

JAVA y TOMCAT



Ponentes:

BRIO

Gaper



LAMP y desarrollo en
plataformas libres



Preparandonos para Java



Instalando J2SE

xhost +

su -c “./jdk1.5.0_06.bin”

- J2SE
- J2EE
- J2ME

Es necesario configurar las variables de entorno:

1.- Editamos /root/.bashrc

```
PATH=$PATH:/opt/jdk1.5.0_06/bin/  
export PATH
```

Esto nos permite contar con JAVA dentro del PATH para poderlo ejecutar.





Preparandonos para Tomcat

Descomprimos tomcat:

```
tar -xvzf apache-tomcat-5.5.16.tar.gz
```

lo movemos a /usr/local

```
mv apache-tomcat-5.5.16 /usr/local/tomcat
```

agregamos las variables de entorno en /root/.bashrc

```
export JAVA_HOME=/opt/jdk1.5.0_06
```

```
export CATALINA_HOME=/usr/local/tomcat;
```





Probando Tomcat:

El script de ejecución de tomcat nos permite iniciar el servidor:

```
$CATALINA_HOME/bin/startup.sh
```

Para detenerlo contamos con un script especial:

```
$ CATALINA_HOME/bin/shutdown.sh
```

Una vez ejecutándose podemos ver el resultado en el browser:

Archivo Editar Ver Ir Marcadores Herramientas Ayuda

Comenzar con Firefox ELPAIS.es: El diario di...

Apache Tomcat 5.5.16

The Apache Software Foundation
<http://www.apache.org/>

Administration

- [Status](#)
- [Tomcat Administration](#)
- [Tomcat Manager](#)

Documentation

- [Release Notes](#)
- [Change Log](#)
- [Tomcat Documentation](#)

Tomcat Online

- [Home Page](#)
- [FAQ](#)
- [Bug Database](#)
- [Open Bugs](#)
- [Users Mailing List](#)
- [Developers Mailing List](#)

If you're seeing this page via a web browser, it means you've setup Tomcat successfully. Congratulations!

As you may have guessed by now, this is the default Tomcat home page. It can be found on the local filesystem at

```
$CATALINA_HOME/webapps/ROOT/index.jsp
```

where "\$CATALINA_HOME" is the root of the Tomcat installation directory. If you're seeing this page, and you don't think you should be, then either you're either a user who has arrived at new installation of Tomcat, or you're an administrator who hasn't got his/her setup quite right. Providing the latter is the case, please refer to the [Tomcat Documentation](#) for more detailed setup and administration information than is found in the INSTALL file.

NOTE: This page is precompiled. If you change it, this page will not change since it was compiled into a servlet at build time. (See `$CATALINA_HOME/webapps/ROOT/WEB-INF/web.xml` as to how it was mapped.)

NOTE: For security reasons, using the administration webapp is restricted to users with role "admin". The manager webapp is restricted to users with role "manager". Users are defined in `$CATALINA_HOME/conf/tomcat-users.xml`.



TOMCAT



Servidor de aplicaciones, funciona como contenedor de servlets y JSP's

Tomcat no funciona con cualquier servidor web con soporte para servlets y JSPs. Tomcat incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets del Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede, asimismo, funcionar como servidor web por sí mismo. Opera de tal manera en entornos de desarrollo poco exigentes en términos de velocidad y de manejo de transacciones.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual.



TOMCAT



La jerarquía de directorios de instalación de Tomcat incluye:

- bin - arranque, cierre, y otros scripts y ejecutables
- common - clases comunes que pueden utilizar Catalina y las aplicaciones web
- conf - ficheros XML y archivos para la configuración de Tomcat
- logs - logs de Catalina y de las aplicaciones
- server - clases utilizadas solamente por Catalina
- shared - clases compartidas por todas las aplicaciones web
- webapps - directorio que contiene las aplicaciones web
- work - almacenamiento temporal de ficheros y directorios



Apache & Tomcat



El funcionamiento principal de Apache desde su creación fue la de aceptar y responder requisiciones de Páginas en Internet estas requisiciones correspondían a documentos estáticos (puro HTML), es por esto que cuando se requiere ejecutar algún tipo de contenido dinámico (programas) como "Java", es necesario coordinar los esfuerzos de Apache con otro ambiente, en el caso de "Java" es precisamente "Tomcat" quien ofrece facilidades para ejecutar los dos componentes más utilizados en ambientes "Java": "JSP ("Java Server Pages")" y "Servlets".

La comunicación entre Apache y tomcat se hace por medio del módulo mod_jk



Tomcat & PHP



Instalación de PHP:

```
./configure --with-servlet=$TOMCAT_HOME --with-java=$JAVA_HOME  
make
```

Se crearán 2 archivos sapi/servlet/phpsrvlt.jar y libs/libphp4.so

Copiar el archivo phpsrvlt.jar en \$TOMCAT_HOME/common/lib

Copiar la sección de servlet y de servlet-mapping en el archivo
\$PHP_HOME/sapi/servlet/web.xml al web.xml de tomcat \$TOMCAT_HOME/conf/web.xml



Tomcat & PHP



Incluir al path la librería creada en el primer paso (libphp4.so)

```
LD_LIBRARY_PATH=$PHP_HOME/libs
```

```
export LD_LIBRARY_PATH
```

Para comprobar que la instalación se hizo correctamente crear un archivo info.php que contenga:

```
<?php phpinfo();?>
```

Al entrar a <http://127.0.0.1:8080/info.php> deberá aparecer información acerca del PHP instalado.





