

**Roteiro de Migração RDBMS**

**Projeto de Software**

**Alphalinc Upgrade**

**Maio/2014**



***TASC (the alpha supply chain)***

# Índice

[1 Índice 2](#_Toc390099243)

[2 Objetivo 3](#_Toc390099244)

[3 Histórico de Revisões 3](#_Toc390099245)

[4 Requisitos para montagem do ambiente 4](#_Toc390099246)

[4.1 Servidor Caché 4](#_Toc390099247)

[4.2 Servidor Oracle 4](#_Toc390099248)

[5 Instalação e configuração do ambiente 4](#_Toc390099249)

[5.1 Instalação do cliente Oracle 11g. 4](#_Toc390099250)

[5.2 Configuração da fonte de dados ODBC (DSN) 5](#_Toc390099251)

[5.3 Instalação do pacote de migração 7](#_Toc390099252)

[5.4 Demais premissas para migração dos dados 7](#_Toc390099253)

[6 Processo de Migração 8](#_Toc390099254)

[6.1 Importação das packages 8](#_Toc390099255)

[6.2 Inicialização das configurações 9](#_Toc390099256)

[6.3 Geração das definições das tabelas, colunas e triggers 10](#_Toc390099257)

[6.4 Exportação dos dados 10](#_Toc390099258)

[6.5 Carga dos dados 11](#_Toc390099259)

[6.6 Validação dos dados 11](#_Toc390099260)

[6.7 Geração das constraints e outras definições secundárias 11](#_Toc390099261)

[6.8 Sintaxe do comando de exportação 11](#_Toc390099262)

[7 Anexos 13](#_Toc390099263)

[7.1 Sintaxe do comando de exportação (Caché) 13](#_Toc390099264)

[7.2 Pacote de migração RDBMS 13](#_Toc390099265)

# Objetivo

Este documento tem como objetivo definir um roteiro inicial para migração das classes e dos dados do Alphalinc para um novo banco de dados relacional (Oracle), contemplando a instalação do cliente Oracle 11g, a configuração da ferramenta de migração e a execução das etapas necessárias para realização deste processo. O roteiro discrimina o ambiente utilizado para POC1 do projeto e todas as atividades necessárias para reprodução completa da migração.

# Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| ***Data*** | ***Versão*** | ***Autor*** | ***Descrição*** |
| 09/05/14 | 1.00 | Reinaldo Costa | Criação do documento |
| 12/05/14 | 1.01 | Alexandre van den Mosselaar | Revisão e complemento do documento |
| 28/05/14 | 1.02 | Reinaldo Costa | Alterado itens 4.1, 4.2, 6.2 e 6.7 |
| 02/06/14 | 1.03 | Alexandre van den Mosselaar | Demais premissas para migração (5.4) |
| 09/06/14 | 1.04 | Alexandre van den Mosselaar | Alterações nos itens 5.4, 6.6 e 6.8 |
| 13/06/14 | 1.05 | Alexandre van den Mosselaar | Alterações nos itens 5.4, 6.3 e 6.7 |
|  |  |  |  |

# Requisitos para montagem do ambiente

# Servidor Caché

* Servidor Caché (Origem dos dados)
  + Caché 2011.2 64bit
  + Client do Oracle 11g
  + Configuração do TNSNAMES.ora
  + Configuração da fonte de dados ODBC (DSN)
  + Scripts ref. á ferramenta de migração
  + Efetuar as instalações / configurações com o usuário administrador.

# Servidor Oracle

* Servidor Oracle (Destino dos dados)
  + Oracle 11g 64bit
  + Criar banco de dados com formato unicode – UTF8 (se necessário)
  + Configuração do Listener e TNSNAMES.ora
  + Testar comunicação do banco comando: tnsping <nome do banco>
  + Efetuar as instalações / configurações com o usuário administrador.

# Instalação e configuração do ambiente

# Instalação do cliente Oracle 11g.

* Efetuar a instalação do cliente do Oracle 11g no servidor Caché.

