# Software y programa. Tipos de software

1. ¿Cómo se denomina al conjunto de programas que tienen una finalidad más o menos concreta?

Las aplicaciones informáticas.

1. ¿Qué tipo de software es un videojuego?

Es de tipo aplicación.

1. ¿Cómo se denomina al conjunto de herramientas que nos permiten desarrollarprogramas informáticos?

Software de programación

1. ¿Qué tipo de software es un depurador?

Es de tipo programador

1. ¿Qué tipo de software es un compilador?

Es de tipo programador

1. ¿De qué dos partes consta un ordenador?

Del hardware y el software.

1. ¿Cómo se denomina al software base que ha de estar instalado y configurado en nuestro ordenador para que las aplicaciones puedan ejecutarse y funcionar?

Se llama sistema operativo.

1. ¿Qué tipo de software es un controlador de dispositivo?

Es de tipo sistema.

1. ¿Qué tipo de software es una aplicación ofimática?

Es de tipo aplicación.

1. ¿Qué tipo de software es un sistema operativo?

Es de tipo sistema.

# Relación hardware-software

1. ¿Cuál es la relación hardware-software desde el punto de vista del sistema operativo?

El sistema operativo es el encargado de coordinar al hardware durante el funcionamiento del ordenador, actuando como intermediario entre éste y las aplicaciones que están corriendo en un momento dado. Todas las aplicaciones necesitan recursos hardware durante su ejecución (tiempo de CPU, espacio en memoria RAM, tratamiento de interrupciones, gestión de los dispositivos de Entrada/Salida, etc.). Será siempre el sistema operativo el encargado de controlar todos estos aspectos de manera "oculta" para las aplicaciones y para el usuario.

1. Define hardware

El conjunto de dispositivos físicos que forman parte de un ordenador, por ejemplo: impresora, pantalla, teclado, ratón, disco duro externo, usb.

1. Investiga en google cual fue la gran aportación de la Arquitectura Von Neumann. (Pista busca el concepto de programa almacenado)

Las computadoras son máquinas de arquitectura von Neumann cuando: Tanto los programas como los datos se almacenan en una memoria en común. Esto hace posible la ejecución de comandos de la misma forma que los datos. Cada celda de memoria de la máquina se identifica con un número único, llamado dirección.

1. ¿Cuál es la relación hardware-software desde el punto de vista de las aplicaciones?

Una aplicación no es más que un conjunto de programas, y están escritos en algún lenguaje de programación que el hardware y el equipo deben interpretar y ejecutar.

1. Investiga cómo podemos con el “Administrador de tareas” de Windows ver como el sistema operativo asigna recursos a las aplicaciones. (Haz alguna captura donde se vea la asignación).

**La mayoría de usuarios utiliza el administrador de tareas de Windows únicamente para finalizar una aplicación congelada que bloquea el sistema. Y poco más. Seguramente se están perdiendo otras posibilidades que ofrece esta herramienta.**

**Cómo acceder al administrador de tareas**

* **Atajo de teclado Ctrl + Alt + Supr. Este es el método que casi todo el mundo conoce. Se puede utilizar pero en Windows 10, no se inicia directamente el Administrador de tareas y tendrás que hacer un clic adicional para ponerlo en marcha.**
* **Atajo de teclado Ctrl + Shift + Esc. Mucho mejor que el anterior, lanza inmediatamente el Administrador de tareas y encima se puede utilizar con una sola mano.**

Interfaz de usuario gráfica, Aplicación, Tabla, Word

Descripción generada automáticamente

* **Barra de tareas. Si haces clic derecho en la barra de tareas hay una opción para acceder al administrador.**

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

* **Menú de usuario avanzado. Otro acceso rápido utilizando el ratón. Haz clic en el botón derecho en el botón de inicio para acceder al menú avanzado y también encontrarás este Task Manager.**

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

# Licencias de Software

1. ¿En qué se parecen el Shareware y el Freeware?

Que ambos permiten la redistribución.

1. ¿Quién elige la licencia de distribución de un software?

El desarrollador.

1. ¿Cómo se denomina a un conjunto de derechos exclusivos garantizados por un gobierno o autoridad al inventor de un nuevo producto?

Se le denomina patente.

1. ¿Qué es el software semi libre?

