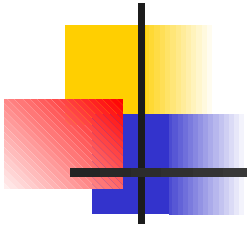


PAN: *Lenguaje de configuración de alto nivel*



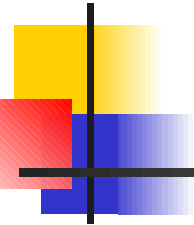
Juan José Pardo Navarro

juanjo.pardo@uam.es

Departamento de Física Teórica

Universidad Autónoma de Madrid

PAN: Lenguaje de configuración de alto nivel



1. Introducción a PAN

1.1 CDB y Pan

1.2 Funcionamiento del compilador de Pan (panc)

1.3 Características principales de PAN

1.4 Forma de representar la información

2. Sintaxis de PAN

3. Comentar un ejemplo de código PAN

4. Ejercicios

PAN: *Lenguaje de configuración de alto nivel*

1.1 CDB PAN

♦ Perfiles:

- ♦ Elementos de configuración de los clientes
- ♦ Existe en el servidor CDB un perfil por cada equipo cliente
- ♦ Son escritos utilizando el lenguaje PAN



PAN: *Lenguaje de configuración de alto nivel*

1.2 Funcionamiento del compilador (panc)

panc -x compact profile.tpl



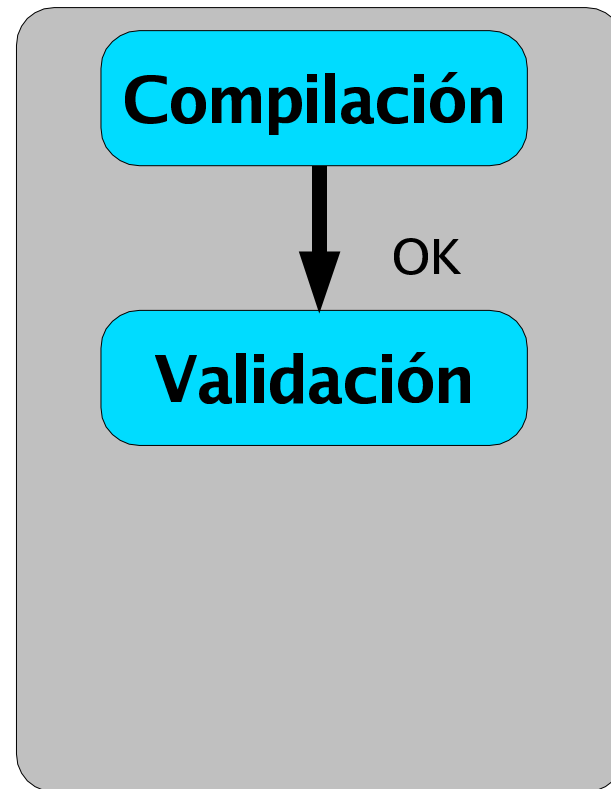
Compilación

A large, light gray rounded rectangle with a thin black border. At the top center, there is a smaller, bright blue rounded rectangle with a thin black border. Inside the blue rectangle, the word "Compilación" is written in bold black text.

