

# Programación distribuida con ADA95 bajo GNU/Linux (I)

Alvaro López Ortega

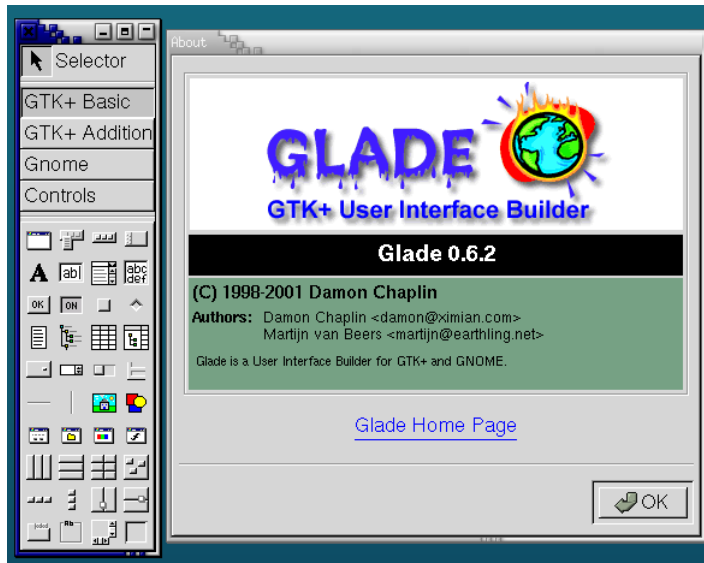
alvaro@godsmaze.org

## 1. Introducción a GNAT-Glade



En primer lugar hay que aclarar que el nombre de esta librería puede confundir a los usuarios y programadores de GTK+ y GNOME. Existe una aplicación muy extendida para el diseño de interfaces gráficas que se llama 'Glade'. Un grandísimo número de lenguajes de programación disponen de librerías para poder leer los ficheros de interfaces que genera Glade (Python, C++, Scheme, Ruby, Eiffel, Haskell, etc). Pues bien, GNAT-Glade no tiene nada que ver con esta (magnífica ;-) herramienta.

Figura 1. Este Glade NO es.



GNAT-Glade es una extensión para GNAT95, el compilador libre (licenciado bajo GPL) de Ada95, que permite desarrollar aplicaciones distribuidas con este lenguaje.

La base de las aplicaciones distribuidas de Ada95 son las 'particiones'. Básicamente una aplicación distribuida se compone de al menos un par de particiones.

Es posible utilizar GNAT-Glade de dos formas diferentes:

- Con varias particiones sobre la misma máquina.
- Con varias particiones sobre diferentes máquinas que formen parte de una red de computadoras.

Desde luego resulta mucho más interesante la segunda de las opciones. Es más, para desarrollar aplicaciones con varias particiones sobre una misma máquina hay muchos casos en que sería más conveniente no utilizar Glade y basarse únicamente en los mecanismos de concurrencia de GNAT (Ada95): la aplicación será más eficiente.

## 2. ¿Cómo funciona GNAT-Glade?

Cada una de las particiones de una aplicación basada en GNAT-Glade, a la hora de la compilación se va a convertir en un ejecutable independiente. Cada uno de estos ejecutables serán los que se ejecuten por

separado y se comuniquen entre ellos.

Existe una herramienta que facilita todo este proceso: gnatdist.

Gnatdist lee un fichero de configuración en el que se especifica como queremos distribuir la aplicación y genera todos los ejecutables necesarios. De esta forma, es posible probar diferentes formas de distribuir una misma aplicación simplemente con lanzar gnatdist con un fichero de configuración distinto, sin necesidad de modificar el código de la aplicación.

## 3. Lenguaje de configuración de GnatDist

Las configuraciones de GnatDist se escriben en un lenguaje muy parecido a ADA'95. Es importante que todas las configuraciones se guarden en ficheros con extensión ".cfg".

Para lanzar la compilación de una aplicación distribuida con GnatDist únicamente es necesario ejecutar gnatdist dándole como parámetro el nombre del fichero de configuración. Por ejemplo:

```
alo@servidor:~$ gnatdist Ejemplo_Configuracion1.cfg
```

### 3.1. ¿Cómo se escriben las configuraciones?

En cualquier punto de la configuración es posible usar comentarios, que al igual que en Ada se comienzan con los caracteres '-- '.

Los ficheros de configuración han de contener un bloque "configuration", cuyo nombre ha de coincidir, al igual que en el caso de los paquetes, con el nombre del fichero en el que se encuentra. Es decir:

```
configuration Ejemplo_Configuracion1 is
  --
  -- Código de la configuración
  --
end Ejemplo_Configuracion1;
```

CONTINUAR !!! .. no esta terminado

## 4. Primer ejemplo

El movimiento se aprender andando.. así que, vamos a por el primer ejemplo.

