FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

UPBox: Armazenamento na Nuvem para Dados de Investigação da U. Porto

José Pedro Marques Barbosa

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Maria Cristina de Carvalho Alves Ribeiro (Professor Auxiliar)

Co-orientador: João António Correia Lopes (Professor Auxiliar)

28 de Fevereiro de 2013

UPBox: Armazenamento na Nuvem para Dados de Investigação da U. Porto

José Pedro Marques Barbosa

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Doutor Rui Carlos Camacho de Sousa Ferreira da Silva
Arguente: Doutor José Luís Brinquete Borbinha
Vogal: Doutora Maria Cristina de Carvalho Alves Ribeiro

---

28 de Fevereiro de 2013

Resumo

As novas tecnologias digitais impulsionaram a geração de dados científicos, por um lado devido à capacidade de armazenamento digital, por outro devido à evolução de métodos e ferramentas de investigação. A partilha de dados científicos é essencial no processo de novas descobertas e é a forma pela qual os investigadores ganham reputação pelo seu trabalho.

Várias entidades, como por exemplo universidades e comunidades de investigação, disponibilizam serviços e infraestruturas para auxiliar a curadoria e partilha de dados. Contudo, o processo de depósito de dados nestas infraestruturas é complexo e exige um esforço conjunto de curadores e investigadores, pelo que muitos dos dados de investigação gerados acabam por não chegar a estes repositórios.

O objetivo deste trabalho é promover a partilha de dados de investigação na Universidade do Porto, aproximando os investigadores do processo de curadoria. Esta dissertação propõe uma abordagem diferente à curadoria de dados, fomentando a participação dos investigadores na gestão colaborativa dos seus dados de investigação. A adoção dos métodos aqui propostos permitirá, no futuro, agilizar e automatizar o processo de curadoria e submissão no repositório da Universidade do Porto.

Segundo os resultados de uma auditoria feita na Universidade do Porto com a finalidade de efetuar um levantamento das práticas dos investigadores na gestão dos seus dados, verificou-se que alguns investigadores gerem e partilham os seus dados de investigação através de aplicações de armazenamento na nuvem, como por exemplo a Dropbox.

A solução proposta, UPBox, é um serviço que suporta a gestão colaborativa de dados de investigação durante todo o processo de investigação, mantendo os investigadores em controlo da organização dos seus conjuntos de dados. Este serviço de armazenamento de dados de investigação na nuvem permite ao investigador anotar os seus dados e, quando pertinente, submete-los para curadoria, com vista a serem disponibilizados no repositório de dados da Universidade do Porto. Esta plataforma simples e familiar aos investigadores servirá, então, como intermediário entre o investigador e o repositório de dados, permitindo agilizar e facilitar todo o processo de depósito.

A UPBox é uma aplicação *web* que permite a um investigador criar áreas de trabalho por projetos, estruturadas em diretórios, e partilhá-las com outros investigadores. Está integrada com um sistema de anotação, desenvolvido num projeto paralelo, o DataNotes, que permite anotar ficheiros e diretórios.

Com o objetivo de validar a aceitação da solução, foram efetuados testes de usabilidade com investigadores. Estes testes permitiram avaliar funcionalidades e identificar melhorias a realizar, bem como tirar conclusões acerca da utilidade do protótipo desenvolvido.

Abstract

The new digital technologies boosted the generation of scientific data, on one hand, due to the digital storage capacity, on the other, due to the evolution of methods and research tools. The sharing of scientific data is essential in the discovery process and represents the dominant means by which researchers can earn credits for their work.

Several entities, such as universities and research communities, provide some infrastructures and services to improve data sharing and curation. However, the data deposit process on these insfrastructures is complex and requires the joint effort of curators and researchers, whereby much of the generated research data doesn't reach these repositories.

The purpose of this work is to promote the sharing of research data in the University of Porto, bringing the researchers closer to the data curation process. This dissertation proposes a different approach to data curation, encouraging the participation of researchers in collaborative management of their research data. The adoption of the proposed methods will, in the future, accelerate and automate the submission and curation process into the University of Porto's repository.

According to the results of an audit conducted at University of Porto in order to survey the researcher's practices in the management of their data, it was verified that some researchers manage and share their research data through cloud storage applications like Dropbox.

The proposed solution, UPBox, is a service that supports the collaborative management of research data during the research process, keeping researchers in control of their data sets. This cloud storage service allows the researcher to annotate their data and, when appropriate, submit them for curation in order to be available in the University of Porto's repository. This simple and familiar platform works as a proxy between the researcher and the data repository, accelerating the submission process into the repository.

UPBox is a web application that allows the creation of data workspaces structured in directories and sharing them with other researchers. It is integrated with an annotation system, devoloped in a parallel project, DataNotes, which enables the annotation of files and directories.

To validate the acceptance of the solution, a group of researchers were requested to conduct a usability test. The test has evaluated the available features, identified improvements and new features and provided insight on the usefulness of the the developed prototype.

Agradecimentos

A realização desta dissertação marca o fim de uma importante etapa da minha vida. Gostaria de agradecer a todos aqueles que contribuíram, de forma decisiva, para a sua concretização.

Aos professores Cristina Ribeiro e João Correia Lopes pela disponibilidade, colaboração e capacidade de estímulo ao longo de todas as fases desta dissertação.

Um especial agradecimento ao João Rocha da Silva, por toda a disponibilidade e apoio na definição deste trabalho.

À FEUP, especialmente a todos os professores que me acompanharam durante todo este percurso. Obrigado pelo conhecimento transmitido e grau de exigência elevado.

Aos meus pais e irmão, por todo o apoio durante estes cinco anos e pelo sacrifício que fizeram para que chegasse a esta fase da vida académica.

À minha namorada, pelas constantes palavras de incentivo e motivação que me permitiram seguir em frente nos momentos mais difíceis.

Por fim, mas não menos importante, aos meus amigos. Obrigado pelos bons conselhos que me deram e pela companhia durante as longas noites de trabalho.

José Barbosa

"Everything should be made as simple as possible, but not simpler."

Albert Einstein

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

API	Application Programming Interface
CSS	Cascading Style Sheets
FTP	File Transfer Protocol
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a service
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
RDF	Resource Description Framework
SIGARRA	Sistema de Informação para a Gestão Agregada dos Recursos e dos Registos Académico
SaaS	Storage as a Service
UCD	User-Centered Design
UP	Universidade do Porto
WebDav	Web Distributed Authoring and Versioning
XHR	XMLHttpRequest
XML	Extensible Markup Language

Capítulo 1

Introdução

Nas últimas décadas tem-se assistido a um aumento significativo na quantidade e complexidade de dados de investigação, muito devido à evolução de métodos, instrumentos e ferramentas aliados à capacidade de armazenamento digital atual [Fer06].

A preservação e partilha de dados científicos é essencial para reutilização por parte da comunidade. Contudo, ainda existem muitos investigadores que não partilham os seus dados de investigação por vários motivos, entre os quais a complexidade do processo de depósito de dados em repositórios [RRC12b, RRC11].

1.1 Contexto

A Universidade do Porto (UP) desenvolveu um projeto denominado *UPData* cujo objetivo foi o estudo das necessidades de curadoria de dados científicos em múltiplos domínios de investigação. No âmbito deste projeto foi desenvolvido um protótipo de um repositório de dados.

O trabalho desenvolvido nesta dissertação pretende aproximar os investigadores do processo de curadoria, oferecendo-lhes um serviço baseado em armazenamento na nuvem que lhe facilite a gestão e partilha dos seus dados de investigação.

1.2 Motivação e Objetivos

O processo estabelecido para o depósito de dados no atual repositório da UP supõe o contacto direto entre o curador e o investigador. Assim, o objetivo principal desta dissertação é aproximar os investigadores do processo de curadoria através de uma plataforma baseada em serviços na nuvem.

O protótipo desenvolvido atua na área da gestão e partilha de dados de investigação e está integrado com um sistema de anotação para permitir a descrição de ficheiros e diretórios. Espera-se, futuramente, oferecer a funcionalidade de submeter dados no repositório de dados da UP através da

UPBox. Alguns investigadores da UP armazenam os seus dados em serviços de armazenamento na nuvem externos à universidade. Com este serviço pretende-se que os dados de investigação sejam albergados em servidores seguros da UP, ao invés de servidores externos. A solução proposta inclui uma API de forma a tornar o sistema expansível a clientes externos que criem novas funcionalidades para a gestão de dados de investigação.

Questões de usabilidade, segurança e privacidade devem ser estudadas e abordadas, com o intuito de oferecer uma experiência de navegação o mais simples e segura possível.

O sistema deve ser integrável com qualquer repositório de dados, como tal devem ser mantidos todos os ficheiros armazenados por investigadores, assim como as suas anotações.

A solução desenvolvida deve ser avaliada recorrendo a testes de usabilidade que permitirão detetar aspetos a melhorar, bem como novas funcionalidades a implementar. Estão previstos testes à UPBox, no âmbito de outro projeto, que incluem o estudo da sua integração com o sistema de anotações e o repositório de dados da UP.

Esta dissertação visa melhorar o atual fluxo de inserção de dados no repositório da UP, com o objetivo de incentivar investigadores à submissão dos seus dados de investigação. Consequentemente, espera-se uma maior visibilidade da UP no que diz respeito à partilha de dados na *web*.

1.3 Estrutura da Dissertação

Para além do capítulo de introdução, esta dissertação inclui cinco capítulos. O Capítulo 2 aborda o tema da curadoria de dados e são apresentados alguns repositórios de dados relevantes nesta temática. O armazenamento na nuvem é abordado no Capítulo 3 , sendo referidos serviços e tecnologias nessa área. O Capítulo 4 descreve o problema inerente a esta dissertação e define o sistema a desenvolver através de requisitos. O Capítulo 5 é relativo ao desenvolvimento do protótipo, sendo abordada a arquitetura, a metodologia e a implementação. Os testes de usabilidade são descritos no Capítulo 6. Finalmente, no Capítulo 7 são tiradas conclusões do projeto e sugeridas orientações para o trabalho futuro.

Capítulo 2

Dados Científicos e Repositórios de Dados

O presente capítulo serve como enquadramento ao contexto deste trabalho. Inicialmente são introduzidos os conceitos de preservação e curadoria de dados e, é discutida a importância dos repositórios de dados científicos para o fluxo de trabalho de preservação.

Um crescente número de instituições tem vindo a disponibilizar repositórios para partilha e preservação de dados científicos. Nesse âmbito, são apresentadas várias iniciativas na área da gestão de dados científicos, com especial ênfase para as abordagens aos desafios encontrados.

2.1 Dados Científicos: Acesso e Preservação

Segundo Galileu, a ciência moderna baseia-se na observação e experimentação. No decorrer da investigação torna-se vital que os investigadores registem dados pertinentes acerca das observações, ensaios e experiências realizados. Mais recentemente tem-se assistido a uma crescente preocupação na preservação desses registos, ou dados científicos, que permitem a contextualização e compreensão dos mesmos, por parte de outros investigadores [Fer06].

Nas duas últimas décadas, tem-se assistido a um aumento significativo na quantidade e complexidade destes dados devido à constante evolução de métodos, instrumentos e ferramentas aliados à capacidade de armazenamento digital atual. Assim, a ciência assume uma nova dimensão: ciência intensamente baseada em dados[1] [RSR+10].

No contexto de investigação, *datasets*[2] são coleções de dados, geralmente representados sob forma tabular [RRC12b]. Cada coluna representa uma variável e cada linha representa um membro do *dataset*. Um exemplo de um *dataset* relativo à área da gravimetria poderá ser visualizado na Figura 2.1

[1] Do Inglês: *data-intensive science*.
[2] Conjuntos de dados.

grav.gpstime	grav.latitude	grav.longitude	grav.height
488743.999839	38.76028045	-27.084165796	113.155
488744.99984	38.760280428	-27.084165802	113.158
488745.999842	38.760280441	-27.084165801	113.159
488746.999843	38.76028044	-27.084165808	113.158
488747.999844	38.760280415	-27.084165805	113.158
488748.999845	38.760280466	-27.084165815	113.153

Figura 2.1: Exemplo de *dataset* na área da gravimetria [RRC12a].

Os *datasets* devem ser estruturados e organizados de forma a registarem factos relacionados num formato comum para serem facilmente acedidos e interpretados no futuro. A preservação destes *datasets* é, assim, essencial para a reutilização por parte da comunidade. Segundo a *UK e-Science*, "a partilha de dados e de recursos serão a chave para a resolução dos novos problemas da ciência e da engenharia" [Hey03].

A consciencialização da necessidade de partilha de *datasets* é cada vez maior por parte das entidades científicas, universidades e autoridades governamentais. Um exemplo disso são dados públicos sobre cidades (descrições, limites das fronteiras, pontos de interesse, entre outros) que, desde 2009, têm vindo a ser disponibilizados pelos Estados Unidos da América com o objetivo de poderem ser usados e acedidos facilmente por qualquer pessoa [SL12]. Como consequência desta iniciativa surgem várias aplicações, principalmente, móveis e *web* que utilizam estes dados em vários contextos.

Cada vez mais os investigadores estão envolvidos no processo de partilha dos seus dados, devido às vantagens oferecidas a longo prazo. Este processo concede maior dimensão, visibilidade e eventualmente continuidade ao trabalho desenvolvido. Contudo, ainda existem muitos investigadores que não partilham os seus dados, por vários motivos. Nesse sentido, as instituições de investigação têm vindo a criar políticas institucionais que induzem o depósito de dados em repositórios adequados e incentivam à partilha [RSR+10].

A partilha de *datasets* na *web* tem sido impulsionada através de repositórios de dados disponibilizados por várias entidades. Associados a estes repositórios encontram-se conceitos como a curadoria e gestão de dados que serão abordadas na próxima secção.

2.2 Curadoria de Dados

A curadoria de dados envolve manutenção, preservação e enriquecimento de dados de investigação durante o seu ciclo de vida [Cen12]. A sua gestão ativa reduz o risco de obsolescência e diminui possíveis incoerências. Outro objetivo da curadoria é reduzir a duplicação de dados, permitindo, a longo prazo, uma pesquisa de qualidade.

Em 2005 foram produzidos 150 *exabytes* de dados [Eco10a] e, como é possível verificar na Figura 2.2, desde 2007 que a quantidade de dados produzidos pela sociedade deixou de ser suportada

pela capacidade de armazenamento disponível [Eco10a]. Assim sendo, a curadoria surge, também, como um filtro ao crescimento exponencial da geração de dados nos últimos anos [Eco10b].

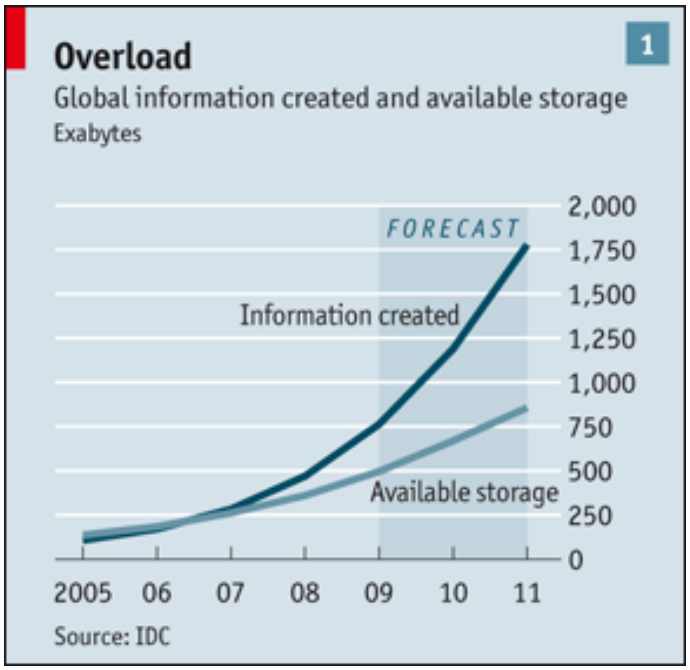

Figura 2.2: Dados criados e armazenamento disponível nos últimos anos [Eco10a].

Tipicamente existem dois tipos de atores envolvidos no processo de curadoria: investigadores e curadores. Um curador é responsável por colocar dados de investigação nos repositórios. O curador lida com documentos de outras pessoas e debruça-se, essencialmente, na preservação, indexação e classificação de documentos para que possam ser acedidos por outros através do respetivo repositório. O papel do investigador é efetuar publicações das suas investigações e experiências, dando importância à organização e anotação dos seus dados [Bun04], na medida em que a preservação é assegurada por curadores. Tradicionalmente, estes investigadores e curadores interagem durante todo o processo de depósito de dados nos repositórios, de forma a minimizar alguns dos problemas associados ao processo.

De seguida, serão apresentados os problemas mais comuns durante o processo de curadoria de dados:

- Tipicamente os *datasets* são imutáveis. E se estes mudarem frequentemente? [Bun04]
- Os repositórios multidisciplinares incluem dados de áreas bastantes distintas. Nesse caso, será um curador capaz de gerir todo o tipo de dados?
- Como selecionar os *datasets* para depositar? É pertinente preservar todos os dados gerados pelas comunidades? [RSR+10]

Cada vez mais, os responsáveis por repositórios de dados têm vindo a oferecer ferramentas, técnicas e serviços para auxiliar curadores e investigadores no processo de curadoria e resolver

alguns dos problemas acima descritos [RSR+10]. Numa tentativa de minimizar problemas, em projetos de grande dimensão, é pertinente incluir um curador na equipa de investigação [RSR+10].

As vantagens da curadoria só se fazem sentir a longo prazo, e por isso nem sempre é fácil sensibilizar a comunidade a contribuir neste processo. A DCC[3], um centro de dados do Reino Unido, implementou um conjunto de atividades com o objetivo de catalisar novos projetos de investigação e partilha resultados. As iniciativas incidiram, principalmente, em [BGAT05]:

- Promover a necessidade de curadoria entre as comunidades de cientistas e investigadores;
- Prestar serviços, com o intuito de auxiliar e facilitar a curadoria;
- Realizar ações de partilha de experiências e conhecimento em curadoria entre diversas áreas de conhecimento;
- Desenvolver tecnologia de apoio à curadoria;
- Promover investigação na área de curadoria, inovando com novas técnicas e serviços a disponibilizar à comunidade.

Este tipo de iniciativas, a longo prazo, têm tido sucesso em várias universidades e comunidades de investigação [BGAT05], contudo nem todas as comunidades têm recursos nem serviços para atuar e inovar nesta área [RSR+10].

Segundo um estudo sobre diferenças disciplinares na área da curadoria [Lyo10], torna-se necessário desenvolver estratégias para disciplinas diferentes, pois a abordagem tradicional não será suficiente para responder às necessidades dos investigadores nas diversas áreas. É precisamente este um dos principais desafios de universidades detentoras de infraestruturas de curadoria de dados científicos.

2.2.1 Projeto *DataStar*

A universidade norte-americana *Cornell University* tem vindo a desenvolver um projeto, denominado por *DataStar*, cujo objetivo é oferecer um repositório a investigadores, que apoia a partilha e gestão de dados científicos durante todo o processo de investigação [SL10]. Este repositório inclui a possibilidade de anotar os dados armazenados e submetê-los nos repositórios da instituição com a ajuda de bibliotecários.

Esta abordagem à curadoria de dados corresponde a um maior envolvimento dos investigadores, tal como se pretende neste trabalho. A ideia do projeto *DataStar* é oferecer uma plataforma de gestão de dados de investigação aos investigadores e auxiliá-los na submissão dos seus dados nos repositórios, fomentando, assim, volume de submissões. Atualmente, o projeto *DataStar* ainda não se encontra concluído, pelo que não foram tiradas conclusões acerca dos resultados desta abordagem.

[3] *The Digital Curation Centre.*

2.3 Repositórios de Dados: Abordagens à Curadoria

A fraca robustez das atuais infraestruturas é uma preocupação para as organizações que lidam com grandes quantidades de dados científicos. Serão considerados quatro cenários distintos que representam a abordagem à curadoria de dados, por parte destas organizações. Esta subdivisão foi baseada no relatório "Os Repositórios De Dados Científicos: Estado Da Arte" [RSR+10].

Curadoria por Investigadores ou Técnicos

É comum este cenário em universidades ou centros de investigação em que não existe nenhuma política institucional para a curadoria. Isto acontece, mais precisamente, em grupos que necessitem de componentes de processamento de dados em áreas pouco exploradas a nível de formatos normalizados.

Os sistemas informáticos destas infraestruturas são, geralmente, sólidos e garantem que os dados se mantêm ativos, apesar dos dados não serem salvaguardados sistematicamente.

Como exemplos de repositórios temos o CAVA[4], "Human Communication: an Audiovisual Archive" e o OASIS[5], "Open Access Series of Imaging Studies". Estes repositórios foram desenvolvidos e mantidos através de financiamentos de investigação [RSR+10].

Curadoria por Organizações Científicas

Este cenário envolve, geralmente, um trabalho em conjunto de universidades e instituições científicas que fornecem serviços de acesso dentro de uma comunidade científica. Por norma estas associações cientificas desenvolvem ou contratam infraestruturas de curadoria de dados [RSR+10].

