autentia

Archiva
Gestión de Repositorios Maven

Ángel García Jerez

Índice

- Conceptos fundamentales de Maven
  - ¿Qué es?
  - ¿Cómo funciona?
- ¿El porqué de la gestión de repositorios Maven?
- Introducción a Archiva
  - Instalación
  - Configuración
  - Administración
  - Particularización de la autenticación
- Nuestra experiencia
- Ronda de Preguntas

Conceptos fundamentales de Maven

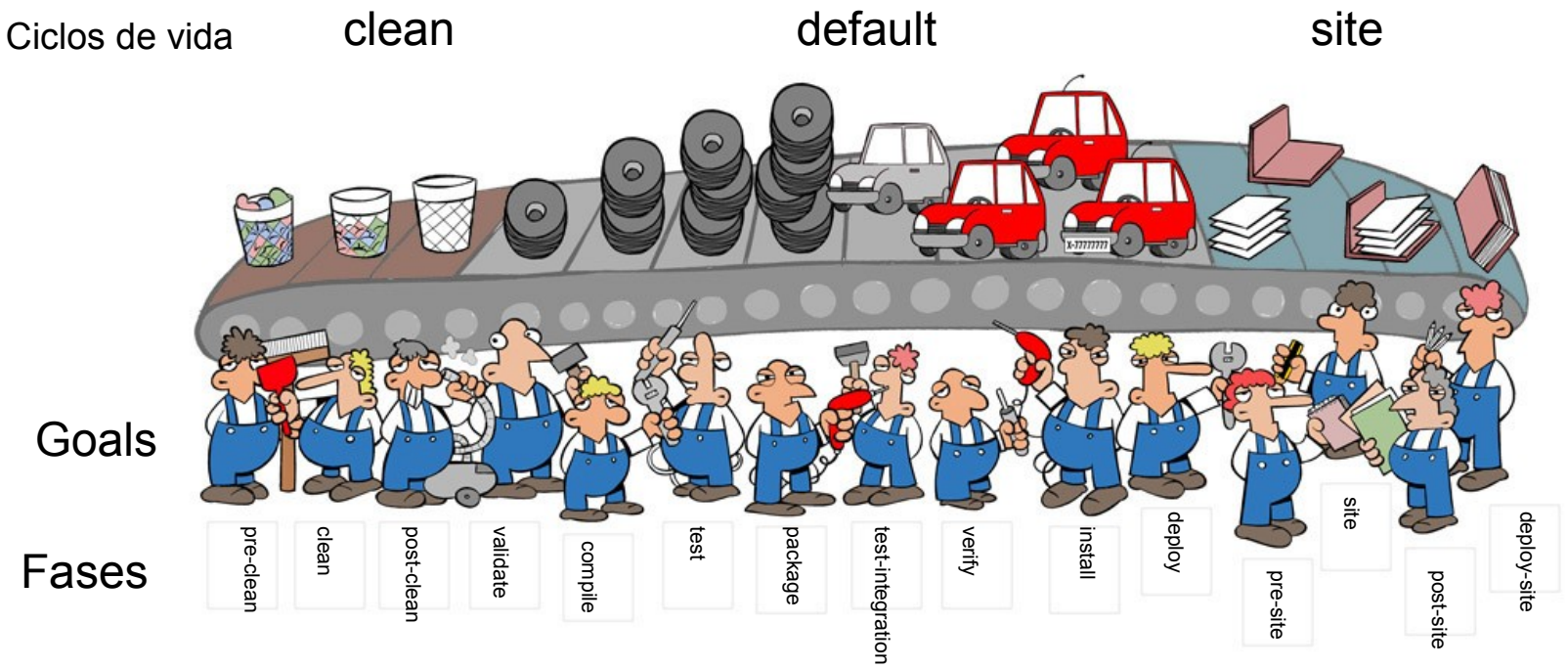
- ¿Qué es Maven?

