



CAPÍTULO 7

CAPÍTULO 7

O ENSINO E APRENDIZAGEM DA PROGRAMAÇÃO DE COMPUTADORES NO ENSINO A DISTÂNCIA: UMA PROPOSTA DE INSTANCIAÇÃO DO MODELO PEDAGÓGICO VIRTUAL DA UNIVERSIDADE ABERTA

Elizabeth Simão Carvalho

Centro de Investigação em Arte e Comunicação, Universidade Aberta, Rua da Escola Politécnica, 147, 1269-001 Lisboa, Portugal

Adérito Fernandes-Marcos

Centro de Investigação em Arte e Comunicação, Laboratório de Educação a Distância e eLearning, Universidade Aberta, Rua da Escola Politécnica, 147, 1269-001 Lisboa, Portugal

RESUMO

Hoje em dia o ensino e aprendizagem a distância online a nível universitário são, em grande medida, baseados em estratégias de aprendizagem colaborativa, onde, além de aprenderem sozinhos, os estudantes também participam de ações colaborativas dentro de uma classe virtual. O nível de interação do estudante online representa um fator fundamental para o sucesso do processo de ensino-aprendizagem pois é a base da partilha de informação e construção do conhecimento entre estudantes e professores, enquanto todas as ações e atividades integram um modelo pedagógico comum. Existem muitas diferenças e desafios nas áreas de ensino em termos de instanciação do modelo pedagógico e adoção das estratégias de ensino-aprendizagem, como por exemplo, entre as áreas das ciências sociais e das engenharias. Uma área que atrai especial atenção como um todo é a da ciência da computação (CS), e de forma específica, a da programação de computadores. A programação de computadores exige, em primeiro lugar, o desenvolvimento de um bom raciocínio lógico e uma estratégia de resolução segundo uma abordagem “dividir para conquistar”, onde os principais problemas são divididos em problemas menores que são resolvidos individualmente. A programação exige também uma combinação entre o trabalho individual e em grupo, com elevados níveis de revisão e depuração do código fonte

ABSTRACT

Online distance teaching and learning at university level is nowadays, to a great extent, based on collaborative learning strategies where, besides learning alone, students also take part in collaborative actions within a virtual class. The level of online student interaction represents a pivotal factor for the success of the teaching-learning process as it bases the sharing of information and knowledge construction among students and teachers while all actions and activities integrate a common pedagogic model. Differences and diverse challenges exist among teaching areas regarding the instantiation of the pedagogic model and adoption of the teaching-learning strategies, as for instance between social science and engineering areas. An area attracting special attention is computer science (CS), as a whole, and computer programming specifically. Computer programming requires, first of all, development of a good logical thinking and a problem-solving strategy oriented to “divide to conquer” approach by splitting main problems in smaller ones and solve those individually. It requires high combinations of individual and group work engaged in very labor-intensive code revision and debugging. Online computer programming teaching comprises these aspects, which demands high levels of student-student and student-teacher interaction.

In this chapter we discuss and present our own experience in online teaching and

em desenvolvimento. O ensino online de programação de computadores é constituído por estes aspectos, exigindo um elevado grau de interação entre estudantes e entre estudantes e professor.

Neste capítulo, vamos discutir e apresentar a nossa experiência no ensino *online* da programação de computadores com base no modelo pedagógico virtual da Universidade Aberta, e propor a sua instanciação e extensão específica para incluir novas estratégias de aprendizagem colaborativa e uma abordagem construtivista para o processo global de aprendizagem.

Palavras-chave: Ensino *online* da programação; Ensino-aprendizagem online; Classe virtual; Modelo pedagógico virtual.

learning computer programming based on the pedagogic virtual model of Portuguese Open University while proposing its specific instantiation and extension to include new strategies of collaborative learning and a constructivist approach to the overall learning process.

Keywords: *Online computer programming teaching; Online teaching-learning; Virtual Classroom; Virtual Pedagogical Model.*

1 | INTRODUÇÃO

O ensino e aprendizagem a distância *online* (EAD) a nível universitário, está se tornando comum em todo o mundo. Por um lado, ele é uma consequência direta do rápido desenvolvimento da tecnologia da informação e comunicação (TIC), e por outro, pelo aumento da demanda na flexibilidade da relação espaço-tempo e da inovação tecnológica e pedagógica imposta pelos alunos (e potenciais alunos) às universidades, passando a oferta a ser mais baseada na web e no ensino a distância. Esta tendência tem levado as universidades, especialmente as universidades abertas, em busca de novos métodos de ensino e aprendizagem que possam explorar os meios tecnológicos para os limites que as TIC podem realmente oferecer.

O EAD é descrito por (Nichols, 2003) como “a educação que ocorre somente através da Web”, isto é, impostas não consiste de quaisquer materiais de aprendizagem físicos enviados para alunos ou contato presencial. A aprendizagem puramente *online* é, essencialmente, o uso de ferramentas de *eLearning* em uma modalidade de educação a distância utilizando a Web como o único meio para toda a aprendizagem do aluno e contato. “Embora esta afirmação ainda seja válida, a noção de EAD evoluiu para incluir aspectos como a aprendizagem colaborativa (Garrison, 2009), a participação *online* (Hrastinski, 2009), aprendizagem conectivista (Anderson, 2010) e, mais recentemente, cursos *online* massivos abertos (Yuan & Powell, 2013), para citar alguns.

Os objetos e artefatos 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 *online* (ver fig . 1).

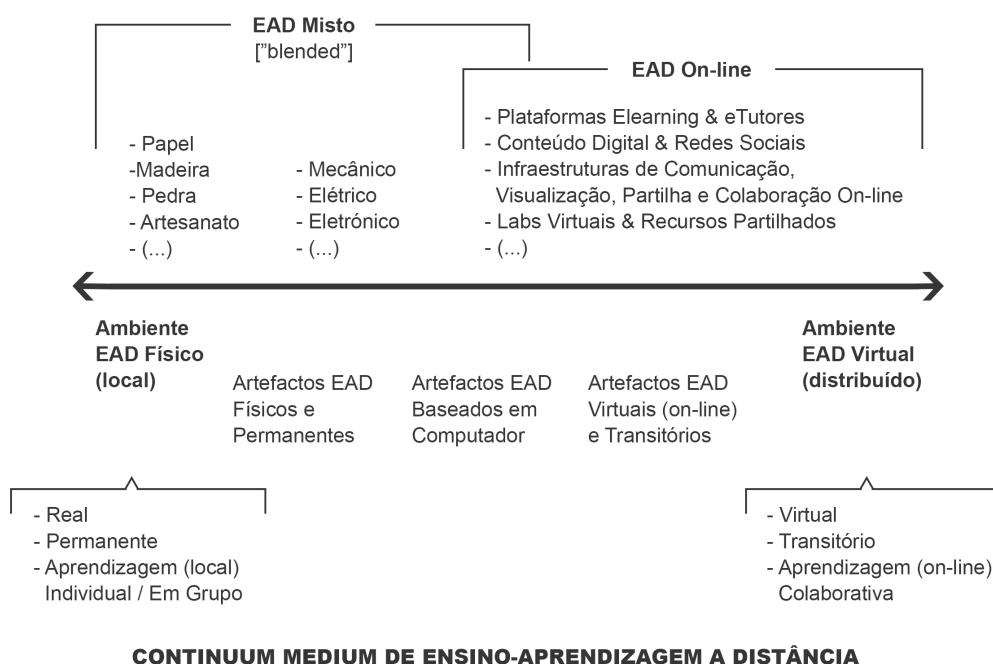


Figura 1 – O Continuum Medium de Ensino-Aprendizagem a Distância.