Os dados contidos nestes repositórios são de áreas específicas e assiste-se a uma descrição mais especializada. Normalmente estes dados são restritos à comunidade em que se inserem, no entanto algumas associações tornam os seus dados públicos.

Como exemplo deste cenário temos o repositório Holandês DANS, "Data Archiving and Networked Service"[6], na área das artes, humanidades e ciências sociais, suportado pela KNAW [7] e NWO[8]. O objetivo deste repositório é manter os dados acessíveis permanentemente dando especial foco ao histórico de dados [Rom10]. Naturalmente, este repositório disponibiliza um conjunto de serviços para depósito, análise e descarga de dados.

Outros exemplos neste cenário são os repositórios ICPSR, "Inter University Consortium for Political and Social Research", *UK Data Archive*, e NCBI, "National Center for Biotechnology Information". Este último, na área da biotecnologia e biomedicina, permite aos investigadores utilizarem diretamente um conjunto de bases de dados especializadas, pelo que pressupõe alguma familiaridade com o tema e suas representações.

[4] http://www.jisc.ac.uk/whatwedo/programmes/inf11/sue2/cava.aspx
[5] http://www.oasis-brains.org/
[6] http://www.dans.knaw.nl/
[7] Royal Netherlands Academy of Arts and Sciences.
[8] Netherlands Organization for Scientific Research

Curadoria por Universidades ou Centros de Investigação

Este cenário é em todo semelhante à abordagem referida anteriormente mas inclui dados de diversas áreas por se tratarem de iniciativas de universidades ou centros de investigação. O objetivo principal destes repositórios é dar visibilidade às instituições participantes. A curadoria é abordada de várias formas nestes cenários, sendo mais comum a utilização de bibliotecas e centros de computação. Um dos maiores problemas neste panorama é a inexistência de descrições especializadas para os conjuntos de dados das diversas áreas.

Devido à variedade de áreas com que estes repositórios lidam, é importante manter proximidade entre curadores e investigadores de forma a assegurar o sucesso na curadoria de dados nas bibliotecas. Por norma existem infraestruturas duráveis que permitem registar, descrever, pesquisar e aceder aos dados científicos.

O repositório escocês *Datashare*[9], da Universidade de Edimburgo, é um exemplo deste cenário gerido por uma biblioteca. Neste caso há uma parceria com o EDINA[10], um centro de dados académicos que disponibiliza serviços à comunidade [Dat07].

Curadoria por Organismos Oficiais

Este cenário surge em países onde a sensibilidade à curadoria de dados científicos é maior, onde estas iniciativas partem de organismos de gestão da ciência do país que financiam estes projetos. A distância entre o serviço e os investigadores acaba por ser uma fragilidade desta abordagem que, aliada aos dados de diversas áreas incluídos nestes repositórios, acaba por prejudicar a descrição especializada dos dados [RSR+10].

Um exemplo deste cenário é o ANDS[11], "Australian National Data Service, financiado por diversos organismos oficiais australianos como o objectivo de dar visibilidade aos seus dados cientificos na *web*. Outros exemplos igualmente importantes são o DataONE[12] e o NBII[13], este último que recentemente foi terminado devido a cortes de orçamento do governo dos Estados Unidos da América.

2.4 Plataformas para Repositórios de Dados

Existem várias plataformas de repositórios digitais que inicialmente foram desenvolvidas para literatura científica, mas que atualmente podem ser utilizadas para diversos tipos de conteúdos digitais [RSR+10]. As plataformas mais utilizadas atualmente são o DSpace [DSp12] e o EPrints [oS11].

O DSpace é um plataforma de suporte a repositórios *open-source*, gerida pela organização DuraSpace[14]. Destina-se, essencialmente, a organizações académicas ou sem fins lucrativos. Esta

[9] http://datashare.is.ed.ac.uk/
[10] http://edina.ac.uk/
[11] http://www.ands.org.au/
[12] https://www.dataone.org/
[13] https://www.dataone.org/
[14] http://www.duraspace.org/

plataforma permite a preservação e acesso de todo o tipo de conteúdo digital, incluindo texto, imagens, vídeos e *datasets* [DSp12].

Atualmente, cerca de 1400 instituições usam esta plataforma [DSp12] como repositório de dados, estando a UP incluída nesta contagem.

O EPrints é uma plataforma bastante divulgada para repositórios institucionais utilizada essencialmente para gestão de documentos. Foi criada e é mantida pela Universidade de Southampton, Reino Unido [oS11]. As suas funcionalidades têm vindo a ser expandidas ao longo dos últimos anos, sendo que já é utilizada para acesso e preservação de qualquer tipo de conteúdo digital, à semelhança do DSpace.

2.5 UPData: Curadoria na Universidade do Porto

A Universidade do Porto tem vindo a desenvolver um projeto denominado *UPData* que tem como objetivo determinar as principais necessidades de curadoria de dados científicos em diversas áreas de investigação pertencentes à instituição. Atualmente, existe uma equipa de investigadores a cooperar com este projeto [RRC12b, RRC11].

Existe já um protótipo de um repositório de dados experimental a ser utilizado pela equipa de investigadores. Este protótipo é uma extensão à plataforma *open-source* DSpace. Esta plataforma permite a diversas organizações instalar facilmente um repositório de dados, com facilidade de personalização.

Capítulo 3

Armazenamento na Nuvem

O presente capítulo serve como enquadramento à temática de armazenamento na nuvem. Inicialmente são introduzidos os conceitos relacionados com o armazenamento na nuvem e é apresentada a arquitetura típica destes sistemas. De seguida, são apresentados os principais serviços de computação na nuvem e são enumeradas algumas plataformas *open-source* que fornecem esses serviços. Por fim, são enumeradas vantagens e desafios da adoção do modelo de armazenamento na nuvem.

3.1 Introdução

A utilização do armazenamento na nuvem tem ganho popularidade desde a década de 90, com o aumento de largura de banda da Internet. Desde então, tem-se assistido cada vez mais a uma massificação deste conceito [Moh12].

O armazenamento na nuvem é um modelo de armazenamento *online* onde os dados são mantidos, geridos e salvaguardados remotamente e são disponibilizados aos utilizadores, através da Internet [DSp11].

Os dados, a partir deste modelo, são armazenados em múltiplos servidores em vez de servidores dedicados, tipicamente empregues em redes tradicionais de armazenamento de dados. A localização dos ficheiros pode mudar a qualquer momento, visto que o sistema gere dinamicamente o espaço disponível nos vários servidores e equilibra o armazenamento, utilizando algoritmos de otimização. Contudo, apesar desta localização variável, o utilizador vê os ficheiros numa localização "estática", sendo-lhe permitida a gestão dos seus dados como se estivesse a utilizar o seu próprio computador [WPG+10].

O acesso a estes serviços pode ser efetuado através de uma *interface web*, através de API e, em alguns casos, é fornecido o acesso através de protocolos de comunicação como FTP (File Transfer Protocol) e WebDav (Web Distributed Authoring and Versioning). Este tópico será discutido mais detalhadamente na Secção 3.4.1.

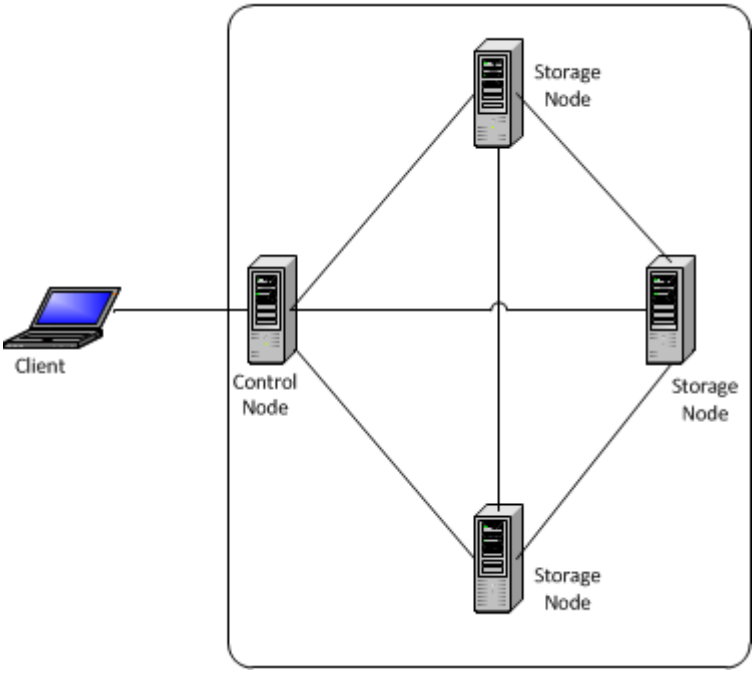

Figura 3.1: Arquitetura típica de sistemas de armazenamento na nuvem.

Existem vários tipos de sistemas de armazenamento na nuvem, alguns com um foco específico, como por exemplo, arquivar emails, outros que lidam com todo o tipo de dados e permitem a sua gestão remota. A arquitetura típica destes sistemas inclui um servidor de controlo e vários servidores de armazenamento interligados, como se pode verificar na Figura 3.1.

Tipicamente, o armazenamento na nuvem é mais barato do que a instalação e manutenção de servidores dedicados. Por este motivo, várias empresas têm adotado este modelo como resposta ao crescimento exponencial da geração de documentos. Para além de vantagens financeiras, existe replicação de dados nestes sistemas. Por motivos de segurança, os dados armazenamos na nuvem são replicados por vários servidores locais, e portanto, se por algum motivo um servidor falhar, é garantida a persistência de dados [SCL11].

3.2 Armazenamento na nuvem como um Serviço

O armazenamento na nuvem como um serviço[1] é um modelo de negócio em que grandes empresas alugam espaço das suas infraestruturas a outras empresas ou indivíduos.

Desde 2007, quando a Google propôs o modelo de armazenamento na nuvem formalmente, que têm emergido vários fornecedores deste tipo de serviços. Estes fornecedores alugam espaço de armazenamento baseado no custo por *gigabyte* e por largura de banda e garantem a manutenção e gestão das suas infraestruturas. Os seus sistemas devem obedecer a um conjunto muito rigoroso de

[1] Do inglês: *Storage as a Service(SaaS).*

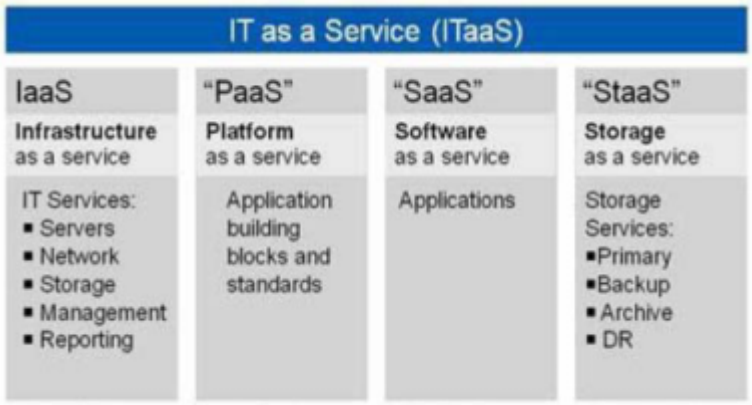

Figura 3.2: Tipos de serviços na nuvem com exemplos [WPG+10].

requisitos, incluindo replicação e consistência de dados, garantindo disponibilidade, desempenho, confidencialidade e segurança [HZZS11].

Exemplos de serviços neste ramo são: Amazon S3[2], Microsoft Azure[3] e IBM Blue Cloud[4].

Várias empresas contratam este tipo de serviços especializados a terceiros por ser mais viável economicamente, face à contratação de técnicos para implementar e manter infraestruturas de armazenamento. Isto permite à organização a possibilidade de se focar na sua área de negócio [RJKG10].

Existem vários serviços de computação na nuvem, para além do armazenamento, sendo os principais apresentados na Figura 3.2.

Em *Infraestrutura como Serviço*[5] trata-se da oferta de recursos computacionais, como por exemplo capacidade de processamento. Tipicamente os recursos são oferecidos através de máquinas virtuais, existindo uma grande escalabilidade do serviço conforme as necessidades do cliente [WPG+10]. Um exemplo deste serviço é o Amazon EC2[6].

Em *Plataforma como Serviço*[7] trata-se da oferta de uma plataforma que inclui um ambiente para produção de soluções de *software*. Desta forma o programador poderá abstrair-se do custo e complexidade de gestão de recursos de *software* e *hardware* subjacentes [WPG+10]. Exemplos deste serviço são o Heroku[8] e o Google App Engine[9].

Em *Software como Serviço*[10] trata-se da oferta de *software* na nuvem, acessível através da *web* sem a necessidade da instalação por parte dos clientes [WPG+10]. Estes serviços têm vindo a ganhar grande popularidade pela sua capacidade de centralização de recursos e por promoverem

[2] http://aws.amazon.com/s3/
[3] http://www.windowsazure.com/
[4] http://www.ibm.com/ibm/cloud/
[5] Do inglês: Infrastructure as a Service (IaaS).
[6] http://aws.amazon.com/ec2/
[7] Do inglês: Plataform as a Service (PaaS).
[8] http://www.heroku.com/
[9] http://code.google.com/appengine/
[10] Do inglês: Software as a Service (SaaS).

o trabalho colaborativo. Um exemplo bastante conhecido é o Google Apps[11] que oferece um conjunto de aplicações de escritório aos seus utilizadores.

3.3 Plataformas *open-source* de Serviços na Nuvem

Com a rápida evolução da computação na nuvem, têm surgido várias plataformas *open-source* de serviços na nuvem. Estas plataformas fornecem um conjunto de *software* que, quando instalado e configurado em infraestruturas apropriadas, permitem a terceiros a criação da sua própria nuvem [WGL+12].

Em alguns casos, poderá ser mais viável a empresas e organizações criarem a sua própria nuvem, ao invés de recorrer a fornecedores destes serviços. Este cenário é mais comum em grandes empresas ou organizações que, por um lado, utilizam muitos recursos na nuvem, o que inviabiliza economicamente a contratação destes recursos. Por outro lado, por questões de confidencialidade, os dados poderão ter que ser mantidos sobre infraestruturas próprias [RJKG10].

Neste contexto, várias entidades recorrem a plataformas *open-source* como meio de implementação das suas próprias nuvens, de forma a orquestrarem serviços de armazenamento, rede, virtualização e monitorização altamente seguros [WGL+12].

As plataformas *open-source* de serviços na nuvem mais utilizadas são o OpenNebula[12] e o OpenStack[13]. Estes projetos têm grandes comunidades envolvidas, inclusive várias empresas ativas no seu desenvolvimento, pois tratam-se de projetos de grande dimensão que envolvem diversas áreas tecnológicas.

Ambas as plataformas têm o objetivo de fornecer serviços de instalação simples, altamente escaláveis e com grande variedade de serviços, sendo o OpenNebula orientado para IaaS e o OpenStack para IaaS e SaaS.

Empresas como a Cisco, Intel, HP e Canonical utilizam OpenStack como plataforma de gestão das suas nuvens, enquanto que a DELL, SAP, IBM e RIM utilizam OpenNebula. Atualmente a FEUP está a desenvolver um projeto baseado em OpenStack com o objetivo de oferecer serviços de armazenamento e virtualização na nuvem à comunidade da UP.

3.4 Aplicações e Serviços de Armazenamento

Existem várias empresas que aproveitam o mercado do armazenamento na nuvem para criarem o seu modelo de negócio.

Atualmente, um serviço muito comum é a oferta de armazenamento *online* que permite aos utilizadores a sincronização e envio de ficheiros para a nuvem. É oferecido acesso a este sistema de armazenamento através de um navegador *web* ou de aplicações dedicadas que, quando instaladas num computador pessoal, sincronizam um diretório local com o respetivo diretório na nuvem.

[11] http://www.google.com/apps/
[12] http://opennebula.org/
[13] http://www.openstack.org/

Como exemplos que ilustram estes serviços temos a Dropbox[14], Microsoft Skydrive[15], Google Drive[16] e Ubuntu One[17]. Geralmente, as entidades responsáveis por estes serviços gerem grandes armazéns de dados em localizações geográficas distintas, no entanto a Dropbox recorre aos serviço da Amazon S3[18] como fornecedor de armazenamento na nuvem.

A simplicidade destes sistemas é a chave para o seu sucesso. Em fevereiro de 2012 a Dropbox chegou aos 50 milhões de utilizadores e segundo Drew Houston, o seu cofundador, o seu êxito pode ser resumido numa palavra: nuvem [Mur12]. Houve, ainda, uma preocupação e esforço em "criar um produto simples e elegante que satisfizesse os utilizadores".

Dentro do mesmo conceito existem várias aplicações *open-source* que podem ser instaladas numa infraestrutura pessoal e, tal como os serviços já referidos, permitem o acesso ao sistema de ficheiros através da Internet. A ownCloud[19] e FTPbox[20] são exemplos destas aplicações.

A ownCloud permite a utilização de armazenamento do servidor local ou, por outro lado, recorrer aos serviços Amazon S3 e Google App Engine como fornecedores de armazenamento. Este sistema dedicado poderá ser acedido através de qualquer dispositivo com ligação à Internet, através de um navegador *web* ou de aplicações dedicadas *open-source* fornecidas pela ownCloud. Desta forma, qualquer indivíduo ou entidade pode criar um sistema de armazenamento próprio baseado nesta plataforma, precisando apenas de instalar a aplicação servidor numa infraestrutura dedicada.

Estas soluções *open-source* não são expansíveis e são pouco personalizáveis, o que impossibilita a reutilização de algum dos seus módulos por parte de outras aplicações.

3.4.1 Acesso ao Armazenamento na Nuvem

O acesso a ficheiros armazenados em serviços de armazenamento tradicional é efetuado diretamente com o servidor, através de protocolos de comunicação como o FTP e WebDav. O WebDav é uma extensão do protocolo HTTP que detém métodos que permitem a anotação de ficheiros, a gestão de controlos de acesso, a gestão de versões, e ainda, possui métodos que implementam operações sobre ficheiros como copiar, eliminar, mover e o envio de ficheiros para o repositório [DN]. Devido a estes métodos o protocolo WebDav é muito usado em aplicações colaborativas e em aplicações que envolvam a gestão de ficheiros remota.

O armazenamento na nuvem é uma evolução do armazenamento hospedado que oferece o acesso através de API sofisticadas que permitem a abstração destes protocolos [WPG+10]. No entanto, é frequente ser suportado WebDav e FTP nestes serviços com o objetivo de permitir a integração com aplicações externas que implementem estes protocolos.

[14] http://www.dropbox.com/
[15] http://skydrive.live.com/
[16] http://drive.google.com/start
[17] http://one.ubuntu.com/
[18] http://aws.amazon.com/s3/
[19] http://owncloud.org/
[20] http://ftpbox.org/

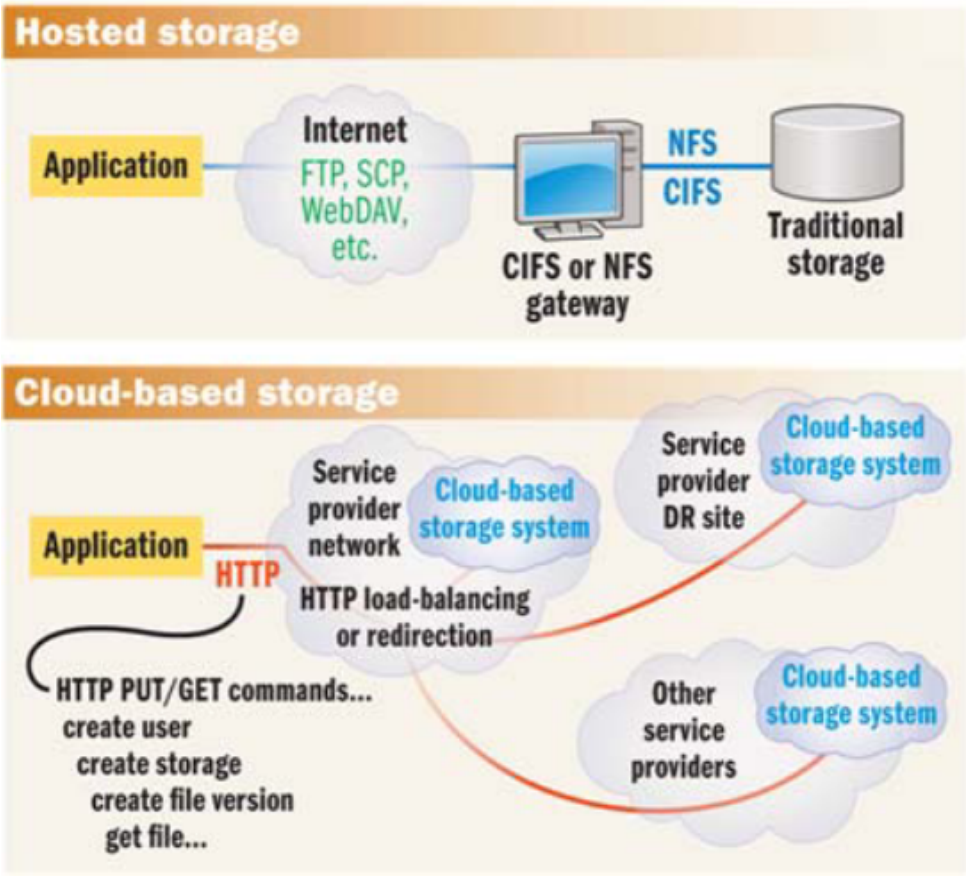

Figura 3.3: Evolução do acesso ao armazenamento na nuvem [WPG+10].

