

Help To Programming: uma Ferramenta para o Ensino e Aprendizagem da Programação

José Figueiredo¹[0000-0002-8501-1686] and Francisco J. García-Peñalvo²[0000-0001-9987-5584]

¹ Research Unit for Inland Development, Polytechnic of Guarda, Portugal, jfig@ipg.pt

² Computer Science Department, Research Institute for Educational Sciences, GRIAL research group, University of Salamanca, fgarcia@usal.es

Resumo. Existe a ideia generalizada de que o ensino e aprendizagem da programação é difícil. Desde que surgiram as linguagens de programação que este tema é estudado e investigado por todos os que se dedicam a esta área das ciências da computação. Os conceitos básicos da programação fazem parte de muitos cursos de ensino superior nas mais diversas áreas do conhecimento. As dificuldades no ensino e aprendizagem da programação refletem-se não só nas altas taxas de reprovação, mas também, e talvez a mais preocupante, nas altas percentagens de abandono, na falta de motivação e de interesse dos alunos. Neste trabalho apresentamos algumas das razões nas dificuldades do ensino e na aprendizagem da programação. Descrevemos um conjunto de estratégias de ensino e aprendizagem de introdução à programação de modo a reduzir este problema. Este conjunto de estratégias é auxiliado por uma aplicação *HTProgramming* que nos permitirá acompanhar em pormenor o desenvolvimento de cada aluno, nas diferentes fases do processo de aprendizagem. À medida que o aluno constrói o seu perfil de aprendizagem será possível aplicar um modelo preditivo de sucesso ou insucesso. É possível ao aluno melhorar aspetos específicos do seu perfil de aprendizagem e ao professor ter um conhecimento preciso do nível de conhecimento de cada aluno, e intervir rapidamente se necessário. Os resultados obtidos são encorajadores. Os alunos estão mais interessados, motivados e envolvidos no processo de ensino e aprendizagem e sentem-se mais confiantes com a possibilidade de aprender e praticar ao seu próprio ritmo, sem o receio de errar.

Keywords: programming, CS1, teaching programming, learning programming, intelligent tutoring systems, ITS.

1 Contexto e Motivação

Os alunos de hoje devem ser capazes de se adaptar a um ambiente dinâmico rodeado de novas tecnologias. O conhecimento básico de informática não é suficiente para permanecer competitivo no exigente mercado de trabalho de hoje e ter os conhecimentos necessários para os anos futuros. É essencial para os alunos desenvolver conhecimentos mais profundos na área das ciências da computação. Desenvolver com-

petências transversais como a criatividade, a resolução de problemas, a persistência, a colaboração, a comunicação e o pensamento crítico são aptidões cruciais para o sucesso de um aluno no futuro, em face da constante evolução da tecnologia, independentemente de sua área de estudo. Os cursos de programação são geralmente considerados eficazes para promover e desenvolver essas habilidades ou conhecimento. No entanto, os cursos de programação têm uma conotação negativa, existe uma opinião generalizada entre os alunos de que a programação é difícil [11].

Idealmente um trabalho de investigação, como é o caso de um doutoramento, deve estar inserido numa área que fascine o candidato, para conseguir suportar longos anos de trabalho árduo. Os trinta anos dedicados ao ensino, principalmente ao ensino da introdução à programação, têm sido motivantes e de constante desafio. Todos os anos cerca de uma centena de alunos fazem parte da pauta da unidade curricular de introdução à programação, do primeiro ano do curso de engenharia informática. Todos os anos são efetuadas pequenas adaptações e experiências de forma a adaptar e ensinar as competências exigidas, às características e conhecimentos dos alunos. Porém, quase invariavelmente, o resultado é o mesmo. Altas taxas de reprovação e abandono. Esta situação é especialmente preocupante em alunos do curso de engenharia informática, onde a programação é uma das principais áreas. Os alunos com dificuldades na área da programação perdem o interesse pelo curso, o que pode levar ao abandono.

