

**PADRÃO DE DESENVOLVIMENTO**

**(Plataforma Talend\_Open)**

Este documento tem como objetivo informar os padrões de desenvolvimento na plataforma Talend Open estabelecidos para a unidade Honda HDA

|  |  |
| --- | --- |
| APROVADOR | ELABORADOR |
|  |  |
| Claudio Spindola | Christyan Mendonça |

Manaus, 15 de março de 2025

Índice

**ASSUNTO PÁG**

[1 INTRODUÇÃO 3](#_Toc194992354)

[2 NOMENCLATURA PADRÃO DE PROGRAMAS: 3](#_Toc194992355)

[3 GESTÃO DOS PROGRAMAS TALEND OPEN NA HDA: 3](#_Toc194992356)

[4 REPOSITÓRIO DE FONTES: 4](#_Toc194992357)

[5 FUNCIONALIDADES DE CONTROLE: 4](#_Toc194992358)

[6 FREQUENCIA DE EXECUÇÃO: 5](#_Toc194992359)

[7 SERVIDOR EXECUÇÃO DE JOBS: 5](#_Toc194992360)

[8 SERVIDOR BASE DE DADOS (CONTROLE): 5](#_Toc194992361)

[9 MONITORAMENTO DA EXECUÇÃO DE JOB’s: 6](#_Toc194992362)

[10 ARQUIVO DE LOG TXT: 7](#_Toc194992363)

[11 AVISOS POR EMAIL: 7](#_Toc194992364)

[12 DESIGN PATTERNS: 8](#_Toc194992365)

[13 SEQUÊNCIA DE ATIVIDADES DO JOB DE CONTROLE: 8](#_Toc194992366)

[13.1 Inicialização + Executar Rotinas com arquivo de log txt do Job: 9](#_Toc194992367)

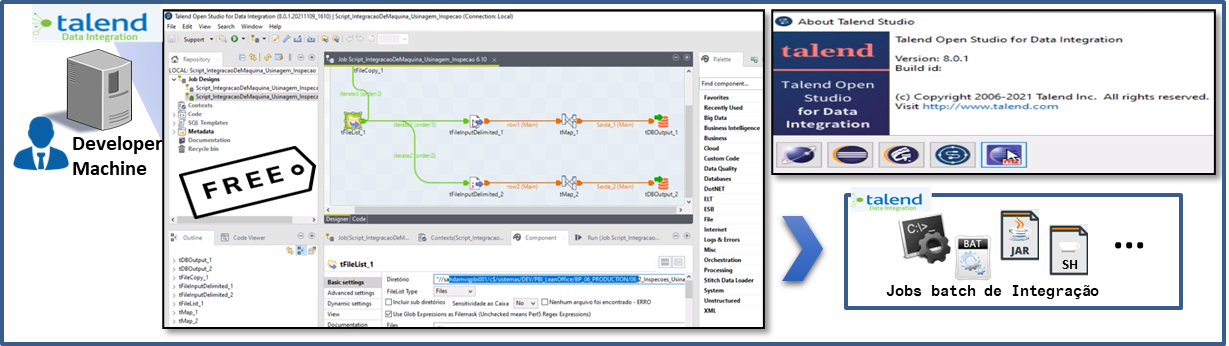
[13.2 Executar rotinas de Verificação da tabela de controle: 9](#_Toc194992368)

[13.3 Executar rotinas das regras de negócio: 10](#_Toc194992369)

[13.4 Finalização + Atualização da tabela de controle: 12](#_Toc194992370)

# INTRODUÇÃO

Este documento tem como objetivo apresentar os padrões de desenvolvimento na plataforma Talend Open estabelecidos na HDA. A versão do Talend Opend da HDA pode ser visualizada a seguir:



*Figura 1 – Versão do Talend Open da HDA*

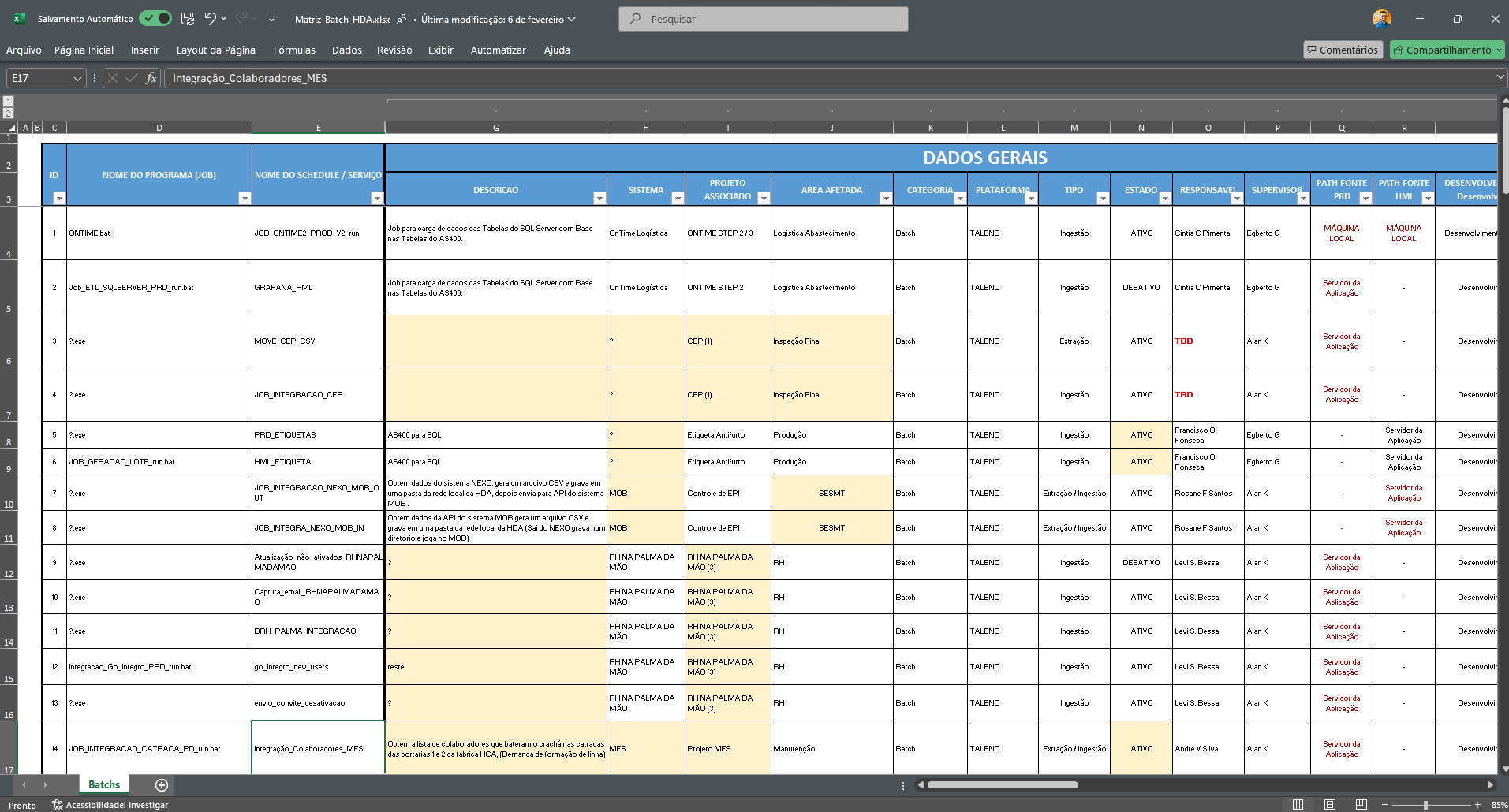
# NOMENCLATURA PADRÃO DE PROGRAMAS:

*<‘****job\_****’> + <* ***até 8 iniciais identificadores do depto ou setor ou agrupamento de businnes area ou assunto de interesse****> + ‘****\_****’ + <* ***até 10 iniciais identificadores do assunto do ETL >*** *+ ‘****\_****’ + <****numeração de 001 a 999****>*

***Exemplo Executável:*** *job\_COALIHDA\_1.bat*

# GESTÃO DOS PROGRAMAS TALEND OPEN NA HDA:

A fim de garantir o controle do ambiente de execução dos programas Talend Open da HDA, é obrigatório que todos eles, desenvolvidos e em desenvolvimento, sejam registrados na planilha [Matriz\_Batch\_HDA.xlsx](https://globalhonda.sharepoint.com/:x:/r/sites/sahda-mao218/Shared%20Documents/Arquitetura/2.%20Projetos/101KI/Estruturacao_Talend/Matriz_Batch_HDA.xlsx?d=w127bbff9d3ac4b348f75d4760dee75fe&csf=1&web=1&e=MLghDu), sendo inclusive esse fator pré-requisito para aprovação de novas implantações nas reuniões de CAB.

****

*Figura 2 – Planilha de Controle de Jobs Talend da HDA*

# REPOSITÓRIO DE FONTES:

O código fonte dos Jobs desenvolvidos na plataforma Talend Open pela HDA devem ser armazenados nos folders GPWORK e GPPROD do servidor SAHDAMVPAPPS017, que inclusive ja é utilizado na HDA para armazenar fontes de outras plataformas de desenvolvimento de software.

