

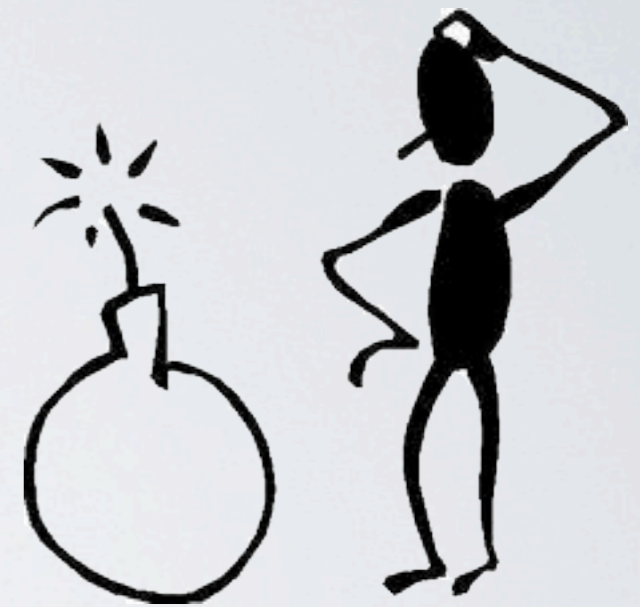
# DEPLOYMENTS Y EL CICLO DE DESARROLLO EN DRUPAL

Nestor Mata Cuthbert  
Senior Web Developer  
Up And Running Software



# EL PROBLEMA

- Mas de 1 programador
- Mas de 1 servidor
- Mas de 1 ambiente
- Mas de una version





# EL PROBLEMA

- Integración
- Manejo de versiones
- Seguridad del código (ante perdidas, mantener historicos)
- Pruebas
- Puesta en producción
- Drupal tiene mucha configuracion y estructura en DB



# INTEGRACION DE CÓDIGO MANEJO DE VERSIONES

- Perdida de horas de trabajo
- Versiones sobrescritas por otra persona o uno mismo
- Accidentes
- Conflictos en lo implementado por diferentes personas del equipo
- El enfoque que se utilizó no funcionó





# CONTROL DE VERSIONES

- **No hay excusa**, se debe hacer como regla
- Mantiene historicos de las modificaciones
- Hace sencillo volver a una versión anterior
- Comparar cambios que pudieron producir un problema
- Ingrar código de diferentes programadores
- Tener diferentes versiones en diferentes ambientes
- Crear esquema de aprobación por versiones
- Herramienta para proceso deployments



