The background is a deep blue gradient. It features several stylized, glowing virus particles. On the left, a large, irregular cluster of cells or viral components is visible. Scattered throughout the frame are several spherical virus particles, each covered in small, triangular protrusions, resembling coronaviruses. The overall aesthetic is scientific and digital.

Engenharia de dados aplicado à COVID 19

01.

DESENVOLVIMENTOS

Explicação e Funcionamento do back-end, Databricks e Front-end.

02.

EVOLUÇÃO DA SOLUÇÃO

Melhorias possíveis e lições aprendidas.

DESENVOLVIMENTOS



Back-End

Requisição de informações via API, verificação da informação e inserção no banco de dados



Databricks

Extração da informação do banco de dados, transformação, testes de hipóteses e insights

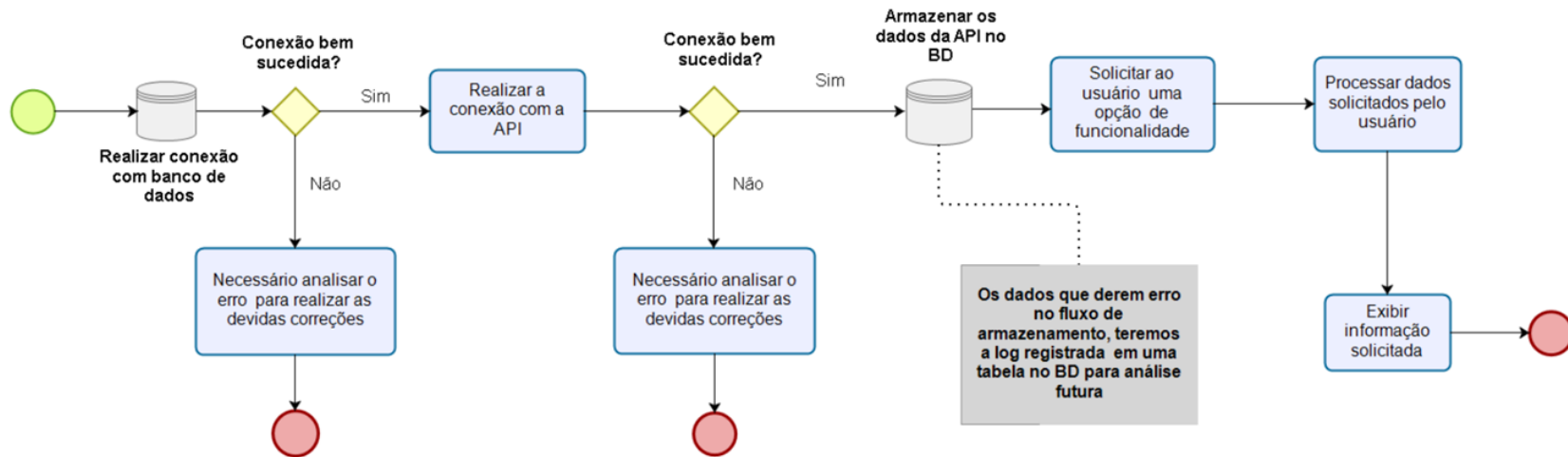


Power BI

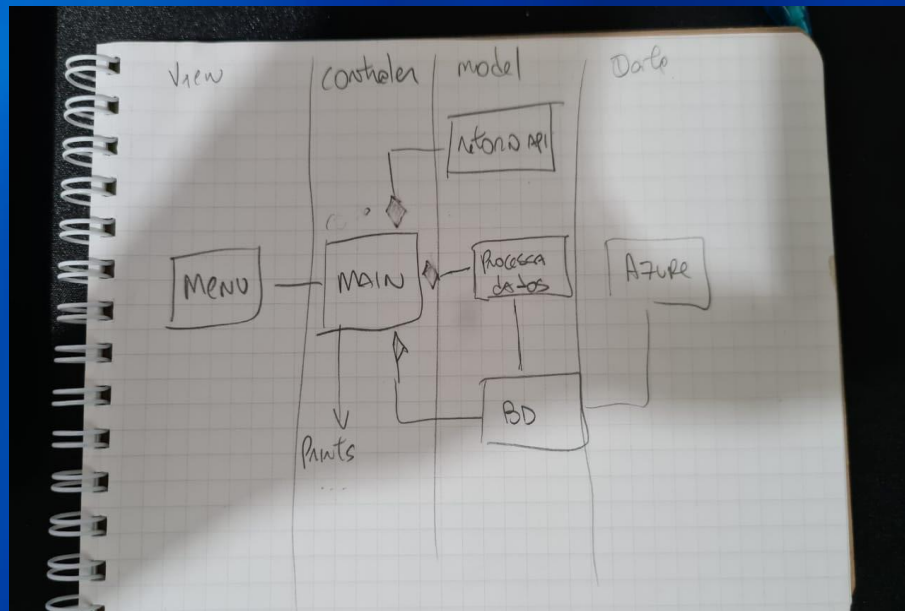
Dashboards com informações e indicadores dos casos de Covid no mundo/Brasil.

ARQUITETURA BACK-END

Funcionalidade software



ESTRUTURA DO SOFTWARE - BACK-END



FUNCIONALIDADE FRONT-END

Divisão dos Notebooks

Ingestão

Realizar a ingestão dos datasets que estão no banco SQL na Azure em um diretório de arquivos raw dentro do DBFS.