O EAD é muito frequentemente baseado nos princípios da aprendizagem centrada no aluno; flexibilidade (espacial e temporal) de aprendizagem; e interação *online*, 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 (Graham, 2005, Tinoca, 2010; Marcos & Coelho, 2012). Além disso, o EAD ou simplesmente o *eLearning*, é necessariamente baseado em um ambiente tecnológico de formação e, como tal, deve estar enraizado em estruturas epistemológicas (ver Fig. 2) 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 TIC, como a participação conjunta de especialistas de todo o mundo e estudantes nas mesmas sessões *online*, 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 a 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.

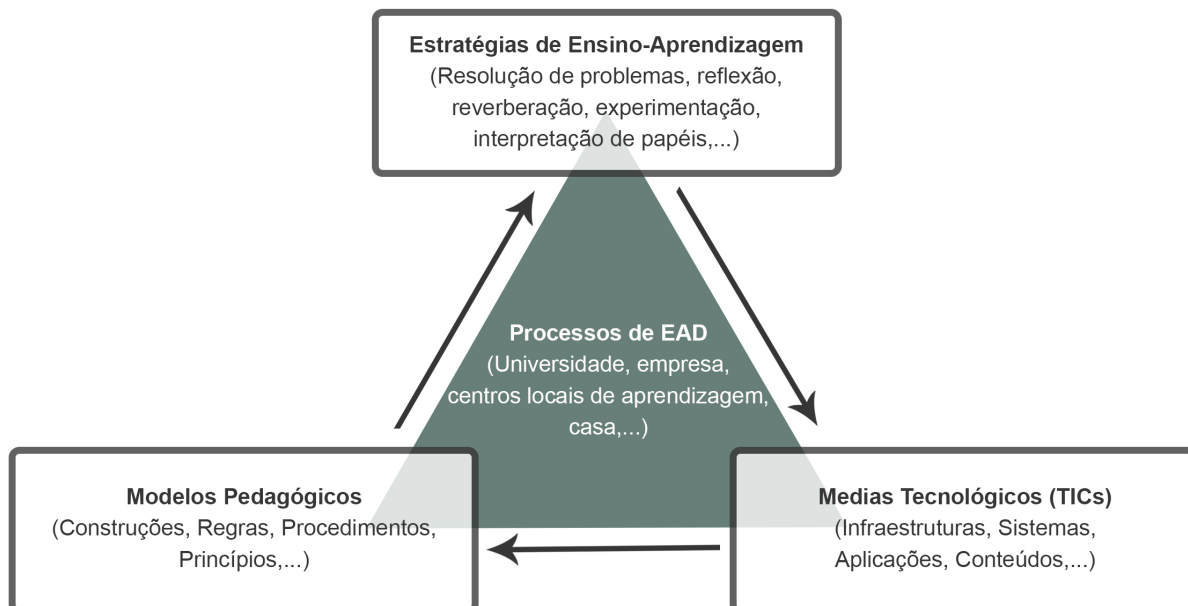


Figura 2 – Os três vetores do EAD (adaptado de Dabbagh, 2005).

Ao ensinar a programação de computadores, em particular, na forma EAD, enfrentamos frequentemente maiores dificuldades em promover a participação dos alunos 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, se comparado com outras áreas de ensino (por exemplo, humanidades), onde o trabalho está mais orientado para os trabalhos individuais. Os estudantes de programação de computadores enfrentam uma inibição natural na apresentação pública de suas dúvidas ou questões, pois as consideram ser de menor valor ou temem expor a sua ignorância sobre assuntos mais técnicos. A participação do aluno é muitas vezes mais do que a simples comunicação verbal de uma opinião, mas sobretudo o expor de uma questão técnica muito específica. A co-construção de programas, a partilha de informações e processos, e a orientação individual são atividades cruciais em EAD no ensino da programação

de computadores, exigindo elevados níveis de empenho dos estudantes e a colaboração no seio da comunidade de aprendizagem (Oncu & Cakir, 2011). Neste capítulo, discutimos e apresentamos uma experiência concreta de EAD ao nível da programação de computadores com base no modelo pedagógico virtual (MPV) da Universidade Aberta, enquanto propomos a sua instanciação e extensão específica para incluir novas estratégias de aprendizagem colaborativa e uma abordagem construtivista para o processo global da aprendizagem *online*.

O capítulo está organizado da seguinte forma: primeiro apresentamos uma introdução global, incluindo a motivação e a apresentação do modelo pedagógico virtual da Universidade Aberta. Em seguida, na secção 2 apresentamos o EAD aplicado ao ensino da programação de computadores e contextualizado no modelo pedagógico virtual, e na secção 3, fazemos algumas considerações sobre como poderíamos adaptar este mesmo modelo. Finalmente, na secção 4, são tecidas algumas conclusões mais relevantes.

1.1 | Motivação

A programação de computadores (ou codificação como hoje em dia é também denominada esta tarefa), exige, em primeiro lugar, o desenvolvimento de um adequado raciocínio lógico. Quando se tem um problema de programação em mãos, é preciso ser capaz de dividi-lo em pequenos blocos elementares, já que o código é desejavelmente construído de acordo com uma abordagem do tipo “dividir para conquistar”. Isto implica numa adequada capacidade de abstração, necessária para codificar um programa a partir da sua estrutura lógica mais elementar até à mais geral. Por exemplo, a partir de diagramas especificados em um pedaço de papel, deve ser possível criar código para um

determinado problema. Finalmente, existem várias linguagens de programação, que por sua vez, abarcam gramáticas e sintaxes nem sempre semelhantes, ou até mesmo ambientes de desenvolvimento (construção e execução) com características funcionais também bastante diversas. O mais problemático é que ao contrário de muitas outras áreas, cada problema requer, na maioria dos casos, a implementação de um novo programa, porque as soluções existentes dificilmente podem ser repetidas (apesar de se tentar reutilizar o código, tanto quanto possível). A paciência, a concentração, a persistência e a prática (intensiva) são características muito relevantes para desenvolver código.

Tudo isso mostra que a programação e, conseqüentemente, a sua aprendizagem impõe exigências muito específicas e diferentes da maioria das áreas do conhecimento. Não é comparável a aprendizagem de programação com a aprendizagem de Português, História, Física ou Matemática, por exemplo. Embora os dois últimos também requeiram raciocínio abstrato e lógico, as formulações são em geral as mesmas, e há diferenças em relação a programação. A Matemática acarreta uma lógica de codificação formal, que é também a programação de *software*. No entanto, a programação pode atingir um nível maior de abstração. Os programas podem ser desenvolvidos sem saber Matemática, mas para desenvolver graus mais sofisticados em programação (algoritmos de análise, desenvolvimento de motores, etc.) o conhecimento da matemática é essencial. Esta necessidade torna-se clara quando confrontados com problemas difíceis das ciências da computação, e quando mais baixo é o nível de abstração necessário.

