

UNIVERSIDADE FEDERAL DE OURO PRETO  
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS  
DEPARTAMENTO DE COMPUTAÇÃO

Mateus Vitor Pereira Lana

**UFOP ATIVA: UMA FERRAMENTA PARA  
EXTRAÇÃO E ANÁLISE DE DADOS  
ACADÊMICOS DE SERVIDORES DA UFOP  
BASEADA NO CURRÍCULO LATTES**

Ouro Preto, MG  
2019

Mateus Vitor Pereira Lana

UFOP ATIVA: UMA FERRAMENTA PARA EXTRAÇÃO E ANÁLISE DE DADOS  
ACADÊMICOS DE SERVIDORES DA UFOP BASEADA NO CURRÍCULO LATTES

Monografia II apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

**Orientador:** Prof. Dr. Anderson Almeida Ferreira

**Coorientador:** Msc. Helgem de Souza Ribeiro Martins

Ouro Preto, MG  
2019

cutte

Lana, Mateus Vitor Pereira.

UFOP Ativa: uma ferramenta para extração e análise de dados acadêmicos de servidores da UFOP baseada no currículo Lattes [manuscrito] / Mateus Vitor Pereira Lana. - 2019.

70f.: il.; color; tabs.

Orientador: Prof. Dr. Anderson Almeida Ferrreira.

Coorientador: Prof. MSc. Helgem de Souza Ribeiro Martins.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Biológicas. Departamento de Computação.

1. Plataforma Lattes. 2. Pesquisa. 3. Banco de dados. 4. Extração de dados. 5. Servidores. I. Ferreira, Anderson Almeida. II. Martins, Helgem de Souza Ribeiro. III. Universidade Federal de Ouro Preto. IV. Título.

CDU:

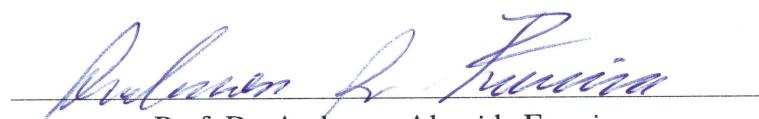
Catalogação: [ficha.sisbin@ufop.edu.br](mailto:ficha.sisbin@ufop.edu.br)

Mateus Vitor Pereira Lana

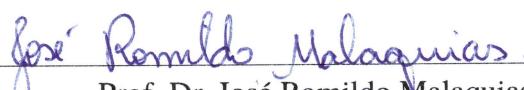
**UFOP ATIVA: UMA FERRAMENTA PARA EXTRAÇÃO E ANÁLISE  
DE DADOS ACADÊMICOS DE SERVIDORES DA UFOP BASEADA  
NO CURRÍCULO LATTES**

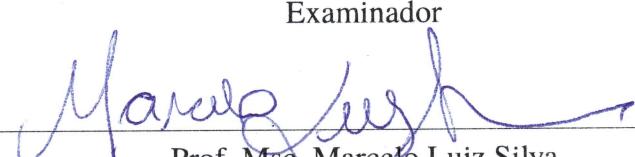
Monografia II apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau em Bacharel em Ciência da Computação.

Aprovada em Ouro Preto, 12 de dezembro de 2019.

  
Prof. Dr. Anderson Almeida Ferreira  
Universidade Federal de Ouro Preto - UFOP  
Orientador

  
Msc. Helgem de Souza Ribeiro Martins  
Universidade Federal de Ouro Preto - UFOP  
Coorientador

  
Prof. Dr. José Romildo Malaquias  
Universidade Federal de Ouro Preto - UFOP  
Examinador

  
Prof. Msc. Marcelo Luiz Silva  
Universidade Federal de Ouro Preto - UFOP  
Examinador

# Agradecimentos

Primeiramente a Deus por ter me dado muita força e saúde e estar sempre ao meu lado nas dificuldades.

A meus pais, meus irmãos e minha família, por acreditarem em mim e por estarem sempre me incentivando e apoiando.

Aos meus amigos da faculdade e os demais, pelo apoio durante este período extremamente importante da minha vida.

Ao meu orientador Anderson e o coorientador Helgem por todo apoio e paciência ao longo da elaboração do meu projeto final.

A todos os professores da instituição, em especial do departamento de computação, por todos os conselhos e ajuda durante os meus estudos.

A todos os funcionários da Universidade Federal de Ouro Preto e a própria instituição por todo apoio e por proporcionarem um ambiente propício para o desenvolvimento do meu trabalho de conclusão de curso.

A todas as pessoas que direta ou indiretamente contribuíram para a realização da minha pesquisa.

# Resumo

O conhecimento, concebido como algo imensurável, faz-se um pressuposto condutor da nossa vivência cotidiana. Dessa forma, é possível perceber que esta afirmativa pode se ajustar de maneira precisa quando o assunto está relacionado à pesquisa de um modo geral, que pode ser compreendida como a semente geradora desse conhecimento. A humanidade foi e continuará sendo movida pelo conhecimento e pelas inovações que são oriundas das pesquisas que são realizadas dia após dia no mundo inteiro. De acordo com pesquisas da *Clarivate Analytics (CROSS; THOMSON; SINCLAIR, 2018)*, no período de 2011 a 2016, o Brasil é o 13º maior produtor mundial de publicações de pesquisas e sua produção acadêmica cresce anualmente. Atualmente, a UFOP conta com, além de vários servidores e alunos, mais de novecentos professores efetivos e, grande parte destes estão envolvidos em diversos projetos e realizam pesquisas e orientações nas mais diversas áreas de estudo presentes na universidade. Para instituições de ensino superior, como a UFOP, é muito importante que esteja visível e facilmente acessível levantamentos de dados estatísticos relevantes, que estejam relacionados a produção de pesquisa científica realizada por seus servidores, departamentos e institutos. Sendo assim, este projeto teve como principal finalidade o desenvolvimento de um sistema que agrupe e organize em um banco de dados as informações, provenientes da Plataforma Lattes, que estejam relacionadas aos servidores atuantes na UFOP. Além disso, o sistema teve como objetivo ser simples, com uma interface amigável e intuitiva, buscando uma boa organização das informações.

**Palavras-chave:** Plataforma Lattes. Pesquisa. Banco de Dados. Extração de Dados. Currículo Lattes. Servidores.

# Abstract

Knowledge, conceived as something immeasurable, becomes a guiding presupposition of our daily experience. In this way, it is possible to perceive that this affirmative can adjust of precise way when the subject is related to the research in general, that can be understood as the seed that generates this knowledge. Humanity has been and will continue to be driven by the knowledge and innovations that come from research that is carried out day after day all over the world. According to Clarivate Analytics' surveys ([CROSS; THOMSON; SINCLAIR, 2018](#)), from 2011 to 2016, Brazil is the thirteenth largest producer of research publications worldwide and its academic production grows annually. Currently, the Federal University of Ouro Preto has more than nine hundred effective professors who are involved in several projects and carry out researches and orientations in the most diverse areas of studies present at the university. For higher education institutions such as UFOP, it is very important that surveys of relevant statistical data that are related to the production of scientific research carried out by the various professors, departments and institutes of these institutions are visible and easily accessible. Therefore, this project had as its main purpose the development of a system that gathers and organizes in a database the information, coming from the Lattes platform, that is related to the servers acting on UFOP. In addition, the system aimed to be simple, with a friendly and intuitive interface, seeking a good organization of information.

**Keywords:** Lattes Platform, Research, Database, Data Extraction, Curriculum Lattes, Servers.

# Listas de Ilustrações

Figura 1.1 – Gráfico de Produção Científica dos Docentes da UFOP. . . . .	2
Figura 2.1 – (a) Definição de DTD, (b) Documento XML que segue a definição. . . . .	8
Figura 2.2 – Categorias de entidades, atributos e relacionamentos. . . . .	10
Figura 2.3 – Diagrama de entidade e relacionamento. . . . .	11
Figura 2.4 – Processo da metodologia ágil Scrum. . . . .	12
Figura 3.1 – Diagrama de Entidade e Relacionamento. . . . .	28
Figura 3.2 – Diagrama de Contexto da Aplicação. . . . .	30
Figura 3.3 – Hierarquia de Atores da Aplicação. . . . .	31
Figura 3.4 – Tela inicial da aplicação. . . . .	32
Figura 3.5 – Inserção de um texto no campo de pesquisa. . . . .	33
Figura 3.6 – Tela da aba "Dados Relevantes". . . . .	33
Figura 3.7 – Tela da aba "Fale Conosco". . . . .	34
Figura 3.8 – Tela da aba "Sobre a Plataforma". . . . .	34
Figura 3.9 – Visualização do perfil de um servidor. . . . .	35
Figura 3.10–Tela inicial da aplicação para o administrador. . . . .	35
Figura 3.11–Tela dos Servidores Cadastrados para o administrador. . . . .	36
Figura 3.12–Alerta no momento de remoção de um servidor. . . . .	36
Figura 3.13–Tela de “Relatórios” para o administrador. . . . .	37
Figura 3.14–Fluxograma do processo de download automático dos Currículos. . . . .	48
Figura 3.15–Processos para acessar e construir documentos XML com base em aplicações Java. . . . .	50
Figura 3.16–Fluxograma do processo de mapeamento para Objetos Java. . . . .	51
Figura 3.17–Exemplo do mapeamento de atributos simples de uma pessoa com métodos <i>setters</i> . . . . .	52
Figura 3.18–Exemplo do mapeamento da lista de áreas de uma pessoa utilizando <i>add</i> em <i>list</i> . .	52
Figura 3.19–Exemplo de mapeamento dos artigos de uma pessoa utilizando <i>add</i> em <i>list</i> . .	53
Figura 3.20–Fluxograma do mapeamento Objeto-Relacional. . . . .	53
Figura 3.21–Exemplo: deduplicação de dados. . . . .	54
Figura 3.22–Exemplo da configuração adicionada ao <i>pom.xml</i> para utilizar o SGBD PostgreSQL na aplicação. . . . .	56
Figura 3.23–Exemplo da configuração adicionada <i>application.properties</i> para utilizar o SGBD PostgreSQL na aplicação. . . . .	56
Figura 3.24–Definição da entidade <i>pessoa</i> , sua especialização e uma chave primária <i>id</i> . .	57
Figura 3.25–Definição de relacionamento $N \times M$ gerando a tabela <i>pessoa_producao</i> , (a) Lado da classe Producao (b) Lado da classe Pessoa . . . . .	57

Figura 3.26–Definição de relacionamento $1 \times N$ , (a) Lado da classe Palavra_Chave (b)	
Lado da classe Pessoa . . . . .	58
Figura 3.27–Interface que define o repositório do modelo <i>Pessoa</i> . . . . .	58
Figura 3.28–Classe que define o serviço referente ao modelo <i>Pessoa</i> . . . . .	59
Figura 3.29–Classe que define o recurso referente ao serviço <i>Pessoa</i> . . . . .	59
Figura 3.30–Processamento de uma requisição pelo <i>backend</i> . . . . .	60
Figura 3.31–Exemplo de componente que representa uma <i>Unidade</i> . . . . .	61
Figura 3.32–Exemplo de serviço para buscar <i>Unidades</i> . . . . .	61
Figura 3.33–Exemplo de um modelo que define a estrutura de um objeto <i>Unidade</i> . . . . .	62
Figura 3.34–Funcionamento do <i>frontend</i> . . . . .	62
Figura 3.35–Tela inicial da aplicação. . . . .	62
Figura 4.1 – Tela inicial da aplicação com realização de uma pesquisa. . . . .	64
Figura 4.2 – Tela "Dados Relevantes". . . . .	64
Figura 4.3 – Tela de um servidor. . . . .	65
Figura 4.4 – Tela de uma unidade. . . . .	66
Figura 4.5 – Tela de um departamento. . . . .	66

# **Lista de Tabelas**

Tabela 2.1 – Processo unificado de desenvolvimento de um software. . . . .	13
Tabela 3.1 – Descrição do Diagrama de Contexto. . . . .	31
Tabela 3.2 – Descrição da Hierarquia de Atores. . . . .	32

# **Lista de Abreviaturas e Siglas**

API	Application Programming Interface
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
CORS	Cross-Origin Resource Sharing
CPF	Cadastro de Pessoa Física
CSS	Cascading Style Sheets
CSV	Comma-separated Values
DOI	Digital Object Identifier
DTD	Document Type Definition
EUA	Estados Unidos da América
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ISBN	International Standard Book Number
ISSN	International Standard Serial Number
Java EE	Java Platform, Enterprise Edition
JAXB	Java Architecture for XML Binding
JPA	Java Persistence API
JSON	JavaScript Object Notation
NTI	Núcleo de Tecnologia da Informação
ORM	Object-relational mapping
PHP	Hypertext Preprocessor
PROPP	Pró-Reitoria de Pesquisa e Pós-Graduação
SBBD	Simpósio Brasileiro de Bancos de Dados
SGBD	Sistema Gerenciador de Banco de Dados

SGML	Standard Generalized Markup Language
SOAP	Simple Object Access Protocol
UFMG	Universidade Federal de Minas Gerais
UFOP	Universidade Federal de Ouro Preto
URL	Uniform Resource Locator
UX	User Experience
W3C	World Wide Web Consortium
WS	Web Service
WSDL	Web Services Description Language
XML	Extensible Markup Language

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Justificativa	2
1.2	Objetivos Geral e Específicos	3
1.3	Organização do Trabalho	3
<b>2</b>	<b>Revisão Bibliográfica</b>	<b>5</b>
2.1	Trabalhos Relacionados	5
2.2	Fundamentação Teórica	6
2.2.1	Dados Semiestruturados, XML e DTD	7
2.2.2	Estrutura do Currículo Lattes	8
2.2.3	Modelagem de Banco de Dados	10
2.2.4	Processo de Concepção de Software	11
<b>3</b>	<b>Desenvolvimento</b>	<b>14</b>
3.1	Modelagem do Banco de Dados	14
3.1.1	Dicionário de Dados	15
3.1.1.1	Aluno	15
3.1.1.2	Area	15
3.1.1.3	Arte_Cenica	15
3.1.1.4	Arte_Visual	16
3.1.1.5	Artigo	16
3.1.1.6	Atuacao_Profissional	17
3.1.1.7	Autor	17
3.1.1.8	Autor_Externo	18
3.1.1.9	Capitulo_de_Livro	18
3.1.1.10	Departamento	19
3.1.1.11	Financiador	19
3.1.1.12	Formacao_Academica	19
3.1.1.13	Livro	20
3.1.1.14	Musica	20
3.1.1.15	Orientacao_Concluida	21
3.1.1.16	Palavra_Chave	21
3.1.1.17	Patente	22
3.1.1.18	Pessoa	22
3.1.1.19	Producao	23
3.1.1.20	Producao_Artistica_Cultural	23
3.1.1.21	Producao_Bibliografica	24
3.1.1.22	Producao_Tecnica	24

3.1.1.23	Produto_Tecnologico . . . . .	24
3.1.1.24	Programa_Pos_Graduacao . . . . .	24
3.1.1.25	Projeto_de_Pesquisa . . . . .	25
3.1.1.26	Servidor . . . . .	25
3.1.1.27	Software . . . . .	25
3.1.1.28	Trabalho_em_Evento . . . . .	26
3.1.1.29	Trabalho_Tecnico . . . . .	26
3.1.1.30	Unidade . . . . .	27
3.1.1.31	Vinculo_Institucional . . . . .	27
3.2	Concepção do Software . . . . .	29
3.2.1	Objetivo . . . . .	29
3.2.2	Missão e Descrição do Produto . . . . .	29
3.2.2.1	Nome do Produto e Componentes . . . . .	29
3.2.2.2	Missão do Produto . . . . .	29
3.2.2.3	Descrição do Produto . . . . .	29
3.2.3	Modelo de Negócio . . . . .	29
3.2.4	Escopo Positivo e Negativo . . . . .	30
3.2.4.1	Positivo . . . . .	30
3.2.4.2	Negativo . . . . .	30
3.2.5	Diagrama de Contexto . . . . .	30
3.2.6	Hierarquia de Atores . . . . .	31
3.2.7	Protótipo . . . . .	32
3.2.7.1	Perspectiva do Usuário Visitante . . . . .	32
3.2.7.2	Perspectiva de Administrador . . . . .	35
3.2.8	Backlog . . . . .	37
3.2.8.1	Serviço de Navegação . . . . .	37
3.2.8.2	Serviço de Gestão de Dados de Servidores . . . . .	37
3.2.8.3	Serviço de Gestão de Dados de Unidades . . . . .	38
3.2.8.4	Serviço de Manutenção do Sistema . . . . .	38
3.2.8.5	Serviço Emissão de Relatórios . . . . .	38
3.2.9	Estórias de Usuário . . . . .	38
3.2.9.1	Serviço de Navegação . . . . .	38
3.2.9.2	Serviço de Gestão de Dados de Servidores . . . . .	39
3.2.9.3	Serviço de Gestão de Dados de Unidades . . . . .	40
3.2.9.4	Serviço de Manutenção do Sistema . . . . .	41
3.2.9.5	Serviço de Emissão de Relatórios . . . . .	41
3.2.10	Cenários de Teste . . . . .	42
3.2.10.1	Serviço de Navegação . . . . .	42
3.2.10.2	Serviço de Gestão de Dados de Servidores . . . . .	43

3.2.10.3	Serviço de Gestão de Dados de Unidades . . . . .	45
3.2.10.4	Serviço de Manutenção do Sistema . . . . .	47
3.2.10.5	Serviço de Emissão de Relatórios . . . . .	47
3.3	Extração dos Dados . . . . .	48
3.3.1	Download Automático dos Currículos . . . . .	48
3.3.2	Extração dos Dados do XML . . . . .	49
3.4	Mapeamento para o Banco de Dados do Sistema . . . . .	50
3.4.1	Mapeamento para Objetos Java . . . . .	51
3.4.2	Mapeamento Objeto-Relacional, JPA e Hibernate . . . . .	53
3.5	Deduplicação de Produções . . . . .	54
3.6	Implementação do <i>Frontend</i> e do <i>Backend</i> do UFOP Ativa . . . . .	55
3.6.1	Backend . . . . .	56
3.6.2	Frontend . . . . .	60
<b>4</b>	<b>Estudo de Caso . . . . .</b>	<b>63</b>
<b>5</b>	<b>Conclusão . . . . .</b>	<b>67</b>
	<b>Referências . . . . .</b>	<b>69</b>

# 1 Introdução

A humanidade foi e continuará sendo movida pelo conhecimento e pelas inovações, que são oriundas das pesquisas realizadas dia após dia no mundo inteiro. A nossa sociedade traz consigo a procura pelo conhecimento, a busca para encontrar soluções para problemas e para descobrir melhorias em diversas áreas do conhecimento. Essa atividade sempre fez e seguirá fazendo parte do processo de desenvolvimento do indivíduo, que se reinventa a cada dia, encarando e solucionando problemas, construindo processos e soluções eficazes para os mesmos. Com a evolução dos estudos, das tecnologias e consequente desenvolvimento da prática da pesquisa, surgiu a pesquisa científica, que traz ao seu lado progressos de conhecimentos de diversas áreas, convertendo-se em vantagens sociais e econômicas (NETO, 2017).

“O conhecimento, concebido como algo imensurável, faz-se um pressuposto norteador da nossa vivência cotidiana” (DUARTE, 2012). Sendo assim, é possível perceber que esta afirmativa pode se ajustar de maneira precisa quando o assunto está relacionado à pesquisa de um modo geral, que pode ser compreendida como a semente geradora deste conhecimento. De forma mais específica, ao falar sobre a pesquisa científica, esta pode ser definida como uma prática que é realizada por pesquisadores, com o objetivo de adquirir conhecimento, para que possam evoluir na vida intelectual, por meio de novas descobertas ou estudos aprofundados em diversos assuntos (DUARTE, 2012). Assim, a pesquisa comprehende algumas das atividades que são desenvolvidas ao longo da vida acadêmica, a qual exige do pesquisador competência e um estudo bem executado, sendo de grande importância e em alguns casos até mesmo essencial aos realizadores.

A pesquisa científica no nosso país deu um salto significativo nos últimos cinquenta anos e vários foram os fatores e atores que contribuíram para isso. É possível compreender que parte deste avanço deve ser creditada a consolidação da política de pós-graduação implantada nos anos sessenta nas principais universidades do Brasil (VERCESI et al., 2002). De acordo com pesquisas analíticas realizadas pela *Clarivate Analytics* (CROSS; THOMSON; SINCLAIR, 2018), a partir de dados da *Web of Science*, no período de 2011 a 2016 o Brasil é o 13º maior produtor mundial de publicações de pesquisa e sua produção acadêmica cresce anualmente. Foram mais de 250 mil papers produzidos no período, 10% da quantidade produzida nos EUA, que ocupam a primeira posição.

Atualmente a Universidade Federal de Ouro Preto conta com mais de mil e seiscentos servidores em atuação e, dentre esses, mais de novecentos professores efetivos que atuam nos 55 cursos oferecidos pela instituição<sup>1</sup>. Vários desses docentes estão envolvidos em diversos projetos e realizam pesquisas e orientações nas mais diversas áreas de estudo presentes na universidade.

<sup>1</sup> <<https://ufop.br/ufop-em-numeros>> acessado em 03/06/2019

Uma planilha de produtividade gerada pela plataforma Somos UFOP, disponível no site da PROPP<sup>2</sup>, mostra a evolução das publicações acadêmicas dos docentes da UFOP entre 2006 e 2016, a partir de dados provenientes da Plataforma Lattes<sup>3</sup>, em relação a três categorias: artigos em periódicos, livros e capítulos e trabalhos em eventos. A Figura 1.1 ilustra os resultados.

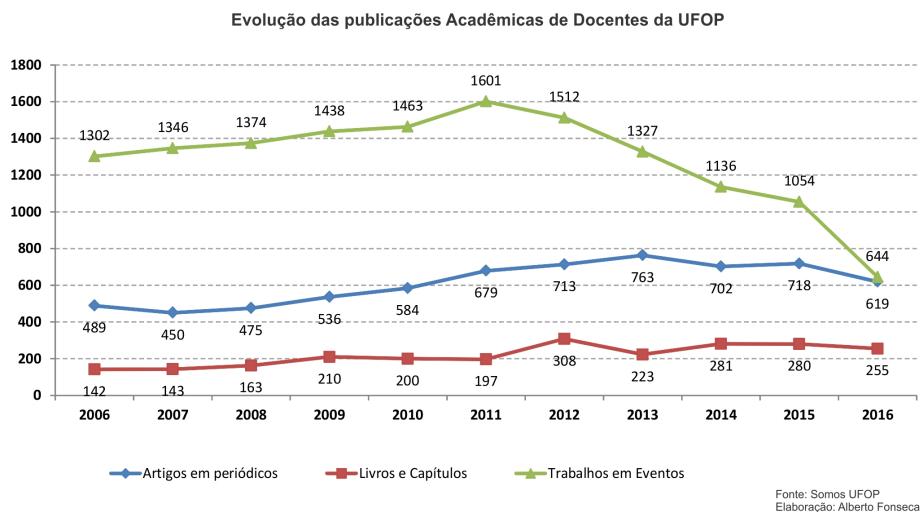


Figura 1.1 – Gráfico de Produção Científica dos Docentes da UFOP.