A Figura 3.3 mostra a evolução do armazenamento na nuvem com base no armazenamento tradicional ou hospedado.

3.5 Vantagens e Desafios

Como conclusão, existem vários motivos para o crescimento da popularidade do armazenamento na nuvem e para a sua viabilidade no negócio. De seguida, é apresentada uma lista de cinco benefícios chave na utilização deste tipo de armazenamento em aplicações que dele tirem proveito [WPG+10].

- Aplicações que tirem partido de armazenamento na nuvem são mais fáceis de configurar e gerir do que as tradicionais. Toda a complexidade do sistema de armazenamento é da responsabilidade do fornecedor do serviço.
- Geralmente, o armazenamento na nuvem é mais viável economicamente visto eliminar custos associados a sistemas dedicados. Atingir a qualidade (em termos de escalabilidade, segurança, disponibilidade e persistência) dos serviços disponibilizados por detentores de grandes centros de dados é muito dispendioso e praticamente inalcançável pela maior parte

das empresas. Por este motivo, os serviços de armazenamento na nuvem geridos por terceiros são, na maior parte das vezes, compensatórios.

- As atualizações de *hardware* em sistemas tradicionais causam interrupções no acesso ao armazenamento. Com armazenamento na nuvem estas atualizações não serão visíveis ao utilizador final, eliminando as interrupções no serviço.

- Os serviços de armazenamento na nuvem mantêm várias cópias de segurança em armazéns de dados situados em diversas zonas geográficas. Em caso de catástrofe natural numa zona, ou se simplesmente um armazém de dados falhar, não haverá perda de dados nem quebra no serviço.

- Os planeamentos de armazenamento detalhados já não serão um problema. Serviços de armazenamento na nuvem são flexíveis e permitem armazenamento conforme as necessidades.

Recentemente, foi feito um estudo pela Faculdade de Economia e Ciência Políticas de Londres[21] e Accenture[22] baseado em questionários a mais de mil executivos das tecnologias da informação (TI), bem como, em entrevistas a trinta e cinco prestadores de serviços na área do armazenamento na nuvem. Os entrevistados na área das TI mostraram-se mais cautelosos relativamente a prazos realistas para a implementação de nuvens do que os da área dos negócios, que estão mais interessados em soluções ágeis e rentáveis a curto prazo. Existem vários desafios na implementação de soluções de armazenamento na nuvem e esse é o motivo para a prudência dos executivos das TI [WVW12]. De seguida, são apresentados alguns desses desafios.

- Segurança: A segurança é um desafio comum a todas as aplicações acessíveis através da Internet e não um problema especifico do armazenamento na nuvem. Contudo, os grandes fornecedores de armazenamento na nuvem têm a capacidade de investir em *hardware* e *software* mais sofisticado para análise de comportamentos incomuns e deteção de vulnerabilidades, sendo que a resposta a ataques é, geralmente, bastante eficaz [Sab11].

- Aprisionamento tecnológico[23]: Atualmente, a mudança de fornecedor de serviços de armazenamento na nuvem implica custos substanciais [WVW12].

- Gestão da nuvem: Uma das grandes vantagens da utilização de serviços na nuvem disponibilizados por terceiros é a facilidade da sua atualização ou alteração, sem necessidade de intervenção interna. Esta funcionalidade disponibilizada por fornecedores destes serviços pode ser difícil de gerir [WVW12].

Os sistemas de armazenamento na nuvem são projetados para serem escaláveis e fáceis de manter. A contratação destes serviços a terceiros permite a abstração da sua complexidade, no entanto, deve ser bem planeada tendo em conta os compromissos acima abordados.

[21] London School of Economics and Political Science
[22] http://www.accenture.com/
[23] Do inglês: Vendor lock-in.

Capítulo 4

Especificação da UPBox

Este capítulo é dedicado à especificação do sistema desenvolvido. Inicialmente é feita uma descrição do problema abordado nesta dissertação e é apresentada a proposta de solução desenvolvida. Seguidamente o sistema é descrito através de requisitos funcionais e não funcionais. Por fim, são apresentados os casos de utilização da UPBox.

4.1 Descrição do Problema

Atualmente, o depósito de dados científicos no repositório experimental da UP é realizado manualmente, através do contacto direto entre o curador e o investigador. Este processo torna-se moroso, pois requer que o investigador prepare o conjunto de dados a submeter e, com ajuda do curador, os anote devidamente, para este proceder ao seu depósito.

Este projeto propõe uma nova abordagem à curadoria de dados cujo objetivo é agilizar e automatizar o processo de curadoria e submissão de dados no repositório da UP, aproximando os investigadores do processo de curadoria através de um serviço familiar de gestão e centralização dos seus dados de investigação na nuvem.

Esta ideia surgiu aquando dos questionários no âmbito do projeto UPData, em que a finalidade era efetuar um levantamento das práticas dos investigadores na gestão dos seus dados. Neste projeto, concluiu-se que grande parte dos investigadores usavam o email como ferramenta de partilha e backup dos seus dados. Por sua vez, outros já utilizavam aplicações de armazenamento na nuvem para este efeito, sendo a aplicação mais popular a *DropBox*.

Com esta tendência de alguns investigadores gerirem os seus dados de investigação em serviços de armazenamento na nuvem, a solução proposta, UPBox, pretende ser um serviço de armazenamento de dados de investigação na nuvem que permita ao investigador anotar os seus dados e, quando pertinente, submetê-los para curadoria com vista a serem disponibilizados no repositório de dados da UP.

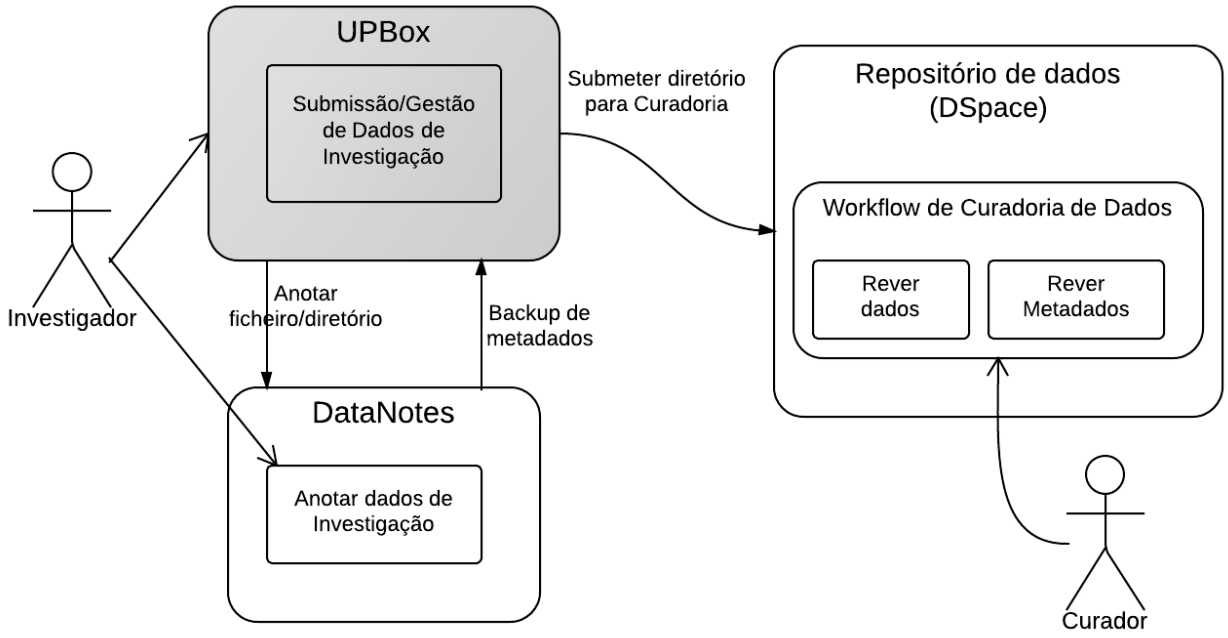

Figura 4.1: Fluxo de trabalho para a curadoria de dados na UP.

Esta abordagem pretende incluir o investigador no processo de curadoria, oferecendo-lhe esta plataforma que mantém os seus dados em servidores controlados pela instituição.

A Figura 4.1 mostra a posição da UPBox no processo de curadoria, bem como a sua integração com outras plataformas. O primeiro contacto do investigador com este processo será com a UPBox com a submissão e gestão dos seus dados de investigação. A qualquer momento o investigador poderá anotar os datasets, sendo para tal direcionado para o DataNotes, um sistema de anotação de ficheiros e diretórios. Estas anotações poderão ser criadas e editadas quando oportuno e sempre que se mostre necessário, sendo que no final de cada edição o utilizador é reencaminhado para a UPBox e é recebido um ficheiro com o *backup* da anotação efetuada. Esta anotação servirá para uso futuro, fora do âmbito desta dissertação.

Quando pertinente, o investigador poderá submeter os seus dados e anotações para curadoria, com vista a serem publicados no repositório de dados da UP, através da UPBox. Estes dados serão disponibilizados ao curador através da ferramenta de curadoria de dados do DSpace para os validar e submeter no repositório. Caso surja alguma dúvida, aquando da curadoria, o investigador será contactado diretamente pelo curador para resolver o problema.

Os seguintes subcapítulos detalham os requisitos funcionais e não funcionais da solução proposta, bem como a descrição dos seus casos de uso.

Tabela 4.1: Requisitos funcionais do sistema.

Identificador	**Descrição**
RF01	Autenticação com credenciais da UP.
RF02	Gestão de projetos e de acesso aos mesmos.
RF03	Descarregar ficheiros armazenados.
RF04	Gestão de ficheiros armazenados.
RF05	Carregamentos de múltiplos de ficheiros.
RF06	Compressão de ficheiros.
RF07	Descompressão de ficheiros.
RF08	Navegação em diretórios idêntica a sistemas de ficheiros.
RF09	Deteção de erros em criação de diretórios e upload de ficheiros.
RF10	Obtenção de dados através de API.
RF11	Comunicação com o *Datanotes* para anotação de ficheiros.
RF12	Receção de backups de anotações do *Datanotes*.
RF13	Marcação de diretórios para curadoria.

4.2 Requisitos

Esta secção é dedicada à descrição dos requisitos para a solução proposta. Existem dois tipos de requisitos: funcionais e não funcionais. Os requisitos funcionais [Som06] são definidos como serviços que o sistema deve fornecer e como o sistema se deve comportar determinadas situações. Os requisitos não funcionais são definidos como restrições sobre os serviços oferecidos pelo sistema.

4.2.1 Requisitos Funcionais

A Tabela 4.1 define os requisitos funcionais definidos para esta dissertação.

A autenticação no protótipo desenvolvido deve ser efetuada com as credenciais do utilizador do SIGARRA[1]. Isto deve-se ao facto de todos os utilizadores do sistema pertencerem à UP, evitando, assim, a necessidade de um registo.

O sistema desenvolvido pretende ser colaborativo, oferecendo funcionalidades de criação e gestão de projetos, bem como gestão de permissões de acesso ao mesmo. Um projeto tem associado vários ficheiros e diretórios que podem ser geridos e acedidos remotamente por qualquer colaborador do projeto.

O requisito funcional RF05 refere-se ao carregamento de dados de investigação. O sistema deve permitir o *upload* simultâneo de vários ficheiros e mostrar o estado da operação ao utilizador do sistema. Deve permitir, também, efetuar mais *uploads* de ficheiros para o sistema ainda que um *upload* anterior não tenha terminado.

Deverá ser disponibilizada uma API para acesso aos dados presentes no servidor para permitir a expansão da UPBox a outras aplicações que, por exemplo, permitam sincronizar um diretório pessoal com um do servidor.

[1] Sistema de Informação para a Gestão Agregada dos Recursos e dos Registos Académico.

O sistema deve encaminhar o utilizador para o DataNotes caso este pretenda anotar um ficheiro ou diretório. A qualquer momento poderá ser recebido um *backup* de uma anotação por parte do DataNotes, que deve ser armazenada localmente para preservação e utilização futura.

Por fim, quando os dados de investigação estiverem prontos para serem migrados para o repositório de dados da UP, o sistema deve permitir ao investigador a disponibilização de um diretório para curadoria.

4.2.2 Requisitos Não Funcionais

É importante oferecer um serviço vantajoso aos investigadores, do qual estes tirem o máximo proveito e que resolva alguns dos seus problemas com grande simplicidade. A usabilidade e simplicidade são portanto requisitos chave, não funcionais, da UPBox. Assim sendo, foram adotadas as cinco componentes de usabilidade sugeridas por Jakob Nielsen [Nie94]:

- **Facilidade de aprendizagem**: O utilizador consegue facilmente utilizar o sistema sem o conhecer;
- **Utilização eficiente**: Um utilizador que conhece o sistema efetua tarefas rapidamente;
- **Facilidade de memorização**: Um utilizador menos frequente consegue utilizar o sistema sem ter que aprender a utilizá-lo novamente;
- **Baixa taxa de erros**: Os utilizadores cometem poucos erros a utilizar o sistema e, se cometerem algum, facilmente o resolvem;
- **Satisfação**: Os utilizadores têm prazer em utilizar o sistema.

Tendo em conta estes aspetos, pretende-se que o sistema seja simples e que utilize padrões de *design* utilizados por sistemas que os investigadores utilizam, facilitando, assim, a aprendizagem à utilização do mesmo. Para além disto é importante incluir várias ajudas e dicas de utilização, bem como feedback na sua utilização, como por exemplo deteção de erros e sugestões de correção.

A segurança e privacidade são, também, aspetos a ter em conta neste sistema. Todos os dados armazenados no sistema só podem ser acedidos pelos seus criadores ou por investigadores com acesso ao projeto. Os dados só poderão ser acedidos por outros após o seu detentor os disponibilizar para curadoria. Neste processo de curadoria os dados vão ser analisados por curadores com vista a serem disponibilizados à comunidade da UP no repositório de dados.

O sistema deve ser interoperável, visto que o terá que comunicar com o sistema de anotação, DataNotes, e com o sistema de autenticação da UP. Esta comunicação terá que ser o mais transparente possível de forma a não confundir o utilizador na navegação.

É importante garantir confiabilidade, fiabilidade e robustez no sistema. Os dados dos investigadores estarão em servidores da instituição e inacessíveis a utilizadores não autorizados. Por outro lado é importante garantir que o sistema esteja sempre disponível aos investigadores.

4.3 Casos de uso

A Figura 4.2 apresenta os casos de uso do sistema através de um diagrama.

Os casos de uso englobam um módulo de gestão de ficheiros remota com funcionalidades que permitam ao investigador gerir os seus dados. Este módulo inclui a anotação de ficheiros, uma funcionalidade disponibilizada pelo sistema de anotação externo[2].

Como o sistema é dedicado a investigadores da UP, a autenticação será efetuada com as credenciais da UP evitando, assim, um registo para a sua utilização.

Falta referir que o sistema possui funcionalidades que permitem a criação de projetos e gestão de colaboradores. Um investigador poderá criar um projeto e, se pretender, adicionar outro investigador ao projeto, dando-lhe acesso aos ficheiros lá contidos e permissão para adicionar novos ficheiros .

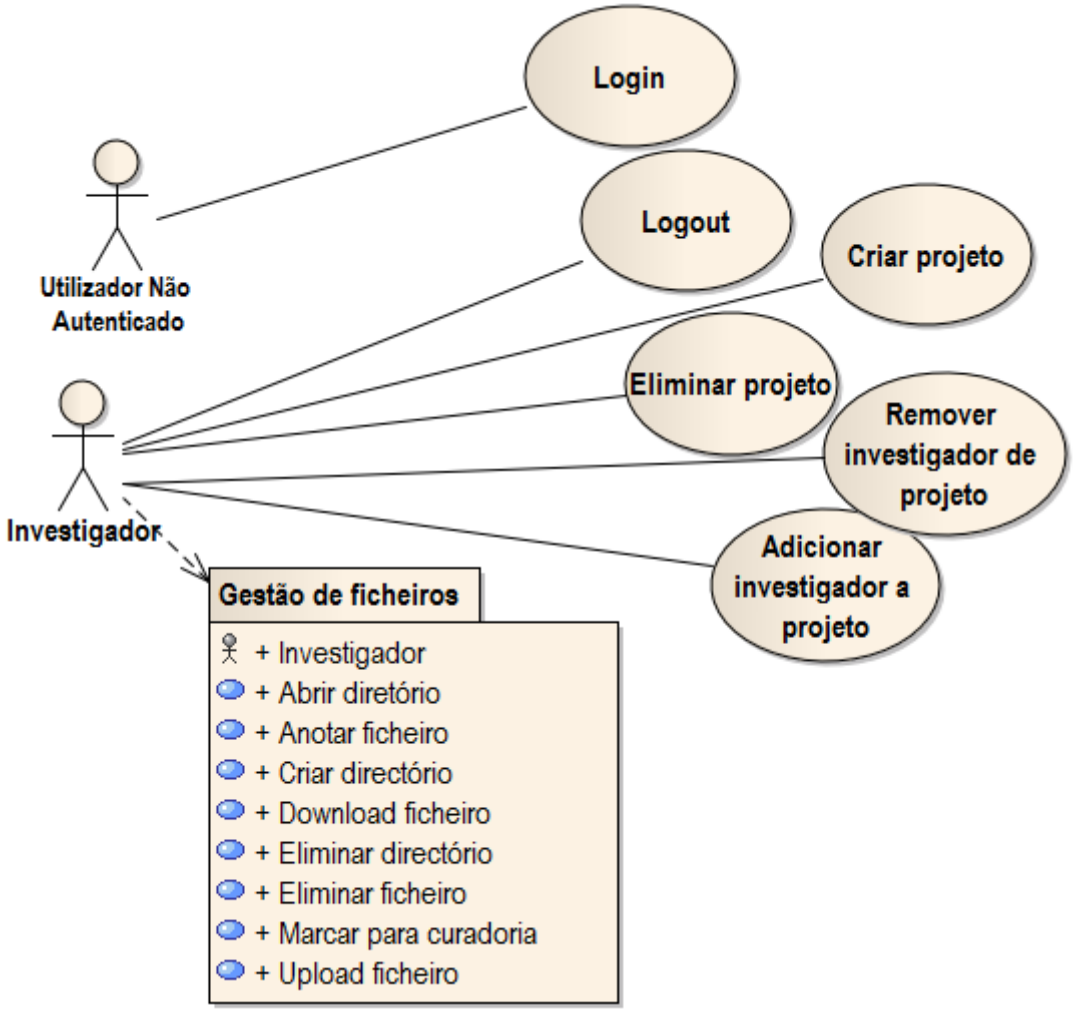

Figura 4.2: Casos de uso do sistema.

[2] *Datanotes*

Capítulo 5

Desenvolvimento da UPBox

Este capítulo é dedicado à apresentação do desenvolvimento da UPBox. Na Secção 5.1 é definida a arquitetura do sistema, e são descritos os módulos do sistema, apresentados o modelo conceptual e o modelo de dados e definida a API a desenvolver. A Secção 5.2 descreve a metodologia de desenvolvimento adotada e, por fim, na Secção 5.3, são descritas algumas fases de implementação do protótipo desenvolvido, bem como as decisões tomadas ao longo de todo o desenvolvimento.

5.1 Arquitetura

A Figura 5.1 apresenta o diagrama de componentes, ligações e máquinas onde é retratada a arquitetura física do sistema.

O sistema desenvolvido e a base de dados encontram-se hospedados num servidor que fornece duas interfaces de comunicação: o *web site* e a API. Esta última é descrita na Secção 5.1.4. O *web site* poderá ser acedido por qualquer utilizador através de um navegador. A API fornece serviços para, futuramente, a UPBox ser estendida a clientes externos e serviços para a comunicação com o DataNotes.

Atualmente, o protótipo desenvolvido utiliza armazenamento local do servidor, no entanto, quando a aplicação for colocada em ambiente de produção, prevê-se a utilização de um sistema de armazenamento externo, à semelhança do *feupload*[1]. A migração para este sistema de armazenamento externo será simples, isto porque o servidor o vê como um diretório local. O sistema de ficheiros RAID5 garante a persistência de dados por *hardware* visto existir replicação de dados e a ligação ao servidor é feita por fibra ótica, permitindo assim uma maior largura de banda.

[1] Serviço desenvolvido pelo Núcleo de Informática da FEUP e mantido pelo CICA que permite o armazenamento e partilha de ficheiros.

