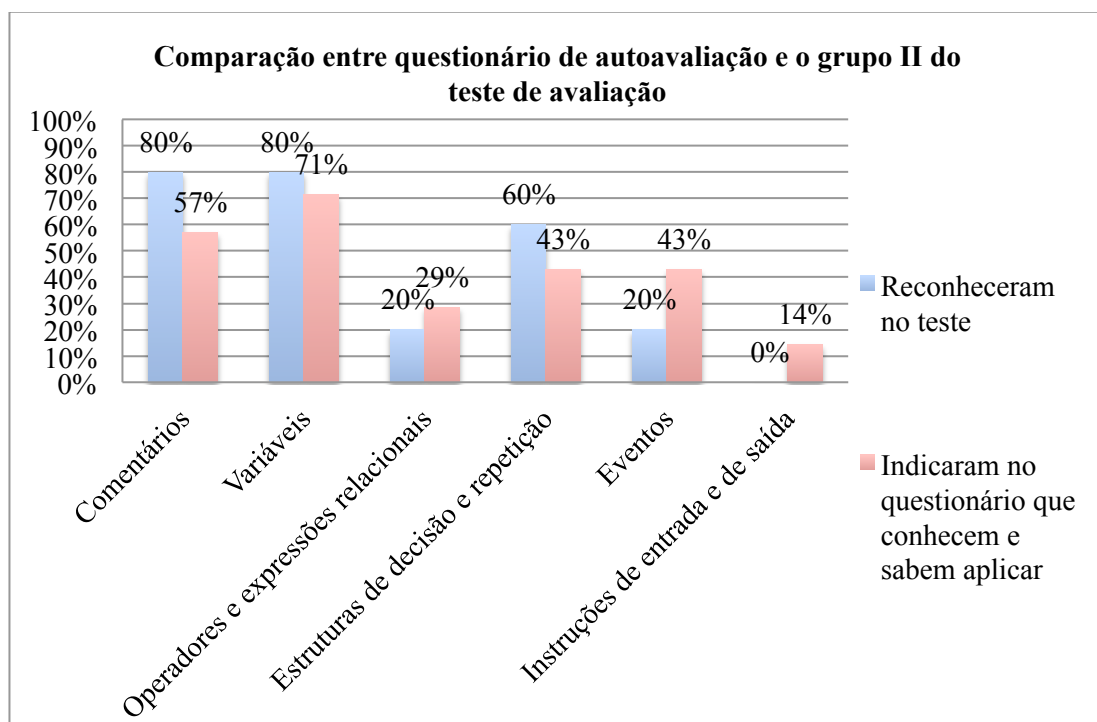


Figura 9. Percentagem de alunos que afirmaram, no questionário de autoavaliação, conhecer e saber aplicar um conceito e que reconheceram a aplicação desse conceito no grupo II do teste de avaliação.

No que diz respeito em saber aplicar conceitos de programação, verificou-se melhorias nos resultados dos alunos em relação a alguns conceitos. Existiram alunos que obtiveram classificações N2 a um maior número de conceitos, nomeadamente Variáveis e Instruções de Entrada e Saída.

Comparando os resultados do questionário de autodiagnóstico com o questionário de autoavaliação, verificou-se que os alunos indicaram que melhoraram os conhecimentos sobre os conceitos:

- Fluxogramas;
- Comentários;
- Operadores e Expressões Relacionais;
- Estruturas de Decisão e Repetição;

- Eventos.

As diferenças observadas nos questionários podem ser justificadas pelo facto de os testes de diagnóstico e de avaliação terem sido realizados imediatamente a seguir aos respectivos questionários. Desta forma, os alunos poderão ter tido uma maior consciência dos seus conhecimentos, dando respostas diferentes no questionário de autoavaliação em comparação com o questionário de autodiagnóstico. A possibilidade de os alunos consultarem manuais e poderem utilizar a Internet durante a realização dos testes, poderão ser outros factores que influenciaram os resultados obtidos, uma vez que poderá ter contribuído para os alunos aprenderem sobre alguns dos conceitos abordados.

Comparando os resultados do questionário de autodiagnóstico e com o grupo I do teste de diagnóstico, verificou-se que a percentagem de alunos que conseguiu aplicar os conceitos é superior à percentagem de alunos que afirmou no questionário conhecer e saber aplicar o conceito (*Figura 10*).

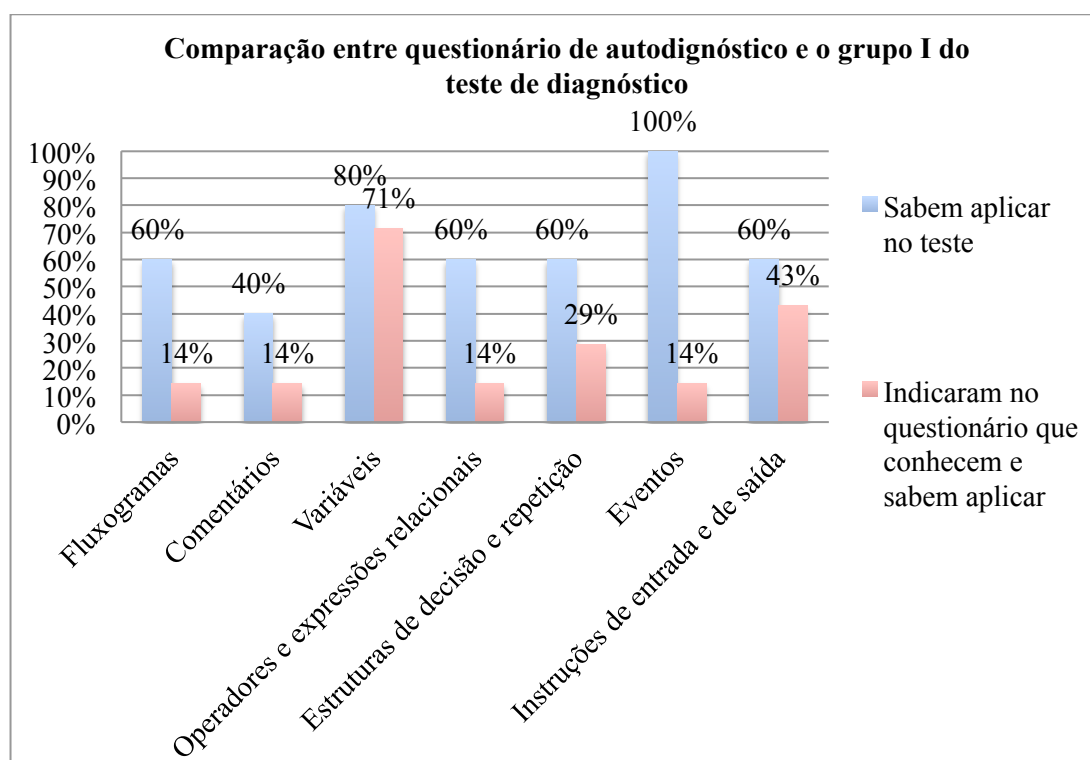


Figura 10. Percentagem de alunos que afirmaram no questionário de autodiagnóstico conhecer e saber aplicar um conceito e que aplicaram o conceito no grupo I do teste de diagnóstico.

Os resultados da comparação entre o questionário de autoavaliação com o grupo I do teste de avaliação são semelhantes aos obtidos entre o questionário de autodiagnóstico e o teste de diagnóstico. Em ambos os casos, a percentagem de alunos que consegue aplicar os conceitos é maior do que a percentagem de aluno que o referiu no questionário (*Figura 11*).

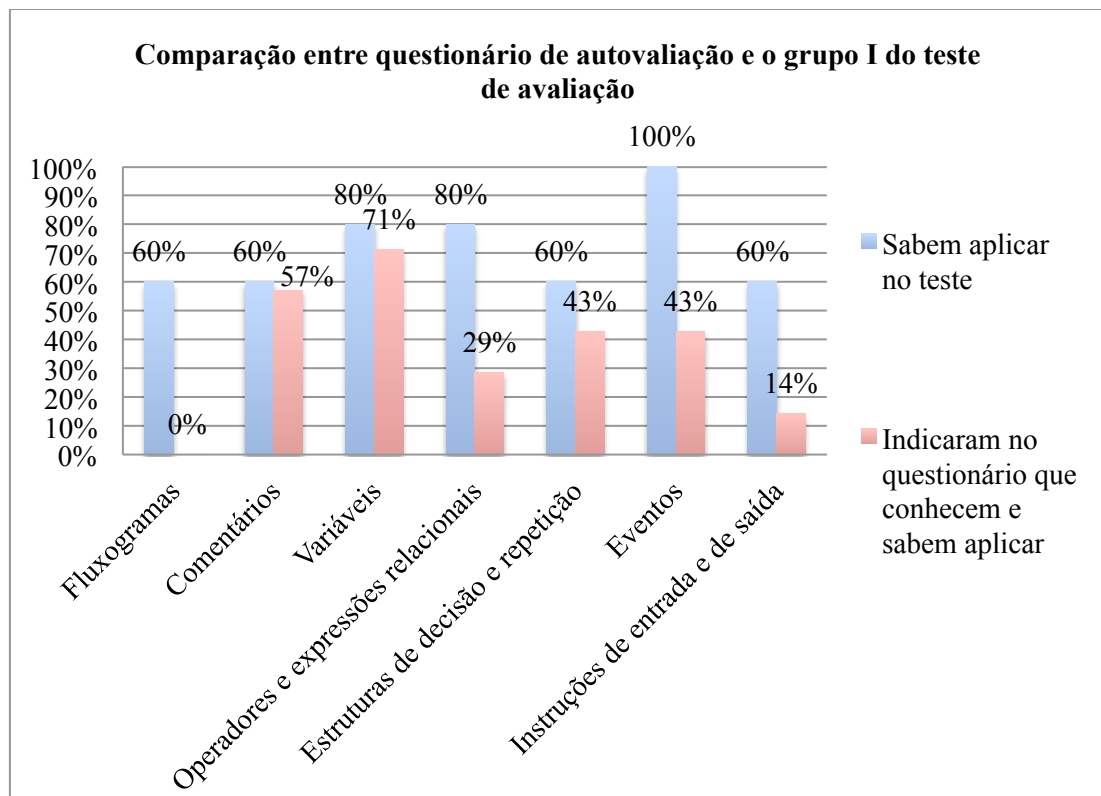


Figura 11. Percentagem de alunos que afirmaram no questionário de autoavaliação conhecer e saber aplicar um conceito e que aplicaram o conceito no grupo I do teste de avaliação.

A diferença de resultados entre os questionários e os testes, poderá ser consequência de os testes terem sido resolvidos com consulta de manuais, exercícios resolvidos anteriormente pelos alunos e na Internet. Os alunos poderiam não conhecer o conceito ou conhecer e não saberem aplicar na programação, no entanto, ao realizarem pesquisas na Internet ou em manuais, podem ter aprendido esses conceitos e aplica-los/reconhece-los na programação.

Comparando as respostas nos dois questionários, os resultados podem refletir uma maior consciência dos alunos dos seus reais conhecimentos ou uma aprendizagem resultante da atividades da realização do teste de diagnóstico e da

utilização da robótica educativa. Na discussão de grupo alguns alunos afirmaram que as respostas que deram nos questionários não correspondiam às dificuldades sentidas no teste. No entanto, os resultados obtidos nos testes foram melhores do que os níveis de conhecimentos que tinham sobre os conceitos.

Todos os grupos de trabalho souberam criar programas que resolvem um problema. Uma evidência dessa capacidade, foi terem conseguido programar os robôs da Lego® Mindstorms® NXT 2.0 para realizarem o estacionamento automático. No entanto, um dos grupo solucionou para dois tipos de estacionamento e os outros dois grupos conseguiram para os três tipos de estacionamento. O grupo que apenas solucionou para dois tipos de estacionamento, poderá ser explicado por duas das três aulas que os grupos tiveram para solucionar o estacionamento automático, este grupo foi representado por apenas um aluno, tendo o seu colega de grupo estado ausente.

Um dos grupos, nas soluções inicialmente descritas, indicaram que iriam utilizar três sensores ultrassônicos, para encontrar um lugar vazio e evitar a colisão com os possíveis veículos estacionados nos lugares adjacentes ao lugar vazio. Também indicaram pretenderem usar um sensor de cor, para detetar a linha final de estacionamento e informar o robô de que este terminou o estacionamento para os tipos em espinha e em paralelo. Este grupo não esteve presente na terceira aula e apenas um dos elementos na quarta aula, o que não permitiu que tivessem tempo para concretizar as soluções inicialmente descritas. Assim, utilizaram apenas um sensor ultrassônico para detetar um lugar vazio, como os restantes grupos.

