



Teaching Software Development for the Cloud: An Experience Report

Fernando Antonio Mota Trinta
Universidade Federal do Ceará
Fortaleza, Ceará
fernando.trinta@dc.ufc.br

Emanuele Santos
Universidade Federal do Ceará
Fortaleza, Ceará
emanuele@dc.ufc.br

ABSTRACT

Cloud computing is an increasingly popular IT model for resource provisioning due to the ease of purchasing and managing these resources, as well as cost optimization by delegating server and application maintenance to third-party companies. Cloud computing brings benefits for the development of applications and systems, but it also poses new challenges to be faced by those who want to take advantage of cloud capabilities. These challenges include issues such as choosing which type of provisioning model is most appropriate for a particular application, or even if the migration of an application to the cloud itself is feasible or not. Since 2012, the Federal University of Ceará has been offering a course that addresses the challenges of developing software for the cloud. This article presents an experience report after offering this course five times over the last five years. We conducted a survey with more than 50 students who took the course. Their feedback was important to gather their impressions about the relevance of the theme for software engineering. At last, we also discuss challenges in teaching this theme, as well as solutions we have used to overcome them.

CCS CONCEPTS

• **Social and professional topics** → **Software engineering education**; • **Computer systems organization** → *Cloud computing*;

KEYWORDS

Cloud Computing, Software Engineering

ACM Reference format:

Fernando Antonio Mota Trinta and Emanuele Santos. 2017. Teaching Software Development for the Cloud: An Experience Report. In *Proceedings of SBES'17, Fortaleza, CE, Brazil, September 20–22, 2017*, 10 pages. <https://doi.org/10.1145/3131151.3131184>

1 INTRODUÇÃO

Computação em Nuvem (do inglês, *Cloud Computing*) representa um novo modelo de provisionamento de recursos de TI que tem feito bastante sucesso nos últimos anos. Este modelo promove a oferta de uma gama de diferentes recursos de TI (servidores, espaço

de armazenamento, aplicações, dentre outros), provisionados de modo rápido e ágil por meio de técnicas de virtualização, em que seus usuários são tarifados de acordo com seu consumo, conhecido como pay-per-use.

Esta forma de oferta de recursos traz benefícios como melhor aproveitamento de recursos por meio da consolidação de servidores reais e consequente diminuição de custos de operação, maior flexibilidade para gerenciamento de aplicações e recursos, maior facilidade para lidar com grandes variações no uso de recursos, dentre outros. Este cenário materializa a visão de Computação Utilitária [7], preconizada desde a década de 60 por renomados pesquisadores como John McCarthy, que em seu discurso no centenário do MIT, indicava que no futuro, recursos computacionais estariam organizados como bens ou serviços públicos, de modo similar aos de telefonia, energia elétrica ou água. Estes serviços seriam a base para uma nova e importante indústria. Por meio de uma conexão à Internet, usuários tem acesso a uma infinidade de recursos.

De fato, a nuvem permite que engenheiros de software tenham acesso a uma grande quantidade de recursos, gerando um imenso potencial de processamento e armazenamento de informações. Mas lidar com estes recursos de uma forma organizada e sistemática não é trivial. Um estudo realizado sobre o impacto da computação em nuvem sobre a engenharia de software [2] aponta uma série de desafios a serem enfrentados por desenvolvedores de software para a nuvem. Muitos dos desafios apresentados neste trabalho permanecem atuais [1, capítulo 9], e servem de motivação para que desde 2012 seja ofertada no departamento de computação da Universidade Federal do Ceará (UFC) a disciplina intitulada “Desenvolvimento de Software para Nuvem”. Tal disciplina busca permitir que alunos de graduação e pós-graduação da UFC estudem como a computação em nuvem pode beneficiar o desenvolvimento de aplicações escaláveis e flexíveis. A disciplina utiliza uma metodologia teórico-prática de modo que os alunos possam vivenciar na prática alguns dos desafios citados na literatura sobre o desenvolvimento de software para a nuvem, e as principais abordagens utilizadas para contornar estes problemas.

Este artigo apresenta um relato da experiência da realização de cinco edições desta disciplina, ao longo destes cinco anos. São apresentados neste documento, um referencial teórico dos assuntos abordados na disciplina (Seção 2) e da motivação para sua criação (Seção 3). Além disso, a Seção 4 fornece uma visão geral da disciplina, os tópicos abordados, plano de aula e avaliação. A partir de um questionário feito com mais de 50 alunos das diferentes turmas, foram coletadas informações sobre diversos aspectos da disciplina. Esta avaliação (Seção 5) aborda aspectos como importância da ementa, assuntos mais importantes e mais irrelevantes, dentre outros, cujos resultados são apresentados na Seção 6. A Seção 7

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SBES'17, September 20–22, 2017, Fortaleza, CE, Brazil

© 2017 Association for Computing Machinery.

ACM ISBN 978-1-4503-5326-7/17/09...\$15.00

<https://doi.org/10.1145/3131151.3131184>

apresenta um apanhado das principais dificuldades encontradas pelo professor da disciplina, e soluções utilizadas para contornar tais problemas. A Seção 8 apresenta outras experiências de ensino de computação em nuvem no Brasil e no Mundo. Por fim, a Seção 9 conclui o artigo.

2 COMPUTAÇÃO EM NUVEM

As ideias e benefícios associados à computação em nuvem têm despertado o interesse tanto na indústria quanto na academia. Se para alguns trata-se de uma grande revolução, para outros não passa do uso de conceitos e tecnologias já conhecidos há bastante tempo, como por exemplo virtualização. Apesar da falta de consenso inicial, a computação em nuvem criou uma terminologia própria ao longo dos últimos anos, como propostas de definição para seus conceitos e características principais.

Uma das mais completas definições sobre a computação em nuvem é proposta por Vaquero e seus coautores [9], que apresentam a nuvem como um grande repositório de recursos virtualizados facilmente utilizáveis e acessíveis (como hardware, plataformas de desenvolvimento e/ou serviços). Esses recursos podem ser dinamicamente reconfigurados para ajustar a carga (escala) variável do sistema, permitindo também um uso ótimo dos recursos. Esse repositório de recursos é geralmente explorado por um modelo de pagamento por uso, no qual garantias são oferecidas por um provedor de infraestrutura por meio de acordos de níveis de serviço (do inglês, *Service Level Agreements*). Porém, a definição mais referenciada na literatura é aquela proposta pelo *National Institute of Standards and Technology* (NIST) [4], que diz que a computação em nuvem é um modelo computacional que permite o acesso, sob demanda e através da rede (comumente Internet), a um “pool” de recursos computacionais (redes, servidores virtuais ou físicos, armazenamento, aplicações e serviços) que podem ser rapidamente provisionados. O relatório do NIST indica ainda que a computação em nuvem possui cinco características essenciais, três modelos de entrega de serviços e quatro modelos de implantação. A Figura 1 ilustra tais conceitos.