# FUNCIONALIDADES DE CONTROLE:

Além das rotinas core que serão desenvolvidas para atender as regras de negócio da solução do projeto, todo job Talend Open deve implementar rotinas de controle da execução. Essas rotinas devem sempre observar as funcionalidades de controle a seguir:

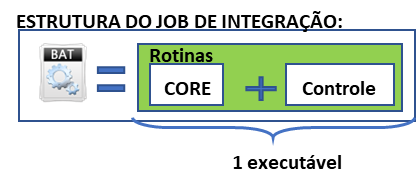
. Parametrização das Configurações do JOB (ativação da execução, envio de alertas, deleção de dados, etc);

. Envio de alertas por e-mail;

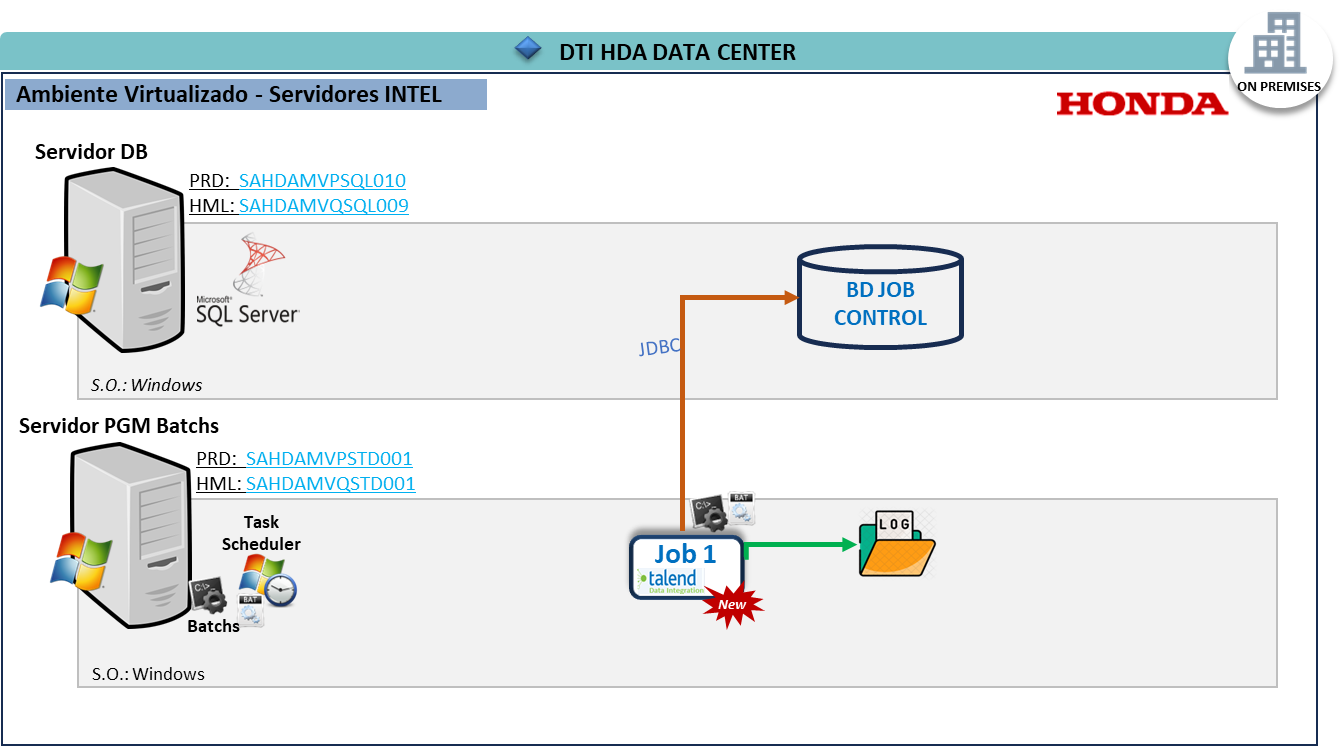
. Tratamento de falhas de execução (Tentar executar novamente);

. Capturar erros para Geração de Log de execução em arquivo txt;

. Autonomia na frequência e nos intervalos de execução;



A arquitetura básica da solução dos jobs talend pode ser observada a seguir:



*Figura 3 – Arquitetura dos Jobs Talend Open*

# FREQUENCIA DE EXECUÇÃO:

a) Resumo: O job desenvolvido na plataforma Talend Open deve ser autônomo para controlar a frequência de execução. Ele será desencadeado a primeira vez pelo Task Scheduler do Windows, mas depois disso, ele se autogerencia, sem depender de componentes externos para executar suas rotinas em intervalos regulares, parametrizados em uma tabela de controle SQL Server. O Objetivo disso é se obter uma execução independe de um serviço externo (Task Scheduler), podendo ser controlada via parâmetros cadastrados no banco de dados de controle. Essa abordagem já é homologada em outras plataformas de desenvolvimento de Jobs. Exemplo do algoritmo de controle de intervalo de execução:

INÍCIO

// Inicializa Variaveis

&FLGLOOPINFINITO = 'S' // Ctrl Loop Infinito

&TempoParaAguardar = 30 // 30 segundos default

&FlagAtivo = 'S'

DO WHILE &FLGLOOPINFINITO = 'S' // Inicializa LOOP INFINITO

DO '1) RotinasJob1()'

DO '2) AguardaProximaExecucao(&TempoParaAguardar)' // Executa Sleep

ENDDO

FIM

b) Abordagem de exceção: dividir o job de controle (em execução contínua com sleep) e job de regras de negócio: O job de controle deve ser desencadeado pelo Task Scheduler do Windows, como planejado anteriormente, ou por outro mecanismo de agendamento interno se disponível.

c) Extra fluxos: Em alguns casos específicos de erro, o job se autoajusta para executar novamente após um Intervalo de 1 min (fixo); porém, quando necessário, deve ser possível também fazer a Execução Manual diretamente no servidor;

# SERVIDOR EXECUÇÃO DE JOBS:

. SAHDAMVQSTD001 (HML) . SAHDAMVPSTD001 (PRD)

. Usuário de Execução do Job: Seu projeto deve possuir um usuário de rede próprio para executar os Jobs nesse servidor que não seja compartilhado com outras aplicações/projetos. Abra um chamado para criação desse usuário.

. Estrutura de Pastas no Servidor HML e PRD:

.PathInstalacao => 'd:\job\_batch\<PROJETO>\xxx.bat'; | Exemplo 'd:\job\_batch\<PROJETO>\job\_xxx.exe'

. PathLogs => 'd:\job\_batch\<PROJETO>\log\xxx.txt'; | Exemplo d:\job\_batch\<PROJETO>\log\20220826\_job\_xxx.txt';

. PathCarga => 'd:\job\_batch\<PROJETO>\carga\xxx.xls'; | Exemplo d:\job\_batch\<PROJETO>\carga\xxxx.xls';

# SERVIDOR BASE DE DADOS (CONTROLE):

. Hostname HML: SAHDAMVQSQL009 . IP: 10.146.2.19

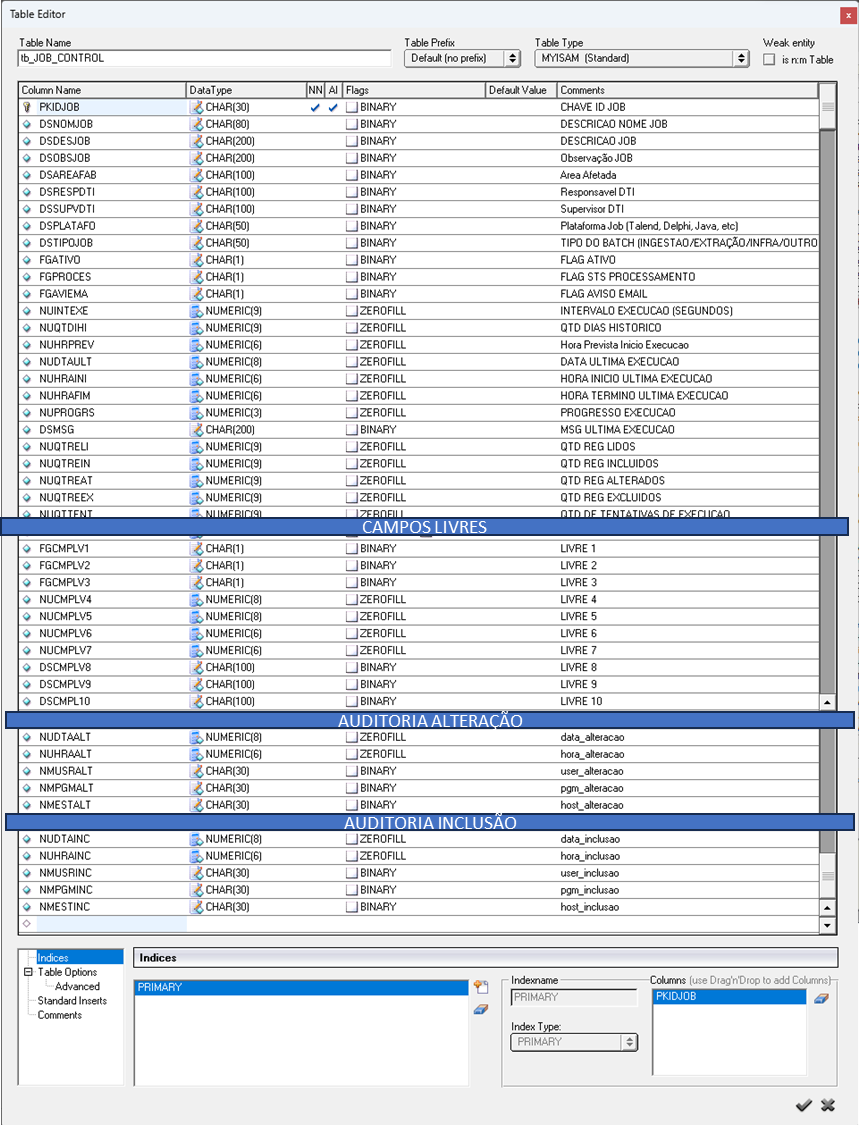
. Hostname PRD: SAHDAMVPSQL010 . IP: 10.146.2.27

. Usuário de acesso ao banco de dados: Seu projeto deve possuir um usuário de banco próprio para acessar essa base de dados que não seja compartilhado com outras aplicações. Abra um chamado para criação desse usuário.

