

**CENTRO UNIVERSITÁRIO CARIOCA**

**ENGENHARIA DA COMPUTAÇÃO**

**YURI CONCEIÇÃO DE BRITO SCARANNI**

**UMA ABORDAGEM DIDÁTICA DO ECOSSISTEMA DA INFRAESTRUTURA DE CIÊNCIA DE DADOS**

**RIO DE JANEIRO**

**2021**

**YURI CONCEIÇÃO DE BRITO SCARANNI**

**UMA ABORDAGEM DIDÁTICA DO ECOSSISTEMA DA INFRAESTRUTURA DE CIÊNCIA DE DADOS**

Trabalho de Conclusão de Curso apresentado ao Centro Universitário Carioca, como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: Prof. D. Sc. Sérgio Assunção Monteiro

RIO DE JANEIRO

2021

|  |
| --- |
| Yuri Conceição de Brito Scaranni  Uma Abordagem Didática do Ecossistema da Infraestrutura de Ciência de Dados / Yuri Conceição de Brito Scaranni. - Rio de Janeiro, 2021.  16 f.  Orientador: Sérgio Assunção Monteiro.  TCC (Graduação em Engenharia da Computação) – Centro universitário Unicarioca. Rio de Janeiro, 2021.  1. ETL. 2. Análise de dados. 3. Python. 4. Didático. 5. Bancos de dados. |

YURI CONCEIÇÃO DE BRITO SCARANNI

UMA ABORDAGEM DIDÁTICA DO ECOSSISTEMA DA INFRAESTRUTURA DE CIÊNCIA DE DADOS

Trabalho de Conclusão de Curso apresentado ao Centro Universitário Carioca, como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

BANCA EXAMINADORA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Sérgio Monteiro Assunção, D.Sc - Orientador

Centro Universitário Carioca

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. 2,

Centro Universitário Carioca

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. 3,

Centro Universitário Carioca

**AGRADECIMENTOS**

Agradecemos a todos que me apoiaram desde o início deste sonho, aos meus amigos, aos meus professores, ao meu orientador prof. Sergio Monteiro Assunção por todo suporte no desenvolver deste trabalho e em especial a minha namorada e companheira de todos os momentos Evlyn Moraes.

**RESUMO**

A cada dia que passa observamos um aumento considerável no volume de dados que somos capazes de produzir. Seu armazenamento, antes com propósito único de gerar um histórico, ganhou mais significado quando o ramo de análise de dados cresceu e pôde mostrar informações extremamente úteis na tomada de decisão que antes passavam despercebidas. No meio desse processo, entre a geração do dado e a compreensão do mesmo fica a infraestrutura responsável por organizar esses dados se utilizando de processos automatizados para isso e é neste ponto que profissionais voltados para a ciência de dados se destacam, sendo os encarregados pela montagem de toda a engenharia envolvida no processo. Nesse contexto, o presente trabalho trata de explicar como as diversas ferramentas para processamento e análise de dados se contextualizam formando o ecossistema que viabiliza desenvolver aplicações para ciência de dados. Faremos uma abordagem tanto teórica, como prática.

**Palavras chave:** ETL, Análise de dados, Python, Didático, Bancos de dados.

**ABSTRACT**

Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido. Texto traduzido.

**Keywords:** ETL, Data analysis, Python, Didactic, Databases.

**LISTA DE ILUSTRAÇÕES**

[Figura 1 – Modelo básico ETL 18](#_Toc58093865)

[Figura 2 – Linguagem baixo nível e alto nível 19](#_Toc58093866)

[Figura 3 – Instalação da biblioteca Requests 22](#_Toc58093867)

[Figura 4 – Código Python extração de dado 23](#_Toc58093868)

[Figura 5 – CSV extraído 24](#_Toc58093869)

[Figura 6 – Instalação da biblioteca Pandas 25](#_Toc58093870)

[Figura 7 – Código Python de transformação de um arquivo CSV 26](#_Toc58093871)

[Figura 8 – WAMP em execução 27](#_Toc58093872)

[Figura 9 – Página *localhost* onde *WAMPserver* é executado 28](#_Toc58093873)

[Figura 10 – Página de *login PhpMyAdmin* 29](#_Toc58093874)

