

Universidade da Beira Interior

Departamento de Informática



**Departamento de
Informática**

Desenvolvimento de Software para a Nuvem - Projeto Laboratorial

Elaborado por:

**Bernardo Brito Barroca [49435]
Pedro Miguel da Silva Neves [46925]**

Orientador:

Professor Tiago Simões

19 de dezembro de 2024

Conteúdo

1	Introdução	1
1.1	Âmbito, Enquadramento e Motivação	1
1.2	Problema e Objetivos do Trabalho	1
1.3	Abordagem	2
1.4	Organização do Documento	2
2	Estado da Arte, Trabalhos Relacionados	3
2.1	Introdução	3
2.2	Estado-da-Arte	3
3	Engenharia de Software	4
3.1	Introdução	4
3.2	Requisitos Funcionais	4
3.3	Diagramas	4
3.4	Tecnologias Importantes	7
3.4.1	Supabase	7
3.4.2	Hadoop	7
3.4.3	Combinação das Tecnologias	7
4	Execução/Desenvolvimento	8
4.1	Introdução	8
4.2	Dependências	8
4.3	Detalhes de Implementação	9
4.4	Procedimentos de Instalação	12
4.5	Conclusão	12
5	Conclusões e Trabalho Futuro	13
5.1	Conclusões	13
5.2	Trabalho Futuro	13

Capítulo 1

Introdução

1.1 Âmbito, Enquadramento e Motivação

Este documento é um relatório de um projeto laboratorial que tem como objetivo principal o desenvolvimento de um conjunto de serviços para processamento e análise de catálogos do Projeto Gutenberg.

Este projeto laboratorial insere-se na área de computação distribuída de big data. Para trabalhar esta área será usada a framework Apache Hadoop que implementa o modelo de programação MapReduce.

Devido a este crescer constante de quantidade de dados necessários tratar, é importante abordar diferentes soluções para este tratamento e análise de dados.

1.2 Problema e Objetivos do Trabalho

Neste projeto, o problema em específico que será abordado é a gestão e análise de livros existentes no Projeto Gutenberg, que se trata de uma biblioteca com mais de 70000 eBooks gratuitos.

No fim deste projeto, pretende-se ter capacidade de desenvolver um conjunto de serviços que seja capaz de resolver o problema supracitado e, a partir do conhecimento adquirido, conseguir abordar outros problemas da mesma área.

1.3 Abordagem

Para conseguir resolver o problema identificado, será utilizado o estilo arquitetural REST no desenvolvimento do conjunto de serviços referido no primeiro parágrafo. Será também usada a plataforma Supabase para desenvolvimento e instalação desses mesmos serviços e, a framework escolhida para processar e analisar o catálogo será o Apache Hadoop.

Após a identificação das tecnologias a serem usadas, a ordem de trabalho passa pelos seguintes tópicos:

1. **Análise dos Requisitos** - Identificar as funcionalidades necessárias para a conclusão do projeto;
2. **Preparação do Ambiente de Desenvolvimento** - Configuração do Supabase para gestão dos dados, configuração da framework Hadoop numa VM criada pelo hypervisor Oracle Virtual Box, configuração inicial do ambiente Node.js para o middleware;
3. **Implementação das Funcionalidades de Gestão do Catálogo** - Criação das rotas API REST (GET, POST, PUT, DELETE) para gerir os dados do catálogo e, primeiro carregamento do catálogo no Supabase;
4. **Desenvolvimento do MapReduce com Hadoop** - Implementação das funções de mapeamento e redução em Python;
5. **Desenvolvimento do Middleware para Comunicação** - Implementação de um backend em Node.js para comunicação entre Hadoop e Supabase;
6. **Validação e Testes** - Testar as funcionalidades, como por exemplo, operações REST. Também, validar os resultados do processamento Hadoop.

1.4 Organização do Documento

Capítulo 2

Estado da Arte, Trabalhos Relacionados

2.1 Introdução

Este capítulo descreve algumas das ferramentas e tecnologias fundamentais que possibilitam a implementação de projetos semelhantes ao desenvolvido neste trabalho. O capítulo está estruturado da seguinte forma: na seção 2.2, explora-se o estado-da-arte das principais tecnologias utilizadas.

