

# Ensinar programação em ambientes *e-learning*: preocupações e propostas no âmbito do modelo pedagógico virtual da Universidade Aberta de Portugal

*Teaching programming in e-learning environments: concerns and proposals within the scope of the virtual pedagogical model of the Open University of Portugal*

Marcos Luiz Mucheroni\*  
Elizabeth Simão Carvalho\*\*  
Adérito Fernandes Marcos\*\*\*

## Resumo

O ensino e aprendizagem da programação de computadores em ambiente *e-learning* constitui uma área que tem vindo a atrair um interesse crescente tanto por parte das universidades de ensino em rede como pelos investigadores do *e-learning*. Aprender a programar computadores requer o desenvolvimento de raciocínio lógico da parte do estudante, enquanto aprende a dominar as potencialidades de uma determinada linguagem de programação, a construir estruturas de dados e a desenvolver uma estratégia de resolução de problemas pela sua divisão e modularização em problemas mais simples de mais fácil conquista. Adicionalmente importa que o estudante desenvolva competências de trabalho em grupo, de estudo individual com elevados níveis de concentração para realizar trabalho de revisão e depuração de código fonte. O ensino *on-line* de programação de computadores é constituído por estes aspetos, exigindo um elevado grau de interação entre estudantes e destes com o professor. Neste artigo, apresentamos um conjunto de reflexões e propostas concretas de estratégias de implementação do ensino e aprendizagem da programação de computadores, de nível universitário, explorando o método de “design” de instruções como uma abordagem do diálogo, uma adaptação da taxonomia de Bloom, com base na experiência e no modelo pedagógico virtual da Universidade Aberta de Portugal.

*Palavras-chave:* *E-learning*. Ensino e aprendizagem da programação. Ensino-aprendizagem *on-line*. Classe virtual. Modelo pedagógico virtual.

Recebido em 30/09/2018 – Aprovado em 12/02/2019

<http://dx.doi.org/10.5335/rep.v26i2.8703>

\* Doutor em Engenharia Elétrica pela Universidade de São Paulo. Pós-doutor pela Universidade Aberta de Lisboa, Portugal. Integrante do Centro de Investigação em Arte e Comunicação, Universidade Aberta, Lisboa, Portugal. E-mail: mucheroni.marcos@gmail.com

\*\* Doutora em Tecnologia e Sistemas de Informação pela Universidade do Minho. Integrante do Centro de Investigação em Arte e Comunicação, Universidade Aberta, Lisboa, Portugal. E-mail: elizabeth.carvalho@uab.pt

\*\*\* Doutor em Engenharia pela Darmstadt University of Technology e em Informática pela Universidade do Minho. É professor catedrático de nomeação definitiva da Universidade Aberta, no Departamento de Ciências e Tecnologia, Lisboa, Portugal. E-mail: aderito.marcos@gmail.com

## Abstract

The teaching and learning of computer programming in e-learning environment is an area that has attracted increasing interest from both the universities in network and the e-learning researchers. Learning to code in computers requires the development of logical reasoning on the part of the student, while learning to master the potentialities of a particular programming language, building data structures and developing a strategy of problem solving by its division and modularization into simpler problems of easier conquest. In addition, it is important that the student develops group working skills, besides individual studies maintaining high levels of concentration to perform review and debugging work of the source code. The online teaching of computer programming consists of these aspects, demanding a high degree of interaction between students and these with the teacher. In this article we present a set of concrete reflections and proposals of strategies for the implementation of teaching and learning of computer programming at university level, exploring the method of "design" of instructions as an approach to dialogue, an adaptation of Bloom's taxonomy, based on the experience and virtual pedagogical model of the Open University of Portugal.

*Keywords:* E-learning. Teaching and learning in programming. Teach-learning online. Virtual class. Virtual pedagogic model.

## Introdução

As primeiras linguagens de computação em *Assembler* e *Fortran* tinham a preocupação de tornar os problemas matemáticos ou numéricos em código de computador, partindo de dois princípios fundamentais: era preciso traduzir rotinas de tratamento matemático e numérico em código, e, em segundo lugar deveriam ser aplicadas a diversas entradas possíveis de modo que o código fosse escrito apenas uma vez. A evolução para linguagens comerciais como o *Cobol*, na década de 1960, impuseram um ritmo novo, onde era necessário rapidamente formar programadores para que a crescente indústria de computadores mainframes dispusesse de profissionais, essencialmente operadores e programadores, capazes de atender às crescentes demandas do comércio, empresas estatais e industriais. No sentido de dar resposta a essa demanda, surgiu o BASIC (*Beginner's All-purpose Symbolic Instruction Code*), uma linguagem simples e rápida de programar, com código interpretado, quer dizer, feito linha a linha. Importa recordar aqui que esta foi também a época dos grandes algoritmos e que a ideia de otimização deu origem a programas de busca e ordenação (*Shell and Sorting*), os famosos algoritmos de Donald Knuth (1997) e Edsger Dijkstra (1997), ambos em terceira edição, mas datados na primeira edição nos anos 60 e 70, respectivamente.

Após um período em que as linguagens voltaram a se tornarem complexas muito focalizadas em instruções de baixo nível, ao nível da máquina, surgiram as linguagens *Pascal*, *Prolog* e *Smalltalk*, entre outras, orientadas para os ambientes caseiros e traziam conceitos de Inteligência Artificial, tendo-se popularizado a lin-

guagem C, inicialmente feita para Sistemas Operativos, levando a que uma nova etapa de simplificação tenha visto o seu advento.

