



Universidade do Minho

Escola de Engenharia

Departamento de Informática

José Carlos Lima Martins

**CLAV:
API de dados e Autenticação**

Relatório de Pré-Dissertação

November 2019



Universidade do Minho

Escola de Engenharia

Departamento de Informática

José Carlos Lima Martins

**CLAV:
API de dados e Autenticação**

Relatório de Pré-Dissertação

Master dissertation

Master Degree in Computer Science

Dissertation supervised by

José Carlos Leite Ramalho

November 2019

ABSTRACT

Write abstract here (en)

RESUMO

Escrever aqui resumo (pt)

CONTEÚDO

1	INTRODUÇÃO	2
1.1	Objetivos	3
2	ESTADO DA ARTE	4
2.1	JavaScript e a Programação Assíncrona	4
2.1.1	Callbacks	4
2.1.2	Promessas	4
2.2	REST	4
2.3	express	4
2.4	passport	5
2.5	jsonwebtoken	5
2.6	passport-jwt	5
2.7	CORS	5
2.8	axios	5
2.9	HTTP Status	5
2.10	Headers do HTTP	5
2.11	Autenticação.gov	5
2.12	exceljs	5
2.13	MongoDB	5
2.14	mongoose	5
2.15	Web Semântica	5
2.15.1	RDF	5
2.15.2	SPARQL	5
2.16	GraphDB	6
2.17	Swagger	6
2.18	Swagger-UI	6
2.19	yaml-include	6
2.20	swagger-ui-express	6
2.21	js-yaml	6
2.22	Nginx	6
2.23	Ontologia	6
2.24	Docker	6
2.25	Docker Compose	6
2.26	Lista Consolidada	6
2.27	Tabelas de Seleção	6

2.28	Cache e Fecho Transitivo	6
3	O PROBLEMA E OS SEUS DESAFIOS	7
4	CONCLUSÃO	8

LISTA DE FIGURAS

LISTA DE TABELAS

GLOSSÁRIO

Application Programming Interface Interface ou protocolo de comunicação entre um cliente e um servidor [i](#)

ontologia Representação de conhecimento (conceitos e as relações entre estes) [2](#)

Simplex Programa de Simplificação Administrativa e Legislativa [2](#)

LISTA DE ACRÓNIMOS

AP Administração Pública [2](#)

API Application Programming Interface [3](#), *Glossary*: [Application Programming Interface](#)

CLAV Classificação e Avaliação da Informação Pública [2](#), [3](#)

CSV Comma Separated Values [3](#)

DGLAB Direção-Geral do Livro, dos Arquivos e das Bibliotecas [2](#)

JSON JavaScript Object Notation [3](#)

LC Lista Consolidada [2](#), [3](#)

RDF Resource Description Framework [3](#)

REST Representational State Transfer [3](#)

UM Universidade do Minho [2](#)

XML Extensible Markup Language [3](#)

INTRODUÇÃO

Vemos atualmente a mudança de paradigma em várias organizações e governos em relação a políticas e estratégias para a disponibilização de dados abertos nos domínios das ciências e da [Administração Pública](#). Quanto à [Administração Pública](#) portuguesa têm sido promovidas políticas para a sua transformação digital com o objetivo de otimização de processos, a modernização de procedimentos administrativos e a redução de papel. De certa forma a agilização de procedimentos da [Administração Pública](#) portuguesa. (Lourenço et al., 2019)

De forma a alcançar estes objetivos a [Administração Pública](#) (AP) tem desmaterializado processos e tem promovido a adoção de sistemas de gestão documental eletrónica bem como da digitalização de documentos destinados a serem arquivados. (Lourenço et al., 2019)

Por forma a continuar esta transformação da AP a [Direção-Geral do Livro, dos Arquivos e das Bibliotecas](#) (DGLAB) apresentou a iniciativa da [Lista Consolidada](#) (LC) para a classificação e avaliação da informação pública. A LC serve de referencial para a construção normalizada dos planos de classificação e tabelas de seleção das entidades que executam funções do Estado. (Lourenço et al., 2019)

