
BioBD@PUC-Rio

Busc@NIMA

Arquitetura de Software

Versão 1.2

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

Histórico da Revisão

Data	Versão	Descrição	Autor
17/06/2020	1.0	Versão Inicial	Veronica dos Santos
30/06/2020	1.1	Engenharia Reversa da aplicação de busca	Veronica dos Santos
07/07/2020	1.2	Inclusão do fluxo do Git enviado pela Mariana	Veronica dos Santos

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

Índice Analítico

Introdução	4
Finalidade	4
Definições, Acrônimos e Abreviações	4
Visão Geral	6
Detalhamento Técnico dos Requisitos Não-funcionais	7
Segurança	7
Usabilidade	8
Confiabilidade	8
Desempenho	9
Portabilidade e Compatibilidade	9
Disponibilidade	10
Dados e Armazenamento	10
Verificação e Validação	11
Componentes de solução	11
Banco de Dados	11
AllegroGraph	11
Visão Conceitual de Dados	15
Visão Lógica de Dados	17
Ferramenta de ETL	20
Linked Pipe ETL	20
Pipeline CV Lattes	22
Pipeline de Dados de Cadastro	27
Pipeline Disciplinas	28
Ferramenta de Busca	31
Visão de Implantação	36
Controle de Versão	37

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

Arquitetura de Software

1. Introdução

Este documento oferece uma visão arquitetural abrangente da ferramenta Busc@NIMA. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação aos componentes utilizados para atender aos requisitos funcionais e não funcionais além de características e restrições especificadas no documento *Especificação dos Requisitos de Sistema*.

1.1. Finalidade

Esse documento serve de entrada para as decisões de projeto técnico, codificação além de fornecer insumos adicionais para o planejamento dos testes de sistema e de aceitação.

1.2. Definições, Acrônimos e Abreviações

Cloud Computing

Computação em nuvem é uma solução de TI que permite a disponibilidade, sob demanda, de serviços de informática através da internet sem o gerenciamento ativo direto do utilizador. Há três tipos de nuvem: a pública, a privada e a híbrida. Cada uma é voltada às diferentes necessidades das empresas. A nuvem privada do Departamento de Informática (Cloud-DI) oferece serviços do tipo IaaS (Infrastructure as a Service), PaaS (Platform as a Service) e SaaS (Software as a Service) para a comunidade de professores e alunos do departamento.

DTD

Document Type Definition

Documento que descreve as regras de elementos, hierarquia e atributos, ou seja, um esquema para um arquivo XML.

N-Quads, N-Triples

Formato de texto simples baseado em linha para codificar um conjunto de dados RDF.

NoSQL

Not Only SQL

Tecnologias de banco de dados não relacionais que visam satisfazer os requisitos de alta disponibilidade e escalabilidade. Ao contrário das tecnologias de banco de dados relacional que suportam modelos de dados com esquema único (*schema full*) que precisa estar definido antes da carga de dados (esquema na gravação), os repositórios NoSQL adotam modelos de dados mais flexíveis que podem ser sem esquema único ou com vários esquemas (*schema less*) onde os

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

dados podem ser interpretados somente no nível do aplicativo (esquema na leitura).

OWL

Web Ontology Language

Linguagem da Web Semântica projetada para representar conhecimento rico e complexo sobre coisas, grupos de coisas e relações entre elas, recomendação da W3C. OWL é uma linguagem baseada em lógica computacional projetada para processamento por máquinas, sendo mais adequada para esse uso do que XML, RDF e RDFS (RDF Schema), por fornecer vocabulário adicional com uma semântica formal. A OWL foi baseada nas linguagens OIL e DAML+OIL.

RDF / XML

Formato XML para RDF em termos de namespaces em XML, o conjunto de informações XML e a base XML.

RDFS

RDF Schema

Um Esquema RDF é construído sobre o modelo de grafo RDF e é especificado como um conjunto de classes, propriedades e restrições entre seus relacionamentos. Um Esquema RDF permite, por exemplo, que recursos possam ser instâncias de uma ou mais classes através da propriedade *rdf:type*. Com RDFS também é possível organizar classe de forma hierárquica através da propriedade *rdfs:subClassOf* para especificar tais relacionamentos entre classes.

SPARQL

SPARQL Protocol and RDF Query Language

Linguagem padronizada pela W3C para a consulta de grafos RDF.

TriG

Sintaxe textual para RDF que permite que um conjunto de dados RDF seja completamente escrito em um formato de texto compacto e natural, com abreviações para padrões de uso e tipos de dados comuns. TriG é uma extensão do formato Turtle.

TriX

Formato XML para RDF. O grafo nomeado é um conjunto de triplas nomeados por um URI. Esse URI pode ser usado fora ou dentro do grafo para fazer referência a ele. TriX é uma serialização para gráficos nomeados. O TriX visa fornecer uma representação XML altamente normalizada e consistente para gráficos RDF, permitindo o uso efetivo de ferramentas XML genéricas, como XSLT, XQuery, etc.

Turtle

Sintaxe textual para RDF. O grafo RDF pode ser completamente escrito em um formato de texto compacto e natural, com abreviações para padrões de uso e tipos de dados comuns. O Turtle fornece níveis de compatibilidade com o formato N-Triplos, bem como a sintaxe de padrão triplo

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

da Recomendação SPARQL W3C.

URI

Uniform Resource Identifier

Utilizado para identificar unicamente recursos na Web. Os URIs podem ser criados para referir-se a qualquer coisa que precise ser referenciada em uma declaração, não somente a localização de recursos com endereços na Web (URL). A identificação de sujeitos, predicados e objetos em declarações RDF é feito com base na referências URIs, sendo esse um URI juntamente com um identificador de fragmento opcional no final.

URL

Uniform Resource Locator

Endereço de um recurso na Web, uma sequência de caracteres que identifica a localização de uma página Web.

VPN

Virtual Private Network

Rede de comunicações entre computadores e outros dispositivos que têm acesso restrito a quem tem as credenciais necessárias.

XSLT

eXtensible Stylesheet Language Transformations

Linguagem de marcação XML usada para criar documentos XSL que são usados para transformar a apresentação dos documentos XML nos browsers e outros aplicativos que os suportem assim como para converter arquivos XML em outros formatos.

Outras definições podem ser encontradas no documento de *Especificação de Requisitos de Sistema*.

1.3. Visão Geral

No item 2 serão detalhados os requisitos não funcionais elencados no item 2.2 do documento *Especificação de Requisitos de Sistema*. O item 3 estabelece as orientações quanto a Verificação e Validação dos artefatos de software. O item 4 apresenta os 3 principais componentes da solução: o banco de dados NoSQL, a ferramenta de ETL e a aplicação de busca por palavras chaves. No item 5 é apresentada a visão de Implantação do sistema Busc@NIMA.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

2. Detalhamento Técnico dos Requisitos Não-funcionais

2.1. Segurança

[RNF1] Restringir a troca de informações via browser do usuário
O software não deve transportar informações do negócio na URL de ativação das páginas dinâmicas (método GET). Essas informações devem ser transportadas, obrigatoriamente, via POST.
[RNF2] Apenas a equipe de desenvolvimento pode acessar os servidores e software backend do produto
<p>O acesso aos servidores de aplicação, banco de dados e ferramenta de ETL são restritos a equipe de desenvolvedores do BioBD.</p> <p>Os usuários finais terão acesso somente a aplicação de busca via browser.</p> <p>As senhas de acesso serão guardadas em repositório de senhas criptografado.</p>
[RNF3] Restringir as permissões de acesso aos componentes/ambientes com perfil adequado aos usuários e funções
<p>Os usuários administradores serão usados somente para ações que requerem alto nível de permissão nos servidores.</p> <p>A aplicação de consulta fará acesso ao banco usando um usuário com acesso somente de consulta aos repositórios.</p> <p>A ferramenta de ETL deverá ser configurada para ter acesso as pastas de arquivos locais para gravação e leitura mas não deverá ter acesso remoto habilitado na porta 80, sendo exclusivamente executada via VPN com o servidor.</p>
[RNF4] Proteger a entrada de dados da ferramenta de consulta de ataques
A entrada de dados das palavras de busca deve ser construída de modo a impedir ataques como SQL Injection

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

Não está previsto o uso de captcha para evitar ataques DDoS

2.2. Usabilidade

[RNF5] Uso de design responsivo na interface gráfica da ferramenta de busca

O design responsivo da web é uma abordagem de desenvolvimento que torna as páginas da web renderizadas bem em uma variedade de dispositivos e tamanhos de janela ou tela.

As telas da ferramenta de busca deverão se adequar ao dispositivo de acesso (desktop, smartphone ou tablet)

[RNF6] Utilizar hints nos campos e botões de navegação da ferramenta de busca

[RNF7] Utilizar logotipo da PUC-Rio, do NIMA e do Departamento de Informática na parte superior da ferramenta de busca



2.3. Confiabilidade

[RNF8] Configurar os componentes da ferramenta de busca para restart automático em caso de queda da VM ou outros problema de infra estrutura

Criar um script de start e outro de stop com os comandos adequados e com registro em arquivo de log local do servidor para incluir nos procedimentos de boot da máquina

You can start AllegroGraph by running:
`/home/cloud-di/ag-6.4.2/bin/agraph-control --config ag-6.4.2/lib/agraph.cfg start`

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

You can stop AllegroGraph by running:
/home/cloud-di/ag-6.4.2/bin/agraph-control --config ag-6.4.2/lib/agraph.cfg
stop

[RNF9] Criar página de erro padrão para ferramenta de busca

Qualquer falha na aplicação que possa ocorrer quando o usuário estiver navegando na ferramenta de busca deve ser direcionada para uma página de erro padrão com a seguinte mensagem:

“Aconteceu um erro inesperado. Por favor, tente novamente mais tarde. Caso o erro persista, entre em contato com xxxxxxxxx”

[RNF10] Realizar backup periódico das máquinas virtuais que atendem ao projeto, contemplando principalmente a base de dados e os arquivos de carga

O processo de backup já é realizado pela infraestrutura da Cloud do DI

2.4. Desempenho

[RNF11] Exibir o resultado da consulta na ferramenta de busca em no máximo 2 minutos, sendo o tempo desejável de 15 segundos.