Hoje são populares linguagens como o *PHP*, *Python* e agora a emergente *Julia 1.0* (2018), nos ambientes de programação para telemóveis e desenvolvimentos caseiros, como o *Arduíno* que promete crescer na *IoT* (Internet das Coisas), Realidade Aumentada e Virtual, sendo que esta linguagem contempla este novo cenário da programação. Neste meio tempo a rede *World Wide Web* ou simplesmente *Web*, fez evoluir a programação *Java*, posteriormente comprada pela empresa de bancos de dados *Oracle*, que após uma guerra com desenvolvedores, permaneceu livre e presente entre programadores de código aberto (*Open Source*).

Existem outras linguagens e factos históricos relevantes, como o surgimento do *Linux* feito em *crowdsourcing* por programadores em todo mundo, o surgimento do *Wikipedia* e o *DBPedia*, a Web Semântica e a linguagem de marcação *XML*, as ontologias, etc.

O importante aqui é enfatizar dois pontos em constante tensão: a lógica de programação e o desenvolvimento de algoritmos que independentemente da linguagem requerem treinamento e a programação, depuração e versionamento em ambientes de programação que requerem um esforço de trabalho relativamente elevado, uma disciplina férrea necessária à depuração de código e, muitas vezes, um esforço de trabalho em equipe, competências que nem sempre são ensinadas e treinadas nos cursos de programação de computadores, sejam estes presenciais em sala, ou cursos *on-line* em regime de *e-learning*.

O esforço extra verificado nos cursos *on-line* em regime de *e-learning* é o de procurar estabelecer uma disciplina de evolução no tratamento do código, indo do algoritmo ao programa final e suas versões, sem que o estudante perca a noção de conceitos importantes que evoluem dentro desta lógica. Se há algo em que o sucesso do ensino e aprendizagem *on-line* pode ser significativo é justamente a realização de cursos de programação, admitindo-se que estes não padecem das enfermidades graves da programação, i.e., os algoritmos feitos na força bruta sem serem pensados e amadurecidos, os ambientes de aprendizagem acelerada que queimam etapas, e por último, o esforço de versionamento e de detecção de *bugs*.

Com um ensino revolucionário desde a mudança curricular em seu ensino básico até ao ensino especializado de programação *on-line*, a Finlândia oferece uma rica experiência para o ensino de programação.

Belle Selene Xia (2017), do Departamento de Informação e Ciência da Computação da Universidade de Aalto da Finlândia, realizou um importante estudo para

entender como manter a motivação, entender os estilos de aprendizado, colaborando com a qualidade de ensino em ambientes *on-line*.

O objetivo de seu trabalho consistiu em analisar como diferentes teorias educacionais podiam colaborar com o estudo e a aprendizagem, assim como em superar problemas como as taxas de retenção e o problema da evasão.

O artigo começa apontando o problema que tocamos na introdução, citando Lister e Leaney (2003), onde explica que o ponto de vista do professor deve ter como objetivo desenvolver nos estudantes a capacidade de implementar programas, menos que expressar os problemas de computação expressos apenas em termos de programação, sem esquecer a motivação dos estudantes que é objetivo principal da aprendizagem, alguns experimentos (XIA, 2013; XIA; LIITIÄINEN, 2013) foram feitos pela autora.

Definindo de forma ampla o que é o ensino e a experiência pessoal em determinado fenômeno, Bruse et al. (2004) propõem que o aprimoramento de um estudante em um fenômeno dado significa a compreensão adequada das dimensões críticas de sua experiência para este fenômeno. Esta autora também definiu as principais dificuldades que incrementam as taxas de retenção / insucesso nos cursos de programação:

- a) dificuldades na leitura e escrita de códigos de programa: Para poder escrever corretamente códigos de programa, os estudantes precisam primeiro ler códigos de programa, que é um dos principais desafios da programação da educação, especialmente para aqueles sem ciência da computação anterior fundo (BUCK; STUCKI, 2000);
- b) dificuldades na criação de declarações de programa: os alunos de programação não aprendem a escrever e programas de design como habilidades de programação, são cognitivamente habilidades complexas que exigem uma profunda compreensão de conteúdo estruturalmente complexo (DEHNADI; BORNAT, 2006);
- c) dificuldades em rastrear os códigos do programa: os resultados da pesquisa mostraram que um grande grupo de estudantes não puderam rastrear os programas de maneira sistemática mediante solicitação. Este fenômeno é especialmente evidente entre os programadores iniciantes (LISTER; LEANEY, 2003).

O ensino-aprendizagem da programação, se em ambiente *on-line* de *e-learning*, acrescenta dificuldades específicas ao nível da necessidade de retorno (*feedback*),

as revisões e os exames de proficiência ao nível do domínio da linguagem, as etapas em que este acompanhamento deve ser feito para cursos onde o ensino é contínuo e assíncrono, onde etapas intermediárias podem ser realizadas a qualquer momento pelo estudante, e ainda o complexo relacionamento com os monitores e professores que devem entender a etapa e a sequência que os estudantes realizam ainda que em sequência assíncrona e a distância.

É necessário portanto um ambiente de “design” de instruções contextualizado porém com uma adequada concepção metodológica (FILATRO, 2004) que lhe forneça significação, com princípios teóricos que se relacionem à prática educativa e possibilite um processo de ensino-aprendizagem de qualidade e que atenda às demandas do mundo do trabalho e de uma realidade onde estão cada vez mais presentes os artefatos digitais.

Para fazer emergir a criatividade, a arte e uma forma de sabedoria prática, que os gregos chamavam de *phronesis*, como um acréscimo à *práxis* e a *techné*, onde o ensino atual requer maior aproximação afetiva e cognitiva para o desenvolvimento, para isto uma metodologia mais aderente ao desenvolvimento criativo e único de algoritmos, e cada colaboração traz algo de único, o que é pensado numa metodologia fenomenográfica, e a criação consequente.

