**Unidad 1**

Contenido

[1 Software y programa. Tipos de software 1](#_Toc527963370)

[2 Relación hardware-software 4](#_Toc527963371)

[3 Licencias de Software 5](#_Toc527963372)

[4 Ciclo de vida del software 7](#_Toc527963373)

[4.1 Modelos de ciclo de vida 8](#_Toc527963374)

[5 Fases en el desarrollo y ejecución del software 9](#_Toc527963375)

[5.1 Análisis 9](#_Toc527963376)

[5.2 Diseño 12](#_Toc527963377)

[5.2.1 Diseño estructurado 12](#_Toc527963378)

[5.2.2 Diseño orientado a objetos 12](#_Toc527963379)

[5.3 Codificación 13](#_Toc527963380)

[5.3.1 Fases en la obtención del código 13](#_Toc527963381)

[5.3.2 Máquinas virtuales 14](#_Toc527963382)

[5.3.3 Sdg 15](#_Toc527963383)

[5.3.4 Entornos de ejecución 15](#_Toc527963384)

[5.3.5 Java RuntimeEnviroment 15](#_Toc527963385)

[5.4 Pruebas 16](#_Toc527963386)

[5.5 Documentación 16](#_Toc527963387)

[5.6 Explotación 18](#_Toc527963388)

[5.7 Mantenimiento 18](#_Toc527963389)

[6 Lenguajes de programación 19](#_Toc527963390)

[6.1 Concepto y características 19](#_Toc527963391)

[6.2 Lenguajes de programación estructurados 21](#_Toc527963392)

[6.3 Lenguajes de programación orientados a objetos 22](#_Toc527963393)

[7 Herramientas de apoyo al desarrollo del software 23](#_Toc527963394)

# Software y programa. Tipos de software

* 1. Preguntas

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?

Aplicaciones informáticas

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

Software de programación

1. ¿Qué tipo de software es un depurador?

De programación

1. ¿Qué tipo de software es un compilador?

De programación

1. ¿De qué dos partes consta un ordenador?

Hardware y 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?

Sistema operativo

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

De Sistemas

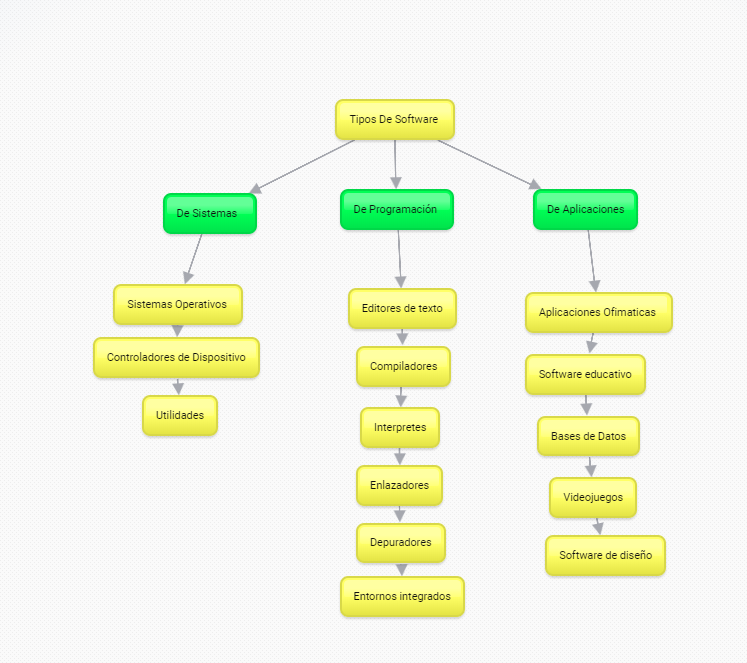
1. ¿Qué tipo de software es una aplicación ofimática?