As soluções descritas na plataforma PBworks e programadas nos robôs da Lego, os alunos tiveram a preocupação de que o estacionamento automático fosse o mais parecido possível ao realizado por um condutor humano. Esta preocupação foi manifestada em diversos estudos sobre o estacionamento automático de veículos inteligentes (Camus, Coelho & Quadrado, 2002; Szádeczky-Kardoss, Kiss & Wahl, 2008). Uma outra preocupação manifestada nas soluções dos grupos de trabalho e em estudos de investigação, foi que o robô tivesse a capacidade de encontrar um lugar de estacionamento vazio e detetar outros veículos que se encontrassem nos restantes lugares (Camus et al., 2002; Ryu, Kim & Oh, 2008; Szádeczky-Kardoss et al., 2008; Xu, Chen & Xie, 2000; Zhao & Collins, 2005).

A realização de uma discussão em grupo foi importante para os alunos refletirem e indicarem de que forma o condutor sabe que existe um lugar vazio,

como sabe que colidiu com outro veículo estacionado e como sabe que está dentro dos limites do lugar de estacionamento. Desta forma, os alunos tiveram uma melhor percepção de quais os sensores que poderiam utilizar, uma vez que é com base nestes sensores que um veículo inteligente trata a informação recebida, planeia e executa os movimentos necessários (Ryu et al., 2008; Szádeczky-Kardoss et al., 2008; Xu et al., 2000; Zhao & Collins, 2005). Assim, os grupos de trabalho optaram por utilizar apenas um sensor ultrassónico que deteta o lugar de estacionamento vazio.

Em relação aos movimentos que o robô deve executar para cada tipo de estacionamento, as soluções apresentadas não permitem que os movimentos sejam adaptados automaticamente a diferentes situações. Os alunos definiram valores específicos para a velocidade, a distância e o ângulo do sentido, de acordo com as situações apresentadas na maquete. Assim, as soluções encontradas pelos alunos só podem ser aplicadas às situações em que foram testadas. A determinação previa dos movimentos que o veículo devia executar é um método que já foi aplicado em alguns estudos de investigação (Camus et al., 2002; Xu et al., 2000).

Na utilização de uma descrição das soluções para cada tipo de estacionamento, um grupo de trabalho, alterou a sua página no decorrer da intervenção, mas apesar de ir reforçar o pedido, deixou-o de o fazer. Aparentemente, deveu-se ao facto de o grupo ter-se apercebido que não teriam tempo para programar toda a solução dentro do número de aulas estabelecidos para essa atividade. A falta de tempo deveu-se a algumas dificuldades que o grupo sentiu em programar alguns passos das suas soluções e na quarta aula, um dos elementos não esteve presente.

De acordo com o registo das aulas em que foram realizadas atividades utilizando os robôs da Lego® Mindstorms® NXT 2.0, totalizando três aulas, os alunos presentes mostraram estar interessados nas atividades. O que está de acordo com o que foi transmitido pela maioria dos alunos na discussão de grupo, em que manifestaram que a existência de aulas mais dinâmicas, exercícios com o objetivos do seu interesse e a utilização de robôs como os da Lego, tornaria o ensino e a aprendizagem da programação mais interessante. O facto de dois dos três grupos terem conseguido programar o robô para o estacionamento nos três tipos apresentados, poderá evidenciar interesse dos alunos em solucionar o problema do estacionamento automático e utilizarem como ferramenta de aprendizagem os robôs da Lego® Mindstorms® NXT 2.0.

Durante a discussão de grupo procurou-se obter algumas opiniões sobre a relevância da utilização da robótica educativa na compreensão do aluno em relação aos efeitos da programação desenvolvida e aos conceitos envolvidos. Na utilização dos robôs Lego, todos os alunos consideraram vantajoso o facto de existir uma interação direta com o programa para o desenvolvimento do código juntamente com o robô onde conseguiam visualizar as consequências da programação desenvolvida. Outra vantagem mencionado por todos os alunos foi a simplicidade da linguagem de programação utilizada devido se tratar de uma linguagem de programação visual e a utilização dos robôs associada à programação que desenvolveu, de acordo com os alunos, tornou a atividade de programar mais prático em relação a outras linguagens de programação que tinham utilizado nas aulas. Em relação à percepção dos conceitos de programação envolvidos na programação dos robôs da Lego, a maior parte dos alunos manifestou que não se apercebiam da utilização desses conceitos. No entanto, os alunos indicaram que ao longo da intervenção essa percepção foi alterada devido às orientações do professor em que eram referidos os conceitos. A utilização do robô da Lego poderá contribuir para a percepção das consequências da programação que os alunos desenvolvem. No entanto, em relação à percepção dos conceitos envolvidos, a utilização do robô da Lego poderá não ser eficaz se o professor não desempenhar um papel orientador durante o desenvolvimento da programação realizada pelos alunos.

Na comparação dos resultados do teste de diagnóstico com o teste de avaliação, verificaram-se piores resultados no teste de avaliação para o conceito Eventos. Menos alunos reconheceram o conceito num excerto de código na linguagem Visual Basic. No entanto, nas questões do grupo I do teste de avaliação, todos os alunos aplicaram o conceito Eventos. Esse motivo, poderá dever-se ao facto de as questões onde é pedido aos alunos para programarem um evento, esta encontra-se contextualizada com uma situação. Assim, os alunos podem não se aperceber de precisarem aplicar o conceito Evento, mas sim do resultado que se espera obter.

Os resultados obtidos com os testes de diagnóstico e de avaliação, podem não ter evidenciado os verdadeiros conhecimentos dos alunos. Apesar de um dos métodos de avaliação da preferência do aluno ser através de testes práticos, verificou-se por parte de alguns alunos falta de responsabilidade, ao não entregarem todos os ficheiros que criaram para a resolução dos testes. Mesmo depois de ter sido pedido que procurassem pelos ficheiros em falta e os entregassem, os alunos que não o fizeram, tendo afirmado que não os encontravam. Após o teste de diagnóstico e ter-

me apercebido que alguns alunos não entregaram os ficheiros, a turma foi alertada para esse facto e para entregarem todos os ficheiros no momento do teste de avaliação. Na aula em que foi realizado o teste de avaliação, a turma foi novamente alertada para entregarem todos os ficheiros. No entanto, alguns alunos não o fizeram. Para além da falta de responsabilidade, verificou-se que o interesse em realizar o teste de avaliação foi menor do que no momento de realizarem o teste diagnóstico. Assim, estes factores podem ter influenciado os resultados apresentados sobre os testes de diagnóstico e de avaliação.

Na discussão de grupo, alguns alunos afirmaram terem obtido uma maior consciência dos conhecimentos que detinham sobre alguns conceitos de programação quando foi realizado um questionário de autodiagnóstico seguido do teste de diagnóstico. Essa maior consciência, de acordo com a maioria dos alunos, foi manifestada nas respostas que cada um deu nos questionários de autodiagnóstico e autoavaliação, em que afirmaram que no questionário de autoavaliação mudaram as suas respostas em relação ao questionário de autodiagnóstico. No entanto, de referir que em alguns alunos as respostas dadas nos questionários não correspondiam aos resultados dos testes. Essas diferenças poderão ser consequência de os alunos não terem a consciência de que estão a aplicar um conceito de programação quando escrevem código numa linguagem de programação.

8. Reflexão Crítica

Tendo como objetivo a consolidação de alguns conceitos de programação, os resultados podem demonstrar que a intervenção conseguiu atingir o objetivo a que estava proposto. No entanto, o objetivo só poderia ter sido alcançado associando o ensino de conceitos de programação com a utilização da Robótica Educativa, juntamente com um conjunto de princípios pedagógicos e estratégias, nomeadamente a Aprendizagem Baseada em Problemas.

De acordo com as indicações dadas pelos alunos, a percepção da aplicação de alguns conceitos na programação e utilização dos robôs da Lego® Mindstorms® NXT 2.0, foi concretizada devido às questões, orientações e feedback dados pelo professor aos alunos durante o decorrer das atividades. Assim, a aplicação da Robótica Educativa só poderá ser possível se estiver associada a um papel do professor como orientador das aprendizagens.

A Robótica Educativa poderá também ter tido influência no interesse manifestado pelos alunos no decorrer das aulas em que decorreu as atividades com os robôs. De acordo com os registos das aulas e a discussão de grupo depois de ter sido realizada a intervenção, foi evidenciado esse interesse, tendo sido justificado pela maior parte dos alunos pela facilidade de programarem os robôs através de uma Linguagem de Programação Visual, pelo *feedback* da programação desenvolvida transmitido pelos robôs e pela aplicação de estratégias de ensino diferentes à que estava a ser aplicado pelo professor cooperante. No entanto, o interesse pela programação não foi alterado, tendo a maior parte dos alunos afirmado que não gostam de programar e não pretendem aplicar na sua vida profissional os conhecimentos adquiridos na disciplina de Linguagens de Programação.

Existe um conjunto de possíveis limitações na intervenção realizada, sendo uma delas o número reduzido de aulas, três, em que foi realizado atividades aplicando a Robótica Educativa com os robôs da Lego® Mindstorms® NXT 2.0. Outra limitação foi o número reduzido de alunos envolvidos, sete, tendo na segunda aula sido reduzido para cinco e na terceira aula para quatro, devido à falta de assiduidade. Durante a realização do teste de diagnóstico e de avaliação, verificou-se em dois alunos falta de interesse e de responsabilidade ao não entregarem alguns ficheiros com as respostas aos testes, não tendo sido considerado as respostas que se encontravam nesses ficheiros nos resultados apresentados. Este factor juntamente

com a possibilidade de os alunos poderem consultar manuais e pesquisar na Internet, durante a realização dos testes poderá ter influenciado os resultados, apesar de alguns alunos afirmarem que têm dificuldade em encontrar soluções para os problemas que encontram na programação.

A realização da avaliação da intervenção ou das aprendizagens com base na realização de testes de diagnóstico e de avaliação, poderá não ser o instrumento mais exato dos conhecimentos adquiridos pelos alunos. Os testes foram realizados em momentos pontuais da intervenção, duas aulas, podendo os seus resultados serem influenciados por diversos motivos, nomeadamente pelo interesse dos alunos. Este facto pode ter influenciado a atitude dos alunos perante a realização dos testes, tendo-se verificado um menor interesse por parte de alguns alunos na realização do teste de avaliação, em que foi necessário questioná-los sobre essa falta de interesse. No entanto, foi manifestado por alguns alunos preocupações com a avaliação dos testes e as consequências na classificação do módulo 17, apesar de ter sido transmitido que os objetivos dos testes era determinar se a intervenção teria influenciado os conhecimentos que detinham sobre alguns conceitos de programação e se teria contribuído para que os alunos se apercebessem dos seus conhecimentos e dificuldades na programação. Esta preocupação poderá ser justificada pelos critérios de avaliação determinados para a disciplina, em que os testes estão incluídos na avaliação dos domínios das competências e cognitivo com um peso de 55% da nota final.

Durante a intervenção assumi a preocupação de que os alunos teriam que realizar aprendizagens em todas as cinco aulas, incluindo aquelas que serviriam como avaliação através da realização de questionários e de testes. Ao substituir o professor cooperante na lecionação das aulas, não considerei ético a possibilidade de os alunos estarem presentes em duas aulas sem que o objetivo principal dessas aulas não fosse as suas aprendizagens. Esta preocupação associada à minha experiência profissional como programador em que é prática comum a consulta de manuais para o desenvolvimento de programas, em qualquer linguagem de programação. Assim, foi permitido aos alunos que o fizessem na realização dos testes. Para além disso, durante a realização dos testes, disponibilizei-me em esclarecer algumas dúvidas de interpretação das questões através da reformulação das mesmas. Assim, os resultados do teste de diagnóstico e do teste de avaliação e as diferenças observadas entre os dois poderão estar relacionadas com as aprendizagens adquiridas pelos alunos

durante a realização dos testes juntamente com a realização das atividades usando a robótica educativa.

A realização de questionários, que permitisse aos alunos refletirem sobre os conhecimentos que tinham sobre conceitos de programação, seguidos de um teste, em que neste caso teriam que aplicar e reconhecer a aplicação de alguns conceitos de programação, poderá ter permitido que os alunos ficassem com uma melhor percepção dos seus conhecimentos. Este facto foi comunicado pela maioria dos alunos durante a discussão de grupo. No entanto, para esta estratégia sirva como orientação de futuras aprendizagens, poderá ter que existir um maior interesse na programação pelos alunos, uma vez que um dos factores mencionados por alguns alunos para as dificuldades que sentem em programar foi a falta de interesse que têm.