## A taxonomia de Bloom, estudos fenomenográficos e “design” de instruções

Recorrer ao simples processo de cópia e colagem (*copy and paste*) de informação conforme está disponível na Web para conhecimento de um determinado assunto tem vindo a assumir-se como uma prática corrente no mundo do ensino, também de nível universitário, somente contrariado com a imposição de maiores níveis de exigência, assim como o incentivo da criatividade e do desenvolvimento cognitivo, afetivos e psicomotores, como propõe Bloom (KRATHWOHL, 2002), e pode fazer emergir a necessária criatividade e arte, que é necessária e aplicável ao desenvolvimento de algoritmos. A classificação proposta por Bloom dividiu as possibilidades de aprendizagem em três grandes domínios:

- a) o cognitivo, abrangendo a aprendizagem intelectual;
- b) o afetivo, abrangendo os aspectos de sensibilização e gradação de valores;
- c) o psicomotor, abrangendo as habilidades de execução de tarefas que envolvem o aparelho motor.

A taxonomia de Bloom é adequada e adaptável à programação, porque, em sua visão, para a solução de problemas que exigem capacidades intelectuais específicas, o estudante deve organizar e reorganizar um problema, reconhecer o material necessário e usá-lo no contexto do problema, isto quando o problema é novo e não familiar ao estudante, i.e., esta é a capacidade de elaborar algoritmos.

A partir da Taxonomia de Bloom pode-se explorar o método da fenomenografia, que nasceu de uma pergunta de “porque uma pessoa aprende melhor que a outra” (MARTON, 1994, p. 4424) e se afasta do mito que afirma que algumas pessoas serão “mais aptas” que outras, existindo diferenças, porém não se deve aplicar isto genericamente, aplica-se ao ensino-aprendizagem *on-line*, pois é uma forma de inclusão, abarcando as diferenças dos estudantes.

Utiliza-se o exemplo da linguagem *Julia 1.0* apresentada em setembro de 2018, com aval do MIT (Massachusetts Institute of Technology), mas com grande possibilidade de tornar-se emergente, para exemplificar o uso de Taxonomia de Bloom em ensino de programação *on-line* (Figura 1).

Figura 1 – Etapas de aprendizagem na linguagem *Julia 1.0* recém-lançada



Fonte: dos autores.

A fenomenografia pretende ser, a partir da ciência empírica da aprendizagem, uma réplica da análise fenomenológica, procurando destacar-se, ao mesmo tempo, tanto da filosofia como da psicologia, ou seja, visa relevar os aspectos cognitivo,

psicomotor e afetivo de Bloom. Algumas propostas do grupo de Gotemburgo são sintetizadas sobre a investigação do mundo da experiência, na medida em que se refere à aprendizagem colaborativa de alunos e de professores, remodelando portanto o modelo da sala de aula, uma abordagem com forte aplicação no ensino-aprendizagem online.

Deve-se colocar em parênteses a própria experiência e olhar o outro, tratando-se de descrever qualitativamente os outros no diálogo, numa entrevista ou numa aula presencial ou não. Os componentes desta abordagem são também pioneiros na consideração de diferentes estilos de aprendizagem, não se tratando, portanto, de abandonar este método, mas de ampliá-lo para cenários de ensino-aprendizagem *on-line*. Por último, é necessário apresentar o método fenomenológico de categorização dos dados, um exemplo de como um grupo de professores que conceba a aprendizagem de seus estudantes, como uma proposta de prática para fazer em aula e também chegar a conclusões. Aqui o *design* instrucional, geralmente mais ligado à estética do que a um método de diálogo, tem conforme Filatro (2004, p. 6) a intenção de dirigir-se à: “[...] ação intencional e sistemática de ensino, que envolve o planejamento, o desenvolvimento e a utilização de métodos, técnicas, atividades, materiais, eventos e produtos educacionais em situações didáticas específicas, a fim de facilitar a aprendizagem humana a partir dos princípios de aprendizagem e instrução conhecidos”, ou seja, uma desfiliação a métodos rígidos e automáticos.

O autor citado não está nem na origem nem entre os citados do design instrucional, tem duas vantagens que devem ser ressaltadas, desfiliação das pedagogias “presenciais” tais como: o construtivismo, skinerismo e outras, e uma forte adesão à tecnologia com contextualização.

Deve-se, então, pensar em um conjunto de tarefas, e onde se pretende aplicá-las na construção de algoritmos; utilizando a fundamentação dos objetivos de aprendizagem, uma tarefa, no caso algoritmos para programação devem ser:

- a) informar os objetivos da aprendizagem;
- b) ser formativo – mobilizando habilidades e competências;
- c) ser processual – graduando a complexidade das atividades;
- d) provocar reflexão sobre a prática da elaboração antes da programação;
- e) levar a mudança de comportamentos, valores e atitudes na elaboração.

É um fato aceito, mesmo no mundo acadêmico, que etapas muito demoradas na construção de algoritmos podem afastar bons estudantes do universo de programação, então os aspectos cognitivo e afetivos da taxonomia de Bloom não podem



ser negligenciados, isto pode ser pensado em algoritmos criativos, a computação tem um número grande de exemplos desde algoritmos de Classificação e Ordenação, Torre de Hanói, o algoritmo de Dijkstra do caminho num grafo, com um número grande de aplicações atuais, e muitos outros.

