

PROCESADORES DE LENGUAJE

ANT

JUAN DíEZ-YANGUAS BARBER



CONTENIDO

- ❖ ¿Qué es ANT?
- ❖ Instalación de ANT
- ❖ Estructura de un proyecto Java
- ❖ Fichero de configuración
- ❖ Ejemplo sencillo
- ❖ Compilación
- ❖ Empaquetado
- ❖ Documentación
- ❖ Limpieza e inicialización
- ❖ Utilización con librerías externas
- ❖ Ejecución
- ❖ JFlex y CUP

¿QUÉ ES ANT?

- ❖ ANT es una herramienta del proyecto jakarta de apache
- ❖ Equivalente al make de C, más potente y diseñado pensando en Java
- ❖ Desarrollado en Java, no depende de las órdenes de la Shell de cada sistema
- ❖ Se basa en un fichero xml de configuración (build.xml)

REQUISITOS PARA LA INSTALACIÓN

- ❖ ANT: <http://ant.apache.org/bindownload.cgi>
- ❖ JDK (no deja de ser una aplicación escrita en Java)
- ❖ Parser XML (si se baja la versión binaria de ANT ya viene incluido)

INSTRUCCIONES DE INSTALACIÓN

❖ Instrucciones: <http://ant.apache.org/manual/index.html>



Asumiendo que ANT esta en c:\ant\

```
set ANT_HOME=c:\ant  
set JAVA_HOME=c:\jdk-1.5.0.05  
set PATH=%PATH%;%ANT_HOME%\bin
```

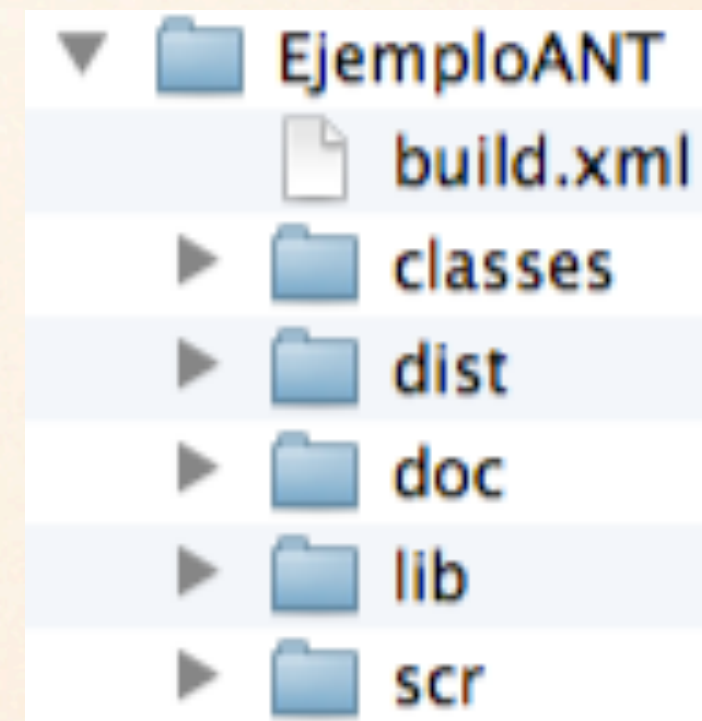


Asumiendo que ANT esta en /usr/local/ant

```
export ANT_HOME=/usr/local/ant  
export JAVA_HOME=/usr/local/jdk-1.5.0.05  
export PATH=${PATH}:${ANT_HOME}/bin
```


ESTRUCTURA DE UN PROYECTO JAVA

- ❖ **classes:** Clases Java compiladas
- ❖ **dist:** Contiene un jar con los archivos `.class` y lo necesario para ejecutar la aplicación desde el mismo jar
- ❖ **src:** Contiene el código fuente organizado en paquetes
- ❖ **doc:** Contiene la documentación del proyecto en formato javadoc
- ❖ **lib:** Librerías externas usadas



FICHERO DE CONFIGURACIÓN

❖ **project**

- ❖ Elemento raíz del fichero xml
- ❖ Solo puede haber uno en el fichero, define al propio proyecto
- ❖ Atributos: **name**, **default**, **basedir**

```
<project name="ProbandoAnt" default="todo" basedir=".">
```


