

---

# Sistema de cargas sintéticas

**Diseño y Evaluación de Configuraciones**

**Curso 2012-13**



**Jose M Drake · Miguel Telleria**

drakej AT unican.es telleriam AT unican.es

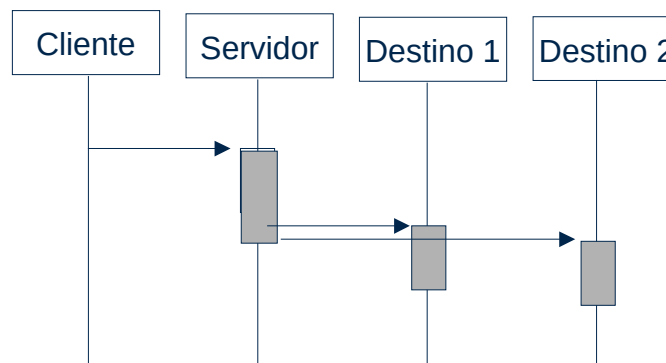
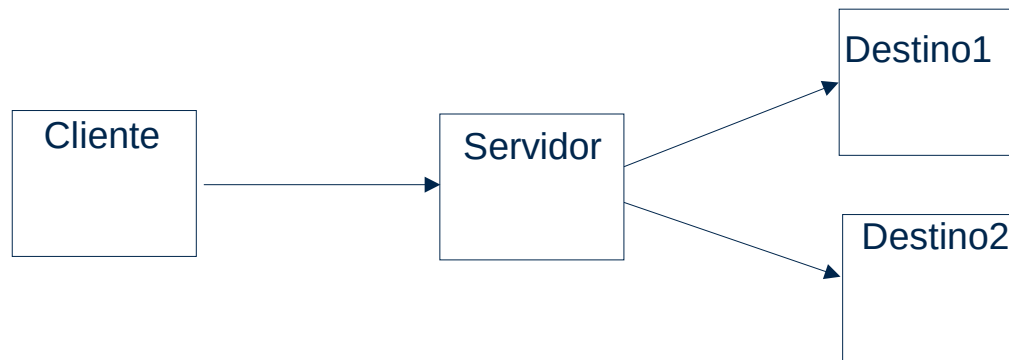
**Computadores y Tiempo Real**