  Maven es una herramienta que nos ayuda a automatizar el proceso de construcción y gestión de los proyectos.
- Objetivos:
  - Reducir la complejidad en el proceso de construcción.
  - Tener un sistema uniforme de construcción.
  - Ayudar a realizar mejores practicas de desarrollo incorporando pruebas unitarias.
  - Mayor información de calidad en los proyectos
    - Changelog
    - Lista de dependencias
    - Informes unitarios
    - Lista de mail
    - Desarrolladores

Conceptos fundamentales de Maven

- ¿Cómo funciona Maven?

  Maven haciendo un símil sería una cadena de montaje de una fabrica de coches.



Conceptos fundamentales de Maven

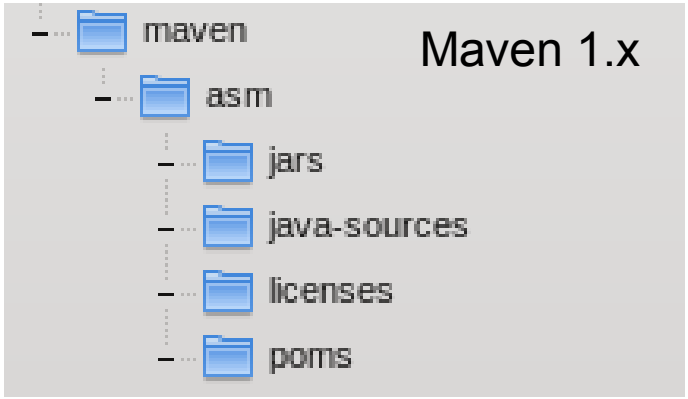
- POM (Project Object Model)

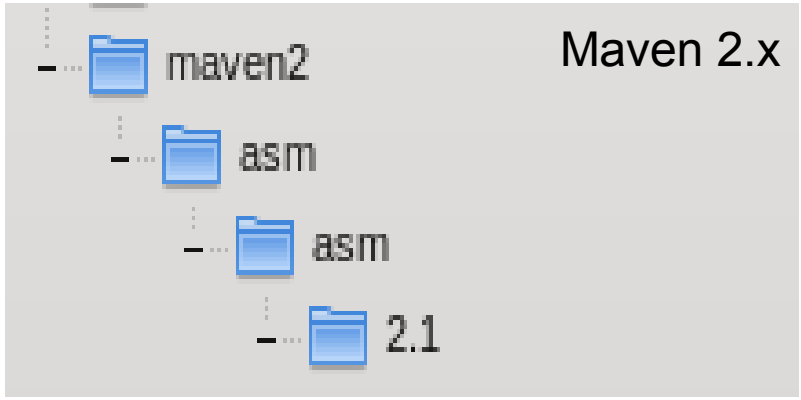
  Es un fichero XML obligatorio en todo proyecto Maven, donde se incluye la información (meta-datos) necesaria para que éste pueda construir y gestionar nuestro proyecto.

  - groupId
  - artifactId
  - Version
  - packaging (optional): jar, war, pom, ear.
  - Dependencies
  - .....