A nossa principal motivação para o desenvolvimento deste trabalho é compreender quais as dificuldades dos alunos, quais os fatores que mais influenciam o seu processo de aprendizagem da programação, quais as ferramentas e / ou métodos ou tecnologias que podem ser utilizados para reduzir os problemas de ensino e aprendizagem do curso inicial de programação. Para a análise destes fatores é necessário um acompanhamento permanente e efetivo de cada aluno, para tal é proposto o desenvolvimento de um protótipo de um ambiente de aprendizagem de programação para ajudar os alunos a superar suas dificuldades e, para o professor conhecer a cada momento o nível de conhecimento de cada aluno.

O artigo está estruturado da seguinte forma. A próxima seção descreve o estado da arte. A seção 3 apresenta o problema, objetivos e questões de investigação. A seção 4 aborda a metodologia de investigação. Na seção 5 procuramos descrever o trabalho realizado e apresentamos a aplicação desenvolvida para auxílio do nosso propósito. A seção 6 apresenta e discute os resultados até o momento. A seção 7 descreve o estado da dissertação e as contribuições esperadas.

2 Estado da arte

As dificuldades no ensino e na aprendizagem da programação são um problema universal. Desde que surgiram as primeiras linguagens de programação que este tema tem sido alvo de estudos e de reflexões. São vários os trabalhos onde o tema é abordado[8, 11–14]. No relatório desenvolvido pelo grupo de trabalho da Conference in Innovation and Technology in Computer Science Education (ITiCSE) em [13], por exemplo, o objetivo principal foi o de obter uma visão geral sobre os principais temas abordados na área da introdução à programação. Como resultado verificamos que um

grande número de trabalhos aborda temas como: identificar as dificuldades dos alunos, desenhar ferramentas para auxiliar os alunos, incentivar os alunos a alterar o seu comportamento tendo em vista o sucesso, previsão do sucesso e a deteção de comportamentos indesejáveis, entre outros.

Porque é que programar é difícil? Segundo Derek Sleeman, no seu trabalho em [15], refere que o principal papel do programador é, em primeiro lugar, construir um plano, uma estratégia que o levará à resolução de um problema. Em segundo lugar, é necessário conhecer a sintaxe e semântica de uma linguagem de programação. De seguida, é necessário que a estratégia de resolução esteja de acordo com os recursos disponíveis na linguagem de programação. Esta coordenação entre tarefas aumenta significativamente a complexidade da atividade de programação.

Relativamente às altas taxas de reprovação nos cursos de introdução à programação o trabalho realizado por Jens Bennedsen e Michael Caspersen, autores do trabalho apresentado em [2], e repetido 12 anos depois no trabalho apresentado em [1], os seus autores concluem que essas taxas não são preocupantes. Ou seja, taxas de reprovação na ordem dos 33% em 2007, e de 28% em 2017, não podem ser consideradas elevadas.

3 Problema, Objetivos e Questões de Investigação

Depois de uma breve descrição, contextualização e motivação sobre o tema das dificuldades de aprendizagem inicial da programação, identificamos o nosso problema como sendo as altas taxas de reprovação, ou os alunos não adquirem as competências básicas na área, a desmotivação e abandono dos alunos na unidade curricular de introdução à programação, do curso de engenharia informática. Este problema revela-se especialmente preocupante por ser uma unidade curricular basilar no curso de engenharia informática. Por outro lado, é difícil para os professores terem uma correta perceção sobre os conhecimentos e dificuldades de cada aluno, e intervir rapidamente com a ajuda individualizada e necessária de cada aluno.