PAN: *Lenguaje de configuración de alto nivel*

1.2 Funcionamiento del compilador (panc)

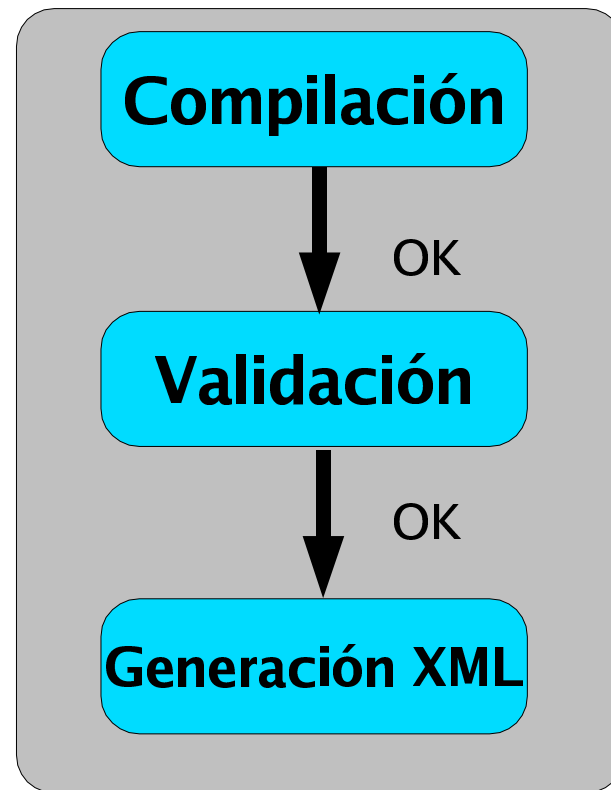
panc -x compact profile.tpl



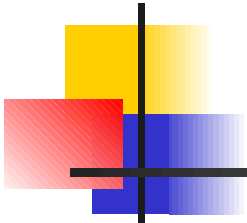
PAN: *Lenguaje de configuración de alto nivel*

1.2 Funcionamiento del compilador (panc)

panc -x compact profile.tpl



PAN: *Lenguaje de configuración de alto nivel*



1.3 Características del lenguaje PAN

- Lenguaje de programación declarativa
- Ofrece nivel de abstracción
- Permite jerarquizar la información
- Comprobación de tipos

PAN: *Lenguaje de configuración de alto nivel*

1.4 Representación de la información

- Elemento de configuración: estructura de árbol
 - Propiedad
 - Recurso (list,nlist)
- Path: Identificador de un elemento de configuración

Propiedad

/profile/network/interfaces/eth0/ip

Recurso

/profile/network/interfaces/eth0

/profile/network/interfaces

/profile/network

/profile

Path relativo

interfaces/eth0/ip

PAN: *Lenguaje de configuración de alto nivel*

1.4 Representación de la información

- Elemento de configuración: estructura de árbol
 - Propiedad
 - Recurso (list,nlist)
- Path: Identificador de un elemento de configuración

Ejemplo nlist (interfaces)

“/profile/network/interfaces/eth0/ip” = “150.244.10.200”

“/profile/network/interfaces/eth0/gateway” = “150.244.10.1”

“/profile/network/interfaces/eth0/netmask” = “255.255.255.0”

“/profile/network/interfaces/eth1/ip” = “192.168.0.100”

“/profile/network/interfaces/eth1/gateway” = “192.168.0.1”

“/profile/network/interfaces/eth1/netmask” = “255.255.255.0”

PAN: *Lenguaje de configuración de alto nivel*

1.4 Representación de la información

- Elemento de configuración: estructura de árbol
 - Propiedad
 - Recurso (list,nlist)
- Path: Identificador de un elemento de configuración

Ejemplo de list (ram)

“/profile/hardware/ram/0/type” = “DDR”

