INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

Ensino-Aprendizagem de Programação

Luís Manuel Alves

Bragança

Novembro de 2019

Título: **Ensino-Aprendizagem de Programação**; Autor: Luís M. Alves; Suporte: Eletrónico; Formato: PDF / PDF/A; ISBN: 978-989-33-4480-4

- esta página foi deixada deliberadamente em branco -

Lição a apresentar na Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Bragança para prestação de Provas Públicas de Avaliação da Competência Pedagógica e Técnico-Científica, prevista nos n.º 9, 10 e 11 do artigo 6.º e do n.º 5 do artigo 8.º-A do Decreto-Lei n.º 207/2009, de 31 de agosto, alterado e aditado pela Lei n.º 7/2010, de 13 de maio e pelo Decreto-Lei n.º45/2016, de 17 de agosto, alterado pela Lei n.º 65/2017, de 9 de agosto.

Nota Prévia

O presente trabalho foi elaborado para efeitos da Lição que o docente se propõe apresentar nas Provas Públicas de Avaliação da Competência Pedagógica e Técnico-científica, cujo tema escolhido foi "Ensino-Aprendizagem de Programação". O seu conteúdo, a metodologia e as fontes que lhe serviram de base são da sua inteira responsabilidade e são o resultado da experiência acumulada nos mais de vinte anos que já leciona, no projeto internacional em que participou e na investigação conduzida no âmbito dos trabalhos de doutoramento.

O ensino-aprendizagem de linguagens de programação é uma tarefa difícil reconhecida mundialmente. Várias têm sido as tentativas para atenuar esta dificuldade, nomeadamente, na adoção de novas metodologias de ensino, novas ferramentas de suporte e novas linguagens de programação. Na presente Lição são apresentados alguns dados obtidos num projeto internacional sobre esta temática envolvendo várias instituições de ensino superior de Portugal e da Sérvia em que o autor participou ativamente.

Na lecionação das Unidades Curriculares (UCs) de Programação I e Programação II aos cursos de licenciatura em Engenharia Informática e Informática de Gestão da Escola Superior de Tecnologia e Gestão (ESTiG) do Instituto Politécnico de Bragança (IPB), foram percecionadas dificuldades evidenciadas pelos estudantes na apreensão e aplicação de determinados conceitos. Motivo que impulsionou o desenvolvimento de técnicas e metodologias potenciadoras da transferência de conhecimento e de aprendizagem. Assim, ao longo dos anos têm-se potenciado o ensino baseado na prática, com um envolvimento ativo e colaborativo dos estudantes.

Naquelas UCs, após a explicação teórica dos conceitos envolvidos na programação é efetuada uma aplicação prática dos mesmos, envolvendo ativamente os estudantes. Em todas as aulas são usados os computadores dos laboratórios ou os portáteis pessoais dos estudantes na realização de exercícios distribuídos por fichas de trabalho que cobrem os diferentes assuntos do conteúdo programático de cada Unidade Curricular (UC). Nos últimos anos, têm sido introduzidas ferramentas *online* de suporte ao ensino da programação.

A criação de um ambiente académico que permita a aquisição de *soft skills* por estudantes das Tecnologias da Informação é um desafio encorajador. Neste sentido, é apresentado nesta Lição um ambiente criado na Universidade do Minho que potencia a aquisição destas competências. Este ambiente tem por base o Ensino Baseado em Projetos (*Project-Based Learning* (PBL)), cujo autor desta lição participou como aluno de doutoramento.

Índice Geral

Lista de Siglas e Acrónimos

DM	*Data Mining*
DIC	Departamento de Informática e Comunicações
ESTiG	Escola Superior de Tecnologia e Gestão
ECTS	*European Credit Transfer and Accumulation System*
IDE	*Integrated Development Environment*
IPB	Instituto Politécnico de Bragança
LMS	*Learning Management Systems*
PBL	*Project-Based Learning*
UC	Unidade Curricular

Índice de Figuras

Índice de Tabelas

Capítulo 1
Âmbito e Propósito

Este capítulo esclarece qual o âmbito e o propósito da Lição, explicando a escolha do grupo de unidades curriculares. O capítulo explicita, justificadamente, a visão do autor sobre o que deve versar este tipo de Lição e termina com a descrição da forma como a mesma está estruturada.

1.1 Introdução

Esta Lição contém o programa, o conteúdo e os métodos de ensino referentes a um grupo de unidades curriculares no âmbito da área disciplinar das Ciências da Computação, como satisfação parcial dos requisitos para a prestação de Provas Públicas de Avaliação da Competência Pedagógica e Técnico-Científica, prevista nos n.º 9, 10 e 11 do artigo 6.º e do n.º 5 do artigo 8.º-A do Decreto-Lei n.º 207/2009, de 31 de agosto, alterado e aditado pela Lei n.º 7/2010, de 13 de maio e pelo Decreto-Lei n.º45/2016, de 17 de agosto, alterado pela Lei n.º 65/2017, de 9 de agosto.

O grupo de unidades escolhido integra as unidades curriculares de 'Programação I' e 'Programação II' que têm sido lecionadas aos cursos de licenciatura em Engenharia Informática e Informática de Gestão promovidas pelo Departamento de Informática e Comunicações (DIC) da Escola Superior de Tecnologia e Gestão (ESTiG) do Instituto Politécnico de Bragança (IPB).

A escolha deste grupo de unidades curriculares não ocorreu sem razão fundamentada. Por um lado, trata-se de um grupo de unidades curriculares que se situa na área disciplinar em que o autor tem centrado a sua atividade pedagógica desde a sua integração no IPB. Por outro

lado, contribuiu também a escolha o facto de, nos últimos anos, o autor ter desenvolvimento parte da atividade científica nesta área, nomeadamente, no âmbito de trabalhos de doutoramento e num projeto internacional bilateral entre Portugal e a Sérvia. De notar que, havendo consciência de que o ensino-aprendizagem de uma linguagem de programação, não é tarefa fácil, sobretudo se for a primeira, têm sido promovidas algumas medidas para atenuar este esforço.

De realçar que o grupo de unidades curriculares em análise têm uma importância vital no bom desempenho dos estudantes ao longo do curso, pois, servem como alicerces para as unidades curriculares subsequentes. Assim, os estudantes deveriam ter todo o interesse em adquirir o máximo de competências na aprendizagem da linguagem de programação introdutória. Estas UCs têm um esforço de aprendizagem de 6 ECTS contabilizando um número total de horas de trabalho de 162 horas, sendo 60 das quais aulas de contacto de cariz prático e laboratorial. Na ESTiG, os cursos de licenciatura têm 3 anos de duração, compreendendo 6 semestres, com um total de 30 unidades curriculares em que o esforço de aprendizagem totaliza 180 ECTS.

1.2 Natureza e Estrutura da Lição

Para além da própria legislação mencionada no início da secção anterior, são escassos ou inexistentes os referenciais ou recomendações para este tipo de Lição. Contudo, houve uma preocupação latente do autor em procurar sugestões de colegas, nomeadamente, de Professores Coordenadores, no sentido de elaborar uma Lição o mais interessante possível.

Desta forma, entende-se que, para além de um sumário desenvolvido dos conteúdos programáticos do grupo de unidades curriculares escolhido, esta Lição deve: (1) apresentar os resultados obtidos num projeto internacional sobre o ensino-aprendizagem de programação; (2) apresentar algumas ferramentas de apoio ao ensino-aprendizagem de programação; (3) divulgar uma experiência pedagógica baseada em Ensino Baseado em Projetos.

Com base na experiência docente de mais de duas décadas do autor, esta Lição expõe uma visão pessoal da forma como o ensino das temáticas de Programação em cursos do ensino superior deve ser organizado. Este documento estrutura sumariamente os grandes temas das unidades curriculares e tece algumas considerações de índole pedagógica sobre os métodos de ensino e de avaliação adotados. O relatório está organizado em quatro capítulos:

(1) *Introdução*. Este capítulo esclarece qual o âmbito e o propósito da Lição, explicando a escolha do grupo de unidades curriculares. O capítulo explicita, justificadamente, a visão do autor sobre o que deve versar este tipo de Lição e termina com a descrição da forma como a mesma está estruturada.

(2) *Grupo de Unidades Curriculares de Programação*. Este capítulo caracteriza o grupo de unidades curriculares segundo os aspetos habitualmente realçados, nomeadamente a descrição dos conteúdos programáticos propostos, os resultados de aprendizagem que se pretendem atingir, a lista bibliográfica e de leituras recomendadas, as metodologias de ensino e respetiva justificação, os métodos de avaliação adotados e os principais recursos necessários para o seu funcionamento.

(3) *Esforços de Melhoria no Ensino de Competências de Programação*. Neste capítulo após uma breve introdução são apresentados os projetos de investigação na área do ensino-aprendizagem de Programação em que o autor esteve e estará envolvido. O primeiro projeto, que está em fase final, versa o ensino-aprendizagem da programação maioritariamente a estudantes iniciantes nesta temática. Englobado neste projeto, é apresentada uma ferramenta de apoio ao ensino da programação, em que são apresentados alguns dados apreciativos dos estudantes. O segundo projeto versa a área da Engenharia e Gestão do Processo de Desenvolvimento de Software num sentido mais lato no qual é apresentada uma experiência pedagógica baseada em PBL de elevado interesse atual.

(4) *Conclusões*. Este capítulo apresenta o sumário dos principais aspetos referidos nesta Lição. De uma forma sucinta são apresentadas as principais conclusões obtidas, de forma a interligar todos os assuntos abordados.

Ao longo do texto desta Lição, sempre que adequado são apresentadas referências bibliográficas que reforcem as ideias referidas. Assim, é possível identificar as publicações que foram consultadas e interrelacionar as várias temáticas envolvidas.

Nesta Lição, em termos de grafia, adotam-se as aspas do tipo " " para transcrições de textos escritos por outros autores, as aspas do tipo ' ' para evidenciar a forma de designar ou

referenciar algum conceito ou termo e as aspas do tipo « » para obviar alguma expressão menos cuidada quando se recorre, por exemplo, por questões de expressividade linguística, a terminologia coloquial, claramente desaconselhada num relatório de cariz académico como este. Recorre-se à grafia em itálico sempre que se utilizam palavras ou expressões escritas em língua estrangeira.

Capítulo 2
Grupo de Unidades Curriculares de Programação

Este capítulo caracteriza o grupo de unidades curriculares segundo os aspetos habitualmente realçados, nomeadamente a descrição dos conteúdos programáticos propostos, os resultados de aprendizagem e competências que se pretendem atingir, a lista bibliográfica, as metodologias de ensino e respetiva justificação, os métodos de avaliação adotados e os principais recursos necessários para o seu funcionamento.

2.1 Introdução

O grupo de unidades curriculares escolhido para esta Lição tem como denominador comum o ensino da programação. Os conteúdos programáticos estão estruturados de forma a que os estudantes compreendam o processo de desenvolvimento de software desde a aplicação de conceitos básicos até conceitos de média e elevada complexidade.

Numa instituição de ensino superior, não deve existir a preocupação de ensinar, como fito único, a última novidade tecnológica (seja uma linguagem de programação, seja um protocolo de comunicação, seja um sistema operativo), mas, pelo contrário, deve promover-se os fundamentos metodológicos que confiram aos seus diplomados a capacidade de apreender, manipular, conceber e utilizar tecnologia (Parnas, 1990). Concretamente, no caso da programação, mais do que ensinar a sintaxe, a gramática e o vocabulário da linguagem de programação escolhida é «obrigação» dos professores encontrar métodos e ferramentas que auxiliem a potenciar o raciocínio lógico dos estudantes.

Os professores devem fazer um esforço para ensinar a aprender ou, dito de outra forma (segundo a terminologia recomendada no âmbito do Processo de Bolonha), os estudantes devem aprender a aprender. Esta abordagem no desenvolvimento das competências dos estudantes é muito importante, sobretudo na área da Informática, devido à sua rapidíssima evolução. Tem sido preocupação constante dos docentes que lecionam este grupo de unidades curriculares encontrar novas metodologias e ferramentas que potenciem esta abordagem, tendo sempre em vista o estabelecimento dos objetivos educacionais a alcançar.

Os objetivos de uma unidade curricular devem estar relacionados com aquilo que os estudantes deverão ser capazes de fazer, no final da sua frequência. A identificação de objetivos claros é uma intenção fundamental na definição da oferta curricular, possibilitando uma opção racional dos conteúdos e das atividades de aprendizagem e facilitando o estabelecimento de uma estratégia de avaliação (Newble & Cannon, 2000).

Além de um ensino organizado, claro e estimulante, o alinhamento dos propósitos genéricos da unidade curricular (também designados de "*course outlines*") com os objetivos do curso (também designados de "*general program objectives*") e com os objetivos, a longo prazo, dos próprios estudantes, é um dos fatores indicado por especialistas em pedagogia como capaz de aumentar o desempenho dos estudantes. Para tal ser possível, é crucial indicar aos estudantes, de uma forma clara, quais os objetivos da unidade curricular (também designados de "*course general instructional objectives*"), bem como quais os resultados de aprendizagem (também designados de "*learning outcomes and and competences*") que decorrem da realização dos objetivos estabelecidos. Um resultado de aprendizagem deve ser descrito, em termos conceptuais, como capacidades ou estados do conhecimento (Pratt, 1994).

Atendendo que as unidades curriculares de Programação I e Programação II pertencem aos planos curriculares dos cursos de licenciatura em Engenharia Informática e Informática de Gestão convém apresentar o perfil do programa de estudos e os principais resultados da aprendizagem para cada um dos cursos.

No caso de Informática de Gestão, "o perfil do programa de estudos compreende o desenvolvimento do conhecimento técnico-científico e do saber de natureza profissional nas áreas de estudos de Informática de Gestão: Ciências da Computação; Sistemas de Informação; Engenharia de Computadores. Como principais resultados de aprendizagem é referido que o grau de licenciado em Informática de Gestão é conferido aos alunos que demonstrem possuir os conhecimentos, as capacidades e as competências necessários para resolver problemas e evidenciar uma abordagem profissional nos seguintes domínios: suporte à gestão e administração, gestão e administração de bases de dados e aplicações informáticas, planeamento e

gestão de projetos informáticos, análise de sistemas, auditoria e consultoria em sistemas de informação, programação de computadores, administração e manutenção de redes de comunicações e sistemas informáticos, serviços Web e segurança de sistemas de informação"[1].