FICHERO DE CONFIGURACIÓN

- ❖ **property** o propiedad
 - ❖ Parámetros en forma de par clave-valor (case-sensitive)
 - ❖ Para personalizar el proceso de construcción
 - ❖ Útiles para acceder a un dato de manera repetitiva
 - ❖ Atributos: **name**, **value**

```
<property name="fuente" value="scr" />
```


FICHERO DE CONFIGURACIÓN

❖ **target** u objetivo

- ❖ Conjunto de tareas a aplicar a la aplicación en algún momento
- ❖ Pueden existir dependencias entre objetivos
- ❖ Atributos: **name**, **depends**, **description**

```
<target name="ejemplo" depends="dependencia" description="Ejemplo" >  
</target>
```

FICHERO DE CONFIGURACIÓN

❖ **task** o tarea

- ❖ Código ejecutable a aplicar a la aplicación.
- ❖ Hay numerosas implementadas por defecto
- ❖ Puede contener distintas propiedades (ej: classpath)
- ❖ Atributos: dependientes de la tarea

```
<echo message="Tarea sencilla" />
```


EJEMPLO INICIAL

```
<?xml version="1.0" encoding="UTF-8" ?>
```

```
<project name="ProbandoAnt" default="mensaje" basedir=".">
```

```
<!-- propiedades globales del proyecto -->
```

```
<property name="fuente" value="scr" />
```

```
<!-- Objetivo -->
```

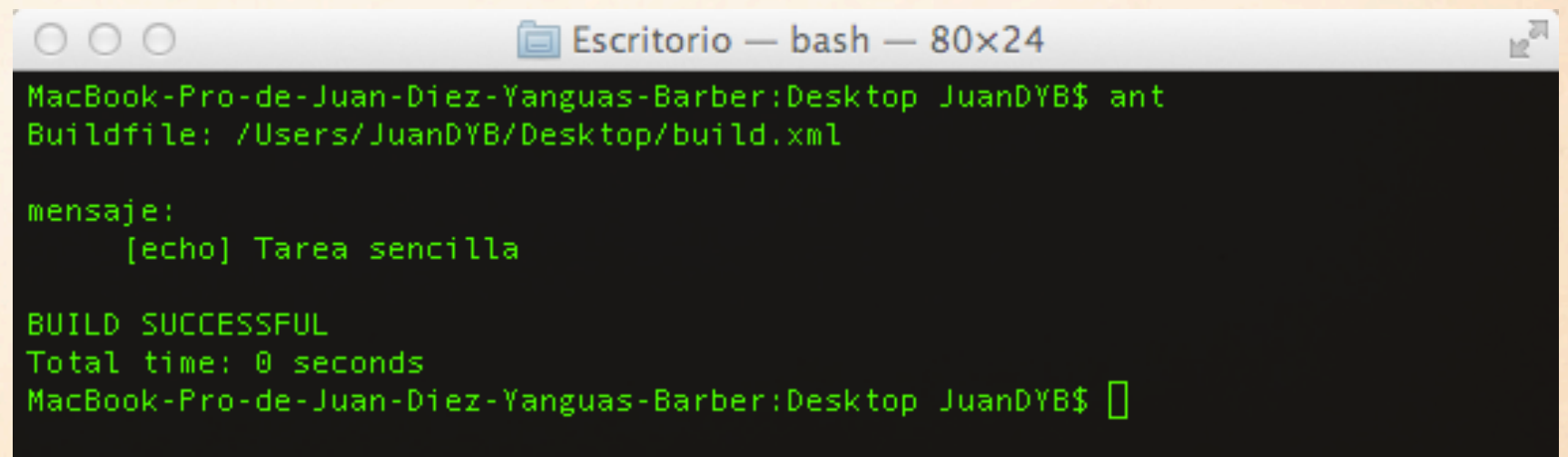
```
<target name="mensaje">
```

```
<!-- Tarea -->
```

```
<echo message="Tarea sencilla" />
```

```
</target>
```

```
</project>
```



```
Escritorio — bash — 80x24
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:Desktop JuanDYB$ ant
Buildfile: /Users/JuanDYB/Desktop/build.xml

mensaje:
    [echo] Tarea sencilla

BUILD SUCCESSFUL
Total time: 0 seconds
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:Desktop JuanDYB$
```

DEPENDENCIAS

- ❖ Una dependencia implica que un objetivo no será ejecutado sin antes haberse ejecutado los objetivos de los que depende