A aplicação de uma estratégia de aprendizagem baseada em problemas em associação à realização de trabalho em grupo e do papel de orientador pelo professor, permitiu a aplicação dos princípios pedagógicos de autonomia, desafio/motivação, democracia, relevância, reflexão e responsabilidade. Para o princípio pedagógico de desafio/motivação, revelou-se importante a existência de três níveis de dificuldade no problema do estacionamento automático. Foi manifestado em sala de aula uma competitividade saudável entre os grupos, existindo trocas de ideias e propostas de soluções para os problemas que cada grupo enfrentou. No entanto, no princípio pedagógico de democracia, foi necessário uma maior intervenção da minha parte, uma vez que existiram momentos de conflito entre elementos do mesmo grupo. A minha intervenção foi em conciliar a existência de ideias opostas dentro do grupo e na divisão de tarefas na realização do trabalho. Para além disso, foi necessário gerir e motivar a participação de alguns alunos, uma vez que não conseguiam participar oralmente na transmissão das suas ideias porque as suas opiniões não eram valorizados pelos colegas de turma.

Apesar de não poder afirmar que a utilização da Robótica Educativa, utilizando-se robôs da Lego® Mindstorms® NXT 2.0, contribui para a consolidação de conceitos de programação, considero que poderá ser uma alternativa para o ensino de alguns conceitos de programação e a alunos que estejam a iniciar-se na programação. Para além disso, a sua utilização em conjunto com a Aprendizagem Baseada em Problemas, permitem contextualizar as aprendizagens dos alunos, principalmente em alunos de cursos profissionais.

Como professor e considerando que não existem alunos e turmas iguais, devo ter em consideração as especificidades de cada um na escolha de metodologias e estratégias de ensino. Para além das especificidades dos alunos e das turmas, os objetivos e as competências a desenvolver em cada disciplina ou unidade didática são diferentes. Assim, estes também devem ser fatores na definição das estratégias e na escolha das metodologias aplicar. No entanto, considero que a Robótica Educativa e Aprendizagem Baseada em Problemas têm um papel muito importante na preparação dos alunos para a sua vida profissional e também pessoal. A autonomia na aprendizagem, a reflexão sobre os conhecimentos presentes e necessidades de aprendizagem futuras, a capacidade de resolução de problemas, a capacidade de trabalhar em grupo e a capacidade de decisão com uma base democrática, permitem preparar os alunos para viver em sociedade e enfrentar os diversos problemas que daí advêm.

9. Referências

- Agência Nacional para a Qualificação. (s.d.). *Cursos Profissionais*. Obtido em 25 de Outubro de 2011, de Agência Nacional para a Qualificação:
<http://www.anq.gov.pt/default.aspx?access=1>
- Almeida, E. S., Costa, E. B., Silva, K. S., Paes, R. B., Almeida, A. A., & Braga, J. D. (2002). AMBAP: Um Ambiente de Apoio ao Aprendizado de Programação. *X Workshop sobre Educação em Computação*. Florianópolis: Anais do WEI 2002/ SBC2002.
- Angeles, J. (2007). *Fundamentals of Robotic Mechanical Systems: Theory, Methods, And Algorithms*. New York: Springer Science+Business Media.
- Bacaroglo, M. (2005). Robótica Educacional Uma metodologia educacional. Londrina, Paraná, Brasil: Universidade Estadual de Londrina.
- Barge, S. (Setembro de 2000). *Principles of Problem and Project Based Learning*. Obtido em 11 de Novembro de 2011, de Aalborg University:
[http://files.portal.aau.dk/filesharing/download?filename=aau/aau/2010/~pub/PBL_aalborg_modellen.pdf](http://files.portal.aau.dk/filesharing/download?filename=aau/aau/2010/~/pub/PBL_aalborg_modellen.pdf)
- Benitti, F. B., Vahldick, A., Urban, D. L., Krueger, M. L., & Halma, A. (2009). Experimentação com Robótica Educativa no Ensino Médio: ambiente, atividades e resultados. *WIE - XV Workshop sobre Informática na Escola*.
- Benzeltout, B., & Blanchfield, P. (2009). Computer Game to Teach Programming Constructs. *The 3rd European on Games Based Learning*, (pp. 17-23). Graz.
- Bossé, M. J., & Nandakumar, N. R. (Dezembro de 2000). Real-World Problem-Solving, Pedagogy, and Efficient Programming Algorithms in Computer Education. *SIGCSE Bulletin*, 32 (4), pp. 66-69.
- Burnett, M. M., & Ambler, A. L. (1992). A Declarative Approach to Event-Handling in Visual Programming Languages. *1992 IEEE Workshop on Visual Languages* (pp. 34-40). Seattle: IEEE.
- Camus, C., Coelho, P., & Quadrado, J. C. (2002). A New Concept on Automatic Parking of na Electric Vehicle. *6th WSEAS International Multiconference on Circuits, Systems, Communications and Computers*. Creta.
- Carter, J., & Jenkins, T. (2010). The Problems of Teaching Programming: Do They Change With Time? *11th Annual Conference of the Subject Centre for Information and Computer Sciences*. Newtownabbey: HE Academy.
- Coelho, J. (2010). Conceitos e exercícios de programação: utilizando Linguagem C. Universidade Aberta.
- Coleman, S., & Nichols, E. (2010). Enhancing Student Engagement Through Pair Programming. *11th Annual Conference of the Subject Centre for Information and Computer Sciences*. Newtownabbey: HE Academy.

- Conselho Geral. (2010). *Projecto Educativo 2010-2013*. Lisboa: Escola Secundária com 3º Ciclo Padre António Vieira.
- Coull, N. J., & Duncan, I. M. (2011). Emergent requirements for supporting introductory programming. *ITALICS*, 10 (1), 78-85.
- Department of Education and Early Childhood Development. (2004). *Principles of Learning and Teaching P-12*. Obtido em 1 de Dezembro de 2011, de Department of Education and Early Childhood Development: <http://www.education.vic.gov.au/studentlearning/teachingprinciples/principles/default.htm>
- Direcção-Geral de Formação Vocacional. (2005). Programa Componente de Formação Técnica Disciplina de Linguagens de Programação. Direcção-Geral de Formação Vocacional.
- Dunican, E. (2002). Making The Analogy: Alternative Delivery Techniques for First Year Programming Courses. *14th Workshop of the Psychology of Programming Interest Group* (pp. 89-99). Uxbridge: J. Kuljis, L. Baldwin & R. Scoble.
- EdrawSoft. (2012). *Edraw Flowchart*. Obtido em 23 de Março de 2012, de Edraw Professional Diagram Solution: <http://www.edrawsoft.com/Flowchart-Definition.php>
- Escola Secundária Padre António Vieira. (n.d.). *Rui Jervis Athouguia*. Obtido em 18 de Outubro de 2011, de Escola Secundária c/ 3º Ciclo Padre António Vieira: <http://www.espav.edu.pt/Espav2008/Conteudos/Escola/Arquitecto/Arquitecto.htm>
- Eurostat, Instituto Nacional de Estatística & Ministérios. (14 de Dezembro de 2010). *Agregados domésticos privados com computador, com ligação à Internet em casa e com ligação à Internet através de banda larga (%) na Europa*. Obtido em 17 de Janeiro de 2012, de PORDATA: <http://www.pordata.pt/Europa/Agregados+domesticos+privados+com+computador++com+ligacao+a+Internet+em+casa+e+com+ligacao+a+Internet+atraves+de+banda+larga+%28percentagem%29-1494>
- Eurostat, Instituto Nacional de Estatística & Ministérios. (14 de Dezembro de 2010). *Agregados domésticos privados com computador, com ligação à Internet em casa e com ligação à Internet através de banda larga (%) na Europa*. Obtido em 17 de Janeiro de 2012, de PORDATA: <http://www.pordata.pt/Europa/Agregados+domesticos+privados+com+computador++com+ligacao+a+Internet+em+casa+e+com+ligacao+a+Internet+atraves+de+banda+larga+%28percentagem%29-1494>
- Fernandes, S. R. (Setembro de 2010). Aprendizagem baseada em projectos no contexto do ensino superior: avaliação de um dispositivo pedagógico no ensino de engenharia. Braga: Universidade do Minho.
- Friendly, L. (1995). The Design of Distributed Hyperlinked Programming Documentation. *International Workshop on Hypermedia Design '95*. Sun Microsystems, Inc.

- Gabini, W., & Diniz, R. (2009). Os professores de química e o uso do computador em sala de aula: discussão de um processo de formação continuada. *Ciência & Educação*, 15 (2), 343-358.
- Goel, S., & Kathuria, V. (2010). A Novel Approach for Collaborative Pair Programming. *Journal of Information Technology Education*, 9, 183-196.
- Gomes, A., & Mendes, A. J. (2007). Learning to program - difficulties and solutions. *International Conference on Engineering Education – ICEE 2007*. Coimbra.
- 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*, 1 (1), 93-103.
- grupo 550. (2011). Critérios Específicos de Avaliação das Disciplinas de Informática. Lisboa: Escola Secundária Padre António Vieira.
- Gudwin, R. R. (1997). Linguagens de Programação. Brasil: Universidade Estadual de Campinas.
- Guerreiro, P. (1986). A mesma velha questão: como ensinar Programação? *Memoria do Quinto Congreso Iberoamericano de Educación Superior en Computación*. Mexico City: UNAM.
- Hennessy, J. L., & Patterson, D. A. (2012). *Computer Architecture: A Quantitative Approach*. Waltham, USA: Elsevier Inc.
- IBM. (Março de 1970). Flowcharting Techniques. *Data Processing Techniques*. White Plains, New York, United States of America: IBM, Technical Publications Department.
- Instituto Nacional de Estatística. (27 de Dezembro de 2011). *Indivíduos que utilizam computador e Internet em % do total de indivíduos: por nível de escolaridade mais elevado completo - Portugal*. Obtido em 17 de Janeiro de 2012, de PORDATA:
<http://www.pordata.pt/Portugal/Indiv%C3%BAduos+que+utilizam+computador+e+Internet+em+percentagem+do+total+de+indiv%C3%BAduos+por+n%C3%ADvel+de+escolaridade+mais+elevado+completo-1141>
- Instituto Nacional de Estatística. (27 de Dezembro de 2011). *Indivíduos que utilizam computador e Internet em % do total de indivíduos: por nível de escolaridade mais elevado completo - Portugal*. Obtido em 17 de Janeiro de 2012, de PORDATA:
<http://www.pordata.pt/Portugal/Indiv%C3%BAduos+que+utilizam+computador+e+Internet+em+percentagem+do+total+de+indiv%C3%BAduos+por+n%C3%ADvel+de+escolaridade+mais+elevado+completo-1141>
- Jenkins, T. (2002). On the Difficulty of Learning to Program. *3rd Annual LTSN-ICS Conference* (pp. 53-58). Loughborough: LTSN Centre for Information and Computer Sciences.
- Jenkins, T. (2001). Teaching Programming – A Journey from Teacher to Motivator. *2nd Annual LTSN-ICS Conference*. London: LTSN Centre for Information and Computer Sciences.

- Kanakadoss, S. (2005). *SIMBUILDER: AN INVESTIGATION AND USABILITY STUDY OF NOVICE PROGRAMMING TECHNIQUES*. Alabama: Auburn University.
- Keckler, S. W., Dally, W. J., Chang, A., Lee, W. S., & Chatterjee, S. (1999). Concurrent Event Handling Through Multithreading. *IEEE Transactions on Computers* , 48 (9), 903-916.
- Kelly, J. F. (2007). Lego Mindstorms NXT-G Programming Guide. *apress* .
- Kolmos, A., Kuru, S., Hansen, H., Eskil, T., Podesta, L., Fink, F., et al. (2007). *Problem Based Learning*. TREE – Teaching and Research in Engineering in Europe.
- Lencastre, J. A., Monteiro, A., Silva, A., Cardoso, N., & Machado, P. (2011). Aprendizagem Baseada em Problemas e Objectos de Aprendizagem. *XI Congreso Internacional Galego-Portugués de Psicopedagogía* (pp. 1629-1640). Coruña: Universidade da Coruña.
- Luz, S. C. (2011). A Influência do Feedback no Processo Auto-Regulatório da Aprendizagem em Estudantes do Ensino Superior. Lisboa, Portugal: Universidade de Lisboa.
- Major, L., Kyriacou, T., & Brereton, P. (2011). Experiences of Prospective High School Teachers Using a Programming Teaching Tool. *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, (pp. 126-131). Koli National Park.
- Mendes, A. J. (2002). *Software educativo para apoio à aprendizagem de programação*. Obtido em 10 de Abril de 2012, de Universidad de Chile: http://www.c5.cl/ieinvestiga/actas/tise01/pags/charlas/charla_mendes.htm
- Microsoft. (s.d.). *C# Programming Guide*. Obtido em 23 de Março de 2012, de msdn: <http://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>
- Ministério da Educação. (2 de Janeiro de 1998). Decreto-Lei n.º 1/98 de 2 de Janeiro. *Diário da República - I Série-A* (1), pp. 2-28.
- Ministério da Educação. (2005). Portaria n.º 913/2005. *Diário da República - I Série-B* (185), 5806-5807.
- Moreira, M. A. (2005). “Na Sombra das Maiorias Silenciosas”: por uma Educação Autêntica e Transformadora. *Currículo sem Fronteiras* , 5 (1), 70-95.
- Motil, J., & Epstein, D. (2000). *JJ: A language designed for beginners (less is more)*. Obtido em 24 de Dezembro de 2011, de Step into Java: www.publicstaticvoidmain.com/JJ_A_Language_Designed_For_Beginners_LessIsMore.pdf
- Myers, B. (Janeiro de 1998). Natural Programming: Project Overview and Proposal. Institute for Software Research.
- Oracle. (s.d.). *How to Write Doc Comments for the Javadoc Tool*. Obtido em 23 de Março de 2012, de Oracle:

