**Cd Passo a Passo – APP CONSISTE**

**CONSISTE + XTRBrasil + Instalações Consiste**

Procedimento Item 10

Trello – Organizador de tarefas

https://trello.com/b/i2zGkvcl/eng-de-dados

Interface gráfica do usuário, Texto, Site

Descrição gerada automaticamente

Revisão - coloco depois de falar com Daniel Assis – quando der algum problema que não conseguiu resolver.

To do – pega o projeto leva para seu nome (arrastar) definindo que este projeto será trabalhado por vc

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Arrastar quando terminar – Concluído

Ao entregar três informações

1. Qual é código operação a ser usado ED-004 de acordo com o instrução iT-ED-004
2. Necessário Reoperar – vai para operação ou vai fazer check list? OU Não necessário Reoperar
3. O que foi feito.

Ex:

* Instrução de trabalho usada: OP-004
* Necessário reoperar a carga: SIM
* O que foi feito no app: “informar o que foi feito”

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Exemplo: e734 Agentes Econômicos Fiscalizados pela ANP - DPC 10/02/2023

<https://gitlab.consiste.com.br/consiste>

Fluxo – Pegar GitLab o Programa pai

XTR Content Modelos – onde pego os dos Migradores

XTR Content – Pasta que coloco os códigos

**Seguir a Instrução de trabalho +**

**XTR Origem colocar em “Desenvolvimento” Guia - Geral**

<http://xtrdb01.consiste.com.br:9200/>

Formulário de CARGA: contém dados temporais errados - Não edita

* **Carregar os documentos que vem da CURADORIA:**

[\\thor.consiste.com.br\sys\migra\XTRBrasilCuradoria](file:///\\thor.consiste.com.br\sys\migra\XTRBrasilCuradoria)

Copiar o documentos CSV e Excel para pasta local onde o projeto será trabalhado

Criar uma pasta eXXXXArquivos e colar os arquivos

* **Abrir pasta para iniciar o projeto + clonar modelo do GITLAB:**

**OBS: sempre do XTR content**

**Se tiver que criar do começo clonar:**

[**https://gitlab.consiste.com.br/consiste/xtr-content-modelos**](https://gitlab.consiste.com.br/consiste/xtr-content-modelos)

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

1. **Em - Consiste Consultoria e Sistemas Ltda – Grupo do GITLAB entre na pasta XTR Content Modelos – selecione o modelo de acordo com o trabalho a ser realizado – Nome da Entidade**

Detectando que se trata do ED 004 – DOCKER

|  |  |
| --- | --- |
|  | Conjunto de pastas que identificam ser uma entidade de PYTHON/DOCKER – ED 004 |

1. Clonar modelo (no gitlab escolha o migrador – entre – clone - clone with SSH)
2. Abrir o Git Bash na pasta do arquivo e clonar o modelo para máquina local.
3. COMANDO – git clone “nome do arquivo modelo CLONADO”

* **Pra Mudar Branch para develop – git bash na pasta do arquivo**

1. Cd “nome da pasta do arquivo”
2. Git checkout develop

* **Excluir a pasta .git – oculta**

1. No Explore – VIEW (barra principal) – Show – Hidden Items
2. Irá aparecer o arquivo .GIT
3. Deletar o arquivo .git – com isso vc quebra a Branch com o GITLAB

* **Iniciar ajustes do APP**
  + Pasta Cycle-controller – editar os aquivos que vieram do modelo: e yml - pad++

1. -play.sh
   * + indice eXXXX – colocar o nome do app
2. yml
   * + Editar o nome do arquivo – eXXX – colocar o nome do APP

**Ajustes pasta files**

OBS: verificar se há dados mais atualizados na base de origem – MANUAL PARA EXTRAÇÃO DEDADOS

1. Backup – Preparar um histórico do arquivo original
   1. Cole o arquivo original dos dados
   2. Para não ocupar muito espaço selecione todas as linhas EXCETO as 10 primeiras e apague - fazer o um backup das 10 primeiras linhas (apagar o restante) – para ter uma memória de como estava no início o arquivo
2. dataIngestorAssets – sem ação ...\files\dataIngestorAssets\log
3. dataNormalizadorAssets – Apagar todos os arquivos dessa pasta, O PYTHON irá gerar um arquivo .CSV após ser executado. (PODE NÃO ESTAR PRESENTE???)
4. Input
   1. Arquivos JSON
      1. Metadata.json – retirar dados do formulário de carga:

Exemplo:

"linkorigem": "https://www.gov.br/anp/pt-br/...-abertos",

"origem": "ANP",

"meta\_qtd\_e734": 1

* + 1. Controller.json – "elasticSearchServer" – deve ser chaveado com "REVIEW" ou “PRODUCTION”

"bulksize": 100 - mostra a quantidade de linhas que via ser carregada por vez – testar de 100 a 1000, de acordo com o tamanho do arquivo.