# CONTROL DE VERSIONES

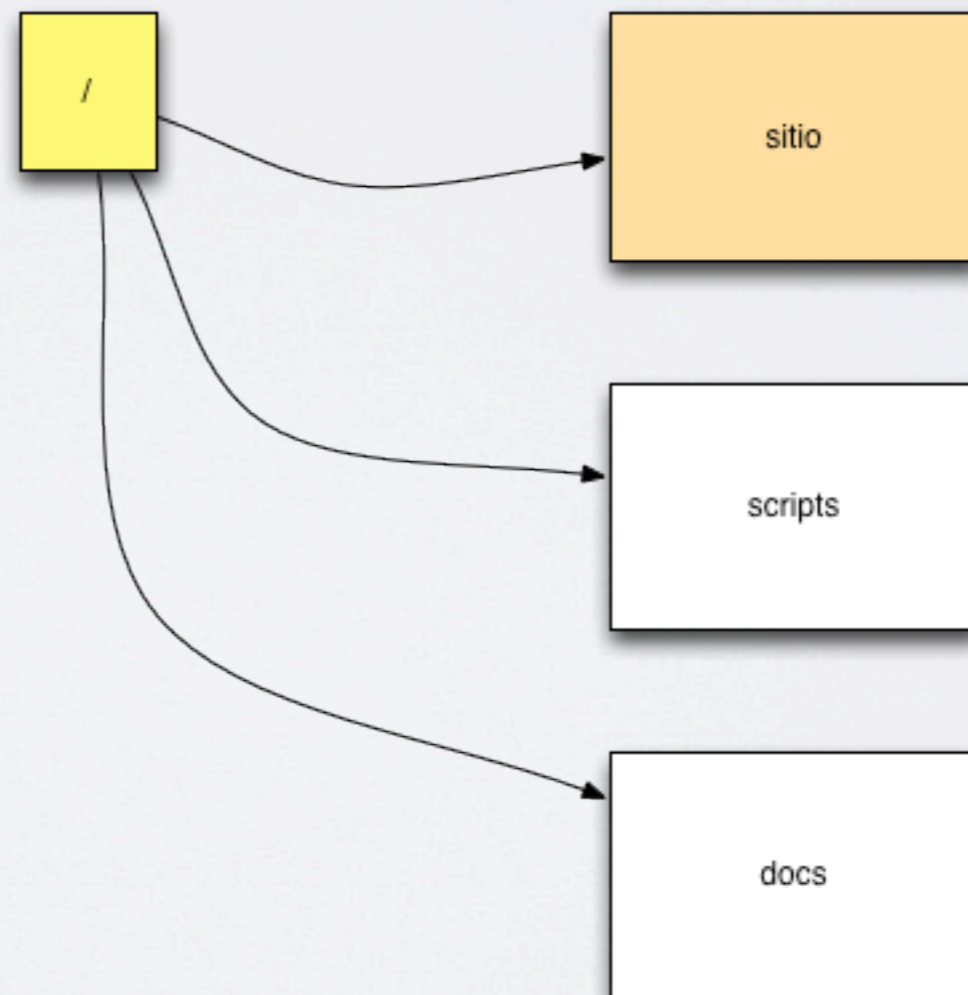
- GIT
- SVN
- Otros
- Herramientas gratuitas
  - Unfuddle (<http://www.unfuddle.com>)