Fonte: (PROPP, 2016)

Para as instituições de ensino superior, como a UFOP, é muito importante que esteja visível e facilmente acessível levantamentos de dados estatísticos relevantes, que estejam relacionados a produção de pesquisa científica realizada pelos diversos servidores, departamentos e institutos. Além de transparecer melhor essas informações para as instituições, que são de grande valia internamente, também permite que qualquer pessoa acesse e visualize como se desenvolve a pesquisa e quais são as áreas mais aprofundadas nas instituições que ela tem interesse em ingressar ou já está inserida.

O governo brasileiro disponibiliza, por meio da Plataforma Lattes, desenvolvida pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), um repositório de dados com currículos, grupos de pesquisa e instituições, em um único sistema de informações, das áreas de Ciência e Tecnologia. Essa plataforma é aberta para utilização por pessoas interessadas e foi criada para facilitar as ações de planejamento, gestão e operacionalização do fomento à pesquisa.

## 1.1 Justificativa

Algumas instituições acadêmicas do Brasil, ainda não apresentam plataformas que possibilitem o acesso aos projetos nos quais seus funcionários estão envolvidos, bem como outras informações estatísticas que sejam relevantes. Tendo em vista esse cenário, seria interessante uma

<sup>2</sup> <<https://propp.ufop.br/pesquisa/planilha-de-produtividade>> acessado em 03/06/2019

<sup>3</sup> <<http://lattes.cnpq.br/>> acessado em 03/06/2019

ferramenta que extraísse informações acadêmicas a respeito dos funcionários de uma determinada instituição, disponíveis na Plataforma Lattes, e organizasse para um sistema que apresente uma interface amigável e mecanismos de acesso a informações coerentes e com o mínimo possível de erros, como duplicação de informações, por exemplo.

Dessa forma, este projeto se justifica pelo fato de procurar facilitar o acesso a informações dos servidores atuantes em atividades de pesquisa na UFOP, como publicações, áreas de atuação, projetos nos quais estão envolvidos, orientações, especialidades, dentre outras. Além disso, fazer levantamentos e exposição de informações importantes para a instituição, por meio de gráficos e tabelas, por exemplo. O projeto propõe uma plataforma simples, com uma interface amigável e que seja bem intuitiva, buscando uma boa organização das informações.

## 1.2 Objetivos Geral e Específicos

A finalidade principal deste projeto é desenvolver a primeira versão do UFOP Ativa, um sistema que agrupa e organiza em um banco de dados as informações, provenientes da Plataforma Lattes, dos servidores em atuação na UFOP. Além disso, este projeto busca elaborar um sistema que apresente uma interface amigável e de fácil acesso e interação. Nesse sentido, a metodologia adotada foi aplicada visando os seguintes objetivos específicos:

- Realizar o projeto do software UFOP Ativa;
- Desenvolver o módulo de importação dos dados do currículo Lattes de servidores da UFOP;
- Implementar a versão incial do UFOP Ativa;
- Realizar um estudo de caso para mostrar a aplicabilidade da ferramenta.

Como resultados esperados tem-se:

- Contribuir para a facilidade de acesso a informações a respeito dos servidores da UFOP;
- Colaborar com as futuras análises que possam usufruir do sistema concebido;
- Contribuir para uma maior organização do acervo de informações que estão relacionadas às pesquisas realizadas na UFOP.

## 1.3 Organização do Trabalho

A organização deste trabalho segue a seguinte sequência de assuntos: no Capítulo 2, é apresentada uma revisão da literatura, que está fragmentada em duas seções, sendo elas: Trabalhos Relacionados e Fundamentação Teórica. Em seguida, no Capítulo 3, é apresentado

o desenvolvimento do trabalho, com as seções Modelagem do Banco de Dados, Concepção do Software, Extração dos Dados, Mapeamento para o Banco de Dados do Sistema, Deduplicação de Produções e Implementação do *Frontend* e do *Backend* do UFOP Ativa. Na sequência, o Capítulo 4, exibe o estudo de caso do sistema e o Capítulo 5 conclui o trabalho.

## 2 Revisão Bibliográfica

Neste capítulo, há duas seções, que se referem respectivamente aos trabalhos relacionados e a fundamentação teórica. A primeira parte é composta por uma breve apresentação de alguns trabalhos existentes na literatura que possuem alguma relação com o assunto abordado neste projeto ou que propõem uma ideia semelhante a que será desenvolvida. É importante ressaltar que não são muitos os trabalhos que apresentam uma abordagem semelhante a deste trabalho. Os presentes trabalhos relacionados, contribuiram e foram importantes para o desenvolvimento deste projeto. A seção seguinte apresenta toda uma fundamentação teórica necessária para o melhor entendimento deste trabalho, esclarecendo alguns assuntos que serão abordados no mesmo.

### 2.1 Trabalhos Relacionados

Mena-Chalco e Junior (2009) desenvolveram uma ferramenta denominada *scriptLattes*, um projeto de software livre, que é um script GNU-GPL desenvolvido para a extração e compilação automática de: (1) produções bibliográficas, (2) produções técnicas, (3) produções artísticas, (4) orientações, (5) projetos de pesquisa, (6) prêmios e títulos, (7) grafo de colaborações, (8) mapa de geolocalização, e (9) coautoria e internacionalização, de um conjunto de pesquisadores cadastrados na Plataforma Lattes. Associações de Qualis para as produções acadêmicas publicadas em Congressos e Revistas também são considerados.

O *scriptLattes* baixa automaticamente os currículos da Plataforma Lattes em formato *HTML* (livremente disponíveis na rede) de um grupo de pessoas de interesse, compila as listas de produções, tratando apropriadamente as produções duplicadas e similares. São geradas páginas também em *HTML* com listas de produções e orientações separadas por tipo e colocadas em ordem cronológica invertida. Adicionalmente são criados automaticamente vários grafos (redes) de co-autoria entre os membros do grupo de interesse e um mapa de geolocalização dos membros e alunos (de pós-doutorado, doutorado e mestrado) com orientação concluída. Os relatórios gerados permitem avaliar, analisar ou documentar a produção de grupos de pesquisa.

Digiampietri et al. (2012) propõem a identificação rápida de coautorias e a utilização de métricas globais sobre redes de coautoria extraídas de publicações em diferentes períodos de tempo (triênios). Esta proposta representa um avanço na direção da caracterização massiva de extensos grupos acadêmicos, cadastrados na Plataforma Lattes, através de métricas. A relevância desse trabalho recai sobre as vantagens das análises que são realizadas considerando as métricas topológicas de grafos correspondentes a Grandes Áreas do Conhecimento no Brasil. Essas análises são muito importantes para compreender/interpretar a Ciência Brasileira no âmbito das colaborações científicas.

Em outro trabalho, Digiampietri et al. (2015) apresentam um banco de dados formado por mais de um milhão de currículos minerados da Plataforma Lattes e que foram processados, organizados e analisados para servirem de base para a produção e análise de redes sociais de pesquisa. Neste artigo, diversas características desse banco foram apresentadas, bem como algumas relações identificadas entre os currículos, as quais ligam orientandos e orientadores e coautores de publicações científicas. Além disso, são discutidos alguns cuidados que devem ser considerados nos trabalhos que analisam currículos da Plataforma Lattes.

Dias, Moita e Dias (2016) apresentam uma plataforma concebida para extrair toda informação dos currículos presentes na Plataforma Lattes, constituindo um grande repositório de dados científicos. Além disso, são implementados métodos para análises bibliométricas dos dados e identificação de redes de colaboração científica. Os resultados obtidos são apresentados por meio de estudos, que visam obter uma visão geral sobre o repositório de currículos e como o conteúdo destes currículos pode ser manipulado para a caracterização dessas redes de colaboração. Infere-se que estes currículos representam uma fonte profundamente farta de informações científicas e que sua aplicação para estudos bibliométricos e embasados em análise de redes científicas permite obter resultados significativos para compreensão de como a ciência brasileira tem sido realizada. Portanto, todo o potencial da Plataforma Lattes como uma intermediadora de análises bibliométricas é apresentado neste trabalho, sendo proposta para validar esta afirmativa uma plataforma capaz de extrair e analisar todo a coleção de dados com um baixo custo computacional e precisão aceitável.

A plataforma Somos UFMG (SOMOS, 2017)<sup>1</sup>, desenvolvida por Ado Jorio de Vasconcelos, Antônio Leite, Cassiano Rabelo e João Luiz Campos, da UFMG, segundo a própria descrição do sistema, busca facilitar o mapeamento das competências da universidade e demais instituições de ensino que usufruem do serviço, com o objetivo de incrementar a interação das mesmas em áreas de pesquisa científica e tecnológica com instituições públicas e privadas. Ainda afirmam que, através dessa plataforma, é possível identificar os pesquisadores, suas especialidades e produção científica, além de informações sobre Unidades, Departamentos, ativos de propriedade intelectual, infraestrutura instalada nos laboratórios, dentre outras informações. As competências podem ser encontradas de maneira simples e organizada, de acordo com a área de interesse do usuário. É válido salientar que os dados apresentadas nesse sistema também foram extraídas da Plataforma Lattes.

## 2.2 Fundamentação Teórica

Esta seção está subdividida em quatro subseções nas quais serão esclarecidos alguns conceitos que são importantes para melhor entendimento deste trabalho. Na Subseção 2.2.1 há um esclarecimento em relação a alguns conceitos referentes a dados semi-estruturados e,

---

<sup>1</sup> <<http://somos.ufmg.br>> acessado em 03/06/2019

na sequência, na Subseção 2.2.2, há uma descrição bem formalizada da estrutura que compõe os currículos da Plataforma Lattes. Em seguida, na Subseção 2.2.3, é discutida modelagem de bancos de dados. E, para finalizar, no último segmento, Subseção 2.2.4, são apresentadas algumas informações importantes que tangem a concepção de software.

### 2.2.1 Dados Semiestruturados, XML e DTD

Abiteboul (1997) afirma que dados semiestruturados apresentam uma representação estrutural heterogênea, não sendo nem completamente não-estruturados nem estritamente tipados. Dados Web se enquadram nessa definição: em alguns casos os dados possuem uma descrição uniforme (um catálogo de produtos), em outros, algum padrão estrutural pode ser identificado (um conjunto de documentos no formato de artigo), ou então, praticamente não existem informações descritivas associadas (um arquivo de imagem). Buneman (1997) assegura ainda que dados semiestruturados são dados nos quais o esquema de representação está presente (de forma explícita ou implícita) juntamente com o dado, ou seja, o mesmo é auto-descritivo. Isto significa que uma análise do dado deve ser feita para que a sua estrutura possa ser identificada e extraída.

Os dados semiestruturados compõe uma configuração de dado estruturado que não se apresenta em conformidade com a estrutura formal de modelos de dados relativos a bancos de dados relacionais ou modelos lógicos tradicionais, porém possuem alguns marcadores, como *tags*, por exemplo, que têm como finalidade separar elementos semânticos e estabelecer categorias e hierarquias de registros e campos em relação a esses dados. As informações em bancos semiestruturados, diferentemente de bancos relacionais que utilizam relações (tabelas), são armazenadas e manipuladas em formato XML, por exemplo.

XML é um padrão para publicação, combinação e intercâmbio de documentos multimídia, desenvolvido pelo consórcio W3C (W3C, 2008; BRADLEY, 2002). Assim como outras linguagens de marcação, XML lida com instruções embutidas no corpo de documentos chamadas *tags*, que permitem a descrição de dados. “XML tem como base linguagens mais antigas como SGML e HTML, sendo atualmente empregada na representação de dados estruturados e semiestruturados e seu intercâmbio na Web” (MELLO et al., 2000).

Ainda segundo Mello et al. (2000), o principal conceito dentro do XML é o elemento, que descreve uma unidade atômica ou não-atômica de dado. Existe a possibilidade de um ou mais elementos serem definidos antecipadamente por meio de um esquema, uma DTD, por exemplo, que determina um padrão para marcação de informações em documentos, definindo uma hierarquia de elementos, na qual um elemento é a origem desta hierarquia. Para que um documento XML esteja em conformidade com um esquema, somente os elementos e as estruturas de aninhamento entre elementos definidas no esquema são permitidos no corpo do documento, esta propriedade é validada por *parsers* XML. A Figura 2.1 mostra um exemplo de um arquivo DTD de descrição e um documento XML que segue esta descrição que foram utilizadas para um domínio de artigos científicos do evento SBBD.

<pre> &lt;!-- SBBB DTD versão 1.3 --&gt; &lt;!ELEMENT SBBB(artigo)&gt; &lt;!ELEMENT artigo (título, autor+, resumo, seção+)&gt; &lt;!ELEMENT autor (nomea, instituição+, email*)&gt; &lt;!ELEMENT seção (nomes, texto)&gt; &lt;!ELEMENT instituição (nomei, endereço?)&gt; &lt;!ELEMENT título (#PCDATA)&gt; &lt;!ELEMENT resumo (#PCDATA)&gt; ... &lt;!ATTLIST SBBB ano CDATA #REQUIRED       local CDATA "Rio"&gt; </pre>	<pre> &lt;?xml version="1.0" encoding="UTF-8" standalone="no" ?&gt; &lt;!DOCTYPE SBBB SYSTEM SBBB.dtd&gt; &lt;SBBB ano="2000"&gt; &lt;/artigo&gt; &lt;título&gt; Dados Semi-Estruturados &lt;/título&gt; &lt;autor&gt;   &lt;nomea&gt; Ronaldo Mello &lt;/nomea&gt;   &lt;instituição&gt;     &lt;nomei&gt; UFRGS &lt;/nomei&gt;     &lt;endereço&gt; Bento Gonçalves, 9500     &lt;/endereço&gt;   &lt;/instituição&gt;   &lt;instituição&gt;     &lt;nomei&gt; UFSC &lt;/nomei&gt;     &lt;endereço&gt;       &lt;email&gt; ronaldo@inf.ufrgs.br &lt;/email&gt;       &lt;email&gt; ronaldo@inf.ufsc.br &lt;/email&gt;     &lt;/endereço&gt;   &lt;/instituição&gt;   &lt;autor&gt;     &lt;nomea&gt; Carlos Heuser &lt;/nomea&gt;     &lt;instituição&gt;       &lt;nomei&gt; UFRGS &lt;/nomei&gt;       &lt;endereço&gt; Bento Gonçalves, 9500       &lt;/endereço&gt;     &lt;/instituição&gt;     &lt;email&gt; heuser@inf.ufrgs.br &lt;/email&gt;   &lt;/autor&gt;   &lt;resumo&gt; ... &lt;/resumo&gt;   &lt;seção&gt;     &lt;nomes&gt; Introdução &lt;/nomes&gt;     &lt;texto&gt; ... &lt;/texto&gt;   &lt;/seção&gt;   ... &lt;/artigo&gt; ... &lt;/SBBB&gt; </pre>
---	---

(a)

(b)

Figura 2.1 – (a) Definição de DTD, (b) Documento XML que segue a definição.

Fonte: (MELLO et al., 2000)

O surgimento do XML como um padrão para a representação de dados na Internet teve como objetivo facilitar a publicação em meios eletrônicos, por prover uma sintaxe simples, legível para computadores e seres humanos, afirma Almeida (2002).

## 2.2.2 Estrutura do Currículo Lattes

A estrutura dos currículos presentes na base de dados do Lattes é disponibilizada abertamente por meio de um arquivo no formato DTD<sup>2</sup>, (LATTES, 2010). As informações do currículo estão estruturadas de forma hierárquica. Os currículos da plataforma estão divididos em módulos e cada um deles contém inúmeros elementos, (IPA, 2016). Os módulos são: apresentação, dados gerais, produção bibliográfica, produção técnica, orientações, produção cultural, eventos, bancas e indicadores. Cada um desses elementos serão brevemente expandidos a seguir.

A apresentação é o módulo introdutório do esquema, em que é apresentado um resumo do currículo do usuário, gerado de maneira automática pela aplicação de acordo com as informações inseridas pelo mesmo. Este resumo pode ser editado pelo usuário caso ele deseje.

No módulo de dados gerais, são agrupadas informações básicas de caráter pessoal do usuário. Sendo assim, há o armazenamento dos dados de identificação, endereços, formação acadêmica e complementar, atuação profissional (linhas de pesquisa e projetos), áreas de atuação,

<sup>2</sup> <<http://lattes.cnpq.br/web/plataforma-lattes/extracao-de-dados/>> acessado em 03/06/2019

idiomas, prêmios e títulos honoríficos e um espaço adicional denominado outras informações relevantes, no qual o usuário adiciona informações não citadas que julgue pertinente.

O módulo denominado produção bibliográfica concentra toda a produção bibliográfica realizada por um usuário, artigos completos, livros e capítulos, textos em jornais ou revistas, comunicações em anais de congressos e periódicos, artigos aceitos para publicação, apresentação de trabalho, traduções, partituras, prefácio/posfácio, dentre outras publicações bibliográficas.

Produção técnica é um módulo que agrupa informações a respeito da produção técnica elaborada por um usuário. As subdivisões presentes dentro de produção técnica são softwares, produtos, cartas, mapas ou similares, processos técnicos, trabalhos técnicos, curso de curta duração ministrado, desenvolvimento de material didático ou instrucional, editoração, manutenção de obra artística, maquete, programa de rádio ou TV, relatório de pesquisa e outra produção técnica.

O módulo seguinte é o de orientações, que, como o próprio nome já sugere, está destinado a armazenar todas as orientações ou supervisões (concluídas ou em andamento) nas quais o usuário está envolvido. Evidentemente os itens presentes nesse módulo são: orientações e supervisões concluídas e orientações e supervisões em andamento.

Prosseguindo temos um módulo chamado de produção cultural, que reúne dados relacionados à produção artística ou cultural do usuário. Nesse módulo, existem os seguintes elementos: apresentação de obra artística, arranjo musical, composição musical, obra de artes visuais, sonoplastia, programas de rádio ou TV, outra produção artística/cultura e demais trabalhos.

O módulo de eventos agrupa informações a respeito de eventos que o usuário tenha participado. Esse módulo se encontra dividido da seguinte maneira: organização de evento, participação em bancas de trabalhos de conclusão, participação em bancas de comissões julgadoras, participações em eventos, congressos, e demais informações.

Existe ainda o módulo bancas, esse módulo armazena informações a respeito das bancas nas quais o usuário esteve presente. Existem duas subdivisões, participação em bancas de trabalhos de conclusão e participação em bancas de comissões julgadoras.

Para finalizar há o módulo indicadores, que agrupa informações a respeito de indicadores de relevância ou de dados a respeito de toda a produção do usuário. As áreas nas quais este módulo se subdivide são indicadores de relevância e indicadores de produção.

É importante ressaltar que o preenchimento ou não de alguns itens específicos em alguns currículos varia de acordo com a área de atuação de cada usuário cadastrado, ou no caso deste trabalho, de cada servidor. Dessa forma, um servidor que atue na área de Ciência da Computação, por exemplo, pode não apresentar informações relacionadas ao módulo de produção cultural, diferentemente de um outro docente que atue em áreas como Artes Cênicas e Música, por exemplo. É notório que a estrutura dos currículos presentes no Lattes é bem extensa, o que pode demandar

um tempo considerável para entendimento, extração e mapeamento da mesma em um banco de dados.

### 2.2.3 Modelagem de Banco de Dados

De acordo com Elmasri et al. (2005), vários modelos de dados costumam ser propostos e pode-se categorizá-los levando em consideração os conceitos usados para especificar a estrutura do banco de dados. Sendo assim, comprehende-se que existem modelagens de alto nível, ou modelos de dados conceituais, que apresentam pricípios que definem os dados da forma como os usuários os percebem e, além disso, existem também modelagens de mais baixo nível, também denominadas modelos de dados físicos, que possuem conceitos que retratam as particularidades de como as informações estão armazenadas em um sistema.

O modelo de entidade e relacionamentos proposto por Chen (1976) evoluiu muito com o passar dos anos desde o seu surgimento, se caracterizando por enfoques que o aproxima das concepções e do cenário de orientação a objetos. Esse modelo de dados conceitual utiliza conceitos como entidades, atributos e relacionamentos. Uma entidade retrata um conceito ou um objeto do mundo real, como um *projeto* ou um *professor*, que são descritos no banco de dados. Já o atributo equivale a alguma caracterísica ou propriedade relevante que auxilia na tarefa de descrição de uma entidade. Pode-se exemplificar com informações como *nome* e *CPF* do professor. O relacionamento é utilizado para associar uma ou mais entidades, como, por exemplo, pode-se ter um relacionamento que caracteriza a relação de um *professor* com um *projeto*, denominado *está vinculado*. Entidades, atributos e relacionamentos ainda apresentam variações de classificação que podem definir algum comportamento específico para essas características, como mostra a Figura 2.2. Na Figura 2.3 há um exemplo de um diagrama de entidade e relacionamento.

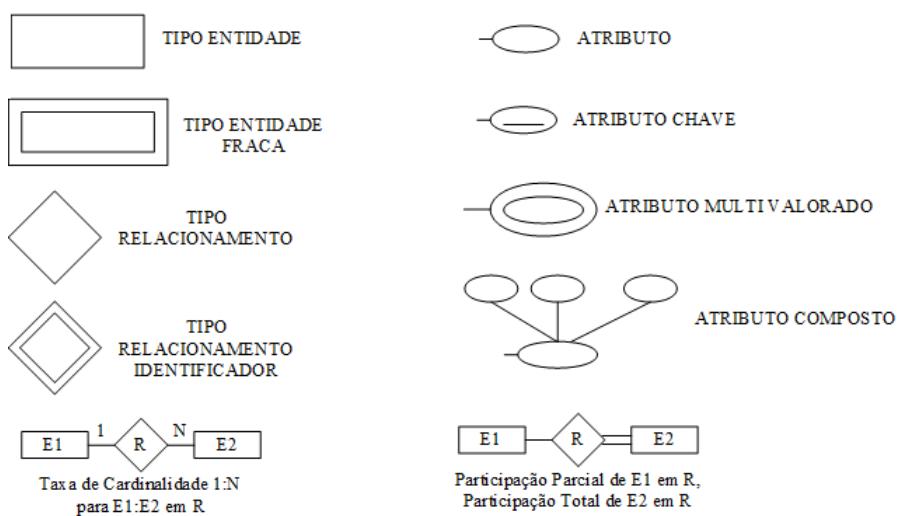


Figura 2.2 – Categorias de entidades, atributos e relacionamentos.

Fonte: (JARDIM, 2008)

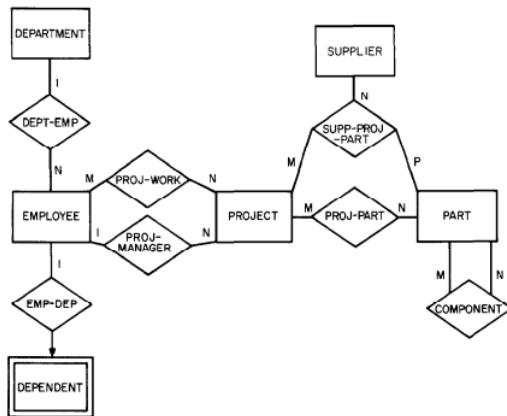


Figura 2.3 – Diagrama de entidade e relacionamento.