No caso de Engenharia Informática "o perfil do programa de estudos compreende o desenvolvimento do conhecimento técnico-científico e do saber de natureza profissional nas áreas de estudos de Engenharia Informática: Engenharia de Computadores; Ciências da Computação; Sistemas de Informação; Projeto. Como principais resultados de aprendizagem é referido que o "grau de licenciado em Engenharia Informática é conferido aos alunos que demonstrem ter, nesta área, a capacidade de:

(1) desenvolver e aprofundar os conhecimentos adquiridos;

(2) aplicar os conhecimentos adquiridos, de forma a evidenciar uma abordagem profissional ao trabalho desenvolvido;

(3) resolver problemas e de construir e fundamentar a sua própria argumentação;

(4) recolher, selecionar e interpretar a informação relevante, que habilite a fundamentar as soluções preconizadas e os juízos emitidos, incluindo na análise os aspetos sociais, científicos e éticos relevantes;

(5) comunicar informação, ideias, problemas e soluções, tanto a públicos constituídos por especialistas como por não especialistas;

(6) desenvolver competências que permitam uma aprendizagem ao longo da vida com elevado grau de autonomia"[2].

Nas secções seguintes são apresentados alguns detalhes das unidades curriculares de Programação I e Programação II existentes no seu guia de ECTS.

2.2 Programação I: Programa e Resultados de Aprendizagem

A unidade curricular de 'Programação I', com o código 9119-606-1104-00-18 (Engenharia Informática) e o código 9186-361-1105-00-18 (Informática de Gestão) no catálogo de cursos do IPB, apresenta um regime de funcionamento semestral, adota o tipo de ensino presencial e é ministrada em língua Portuguesa e em língua Inglesa (uma turma). Esta é uma UC com 6

[1] http://portal3.ipb.pt/index.php/pt/guiaects/cursos/licenciaturas/curso?cod_escola=3043&cod_curso=9186
[2] http://portal3.ipb.pt/index.php/pt/guiaects/cursos/licenciaturas/curso?cod_escola=3043&cod_curso=9119

créditos ECTS, que correspondem a 162 horas de esforço total, das quais 60 ministradas segundo uma escolaridade de ensino Prático e Laboratorial; das 162 horas, 102 devem corresponder a trabalho autónomo. Em termos de planos de estudos, Programação I integra-se na área científica das Ciências da Computação. A sua duração letiva no semestre é de 20 semanas, 5 das quais corresponde ao período de avaliação. A UC tem uma carga horária de 4 horas, divididas por duas aulas de duas horas.

Em termos de origem, Programação I é a «herdeira» da disciplina semestral de 'Linguagens da Programação' lecionada às anteriores Licenciaturas em Engenharia Informática, Engenharia Eletrotécnica e Informática de Gestão. Em 2006/07 passou a ser ministrada no 1º semestre do 1º ano apenas às licenciaturas em Engenharia Informática e Informática de Gestão. Esta reformulação coincide com a adequação dos cursos ao Tratado de Bolonha.

O programa da UC de Programação I encontra-se estruturado em várias unidades de aprendizagem, com o intuito de enquadrar, de uma forma coerente do ponto de vista temático e eficaz em termos do processo de ensino-aprendizagem, conjuntos canónicos de resultados de aprendizagem. A **Tabela 2.1** apresenta o conteúdo programático da UC, onde se podem observar sete unidades de aprendizagem.

Esta UC promove um plano de formação inicial do ensino da programação. Assim, após um primeiro capítulo onde são abordados conceitos genéricos sobre a programação de computadores, as linguagens de programação e o processo de desenvolvimento de software os restantes capítulos são dedicados ao ensino da linguagem de programação C.

O conteúdo programático foi concebido para estudantes sem qualquer conhecimento ao nível da programação, embora, na prática, cheguem às nossas salas de aula estudantes com algum *background* nesta área. Contudo, é preocupação constante do corpo docente manter um nível equilibrado na apresentação e aplicação dos conhecimentos.

Desde o ano letivo de 2006/2007 que a primeira linguagem de programação lecionada às licenciaturas em Engenharia Informática e Informática de Gestão é a linguagem C, anteriormente era lecionada a linguagem de programação Pascal. A escolha da «melhor» primeira linguagem de programação não é unanime, havendo diversas opiniões a este nível. Para verificar esta afirmação, basta percorrer os planos de estudos de cursos na área da informática disponibilizadas pelas universidades e institutos existentes em Portugal. A nível internacional a situação também não é muito diferente.

Tabela 2.1 – Conteúdo da unidade curricular Programação I

1. Conceitos introdutórios:
- Programação de computadores;
- Linguagens de programação;
- Fases de desenvolvimento de um programa;
- A linguagem C.

2. Dados de tipo elementar:
- Tipos de dados, declaração de variáveis;
- Conceito de constante, definição de constantes simbólicas;
- Operações aritméticas, instruções, instrução de atribuição, conversões de tipo;
- Instruções de leitura e de escrita na consola.

3. Testes e condições:
- Condições e valores lógicos;
- Operadores lógicos e operadores relacionais;
- As instruções de seleção *if*, *if-else* e *switch*.

4. Instruções de iteração:
- Instrução *while*;
- Instrução *do-while*;
- Instrução *for*.

5. Funções:
- Conceito de função e estrutura de uma função em C;
- Parâmetros passados por valor;
- Conceito de variável local/global, interna/externa e automática/estática.

6. Vetores:
- Declaração e inicialização automática de vetores;
- Passagem de vetores para funções;
- Processamento de vetores;
- *Arrays* multidimensionais.

7. *Strings*:
- Principais funções de manipulação de *strings*;
- Desenvolvimento de funções específicas para o tratamento de *strings*.

Segundo o índice da comunidade de programação TIOBE[3], um dos indicadores que mede a popularidade das linguagens de programação, a linguagem C aparece em segundo lugar logo a seguir ao Java. De notar que a linguagem C tem permanecido nos lugares cimeiros ao longo dos anos. Este índice é atualizado uma vez por mês. As classificações são baseadas no número de engenheiros qualificados em todo o mundo, cursos e fornecedores de serviços e produtos da área.

Os resultados de aprendizagem e competências (*Learning outcomes and competences*) que se espera que os estudantes venham a atingir está centrado em três domínios essenciais, a estruturação do raciocínio lógico para delinear uma solução, a implementação da solução em linguagem C e a utilização de ferramentas de desenvolvimento. A **Tabela 2.2** mostra os três

[3] https://www.tiobe.com/tiobe-index//

resultados da aprendizagem estruturantes desta UC. Nos últimos três anos letivos tem-se motivado e «obrigado» os alunos a usarem o IDE (*Integrated Development Environment*) *Microsoft Visual Studio Community* (atualmente na versão 2019) por dois motivos: (1) por ser usado em UCs de anos subsequentes; (2) por ser gratuito para fins académicos.

Tabela 2.2 – Resultados da aprendizagem e competências de Programação I

No fim da unidade curricular o aluno deve ser capaz de:
1. Estruturar um raciocínio que lhe permita delinear uma solução e implementar um programa em C que cumpra os objetivos pretendidos, para problemas de pequena/média complexidade.
2. Aplicar conhecimentos de programação imperativa, na linguagem C, designadamente estruturar um programa em funções, compreender a passagem de parâmetros por valor e processar *arrays* e *strings*.
3. Utilizar as ferramentas de depuração no ambiente de desenvolvimento (*Visual Studio*).

2.3 Programação II: Programa e Resultados de Aprendizagem

A unidade curricular de 'Programação II', com o código 9119-606-1205-00-18 (Engenharia Informática) e o código 9186-361-1204-00-18 (Informática de Gestão) no catálogo de cursos do IPB, apresenta um regime de funcionamento semestral, adota o tipo de ensino presencial e é ministrada em língua Portuguesa e em língua Inglesa (uma turma). Esta é uma UC com 6 créditos ECTS, que correspondem a 162 horas de esforço total, das quais 60 ministradas segundo uma escolaridade de ensino Prático e Laboratorial; das 162 horas, 102 devem corresponder a trabalho autónomo. Em termos de planos de estudos, Programação II integra-se na área científica de Ciências da Computação. A sua duração letiva no semestre é de 20 semanas, 5 das quais corresponde ao período de avaliação. A UC tem uma carga horária de 4 horas, divididas por duas aulas de duas horas.

Em termos de origem, Programação II é também «herdeira» da disciplina semestral de 'Linguagens da Programação' lecionada às anteriores Licenciaturas em Engenharia Informática, Engenharia Eletrotécnica e Informática de Gestão, em 2006/07 passou a ser ministrada no 2º semestre do 1º ano apenas às licenciaturas em Engenharia Informática e Informática de Gestão. Esta reformulação coincide com a adequação dos cursos ao Tratado de Bolonha.

O programa da UC de Programação II encontra-se estruturado em várias unidades de aprendizagem, com o intuito de abordar conceitos avançados da linguagem de programação

C. A **Tabela 2.3** apresenta o conteúdo programático da UC, onde se podem observar seis unidades de aprendizagem. Com exceção da primeira, as restantes unidades de aprendizagem complementam os conhecimentos abordados em Programação I.

Tabela 2.3 **– Conteúdo da unidade curricular Programação II**

1. Boas práticas de programação:
- Programação modular;
- Documentação de código;
- Controle de versões.

2. Apontadores:
- Noção de variável, endereço e apontador;
- Declaração e inicialização de variáveis do tipo apontador;
- Operadores de apontadores;
- Relação entre apontadores e vetores;
- Apontadores de apontadores.

3. Passagem de parâmetros:
- Utilização da memória entre chamadas a funções;
- Passagem de parâmetros por valor;
- Passagem dos endereços das variáveis;
- Passagem de *arrays* para funções;
- Passagem de argumentos na linha de comando;
- Recursividade.

4. Ficheiros:
- Noção de ficheiro, periféricos e *streams*;
- Funções de manipulação de ficheiros;
- Formas de abertura de ficheiros;
- Leitura e escrita em ficheiros de texto;
- Leitura e escrita em ficheiros binários;
- Acesso sequencial e acesso direto a ficheiros;
- Deteção do final de ficheiro.

5. Estruturas e enumerações:
- Noção de estrutura, declaração e inicialização de estruturas;
- Acesso aos campos de uma estrutura;
- Passagem de estruturas para funções;
- Ficheiros de estruturas;
- Definição de tipos enumerados;
- Leitura e escrita de variáveis de tipo enumerado;
- Definição de novos tipos de dados.

6. Memória dinâmica:
- Alocação e libertação de memória;
- Funções que retornam memória alocada dinamicamente;
- Estruturas de dados dinâmicas;
- Implementação e manipulação de listas ligadas.

As duas UCs, Programação I e Programação II permitem que o estudante adquira uma base sólida na área da programação. Sendo esta, uma área chave para ambos os cursos, Engenharia Informática e Informática de Gestão, é de todo importante que o estudante adquira o máximo de competências nesta área. O sucesso de várias UCs em anos posteriores do curso

muito dependem do sucesso destas duas UCs. Convém referir ainda que, a programação será certamente uma das áreas em que o estudante irá trabalhar quando terminar o seu curso.

Os resultados de aprendizagem e competências que se espera que os estudantes venham a atingir está centrado em três vertentes essenciais, o desenvolvimento de programas de média/elevada complexidade, a aplicação de conceitos intermédios e a aplicação de conceitos avançados da linguagem C. A **Tabela 2.4** mostra os três resultados da aprendizagem estruturantes desta UC.

Tabela 2.4 - Resultados da aprendizagem e competências de Programação II

No fim da unidade curricular o aluno deve ser capaz de:
1. Desenvolver programas em C de média/elevada complexidade.
2. Aplicar conhecimentos intermédios de programação na linguagem C: dominar a utilização de apontadores, dominar a passagem de parâmetros para funções.
3. Aplicar conhecimentos avançados de programação na linguagem C: utilizar estruturas e enumerações, definir novos tipos de dados, utilizar memória dinâmica, utilizar ficheiros.

2.4 Método de Ensino e de Aprendizagem

As aulas presenciais são realizadas em laboratórios de informática, em que alguns alunos (poucos) usam os computadores da escola e outros (muitos) usam os portáteis pessoais. Cada aula tem a duração de duas horas e comporta sempre uma parte de exposição dos conceitos, digamos, uma parte mais teórica, e uma parte de aplicação desses mesmos conceitos, digamos, uma parte mais prática.

O método de ensino utilizado na componente teórica é o expositivo, que possibilita a transmissão de conhecimentos com continuidade e com um dispêndio mínimo de tempo. Na componente prática da aula, o método mais utilizado é o ativo, suscitando dessa forma a atividade dos alunos através da resolução de exercícios práticos no computador. Espera-se ainda que o aluno realize um conjunto de tarefas nas horas não presenciais.

Alguns pedagogos argumentam que os docentes, especialmente os do ensino superior, dedicam demasiado tempo à exposição oral e, nas últimas décadas, têm sido desenvolvidos vários esforços para se criarem modelos alternativos. A exposição formal da informação continua, no entanto, a prevalecer como o modelo de ensino mais comum e a quantidade de tempo a ela dedicada tem-se mantido estável ao longo dos tempos (Arends, 2008).

As aulas são apoiadas por uma série de diapositivos (slides), organizadas para dar suporte direto ao programa das UCs. Sem constituir uma monografia de apoio às mesmas, este conjunto de diapositivos possibilita aos estudantes bem mais do que uma mera enumeração dos tópicos em estudo, contendo os principais conceitos e uma síntese dos assuntos abordados, tudo isto numa ótica orientada, explícita e diretamente, para os objetivos das UCs. Durante a apresentação é estimulada a intervenção dos estudantes, recorrendo a exercícios práticos que ilustram os objetos em estudo. O espírito crítico e a consequente participação ativa dos estudantes são estimulados pela apresentação de questões ou desafios enquadrados nos tópicos em estudo.

Como veremos no capítulo seguinte, o autor desta Lição tem participado em projetos de investigação nacionais e internacionais que abordam o ensino-aprendizagem da programação no sentido de melhorar a motivação e o desempenho dos alunos. A melhoria pode ser efetuada quer ao nível das ferramentas de suporte ao ensino quer ao nível das metodologias de ensino.

Por se tratar de uma continuação dos conteúdos apreendidos em Programação I também na UC Programação II é implementado o mesmo método de ensino e de aprendizagem. Nesse sentido, existe um esforço por parte da coordenação do DIC em manter o corpo docente nas duas UCs.

2.5 Alternativas de Avaliação

A avaliação implementada nesta UC tem sofrido algumas alterações ao longo dos últimos anos. Tem havido uma preocupação crescente do corpo docente em distribuir a avaliação ao longo semestre concretizada em vários momentos de avaliação. O objetivo principal das avaliações intercalares é «obrigar» os estudantes a trabalharem continuamente, a única forma de aprenderem a programar. Nos últimos anos a avaliação distribuída tem sido realizada em computador, de forma individual, onde é proposto ao aluno um conjunto de problemas para o qual necessita encontrar uma solução e respetiva codificação na linguagem C.