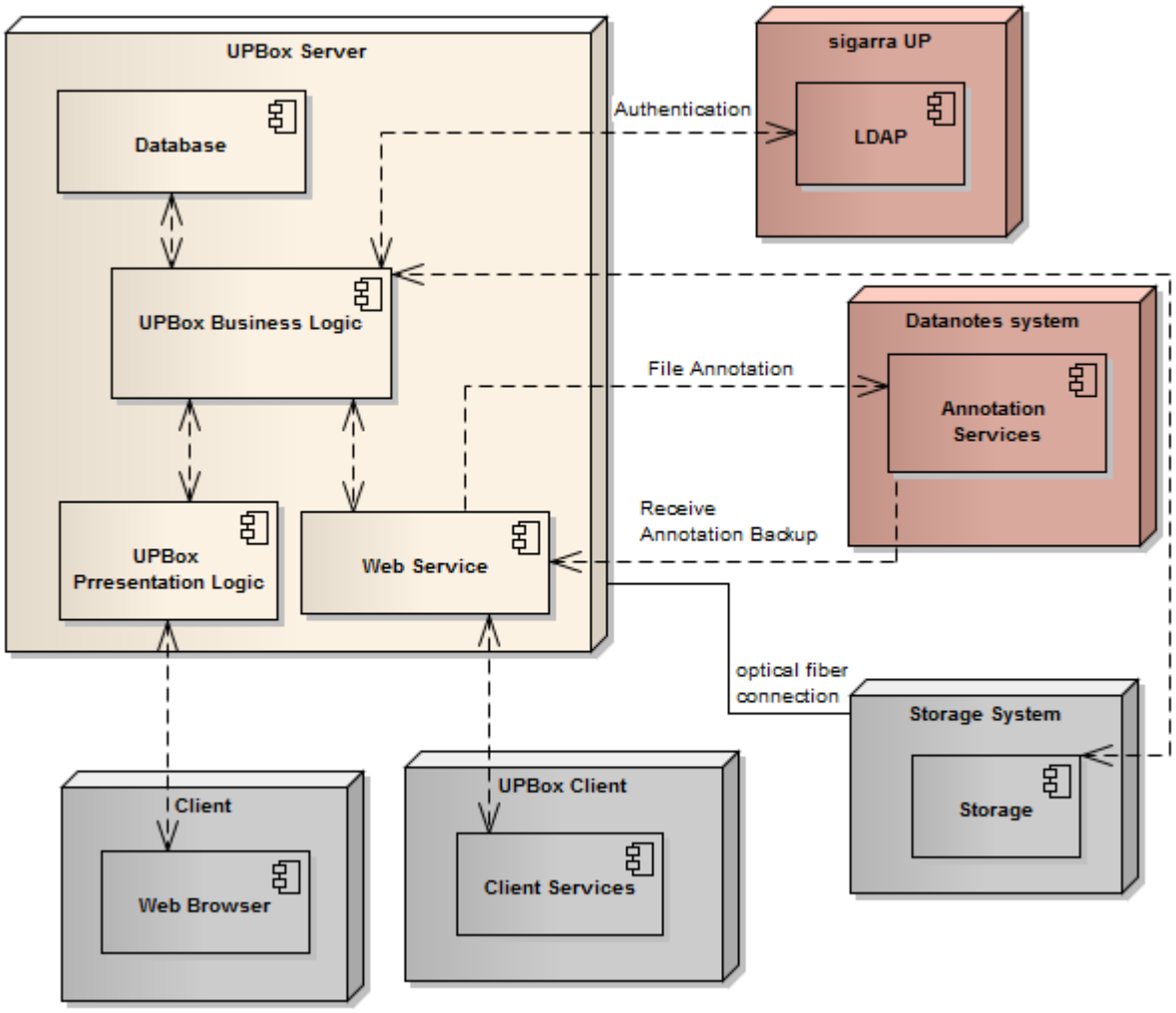

Figura 5.1: Diagrama de instalação.

Idealmente, o sistema devia recorrer a uma infraestrutura de armazenamento na nuvem idêntico ao Amazon S3, que evitaria a necessidade de criação do módulo de armazenamento da aplicação, abstraindo o programador de conceitos de segurança, escalabilidade e cópias de segurança. No entanto, não existe, atualmente, nenhum sistema de armazenamento na nuvem disponível para aplicações desenvolvidas na UP. Atualmente, o CICA está envolvido num projeto que visa o desenvolvimento de uma plataforma de serviços na nuvem baseada em *OpenStack*, sendo um deles a disponibilização de armazenamento a aplicações externas que poderá ser usado futuramente pela UPBox.

Para autenticação de utilizadores no sistema com as suas credenciais da UP recorre-se ao SIGARRA, através de uma ligação segundo o protocolo LDAP.

5.1.1 Módulos do Sistema

O sistema desenvolvido pode ser decomposto em quatro módulos: utilizadores, projetos, armazenamento e anotações.

O módulo de utilizadores é responsável pela autenticação de utilizadores no sistema e pelo seu registo na base de dados. Estes utilizadores podem adicionar ao sistema projetos de investigação e partilhá-los com outros utilizadores, com o intuito de lhes dar permissões de acesso. Os projetos têm associados conjuntos de diretórios e ficheiros disponibilizados pelos seus intervenientes.

O módulo de armazenamento é responsável pela gestão de ficheiros e diretórios. São fornecidas funcionalidades para criação, remoção e anotação de diretórios e para *upload*, descarga e anotação de ficheiros. Este módulo é, também, responsável por gerir o sistema de *uploads* múltiplos e simultâneos.

O módulo de anotação é responsável por receber *backups* de anotações do sistema de anotações e mantê-las e armazená-las paralelamente aos ficheiros e diretórios. Este módulo fornece toda a comunicação com o sistema externo de anotação e é responsável por reencaminhar corretamente os utilizadores para o mesmo.

5.1.2 Modelo Conceptual do Domínio

A Figura 5.2 apresenta o diagrama de classes simplificado do sistema, contendo as suas classes e principais atributos.

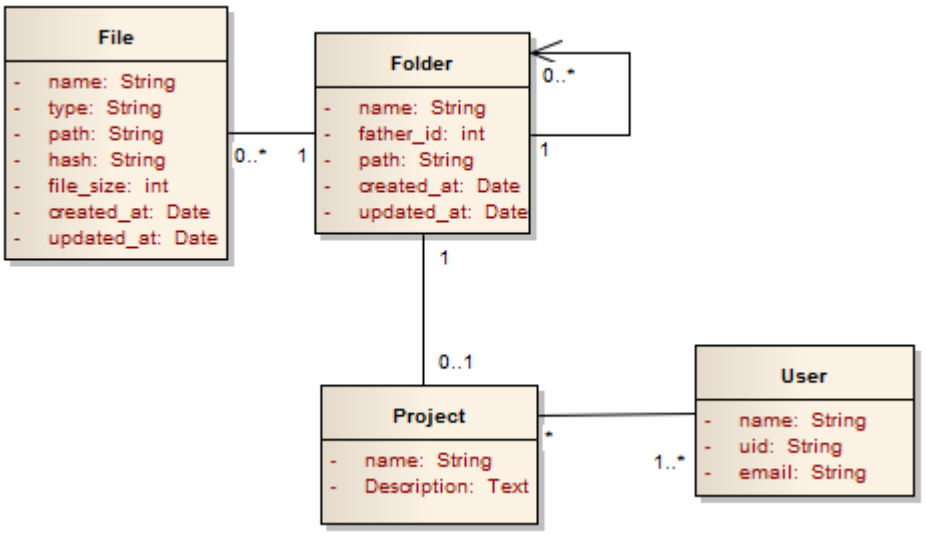

Figura 5.2: Modelo conceptual do domínio.

Os conceitos inerentes às classes já foram previamente descritos, pelo que só serão explicadas as suas relações. Um utilizador do sistema poderá pertencer a vários projetos podendo ser criador, ou não, de alguns deles. Um projeto tem associado um diretório onde serão armazenados todos os dados com eles relacionados.

As classes *Folder*, referente aos diretórios, e *File*, referente aos ficheiros, estão ligadas de modo a criarem uma estrutura em forma de árvore. Uma *Folder* poderá ter vários filhos do tipo *Folder* e do tipo *File*. A Figura A.1, presente no Anexo A, demonstra a estrutura em árvore de um diretório exemplo.

5.1.3 Esquema da Base de Dados

A Figura 5.3 apresenta o diagrama de base de dados mapeado a partir do modelo de classes da Figura 5.2.

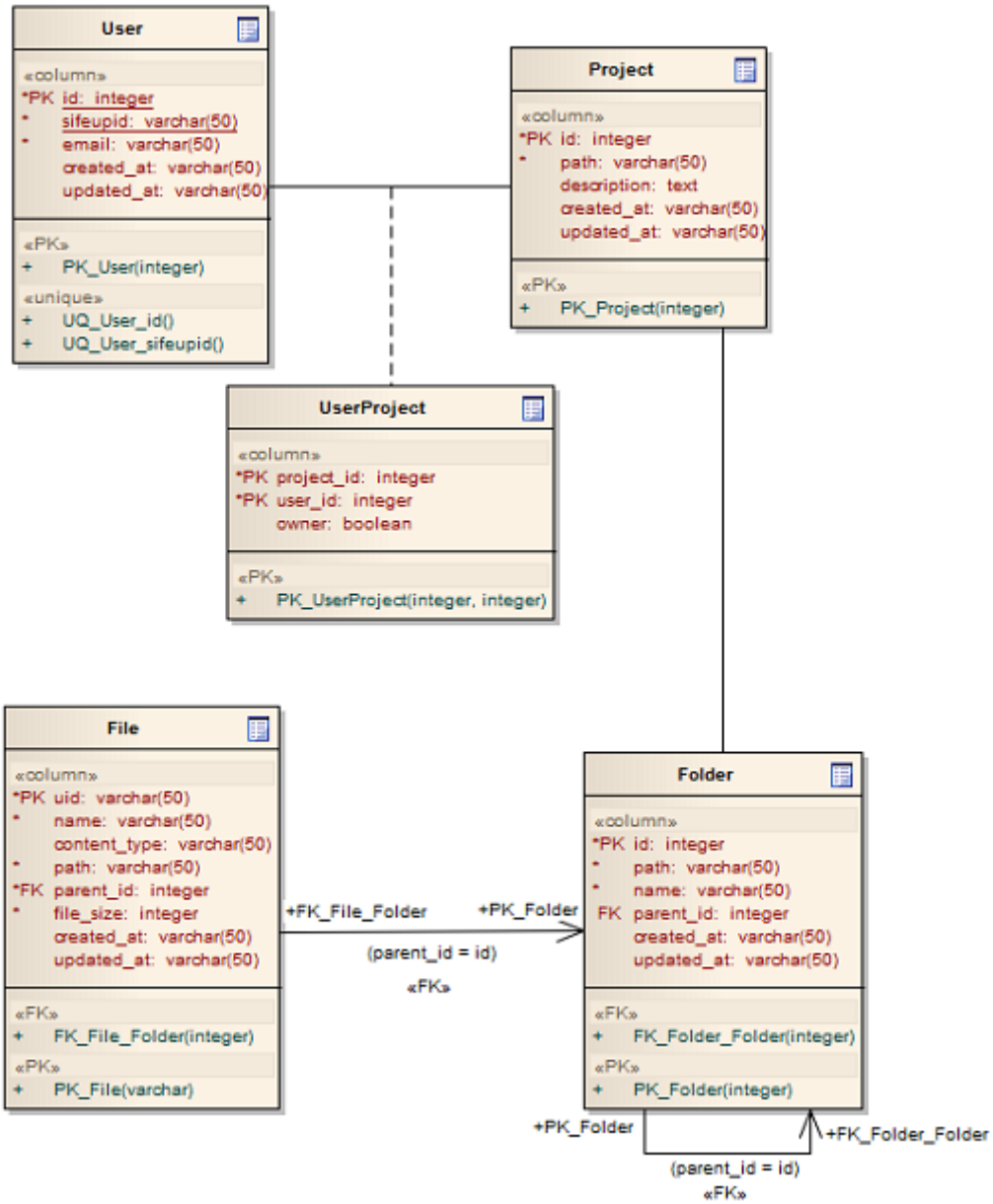

Figura 5.3: Diagrama de base de dados.

Este modelo foi concebido para armazenar e relacionar toda a informação previamente mencionada. Todas as tabelas podem, facilmente, ser associadas às classes apresentadas na secção anterior. A única tabela que não está relacionada com nenhuma das classes anteriores é a *User-Project*, pois trata-se de uma tabela auxiliar intermédia que permite a ligação entre utilizadores e projetos.

É importante referir que a classe *Folder* possui uma chave estrangeira para ela própria, através do atributo *parent_id*. A classe *File* também detém uma chave estrangeira para a classe *Folder*. Estas ligações permitem a estruturação de diretórios numa árvore.

5.1.4 Especificação da API

As operações suportadas pelo sistema estão listadas na Tabela 5.1. Estas operações estão descritas com maior detalhe no Anexo B.

Tabela 5.1: Lista de operações da API.

Operação	**URI**	**Método HTTP**	**Parâmetros**	**Retorno**	**HTTP Status Codes**
OP01	/login	POST	Username, Password	cookie	200; 400
OP02	/home/<PROJ>/<PATH>	GET	cookie (user); datanoteskey (for datanotes)	Estrutura do diretório em XML ou JSON	200; 400; 401; 404
OP03	/download/<UID>	GET	cookie	O ficheiro pedido.	200; 401; 404
OP04	/upload	POST	cookie, path, file	-	200; 401; 404
OP05	/create_folder	POST	cookie, path	-	200; 401; 403; 406
OP06	/delete_folder	GET	cookie, id	-	200; 401; 404;
OP07	/delete_file	DELETE	cookie, uid	-	200; 401; 404;
OP08	/userprojects	DELETE	cookie (user); datanoteskey (datanotes); uid	*Array* de projetos em JSON ou XML.	200; 401; 404;
OP09	/backup_annotation	POST	datanoteskey; path	-	200; 404;

Os serviços OP02, OP08 e OP09 são relativos à comunicação com o sistema de anotação. Nestes pedidos é requerida a *datanoteskey* que é uma chave pré-combinada e que garante a autenticidade do pedido. O método OP02 retorna a árvore do diretório especificado no pedido em XML ou JSON. A operação OP08 retorna os projetos de um utilizador e a OP09 submete um backup de uma anotação para ser armazenada no sistema.

À exceção do serviço OP09, todos os restantes podem ser utilizados por aplicações que estendem a UPBox. A operação de */login* retorna um *cookie* que deverá ser enviada sempre que for efetuado qualquer outro pedido para garantir a sua autenticidade. Os serviços oferecidos englobam funcionalidades de obtenção de dados por parte do servidor, bem como operações sobre ficheiros contidos no mesmo.

5.2 Metodologia

A metodologia adotada para o desenvolvimento do protótipo foi a *user-centered design*(UCD). Esta metodologia baseia-se na compreensão das necessidades e objetivos dos utilizadores, aquando do planeamento e desenvolvimento, de forma a obter um produto mais adequado e usável.

Todas as interfaces, descritas na Secção 5.3.5, foram criadas tendo por base guias de *design* para aplicações *web* já existentes. Também foram estudadas e analisadas aplicações familiares e semelhantes às do protótipo a desenvolver, de modo a fornecer um serviço de fácil aprendizagem e de rápida execução de tarefas.

A representação do processo de desenvolvimento da metodologia UCD pode ser observada na Figura 5.4.

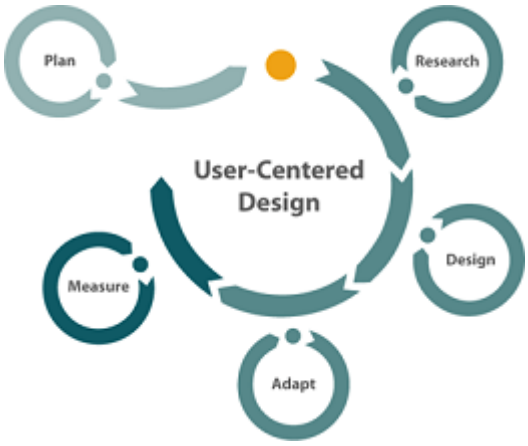

Figura 5.4: Processo de desenvolvimento em *user-centered design*[2].

5.3 Implementação

Esta secção descreve a fase de implementação do protótipo desenvolvido, onde são enumeradas as tecnologias adotadas e, posteriormente, é exposta toda a implementação dos módulos previamente definidos, justificando, sempre que necessário, as escolhas efetuadas. Por fim, é dada a conhecer a abordagem ao instalação da aplicação, enumerando as soluções aos problemas inerentes.

[2]Imagem retirada de: http://www.sapdesignguild.org/resources/ucd_process.asp

5.3.1 Tecnologias

O protótipo foi desenvolvido em Ruby on Rails[3]. A escolha desta *framework* deveu-se à sua componente ágil de desenvolvimento, bem como à familiaridade com a mesma, o que permitiu responder rapidamente às mudanças ao longo do desenvolvimento. Para a *interface web* foi utilizado HTML e CSS, recorrendo sempre que necessário à linguagem de *scripting javascript* e à *framework jquery*. Recorreu-se, também, à *framework Bootstrap*[4], de forma a agilizar o desenvolvimento de *interfaces* simples e intuitivas.

Relativamente à base de dados utilizou-se *SQLite*[5] para desenvolvimento, devido à sua simplicidade e integração com Ruby on Rails. Em ambiente de produção foi utilizado *PostgreSQL*[6] por obter um melhor desempenho e escalabilidade a longo prazo.

A aplicação foi implementada e testada no sistema operativo *Linux Mint*[7] e *Debian*[8], no entanto funcionará em qualquer sistema operativo baseado em UNIX[9]. Utilizou-se a ferramenta *nginx*[10] como servidor *web*, aquando da instalação da aplicação.

5.3.2 Autenticação

Um dos requisitos do sistema[11] é a autenticação de utilizadores com as suas credenciais do *SIGARRA*. A única forma de o fazer é através do protocolo LDAP disponibilizado pelos servidores do CICA[12] baseado na ferramenta *open-source OpenLDAP* [13], sendo a versão do protocolo a V3[14].

O LDAP é um protocolo cliente-servidor para acesso a serviços em diretórios, concebido para fornecer um diretório digital, que é equivalente, por exemplo, a um catálogo de endereços. Esta estrutura de diretórios é idêntica a uma base de dados, onde os dados podem ser organizados em árvores como as de um sistema de ficheiros [Mal07].

Existe uma camada de serviços fornecidos pelo LDAP como a pesquisa em entidades complexas aplicando filtros, comparação de resultados, leitura e escrita em diretórios e autenticação, sendo esta última a mais relevante neste projeto.

A árvore parcial dos diretórios do SiFEUP é apresentada na Figura 5.5. A componente de domínio (*dn*) está relacionada com o DNS, no caso do SIGARRA é *fe.up.pt*. O *ou* é a unidade da organização e o *uid* é o identificador do utilizador.

[3] http://rubyonrails.org/
[4] http://twitter.github.com/bootstrap/
[5] http://www.sqlite.org/
[6] http://www.postgresql.org/
[7] http://www.linuxmint.com/
[8] http://www.debian.org/
[9] http://www.unix.org/
[10] http://nginx.org/
[11] RF01 na Tabela 4.1
[12] http://sigarra.up.pt/feup/pt/noticias_geral.ver_noticia?p_nr=3201
[13] http://www.openldap.org
[14] Mais informação em: http://www.ietf.org/rfc/rfc3377.txt

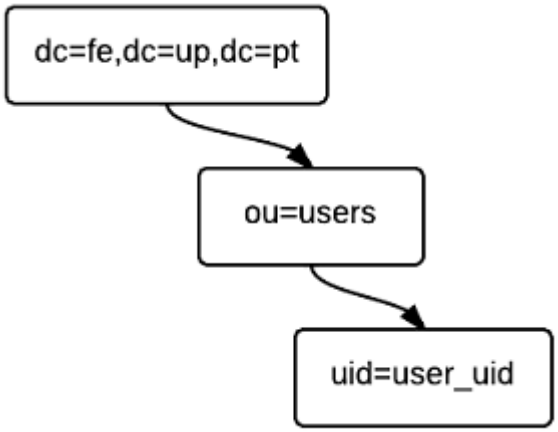

Figura 5.5: Árvore parcial dos diretórios do *SIGARRA*.

Cada utilizador possui um nome distinto (*dn*) pelo qual é identificado. Este é composto pela informação contida na árvore de baixo para cima, neste caso seria: *dn: uid=user_id, ou=users, dc=fe, dc=up, dc=pt*.

Para autenticar o utilizador na UPBox é utilizado o método *BIND* do LDAP, que retorna os dados de um utilizador dado o seu *dn* e chave de acesso. Se for a primeira autenticação do utilizador, os seus dados são guardados na base de dados, à exceção da senha de acesso, e é criada uma sessão para garantir a autenticidade da sua navegação.

