

# Futuro profesional, empresa y Software Libre

**Sancho Lerena**

[sancho.lerena@artica.es](mailto:sancho.lerena@artica.es)

Marzo 2007

# Contenido

- ▶ Presentación.
- ▶ La empresa y el Software Libre (introducción)
- ▶ Mercado laboral en T.I.
- ▶ Proyectos de Software Libre para la empresa.
- ▶ Gestión de proyectos de Software Libre desde la empresa.
- ▶ Futuro del Software Libre en la empresa.





**Artica Soluciones Tecnológicas S.L** (12/05)

<http://www.artica.es>



**Pandora FMS** (2004)

<http://pandora.sourceforge.net>



**Babel Enterprise** (2005)

<http://babel.sourceforge.net>

# Bloque 1.

## La empresa y el Software Libre (introducción)



## 1. Introducción SL y empresa

No se trata de que las empresas usen Software Libre, sino de que las empresas **puedan ganar dinero** con el SL.

Para las empresas proveedoras de servicios de Tecnología de la Información, **el Software Libre es un medio, no un fin.**

Uno de los objetivos de cualquier empresa es **producir beneficios.**

**Esto no es contradictorio con el espíritu GNU.** Desde siempre se ha dicho que *libre no significa gratis* y por lo tanto no significa “NO comercial”. Es una diferencia marcada de la licencia GPL respecto a otras licencias OpenSource.

Tampoco es contradictorio obtener beneficios con garantizar los derechos de los usuarios.

El Software Libre es como una **receta de cocina:**

No importa la receta únicamente, **importa saber cocinar.**

El negocio está en saber cocinar, **no en poseer el "secreto"** de la receta.

## 1. Introducción SL y empresa

El software es una herramienta.

No valoramos el negocio de la venta de herramientas sino el de la **venta de conocimiento**.

Se deben vender el conocimiento de quien mejor sabe usar esas herramientas.

El negocio está en los **servicios** y en la personalización de las herramientas.



## **Modelos de negocio :**

- **Oferta de servicios (Hecker & Young).**  
Ejemplo: Modelos asociados a GNU/Linux.
- **Desarrollo de un producto propio**  
Ejemplo: MySQL.

## **Oferta de servicios (Hecker & Young)**

- Consultoría.
- Integración y adaptación.
- Soporte.
- Formación.

## Oferta de servicios

- GNU/Linux es el mejor ejemplo de modelo de negocio basado en el modelo de servicios de Hecker & Young.
- Existen diferentes tipos de empresas, con diferentes perfiles, todos ellos centrados en torno a GNU/Linux:

- Consultoría: Implantación y desarrollo de proyectos en grandes y pequeños entornos: Ej: IBM.
- Integración y adaptación: Desarrollan pequeñas piezas y las integran en proyectos: Ej: RedHat.
- Soporte: Ofrecen soporte a usuario final, o otras empresas. Ej: HP, IBM, Oracle
- Formación: Ofrecen formación a usuarios y empresas.



**redhat**

## Desarrollo de un producto propio

- Ventaja estratégica.
- Mejor conocimiento; más garantía.
- Venta de la marca. El producto como marketing.
- Oferta de servicios (Hecker & Young).



## Cooperación competitiva

**Lo que me hace más fuerte a mí, también te lo hace a tí.** Esto produce una continua mejora para el proyecto, para la comunidad, y elimina automáticamente aquellos que no innovan.

### Ejemplos:

**Apache:** Colaboran en él varias empresas, grandes y pequeñas. Todas ellas no tienen más remedio que competir mejorándolo.

**Distribuciones de GNU/Linux:** Compiten en un mismo mercado y todas se benefician de las mejoras de sus competidoras. Las que no innovan desaparecen.



## ¿ El Software Libre no es negocio ?

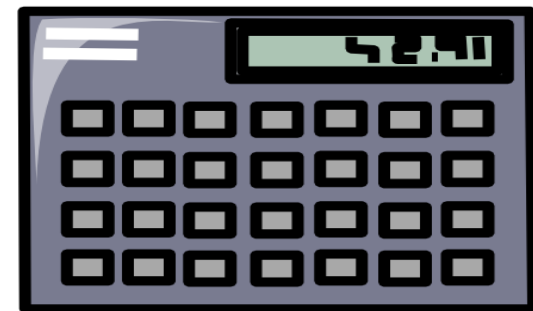
- SUN Microsystems (OpenSolaris, JAVA).
- IBM (Apache, Redhat y otros ejemplos).
- Redhat.
- Novell (SuSE)
- HP (Soporte usuarios Debian).

