UNIVERSIDADE DE CAXIAS DO SUL
ÁREA DO CONHECIMENTO DE CIÊNCIAS EXATAS E ENGENHARIAS

VENICIUS BREGALDA

COLETA DOS INDICADORES DA PLATAFORMA DE CIDADES DO CONHECIMENTO

CAXIAS DO SUL

2021

VENICIUS BREGALDA

COLETA DOS INDICADORES DA PLATAFORMA DE CIDADES DO CONHECIMENTO

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Orientador: Prof. Dr. Daniel Luis Notari

CAXIAS DO SUL

2021

VENICIUS BREGALDA

COLETA DOS INDICADORES DA PLATAFORMA DE CIDADES DO CONHECIMENTO

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Bacharel em Ciência da Computação na Área do Conhecimento de Ciências Exatas e Engenharias da Universidade de Caxias do Sul.

Aprovado em 29/06/2021

BANCA EXAMINADORA

Prof. Dr. Daniel Luis Notari
Universidade de Caxias do Sul - UCS

Prof. Dra. Ana Cristina Fachinelli
Universidade de Caxias do Sul - UCS

Prof. Dra. Helena Graziottin Ribeiro
Universidade de Caxias do Sul - UCS

Aos meus pais Luiz Carlos e Marivete.

AGRADECIMENTOS

Agradeço à minha família. Aos meus pais e irmã que sempre foram meu maior apoio e incentivo; à minha namorada Katiane que esteve sempre ao meu lado e aos meus amigos Adriano, Darlen, Bruno, Jéssica, Julio e Leopoldo pelo incentivo e que nunca mediram esforços para me ajudar, sanar minhas dúvidas e me mostrar novos caminhos.

Agradeço ao meu orientador, Profº. Daniel Luis Notari, que com exímio me guiou por essa pesquisa e compartilhou dos seus conhecimentos comigo de forma paciente e compreensiva. Aos professores, coordenadores e demais funcionários da UCS que fizeram possível toda minha jornada pela graduação. Agradeço a todos que me apoiaram e compartilharam, conhecimento e carinho no trajeto para a alcançar este objetivo. Ofereço a vocês toda minha gratidão.

"Inteligência é a capacidade de se adaptar à mudança."

Stephen Hawking

RESUMO

A utilização do Sistema de Capitais Genérico contribui com a identificação, avaliação e desenvolvimento de Cidades do Conhecimento e Cidades Inteligentes. A Plataforma de Cidades do Conhecimento, buscou automatizar a entrada de dados e geração de gráficos e tabelas comparativas dos indicadores de Sistemas de Capitais para o gerenciamento do conhecimento baseado no Sistema de Capitais Genérico. A grande quantidade e heterogeneidade das fontes de dados, faz com que o processo de integração dos dados para a plataforma torne-se ineficiente e manual. Este trabalho tem como objetivo desenvolver um software para carga semi-automática de alguns indicadores do Sistema de Capitais Genérico utilizados na Plataforma de Cidades do Conhecimento. Tal sistema proporcionará ao usuário mapear e vincular indicadores de um Sistema de Capitais Genéricos com dados das fontes de dados que serão integradas executando um processo de extração, transformação e carga (ETL) para o banco oficial da plataforma de Cidades do Conhecimento. Para entender o problema e definir as funcionalidades do sistema, foi executado criação de requisitos e a arquitetura para o desenvolvimento do software. Os resultados obtidos com a aplicação criada, demonstraram que a mesma resolveu os problemas que eram enfrentados na integração dos dados, automatizando processos que eram feitos de forma manual e garantindo maior integridade dos dados de cada indicador.

Palavras-chave: Cidades do conhecimento. Integração de dados. Sistema de Capitais.

ABSTRACT

The use of the Generic Capital System contributes to the identification, assessment, and development of Cities of Knowledge and Smart Cities. The Cities of Knowledge's Platform automated data entry, graphs, and comparative tables of Capital Systems indicators for knowledge management based on the Generic Capital System. The large amount and heterogeneity of data sources make the platform's data integration process inefficient and manual. This work aims to develop one software for semi-automatic loading of some indicators of the Generic Capital System used in the Cities of Knowledge's Platform. The system will allow the user to map and link indicators of a Generic Capital System with data from integrated data sources executing an extraction, transformation, and loading process (ETL) to the official database of the Cities of Knowledge's platform. To understand the problem and define the system's functionalities, requirements were created, and the software development architecture was performed. The results obtained with the created application showed that the problems faced in data integration were solved, automating processes once performed manually and ensuring greater data integrity for each indicator.

Keywords: City Living. Capital System. Data integration.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE ABREVIATURAS E SIGLAS

ANATEL	Agência Nacional de Telecomunicações
DATASUS	Departamento de informática do Sistema Único de Saúdedo Brasil
ETL	Extract Transform Load
IBASE	Instituto Brasileiro de Análises Sociais e Econômicas
PPGA	Programa de Pós-Graduação em Administração
SGBD	Sistema de gerenciamento de banco de dados
API	*Application Programming Interface*
REST	*Representational State Transfer*
SPA	*Single-Page Applications*
HTTP	*Hypertext Transfer Protocol*
ER	Entidade Relacionamento
BCB	Banco Central do Brasil
SIDRA	Sistema IBGE de Recuperação Automática
SNIS	Sistema Nacional de Informações sobre Saneamento

SUMÁRIO

1 INTRODUÇÃO

O conhecimento é a fonte crítica de progresso desde as origens da espécie humana na terra. Nações e organizações internacionais perceberam que os desafios que as sociedades modernas enfrentam exigem estratégias de desenvolvimento baseados no conhecimento (ERGAZAKIS; METAXIOTIS; PSARRAS, 2004). Com o conhecimento, as cidades são idealizadas para encorajar a produção e a circulação de conhecimento em um ambiente humano ecologicamente conservado, economicamente seguro, socialmente justo e bem governado (YIGITCANLAR, 2011; MICHELAM; CORTESE; YIGITCANLAR, 2020).

Com o desenvolvimento baseado no conhecimento surgem as cidades do conhecimento, que possuem características estruturais, estabelecidas visando alcançar sustentabilidade, melhoria da qualidade de vida, fornecer os serviços necessários, enriquecer a cultura e o saber e aumentar o desenvolvimento intelectual humano (YIGITCANLAR; O'CONNOR; WESTERMAN, 2008).

Uma Cidade do Conhecimento é identificada a partir de uma metodologia necessária para democratizar o processo de criar, armazenar, compartilhar e utilizar o conhecimento durante o desenvolvimento urbano, e nesse contexto se encaixa o sistema de capitais (CARRILLO, 2006). A Teoria dos Sistemas de Capitais foi construída por (CARRILLO, 2002) para realizar a transição dos sistemas de valor da produção predominantemente baseada no material, para a produção predominantemente baseada em conhecimento. O artigo (FACHINELLI; CARRILLO; D'ARISBO, 2014) mostra a pesquisa do uso do sistema genérico de capital como ferramenta para identificar uma possível cidade do conhecimento brasileira. Foram utilizados as diferentes categorias de Capitais: Identidade, Inteligência, Relacional, Financeiro, Investimento, Humano Individual, Humano Coletivo, Instrumental-Material e Instrumental-Intangível. Os dados e gráficos foram inseridos e manipulados manualmente através de uma planilha eletrônica, o que poderia gerar inconsistência nas informações.

O artigo de (NOTARI; BATTISTELO; MOLIN, 2019) mostra o objetivo de construir um software para armazenar, calcular e consultar os indicadores de Cidade do Conhecimento. Entre os objetivos descritos no trabalho, houve um que não foi possível ser realizado, a respeito da resolução da entrada dos dados incorreta: a importação dos dados para o banco de dados do software criado, que continua utilizando uma planilha eletrônica. Outro desafio descrito na área dos trabalhos futuros é o de identificar uma maneira de vincular automaticamente os indicadores de uma categoria de capital com a entrada dos dados.

1.1 PROBLEMA DE PESQUISA

O CityLivingLab é um laboratório para a disseminação do conhecimento envolvendo a temática de Cidades Inteligentes e Desenvolvimento Baseado no Conhecimento, composto por professores, pesquisadores e alunos vinculados ao Programa de Pós-Graduação em Administração (PPGA). Atualmente, as informações do laboratório estão disponíveis na web em https://www.citylivinglab.com/. Um dos produtos disponíveis é a Plataforma de Desenvolvimento Baseado em Conhecimento projetada a partir do trabalho de (FACHINELLI; CARRILLO; D'ARISBO, 2014).

A Aplicação de Cidades do Conhecimento tem o intuito de armazenar, calcular e consultar os indicadores de Sistema de Capitais para identificar e estudar o desenvolvimento de cidades como ferramenta para identificar uma possível Cidade do Conhecimento. A ferramenta utiliza indicadores separados em categorias de Capitais, sendo: Identidade, Inteligência, Relacional, Financeiro, Investimento, Humano, Individual, Humano Coletivo, Instrumental-Material e Instrumental-Intangível.

Atualmente, o processo de carga de dados é feito manualmente, a atualização dos dados envolve uso de diferentes fontes, e esses dados são armazenados em uma planilha eletrônica utilizada para realizar a integração com o banco de dados da aplicação, o que resulta em entrada de informações incorretas e uma quantidade desnecessária de esforço manual.

Com essas informações, a questão a ser respondidada na pesquisa é: É possível criar uma processo de integração de dados que atualize os valores de entrada nos corretos indicadores de cada Capital de forma semi-automática?

1.2 OBJETIVO

Desenvolver um software para carga semi-automática de alguns indicadores do Sistema de Capitais utilizados na Plataforma de Cidades do Conhecimento.

1.3 ESTRUTURA DO TEXTO

Este documento começa apresentando a introdução, problema de pesquisa e objetivo para que o leitor entenda o contexto principal. No capítulo 2 são apresentadas informações e conceitos de Cidades do Conhecimento e Sistemas de Capitais, funcionalidades da Aplicação de Cidades do conhecimento e definida a nova carga de dados para a mesma. Para realizar a integração são utilizadas tecnologias e técnicas de integração de dados, por este motivo no capítulo 3 é realizada uma breve discussão sobre o assunto, respectivas ferramentas utilizadas e os trabalhos relacionados. No capítulo 4 é descrita a proposta de solução do problema do trabalho, discutindo o problema a ser resolvido, a abordagem encontrada para a solução na modelagem do software e por fim as considerações finais. O capítulo 5 são apresentadas informações da

implementação da proposta de solução, que detalha a maneira que a aplicação foi desenvolvida, explicando as tecnologias usadas e a estrutura da aplicação. No capítulo 6, estudo de caso, que apresenta o funcionamento da aplicação, particularidades das fontes de dados, fontes de dados utilizadas e os requisitos cumpridos. No capítulo 7 é descrita a conclusão do trabalho, que realiza uma análise buscando identificar se os problemas levantados foram resolvidos ou minimizados.

2 CIDADES INTELIGENTES

O caminho para tornar as cidades inteligentes e sustentáveis, passa também pela assimilação do conhecimento (CARRILLO *et al.*, 2014). Uma cidade inteligente seria aquela que atinge suas metas de sustentabilidade com o apoio de tecnologias modernas (CHANG *et al.*, 2018). A cidade inteligente representa o desafio do futuro, um modelo de cidade onde a tecnologia está em serviço à pessoa e à melhoria da qualidade de vida econômica e social (LAZAROIU; ROSCIA, 2012).

2.1 CIDADES DO CONHECIMENTO

Do final da década de 1990 em diante, estudiosos começaram a levantar a questão crítica de como lidar com os desafios do desenvolvimento que exigem uma base de conhecimento para as cidades e seus cidadãos, que só pode ser alcançado por meio do desenvolvimento baseado no conhecimento (CARRILLO *et al.*, 2014).

O desenvolvimento urbano baseado no conhecimento levou cidades a formular estratégias de desenvolvimento para o progresso baseado no conhecimento, e o resultado consequente deste desenvolvimento é a formação de Cidades do Conhecimento (SARIMIN; YIGITCANLAR, 2012). Cidades do Conhecimento têm um amplo significado envolvendo diversas áreas de estudo aplicados em cidades, regiões, bairros, comportamentos sociais, econômicos e culturais. Uma Cidade do Conhecimento é uma cidade que realiza pesquisas para a criação de valor em todas as suas áreas, desenvolve elevados padrões de vida e apoio cultural, tem alto desenvolvimento econômico, entre outros aspectos, incluindo maior nível de renda, educação, treinamento e pesquisa, e em simultâneo é uma economia regional do conhecimento impulsionada com exportações de alto valor agregado, criadas através de pesquisa, tecnologia, inteligência e propositalmente projetado para encorajar o cultivo do conhecimento (CARRILLO *et al.*, 2014).

Um conceito para definir Cidades do Conhecimento é aquela que visa o desenvolvimento baseado em conhecimento, incentivando continuamente os processos de gerenciamento do conhecimento. Isto pode ser alcançado por meio da contínua interação entre seus agentes de conhecimento e, em simultâneo, entre eles e os agentes de conhecimento de outras cidades. Design apropriado, redes de tecnologia da informação, comunicação e infraestruturas apoiam essas interações (ERGAZAKIS; METAXIOTIS; PSARRAS, 2004).

Carrillo, explana que a possibilidade para o desenvolvimento de uma Cidade do Conhecimento vai além de prover melhor habitação, transporte ou inovação. Para ele uma Cidade do Conhecimento é uma cidade onde a cidadania compromete-se em uma tentativa deliberada e sistemática de identificar e desenvolver o seu sistema de capitais, com uma abordagem equilibrada e sustentável (CARRILLO, 2004).

2.2 SISTEMAS DE CAPITAIS

Um dos estudos na área de cidade do conhecimento é o Sistema de Capitais, um sistema de gestão do conhecimento baseado em categorias com uma classificação completa e consistente. O artigo (FACHINELLI; CARRILLO; D'ARISBO, 2014) mostra que no Brasil já foi realizado um estudo na convergência conceitual entre o Sistema de Capitais Genérico e o modelo brasileiro de economia para o desenvolvimento de uma emergente cidade do conhecimento, o primeiro caso de utilização do Sistema de Capitais Genérico foi na cidade de Bento Gonçalves.

A aplicação do Sistema de Capitais mostrou ser um método de operacionalização que permite o desdobramento de medidas específicas para o desenvolvimento baseado no conhecimento e efetivamente ajuda a entender onde o valor e potenciais existentes em vários setores de uma cidade estão (FACHINELLI; CARRILLO; D'ARISBO, 2014). Segundo (CARRILLO, 2002), a abordagem dos Sistemas de Capital visa representar todas as dimensões de valor para a tomada de decisão coletiva. O sistema é subdividido em cinco metas capitais.

a) referencial: elementos que permitem a identificação e alinhamento de todos outros elementos de valor. É composto de um Capital de Identidade e um Capital de Inteligência;

b) articulação: elementos que permitem a interconexão ou troca entre elementos de valor. É composto de um Capital Relacional e um Capital Financeiro;

c) entrada: elemento de valor de outro sistema que vem como atributo de entrada. É composto de um Capital Investimento;

d) produção: capacidade de geração de valor de cada indivíduo, e os meios de produção que qualquer outro capital utiliza para alavancar sua própria capacidade de geração de valor. É composto de um Capital Agente subdividido em um ítem Humano Individual e Humano Coletivo, e um Capital Instrumental;

e) saída: os valores gerados por todos os outros elementos de valor que ainda não foram enquadrados em outra forma de capital. É composto de um Capital Produto.

É possível capturar de forma completa e consistente o sistema de capital de qualquer entidade. Isso se aplica a todos os indivíduos, organizações e sociedades (CARRILLO, 2002).

2.3 PLATAFORMA CIDADES DO CONHECIMENTO

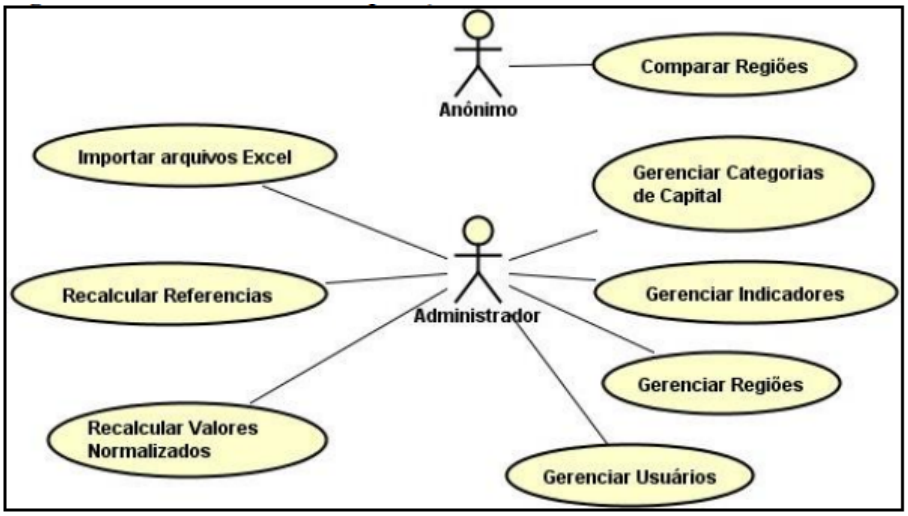
Living labs são redes que auxiliam na criação de inovações que têm uma identificação superior com as necessidades do usuário e pode ser prontamente escalada globalmente (LEMINEN; WESTERLUND; NYSTRöM, 2012). *O City Living Lab*[1] é um laboratório vivo para inovação com dimensões sociais e tecnológicas simultaneamente em uma parceria entre negócios-cidadãos-governo-academia. Trata-se de um ecossistema em que diferentes parceiros trabalham

[1] https://www.citylivinglab.com/

lado a lado, compartilhando conhecimento enquanto interagem com uma ampla variedade de conhecimentos e tecnologias, induzindo assim um terreno fértil para inovação, pesquisa e comunicação interdisciplinar.