O aspecto afetivo já foi ressaltada a questão colaborativa, fundamental no desenvolvimento de programação atual, mas também deve-se acrescentar a inserção em fóruns e o incentivo à discussão entre os alunos de fatores relevantes de desenvolvimento dos algoritmos propostos, onde um fórum vazio pode indicar que a questão afetiva foi negligenciada.

Por último, o aspecto psicomotor, em que ambientes e plataformas de *e-learning* evidenciam muitas vezes ter dificuldades intrínsecas pelo que se devem agilizar estas explicações, retomando sempre que preciso as formas e ambientes de programação com suas bibliotecas de programas prontos, indicar sítios Web de buscas, comunidades online e outras, mas principalmente retornar às explicações sobre o bom uso de plataformas sempre que for necessário, pois muitas vezes o aluno mais tímido evita fazer questões que podem parecer óbvias, mas que nem sempre o são, e, portanto, a plataforma deve ser dominada em um processo que é progressivo: depuração, compilação e bibliotecas de linguagens.

## Um quadro de reflexão sobre aprendizagem de programação

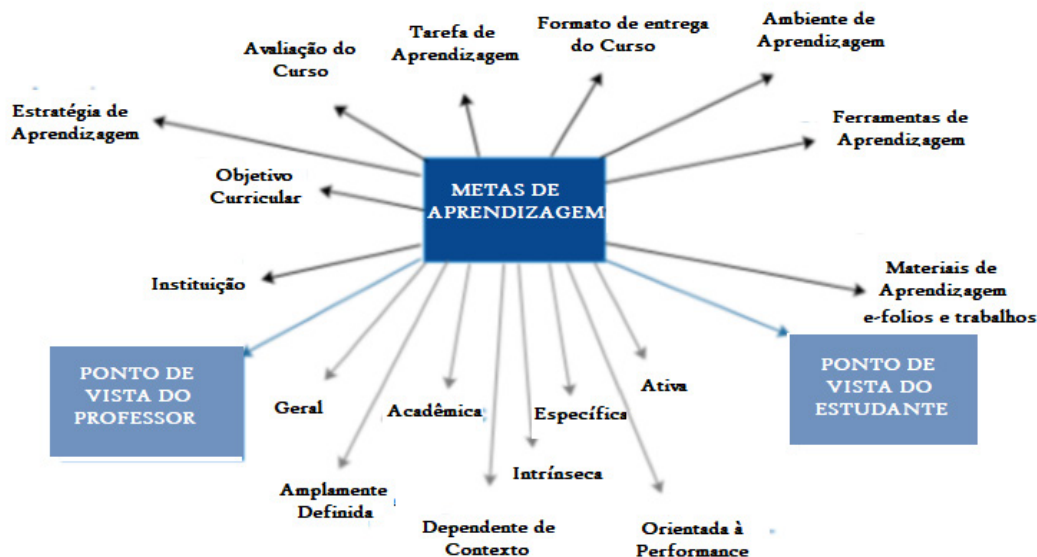
Os autores propõem um conjunto de reflexões que afetam diretamente a aprendizagem, em especial no ensino de programação, fazendo uma adaptação de seu quadro conceitual ao de Bellen Selen Xia (2017), pode-se definir alguns pontos neste enfoque: definição de metas de aprendizagem é usada para que os resultados sejam motivantes e empáticos, assim elas devem ser monitoradas e controladas de acordo com as demandas do curso e os contextos de aprendizagem.

Conforme o experimento feito (XIA, 2017), foi verificado que em alguns casos as metas de aprendizagem podem ser usadas de forma intercambiável com as metas de desempenho e resultados de aprendizado, o que é perfeitamente aplicável ao ensino *on-line* estabelecendo etapas de aprendizado. Outros fatores considerados são a experiência prévia do aluno, a idade e a atitude em relação à aprendizagem,



preferências de formato de curso e o período médio do estudo são os principais fatores explicativos por trás dos diferentes resultados, o que foi verificado em outros trabalhos (XIA, 2013; XIA; LIITIÄINEN, 2013), conforme Figura 2.

Figura 2 – Quadro de fatores que afetam as metas educacionais



Fonte: adaptado de Xia (2017).

A análise ampliada de fatores que afetam as metas educacionais incluem as metas de aprendizagem do ponto de vista do aluno: em seus parâmetros dependentes de contexto que incluem as metas acadêmicas intrínseca, específica e ativa, com materiais de aprendizagem (e-fólios e trabalhos, por exemplo); do ponto de vista do professor: os objetivos gerais acadêmicos, as definições amplas e as dependentes de contexto.

O aspecto contextual para uma universidade que tenha um conjunto de usuários espalhados por diferentes realidades também deve ser levado em conta, embora os objetivos acadêmicos devam ter definições amplas, do ponto de vista do estudante, é preciso entender o contexto pessoal, econômico e cultural para que o aluno possa ter um desenvolvimento efetivo.

Estas metas devem ser pensadas, do ponto de vista do Estudante e do Professor, após a consolidação das matrículas, analisando o conjunto dos alunos e observando aspectos específicos quando forem necessários.

## O ensino-aprendizagem a distância *on-line*

A internet tornou-se uma das formas vitais para disponibilizar recursos para pesquisa e aprendizagem para professores e para estudantes compartilhar e adquirir informações (RICHARD; HAYA, 2009). O ensino-aprendizagem a distância (EAD) *on-line* é baseado em tecnologia que engloba a utilização da internet e de outras tecnologias importantes para produzir materiais para o aprendizado, ensinar alunos e também regular cursos em uma organização (FRY, 2001).

