Roteiro da Apresentação do Projeto

Equipe e tópicos de fala por responsabilidade e entregáveis

Visão Geral Técnica

- Arquitetura com Spark standalone orquestrado por Docker Compose
- HDFS configurado com fs.defaultFS em hdfs://namenode:8020
- UIs: Spark Master em 8080, Jupyter (spark-client) em 8888, NameNode em 9870, DataNode em 9864
- Pipeline: Coleta → Curadoria → Janelas de evento → Notebooks/Visualizações

Ana Luiza Pazze — Arquitetura e Infraestrutura

Responsabilidades

- Configuração do ambiente Apache Spark
- Implementação da arquitetura de dados distribuída
- Configuração do HDFS e sistemas de armazenamento
- Otimização de performance do cluster Spark
- Implementação de pipelines ETL

Entregáveis

- Documentação da arquitetura técnica
- Scripts de configuração do ambiente
- Pipeline de ETL funcional
- Relatório de performance e otimizações

Roteiro de Fala

Apresentar a topologia do cluster (master, workers) e como o Docker Compose orquestra os serviços.

Explicar o HDFS: fs.defaultFS (hdfs://namenode:8020), partições e políticas de replicação.

Mostrar como os notebooks e jobs se conectam ao cluster e ao HDFS.

Comentar otimizações aplicadas (memória do worker, paralelismo, particionamento por símbolo).

Felipe Martins — API e Coleta de Dados

Responsabilidades

- Implementação da integração com Yahoo Finance API
- Desenvolvimento de coletores de dados de notícias
- Implementação de web scraping para dados complementares
- Tratamento de rate limits e otimização de requisições

- Validação e limpeza de dados coletados

Entregáveis

- Módulos de coleta de dados funcionais
- Documentação das APIs utilizadas
- Scripts de validação de dados
- Relatório de qualidade dos dados coletados

Roteiro de Fala

Demonstrar o coletor Yahoo Finance (símbolos, colunas, limpeza inicial e deduplicação).

Descrever estratégias para rate limits (pausas, retries) e fontes complementares.

Explicar validações: datas válidas, presença de preços e volumes, símbolos com categoria.

Indicar como os dados brutos são enviados ao HDFS e integrados ao pipeline.

Pedro Silva — Análise de Dados

Responsabilidades

- Análise exploratória dos dados financeiros
- Implementação de análises estatísticas
- Desenvolvimento de métricas de impacto
- Análise de correlações e causalidade
- Validação estatística dos resultados

Entregáveis

- Relatórios de análise exploratória
- Implementação de testes estatísticos
- Métricas de impacto definidas e calculadas
- Relatório de correlações identificadas

Roteiro de Fala

Mostrar EDA: distribuição por categorias, evolução temporal e cobertura dos símbolos.

Descrever métricas de impacto com base em janelas de evento (pré/durante/pós).

Apresentar correlações entre categorias e discutir hipóteses de causalidade.

Citar testes estatísticos aplicados e critérios de significância.

Anny Caroline Sousa — Machine Learning

Responsabilidades

- Desenvolvimento de modelos preditivos

- Implementação de algoritmos de classificação
- Análise de sentimento de notícias
- Otimização de hiperparâmetros
- Validação e avaliação de modelos

Entregáveis

- Modelos de machine learning treinados
- Relatório de performance dos modelos
- Sistema de análise de sentimento
- Documentação dos algoritmos implementados

Roteiro de Fala

Apresentar abordagem de features (preços, índices normalizados, sinais de notícias).

Descrever modelos e tarefas (regressão, classificação, previsão de impacto).

Explicar avaliação: métricas, validação cruzada e ajustes de hiperparâmetros.

Mostrar como os resultados se integram ao dashboard e às janelas de evento.

Ricardo Areas — Visualização e Dashboard

Responsabilidades

- Desenvolvimento de visualizações interativas
- Criação de dashboards executivos
- Implementação de relatórios automatizados
- Design de interface de usuário
- Otimização de performance das visualizações

Entregáveis

- Dashboard interativo funcional
- Biblioteca de visualizações reutilizáveis
- Relatórios automatizados
- Documentação de uso das visualizações

Roteiro de Fala

Demonstrar gráfico interativo com filtros de evento e categoria (Plotly/ipywidgets).

Explicar layout e usabilidade do dashboard executivo e KPIs exibidos.

Comentar automações de relatórios e parâmetros ajustáveis.

Apontar otimizações de performance para volumes maiores de dados.

Fabio Silva — Gestão de Projeto

Responsabilidades

- Coordenação geral do projeto
- Gestão de cronograma e entregas
- Documentação técnica e acadêmica
- Preparação de apresentações
- Controle de qualidade e integração

Entregáveis

- Documentação completa do projeto
- Relatório final acadêmico
- Apresentações executivas
- Plano de projeto e cronograma

Roteiro de Fala

Contextualizar objetivos, escopo e marcos do cronograma.

Apresentar integrações entre times e gestão de riscos.

Destacar entregáveis, qualidade e próximos passos.