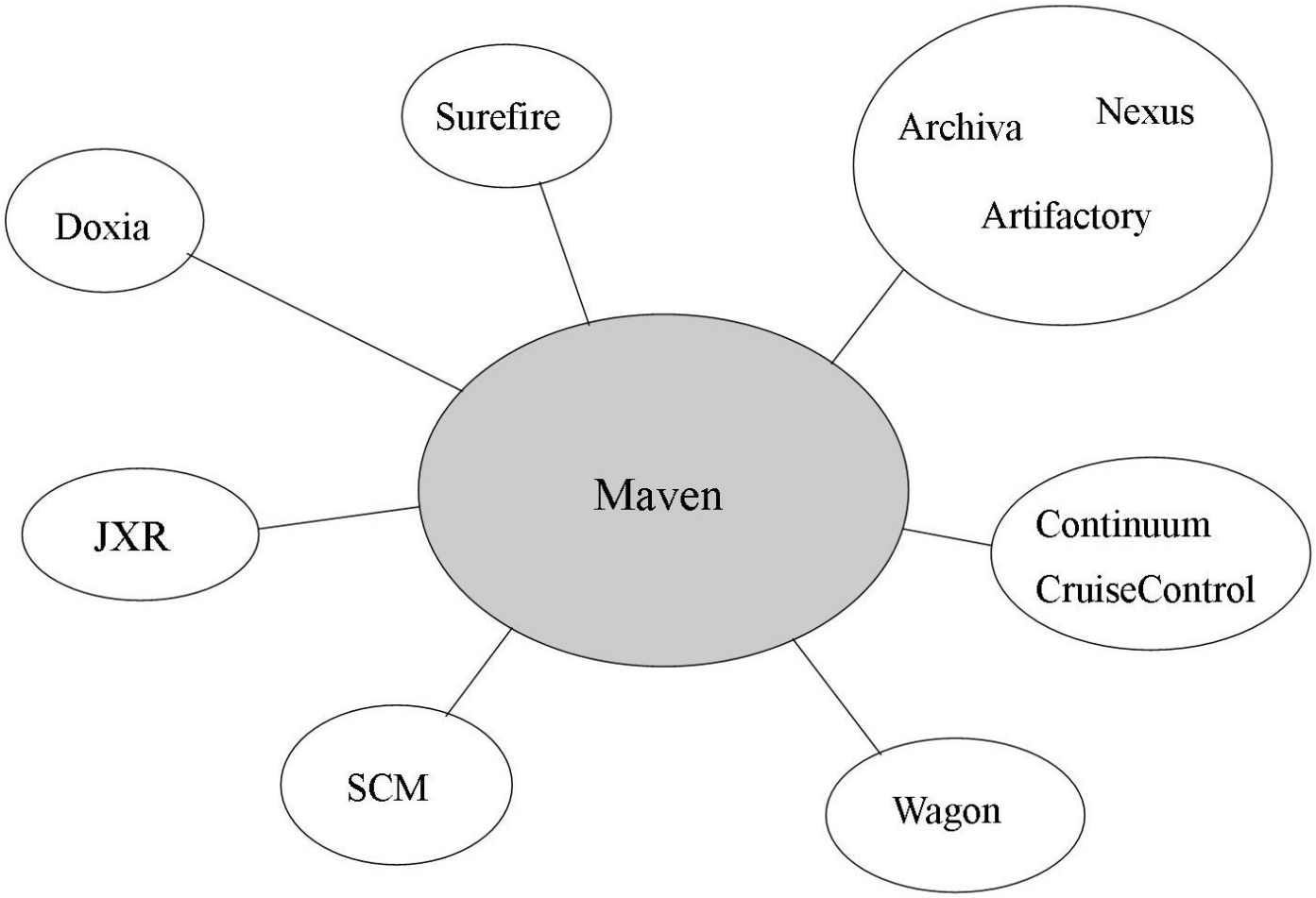
Conceptos fundamentales de Maven

- ¿Qué hace Maven con los artefactos generados?
  Finalizado el proceso de construcción del artefacto, Maven lo deposita en repositorios.
- Dos tipos:
  - **Repositorio local**: situado en la máquina del desarrollador. Almacena artefactos instalados (maven install) y descargados de repositorios remotos.
  - **Repositorio remotos**: repositorios accesibles a través de protocolos file:// y http://.
    - **internos**: utilizados por las empresas para almacenar sus artefactos que son compartidos por los desarrolladores.
    - **externos**: repositorios públicos utilizados para almacenar artefactos de terceros.
- Dos estructuras dependiendo de la versión del repositorio:





Conceptos fundamentales de Maven



¿El porqué de la Gestión de repositorios Maven?

- A simple vista Maven puede parecer un sistema perfecto.
- Problemas:
  - Excesivo consumo de ancho de banda en equipos de desarrollo corporativos
  - Aumento del tiempo de construcción dependiendo de la política de actualización de los repositorios.
  - No existe control sobre los artefactos descargados por los desarrolladores (compatibilidad de licencias, estandarización a una versión de un producto, etc..)
- Gestores de repositorios Maven:
  - Proximity/Nexus
  - Artifactory
  - Apache Archiva
  - DSMP (Dead Simple Proxy)