Fonte: ([CHEN, 1976](#))

Entre esses dois extremos, modelagem conceitual e modelagem física, está uma classe de modelos de dados representacionais (ou de implementação ou lógicos), que oferece os conceitos que podem ser entendidos pelos usuários finais, mas não excessivamente distantes da forma como os dados estão organizados dentro do computador, afirma [Elmasri et al. \(2005\)](#). O modelo lógico leva em consideração algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. O modelo lógico deve ser criado levando em conta o esquema de dados criado na modelagem conceitual e adota alguma tecnologia, pode ser: relacional, orientado a objetos, orientado a colunas, entre outros.

Um SGBD, de acordo com [Elmasri et al. \(2005\)](#), representa uma coleção de programas que permite aos usuários criar e manter um banco de dados. Logo, o SGBD compreende um sistema de software que permite e simplifica os processos de construção, definição, manipulação e compartilhamento de bancos de dados entre várias aplicações e usuários. Sendo assim, no modelo físico se escolhe qual SGBD será usado levando em consideração as limitações impostas pelo mesmo. Este modelo deve ser criado sempre com base no esquema de dados produzidos na modelagem lógica. Alguns exemplos de SGBDs são PostgreSQL, MySQL, MongoDB, entre outros.

## 2.2.4 Processo de Concepção de Software

Um processo é um conjunto de etapas parcialmente ordenadas constituídas de ações, atividades, tarefas, práticas e transformações realizadas na criação de algum artefato ou para alcançar um objetivo. Esse processo vai se construindo ao longo do tempo e o objetivo está relacionado a um ou mais resultados concretos no final. No contexto da Engenharia de Software, “um processo não é uma prescrição rígida de como desenvolver um software. Ao contrário, é uma abordagem adaptável que possibilita às pessoas (a equipe de software) realizar o trabalho de

selecionar e escolher o conjunto apropriado de ações e tarefas” (PRESSMAN; MAXIM, 2016).

Pode-se estabelecer processos para diversas tarefas como contratação, aquisição, desenvolvimento e manutenção de software. Além disso, subprocessos também podem ser definidos para definir esses processos, como por exemplo no desenvolvimento, que compreende tarefas como levantamento de requisitos, análise, *storyboard*, implementação e teste (FILHO, 2003). Portanto, é interessante, porém não obrigatório, que no processo de desenvolvimento de um produto de software, haja inicialmente a criação de um documento de especificação do mesmo, que levante aspectos relevantes e prescindíveis relacionados a ele, não impedindo que no decorrer no desenvolvimento, alterações sejam realizadas neste documento. Em seguida, no desenvolvimento propriamente dito, os responsáveis escolhem um modelo de ciclo de vida para implementação do produto, como exemplo há o modelo em cascata, o de prototipagem evolutiva, entrega por estágios e a metodologia ágil *Scrum*, muito utilizada no contexto atual. A Figura 2.4 ilustra o funcionamento dessa metodologia.

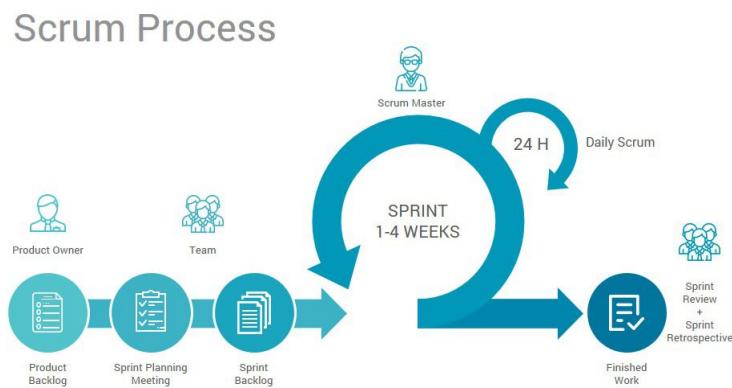


Figura 2.4 – Processo da metodologia ágil Scrum.

Fonte: (EDUCATION, 2018)

Na grande maioria dos casos, o processo de desenvolvimento de um software de sucesso depende extremamente da participação do cliente (ou um representante do mesmo) que está solicitando a aplicação, dentro do processo de desenvolvimento. Além disso, é interessante manter entregas ao cliente de liberações parciais do produto, pois, segundo Filho (2003), isso aumenta a visibilidade do projeto, o que geralmente é um fator muito importante no relacionamento com o cliente. O processo unificado de desenvolvimento de um software pode ser segmentado nas seguintes etapas: Concepção, Elaboração, Construção e Transição; não representando uma maneira exclusiva de se organizar esse processo. A Tabela 2.1 descreve as etapas desse processo. A fase de Elaboração é iniciada depois que o cliente aprova a proposta de especificação.

Tabela 2.1 – Processo unificado de desenvolvimento de um software.

Fase	Descrição
<b>Concepção</b>	Fluxo que visa obter um conjunto de requisitos de um produto, acordado entre cliente e fornecedor.
<b>Elaboração</b>	Fase na qual o produto é detalhado o suficiente para permitir um planejamento acurado da fase de construção.
<b>Construção</b>	Fase na qual é produzida uma versão completamente operacional do produto.
<b>Transição</b>	Fase na qual o produto é colocado à disposição de uma comunidade de usuários.

# 3 Desenvolvimento

Neste capítulo, é descrito e detalhado os passos do desenvolvimento do UFOP Ativa. Ele está subdividio em seis seções. As seções 3.1 e 3.2, abordam respectivamente a modelagem do banco de dados e a parte de concepção do software. Nas Seções 3.3 e 3.4 detalha-se a extração dos dados e o mapeamento para o banco do sistema. Ao final, a Seção 3.5 trata da parte de deduplicação das produções e a Seção 3.6 do desenvolvimento do sistema.

É importante ressaltar que as tecnologias utilizadas no desenvolvimento deste sistema são: Java Spring Boot<sup>1</sup>, Angular<sup>2</sup> e PostegreSQL<sup>3</sup> (demais ferramentas utilizadas são detalhadas ao longo do capítulo). A escolha das tecnologias baseia-se na premissa de que o NTI da UFOP futuramente dará suporte ao sistema desenvolvido e estas são as tecnologias nas quais eles constroem a maioria dos sistemas.

O Spring Boot, utilizado no *backend*, é um *framework* para a plataforma Java proposto por Johnson (2004), que surgiu para simplificar o processo de configuração e publicação de aplicações. O objetivo é que os projetos estejam em funcionamento o mais rápido possível sem causar muitas dificuldades. Essa tecnologia cumpre esse propósito favorecendo a convenção sobre a configuração, ou seja, o desenvolvedor necessita apenas escolher quais módulos deseja utilizar (Template, Persistência, WEB, Segurança, etc.) que ele já faz o trabalho de reconhecer e configurar. Porém, por mais que ele já deixe configurado, nada impede que o projetista customize da forma que necessitar e preferir.

Angular trata-se de um *framework* e plataforma para construção da interface (*frontend*) de aplicações utilizando CSS, HTML e principalmente *JavaScript* que foi desenvolvido pela Google (2016). Esse framework apresenta alguns elementos básicos que tornam essa construção interessante como templates, módulos, serviços, componentes e ferramentas de infraestrutura que automatizam tarefas, como a realização de testes unitários de um projeto de software.

O PostgreSQL é um SGBD objeto-relacional que é disponibilizado abertamente para uso como um projeto de código aberto. Ele possui mais de 30 anos de desenvolvimento ativo que lhe garantiu uma forte reputação de confiabilidade, robustez de recursos e desempenho.

## 3.1 Modelagem do Banco de Dados

Nesta seção, são descritas de maneira sucinta cada tipo de entidade presente no esquema preliminar do banco de dados e seus atributos. O esquema conta com 31 entidades, mais de 100 atributos e 16 relacionamentos. A modelagem foi elaborada levando em consideração os aspectos

---

<sup>1</sup> <<https://spring.io/>> acessado em 05/06/2019

<sup>2</sup> <<https://angular.io/>> acessado em 05/06/2019

<sup>3</sup> <<https://www.postgresql.org/>> acessado em 05/06/2019

levantados pela PROPP e as informações que interessam a mesma. Na subseção 3.1.1 há uma descrição do esquema conceitual do banco. A figura 3.1 ao final da seção mostra o diagrama de entidade e relacionamento do banco, os atributos foram ocultados na representação para facilitar a visualização.

### 3.1.1 Dicionário de Dados

#### 3.1.1.1 Aluno

Tipo de entidade que representa um aluno por meio do atributo composto *curso* que é subdividido em *nome* e *nivel*. Se relaciona com a entidade *Servidor*.

- **curso**: atributo composto que representa um curso.
- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia o curso do aluno.
- **nivel**: atributo textual simples, monovalorado e armazenado, que classifica o curso do aluno (Graduação, Mestrado, Doutorado, Pós-Doutorado).

#### 3.1.1.2 Area

Tipo de entidade que representa a área de atuação de uma pessoa por meio dos atributos *grande\_area*, *nome\_area*, *sub\_area*, *especialidade* e *id* (chave). Se relaciona com a entidade *Pessoa*.

- **grande\_area**: atributo textual simples, monovalorado e armazenado, que descreve uma área.
- **nome\_area**: atributo textual simples, monovalorado e armazenado, que nomeia uma área.
- **sub\_area**: atributo textual simples, monovalorado e armazenado, que nomeia uma subárea.
- **especialidade**: atributo textual simples, monovalorado e armazenado, que nomeia uma especialidade dentro da área.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente uma área.

#### 3.1.1.3 Arte\_Cenica

Tipo de entidade derivada da entidade *Producao\_Artistica\_Cultural* que representa uma arte cênica produzida por uma pessoa por meio dos atributos *data\_estreia*, *local\_estreia*, *data\_encerramento*, *tipo\_evento* e um atributo composto *obra\_referencia* que é representado por *nome*, *ano* e *autor*.

- **data\_estreia**: atributo simples, monovalorado e armazenado, que representa a data de estreia da arte cênica.
- **local\_estreia**: atributo textual simples, monovalorado e armazenado, que nomeia o local de estreia da arte cênica.
- **data\_encerramento**: atributo simples, monovalorado e armazenado, que representa a data de encerramento da arte cênica.
- **tipo\_evento**: atributo textual simples, monovalorado e armazenado, que descreve o tipo do evento de apresentação da arte cênica.
- **obra\_referencia**: atributo composto que representa uma obra referência para a arte cênica.
- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia a obra referência.
- **ano**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano da obra referência.
- **autor**: atributo textual simples, monovalorado e armazenado, que nomeia o autor da obra referência.

#### 3.1.1.4 Arte\_Visual

Tipo de entidade derivada da entidade *Producao\_Artistica\_Cultural* que representa uma arte visual produzida por uma pessoa por meio dos atributos específicos *atividade\_autores* e *cidade\_evento*.

- **atividade\_autores**: atributo textual simples, monovalorado e armazenado, que descreve a atividade dos autores na arte visual.
- **cidade\_evento**: atributo textual simples, monovalorado e armazenado, que nomeia a cidade do evento de exibição da arte visual.

#### 3.1.1.5 Artigo

Tipo de entidade derivada da entidade *Producao\_Bibliografica* que representa um artigo produzido por uma pessoa por meio dos atributos *titulo\_periodico\_revista*, *situacao\_de\_publicacao*, *issn*, *pagina\_inicial* e *pagina\_final*.

- **titulo\_periodico\_revista**: atributo textual simples, monovalorado e armazenado, que enumera o periódico/revista de divulgação do artigo.
- **situacao\_de\_publicacao**: atributo simples, monovalorado e armazenado, que representa a situação da publicação do artigo (publicado, aceito para publicação).

- **issn**: atributo textual, simples, monovalorado e armazenado, que representa o ISSN do artigo.
- **pagina\_inicial**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a página inicial do artigo.
- **pagina\_final**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a página final do artigo.

### 3.1.1.6 Atuacao\_Profissional

Tipo de entidade que representa às informações a respeito de uma atuação profissional por meio dos atributos *ano\_inicio*, *ano\_fim*, *descricao*, *tipo* e *id* (chave). Se relaciona com a entidade *Pessoa*.

- **descricao**: atributo textual simples, monovalorado e armazenado, que descreve a atuação profissional.
- **tipo**: atributo textual simples, monovalorado e armazenado, que representa o tipo de atuação profissional.
- **ano\_inicio**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de início da atuação profissional.
- **ano\_fim**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de fim da atuação profissional.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente uma atuação profissional.

### 3.1.1.7 Autor

Tipo de entidade que representa as informações a respeito de um autor que trabalhou junto a uma pessoa em alguma produção, através dos atributos *nome\_citacao*, *nome\_completo*, *ordem\_autoria*, *id\_lattes* e *id* (chave). Se relaciona com a entidade *Producao*.

- **nome\_citacao**: atributo textual simples, multivalorado e armazenado, que descreve os nomes em citações do autor.
- **nome\_completo**: atributo textual simples, monovalorado e armazenado, que nomeia o autor.
- **ordem\_autoria**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a ordem de autoria em relação a produção.

- **id\_lattes**: atributo textual simples, monovalorado e armazenado, que representa o identificador do lattes do autor.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identificaunicamente um autor.

### 3.1.1.8 Autor\_Externo

Tipo de entidade que representa as informações a respeito de um autor externo à instituição que trabalhou junto a uma pessoa em alguma produção, através dos atributos *nome\_citacao*, *nome*, *link\_lattes*, *id\_lattes* e *id* (chave). Se relaciona com a entidade *Producao*.

- **nome\_citacao**: atributo textual simples, multivalorado e armazenado, que descreve os nomes em citações do autor externo.
- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia o autor externo.
- **link\_lattes**: atributo textual simples, monovalorado e armazenado, que representa o link do lattes do autor externo (caso exista).
- **id\_lattes**: atributo textual simples, monovalorado e armazenado, que representa o identificador do lattes do autor externo (caso exista).
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identificaunicamente um autor externo.

### 3.1.1.9 Capitulo\_de\_Livro

Tipo de entidade derivada da entidade *Producao\_Bibliografica* que representa um capítulo de livro produzido por uma pessoa por meio dos atributos *livro* (composto por *titulo* e *isbn*), *pagina\_inicial* e *pagina\_final*.

- **livro**: atributo composto que representa o livro no qual o capítulo está inserido.
- **titulo**: atributo textual simples, monovalorado e armazenado, que entitula o livro.
- **isbn**: atributo textual simples, monovalorado e armazenado, que representa o ISBN do livro.
- **pagina\_inicial**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a página inicial do capítulo.
- **pagina\_final**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a página final do capítulo.

### 3.1.1.10 Departamento

Tipo de entidade que representa um departamento por meio dos atributos *nome*, *sigla* e *id* (chave). Se relaciona com as entidades *Unidade* e *Servidor*.

- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia o departamento.
- **sigla**: atributo textual simples, monovalorado e armazenado, que representa a sigla do departamento.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente um departamento.

### 3.1.1.11 Financiador

Tipo de entidade que representa um financiador de um projeto de pesquisa por meio dos atributos *nome*, *codigo* e *id* (chave). Se relaciona com a entidade *Projeto\_de\_Pesquisa*.

- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia o financiador.
- **codigo**: atributo textual simples, monovalorado e armazenado, que representa o código do financiador.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente um financiador.

### 3.1.1.12 Formacao\_Academica

Tipo de entidade que apresenta informações a respeito de uma formação acadêmica por intermédio dos atributos *tipo*, *curso*, *instituicao*, *ano\_inicio*, *ano\_conclusao*, *titulo\_trabalho*, *situacao* e *id* (chave). Se relaciona com a entidade *Pessoa*.

- **tipo**: atributo textual simples, monovalorado e armazenado, que representa o tipo da formação acadêmica (Graduação, Mestrado, Doutorado, Pós-Doutorado e Especialização).
- **curso**: atributo textual simples, monovalorado e armazenado, que nomeia o curso no qual se deu a formação acadêmica.
- **instituicao**: atributo textual simples, monovalorado e armazenado, que armazena a instituição na qual se deu a formação acadêmica.
- **ano\_inicio**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de início da formação acadêmica.
- **ano\_conclusao**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de conclusão da formação acadêmica.

- **titulo\_trabalho**: atributo textual simples, monovalorado e armazenado, que intitula o trabalho de conclusão da formação acadêmica.
- **situacao**: atributo simples, monovalorado e armazenado, que representa a situação da formação acadêmica (em andamento, concluída).
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente uma formação acadêmica.

### 3.1.1.13 Livro

Tipo de entidade derivada da entidade *Producao\_Bibliografica* que representa um livro produzido por uma pessoa por meio dos atributos *editora*, *edicao\_revisao* e *isbn*.

- **editora**: atributo textual simples, monovalorado e armazenado, que nomeia a editora do livro.
- **edicao\_revisao**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a edição do livro.
- **isbn**: atributo textual simples, monovalorado e armazenado, que representa o ISBN do livro.

### 3.1.1.14 Musica

Tipo de entidade derivada da entidade *Producao\_Artistica\_Cultural* que representa uma música produzida por uma pessoa por meio dos atributos *data\_estreia*, *local\_estreia*, *tipo\_evento* e um atributo composto *obra\_referencia* que é representado por *nome*, *ano* e *autor*.

- **data\_estreia**: atributo simples, monovalorado e armazenado, que representa a data de estreia da música.
- **local\_estreia**: atributo textual simples, monovalorado e armazenado, que nomeia o local de estreia da música.
- **tipo\_evento**: atributo textual simples, monovalorado e armazenado, que descreve o tipo do evento de apresentação da música.
- **obra\_referencia**: atributo composto que representa uma obra referência para a música.
- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia a obra referência.
- **ano**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano da obra referência.
- **autor**: atributo textual simples, monovalorado e armazenado, que nomeia o autor da obra referência.

### 3.1.1.15 Orientacao\_Concluida

Tipo de entidade que representa uma orientação concluída por um servidor por meio dos atributos *nome\_financiador*, *tipo*, *instituicao*, *ano*, *doi*, um atributo composto *titulo* que é representado por *portugues* e *ingles*, além do *id* (chave). Se relaciona com a entidade *Servidor*.

- **nome\_financiador:** atributo textual simples, monovalorado e armazenado, que nomeia o financiador do trabalho referente a orientação.
- **tipo:** atributo textual simples, monovalorado e armazenado, que representa o tipo da orientação (Mestrado, Doutorado, Pós-Doutorado e Outro).
- **instituicao:** atributo textual simples, monovalorado e armazenado, que nomeia a instituição relacionada ao trabalho referente a orientação.
- **ano:** atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de conclusão da orientação.
- **doi:** atributo textual simples, monovalorado e armazenado, que descreve o doi (caso exista) referente ao trabalho relacionado a orientação.
- **titulo:** atributo composto que representa o título do trabalho orientado.
- **portugues:** atributo textual simples, monovalorado e armazenado, que representa o título em português.
- **ingles:** atributo textual simples, monovalorado e armazenado, que representa o título em inglês.
- **id (chave primária):** atributo numérico simples, monovalorado e armazenado, que identifica unicamente uma orientação concluída.

### 3.1.1.16 Palavra\_Chave

Tipo de entidade que representa uma palavra chave utilizada por uma pessoa por meio dos atributos *palavra*, *numero\_utilizacao* e *id* (chave). Se relaciona com a entidade *Pessoa*.

- **palavra:** atributo textual simples, monovalorado e armazenado, que representa a palavra chave.
- **numero\_utilizacao:** atributo numérico inteiro, simples, monovalorado e armazenado, que representa o numero de utilizações da palavra chave.
- **id (chave primária):** atributo numérico simples, monovalorado e armazenado, que identifica unicamente uma palavra chave.

### 3.1.1.17 Patente

Tipo de entidade derivada da entidade *Producao\_Tecnica* que representa uma patente produzida por uma pessoa por meio do atributo *categoria*.

- **categoria:** atributo textual simples, monovalorado e armazenado, que nomeia a categoria da patente.

### 3.1.1.18 Pessoa

Tipo de entidade que representa uma pessoa da instituição através dos atributos *cpf*, *nome\_completo*, *nacionalidade*, *link\_lattes*, *link\_foto*, *link\_perfil*, *nome\_citacoes* (multivalorado), *sexo*, *raca*, *id\_lattes*, *data\_atualizacao* e *id* (chave). Se relaciona com as entidades *Projeto\_de\_Pesquisa*, *Area*, *Formacao\_Academica*, *Producao*, *Atuacao\_Profissional* e *Vinculo\_Institucional*.

- **cpf:** atributo textual simples, monovalorado e armazenado que representa o CPF de uma pessoa.
- **nome\_completo:** atributo textual simples, monovalorado e armazenado, que nomeia a pessoa.
- **nacionalidade:** atributo textual simples, monovalorado e armazenado, que define a nacionalidade da pessoa.
- **link\_lattes:** atributo textual simples, monovalorado e armazenado, que armazena o link do lattes da pessoa.
- **link\_foto:** atributo simples, monovalorado e armazenado, que armazena o link da foto da pessoa no lattes.
- **link\_perfil:** atributo textual simples, monovalorado e armazenado, que armazena o link do perfil da pessoa na plataforma.
- **nome\_citacoes:** atributo textual simples, multivalorado e armazenado, que define os nomes em citações da pessoa.
- **sexo:** atributo textual simples, monovalorado e armazenado que armazena o sexo da pessoa.
- **raca:** atributo textual simples, monovalorado e armazenado, que define a raça da pessoa.
- **id\_lattes:** atributo textual simples, monovalorado e armazenado, que representa o identificador do lattes da pessoa.
- **data\_atualizacao:** atributo simples, monovalorado e armazenado, que define a data de atualização do lattes da pessoa.

- **id (chave primária)**: atributo numérico simples, monovalorado e armazenado, que identifica unicamente uma pessoa.

### 3.1.1.19 Producao

Tipo de entidade que representa as produções de uma pessoa por meio dos atributos *titulo*, *ano*, *pais* e *id* (chave). Se relaciona com as entidades *Pessoa*, *Autor* e *Autor\_Externo* e possui especialização para as entidades *Producao\_Bibliografica*, *Producao\_Tecnica* e *Producao\_Artistica\_Cultural*.

- **titulo**: atributo textual simples, monovalorado e armazenado, que representa o título da produção.
- **ano**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano da produção.
- **pais**: atributo textual simples, monovalorado e armazenado, que se refere ao país da produção.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente uma produção.

### 3.1.1.20 Producao\_Artistica\_Cultural

Tipo de entidade que é uma subclasse da entidade *Producao*, apresenta como atributos específicos *temporada*, *local\_evento*, *instituicao\_promotora\_evento* e o atributo composto *premicao*, subdividido em *nome* e *instituicao\_promotora*. Apresenta especialização para as entidades *Arte\_Cenica*, *Arte\_Visual* e *Musica*.