<http://www.ctr.unican.es>

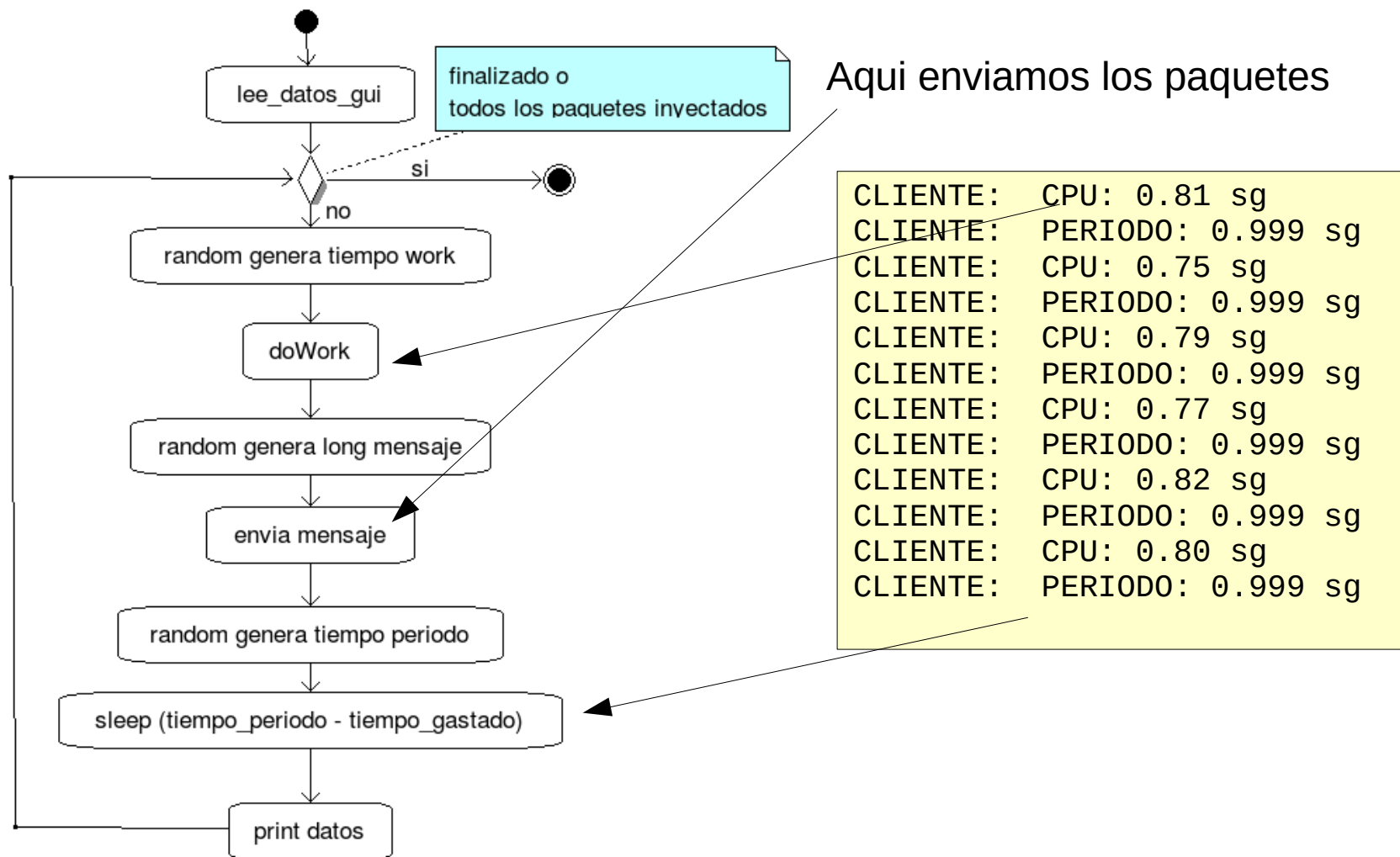
---

# Mecano cliente servidor

- Permite reproducir aplicaciones complejas cliente servidor



# Funcionalidad cliente



# GUI del cliente

The diagram illustrates the GUI for the client, showing various configuration options and distribution types. The GUI is titled "Cliente" and includes the following fields:

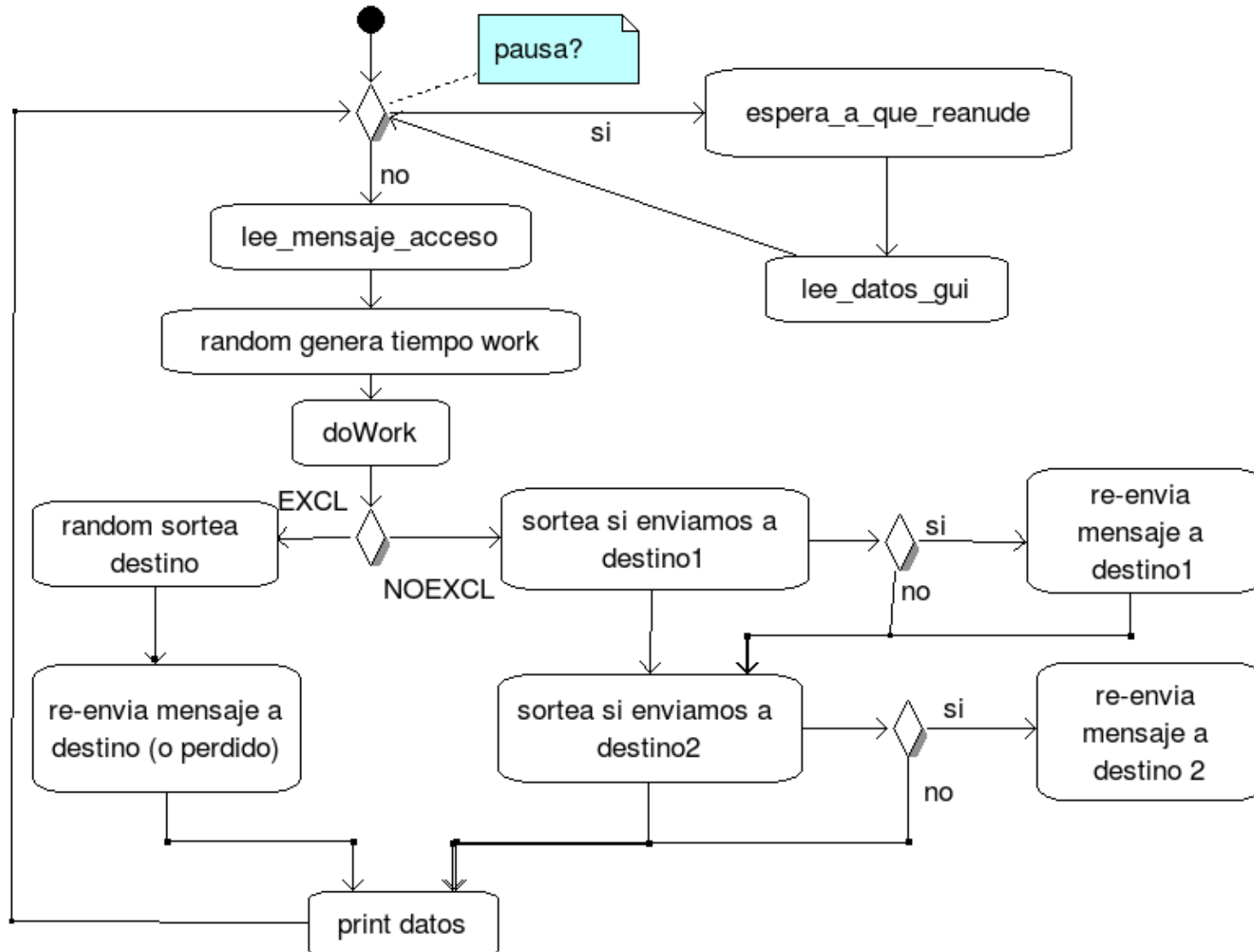
- Nr paquetes (0 ilimitado): 0
- Start / Pausa buttons
- Periodo (sg):
  - Distribucion: **CONSTANTE**
  - Nominal: 1.0
  - Rango: 0.0
- Long mensaje (bytes):
  - Distribucion: UNIFORME
  - Nominal: 256
  - Rango: 0
- Tiempo CPU (sg):
  - Distribucion: EXPONENCIAL
  - Nominal: 0.2
  - Rango: 0.0
- Destino:
  - Direccion IP: 127.0.0.1
  - Puerto: 12000