. Formato de DATA e Hora a ser considerado para campos de auditoria de inclusão, alteração e envio:

. DATA: YYYYMMDD . HORA: hhmmss

. Estrutura da Tabela de Controle de Jobs:

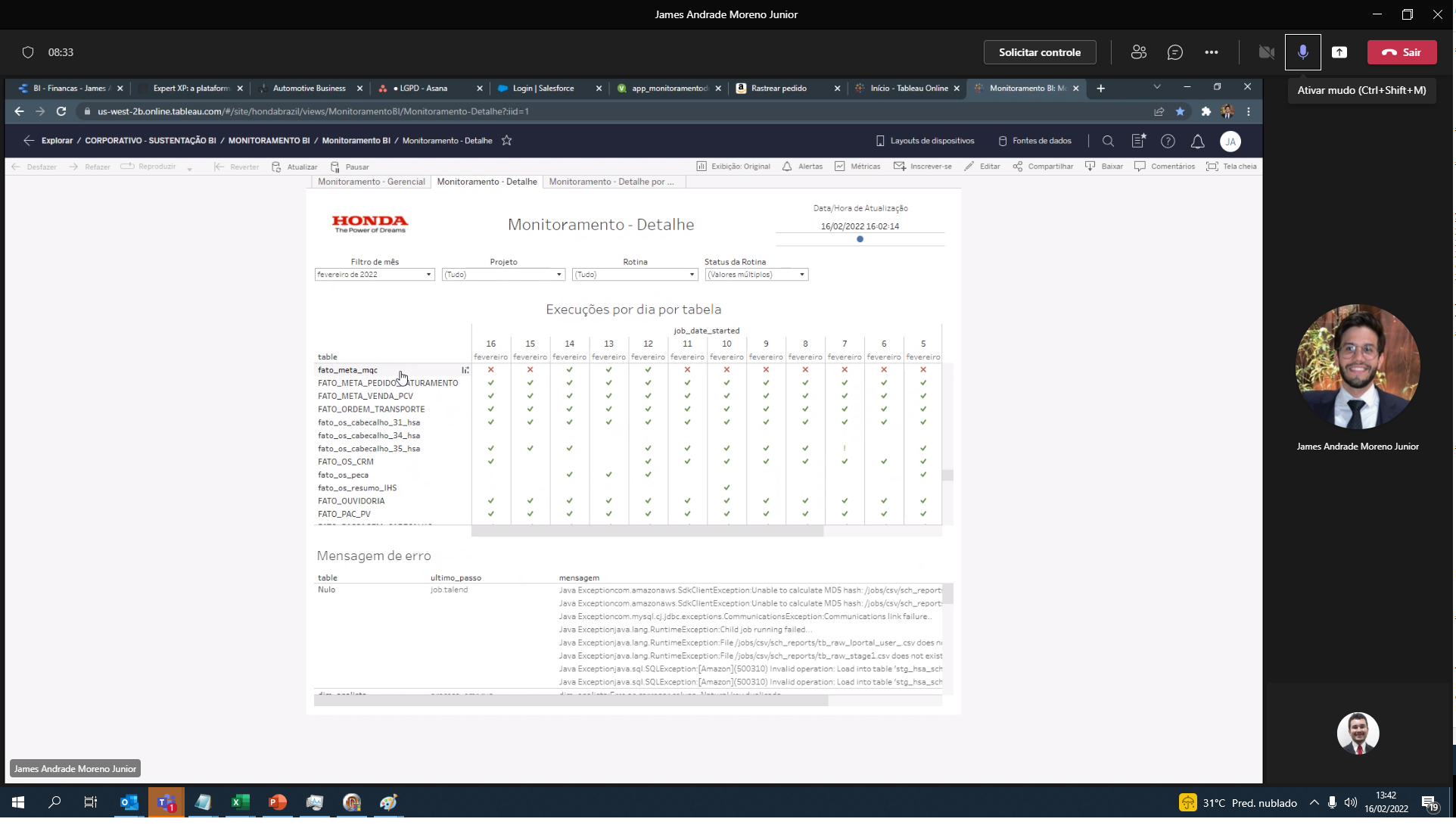


*Figura 4 – Estrutura da Tabela de Controle*

# MONITORAMENTO DA EXECUÇÃO DE JOB’s:

Cada Job que estiver cadastrado na tabela de controle terá sua execução monitorada por meio de uma tela de observabilidade (a ser desenvolvida). Com isso, obteremos alguns benefícios como:

|  |
| --- |
| **Governança dos Job’s (Cadastro por projeto, processo, categoria, ordem de execução, outros);** |
| **Trata falhas de execução do Job (Envio de alertas, tentar executar novamente, etc);** |
| **Controles e configurações do Job (ativação da execução, do envio de alertas, deleção de dados, etc);** |