Configuração para acesso ao banco de dados

OBS: Não esquecer de liberar o IP do Databricks nas configurações do banco no Azure.

```
Cmd 3
jdbcHostname = "datazilla.database.windows.net"
jdbcDatabase = "datazilla"
jdbcPort = 1433
jdbcUrl = "jdbc:sqlserver://{0}:{1};database={2}".format(jdbcHostname, jdbcPort, jdbcDatabase)
connectionProperties = {
    "user": 'datazilla',
    "password": 'gama123456@$',
    "driver": "com.microsoft.sqlserver.jdbc.SQL"
}
```

Transformações

Realizar transformações nos datasets, de modo a converter o dado ingestado previamente no formato mais otimizado para Big fisicamente quando necessário.

Transformação das tabelas de formato JSON para Parquet

Definindo um schema para cada tabela

```
Cmd 4
from pyspark.sql.types import StructType, StructField, StringType, IntegerType, DateType
```

Carregamento dos dados em parquet

```
Cmd 2
df_data_by_country = spark.read.parquet("dbfs:/FileStore/_covid_data_lake/_ready/_data_by_country/data_by_country.parquet")
df_summary_country = spark.read.parquet("dbfs:/FileStore/_covid_data_lake/_ready/_summary/summary_country.parquet")
```