A **Tabela 2.5** apresenta as duas alternativas de avaliação da UC Programação I. A alternativa 1 é a que diz respeito à avaliação distribuída, onde são realizados dois momentos de avaliação que ocorrem ao longo do semestre e um terceiro já na época de exames. A alternativa 2 é aplicada ao exame de recurso e aos exames de época especial, onde a avaliação é efetuada em papel.

Tabela 2.5 – Alternativas de Avaliação

1. Alternativa 1 - (Ordinário, Trabalhador) (Final) - Ficha de Avaliação nº 1 - 30% - Ficha de Avaliação nº 2 - 30% - Ficha de Avaliação nº 3 - 40% (A realizar na Época de Avaliação Final)
2. Alternativa 2 - (Ordinário, Trabalhador) (Recurso, Especial) - Exame Final Escrito - 100%

A avaliação é sempre um dos aspetos mais delicados de qualquer UC, expondo os estudantes a um entendimento sobre a capacidade de envolvimento e apropriação efetiva de saberes aí veiculados. Na verdade, a obtenção da melhor «nota» constitui o catalisador mais relevante na vida de um estudante. Não raras vezes, o estudante procura obter aprovação à UC mais do que obter e solidificar conhecimentos. Obviamente que uma atitude destas irá ter consequências na obtenção de aprovação de UCs subsequentes e quando o estudante se deparar com situações reais de trabalho.

A avaliação é tida, nos tempos atuais, como um ponto de partida privilegiado para o estudo do processo de ensino e aprendizagem, pois, espera-se que se coloque o estudante como sujeito ativo desse processo e não que se o coloque a sujeitar-se a ele (Machado, 2013). Na verdade, abordar a problemática da avaliação é, necessariamente, problematizar muitas das questões essenciais da pedagogia. A procura do melhor modelo de avaliação em termos de justiça, adequabilidade e objetividade é um desafio interessante e que tem preocupado diversos investigadores e pedagogos.

Por haver uma ligação estreita entre as duas UCs, Programação I e Programação II, como explicado anteriormente, entendeu-se aplicar as mesmas alternativas de avaliação. Assim, os estudantes conseguem criar hábitos de trabalhos que vão ao encontro da aprendizagem da programação. De facto, a realização de avaliações em computador «exige» um treino prévio dos estudantes que só é alcançado com empenho e dedicação dentro e fora da sala de aula.

2.6 Bibliografia Recomendada

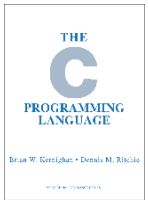
A existência de diversa documentação sobre a linguagem de programação C se por um lado é positiva, por outro, torna difícil a escolha da mais adequada atendendo ao perfil dos

nossos estudantes. Assim, para ambas UCs, Programação I e Programação II, procurou-se encontrar os «melhores» manuais pedagógicos existentes quer em língua Portuguesa quer em língua Inglesa.

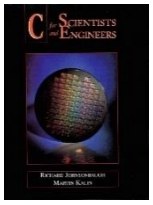
Considera-se desejável que, qualquer estudante destas duas unidades curriculares, independentemente dos conhecimentos específicos que cada uma das unidades promove em profundidade, conheça e consulte com regularidade os seguintes manuais genéricos, como forma de ir desenvolvendo conhecimento da área e sensibilidade para as diversas problemáticas abordadas no âmbito da programação:

Linguagem C
Luís Damas
FCA, 1999
ISBN: 9789727221561
https://www.fca.pt/pt/catalogo/informatica/programacao/linguagem-c/

Introdução à Programação
António Rocha
FCA, 2006
ISBN: 9789727225248
https://www.fca.pt/pt/catalogo/informatica/programacao/introducao-a-programacao-usando-c/



The C Programming Language
Brian W. Kernighan, Dennis M. Ritchie
Prentice-Hall, 2nd Edition, 1988
ISBN: 9780131103627
https://www.pearson.com/us/higher-education/program/Kernighan-C-Programming-Language-2nd-Edition/PGM54487.html



C for Scientists and Engineers
Richard Johnsonbaugh, Martin Kalin
Prentice-Hall, 1997
ISBN-13: 9780023611360
http://catalogue.pearsoned.co.uk/catalog/academic/product/0,1144,0023611367-CRS,00.html

Capítulo 3
Esforços de Melhoria do Ensino-Aprendizagem de Competências de Programação

Este capítulo apresenta os esforços de melhoria do ensino-aprendizagem de competências na área da programação. Após a introdução, é apresentado um projeto de investigação internacional que têm como principal preocupação a melhoria contínua do processo ensino-aprendizagem. Por ser relevante e atual é apresentada uma experiência pedagógica desenvolvida numa instituição de ensino superior Portuguesa alicerçada no ensino baseado em projetos.

3.1 Introdução

O ensino da programação tornou-se ubíquo no ensino superior em todo o mundo. De facto, vários planos de estudos de cursos na área das ciências, tecnologias, engenharia e outras incluem uma ou mais UCs de programação. No entanto, aprender a programar é considerada

uma tarefa complexa e muitas instituições de ensino superior enfrentam o problema das baixas taxas de sucesso em UCs de programação introdutórias.

Qualquer novo método ou técnica de ensino que vise melhorar a aprendizagem da programação seria mais valorizado se pudesse ser aplicado com sucesso em diferentes cursos de diferentes países (Luis M. Alves et al., 2018). No entanto, cada curso é único em termos de estudantes, professor e contexto educacional em que está organizado. Além dos regulamentos nacionais e das políticas seguidas pelas instituições de ensino superior, as decisões dos professores e, às vezes, até as dos alunos, podem afetar a estrutura, os requisitos e os resultados de uma UC.

De qualquer forma, a comparação entre UCs, que pode não ser uma tarefa fácil, mesmo para duas UCs sobre a mesma temática, pode fornecer informações úteis para planear melhorias no ensino. Para que uma melhoria seja geralmente aplicável, ela precisa ser avaliada em diferentes ambientes que já foram estudados e suficientemente compreendidos (Luis M. Alves et al., 2018).

As dificuldades nas UCs de programação introdutórias foram categorizadas como relacionadas à natureza da programação, métodos de ensino, métodos de estudo, habilidades e atitudes dos alunos ou desafios psicológicos enfrentados pelos alunos (A. Gomes & Mendes, 2007). Por exemplo, num inquérito efetuado a estudantes em Portugal, a falta de esforço ou persistência pessoal e a falta de motivação foram alguns dos motivos comuns relatados como dificuldades de aprendizagem (A. J. Gomes, Santos, & Mendes, 2012). Noutros estudos envolvendo estudantes em Portugal, verificou-se que as notas finais na UC de programação introdutória estavam fortemente correlacionadas com as notas finais obtidas nas UCs de matemática e que os alunos que falharam na UC de programação introdutória também tinham fraca compreensão dos conceitos matemáticos básicos e dificuldades na resolução de problemas (A. J. Gomes & Mendes, 2010).

De acordo com os resultados obtidos em vários estudos, alguns dos quais referidos anteriormente, é evidente que o processo de ensino-aprendizagem da programação é exigente

e que alunos e professores podem enfrentar um conjunto variado de dificuldades na melhoria deste processo.

3.2 Projeto Bilateral entre Portugal e a Sérvia

Nesta secção é descrito um projeto de investigação[4] ainda em curso referente ao biénio 2018/2019 entre instituições do ensino superior de Portugal e da Sérvia em que o autor desta Lição é membro participante. Trata-se de um projeto no âmbito da "Cooperação Transnacional"[5] sob a alçada de um "Acordo entre Portugal e a Sérvia", cujos principais promotores são a Fundação para a Ciência e a Tecnologia (FCT) do lado Português e o Ministério da Educação, Ciência e Desenvolvimento Tecnológico do lado Sérvio. Em destaque, o edital do concurso refere que esta cooperação bilateral visa o intercâmbio de investigadores no âmbito de projetos comuns de investigação.

O projeto referido tem como título "*Data mining based evaluation of IT teaching practice in Portugal and Serbia*" e tem como instituições participantes, a Universidade do Minho (Departamento de Informática), o Instituto Politécnico de Bragança (ESTiG) e a *University of Novi Sad* (*Faculty of Technical Sciences*).

O grupo de investigação Português envolvido no projeto inclui dois professores com o grau de Doutor e dois estudantes de doutoramento e o grupo de investigação Sérvio inclui um Professor com o grau de Doutor e três estudantes de doutoramento. Todos os membros envolvidos no projeto têm experiência na lecionação de cursos diversos no âmbito das Linguagens de Programação o que leva a que reconheçam e se preocupem com a dificuldade que a maioria dos estudantes tem neste tipo de UCs.

[4] https://www.fct.pt/apoios/cooptrans/servia/docs/mapa_Servia_18-19_financiados.pdf
[5] https://www.fct.pt/apoios/cooptrans/servia/

3.2.1 Contextualização

No mercado de trabalho de hoje há uma necessidade crescente de programadores de computadores. Esta procura coloca os sistemas de educação sob pressão para ensinar programação a cada vez mais estudantes sem aumentar os recursos disponíveis. Nestas circunstâncias e perante turmas com elevado número de estudantes, os professores têm tendência para seguir métodos tradicionais de ensino o que pode levar a uma redução da qualidade de ensino e a um substancial aumento de insucesso escolar.

O uso de software de apoio ao ensino, como é o caso dos *Learning Management Systems* (LMS), pode dar uma resposta parcial a este problema. Este tipo de sistemas permite disponibilizar materiais, fornecer instruções aos estudantes, dar respostas, enfim, guiar os estudantes durante o processo de aprendizagem, aumentando a motivação do estudante na perseguição de soluções e proporcionando uma forma de avaliação das competências adquiridas.

Estes sistemas embora de uso geral, podem ser usados com grande facilidade e sucesso nas UCs de programação. Além disso, se expandidos com uma componente inteligente permitem desenvolver tarefas de tutoria que, para além de evitar a presença física permanente dos professores, aumentam a capacidade de estudo autónomo proporcionando ao aluno um contexto motivacional e conjunto de ajudas criteriosamente escolhidas.

Os dados recolhidos automaticamente pelos LMS podem ser analisados e minerados de forma a fornecerem dados sobre a atividade dos estudantes. A integração do processo de *Data Mining* (DM) no LMS permite obter informações importantes sobre o progresso dos estudantes e sobre a qualidade dos cursos.

3.2.2 Objetivos

O principal objetivo deste projeto de investigação é o de melhorar o processo do ensino-aprendizagem da programação em Portugal e na Sérvia através da mineração dos dados da aprendizagem recolhidos automaticamente por um LMS devidamente artilhado.

Concretamente, os objetivos a alcançar com o projeto são:

1) Preparar um LMS adaptando-o para o caso específico do ensino da programação ao nível do ensino superior;

2) Aperfeiçoar o LMS escolhido com capacidade para realização de tarefas de DM sobre os dados automaticamente recolhidos;

3) Introduzir o LMS adaptado e aperfeiçoado em UCs relacionadas com o ensino de Paradigmas e Linguagens de Programação selecionadas em cada uma das instituições parceiras;

4) Realizar experimentos com o dito LMS nos cursos selecionados;

5) Descobrir padrões de comportamento de aprendizagem demonstrados pelos estudantes durante o uso do LMS recorrendo a técnicas de mineração de dados. O propósito é comparar os resultados extraídos entre instituições e entre países distintos.

Enquadrado neste projeto, a utilização de ferramentas e tutores inteligentes que melhorem o processo de ensino-aprendizagem da programação foi uma preocupação constante do grupo de investigação.

3.2.3 Estudo Prévio da Situação Atual das UCs de Programação

Numa primeira fase o projeto começou por um estudo prévio da situação atual relativamente aos dados de sucesso escolar das UCs de programação das instituições de ensino superior de ambos os países. Nesta fase procurou-se identificar as causas do problema do insucesso. Neste sentido, foram efetuados e disponibilizados aos estudantes de UCs de programação, dois questionários que nos permitiram recolher alguma informação sobre esta problemática. O objetivo da análise desta informação seria definir qual a estratégia a seguir no desenvolvimento de um novo sistema de apoio ao ensino-aprendizagem das linguagens de programação. De notar que, este tópico enquadra-se completamente no projeto de doutoramento

de um dos doutorandos que integram a equipa portuguesa, sendo também uma área de aplicação dos estudos realizados pelo outro doutorando.

Nesta primeira fase, com base nos dados recolhidos sobre as UCs de programação, foi publicado o artigo com o título "*A Comparison of Introductory Programming Courses between Portugal and Serbia*" (Luis M. Alves et al., 2018) na *International Conference on Applied Internet and Information Technologies* (AIIT 2018)[6] em outubro de 2018 em Bitola na República da Macedónia. O objetivo do estudo publicado neste artigo foi o de adquirir mais conhecimento sobre as diferenças ao nível do desempenho académico, ensino e avaliação das UCs de programação introdutórias em Portugal e na Sérvia. É expectável que os resultados obtidos neste estudo sirvam de base para pesquisas subsequentes sobre como melhorar a atual prática de ensino nos dois países.

Neste estudo foram recolhidos dados anónimos sobre um grupo de UCs de programação do ensino superior em Portugal e na Sérvia. Por uma questão de manter o anonimato, a identificação dos estudantes nunca foi solicitada em momento algum. Os anos letivos para os quais foram recolhidos os dados vão desde 2013/2014 a 2016/2017. O conjunto de dados referem-se a duas instituições de Portugal, codificadas como PI-A (Universidade do Minho) e PI-B (Instituto Politécnico de Bragança), e uma instituição da Sérvia (*University of Novi Sad*), codificada como SI-A.

No total foram analisadas quinze UCs codificadas da seguinte forma:

- PC-A1 a PC-A7 a que correspondem sete UCs lecionadas na instituição PI-A que comportam no total quatro programas curriculares diferentes;
- PC-B1 a PC-B5 a que correspondem cinco UCs lecionadas na instituição PI-B que comportam no total três programas curriculares diferentes;
- SC-A1 a SC-A3 a que correspondem três UCs lecionadas na instituição SI-A que comportam três programas curriculares diferentes.

[6] https://aiitconference.org/2018/

Para cada UC foram recolhidos os seguintes dados: ano letivo, código da instituição, código do curso, código da UC, linguagem de programação lecionada, número de alunos inscritos, valores de três indicadores, métodos de ensino e métodos de avaliação predominantemente utilizados na UC.

