

---

Tema 1:

# Introducción al software libre

# Objetivos de la asignatura

---

- ¿Qué es el software libre?
- ¿Qué es y qué implicaciones tiene la licencia de un programa libre?
- ¿Cómo se está desarrollando el software libre?
- ¿Cómo se financian los proyectos de software libre, qué modelos de negocio se están experimentando relacionados con ellos?
- ¿Qué motiva a los desarrolladores, especialmente a los que son voluntarios, a involucrarse en proyectos de software libre?
- ¿Cómo son estos desarrolladores?
- ¿Cómo se coordinan en sus proyectos, y cómo es el software que producen?

# Definición de software libre

---

- Características del **software privativo** (antónimo de software libre).
  - El comercializador de un programa impone las condiciones de uso.
  - Si un programa tiene errores solo su productor puede arreglarlos.
  - No se puede adaptar un programa a necesidades específicas propias
  - Modelo de ingresos fundamentalmente basado en prohibir al usuario su derecho de copia
  
- Definición de software libre (SL), tal y como fue concebido por **Richard Stallman**, es todo software que cumple las siguiente libertades:
  - 1.- Libertad para ejecutar el programa en cualquier sitio, con cualquier propósito y para siempre.
  
  - 2.- Libertad para estudiarlo y adaptarlo a nuestras necesidades. Esto exige el acceso al código fuente.
  
  - 3.- Libertad de redistribución, de modo que se nos permita colaborar con vecinos y amigos.
  
  - 4.- Libertad para mejorar el programa y publicar las mejoras. También exige el código fuente.

# Definición de software libre

---

## ■ Licencias

- Es el mecanismo que se utiliza para garantizar, de acuerdo con la legalidad vigente lo siguiente:
  - Las 4 libertades del SL.
  - Permisos para que el receptor del programa pueda ejercer esas libertades
  - Restricciones
    - Dar crédito a los autores originales en caso de redistribución.
    - No pueden ir en contra de las 4 libertades del SL.
- Desde un punto de vista práctico hay varios textos que definen más precisamente que condiciones tiene que cumplir una licencia para ser considerada como de SL
  - Definición de SL de la Free Software Foundation.  
<http://www.gnu.org/philosophy/free-sw.html>
  - Directrices de Debian para decidir si un programa es libre.  
[http://www.debian.org/social\\_contract.html#guidelines](http://www.debian.org/social_contract.html#guidelines)

# Definición de software libre

---

- Free software → no significa software gratuito
  - Sin embargo,
    - por la 3ª libertad, cualquiera que posea un programa puede redistribuirlo sin pedir contraprestación económica ni permiso,
    - lo que hace que el coste de un programa tienda al coste marginal de copia.
  - Puede cobrarse por el valor añadido que supone
    - dar una cierta garantía,
    - selección, actualización y organización de un conjunto de programas,
    - servicio técnico,
    - personalización del software, etc.

# Definición de software libre

---

## ■ Términos relacionados

### ■ Open source software (programas de fuente abierta)

- Promovido por Eric Raymond y la Open Source Initiative

<http://opensource.org/>

<http://opensource.org/faq#free-software>

- Criticado por Richard Stallman y la Free Software Foundation

<https://www.gnu.org/philosophy/open-source-misses-the-point.es.html>

- Hace énfasis en la disponibilidad del código fuente no en la libertad.

- Más políticamente aséptico.

- La disponibilidad del código como ventaja técnica para:

- El desarrollo de mejores modelos de desarrollo y negocio.
- Mayor seguridad.
- Mayor detección/corrección de errores.
- Mayor integración, etc.

- Más información sobre software libre vs fuente abierta

- Free Software vs Open Source

<http://tuxfiles.wordpress.com/free-software-vs-open-source/>

- ¿Código abierto es lo mismo que software libre?

<http://www.cobdc.net/programarilliure/codigo-abierto-es-lo-mismo-que-software-libre/>

# Definición de software libre

---

- Términos relacionados

- Freeware (programas gratuitos)

- Normalmente se distribuyen sólo en binario, y se pueden obtener sin coste.
    - A veces se obtiene también permiso de redistribución, pero otras no, pudiendo obtenerse entonces sólo del sitio “oficial” mantenido a ese efecto.
    - Es habitual que estos programas se usen para promocionar otros programas (típicamente con funcionalidad más completa) o servicios.
    - Ejemplos:
      - Skype,
      - Google Earth o
      - Microsoft Messenger.