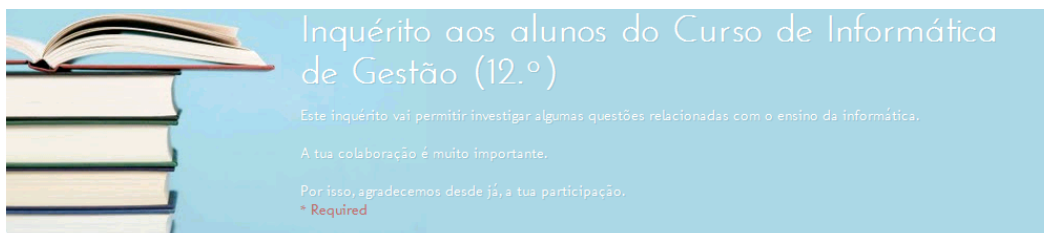
<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html#format>

- Páztor, A., Pap-Szigetu, R., & Torok, E. L. (2010). Effects of Using Model Robots in the Education of Programming. *Informatics in Education* , 9 (1), 133-140.
- Parque Escolar. (s.d.). *Escola Secundária de Padre António Vieira*. Obtido em 20 de Outubro de 2011, de *parqueEscolar*: <http://www.parque-escolar.pt/escola-secundaria-padre-antonio-vieira.php>
- Pimentel, E. P., França, V. F., & Omar, N. (2003). A caminho de um ambiente de avaliação e acompanhamento contínuo da aprendizagem em Programação de Computadores. *II Workshop de Educação em Computação e Informática do Estado de Minas Gerais*. Poços de Caldas.
- Pinheiro, M. M., Sarrico, C. S., & Santiago, R. A. (2010). Altos e Baixos de um Casamento: PBL e Motivações dos Alunos. *Conectando Pessoas, Ideias e Comunidades*. São Paulo: Congresso Internacional PBL2010.
- Preston, D. (2006). Adapting pair programming pedagogy for use in computer literacy courses. *Computing Sciences in Colleges* , 21 (5), 84-93.
- Rocha, R. (2006). Utilização da robótica pedagógica no processo de ensino-aprendizagem de programação de computadores. Belo Horizonte, Brasil: Centro Federal e Educação Tecnológica de Minas Gerais - CEFET.
- Rodrigues, M. d., & Mata, J. T. (Setembro de 2003). A utilização de computador e da internet pela população portuguesa. *Sociologia, Problemas e Práticas* , 43 , 161-178.
- Roldão, M. C. (2009). *Estratégias de ensino - O saber e o agir do professor*. Vila Nova de Gaia: Fundação Manuel Leão.
- Ryu, Y.-W., Kim, S.-Y., & Oh, S.-Y. (2008). Robust Automatic Parking without Odometry using an Evolutionary Fuzzy Logic Controller. *International Journal of Control, Automation, and Systems* , 6 (3), 434-443.
- Saeli, M., Perrenet, J., Jochems, W. M., & Zwaneveld, B. (2011). Teaching Programming in Secondary School: A Pedagogical Content Knowledge Perspective. *Informatics in Education* , 10 (1), pp. 73-88.
- Samek, M. (2009). *Practical UML statecharts in C/C++: event-driven programming for embedded systems*. Oxford, UK: Elsevier Inc.
- Santos, L. (2002). Auto-avaliação regulada: porquê, o quê e como? In P. Abrantes, & F. Araújo, *Avaliação das aprendizagens* (pp. 75-84). Lisboa: Ministério de Educação. Departamento do Ensino Básico.
- Santos, R. P., & Costa, H. A. (2006). Análise de Metodologias e Ambientes de Ensino para Algoritmos, Estruturas de Dados e Programação aos iniciantes em Computação e Informática. *INFOCOMP Journal of Computer Science* , 5 (1), 41-50.

- Sartzemi, M., Dagdilelis, V., & Kagani, K. (2008). Teaching Introductory Programming Concepts with Lego MindStorms in Greek High Schools: A Two-Year Experience. In Y. Takahashi (Ed.), *Service Robot Applications* (pp. 343-368). InTech.
- Serrazina, M. d. (Junho de 2009). O programa de formação contínua em matemática para professores do 1º e 2º ciclo do ensino básico. *Interacções* (12), pp. 4-22.
- Sloane, K. D., & Linn, M. C. (1988). Instructional Conditions in Pascal Programming Classes. In R. E. Mayer, *Teaching and learning computer programming: multiple research perspectives* (pp. 207-232). Hillsdale, New Jersey: Lawrence Erlbaum Associates, Inc.
- Sobral, S. C. (Março de 2008). Tese de Doutoramento em Tecnologias e Sistemas de Informação. *B-Learning em disciplinas introdutórias de programação*. Universidade do Minho.
- Sockalingam, N., Rotgans, J., & Schmidt, H. G. (2011). Student and tutor perceptions on attributes of effective problems in problem-based learning. *Higher Education*, 62 (1), 1-16.
- Szádeczky-Kardoss, E., Kiss, B., & Wahl, I. (2008). Design of a Semi-Autonomous Parking Assist System. *The European Control Conference 2009*.
- The LEGO Group. (2012). *Mindstorms*. Obtido em 16 de Abril de 2012, de Lego: <http://mindstorms.lego.com/en-us/whatisnxt/default.aspx>
- United States Department of Labor. (17 de Setembro de 2009). *Computer Software Engineers and Computer Programmers*. Obtido em 19 de Janeiro de 2012, de Bureau of Labor Statistics: <http://www.bls.gov/oco/ocos303.htm>
- W3C. (15 de Março de 2012). *HTML: The Markup Language*. Obtido em 23 de Março de 2012, de W3C: <http://www.w3.org/TR/2012/WD-html-markup-20120315/>
- Wu, C.-C., Tseng, I.-C., & Huang, S.-L. (2008). Visualisation of program behaviors: physical robots versus robot simulators. In R. T. Mittermeir, & M. M. Syslo (Edits.), *Informatics Education – Supporting Computational Thinking. Proceedings of the Third Conference on Informatics in Secondary Schools – Evolution and Perspectives* (pp. 53-62). Torun, Polónia: Springer.
- Xu, J., Chen, G., & Xie, M. (2000). Vision-Guided Automatic Parking for Smart Car. *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 Cat No00TH8511* (pp. 725-730). Dearborn: IEEE.
- Zhao, Y., & Collins, E. G. (2005). Robust Automatic Parallel Parking in Tight Spaces via Fuzzy Logic. *Robotics and Autonomous System*, 51 (2-3), 111-127.

10. Anexos

Anexo A – Inquérito de Caracterização da Turma



Inquérito aos alunos do Curso de Informática de Gestão (12.º)

Este inquérito vai permitir investigar algumas questões relacionadas com o ensino da informática.

A tua colaboração é muito importante.

Por isso, agradecemos desde já, a tua participação.

*** Required**

1. Quantos anos tens? *

2. Tens computador em casa? *

- Sim
 Não

3. Tens Internet em casa? *

- Sim
 Não

4. Qual o motivo que te levou a escolher o curso que frequentas? *

	Nada importante	Pouco importante	Importante	Muito importante
Gosto pela informática	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gosto em jogar em PCs e consolas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gosto em usar e conhecer novas tecnologias	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Será útil no meu futuro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gosto pela programação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Quero trabalhar em informática	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gosto por gestão e contabilidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Outro motivo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

5. Se indicaste "Outro motivo" na questão anterior, diz qual ou quais foram.

6. O curso está a ir de acordo com o que estavas à espera? *

- Sim
 Não

7. Diz-nos de que forma gostas do curso: *

	Não gosto	Gosto Pouco	Gosto Muito	Gosto Bastante
Da escola	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dos professores	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dos colegas da turma	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Dos colegas da escola	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Das disciplinas de informática	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Das disciplinas que não são de informática	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Localização da escola	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Do horário das aulas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Outro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

8. Se indicaste "Outro" na questão anterior, diz qual ou quais são.

9. Em relação às seguintes disciplinas, indica o teu nível de satisfação em cada uma. *

	Nada Satisfeito	Pouco Satisfeito	Satisfeito	Muito Satisfeito
Linguagens da Programação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sistemas de informação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Organização de Empresas e Aplicações de Gestão	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aplicações Informáticas e Sistemas de Exploração	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Economia	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Matemática	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Português	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Língua Estrangeira (Inglês)	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Área de Integração	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Educação Física	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Tecnologias de Informação e Comunicação	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

10. Se mudasses alguma coisa numa disciplina, para gostares mais dela, o que seria? *

11. Depois de entrares no curso, o teu gosto pela informática... *

- Diminui
 Manteve-se
 Aumentou

12. O que gostarias de aprender nas aulas de informática e que até agora não foi abordado? *

13. Quantas vezes é que utilizas as seguintes tecnologias de Informação e Comunicação/Plataformas? *

	Nunca	Raramente (1 vez por mês)	Algumas vezes (1 a 2 vezes)	Frequentemente (3 a 5 vezes)	Todos os dias
Televisão	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Telemóvel	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Computador	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Consola de jogos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Robô NXT da LEGO	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Rádio / Leitor de MP3 ou MP4	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Moodle	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Redes sociais, como o Facebook	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Email	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Blog	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Wiki	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Foruns	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Chat, como o Messenger	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Outro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

14. Se indicaste "Outro" na questão anterior, diz qual ou quais são.

15. Como é que são os teus conhecimentos em informática? *

- Fracos
- Suficientes
- Bons
- Muito Bons

16. Como é que classificas os teus conhecimentos nas seguintes linguagens? *

	Nenhum	Básico	Bom	Muito Bom
Visual Basic	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PASCAL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
C++	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
JAVA	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SQL	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Javascript	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
HTML	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
PHP	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Outra	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

17. Se indicaste "Outra" na questão anterior, diz qual ou quais são.

18. O que pensas fazer depois de terminares o curso? *

(Podes seleccionar várias opções)

- Trabalhar na área da informática
- Trabalhar mas não na área de informática
- Prosseguir os estudos na área de informática
- Prosseguir os estudos mas não na área de informática
- Outro

19. Se indicaste "Outro" na questão anterior, diz qual ou quais são.

20. Como é que gostas mais de trabalhar? *

- Sozinho
- Apenas com 1 colega
- Em grupo, com 2 ou mais colegas

21. Nas aulas, quando tens uma dificuldade o que fazes? *

(Coloca, por ordem, o que costumás fazer)

	1.º Lugar	2.º Lugar	3.º Lugar	4.º Lugar	5.º Lugar
Peço ajuda ao professor	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Peço ajuda aos colegas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vou à procura na Internet	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Vou ver os manuais dados pelo professor	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Não peço ajuda	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Outro	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

22. Se indicaste "Outro" na questão anterior, diz qual ou quais são.**23. Em relação ao modo como és avaliado, preferes: ***

	Não gosto	Gosto
Trabalhos de grupo	<input type="radio"/>	<input type="radio"/>
Fichas formativas	<input type="radio"/>	<input type="radio"/>
Teste teórico	<input type="radio"/>	<input type="radio"/>
Teste prático	<input type="radio"/>	<input type="radio"/>
Teste Teórico-Prático	<input type="radio"/>	<input type="radio"/>
Atitudes (assiduidade, pontualidade, autonomia, responsabilidade, ...)	<input type="radio"/>	<input type="radio"/>
Capacidades (destreza na utilização do equipamento e do Software, utilização do vocabulário técnico, resolução de problemas,...)	<input type="radio"/>	<input type="radio"/>
Outro	<input type="radio"/>	<input type="radio"/>

24. Se indicaste "Outro" na questão anterior, diz qual ou quais são.**25. Quando estudas FORA da ESCOLA, diz-nos quando o fazes: ***

- Não estudo fora da escola
- Antes de uma avaliação, como por exemplo, um teste
- Uma vez por semana
- Quando tenho dificuldades
- 2 a 3 vezes por semana
- Todos os dias apesar de não ter dificuldades, nem ter avaliações

26. Com quem é que costumás estudar? *

- Com ninguém, gosto de estudar sozinho
- Com o(s) colega(s) que costumo fazer grupo nas aulas
- Com um grupo de amigos
- Não estudo