5.3.3 Gestão de ficheiros

Após a autenticação o utilizador terá acesso ao resto das funcionalidades da aplicação, que englobam o módulo de projetos e de gestão de ficheiros. Os dados serão armazenados na UPBox associados a projetos, portanto implementou-se um módulo que permite a criação de projetos e a gestão do mesmo.

Quando um utilizador abre um projeto, entra no ambiente de gestão de ficheiros, como pode ser visualizado na Figura 5.12. Este ambiente inclui, também, a gestão de acessos ao projeto onde é possível dar permissão de acesso ao projeto a outros utilizadores.

As secções que se seguem descrevem alguns pormenores e decisões de implementação na gestão de ficheiros e, no final, são apresentadas algumas considerações de interface e usabilidade a ter em conta na implementação.

5.3.3.1 Estruturas de diretórios

A Figura 5.6 mostra uma estrutura de diretórios exemplo presente no sistema de armazenamento. A estrutura de diretórios é, também, mantida na base de dados por motivos de consistência.

Cada utilizador tem associado um diretório, no qual são armazenados todos os projetos criados por este. Sempre que um utilizador dá permissão de acesso ao projeto a outro utilizador os dados

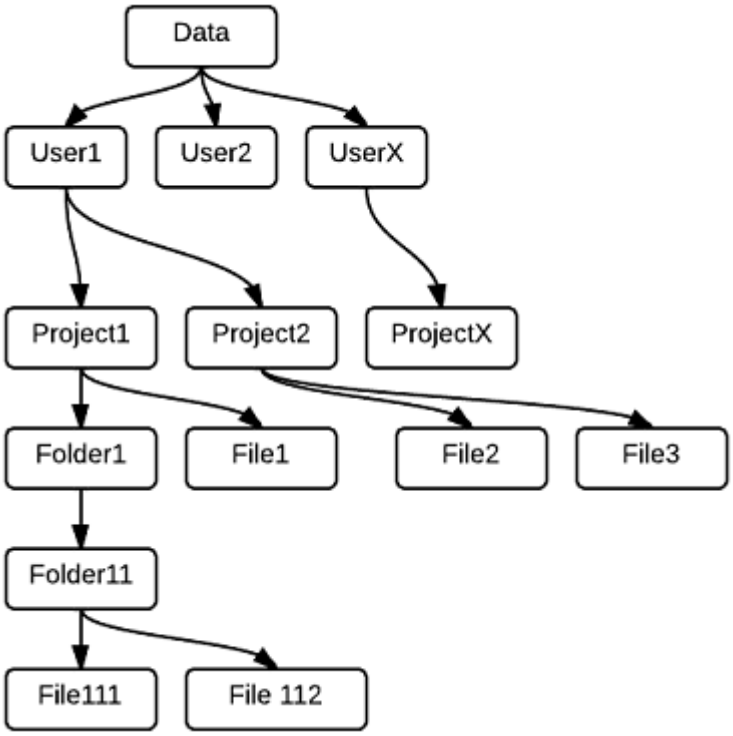

Figura 5.6: Estrutura de diretórios presente no servidor.

não são replicados para o seu diretório, simplesmente é-lhe dada permissão de acesso ao diretório do projeto em questão.

É importante ter em conta os caracteres não permitidos na criação de diretórios, para evitar erros e eventuais ataques ao sistema, esses caracteres são: `! # $ % & ' ( ) + , - . ; = @ [ ] { } "* / : < > ? |.`

A interface de navegação na estrutura de diretórios de um projeto encontra-se descrita na Secção 5.3.5.

5.3.3.2 *Uploader* de Ficheiros

Segundo o requisito RF05, especificado na Tabela 4.1, deve ser permitido o *upload* de múltiplos ficheiros no sistema e deve ser permitido ao utilizador iniciar outro, sem que os anteriores tenham terminado.

O diagrama de sequência da Figura 5.7 ilustra as interações e operações aquando do carregamento de *n* ficheiros.

Tradicionalmente, o carregamento de ficheiros na web tem uma má usabilidade. Inicialmente, os carregamentos começaram por ser parte integrante de um formulário, em que o utilizador não recebia qualquer feedback por parte da aplicação sobre o estado do upload. No decorrer dos últimos anos, têm surgido várias alternativas para melhorar os mecanismos de carregamento, melhorando-se a sua usabilidade e sendo permitidos múltiplos upload.

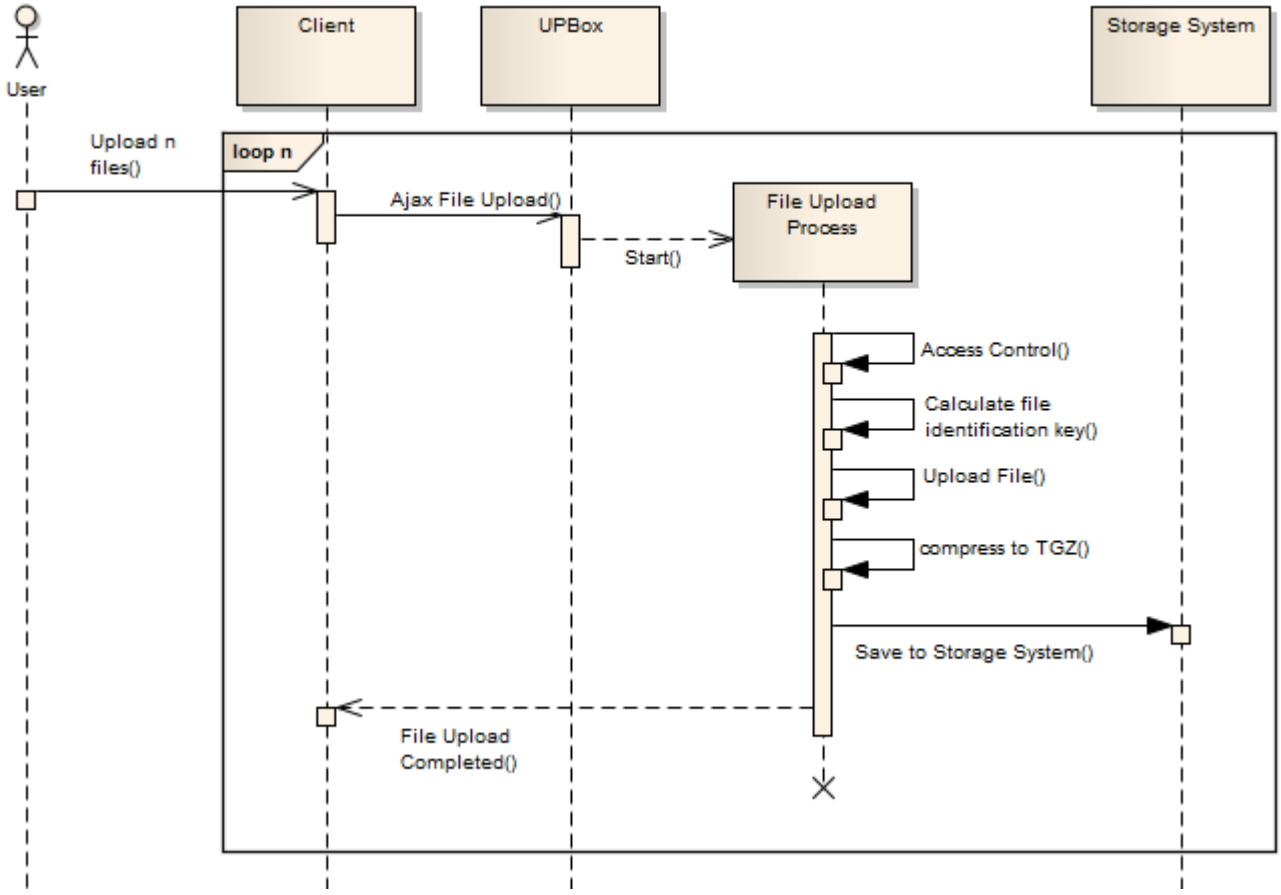

Figura 5.7: Diagrama de sequência do carregamento de *n* ficheiros.

Existem vários *plugins*, como é o caso do *Uploadify*[15] e *jQuery-File-Upload*[16], que implementam este mecanismo de múltiplos *uploads* por *AJAX* do lado do cliente, no entanto estes são pouco personalizáveis e de difícil integração. Assim sendo, decidiu-se implementar um gestor de carregamentos de raiz com o objetivo de oferecer um carregamento de ficheiros mais simples e personalizado.

Recentemente, com a introdução do HTML5, foi disponibilizado um tipo de objeto novo, denominado por *FormData*[17], que permite enviar dados de formulários através de JavaScript. Já existiam soluções idênticas a esta, no entanto não eram unificadas. Desta forma cada fornecedor de navegadores *web* implementava a sua solução a este problema.

Assim, a solução encontrada para o *upload* de múltiplos ficheiros foi a criação de um objeto do tipo *FormData* para cada ficheiro que é enviado por *JavaScript*, assincronamente para o servidor.

O envio assíncrono para o servidor foi feito com recurso à API XMLHttpRequest[18] (XHR)

[15] http://www.uploadify.com/
[16] https://github.com/blueimp/jQuery-File-Upload
[17] https://developer.mozilla.org/en-US/docs/DOM/XMLHttpRequest/FormData/Using_FormData_Objects
[18] http://www.w3.org/TR/XMLHttpRequest/

disponível para *JavaScript*. Esta API permite efetuar pedidos AJAX ao servidor e receber a resposta diretamente no *script*. O XHR disponibiliza métodos que permitem envio de objetos do tipo *FormData* e controlar o seu envio através de *handlers*.

Desta forma, o servidor recebe ficheiros individualmente como se fossem submetidos através de um formulário normal, sendo que o cliente é responsável por enviar os ficheiros individualmente através de pedidos XHR e controlar o seu início, progresso e fim recorrendo a *handlers*.

O servidor, quando recebe um pedido de *upload*, é responsável por verificar se o utilizador tem acesso ao projeto em questão, por calcular o identificador único do ficheiro, por receber o ficheiro, comprimi-lo e guardá-lo no sistema de armazenamento. Este processo pode ser observado na Figura 5.7, onde se verifica que o servidor é capaz de receber vários pedidos de *upload* simultaneamente.

Cálculo do Identificador do Ficheiro

Cada ficheiro terá associado um identificador no sistema que deverá ser único. Este identificador será utilizado para efetuar um pedido de *download* e, aquando de um carregamento, alterar o nome do ficheiro em questão para o seu identificador único.

Por motivos de segurança o identificador não poderá ser sequencial, porque seria trivial efetuar pedidos de *download* de ficheiros aos quais não se tenha acesso, apesar do sistema verificar sempre se o utilizador tem acesso ao ficheiro em questão.

Assim, uma boa solução para criação de um identificador único para o sistema foi a utilização de um algoritmo criptográfico. Desta forma, o resultado desta encriptação pode ser utilizado, não só para identificar o ficheiro, mas também para verificar a consistência dos ficheiros recebidos.

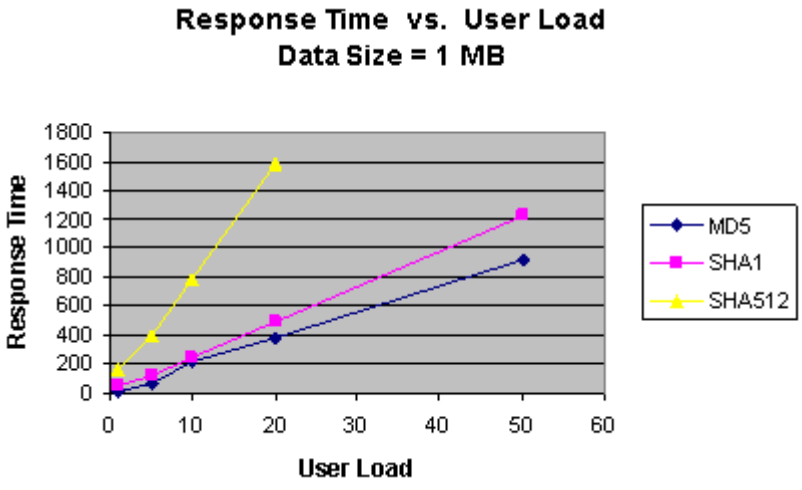

Figura 5.8: Comparativo de tempos de resposta de alguns algoritmos criptográficos[Dha12].

O algoritmo escolhido para este efeito foi o MD5, por um lado, porque apresenta um tamanho de 32 caracteres, por outro, porque possui um melhor desempenho relativamente a outros algoritmos criptográficos. A Figura 5.8 mostra um comparativo de performance efetuado a alguns

algoritmos criptográficos.

O algoritmo MD5 não garante unicidade, de acordo com o paradoxo do aniversário o número de ficheiros necessário para que haja 50% de probabilidade de colisão é 2^{64} [Tho05]. Uma solução comum para este problema é adicionar um conjunto de caracteres aleatórios à entrada do algoritmo. Neste caso, o conjunto é representado pela data atual em milissegundos. Assim, sempre que haja uma colisão no identificador de algum ficheiro é efetuado um novo cálculo do identificador e assim por diante até que não haja nenhuma colisão.

Carregamento e Compressão de ficheiros

Devido ao mecanismo de carregamento de ficheiros implementado do lado do cliente, o servidor recebe pedidos de carregamento de um único ficheiro. O carregamento é efetuado por *multipart*, de forma a receber ficheiros através do formulário do cliente.

Após o upload, o ficheiro é comprimido e, posteriormente, armazenado no sistema de armazenamento, alterando o seu nome para o identificador único previamente calculado.

Tipicamente, de forma a rentabilizar o espaço de armazenamento, os sistemas de armazenamento na nuvem comprimem os ficheiros armazenados com algoritmos pouco custosos em termos de processamento [CL11].

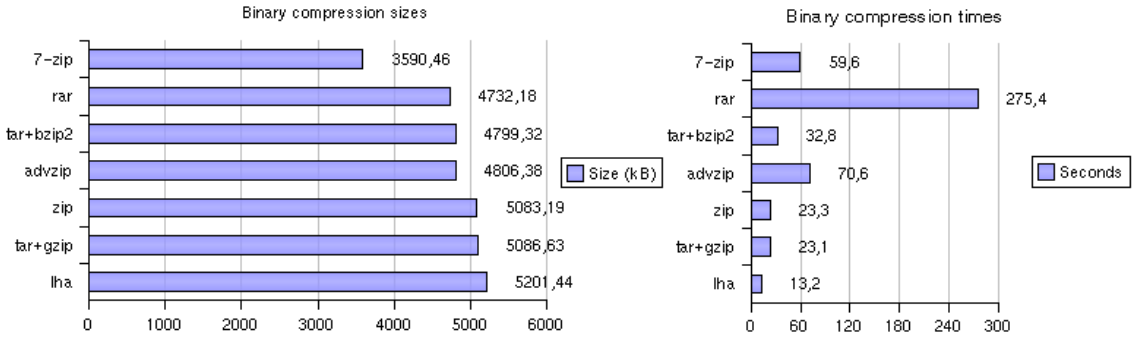

Figura 5.9: Comparativo de métodos de compressão de ficheiros em tamanho final e tempo de compressão[19].

Assim, o algoritmo escolhido para compressão de ficheiros foi o *tar+gzip*, por apresentar um bom rácio compressão/tempo e por ser suportado nativamente em *Ruby on Rails*. A Figura 5.9 ilustra uma análise a vários métodos de compressão de ficheiros, em termos de tamanho final e tempo de compressão.

Quando o carregamento de um ficheiro termina, o servidor envia uma resposta ao cliente com os dados do ficheiro armazenado e assim o cliente pode atualizar a interface apresentada ao utilizador.

[19] Imagem retirada de: http://warp.povusers.org/ArchiverComparison/

5.3.3.3 Descarga de ficheiros

O diagrama de sequência do download de um ficheiro encontra-se representado na Figura 5.10. Os pedidos de download ao servidor devem incluir o identificador único do ficheiro em questão, um exemplo do URL de um destes pedidos é: `/download/a19fb9b8caa73c702a48c4562411ebcf`.

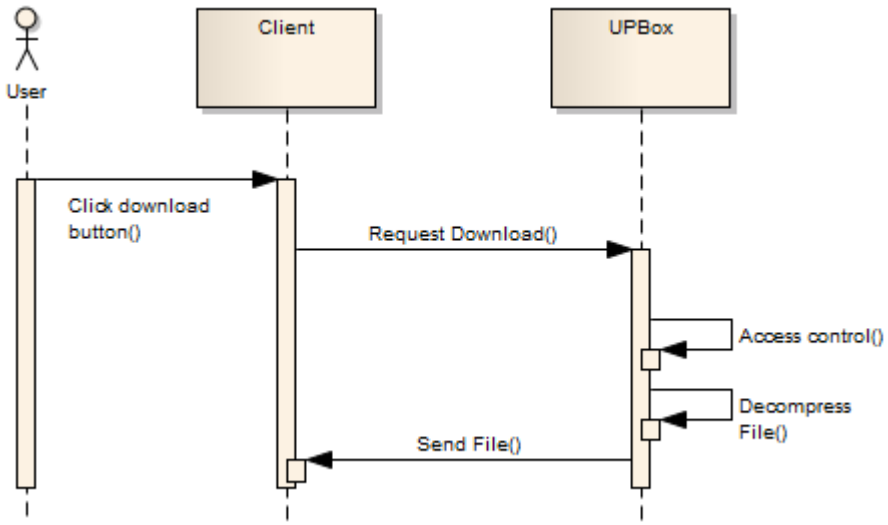

Figura 5.10: Diagrama de sequência do *download* de um ficheiro.

Os ficheiros estão armazenados num local privado e é impossível aceder-lhes diretamente, sendo a única forma de os obter através do pedido de *download*.

Sempre que o servidor recebe um pedido, é verificado se o utilizador tem acesso ao projeto ao qual o ficheiro pertence. Em caso afirmativo o ficheiro em questão é lido, descomprimido e enviado ao utilizador com o seu nome original.

5.3.4 Anotação de ficheiros e diretórios

A UPBox recorre ao sistema de anotação, *DataNotes*, para permitir a anotação de ficheiros. A comunicação entre estes sistemas é apresentada no diagrama de sequência da Figura 5.11.

A solução desenvolvida abstrai-se de todos os conceitos associados à anotação de dados. Quando um utilizador pretende anotar um diretório ou ficheiro, o sistema gera uma ligação ao *DataNotes* que o encaminha para o local de anotação do ficheiro em questão.

O *DataNotes* [Gou13] é uma plataforma colaborativa onde os investigadores podem anotar pastas e ficheiros relativos a um projeto, com base em vocabulários multidisciplinares previamente introduzidos na plataforma por um curador.

Sempre que o investigador termina a anotação de um ficheiro ou diretório, o *DataNotes* envia um *backup* da anotação através do operação */backup_annotation* (OP09), definida na Tabela 5.1.

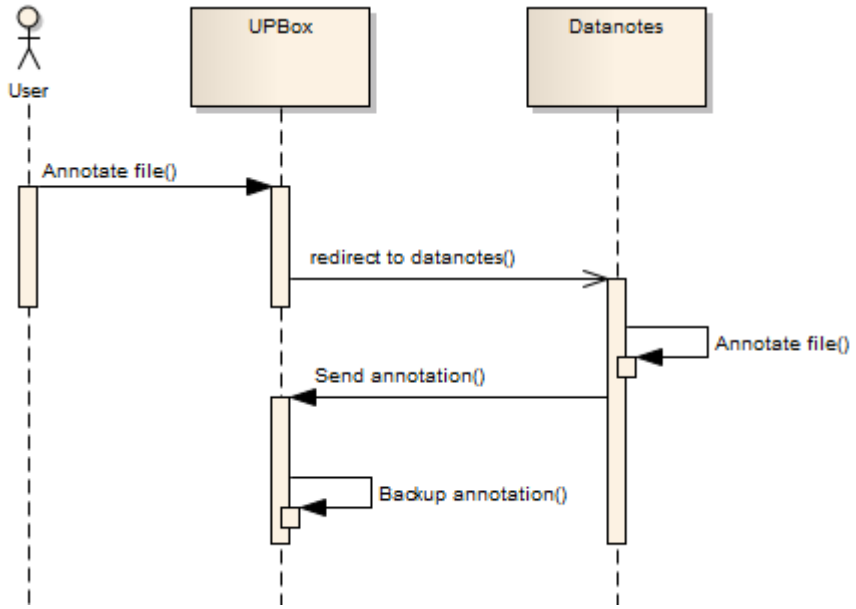

Figura 5.11: Diagrama de sequência da comunicação entre a UPBox e o DataNotes.

A anotação enviada tem o formato RDF e é armazenada numa estrutura paralela à estrutura de diretórios descrita no Secção 5.6.

As anotações são armazenadas no sistema para, futuramente, ser criado um pacote com a estrutura de dados de um projeto e respetivas anotações, a ser enviado para o repositório aberto da UP. Este processo de submissão de pacotes está fora do âmbito desta dissertação.

