

# Desafio AeC Automação

---

## Descrição

Esse documento visa dar um overview sobre o projeto de **web scraping do site Alura**, na parte de cursos, artigos e demais conteúdos retornados na pesquisa do termo enviado para o site.

Foi utilizado o tipo de projeto Console por se tratar de um projeto mais rápido de ser implementado, requerendo menos codificação.

## Arquitetura

A solução apresentada segue padrões de arquitetura DDD, com as seguintes camadas (projetos):

- **Console** – Onde reside o Program Main e de onde parte as chamadas dos serviços de scraping. **Faz referência aos projetos: Common e Domain**
- **Domain** – Onde temos as definições dos contratos/interfaces a serem implementadas pelas demais camadas, e onde se tem a única entidade “**AluraContent**” dentro da pasta Alura, que vai representa a unidade de um documento/linha/registro associado aos dados dos cursos e artigos e todo e qualquer conteúdo recuperados do site Alura. **Não faz referência a nenhum projeto.**
- **Service** – Onde temos as implementações das operações crud a serem realizadas no modelo de armazenamento nosql do projeto, também dividida em pastas, contendo inicialmente apenas a pasta Alura. **Faz referência ao projeto Domain.**
- **Common** – Onde temos a parte responsável pela **injeção de dependência** do projeto. **Faz referência aos projetos Domain e Service.**

O projeto faz o armazenamento de dados em um banco **NoSql MongoDB**, rodando em um **container Docker**. **Para fins de teste, o banco foi levantado sem usuário e senha.**

O projeto acompanha um arquivo **docker-compose.yml** para ser executado antes dos teste, para que o container do banco seja levantado. Abaixo a estrutura do arquivo.

```
version: "3.8"
services:
  mongodb:
    container_name: mongodb_scraping
    image: mongo:latest
    ports:
      - 27017:27017
    volumes:
      - mongo_data:/data/db
    restart: always
volumes:
  mongo_data:
```

## Observação:

- **Os testes foram realizados localmente numa máquina rodando em ambiente Windows com Docker for Windows.**

# Desafio AeC Automação

---

## Decisões Técnicas

Para o processo de scraping no site Alura, foi criada uma classe específica para o serviço, deixando toda a lógica dentro dela. Com isso, outras classes poderão ser anexadas futuramente com outros scrapings, deixando a lógica de cada serviço isolado e de fácil manutenção.

Foi usado um modelo **NoSQL de banco para facilitar os testes** e por ser um banco mais rápido de ser usado e para os fins deste teste.

Foi baixado do **Docker Hub a versão mais atual da imagem do banco**, assim como o **package NuGet para uso do MongoDB**, é a mais recente.

Nas camadas **Domain e Service**, buscou-se aplicar o pattern de **classes genéricas** para implementar as operações de banco, fazendo uso da injeção de dependência, facilitando as chamadas declarativas das operações, diminuindo acoplamentos e facilitando a manutenibilidade do código.

### Observações:

- *Dentro da solution, se encontra anexado o binário do navegador Chrome, que é acionado durante a execução do projeto. Leia o tópico Observação desse manual para mais detalhes.*
- *Para poder realizar os testes do sistema, primeiro abra um terminal na pasta onde se encontra o projeto Console, e execute o comando `docker-compose up -d`, para que seja baixada a imagem do MongoDB e levantado o container.*
- *Sugere-se que utilize uma IDE gráfica que facilite a visualização dos dados capturados. Sugiro o MongoDB Compass.*
- *Para download da IDE MongoDB Compass, acesse o link abaixo:*

<https://www.mongodb.com/try/download/compass>

# Desafio AeC Automação

---

## Lógica

O sistema faz a captura dos seguintes dados:

- **Título do Conteúdo**
- **Descrição do Conteúdo**
- **Link do Conteúdo**
- **Nomes dos Professores**
- **Carga Horária**

Esses dados estão representados na classe **AluraContent** localizada no projeto **Domain**.

A lógica segue basicamente dois passos:

- Primeiro passo é percorrer todas as páginas de conteúdos do site, retornados da pesquisa, recuperando **título, descrição e link do conteúdo**.
- Num segundo passo, por meio do link salvo em cada conteúdo, é acessada a página de cada um e recuperado o **nome dos professores** e a **carga horária**.

Foram aplicados try/catchs em pontos críticos do projeto para garantir que o mesmo rode sem quebras.

### Observações:

- *Um ponto fundamental para execução do projeto, caso venha a ser feito via linha de código, é que o mesmo requer a passagem de apenas um parâmetro de pesquisa.*
- *Para fins apenas de teste, foram apenas considerados termos simples, sem espaços em branco. Portanto, no exemplo usado no desenvolvimento, foram os termos “IA” ou “Asp.net”, mas podem ser usados outros, desde que apenas 1 único termo sem espaços em branco.*

# Desafio AeC Automação

## Observação

Após baixar o projeto para a máquina local, abrir o arquivo **App.config**, e alterar o caminho indicado no box vermelho, apontando para o caminho onde você instalou a solution, mantendo apenas o trecho final “\\Console\_Scraping”. Atentar para o uso de duplas de “\\”.

