# SISTEMA DE GERENCIAMENTO DE CONTEÚDO

Grupo 4

**CP904TIN2** 

Alanys Ribeiro Santos, RA: 200755
Alison Leme dos Santos, RA: 171422
André Luiz dos Santos Leite, RA: 200036
Enzo Yuki Takamura Pasqualini, RA: 200477

Felipe Moraes Marcello, RA: 200370 Raphael Augusto Santos, RA: 200361

Prof. Marco Antônio Montebello Junior



# INTRODUÇÃO

A digitalização transformou a produção, armazenamento e consumo de conteúdo, especialmente em mídia e notícias. Plataformas eficientes de gerenciamento de conteúdo são cruciais para lidar com grandes volumes de dados. A escalabilidade, desempenho e disponibilidade são essenciais para uma experiência de usuário consistente e rápida. O Apache Cassandra é uma escolha valiosa para sistemas de gerenciamento de conteúdo devido à sua capacidade de escalar horizontalmente, alta disponibilidade e eficiência no gerenciamento de diversos tipos de dados, enfrentando assim os desafios modernos de forma robusta.

Figura 1. Logo da Cassandra.



### **JUSTIFICATIVA**

O Apache Cassandra é ideal por sua alta disponibilidade, tolerância a falhas e escalabilidade. Sua sintaxe SQL (CQL) facilita a adoção por desenvolvedores, e sua modelagem de dados flexível adapta-se a diversos requisitos de aplicativos.

### **OBJETIVOS**

Desenvolver uma API de gerenciamento de conteúdo utilizando o banco de dados Cassandra, um banco não relacional, para garantir armazenamento de dados mais flexível e eficiente. A API deve assegurar escalabilidade horizontal, alto desempenho e alta disponibilidade. Além disso, deve oferecer uma modelagem de dados eficaz e proporcionar uma experiência de usuário rápida e intuitiva para o armazenamento e entrega de artigos, imagens e vídeos.

### **ORÇAMENTO**

**Tabela 1**. Orçamento.

Orçamento		
Tecnologia	Preço	Limitações c/ base no Preço
Visual Studio Code	Gratuito	-
Postman	Gratuito	80GB Armaz. & 20 M. read/write
Astra Datastax	Gratuito	25 collections
Cassandra	Open Source	-

## RESULTADOS E VALIDAÇÃO

Para validação da API, foram criadas duas tabelas distintas, uma de "Autores" e outra de "Artigos". Na tabela "Artigos", a relação entre as duas tabelas é feita através do "autor\_id", conforme figuras 1 e 2 abaixo.

Figura 2. Tabela de Autores.

```
token@cqlsh> CREATE TABLE facens.autores (
... id UUID PRIMARY KEY,
... nome TEXT,
... email TEXT,
... bio TEXT
...);
```

Figura 3. Tabela de Artigos.

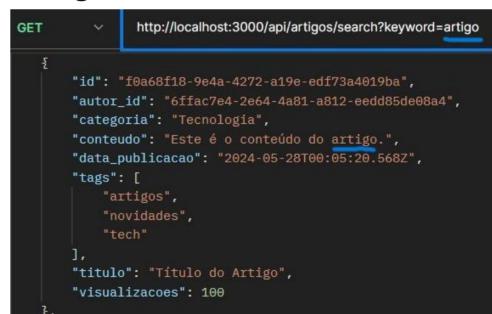
```
token@cqlsh:facens> CREATE TABLE artigos (
... id UUID PRIMARY KEY,
... autor_id UUID,
... titulo TEXT,
... categoria TEXT,
... conteudo TEXT,
... data_publicacao TIMESTAMP,
... tags SET<TEXT>,
... visualizacoes INT
...):
```

Como exemplo, será realizada uma busca pela palavra "Artigo" dentro do conteúdo, conforme ilustrado nas figuras 4 e 5, respectivamente.

Figura 4. Comando Executado.

```
const query =
   "SELECT * FROM facens.artigos WHERE TOKEN(conteudo) IN TOKEN(?)";
```

Figura 5. Retorno do Comando.



#### **CONCLUSÃO**

O projeto alcançou com sucesso o desenvolvimento de uma API de gerenciamento de conteúdo utilizando o banco de dados Cassandra, assegurando escalabilidade horizontal, alto desempenho e alta disponibilidade. A flexibilidade do Cassandra permitiu trabalhar com um modelo de dados semelhante ao SQL, beneficiando-se da arquitetura de linhas e colunas e da arquitetura de documentos. A fácil configuração e suporte a diversas linguagens foram pontos positivos. No entanto, a ausência de suporte nativo para consultas complexas e a complexidade na configuração e manutenção de clusters apresentaram desafios significativos.

#### **AGRADECIMENTOS**

Aos membros do grupo, os quais ajudaram desde o início do projeto até o final.

Ao professor, pelos auxílios e correções do projeto.