Anexo B – Planos de Aula

<p>Aula n.º: 1</p> <p>Tempo letivo: 90 minutos</p>	<p>Módulo: 17 - Desenvolvimento de uma aplicação de Gestão</p> <p>Disciplina: Linguagens de Programação</p> <p>Curso: Informática de Gestão</p>
<p>Objetivos:</p> <p>No final da aula, o aluno é capaz de:</p> <ul style="list-style-type: none">• Identificar alguns dos seus pontos fortes e fracos na programação;• Indicar o que vai ser realizado nas aulas seguintes.	<p>Conteúdos temáticos presentes:</p> <ul style="list-style-type: none">• Lógica de programação• Sequência lógica• Instruções• Algoritmos• Fluxogramas• Comentários• Variáveis• Operadores e expressões relacionais• Estruturas de decisão e de repetição• Eventos• Instruções de entrada e de saída• Teste, deteção e correção de erros

Atividades	Duração (minutos)	Metodologia	Instrumentos de avaliação	Recursos
Apresentação dos objectivos da aula e das aulas seguintes	20	Apresentação visual e oral, dos objetivos da aula. Apresentação visual e oral, dos objetivos das aulas seguintes.	Grelha de registos	<ul style="list-style-type: none"> • Computador • Videoprojector • Programa de apresentações
Realização de um autodiagnóstico	10	Preenchimento de um questionário, para autodiagnóstico dos conhecimentos de programação. Permite uma reflexão critica do aluno.	Questionário	<ul style="list-style-type: none"> • Computador • Internet
Realização de um teste de diagnóstico	60	Respostas a questões com vários níveis de dificuldade, para diagnóstico dos conhecimentos reais de programação. Permite uma reflexão critica do aluno, entre as respostas às questões e a atividade anterior.	Ficheiros produzidos	<ul style="list-style-type: none"> • Computador • Internet • Editor de Visual Basic • Editor de fluxogramas

<p>Aula n.º: 2</p> <p>Tempo letivo: 90 minutos</p>	<p>Módulo: 17 - Desenvolvimento de uma aplicação de Gestão</p> <p>Disciplina: Linguagens de Programação</p> <p>Curso: Informática de Gestão</p>
<p>Objetivos:</p> <p>No final da aula, o aluno é capaz de:</p> <ul style="list-style-type: none"> • Relacionar os seus conhecimentos/experiências pessoais com novos conhecimentos; • Refletir sobre um problema e a sua solução; • Divulgar possíveis soluções de um problema; • Escrever possíveis soluções de um problema, utilizando a sua língua natural. 	<p>Conteúdos temáticos presentes:</p> <ul style="list-style-type: none"> • Lógica de programação • Sequência lógica • Instruções • Algoritmos • Fluxogramas • Comentários • Estruturas de decisão e de repetição • Instruções de entrada e de saída • Teste, deteção e correção de erros

Atividades	Duração (minutos)	Metodologia	Instrumentos de avaliação	Recursos
Apresentação dos objectivos da aula	5	Apresentação visual e oral dos objetivos da aula.	Grelha de registos	<ul style="list-style-type: none"> • Computador • Videoprojector • Programa de apresentações
Enquadramento e apresentação do problema	10	<p>Apresentação oral do problema “como é que um automóvel consegue estacionar de uma forma automática, sem o auxílio do condutor?”.</p> <p>Enquadramento com as necessidades das empresas integrarem os Sistemas de Informação, na sua organização.</p> <p>Relacionar com o estágio, sendo este um possível problema que poderão ter que resolver.</p>	Grelha de registos	<ul style="list-style-type: none"> • Computador • Videoprojector • Programa de apresentações
Visionamento de um vídeo	5	Apresentação visual de um caso real de estacionamento automático de automóveis.	Grelha de registos	<ul style="list-style-type: none"> • Computador • Videoprojector • Internet
Debate de ideias	15	<p>Participação voluntária dos alunos, na troca de ideias e experiências dos alunos, que possam contribuir para a solução do problema.</p> <p>Podem exemplificar usando um modelo de um parque de estacionamento e automóveis em miniaturas, para poderem simular o estacionamento e</p>	Grelha de registos	<ul style="list-style-type: none"> • Modelo de um parque de estacionamento. • Lego NXT • Mindstorm

		desenvolver a autonomia. No quadro, será indicado uma solução para cada tipo de estacionamento. Serão colocadas questões orientadoras aos alunos.		<ul style="list-style-type: none"> • Quadro branco
Registo e reflexão de possíveis soluções	20	<p>Cada grupo ocupará um computador e acederão ao site PBworks.</p> <p>O professor irá demonstrar como se acede ao PBworks e se utilizam as principais funcionalidades. Com isso, irá ser criado uma página do seu grupo e uma página para a descrição das soluções.</p> <p>Cada grupo irá escrever os passos das suas soluções, registando o trabalho desenvolvido. Estas soluções deverão ser escritas e expressas usando a língua natural do aluno.</p> <p>Os alunos como apoio, poderão simular as soluções descritas no modelo do parque de estacionamento e desenvolver a autonomia.</p> <p>Se for necessário, os grupos poderão alterar as soluções encontradas, alterando simultaneamente a página criada.</p> <p>Serão colocadas questões orientadoras aos alunos.</p> <p>O professor será o mediador do trabalho desenvolvido em grupo.</p>	<p>Grelha de registos</p> <p>Páginas criadas no PBworks</p>	<ul style="list-style-type: none"> • Computador • Videoprojector • Internet • PBworks • Modelo de um parque de estacionamento • Lego NXT • Mindstorm
Introdução ao software LEGO NXT	35	Como introdução ao software a utilizar na solução do problema, será feito um exercício, onde será programado o robô para andar em frente. Este exercício, permite ser o ponto de partida para a concretização da solução ao problema.	<p>Grelha de registos</p> <p>Ficheiro produzido do exercício</p>	<ul style="list-style-type: none"> • Computador • Videoprojector • Internet • Modelo de um parque

		Cada grupo deverá resolver o exercício autonomamente. O professor colocará questões orientadores e será o mediador do trabalho desenvolvido em grupo.		de estacionamento • Lego NXT Mindstorm
--	--	--	--	--

<p>Aula n.º: 3</p> <p>Tempo letivo: 90 minutos</p>	<p>Módulo: 17 - Desenvolvimento de uma aplicação de Gestão</p> <p>Disciplina: Linguagens de Programação</p> <p>Curso: Informática de Gestão</p>
<p>Objetivos:</p> <p>No final da aula, o aluno é capaz de:</p> <ul style="list-style-type: none"> • Relacionar os seus conhecimentos/experiências pessoais com novos conhecimentos; • Refletir sobre um problema e a sua solução; • Escrever possíveis soluções de um problema, utilizando a sua língua natural; • Transpor a descrição, escrita em língua natural, de uma solução para a programação dessa solução. 	<p>Conteúdos temáticos presentes:</p> <ul style="list-style-type: none"> • Lógica de programação • Sequência lógica • Instruções • Algoritmos • Fluxogramas • Comentários • Variáveis • Operadores e expressões relacionais • Estruturas de decisão e de repetição • Eventos • Instruções de entrada e de saída • Teste, deteção e correção de erros

Atividades	Duração (minutos)	Metodologia	Instrumentos de avaliação	Recursos
Desenvolvimento da solução	90	<p>Utilizando as soluções encontradas e descritas na página criada, na plataforma PBworks, os grupos irão programar os robôs. Sendo assim, os grupos, programarão as soluções e as testarão, usando o modelo do parque de estacionamento. Durante esse desenvolvimento e testes, os grupos poderão alterar a descrição das soluções, colocadas na plataforma PBworks.</p> <p>Cada grupo deverá resolver o exercício autonomamente.</p> <p>O professor colocará questões orientadores e será o mediador do trabalho desenvolvido em grupo.</p>	<p>Grelha de registos</p> <p>Páginas criadas no PBworks</p>	<ul style="list-style-type: none"> • Computador • Internet • PBworks • Modelo de um parque de estacionamento • Lego NXT • Mindstorm

<p>Aula n.º: 4</p> <p>Tempo letivo: 90 minutos</p>	<p>Módulo: 17 - Desenvolvimento de uma aplicação de Gestão</p> <p>Disciplina: Linguagens de Programação</p> <p>Curso: Informática de Gestão</p>
<p>Objetivos:</p> <p>No final da aula, o aluno é capaz de:</p> <ul style="list-style-type: none"> • Relacionar os seus conhecimentos/experiências pessoais com novos conhecimentos; • Refletir sobre um problema e a sua solução; • Escrever possíveis soluções de um problema, utilizando a sua língua natural; • Transpor a descrição, escrita em língua natural, de uma solução para a programação dessa solução. 	<p>Conteúdos temáticos presentes:</p> <ul style="list-style-type: none"> • Lógica de programação • Sequência lógica • Instruções • Algoritmos • Fluxogramas • Comentários • Variáveis • Operadores e expressões relacionais • Estruturas de decisão e de repetição • Eventos • Instruções de entrada e de saída • Teste, deteção e correção de erros

Atividades	Duração	Metodologia	Instrumentos de avaliação	Recursos
Desenvolvimento da solução	85	Utilizando as soluções encontradas e descritas na página criada, na plataforma PBworks, os grupos irão programar os robôs. Sendo assim, os grupos, programarão as soluções e as testarão, usando o modelo do parque de estacionamento. Durante esse desenvolvimento e testes, os grupos poderão alterar a descrição das soluções, colocadas na plataforma PBworks. Cada grupo deverá resolver o exercício autonomamente. O professor colocará questões orientadores e será o mediador do trabalho desenvolvido em grupo.	Grelha de registos Páginas criadas no PBworks	<ul style="list-style-type: none"> • Computador • Internet • PBworks • Modelo de um parque de estacionamento • Lego NXT Mindstorm
Divulgação da solução	5	Os alunos irão disponibilizar, na plataforma PBworks, o(s) programa(s) criado(s) que traduz(em) a solução do problema.	Páginas criadas no PBworks Ficheiros produzidos	<ul style="list-style-type: none"> • Computador • Internet • PBworks • Modelo de um parque de estacionamento • Lego NXT Mindstorm

<p>Aula n.º: 5</p> <p>Tempo letivo: 90 minutos</p>	<p>Módulo: 17 - Desenvolvimento de uma aplicação de Gestão</p> <p>Disciplina: Linguagens de Programação</p> <p>Curso: Informática de Gestão</p>
<p>Objetivos:</p> <p>No final da aula, o aluno é capaz de:</p> <ul style="list-style-type: none"> • Refletir criticamente sobre a solução encontrada; • Identificar alguns dos seus pontos fortes e fracos na programação. 	<p>Conteúdos temáticos presentes:</p> <ul style="list-style-type: none"> • Lógica de programação • Sequência lógica • Instruções • Algoritmos • Fluxogramas • Comentários • Variáveis • Operadores e expressões relacionais • Estruturas de decisão e de repetição • Eventos • Instruções de entrada e de saída • Teste, deteção e correção de erros

Atividades	Duração	Metodologia	Instrumentos de avaliação	Recursos
Demonstração à turma	20	Divulgação e apresentação da solução de cada grupo à turma. Permite verificar, qual a melhor solução que foi desenvolvida, verificando se foram cumpridas as regras obrigatórias e divulgadas quando se colocou o problema à turma.	Grelha de registos	<ul style="list-style-type: none"> • Modelo de um parque de estacionamento • Lego NXT Mindstorm
Realização de uma autoavaliação	10	Preenchimento de um questionário, para autoavaliação dos conhecimentos de programação. Permite uma reflexão crítica do aluno.	Questionário	<ul style="list-style-type: none"> • Computador • Internet
Realização de um teste de avaliação	60	Respostas a questões com vários níveis de dificuldade, para avaliação dos conhecimentos reais de programação. Permite uma reflexão crítica do aluno, entre as respostas às questões e a atividade anterior.	Ficheiros produzidos	<ul style="list-style-type: none"> • Computador • Internet • Editor de Visual Basic • Editor de fluxogramas

Anexo C – Autorizações para a Recolha de Dados

Pedido à Direção da Escola

Exmo. Senhor(a) Diretor(a) da
<nome da escola>

<nome do mestrando>, mestrando da Universidade de Lisboa, integrados nesta escola, orientados pelo professor <nome do orientador> e contando com a cooperação do(s) professore(s) <nome do professor cooperante>, vêm requer a V. Ex.^a autorização para procederem à recolha de dados, no âmbito da intervenção relacionada com a lecionação das aulas a serem realizadas ao longo de um máximo de <nº de dias> dias, com a <instrumentos de recolha de dados>, à turma <nome da turma> do <ano de escolaridade>.º ano do curso profissional de <nome do curso>.

A intervenção visa a elaboração do relatório no âmbito da unidade curricular de Iniciação à Prática Profissional III e IV, pertencente ao ciclo de estudos conducentes ao grau de Mestre em Ensino da Informática da Universidade de Lisboa. Os dados recolhidos terão um carácter **confidencial**, servindo apenas para a fundamentação da parte empírica do trabalho e serão salvaguardadas todas as **questões éticas e legais** de recolha de dados.

