

# SOFTWARE PROPIETARIO

Salinas

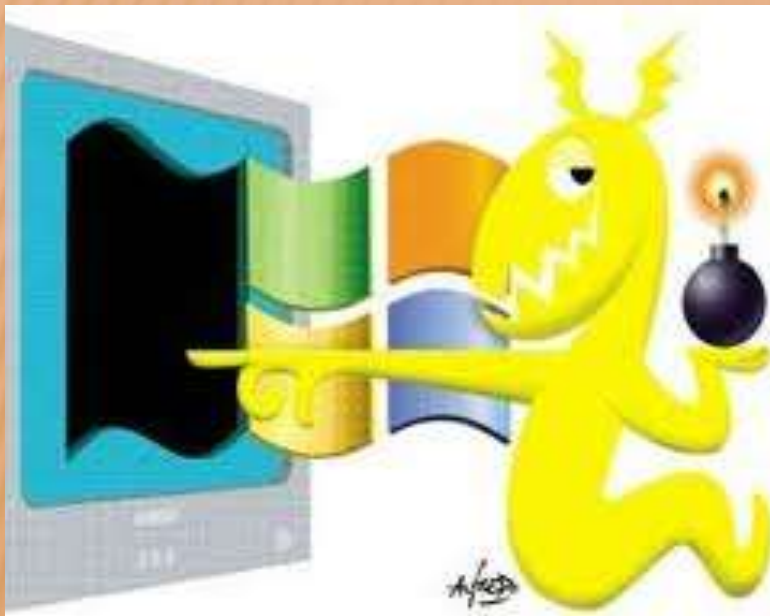
Güicho Diego

Eduardo

Obregón León

Nancy

Marever



# ÍNDICE\*

- I. Software
- II. Estructura de Software
- III. Software propietario
- IV. Su historia
- V. Ventajas & Desventajas
- VI. Características
- VII. Ejemplos
- VIII. Acerca de Shareware
- IX. Críticas



---

SOFTWARE



- ✖ Primero que nada,  
¿Que es software y cuál es su estructura?  
Software se refiere al equipamiento lógico o soporte lógico de una computadora digital, y comprende el conjunto de los componentes lógicos necesarios para hacer posible la realización de tareas específicas;  
en contraposición  
a los componentes físicos del sistema, llamados hardware.



---

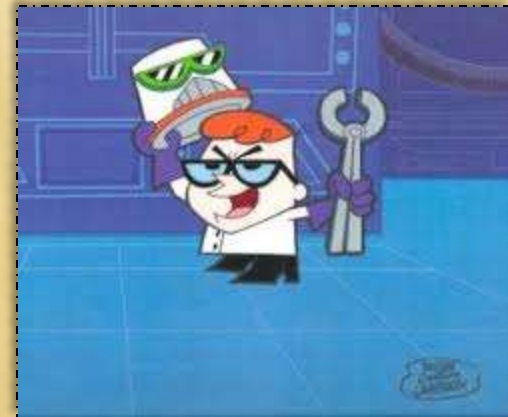
# SU ESTRUCTURA

## ✖ Ahora bien , hablemos de su estructura:

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de Software o Arquitectura lógica. Se refiere a un grupo de abstracciones y patrones que nos brindan un esquema de referencia útil para guiarnos en el desarrollo de software dentro de un sistema informático.

Así, los programadores, diseñadores, ingenieros y analistas pueden trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo deseado.

Algunos objetivos dentro de un esquema de Arquitectura de Software pueden ser: el software debe ser mantenible, esto es, fácilmente analizable, modificable, corregible; también puede ser un objetivo el nivel de interacción con otros sistemas informáticos, o su escalabilidad.





- 
- ✖ Estas Arquitecturas están definidas muchas veces por el tipo de tecnología a la cual se enfrenta un programador o grupo de programadores, por lo cual algunos tipos de arquitectura son más recomendables que otras para ciertas tecnologías.
  - ✖ Cada tarea de computación es asignada a una computadora, por lo cual una Arquitectura determinada debe ser implementada físicamente y definir de forma abstracta los componentes que tomarán parte en las tareas y sus interfaces comunicativas.

- 
- ✘ Todo esto se desarrolla a "alto nivel", ensamblando elementos para lograr la mayor funcionalidad posible siendo a la vez portable, logrando disponibilidad, escalabilidad y confiabilidad.
  - ✘ Como ejemplos de Arquitecturas podemos citar las monolíticas (los grupos funcionales del software están altamente acoplados entre sí), cliente-servidor (se reparte la carga de cómputo en dos partes independientes), y la arquitectura de tres niveles (la carga se divide entre tres partes: presentación, cálculo y almacenamiento).