De Aplicaciones

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

De Sistemas

* 1. Dudas
  2. Mapa conceptual



# 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.

1. Define hardware

Es un conjunto de dispositivos físicos que conforman un ordenador

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

Un computador de programa almacenado es el que almacena las [instrucciones del programa](https://es.wikipedia.org/wiki/Conjunto_de_instrucciones) en la [memoria](https://es.wikipedia.org/wiki/Memoria_de_computadora) electrónica. ​ Con frecuencia la definición es extendida con el requerimiento de que el tratamiento en memoria de datos y programas sea intercambiable o uniforme.

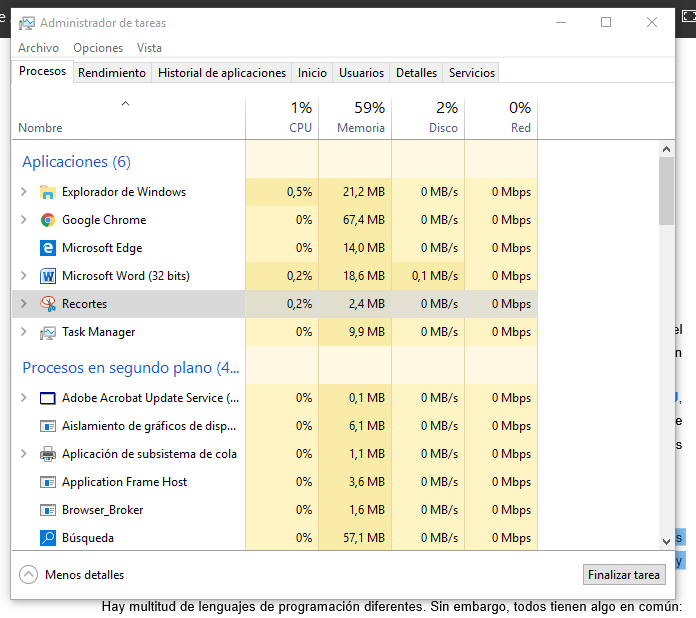
Un computador con una [arquitectura de von Neumann](https://es.wikipedia.org/wiki/Arquitectura_de_von_Neumann) almacena los datos y las instrucciones del programa en la misma memoria. Un computador con una [arquitectura Harvard](https://es.wikipedia.org/wiki/Arquitectura_Harvard) tiene memorias separadas para almacenar los datos y los programas.[5](https://es.wikipedia.org/wiki/Computador_de_programa_almacenado#cite_note-Page2009-5)

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

Ya hemos dicho que una aplicación no es otra cosa que un conjunto de programas, y que éstos están escritos en algún lenguaje de programación que el hardware del equipo debe interpretar y ejecutar.

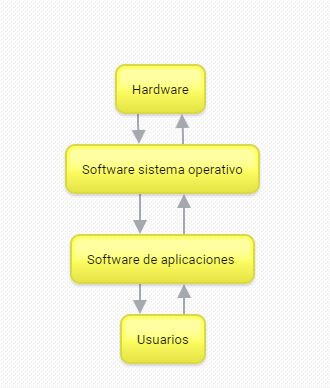
1. Investiga cómo podemos con el “Administrador de tareas” de Windows ver como el sistema operativo asigna recursos a las aplicaciones. (Haz alguna captura donde se vea la asignación).

En más detalles nos lo muestra detalladamente



2.1 Dudas

2.2 Mapa conceptual



# Licencias de Software

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

En que son para la redistribución

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

El desarrollador del software

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

Una patente

1. ¿Qué es el software semi libre?

Aquél que no es libre, pero viene con autorización de usar, copiar, distribuir y modificar para particulares sin fines de lucro

1. ¿Qué es un programa adware?

Subprograma que descarga publicidad sobre otro programa principal.

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

Derecho de autor o copyright es la forma de protección proporcionada por las leyes vigentes en la mayoría de los países para los autores de obras originales

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

En que el semi libre viene con autorización de usar, copiar, distribuir y modificar para particulares sin fines de lucro

1. ¿Qué es el software de dominio público?