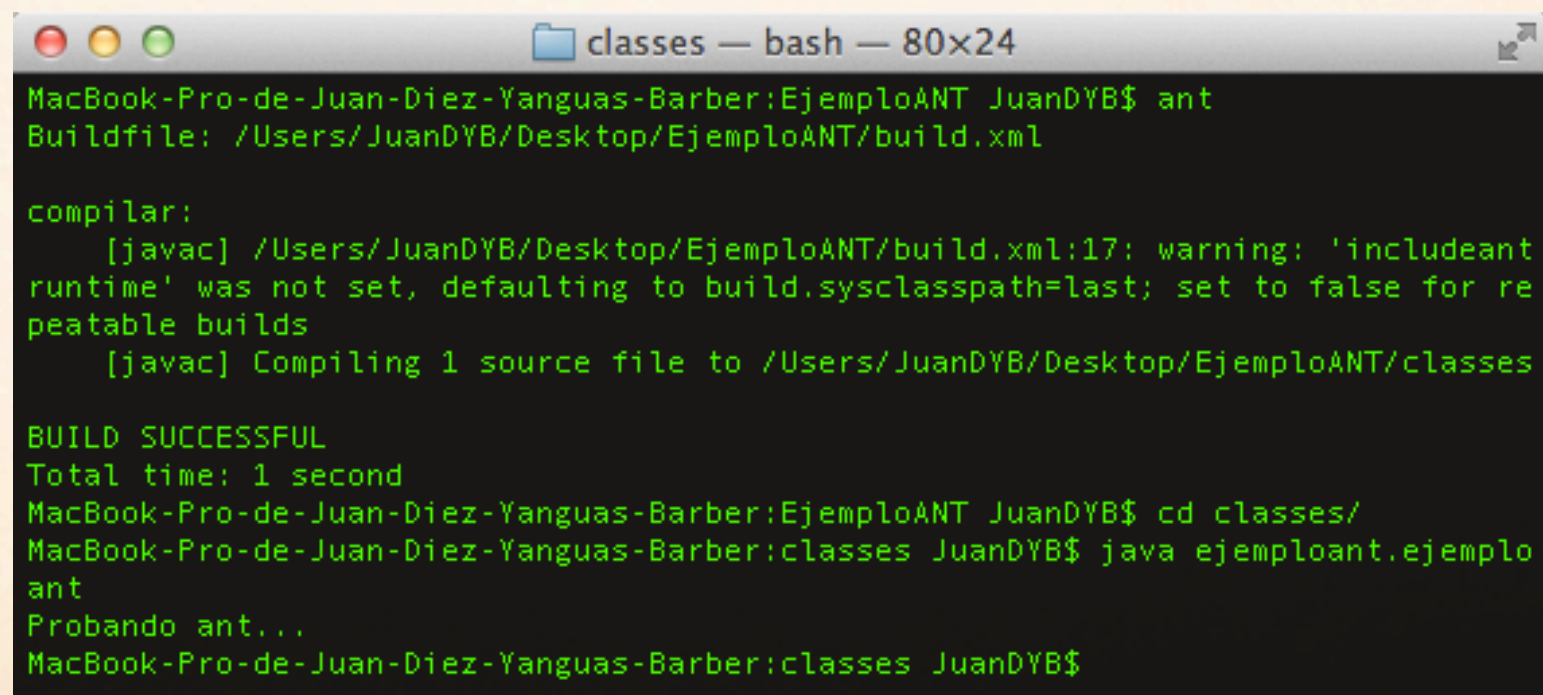
```
<target name="empaqueta" depends="compila">  
    <!-- Tareas a realizar -->  
</target>
```

```
<target name="compila">  
    <!-- Tareas a realizar -->  
</target>
```


COMPILACIÓN

- ❖ Para la compilación usar la tarea javac
- ❖ Atributos: `srcdir`, `destdir`

```
<javac srcdir="${fuente}" destdir="${destino}" />
```