---

# SOFTWARE PROPIETARIO

## Software propietario

El **software propietario** (también llamado **privativo**, de **código cerrado** o **software no libre**) es cualquier programa informático en el que el usuario final tiene limitaciones para usarlo, modificarlo o redistribuirlo (con o sin modificaciones), o cuyo código fuente no está disponible o el acceso a éste se encuentra restringido por un acuerdo de licencia, o por tecnología anti copia.

Para la Fundación para el Software Libre (FSF) este concepto se aplica a cualquier software que no es libre o que sólo lo es parcialmente (semilibre), sea porque su uso, redistribución o modificación está prohibida, o requiere permiso expreso del titular del software.



---

# HISTORIA

WISCONSIN



- ✗ HISTORIA:
- ✗ • Nace en la década de los 70.
- ✗ • Se crean Licencias, para la adquisición y disfrute del Software Propietario.
- ✗ • En 1975, Bill Gates y Paúl Allen, fundan Microsoft.
- ✗ • En 1991, IBM y Microsoft, dejan de cooperar en sistemas operativos.  
IBM: MS-Dos y Microsoft con Windows.
- ✗ • Nacen empresas de software propietario: Adobe, Autodesk y Macromedia.
- ✗ • En 1997, Microsoft es demandada por Justicia de los EEUU. Monopolio comprobado.
- ✗ • A partir del año 2002, se empezó a operar los sistemas operacionales con software libre.
- ✗ • Windows 3.0 se lanza en 1990.
- ✗ • En 1993 Nace Windows NT.



---

# VENTAJAS & DESVENTAJAS



- ✗ VENTAJAS.
- ✗ 1. Empresas tienen demasiados controles de calidad.
- ✗ 2. Tiene personal muy capacitados y con mucha experiencia.
- ✗ 3. El mercado lo tomo como estándar, así que la gran mayoría sabe usarlo.
- ✗ 4. Existen aplicaciones muy específicas que no existirían si no fuera propietario.
- ✗ 5. Existen muchas aplicaciones que solo corren en Windows.



- ✗ DESVENTAJAS.
- ✗ 1. Es difícil aprender a utilizar otro Software, sin un curso.
- ✗ 2. El funcionamiento del Software es complicado y solo lo saben los desarrolladores.
- ✗ 3. El soporte técnico es insuficiente y tarda en dar una respuesta.
- ✗ 4. Es ilegal modificar una aplicación.
- ✗ 5. Los clientes quedan dependiendo del proveedor.
- ✗ 6. La empresa puede quebrar y se pierde soporte técnico y actualización.
- ✗ 7. Las licencias son caras.



---

# CARACTERÍSTICAS



- ✖ CARACTERÍSTICAS DEL SOFTWARE PROPIETARIO:
- ✖ • Dependencia de un proveedor.
- ✖ • Falta de Garantías.
- ✖ • Falta de Soporte.
- ✖ Es importante recalcar algunos puntos importantes para un buen desarrollo del mismo:
- ✖ - El software se desarrolla o construye; no se manufactura en el sentido clásico. A pesar de que existen similitudes entre el desarrollo del software y la manufactura del hardware, las dos actividades serían diferentes en lo fundamental.





- ✗ · El software no se desgasta. El software es inmune a los males ambientales que desgasten el hardware. Por lo tanto la curva de tasas de fallas para el software debería tener la forma de la “curva idealizada”. Los defectos sin descubrir causan tasas de fallas altas en las primeras etapas de vida de un programa. Sin embargo, los errores se corrigen y la curva se aplanan: el software no se desgasta, pero si se deteriora.
- ✗ · A pesar de que la industria tiene una tendencia hacia la construcción por componentes, la mayoría del software aun se construye a la medida. Un componente de software se debe diseñar e implementar de forma que puede utilizarse en muchos programas diferentes

- ✗ CARACTERÍSTICAS DE UN BUEN SOFTWARE
- ✗ CORRECCIÓN.
- ✗ FIABILIDAD.
- ✗ EFICIENCIA.
- ✗ INTEGRIDAD.
- ✗ FACILIDAD DE USO.
- ✗ FACILIDAD DE MANTENIMIENTO.
- ✗ FLEXIBILIDAD.
- ✗ FACILIDAD DE PRUEBA.
- ✗ PORTABILIDAD.
- ✗ FACILIDAD DE REHUSO.
- ✗ INTEROPERABILIDAD.