Existem diversas formas de classificar os tipos de ensino a distância. De acordo com Algahtani (2011), houve algumas classificações baseadas na extensão de seu envolvimento no processo de educação. Algumas classificações são baseadas também no sincronismo da interação. Algahtani (2011) dividiu o ensino a distância em dois tipos básicos, consistindo no ensino a distância baseado no computador e na internet.

O EAD é descrito por Nichols (2003) como “a educação que ocorre somente através da Web”, isto é, ele não consiste de quaisquer materiais de aprendizagem físicos enviados para alunos ou contato presencial. A aprendizagem puramente *on-line* é, essencialmente, o uso de ferramentas de *e-learning* em uma modalidade de educação a distância, utilizando a Web como o único meio para toda a aprendizagem do aluno e contato. Alguns estudos dão vantagem ao ensino a distância em função da sua capacidade de se concentrar nas necessidades que os estudantes apresentam no ambiente virtual de aula. Por exemplo, Marc (2000), em sua pesquisa sobre estratégias de aprendizagem para a distribuição do conhecimento na era digital observou que uma das vantagens de do ensino a distância é exatamente esse foco diferenciado, pois no ensino tradicional, a ênfase é normalmente dada às necessidades dos professores ou ainda, das próprias instituições de ensino.

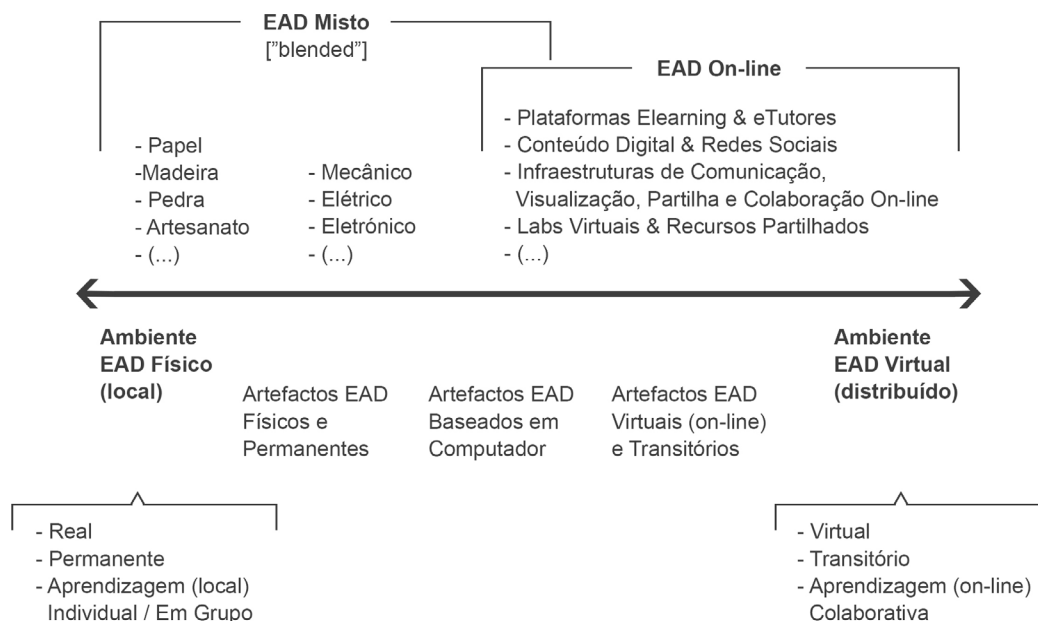
Existem muitas vantagens na adoção de um ensino a distância, embora como evidente, também existam alguns senões. A lista a seguir identifica algumas das melhores potencialidades que este tipo de ensino pode introduzir:

- a) é extremamente flexível em relação ao tempo ou a localização em que o ensino/aprendizagem decorrem (SMEDLEY, 2010);
- b) a quantidade de informação que é possível aceder é maior, pois com o auxílio de recursos eletrônicos (vídeos, hiperligações, demos, etc.), o espaço da sala de aula virtual é enriquecido, permitindo um aumento dos conhecimentos e qualificações;

- c) existe a possibilidade de conversação assíncrona entre os alunos e alunos-professor (normalmente com fóruns de discussão), mas também é possível esta decorrer pontualmente de forma síncrona (videoconferência);
- d) as barreiras que podem existir pela timidez ou outros fatores limitantes em nível de participação são minoradas, pois a base de funcionamento do ensino é a elevada interação entre os participantes (normalmente com fóruns);
- e) é uma solução de baixo-custo, pois não exige deslocamentos de alunos ou ainda encargos aumentados pela presença de alunos no espaço, por parte da instituição de ensino;
- f) a aprendizagem sempre leva em consideração as diferenças de aprendizagem dos alunos. Alguns alunos, por exemplo, preferem se concentrar em determinadas partes do curso, enquanto outros estão dispostos a rever todo ele;
- g) o ensino a distância reduz a necessidade de professores, técnicos de laboratório entre outros elementos humanos que normalmente tem que existir num ensino presencial;
- h) a utilização do ensino a distância permite a autoestimulação. Por exemplo o modo assíncrono permite que cada aluno possa estudar no seu próprio ritmo e velocidade, lenta ou rápida. Concluindo-se, aumenta a satisfação e diminui o *stress* (CODONE, 2001; AMER, 2007; URDAN; WEGGEN, 2000; ALGAHTANI, 2011; MARC, 2002; KLEIN; WARE, 2003).

Os objetos e artefactos de ensino e de aprendizagem são de natureza informacional e comunicacional; eles são construídos propositadamente para facilitar e implementar o processo de aprendizagem. Sua criação visa reforçar estados de espírito, para induzir a curiosidade e construções cognitivas. Os objetos de aprendizagem são partes basilares de todos os ambientes de ensino e aprendizagem. Estes podem ser definidos como parte de um meio contínuo de ensino e aprendizagem que inclui ambientes muito físicos e presenciais até espaços virtuais distribuídos *on-line* (Figura 3).