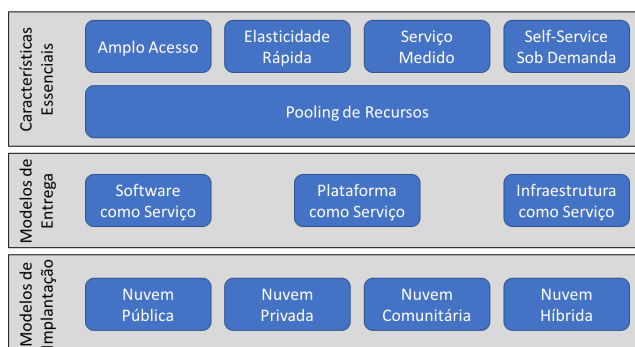


Figura 1: Visão NIST para computação em nuvem

Fonte: Adaptado de [4]

As características essenciais incluem:

- (1) **Auto-serviço sob demanda:** indica que usuários de nuvem podem se servir dos recursos e torná-los ativos sem a necessidade de longas negociações e intervenção humana nesta negociação;
- (2) **Acesso amplo a recursos:** permite que aplicações sejam acessadas por meio de padrões conhecidos (como APIs REST) por diversos tipos de clientes (*browsers*, *smartphones*), onde os detalhes de funcionamento dos recursos são escondidos dos usuários;
- (3) **pooling de recursos:** os provedores são agrupados para servir a vários consumidores usando um modelo multi-inquilino, com recursos físicos e virtuais diferentes sendo atribuídos e reatribuídos de forma dinâmica, de acordo com a demanda do consumidor;
- (4) **Elasticidade rápida:** recursos podem ser provisionados de forma rápida, elástica e, de preferência, automaticamente. Recursos são alocados quando acontece o aumento rápido da demanda, e são liberados rapidamente quando esta demanda diminui;
- (5) **Serviço Medido:** a nuvem controla e otimiza recursos de forma automática, por meio de recursos de medição, de forma adequada ao tipo de serviço ofertado (por exemplo, quantidade de espaço alocado para armazenamento de dados).

Na nuvem, a principal abstração é o “serviço”, o qual permite encapsular os mais diferentes tipos de recursos. Na classificação do NIST, serviços são classificados em três modelos de entrega. Estes modelos podem ser vistos como camadas hierárquicas para oferta de serviços em um típico ambiente de TI. No nível mais baixo temos a Infraestrutura como Serviço (IaaS - do inglês *Infrastructure as a Service*), onde recursos como máquinas, armazenamento e rede são ofertados para formar a infraestrutura de nível mais baixo para funcionamento dos sistemas. São exemplos de provedores de nuvem IaaS a Amazon Web Services (AWS)¹ e a Google Cloud Platform². Os usuários do IaaS são arquitetos de sistemas que lidam com as configurações das máquinas e da rede. O nível intermediário é o da Plataforma como Serviço (PaaS - do inglês *Platform as a Service*), cujo papel é fornecer um arcabouço de desenvolvimento voltado especificamente para aplicações na nuvem. Os principais interessados neste nível são os engenheiros de software que querem construir aplicações para tirar proveito da nuvem. Exemplos de PaaS incluem o Google App Engine e a Microsoft Azure. O nível mais alto é de Software como Serviço (SaaS - do inglês *Software as a Service*), onde aplicações completas são executadas na infraestrutura da nuvem, e utilizadas por usuários comuns por meio de um *browser* web. Exemplos de aplicações neste modelo incluem o pacote GoogleDocs ou o CRM Salesforce.

Em relação à implantação, nuvens públicas são aquelas em que a infraestrutura de nuvem é de propriedade de uma organização que oferece serviços ao público em geral. Já as nuvens privadas são operadas por uma organização particular para suporte aos seus sistemas. Nuvens comunitárias compartilham a infraestrutura da nuvem entre várias organizações que possuem as mesmas preocupações. Por fim, as nuvens híbridas são aquelas compostas por duas

¹<http://aws.amazon.com>

²<http://gcp.google.com>

ou mais nuvens de características distintas (privada, comunitária ou pública).

3 DESENVOLVIMENTO DE SOFTWARE PARA NUVEM

A computação em nuvem envolve várias áreas da computação como redes de computadores, sistemas distribuídos e engenharia de software. Em relação ao último tópico, o desenvolvimento de software utilizando recursos da nuvem impõe novos requisitos e preocupações para engenheiros de software. Porém, o impacto da nuvem para a engenharia de software depende também da forma como recursos da nuvem são utilizados no desenvolvimento das aplicações. Por exemplo, um cenário onde recursos de IaaS são usados para virtualizar servidores e hospedar uma típica aplicação Web (servidores de aplicação, servidores de bancos de dados, etc) apresenta poucas diferenças do modelo tradicional utilizado para implantação dessas aplicações. Porém em outro cenário, o desenvolvimento de aplicações do modelo “Software como Serviço” faz com que requisitos como elasticidade ou uso da abordagem multi-inquilino (do inglês, *multitenancy*) sejam fatores indispensáveis no projeto e implementação destas aplicações. O primeiro caso é visto em geral como a ideia de “desenvolver com a nuvem”, enquanto o segundo seria o “desenvolver para a nuvem”.

Alcançar os benefícios proporcionados pela nuvem impõe certas tomadas de decisões, e a responsabilidade de se arcar com as consequências das escolhas tomadas. A literatura aponta uma série de desafios para desenvolvedores de aplicações em nuvem [2] [1, capítulo 9]. Um dos desafios mais citados é o *Vendor Lockin*, causado pela dificuldade na migração de aplicações entre diferentes provedores de nuvem, especialmente no modelo de PaaS. Isso acontece porque cada provedor possui componentes, modelo de programação e APIs específicas, fazendo com que portar uma aplicação para outro provedor de nuvem se torne uma tarefa complexa ou até impossível. Soluções para o problema vão desde o uso de *Model Driven Engineering* ou até mesmo plataformas de middleware para abstrair diferenças entre os provedores.

Um cenário parecido é o da “cloud migration” [5], que trata da migração de aplicações existentes e desenvolvidas no modelo “tradicional”, para o modelo da nuvem, com suporte a elasticidade, alta disponibilidade, dentre outros. Muitas vezes, esta migração requer uma completa reengenharia da aplicação. Essa situação pode inviabilizar a decisão de se migrar para a nuvem devido aos altos custos envolvidos neste processo.

Mais recentemente, novos cenários têm sido propostos para sistemas em nuvem, trazendo também novos desafios. Um exemplo é o suporte à interoperabilidade em ambientes multicloud, no qual aplicações dinâmicas e complexas tiram proveito de componentes heterogêneos espalhados em múltiplos provedores, de modo a ter ganhos em relação a desempenho, disponibilidade ou diminuição de custos [6].

Além dos desafios citados, pode-se citar também a necessidade de estudos em relação a processos de desenvolvimento adequados para a nuvem, incluindo custos com testes em um ambiente onde os recursos são tarifados de acordo com o uso [10], a segurança dos dados armazenados em infra-estrutura de terceiros, a adequação de modelos existentes de governança na nuvem e mesmo questões

legais que impedem a execução de determinadas aplicações na nuvem.

4 PROPOSTA DA DISCIPLINA

4.1 Histórico

Baseado na relevância sobre o tema, no seu potencial campo de pesquisa e no interesse pessoal pelo desenvolvimento de software para nuvem, um dos autores deste artigo começou a estudar o tema em 2009. O conhecimento gerado a partir de suas pesquisas o levou a projetar uma disciplina de pós-graduação, com objetivo de preparar seus alunos de mestrado e doutorado para pesquisa na área. Como resultado, a primeira edição da disciplina foi ofertada no segundo semestre de 2012.

Inicialmente, a disciplina foi planejada como uma edição particular da disciplina “Tópicos Avançados em Sistemas de Informação II” do programa de Mestrado e Doutorado em Ciência da Computação (MDCC) da Universidade Federal do Ceará. As disciplinas de “tópicos avançados” são utilizadas pelos docentes do MDCC para tratar assuntos emergentes e de seu interesse, geralmente voltados a uma linha de pesquisa do programa. Em 2015, a reformulação do curso de graduação em Ciência da Computação da UFC permitiu a criação de novas disciplinas. Neste ano foi então criada uma nova disciplina com o nome de “Desenvolvimento de Software para Nuvem”, e incluída nos projetos pedagógicos dos cursos de Ciência da Computação e de Engenharia da Computação. A disciplina passou então a mesclar turmas com alunos de graduação, de mestrado e de doutorado.