Neste contexto torna-se claro que a aprendizagem de programação exige estratégias pedagógicas diferenciadas em seu acompanhamento e apoio. Os desafios adicionais no ensino da programação indicam remotamente uma

clara necessidade de adaptar o modelo de ensino atual. Novas estratégias devem ser incluídas e suposições atuais revistas e possivelmente adaptadas para este tipo de educação se tornar realmente um sucesso. Os resultados das interações dos alunos nas disciplinas de programação servem como um indicador claro desta necessidade.

Este capítulo vai abordar esta questão tão importante e relevante na educação no universo do ensino a distância, propondo uma instanciação do modelo pedagógico virtual de Universidade Aberta para o ensino-aprendizagem de programação utilizando estratégias de aprendizagem colaborativa e uma abordagem construtivista para o processo de aprendizado.

1.2 | O Modelo Pedagógico Virtual da Universidade Aberta e o caso do EaD das Ciências da Computação

O Modelo Pedagógico Virtual 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 EAD para os cursos de terceiro ciclo. O VPM 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 VPM promove políticas de interação entre alunos, através da aprendizagem colaborativa e é focado na avaliação contínua, baseando-se em quatro grandes princípios, a saber:

- 1) O princípio da aprendizagem centrada no estudante, onde os alunos são considerados indivíduos ativos, construtores do seu próprio conhecimento, dirigindo seu processo de aprendizagem como parte integrante de uma comunidade de aprendizagem;

- 2) O princípio da flexibilidade, onde os alunos podem aprender em qualquer lugar e em qualquer momento, independentemente das limitações de espaço-tempo que, em contrapartida, impõe ensino em sala de aula;
- 3) O princípio da interação *online*, que se estende para o novo tipo de interação aluno-aluno que ocorre em grupos de discussão assíncronas dentro de cada classe virtual e é a base para a aprendizagem colaborativa;
- 4) O princípio da inclusão digital, onde a instituição de ensino é o agente de formação e transmissão de competências básicas em TIC para aqueles que se encontram excluídos deste tipo de conhecimento e, portanto, incapazes de frequentar o ensino superior no modo presencial;

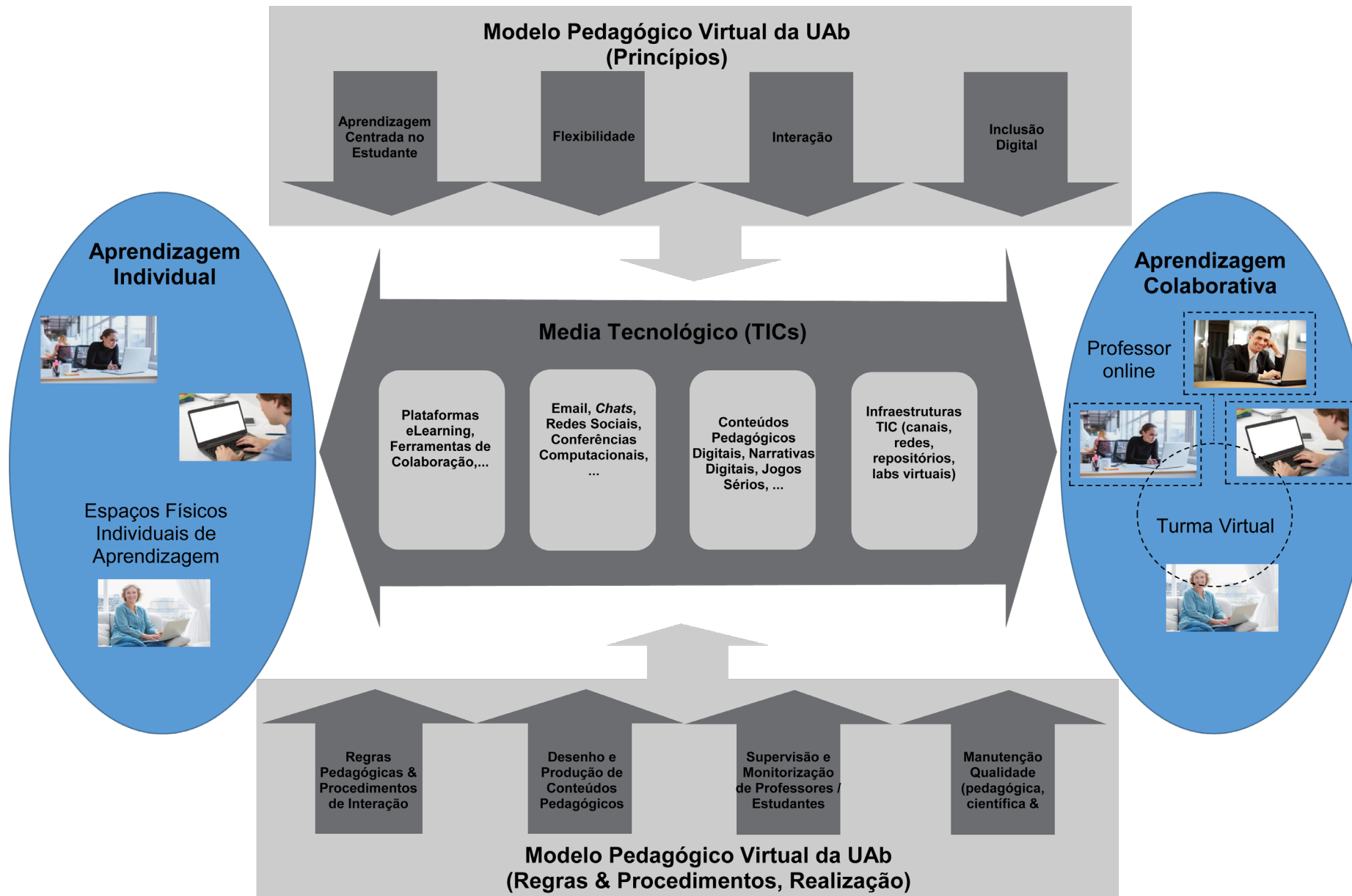


Figura 3 – Visão geral do Modelo Pedagógico Virtual da UAb.

O VPM fornece um modelo de referência para o planeamento completo (ver figura 3), organização e implementação de cursos *online* de nível universitário, explorando plenamente as facilidades oferecidas pelas TIC. Ele especifica que a aprendizagem colaborativa tem lugar no contexto das aulas virtuais com um número máximo de alunos (60, 30 de graduação e pós-graduação, respectivamente), enquanto a aprendizagem individual ocorre em lugares distribuídos geograficamente. As figuras 4 e 5 ilustram o VPM em ação.