2.2 Estado-da-Arte

Apache Spark - O Apache Spark seria excelente uma alternativa ao Hadoop. Esta framework oferece um desempenho mais rápido que o Hadoop, e é indicada para projetos que exigem processamento de dados em tempo real.

Firebase - O Firebase é uma plataforma que oferece uma solução sólida para a gestão e armazenamento de dados de catálogos, sendo uma alternativa ao Supabase -ferramenta utilizada neste projeto.

Capítulo 3

Engenharia de Software

3.1 Introdução

Este capítulo aborda os aspetos relacionados à engenharia de software que sustentam o desenvolvimento do projeto. Primeiramente, vão ser definidos os requisitos funcionais que descrevem as principais funcionalidades esperadas do sistema. Após isto, serão apresentados alguns diagramas para detalhar a interação entre os utilizadores e o sistema.

3.2 Requisitos Funcionais

Os requisitos funcionais do sistema que foram desenvolvidos para a realização deste projeto são os seguintes:

1. **Gestão do Catálogo de Livros** - Permitir as operações GET, DELETE, PUT E POST no catálogo de livros;
2. **Análise do Catálogo de Livros** - Realizar a contagem de livros publicados por ano;
3. **Armazenamento de Dados Processados** - Guardar os resultados da análise feita ao catálogo no Supabase.

3.3 Diagramas

Alguns diagramas criados para ajudar a entender a interação do utilizador com o sistema vão agora ser apresentados. Foram desenvolvidos três diagramas de caso de uso para ilustrar a interação POST, DELETE e PUT respetivamente ; um diagrama de atividade foi desenvolvido para explicar a interação

GET; por fim, foi desenvolvido um diagrama de atividade para ilustrar a interação de armazenar os resultados da análise feita ao catálogo:

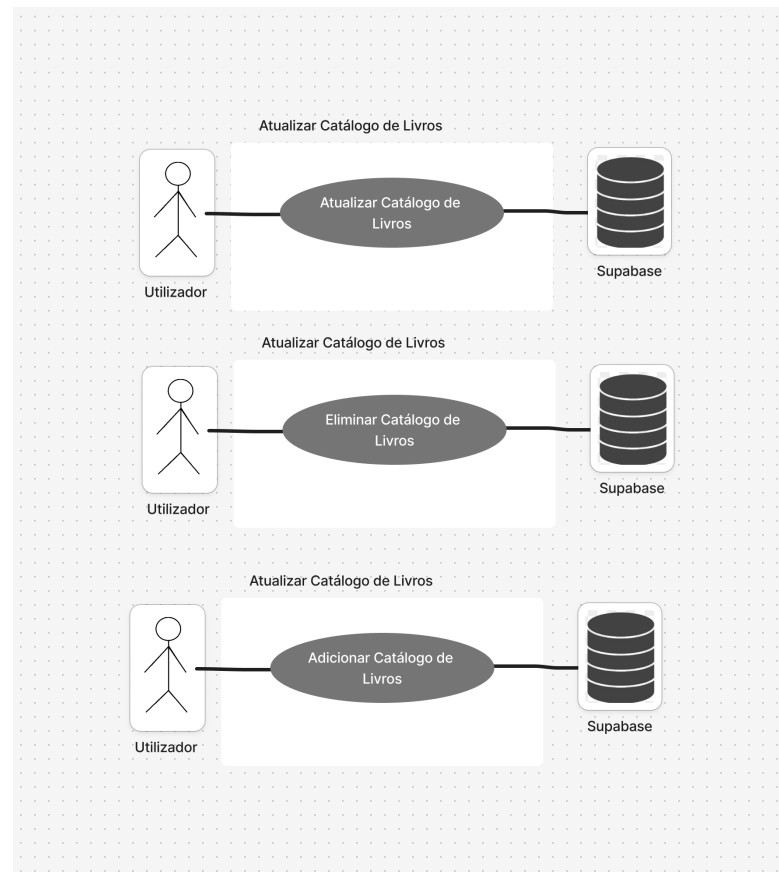


Figura 3.1: Diagramas de Caso de Uso

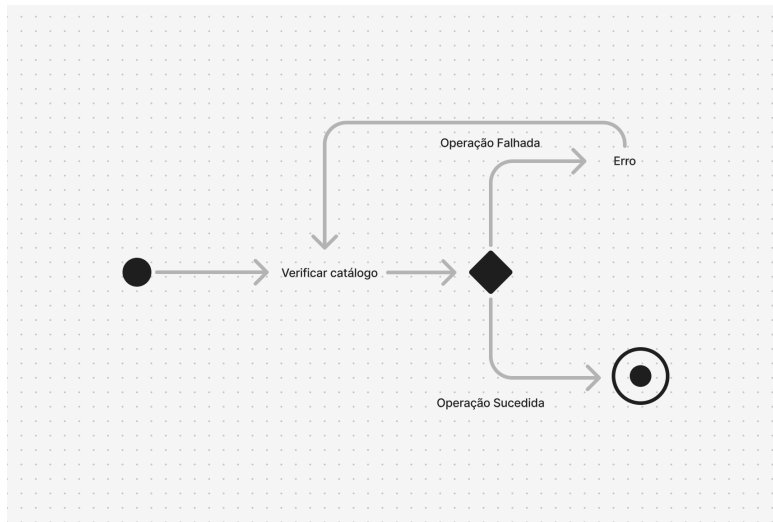


Figura 3.2: Diagrama de Atividade

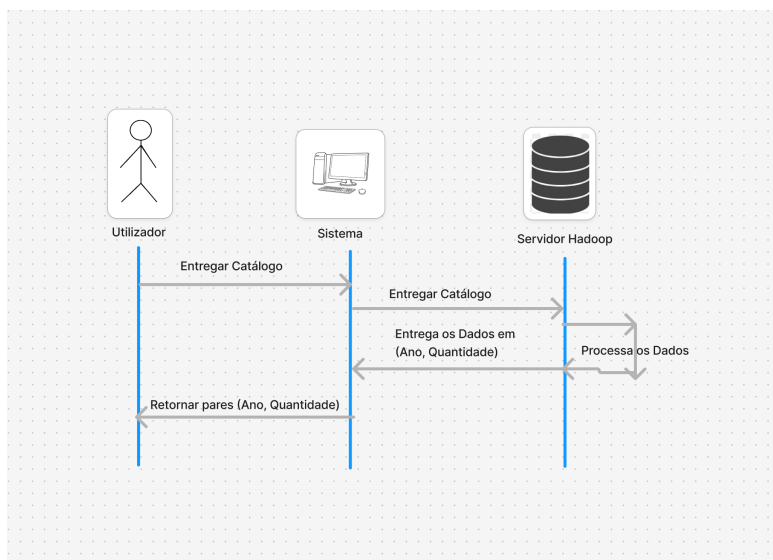


Figura 3.3: Diagrama de Sequência

3.4 Tecnologias Importantes

3.4.1 Supabase

O Supabase foi escolhido para o desenvolvimento deste projeto, devido a ser uma solução de back-end completa e de fácil implementação. A sua base de dados relacional permite uma estruturação organizada e eficiente dos dados a armazenar dos catálogos de livros.

Outro fator importante para a escolha do Supabase foi a sua fácil integração com API's REST.

3.4.2 Hadoop

Foi escolhida a framework Hadoop para o processamento e análise dos dados do catálogo de livros. Esta escolha deveu-se à sua capacidade de lidar com grandes volumes de dados de forma distribuída e escalável.

Ao usar o modelo de programação MapReduce, tornou-se a principal candidata, visto ser este modelo o necessário para realizar o objetivo do projeto.

3.4.3 Combinação das Tecnologias

Ao combinarmos o Supabase com o Hadoop, tornou-se a dupla crucial para o nosso trabalho: o Supabase gere e armazena o catálogo de livros, e o Hadoop processa distribuidamente os dados requisitados.

Assim, ao integrarmos as duas tecnologias, o sistema consegue processar os dados com o Hadoop e, consequentemente, armazená-los na base de dados do Supabase. Com esta combinação, foi criada uma solução escalável, robusta e eficiente para a gestão e análise do catálogo de livros.

Capítulo 4

Execução/Desenvolvimento

4.1 Introdução

Este capítulo descreve o processo de execução e desenvolvimento do projeto, abordando as principais etapas realizadas para garantir as funcionalidades necessárias do sistema desenvolvido. Os seguintes tópicos serão abordados: dependências necessárias, implementações realizadas em cada componente do sistema; procedimento para instalação e configuração do ambiente; considerações finais sobre o processo de desenvolvimento.