Os três indicadores que consideramos relevantes para avaliar o desempenho dos estudantes foram:

- Aprovados versus Avaliados – dá-nos o rácio do número de estudantes aprovados em relação ao número de estudantes avaliados à UC;
- Aprovados versus Inscritos – dá-nos o rácio do número de estudantes aprovados em relação ao número de estudantes inscritos à UC;
- Avaliados versus inscritos – dá-nos o rácio do número de estudantes avaliados em relação ao número de estudantes inscritos à UC.

Estes três indicadores foram também usados como métrica para avaliar o desempenho dos estudantes ao longo de vários anos académicos consecutivos num estudo efetuado em Portugal (Mendes, Paquete, Cardoso, & Gomes, 2012).

A definição do estado de ser avaliado foi relativamente diferente entre os dois países. Assim, para as UCs analisadas na Sérvia, ou seja, as UCs de SC-A1 a SC-A3, um aluno foi considerado avaliado apenas se tivesse obtido uma classificação superior a zero em pelo menos duas avaliações diferentes. Essa definição foi adotada porque as UCs SC-A1 a SC-A3 tiveram várias avaliações intercalares ao longo do semestre. Em Portugal também as UCs têm várias avaliações intercalares, no entanto, não foi considerada a restrição dos estudantes necessitarem de ter uma classificação superior a zero para serem considerados avaliados.

Devido às limitações dos dados disponíveis sobre as UCs lecionadas na Sérvia, os valores dos três indicadores de desempenho tiveram que ser calculados de forma diferente para as UCs SC-A1 a SC-A3, ou seja, apenas os estudantes que estavam inscritos na UC pela primeira

vez foram considerados e os valores dos indicadores que dependem do número de estudantes avaliados tiveram que ser baseados em determinadas estimativas.

Para o nosso estudo consideramos os seguintes métodos de ensino:

- 1 - Resolução de exercícios em papel, ou seja, os estudantes não resolvem os exercícios usando os computadores, mas sim usando lápis e papel;
- 2 - Resolução de exercícios usando o computador, ou seja, os estudantes usam um editor e um compilador de linha de comando ou um ambiente de desenvolvimento integrado (IDE) para resolver os exercícios;
- 3 – Aplicação do Ensino baseado em projetos (PBL), isto é, os projetos são desenvolvidos durante as aulas e os conceitos de programação são introduzidos quando necessário no projeto;
- 4 - Aplicação de um Sistema de Gestão de Aprendizagem (LMS), isto é, uma plataforma de software é usada para gerir o trabalho dentro e fora da sala de aula. Basicamente, um LMS é uma plataforma de ensino online projetada a partir de uma metodologia pedagógica com o intuito de promover e disseminar a educação através da modalidade de ensino à distância;
- 5 – Utilização de editores orientados à sintaxe, isto é, os editores orientados à sintaxe são usados pelos estudantes quando programam dentro e fora da sala de aula;
- 6 – Utilização de depuradores (*debuggers*), ou seja, os depuradores são usados recorrentemente pelos estudantes no desenvolvimento de programas.

Quanto à avaliação do desempenho dos estudantes foram considerados os seguintes métodos de avaliação:

- 1 - Avaliação baseada em projetos, ou seja, os estudantes desenvolvem o código para resolver um problema e depois enviam ou apresentam o código ao professor;
- 2 - Trabalho em grupo ou individual, ou seja, o código é produzido por um estudante ou por um grupo de estudantes;

- 3 - Avaliação intercalar com exercícios efetuados em papel, ou seja, a avaliação decorre em períodos estabelecidos do semestre letivo em que os estudantes usam lápis e papel para resolver os exercícios de programação propostos;
- 4 - Avaliação intercalar com exercícios efetuados no computador, ou seja, a avaliação ocorre em períodos estabelecidos do semestre em que os estudantes, de forma individual, usam os computadores da escola para implementar os programas dos exercícios propostos;
- 5 - Avaliação por exame, ou seja, os alunos fazem um exame na época de exames, usando computadores ou lápis e papel;
- 6 - Avaliação automatizada, ou seja, o professor usa uma ferramenta especial para avaliar automaticamente o código do estudante;
- 7 - Avaliação por pares, ou seja, o código do estudante é avaliado por outros estudantes;
- 8 - Avaliações múltiplas, ou seja, os alunos têm várias avaliações ao longo do semestre académico.

Como pode ser observado pela **Tabela 3.1** em cada uma das UCs objeto do presente estudo foram usados múltiplos métodos de ensino e de avaliação.

Tabela 3.1 – Visão global das UCs de Programação

Código da Instituição	Código da UC	Linguagem de Programação	Método de Ensino	Método de Avaliação
PI-A	PC-A1	Haskell	2, 3	3, 4, 5
PI-A	PC-A2	Haskell	2, 3	3, 4, 5
PI-A	PC-A3	Haskell	2, 3	3, 4, 5
PI-A	PC-A4	C	2, 3	3, 4, 5
PI-A	PC-A5	C	2, 3	3, 4, 5
PI-A	PC-A6	C	2, 3	3, 4, 5
PI-A	PC-A7	C	2, 3	2, 5
PI-B	PC-B1	C	2, 5	3, 4, 5
PI-B	PC-B2	C	2, 5	3, 4, 5
PI-B	PC-B3	C	2, 5, 6	3, 4, 5
PI-B	PC-B4	C	2, 5	3, 5
PI-B	PC-B5	MATLAB	2, 5	3, 4, 5
SI-A	SC-A1	C	2, 5	5, 8
SI-A	SC-A2	C	2, 5	5, 8
SI-A	SC-A3	C	2, 5	5, 8

Por uma questão de manter o anonimato, pois a informação aqui apresentada é algo delicada, optou-se por manter codificada a identificação dos cursos e das UCs da Universidade do Minho e da *University of Novi Sad*. Em relação às UCs e respetivos cursos lecionados na ESTiG, optou-se pela sua identificação, pois, trata-se de dados internos obtidos pelo autor desta Lição. Assim, a **Tabela 3.2** mostra a correspondência entre os códigos da **Tabela 3.1** e as UCs lecionadas na ESTiG.

Tabela 3.2 – UCs de Programação da ESTiG objeto de estudo

Código da Instituição	Código da UC	UC	Curso
PI-B	PC-B1	Programação I	Licenciatura em Engenharia Informática
PI-B	PC-B2	Programação I	Licenciatura em Informática de Gestão
PI-B	PC-B3	Programação II	Licenciatura em Engenharia Informática
PI-B	PC-B4	Programação II	Licenciatura em Informática de Gestão
PI-B	PC-B5	Algoritmia e Programação	Licenciatura em Engenharia Mecânica

Na **Tabela 3.1**, existem alguns dados básicos sobre cada UC em análise, principalmente, a linguagem de programação ensinada e os métodos de ensino e avaliação aplicados. Assim, a linguagem de programação dominante é o C, seguida por Haskell e MATLAB. O domínio da linguagem C nos cursos de programação coincide com o que outros investigadores já encontraram em estudos anteriores. Num estudo sobre UCs de programação no ensino superior europeu, concluiu-se que o C é a linguagem de programação mais frequentemente ensinada, com forte vantagem sobre as outras linguagens de programação, especialmente em UCs de ensino introdutório (Aleksic & Ivanovic, 2016).

Em ambos os países, os métodos de ensino envolveram exercícios efetuados em computador e, em menor grau, o uso de editores orientados à sintaxe. Por outro lado, o PBL foi aplicado apenas em Portugal, ou seja, apenas nos cursos da instituição PI-A (Universidade do Minho).

Neste estudo, também foi encontrada alguma variação entre as escolhas dos métodos de avaliação. Em ambos os países, foi usada a avaliação por exame, mas a implementação da avaliação intercalar ao longo do período diferiu. Em Portugal, esta avaliação ocorre em datas e em número bem definidas, enquanto que na Sérvia existem várias avaliações organizadas ao longo de todo o semestre.

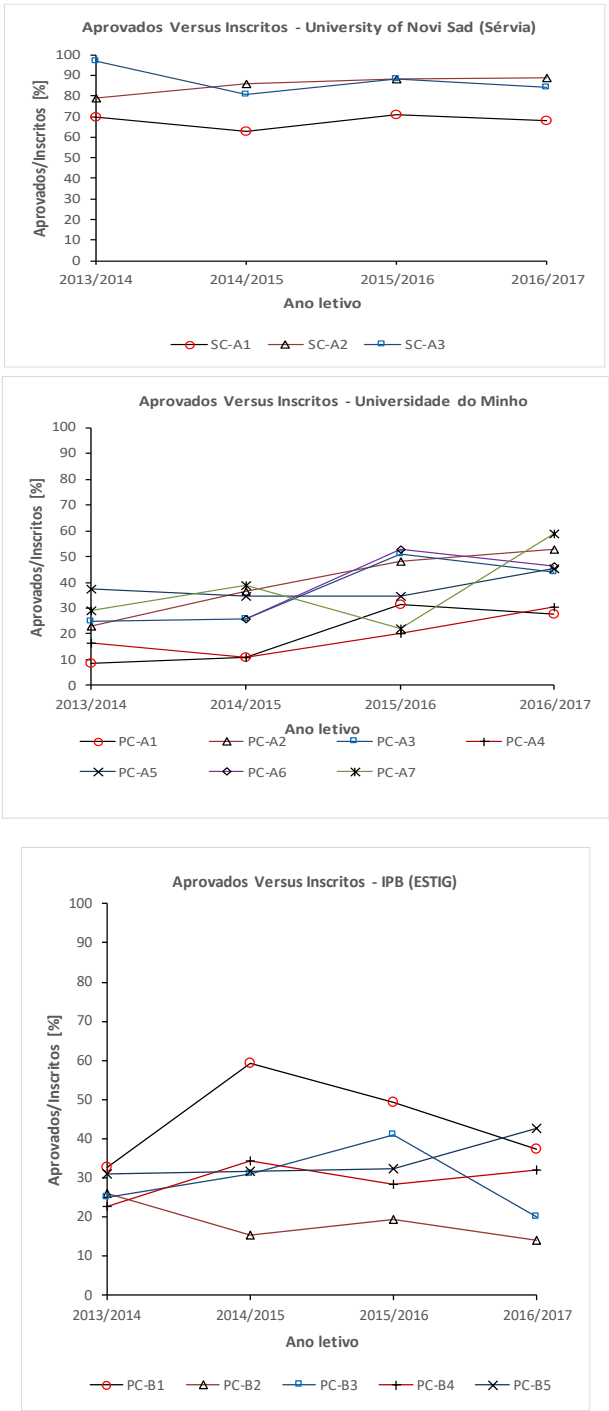


Figura 3-1: Alunos Aprovados Versus Inscritos nas UCs das três instituições

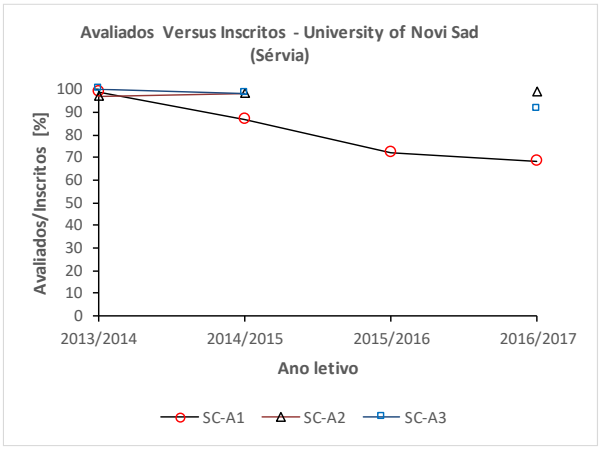
Na **Figura 3-1**, podemos observar o rácio entre os estudantes aprovados em relação aos estudantes inscritos para cada UC ao longo dos vários anos letivos analisados. Os dois últimos gráficos da figura mostram os dados relativos às instituições de Portugal, ao passo que, o gráfico na parte superior da figura mostra os dados relativos à instituição da Sérvia. Assim, podemos verificar que as duas instituições em Portugal têm rácios de aprovação semelhantes, as quais podem ser classificados como baixos a moderados. Por outro lado, o rácio de estudantes aprovados em relação aos estudantes inscritos é geralmente melhor nas UCs dos cursos da Sérvia, os quais podem ser classificados como altos.

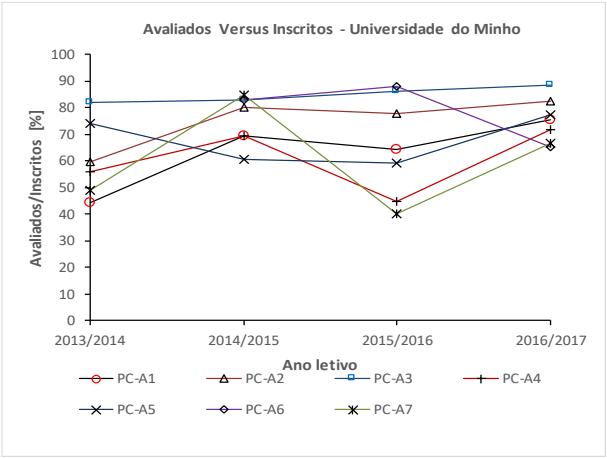
Como os dados compilados não incluem as notas de avaliação dos estudantes, não podemos especular se as taxas de aprovação mais altas verificadas na Sérvia foram ou não o resultado de um conhecimento potencialmente melhor dos seus estudantes.

Numa pesquisa sobre as das taxas de aprovação e reprovação nos cursos de programação introdutórios, a taxa de aprovação média relatada foi de 72% em que todas UCs foram igualmente ponderadas (Bennedsen & Caspersen, 2019). Em relação a esse valor, as taxas de aprovação podem ser consideradas acima da média para a instituição da Sérvia e abaixo da média para as instituições em Portugal.

Pelo menos em parte, a diferença encontrada nas taxas de aprovação entre Portugal e a Sérvia deve-se ao facto das taxas de desempenho das UCs da Sérvia contabilizarem apenas os estudantes que se inscreveram na UC pela primeira vez num determinado ano letivo. No entanto, com base no conhecimento obtido a partir de UCs de anos letivos anteriores na Sérvia, pode-se supor que, mesmo que todos os estudantes inscritos fossem considerados, as taxas de aprovação na Sérvia seriam ainda assim, em média, ainda mais altas do que em Portugal. Neste outro cenário, as taxas de aprovação na Sérvia, provavelmente, teriam de ser movidas para a categoria moderada a alta.