[Figura 11 – Editor de texto SQL do *PhpMyAdmin* 30](#_Toc58093875)

[Figura 12 – Comando de criação tabela temporária 31](#_Toc58093876)

[Figura 13 – Comando de criação tabela definitiva 31](#_Toc58093877)

[Figura 14 – Instalação de biblioteca *SQLAlchemy* 32](#_Toc58093878)

[Figura 15 – Instalação da biblioteca auxiliar de conexão *PyMySQL* 32](#_Toc58093879)

[Figura 16 – Código de carga de dados no banco *MySQL* 33](#_Toc58093880)

[Figura 17 – Retorno da consulta no banco de dados 33](#_Toc58093881)

SUMÁRIO

[1 INTRODUÇÃO 16](#_Toc58367016)

[**2 OBJETIVO DO TRABALHO 1**](#_Toc58367019)**7**

[**3 TECNOLOGIAS**](#_Toc58367043) **17**

[3.1 LINGUAGEM DE PROGAMAÇÃO 17](#_Toc58367044)

[3.2 PYTHON 18](#_Toc58367045)

[3.2.1 Pandas 19](#_Toc58367046)

3.3 Banco de dados 19

3.3.1 Sql 19

3.3.2 WAMP 20

[4 METODOLOGIA 21](#_Toc58367056)

[5 PROBLEMA 21](#_Toc58367078)

[5 CONCLUSÃO 29](#_Toc58367078)

[5 REFERÊNCIAS BIBLIOGRÁFICAS](#_Toc58367078) 30

**1 Introdução**

As análises de dados que suportem decisões de negócios têm se tornado algo comum em qualquer grande empresa, uma origem de dados contém milhares de informações, elas são persistidas em um banco de dados e a partir desses vários profissionais realizam análises que possibilitam insights e melhor clareza sobre como está a empresa hoje, onde está falhando e como é possível melhorar.

Com essa explicação superficial é possível ter uma dimensão do quão grande é esse universo relacionado a dados, envolve diversos setores, profissionais, ferramentas e áreas do conhecimento. Mas como isso acontece efetivamente? Os dados não saem de uma origem para um destino (e posterior análise) com mágica, isso ocorre através um processo chamado ETL, que é quase mágica.

O ETL, extract – transform – load (em português extração – transformação – carga), nada mais é do que o processo que captura/extrai os dados em qualquer fonte que seja, trata as imperfeições e adequações necessárias e então realiza a carga desses dados tratados em um destino que possibilite a fácil visualização dos dados.

O ecossistema da infraestrutura para ciência de dados cobre uma gama de formatos e origens, como podemos ver na figura 1.

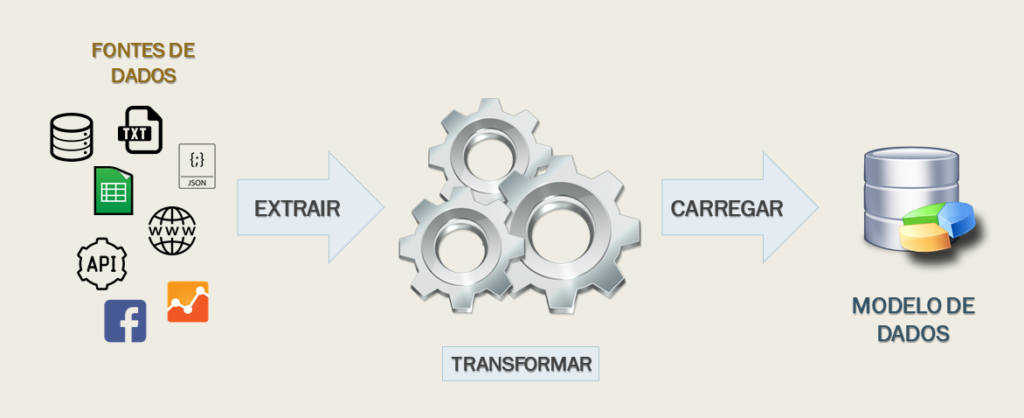


Figura 1: Modelo básico ETL. Fonte: https://uaismart.com/o-que-e-etl/.