* + 1. fieldsByTyoe.json - ver NORMALIZADOR (Python)

Deve ser copiado todas os nomes das colunas e classificar os tipos que podem ser:

CEP, CNPJ, CONTROLE,CPF,DATA, MUNICIPIO, NUMERO, TEXTO, UF

***OBS: Números que não serão utilizados para algum tipo de cálculo devem ser considerados “TEXTO”*** -

* + 1. Input.json

Carga – ver no origem – entidade big – número de carga (ver o que não está em produção)

Entidade: eXXXX

Periodo – Referente ao período dos dados que estão sento adicionados.

Encoding:

1. Source – colocar o arquivo .CSV / .XLXS original na pasta .. files\input\source
2. Pasta Ingestor – Importar para o eclipse (JAVA)
   1. Editados os comandos para execução do APP

businessRules

XTROrigensTypes.java

CEP, CNPJ, CONTROLE,CPF,DATA, MUNICIPIO, NUMERO, TEXTO, UF

tratarCamposDoTipoTEXTO(docOut);

tratarCamposDoTipoCEP(docOut);

tratarCamposDoTipoLongitude(docOut);

tratarCamposDoTipoLatitude(docOut);

tratarCamposDoTipoCPF(docOut);

tratarCamposDoTipoCNPJ(docOut);

criarCampoGeoHash(docOut);

tratarCamposDoTipoDATA(docOut);

tratarCamposDoTipoEMAIL(docOut);

tratarCamposDoTipoNUMERO(docOut);

tratarCamposDoTipoString(docOut);

consultaMuni(docOut);

1. Pasta NORMALIZADOR (Python)
   * + Abra o arquivo main.py – abrir no VSCODE para fazer edições necessárias, Ajustes das colunas, etc
     + Rode o debug e execute para fazer as atualizações no arquivo. Isso vai atualizar ou gerar um arquivo .CSV na pasta “dataNormalizadorAssets”.
     + Verifique na pasta:  se o arquivo foi atualizado, ver hora/data de atualização do arquivo.

**OBS: pode ser necessário apagar o arquivo na pasta “dataNormalizadosAssets” para evitar bugs.**

Verificar as COLUNAS – Base Vs Python Vs fildsByType

Pegar as colunas e associar conforme o que está no XTR Entidades BIG – Chamar o programa e verificar se as colunas dos dados fontes são iguais para parear conforme o que se encontra na entidade – Lapis (editar) - Guia

    colunasNovas = {'UF': 'uf',

                    'Município': 'municipio',

                    'Bairro': 'bairro',

                    'Endereço': 'endereco',

                    'CNPJ/CPF': 'cnpjoucpf',

                    'Agente Econômico': 'agente\_economico',

                    'Segmento Fiscalizado': 'segmento\_fiscalizado',

                    'Data DF': 'data\_fiscalizacao',

                    'Procedimento de Fiscalização': 'procedimento\_fiscalizacao',

                    'Resultado': 'resultado',

                    'Número do Documento': 'numero\_documento',

                    }

**Exemplo de Erro que pode ser encontrado:**

**Uma imagem contendo Texto

Descrição gerada automaticamente**

**Ok**

**aJUSTAR**

**checar os tipos de dados da planilha no PYTHON**

types=df1.dtypes

print(types)

exit(0)

**Elastic**

<http://xtrdb01.consiste.com.br:5601/s/dev-e-manager/app/dev_tools#/console>

Tela de celular com aplicativo aberto

Descrição gerada automaticamente

Para fazer os ajustes ou base nova verificar o que existe no Elastic sobre a entidade, com o seguinte comando: (eXXXX\_cX = numero e versão da entidade).

***GET eXXXX\_cX /\_search***

Caso a entidade já tenha dados e seja necessário ajustes no Banco de Dados de teste (Elastic) delete os dados para nova atualização.