Na instalação do cliente, escolha a opção “**Administrator**” (Installation Type)

No path, retire o nome do usuário do path:

DE: C:\app\**administratos\**product\11.2.0\client\_1\network\admin

PARA: C:\app\product\11.2.0\client\_1\network\admin

No path C:\app\product\11.2.0\client\_1\network\admin

Crie o arquivo tnsnames.ora com o conteúdo abaixo:

**ORA11H02 =**

**(DESCRIPTION =**

**(ADDRESS\_LIST =**

**(ADDRESS = (PROTOCOL = TCP)(HOST = SVWDISC21)(PORT = 1521)))**

**(CONNECT\_DATA =**

**(SERVICE\_NAME = ORA11H02)))**

Alterar o SID, Host e Service name conforme a localização de seu banco de dados.

Para efetuar um teste de comunicação, no Prompt de comando execute:

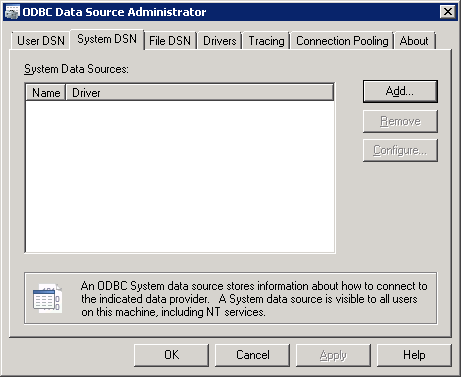
**tnsping ORA11H02**

# Configuração da fonte de dados ODBC (DSN)

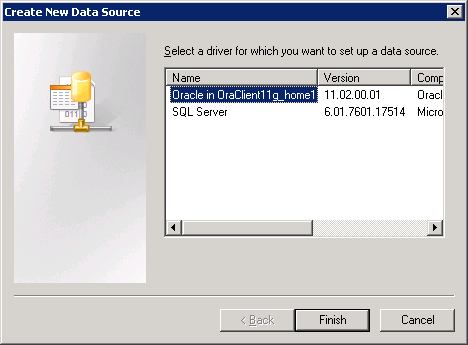
* Na Tela inicial do Windows clique em (no servidor Caché):

**START** -> **Administrative Tools** -> **Data Source (ODBC)**

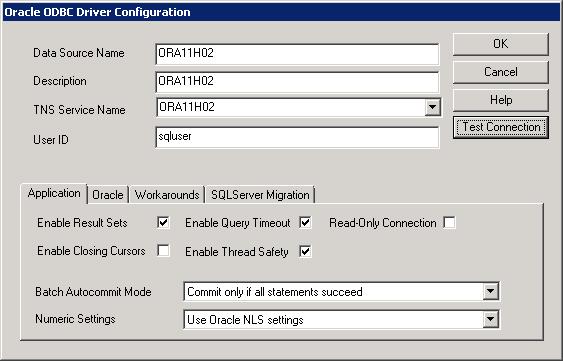
Clique na aba **System DNS**



Nesta tela Clique em **Add.**



Selecione a opção **Oracle in OraClient11g** e clique em **Finish.**



Informe o Data Source Name, Description e selecione o TNS Service Name.

Para o campo **User** **ID** coloque um usuário válido do banco de dados . Efetue teste clicando em **Test Connection** e informando a senha. Depois de concluído o teste clique em **OK.**

# Instalação do pacote de migração

No servidor Caché, no caminho: ...\Alphalinc\ deve-se descompactar o pacote ALUP\_MigracaoRDBMS.zip.