4.2 Ementa e Avaliação

Para a graduação, a disciplina é classificada como optativa, com carga-horária de 64 horas (a mesma carga-horária da disciplina da pós-graduação), o que equivale a dois dias de aula por semana durante um semestre. A única disciplina pré-requisito é a de “Técnicas de Programação I”, que trata do paradigma orientado a objetos. Esta restrição se deve principalmente ao fato de que a disciplina tem caráter prático, onde alunos devem exercitar conceitos teóricos aprendidos em trabalhos que exigem habilidades de programação ao longo da disciplina, com gradual aumento de complexidade.

A ementa inicialmente proposta incluía: (i) Conceitos Básicos de Computação em Nuvem, (ii) Modelos de Serviços e de Implantação em Nuvem, (iii) Virtualização, (iv) Desenvolvimento com PaaS, (v) Desenvolvimento SaaS e Multitenancy, e (vi) Desafios para software em nuvem. Porém, ao longo das edições da disciplina, novos tópicos foram adicionados, como elasticidade e desenvolvimento multicloud.

A avaliação é feita de forma diferenciada para alunos de graduação e de pós. Em ambos os casos, os alunos fazem uma ou duas provas teóricas, e têm seus trabalhos práticos avaliados para composição de suas notas. Todos precisam ainda realizar um trabalho final de cunho prático, em um tema relacionado à disciplina. Porém, alunos de pós-graduação precisam também apresentar um seminário sobre algum assunto ou artigo específico cujo tema esteja relacionado à disciplina. Além disso, os trabalhos realizados por alunos de pós são avaliados com maior rigidez.

4.3 Infraestrutura utilizada

Uma das maiores preocupações na concepção da disciplina foi o suporte necessário para que os alunos pudessem praticar os conceitos aprendidos ao longo do curso. Neste sentido, a grande questão inicial foi o acesso às infraestruturas de nuvem, de preferência nos diversos modelos apresentados na Seção 2.

No caso da UFC, o departamento de computação possui uma nuvem computacional privada, adquirida por meio de diversos projetos de pesquisa ao longo dos últimos anos, e que é operada atualmente por meio do gerenciador OpenStack. A princípio, esta seria a solução utilizada pelos alunos para seus trabalhos.

Porém, uma outra solução se mostrou mais atrativa e eficiente para o curso. Trata-se do uso dos serviços da Amazon Web Services (AWS). A AWS é considerada a maior provedora de serviços de nuvem e permite que usuários possam ter acesso gratuito a alguns recursos de sua nuvem pública durante o primeiro ano da criação de suas contas de acesso. A AWS fornece um conjunto de serviços em uma camada gratuita (*free tier*), como forma de atrair desenvolvedores a conhecer os recursos oferecidos. O uso da AWS se mostrou bastante efetivo devido aos níveis de maturidade, qualidade, inovação e facilidade de uso dos seus serviços ofertados. Dentre os serviços utilizados, destacam-se:

- (1) **Elastic Compute Cloud (EC2)**: serviço que permite alugar máquinas virtuais para executar suas próprias aplicações;
- (2) **Relational Database Service (RDS)**: serviço para criação e gerenciamento de instâncias de banco de dados relacionais;
- (3) **DynamoDB**: serviço que permite acesso a um banco de dados NoSQL escalável;
- (4) **Simple Storage Service (S3)**: serviço para armazenamento e recuperação escalável de objetos (arquivos binários, textos, fotos) com uma interface de web service simples;
- (5) **Elastic MapReduce (EMR)**: serviço que permite a execução de tarefas MapReduce na infraestrutura sob a estrutura da EC2.

Além destes serviços, a AWS possui soluções como o *AutoScaling Groups* e o *Load Balancing*, que promovem elasticidade horizontal, a partir da inclusão e retirada de réplicas de servidores de aplicação mediante gatilhos que indicam o aumento ou diminuição na demanda de uma determinada aplicação; ou o *Elastic Beanstalk*, uma espécie de PaaS da AWS. Porém, talvez a maior justificativa para uso da AWS seja a liderança que a mesma exerce no mercado mundial de provedores de nuvem. Pode-se dizer que a AWS é um padrão *de facto*, onde seus concorrentes buscam copiar os serviços ofertados. Isso acaba também sendo um atrativo para os alunos que querem conhecer um assunto popular do mercado de trabalho.

O uso dos serviços da AWS foi facilitado ainda mais depois que, a partir da criação da disciplina, tomou-se conhecimento da iniciativa AWS Educate da Amazon³. Por meio de um cadastro institucional gratuito, a Amazon passou a oferecer créditos a alunos e professores para uso de seus serviços, e que podem ser renovados a cada ano.

Apesar da importância da AWS e suporte à disciplina, outras soluções de nuvem também são utilizadas na disciplina. Um exemplo é o Google App Engine (GAE), ferramenta de PaaS proposta pela Google para desenvolvimento de aplicações na nuvem. De modo

Tabela 1: Plano de Aula

Conteúdo Abordado	Carga Horária
Introdução à Computação em Nuvem	6h
Infraestrutura como Serviço: Amazon Web Services	6h
Plataforma como Serviço: Google App Engine	4h
Soluções Open-Source para Computação em Nuvens Privadas	4h
Desafios da Engenharia de Software para nuvem	4h
Virtualização	2h
Elasticidade	4h
Processamento Intensivo de Dados com MapReduce	4h
Desenvolvimento Multi-inquilino	4h
Tópicos Avançados em Computação em Nuvem	8h
Seminários de alunos de pós-graduação	6h
Trabalhos finais	8h
Total	60h*

* Nota: As quatro horas restantes eram usadas para provas escritas

muito parecido, o GAE também permite que novos usuários da plataforma possam desenvolver aplicações sem cobranças durante um ano, desde que suas aplicações não consumam recursos de modo exacerbado. O uso do GAE é apresentado como um estudo de caso para o tema de Plataforma como Serviço.

4.4 Plano de Aulas

Durante a concepção da disciplina, uma preocupação foi fazer com que os alunos vivenciassem os conteúdos teóricos apresentados em sala de aula. Dessa forma, o plano de aulas da disciplina procurou sempre incluir atividades práticas após a apresentação de conteúdo teórico. Essa disposição é resumida na Tabela 1 e explicada a seguir.

Como em qualquer outra disciplina, as primeiras aulas apresentaram os conceitos básicos relacionados à computação em nuvem. Basicamente são apresentados e discutidos os conceitos introduzidos na Seção 2, além de exemplos de sucesso no uso da computação em nuvem. Em geral, esta parte da disciplina dura 2 a 3 aulas, e objetiva preparar os alunos para os próximos tópicos da disciplina.

Em seguida, os alunos têm o seu primeiro contato com atividades práticas na nuvem. São apresentados os serviços básicos da AWS, a saber: EC2, S3 e o RDS. Nesta demonstração também é apresentado o programa AWS Educate. A partir desta apresentação, os alunos são orientados a se cadastrar no programa para ter acesso aos créditos para uso na AWS.

Na aula seguinte, os alunos recebem sua primeira tarefa: a configuração de um gestor de conteúdo bastante conhecido, o WordPress⁴, que deve ser instalado na AWS EC2. Esta atividade é realizada em laboratório com acompanhamento do professor. Estas

³<http://www.awseducate.com>

⁴<http://wordpress.com>

aulas permitem que o aluno tenha experiência com o modelo de Infraestrutura como Serviço descrito na Seção 2.

A etapa seguinte consiste em apresentar como os recursos da AWS podem ser utilizados de modo programático, por meio de uma API Rest. Esta é a forma mais comum de oferta de recursos por provedores de nuvem de IaaS. Desta maneira, desenvolvedores podem manipular recursos por meio de programas escritos em diferentes linguagens de programação. Isso permite que o primeiro trabalho prático de desenvolvimento seja realizado por grupos de até 3 alunos. O trabalho consiste no desenvolvimento de uma aplicação Web para gerência de contatos (nome, telefone, email) utilizando os serviços da AWS. Os serviços devem ser utilizados da seguinte forma: os dados de cada contato devem ser armazenados na RDS, a foto do contato deve ser salva no S3, um log das operações realizadas deve ser salvo no DynamoDB.