- **temporada**: atributo textual simples, monovalorado e armazenado, que armazena a temporada da produção.
- **local\_evento**: atributo textual simples, monovalorado e armazenado, que nomeia o local do evento da produção.
- **instituicao\_promotora\_evento**: atributo textual simples, monovalorado e armazenado, que nomeia a instituição promotora do evento da produção.
- **premicao**: atributo composto que representa a premiação recebida pela produção.
- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia a premiação.
- **instituicao\_promotora**: atributo textual simples, monovalorado e armazenado, que nomeia a instituição promotora da premiação.

### 3.1.1.21 Producao\_Bibliografica

Tipo de entidade que é uma subclasse da entidade *Producao* e representa as produções bibliográficas através dos atributos específicos *doi*, *local* e *numero\_paginas*. Apresenta especialização para as entidades *Trabalho\_em\_Evento*, *Livro*, *Capitulo\_de\_Livro* e *Artigo*.

- **doi**: atributo textual simples, monovalorado e armazenado, que se refere ao doi da produção bibliográfica, caso ela possua.
- **local**: atributo textual simples, monovalorado e armazenado, que se refere ao local de publicação da produção bibliográfica.
- **numero\_paginas**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o número de páginas da produção bibliográfica.

### 3.1.1.22 Producao\_Tecnica

Tipo de entidade que é uma subclasse da entidade *Producao*, apresenta como atributos específicos *instituicao\_financiadora* e *finalidade*. Apresenta especialização para as entidades *Software*, *Produto\_Tecnologico*, *Trabalho\_Tecnico* e *Patente*.

- **instituicao\_financiadora**: atributo textual simples, monovalorado e armazenado, que nomeia a instituição financiadora da produção.
- **finalidade**: atributo textual simples, monovalorado e armazenado, que descreve a finalidade da produção.

### 3.1.1.23 Produto\_Tecnologico

Tipo de entidade que é uma subclasse da entidade *Producao\_Tecnica*, apresenta como atributo específico *disponibilidade*.

- **disponibilidade**: atributo textual simples, monovalorado e armazenado, que descreve a disponibilidade do produto.

### 3.1.1.24 Programa\_Pos\_Graduacao

Tipo de entidade que representa um programa de pós-graduação por meio dos atributos *nome* e *id* (chave). Se relaciona com a entidade *Servidor*.

- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia um programa de pós-graduação.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente um programa de pós-graduação.

### 3.1.1.25 Projeto\_de\_Pesquisa

Tipo de entidade que representa um projeto de pesquisa realizado por uma pessoa através dos atributos *situacao*, *ano\_inicio*, *ano\_conclusao*, *titulo*, *descricao* e *id* (chave). Se relaciona com as entidades *Pessoa* e *Financiador*.

- **situacao:** atributo textual simples, monovalorado e armazenado, que representa a situação do projeto de pesquisa (em andamento ou concluído).
- **ano\_inicio:** atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de inicio do projeto de pesquisa.
- **ano\_conclusao:** atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de conclusão do projeto de pesquisa.
- **titulo:** atributo textual simples, monovalorado e armazenado, que intitula o projeto de pesquisa.
- **descricao:** atributo textual simples, monovalorado e armazenado, que descreve o projeto de pesquisa.
- **id (chave primária):** atributo numérico inteiro, simples, monovalorado e armazenado, que identifica unicamente um projeto de pesquisa.

### 3.1.1.26 Servidor

Tipo de entidade que representa um servidor da instituição através dos atributos *tipo* e *ppgs*. Se relaciona com as entidades *Aluno*, *Unidade*, *Departamento*, *Orientacao\_Concluida* e *Programa\_Pos\_Graduacao*.

- **tipo:** atributo simples, monovalorado e armazenado, que descreve o tipo de servidor (Técnico ou Docente).
- **ppgs:** atributo simples, monovalorado e armazenado, que representa o(s) Programa(s) de Pós-Graduação do servidor.

### 3.1.1.27 Software

Tipo de entidade derivada da entidade *Producao\_Tecnologica* que representa um software produzido por uma pessoa através dos atributos *plataforma*, *ambiente* e *disponibilidade*.

- **plataforma:** atributo textual simples, monovalorado e armazenado, que descreve a plataforma do software.

- **ambiente**: atributo textual simples, monovalorado e armazenado, que descreve o ambiente de desenvolvimento do software.
- **disponibilidade**: atributo textual simples, monovalorado e armazenado, que descreve a disponibilidade do software.

### 3.1.1.28 Trabalho\_em\_Evento

Tipo de entidade derivada da entidade *Producao\_Bibliografica* que representa um trabalho em evento produzido por uma pessoa através dos atributos *cidade\_evento*, *titulo\_anais\_proceedings*, *volume*, *fasciculo*, *pagina\_inicial*, *pagina\_final*, *editora*, *isbn* e *nome\_evento* (composto por *portugues* e *ingles*).

- **cidade\_evento**: atributo textual simples, monovalorado e armazenado, que nomeia a cidade do evento de publicação do trabalho.
- **titulo\_anais\_proceedings**: atributo textual simples, monovalorado e armazenado, que entitula o anais ou proceedings de divulgação do trabalho.
- **pagina\_inicial**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a página inicial do trabalho.
- **pagina\_final**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa a página final do trabalho.
- **editora**: atributo textual simples, monovalorado e armazenado, que nomeia a editora do trabalho.
- **isbn**: atributo textual simples, monovalorado e armazenado, que representa o ISBN do trabalho.
- **nome\_evento**: atributo composto que nomeia o evento de publicação do trabalho.
- **portugues**: atributo textual simples, monovalorado e armazenado, que nomeia o evento em português.
- **ingles**: atributo textual simples, monovalorado e armazenado, que nomeia o evento em inglês.

### 3.1.1.29 Trabalho\_Tecnico

Tipo de entidade derivada da entidade *Producao\_Tecnica* que representa um trabalho técnico produzido por uma pessoa através dos atributos *cidade*, *duracao*, *numero\_paginas* e *disponibilidade*.

- **cidade**: atributo textual simples, monovalorado e armazenado, que nomeia a cidade de publicação do trabalho.
- **duracao**: atributo textual simples, monovalorado e armazenado, que representa a duração do trabalho.
- **numero\_paginas**: atributo numérico simples, monovalorado e armazenado, que representa o número de páginas do trabalho.
- **disponibilidade**: atributo textual simples, monovalorado e armazenado, que representa a disponibilidade do trabalho.

### 3.1.1.30 Unidade

Tipo de entidade que representa uma unidade por meio dos atributos *nome*, *sigla* e *id* (chave). Se relaciona com as entidades *Servidor* e *Departamento*.

- **nome**: atributo textual simples, monovalorado e armazenado, que nomeia a unidade.
- **sigla**: atributo textual simples, monovalorado e armazenado, que representa a sigla da unidade.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identificaunicamente uma unidade.

### 3.1.1.31 Vinculo\_Institucional

Tipo de entidade que representa um vínculo institucional com a universidade por meio dos atributos *ano\_inicio*, *ano\_fim*, *enquadramento\_funcional* e *id* (chave). Se relaciona com a entidade *Pessoa*.

- **ano\_inicio**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de início do vínculo.
- **ano\_fim**: atributo numérico inteiro, simples, monovalorado e armazenado, que representa o ano de fim do vínculo.
- **enquadramento\_funcional**: atributo textual simples, monovalorado e armazenado, que representa o enquadramento funcional.
- **id (chave primária)**: atributo numérico inteiro, simples, monovalorado e armazenado, que identificaunicamente um vínculo.

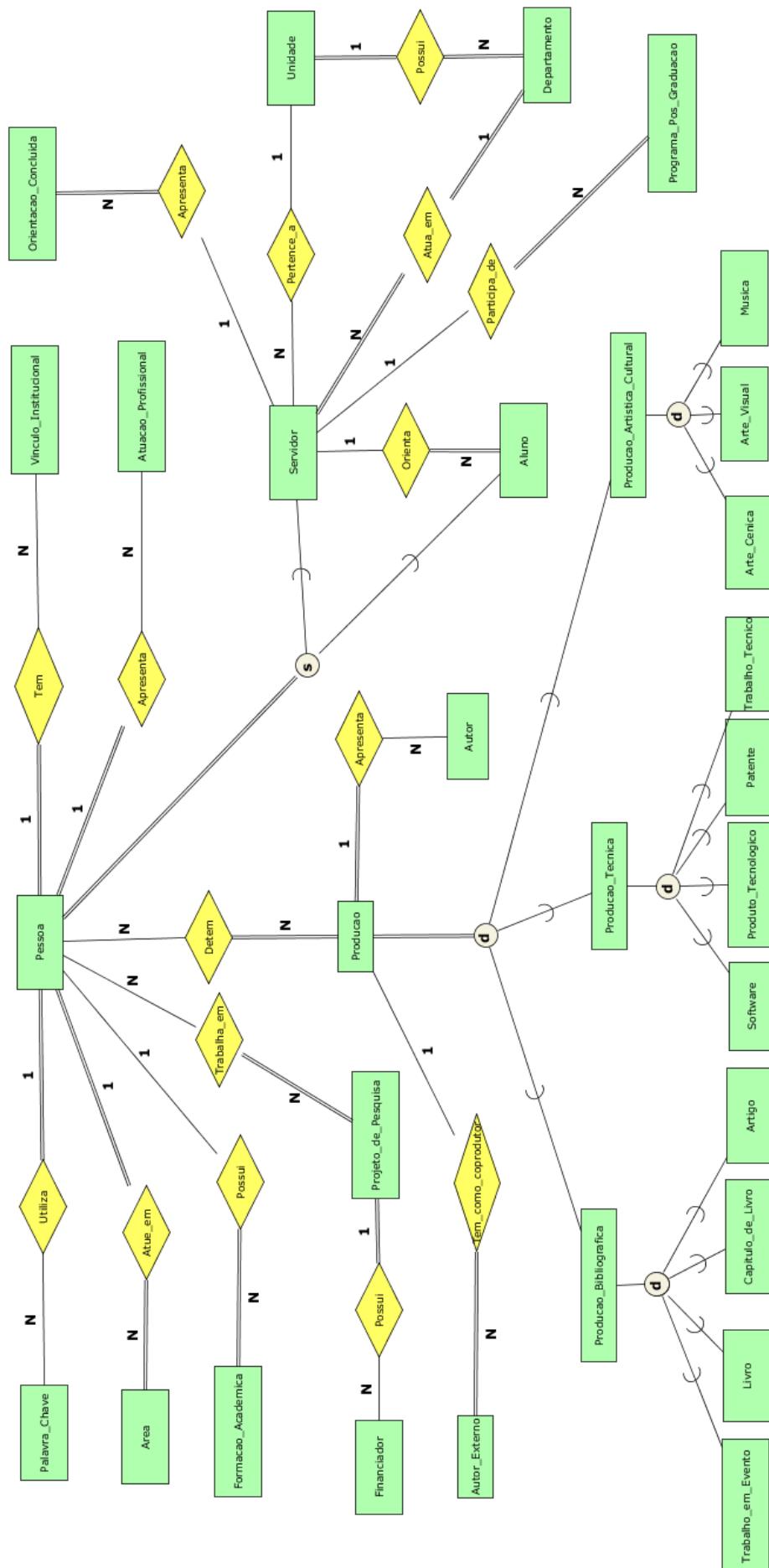


Figura 3.1 – Diagrama de Entidade e Relacionamento.

## 3.2 Concepção do Software

Descreve-se a concepção do software ao longo das dez subseções presentes nesta seção. Respectivamente são esclarecidos objetivo, missão e descrição, modelo de negócio, escopo positivo e negativo, diagrama de contexto, hierarquia de atores, protótipo, *backlog*, estórias de usuário e cenários de teste.

### 3.2.1 Objetivo

Esta concepção especifica um software *Web* para acessar os dados, provenientes da Plataforma Lattes, sobre os servidores atuantes na UFOP, bem como relatórios estatísticos acerca desses dados.

### 3.2.2 Missão e Descrição do Produto

#### 3.2.2.1 Nome do Produto e Componentes

O software chama-se UFOP Ativa. O sistema divide-se em 2 componentes, que dependem do ator que está utilizando o sistema (Usuário Visitante, Administrador), ou seja, no sistema implantado o Usuário Visitante tem sua interface de acordo com suas permissões e o Administrador de acordo com as dele.

#### 3.2.2.2 Missão do Produto

Auxiliar a universidade e a PROPP na gestão, divulgação, organização e acompanhamento dos projetos, pesquisas, produções e demais atividades realizados por servidores que atuam na mesma. Desta maneira, promover o aumento dos interesses voltados para as atividades realizadas na UFOP.

#### 3.2.2.3 Descrição do Produto

O produto é um sistema *Web* que oferece serviços de exibição de informações e relatórios dos servidores, unidades, departamentos e programas de pós-graduação relacionados a UFOP, bem como os projetos, pesquisas e demais atividades realizadas pelos mesmos.

### 3.2.3 Modelo de Negócio

A princípio, não existe um planejamento em relação a monetização do software, por se tratar de uma ferramenta institucional cuja solicitação foi realizada pela PROPP. A depender das funcionalidades e do desempenho do software, futuramente pode-se analisar essa possibilidade.

### 3.2.4 Escopo Positivo e Negativo

#### 3.2.4.1 Positivo

O sistema permite aos usuários consultar e visualizar dados que são extremamente importantes, principalmente para avaliações internas da universidade, garantindo um esclarecimento mais palpável em relação às pesquisas científicas, formação de recursos humanos, especialidades, dentre outras informações relevantes a nível institucional.

#### 3.2.4.2 Negativo

O sistema não permite aos usuários visualizar informações acerca de redes de coautoria e dados completos sobre as especialidades de um servidor. Além disso, não realizou-se ainda testes automatizados e avaliação de UX na aplicação. Existem também adaptações a serem feitas no que se refere a questão de segurança.

### 3.2.5 Diagrama de Contexto

A Figura 3.2 apresenta o diagrama de contexto da aplicação, com seus respectivos atores e serviços. A Tabela 3.1 exibe a descrição dos serviços.

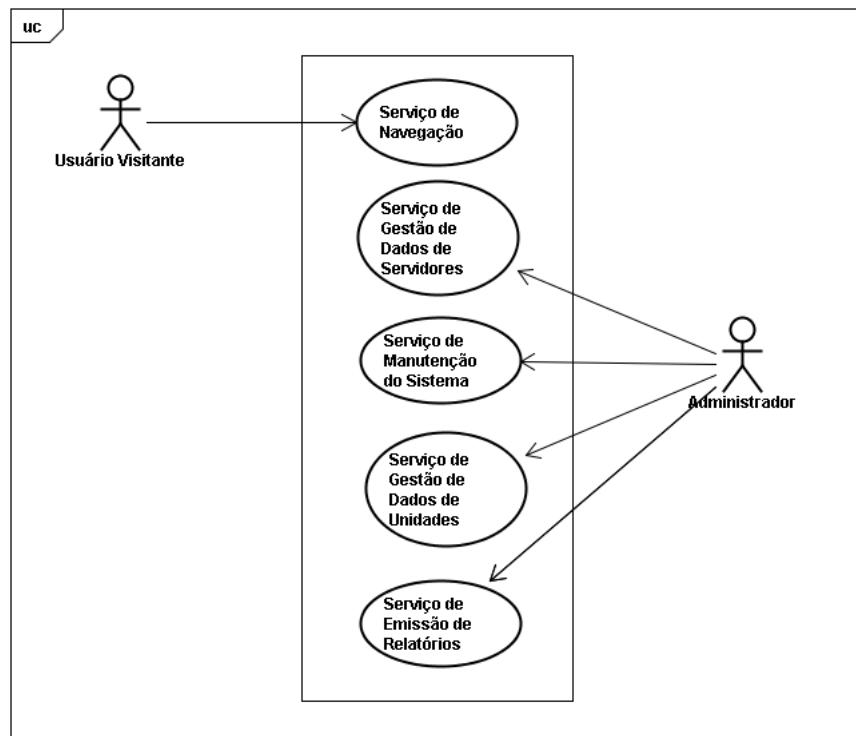


Figura 3.2 – Diagrama de Contexto da Aplicação.

Tabela 3.1 – Descrição do Diagrama de Contexto.

Serviço	Descrição
<b>Serviço de Navegação</b>	Permite ao usuário visitante pesquisar nomes ou palavras-chave no sistema e navegar pelas abas da aplicação.
<b>Serviço de Gestão de Dados de Servidores</b>	Possibilita ao administrador gerir os dados da aplicação relacionados aos servidores e programas de pós-graduação nos quais eles participam, realizando atualizações, inserindo e removendo informações.
<b>Serviço de Gestão de Dados de Unidades</b>	Possibilita ao administrador gerir os dados da aplicação relacionados as unidades e departamentos pertencentes as mesmas, realizando atualizações, inserindo e removendo informações.
<b>Serviço de Manutenção do Sistema</b>	Permite ao administrador realizar a manutenção do sistema realizando atualizações, fazendo backups e restaurando-o se necessário. Permite também que este dê ou remova permissões aos usuários.
<b>Serviço de Emissão de Relatórios</b>	Permite ao administrador e ao sistema fazer levantamentos em relação às informações armazenadas bem como gerar relatórios estatísticos baseados nas mesmas.

### 3.2.6 Hierarquia de Atores

A Figura 3.3 apresenta a hierarquia dos atores da aplicação, mostrando que o Administrador pode realizar tanto as suas tarefas quanto as tarefas realizadas pelo Usuário Visitante. A Tabela 3.2 expõe a descrição de cada ator.

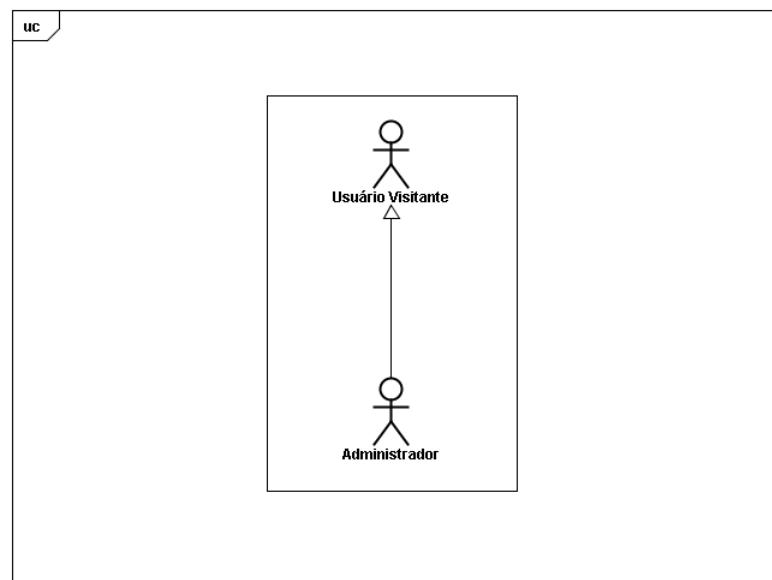


Figura 3.3 – Hierarquia de Atores da Aplicação.

Tabela 3.2 – Descrição da Hierarquia de Atores.

Ator	Descrição
<b>Usuário Visitante</b>	Qualquer pessoa física interessada em utilizar o sistema, mas principalmente pessoas interessadas nas atividades realizadas na UFOP.
<b>Administrador</b>	Alguma pessoa específica que seja responsável pela aplicação ou alguém em quem a mesma confia, que tenha habilidade para dar manutenção no sistema.

### 3.2.7 Protótipo

As telas apresentadas nesta subseção são meramente ilustrativas e a aplicação final pode apresentar algumas divergências. A exibição dessas telas estão divididas em duas perspectivas que serão expostas nos tópicos 3.2.7.1 e 3.2.7.2, que representam respetivamente os atores Usuário Visitante e Administrador.

#### 3.2.7.1 Perspectiva do Usuário Visitante

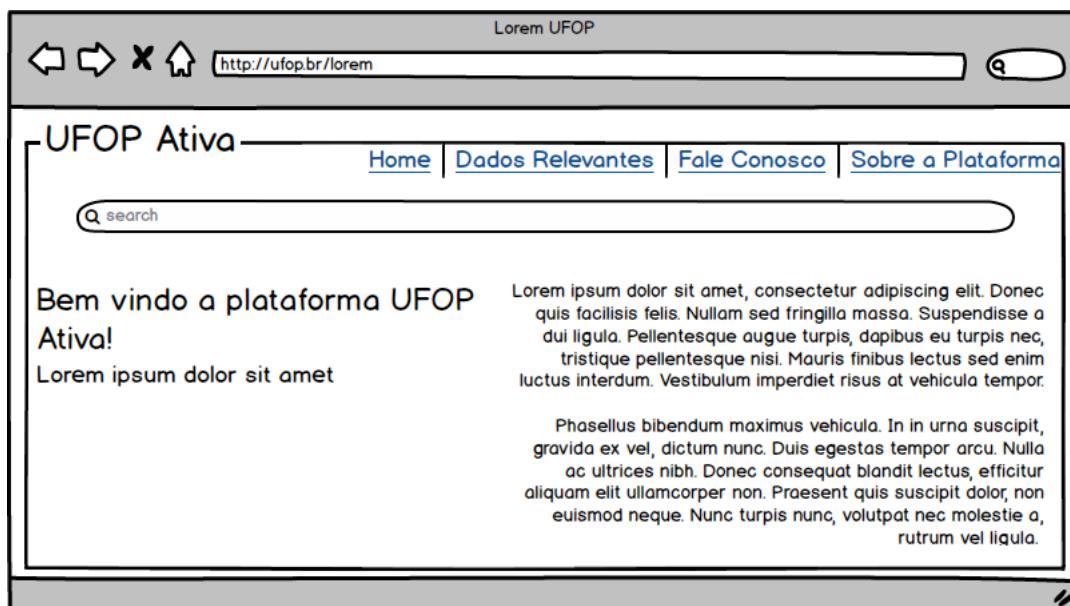


Figura 3.4 – Tela inicial da aplicação.

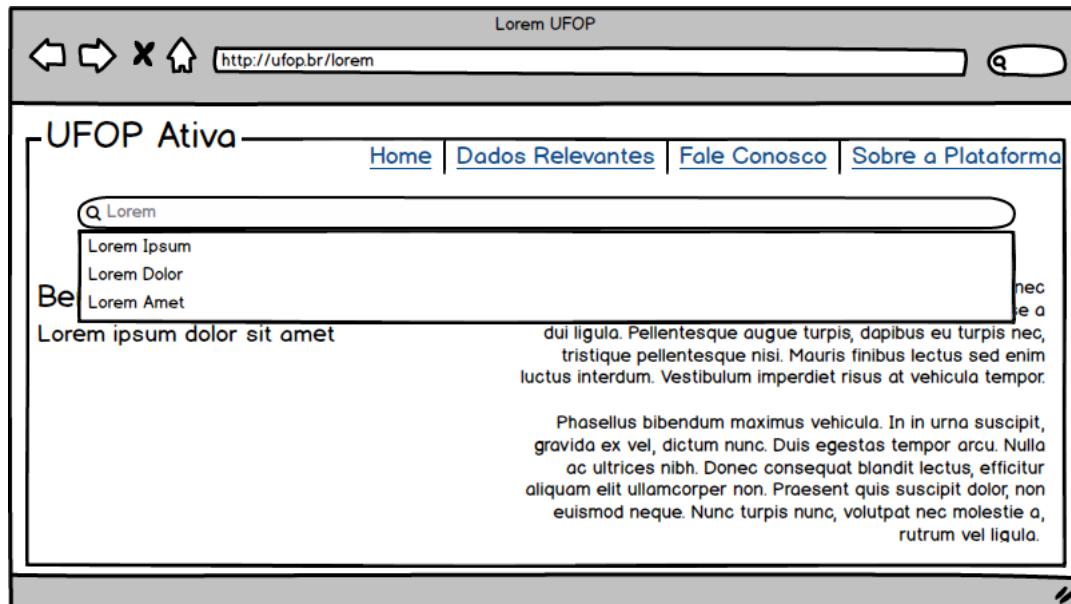


Figura 3.5 – Inserção de um texto no campo de pesquisa.