O AllegroGraph possui duas diretivas de timeout: queryTimeoutquery (default sem timeout) e serviceTimeoutquery (default 5 minutos)

[RNF12] Suportar no mínimo 5 usuários simultâneos na ferramenta de busca sem degradação no tempo de resposta

O AllegroGraph não tem configuração de mínimo ou máximo de processos de usuário

2.5. Portabilidade e Compatibilidade

[RNF13] Suportar o uso da ferramenta de busca nos principais browsers de dispositivos com sistema operacional Windows, Linux e Android

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

O software deve ser compatível com os browsers a seguir nas versões especificadas ou superiores: IE (x.x), Google Chrome (x.x), Firefox (x.x),

[RNF14] Suportar a carga de dados a partir de arquivos em diferentes formatos como XML, RDF, banco de dados, CSV e planilhas Excel

A ferramenta de ETL deve possuir componentes de leitura de diversos formatos de dados

2.6. Disponibilidade

[RNF15] O sistema deve estar disponível 99% do tempo.

A disponibilidade do sistema depende da disponibilidade dos serviços da Cloud do DI.

2.7. Dados e Armazenamento

[RNF16] Suportar modelo de dados flexível

Em função da variedade de fonte de dados que irão compor a base de dados que suporta a ferramenta de busca não será possível estabelecer uma modelagem rígida a priori. A facilidade de manutenção do esquema do banco de dados visa reduzir o custo da evolução do esquema.

Um modelo de dados sem esquema rígido permite que os dados tenham estruturas arbitrárias, pois não são explicitamente definidos por uma linguagem de definição de dados (esquema na gravação). Em vez disso, eles são implicitamente codificados pela lógica do aplicativo (esquema na leitura). esses aplicativos têm uma evolução constante do esquema de dados e se beneficiam do modelo de dados flexível.

Os dados serão armazenados em um NoSQL store com esquema flexível.

[RNF17] Suportar indexação de dados em formato texto

A indexação de textos possibilita a busca por palavras-chave e recuperação de informações em documentos de textos de forma eficiente.

Através da indexação é possível representar uma grande coleção de documentos em forma de texto com uma descrição abreviada do conteúdo de cada documento integrante. A recuperação de informação (RI) consiste em encontrar informações não-estruturadas a partir do fornecimento de parte da informação desejada (palavras-chave), também conhecidos como

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

descritores de busca.

Os dados serão armazenados em um NoSQL store com suporte a busca por conteúdo texto (e não só busca por chaves).

3.Verificação e Validação

Os documentos Especificação de Requisitos de Sistema e Arquitetura de Software (artefatos de software) serão submetidos a validação (para identificar se estamos construindo o produto certo) e verificação (para identificar se estamos construindo certo o produto). A validação e verificação estática permite a identificação de erros e defeitos de especificação. Por exemplo, a verificação de requisitos funcionais pode ser usada para detectar se os mesmos são consistentes entre si e estão corretamente especificados. A técnica utilizada para Detecção de Defeitos é *Ad-hoc*, onde um membro da equipe lê o documento de acordo com sua perspectiva para identificar omissão, fato incorreto, inconsistência, ambiguidade ou informação estranha (taxonomia padrão IEEE 830, 1998).

As eventuais falhas de softwares da ferramenta de busca e dos processos de conversão e carga de dados serão tratadas no Planejamento e Execução dos Testes de Sistema.

4.Componentes de solução

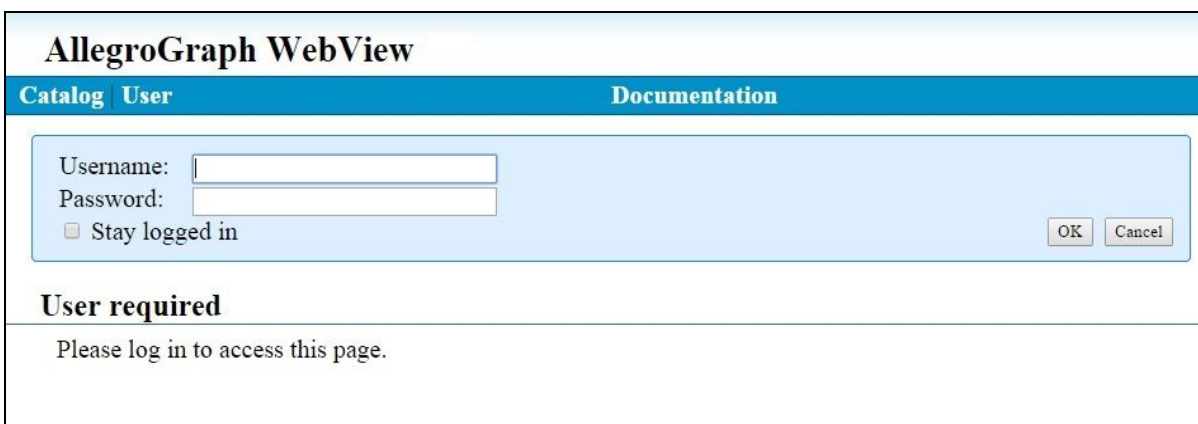
4.1. Banco de Dados

4.1.1. AllegroGraph

AllegroGraph, em sua versão 6.4.2, é um Triple Store, um tipo específico de NoSQL Graph Store. Os grafos no modelo RDF são armazenados na forma de triplas (também chamados de *assertions*) assim especificados: sujeito (s), predicado (p) e objeto (o). Cada tripla também recebe o campo grafo (g) e um ID (i).

Além do armazenamento, o produto oferece: (1) ferramenta de linhas de comando (*agtool*) para carga, exportação, consulta, ..., (2) ferramenta Web para administração - AllegroGraph WebView (figura 1) que permite a criação de usuários, carga e exportação de dados, backup, execução de consultas,, (3) integração com softwares de terceiros como Apache Solr, MongoDB, Anaconda, ... e (4) ferramenta desktop para consulta, atualização e visualização do grafo armazenado nos repositórios - Gruff.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	



AllegroGraph WebView

Catalog User Documentation

Username:

Password:

☐ Stay logged in

OK Cancel

User required

Please log in to access this page.

Figura 1 - AllegroGraph WebView

Porém o AllegroGraph não restringe o conteúdo dos repositórios a triplas com RDF puro. É possível representar qualquer estrutura de dados de grafo tratando nós como sujeitos e objetos, arestas como predicados e criando um tripla para cada aresta. Com essa característica, apesar do AllegroGraph ser um Triple Store, ele também pode ser usado como um banco de dados orientado a gráficos genérico.

Em relação a bancos de dados relacionais, a escolha por um banco de dados de grafo para atender ao projeto se deu: (1) pela oportunidade de explorar a tecnologia de bancos de dados NoSQL e (2) pela necessidade de flexibilidade na definição evolutiva do schema de dados da aplicação, que acrescenta novas classes e predicados a medida que novas fontes de dados são incorporadas [RNF16].

De modo a otimizar a busca por triplas, o AllegroGraph realiza automaticamente a criação de sete índices. Cada índice utiliza uma ordem de classificação das triplas, por exemplo, o índice spogi classifica primeiro o sujeito (s), depois predicado, objeto, gráfico e, finalmente, o id. O conjunto padrão de índices é: spogi, posgi, ospgi, gspoi, gposi, gospi e i (figura 2). A partir dessa configuração é possível criar ou remover índices, por exemplo, se seu aplicativo não usar subgrafos, os índices iniciados com "g" nunca serão usados e podem ser eliminados, otimizando a carga de dados. A indexação de triplas contribui para o atendimento aos requisitos [RNF11] e [RNF12] na recuperação de dados do pesquisador.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