En este ejemplo vamos a crear una pequeña aplicación compuesta únicamente de dos particiones:

- La primera de ellas es un servidor de sumas y restas (al más puro estilo RPC). En el se van a definir dos operaciones: suma y resta, que dados dos números enteros, van a devolver el resultado de aplicar la operación elegida sobre ambos.
- La segunda de ellas es un pequeño cliente que efectuará un par de operaciones para comprobar que efectivamente el servidor responde.

## 4.1. calculadora.ads

```
package Calculadora is

pragma Remote_Call_Interface;

function Sumar (Operador1 : in Integer;
               Operador2 : in Integer)
  return Integer;

function Restar (Operador1 : in Integer;
                Operador2 : in Integer)
  return Integer;

end Calculadora;
```

En esta definición del paquete llama la atención la instrucción 'pragma'. Un pragma es simplemente una directiva para el compilador. En concreto, en este ejemplo, el pragma `Remote_Call_Interface` hace que se exporte la interfaz del paquete para que otras particiones puedan realizar llamadas a sus funciones, es decir, básicamente una llamada RPC.

## 4.2. calculadora.adb

```
package body Calculadora is

function Sumar (Operador1 : in Integer;
               Operador2 : in Integer)
  return Integer is
begin
  return Operador1 + Operador2;
end Sumar;

function Restar (Operador1 : in Integer;
                Operador2 : in Integer)
  return Integer is
```

```
begin
    return Operador1 - Operador2;
end Restar;

end Calculadora;
```

Este fichero es únicamente la implementación de las funciones del paquete calculadora.

### 4.3. cliente.adb

```
with Text_IO; use Text_IO;
with Calculadora;

procedure Cliente is
begin

    Put_Line ("- Calculadora, ¿cuanto es 321+123? = " &
              Integer'Image (Calculadora.Sumar (321,123)));

    Put_Line ("- Calculadora, ¿cuanto es 321-123? = " &
              Integer'Image (Calculadora.Restar (321,123)));

end Cliente;
```

Por último, el cliente. Este programa hace un par de llamadas a las funciones exportadas por el proceso de calculadora. Como se puede ver el código no tiene en cuenta si el proceso calculadora se encuentra corriendo en la misma máquina o en otra. Simplemente realiza llamadas a las funciones de la calculadora. De todo lo demás, que es mucho, ya se ha encargado gnatdist y se encarga ADA.

### 4.4. ejemplo.cfg

```
configuration ejemplo is
pragma Starter (Ada);

Particion1: Partition := (Calculadora);
Particion2: Partition := (Cliente);

procedure Cliente is in Particion2;
end ejemplo;
```

Este es el fichero de configuración/compilación de gnatdist.

El pragma Starter describe como queremos que gnatdist compile el proyecto. Existen tres posibilidades: Ada, Shell y None. En este caso es uno de los ejecutables el que lance todos los demás.

A continuación se definen las dos particiones que se han utilizado en este ejemplo: una para la calculadora y la segunda para el cliente que le realiza peticiones.

Por último se especifica cual es la parte principal (el main). Esta partición, lógicamente, ha de tener un body.

**Importante:** Cuidado con los nombres de los ficheros: han de coincidir con el nombre del paquete y además, han de estar en minúsculas. De no ser así gnatdist producirá un error.

## 4.5. Compilación y ejecución del programa

Para compilar la aplicación, como ya hemos visto, simplemente hay que ejecutar gnatdist:

```
$ gnatdist ejemplo.cfg
```

Si no ha habido ningún problema, se habrá producido una salida como esta:

```
gnatdist: checking configuration consistency
-----
---- Configuration report ----
-----
Configuration :
  Name      : ejemplo
  Main      : cliente
  Starter   : Ada code

Partition particion1
  Units     :
            - calculadora (rci)

Partition particion2
  Main      : cliente
  Units     :
  Name      : ejemplo
  Main      : cliente
  Starter   : Ada code

Partition particion1
  Units     :
            - calculadora (rci)

Partition particion2
  Main      : cliente
  Units     :
            - cliente (normal)
```

```
-----  
gnatdist: building calculadora caller stubs from calculadora.ads  
gnatdist: building calculadora receiver stubs from calculadora.adb  
gnatdist: building partition particion1  
gnatdist: building partition particion2  
gnatdist: generating starter cliente
```

En este momento ya tenemos construidos todos los ejecutables:

```
lrwxrwxrwx    1 alo      alo           10 Oct  9 22:33 cliente -> particion2  
-rwxr-xr-x    1 alo      alo          3663802 Oct  9 22:33 particion1  
-rwxr-xr-x    1 alo      alo          3723193 Oct  9 22:33 particion2
```

Como podemos ver, al especificar el Pragma Starter (Ada) en el fichero de configuración, gnatdist ha generado un link simbólico al ejecutable que va a lanzar los demás.