A próxima etapa da disciplina é o estudo de plataformas como serviço (PaaS). Em todas as edições da disciplina, esta etapa utilizou o Google App Engine (GAE) para as práticas. É feita uma revisão sobre o modelo PaaS, seus benefícios e desvantagens, para então se detalhar especificadamente o GAE. É demonstrado o desenvolvimento de aplicação simples, no estilo *HelloWorld*. Como tarefa para os alunos é solicitada o desenvolvimento de uma versão GAE da aplicação de contatos feita para a AWS na etapa anterior. O grande objetivo desta tarefa é mostrar ao aluno que ao usar uma PaaS, benefícios como elasticidade são garantidas automaticamente pela plataforma, o que não foi tratado na versão anterior. Porém, também são discutidas as imposições feitas pelo GAE em relação às componentes disponíveis, linguagens de programação compatíveis, dentre outros fatores. O debate gira em torno do *tradeoff* do uso desta abordagem.

As aulas seguintes apresentam plataformas *open-source* para administração de nuvens privadas. São apresentadas 4 soluções: Eucalyptus, OpenNebula, CloudStack e OpenStack. Para cada solução são apresentadas suas principais características, arquitetura, deficiências e virtudes. É dado um enfoque maior à plataforma OpenStack por ser aquela disponível no departamento, e pela popularidade da solução no mercado global. No caso desta última, é feita uma demonstração utilizando a nuvem privada do departamento, fazendo uma comparação com os serviços e recursos disponíveis na AWS. Esta aula permite demonstrar aos alunos que apesar de existirem serviços semelhantes para virtualização de servidores como na Amazon EC2, ou mesmo alocação de espaço como na Amazon S3, recursos avançados semelhantes aos encontrados na AWS não estão disponíveis no OpenStack. Os alunos são então orientados a desenvolver uma nova versão da aplicação de controle de contatos para plataforma OpenStack.

Após as aulas e trabalhos práticos, as aulas seguintes são uma revisão sobre os conceitos aprendidos e as dificuldades encontradas. Neste momento são apresentados artigos sobre os impactos e desafios para o desenvolvimento de software para/com a nuvem. Neste ponto da disciplina, os alunos já vivenciaram na prática alguns dos desafios apontados nos artigos, como aqueles enfrentados ao se tentar portar a aplicação desenvolvida para AWS para o GAE. São discutidas quais seriam as possíveis soluções para os problemas, e quais foram as ações tomadas por cada grupo de alunos.

O tópico explorado a seguir é a virtualização. São apresentados os conceitos, motivação, principais abordagens de virtualização e tipos

de hipervisores. O conceito é contextualizado com sua aplicação no ambiente de nuvem, e são apresentados exemplos de virtualizadores, como VirtualBox e o XenServer.

O tópico seguinte é um dos mais importantes na disciplina: o suporte a elasticidade em ambientes de nuvem. São apresentados os principais conceitos e as principais estratégias existentes para adicionar elasticidade a uma aplicação na nuvem. Apesar da elasticidade ser garantida no modelo PaaS, as aplicações em nuvem que usam apenas serviços de IaaS não dão suporte automático a esta característica. Este é o caso da primeira versão da aplicação desenvolvida pelos alunos na Amazon AWS. Para lidar com a situação são apresentados os serviços de *AutoScaling* e *Load Balancing*, mencionados previamente. Como nova tarefa, os alunos devem utilizar tais serviços para criar regras para adição e remoção de servidores, de acordo com a demanda de uso.

Na continuação, o próximo assunto é o processamento intensivo de dados na nuvem. Este é um assunto de bastante interesse dos alunos, pois relaciona-se com temas bastante atuais como a ciência de dados (Data Science) e Big Data. O grande foco desta etapa é apresentar o MapReduce, um modelo de programação que permite o processamento de grandes massas de dados, de modo paralelo e distribuído, utilizando um *cluster* de servidores virtuais provisionados pela nuvem. Os alunos são apresentados aos dois principais frameworks MapReduce existentes: o Apache Hadoop e Apache Spark. São feitas demonstrações de processamento com as duas soluções utilizando o serviço de Elastic MapReduce da AWS, em que a configuração para execução dos frameworks é feita de forma menos trabalhosa do que em uma nuvem privada ou em um conjunto de nós físicos. Em seguida é solicitada uma nova tarefa para os alunos, que tipicamente consiste em realizar uma análise sobre um *dataset* fornecido pelo professor.

O último tema considerado fixo abordado na disciplina é o desenvolvimento de software multi-inquilino (ou multi-arrendatário). São apresentadas as principais abordagens e modelos de organização de um software multi-inquilino, suas vantagens e desvantagens. Como tarefa relacionada, os grupos devem criar versões multi-inquilino da aplicação de lista de contatos, desenvolvida nas aulas anteriores da disciplina.

Os assuntos vistos até esta altura da disciplina são considerados essenciais, e foram vistos em todas as edições da disciplina até hoje. As aulas seguintes tratam de assuntos avançados ou correlatos ao desenvolvimento de software em nuvem. Alguns dos tópicos abordados ao longo das várias edições da disciplina incluem:

- (1) **Desenvolvimento MultiCloud:** no qual são estudadas as vantagens, desafios e principais abordagens para desenvolvimento de aplicações em execução sob múltiplos provedores de nuvem;
- (2) **Mobile Cloud Computing:** tópico que discute como complementar os recursos escassos das aplicações móveis com o potencial de processamento e armazenamento da nuvem;
- (3) **Segurança:** tema que gera bastante debate sobre o aumento ou diminuição da segurança das aplicações e dados, quando hospedados na nuvem;
- (4) **Tarifação:** aulas em que se estudam modelos econômicos e formas inovadoras para a cobrança de recursos na nuvem.

Estes assuntos variaram ao longo das edições da disciplina. Em alguns casos, outros professores ou alunos de doutorado da UFC ou de outras universidades locais, e que pesquisam sobre temas relacionados à disciplina também foram convidados e aceitaram falar sobre seus temas, o que incluiu *Cloud Gaming*, *Fog Computing*, dentre outros.

A parte final da disciplina é destinada aos seminários dos alunos de pós-graduação, e aos trabalhos finais dos grupos. Os trabalhos finais são de livre escolha, mas devem estar relacionados a algum assunto visto no curso. O professor também já oferece uma lista de possíveis trabalhos. Alguns dos trabalhos já realizados incluíram o desenvolvimento de algum serviço AWS na plataforma OpenStack, o desenvolvimento de uma aplicação utilizando um framework de Mobile cloud Computing, ou desenvolvimento de uma aplicação multicloud. Ao final da disciplina, os trabalhos são apresentados para todos os alunos.

A realização de provas é definida de acordo com o andamento da turma. Em geral, uma avaliação é feita após o tópico de elasticidade. Uma segunda prova é realizada no final de disciplina. Para computar a nota final do aluno, as notas, trabalhos parciais e trabalhos são consideradas, com um peso maior para os trabalhos práticos.

5 AVALIAÇÃO

Desde sua primeira edição (no semestre 2012.2) até a edição atual (2017.1), a disciplina de “Desenvolvimento de Software para Nuvem” já foi ofertada cinco vezes. A primeira turma teve 11 inscritos, todos alunos de pós-graduação. A turma atual tem 41 alunos, sendo 38 de graduação, o que representa um aumento de mais de 370%. Este interesse motivou a realização de uma pesquisa entre os alunos das diversas edições da disciplina, para obter um feedback sobre os assuntos abordados, expectativas alcançadas, temas mais relevantes, metodologia aplicada, além de obter críticas e sugestões de melhoria.