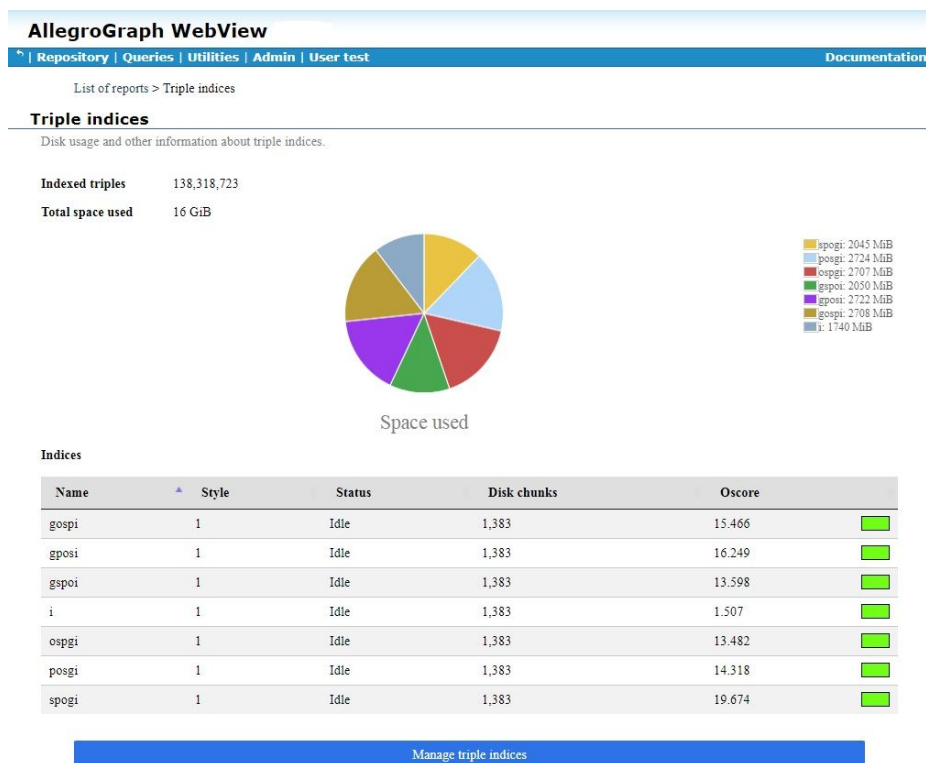


Figura 2 - Indexação de Triplas

O AllegroGraph, na versão 6.4.2, permite carregar dados grafos RDF nos formatos N-Quads, N-Triples, RDF / XML, TriG, TriX e Turtle. A partir da versão 7 foi incluído o suporte a carga de JSON e CSV, o que passou a classificá-lo como Multi-modelo. Para recuperar as triplas armazenadas é possível utilizar as interfaces de consulta: SPARQL, Prolog e o raciocínio RDFS ++, sendo que o subsistema SPARQL da AllegroGraph é aderente ao padrão W3C, incluindo um otimizador de consultas, e tem suporte completo para grafos nomeados.

Allegro Graph suporta nativamente a indexação de texto livre assim como possui integração com o Apache Solr, uma plataforma de indexação / pesquisa de texto livre de código aberto do projeto Apache Lucene. A indexação de texto livre permite mapear rapidamente palavras e frases para triplas, quer seja em sujeitos, predicados ou objetos. De acordo com o fornecedor, a indexação nativa possui uma API simples, suporta várias funcionalidades como múltiplos idiomas, *stemming*, lista de *stopwords*, sinônimos, palavras reservadas e também não requer sincronização externa como no caso do Solr.

A criação de índices pode especificar um ou mais predicados específicos, incluindo uma opção para indexar todos os predicados, conforme figura 3. Além disso, um índice pode ser configurado para incluir:

1. Todos os literais, nenhum literal ou tipos específicos de literais.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

2. O URI completo de um recurso, apenas o nome local do recurso (após o # ou /) ou ignorar completamente os URIs do recurso.
3. Qualquer combinação das quatro partes de uma "tripla": sujeito, predicado, objeto e gráfico.

Com o suporte de um índice de texto livre, é possível realizar buscas otimizadas usando: (1) expressões booleanas ("sustentável" AND "habitação"), (2) curingas ("ciências *" OR "engenharia *") e (3) expressões exatas ("meio ambiente"). Porém, é importante considerar que a criação desse tipo de índice afeta o processo de carga/atualização de dados. De acordo com o fornecedor, pode ocorrer uma redução entre 5 e 25%, dependendo do número de predicados envolvidos e dos tipos de dados de texto existentes na base de dados. A indexação de texto contribui para o atendimento aos requisitos [RNF11] e [RNF12] na busca por palavras chaves.

Essas funcionalidades atendem a versão inicial da ferramenta de busca para realizar a busca simples por palavras chaves [RNF17], [RF1]. Na próxima versão o Solr poderá ser melhor explorado para avaliar as funcionalidades que não estão presentes na indexação nativa como classificação de relevância, encontrar palavras próximas umas das outras, pesquisa facetada, *clustering*, etc ...

Para cada tripla, além do grafo (g) e do ID (i) é possível associar atributos como pares de chave-valor. As chaves referem-se às definições de atributo que devem ser adicionadas no momento da carga e não podem ser alteradas depois. Os usos possíveis sugeridos pelo fornecedor para atributos incluem: (1) controle de acesso para restringir que um usuário acesse triplas com certos atributos e (2) fragmentação para garantir que triplas relacionadas sejam sempre armazenadas juntas no mesmo fragmento. Um futuro uso de interesse do sistema pode ser registrar a fonte da qual uma tripla foi importada [RF5], [RF6], [RF7] e identificar as triplas incluídas pelo próprio pesquisador [RF12].

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

Figura 3 - Parâmetros de configuração de Indexação de Texto Livre nativo

4.1.2. Visão Conceitual de Dados

A figura 4 apresenta o Diagrama de Classes de Domínio. As classes, relacionamentos e atributos foram mapeados para Ontologias de Domínio, conforme indicam os prefixos. A relação de ontologias e os respectivos são as mesmas que constam na descrição do requisito funcional [RF8]. As fontes de dados dessas informações são a plataforma Lattes e o Sistema de Cadastro de Funcionários da PUC-Rio [UC4], [UC5]. As entidades, relacionamento e atributos das informações extraídas do SAU ainda estão em fase de modelagem e construção do processo de ETL[UC6] e serão incorporados ao diagrama por engenharia reversa.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

4.1.3. Visão Lógica de Dados

O AllegroGraph permite a separação de conjunto de triplas em repositórios. A distribuição Free que está em uso possui a limitação de 5 milhões de triplas e 3 servidores e foi necessário separar o repositório de triplas de CV Lattes em 3, sendo eles: carga-lattes-alunos, carga-lattes-professores e carga-lattes-professores2. Esses 3 repositórios logicamente representam o grafo CV Lattes.

O mecanismo de federação da AllegroGraph facilita a conexão com vários repositórios (na mesma instância, em várias instâncias na mesma máquina ou várias instâncias em um cluster de máquinas) tratando como uma única conexão a um repositório federado e realizando consultas em paralelo. Esse mecanismo poderá ser utilizado para que esses 3 repositórios sejam acessados pela aplicação como sendo um único repositório federado.

Além do grafo CV Lattes, foram criados mais dois repositórios: matriculas_puc_etl, que representa o grafo Matrículas, e disciplinas_puc_etl, que representa o grafo Disciplinas. As classes e predicados presentes em cada repositório são apresentadas na tabela 1.

Grafos		
CV Lattes	Matrículas	Disciplinas
Repositórios		
carga-lattes*	matriculas_puc_etl	disciplinas_puc_etl
Classes		
Total: 20 Em uso: 13	Total: 0 Em uso: 0	Total: 3 Em uso: 0
vivo:Student (carga-lattes-alunos) vivo:FacultyMember (carga-lattes-professores*) foaf:Document foaf:Organization foaf:Group foaf:Person		ccso:Course ccso:Professor ccso:Syllabus

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

bibo:Book bibo:Conference bibo:Periodical bibo:Journal bibo:Proceedings bibo:Chapter bibo:Article bibo:Thesis <i>cnpqterms:Especialidade</i> <i>cnpqterms:subArea</i> <i>cnpqterms:Area</i> <i>cnpqterms:grandeArea</i> <i>doac:LanguageSkill</i> <i>geo:SpatialThing</i> <i>bio:Birth</i>		
Predicados		
Total: 47 Em uso: 28	Total: 6 Em uso: 6	Total: 7 Em uso: 0
foaf:name foaf:mbox foaf:homepage foaf:identifier foaf:member foaf:citationName foaf:topic_interest dcterms:issued dcterms:isPartOf dcterms:publisher dcterms:subject dcterms:isReferencedBy bibo:pageStart bibo:pageEnd bibo:issuer	foaf:name foaf:mbox foaf:homepage owl:sameAs lattes matricula	<i>foaf:name</i> <i>ccso:csName</i> <i>ccso:code</i> <i>ccso:hasInstructor</i> <i>ccso:description</i> <i>ccso:hasSyllabus</i> <i>rdf:type</i>

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

bibo:presentedAt bibo:issn bibo:identifier bibo:degree bibo:isbn bibo:doi bio:biography dc:title dc:creator dc:type dc:contributor dc:language skos:prefLabel rdf:type <i>rdf:first</i> <i>rdf:rest</i> <i>bio:date</i> <i>bibo:authorList</i> <i>bibo:volume</i> <i>doac:comprehension</i> <i>doac:writes</i> <i>doac:reads</i> <i>doac:speaks</i> <i>doac:language</i> <i>doac:skill</i> <i>geo:place</i> <i>skos:broader</i> <i>skos:related</i> <i>bio:event</i> <i>bio:place</i>		
--	--	--

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

4.2. Ferramenta de ETL

4.2.1. Linked Pipe ETL

LinkedPipes ETL é uma ferramenta de ETL leve, especializada para conversão e carga de Linked Open Data desenvolvida por um time de pesquisadores da Faculdade de Engenharia de Software da Charles University em Praga. Nos testes dessa ferramenta foi possível verificar que a ferramenta atende aos requisitos RF5, RF6, RF7 e RF8 pois permite a conversão de vários formatos de entrada (CSV para RDF, XML para RDF, XLS para RDF, etc) e também a manipulação de dados em formato RDF usando SPARQL.