A plataforma dentro do *City Liging Lab* tem as funções de geração de gráficos com os indicadores de capitais e a comparação de Sistemas de Capitais. A origem da plataforma se deu pelo fato de os dados e gráficos serem manipulados e gerados através de uma planilha eletrônica no caso de estudo aplicado em Bento Gonçalves (FACHINELLI; CARRILLO; D'ARISBO, 2014), o que resulta em entrada de dados incorretos, redundância de dados e informação dispersa em diferentes repositórios. A solução encontrada foi desenvolver uma aplicação web cuja modelagem do sistema é composta por duas categorias de usuários, anônimos e administradores (NOTARI; BATTISTELO; MOLIN, 2019), e as funcionalidades de cada categoria de usuário estão apresentadas na Figura 1.

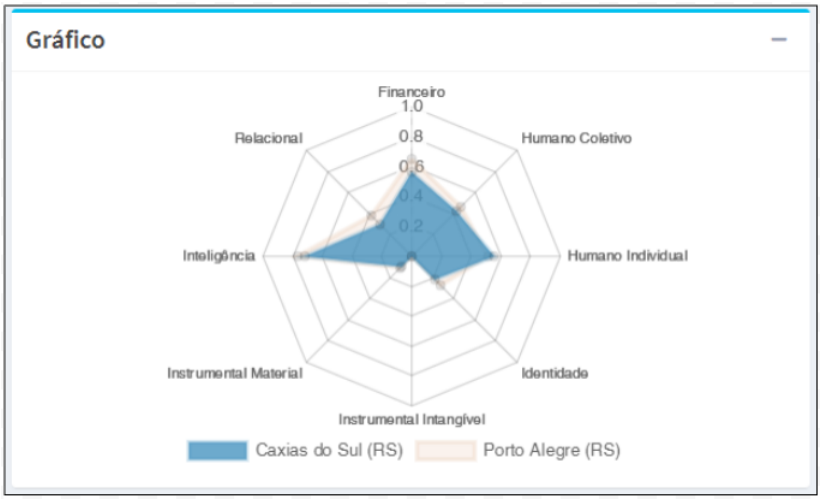
Figura 1 – Funcionalidade da aplicação



Fonte: (NOTARI; BATTISTELO; MOLIN, 2019).

A principal funcionalidade da aplicação é comparar Sistemas de Capitais e o usuário anônimo terá disponível em sua tela inicial uma lista com todas as regiões cadastradas no sistema, tendo a possibilidade de filtrar regiões por nome ou ano e adicionar uma ou mais regiões na comparação atual. Ao realizar a comparação a tela disponibiliza um gráfico comparativo entre as áreas selecionadas para cada Sistema de Capital (NOTARI; BATTISTELO; MOLIN, 2019) conforme Figura 2.

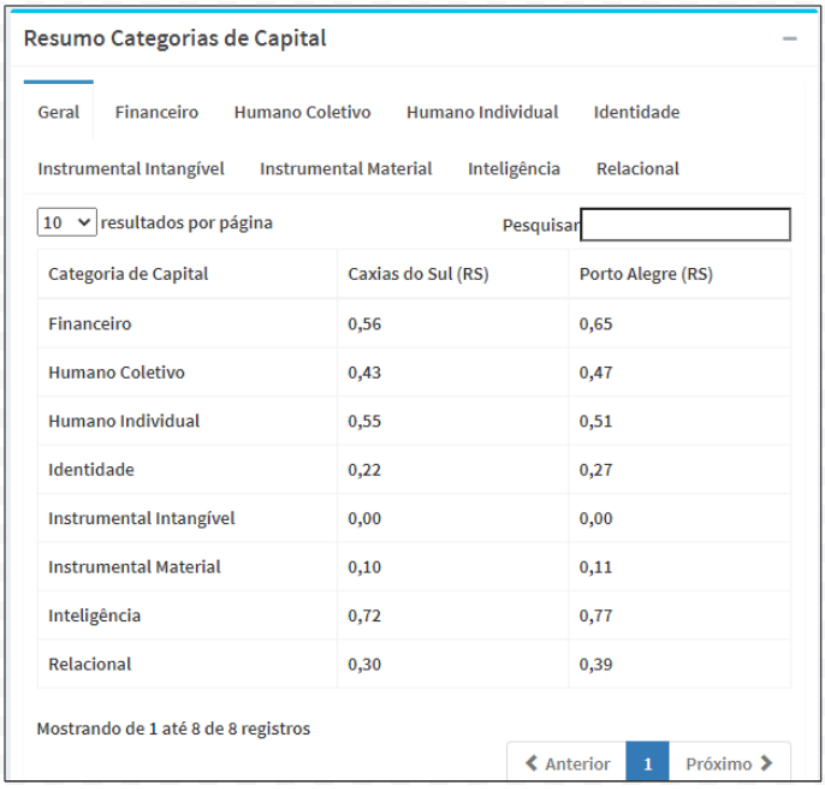
Figura 2 – Gráfico comparativo



Fonte: (NOTARI; BATTISTELO; MOLIN, 2019).

A tela disponibiliza também uma tabela geral com os valores utilizados na montagem do gráfico, e mais uma para cada categoria de capital cadastrada no sistema com os indicadores utilizados e seus respectivos valores (NOTARI; BATTISTELO; MOLIN, 2019). A Figura 3 mostra o resumo geral das categorias de capital realizando a comparação dentre as regiões selecionadas.

Figura 3 – Tabela Geral



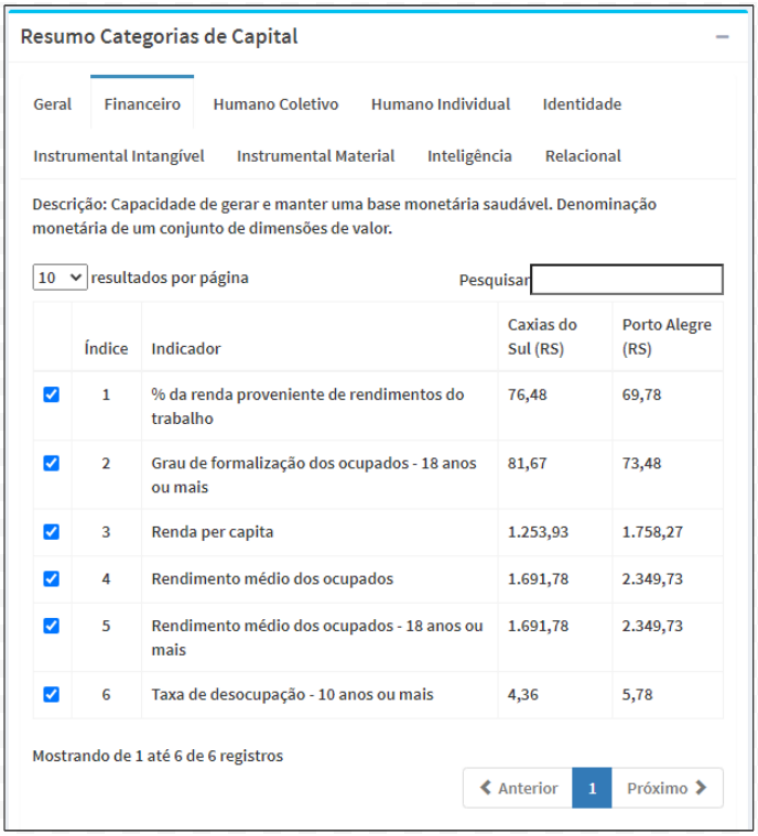
Categoria de Capital	Caxias do Sul (RS)	Porto Alegre (RS)
Financeiro	0,56	0,65
Humano Coletivo	0,43	0,47
Humano Individual	0,55	0,51
Identidade	0,22	0,27
Instrumental Intangível	0,00	0,00
Instrumental Material	0,10	0,11
Inteligência	0,72	0,77
Relacional	0,30	0,39

Fonte: (NOTARI; BATTISTELO; MOLIN, 2019).

A Figura 4 mostra a Categoria de Capital Financeiro selecionada e a lista de seus in-

dicadores, percentual da renda proveniente de rendimentos do trabalho, grau de formalização dos ocupados - 18 anos ou mais, renda per capita, rendimento médio dos ocupados, rendimento médio dos ocupados - 18 anos ou mais e taxa de desocupação - 10 anos ou mais. Para cada Categoria de Capital selecionada mostra os seus indicadores e a comparação das regiões selecionadas.

Figura 4 – Tabela capital financeiro e indicadores



Fonte: (NOTARI; BATTISTELO; MOLIN, 2019).

2.3.1 Nova carga de dados

A aplicação Cidades do Conhecimento mostra atualmente 67 indicadores relacionados ao Sistema de Capitais, conforme quantidade totalizada na Figura 5 e na lista detalhada no anexo A. Esta lista mostra todos os indicadores que os responsáveis da Aplicação de Cidades do Conhecimento definiram. Para realizar a integração para o banco de dados relacional usando o Sistema de gerenciamento de banco de dados (SGBD) *MySQL*[2], são agrupados todos os dados em uma única planilha, cada um desses indicadores tem a origem dos dados de diferentes fontes. Os responsáveis pela Aplicação realizaram um levantamento da origem das fontes de dados, ano, qual é a quantidade de municípios e quais os cálculos necessários para cada um desses indicadores.

[2] https://www.mysql.com/

Figura 5 – Quantidade de indicadores por sistema de capital

Capital	Quantidade Indicadores
Identidade	8
Inteligência	7
Relacional	8
Financeiro	7
Humano coletivo	9
humano individual	10
Instrumental tangível	9
Instrumental intangível	9
TOTAL	67

Fonte: Própria.

A nova carga de dados proposta neste trabalho não é feita em um agrupamento de dados para uma planilha eletrônica, cada fonte de dados com os dados brutos pode ser disponibilizada em um repositório. Após o cadastro das fontes de dados e vínculo de campos com indicadores, é realizada a integração.

3 INTEGRAÇÃO DE DADOS

A integração de dados consiste em combinar dados de diferentes fontes para obter informações valiosas. A integração é importante para permitir que usuários tenham uma visão unificada de dados heterogêneos e consultem facilmente diferentes informações sobre os mesmos assuntos (BOUZEGHOUB *et al.*, 2002).

Integração de dados e Extract Transform Load (ETL) são conceitos distintos. ETL é uma das formas de realizar integração de dados, na qual os dados são buscados em fontes de dados de origem, processados e geram um novo conjunto de dados, armazenado de forma separada dos dados de origem, enquanto integração de dados é uma arquitetura. Assim, a ETL pode ser utilizada como parte da etapa de integração de dados, mas não necessariamente representar toda a integração. Além da ETL, a tarefa de integração de dados também inclui modelagem de dados, criação de perfil dos dados, integração em tempo real, governança de dados, entre outras (DOAN *et al.*, 2017).

3.1 ETL

O processo de ETL possui alguns objetivos como remover erros e corrigir dados ausentes, fornecer medidas documentadas de confiança para os dados, capturar o fluxo de dados transacionais para proteção, ajustar os dados de múltiplas fontes a serem usadas juntas e estruturar os dados a serem utilizados pelas ferramentas do usuário final(KIMBALL; CASERTA, 2004).

ETL é um processo que realiza a extração de dados das mais diversas fontes geradas por sistemas computacionais, bem como realizar transformações e modificações nos dados coletados e realizar atividades de carga dos dados transformados para as bases de dados de destino (YULIANTO, 2019). O processo ETL é composto de três etapas: Extração, Transformação e Carga de dados.

3.1.1 Extração de dados

Consiste em fazer a extração dos dados que se deseja carregar. Cada dado a ser extraído pode estar em diferentes sistemas (diferentes fontes de dados), as informações podem ser encontradas nas mais variadas fontes, como, em bancos de dados relacionais, em arquivos-textos, arquivos binários, planilhas eletrônicas, banco de dados orientado a objeto, etc.

O grau de dificuldade depende diretamente de como será o cenário a enfrentar nos sistemas de origem, que podem estar armazenados em esquemas comuns (homogêneos) ou em estruturas diferentes (heterogêneas); em um banco de dados comum ou com os dados espalhados por diferentes bancos (NETO, 2012).

3.1.2 Transformação dos dados

Esta etapa é a fase subsequente à extração dos dados e é nesta etapa que realizamos os devidos ajustes, podendo assim melhorar a qualidade dos dados. Nesta fase são criadas as regras de conversão conforme os padrões estabelecidos e também é realizada a transformação e a limpeza dos dados. Correção de erros de digitação, substituição de caracteres desconhecidos são alguns exemplos desta limpeza. Esta é a etapa em que é necessário realizar o maior esforço no processo de ETL, embora o grau de dificuldade dependa diretamente de fatores determinados pela qualidade dos dados de origem, das regras de negócios a serem aplicadas e da disponibilidade ou não de documentação atualizada das bases de dados transacionais (NETO, 2012).

A transformação é onde são definidas as regras e funções que definem a qualidade dos dados. Segundo (KIMBALL; CASERTA, 2004), as características mais relevantes para garantir a qualidade de dados são estão listadas em quatro pontos:

a) precisão: os dados não podem perder suas características originais, os valores e descrições descrevem seus objetivos associados de forma sincera e fiel;

b) unicidade: os valores e as descrições nos dados podem ser tomados para ter apenas um significado;

c) completude: não gerando dados parciais de todo o conjunto relevante às análises;

d) consistência: os valores e as descrições nos dados usam uma convenção de notação constante para transmitir o seu significado, ou seja, os fatos devem apresentar consistência com as dimensões que o compõem.

3.1.3 Carga dos dados

A última etapa do processo ETL é a de carga de dados para o novo ambiente. Consiste, em carregar os dados obtidos nas fases anteriores em um arquivo ou banco de dados. Os dados já padronizados são organizados conforme a modelagem da base de dados destino, de modo a garantir a integridade dos mesmos para o carregamento. Nessa etapa são verificadas as disposições dos dados nas tabelas, seus atributos, as referências de chaves estrangeiras, com o propósito de evitar repetição ou perda de dados.

Antes de realizar o processo de carga, é necessário levantar algumas considerações, como, saber com que frequência o processo de ETL será executado, e se será sempre incremental ou total. É importante saber a frequência porque o carregamento de informações pode ter um impacto negativo no desempenho do sistema de dados destino (SILVA, 2016). Após a conclusão de um processo de carregamento dos dados para seu destino, o fluxo do processo de ETL é dado como encerrado, e os dados já estão disponíveis para serem consultados ou carregados.

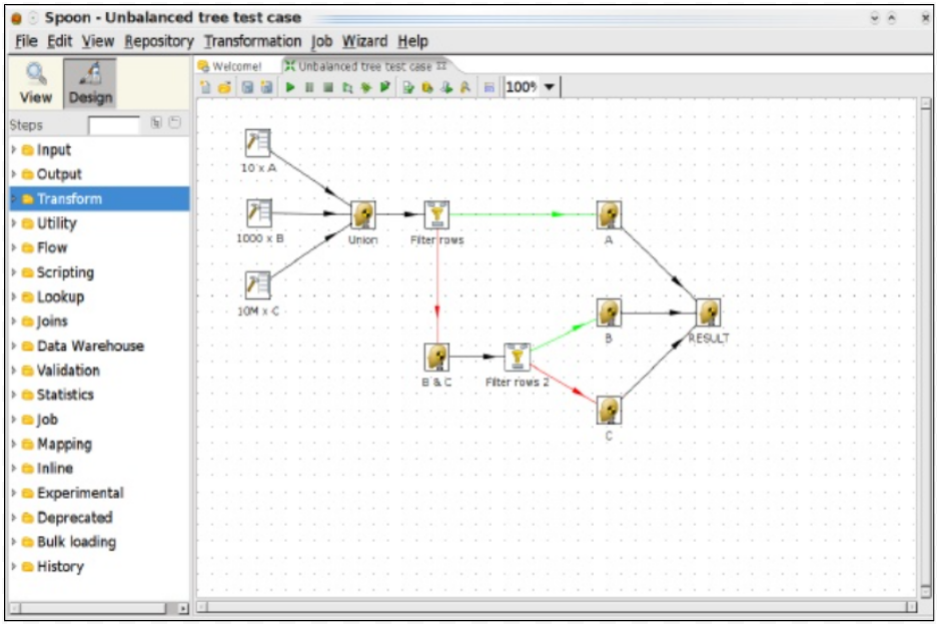
3.2 FERRAMENTAS

Nesta seção, descreve-se o que são e o que fazem algumas ferramentas de ETL, suas características e benefícios, como solução de extração, transformação e carga de dados para um banco de dados no processo de integração.