Referências úteis

- Docker Desktop (imagens de containers e volumes)
- UIs: http://localhost:8080 (Spark Master), http://localhost:9870 (NameNode), http://localhost:9864 (DataNode), http://localhost:8888 (Jupyter)
- HDFS base: /datasets/yahoo_finance/curated
- Notebooks: notebooks/event_analysis_hdfs.ipynb

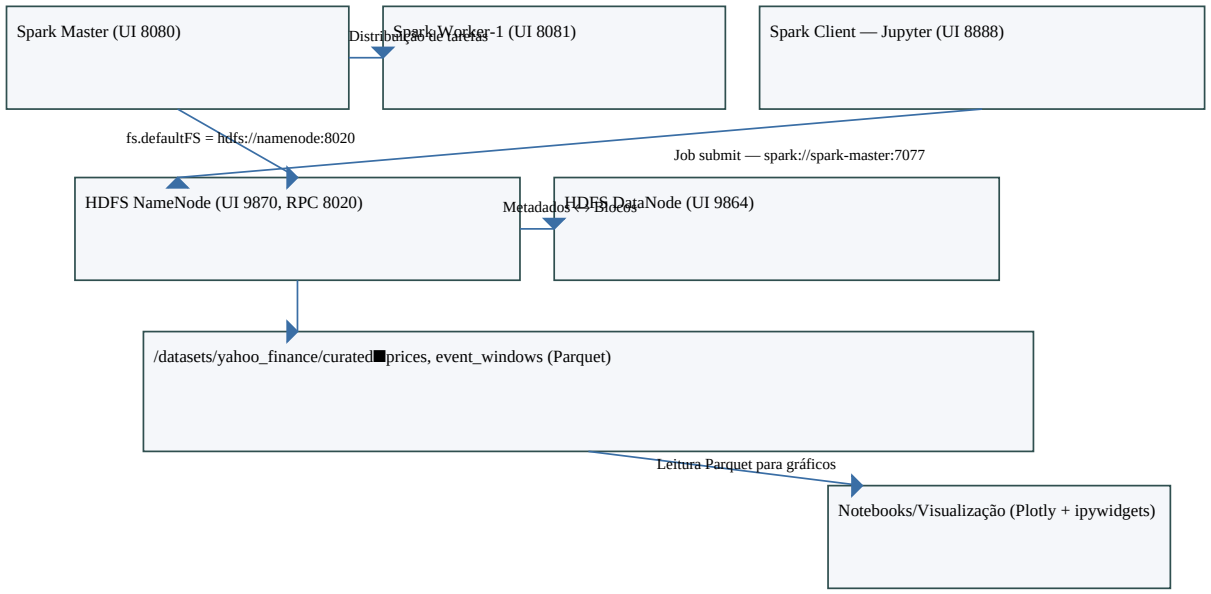
Desenho de Arquitetura

Relação entre Docker Compose, Spark, HDFS e Visualização.

Legenda do Diagrama

- Orquestração: Docker Compose (rede spark-hadoop-net)
- UIs: 8080(Spark), 8888(Jupyter), 9870(NameNode), 9864(DataNode)
- Dados: raw → curated/prices → curated/event_windows

Arquitetura Geral do Projeto



Explicação Detalhada do Desenvolvimento e Configurações

- Orquestração: serviços Spark (master, worker, client) e HDFS (namenode, datanode) sob Docker Compose em rede comum.
- Configurações: spark-defaults define fs.defaultFS=hdfs://namenode:8020; jobs/notebooks usam esse endpoint para IO.
- Coleta: módulo Yahoo Finance produz CSVs brutos, enviados ao HDFS em /datasets/yahoo_finance/raw.
- Curadoria: job Spark lê CSVs, normaliza colunas (datas, AdjClose), enriquece com categorias e escreve Parquet em curated/prices.
- Janelas de Evento: job Spark recorta pre/during/post por âncora, particiona por categoria/símbolo e escreve em curated/event_windows.

- Análise/Visualização: notebook lê event_windows do HDFS, normaliza índices por âncora, agrega por categoria/data e plota gráficos interativos.
- Fluxo de dados: Spark lê/escreve via NameNode que referencia blocos no DataNode; visualização consome Parquet diretamente do HDFS.
- Observabilidade: UIs expostas — Spark (8080), Jupyter (8888), NameNode (9870), DataNode (9864) para acompanhar jobs e estado de armazenamento.

Normalização Base 100 (Índice)

- Definição: índice = (preço atual / preço de referência) × 100.
- Preço de referência (baseline): valor por categoria no dia da âncora do evento; se indisponível, usa-se o primeiro dia disponível por categoria.
- Interpretação: 100 representa o valor na âncora; 110 = +10% em relação à âncora; 95 = –5%.
- Vantagens: torna comparáveis categorias com escalas diferentes; facilita leitura de impactos relativos.
- Limitações: perde magnitude absoluta; sensível à escolha da âncora e à cobertura de dados no dia da âncora.
- Implementação: normalize_by_anchor_spark cria a coluna 'index'; aggregate_for_plot agrega por 'category' e 'date' para visualização.

Exemplo: se a categoria A tem preço de referência 50 na âncora e preço 55 em um dia posterior, então índice = (55/50)×100 = 110 (crescimento de 10%).