Para isso, foi elaborado um questionário de avaliação online na plataforma Google Forms⁵ com 32 questões relacionadas à disciplina. Todas as questões eram objetivas, com exceção de uma questão aberta de sugestões e críticas à disciplina.

O questionário foi dividido em quatro seções. A primeira trata de separar alunos de graduação e pós-graduação, além de obter as motivações que os levaram a se matricular na disciplina. Uma segunda seção busca informações sobre conhecimentos prévios sobre computação em nuvem, e sobre o desenvolvimento de software para nuvem.

Por se tratar de um assunto multidisciplinar dentro da ciência da computação, uma parte do questionário obtém informações sobre o *background* do aluno em assuntos considerados pré-requisitos como Engenharia de Software, Redes de Computadores e Sistemas Distribuídos, Análise e Projeto de Sistemas e Técnicas de Programação, e se estes assuntos eram tidos como importantes ou não para melhor entendimento da disciplina.

A seção seguinte avalia a percepção de dificuldade dos alunos em relação aos diversos tópicos apresentados na disciplina, bem como a satisfação dos mesmos em relação à ordem de apresentação dos tópicos.

Por fim, a última parte avalia o resultado final da disciplina em relação às expectativas iniciais de cada aluno, sua visão sobre a

importância do tema na sua formação profissional, a necessidade ou não de uma disciplina específica para tratar o tema, e a relação entre teoria e prática utilizada na disciplina. Nesta seção, também havia espaço para críticas e sugestões, por meio de uma questão aberta.

Ao longo das seis edições da disciplina (incluindo a turma 2017.1), 91 alunos se inscreveram na mesma, sendo 32 de pós-graduação e 59 de graduação. Foi feita uma solicitação via e-mail a todos ex-alunos, como também a alunos do semestre 2017.1, os quais até a escrita deste artigo, ainda não haviam terminado a disciplina. Esta turma está na fase de estudos sobre o paradigma *MapReduce*, e mesmo sem ter completado a disciplina, boa parte dos principais temas já haviam sido abordados. O preenchimento dos dados foi feito de forma anônima, de modo a incentivar a participação do maior número de alunos, e não inibi-los em relação às críticas. Do total, 52 alunos (34 de graduação e 18 de pós-graduação) atenderam ao convite. A seção seguinte apresenta um sumário das respostas destes alunos.

6 RESULTADOS

Os dados gerados pelos alunos permitiu a construção de gráficos para uma melhor visualização dos resultados obtidos. Porém, devido às restrições de espaço para apresentação de resultados, mostraremos aqui aqueles que consideramos mais relevantes para nossa avaliação. Também por este motivo, excetuando-se a Figura 2, o grupo de alunos é tratado sem diferenciação em relação ao nível do curso em que cada aluno estava matriculado. A Figura 2 apresenta a distribuição dos alunos em relação às cinco edições da disciplina.

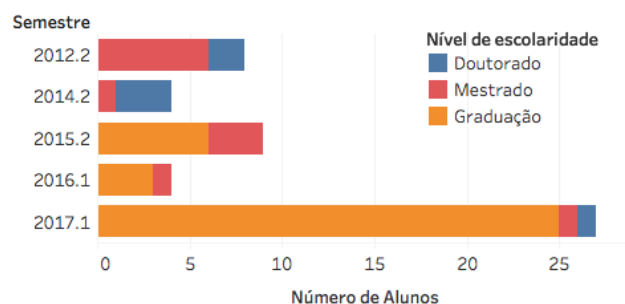


Figura 2: Distribuição dos alunos que responderam ao questionário

Na Figura 2, percebe-se uma concentração maior de alunos de graduação, e concentrados no semestre 2017.1. Em relação à primeira evidência, entende-se como fato comum, pelo menos na UFC, que em geral turmas de graduação são bem maiores que as de pós-graduação. A concentração de respostas da turma atual (2017.1) pode ser explicada por esta turma ter sido a de maior número de alunos em uma edição da disciplina (mais que o dobro da segunda maior turma), e pelo fato de que alunos de turmas passadas (alguns há mais de cinco anos) já não terem o mesmo compromisso que alunos de uma disciplina em andamento.

Uma outra questão diz respeito à motivação de cada aluno para cursar a disciplina. Cada aluno podia selecionar mais de uma opção dentre as oferecidas, e a Figura 3 mostra que a maior parte dos

⁵<http://forms.google.com>

O que me levou a me matricular na disciplina?

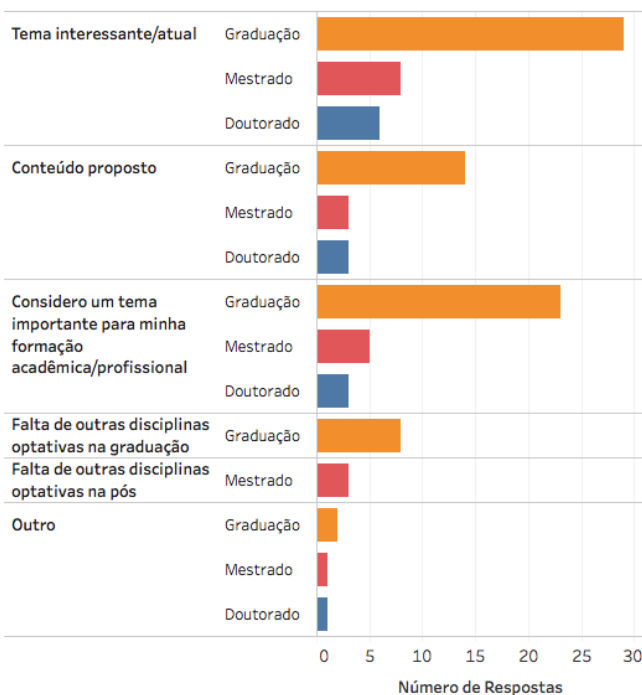


Figura 3: Motivação para cursar a disciplina

alunos optou pela disciplina por interesse pessoal ou por achar o tema relevante para sua formação. Este dado é importante pois havia uma queixa recorrente nos alunos de graduação dos cursos vinculados ao departamento de computação sobre a falta de opções de disciplina optativas, o que levava os alunos a terem que se inscrever em disciplinas que não necessariamente eram de seu interesse. Pela respostas dos alunos, não se trata do caso da disciplina em questão.

Outro levantamento importante foi em relação aos conhecimentos necessários para um bom acompanhamento da disciplina. Em vários momentos, conceitos de diferentes áreas e disciplinas são utilizados para explicação de assuntos relacionados à nuvem. A Figura 4 resume as respostas dos alunos para cinco questões do formulário. Cada questão utilizava a escala Likert para colher a opinião do aluno em relação à necessidade de conhecimentos em 5 tópicos relacionados à disciplina.

O resultado ratificou a decisão de se impor o pré-requisito de uma disciplina como “Técnicas de Programação”, uma vez que bons conhecimentos de implementação são fundamentais para os alunos conseguirem realizar os trabalhos práticos. Porém, a Figura 4 também indica que conhecimentos nas outras áreas são importantes (com uma ênfase menor) para o sucesso na disciplina. Esta importância poderia gerar problemas no acompanhamento da disciplina por alguns alunos. Porém, de acordo o perfil dos alunos da enquête, a grande maioria já havia cursado também disciplinas de “Engenharia de Software”, “Análise e Projeto de Sistemas” e “Redes de Computadores”. Já em relação à disciplina de “Sistemas Distribuídos”, metade dos alunos afirmaram já terem cursado a disciplina.