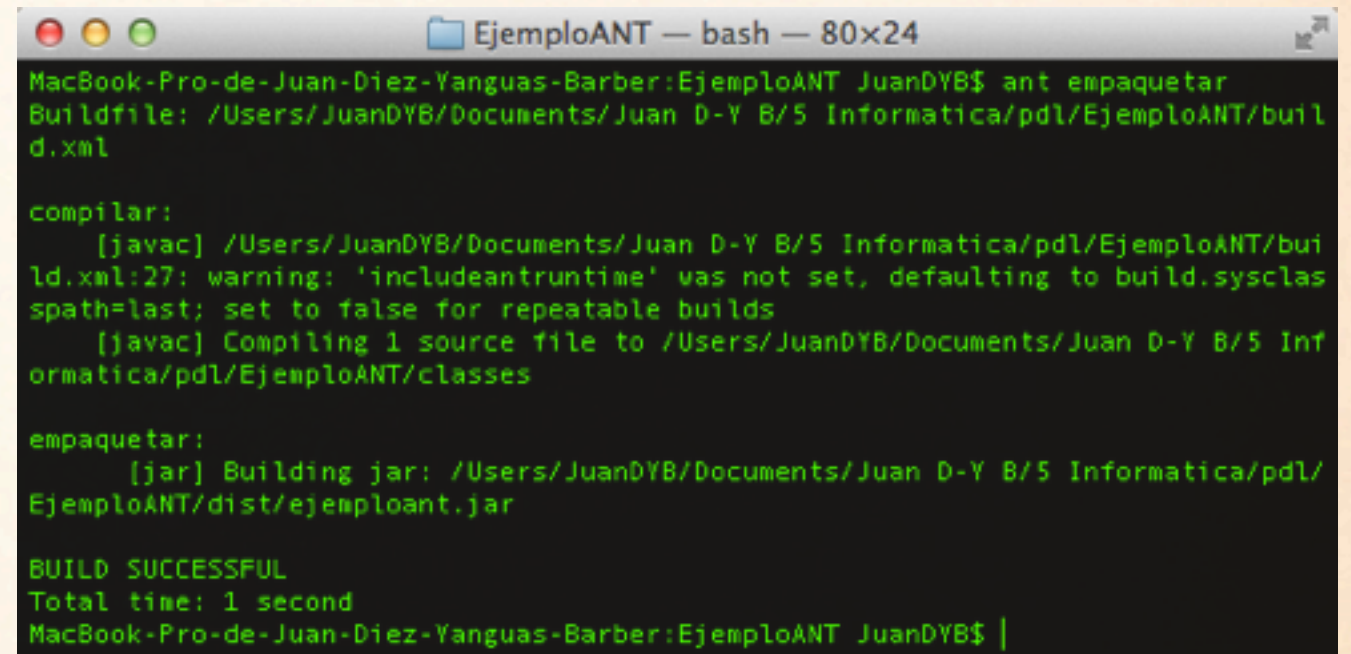
```
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$ ant
Buildfile: /Users/JuanDYB/Desktop/EjemploANT/build.xml

compilar:
[javac] /Users/JuanDYB/Desktop/EjemploANT/build.xml:17: warning: 'includeant
runtime' was not set, defaulting to build.sysclasspath=last; set to false for re
peatable builds
[javac] Compiling 1 source file to /Users/JuanDYB/Desktop/EjemploANT/classes

BUILD SUCCESSFUL
Total time: 1 second
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$ cd classes/
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:classes JuanDYB$ java ejemploant.ejemplo
ant
Probando ant...
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:classes JuanDYB$
```

EMPAQUETADO

- ❖ Tarea **jar** indicando archivos a incluir y el archivo destino
- ❖ Indicar meta-información para crear un archivo jar autoejecutable. Elemento **manifest**



```
EjemploANT — bash — 80x24
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$ ant empaquetar
Buildfile: /Users/JuanDYB/Documents/Juan D-Y B/5 Informatica/pdL/EjemploANT/build.xml

compiler:
[javac] /Users/JuanDYB/Documents/Juan D-Y B/5 Informatica/pdL/EjemploANT/build.xml:27: warning: 'includeantruntime' was not set, defaulting to build.sysclasspath=last; set to false for repeatable builds
[javac] Compiling 1 source file to /Users/JuanDYB/Documents/Juan D-Y B/5 Informatica/pdL/EjemploANT/classes

empaquetar:
[jar] Building jar: /Users/JuanDYB/Documents/Juan D-Y B/5 Informatica/pdL/EjemploANT/dist/ejemploant.jar

BUILD SUCCESSFUL
Total time: 1 second
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$
```

```
<target name="empaquetar" depends="compilar">
  <jar destfile="${dist}/ejemploant.jar" basedir="${destino}" includes="**">
    <manifest>
      <attribute name="Built-By" value="${user.name}" />
      <attribute name="Main-Class" value="ejemploant.ejemploant" />
    </manifest>
  </jar>
</target>
```