3.2.1 Pentaho Data Integration

A ferramenta Pentaho Data Integration[1] (PDI) ou simplesmente Kettle é uma ferramenta ETL para a integração de dados (Figura 6). Ela traz um vasto recurso para suporte a extração, transformação e carregamento de dados. Existem dois conceitos importantes no Kettle: Transformação e Job. O primeiro é uma rotina que contém uma coleção de passos para a realização de uma tarefa: ler um arquivo CSV, remover uma coluna e exportar os dados em formato XML. Já o segundo, é uma coleção de rotinas, em que é possível executar várias transformações e até mesmo outros Jobs (PENTAHO, 2021).

Figura 6 – Modelo PDI / Kettle



Fonte: (PENTAHO, 2021).

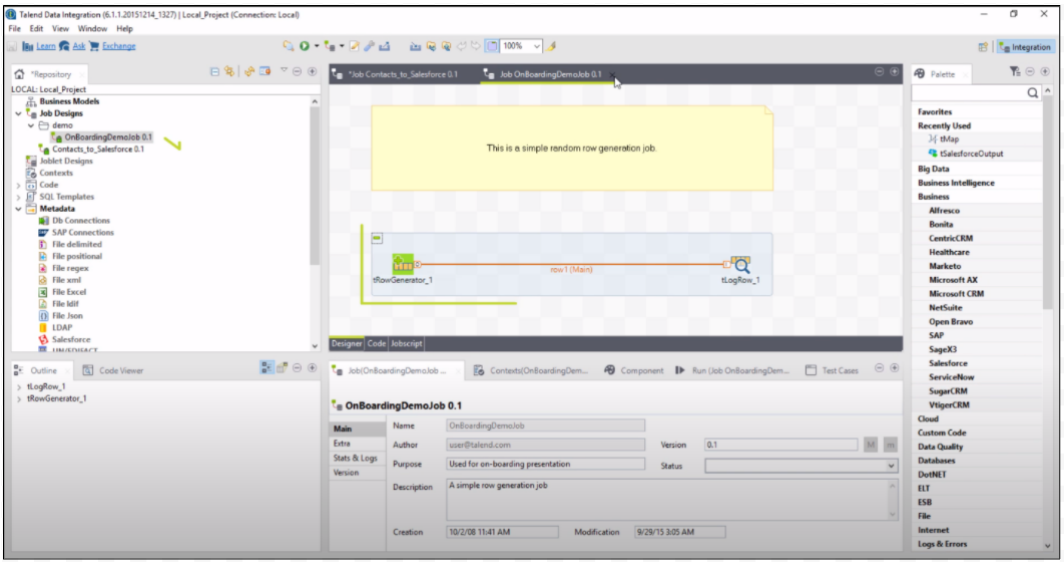
Uma característica do Kettle é que Transformações e Jobs são salvos em arquivos XML. Isso significa que para executar os procedimentos contidos nesses arquivos é preciso um interpretador. Tal interpretador já vem com a ferramenta, porém, para executar os procedimentos XML em uma plataforma externa ao Pentaho é preciso ter um ambiente previamente configurado.

[1] http://www.pentaho.com/product/data-integration

3.2.2 Talend Open Studio for Data Integration

O Talend Open Studio for Data Integration[2] (TOSDI) é uma ferramenta ETL de código aberto para integração de dados. Criado pela empresa Talend, o TOSDI contém componentes e conectores, que permitem fazer extrações, transformações e cargas, desde acesso à simples base de dados até extração e carregamento em complexos Web services (TALEND, 2021).

Figura 7 – Talend Open Studio



Fonte: (TALEND, 2021).

Outro ponto interessante é que o ambiente gráfico Figura 7 da ferramenta é baseado na interface de desenvolvimento Eclipse, mas voltado para a integração de dados. Talend não requer um servidor de execução de processos ETL. A vantagem da geração de código é que é mais simples integrar os processos ETL dentro de outros aplicativos e os modelos são de portabilidade mais flexível.

3.2.3 Python e Pandas

Criada em 1991 pelo holandês Guido van Rossum a linguagem de programação Python[3] é de alto nível, de fácil leitura e está entre as mais populares no mundo (REDMONK, 2021), além de permitir diversas integrações e disponibilizar acesso às mais diferentes bibliotecas, permitindo versatilidade no desenvolvimento de aplicações que possibilitam sua execução nos mais diversos campos da computação, dentre eles o campo de Ciência de Dados.

Pandas[4] é uma biblioteca licenciada com código aberto que oferece estruturas de dados de alto desempenho e fácil utilização. No que tange as necessárias verificações e análises estatísticas, como cálculos de desvio padrão, soma, média, quartil, valores mínimos e máximos, entre

[2] http://www.talend.com/
[3] https://www.python.org/
[4] https://pandas.pydata.org/

outros, tais análises são facilmente realizadas através de funções simples existentes na biblioteca. Além de possibilitar a junção de dois ou mais conjuntos de dados, realizar agrupamentos entre diferentes categorias de dados, realizar a conversão e criação de séries temporais.

3.2.4 Considerações sobre as ferramentas

Analisando as ferramentas de ETL, Pentaho e Talend: são ferramentas gratuitas, possuem uma interface gráfica que possibilita arrastar os componentes necessários para criar o fluxo de integração e necessita conhecimento específico. O Python é uma linguagem versátil, suporta múltiplos paradigmas de programação e estruturas de dados complexas.

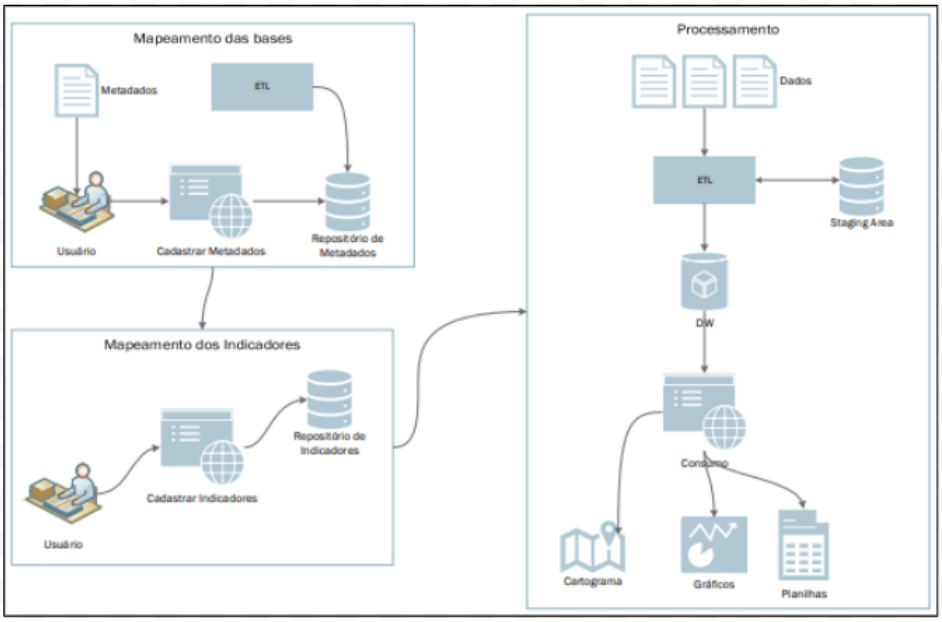
A automatização de ETL se faz necessária tendo em vista que se tornam repetitivos os procedimentos menores que compõe esse processo de integração de dados. Por ter uma comunidade bastante ativa, ser uma linguagem versátil de fácil aprendizado, utilização e um conhecimento prévio, optou-se pela utilização do Python e a biblioteca Pandas para realizar o processo de integração.

3.3 TRABALHOS RELACIONADOS

Realizado a pesquisa de trabalhos científicos cujos títulos ou resumos fossem relacionados com as palavras chaves, *data integration*, *data extraction*, *data import*, *smart city* e *smart cities*.

CARVANO (2018) descreve a solução de Data Warehousing que possibilita ao Instituto Brasileiro de Análises Sociais e Econômicas (IBASE) ter autonomia no processo de produção de indicadores sociais a partir de dados públicos abertos. Um dos objetivos específicos foi desenvolver uma solução de ETL que possibilite aos utilizadores da Instituição a criação dos indicadores sociais. Optou-se por desenvolver uma solução de ETL baseada na linguagem de programação Python, os arquivos de dados são disponibilizados em formato CSV. O fluxo de carga da solução é composto por três macro etapas: Mapeamento das bases de Dados, Mapeamento dos Indicadores e Processamento dos Indicadores visualizado na (Figura 8). Na etapa de mapeamento das bases foi feito um cadastro do caminho de cada arquivo para a importação, na etapa de mapeamento de indicadores foi feito o cadastro para cada indicador realizando o de para com as bases de dados e por fim a etapa de processamento em que são processados os dados e armazenados nas estruturas dimensões e fatos no *data warehouse*. Os pontos positivos foram a utilização da linguagem Python no processamento de dados e o mapeamento dos metadados para permitir a leitura e extração de dados. Como limitações não se descarta a necessidade de utilizadores capacitados e especialistas durante o processo de produção dos indicadores.

Figura 8 – Fluxo de integração



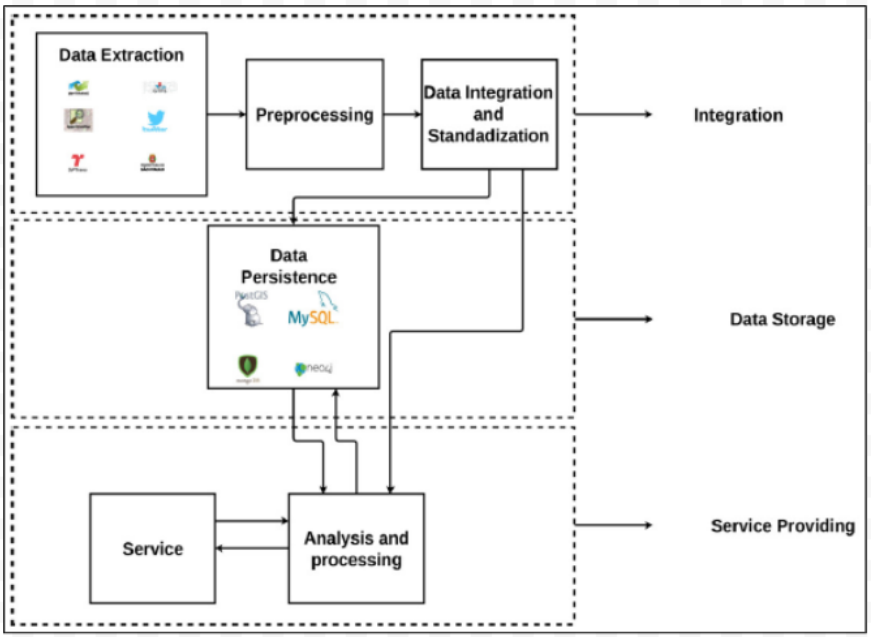
Fonte: (CARVANO, 2018).

ALMEIDA *et al.* (2020) propõe o objetivo de prover uma visão geral atualizada da literatura acerca de soluções para extração, integração e importação de dados heterogêneos em cidades inteligentes. No mapeamento sistemático realizado, 28 estudos foram selecionados e analisados. O estudo mostra os principais desafios de pesquisa e desenvolvimento separado em quatro itens: relacionamento entre dados, heterogeneidade dos dados, inconsistências dos dados e frequência de atualização dos dados. Foi observado pelo autor que a maior parte dos estudos recorre a estratégias de consolidação de dados e vocabulário semântico com vistas a simplicidade, eficiência e padronização dos dados. Existe uma tendência na utilização de modelos semânticos em combinação com a utilização de RDF, visando uma melhor padronização e integração de informações de dados heterogêneos.

SAMU (2018) apresenta uma ferramenta para disponibilizar os dados censitários do censo demográfico de 2010 realizado pelo IBGE de forma simples e eficaz. A implementação foi feita fragmentando em alguns módulos de extração e conversão transformação e consulta, adotando como modelo base o conceito de ETL utilizando a linguagem Python. Dificuldades foram encontradas na automatização da extração de conteúdo de um arquivo ZIP e a divisão dos arquivos do Estado de São Paulo entre a capital e os demais municípios, comportamento não presentes nas demais Unidades Federativas sendo necessário controlar as situações. O autor afirma que a escolha da linguagem Python mostrou-se bastante útil pelo fato desta linguagem apresentar um alto grau de flexibilidade e conter em seu escopo diversas bibliotecas de manipulação de dados. Outra ferramenta utilizada foi a biblioteca Pandas que possibilitou a análise e validação das etapas de desenvolvimento.

FORTINI; DAVIS (2018) propõe o estudo de técnicas e métodos para a análise, integração e visualização de dados urbanos de múltiplas fontes heterogêneas, permitindo a criação de ferramentas para análise de dados urbanos, com foco em transporte. O objetivo principal era permitir a comparação de acessos e mobilidade em uma cidade usando dois modos de transporte diferentes: público (mais especificamente ônibus) e transporte privado (individual). O framework é composto por três camadas, exibidas na Figura 9: Integração, Armazenamento de Dados e Serviços. Na camada Integração é realizada a extração de dados de fontes online, processamento com algoritmos e ferramentas auxiliares ajudando na limpeza e filtragem dos dados recentemente extraídos de fontes online e realizada a padronização dos dados, de modo a seguir os padrões de nomenclatura e tipagem esperado pelos algoritmos implementados. Na etapa Armazenamento de Dados são armazenados dados heterogêneos organizados de uma maneira híbrida com um índice espacial, tabelas hash, listas classificadas e uma lista adjacente. Na camada de Serviços, o resultado é entregue ao usuário do aplicativo ou sistema, que pode ser resultado de um modelo de previsão implementado por uma mineração de dados ou algoritmo de aprendizado de máquina, a materialização de uma visualização técnica, ou ambos.

Figura 9 – Fluxo de integração



Fonte: (FORTINI; DAVIS, 2018).

4 PROPOSTA DE SOLUÇÃO

Nesta etapa será desenvolvida a proposta e solução dos problemas descritos no objetivo deste trabalho. Para resolver os desafios na integração atual, será desenvolvida uma arquitetura de integração, construindo um software para realizar o mapeamento das bases de dados de entrada e o processo ETL para realizar a carga dos dados no banco de dados da plataforma de Cidades do Conhecimento.

4.1 DESCRIÇÃO DO PROBLEMA

Como visto anteriormente no capítulo 2.3 foi apresentada a ferramenta de Cidades do Conhecimento que realiza a integração dos dados de uma planilha eletrônica para o banco de dados da aplicação e geração de gráficos em um *dashboard* (NOTARI; BATTISTELO; MOLIN, 2019).

Segundo NOTARI; BATTISTELO; MOLIN (2019) não foi possível resolver o problema de entrada de dados incorreta, isto porque hoje não existe uma regra que permita identificar a qual categoria de capital um indicador pertence, fazendo com que esta análise, e consequentemente a adaptação da planilha eletrônica Microsoft Excel de importação, continuem sendo efetuadas manualmente. O usuário realiza uma análise nas diversas fontes de dados buscando os indicadores e copia as informações manualmente, agrupando na planilha eletrônica para posteriormente realizar a integração. Toda vez que necessário realizar uma atualização, o usuário precisa repetir o processo manual de encontrar os indicadores, realizar os cálculos (se existir), limpar dados indevidos, etc. A não existência de uma maneira automática de identificar os indicadores de categoria de capital com a entrada de dados, impossibilita a realização da integração de dados sem uma planilha eletrônica criada manualmente.

