



**Paulo Jorge  
Nascimento Oliveira**

**Sistema de Educação Online para  
Desenvolvimento de Software**

**Online Education System for Software  
Development**





**Paulo Jorge  
Nascimento Oliveira**

**Sistema de Educação Online para  
Desenvolvimento de Software**

**Online Education System for Software  
Development**

*“Education is a path to prosperity, better economics, jobs, and frankly, world peace. (...) An educated world is simply a better world.”*

— Anant Agarwal, CEO of MOOC nonprofit edX





**Paulo Jorge  
Nascimento Oliveira**

**Sistema de Educação Online para  
Desenvolvimento de Software**

**Online Education System for Software  
Development**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Sistemas de Informação, realizada sob a orientação científica do Doutor Diogo Nuno Pereira Gomes, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor João Paulo Silva Baraca, Professor auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.



**o júri / the jury**

presidente / president

**Prof. Doutor Luís Filipe de Seabra Lopes**

Professor Associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

vogais / examiners committee

**Prof. Doutor Carlos Manuel das Neves Santos**

Professor Auxiliar do Departamento de Comunicação e Arte da Universidade de Aveiro

**Prof. Doutor Diogo Nuno Pereira Gomes**

Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro





**agradecimentos /  
acknowledgements**

Agradeço aos orientadores por estarem disponíveis para aceitar a proposta de dissertação e por todo o apoio durante o decorrer da mesma.

Agradeço aos membros do grupo de investigação ATNoG (Aveiro Telecommunications and Networks Group), pela ajuda prestada ao longo da dissertação, contributo esse que ajudou a ultrapassar os vários desafios que ocorreram durante a mesma.

Por fim, quero agradecer aos amigos, aos pais e à namorada pelo apoio e paciência que contribuíram para ultrapassar mais uma etapa da vida.



## Palavras Chave

E-Learning, IDE, Programação, Web, Cloud, SOA.

## Resumo

A tecnologia teve um papel fundamental na evolução dos processos de ensino, nomeadamente o uso de formatos multimédia que permitem aperfeiçoar a exposição de conteúdo e a troca de informação como resultado deu-se a proliferação de alguns sistemas de *E-Learning*.

A adoção destes sistemas de *E-Learning* surgem de necessidades ou preocupações, como por exemplo, ensino à distância, partilha de documentos, métodos de avaliação autónoma entre outras funcionalidades, das quais foram parcialmente resolvidas com o uso da *Internet* e de outras tecnologias.

Atualmente, sistemas de referência, como o caso do MOODLE, não se focalizam numa área específica da educação e as ferramentas que disponibilizam permitem mitigar as necessidades anteriormente mencionadas. Mas em áreas com necessidades mais específicas como é o exemplo da informática, mais precisamente o ensino de competências de programação, estas funcionalidades são um suporte limitado aos docentes que as usam. Recentemente, têm sido feito esforços, como por exemplo, o desenvolvimento de plataformas para competições de programação ou sistemas autónomos de avaliação de código.

Nesta dissertação propõe-se uma nova solução para o ensino de programação desenvolvida em tecnologias *web*. A solução está dividida em duas partes, um formato interativo para a transmissão de conhecimento teórico-prático e um ambiente partilhado para o desenvolvimento de trabalhos de grupo.



**Keywords**

E-Learning, IDE, Programming, Web, Cloud, SOA.

**Abstract**

Technology was essential to the evolution of learning methods more precisely the use of multimedia formats which allow the improvement of content exposure and the exchange of information as a result some of the E-Learning systems proliferated.

The adoption of these E-Learning systems arise from needs or concerns, such as distance learning, sharing documents, autonomous evaluation among other functionalities and were solved with the Internet and other technologies.

Nowadays, some acknowledged systems, such as MOODLE, do not focus on a specific area of education and his tools mitigate the needs mention above. But in areas with specific needs such as computer science, like teaching programming skills, these features are offer a limited support to teachers who use them. Recently, efforts have been made to develop platforms for programming competitions or autonomous evaluation systems.

On this dissertation a new solution is proposed to be used to teach programming and is being developed based on Web technologies. This solution is made into two parts, the first is a new format to display interactive content and the second is a shared environment for group work.



# CONTEÚDO

---

CONTEÚDO . . . . .	i
LISTA DE FIGURAS . . . . .	iii
LISTA DE TABELAS . . . . .	v
GLOSSÁRIO . . . . .	vii
1 INTRODUÇÃO . . . . .	1
1.1 Motivação . . . . .	1
1.2 Objetivos e Desafios . . . . .	2
1.3 Requisitos . . . . .	3
1.4 Organização do Documento . . . . .	4
2 ESTADO DA ARTE . . . . .	5
2.1 Enquadramento . . . . .	5
2.2 Open Education . . . . .	7
2.3 E-Learning . . . . .	9
2.4 E-Learning 2.0 . . . . .	14
2.5 Serviços Online . . . . .	14
2.5.1 edX . . . . .	15
2.5.2 Udemy . . . . .	16
2.5.3 Codecademy . . . . .	17
2.5.4 CS50 . . . . .	18
2.5.5 Codio . . . . .	22
2.5.6 Nixty . . . . .	23
2.6 Soluções Self-hosted . . . . .	25
2.6.1 Moodle . . . . .	25
2.6.2 SAAL . . . . .	27
2.6.3 PBeL . . . . .	29
2.6.4 PyCharm Edu . . . . .	33
2.7 Ferramentas . . . . .	34
2.7.1 Ambientes de Desenvolvimento . . . . .	35
2.7.2 Ambientes de Aprendizagem . . . . .	45
2.8 Conclusão . . . . .	47
3 NOVA SOLUÇÃO DE ENSINO À DISTÂNCIA . . . . .	49

3.1	Solução Conceptual . . . . .	49
3.2	Discussão . . . . .	53
3.3	Arquitetura Proposta . . . . .	55
3.3.1	Utilizador . . . . .	56
3.3.2	Aplicação Web . . . . .	56
3.3.3	Plataforma de Cloud . . . . .	57
3.3.4	Base de Dados . . . . .	57
3.3.5	Serviço de Identidade . . . . .	58
3.3.6	Broker . . . . .	58
3.4	Conclusão . . . . .	59
4	IMPLEMENTAÇÃO . . . . .	61
4.1	Componentes . . . . .	61
4.1.1	Aplicação Web . . . . .	63
4.1.2	Base de Dados . . . . .	64
4.1.3	Camada de Abstração de Autenticação . . . . .	65
4.1.4	Camada de Abstração da Cloud . . . . .	70
4.1.5	Broker . . . . .	72
4.1.6	Serviço de Cloud . . . . .	73
4.2	Provisão . . . . .	74
4.3	Demonstração . . . . .	76
4.3.1	Autenticação . . . . .	76
4.3.2	Aceder ao Ambiente de Aprendizagem . . . . .	80
4.3.3	Aceder ao Ambiente de Desenvolvimento . . . . .	83
4.3.4	Submeter Ambiente de Aprendizagem . . . . .	91
4.4	Conclusão . . . . .	93
5	RESULTADOS . . . . .	95
5.1	Experiência . . . . .	95
5.2	Duração . . . . .	97
5.3	Processamento . . . . .	99
5.4	Memória . . . . .	101
5.5	Armazenamento . . . . .	103
5.6	Manutenção . . . . .	104
6	CONCLUSÃO . . . . .	105
6.1	Contribuições . . . . .	106
6.2	Trabalho Futuro . . . . .	106
	REFERÊNCIAS . . . . .	109



# LISTA DE FIGURAS

---

2.1	As dimensões do <i>Open Education</i> [19]	8
2.2	Comparação entre dispositivos móveis <i>versus</i> a população mundial[30]	11
2.3	Número de MOOCs ao longo dos anos[36]	13
2.4	Página de um curso na plataforma Edx[42]	15
2.5	Página de um curso na plataforma Udemy[43]	16
2.6	Exercício de um curso de SQL na plataforma Codecademy[44]	17
2.7	Inscrições na disciplina CS50 entr 1989 e 2009[25]	18
2.8	Arquitetura do CS50 em formato de Cloud[45]	19
2.9	Registo das visitas únicas diárias pelo Google Analytics[48]	20
2.10	Objetivos principais na origem da inscrição[48]	20
2.11	Lista de cursos recomendados na plataforma Codio[50]	22
2.12	Lista de cursos recomendados na plataforma Nixty[58]	24
2.13	Exemplo de pergunta com o MOODLE <i>plugin</i> “javaunittest”[62]	26
2.14	Exemplo de uma classificação com o MOODLE <i>plugin</i> “javaunittest”[62]	26
2.15	<i>Screenshot</i> da arquitetura do SAAL[5]	27
2.16	Arquitetura da plataforma PBeL[63]	30
2.17	Arquitetura mais detalhada da plataforma PBeL	31
2.18	<i>Screenshot</i> do curso “Introduction to Python”[64]	33
2.19	<i>Screenshot</i> da utilização do SourceLair num projeto de PHP[65]	38
2.20	<i>Screenshot</i> da utilização do Cloud9[67]	39
2.21	<i>Screenshot</i> da utilização do Codenvy[68]	40
2.22	<i>Screenshot</i> da utilização do CodeAnywhere[69]	41
2.23	<i>Screenshot</i> da utilização do Koding[70]	42
2.24	<i>Screenshot</i> da utilização do Nitrous[71]	43
2.25	<i>Screenshot</i> da utilização do Codiad[72]	44
2.26	<i>Screenshot</i> do editor Gitbook[73]	45
2.27	Exemplo de um exercício em Javascript num livro do Gitbook	46
3.1	Diagrama dos <i>use cases</i> da funcionalidade de “Ambiente de Desenvolvimento”	51
3.2	Diagrama dos <i>use cases</i> da funcionalidade de “Ambiente de Aprendizagem”	53
3.3	Arquitetura da implementação do sistema	56
4.1	Arquitetura da implementação do sistema	62
4.2	Exemplo de <i>data binding</i> com auxílio de um <i>controller</i> [80]	63
4.3	Comparação de pedidos por segundo do Nginx com outras alternativas[81]	64
4.4	Fluxo de interação com o servidor de Identidades da UA[82]	66
4.5	Módulos do serviço de autenticação	68

4.6	Módulos do serviço de <i>cloud</i> . . . . .	71
4.7	Comparação entre Virtual Machines (genéricas) e Docker Containers[83] . . . . .	73
4.8	Diagrama de provisão da plataforma . . . . .	75
4.9	Página de <i>login</i> da aplicação <i>web</i> . . . . .	76
4.10	Diagrama de sequência do processo de autenticação a partir da aplicação <i>web</i> . . . . .	77
4.11	Página de autenticação da UA . . . . .	78
4.12	Página de autorização para a plataforma . . . . .	78
4.13	Página principal da plataforma . . . . .	79
4.14	Página que contém a lista de ambientes de aprendizagem da aplicação <i>web</i> . . . . .	80
4.15	Página (parcial) de um ambiente de aprendizagem aberto . . . . .	81
4.16	Bloco de um exercício com resultado positivo de um ambiente de aprendizagem . . . . .	81
4.17	Bloco de um exercício com resultado negativo de um ambiente de aprendizagem . . . . .	82
4.18	Página que contém a lista de ambientes de desenvolvimento da aplicação <i>web</i> . . . . .	83
4.19	Página para a requisição de um novo ambiente de desenvolvimento . . . . .	84
4.20	Página de boas vindas do ambiente de desenvolvimento . . . . .	85
4.21	Interação com o terminal do ambiente de desenvolvimento . . . . .	86
4.22	Caixa de diálogo para guardar um novo ficheiro . . . . .	87
4.23	Alerta do IDE para uma linha com erro de código . . . . .	88
4.24	Execução do projeto pelo terminal . . . . .	89
4.25	Execução do projeto pelo botão “Run” . . . . .	90
4.26	Criação do curso de Recursividade no editor do Gitbook . . . . .	91
4.27	Diagrama de sequência do processo de submeter um ambiente de aprendizagem a partir da interface REST . . . . .	93
5.1	Duração da provisão dos ambientes de desenvolvimento . . . . .	98
5.2	Recolha o consumo do CPU na máquina “Aurora” durante o teste . . . . .	99
5.3	Recolha o consumo do CPU na máquina “Borealis” durante o teste . . . . .	100
5.4	Recolha o consumo da memória RAM na máquina “Aurora” durante o teste . . . . .	101
5.5	Recolha o consumo da memória RAM na máquina “Borealis” durante o teste . . . . .	102
5.6	Representação do armazenamento da máquina “Borealis” . . . . .	103
5.7	Processos ativos da máquina “Borealis” do primeiro conjunto . . . . .	104
5.8	Processos ativos da máquina “Borealis” do segundo conjunto . . . . .	104

# LISTA DE TABELAS

---

2.1	Comparação entre os possíveis IDE a serem usados na implementação . . . . .	37
2.2	Comparação entre os vários serviços apresentados no Estado da Arte . . . . .	48
3.1	Descrição dos Atores . . . . .	51
3.2	Descrição dos <i>use cases</i> da funcionalidade de “Ambiente de Desenvolvimento” .	52
3.3	Descrição dos <i>use cases</i> da funcionalidade de “Ambiente de Ensino” . . . . .	53
4.1	Descrição do documento <i>Course</i> . . . . .	65
4.2	Descrição do documento <i>Workplace</i> . . . . .	65
4.3	Descrição dos recursos na API do serviço de autenticação . . . . .	67
4.4	Descrição dos recursos na API do serviço de <i>cloud</i> . . . . .	70
4.5	<i>Template</i> das VM na plataforma Proxmox . . . . .	74
5.1	<i>Timeline</i> dos pedidos ao serviço de <i>cloud</i> . . . . .	97



# GLOSSÁRIO

---

<b>AECT</b>	Association for Educational Communications and Technology	<b>IaaS</b>	Infrastructure as a Service
<b>AICC</b>	Aviation Industry Computer-Based Training Committee	<b>IDE</b>	Integrated Development System
<b>AMQP</b>	Advanced Message Queuing Protocol	<b>IdP</b>	Identity Provider
<b>AJAX</b>	Asynchronous JavaScript and XML	<b>I/O</b>	Input/Output
<b>API</b>	Application Programming Interface	<b>IRC</b>	Internet Relay Chat
<b>ATNoG</b>	Aveiro Telecommunications and Networks Group	<b>JSON</b>	JavaScript Object Notation
<b>AWS</b>	Amazon Web Services	<b>TI</b>	Tecnologias de Informação
<b>CPU</b>	Central Processing Unit	<b>KVM</b>	Kernel-based Virtual Machine
<b>CRUD</b>	Create, Read, Update, Delete	<b>LDAP</b>	Lightweight Directory Access Protocol
<b>CS50</b>	Computer Science 50	<b>LMS</b>	Learning Management System
<b>CS50x</b>	Computer Science 50 Edx Version	<b>MIT</b>	Massachusetts Institute of Technology
<b>CSS</b>	Cascading Style Sheets	<b>MOOC</b>	Massive Open Online Course
<b>CMS</b>	Course Management System	<b>MOODLE</b>	Modular Object-Oriented Dynamic Learning Environment
<b>DETI</b>	Departamento da Eletrónica, Telecomunicações e Informática	<b>MVC</b>	Model-View-Controller
<b>DFS</b>	Distributed File System	<b>MVW</b>	Model-View-Whatever
<b>EBS</b>	Elastic Block Storage	<b>NFS</b>	Network File System
<b>EC2</b>	Elastic Compute Cloud	<b>OSS</b>	Open-Source Software
<b>ETL</b>	Extract, Transform, Load	<b>PaaS</b>	Platform as a Service
<b>ePUB</b>	Electronic Publication	<b>PBeL</b>	Problem-Based e-Learning
<b>FTP</b>	File Transfer Protocol	<b>PBL</b>	Problem-Based Learning
<b>GCE</b>	Google Compute Engine	<b>PDF</b>	Portable Document Format
<b>GUI</b>	Graphical User Interface	<b>PHP</b>	PHP HyperText Pre-Processor
<b>HTML</b>	HyperText Markup Language	<b>PLE</b>	Personal Learning Environments
<b>HTTP</b>	HyperText Transfer Protocol	<b>QEMU</b>	Quick EMUlator
		<b>RAM</b>	Random-Access Memory
		<b>REPL</b>	Read-Eval-Print Loop
		<b>REST</b>	REpresentational State Transfer

<b>S3</b>	Simple Storage Service	<b>TIC</b>	Tecnologias de Informação e Comunicação
<b>SAAL</b>	Sistema de Apoio à Aprendizagem de Linguagens	<b>UA</b>	Universidade de Aveiro
<b>SCORM</b>	Sharable Content Object Reference Model	<b>UC</b>	Unidade Curricular
<b>SGBD</b>	Sistemas de Gestão de Base de Dados	<b>UE</b>	União Europeia
<b>SOA</b>	Service Oriented Architecture	<b>UU</b>	Utilizador Universal
<b>SQL</b>	Structured Query Language	<b>UUID</b>	Universally Unique Identifier
<b>SSH</b>	Secure Shell	<b>URL</b>	Uniform Resource Locator
<b>SVN</b>	Apache Subversion	<b>VCS</b>	Version Control System
<b>TI</b>	Tecnologia da informação	<b>VLE</b>	Virtual Learning Environments
		<b>VM</b>	Virtual Machine
		<b>W3C</b>	World Wide Web Consortium

# INTRODUÇÃO

---

*Nesta dissertação pretende-se desenvolver uma possível plataforma de E-Learning no contexto das áreas de Aprendizagem e Desenvolvimento de Software, os objetivos são fornecer novas ferramentas para promover o envolvimento do aluno no processo de aprendizagem e facilitar a colaboração dos alunos em trabalhos de equipa.*

## 1.1 MOTIVAÇÃO

Ao longo do tempo tem-se assistido à proliferação de ferramentas de ensino à distancia[1][2], que tem ganho destaque através de plataformas como Blackboard<sup>1</sup> e Modular Object-Oriented Dynamic Learning Environment (MOODLE)<sup>2</sup>[3]. Estas plataformas apesar de possuírem várias diferenças entre si, tentam focar-se nos aspetos mais tradicionais do ensino servindo de suporte para os docentes e alunos. As funcionalidades mais conhecidas são a gestão de alunos e de turmas, a gestão de conteúdos (documentos e trabalhos), aplicação de diferentes métodos de avaliação online (questionários e testes de escolha múltipla) e por fim os fóruns para promover a comunicação entre alunos e docentes. Em algumas plataformas para além do conjunto base de funcionalidades permitem ainda a adição de novas funcionalidades, através da instalação de blocos de código (conhecidos como *plugins*) desenvolvidos por terceiros. Estas funcionalidades são um eficaz suporte à educação e são transversais a qualquer área de estudo. Mas existem disciplinas mais técnicas, como por exemplo a programação, em que a utilização destas plataformas é exclusivamente como meio de comunicação e de partilha de documentos. No entanto, as disciplinas de programação continuam a ter algumas necessidades como a avaliação automática/autónoma e suporte à sintaxe de uma dada linguagem de programação. Quanto à avaliação automática de conteúdos práticos em disciplinas de programação, não é suportada nativamente por estas plataformas, pelo que recentemente já foi abordado por alguns grupos de investigação[4], resultando em soluções independentes como Mooshak<sup>3</sup>.

---

<sup>1</sup><http://blackboard.com/>

<sup>2</sup><https://moodle.org/>

<sup>3</sup><https://mooshak.dcc.fc.up.pt/~zp/mooshak/>

O Mooshak consiste num serviço que é disponibilizado à parte do sistema de *E-Learning*, que tem como funcionalidade receber ficheiros de código submetidos pelos alunos, e de forma autónoma executa-os em ambiente controlado, por fim consoante a avaliação do *output* gerado classifica-os como positivos ou negativos. Um sistema deste tipo avalia o código com base na definição prévia de um *output* feita pelo docente e caso o código submetido gere o mesmo *output*, então o código é classificado como positivo.

## 1.2 OBJETIVOS E DESAFIOS

Recentemente alguns docentes que pertencem ao Departamento da Eletrónica, Telecomunicações e Informática (DETI) têm feito um esforço para desenvolver e distribuir novas ferramentas de *E-Learning* que melhorem os métodos de ensino nos seus cursos. Como é o caso do Sistema de Apoio à Aprendizagem de Linguagens (SAAL)[5], trabalho desenvolvido pelo colega Pedro Estima da mesma instituição, que tem como base o conceito do Mooshak para permitir a avaliação automática das aulas práticas de programação.

O Mooshak como mencionado anteriormente é limitado à sua função básica de submissão e classificação de código, este funciona de forma independente sem estar agregado a nenhuma plataforma de *E-Learning*, logo não permite gestão de trabalhos, turmas e alunos.

As limitações por parte desta plataforma levaram ao desenvolvimento de uma solução, o SAAL, no âmbito de uma dissertação de mestrado, uma solução *in-house* para substituir o Mooshak e que fosse possível integrar com a atual plataforma de *E-Learning* ou com os serviços pré-existentes na instituição.

Após o desenvolvimento do SAAL, devido à necessidade de avaliação automática de exercícios básicos de programação, surgiu a ideia de analisar possíveis evoluções das atuais plataforma de *E-Learning* no contexto da informática.

A recente evolução de algumas tecnologias levou à disponibilização de novos formatos de conteúdo para o auxílio das aulas teórico-práticas, em simultâneo a proliferação de sistemas de comunicação[6][7][8] sugere que a comunicação em tempo-real é relevante para a colaboração em projetos coletivos.

Face a esta evolução e aos esforços anteriores efetuados ao nível do ensino de linguagens de programação este trabalho é um complemento e tem como objetivo o desenvolvimento de uma solução capaz de auxiliar o docente e o aluno nas disciplinas de programação. Ao docente é fornecido uma nova forma de expor a matéria através da disponibilização de um novo formato de conteúdo que permite ao aluno interagir com a matéria. Enquanto ao aluno é fornecido um ambiente remoto para a partilha de projetos de programação complexos com os seus colegas de turma como com os seus docentes. Ambas as novidades tem o intuito de fomentar assim uma melhor participação/colaboração e se possível a obtenção de melhores resultados na aprendizagem como nos trabalhos realizados.



## 1.3 REQUISITOS

A solução proposta possui várias condições à sua conceção, e a sua compreensão é necessária para perceber como as funcionalidades serão elaboradas e implementadas. O objetivo desta secção é identificar preocupações que à partida possam não ser evidentes, compreender como a abordagem a cada uma delas influencia o desenvolvimento da plataforma e como esta deverá ser disponibilizada à sua comunidade estudantil. Esta análise de requisitos é importante para identificar a dimensão do público e se possível inferir um nível de consumo/utilização da plataforma por parte dos seus futuros utilizadores. Em última instância este trabalho é um *proof-of-concept* para as plataformas de *E-Learning* do sector da Ciência de Computação, que pretende demonstrar a utilidade de consolas interativas em aulas teóricas-práticas e a utilização de ambientes remotos para a colaboração em trabalhos de grupo. Como tal o desenvolvimento deste trabalho ocorreu no âmbito da Universidade de Aveiro (UA) por esse motivo faz sentido estudar o cenário do DETI, o departamento alvo para futura utilização quando esta plataforma estiver operacional.

Pressupõe-se neste cenário de teste um público alvo aproximadamente de 200 a 300 novos alunos (por ano letivo), a compreensão do crescimento do público alvo prepara para o desenvolvimento de um sistema que seja dotado de flexibilidade e de escalabilidade para responder com qualidade a um número elevado (e em crescimento) de utilizadores. Por outro lado, a cada semestre um aluno, dependendo da Unidade Curricular (UC) em causa, pode ter entre 1 a 3 trabalho(s) prático(s), convém mencionar que não ocorre em todas as UC tendo em conta que algumas são fundamentalmente teóricas, esta dimensão permite inferir a quantidade de espaço utilizado em conteúdo teórico-prático ou na disponibilização dos ambientes de desenvolvimento partilhados.

A disponibilização de uma plataforma para um público de elevado número requer uma boa infraestrutura, assume-se que as instituições de ensino são detentoras de infraestruturas modestas, porque são entidades com orçamentos rígidos, logo a utilização de recursos como processamento, memória e armazenamento é uma operação que requer uma análise cuidada para que não haja excessos ou desperdícios nestes consumos, mas não se pode esquecer que este trabalho tem como alvo um número elevado de utilizadores ao qual é preciso fornecer um serviço. Algumas instituições ou os seus grupos de investigação poderão ser detentores de *datacenters* ou consumidores de *clouds* privadas, isto influencia a forma como a plataforma pode ser desenvolvida, demonstra que podem existir múltiplos cenários de provisão, como tal se o desenvolvimento for fechado a um único cenário impede que outras instituições com infraestruturas ou serviços diferentes não tenham condições para usar esta solução. Desta forma, o desenvolvimento da plataforma terá de possuir uma estrutura modular para que seja possível estender as suas funcionalidades consoante um conjunto de dependências que dada instituição tenha.

Este trabalho também tem como requisito fomentar a colaboração recorrendo a técnicas de *pair programming* ou a disponibilização de novos formatos de conteúdo que resultem num maior envolvimento dos alunos na aquisição de teoria e na forma como a aplicam.

Em suma, estes pressupostos influenciam diretamente a solução desenvolvida, na medida em que o *design*, a implementação e respetivos resultados serão afetados se algum destes fatores não for tido em conta. A plataforma terá de ser ágil quanto à escala dos pedidos dos utilizadores e perante a alocação de recursos de armazenamento ou processamento; permitir flexibilidade na troca de ambientes de trabalho devido às diferentes aulas lecionadas normalmente não estarem restritas a uma única linguagem de programação, conceito, paradigma ou tecnologia; pode ser tecnologicamente agnóstico e permitir o utilizador aceder à plataforma em diferentes equipamentos e por fim reutilizar serviços e plataformas que a instituição já disponibiliza, como por exemplo, o sistema de autenticação universal.

## 1.4 ORGANIZAÇÃO DO DOCUMENTO

Esta dissertação é organizada em 6 capítulos, dispostos da seguinte maneira:

- **Capítulo 2:** Introdução aos conceitos inerentes à *Open Education* e ao *E-Learning*, identificação do estado-de-arte e das características e objetivos que diferenciam as ferramentas enumeradas;
- **Capítulo 3:** Apresentação das funcionalidades, conceitos e requisitos que descrevem uma proposta para uma nova solução de *E-Learning*;
- **Capítulo 4:** Exposição da arquitetura efetiva, enumeração das tecnologias utilizadas e outros detalhes respetivos ao desenvolvimento da plataforma;
- **Capítulo 5:** Descrição da experiência realizada e apresentação dos resultados obtidos;
- **Capítulo 6:** Apresentação das conclusões adquiridas na realização do trabalho e identificação de melhorias para possível trabalho futuro.

## ESTADO DA ARTE

---

*Neste capítulo pretende-se contextualizar o estado do E-Learning e ao mesmo tempo identificar a evolução do qual o setor foi alvo e de que forma isso contribuiu para as soluções da atualidade. Após a compreensão deste enquadramento será exposta a análise realizada a diferentes soluções, exclusivas ou não à área da informática e da programação. Cada solução será exposta e avaliada e se possível identificar o formato os conteúdos são disponibilizados e que tipo de interação é permitida aos seus utilizadores, deste modo enumerar vantagens ou desvantagens que tem para a prática dos exercícios de programação. Por fim, foi realizada uma pesquisa de ferramentas e estudar se é possível a sua integração da implementação da solução proposta (no capítulo seguinte).*

### 2.1 ENQUADRAMENTO

Nos últimos anos as Tecnologias de Informação e Comunicação (TIC) têm recebido atenção por parte dos sistemas de ensino, em meados de 1960, o sistema PLATO[9] que consiste numa experiência para avaliação da utilização de um computador numa sala de aula como ferramenta de suporte. Também a Universidade de Standford, desenvolveu uma experiência semelhante que consistiu na introdução de um computador na sala de aula do Ensino Básico para os alunos realizarem exercícios de aritmética[10].

Inicialmente, as soluções anteriormente enunciadas tinham como foco melhorar os métodos de ensino tradicionais, mas a dada altura outras preocupações surgiram como o ensino à distância. Em 2002 a distribuição de conteúdos de forma gratuita (*OpenCourseWare*) ganhou uma nova dimensão quando o Massachusetts Institute of Technology (MIT) aderiu ao movimento[11][12] e em 2007 já tinham cerca de 1500 cursos públicos, e um recorde mensal de visitas que chegou a atingir os 2 milhões[12].

Outras soluções emergiram quando a necessidade passou além da distribuição, nomeadamente gestão de turmas/alunos. Em 2003, surgiu o MOODLE[13], numa fase inicial era um simples Course Management System (CMS), com o passar dos anos e com a evolução do seu sistema e das suas funcionalidades, o anterior CMS deu espaço ao aparecimento dos Learning Management System (LMS), que para além de abranger as características de um CMS tinha uma infraestrutura suportada por standards como SCORM e AICC que permite integração com outros sistemas (*courseware*, autenticação, recursos humanos), organização de currículos e certificações, suporte para realizar avaliações aos alunos e permite flexibilidade quanto ao método de ensino (que pode ser meramente um suporte à sala de aula ou um sistema *online*)[14].

Em 2008 surgiu um novo conceito, o Massive Open Online Course (MOOC), que tinha como principal objetivo eliminar diferentes barreiras à educação. Surgiu numa época em que o acesso ao conhecimento e a vários conteúdos era de restrito acesso, para proteger a sua propriedade intelectual. Desta forma, as instituições só permitiam o acesso ao conhecimento a indivíduos que fossem da instituição ou que pagassem para aceder ao mesmo. Com esta inovação, foi possível a diferentes instituições tornar o acesso ao conhecimento livre e aberto a qualquer indivíduo, o que por sua vez elimina a barreira financeira devido aos conteúdos serem grátis, e também elimina a barreira geográfica porque permite a qualquer indivíduo aceder ao conteúdo em qualquer lugar desde que possua uma ligação à *internet*[15].

Só quatro anos mais tarde, o movimento teve um foco maior[16], este deveu-se à adesão de instituições como Princeton e Brown à plataforma Coursera<sup>1</sup> enquanto o número de utilizadores aumentava. Os MOOC são uma solução que permite o ensino à distância e de forma gratuita, que promove oportunidades de educação iguais a todos os alunos, sejam eles privilegiados (financeiramente ou geograficamente) ou não.

Entretanto, a disseminação dos conceitos e tecnologias da *web 2.0*, que fundamentalmente fomentam a colaboração e comunicação entre os utilizadores, permitiu criar ferramentas para a partilha de conteúdos, conhecimento e experiências e conseqüentemente o surgimento de comunidades *online*. Na perspetiva de conteúdo, plataformas como *blogs* ou *wikis* aumentaram o número de fontes de informação, porque é um veículo que qualquer um pode utilizar para partilhar conteúdos com o resto do mundo, e normalmente estas plataformas permitem uma dinâmica de interação entre produtores e consumidores podendo resultar em correções ou adicionar valor no conteúdo originalmente criado. Este aumento de conteúdos disponíveis é propício à aquisição de conhecimento e como tal surgiu o conceito de *Mashup* que pretende reaproveitar a informação disponível e dispersa pela *web*. Através do conceito *Mashup* é possível agregar múltiplas fontes de informação (aplicações, serviços ou recursos) e gerar o seu ambiente personalizado de aprendizagem [17]. Em comunidades deste género, o papel do docente sofre uma alteração deixando de ser o produtor de conteúdo e é tido como um *enabler* que disponibiliza distribuição e partilha de informação usando uma plataforma central, por exemplo, um LMS. Desta forma, o aluno tem a possibilidade de configurar um Personal Learning Environments (PLE) de um modo ainda mais flexível com acesso a recursos de outras entidades.

---

<sup>1</sup><https://www.coursera.org/>

## 2.2 OPEN EDUCATION

Como parte da sua agenda política, a Comissão Europeia pretende “abrir” a educação para reduzir ou até mesmo remover barreiras que existem no acesso à mesma, por exemplo, barreiras financeiras, geográficas ou de tempo, e como resultado modernizar a educação superior na Europa de modo a permitir que as pessoas possam se formar ou adquirir novas competências de uma forma flexível de baixos custos, uma medida face à crise económica que a Europa enfrenta atualmente. Dado que o método atual é fundamentalmente suportado por tecnologias digitais, a aplicação destas medidas espera originar uma modernização do ensino superior na Europa, e por fim possivelmente disponibilizar educação em que as diferenças entre os métodos de educação formal e informal (por exemplo, MOOC) sejam reduzidas.

O projecto OpenEdu[18] pretende abordar esta temática e definir o conceito de *Open Education* apesar do que o nome sugere não está exclusivamente relacionado com a distribuição ou acesso livre a conteúdos como era entendido nos anos 60, atualmente este conceito pode adotar vários significados. Um dos objetivos deste projeto é resolver estas diferenças e encontrar uma definição comum a todos, como resultado pretende disponibilizar uma plataforma de *Open Education* com o suporte das TIC, como já tem vindo a ser comum nas instituições de ensino em vários membros da União Europeia (UE). Com a utilização das TIC, a plataforma *Open Education* pretende tornar o ensino mais acessível, abrangente e customizável, e em último caso resolver conflitos entre métodos formais e informais de educação. Esta plataforma é direcionada para as instituições de ensino superior, para os diferentes estados membros e legisladores e que a usem como ferramenta para incentivar a inovação na educação[19].

As ferramentas usadas para formular a plataforma *Open Education* foram várias, o OpenCases[20] que foi realizado em algumas instituições do ensino superior na Europa e tinha como objetivo auferir a compreensão de uma educação aberta e averiguar como e porquê de algumas instituições ou iniciativas governamentais abordarem o tópico da educação aberta. A recolha de dados para estes caso de estudo foi através de entrevistas aos membros seniores da instituição. O OpenCred[21] é um estudo focado na credenciação e reconhecimento dos paradigmas disponibilizados pelo ensino informal (como os MOOC), e se possível compreender as dificuldades que os aprendizes, as instituições e os empregadores enfrentam para credenciar ou reconhecer formação sustentada por estes métodos, neste caso também foi realizado um conjunto de entrevistas distribuídas de forma igual entre académicos de instituições do ensino superior, utilizadores de plataformas MOOC e funcionários de empresas recrutadoras. Um outro estudo à aprendizagem por MOOC, é o MOOCKnowledge que no momento em que o relatório[19] foi lançado ainda estava a decorrer mas consiste na realização de vários questionários (antes do curso, depois do curso e *follow-up* um ano depois) realizados a diferentes alunos de MOOC fornecidos por entidade europeias, e tem como objetivo identificar a sua motivação, experiência de aprendizagem, impacto na vida profissional e pessoal na participação de um MOOC. Por fim, a última ferramenta usada, o OpenSurvey[22] que é a realização de questionários a diferentes instituições do ensino superior de diferentes estados membros da UE, e identificar se e como utilizam políticas de *Open Education*, ou se não utilizam e porquê.

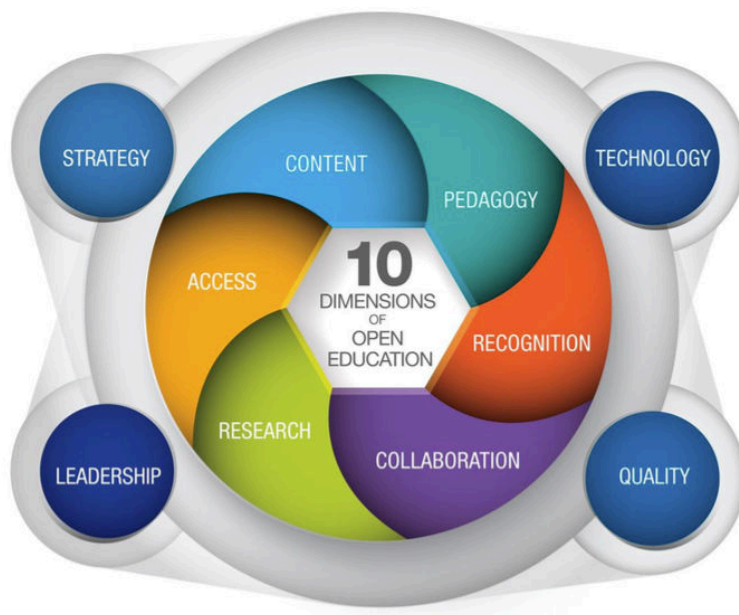


Figura 2.1: As dimensões do *Open Education*[19]

Como resultado do projeto OpenEdu realizado pela Comissão Europeia foram identificadas várias dimensões que descrevem o significado de *open* em *Open Education* e que estão divididas em dois grupos, como pode ser visto na Figura 2.1.

O primeiro grupo é constituído pelas dimensões que classificam **o que é** uma educação *open*. Para providenciar uma educação aberta, as dimensões consideram que o **acesso** a materiais de ensino e a dados/resultados de investigação deve ser livre. Resulta na remoção de possíveis barreiras ao **conteúdo** como dar permissão de acesso a conteúdos, cursos, comunidades que anteriormente estariam de alguma forma bloqueadas ou inacessíveis. Recorrer a um uso mais amplo das tecnologias para melhorar a **pedagogia** e a qualidade dos processos de ensino para aumentar o envolvimento dos alunos. O **reconhecimento** através da criação de métodos para certificação por instituições creditadas ou métodos alternativos que permitem a um indivíduo adquirir certificação para um dado conhecimento ou competência fornecida por uma instituição não creditada. Uma dimensão que se preocupa com a criação de sinergias para melhorar os resultados que as instituições e as diferentes entidades de investigação alcançam, com recurso a **colaboração** entre indivíduos e instituições porque encoraja uma comunicação aberta sem barreiras e possivelmente resulta na produção de novos recursos e contribui para o desenvolvimento de novos métodos e conhecimento de melhor qualidade. Uma **investigação** sem barreiras resulta na partilha e acesso livre dos resultados das mais variadas experiências com o objetivo de acelerar o processo científico com a troca de conhecimento e fomenta a colaboração ente peritos por consequente resulta em experiências e resultados mais ricos.

O segundo grupo é um outro conjunto de dimensões transversais aos pontos referidos no parágrafo anterior. A sua compreensão permite classificar **como** obter uma educação *open*. Através da **estratégia** é possível o planeamento e implementação de políticas por parte das instituições para abordar oportunidades de educação aberta, tendo em conta os recursos e capacidades de que dispõe. Entretanto, com uma boa **liderança** elabora-se a promoção de medidas sustentáveis que suportem adoção de uma educação aberta na instituição. A dimensão **tecnológica**, desde a infraestrutura até ao *software*, tem como principal objetivo ser um *enabler* para todas as outras dimensões e como resultado melhorar a qualidade dos processos de ensino, como facilitar a aquisição e partilha de conhecimento. Por fim, a **qualidade**, uma dimensão inerente a tudo e esta dimensão expressa-se em 5 conceitos de qualidade (eficácia, impacto, disponibilidade, precisão e excelência) para classificar os esforços/resultados aplicados nas dimensões do primeiro grupo[19].

As considerações finais desta investigação demonstram que uma educação aberta é uma prioridade na Europa mas que várias instituições de ensino superior ainda não têm uma estratégia bem definida devido a algumas ainda não possuírem uma compreensão necessária para obterem uma educação aberta, outras assimilam o movimento como disponibilizar os recursos ou MOOC, por fim, existe uma opinião generalizada no aspeto positivo de uma educação aberta, mas não conseguem definir com clareza as práticas e benefícios envolvidos. Posto isto, definiram algumas recomendações: definir uma estratégia que incorpore as várias dimensões especificadas pela plataforma de *Open Education* e se possível resultar numa possível reestruturação da educação superior, que as políticas de educação aberta façam parte da macro-estratégia das instituições para enriquecer a sua oferta, promover relações intra-, inter- e além fronteira para um impacto global e desta forma melhorar os recursos e o capital humano, e por fim, fomentar uma mentalidade aberta e disponível para aplicar mudanças nos métodos ou horizontes[19].

## 2.3 E-LEARNING

A Association for Educational Communications and Technology (AECT)<sup>2</sup> é uma associação internacional com o objetivo de investigar e promover novos métodos de utilização da tecnologia para melhorar os processos de aprendizagem no sector de educação. Os membros desta associação podem pertencer ao ensino secundário/superior, à Indústria, etc, e em 2008, escreveram a sua quinta definição para “Educational Technology”:

“Educational Technology is the study and ethical practice of facilitating learning and improving performance by creating, using and managing appropriate technological processes and resources.” [23]

Tendo em conta esta definição, as primeiras implementações de plataformas de *E-Learning* remontam aos anos 60 quando os alunos usavam terminais em rede para colaborar com uma comunidade *online*[9] ou a aceder a conteúdos auxiliares às classes[10], e não para os meados do anos 2000 quando surgiu a massificação de algumas plataformas nas instituições.

A história regista que estes primeiros passos no *E-Learning* eram realizados em pequenos domínios (redes privadas) e que à medida que as funcionalidades demonstravam resultados positivos, as plataformas ganhavam adeptos e com auxílio da evolução tecnológica o número de utilizadores era expandido passando de pequenas redes a redes de maior dimensão, em último caso, redes como a *Internet*.

---

<sup>2</sup><http://aect.site-ym.com>

O MOODLE, atualmente conhecido como um LMS, quando apresentado à comunidade académica era um CMS e era assim denominado pelos seus criadores Dougiamas e Taylor. Por definição, um CMS é um repositório de conteúdos e possibilita uma espécie de gestão básica entre alunos, disciplinas e projetos.

Com a evolução da *web*, mais precisamente na transição entre a versão 1.0 e a 2.0, em 2005, o MOODLE já era classificado como um LMS e um dos mais populares na altura[24].

Posteriormente em “Harvard College”, mais precisamente em 2007, a instituição enfrentava na altura um problema devido à estagnação (que era três vezes inferior ao pico que atingiram alguns anos antes) de inscrições num dos seus cursos de Ciências de Computação. Como tal, decidiram recorrer às TIC para promover atividade fora do tempo de aulas, permitido o acesso aos recursos das disciplinas (em diversos formatos, como por exemplo, vídeo) fora da rede privada da instituição (por exemplo, em casa), a disponibilização de novos métodos de atendimento ao aluno para esclarecimento de dúvidas recorrendo a soluções virtuais/digitais, a disponibilização de um fórum no qual os alunos poderiam participar anonimamente, entre outras medidas. No qual ficou provado que estas melhorias têm repercussões nos presentes/futuros alunos, devido aos resultados que indicam um aumento considerável de inscrições, assim como também houve um aumento de alunos a completar o curso nos anos que se seguiram[25].

Paralelamente, outras experiências ocorriam no âmbito do *E-Learning*, não tanto focados na *web* ou nas plataformas mais conhecidas, mas mais relacionadas com conceitos de Mundos Virtuais. A disponibilização de mundo virtual consistia na utilização de um simulador 3D, denominado *Second Life*<sup>3</sup>. A UA foi “a primeira instituição do ensino superior portuguesa a abrir um espaço”[26] neste mundo virtual. O objetivo da equipa por detrás deste projeto foi fundamentalmente criar espaços com condições para fomentar a discussão dos diversos temas lecionados nas aulas e comunicação entre os alunos, e também promover o interesse em atividades de lazer com componente social.