Aquél que no está protegido con copyright es decir carece de licencia o no hay forma de determinarla pues se desconoce al autor. .

1. ¿Qué es el software libre?

El software libre se refiere a la Libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el Software

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

Una licencia de software

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

Software propietario

1. ¿El software libre es gratuito?

No tiene por qué ser gratuito, del mismo modo en que el Freeware no tiene por qué ser libre.

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

En que el semi libre viene con autorización de usar, copiar, distribuir y modificar para particulares sin fines de lucro

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

Freeware

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

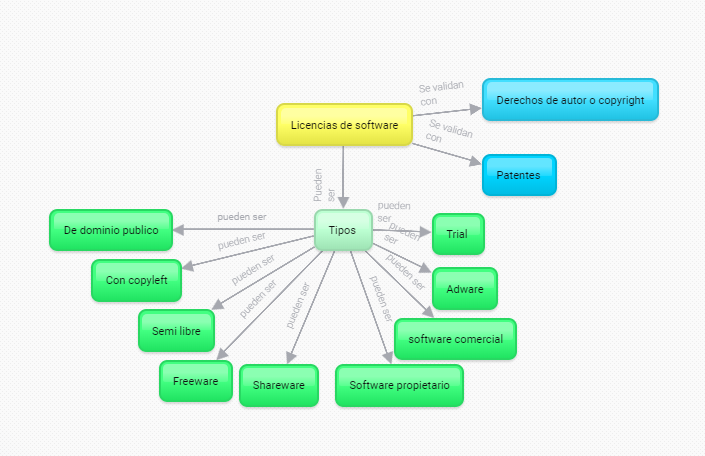
El shareware hay que pagar por su uso

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

El trial es la versión gratuita de un shareware para que puedas probar el programa antes de pagarlo

3.1 Dudas

3.2 Mapa conceptual



# Ciclo de vida del software

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

Varios documentos

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

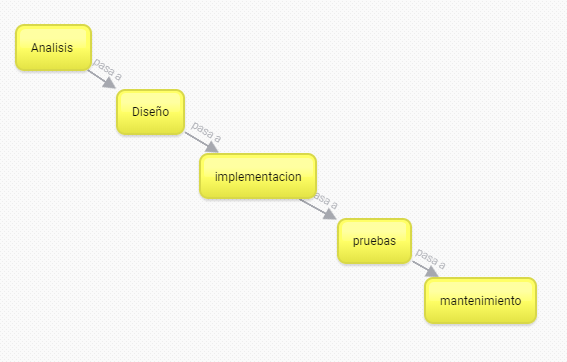
El conjunto de fases por las que pasa el sistema que se está desarrollando desde que nace la idea inicial hasta que el software es retirado o reemplazado por otro más adecuado.

1. ¿Qué similitudes y diferencias hay entre el concepto de desarrollo del software y el de ciclo de vida del software?

El desarrollo es el inicio de la vida del software, es el mismo tiempo pero el ciclo divide el tiempo en etapas

1. ¿Qué entendemos por desarrollo del software?

Todo el proceso que ocurre desde que se concibe una idea hasta que un programa está implementado en el ordenador y funcionando



## Modelos de ciclo de vida

1. Investiga en internet cuál de los dos modelos evolutivos realiza un análisis de riesgos en cada evolución.

Enfoque utilizando prototipos: El objetivo es entender los requisitos del usuario y trabajar para mejorar la calidad de los requisitos. A diferencia del desarrollo exploratorio, se comienza por definir los requisitos que no están claros para el usuario y se utiliza un prototipo para experimentar con ellos. El prototipo ayuda a terminar de definir estos requisitos.

1. ¿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?

Los 2 Modelos evolutivos

1. ¿Cuál es la fase del ciclo de vida que conlleva menos esfuerzo del total del proyecto?, ¿por qué crees que es así?