O mesmo criará as seguintes pastas:

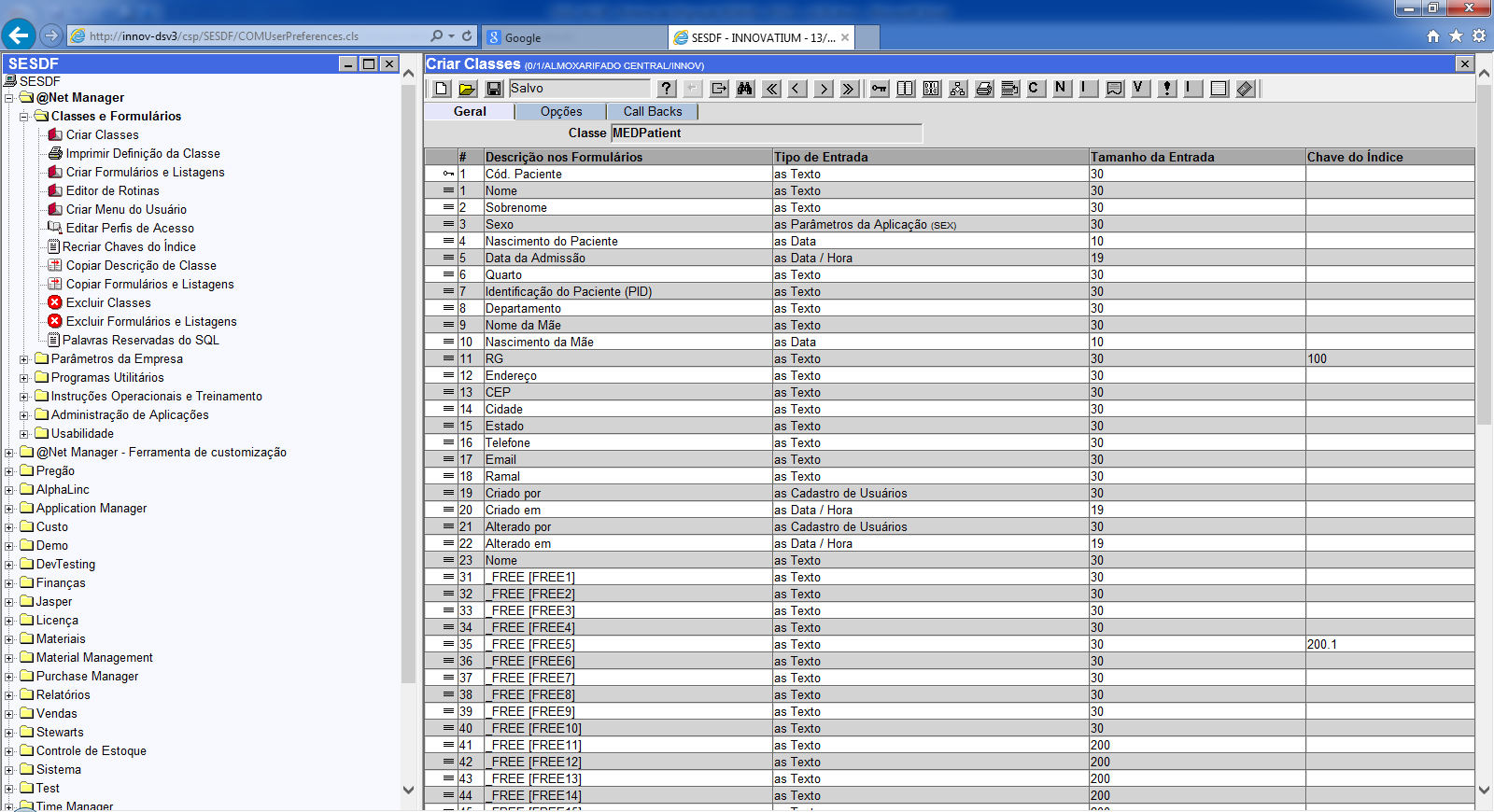
* ...\Alphalinc\Oracle\meta-scripts
* ...\Alphalinc\Oracle\packages
* ...\Alphalinc\Oracle\scripts

Após descompactar o pacote, instalar no Caché, junto ao namespace do Alphalinc o pacote ..\Alphalic\Oracle\glib\_rdbms.xml