DOCUMENTACIÓN

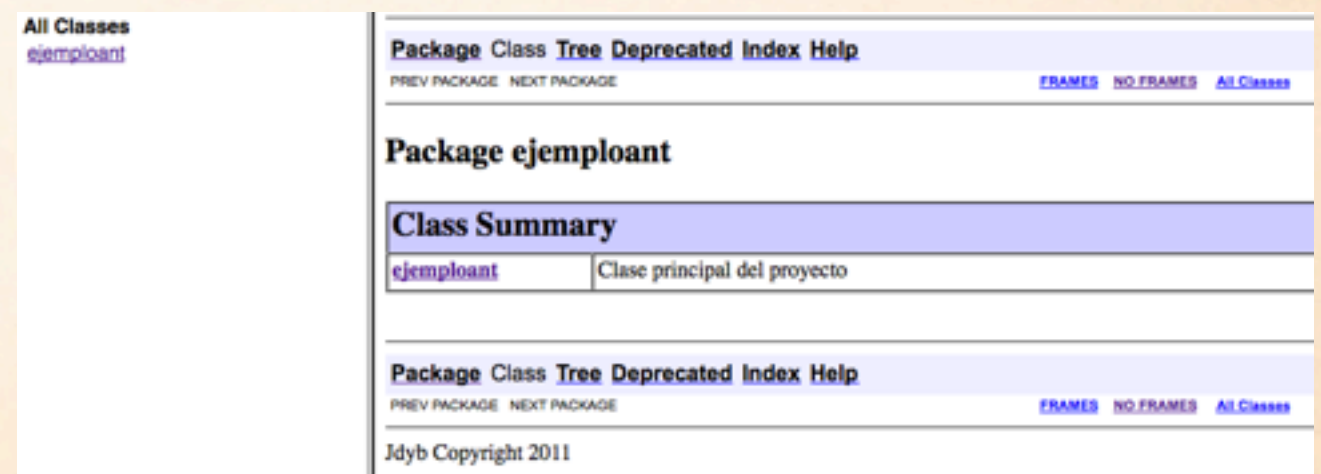
- ❖ Tarea **javadoc** indicando fuentes donde se encuentra la documentación
- ❖ Es posible la personalización de la documentación con ciertos atributos

```
EjemploANT — bash — 80x24
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$ ant documentar
Buildfile: /Users/JuanDYB/Documents/Juan D-Y B/5 Informatica/pd1/EjemploANT/build.xml

documentar:
[javadoc] Generating Javadoc
[javadoc] Javadoc execution
[javadoc] Loading source files for package ejemploant...
[javadoc] Constructing Javadoc information...
[javadoc] Standard Doclet version 1.6.0_29
[javadoc] Building tree for all the packages and classes...
[javadoc] Building index for all the packages and classes...
[javadoc] Building index for all classes...

BUILD SUCCESSFUL
Total time: 2 seconds
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$ |
```

```
<target name="documentar">
  <javadoc packagenames="*"
    sourcepath="${fuente}"
    destdir="${javadoc}"
    author="true"
    version="true"
    private="true"
    locale="es"
    windowtitle="Prueba de ANT"
    doctitle="Prueba de ANT"
    bottom="Jdyb Copyright 2011">
  </javadoc>
</target>
```

