Introdução a Big Data

Neste trabalho final da disciplina de introdução a Big Data, será apresentado os seguintes tópicos, conforme orientação acadêmica:

1. Descrição das opções listadas: Apaches Hudi, Iceberg e DeltaLake;
2. Comparação de características entre as opções anteriores;
3. Projeto de um data lake para atender a um caso de uso (fictício ou não).
4. **Descrição das opções listadas:**
   1. **Apache Hudi**: é uma plataforma open source desenvolvida pela Uber. Visa simplificar e otimizar a ingestão, gerenciamento e transformação de grandes volumes de dados em sistemas distribuídos. “Hudi” significa “Hadoop Upserts Deletes and Incrementals” e reflete sua capacidade única de suportar operações de atualização incremental em conjuntos de dados distribuídos. Site: <https://hudi.apache.org/>

**Principais características:**

* Suporte a atualizações incrementais: uma das propriedades exclusivas do Apache Hudi é a sua capacidade de efetivamente suportar atualizações incrementais nas operações de grandes conjuntos de dados dispersos. Isso significa que o usuário pode efetuar inserções, exclusões e atualizações nos dados previamente existentes sem precisar recalcular os conjuntos de dados dispersos completos.
* Suporte ACID: capaz de garantir atomicidade, consistência, isolamento e durabilidade (ACID) para operações de escrita, o Hudi assegura a integridade e consistência dos dados em ambientes distribuídos e concorrentes.
* Controle de particionamento e pesquisa de passado: com recursos avançados de controle de particionamento, ele facilita a organização e recuperação eficiente de dados. Além disso, é possível realizar pesquisas definindo uma data passada, como um snapshot dos dados em determinada data.
* Operações de atualização e exclusão: O Hudi permite realizar operações de atualização (upserts) e exclusão (deletes) de dados de forma eficiente e escalável. Essa capacidade é especialmente útil em cenários onde correções ou remoções nos dados são necessárias após a ingestão inicial. Upsert seria a junção de Update + Insert onde caso exista o dado, faça update, caso contrário, insira.
* Compatibilidade com diferentes formatos de arquivos: é compatível com uma variedade de formatos de arquivos, como Parquet, HFile, Avro e ORC, amplamente utilizados em ambientes de Data Lake para armazenamento de dados estruturados.
* Integração com Hadoop: como parte integrante do ecossistema Hadoop, integra-se com outras ferramentas e frameworks como Apache Spark, Hive e HDFS. Simplificando a construção de pipelines de dados completos e escaláveis.
* Gerenciamento robusto de metadados e índices: possui um sistema robusto de gerenciamento de metadados, que registra informações sobre operações realizadas nos dados, versões de dados e outras informações relevantes. Também suporta índices, visando melhorar o desempenho das consultas.
* Serviços de otimização: oferece vários serviços para melhorar a performance, como definição de tamanho de arquivo automático, clusterização e compactação de dados, limpeza automática de arquivos antigos.
  1. **Apache Iceberg:** é um projeto de open source, desenvolvido pela Netflix e projetado para simplificar o gerenciamento de tabelas de dados em ambientes de Data Lakes distribuídos. Resumidamente, consiste em um formato de tabela de dados distribuídos, específico para grandes volumes de dados. Site: <https://iceberg.apache.org/>

