**Descrição do Projeto e Decisões Técnicas**

**1. Estrutura do Projeto**

O projeto consiste em acessar o site da Alura e fazer uma busca de cursos por um valor pré-definido capturando os dados do nome do curso, carga horária, professor e descrição do curso e armazenando esses dados em uma base de dados access.

O projeto foi desenvolvido em camadas respeitando a abordagem **Domain-Driven Design (DDD)** juntamente com a **injeção de dependência** permitindo um fácil entendimento e manutenção no código do projeto.

**2. Injeção Direta de URLs e Processamento de Cursos**

A automação foi configurada para navegar diretamente a URL de busca da Alura, por exemplo, [**https://www.alura.com.br/formacao-excel**](https://www.alura.com.br/formacao-excel), refletindo em um ganho de tempo ao capturar as informações de cada curso e acelerando o processo. Além disso, o scraper lista todas as páginas de cursos disponíveis, extrai os links de todos os cursos e, em seguida, acessa cada link individualmente para buscar as informações detalhadas dos cursos.

Antes de navegar na URL do curso, a automação verifica se o curso já existe na base de dados, utilizando o link do curso como chave de busca na tabela para evitar buscas desnecessárias e duplicadas.

**3. Serviço de Logs**

Foi adicionado um serviço de log, **ILogService**, para registrar tudo o que acontece durante a execução da automação, como início, erros e sucesso.

**4. Tratamento de erros**

* Efetua 3 tentativas em todos os controles com um intervalo de 1s a cada tentativa.
* Ao navegar nos cursos foi adicionado um tratamento caso o site informe erro 500 – Bad Gateway – O mesmo tenta navegar novamente no curso com um intervalo de 1s.
* Tratamento de caracteres especiais ao montar o nome dos cursos a serem utilizados na navegação por url.
* Todos os processos são adicionados ao log de eventos da automação.
* Caso a versão do Chrome seja incompatível com a versão sendo utilizada pela automação a mesma faz o download da biblioteca e atualiza os drivers necessários para a execução.

**5. Ferramentas e Dependências Usadas**

* **.NET com Visual Studio 2022**: Utilizado para criar e compilar o código.
* **Selenium**: Utilizado para a navegação no Site da Alura e o scraping.
* **System.Data.Odbc**: Usado para realizar a conexão com o banco de dados **Access** e interagir com as tabelas.
* **Banco de Dados Access**: O banco de dados usado para armazenar os dados coletados dos cursos. O arquivo de banco de dados é armazenado em **/Infraestrutura/Data/AluraCourses.accdb** e contém duas tabelas principais:
  + **TBL\_Cursos**: Armazena os dados dos cursos, como título, descrição, duração, etc.
  + **TBL\_Logs**: Armazena os logs da execução da automação, como informações de sucesso e erros.

**6. Fluxo do Programa**

O programa segue o seguinte fluxo de execução:

* **Entrada de Dados**: O termo de busca é verificado ou solicitado ao usuário.
* **Configuração e Injeção de Dependências**: O sistema configura todas as dependências necessárias, como o scraper, repositórios e serviços de log.
* **Scraping**: O scraper inicia a busca pelo curso informado, acessando e navegando pelas páginas de cursos. Para cada curso listado, o scraper extrai os links e, antes de buscar mais detalhes, verifica se o curso já está registrado na base de dados para evitar duplicação.
* **Armazenamento dos Dados**: As informações de cada curso são armazenadas na tabela **TBL\_Cursos** do banco de dados.
* **Registro de Logs**: O progresso e os erros são registrados na tabela **TBL\_Logs** para monitoramento.

**7. Arquitetura e Organização do Código**

O código foi estruturado da seguinte forma:

* **Apresentacao/Program.cs**: O ponto de entrada do programa, onde o curso é informado e a execução do processo é iniciada.
* **Dominio/Entidades/Curso.cs**: Representa a estrutura do curso, com informações extraídas durante o scraping.
* **Dominio/Interfaces/Icourserepository.cs**: Define os métodos necessários para manipular os dados dos cursos no repositório.
* **Dominio/Interfaces/Icoursescraper.cs**: Define o contrato do scraper, que inclui métodos para buscar os cursos.
* **Dominio/Interfaces/Ilogservice.cs**: Define os métodos para o serviço de logs, como registrar erros, informações e advertências.
* **Dominio/Servicos/Courseservice.cs**: Coordena o fluxo de coleta e armazenamento de cursos.
* **Dominio/Servicos/Logservice.cs**: Implementa a interface de logs, gerenciando o armazenamento dos registros de execução.
* **Infraestrutura/Data/Aluracourses.accdb**: O banco de dados onde os dados dos cursos e logs são armazenados.
* **Infraestrutura/Repositorios/Databasecouserepository.cs**: Implementa o repositório de cursos, interagindo com o banco de dados para salvar e recuperar os dados dos cursos.
* **Injeçãodedependencia/Dependencyconfig.cs**: Configura a injeção de dependências, conectando as interfaces e suas implementações.
* **Rpa/Scraper/Seleniumcoursescraper.cs**: A implementação do scraper que usa o Selenium para acessar o site da Alura e coletar os dados dos cursos.