A ferramenta possui vários componentes para criação de processos automatizados de conversão e carga de dados (figura 5), a criação dos processos é realizada através de uma interface gráfica via browser por isso não requer conhecimento de linguagens de programação de baixo nível (figura 6), fazem uso de metadados para configuração dos componentes e permite a execução e acompanhamento dos histórico dos processos gerados no catálogo da própria ferramenta (figura 7), bem como podem ser instalados em servidores separados dos servidores da aplicação pois permitem a conexão a repositórios remotos, como fonte ou destino dos dados.

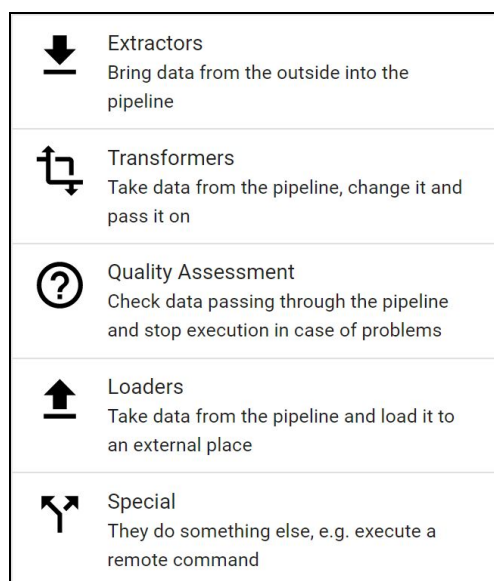


Figura 5 - Classe de Componentes de ETL da LinkedPipes

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

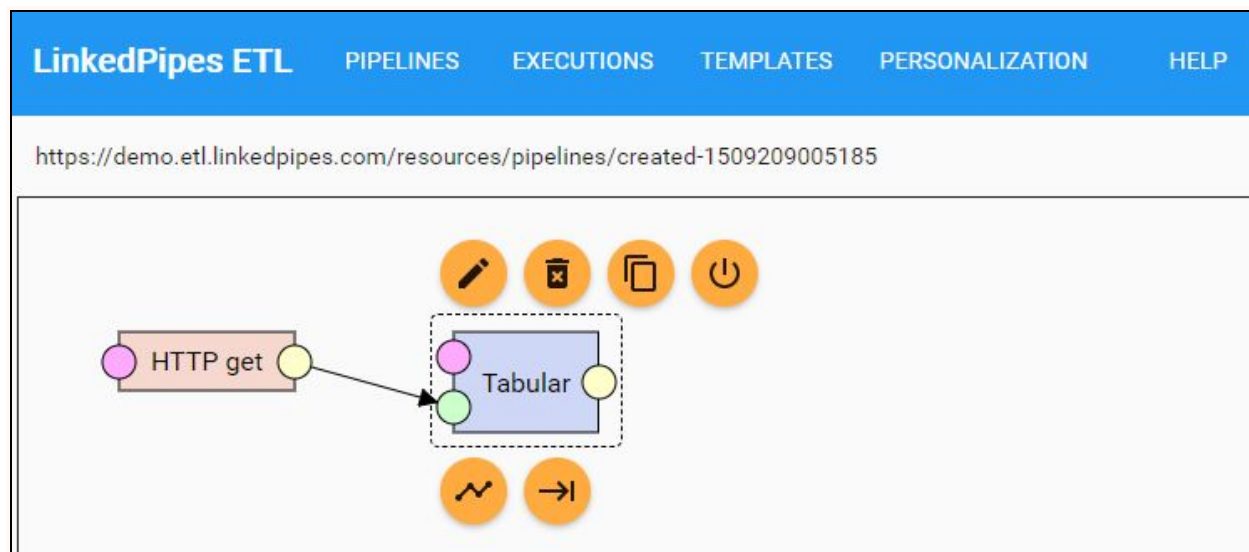


Figura 6 - Criação de Pipelines

Status	Label	Timestamp	Full execution Size	Actions
Runner	PRD - Converter Dados de Cadastro [UC5]	2020-06-20 12:53:36 - ...	Full execution	Play, Delete, Menu
Warning	Carga de dados das disciplinas a partir do arquivo Excel fornecido	2020-06-19 19:54:17, 00:00:00	Full execution Size: 0.32 mB	Play, Delete, Menu
Success	XML to RDF com lattes_professores.xml	2020-06-19 17:00:30, 00:00:01	Full execution Size: 1.57 mB	Play, Delete, Menu
Success	XML to RDF com xml2rdf3.xml	2020-06-19 16:58:11, 00:00:00	Full execution Size: 4.1 mB	Play, Delete, Menu
Success	XML to RDF com xml2rdf3.xml	2020-06-19 16:39:13, 00:00:00	Full execution Size: 4.17 mB	Play, Delete, Menu

Figura 7 - Histórico de execuções

Uma vez que a ferramenta não faz versionamento dos pipelines, o arquivo de configuração deverá ser baixado e atualizado no GitHub na pasta **carga\pipelines**. Também se faz necessário padronizar o nome dos pipelines para associar ao caso de uso que atendem e ao ambiente, uma vez que a ferramenta não tem separação de ambiente, conforme figura 8. O nome do pipeline deverá seguir o seguinte padrão:

<XXX> - <Nome do Caso de Uso> [<Id do Caso de Uso>] - <descriptor adicional> onde, XXX é o ambiente, podendo ser PRD ou DSV e o descriptor adicional poderá diferenciar quando o existir mais de um pipeline para conjunto diferente de dados.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

LinkedPipes ETL	
PIPELINES	EXECUTIONS
TEMPLATES	PERSONALIZATION
HELP	
<p>Label search</p> <p>Search for a tags.</p> <p>https://demo.etl.linkedpipes.com/resources/pipelines/1592393240494</p> <p>XSLT configuration demo</p> <p>DSV - Converter CV Lattes [UC4] - lattes_professores.xsl</p> <p>PRD - Converter Dados de Cadastro [UC5]</p> <p>DSV - Converter CV Lattes [UC4] - xml2rdf3.xsl</p> <p>DSV - Converter dados de disciplinas [UC6] - Graduação e Pós Graduação</p>	

Figura 8 - Nomenclatura padronizada para pipelines

4.2.2. Pipeline CV Lattes

O caso de uso Converter CV Lattes [UC4] teve sua primeira versão implementada com scripts bash e xslt, ou seja, sem uso da ferramenta de ETL, de acordo com o arquivo README.txt que se encontra na pasta **carga** do GitHub. Esta opção se deu para agilizar o protótipo da ferramenta de busca aproveitando o script XSLT existente desenvolvido especificamente para conversão de Lattes.

Os mapeamentos com as Ontologias de Domínio (passo 7 do caso de uso) que se encontram na tabela 2 foram identificados por engenharia reversa do script e durante esse processo foi verificado que o descritor do esquema dos CVs Lattes (**LMPLCurriculo.DTD**) disponibilizado pela Capes não corresponde corretamente a estrutura dos arquivos XML.

Elementos do LMPLCurriculo.DTD	Classes Mapeadas
DADOS-GERAIS	vivo:Student (carga-lattes-alunos)
DADOS-GERAIS	vivo:FacultyMember (carga-lattes-professores*)
CURRICULO-VITAE	foaf:Document
DETALHAMENTO-DO-TRABALHO/@NO ME-DA-EDITORIA DETALHAMENTO-DO-LIVRO/@NOME- DA-EDITORIA	foaf:Organization

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

DETALHAMENTO-DE-ORIENTACOES-C ONCLUIDAS-PARA-MESTRADO/@NOM E-DA-INSTITUICAO	
DETALHAMENTO-DE-ORIENTACOES-C ONCLUIDAS-PARA-DOCTORADO/@NO ME-DA-INSTITUICAO	
DADOS-GERAIS	foaf:Group
DADOS-GERAIS	foaf:Person
LIVRO-PUBLICADO-OU-ORGANIZADO DETALHAMENTO-DO-CAPITULO/@TIT ULO-DO-LIVRO	bibo:Book
DETALHAMENTO-DO-TRABALHO/@NO ME-DO-EVENTO	bibo:Conference
DETALHAMENTO-DO-TEXTO/@TITULO -DO-JORNAL-OU-REVISTA	bibo:Periodical
DETALHAMENTO-DO-ARTIGO/@TITUL O-DO-JORNAL-OU-REVISTA	bibo:Journal
DETALHAMENTO-DO-TRABALHO/@TIT ULO-DOS-ANAIS-OU-PROCEEDINGS	bibo:Proceedings
CAPITULO-DE-LIVRO-PUBLICADO	bibo:Chapter
ARTIGO-PUBLICADO ARTIGO-ACEITO-PARA-PUBLICACAO TRABALHO-EM-EVENTOS	bibo:Article
ORIENTACOES-CONCLUIDAS-PARA-ME STRADO MESTRADO ORIENTACOES-CONCLUIDAS-PARA-DO UTORADO DOUTORADO	bibo:Thesis
AREA-DO-CONHECIMENTO-1 AREA-DO -CONHECIMENTO-2 AREA-DO-CONHEC	<i>cnpqterms:Especialidade</i>