Existem outros serviços fornecidos ao *DataNotes*, o OP02 e OP08, que fornecem os projetos de um utilizador e a respetiva estrutura de diretórios e ficheiros. Desta forma, é permitido ao investigador criar e editar anotações relativas a ficheiros e diretórios dos seus projetos.

5.3.5 Considerações de *Design*

Esta secção é dedicada à descrição das escolhas efetuadas para a interface do protótipo desenvolvido, tendo em conta as componentes de usabilidade sugeridas por Jakob Nielsen descritas na Secção 4.2.2.

Antes de desenhar a interface do protótipo a desenvolver foi necessário fazer um levantamento de aplicações de armazenamento na nuvem familiares aos investigadores. Foi feita uma análise às seguintes aplicações: *Dropbox*, *Ubuntu One* e *Google Drive*. Esta análise serviu para efetuar um levantamento de alguns padrões e abordagens de *design* e adotá-los no desenvolvimento da UPBox. Desta forma espera-se uma curva de aprendizagem suave na utilização do sistema e, posteriormente, uma maior eficiência na realização de tarefas.

A interface que permite a navegação na estrutura de diretórios de um projeto é a página principal da UPBox. Esta encontra-se representada na Figura 5.12. A ideia consistiu em fornecer um

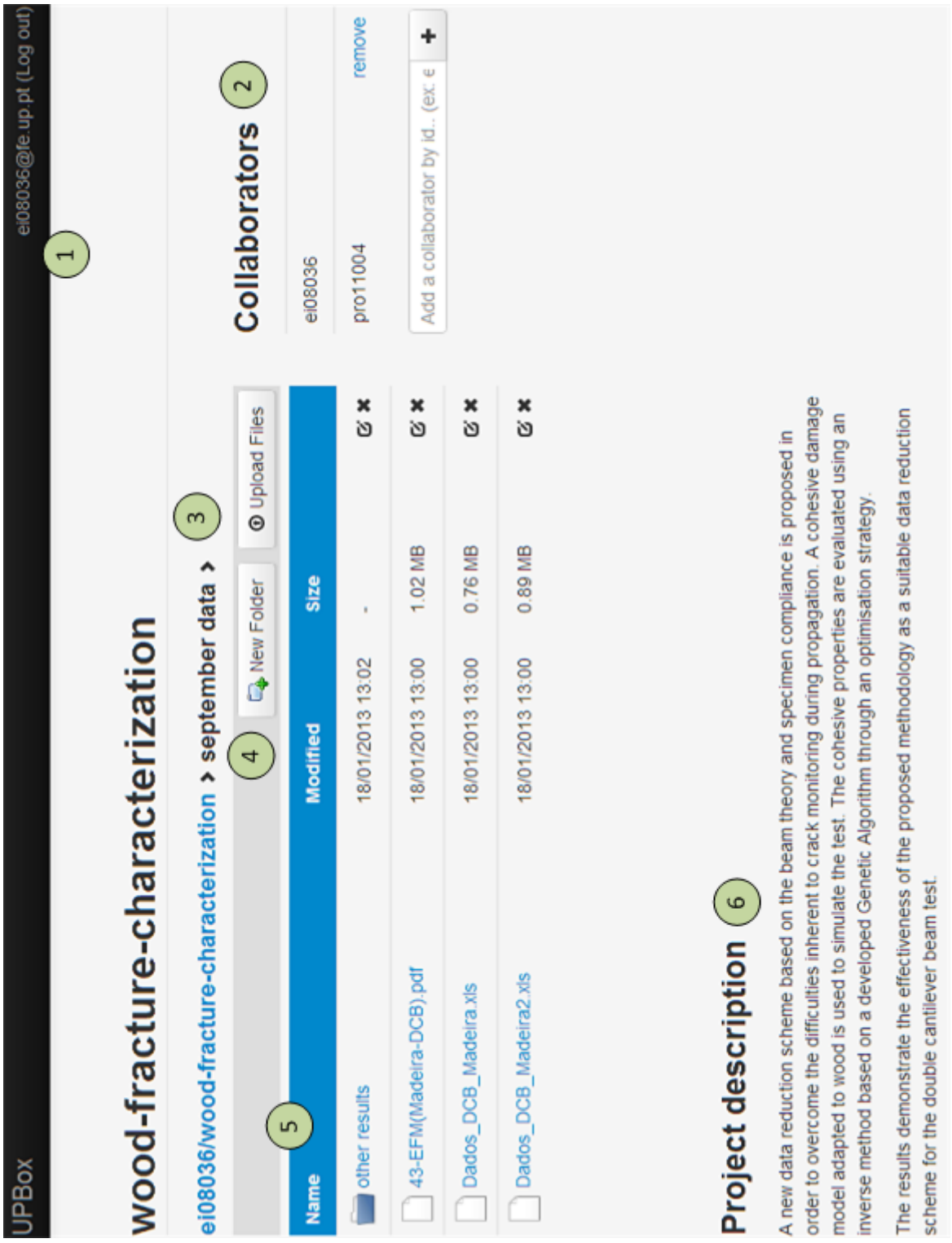

Figura 5.12: Interface principal do sistema — visualização de um diretório de um projeto.

ambiente idêntico a um navegador de ficheiros de um sistema operativo convencional, onde são oferecidas operações básicas sobre os ficheiros.

Na Figura 5.12 - ponto 1 - é apresentado o email do utilizador e um link para efetuar *logout*. Os pontos 2 e 3 são relativos ao projeto em questão, sendo que o primeiro é o gestor de colaboradores, onde é permitido partilhar o projeto com outros utilizadores ou, por outro lado, remover utilizadores; por sua vez, o segundo apresenta a descrição do projeto.

Os pontos 3, 4 e 5 são relativos ao ambiente de navegação em estruturas de ficheiros. O ponto 3 representa a barra de navegação, onde é representado o diretório atual e todos os diretórios acima desse. Tipicamente, os sistemas de ficheiros fornecem ligações aos seus diretórios pai, de forma a facilitar a navegação no sistema.

A rota para a visualização de um diretório foi implementada de forma a tornar os *links* navegáveis, como um sistema de ficheiros tradicional. Por exemplo, o *link* para aceder ao diretório *september data/other results/* do projeto *wood-fracture-characterization* criado pelo utilizador *ei08036* é */home/ei08036/wood-fracture-characterization/september data/other results/*.

O ponto 4 é referente à barra de escrita no sistema. O botão *New Folder* abre um formulário para criação de um diretório, por outro lado o botão *Upload Files* abre o carregador de ficheiros, representado na Figura 5.13, que será descrito com maior detalhe posteriormente. A adição de ficheiros e diretórios é efetuada sem abrir uma nova página, sendo que os ficheiros são automaticamente adicionados à estrutura de dados e é dado *feedback* ao utilizador desta ação.

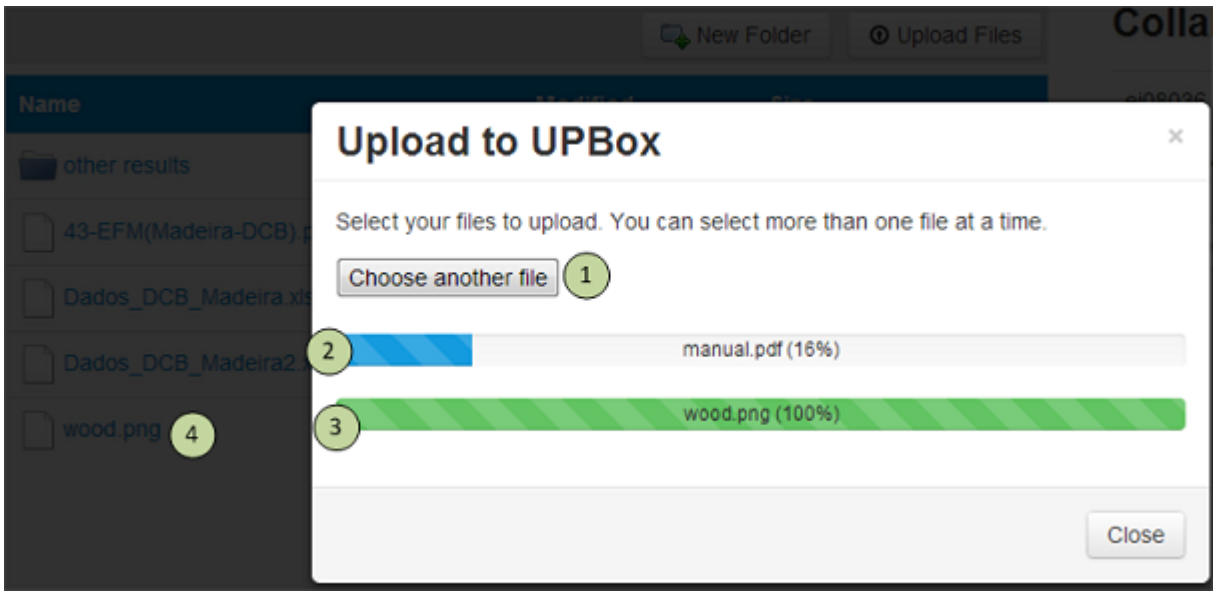

Figura 5.13: Carregador de ficheiros do sistema.

A lista de diretórios e ficheiros do diretório atual encontra-se representado no ponto 5. Nesta lista são dados detalhes do seu nome, data de modificação ou criação e tamanho do ficheiro. São ainda fornecidos ícones que permitem eliminar e anotar o elemento, sendo este último uma ligação para o sistema de anotação *DataNotes*. Os nomes dos elementos encontram-se ligados, sendo que no caso dos ficheiros será para efetuar o seu *download* e no caso dos diretórios, será para explorar os seus ficheiros e diretórios.

O interface do carregador de ficheiros desenvolvido encontra-se ilustrado na Figura 5.13. Tipicamente, os carregamentos de ficheiros são efetuados em formulários compostos por uma caixa de texto, o selecionador do ficheiro e o botão para a sua submissão. No sistema desenvolvido o *upload* de ficheiros foi simplificado e é efetuado com recurso a um único botão, representado pelo ponto 1 da Figura 5.13. Quando esse botão é clicado, é aberto o explorador de ficheiros do navegador que, após a seleção de ficheiros, os carrega automaticamente.

Os pontos 2 e 3 são referentes à lista de ficheiros selecionados para carregamento, e representam o seu progresso e o nome original do ficheiro. Dado o término do carregamento o ficheiro é automaticamente adicionado à lista de ficheiros. No exemplo da Figura 5.13 isso é visível: o ficheiro do ponto 3, *wood.png*, foi carregado e adicionado à lista de ficheiros do diretório (ponto 4).

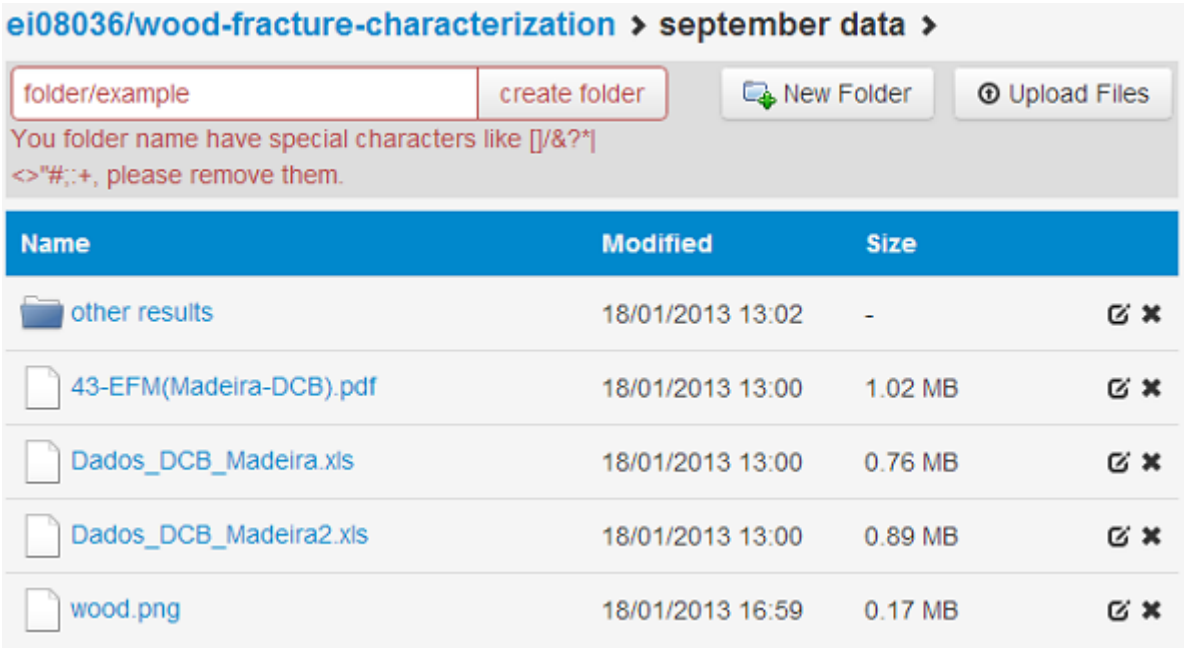

Figura 5.14: Deteção de erro na criação de um diretório e respetiva sugestão de resolução.

É importante referir que o sistema tenta, sempre que possível, detetar erros na sua utilização e sugerir resoluções a esse problema. Um exemplo disso encontra-se representado na Figura 5.14, em que o utilizador tenta criar um diretório com um caráter não permitido, sendo-lhe apresentada uma mensagem de erro e respetiva sugestão de resolução.

5.3.6 Desenvolvimento e teste da API

A API descrita na Secção 5.1.4 foi implementada com recurso à arquitetura *model-view-controller* fornecida pelo *Ruby on Rails*. Com toda a lógica do negócio já implementada nos modelos, mostrou-se necessário apenas de preparar os controladores para receberem pedidos REST

nos formatos JSON e XML. Seguidamente, é necessário responder aos pedidos no formato pedido pelo cliente, processando a resposta a enviar, com recurso às bibliotecas de JSON e XML fornecidas pelo *Ruby*.

Adicionalmente, foi necessário criar um mecanismo de autenticidade dos pedidos. Sempre que é efetuado *login* com sucesso o servidor envia como resposta um *cookie*. Este *cookie* é um conjunto de caracteres aleatórios calculado pelo servidor, que fica associado àquela autenticação do utilizador. Desta forma, sempre que o cliente efetuar um pedido com o *cookie* incluído, o servidor sabe que o utilizador é autêntico pois possui a *cookie* partilhada como prova.

A especificação detalhada da API pode ser consultada no Anexo A, onde as estruturas dos pedidos e respetivas respostas são descritas e exemplificadas.

Foi implementada uma pequena aplicação na linguagem *Java* como exemplo de utilização da API. Esta aplicação testou todos os métodos fornecidos e servirá de exemplo a futuro desenvolvimento de extensões à UPBox.

5.3.7 Instalação

O *deployment* de *software* consiste num conjunto de atividades que tornam o sistema disponível para utilização. Em *Ruby on Rails* esta tarefa poderá ser difícil de realizar, visto não existir nenhum ambiente de produção automatizado integrado na *framework*. Desta forma, todas as tarefas têm que ser realizadas manualmente.

O sistema foi instalado numa máquina virtual baseada em *Debian*. Em primeiro lugar, foi necessário instalar todo o software de suporte à aplicação: o *Ruby*, o *Rails* e o *PostgreSQL*. Seguidamente, testou-se a aplicação *web* em ambiente de desenvolvimento no servidor, de forma a verificar se o software instalado funcionava devidamente.

Para migrar a aplicação para ambiente de produção foi necessário instalar o *nginx*, um servidor *web open-source* que suporta *Ruby on Rails*. Todos os passos e configurações necessárias para colocar a aplicação em ambiente de produção encontram-se descritas detalhadamente no Anexo D.

Após a migração para ambiente de produção, surgiu um problema: o carregamento de ficheiros de grande dimensão congelava e não terminava. Inicialmente, o problema foi resolvido alterando o tamanho máximo permitido nos pedidos ao servidor no ficheiro de configuração do *nginx*. No entanto, o problema persistiu para ficheiros de grande dimensão.

O que acontece aquando do carregamento de um ficheiro é que este é carregado para o diretório de ficheiros temporários controlado pelo sistema operativo, o */tmp*, posteriormente é comprimido e copiado para o seu diretório original. O problema foi que o diretório */tmp* estava limitado em termos de espaço de armazenamento e em ficheiros de grande dimensão este era ocupado rapidamente. Assim, para ultrapassar este obstáculo, foi aumentado o espaço alocado ao diretório de ficheiros temporários.

Tipicamente este problema acontece com maior frequência em distribuições UNIX que utilizem o *tmpfs* como gestor do diretório de ficheiros temporários, pois este limita, por omissão, o seu tamanho.

Capítulo 6

Avaliação

Está prevista a avaliação de todo o processo de depósito de dados no repositório de dados da UP num projeto posterior a este desenvolvimento [CR12]. Este incluirá testes à gestão de dados na *UPBox*, por parte de investigadores, à anotação de dados no *DataNotes* e respetiva gestão de anotações, ao processo de curadoria de dados, por parte de curadores no repositório de dados da UP e à integração de todas estas plataformas.

Apesar de estarem previstos testes à UPBox, realizaram-se testes de usabilidade preliminares ao protótipo desenvolvido. Estes tinham em vista à sua validação, recebendo, assim, *feedback* dos utilizadores e permitindo melhorar alguns aspetos do protótipo para a fase posterior de testes.

6.1 Planeamento do Teste de Usabilidade

O teste foi direcionado a investigadores da UP na área das ciências da informação e de informática. Por motivos de agenda os testes não foram efetuados aos investigadores que colaboraram com testes ao repositório experimental da UP.

O teste de usabilidade implementado, disponível no Anexo C, contém as seguintes tarefas:

1. Autenticação com credenciais da UP.
2. Criação de projetos.
3. Partilha de projeto com outro utilizador.
4. Adicionar diretórios e ficheiros ao projeto.
5. Efetuar *download* de um ficheiro do projeto.
6. Remover diretórios ou ficheiros do projeto.

O teste inclui a tarefa de anotação de um ficheiro ou diretório que não será avaliada neste teste, por ser do âmbito do *DataNotes*.

Foram utilizadas três métricas, sugeridas por Jeff Sauro [Sau10], para avaliar a usabilidade do protótipo desenvolvido: o tempo de execução das tarefas, o número de erros encontrado e o número de cliques efetuado, sendo que por vezes poderá ser relevante o percurso para efetuar a tarefa.

No final do teste foi feito um questionário informal ao investigador com vista a avaliar os seguintes aspetos:

1. Detetar problemas de usabilidade.
2. Recolher impressões sobre a facilidade de uso.
3. Avaliar as expectativas do investigador, com possibilidade de sugestão de novas funcionalidades.

O investigador deve ser colocado à vontade, sendo-lhe dada a oportunidade de explorar o *web site* antes de efetuar o teste de usabilidade. Optou-se por não intervir no teste em momento algum, dizendo-lhe para utilizar a aplicação como se estivesse sozinho. Todas as reações relevantes durante a utilização foram anotadas para serem discutidas no final do teste.

Os testes foram efetuados a 5 indivíduos como sugerido por Jakob Nielsen.

6.2 Resultados

Os resultados dos testes de usabilidade podem ser visualizados nas Figuras 6.1, 6.2 e 6.3.

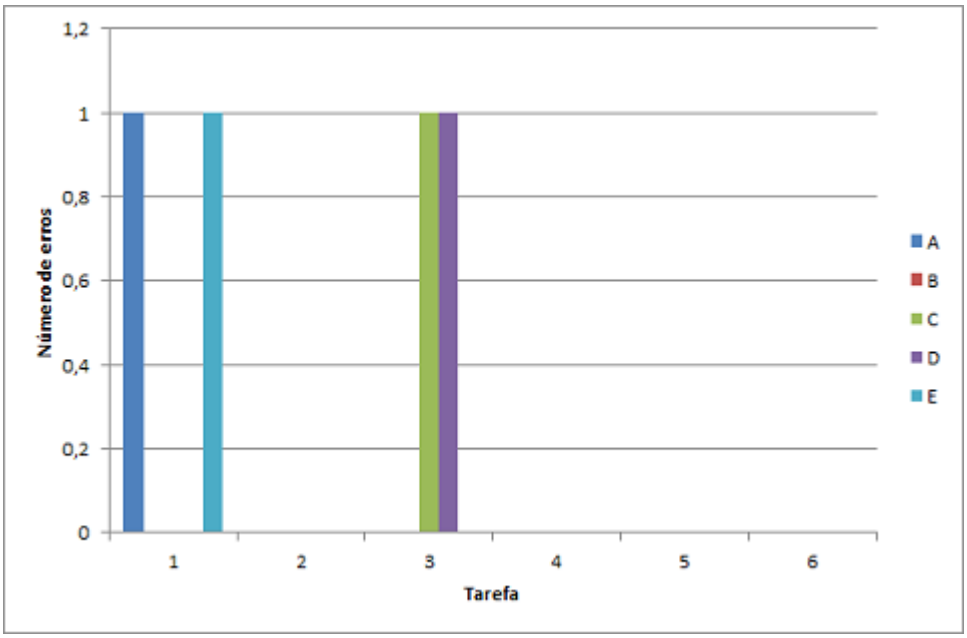

Figura 6.1: Número de erros cometidos pelos investigadores nas tarefas propostas.