Identificado o nosso problema, definimos como o objetivo principal do nosso trabalho o de construir um modelo de ensino e aprendizagem inicial da programação, que permita auxiliar alunos e professores na concretização dos seus objetivos. Para atingir o nosso objetivo, achamos necessário compreender quais as reais dificuldades dos nossos alunos, quais os fatores que mais influenciam o seu processo de aprendizagem, quais as ferramentas e / ou métodos ou tecnologias que podem ser utilizados para reduzir os problemas no ensino e aprendizagem inicial de programação. Para o acompanhamento eficaz de todo o processo de ensino e aprendizagem inicial da programação, vamos recorrer a meios tecnológicos que permitam auxiliar alunos e professores. Como tal, desenvolvemos um protótipo de uma aplicação para ajudar no ensino e aprendizagem inicial da programação, designado por HTProgramming (Help to Programming), o qual será descrito nas seções seguintes.

3.1 Questões de investigação

Identificado o problema e definido o objetivo, formulamos algumas questões de investigação às quais nosso trabalho de investigação pretende responder.

- Que tipo de atividades do pensamento computacional contribuem de modo eficaz para o sucesso?
- Quais são os fatores que mais influenciam o processo de aprendizagem inicial da programação?
- Quais são as atividades do curriculum da aprendizagem inicial da programação mais contribuem para o sucesso?
- O perfil do aluno pode ser visto como um valor que traduz o conhecimento e competências adquiridas pelo aluno?
- O modelo preditivo do insucesso do aluno é eficiente?

4 Metodologia

Com este trabalho pretendemos determinar quais são os métodos, técnicas, atitudes e comportamentos que alunos e professores, em particular no domínio inicial da programação ao nível universitário, com características semelhantes às do nosso modelo de estudo, melhoram o processo educacional. E deste modo, construir um modelo de ensino e aprendizagem inicial da programação que satisfaça os nossos objetivos.

A necessidade de investigar em educação [10] surge quando queremos conhecer melhor o funcionamento de uma determinada situação educacional e, pretendemos responder a múltiplas questões que colocamos sobre como melhorar nossa forma de agir. Perante uma situação que nos interessa conhecer com certo rigor e profundidade, será necessário analisar suas características por meio de pesquisas, identificar os fatores que a influenciam e que a condicionam, compreender as interações e fatores que ocorrem e, se possível, estabelecer as causas que determinam o aparecimento de determinados comportamentos ou resultados para que posteriormente, se for o caso, a tomada de decisão sobre o tipo de intervenção que pode ser realizada sobre essa situação [10].

4.1 Perspetiva e métodos de investigação

Após uma ideia clara do problema de investigação e uma boa base da revisão da literatura, o próximo passo é fazer uma escolha preliminar da metodologia. Tendo em consideração alguns conceitos sobre a perspetiva de investigação quantitativas [16], como: segue uma epistemologia positivista a qual defende que há uma realidade objetiva que pode ser expressa numericamente, uma amostra representativa permite a generalização dos resultados e, ainda, a perspetiva quantitativa enfatiza estudos que são experimentais por natureza, confere importância a medidas, e procura relações.

No que respeita à metodologia de investigação, o tipo de investigação-ação reconhecida como uma metodologia que procura a melhoria das práticas mediante a mudança e a aprendizagem a partir das consequências dessas mudanças. Permite ainda a

participação de todos os implicados. Desenvolve-se numa espiral de ciclos de planificação, ação, observação e reflexão. É, portanto, um processo sistemático de aprendizagem orientado para a ação com objetivo de atingir um certo fim, exigindo que esta seja submetida à prova, permitindo dar uma justificação a partir do trabalho, mediante uma argumentação desenvolvida, comprovada e cientificamente examinada.

No entanto, esta metodologia de investigação enquadra-se numa perspetiva de investigação qualitativa. Segundo, Yin em , por vezes metodologias que usam medidas qualitativas, tendem a utilizar uma perspetiva quantitativa, preocupada com o explorar, descreve e explicar um problema.

A perspetiva e metodologia de investigação carece de melhor atenção e enquadramento no nosso trabalho, no entanto uma perspetiva mista e a metodologia de investigação-ação parece enquadrar-se no nosso trabalho.