La mayor parte del kernel Linux está mantenido por desarrolladores en nómina de una de estas empresas.

## El Software Libre NO es gratis

### TCO. Total Cost of Ownership:

- Coste de exploración de soluciones.
- Coste de evaluación de soluciones.
- Costes de integración.
- Costes de formación.
- Costes de soporte.



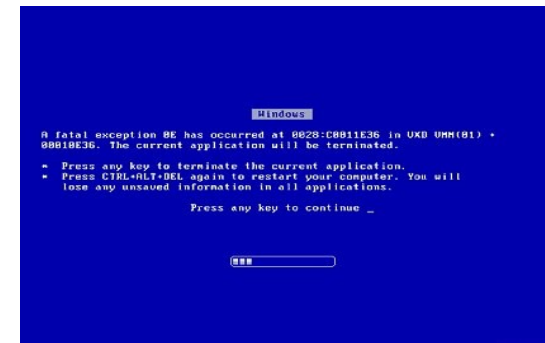
### ¿Qué incluye el TCO?

- Coste de hardware para dicha plataforma.  
Windows Vista requiere mas hardware que GNU/Linux.
- Coste de software para dicha plataforma.  
Incluye S.O, herramientas de gestión y de las aplicaciones.
- Costes de operación de TI.  
Saber operar con un sistema determinado cuesta dinero.



## ¿Qué incluye el TCO?

- Costes de administración de TI.  
Saber administrar un sistema cuesta dinero.
- Coste de las paradas de servicio.  
Las pantallas azules cuestan dinero.



## Costes de operación de TI

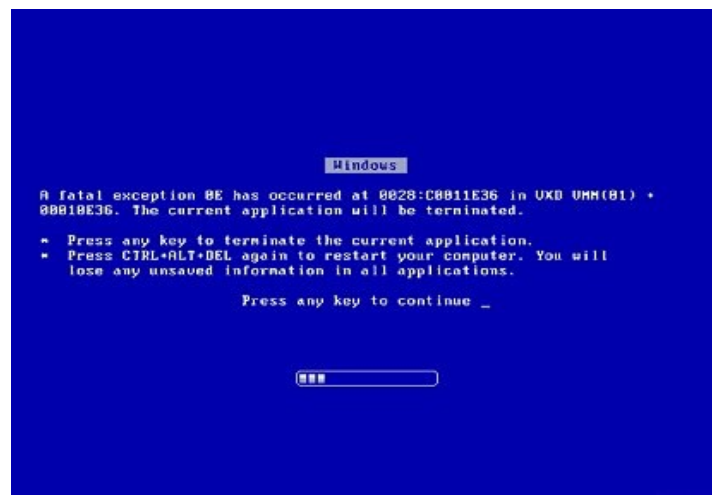
- Service desk.
- Coste de malware: virus, keyloggers, gusanos, spyware...
- Coste de las implantaciones continuas de las medidas de seguridad (parches, reinicios programados, actualización..)
- Despliegue de software.
- Gestión de discos y almacenamiento.
- Gestión de la seguridad.
- Outsourcing y consultoría para la operación.

## **Coste de la administración IT.**

- Gestión del riesgo de la marca.
- Gestión de activos.
- Financiación.
- Formación especializada IT.
- Formación usuarios.
- Outsourcing y consultoría de administración.

## Coste de las paradas de servicio.

- Disponibilidad en número de horas fuera de servicio al año.
- Impacto (\$\$) por hora fuera de servicio.
- Degradación del servicio (horas).
- Impacto (\$\$) por hora de servicio degradada.