**2 Objetivo do trabalho**

Este trabalho tem como objetivo explicar o funcionamento de todo o ecossistema que envolve a análise, persistência e obtenção de um dado, dando ênfase no processo de ETL (Extract-Transform-Load) de um dado. Optando por uma visão mais didática/tutorial para compreensão não só do motivo para se fazer um ETL como também mostrar os primeiros passos para se montar um.

**3 Tecnologias**

**3.1 Linguagem de programação (LP)**

Uma linguagem de programação é um vocabulário e um conjunto de regras gramaticais usadas para escrever programas de computador. Esses programas instruem o computador a realizar determinadas tarefas específicas. Cada linguagem possui um conjunto único de palavras-chaves (palavras que ela reconhece) e uma sintaxe (regras) específica para organizar as instruções dos programas. Em outras palavras, a linguagem de programação (LP) é a forma que encontramos de nos comunicar com o computador, isso através de diversas instruções, essas instruções são passadas através de compiladores que transcrevem o código fonte para comandos de máquina (binário).

Uma LP pode ser classificada como **Alto Nível** ou **Baixo Nível**, LPs de alto nível são aquelas mais amigáveis para seres humanos como Python, Java, JavaScript, C#, Ruby etc. LP de alto nível se aproximam muito da forma com que falamos e escrevemos, quase sempre em inglês, e isso as torna universais e de melhor compreensão. Já as LPs de baixo nível se aproximam da forma como a máquina compreende, a linguagem mais conhecida desse nicho é a Assembly. Seus comandos são mais complexos para seres humanos, porém, bem mais fáceis da máquina entender, o que acelera o processamento de grande volume de dados.

A figura a seguir ilustra a diferença de sintaxe entre um comando alto e baixo nível (comando ‘imprimir na tela’):



Figura 2: Linguagem baixo nível e alto nível. Fonte: https://becode.com.br/linguagens-alto-nivel-x-baixo-nivel/.

**3.2 Python**

O Python é uma linguagem de programação de alto nível criada em 1991 por Guido van Rossum, simples e objetiva a linguagem se popularizou nos últimos anos por isso além da sua imensa flexibilidade, disponibilidade de bibliotecas e grande comunidade. Em Python é multiparadigma, seja ele funcional, orientado a objetos, estruturada, possui tipagem dinâmica e é open source.

Dentre as várias linguagens disponíveis Python se destaca no ramo da ciência de dados por ser robusto para programação de forma geral e possuir muitas bibliotecas de apoio para análise, tratamento e visualização de dados (ex.: Numpy, Pandas, Matplotlib e Scipy).

Com Python é possível fazer desde a extração, tratamento, até a inserção dos dados em um banco de dados, tornando uma linguagem extremamente poderosa.

**3.2.1 Pandas**

O Pandas é um dos pacotes mais utilizados no que diz respeito ao tratamento de dados, ele é um pacote Python que fornece estruturas de dados rápidas, flexíveis e expressivas projetadas para tornar o trabalho com dados estruturados (tabulares, multidimensionais, potencialmente heterogêneos) e de séries temporais fácil e intuitivo. O principal objetivo do pacote é ser o bloco de construção fundamental de alto nível para fazer análises de dados práticas e do mundo real em Python. Além disso, tem o objetivo mais amplo de se tornar a ferramenta de análise / manipulação de dados de código aberto mais poderosa e flexível disponível em qualquer idioma.O Pandas, portanto, colabora muito com milhares de métodos e funções para tratar e estruturar dados, facilitando a vida do programador.

**3.3 Bancos de dados**

Um banco de dados é uma coleção organizada de informações - ou dados - estruturados, geralmente alocados em um sistema de computador. Um banco de dados é geralmente controlado por um sistema de gerenciamento de banco de dados (DBMS). Juntos, os dados e o DBMS, juntamente com os aplicativos associados a eles, são chamados de sistema de banco de dados, geralmente abreviados para apenas banco de dados. A partir dessa definição fica mais simples entender que o banco de dados é, o local onde armazenamos todo tipo de dado. Ele pode ser armazenado dentro de tabelas, como em uma planilha e consultado a qualquer momento, de qualquer lugar que consiga estabelecer uma conexão com o servidor no qual o banco de dados está. Dentre os sistemas gerenciais de bancos de dados (SGBD) mais famosos estão a própria Oracle, MySQL, SQL Server entre outros.