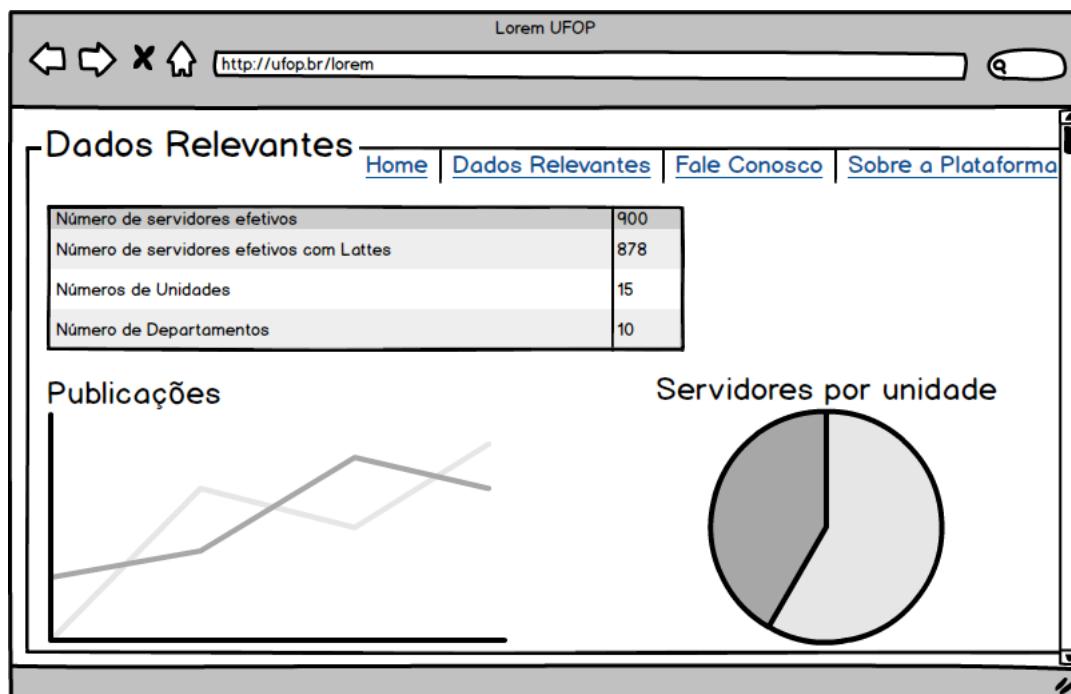


Figura 3.6 – Tela da aba "Dados Relevantes".

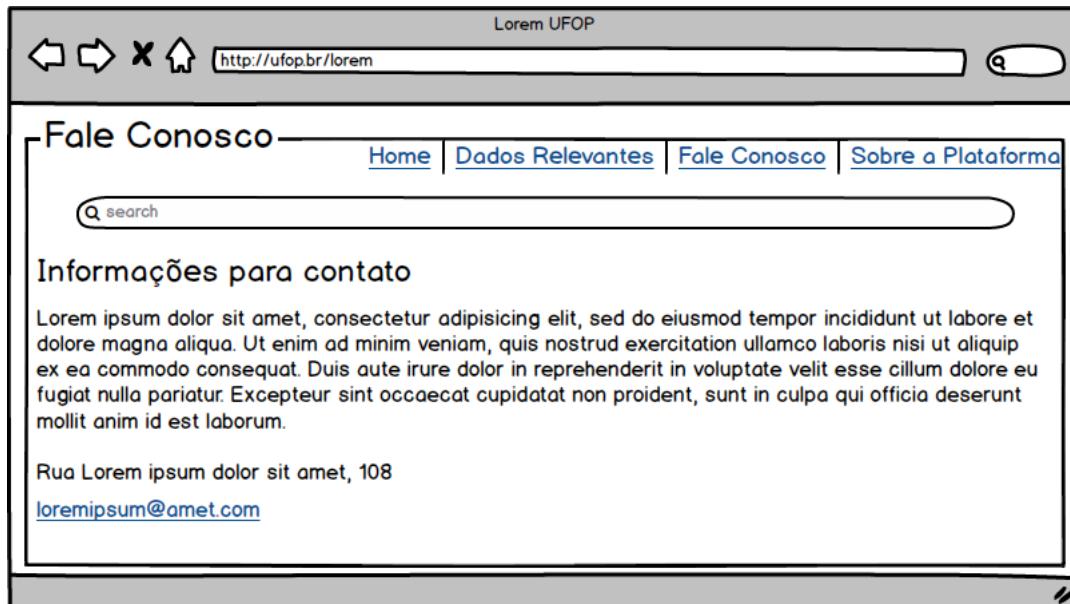


Figura 3.7 – Tela da aba "Fale Conosco".

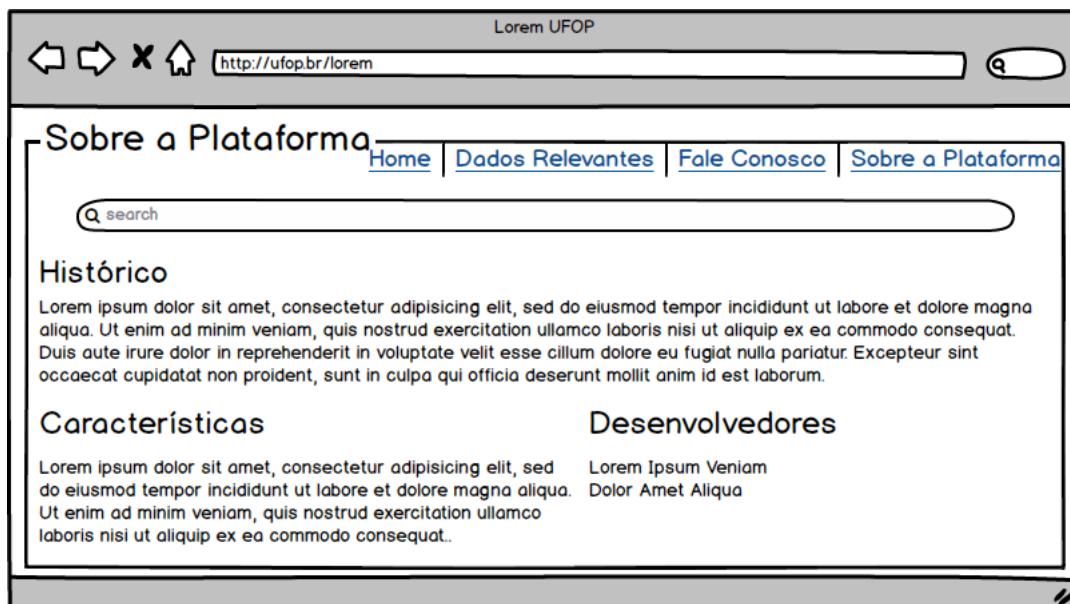


Figura 3.8 – Tela da aba "Sobre a Plataforma".

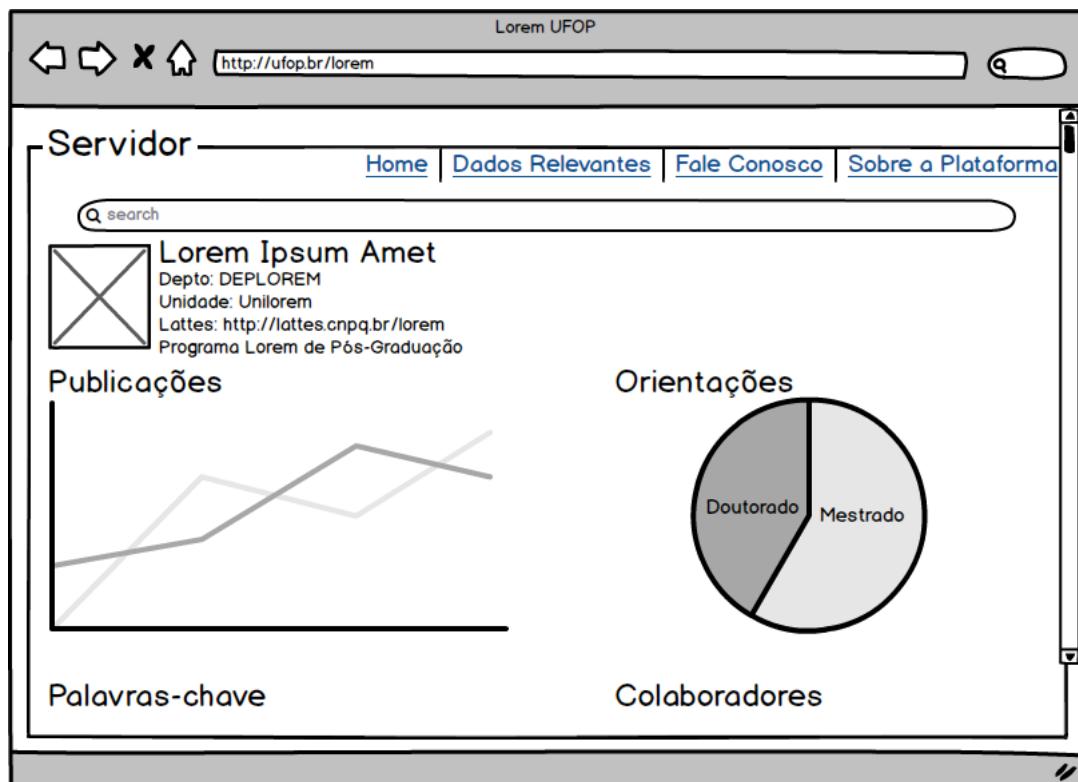


Figura 3.9 – Visualização do perfil de um servidor.

### 3.2.7.2 Perspectiva de Administrador

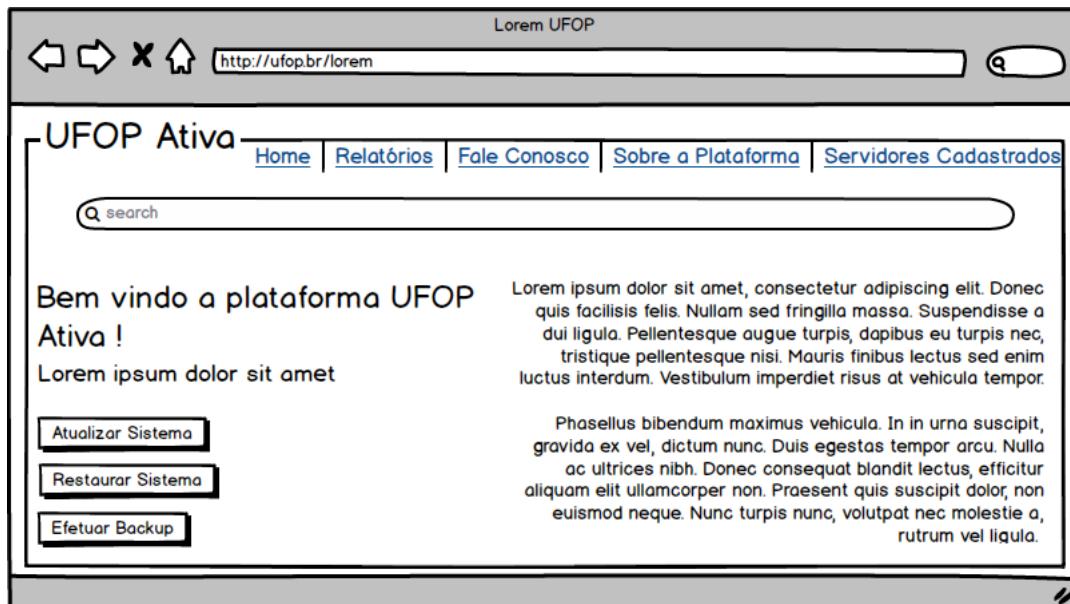


Figura 3.10 – Tela inicial da aplicação para o administrador.

The screenshot shows a web browser window titled "Lorem UFOP" with the URL "http://ufop.br/lorem". The main content is titled "Servidores Cadastrados". At the top right are links to "Home", "Relatórios", "Fale Conosco", and "Sobre a Plataforma". Below this is a search bar with the placeholder "search". The main area contains a table with columns: Nome, Depto, Unidade, and Seletor (checkboxes). The data in the table is as follows:

Nome	Dept	Unidade	Seletor
Giacomo Guilizzoni	Dep1	A	<input type="checkbox"/>
Marco Botton	Dep3	C	<input checked="" type="checkbox"/>
Mariah Maclachlan	Dep2	B	<input type="checkbox"/>
Valerie Liberty	Dep1	A	<input checked="" type="checkbox"/>
Lore Ipsum	Dep4	C	<input type="checkbox"/>

At the bottom are three buttons: "Adicionar Docente", "Atualizar Informações", and "Remover Selecionados".

Figura 3.11 – Tela dos Servidores Cadastrados para o administrador.

This screenshot is similar to Figure 3.11, showing the "Servidores Cadastrados" page. However, it includes a modal alert dialog in the center. The dialog has a title "Alerta" and the message "Tem certeza que deseja remover os selecionados?". Below the message are two buttons: "Não" and "Sim". The rest of the page structure is identical to Figure 3.11, including the table of servers and the bottom buttons.

Figura 3.12 – Alerta no momento de remoção de um servidor.

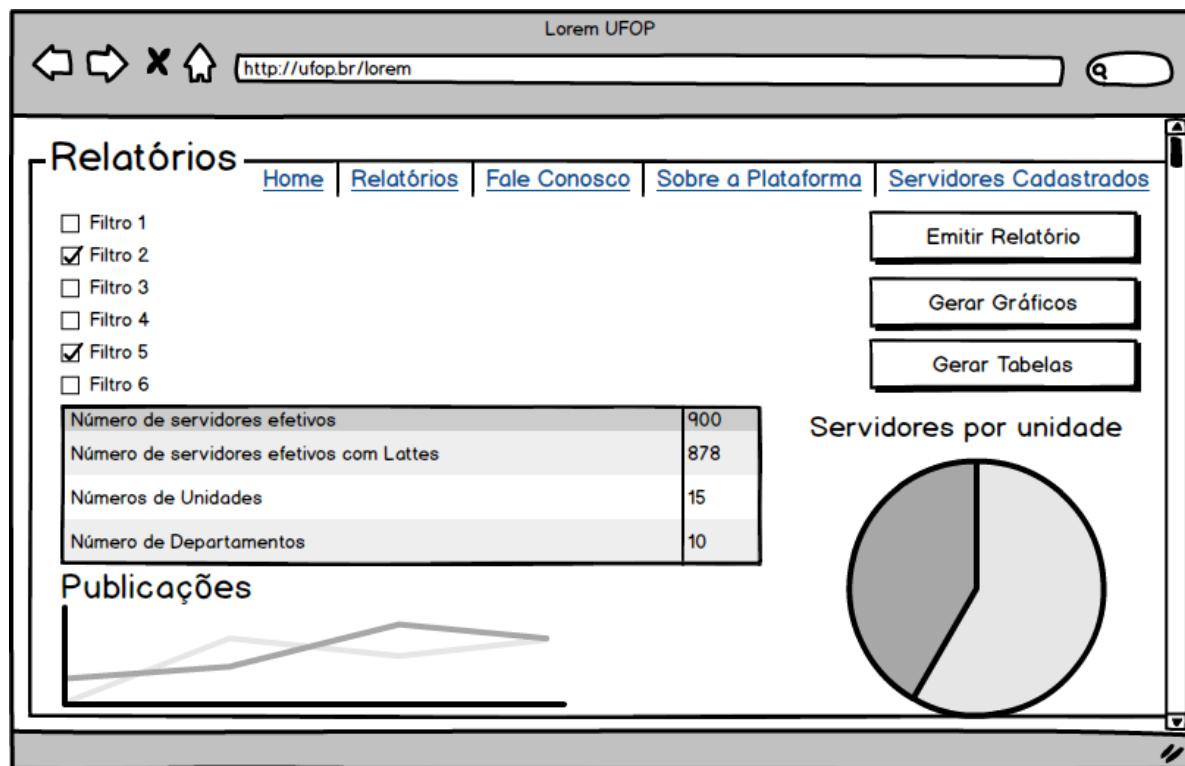


Figura 3.13 – Tela de “Relatórios” para o administrador.

As telas relacionadas a Unidades, Departamentos e Programas de Pós-Graduação cadastrados são análogas a tela de Servidores Cadastrados, bem como a tela de alerta de remoção.

### 3.2.8 Backlog

#### 3.2.8.1 Serviço de Navegação

1. Efetuar Consulta
2. Visualizar dados relevantes
3. Visualizar formas de entrar em contato
4. Visualizar informações sobre a plataforma

#### 3.2.8.2 Serviço de Gestão de Dados de Servidores

1. Inserir Servidor
2. Atualizar Informações de Servidor
3. Remover Servidor

4. Inserir Programa de Pós-Graduação
5. Atualizar Informações de Programa de Pós-Graduação
6. Remover Programa de Pós-Graduação

### **3.2.8.3 Serviço de Gestão de Dados de Unidades**

1. Inserir Unidade
2. Atualizar Informações de Unidade
3. Remover Unidade
4. Inserir Departamento
5. Atualizar Informações de Departamento
6. Remover Departamento

### **3.2.8.4 Serviço de Manutenção do Sistema**

1. Realizar Backup
2. Atualizar Sistema
3. Restaurar Sistema

### **3.2.8.5 Serviço Emissão de Relatórios**

1. Emitir Relatório
2. Gerar Gráficos
3. Gerar Tabelas

## **3.2.9 Estórias de Usuário**

### **3.2.9.1 Serviço de Navegação**

1. Efetuar consulta
  - **Como um** usuário visitante
  - **Com o objetivo de** encontrar uma informação no sistema
  - **Eu gostaria de** efetuar uma consulta
2. Visualizar dados relevantes

- **Como um** usuário visitante
- **Com o objetivo de** abrir uma página para ver os dados gerais referentes a UFOP
- **Eu gostaria de** consultar os dados relevantes

3. Visualizar formas de entrar em contato

- **Como um** usuário visitante
- **Com o objetivo de** abrir uma página para ver formas de contatar os criadores
- **Eu gostaria de** entrar em contato

4. Visualizar informações sobre a plataforma

- **Como um** usuário visitante
- **Com o objetivo de** abrir uma página de informações sobre o sistema
- **Eu gostaria de** acessar a aba sobre a plataforma

### 3.2.9.2 Serviço de Gestão de Dados de Servidores

1. Inserir Servidor

- **Como um** administrador
- **Com o objetivo de** adicionar às informações de um novo servidor ao sistema
- **Eu gostaria de** inserir um servidor

2. Atualizar Informações de Servidor

- **Como um** administrador
- **Com o objetivo de** alterar uma informação de um servidor e visualizar essa alteração
- **Eu gostaria de** atualizar informações de servidor

3. Remover Servidor

- **Como um** administrador
- **Com o objetivo de** não ter mais um servidor específico no sistema
- **Eu gostaria de** remover um servidor

4. Inserir Programa de Pós-Graduação

- **Como um** administrador
- **Com o objetivo de** adicionar às informações de um novo Programa de Pós-Graduação associado a um servidor do sistema

- **Eu gostaria de** inserir um Programa de Pós-Graduação

5. Atualizar Informações de Programa de Pós-Graduação

- **Como um** administrador
- **Com o objetivo de** alterar uma informação de um Programa de Pós-Graduação e visualizar essa alteração
- **Eu gostaria de** atualizar informações de Programa de Pós-Graduação

6. Remover Programa de Pós-Graduação

- **Como um** administrador
- **Com o objetivo de** não ter mais um Programa de Pós-Graduação específico no sistema
- **Eu gostaria de** remover um Programa de Pós-Graduação

### 3.2.9.3 Serviço de Gestão de Dados de Unidades

1. Inserir Unidade

- **Como um** administrador
- **Com o objetivo de** adicionar às informações de uma nova Unidade ao sistema
- **Eu gostaria de** inserir uma unidade

2. Atualizar Informações de Unidade

- **Como um** administrador
- **Com o objetivo de** alterar uma informação de uma Unidade e visualizar essa alteração
- **Eu gostaria de** atualizar informações de unidade

3. Remover Unidade

- **Como um** administrador
- **Com o objetivo de** não ter mais uma unidade específica no sistema
- **Eu gostaria de** remover uma unidade

4. Inserir Departamento

- **Como um** administrador
- **Com o objetivo de** adicionar às informações de um novo departamento associado a uma Unidade do sistema
- **Eu gostaria de** inserir um departamento

## 5. Atualizar Informações de Departamento

- **Como um** administrador
- **Com o objetivo de** alterar uma informação de um departamento e visualizar essa alteração
- **Eu gostaria de** atualizar informações de departamento

## 6. Remover Departamento

- **Como um** administrador
- **Com o objetivo de** não ter mais um departamento específico no sistema
- **Eu gostaria de** remover um departamento

### 3.2.9.4 Serviço de Manutenção do Sistema

#### 1. Realizar Backup

- **Como um** administrador
- **Com o objetivo de** ter uma cópia de segurança dos arquivos do sistema em um determinado diretório
- **Eu gostaria de** realizar um backup

#### 2. Atualizar Sistema

- **Como um** administrador
- **Com o objetivo de** ter funcionando uma versão atualizada do sistema
- **Eu gostaria de** atualizar o sistema

#### 3. Restaurar Sistema

- **Como um** administrador
- **Com o objetivo de** rodar o sistema em uma versão anterior
- **Eu gostaria de** restaurar o sistema

### 3.2.9.5 Serviço de Emissão de Relatórios

#### 1. Emitir Relatório

- **Como um** administrador
- **Com o objetivo de** visualizar determinadas informações em relação as pesquisas instituição

- **Eu gostaria de** emitir relatório

## 2. Gerar Gráficos

- **Como um** administrador
- **Com o objetivo de** visualizar determinadas informações em relação a instituição por meio de gráficos
- **Eu gostaria de** gerar gráficos

## 3. Gerar Tabelas

- **Como um** administrador
- **Com o objetivo de** visualizar determinadas informações em relação a instituição por meio de tabelas
- **Eu gostaria de** gerar tabelas

### 3.2.10 Cenários de Teste

#### 3.2.10.1 Serviço de Navegação

##### 1. Efetuar Consulta

- DADO que eu sou visitante E desejo realizar uma consulta a partir de uma palavra QUANDO eu preencher o campo de pesquisa ENTÃO os resultados começarão a ser exibidos;
- DADO que eu escrevi a minha consulta E existe resultado para a mesma QUANDO eu clicar em um resultado ENTÃO o sistema irá mostrar uma página referente aquela consulta;
- DADO que preenchi o campo de pesquisa E não existe resultado para a mesma ENTÃO o sistema permanecerá na tela sem apresentar resultados.