Oportunamente, será informado o respetivo Diretor de Turma e de Curso e será solicitado aos Encarregados de Educação as devidas autorizações para a participação dos seus educandos neste estudo.

Lisboa, <dia> de <mês> de <ano>.

Pede deferimento,

(<nome do mestrando>)

Informação ao Diretor(a) de Curso e Diretor(a) de Turma

Exmo(s). Senhor(s)

Diretor(a) de Curso e

Diretor(a) de Turma <nome da turma>

<nome do mestrando>, mestrando da Universidade de Lisboa, integrados nesta escola e orientados pelo professor <nome do orientador>, procederei a uma intervenção relacionada com a lecionação das aulas ao longo de um máximo de <nº de dias> dias, à turma <nome da turma> do <ano de escolaridade>.º ano do curso profissional de <nome do curso>.

A intervenção visa a elaboração do relatório referente à unidade curricular de Iniciação à Prática Profissional III e IV, pertencente ao ciclo de estudos conducentes ao grau de Mestre em Ensino da Informática da Universidade de Lisboa.

Para a realização da intervenção, precisamos do contributo dos alunos da turma <nome da turma>, para recolher dados ao longo das aulas intervencionadas com gravação de áudio e vídeo. Os dados recolhidos terão um carácter **confidencial**, servindo apenas para a fundamentação da parte empírica do trabalho, **pelo que não serão divulgados**.

Por esse motivo, vimos informar V. Exa. que iremos proceder à recolha dos dados acima referidos durante as aulas de <nome da disciplina> leccionadas pelo professor cooperante <nome do professor cooperante>.

Informamos ainda, que já foi pedido a autorização à Direção desta Escola.

Agradecemos, desde já, a atenção dispensada.

Com os melhores cumprimentos,

(<nome do mestrando>)

Pedido de autorização aos Encarregados de Educação

Exmo(a). Senhor(a) Encarregado(a) de Educação

No âmbito do ciclo de estudos conducentes ao grau de Mestre em Ensino da Informática da Universidade de Lisboa, pretendemos proceder à recolha de dados com <instrumentos de recolha de dados>, ao longo de um máximo de <nº de dias> dias, à turma <nome da turma> do <ano de escolaridade>.º ano do curso profissional de <nome do curso>.

A intervenção tem em conta as dificuldades dos alunos na programação e pretende perspetivar possíveis soluções que promovam melhores aprendizagens.

Por esse motivo, vimos informar V. Exa. que iremos proceder à recolha dos dados acima referidos durante as aulas de Linguagens de Programação. Os dados recolhidos terão um carácter **confidencial**, servindo apenas para a fundamentação da parte empírica da dissertação, pelo que **não serão divulgados**.

Informamos ainda, que já foi pedido a autorização à Direção desta Escola.

Agradecemos, desde já, a atenção dispensada.

Com os melhores cumprimentos,

(<nome do mestrando>)

Lisboa, <dia> de <mês> de <ano>.

Tomei conhecimento:

O(A) Diretor(a)

Diretor(a) de Curso



Eu, _____ Encarregado(a) de
Educação do(a) aluno(a) _____, n.º
____, da turma __, do __º ano, autorizo o meu educando a contribuir com a sua
participação para o trabalho de investigação do mestrando <nome do mestrando>.

Lisboa, __ de <mês> de <ano>.

Assinatura do Encarregado de Educação

Anexo D – Plataforma PBworks

Estacionamento automático

O estacionamento automático de automóveis, sem a ajuda do condutor, está a ser estudado por muitas empresas, nomeadamente a Volkswagen, que possui protótipos de automóveis que já estacionam automaticamente.



Neste espaço de trabalho, estão as soluções encontradas para o estacionamento automático de automóveis. Cada grupo de alunos criou uma página, com a descrição das soluções que encontrou para o estacionamento automático em 3 tipos de estacionamento:

- em espinha (nível 1)
- em paralelo (nível 2)
- em linha (nível 3)


Para além das descrições, os grupos disponibilizam os ficheiros com a programação realizada de robots da Lego NXT Mindstorm.


Soluções


Aqui encontram-se links para as páginas dos alunos, com as soluções encontradas.

Anexo E – Plataforma Moodle

Lista de tópicos

 Notícias

1  Teste de avaliação

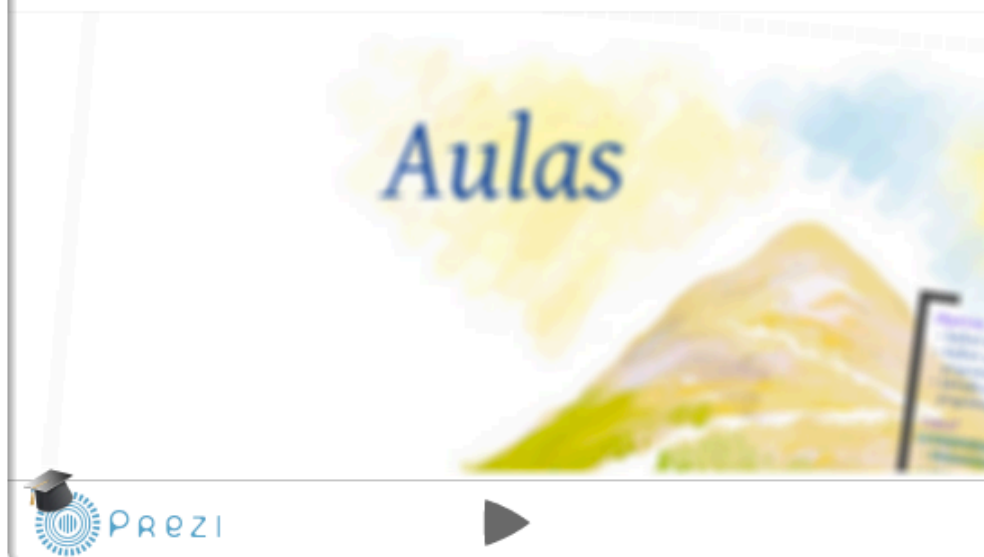
 Submissão dos ficheiros do teste de avaliação

2 <http://ig09.pbworks.com>

3

Aulas de Intervenção

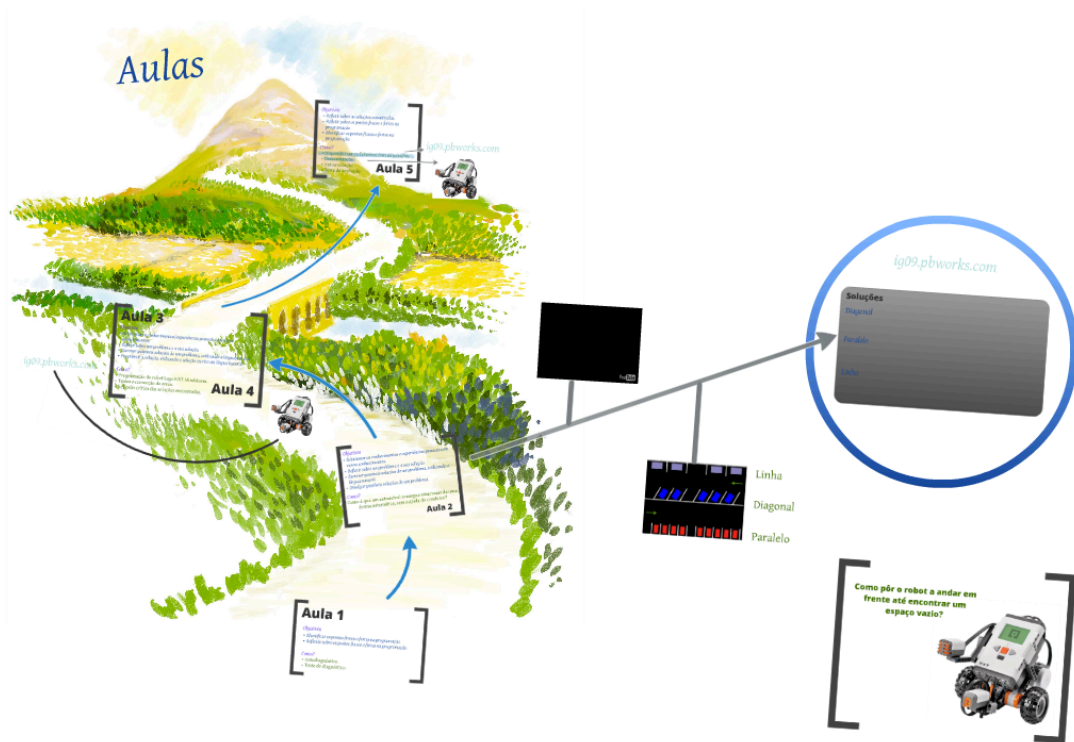
Objetivos e atividades da turma




 Teste de diagnóstico

 Submissão dos ficheiros do teste de diagnóstico

Anexo F – Apresentação das Aulas Realizada aos Alunos



Anexo G – Vídeo Demonstrativo do Estacionamento Automático



The screenshot shows a YouTube video player for a video titled "Volkswagen Park Assist Vision". The video content shows a car's perspective of a parking lot with blue sensor waves emanating from the car, indicating the automatic parking system in action. The video player includes a progress bar at 0:06 / 0:41, a play button, and a volume icon. Below the video player, there are buttons for "Like", "Share", and "Download", along with a view count of 14,588 and a like/dislike count of 9 likes, 0 dislikes.

YouTube [Browse](#) | [Upload](#) [Create Account](#) | [Sign In](#)

Volkswagen "Park Assist Vision"

[cabralreed](#) 96 videos

0:06 / 0:41

Like Share Download 14,588 9 likes, 0 dislikes

Uploaded by [cabralreed](#) on Apr 22, 2008
Nova tecnologia de auxílio a estacionamento automático da VW.
Fonte: www.volkswagen.com

- Volkswagen Touran Animation Park Assist**
by thecarfanatic
6,726 views
3:48
- Park Assist em ação no Volkswagen Tiguan**
by mecanicaonline
32,043 views
3:07
- Park Assist na VW Golf 6**
by elena290185
29,479 views
2:55
- ESTACIONAMENTO**
by TransportesOnline
17,532 views
4:05
- Audi parallel park assist**
by mattyximes
15,861 views
0:36
- Mercedes-Benz new-generation**
by MBHKMBHK
47,905 views
1:21

Anexo H – Questionário

Questionário de autodiagnóstico

Este questionário tem como objetivo refletires sobre os teus conhecimentos de programação. Para isso, terás que classificar os teus conhecimentos sobre conceitos de programação e classificar os teus comportamentos em algumas situações.

* Required

Insero o teu número de aluno. *

Insero o teu primeiro e último nome. *

Classifica os teus conhecimentos sobre conceitos de programação. *

Terás que indicar o teu grau de conhecimento de 1 a 3. Sendo que: 1 - não conheço o conceito; 2 – conheço o conceito, mas não sei aplicar na programação; 3 – conheço o conceito e sei aplicar na programação.

	1	2	3
Fluxogramas	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Comentários	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Variáveis	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Operadores e expressões relacionais	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Estruturas de decisão e repetição	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Eventos	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Instruções de entrada e de saída	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Indica com que frequência te acontece as seguintes situações. *

As situações descritas dizem respeito à lógica de programação, sequência lógica, instruções e algoritmos que aplicas no desenvolvimento de programas.

	1 (nunca)	2 (raramente)	3 (a maior parte das vezes)	4 (sempre)
O código que escrevo é confuso.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sinto dificuldades em alterar o código que escrevo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Facilmente consigo reutilizar código que escrevi em programas anteriores.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Facilmente consigo utilizar o código de outras pessoas, adaptando-o ao programa que estou a desenvolver.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Costumo ter erros de sintaxe no código que escrevo.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Consigo escrever algoritmos que respondem aos problemas que encontro no desenvolvimento de programas.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Powered by [Google Docs](#)

[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Anexo I – Teste de Diagnóstico

Teste de diagnóstico

<nome da turma>

Este teste de diagnóstico, tem como objetivo aplicares os teus conhecimentos de programação. Para além disso, serve para refletires sobre os teus conhecimentos, de acordo com as tuas respostas dadas no questionário de autodiagnóstico.

Existem dois grupos, independentes um do outro, nos quais terás que responder às questões. Para tal, utiliza a linguagem de programação *Visual Basic 6.0*, que tens utilizado nas aulas.

Grupo I

Para responderes aos itens, terás que criar um ou mais ficheiros, onde irás escrever as tuas respostas. No nome desses ficheiros, deverá ser incluído o teu número de aluno e nome.