Figura 3 – O *Continuum Medium* de ensino-aprendizagem a distância



### CONTINUUM MEDIUM DE ENSINO-APRENDIZAGEM A DISTÂNCIA

Fonte: Carvalho e Marcos (2016).

O EAD é muito frequentemente baseado nos princípios da aprendizagem centrada no aluno; flexibilidade (espacial e temporal) de aprendizagem; e interação *on-line*, em particular, a interação assíncrona, o que desfoca as barreiras temporais impostas pelo sincronismo comunicacional e é consistente com o princípio da flexibilidade. A interação é absolutamente fundamental para o processo de ensino-aprendizagem para que os alunos possam efetivamente adquirir os conhecimentos e competências correspondentes. Ela ocorre quando os estudantes estão a participar ativamente em atividades que impliquem uma comunicação entre pares e com o professor, seja contribuindo para uma discussão, resolvendo um exercício, analisando resultados, simplesmente trocando pontos de vista com os seus colegas, ou esclarecendo dúvidas com o professor (CARVALHO; MARCOS, 2016).

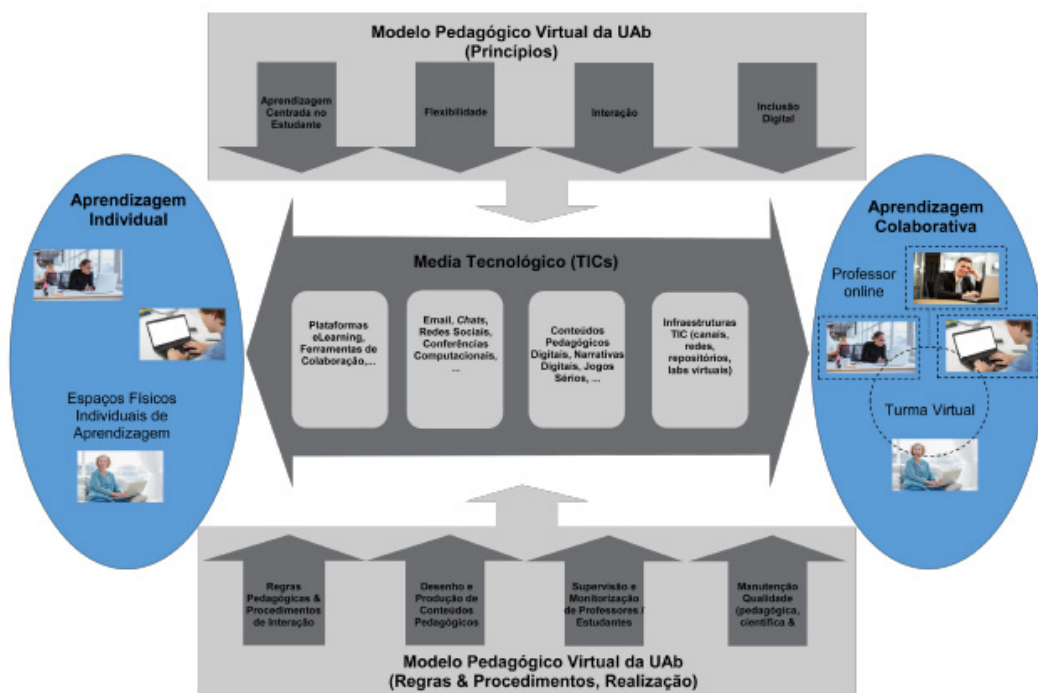
Além disso, o EAD ou simplesmente o *e-learning*, é necessariamente baseado em um ambiente tecnológico de formação e, como tal, deve estar enraizado em estruturas epistemológicas para poder proporcionar um processo eficaz de ensino-

-aprendizagem (DABBAGH, 2005). O EAD refere-se a novas formas de interação e aprendizagem que incluem as facilidades oferecidas pelas tecnologias da informação e comunicação, como a participação conjunta de especialistas de todo o mundo e estudantes nas mesmas sessões *on-line*, ou o acesso imediato aos recursos globais, a oportunidade de se comunicar com um público diversificado, ou a capacidade de compartilhar informações e processos em execução ou em co-construção de conhecimento, entre outros. Estas atividades acentuam o EAD em função das interações com os outros e com as ferramentas comuns da comunidade de ensino-aprendizagem, incitando, assim, a necessidade de construções pedagógicas e modelos como o desenvolvido e aplicado na Universidade Aberta (PEREIRA et al., 2007). Tais modelos visam à modelação da aprendizagem distribuída, o ensino aberto / flexível, redes de aprendizagem assíncronas, comunidades de construção de conhecimento e comunidades de prática, como também a definição do ensino a distância como sendo uma organização metódica e coordenada de formas distribuídas de interação e atividades de aprendizagem para alcançar um objetivo comum.

## O Ensino e aprendizagem *on-line* da programação de computadores no Modelo Pedagógico Virtual da Universidade Aberta

O Modelo Pedagógico Virtual (MPV) definido por (PEREIRA et al., 2007), foi adotado na Universidade Aberta (UAb) e descreve em grande detalhe como projetar e implementar contextos de ensino-aprendizagem em ensino a distância (EAD) para os cursos de terceiro 1º, 2º e 3º ciclos. O MPV visa modelar os processos EAD distribuídos ao criar as bases para redes assíncronas de aprendizagem e construção do conhecimento e comunidades de prática entre os estudantes, tutores e professores da UAb. O MPV promove políticas de interação entre alunos, através da aprendizagem colaborativa e é focado na avaliação contínua. Ele baseia-se em quatro grandes pilares. A Figura 4 fornece uma visão geral do MPV.