# 1. Introducción SL y empresa



## The Value of Linux on HI

Questionnaire

TCO Summary

Details

Print this page

TCO Comparisor					
	Sun Solaris		Linux on HP		Savings with HI
Hardware	\$	178,531	\$	70,130	\$ 108,401
Software	\$	120,000	\$	66,000	\$ 54,000
IT Operations	\$	620,116	\$	181,913	\$ 438,203
IT Administrator	\$	49,797	\$	29,175	\$ 20,622
Facilities and Overhead	\$	50,000	\$	20,000	\$ 30,000
Downtime	\$	46,485	\$	104,592	\$ -58,107
<b>TCO Summary</b>	\$	1,064,929	\$	471,810	\$ 593,119
<b>Savings with HI</b>					56 %
<b>TCO per Syster</b>	\$	106,493	\$	47,181	\$ 59,312
<b>TCO per Use</b>	\$	213	\$	94	\$ 119

TCO Comparison

	Hardware
	Software
	IT Operations
	IT Administration
	Facilities and Overhead
	Downtime

<http://www.artica.es>

calculator.html

## **Bloque 2.**

# Mercado laboral actual en T.I



### **Diferentes perfiles / carreras profesionales**

- Administración de sistemas y operación
- Consultoría
  - Consultoría técnica: Implantación de proyectos.
  - Servicios técnicos: oporte e instalaciones.
  - Formación
- Desarrollo / Programación.
- Docencia en universidades.

### **Administración de sistemas**

- Denostado ( “chispas” ).
- Sin proyección profesional.
- Sueldos medios (máx. 30,000€)
- Creatividad y crecimiento profesional bajo.
- Posible salto a la consultoría si se hace pronto.



### Consultoría

Técnico -> Consultor -> Jefe de proyecto -> Gerente

- Buenos sueldos (30,000 – 40,000 €) \*
- Condiciones laborales duras:
  - Largas jornadas laborales (> 40hr)
  - Viajes frecuentes.
  - Formación continua de productos específicos de gran caducidad.
  - Guardias.
  - Desplazamientos a la periferia, traje y corbata, y **subcontratas**.
- Edad media de “consultor quemado”: 30-35 años

### Desarrollo / programación

Programador -> Analista -> Jefe de proyecto -> Gerente

- Sueldos medios (aprox 25,000€)
- Tecnologías mas demandadas:
  - Java, .net, SAP, cobol, visual Basic.
- Poca proyección profesional, estancamiento y difícil reciclaje.
- El 90% del desarrollo de software en todo el mundo es IN-HOUSE:  
Software para la propia empresa, sin intención de venta a terceros.

### **Tendencias del mercado (1 de 2)**

- Deslocalización: Desarrollo en factorías de software.
- Paquetización de soluciones.
- Suscripciones de servicios de software.
- Servicios gestionados.
- Virtualización.

Todo ello se traduce en una menor necesidad de mano de obra cualificada para realizar lo que hasta ahora se venía haciendo.

### **Tendencias del mercado (2 de 2).**

### **¿Donde desarrollaremos nuestra actividad en el futuro?**

- No la desarrollaremos, la desarrollarán fuera.
- Desarrollo de software fuera de España.
- Mantenimiento de soluciones fuera de España.
- Servicios gestionados fuera de España (SOC, NOC)
- La consultoría se realiza ya con herramientas no producidas aquí, y sólo podemos ofrecer mano de obra que no requiere cualificación para su instalación.
- Nos queda el mantenimiento de grandes aplicaciones: Software bancario: COBOL, Host /390 y similares.

### Soluciones

- Para desarrollar ideas con potencial (de negocio), **es necesario conocer las necesidades del mercado.**
- Para conocer las necesidades del mercado **es necesario trabajar en él varios años.**
- **No cualquier trabajo en TI es válido** para aprender sobre las tendencias del mercado. La consultoría es la mejor forma ya que se ven muchos proyectos en muchos entornos y en diferentes clientes.

## **Bloque 3.**

# Proyectos de Software Libre para la empresa

### **Un caso práctico: Pandora FMS.**

#### **El origen:**

- Originado en el departamento de seguridad de un gran banco.
- Surgido de una necesidad real, de la experiencia diaria.
- Apoyado por el uso diario y alimentándose de necesidades de cada usuario y cada departamento.
- Adaptado desde su nacimiento a las necesidades reales de entornos de producción.

### Un caso práctico: Pandora FMS. Historia

2002

- Version 0 (proyecto ARENA). Un servidor escrito en PERL simplemente procesaba datos no normalizados (RAW) con un formato específico para una plataforma (Checkpoint FW-1 bajo BSD). Las gráficas se generaban usando RRDTool. Se monitorizaron hasta 50 firewalls cada uno con 25 elementos representados. Código cerrado.