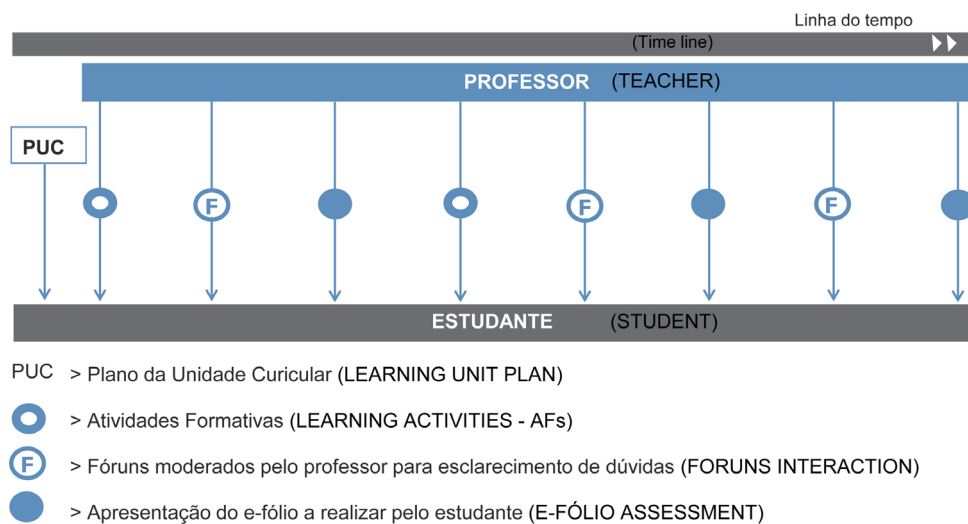


Figure 4 – Visibilidade do professor durante o semestre.

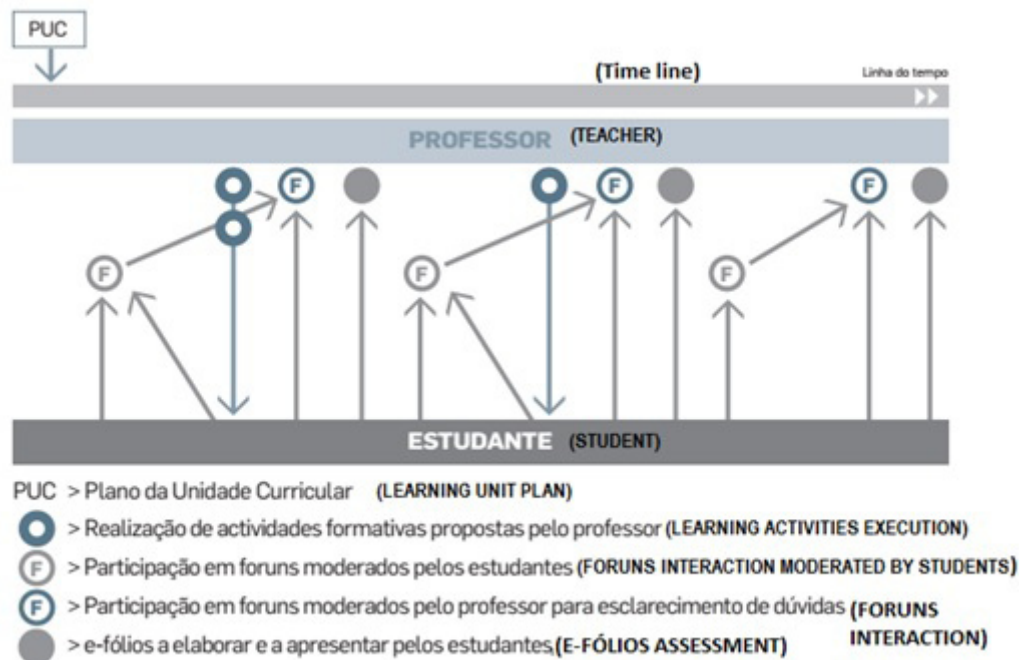


Figure 5 – Potencial percurso de aprendizagem do estudante durante o semestre.

A aplicação do VPM aos processos de ensino / aprendizagem no campo da ciência da computação, em geral, além de exigir elementos das TICs adequados, exigem também que alguns recursos especializados de informática (por exemplo, *clusters*, computadores de alto desempenho, etc.) ou processos de computação (por exemplo, máquinas virtuais, simulação específica) sejam implementados e disponibilizados aos alunos. Além disso, devido às características altamente especializadas das diferentes subáreas da ciência da computação, há uma necessidade de aprendizagem colaborativa mais intensiva para a qual são requeridos espaços virtuais compartilhados onde os alunos podem programar ou testar seus sistemas e aplicativos de computador com a participação de seus colegas. Este espaço compartilhado pode explorar tecnologias inteligentes adaptativas tais como *wikis*, redes sociais focadas ou agentes virtuais inteligentes, entre outros.

Quando os trabalhos estão prontos para demonstração ou exibição, eles têm de ser apresentados em um espaço virtual para torná-los visíveis para toda a classe virtual ou para outros públicos específicos.