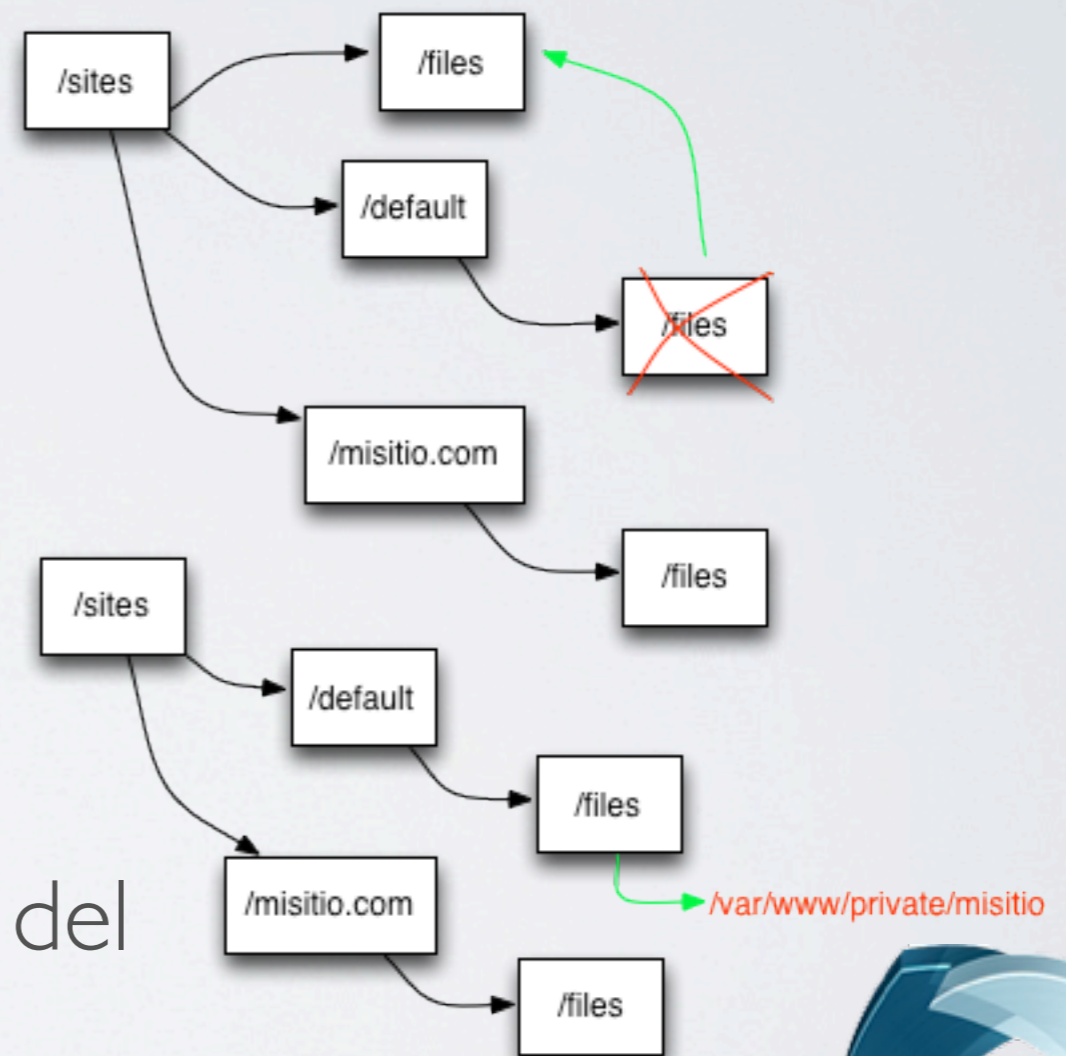
# ESTRUCTURA CONTROL DE VERSIONES

- No utilizar la raíz del código del sitio en la raíz del control de versiones
- Posible estructura:



# DIRECTORIO FILES Y CACHE

- No utilice el directorio /files en el directorio de configuracion del sitio
- Y cree reglas de exclusion en control de versiones
- O use links simbolicos hacia fuera del directorio publico





# PASES A LOS SERVIDORES



# REGLAS

- Es **necesario** definir reglas en el equipo de trabajo
- No modificaciones directas al servidor
- Dias y horas para pases a producción
- Protocolos para pases de alta prioridad
- Protocolo de aprovacion y pase a producción
- Formato de comentario de commits





# PROTOCOLO PASES A PRODUCCIÓN



- En todo momento saber, en cada servidor: version, timestamp, responsable, que contiene y poder comparar versiones
- Cada pase a producción, aprobado por responsable
- Efectuar respaldo antes del pase
- Protocolo de rollback



# UNIDADES DE PRUEBA

- Pruebas automaticas aplicables a una unidad de código para corroborar que cumple con lo esperado
- Es recomendable, aunque implica un cambio cultural en el equipo de desarrollo
- TDD (Test Driven Development), aplaudido y criticado, es una opción
- Reduce el riesgo de incrustar errores en algo que ya es funcional

**TDD**

**ALL CODE IS GUILTY  
UNTIL PROVEN INNOCENT**





# DOCUMENTACIÓN

- Documentar al menos las cosas mas importantes
- Pasar las vacaciones sin ser llamado por emergencia
- Evitar llamar al que esta de vacaciones
- Entrar en crisis porque el unico que conocia se accidente o renuncio



# DOCUMENTACION

- La información de los servidores
- Las claves (de forma segura)
- Los directorios importantes
- Los procedimientos de respaldo y restauración
- Las configuraciones
- Procedimientos importantes
- Etc