A escola precisa de uma aplicação de gestão das turmas dos cursos profissionais. Essa aplicação terá que guardar os nomes e os números dos alunos da turma e associando-o um ao outro. Por isso, pediu aos alunos da turma *<nome da turma>*, que criassem essa aplicação. Desenvolve a programação dessa aplicação, cumprindo o que é pedido em cada um dos itens seguintes. Tem atenção ao facto de existirem pedidos em alguns itens que implicam a alteração da programação que já fizeste anteriormente.

1. Cria um botão, com o texto *inserir nome da turma*.
2. Cria um evento de clique do rato, no botão criado anteriormente.
3. Mostra uma mensagem no ecrã a pedir o nome da turma quando o evento anterior é executado.
4. Guarda o nome da turma que será inserido pelo utilizador na variável *nome_turma*.
5. Adiciona um comentário a indicar o que a variável *nome_turma* guarda.

6. Cria um fluxograma da programação desenvolvida nas alíneas anteriores, num documento à tua escolha, por exemplo do Microsoft Word. O nome do documento deve ser *fluxograma-1* e deves incluir o teu número de aluno e nome.
7. Cria uma estrutura de decisão que possua uma condição que verifica se a variável *nome_turma* é vazia, ou seja, se o nome da turma não foi preenchido. No caso da condição ser verdadeira, mostra uma mensagem de erro, caso contrário, mostra numa mensagem com o nome da turma.
8. Coloca o nome da turma visível na aplicação, num local à tua escolha.
9. Quando o ponteiro do rato ficar em cima do nome da turma deve ser, pedido ao utilizador o número de alunos da turma.
10. Cria uma função que recebe um parâmetro de entrada (número de alunos) e que verifica se o número de alunos inserido pelo utilizador é maior do que o número máximo permitido, 20, ou menor do que o número mínimo, 5. No caso de se verificar alguma destas situações é mostrada uma mensagem de erro.
11. Se o número de alunos inserido pelo utilizador for válido é mostrado esse número à frente do nome da turma.
12. Adiciona um comentário sobre a função criada.
13. Adiciona uma opção, que pode ser selecionado pelo utilizador, em que é mostrado uma lista com os números dos alunos. Por exemplo, se existirem 5 alunos é mostrada uma lista com 5 elementos.
14. Cria um fluxograma da programação necessária para colocar à frente do primeiro aluno da lista o texto *primeiro* e no último aluno o texto *último*. O fluxograma pode ser criado num documento à tua escolha, por exemplo do Microsoft Word. O nome do ficheiro deve ser *fluxograma-2* e deve incluir o teu número de aluno e nome.
15. Aplica o fluxograma que criaste anteriormente, efetuando a sua programação.

Grupo II

De acordo com o código de programação na linguagem *Visual Basic 6.0* apresentado em baixo, responde às questões.

```
Public Class TesteDiagnostico_Turno2

    Private Const num_total As Integer = 31

    Private Sub distribuirListas()

        Dim aux As Integer = 1

        While aux <= num_total

            If aux <= 15 Or aux > 30 Then

                ListBox1.Items.Add(aux)

            Else

                If aux > 15 Then

                    ListBox2.Items.Add(aux)

                End If

            End If

            aux = aux + 1

        End While

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

        distribuirListas()

    End Sub

End Class
```

1. Consegues identificar o uso de uma variável? Se sim, indica o nome e o tipo de dados dessa variável.
2. Existem comentários no código? Se sim, diz o que é indicado nesse comentário.
3. Foi criada alguma função ou procedimento? Se sim, indica quantas, quais os seus nomes e se têm parâmetros de entrada e/ou saída?
4. No código existem estruturas de decisão e repetição. Indica qual o objetivo do ciclo *While*.
5. No código encontra-se descrito um evento. Qual o nome do objeto a que é aplicado esse evento?

Anexo J – Teste de Avaliação

Teste de avaliação

<nome da turma>

Este teste tem como objetivo aplicares os teus conhecimentos de programação. Para além disso, serve para refletires sobre os teus conhecimentos, de acordo com as tuas respostas dadas no questionário de autoavaliação.

Existem dois grupos, independentes um do outro, nos quais terás que responder às questões. Para tal, utiliza a linguagem de programação *Visual Basic 6.0*, que tens utilizado nas aulas.

Grupo I

Para responderes aos itens, terás que criar um ou mais ficheiros, onde irás escrever as tuas respostas. No nome desses ficheiros, deverá ser incluído o teu número de aluno e nome.

Uma empresa de estacionamento de automóveis, sediada em Lisboa, precisa de uma aplicação de gestão dos seus parques. Essa aplicação terá que guardar o nome da empresa e o número total de parques de estacionamento. Para além disso, terá que associar um número de lugares a cada parque de estacionamento. Por isso, pediu aos alunos da turma *<nome da turma>*, que criassem essa aplicação. Desenvolve a programação dessa aplicação, cumprindo o que é pedido em cada um dos itens seguintes. Tem atenção, que existem pedidos em alguns itens que implicam a alteração da programação que já fizeste anteriormente.

1. Cria um botão, com o texto *inserir nome da empresa*.
2. Cria um evento de clique do rato, no botão criado anteriormente.
3. Mostra uma mensagem no ecrã a pedir o nome da empresa quando o evento anterior é executado.
4. Guarda o nome da empresa que será inserido pelo utilizador na variável *nome_empresa*.
5. Adiciona um comentário a indicar o que a variável *nome_empresa* guarda.

6. Cria um fluxograma da programação desenvolvida nas alíneas anteriores, num documento à tua escolha, por exemplo do Microsoft Word. O nome do documento deve ser *fluxograma-1* e deves incluir o teu número de aluno e nome.
7. Cria uma estrutura de decisão que possua uma condição que verifica se a variável *nome_empresa* é vazia, ou seja, se o nome da empresa não foi preenchido. No caso da condição ser verdadeira, mostra uma mensagem de erro, caso contrário, mostra numa mensagem com o nome da empresa.
8. Coloca o nome da empresa visível na aplicação, num local à tua escolha.
9. Quando o ponteiro do rato ficar em cima do nome da empresa deve ser, pedido ao utilizador o número total de parques de estacionamento da empresa.
10. Cria uma função que recebe um parâmetro de entrada (número de total de parques de estacionamento da empresa) e que verifica se o número de total de parques de estacionamento inserido pelo utilizador é maior do que o número máximo permitido, 100, ou menor do que o número mínimo, 0. No caso de se verificar alguma destas situações é mostrada uma mensagem de erro.
11. Se o número de total de parques de estacionamento da empresa inserido pelo utilizador for válido é mostrado esse número à frente do nome da empresa.
12. Adiciona um comentário sobre a função criada.
13. Adiciona uma opção, que pode ser selecionado pelo utilizador, em que é mostrado uma lista com os números dos parques de estacionamento. Por exemplo, se existirem 5 parques é mostrada uma lista com 5 elementos.
14. Cria um fluxograma da programação necessária para colocar à frente do primeiro parque de estacionamento da lista o texto *primeiro* e no último parque de estacionamento o texto *último*. O fluxograma pode ser criado num documento à tua escolha, por exemplo do Microsoft Word. O nome do ficheiro deve ser *fluxograma-2* e deve incluir o teu número de aluno e nome.
15. Aplica o fluxograma que criaste anteriormente, efetuando a sua programação.

Grupo II

De acordo com o código de programação na linguagem *Visual Basic 6.0* apresentado em baixo, responde às questões.

```
Public Class ParqueEstacionamento_Turno2
```

```
'Constante com o número total de lugares no parque de estacionamento
```

```
Private Const num_total As Integer = 10
```

```
Private Sub distribuirLugares()
```

```
Dim lugar As Integer = 1
```

```
While lugar <= num_total
```

```
    If lugar > 0 Or lugar <= 10 Then
```

```
        ListaLugaresLivres.Items.Add(lugar)
```

```
    End If
```

```
        lugar = lugar + 1
```

```
End While
```

```
End Sub
```

```
Private Sub LabelLugares_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles LabelLugares.Click
```

```
    distribuirLugares ()
```

```
End Sub
```

```
End Class
```

1. Foi criada alguma variável? Se sim, indica o nome e o tipo de dados dessa variável.
2. Existem algum comentário ao código escrito? Se sim, diz o que é indicado nesse comentário.
3. Foi criada alguma função? Se sim, indica quais os seus nomes e se têm parâmetros de entrada e/ou saída?
4. No código existem estruturas de decisão e repetição. Descreve o que faz o ciclo *While*.
5. No código encontra-se descrito um evento. Qual o tipo de evento usado?

Anexo K – Grelha de Registos

Grelha de registos

Aula <nº da aula>

Indicadores	Número de aluno						
Responsabilidade							
<i>Cumpra as regras estabelecidas e as diretrizes que lhe são dadas</i>							
<i>Respeita os direitos, opiniões e capacidades dos outros</i>							
<i>Está atento</i>							
<i>Realiza as tarefas propostas</i>							
<i>Mostra-se responsável pelo progresso do seu grupo de trabalho</i>							
<i>Participa espontaneamente</i>							
Autonomia							
<i>Autodisciplina-se</i>							
<i>Tenta ultrapassar as dificuldades</i>							
<i>Preocupa-se com a qualidade do trabalho</i>							

<i>Utiliza o tempo de uma forma organizada</i>							
<i>Identifica as suas dificuldades</i>							
Solidariedade							
<i>Entreajuda na realização de tarefas</i>							
<i>Coopera nos trabalhos de grupo</i>							
<i>Desenvolve o espírito de tolerância</i>							
Comunicação/Expressão							
<i>Exprime-se de forma clara</i>							
<i>Exprime-se com correção linguística</i>							
<i>Utiliza vocabulário específico</i>							

Atitudes e valores	Legenda
Sistemático	<i>A</i>
Dominante	<i>B</i>
Ocasional	<i>C</i>

Anexo L – Guião para a Discussão de Grupo

Diagnóstico e avaliação

1. Quais as tuas principais dificuldades ao realizar os testes de diagnóstico e de avaliação?
2. A realização de uma autoavaliação e de seguida o teste, ajudou-te a identificar os teus pontos fortes e fracos na programação?

Trabalho em grupo

3. Quais foram as dificuldades que sentiste no trabalho em grupo?
4. Preferes trabalhar em grupo ou individualmente?

Utilização de robots

5. A utilização dos robots da Lego ajudaram-te a entender alguns dos conceitos que estavam nos questionários de autodiagnóstico e de autoavaliação?
6. Quais foram as dificuldades que sentiste ao utilizar os robots da Lego?
7. Quais é que achas que são as vantagens na utilização dos robots da Lego?

Aprendizagem de programação

8. Se tivesse sido usado os robots da Lego desde o primeiro módulo da disciplina de Linguagens de Programação, achas que teria ajudado nas tuas aprendizagens na programação?
9. Quais são as dificuldades que sentes ao aprender uma linguagem de programação?
10. O que achas que deve ser alterado no ensino da programação de forma a ajudar-te na aprendizagem?
11. Achas que te ajudou escrever primeiro a solução e só depois ir programá-la?

O professor

12. As indicações dadas pelo professor ajudaram-te na realização do trabalho?
13. A linguagem utilizada pelo professor foi clara?

A intervenção

14. Programaste todas as soluções? Quais os problemas que encontraste?
15. Achas que te ajudou escrever primeiro a solução e só depois ir programá-la?
16. O que é que gostaste mais?
17. O teu interesse pela informática aumentou?