# Demais premissas para migração dos dados

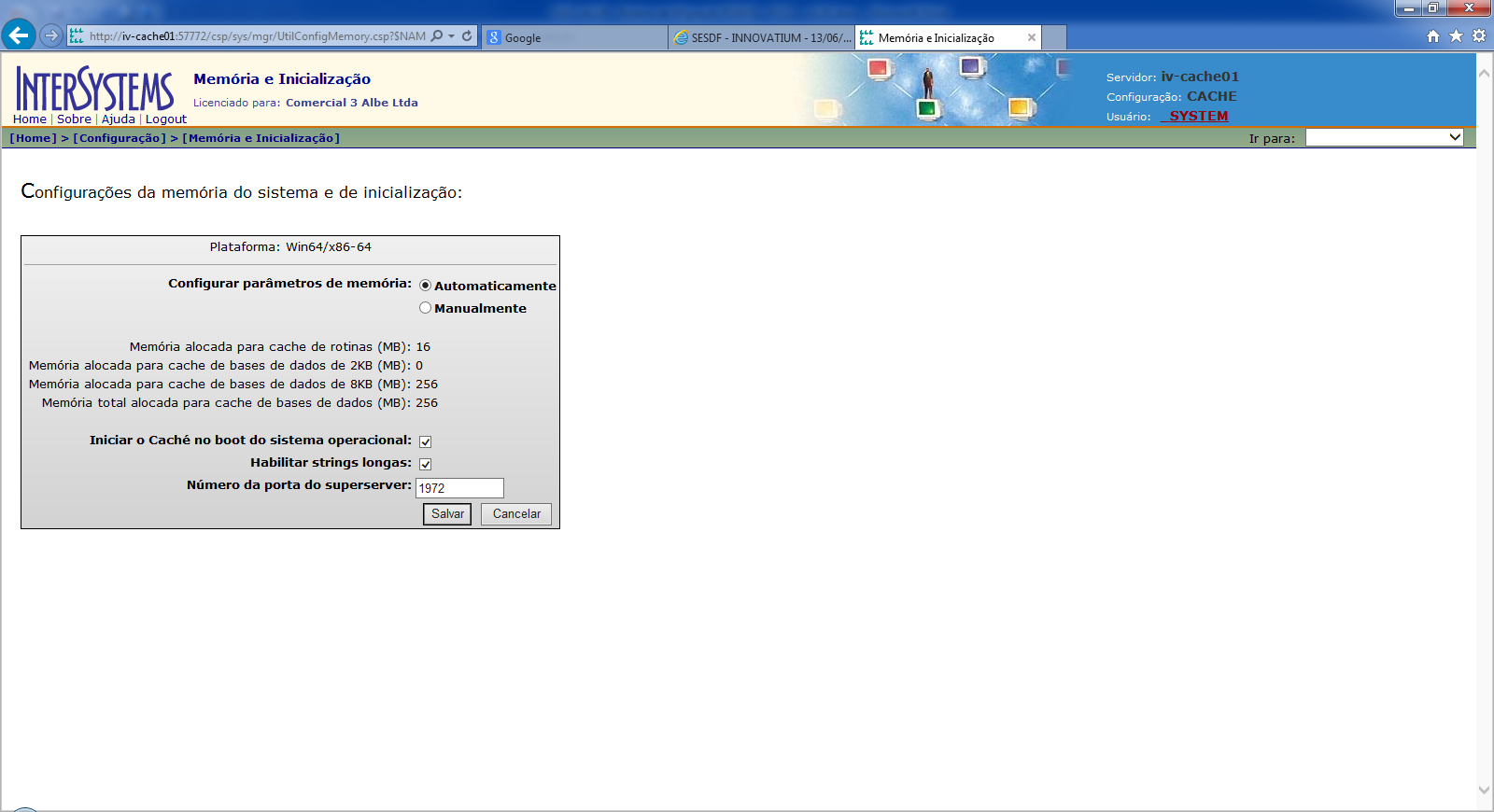
Além das definições e configurações descritas anteriormente, a correta migração dos dados exige o cumprimento das seguintes premissas:

* As classes do Net Manager devem estar compatíveis com as classes Caché (ou seja, não podem existir definições pendentes de compilação que englobam a criação de novas classes ou a criação de novos campos);



*Compilação de Classes dentro do NetManager*

* A base de dados Caché deve estar em formato Unicode (caso contrário, poderão ocorrer erros de wide char);
* A base de dados Caché deve estar configurada para suportar strings longas (vide opção de Configuração / Memória & Inicialização no Portal de Administração do Caché);



*Configuração de Memória e Inicialização*

* A base de dados Caché deve estar configurada com o tamanho máximo de processos (47630) e com um maior limite de locks (mínimo de 9437184) (vide opção de Configuração / Configurações Avançadas).

|  |  |  |
| --- | --- | --- |
| Memory | MaxMemPerProcess | 47630 |
| Memory | LockTableSize | 9437184 |

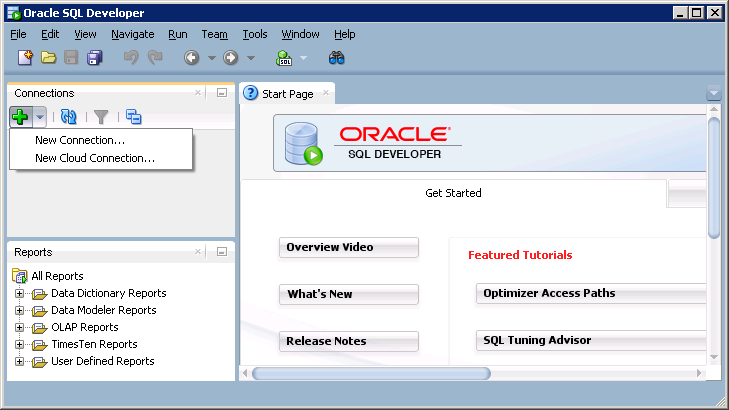
Além destas premissas, outros pontos são recomendados para melhoria do processo e garantia de integridade das informações:

* Reindexação de todos os dados, pois existem diversas com classes com falha de integridade nos índices;
* Atualização das classes que ainda utilizam o tipo de dado %Stream (atualmente o Net Manager trabalha diretamente com strings, ou seja, os dados armazenados não representam o conteúdo definido pelo tipo de dado nestes casos).

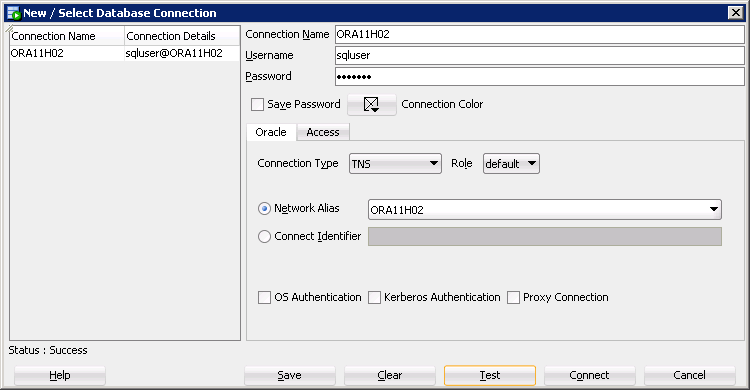
# Processo de Migração

# Importação das packages

* + - Abrir o sqldeveloper clicando em:
* **INICIAR** -> **All Programs** -> **Oracle – OraClient11g** -> **Application** **Development** -> **sqldeveloper**



Na aba connection clicar em **New Connection**

****

Alterar o **Connection type** para **TNS**, selecionar o **Network Alias** e informar **Username** e **Password** depois em **Test** e **Save.**

Executar as packages:

**D:\Alphalinc\Oracle\packages\COMSYSFC\_PCK.sql e**

**D:\Alphalinc\Oracle\packages\MFUNCTION\_PCK.sql**

# Inicialização das configurações

* No servidor Caché:
  + Confirmar no portal de gerenciamento do sistema do Caché, localizado em: **Home / configuração / memória inicialização** a permissão para strings longas.(Enable long string).
  + Abrir uma sessão **terminal** no Caché (informar usuário e senha)
  + No terminal executar o comando:

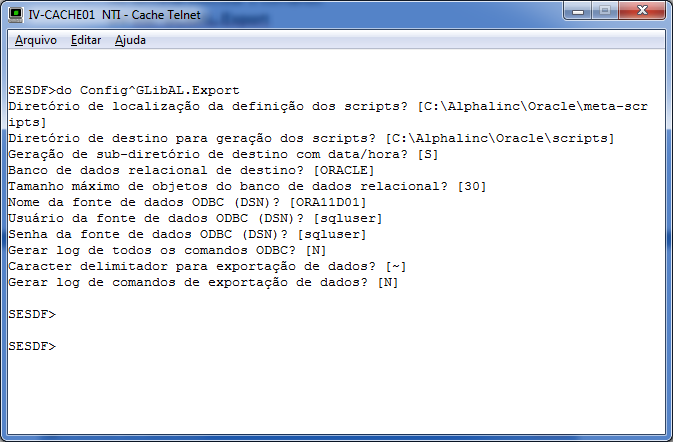
**zn “<namespace>”**

* + No terminal executar o comando:

**Do Init^GLibAL.Export**

* + **Informar os parâmetros de configuração conforme ambiente.**

**Exemplo:**

****

# Geração das definições das tabelas, colunas e triggers

* Executar o script de criação das tabelas e colunas (via ODBC)

**Do ^GLibAL.Export("definitions","\*","ODBC")**

\*\* Incluindo primary keys, sequences e triggers

# Exportação dos dados

* Executar o script de exportação de dados.

**Do ^GLibAL.Export("dataexport","\*")**

# Carga dos dados

* No servidor Oracle copiar os dados gerados pelo servidor Caché.

Copiar para <destino> a pasta \\<servidor>\...\Alphalinc\Oracle\scripts\

* Pelo prompt de comando acessar a pasta copiada no destino e executar o LOAD.bat

# Validação dos dados

* Coletar os dados através do comando:

**Do ^GLibAL.Export("datacount","\*",”ODBC”)**

* Importar o resultado gerado no log "datacount.log" no Excel para analisar as diferenças

# Geração das constraints e outras definições secundárias

* Executar o script de criação das constraints (via ODBC)

**Do ^GLibAL.Export("constraints","\*","ODBC")**

* Ajuste do tamanho das colunas
* Definição das colunas obrigatórias
* Criação dos índices
* Criação das foreign keys
* Criação das colunas virtuais

# Sintaxe do comando de exportação

**Do ^GLibAL.Export("<metascript>","<filter>","<outputclass>")**

<metascript>: nome do arquivo de meta-script (utilizado para geração dos scripts)

<filter>: opções de filtro das classes:

\*: todas as classes

<like expression>: expressão like

<filter1>,<filter2>,...: múltiplos filtros separados por vírgula

<.array>: passagem por referência de uma variável contendo um array (array("name1")="", array("name2")="", ...)

<outputclass>: classe de tratamento do script de saída (no momento é utilizada somente as opções: <em branco> para geração de scripts nos arquivos de log ou <ODBC> para submeter os scripts no banco relacional de destino)

* Após a execução do comando é gerado um arquivo de log conforme o filtro selecionado (<filtro>**\_log**.txt)
* Em caso de erros durante a execução é gerado um outro arquivo contendo os erros encontrados (<filtro>**\_err**.txt)

# Anexos

# Sintaxe do comando de exportação (Caché)

Sintaxe:

Do ^GLibAL.Export("<**meta-scrip**t>","<**filter**>","<**output**>")

<**meta-script**>: definição do meta-script para geração dos scripts

<**filte**r>: filtro das tabelas a serem geradas (\* para todas ou expressão like)

<**output**>: ODBC para conexão direta ao banco ou vazio (geração de arquivos de script).

# Pacote de migração RDBMS

