

# Gestión de proyectos con Software Libre

Sergio Gómez Bachiller  
Coordinador del Aula de Software Libre  
<sergio@uco.es>

10 de marzo de 2010

*Aula de  
Software Libre*



UNIVERSIDAD DE CÓRDOBA

# Contenido

1. Sistemas de control de versiones
2. Sistemas de gestión de proyectos y seguimiento de errores
3. Introducción a *Netbeans*
4. La Forja de *Rediris*
5. El tablón de proyectos del ASL

# ¿Qué es un sistema de control de versiones?

Sistema de gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo.

Aula de  
**Software Libre**



UNIVERSIDAD DE CÓRDOBA

# Sistemas de control de versiones

## Características

- ✓ Almacenaje
- ✓ Cambios
- ✓ Registro

# Sistemas de control de versiones

## Clasificación

- ✓ **Centralizados**
- ✓ **Distribuidos**

# Sistemas de control de versiones

## Funcionamiento

- ✓ **Exclusivos**
- ✓ **Colaborativos**

Aula de  
*Software Libre*



UNIVERSIDAD DE CORDOBA

# Sistemas de control de versiones

## ¿Por qué hay que usarlos?

- ✓ **Prácticas**
- ✓ **Proyectos**

Aula de  
**Software Libre**



UNIVERSIDAD DE CÓRDOBA

# Sistemas de control de versiones

## **Subversion**

**Software de sistema de control de versiones centralizado.**

*Aula de*  
**Software Libre**



UNIVERSIDAD DE CORDOBA



# Subversion

Esquema	Método de acceso
file:///	acceso directo al repositorio (en disco local)
http://	acceso vía protocolo WebDAV a un servidor Apache que entiende de Subversion
https://	igual que http://, pero con cifrado SSL.
svn://	acceso vía un protocolo personalizado a un servidor svnserver.
svn+ssh://	igual que svn://, pero a través de un túnel SSH.

Aula de  
**Software Libre**



UNIVERSIDAD DE CÓRDOBA

# Subversion

## Primeros pasos

### svnadmin

## Herramienta de administración de repositorios

```
sergio@servidor:~$ mkdir svn
```

```
sergio@servidor:~$ cd svn
```

```
sergio@servidor:~/svn$ svnadmin create holamundo
```

*Aula de*  
**Software Libre**



UNIVERSIDAD DE CÓRDOBA

# Sistemas de control de versiones

## Terminología básica

- ✓ **Repositorio**
- ✓ **Rotular ('tag')**
- ✓ **Rama ('branch')**

# Subversion

## Configurando el repositorio

**svn**

**Herramienta de cliente de subversion**

```
sergio@servidor:~$ svn mkdir -m "Estado inicial"  
file:///home/sergio/svn/holamundo/trunk  
file:///home/sergio/svn/holamundo/tags  
file:///home/sergio/svn/holamundo/branches
```

*Aula de*  
**Software Libre**



UNIVERSIDAD DE CÓRDOBA

# Subversion

## Desplegando... (checkout)

**Un despliegue crea una copia de trabajo local desde el repositorio**

```
sergio@servidor:~$ mkdir proyectos
sergio@servidor:~$ cd proyectos
sergio@servidor:~/proyectos$ svn co
file:///home/sergio/svn/holamundo/trunk holamundo
sergio@servidor:~/proyectos$ cd holamundo
```

*Aula de*  
**Software Libre**



# Subversion

## Publicando archivos

**Antes de publicar un archivo hay que añadirlo al repositorio**

```
sergio@servidor$ vim holamundo.c
```

```
sergio@servidor$ svn add holamundo.c
```

```
sergio@servidor$ svn add -m "Añadiendo holamundo.c"
```

*Aula de*  
**Software Libre**



# Subversion

## Accediendo desde el exterior

Con el URI **file:///** accedemos al repositorio local,  
con **svn+ssh:///** podemos acceder a un  
repositorio remoto.

```
sergio@equipo2:~/proyectos$ svn co  
svn+ssh://sergio@ts.uco.es/home/sergio/svn/holamundo/trunk holamundo
```

Aula de  
*Software Libre*



UNIVERSIDAD DE CÓRDOBA

# Subversion

- Actualizar
  - svn update
- Modificar
  - svn add
  - svn delete
  - svn copy
  - svn move
- Examinar
  - svn status
  - svn diff
  - svn revert
- Fusionar
  - svn merge
  - svn resolved
- Enviar
  - svn commit