



**Nestlé** Good Food, Good Life

# Projeto Portal ADI On-Demand



## Sumário

1. Controle de Versão .....	3
2. Objetivo .....	3
3. Premissas para requisitos.....	3
4. Requisitos .....	4



## 1. Controle de Versão

Data	Autor	Versão	Nota de Revisão
27/04/2020	Hélio Paiva	1.0	Versão Inicial

## 2. Objetivo

- Desenvolvimento do portal ADI On-Demand onde os usuários possam iniciar pipeline de forma manual, e fazer upload de data sources dentro do Azure.

## 3. Premissas para requisitos

- Pipeline já criado no data factory,
- Diretório para upload do data source criado no storage account do azure,
- Caminho do diretório cadastrado na tabela tb\_datasource do banco amsbr-dhpr-sql-dqautomation



## 4. Requisitos

- **Controle de acesso**

Os acessos são controlados através do AD do Azure.

Dentro do método Page\_Load() da página default.aspx.cs, é feita a validação e criação da sessão do usuário que está logado.

- **Executar pipeline**

Para a execução do pipeline as informações devem estar cadastradas nas tabelas:

acesso\_ondemand;

perfil\_ondemand;

pipeline\_ondemand;

pipeline\_project\_datasource\_ondemand;

usuario\_ondemand, do banco de dados amsbr-dhpr-sql-dqautomation.

Observação: Ainda não existe a interface no portal para cadastro destas informações. Será inserido na próxima versão.

Para execução do pipeline são utilizados os métodos abaixo que estão dentro da página pipeline.aspx.cs.

- Page\_Load(): carrega as informações e oculta componentes executados na página.

- GetPipeline(): É executado assim que o usuário entra na tela pipeline.aspx e faz a consulta de forma automática na base de dados amsbr-dhpr-sql-dqautomation na view vwControleDeAcesso e traz os pipelines disponíveis para o usuário logado selecionar no dropdown da tela.

- ChamadaAPI(): Com base no pipeline selecionado, faz a chamada do pipeline para execução através da API e aguarda a resposta do servidor para apresentar a mensagem de pipeline em execução ou erro no processamento do pipeline.



- **Upload Data Source**

Para fazer o upload, são utilizados os métodos abaixo que estão dentro da página flatfile.apsx.cs.

- Page\_Load(): carrega as informações e oculta componentes executados na página de forma automática.
- ExecuteMetodos(): Este método é usado como controle de erro dos outros métodos.
- ValidaTemArquivo(): Faz a verificação se existe algum arquivo selecionado para ser feito o upload, este método retorna true para um arquivo selecionado e false para nenhum arquivo selecionado.
- ValidaExtensaoArquivo(): Valida se o arquivo selecionado tem as extensões .csv ou .txt, este método retorna true para arquivo selecionado .csv ou .txt e false para qualquer outro tipo de arquivo.
- ValidaDataSource(): Faz a validação se o nome do arquivo está cadastrado dentro da tabela tb\_datasource e se o usuário logado tem permissão para fazer o upload deste arquivo. Este método retorna true para arquivo cadastrado e usuário com permissão ou false para arquivo não cadastrado ou usuário sem permissão de upload para o arquivo selecionado.
- GetDataSource(): faz o upload do arquivo na camada bronze com base no caminho cadastrado na tabela tb\_datasource. Este método retorna true para caminho encontrado ou false para caminho não encontrado.