A **Figura 3-2** mostra o rácio dos estudantes avaliados em relação aos estudantes inscritos a cada uma das UCs. Como esperado, os valores do rácio Avaliados versus Inscritos são um





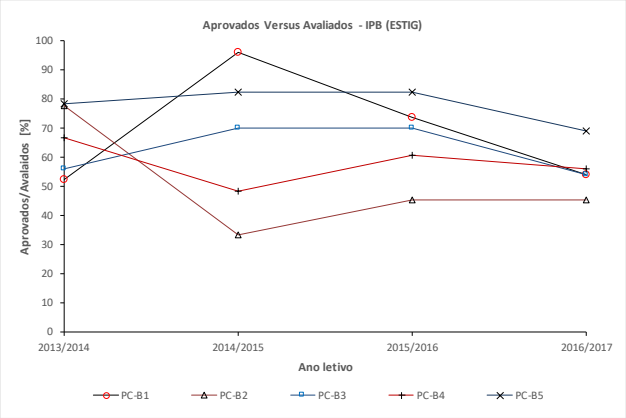


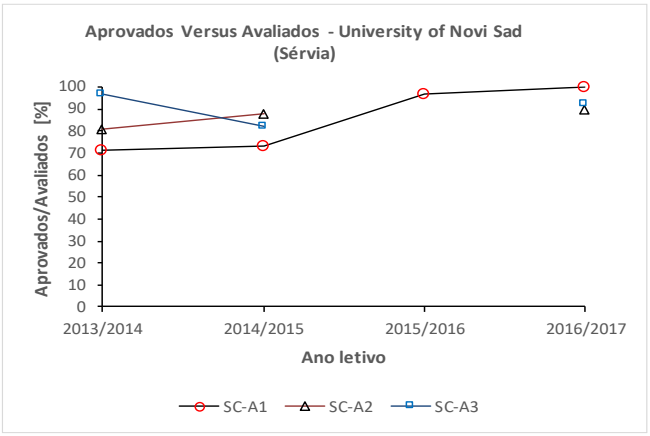
Figura 3-2: Alunos Avaliados versus Inscritos nas UCs das três instituições

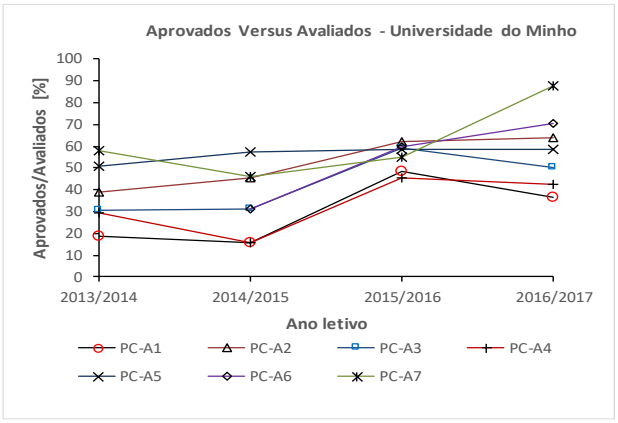
pouco mais altos que os valores correspondentes do rácio Aprovados versus Inscritos. As diferenças na relação Avaliados versus Inscritos entre as instituições em Portugal e na Sérvia são menos pronunciadas do que na relação Aprovados versus Inscritos. No entanto, essas diferenças ainda são visíveis e a instituição da Sérvia apresenta taxas de avaliação mais altas. Os comentários anteriores sobre a influência da construção da amostra nos valores aprovados versus inscritos também podem ser aplicados à interpretação dos valores obtidos na relação avaliados versus inscritos.

A maior diferença entre os rácios Aprovados versus Inscritos e Avaliados versus Inscritos podem ser observadas para a instituição PI-A (Universidade do Minho). Essa diferença pode ser mais facilmente observada na **Figura 3-3**, onde são apresentados os rácios Aprovados versus Avaliados. Com base nesses valores, pode-se concluir que o rácio de Aprovados versus Avaliados é notavelmente mais elevado na Sérvia do que em Portugal.

Em resumo, as principais conclusões que podemos retirar da comparação das UCs na área da programação dos dois países são:

- a linguagem C é a principal escolha para as UCs de programação introdutórias em Portugal e na Sérvia;
- os métodos de ensino diferem entre os dois países, embora a resolução de exercícios usando o computador sejam praticados nos dois países, o PBL é implementado apenas em Portugal;
- os métodos de avaliação diferem entre os dois países, embora os exames sejam praticados pelos dois países, a avaliação intercalar é geralmente preferida em Portugal e múltiplas avaliações ao longo do semestre seja preferida na Sérvia;
- as taxas de avaliação são mais altas na Sérvia;
- as taxas de aprovação são mais altas na Sérvia, mesmo quando apenas os alunos avaliados são considerados.





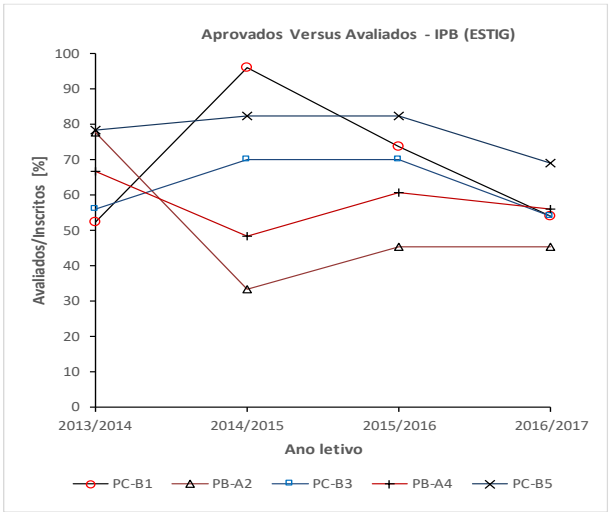


Figura 3-3: Aprovados versus Avaliados nas UCs das três instituições

A distinção mais relevante entre as UCs dos dois países é a diferença nas taxas de aprovação e nas taxas de avaliação. As potenciais causas que podem explicar esta disparidade podem residir nas diferenças dos métodos de ensino, nos requisitos das UCs, no nível de motivação e nas atitudes dos estudantes. Outro fator pode ser a falta de *background* anterior em Informática, concretamente na área da programação.

A introdução das ciências da computação no ensino básico (Passey, 2017) poderia contribuir para uma melhor preparação dos futuros estudantes, o que poderia reduzir os problemas atuais. Com este e projetos futuros pretendemos melhorar as práticas atuais do ensino da programação, o que implica uma análise mais detalhada dos problemas identificados.

3.2.4 Conhecimento, Expectativas e Motivação dos Estudantes

Nesta subsecção são apresentados dados relevantes sobre o conhecimento, as expectativas e a motivação dos estudantes quando iniciam UCs de Programação introdutórias em Portugal e na Sérvia.

A programação é uma competência necessária em várias disciplinas STEM (*Science, Technology, Engineering e Mathematics*) e atualmente em variadíssimos empregos esta competência é muito valorizada. No entanto, é conhecido que os estudantes sentem dificuldades em UCs de programação introdutória. O desempenho académico em programação pode ser influenciado por vários fatores e pode variar entre países, conforme observado na subsecção anterior.

No sentido de obter dados que nos permitissem obter alguns resultados relevantes, um questionário anónimo com perguntas fechadas foi preenchido por estudantes inscritos em UCs de programação em instituições de ensino superior em Portugal e na Sérvia. Após o descarte de dados incorretos, foram consideradas as respostas de 678 estudantes. Seguidamente, os dados foram analisados quantitativamente para identificar as características gerais dos grupos investigados, bem como as diferenças entre os grupos dos dois países.

Geralmente, os estudantes inquiridos manifestaram elevadas expectativas e motivações em relação à programação, no entanto, o seu conhecimento inicial nesta área era geralmente de um nível baixo. Em média, os grupos dos dois países eram semelhantes. As principais diferenças residem no conhecimento inicial, um pouco mais alto para estudantes da Sérvia e expectativas um pouco mais altas para estudantes de Portugal. Estes resultados constituem uma base para uma investigação mais aprofundada das causas das variações observadas anteriormente no desempenho dos estudantes entre Portugal e Sérvia.

Como existem muitos pontos em comum entre os estudantes de Portugal e a Sérvia, um futuro projeto de investigação poderia incluir a aplicação de novas metodologias de ensino e ferramentas que potenciassem o ensino-aprendizagem da programação.

Conhecer as características e atitudes dos estudantes, é sem dúvida, relevante para compreendermos o desempenho académico no estudo da programação. O estudo apresentado a seguir analisa o conhecimento, as expectativas e a motivação inicial dos estudantes como fatores que poderiam ser usados para explicar as diferenças de desempenho observadas em estudos anteriores (Luis M. Alves et al., 2018).

A aprendizagem de programação antes da entrada no ensino superior está associada a um maior desempenho académico inicial em UCs relacionadas com as Tecnologias da Informação e da Comunicação (TIC) (Kori, Pedaste, Leijen, & Tõnisson, 2016).

As expectativas não cumpridas dos estudantes podem levar a várias deceções e até mesmo ao abandono do ensino superior (James, 2002). Os estudantes podem ter expectativas elevadas em relação ao conteúdo do curso, em particular ao seu rigor académico e ao alinhamento com o mercado de trabalho (Kandiko & Mawer, 2013). Além disso, as expectativas podem variar de acordo com o género, conforme demonstrado num estudo sobre as perceções dos estudantes sobre um curso relacionado com a computação (Li, Records, & Fougere, 2004).

A necessidade de motivar os estudantes para a aprendizagem da programação já foi reconhecida em estudos anteriores (Robins, Rountree, & Rountree, 2003). Além disso, numa

UC de programação introdutória foi descoberta uma relação positiva entre a motivação intrínseca e o desempenho da aprendizagem da programação num grupo de estudantes (Bergin & Reilly, 2005).

No estudo (Luis M. Alves et al., 2019), em que o autor desta Lição participou, e que é apresentado nesta subsecção foi efetuado um método de investigação quantitativo. Neste estudo, procurou-se investigar o conhecimento, as expectativas e a motivação inicial dos estudantes em UCs de programação introdutórias em quatro instituições de ensino superior, três de Portugal e uma da Sérvia. Os dados quantitativos coletados foram primeiramente processados e depois analisados usando métodos quantitativos.

Os estudantes responderam a um questionário em inglês, sendo-lhes assegurado o anonimato. O questionário consistiu em 21 questões fechadas, agrupadas em quatro categorias:

- CAT0: quatro questões introdutórias sobre o *background* do estudante, por exemplo, curso em que está matriculado, formação anterior;
- CAT1: três questões sobre o conhecimento inicial de programação, por exemplo, familiaridade com conceitos de algoritmia, conhecimento de linguagens de programação;
- CAT2: seis questões sobre as expectativas em relação à UC de programação introdutória, por exemplo, para aprender uma linguagem de programação, para desenvolver software;
- CAT3: oito questões sobre a motivação em relação à programação, por exemplo, preferência pelo uso de algoritmos para especificar o raciocínio, satisfação em encontrar uma solução para o problema.

Todas as questões eram de resposta obrigatória. Uma escala de resposta de 5 pontos (1 - Nulo, 2 - Reduzido, 3 - Médio, 4 - Alto, 5 - Excelente) foi usada para as questões da CAT1,

enquanto uma escala de resposta de 3 pontos (1 - Discordo, 2 - Neutro, 3 - Concordo) foi utilizado para as questões da CAT2 e CAT3. O questionário foi respondido a meio do semestre do ano letivo de 2018/2019. No total, foram inquiridos 736 estudantes (407 em Portugal e 329 na Sérvia).

O conjunto de dados inicial foi baseado em 22 variáveis com 736 registos, um registo para cada inquirido. Uma variável registou o país do inquirido, enquanto as 21 variáveis restantes registaram as respostas às 21 questões do inquérito.

Todos os registos que continham um valor de resposta inválido para alguma questão não relacionado à inscrição na UC foram excluídos do conjunto de dados. Um valor de resposta foi considerado inválido se o inquirido forneceu um valor de resposta não listado no questionário, forneceu vários valores de resposta ou não forneceu nenhum valor de resposta. Após a exclusão dos registos inválidos, foram considerados 678 registos no conjunto de dados, 389 registos de inquiridos de Portugal e 289 registos de inquiridos da Sérvia.

Com o objetivo de detetar diferenças nas respostas entre Portugal e a Sérvia, o conjunto de dados foi ampliado adicionando três novas variáveis: *Conhecimento Inicial Médio*, *Expectativa Média* e *Motivação Média*, que correspondem às respostas médias às questões das categorias CAT1, CAT2 e CAT3, respetivamente.

As diferenças nas respostas foram examinadas primeiro pela comparação de médias (M), desvios-padrão (DP), medianas (Mdn) e distribuições das variáveis *Conhecimento Inicial Médio*, *Expectativa Média* e *Motivação Média*. As diferenças nas respostas entre os dois países foram avaliadas por testes estatísticos. A independência entre a variável do país e cada variável de resposta das questões das categorias CAT1, CAT2 e CAT3 foi testada usando o teste exato de Fisher.

Todas as análises dos dados foram conduzidas usando a linguagem e o ambiente R para computação estatística[7], o ambiente de desenvolvimento integrado *RStudio Desktop Open*

[7] O projeto R para a computação estatística, URL: https://www.r-project.org

Source Edition[8] e os seguintes pacotes R adicionais com suas dependências[9]: *tables* e *RColorBrewer*.

Conforme evidenciado pela estatística descritiva apresentada na **Tabela 3.3**, os estudantes inquiridos em Portugal e na Sérvia possuem baixo conhecimento inicial de programação (M = 2.25, Mdn = 2.00, Escala 1–5), mas têm expectativas (M = 2.71, Mdn = 2.83, Escala 1–3) e motivação (M = 2.63, Mdn = 2.62, Escala 1–3) elevadas.

Tabela 3.3 – *Conhecimento Inicial Médio*, *Expetativa Média* e *Motivação Média*

		Conhecimento Inicial Médio (Escala 1–5)			***Expectativa Média*** (Escala 1–3)			***Motivação Média*** (Escala 1–3)		
País	**n**	**M**	**SD**	**Mdn**	**M**	**SD**	**Mdn**	**M**	**SD**	**Mdn**
Portugal	389	2.14	0.84	2.00	2.72	0.34	2.83	2.64	0.29	2.62
Sérvia	289	2.38	0.94	2.33	2.70	0.26	2.67	2.62	0.24	2.62
Total	678	2.25	0.89	2.00	2.71	0.31	2.83	2.63	0.27	2.62

A maior diferença entre os dois países está no conhecimento inicial médio, que é mais elevado na Sérvia, seguido pela diferença nas expectativas médias, que são apenas um pouco mais elevadas em Portugal. No entanto, as diferenças no global das respostas entre Portugal e Sérvia parecem ser pequenas. Isso também pode ser observado na **Figura 3-4**, que contém gráficos de caixas com entalhe para as respetivas variáveis.