Implementación, porque una vez echa ya esta

1. ¿En qué modelo se pasa de una etapa a otra sin que sea posible volver hacia atrás?

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?

Modelo en cascada

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

Modelo en cascada con Realimentación

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

Modelo Iterativo Incremental

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

Modelos en cascada

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

Modelo en cascada y iterativo incremental

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

En cascada se introduce una realimentación entre etapas, de forma que podamos 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?

Se especifican los requisitos funcionales y no funcionales del sistema.

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

Estudio de viabilidad. Deducción de requisitos. Análisis de requisitos. Modelado del sistema. Prototipado.

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

Análisis

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

En esta fase se especifican y analizan todos los requisitos del sistema

1. ¿En qué técnica participan muchos tipos de miembros del proyecto y se apoya en la dinámica de grupos?

Desarrollo conjunto de aplicaciones.

1. ¿Qué es un requisito funcional?

Servicios que el sistema debe proporcionar. Qué funciones tendrá que realizar 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?

Restricciones que afectaran 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?

Funcional

1. Enumera las fases del desarrollo del 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.

1. ¿Un requisito que pide que el usuario pueda visualizar en un mapa una ruta en coche de que tipo es?

Funcional

1. ¿Qué representan las entidades externas en un DFD?

Componentes que no forman parte el sistema

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

Entidad Relación

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

Diccionario de datos DD

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

Entrevistas.

1. ¿Qué técnicas se utilizan en el análisis?

Entrevistas.

Brainstorming

Planificación conjunta de requisitos

Desarrollo conjunto de aplicaciones.

1. ¿Qué técnica de representación de requisitos representa el flujo de los datos entre los 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?

Funciones dentro del sistema

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

Un modelo de dominio

1. ¿Un requisito que pide que la aplicación se pueda usar en dispositivos móviles de que tipo es?

Funcional

1. ¿En qué técnica se generan ideas desde diferentes puntos de vista?

Brainstorming

1. ¿Qué representan los almacenes en un DFD?

Representan los datos desde el punto de vista estático, lugar donde se almacenan o desde donde se recuperan

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

Entrevistas

Desarrollo conjunto de aplicaciones

Planificación conjunta de requisitos

Brainstorming

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?

Se traducen los requisitos funcionales y no funcionales en una representación de software.

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

El diseño estructurado basado en el flujo de datos a través del sistema y el diseño orientado a objetos

### Diseño estructurado

1. ¿Qué cuatro niveles tiene el diseño clásico?

Diseño de Datos, Diseño de la interfaz, Diseño arquitectónico, Diseño a nivel de componentes

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

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

Diseño arquitectónico

### Diseño orientado a objetos

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

Es necesario un análisis orientado a objetos (AOO).

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

Subsistema, Clases y objetos, Mensajes, Responsabilidades

1. ¿Qué es UML?

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?

El programador recibe las especificaciones del diseño y las transforma en instrucciones escritas en un lenguaje de programación y almacenadas dentro de un programa.

1. ¿Qué característica deseable del código requiere que el código haga lo especificado en los requisitos?

Corrección

1. ¿Qué implica la programación eficiente?

Modularidad, Corrección, Fácil de leer, Eficiencia, Portabilidad

1. ¿Qué permite un código portable?

Que se pueda implementar en cualquier equipo

1. ¿Qué característica deseable del código lo divide en trozos pequeños?

Modularidad

### Fases en la obtención del código

1. Define 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

Conjunto ordenado de operaciones sistemáticas que permiten hacer cálculos y hallar las soluciones de un tipo de problemas.

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

En código fuente y lo recibe como código objeto

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

1. Se debe partir de las etapas anteriores de análisis y diseño.

2. Se diseñará un algoritmo que simbolice los pasos a seguir para la resolución del problema.

3. Se elegirá una Lenguajes de Programación de alto nivel apropiado para las características del software que se quiere codificar.