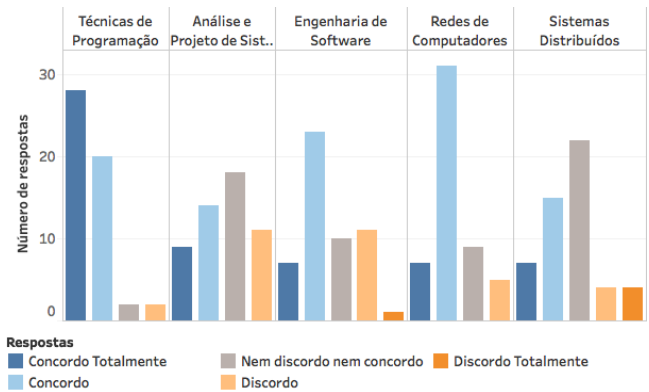


Figura 4: Avaliação dos conhecimentos prévios necessários à disciplina

As figuras 5 e 6 resumem a percepção dos alunos em relação ao conteúdo da disciplina sob duas perspectivas distintas. A primeira (Figura 5) questiona os alunos sobre quais assuntos apresentados na disciplina foram mais facilmente compreendidos, e quais foram mais complicados de se entender. O resultado mostra que os conceitos básicos sobre a nuvem e elasticidade são temas de fácil aprendizagem. Do lado negativo, o uso de PaaS, e especialmente o uso da Google App Engine, apresenta-se como o tema de mais difícil entendimento. Este resultado é justificado pela documentação pobre encontrada sobre a GAE, o que realmente dificultou o uso da plataforma. Este resultado inclusive indica que edições futuras da disciplina devem usar outra infraestrutura para as aulas sobre PaaS.

Em relação à relevância dos conteúdos (Figura 6), não houve um assunto com maior destaque entre vários assuntos da disciplina. Um resultado bem positivo é a avaliação por grande parte dos alunos de que não existe assunto irrelevante na disciplina.

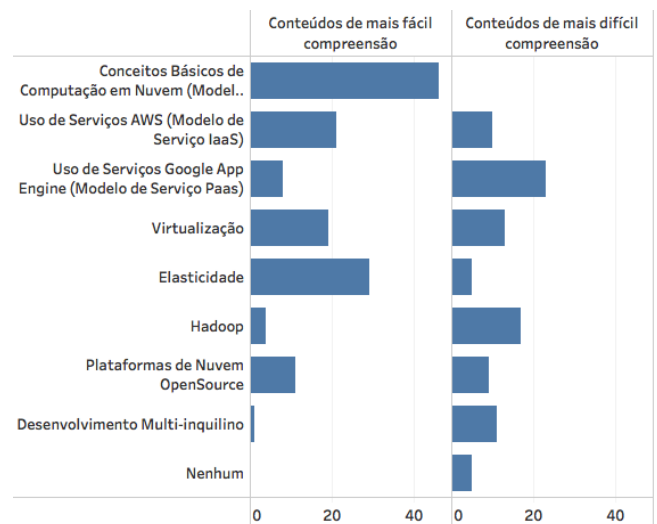


Figura 5: Avaliação de conteúdos de acordo com a facilidade ou dificuldade

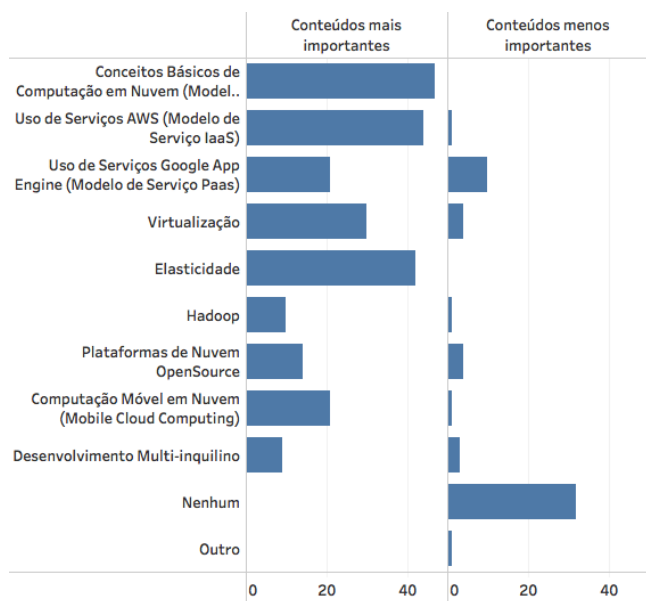


Figura 6: Avaliação de conteúdos de acordo com a relevância

O plano de aulas é um fator muito importante para o sucesso de qualquer disciplina. Por isso, alunos também foram questionados sobre a disposição dos conteúdos ao longo das aulas ministradas. A Figura 7 mostra que a maioria dos alunos ficou satisfeita com o cronograma de aulas e na forma como os temas são inter-relacionados (o que foi explicado na Seção 4). Não houve nenhuma queixa sobre esta disposição.

O questionário também buscou opinião dos alunos após a finalização da disciplina. Primeiro buscou-se avaliar a percepção dos alunos em relação à disposição entre conteúdos teóricos e práticos vistos na disciplina. A Figura 8 mostra que a maioria dos alunos considera que há um equilíbrio entre teoria e prática, o que foi um princípio muito importante no projeto da disciplina.

Já a Figura 9 sumariza a visão dos alunos em relação às expectativas alcançadas, e sobre a necessidade de uma disciplina específica para abordar o desenvolvimento de software para a nuvem. Neste caso, o resultado mostra um sumário apenas para os alunos que já efetivamente concluíram a disciplina (ou seja, sem a turma de 2017.1, uma vez que esta ainda não concluiu a disciplina).

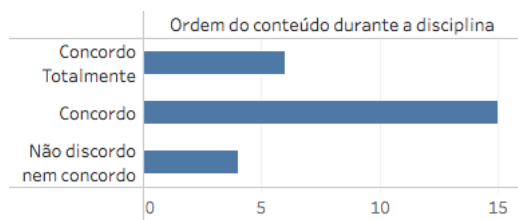


Figura 7: Avaliação do grau de satisfação dos alunos em relação à ordem de apresentação dos conteúdos na disciplina

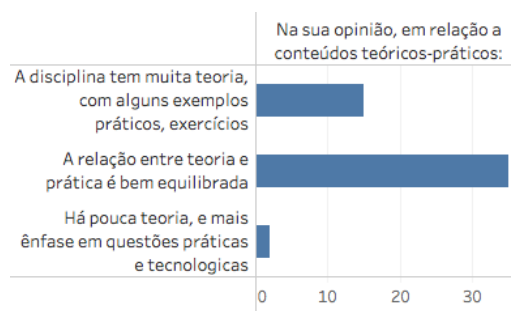


Figura 8: Avaliação da proporção entre teoria e prática utilizadas na disciplina

O resultado mostra uma avaliação 100% positiva em relação às expectativas alcançadas, e que a maior parte dos alunos concorda ser importante tratar o desenvolvimento de software para nuvem em uma disciplina separada. Este resultado corrobora com a necessidade que levou à criação da disciplina.

Em relação às respostas em aberto, o assunto mais comentado foi em relação à avaliação. Houve queixas sobre o método de avaliação ainda aplicar provas tradicionais, e da quantidade de trabalhos práticos. A sugestão de alguns alunos é que apenas as atividades práticas deveriam ser consideradas como avaliação. Outras críticas de um bom número de alunos diz respeito ao uso da Google App Engine como ferramenta de PaaS, devido à documentação pobre e desatualizada encontrada sobre a plataforma. Por fim, uma sugestão também bastante citada pelos alunos é a necessidade de mais aulas práticas em laboratório com o professor ou monitores. Esta última crítica se deu principalmente a um semestre em que por choque de horário com outras disciplinas, não foi possível usar regularmente o laboratório para aulas práticas. Todas estas sugestões foram consideradas para futuras edições da disciplina. A lista completa das observações pode ser consultada no formulário online criado para esta avaliação.