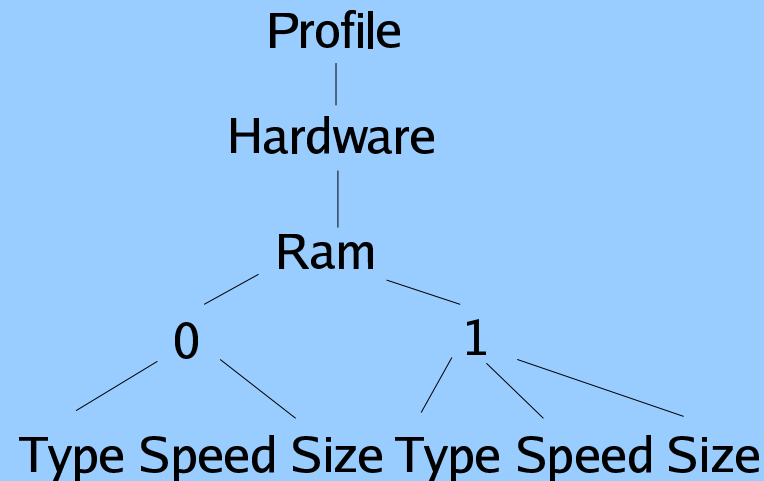
“/profile/hardware/ram/0/speed” = 133

“/profile/hardware/ram/0/size” = 256

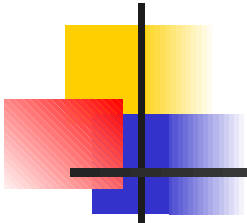
“/profile/hardware/ram/1/type” = “DDR”

“/profile/hardware/ram/1/speed” = 133

“/profile/hardware/ram/1/size” = 256



PAN: Lenguaje de configuración de alto nivel

- 
1. Introducción a PAN
 2. **Sintaxis de PAN**
 - 2.1 Sentencias y su organización
 - 2.2 Comentarios en PAN
 - 2.3 Comprobación de tipos
 - 2.4 Tipos básicos y variables
 - 2.5 Operadores
 - 2.6 Tipos complejos
 - 2.7 Control de flujos
 - 2.8 Funciones
 3. Comentar un ejemplo de código PAN
 4. Ejercicios

PAN: *Lenguaje de configuración de alto nivel*

2.1 Sentencias y su organización (1 / 2)

- ♦ Principales tipos de sentencias
 - ♦ Include:
 - ♦ include identifier ';'
 - ♦ Asignaciones:
 - ♦ path '=' dml ';' # path es un string
 - ♦ Borrado de un subarbol
 - ♦ delete path ';'

PAN: *Lenguaje de configuración de alto nivel*

2.1 Sentencias y su organización (2 / 2)

- Las sentencias se agrupan en plantillas

- Structure template

```
structure template ram;  
"type" = "DDR";  
"speed" = 133;  
"size" = undef;
```

- Ordinary template

```
template hardware;  
"/hardware/ram/0" = create ("ram");  
"/hardware/ram/1" = create ("ram");
```

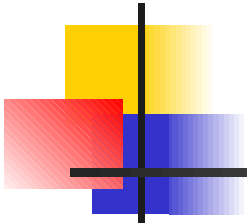
- Declaration template

```
declaration template types;  
define type ram_type = {  
    "type" : string  
    "speed" : long  
    "size" : long  
};  
define type hardware_type {  
    "ram" : ram_type [ ]  
    "total_mem" = long};  
  
define type root_type = {  
    "hardware" : hardware_type };
```

- Object template

```
object template lxxplus001;  
include types;  
type "/" = root_type;  
include hardware;  
"/hardware/ram/0/size" = 128;  
"/hardware/ram/1/size" = 128;  
"/hardware/total_mem" = 256;
```

PAN: *Lenguaje de configuración de alto nivel*



2.2 Comentarios en PAN

- Los nombres de los ficheros que contienen código PAN
 - Coinciden con el identificador que posee el template
 - Poseen la extensión “.tpl”
- Los comentarios se ponen precedidos del símbolo “#”

`“/hardware/ram/0” = create (“ram”) ; # Primer modulo de memoria ram`

PAN: *Lenguaje de configuración de alto nivel*

2.3 Comprobación de tipos

- PAN realiza una fuerte comprobación de tipos, no permitiendo que elementos de distinto tipo se mezclen

```
"/x" = 1 ;    # es un long
```

```
"/x" = "1" ;  # Error: no se puede asignar un string a un long
```

Podemos añadir restricciones a cualquier parte del árbol

```
# restricción: swap será el doble del total de la memoria ram  
define type system_type = {  
  "swap_size" : long with {self == (2* value("/hardware/total_mem"))}  
};
```

PAN: *Lenguaje de configuración de alto nivel*

2.4 Tipos básicos y variables

- Variables
 - No necesitan se declaradas
- Variables globales
 - No se pueden ser redefinida
 - **object**, **self**

```
"swap_size" : long with {self == (2* value("/hardware/total_mem"))};
```