Esse caminho se refere ao local onde o sistema vai buscar o executável **Chromedriver.exe**.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <appSettings>
    <add key="chromepath" value="E:\2-PROJETOS\AeC\ScrapingSolution\\Console_Scraping" />
    <add key="ContentCollectionName" value="Content"></add>
    <add key="ConnectionString" value="mongodb://localhost:27017"></add>
    <add key="DatabaseName" value="Mongodb_scraping"></add>
  </appSettings>
</configuration>
```

## Estrutura de Armazenamento no MongoDB

Abaixo, as colunas que representam o documento “**Content**”, criado no MongoDB, com os dados são preenchidos.

*\_id: ObjectId('66ab7a47761a6fc52d9d5fb6')*

*Title: "Formação C# Web: crie aplicações usando ASP.NET"*

*Description: "Nesta formação você vai aprender tudo o que precisa para criar uma apl..."*

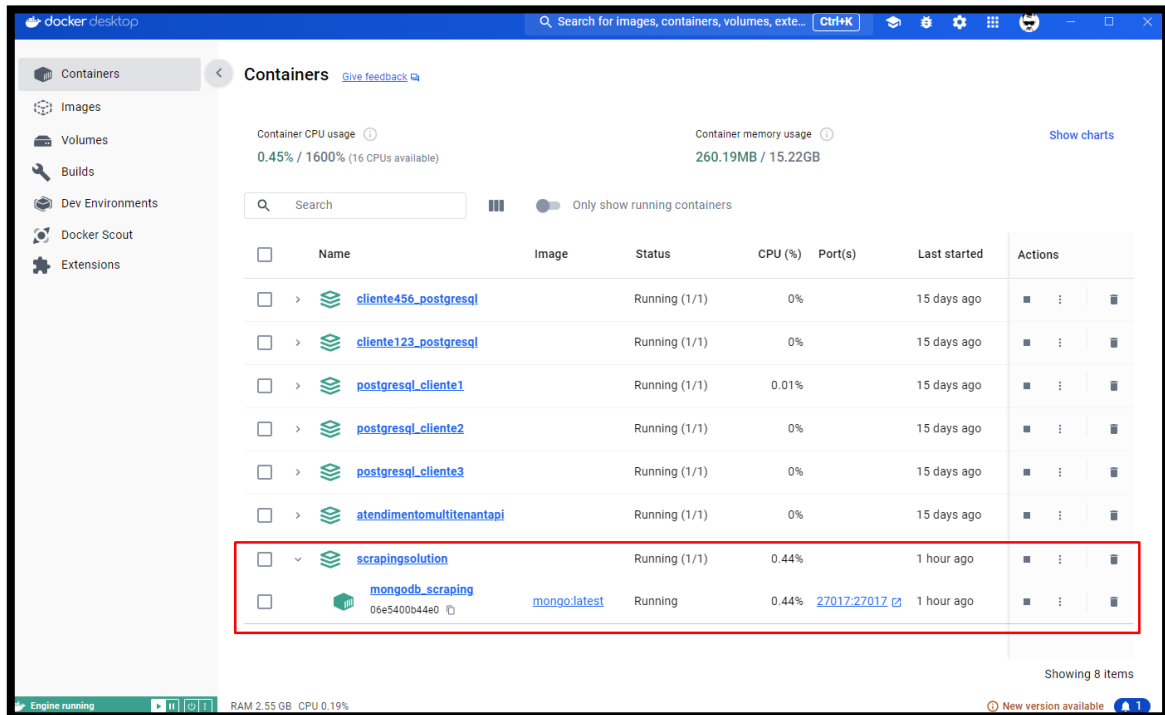
*Link: "https://www.alura.com.br/formacao-csharp-web"*

*Professores: "André Bessa|Daniel Portugal|Jeniffer Bittencourt"*

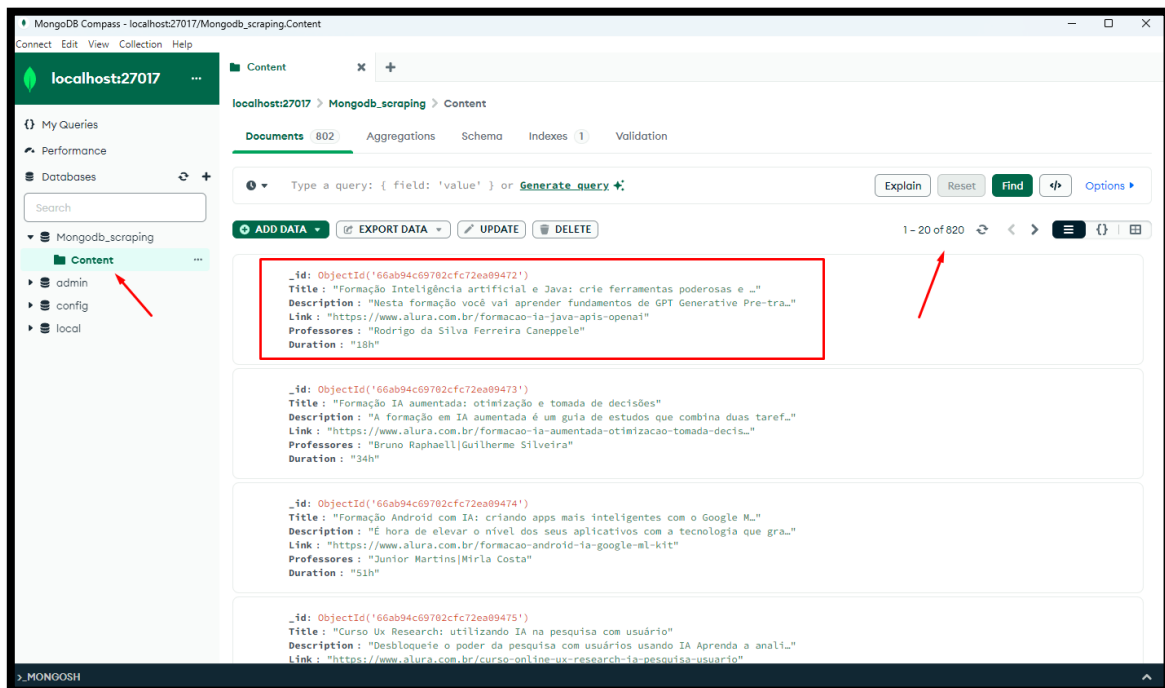
*Duration: "36h"*

# Desafio AeC Automação

## Prints

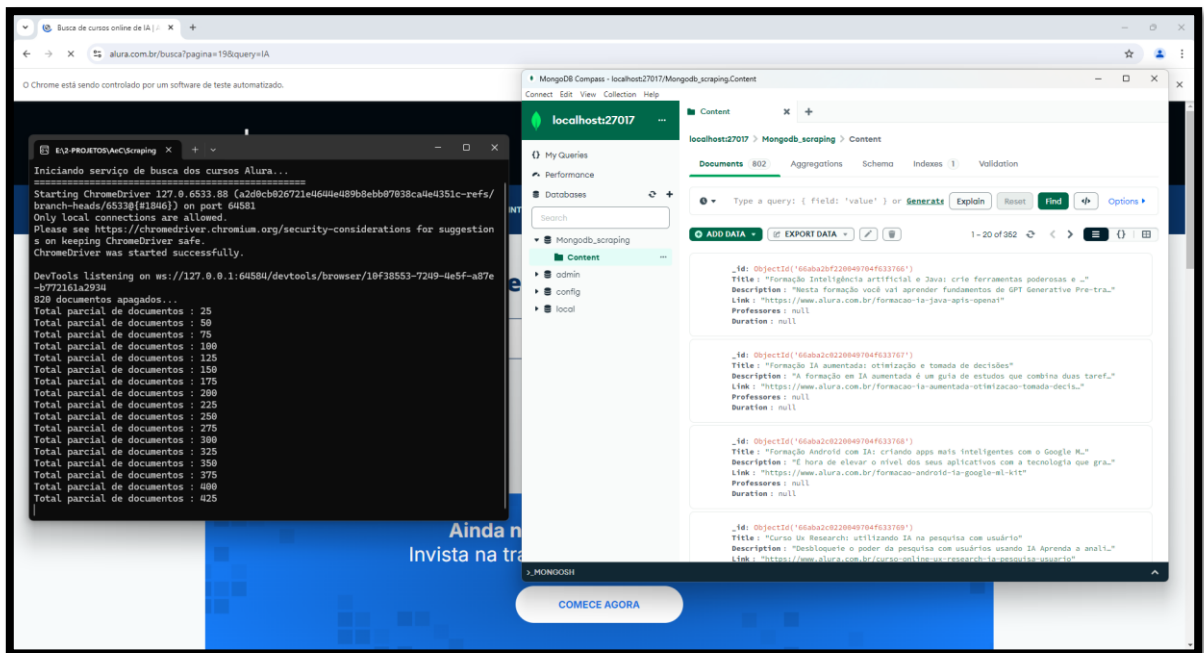


Container do banco MongoDB



Exemplo com alguns dados do banco MongoDB visto pela IDE MongoDB Compass

# Desafio AeC Automação



Ambiente local com o projeto em execução