Aderir ao *Second Life* foi uma ação pioneira da UA, e em 2016 o espaço continua a ser promovido para ser utilizado pelos alunos[27]. Noutras partes do mundo o *Second Life* é também usado como ferramenta para fornecer serviços de ensino à distancia, por exemplo na criação de réplicas virtuais das salas de aulas, e permitir os alunos a assistir as aulas. Também existem utilizadores que usam mundo virtual (e as ferramentas de simulação que fazem parte dele) para criar ou melhorar processos de ensino, como é o caso de cursos de Ciência Forense e *Design Ambiental*[28].

Pelos registos, a UA em 2009 mais uma vez ativa no que respeita ao *E-Learning* depois do espaço virtual em *Second Life*, e ainda foi palco para outras experiências, nomeadamente o projeto SAPO Campus<sup>4</sup>, uma colaboração entre a Universidade e o SAPO, que pretendia ser uma representação de uma Universidade sem paredes alojada na *web* 2.0, e ao mesmo tempo tirar partido das redes sociais para melhorar a aprendizagem e a comunicação entre alunos e docentes, ou seja, a criação de comunidades de aprendizagem.

Ainda no mesmo ano, foi submetido por investigadores da UA um artigo numa revista *online* dedicada ao *E-Learning*, denominado “chmod 777 education”[29] em abstrato sugerem recorrer ao grande potencial da *web* 2.0, à semelhança do que é pretendido pelo SAPO Campus, para permitir a troca de conhecimento de forma aberta com os princípios de colaboração, participação e partilha, originando numa enorme comunidade de aprendizagem sem fronteiras, no qual advogam que é mais um problema cultural do que técnico ou tecnológico.

---

<sup>3</sup><http://secondlife.com>

<sup>4</sup><http://campus.sapo.pt/>



## M-LEARNING E B-LEARNING

No domínio do *E-Learning* existem duas modalidades que se diferenciam pelo tipo de tecnologias adotadas, pelo foco e pelos métodos utilizados. Enquanto que o *E-Learning* distingue-se dos paradigmas tradicionais pela dependência das tecnologias e da *Internet*, o *M-Learning* é um paradigma de ensino à distância caracterizado pela mobilidade com origem no crescimento (ver Figura 2.2) e diversidade dos dispositivos móveis.

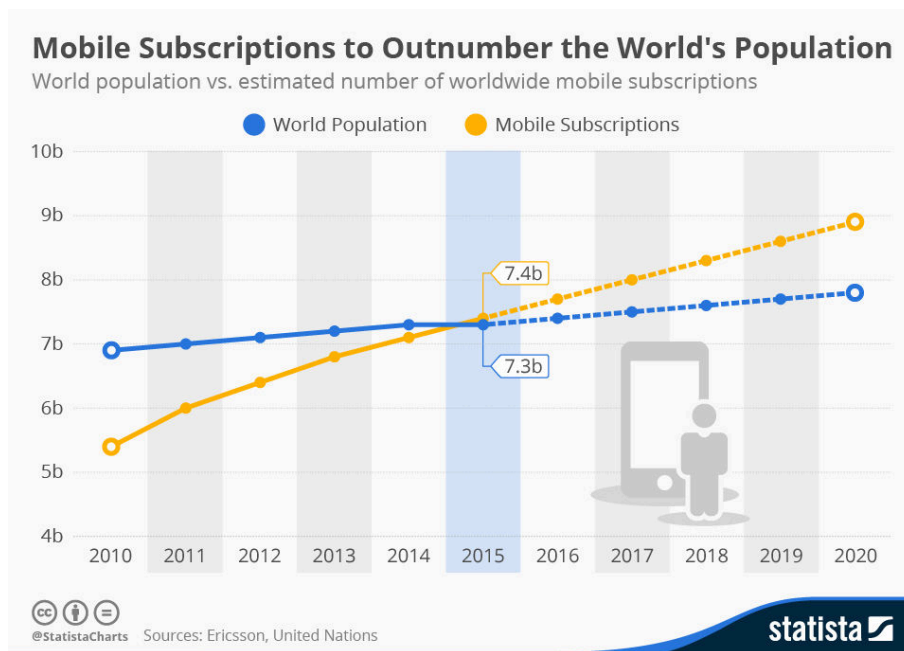


Figura 2.2: Comparação entre dispositivos móveis *versus* a população mundial[30]

Esta tendência que tira proveito das aplicações de *software* e de protocolos de rede para permitir que os seus utilizadores tenham acesso ao conteúdo de aprendizagem sem limitações, acessível em qualquer lugar e a qualquer hora, o que torna numa vantagem para o envolvimento do aluno e permite que os diferentes participantes possam aceder através de diferentes equipamentos como alternativa aos métodos tradicionais que envolvem o aluno estar fisicamente presente com um professor/orador[28].

Um paradigma considerado como disruptivo, porque permite acesso ao conhecimento e a plataformas de educação por parte de indivíduos que habitam em países em desenvolvimento ou com dificuldades sociais ou económicas[31]. Um exemplo de uma plataforma que implementa esta modalidade é a SoloLearn<sup>5</sup>, que é um sistema de *M-Learning* focado nos tópicos de linguagens de programação.

A segunda modalidade é recorrente nos dias de hoje, a sua utilização já é expectável, pelo menos em Portugal já existem diferentes instituições tanto do ensino secundário como do superior que recorrem a plataformas como MOODLE para suportar *B-Learning* nos seus processos de ensino. Este paradigma disponibiliza a utilização das TIC como ferramentas de suporte ao ensino, e a sua utilização deve ocorrer em paralelo com a participação presencial (face a face) na respetiva instituição[28].

<sup>5</sup><https://www.sololearn.com/>

## VIRTUAL LEARNING ENVIRONMENTS E LEARNING MANAGEMENT SYSTEM VS PERSONAL LEARNING ENVIRONMENTS

O surgimento do conceito Virtual Learning Environments (VLE) por definição, recorre a diferentes tecnologias para criar e disponibilizar um espaço virtual de forma a que o utilizador tire partido de novas formas de interação com outros utilizadores, como também tem acesso a uma variedade de recursos. Este tipo de ambiente fomenta a filosofia de aceder em qualquer lugar e a qualquer hora. Normalmente são considerados sistemas abertos porque são dotados de extensibilidade ao nível da funcionalidade base, apesar de distribuírem uma interface consistente por grupos de membros da instituição[32].

Com uma visão diferente, temos os PLE que tem sido considerados como ambientes mais pessoais focados no indivíduo[33]. É visto maioritariamente como uma questão pedagógica em relação à anterior devido a estar centralizada nas necessidades do utilizador, as tecnologias são vistas como um meio para que o utilizador possa personalizar o seu ambiente de forma a criar o seu domínio de aprendizagem[34]. Este controlo é obtido através da personalização das fontes de recursos, e funcionalidades através da integração de serviços externos.

Por fim, tem também como objectivo ser um *standard* e criar uma interface para múltiplas instituições, como meio para a criação de um portefólio flexível através da agregação da informação dessas instituições[35].

## MASSIVE ONLINE OPEN COURSE

Na esperança de se tornarem mais proeminentes no espaço *online*, algumas instituições abordaram temas como dificuldades financeiras, geográficas e sociais, que na perspetiva da comunidade afetavam a qualidade da educação, devido a existir um distanciamento ao nível das oportunidades entre alunos, que eram privilegiados (financeiramente, geograficamente ou socialmente) ou não[15]. Devido a este cenário o conceito introduzido em 2008, o MOOC, atingiu uma mediatização maior em 2012 (ver Subsecção 2.5.1).

Como o nome sugere, o conceito de MOOC é suportado por redes como a *internet* e a *web* com o objetivo de disponibilizar o conhecimento ao mundo de forma gratuita, e a números massivos de alunos que atualmente é relativamente fácil de gerir com recurso a plataformas de *cloud* que permitem grande flexibilidade e recorrem a paradigmas de sistemas distribuídos, mesmo assim algumas decisões de planeamento e desenvolvimento podem afetar a escalabilidade do serviço.

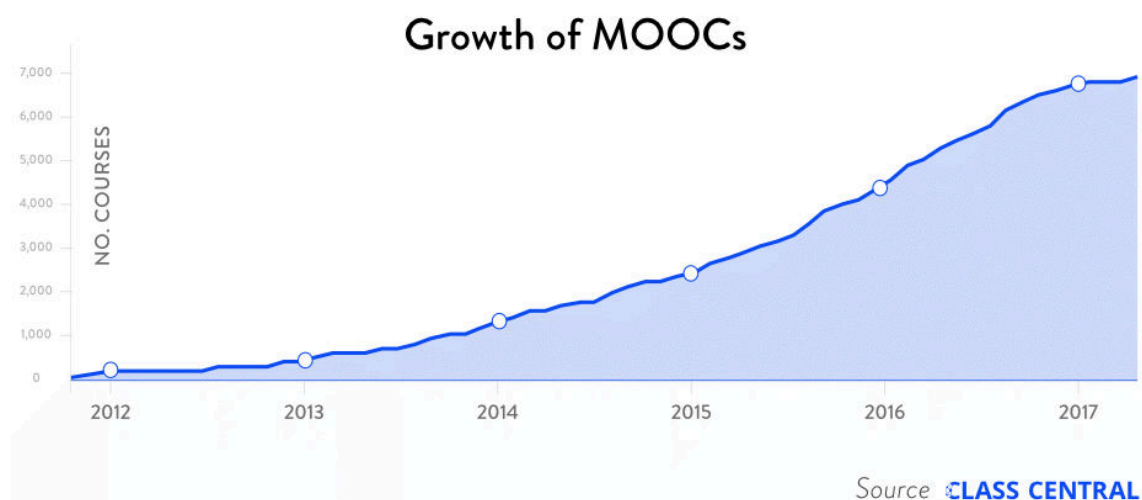


Figura 2.3: Número de MOOCs ao longo dos anos[36]

O *website* Class Central<sup>6</sup> possui um enorme catálogo organizado de MOOC e também se auto-intitula *desearch-engine* de MOOC. Todos os anos realizam um relatório em que partilham e analisam várias estatísticas relacionadas com MOOC.

No último relatório relativo ao ano de 2016 é possível verificar que o MOOC é uma tendência que ainda se encontra em fase de crescimento (ver Figura 2.3). O gráfico da Figura 2.3 apresenta o número total de cursos indexados no seu catálogo sem considerar os que vão sendo eliminados ao longo do tempo. De qualquer forma, no último ano foram identificados mais de 2000 novos cursos em relação ao ano anterior.

No mesmo relatório, foi declarado que no universo MOOC, distribuído entre vários provedores ou serviços, existem aproximadamente 700 universidades e 58 milhões de utilizadores. Com base nos dados retirados dos relatórios de 2015 e 2016, verificou-se uma alteração nos cursos de “Ciências de Computação e de Programação”. No ano 2015, estes cursos estavam separados em duas partes distintas mas em conjunto auferiam 17,18%[37] de todos os MOOC registados e esse valor cresceu ligeiramente em 2016 para os 17,4%[36]. Em termos de percentagem de MOOC disponíveis o setor “Ciências de Computação e de Programação” é o segundo maior, com isto pode se extrapolar um possível interesse do público nestas áreas, justificando assim um investimento para melhorar a qualidade das ferramentas de *E-Learning* deste sector.

<sup>6</sup><https://class-central.com>

## 2.4 E-LEARNING 2.0

Quanto à *web 2.0*, Stephen Downes afirma que não é uma revolução tecnológica mas uma revolução social[38], porque o objetivo era o típico utilizador ser produtor de conteúdo, resultando na ascensão de plataformas como *Wiki*, *Blogs*, e *Podcasts*, e a utilização destas soluções permite a criação de comunidades em torno do conteúdo em que a colaboração entre os vários membros é relevante para a discussão e correção do conteúdo criado.

Em vários momentos estas tecnologias foram vistas como tendo potencial para serem utilizadas em vários cenários académicos, por exemplo a criação de *Podcasts* com áudio e vídeo em que o docente expõe a matéria, a utilização de *Blogs* por parte dos alunos para que possam expor os seus trabalhos e resultados obtendo assim um parecer do público e como resultado dessas interações melhorar o seu conteúdo ou corrigir erros do mesmo, e por fim, a utilização de *Wikis* é uma plataforma que permite a um conjunto de alunos escrever, editar e rever artigos sobre um ou vários assuntos[39].

Existem alguns especialistas que assumem que a utilização de *web 2.0* em paralelo com tecnologias de *E-Learning 1.0* não cria automaticamente o conceito *E-Learning 2.0*, o componente em falta é o fator humano. Existe uma curva de aprendizagem relativamente à evolução tecnológica que advém da *web 2.0*, mas é preciso que os docentes estejam disponíveis para uma transformação do actual paradigma e tirar proveito destas novas ferramentas para uma nova dinâmica entre aluno, o conteúdo e o docente[38][39].

## 2.5 SERVIÇOS ONLINE

Após o enquadramento da área de *E-Learning*, para compor o estado da arte foram realizadas várias análises a diferentes soluções de *E-Learning* e que serão apresentadas ao longo deste capítulo. Cada análise foi realizada com um foco diferente ou com o objetivo de avaliar um conjunto de funcionalidades diferentes. Nas secções seguintes, essas diferenças são analisadas e relacionadas com a solução proposta neste trabalho.

Nesta primeira parte, serão estudadas plataformas de *E-Learning* disponibilizadas como serviços *online* possivelmente comerciais.

## 2.5.1 EDX

Em 2012, o edX<sup>7</sup> surgiu como plataforma do tipo MOOC, atualmente aloja vários cursos de diversas Universidades (MIT, Harvard, Berkeley) e cursos (Arquitetura, Biologia, Química, Ciências de Computação, Física, entre outros.). Normalmente, os cursos disponibilizados pelas instituições são grátis mas sem certificação incluída, para que o aluno tenha direito à certificação terá de pagar pela mesma, um valor definido por quem publica o curso, à exceção dos cursos profissionais que são disponibilizados por entidades que têm cursos mais complexos, neste caso a própria entidade define o valor da inscrição[40].

O formato de ensino à distância nesta plataforma divide os capítulos de aprendizagem por semanas e é assim que se quantifica o tempo do curso, à parte disso, quem disponibiliza o curso pode partilhar conteúdos em vídeo e depois avaliar o aluno através de questionários. Por fim, os cursos são disponibilizados em dois modos: o *self-pace* que é *user-driven* que dá o controlo ao aluno, este pode escolher o ritmo a que deseja realizar o curso; o oposto é *professor-driven* quando é criado o curso dita o ritmo do curso, ao qual o aluno terá que se sujeitar para participar, neste modo o docente tem de definir uma calendarização para curso.

Na tentativa de produzir uma plataforma sem fins lucrativos, diversos intervenientes associados ao edX ajudaram a produzir uma versão *open source*, denominada OpenEdx<sup>8</sup>. Com esta plataforma é possível contribuir para a comunidade e disponibilizar a sua própria plataforma de MOOC. Como principal objetivo, esta plataforma permite que qualquer instituição desenvolva os seus métodos de partilha de conhecimento da forma que seja mais conveniente[41].

The screenshot shows the EdX website interface. At the top, there is a navigation bar with the EdX logo, links for 'Courses', 'Programs', 'Schools & Partners', and 'About', a search bar, and 'Sign In' and 'Register' buttons. The main content area features a course card for 'Introduction to Computer Science and Programming Using Python' by MIT. The card includes a video player thumbnail, the course title, a description: 'A new and updated introduction to computer science as a tool to solve real-world analytical problems using Python 3.5.', the MIT logo, and a green 'Enroll Now' button. Below the button is a checkbox for email notifications. A grey bar below the card states 'Academic credit is available for this course through Charter Oak State College'. To the left of the card, there is an 'About this course' section with a 'See more' link and a 'What you'll learn' section with two bullet points: 'A Notion of computation' and 'The Python programming language'. To the right of the card, there is a sidebar with course details: Length (9 weeks), Effort (15 hours per week), Type (Verified, Earn Credit, Audit), Price (FREE, Add a Verified Certificate for \$49), Institution (MIT), Subject (Computer Science), and Level (Introductory).

Figura 2.4: Página de um curso na plataforma Edx[42]

<sup>7</sup><https://www.edx.org/>

<sup>8</sup><https://open.edx.org/>

## 2.5.2 UDEMY

A UdeMy<sup>9</sup> é uma plataforma que disponibiliza um *marketplace* para o ensino, funciona como *enabler* para os alunos e docentes que se registam na comunidade. Por defeito, pode disponibilizar um MOOC mas é uma decisão a ser tomada pelo criador do curso, que pode ou não publicá-lo de forma gratuita. É uma plataforma semelhante a outros sistemas de *E-Learning*, o conteúdo é essencialmente focado em vídeo, está organizado por capítulos (ver Figura 2.5). No final de cada capítulo pode existir um questionário, que coloca novamente o instruendo numa posição de avaliação. Consoante o teor do curso, o docente pode ou não ter conteúdos digitais auxiliares como: documentos/guias e projetos de exemplo que o aluno pode fazer *download*. Ao contrário de outras plataformas, na UdeMy os cursos disponibilizados são por natureza *self-paced*, ou seja, após a inscrição o instruendo tem o controlo sobre o curso, decidindo a que velocidade quer avançar. Este tipo de plataforma pode ser apelativa para muitos utilizadores porque melhor se adequam aos seu métodos de estudo, para alunos que passam longas horas a estudar, não possuem qualquer impedimento ao nível do tempo, e caso sejam alunos com algumas dificuldades, podem fazer pausas, retomar ou rever, e aprofundar os temas, sem que tenham o risco de perder alguma “aula” ou “matéria”.

A razão porque esta plataforma do tipo *marketplace* não satisfaz os interesses de uma instituição do ensino superior é devido a ser uma plataforma comercial e não é compatível com o atual paradigma de *B-Learning* aplicado nas universidades. Para além disso, como sistema de *E-Learning*, esta solução dispõe de ferramentas semelhantes às disponibilizadas por um LMS como o MOODLE, tanto para a partilha de conteúdo multimédia como para a avaliação da matéria lecionada.

The screenshot displays the course page for 'Java Basics - A Complete course for Beginners'. The course has a rating of 3.8 (21 ratings) and 3,710 students enrolled. The curriculum is organized into sections: 'Introduction' (01:30) and 'Java basics' (04:01:47). The 'Java basics' section includes 17 lectures, such as 'Introduction to java' (02:04), 'Setting up eclipse' (02:33), and 'Hello world program' (04:12). A promotional banner on the right offers a 95% discount, reducing the price from 200€ to 10€. The banner includes a 'Buy Now' button, an 'Add to Cart' button, and a '30-Day Money-Back Guarantee'. The banner also lists the following inclusions: 4.5 hours on-demand video, Full lifetime access, Access on mobile and TV, and Certificate of Completion. A 'Have a coupon?' link is also present.

Figura 2.5: Página de um curso na plataforma UdeMy[43]

<sup>9</sup><https://www.udemy.com/>

### 2.5.3 CODECADEMY

Começou por ser uma pequena plataforma, para o ensino de Javascript, atualmente evoluiu para uma plataforma que tem como objetivo garantir uma melhor experiência para aprendizagem ao nível de diferentes tecnologias *web* e tecnologias complementares ao desenvolvimento de *software*, como por exemplo, AngularJS, HTML, Git, Python entre outras.

Ao contrário das plataformas de *E-Learning* alternativas em que a transmissão de conhecimento teórico é realizado com auxílio de vídeo, questionários ou documentos. A Codecademy<sup>10</sup> é uma plataforma foca-se exclusivamente no ensino de linguagens de programação e ferramentas adjacentes através de consolas interativas. O conceito não é novo no desenvolvimento de *software*, devido a várias linguagens de programação possuírem estas consolas que permite aos programadores por exemplo testarem instruções, no entanto, este cenário está a ser reutilizado para as plataformas de *E-Learning*.

Cada curso é composto por capítulos, e cada capítulo disponibiliza uma consola interativa que permite ao aluno executar um conjunto de tarefas para aprendizagem do tópico em causa, mas com interatividade em tempo-real, a plataforma vai indicando a progressão do aluno e se está a inserir código correto ou incorreto (ver Figura 2.6). Fornece uma experiência diferente ao aluno, porque no momento da aprendizagem não é confrontado com problemas de preparação de projetos (entre outros) e lida simplesmente com tarefas, problemas ou exercícios de forma isolada podendo dedicar toda a sua atenção à sintaxe ou à resolução desses problemas.

É uma plataforma de fácil utilização que permite a aquisição de conceitos básicos e importantes das linguagens disponíveis, em comparação com algumas alternativas perde por ainda não ter a relevância que uma plataforma e com algum tipo de certificação que a comunidade considere válida.

The screenshot displays the Codecademy interface for a SQL course. On the left, there is an 'Objective' section with instructions and a 'Tasks' section with 3/7 tasks completed. The main area shows a SQL code editor with the following code:

```
1 CREATE TABLE tracks (  
2   id INTEGER PRIMARY KEY,  
3   title TEXT,  
4   album_id INTEGER  
5 );  
6  
7 INSERT INTO tracks (id, title, album_id)  
8   VALUES (1, "Smooth Criminal", 8);  
9  
10 INSERT INTO tracks (id, title, album_id)  
11   VALUES (2, "On the Run", 11);  
12  
13 INSERT INTO tracks (id, title, album_id)  
14   VALUES (3, "Time", 11);  
15  
16 INSERT INTO tracks (id, title, album_id)  
17   VALUES (4, "Money", 11);  
18  
19 Select * from tracks;
```

On the right, the 'Query Results' table shows the following data:

id	title	album_id
1	Smooth Criminal	8
2	On the Run	11
3	Time	11
4	Money	11

Below the query results, the 'Database Schema' is shown, including tables for 'artists', 'albums', and 'tracks' with their respective columns and data types.

Figura 2.6: Exercício de um curso de SQL na plataforma Codecademy[44]

<sup>10</sup><https://www.codecademy.com/>

## 2.5.4 CS50

Na universidade de Harvard existe um curso de introdução às ciências de computação, denominado Computer Science 50 (CS50)<sup>11</sup>. E foi utilizado para avaliar os resultados obtidos da adoção de metodologias de *E-Learning*. A primeira iniciativa ocorreu quando confrontados com uma inconformidade no número de alunos inscritos. Entre 1992 e 1996, ocorreu um crescimento acentuado de inscrições de alunos neste curso, no qual atingiu aproximadamente 400 inscrições. As preocupações surgiram quando a adesão de alunos desceu drasticamente nos anos seguintes até 2002, que atingiu o valor mais baixo (entre 1989 e 2009), cerca de 100 inscrições[25].

Após esta descida, o número de alunos continuava baixo e com aumentos insignificantes. Em 2007, insatisfeitos com a situação os responsáveis pelo curso procuraram abordar o problema e resolve-lo. Com a informação que detinham supuseram que o curso sofria de um mau plano ou de uma percepção negativa por parte de futuros alunos, e que o problema não era ao nível dos docentes nem do conteúdo. Porque os dados de inquéritos aos alunos assim o demonstravam[25].

A solução passou por tornar o curso mais acessível, sem descurar da qualidade nem dos objetivos do curso, no novo plano organizaram as lições por ordem de dificuldade, estando os temas mais fáceis no início do semestre. Os processos de ensino também sofreram alterações, sentido a necessidade de recorrer às tecnologias para facilitar o processo de ensino: desde a comunicação entre alunos, até à troca de recursos relativos às aulas. Esta melhoria foi conquistada com a publicação online dos recursos digitais das aulas (eg. vídeo), disponibilização de tutoriais de programação, um fórum anónimo e consultório do docente virtual para comunicação após os períodos de aula[25].

Os resultados obtidos foram visivelmente positivos, nos 3 anos que se seguiram o número de inscrições no CS50 aumentaram cerca 150%, assim como o número de alunos a terminar com sucesso, estas alterações mudaram a imagem do curso no *Campus*, e isso refletiu-se em cursos adjacentes, CS51 (uma especialização após o CS50 focada em Lisp) e o CS1 (um curso de preparação para o CS50) também aumentaram as suas inscrições[25].

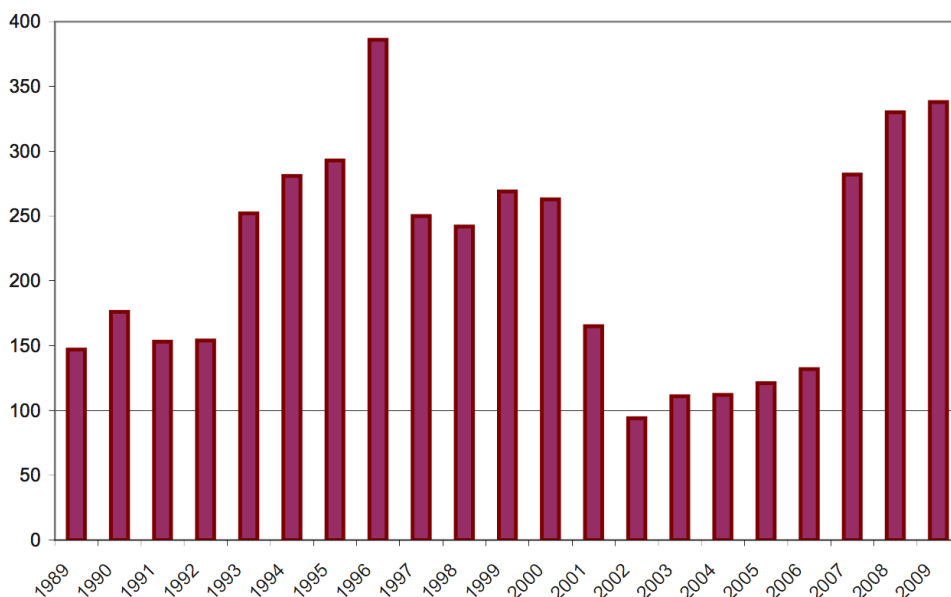


Figura 2.7: Inscrições na disciplina CS50 entr 1989 e 2009[25]

<sup>11</sup><https://cs50.harvard.edu/>



Em 2008, logo após o sucesso da primeira experiência, os responsáveis pelas inovações do curso quiseram mover toda a infraestrutura para a *cloud* por 3 motivos: 1) controlo da infraestrutura sem limite de período de atendimento por parte do departamento de Tecnologia da informação (TI) 2) acessibilidade e acompanhamento do trabalho dos alunos 3) possibilidade de adicionar novos tópicos (escalabilidade, virtualização, processamento *multi-core* e *cloud computing*) ao *syllabus*[45].

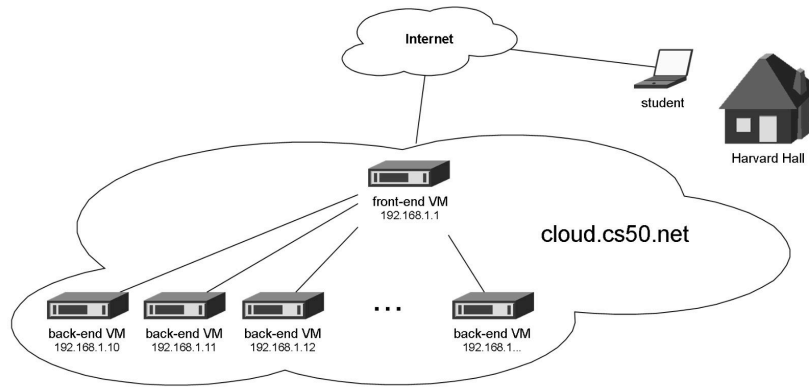


Figura 2.8: Arquitetura do CS50 em formato de Cloud[45]

O serviço de *cloud* usado, foi o Elastic Compute Cloud (EC2) da Amazon Web Services (AWS), como se pode ver na imagem anterior, a arquitetura possui uma máquina que serve de fronteira à rede "*backend*", esta máquina dispõe de Lightweight Directory Access Protocol (LDAP) e Network File System (NFS), e depois redireciona o tráfego Secure Shell (SSH) para o destino `cloud.cs50.net` para a instância de *backend*. O objetivo é disponibilizar um ambiente de desenvolvimento para que os alunos possam resolver e executar os problemas propostos. Como é identificado, não existe a necessidade de dedicar uma instância por aluno, existe sim um volume Elastic Block Storage (EBS) por alunos, que será a sua *home* e aonde terá ao seu dispor ferramentas como `gcc`, `gdb` e `valgrind`. As instâncias *backend* são de número limitado e os volumes estão diretamente ligados à instância *frontend*, e que são exportados à instância de *backend* que o aluno estabelece a ligação (ver Figura 2.8). Para além disso, utilizam o serviço Simple Storage Service (S3) para alojar as *snapshots* criadas por `rsync` e `rsnapshot`. Uma transformação que resultou em sucesso, porque os objetivos iniciais foram atingidos (por exemplo, a autonomia). Alguns contra-tempos surgiram: 1) o tempo necessário para a provisão da rede foi elevado (estimam ~80 horas despendidas) 2) a meio do semestre o espaço de volume disponibilizado totalmente ocupado (originado por estimativas focadas na poupança) 3) aquando da aproximação da entrega de alguns trabalhos práticos, ocorreram *delays* no Input/Output (I/O) originados pelo OpenLDAP.

No final do semestre, a despesa nos serviços da AWS rondaram os \$5,000 o que fez um quantia \$15 por aluno, não esquecendo que no ano desta transformação a AWS já tinha programas de ajuda para a educação. Evidenciando assim, que o custo por aluno não foi resultado de algum acordo exclusivo para realização desta experiência.

No ano letivo seguinte, continuaram com a abordagem à *cloud*, mas já a pensar numa nova alteração, em disponibilizar o mesmo ambiente/cursos em formato *offline* através de tecnologias de Virtual Machine (VM), algo a que chamam "*Course in a Box*".

Em 2012, a mesma equipa voltou à investigação, lançaram o Computer Science 50 Edx Version (CS50x) o primeiro curso de Harvard a ser transformado em MOOC ao abrigo de um programa de ensino à distância[46][47]. O primeiro semestre em modo MOOC, o curso teve ~150,000 inscrições que resultaram em ~11,000 submissões no primeiro trabalho e terminou com ~1,400 certificados entregues. É provável que a causa para que o número de inscrições *versus* o número de certificados sejam muito diferente, devido à inscrição/participação não ter custos, enquanto o certificado tem um custo associado aquando do pedido de emissão.

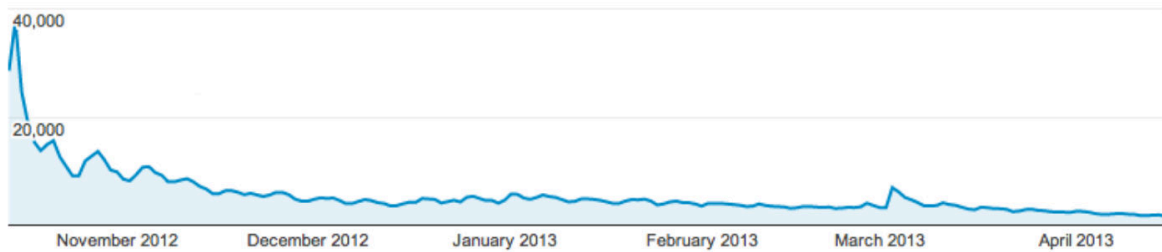


Figura 2.9: Registo das visitas únicas diárias pelo Google Analytics[48]

No gráfico da Figura 2.9, é visível que relativamente à atividade diária houve uma grande procura e que decresceu até às ~2,500 visitas diárias no final do semestre. Provavelmente, o pico inicial deveu-se ao fato de um dos melhores cursos de Harvard ter se tornando público, uma novidade que atraiu efetivamente pessoas interessadas, assim como também muitos curiosos (que podem não ter sentido confiança para avançar no programa).

Esta suposição é fundamentada por resultados originados num inquérito. No gráfico da Figura 2.10 é possível constatar o interesse por parte dos participantes, o objetivo por detrás da inscrição: certificação *versus* conhecimento.

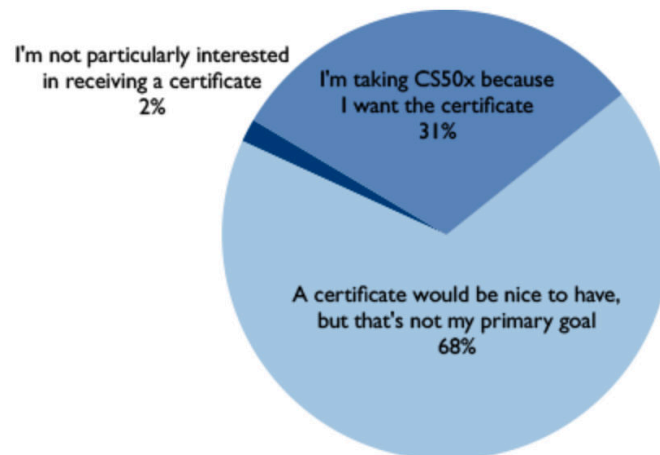


Figura 2.10: Objetivos principais na origem da inscrição[48]

Uma última evolução do curso, ocorreu no em 2015 e onde as conclusões serão apresentados em 2017 numa *workshop*/conferência, consiste na disponibilização de um ambiente semelhante ao que foi fornecido anteriormente e a utilização de tecnologias *web* e *cloud*, disponibilizando um Integrated Development System (IDE) acedido por um *browser* libertando o utilizador de dependências e instalações.

O Cloud9<sup>12</sup> é um IDE suportando por tecnologias de *cloud*, nomeadamente Docker, permite ter um ambiente semelhante ao que era obtido pela instância EC2 com a diferença está mais preparado devido a possuir um IDE e uma Graphical User Interface (GUI), que permite facilmente instalar *plugins* para melhorar a experiência de utilização. Para além disso também é dotado de ferramentas como *command-line* para interagir com o sistema operativo ou sistemas de ficheiros da instância, o *debugger* e os *breakpoints* permitem estudar o código para a resolução de problemas[49].

De todas as evoluções do CS50, a transição para a *cloud* e aceitação da *web* como intermediário de ambientes de desenvolvimento, aproxima-o ao objetivo desta dissertação. Esta semelhança, advém pelo facto da Cloud9 estar em cooperação com algumas instituições do ensino superior para servir de ferramenta de suporte, mas esta dissertação destaca-se na medida em que procura também desenvolver uma funcionalidade focada na distribuição de conteúdo teórico e prático num formato de consolas interativas, algo que até ao momento no CS50 ainda recorre aos vídeos e outros formatos digitais do estilo de documentos.

---

<sup>12</sup><https://c9.io>

## 2.5.5 CODIO

Uma plataforma de *E-Learning* focada para disciplinas de Ciências de Computação que permite que seja utilizada por indivíduos ou por turmas/instituições. Como indivíduo é possível o acesso a um repositório de classes que sejam públicas para a comunidade (ver Figura 2.11), como aluno de uma instituição está sujeito às classes a que está inscrito.

A Codio<sup>13</sup> é uma plataforma que recorrendo a tecnologias de *cloud* e de *web* disponibiliza um IDE que permite a criação de projetos com base em *templates* pré-definidos pela plataforma, esta funcionalidade permite que o aluno possa desenvolver os trabalhos que terá de submeter para avaliação.

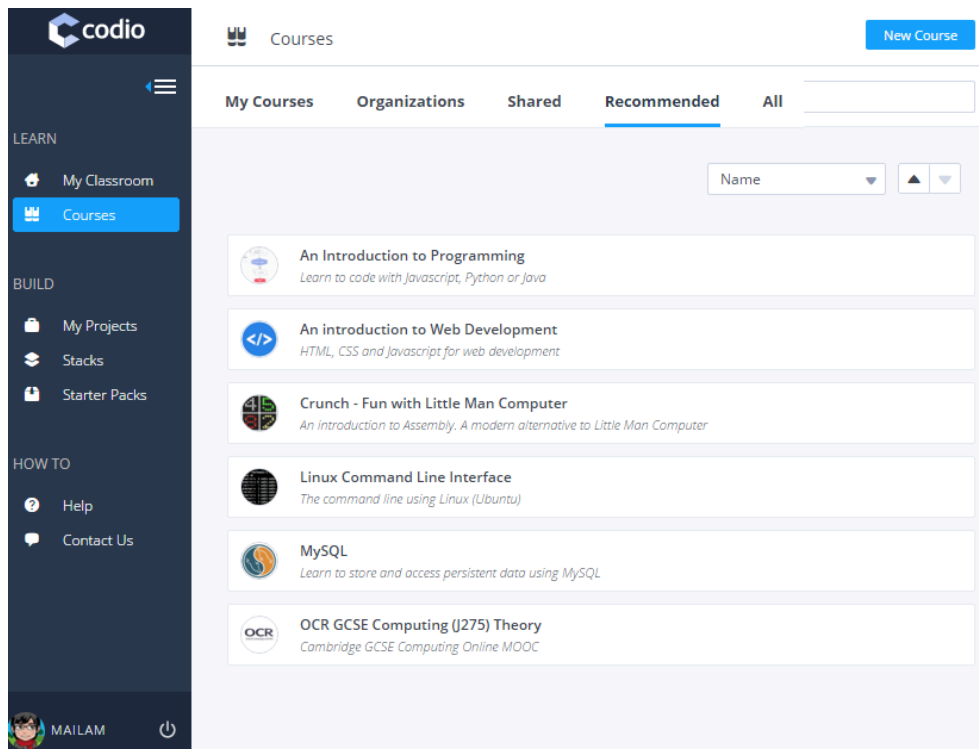


Figura 2.11: Lista de cursos recomendados na plataforma Codio[50]

Sendo uma plataforma de *E-Learning*, o docente tem acesso às funcionalidades de gestão que tem noutras plataformas como: ver o progresso e o trabalho do aluno, ativar funcionalidades de análise e avaliação automática dos trabalhos entre outras tarefas.

Esta plataforma permite ao docente criar aulas interativas com base no IDE, em que a cada capítulo o aluno tem matéria nova e acesso para interagir com o código em causa, de forma a melhor aprender a matéria.

O Codio como plataforma *E-Learning* teve uma evolução próxima do momento em que foi proposta esta dissertação, onde grande parte das funcionalidades desta plataforma são concorrentes ao que se pretende desenvolver na dissertação. Nos últimos avanços, a plataforma passou a ser compatível com plataformas LMS existentes, como o MOODLE e Blackboard, facilitando assim as instituições que já dependiam dessas plataformas para gestão de turmas, alunos, conteúdos, notas, etc. Como desvantagem, este serviço não disponibiliza uma instalação do sistema *on-premise*, um cenário que permite que a instituição tenha o controlo da manutenção e da gestão da infraestrutura.

<sup>13</sup><https://codio.com/>

## 2.5.6 NIXTY

O NIXTY é um LMS, e surgiu com o objetivo de se diferenciar dos outros na medida em que a sua arquitetura iria se basear em alguns *standards*. Nomeadamente, os *standards* da *Web 2.0* para alavancar a *Open Education*. Desta forma, recorreu às tecnologias OpenSocial, OpenID<sup>14</sup> e OpenCourseWare para desenvolver um novo LMS.

O OpenSocial é uma especificação desenhada em 2007 para permitir interoperabilidade entre redes sociais, um projeto que inicialmente foi desenvolvido pela Google juntamente com o MySpace<sup>15</sup>[51]. O objetivo era a criação de uma Application Programming Interface (API) modular que permitisse os programadores criarem aplicações sociais mais ricas através da integração de diferentes comunidades[52]. Em 2014, o World Wide Web Consortium (W3C) percebeu que esta especificação tinha potencial, tendo em conta nessa altura a sua agenda se focava no desenvolvimento de uma *Open Web Platform*[53][54], como dependia fortemente de tecnologias sociais, as duas instituições envolvidas decidiram que o melhor para o futuro dos *standards* da *Social Web* estaria nas mãos de uma instituição maior como a W3C[55].

Numa altura em que a *web* estava a evoluir e o número de aplicações *web* também estava em ascensão, e em que na sua grande maioria precisavam de credenciais de autenticação, um protocolo de autenticação foi desenvolvido, o OpenID, a ideia por detrás deste *standard* era a criação de um sistema descentralizado de autenticação, permitindo que os consumidores da *web* possam ter os *websites* e aplicações centradas numa única entidade, eliminando o cenário de um par de credenciais para cada aplicação, e assim possibilitar a partilha de uma entidade entre todas as aplicações de que fosse cliente[56].

O OpenCourseWare é um conceito similar ao de MOOC, por definição, é a disponibilização via *Internet* dos diferentes conteúdos que constituem um curso, este movimento teve início em meados dos anos 1990 na Universidade de Tübingen, mas só se tornou tendência quando o MIT decidiu criar a plataforma de *OpenCourseWare*[57].

---

<sup>14</sup><https://openid.net>

<sup>15</sup><https://myspace.com/>

The screenshot shows the Nixty platform interface. At the top, there is a navigation bar with links for Home, Courses, Eportfolio, Mail, Blog, Calendar, Contacts, Settings, and Logout. The main content area is titled 'Introduction to Biology' and features a 'Like' button and a social media share icon. Below this, there are sections for 'Announcements', 'Lessons', and 'Discussions'. The 'Lessons' section contains a table with the following data:

Lesson	Progress	Percentage
Lesson 1. Introduction	4 / 7	(57%)
Lesson 2. Biochemistry 1	1 / 4	(25%)
Lesson 3. Biochemistry 2	0 / 6	(0%)
Lesson 4. Biochemistry 3	0 / 6	(0%)
Lesson 5. Biochemistry 4	0 / 6	(0%)
Lesson 6. Genetics 1	0 / 6	(0%)
Lesson 7. Genetics 2	0 / 6	(0%)
Lesson 8. Genetics 3	0 / 6	(0%)
Lesson 9. Human Genetics	0 / 6	(0%)

The 'Discussions' section lists several topics with their authors and the number of responses. The right sidebar contains an 'About This Course' section with a description of the course, an instructor's name (Robert A. Wenberg), and a link to donate to MIT. There is also an 'Instructors' section at the bottom of the sidebar.

Figura 2.12: Lista de cursos recomendados na plataforma Nixty[58]

A utilização destas tecnologias como OpenSocial pretende melhorar o processo de aprendizagem através da reutilização de comunidades sociais já existentes de forma a promover relações e novos paradigmas (recorrer a jogos sociais é uma forma de aumentar o envolvimento dos alunos). O OpenID é uma plataforma poderosa para vários cenários de *Open Education*, com disseminação de diferentes plataformas de *E-Learning*, a criação de uma entidade única é favorável ao processo de conhecimento porque permite aos utilizadores juntarem-se a diferentes comunidades mais facilmente sendo que o maior impacto é a criação de um perfil, que facilmente serve em pleno todas as partes envolvidas aquando da certificação do utilizador possibilitando melhor comunicação entre instituições de ensino e os empregadores. O OpenCourseWare possui o fundamento básico da *Open Education* a promoção e o acesso fácil ao conhecimento, podendo facilmente criar sinergias entre diferentes instituições, melhorando o ensino entregue pelos docentes (que se podem ajudar entre si para a melhorar a qualidade do conteúdo) como a aquisição de conhecimento é mais fácil devido ao acesso “ilimitado” ao conhecimento e aos cursos por parte dos alunos.

Com base nas tecnologias anteriores, o NIXTY é uma plataforma que permite às instituições criarem o seu próprio LMS ou cursos individuais. As funcionalidades disponíveis pelo NIXTY são as já conhecidas pela definição de LMS, com a diferença que são orquestradas com base nos *standards* anteriores, isso permitiu criar uma nova funcionalidade, um sistema de reputação, o *ePortfolio*, no qual permite aos utilizadores da plataforma o registo do progresso e resultados ao longo dos cursos. Isto torna-o diferente de outras alternativas na medida que participar em cursos “não-acreditados” já não é uma tarefa exclusiva de aquisição de conhecimento, com esta solução pode ter alguma afetação no perfil/*curriculum* do individuo[59].