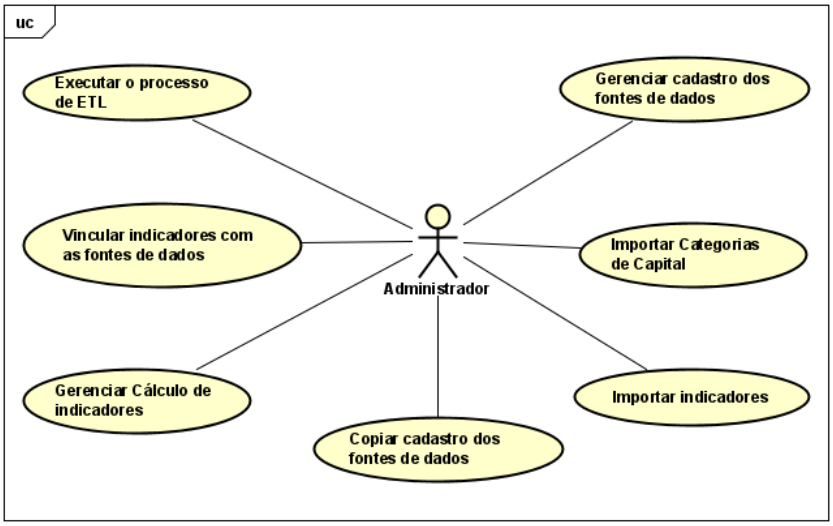
4.2 MODELAGEM DO SOFTWARE

A existência de uma grande variabilidade das estruturas dos dados torna o processo de mapeamento das fontes de dados fundamental para o desenvolvimento da aplicação na primeira etapa. Sem este, as etapas seguintes da solução não têm como ser realizadas. O problema que se coloca nessa etapa de mapeamento é como tratar eficientemente uma variedade tão grande de fontes de dados com diversas estruturas. Na segunda etapa serão usadas as informações cadastradas para realizar a integração dos dados no processo de ETL para que os dados brutos sejam transformados em informações.

4.2.1 Requisitos

Após conversas com pessoas responsáveis pela ferramenta de cidades do conhecimento, foi realizada a análise dos problemas descritos, criado os requisitos funcionais do sistema e os protótipos de tela do software. O sistema é utilizado somente por um usuário, administrador. As funcionalidades estão listadas na Figura 10 no diagrama de casos de uso.

Figura 10 – Funcionalidades da aplicação



Fonte: Própria

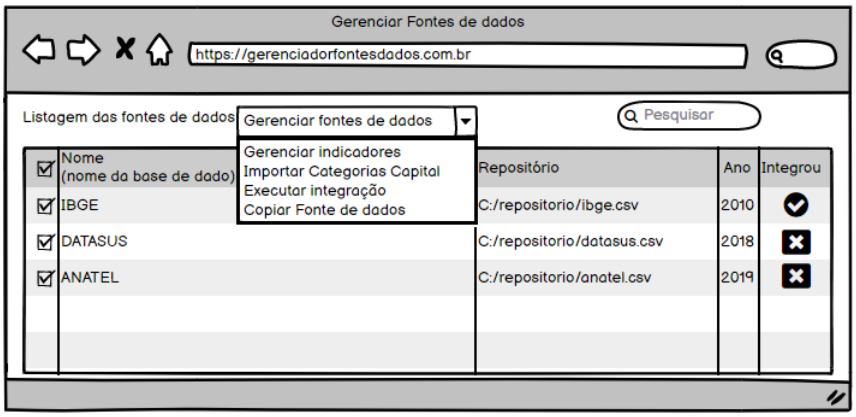
O requisito 1 Importar Categorias de Capital define que o usuário pode visualizar e importar as categorias de capitais. A regra de negócio associada define ser possível importar um arquivo com dados dos Sistemas de Capitais da ferramenta de Cidades do Conhecimento.

O requisito 2 Importar indicadores define que o usuário pode visualizar, e importar os indicadores. A regra de negócio associada define ser obrigatório ter os Sistemas de Capitais importados e importar um arquivo com os indicadores da ferramenta de Cidades do Conhecimento.

O requisito 3 Copiar cadastro das fontes de dados define que o usuário pode copiar uma estrutura de cadastro de fonte de dados. A função desta funcionalidade é evitar um cadastro completo novamente para uma mesma fonte de dados, mas de um período diferente.

A tela inicial Figura 11 do software terá o acesso a todos os requisitos funcionais que permite gerenciar fontes de dados, gerenciar indicadores, importar categorias do sistema de capital, executar a integração e copiar uma estrutura de fonte de dados já cadastrada. A tela exibe uma listagem com informações que permitem identificar a origem da base de dado, ano, e se os dados foram integrados ou não.

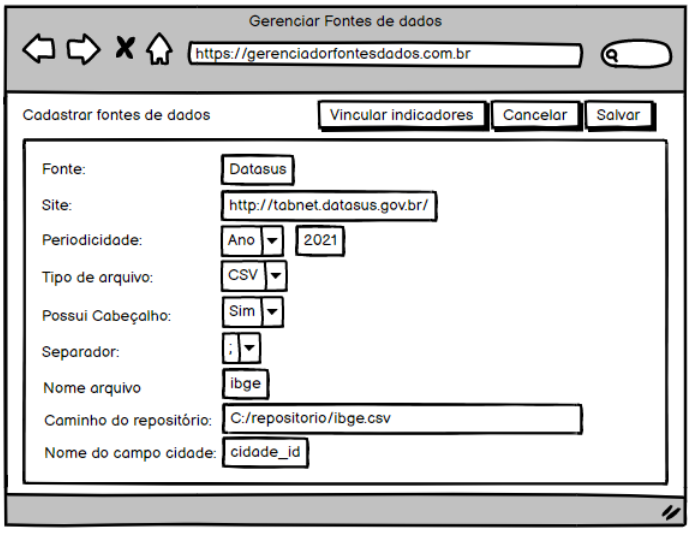
Figura 11 – Tela inicial



Fonte: Própria

O requisito 4 Gerenciar cadastro das fontes de dados define que o usuário pode visualizar, criar, atualizar e deletar as informações referentes às fontes de dados. A função desse formulário inicial apresentado na Figura 12 é identificar as informações básicas de uma base de dados, tais como nome, responsável pela produção, site onde pode ser encontrado, periodicidade, qual é a categoria de arquivo que armazena os dados brutos, existência ou não de cabeçalho, categoria de separador, nome do arquivo, caminho físico onde o arquivo se encontra e o nome do campo cidade que consta no arquivo.

Figura 12 – Cadastro fonte de dados

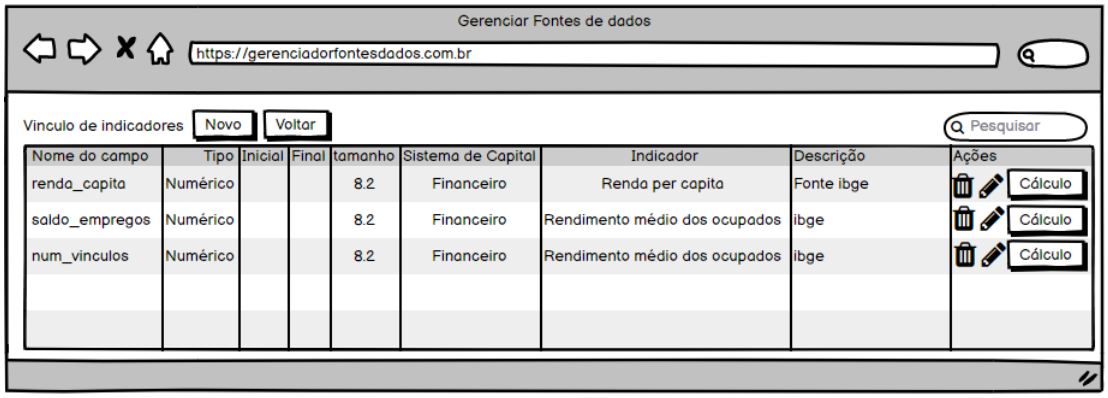


Fonte: Própria

O requisito 5 Vincular indicadores com as fontes de dados define que o usuário pode visualizar, criar, atualizar os indicadores do sistema de capital com os campos das fontes de dados. A regra de negócio associada define ser obrigatório ter realizado o cadastro da fonte de dados e a importação dos indicadores de sistema de capital. Será mostrada uma listagem com os campos do arquivo cadastrado identificando as informações básicas como tipo do campo,

tamanho, valor inicial e final quando o arquivo for txt e o indicador de sistema de capital que aquela informação pertence.

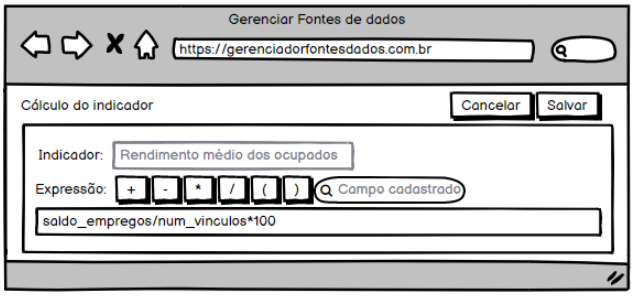
Figura 13 – Vincular indicadores



Fonte: Própria

O requisito 6 Gerenciar Cálculo de indicadores define que o usuário pode visualizar, criar, atualizar e deletar os cálculos de indicadores quando existir. A regra de negócio define que pode ser cadastrado somente um cálculo por indicador de um sistema de capital. O formulário da Figura 14 tem a função de criar uma expressão para um mesmo indicador que será calculado durante o processo de ETL.

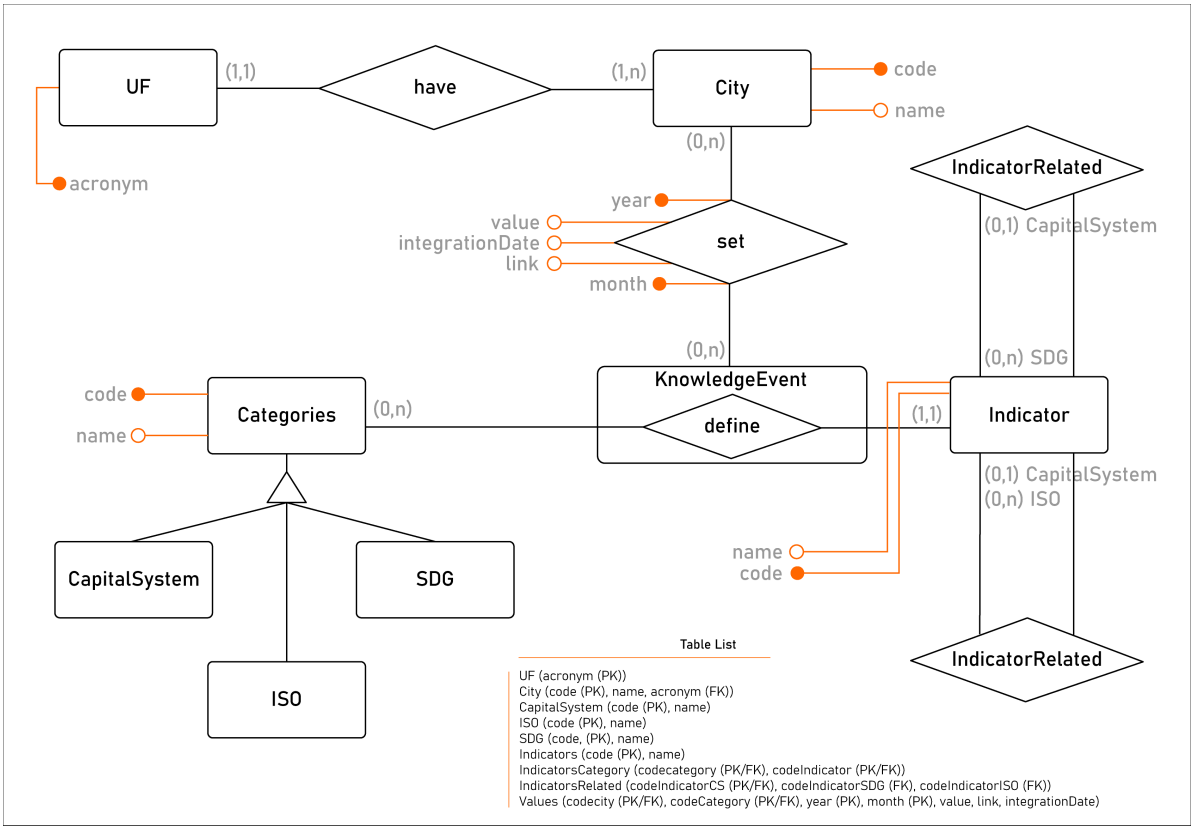
Figura 14 – Gerenciar Cálculo de indicadores



Fonte: Própria

O requisito 7 Executar o processo de ETL permite que o usuário execute o processo de integração dos dados para o banco de dados da ferramenta de Cidades do Conhecimento. A regra de negócio associada define que os arquivos das fontes de dados usados na integração devem respeitar o cadastro de vínculo de indicadores. O processo de ETL utiliza as informações cadastradas para o processamento dos dados, que permitirá que os dados brutos sejam transformados e armazenados no banco de dados da ferramenta de Cidades do Conhecimento. Os responsáveis pela ferramenta de Cidades do conhecimento estão desenvolvendo um novo banco de dados conforme diagrama ER Figura 15 e ficará pronto durante o desenvolvimento do software.

Figura 15 – Diagrama ER do banco de dados da Ferramenta Cidades do Conhecimento



Fonte: (NOTARI; BATTISTELO; MOLIN, 2019)

4.2.2 Modelo de dados do software

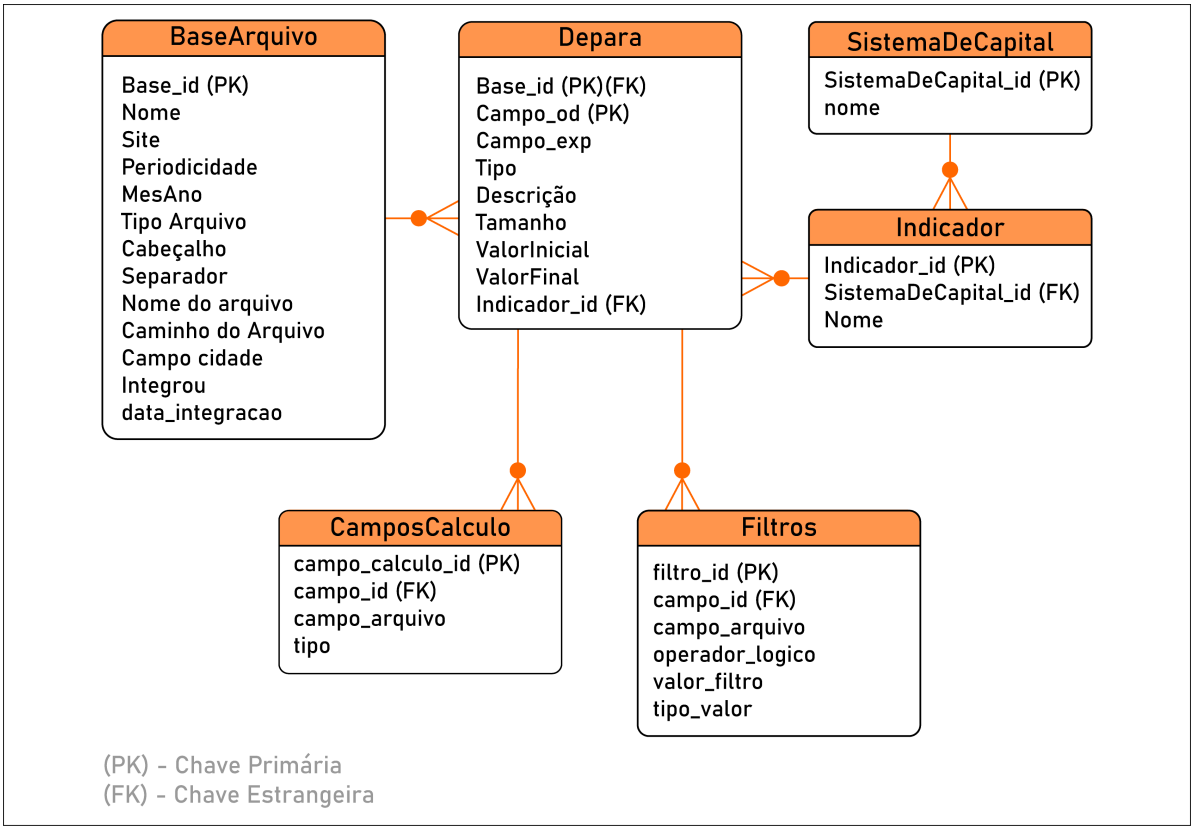
Com uma variedade grande de fontes de dados para integrar e por falta de um padrão na estrutura desses arquivos, para armazenar as informações simplificadamente, o banco de dados foi modelado conforme a Figura 16 utilizando o modelo lógico de banco de dados.