Es un tipo de software que no es libre, pero tiene autorización de usar, copiar, distribuir y modificar para particulares sin ánimo de lucro.

1. ¿Qué es un programa adware?

Es un subprograma que descarga publicidad de otro programa principal.

1. ¿Qué forma de protección proporcionan las leyes para los autores de obras originales?

Derecho de autor o copyright.

1. ¿En qué se parecen el software libre y el semi libre?

En que ambos pueden ser usados, distribuidos, estudiados y modificados.

1. ¿Qué es el software de dominio público?

Es un tipo de software que no está protegido con copyright.

1. ¿Qué es el software libre?

Es un software cuyo código fuente puede ser analizado, usado, redistribuido y modificado con cualquier finalidad.

1. ¿Cómo se denomina a un contrato que se establece entre el desarrollador de un softwaresometido a propiedad intelectual y a derechos de autor, y el usuario?

Se le llama licencia de software.

1. ¿Cómo se denomina al software uso, redistribución o modificación están prohibidos o necesitan una autorización?

El software propietario.

1. ¿El software libre es gratuito?

No, no lo es.

1. ¿En qué se diferencian el software libre y el semi libre?

En que el software semi libre no se considera como libre.

1. ¿Cómo se denomina al software que programas que permiten la redistribución pero no la modificación (y su código fuente no está disponible)

Se le llama Freeware.

1. ¿En qué se diferencian el Shareware y el Freeware?

En que el Shareware se le debe pagar por una Licencia de uso continuado, mientras que el Freeware no permite la modificación.

1. Investiga en internet la relación entre un software con licencia trial y un software shareware.

##### **SHAREWARE**

**Este modelo de distribución consiste en que el programa sea usado por el consumidor de forma gratuita para que éste lo pueda evaluar, sin embargo, tiene restricciones con relación al tiempo de uso, donde una vez que se ha vencido este lapso de prueba, el usuario se ve en la obligación de adquirirlo si quiere seguir usándolo.**

**Es importante mencionar que este tipo de programas se caracteriza en que no son modificables, puesto que no se les puede acceder a su código fuente.**

**FREEWARE**

**Este tipo de software se refiere a todos aquellos programas que son distribuidos totalmente gratis y sin limitaciones de tiempo, sin embargo, no llega a ser libre, puesto que el desarrollador puede llegar a limitar su uso a las empresas, así como también la redistribución por medio de canales no autorizados, solo por mencionar algunos aspectos.**

**El freeware es como una versión mejorada del shareware ¿Por qué? Pues porque puedes utilizar las funciones del programa indefinidamente y de forma gratuita, mientras que con el shareware tendrás una restricción de tiempo para usarlo y posteriormente tendrás que efectuar un pago si deseas continuar con su uso.**

**Al igual que el shareware, generalmente a estos programas tampoco se les puede acceder al código fuente, por lo cual no llegan a ser modificables. Entre los ejemplos más significativos de este tipo de software están el Windows Live Messenger o el Adobe Flash Player.**

# Ciclo de vida del software

1. ¿Qué produce cada etapa y que recibe de la anterior?

-Análisis: se debe comprender de forma detallada el problema produciendo una documentación entendible, completa, fácil de verificar y modificar.

**-Diseño: en esta etapa se deciden que estructura de datos, arquitectura de software, interfaz de usuarios, y procedimientos se van a usar. En esta etapa, recibe la solución del problema de la etapa anterior.**

-Codificación: se codifica todo lo que haya recibido de la etapa anterior.

-**Pruebas: garantizan el correcto funcionamiento del sistema. Se comprueba los criterios de corrección y calidad de la codificación de la etapa anterior.**

-Mantenimiento: se asegura de que el sistema esté adaptado a los cambios. Se pueden dar lugares a futuros errores en el futuro.

1. ¿Qué entendemos por ciclo de vida del software?

Ciclo de Vida del Software como el conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o reemplazado por otro más adecuado.

1. ¿Qué entendemos por desarrollo del software?

*Desarrollar un*[*software*](https://www.ecured.cu/Software)*significa construirlo simplemente mediante su descripción.*

## Modelos de ciclo de vida

1. Investiga en internet cual de los dos modelos evolutivos realiza un análisis de riesgos en cada evolución.
2. ¿Cuál es la fase del ciclo de vida que conlleva más esfuerzo del total del proyecto?