The distribution types are illustrated with graphs:

- UNIFORME**: A graph showing a constant frequency over a range from  $Frecuencia - Rango/2$  to  $Frecuencia + Rango/2$ .
- CONSTANTE**: A graph showing a constant frequency of 100%.
- EXPONENCIAL**: A graph showing an exponential decay of frequency over time.
- BIPOLAR**: A graph showing a bipolar distribution with a peak at 50% and a range from  $Frecuencia - Rango/2$  to  $Frecuencia + Rango/2$ .

A green box highlights the distribution options: **CONSTANTE**, **UNIFORME**, **EXPONENCIAL**, and **BIPOLAR**.

# Funcionalidad del servidor



# GUI del servidor



```
SERVIDOR proc2-a1: WORK: 0.14 sg enviado a DESTIN01
SERVIDOR proc2-a1: WORK: 0.45 sg enviado a DESTIN01
SERVIDOR proc2-a1: WORK: 0.34 sg enviado a DESTIN02
SERVIDOR proc2-a1: WORK: 0.7 sg enviado a DESTIN02
SERVIDOR proc2-a1: WORK: 0.36 sg enviado a DESTIN02
SERVIDOR proc2-a1: WORK: 0.11 sg enviado a DESTIN01
```

---

# Uso de las cargas sintéticas

---

# Formas de lanzar las cargas sintéticas

- Las clases Cliente y Servicio se pueden arrancar de 4 formas:
  - Desde Eclipse sobre el fichero con Run as Java Application
    - No se pueden poner opciones de arranque
  - Desde Eclipse con el Run configurations
  - Desde Eclipse con el Maletín TPTP (cuando veamos el profiler)
  - Externamente a eclipse desde la shell con el comando
    - `java -classpath <workspace/bin>:<path SWT> <opciones JVM>  
<claseMain> <opciones de arranque>`



# Opciones de comando de las cargas sintéticas

- Se han añadido para hacer más automático el proceso de arrancar las aplicaciones.

- Para el **Servicio**

**nombre**

**modo**

**destino1**

**tiempo\_cpu**

**acceso**

**destino2**

- Para el **Cliente**

**nombre**

**periodo**

**destino**

**tiempo\_cpu**

**long\_mensaje**

- Estas opciones se pueden poner:
  - En los lanzadores de Eclipse (Run Config, Profile Config)
  - En la línea de comando

## Opciones de comando (II) (-help)

Usage: **Cliente** sintetico  
Options include:

-help, -?	displays help information
-nr_paquetes <string>	nr_paquetes inyectados (0 ilimitado)
-periodo <string>	CTE EXP UNI BIP,media,rango
-tiempo_cpu <string>	CTE EXP UNI BIP,media,rango
-long_mensaje <string>	CTE EXP UNI BIP,bytes,rango
-destino <string>	IPdest,Portdest
-nombre <string>	Nombre del cliente