2003-2004

- Version 0.8. Reescrito el core en PERL, como FORK del proyecto interno "ARENA". Sólo disponía de agentes sencillos para UNIX (BSD, Linux, y UNIX) y un core monolítico monoproceso que procesaba los datos en formato XML y los almacenaba en una BD. Una sencilla consola visualizaba los datos usando gdlb de forma muy primaria. Licenciado bajo GPL. Nombre inicial "Pandoramon"



### Un caso práctico: Pandora FMS. Historia (II)

Inicios de 2004

- Version 0.9. Se incorporó la posibilidad de generar alertas en un entorno muy simple. Se incorporaron gráficas con JGraph. Se incorporó el primer y rudimentario sistema de gestión de incidentes.

Finales de 2004

- Versión 1.0. Gran avance: se incorporó gestión de eventos, y se mejoró todo el sistema en general. Primera versión con una documentación “formal”.

### Un caso práctico: Pandora FMS. Historia (III)

2005

- Version 1.1. Se crearon los primeros agentes para Windows (escritos en VBScript). Se creó el framework de usuarios con diferentes roles, auditoría interna y administración integrada. Autocreación de módulos y una gran mejora en la gestión de la información y la visualización. Servidor con modelo de proceso multihilo asimétrico no balanceado.

2006

- Version 1.2. Nuevo sistema de red. Nuevos servidores remotos de red, y consola de recepción de traps. Monitorización remota integrada. Procesos multihilo, incorporación de HA. Funcionamiento sobre MySQL en cluster. Agentes nativos para Windows reescritos en C++. Mejora de la escalabilidad con la compresión en RT de la BD.

### Un caso práctico: Pandora FMS. Historia (IV)

2007

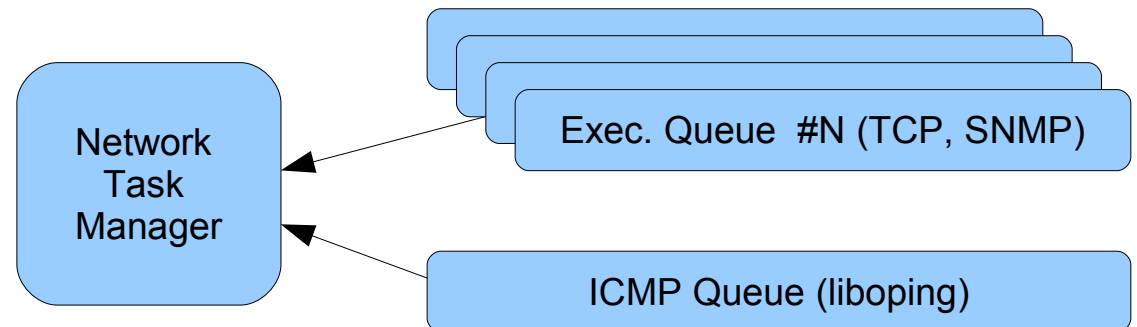
- Version 1.3. Mejora en la escalabilidad: Incremento del rendimiento en el backend SQL hasta un 200%. Creación de wizards de instalación para todas las plataformas y , paquetes RPM. Incorporación de herramientas de reporting en PDF, generación de informes de SLA's, protocolo propio de comunicación con los agentes (Tentacle), consola gráfica JavaScript (drag&drop), mejora en la generación de gráficas, vistas de usuario, y monitorización SNMP mediante templates. Autodescubrimiento de redes.