A tabela BaseArquivo é responsável por identificar as fontes de dados para a integração. Cada arquivo de dados brutos possui informações que, após o cadastro, ficarão salvas nessa tabela. As informações podem ser nome, link do site da origem da informação, periodicidade que define se é anual ou mensal, data das informações, qual é o tipo da extensão do arquivo, se possui cabeçalho no arquivo, o símbolo separador (se houver), nome do arquivo, caminho do repositório que o arquivo está salvo, campo cidade do arquivo usado, campo integrou para definir se os dados foram integrados e o campo data integração preenchido com a data que a integração foi concluída.

A tabela Depara é responsável por armazenar o vínculo dos campos da fonte de dados com os indicadores de sistema de capitais. Esta tabela possui o campo expressão podendo ser o nome do campo do arquivo ou a expressão de cálculo, o tipo do campo, descrição, tamanho do campo, valor inicial e valor final quando for arquivo txt (se houver) e o indicador vinculado.

A tabela CamposCalculo é responsável por armazenar os campos do arquivo utilizados

Figura 16 – Diagrama lógico do banco de dados



Fonte: Própria

na expressão de cálculo. Esta tabela possui o campo arquivo que é informado o nome do campo do arquivo e o tipo.

A tabela Filtros é responsável por armazenar todos os filtros utilizados na busca dos dados de um indicador durante o processo de ETL. Esta tabela possui o campo arquivo que é informado o nome do campo do arquivo, operador lógico podendo ser igual, maior, menor ou diferente, o valor do filtro que é a informação a ser procurada na fonte de dados e o tipo valor que é texto ou número.

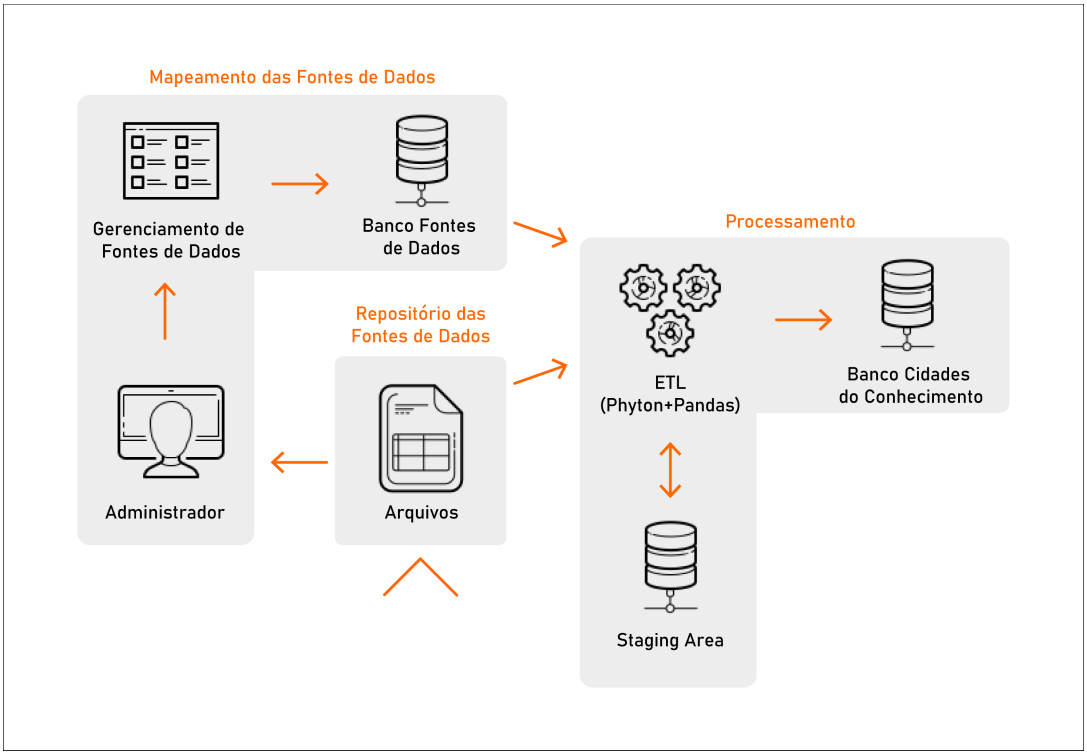
A tabela SistemaDeCapital é responsável por identificar qual categoria de capital contendo o código e nome.

A tabela Indicador é responsável por identificar os indicadores de cada categoria do sistema de capitais. Esta tabela possui o código do indicador, nome e o relacionamento do sistema de capital que o indicador pertence.

4.2.3 Arquitetura de software

A arquitetura pensada para a integração dos dados conforme o desenho Figura 17, se divide em repositório das fontes de dados, mapeamento das fontes de dados e processamento.

Figura 17 – Arquitetura da Solução



Fonte: Própria

O repositório de fontes de dados é um *data lake*[1] que é um tipo de repositório que armazena conjunto de dados brutos em formato nativo. Vão ficar todos arquivos com dados brutos que serão usados no processo ETL, e podem ser utilizados pora outras análises se necessário. O administrador irá disponibilizar manualmente as bases obtidas de diversas fontes.

O mapeamento das fontes de dados é feito pelo administrador que irá analisar os arquivos disponíveis para integração no repositório e cadastrar as informações básicas de cada arquivo na aplicação. Após esse cadastramento inicial, o administrador cadastra os vínculos dos campos do arquivo com os indicadores de um sistema de capital. Esse tem por objetivo registrar informações que serão utilizadas no momento do processamento das informações. Todas as informações cadastradas serão salvas em um banco de dados da aplicação.

A etapa de processamento é a etapa onde eles são modificados e preparados para a inserção no banco de dados da Aplicação de Cidades do Conhecimento. Nesse ponto são feitas as extrações dos dados brutos de seus arquivos originais e a normalização dos dados e cálculos necessários, bem como a carga final. Nesse processo são utilizados a linguagem Python e a biblioteca Pandas que, baseado nas informações previamente cadastradas, com os dados de entrada dos arquivos carregados em um dataframe Pandas, é feito o de para dos dados e realizado processo de integração.

[1] https://www.redhat.com/pt-br/topics/data-storage/what-is-a-data-lake

4.3 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento da proposta de solução, foi considerada uma maneira de solucionar os problemas na integração dos dados de entrada nos indicadores de Sistema de Capital da Aplicação de Cidades do Conhecimento. Durante o estudo, verificou-se que não seria possível criar nenhuma automatização na integração dos dados sem um cadastro prévio das fontes de dados que estão no *data lake* e os dados de entrada vinculados a cada indicador.

Para o correto funcionamento da integração, realizou-se a modelagem de um software para o cadastro das fontes de dados, a criação dos requisitos funcionais e os protótipos de tela, a explicação de cada requisito do software, a criação do diagrama lógico do banco de dados da nova aplicação e a explicação da arquitetura para a solução.

5 IMPLEMENTAÇÃO DA PROPOSTA DE SOLUÇÃO

Neste capítulo é apresentada detalhadamente a maneira em que foi implementada a proposta de solução. O capítulo abordará assuntos como: bibliotecas, frameworks, estrutura da aplicação e codificação. A arquitetura foi dividida em *client-server* e scripts para realizar integração dos dados. Para o desenvolvimento da aplicação, foram escolhidas as linguagens de programação python, javascript e typescript. No gerenciamento do banco de dados foi usado o MySQL. O framework Angular foi utilizado no *front-end* e o ambiente de execução Javascript Node.js no *back-end*.

5.1 FRAMEWORK E BIBLIOTECAS

Com o intuito de organizar e auxiliar o desenvolvimento da aplicação, foram utilizadas algumas bibliotecas. Para rodar a aplicação é necessário a instalação de algumas dependências conforme apêndice A.

- **Angular**[1]: O Angular é um framework de código aberto baseado em Typescript utilizado para a criação de *Single-Page Applications* (SPA), que é uma aplicação web consumida em uma única página;
- **ExpressJS**[2]: é uma estrutura da web que pode ser utilizada em conjunto com Node.js, no desenvolvimento de aplicações *back-end*. Possui um sistema de rotas completo e gerencia diferentes requisições *Hypertext Transfer Protocol* (HTTP) com seus mais diversos verbos;
- **Pandas**[3]: É uma biblioteca licenciada com código aberto que oferece estruturas de dados de alto desempenho, poderosa e flexível, construído com base na linguagem de programação Python;
- **PO UI**[4]: biblioteca de componente com código-fonte aberto para desenvolvimento de componentes de interface e *front-end* para aplicações web responsivas baseado em angular;
- **MySQL Connector**[5]: É uma biblioteca que estabelece a conexão entre o Python e o banco MySQL. Ela foi utilizada para realizar a leitura das informações das fontes de dados mapeadas;

[1] https://angular.io/
[2] https://expressjs.com/pt-br/
[3] https://pandas.pydata.org/
[4] https://po-ui.io/
[5] https://dev.mysql.com/doc/connector-python/en/

- **Sqlalchemy**[6]: Também é uma biblioteca que estabelece a conexão entre o Python e diferentes bancos de dados. Ela foi utilizada para promover a carga dos dados nas tabelas stages e fatos por apresentar um melhor desempenho quando combinada a biblioteca Pandas;

- **Chardet**[7]: É uma biblioteca de código aberto utilizada na detecção automática de codificação de caracteres.

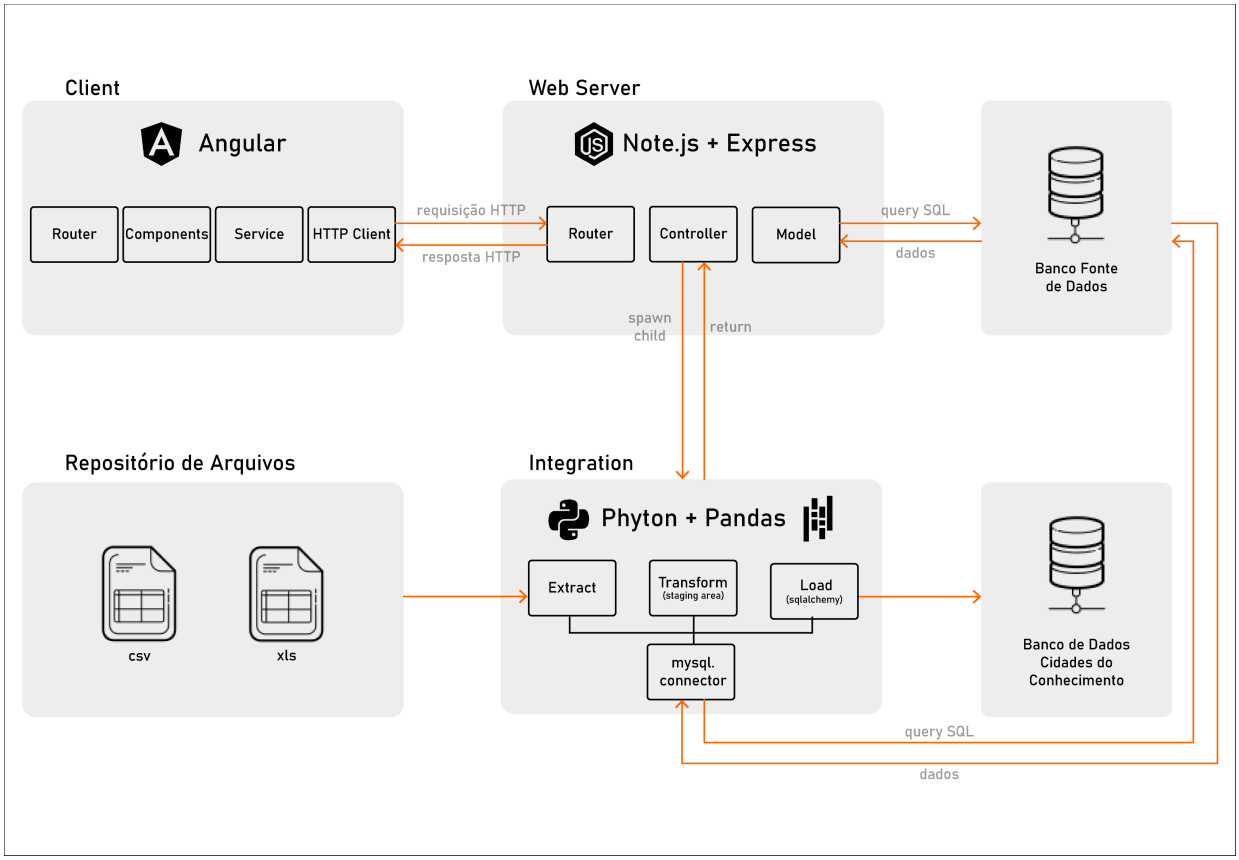
5.2 ESTRUTURA DA APLICAÇÃO

A aplicação *client-server* ficou dividida em *front-end* e *back-end*. O *Client* é a parte do *front-end* do sistema, responsável pela interface visual e de interação do usuário. É através dela que são realizada as requisições *Representational State Transfer* (REST), parte do *back-end*. Esta, no que lhe concerne, é o ponto de comunicação com o banco de dados em que ocorre o armazenamento e manipulação dos dados do sistema ou inicia a integração de dados criando um processo no sistema, com o que foi recebido ou solicitado pelo *Client*.

A comunicação entre tais módulos é feita através do protocolo HTTP. Na Figura 18 podemos ver um fluxograma que representa a comunicação. O Node.js Express utiliza APIs REST e interage com o banco de dados MySQL ou cria um novo processo de integração dos dados realizando o ETL com python e pandas. O Angular *Client* envia e recebe solicitações HTTP usando o HTTP Client, consumindo dados nos componentes. As rotas no Angular são usadas para navegar nas páginas web da aplicação. O processo de integração com Python e Pandas extrai os dados dos repositório e realiza o processo de ETL respeitando os parâmetros cadastrados na aplicação, por fim, carrega os dados para o banco da plataforma de Cidades do Conhecimento.

[6] https://www.sqlalchemy.org/
[7] https://chardet.readthedocs.io/en/latest/

Figura 18 – Arquitetura da aplicação



Fonte: Própria

5.3 FRONT-END

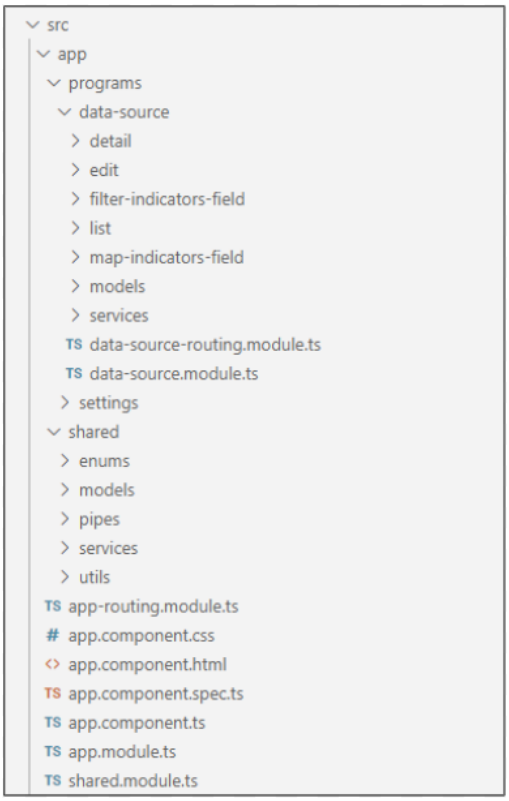
No *front-end* encontram-se os componentes necessários para apresentação da aplicação no navegador. A principal tecnologia utilizada para o desenvolvimento é o *framework* Angular, na versão mais recente até o momento, Angular 12.

Para organizar o *front-end*, foi implementada uma estrutura que visa fazer uma separação em módulos e componentes. A arquitetura do Angular permite organizar a aplicação por módulos através dos *NgModules*[8], que fornecem um contexto para os componentes serem compilados. Cada componente representa uma página ou algum elemento da aplicação formados por um arquivo TS (Typescript), um HTML e um CSS. Esses componentes usam serviços que fornecem funcionalidades específicas e são indiretamente relacionadas a essas visualizações. Os serviços são injetados nos componentes como dependências, tornando o código modular e reutilizável.

Conforme Figura 19 é possível identificar que os componentes de módulos são organizados da seguinte forma: módulos de programas com os componentes pertencentes a cada programa na pasta *programs* e módulo *shared*.

[8] https://angular.io/guide/ngmodules

Figura 19 – Estrutura de Diretórios do front-end



Fonte: Própria

O módulo *Shared*, que tem como objetivo centralizar e reaproveitar componentes, pode ser importado em qualquer outro módulo quando esses itens forem reutilizados. São exemplos *services*, *pipes*, *models*, *utils* e *enums*. O módulo compartilhado não tem nenhuma dependência do restante do aplicativo e não depende de nenhum outro módulo.

Cada programa na pasta *programs* é um módulo que pode ter diversos componentes. Na Figura 19 o programa *data-source* possui os principais componentes da aplicação para o usuário poder realizar as ações.