*Figura 5 – Exemplo de Tela de Monitoramento da Execução de Jobs*

# ARQUIVO DE LOG TXT:

- Gravar um arquivo de Log no path de log do servidor sempre que houver:

. Inicio de cada execução do job;

. Termino de cada execução job;

. Erro encontrado em cada execução;

. Reagendamento (Tentativa depois de um intervalo x)

- Nome do Arquivo de Log deve ser de acordo com o dia e o nome do job atual:

. Nome => 'log\_' + '<Nome\_do\_Job\_Atual>\_' + YYYYMMDD + '.txt'

- Escrita de cada Linha do Log:

. DD/MM/AAAA - HH:MM:SS - <TIPO (ERRO/AVISO/SUCESSO)> - <Texto do Log>

- Arquivos txt devem ser deletados de acordo com a quantidade de dias definida na tabela de controle, campo NUQTDIHI: Obtém data atual e subtrai da qtd de dias pre-configuradas na tabela de controle (regra abaixo), ou 90d por default;

# AVISOS POR EMAIL:

6.1. Definir Email para envio de msg's de alertas. Exemplo: SA HDA\_MAO\_INTEGRACAOROTA <hdamaointegracaorotarh@honda.com.br>

# DESIGN PATTERNS:

O desenvolvimento na plataforma Talend Open praticado na HDA deve seguir boas práticas de design patterns:

a) Commit Interval: Configure intervalos de commit apropriados nos componentes de BD (tDBOutput). Um intervalo de commit maior pode reduzir a frequência de commits e, consequentemente, o número de transações;

b) Adicione componentes de logging e monitoramento para rastrear a execução dos jobs e identificar problemas rapidamente. Cada erro no Job deve ser capturado e escrito no Log txt da aplicação. De preferência utilizar componentes tLogCatcher e tStatCatcher para monitorar e registrar a performance e logs dos Jobs;

c) Usar Conexões Persistentes para evitar a sobrecarga de abrir e fechar conexões de banco de dados repetidamente. Isso pode ser feito configurando a propriedade Use Existing Connection nos componentes de BD;

d) Sempre fechar transações abertas: Manter transações abertas por longos períodos pode causar um acúmulo no log de transações, pois o SQL Server precisa registrar todas as alterações até que a transação seja confirmada;

e) Dividir transações longas em transações menores: Divida o job em transações menores para permitir que o log de transações seja truncado com mais frequência. Se o job não for dividido em transações menores, ele pode manter transações abertas por longos períodos, impedindo que o log de transações seja truncado;

f) Padrão de Job Simples: Crie jobs simples e modulares que realizem tarefas específicas. Isso facilita a manutenção e a reutilização de componentes.

g) Padrão de Job Pai/Filho: Utilize jobs pai para orquestrar a execução de jobs filhos. Isso permite uma melhor organização e controle dos processos.

h) Reduzir o número de transações individuais:

. Sempre usar inclusão / alteração / Deleção em massa, ao invés de transações registro a registro. Utilizar componentes de carregamento em massa (bulk loaders) disponíveis no Talend, como tBulkExec para BD’s específicos, reduzindo o número de transações individuais. No entanto, atentar para as configurações adequadas para evitar sobrecarga no BD: Uso inadequado de operações de bulk pode gerar grandes volumes de log se não forem configuradas corretamente;

. Sempre usar comando truncate ao invés do delete quando for necessário limpar tabelas;

. Realizar o máximo de transformações possíveis na memória antes de carregar os dados no BD. Utilizar tMap antes de inserir dados no destino, porém faça Otimização do tMap: Configure o tMap para armazenar dados temporários no disco em vez de na memória. Isso pode ser feito ativando a opção "Store temp data" no tMap1.