**Principais características:**

* + Versionamento: Permite o controle das versões das tabelas de dados, possibilitando a visualização das alterações ao longo do tempo e a recuperação de versões anteriores dos dados. Serve para rastrear e analisar mudanças históricas nos dados.
  + Compactação e consistência de dados: reduz o espaço de armazenamento necessário, especialmente útil para grandes conjuntos de dados, além de promover a consistência dos dados.
  + Suporta diferentes formatos de arquivos: Aceita vários formatos de arquivos, incluindo Parquet e Avro, amplamente utilizados em ambientes de Data Lake para armazenamento de dados estruturados.
  + Esquema evolutivo: permitindo que os esquemas(estruturas) sejam modificados sem a necessidade de reescrever os dados existentes. Proporciona flexibilidade e facilidade na evolução dos modelos de dados.
  + Otimizações de desempenho: Implementa várias otimizações para consultas analíticas, incluindo pruning(poda) de dados não necessários e compressão de dados, resultando em consultas mais eficientes e tempos de resposta mais rápidos.
  + Integração com frameworks de processamento: Integra-se com o Apache Spark, Hive e Presto, permitindo uso do Iceberg em diversos ambientes de processamento de dados. Alguns serviços da AWS também suportam nativamente o Iceberg, como AWS Athena, EMR e Glue.
  + Compatibilidade com diversos sistemas de armazenamento: compatível com uma variedade de sistemas de armazenamento, como HDFS e serviços de armazenamento em nuvem, como AWS S3 e Azure Data Lake Storage proporcionando a flexibilidade na escolha do ambiente de armazenamento.
  1. **Delta Lake:** é uma camada de armazenamento e formato padrão open source para todas as operações no Databricks, oferecendo uma extensão dos arquivos Parquet, gerando um log de transações ACID. Oferecendo recursos avançados às APIs do Apache Spark, o Delta Lake garante transações seguras, qualidade de dados e processamento em tempo real.

**Principais características:**

* Processamento em tempo real: além do processamento em lote, suporta processamento em tempo real, permitindo consultas interativas e análises de streaming sobre os mesmos dados. Muito importante para análises em tempo real em ambientes que exigem decisões rápidas.
* Esquema evolutivo e invariante: permite a evolução do esquema de dados de forma invariável, ou seja, os esquemas podem ser alterados sem reescrever os dados existentes. Isso facilita a adaptação dos modelos de dados conforme as necessidades do negócio evoluem e se ajustam.
* Controle transacional ACID: garante a integridade dos dados durante inserções, atualizações e exclusões. Isso muito importante para ambientes sensíveis à consistência, como em setores financeiros e ou ambientes transacionais.
* Suporte a diversos formatos de arquivos: assim como o Apache Spark, suporta uma variedade de formatos de arquivos, incluindo Parquet e Delta, oferecendo flexibilidade na escolha do formato mais adequado para os dados.
* Otimizações de desempenho: Implementa várias otimizações, como compactação de arquivos e pruning de partições desnecessárias para melhorar a performance das consultas e reduzir o tempo de processamento.
* Integração com ferramentas da databricks: integrado nativamente ao ambiente Databricks, sua implementação e gerenciamento dentro do ecossistema é simplificado. Também se integra facilmente com outras ferramentas, como Apache Spark, Apache Hadoop e Apache Airflow.

1. **Comparação de características entre as opções (Apache Hudi, Apache Iceberg e Databricks Delta Lake).**

Tabela comparativa criada com base nas informações apresentadas anteriormente:

|  |  |  |  |
| --- | --- | --- | --- |
| **Característica** | **Apache Hudi** | **Apache Iceberg** | **Delta Lake** |
| **Modelo de Dados** | Baseado em registros versionados (Parquet) | Baseado em tabelas particionadas | Baseado em arquivos versionados e unificados – Uniform\* |
| **Formatos de Arquivo** | Principalmente Parquet e Avro | Parquet, Avro, ORC, entre outros | Principalmente Parquet |
| **Gerenciamento de Metadados** | Automático para versionamento e rastreamento | Eficiente para snapshot e controle de versão | Automático para controle de versão |
| **Transações** | Suporta transações ACID para garantir a integridade dos dados durante operações de gravação e exclusão | Suporta transações ACID para garantir a integridade dos dados durante operações de gravação e exclusão | Suporta transações ACID para garantir a integridade dos dados durante operações de gravação e exclusão |
| **Evolução de Esquema** | Oferece evolução de esquema flexível, garantindo compatibilidade com versões anteriores | Oferece evolução de esquema flexível, garantindo compatibilidade com versões anteriores | Oferece evolução de esquema flexível, garantindo compatibilidade com versões anteriores |
| **Integração com Hadoop** | Integra-se facilmente com o Hadoop e é amplamente utilizado em plataformas como Apache Spark e Apache Hive | Também se integra bem com o ecossistema Hadoop e é suportado por várias ferramentas, incluindo Spark, Hive e Presto | Integração nativa com o ecossistema Databricks e suporte total com o Apache Spark |
| **Escalabilidade** | Projetado para lidar com cargas de trabalho de data lake em grande escala, oferecendo alta escalabilidade e desempenho | Altamente escalável e trabalha com grandes volumes de dados distribuídos em várias camadas de armazenamento | Altamente escalável, adaptando-se bem a cargas de trabalho de qualquer tamanho |
| **Comunidade e Maturidade** | Suportado por uma comunidade ativa, mas a maturidade pode variar dependendo dos casos de uso específicos e das necessidades da empresa | Também suportado por uma comunidade ativa, mas a maturidade pode variar dependendo dos casos de uso específicos e das necessidades da empresa | Suportado pela Databricks, com atualizações regulares e ampla adoção na indústria |
| **Particionamento** | Suporte a particionamento para melhorar a eficiência de consultas e operações de ETL | Oferece suporte a particionamento para organização eficiente de dados em várias dimensões | Suporta particionamento para otimizar operações de leitura e escrita |
| **Indexação** | Indexação global e local para melhorar o desempenho de consultas | Índices para otimização de leitura, permitindo consultas eficientes em grandes conjuntos de dados | Indexação automática para melhorar o desempenho de consultas e operações de mesclagem |

\* <https://www.databricks.com/blog/announcing-delta-lake-30-new-universal-format-and-liquid-clustering>

Comparando as principais características individualmente:

**Apache Hudi versus Apache Iceberg versus Delta Lake:**

Todos são open source, porém com focos diferentes. O Hudi possui como base uma capacidade de gestão incremental e atualizações eficientes, enquanto o Iceberg possui recursos avançados como versionamento de arquivos e esquema evolutivo. Já o Delta Lake se destaca pelas operações ACID e consultas em real time. Tanto o Delta Lake como o Iceberg possuem recursos como controle transacional e suporte a formatos populares, porém o Delta Lake é integrado ao Spark da Apache oferecendo assim consultas de batch e streaming, sendo um forte candidato para processamento e análise de dados em real time.   
 A escolha de cada deve ser ponderada visando as características e prioridades de cada projeto, por exemplo a garantia da consistência dos dados durante as transações e atualizações, a complexidade analítica das consultas ou então a velocidade de retorno das consultas.  
 Site para consulta mais aprofundada sobre a comparação entre os 3: <https://www.onehouse.ai/blog/apache-hudi-vs-delta-lake-vs-apache-iceberg-lakehouse-feature-comparison>

1. Projeto de um data lake para atender a um caso de uso.

Atualmente na visão de negócios, dados são mais que números ou fatos, dados são a base para tomada de decisões sobrea as estratégias e direções a serem seguidas.

Dados são gerados e armazenados em todas as áreas, desde o pequeno comércio até a multinacional que precisa de velocidade para armazenar e consultar estes dados. É nesse ponto que um Data Lake é relevante.  
   
Mas qual a razão para utilizar um data lake ao invés de um banco de dados convencional? Pelos 4 motivos a seguir:

1. Escalabilidade;
2. Flexibilidade;
3. Custo Eficiente;
4. Análise de dados Avançada.

Escalabilidade por manusear, tratar e comportar grandes volumes de dados de vários tipos. Imagine que o core do seu negócio possa ser armazenado em um balde de água, trazendo para o sentido de data lake, conforme o tempo passa e a empresa cresce, é necessário trocar e aumentar este armazenamento para uma piscina e logo mais para um lago.  
  
Flexibilidade para se adaptar a diferentes tipos de informações, supondo que seja necessária criar uma nova área de negócio, ou uma nova empresa do grupo empresarial. O data lake pode ser o mesmo já existente, podendo ainda ser utilizado para consultas e análises entre as empresas, uma visão mais superior.