- ***detail***: Mostra todas as informações referente às fontes de dados e indicadores cadastrados;
- ***edit***: Cadastro e alteração das informações da fonte de dados;
- ***filter-indicators-field***: Cadastro e alteração dos filtros de um indicador para realizar na fonte de dados;
- ***list***: listagem de todas as fontes de dados com paginação. Mostra o status da integração e permite selecionar os registros para integração ou cópia para um novo cadastro;
- ***map-indicators-field***: cadastro e alteração do de para indicador com a informação de um campo do arquivo de fonte de dados ou uma expressão de cálculo.

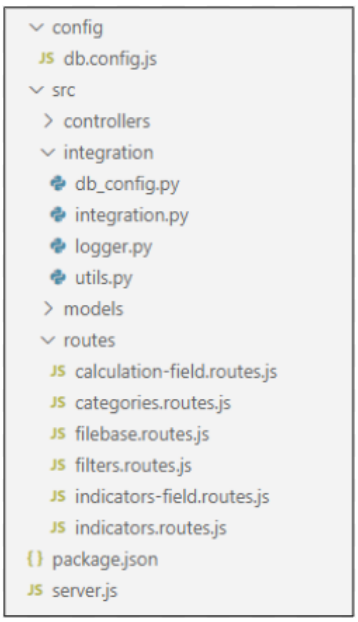
5.4 BACK-END

Para construir o back-end do sistema foi implementada uma *Application Programming Interface* (API) RESTful usando Node.js. O Node.js é um ambiente de execução Javascript *server-side*, isso significa que pode rodar uma aplicação não dependendo de um *browser* para execução. O objetivo principal desta aplicação é fornecer e manipular os dados do sistema, que ficam armazenados no banco de dados e criar a fila de integração das fontes de dados.

Uma API permite a comunicação entre aplicações. As API's foram implementadas utilizando a arquitetura REST, que consiste em princípios, regras e limitações para aplicações web. Quando uma API é implementada sobre a arquitetura REST, é dito que tal API é RESTful (MASSE, 2011).

A Figura 20 contém a estrutura de pastas: *routes*, *controllers*, *models* e *integration*. Os modelos são criados e executados às *queries* do banco de dados e definem a estrutura dos dados, sendo tais estruturas similares às tabelas do banco de dados, e os controladores definem e executam os métodos que manipulam tais dados, realizando ações como ler, criar, atualizar e excluir ou abrir um processo de integração. As rotas são utilizadas para chamar o controlador responsável pelo serviço e o método que irá executá-lo. Na pasta *integration* estão os códigos escritos em python que realizam o processo de ETL das fontes de dados.

Figura 20 – Estrutura de Diretórios do back-end



Fonte: Própria

Para acessar tais métodos, o Node.js, em conjunto com o framework Express, mapeia os serviços disponíveis pela aplicação usando rotas e métodos do padrão HTTP, como GET, POST, PUT e DELETE. Assim, quando uma requisição HTTP é efetuada para a API, o mapeamento de rotas é usado para chamar o controlador responsável pelo serviço e o método que irá executá-lo.

A pasta *integration* da Figura 20, possui os arquivos em python responsáveis pela integração dos dados. O arquivo db-config.py realiza as conexões com os bancos de dados utilizando as bibliotecas Sqlalchemy e MySQL Connector. O integration.py executa todo fluxo ETL para extrair os dados dos arquivos, limpar e carregar no banco da aplicação de Cidades do Conhecimento. O logger.py cria um arquivo de registro das atividades da integração. O utils.py disponibiliza funções que podem ser utilizadas em toda aplicação python.

Para executar a integração o usuário seleciona um ou mais registros para serem integrados no *front-end*, com isso, é criada uma requisição usando o método HTTP POST. O servidor *back-end* recebe essa requisição, é feito um mapeamento de rotas, usando o método HTTP para identificar qual serviço precisará ser realizado e passar para o controlador conforme exemplo do Algoritmo 1 na linha nove.

Algoritmo 1 – Trecho do arquivo filebase.routes.js

```
1 const express = require('express')
2 const router = express.Router()
3 const fileBaseController = require('../controllers/filebase.controller');
4
5 // Retrieve all fileBases
6 router.get('/', fileBaseController.findAll);
7
8 //Integration filebases
9 router.post('/integration', fileBaseController.integration);
```

Fonte: Própria

Nesse caso irá chamar o método *integration* no controlador para criar a fila de integração. Esse método recebe uma lista de um ou mais registros e montará a *query* do banco de dados que irá mudar o status dos registros para ficar na fila de integração. Após executada a *query*, é executado o método *child process spawn* que gera o processo filho de forma assíncrona, sem bloquear o loop de eventos do Node.js, para executar o script python que irá realizar a integração conforme trecho do método *integration* no algoritmo 2.

Algoritmo 2 – Gerando processo filho.js

```
1 ...
2 FileBase.updateStatusPreProcess(filterInCode,
3     function(err, filebase) {
4         if (err)
5             res.send(err);
6         var spawn = require('child_process').spawn;
7         spawn('python', ['src/integration/integration.py']);
8         res.json({ error:false, message: 'Integration started' });
9 });
```

Fonte: Própria

5.5 INTEGRAÇÃO

A aplicação permite ao usuário realizar o mapeamento das fontes de dados e o cadastro dos vínculos dos indicadores, filtros e cálculos caso necessário. Essas informações serão utilizadas para o processamento dos dados, isso é o processo de ETL, que permitirá que os dados brutos sejam transformados em informações e integradas ao banco de Cidades do Conhecimento. Para isso, foram desenvolvidos scripts em linguagem Python que, baseados nas informações previamente cadastradas, conseguissem realizar tal conjunto de tarefas.

Ao iniciar a integração é realizada uma busca nos registros de fontes de dados da aplicação com status aguardando integração (*status-integration* = 'A'), alterado no Node.js antes de abrir o processo filho, conforme aparece na etapa A da Figura 21. Existindo fontes para integrar, é realizado uma busca das características de cada fonte de dado que foi mapeada. Nessa etapa retornam informações da tabela BaseArquivo Figura 16, do banco de dados da aplicação.

Para facilitar o processo de ETL, optou-se pela criação de uma tabela temporária que age como sendo uma "*Staging Area*", para a carga e tratamento dos dados brutos. Por se tratar de uma área temporária, há a garantia de que, não serão inseridos dados inconsistentes no banco de dados Cidades do Conhecimento.

Após apagar os dados da tabela na "*Staging Area*", é alterado o status de integração para integrando (*status-integration* = 'I'), com isso, o usuário utilizando a aplicação consegue identificar quais registros estão integrando. Ao alterar o status para integrando, é realizada uma busca de todos os indicadores vinculados às fontes de dados e nessa etapa retornam as informações das tabelas Depara e CamposCalculo, Figura 16.

Com a lista de indicadores para integrar, é iniciado o processo de extração dos dados dos arquivos, etapa B da Figura 21. Antes de realizar a leitura do arquivo com o Pandas é verificado qual o *econding*, sendo este a codificação de caracteres, utilizando a biblioteca Chardet. Depois o algoritmo realiza a leitura dos dados dos arquivos com base nas características mapeadas, tais como a categoria de arquivo (csv ou xls), categoria de delimitador de campos e quais os campos do arquivo serão utilizados para não trazer informações desnecessárias na memória. Os dados são enviados para um Dataframe do Pandas, que é uma estrutura de dados bidimensional com os dados alinhados de forma tabular em linhas e colunas. A extração dos dados é realizada agrupando mil linhas de dados para o Dataframe por vez para não sobrecarregar a memória durante a integração.

Na sequência o algoritmo verifica se existem filtros mapeados, tabela Fitros da Figura 16 para o indicador que esta sendo integrado, se existir, é aplicado o filtro utilizando a função do Pandas *df.query()* no Dataframe. A seguir, o algoritmo faz a conversão das categorias de dados necessárias para os campos de valores.

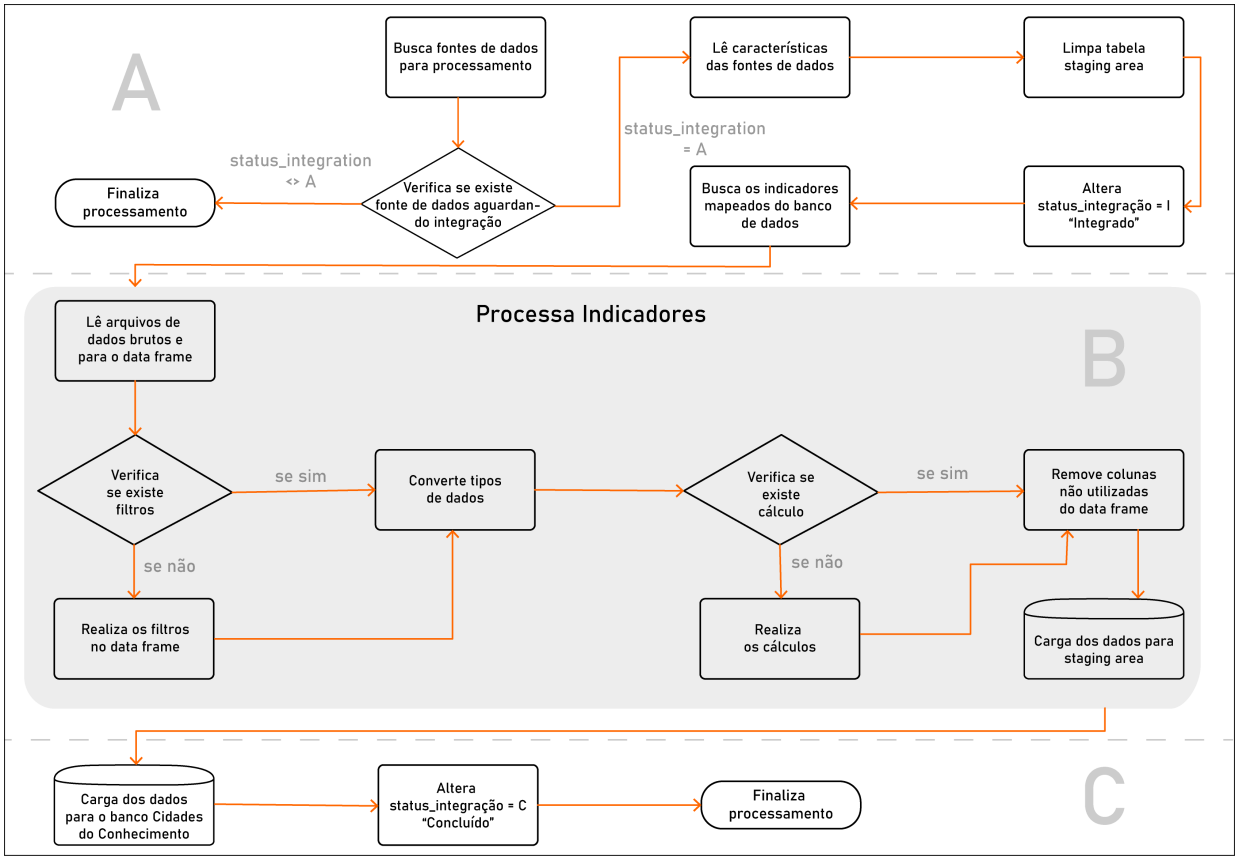
Continuando, o algoritmo verifica se existe o cadastro de uma expressão para realizar o cálculo para o indicador, se existir é usado a função do Pandas *df.eval()*. Nesta etapa a expressão

já vem definida pelo usuário ao realizar o mapeamento dos dados na aplicação, com isso, é aplicado a expressão cadastrada na função para executar o cálculo, para isso, todos os campos cadastrados que serão utilizados na expressão, são convertidos para o tipo correto.

Seguindo o fluxo do algoritmo, são removidas as colunas que não serão mais utilizadas no Dataframe. Com a limpeza realizada no Dataframe, passa-se a ter somente os campos código da cidade e o valor ou resultado do cálculo, extraídos do arquivo. Neste mesmo Dataframe são renomeados os campos para *codeCity* e *value* e na sequência criado mais quatro campos, *codeIndicator*, *year*, *month* e *link* inserindo as informações da fonte de dados mapeada na aplicação conforme o que é esperado no banco de dados Cidades do Conhecimento. Após realizadas essas alterações no Dataframe, são inseridos os dados na "*Staging area*".

Ao finalizar a limpeza, manipulação e a carga para a "*Staging area*", os dados serão copiados e inseridos na tabela *values* do banco Cidades do Conhecimento Figura 15, etapa C da Figura 21. Para concluir, é atualizado o status de integração para concluído (*status-integration* = 'C') e o processo é finalizado.

Figura 21 – Etapa final da integração



Fonte: Própria

5.5.1 Tratamento de Erros

O usuário precisa conseguir identificar os erros que podem ocorrer durante o processamento de dados, com isso, foi utilizada a biblioteca *"Logging"* que possibilita realizar os registros das atividades. A cada integração é criado um arquivo de log novo no caminho configurado.

Em todas as etapas da integração foram utilizados tratamentos de exceções conforme exemplo do Algoritmo 3. Caso ocorra algum erro, além de gravar no arquivo de acompanhamento é modificado o status de integração para erro (*status-integration* = 'E').

Algoritmo 3 – Tratamento de exceção em Python

```
1 try:
2   df = pd.read_csv(v_directory, delimiter = v_separator)
3   logger.info('Lendo csv')
4 except Exception as e:
5   logger.error('Erro na leitura do csv')
6   logger.error(e)
7   utils.setStatusError(v_code)
```

Fonte: Própria

Durante o ETL, após concluir a transformação dos dados processando todos indicadores da fonte de dados, os dados que estão na *staging area*, são copiados para o banco oficial, nesta etapa utilizando o tratamento de exceções, caso algum erro ocorra é realizado o *rollback* da execução, garantindo a integridade do banco de dados. Se não ocorrer erros, é feito o *commit* da execução para gravar os dados em definitivo no banco de dados.

Dessa forma, o usuário da aplicação pode acompanhar o processamento dos dados e caso algum problema ocorra durante o processamento, será possível identificar quais são os registros na aplicação que estão com o status de erro e poder encontrar o arquivo de log para análise.

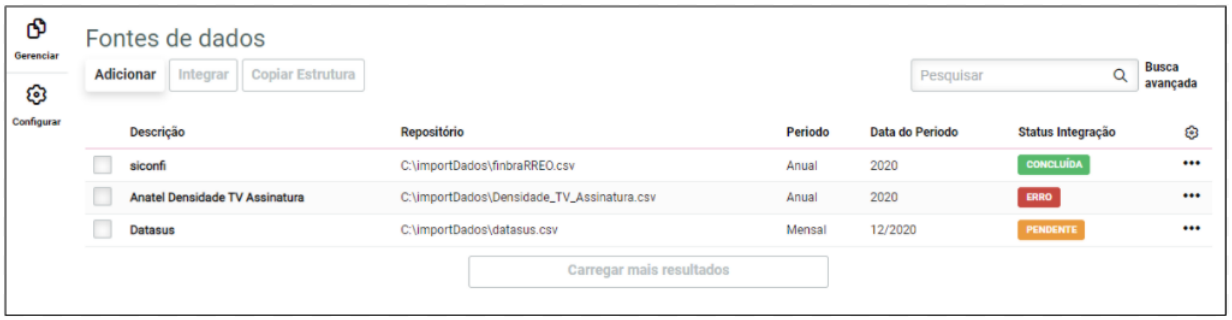
6 ESTUDO DE CASO

Este capítulo contempla um estudo de caso acerca da aplicação desenvolvida com o intuito de validar se os problemas citados no capítulo 4 foram resolvidos ou minimizados e se contempla todos os requisitos da proposta solução. No estudo de caso é apresentado o funcionamento da aplicação, particularidades das fontes de dado e as fontes de dados usadas.

6.1 FUNCIONAMENTO DA APLICAÇÃO

O usuário terá disponível em sua tela inicial Figura 22 a opção de gerenciar fontes de dados, uma listagem com paginação de todas as fontes de dados já cadastradas, podendo verificar qual o status da integração de cada fonte de dados. A opção 'Configurar' no menu ao lado, abre a tela para importação dos Sistemas de Capitais ou dos Indicadores utilizados na aplicação. As três principais ações da tela inicial são, adicionar, integrar e copiar estrutura. A opção adicionar irá abrir uma tela para mapear uma nova fonte de dados. Ao selecionar um ou mais registros serão habilitados os botões para integração, ou realizar a cópia da estrutura cadastrada. Ao clicar no link da coluna 'Descrição' de algum dos registros da lista será direcionado para tela de detalhes.