Las mejoras.

1. ¿Qué modelo va elaborando sucesivas versiones hasta llegar al producto final?

Modelo evolutivo.

1. ¿Cuál es la fase del ciclo de vida que conlleva menos esfuerzo del total del proyecto?, ¿por qué crees que es así?
2. ¿En qué modelo se pasa de una etapa a otra sin que sea posible volver hacia atrás?

El modelo en cascada.

1. ¿Qué modelo es aconsejable usar para un proyecto pequeño en el que no habrá variaciones en los requisitos durante el desarrollo del mismo?

El modelo en cascada.

1. ¿Qué modelo es aconsejable usar para un proyecto que tendrá pocos cambios en los requisitos?

El modelo en cascada con Retroalimentación.

1. ¿Qué modelo está basado en el modelo en cascada con retroalimentación?

El Modelo Iterativo Incremental

1. ¿Qué modelos realizan varias versiones del software cada vez más refinadas?

El modelo evolutivo.

1. ¿Qué modelo de ciclo de vida requiere conocer todos los requisitos de antemano?

El modelo en Cascada.

1. ¿Qué diferencia hay entre el modelo en cascada y el de cascada con retroalimentación?

En el modelo en cascada no se puede volver atrás de unas etapas a otras y en el del retroalimentación se puede volver atrás en cualquier momento para corregir, modificar o depurar algún aspecto.

# Fases en el desarrollo y ejecución del software

## Análisis

1. ¿Qué dos tipos de requisitos establece el análisis?

Requisitos funcionales y no funcionales.

1. ¿Qué se establece en la fase de análisis?

Esta es la primera fase del proyecto es la más complicada y la que más depende de la capacidad del analista. Es la fase de mayor importancia en el desarrollo del proyecto y todo lo demás dependerá de lo bien detallada que esté. También es la más complicada, ya que no está automatizada y depende en gran medida del analista que la realice. En esta fase es esencial una buena comunicación entre el cliente y los desarrolladores.

1. ¿En qué técnica las entrevistas están dirigidas a la alta dirección?

En la planificación conjunta de requisitos.

1. ¿Qué se obtiene al terminar la fase de análisis?

Al final de la fase de análisis se obtiene el documento ERS (Especificación de Requisitos Software).¿En qué técnica participan muchos tipos de miembros del proyecto y se apoya en la dinámica de grupos?

1. ¿Qué es un requisito funcional?

Es el servicio que el sistema debe proporcionar. Función a realizar por la aplicación. Qué respuesta dará la aplicación ante todas las entradas. Cómo se comportará la aplicación en situaciones inesperadas.

1. ¿Qué es un requisito no funcional?

Es la restricción que afectará al sistema. Tiempos de respuesta del programa, legislación aplicable, tratamiento ante la simultaneidad de peticiones, etc.

1. ¿Un requisito que pide que el usuario pueda hacer una copia de seguridad de que tipo es?

Requisito funcional.

1. Enumera las fases del desarrollo del software
2. Análisis de requisitos
3. Diseño
4. Codificación
5. Pruebas
6. Documentación.
7. Explotación
8. Mantenimiento
9. ¿Qué representan las entidades externas en un DFD?

Las entidades externas representan componentes que no forman parte del sistema. Que proporcionan datos o los reciben. (rectángulos)

1. ¿En qué técnica de representación de requisitos se representan con rectángulos las entidades y con rombos la relaciones?

Diagramas Entidad / Relación DER. Usado para representar los datos y la forma en la que se relacionan entre ellos. Está formado por entidades (rectángulo) y relaciones (rombo).

1. ¿Qué representan los flujos de datos en un DFD?

El flujo de datos representa el movimiento de datos del sistema (flechas)

1. ¿Qué técnica consiste en hablar con el cliente?

La entrevista.

1. ¿Qué técnicas se utilizan en el análisis?

Entrevistas.

**Desarrollo conjunto de aplicaciones.**

Planificación conjunta de requisitos.

**Brainstorming.**

1. ¿Qué técnica de representación de requisitos representa el flujo de los datos entrelos distintos procesos, entidades externas y almacenes que forman el sistema?

Diagramas de flujo de datos DFD

1. ¿Qué técnica de representación de requisitos describe detalladamente los datos utilizados por el sistema?