***DELETE eXXXX\_Cx***

Outros Comandos:

Contar dados:

***GET eXXXX\_cX/\_count***

***{***

***"query": {***

***"match": {***

***"periodo": "2020"***

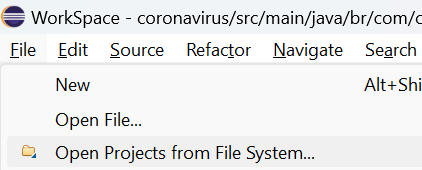
***}***

***}***

***}***

***Por ser ED-004***

Uma vez apagado os dados no BD no Elastic entre no ECLIPCE IDE para fazer os ajustes necessários no código:



Editar app no JAVA

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

O que deve ser verificado está no Atualizadorpadrão & DataTranspoter.java

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Se for substitutiva refazer o mapping:

* Todos como float deve ser double
* Refazer o mapping
* D:\CONSISTE\WorkSpace\Projetos\produtoras-de-biocombustiveis-autorizadas-pela-anp\ingestor\src\main\java\br\com\consiste\atualizadorpadrao\controller
* IndexMapping.criarEntidade(fieldsByType, "http://" + elasticSearchServer.getServerName() + ":9200/", index, camposLong);

Texto

Descrição gerada automaticamente

* Map<String, XTROrigemTypes> fieldsByType = scheme.getFieldsByType();
* Uma imagem contendo Interface gráfica do usuário

  Descrição gerada automaticamente

*private List<String> camposLong = new ArrayList();*

*camposLong = Arrays.asList(new String[]{"campoInteiro"}); // declare seus campos inteiros aqui*

*import br.com.consiste.atualizadorpadrao.model.businessRules.XTROrigemTypes;*

*Map<String, XTROrigemTypes> fieldsByType = scheme.getFieldsByType();*

***Fazer o MAPPING***

*IndexMapping.criarEntidade(fieldsByType, "http://" + elasticSearchServer.getServerName() + ":9200/", index, camposLong);*

Texto

Descrição gerada automaticamente

1. public class Metadata {
2. private Gson gsonTool = new Gson();
4. private final String metadataFormPath = "../files/input/metadata.json";
5. public static String thisYear = CalendarioHoje.getThisYear();
6. public static String thisMonth = CalendarioHoje.getThisMonth();
7. private FileSupport fileSupport = new FileUtilities();
8. public Map<String, Object> customMetadata = new LinkedHashMap<>();
9. public String reader;
11. public Metadata() {
12. reader = fileSupport.readFrom(metadataFormPath);
13. Type typeOfT = new TypeToken<Map<String, Object>>() {}.getType();
14. this.customMetadata = gsonTool.fromJson(reader, typeOfT);
15. customMetadata.keySet().forEach((key) -> {
16. if(customMetadata.get(key).getClass().getSimpleName().equals("Double")){
17. Double teste = (double) customMetadata.get(key);
18. customMetadata.put(key, teste.intValue());
19. }
20. });
21. }
23. public void include(Map<String,Object> docOut) {
24. docOut.put("ano", thisYear);
25. docOut.put("mes", thisMonth);
26. docOut.putAll(customMetadata);
27. }
28. }

Docker

Último teste para entrega

Texto

Descrição gerada automaticamente

Guit Bash na pasta do main.py

Tela de celular com aplicativo aberto

Descrição gerada automaticamente com confiança baixa

Vai gerar os aquivos Dockers

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Testar se está tudo ok:

Tem que olhar para o do banco de dados de teste, temos que mandar ele olhar.

Portafinal sempre 9200

<http://xtrdb01.consiste.com.br:9200/>

salvar

verificar

apagar

<http://xtrdb01.consiste.com.br:9200/>

salvar

Bando de dados teste

Sempre apagar depois do teste acima

<http://xtr-verde.consiste.com.br:5601/app/dev_tools#/console>

para ver no DB de produção: GET e720\_c4/\_search

Apagar os dados preservando a entidade:

POST e720\_c4/\_delete\_by\_query

{

"query":{

"match\_all":{

}

}

}

Contar item do document:

GET e720\_c4/\_count

Texto

Descrição gerada automaticamente

E+tab

Docker – cria um container semelhante a maquina virtual que executa o python e depois cria o java e testa.