JAVA

¿Qué es Java?

Java, Pasado y presente

Arquitectura del lenguaje

Arquitectura de Seguridad del lenguaje





Arquitectura del lenguaje

Java Virtual Machine

Bytecodes

Cargador de clases

JAR – Java Archive - Firmados y paquetes sellados

Otras características – Arrays - Jit





Arquitectura de seguridad

Código local y código remoto – Firmas

Caja de arena de seguridad – Sandbox model

Características del lenguaje/compilador

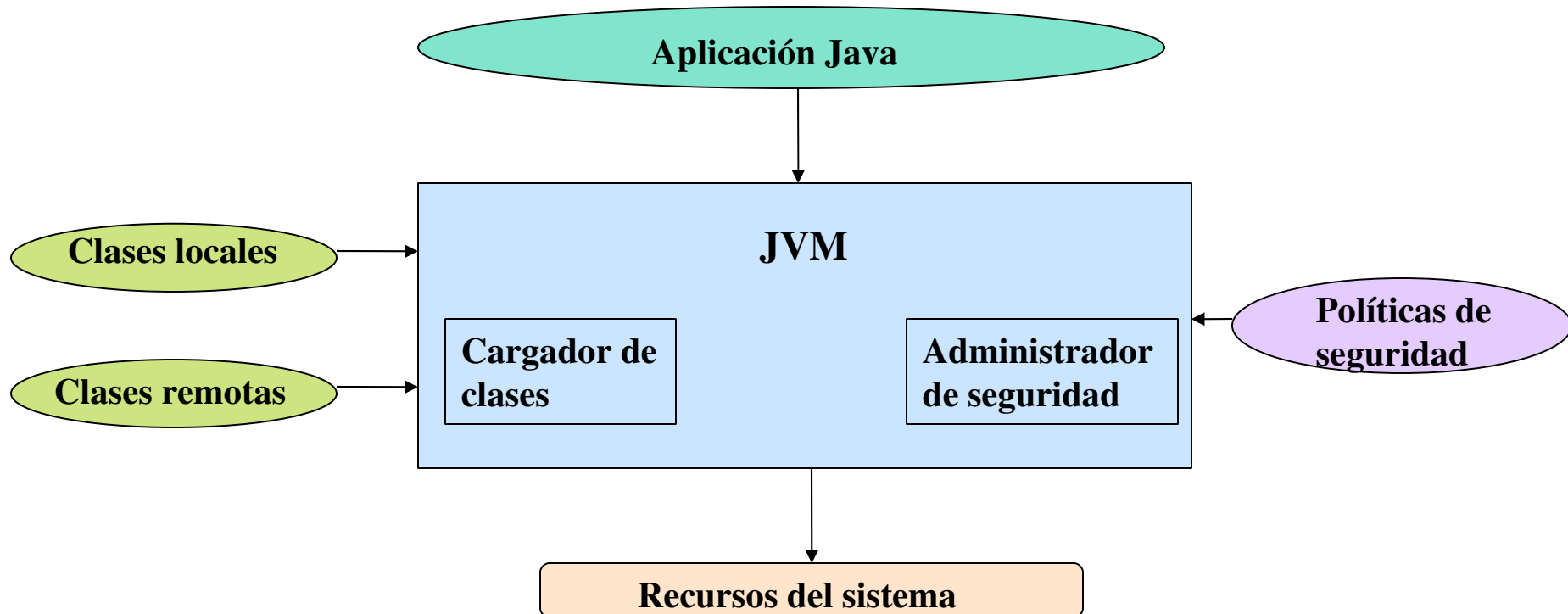
Verificador de código de bytecodes

Cargador de clases

Gestor de Seguridad



Arquitectura de seguridad





Dominios de Protección

Dominios de Protección

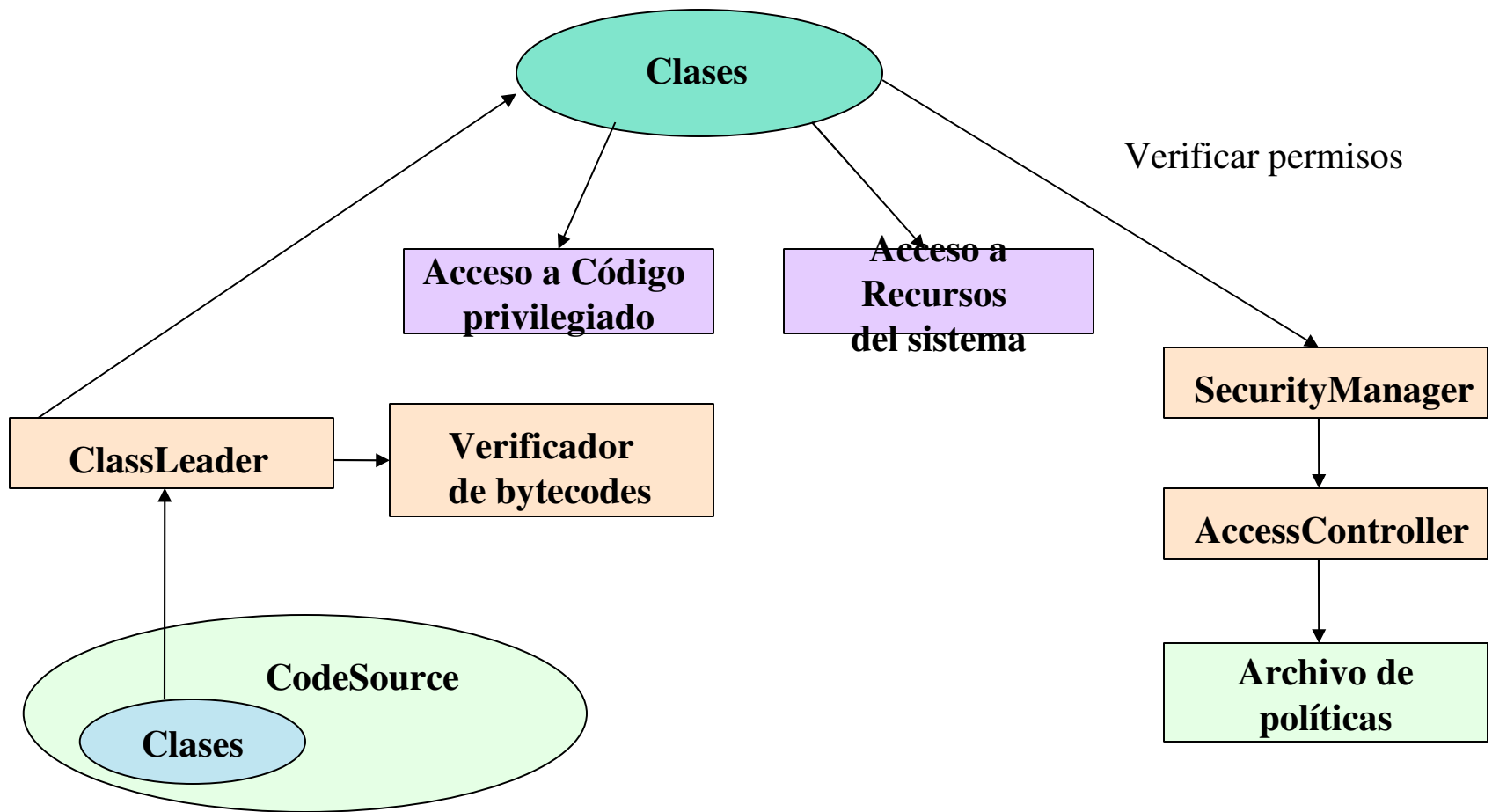
Seguridad de aplicación - A quien se le permite ejecutar qué

Seguridad de Sistema - A qué recursos se puede acceder

Es un conjunto de tecnologías, API y arquitectura de java básica que se combinan para proporcionar a la aplicación Java un entorno seguro



Dominios de Protección





Dominios de Protección

Tipos de dominios de protección

Del Sistema

Controla el acceso a recursos como pueden ser el sistema de archivos ó las conexiones de red.

De la aplicación

Controla el acceso a diversos componentes o partes de una aplicación.





Administrador de seguridad

Todo el código del JDK es código del sistema y se considera fiable, por lo que se le otorgan todos los permisos. Por omisión el resto del código de la aplicación no se ejecuta bajo el control de un administrador de seguridad.