4.2 Contexto e grupo de estudo

Este trabalho envolve os alunos da unidade curricular de introdução à programação, do 1º ano, 1º semestre, do curso de Engenharia Informática, do Instituto politécnico da Guarda. O Instituto Politécnico da Guarda (IPG) é uma instituição de ensino superior situada no interior de Portugal. A unidade curricular de introdução à programação tem como plano curricular a introdução à programação em linguagem C. Normalmente, a unidade curricular de introdução à programação tem cerca de uma centena de alunos inscritos.

Nosso grupo de estudo tem características muito especiais que podem afetar, em nossa opinião, o processo de aprendizagem inicial da programação:

- O curso de engenharia de informática, IPG, geralmente não é a primeira escolha dos alunos, o que em algumas circunstâncias pode afetar o empenho e a motivação dos alunos.
- A nota média, nos últimos anos, está entre 10 e 12 valores.
- Os alunos revelam algumas dificuldades gerais na área da Ciência, Tecnologia, Engenharia e Matemática.
- Muitos de nossos alunos nunca tiveram cursos de programação, nem a oportunidade de praticar atividades do pensamento computacional.
- Nos últimos 4 anos, temos recebido alunos oriundos dos Países Africanos de Língua Oficial Portuguesa (PALOP), que na sua maioria revelam problemas na sua formação geral.

4.3 Métodos e instrumentos de recolha de dados

Depois de identificado o tipo de investigação a realizar, formulado o problema e definidas as questões pertinentes da investigação, torna-se necessário elaborar o plano de pesquisa que oriente o investigador nos processos de recolha, análise e interpretação dos dados. O plano de pesquisa tem como objetivo dar uma resposta precisa às questões colocadas. O investigador deve assegurar-se que os métodos e técnicas de recolha de informação são utilizados de forma a obter informação suficiente e pertinente [16].

A recolha de dados é efetuada através da aplicação, *HTProgramming*, construída para o efeito. Os dados são recolhidos constantemente ao longo da duração da unidade curricular de introdução à programação. Os dados são recolhidos em contexto de sala de aula ou em trabalho individual realizado pelo aluno. O conjunto de dados são utilizados no treino de uma rede neuronal [4], que será utilizada para prever o sucesso ou insucesso de cada aluno. Ao mesmo tempo, o conjunto de resultados obtidos por cada aluno será utilizado para construir o seu perfil [6].

Nas secções seguintes descrevemos o trabalho realizado na tentativa de melhor perceber e resolver o problema do ensino e aprendizagem da programação.

5 Trabalho realizado

Este capítulo tem como objetivo dar uma visão geral dos trabalhos e estudos realizados em apoio ao nosso objetivo principal. Ou seja, encontrar uma proposta de um modelo de ensino e aprendizagem inicial da programação que melhor significativamente os resultados no nosso grupo de estudo, mas também que possa ser utilizado de forma generalizada no ensino inicial da programação em cursos universitários.

5.1 Atividades NE e do pensamento computacional

Nas três décadas dedicadas ao ensino da aprendizagem inicial da programação, que nos preocupamos com as competências adquiridas, com o abandono e com a falta de motivação dos alunos. Através da pesquisa bibliográfica é facilmente comprovável que é um problema universal e de preocupação por todos os que se dedicam ao ensino desta área de estudos.

Iniciamos o nosso trabalho com a caracterização dos nossos alunos. A caracterização do aluno de programação é efetuada através de um questionário ao aluno, onde para além, da sua caracterização pessoal é questionado ao aluno se já tem conhecimento sobre programação e como avalia os seus conhecimentos. Em face dos resultados, e na nossa experiência, começamos por elaborar um conjunto de atividades relacionadas com o pensamento computacional, ou diretamente relacionadas com as capacidades cognitivas para a programação.

Atividades NE. Atividades no-electronic, ou sem o uso de computadores. A maioria dos alunos que ingressam em cursos de engenharia, onde a programação é uma competência crucial, nunca teve a oportunidade de desenvolver habilidades de pensamento computacional. Deste sentido, desenvolvemos um conjunto de atividades com esse objetivo.