##### 2. Visualizar Dados Relevantes

- DADO que eu sou um visitante E estou na tela inicial do sistema E desejo conferir os dados relevantes referentes a pesquisa realizada na universidade QUANDO eu clicar na aba “Dados Relevantes” na parte superior da tela ENTÃO uma nova tela com os dados será exibida;
- DADO que eu sou um visitante E estou na tela de contato E desejo conferir os dados relevantes referentes a pesquisa realizada na universidade QUANDO eu clicar na aba “Dados Relevantes” na parte superior da tela ENTÃO uma nova tela com os dados será exibida;

- DADO que eu sou um visitante E estou na tela sobre o sistema E desejo conferir os dados relevantes referentes a pesquisa realizada na universidade QUANDO eu clicar na aba “Dados Relevantes” na parte superior da tela ENTÃO uma nova tela com os dados será exibida.

### 3. Visualizar formas de entrar em contato

- DADO que eu sou um visitante E estou na tela inicial do sistema E desejo conferir às informações para entrar em contato com os desenvolvedores QUANDO eu clicar na aba “Entrar em Contato” na parte superior da tela ENTÃO uma nova tela será exibida apresentando as informações de contato;
- DADO que eu sou um visitante E estou na tela de dados relevantes E desejo conferir às informações para entrar em contato com os desenvolvedores QUANDO eu clicar na aba “Entrar em Contato” na parte superior da tela ENTÃO uma nova tela será exibida apresentando as informações de contato;
- DADO que eu sou um visitante E estou na tela sobre o sistema E desejo conferir às informações para entrar em contato com os desenvolvedores QUANDO eu clicar na aba “Entrar em Contato” na parte superior da tela ENTÃO uma nova tela será exibida apresentando as informações de contato.

### 4. Visualizar informações sobre a plataforma

- DADO que eu sou um visitante E estou na tela inicial do sistema E desejo conferir às informações sobre o desenvolvimento da plataforma QUANDO eu clicar na aba “Sobre o sistema” na parte superior da tela ENTÃO uma nova tela será exibida apresentando as informações de contato;
- DADO que eu sou um visitante E estou na tela inicial de dados relevantes E desejo conferir às informações sobre o desenvolvimento da plataforma QUANDO eu clicar na aba “Sobre o sistema” na parte superior da tela ENTÃO uma nova tela será exibida apresentando as informações de contato;
- DADO que eu sou um visitante E estou na tela inicial de contato E desejo conferir às informações sobre o desenvolvimento da plataforma QUANDO eu clicar na aba “Sobre o sistema” na parte superior da tela ENTÃO uma nova tela será exibida apresentando as informações de contato.

#### 3.2.10.2 Serviço de Gestão de Dados de Servidores

##### 1. Inserir Servidor

- DADO que eu sou um administrador E desejo adicionar um as informações de um novo servidor no sistema DADO que eu já possua os dados em um formato pronto

para serem inseridos QUANDO eu clicar em “Adicionar Servidor” ENTÃO o servidor será incluído no sistema E uma mensagem de confirmação será exibida;

- DADO que eu não possua os dados em um formato pronto para serem inseridos QUANDO eu clicar em “Adicionar Servidor” ENTÃO uma mensagem de erro será exibida.

## 2. Atualizar Informações de Servidor

- DADO que eu sou administrador E estou na tela de servidores cadastrados E desejo atualizar as informações dos servidores DADO que uma nova atualização esteja pronta para ser adicionada QUANDO eu clicar em atualizar informações ENTÃO o sistema irá iniciar as atualizações;
- DADO que não haja uma nova atualização para ser adicionada ENTÃO o sistema exibirá uma mensagem.

## 3. Remover Servidor

- DADO que eu sou um administrador E estou na tela de servidores cadastrados E desejo remover as informações de um servidor do sistema QUANDO eu buscar pelo servidor a ser removido na caixa de pesquisa E clicar em remover ENTÃO uma caixa de alerta irá abrir;
- DADO que eu sou um administrador E estou na tela de servidores cadastrados E desejo remover as informações de um servidor do sistema QUANDO eu selecionar o servidor a ser removido E clicar em remover ENTÃO uma caixa de alerta irá abrir;
- DADO que a caixa de alerta está aberta QUANDO eu clicar em “sim” o servidor será removido do sistema;
- DADO que a caixa de alerta está aberta QUANDO eu clicar em “não” o servidor não será removido do sistema.

## 4. Inserir Programa de Pós-Graduação

- DADO que eu sou um administrador E desejo adicionar um as informações de um novo programa de pós-graduação no sistema DADO que eu já preenchi os campos sobre o programa a ser inserido QUANDO eu clicar em “Adicionar Programa de Pós-Graduação” ENTÃO o programa de pós-graduação será incluído no sistema E uma mensagem de confirmação será exibida;
- DADO que eu não preenchi os campos sobre o programa a ser inserido QUANDO eu clicar em “Adicionar Programa de Pós-Graduação” ENTÃO uma mensagem de erro será exibida.

## 5. Atualizar Informações de Programa de Pós-Graduação

- DADO que eu sou administrador E estou na tela de programas de pós-graduação cadastrados E desejo atualizar as informações dos programas de pós-graduação DADO que eu preenchi os campos a serem atualizados QUANDO eu clicar em atualizar informações ENTÃO o sistema irá atualizar as informações;
- DADO que eu não preenchi os campos a serem atualizados ENTÃO o sistema exibirá uma mensagem.

## 6. Remover Programa de Pós-Graduação

- DADO que eu sou um administrador E estou na tela de programas de pós-graduação cadastrados E desejo remover as informações de um programa de pós-graduação do sistema QUANDO eu buscar pelo programa de pós-graduação a ser removido na caixa de pesquisa E clicar em remover ENTÃO uma caixa de alerta irá abrir;
- DADO que eu sou um administrador E estou na tela de programas de pós-graduação cadastrados E desejo remover as informações de um programa de pós-graduação do sistema QUANDO eu selecionar o programa de pós-graduação a ser removido E clicar em remover ENTÃO uma caixa de alerta irá abrir;
- DADO que a caixa de alerta está aberta QUANDO eu clicar em “sim” o programa de pós-graduação será removido do sistema;
- DADO que a caixa de alerta está aberta QUANDO eu clicar em “não” o programa de pós-graduação não será removido do sistema.

### 3.2.10.3 Serviço de Gestão de Dados de Unidades

#### 1. Inserir Unidade

- DADO que eu sou um administrador E desejo adicionar umas informações de uma nova unidade no sistema DADO que eu já preenchi os campos sobre a unidade a ser inserida QUANDO eu clicar em “Adicionar Unidade” ENTÃO a unidade será incluída no sistema E uma mensagem de confirmação será exibida;
- DADO que eu não preenchi os campos sobre o programa a ser inserido QUANDO eu clicar em “Adicionar Unidade” ENTÃO uma mensagem de erro será exibida.

#### 2. Atualizar Informações de Unidade

- DADO que eu sou administrador E estou na tela de unidades cadastradas E desejo atualizar as informações de uma unidade DADO que eu preenchi os campos a serem atualizados QUANDO eu clicar em atualizar informações ENTÃO o sistema irá iniciar as atualizações;
- DADO que eu não preenchi os campos a serem atualizados ENTÃO o sistema exibirá uma mensagem.

### 3. Remover Unidade

- DADO que eu sou um administrador E estou na tela de unidades cadastradas E desejo remover as informações de uma unidade do sistema QUANDO eu buscar pela unidade a ser removida na caixa de pesquisa E clicar em remover ENTÃO uma caixa de alerta irá abrir;
- DADO que eu sou um administrador E estou na tela de unidades cadastradas E desejo remover as informações de uma unidade do sistema QUANDO eu selecionar a unidade a ser removida E clicar em remover ENTÃO uma caixa de alerta irá abrir;
- DADO que a caixa de alerta está aberta QUANDO eu clicar em “sim” a unidade será removida do sistema;
- DADO que a caixa de alerta está aberta QUANDO eu clicar em “não” a unidade não será removida do sistema.

### 4. Inserir Departamento

- DADO que eu sou um administrador E desejo adicionar um as informações de um novo departamento no sistema DADO que eu já preenchi os campos sobre o departamento a ser inserido QUANDO eu clicar em “Adicionar Departamento” ENTÃO o departamento será incluído no sistema E uma mensagem de confirmação será exibida;
- DADO que eu não preenchi os campos sobre o departamento a ser inserido QUANDO eu clicar em “Adicionar Departamento” ENTÃO uma mensagem de erro será exibida.

### 5. Atualizar Informações de Departamento

- DADO que eu sou administrador E estou na tela de departamentos cadastrados E desejo atualizar as informações dos departamentos DADO que eu preenchi os campos a serem atualizados QUANDO eu clicar em atualizar informações ENTÃO o sistema irá atualizar as informações;
- DADO que eu não preenchi os campos a serem atualizados ENTÃO o sistema exibirá uma mensagem.

### 6. Remover Departamento

- DADO que eu sou um administrador E estou na tela de departamentos cadastrados E desejo remover as informações de um departamentos do sistema QUANDO eu buscar pelo departamento a ser removido na caixa de pesquisa E clicar em remover ENTÃO uma caixa de alerta irá abrir;
- DADO que eu sou um administrador E estou na tela de departamentos cadastrados E desejo remover as informações de um departamento do sistema QUANDO eu selecionar o departamento a ser removido E clicar em remover ENTÃO uma caixa de alerta irá abrir;

- DADO que a caixa de alerta está aberta QUANDO eu clicar em “sim” o departamento será removido do sistema;
- DADO que a caixa de alerta está aberta QUANDO eu clicar em “não” o departamento não será removido do sistema.

### 3.2.10.4 Serviço de Manutenção do Sistema

#### 1. Realizar Backup

- DADO que eu sou um administrador E estou na tela inicial de administrador E desejo fazer um backup do sistema QUANDO eu clicar em “Efetuar Backup” ENTÃO o sistema irá fazer um backup dos dados.

#### 2. Atualizar Sistema

- DADO que eu sou um administrador E estou na tela inicial de administrador E preciso atualizar a versão do sistema DADO que tenha uma versão nova do sistema a ser colocada em uso QUANDO eu clicar em “Atualizar Sistema” ENTÃO o sistema será atualizado.

#### 3. Restaurar Sistema

- DADO que eu sou um administrador E estou na tela inicial de administrador E preciso voltar o sistema para uma versão anterior QUANDO eu clicar em “Restaurar Sistema” ENTÃO o sistema irá restaurar para a versão anterior.

### 3.2.10.5 Serviço de Emissão de Relatórios

#### 1. Emitir Relatório

- DADO que eu sou um administrador E estou na tela de relatórios E desejo gerar um determinado relatório QUANDO eu selecionar os filtros desejados E clicar Emitir Relatório ENTÃO o sistema irá gerar um relatório que será baixado;

#### 2. Gerar Gráficos

- DADO que eu sou um administrador E estou na tela de relatórios E desejo gerar um relatório por meio de gráficos QUANDO eu selecionar os filtros desejados E clicar Gerar Gráfico ENTÃO o sistema irá gerar os gráficos E estes poderão ser baixados ;

#### 3. Gerar Tabelas

- DADO que eu sou um administrador E estou na tela de relatórios E desejo gerar um relatório por meio de tabelas QUANDO eu selecionar os filtros desejados E clicar Gerar Gráfico ENTÃO o sistema irá gerar as tabelas E estas poderão ser baixados;

### 3.3 Extração dos Dados

A presente seção descreve a metodologia adotada para realizar a extração dos dados referentes aos servidores da UFOP por meio de seus currículos Lattes. A PROPP disponibilizou um arquivo no formato CSV que contém nome, CPF, escolaridade, categoria, departamento, unidade e programa de pós-graduação dos atuais 1.653 servidores ativos da universidade. Esse arquivo foi utilizado como fonte de dados sobre os servidores e para coletar os dados desses servidores na Plataforma Lattes. A seguir, as Subseções 3.3.1 e 3.3.2 detalham, respectivamente, os processos de download dos currículos em formato XML e a extração desses dados para importação no banco relacional.

É relevante destacar que a UFOP, por meio de um ofício devidamente assinado pelo seu dirigente máximo e encaminhado à Presidência do CNPq, solicitou o acesso a extração dos dados presentes na Plataforma Lattes. Este ofício contém a exposição de motivos e destinação a ser dada aos dados a serem extraídos. Esta solicitação possibilitou o acesso a um *WebService SOAP* de extração dos dados, por meio de uma máquina específica, que foi cedida por empréstimo pelo NTI da UFOP.

#### 3.3.1 Download Automático dos Currículos

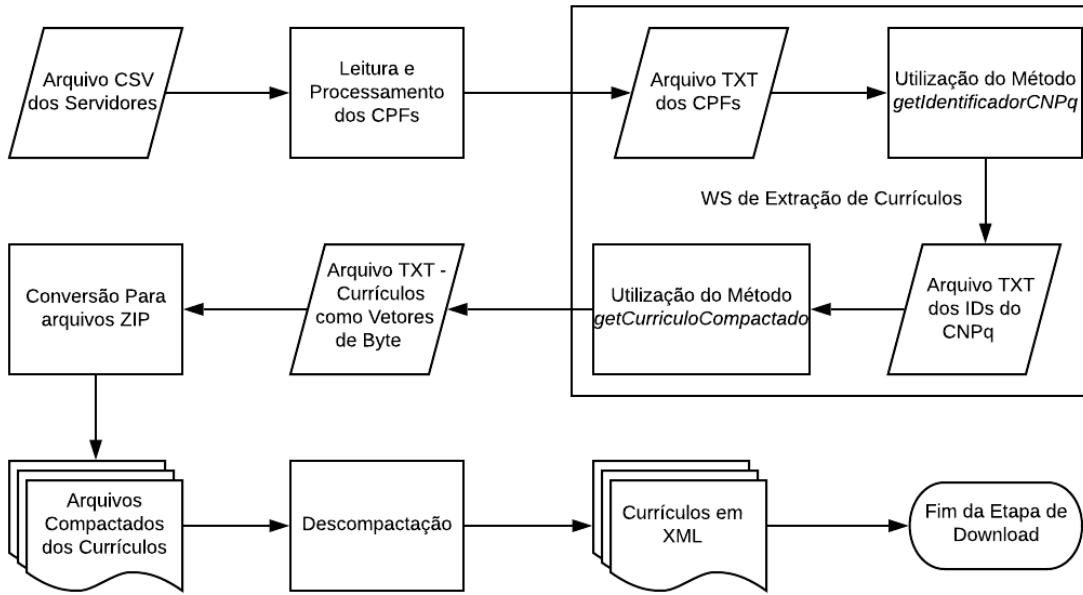


Figura 3.14 – Fluxograma do processo de download automático dos Currículos.

A partir do arquivo CSV dos servidores, implementou-se um programa em Java que realiza a leitura dos dados presentes no mesmo e efetua o processamento dos CPFs para remover os caracteres especiais e deixar apenas os números. Em seguida, esses CPFs foram escritos em

um arquivo de texto, separados por quebra de linha. Este arquivo é utilizado como parâmetro de entrada para obter os identificadores do CNPq dos servidores.

A tarefa de consumir o *WebService* fornecido é implementada em dois algoritmos na linguagem PHP, utilizando a ferramenta cURL, que serve para criar requisições em diversos protocolos, incluindo HTTP e HTTPS, e obter conteúdo remoto. O serviço SOAP de extração de currículos disponibilizado pelo CNPq é definido em um WSDL com alguns métodos que podem ser utilizados para a obtenção das informações desejadas. Na implementação deste trabalho, usufruiu-se de dois desses métodos. As implementações em PHP são encarregadas de invocar esses métodos.

Primeiramente, o método *getIdentificadorCNPq* é utilizado. Este recebe como parâmetros do tipo *string* o CPF, o nome completo e a data de nascimento de uma pessoa (não sendo obrigatório o uso de todos). O retorno é uma página HTML com o identificador do CNPq da pessoa no elemento *body*. O método foi aplicado à lista de CPFs gerada a partir da implementação anteriormente descrita. As requisições foram feitas utilizando a URL do serviço e os identificadores, obtidos como resposta, foram salvos em um arquivo texto, separados por quebra de linha.

Em seguida utilizou-se o método *getCurriculoCompactado*, que recebe como parâmetro apenas o identificador do CNPq e retorna a representação do currículo compactado no formato de um vetor de *byte*, também no elemento *body* de uma página HTML. Percorreu-se o arquivo de identificadores gerado pela implementação anterior, passando-os como parâmetro para as requisições feitas na URL do método atual. Os vetores resultantes também foram salvos separados por quebra de linha em um arquivo texto.

Após executar os métodos descritos anteriormente, o arquivo resultante, preenchido com os vetores de bytes dos currículos, foi manipulado por um programa em Java que lê cada linha do arquivo e converte de *string* para um vetor de *bytes*. Processou-se os vetores, salvando-os como arquivos compactados em um diretório. Outra implementação em Java foi responsável por fazer a leitura de cada um dos currículos compactados e descompactá-los gravando-os em formato XML dentro de outro diretório.

### 3.3.2 Extração dos Dados do XML

Para facilitar a parte de mapeamento dos dados provenientes dos currículos para o banco de dados proposto, os currículos em formato XML foram processados no *backend* da aplicação principal. Para extrair os dados dos arquivos em formato XML, utilizou-se uma API da plataforma Java EE denominada JAXB.

O JAXB permite mapear classes Java para representações XML e o inverso, ou seja, converte documentos XML em objetos java, opção aplicada neste trabalho. Sendo assim, primeiramente utilizou-se esse recurso para gerar classes java a partir do arquivo DTD que define a

estrutura dos currículos. Em seguida, já com as classes definidas, foram extraídos os dados de cada currículo e geradas instâncias dos objetos referentes a essas classes. A figura 3.15 representa o funcionamento do JAXB.

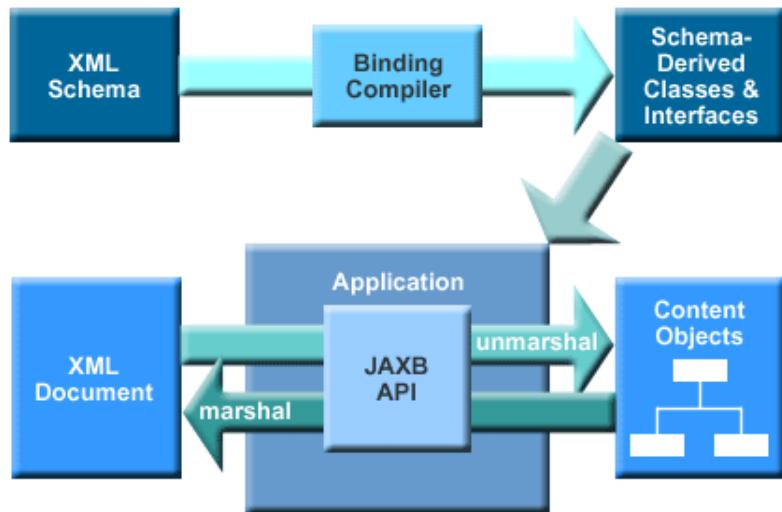


Figura 3.15 – Processos para acessar e construir documentos XML com base em aplicações Java.

Fonte: ([NETBEANS, 2011](#))

É importante relatar que surgiram problemas ao extrair um dos 1369 currículos manipulados. Aparentemente aconteceu algum erro na geração do arquivo XML do currículo por parte da própria Plataforma Lattes, pois o arquivo apresenta apenas um elemento de erro com a mensagem “Erro ao recuperar o XML”.

### 3.4 Mapeamento para o Banco de Dados do Sistema

O processo de mapeamento dos dados extraídos dos currículos em XML para a estrutura do banco de dados da aplicação é detalhado nesta seção. As subseções a seguir descrevem o processo de mapeamento dos dados, que foi subdividido nas tarefas de mapeamento para objetos java e mapeamento objeto-relacional. É importante ressaltar que apesar da maioria das informações mapeadas serem provenientes do currículo, alguns dados foram obtidos do arquivo CSV disponibilizado.

### 3.4.1 Mapeamento para Objetos Java

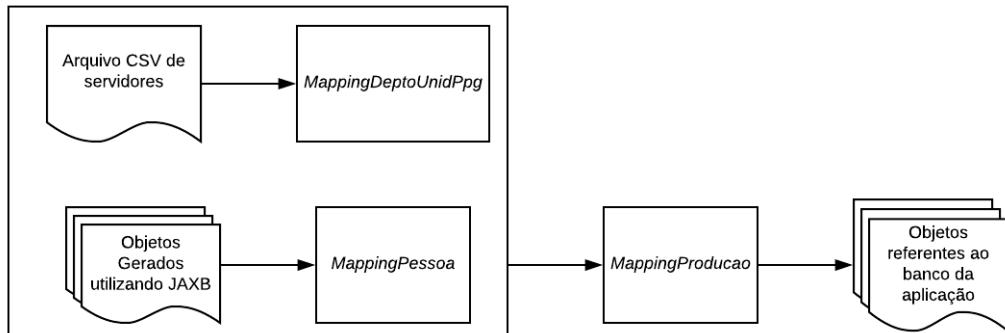


Figura 3.16 – Fluxograma do processo de mapeamento para Objetos Java.

As classes de mapeamento dos dados do currículo apenas definem as tarefas de armazenar informações nos objetos que especificam as entidades do banco de dados. Para isso, elas manipulam os objetos gerados a partir do currículo em XML (utilizando JAXB), através da chamada de métodos *getters* para obtenção das informações. Basicamente o que é feito nos objetos que representam o banco do sistema é a chamada de métodos *setters* para salvar atributos comuns ou *add* para colocar em listas de objetos. Esses dados posteriormente são mapeados e armazenados no banco de dados (tarefa esclarecida na Subseção 3.20).

A primeira parte do mapeamento das informações é responsável pelos dados relacionados ao arquivo CSV, não utilizando informações provenientes do currículo. Sendo assim, temos uma classe denominada *MappingDeptUnidPpg*, que possui apenas um método chamado *mapping*, responsável por mapear a correlação de um servidor com o departamento ao qual pertence, a unidade a qual faz parte, caso exista, e ao(s) Programa(s) de Pós-Graduação ao qual está vinculado, caso exista.