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

<p>IMENTO-3/@NOME-DA-ESPECIALIDADE</p> <p>../@NOME-DA-SUB-AREA-DO-CONHECIMENTO</p> <p>../@NOME-DA-AREA-DO-CONHECIMENTO</p> <p>../@NOME-GRANDE-AREA-DO-CONHECIMENTO</p> <p>IDIOMA</p> <p>DADOS-GERAIS</p>	<p><i>cnpqterms:subArea</i></p> <p><i>cnpqterms:Area</i></p> <p><i>cnpqterms:grandeArea</i></p> <p><i>doac:LanguageSkill</i></p> <p><i>geo:SpatialThing</i></p> <p><i>bio:Birth</i></p>
Atributos do LMPLCurriculo.DTD	Predicados mapeados
<p>AUTORES/@NOME-COMPLETO-DO-AUTOR</p> <p>DADOS-GERAIS/@NOME-COMPLETO</p> <p>DADOS-GERAIS/@E-MAIL</p> <p>DADOS-GERAIS/@HOME-PAGE</p> <p>DADOS-GERAIS/@NRO-ID-CNPQ</p> <p>DADOS-GERAIS</p> <p>DADOS-GERAIS/@NOME-PARA-CITACAO</p> <p>AREA-DE-ATUACAO</p> <p>DADOS-BASICOS-DO-TRABALHO/@ANO-DO-TRABALHO</p> <p>DADOS-BASICOS-DO-TEXTO/@ANO-DO-TEXT</p> <p>DADOS-BASICOS-DO-ARTIGO/@ANO-DO-ARTIGO</p> <p>DADOS-BASICOS-DO-LIVRO/@ANO</p>	<p>foaf:name</p> <p>foaf:mbox</p> <p>foaf:homepage</p> <p>foaf:identifier</p> <p>foaf:member</p> <p>foaf:citationName</p> <p>foaf:topic_interest</p> <p>dcterms:issued</p>

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

DADOS-BASICOS-DO-CAPITULO/@ANO	
DETALHAMENTO-DO-CAPITULO	dcterms:isPartOf
DETALHAMENTO-DO-TRABALHO	
DETALHAMENTO-DO-ARTIGO	
DETALHAMENTO-DO-TEXTO	
DETALHAMENTO-DO-TRABALHO/@NO ME-DA-EDITORIA	dcterms:publisher
DETALHAMENTO-DO-LIVRO/@NOME- DA-EDITORIA	
AREA-DO-CONHECIMENTO-1 AREA-DO -CONHECIMENTO-2 AREA-DO-CONHEC IMENTO-3	dcterms:subject
	dcterms:isReferencedBy
DETALHAMENTO-DO-ARTIGO/@PAGIN A-INICIAL	bibo:pageStart
DETALHAMENTO-DO-ARTIGO/@PAGIN A-FINAL	bibo:pageEnd
DETALHAMENTO-DE-ORIENTACOES-C ONCLUIDAS-PARA-MESTRADO/@CODI GO-INSTITUICAO	bibo:issner
DETALHAMENTO-DE-ORIENTACOES-C ONCLUIDAS-PARA-DOCTORADO/@CO DIGO-INSTITUICAO	
DETALHAMENTO-DO-TRABALHO/@NO ME-DO-EVENTO	bibo:presentedAt
DETALHAMENTO-DO-ARTIGO/@ISSN	bibo:issn
DETALHAMENTO-DO-TEXTO/@ISSN	
DADOS-GERAIS/@NUMERO-IDENTIFIC ADOR	bibo:identifier
ORIENTACOES-CONCLUIDAS-PARA-ME STRADO	bibo:degree

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

MESTRADO ORIENTACOES-CONCLUIDAS-PARA-DO UTORADO DOUTORADO	
DETALHAMENTO-DO-TRABALHO/@ISBN DETALHAMENTO-DO-LIVRO/@ISBN	bibo:isbn
DADOS-BASICOS-DO-CAPITULO/@DOI DADOS-BASICOS-DO-TRABALHO/@DOI	bibo:doi
RESUMO-CV/@TEXTO-RESUMO-CV-RH RESUMO-CV/@TEXTO-RESUMO-CV-RH-EN	bio:biography
DADOS-BASICOS-DO-TRABALHO/@TITULO-DO-TRABALHO DETALHAMENTO-DO-TRABALHO/@TITULO-DOS-ANAIS-OU-PROCEEDINGS DETALHAMENTO-DO-ARTIGO/@TITULO-DO-PERIODICO-OU-REVISTA DETALHAMENTO-DO-TEXTO/@TITULO-DO-JORNAL-OU-REVISTA DADOS-BASICOS-DO-TEXTO/@TITULO-DO-TEXTO DADOS-BASICOS-DO-ARTIGO/@TITULO-DO-ARTIGO-INGLES DADOS-BASICOS-DO-LIVRO/@TITULO-DO-LIVRO	dc:title
DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-MESTRADO/@NOME-DO-ORIENTADO DETALHAMENTO-DE-ORIENTACOES-CONCLUIDAS-PARA-DOUTORADO/@NOME-DO-ORIENTADO	dc:creator
@NUMERO-ID-ORIENTADOR	dc:type dc:contributor

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

@IDIOMA	dc:language
@NOME-DA-ESPECIALIDADE	skos:prefLabel
@NOME-DA-SUB-AREA-DO-CONHECIMENTO	
@NOME-DA-AREA-DO-CONHECIMENTO	
@NOME-GRANDE-AREA-DO-CONHECIMENTO	

Tabela 2 - Mapeamento UC5

O pipeline na ferramenta de ETL está sendo modelado de modo a garantir a geração do mesmo arquivo rdf de saída e usando o script genérico de conversão XML para RDF (xml2rdf3) disponível no [SourceForge](#).

4.2.3. Pipeline de Dados de Cadastro

Esse pipeline único visa atender ao caso de uso Converter Dados de Cadastro [UC5], foi configurado conforme figura 9. O componente "**Tabular**", segundo componente do pipeline e disponibilizado pela ferramenta de ETL, é um conversor de CSV para RDF que segue o padrão da W3C (Generating RDF from Tabular Data on the Web). Nos parâmetros de configuração do componente foi utilizado `http://nima.puc-rio.br#matricula.{matricula}` como o identificador da linha do arquivo CSV.

Depois da convertido foram utilizados 5 componentes "**SPARQL update**" para atender aos passos 5 ao 7 do caso de uso. Os mapeamentos com as Ontologias de Domínio se encontram na tabela 3 (onde as colunas em itálico foram removidas da conversão). O predicado *owl:sameAs* foi utilizado para criar um tripla que vincula do URI da matrícula com o URI do autor do Lattes (passo 6 do caso de uso).

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

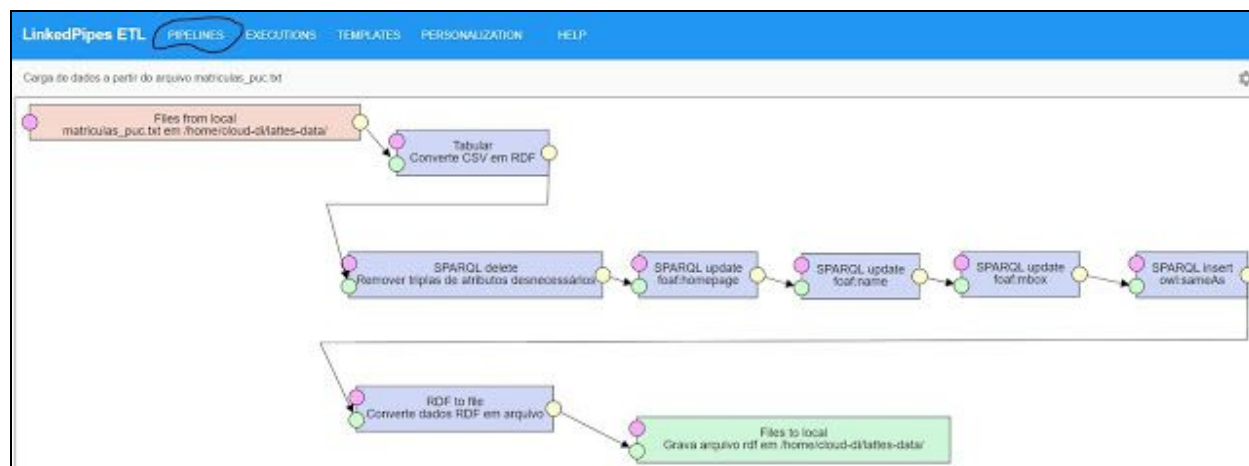


Figura 9 - Pipeline Dados de Cadastro [UC5]

Colunas do Arquivo CSV	Predicados mapeados
pessoa email home_page lattes matricula quadro usuário apelidos	foaf:name foaf:mbox foaf:homepage lattes matricula

Tabela 3 - Mapeamento UC5

4.2.4. Pipeline Disciplinas

O caso de uso Converter Dados Disciplinas [UC6] será implementado em diferentes pipelines de acordo com as colunas de cada aba da planilha gerada e os mapeamentos com as Ontologias de Domínio (passo 7 do caso de uso) se encontram na tabela 4.