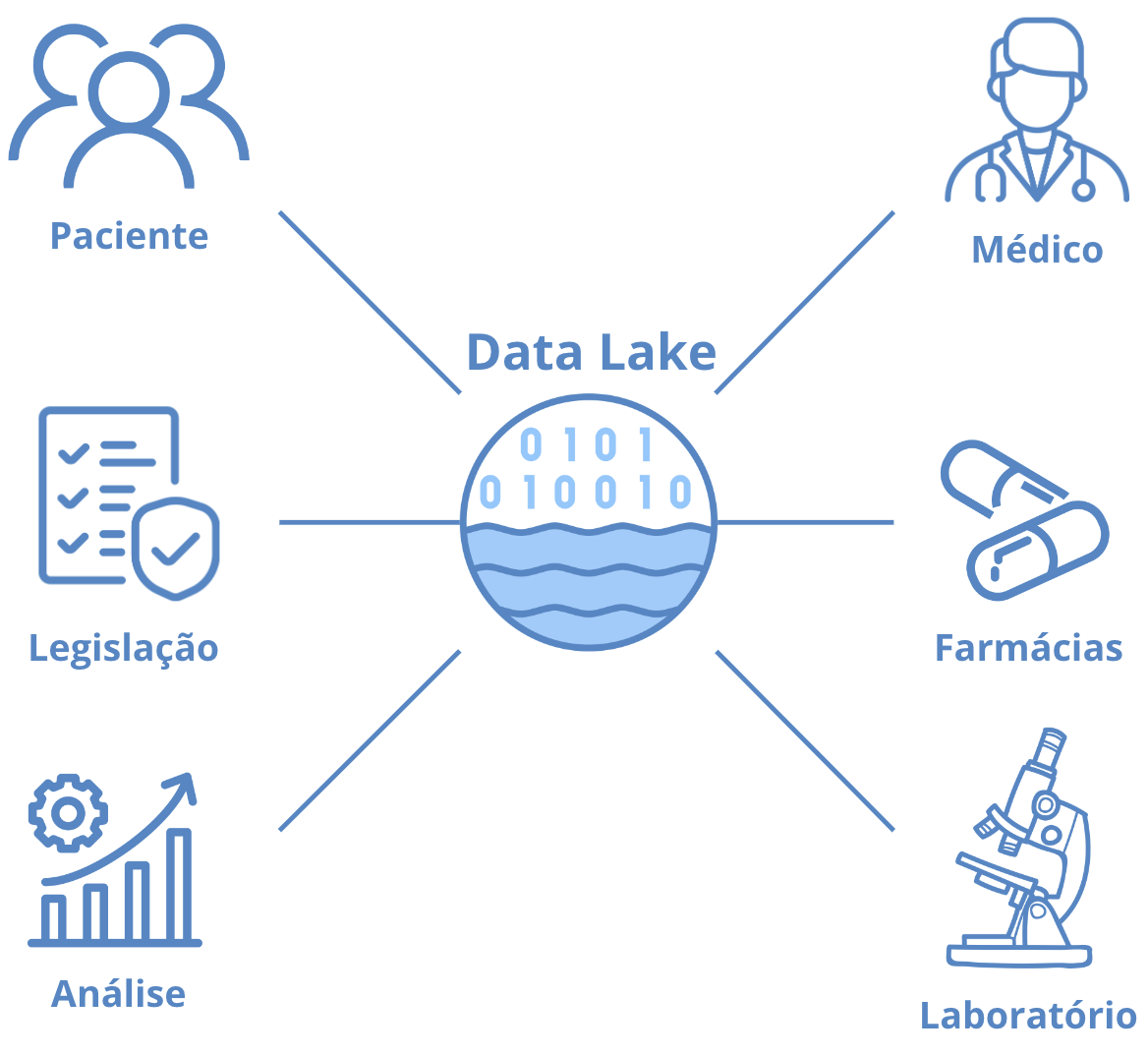
Custo eficiente, pois, como vimos acima, os data lakes são criados utilizando softwares open source, ou seja, não há custo de licenciamento e implantação por parte do fabricante, além de que possuem a capacidade de se integrar com outras ferramentas de análise facilmente. O custo de armazenamento em nuvem escalável também pode ser considerado neste item, pois, utiliza-se somente o necessário de armazenamento. Num passado breve, era necessário adquirir armazenamento on-site, com uma margem e previsão de crescimento, onde comprava-se um volume maior de armazenamento para ficar esperando ser utilizado.