En esta prueba vamos a ejecutar los dos programas en la misma máquina, más adelante se explicará como hacerlo en varias.

```
$ ./cliente  
Host for "particion1": localhost  
- Calculadora, ¿cuanto es 321+123? = 444  
- Calculadora, ¿cuanto es 321-123? = 198
```

Ahora bien.. ¿estamos seguros de que se ha ejecutado un programa paralelo? ¿estamos seguros de que en realizar no se la linkado el paquete Calculadora en los dos ejecutables?

Como podemos ver en los procesos, realmente se han ejecutado los dos programas:

```
[..]  
7701 pts/10    S        0:00  \_  bash  
8753 pts/10    S        0:00  |   \_  ./cliente  
8754 pts/10    S        0:00  |       \_  ./cliente  
8755 pts/10    S        0:00  |           \_  ./cliente  
8793 pts/10    S        0:00  |               \_  ./cliente  
  
[..]  
8788 ?        S        0:00  /home/alo/prog/glade/1/particion1 --detach --boot_location tcp://  
8790 ?        R        0:00  \_ /home/alo/prog/glade/1/particion1 --detach --boot_location tc  
8791 ?        S        0:00  \_ /home/alo/prog/glade/1/particion1 --detach -  
-boot_location tcp://localhost:35802  
8792 ?        S        0:00  \_ /home/alo/prog/glade/1/particion1 --detach -  
-boot_location tcp://localhost:35802
```

```
8794 ?      R      0:00      \_ /home/alo/prog/glade/1/particion1 --detach -  
-boot_location tcp://localhost:35802
```

Es más, si examinamos los símbolos de los dos ejecutables podemos ver como la calculadora tiene linkadas las funciones de suma y resta:

```
08050500 g      F .text 00000011      calculadora__restar  
080504f0 g      F .text 0000000d      calculadora__sumar
```

y el cliente, además, tiene las funciones que usa gnatdist para el acceso:

```
080502c0 l      F .text 00000018      calculadora__sumar__clean.0  
080502e0 l      F .text 0000005a      calculadora__sumar__input27__read30.2  
08050340 l      F .text 0000018a      calculadora__sumar__input27.1  
080504d0 g      F .text 000002bf      calculadora__sumar  
08050790 l      F .text 00000018      calculadora__restar__clean.3  
080507b0 l      F .text 0000005a      calculadora__restar__input67__read70.5  
08050810 l      F .text 0000018a      calculadora__restar__input67.4  
080509a0 g      F .text 000002bf      calculadora__restar
```

## 5. Instalación bajo Debian GNU/Linux

Instalación de GNAT (compilador de Ada95) y Glade para GNAT (extensión para soporte de programación distribuida).

Para realizar la instalación en el sistema es imprescindible estar conectado como usuario 'root':

```
# apt-get install gnat gnat-glade  
Reading Package Lists... Done  
Building Dependency Tree... Done  
The following NEW packages will be installed:  
  gnat gnat-glade  
0 packages upgraded, 2 newly installed, 0 to remove and 2 not upgraded.  
Need to get 10.1MB of archives. After unpacking 40.2MB will be used.  
Get:1 http://http.us.debian.org unstable/main gnat 3.13p-8 [6789kB]  
Get:2 http://http.us.debian.org unstable/main gnat-glade 3.13p-6 [3339kB]  
Fetched 10.1MB in 6m30s (25.9kB/s)  
Selecting previously deselected package gnat.  
(Reading database ... 93759 files and directories currently installed.)  
Unpacking gnat (from ../archives/gnat_3.13p-8_i386.deb) ...  
Selecting previously deselected package gnat-glade.  
Unpacking gnat-glade (from ../gnat-glade_3.13p-6_i386.deb) ...  
Setting up gnat (3.13p-8) ...
```



```
Setting up gnat-glade (3.13p-6) ...
```

Instalación de la documentación de ambos paquetes. Estos paquetes son opcionales aunque muy recomendables.

```
# apt-get install gnat-glade-doc gnat-doc
Reading Package Lists... Done
Building Dependency Tree... Done
The following NEW packages will be installed:
  gnat-doc gnat-glade-doc
0 packages upgraded, 2 newly installed, 0 to remove and 2 not upgraded.
Need to get 3708kB of archives. After unpacking 6567kB will be used.
Get:1 http://http.us.debian.org unstable/main gnat-doc 3.13p-2 [3399kB]
Get:2 http://http.us.debian.org unstable/main gnat-glade-doc 3.13p-2 [309kB]
Fetched 3708kB in 2m19s (26.5kB/s)
Selecting previously deselected package gnat-doc.
(Reading database ... 95179 files and directories currently installed.)
Unpacking gnat-doc (from .../gnat-doc_3.13p-2_all.deb) ...
Selecting previously deselected package gnat-glade-doc.
Unpacking gnat-glade-doc (from .../gnat-glade-doc_3.13p-2_all.deb) ...
Setting up gnat-doc (3.13p-2) ...

Setting up gnat-glade-doc (3.13p-2) ...
```