Fin de la rama 1.x

## Un caso práctico: Pandora FMS. Historia (V)

2007 – principios de 2008

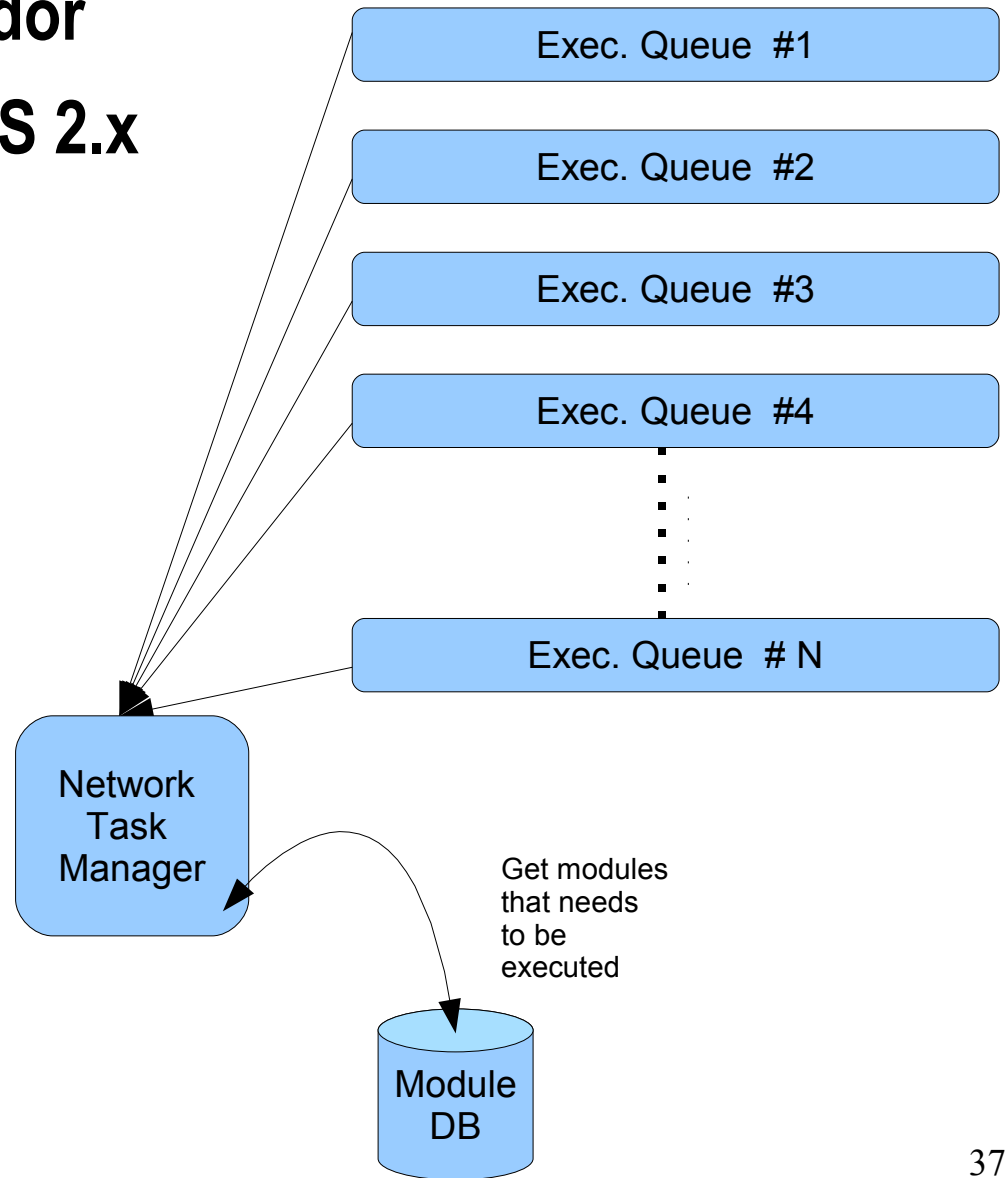
- Versión 2.0. Reescritura del core en C. Multiproceso de ejecución de tareas en red.



- Gestión SNMP mediante plantillas y autodescubrimiento.
- Balanceo de carga y alta disponibilidad integrada en Pandora FMS.
- Alertas combinadas en cascada. Co-relación de alertas.

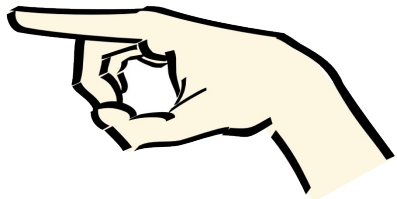
## Arquitectura del servidor de red de Pandora FMS 2.x

- Capaz de monitorizar 10,000 elementos cada 300 segundos.
- Manteniendo un histórico de *meses* de *todos* los datos.



### El futuro de Pandora FMS

- Mayor escalabilidad (Existen organizaciones con muchos más de 10,000 elementos activos).
- Motor de corelación modular (Plugins) para integración con otras herramientas.
- Auto monitorización y detección de nuevos sistemas y redes (en proceso desde la v1.3) \*
- Gestión remota de agentes.
- Inteligencia Artificial aplicada a patrones de tráfico para predecir problemas antes de que ocurran. \*



(\* ) No, no es C/F, OpenView ya lo hace desde hace años.