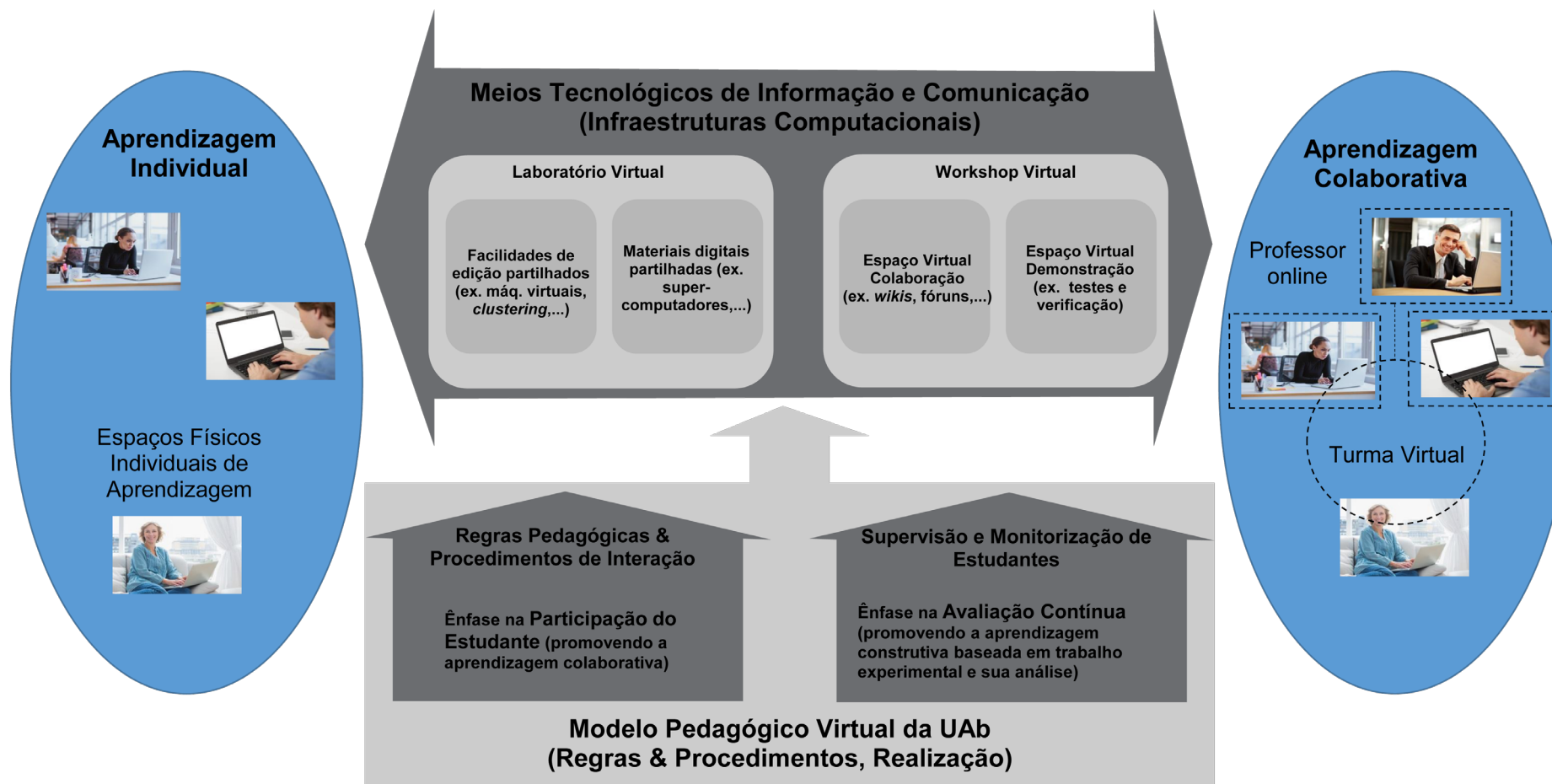


Figura 6 – Visão global da adaptação do MPV da UAb ao EaD da ciência da computação.

Os cenários EAD em ciência da computação implicam numa ligeira adaptação das regras, procedimentos e mecanismos de implementação do VPM para enfatizar (e promover) o nível de interação entre os alunos e entre os alunos e o professor, que é essencial para a aprendizagem colaborativa acontecer. Além disso, a supervisão e monitorização dos trabalhos do aluno são especialmente focadas na aprendizagem e avaliação contínuas, pois algumas subáreas da ciência da computação, como a programação de computadores, não são adequadas a apenas uma avaliação escrita. Nestes casos, os alunos devem aprender fazendo o trabalho e mostrando o que aprenderam ao demonstrá-lo.

Na Figura 6, está retratada a adaptação do VPM ao caso específico da ciência da computação. Além dos outros módulos, dividimos a parcela respectiva as TICs em laboratórios e workshops virtuais. Na primeira, os alunos podem encontrar todas as facilidades e os recursos necessários para executar tarefas que exigem recursos computacionais específicos. Este trabalho é geralmente do tipo autónomo. Quando os alunos se envolvem em trabalho colaborativo, através da partilha de problemas ou ao completar tarefas em parceria, eles utilizam os recursos disponibilizados nas Workshops virtuais, incluindo a demonstração e apresentação dos resultados em um espaço virtual. Os laboratórios e workshops virtuais serão apresentados em detalhe mais a frente neste capítulo.

2 | O EaD DA PROGRAMAÇÃO DE COMPUTADORES NO MODELO PEDAGÓGICO VIRTUAL DA UAb

2.1 | Fundamentação

De acordo com Wang (2011), os alunos, especialmente os novatos, podem precisar de orientação constante. Em um ambiente *online*, os alunos não têm o benefício de obter um comentário imediato dos instrutores. Além disso, um aspecto muito importante para aprender a programar é ver como os outros alunos abordam o mesmo problema, e aprender a trabalhar em conjunto através de projetos em grupo. Em um ambiente *online*, a colaboração entre os alunos não é fácil. Algumas das sugestões apresentadas pelo autor são: 1) a criação de um laboratório de informática virtual; 2) adicionar mais materiais multimédia no curso e 3) a criação de um sentido de comunidade entre os alunos.

Milne & Roe (2002) realizaram questionários sobre os vários conceitos e tópicos de programação orientada a objetos que os estudantes acham mais difíceis em lidar nos cursos introdutórios. A análise estatística dos resultados mostrou que os temas que dependem de uma compreensão clara de ponteiros e conceitos relacionados à memória (tais como construtores de cópia e funções virtuais) são os mais difíceis. Eles acharam que esses conceitos são difíceis por causa da incapacidade do aluno para compreender o que está acontecendo com o seu programa a nível de memória, pois não conseguem criar um modelo mental claro da sua execução. Estes resultados sugerem que uma abordagem mais clara para ensinar esses tópicos seria benéfica para os alunos, como uma abordagem orientada à visualização.

Os MOOCs (*Massive Open Online Courses*) ganharam recentemente popularidade em várias universidades e até mesmo em sociedades globais. Um fator crítico para o seu sucesso e eficácia no ensino e aprendizagem é atribuição de classificação. As formas tradicionais de atribuição de classificação

não são escaláveis e não fornecem um *feedback* em tempo útil ou de forma interativa para estudantes. Para resolver estas questões, Tillman et al. (2013) desenvolveram uma plataforma de jogos interativos para *eLearning* designada de Pex4Fun (ver figura 7). O Pex4Fun é um ambiente para ensino e aprendizagem da programação de *software* tanto a nível introdutório como avançado. O Pex4Fun é um bom exemplo de como os jogos sérios podem auxiliar o ensino da programação.

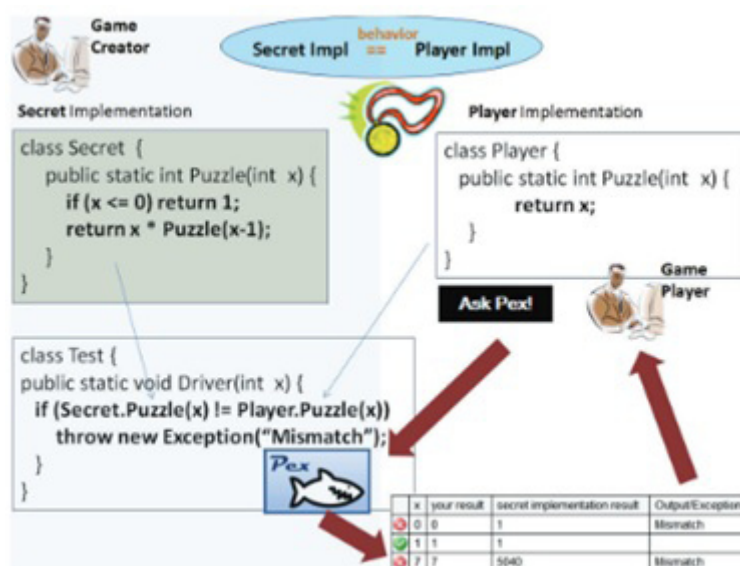


Figura 7 – Pex4Fun (Tillman et al. 2013)

2.2 | EaD da Programação de Computadores

A disciplina de programação orientada a objetos (POO) visa proporcionar aos alunos conhecimentos e práticas relativas aos princípios, conceitos principais, modelos e técnicas relacionadas com programação de computadores baseado no paradigma orientado a objetos. O conteúdo programático da disciplina inclui a análise do paradigma de programação por objetos, algoritmos e blocos de código, como também várias técnicas, visando o ensino da correta forma de resolver problemas na programação orientada a objetos, enquanto os alunos

são desafiados a projetar e implementar novas abordagens ou melhorias das já existentes. A linguagem de programação e ambiente adoptados são o C ++ e *Eclipse IDE*, respectivamente.