Análise exploratória dos dados

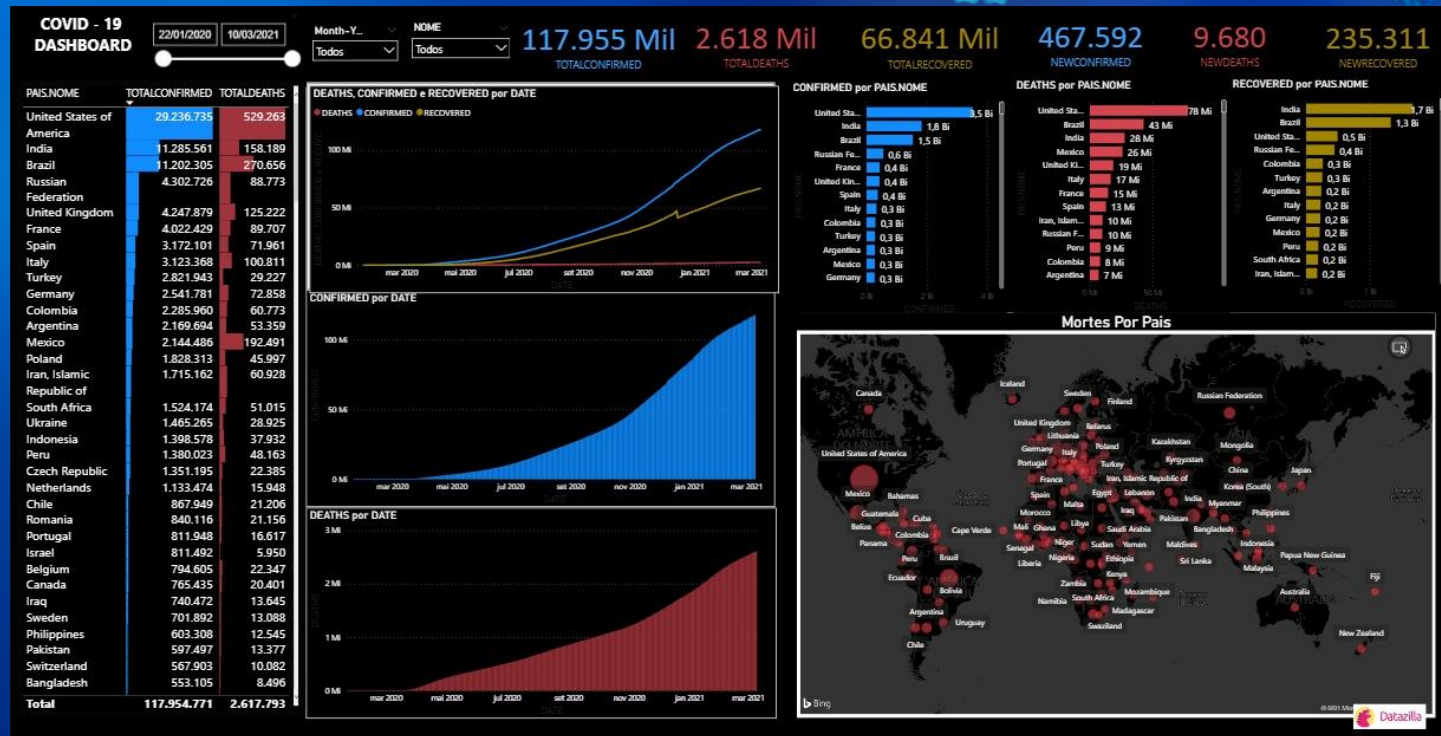
Aqui realizaremos uma série de queries sobre os dados

Comparação do número de casos do Brasil com outros três 'epicentros' da COVID-19

```
Cmd 5
import pyspark.sql.functions as f
```

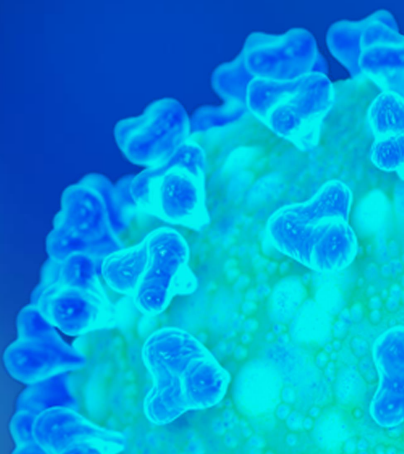
FUNCIONALIDADE FRONT-END

Conexão com PowerBI



DIFICULDADES E LIÇÕES APRENDIDAS



- API Instável – Foi necessário criar caches em disco e trabalhar extensivamente com exceções
 - Validar os tipos criando Schema de dados no DataFrame
 - Processar os dados antes de enviar verificando duplicidade através de Timestamp e Nomes dos Países, criando o relacionamento de chaves
 - Conexão de internet Instável forçou inserir linha a linha, fazendo commits depois de um número de inserts (ao invés de se utilizar funções otimizadas para muitos inserts)
 - Maior controle para o usuário, fragmentando as tarefas e possibilitando ao usuário fazê-las de acordo com a necessidade
 - Foco na Qualidade, Desenvolvimento para “outra” equipe, Boas Práticas de Design
- 



OBRIGADO!

GRUPO 2 - Datazilla
George Razera
Tiago Fernandes DAgostino
Viviane Jordão Nyitray
Judson

CREDITS: This presentation template was created by Slidesgo,
including icons by Flaticon, and infographics & images by Freepik