

Montando a gestão

Os recursos do sistema ERP e CRM ADempiere já foram mostrados na última edição. Veja agora como obter e instalar o código-fonte desse sistema de Código Aberto.

por Eduardo Montenegro



Maxime Perron Galsky - www.sxc.hu

Na última edição da **Linux Magazine** publicamos[1] uma visão geral sobre os principais recursos e características técnicas do sistema de ERP & CRM ADempiere[2]. A partir deste mês, iremos publicar uma série de quatro tutoriais, com o intuito de guiar os primeiros passos na utilização desse poderoso e flexível sistema.

Após abordarmos a instalação do sistema neste primeiro tutorial, explicando os passos necessários para a instalação do ADempiere em uma

estação Linux com o banco de dados PostgreSQL, o próximo artigo tratará dos principais recursos de desenvolvimento no sistema utilizando seu dicionário de dados. Utilizaremos o dicionário de dados para criar novas tabelas no banco de dados, além de novas janelas e regras de validação de dados utilizadas para a entrada de dados pelo usuário.

No terceiro artigo, abordaremos o código-fonte do ADempiere e o desenvolvimento em *Java* com o IDE *Eclipse*. Vamos explorar a estrutura

do código-fonte do sistema e os principais recursos de desenvolvimento para ele utilizando *Java*.

Por último, o quarto artigo discorrerá sobre o projeto ADempiere LBR, um complemento para o sistema ADempiere que adiciona os recursos necessários à utilização do sistema por empresas brasileiras. Nesse último tutorial, veremos como adicionar o projeto LBR ao código-fonte padrão do ADempiere e quais as principais características e recursos já disponíveis no sistema para as empresas nacionais.

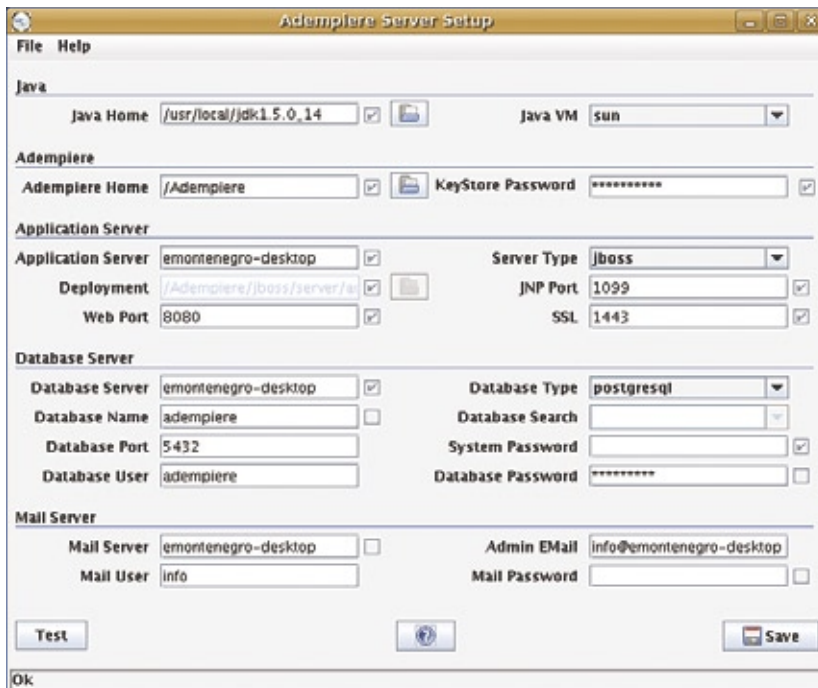


Figura 1 Confira se a tela de configuração exibe os parâmetros corretos para a instalação do ADempiere.

Instalação

Vamos agora apresentar algumas características principais do ADempiere, sem esgotarmos o assunto ou nos aprofundarmos muito em determinados tópicos.

Existem diversas fontes de informação disponíveis na internet que podem ajudar a ampliar os conhecimentos sobre o sistema, como o site do projeto[2] com seu wiki[3], sua página no *Sourceforge*[4], a página do Projeto *ADempiereLBR*[5] e o fórum do sistema em língua portuguesa[6].

Este tutorial usará como distribuição o *Ubuntu 7.10*[7] em sua versão *i386*, empregando o *PostgreSQL 8.2.6*[8] como banco de dados, o *Java Development Kit 5.0 Update 14*[9] e o conector *PL/Java 1.4.0*[10].