Atualmente, esta plataforma parece estar indisponível, o seu endereço anteriormente utilizado por eles agora aloja um página estática com conteúdo em nada relacionado com o suposto.

## 2.6 SOLUÇÕES SELF-HOSTED

Ao contrário da secção anterior, esta analisa soluções que foram desenvolvidas em âmbito académico ou que foram desenvolvidas para serem *self-hosted*, isto é, as instituições podem instalar estas soluções nas suas infraestruturas e depois configurá-las de acordo com as necessidades dos seus utilizadores. Também existe uma solução que é disponibilizada através da instalação direta no equipamento do aluno. Nesta secção, serão avaliadas plataformas que permitam ser instaladas e disponibilizadas num infraestruturas privada, a maioria destas soluções apresentadas são muito semelhantes porque focam-se fundamentalmente no suporte ao ensino de linguagens de programação ou avaliação de exercícios de programação, com a exceção do MOODLE que não se foca numa única área do ensino.

### 2.6.1 MOODLE

A análise do MOODLE neste documento deve-se principalmente por ser o LMS usado pela UA, mas também por ter uma comunidade grande [60] e por ser um dos LMS mais conhecidos[61], possivelmente é o resultado da proliferação dos sistemas de *E-Learning*.

A plataforma surgiu no âmbito da tese de doutoramento do Martin Dougiamas em parceria com Peter Taylor, e que desenvolveu a primeira versão do MOODLE, com o intuito de ser uma ferramenta de suporte ao ensino à distância, e à descoberta de novos paradigmas na pedagogia, ainda longe do atual LMS. Em 2003, Martin e Peter, possuíam um CMS usado por várias centenas de instituições e docentes[13][61].

No início a plataforma era um CMS que permitia organizar unidades de ensino, do qual disponha de um repositório de documentos e de fóruns para promover a comunicação entre alunos e docentes[13].

Atualmente é um LMS de referência, porque para além de ser uma solução *open source* é também caracterizado pela sua interface simplista, de baixa curva de aprendizagem[24][61] e flexibilidade perante as funcionalidades disponíveis e pela comunidade responsável por desenvolver funcionalidades extra à instalação base, dispõe de mais funcionalidades: i) colaborar com outros membros a partir de fóruns/*wikis/chat* ii) avaliar os alunos com *assessments/quizzes* iii) interagir com a comunidade estudantil a partir de *surveys* iv) acompanhar a evolução do aluno através de um *Gradebook* v) gestão de utilizadores e cursos vi) personalizar as interfaces vii) suporte de *standards* para *Achievements* (Mozilla OpenBadges<sup>16</sup>). Existem muitas mais funcionalidades disponíveis a partir da instalação base e se considerarmos os *plugins* desenvolvidos por terceiros esse número aumenta muito mais, mas só foram apresentadas algumas para manter a descrição deste LMS concisa.

A comunidade em torno do MOODLE é uma mais valia na plataforma e um suporte no desenvolvimento (por exemplo: auxílio nas traduções) e também para identificar ou resolver problemas. O MOODLE possui um sistema de *plugins* que disponibiliza uma interface aos *developers* para acrescentar funcionalidades extra à distribuição base da plataforma, esta comunidade, parcialmente adjacente à comunidade Open-Source Software (OSS), procura entre muitas outras atividades desenvolver e publicar as suas contribuições que ficam disponíveis para os restantes utilizadores, numa espécie de *marketplace*<sup>17</sup>. E como é comum em comunidades *onlines* existem fóruns, sistemas de classificação e comentários, tudo ferramentas que permitem à comunidade fornecer a sua opinião, ou notificar problemas em cada submissão no *marketplace*.

---

<sup>16</sup><http://openbadges.org/>

<sup>17</sup><https://moodle.org/plugins/index.php>

O MOODLE não se dedica a nenhum setor do ensino em específico, as suas funcionalidades são genéricas, e como tal é limitado para conteúdos como algumas disciplinas de Ciências de Computação. Como exemplo, a instalação do MOODLE usada na UA para as Unidades Curriculares de programação uma dessas funcionalidades é a submissão dos conteúdos teóricos, e dos exercícios em formato digital (PDF), outras funcionalidades para auxílio das UC de Programação obtidas por instalação de plataformas de terceiros (Mooshak) ou desenvolvimento de soluções internas (ver Subseção 2.6.2).

Na análise realizada focada na utilização do MOODLE num cenário de ensino de disciplinas de programação, foi encontrado um contributo desenvolvido por membros da comunidade que implementa uma funcionalidade de avaliação automática, baseado em conceitos de *Unit Testing*. Do qual o conceito base resume-se à inserção de um valor (*input*) sujeito a uma operação e quando terminar verificar se o resultado (*output*) é igual ao resultado esperado.

Na Figura 2.13 o aluno tem acesso às perguntas e indicações que o docente forneceu por questão, e associado a cada questão tem um bloco onde pode introduzir a sua resposta.

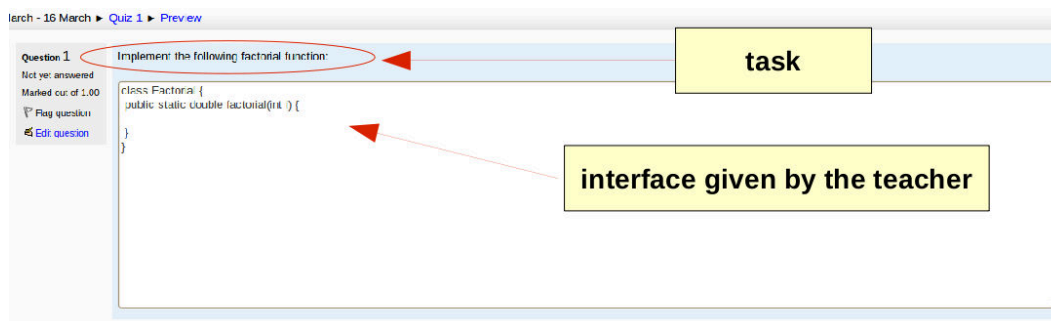


Figura 2.13: Exemplo de pergunta com o MOODLE *plugin* “javaunittest”[62]

Após a submissão dos trabalhos dos alunos, com o auxílio da *framework* JUnit, o MOODLE executa as respostas e avalia consoante o resultado, através das respostas inseridas pelo docente resultando na classificação automática da questão, avaliando assim o aluno de forma autónoma.

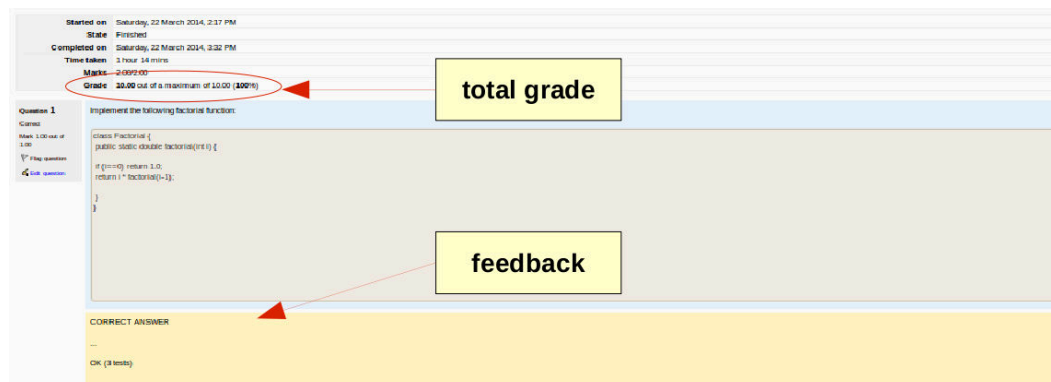


Figura 2.14: Exemplo de uma classificação com o MOODLE *plugin* “javaunittest”[62]

O *plugin* denomina-se “javaunittest”[62] baseado num sistema de questionário (em que o docente adiciona perguntas e valida-as com a resposta que é esperada) dedicado à avaliação de conhecimentos em Java.



## 2.6.2 SAAL

Ao contrário de outras plataformas, orientadas a competições e a validação automática de exercícios de programação, o SAAL[5] resulta do esforço de um aluno e seus orientadores para a evolução de uma plataforma utilizado nas aulas de programação, a Mooshak<sup>18</sup>, uma plataforma mais focada na submissão de exercícios e na respetiva classificação do resultado (nota), sem qualquer funcionalidade ao nível da gestão de cursos, turmas entre outras funcionalidades, que facilmente fazem parte do conceito de outras plataformas de *E-Learning*.

A criação do SAAL permitiu a inclusão de funcionalidades básicas de *E-Learning* esquecidas por algumas plataformas concorrentes. Em determinados pontos de vistas, este projeto pode não ser considerado um LMS porque não suporta *standards*, como por exemplo, Sharable Content Object Reference Model (SCORM) nem estão implementadas funcionalidades, como por exemplo, autenticação, gestão de alunos e disciplinas não estão implementadas [14]. Em vez disso, estas funcionalidades são consumidas num paradigma de colaboração com outros sistemas, por exemplo, para autenticar um utilizador a plataforma SAAL consome um serviço de identidade onde recolhe informação das disciplinas e turmas e cria um perfil local para gerir as operações internas. Pela análise efetuada pelo colega Pedro Estima[5], as outras plataformas com funcionalidades de *E-Learning* não tinham capacidade para executar programas, nem classificar o seu resultado, e as que tinham não possuíam funcionalidades de gestão de turmas (um dos fundamentos das plataformas de *E-Learning* mais básicos).

Desta forma, foi desenvolvido o SAAL, uma plataforma Service Oriented Architecture (SOA) que tira proveito do LMS já utilizado pela instituição para efeitos de autenticação, gestão de alunos, turmas e trabalhos práticos, e acrescenta um serviço para submissão e classificação de código.

Como é possível ver na Figura 2.15, a plataforma é composta por macro-componentes: a aplicação *web*, os *webservices* e o ambiente de execução. A aplicação *web* é a interface de acesso dos alunos e dos docentes às funcionalidades de gestão, submissão e avaliação, os *webservices* consistem na abstração da base de dados e das operações privadas que permite lidar com a gestão de turmas, alunos e docentes e respetivas relações, e também abstrai a interação com o componente de execução. No que diz respeito ao acesso à plataforma, requer que o aluno/docente esteja devidamente inscrito na instituição devido a utilizar o serviço dedicado de autenticação e autorização, denominado por Identity Provider (IdP).

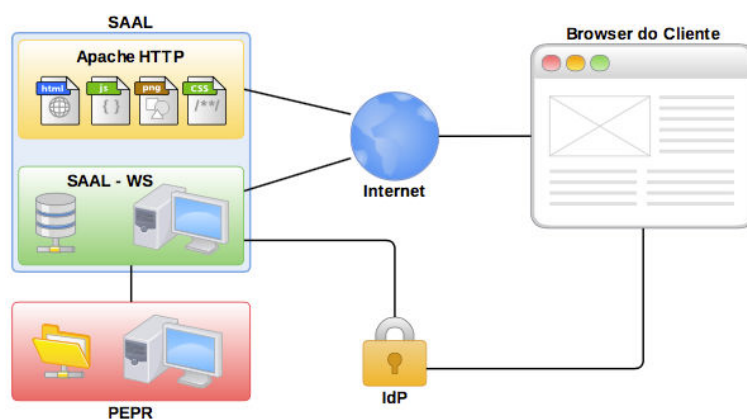


Figura 2.15: *Screenshot* da arquitetura do SAAL[5]

<sup>18</sup><http://mooshak.dcc.fc.up.pt/>

Como ferramenta de suporte à avaliação, o projeto teve um resultado bastante positivo, o docente ganha um ferramenta que lhe permite realizar avaliações melhores e mais flexíveis para os seus alunos. E como foi abordado na dissertação do SAAL, as características da plataforma desenvolvida além de focar avaliação automática mas também pode ser utilizada durante as aulas de forma a que os alunos tenham um sistema em que possam estudar e submeter as várias resoluções de uma “folha de exercícios” e receber *feedback* sem ter que esperar por uma correção do docente. Estas funcionalidades têm semelhanças com plataformas de competição de código, uma vantagem do sistema, porque se assim a instituição e o seu corpo docente desejarem podem realizar competições de programação, e se possível gerar mais envolvimento dos alunos na aprendizagem das temáticas lecionadas como preparação para essas mesmas competições.

Apesar de prestar auxílio ao momento de avaliação, às aulas ou até em atividades extracurriculares, esta plataforma não suporta o acompanhamento do aluno na perspetiva do docente, algo importante para identificar as dificuldades que alunos estejam a ter ou para ter conhecimento de como os alunos assimilam toda a matéria. Com o SAAL, o acompanhamento do docente é feito tendo acesso ao estado/resultado do exercício depois da submissão do aluno é possível inferir as suas dificuldades com base na nota ou métricas como o tempo de execução, o *feedback* ocorre no final do processo. Integrar conceitos de colaboração, como sugeridas no *E-Learning 2.0*[17] iria permitir ao docente acompanhar os seus alunos aquando o uso destas plataformas na realização de trabalho ou acompanhar os seus estudos, algo que o SAAL ainda não permite.

Quanto a trabalhos de grupos, normalmente mais complexos tanto no problema, na resolução como na dimensão, o SAAL não suporta esse conceito. A avaliação automática de um trabalho de maior dimensão e complexidade é difícil de obter, mas o acompanhamento ao trabalho e aos alunos por parte do docente fora da sala de aula pode ser obtido se houver um ambiente remoto partilhado, algo que é proposto nesta dissertação.

### 2.6.3 PBEL

Golam Bashir e Abu Hoque[63] abordaram este tópico à semelhança dos responsáveis do CS50 e identificaram as dificuldades inerentes ao ensino de temáticas nas Ciências de Computação, mais precisamente, o foco deles é lecionar linguagens de programação. Após investigarem chegaram a conclusão que o ensino de linguagens de programação nos atuais paradigmas é uma tarefa difícil. Existe um medo por parte dos alunos quando estão inscritos a estas disciplinas, e as estatísticas demonstram que este medo se transforma em algo mais sério, quando existem desistências entre os 30% e os 50%. Como Golam e Abu identificam no seu estudo[63], os atuais métodos de ensino: a) não promovem o envolvimento dos alunos b) não fornece aos alunos a devida preparação para serem auto-didatas a longo prazo c) estes métodos incluem que os alunos assistam a palestras ou aulas presenciais. De notar, que neste último ponto, os autores afirmam que cativar a atenção do aluno pode ser um problema porque a atenção dos mesmos tem uma relação inversa com a duração da aula/palestra.

A abordagem proposta é que através de um método pedagógico já bastante usado na área e nos atuais paradigmas de ensino, denominado por Problem-Based Learning (PBL), com o intuito de aumentar a qualidade de formação e empenho dos alunos. Resume-se à aproximação do mundo real nos problemas/exercícios/trabalhos disponibilizados aos alunos, um método que necessita que os alunos tenham ou desenvolvam aptidões, conhecimento e estratégias para a sua resolução, resultando na aquisição de experiência valiosa para a aprendizagem do mesmo.

Os autores da experiência repararam que PBL como método tinha melhor performance que outros métodos tradicionais, mas dada a sua natureza não é inteiramente adequado para a aprendizagem de linguagens de programação como também acontece com as plataformas de *E-Learning* em que na sua grande maioria das funcionalidades são focadas nos métodos de ensino tradicionais e não à prática e estudo das linguagens de programação.

Deste modo, surgiu a oportunidade em fundir os dois conceitos e desenvolver o Problem-Based e-Learning (PBeL)[63] e tirar partido das vantagens de ambos domínios para proporcionar um novo sistema, no qual permita aos alunos o desenvolvimento de competências adquiridas através da resolução de problemas (reais) e com a flexibilidade de um sistema acessível em qualquer lugar e em qualquer altura. Este sistema mitiga as falhas por eles antes encontradas, fomenta a auto-aprendizagem e a criação de métodos de estudo e estratégias para a resolução de problemas.

A solução desta equipa tem como fatores-chave o desenvolvimento de um modelo para a criação de problemas e salvaguardá-los num repositório próprio para o efeito. A implementação da solução é representada na Figura 2.16.

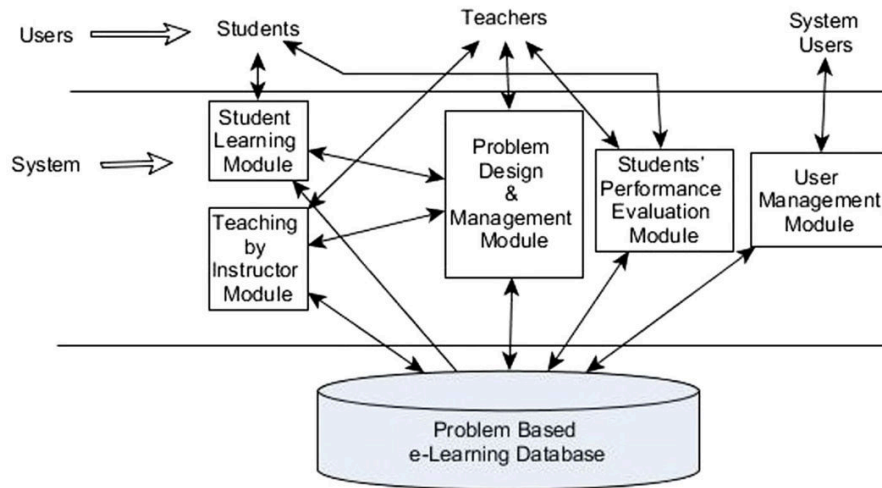


Figura 2.16: Arquitetura da plataforma PBeL[63]

## MÓDULO DE CONCEÇÃO E MANUTENÇÃO DE PROBLEMAS

Este módulo (na Figura 2.16 denominado por *Problem design and management module*) com maior relevância da plataforma porque é o centro da solução, é a partir dele que docentes tem a capacidade de definir um problema e enumerar a respetiva solução que serão salvaguardados no repositório.

## MÓDULO DE APRENDIZAGEM PARA ALUNOS

Este módulo é interface para os alunos (na Figura 2.16 denominado por *Student Learning Module*) serve os alunos, de forma a que possam escolher que problemas resolver, ter acesso aos materiais necessários para a resolução do mesmo e por fim submeter a sua resolução para avaliação.

## MÓDULO DE ENSINO PARA DOCENTES

À semelhança do módulo anterior é também uma interface mas direcionada aos docentes, a partir deste (na Figura 2.16 denominado por *Teaching by Instructor Module*) permite que tenham a capacidade de gerar os problemas pelo método enunciado no primeiro módulo, e depois fazer manutenção e seleção de problemas para cenários de avaliação. Existe a possibilidade de avaliação manual das resoluções submetidas pelos alunos para cenários em que a avaliação automática não está disponível.

## MÓDULO DE AVALIAÇÃO DE ALUNOS

A partir deste módulo (na Figura 2.16 denominado por *Students' performance evaluation module*) o docente pode criar trabalhos ou testes e atribuí-los aos alunos sujeitos no momento de avaliação. Como esclarecido anteriormente, o docente cria um problema com a respetiva solução, e através deste módulo quando o aluno participa no momento de avaliação e submete a sua proposta de resolução, o sistema de forma autónoma avalia e classifica a resolução do aluno. Este módulo resulta diretamente numa descida de carga ao que os docentes estão sujeitos a cada momento de avaliação.

## MÓDULO DE GESTÃO DE ALUNOS

Esta plataforma está isolada de outras plataformas de *E-Learning*, desta forma a gestão administrativa da plataforma não pode ser delegada a “outros”, resulta na implementação de um módulo (na Figura 2.16 denominado por *User management and administrative module*) que permite aos responsáveis da plataforma gerir os alunos, docentes e respetivas permissões.

## REPOSITÓRIO DE PROBLEMAS

Este módulo (na Figura 2.16 por *Problem Based e-Learning Database*) resume-se ao alojamento de problemas e respetivos conjuntos de soluções, tem a particularidade de permitir flexibilidade ao nível da linguagem de programação, isto é, o repositório permite a distinção de problemas por linguagem.

A flexibilidade é atingida com apoio dos módulos de Módulo de Aprendizagem para Alunos e Módulo de Ensino para Docentes que no momento de acesso permite pesquisar e classificar pela linguagem de acesso, separando assim os problemas no repositório.

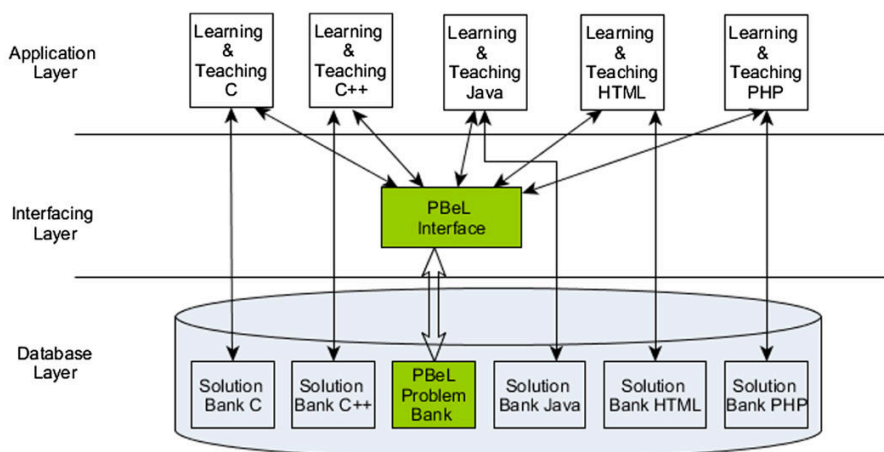


Figura 2.17: Arquitetura mais detalhada da plataforma PBeL

Após o desenvolvimento, os responsáveis disponibilizaram a plataforma aos alunos, e realizaram um inquérito para averiguar a usabilidade e qualidade das interações entre utilizador e plataforma, e em última instância classificarem o novo método pedagógico. Os resultados[63] obtido foram tido como positivos:

- 80% satisfeitos com a plataforma (conteúdos e funcionalidades)
- 85% acredita que foi extremamente útil durante a aprendizagem
- 85% afirma que o novo método melhorou as suas competências para resolução de problemas
- 90% considera que a nova plataforma fará parte do processo de aprendizagem

Esta plataforma demonstra uma evolução na área das ferramentas de *E-Learning* focadas à informática. Quanto a objetivos ambiciona facilitar a aprendizagem de linguagens de programação através de um processo de resolução de problemas, algo que está inerente a disciplinas de introdução (como pode ser visto em algumas parte do programa do CS50). E pelos resultados apresentados nesta secção, o PBeL é capaz de criar uma ótima ligação com os alunos, estando no final mais aptos tanto quanto à resolução de problemas quanto à linguagem de programação.

O projeto PBeL em comparação com esta dissertação possui objetivos semelhantes, que consiste na distribuição de novas ferramentas para auxiliar o ensino das linguagens de programação. A abordagem por parte de cada projeto é que é diferente nomeadamente no que diz respeito ao suporte da sintaxe da linguagem de programação, a abordagem do PBeL foca a resolução de um problema e a sucessiva realização de vários exercícios auxilia a aprendizagem da sintaxe. No caso desta dissertação a abordagem consiste na disponibilização de um formato com consolas interativas permite que o aluno possa aprender a sintaxe da linguagem e também os conceitos teóricos e não focar em problemas específicos. Também nesta dissertação existem os ambientes de desenvolvimento, algo que difere no resultado, porque estes estão desenhados para projetos de grupo e/ou problemas mais complexos, como a criação de sistemas distribuídos ou aplicações *web*, algo que o PBeL não está preparado para disponibilizar.

## 2.6.4 PYCHARM EDU

A JetBrains<sup>19</sup> é uma empresa de referência, devido a produzir alguns dos melhores IDE do mercado. Possui vários IDE para as linguagens mais utilizadas: Java, PHP, Python, Ruby e C. Em 2010, a empresa decidiu também contribuir para o setor do *E-Learning*, desenvolvendo um IDE transformado para auxiliar a aprendizagem de Python, denominado PyCharm Edu<sup>20</sup>. Na última versão, o software permite realizar cursos definidos pela comunidade que estão disponíveis publicamente num repositório da JetBrains (publicados através do próprio IDE), permite ainda criar cursos, e partilhá-los com os alunos por uma rede local ou pelo repositório público.

O conceito resume-se a consolas interativas, o curso é um conjunto de tarefas, cada tarefa tem uma descrição e (possivelmente) um código base, que o aluno terá de corrigir e complementar para que quando executado retorne um resultado esperado (anteriormente definido pelo criador do curso). Tira partido de algumas funcionalidades de um IDE, nomeadamente no que respeita a ferramentas de *debug* (como os *breakpoints*).

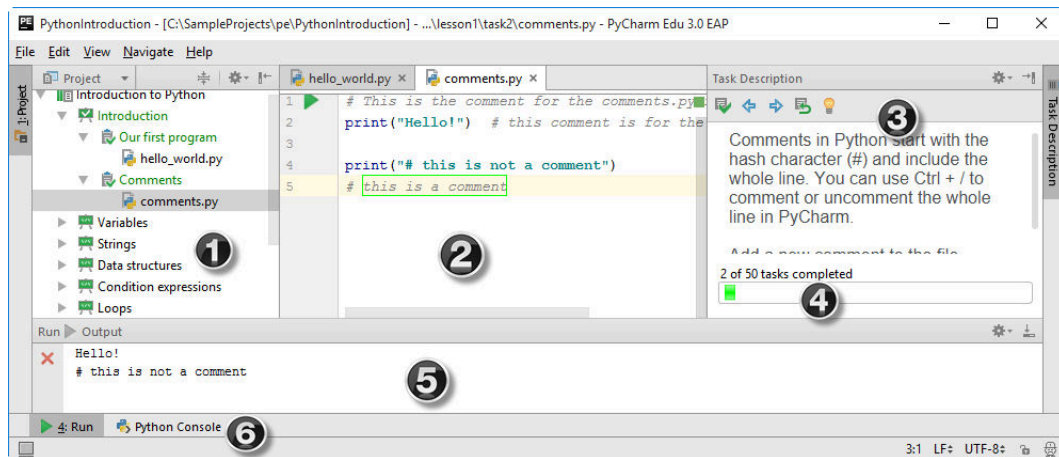


Figura 2.18: Screenshot do curso “Introduction to Python”[64]

Como é possível ver na Figura 2.18, disponibiliza um ambiente onde o utilizador conhece o conteúdo do curso (ver ponto 1), tem ao seu dispor um editor já com algumas pistas desde código por preencher e até comentários (ver ponto 2), por cada tarefa existe uma descrição e a teoria associada à mesma (ver ponto 3), o aluno pode acompanhar o seu progresso (ver ponto 4) e por fim submeter o seu trabalho para avaliação (ver pontos 5 e 6).

Verificam-se algumas semelhanças com outras plataformas (MOODLE, SAAL e Codecademy) na utilização de alguns métodos de aprendizagem como: blocos de código por preencher, comparação do *output*, etc, com a diferença que é distribuído como aplicação *desktop* e inteiramente focado na linguagem Python.

Comparativamente ao que se pretendeu desenvolver nesta dissertação, em que o objetivo é fornecer uma funcionalidade semelhante e se possível com um processo mais acessível para a criação e a distribuição dos exercícios com base em tecnologias/serviços *web*.

<sup>19</sup><https://www.jetbrains.com>

<sup>20</sup><https://www.jetbrains.com/pycharm-edu>

## 2.7 FERRAMENTAS

Nesta secção final, a análise foca-se em ferramentas para possível integração na fase de desenvolvimento das funcionalidades propostas. As ferramentas foram estudadas e divididas de acordo com a sua utilidade. Em primeiro lugar serão apresentadas ferramentas para disponibilizar ambientes de desenvolvimento com base em tecnologias *web* e/ou de *cloud* para permitir a criação de aplicações a partir de um *browser*. Por fim, vem o estudo a novos formatos de conteúdo para que seja possível expor a matéria teórico-prática de forma mais dinâmica e que permita interação do aluno com o conteúdo lecionado.

Quando a proposta de dissertação foi submetida aos orientadores a ideia das funcionalidades principais declaradas na proposta tinham como base dois projetos *open-source* (ver em mais detalhe no Capítulo 4).

Esta análise aconteceu em duas fases, a primeira consistiu em procurar soluções que permitam criar os ambientes de desenvolvimento *on-demand*, estes ambientes de desenvolvimento consistem na disponibilização de diferentes ferramentas usadas para o desenvolvimento de programas aglomeradas num *bundle*. Nos vulgos, editores de texto ou IDE, algumas destas ferramentas costumam ser identificadas como *syntax highlight*, gestão de projetos, *debugging/breakpoints* entre outras.

Entre as várias alternativas pesquisadas, o requisito que detem maior relevância que deve estar incluído neste *IDE-in-a-box* é ser desenvolvido em tecnologias *web* para que possa ser consumido por um *browser*. Os outros requisitos serão classificados como extras, e caso seja possível, os que melhor sirvam os alunos e que estejam mais próximos dos seus hábitos de trabalho.

A segunda fase, semelhante à primeira, consistiu em pesquisar um projeto que permita criar um ambiente de aprendizagem. Estes ambientes de aprendizagem deverão conseguir agregar os conceitos teóricos e práticos numa única interface, desta forma a relação do aluno entre a matéria teórica e prática pode ser agilizada, de maneira que o aluno tenha contacto com ambas na mesma interface.

Neste ambiente o aluno poderá adquirir conteúdo teórico e poderá testar o seu novo conhecimento através de elementos gráficos que permitam inserir código e testar os novos conceitos. À semelhança dos ambientes de desenvolvimento mencionados neste capítulo, o formato da distribuição destes ambientes de aprendizagem deverão estar disponíveis com base em tecnologias *web*.



### 2.7.1 AMBIENTES DE DESENVOLVIMENTO

Para a pesquisa de soluções dos ambientes de desenvolvimento, teve-se como preocupação a qualidade de escrita e execução de código, para cenários de exercícios ou desenvolvimento de projetos singulares ou em grupos de trabalho. Tendo em conta este cenário é necessário declarar parâmetros de classificação ou comparação, para isso foram eleitos tópicos que normalmente fazem parte dos planos curriculares lecionados nos diversos cursos do DETI. Esta lista (apresentada na Tabela 2.1) garante que os temas mais importantes seriam abrangidos e suportados pelo ambiente de desenvolvimento.

Aquando a experimentação das diferentes alternativas que serão abordadas posteriormente, os tópicos abordados nos exercícios práticos foram: a) operações de aritmética, porque muitas disciplinas de iniciação refugiam-se em exemplos básicos de aritmética e no funcionamento das calculadoras para a exposição de matéria; b) operações de acesso ao sistema de ficheiros (I/O), devido a várias disciplinas de nível de iniciação e intermédio; c) operações de gestão de base de dados para incluir as disciplinas dedicadas a temáticas como por exemplo Structured Query Language (SQL) ou Extract, Transform, Load (ETL); d) operações de multi-processamento (*threads*) para disciplinas de teor mais avançado normalmente associadas a temáticas de sistemas operativos, processos ou sistemas distribuídos e por fim e) operações de serviços *web* para as disciplinas dedicadas aos conceitos da *web* e dos serviços.

E para obter uma análise o mais isenta possível foi necessário que as várias opções analisadas fossem o mais diversificadas possível, nomeadamente, capazes de suportar diferentes linguagens de programação, sendo as mais comuns: Java, Python. Apesar de só existirem duas linguagens unânimes quanto à sua utilização nos diversos cursos, em algum momento os alunos acabam por utilizar outras, normalmente focadas na *web*: Ruby ou Javascript. Tendo em conta estas necessidades associadas às disciplinas, as linguagens eleitas para os testes práticos nos ambientes de desenvolvimento foram: Java, Python e Javascript(Nodejs).

A preparação da análise incluiu o desenvolvimento de alguns blocos de código (exercícios) com base nos pontos anteriormente referidos, por exemplo, a criação de diferentes blocos de código focado em operações aritméticas para ser executados em ambientes Java, Python e Nodejs. Pretende-se com esta ação tentar obter um termo de comparação entre cada alternativa e servir também como método para classificar de acordo com a sua performance e relativamente aos requisitos impostos. O bloco de código 1 é um dos diferentes blocos que foram criados:

```
# 3rd code: database CRUD operations (MySQL and Redis)

connection = pymysql.connect(host='localhost', user='root', password='',
                             database='aurora',
                             cursorclass=pymysql.cursors.DictCursor)

try:
    with connection.cursor() as cursor:
        sql = 'DROP TABLE IF EXISTS python'
        cursor.execute(sql)
        result = cursor.rowcount

        sql = 'CREATE TABLE python (`id` int, `acro` varchar(7), `description` varchar(55))'
        cursor.execute(sql)
        result = cursor.rowcount

        sql = 'INSERT INTO python (`id`, `acro`, `description`) VALUES (1, \'IP\', \'Internet
        ↵ Protocol\'), (2, \'HTTP\', \'Hyper-Text Transport Protocol\')'
        cursor.execute(sql)
        result = cursor.rowcount

        sql = 'UPDATE python SET description=NULL WHERE id=2'
        cursor.execute(sql)
        result = cursor.rowcount

        sql = 'DELETE FROM python WHERE id=1'
        cursor.execute(sql)
        result = cursor.rowcount
    connection.commit()
finally:
    connection.close()

client = redis.StrictRedis(host='localhost', port=6379, db=0)
client.set('WLAN', 'Wide Area Network') #typo on purpose
client.set('LAN', 'Local Area Network')
# fixing the previous typo
client.set('WLAN', 'Wireless Local Area Network')
client.get('WLAN')
client.delete('LAN')
del client
```

Código 1: Bloco de código em Python referente aos exercícios com SQL

O objetivo do bloco de código 1 é realizar operações Create, Read, Update, Delete (CRUD) sobre diferentes tipos de base de dados num ambiente de Python.

No que diz respeito às base de dados estão divididas em grupos de SQL e NO-SQL, para o primeiro grupo foi eleito MySQL, por ser uma opção muito comum entre a comunidade *open-source* (como pode ser verificado no site DB-Engines<sup>21</sup>), quanto ao grupo do NO-SQL foi eleito o Redis devido a ser uma base de dados bastante flexível podendo assim adotar vários papéis (como por exemplo *cache* ou *broker*) e porque à semelhança da escolha anterior também é uma solução que atrai bastantes utilizadores (como pode ser verificado no site DB-Engines<sup>22</sup>).

Após a elaboração dos requisitos e dos objetivos, o passo seguinte foi a procura de ambientes de desenvolvimento, do qual resultou numa lista de opções, contendo desde simples editores de texto a IDEs como: Tedit, SubEthaEdit4, SourceLair, Floobits, Complir, Cloud9, Codebox, Codenvy, CodeAnywhere, Koding, Nitrous, Codiad, Nide, Neutron Drive, Collide.

O tempo que iria ser necessário dedicar a cada um deles para testar e classificar iria ser elevado, devido a isso existiu um processo para filtrar e diminuir as opções de escolha. Na redução das opções considerou-se propriedades como tipo (IDEs vs editores de texto), se eram aplicações ou como foi visto em alguns casos extensões para *browser*, por fim, diferenciar se eram simples compiladores de código, editores ou serviços de *cloud/web*, e também identificar se possuíam versões *open-source* de algumas funcionalidades ou do projeto completo.

Tabela 2.1: Comparação entre os possíveis IDE a serem usados na implementação

Funcionalidade	Source Lair	Cloud9	Codenvy	Code-Anywhere	Koding	Nitrous	Codiad
Tipo	Cloud	Cloud	Cloud	Cloud	Cloud	Cloud	Self-hosted
Suporte para CVS	Git, Hg	Git, FTP	Git, SVN	Git, SSH, SVN	Git, FTP	Git	Git
Suporta SSH key	✓	✓	✓	N \ A	✓	N \ A	✓
Versão Open-Source	Parcial	✓	✓	N \ A	Parcial	Parcial	✓
Colaboração com outros utilizadores	-	✓	-	-	-	✓ (beta)	-
Suporte de base de dados	MySQL	MySQL, Redis	3rd party hosted	MySQL, Redis	MySQL, Redis	MySQL, Redis	Self-hosted/3rd party hosted

As opções que foram descartadas não se adequavam, porque algumas eram extensões de *browser* o que não faz sentido basear uma plataforma de *E-Learning* numa extensão que basicamente encapsula o código da página *web* numa aplicação, do ponto de vista técnico algo que não interessava porque iria exigir um esforço maior para desenvolver diferentes extensões para os *browsers* mais usados no mercado e acarreta ter que aprender os manifestos de extensão de cada *browser* e respetivas APIs; relativamente a outras opções não se adequavam devido a serem simples compiladores sendo demasiados básicos perante o que se pretendia, procurava-se algo mais completo (como funcionalidades de *debug* e gestão de projetos). Outras opções eram serviços *web*, sem versão grátis e *open source*, que impediam uma implementação inteiramente *on-premise* se assim a instituição o desejar. No final do processo, a propriedade com maior influência na comparação dos produtos foi a disponibilidade de versões *open source*, onde os resultados deste processo estão apresentados na seguinte Tabela 2.1.

<sup>21</sup><https://db-engines.com/en/system/MySQL>

<sup>22</sup><https://db-engines.com/en/system/Redis>

## SOURCELAIR

No momento da experiência o SourceLair<sup>23</sup> demonstrou ser um produto pouco consolidado, basicamente o serviço oferecia uma máquina e um simples IDE (ver Figura 2.19). O IDE que disponibilizavam continha algumas funcionalidades primárias, mas falhou numa das experiências, o acesso a um Sistemas de Gestão de Base de Dados (SGBD) não era trivial e de difícil processo de instalação.

Após alguma investigação e com ajuda da equipa responsável pelo serviço foi possível descobrir que a provisão de um SGBD, era feito através da submissão de um *ticket* de suporte e que estava sujeito avaliação e justificação (da utilização) à equipa de *developers*, este tipo de fluxo não era o desejado, por isso, deu-se o teste por concluído e classificando o tópico das *databases* de dados como incompleto.

O SourceLair tem uma particularidade, no acesso ao terminal existe um *package manager* próprio para instalar componentes para o sistema de desenvolvimento Apesar de tudo, é uma opção pouco viável porque o *core* não está disponível em formato *open source*, só alguns módulos independentes é que estão disponíveis.

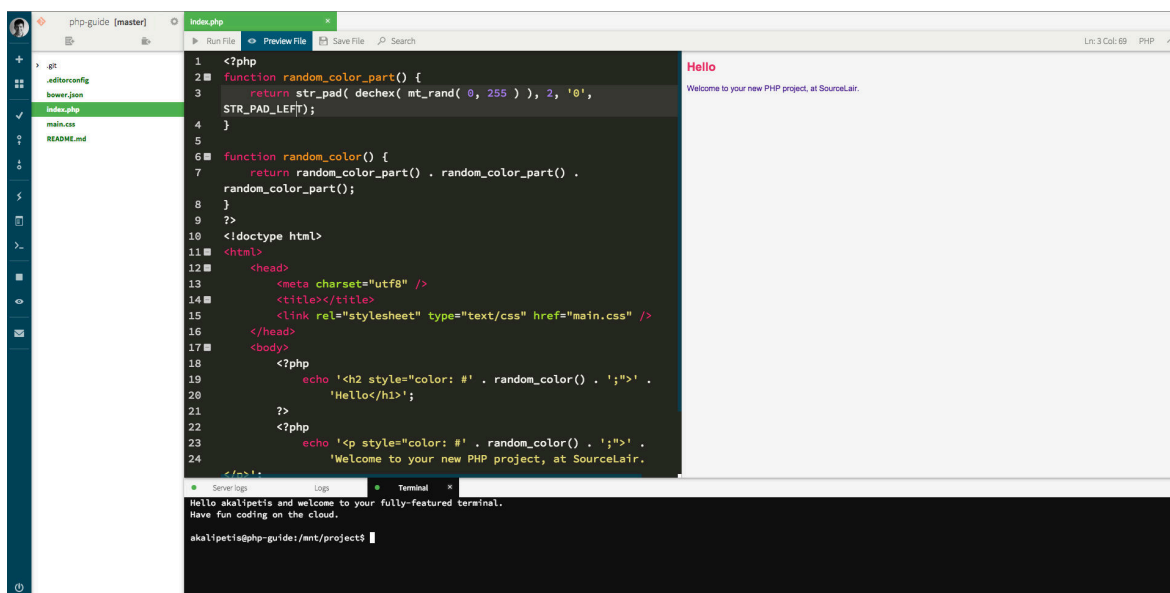


Figura 2.19: *Screenshot* da utilização do SourceLair num projeto de PHP[65]

<sup>23</sup><https://www.sourcelair.com>

## CLOUD9

A Cloud9 provavelmente uma das primeiras empresas a desenvolver e disponibilizar um IDE através de um serviço de *cloud* e *web*. O fator importante na solução por eles apresentada é a utilização do *Ace Editor*<sup>24</sup>, um editor de texto usado há muito tempo em diferentes projetos em *frontend*. Por exemplo, Github e Khan Academy[66]. Disponibilizam, o que eles chamam *workspaces*, pode ser visto como uma unidade de processamento (VM/instância), o qual vem acompanhado de um IDE e de um Terminal. O IDE tem muitas semelhanças com outros já conhecidos, sendo uma tendência no mercado, o Sublime Text<sup>25</sup>, por exemplo: *multi-selection* e as *action keys*.

Distingue-se da concorrência por particularidades como: colaboração em tempo-real, partilha de aplicações a partir de um endereço público, *live support* que consiste numa integração com um serviço de orientação (*mentorship*) chamado HackHands<sup>26</sup>, ferramentas de *debug*, ou seja tão completo como os ditos *full-fledged* IDEs.