Nas atividades de seguir e dar instruções, os alunos desenham numa folha de papel o que outro aluno ou o professor descreve. A utilização deste tipo de exercícios tem como objetivo aumentar o desenvolvimento do raciocínio cognitivo e visualização espacial dos alunos, fortemente associado às características necessárias à programação [3, 7].

O desenho de mapas consiste no desenho em papel do caminho necessário para nos deslocarmos entre um local A e um local B. Com a utilização deste tipo de exercícios,

pretendemos desenvolver as capacidades dos alunos em planejar, na conceção e na descrição de uma situação concreta. Estudos têm demonstrado a relação entre o estilo e o nível de detalhe na descrição e construção de um mapa com os objetivos de um curso de programação.

Origami ou dobrar papel é uma arte secular japonesa difundida em todo o mundo, conhecida pelo desenvolvimento de características, tais como: percepção visual e espacial, coordenação motora fina, memória, alívio de stresse e tensão, paciência e persistência; autoconfiança, pensamento lógico e atenção e concentração.

Como resultado deste estudo surgiram as seguintes publicações [3, 7].

5.2 Construção do perfil do aluno

Com o objetivo de responder às questões de investigação, como: “Quais são os fatores que mais influenciam o processo de aprendizagem inicial da programação?”, e “Quais são as atividades do curriculum da aprendizagem inicial da programação mais contribuem para o sucesso?”, surgiu o trabalho apresentado em [6]. A construção do perfil de competências de cada aluno em programação é baseada no conceito de videojogos atuais, como FIFA ou Assassin's Creed, por exemplo. Onde, os personagens são incitados a construir e melhorar suas características e habilidades em áreas específicas para completar suas tarefas ou mudar de nível. Por exemplo, um jogador do FIFA pode treinar a marcação de penáltis, dribles, cobranças de falta, de cantos e outras ações para melhorar suas habilidades durante o jogo. Da mesma forma, queremos que cada aluno seja capaz de melhorar e aprofundar suas habilidades e competências na programação, realizando um conjunto de exercícios adequados e trabalhados para cada aluno e situação. É nossa intenção poder identificar o conjunto mínimo de competências necessárias para o sucesso do aluno no curso.

5.3 Prever o insucesso do aluno

A identificação precoce de potenciais problemas e resposta imediata é fundamental para evitar o insucesso do aluno e reduzir as taxas de abandono. Como tal, propomos um modelo preditivo, baseado numa *machine learning* (rede neuronal), com base no perfil do aluno [4]. O modelo resultante permite que os professores identifiquem precocemente os alunos com maior probabilidade de insucesso, permitindo-lhes dedicar mais tempo a esses alunos e tentar novas estratégias para melhorar suas competências em programação.

5.4 Gamificação

Os jogos têm um importante poder motivacional. Eles tiram partido de um conjunto de ferramentas para encorajar as pessoas a interagirem com eles apenas pelo prazer de jogar e pela possibilidade de ganhar. Usar elementos de jogos em contextos não relacionados a jogos, designado por gamificação, é uma das boas formas de motivar e estimular os alunos a aprender programação. Para avaliar como a gamificação pode influenciar a aprendizagem inicial da programação realizamos um estudo apresentado em [5]. Neste estudo foram usadas algumas técnicas de gamificação como *leader*

board, pontos, níveis, *badges*, feedback imediato, caixa de surpresa e torneio a eliminar, semelhante a uma liga dos campeões.

5.5 HTProgramming

É de primordial importância o desenvolvimento de uma ferramenta de auxílio à gestão da informação. Neste sentido, construímos uma ferramenta designada por *HTProgramming - Help To Programming*. Esta ferramenta visa auxiliar alunos e professores em todo o processo de ensino e aprendizagem inicial da programação. Na Fig.1 apresentamos o esquema geral da aplicação *HTProgramming*, constituído por dois módulos, o módulo de administração e o módulo do aluno, os quais têm acesso a uma base de dados remota. Descrevemos de seguida cada um dos módulos.