Aba Graduação	
Linhas	Classes Mapeadas (rdf:type)
	ccso:Course
Colunas	Classes Mapeadas
Destaques da Ementa [C]	ccso:Syllabus

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

Professor [D-M]	ccso:Professor
	Predicados mapeados
Nome[B] Código[A]	ccso:csName ccso:code
Professor1..10 [D-M] Destaques da Ementa [C]	foaf:name, ccso:hasInstructor ccso:description, ccso: hasSyllabus

Aba Pós-Graduação	
Linhas	Classes Mapeadas (rdf:type)
	ccso:Course
Colunas	Classes Mapeadas
Destaques da Ementa [C] Professor [D]	ccso:Syllabus ccso:Professor
	Predicados mapeados
Nome[B] Código[A]	ccso:csName ccso:code
Professor [D] Destaques da Ementa [C]	foaf:name, ccso:hasInstructor ccso:description, ccso: hasSyllabus

Tabela 4 - Mapeamento UC6

O pipeline de disciplinas da Graduação e Pós Graduação foi configurado conforme figura 10 e fez uso do componente “**Excel to CSV**”, configurado como na figura 11, para converter as abas Graduação e Pós-Graduação da planilha em arquivos CSV que alimentaram os demais componentes do pipeline. Ainda existem outras abas do arquivo original a serem mapeadas para a conversão e carga.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

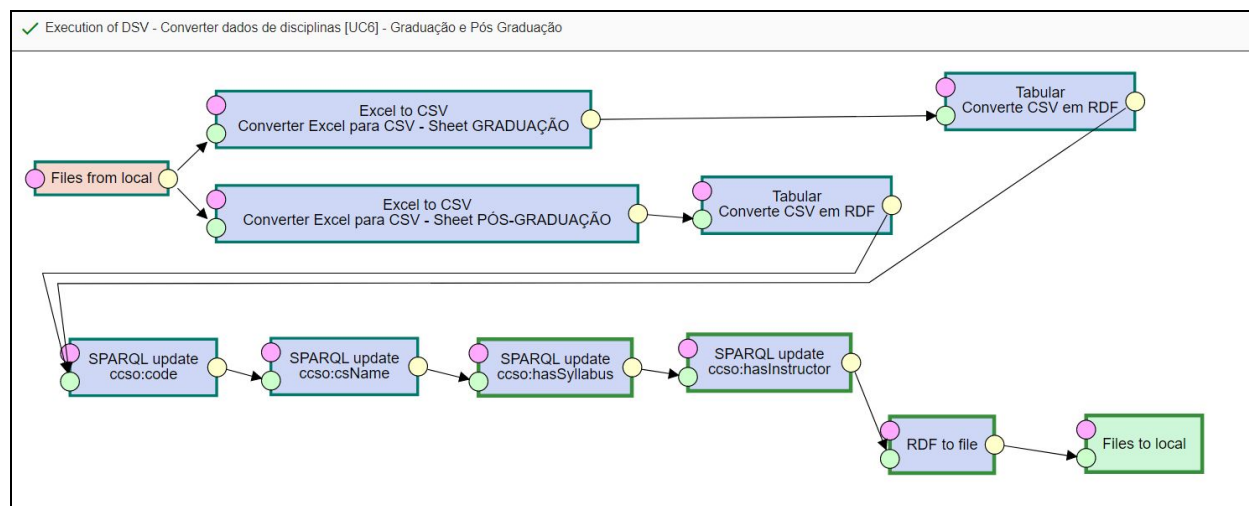


Figura 10 - Pipeline Dados de Disciplinas [UC6] - Graduação e Pós Graduação

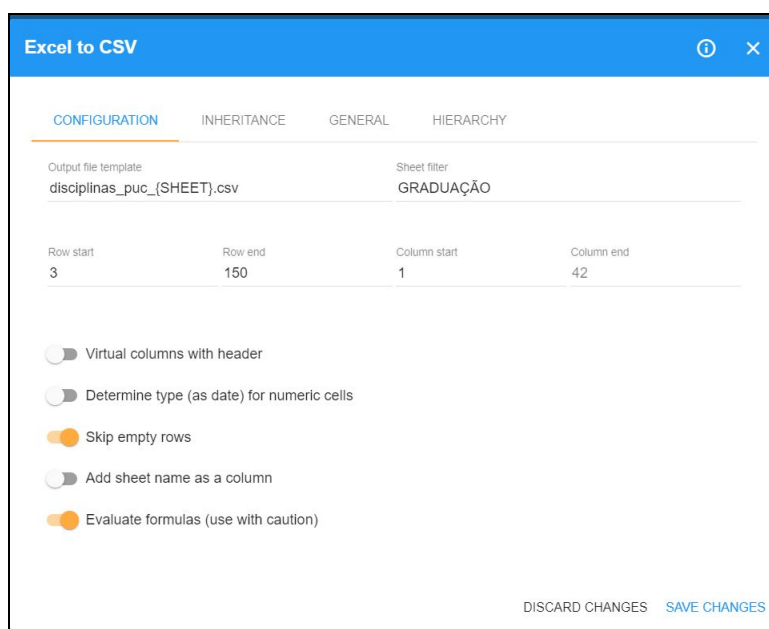


Figura 11 - Configuração do componente “Excel to CSV”

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

4.3. Ferramenta de Busca

O desenvolvimento da ferramenta de busca fez uso do Framework **Flask-APPBuilder** (FAB). As views da hierarquia FAB funcionam como uma classe que representa um conceito e apresenta as visões e métodos para implementá-lo. Cada view é um modelo do FAB que será criado automaticamente. Todas as views são herdadas da classe *BaseView*. Seu construtor é responsável por registrar as URLs. Essa view permite implementar suas próprias páginas personalizadas, anexá-las a um menu ou vinculá-las a partir de qualquer ponto do seu site.

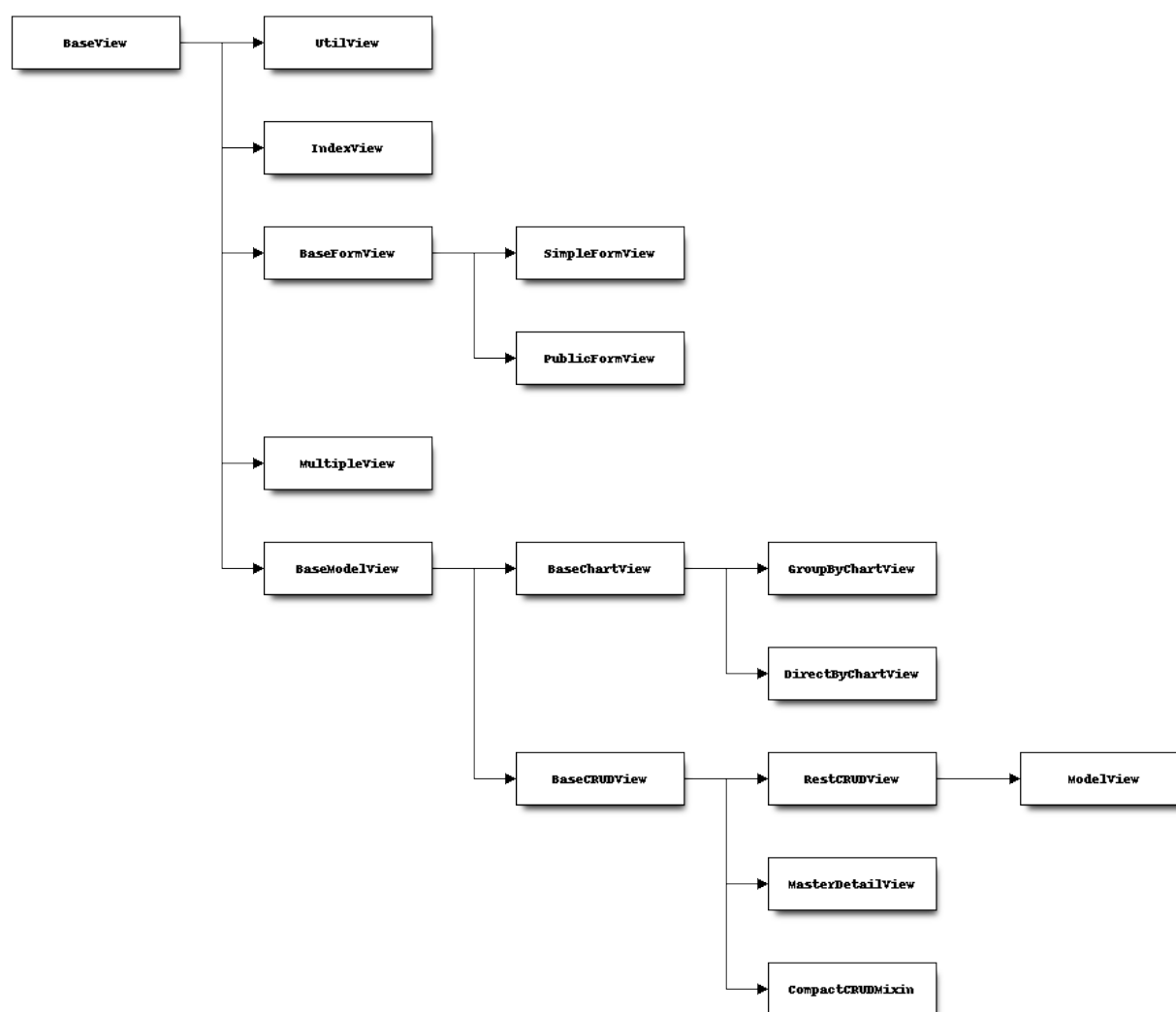


Figura 12 - Hierarquia de classes do FAB

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

A subclasse *BaseFormView* é indicada para construir formulários personalizados que não são baseados no Modelo de Banco de Dados (CRUD) como é o caso do formulário de busca por palavras chaves. Na figura 13 é possível verificar a hierarquia de classes da aplicação que onde a classe *SearchForm* é subclasse de *NimaBaseForm* (*BaseFormView*) que por sua vez é subclasse de *FlaskForm* (*BaseView*).