Diccionario de datos DD

1. ¿Qué identifican los procesos en un DFD?

Los procesos de identifican funciones dentro del sistema (burbujas ovaladas)

1. ¿Qué técnica de representación de requisitos utiliza escenarios para describir como se usa la aplicación?

Casos de uso

1. ¿Un requisito que pide que la aplicación se pueda usar en dispositivos móviles de que tipo es?
2. ¿En qué técnica se generan ideas desde diferentes puntos de vista?

Brainstorming.

1. ¿Qué representan los almacenes en un DFD?

Los almacenes representan los datos desde el punto de vista estático, lugar donde se almacenan o desde donde se recuperan. (dos líneas horizontales paralelas)

1. ¿Qué técnicas se utilizan para representar los requisitos?

Diagramas de flujo de datos DFD.

**Diagramas Entidad / Relación DER.**

Diccionario de datos DD.

**Prototipos.**

Casos de uso.

**Un modelo de dominio**

1. ¿Qué especifica el documento de especificación de requisitos de software?

La planificación de las reuniones que van a tener lugar.

• Relación de los objetivos del usuario cliente y del sistema.

• Relación de los requisitos funcionales y no funcionales del sistema.

• Relación de objetivos prioritarios y temporización.

• Reconocimiento de requisitos mal planteados o que conllevan contradicciones, etc.

## Diseño

1. ¿Qué se realiza en esta fase?

Diseño: Se divide el sistema en partes y se determina la función de cada una

1. ¿Qué dos tipos de diseños se pueden utilizar?

Diseño estructurado y diseño orientado a objetos.

### Diseño estructurado

1. ¿Qué cuatro niveles tiene el diseño clásico?

Diseño de datos.

**Diseño arquitectónico.**

Diseño de la interfaz.

**Diseño a nivel de componentes (procedimental).**

1. ¿Qué fase del diseño de estructurado diseña las estructuras de datos?

Diseño de datos.

1. ¿Qué fase del diseño de estructurado detalla la estructura de cada componente?

Diseño arquitectónico.

1. ¿Qué fase del diseño de estructurado divide el proyecto en módulos?

Diseño arquitectónico.

1. ¿Qué fase del diseño estructurado crea los formatos de pantalla?

Diseño de la interfaz.

1. ¿Qué fase del diseño de estructurado utiliza diagramas de flujo, pseudocódigo etc?

Diseño a nivel de componentes (procedimental).

1. ¿Qué fase del diseño de estructurado diseña la estructura de componentes de software?

Diseño de datos.

### Diseño orientado a objetos

1. ¿Qué requisito debe cumplir la fase de análisis para poder realizar un diseño orientado a objetos?

Es necesario partir de un análisis orientado a análisis.

1. ¿Qué cuatro capas define el diseño orientado a objetos?

Subsistemas.

**Clases y objetos.**

Mensajes

**Responsabilidades.**

1. ¿Qué es UML?

UML es un Lenguaje de modelado basado en diagramas que sirve para expresar modelos y que se ha convertido en un estándar de las metodologías de desarrollo orientado a objetos.

## Codificación

1. ¿Qué ventajas proporciona el escribir código fácil de leer?
2. ¿Qué característica deseable del código requiere que el código haga lo especificado en los requisitos?
3. ¿Qué implica la programación eficiente?
4. ¿Qué permite un código portable?
5. ¿Qué característica deseable del código lo divide en trozos pequeños?

### Fases en la obtención del código

1. Define código fuente

El código fuente es el conjunto de instrucciones que la computadora deberá realizar, escritas por los programadores en algún lenguaje de alto nivel utilizando un editor de texto.

1. Define algoritmo

Es un conjunto de pasos a seguir para obtener la solución del problema. El algoritmo lo diseñamos en pseudocódigo y con él, la codificación posterior a algún Lenguaje de Programación determinado será más rápida y directa.

1. ¿En qué lenguaje escribimos un código fuente y en que lenguaje debe recibirlo un ordenador?

Se escribe en Lenguaje de alto nivel usando un editor de texto y el ordenador lo recibe en lenguaje máquina.

1. ¿Cuáles son los pasos para obtener el código fuente de una aplicación?
2. ¿Qué es el código objeto?

