

Ejercicios Tema 1

2.- ¿Qué diferencias existen entre las distintas clases de Software según los derechos de autor? Proponga ejemplos de programas de las distintas clases.

Existen 3 clases de Software según los derechos de autor:

- *Código abierto:*

- Existen personas que utilizan la expresión de “código abierto” para referirse más o menos a la misma categoría a la que pertenece el “software libre”.

Sin embargo no son el mismo tipo de software, ya que el *código abierto* acepta algunas directivas que se consideran demasiado restrictivas y viceversa, ya que hay algunas licencias de *software libre* que ellos no han aceptado.

Aún así las diferencias entre ellos son mínimas, ya que la mayoría del software libre es código abierto, y casi todo el código abierto es software libre.

- *Software libre:*

- es aquel que se suministra con autorización para que cualquiera pueda, usarlo, copiarlo y/o distribuirlo, ya sea con o sin modificaciones, de forma gratuita o mediante pago. Esto quiere decir que el *código fuente debe estar disponible*.

- Ejemplos:

- Sistema Operativo LINUX
- Open Office
- NotePad ++
- eMule
- FileZilla
- Etc.

- *Código cerrado:*

- El software cerrado o privativo, es otro nombre para designar el software que no es libre. Antiguamente, estaba subdividido el software que no es libre en “software semilibre” (podía modificarse y redistribuirse sin fines comerciales) y el “software privativo” (no podía ser modificado ni redistribuido).

Hoy día se usa el término de “software privativo” como sinónimo de software que no es libre.

- Ejemplos:

- Windows (Microsoft)
- Microsoft Office (Microsoft)
- iTunes (Apple)
- Nero (NeroAG)
- Norton Antivirus (Symantec)
- Etc.

- De dominio público:
 - Es aquel que no tiene derechos de autor. Si el código fuente es de dominio público, se trata de un caso especial de *software libre sin copyleft*₁, lo que quiere decir que puede haber algunas copias o versiones que no sean libres en absoluto.

Existen casos en los que un programa puede ser de dominio público pero no disponer del código fuente de forma libre. Algunas personas utilizan el término “dominio público” para referirse a “libre” o “gratuito”, cosa que es falsa.

- Ejemplos:
 - UNIX
 - SQLite
 - Etc.

1. Programas con permiso de redistribución y modificación, como así también con el permiso de agregarle restricciones.

2.- ¿En qué consiste la Programación Extrema (XP)?

La programación extrema es una *metodología de desarrollo ágil* que tiene como principal objetivo aumentar la productividad a la hora del desarrollo de software, dando prioridad a los trabajos que dan un resultado directo.

Una metodología de desarrollo ágil tiene como punto más fuerte, la adaptación a cualquier cambio en un proyecto para aumentar sus posibilidades de éxito.

Cuenta con varios principios:

- Los individuos y sus interacciones son más importantes que los procesos y las herramientas.
- El software que funciona es más importante que la [documentación exhaustiva](#).
- Colaboración con el cliente en lugar de negociación de contratos.
- No hay que seguir un plan cerrado, sino adaptarse al cambio.

Tenemos **12 principios básicos** que se agrupan en 4 categorías:

- **Retroalimentación:**
 - Principio de pruebas:
 - Basicamente se define lo que debe hacer el software desarrollado.
 - Planificación:
 - El cliente definirá sus necesidades para concretar las actividades que el sistema debe realizar.
 - Cliente in-situ:
 - El cliente/representante, deberá formar parte del equipo de desarrollo.
 - Pair-programming:
 - Consiste en escribir código en parejas compartiendo una sola máquina.
- **Proceso continuo en lugar de por bloques:**
 - Integración continua:
 - consiste en implementar de forma progresiva las nuevas características del software.

- Refactorización:
 - mediante la eliminación de código duplicado se mejora el diseño del sistema.
- Entregas pequeñas:
 - se evalúa el producto en un ambiente real mediante la colocación de un sistemas sencillo en producción.
- **Entendimiento compartido**
 - Diseño simple:
 - el mejor programa será aquel que cumpla los requisitos y sea más simple.
 - Metáfora:
 - expresa la visión evolutiva del proyecto y define los objetivos del sistema mediante una historia.
 - Propiedad colectiva del código:
 - el código tiene propiedad compartida. Nadie es propietario de nada, ni si quiera de lo que ha desarrollado.
 - Estandar de programación
 - define las reglas para escribir y documentar el código, además de como se comunican las diferentes piezas de código.
- **Bienestar del programador**
 - Semana de **40 horas**:
 - los programadores cansados escriben peor código. *Es importante minimizar las horas extras y mantener a los programadores frescos y descansados.*

Referencias:

1. Programación extrema (<https://geekytheory.com/programacion-extrema-que-es-y-principios-basicos>)
2. Clases de software (<https://www.gnu.org/philosophy/categories.es.html>)