Exemplo 1: Compilação

```
01 ~/src/olamundo $ make
02 Scanning dependencies of target olamundo
03 [100%] Building C object CMakeFiles/ola.dir/main.o
04 Linking C executable olamundo
05 [100%] Built target olamundo
06 ~/src/olamundo $ ./olamundo
07 Ola mundo!
```

O sistema de testes foi instalado numa máquina virtual VMware com 12 GB de disco e 1 GB de memória, com as opções padrão do instalador do Ubuntu. Com o intuito de manter a padronização, criaremos um usuário para executar o ADempiere, com o surpreendentemente nome de *adempiere*. Isso pode ser efetuado pelo próprio menu do *Gnome*.

Pacote problemático

Segundo relatos nos fóruns do ADempiere, o pacote do PostgreSQL nas distribuições derivadas do *Debian* (incluindo o próprio *Debian*) vem com certas opções desativadas, as quais são necessárias para o funcionamento correto do conector do banco de dados com o Java, *PL/Java*. Portanto, é extremamente recomendável compilar o PostgreSQL a partir do código-fonte conforme liberado por seus desenvolvedores centrais.

Para isso, primeiro é necessário instalar os pacotes do compilador C++ do GCC e as bibliotecas *Libreadline* e *Zlib1g* através do comando:

```
$ sudo aptitude install g++
↳libreadline-dev zlib1g-dev
```

Em seguida, deve-se baixar o código-fonte do PostgreSQL em [11] e salvá-lo em `/usr/local/src/postgresql/`. Note que esse procedimento deve ser feito sob o usuário *adempiere*, mas o diretório de destino não pode ser criado por ele. Então, são necessários os seguintes comandos no Ubuntu:

```
$ sudo mkdir -p /usr/local/src/
↳postgresql
$ sudo chown adempiere /usr/local/
↳src/postgresql
```

Após descompactar o código-fonte dentro desse diretório, é hora de compilar e instalar o banco de dados:

```
$ ./configure
$ make
$ sudo make install
```

Em seguida, precisamos preparar o PostgreSQL para execução. Primeiro, é necessário criar o usuário que irá executar o processo do banco de dados:

```
$ sudo useradd postgres
$ sudo mkdir /usr/local/pgsql/data
$ sudo chown postgres /usr/local/
↳pgsql/data
```

Depois, inicializamos o banco de dados:

```
$ su -l postgres
$ /usr/local/pgsql/bin/initdb -D /
↳usr/local/pgsql/data
```

Como se trata de um ambiente de testes e desenvolvimento, vamos configurar o PostgreSQL para permitir a conexão de qualquer usuário em qualquer interface de rede a qualquer banco de dados que esteja configurado. Ainda como usuário *postgres*, substitua no arquivo `/usr/local/pgsql/data/pg_hba.conf` a linha:

```
host all all 127.0.0.1/32 trust
```

pela linha:

```
host all all 0.0.0.0/0 trust
```

Agora podemos iniciar e parar o banco de dados, respectivamente, como usuário *postgres*, através dos comandos:

```
$ /usr/local/pgsql/bin/pg_ctl -D /
↳usr/local/pgsql/data -o "-i" -l
↳logfile start
$ /usr/local/pgsql/bin/pg_ctl -D /
↳usr/local/pgsql/data -o "-i" -m
↳fast -l logfile stop
```

Java

Com o banco corretamente instalado, configurado e iniciado, vamos agora instalar o ambiente Java com os comandos abaixo:

```
$ sudo sh ./jdk-1_5_0_14-linux-
↳i586.bin
$ sudo mv ./jdk1.5.0_14 /usr/local
```

Em seguida, para instalar o PL/Java no PostgreSQL, é preciso definir no arquivo `/etc/profile` as variáveis de ambiente *JAVA_HOME*, *PGDATA* e *PATH*, acrescentando essas linhas ao seu final:

```
export JAVA_HOME=/usr/local/
↳jdk1.5.0_14
export PGDATA=/usr/local/pgsql/
↳data
export PATH=$PATH:/usr/local/
↳pgsql/bin:/usr/local/jdk1.5.0_
↳14/bin
```

As alterações podem ser efetivadas com `source /etc/profile`.

O PL/Java também deve ser instalado a partir de seu código-fonte, que se encontra em [12].