**3.3.1** **SQL**

Structured Query Language, é uma linguagem de programação em bancos de dados padronizada e universal, já que serve para todos os SGBD, com pequenas variações de termos ou sintaxe entre eles, ou seja, tem alguns comandos que são entendidos apenas pelo MySQL, outros apenas pelo Oracle, mas no geral isso não é uma dificuldade já que esses comandos exclusivos são bem esporádicos, os comandos universais são usados com muito mais frequência. SQL é uma linguagem utilizada para consultar (essas consultas são chamadas de Query) e visualizar dados dentro do banco de dados, além de alterar estrutura de tabelas, criar bases de dados, inserir e atualizar valores, excluir tabelas e registros, em resumo, SQL é a linguagem que se comunica com o banco de dados executando todos os comandos dentro dele.

**3.3.2 WAMP**

WAMP, também conhecido como WAMP server, é um acrônimo para Windows Apache MySQL Php (Para sistemas Linux é conhecido como LAMP – Linux Apache MySQL Php). Ele é um ambiente de desenvolvimento Web do Windows. Ele permite que você crie aplicativos web com Apache2, PHP e um banco de dados MySQL. Paralelamente, PhpMyAdmin permite que você gerencie facilmente seu banco de dados. Com WAMP se torna dispensável um servidor para abrigar o nosso banco de dados, desta forma, facilitando testes e desenvolvimento. O WAMP cria um servidor local, conhecido por localhost, que permite que ao utilizar um banco de dados, no caso MySQL, através do PhpMyAdmin seja possível fazer query’s e alterações no banco de dados, tudo isso via browser, na url ‘localhost’.

**3.4 ETL – Extract, Transform, Load**

Extrair (E) – Como o próprio nome indica, a parte de extração de um ETL diz respeito a obtenção do dado em si. Várias ferramentas podem estar envolvidas variando de acordo com cada processo. A forma mais comum de se ver uma extração é através de uma linguagem de programação que se conecta com uma origem ou simplesmente faz o download dessa origem, dentre as linguagens mais conhecidas que realizam essa operação o Python é de longe a mais consolidada no processo, muito devido a sua facilidade em lidar com dados e seu tratamento além da imensa gama de bibliotecas que ele possui.

Transformar (T) – A transformação nesse caso abrange toda a etapa de limpeza do dado, uma vez que o mesmo pode estar em uma codificação ruim para leitura, com mais colunas do que as necessárias, não traduzido, etc. A tecnologia responsável pela transformação ainda é a linguagem de programação, geralmente também Python. Ao fim desta etapa é esperado o arquivo em um formato pronto para leitura imediata (xlsx, txt) ou inserção em uma base de dados (csv).

Carregar (L) – Nesta etapa, de carga, o dado extraído já está pronto para ser repassado ao seu destino para então ser consultado, a linguagem de programação continua atuando ao abrir conexão com bancos de dados para então fazer a inserção dos dados no banco, ou agindo para fazer upload do arquivo em alguma plataforma que disponibilize a leitura/consulta.

A critério de curiosidade, já existem ferramentas que fazem todo o processo de ETL automaticamente alguns exemplos são: IBM InfoSphere DataStage , Informática Power Center, SAP BusinessObjects Data Services ,Microsoft SQL Server Integration Services (SSIS), Pentaho Data Integration, Oracle Data Integrator (ODI), entre outros, mas neste trabalho faremos todo o processo de forma pura, utilizando apenas Python.

**4 Metodologia**

A Infraestrutura que está envolvida na ciência de dados é imensa e por vezes confusa, mas por ter um peso tão grande na tomada de decisão de muitas empresas é algo extremamente necessário na era da informação. Neste capítulo vamos mergulhar na parte prática envolvida neste processo, utilizando dados públicos disponíveis (conhecidos como open data) sobre o COVID-19 advindos do portal Our World in Data.