4. Se procederá a la codificación del algoritmo antes diseñado.

1. ¿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.

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

Compilación, Interpretación

1. ¿En qué se diferencian la compilación y la interpretación?

En compilación se realiza sobre todo el código y en interpretación se hace línea a línea

Y da la detección de errores más detallada

1. ¿Cómo se denomina el proceso de unir todos los archivos de código objeto?

Ejecutable.

1. ¿Cómo se denomina el software que realiza el proceso de enlazado?

Linker

1. ¿Se puede ejecutar el código obtenido del proceso de compilación?

No, faltaría enlazarlo

1. ¿Se puede ejecutar el código obtenido del proceso de enlazado?

Si

### Máquinas virtuales

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

Es separar el funcionamiento del ordenador de los componentes hardware instalados.

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

Podremos desarrollar y ejecutar una aplicación sobre cualquier equipo

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

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?

es un servicio de máquina virtual que sirve como base software para la ejecución de programas.

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

De un conjunto de utilidades que permiten la ejecución de programas.

1. ¿Qué es el runtime?

El tiempo de ejecución

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.

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?

porque necesitan ser compatibles entre sí.

1. ¿A quién pertenece el entorno de ejecución?

Funciona como intermediario entre el lenguaje fuente y el sistema operativo, y consigue ejecutar aplicaciones.

### Java RuntimeEnviroment

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

(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?

Validación

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

Verificación

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

Pruebas de caja negra

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

Pruebas de caja blanca

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

Pruebas de caja negra

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?

Todas

1. ¿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 USO

## Explotación

1. ¿En qué consiste la fase de explotación?

Es la fase en que los usuarios finales conocen la aplicación y comienzan a utilizarla.

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?

Proporcionar soporte al usuario

1. ¿Quién se encarga de realizar las configuraciones en la fase de configuración?

Los programadores

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

El Programador, el cliente o puede ser automática

## 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

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, ensamblador, de alto nivel, visual

1. Define lenguaje de programación

Son los que nos permiten comunicarnos con el hardware del ordenador.

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

Lenguaje de alto nivel basado en código

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 basado 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?

Lenguajes interpretados:

1. ¿Cómo se denomina a los lenguajes que están más próximos al funcionamiento interno del ordenador?

Lenguajes de script

1. ¿Cómo se denomina a los lenguajes que utilizan un enlazador o linker?

Lenguajes compilados

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

Sintaxis: normas de construcción permitida de los símbolos del lenguaje.

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?

Alfabeto: conjunto de símbolos permitidos.

Sintaxis: normas de construcción permitidas de los símbolos del lenguaje.

Semántica: significado de las construcciones para hacer acciones válidas.

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

Mixtos

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

Lenguajes imperativos

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

Alfabeto: conjunto de símbolos permitidos

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

Lenguajes interpretados

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

Un programa interprete

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

Lenguajes de Programación de Alto nivel

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

Lenguajes compilados:

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

Lenguajes de propósito específico

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

Semántica: significado de las construcciones para hacer acciones válidas.

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

Lenguajes de propósito general

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

Lenguajes de programación de sistemas

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

Lenguajes 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)?

Lenguajes compilados

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

Lenguajes imperativos

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

Lenguajes de Programación Orientados a Objetos

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

Lenguajes de Programación Estructurados

## 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?

Estructurado

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

De Objetos

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?

Estructurados

1. ¿Qué nos permite diferenciar los objetos entre sí en un programa orientado a objetos?

Los atributos

1. ¿Qué nombre recibe una colección de objetos con características similares?

Clase

1. ¿Cómo se comunican los objetos con otros produciéndose un cambio de estado de los mismos?

Métodos

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

Orientado a objetos

# Herramientas de apoyo al desarrollo del software

1. ¿Qué es un IDE?

Entorno integrado de desarrollo

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

Diferentes

1. ¿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.

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 y spring