Da mesma forma que fizemos com o PostgreSQL, vamos descompactar o código-fonte do PL/Java num diretório especificamente criado para esse fim, e

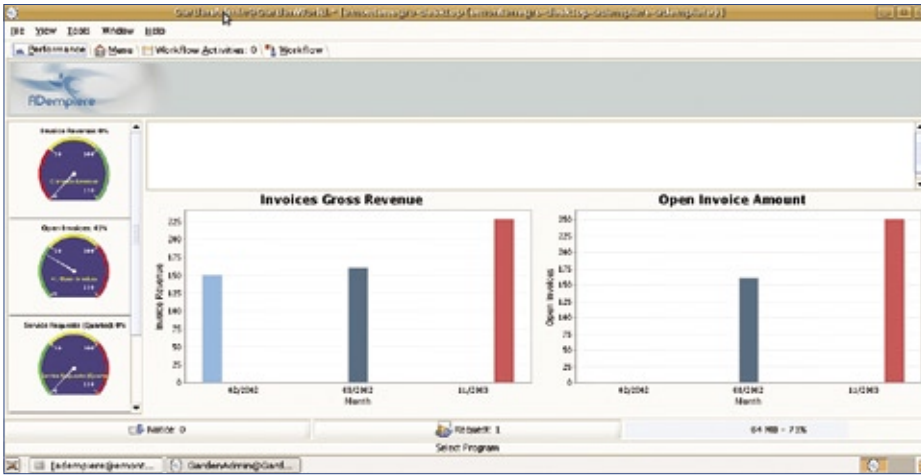


Figura 2 O aplicativo cliente do ADempiere oferece uma interface com boa usabilidade.

compilá-lo como mostrado no exemplo 1.

Agora precisamos criar um arquivo chamado `postgresql.conf` no diretório `/etc/ld.so.conf.d` com as seguintes linhas:

```
/usr/local/jdk1.5.0_14/jre/lib/
↳ i386
/usr/local/jdk1.5.0_14/jre/lib/
↳ i386/client
```

Finalmente, como usuário `postgres`, é preciso editar o arquivo `/usr/local/pgsql/data/postgresql.conf` para configurar o PL/Java. No final do arquivo, inclua as duas linhas:

```
custom_variable_classes = 'pljava'
pljava.classpath = '/usr/local/
↳ postgresql/lib/pljava.jar'
```

Finalmente, podemos prosseguir com a instalação do ADempiere. Vamos instalá-lo no diretório raiz (`/`), e para isso devemos primeiro criar um diretório para o ADempiere e atribuir-lhe o usuário `adempiere`:

```
$ sudo mkdir /ADempiere
$ sudo chown adempiere /ADempiere
```

No site do projeto do ADempiere no Sourceforge[4], devemos baixar o arquivo `ADempiere_331b.tar.gz`.

Em seguida, vamos descompactar o conteúdo do arquivo no diretório `ADempiere/`.

```
$ tar xzvf ./ADempiere_331b.tar.gz
$ mv ./ADempiere/* /ADempiere
```

Agora, dentro do diretório `ADempiere/` executaremos seu script de instalação:

```
$ sh ./RUN_setup.sh
```

Na janela de configurações que será exibida, é fundamental confirmar se os parâmetros estão definidos conforme a figura 1.

Feito isso, basta selecionar os botões `Test` e em seguida `Save`.

No arquivo `/etc/profile` é necessário incluir a linha:

```
export ADEMPIERE_HOME=/ADempiere
```

Agora vamos importar o banco de dados do ADempiere para o PostgreSQL com o comando:

```
$ /ADempiere/Utils/RUN_
↳ ImportADempiere.sh
```

Após a importação do banco de dados, o servidor de aplicação é iniciado, e em seguida podemos iniciar também o aplicativo cliente (figura 2):

```
$ /ADempiere/Utils/RUN_Server2.sh
$ /ADempiere/RUN_ADempiere.sh
```

Conclusão

Seguindo este tutorial, o ADempiere estará instalado e configurado. No próximo artigo, vamos explorar algumas características do desenvolvimento do sistema utilizando o “dicionário de dados”. ■

Mais informações

- [1] “Bom modelo”, Linux Magazine #39: http://www.linuxmagazine.com.br/article/bom_modelo
- [2] ADempiere: <http://www.adempiere.com/>
- [3] Wiki do ADempiere: <http://www.adempiere.com/wiki>
- [4] ADempiere no Sourceforge: <http://sf.net/projects/adempiere>
- [5] Projeto ADempiereLBR: <http://sf.net/projects/adempiereLBR>
- [6] Fórum do ADempiere em língua portuguesa: <http://forum.kenos.com.br>
- [7] Ubuntu: <http://www.ubuntu.com>
- [8] PostgreSQL 8.2.6 (<http://www.postgresql.org>)
- [9] Java Development Kit 5.0 Update 14 (<http://www.sun.com>)
- [10] PL/Java 1.4.0 (<http://pgfoundry.org/projects/pljava>)
- [11] Código-fonte do PostgreSQL 8.2.6: <http://www.postgresql.org/ftp/source/v8.2.6/>
- [12] Código-fonte do PL/Java: <http://pgfoundry.org/projects/pljava>