El código objeto es un código intermedio. Es el resultado de traducir código fuente a un código equivalente formado por unos y ceros que aún no puede ser ejecutado directamente por la computadora. Es decir, es el código resultante de la compilación del código fuente.

1. ¿Mediante que dos procesos se puede obtener un código objeto?

Compilación: El proceso de traducción se realiza sobre todo el código fuente, en un solo paso. Se crea código objeto que habrá que enlazar. El software responsable se llama compilador.

Interpretación: El proceso de traducción del código fuente se realiza línea a línea y se ejecuta simultáneamente. No existe código objeto intermedio. El software responsable se llama intérprete. El proceso de traducción es más lento que en el caso de la compilación, pero es recomendable cuando el programador es inexperto, ya que da la detección de errores es más detallada.

El código objeto es código binario, pero no puede ser ejecutado por la computadora.

1. ¿En qué se diferencian la compilación y la interpretación?
2. ¿Cómo se denomina el proceso de unir todos los archivos de código objeto?

Código ejecutable.

1. ¿Cómo se denomina el software que realiza el proceso de enlazado?
2. ¿Se puede ejecutar el código obtenido del proceso de compilación?
3. ¿Se puede ejecutar el código obtenido del proceso de enlazado?

### Máquinas virtuales

1. ¿Cuál es la función principal de las máquinas virtuales?

Separar el funcionamiento del ordenador de los componentes hardware instalados. Esta capa de software desempeña un papel muy importante en el funcionamiento de los lenguajes de programación, tanto compilados como interpretados.

1. ¿Qué nos permite el uso de una máquina virtual?

Una máquina virtual nos permite desarrollar y ejecutar una aplicación sobre cualquier equipo, independientemente de las características concretas de los componentes físicos instalados. Esto garantiza la portabilidad de las aplicaciones.

1. ¿Cuáles son las funciones principales de una máquina virtual?

Las funciones principales de una máquina virtual son las siguientes:

• Conseguir que las aplicaciones sean portables.

• Reservar memoria para los objetos que se crean y liberar la memoria no utilizada.

• Comunicarse con el sistema donde se instala la aplicación (huésped), para el control de los dispositivos hardware implicados en los procesos.

• Cumplimiento de las normas de seguridad de las aplicaciones.

### Sdg

### Entornos de ejecución

1. ¿Qué es un entorno de ejecución?

Un entorno de ejecución es un servicio de máquina virtual que sirve como base software para la ejecución de programas. En ocasiones pertenece al propio sistema operativo, pero también se puede instalar como software independiente que funcionará por debajo de la aplicación. Es decir, es un conjunto de utilidades que permiten la ejecución de programas.

1. ¿De qué partes consta el entorno de ejecución?

El Entorno de Ejecución está formado por la máquina virtual y los API's (bibliotecas de clases estándar, necesarias para que la aplicación, escrita en algún Lenguaje de Programación pueda ser ejecutada).¿Qué es el runtime?

Es el tiempo que tarda un programa en ejecutarse en la computadora.

1. ¿Cuáles son las funciones de los entornos de ejecución?

* Configurar la memoria principal disponible en el sistema.

• Enlazar los archivos del programa con las bibliotecas existentes y con los subprogramas creados. Considerando que las bibliotecas son el conjunto de subprogramas que sirven para desarrollar o comunicar componentes software pero que ya existen previamente y los subprogramas serán aquellos que hemos creado a propósito para el programa.

• Depurar los programas: comprobar la existencia (o no existencia) de errores semánticos del lenguaje (los sintácticos ya se detectaron en la compilación).

1. ¿Por qué la máquina virtual y las API’s se suelen distribuir juntas?

**Estos dos componentes se suelen distribuir conjuntamente, porque necesitan ser compatibles entre sí.**

1. ¿A quién pertenece el entorno de ejecución?

En ocasiones pertenece al propio sistema operativo, pero también se puede instalar como software independiente que funcionará por debajo de la aplicación. Es decir, es un conjunto de utilidades que permiten la ejecución de programas.

### Java RuntimeEnviroment

1. ¿Qué es el Java RuntimeEnvironment (JRE)?

Se denomina JRE al Java Runtime Environment (entorno en tiempo de ejecución Java). El JRE se compone de un conjunto de utilidades que permitirá la ejecución de programas java sobre cualquier tipo de plataforma.