Ela assume uma carga horária total de 156 horas, sendo 26 horas de contacto. A avaliação dos alunos é feita através de dois documentos digitais escrito (chamados e-fólios) durante o semestre e uma avaliação em sala de aula (chamado p-fólio), no final do semestre. Os e-folios contribuem com 40% para a nota final, enquanto o p-fólio, 60%. Se eles falharem, eles têm uma última oportunidade de serem aprovados com um exame de recurso. Os e-fólios dentro do contexto desta disciplina são projetos em C ++ que são implementados para um determinado cenário-problema.

O ambiente *Moodle* da disciplina é composto por sete tópicos, em que o aluno enfrenta um crescente grau de complexidade, e é convidado a executar sempre uma atividade de aprendizagem (que não computa para a nota final e não é obrigatória, designada de AF). Cada tópico tem a duração de 2 semanas (excepto o primeiro) e inclui material didático (com vastos exemplos de códigos e links relevantes, e até mesmo vídeos) especialmente desenvolvidos para os alunos, além da solução das atividades de aprendizagem e e-fólios. Todos os temas têm um fórum de discussão onde o apoio ao conteúdo do tópico é garantido pelo professor de forma assíncrona. O professor sempre responde as perguntas no máximo em 48 horas (fim de semana) ou em base diária (de segunda a sexta-feira), além de estimular os alunos a participar. A tabela 1 mostra em pormenor a forma como a disciplina está organizada:

Tabela 1 – Organização da disciplina

Tópico	Conteúdo	Objetivos	Duração
1	Introdução à programação orientada a objetos (POO)	Familiarização com o conceito de programação e qualidade do software orientado a objetos; instalação do Eclipse IDE e sua configuração.	1 semana
2	Recursos de Programação não relacionados especificamente com POO	Familiarização com as funcionalidades em C++ não diretamente relacionadas com a POO, herdadas em larga medida da linguagem C; implementação de um programa em C++ recorrendo a funções e outros recursos não diretamente relacionados com POO.	2 semanas
3	Definição e criação de classes e seus atributos	Familiarização com o conceito de classe, objeto e os seus atributos; uso de construtores, destrutores, controlo de acessos, classes amigas e interligadas	2 semanas
4	Sobrecarga, conversão e matrizes de objetos	Sobrecarga de funções membro e operadores; conversão entre objetos e tipos simples; matrizes e listas de objetos	2 semanas
5	Herança simples	Criação de classes e métodos com mecanismos individuais de herança simples	2 semanas
6	Herança múltipla e polimorfismo	Criação de classes com mecanismos de herança múltipla; classes virtuais	2 semanas
7	Bibliotecas para gestão de fluxo de dados, templates e tratamento de exceções	Utilização e criação de templates tratamento de exceções, uso de bibliotecas para gestão de fluxo de dados (entrada e saída)	2 semanas

No final dos tópicos 4 e 6, os e-fólios A e B, respectivamente, são propostos para os alunos. O p-fólio é realizado após o final do tópico 7. O exame de recurso tem lugar antes do início do semestre subsequente (geralmente, em setembro, porque o semestre é de março a julho). Todas as AFs e e-fólios são resolvidos pelo professor e colocados à disposição dos alunos após os prazos definidos para cada um deles.

2.3 | Limitações e necessidades

Em geral, com base nos resultados do nível de interação dos alunos que foram detetados, podemos assumir que a sua qualidade é inferior ao que era esperado ou suposto ser o ideal. Isso indica que, possivelmente, o modelo de *eLearning* não está bem adaptado para trabalhar com disciplinas em que linguagens de programação são ensinadas e outras estratégias didáticas-pedagógicas, têm de ser acrescentadas nestes casos. Embora isso seja importante, devemos também levar em linha de conta que a ensino *online* tem levantado algumas preocupações sobre as taxas elevadas de desistência de estudantes em cursos e programas *online*. Em geral, os cursos e programas *online* têm taxas mais elevadas de insucesso em relação aos cursos presenciais (Bowers & Kumar, 2015).

Uma conclusão muito importante é a subutilização significativa dos fóruns por todos os alunos, e principalmente, a falta de provas de que a sua subutilização ajuda a conduzir ao insucesso escolar. Algumas das limitações podem ser causadas pelo fato de que a maioria dos alunos serem trabalhadores-estudantes e com idade madura, com família constituída, tendo assim tempo limitado (responsabilidades familiares e profissionais) para se dedicar ao estudo. Outro fator importante é a especificidade resultante das exigências da programação na aprendizagem. De acordo com Nandi et al. (2012) o conteúdo para o curso de programação é mais prescritivo, algorítmico, e com um foco mais estreito; portanto, as oportunidades para a discussão direta e perguntas são limitadas. Muitas vezes, uma única solução divulgada por um estudante em resposta a um problema levantado por um outro estudante ou o instrutor termina a discussão nesse ponto. A mesma situação se aplica para os exercícios. Uma vez que a solução é obtida, há pouca variedade de soluções a serem fornecidas. Isto

pode explicar a baixa atividade nos fóruns.

A introdução de reuniões regulares síncronas ou o desenvolvimento de conteúdos multimédia para ensinar de forma interativa como programar podem constituir formas futuras a explorar (utilizando realidade virtual e aumentada, por exemplo) no sentido de reforçar a ligação do estudante ao processo de ensino-aprendizagem em curso. Neste ponto torna-se clara a necessidade premente que se investigue em detalhe e profundidade as limitações e inconvenientes que estão na origem do abandono ou desinteresse por parte dos alunos ou mesmo as razões que levam aos reduzidos níveis da qualidade da própria interação no ambiente de ensino a distância, no caso de POO. Também um trabalho futuro de investigação deverá abarcar a comparação destes resultados com os obtidos a partir de outras disciplinas de programação lecionadas no âmbito de outra oferta da UAb na área da Informática.

3 | ADAPTAÇÃO DO MODELO PEDAGÓGICO VIRTUAL DA UAb

3.1 | Avaliação Flexível

No sentido de aumentar a flexibilidade do EAD proporcionar aos alunos mais controlo sobre o processo de avaliação representa também uma abordagem a ser seguida. Embora possa haver uma resistência considerável (Irwin & Hepplestone, 2012) em assegurar aos estudantes um maior “controlo” das suas avaliações, de facto o desenvolvimento de aprendentes com elevada autonomia sobre o seu próprio percurso de aprendizagem representa o reflexo de um ambiente de ensino universitário amadurecido e constitui em si um objetivo global de todo e qualquer sistema de EAD. Pacharna, Baya & FeltonaThey (2013) descrevem o desenvolvimento de uma abordagem que permitiu aos alunos determinar os pesos atribuídos a cada componente do curso e realocar os pesos em resposta às pontuações alcançadas. Essa abordagem é comparada com outras mais tradicionais, sendo particularmente demonstrado que oferecer aos estudantes mais controlo sobre suas avaliações pode influenciar a sua motivação e a obtenção de melhores notas. Seus resultados mostram que a adoção de uma abordagem deste tipo no início do semestre letivo teve pouco impacto, mas que um sistema flexível que permita uma realocação dos pesos acarreta um impacto positivo sobre as notas em geral e, como os próprios alunos relataram, impacta também sobre a motivação e as atitudes mais proactivas.