Figura 22 – Tela inicial

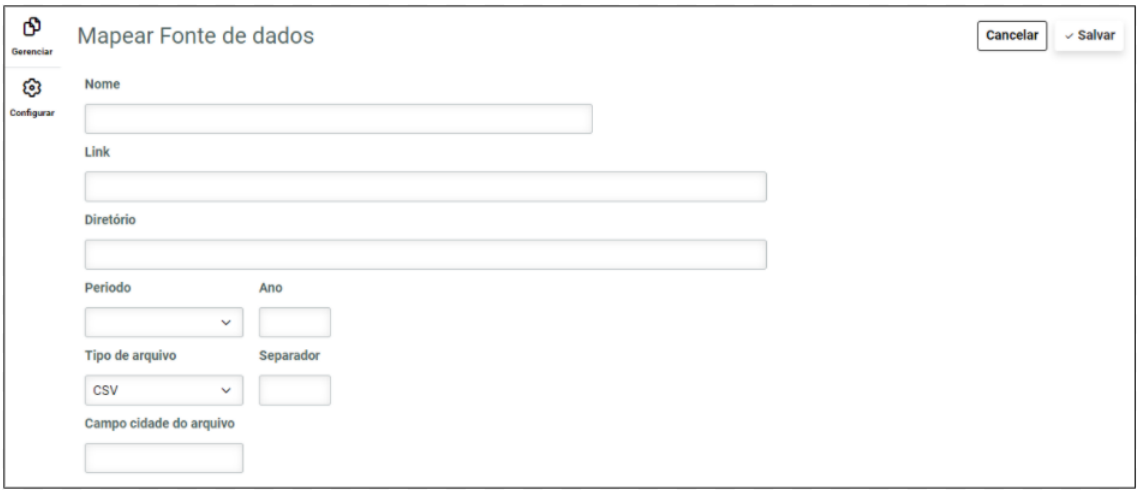


Fonte: Própria

Ao clicar no botão 'Adicionar' irá abrir a tela para mapear as fontes de dados Figura 23, em que é possível informar o nome da fonte de dados, link da origem dos dados, diretório que o arquivo foi salvo, período, que pode ser anual ou mensal, categoria de arquivo, que no momento só possui a opção CSV, separador utilizado para identificar as colunas, e, por fim, o campo cidade do arquivo, onde é informado qual o campo do arquivo que possui o código da cidade.

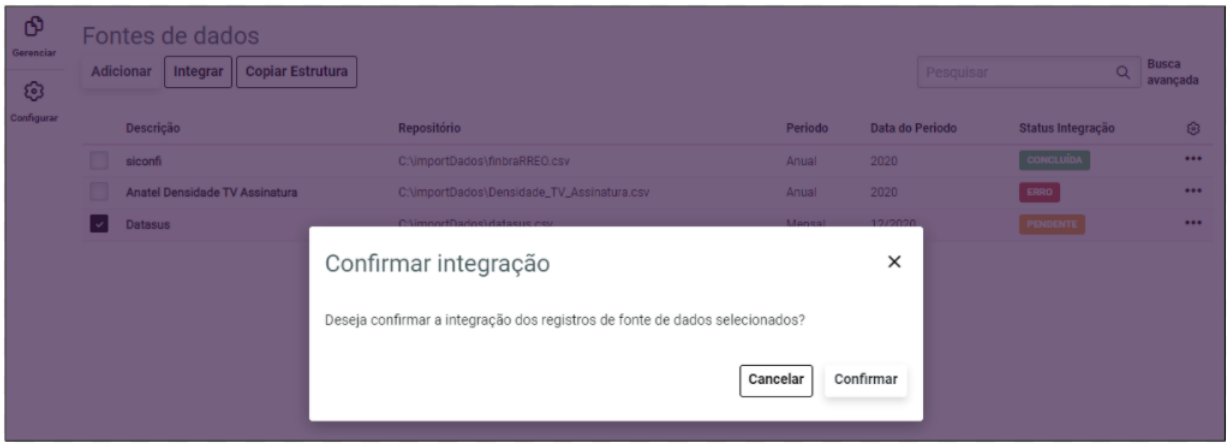
Selecionando um ou mais registros da listagem com status igual a pendente ou erro, será desabilitado o botão 'Integrar' podendo ser iniciada uma nova integração Figura 24.

Figura 23 – Mapear uma fonte de dados



Fonte: Própria

Figura 24 – Realizando Integração



Fonte: Própria

6.1.1 Detalhes e vínculo de indicadores

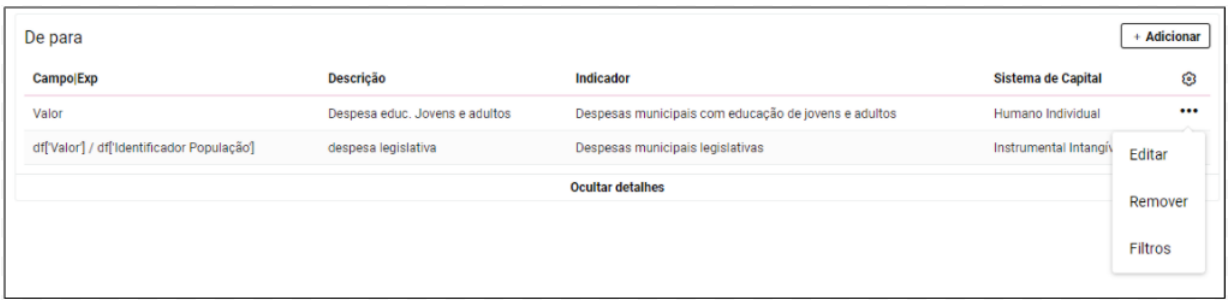
Ao entrar na tela de detalhes Figura 25, é possível visualizar todas as informações já cadastradas referente à fonte de dados, e, além disso, um espaço ’De para’ para realizar o vínculo das informações do arquivo com indicadores do Sistema de Capital. Ao clicar no botão ’Adicionar’ irá abrir uma tela podendo informar o campo do arquivo ou criar uma expressão que será calculada no processo de integração, vinculando a um indicador.

Em cada registro ’De para’ cadastrado é possível realizar uma ação Figura 30, editar, remover ou abrir a tela para cadastrar filtros de busca.

Figura 25 – Tela de detalhes da fonte de dados

Fonte: Própria

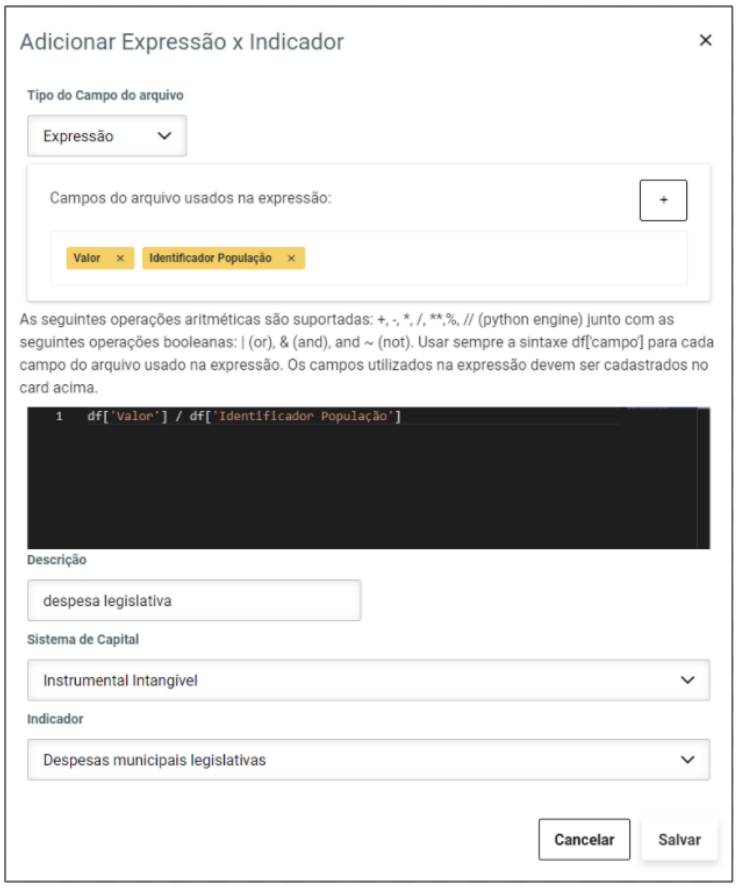
Figura 26 – Ações do registro de vínculo do indicador



Fonte: Própria

Ao abrir a tela para adicionar um novo vínculo ao indicador é possível escolher qual o tipo do campo do arquivo, se selecionar a opção ’Expressão’ Figura 27, irá disponibilizar um espaço para informar quais campos do arquivo serão utilizados na expressão, necessário informar os campos que serão utilizados no cálculo, pois serão utilizados na leitura do arquivo trazendo somente as informações necessárias. A expressão deve respeitar as operações aritméticas e operações booleanas, portanto nesta etapa é necessário um nível básico de conhecimento em lógica de programação. Quando utilizado um campo na expressão é obrigatório o uso da sintaxe df[’campo’], com isso, a expressão já chega pronta para ser executada no processo de ETL. No exemplo da Figura 27 está sendo feito um cálculo do valor dividido pela população, dois campos do arquivo, cadastrados no *card* de campos dos arquivos usados na expressão.

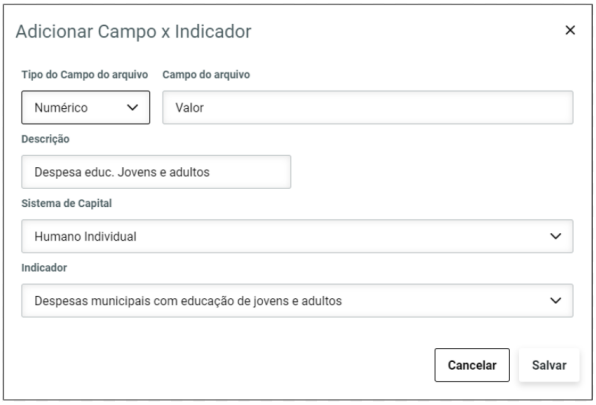
Figura 27 – Cadastro expressão x indicador



Fonte: Própria

Ao abrir a mesma tela para adicionar um novo vínculo ao indicador, escolhendo o tipo do campo do arquivo como numérico Figura 28, é necessário informar qual o campo do arquivo de dados que será buscado o valor do indicador. Nesse caso não irá sofrer nenhuma alteração, será copiado direto o valor do arquivo.

Figura 28 – Cadastro campo x indicador



Fonte: Própria

Caso seja aberta a tela de cadastro de filtros Figura 29, é possível informar qual o campo

do arquivo, operador lógico e a informação a ser procurada no campo informado. Foi necessário disponibilizar o cadastro de filtros, alguns arquivos possuem informações de diversos indicadores, sendo necessário procurar somente as corretas.

Figura 29 – Filtros do De para

Fonte: Própria

6.2 PARTICULARIDADES DAS FONTES DE DADOS

Foram realizados testes somente com fontes de dados CSV e dentre os arquivos deste tipo, existem particularidades. A primeira dificuldade encontrada no decorrer do projeto foram os arquivos que possuem dados de diferentes anos e meses. Em primeiro momento a situação foi resolvida manualmente, abrindo o arquivo e copiando em diversos outros arquivos respeitando as datas.

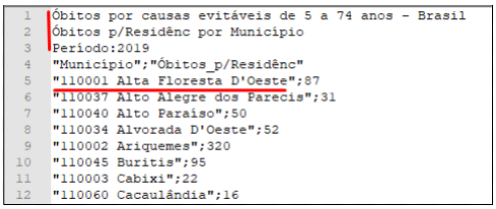
Ao continuar com testes de fontes de dados diferentes para outros indicadores, foi encontrado outro problema que é o fato de uma fonte de dados, poder ter vários indicadores com o mesmo campo valor, com isso, se tornou muito complexo ajustar os arquivos manualmente para poder utilizar na aplicação, visto que o objetivo da aplicação é buscar automatizar a maior quantidade possível de processos. A partir desse segundo problema foi pensado na solução de cadastrar filtros na aplicação para serem utilizados durante o processo de ETL resolvendo os dois problemas.

Pequenos ajustes ainda são necessários realizar manualmente em alguns arquivos para funcionarem plenamente na integração. Os arquivos Departamento de informática do Sistema Único de Saúdedo Brasil (DATASUS)[1], Siconfi[2], Banco Central do Brasil (BCB)[3], Sistema IBGE de Recuperação Automática (SIDRA)[4] que possuem linhas acima do cabeçalho das colunas precisam ser removidas manualmente. Outro exemplo mais particular, além das linhas acima do cabeçalho as fontes de dados do DATASUS, possuem o código e descrição da cidade

[1] http://www2.datasus.gov.br/DATASUS/index.php?area=0201
[2] https://siconfi.tesouro.gov.br/siconfi/pages/public/conteudo/conteudo.jsf
[3] https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios
[4] https://sidra.ibge.gov.br/tabela/5937

na mesma coluna Figura 30, dando erro de categoria de dados diferentes durante a integração, portanto, é necessário um ajuste manual nesse caso, adicionado o separador na coluna. Outra particularidade para as fontes de dados DATASUS, que para realizar alguns cálculos é necessário agrupar os dados de fontes separadas em um único arquivo para integração, pelo fato das fontes de dados DATASUS serem disponibilizadas separadas por indicadores.

Figura 30 – Particularidade do fonte de dados DATASUS



Fonte: Própria

Na Tabela 1 possui a lista da origem das fontes de dados extraídas e testadas na aplicação mostrando se necessário ajuste para funcionar a integração dos dados. As únicas fontes de dados que não precisam de pequenos ajustes manuais são Agência Nacional de Telecomunicações (ANATEL)[5] e Sistema Nacional de Informações sobre Saneamento (SNIS)[6].

Tabela 1 – Ajustes manuais nas fontes de dado

Origem da fonte de dado	**Ajustes manuais**
Siconfi	Sim
ANATEL	Não
SIDRA	Sim
BCB	Sim
DATASUS	Sim
SNIS	Não

Fonte: Própria.

6.3 FONTES DE DADOS USADAS

Durante o desenvolvimento da aplicação e para validação, foram realizados vários testes utilizando diversas fontes de dados. As fontes de dados possuem diversos tamanhos e todas aceitas pela aplicação que foi desenvolvida para não sobrecarregar a integração. A fonte de dado com maior número de linhas é o do Siconfi do ano 2020 com aproximadamente 3 milhões de linhas de registro e um arquivo. Com as fontes testadas foi possível obter o resultado de 40 categorias de indicadores integradas de um total de 67, cerca de 60%, totalizando 222.720 mil registros integrados no banco de Cidades do Conhecimento. Na Tabela 2 lista dos indicadores utilizados.

[5] https://dados.gov.br/organization/agencia-nacional-de-telecomunicacoes-anatel
[6] http://app4.mdr.gov.br/serieHistorica/

Tabela 2 – Indicadores testados com sucesso

Indicadores	Origem dos dados
Despesas municipais com cultura	Siconfi
Despesas municipais com planejamento e orçamento	Siconfi
Densidade de Telefonia Fixa	ANATEL
Densidade de TV por Assinatura	ANATEL
Densidade de telefonia móvel	ANATEL
Densidade de banda larga fixa	ANATEL
Despesas municipais com desporto comunitário	Siconfi
Despesas municipais com lazer	Siconfi
Mortes por causas externas (acidentes e violências) a cada dez mil hab	DATASUS
Número de divórcios concedidos	SIDRA
Despesas municipais com comércio e serviços	Siconfi
Poupança média por habitante	BCB
Despesas municipais com saúde	Siconfi
Despesas municipais com assistência comunitária	Siconfi
Despesas municipais com habitação	Siconfi
Despesas municipais com saneamento rural e urbano	Siconfi
Mortes por doenças de alto impacto a cada 100 mortes	DATASUS
Óbitos por causas evitáveis em menores de 5 anos	DATASUS
Óbitos por causas evitáveis de 5 a 74 anos	DATASUS
Imunizações BCG	DATASUS
Despesas municipais com assistência à criança e ao adolescente	Siconfi
Despesas municipais com educação de jovens e adultos	Siconfi
Despesas municipais com educação especial	Siconfi
Mortalidade infantil	DATASUS
Percent de mortes em acidentes de trânsito c/ relação ao total de mortes	DATASUS
Despesas municipais com segurança pública	Siconfi
Despesas municipais com gestão ambiental	Siconfi
Despesas municipais com transporte	Siconfi
Índice atend de esgoto referido aos munic atendidos com água	SNIS
Índice de atendimento total de água	SNIS
Índice de coleta de população atendida no munic por coleta de resíduos	SNIS
Número de leitos hospitalares para cada mil habitantes	DATASUS
Despesas municipais legislativas	Siconfi
Despesas municipais com ação legislativa	Siconfi
Despesas municipais com administração	Siconfi
Despesas municipais com administração financeira	Siconfi
Despesas municipais com controle interno	Siconfi
Despesas municipais com ensino fundamental	Siconfi
Despesas municipais com ensino médio	Siconfi
Despesas municipais com ensino superior	Siconfi

Fonte: Própria.