## Pruebas

1. ¿Las pruebas de caja negra y caja blanca son pruebas de verificación o de validación?

Son pruebas de validación.

1. ¿En qué tipo de pruebas se intenta comprobar si el software implementa correctamente la función para la que está diseñado?

Pruebas de verificación.

1. ¿Qué pruebas se centran en validar los requisitos funcionales sin fijarse en el funcionamiento interno del programa?

Pruebas de validación.

1. ¿Qué dos tipos de pruebas se realizan para comprobar que hemos construido el software de forma correcta?

Pruebas de verificación y pruebas de validación.

1. ¿En qué tipo de pruebas se intenta comprobar si el software cumple los requisitos del cliente?

Pruebas de validación.

1. ¿Qué pruebas se centran en validar la estructura interna del programa?

**Pruebas de caja blanca.**

## Documentación

1. ¿En qué documento queda reflejada la información para la puesta en marcha de la aplicación?

Guía de instalación.

1. ¿En qué documento quedan reflejadas las pruebas realizadas?

Guía Técnica.

1. ¿Cuál es el objetivo de la guía de instalación?

Dar toda la información necesaria para garantizar que la implantación de la aplicación se realice de forma segura, confiable y precisa.

1. ¿En qué documento quedan reflejados ejemplos de uso del programa?

Guía de uso.

1. ¿Cuál es el objetivo de la guía técnica?

Facilitar un correcto desarrollo, realizar correcciones en los programas y permitir un mantenimiento futuro.

1. ¿En qué documento quedan reflejados la solución de posibles problemas que puedan surgir al utilizar la aplicación?

Guía de uso.

1. ¿En qué documento queda reflejada información sobre la seguridad del sistema?

Guía de instalación.

1. ¿A quién va dirigida la guía de instalación?

Al personal informático responsable de la instalación, en colaboración con los usuarios que van a usar la aplicación (clientes).

1. ¿En qué documento queda reflejado la codificación de los programas?

Guía técnica.

1. ¿A quién va dirigida la guía técnica?

Al personal técnico en informática (analistas y programadores).

1. ¿En qué documento queda reflejada la descripción de la funcionalidad de la aplicación?

Guía de uso.

1. ¿En qué documento queda reflejado el diseño de la aplicación?

Guía Técnica.

1. ¿Qué fases de un proyecto es necesario documentar?
2. ¿En qué documento quedan reflejados requisitos software de la aplicación?

Guía de uso.

1. ¿A quién va dirigida la guía de uso?

A los usuarios que van a usar la aplicación (clientes).

1. ¿Cuál es el objetivo de la guía de uso?

Dar a los usuarios finales toda la información necesaria para utilizar la aplicación.

1. ¿En qué documento queda reflejada la forma de comenzar a ejecutar la aplicación?

Guía de instalación.

## Explotación

1. ¿En qué consiste la fase de explotación?

La explotación es la fase en que los usuarios finales conocen la aplicación y comienzan a utilizarla. La explotación es la instalación, puesta a punto y funcionamiento de la aplicación en el equipo final del cliente.

1. ¿En qué consisten las beta test?

Son las últimas pruebas que se realizan en los propios equipos del cliente y bajo cargas normales de trabajo.

1. ¿Una vez se alcanza la fase de explotación del software es necesaria alguna labor por parte del equipo de desarrollo?
2. ¿Quién se encarga de realizar las configuraciones en la fase de configuración?

Los propios usuarios finales.

1. ¿Quién debería realizar la configuración del software si es una aplicación hecha a medida del cliente?

## Mantenimiento

1. ¿Cómo se denomina a los cambios que se realizan en el software para adaptarse a las nuevas tendencias del mercado, a nuevos componentes hardware, etc.?

Adaptativos.

1. ¿Cómo se denomina a los cambios que se realizan en el software para satisfacer nuevas necesidades del cliente?

Evolutivos.

1. La duración de la etapa de mantenimiento es corta. ¿Verdadero o falso?

Falso. Es la más larga de todo el ciclo de vida del software.

1. ¿Cómo se denomina a los cambios que se realizan en el software para mejorar la funcionalidad del software?

Perfectivos.

1. ¿Cómo se denomina a los cambios que se realizan en el software para eliminar errores?