O mapeamento da entidade Pessoa é definido pela classe *MappingPessoa*, que por meio do método *mapping*, mapeia os atributos básicos da pessoa e utiliza outros cinco métodos para mapear Áreas, Formações Acadêmicas, Projetos de Pesquisa, Financiadores, Atuações Profissionais e Orientações. Além disso, ainda há o mapeamento de Palavras Chave e Vínculos Institucionais. As Figuras 3.17 e 3.18 exemplificam o mapeamento de alguns atributos e objetos.

```

public void mapping(CURRICULOVITAE cv) throws ParseException {
    pessoa.setNomeCompleto(cv.getDADOSGERAIS().getNOMECOMPLETO());
    if(cv.getNUMEROIDENTIFICADOR() != null) {
        pessoa.setLinkLattes("lattes.cnpq.br/" + cv.getNUMEROIDENTIFICADOR());
        pessoa.setIdLattes(cv.getNUMEROIDENTIFICADOR());
    }
    /* Demais mapeamentos */
}

```

Figura 3.17 – Exemplo do mapeamento de atributos simples de uma pessoa com métodos *setters*.

```

public List<Area> mappingAreas(List<AREADEATUACAO> areasAtuacao) {
    Area area = new Area();
    int size = areasAtuacao.size();
    List<Area> areasMapeadas = new ArrayList<Area>();

    for(int i=0; i<size; i++) {
        area.setGrandeArea(areasAtuacao.get(i).getNOMEGRANDEAREADOCONHECIMENTO());
        area.setNomeArea(areasAtuacao.get(i).getNOMEDAAREADOCONHECIMENTO());
        area.setSubArea(areasAtuacao.get(i).getNOMEDASUBAREADOCONHECIMENTO());
        area.setEspecialidade(areasAtuacao.get(i).getNOMEDAESPECIALIDADE());
        areasMapeadas.add(area);
        area = new Area();
    }
    return areasMapeadas;
}

```

Figura 3.18 – Exemplo do mapeamento da lista de áreas de uma pessoa utilizando *add* em *list*.

Para finalizar, há a classe *MappingProducao*, que é responsável por definir o mapeamento das produções de uma pessoa. O método *mapping* dessa classe realiza a chamada de três métodos responsáveis por mapear Produções Bibliográficas, Produções Técnicas e Produções Artísticas Culturais. Nesses três métodos, há chamadas a diversos outros métodos para mapear especificamente os vários tipos de produções existentes na estrutura do banco de dados. A Figura 3.19 ilustra o mapeamento de um tipo de produção.

```

public void mappingArtigos(List<ARTIGOPUBLICADO> artigosPublicados) {
    Artigo artigo = new Artigo();

    for(int i=0; i<artigosPublicados.size(); i++) {

        //ATRIBUTOS DA SUPERCLASSE PRODUCAO
        artigo.setTituloPortugues(artigosPublicados.get(i).getDADOSBASICOSDOARTIGO().getTITULODOARTIGO());
        artigo.setTituloIngles(artigosPublicados.get(i).getDADOSBASICOSDOARTIGO().getTITULODOARTIGOINGLES());
        if(MyFunctions.isNumber(artigosPublicados.get(i).getDADOSBASICOSDOARTIGO().getANO DOARTIGO()))
            artigo.setAno(new Integer (artigosPublicados.get(i).getDADOSBASICOSDOARTIGO().getANO DOARTIGO()));
        artigo.setPais(artigosPublicados.get(i).getDADOSBASICOSDOARTIGO().getPAISDEPUBLICACAO());

        //ATRIBUTOS DA PRIMEIRA SUBCLASSE PRODUCAOBIBLIOGRAFICA (OCULTA)
        artigo.setDoi(artigosPublicados.get(i).getDADOSBASICOSDOARTIGO().getDOI());
        artigo.setLocal(artigosPublicados.get(i).getDADOSBASICOSDOARTIGO().getPAISDEPUBLICACAO());
        if(MyFunctions.isNumber(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAFINAL()) &&
           MyFunctions.isNumber(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAINICIAL()) &&
           (MyFunctions.smallerThanNine(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAFINAL()) ||
            && MyFunctions.smallerThanNine(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAINICIAL())))
            Integer pgFinal = new Integer (artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAFINAL());
            Integer pgInicial = new Integer (artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAINICIAL());
            artigo.setNumeroPaginas(new Integer (pgFinal - pgInicial));
        }

        //ATRIBUTOS DA SUBCLASSE ARTIGO
        if(MyFunctions.isNumber(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAFINAL()))
            if(MyFunctions.smallerThanNine(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAFINAL()))
                artigo.setPáginaFinal(new Integer (artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAFINAL()));
        if(MyFunctions.isNumber(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAINICIAL()))
            if(MyFunctions.smallerThanNine(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAINICIAL()))
                artigo.setPáginaInicial(new Integer (artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getPAGINAINICIAL()));
        artigo.setIssn(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getISSN());
        artigo.setTituloPeriodicoRevista(artigosPublicados.get(i).getDETALHAMENTODOARTIGO().getTITULODOPERIODICOUREVISTA());

        //MAPPING PALAVRAS CHAVES
        palavrasChaves.addAll(dbMap.mappingPalavrasChaves(artigosPublicados.get(i).getPALAVRASCHAVE()));

        //MAPPING AUTORES
        artigo.getAutores().addAll(mappingAutores(artigosPublicados.get(i).getAUTORES()));

        //ADD EM PRODUCOES
        artigos.add(artigo);
        artigo = new Artigo();
    }
}

```

Figura 3.19 – Exemplo de mapeamento dos artigos de uma pessoa utilizando *add* em *list*.

### 3.4.2 Mapeamento Objeto-Relacional, JPA e Hibernate



Figura 3.20 – Fluxograma do mapeamento Objeto-Relacional.

Aplicações Java por padrão são implementadas seguindo o paradigma de programação orientada a objetos, no qual os dados são representados por meio de classes e atributos. Além disso, ainda existem alguns conceitos que podem ser aplicados, como herança, polimorfismo, composições, dentre outros. No entanto, a programação orientada a objetos se difere consideravelmente do esquema relacional. Para representar os dados em bancos relacionais, utilizam-se relações (tabelas) e atributos (colunas). As tabelas geralmente possuem chave primária (PK) e podem ser relacionadas por meio da criação de chaves estrangeiras (FK) a outras tabelas.

É possível perceber a existente divergência entre os paradigmas, o que gera bastante trabalho: a todo momento é necessário “transformar” objetos em registros e registros em objetos.

Sendo assim, é interessante usar ferramentas que possam fazer esse mapeamento de maneira automática. Neste contexto entra a JPA e o Hibernate, ferramentas que são confundidas em algumas situações, mas que se diferem e podem ser utilizadas juntas.

A JPA é uma API padrão da linguagem Java que descreve uma interface comum para *frameworks* de persistência de dados. Ela define um meio de mapeamento objeto-relacional para objetos Java simples e comuns, denominados *beans* de entidade. Já o Hibernate é um *framework* para o mapeamento objeto-relacional escrito na linguagem Java.

Portanto, o Hibernate é de fato o framework ORM, ou seja, a implementação física do que será utilizado para persistir, remover, atualizar ou buscar dados no SGBD. Por outro lado, o JPA é uma camada que descreve uma interface comum para frameworks ORM. Neste trabalho, utilizou-se as duas ferramentas.

### 3.5 Deduplicação de Produções

Em computação, como está exemplificado na Figura 3.21, a deduplicação consiste basicamente em eliminar cópias duplicadas de dados com o intuito principal de reduzir a sobrecarga de armazenamento. Além disso, em alguns contextos a deduplicação também é fundamental para possibilitar a veracidade de indicadores que são apresentados em sistemas de informação. No sistema proposto, por exemplo, ao exibir informações, a intenção é não contar mais de uma vez uma produção que é compartilhada por dois ou mais servidores. O essencial é que os dados mostrados pela aplicação a respeito das produções por departamentos, unidades e instituição sejam reais.

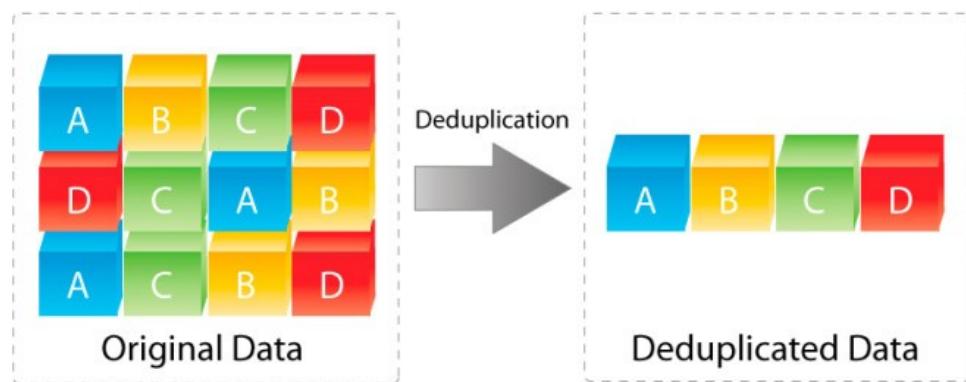


Figura 3.21 – Exemplo: deduplicação de dados.

Fonte: (DUARTE, 2019)

Esta seção, descreve as técnicas adotadas no processo de deduplicação das produções que são extraídas na ferramenta implementada. Nesta primeira versão, a deduplicação foi realizada

nos artigos completos publicados em periódicos, devido a maior interesse nesses dados e também por serem mais padronizados.

Para realizar a tarefa de deduplicação criou-se um pacote separado denominado *deduplication*. Foi criada uma classe e implementado o método *deduplicateArtigos* e um método auxiliar denominado *contains*. O primeiro método percorre uma lista de artigos recebida como parâmetro e utiliza o método *contains* para verificar se o artigo a ser analisado já foi adicionado à lista de artigos que será retornada (sem duplicatas).

No método *contains* verifica-se a similaridade de dois artigos de maneira hierárquica. Primeiramente, se ambos apresentarem DOI, é verificado se possuem o mesmo valor para este campo. Caso contrário, utiliza-se um índice de similaridade baseado na *levenshtein distance* (distância de edição) para analisar a similaridade do título e do periódico ou revista entre as duas produções. A similaridade é aceita se a comparação superar o limiar de 0.9 definido.

A distância de edição é representada pela quantidade mínima de inserções, remoções ou substituições de caracteres necessárias para transformar um termo em outro. Para obter uma taxa percentual de similaridade realiza-se a operação apresentada a seguir, na qual, *distEdicao* representa a distância de edição entre dois termos e *maiorTermo* representa o comprimento do maior termo.

$$\text{similaridade} = 1 - \frac{\text{distEdicao}}{\text{maiorTermo}}$$

A efetividade da deduplicação implementada foi apurada realizando testes com até 5 currículos que apresentam produções em comum, apresentando resultados condizentes com apurações realizadas manualmente.

## 3.6 Implementação do *Frontend* e do *Backend* do UFOP Ativa

Nesta seção, é descrito o processo de desenvolvimento da primeira versão do sistema UFOP Ativa. Como já foi esclarecido anteriormente, para a implementação do *backend* utilizou-se a linguagem Java juntamente com o *framework* Spring Boot, comunicando com um SGBD no PostgreSQL. No *frontend*, o *framework* Angular foi a opção adotada. As subseções 3.6.1 e 3.6.2 detalham mais a construção do *backend* e *frontend*, respectivamente.

É importante ressaltar a utilização do Maven<sup>4</sup> na construção do *backend*, que é uma ferramenta de automação de compilação hospedada pela Apache Software Foundation<sup>5</sup> e utilizada principalmente em projetos Java. O Maven descreve o projeto de software a ser construído através de um arquivo XML denominado *pom*, além disso descreve também suas dependências, diretórios, plug-ins, a ordem de compilação, dentre outras propriedades. A partir do arquivo *pom*, o Maven constrói e baixa o conteúdo definido.

<sup>4</sup> <<https://maven.apache.org/>>

<sup>5</sup> <<https://www.apache.org/>>

Por motivos de segurança, navegadores restringem requisições *cross-origin* HTTP a partir de *scripts*. Requisições feitas por meio de *XMLHttpRequest*, por exemplo, seguem a política de mesma origem (*same-origin policy*). Portanto, uma aplicação *Web* que faz uso desta API pode somente fazer requisições HTTP da mesma origem (domínio, protocolo e porta) da qual a aplicação foi carregada. Nesse contexto, que entra o mecanismo de CORS, que como a própria definição sugere, permite o compartilhamento de recursos com origens diferentes. Para isso, o CORS utiliza cabeçalhos adicionais HTTP para comunicar ao navegador que autorize uma aplicação *Web* seja executada em uma origem (domínio) com permissão para acessar recursos selecionados de um servidor, em uma origem distinta.

### 3.6.1 Backend

O Spring Boot possibilita algumas facilidades na construção do *backend* de aplicações. Uma dessas facilidades é o uso de arquivos de configuração e/ou inicialização que, através da inserção de dependências ou propriedades dos módulos a serem utilizados, o próprio *framework* reconhece e configura os mesmos. Exemplos de arquivos de configuração manipulados são o *pom.xml* (do Maven) e o *application.properties* (utilizado no Spring Boot). A configuração do PostgreSQL como SGBD do sistema foi feita através desses arquivos. As Figuras 3.22 e 3.23 mostram como configurar a aplicação para a utilização do PostgreSQL.

```
<dependency>
    <groupId>org.postgresql</groupId>
    <artifactId>postgresql</artifactId>
    <scope>runtime</scope>
</dependency>
```

Figura 3.22 – Exemplo da configuração adicionada ao *pom.xml* para utilizar o SGBD PostgreSQL na aplicação.

```
## Spring DATASOURCE (DataSourceAutoConfiguration & DataSourceProperties)
spring.datasource.url=jdbc:postgresql://localhost:5432/ufop-atival
spring.datasource.username= postgres
spring.datasource.password=

# The SQL dialect makes Hibernate generate better SQL for the chosen database
spring.jpa.properties.hibernate.dialect = org.hibernate.dialect.PostgreSQLDialect

# Hibernate ddl auto (create, create-drop, validate, update)
spring.jpa.hibernate.ddl-auto = update
```

Figura 3.23 – Exemplo da configuração adicionada *application.properties* para utilizar o SGBD PostgreSQL na aplicação.

Feitas todas as configurações, a implementação do *backend* baseou-se em cinco pacotes: *config*, *models*, *repositories*, *resources* e *services*. O pacote *config* é responsável por configurar o

CORS e permitir que o *frontend* envie requisições ao *backend* sem ser bloqueado pela política da mesma origem (*same-origin policy*). Os demais pacotes estão relacionados aos recursos da aplicação propriamente dita.

No pacote *models*, define-se os modelos, representados por classes java, que descrevem todos os tipos entidades do banco de dados do sistema, com seus respectivos atributos, exceto as entidades *Producao\_Bibliografica*, *Producao\_Tecnica* e *Producao\_Artistica\_Cultural*, as quais os atributos foram adicionados nas classes derivadas das mesmas. Para representar as produções que não se enquadram em nenhuma das derivações criou-se os modelos *OutraProducaoBibliografica*, *OutraProducaoTecnica* e *OutraProducaoArtisticaCultural*.

Ainda nos modelos, são estabelecidos através de anotações e java, padrões de mapeamento para o esquema relacional de características como relacionamentos, especializações, chaves estrangeiras, chaves primárias, atributos multivvalorados, dentre outras. O mapeamento propriamente dito é realizado posteriormente utilizando Hibernate e JPA, conceitos que já foram abordados anteriormente. As Figuras 3.24, 3.25 e 3.26 mostram algumas anotações utilizadas para definição de entidades, especializações, chaves primárias e relacionamentos.

```

@Entity
@Inheritance(strategy=InheritanceType.JOINED)
public abstract class Pessoa implements Serializable {
    private static final long serialVersionUID = 1L;

    @Id
    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private Integer id;

    ...
}

```

Figura 3.24 – Definição da entidade *pessoa*, sua especialização e uma chave primária *id*.

```

(a) @ManyToMany
@JoinTable(
    name="pessoa_producao"
    , joinColumns={
        @JoinColumn(name="id_producao")
    }
    , inverseJoinColumns={
        @JoinColumn(name="id_pessoa")
    }
)
private List<Pessoa> pessoas = new ArrayList<Pessoa>();

(b) @ManyToMany(mappedBy="pessoas")
private List<Producao> producoes = new ArrayList<>();

```

Figura 3.25 – Definição de relacionamento  $N \times M$  gerando a tabela *pessoa\_producao*, (a) Lado da classe Producao (b) Lado da classe Pessoa

```
(a) @ManyToOne
    @JoinColumn(name="id_pessoa")
    public Pessoa pessoa;

(b) @OneToMany(mappedBy="pessoa")
    private List<PalavraChave> palavrasChave = new ArrayList<>();
```

Figura 3.26 – Definição de relacionamento  $1 \times N$ , (a) Lado da classe Palavra\_Chave (b) Lado da classe Pessoa

No pacote *repositories*, definiram-se interfaces java (repositórios) referentes às classes criadas no pacote *models*, exceto para as classes derivadas. Essas interfaces estendem a implementação da classe *JpaRepository* e são responsáveis pela comunicação com o banco de dados. A classe *JpaRepository* já apresenta, por definição, alguns métodos como *save* e *delete*, para manipulação do banco de dados. A Figura 3.27 mostra um exemplo de repositório.

```
package br.ufop.ufopativa.repositories;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

import br.ufop.ufopativa.models.Pessoa;

@Repository
public interface PessoaRepository extends JpaRepository<Pessoa, Integer> {
}
```

Figura 3.27 – Interface que define o repositório do modelo *Pessoa*.

Para entender o pacote *resources*, primeiramente é necessário esclarecer o pacote *services*. Neste pacote, definiram-se serviços (representados por classes) para os modelos criados no pacote *models* que sejam precisos, no caso deste trabalho, para objetos que serão enviados ao *frontend* em rotas diferentes. Dentro de cada serviço, cria-se um atributo do tipo do repositório referente ao serviço. Esse repositório é utilizado para invocar os métodos já existentes ou para implementação de novos. O papel dos serviços é basicamente ser um intermediário entre as requisições HTTP, manipuladas pelos recursos (pacote *resources*) e a comunicação com o banco relacional, realizada pelos repositórios (pacote *repositories*). A Figura 3.28 mostra um exemplo de serviço.

```

@Service
public class PessoaService {

    @Autowired
    private PessoaRepository rep;

    public Pessoa findPessoa(Integer id) {
        Optional<Pessoa> obj = rep.findById(id);
        return obj.orElseThrow(() -> new ObjectNotFoundException(
            "Objeto não encontrado! Id: " + id + ", Tipo: " + Pessoa.class.getName()));
    }

    public List<Pessoa> getPessoas() {
        return rep.findAll();
    }

}

```

Figura 3.28 – Classe que define o serviço referente ao modelo *Pessoa*.

O pacote *resources* é responsável pelo controle das requisições HTTP recebidas. Nele, são criados, para cada serviço definido, o que denomina-se recursos (representados por classes java). Em cada recurso, cria-se um atributo do tipo do serviço referente ao recurso, este serviço é utilizado para invocar os métodos definidos (ou chamados) pelo serviço equivalente. Além disso, também define-se a(s) rota(s) nas quais serão representados em formato JSON, após processadas, as respostas das requisições realizadas pelo *frontend*. A Figura 3.29 apresenta um exemplo de recurso.

```

@RestController
@RequestMapping(value="/pessoas")
public class PessoaResource {

    @Autowired
    private PessoaService service;

    @RequestMapping(value="/{id}", method=RequestMethod.GET)
    public ResponseEntity<?> find(@PathVariable Integer id) {
        Pessoa obj = service.findPessoa(id);
        return ResponseEntity.ok().body(obj);
    }

    @GetMapping
    public List<Pessoa> pessoas() {
        return service.getPessoas();
    }
}

```

Figura 3.29 – Classe que define o recurso referente ao serviço *Pessoa*.

Para exemplificar, uma requisição feita pelo *frontend* no domínio de hospedagem do *backend*, complementado por “/pessoas/10”, é processada da seguinte forma. Primeiramente é recebida pelo recurso do tipo *PessoaResource*, que irá invocar o método *findPessoa* através do

atributo *service* do tipo *PessoaService* passando o identificador 10 recebido na rota. O serviço por sua vez invoca o método *findById* por meio do seu atributo *rep* do tipo *PessoaRepository*, passando o *id* recebido. O repositório responsabiliza-se por comunicar com o banco de dados e retornar o resultado da requisição para o serviço, que em seguida retorna para o recurso e este posteriormente envia a resposta para a rota especificada em formato JSON. A figura 3.30 representa o funcionamento do *backend*.

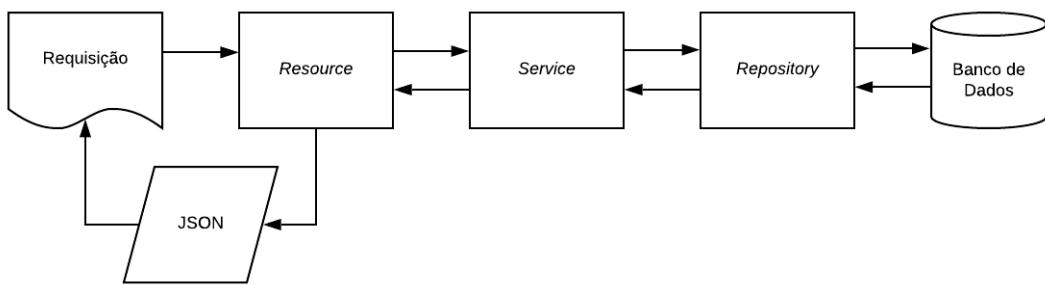


Figura 3.30 – Processamento de uma requisição pelo *backend*.

### 3.6.2 Frontend

Assim como o Spring Boot, o Angular também apresenta simplificações que facilitam a implementação de projetos *frontend*. A ferramenta Angular CLI dispõe de comandos que podem ser utilizados para criar um projeto Angular pré-configurado e outras propriedades de um projeto como componentes, serviços, classes, dentre outras. Os recursos desta ferramenta foram significativamente utilizados no desenvolvimento do *frontend* do sistema, principalmente na criação de componentes.

Os componentes em Angular, basicamente representam objetos que serão manipulados na interface da aplicação. Um componente se divide em três partes, as importações necessárias; o decorador, que define as diferentes propriedades que vão estruturar o componente; e a classe exportada propriamente dita, que define toda a lógica do componente, desde a criação de atributos até a definição de funções inerentes ao componente. Dentro do arquivo HTML do componente é possível ter acesso a funcionalidades e atributos do mesmo, permitindo que seja possível visualizar os dados de um componente na tela da aplicação. A Figura 3.31 ilustra um componente.

```

import { Component, OnInit } from '@angular/core';
import { ActivatedRoute } from '@angular/router'

import { UnidadesService } from '../unidades.service'
import { Unidade } from './unidade.model'

@Component({
  selector: 'ua-unidade',
  templateUrl: './unidade.component.html',
  styleUrls: ['./unidade.component.css']
})
export class UnidadeComponent implements OnInit {

  unidade: Unidade

  constructor(private unidadeService: UnidadesService, private route: ActivatedRoute) { }

  ngOnInit() {
    this.unidadeService.unidadeById(this.route.snapshot.params['id'])
      .subscribe(unidade => this.unidade = unidade)
  }

}

```

Figura 3.31 – Exemplo de componente que representa uma *Unidade*.