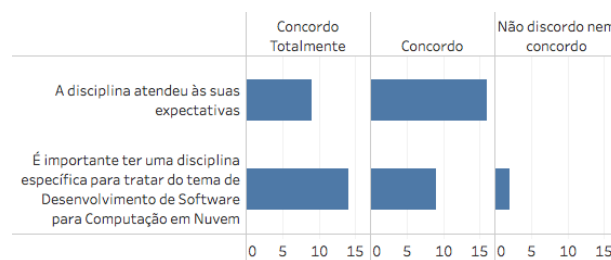


Figura 9: Avaliação dos alunos que concluíram a disciplina sobre as expectativas e a importância da disciplina

7 LIÇÕES APRENDIDAS

Os resultados obtidos na Seção 6 foram bastante interessantes, porém exclusivamente na visão dos alunos que cursaram, ou ainda cursavam a disciplina até a escrita deste artigo. Nesta seção faz-se uma análise subjetiva, a partir da perspectiva do professor da disciplina, em relação aos principais desafios encontrados, pontos

positivos e negativos ao longo das diversas turmas, e dicas para aqueles que queiram replicar a disciplina.

Primeiramente, um ponto positivo foi permitir que alunos de graduação e pós-graduação pudessem cursar a mesma disciplina. Os alunos de graduação se mostram interessados por temas propostos pelos alunos de pós, e alguns deles acabam se inscrevendo para processo seletivo no programa de pós-graduação da UFC, indicando inclusive o desejo de realizar pesquisa em temas apresentados na disciplina.

Um dos maiores problemas foi em relação ao uso dos serviços de nuvens públicas, como a AWS e a Google App Engine. Apesar destas plataformas oferecerem facilidades para sua utilização, em geral todas pedem que seja feito um cadastro para uso de seus serviços. Este cadastro exige que o usuário possua um cartão de crédito internacional para validar a conta. No caso particular da UFC, vários alunos não possuem tal requisito. A solução para o problema passou por diferentes estratégias. Uma foi a orientação para a aquisição de cartões internacionais pré-pagos de baixo custo, que podem ser adquiridos em lojas de departamento e supermercados. Quando esta solução não era viável, uma solução encontrada foi a criação de subcontas a partir da conta do próprio professor da disciplina. Este é um recurso possível na AWS, mas não na Google App Engine, o que fez dessa solução apenas um paliativo. Apesar de ser um problema para o uso individual do aluno, o problema era atenuado pois os trabalhos e atividades práticas em feitos em grupos de até 3 alunos, onde pelo menos um dos alunos não tinha problema para acesso aos serviços das nuvens públicas.

O fato de trabalhar com recursos que são tarifados e que podem gerar cobranças exige certos cuidados. Por exemplo, o mecanismo típico para uso dos serviços de provedores de nuvem é baseado no uso de chaves pública e privada. As aplicações utilizam estas chaves, e muitas vezes o fazem escrevendo diretamente sobre o código (*hard-coded*). Ao armazenar as aplicações em repositórios públicos como o GitHub, as chaves se tornam vulneráveis. Durante a disciplina, houve o caso em que um aluno teve sua conta comprometida por tal prática. O assunto foi contornado com ajuda do suporte do serviço AWS Educate, e o aluno não teve prejuízo. Portanto, é necessário reforçar com alunos as boas práticas de segurança no uso dos serviços, bem como orientá-los sobre a escolha dos tipos de recursos utilizados. Em geral, a orientação dada na disciplina foi a de sempre usar os recursos mais baratos, desalocar os recursos logo que possível e manter seguras suas chaves de acesso. Isso foi reforçado a cada prática, ilustrando as possíveis ameaças.

É preciso também que as atividades práticas sejam adequadas aos recursos disponíveis pelos provedores. No caso da disciplina em questão, a aplicação que serve de base para os trabalhos (a lista de contatos) utiliza sempre serviços da faixa gratuita (*free tier*). Porém, uma mudança na tarifação de um serviço da Google App Engine (no caso, o *BigTable*) fez com que o uso do serviço se tornasse inviável pelos alunos. Com isso, houve a necessidade de readequação da tarefa para contornar o problema.

A disciplina não impôs nenhuma linguagem de programação específica, como Java, Ruby ou Python. Para o aluno torna-se uma vantagem, pois o mesmo não se vê forçado a trabalhar em nenhuma linguagem que não seja aquela de sua preferência, desde que a mesma seja compatível com os trabalhos exigidos. Para o professor porém, acaba sendo complicado ter que lidar com os trabalhos ou

mesmo auxiliar os alunos em diferentes linguagens. Porém apesar de ser um problema, impor uma linguagem a toda turma não parece ser a estratégia mais adequada.

Por fim, as plataformas de nuvem pública estão em contínua evolução, o que faz com que constantemente a interface de gerenciamento das nuvens (o chamado *dashboard*) e o uso de seus serviços sofram profundas mudanças em curtos espaços de tempo. Com isso, manter atualizado o material de ensino e mesmo o conhecimento técnico para cada tecnologia requer um esforço tremendo por parte do professor. A solução para o problema é utilizar a própria documentação dos provedores para as aulas práticas, e contar com o auxílio de alunos de mestrado e doutorado orientados pelo professor, e que estejam em estágio em docência na pós-graduação, para lidar com questões mais tecnológicas.

8 DISCIPLINAS OU CURSOS SEMELHANTES

A partir do crescimento no tema, computação em nuvem tem sido explorada em diversos cursos no Brasil e no Mundo, nos mais diferentes níveis. É possível encontrar desde cursos de extensão de curta duração (com menos de 20h/aula)⁶ até cursos completos de especialização *lato sensu*^{7,8,9,10}.

Devido à sua interdisciplinaridade, cursos relacionados à computação em nuvem podem abordar diferentes aspectos, como aqueles relacionados à infraestrutura de redes e virtualização, ou processamento intensivo de dados e até mesmo gestão e governança de centros de dados no modelo de nuvem. Porém, a disciplina apresentada neste artigo tem foco no desenvolvimento de software para nuvem. Com este escopo mais restrito, existem algumas outras experiências com propósito semelhante, tanto no Brasil quanto no exterior.

Uma delas é a disciplina de Engenharia de Software da Universidade da Califórnia, Berkeley. Esta é uma disciplina regular de graduação, mas que desde 2010 tem o foco no desenvolvimento ágil de aplicações, onde a computação em nuvem é vista como uma plataforma de suporte para construção de aplicações no modelo Software como Serviço. De acordo com as informações de seu Web-Site¹¹, o curso cobre tópicos como desenvolvimento dirigido por testes, projeto centrado no usuário, padrões de projeto, refatoramento e código legado, integração contínua com pequenos times. Uma prática interessante é a identificação de clientes externos cujas necessidades possam ser atendidas pelos alunos da disciplina. Os alunos são agrupados em equipes de 4 a 6 estudantes que escolhem os projetos de seu maior interesse, e que ficam com a responsabilidade de entregar um protótipo funcional no modelo SaaS para atender as necessidades de seu respectivo cliente. Os protótipos são desenvolvidos para a nuvem pública Heroku. Os alunos utilizam o framework rails para construir suas aplicações e são apresentados a um conjunto de ferramentas para integração contínua, testes automatizados e controle de versão para gerenciarem seus projetos. O

⁶[http://www.cce.puc-rio.br/sitecce/website/website.dll/folder?nCurso=computacao-na-nuvem-\(cloud-computing\)&nInst=cce](http://www.cce.puc-rio.br/sitecce/website/website.dll/folder?nCurso=computacao-na-nuvem-(cloud-computing)&nInst=cce)

⁷<http://www.ead.senac.br/pos-graduacao/cloud-computing/>

⁸<https://www.fiap.com.br/mba/mba-em-arquiteturas-de-redes-e-cloud-computing/>

⁹<http://up.mackenzie.br/lato-sensu/sao-paulo/tecnologia-da-informacao/computacao-em-nuvem-cloud-computing/>

¹⁰<http://www2.sorocaba.ufscar.br/getic/>

¹¹<http://www.saasbook.info/about/ucbcs169>

uso destas ferramentas é monitorado pelos professores para acompanhar o andamento dos projetos, e também avaliar as equipes. A experiência dos professores levou a criação de um livro [3] que tornou-se uma referência para o ensino da Engenharia de Software, e a elaboração de uma versão *online* da disciplina.