### Necesidades iniciales

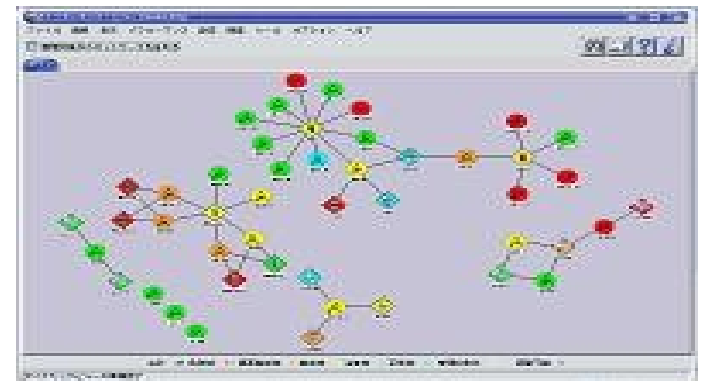
- Multiusuario. Diferentes roles de usuario.
- Administración centralizada y con un interfaz gráfico distribuido.
- Capaz de gestionar cualquier plataforma operativa, especialmente sistemas IPSO (BSD), y UNIX (Solaris, AIX, y HP-UX).
- Necesidad de recolectar información muy específica (Firewalls, aplicaciones proxy inverso, IDS, carga en dispositivos muy específicos via SNMP, detección de sobrecargas en servicios de diversa índole.
- Flexibilidad para su implantación.
- Seguridad en las comunicaciones.
- Sin requisitos previos para su funcionamiento en las maquinas monitorizadas.
- Alertas flexibles que permitan ejecutar scripts.



### Pros y contras

Herramientas libres: Zabbix, Nagios, jffnms, ZenOSS, BigBrother

- **Escalabilidad.** La mayoría de esos sistemas están basados en un backend con base de datos estadísticas RRD. Su escalabilidad está severamente limitada.
- **Backend SQL.** Mantiene un histórico de cada elemento monitorizado. Capaz de aportar expansiones y características tales como reporting avanzado, gráficas, etc
- **Consola gráfica.** Ninguna solución libre tiene una consola como cualquier consola standard en el mundo de software privativo:

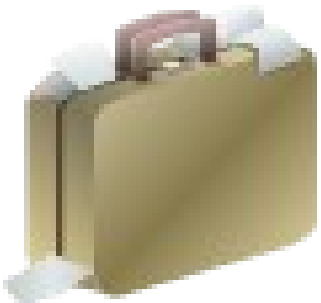




### Pros y contras

Herramientas libres: *Zabbix, Nagios, jffnms, ZenOSS, BigBrother*

- **Informes y SLA's.** Ninguno de ellos genera SLA's para cubrir la demanda de gestión según estándares como **ITIL**.
- **Gestión eficiente de SNMP.** Si no hablas SNMP no hablas en serio.
- **Autogestión de la BD:** Autocompactación. Crecimiento controlado mediante interpolación programada y compresión en tiempo real de los datos almacenados.
- **Redundancia, alta disponibilidad, y componentes distribuidos.** Esencial en grandes entornos.



### Pros y contras

Herramientas no libres: OpenView, Patrol, Tivoli.

- Flexibilidad. Monitorizar algo con Pandora FMS es trivial.
- Código abierto y licenciado bajo GPL2.
- Requerimientos mínimos.
- Multiplataforma real: Independencia total del proveedor.
- Su TCO es mucho menor.



### Proyecto de Software Libre para la empresa

- Requiere I+D continuo -> Mucho trabajo y esfuerzo.
- Colaboración con universidades: cerebro y ciencia.
- Colaboración otras empresas: Desarrollo de características empresariales (reporting, gestión, documentación, características técnicas específicas, y mucha consultoría).
- Colaboración con individuos independientes. Mucho feedback e ideas frescas.

## **Bloque 4.**

# Gestión de proyectos de Software Libre desde la empresa.



### **Crecimiento del proyecto -> Más aportaciones**

- Publicidad. Difusión en listas y foros públicos. Google es tu amigo.
- Creación de comunidad: Listas de correo y foros.
- Versionado continuo.
- Soporte eficiente.
- Documentación exhaustiva.
- Multilinguaje y desarrollo en inglés.