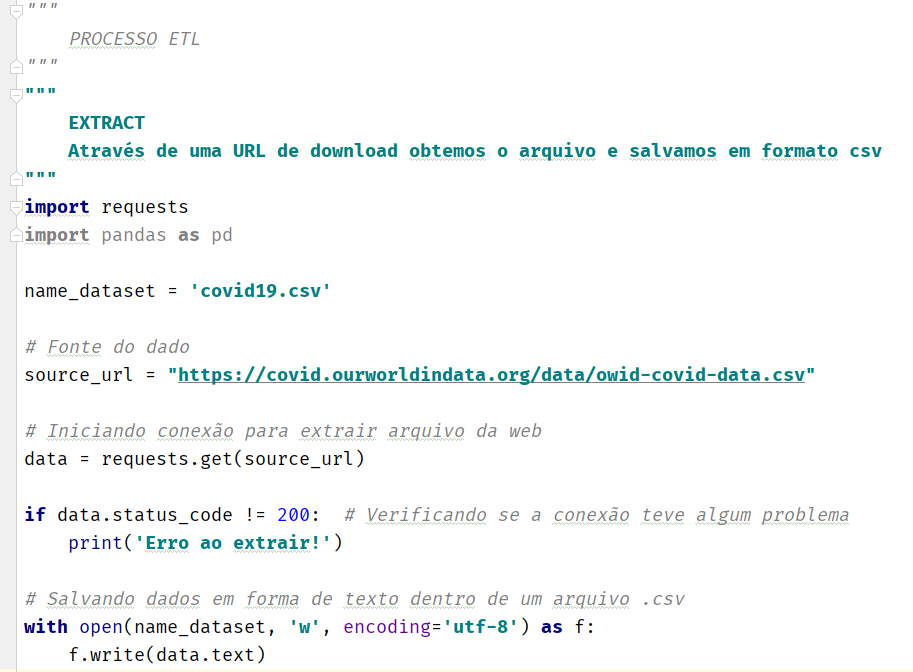
**5 Problema**

Extrair dados referentes ao COVID-19 e armazená-los em uma tabela dentro do banco de dados MySQL. Garantir que somente dados novos sejam inseridos, sem duplicidade e criar um agendamento para que essa extração aconteça todos os dias, mantendo assim os dados sempre atualizados.

Passo 1: Precisamos instalar a biblioteca para obtenção do arquivo, ***requests*** (figura 3).

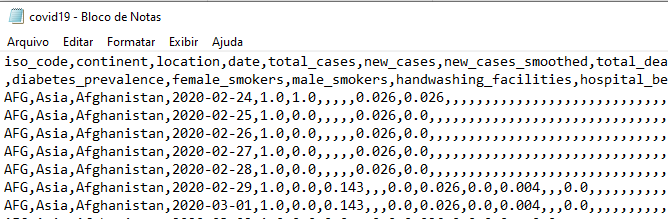


Figura 3: instalação da biblioteca Requests.

Então com o trecho de código a seguir, realizamos o download do arquivo que contém os dados (figura 4):

*Figura 4: Código Python extração do dado.*

Os dados então são salvos no arquivo ‘covid19.csv’ (figura 5).



*Figura 5: CSV extraído.*

Passo 2: Instalação da biblioteca de tratamento de dados, ***pandas*** (figura 6).



Figura 6: instalação da biblioteca Pandas.

Com as bibliotecas de transformação instaladas podemos realizar a transformação (figura 7)

1. Selecionando da coluna ‘location’ somente dados do Brasil e realizando a tradução do nome ‘Brazil’ para a nossa escrita.
2. Filtrando somente as colunas principais.
3. Traduzindo o nome das colunas, as renomeando e salvando logo em seguida.



Figura 7: código python de transformação do arquivo csv.

Passo 3: Nesta etapa fazemos a inserção dos dados em uma base de dados, a base escolhida foi o MySQL. Por se tratar de uma implementação de desenvolvimento/teste, utilizaremos o WAMP server para que o mesmo crie um servidor local e possamos utilizar o MySQL localmente, para isso fazemos a instalação do WAMP baixado no site: <https://www.wampserver.com/en/>.

Após download e instalação, executamos o WAMP, onde ele iniciará os serviços para que o MySQL funcione automaticamente, após inicialização da aplicação um *tray icon* na cor verde (figura 8) indicará que está tudo funcionando corretamente.



Figura 8: WAMP em execução

Com WAMP nós temos um banco de dados (MySQL) e um visualizador de dados (PhpMyAdmin) disponível. Então iremos até o visualizador de dados para criar as duas tabelas que precisaremos, uma temporária e a tabela que guardará os dados permanentes.