Módulo de Administração. Este módulo é utilizado pelo professor para a gestão e controlo do processo de aprendizagem. Na Fig.2 destacamos as principais funções deste módulo. Iniciamos a apresentação das funcionalidades deste módulo como a janela principal, identificada na imagem como “Principal”. Nesta janela o professor tem uma listagem completa de todos os alunos inscritos à unidade curricular. Na listagem é apresentada uma fotografia do aluno, o número de aluno, o nome, e, ainda, o valor do perfil do aluno.

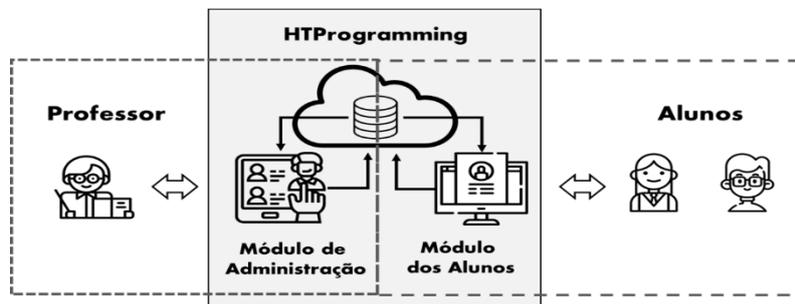


Fig. 1. Esquema geral de descrição da plataforma *HTProgramming*.

Através da seleção de um aluno, o professor pode obter os “Dados individuais do Aluno”, ou seja, o permite visualizar todas as atividades executadas pelo aluno, assim como, os seus resultados. O professor tem, também aos seus dispor a inserção de atividades de codificação e *Parson Problems*, identificadas na Fig. 2 como “Definição de atividades de codificação” e “Definição de atividades de Parson Problems”, respetivamente.

A aplicação permite gerar os dados a utilizar no sistema preditivo, identificado na Fig.2 como “Resumo de dados A utilizar Expert System”. Destacamos, também, a funcionalidade “Gamificação Champions League”. Nesta funcionalidade são sorteados, aleatoriamente ou por perfil, os jogos entre os alunos. Esta funcionalidade surge do trabalho realizado sobre a utilização de gamificação em sala de aula, onde se constatou ser uma das atividades que suscitou maior interesse e entusiasmo.



Fig. 2. Principais destaques do módulo de administração.

Módulo dos Alunos. Este módulo destina-se à utilização pelos alunos inscritos na unidade curricular. Na Fig. 3 apresentamos algumas das funcionalidades da aplicação ao dispor do aluno. A aplicação inicia por um sistema de identificação do aluno, representado na figura por “Login”. O aluno deve completar os seus dados pessoais, como a sua data de nascimento, o email e outros, na área “Dados pessoais do Aluno”. O aluno pode escolher o conjunto de atividades que deseja trabalhar. No entanto, é sugerido iniciar questionários de caracterização do aluno em programação e de aptidão cognitiva, *Punched Holes*, funcionalidades não apresentadas na figura. De seguida, o aluno tem ao seu dispor um conjunto de atividades relacionadas diretamente com os conteúdos abordados na unidade curricular de introdução à programação.



Fig. 3. Destaques do módulo do aluno.

A funcionalidade identificada na Fig. 3 como “Atividades Conceitos Básicos”, contém atividades relacionadas com conteúdos introdutórios a qualquer linguagem de