No Brasil também é possível encontrar disciplinas que abordam o tema de desenvolvimento de software para nuvem em outras Universidades. Na Universidade de São Paulo (USP), a disciplina intitulada “Construção de Software como Serviço em Computação em Nuvem” é fortemente baseada na disciplina de Berkeley, e inclusive seu proponente traduziu o livro gerado em Berkeley para o português. Os objetivos incluem a introdução aos conceitos básicos de desenvolvimento de software como serviço em ambientes de computação em nuvem, fazendo com que ao término da disciplina, o estudante seja capaz de desenvolver sistemas Web com interfaces de usuário dinâmicas para sistemas de informação de médio porte com persistência de dados em bancos de dados relacionais. A metodologia de desenvolvimento baseia-se tanto nos fundamentos de arquiteturas orientadas a objetos de alta qualidade quanto em métodos ágeis de desenvolvimento de software.

Outras disciplinas relacionadas são ofertadas pela Universidade Federal de Pernambuco. São elas “Engenharia de Software” e “Serviços de Infraestrutura de Nuvem”, ambas do curso de Sistemas de Informação. A primeira segue o modelo da disciplina de Berkeley, inclusive adotando seu livro-base [3]. A disciplina segue boa parte do ciclo de desenvolvimento de aplicações, abordando desde o relacionamento com clientes até a fase de desenvolvimento e testes. A disciplina tem abordagem prática usando a metodologia de ensino baseada em problemas. Ao longo das aulas, os alunos são divididos em equipes, que tratam o problema da construção de um software, passando por todas as fases de seu desenvolvimento: da concepção até a entrega. Já a segunda educa os participantes sobre a implantação de nuvem e modelos de serviços, infraestrutura de nuvem e os principais obstáculos e considerações na migração para a computação em nuvem. O curso abrange as tecnologias necessárias para construir um data center clássico (tradicional), virtualizado e em nuvem. Essas tecnologias incluem computação, armazenamento, rede, virtualização de desktops e aplicações. Outras áreas de foco incluem backup/recuperação, continuidade de negócios, segurança e gerenciamento. Os alunos aprendem sobre as principais considerações e etapas envolvidas na transição do estado atual de seu data center para um ambiente de computação em nuvem.

De acordo com as informações obtidas sobre cada disciplina citada nessa seção, tratam-se de exemplos de cursos que utilizam a nuvem para ensino dos conceitos tradicionais da Engenharia de Software. Porém, não há uma ênfase maior sobre as diferenças e desafios que a nuvem traz para o desenvolvimento de software. De certo modo, isso pode levar a uma ponderação sobre o que seria tão diferente no uso da nuvem em relação a abordagens já conhecidas para o desenvolvimento de software, como por exemplo, as aplicações orientadas a serviço. Na visão destes autores, essencialmente não há grandes diferenças técnicas sobre o uso de APIs REST, abstrações de serviços e outros conceitos comuns. Porém, ao conceber aplicações para nuvem, alguns aspectos tem maior importância, como disponibilidade, confiabilidade, e especialmente, escalabilidade. Nossa disciplina foca nesses aspectos com mais atenção, além de abordar outros temas que, no nosso entendimento, também são

relevantes para o desenvolvimento de software atual, como uso de frameworks para computação intensiva de dados, abordagem multi-inquilino e integração da nuvem com dispositivos móveis.

9 CONCLUSÕES

A computação em nuvem é um tema recente e em crescente popularização dentro do mercado de TI. O tema tem sido explorado em diversas áreas correlatadas como redes de computadores, sistemas distribuídos e engenharia de software. O desenvolvimento de aplicações em nuvem traz novos desafios que precisam ser abordados por engenheiros de software, arquitetos e demais envolvidos no processo de criação de aplicações. Da mesma forma, é necessário que o assunto seja estudado por educadores e alunos de cursos como ciência e engenharia da computação ou sistemas de informação [8].

A proposta da disciplina “Desenvolvimento de Software para Nuvem” vai ao encontro desta questão. Sua proposta é abordar de uma forma didática os principais conceitos, desafios e estratégias para implementação de aplicações em nuvem, permitindo que alunos possam utilizar os ensinamentos obtidos por meio de exercícios e trabalhos práticos.

Ao longo de cinco anos de experiência e cinco edições, os resultados obtidos a partir do levantamento feito para este artigo mostraram que a disciplina tem conseguido alcançar seus objetivos. A disciplina também tem contribuído indiretamente na cultura de inovação do departamento de computação. Alunos com ideias para *startups* têm procurado a disciplina para entender como a nuvem pode alavancar seus produtos, de forma rápida, ágil e sem grandes custos iniciais.

Como ponto de melhoria, pretende-se incluir novos tópicos ao longo das próximas edições. Um dos temas é o de processos de desenvolvimento e governança de aplicações em nuvem, pois esta é uma parte importante, mas ainda não vista na disciplina.

REFERÊNCIAS

- [1] K. Chandrasekaran. 2014. *Essentials of Cloud Computing* (1st ed.). Chapman & Hall/CRC.
- [2] E. A. N. d. Silva and D. Lucrédio. 2012. Software Engineering for the Cloud: A Research Roadmap. In *2012 26th Brazilian Symposium on Software Engineering*. 71–80. <https://doi.org/10.1109/SBES.2012.12>
- [3] A. Fox and D.A. Patterson. 2013. *Engineering Software as a Service: An Agile Approach Using Cloud Computing*. Strawberry Canyon LLC. <https://books.google.com.br/books?id=3kqjmwEACAAJ>
- [4] Peter M. Mell and Timothy Grance. 2011. *SP 800-145. The NIST Definition of Cloud Computing*. Technical Report. Gaithersburg, MD, United States.
- [5] N. C. Mendonça. 2014. Architectural Options for Cloud Migration. *Computer* 47, 8 (Aug 2014), 62–66. <https://doi.org/10.1109/MC.2014.203>
- [6] Elias Nogueira, Ana Moreira, Daniel Lucrédio, Vinicius Garcia, and Renata Fortes. 2016. Issues on developing interoperable cloud applications: definitions, concepts, approaches, requirements, characteristics and evaluation models. *Journal of Software Engineering Research and Development* 4, 1 (2016), 7. <https://doi.org/10.1186/s40411-016-0033-6>
- [7] D. F. Parkhill. 1966. *The challenge of the computer utility*. Addison-Wesley Professional, USA.
- [8] Ian Sommerville. 2013. Teaching cloud computing: A software engineering perspective. *Journal of Systems and Software* (Feb. 2013). <https://doi.org/10.1016/j.jss.2013.01.050>
- [9] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. 2008. A Break in the Clouds: Towards a Cloud Definition. *SIGCOMM Comput. Commun. Rev.* 39, 1 (Dec. 2008), 50–55. <https://doi.org/10.1145/1496091.1496100>
- [10] Junzan Zhou, Shanping Li, Zhen Zhang, and Zhen Ye. 2013. Position Paper: Cloud-based Performance Testing: Issues and Challenges. In *Proceedings of the 2013 International Workshop on Hot Topics in Cloud Services (HotTopicS '13)*. ACM, New York, NY, USA, 55–62. <https://doi.org/10.1145/2462307.2462321>