Nasce assim o projeto [Classificação e Avaliação da Informação Pública](#) (CLAV) com um dos seus objetivos primordiais a operacionalização da utilização da LC, numa colaboração entre a DGLAB e a [Universidade do Minho](#) (UM) e financiado pelo Simplex. (Lourenço et al., 2019)

A plataforma CLAV disponibiliza em formato aberto uma [ontologia](#) com as funções e processos de negócio das entidades que exercem funções públicas (ou seja a LC) associadas a um catálogo de legislação e de organismos. Desta forma, a CLAV viabiliza a desmaterialização dos procedimentos associados à elaboração de tabelas de seleção tendo como base a LC e ao controlo de eliminação e arquivamento da informação pública através da integração das tabelas de seleção nos sistemas de informação das entidades públicas alertando-as quando determinado documento deve ser arquivado ou eliminado. Esta integração promove também a interoperabilidade através da utilização de uma linguagem comum (a LC) usada no registo, na classificação e na avaliação da informação pública. (Lourenço et al., 2019)

1.1 OBJETIVOS

A continuação do desenvolvimento da [API](#) de dados da [CLAV](#) nesta dissertação, seguindo uma metodologia [REST](#), permite a processos ou aplicações aceder aos dados sem a intervenção humana para além de suportar a plataforma [CLAV](#). Um dos objetivos da [API](#) de dados é permitir futuramente a criação de novas aplicações através desta. Como tal, é extramamente essencial que a [API](#) de dados do [CLAV](#) possua uma boa documentação ajudando futuros programadores ou utilizadores a utilizar a [API](#). Advém daí a necessidade de nesta dissertação realizar a documentação da [API](#) de dados em *Swagger*.

Apesar de o projeto ter em mente a disponibilização aberta de informação pública é necessário controlar a adição, edição e eliminação da informação presente na [Lista Consolidada](#), bem como a informação de utilizadores, da legislação, das entidades, etc, mantendo-a consistente e correta. É, portanto, necessário controlar os acessos à [API](#) de dados com múltiplos níveis de acesso restringindo as operações que cada utilizador pode realizar consoante o seu nível. Desta forma garante-se que apenas pessoal autorizado pode realizar modificações aos dados.

Este controlo de acesso exige a existência de formas de autenticação. Como um cofre para o qual ninguém tem a chave não é útil pelo facto de que algo lá guardado ficará eternamente inacessível, também algo com controlo de acesso seria inútil caso não fosse possível ultrapassar esse controlo de alguma forma. Assim, uma das formas de autenticação usadas, Autenticação.gov, criada pelo Estado português, permite a autenticação dos cidadãos portugueses nos vários serviços públicos [AMA](#) entre os quais, a Segurança Social, o Serviço Nacional de Saúde e a Autoridade Tributária Aduaneira. Sendo este um projeto do Governo Português, pretende-se que seja possível a autenticação no [CLAV](#) através do Autenticação.gov.

Por forma a contrariar o aumento da complexidade da [API](#) de dados com a adição do controlo de acesso e da autenticação pretende-se investigar se a criação de um API Gateway simplifica a comunicação entre interface/utilizadores e a [API](#) de dados.

Resumidamente, os objetivos desta dissertação são:

- Documentação em *Swagger* da [API](#) de dados da [CLAV](#)
- Adição de formatos de exportação à [API](#) de dados da [CLAV](#) (para além do já presente [JSON](#), adicionar [CSV](#), [XML](#) e [RDF](#))
- (Continuação da) Integração do Autenticação.gov na [CLAV](#)
- Proteção da [API](#) de dados da [CLAV](#) com múltiplos níveis de acesso
- Estudo da criação de um [API](#) Gateway
- Integração do [CLAV](#) no iAP