Usage: **Servicio** sintetico  
Options include:

-help, -?	displays help information
-modo <string>	EXCL NOEXCL
-acceso <string>	puerto
-tiempo_cpu <string>	CTE EXP UNI BIP,media,rango
-destino1 <string>	Porcentage,IPdest,Portdest
-destino2 <string>	Porcentage,IPdest,Portdest
-nombre <string>	Nombre del servicio

Nota: Cualquier opción ausente tomará un valor por defecto (no hace falta poner todo)

# Construcción de la línea de comando

- **Ejemplo de Cliente (todo en una línea):**
  - Java + opciones de la JVM
  - Classpath (raíz de los paquetes .class y SWT)
  - Clase main (con su paquete)
  - Opciones de comando

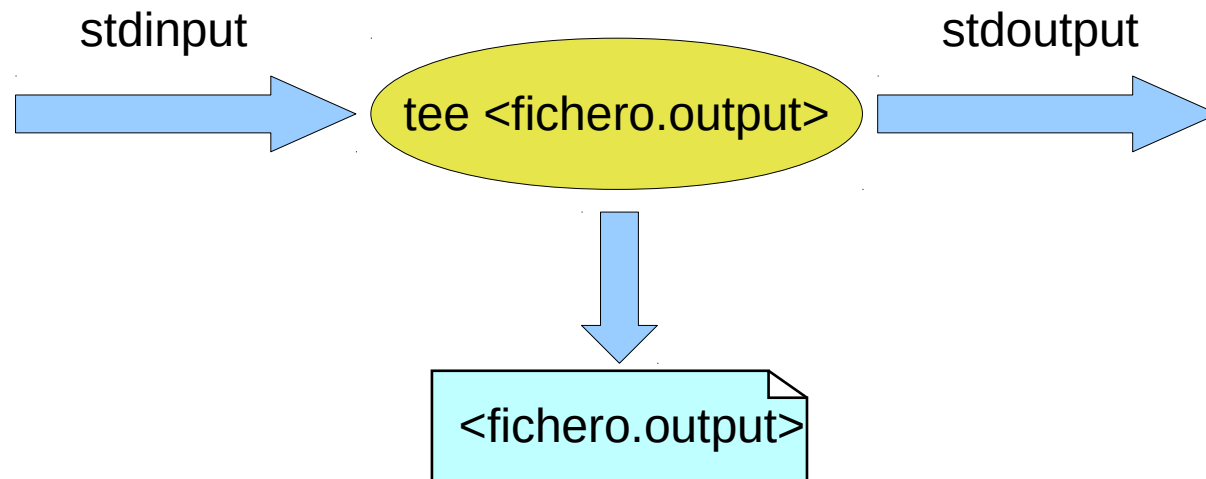
```
java -Dfile.encoding=UTF-8
-classpath /home/miguel/temp/egit/LoadProject/LoadProyect/bin:/usr/share/java/swt.jar
clienteservicio/Cliente
-nombre proc1/a -nr_paquetes 5 -periodo CTE,0.2,0 -tiempo_cpu EXP,0.05,0
-long_mensaje CTE,256,0 -destino 127.0.0.1,12000
```

- **Ejemplo de servicio:**

```
java -Dfile.encoding=UTF-8
-classpath /home/miguel/temp/egit/LoadProject/LoadProyect/bin:/usr/share/java/swt.jar
clienteservicio/Servicio
-nombre proc2/b1 -acceso 12006 -modo NOEXCL -tiempo_cpu UNI,0.5,0.2
-destino1 100,127.0.0.1,12008 -destino2 100,127.0.0.1,12010
```