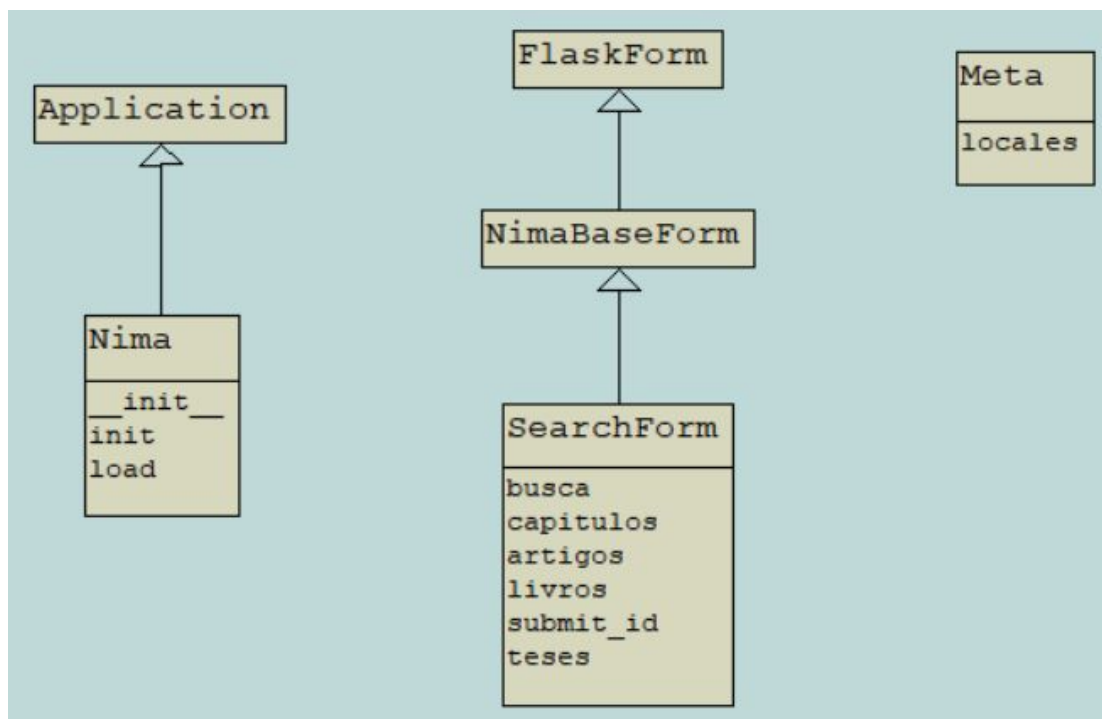


Figura 13 - Diagrama de Classes da aplicação gerado no **pynsource** por engenharia reversa.

O setup inicial da ferramenta de busca foi configurado com a seguinte estrutura de arquivos e pastas da tabela a seguir. Caso seja necessário o refatoramento da aplicação, a modularização da ferramenta de busca poderá ser realizado usando o recurso de **Flask Blueprints**.

.(raiz)	config.py run.py
\app	__init__.py base_form.py connection.py forms.py views.py
\app\static\css	main.css

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

\app\static\images	loading.gif lodoDI.png logo-nima.png puc-rio-logo.png
\app\templates	about.html base.html error.html index.html notfound.html

A interface de usuário é criada a partir dos templates abaixo com o apoio do **Jinja**, um mecanismo de templates para Python que permite usar HTML dentro das aplicações Python. Este recurso tem completo suporte unicode, e para aplicativos que necessitam de maior segurança, adiciona execução em área restrita e escape automático. A seguir a especificação dos templates de Tela que atendem ao [UC1] e o respectivo diagrama de estado do objeto :NIMABaseForm é apresentado na figura 15.

Base (base.html) - aplicado em todos os templates
Cabeçalho Retorno ao Início Rodapé Última atualização dos Lattes aqui utilizados: <data> © 2020 BioBD PUC-Rio
Index (index.html) - formulário de busca - Estado {Index}
Cabeçalho com imagens Entrada [UC1] - passo 1,2 - Estado {Search} Digite as palavras-chave que quer pesquisar: {busca} O que deseja incluir na busca? artigos, livros, teses, capitulos Marcar/Desmarcar todos: Botão Pesquisar Saída [UC1] - passo 5,6 - Estado {Result} Temos {{nResultados}} pesquisadores relacionados com a(s) palavra(s) {{busca}} : Tabela de Resultado: Nome, Tipo, Ocorrências Botões de paginação, ordenação e filtro na tabela de resultados

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

About (about.html) - detalhamento do pesquisador selecionado - Estado {About}
Saída [UC1] - passo 7 Nome do Pesquisador Lista de palavras chaves (string de busca) BIOGRAFIA, EMAIL, HOMEPAGE, ARTIGOS, LIVROS, TESES, CAPITULOS, DOCUMENTOS
Nenhum resultado encontrado (notfound.html) - Estado {NotFound}
Nenhum resultado correspondente foi encontrado em sua pesquisa - {{ busca }}. [UC1] - E2
Erro Genérico (error.html) - captura de qualquer erro não tratado pela aplicação [RNF9] - Estado {Error}
Desculpe mas aconteceu um erro durante a pesquisa. Por favor tente de novo mais tarde.

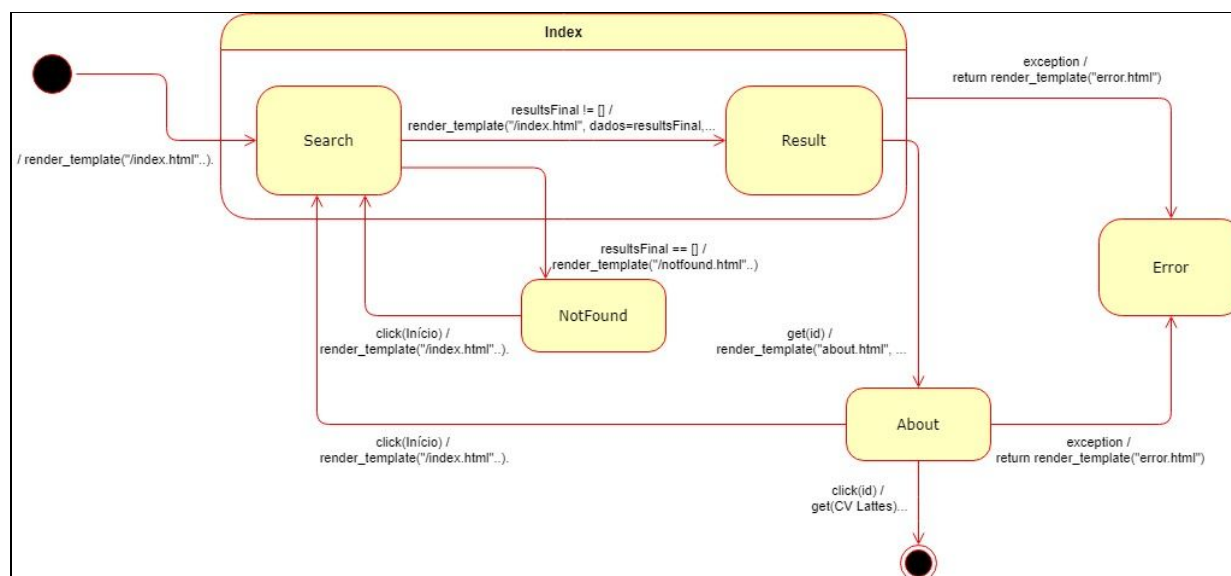


Figura 14 - Diagrama de Estados do :NIMABaseForm

A API AllegroGraph Python permite a comunicação com um servidor AllegroGraph por HTTP ou HTTPS a partir de um aplicativo baseado em Python. Essa API fornece métodos para criar, consultar e manter dados RDF e para gerenciar os triplos armazenados. As versões Python 2.7+ e 3.5+ são suportadas. A instalação da API deve ser realizada através do sistema de gerenciamento de pacotes do Anaconda. A API cria uma cadeia de objetos Python do lado do cliente, terminando em um objeto "conexão". Este objeto contém os métodos que permitem ao aplicativo

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

manipular triplas em um repositório específico. O processo geral é apresentado na figura 15.

Para a configuração de acesso ao servidor de banco de dados AllegroGraph é necessário definir as variáveis de ambiente conforme o quadro 1. De modo a garantir o requisito de segurança [RNF3] foi criado um usuário específico para a aplicação Web, chamado **web_app**, com acesso somente de leitura nos repositórios. Essas configurações assim como a identificação dos repositórios que estão sendo acessados pela aplicação devem ser especificados no arquivo connection.py

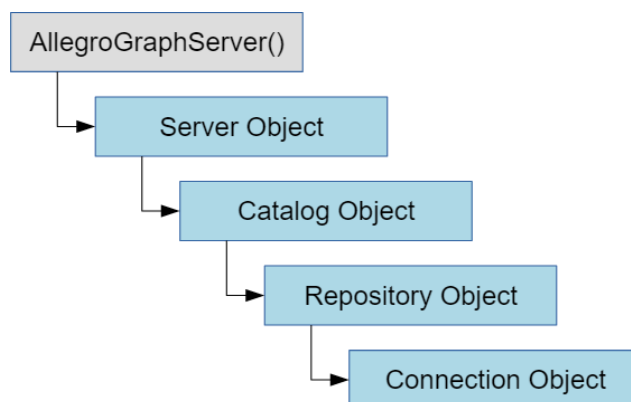


Figura 15 - cadeia de objetos da API Python

```

AGRAPH_HOST = os.environ.get('AGRAPH_HOST')
AGRAPH_PORT = int(os.environ.get('AGRAPH_PORT', '10035'))
AGRAPH_USER = os.environ.get('AGRAPH_USER')
AGRAPH_PASSWORD = os.environ.get('AGRAPH_PASSWORD')
  
```

Quadro 1 - configuração de acesso ao banco

No framework **Flask**, todas as classes para acesso a dados visam abstrair o back-end. A interface *BaseInterface*, estabelece uma camada exclusiva para acesso a dados através de APIs conforme apresentado na figura 16. Esse recurso permite reduzir o acoplamento entre as classes da aplicação e as classes de persistência porém o uso da API não seguiu esse padrão e esse trecho da aplicação poderá sofrer refatoração.