Análise de dados Avançada, a possibilidade de armazenamento em grande escala, escalabilidade de hardware/software e baixo custo, garante que a análise de dados seja avançada, pois, é possível analisar qualquer faceta dos dados, mesmo que precise aumentar o processamento ou utilizar um novo software.

**O Projeto: Data Lake para área de Saúde**

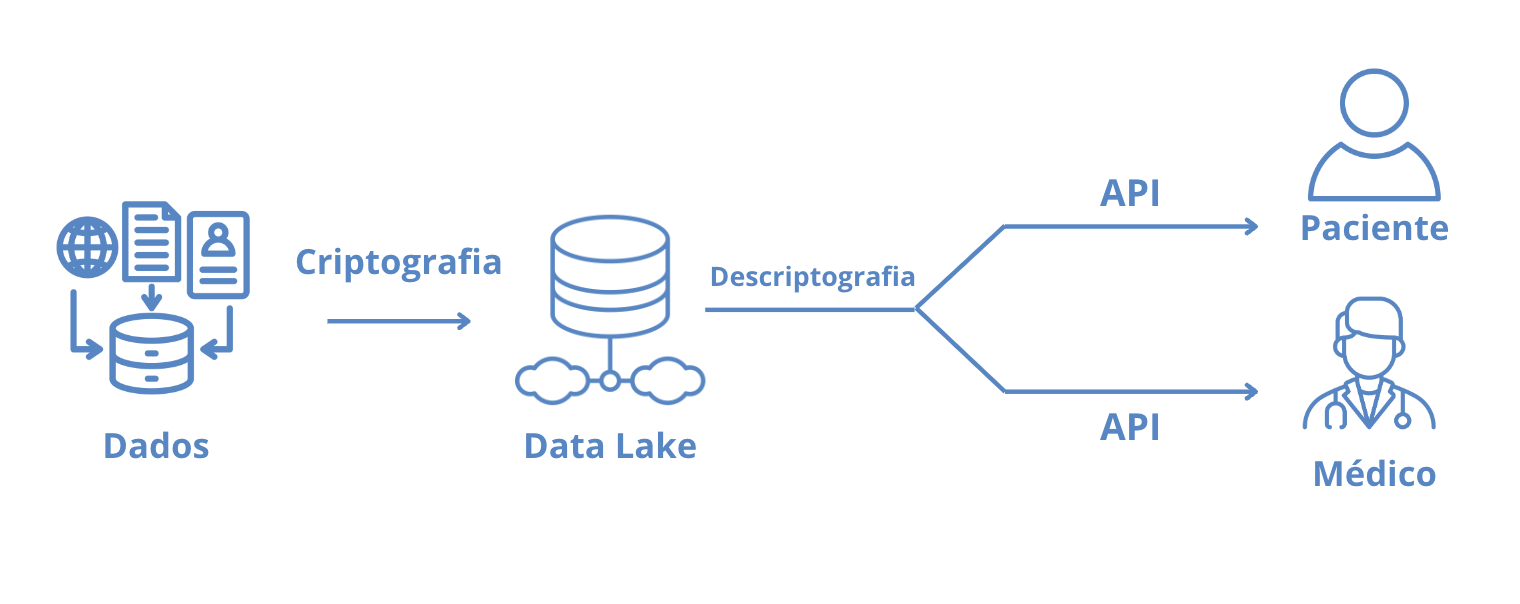
Com base nesta introdução, foi elaborado um projeto de estruturação para um Data Lake que recebe e armazena dados de uma operadora de hospitais em âmbito nacional. As informações contidas neste trabalho são fictícias e foram planejadas para resolver possíveis dificuldades encontradas por essas operadoras.