. Batch Processing: Configure o processamento em lote para agrupar operações de inserção, atualização ou exclusão. Isso pode ser feito utilizando componentes como tBatchRow para agrupar registros antes de enviá-los para o banco de dados;

# SEQUÊNCIA DE ATIVIDADES DO JOB DE CONTROLE:

Em linhas gerais podemos dividir a sequência de atividades dos Jobs nessas 4 sub-rotinas, que serão detalhadas a seguir:

a) Inicialização + Executar Rotinas com arquivo de log txt do Job;

b) Executar rotinas de Verificação da tabela de controle;

c) Executar rotinas das regras de negócio;

d) Finalização + Atualização da tabela de controle e Arquivos de Log txt do Job;

## Inicialização + Executar Rotinas com arquivo de log txt do Job:

a) O Arquivo txt de Log do dia deve ser criado se não existir:

Nome => '<Nome\_do\_Job\_Atual>\_' + YYYYMMDD + '.txt'

b) Escreve no arquivo de Log txt uma mensagem de "SUCESSO": <Nome\_do\_Job\_Atual> Inicializado;

c) Os arquivos txt de Log que forem antigos devem ser deletados. Para saber se um arquivo é considerado antigo, obter o valor do campo NUQTDIHI (QTD DIAS HISTORICO) da tabela de controle. Depois obtém data atual e subtrai do valor obtido; então, delete todos os arquivos com datas anteriores ao resultado obtido nessa subtração; Caso o valor do campo NUQTDHI não esteja preenchido, considerar 90 dias fixo (por default);

## Executar rotinas de Verificação da tabela de controle:

a) Executa uma consulta na BD de Controle, na tabela 'tbjobetl' where o campo 'PKIDJOB' = <Nome\_do\_Job\_Atual>'. Em seguida, obtém o valor do campo NUINTVEX e atualiza no intervalo de execução do Job no próprio Job (armazena em memória para uso no comando sleep posteriormente). Além desse campo, obtém também os valores dos campos NUQTDIHI (QTD DIAS HISTORICO DO LOG), FGAVIEMA (FLAG AVISO EMAIL) e armazena na memória. O campo NUQTDIHI (QTD DIAS HISTORICO) deve ser usado nas próximas execuções para limpeza do arquivo de log txt. Depois, verifica se o campo FGATIVO = 'S':

. Caso positivo, continua o Job (vai para o passo b desse item 9.2);

. Caso negativo, finaliza o Job: Escreve no arquivo de Log txt um "AVISO": Finalizando processamento pois o Job está Desativado; aguarda tempo de intervalo configurado para o job nos passos anteriores; reinicia o passo 9.1;

b) Executa uma nova consulta na BD de Controle, na tabela 'tbjobetl' where o campo 'PKIDJOB' = <Nome\_do\_Job\_2>. Realiza a seguinte validação com o valor do campo FGPROCES:

**Se** o campo FGPROCES = 'N' então:

Executa o update a seguir na tabela 'tbjobetl' alterando os campos a seguir where o campo 'PKIDJOB' = <Nome\_do\_Job\_Atual>:

*update*

*tbjobetl*

*set*

*FGPROCES = 'S', NUDTAULT = Data atual, NUHRAINI = hora atual, NUHRAFIM = 0, NUPROGRS = 0, DSMSG = 'INICANDO PROCESSAMENTO', NUQTRELI =0, NUQTREIN =0, NUQTREAT =0, NUQTREEX =0, NUQTTENT =0, NUDTAALT = data atual, NUHRAALT = hora atual, NMPRGALT = Nome do Job Talend, NMUSRALT = Nome do Usuário do Job Talend, NMESTALT = Nome da Máquina Onde o Job Talend Está Executando*

*where*

*PKIDJOB = <Nome\_do\_Job\_Atual>*

. Passa para o passo 9.3;

**Senão**:

Executa o update a seguir na tabela 'tbjobetl' alterando os campos a seguir where o campo 'PKIDJOB' = <Nome\_do\_Job\_Atual>:

*update*

*tbjobetl*

*set*

*DSMSG = 'AGUARDANDO FIM PROCESSAMENTO JOB 2', NUQTTENT +=1, NUDTAALT = data atual, NUHRAALT = hora atual, NMPRGALT = Nome do Job Talend, NMUSRALT = Nome do Usuário do Job Talend, NMESTALT = Nome da Maquina Onde o Job Talend Está Executando*

*Where*

*PKIDJOB = <Nome\_do\_Job\_Atual>*

. Em seguida, se o campo Flag Aviso Email do Job está preenchido com 'S' e NUQTTENT in (10, 20, 30), então, envia Email para o endereço do grupo supracitado, informando a qtd de tentativas de execução do Job ocorridas sem sucesso. Depois, escreve no arquivo txt de Log um "AVISO": Tentativa nº X. Aguardando fim do processamento do Job 2. Não é possível executar agora. Entrando em modo de espera (1min); então aguarda 1 min (Fixo) e depois reinicia o passo 9.1;

## Executar rotinas das regras de negócio:

. Nesse ponto são implementadas as lógicas do programa que atenderam a regra de negócio da solução;

. **Obs1**: Antes de carregar dados em fontes de dados, iniciar o controle de transação para possibilitar o rollback em caso de erros.