EJEMPLOS



- ✗ Ya habíamos mencionado que el Software propietario Software que no es ni libre ni semi-libre. El uso, modificación y redistribución está prohibido y se requiere que se adquiriera un permiso.

5 ejemplos:

Windows XP, vista, 98, 2000 etc.

Antivirus de diferentes marcas

Programas de tareas como Office

Programas Editores de Música video etc. Así como  
Convertidores

Programas de Creación y modificación ya sea de sitios  
web o tridimensionales en un ejemplo rápido

Macromedia en sus diferentes versiones



---

# SHAREWARE

## ✗ Shareware

Es software que permite a los usuarios probar y realizar copias de los programas, pero requiere que todos aquellos que decidan usar el programa adquieran una licencia de uso. El shareware no es un tipo de programa sino una forma de distribución. El autor del programa no retiene los derechos de distribución, pero si los demás. Generalmente el software shareware se distribuye sin fuentes, y no se permite ni modificarlo ni redistribuirlo salvo en los términos anteriores. Es software que viene con autorización para la gente de redistribuir copias, pero dice que quien continúe haciendo uso de una copia deberá pagar un cargo por licencia. El shareware no es software libre, ni siquiera semilibre. Existen dos razones por las que no lo es:





- 
- ✗ 1. Para la mayoría del shareware, el código fuente no está disponible; de esta manera, usted no puede modificar el programa en absoluto.  
2. El shareware no viene con autorización para hacer una copia e instalarlo sin pagar una cantidad por licencia, ni aún para particulares involucrados en actividades sin ánimo de lucro. (En la práctica, la gente a menudo hace caso omiso a los términos de distribución y lo hace de todas formas, pero los términos no lo permiten.).  
Pero por otro lado, el shareware aporta una ventaja con respecto a los programas distribuidos por los canales tradicionales, y es que el usuario prueba el programa antes de comprarlo, evitando de esta forma desagradables sorpresas, y teniendo la seguridad que lo que compra va a ser de su entera satisfacción.

- ✖ Naturalmente que hay muchos usuarios que usan programas shareware y no se registran. Obviamente nadie va a ir a su casa para ver si está utilizando algún programa y no ha pagado por él, pero todos deberíamos de comprender que el shareware nació con la idea de proporcionar software a bajo precio, y que si no cumplimos con nuestra obligación moral de registrarnos en los programas que utilizamos, el objetivo no se va a cumplir.



- ✖ Como los programadores saben que el porcentaje de usuarios que se registran en sus programas es muy bajo comparado con el porcentaje de usuarios que realmente está utilizando el programa, generalmente ofrecen a los usuarios que se registran algunas ventajas como manuales impresos, ayuda técnica gratuita o habilitación para opciones deshabilitadas en la versión no registrada, como la posibilidad de guardar trabajos, imprimirlos o exportarlos a formatos más estándares.





- ✖ 5 Ejemplos:  
Internet Explorer  
Flash Player  
Adobe Reader  
Adobe Shockwave  
Java  
etc.



- ✖ Todos estos programas los puedes copiar y redistribuir sin ningún problema solo se tiene que firmar el acuerdo de licencia

---

# CRÍTICAS



## ✖ Críticas

- ✖ En la filosofía del proyecto GNU consideran *inmoral* la instalación del software de código cerrado. Su fundador, Richard Stallman ha indicado en múltiples ocasiones que el software de código cerrado al ser adquirido por un particular genera inmediatamente un dilema ético si se muestra tal programa a un amigo y este nos lo pide, aparece el dilema: o soy un buen cliente y entonces mal amigo -por no compartir- o soy un buen amigo y entonces mal cliente y en algunos casos perseguido por la ley por prestar mi software.
- ✖ Dadas las características del software de código cerrado un usuario común ignora absolutamente el contenido del mismo y por tanto si existe dentro de las líneas del código alguna amenaza contra su equipo o su información, además el usuario no sólo tiene prohibido el intentar eliminar o cambiar esa parte del código sino que puede ser perseguido por la ley por el hecho de intentar conocer si existe tal amenaza en dicho software.
- ✖ Por otra parte, en una sociedad de la información, el software se ha convertido en una herramienta importante de productividad, y una licencia de software privativo constituye un acuerdo o contrato entre dos sujetos jurídicos que voluntariamente acuerdan las condiciones de uso de un programa.



MUCHAS GRACIAS \* 😊