1. Abra o navegador de preferência.
2. Acesse na URL: localhost.
3. Na parte inferior, ao centro, em “Your Aliases” clique em “phpmyadmin” (figura 9).

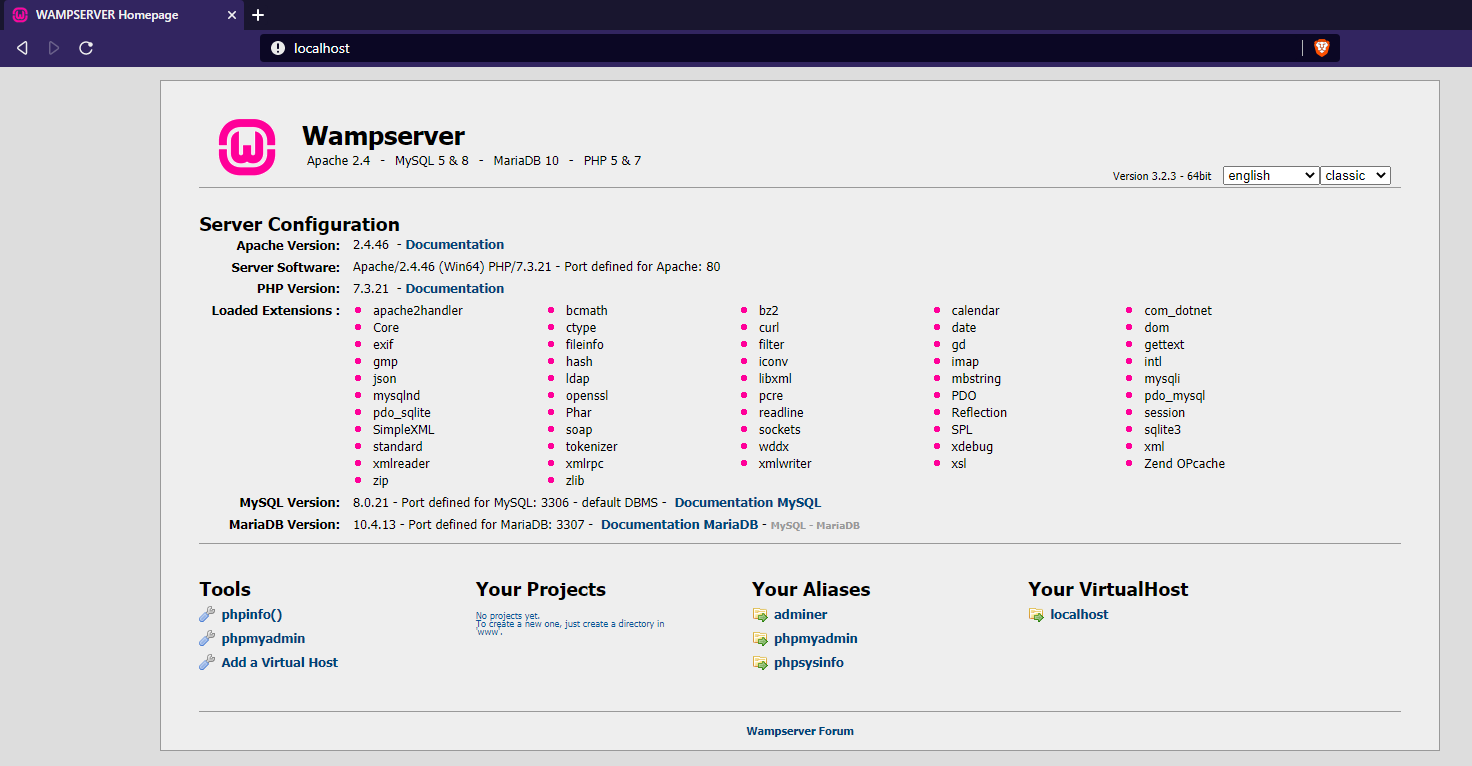


Figura 9: página localhost onde wampserver é executado.