Não foram identificadas interfaces de comunicação com outros sistemas ou dispositivos. Não foram identificadas interfaces de hardware específicas para o sistema.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

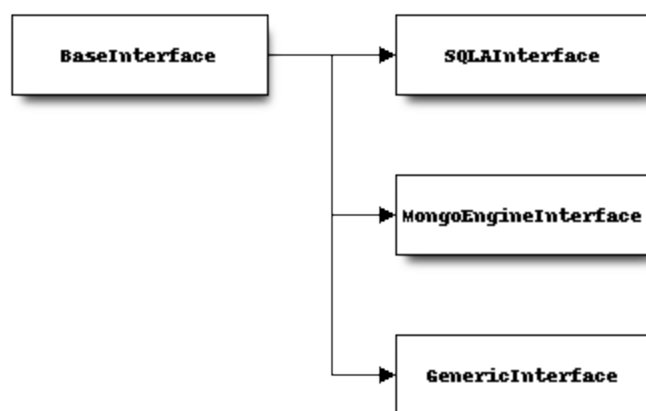


Figura 16 - Hierarquia de interfaces de acesso a dados

5. Visão de Implantação

O diagrama de implantação com os principais componentes do sistema Busc@NIMA é apresentado na figura 17.

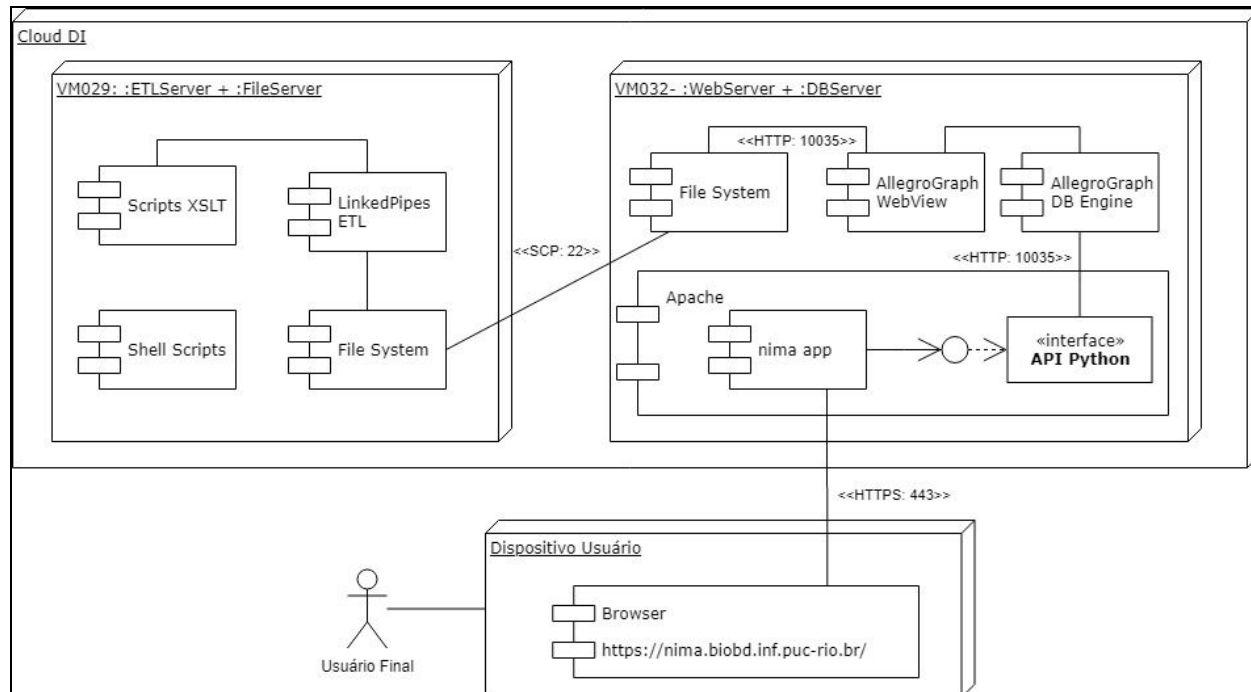


Figura 17 - Visão de Implantação Busc@NIMA

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

6. Controle de Versão

Todos os artefatos associados ao sistema Busc@NIMA deverão ser submetidos a controle de versão através do GitHub. O fluxo para a promoção do código entre ambientes está descrito a seguir e é apresentado na figura 18.

- 1- As branches de Development, Devtest e Release são criadas a partir da branch Master;
- 2- A cada determinado tempo ou término de um determinado número de features (decidido por cada equipe) é feito um "congelamento" do que foi feito para que seja feito um merge para DevTest, onde alguém irá testar se as novas adições estão funcionando como esperado;
- 3- Após testar tudo em DevTest, os bugs devem ser corrigidos em DevTest e então testados novamente. Assim que for garantido que está tudo funcionando como esperado em DevTest, é feito um merge com Development;
- 4- No fim de um sprint, após forem terminados os consertos de bugs de DevTest, é feito um merge para Release, onde o usuário irá homologar;
- 5- Bugs encontrados em Release podem ser corrigidos em DevTest caso não haja nenhum novo desenvolvimento acontecendo ou pode ser criada uma branch Hotfix temporária que fará merge com Release e corrigidos lá;
- 6- Uma vez terminada a homologação em Release é feito um merge para Master e assim pode ser feito um "git pull" em produção;
- 7- Bugs encontrados eventualmente em Master utilizam uma branch Hotfix temporária para a correção, a qual, ao fim do conserto, é desfeita novamente;
- 8- Ao fim de um bugfix na branch Hotfix é feito um merge:
 - *Bug em Release: merge em Release e Development.
 - *Bug em Master: merge em Master, Release e Development.

Busc@NIMA	Version: 1.2
Arquitetura de Software	Date: 07/07/2020
ASW_BuscaNIMA	

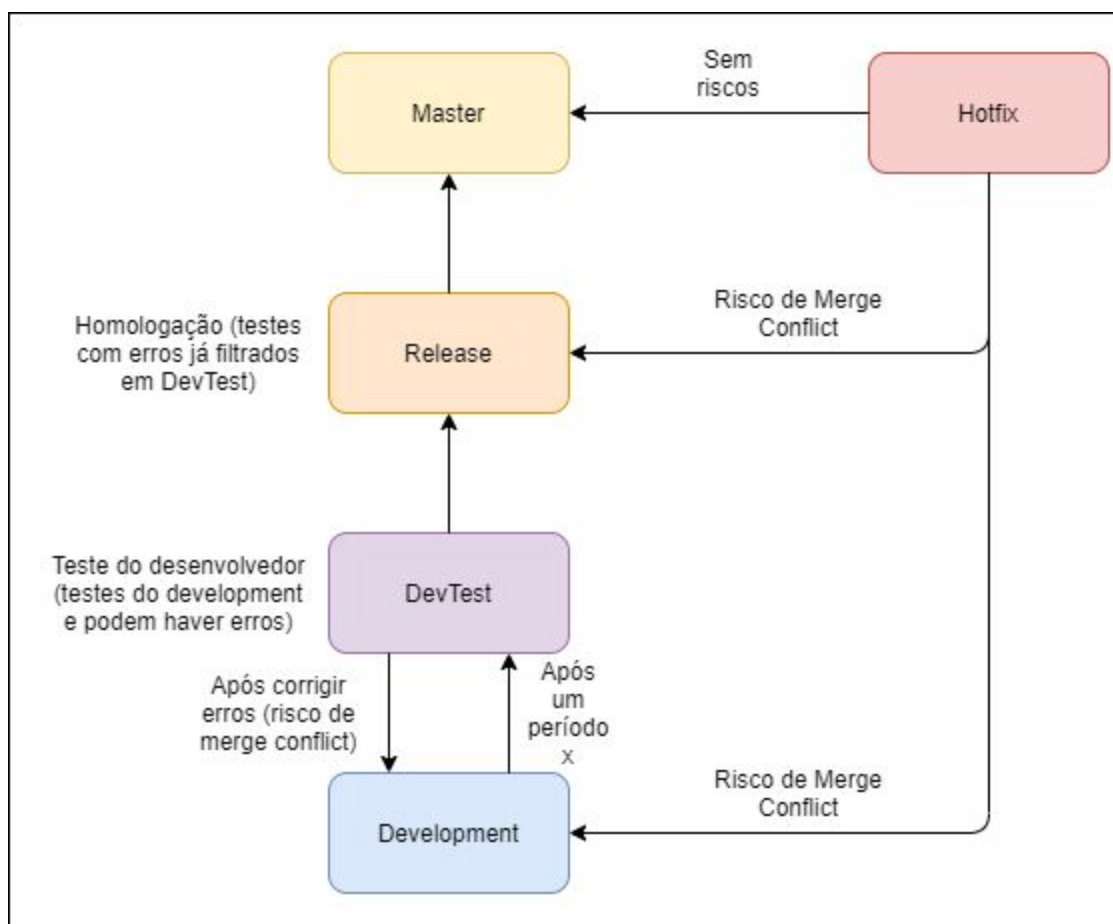


Figura 18 - Fluxo de promoção de código