Correctivos.

# Lenguajes de programación

1. Enumera los 4 tipos de lenguajes de programación en orden evolutivo.

Lenguaje máquina.

**Lenguaje ensamblador.**

Lenguaje de alto nivel basado en código.

**Lenguaje visual.**

1. Define lenguaje de programación

Es un idioma creado de forma artificial, formado por un conjunto de símbolos y normas que se aplican sobre un alfabeto para obtener un código, que el hardware de la computadora pueda entender y ejecutar. Nos permite comunicarnos con el hardware del ordenador.

1. ¿Qué tipo de lenguaje de programación utiliza mnemotécnicos?

Lenguaje ensamblador.

1. ¿Qué tipo de lenguaje de programación es único para cada procesador?

Lenguaje máquina.

1. ¿Qué tipo de lenguaje de programación utiliza sentencias y órdenes derivadas del idioma inglés?

Lenguaje de alto nivel basados en código:

1. ¿En qué tipo de lenguaje de programación se programa gráficamente usando el ratón?

Lenguajes visuales

1. ¿Qué tipo de lenguaje de programación entiende directamente el ordenador y no necesita por lo tanto traducción?

Lenguaje máquina.

## Concepto y características

1. ¿Cuáles son los dos tipos de lenguajes de bajo nivel?

Lenguaje ensamblador.

**Lenguaje máquina.**

1. ¿El lenguaje LISP que tipo de lenguaje es según la clasificación de la forma de ejecutarse?

Lenguaje interpretado.

1. ¿Cómo se denomina a los lenguajes que están más próximos al funcionamiento interno del ordenador?
2. ¿Cómo se denomina a los lenguajes que utilizan un enlazador o linker?

Ejecutables.

1. ¿Cómo se denomina a las normas de construcción permitidas de los símbolos del lenguaje en un lenguaje de programación?

Sintaxis.

1. ¿Cómo se denomina a los lenguajes hechos para un objetivo muy concreto?

Lenguajes de propósito específico.

1. ¿Cómo se denomina a los lenguajes diseñados para realizar sistemas operativos o drivers?

Lenguajes de programación de sistemas.

1. ¿De qué tres elementos está formado un lenguaje de programación?

Del alfabeto, la sintaxis y la semántica.

1. ¿Cómo se denominaa los lenguajes que primero pasan por una fase de compilación y luego es interpretado?

Lenguajes mixtos.

1. ¿Cómo se denominaa los lenguajes que indican **que** hay que hacer?

Lengujes declarativos.

1. ¿Cómo se denomina al conjunto de símbolos permitidos por un lenguaje de programación?

El alfabeto.

1. ¿Cómo se denomina a los lenguajes en los que un programa ejecuta las instrucciones del programa de manera directa?

Se les llama lenguajes interpretados.

1. ¿El lenguaje LISP que tipo de lenguaje es según la clasificación del paradigma de programación?

Es de tipo lenguaje declarativo.

1. ¿Cómo se denomina a los lenguajes que están más próximos al razonamiento humano?

Se les llama lenguajes de programación de alto nivel.

1. ¿El lenguaje C que tipo de lenguaje es según la clasificación del paradigma de programación?

Es de tipo lenguaje imperativo.

1. ¿El lenguaje Csound que tipo de lenguaje es según el tipo de problemas que resuelve?

Es de tipo propósito específico.

1. ¿Cómo se denomina al significado de las construcciones para hacer acciones válidas en un lenguaje de programación?

Se le llama semántica.

1. ¿Cómo se denomina a los lenguajes aptos para todo tipo de tareas?

Se les llama de propósito general.

1. ¿El lenguaje C que tipo de lenguaje es según el tipo de problemas que resuelve?

Es de tipo programación de sistemas.

1. ¿Cómo se denomina a los lenguajes diseñados para realizar tareas de control y auxiliares?

Se les llama lenguaje de script.

1. ¿Cómo se denomina a los lenguajes en los que un programa traductor traduce el código del programa (código fuente) en código máquina (código objeto)?

Se le llama lenguaje compilado.

1. ¿Cómo se denominaa los lenguajes que indican **como** hay que hacer la tarea?

Se les llama lenguajes imperativos.