1. O utilizador é: root e o campo senha fica em branco (figura 10).

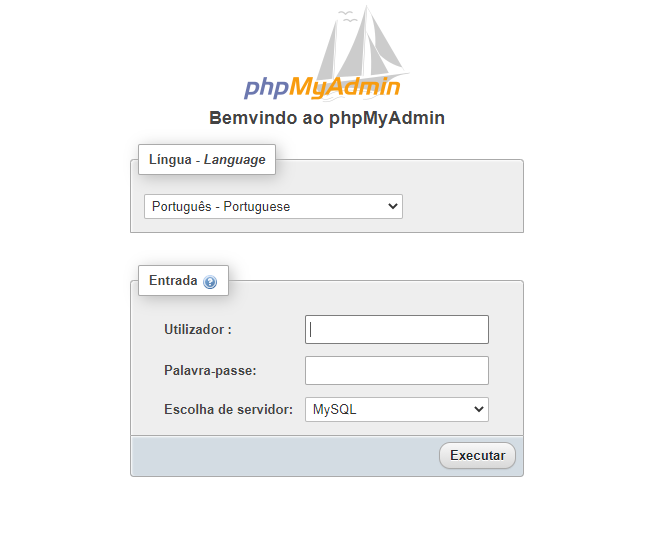


Figura 10: página de login PhpMyAdmin.

1. Clicando em SQL, abrimos o editor de texto SQL (figura 11) do visualizador, aqui criaremos a tabela temporária e a tabela permanente.

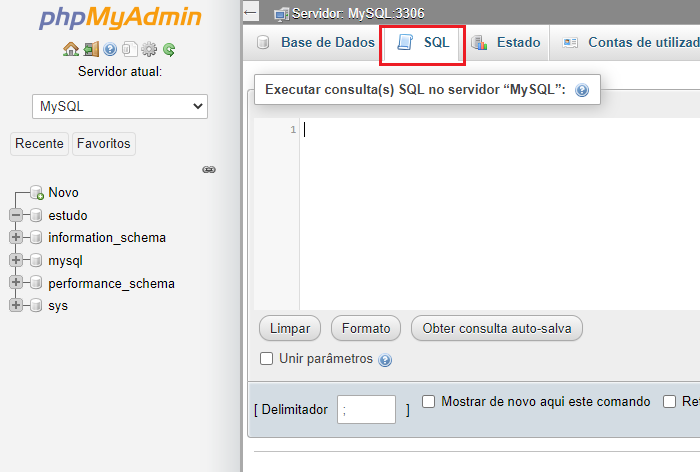


Figura 11: Editor de texto SQL do PhpMyAdmin.

1. Criamos a tabela temporária (figura 12).

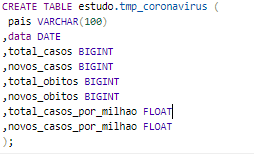


Figura 12: Comando de criação tabela temporária.

1. E em seguida a tabela permanente (figura 13), que diferentemente da tabela temporária, possui um campo chave, data, o que significa que não teremos nesta tabela duas ou mais registros com a mesma data, o que garante uma análise correta dos nossos dados.

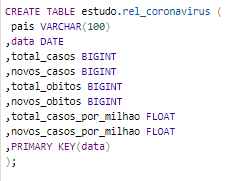


Figura 13: Comando de criação tabela definitiva.

1. Em seguida fazemos a instalação da biblioteca responsável pela conexão do banco de dados no Python, a SQLAlchemy (figura 14).



Figura 14: instalação da biblioteca SQLAlchemy.

1. Como parte das dependências da biblioteca SQLAlchemy, precisamos também da biblioteca pymysql (figura 15).



Figura 15: instalação da biblioteca auxiliar de conexão PyMySQL.

Agora podemos executar o script de persistência dos dados (figura 16), primeiro abrindo conexão com o banco de dados, depois limpando todos os dados da tabela temporária, inserindo todos os dados do nosso arquivo nela e por último selecionando somente os dados que ainda não existem na nossa tabela permanente para serem inseridos.



Figura 16: Código de carga de dados no banco MySQL.

Passo 4: Com os dados já dentro do banco, podemos visualizá-los com o auxílio do WAMP. Já logado, é possível enxergar os dados no schema Estudo e nas tabelas rel\_coronavirus e tmp\_coronavirus (figura 17).