Introducción a Archiva

- Archiva es una aplicación web que permite gestionar repositorios Maven y mucho más.
- Resuelve los problemas que pueden aparecer en organizaciones:
  - consumo de ancho de banda
  - resude el tiempo de construcción
  - mayor control de los artefactos descargados.
- Características:
  - Autenticación/Autorización (roles) en Archiva y repositorios gestionados.
  - Gestión de usuarios.
  - Proxy de repositorios remotos (cacheo de los artefactos descargados).
  - Gestión de repositorios internos.
  - Búsqueda de artefacto en los repositorios definidos.
  - Navegación sobre los artefactos.
  - Identificación de los artefactos anónimos.
  - Repositorios Virtual.
  - RSS

Instalación de Archiva

- Jakarta distribuye dos alternativas de instalación:
  - Instalando Archiva como una aplicación web desplegada sobre un servidor de aplicaciones.
  - Instalación con una aplicación standalone.
- Instalando Archiva como aplicación standalone:
  - Jetty: como Servidor Web.
  - Apache Derby: base de datos relacional.
    - Dos esquemas:
      - usuarios (información sobre los usuarios)
      - archiva (información sobre los repositorios)
  - Requerimientos mínimos:
    - JDK 1.4 o superior.
    - Espacio mínimos alrededor de 20 MB.
    - SSOO: Windows, Linux, Solaris y Mac OS X.

Configuración

- Archiva utiliza Redback para incorporar las funcionalidades de autenticación, autorización (roles) y gestión de usuarios.
- Redback es framework que incorpora a las aplicaciones web funcionalidades de seguridad habituales en cualquier aplicación de manera simple y sencilla .
- Cualquier aplicación que incluya este framework es configurada de la misma forma (Continuum, herramienta de integración continua).
- La configuración de Redback se centraliza en un fichero de propiedades (security.properties) situado en el directorio de configuración de archiva o en el directorio .m2 del directorio home del usuario. Se puede dividir en varios grupos:
  - **JDBC Setup**: propiedades para acceder al gestor de base de datos.
  - **Email Settings**: propiedades para el correcto envío de email.
  - **Auto Login Settings**: propiedades para el recordatorio de acceso.
  - **Default Usernames Values**: propiedades sobre los usuarios administradores y invitados.
  - **Security Policies**: propiedades para definir la política de seguridad de las contraseñas.
  - **Password Rules**: propiedades para definir el formato de las contraseñas.
  - **LDAP Settings**: propiedades para la autenticación contra LDAP.

JDBC Setup

- Archiva utiliza por defecto Derby como sistema gestor de base de datos. Archiva trabaja con dos modelos: uno para los datos de los usuarios y otro para los datos de los repositorios. Entre las propiedades que podemos personalizar se encuentra:

  - **jdbc.driver.name** (org.apache.derby.jdbc.EmbeddedDriver): driver de acceso a la base de datos de usuario y repositorios.
  - **jdbc.url** (jdbc:derby:${plexus.home}/database;create=true): dirección de acceso a la base de datos.
  - **jdbc.username** (sa): nombre del usuario que se utiliza para conectarse a la base de datos
  - **jdbc.password**: contraseña que se utiliza para conectarse a la base de datos.

Email Settings

- Archiva utiliza un servidor de correo como parte del proceso de validación y reactivación de las cuentas creadas. Su configuración es fundamental para el correcto funcionamiento de la aplicación, aunque puede desactivarse añadiendo la propiedad email.validation.required=false en el fichero security.properties.

  - **email.smtp.host** (localhost): maquina donde se encuentra situado el servidor de correo.
  - **email.smtp.port** (25): puerto donde se encuentra escuchando nuestro servidor de correo.
  - **email.smtp.ssl.enabled** (false): propiedad para activar el acceso a servidores de correos utilizando el protocolo SSL.
  - **email.smtp.tls.enabled** (false): propiedad para activar el acceso a servidores de correos utilizando el protocolo TLS.
  - **email.smtp.username**: nombre de usuario utilizado para enviar los correos.
  - **email.smtp.password**: contraseña del usuario utilizado para enviar los correos.
  - **email.from.address** (${user.name}@localhost): dirección de correo que se establecerá en el campo "from" de los correos.
  - **email.from.name**: nombre del usuario que se adjuntará a la dirección de correo del campo "from".
  - **email.validation.required** (true): propiedad que permite activar/desactivar la confirmación del alta de los usuarios.
  - **email.validation.timeout** (2880): periodo de tiempo en minutos que el usuario tiene para poder confirmación el correo de activación.
  - **email.validation.subject** (Welcome): valor utilizado en el campo "Asunto" en los correo de activación de los usuarios.
  - **email.feedback.path** (/feedback.action): valor utilizado como enlace de acceso a la aplicación desde los correos. Si empieza por "/" se añade al final del valor de la variable application.url o puede incluirse valores como mailto:xxx@yyy.com.