Figura 4 – Visão geral do Modelo Pedagógico da Universidade Aberta



Fonte: Carvalho e Marcos (2016).

O ensino da programação de computadores exige que o aluno desenvolva um raciocínio lógico, capaz de traduzir a solução de um problema, num algoritmo passível de codificação numa determinada linguagem de programação. O grande desafio é o aluno conseguir raciocinar de uma forma organizada e lógica, da forma como a implementação numa determinada linguagem deve ocorrer. Nem todas as linguagens possuem a mesma sintaxe ou ainda, paradigma (p. ex. o procedimental versus o orientado por objetos).

No ensino tradicional, esse tipo de aprendizagem é feita com o auxílio de aula prática e em laboratórios em que o aluno dispõe de computadores e instrutores para o auxiliar. No ensino a distância, o desafio é bem maior, pois não existe o instrutor ao lado, ou ainda um ambiente computacional configurado e pronto para a sua utilização. O aluno deve ser capaz de realizar de forma autômata a instalação e configuração de seu ambiente de trabalho. Pela sua natureza eminentemente prática, o ensino da programação a distância *on-line* é um dos maiores desafios neste tipo de ambiente de aprendizagem.

Na Universidade Aberta, especialmente no ensino de unidades curriculares de programação, muitas soluções têm sido implementadas e melhoradas ao longo dos anos. Presentemente, a licenciatura de Engenharia Informática, conta com unidades curriculares laboratoriais, onde o aluno pode aceder remotamente Laboratórios de Aprendizagem da CISCO, plataforma de codificação Hackerrank, computadores em cluster entre outras coisas. Dentro das unidades curriculares, os alunos dispõem de testes para autoavaliação, pequenos vídeos explicativos, exemplos de código, entre outros elementos digitais para suporte ao aprendizado. Além disso, a interação assíncrona é fortemente utilizada. Em algumas das unidades curriculares, ocorrem algumas sessões síncronas entre os professores e os alunos.

No MPV, a nível de 1º ciclos, o aluno pode escolher entre duas formas de avaliação: a contínua ou a final. Na avaliação contínua, o aluno realiza dois e-fólios ao longo do semestre letivo, que no caso de unidades curriculares de programação, sempre implicam na codificação de um programa em solução a um problema enunciado no e-fólio. O e-fólio é realizado num determinado prazo máximo (normalmente de 10 dias, incluindo dois finais de semana). Após esse prazo, o aluno efetua o carregamento de seu trabalho (normalmente algum código fonte) e é avaliado numa escala de 0 a 4 para cada um dos e-fólios realizados. O aluno ainda realiza uma prova escrita, designada de p-fólio (que pode se realizar no prazo normal ou em época especial), que tem um valor máximo de 12 pontos. Na avaliação final, o aluno realiza apenas uma prova escrita que vale 20 pontos no máximo. Ambas as provas escritas, igualmente como os e-fólios, incidem sobre questões práticas de codificação.

Para ensinar como desenvolver e implementar (CARVALHO; MARCOS, 2016) o código de um programa de computador em um ambiente baseado completamente em ensino a distância comporta ainda uma série de desafios e limitações. Da nossa experiência em EAD de programação de computadores, detectamos que as maiores dificuldades enfrentadas prendem-se com a promoção da participação dos alunos no processo de ensino-aprendizagem e o seu envolvimento em atividades que exigem o trabalho em grupo através da partilha de problemas ou a criação de uma solução em conjunto.



## Considerações finais

O artigo analisa, à luz da reflexão de Belle Selene Xia (2017), as dificuldades e metas que devem ser pensadas a partir de estilos de aprendizagem que colaborem com a qualidade de ensino de programação *on-line*, e teve por objeto a plataforma de ensino da Universidade Aberta de Lisboa.

Para análise, trouxe o método de “design” de introduções como um método de diálogo, conforme Filatro (2004) que teve o objetivo de dirigir-se a ação intencional e sistemática de ensino, que envolve o planejamento, o desenvolvimento, técnicas, atividades, materiais, eventos e produtos educacionais em ambiente *on-line*.

Traz ainda a reflexão que para o ensino de algoritmos que envolve criatividade e pensamento é necessária, utilizando a Taxonomia de Bloom, que envolve os domínios cognitivo, afetivo e psicomotor, e sua relação com o ensino de programação *on-line*.

Os métodos implementados devem observar os pontos de vista do estudante e do professor, contemplando uma análise ampliada de fatores que interferem nas metas educacionais e seus parâmetros são dependentes de contexto que incluem as metas acadêmicas: intrínseca, específica e ativa, assim como os materiais de aprendizagem (e-fólios e trabalhos, no caso da Universidade Aberta).

A nível da Universidade Aberta, apesar de todos os esforços que têm sido empreendidos no sentido de ultrapassar as dificuldades no aprendizado da programação de computadores, ainda existe um longo caminho a percorrer. O Modelo Pedagógico Virtual precisa de ser revisto e adaptado às novas exigências tecnológicas, nomeadamente o aprendizado da codificação de linguagens ou ainda a configuração de redes e sistemas remotamente. Essa adaptação é necessária, pois, só assim, conseguiremos atender adequadamente a essas demandas e necessidades específicas no sentido de um aprendizado apropriado da programação de computadores. O aperfeiçoamento dos laboratórios virtuais, incluindo tecnologia de realidade virtual ou/e aumentada para suporte a interação e simulação, a inclusão de *workshops* síncronas para exercício de programação, a possibilidade da realização remota de exames de codificação de computadores com computadores ou ainda a inclusão de tutores virtuais *on-line* são algumas das possibilidades a serem exploradas e implementadas. Um melhor controle e, portanto, o conhecimento de como os alunos utilizam os recursos e se inter-relacionam, entre si e com o professor constitui um contributo adicional para desenvolver soluções mais adequadas ao EAD da programação de computadores.