Para resolver a problemática da desorganização dos dados e promover a eficácia na tomada de decisões estratégicas, propõe-se a centralização das informações em um Data Lake. Este repositório único permitirá o acesso instantâneo às informações por diversos agentes da área da saúde, desde pacientes até analistas de dados. A segurança dos dados é uma preocupação essencial e será garantida por uma abordagem abrangente de criptografia em toda a rede de dados, protegendo a confidencialidade das informações dos pacientes e assegurando a conformidade com regulamentações de privacidade.



A estruturação dos dados será realizada com base no padrão HL7 - Fast Healthcare Interoperability Resources (FHIR), facilitando a integração e a interoperabilidade dos sistemas. O uso de módulos permitirá a representação de entidades de dados específicas, como pacientes, consultas médicas e resultados de laboratório, proporcionando uma estrutura flexível e expansível para o Data Lake.

Para proporcionar uma interface de acesso intuitiva e operável para pacientes e médicos, serão implementadas APIs que organizarão os dados armazenados no Data Lake e os disponibilizarão de maneira estruturada e compreensível. Essa interface personalizada permitirá que cada tipo de usuário acesse informações relevantes de forma eficiente e segura.



Em resumo, a estruturação do Data Lake com base nos princípios de centralização, segurança dos dados, interoperabilidade e interfaces de acesso intuitivas é fundamental para otimizar a análise e o compartilhamento de informações na área da saúde. Essa abordagem contribuirá significativamente para a melhoria dos serviços de saúde, facilitando a tomada de decisões informadas e a implementação de políticas de saúde eficazes.

Estruturas de Armazenamento de Dados:

* Utilização do Amazon Simple Storage Service (S3) em conjunto com o Amazon HealthLake como sistema de armazenamento distribuído em nuvem para escalabilidade e resiliência, garantindo segurança cibernética e eficiência das análises, utilizando conceitos de IA e Machine Learning. Site: <https://aws.amazon.com/pt/healthlake/>
* Formato de dados Parquet para otimizar o desempenho do processamento analítico e maior compressão, se compararmos ao .csv;

Estruturas de Processamento de Dados:

* Streaming: Processamento em tempo real dos dados de saúde para detecção de surtos epidemiológicos, monitoramento de epidemias e notificação de eventos de interesse em saúde pública.
* Batch: Processamento periódico para geração de relatórios de indicadores de saúde, análises epidemiológicas e avaliação de desempenho de políticas de saúde.

Tecnologias Utilizadas:

* Apache Spark: Plataforma de processamento de dados distribuído para análise em larga escala.
* Apache Hive: Data Warehouse para consultas SQL sobre grandes conjuntos de dados de saúde.
* Apache Kafka: Plataforma de streaming para processamento de dados em tempo real.

Estimativa de Volume de Dados:

* Milhões de registros de pacientes, consultas, exames, imagens e prontuários gerados diariamente em todo o país.
* Estimativa de petabytes de dados por ano.

Estimativas de Estrutura de Hardware e Software:

* Armazenamento:
  + Amazon AWS S3.
* Capacidade:
  + Vários petabytes (dimensionado para comportar dados de saúde em larga escala, que podem ser escalonados. Média de crescimento anual em 10%).
* Tipo de instância:
  + S3 Standard e HealthLake Advanced.
* Escalabilidade:
  + De forma automática, segundo informações do site da AWS.

Com este projeto de DataLake, o sistema de saúde nacional poderá consolidar e analisar dados em tempo real para monitoramento epidemiológico, planejamento de recursos de saúde, detecção precoce de doenças e avaliação de políticas de saúde, promovendo campanhas de conscientização sobre prevenção de doenças e melhorando os serviços de saúde pública e o bem-estar geral da população.