PAN: *Lenguaje de configuración de alto nivel*

2.4 Tipos básicos y variables

- Tipos básicos

- Boolean

True o false

- Long

245 0 0x20 0xdeadb 0755

- Double

0.01 3.1416 1e-8 6.034582e23

- String

'cadena simple' "cadena mas flexible \t y compleja \n"

- Undef

- Representa un elemento indefinido

- Estos para que puedan ser convertidos a LLD debe ser sobrescrito

"ip" = undef;

PAN: *Lenguaje de configuración de alto nivel*

2.5 Operadores

- Operadores

- Aritméticos

Unarios: '-'

Binarios: '+' | '-' | '*' | '/' | '%'

- De strings

Binarios: '+'

- Comparación sobre strings o números

Binarios: '<' | '<=' | '>' | '>=' | '==' | '!='

- Booleanos

Unarios: '!'

Binarios: '&&' | '||'

PAN: *Lenguaje de configuración de alto nivel*

2.6 Tipos de datos complejos

♦ Alias

```
define type ulong1 = long with self >= 0;  
define type ulong1= long (0..); # es equivalente al anterior  
define type port = long (0..65535);
```

♦ Link

```
define type disk_link = long (8..) * with match (self, '^(hda|hdb)$');  
type “/device“ = disk_link;  
“/device“ = “/hda“; # comprobación de que /device sea hda o hdb  
“/hda“= 20; # comprobación valor sea >= 8
```

PAN: *Lenguaje de configuración de alto nivel*

2.6 Tipos de datos complejos

Registro

```
define type network {  
    "ip": string;  
    "gateway": string;  
};
```

list de registros

```
"/profile/network/interfaces/0/ip" = "150.244.10.200"  
"/profile/network/interfaces/0/gateway" = "150.244.10.1"  
"/profile/network/interfaces/1/ip" = "192.168.0.100"  
"/profile/network/interfaces/1/gateway" = "192.168.0.1"
```

nlist de registros

```
"/profile/network/interfaces/eth0/ip" = "150.244.10.200"  
"/profile/network/interfaces/eth0/gateway" = "150.244.10.1"  
"/profile/network/interfaces/eth1/ip" = "192.168.0.100"  
"/profile/network/interfaces/eth1/gateway" = "192.168.0.1"
```

PAN: *Lenguaje de configuración de alto nivel*

2.6 Tipos de datos complejos

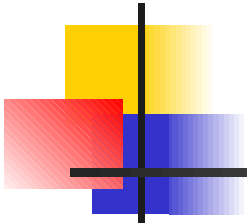
• List

```
dns = string []; # estaría en la declaración de tipos
"/dns/0" = "150.244.10.200";
"/dns/1" = "150.244.10.100";
dns = list ("150.244.10.200", "150.244.10.100");
z = dns[1]; #asignamos a z = "150.244.10.100"
```

• Tabla (nlist)

```
maquinas = string { }; # estaría en la declaración de tipos
"/maquinas/maqueta1" = "192.168.0.100";
"/maquinas/maqueta2" = "192.168.0.101";
maquinas = nlist ("maqueta1", "192.168.0.100", "maqueta2", "192.168.0.101");
ip1 = maquinas["maqueta1"]; ip2 = maquinas["maqueta2"];
```

PAN: *Lenguaje de configuración de alto nivel*



2.7 Control de flujo

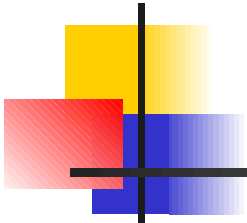
- Secuencial
- Condicional

```
if (value ("/hardware/memory/size") > 256) {  
    swap = 512;  
} else { swap = 256 };
```