# Definición de software libre

---

## ■ Términos relacionados

### ■ Shareware

- Es un método de distribución, ya que los programas, generalmente sin fuentes, se pueden copiar libremente, pero no usar continuadamente sin pagarlos.
- La exigencia de pago puede estar incentivada por:
  - funcionalidad limitada,
  - mensajes molestos,
  - una simple apelación a la moral del usuario,
  - Amenazas legales, etc.

### ■ Charityware, Careware

- Generalmente shareware, pero cuyo pago o aportación voluntaria es para una organización caritativa patrocinada.
- Ejemplo, *vim*: solicita aportaciones voluntarias.



# Definición de software libre

---

## ■ Términos relacionados

### ■ Dominio público

- El autor renuncia absolutamente a todos sus derechos.
- Tiene que estar declarado explícitamente en el programa, ya que si no se dice nada, el programa es propietario.
- Si además se proporcionan los fuentes, el programa es libre.

### ■ Copyleft

- Software libre cuya licencia obliga a que las modificaciones que se distribuyan sean también libres.

### ■ Propietario, Cerrado, No libre

- Software que no es libre ni de fuente abierta.

# Motivaciones

---

## ■ Motivación ética

- Abanderada por la Free Software Foundation ([www.fsf.org](http://www.fsf.org))
- Heredera de la cultura hacker.
- Partidaria del apelativo libre.
- Basada en que el software es conocimiento que debe poder difundirse sin trabas, y que su ocultación es una actitud antisocial y que la posibilidad de modificar programas es una forma de libertad de expresión.

## ■ Motivación pragmática

- Abanderada por la Open Source Initiative ([www.opensource.org](http://www.opensource.org)).
- Partidaria del apelativo fuente abierta.
- Basada en ventajas técnicas y económicas.

## ■ Otras motivaciones

- Diversión.
- Aprendizaje.
- Económica basada normalmente en modelos de negocio sostenible.
- Etc.

# Consecuencias de las libertades del software

---

## ■ Económica

- No es posible obtener mucho dinero con la distribución.
- Se necesitan otros modelos de negocio y otros mecanismos de financiación.
- En muchos casos la falta de soporte o la calidad escasa están relacionadas con la financiación.
  - En muchos casos son falsas.
  - Incluso software sin ningún tipo de financiación suele ofrecer muy buen soporte a través de foros de usuarios y desarrolladores, y muchas veces tiene gran calidad.
- El modelo de costes del software libre es muy distinto del propietario,
  - gran parte de él se ha desarrollado fuera de la economía formal monetaria,
  - muchas veces con mecanismos de trueque:
    - “yo te doy un programa que te interesa y tú lo adaptas a tu arquitectura y le haces mejoras que a ti te interesan”.
- Gran parte de los costes disminuyen por el hecho de ser libre.
  - Se pueden reutilizar software.
  - La distribución se hace por Internet y tiene un coste mucho menor.
  - Propaganda gratuita en foros públicos destinados a ello.

# Consecuencias de las libertades del software

---

- La calidad

- Derivada de la *colaboración voluntaria* de gente que contribuye:
  - en el desarrollo,
  - prueba y reporte de errores en entornos y situaciones inimaginables por el desarrollador original.
- Si un programa no ofrece la calidad suficiente, *la competencia* puede tomarlo y mejorarlo, partiendo de lo que hay.
- La colaboración y la competencia, dos poderosos mecanismos, se combinan para conseguir mejor calidad.

- Examinemos ahora las consecuencias beneficiosas según el destinatario.

# Consecuencias de las libertades del software. Según el destinatario.

## ■ Para el usuario final

### ■ Verdadera competencia

- Tendencia al monopolio en el mercado de software privativo
- No depende necesariamente del soporte del fabricante del software,
  - puede haber múltiples empresas, quizá pequeñas, que disponiendo del fuente y de conocimientos, puedan hacer negocio manteniendo determinados programas libres.

### ■ Evaluación de la calidad del software

- Ya no se depende tanto de la fiabilidad del fabricante
- Aceptación de la comunidad y la disponibilidad de los fuentes.