## Referências

- AMER, T. *E-learning and Education*. Cairo: Dar Alshehab publication, 2007.
- ALGAHTANI, A. F. *Evaluating the Effectiveness of the E-Learning experience in Some Universities in Saudi Arabia from male Students' Perceptions*. Durham: Durham University, 2011.
- BRUSE, C. et al. Ways of experiencing the act of learning to program: a phenomenographic study of introductory programming students at university. *Journal of Information Technology Education*, n. 3, p. 143-157, 2004.
- BUCK, D.; STUCKI, D. J. Design early considered harmful: graduated exposure to complexity and structure based on levels of cognitive development. *SIGCSE Bulletin*, v. 32, n. 1, p. 75-79, 2000.
- CARVALHO, E.; MARCOS, A. F. O ensino-aprendizagem da programação de computadores no ensino a distância online: uma proposta de instanciação do modelo pedagógico virtual da Universidade aberta: para uma universidade do futuro. *Modelo Pedagógico Virtual da Universidade Aberta*, p. 1-112, 2016.
- CODONE, S. *Ane-Learning Primer, Raytheon Interactive*. 2001. Disponível em: <[http://faculty.mercer.edu/codone\\_s/learningprimer.PDF](http://faculty.mercer.edu/codone_s/learningprimer.PDF)>. Acesso em: 28 set. 2018.
- DABBAGH, N. Pedagogical models for E-Learning: a theory-based design framework. *International Journal of Technology in Teaching and Learning*, v. 1, n. 1, p. 25-44, 2005.
- DEHNADI, S.; BORNAT, R. *The camel has two humps (working title)*. 2006. Disponível em: <<http://www.eis.mdx.ac.uk/research/PhDArea/saeed/paper1.pdf>>. Acesso em: 28 ago. 2018.
- DIJKSTRA, E. *A discipline of Programming*. 3. ed. [S. l.]: Prentice Hall, 1997.
- FRY, K. E-learning markets and providers: some issues and prospects. *Education Training*, p. 233-239, 2001.
- JULIA 1.0 Blog. Disponível em: <<https://julialang.org/blog/2018/08/one-point-zero>>. Acesso em: 28 ago. 2018.
- KNUTH, D. E. *The art of computer programming: fundamental algorithms*. 3. ed. Stanford: Addison Wesley Longman, 1997.
- KRATHWOHL, D. R. A revision of Bloom's taxonomy: an overview. *Theory into Practice*, v. 41, n. 4, p. 212-218, 2002.
- FILATRO, A. *Design instrucional contextualizado: educação e tecnologia*. São Paulo: Senac, 2004.
- LISTER, R.; LEANEY, J. Introductory programming, criterion referencing, and Bloom. *SIGCSE Bulletin*, v. 35, n. 1, p. 143-147, 2003.
- MARTON, F. Phenomenography. In: HUSÉN, Torsten; POSTLEWAITE, Neville (Ed.). *The International Encyclopedia of Education*. Oxford: Pergamon Press, 1994. p. 4424-4429.
- PINTOR, M. Fenomenología y fenomenografía: Punto de encuentro entre la filosofía y la ciencia en el mundo del aprendizaje. *Paideia: Revista de filosofía y didáctica filosófica*, v. 23, n. 59, 2002.

PEREIRA, A. et al. Modelo pedagógico virtual da Universidade Aberta. *Práticas e Cenários de Inovação em Educação*, p. 190-0, 2016.

NICHOLS, M. A Theory for E-Learning. *Journal of Educational Technology and Society*, n. 6, p. 1-10, 2003.

SMEEELEY, J. K. Modelling the impact of knowledge management using technology. *OR Insight*, v. 23, p. 233-250, 2010.

KLEIN, D.; WARE, M. E-learning new opportunities in continuing professional development. *Learned Publishing*, v. 16, n. 1, p. 34-46, 2003.

URDAN, T. A.; WEGGEN, C. C. *Corporate E-learning Exploring a New Frontier*. San Francisco, CA: WR Hambrecht and Co., 2000. Disponível em: <<http://papers.cumincaad.org/data/works/att/2c7d.content.pdf>>. Acesso em: 28 set. 2018.

MARC, J. R. Book review: e-learning strategies for delivering knowledge in the digital age. *Internet and Higher Education*, n. 5, p. 185-188, 2002.

RICHARD, H.; HAYA, A. examining student decision to adopt Web 2.0 technologies: theory and empirical tests. *Journal of Computing in Higher Education*, v. 21, n. 3, p. 183-198, 2009.

XIA, B. S. Learning outcomes and knowledge sharing using web-based technologies in the Finnish forest education from an educational point of view. *E-Learning and Digital Media*, v. 10, n. 1, p. 95-106, 2013.

\_\_\_\_\_. Na in-depth Analysis of Learning Goals in Higher Education: Evidence from the Programming Education. *Journal of Learning Design*, v. 10, n. 2, 2017.

XIA, B. S.; LIITIÄINEN, E. Economics of education and work life demand in terms of earnings and skills. *Citizenship, Social and Economics Education*, v. 13, n. 1, p. 67-77, 2013.