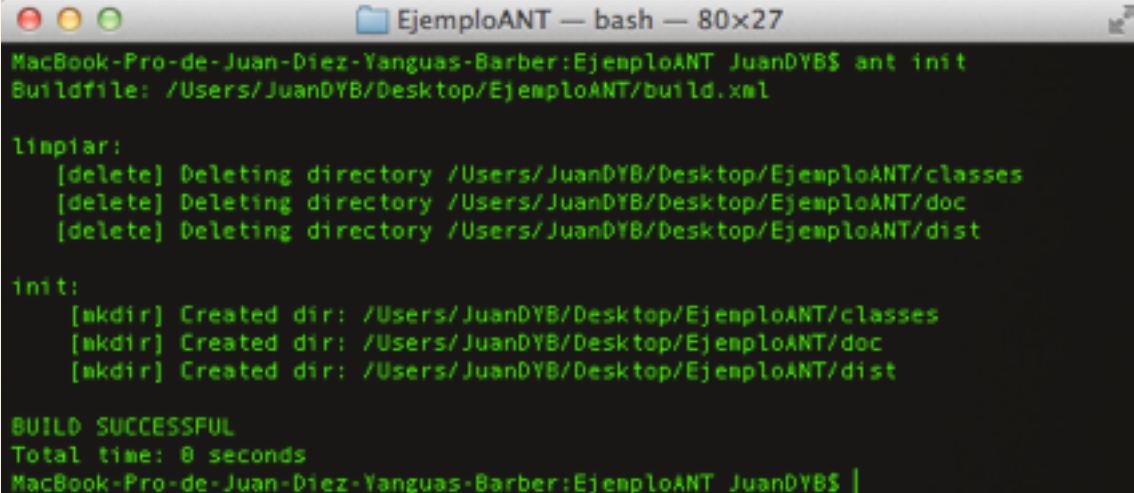


LIMPIEZA E INICIALIZACIÓN

- ❖ Con la tarea **delete** se pueden borrar directorios
- ❖ Con la tarea **mkdir** se pueden crear directorios

```
<target name="limpiar">
    <delete dir="${destino}" />
    <delete dir="${javadoc}" />
    <delete dir="${dist}" />
</target>

<target name="init" depends="limpiar">
    <mkdir dir="${destino}" />
    <mkdir dir="${javadoc}" />
    <mkdir dir="${dist}" />
</target>
```



The screenshot shows a terminal window titled "EjemploANT — bash — 80x27". The user runs the command "ant init". The output shows the "limpiar" target deleting three directories: "/Users/JuanDYB/Desktop/EjemploANT/classes", "/Users/JuanDYB/Desktop/EjemploANT/doc", and "/Users/JuanDYB/Desktop/EjemploANT/dist". Then, the "init" target creates these three directories. The build concludes with "BUILD SUCCESSFUL" and a total time of 8 seconds.

```
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$ ant init
Buildfile: /Users/JuanDYB/Desktop/EjemploANT/build.xml

limpiar:
[delete] Deleting directory /Users/JuanDYB/Desktop/EjemploANT/classes
[delete] Deleting directory /Users/JuanDYB/Desktop/EjemploANT/doc
[delete] Deleting directory /Users/JuanDYB/Desktop/EjemploANT/dist

init:
[mkdir] Created dir: /Users/JuanDYB/Desktop/EjemploANT/classes
[mkdir] Created dir: /Users/JuanDYB/Desktop/EjemploANT/doc
[mkdir] Created dir: /Users/JuanDYB/Desktop/EjemploANT/dist

BUILD SUCCESSFUL
Total time: 8 seconds
MacBook-Pro-de-Juan-Diez-Yanguas-Barber:EjemploANT JuanDYB$
```


UTILIZACIÓN CON LIBRERÍAS EXTERNAS

- ❖ Con el elemento `path` se hace referencia a un conjunto de archivos y será posible su uso en varios objetivos
- ❖ Seleccionamos con `pathelement` el directorio de nuestro proyecto
- ❖ Usamos el elemento `fileset` para seleccionar un conjunto de archivos
- ❖ Es posible el uso de filtros para seleccionar únicamente determinados tipos de archivos. Usamos el elemento `include`

```
<path id="path.libs">  
  <pathelement location="." />  
  <fileset dir="lib">  
    <include name="**/*.jar" />  
  </fileset>  
</path>
```

COMPILACIÓN Y DOCUMENTACIÓN CON LIBRERÍAS EXTERNAS

- ❖ Usaremos el id que hemos dado al path de librerías para añadirlo al **classpath**

```
<target name="compilar" depends="init">
  <javac srcdir="${fuente}" destdir="${destino}" >
    <classpath refid="path.libs" />
  </javac>
</target>
```

```
<target name="documentar">
  <javadoc packagenames="*"
    sourcepath="${fuente}"
    destdir="${javadoc}"
    author="true"
    version="true"
    private="true"
    locale="es"
    windowtitle="Prueba de ANT"
    doctitle="Prueba de ANT"
    bottom="Jdyb Copyright 2011">
    <classpath refid="path.libs" />
  </javadoc>
</target>
```