4.2 Dependências

O desenvolvimento deste projeto envolveu o uso de algumas tecnologias e ferramentas que desempenharam papéis fundamentais na construção do sistema. Algumas das dependências principais incluem:

- Supabase - Utilizado como plataforma de back-end para gestão de dados, utilizando uma base de dados relacional (PostgreSQL). Também utilizado para interagir com o sistema a partir de APIs REST,
- Hadoop - Framework utilizada para processar o catálogo, com a utilização do modelo de programação MapReduce para funções de mapeamento e redução.
- Python - Linguagem de programação utilizada para escrever as funções `mapper.py` e `reducer.py`, estas que são executadas no ambiente Hadoop.
- Node.js - Framework utilizada para criação de APIs REST (GET, PUT, DELETE, POST)..

Estas são algumas das dependências mais importantes para o desenvolvimento do projeto.

4.3 Detalhes de Implementação

1. Gestão de Catálogos - Foram escritas API's REST utilizando o Node.js, com as funcionalidades GET, DELETE, POST, PUT de modo a poder consultar, remover, criar e atualizar itens do catálogo. Esta implementação pode ser visualizada nas figuras seguintes:

```
// GET para procurar todos os catálogos
router.get("/catalogs", async (req, res) => {
  try {
    const response = await axios.get(`${SUPABASE_URL}/rest/v1/catalog`, {
      headers: {
        apikey: SUPABASE_KEY,
      },
    });
    res.json(response.data);
  } catch (error) {
    res.status(500).send(error.message);
  }
});
```

Figura 4.1: Request GET

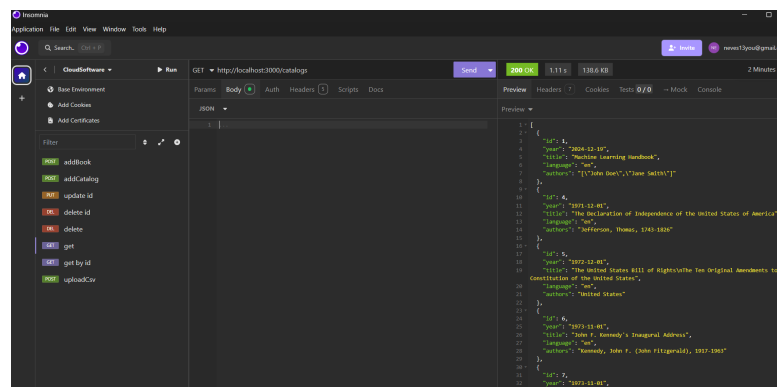


Figura 4.2: Return do GET

2. Análise dos Catálogos - As análises dos catálogos foram feitas através da framework Hadoop. Implementaram-se funções de mapeamento e redução em Python, que processam os dados dos catálogos, extraem o ano de publicação e realizam operações de contagem e agrupamento dos livros emitidos por ano. Nas seguintes figuras encontram-se a função mapper, a função reducer e o output do MapReduce, respetivamente.

```
#!/usr/bin/env python3
import sys

for line in sys.stdin:
    line = line.strip()

    try:

        first_comma_index = line.index(",")

        year = line[first_comma_index + 1:first_comma_index + 5]

        if year.isdigit():
            print(f"{year}\t1")
    except (ValueError, IndexError):
        continue
```

Figura 4.3: mapper.py

```
#!/usr/bin/env python3
import sys

current_year = None
current_count = 0

print(f"{'Year':<15}{'Count':<10}")
print("=" * 25)

for line in sys.stdin:
    line = line.strip()

    try:
        year, count = line.split("\t")
        count = int(count)
    except ValueError:
        continue

    if current_year == year:
        current_count += count
    else:
        if current_year is not None:
            print(f"{'current_year':<15}{'current_count':<10}")
            current_year = year
            current_count = count

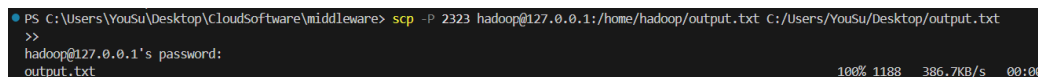
if current_year is not None:
    print(f"{'current_year':<15}{'current_count':<10}")
```