Se puede cargar opcionalmente un administrador de seguridad.

Herramientas con las cuales se puede hacer:

JAAS

Roles de seguridad de J2EE





Permisos comunes asociados a recursos del sistema

Los permisos que se asignan mediante la instrucción `grant` son subclases de la clase *java.security.Permission*

Permisos mas comunes:

FilePermission : *Identifica los archivos y directorios a los que se puede tener acceso utilizando las funciones de entrada y salida a archivos y a qué tipos de accesos se permiten.*

SocketPermission: *Identifica los puertos de red a los que se puede tener acceso y qué acciones se pueden realizar con dichos puertos.*

RuntimePermission: *Permiso base para las operaciones de tiempo de ejecución. Permite restringir operaciones como salir de la máquina virtual y cargar bibliotecas externas*

AllPermission: *Clase que indica que se permiten todos los permisos sobre todas las acciones. Se recomienda sólo para pruebas.*



Permisos comunes asociados a recursos del sistema



Ejemplo:

```
grant{  
    permission java.security.SocketPermission "127.0.0.1:1024-1100", "accept, listen, connect";  
}
```



JAAS



Este paquete proporciona la capacidad de determinar quién puede hacer qué.

Proporciona a las aplicaciones Java servicios de autenticación y autorización.

La autenticación garantiza que un individuo es quien dice ser.

La autorización verifica que el usuario autenticado está autorizado a hacer lo que está intentando hacer.