. **Obs2**: Não deve haver registros com valores nulos nas colunas de qualquer tbl das BDs: Os campos das tabelas que não existirem nas consultas realizadas devem ser preenchidos com '' ou 0 durante a execução da integração;

. **Obs3**: Os campos de auditoria não podem possuir valores nulos. Se for realizar uma operação de inclusão no banco, preencher os campos de auditoria inclusão com os valores reais e os campos de auditoria de alteração com 0 e ''. Senão, se for realizar uma operação de atualização no banco, preencher os campos de auditoria alteração com os valores reais, mas, não alterar os campos de auditoria de inclusão.

. **Obs4**: Para cada marco de processo realizado pelo Job (ex: tabela atualizada/carregada, etc), realizar as seguintes operações:

a) atualizar os campos na tabela de controle 'tbjobetl', alterando os campos a seguir where o campo 'PKIDJOB' = <Nome\_do\_Job\_Atual>:

update tbjobetl set

NUPROGRS = (100\*NRO\_TABELA\_ATUAL) / 5, -- (Cálculo do progresso % de acordo com o progresso do marco), DSMSG = 'TBL X de 5 CARREGADA - <NOME DA TABELA>', NUQTRELI = NUQTRELI + QTD REG LIDOS NA TABELA ATUAL, NUQTREIN = NUQTREIN + QTD REG INCLUIDOS NA TABELA ATUAL, NUQTREAT = NUQTREAT + QTD REG ALTERADOS NA TABELA ATUAL, NUQTREEX = NUQTREEX + QTD REG EXCLUIDOS NA TABELA ATUAL, NUDTAALT = data atual, NUHRAALT = hora atual, NMPRGALT = Nome do Job Talend, NMUSRALT = Nome do Usuário do Job Talend, NMESTALT = Nome da Máquina Onde o Job Talend Está Executando

Where

PKIDJOB = <Nome\_do\_Job\_Atual>

b) Escreve no arquivo de Log uma mensagem "SUCESSO" referente ao marco concluído. Exemplo: Sincronização da Tabela <XXXXX> Completado. <X de 5>.

. **Obs5**: Caso ocorram erros na execução do Job para algum dos marcos ou em qualquer outro processamento, atualizar a tabela de controle 'tbJOBETL', alterando os campos a seguir where o campo 'PKIDJOB' = <Nome\_do\_Job\_Atual>:

*update*

*tbjobetl*

*set*

*DSMSG = 'ERRO CARREGANDO TBL X de 5 - NOME DA TABELA', FGPROCES = 'N', NUDTAULT = Data atual, NUHRAFIM = hora atual, NUDTAALT = data atual, NUHRAALT = hora atual, NMPRGALT = Nome do Job Talend, NMUSRALT = Nome do Usuário do Job Talend, NMESTALT = Nome da Máquina Onde o Job Talend Está Executando*

*where*

*PKIDJOBL = <Nome\_do\_Job\_Atual>*

. Escreve no arquivo de Log uma menagem de "ERRO": 'ERRO CARREGANDO TBL X de 5 - NOME DA TABELA'. PROCESSAMENTO SERÁ ABORTADO. Descrição do Erro: XXX';

. Em seguida, se o campo Flag Aviso Email do Job está preenchido com 'S', então, consolida descrição do erro e, envia por Email para o endereço do grupo XXXXXX informando o erro ocorrido e informações sobre o Job.

. Fazer Rollback do que já foi atualizado apenas na tabela atual. As tabelas anteriores que foram atualizadas com sucesso não precisam de rollback;

. Finaliza processamento do Job para esperar a próxima execução conforme intervalo configurado no início do job;

. Aguarda tempo de intervalo configurado para o job;

. Reinicia o passo 6.1;

## Finalização + Atualização da tabela de controle:

. Após concluir o processamento das lógicas do programa que atenderão a regra de negócio da solução, executar os procedimentos a seguir.

. Executa um update na tabela de controle 'tbJOBETL' alterando os campos a seguir where o campo 'PKIDJOB' = <Nome\_do\_Job\_Atual>: flg\_processando de 'S' para 'N', NUHRAFIM = hora atual; NUPROGRS = 100, DSMSG = 'FINALIZADO COM SUCESO'; NUQTRELI =QTD REG LIDOS; NUQTREIN =QTD REG INCLUIDO; NUQTREAT =QTD REG ALTERADO; NUQTREEX=QTD REG EXCLUIDO; NUQTTENT =0; os demais de auditoria alteração;

. Escreve no arquivo de Log uma mensagem de "SUCESSO": Processamento do Job Finalizado;

. Aguarda tempo de intervalo configurado para o job;

. Reinicia o passo 6.1;

**Histórico de alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Item | Alteração efetuada | Autor |
|  |  |  |  |
|  |  |  |  |