Os serviços são responsáveis pela comunicação com o *backend*, requisições HTTP são feitas dentro deles. A resposta recebida é mapeada para retirar apenas as informações relevantes em formato JSON. A estrutura dos dados pode ser definida por um modelo, que basicamente é uma interface que define os atributos que são recebidos para um componente específico. As Figuras 3.32 e 3.33 mostram respectivamente um serviço e um modelo. A figura 3.34 mostra o funcionamento do *frontend*.

```

import { Injectable } from '@angular/core'
import { HttpClient } from '@angular/common/http'
import { Observable } from 'rxjs'
import { map, catchError } from 'rxjs/operators'

import { Unidade } from './unidade/unidade.model'

import { UA_API } from '../app.api'

@Injectable()
export class UnidadesService {
  constructor(private http: HttpClient){}

  unidades(): Observable<Unidade[]>{
    return this.http.get<Unidade[]>(`${UA_API}/unidades`)
      .pipe(
        map(response => response.json())
      )
  }

  unidadeById(id: number): Observable<Unidade>{
    return this.http.get<Unidade>(`${UA_API}/unidades/${id}`)
      .pipe(
        map(response => response.json())
      )
  }
}

```

Figura 3.32 – Exemplo de serviço para buscar *Unidades*.

```
export interface Unidade {
    id: number
    nome: string
    sigla: string
    departamentos: Departamento[]
}
```

Figura 3.33 – Exemplo de um modelo que define a estrutura de um objeto *Unidade*.

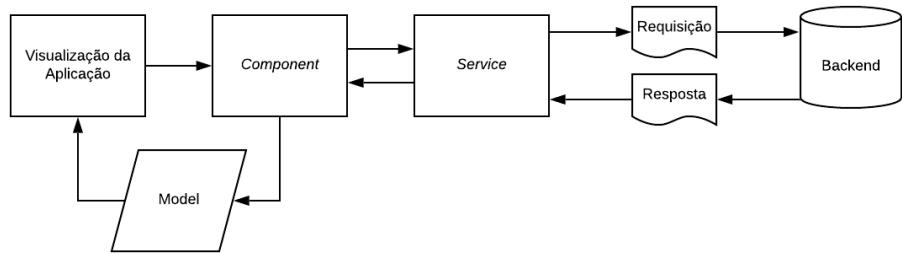


Figura 3.34 – Funcionamento do *frontend*.

A estilização da aplicação usufrui do *framework* de código aberto Bootstrap, utilizado para o desenvolvimento de componentes de interface e *frontend* para sites e aplicações *Web* usando HTML, CSS e JavaScript. Esse *framework* baseia-se em modelos de *design*, que possibilitam a experiência do usuário em um site amigável e responsivo. Algumas pequenas redefinições de estilo foram feitas para dar a aplicação um visual próprio. A figura 3.35 apresenta a página inicial do sistema, ilustrando um pouco da estilização adotada.



Figura 3.35 – Tela inicial da aplicação.

## 4 Estudo de Caso

Neste capítulo são descritos alguns estudos de caso do sistema UFOP Ativa, apresentado por meio de telas da versão inicial da aplicação desenvolvida. É possível perceber que a aplicação facilita o acesso a informações relacionadas aos servidores da UFOP, apresentando-as por meio de uma interface intuitiva, organizada e bem feita. A seguir, são mostradas e descritas algumas telas do sistema, simulando o uso por um usuário. A Figura 4.3, por questões de privacidade e segurança, apresenta informações fictícias.

A tela 4.1 simula o resultado da inserção de um caractere no campo de pesquisa presente na tela inicial. Ao abrir o sistema, quando um usuário começa a inserir uma pesquisa os resultados mostram servidores e unidades acadêmicas.

A figura 4.3, apresenta uma tela simulando a exibição do perfil de um servidor após o usuário ter clicado no nome exibido na pesquisa inicial. Ao clicar nos botões que representam as categorias de cada tipo de produção é expandida uma lista com ano e título das produções pertencentes a categoria. Na parte de especialidades existe um mecanismo semelhante, porém na expansão é exibida as informações Grande Área, Área e Sub-área referentes a especialidade.

Ao clicar na aba “Dados Relevantes” o usuário visualiza a tela representada na figura 4.2. Ao clicar em alguma unidade acadêmica presente na listagem o usuário é redirecionado para a tela referente a unidade.

Na tela representada na figura 4.4 é possível visualizar a página referente a uma unidade acadêmica. Ao clicar em algum dos botões referentes aos departamentos o usuário é redirecionado para a página do departamento.

A figura 4.5 demonstra a tela relacionada a um departamento. Ao clicar em algum dos servidores pertencentes ao departamento a aplicação redireciona para a página do servidor em questão.

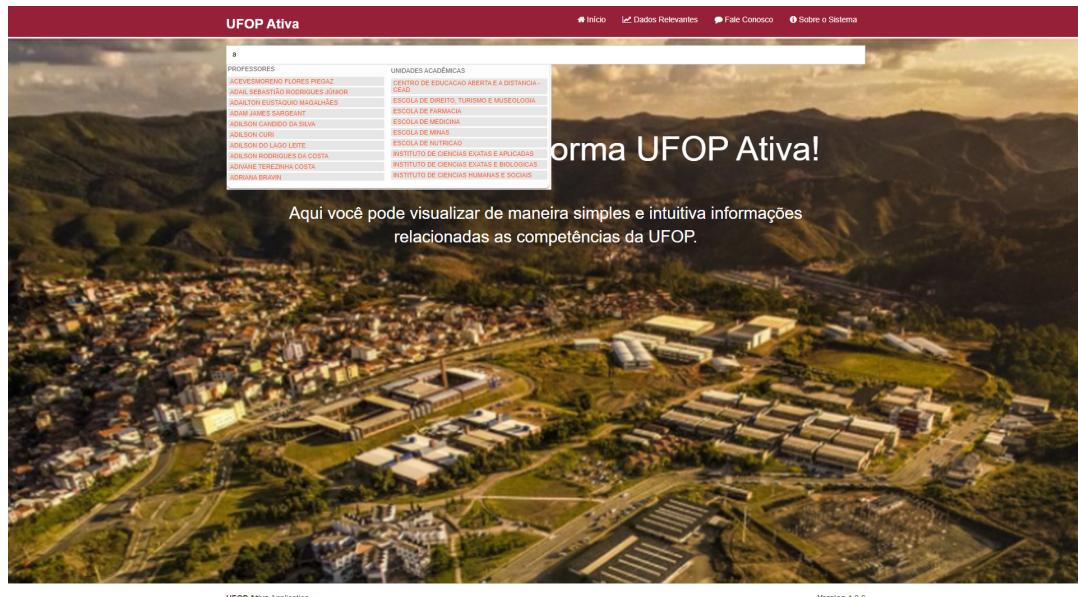


Figura 4.1 – Tela inicial da aplicação com realização de uma pesquisa.

NÚMERO DE SERVIDORES DA UFOP	1653
NÚMERO DE SERVIDORES COM LATTES	1369 (82,81%)
NÚMERO DE UNIDADES CADASTRADAS	12
NÚMERO DE DEPARTAMENTOS CADASTRADOS	137

#	Unidade
01	INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS
02	ESCOLA DE MINAS
03	INSTITUTO DE CIENCIAS HUMANAS E SOCIAIS
04	INSTITUTO DE CIENCIAS EXATAS E APLICADAS
05	INSTITUTO DE CIENCIAS SOCIAIS E APLICADAS
06	ESCOLA DE MEDICINA
07	ESCOLA DE DIREITO, TURISMO E MUSEOLOGIA
08	INSTITUTO DE FILOSOFIA ARTES E CULTURA
09	ESCOLA DE FARMACIA
10	ESCOLA DE NUTRICAO
11	CENTRO DE EDUCACAO ABERTA E A DISTANCIA
12	CENTRO DESPORTIVO

Figura 4.2 – Tela "Dados Relevantes".

**UFOP Ativa**

[Início](#) [Dados Relevantes](#) [Fale Conosco](#) [Sobre o Sistema](#)

## Servidor Docente

### João da Silva Vieira

UNIDADE: INSTITUTO DE CIÉNCIAS EXATAS E BIOLÓGICAS (ICEB)  
DEPARTAMENTO: DEPARTAMENTO DE COMPUTAÇÃO  
LATTEs: lattes.cnpq.br/0000000000000000  
ÚLTIMA ATUALIZAÇÃO DO LATTEs: 26/11/2019  
NOMES DE CITAÇÕES: Vieira, J. S./VIEIRA, João S./VIEIRA, JOÃO SILVA

[Visualizar Gráficos](#)

---

#### Formação Acadêmica

1996	Graduação	Engenharia Mecatrônica	Universidade Federal de São João Del Rey
2002	Mestrado	Ciências da Computação	Universidade Federal de Ouro Preto
2015	Doutorado	Ciências da Computação	Universidade Federal de Ouro Preto

#### Orientações Concluídas

Mestrado: 7
Doutorado: 2
Pós-Doutorado: 0
Outras: 35

---

### Projetos de Pesquisa em Andamento

2008 - Atual	Ambiente Colaborativo para Análise de Imagens
2012 - Atual	Processamento Digital de Imagens aplicados na área da saúde
2016 - Atual	Elaboração de Sistemas de Indicação

---

#### Produção Bibliográfica

- [Artigos Aceitos para Publicação \(2\)](#) +
- 2019 Study of Security Camera Images Processing Applied in Crime Identification
- 2017 CNN's Application in Biometric Recognition from Ear Imaging
- [Artigos Publicados \(13\)](#) +
- [Trabalhos em Eventos \(2\)](#) +
- [Outras Produções Bibliográficas \(40\)](#) +

#### Produção Técnica

- [Software \(1\)](#) +
- 2005 Ambiente de Melhoria de Qualidade de Imagens
- [Produto Tecnológico \(4\)](#) +
- [Outras Produções Técnicas \(31\)](#) +

---

#### Especialidades

- [Processamento de Imagens](#) +
- [Reconhecimento de Padrões](#) +

Grande Área	Área	Sub-área
Ciências Exatas e da Terra	Ciência da Computação	Metodologia e Técnicas da Computação

- [Linguagens de Programação](#) +

#### Palavras Chave

Palavra	Nº Utilizações
processamento de imagens	14
filmes ultrafinos	9
visão computacional	8
simulação computacional	8
polymers	5
dosimeter	4

Figura 4.3 – Tela de um servidor.

The screenshot shows the UFOP Ativa system interface. At the top, there is a red header bar with the text "UFOP Ativa". Below the header, the title "Unidade: INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS (ICEB)" is displayed. Underneath the title, it says "Departamentos: 7" and "Servidores: 273". A section titled "Departamentos" lists seven departments: DEPARTAMENTO DE QUIMICA, DEPARTAMENTO DE MATEMATICA, DEPARTAMENTO DE CIENCIAS BIOLOGICAS, DEPARTAMENTO DE FISICA, DEPARTAMENTO DE COMPUTACAO, DEPARTAMENTO DE BIODIVERSIDADE, EVOLUCAO E MEIO AMBIENTE, and DEPARTAMENTO DE ESTATISTICA.

Figura 4.4 – Tela de uma unidade.

The screenshot shows the UFOP Ativa system interface for the DEPARTAMENTO DE COMPUTACAO department. At the top, it displays the department name and the unit: "INSTITUTO DE CIENCIAS EXATAS E BIOLOGICAS". Below this, a section titled "Servidores (66) Por ordem alfabetica" lists 19 names, each preceded by a number from 1 to 19:

1	ALAN ROBERT RESENDE DE FREITAS
2	ALEXANDRE MAGNO DE SOUSA
3	ALVARO RODRIGUES PEREIRA JUNIOR
4	AMANDA SAVIO NASCIMENTO E SILVA
5	ANDERSON ALMEIDA FERREIRA
6	ANDREA GOMES CAMPOS BIANCHI
7	BRUNO CERQUEIRA HOTT
8	BRUNO PEREIRA DOS SANTOS
9	BRUNO RABELLO MONTEIRO
10	CARLOS ALBERTO MARQUES PIETROBON
11	CARLOS FREDÉRICIO MARCELO DA CUNHA CAVALCANTI
12	CARLOS HENRIQUE GOMES FERREIRA
13	CRISTIANO AMARO DA MATTIA
14	DARLAN NUNES DE BRITO
15	DAYANNE GOUBEIA COELHO
16	DIEGO ZUQUIM GUIMARAES
17	EDUARDO DA SILVA RIBEIRO
18	EDUARDO JOSE DA SILVA LUZ
19	ELTON JOSE DA SILVA

Figura 4.5 – Tela de um departamento.

## 5 Conclusão

Um sistema que permite aos usuários visualizar informações acerca de pesquisas realizadas na UFOP contribui para análises acadêmicas internas e externas. Neste trabalho projetou-se e desenvolveu-se uma versão inicial do UFOP Ativa, uma ferramenta que auxilia a exibição e a análise de dados acadêmicos de servidores da UFOP, por meio de informações provenientes da Plataforma Lattes. O sistema denominado permite que os usuários naveguem e acessem informações de servidores, unidades e departamentos da universidade, em um ambiente amigável e de fácil entendimento.

É nítido que o sistema colabora no acesso a informações acerca da universidade e dos seus servidores, por meio de uma interface que facilita a visualização dos dados de uma forma bem organizada. Além disso, os dados extraídos contribuem para análises que possam ser realizadas por meio do sistema.

É notável que a base de dados de currículos da Plataforma Lattes é uma base de pesquisadores bem completa e comprehende uma vasta quantidade de informações diversificadas. Sendo assim, essas informações podem ser exploradas e manipuladas para o levantamento de diversos estudos e dados estatísticos relevantes, para as instituições de ensino em geral.

A elaboração deste projeto contribuiu substancialmente para o conhecimento na área estudada, no desenvolvimento de pesquisas científicas e nas tecnologias utilizadas para o desenvolvimento do mesmo. É importante ressaltar que ainda há propriedades a serem exploradas dentro do sistema construído. A elaboração deste projeto é de grande valia para a UFOP e pode ser evoluído por meio de melhorias e adição de funcionalidades. O software apresentado é apenas um versão inicial de uma ferramenta que é extremamente significativa para a instituição.

## Trabalhos Futuros

Como trabalhos futuros existem módulos levantados na especificação a serem implementados e possibilidades de incrementos ao sistema desenvolvido neste trabalho. Primeiramente, a perspectiva de usuário administrador destaca-se como um módulo bem interessante a ser explorado, visto que agrega muito valor principalmente para as avaliações internas da UFOP.

Sobre o sistema construído até então, seria interessante adaptá-lo para que a extração de informações possa ser realizada de maneira automática periodicamente (ou manualmente quando se julgar necessário), dessa forma, manter sempre o sistema atualizado. O mapeamento de coautoria de uma servidor é outra tarefa a ser explorada, construindo redes por meio de grafos e identificando coautores que atuam na própria UFOP e autores considerados externos.

Outro recurso relevante é o mapeamento completo das especialidades de um servidor,

visto que, no sistema atual foram mapeadas apenas as especialidades que estão relacionadas ao elemento *Areas*, referentes às áreas de atuação de um servidor. Além disso, é pertinente uma refatoração da implementação do mapeamento das informações extraídas do XML para os objetos que representam o banco da aplicação.

A adição de um módulo relacionado a produtividade interna é outra possibilidade de evolução para o sistema construído. Além disso, a inclusão de pessoas externas por meio de um CPF fornecido como entrada pode facilitar análises curriculares e outras ponderações.

É possível explorar também uma estratégia que é de extrema importância dentro da área de Engenharia de Software, que é a realização de testes. A elaboração de testes unitários e de integração para aplicação irão garantir uma maior cobertura de erros, principalmente quando for implementado o módulo de administrador. É imprescindível também trabalhar a parte de segurança do sistema, para não surgir muitos problemas neste âmbito futuramente.

# Referências

- ABITEBOUL, S. Querying semi-structured data. In: *Proceedings of the 6th International Conference on Database Theory*. London, UK, UK: Springer-Verlag, 1997. (ICDT '97), p. 1–18. ISBN 3-540-62222-5. Disponível em: <<http://dl.acm.org/citation.cfm?id=645502.656103>>.
- ALMEIDA, M. B. Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares. *Cia da Informação*, scielo, v. 31, p. 5 – 13, 08 2002. ISSN 0100-1965. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-19652002000200001&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652002000200001&nrm=iso)>.
- BRADLEY, N. *The XML Companion*. 3rd ed.. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002. ISBN 0-201-77059-8.
- BUNEMAN, P. Semistructured data. In: *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. New York, NY, USA: ACM, 1997. (PODS '97), p. 117–121. ISBN 0-89791-910-6. Disponível em: <<http://doi.acm.org/10.1145/263661.263675>>.
- CHEN, P. P.-S. The entity-relationship model—toward a unified view of data. *ACM Transactions on Database Systems (TODS)*, Acm, v. 1, n. 1, p. 9–36, 1976.
- CROSS, D.; THOMSON, S.; SINCLAIR, A. Research in brazil: a report for capes by clarivate analytics. 2018. v. 1, 2018. Disponível em: <<http://www.capes.gov.br/images/stories/download/diversos/17012018-CAPES-InCitesReport-Final.pdf>>.
- DIAS, T.; MOITA, G.; DIAS, P. Adoção da plataforma lattes como fonte de dados para caracterização de redes científicas. *Encontros Bibli: revista eletrônica de biblioteconomia e ciência da informação*, v. 21, n. 47, p. 16–26, 2016. ISSN 1518-2924. Disponível em: <<https://periodicos.ufsc.br/index.php/eb/article/view/1518-2924.2016v21n47p16>>.
- DIGIAMPIETRI, L.; MENA-CHALCO, J.; PéREZ-ALCÁZAR, J.; TUESTA, E.; DELGADO, K.; MUGNAINI, R.; SILVA, G.; LIMA, J. Extração, caracterização e análises de dados de currículos lattes. *Revista Eletrônica de Sistemas de Informação*, v. 14, n. 2, 2015. ISSN 1677-3071. Disponível em: <<http://www.periodicosibepes.org.br/index.php/reinfo/article/view/1528>>.
- DIGIAMPIETRI, L. A.; MENA-CHALCO, J. P.; PéREZ-ALCÁZAR, J. d. J.; TUESTA, E. F.; DELGADO, K. V.; MUGNAINI, R.; SILVA, G. S. Minerando e caracterizando dados de currículos lattes. In: *Brazilian Workshop on Social Network Analysis and Mining - BraSNAM*. [S.l.: s.n.], 2012.
- DUARTE, L. *Deduplicação de Dados*. 2019. Disponível em: <<https://comoaprenderwindows.com.br/backup/deduplicacao-de-dados/>>.
- DUARTE, V. M. d. N. *Pesquisa Científica: Interesses e Motivações*. 2012. Disponível em: <<http://brasileesco.la/m14568>>.
- EDUCATION, B. T. *Curso de Scrum. Por que estudar?* 2018. Disponível em: <<https://blog.trainning.com.br/projetos/curso-de-scrum/>>.

- ELMASRI, R.; NAVATHE, S. B.; PINHEIRO, M. G. et al. Sistemas de banco de dados. Pearson Addison Wesley São Paulo, 2005.
- FILHO, W. de P. P. *Engenharia de software*. [S.l.]: LTC, 2003. v. 2.
- GOOGLE. *One framework mobile desktop*. 2016. Disponível em: <<https://angular.io/>>.
- IPA, C. U. M. *Preenchendo o Currículo Lattes*. 2016. Disponível em: <[http://ipametodista.edu.br/pos-graduacao/artigos-e-documentos/doc\\_dsc\\_nome\\_arqui20060912181743.pdf/view](http://ipametodista.edu.br/pos-graduacao/artigos-e-documentos/doc_dsc_nome_arqui20060912181743.pdf/view)>.
- JARDIM, R. *Diagrama Entidade Relacionamento - DER*. 2008. Disponível em: <<https://sites.google.com/site/fkbancodedados1/modelodados/der>>.
- JOHNSON, R. *Expert one-on-one J2EE design and development*. [S.l.]: John Wiley & Sons, 2004.
- LATTES, P. *Extração de Dados*. 2010. Disponível em: <<http://lattes.cnpq.br/web/plataforma-lattes/extracao-de-dados/>>.
- MELLO, R. d. S.; DORNELES, C. F.; KADE, A.; BRAGANHOLO, V.; HEUSER, C. A. Dados semi-estruturados. *XV Simpósio Brasileiro de Banco de Dados*, 2000.
- MENA-CHALCO, J. P.; JUNIOR, R. M. C. scriptlattes: an open-source knowledge extraction system from the lattes platform. *Journal of the Brazilian Computer Society*, v. 15, n. 4, p. 31–39, Dec 2009. ISSN 1678-4804. Disponível em: <<https://doi.org/10.1007/BF03194511>>.
- NETBEANS. *Vinculando WSDL a Java com JAXB*. 2011. Disponível em: <[https://netbeans.org/kb/74/websvc/jaxb\\_pt\\_BR.html](https://netbeans.org/kb/74/websvc/jaxb_pt_BR.html)>.
- NETO, A. B. d. C. A importância da pesquisa científica no brasil. 2017. Disponível em: <<https://www.passeidireto.com/arquivo/53564343/artigo-a-importancia-da-pesquisa-cientifica-no-brasil>>.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [s.n.], 2016. Disponível em: <[https://books.google.com.br/books?hl=pt-BR&lr=&id=wexzCwAAQBAJ&oi=fnd&pg=PR1&dq=engenharia+de+software&ots=0N-FqKLB\\_W&sig=Fj1ebDIXEmfDfVLaxuRVrSC8mVU#v=onepage&q=engenharia\%20de\%20software&f=false](https://books.google.com.br/books?hl=pt-BR&lr=&id=wexzCwAAQBAJ&oi=fnd&pg=PR1&dq=engenharia+de+software&ots=0N-FqKLB_W&sig=Fj1ebDIXEmfDfVLaxuRVrSC8mVU#v=onepage&q=engenharia\%20de\%20software&f=false)>, publisher={McGrawHillBras}.
- PROPP. *Planilha de Produtividade*. 2016. Disponível em: <<https://propp.ufop.br/pesquisa/planilha-de-produtividade>>.
- SOMOS. *Somos UFMG*. 2017. Disponível em: <<http://somos.ufmg.br/>>.
- VERCESI, A.; HOGAN, D. J.; CHAMBOULEYRON, I.; MARTÍNEZ, J. M.; ARRUDA, J. R. d. F.; ALVES, O. L.; ARRUDA, P.; HOFFMANN, R. Desafios da pesquisa no brasil: uma contribuição ao debate. *São Paulo em Perspectiva*, scielo, v. 16, p. 15 – 23, 10 2002. ISSN 0102-8839. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0102-88392002000400004&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0102-88392002000400004&nrm=iso)>.
- W3C. *eXtensible Markup Language (XML)*. 2008. Disponível em: <<https://www.w3.org/XML/>>.