Anexo M – Critérios de Avaliação do Teste de Diagnóstico

Critérios de Avaliação do Grupo I do Teste de Diagnóstico

Questão	Conceito envolvidos	Critério	Nível
1	---	Criou um botão.	---
	---	Colocou o texto correto no botão.	---
2	<i>Eventos</i>	Associou um evento ao botão?	N1
	<i>Eventos</i>	O evento que criou é o de clique do rato no botão?	
3	<i>Eventos</i>	Quando o evento é ativo acontece alguma coisa?	N1
4	<i>Variáveis</i>	Criou uma variável?	N1
	<i>Variáveis</i>	A variável tem o nome <i>nome_turma</i> ?	
	<i>Operadores e expressões relacionais</i>	A variável criada ou outra foi igualada a algum valor?	N1
	<i>Instruções de entrada e de saída</i>	Guarda o valor indicado pelo utilizador na mensagem?	N1
5	<i>Comentários</i>	Criou um comentário?	N1
	<i>Comentários</i>	O comentário diz respeito ao código escrito?	
6	<i>Fluxogramas</i>	Representou um fluxograma?	N1
	<i>Eventos</i>	É representado um evento?	N1
	<i>Variáveis</i>	É representado o uso de uma variável?	N1
	<i>Operadores e expressões relacionais</i>	É representado o uso de um operador relacional?	N1
7	<i>Estruturas de decisão e repetição</i>	Criou um estrutura de decisão?	N1
	<i>Operadores e expressões relacionais</i>	É utilizada uma expressão relacional?	N1
	<i>Operadores e</i>	A variável com o valor nome da turma é associada	

	<i>expressões relacionais</i>	a uma mensagem?	
	<i>Variáveis</i>	É utilizada a variável criada com o valor nome da turma?	N2
	<i>Instruções de entrada e de saída</i>	Mostra uma mensagem com o nome da turma?	N1
8	<i>Variáveis</i>	A variável com o valor nome da turma é associada a algum componente da janela?	N2
	<i>Instruções de entrada e de saída</i>	Coloca o nome da turma na janela da aplicação?	N1
	<i>Eventos</i>	Criou um novo evento?	
	<i>Eventos</i>	O evento foi associado à componente com o nome da turma?	N2
9	<i>Eventos</i>	O evento que criou é o de passar o rato por cima?	
	<i>Eventos</i>	Quando o evento é ativo acontece alguma coisa?	
	<i>Variáveis</i>	Criou uma nova variável?	N2
	<i>Operadores e expressões relacionais</i>	A variável foi igualada a algum valor?	N2
	<i>Estruturas de decisão e repetição</i>	Criou uma nova estrutura de decisão?	N2
10	<i>Operadores e expressões relacionais</i>	É utilizada uma expressão relacional?	N2
	<i>Variáveis</i>	É utilizada uma variável com o valor número de alunos?	N2
	<i>Instruções de entrada e de saída</i>	Guarda o valor indicado pelo utilizador na mensagem?	N2
11	<i>Variáveis</i>	A variável com o valor número de alunos é associada ao nome da turma?	N2
	<i>Instruções de</i>	É mostrado ao utilizador o número de alunos?	N1

entrada e de saída

12	<i>Comentários</i>	Criou um comentário?	N2
	<i>Comentários</i>	O comentário diz respeito ao código escrito?	
	<i>Comentários</i>	Mencionou a existência dos parâmetros de entrada ou de saída?	
13	<i>Eventos</i>	Associou um novo evento a um novo botão ou componente?	N2
	<i>Eventos</i>	O evento que criou é o de clique do rato no botão?	
	<i>Eventos</i>	Quando o evento é ativo acontece alguma coisa?	
	<i>Estruturas de decisão e repetição</i>	Criou uma estrutura de repetição?	N2
	<i>Operadores e expressões relacionais</i>	Na condição é utilizada uma expressão relacional?	N2
	<i>Variáveis</i>	É criada alguma variável?	N2
	<i>Variáveis</i>	Alguma variável é usada?	
	<i>Operadores e expressões relacionais</i>	É utilizado algum operador relacional?	N2
	<i>Instruções de entrada e de saída</i>	Mostra uma lista com os números dos alunos?	N2
	14	<i>Fluxogramas</i>	Representou um novo fluxograma?
<i>Variáveis</i>		É representado o uso de uma variável?	N2
<i>Estruturas de decisão e repetição</i>		É representado o uso de uma estrutura de decisão?	N2
<i>Estruturas de decisão e repetição</i>		É representado o uso de uma estrutura de repetição?	
<i>Operadores e expressões relacionais</i>		É representado o uso de um operador relacional?	N2

	<i>Operadores e expressões relacionais</i>	É representado o uso de uma expressão relacional?	
	<i>Instruções de entrada e de saída</i>	É representado a escrita de uma lista com os números de alunos e o texto?	N2
	<i>Fluxogramas</i>	O código criado representa o que foi definido no fluxograma?	N2
	<i>Estruturas de decisão e repetição</i>	Alterou a estrutura de repetição anterior?	N2
	<i>Estruturas de decisão e repetição</i>	Criou uma nova estrutura de repetição?	
	<i>Variáveis</i>	Existe a manipulação de alguma variável?	N2
15	<i>Operadores e expressões relacionais</i>	Criou uma estrutura de decisão?	
	<i>Operadores e expressões relacionais</i>	Utiliza algum operador relacional?	N2
	<i>Operadores e expressões relacionais</i>	Utiliza alguma expressão relacional?	
	<i>Instruções de entrada e de saída</i>	Mostra uma lista com os números dos alunos e o texto?	N2

Cr terios de Avalia o do Grupo II do Teste de Diagn stico

Quest�o	Conceito	Cr�terios
1	<i>Vari�veis</i>	Identificou uma vari�vel?
	<i>Vari�veis</i>	Indicou o nome correto?
	<i>Vari�veis</i>	Indicou o tipo de dados correto?
2	<i>Coment�rios</i>	Identificou um coment�rio?
3	---	Indicou a exist�ncia de uma ou mais fun�o�es?
	---	Indicou o nome correto de alguma das fun�o�es?
	---	Indicou a exist�ncia de par�metros de entrada ou de sa�da?
4	<i>Estruturas de decis�o e repeti�o</i>	Indicou o objetivo do ciclo While?
	<i>Estruturas de decis�o e repeti�o</i>	Menciona a exist�ncia de alguma estrutura de decis�o?
	<i>Operadores e express�es relacionais</i>	Menciona a exist�ncia de operadores ou express�es relacionais?
	<i>Vari�veis</i>	Menciona a manipula�o de vari�veis?
5	<i>Instru�o�es de entrada e de sa�da</i>	Menciona que � mostrada informa�o ao utilizador?
	<i>Eventos</i>	Identificou o evento?

Anexo N - Critérios de Avaliação do Teste de Avaliação

Critérios de Avaliação do Grupo I do Teste de Avaliação

Questão	Conceito	Critério	Nível
1	---	Criou um botão.	---
	---	Colocou o texto correto no botão.	---
2	<i>Eventos</i>	Associou um evento ao botão?	N1
	<i>Eventos</i>	O evento que criou é o de clique do rato no botão?	
3	<i>Eventos</i>	Quando o evento é ativo acontece alguma coisa?	N1
4	<i>Variáveis</i>	Criou uma variável?	N1
	<i>Variáveis</i>	A variável tem o nome <i>nome_empresa</i> ?	
	<i>Operadores e expressões relacionais</i>	A variável criada ou outra foi igualada a algum valor?	N1
	<i>Instruções de entrada e de saída</i>	Guarda o valor indicado pelo utilizador na mensagem?	N1
5	<i>Comentários</i>	Criou um comentário?	N1
	<i>Comentários</i>	O comentário diz respeito ao código escrito?	
6	<i>Fluxogramas</i>	Representou um fluxograma?	N1
	<i>Eventos</i>	É representado um evento?	N1
	<i>Variáveis</i>	É representado o uso de uma variável?	N1
7	<i>Operadores e expressões relacionais</i>	É representado o uso de um operador relacional?	N1
	<i>Estruturas de decisão e repetição</i>	Criou um estrutura de decisão?	N1
	<i>Operadores e expressões relacionais</i>	É utilizada uma expressão relacional?	N1

	<i>Operadores e expressões relacionais</i>	A variável com o valor nome da turma é associada a uma mensagem?	
	<i>Variáveis</i>	É utilizada a variável criada com o valor nome da empresa?	N2
	<i>Instruções de entrada e de saída</i>	Mostra uma mensagem com o nome da empresa?	N1
8	<i>Variáveis</i>	A variável com o valor nome da empresa é associada a algum componente da janela?	N2
	<i>Instruções de entrada e de saída</i>	Coloca o nome da turma na janela da aplicação?	N1
9	<i>Eventos</i>	Criou um novo evento?	
	<i>Eventos</i>	O evento foi associado à componente com o nome da empresa?	N2
	<i>Eventos</i>	O evento que criou é o de passar o rato por cima?	
	<i>Eventos</i>	Quando o evento é ativo acontece alguma coisa?	
	<i>Variáveis</i>	Criou uma nova variável?	N2
	<i>Operadores e expressões relacionais</i>	A variável foi igualada a algum valor?	N2
	<i>Estruturas de decisão e repetição</i>	Criou uma nova estrutura de decisão?	N2
10	<i>Operadores e expressões relacionais</i>	É utilizada uma expressão relacional?	N2
	<i>Variáveis</i>	É utilizada uma variável com o valor número de parques?	N2
	<i>Instruções de entrada e de saída</i>	Guarda o valor indicado pelo utilizador na mensagem?	N2
11	<i>Variáveis</i>	A variável com o valor número de parques é	N2

associada ao nome da empresa?		
	<i>Instruções de entrada e de saída</i>	É mostrado ao utilizador o número de parques? N1
12	<i>Comentários</i>	Criou um comentário?
	<i>Comentários</i>	O comentário diz respeito ao código escrito? N2
	<i>Comentários</i>	Mencionou a existência dos parametros de entrada ou de saída?
	<i>Eventos</i>	Associou um novo evento a um novo botão ou componente? N2
	<i>Eventos</i>	O evento que criou é o de clique do rato no botão?
	<i>Eventos</i>	Quando o evento é ativo acontece alguma coisa?
13	<i>Estruturas de decisão e repetição</i>	Criou uma estrutura de repetição? N2
	<i>Operadores e expressões relacionais</i>	Na condição é utilizada uma expressão relacional? N2
	<i>Variáveis</i>	É criada alguma variável? N2
	<i>Variáveis</i>	Alguma variável é usada?
	<i>Operadores e expressões relacionais</i>	É utilizado algum operador relacional? N2
	<i>Instruções de entrada e de saída</i>	Mostra uma lista com os números dos alunos? N2
	<i>Fluxogramas</i>	Representou um novo fluxograma? N2
	<i>Variáveis</i>	É representado o uso de uma variável? N2
14	<i>Estruturas de decisão e repetição</i>	É representado o uso de uma estrutura de decisão? N2
	<i>Estruturas de decisão e repetição</i>	É representado o uso de uma estrutura de repetição?

	<i>repetição</i>		
	<i>Operadores e expressões relacionais</i>	É representado o uso de um operador relacional?	N2
	<i>Operadores e expressões relacionais</i>	É representado o uso de uma expressão relacional?	
	<i>Instruções de entrada e de saída</i>	É representado a escrita de uma lista com os números de parques e o texto?	N2
	<i>Fluxogramas</i>	O código criado representa o que foi definido no fluxograma?	N2
	<i>Estruturas de decisão e repetição</i>	Alterou a estrutura de repetição anterior?	N2
	<i>Estruturas de decisão e repetição</i>	Criou uma nova estrutura de repetição?	
	<i>Variáveis</i>	Existe a manipulação de alguma variável?	N2
15	<i>Operadores e expressões relacionais</i>	Criou uma estrutura de decisão?	
	<i>Operadores e expressões relacionais</i>	Utiliza algum operador relacional?	N2
	<i>Operadores e expressões relacionais</i>	Utiliza alguma expressão relacional?	
	<i>Instruções de entrada e de saída</i>	Mostra uma lista com os números de parques e o texto?	N2

Critérios de Avaliação do Grupo II do Teste de Avaliação

Questão	Conceito	Crítérios
1	<i>Variáveis</i>	Identificou uma variável?
	<i>Variáveis</i>	Indicou o nome correto?
	<i>Variáveis</i>	Indicou o tipo de dados correto?
2	<i>Comentários</i>	Identificou um comentário?
3	---	Indicou a existência de uma ou mais funções?
	---	Indicou o nome correto de alguma das funções?
	---	Indicou a existência de parâmetros de entrada ou de saída?
4	<i>Estruturas de decisão e repetição</i>	Indicou o objetivo do ciclo While?
	<i>Estruturas de decisão e repetição</i>	Menciona a existência de alguma estrutura de decisão?
	<i>Operadores e expressões relacionais</i>	Menciona a existência de operadores ou expressões relacionais?
	<i>Variáveis</i>	Menciona a manipulação de variáveis?
5	<i>Instruções de entrada e de saída</i>	Menciona que é mostrada informação ao utilizador?
	<i>Eventos</i>	Identificou o evento?

Anexo O – Relação Entre os Questionários e os Testes

Relação Entre os Conceitos dos Questionários e as Questões do Grupo I dos Testes.

Questão	Conceitos						
	Fluxogramas	Comentários	Variáveis	Operadores e expressões relacionais	Estruturas de decisão e repetição	Eventos	Instruções de entrada e de saída
1							
2						N1	
3						N1	
4			N1	N1			N1
5		N1					
6	N1		N1	N1		N1	
7			N2	N1	N1		N1
8			N2				N1

9		N2	N2		N2	
10		N2	N2	N2		N2
11		N2				N1
12	N2					
13		N2	N2	N2	N2	N2
14	N2	N2	N2	N2		N2
15	N2	N2	N2	N2		N2

Nota: Sabe aplicar o conceito quando pedido diretamente (N1); Sabe aplicar o conceito quando não é pedido diretamente (N2).

Relação Entre os Conceitos dos Questionários e as Questões do Grupo II dos Testes.

Questão	Conceito						
	Fluxogramas	Comentários	Variáveis	Operadores e expressões relacionais	Estruturas de decisão e repetição	Eventos	Instruções de entrada e de saída
1			X				
2		X					
3							
4			X	X	X		X
5						X	

Nota: Presente (X)