programação, como: nomes e identificadores, tipos de dados, palavras reservadas e instruções de input e output. Na funcionalidade “Atividades Parson Problems” o aluno é confrontado com várias programas que são apresentados com as suas instruções desordenadas. O objetivo do aluno é ordenar as instruções de código de um programa com um propósito específico. As “Atividades Codificação”, tal como o próprio nome indica, são atividades onde o aluno tem de resolver problemas de codificação. Este conjunto de atividades de codificação estão identificados por grupos, como: conceitos básicos, estruturas condicionais, estruturas iterativas, arrays, e conceitos avançados. Nesta funcionalidade da aplicação *HTProgramming*, é apresentado um enunciado, ao qual o aluno deve escrever o programa num qualquer editor de texto e gravar com a com a extensão habitual do programa em linguagem C. Depois deve submeter o programa na aplicação e testar a sua funcionalidade. O programa submetido é testado com um conjunto de testes predefinidos pelo professor e avaliado, com o valor entre zero e vinte, classificação habitualmente utilizada. É importante salientar que todas as atividades são de resultado imediato, ao concluir uma atividade é apresentado ao aluno o seu resultado.

De acordo com o resultado obtido pelo aluno em cada uma das atividades, é gerada uma mensagem ou comentário para o aluno e professor, de acordo com o limiar definido pelo professor. Esta funcionalidade está identificada na Fig. 3 como “Sugestões e Comentários sobre a atividades realizada”.

6 Resultados até à data

Nos últimos cinco anos, tal como descrito, efetuamos um conjunto de trabalhos e experiências em busca de respostas para o nosso problema, as dificuldades de ensino e aprendizagem inicial da programação. Nos diversos trabalhos apresentados, seguimos várias abordagens de acordo com a nossa experiência e pelas sugestões de outros trabalhos efetuado na área. Ao fim destes anos, culminamos com a construção de uma aplicação que agrega este conjunto de atividades, ou experiências, com o intuito de melhorar e desenvolver as competências do aluno em programação, mas também ajudar o professor nessa árdua tarefa. A aplicação desenvolvida permitirá aferir em cada momento o estado de cada aluno e, também, em face dos resultados conhecidos prever o seu sucesso ou insucesso. Para além, das sugestões automáticas para corrigir problemas, o professor pode intervir ajudando o aluno a superar as suas dificuldades.

Até à data os resultados obtidos são animadores e encorajadores. No entanto, no momento da elaboração deste trabalho ainda não temos todos os resultados da unidade curricular, uma vez que ainda não terminou. Por outro lado, os resultados obtidos pela aplicação *HTProgramming* estão bem organizados e com detalhe. Os resultados antes da aplicação, estão organizados de outra forma, o que torna difícil o treino da nossa rede neuronal e, a conseqüente previsão do sucesso e insucesso do aluno.

7 Estado da dissertação

Estamos fortemente dependentes da obtenção dos dados dos alunos e da ferramenta de recolha de dados. Por outro lado, a unidade curricular é parte integrante do plano curricular do curso de Engenharia Informática, do 1º ano e no 1º semestre, pelo que se alguma coisa não acontece como o esperado, temos de esperar até ao próximo ano para a realização da experiência. Na nossa opinião, a aplicação desenvolvida é a solução para uma correta recolha e análise dos dados. Este ano letivo é o primeiro ano que a aplicação foi efetivamente utilizada em sala de aula, pelo que foi importante realizar alguns ajustes e alterações ao desenvolvimento da aplicação. É importante continuar a utilização da aplicação para melhorar a sua eficiência.

É importante referir que o surto de COVID-19 teve um forte impacto nas nossas vidas e no normal funcionamento do processo de ensino e aprendizagens [9]. Foi necessário fazer grandes alterações de comportamento e funcionamento, pelo que foi de alguma forma comprometido o nosso trabalho.

Consideramos nosso trabalho de grande utilidade para encontrar a solução para os problemas de ensino e aprendizagem inicial da programação. A análise dinâmica e constante dos alunos na programação, com base na construção do perfil do aluno, permitirá avaliar o aluno em cada momento do seu percurso e agir de forma imediata. Este modelo permitirá detetar precocemente as dificuldades e problemas dos alunos e incentivá-los a melhorar e a sentir que estão a ser acompanhados de forma muito pessoal, uma vez que os exercícios são personalizados de acordo com as suas necessidades e características. Com o modelo descrito será possível estabelecer uma relação através do perfil do aluno e os seus resultados finais do curso de programação. Uma rede neuronal treinada que nos dá a probabilidade de um aluno com determinado perfil tem insucesso no curso. Tudo isto auxiliado pela aplicação *HTProgramming*, que será uma ferramenta preciosa para atingir os objetivos propostos.