- Bucle

```
i=0;  
while (i <= 9) {  
    i= i + 1; };
```

PAN: *Lenguaje de configuración de alto nivel*



2.8 Funciones

```
define function increment = {  
    if (argc==2){  
        val = argv[0] + argv[1];  
        return (val);  
    } else { return 0 };  
};
```

ejemplo de llamada a la funcion

i=3;

x=increment (i,2); # x tendrá el valor 5

PAN: *Lenguaje de configuración de alto nivel*

2.8 Funciones

• **create (name: string,...) : nlist**
• structure template mount_cdrom
 “device” = undef;
 “path” = “/mnt/cdrom”; };
“/system/mounts/0” = create (“mount_cdrom”, “device”, “hdc”);
sería equivalente a
“/system/mounts/0” = create (“mount_cdrom”);
“/system/mounts/0/device” = “hdc”;

• **list (....) : list**

“/system/dns” = list (“150.244.9.200”, “150.244.9.100”);
sería equivalente a
“/system/dns/0” = “150.244.9.200”;
“/system/dns/1” = “150.244.9.100”;

PAN: *Lenguaje de configuración de alto nivel*

2.8 Funciones

• **value (path:string) : element**

• `"/x" = 100;`

`"/y" = 2 * value ("/x"); # /y será 200`

• `"/system/dns" = list ("150.244.9.200","150.244.9.100");`

`z= (value("/system/dns/0") == value("/system/dns/1")) ; # z es false`

sería equivalente a

`v= value("/system/dns");`

`z= (v[0] == v[1]) ; # z es false`

• **match (arg: string, regexp : string): boolean**

• *# device_t es un string que puede ser solo disk, cd*

`device_t : string with match (self, '^(disk|cd)$');`

`device_t = "cdrom"; # Error`

PAN: *Lenguaje de configuración de alto nivel*

2.8 Funciones

- **first (arg:resource, key:identifier, value:identifier) : boolean**
- **next (arg:resource, key:identifier, value:identifier) : boolean**

```
numlist = list (1,2,4,8);  
sum =0;  
ok = first (numlist,iter,v);  
while (ok) {  
    sum = sum + v;  
    ok = next (numlist,iter,v);  
};  
#sum = 15
```

PAN: *Lenguaje de configuración de alto nivel*

1. Introducción a PAN

1.1 CDB y Pan

1.2 Funcionamiento del compilador de Pan (panc)

1.3 Características principales de PAN

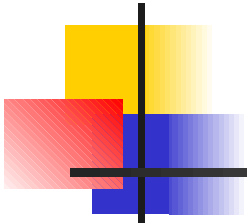
1.4 Forma de representar la información

2. Sintaxis de PAN

3. Comentar un ejemplo de código PAN

4. Ejercicios

PAN: *Lenguaje de configuración de alto nivel*



Conexión

- ♦ Estructura de directorios:

Directorio:

Ejercicio1

Ejercicio2

PAN: *Lenguaje de configuración de alto nivel*

2.6 Comentario de un ejemplo

➤ Ficheros

maqueta.tpl

types.tpl

network.tpl

system.tpl

hardware.tpl

ram.tpl

software.tpl

edg_cdb_1_0_0_1_i386.tpl

edg_pan_2_0_2_1_i386.tpl

PAN: *Lenguaje de configuración de alto nivel*

2.6 Comentario de un ejemplo

➤ Ficheros

maqueta.tpl

types.tpl

network.tpl

system.tpl

hardware.tpl

ram.tpl

software.tpl

edg_cdb_1_0_0_1_i386.tpl

edg_pan_2_0_2_1_i386.tpl

```
template software;  
"/software/packages/edg_cdb_i386" = create  
("edg_cdb_1_0_0_1_i386","arch","i386");  
"/software/packages/edg_pan_i386" = create  
("edg_pan_2_0_2_1_i386","arch","i386");
```

```
structure template edg_cdb_1_0_0_1_i386;  
"name" = "cdb";  
"version" = "1.0.0";  
"arch" = undef;  
"type" = "rpm";
```

```
structure template edg_pan_2_0_2_1_i386;  
"name" = "pan";  
"version" = "2.0.2";  
"arch" = undef;  
"type" = "rpm";
```