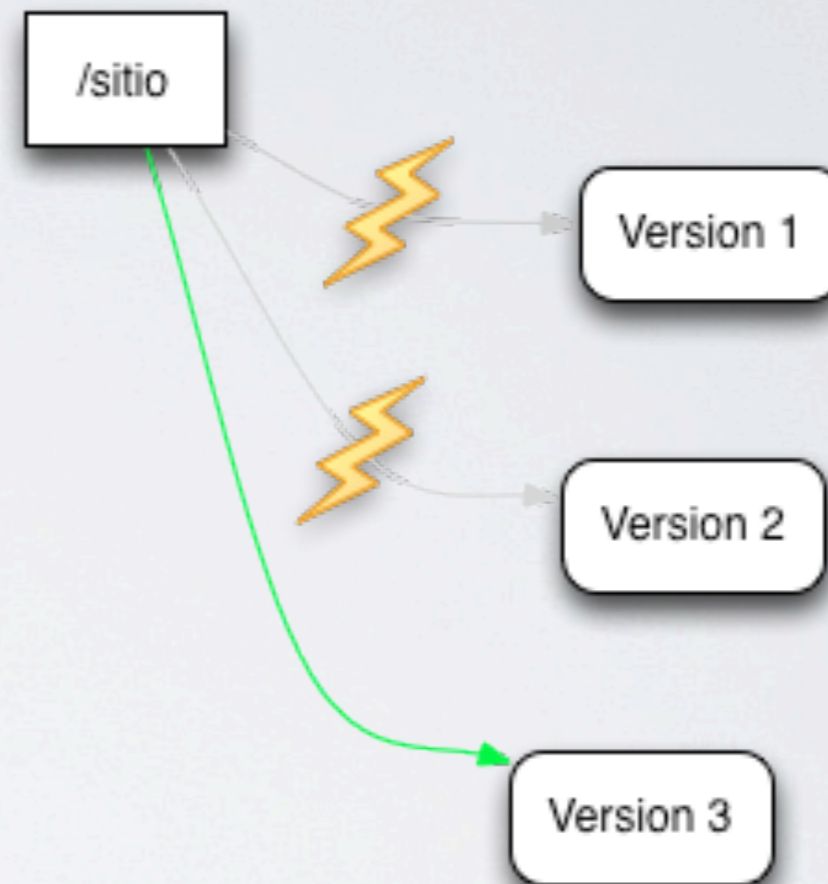
# DOCUMENTACION

- Se puede guardar parte de esta en el control de versiones



# ENFOQUES DE PASES DE CODIGO Y ARCHIVOS

- Pull from server
- Push to server
- En combinacion con push o pull
  - Enfoque con links simbolicos
    - Un directorio nuevo para cada pase





# MANEJO DE CONFIGURACIONES EN DB

- Si solo fuese código, sería relativamente sencillo
- Una vez que se ha puesto el sitio en producción se complica el pase de configuraciones que existen en la base de datos
- No hay una solución sencilla y definitiva



# SOLUCIONES A CONFIGURACION EN DB

- Update Hooks
- SQL Scripts
- Patterns
- Exportables
- Documentación de pasos manuales
- Database Diff
- Features





# FEATURES

- No se desarrollo como método para efectuar pases a producción
- Forma de empaquetar funcionalidad instalable
- Por su concepto, puede ser utilizado para hacer deploy de funcionalidad
- Convierte configuraciones a modulos en código
- Código se puede manejar control de versiones



# FEATURES

- Permite inspeccionar los cambios de configuración y estructura
- Genera código organizado en módulos
- Esquema bastante fiable para pases





# HERRAMIENTAS PARA DEPLOY

- Aegir: Un administrador de sitios hecho en Drupal
- Capistrano: Herramienta hecha en Ruby que a travez de recetas permite hacer un deployment
- Custom Scripts

# PREGUNTAS







# **UP AND RUNNING**

## **SOFTWARE DEVELOPMENT**

**GREAT PEOPLE + GREAT PROCESS = GREAT RESULTS**

## **Unase a la división Drupal**

**Nestor Mata Cuthbert**  
**Senior Web Developer**

nestor@profesional.co.cr

Twitter: @nestormata

Skype: nestor.mata