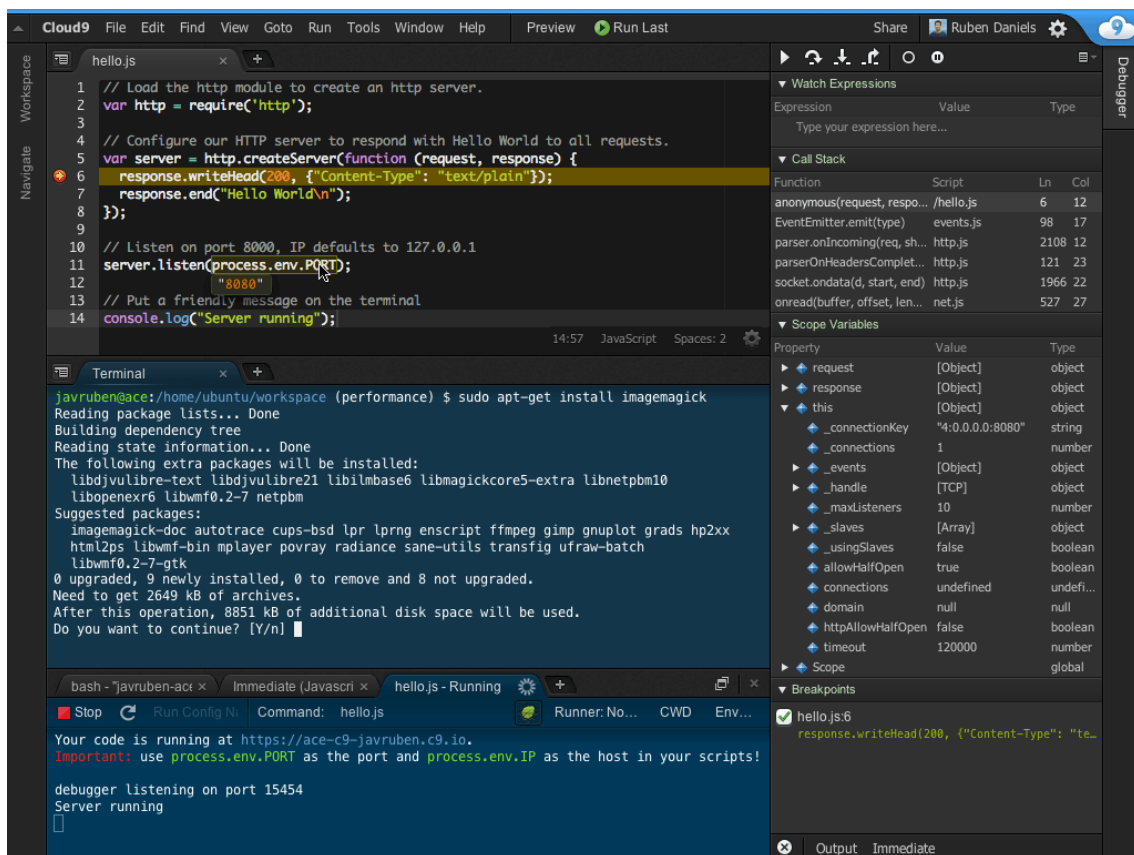


Figura 2.20: Screenshot da utilização do Cloud9[67]

<sup>24</sup><https://ace.c9.io>

<sup>25</sup><https://www.sublimetext.com>

<sup>26</sup><https://hackhands.com>

## CODENVY

O editor do Codenvy<sup>27</sup>, é um produto que existe à algum tempo, pertence ao conjunto de primeiras empresas a tentar oferecer *Workspaces* na *Web* para *developers*.

Após a importação do projeto de teste através da criação de um *workspace* sem perfil, foi necessário alguma configuração, nomeadamente a configuração de um *runner*, que é um processo que permite através de uma interface gráfica definir os parâmetros de execução do projeto na instância fornecida, não estando disponível um terminal ou outra funcionalidade para aceder ao *core* do *workspace*. As operações básicas são feitas a partir do editor de texto, de menu de contextos e pelos *runners*.

O manifesto de um *runner* é semelhante a manifesto de imagens Docker (Dockerfile), sendo um manifesto com vários comandos, dá ao utilizador a possibilidade de escolher e configurar uma *stack* para o seu projeto. Por sua vez, projetos mais simples (que só tenham uma linguagem por exemplo) tornam-se mais fáceis de executar por já existir um *runner* predefinido e pronto para que seja executado numa questão de segundos, e fazer o *deploy* do projeto.

Mas ao contrário do que era esperado, a situação dificulta em muito quando o projeto/código que se importa é diferente, como no caso do projeto experiência, que é constituído por um conjunto de sub-projetos que usa diferentes linguagens de programação e *frameworks*. Até a simples tarefa de executar *run* fica visivelmente mais complexa, porque a criação de um *runner* válido, demorou muito tempo, tornando a tarefa frustrante, até com manifestos genéricos baseados numa imagem de Linux.

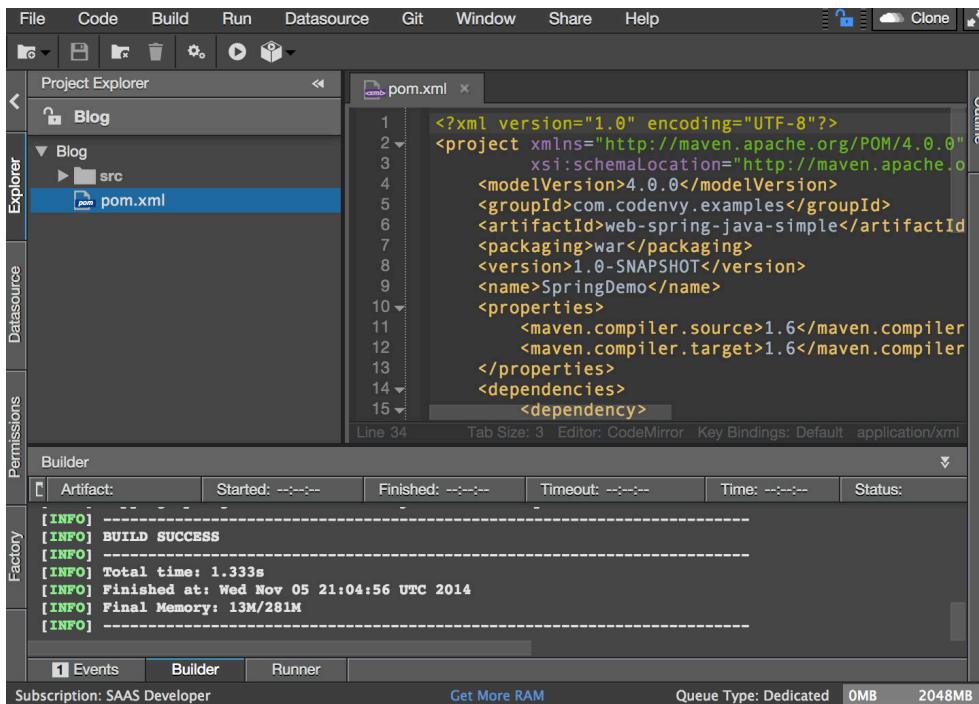


Figura 2.21: Screenshot da utilização do Codenvy[68]

<sup>27</sup><https://codenvy.com/>

## CODEANYWHERE

O CodeAnywhere<sup>28</sup> é uma das alternativas para IDEs em serviço de *cloud*, muito simples e semelhante à concorrência.

Disponibiliza *workspaces* (no qual denomina *devbox*), baseados em *stacks* pré-definidas para facilitar a configuração e usabilidade do ambiente de desenvolvimento para com o utilizador.

Possui uma particularidade, é dada muita importância às integrações de serviços, desde a autenticação (Facebook, Bitbucket, Github), como na criação do projetos adaptando as suas *stacks* ao projetos importados por Git, como também a outros serviços de armazenamento (Dropbox, Google Drive, Amazon Web Services S3).

Embora fosse uma boa opção torna-se inviável por não disponibilizar uma versão *open source*.

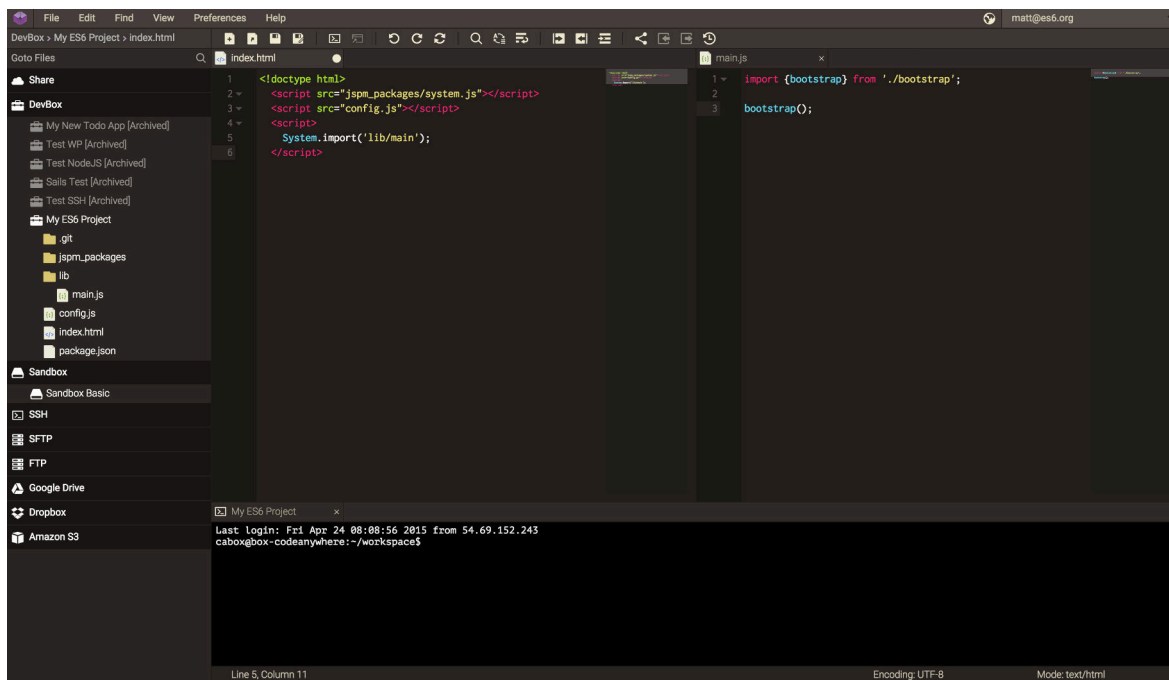


Figura 2.22: *Screenshot* da utilização do CodeAnywhere[69]

---

<sup>28</sup><https://www.codeanywhere.com/>

## KODING

A Koding<sup>29</sup> teve um percurso interessante, apareceu numa altura em que as redes sociais estavam a emergir, e foi apresentada como uma rede social para *developers* e que possuía também *workspaces* para partilhar com amigos.

Na sua primeira versão, focava-se na parte social, no qual disponibilizava um mural, em os utilizadores escreviam *posts* dos seguintes tipos: simples partilhas de notícias/*links/protips*, blocos de código para determinado objetivo, e por fim projetos e/ou ficheiros. A sua plataforma já tinha introduzido o conceito de *workspace/addons* mas como suporte à partilha de conteúdo na parte de rede social, sendo o foco da plataforma a rede social e a comunicação entre *developers*.

À semelhança do Cloud9 foi outra plataforma que suportou o seu sistema na utilização do Ace Editor, a ser usado como editor de texto (para os projetos) e a escrita de blocos de código (para os *posts*).

Depois de as fases *alpha* e *beta* passarem, num espaço temporal de vários meses, a plataforma Koding sofreu uma metamorfose, e agora o seu foco é os *workspaces*. A parte social da versão anterior, também sofreu alterações, agora existe em forma de *chatrooms* (à semelhança dos canais de IRC) e mensagens privadas entre utilizadores. Quanto à componente de *workspaces*, a plataforma evoluiu mas nada de muito drástico, aparentemente está mais completa e robusta, mas com ligeiros problemas de performance quando uma instância é disponibilizada e também com a componente que disponibiliza o terminal pelo *browser*, com ligeiros *bugs* de utilização (*focus*, múltiplas janelas, etc).

Por fim, no âmbito da reutilização, possui alguns módulos disponíveis em *open source*, mas não o suficiente para a implementação da solução proposta.

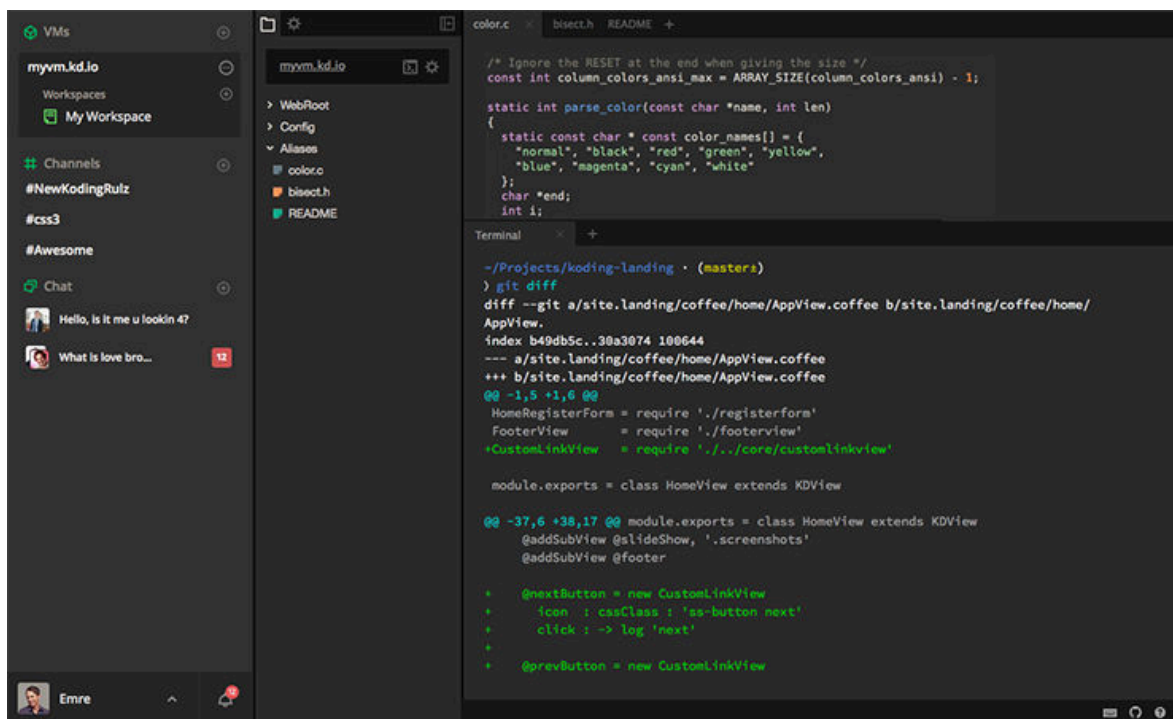


Figura 2.23: Screenshot da utilização do Koding[70]

<sup>29</sup><https://www.koding.com/>

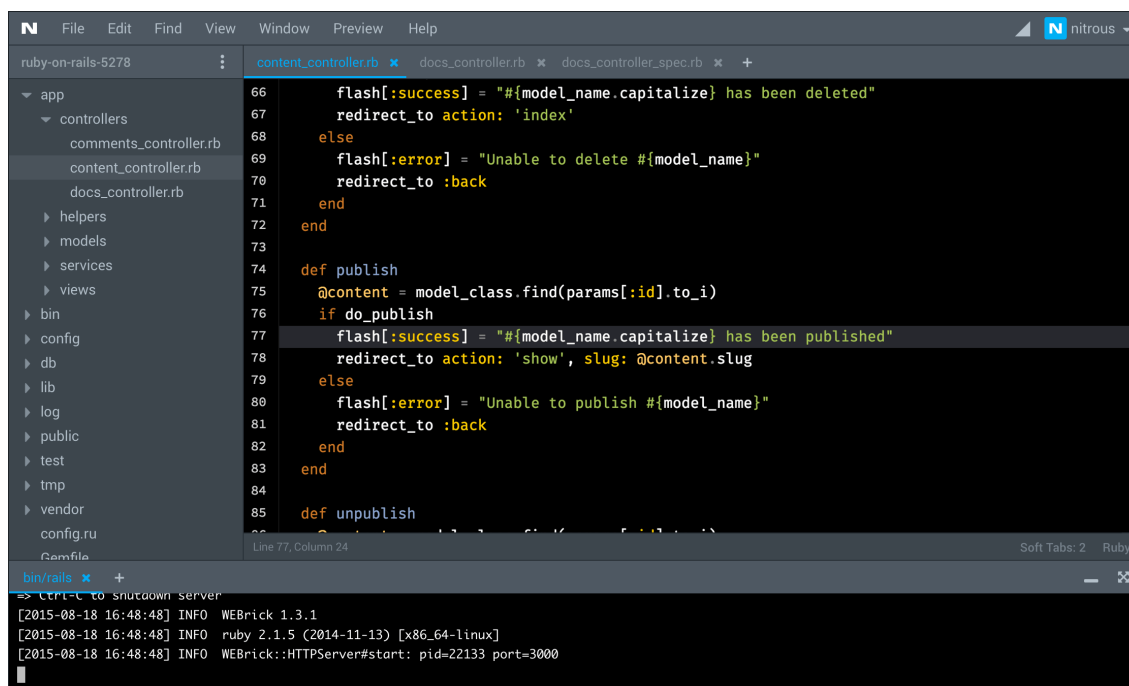


## NITROUS

Aquando a experimentação do Nitrous<sup>30</sup>, houve uma particularidade interessante que até ao momento era incomum encontrar nos outros serviços analisados, a quantificação/gestão de recursos com base em métricas ou unidades (por exemplo: tempo, unidades monetárias/créditos) e com *templates* semelhantes aos que habitualmente eram vistos em soluções de IaaS e PaaS. O Nitrous decidiu fazer algo semelhante com uma moeda virtual, denominada NOS, e com isto os utilizadores podem fazer *upgrades* e escalar os seus ambientes.

Quanto ao *workspace*, disponibiliza um serviço mais ou menos semelhante à concorrência, com a exceção de fornecer um módulo de colaboração embutido no editor (também à semelhança do Cloud9, como vimos anteriormente) e o acesso à base do ambiente é feito também pelo terminal como noutras soluções.

A gestão do ambiente é feito através do terminal, e independente do *template/box* que se escolha, esta dotada das linguagens: Nodejs, Python, Ruby e respetivos *package managers*, entre outras ferramentas. Para ambientes de testes (aplicações *web*) confina a execução e algumas portas, mas depois a aplicação fica disponível num endereço público.



The screenshot shows the Nitrous IDE interface. The top menu bar includes 'File', 'Edit', 'Find', 'View', 'Window', 'Preview', and 'Help'. The main editor area displays a Ruby file named 'content\_controller.rb' with the following code:

```
66     flash[:success] = "#{model_name.capitalize} has been deleted"
67     redirect_to action: 'index'
68   else
69     flash[:error] = "Unable to delete #{model_name}"
70     redirect_to :back
71   end
72 end
73
74 def publish
75   @content = model_class.find(params[:id].to_i)
76   if do_publish
77     flash[:success] = "#{model_name.capitalize} has been published"
78     redirect_to action: 'show', slug: @content.slug
79   else
80     flash[:error] = "Unable to publish #{model_name}"
81     redirect_to :back
82   end
83 end
84
85 def unpublish
```

The bottom terminal window shows the output of the 'bin/rails' command:

```
bin/rails
=> CTRL-C to shutdown server
[2015-08-18 16:48:48] INFO WEBrick 1.3.1
[2015-08-18 16:48:48] INFO ruby 2.1.5 (2014-11-13) [x86_64-linux]
[2015-08-18 16:48:48] INFO WEBrick::HTTPServer#start: pid=22133 port=3000
```

Figura 2.24: *Screenshot* da utilização do Nitrous[71]

Este serviço foi descontinuado no último trimestre de 2016, a atualmente a sua página principal está ativa e descreve o estado do serviço e como adquirir os produtos open-source por eles desenvolvidos.

<sup>30</sup><https://www.nitrous.io/>

## CODIAD

Ao contrário das soluções analisadas anteriormente, o Codiad<sup>31</sup> é primeiro IDE desta análise que é disponibilizado exclusivamente como solução *open source* gratuita. A ideia do projeto é disponibilizar um IDE com tecnologias *web*, destaca-se por ser minimalista, inicialmente parece um simples editor de texto com algumas funcionalidades extra. Após alguma análise, o Codiad demonstra propriedades de extensibilidade, permite a criação/instalação de *plugins* para ampliar as suas funcionalidades e qualidade. Disponibiliza um *marketplace*, com uma lista de *plugins* predefinida pelo projeto, que estão alojados no Github. Através do *marketplace*, o utilizador pode instalar/desinstalar qualquer *plugin* da lista fornecida, ou então, fornecer um Uniform Resource Locator (URL) de um repositório Github.

O nível do editor fica aquém das outras alternativas, quando comparando a quantidade e qualidade das funcionalidades bases, o Codiad destaca a sua interface simples e básica de gestão de ficheiros/projetos (permite as operações básicas de um *filesystem manager*) e por fim também permite *syntax highlight*. O utilização de algumas tecnologias, por exemplo, Git terá que ser recorrendo à instalação das mesmas na máquina que disponibiliza o Codiad.

Da lista inicial de *plugins* não existem *plugins* relevantes que permitam escrita inteligente (*auto-complete*, etc). Possivelmente podia-se procurar implementações destes extras em comunidades como o Github ou desenvolver os vários *plugins* para disponibilizar os extras que as outras plataformas disponibilizam de forma nativa.

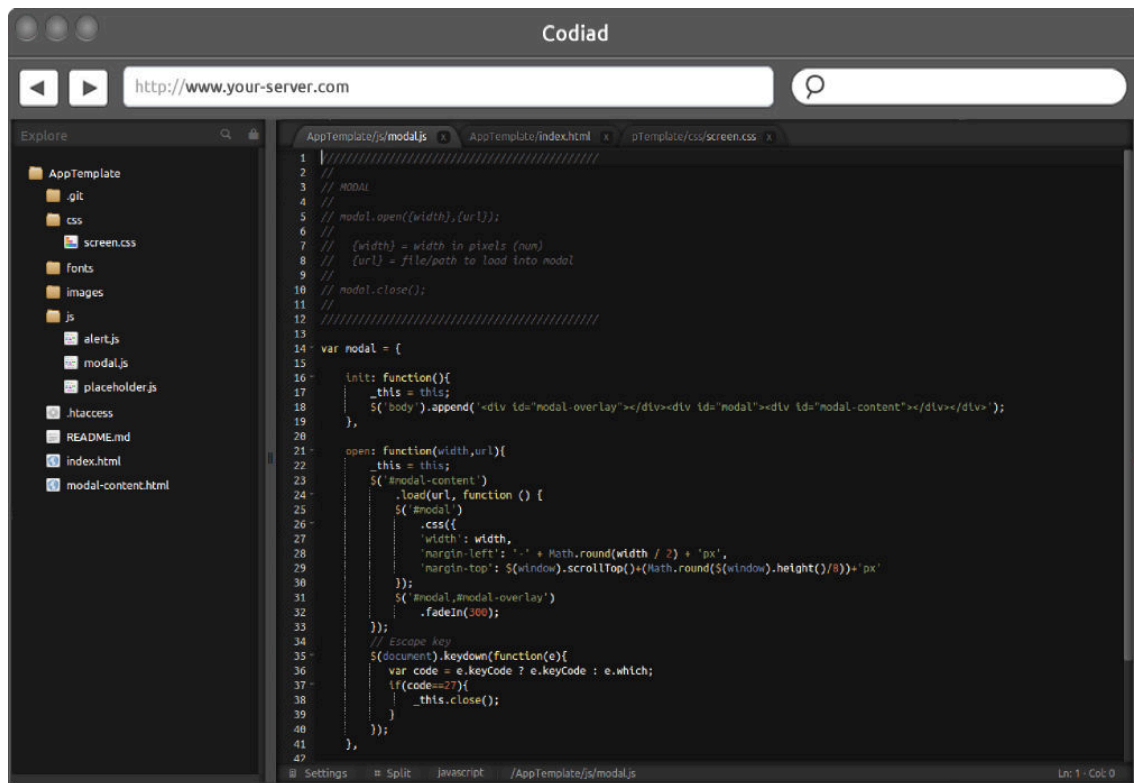


Figura 2.25: *Screenshot* da utilização do Codiad[72]

<sup>31</sup><http://codiad.com/>

## 2.7.2 AMBIENTES DE APRENDIZAGEM

Para esta funcionalidade, a pesquisa utilizou os mesmos métodos e parâmetros mas com resultados diferentes, a começar pelas opções encontradas. A lista de resultados é drasticamente inferior à dos ambientes de desenvolvimento, porque só houve um resultado que preenchesse as necessidades da plataforma, denominado Gitbook<sup>32</sup>.

Felizmente, para além de preencher os requisitos descritos anteriormente este serviço sendo um serviço comercial também possui ferramentas *open source*, este projeto foca-se na publicação de obras escritas com auxílio de diferentes tecnologias, permite a escrita e o desenvolvimento dos ambientes de aprendizagem ambicionados no início do capítulo.

### GITBOOK

Em 2004, os fundadores decidiram desenvolver um formato *open source* para escrita de livros baseados na tecnologia de controlo de versões, Git<sup>33</sup>, resultando numa solução que agrega várias tecnologias: A) para o formato de escrita usam uma *markup language* denominada Markdown<sup>34</sup> de fácil utilização, porque não envolve métodos complexos de formatação de documentos e também é uma linguagem bastante usada em alguns sistemas de *blogs*, como em comunidades de programadores (por exemplo Github); B) o editor (ver Figura Figura 2.26) permite ao utilizador escrever em Markdown e ter uma *preview* do seu documento, entre outras funcionalidades como formatar através da interface gráfica ou pela respetiva sintaxe, permite gerir versões do documento (realizando Git *commits* locais) através do *software* e publicar o seu conteúdo e por fim C) o serviço *online* aonde estão publicados os diferentes conteúdos realizados pela comunidade.



Figura 2.26: *Screenshot* do editor Gitbook[73]

<sup>32</sup><https://www.gitbook.com>

<sup>33</sup><https://git-scm.com>

<sup>34</sup><http://daringfireball.net/projects/markdown/>

As vantagens em usar o ecossistema Gitbook em detrimento de outras opções de escrita é pelo facto que o formato Markdown suporta um sistema de *plugins*, para aumentar o tipo de conteúdos das publicações resultando possivelmente numa nova experiência do leitor.

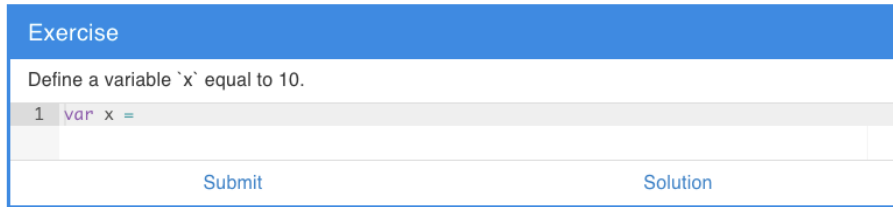


Figura 2.27: Exemplo de um exercício em Javascript num livro do Gitbook

Um *plugin* que é indicado para esta dissertação e que é muito relevante para utilização do Gitbook como ferramenta a integrar, é denominado “exercises”<sup>35</sup> e permite a criação de exercícios de programação interativos (ver Figura 2.27). Para escrever um exercício, é necessário respeitar um formato (ver bloco de código 2), em que a primeira parte é o enunciado, seguido da preparação ou código inicial para servir de apoio, a terceira parte é a solução do exercício, a quarta parte é um bloco (semelhante a um teste unitário) para validar os resultados do exercício e por fim um bloco de funções ou outros blocos de código que possam servir de auxílio ao leitor.

```
{% exercise %}
Define a variable `x` equal to 10.
{% initial %}
var x =
{% solution %}
var x = 10;
{% validation %}
assert(x == 10);
{% context %}
// This is context code available everywhere
// The user will be able to call magicFunc in his code
function magicFunc() {
  return 3;
}
{% endexercise %}
```

Código 2: Exemplo de como criar um simples exercício no Gitbook[74]

Outra vantagem que complementa a anterior, e que entre as várias ferramentas disponibilizadas existe uma que permite converter o conteúdo em Markdown para formatos conhecidos como Portable Document Format (PDF), Electronic Publication (ePUB), HyperText Markup Language (HTML). A conversão em HTML com o *plugin* é importante para obter a funcionalidade de ambientes de desenvolvimento imersivos para a *web* que unifica conteúdos teóricos e práticos num formato acessível e interativo.

<sup>35</sup><https://plugins.gitbook.com/plugin/exercises>

## 2.8 CONCLUSÃO

Na análise quanto ao Estado de Arte é possível identificar que nos últimos anos o uso da tecnologia tem resultado na mitigação de barreiras, para uma melhor distribuição de recursos e para a disponibilização de um ensino com melhor qualidade. É possível identificar que o aumento de equipamentos como os *smartphones* resulta na criação de novos conceitos, como aconteceu com o *M-Learning*. Em grande parte, a evolução do setor de *E-Learning* aconteceu devido à disseminação da *Internet*, isto é, a par da facilidade de acesso a ligações de *Internet* em diferentes parte do globo, plataformas como os CMS, LMS e os MOOC geraram uma “revolução” no ensino, dando posteriormente origem a movimentos como o projeto “OpenEducation” da UE em que pretende definir os objetivos para uma educação mais aberta e para a importância das tecnologias (e outros aspetos) para uma melhor educação. Em paralelo, os casos de sucesso como o CS50, demonstram ao resto do mundo a importância do uso das tecnologias, e como é relevante estar a par das respetivas evoluções, como foi visto nas últimas alterações do curso, resulta no desenvolvimento de novas ferramentas que motivam o desenvolvimento de novos métodos para uma melhor educação.

Estas evoluções e sua aceitação têm ocorrido a um ritmo pouco favorável como se vê neste capítulo. Plataformas como o CS50 (ver Subseção 2.5.4) introduziram uma recente alteração, a utilização do editor de código, o Cloud9, que só será usado no curso após quase 6 anos depois da tecnologia estar disponível ao público, também se verifica, uma adesão pouco atrativa para estes sistemas, porque após vários anos plataformas como o SAAL e o Mooshak continuam a ter pouca visibilidade, da mesma forma como o *plugin* do MOODLE. A falta de ferramentas devidamente competentes para programação, parece ser partilhada com outras instituições e é quando aparece recentemente o PBeL (ver Subseção 2.6.3), um sistema que pretende responder à uma necessidade semelhante à identificada nesta dissertação.

Neste capítulo foram enumeradas plataformas, ferramentas e serviços apesar de estarem no âmbito do *E-Learning* e também apesar das semelhanças, são soluções diferentes e que focam-se em problemas ou necessidades diferentes, mesmo assim foram identificados segmentos em que estas soluções possam ser colocadas lado a lado e comparadas como “iguais” (ver Tabela 2.2), tentando evidenciar o que é relevante mesmo que tenham detalhes interessantes (como foram vistos anteriormente ao longo do capítulo). Da Tabela 2.2, o **público** representa os utilizadores e para quem foi desenhada a solução, o **formato** classifica como os utilizadores interagem com a solução, o **conteúdo** identifica que tipo de conteúdo a solução disponibiliza ao seu público, por fim, as **funcionalidades** descrevem com mais detalhes os objetivos e as necessidades que a plataforma tenta colmatar.

Tabela 2.2: Comparação entre os vários serviços apresentados no Estado da Arte

	Público	Formato	Conteúdo	Funcionalidades principais
edX	indivíduos	aplicação <i>web</i>	MOOC	inscrição, participação e certificação de MOOC
Udemy	indivíduos	aplicação <i>web</i>	<i>marketplace</i> de cursos <i>online</i>	disponibiliza cursos (grátis ou pagos) através de vídeo e com suporte a ficheiros multimédia, avaliação por questionários de escolha, certificação
Codecademy	indivíduos	aplicação <i>web</i>	consolas interativas	disponibilização de consolas interativas para aprendizagem de linguagens de programação e tecnologias adjacentes
CS50	Harvard ( <i>B-Learning</i> ) e indivíduos (MOOC)	reformulação de um curso de uma instituição	<i>B-Learning</i> & MOOC	reestruturação do plano curricular e das ferramentas usadas para melhorar o envolvimento dos alunos e por consequente obtenção de melhores resultados
Codio	instituições de ensino e indivíduos	LMS exclusivo para Ciências da Computação	multimédia, texto e consolas interativas	<i>online</i> IDE, funcionalidades de LMS, cursos com consolas interativas
Nixty	—	LMS	multimédia e texto	gerir turmas, disciplinas, cursos e conteúdos
MOODLE	instituições de ensino	LMS	multimédia e texto	gerir turmas, disciplinas, cursos e conteúdos
SAAL	instituições de ensino	aplicação <i>web</i>	-	submissão e avaliação automática de exercícios de programação
PBeL	solução privada	plataforma de <i>E-Learning</i>	-	realização de exercícios de linguagens de programação, avaliação manual/automática
PyCharm Edu	indivíduos	aplicação <i>desktop</i> (IDE)	cursos de Python	realização de cursos de Python, criar cursos, submeter cursos para repositório público/privado

# NOVA SOLUÇÃO DE ENSINO À DISTÂNCIA

---

*No capítulo anterior verificou-se que no âmbito das disciplinas de informática os sistemas de E-Learning mais relevantes utilizam conteúdo multimédia para o ensino de linguagens de programação, salvo algumas exceções que fornecem novos tipos de interação mas dada a sua natureza não é possível integrar com as plataformas que as instituições usam. Devido a estas diferenças, surge a criação de uma solução, em que o principal objetivo é dotar os docentes e alunos de ferramentas que permitam melhorar os métodos tradicionais de ensino em disciplinas de programação.*

## 3.1 SOLUÇÃO CONCEPTUAL

No âmbito dos sistemas de *E-Learning* dedicados a conteúdos lecionados nos cursos de Informática (rever a análise realizada no Capítulo 2) verifica-se que existe uma tendência de alguns serviços se apoiarem em diferentes tecnologias para auxiliar ou contribuir para a evolução dos métodos de aprendizagem, desde a participação em aulas através do ensino à distância até ao conceito de laboratórios *online*. Entretanto na mesma análise identifica-se que LMS de referência, como por exemplo, o MOODLE através da instalação da solução base não distribui funcionalidades do tipo laboratório que permitam novos métodos de ensino de programação.

Na realização da proposta desta dissertação essa lacuna foi vista como uma oportunidade para promover a implementação prática de um ambiente, que consiste no desenvolvimento de um protótipo que irá contemplar duas funcionalidades: ambientes de aprendizagem e ambientes de desenvolvimento. Estas funcionalidades tem focos diferentes no domínio do ensino, o primeiro foca-se em distribuir um novo método em que o docente transmite conhecimento para os seus alunos e que nos presentes dias é feito com recurso a ficheiros multimédia (por exemplo, vídeo ou PDF) e o segundo foca-se nas condições em que o aluno coloca em prática os conteúdos que lhe foram previamente lecionados (por exemplo, resolução de problemas ou aplicações de algoritmos ou projetos de grupo). O objetivo é que o protótipo integre as duas funcionalidades, que conceptualmente são problemas opostos, e que os disponibilize num único serviço.

Apesar que no atual LMS não são contempladas essas preocupações que foram anteriormente enumeradas, esta solução tem a preocupação da reutilização de alguns serviços: o serviço de autenticação de utilizadores, o serviço de gestão de turmas ou disciplinas, a comunicação entre alunos e docentes e outras funcionalidades já disponíveis pela instituição ou LMS.

## DISPONIBILIZAÇÃO DINÂMICA DE AMBIENTES DE DESENVOLVIMENTO

Em disciplinas que envolvem programação, por vezes, a componente prática é avaliada por projetos, sejam individuais como em grupo. No decorrer destes períodos de avaliação, para desenvolver o projeto ou produzir os resultados esperados, os alunos recorrem a diferentes ferramentas para realizar diferentes tarefas: desenvolver código, distribuir/partilhar código, entre outras. Mas em algumas ferramentas, como por exemplo no momento em que os elementos da equipa estão a desenvolver código simultaneamente, e quando estes precisam de interagir sobre o mesmo bloco de código, fazem-no em diferentes formatos, se estiverem fisicamente no mesmo espaço recorrem a *pair programming*, mas quando estão a cooperar de forma remota podem surgir problemas e neste momento recorrem a métodos de *screen-share* e vídeo-chamada ou a serviços de Version Control System (VCS), como por exemplo: Git ou Apache Subversion (SVN). Uma mais valia para um melhor trabalho em equipa é a possibilidade de no âmbito destas ferramentas os membros do grupo poderem desenvolver código e interagir com diferentes blocos de código em tempo-real. Esta funcionalidade também tem em vista os docentes, na medida em que podem interagir, monitorizar e comentar os avanços dos projetos com os respetivos alunos no ambiente fora da sala de aula, este tipo de interação tem sido frequente e muito bem recebida noutros formatos (ver a Subsecção 2.5.4).

Posto isto, a implementação desta funcionalidade consiste na disponibilização de um ambiente de desenvolvimento (IDE) com capacidades de colaboração e partilha em tempo-real (ver Figura 3.1).



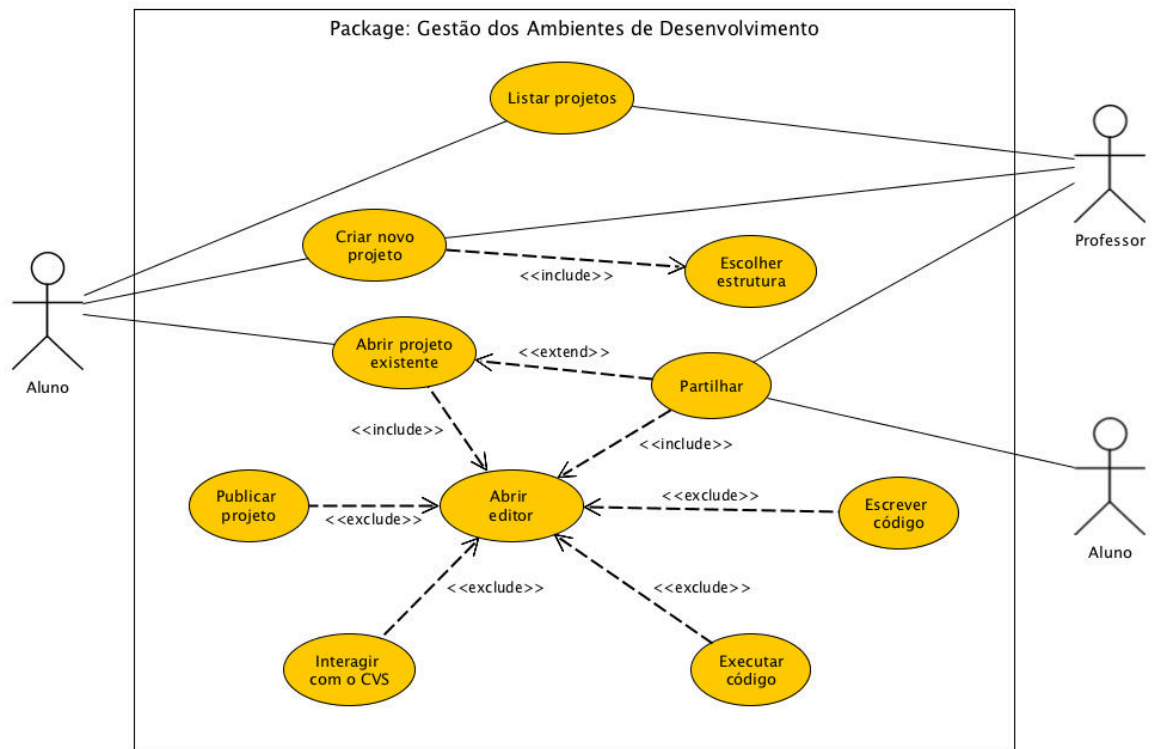


Figura 3.1: Diagrama dos *use cases* da funcionalidade de “Ambiente de Desenvolvimento”

O conceito de ambientes de desenvolvimento está associado a três atores: o **aluno** e o **professor**.

Tabela 3.1: Descrição dos Atores

Ator	Definição
<b>Aluno</b>	O utilizador com sessão autenticada no sistema, inscrito na instituição de ensino e nas UC de programação
<b>Professor</b>	Um utilizador que pertence a corpo docente da instituição de ensino e com sessão autenticada

No âmbito dos ambientes de desenvolvimento, os atores podem executar as seguintes tarefas:

Tabela 3.2: Descrição dos *use cases* da funcionalidade de “Ambiente de Desenvolvimento”

Tarefa	Definição
Listar projetos	Possibilita que um utilizador (aluno/professor) liste os projetos que criou ou que foram partilhados consigo
Criar novo projeto	Permite a um utilizador criar um novo projeto no qual requer que escolha qual a estrutura (explicada a seguir) do projeto
Escolher estrutura	Capacidade que o utilizador tem de discriminar o tipo de projeto, linguagem de programação, dependências/ <i>frameworks</i> , entre outras características
Abrir projeto existente	Um utilizador (aluno/professor) tem a possibilidade de abrir um projeto criado por si ou partilhado consigo
Partilhar	Possibilita que um utilizador (aluno/professor) possa partilhar o seu projeto e código com outros utilizadores(colega/professor)
Abrir editor	Possibilita que os utilizadores (alunos/professores) abram o projeto com um editor (IDE) para que possam editar o código do mesmo
Interagir com o VCS	Possibilita que através do editor os utilizadores possam interagir com o sistema VCS vigente na sua instituição
Escrever código	Após abrir o editor os utilizadores tem a liberdade de escrever código no projeto e gerir a estrutura de ficheiros do mesmo, a escrita de código irá ter auxílios diferentes consoante o suporte que é dado pela configuração da estrutura em tarefas anteriores
Executar código	A possibilidade dos utilizadores executarem o seu projeto com as devidas especificações
Publicar código	Em alguns tipos de projetos (por exemplo, quando recorrem a tecnologias <i>web</i> ou sistemas distribuídos) e se existir a necessidade de serem públicos, existe a possibilidade dos utilizadores "publicarem"o projeto tornando-o publico

## DISPONIBILIZAÇÃO DINÂMICA DE AMBIENTES DE APRENDIZAGEM IMERSIVOS

O atual panorama dos métodos de ensino para a prática de programação, estão fortemente dependentes de material digital, por exemplo, tutoriais em formato de slides ou vídeo, é algo que se tornou comum por ser o formato de conteúdo ao qual os alunos estão habituados e que as plataformas LMS conhecem e sabem lidar.

Através de alguns serviços e soluções como os que foram referidos na Capítulo 2 constata-se métodos de aprendizagem mais interativos (consultar Subseção 2.6.3) que podem trazer melhorias no que toca ao envolvimento por parte dos alunos quando comparados com os métodos tradicionais.

A funcionalidade surge da ideia de ter um ambiente mais interativo e imersivo que permita aos alunos um maior envolvimento no processo de aprendizagem e obter melhores resultados. O objetivo é proporcionar o material teórico em paralelo com o material prático de forma integrada na tentativa de facilitar a compreensão da matéria na obtenção de conhecimento por parte do aluno, esta funcionalidade consiste na implementação de consolas interativas no contexto das plataformas de *E-Learning* da respetiva instituição (ver Figura 3.2).

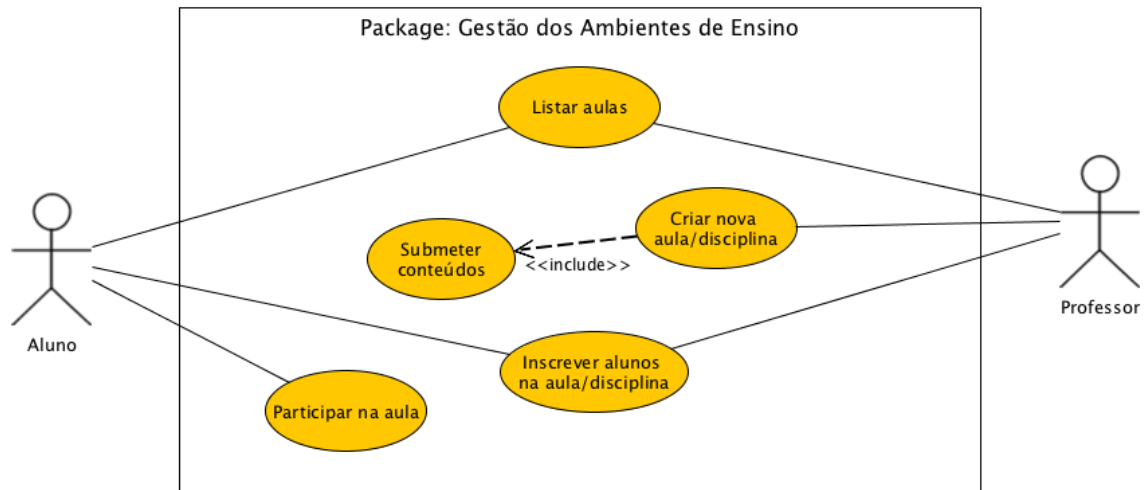


Figura 3.2: Diagrama dos *use cases* da funcionalidade de “Ambiente de Aprendizagem”

O conceito de ambientes de aprendizagem está associado a três atores: o **aluno** e o **professor**.