No caso da disciplina de POO seria utilizada a mesma estratégia. As AFs seriam também utilizadas para calcular a pontuação final do aluno, no caso de insucesso nos e-fólios, por exemplo. Cada AF teria um peso diferente que o aluno atribuiria em conformidade.

3.2 | Laboratórios Virtuais e Workshops

De forma a ultrapassar algumas das limitações que o EAD da programação enfrenta, também podemos considerar a inclusão de novas abordagens tecnológicas, tais como os laboratórios virtuais. A tecnologia computacional e a Internet têm o potencial de fornecer um ambiente de aprendizagem altamente interativo e poderoso para as disciplinas de engenharia. Nas aplicações laboratoriais, do ponto de vista técnico, todos os problemas de engenharia lidam com alguma quantidade física. Um computador equipado com circuitos e interface apropriados, além de *software* e sistemas para a aquisição de dados, podem criar uma representação gráfica visual para estas quantidades e processar os dados adquiridos. Essas experiências podem ser disponibilizadas para pontos remotos onde o aluno se localiza. O mesmo pode ser estendido para o processo de codificação.

Por exemplo, um laboratório de química virtual (Dalgarno, Bishop & Bedgood, 2012) foi desenvolvido na Universidade Charles Sturt, com base em um modelo 3D preciso do laboratório de ensino de graduação em Wagga Wagga. A versão inicial do laboratório virtual foi projetado para permitir que os estudantes de ensino à distância em química se familiarizassem com o laboratório antes do seu período em residência. Ele permite a exploração livre e a coleta e agrupamento de componentes dos aparelhos. Ele também permite que os alunos leiam informações sobre os componentes dos aparelhos e os procedimentos laboratoriais.

Para aprender e desenvolver o código, o aluno deve utilizar algum ambiente em desenvolvimento (como o *Eclipse IDE* ou *Visual Studio*). Esses ambientes basicamente compilam o código, gerando um ficheiro executável. Se há algum erro de sintaxe, o compilador indica isso, mas se o erro acontece

a um nível semântico, o executável não será executado e apenas uma reduzida explicação é normalmente enviada para o programador. Mesmo se considerarmos as advertências que o compilador apresenta, muitas delas não são muito claras. Isto significa que um outro nível de assistência deverá ser concedido ao estudante, ou em outras palavras, o laboratório de programação deve funcionar tanto como um compilador, mas incluir a análise semântica do código, além de sua sintaxe dever ser muito bem explicada. Tal compilador deve ser criado e desenhado com foco em exercícios específicos, que por sua vez, são definidos para cobrir um âmbito limitado do processo de aprendizagem de programação. Assim, como o aluno sozinho ou em pequenos grupos necessita da assistência próxima do professor *online* (remoto), as atividades laboratoriais tornam-se parte de um Workshop Virtual *online* organizada com base em um conjunto de objetivos de ensino / aprendizagem e em torno de uma discussão *online* síncrona ou assíncrona, que pode também incluir a demonstração e apresentação dos resultados. Tais processos colaborativos são fortemente monitorizados, explorando papéis sociais em que o professor assume o controlo principal.

3.3 | Controlo e Monitorização finos

Outra ferramenta útil que pode ajudar a dar um melhor suporte ao estudante é a existência de formas avançadas para monitorizar o comportamento e interação do aluno com a plataforma e, por consequência, também a interação entre pares e com o professor. Se soubermos melhor como os alunos utilizam os recursos seremos também capazes de facultar melhores estratégias para a aprendizagem. Em termos de monitorização, o desenvolvimento de um *dashboard* ou painel de controlo fino, introduz um nível mais elevado e fino de

controle e monitorização.

Murray et al. (2013) implementaram um protótipo de *dashboard* facilitador que fornece a terceiros uma “visão panorâmica” do estado e do fluxo das interações *online*. Eles o têm utilizado sob o ponto de vista do professor bem como uma “ferramenta para conscientizar” os participantes da plataforma. Eles construíram uma API (veja a figura 8) que permite que o *dashboard* receba atualizações em tempo real sobre o estado dos diálogos e textos postados a partir do sistema *Mediem* de fóruns de diálogo. Para além disso, eles fizeram as primeiras implementações para identificar automaticamente as fases do diálogo (por exemplo, apresentações, deliberação, impasses, persuasão) e pontos de oportunidade para intervenções (por exemplo, silêncios ou ausência de reação, mudanças de fase ou tom, tensões emocionais súbitas de múltiplos participantes).

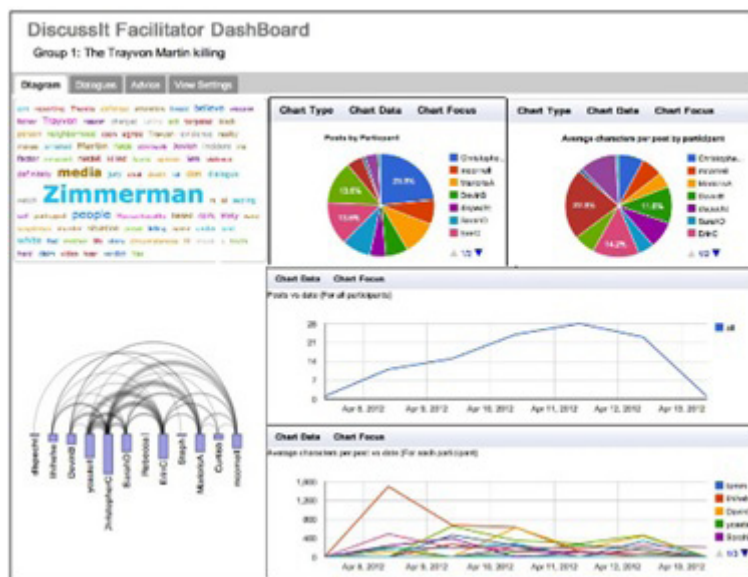


Figura 8 – Dashboard (Murray et al., 2013)

Com base no que foi discutido, a figura 9 mostra o Modelo Pedagógico Virtual da UAb adaptado para o EAD em programação de computadores.