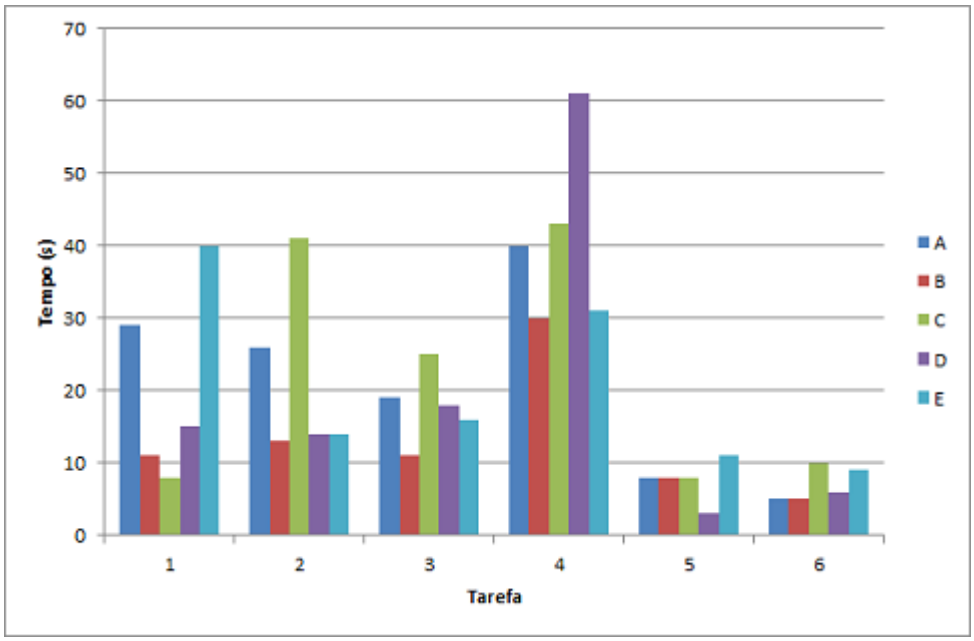

Figura 6.2: Tempo de execução das tarefas propostas.

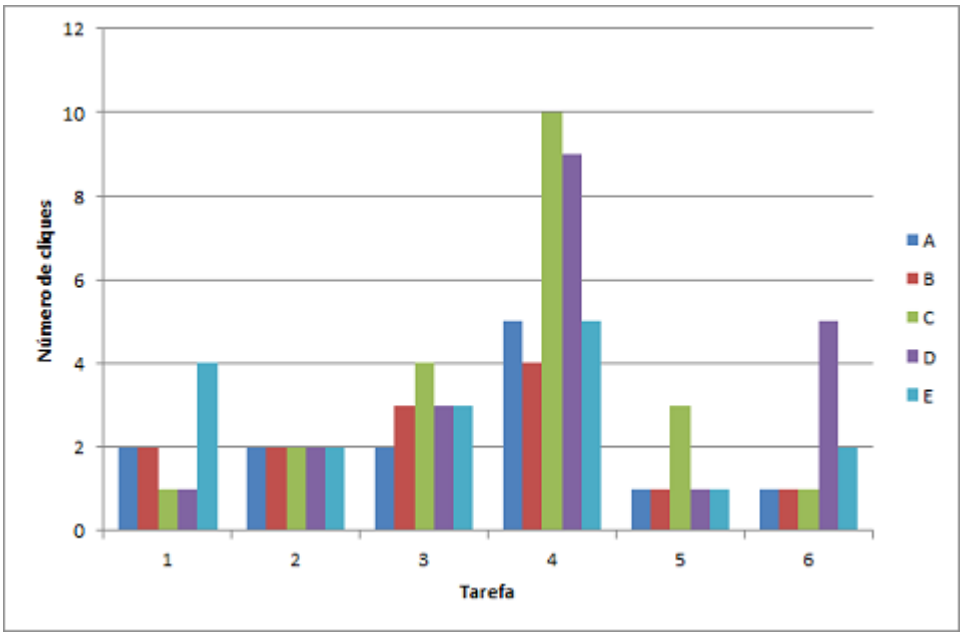

Figura 6.3: Número de cliques ao executar as tarefas propostas.

Relativamente aos questionários informais, foram levantadas várias opiniões acerca do protótipo desenvolvido. Três investigadores consideraram que a adição de utilizadores ao projeto poderia ser melhorada. Foi sugerida a possibilidade de adição de investigadores não registados no sistema e, por outro lado, permitir adicioná-los através do seu nome. Outro investigador referiu a possibilidade de acrescentar um ícone adicional para efetuar a descarga de um ficheiro, mantendo

o *link* original para esse efeito no nome do ficheiro. O último não apontou nenhuma melhoria na usabilidade do protótipo.

Todos os investigadores consideraram o protótipo bastante usável, sendo a realização de tarefas efetuada de forma simples e rápida. Um dos investigadores referiu ainda que o protótipo segue as normas de *design* de *web sites* idênticos.

Relativamente à última questão, acerca das expectativas de funcionalidades no protótipo, foi sugerido por três investigadores o desenvolvimento de uma aplicação cliente extensível à UPBox. Outro investigador queria ver implementada a funcionalidade de permitir efetuar o carregamento de uma estrutura de dados, através de um passo simples. O último sugeriu a implementação da pré-visualização dos ficheiros carregados, de modo a permitir ver o seu conteúdo sem efetuar o *download* do mesmo.

Analisando o gráfico da Figura 6.1 verifica-se que foram cometidos poucos erros na interação com o protótipo. Foram detetados erros na primeira tarefa, relativa ao *login*, por erro na inserção das credenciais de acesso. Foram, igualmente, cometidos erros ao adicionar um investigador ao projeto, devido à interface ser algo confusa, como alguns investigadores referiram no questionário.

Relativamente ao tempo de execução das tarefas, ilustrado no gráfico da Figura 6.2, verificou-se que todas se realizaram em tempos bastante satisfatórios, concluindo que o armazenamento de dados é pouco custoso em termos temporais. O tempo médio da ação para começar o carregamento de dados no repositório é de, aproximadamente, 40 segundos.

Relativamente ao gráfico da Figura 6.3, verifica-se que as tarefas foram executadas recorrendo a poucos cliques. Este facto já era esperado, devido ao baixo tempo na realização das mesmas. A única tarefa que exigiu mais cliques foi a que sugeria a adição de ficheiros e diretórios ao projeto, em que foram criadas entre 0 a 2 pastas por investigador e efetuados entre 3 a 7 carregamentos de ficheiros.

6.3 Conclusões

Os gráficos anteriormente descritos e o questionário informal permitiram avaliar melhorias a algumas funcionalidades do protótipo, bem como à definição de novas funcionalidades a implementar em futuras fases de desenvolvimento da UPBox. Em primeiro lugar, é esperado pelos investigadores a expansão do protótipo a uma aplicação cliente, que permita sincronizar os ficheiros do servidor com o computador pessoal. Esta aplicação já está planeada desde o início do projeto e o protótipo foi preparado para permitir a sua expansão futuramente.

Outra funcionalidade a ser implementada será o carregamento de diretórios, permitindo a criação de estruturas de dados de forma mais rápida. Por outro lado, também poderá ser implementada a funcionalidade de descarregar diretórios, que por motivos tecnológicos terão que ser previamente comprimidas formando um único ficheiro.

Como melhorias, será essencial rever a adição de investigadores aos projetos. Para isso, poderão ser adicionadas dicas de utilização dessa funcionalidade. O sistema também poderia permitir adicionar investigadores não registados na aplicação, sendo-lhes enviando um email a notificar do

projeto ao qual foram adicionados. Como sugerido por um investigador, poderá ser integrado na funcionalidade de preenchimento automático o nome do investigador, para além do seu identificador do SiFEUP.

Por motivos de agenda não foi possível testar a integração da *UPBox* com o *DataNotes*, no entanto esses testes estão planeados e, até lá, serão corrigidas algumas falhas do protótipo detetadas nesta fase de testes.

Em suma, o protótipo desenvolvido foi bem aceite pelos investigadores devido à sua facilidade de uso e às funcionalidades disponibilizadas. Esperam-se futuros desenvolvimentos que visem a melhoria de algumas funcionalidades, assim como a implementação de outras que melhorem a interação com o utilizador.

Capítulo 7

Conclusões e Trabalho Futuro

A partilha de dados em repositórios institucionais tem ganho relevância nos últimos anos, na medida em que o acesso a dados é essencial no processo de novas descobertas científicas.

Várias organizações têm vindo a investir na investigação de abordagens à curadoria e no desenvolvimento de sofisticados repositórios de dados de investigação. Também têm sido tomadas várias medidas que incentivam, ou até obrigam, à sua partilha. Nesse sentido, está a ser testado, atualmente, um protótipo do repositório de dados experimental na UP que albergará dados de investigação dos seus investigadores.

Nesta dissertação desenvolveu-se um protótipo cujo objetivo é suportar a gestão e partilha de dados durante o processo de investigação e permitir a sua anotação. Futuramente espera-se integrar este protótipo com o repositório da UP, de forma a permitir a submissão desses dados e anotações no mesmo de forma simples e transparente.

O protótipo desenvolvido, UPBox, consiste numa aplicação *web* que permite o depósito e partilha de dados de investigação na nuvem. A UPBox integra com um sistema de anotação, o DataNotes, de modo a permitir anotar ficheiros com recurso a vocabulários multidisciplinares.

Foi desenvolvido um *web service* que disponibiliza métodos a serem utilizados por futuras aplicações cliente que estendam a UPBox.

Estão planeados testes à UPBox e a outros sistemas, fora do âmbito desta dissertação, que irão avaliar todo o fluxo de trabalho da inserção de dados no repositório da UP. Mesmo assim, foram realizados testes de usabilidade e questionários para detetar falhas no protótipo desenvolvido e avaliar a sua usabilidade.

Os objetivos propostos para este trabalho foram conseguidos. Os investigadores alvo dos testes de usabilidade concordaram, relativamente à facilidade de uso do sistema e ainda referiram que este satisfaz as suas necessidades de gestão colaborativa de dados de investigação. É importante referir que o tempo médio para iniciar o carregamento de ficheiros no sistema é de 40 segundos, o que se considera bastante satisfatório.

Futuramente, com a integração da UPBox com o repositório de dados da UP, espera-se que os investigadores se sintam motivados a submeter os seus dados de investigação no repositório da UP com a utilização deste serviço, visto este ter simplificado todo processo de submissão.

A seguinte secção descreve as sugestões de trabalho futuro, onde se verifica que ainda existe um longo caminho a percorrer para provar este conceito. Também são enumeradas algumas sugestões de melhoria da aplicação baseadas nos testes de usabilidade realizados.

7.1 Trabalho Futuro

Esta dissertação pode ser considerada a primeira abordagem a este estudo, pelo que existem vários trabalhos relacionados a serem desenvolvidos e, também, algumas melhorias a considerar no protótipo desenvolvido.

O fluxo de trabalho para o depósito de dados no repositório da UP segundo esta abordagem não está completamente desenvolvido. Atualmente é permitido ao investigador gerir os seus dados de investigação, com recurso à UPBox, e anotá-los, recorrendo ao DataNotes, como pode ser verificado na Figura 4.1. A curadoria de dados é efetuada posteriormente no repositório de dados, baseado em DSpace, pelo curador. Para terminar este processo de submissão, é necessário submeter os dados de investigação e respetivas anotações como disponíveis para curadoria no repositório de dados. Atualmente esta tarefa pode ser efetuada manualmente, por um curador, no entanto é pertinente automatizá-la pois trata-se de um processo bastante complexo e moroso.

Assim, abre-se aqui a possibilidade da automatização do processo de disponibilização de dados e anotações para curadoria. O DSpace disponibiliza uma API que permite a submissão para curadoria de um pacote composto por dados de investigação e anotações. Desta forma os dados poderão ser curados através das ferramentas de curadoria disponibilizadas pelo DSpace, com os quais os curadores estão familiarizados. O protótipo está preparado para a inclusão desta funcionalidade, visto albergar os dados de investigação e respetivas anotações que constituirão o pacote a ser submetido no repositório de dados da UP.

Os testes de usabilidade efetuados permitiram avaliar melhorias a algumas funcionalidades do protótipo, bem como à definição de novas funcionalidades a implementar em futuras fases de desenvolvimento.

Os serviços de armazenamento na nuvem, com os quais os investigadores estão familiarizados, fornecem aplicações cliente para vários sistemas operativos que estendem as funcionalidades do *web site*. Estas aplicações cliente têm bastante sucesso atualmente, pois permitem a sincronização transparente do armazenamento local com a nuvem.

Desta forma, esperam-se futuros desenvolvimentos de aplicações cliente que estendam funcionalidades do *web site*. Foi disponibilizado um *web service* que disponibiliza vários serviços direcionados a este tipo de aplicações.

Os testes de usabilidade permitiram concluir que a interação com gestor de colaboradores de um projeto pode tornar-se confusa e pouco usável. Como melhoria, sugere-se a adição investigadores ao projeto que não estejam registados no sistema, sendo-lhes enviado um email como

notificação. A interface do gestor de tarefas poderá também ser revista, adicionando o nome do utilizador ao preenchimento automático.

Foram sugeridas, por investigadores, funcionalidades extra a implementar em futuras fases de desenvolvimento aquando dos testes efetuados. Estas incluem: carregamento de diretórios, *download* de diretórios e registo de operações efetuadas.

Por fim, será necessário efetuar testes a todo o fluxo de trabalho para a inserção de dados no repositório da UP. Espera-se, com estes testes, avaliar a integração da UPBox com o DataNotes e o repositório da UP, bem como reavaliar a usabilidade do protótipo desenvolvido.

Referências

[BGAT05] Peter Burnhill, David Giaretta, Malcolm Atkinson e H A Tii. The Digital Curation Centre: A Vision for Digital Curation. (Dcc):31–41, 2005.

[Bun04] P Bunenan. The two cultures of digital curation. *Proceedings. 16th International Conference on Scientific and Statistical Database Management, 2004.*, pages 7–7, 2004. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1311188, doi:10.1109/SSDM.2004.1311188.

[Cen12] The Digital Curation Centre. Digital Curation Centre. Disponível em http://www.dcc.ac.uk/digital-curation/what-digital-curation, acedido em 22 de Junho de 2012, 2012.

[CL11] C Constantinescu e Maohua Lu. Quick Estimation of Data Compression and De-duplication for Large Storage Systems. In *Data Compression, Communications and Processing (CCP), 2011 First International Conference on*, pages 98–102, 2011. doi:10.1109/CCP.2011.41.

[CR12] João Castro e Cristina Ribeiro. *Estudo de utilização do Repositório de dados científicos da Universidade do Porto*. PhD thesis, Universidade do Porto, 2012.

[Dat07] DataShare. DataShare Project. Disponível em http://www.disc-uk.org/datashare.html, acedido em 22 de Junho de 2012, 2007.

[Dha12] Priya Dhawan. Performance Comparison: Security Design Choices. Disponível em http://msdn.microsoft.com/en-us/library/ms978415.aspx#bdadotnetarch15_topic4, acedido em 22 de Junho de 2012, 2012.

[DN] F. Dridi e G. Neumann. How to implement Web-based groupware systems based on WebDAV. *Proceedings. IEEE 8th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'99)*, pages 114–119. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=805185, doi:10.1109/ENABL.1999.805185.

[DSp11] DSpace. Cloud Storage. Disponível em http://searchcloudstorage.techtarget.com/definition/cloud-storage, acedido em 22 de Junho de 2012, 2011.

[DSp12] DSpace. DSpace. Disponível em http://www.dspace.org/introducing, acedido em 22 de Junho de 2012, 2012.

[Eco10a] The Economist. Data, data everywhere. Disponível em http://www.economist.com/node/15557443, acedido em 22 de Junho de 2012, 2010.

[Eco10b] The Economist. The data deluge. Disponível em http://www.economist.com/node/15579717?story_id=15579717, acedido em 22 de Junho de 2012, 2010.

[Fer06] Miguel Ferreira. Introdução à Preservação Digital: Conceitos, estratégias e actuais consensos. 2006. URL: http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstract.

[Gou13] Mariana Gouveia. *DataNotes - Um sistema colaborativo para anotação de estruturas de diretórios*. PhD thesis, Faculdade de Engenharia da Universidade do Porto, 2013.

[Hey03] T Hey. The Data Deluge: An e-Science Perspective. *Grid computing*, (January 2003):1–17, 2003. URL: http://onlinelibrary.wiley.com/doi/10.1002/cbdv.200490137/abstracthttp://onlinelibrary.wiley.com/doi/10.1002/0470867167.ch36/summary.

[HZZS11] Guannan Hu, Wu Zhang, Wenhao Zhu e Shijun Shen. A dynamic user-integrated cloud computing architecture. *Proceedings of the 2011 International Conference on Innovative Computing and Cloud Computing - ICCC '11*, pages 36–40, 2011. URL: http://dl.acm.org/citation.cfm?doid=2071639.2071649, doi:10.1145/2071639.2071649.

[Lyo10] Liz Lyon. Data Dimensions: Disciplinary Differences in Research Data Sharing , Reuse and Long term Viability A comparative review based on sixteen case studies. (January), 2010.

[Mal07] Luiz Ernesto Pinheiro Malère. LDAP Linux HOWTO. Disponível em http://tldp.org/HOWTO/LDAP-HOWTO/index.html, acedido em 20 de Setembro de 2012, 2007.

[Moh12] Arif Mohamed. A history of cloud computing. Disponível em http://www.computerweekly.com/feature/A-history-of-cloud-computing, acedido em 22 de Junho de 2012, 2012.

[Mur12] David Murphy. Why is Dropbox Successful? It's the Simplicity. Disponível em http://www.pcmag.com/article2/0,2817,2400746,00.asp, acedido em 20 de Junho de 2012, 2012.

[Nie94] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann Series in Interactive Technologies. Morgan Kaufmann, 1994. URL: http://books.google.pt/books?id=95As2OF67f0C.

[oS11] University of Southampton. EPrints - The original institutional repository solution. Disponível em http://www.eprints.org/, acedido em 22 de Dezembro de 2012, 2011.

[RJKG10] Bhaskar Prasad Rimal, Admela Jukan, Dimitrios Katsaros e Yves Goeleven. Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach. *Journal of Grid Computing*, 9(1):3–26, December 2010. URL: http://www.springerlink.com/index/10.1007/s10723-010-9171-y, doi:10.1007/s10723-010-9171-y.

[Rom10] Jeroen Rombouts. Building a'data repository'for heterogenous technical research communities through collaborations. 2010. URL: http://docs.lib.purdue.edu/iatul2010/conf/day2/10/.

[RRC11] João Rocha, Cristina Ribeiro e João Correia Lopes. UPData: A Data Curation Experiment at U.Porto using DSpace. pages 224–227, 2011.

[RRC12a] João Rocha, Cristina Ribeiro e João Correia Lopes. Managing research data at U. Porto: requirements , technologies and services. *Innovations in XML Applications and Metadata Management: Advancing Technologies*, 2012. doi:10.4018/978-1-4666-2669-0.

[RRC12b] João Rocha, Cristina Ribeiro e João Correia Lopes. Managing research data at U. Porto: requirements , technologies and services. *Innovations in XML Applications and Metadata Management: Advancing Technologies*, 2012.

[RSR+10] Cristina Ribeiro, Ricardo Saraiva, Eloy Rodrigues, Matos Fernandes, Cristina Marques Gomes e José Carvalho. Os Repositórios de Dados Científicos: Estado da Arte. 2010.

[Sab11] Farzad Sabahi. Cloud computing security threats and responses. *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 245–249, May 2011. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6014715, doi:10.1109/ICCSN.2011.6014715.

[Sau10] J Sauro. *A Practical Guide to Measuring Usability: 72 Answers to the Most Common Questions About Quantifying the Usability of Websites and Software*. CreateSpace, 2010. URL: http://books.google.pt/books?id=OyoFTwEACAAJ.

[SCL11] Fu-quan Sun, Xu Cheng e Chao Liu. Research on Hadoop-based Enterprise File Cloud. *Awareness Science and Technology (iCAST), 2011 3rd International Conference on*, 2011. doi:10.1109/ICAwST.2011.6163114.

[SL10] Gail Steinhart e Albert R Mann Library. DataStaR : a data staging repository to support the sharing and publication of research data. 2010.

[SL12] Martin Stephenson e Giusy Di Lorenzo. Open Innovation Portal : a collaborative platform for open city data sharing. (March):522–524, 2012.

[Som06] Ian Sommerville. *Software Engineering*. Addison Wesley, 8 edition, 2006.