Tabela 3.3: Descrição dos *use cases* da funcionalidade de “Ambiente de Ensino”

Ator	Definição
<b>Aluno</b>	O utilizador com sessão autenticada no sistema, inscrito na instituição de ensino e nas UC de programação
<b>Professor</b>	Um utilizador que pertence a corpo docente da instituição de ensino e com sessão autenticada

## 3.2 DISCUSSÃO

Na Capítulo 2, é evidenciado que a UA já detém duas plataformas de *E-Learning* (relembrando MOODLE e SAAL) sendo ambas extensíveis. No caso do MOODLE é feito através de um sistema de *plugins* usando PHP HyperText Pre-Processor (PHP) como linguagem de programação. Os *plugins* encontram-se divididos por tipos, por exemplo o *plugin* avaliado na Capítulo 2 é do tipo *Question* indicado para o cenário de avaliação automática, para a funcionalidade de ambiente de aprendizagem o *plugin*, que por definição parece ser mais indicado é o *Course Format* que permite a criação de um novo formato ou representação de um curso, ou até mesmo o *Book Tool* mas este não foi possível averiguar, por falta de informação na documentação oficial.

Da mesma forma que o MOODLE, o SAAL apresenta-se como sendo modular, mas com a diferença que não tem disponível uma descrição clara de como é possível estender as funcionalidades base do sistema. É possível inferir que em última instância, o aumento de funcionalidades é atingido por adição direta ao código-fonte do *backend* e adicionar a respectivas *views* (de notar, que usa Model-View-Controller (MVC) com Backbone.js<sup>1</sup>) para os endpoints criados.

Posto isto, em relação à implementação das funcionalidades para a nova solução de ensino de linguagens de programação, podemos assumir que existe a possibilidade de estender ambas as plataformas de *E-Learning* analisadas, o MOODLE e o SAAL, mas esta atualização não será feita em detrimento do desenvolvimento de raiz de uma nova solução. Incrementar as funcionalidades ao MOODLE não é tido em conta como opção, sendo uma plataforma que usa PHP tanto no código base como no desenvolvimento de *plugins*, a utilização desta linguagem não faz parte do interesse para o desenvolvimento devido a ser uma linguagem por vezes classificada como *loosely typed* (pelo menos antes da versão 7) e num projeto grande dimensão e com uma determinada complexidade esta característica é vista como um desafio adicional à manutenção do mesmo e possivelmente dificulta o desenvolvimento da prova de conceito das várias funcionalidades de *E-Learning* pretendidas com este trabalho. De um ponto vista técnico, existe um conjunto de testes de performance[75] que combina várias linguagens com diferentes *frameworks* para recolher dados como número de pedidos por segundo entre outros, e as combinações de PHP em comparação a combinações de outras linguagens (Java, Python, etc) digamos que tem menos desempenho, porque em nenhum dos testes ocupou os primeiros 5 lugares da tabela classificativa. Quanto ao projeto SAAL dada a falta de documentação relativamente à extensibilidade do sistema, torna a tarefa de adicionar funcionalidades num desafio de *reverse engineering* ao código base, algo desmotivador porque adiciona mais complexidade e faz com que o tempo despendido para esta compreensão do projeto desvia atenção necessária para a implementação das novas funcionalidades propostas. Para além disto, o fator decisivo foi no momento da submissão da proposta da dissertação e que já existia um conhecimento prévio de algumas ferramentas analisadas nas secções Subsecção 2.7.1 e Subsecção 2.7.2, e depois da análise cuidada, fez sentido a utilização destas ferramentas para a implementação das funcionalidades desejadas.

Não tendo as condições ideais para acrescentar valor em plataformas como MOODLE ou SAAL, o objetivo deste trabalho é o desenvolvimento de uma aplicação *web* que permita a um aluno aceder a um ambiente de desenvolvimento a partir do *browser*, e que permita a um professor disponibilizar uma aula interativa que possa ser visualizada a partir do *browser*. Utilizando as ferramentas avaliadas na Secção 2.7, desta análise, as ferramentas eleitas para integrar na solução é a versão *open source* do IDE Cloud9 e o formato de conteúdo Gitbook, e também ferramentas adjacentes ao mesmo formato. A solução proposta posteriormente neste capítulo será em torno da possibilidade de escalar as funcionalidades destas duas ferramentas para fornecer uma plataforma de *E-Learning* para as Ciências de Computação, mais precisamente para as UC que se foquem em programação e respetivas ferramentas de desenvolvimento.

---

<sup>1</sup><http://backbonejs.org/>

### 3.3 ARQUITETURA PROPOSTA

No início do documento vários objetivos foram enumerados mas só alguns se preocupam com o estado da infraestrutura, nomeadamente a flexibilidade e a escalabilidade da mesma, pelo que estas considerações tem de ser avaliadas na fase da conceção. Tendo isto em consideração, a escolha para o momento da implementação é adotar o paradigma SOA quanto à arquitetura da solução. Estes conceitos favorecem a partilha ou reaproveitamento de serviços através da composição ou orquestração, desta forma é possível que a criação de valor surja quando se juntam diferentes serviços para alcançar um ou mais objetivos. Juntamente a estes conceitos, a utilização de tecnologias de *cloud*, por definição fornecem alguma flexibilidade em termos de recursos (sejam de armazenamento, processamento ou redes) permitindo assim uma melhor resposta às necessidades variáveis (alunos, trabalhos, etc) do departamento. Por fim, na perspetiva de desenvolvimento, focar em tecnologias *web* irá permitir uma maior adoção por parte dos utilizadores porque através de um *browser* é possível servir múltiplos clientes (equipamentos e sistemas operativos) e com um índice de manutenção inferior.

Numa infraestrutura SOA, é importante definir a comunicação entre os vários componentes que vão compor a solução, deste modo é necessário escolher os protocolos para essa interação. Para efeitos de comunicação síncrona entre os diferentes componentes foi optado por utilizar o protocolo HyperText Transfer Protocol (HTTP) para implementar interfaces REpresentational State Transfer (REST) que são orientadas ao conceito de recurso e para comunicar com os serviços de terceiros. O conceito de REST define que deve ser orientado a recursos, por esse motivo foi definido a utilização do recurso *workplace* para representar os ambientes de desenvolvimento de um aluno e do recurso *course* para representar o conteúdo teórico-prático gerado pelo docente. A implementação das interfaces REST mapeia as operações CRUD de um para um aos métodos do protocolo HTTP, que permite criar as interações necessárias com os recursos disponíveis. Atualmente a utilização de interfaces REST é muito utilizado em ambientes SOA pela sua facilidade de desenvolvimento e integração.

Apesar que o protocolo anterior pode ser usado para comunicação assíncrona, por exemplo num *browser* usando Asynchronous JavaScript and XML (AJAX) ou num programa com linguagens que suportem o paradigma *event-driven* como Javascript, mas para este tipo de comunicação foi optado por se utilizar um protocolo diferente, Advanced Message Queuing Protocol (AMQP), por suportar funcionalidades como *publish-subscribe* que permite lidar com uma *queue* de mensagens com vários intervenientes em que alguns estão responsáveis por submeter as mensagens enquanto outros estão à espera de receber as mensagens e operar consoante o seu conteúdo. Neste projeto, a utilização de *publish-subscribe* um padrão utilizado recorrentemente em sistemas distribuídos ou com o paradigma SOA, será importante para a comunicação entre a solução proposta e a plataforma de *cloud*. A utilização deste protocolo para comunicação assíncrona é para evitar constrangimentos na execução dos componentes ou no fluxo da plataforma desenvolvida.

A conceção desta solução sugere que a plataforma de *E-Learning* terá uma estrutura semelhante à representada na Figura 3.3.

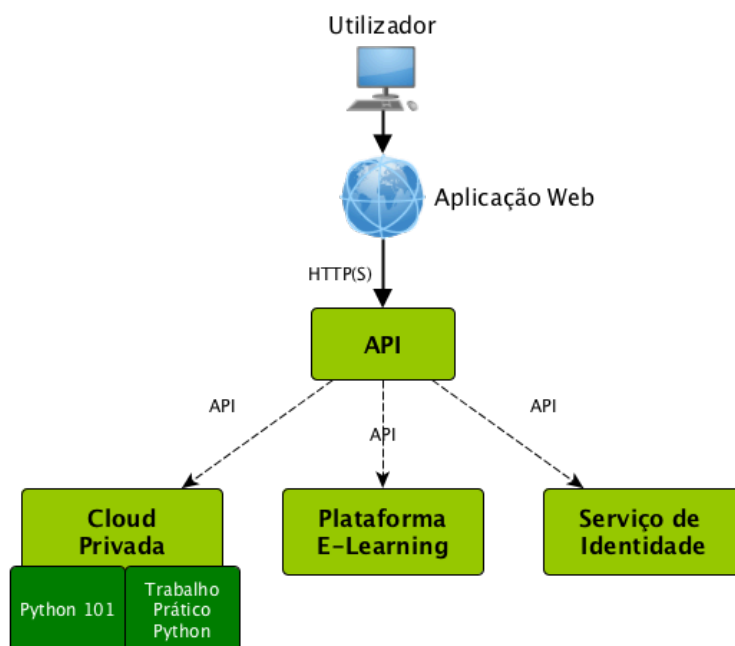


Figura 3.3: Arquitetura da implementação do sistema

### 3.3.1 UTILIZADOR

O Utilizador para aceder à plataforma tem de ter em sua posse um equipamento com *browser* e as credenciais de acesso à plataforma, por exemplo o utilizador único disponibilizado pela instituição, após o processo de autenticação e autorização, o sistema cria um perfil para o utilizador.

Existem dois tipos de utilizador, o aluno e o docente. No ambiente de desenvolvimento, a distinção dos perfis é irrelevante, ambos terão acesso ao editor de texto, gestão de projeto e terminal da instância, para que não haja limitações da interação entre ambos. Quanto ao ambiente de aprendizagem, os utilizadores são distintos porque o docente conceptualmente é o único com permissões para criar conteúdos para as disciplinas, por esse motivo também é o único que pode criar os ambientes de aprendizagem, enquanto que o aluno só pode consumir a interface interativa do ambiente.

### 3.3.2 APLICAÇÃO WEB

Representa mais do que a GUI, que deverá ser simplista e funcional, para além disso funcionará como compositor de serviços que realizará alguma lógica no lado do cliente, como compor as interações do cliente com os serviços de autenticação ou os serviços de criação de ambientes de desenvolvimento, usando diferentes tecnologias, como AJAX, que permitem uma interface dinâmica. Nos pedidos assíncronos aos serviços a interface não necessita de se redesenhar por completo para atualizar informação de um único elemento.

### 3.3.3 PLATAFORMA DE CLOUD

A criação dinâmica e flexível de ambientes (aprendizagem e desenvolvimento) requer razoáveis quantidades de recursos e uma gestão ágil para melhor aproveitamento desses mesmos recursos, esse é um dos desafios da solução, e caso seja possível, reaproveitar a infraestrutura da instituição sobre o pressuposto que está abstraída por uma plataforma de *cloud* privada, seja ela *on-premise* vs *off-premise* ou comercial vs *open source*.

A utilização de um conceito como *cloud* é favorável para a obtenção de uma plataforma flexível e que permita escalabilidade perante as suas necessidades.

De forma mais direta, o componente *cloud privada* da Figura 3.3 representa qualquer plataforma(s) utilizada(s) pela instituição, por exemplo: Openstack<sup>2</sup>, AWS<sup>3</sup>, Microsoft Azure<sup>4</sup>, etc.

Não estando limitado a um serviço de *cloud* específico, a solução terá que se adaptar através da utilização de uma camada intermédia (*middleware*), esta abstração é necessária devido a cada serviço ser dotado de um conjunto de conceitos e funcionalidades que não são transversais a todas as opções, por exemplo, diferenças entre instâncias de Platform as a Service (PaaS) para um Infrastructure as a Service (IaaS), ou as diferenças entre instâncias da AWS EC2<sup>5</sup> ou do Openstack<sup>6</sup>, sendo que os diferentes fornecedores atribuem um nome/conceito que pode diferir para as suas unidades de processamento, e cada um acarreta características diferentes, alguns denominam de *worker*, *instance* ou *gear*.

Neste documento, e na descrição da solução, o termo usado para mencionar a unidade de processamento será *instance*, uma representação agnóstica à tecnologia usada e aos conceitos inerentes. A *instance* vai representar os diferentes ambientes gerados, ou seja, uma *instance* terá em exclusivo um ambiente de desenvolvimento por aluno/projeto (se a política de utilização da instituição ou até a disponibilidade dos recursos o permitir) ou um ambiente de aprendizagem. Para um maior proveito, terá de existir este isolamento por *instance* e também permitir maior flexibilidade na manutenção da própria infraestrutura como para gerir a escalabilidade do sistema.

Na Figura 3.3, a existência da abstração não é evidente mas existirá no âmbito do agregador/orquestrador.

### 3.3.4 BASE DE DADOS

Com a orquestração de diferentes serviços com diferentes conceitos, será necessário registar alguma informação, em alguns casos será dados duplicados, noutros a criação de novos dados. A informação obtida a partir desta base de dados irá permitir reagir e organizar a orquestração dos diferentes serviços. Existem exceções como que para evitar constantes pedidos para uma informação de suporte, será guardada alguma informação como se uma *cache* se tratasse.

---

<sup>2</sup><http://www.openstack.org/>

<sup>3</sup><https://aws.amazon.com/>

<sup>4</sup><https://azure.microsoft.com>

<sup>5</sup><https://aws.amazon.com/ec2/>

<sup>6</sup><http://www.openstack.org/software/releases/mitaka/components/nova>

### 3.3.5 SERVIÇO DE IDENTIDADE

Mais uma nova camada de abstração, da mesma forma que foi assumido que a solução é agnóstica quanto à infraestrutura, também haverá essa abordagem neste nível, abordar a situação como se o sistema de autenticação fosse desconhecido e proporcionar a autenticação/autorização por uma API, e com uma filosofia de configuração (ou *plugins*) acionar os métodos de autenticação desejados: A) *username/password* B) redes sociais(Twitter<sup>7</sup>, Facebook<sup>8</sup>, etc) C) outras plataformas (OpenID, OAuth<sup>9</sup>, etc).

Este componente representa o serviço de autenticação e de identidades da instituição, no caso da UA existe uma página[76] dedicada ao mesmo.

O serviço é implementado em OAuth na versão 1.0a, denominada por IdentityUA, basicamente é através deste componente que o utilizador terá de usar as credenciais do Utilizador Universal (UU) para poder autenticar na nova plataforma (OAuth *application*) e assim aceder ao novo serviço.

O OAuth é um *standard* criado em 2007 e revisto em 2009 criado por uma comunidade de *web developers* para resolver um problema de partilhar o acesso a recursos privados. Numa altura em que se usava o modelo cliente-servidor para dar acesso a recursos, e havendo falta de um *standard*, apesar de haver soluções comerciais como OpenID[77]. Alterou-se o anterior modelo para incluir um interveniente, o “resource owner”, a entidade dona dos recursos/conteúdos. Com OAuth foi criado um método em que diferentes aplicações ou serviços possam aceder a vários recursos em nome do “resource owner”, desde que lhe sejam dadas aos clientes a devida autorização para aceder aos recursos.

Para complementar o *standard* também define um meio que permite ao utilizador controlar o acesso de serviços de terceiros aceder aos seus recursos. O *standard* é baseado em *tokens* gerados após um processo de autenticação e autorização, cada *token* possui uma validade e um *scope* (lista de permissões)[78].

### 3.3.6 BROKER

Apesar da plataforma estar a ser direcionada para se adaptar a diferentes serviços de *cloud*, existe a possibilidade de o serviço não disponibilizar uma interface assíncrona, para isso, este componente irá disponibilizar um *worker* ou *queue* para interagir com a *cloud* de forma assíncrona, e mitigar a possibilidade de *bottleneck* na *cloud* API em momentos que os pedidos à mesma podem atingir picos devido ao acesso.

---

<sup>7</sup><https://twitter.com/>

<sup>8</sup><https://www.facebook.com/>

<sup>9</sup><https://oauth.net/>



## 3.4 CONCLUSÃO

A UA é uma instituição que encara esta problemática com empenho porque em diferentes momentos já dedicou recursos para a evolução dos atuais métodos utilizados no DETI para os seus cursos, como os casos apresentados nesta dissertação. Desta propõe-se a evolução de duas características do Ensino: a forma como o conhecimento é distribuído e como o conhecimento é posto em prática, para isso, a proposta consiste em desenvolver um sistema no topo de plataformas (como o serviço de autenticação) que suportem o funcionamento destas funcionalidades dentro do domínio de *E-Learning*.



# IMPLEMENTAÇÃO

---

*Neste capítulo estão descritos os vários passos da implementação que acompanharam as decisões tomadas no capítulo anterior, desde as preocupações de escalabilidade até a um cliente de multi-plataforma. Abordando com detalhe questões específicas que tenham afetado a implementação ou distribuição da solução como serviço. Por fim, uma demonstração das duas macro-funcionalidades do sistema evidenciado como os componentes mais relevantes se comunicam.*

## 4.1 COMPONENTES

Esta implementação teve como base a utilização de duas ferramentas eleitas após a análise apresentada na Seção 2.7. A forma de funcionamento destas ferramentas condicionam a implementação da solução proposta. Em primeiro lugar, deve-se considerar que a provisão dos ambiente de desenvolvimentos reflete na necessidade de instalar o IDE e respectivas dependências da *stack* escolhida num ambiente remoto. Depois, no caso dos ambientes de aprendizagem, convém evidenciar que na análise realizada não foi encontrado um editor *web* disponível. Algo que afeta a escrita do conteúdo a partir da interface da plataforma, devido a esta limitação, a edição dos cursos terá de ser feita externamente à solução desenvolvida. Quando a edição estiver terminada, assim já pode ser submetida para visualização. Por este motivo, o desenvolvimento da solução efetiva sofreu ligeiras alterações perante o que foi o desenho inicial, com o objetivo de uma melhor integração.

Como é possível verificar na Figura 4.1, a mudança mais significativa deve-se ao facto de não existir um ponto central de acesso por parte da aplicação *web*, que anteriormente esta acedia a esse mesmo componente que orquestrava os acesso às funcionalidades, atualmente essa operação de orquestração acontece ao nível do cliente e a combinação de funcionalidades é feita no *browser* com o acesso à aplicação *web*.

Outra alteração é a adição de camadas de abstração para o domínio de identidade e o dos ambientes (na Figura 4.1 denominado por “Camada de Abstração da Cloud”). A abstração permite a aplicação *web* comunicar com a plataforma com uma “linguagem” estanque independentemente das tecnologias e protocolos de provisão e autenticação, fornecendo alguma flexibilidade perante as diferentes infraestruturas que os diferentes institutos de educação possuam, resultando assim que cada instituição que use esta solução possa ter dependências diferentes e sendo o seu único esforço a respetiva configuração. Caso um protocolo/tecnologia ainda não esteja disponível, o sistema de *plugins* de cada camada permite que em casos simples se resolva com uma simples instalação do um módulo para o respetivo protocolo/tecnologia. Em casos mais complexos envolva a escrita do novo *plugin*. De qualquer forma, cada camada foi idealizada para ser extensível às necessidades da instituição.

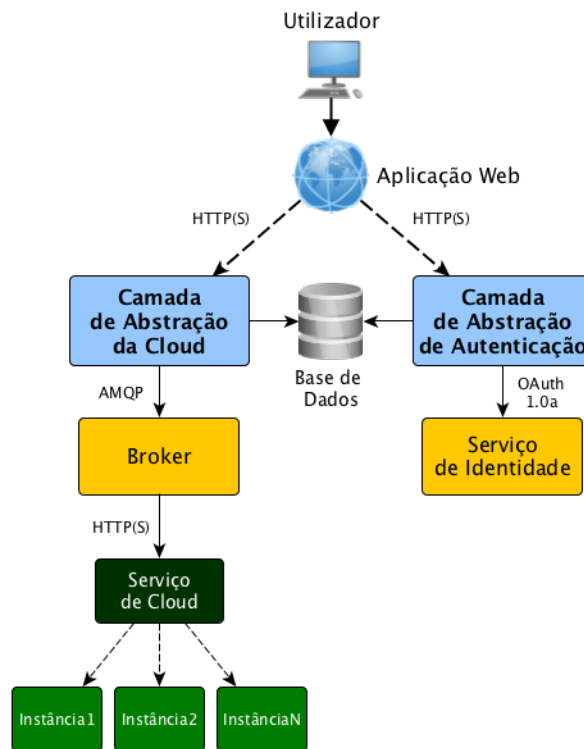


Figura 4.1: Arquitetura da implementação do sistema

No capítulo anterior, na Figura 3.3 está um referência a uma possível interação com “Plataform de E-Learning” em abstrato, essa interação não ocorreu na versão da solução implementada. Deve-se às dificuldades que surgiram durante a implementação dos serviços de abstração, situação que é discutida com mais detalhes posteriormente neste capítulo.

### 4.1.1 APLICAÇÃO WEB

A interface que o utilizador irá utilizar para interagir com as funcionalidades disponibilizadas pela nova plataforma de *E-Learning* será através da aplicação *web*. Esta aplicação está desenvolvida com base em tecnologias como HTML para a criação da GUI, que define o posicionamento e a hierarquia dos elementos que constituem a página *web*, também denominado por *view*. Depois é necessário estilizar/formatar os elementos que constituem a *view* e a tecnologia *standard* para essa tarefa é o Cascading Style Sheets (CSS), para algumas utilizações esta tecnologia pode ser limitada, por esse motivo, foi usado o LESS<sup>1</sup> permite estender a utilização base do CSS. O LESS é um *pre-processor* que permite estilizar/formatar as páginas com o auxílio de funções e outras técnicas (por exemplo, hierarquia de estilos). Por fim, o processamento dos dados que o *backend* fornece vai ser processado por Javascript. A utilização de Angular<sup>2</sup> que através de *services* e *controllers* permite gerir e implementar a lógica da aplicação no lado do cliente, e melhora a desenvolvimento do cliente porque possui conceitos e uma estrutura que permite organizar melhor o código e lidar de forma mais ágil a manutenção do projeto. A utilização de um *framework* de Javascript permite e está responsável por interagir com as diferentes APIs e invocar operações básicas para construção da página. A razão porque este processamento foi escolhido para ser executado no lado do cliente é para tirar a carga do servidor ao lidar com funções de *templating* ou outras operações deste género. Atualmente, os equipamentos usados pelos clientes têm poder de processamento suficiente para ficar responsáveis por estas operações, e desta forma, o servidor liberta recursos para gastar com outras operações mais complexas, nomeadamente a provisão de ambientes de desenvolvimento.

O HTML é usado para criar uma *master page* que depois através da implementação de outras técnicas (*Views*) permite popular partes da página de forma dinâmica. O LESS define classes e outros parâmetros que definem o estilo e a formatação da página e dos elementos HTML da página e das *views*. Sendo um *pre-processor* terá de ser gerado um CSS para ser interpretado pelo HTML e o pelo *browser*. À semelhança do padrão MVC a utilização do Angular implementa Model-View-Whatever (MVW) (ver Figura 4.2), enquanto o primeiro tem o *Controller* responsável por resolver a comunicação entre o *model* e a *view*, o segundo não tem restrições por isso o “W” ser de *whatever* e fica ao critério de cada um (no Angular existem vários componentes: *controller*, *services*, etc)[79].

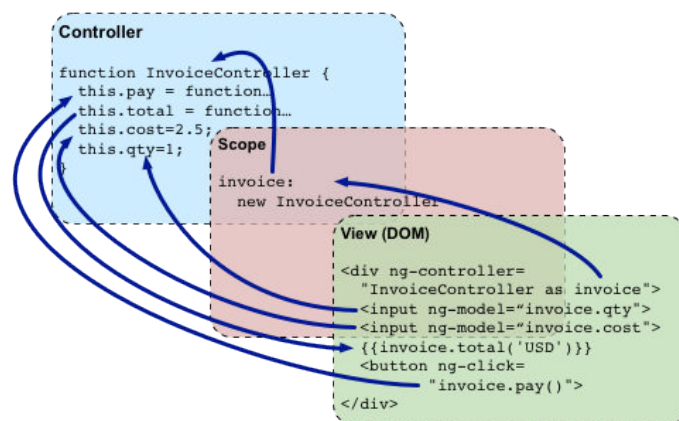


Figura 4.2: Exemplo de *data binding* com auxílio de um *controller*[80]

<sup>1</sup><http://lesscss.org>

<sup>2</sup><https://angularjs.org>

A aplicação *web*, mais precisamente os ficheiros LESS e Javascript são compilados e *minified* e são disponibilizados recorrendo a um servidor *web*, entre as várias opções, o servidor *web* escolhido por experiência prévia, por motivos de performance[81] foi o Nginx<sup>3</sup> e por ser de fácil configuração para cenários em que é necessário gerir a carga (*load balancing*).

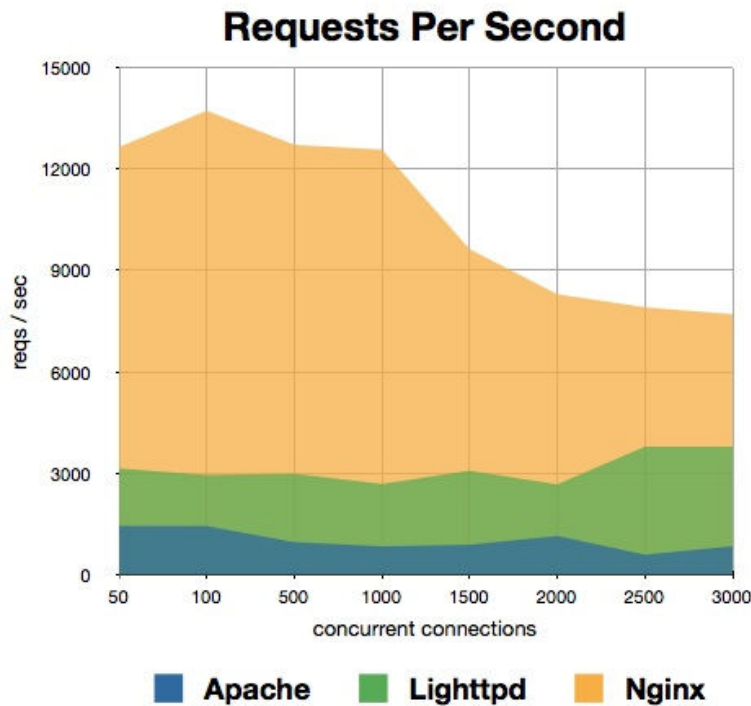


Figura 4.3: Comparação de pedidos por segundo do Nginx com outras alternativas[81]

#### 4.1.2 BASE DE DADOS

Para uma melhor coordenação dos diferentes serviços suportados e também para manutenção dos novos conceitos criados pelas abstrações, existe a necessidade de registar alguma informação, por exemplo: a conversão de propriedades abstratas para o conceito específico do serviço dependente, guardar os ambientes de aprendizagem/desenvolvimento, permissões de acesso, entre outras informações.

Por exemplo, no caso do conceito *workplace* que é o modelo para o ambiente de desenvolvimento, uma das suas propriedades, que une a abstração com a informação do serviço de *cloud* subjacente, é o tipo de ambiente (denominado por *stack*) que define o instância, porque um ambiente para Python usa uma instância diferente de um ambiente de Java.

Dos requisitos funcionais não existe atualmente uma necessidade de realizar relações entre as “entidades” devido a isto não existe um motivo forte para utilizar exclusivamente SQL logo com um espectro de opções maior, a solução eleita é MongoDB<sup>4</sup> principalmente por ser dedicada a documentos, que é o tipo de objeto usado para guardar as informações entre as camadas de abstração e respetivo serviço e também pela facilidade de converter Documentos em JavaScript Object Notation (JSON).

<sup>3</sup><https://www.nginx.com>

<sup>4</sup><https://www.mongodb.com>

Na base-de-dados são guardados vários documentos, os mais relevantes são *Course* (ver Tabela 4.1) e *Workplace* (ver Tabela 4.2).

Tabela 4.1: Descrição do documento *Course*

Propriedade	Tipo	Descrição
uuid	UUID	O identificador único para cada curso
name	<i>String</i>	O nome público do curso
description	<i>String</i>	A descrição do curso
createdAt	<i>DateTime</i>	A data de quando o curso foi criado
container	<i>String</i>	O nome usado para a criação do <i>container</i> na plataforma de <i>cloud</i>
link	<i>String</i>	O endereço para a página inicial do curso

Tabela 4.2: Descrição do documento *Workplace*

Propriedade	Tipo	Descrição
uuid	UUID	O identificador único para cada ambiente
name	<i>String</i>	O nome público do ambiente
stack	<i>String</i>	A <i>stack</i> do ambiente
user	<i>String</i>	O Universally Unique Identifier (UUID) do utilizador que criou o ambiente
fqdn	<i>String</i>	O endereço em que o ambiente está disponível
port	<i>String</i>	A porta em que o ambiente está disponível
createdAt	<i>DateTime</i>	A data de quando o ambiente foi criado

### 4.1.3 CAMADA DE ABSTRAÇÃO DE AUTENTICAÇÃO

O módulo de abstração para proceder à autenticação do sistema, é através de uma API REST desenvolvida com a *framework* Node.js<sup>5</sup> e Javascript como linguagem de programação. A escolha desta tecnologia deve-se ao facto de a comunidade ter desenvolvido um *middleware* para diferentes métodos de autenticação de fácil manutenção, e que permite criar um módulo para o protocolo que a UA utiliza.

A instituição usa uma versão muito específica OAuth, a versão 1.0a, o serviço é denominado por IdentityUA, uma abstração a um outro serviço, um IdP[76].

---

<sup>5</sup><https://nodejs.org>

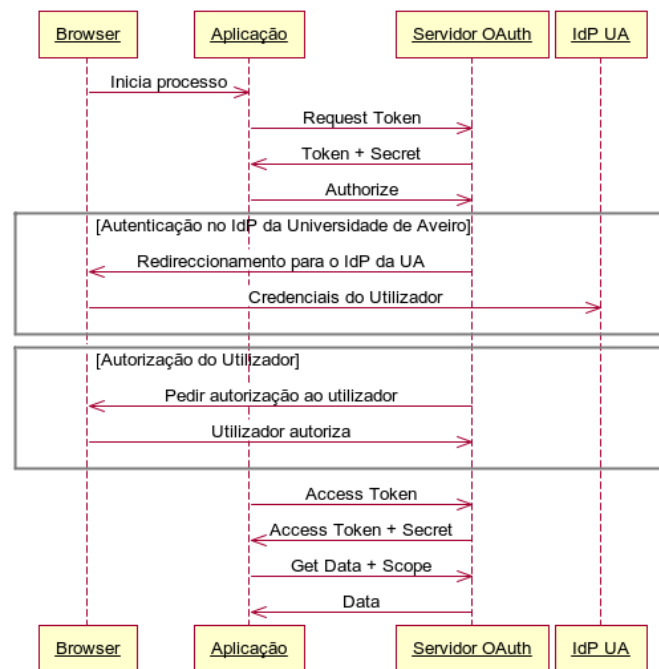


Figura 4.4: Fluxo de interação com o servidor de Identidades da UA[82]

Na imagem da arquitetura, o IdentityUA é representado pelo bloco genérico denominado de serviço de identidade, devido a utilização do *middleware* permite ter vários módulos de autenticação a funcionar em paralelo, resumidamente, através deste componente o utilizador terá de usar as credenciais do UU para poder autenticar na nova plataforma (OAuth *application*) e assim aceder ao novo serviço.

O *middleware* é o Passport<sup>6</sup> o qual tem uma integração fluída com outra *framework* de Node.js, Express<sup>7</sup> uma *framework web*. Devido à integração, o Express, foi a escolha para implementar a API REST que abstrai o *middleware*.

Como foi referido anteriormente, a UA usa uma versão muito específica do OAuth (para relembrar 1.0a), para que o *middleware* funcionasse nesta versão foi necessário desenvolver uma *Strategy* para a instituição. Ao *plugin* desenvolvido para a *framework* Passport, é denominado de *Strategy*.

Para além disso, foi adicionado ao projeto um gestor dinâmico de *strategies*, quando este módulo foi desenvolvido, o Passport, declarava os métodos de autenticação de forma explicita, é necessário inicializar o módulo e depois atribuir uma instância de uma *strategy* a um *endpoint* (URL). Isto afeta a flexibilidade da plataforma e para evitar que o Administrador tenha de corrigir código-fonte aquando a adição ou alteração de serviços de autenticação, o código quando executado interpreta um ficheiro de configuração que irá fazer o *load* dos serviços implementados.

<sup>6</sup><http://passportjs.org>

<sup>7</sup><http://expressjs.com>



Desta forma, a manutenção deste serviço de abstração poderá ser feita através de um ficheiro de configuração. Apesar da filosofia aqui discutida, atualmente a plataforma só tem os módulos de *username/password* e IdentityUA (OAuth1.0a). Não invalidando o que foi dito anteriormente, no protótipo atual apenas são utilizados estes 2 métodos de autenticação, sendo que o suporte de outros, embora modelar, teria de ser adicionado posteriormente.

Tabela 4.3: Descrição dos recursos na API do serviço de autenticação

URL	Definição
GET <address>/user/<id>	Consultar um Utilizador da plataforma
DELETE <address>/user/<id>	Eliminar um Utilizador da plataforma
GET <address>/provider	Listar os serviços de autenticação ativos na plataforma
GET <address>/provider/<name>	Listar os detalhe de um serviço de autenticação ativo na plataforma
POST <address>/provider/email/login	Iniciar o processo de autenticação por sistema de <i>Email/Username</i> e <i>Password</i>
POST <address>/provider/email/logout	Terminar a sessão do sistema de <i>Email/Username</i> e <i>Password</i>
POST <address>/provider/email/register	Registrar um novo utilizador pelo sistema de <i>Email/Username</i> e <i>Password</i>
POST <address>/provider/oauth1a/login	Iniciar o processo de autenticação pelo IdentityUA
POST <address>/provider/oauth1a/logout	Terminar a sessão pelo IdentityUA
POST <address>/provider/oauth1a/callback	Abstração da página de destino ( <i>callback</i> ) registado nas OAuth Apps da IdentityUA

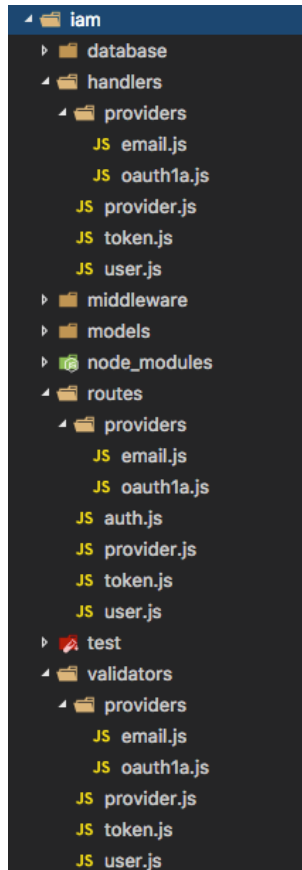


Figura 4.5: Módulos do serviço de autenticação

A possibilidade de extensão anteriormente mencionada quanto a este componente é feita através da adição de *plugins* (ficheiros Javascript) em vários diretórios (ver Figura 4.5) do componente, mais precisamente um **router** e um **handler**: o **routes/providers** é o diretório onde se adiciona o ficheiro relativamente aos *endpoints* que o serviço de autenticação irá precisar (ver bloco de código 3); e o **handlers/providers** que é o diretório que contém os ficheiros com processamento ao respetivo *endpoint* (ver bloco de código 4).

```

// imports
// variables definitions

var resolve = function (token, tokenSecret, profile, done) {
  // 'userObject' is a object constructed from information retrieved from IdP
  var query = { id: userObject.id }
  // verify if user exist to create it or update token
  User.findOneOrCreate(query, userObject, function (err, user) {
    // error handling, etc.
  })
}

// UA custom-made Passport Strategy
var strategy = new IdentityUaStrategy(options, resolve)
passport.use(strategy)
router.get('/login', passport.authenticate('identityua'))
router.post('/logout', passport.authenticate('identityua'), validate(validator.logout),
  ↪ handler.logout)
router.get('/callback', passport.authenticate('identityua'), handler.callback)

// this information enables the frontend to process icons, colors, etc.
exports = module.exports = {
  router: router,
  args: {
    embed: false,
    name: process.env.OAUTH_FULLNAME,
    icon: 'fa-university',
    style: 'btn-ua',
    url: '/auth/oauthia/login',
    redirect: true
  }
}
}

```

Código 3: Definição parcial dos *endpoints* para o serviço de OAuth do IdP da UA

No bloco de código 3 é apresentada uma representação do objeto `router` que foi desenvolvido para o *plugin* do IdP da UA, e é responsável por descrever as várias rotas para um determinado *plugin* que serão registadas na *framework web*, a definição de várias rotas permite que sejam agrupados a um caminho relativo, neste caso, foram definidas 3 rotas que serão sufixos, por exemplo, `<address>/oauth/login`. As rotas são configuradas com as funções `get`, `post` ou os outros métodos HTTP que sejam necessários, o primeiro representa o endereço, o segundo parâmetro representa a invocação do middleware (o objeto *Strategy*) de autenticação, e os seguintes parâmetros, caso existam seguem esta ordem, o `validator` que valida os *headers* e a *query* do pedido HTTP, e por fim, o `handler` que é o bloco que processa a operação adicional à autenticação a ser executado (por exemplo, revogar um token). Por fim, na última linha do bloco o `exports` como o nome indica exporta um objeto para ser importado no ficheiro *main* do componente e registo das rotas no objeto principal do servidor.

```

// imports
var handler = {}

handler.logout = function (request, reply) {
  reply.status(200).send('logout')
}

handler.callback = function (request, response) {
  var user = request.user // retrieve user information from request
  var token = new Token({ value: user.token, issuedFor: user.id })
  // create the token based on the information retrieved
  token.save(function (err, doc) {
    if (err) {
      // retrieve the error
    }

    // redirect to profile page
  })
}

exports = module.exports = handler

```

Código 4: Definição parcial do objeto *handler* usado no ficheiro de rotas

O objeto `handler` importado e utilizado no bloco de código 3 está representado no bloco de código 4, o seu formato é simples, um objeto em que não existe restrição para as propriedades declaradas (fica ao critério de quem cria o *plugin*), e cada uma corresponde a uma *callback*, estas podem ser usadas após importação do ficheiro, como é possível verificar no bloco de código anterior do `router`. Dentro de cada *callback* realizam-se as operações que forem necessárias para autenticação ou autorização do acesso, neste exemplo, na função `callback` é retirada informação do `request` (que vêm do IdP) e utiliza-se essa mesma informação para gerar um objeto `Token` e registar esta instância de `Token` na base de dados, e se não houver erros retornar uma resposta de sucesso.

#### 4.1.4 CAMADA DE ABSTRAÇÃO DA CLOUD

Este componente é a implementação de uma simples API REST para abstrair a lógica realizada pelo *Broker* e a interação com a *Cloud* privada.

A implementação do componente é desenvolvida em Python para se poder ter uma tecnologia específica no *Broker* em que o código-fonte é também em Python. A *framework* escolhida para criar a interface REST é a Flask<sup>8</sup> uma *framework* *Web* minimalista mas flexível ao mesmo tempo.

Tabela 4.4: Descrição dos recursos na API do serviço de *cloud*

URL	Definição
GET <address>/workplace	Consultar a lista de <i>Workplace</i> registados na plataforma
POST <address>/workplace	Criação de um novo <i>Workplace</i> na plataforma
GET <address>/course	Consultar a lista de <i>Course</i> registados na plataforma
POST <address>/course	Submissão de um novo <i>Course</i> na plataforma
POST <address>/stack	Criação de uma nova <i>Stack</i> para o <i>Workplace</i> na plataforma

<sup>8</sup><http://flask.pocoo.org/>

Este componente possui vários diretórios (ver Figura 4.6), os mais relevantes são: o `blueprints` é a definição dos recursos para a API; o `middleware` é o módulo responsável por fazer a comunicação com o serviço de autenticação para validar que o cliente está devidamente autenticado e autorizado para a tarefa que pediu; o `models` aonde estão declarados os modelos para a base-de-dados; e por fim, o `worker` aonde foram definidas as tarefas para serem executadas através do `broker`.

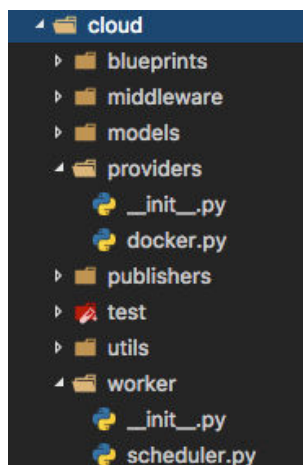


Figura 4.6: Módulos do serviço de *cloud*

A possibilidade de extensão anteriormente mencionada quanto a este componente é feita através da adição de *plugins* (ficheiros Python) ao módulo Python denominado `providers` (ver na Figura 4.6), cada ficheiro terá de possuir uma função `deploy` (ver bloco de código 5), e depois adicionar uma condição à tarefa de `deploy` do Celery (no ficheiro `scheduler.py`, ver na Figura 4.6) e invocar a função definida no *plugin*.

```
# ...
# (imports here)
# ...
# (variables defined here)

def deploy(stack, name, host_port, volumes=None):
    msg = 'docker deploy: ' + stack
    image = None
    images = docker.list_images()
    for img in images:
        if img.name == stack:
            image = img

    if image is None:
        print('docker-driver-debug: failed image is none')
        return 'failed to instance workspace'
    else:
        expose = {'80/tcp': {}}
        bindings = {'80/tcp': [{'HostPort': host_port}]}

        # docker is driver instance retrieved by libcloud framework
        res = docker.deploy_container(name, image, exposed_ports=expose, port_bindings=bindings,
                                     user='ubuntu', volumes=volumes)

    return msg
```

Código 5: Definição do método `deploy` para o serviço Docker

### 4.1.5 BROKER

Este componente implementa uma *queue* para ser possível interagir com uma larga escala de pedidos e utilizadores de forma assíncrona de forma a não criar nenhum *bottleneck* na criação dos diferentes ambientes.

A implementação da *queue* é feita recorrendo a duas tecnologias, o Celery<sup>9</sup> desenvolvido em Python, permite trabalhar a um nível mais alto, utiliza o conceito de *task* (tarefa) é necessário programar a tarefa para o *worker* do Celery a interprete e execute. A arquitetura do Celery permite executar tarefas de forma distribuída e assíncrona para isso recorre a um *Broker* e um *cluster* de *workers*, das diferentes opções tecnológicas para implementar o *Broker* foi escolhido (por afinidade) o RabbitMQ<sup>10</sup>.