PAN: *Lenguaje de configuración de alto nivel*

2.6 Comentario de un ejemplo

➤ Ficheros

maqueta.tpl

types.tpl

network.tpl

system.tpl

hardware.tpl

ram.tpl

software.tpl

edg_cdb_1_0_0_1_i386.tpl

edg_pan_2_0_2_1_i386.tpl

```
template hardware;  
"/hardware/ram" = list (create ("ram","size",128),  
                        create ("ram","size",128));  
"/hardware/total_mem" = 256;
```

```
structure template ram;  
"type" = "DDR";  
"speed" = 133;  
"size" = undef;
```

PAN: *Lenguaje de configuración de alto nivel*

2.6 Comentario de un ejemplo

➤ Ficheros

maqueta.tpl
types.tpl

network.tpl

system.tpl

hardware.tpl
ram.tpl

software.tpl
edg_cdb_1_0_0_1_i386.tpl
edg_pan_2_0_2_1_i386.tpl

```
template network;  
"/network/ip" = undef;  
"/network/netmask" = "255.255.0.0";  
"/network/gateway" = "150.244.10.1";
```

```
template system;  
"/system/swap_size" = 256;
```


PAN: *Lenguaje de configuración de alto nivel*

2.6 Comentario de un ejemplo

◆ Ficheros

maqueta.tpl

types.tpl

network.tpl

system.tpl

hardware.tpl

ram.tpl

software.tpl

edg_cdb_1_0_0_1_i386.tpl

edg_pan_2_0_2_1_i386.tpl

```
declaration template types;
define type network_type = {
    "ip" : string
    "netmask" : string
    "gateway" : string };

define type system_type = {
    "swap_size" : long };

define type ram_type = {
    "type" : string
    "speed" : long
    "size" : long };
define type package = { };
define type software_type = { };
define type hardware_type = { };
define type root_type = {
    "network" : network_type
    "system" : system_type
    "hardware" : hardware_type
    "software" : software_type };
```

PAN: *Lenguaje de configuración de alto nivel*

2.6 Comentario de un ejemplo

➤ Ficheros

maqueta.tpl

types.tpl

network.tpl

system.tpl

hardware.tpl

ram.tpl

software.tpl

edg_cdb_1_0_0_1_i386.tpl

edg_pan_2_0_2_1_i386.tpl

```
object template maqueta;  
include types;  
type "/" = root_type;  
include network;  
include system;  
include hardware;  
include software;  
"/network/ip" = "150.244.10.56";
```

PAN: Lenguaje de configuración de alto nivel

1. Introducción a PAN

1.1 CDB y Pan

1.2 Funcionamiento del compilador de Pan (panc)

1.3 Características principales de PAN

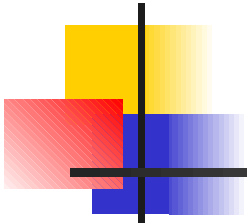
1.4 Forma de representar la información

2. Sintaxis de PAN

3. Comentar un ejemplo de código PAN

4. Ejercicios

PAN: *Lenguaje de configuración de alto nivel*



4. Ejercicios

EJERCICIO 1

1. Escribir los tipos `package_type`, `software_type` y `hardware_type`
2. Añade la validación para que el tamaño del swap (`/system/swap_size`) sea el doble de la memoria ram total (`/hardware/total_mem`)
3. Añade la validación a la definición de `package_type` para que “arch” solo pueda tomar los valores “i386” y “ppc”
4. Añadir la restricción para que tenga ha haber al menos un módulo ram
5. Realiza una función llamada `calc_total` que calcule el tamaño total de memoria ram. La función tendrá dos argumentos (1) `nlist` y (2) `y` un string en el que se indique el nombre de la propiedad (`size`)

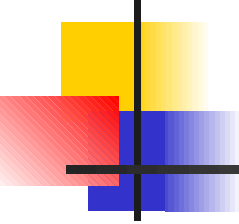
PAN: *Lenguaje de configuración de alto nivel*