ESTADO DA ARTE

2.1 JAVASCRIPT E A PROGRAMAÇÃO ASSÍNCRONA

A linguagem *JavaScript* teve a sua origem no *browser Netscape Navigator* como uma forma de adicionar programas a páginas web. [Haverbeke \(2018\)](#) Hoje em dia é usado por todos os *browsers* e tornou possível as aplicações web onde o utilizador pode interagir sem precisar de realizar *refresh* ao fim de cada ação. Contudo o *Javascript* é bastante liberal no que permite o programador escrever, facilitando a aprendizagem para novos programadores mas tornando a tarefa de resolver e encontrar problemas bem mais difícil visto não apontar aonde estão esses problemas. Ainda assim esta flexibilidade permite uma quantidade de técnicas que não são possíveis em linguagens mais rígidas e que permitem ultrapassar algumas falhas do *Javascript*. [Haverbeke \(2018\)](#)

2.1.1 *Callbacks*

2.1.2 *Promessas*

2.2 REST

[Richardson and Ruby \(2007\)](#)

2.3 EXPRESS

[Satheesh et al. \(2015\)](#)

2.4 PASSPORT

2.5 JSONWEBTOKEN

2.6 PASSPORT-JWT

2.7 CORS

2.8 AXIOS

2.9 HTTP STATUS

2.10 HEADERS DO HTTP

2.11 AUTENTICAÇÃO.GOV

[AMA \(2018\)](#)

2.12 EXCELJS

2.13 MONGODB

[Satheesh et al. \(2015\)](#)

2.14 MONGOOSE

2.15 WEB SEMÂNTICA

[DuCharme \(2011\)](#)

2.15.1 *RDF*

[DuCharme \(2011\)](#)

2.15.2 *SPARQL*

[DuCharme \(2011\)](#)

2.16 GRAPHDB

2.17 SWAGGER

2.18 SWAGGER-UI

2.19 YAML-INCLUDE

2.20 SWAGGER-UI-EXPRESS

2.21 JS-YAML

2.22 NGINX

[Satheesh et al. \(2018\)](#)

2.23 ONTOLOGIA

[Arp et al. \(2015\)](#)

2.24 DOCKER

[Mouat \(2015\)](#)

2.25 DOCKER COMPOSE

[Mouat \(2015\)](#)

2.26 LISTA CONSOLIDADA

2.27 TABELAS DE SELEÇÃO

2.28 CACHE E FECHO TRANSITIVO

O PROBLEMA E OS SEUS DESAFIOS

CONCLUSÃO

BIBLIOGRAFIA

AMA. Autenticação.gov. URL <https://autenticacao.gov.pt/fa/Default.aspx>.

AMA. *Autenticação.gov - Fornecedor de autenticação da Administração Pública Portuguesa*, 1.5.1 edition, 12 2018.

Robert Arp, Barry Smith, and Andrew Spear. *Building Ontologies with Basic Formal Ontology*. MIT Press, 1st edition, 7 2015. ISBN 978-0-262-52781-1.

Bob DuCharme. *Learning SPARQL*. O'Reilly, 1st edition, 7 2011. ISBN 978-1-449-30659-5.

Marijn Haverbeke. *Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming*. No Starch Press, 3rd edition, 12 2018. ISBN 978-1-59327-950-9.

Alexandra Lourenço, José Carlos Ramalho, Maria Rita Gago, and Pedro Penteado. *Plataforma clav: contributo para a disponibilização de dados abertos da administração pública em portugal*, 7 2019. URL http://eprints.rclis.org/38643/1/Plantilla_EDICIC2019Barcelona-PT-COM_Penteado_v08.pdf.

Adrian Mouat. *Using Docker*. O'Reilly, 1st edition, 12 2015. ISBN 978-1-491-91576-9.

Leonard Richardson and Sam Ruby. *RESTful Web Services*. O'Reilly, 1st edition, 5 2007. ISBN 978-0-596-52926-0.

Mithun Satheesh, Mithun Satheesh, and Jason Krol. *Web Development with MongoDB and NodeJS*. Packt Publishing, 2nd edition, 10 2015. ISBN 978-1-78528-752-7.

Mithun Satheesh, Mithun Satheesh, and Jason Krol. *NGINX Cookbook*. O'Reilly, 1st edition, 11 2018. ISBN 978-1-491-96893-2. Second Release.

NB: place here information about funding, FCT project, etc in which the work is framed. Leave empty otherwise.