1. ¿El lenguaje JAVA que tipo de lenguaje es según la clasificación del paradigma de programación?

Es de tipo programación orientados a objetos.

1. ¿El lenguaje C que tipo de lenguaje es según la clasificación de la forma de ejecutarse?

Es de tipo compilado.

## Lenguajes de programación estructurados

1. ¿Qué tres tipos de sentencia permite la programación estructurada?

• Sentencias secuenciales.

• Sentencias selectivas (condicionales).

• Sentencias repetitivas (iteraciones o bucles).

1. ¿Cuáles son las ventajas de la programación estructurada?

• Los programas son fáciles de leer, sencillos y rápidos.

• El mantenimiento de los programas es sencillo.

• La estructura del programa es sencilla y clara.

1. ¿Cuáles son las desventajas de la programación estructurada?

• Todo el programa se concentra en un único bloque (si se hace demasiado grande es difícil manejarlo).

• No permite reutilización eficaz de código, ya que todo va "en uno". Es por esto que a la programación estructurada le sustituyó la programación modular, donde los programas se codifican por módulos y bloques, permitiendo mayor funcionalidad.

1. ¿Hacía que evolucionó la programación estructurada?

La Programación estructurada evolucionó hacia la Programación modular, que divide el programa en trozos de código llamados módulos con una funcionalidad concreta, que podrán ser reutilizables

## Lenguajes de programación orientados a objetos

1. ¿El lenguaje Pascal es estructurado u orientado a objetos?

Es estructurado.

1. ¿De qué están compuestos los programas orientados a objetos?

En la P.O.O. (Programación Orientada a Objetos) los programas se componen de objetos independientes entre sí que colaboran para realizar acciones Los objetos son reutilizables para proyectos futuros.

1. ¿El lenguaje Java es estructurado u orientado a objetos?

Orientado a Objetos.

1. ¿Qué ventajas presenta la programación orientada a objetos?

• El código es reutilizable.

• Si hay algún error, es más fácil de localizar y depurar en un objeto que en un programa entero.

1. ¿El lenguaje C es estructurado u orientado a objetos?

Orientado a Objetos.

1. ¿Qué nos permite diferenciar los objetos entre sí en un programa orientado a objetos?
2. ¿Qué nombre recibe una colección de objetos con características similares?
3. ¿Cómo se comunican los objetos con otros produciéndose un cambio de estado de los mismos?
4. ¿El lenguaje C++ es estructurado u orientado a objetos?

# Herramientas de apoyo al desarrollo del software

1. ¿Qué es un IDE?
2. ¿Cada IDE puede funcionar con diferentes lenguajes de programación o está diseñado para un lenguaje en concreto?
3. ¿Qué es un framework?

Un framework es una estructura de ayuda al programador, en base a la cual podemos desarrollar proyectos sin partir desde cero. Se trata de una plataforma software donde están definidos programas soporte, bibliotecas, DEPARTAMENTO DE INFORMÁTICA Y COMUNICACIONES Desarrollo de Aplicaciones Web Módulo de Entornos de Desarrollo Unidad 1: Desarrollo de Software 28 lenguaje interpretado, etc., que ayuda a desarrollar y unir los diferentes módulos o partes de un proyecto.

1. ¿Qué ventajas presenta el uso de un framework?

♣ Desarrollo rápido de software.

♣ Reutilización de partes de código para otras aplicaciones.

♣ Diseño uniforme del software.

♣ Portabilidad de aplicaciones de un computador a otro, ya que los bytecodes que se generan a partir del lenguaje fuente podrán ser ejecutados sobre cualquier máquina virtual.

1. ¿Qué desventajas presenta el uso de un framework?

* Gran dependencia del código respecto al framework utilizado (sin cambiamos de framework, habrá que reescribir gran parte de la aplicación).
* La instalación e implementación del framework en nuestro equipo consume bastantes recursos del sistema.

1. Pon dos ejemplos de frameworks

• .NET es un framework para desarrollar aplicaciones sobre Windows. Ofrece el "Visual Studio .net" que nos da facilidades para construir aplicaciones y su motor es el ".Net framework" que permite ejecutar dichas aplicaciones. Es un componente que se instala sobre el sistema operativo.

• Spring de Java. Son conjuntos de bibliotecas (API's) para el desarrollo y ejecución de aplicaciones.