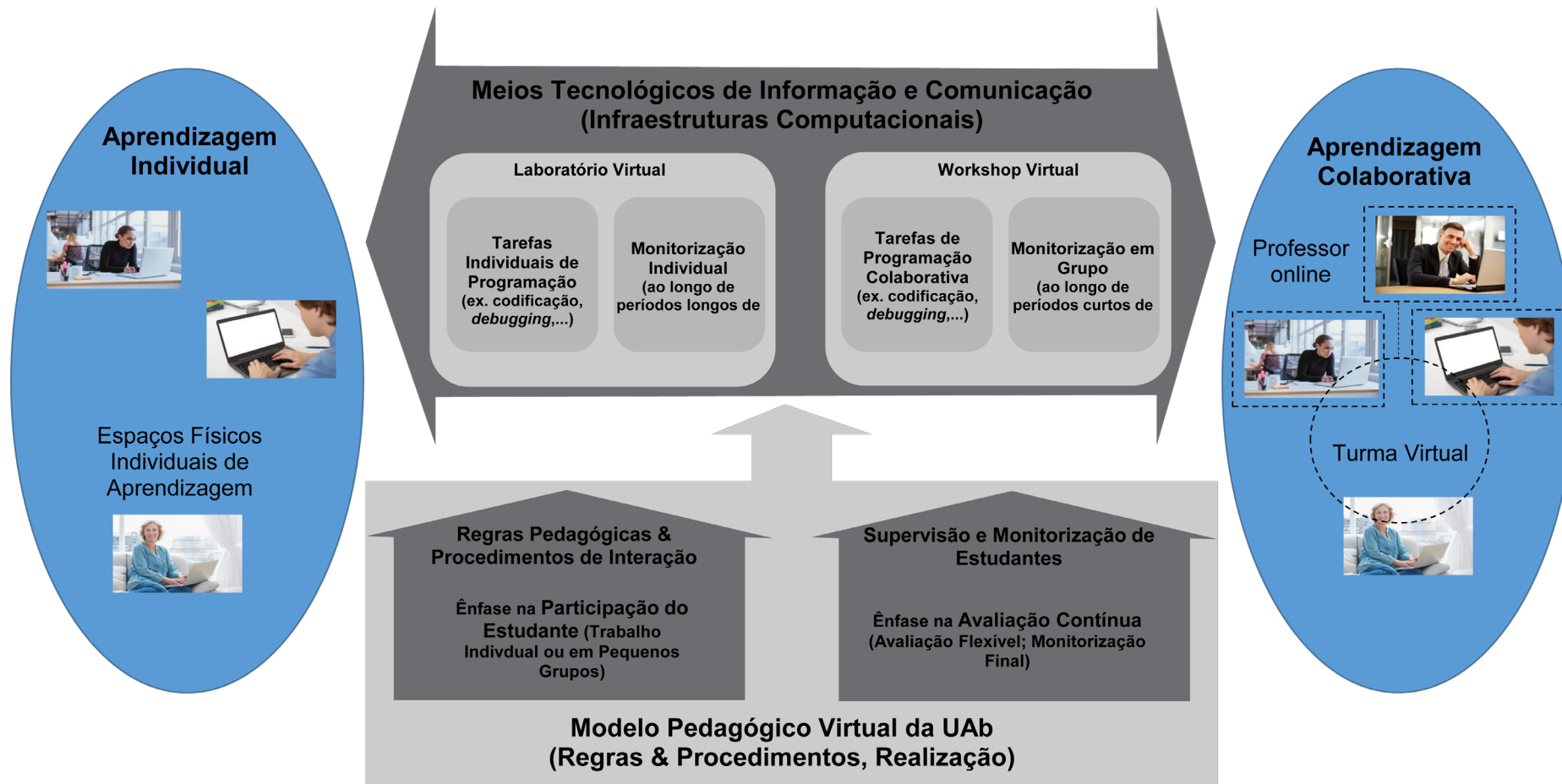


Figura 9 – O MPV da UAb adaptado ao EaD de programação de computadores.

4 | CONCLUSÕES

Para ensinar como desenvolver e implementar o código de um programa de computador em um ambiente baseado completamente em *eLearning* comporta ainda uma série de desafios e limitações. Da nossa experiência em EAD de programação de computadores detetamos 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. Precisamos adaptar o VPM da UAb, a fim de atender adequadamente a essas demandas e necessidades específicas no sentido de um aprendizado apropriado da programação de computadores. Avaliações mais flexíveis e laboratórios virtuais e workshops podem ser algumas das novas estratégias a considerar, além de outros. Um melhor controlo e, portanto, o conhecimento de como os alunos utilizam os recursos e se interrelacionam, entre si e com o professor, constitui um contributo adicional para desenvolver soluções mais adequadas ao EAD da programação de computadores.

5 | BIBLIOGRAFIA

- Bowers, J. & Kumar, P. (2015). Students' Perceptions of Teaching and Social Presence: A Comparative Analysis of Face-to-Face and Online Learning Environments. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, 10(1), 27-44.
- Dabbagh, N. (2005). Pedagogical models for E-Learning: A theory-based design framework. *International Journal of Technology in Teaching and Learning*, 1(1), 25-44.
- Dalgarno, B. et al. (2012, November). The potential of virtual laboratories for distance education science teaching: reflections from the development and evaluation of a virtual chemistry laboratory. *Proceedings of The Australian Conference on Science and Mathematics Education (formerly UniServe Science Conference)* (Vol. 9).
- Marcos, A. & Coelho, J. (2012). Estratégias pedagógicas para o ensino-aprendizagem da expressão gráfica digital no ensino a distância online. In Catarina S. Martins; Manuela Terrasêca & Vítor Martins (ed.). *À procura de renovações de estratégias e narrativas sobre educação artística* (pp. 165-188). Edição da GESTO Cooperativa Cultural, CRL, 1ª edição, abril 2012, ISBN: 978-989-20-3000-5.<<http://hdl.handle.net/10400.2/2105>>. Indexação: Biblioteca Nacional.
- Milne, I. & Rowe, G. (2002). Difficulties in learning and teaching programming-views of students and tutors. *Education and Information technologies*, 7(1), 55-66.
- Murray, T. et al. (2013). A Prototype Facilitators Dashboard: Assessing and visualizing dialogue quality in online deliberations for education and work.

Submitted to 2013 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government.

- Nandi, D. et al. (2012). Evaluating the quality of interaction in asynchronous discussion forums in fully online courses. *Distance education*, 33(1), 5-30.
- Oncu, S. & Cakir, H. (2011). Research in Online Learning Environments: Priorities and Methodologies. *Computers & Education*, DOI: 10.1016/j.compendu.2010.12.009
- Tillmann, N. et al. (2013, May). Teaching and learning programming and software engineering via interactive gaming. In *Software Engineering (ICSE), 2013 35th International Conference on* (pp. 1117-1126). IEEE.
- Wang, W. (2011). Teaching programming online. In International conference on the future of education. <http://www.pixel-online.net/edu_future/common/download/Paper_pdf/ELE19-Wang.pdf>.
- Irwin, B. & Hepplestone, S. (2012). Examining increased flexibility in assessment formats. *Assessment & Evaluation in Higher Education*, 37(7), 773-785.
- Pacharn, P. et al. (2013). The Impact of a Flexible Assessment System on Students' Motivation, Performance and Attitude. *Accounting Education*, 22(2), 147-167.
- Tinoca, L. et al. (2010). Online group work patterns: how to promote a successful collaboration. In Dirckinck-Holmfeld L. et al. (ed.). *Proceedings of the Seventh International Conference on Networked Learning 2010*. Danmark: Aalborg, April 2010.