Foram desenvolvidas duas tarefas: uma para fazer *deploy* (ver bloco de código 6) do ambiente de desenvolvimento e o outro para publicar o ambiente de ensino, que depois de lidar com o ficheiros invoca a função anterior para a provisão do ambiente de aprendizagem.

```
# ...
# (imports here)
# ...
# (variables defined here)

@app.task
def deploy(provider, stack, name, host_port, volumes):
    try:
        if provider == Provider.get('DOCKER'):
            # (some debug and monitor code)

            res = docker.deploy(stack, name, host_port, volumes)
            return True
        else:
            return False
    except Exception as e:
        raise e
    return False
```

Código 6: Código fragmentado da tarefa *deploy* dos ambientes

A criação do ambiente de desenvolvimento é realizado por um *middleware* denominado Apache Libcloud<sup>11</sup>, desenvolvido em Python, e que suporta alguns conceitos de *cloud*, nesta dissertação, o único que é relevante é os módulos relativamente a *Compute*, para além disso suporta imensos fornecedores de *cloud* entre os mais conhecidos Openstack, AWS EC2, Google Compute Engine (GCE), Docker, etc.

<sup>9</sup><http://www.celeryproject.org>

<sup>10</sup><http://www.rabbitmq.com>

<sup>11</sup><http://libcloud.apache.org>

#### 4.1.6 SERVIÇO DE CLOUD

O grupo de Investigação, Aveiro Telecommunications and Networks Group (ATNoG) ao qual estive associado durante o desenvolvimento da dissertação, possui uma *cloud* privada em Openstack na versão Icehouse. O módulo usa diferentes *hypervisors* para a criação de instâncias, por exemplo Kernel-based Virtual Machine (KVM) <sup>12</sup> ou Quick EMUlator (QEMU) <sup>13</sup>, o qual alguns projetos do grupo já os usavam para as suas VM, possivelmente poderá ser uma solução a ser utilizada no desenvolvimento do protótipo para disponibilizar os ambientes.

Entretanto, uma outra tecnologia emergia no domínio da virtualização, nomeadamente Docker<sup>14</sup>, com esta tendência a ganhar tração e adeptos surgiu a discussão qual melhor opção para virtualizar os diferentes ambientes que a plataforma precisa.

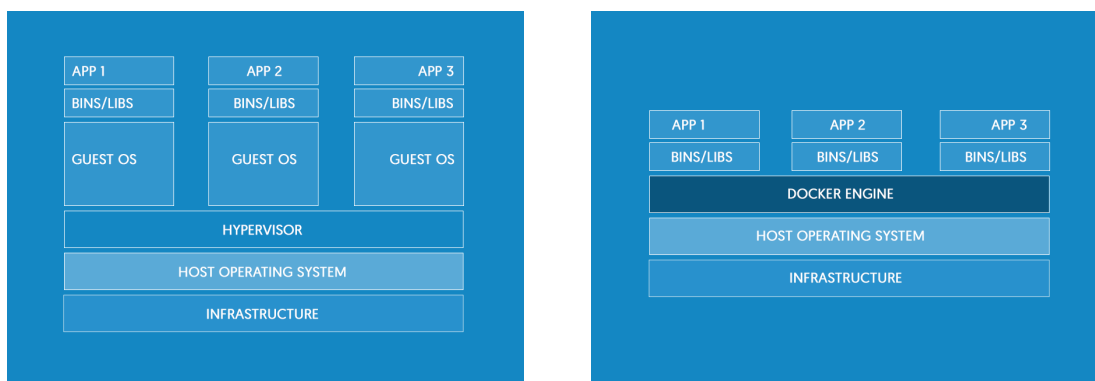


Figura 4.7: Comparação entre Virtual Machines (genéricas) e Docker Containers[83]

A diferença entre os dois é visível na Figura 4.7, digamos que é retirada uma camada entre a tecnologia de virtualização e o processo ou aplicação. A utilização de Docker tem uma elevada relevância porque a criação de ambientes em grande escala torna-se mais leve a nível de recursos consumidos. Por exemplo, enquanto que pelos meios tradicionais se disponibiliza dois ambientes de desenvolvimento de Python requer ambos sejam criados em separado e em duplicando as dependências para cada ambiente, a utilização de Docker permite a partilha (por camadas) dessas dependências fazendo com que os recursos usados não sejam multiplicados pelo número de instâncias “iguais”.

Dada a esta característica tentou-se usar Docker na versão Icehouse do Openstack, um desafio que não teve um fecho positivo, na medida em que esta versão possuía conflitos com a utilização de Docker. Então o desenvolvimento deste componente prosseguiu com uma *cloud* baseada puramente em Docker Engine<sup>15</sup> ou num cluster (com Docker Swarm<sup>16</sup>).

Desta forma, disponibilizou-se a API REST do Docker Engine para que os módulos anteriormente descritos possam interagir de forma distribuída de modo a colaborarem para a criação e manutenção do ciclo de vida dos *containers*.

<sup>12</sup><http://www.linux-kvm.org>

<sup>13</sup><http://qemu.org>

<sup>14</sup><https://www.docker.com>

<sup>15</sup><https://www.docker.com/products/docker-engine>

<sup>16</sup><https://www.docker.com/products/docker-swarm>

## 4.2 PROVISÃO

Como foi mencionado em capítulos anteriores, o trabalho foi realizado no âmbito do grupo ATNoG, ou seja, tirando partido da sua infraestrutura e recursos e que disponibilizou a sua plataforma de virtualização para criar as VMs necessárias à provisão da solução, este grupo usa como plataforma de virtualização é o Proxmox Virtual Environment<sup>17</sup>, um projeto *open source* e grátis que cria VMs com base na tecnologia KVM. O *template* que o grupo disponibiliza aos seus alunos e membros para a criação de VM possui as características descritas na Tabela 4.5.

Tabela 4.5: *Template* das VM na plataforma Proxmox

Característica	Valor
Processamento	1 CPU
Memória	512MB (488MB)
Armazenamento	30GB
Tecnologia de virtualização	KVM
Sistema operativo	Linux 4.X / 3.X / 2.6 Kernel

De notar que na linha da memória existem dois valores, isto deve-se ao fato da interface de monitorização do Proxmox e também na descrição do *template* mencionarem o primeiro valor, mas depois da VM estar iniciada, o valor efetivo considerado pelo sistema operativo da VM é de 488MB.

A utilização desta plataforma não possui requisitos ao nível do número de máquinas, desta forma a provisão da plataforma será feita tendo em mente um cenário de testes que simule um cenário de produção, isto é, distribuir os componentes entre várias máquinas na infraestrutura (ver Figura 4.8) mas tendo em conta que o poder de processamento delas é limitado. Desta forma, independentemente de a tecnologia Docker ser usada como dependência para uma funcionalidade desta plataforma, também foi usada para disponibilizar mais do que um serviço por máquina de forma fácil de provisão (utilizando um simples Dockerfile por componente).

---

<sup>17</sup><https://www.proxmox.com>



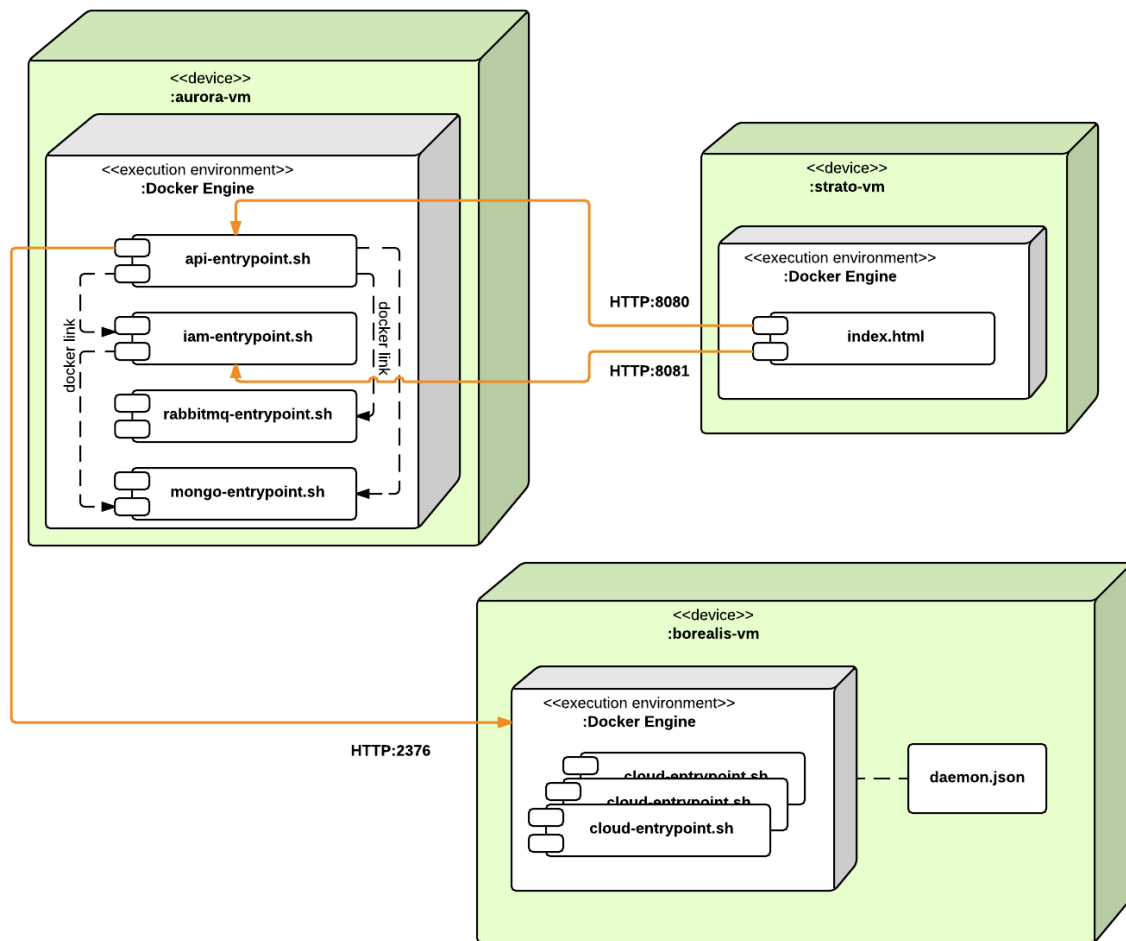


Figura 4.8: Diagrama de provisão da plataforma

Foram utilizadas três máquinas, a provisão dos serviços da plataforma teve a seguinte distribuição, na máquina “Aurora” está disponível uma instância de MongoDB, uma instância de RabbitMQ, uma instância do serviço de autenticação e uma instância do serviço de *cloud*, ligadas entre si usando uma rede privada no ambiente Docker, através de uma funcionalidade do Docker denominada *link*. Na máquina de “Borealis” possui simplesmente um ambiente Docker (versão 1.11) com a HTTP API ativa, e a pedido dos outros componentes os vários ambientes de desenvolvimento/aprendizagem serão provisionados nesta máquina. Por fim, a última máquina “Strato” só contém a página dinâmica, que serve de interface gráfica à plataforma.

## 4.3 DEMONSTRAÇÃO

Nesta secção pretende-se apresentar os resultados da implementação da plataforma do ponto de vista do utilizador, o objetivo é demonstrar as possíveis interações que os utilizadores, tanto o aluno como o docente, podem realizar quando acedem à plataforma. A exposição da aplicação terá um formato ao género de guia e evidenciar alguns detalhes a cada interação, e com recurso a alguns diagramas para uma melhor compreensão deles. Para isso, vamos supor que o docente está a preparar uma aula sobre conceitos de recursividade, e o aluno está a trabalhar num projeto para um disciplina de introdução a Python.

### 4.3.1 AUTENTICAÇÃO

O aluno para poder iniciar o seu projeto em Python, tem de aceder à plataforma pelo *browser* e caso não esteja autenticado será redirecionado para o ecrã na Figura 4.9, nesta interface o aluno têm à sua disposição dois métodos de autenticação, neste exemplo, os métodos por *Username/Password* e o método que utiliza as credencias de UU da UA. Possivelmente se houvessem outros métodos de autenticação disponíveis, como redes sociais como o Twitter, estariam dispostos em formato de botão com uma configuração semelhante ao que diz “Universidade”.

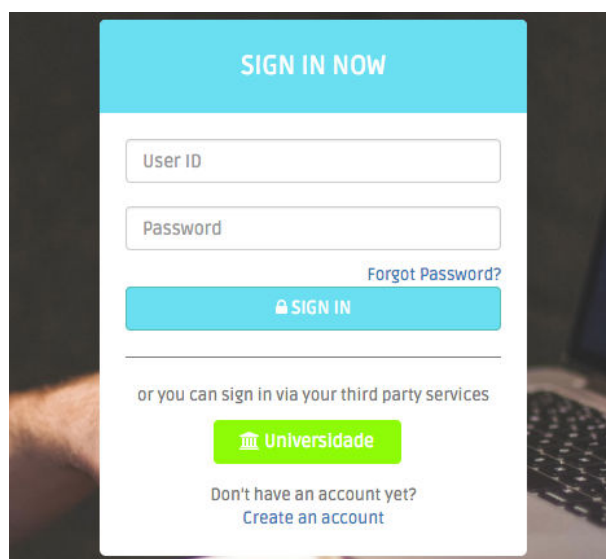


Figura 4.9: Página de *login* da aplicação *web*

O diagrama da Figura 4.10 explica as diferentes interações entre o aluno (“Bob”) e as restantes partes que compõem o sistema, o *screenshot* é o resultado das primeiras interações entre o Utilizador e a aplicação *web* (“WebApp”) que resulta na renderização da página que que é possível verificar; depois do utilizador escolher uma opção a camada de abstração de autenticação (“AuthAPI”) é o intermediário e redireciona para o destino correto, se for o primeiro método este componente de autenticação gera os dados necessários para manter para futuras interações, se tiver de se autenticar em serviços externos (“ThirdParty”), por exemplo IdentityUA, o componente interage com o serviço de autenticação em causa e envia a resposta para os componentes anteriores até que o utilizador seja redirecionado para a página oficial do serviço, desta forma os diferentes componentes nunca terão acesso às credenciais, só aos dados que os serviços externos o retornam após a autorização de acesso que neste caso são só *email* e *token*, após a autorização ser efetuada, independentemente do método, a aplicação *web* termina o processo redirecionado para a página de perfil.

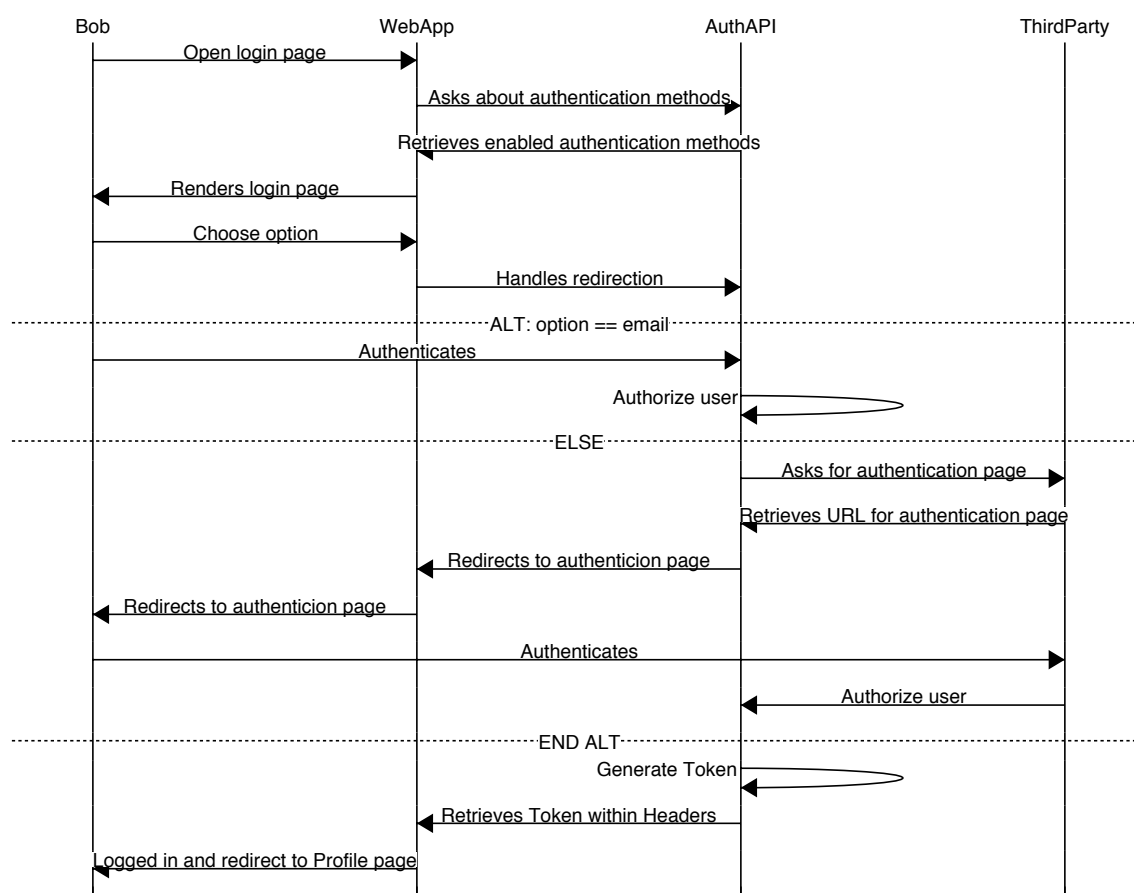
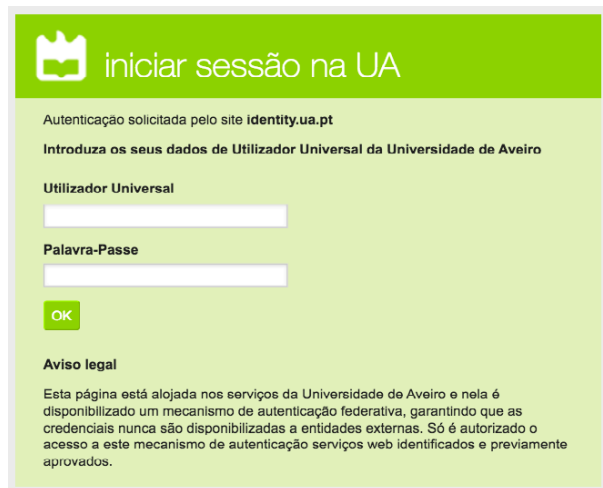


Figura 4.10: Diagrama de sequência do processo de autenticação a partir da aplicação *web*

Na prática o aluno após clicar no botão “Universidade” da Figura 4.9 irá ser redirecionado para a página do IdP da UA, e nesta nova página o aluno irá proceder a autenticação digitando o *email* e a *password* no formulário disponível (ver Figura 4.11).



iniciar sessão na UA

Autenticação solicitada pelo site identity.ua.pt

Introduza os seus dados de Utilizador Universal da Universidade de Aveiro

Utilizador Universal

Palavra-Passe

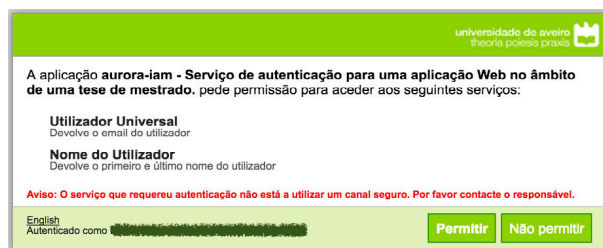
OK

**Aviso legal**

Esta página está alojada nos serviços da Universidade de Aveiro e nela é disponibilizado um mecanismo de autenticação federativa, garantindo que as credenciais nunca são disponibilizadas a entidades externas. Só é autorizado o acesso a este mecanismo de autenticação serviços web identificados e previamente aprovados.

Figura 4.11: Página de autenticação da UA

Se a autenticação for bem sucedida, o aluno irá ver uma nova página (ver Figura 4.12) que descreve as informações que a aplicação a qual se está a ligar precisa de aceder, e fornece duas opções ao aluno, autorizar o acesso e assim prosseguir para a plataforma ou negar o acesso a esses dados e interromper o processo de autenticação e autorização e é lhe negando o acesso à plataforma.



universidade de aveiro  
theoria potestis praxis

A aplicação aurora-iam - Serviço de autenticação para uma aplicação Web no âmbito de uma tese de mestrado, pede permissão para aceder aos seguintes serviços:

**Utilizador Universal**  
Devolve o email do utilizador

**Nome do Utilizador**  
Devolve o primeiro e último nome do utilizador

**Aviso:** O serviço que requereu autenticação não está a utilizar um canal seguro. Por favor contacte o responsável.

English  
Autenticado como [nome de usuário]

Permitir Não permitir

Figura 4.12: Página de autorização para a plataforma

Quando o processo de autenticação e autorização terminar o aluno é redirecionado para a página principal (ver Figura 4.13) da plataforma com um simples quadro em que identifica o número de cursos e de ambientes disponíveis.

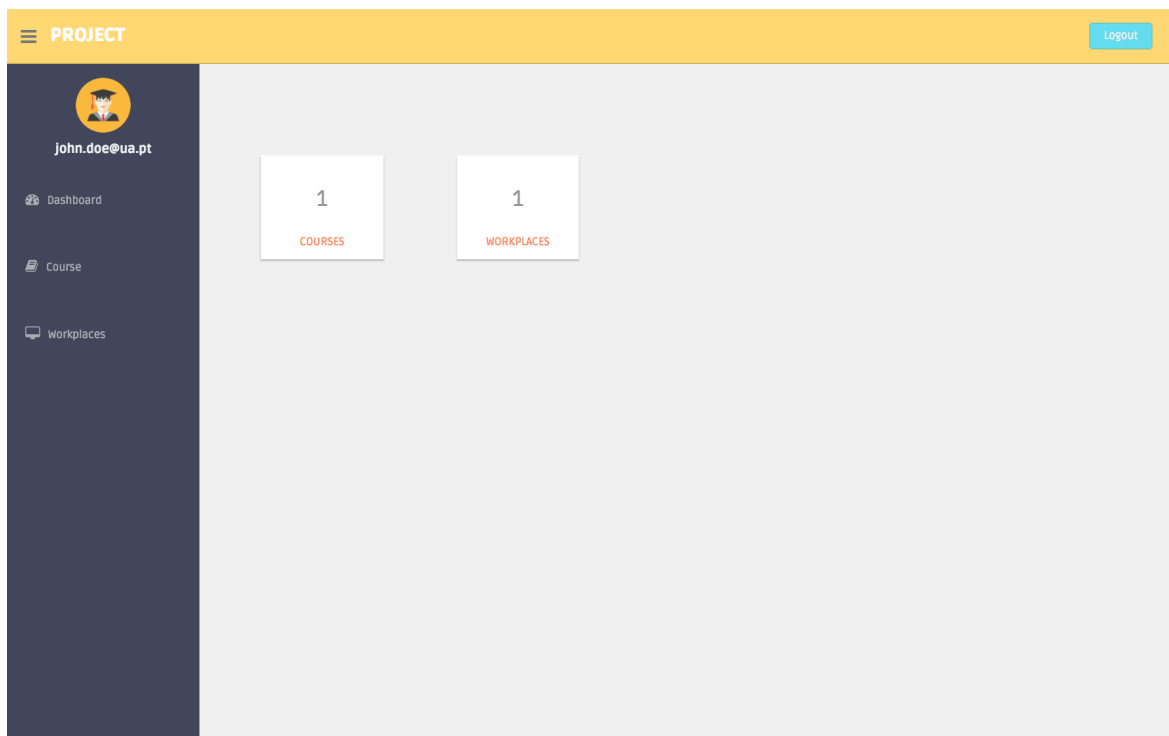


Figura 4.13: Página principal da plataforma

### 4.3.2 ACEDER AO AMBIENTE DE APRENDIZAGEM

Quando o aluno é redirecionado para a página principal é confrontado com um menu de navegação ao lado esquerdo na interface (ver Figura 4.14). Das 3 opções, se quiser aceder a um ambiente de aprendizagem (ou curso) disponível, tem de clicar na opção *Courses*.

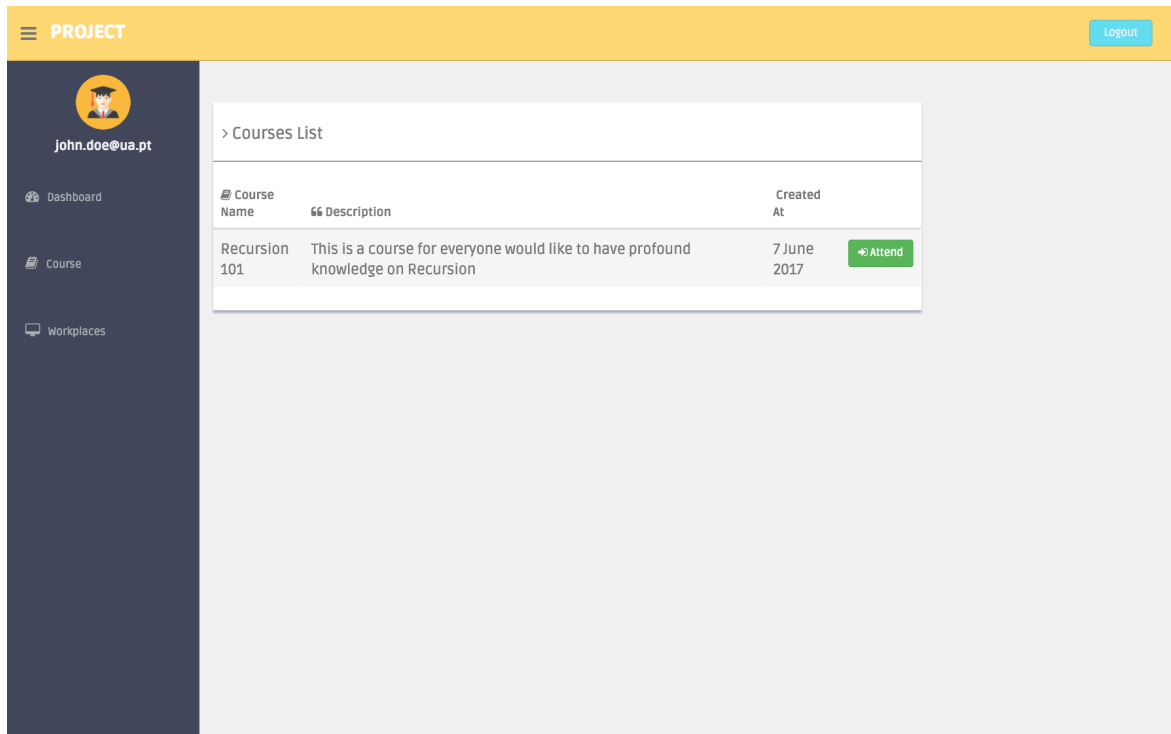


Figura 4.14: Página que contém a lista de ambientes de aprendizagem da aplicação *web*

Nesta página existe uma lista de vários cursos e em cada linha da lista a informação disponível é o nome e a descrição. Para participar neste curso o aluno terá de clicar no botão “Attend” que está à direita do nome do curso que deseja participar.

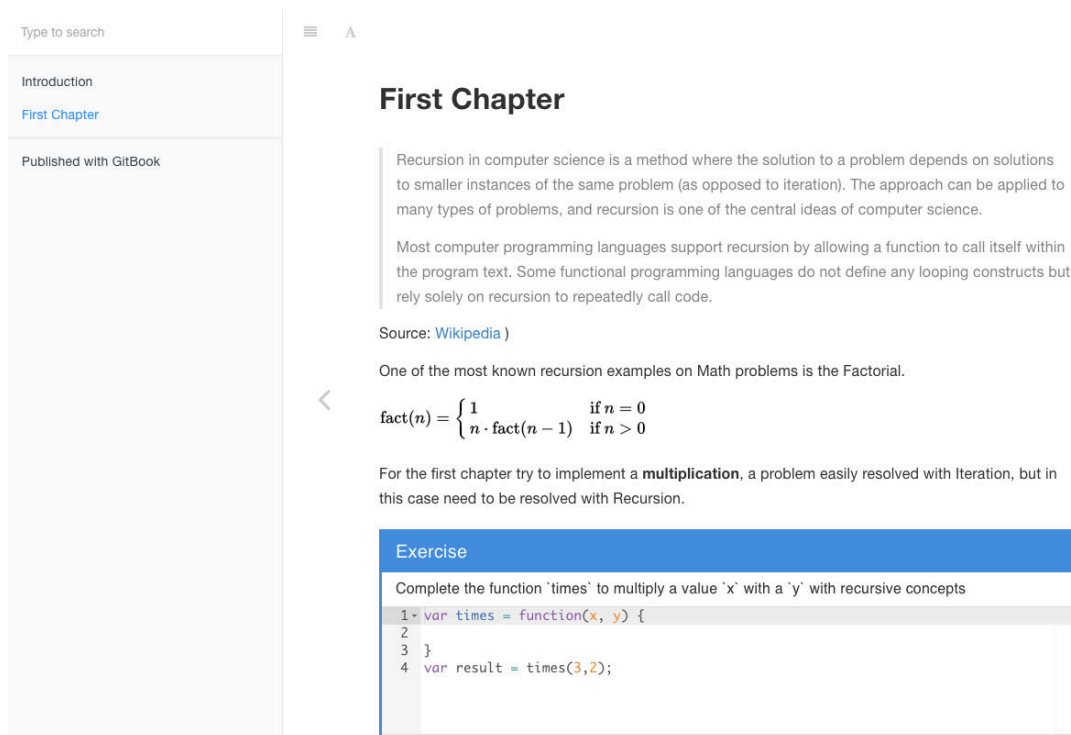


Figura 4.15: Página (parcial) de um ambiente de aprendizagem aberto

No curso acessado (ver Figura 4.15) é composto por vários capítulos que estão enumerados à esquerda da interface e neste curso em específico logo no primeiro capítulo é possível verificar que existe um conteúdo teórico acompanhado de conteúdo prático. Depois da aquisição do conhecimento teórico, o aluno pode por em prática o que leu utilizando a consola interativa. As consolas estão representadas por umas caixas azuis e permite ao aluno escrever a sua resposta ao exercício. Ao escrever a resposta na consola basta validar clicando no botão “Submit” receberá a notificação quanto ao estado da resolução. Em caso de dificuldades na resolução do exercício ou dúvidas quanto ao que o exercício pretende, o aluno pode receber uma ajuda com o botão “Solution”, que completa o exercício por si.

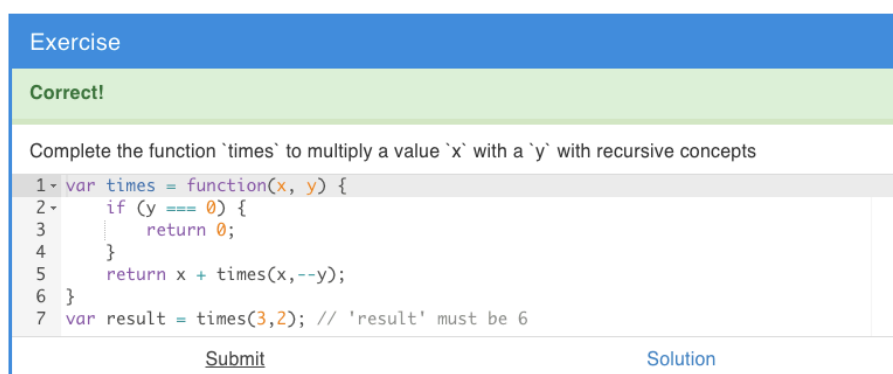


Figura 4.16: Bloco de um exercício com resultado positivo de um ambiente de aprendizagem

O *feedback* da resolução do exercício pode vir em dois formatos uma notificação verde em caso de sucesso (ver Figura 4.16), e em caso de uma resposta errada aparece a notificação a vermelho (ver Figura 4.17).

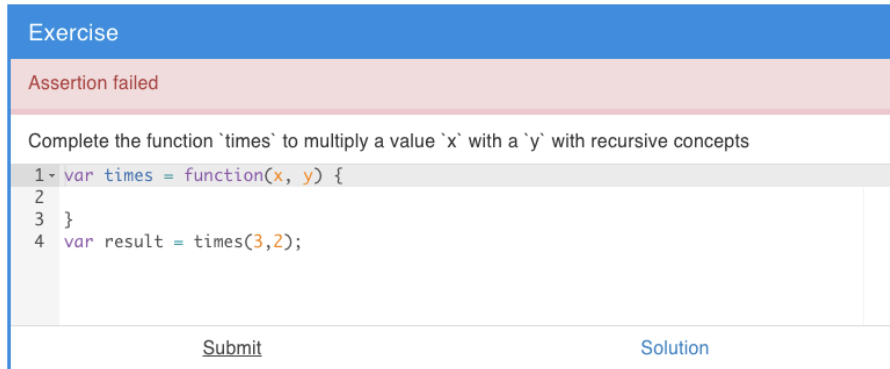


Figura 4.17: Bloco de um exercício com resultado negativo de um ambiente de aprendizagem



### 4.3.3 ACEDER AO AMBIENTE DE DESENVOLVIMENTO

Neste momento o aluno em causa, já participou na aula, e agora tem um trabalho de uma outra disciplina para fazer, nesta disciplina hipotética, o trabalho é para ser realizado em Python. Posto isto, o aluno acede à plataforma, e na página principal, através do menu de navegação à esquerda da interface, terá de escolher a opção *Workplaces*, que lhe permite aceder à funcionalidade de gestão de ambientes de desenvolvimento (ver Figura 4.18).

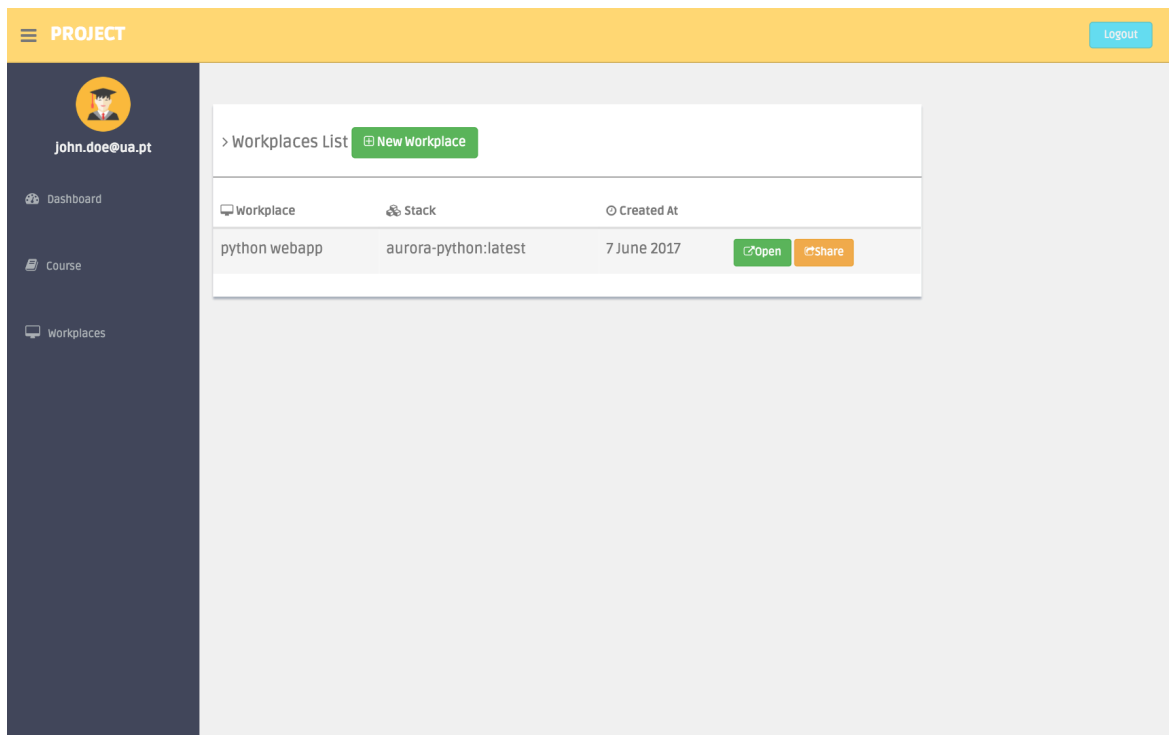
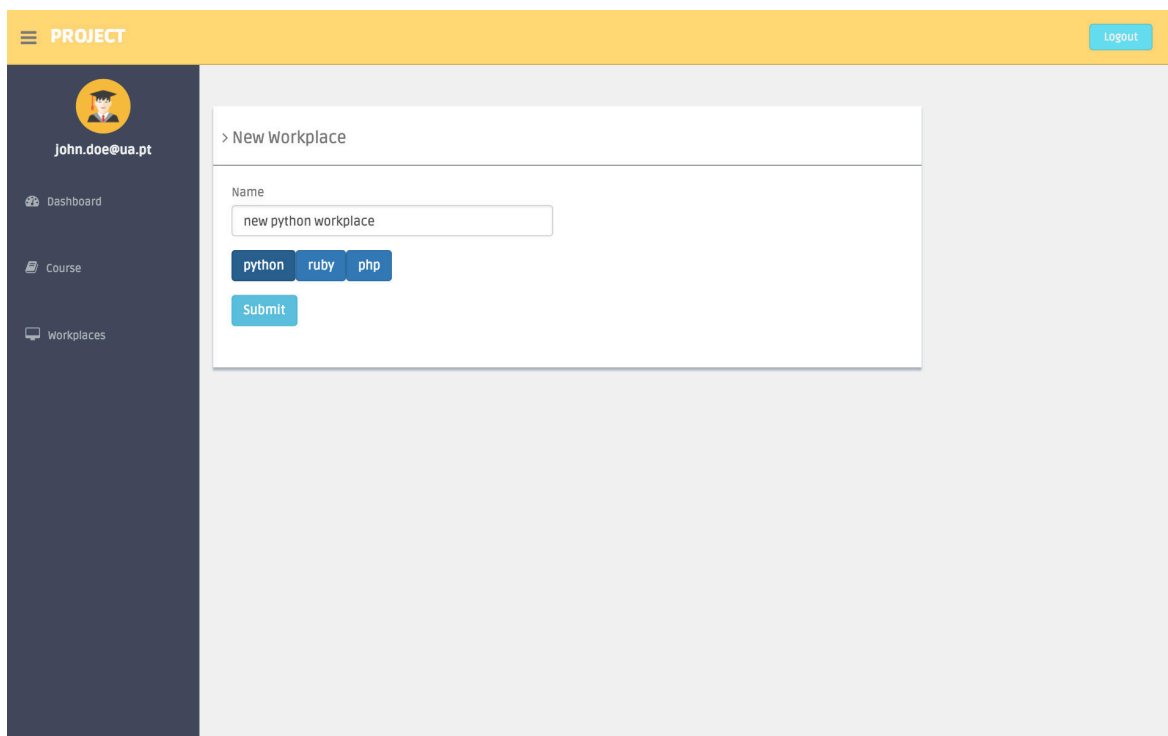


Figura 4.18: Página que contém a lista de ambientes de desenvolvimento da aplicação *web*

Para criar um novo ambiente de desenvolvimento o aluno, terá de clicar no botão “New Workplace” situado no topo da página, que o redireciona para uma nova página (ver Figura 4.19) onde pode configurar o seu novo ambiente de desenvolvimento. As opções de configuração incluem o nome e a *stack/template* que serve para disponibilizar um ambiente já com algumas dependências relacionadas com a *stack* escolhida.



The screenshot shows a web application interface for creating a new development environment. At the top, there is a yellow header with the word "PROJECT" on the left and a "Logout" button on the right. A dark blue sidebar on the left contains a user profile for "John.doe@ua.pt" and navigation links for "Dashboard", "Course", and "Workplaces". The main content area is titled "> New Workplace" and contains a form with a "Name" field filled with "new python workplace". Below the name field are three buttons labeled "python", "ruby", and "php", with "python" selected. A "Submit" button is located at the bottom of the form.

Figura 4.19: Página para a requisição de um novo ambiente de desenvolvimento

Para terminar o pedido de novo ambiente, basta clicar no botão “Submit”, e será reenviado para a página que contém a lista de ambientes de desenvolvimento disponíveis. Existe um tempo de espera até o ambiente de desenvolvimento estar completamente disponível, este período de espera não possui um elemento gráfico (uma *loading bar* ou notificação), no mínimo o aluno terá de esperar cerca de 2 minutos até o mesmo estar pronto a ser acedido.

Para abrir um ambiente de desenvolvimento basta no clicar no botão “Open” e a aplicação *web* redireciona para uma nova página, o IDE com uma página de boas vindas (ver Figura 4.20).

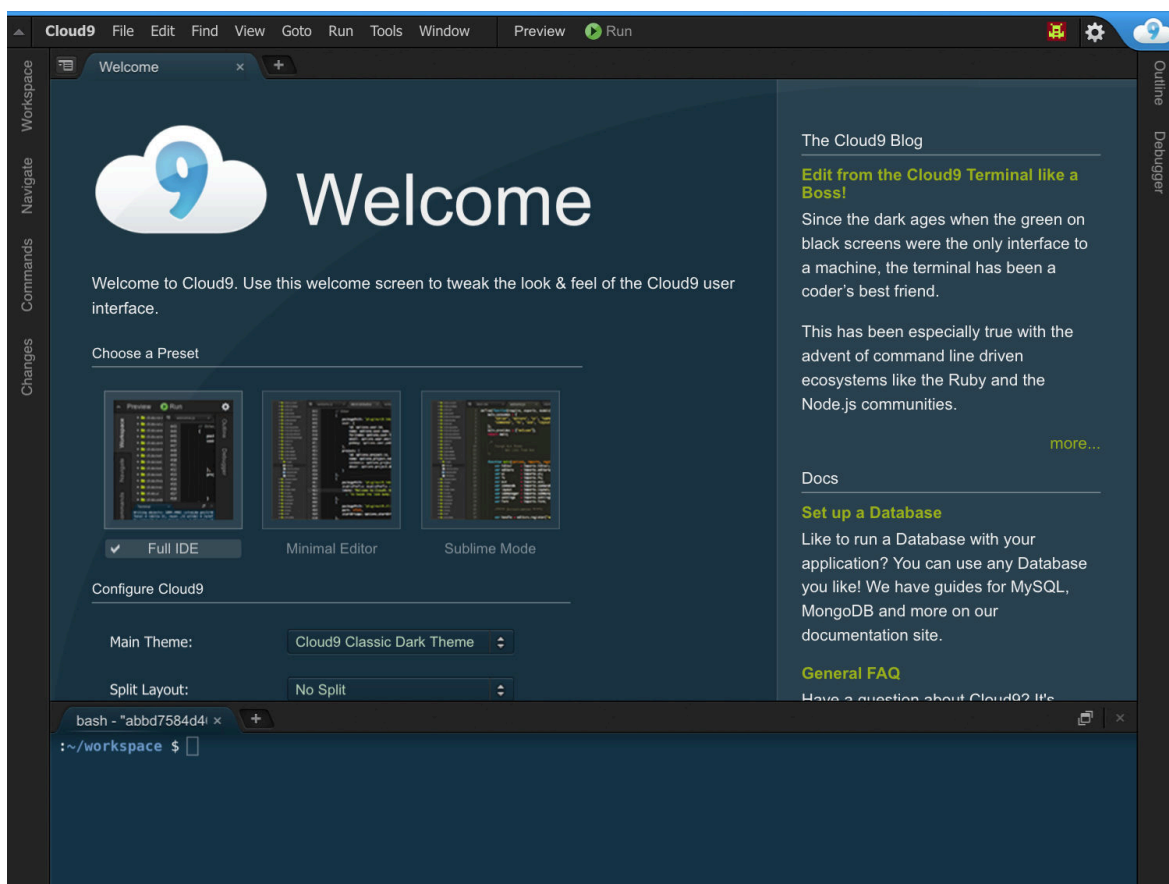


Figura 4.20: Página de boas vindas do ambiente de desenvolvimento

Na Figura 4.20 é possível verificar que do lado esquerdo da interface está o gestor de ficheiros do projeto semelhante aos explorador de ficheiros dos editores de texto mais conhecidos. No centro da página, quando um ficheiro estiver aberto, o utilizador tem acesso ao editor de texto com alguns extras como *syntax highlight* e as outras funcionalidades mencionadas na análise na Seção 3.1. O editor usa um sistema de *tabs* para permitir ao utilizador interagir com diferentes ficheiros abertos, do lado direito, permite ao utilizador acionar as operações de *debugging*.