Em relação às respostas às oito questões da CAT3 relativas à inquirição da motivação dos estudantes em relação à aprendizagem da programação, apenas duas apresentaram diferenças significativas. Essas questões são: "*Do you find challenging to code algorithms and run programs?*" e "*Do you like the 'trial & error' method to develop solutions?*". A distribuição das respostas a essas duas questões pode ser visualizada na **Figura 3-5** usando gráficos em mosaico.

A principal diferença nas respostas às duas questões é a observação de uma maior concordância dos estudantes portugueses sobre o desafio que representa a codificação de al-

[8] Open source and enterprise-ready professional software for data science - RStudio, URL: https://www.rstudio.com
[9] The comprehensive R archive network, URL: https://cran.r-project.org

goritmos e a execução programas e pela preferência do método de tentativa e erro no desenvolvimento de soluções. Na grande maioria das questões analisadas (15 em 17), a variação nas respostas entre os entrevistados de Portugal e da Sérvia não foi estatisticamente significativa, o que reforça a observação de que os entrevistados dos dois países são geralmente semelhantes em termos de conhecimento inicial, expectativas e motivação.

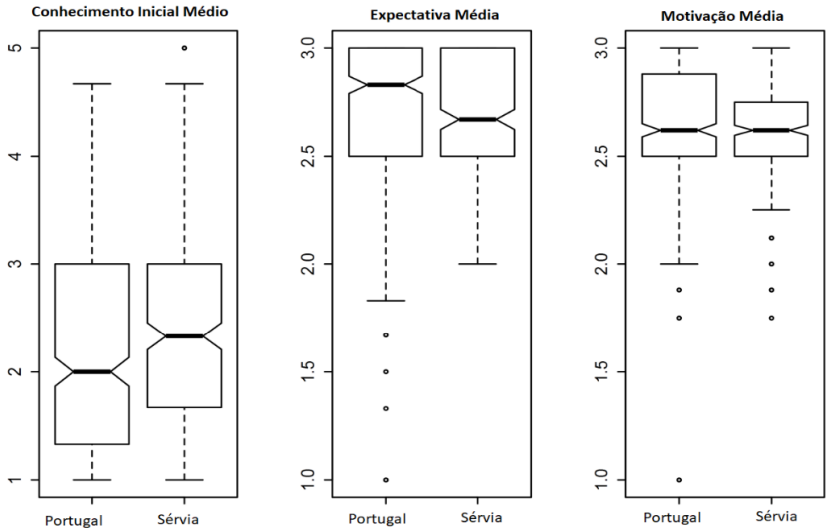


Figura 3-4: Gráfico com o *Conhecimento Inicial Médio*, *Expetativa Média* e *Motivação Média*

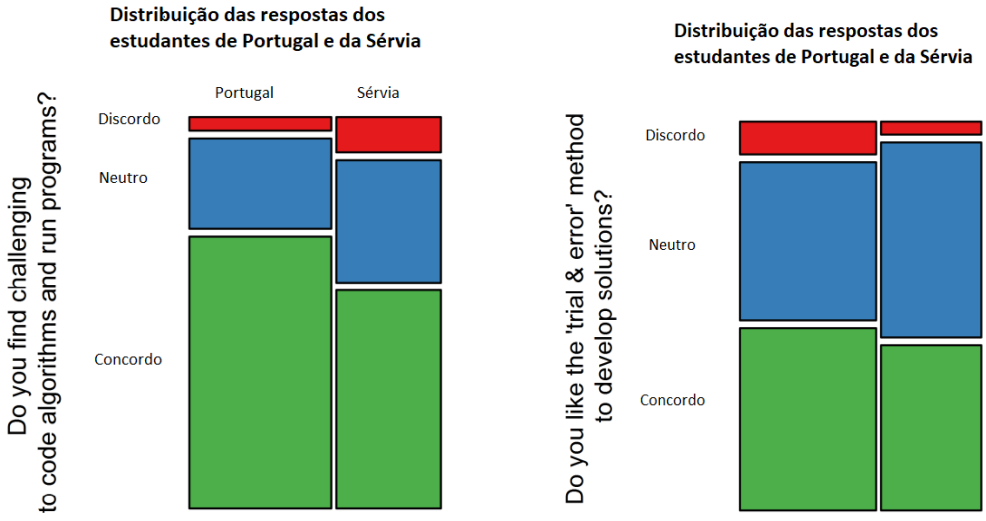


Figura 3-5: Gráfico em mosaico com respostas relativas à Motivação dos estudantes

Em conclusão, os estudantes de Portugal e da Sérvia inquiridos reportaram baixo conhecimento inicial sobre programação, mas elevadas expetativas em relação à UC de programação introdutória e elevada motivação em relação à programação em geral.

De um modo geral, as diferenças de opinião entre os estudantes inquiridos dos dois países não são significativas. Em particular, há a assinalar um maior conhecimento inicial dos estudantes Sérvios e expetativas médias ligeiramente mais elevadas dos estudantes Portugueses. Em relação à motivação, as respostas que evidenciaram diferenças significativas dizem respeito ao aspeto desafiador da programação e ao uso do método de tentativa e erro no desenvolvimento de soluções.

O ensino-aprendizagem de programação tem sido uma preocupação crescente quer a nível nacional quer a nível internacional. O contacto com a programação em fases inicias do processo educativo com certeza que potenciaria um maior sucesso na aquisição desta competência nos diversos cursos do ensino superior. Neste sentido, existem já, algumas ações de melhoria do ensino da programação a nível nacional e internacional, como é o caso da iniciativa de melhoria das competências digitais *Portugal INCoDe.2030*[10] e o movimento EU Code Week[11].

3.3 Ferramentas de Apoio ao Ensino da Programação

No âmbito do projeto referido na secção anterior, foi introduzida nas aulas de Programação II uma ferramenta de apoio ao ensino-aprendizagem da programação. Tratasse do *Python Tutor*[12], uma ferramenta criada por Philip Guo[13] para ajudar todo o público interessado em aprender a 'arte' de programar.

[10] Portugal INCoDe.2030, URL: https://www.incode2030.gov.pt
[11] Europe Code Week, URL: https://codeweek.eu
[12] http://www.pythontutor.com/
[13] http://pgbovine.net/

Para além de C, o *Python Tutor* permite a compilação e execução de programas *online* em Python, Java, C++, JavaScript e Ruby. A possibilidade de executar o programa linha a linha e a visualização das diferentes estruturas de dados do programa são as principais vantagens da utilização desta ferramenta. Como principal desvantagem é de referir a impossibilidade de utilizar funções de leitura de dados, obrigando a uma inicialização prévia das variáveis.

O *Python Tutor* pode ser acedido a partir de qualquer *browser*. Cada vez que iniciamos uma sessão, é-nos atribuído um código de utilizador (*user*) a partir do qual podemos entrar num *Chat* e questionar a comunidade que está *online* sobre determinado assunto do nosso programa. Normalmente, não é difícil obter ajuda por parte de algum utilizador que esteja a usar a mesma linguagem de programação. Outro aspeto positivo da ferramenta é a possibilidade de praticar programação colaborativa.

A **Figura 3-6** mostra a execução de um programa em C no *Phyton Tutor*. Tratasse de um programa que determina o maior valor existente num vetor. Na parte esquerda da figura podemos ver o código onde a seta de cor verde (seta de cima) indica a instrução executada e a seta a vermelho (seta de baixo) indica a próxima instrução a executar. No lado direito da figura em cima podemos observar uma janela de output onde é apresentado o resultado e em baixo é apresentado o aspeto gráfico das estruturas de dados utilizadas.

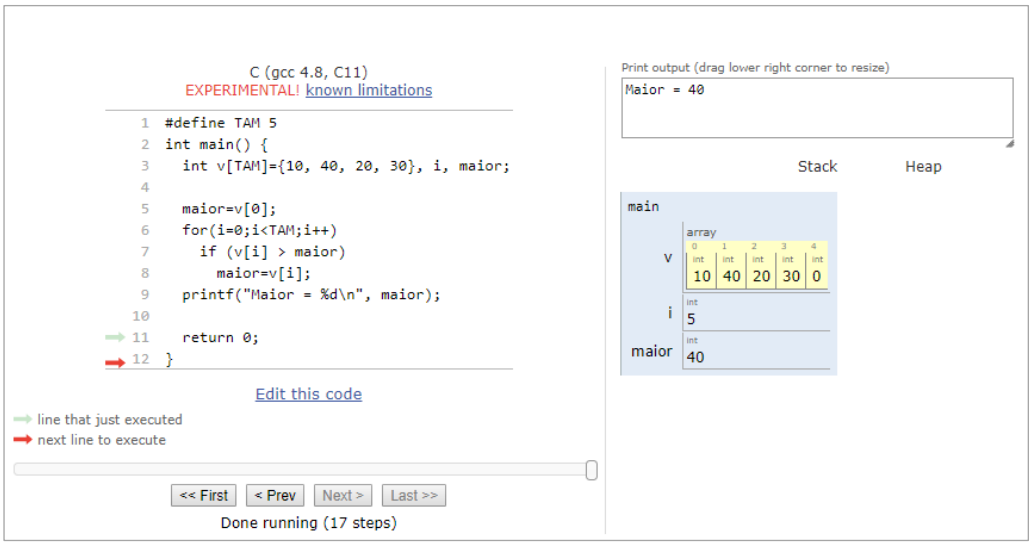


Figura 3-6: Execução de um programa em C no *Phyton Tutor*

No ano letivo de 2018/2019, integrado no projeto referido na secção anterior, o *Phython Tutor* foi introduzido pela primeira vez nas aulas das UCs de Programação I e Programação II aos cursos de licenciatura em Engenharia Informática e Informática de Gestão da ESTIG. No final do semestre foi solicitado aos estudantes que respondessem a um questionário online com sete questões relativas à utilização desta nova ferramenta. Na verdade, o questionário era mais abrangente e tinha 31 questões divididas por três temas: *Basic Information*, *Introductory Programming Course* e *C Tutor/Phyton Tutor*.

O questionário, que denominamos de *Programming Check-out* foi respondido por 103 estudantes daquelas licenciaturas. A **Figura 3-7** apresenta a utilização do *Phyton Tutor* pelos estudantes de Programação I e Programação II inquiridos. Podemos verificar que cerca de 76% dos estudantes usaram esta ferramenta, dentro e/ou fora da sala de aula. No entanto, quase metade dos estudantes (44.7%) apenas o fizessem dentro da sala de aula, aspeto que teremos de melhorar.

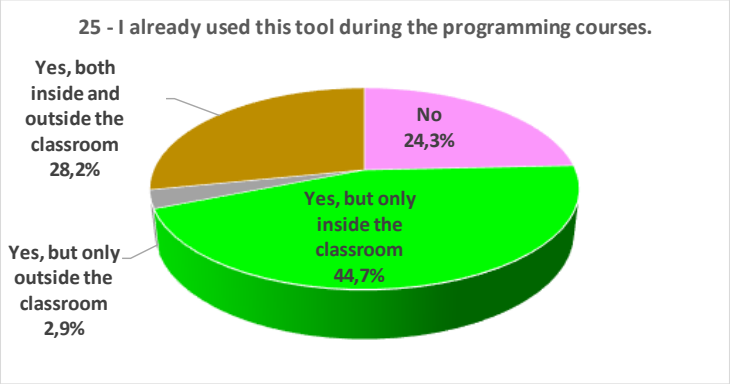


Figura 3-7: Utilização do *Python Tutor* pelos alunos de Programação I e Programação II

A **Figura 3-8** mostra o resultado das respostas dos estudantes inquiridos às restantes seis questões relativas à utilização do *Python Tutor*. Somando as respostas 'Concordo' (*Agree*) e 'Concordo plenamente' (*Strongly agree*) das questões 26 e 28 a 31 obtemos resultados entre 53.4% (questão 29) e 69.9% (questão 26). Estes valores revelam uma experiência positiva da utilização da ferramenta em vários aspetos importantes do ensino-aprendizagem da programação.

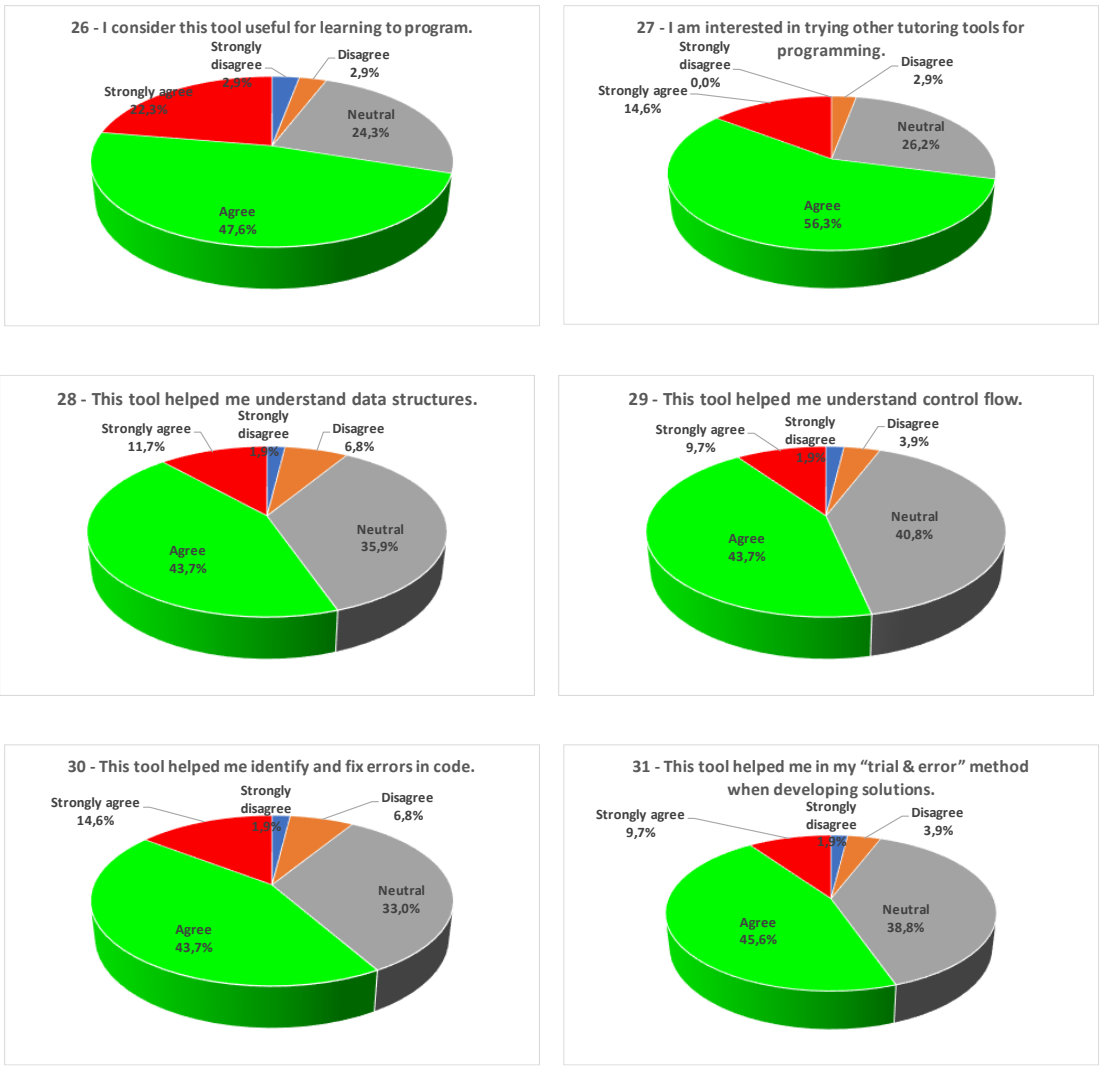


Figura 3-8: Gráficos com as respostas dos estudantes sobre a utilização do *Phyton Tutor*

As respostas da questão 27 revelam que os estudantes têm interesse em experimentar novas ferramentas que potenciem a aprendizagem da programação. De facto, 70.9% dos estudantes inquiridos reponderam positivamente a esta questão.