4. Ejercicios

Solución 1

```
declaration template types;
.....
define type package_type = {
    "name" : string
    "version" : string
    "arch" : string
    "type" : string
};
define type software_type = {
    "packages" : package_type {}
};
define type hardware_type = {
    "ram": ram_type []
    "total_mem" : long
};
.....
```

PAN: *Lenguaje de configuración de alto nivel*



4. Ejercicios

Solución 2

```
define type system_type = {  
    "swap_size" : long with  
        (self == (2*value ("/hardware/total_mem"))) };
```

Solución 3

```
define type package_type = {  
    .....  
    "arch" : string with match(self , '^(i386|ppc)$')  
    ..... };
```

Solución 4

```
define type hardware_type = {  
    "ram": ram_type [1..]  
    "total_mem" : long  
};
```

PAN: *Lenguaje de configuración de alto nivel*

4. Ejercicios

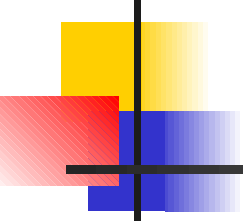
Solución 5

```
template functions;
define function calc_total =
{
    tam = 0;
    ok=first (argv[0],k,v);
    while (ok){
        tam=tam+v[argv[1]];
        ok=next(argv[0],k,v);
    };
    return (tam);
};
```

```
template hardware;
include functions;
"/hardware/ram" = list (
    create ("ram","size",128),
    create ("ram","size",128));

"/hardware/total_mem" =
    calc_total (value ("/hardware/ram"), "size" );
```

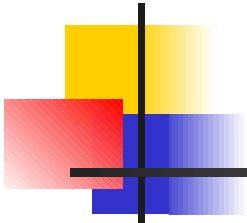
PAN: *Lenguaje de configuración de alto nivel*



Gestión de CDB

- ♦ Gestión Remota: [interfaz SOAP](#)
- ♦ Gestión Local:
 - ♦ Debemos estar conectados como usuario cdb
 - ♦ Utilidad: [cdb-simple-cli](#)
 - `cdb-simple-cli -- help`
 - `cdb-simple-cli -- list`
 - `cdb-simple-cli -- add <object_template.tpl>`

PAN: *Lenguaje de configuración de alto nivel*



4. Ejercicios

EJERCICIO 2

1. Lista los templates insertados en la CDB e inserta el perfil `profile_heppc11.tpl` en la CDB
2. Genera un perfil parecido al existente (`profile_heppc11.tpl`), para una máquina llamada heppc12.ft.uam.es con:
IP="150.244.10.47" y MAC="00-50-04-4B-CD-E0"
3. Añade el perfil creado a CDB
(ver xml en `/opt/edg/var/run/cdb/lld/xml`)

PAN: *Lenguaje de configuración de alto nivel*

4. Ejercicios

profile_heppc11.tpl

pro_declaration_profile_base

.....

pro_hardware_2000

netinfo

pro_type_hepuam1

.....

```
template netinfo;
```

```
"/system/network/domainname" = "ft.uam.es";
```

```
"/system/network/interfaces/eth0/gateway" = "150.244.10.1";
```

```
"/system/network/interfaces/eth0/netmask" = "255.255.255.0";
```

```
"/hardware/location" = "Laboratorio HEP";
```

PAN: *Lenguaje de configuración de alto nivel*

4. Ejercicios

profile_heppc11.tpl

pro_declaration_profile_base

.....

pro_hardware_2000

netinfo

pro_type_hepuam1

.....

```
object template profile_heppc11;
```

```
include pro_declaration_profile_base;
```

```
include pro_hardware_2000;
```

```
include netinfo;
```

```
include pro_type_hepuam1;
```

```
"/system/network/hostname" = "heppc11";
```

```
"/system/network/interfaces/eth0/ip" = "150.244.10.44";
```

```
"/hardware/cards/nic/0/hwid" = "00-50-04-4B-CD-E3";
```