Na parte de baixo, o utilizador tem um conjunto de *tabs*: o terminal (como se pode ver ler *bash*) embutido e permite ao utilizador aceder ao sistema da VM inerente a este ambiente de desenvolvimento (ver Figura 4.21), e apesar de não estar visível existe a possibilidade de invocar uma consola interativa (também denominada por Read-Eval-Print Loop (REPL) de Javascript). Na parte de cima estão disponíveis vários menus com várias operações comuns de criar ficheiros, seleccionar, copiar texto, etc.

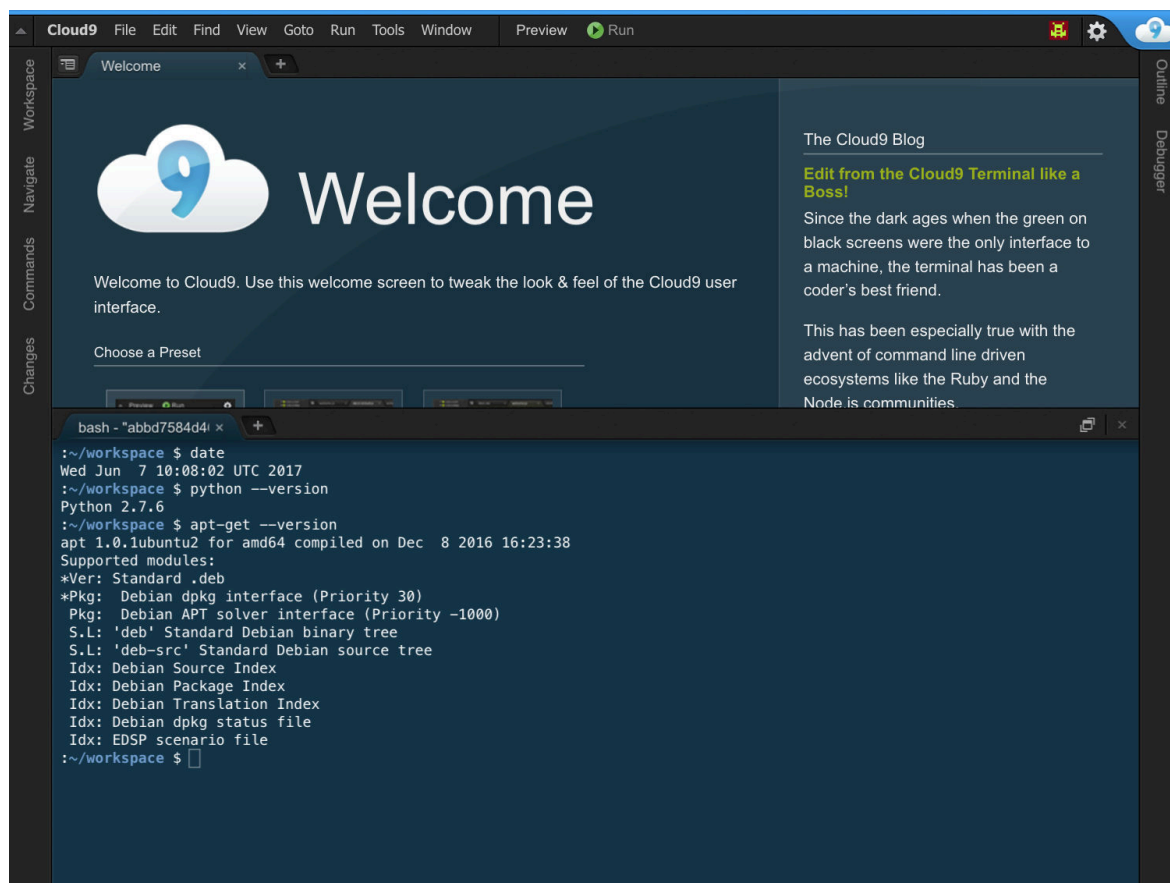


Figura 4.21: Interação com o terminal do ambiente de desenvolvimento

Neste momento, o aluno já tem o ambiente aberto e pode começar o seu trabalho, de notar que a escolha de uma *stack* já instala várias dependências para essa linguagem, mas neste caso o aluno quer definir o teu ficheiro de dependências e cria um novo ficheiro ou pela combinações de teclas CTRL+N ou pelo menu **File > New File**.

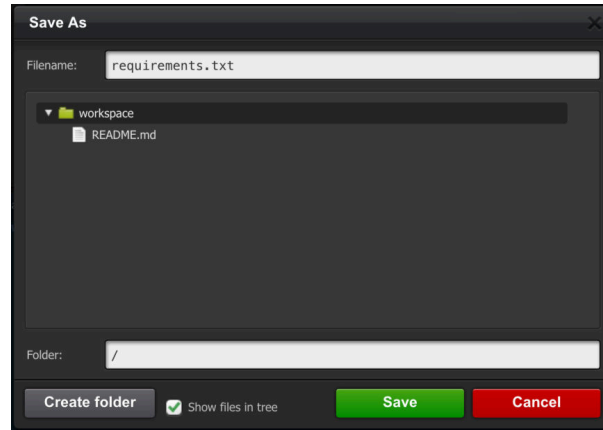


Figura 4.22: Caixa de diálogo para guardar um novo ficheiro

Para guardar o resultado basta usar a combinações de teclas CTRL+S ou o menu na parte superior da interface. E aparece uma caixa de dialogo para que o utilizador possa definir as condições em que deseja guardar o ficheiro (ver Figura 4.22).

A seguir o passo lógico do aluno é dar início ao desenvolvimento do seu projeto, no caso de cometer um erro de sintaxe enquanto está a escrever o código que pretende, como foi identificado anteriormente o IDE faz verificação de erros, enquanto se escreve (ver Figura 4.23).

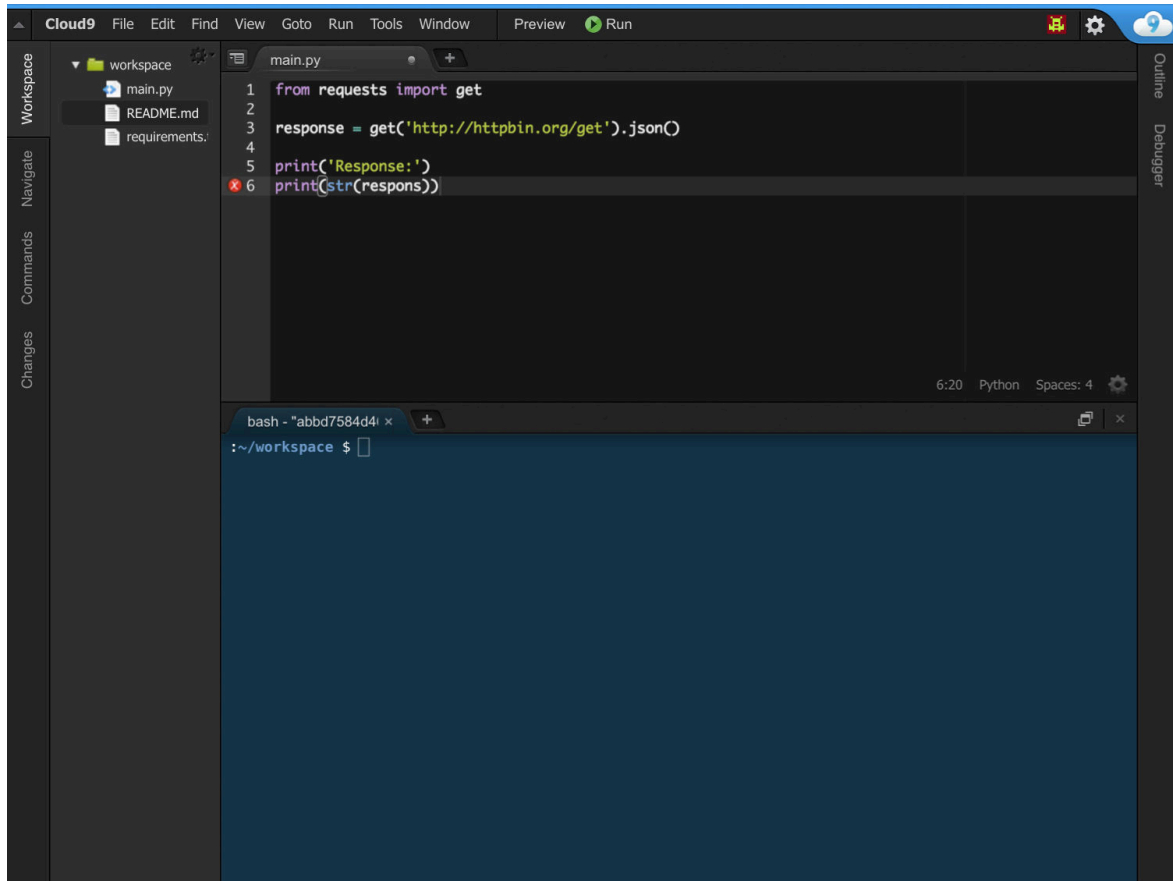
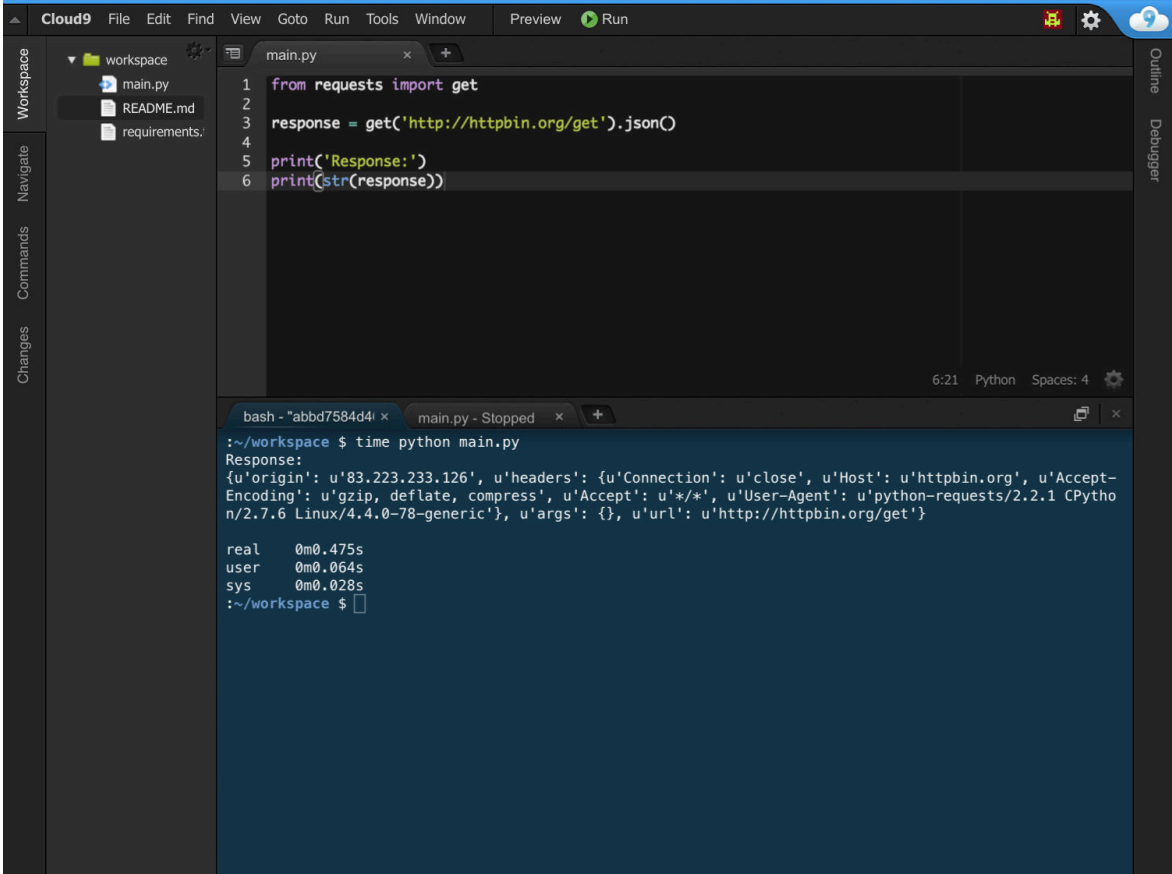


Figura 4.23: Alerta do IDE para uma linha com erro de código

Depois de o código estar resolvido e o ficheiro gravado o aluno pode executar o seu projeto de duas formas, a primeira é pelo terminal e escreve manualmente o comando a ser executado (ver Figura 4.24) ou então pelo botão “Run” no topo da página e o IDE resolve automaticamente que ficheiro executar (ver Figura 4.25).



```
Cloud9 File Edit Find View Goto Run Tools Window Preview Run
workspace
main.py
README.md
requirements.
main.py
1 from requests import get
2
3 response = get('http://httpbin.org/get').json()
4
5 print('Response:')
6 print(str(response))
6:21 Python Spaces: 4
bash - "abbd7584d4" x main.py - Stopped x +
~/workspace $ time python main.py
Response:
{'u'origin': u'83.223.233.126', u'headers': {u'Connection': u'close', u'Host': u'httpbin.org', u'Accept-
Encoding': u'gzip, deflate, compress', u'Accept': u'*/*', u'User-Agent': u'python-requests/2.2.1 (Pytho
n/2.7.6 Linux/4.4.0-78-generic)}, u'args': {}, u'url': u'http://httpbin.org/get'}

real    0m0.475s
user    0m0.064s
sys     0m0.028s
~/workspace $
```

Figura 4.24: Execução do projeto pelo terminal

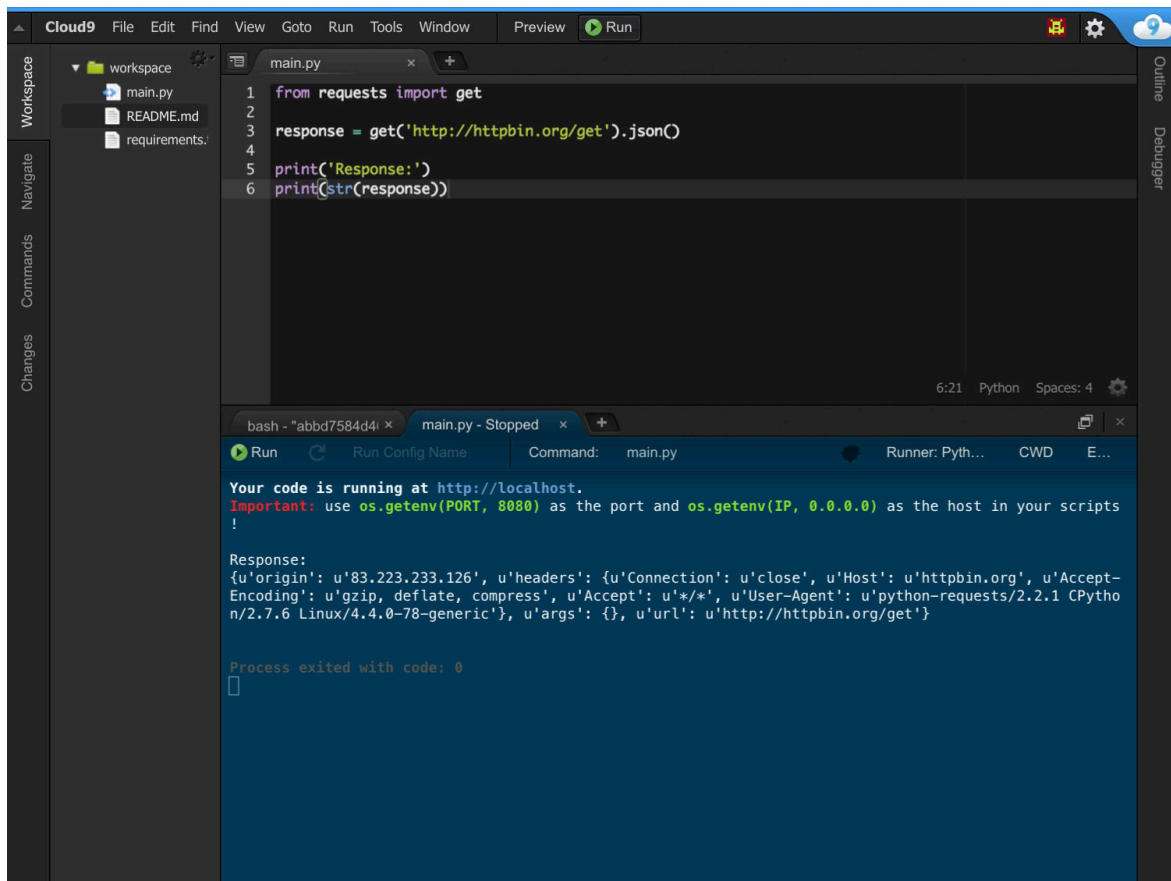


Figura 4.25: Execução do projeto pelo botão “Run”



### 4.3.4 SUBMETER AMBIENTE DE APRENDIZAGEM

O docente desta demonstração quer criar e disponibilizar um curso focado ao conceito da recursividade, a funcionalidade para esta operação está diferente da proposta em Seção 2.7.2, por não estar disponível um editor com uma interface *web* como as restantes funcionalidades. Assim sendo a criação dos conteúdos é feita pelo editor em versão *desktop*<sup>18</sup> e o resultado exportado para o sistema de ficheiros.

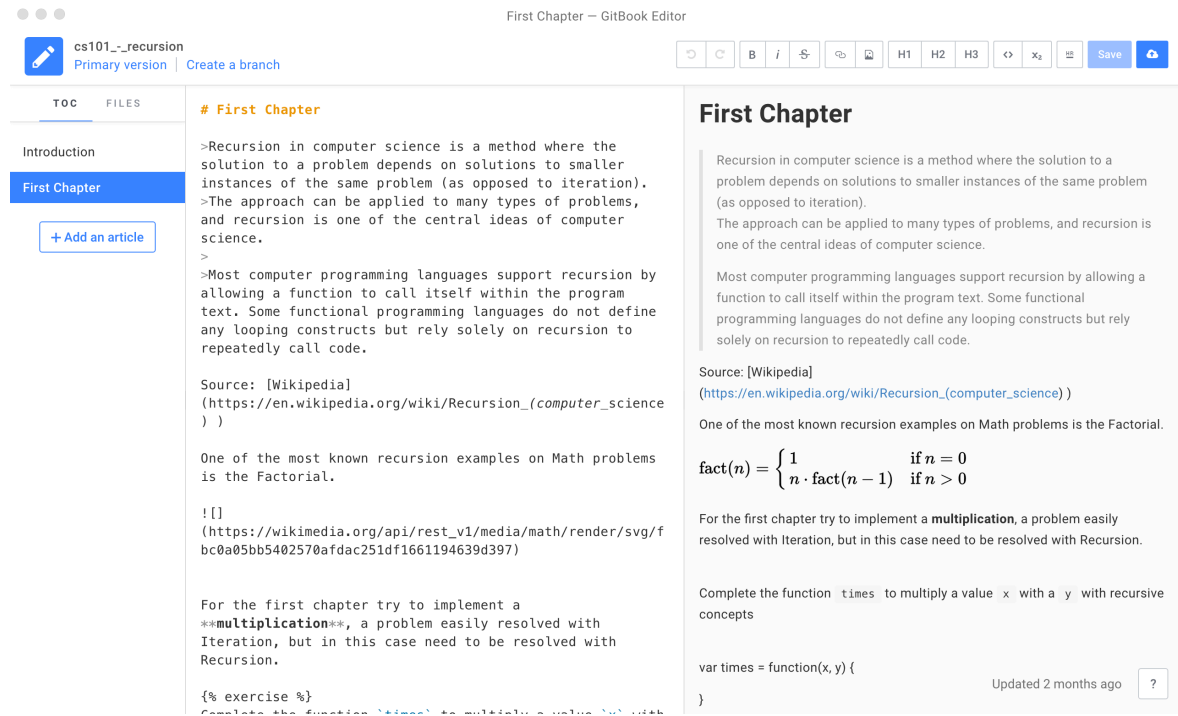


Figura 4.26: Criação do curso de Recursividade no editor do Gitbook

Na Figura 4.26 apresenta a interface que o docente tem de usar para a criação do conteúdo, do lado esquerdo é a lista de capítulos que compõem o livro (relembrando que capítulos e livro pertencem à taxonomia do Gitbook), estes ficheiros são salvaguardados localmente com extensões de Markdown. Na parte central, é o elemento onde o docente pode escrever o conteúdo do livro, à medida que escreve o conteúdo é compilado em tempo-real e mostrado no elemento do lado direito do editor, assim o docente sabe logo como o aluno irá ver o conteúdo que escreve. Infelizmente o conteúdo dos exercícios não consegue ser devidamente compilado pelo editor, e caixa azul que deveria aparecer não aparece, e só mostra a parte do texto do exercício.

<sup>18</sup><https://www.gitbook.com/editor>

Para converter para um novo formato interativo (HTML), o docente executa a ferramenta de conversão pelo terminal (ver bloco de código 7), `gitbook-cli`<sup>19</sup>, os conteúdos em formato HTML são produzidos.

```
gitbook build /path/to/markdown/files
```

### Código 7: Execução do comando Gitbook para exportar para HTML

Depois, o docente poderá submeter na plataforma da seguinte forma, em primeiro lugar criar o curso (ver bloco de código 8) e se o curso foi criado com sucesso submeter os ficheiros para esse curso (ver bloco de código 9).

```
curl -X POST \  
  http://api:5000/course \  
  -H 'content-type: application/json' \  
  -H 'x-authorization: _43af11da3df40c417701d25c722d74dc0195a87da1' \  
  -d '{"name": "Recursion 101",  
      "description": "This is a course for everyone would like to have profound knowledge on  
↵ Recursion"}'
```

### Código 8: Execução do comando para criação do curso

Como resposta é retornado um objeto JSON que entre várias informações, os mais relevantes é o UUID e o URL, o primeiro será usado no pedido de submissão para indicar especificamente a quem atribuir os ficheiros submetidos e o segundo é para identificar o curso com base num URL para evitar interromper o fluxo de interações para pesquisar o endereço onde ficou alojado o mesmo.

```
curl -X POST \  
  http://api:5000/course/5d4f91ea-7826-4614-b947-5e8511f71483/book \  
  -H 'content-type: multipart/form-data' \  
  -H 'x-authorization: _19397338c2058c230eede14fadf511f499dc5cb8e4' \  
  -F book=@/Users/user/Documents/book-recursion-complete.tgz
```

### Código 9: Execução do comando para submeter os ficheiros para o curso

De notar que nestes pedidos o valor do *header* `x-authorization` é referente ao *token*.

---

<sup>19</sup><https://github.com/GitbookIO/gitbook-cli>

A interação anteriormente descrita é demonstrado pelo diagrama da Figura 4.27, o utilizador (“Bob”) através de um pedido `POST` para o endereço `<address>/course` com as devidas informações no *body* do pedido para a API REST e que invoca a operação para criar o ambiente de aprendizagem (ou curso como é denominado no diagrama), isto é preparar o sistema e criar um registo na base de dados, o passo seguinte com base num “ID” retornado pelo pedido anterior e que vai fazer parte do novo pedido `POST` para o endereço `<address>/course/<id>/book` em que no *body* vão os ficheiros anteriormente gerados, a camada de abstração de *cloud* (“CloudAPI”) recebe os ficheiros e redireciona-os para o *broker* que os guarda num NFS para posteriormente serem utilizados, após guardar os ficheiros o *broker* interage com o respetivo serviço de *cloud* (“CloudService”) e identifica quais os ficheiros que o serviço de *cloud* terá de importar na criação da instância; quando a operação estiver terminada a camada de abstração de *cloud* retorna uma resposta com as respetivas informações para se aceder ao novo ambiente de aprendizagem criado. Esta operação não está disponível pela interface gráfica, só através da API.

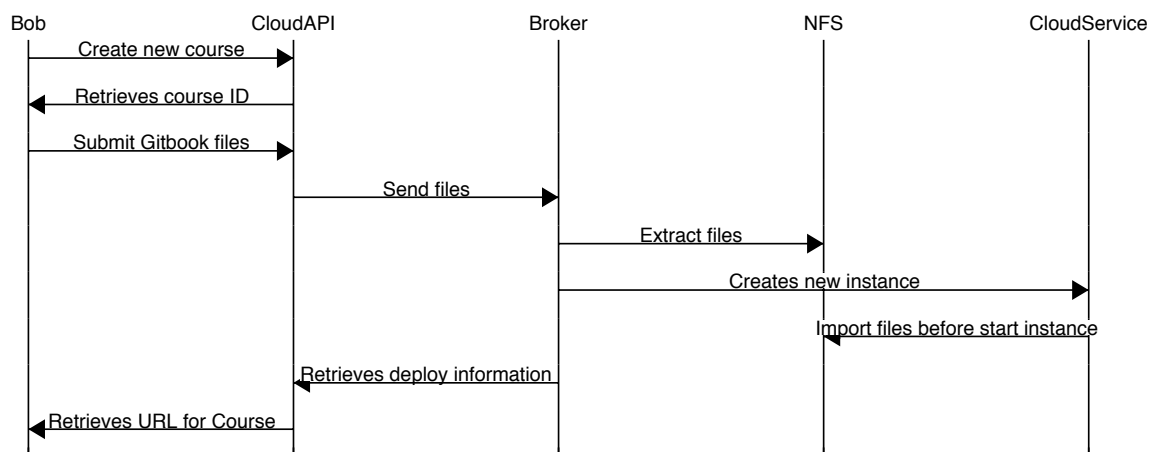


Figura 4.27: Diagrama de sequência do processo de submeter um ambiente de aprendizagem a partir da interface REST

## 4.4 CONCLUSÃO

Apesar das funcionalidades desenvolvidas e apresentadas no decorrer do presente capítulo de acordo com as funcionalidades propostas existem partes do desenvolvimento que ficaram incompletas. De forma a melhor focar os esforços no desenvolvimento e com o objetivo de realizar a prova de conceito das funcionalidades principais, no processo de gestão de tempo alguns detalhes foram colocados num segundo plano, acabando por ficarem incompletos ou não implementados, como por exemplo, a funcionalidade do contexto de utilizador que permitem distinguir o tipo de utilizador (docente ou aluno). O utilizador do tipo docente, com o estado atual da implementação ficou prejudicado porque a *web* GUI não permite redigir e publicar um novo curso devido ao facto de o editor Gitbook só existir em versão *desktop*, e é forçado a usar este editor para redigir e quando estiver terminado submete o curso através de um cliente HTTP e submeter pela API.

Em relação ao utilizador do tipo aluno, fica limitado em duas situações, a primeira é nos ambientes de aprendizagem que o *plugin* dos exercícios só suporta uma linguagem de programação, que é Javascript, em teoria existe a possibilidade de usar outras linguagens mas não foi comprovado. O *plugin* do Gitbook responsável pela funcionalidade relativa aos exercícios usa uma dependência, o JSREPL<sup>20</sup>, e que pela descrição da última versão disponível suporta várias linguagens entre elas Python e Ruby, de notar que entretanto, os responsáveis pelo projeto deixaram de manter a biblioteca, e agora disponibilizam a mesma função em formato de API HTTP. Entretanto, no projeto do *plugin* de exercícios este problema é identificado e ocorre por opção dos responsáveis porque nunca atualizaram para que a sintaxe de exercícios desse suporte a mais linguagens para além do Javascript. Para esta funcionalidade ficar completa, existem dois problemas a ser resolvidos, a manutenção do JSREPL ou a pesquisa de uma nova alternativa, e também a alteração da sintaxe do exercício e assim o criador identificar a linguagem que pretende que seja o exercício. Quanto à segunda limitação do utilizador do tipo aluno, a funcionalidade de partilha dos ambientes de desenvolvimento está incompleta de acordo com os requisitos, porque era suposto o *owner* de cada ambiente definir os utilizadores com quem gostaria de partilhar a visibilidade do conteúdo, e no resultado final os ambientes são partilhados por *link*.

---

<sup>20</sup><https://github.com/replit/jsrepl>

## RESULTADOS

---

*Dada a natureza deste projeto e dos objetivos anunciados no início do documento, existia uma preocupação que uma solução destas consuma muitos recursos para sustentar a sua infraestrutura. Deste modo, foi provisionado o protótipo, e enquanto esteve operacional num cenário limitado foram recolhidas várias métricas com suporte a diferentes ferramentas. Os valores recolhidos são apresentados no decorrer do capítulo enquanto se justifica determinados eventos e se formulam as respetivas conclusões.*

### 5.1 EXPERIÊNCIA

Após a compreensão do estado da implementação, neste capítulo partilha-se os resultados de uma experiência. A realização desta experiência pretende simular um cenário de utilização num ambiente controlado porque a solução implementada e disponibilizada é um protótipo (pelo motivos anteriormente discutidos), e sendo uma prova de conceito não está nas condições ideais para se disponibilizar ao público alvo. De qualquer forma, os recursos disponíveis para pôr a solução num cenário de produção são limitados, algo que será comprovado com as conclusões apresentadas neste capítulo.

Tendo em conta a infraestrutura mencionada no Seção 4.2, com a experiência aqui apresentada, pretende-se identificar o seu comportamento e registar o consumo de recursos durante a sua utilização. Foram formulados alguns testes com a consideração às limitações anteriormente mencionadas.

Para monitorizar as máquinas, existiam várias ferramentas ou usar soluções com formato agente-servidor, foi eleita uma simples ferramenta que agrega várias funções de monitorização de armazenamento, processamento, memória e rede. A ferramenta é `atop`<sup>1</sup> e foi instalada em cada máquina. O comando executado regista uma amostra de valores por cada segundo durante 7200 segundos e cada amostra contém informações de processamento, memória, disco e rede (ver o bloco de código 10).

---

<sup>1</sup><https://www.atoptool.nl>

```
// -g retrieve generic/process values
// -m retrieve memory values
// -d retrieve disk values
// -n retrieve network values
atop -w my.log -g -m -d -n 1 7200
```

### Código 10: Exemplo do comando de monitorização

A experiência consiste em solicitar vários ambientes de desenvolvimento (ver bloco de código 11) a partir da HTTP API do serviço de *cloud* disponibilizado na máquina “Aurora” (ver Figura 4.8), registrando as várias etapas na base de dados, desde a realização do pedido até ao momento em que o mesmo ambiente fica disponível para ser utilizado pelo aluno, como se fosse um teste de integração (*end-to-end*). A utilização de um cliente HTTP como `curl`<sup>2</sup> com o objetivo de remover uma possível entropia do portal ou do servidor *web* durante o teste, porque o foco do teste é relativamente à criação dos ambientes por ser uma operação complexa.

```
curl -X POST http://aurora.vm.atnog.av.it.pt:8080/workplace -H 'content-type: application/json' -H
↳ 'x-authorization: _19397338c2058c230eede14fadf511f499dc5cb8e4' -d '{"stack":
↳ "aurora-python:latest", "user": "bbe04936-2043-4b41-b099-f3772d3e9ade", "name": "demo0"}'
```

### Código 11: Pedido curl à API de *cloud* para criar um ambiente em Python

O número de vezes que o teste foi repetido deve-se às limitações físicas das máquinas (isto é, os baixos recursos) e o número de ambientes inferido aquando da provisão da plataforma é 6 a 8 ambientes de desenvolvimento. Cada ambiente foi criado isoladamente e com intervalos variáveis entre cada provisão, porque o tempo dos intervalos aumentava à medida que a duração da provisão também aumentava.

Tendo em conta, estas mesmas limitações a meio desta experiência, os ambientes de desenvolvimento foram apagados, porque a máquina “Borealis” (ver Figura 4.8) estava a consumir o máximo dos recursos da máquina. Foram retomados os testes até ao mesmo cenário, o uso total dos recursos da máquina “Borealis”, isto permitiu entender alguns dos valores que serão apresentados mais à frente no capítulo.

Por fim, os pedidos à API do serviço de *cloud* tiveram na ordem apresenta na Tabela 5.1

---

<sup>2</sup><https://curl.haxx.se/>

Tabela 5.1: *Timeline* dos pedidos ao serviço de *cloud*

<b>Data</b>	<b>Hora</b>	<b>Nome do Ambiente</b>
2017-05-07	12:41:16.589	demoA0
2017-05-07	12:43:07.377	demoA1
2017-05-07	12:45:10.309	demoA2
2017-05-07	12:48:12.502	demoA3
2017-05-07	12:53:09.917	demoA4
2017-05-07	12:59:11.941	demoA5
2017-05-07	13:14:11.174	demoA6
2017-05-07	13:25:02.735	demoA7
2017-05-07	13:32:43.804	demoB0
2017-05-07	13:34:41.223	demoB1
2017-05-07	13:36:42.870	demoB2
2017-05-07	13:39:45.458	demoB3
2017-05-07	13:44:52.936	demoB4
2017-05-07	13:51:22.291	demoB5
2017-05-07	13:58:51.588	demoB6
2017-05-07	14:02:07.112	demoB7

Mais especificamente, a primeira parte da experiência consistiu em disponibilizar os ambientes desde o “demoA0” até ao “demoA7”, deixou-se os ambientes operacionais durante alguns minutos e depois parou-se os *containers*. Na segunda parte, foram realizados mais um conjunto de pedidos para criar os ambientes “demoB0” até ao “demoB7”. Por fim, foram deixados os ambientes *online* por alguns minutos e depois parou-se os mesmos para finalizar a experiência.

## 5.2 DURAÇÃO

Do ponto de vista de experiência de utilização por parte de um utilizador foi recolhido o tempo que cada ambiente de desenvolvimento demora até estar pronto. Esta análise tem interesse para identificar o tempo que um aluno terá que esperar até ter o seu ambiente disponível para trabalhar. Na Figura 5.1 consegue-se ver a duração de cada ambiente criado nesta experiência.

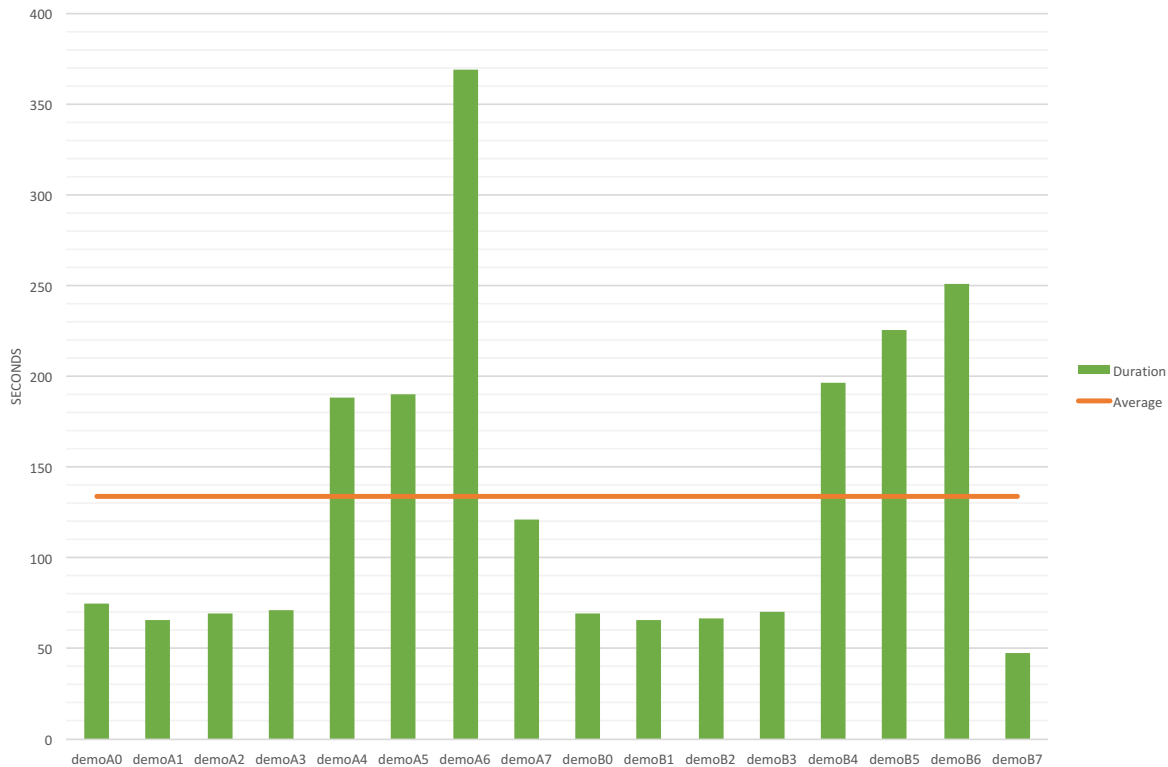


Figura 5.1: Duração da provisão dos ambientes de desenvolvimento

A provisão mais rápida, resultou de um ambiente que supostamente deveria demorar mais tempo, o “demoB7”, resultou num falso positivo, devido a estar a ser provisionado numa altura de stress, tudo indica que a carência de recursos afetou a provisão do ambiente, porque um passo intermédio (instalação de uma dependência) falhou de forma pouco destrutiva e que permitiu o processo criar um Ambiente que iria registar um “sucesso” mas que depois com a sua utilização iria falhar.

A segunda provisão mais rápida foi o ambiente “demoB1” demorou cerca de 1 minuto e 6 segundos, também é perceptível, na Figura 5.1, que à medida que o número de ambientes aumenta, numa infraestrutura limitada, a duração da provisão também aumenta. Como é possível verificar, no mesmo gráfico, o ambiente mais lento a ser disponibilizado demorou 6 minutos e 9 segundos com o nome “demoA6”.

Como já foi referido anteriormente e facilmente visível no gráfico, a descida dos valores na provisão do “demoB0” ocorre por causa da limpeza e preparação da segunda parte, e quando os testes foram replicados, apesar de algumas diferenças, o padrão do consumo é relativamente semelhante.

Com base na amostra recolhida, o tempo médio de uma provisão é de 134 segundos, aproximadamente 2 minutos e 20 segundos, com um desvio padrão de 93 segundos, aproximadamente 1 minuto e 32 segundos. Estes valores são um pouco altos, principalmente o desvio padrão, tendo em conta uma amostra pequena é normal que existam variações maiores quando comparado com amostras maiores. Mas uma variação de 93 segundos só infere como a plataforma iria responder em sinal de *stress*, que nas condições desta experiência é fácil de chegar a esse estado após provisionar 3~4 ambientes. Estes valores só esclarecem que, numa infraestrutura limitada ou sobre-lotada, que a provisão de novos ambientes terá efetivamente um impacto na duração da provisão.



Quanto ao tempo da provisão em condições de menos constrangimento para a infraestrutura, as operações da instalação das dependências de cada ambiente teriam de ser otimizadas ou reduzidas, se for possível.

### 5.3 PROCESSAMENTO

Do ponto de vista de impacto no processamento da infraestrutura da plataforma recolheu-se os consumos totais do Central Processing Unit (CPU) da máquina “Aurora” (ver Figura 5.2) e “Borealis” (ver Figura 5.3). Na inicio da experiência optou-se por monitorizar a máquina “Aurora” com todos os serviços da plataforma desligados, durante aproximadamente 5 minutos, e depois ligou-se cada serviço com intervalos de 1 minuto. Como é possível ver no gráfico por volta das 12:35:36 os serviços estão ligados, é possível verificar nos valores que subiram ligeiramente na escala.

Quanto aos valores da máquina “Aurora” em relação aos pedidos do teste é possível identificar que as oscilações passam de um estado normal de aproximadamente de 10% para os 100% de forma irregular e sem estar relacionado com os pedidos de provisionamento de um ambiente.

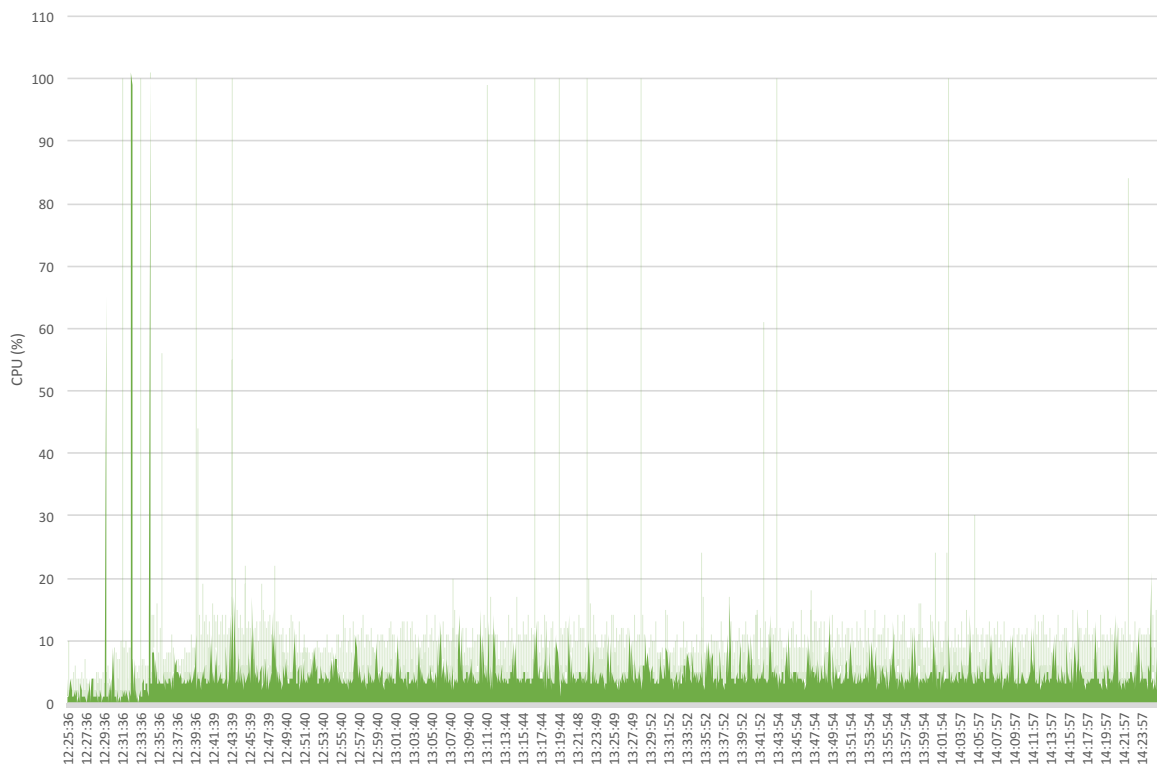


Figura 5.2: Recolha o consumo do CPU na máquina “Aurora” durante o teste

Sobre os valores da máquina “Borealis” já são mais explícitos, quanto à solução esta máquina só contém o *engine* do Docker, o processamento normalmente está abaixo dos 5% de utilização e a quando um pedido o processamento sobe para os 100%, quando terminada a provisão o CPU estabiliza. De notar que o período de descanso entre cada provisão é curto daí os valores oscilarem muito entre cada provisão.