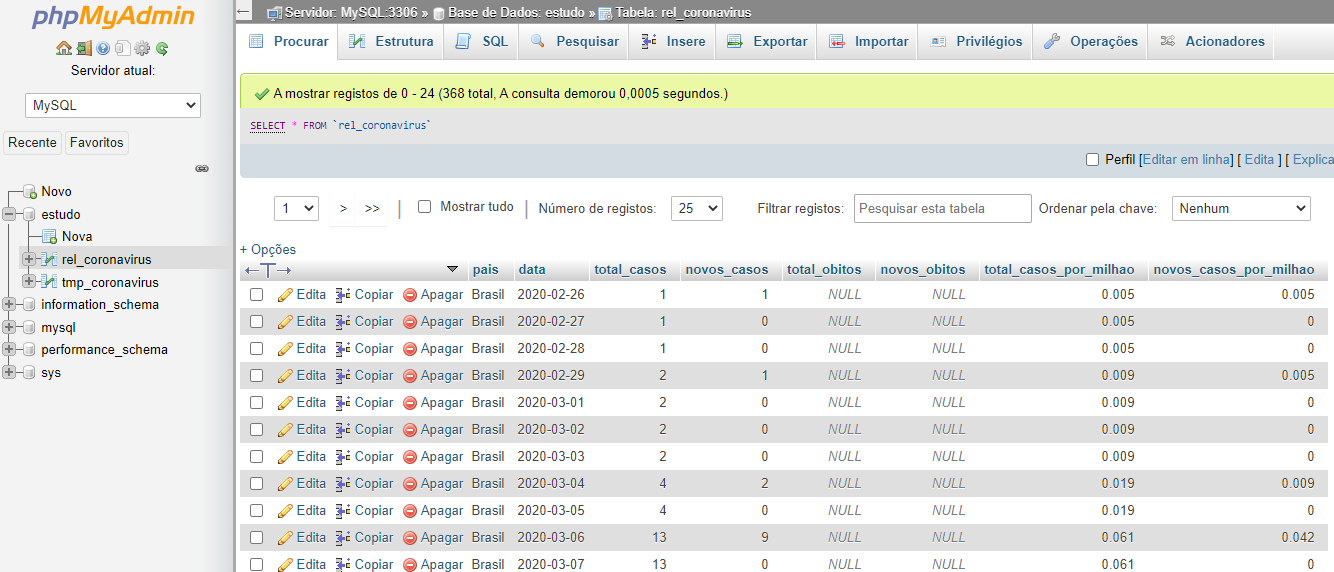


Figura 17: Retorno da consulta no banco de dados.

**6 Conclusão**

Conforme observamos ao longo deste trabalho, o processo de ETL é uma parte gigantesca da infraestrutura de ciência de dados. Embora extensa, através do Python e suas bibliotecas é possível encurtar caminhos e realizar o processo de persistência de dados para análise de praticamente todas as fontes de dados. Embora o processo aqui mostrado seja o mais consolidado, existem milhares de outras formas de se realizar um ETL, desde outras bibliotecas de extração, desde o uso de laços de repetição para o tratamento de dados, dispensando o pacote pandas, além do que, existem também muitos outros bancos de dados para inserção e clientes de bancos de dados para sua visualização.

**7 Referências Bibliográficas**

[1] BECODE. **Linguagens de Alto Nível vs. Baixo Nível: qual é melhor?**. Disponível em: https://becode.com.br/linguagens-alto-nivel-x-baixo-nivel/. Acesso em: 17 fev. 2021.

[2] ORACLE. **O que é um banco de dados?**. Disponível em: https://www.oracle.com/br/database/what-is-database/. Acesso em: 17 fev. 2021.

[3] PYPI. **Pandas 1.2.2**. Disponível em: https://pypi.org/project/pandas/#:~:text=pandas%20is%20a%20Python%20package,data%20both%20easy%20and%20intuitive.&text=Tabular%20data%20with%20heterogeneously%2Dtyped,SQL%20table%20or%20Excel%20spreadsheet. Acesso em: 17 fev. 2021.

[4] PYSCIENCE-BRASIL. **Python: O que é? Por que usar?**. Disponível em: http://pyscience-brasil.wikidot.com/python:python-oq-e-pq. Acesso em: 17 fev. 2021.

[5] UAI SMART. **O QUE É ETL E SUA IMPORTÂNCIA PARA O BUSINESS INTELLIGENCE**. Disponível em: https://uaismart.com/o-que-e-etl/. Acesso em: 17 fev. 2021.