Figura 4.4: reducer.py

```
hadoop@neves:~$ hadoop fs -cat /user/Hadoop/output/* > output.txt
hadoop@neves:~$ ls
catalog.csv  convert.py  hadoop  hadoop-3.3.4.tar.gz  hadoopdata  mapper.py  output.txt  reducer.py
```

Figura 4.5: Ficheiro Output

Armazenamento e Exibição de Dados- A partir do ficheiro output.txt, inserimos os dados na base de dados. Para extrair o ficheiro da máquina virtual, executamos o comando de cópia segura scp na máquina local com a porta (-p 2323), pois foi a porta configurada por nós na VM. O armazenamento na base de dados é feito como se pode observar nas seguintes figuras:



```
PS C:\Users\YouSu\Desktop\CloudSoftware\middleware> scp -P 2323 hadoop@127.0.0.1:/home/hadoop/output.txt C:\Users\YouSu\Desktop/output.txt
>>
hadoop@127.0.0.1's password:
output.txt
100% 1188 386.7KB/s 00:00
```

Figura 4.6: Extrair o ficheiro output.txt para a máquina local

4.4 Procedimentos de Instalação

Para fazer todas as instalações necessárias, seguimos todos os passos que estão descritos na seguinte aula: <https://hackmd.io/@UBI/HkiLqnmV6>. Todos os passos seguidos para a instalação podem ser encontrados nos pontos 3.1 e 3.2 dessa mesma aula.

Esta aula é da autoria do Professor Tiago Simões.

4.5 Conclusão

Neste capítulo foi abordado o processo de execução e desenvolvimento do sistema, desde as dependências necessárias até os detalhes da implementação e a instalação do sistema.

As ferramentas utilizadas foram fundamentais para garantir que os requisitos do projeto fossem cumpridos. O procedimento de instalação foi referido para facilitar a implementação e configuração do ambiente, permitindo que qualquer utilizador possa replicar o sistema no seu próprio ambiente de desenvolvimento.

A conclusão deste capítulo mostra que, o projeto é capaz de fornecer uma solução robusta e funcional para a gestão e análise de catálogos de livros.

Capítulo 5

Conclusões e Trabalho Futuro

5.1 Conclusões

Ao longo deste trabalho, foi possível atingir os objetivos propostos inicialmente. A implementação bem-sucedida das APIs REST para a gestão de catálogos, o processamento eficiente dos dados através do uso de MapReduce, e o armazenamento estruturado utilizando o SupaBase comprovaam a viabilidade da solução desenvolvida. Esses resultados demonstram a capacidade do sistema de atender às necessidades de consulta, manipulação e análise de dados de forma eficaz e escalável.

Uma das principais conclusões deste trabalho foi a importância de se combinar tecnologias específicas para cada parte do projeto, maximizando o desempenho geral. A integração das ferramentas demonstrou ser eficiente tanto em termos de desenvolvimento quanto na execução prática do sistema. A abordagem tomada para este projeto, demonstrou facilidade em lidar com grandes volumes de dados.

Por fim, o aprendizado obtido em relação ao uso de uma combinação de tecnologias para resolver o problema que é o processamento e análise de um grande volume de dados, um dos pontos positivos do projeto.

5.2 Trabalho Futuro

Embora os objetivos principais tenham sido atingidos, poderiam ter sido cumpridos alguns requisitos adicionais. Para além dos requisitos adicionais, ainda existem algumas funcionalidades que seriam interessantes de implementar com

mais tempo, como por exemplo, melhorar o sistema para incluir funcionalidades mais avançadas. Uma funcionalidade bastante interessante para adicionar, seriam análises preditivas baseadas em machine learning para prever tendências nos catálogos de dados.

Além disso, também poderia ser criada uma interface, para a gestão de dados oferecer uma melhor experiência ao utilizador. Atualmente, o sistema depende de interações via APIs, mas uma interface gráfica interativa poderia simplificar o uso e tornar o sistema mais interessante.

Por fim, seria interessante explorar a aplicabilidade do sistema em outros domínios além da gestão de catálogos de livros. Muitos setores enfrentam dificuldades no processamento e gestão de grandes volumes de dados, tornando realmente interessante a possibilidade de ajustar o sistema para outro tipo de dados.

Em suma, este projeto apresenta potencial para evoluções, sendo uma base sólida para trabalhos que pretendem explorar ainda mais as capacidades das tecnologias utilizadas.