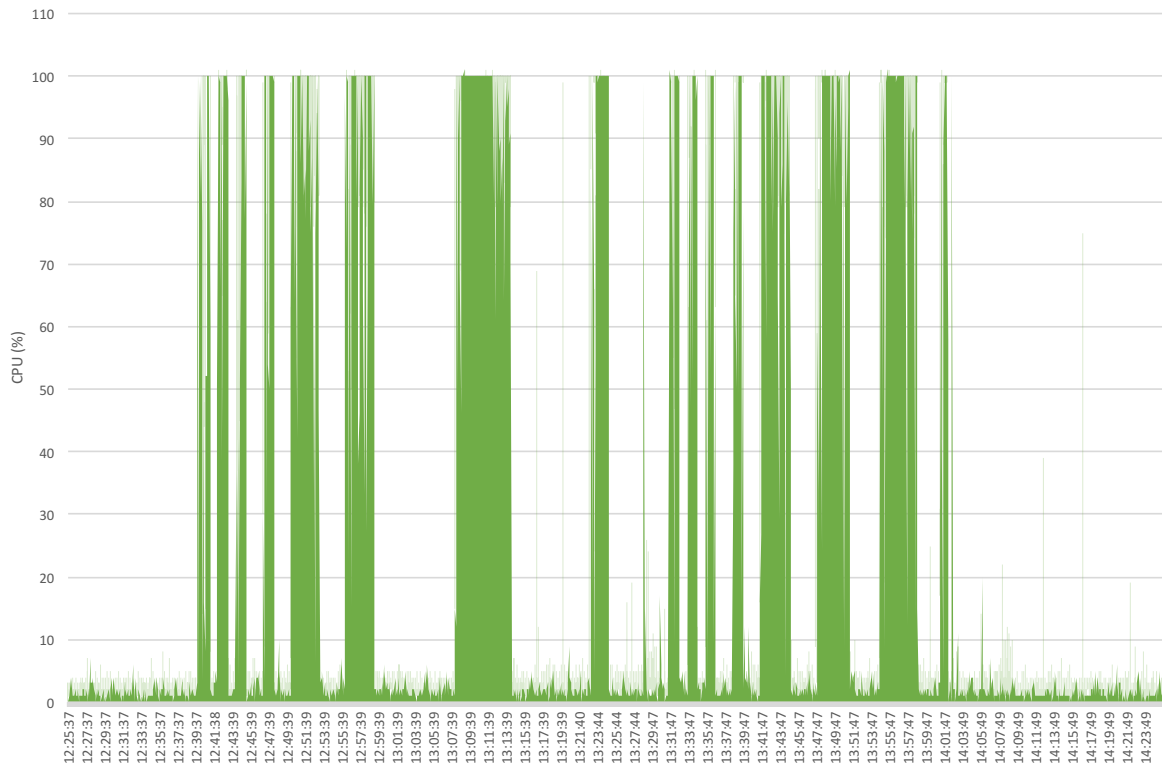


Figura 5.3: Recolha o consumo do CPU na máquina “Borealis” durante o teste

## 5.4 MEMÓRIA

Da mesma forma que se analisou na Seção 5.3, quanto à máquina “Aurora”, o consumo da Random-Access Memory (RAM) relativo a quando se iniciam os serviços que acaba por volta das 12:35, depois disso, as variações são pouco significativas porque nunca descem dos 450MB de utilização (ver Figura 5.4). Significa que os 4 componentes da plataforma estão a usar toda a memória que lhes é disponibilizada.

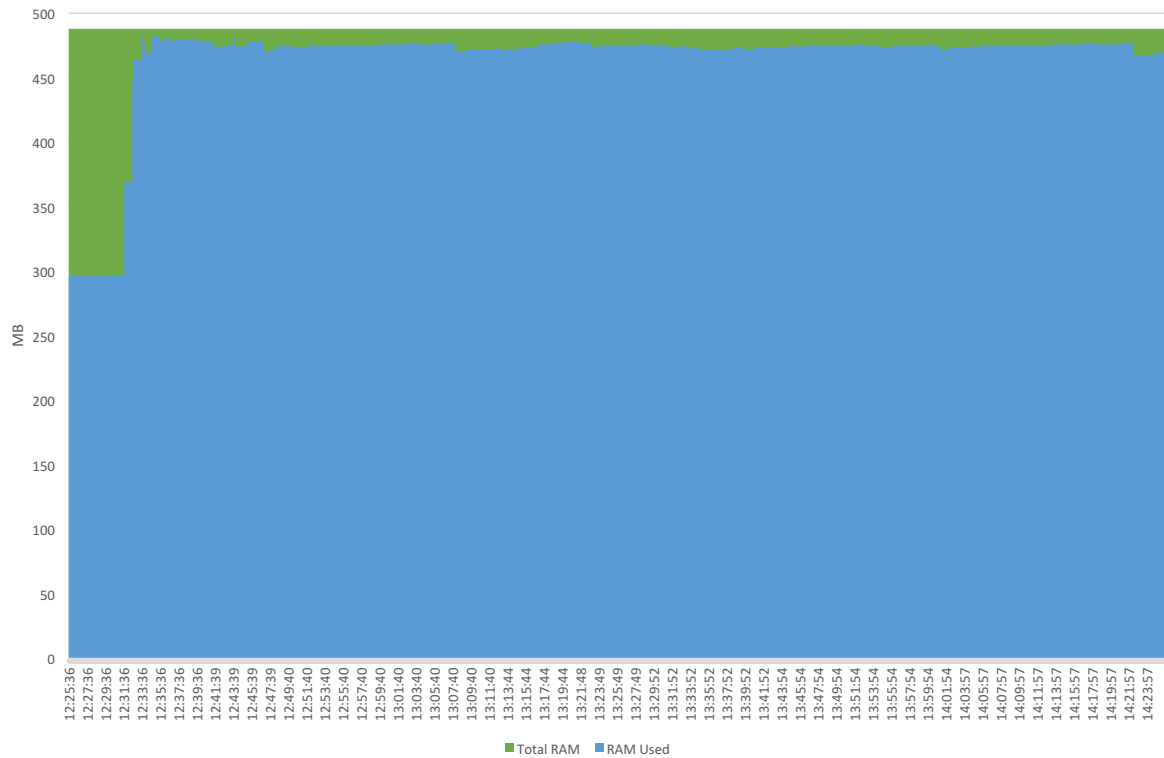


Figura 5.4: Recolha o consumo da memória RAM na máquina “Aurora” durante o teste

Novamente, o caso muda quanto à máquina “Borealis”, porque como aconteceu na Seção 5.3, os consumos oscilam consoante os intervalos entre cada provisionamento do ambiente (ver Figura 5.5).

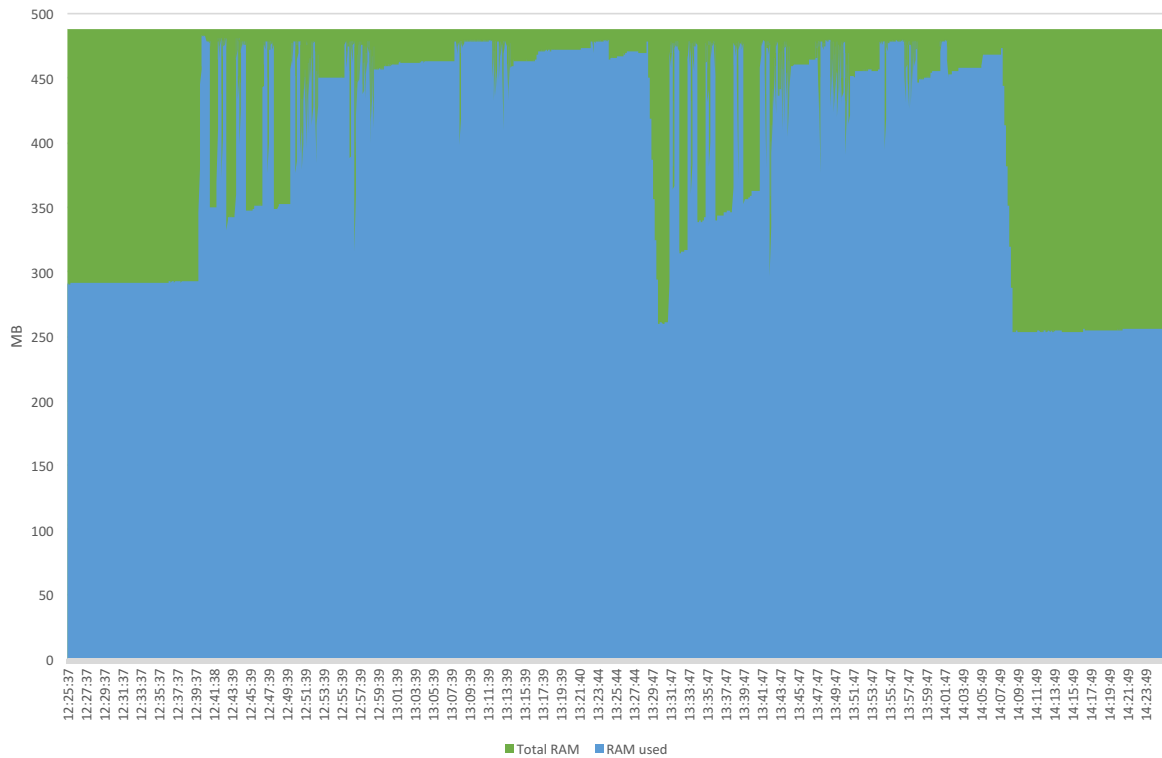


Figura 5.5: Recolha o consumo da memória RAM na máquina “Borealis” durante o teste

A provisão de cada ambiente é identificada pelas oscilações, a primeiro pedido faz com que a memória oscile aproximadamente dos 300MB até aos 480MB, de notar que nas provisões seguintes as variações do uso da memória são maiores, mas à medida que o número de ambientes sobe a oscilação diminui entre o estado de descanso e o estado de provisão. Quanto à descida por volta das 13:30 representa a pausa anteriormente mencionada entre as duas partes da experiência, e a última descida deve-se à paragem dos vários ambientes.

## 5.5 ARMAZENAMENTO

Outro t3pico interessante a analisar 3e o consumo do armazenamento da m3quina com a plataforma de *cloud*, para compreender como a cria33o de v3rios ambientes influencia o espa3o em disco. De notar que os ambientes est3o vazios no momento da sua cria33o, e que cada projeto com o seu conjunto de ficheiros ir3 afetar de forma diferente o armazenamento.

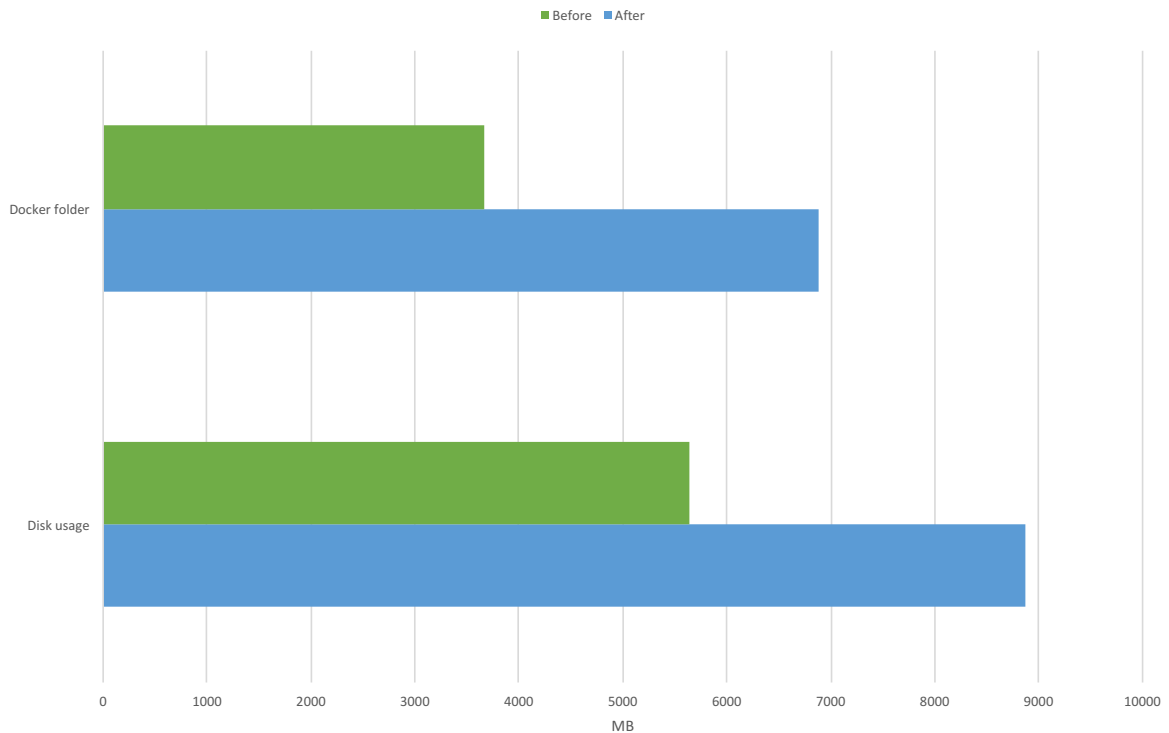


Figura 5.6: Representa33o do armazenamento da m3quina “Borealis”

Usando o comando `docker` 3e poss3vel verificar o tamanho da imagem relativamente ao ambiente em Python, e utiliza do 3,45GB do armazenamento, com uma tecnologia como Docker a cria33o dos v3rios ambientes n3o resulta numa multiplica33o direta dos 3,45GB pelo n3mero de ambientes. Podemos constatar no gr3fico da Figura 5.6, que a cria33o de 8 ambientes ativos em simult3neo afetam no disco em 3230MB (excluindo o valor que imagem ocupa, porque esta foi criada antes da experi3ncia iniciar), e como 3e poss3vel verificar esse espa3o todo deve-se a um diret3rio do Docker, o `/var/lib/docker` que 3e respons3vel por guardar a informa33o por *container*.

## 5.6 MANUTENÇÃO

Após cada parte da experiência optou-se por deixar os vários ambientes a correr em simultâneo para identificar o seu consumo em *idle*. E com o comando `top` foram obtidas duas listas de processos da máquina “Borealis”, a primeira está na Figura 5.7 e a segunda na Figura 5.8.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
9609	atnogus+	20	0	908512	26468	5804	S	0.0	5.3	0:00.56	node
4730	atnogus+	20	0	908512	21376	712	S	0.0	4.3	0:00.71	node
3463	atnogus+	20	0	908512	21308	644	S	0.0	4.3	0:00.76	node
6648	atnogus+	20	0	908512	21296	632	S	0.0	4.3	0:00.74	node
7927	atnogus+	20	0	908512	21036	0	S	0.0	4.2	0:01.30	node
5433	atnogus+	20	0	908512	21024	360	S	0.0	4.2	0:00.70	node
4093	atnogus+	20	0	908512	20660	0	S	0.0	4.1	0:00.69	node

Figura 5.7: Processos ativos da máquina “Borealis” do primeiro conjunto

Os ambientes podem ser identificados nas imagens como os processos com nome `node` que estão a ser executados dentro dos *containers* e que disponibilizam a página do ambiente.

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
16456	atnogus+	20	0	908512	25124	4312	S	0.0	5.0	0:00.56	node
17705	atnogus+	20	0	908512	24944	4336	S	0.0	5.0	0:00.48	node
13933	atnogus+	20	0	908512	21648	984	S	0.0	4.3	0:00.68	node
12597	atnogus+	20	0	908512	21456	792	S	0.0	4.3	0:00.69	node
1457	root	20	0	607416	21368	6196	S	1.0	4.3	0:09.76	docker
15163	atnogus+	20	0	908512	20988	0	S	0.0	4.2	0:00.91	node
13241	atnogus+	20	0	908512	20668	0	S	0.0	4.1	0:00.68	node
11969	atnogus+	20	0	908512	20664	0	S	0.0	4.1	0:00.71	node

Figura 5.8: Processos ativos da máquina “Borealis” do segundo conjunto

Como é possível nas imagens, em ambas as partes os ambientes que estão em *idle*, consomem 0% do CPU enquanto o uso da memória varia entre os 4% e os 5%.

## CONCLUSÃO

---

*Estamos perante uma época em que as tecnologias de web e cloud são o palco para muitas inovações em vários serviços de E-Learning. E usam nas de forma diferenciada para disponibilizar serviços e conteúdos para gerar envolvimento do possível aluno. Por exemplo, com a plataforma edX, hoje em dia é possível acompanhar em direto uma aula de uma instituição de outro continente para adquirir um conhecimento específico para o qual uma instituição do meu país não tenha essa formação. E no final desse curso, se assim for desejado obter a respetiva certificação. Depois existem outros casos, por exemplo a Udemy, que criaram uma plataforma democrática em que cada utilizador pode ser o produtor e o consumidor de diferentes cursos em diferentes áreas ao ritmo que desejar. E um simples sistema de classificação e comentários faz sobressair os melhores conteúdos.*

O principal objetivo deste trabalho era projetar uma nova solução de *E-Learning* para fornecer novos métodos e ferramentas para cursos de programação. O maior desafio era identificar como disponibilizar as inovações em conjunto num único serviço.

A solução concebida fornece um novo formato para expor matéria teórica-prática em que a inovação é a interatividade entre o aluno e a matéria a ser lecionada, esta funcionalidade é limitada porque só suporta uma única linguagem de programação (para mais informação ver Seção 6.2), a outra novidade é a disponibilização de ambientes de desenvolvimento partilhados em tempo-real, um conceito que visa a cooperação entre alunos e até com docentes no decorrer de projetos ou trabalhos de grupo. A sua implementação suporta diferentes linguagens e ferramentas de desenvolvimento com o auxílio de tecnologias de virtualização para fornecer isolamento ao nível do utilizador e dos projetos, mas o fato de estar parcialmente implementada faz com que seja considerado um protótipo. A funcionalidade que tem de ser completada em futuras iterações é a partilha do ambiente, que atualmente existe a partir da troca do URL com os colegas, significando que a sua visibilidade depende de com quem partilham o mesmo *link*, mas o ideal era através de um sistema de autorização com base num registo de utilizadores, que existe, mas atualmente só serve para autenticação e acesso ao portal do serviço.

Por fim, o protótipo ficou implementado e foi possível comprovar e experimentar os conceitos propostos para futuras plataformas de *E-Learning*, e assim fornecer algum suporte aos docentes e aos alunos em cursos de Informática.

## 6.1 CONTRIBUIÇÕES

Durante o desenvolvimento deste trabalho houve alguns momentos em que bibliotecas utilizadas estavam de alguma forma incompletas para as necessidades da implementação ou então tinham *bugs* nas versões disponíveis, foi necessário intervir para concluir a fase do desenvolvimento. A primeira contribuição[84] foi no âmbito da API de abstração ao serviço de identidade/autenticação, como já foi referido durante a fase do desenvolvimento foi eleita como dependência para este componente a utilização de uma *framework* para Nodejs denominada Passportjs, que possui vários módulos para muitos sistemas de autenticação, mais especificamente possui o módulo para a versão OAuth1.0 e outro módulo para a versão OAuth2.0, mas no cenário de testes era necessário usar a versão OAuth1.0a, e por este motivo foi desenvolvida um novo módulo (na terminologia do projeto é denominada por *Strategy*) que suporte a versão 1.0a do protocolo OAuth. A última contribuição[85] foi no âmbito da API da abstração de *cloud*, a biblioteca para permitir que este componente fosse agnóstico da solução *cloud* usava uma biblioteca denominada Apache Libcloud, como foi anteriormente mencionada, e que o *driver* para a tecnologia Docker possuía código com *bugs* que impedia de fazer a leitura correta da mensagem de retorno quando se efetuava pedidos à API HTTP do Docker, foi necessário corrigir algum código deste *driver*, o resultado final permitia listar imagens e criar *containers* sem problemas como era pretendido inicialmente.

## 6.2 TRABALHO FUTURO

Este trabalho resulta simplesmente do esforço que já se manifestou nos últimos anos em alguns departamentos da UA. Desde plataformas sociais[86] a plataformas mais técnicas[5] e esta dissertação é mais um passo nessa evolução, com um foco mais prático. Dito isto, ainda existe espaço para melhorarias, neste protótipo como em futuros métodos e ferramentas.

Quanto às melhorias deste trabalho, o desenvolvimento do protótipo têm de ser finalizado tendo como objetivo disponibilizar a plataforma num cenário de produção, isso acarreta preocupações de escalabilidade, sendo uma solução com um comportamento favorável ao consumo grandes quantidades de recursos quando sujeito a milhares de utilizadores. Por exemplo, os componentes de “Camada de Abstração da Cloud” e de “Camada de Abstração de Autenticação” poderão ser complementados para suportar outros serviços alternativos de *Cloud* e autenticação/autorização. Mas se o Docker for o serviço de *cloud* de eleição também é necessário tirar proveito de *updates* recentes feitos à plataforma. Fortalecer o isolamento e restringir o acesso dos ambientes (*containers*) ao *host* e também limitar o consumo de recursos por parte dos ambientes. Do ponto de vista de serviço, o melhor é ter diferentes *flavours* porque existem projetos com mais exigências que outros.

Depois do ponto de vista de infraestrutura, como mencionado anteriormente, uma utilização em grande escala resulta num aumento significativo da utilização de armazenamento, a utilização de tecnologias de NFS ou Distributed File System (DFS), por exemplo GlusterFS<sup>1</sup>, são possíveis soluções para uma iteração futura.

---

<sup>1</sup><https://www.gluster.org>



Recentemente tem-se discutido no âmbito da educação a utilização de conceitos de *Gamification*, por exemplo: *Badges*, *Achievements* ou *Skill tree*. O conceito *Gamification* é a implementação de estratégias e princípios que foram desenvolvidos para jogos e aplicá-los noutras áreas[87], neste caso no ensino. Existem estudos[88] indicam que *Gamification* ajuda no envolvimento das pessoas nas diferentes atividades, por isso, era interessante numa iteração futura desta plataforma implementar estes conceitos.

De uma breve análise descobriu-se que a Mozilla<sup>2</sup> já tem um projeto relacionado com *Badges* e *Achievements*[89], mas a criação de uma *Skill Tree* pode ser uma inovação porque é uma forma de criar um portfólio (muitas plataformas abordam o assunto de *E-Portfolio* como sendo uma mais valia[59]) que representa as capacidades do aluno, se este portfólio, for público facilita a interação na comunidade estudantil (procurar de pessoas com características específicas para novos projetos) ou no mercado de trabalho na medida em que as empresas e recrutadoras podem consultar essa informação.

---

<sup>2</sup><https://www.mozilla.org>



## REFERÊNCIAS

---

- [1] T. McCue, *Online learning industry poised for \$107 billion in 2015*, [Online; accessed 22-July-2017], 2014. endereço: <https://www.forbes.com/sites/tjmccue/2014/08/27/online-learning-industry-poised-for-107-billion-in-2015>.
- [2] C. Pappas, *The top elearning statistics and facts for 2015 you need to know*, [Online; accessed 22-July-2017], 2015. endereço: <https://elearningindustry.com/elearning-statistics-and-facts-for-2015>.
- [3] —, *The top LMS statistics and facts for 2015 you need to know*, [Online; accessed 22-July-2017], 2015. endereço: <https://elearningindustry.com/top-lms-statistics-and-facts-for-2015>.
- [4] G. Rössling, M. McNally, P. Crescenzi, A. Radenski, P. Ihantola e M. G. Sánchez-Torrubia, «Adapting Moodle to better support CS education», em *Proceedings of the 2010 ITiCSE Working Group Reports*, sér. ITiCSE-WGR '10, Ankara, Turkey: ACM, 2010, pp. 15–27, ISBN: 978-1-4503-0677-5. DOI: 10.1145/1971681.1971684. endereço: <http://doi.acm.org/10.1145/1971681.1971684>.
- [5] P. M. O. Estima, *Plataforma de apoio à aprendizagem de linguagens de programação*, [Online; accessed 22-September-2016], 2013. endereço: <http://hdl.handle.net/10773/11778>.
- [6] E. Hamburger, *Slack is killing email*, [Online; accessed 1-June-2017], 2014. endereço: <https://www.theverge.com/2014/8/12/5991005/slack-is-killing-email-yes-really>.
- [7] FreeCodeCamp, *So yeah we tried slack... and we deeply regretted it*, [Online; accessed 1-June-2017], 2015. endereço: <https://medium.freecodecamp.com/so-yeah-we-tried-slack-and-we-deeply-regretted-it-391bcc714c81>.
- [8] M. Weinberger, *Slack's main rival, Atlassian HipChat, is upping the ante with embedded apps*, [Online; accessed 1-June-2017], 2015. endereço: <http://www.businessinsider.com/hipchat-connect-gets-integrated-apps-2015-11>.
- [9] D. R. Woodley, *PLATO: The emergence of online community*, [Online; accessed 11-October-2016], 1995. endereço: <http://thinkofit.com/plato/dwplato.htm>.
- [10] P. Suppes, M. Jerman e G. Groen, «Arithmetic drills and review on a computer-based teletype», *The Arithmetic Teacher*, pp. 303–309, 1966.
- [11] C. Goldberg, *Auditing classes at M.I.T., on the web and free*, [Online; accessed 11-October-2016], 2001. endereço: [www.nytimes.com/2001/04/04/us/auditing-classes-at-mit-on-the-web-and-free.html](http://www.nytimes.com/2001/04/04/us/auditing-classes-at-mit-on-the-web-and-free.html).
- [12] MIT, *15 years of MIT OpenCourseware*, [Online; accessed 11-October-2016], 2016. endereço: <https://ocw.mit.edu/about/15-years>.

- [13] M. Dougiamas e P. Taylor, «Moodle: Using learning communities to create an open source course management system», em *Proc. World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA) 2003*.
- [14] W. R. Watson e S. L. Watson, «What are learning management systems, what are they not, and what should they become», *TechTrends*, vol. 51, nº 2, p. 29, 2007.
- [15] J. Kay, P. Reimann, E. Diebold e B. Kummerfeld, «MOOCs: So many learners, so much potential», *Technology*, vol. 52, nº 1, pp. 49–67, 2013.
- [16] L. Pappano, *The year of the MOOC*, [Online; accessed 11-October-2016], 2012. endereço: [www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html](http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html).
- [17] A. Auinger, D. Nedbal, A. Holzinger, N. Scerbakov e M. Ebner, «MashUps for e-learning 2.0 simple Personal Learning Environments (PLE) for frequent computer users», *Advances in Industrial Engineering, Information and Water Resources*, pp. 649–661, 2013.
- [18] E. Comission, *OpenEdu: A study on strategies for opening up education*, [Online; accessed 22-September-2016], 2015. endereço: <http://www.openeducationeuropa.eu/en/project/openedu>.
- [19] Inamorato dos Santos, A.; Punie, Y. and Castaño-Muñoz, J., «Opening up education: A support framework for higher education institutions», Directorate Growth & Innovation e JRC-Seville, Joint Research Centre, rel. téc., 2016, Publications Office of the European Union, <http://publications.jrc.ec.europa.eu/repository/handle/JRC101436>.
- [20] Souto-Otero, M. and Inamorato dos Santos, A. and Shields, R. and Lazetic, P. and Castaño Muñoz, J. and Devaux, A. and Oberheidt, S. and Punie, Y., «OpenCases: Case studies on openness in education», Directorate Growth & Innovation e JRC-Seville, Joint Research Centre, rel. téc., 2016, Publications Office of the European Union, <http://publications.jrc.ec.europa.eu/repository/handle/JRC101533>.
- [21] Whitthaus, G. and Inamorato dos Santos, A. and Childs, M and Tannhauser, A. and Conole, G. and Nkuyubwatsi, B. and Punie, Y., «Validation of non-formal MOOC-based learning: An analysis of assessment and recognition practices in europe (OpenCred)», Directorate Growth & Innovation e JRC-Seville, Joint Research Centre, rel. téc., 2016, Publications Office of the European Union, <http://publications.jrc.ec.europa.eu/repository/handle/JRC96968>.
- [22] Castaño Muñoz, J. and Punie, Y and Inamorato dos Santos, A. and Mitic, M. and Morais, R., «How are higher education institutions dealing with openness? A survey of practices, beliefs, and strategies in five european countries», Directorate Growth & Innovation e JRC-Seville, Joint Research Centre, rel. téc., 2016, Publications Office of the European Union, <http://publications.jrc.ec.europa.eu/repository/handle/JRC99959>.
- [23] «Reflections on the 2008 AECT definitions of the field», *TechTrends*, vol. 52, nº 1, pp. 24–25, 2008, ISSN: 1559-7075. DOI: 10.1007/s11528-008-0108-2. endereço: <http://dx.doi.org/10.1007/s11528-008-0108-2>.
- [24] B. Beatty e C. Ulasewicz, «Faculty perspectives on moving from Blackboard to the Moodle Learning Management System», *TechTrends*, vol. 50, nº 4, pp. 36–45, 2006, ISSN: 1559-7075. DOI: 10.1007/s11528-006-0036-y. endereço: <http://dx.doi.org/10.1007/s11528-006-0036-y>.
- [25] D. J. Malan, «Reinventing CS50», em *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, sér. SIGCSE '10, Milwaukee, Wisconsin, USA: ACM, 2010, pp. 152–156, ISBN: 978-1-4503-0006-3. DOI: 10.1145/1734263.1734316. endereço: <http://doi.acm.org/10.1145/1734263.1734316>.

- [26] E. I. P. S.A., *Universidade de Aveiro em ilha virtual*, [Online; accessed 8-September-2016], 2016. endereço: [http://expresso.sapo.pt/dossies/dossiest\\_actualidade/dos\\_second\\_life/universidade-de-aveiro-em-ilha-virtual=f95407](http://expresso.sapo.pt/dossies/dossiest_actualidade/dos_second_life/universidade-de-aveiro-em-ilha-virtual=f95407).
- [27] CIDTFF, *Apresentação da ilha da Universidade de Aveiro em SecondLife® (SL) – a second.ua*, [Online; accessed 6-June-2017], 2016. endereço: <http://blogs.ua.pt/cidtff/index.php/2016/10/28/apresentacao-da-ilha-da-universidade-de-aveiro-em-secondlife-sl-a-second-ua/>.
- [28] M. Pinto, F. Souza, F. Nogueira, A. Balula, L. Pedro, L. Pombo, F. Ramos, A. Moreira e D. Coelho, «Tecnologias da comunicação no ensino superior: Revisão da literatura internacional», *Revista entreideias: Educação, cultura e sociedade*, vol. 2, n° 1, 2013.
- [29] C. Santos e L. Pedro, «Chmod 777 education», *ELearn*, vol. 2009, n° 4, abr. de 2009, ISSN: 1535-394X. DOI: 10.1145/1595384.1594178. endereço: <http://doi.acm.org/10.1145/1595384.1594178>.
- [30] F. Ritcher, *Mobile subscriptions to outnumber the world's population*, [Online; accessed 25-May-2017]. endereço: <https://www.statista.com/chart/4022/mobile-subscriptions-and-world-population/>.
- [31] A. Robertson, *Bill Gates on The Verge: Can online classrooms help the developing world catch up? | The Verge*, [Online; accessed 7-February-2017], 2015. endereço: <http://www.theverge.com/2015/2/11/8014563/bill-gates-education-future-of-online-courses-third-world>.
- [32] M. Fetaji, S. Loskovska, B. Fetaji e M. Ebibi, «Combining Virtual Learning Environment and Integrated Development Environment to enhance e-learning», em *2007 29th International Conference on Information Technology Interfaces*, IEEE, 2007, pp. 319–324.
- [33] A. Dias, C. Santos, C. Costa, L. B. Gouveia, L. Pedro, P. Peres, P. Simões e S. Torrão, «LMS vs PLE: Fusão ou choque?»,
- [34] G. Attwell, «Personal Learning Environments - the future of elearning?», *Elearning papers*, vol. 2, n° 1, pp. 1–8, 2007.
- [35] M. Van Harmelen, «Personal Learning Environments», em *ICALT*, vol. 6, 2006, pp. 815–816.
- [36] D. Shah, *By the numbers: MOOCS in 2016*, [Online; accessed 12-May-2017], 2016. endereço: <https://www.class-central.com/report/mooc-stats-2016/>.
- [37] —, *By the numbers: MOOCS in 2015*, [Online; accessed 12-May-2017], 2015. endereço: <https://www.class-central.com/report/mooc-stats-2016/>.
- [38] S. Downes, *E-learning 2.0*, [Online; accessed 12-May-2017], 2005. endereço: <http://elearnmag.acm.org/featured.cfm?aid=1104968>.
- [39] M. Ebner, «E-Learning 2.0 = e-Learning 1.0 + Web 2.0?», em *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, abr. de 2007, pp. 1235–1239. DOI: 10.1109/ARES.2007.74.
- [40] edX, *About us | edX*, [Online; accessed 16-October-2016], 2016. endereço: <https://www.edx.org/about-us>.
- [41] S. R. 2013, *Stanford to collaborate with edX to develop a free, open source online learning platform*, [Online; accessed 16-October-2016], 2013. endereço: <http://news.stanford.edu/news/2013/april/edx-collaborate-platform-030313.html>.
- [42] edX, *edX | free online courses from the world's best universities*, [Online; accessed 15-May-2017], 2017. endereço: <https://www.edx.org>.

- [43] Udemy, *Udemy online courses - learn anything, on your schedule*, [Online; accessed 4-May-2017], 2017. endereço: <https://www.udemy.com/>.
- [44] Codecademy, *Learn to code | codecademy*, [Online; accessed 17-May-2017], 2017. endereço: <https://www.codecademy.com>.
- [45] D. J. Malan, «Moving CS50 into the cloud», *J. Comput. Sci. Coll.*, vol. 25, n<sup>o</sup> 6, pp. 111–120, jun. de 2010, ISSN: 1937-4771. endereço: <http://dl.acm.org/citation.cfm?id=1791129.1791153>.
- [46] —, «Implementing a Massive Open Online Course (MOOC)», *J. Comput. Sci. Coll.*, vol. 28, n<sup>o</sup> 6, pp. 136–137, jun. de 2013, ISSN: 1937-4771. endereço: <http://dl.acm.org/citation.cfm?id=2460156.2460182>.
- [47] HarvardX, *CS50: Introduction to computer science | edX*, [Online; accessed 26-September-2016], 2016. endereço: <https://www.edx.org/course/introduction-computer-science-harvardx-cs50x>.
- [48] D. J. Malan, *This was CS50x*, [Online; accessed 25-October-2016], 2016. endereço: <https://medium.com/@cs50/this-was-cs50x-82be0995862b#.cg1674vys>.
- [49] D. Armendariz, D. J. Malan e N. Onken, «A web-based IDE for teaching with any language (abstract only)», em *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, sér. SIGCSE '16, Memphis, Tennessee, USA: ACM, 2016, pp. 711–711, ISBN: 978-1-4503-3685-7. DOI: 10.1145/2839509.2844712. endereço: <http://doi.acm.org/10.1145/2839509.2844712>.
- [50] C. HQ, *Codio - teaching code*, [Online; accessed 4-May-2017], 2017. endereço: <https://codio.com/>.
- [51] Wikipedia, *OpenSocial — Wikipedia, The Free Encyclopedia*, [Online; accessed 24-June-2016], 2016. endereço: <https://en.wikipedia.org/w/index.php?title=OpenSocial&oldid=726771947>.
- [52] G. Inc, *Google launches OpenSocial to spread social applications across the web*, [Online; accessed 3-October-2016], 2007. endereço: [http://googlepress.blogspot.pt/2007/11/google-launches-opensocial-to-spread\\_01.html](http://googlepress.blogspot.pt/2007/11/google-launches-opensocial-to-spread_01.html).
- [53] W3C, *The next open web platform*, [Online; accessed 3-October-2016], 2010. endereço: <https://www.w3.org/2010/Talks/0927-html5-plh/Overview.html>.
- [54] J. J. (W3C), *Application foundations for the Open Web Platform*, [Online; accessed 3-October-2016], 2014. endereço: <https://www.w3.org/blog/2014/10/application-foundations-for-the-open-web-platform/>.
- [55] W3C, *Opensocial foundation moving standards work to W3C social web activity*, [Online; accessed 3-October-2016], 2014. endereço: <https://www.w3.org/2014/12/opensocial.html.en>.
- [56] Wikipedia, *OpenID — Wikipedia, The Free Encyclopedia*, [Online; accessed 26-September-2016], 2016. endereço: <https://en.wikipedia.org/w/index.php?title=OpenID&oldid=741204342>.
- [57] —, *OpenCourseWare — Wikipedia, The Free Encyclopedia*, [Online; accessed 25-August-2016], 2016. endereço: <https://en.wikipedia.org/w/index.php?title=OpenCourseWare&oldid=736154974>.
- [58] S. Lacy, *Nixty launches with ambitions to build something huge in elearning*, [Online; accessed 3-June-2017], 2010. endereço: <https://techcrunch.com/2010/07/13/nixty-launch/>.
- [59] M. Feldstein e G. L. Moriarty, «Web 2.0 LMS opportunities and obstacles: Exploring OpenSocial, OpenID, and OpenCourseWare in NIXTY», *On the Horizon*, vol. 17, n<sup>o</sup> 3, pp. 226–231, 2009.

- [60] Moodle.org: Moodle statistics, [Online; accessed 25-July-2017]. endereço: <https://moodle.net/stats>.
- [61] M. Machado e E. Tao, «Blackboard vs. MOODLE: Comparing user experience of learning management systems», em *2007 37th Annual Frontiers In Education Conference-Global Engineering: Knowledge Without Borders, Opportunities Without Passports*, IEEE, 2007, S4J-7.
- [62] G. Bertalan e M. Rumler, *Moodle plugins directory: Javaunittest*, [Online; accessed 25-October-2016], 2016. endereço: [https://moodle.org/plugins/view.php?plugin=jquery\\_javascript](https://moodle.org/plugins/view.php?plugin=jquery_javascript).
- [63] G. M. M. Bashir e A. S. M. L. Hoque, «An effective learning and teaching model for programming languages», *Journal of Computers in Education*, pp. 1-25, 2016.
- [64] P. Edu, *Python IDE to learn programming quickly & efficiently*, [Online; accessed 25-October-2016], 2016. endereço: <https://www.jetbrains.com/pycharm-edu/quickstart/>.
- [65] SourceLair, *LA(M)P stack | SourceLair guides*, [Online; accessed 25-October-2016], 2016. endereço: <https://www.sourcelair.com/guides/compact/lamp-stack#speeding-up-development-with-split-view>.
- [66] Ajax.org, *Ace - the high performance code editor for the web*, [Online; accessed 4-May-2017], 2017. endereço: <https://ace.c9.io/#production=&nav=production>.
- [67] C. I. Inc, *Cloud9 - your development environment, in the cloud*, [Online; accessed 25-October-2016], 2016. endereço: <https://c9.io/>.
- [68] T. Jewell, *An overview of codenvy in 7 steps - part 1*, [Online; accessed 25-May-2017]. endereço: <https://blog.codenvy.com/an-overview-of-codenvy-in-7-steps-part-1-c42f88802b34>.
- [69] V. Jukić, *Start writing ECMAScript 6 ready code with Codeanywhere*, [Online; accessed 25-May-2017]. endereço: <https://blog.codeanywhere.com/start-writing-ecmascript-6-ready-code-with-codeanywhere/>.
- [70] N. Gupta, *Real terminal, all chrome — no nonsense*, [Online; accessed 25-May-2017]. endereço: <https://www.koding.com/blog/2013/09/realterminal-all-chrome-no-nonsense/>.
- [71] Nitrous, *Nitrous features*, [Online; accessed 25-October-2016], 2016. endereço: <https://www.nitrous.io/features>.
- [72] Codiad, *Codiad*, [Online; accessed 25-October-2016], 2016. endereço: <http://codiad.com/>.
- [73] Gitbook, *Gitbook · documentation made easy*, [Online; accessed 25-May-2017], 2017. endereço: <https://www.gitbook.com/>.
- [74] *Github - gitbookio/plugin-exercises: Interactive exercises in a gitbook*, [Online; accessed 25-May-2017]. endereço: <https://github.com/GitbookIO/plugin-exercises>.
- [75] TechEmpower, *Round 9 results - techempower framework benchmarks*, [Online; accessed 5-June-2017], 2017. endereço: <https://www.techempower.com/benchmarks/#section=data-r9&hw=ph&test=query&l=7h80ov>.
- [76] U. de Aveiro, *OAuth@UA*, [Online; accessed 5-October-2016], 2016. endereço: <http://identity.ua.pt>.
- [77] E. Hammer-Lahav, *Introduction — OAuth*, [Online; accessed 11-May-2017], 2007. endereço: <https://oauth.net/about/introduction/>.
- [78] E. E. Hammer-Lahav, «The OAuth 1.0 Protocol», RFC 5849, abr. de 2010. endereço: <https://tools.ietf.org/html/rfc5849>.

- [79] A. G. Community, *Github - angular/angular.js: HTML enhanced for web apps*, [Online; accessed 20-October-2016], 2016. endereço: <https://github.com/angular/angular.js>.
- [80] Google, *Angularjs: Developer guide: Conceptual overview*, [Online; accessed 20-October-2016], 2016. endereço: <https://docs.angularjs.org/guide/concepts>.
- [81] DreamHost, *Web server performance comparison – dreamhost*, [Online; accessed 20-October-2016], 2016. endereço: <https://help.dreamhost.com/hc/en-us/articles/215945987-Web-server-performance-comparison>.
- [82] *OAuth@UA > Bem-vindo*, [Online; accessed 25-May-2017]. endereço: <https://identity.ua.pt/oauth/>.
- [83] Docker, *What is docker?*, [Online; accessed 20-October-2016], 2016. endereço: <https://www.docker.com/what-docker>.
- [84] Github, *Github - poliveira89/passport-identityua: University of aveiro authentication strategy for passport*, [Online; accessed 1-June-2017]. endereço: <https://github.com/poliveira89/passport-identityua>.
- [85] —, *Github - poliveira89/libcloud at 1.0.0-pre2*, [Online; accessed 1-June-2017]. endereço: <https://github.com/poliveira89/libcloud/tree/1.0.0-pre2>.
- [86] S. Labs, *SAPO Campus*, [Online; accessed 21-May-2017]. endereço: <http://campus.sapo.pt/>.
- [87] J. Atwood, *The gamification*, [Online; accessed 23-October-2016], 2011. endereço: <https://blog.codinghorror.com/the-gamification>.
- [88] R. Raymer, «Gamification: Using game mechanics to enhance elearning», *ELearn*, vol. 2011, nº 9, set. de 2011, ISSN: 1535-394X. DOI: 10.1145/2025356.2031772. endereço: <http://doi.acm.org/10.1145/2025356.2031772>.
- [89] Mozilla, *Open Badges Homepage*, [Online; accessed 3-June-2017]. endereço: <https://openbadges.org/>.



Estes anexos só estão disponíveis para consulta através do CD-ROM.  
Queira por favor dirigir-se ao balcão de atendimento da Biblioteca.

Serviços de Biblioteca, Informação Documental e Museologia  
Universidade de Aveiro