EMPAQUETADO CON LIBRERÍAS EXTERNAS

- ❖ La inclusión de un jar en el paquete final implica la previa descompresión del mismo. Usar `zipfileset`
- ❖ Como consecuencia del paso anterior podríamos reemplazar por accidente la carpeta META-INF de nuestro paquete final

```
<target name="empaquetar" depends="compilar">
  <jar destfile="${dist}/ejemploant.jar" basedir="${destino}" includes="**">
    <manifest>
      <attribute name="Built-By" value="${user.name}" />
      <attribute name="Main-Class" value="ejemploant.ejemploant" />
    </manifest>
    <zipfileset src="lib/jdom.jar" excludes="META-INF/*" />
  </jar>
</target>
```

EJECUCIÓN DE CLASE JAVA

- ❖ Tarea `java` para ejecución de una clase java
- ❖ Atributo `classname` define el nombre de la clase a ejecutar
- ❖ Atributo `classpath` define el directorio donde se encuentra la clase a ejecutar
- ❖ Elemento `arg` define los argumentos con el atributo `value`
- ❖ Referencia a las librerías externas

```
<target name="run" depends="compilar" >  
    <java classname="ejemploant.ejemploant" classpath="${destino}">  
        <arg value="build.xml"/>  
        <classpath refid="path.libs" />  
    </java>  
</target>
```


EJECUCIÓN DE ARCHIVO JAR

- ❖ Tarea **java** para la ejecución de un archivo jar
- ❖ Indicar correctamente la clase `main` en el manifiesto
- ❖ Atributo **jar** indica el fichero a ejecutar
- ❖ Obligatorio usar el atributo **fork** puesto que se va a ejecutar una nueva instancia de la VM
- ❖ Usar **maxmemory** si se quiere especificar el tamaño de memoria a reservar de la nueva VM

```
<target name="runjar" depends="empaquetar">  
    <java jar="${dist}/ejemploant.jar" fork="true" maxmemory="128m">  
        <arg value="build.xml"/>  
        <classpath refid="path.libs" />  
    </java>  
</target>
```

USO DE ANT CON JFLEX

- ❖ JFlex tiene definida una tarea de ant dentro de paquete.
- ❖ Invocar la tarea con la directiva `taskdef`
- ❖ Usar la tarea dentro de un objetivo con el nombre que se le ha dado a la tarea

```
<taskdef classpathref="path.libs" classname="JFlex.anttask.JFlexTask" name="jflex" />
```

```
<target name="generateGrammar">  
    <jflex file="${grammar}/reglas.flex" destdir="${fuente}"/>  
</target>
```


USO DE ANT CON CUP

- ❖ CUP, por su parte también cuenta con una tarea definida para ant
- ❖ Invocaremos la tarea con `taskdef`
- ❖ Usar la tarea creada con el nombre dado dentro de un objetivo
- ❖ Si se quiere crear una interfaz en vez de una clase usar atributo `interface`

```
<taskdef classpathref="path.libs" classname="java_cup.anttask.CUPTask" name="cup" />
```

```
<target name="generateGrammar">  
    <cup srcfile="${grammar}/Gram.cup" destdir="${fuente}"/>  
</target>
```

BIBLIOGRAFÍA

- ❖ Javahispano: http://www.javahispano.com/contenidos/es/introduccion_a_ant/
- ❖ Chuidiang: <http://www.chuidiang.com/java/herramientas/ant.php>
- ❖ Gustavo Peiretti: <http://gustavopeiretti.com/2009/crear-jar-con-librerias-externas-usando-ant/>
- ❖ Manual ANT: <http://ant.apache.org/manual/index.html>
- ❖ Manual CUP: <http://www2.cs.tum.edu/projects/cup/manual.html#ant>
- ❖ Manual JFlex: http://jflex.de/jflex_anttask.html

PREGUNTAS



RECURSOS

❖ Recursos usados en la presentación

❖ JFlex: <http://jflex.de/jflex-1.4.3.zip>

❖ CUP: <http://www2.cs.tum.edu/projects/cup/java-cup-11a.jar>

❖ Recursos para el alumno

❖ Explicaciones completas: <http://blog.jdiezfoto.es/informatica/tutorial-ant-uso-con-librerias-externas-y-generadores-jflex-y-cup/>

❖ Ejemplo completo: <http://blog.jdiezfoto.es/wp-content/uploads/2011/11/EjemploANT.zip>