Auto Login Settings/ Default Usernames Values

- Archiva por defecto tiene activada la funcionalidad de auto login, es decir, un usuario una vez logado no necesitaría logarse de nuevo en la aplicación hasta que se expire el periodo de auto login siempre y cuando el usuario seleccione el check de "Remember me" del formulario de login. Esta característica es crítica y más si se dejase con los valores por defecto que tiene configurado Archiva cuyo periodo de expiración del auto login es de un año.
  - **security.rememberme.enabled** (true): propiedad que activa/desactiva la capacidad de auto login en la aplicación.
  - **security.rememberme.timeout** (525600): periodo de expiración del auto login en minutos.
  - **security.signon.timeout** (30): tiempo de expiración de la sesión.
- Archiva también nos permite establecer los usuarios que por defecto tendrán el rol de administrador e invitado en la aplicación.
  - **redback.default.admin** (admin): lista de usuario que tendrán el rol de administrador.
  - **redback.default.guest** (guest): lista de usuario que tendrán el rol de invitado.

Security Policies

- Existen ciertas propiedades que nos permiten cambiar comportamientos en la política de seguridad de las contraseñas. Archiva por defecto viene configurado con una política que puede ser demasiado restrictiva en entornos de desarrollos y puede ser susceptible de modificarse.
  - **security.policy.password.previous.count** (6): número de contraseña previas que no podemos repetir.
  - **security.policy.password.expiration.enabled** (true): permite activar/desactivar la expiración de la contraseña.
  - **security.policy.password.expiration.days** (90): número de días que la contraseña será validad desde el último cambio.
  - **security.policy.password.expiration.notify.days** (10): número de días que Archiva notifica al usuario de estar próximo la expiración de contraseña.
  - **security.policy.allowed.login.attempt** (10): número de acceso erróneos a Archiva antes de bloquear la cuenta.
  - **security.policy.strict.force.password.change.enabled** (true): permite activar/desactivar el proceso de cambio de contraseña cuanto esta ha caducado.

Password Rules

- También se nos permite modificar el formato que deseamos admitir en las contraseñas.
  - **security.policy.password.rule.alphanumeric.enabled** (false): permite activar/desactivar la obligación de que la contraseña contenga caracteres alfanúmericos.
  - **security.policy.password.rule.alphacount.enabled** (true): permite activar/desactivar la obligación de que la contraseña contenga un número de caracteres alfanuméricos como mínimo.
  - **security.policy.password.rule.alphacount.minimum** (1): número mínimo de caracteres alfanumérico permitidos.
  - **security.policy.password.rule.characterlength.enabled** (true): permite activar/desactivar la obligación de que la contraseña contenga un número mínimo/máximo de caracteres.
  - **security.policy.password.rule.characterlength.minimum** (1): número mínimo de caracteres.
  - **security.policy.password.rule.characterlength.maximum** (24): número máximo de caracteres.
  - **security.policy.password.rule.musthave.enabled** (true): permite activar/desactivar la obligatoriedad de rellenar la contraseña.
  - **security.policy.password.rule.numericalcount.enabled** (true): permite activar/desactivar la obligación de que la contraseña contenga números.
  - **security.policy.password.rule.numericalcount.minimum** (1): número mínimo de números.
  - **security.policy.password.rule.reuse.enabled** (true): permite activar/desactivar la comprobación que hace Archiva para saber si la contraseña nueva coincide con algunas de las anteriores (relacionada con la propieadad security.policy.password.previous.count).
  - **security.policy.password.rule.nowhitespace.enabled** (true): permite activar/desactivar que la contraseña contenga o no espacios.