Nos gráficos da **Figura 3-8** observamos também que a percentagem de estudantes que se mostrou neutro perante todas as questões é demasiado elevado. De facto, este valor varia

entre os 24.3% (questão 26) e os 40.8% (questão 29). As razões para este resultado são diversas, no entanto, certamente que terá a ver com o elevado número de alunos inscritos, vários docentes a lecionar as UCs e por ter sido a primeira vez a introduzir-se nas aulas esta ferramenta.

Atualmente, estão já em curso a utilização de outras ferramentas para o apoio do ensino-aprendizagem da programação. Estas ferramentas, de utilização também pela internet, são o *codeboard*[14] e o *URI Online Judge*[15]. Dado que ainda estamos numa fase embrionária da utilização destas ferramentas a sua descrição e exploração está fora do âmbito desta Lição.

3.4 Comparação da Avaliação dos Estudantes Antes e Depois da Introdução do *Python Tutor*

Nesta secção é apresentada uma comparação entre os resultados obtidos pelos estudantes antes e depois da introdução do *Python Tutor* na UC de Programação II. Dado que esta ferramenta foi introduzida pela primeira vez no ano letivo de 2018/2019, a comparação vai ser efetuada com os resultados da avaliação obtidos pelos estudantes do ano letivo imediatamente anterior, portanto 2017/2018. Assim, a **Tabela 3.4** mostra o número de estudantes aprovados, reprovados, avaliados e inscritos de Programação II das Licenciaturas de Engenharia Informática (LEI) e Informática de Gestão (LIG) no ano letivo de 2017/2018.

Tabela 3.4 - Resultados da Avaliação dos estudantes de Programação II em 2017/2018

Nº de Estudantes	**Curso**		**LEI + LIG**
	LEI	LIG	
Aprovados	44	7	51
Reprovados	44	18	62
Avaliados	88	25	113
Inscritos	181	62	243
Aprovados/Avaliados (%)	**50.0**	28.0	**45.1**
Aprovados/Inscritos (%)	24.3	11.3	21.0
Avaliados/Inscritos (%)	48.6	40.3	46.5

[14] https://codeboard.io/
[15] https://www.urionlinejudge.com.br/

As três últimas linhas da **Tabela 3.4** mostram as taxas de estudantes aprovados versus avaliados, aprovados versus inscritos e avaliados versus inscritos. Obviamente que estes valores foram obtidos com base nos valores inseridos nas quatro primeiras linhas da mesma tabela.

Tabela 3.5 - Resultados da Avaliação dos estudantes de Programação II em 2018/2019

Nº de Estudantes	**Curso**		**LEI + LIG**
	LEI	LIG	
Aprovados	90	3	93
Reprovados	36	29	65
Avaliados	126	32	158
Inscritos	221	83	304
Aprovados/Avaliados (%)	**71.4**	9.4	**58.9**
Aprovados/Inscritos (%)	40.7	3.6	30.6
Avaliados/Inscritos (%)	57.0	38.6	52.0

A **Tabela 3.5** mostra o número de estudantes aprovados, reprovados, avaliados e inscritos de Programação II de LEI e de LIG no ano letivo de 2018/2019. Portanto, o ano letivo onde foi introduzido pela primeira vez o *Python Tutor*. Esta ferramenta foi utilizada pelos estudantes dentro e fora da sala de aula em diversos temas do conteúdo programático de Programação II. Também na **Tabela 3.5** as três últimas linhas mostram as taxas de estudantes aprovados versus avaliados, aprovados versus inscritos e avaliados versus inscritos.

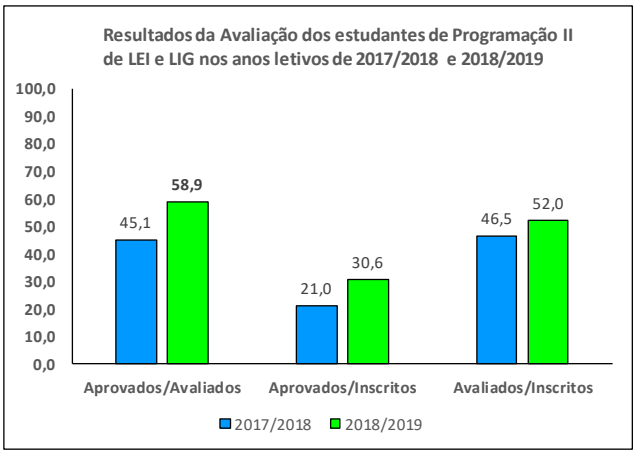


Figura 3-9: Gráfico com os Resultados da Avaliação de Programação II de LEI e LIG

Na **Figura 3-9** podemos observar a comparação dos resultados da avaliação dos estudantes de Programação II de LEI e LIG nos anos letivos de 2017/2018 (antes da introdução do

Python Tutor) e 2018/2019 (depois da introdução do *Python Tutor*). O gráfico mostra os indicadores que nos parecem mais importantes que são as taxas de estudantes aprovados versus avaliados, aprovados versus inscritos e avaliados versus inscritos. Em termos globais, considerando os estudantes de LEI e LIG, podemos concluir que todas estas taxas apresentam uma subida dos valores no ano letivo de 2018/2019 comparativamente ao verificado no ano letivo de 2017/2018. Concretamente, a taxa de estudantes aprovados versus avaliados, indicador que nos parece mais importante, aumentou quase 14% (13.8%), o que revela um indício positivo da utilização do *Phyton Tutor* pelos estudantes.

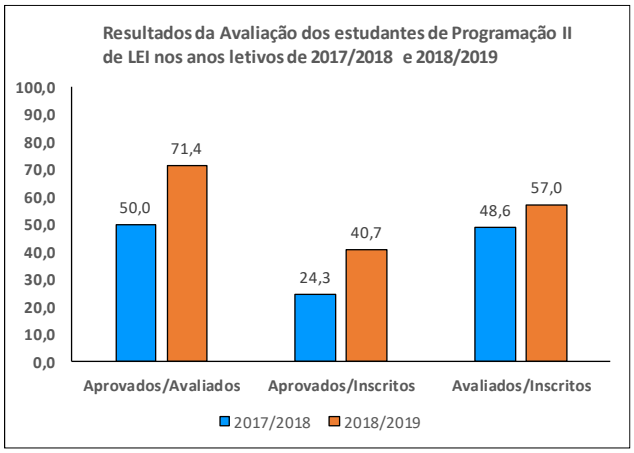


Figura 3-10: Gráfico com os Resultados da Avaliação de Programação II de LEI

O gráfico da **Figura 3-10** mostra a comparação dos resultados da avaliação dos estudantes de Programação II de LEI nos anos letivos de 2017/2018 e 2018/2019. Neste gráfico podemos observar que ocorreu um aumento de todas as taxas consideradas. Neste curso, podemos observar que o aumento de estudantes aprovados versus avaliados foi de 21.4%, o que nos parece um indicador bem positivo do efeito da utilização do *Python Tutor* dentro e fora da sala de aula.

O gráfico da **Figura 3-11** mostra a comparação dos resultados da avaliação dos estudantes de Programação II de LIG nos anos letivos de 2017/2018 e 2018/2019. Neste curso, a utilização do *Phyton Tutor* já não se revelou tão positiva, pois verificou-se uma diminuição dos

valores de todas as taxas consideradas. No caso da taxa de estudantes aprovados versus avaliados ocorreu uma diminuição de 18.6%.

Numa observação mais atenta, podemos verificar que apenas 7 estudantes de LIG obtiveram aprovação no ano letivo de 2017/2018 e apenas 3 no ano letivo de 2018/2019. Em nossa opinião, o motivo de tão baixas taxas de sucesso neste curso, tem a ver com a muito baixa preparação inicial dos estudantes ao nível da Informática em geral e da programação em particular. De facto, a grande maioria dos estudantes de LIG são oriundos de Países Africanos de Língua Oficial Portuguesa (PALOP) que revelam também algumas dificuldades ao nível do Português e da Matemática. A acrescentar ainda a estas dificuldades, é o facto de vários destes estudantes, por motivos burocráticos, chegarem ao longo do 1º semestre o que implica a perda de diversas aulas iniciais que se revelam importantes na aprendizagem de uma linguagem de programação.

Concluímos assim, que os resultados menos positivos obtidos pelos estudantes de LIG não tem a ver diretamente com a utilização do *Python Tutor*, mas sim com diversos fatores que não colocam os estudantes de LEI e LIG no mesmo ponto de partida. O desafio em motivar os estudantes de LIG a aprenderem uma linguagem de programação é enorme, no entanto, acreditamos que ferramentas que permitam visualizar as várias estruturas de dados, a execução instrução a instrução e a programação colaborativa poderão ser uma mais valia.

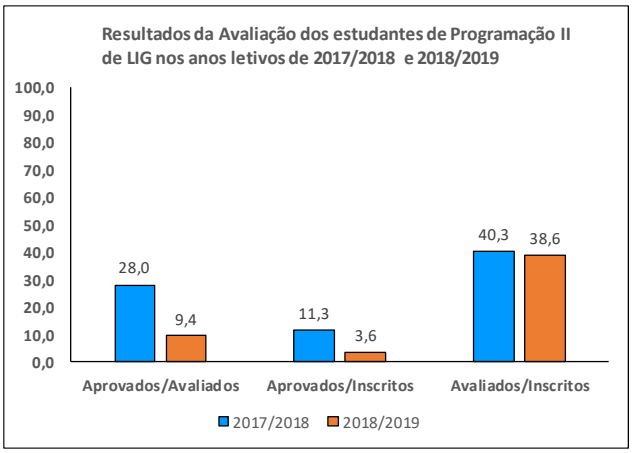


Figura 3-11: Gráfico com os Resultados da Avaliação de Programação II de LIG

Numa apreciação global, os resultados obtidos pelos estudantes de Programação II de LEI e LIG após a introdução do *Python Tutor* revelam que devemos incutir e aprofundar a utilização desta ferramenta. As características que ela apresenta podem certamente motivar os estudantes a aprenderem a árdua tarefa de programar e assim, permitir-lhes encarar depois, as restantes UCs ao longo do curso com mais entusiasmo e dedicação.

3.5 Ensino Baseado em Projetos

Esta secção apresenta uma experiência pedagógica de Ensino Baseado em Projetos (*Project Based Learning* (PBL)) aplicada a grupo de UCs lecionadas por docentes do Departamento de Sistemas de Informação da Universidade do Minho em que o autor desta Lição participou como aluno de doutoramento. Esta experiência encontra-se documentada no capítulo *Project-Based Learning: An Environment to Prepare IT Students for an Industry Career* do livro "*Overcoming Challenges in Software Engineering Education: Delivering Non-Technical Knowledge and Skills*".

O conjunto de UCs envolvidas nesta experiência pedagógica tratam do desenvolvimento de aplicações de informações no seu todo, aonde a programação é apenas uma das suas etapas, talvez a mais importante. Por isto e por o PBL ser uma metodologia de ensino que se tem afirmado nos últimos anos, parece-nos importante deixar uma breve descrição desta experiência.

O Processo de Bolonha sugere a mudança para um ensino baseado no trabalho dos estudantes e na aquisição efetiva de competências. Neste sentido, o *Project Based Learning* (Barrows & Tamblyn, 1980), aplicado inicialmente no ensino da Medicina, surge como uma metodologia adequada para este objetivo.

O reconhecimento da eficácia da abordagem do PBL no ensino da Engenharia deu origem a várias iniciativas[16], nomeadamente projetos científicos de natureza pedagógica, com o

[16] https://www.advance-he.ac.uk/knowledge-hub/engaging-engineering-and-design-students-through-project-based-learning-stem

intuito de produzir guias para a organização de atividades de ensino-aprendizagem segundo os princípios do PBL. Em alguns fóruns, a utilização do PBL no âmbito do ensino da Engenharia adotou a designação de *Project-Led Engineering Education* (PLEE) (Powell, Powell, & Weenk, 2003).

A abordagem PBL aplicada nesta experiência envolvia estudantes de licenciatura, mestrado e doutoramento de cursos na área da Informática da Universidade do Minho. O objetivo era desenvolver uma aplicação informática para um cliente real, que também tinha um papel participativo em todo o processo. A integração das quatro UCs envolvidas nesta abordagem é esquematizada na **Erro! A origem da referência não foi encontrada.**.