References

- [1] Bennedsen, J. and Caspersen, M.E. 2019. Failure Rates in Introductory Programming: 12 Years Later. *ACM Inroads*. 10, 2 (2019), 30–36. DOI:<https://doi.org/10.1145/3324888>.
- [2] Bennedsen, J. and Caspersen, M.E. 2007. Failure rates in introductory programming. *ACM SIGCSE Bulletin*. 39, 2 (Jun. 2007), 32–36. DOI:<https://doi.org/10.1145/1272848.1272879>.
- [3] Figueiredo, J. et al. 2016. Ne-course for learning programming. *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM '16* (New York, New York, USA, 2016), 549–553.
- [4] Figueiredo, J. et al. 2019. Predicting Student Failure in an Introductory Programming Course with Multiple Back-Propagation. *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM'19* (New York, New York, USA, 2019),

- 44–49.
- [5] Figueiredo, J. and García-peñalvo, F.J. 2020. Increasing student motivation in computer programming with gamification. *2020 IEEE Global Engineering Education Conference (EDUCON)* (Porto, 2020), 997–1000.
 - [6] Figueiredo, J. and García-Peñalvo, F.J. 2018. Building Skills in Introductory Programming. *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM'18* (New York, New York, USA, 2018), 46–50.
 - [7] Figueiredo, J. and García-Peñalvo, F.J. 2017. Improving Computational Thinking Using Follow and Give Instructions. *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM 2017* (New York, New York, USA, 2017), 1–7.
 - [8] Fincher, S. et al. 2005. Computer Science at Kent programming courses. 1 (2005).
 - [9] García-Peñalvo, F.J. et al. 2021. Impact of the COVID-19 on Higher Education: An Experience-Based Approach. *Information Technology Trends for a Global and Interdisciplinary Research Community*. (2021), 18. DOI:<https://doi.org/10.4018/978-1-7998-4156-2.ch001>.
 - [10] González, R.-A.M. 2007. *La investigación en la práctica educativa: Guía metodológica de investigación para el diagnóstico y evaluación en los centros docentes*. Ministerio de educación y ciencia, Dirección general de educación, formación profesional e innovación educativa, Centro de Investigación y Documentación Educativa (CIDE).
 - [11] Jenkins, T. 2002. On the Difficulty of Learning to Program. *Language*. 4, (2002), 53–58. DOI:<https://doi.org/10.1109/ISIT.2013.6620675>.
 - [12] López, A. and García-Peñalvo, F. 2020. Evaluación de habilidades del pensamiento computacional para predecir el aprendizaje y retención de estudiantes en la asignatura de programación de computadoras en educación superior. *Revista de Educación a Distancia (RED)*. 20, (2020). DOI:<https://doi.org/10.6018/red.409991>.
 - [13] Luxton-Reilly, A. et al. 2018. Introductory programming: a systematic literature review. *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education - ITiCSE 2018 Companion* (New York, New York, USA, 2018), 55–106.
 - [14] Qian, Y. and Lehman, J. 2017. Students' Misconceptions and Other Difficulties in Introductory Programming. *ACM Transactions on Computing Education*. 18, 1 (Oct. 2017), 1–24. DOI:<https://doi.org/10.1145/3077618>.
 - [15] Sleeman, D. 1986. The challenges of teaching computer programming. *Communications of the ACM*. 29, 9 (Sep. 1986), 840–841. DOI:<https://doi.org/10.1145/6592.214913>.
 - [16] Yin, R.K. 2017. *Case Study Research and Applications: Design and Methods*. SAGE Publications.