### Dirección efectiva de trabajo

- Colaboradores: poco tiempo y formalidad. Talento variable.
- Empleados: tiempo y dedicación. Recursos limitados.
- Otras empresas: enfoque parcial, dirección propia. Conflictos \$\$\$.
- Los objetivos de los proyectos a menudo son diferentes de los objetivos de un producto viable en el mercado. Es importante poner claros los objetivos desde el principio.
- A más tamaño, más dedicación en tareas de gestión.
- Gestión de equipos independientes en paralelo (Branches).
- Reparto atómico de tareas funcionales.

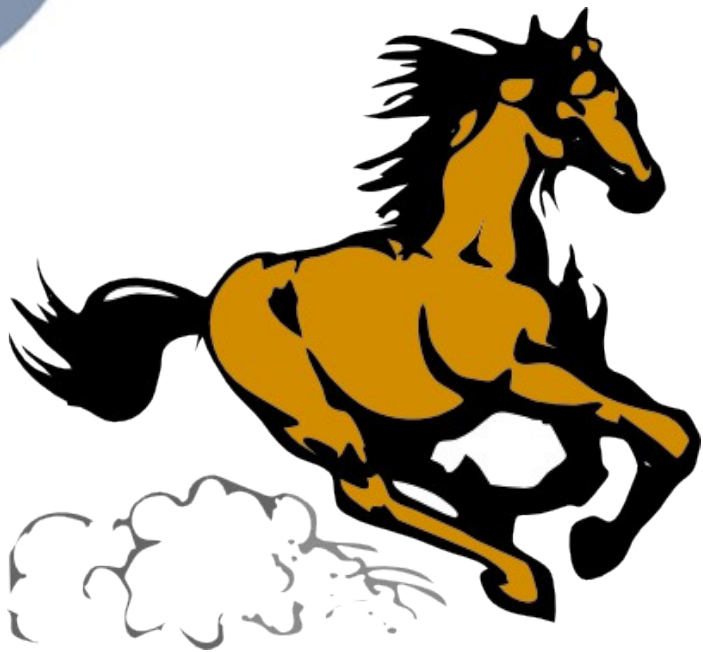


### Herramientas básicas

- Estándares de documentación y buenas prácticas.
- Herramientas colaborativas de gestión documental (DocBook, Wiki)
- Repositorio de código SVN o CVS.
- Listas de correo.
- Gestión de bugs (Tracker).
- Betatesting y equipos de pruebas.
- Equipos de traducción y documentación independientes.

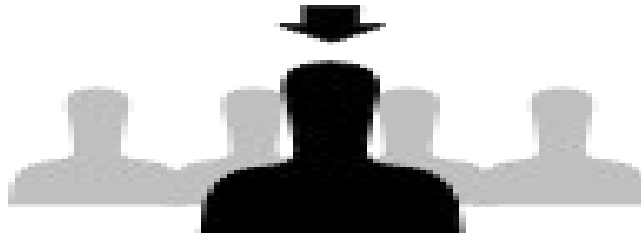
## **Bloque 5.**

# El futuro del Software Libre en la empresa





### ¿¿ El Software Libre no es negocio ??



- SUN, IBM, RedHat, Novell, HP, y SUN han apostado económicamente desde hace años por el software libre y cada vez lo hacen más. Incluso Oracle y SAP están acercando posturas.
- La mayor parte del kernel Linux está mantenido por desarrolladores en nómina de una de estas empresas.
- EEUU en 2007 sólo tuvo un 0.5% de matriculaciones en carreras de informática en sus universidades.
- En España el descenso es notorio. Además existe un importante descenso de natalidad que afecta a la generación de nuevos estudiantes.

### Se avecina una gran demanda de profesionales

- Gran demanda de profesionales de la informática en general, para tareas que no se puedan externalizar.
- Es imprescindible asegurar una capacidad de innovación tecnológica para poder competir con el mercado exterior.
- Habrá una gran demanda de profesionales con conocimientos de Software Libre debido a la fuerte tendencia actual y la gran demanda prevista para el futuro.
- Se demandarán profesionales con conocimientos técnicos y sobre todo de la forma de trabajo inherente al S.L (Metodología, gestión de proyectos) y con redes sociales.
- Con experiencia en colaboración con proyectos de SL y desarrollo en alguna faceta de los mismos.



**Sancho Lerena**  
**slerena@artica.es**

**Gracias**