# Guardar la traza de salida

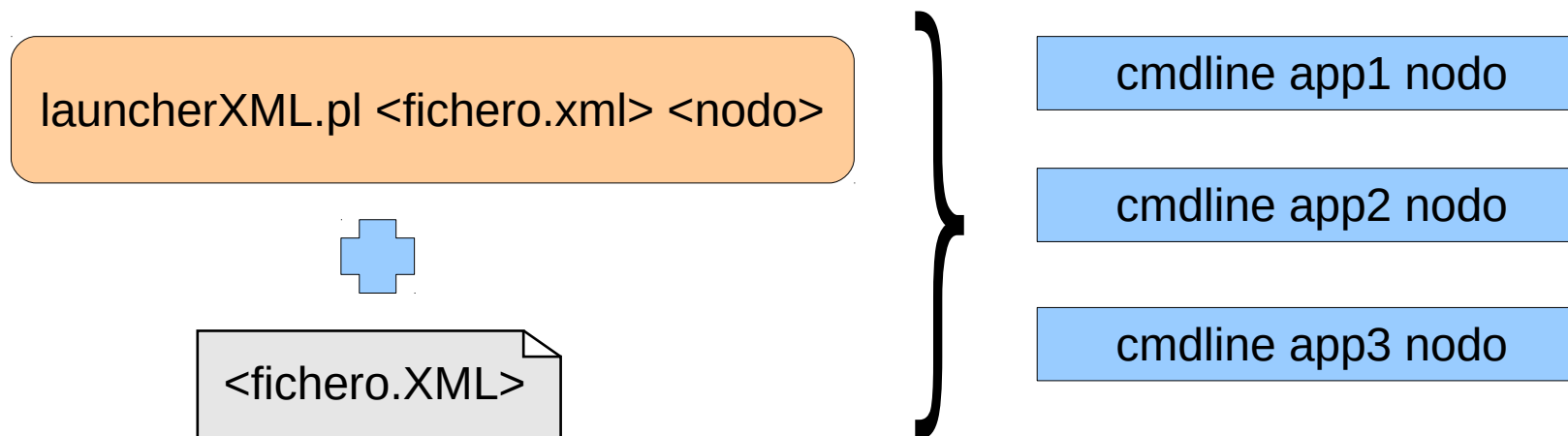
- Utilidad **tee** de linux



```
java -Dfile.encoding=UTF-8
-classpath /home/miguel/temp/egit/LoadProject/LoadProyect/bin:/usr/share/java/swt.jar
clienteservicio/Servicio
-nombre proc2/b1 -acceso 12006 -modo NOEXCL -tiempo_cpu UNI,0.5,0.2
-destino1 100,127.0.0.1,12008 -destino2 100,127.0.0.1,12010
| tee b1.output
```

# LauncherXML.pl

- Utilidad en PERL que genera la línea de comando
  - Relaciona nodos con IP's y puertos (no hace falta repetirlos).
  - Imprime una línea de comando con captura de la salida
  - Lanza el proceso y devuelve el PID resultante.



# Fichero XML de launchXML (nodos y listeners)

```
<?xml version="1.0" encoding="utf-16"?>
<sisistema>
  <nodos nbr="3">
    <nodo id="proc1" ip="127.0.0.1" />
    <nodo id="proc2" ip="127.0.0.1" />
    <nodo id="proc3" ip="127.0.0.1" />
  </nodos>
  <listeners nbr="6">
    <listener id="a1" puerto="12000" />
    <listener id="a2" puerto="12004" />
    <listener id="a3" puerto="12002" />
    <listener id="b1" puerto="12006" />
    <listener id="b2" puerto="12008" />
    <listener id="b3" puerto="12010" />
  </listeners>
</sisistema>
```

# Fichero XML de launchXML (config)

```

<configs nbr="3">
  <config nodo="proc1">
    <app id="a3" class="Servicio" listener="a3">
      <tiempo_cpu distribucion="EXP" nominal="0.03" rango="0" />
    </app>
    <app id="b3" class="Servicio" listener="b3">
      <tiempo_cpu distribucion="UNI" nominal="0.55" rango="0.15" />
    </app>
    <app id="a" class="Cliente">
      <periodo distribucion="CTE" nominal="0.2" rango="0" />
      <tiempo_cpu distribucion="EXP" nominal="0.05" rango="0" />
      <long_mensaje distribucion="CTE" nominal="256" rango="0" />
      <destino nodo="proc2" listener="a1" />
    </app>
  </config>
</configs>
</sistema>

```

definidos en las secciones nodos y listeners

# Utilización de LauncherXML

- Para utilizarlo hay que ajustar:
  - El PATH del workspace en el código Perl de la herramienta

```
my $main_classdir = "/home/alumnos/workspace/LoadProyect/bin";
```
  - Las direcciones IP en el fichero XML de los datos

```
<nodos nbr = "2">
  <nodo id="proc1" ip="172.31.16.65"/>
  <nodo id="proc2" ip="172.31.16.58"/>
</nodos>
```

- Una vez hecho esto se lanza de la forma

```
./launcherXML <fichero_xml> <nombre_nodo> [nombre de app]
```

  - Si no se pone el [nombre de app] se entiende que se lanzan todas

```
./launcherXML.pl practica1.xml proc2
./launcherXML.pl practica1.xml proc1 a1
```