### ■ Eliminación de cajas negras, en las que hay que confiar porque sí.

### ■ Independencia de estrategias de los fabricantes, que pueden decidir unilateralmente dejar de mantener un producto.

# Consecuencias de las libertades del software. Según el destinatario.

## ■ Para el usuario final

- La evaluación de productos antes de adoptarlos ahora es mucho más sencilla.
  - Basta instalar los productos alternativos en nuestro entorno real y probar,
  - Para software propietario hay que fiarse de informes externos o negociar pruebas con los proveedores, lo cual no es siempre posible.
- Posibilidad de personalizar o adaptar el software a sus necesidades.
- Mejora del proceso de detección y corrección de errores.
  - En software propietario suele ser extremadamente penoso, si no imposible, ya que si conseguimos que se repare, muchas veces se hará en la versión siguiente, que podría tardar años en salir, y a veces además será necesario comprarla de nuevo.
  - En software libre, sin embargo, lo podemos hacer nosotros, si estamos cualificados, o contratar el servicio fuera.
- Posibilidad de integrar el programa con otros.
- Posibilidad de auditar su calidad (por ejemplo la seguridad), etc,
- En resumen el control pasa, en gran medida, del proveedor al usuario.

# Consecuencias de las libertades del software. Según el destinatario.

## ■ Para la administración pública

- La administración pública tiene obligaciones especiales con el ciudadano, como:
  - proporcionarle servicios accesibles y neutrales respecto a los fabricantes,
  - garantizar la integridad, utilidad, privacidad y seguridad de sus datos a largo plazo.
- Todo ello la obliga a ser
  - más respetuosa con los estándares,
  - mantener los datos en formatos abiertos,
  - manipular los datos con software que no dependa de estrategia de empresas, generalmente extranjeras.
  - certificar el software como seguro por auditoría interna.
- La adecuación a estándares es una característica notable del software libre.
  - No ocurre así con el software propietario, generalmente ávido de crear mercados cautivos.
- La administración tiene una cierta función de escaparate y guía de la industria
  - debería dirigirse a la creación de un tejido tecnológico generador de riqueza,
    - fomentando empresas cuyo negocio sea,
      - el desarrollo de nuevo software libre para la administración,
      - el mantenimiento, adaptación o auditoría del software existente.

# Consecuencias de las libertades del software. Según el destinatario.

## ■ Para el desarrollador

- Es más fácil competir siendo pequeño y adquirir tecnología punta.
- Puede competir con un producto modificando su propio código.
  - si bien también el competidor copiado se aprovechará de nuestro código (si es copyleft).
- Se puede conseguirse la colaboración gratuita de mucha gente.
- Se tiene acceso a un sistema de distribución prácticamente gratuito y global.
- ¿Cómo obtener recursos económicos si el software realizado no es fruto de un encargo pagado? Más adelante.



# Consecuencias de las libertades del software. Según el destinatario.

## ■ Para el integrador

- No más cajas negras que intentar encajar, a menudo con ingeniería inversa.
- Puede limar asperezas e integrar trozos de programas para conseguir el producto integrado necesario.
- Disponiendo de un conjunto ingente de software libre de donde extraer los módulos que necesite.

## ■ Para el que proporciona mantenimiento y servicios

- Lo sitúa casi en las mismas condiciones que el productor.
  - Si no son las mismas es porque hace falta un conocimiento profundo del programa que sólo el desarrollador posee,
  - por lo que es conveniente que el mantenedor participe en los proyectos que se dedica a mantener.
- El valor añadido de los servicios es mucho más apreciado, ya que el coste del programa es bajo.
- Éste es actualmente el negocio más claro con software libre y con el que es posible un mayor grado de competencia.

# Otras referencias

---

- Conferencia de Richard Stallman en la Universidad de Córdoba 14/4/2008  
[http://www.webislam.com/videos/57218-charla\\_de\\_richard\\_stallman\\_en\\_la\\_universidad\\_de\\_cordoba.html](http://www.webislam.com/videos/57218-charla_de_richard_stallman_en_la_universidad_de_cordoba.html)  
<http://www.youtube.com/watch?v=zPt7LW8uk4I>
- Conferencia de Richard Stallman, “Copyright vs. Comunidad”, en la Facultad de Derecho de la Universidad de Córdoba.  
<http://www.ustream.tv/channel/stallman2010-cordoba>
- The Free Software Foundation (FSF)  
<https://www.fsf.org/>
- Proyecto GNU  
<http://www.gnu.org/home.es.html>
- Open Source Initiative  
<http://www.opensource.org>
- Open Sources. Voices from the Open Source Revolution:  
<http://www.oreilly.com/catalog/opensources/>
- BarraPunto:  
<http://barrapunto.com>