[Tho05] Eric Thompson. MD5 collisions and the impact on computer forensics. *Digital Investigation*, 2(1):36–40, February 2005. URL: http://linkinghub.elsevier.com/retrieve/pii/S1742287605000058, doi:10.1016/j.diin.2005.01.004.

[WGL+12] Xiaolong Wen, Genqiang Gu, Qingchun Li, Yun Gao e Xuejie Zhang. Comparison of open-source cloud management platforms: OpenStack and OpenNebula. *2012 9th International Conference on Fuzzy Systems and Knowledge Discovery*, (Fskd):2457–2461, May 2012. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6234218, doi:10.1109/FSKD.2012.6234218.

[WPG+10] Jiyi Wu, Lingdi Ping, Xiaoping Ge, Ya Wang e Jianqing Fu. Cloud Storage as the Infrastructure of Cloud Computing. *2010 International Conference on Intelligent Computing and Cognitive Informatics*, pages 380–383, June 2010. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5565955, doi:10.1109/ICICCI.2010.119.

[WVW12] Leslie Willcocks, Will Venters e Edgar A. Whitley. Meeting the challenges of cloud computing. Disponível em http://www.accenture.com/us-en/outlook/Pages/outlook-online-2011-challenges-cloud-computing.aspx, acedido em 22 de Junho de 2012, 2012.

Anexo A

Representação da Estrutura de Diretórios

A Figura A.1 mostra a estrutura em árvore de um diretório exemplo. A sua representação em JSON pode ser visualizada na Listagem A.1 e a em XML na Listagem A.2.

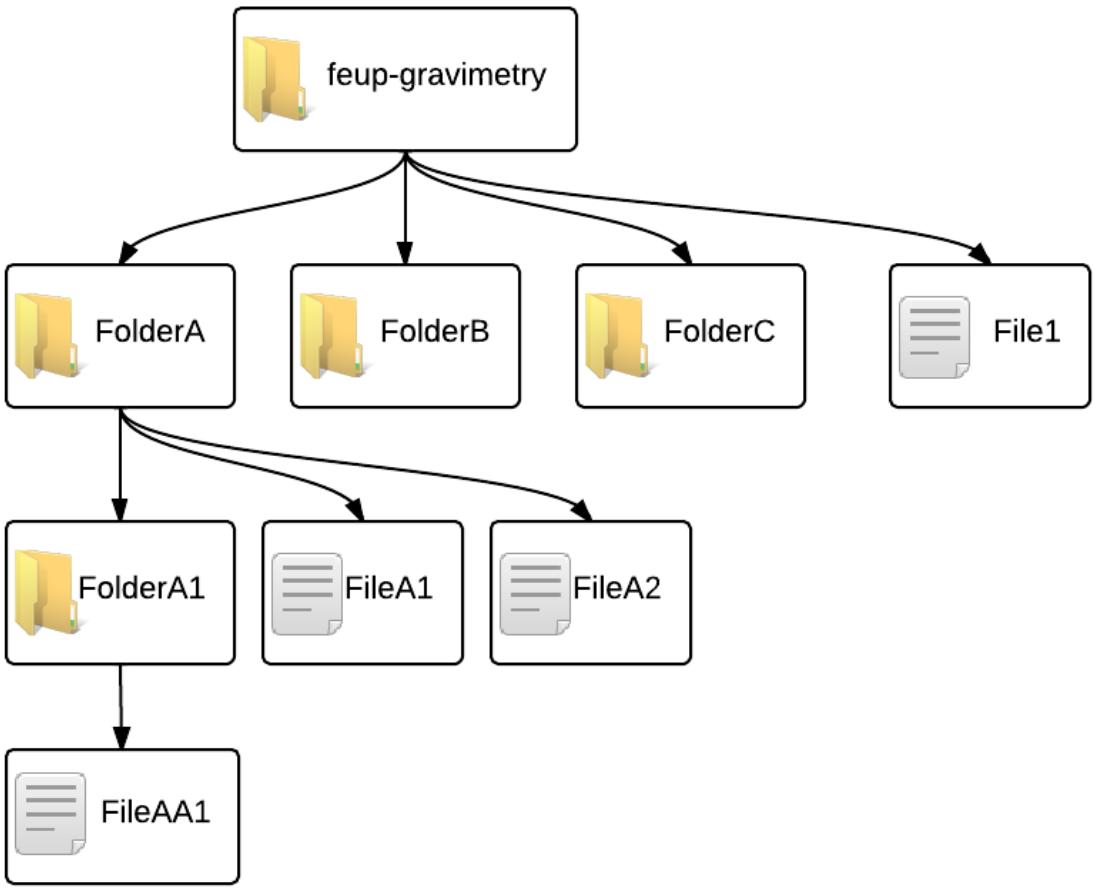

Figura A.1: Estrutura de um diretório exemplo.

Representação da Estrutura de Diretórios

```
{
  id: 34,
  path: "ei08036/feup-gravimetry/",
  name: "feup-gravimetry",
  parent_id: 1,
  created_at: "2013-01-10T14:18:14Z",
  updated_at: "2013-01-10T14:18:14Z",
  files: [
    {
      content_type: "text/plain",
      created_at: "2013-01-10T14:19:24Z",
      file_size: 0,
      folder_id: 34,
      id: 52,
      name: "file1.txt",
      path: "ei08036/feup-gravimetry/",
      uid: "902929cfb03113ae134308e60fa7a3dee3d8b806",
      updated_at: "2013-01-10T14:19:24Z"
    }
  ],
  folders: [
    {
      id: 36,
      path: "ei08036/feup-gravimetry/FolderA/",
      name: "FolderA",
      parent_id: 34,
      created_at: "2013-01-10T14:18:31Z",
      updated_at: "2013-01-10T14:18:31Z",
      files: [
        {
          content_type: "text/plain",
          created_at: "2013-01-10T14:19:50Z",
          file_size: 0,
          folder_id: 36,
          id: 53,
          name: "fileA1.txt",
          path: "ei08036/feup-gravimetry/FolderA/",
          uid: "1b5d6b24e12f6d39b256003bb0811a3049237835",
          updated_at: "2013-01-10T14:19:50Z"
        },
        {
          content_type: "text/plain",
          created_at: "2013-01-10T14:20:02Z",
          file_size: 0,
          folder_id: 36,
          id: 54,
          name: "fileA2.txt",
          path: "ei08036/feup-gravimetry/FolderA/",
```

```
          uid: "059c71f746159de2d60ca72ac5f87024f5eac29d",
          updated_at: "2013-01-10T14:20:02Z"
        }
      ],
      folders: [
        {
          id: 39,
          path: "ei08036/feup-gravimetry/FolderA/FolderA1/",
          name: "FolderA1",
          parent_id: 36,
          created_at: "2013-01-10T16:42:55Z",
          updated_at: "2013-01-10T16:42:55Z",
          files: [
            {
            content_type: "text/plain",
            created_at: "2013-01-10T16:43:13Z",
            file_size: 0,
            folder_id: 39,
            id: 56,
            name: "fileAA1.txt",
            path: "ei08036/feup-gravimetry/FolderA/FolderA1/",
            uid: "6c71897af48b12e6beb606c3ed85356553c6a5fd",
            updated_at: "2013-01-10T16:43:13Z"
            }
          ],
          folders: [ ]
        }
      ]
    },
    {
      id: 37,
      path: "ei08036/feup-gravimetry/FolderB/",
      name: "FolderB",
      parent_id: 34,
      created_at: "2013-01-10T14:18:36Z",
      updated_at: "2013-01-10T14:18:36Z",
      files: [ ],
      folders: [ ]
    },
    {
      id: 38,
      path: "ei08036/feup-gravimetry/FolderC/",
      name: "FolderC",
      parent_id: 34,
      created_at: "2013-01-10T14:18:38Z",
      updated_at: "2013-01-10T14:18:38Z",
      files: [
        {
          content_type: "text/plain",
```

```
          created_at: "2013-01-10T14:20:24Z",
          file_size: 0,
          folder_id: 38,
          id: 55,
          name: "fileC1.txt",
          path: "ei08036/feup-gravimetry/FolderC/",
          uid: "a19fb9b8caa73c702a48c4562411ebcfd178437b",
          updated_at: "2013-01-10T14:20:24Z"
        }
      ],
      folders: [ ]
    }
  ]
}
```

Listagem A.1: Estrutura de um diretório em JSON

```

<hash>
  <id type="integer">34</id>
  <path>ei08036/feup-gravimetry/</path>
  <name>feup-gravimetry</name>
  <parent-id type="integer">1</parent-id>
  <created-at type="datetime">2013-01-10T14:18:14Z</created-at>
  <updated-at type="datetime">2013-01-10T14:18:14Z</updated-at>
  <files type="array">
    <file>
      <content-type>text/plain</content-type>
      <created-at type="datetime">2013-01-10T14:19:24Z</created-at>
      <file-size type="integer">0</file-size>
      <folder-id type="integer">34</folder-id>
      <id type="integer">52</id>
      <name>file1.txt</name>
      <path>ei08036/feup-gravimetry/</path>
      <uid>902929cfb03113ae134308e60fa7a3dee3d8b806</uid>
      <updated-at type="datetime">2013-01-10T14:19:24Z</updated-at>
    </file>
  </files>
  <folders type="array">
    <folder>
      <id type="integer">36</id>
      <path>ei08036/feup-gravimetry/FolderA/</path>
      <name>FolderA</name>
      <parent-id type="integer">34</parent-id>
      <created-at type="datetime">2013-01-10T14:18:31Z</created-at>
      <updated-at type="datetime">2013-01-10T14:18:31Z</updated-at>
      <files type="array">
        <file>
```

```
32          <content-type>text/plain</content-type>
33          <created-at type="datetime">2013-01-10T14:19:50Z</created-at>
34          <file-size type="integer">0</file-size>
35          <folder-id type="integer">36</folder-id>
36          <id type="integer">53</id>
37          <name>fileA1.txt</name>
38          <path>ei08036/feup-gravimetry/FolderA/</path>
39          <uid>1b5d6b24e12f6d39b256003bb0811a3049237835</uid>
40          <updated-at type="datetime">2013-01-10T14:19:50Z</updated-at>
41        </file>
42        <file>
43          <content-type>text/plain</content-type>
44          <created-at type="datetime">2013-01-10T14:20:02Z</created-at>
45          <file-size type="integer">0</file-size>
46          <folder-id type="integer">36</folder-id>
47          <id type="integer">54</id>
48          <name>fileA2.txt</name>
49          <path>ei08036/feup-gravimetry/FolderA/</path>
50          <uid>059c71f746159de2d60ca72ac5f87024f5eac29d</uid>
51          <updated-at type="datetime">2013-01-10T14:20:02Z</updated-at>
52        </file>
53      </files>
54      <folders type="array">
55        <folder>
56        <id type="integer">39</id>
57        <path>ei08036/feup-gravimetry/FolderA/FolderA1/</path>
58        <name>FolderA1</name>
59        <parent-id type="integer">36</parent-id>
60        <created-at type="datetime">2013-01-10T16:42:55Z</created-at>
61        <updated-at type="datetime">2013-01-10T16:42:55Z</updated-at>
62        <files type="array">
63          <file>
64            <content-type>text/plain</content-type>
65            <created-at type="datetime">2013-01-10T16:43:13Z</created-at>
66            <file-size type="integer">0</file-size>
67            <folder-id type="integer">39</folder-id>
68            <id type="integer">56</id>
69            <name>fileAA1.txt</name>
70            <path>ei08036/feup-gravimetry/FolderA/FolderA1/</path>
71            <uid>6c71897af48b12e6beb606c3ed85356553c6a5fd</uid>
72            <updated-at type="datetime">2013-01-10T16:43:13Z</updated-at>
73          </file>
74        </files>
75        <folders type="array" />
76        </folder>
77      </folders>
78    </folder>
79    <folder>
80      <id type="integer">37</id>
```

```
      <path>ei08036/feup-gravimetry/FolderB/</path>
      <name>FolderB</name>
      <parent-id type="integer">34</parent-id>
      <created-at type="datetime">2013-01-10T14:18:36Z</created-at>
      <updated-at type="datetime">2013-01-10T14:18:36Z</updated-at>
      <files type="array" />
      <folders type="array" />
    </folder>
    <folder>
      <id type="integer">38</id>
      <path>ei08036/feup-gravimetry/FolderC/</path>
      <name>FolderC</name>
      <parent-id type="integer">34</parent-id>
      <created-at type="datetime">2013-01-10T14:18:38Z</created-at>
      <updated-at type="datetime">2013-01-10T14:18:38Z</updated-at>
      <files type="array">
        <file>
          <content-type>text/plain</content-type>
          <created-at type="datetime">2013-01-10T14:20:24Z</created-at>
          <file-size type="integer">0</file-size>
          <folder-id type="integer">38</folder-id>
          <id type="integer">55</id>
          <name>fileC1.txt</name>
          <path>ei08036/feup-gravimetry/FolderC/</path>
          <uid>a19fb9b8caa73c702a48c4562411ebcfd178437b</uid>
          <updated-at type="datetime">2013-01-10T14:20:24Z</updated-at>
        </file>
      </files>
      <folders type="array" />
    </folder>
  </folders>
</hash>
```

Listagem A.2: Estrutura de um diretório em XML

Anexo B

Especificação da API

As seguintes tabelas apresentam a especificação detalhada da API desenvolvida.

Tabela B.1: API UPBox - Efetuar autenticação na UPBox.

Operação	OP01
Nome	/login
Descrição	Efetuar *login* na UPBox.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/login*
Método	POST
Parâmetros	*username*: *Username* do utilizador do SiFEUP.
	password: Palavra passe do utilizador do SiFEUP.
Retorno	cookie.
Erros	Failed login: 400, “Bad Request”

Tabela B.2: API UPBox - Receber a árvore de ficheiros e diretórios de um diretório.

Operação	OP02
Nome	/home
Descrição	Obter ficheiros e pastas para o caminho especificado.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/home/<USER>/<PROJECT>/(<FOLDER>)*
Método	GET
Parâmetros	*datanoteskey*: chave partilhada com o DataNotes
	cookie: *Cookie* da sessão do utilizador.
Retorno	A estrutura de diretórios e ficheiros em XML ou JSON. Exemplo no Anexo A.
Erros	Bad request: 400, “Bad Request”
	Bad auth_token: 401, “Unauthorized”
	Directory not found: 404, “Not Found”

Tabela B.3: API UPBox - Pedido para *download* de um ficheiro.

Operação	OP03
Nome	/download
Descrição	Efetuar pedido para *download* de um ficheiro.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/download/<UID>*
Método	GET
Parâmetros	*cookie*: *Cookie* da sessão do utilizador.
Retorno	O ficheiro pedido.
Erros	Bad request: 400, “Bad Request”
	Bad auth_token: 401, “Unauthorized”
	File not found: 404, “Not Found”

Tabela B.4: API UPBox - *Upload* de um ficheiro para a UPBox.

Operação	OP04
Nome	/upload
Descrição	Submeter um ficheiro para a UPBox.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/upload*
Método	POST
Parâmetros	*path*: Diretório pai do ficheiro a adicionar.
	file: Detalhes do ficheiro em *multipart*.
	cookie: *Cookie* da sessão do utilizador.
Retorno	-
Erros	Bad auth_token: 401, “Unauthorized”
	File not found: 404, “Not Found”

Tabela B.5: API UPBox - Criar um diretório.

Operação	OP05
Nome	/create_folder
Descrição	Criar um diretorio.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/create_folder*
Método	POST
Parâmetros	*path*: Diretório pai do diretório a criar.
	cookie: *Cookie* da sessão do utilizador.
Retorno	-
Erros	Bad request: 400, “Bad Request”
	Bad auth_token: 401, “Unauthorized”
	File not found: 404, “Not Found”.
	Empty name or special characters: 406, “Not Acceptable”

Tabela B.6: API UPBox - Eliminar um diretório.

Operação	OP06
Nome	/delete_folder
Descrição	Elimina um diretório e todos os seus filhos.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/delete_folder*
Método	DELETE
Parâmetros	*id*: Identificador do diretório a eliminar.
	cookie: *Cookie* da sessão do utilizador.
Retorno	-
Erros	Bad request: 400, “Bad Request”
	Bad auth_token: 401, “Unauthorized”
	Directory not found: 404, “Not Found”.

Tabela B.7: API UPBox - Eliminar um ficheiro.

Operação	OP07
Nome	/delete_file
Descrição	Elimina um ficheiro da UPBox.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/delete_file*
Método	DELETE
Parâmetros	*uid*: Identificador do ficheiro a eliminar.
	cookie: *Cookie* da sessão do utilizador.
Retorno	-
Erros	Bad request: 400, “Bad Request”
	Bad auth_token: 401, “Unauthorized”
	File not found: 404, “Not Found”.

Tabela B.8: API UPBox - Obter projetos de um utilizador

Operação	OP08
Nome	/userprojects
Descrição	Obter todos os projetos referentes a um utilizador.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/userprojects*
Método	GET
Parâmetros	*uid*: identificação do utilizador
	datanoteskey: chave partilhada com o DataNotes
	cookie: *Cookie* da sessão do utilizador.
Retorno	Um array com os projetos de um utilizador em XML ou JSON.
Erros	Bad auth_token: 401, "Unauthorized"
	User not found: 404, "Not Found".

Tabela B.9: API UPBox - Receção do *backup* de uma anotação.

Operação	OP09
Nome	/backup_annotation
Descrição	Receber uma anotação do DataNotes.
Estrutura URL	*http://dendro-dev.fe.up.pt:3000/backup_annotation*
Método	POST
Parâmetros	*path*: Caminho do ficheiro ou diretório anotado.
	datanoteskey: chave partilhada com o DataNotes
Retorno	-
Erros	Bad auth_token: 401, "Unauthorized"
	File or directory not found: 404, "Not Found".

Anexo C

Teste de Usabilidade

A Universidade do Porto conta atualmente com um repositório de dados experimental para partilha de dados de investigação de diversas áreas dentro da comunidade. A UPBox pretende ser um sistema de armazenamento online que permita a gestão e partilha de dados num grupo de investigação. A qualquer momento o investigador poderá submeter esses dados de investigação para o repositório de dados da UP através da UPBox.

É importante salientar que os dados na UPBox são privados e só podem ser acedidos pelos investigadores relacionados com o projeto em questão e não é obrigatória a submissão dos mesmos no repositório de dados da UP.

Este estudo pretende avaliar o *web site* UPBox quer em termos de usabilidade quer em termos de funcionalidades. Conforme for realizando as tarefas propostas é importante que vá expressando o seu pensamento bem como as dificuldades que vai encontrando.

Imagine que pertence a um grupo de investigação do qual resultaram alguns dados de investigação que devem ser acessíveis a todo o grupo; para isso decidiu utilizar a UPBox para a partilha e gestão de dados dentro do grupo. Se preferir, antes de efetuar as tarefas abaixo enumeradas pode explorar o *web site* durante 5 minutos.

Tarefas

1. Efetuar *login* na aplicação com as credenciais da FEUP/UP.
2. Após esta autenticação crie um projeto.
3. Adicione um utilizador/investigador ao projeto criado.
4. Adicione alguns dados de investigação (diretórios e ficheiros).
5. Efetue *download* de um ficheiro do projeto.
6. Volte à raiz do projeto e elimine um diretório ou um ficheiro.
7. Anote um ficheiro ou um diretório.

Anexo D

Manual de Instalação

Este guia detalha todos os passos necessários para a instalação da UPBox.
URL: http://dendro-dev.fe.up.pt:3000/

1. Efetuar instalação de:

   Ruby 1.9.3

   Rails 3.2.7 ou superior;

   PostgreSQL 8.4;

   SQLite 3.7.3;

   nginx:

   sudo apt-add-repository ppa:brightbox/passenger-nginx

   sudo apt-get update

   sudo apt-get install nginx-full

2. Efetuar *download* do projeto:

   svn checkout https://svn.fe.up.pt/repos/upbox

3. Configurar o ngix:

   nano /opt/nginx/conf/nginx.conf

   substituir o seguinte bloco:

   location / {

   root html;

   index index.html index.htm;

   }

   por:

passenger_enabled on;

root /home/jpedro/upbox/upbox/public/;

client_max_body_size 500M;

Caso seja necessário alterar a porta do servidor deverá ser alterado o valor *listen* do ficheiro *nginx.conf*;

Caso seja necessário alterar a o tamanho máximo dos pedidos ao servidor deverá ser alterado o valor *client_max_body_size* do ficheiro *nginx.conf*;

4. Instalar as *gems* do *rails*:

   Abrir o diretório do projeto

   bundle install

5. Configurar a base de dados:

   rake db:create RAILS_ENV=production

   rake db:migrate RAILS_ENV=production

6. Compilar os *assets*:

   bundle exec rake assets:precompile RAILS_ENV=production

7. Executar o nginx:

   /opt/nginx/sbin/nginx

8. Reiniciar o servidor:

   touch tmp/restart.txt

9. Testar todo o funcionamento num *browser*;

10. Para aceder ao *log* do *debug* do *rails* deverão ser executados os seguintes comandos:

    tail -f log/production.log

    tail -f /opt/nginx/logs/error.log