**ENTREGA**

Apagar dados dentro das pastas do dataIngestorAssets & dataNormalizadorAssets

Copiar a pasta. Exemplo e734

Verificação de integridade:

1. Agregar por:
   1. Municipio
   2. Por ano
   3. Ordenar colunas
   4. Localidade (Quando houver)
   5. Grafias alternativas (Localilidade)

Apagara a pasta antiga e salvar a nova

[\\thor.consiste.com.br\sys\migra\XTRBrasilOperacoes](file:///\\thor.consiste.com.br\sys\migra\XTRBrasilOperacoes\e734\files\input)

Ir para:

Colocar os arquivos trabalhados no servidor XTRBrasilOperações

Controller.json

Mudar - "elasticSearchServer": "PRODUCTION",

Trello

SELCIONAR a entidade adicionar o que foi feito:

* -m

*VERSIONAMENTO*

1. Git lab selecionar o local onde será colocada a entidade pronta.

<https://gitlab.consiste.cogitm.br/consiste>

1. em Search by Name – coloque no nome da entidade para encontrar o local para fazer o versionamento

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

1. Click na Entidade e faça - Clone with SSH
2. No CPU local iniciar o processo de vercionamento
   1. git clone [git@gitlab.consiste.com.br:consiste/xtr-content/agentes-economicos-fiscalizados-pela-anp.git](mailto:git@gitlab.consiste.com.br:consiste/xtr-content/agentes-economicos-fiscalizados-pela-anp.git)

Git clone “CODIGO CLONADO”

* 1. Entre na pasta criada

Cd “nome da pasta clonada”

* 1. $ git checkout develop

Mudar a Branch para develop

* 1. Jogar os documentos da entidade para dentro do novo arquivo pelo explorer e - $ git status

Checar os documentos a serem commitados

* 1. $ git add .

Adiciona todos os documentos ao Git

* 1. $ git commit -m "agente-economicos-fiscalizados-pela-anp"

Commita todos os documentos adicionados

g

$ git push

Empurra os documentos para o GITLAB

1. No GITLAB verificar se os documentos foram adicionados.
2. Interface gráfica do usuário, Texto, Aplicativo, Email

   Descrição gerada automaticamente

**AJUSTAR (quando já iniciado) - APP CONSISTE**

Formulário de CARGA: contém dados temporais errados - Não edita

* **Carregar os documentos que vem da CURADORIA:**

[\\thor.consiste.com.br\sys\migra\XTRBrasilCuradoria](file:///\\thor.consiste.com.br\sys\migra\XTRBrasilCuradoria)

Copiar o documentos CSV e Excel para pasta local onde o projeto será trabalhado

* Criar uma pasta eXXXXArquivos

Ir no Gitlab Consiste - <https://gitlab.consiste.com.br/consiste>

No Search by name achar o APP pelo nome – Switch branch/tag - Develop

Git bash

Ver *VERSIONAMENTO*

*Texto

Descrição gerada automaticamente*

Copia o caminho do ingestor

Abrir Eclipse IDE for Java

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

Colocar o endereço copiado

Editar app no JAVA

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Verificar

DataTransporter.java

Linha 247

Texto

Descrição gerada automaticamente

No caso criar novo campo com os 8 primeiros números do cnpj = cnpj\_basico

Linha 632

Texto

Descrição gerada automaticamente

**Elastic**

<http://xtrdb01.consiste.com.br:5601/s/dev-e-manager/app/dev_tools#/console>

verificar ser as alterações estão corretas.

No arquivo

...\files\dataNormalizadorAssets

* Deletar .CSV file

...\normalizador

* Abrir : Main.py no VSCODE

Para testar editar o as variáveis Path & output.

Exemplo:

Copiar o Path & output (shift + alt + “seta para baixo”)

Copiar os caminhos no disco local;

Ajustar as barras selecionando todas e alterando as barras invertidas (Ctrl + D)

path = "D:/CONSISTE/WorkSpace/Projetos/produtoras-de-biocombustiveis-autorizadas-pela-anp/files/input/source/"

output = "D:/CONSISTE/WorkSpace/Projetos/produtoras-de-biocombustiveis-autorizadas-pela-anp/files/dataNormalizadorAssets/"

# path = "../files/input/source/"

# output = "../files/dataNormalizadorAssets/"

Alternativa ao <http://xtrdb01.consiste.com.br:5601/s/dev-e-manager/app/dev_tools#/console>

**POSTMAN**

GET

GET Status + [url:porta(9200)](file:///D:\CONSISTE\IT\Instru--oTrabaho\porta(9200))/eXXX.cX

Deletar

Del index + [url:9200/eXXX.cX](file:///D:\CONSISTE\IT\Instru--oTrabaho\9200\eXXX.cX)

Short Cut

Methods -> Ctrl + space

**ArrayofData = line,toString().split(“@”);**

fieldsByTyoe.json

Identar Ctrl shift F