6.4 CONSIDERAÇÕES FINAIS

O estudo de caso da aplicação mostrou as funcionalidades da aplicação, detalhando o modo de utilização com alguns exemplos de parametrizações, as particularidades das fontes de dados testadas que a aplicação aceita para realizar a integração e quais foram as fontes de dados usadas com a quantidade de indicadores integrados.

Desta forma, uma relação entre a aplicação e os 7 requisitos definidos na etapa da proposta de solução no capítulo 4 pode ser feita conforme Tabela 3. Pela grande quantidade de indicadores e fontes de dados mapeadas pelos responsáveis da ferramenta, para dar um foco maior no processo de integração desses dados, foi implementada e testada a aplicação para fontes de dados com extensão CSV até o momento. Outros tipos de fontes de dados não testadas possui extensão XLS. Com isso, foi realizada a implementação completa de grande parte dos requisitos, tendo como parcial, apenas o requisito 4 Gerenciar cadastro das fontes de dados.

Tabela 3 – Relação entre implementação e requisitos

Requisito	**Nome**	**Implementação**
R1	Importar Categorias de Capital	Completa
R2	Importar indicadores	Completa
R3	Copiar cadastro das fontes de dados	Completa
R4	Gerenciar cadastro das fontes de dados	Parcial
R5	Vincular indicadores com as fontes de dados	Completa
R6	Gerenciar Cálculo de indicadores	Completa
R7	Executar o processo de ETL	Completa

Fonte: Própria.

7 CONCLUSÕES

A Plataforma Cidades do Conhecimento tem o intuito de armazenar, calcular e consultar os indicadores de Sistema de Capitais para identificar e estudar o desenvolvimento de cidades. Plataforma que surgiu a partir de estudos realizados desde 2014, através do conceito de Sistema de Capitais, que possibilita um melhor entendimento sobre Cidades do Conhecimento. Atualmente a carga dos dados consumidos na plataforma, é realizada manualmente, extraindo os dados de diversas fontes e agrupando em uma planilha. Com isso, a confiabilidade dos dados é baixa, além de gerar um grande esforço manual.

Durante o desenvolvimento da aplicação foram realizados testes com diferentes tipos de fontes de dados, e verificadas as diferenças entre cada arquivo. Foi necessário um encontro com as pessoas responsáveis pela plataforma de Cidades do Conhecimento para entender como extrair os dados para alguns indicadores. Ao analisar as particularidades, foi possível iniciar a implementação da aplicação e do processo de integração.

A solução proposta no presente projeto foi o desenvolvimento de uma aplicação para mapear e gerenciar a integração das fontes de dados de diferentes origens, vinculando os valores ou expressões de cálculo em um indicador referente a um Sistema de Capital. Baseado no mapeamento, é realizada a integração dos dados de forma automática, resolvendo o problema de entrada de dados incorretos e diminuindo o trabalho manual. Ressaltando que algumas fontes de dados precisam de configurações manuais para ser aceitas pela aplicação conforme seção 6.2. Os resultados obtidos foram satisfatórios, integrando os dados para a maior parte dos indicadores e desenvolvido as funcionalidades da aplicação, ficando como pendência apenas a integração de todos os indicadores, visto que os testes da aplicação foram realizados integrando somente fontes de dados em formato CSV.

Logo, a aplicação desenvolvida resolveu os requisitos propostos, possibilitando um ganho de tempo e uma aplicação para facilitar o gerenciamento das fontes de dados integradas, com isso, a plataforma de Cidades do Conhecimento terá disponível mais dados, ganhando assim qualidade, o que poderá ser somado e utilizado em análises dos indicadores das cidades.

A partir das análises, testes, problemas e pendências verificadas no projeto, surgem possibilidades para trabalhos futuros, visto que a aplicação pode estar em constante evolução:

- Identificar uma forma de não precisar realizar nenhum ajuste manual nas fontes de dados, parametrizando tudo no sistema;
- Tratar a integração dos dados para diversas categorias de arquivos, permitindo usar a aplicação de coleta de dados desenvolvida nesse trabalho para todos os indicadores.

REFERÊNCIAS

ALMEIDA, J. G. *et al.* **Extração, Integração e Importação de Dados Heterogêneos em Cidades Inteligentes: Um Mapeamento Sistemático**. Porto Alegre, RS, Brasil: SBC, 2020. 57–70 p. ISSN 2595-2706. Disponível em: <https://sol.sbc.org.br/index.php/courb/article/view/12353>.

BOUZEGHOUB, M. *et al.* Heterogeneous data source integration and evolution. In: HAMEURLAIN, A.; CICCHETTI, R.; TRAUNMÜLLER, R. (Ed.). **Database and Expert Systems Applications**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002. p. 751–757.

CARRILLO, F. J. **Capital systems: implications for a global knowledge agenda**. [S.l.]: Journal of Knowledge Management, 2002. v. 6. 379-399 p.

____. **Capital cities: a taxonomy of capital accounts for knowledge cities**. [S.l.]: Journal of Knowledge Management, 2004. v. 8. 28-46 p.

____. From transitional to radical knowledge-based development. **Journal of Knowledge Management**, v. 10, n. 5, 2006.

CARRILLO, F. J. *et al.* **Knowledge and the city: Concepts, applications and trends of knowledge-based urban development**. [S.l.]: Routledge, 2014.

CARVANO, L. M. F. **A utilização de dados públicos abertos na construção de um Data Warehouse: A construção de um repositório estatísticas educacionais públicas brasileiras**. [S.l.]: Information Management School, 2018.

CHANG, D. L. *et al.* Knowledge-based, smart and sustainable cities: a provocation for a conceptual framework. **Journal of Open Innovation: Technology, Market, and Complexity**, v. 4, n. 1, 2018. Disponível em: <https://www.mdpi.com/2199-8531/4/1/5>.

DOAN, A. *et al.* Toward a system building agenda for data integration. **CoRR**, abs/1710.00027, 2017. Disponível em: <http://arxiv.org/abs/1710.00027>.

ERGAZAKIS, K.; METAXIOTIS, K.; PSARRAS, J. **Towards knowledge cities: conceptual analysis and success stories**. [S.l.]: Journal of Knowledge Management, 2004. v. 8. 5-15 p.

FACHINELLI, A. C.; CARRILLO, F. J.; D'ARISBO, A. Capital system, creative economy and knowledge city transformation: Insights from bento gonçalves, brazil. **Expert Systems with Applications**, v. 41, n. 12, p. 5614–5624, 2014.

FORTINI, P. M. a.; DAVIS, C. A. **Analysis, Integration and Visualization of Urban Data From Multiple Heterogeneous Sources**. Association for Computing Machinery, 2018. 17–26 p. Disponível em: <https://doi.org/10.1145/3284566.3284569>.

KIMBALL, R.; CASERTA, J. **The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming and Delivering Data**. Hoboken, NJ, USA: John Wiley Sons, Inc., 2004. 467 p. ISBN 0764567578.

LAZAROIU, G. C.; ROSCIA, M. Definition methodology for the smart cities model. **Energy**, v. 47, n. 1, p. 326–332, 2012.

LEMINEN, S.; WESTERLUND, M.; NYSTRöM, A.-G. **Living Labs as Open-Innovation Networks**. [S.l.]: Technology Innovation Management Review, 2012. 6-12 p.

MASSE, M. **REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces**. O'Reilly Media, 2011. ISBN 9781449319908. Disponível em: <https://books.google.com.br/books?id=eABpzyTcJNIC>.

MICHELAM, L. D.; CORTESE, T. T. P.; YIGITCANLAR, T. O desenvolvimento urbano baseado no conhecimento como estratégia para promoção de cidades inteligentes e sustentáveis. In: **Anais do Simpósio Brasileiro Online de Gestão Urbana**. Sao Paulo, Brazil: Simpósio Brasileiro Online de Gestão Urbana (SIBOGU), 2020. p. 189–204. Disponível em: <https://eprints.qut.edu.au/207187/>.

NETO, C. M. Avaliação das ferramentas etl open-source talend e kettle para projetos de data warehouse em empresas de pequeno porte. União Metropolitana de Educação e Cultura, Lauro Freitas, RS, Brasil, 2012.

NOTARI, D. L.; BATTISTELO, R.; MOLIN, L. W. **Aplicação web para indicadores de Cidades do Conhecimento**. [S.l.]: Revista Brasileira de Gestão e Inovação, 2019. v. 7. 96-116 p.

PENTAHO. **Open Source Business Intelligence**. 2021. Disponível em: <http://www.pentaho.com/product/data-integration>. Acesso em: 05 jun. 2021.

REDMONK. **The RedMonk Programming Language Rankings: January 2021**. 2021. Disponível em: <https://redmonk.com/sogrady/2021/03/01/language-rankings-1-21/>. Acesso em: 06 jun. 2021.

SAMU, P. H. G. **Pacote de consulta a dados censitários do censo demográfico de 2010 do IBGE utilizando linguagem Python**. [S.l.]: Universidade Federal do Rio de Janeiro, 2018.

SARIMIN, M.; YIGITCANLAR, T. Towards a comprehensive and integrated knowledgebased urban development model: status quo and directions. **International Journal of Knowledge-Based Development**, v. 3, n. 2, p. 175–192, 2012.

SILVA, L. M. M. Etl na era do big data. Universidade de Lisboa, Lisboa, 2016.

TALEND. **Open Integration Solutions**. 2021. Disponível em: <https://www.talend.com/>. Acesso em: 05 jun. 2021.

YIGITCANLAR, T. Position paper: redefining knowledge-based urban development. **International Journal of Knowledge-Based Development**, v. 2, n. 4, p. 340–356, 2011.

YIGITCANLAR, T.; O'CONNOR, K.; WESTERMAN, C. The making of knowledge cities: Melbourne's knowledge-based urban development experience. **Cities**, v. 25, n. 2, p. 63–72, 2008.

YULIANTO, A. A. Extract transform load (etl) process in distributed database academic data warehouse. **APTIKOM Journal on Computer Science and Information Technologies**, v. 1, n. 2, p. 61–68, 2019.

APÊNDICE A – DEPENDÊNCIAS DA APLICAÇÃO

Com relação as dependências este manual lista as dependências de desenvolvimento e para rodar a aplicação.

Para a implantação do sistema, é necessário uma máquina com um servidor com o banco de dados MySQL instalado, instalação do Node.js[1] versão recomendada e a instalação do Python[2] 3.

Após a instalação do Node.js e do Python, as bibliotecas e frameworks para instalar e configurar são:

- Instalação do Angular[3] CLI;

```
1 npm install -g @angular/cli
```

- Instalação da biblioteca Pandas no Python;

```
1 pip install pandas
```

- Instalação do Python Connector MYSQL;

```
1 pip install mysql-connector-python
```

- Instalação a biblioteca do python SQLAlchemy;

```
1 pip install SQLAlchemy
```

- Instalação da biblioteca Chardet.

```
1 pip install chardet
```

[1] https://nodejs.org/en/download/
[2] https://www.python.org/downloads/
[3] https://angular.io/cli

ANEXO A – LISTA DE INDICADORES

Tabela 4 – Lista de indicadores da Plataforma de Cidades do Conhecimento.

(continua)

CAPITAL	DESCRIÇÃO	FONTE
Geral	População	ibge
Geral	População	ibge
Geral	População	ibge
Geral	IFDM	FIRJAN
Identidade	Densidade democgráfica	ibge
Identidade	Despesas municipais com cultura	Siconfi
Identidade	Vínculos ativos	RAIS
Identidade	Proporção trabalhadores formais / população *100	RAIS
Identidade	Importação US$ FOB /hab	MDIC
Identidade	Exportação US$ FOB /hab	MDIC
Identidade	Saldo de empregos / vinculos*100	CAGED
Identidade	Qnt. de imigrantes que entraram no ano / 100 mil hab	SISMIGRA
Inteligência	Despesas municipais - Planejamento e Orçamento	Siconfi (R$/hab)
Inteligência	Densidade_Telefonia_Fixa	ANATEL
Inteligência	Densidade_TV_Assinatura	ANATEL
Inteligência	Densidade_Telefonia_Movel	ANATEL
Inteligência	Densidade_Banda_Larga_Fixa	ANATEL
Inteligência	WIFI PÚBLICA	ibge
Inteligência	Veículos de Imprensa	
Relacional	Despesas municipais - Desporto Comunitário	Siconfi (R$/hab)
Relacional	Despesas municipais - Lazer	Siconfi (R$/hab)
Relacional	Qnt. de imigrantes que entraram no ano / 100 mil hab	SISMIGRA
Relacional	Mortes por causas violentas / 100 mil hab	DATASUS
Relacional	Divórcios (2019)	SIDRA
Relacional	Mulheres em assentos parlamentares (vereadoras)	
Relacional	Mortes por arma de fogo / mortes *100	DATASUS
Relacional	Mulheres no mercado de trabalho	ibge
financeiro	IFDM - Emprego & Rea	FIRJAN
Financeiro	Despesas municipais - Comércio e Serviços	Siconfi (R$/hab)
Financeiro	Prod. Interno Bruto p capita	IBGE

Tabela 4 – Lista de indicadores da Plataforma de Cidades do Conhecimento.

(continuação)

CAPITAL	DESCRIÇÃO	FONTE
Financeiro	Saldo de empregos 2019 / vinculos dez 2019 * 100	CAGED
Financeiro	Grau de formalização dos ocupados	RAIS
Financeiro	Renda média dos trabalhadores % , renda acima 3 SM	RAIS
Financeiro	Poupança média por habitante	ESTBAN
Humano coletivo	IFDM - Educação	FIRJAN
Humano coletivo	Despesas municipais - Saúde	Siconfi (R$/hab)
Humano coletivo	Despesas municipais - Assistência Comunitária	Siconfi (R$/hab)
Humano coletivo	Despesas municipais - Habitação	Siconfi (R$/hab)
Humano coletivo	Despesas municipais - Saneamento Rural + Urbano	Siconfi (R$/hab)
Humano coletivo	Mortes doenças alto impacto	DATASUS
Humano coletivo	Óbitos evitáveis em menos de 5 anos	DATASUS
Humano coletivo	Óbitos evitáveis de 5 a 74 anos - Óbitos_p/Residênc	DATASUS
Humano coletivo	Imunizações BCG	DATASUS
humano individual	IFDM - Saúde	FIRJAN
Humano individual	Desp. municipais - Assist. à Criança e ao Adolescente	Siconfi (R$/hab)
Humano individual	Desp. municipais - Educação de Jovens e Adultos	Siconfi (R$/hab)
Humano individual	Desp. municipais - Educação Especial	Siconfi (R$/hab)
Humano individual	Desp. municipais - Educação Especial	Siconfi (R$/hab)
Humano individual	Mortalidade infantil (obitos / nascidos vivos)	DATASUS
Humano individual	Mortes em acicentes de transito / mortes (%)	DATASUS
Humano individual	IDEB	MEC
Humano individual	Matriculas ed basica/ população	INEP
Humano individual	Matriculas ensino superior / população	INEP
Instr. tangível	Despesas municipais - Segurança Pública	Siconfi (R$/hab)
Instr. tangível	Despesas municipais - Gestão Ambiental	Siconfi (R$/hab)
Instr. tangível	Despesas municipais - Transporte	Siconfi (R$/hab)
Instr. tangível	Esgoto ref. municípios atendidos com água (%)	SNIS
Instr. tangível	IN055_AE - Índice de atendimento total de água (%)	SNIS
Instr. tangível	Índice no município por coleta de resíduos (%)	SNIS
Instr. tangível	Número de leitos hospitalares por 1000 habitantes	DATASUS
Instr. tangível	Mortes por arma de fogo / mortes *100	DATASUS
Instr. tangível	Transporte público (qtd onibus detran)	
Instr. intangível	Despesas municipais - Legislativa	Siconfi (R$/hab)

Tabela 4 – Lista de indicadores da Plataforma de Cidades do Conhecimento.

(conclusão)

CAPITAL	DESCRIÇÃO	FONTE
Instr. intangível	Despesas municipais - Ação Legislativa	Siconfi (R$/hab)
Instr. intangível	Despesas municipais - FU04 - Administração Geral	Siconfi (R$/hab)
Instr. intangível	Despesas municipais - Administração Financeira	Siconfi (R$/hab)
Instr. intangível	Despesas municipais - Controle Interno	Siconfi (R$/hab)
Instr. intangível	Despesas municipais - Ensino Fundamental	Siconfi (R$/hab)
Instr. intangível	Despesas municipais - Ensino Médio	Siconfi (R$/hab)
Instr. intangível	Despesas municipais - Ensino Superior	Siconfi (R$/hab)
Instr. intangível	Habitantes por ong - 2016	SIDRA

Fonte: (NOTARI; BATTISTELO; MOLIN, 2019)