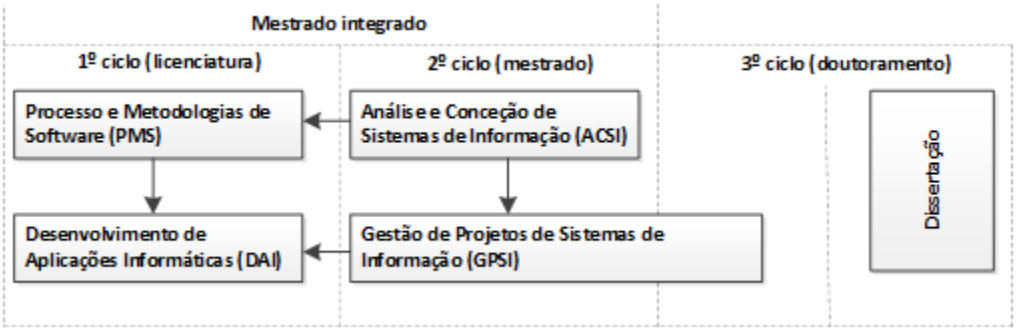


Figura 3-12: Integração entre as unidades curriculares da experiência PBL

Ao nível da licenciatura (1º ciclo) foram envolvidos estudantes das UCs Processo e Metodologias de Software (PMS) e Desenvolvimento de Aplicações Informáticas (DAI). Ao nível do mestrado (2º ciclo) foram envolvidos estudantes de Análise e Conceção de Sistemas de Informação (ACSI) e Gestão de Projetos de Sistemas de Informação (GPSI).

Nesta abordagem PBL, alguns dos estudantes de doutoramento do grupo de investigação SEMAG (*Software Engineering and Management Group*) (SEMAG, 2001) envolviam-se com os estudantes das quatro unidades curriculares, como forma de procederem a estudos científicos sustentados em experiências em contextos educacionais e que têm dado origem a publicações internacionais, ver por exemplo (L. M. Alves, Sousa, Ribeiro, & Machado, 2013). Desta forma, este grupo de quatro unidades curriculares constitui um espaço simultâneo de inovação

pedagógica e científica da qual todos os intervenientes beneficiam (Siqueira, Barbaran, & Becerra, 2008).

O projeto que todos os anos letivos é apresentado aos estudantes de PMS e DAI exige que se tornem aprendentes energéticos. Ninguém lhes dará toda a informação, nem as respostas serão consecutivamente encontradas nos livros. Resolver este problema exige que os estudantes «descubram as queixas», «investiguem as culpas» e desenvolvam o melhor modo para resolver a situação. Assim, este projeto apresenta algumas características consideradas determinantes para a criação de um contexto de enorme catalisação dos fenómenos de ensino-aprendizagem (Luís M. Alves, Ribeiro, & Machado, 2014):

(1) Cliente Real;

(2) Anteprojeto versus projeto;

(3) Equipas grandes;

(4) Rigor processual;

(5) Complexidade da solução;

(6) Outras.

Esta abordagem começou a ser efetivamente implementada no ano letivo de 2009/2010, embora alguns ensaios bem controlados tenham sido testados em anos anteriores. A colaboração do cliente real no desenvolvimento da aplicação informática começou naquele ano académico. Por uma questão logística, este cliente ficava localizado numa zona próxima da Universidade do Minho. A área de atividade do cliente era bastante diversificada. Ao longo dos vários anos letivos tivemos colaborações de uma empresa de construção de piscinas, uma Instituição Particular de Solidariedade Social (IPSS), uma equipa profissional de andebol e uma equipa profissional de futebol.

A título de exemplo, na Erro! A origem da referência não foi encontrada. são compilados os principais dados numéricos relativos aos resultados obtidos pelos estudantes inscritos

nas quatro unidades curriculares no ano letivo de 2011/12. Os dados são reveladores da adequação e sucesso desta metodologia de ensino no contexto do ensino de competências *core* na área da informática.

– Resultados obtidos pelos estudantes de PMS, DAI, ACSI e GPSI em 2011/12

Ano letivo 2011/12	PMS	DAI	ACSI	GPSI
inscritos	143	142	31	39
avaliados	135	103	29	36
aprovados	119	101	29	36
aprovados/avaliados	88.1%	98.1%	100.0%	100.0%
valor médio das classificações finais	14.1	14.0	14.4	15.4

Atualmente, esta abordagem PBL é ainda implementada, embora os processos de desenvolvimento de software tenham sido alterados ao longo dos anos, mas a dinâmica entre as quatro UCs tem-se mantido.

Capítulo 4
Conclusões

Após o enquadramento da Lição, este capítulo apresenta um breve sumário dos resultados alcançados e das principais elações a reter sobre o ensino-aprendizagem de programação de computadores.

Do ponto de vista estritamente formal, esta Lição foi elaborada como satisfação parcial dos requisitos para a prestação de Provas Públicas de Avaliação da Competência Pedagógica e Técnico-Científica. No entanto, perante a inexistência de referenciais ou recomendações para esclarecer a natureza e profundidade deste tipo de Lição, o autor achou por bem não se limitar à descrição do programa, do conteúdo e dos métodos de ensino do grupo de unidades curriculares em análise. Ao invés de selecionar uma parte do conteúdo programático de uma das unidades curriculares e apresentá-lo na Lição, o autor, correndo os riscos inerentes neste tipo de decisões, resolveu apresentar algum trabalho científico envolvendo a temática do ensino-aprendizagem da programação.

A presente Lição enquadra-se na área da programação, mais concretamente, na área do ensino-aprendizagem da programação, tendo sido escolhidas as unidades curriculares de Programação I e Programação II como objeto de estudo.

A experiência pedagógica adquirida pelo autor desta Lição ao longo de mais de duas décadas de ensino permite-lhe afirmar que é uma tarefa extremamente difícil a de ensinar programação, sobretudo, a estudantes que nunca tiveram qualquer contacto com esta temática. Para que haja sucesso nesta árdua tarefa é imprescindível um papel ativo do estudante desde o princípio do processo ensino-aprendizagem. O ponto de partida mais ajustado é o de

potenciar a motivação inicial do estudante, sem a qual, não haverá certamente, um desfecho positivo.

A elevada taxa de insucesso em unidades curriculares de programação é uma preocupação da comunidade internacional, de que o IPB, sobretudo a ESTiG, não é alheia. De facto, diversos têm sido os artigos científicos publicados que reportam preocupações relativas ao ensino-aprendizagem da programação.

O esforço de melhoria no ensino-aprendizagem de competências de programação pode ser conseguido por duas vias distintas que podem complementar-se muito bem. Uma das vias, são as ferramentas de apoio ao processo de ensino-aprendizagem, algumas delas disponíveis online. A outra, são as metodologias de ensino a utilizar, é muito importante encontrar a metodologia mais adequada para o sucesso deste processo.

Esta Lição apresenta um projeto de investigação bilateral entre instituições de ensino superior de Portugal e da Sérvia, onde a programação é o tema central. Inserido no plano de trabalho deste projeto foram produzidos até ao momento dois artigos científicos, um dos quais, reportando um estudo prévio da situação atual das UCs de programação, e o outro, reportando o conhecimento inicial, as expectativas e a motivação dos estudantes inquiridos de UCs de programação introdutória de ambos os países.

A principal conclusão obtida pelos estudos realizados neste projeto é a de que há mais semelhanças do que diferenças nos dois países em relação à temática da programação. De facto, a distinção mais relevante entre as UCs dos dois países é a diferença nas taxas de aprovação e nas taxas de avaliação, com substancial vantagem para a Sérvia, onde estas taxas são mais elevadas. Os estudantes inquiridos de ambos países reportaram baixo conhecimento inicial sobre programação, mas elevadas expetativas em relação à UC de programação introdutória e elevada motivação em relação à programação em geral.

Ainda dentro do contexto deste projeto bilateral foi introduzido em Programação II uma ferramenta online de apoio ao ensino-aprendizagem de programação, o *Phython Tutor*. Embora seja necessária mais investigação, a primeira experiência da utilização desta ferramenta dentro e fora de aula revelou-se positiva. De facto, os estudantes inquiridos responderem positivamente com valores entre os 53,4% e os 69.9% sobre a utilidade da ferramenta para aprenderem a programar, a compreender as estruturas de dados, a compreender as instruções de controlo de fluxo, a identificar e corrigir os erros e a praticar o método de tentativa e erro no desenvolvimento de soluções.

Por se se tratar de uma metodologia atual e que tem sido aplicada em diversas UCs a nível nacional e internacional em várias instituições de ensino superior é apresentada uma experiência pedagógica aplicando o PBL implementada na Universidade do Minho. O autor desta Lição teve participação ativa nesta experiência tendo o papel de investigador no ambiente académico recriado. De facto, o objetivo principal da combinação do conjunto de UCs era o de simular um ambiente empresarial em contexto académico.

Tendo iniciado no ano letivo de 2009/2010 a experiência pedagógica tem-se repetido ao longo dos anos, sendo que, em cada iteração são acrescentados ou testados mecanismos ou características novas. Por exemplo, a formalização da entrega do produto, subcontratação, interoperabilidade tecnológica entre soluções parciais, consultoria de serviços externos, utilização de padrões, concorrência entre outras.

A obtenção de um ambiente empresarial simulado em contexto educacional é uma tarefa que exige grande esforço do corpo docente e dos estudantes envolvidos. A sincronização entre os docentes tem de ser uma preocupação constante, caso contrário, não é possível alcançar os objetivos desejados. A taxa de sucesso dos estudantes é muito elevada, o que motiva, simultaneamente, professores e estudantes.

Como trabalho futuro, uma experiência muito interessante seria sincronizar UCs de programação com UCs de matemática aplicando o PBL na procura de soluções para problemas do mundo real. Desta forma, os estudantes aprenderiam programação ao mesmo tempo que iriam assimilar os conceitos da matemática.

Bibliografia

Aleksic, V., & Ivanovic, M. (2016). Introductory Programming Subject in European Higher Education. *Informatics in Education, 15*(2), 163-182.

Alves, L. M., Gajić, D., Henriques, P. R., Ivančević, V., Lalić, M., Luković, I., . . . Tavares, P. C. (2019, 16-20 September). *Student Entrance Knowledge, Expectations, and Motivation within Introductory Programming Courses in Portugal and Serbia.* Paper presented at the 47th European Society for Engineering Education (SEFI 2019), Budapest, Hungary.

Alves, L. M., Henriques, P. R., Ivančević, V., Lalić, M., Luković, I., Pereira, M. J. V., & Tavares, P. C. (2018, 2018-10-05). *A Comparison of Introductory Programming Courses between Portugal and Serbia.* Paper presented at the International conference on Applied Internet and Information Technologies (AIIT 2018), Bitola, Republic of Macedonia.

Alves, L. M., Ribeiro, P., & Machado, R. J. (2014). Project-Based Learning: An Environment to Prepare IT Students for an Industry Career. In *Overcoming Challenges in Software Engineering Education: Delivering Non-Technical Knowledge and Skills* (pp. 230-249). Hershey, PA, USA: IGI Global.

Alves, L. M., Sousa, A., Ribeiro, P., & Machado, R. J. (2013, 23-26 Oct.). *An empirical study on the estimation of software development effort with use case points.* Paper presented at the 2013 Frontiers in Education Conference, Oklahoma City, Oklahoma, USA.

Arends, R. (2008). *Aprender a Ensinar*: Mc Graw-Hill.

Barrows, H. S., & Tamblyn, R. H. (1980). *Problem-Based Learning: An Approach to Medical Education*. Broadway, New York, USA: Springer.

Bennedsen, J., & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. *ACM Inroads, 10*(2), 30-36. doi:10.1145/3324888

Bergin, S., & Reilly, R. (2005). *The influence of motivation and comfort-level on learning to program.* Paper presented at the Proceedings of 17th Workshop of the Psychology of Programming Interest Group, Brighton, UK.

Gomes, A., & Mendes, A. (2007). *Learning to program - Difficulties and Solutions*. Paper presented at the International Conference on Engineering Education (ICEE 2007), Coimbra, Portugal.

Gomes, A. J., & Mendes, A. (2010). *A study on student performance in first year CS courses*. Paper presented at the Proceedings of the fifteenth annual conference on Innovation and technology in computer science education, Bilkent, Ankara, Turkey.

Gomes, A. J., Santos, A. N., & Mendes, A. (2012). *A study on students' behaviours and attitudes towards learning to program*. Paper presented at the 17th ACM annual conference on Innovation and technology in computer science education, Haifa, Israel.

James, R. (2002). Student's changing expectations of higher education and the consequences of mismatches with the reality. In *Responding to Student Expectations* (pp. 71-83). Paris, France: OECD.

Kandiko, C. B., & Mawer, M. (2013). *Student Expectations and Perceptions of Higher Education*. Retrieved from King's Learning Institute: https://www.kcl.ac.uk/study/learningteaching/kli/People/Research/DL/QAAReport.pdf

Kori, K., Pedaste, M., Leijen, A., & Tõnisson, E. (2016). The role of programming experience in ICT students' learning motivation and academic achievement. *International Journal of Information and Education Technology, 6*(5), 331-337. doi:10.7763/IJIET.2016.V6.709

Li, S., Records, H., & Fougere, K. (2004). An exploratory investigation of gender difference in student selection of a CIS minor. *Issues in Information Systems, 5*(2), 598-604.

Machado, E. A. (2013). *Avaliar é Ser Sujeito ou Sujeitar-se? Elementos para uma Genealogia da Avaliação*: Edições Pedago.

Mendes, A. J., Paquete, L., Cardoso, A., & Gomes, A. (2012, 3-6 Oct.). *Increasing Student Commitment in Introductory Programming Learning.* Paper presented at the Frontiers in Education Conference (FIE 2012), Seattle, WA, USA.

Newble, D., & Cannon, R. (2000). *Handbook for Teachers in Universities and Colleges: A Guide to Improving Teaching Methods* (4th Edition ed.). Abingdon, Oxfordshire, UK: Routledge Falmer.

Parnas, D. L. (1990). Education for computing professionals. *Computer, 23*(1), 17-22. doi:10.1109/2.48796

Passey, D. (2017). Computer science (CS) in the compulsory education curriculum: Implications for future research. *Education and Information Technologies, 22*(2), 421-443. doi:10.1007/s10639-016-9475-z

Powell, W., Powell, P. C., & Weenk, W. (2003). *Project-Led Engineering Education*. Netherland: Lemma Publishers.

Pratt, D. (1994). *Curriculum Planning : A Handbook for Professionals*. Fort Worth, Texas, USA: Harcourt Brace college.

Robins, A. V., Rountree, J., & Rountree, N. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education, 13*, 137-172. doi:10.1076/csed.13.2.137.14200

SEMAG. (2001). Software Engineering and Management Group. Retrieved from https://sites.google.com/a/dsi.uminho.pt/semag/

Siqueira, F. L., Barbaran, G. M. C., & Becerra, J. L. R. (2008, April). *A Software Factory for Education in Software Engineering.* Paper presented at the Proceedings of the 21st Conference on Software Engineering Education & Training (CSEE&T), Charleston, South Carolina, USA.