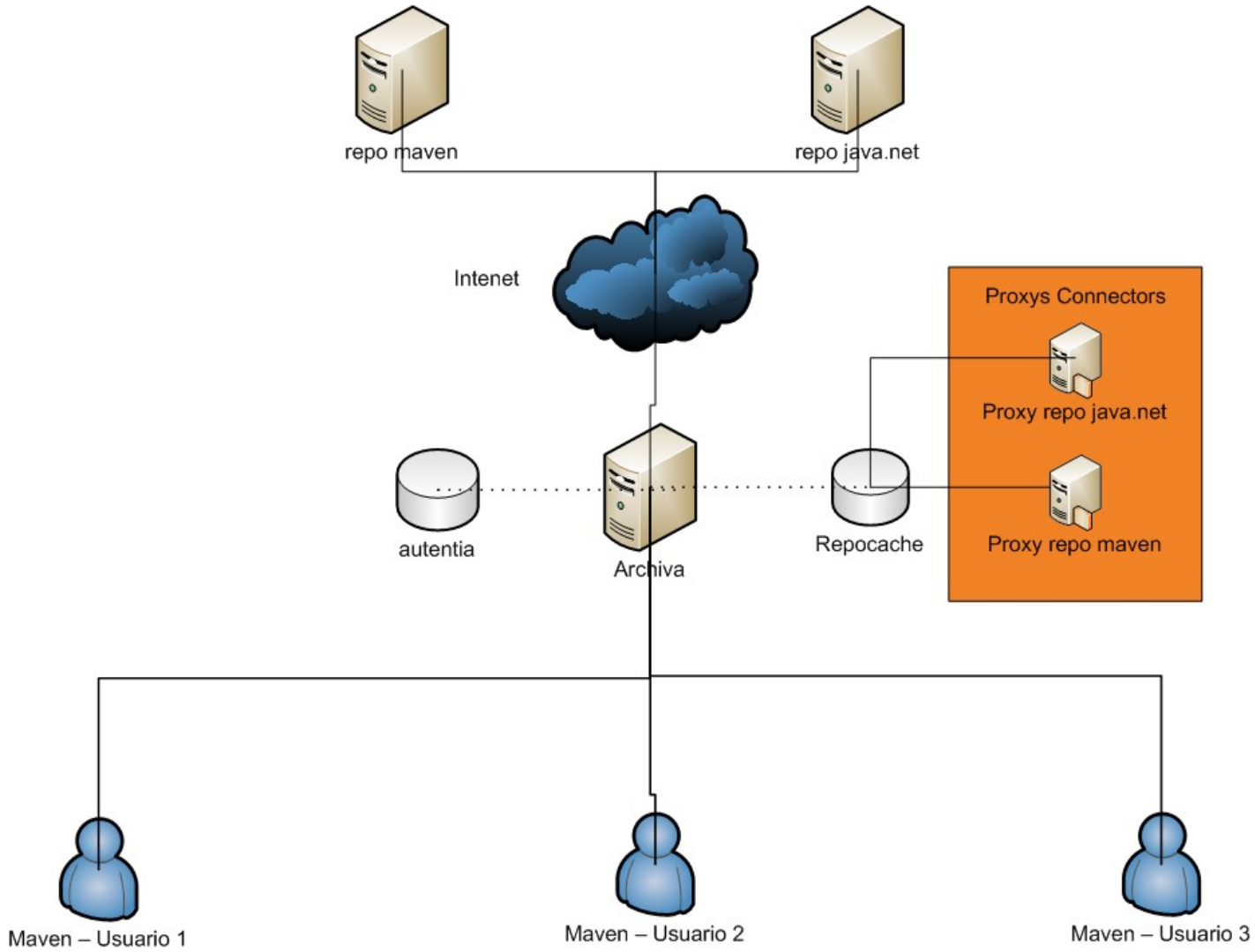
LDAP Settings

- Archiva también nos permite activar la autenticación contra LDAP.
  - **ldap.bind.authenticator.enabled** (false): propiedad que activa el proceso de autenticación "bind authenticator".
  - **ldap.config.hostname**: nombre de la máquina donde se encuentra situado el LDAP.
  - **ldap.config.port**: puerto donde se encuentra escuchando el LDAP.
  - **ldap.config.base.dn**: DN raíz del LDAP.
  - **ldap.config.context.factory**: factoría utilizada para conectarse al LDAP.
  - **ldap.config.bind.dn**: usuario con el que se conectará Archiva al LDAP, debe tener al menos permisos de lectura sobre la rama de usuarios.
  - **ldap.config.password**: contraseña del usuario para conectarse al LDAP.
  - **user.manager.impl** (cached): implementación utilizada para la gestión de usuarios. Puede tener dos posibles valores "**cached**", gestión de usuarios local, o "**ldap**", gestión de usuarios situados en un LDAP.
  - **ldap.config.mapper.attribute.email**: nombre del campo LDAP con el que se mapeará el campo email del usuario.
  - **ldap.config.mapper.attribute.fullname**: nombre del campo LDAP con el que se mapeará el campo nombre completo del usuario.
  - **ldap.config.mapper.attribute.password**: nombre del campo LDAP con el que se mapeará el campo contraseña del usuario.
  - **ldap.config.mapper.attribute.user.id**: nombre del campo LDAP con el que se mapeará el campo login del usuario.
  - **ldap.config.mapper.attribute.user.base.dn**: nombre de la rama donde encontrará la lista de usuarios. Recomendamos encarecidamente que solo se configure esta propiedad en el fichero application.xml y no se defina en el fichero security.properties ya que Archiva contiene un bug que no se ha solucionado.
  - **ldap.config.mapper.attribute.user.object.class**: nombre del objetoClass que tienen los usuarios en el LDAP.
  - **ldap.config.mapper.attribute.user.filter**: filtro que permite restringir la lista de usuarios que puede acceder a Archiva.

Administración

- Con Archiva podremos administrar/gestionar:
  - Usuarios:
    - Añadir, modificar y eliminar usuarios (en sistema de autenticación sin LDAP).
    - Asignar roles.
      - Roles de administración de Archiva:
        - Administrador de usuarios
        - Administrador de la aplicación
      - Roles de acceso a repositorios
        - Global Repository Observer
        - Global Repository Manager
        - Repository Observer
        - Repository Manager
  - Informes
  - Apariencia
  - Subida de artefactos
  - Grupo de Repositorios (Repositorio Virtual)
  - Repositorios internos y remotos
  - Proxy Connectors
  - Legacy Support
  - Network Proxies
  - Repository Scanning
  - Database

Ejemplo practico



Particularización de la autenticación

- Archiva (Redback) permite que la autenticación se realice contra un LDAP.
- Al activar la autenticación contra el LDAP perdemos la mayoría de la funcionalidades de la gestión de usuarios (crear, modificar y eliminar) ya que sólo disponemos de acceso de solo lectura al LDAP.
- La asignación de roles es local a la aplicación.
- Existen dos formas de autenticación contra el LDAP:
  - bind authenticator
  - comprobación manual de la contraseña.
- Los pasos necesarios para configuración son:
  - Añadir varios elementos al fichero application.xml (Plexus).
    - se añade la configuración para la factoría de la conexión LDAP
    - se añade la configuración del mapeo de los campos LDAP con atributos del usuario.
    - se añade la configuración del cacheo de los usuarios
    - si se utiliza la autenticación manual se añade la política de seguridad entre las que está el algoritmo de cifrado de la contraseña.
  - Configuración del fichero security.properties.
  - Modificación de la librería redback-common-ldap (bug).

Nuestra experiencia

- Archiva es una aplicación bastante completa con muchas funcionalidades.
- El LDAP es joven, tiene todavía bugs y su acceso es de solo lectura.
- Es recomendable modificar/eliminar ciertas funcionalidades establecidas por defecto.
- Tener siempre sincronizada la contraseña con nuestro fichero settings.xml.

Ronda de preguntas