# Software y programa. Tipos de software

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

APLICACIONES

1. ¿Qué tipo de software es un videojuego?

SOFTWARE DE APLICACIONES

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

El software de programación

1. ¿Qué tipo de software es un depurador?

software de programacion

1. ¿Qué tipo de software es un compilador?

software de programación

1. ¿De qué dos partes consta un ordenador?

hardware-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?

Software de sistema.

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

Software de aplicaciones

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

Software de sistema.

# Relación hardware-software

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

El sistema operativo (software )es el encargado de coordinar al hardware durante el funcionamiento del ordenador

1. Define hardware

Es el 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)

Propuso la adopción del bit como unidad de medida de la memoria de las computadoras y, además, desarrolló el concepto de los “bits de paridad” para poder paliar la aparición de errores

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

Las aplicación son 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).

Debemos entrar en Administrador de tareas y clicar en detalles después en Rendimiento y con eso podríamos ver el sistema operativo

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

# Licencias de Software

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

Freeware son programas que permiten la redistribución al igual que Shareware que es software con autorización de redistribuir copias ambos se pueden redistribuir.

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?

Patente

1. ¿Qué es el software semi libre?

Es 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?

La forma de protección es para los autores de obras originales incluyendo obras literarias, dramáticas, musicales, artísticas e intelectuales, tanto publicadas como pendientes de publicar

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

El semi libre tiene la autorización de usar, copiar, distribuir y modificar para particulares sin fines de lucro al igual que el software libre.

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?

es un software cuyo código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier finalidad y redistribuido con cambios o mejoras sobre ellas.1

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?

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?

software libre no equivale a gratis, sino que hace referencia a aspectos legales y éticos d e libertades.

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

software libre su código fuente puede ser estudiado, modificado, y utilizado libremente con cualquier finalidad y redistribuido con cambios o mejoras sobre ellas mientras que en el semi libre tiene c 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?

Freeware puedes ser gratuito y el Shareware debes pagar una licencia siempre.

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

La relación que yo he encontrado es que al principio estos software son gratuitos.

# Ciclo de vida del software

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

Análisis: En esta etapa se debe entender y comprender de forma detallada el problema que se va a resolver.

**Diseño: En esta etapa ya sabemos qué es lo que hay que hacer, ahora hay que definir cómo se va a resolver el problema.**

Codificación: En esta etapa se traduce lo descrito en el diseño a una forma legible por la máquina.

**Pruebas: Se comprueba que se cumplen los criterios de corrección y calidad.**

Mantenimiento: Esta fase tiene lugar después de la entrega del software al cliente.

Cada etapa tiene como entrada uno o varios documentos procedentes de las etapas anteriores y produce otros documentos de salida, por ello una tarea importante a realizar en cada etapa es la documentación.

1. ¿Qué entendemos por 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?

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 cual de los dos modelos evolutivos realiza un análisis de riesgos en cada evolución.

Modelo en Espiral; Dentro de cada etapa, tendremos una serie de fases que transcurren desde la planificación, pasando por el análisis de riesgos, el desarrollo y finalizando en la evaluación de lo realizado. Se incorpora también una fase de enlace entre etapas, para facilitar la transición entre las mismas.

1. ¿Cuál es la fase del ciclo de vida que conlleva más esfuerzo del total del proyecto?

El las Mejoras.

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

El modelo espiral.

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

La implementación , porque consiste en la puesta en marcha y ejecución de las acciones previstas en la planificación.

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

Modelo 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 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?

Modelo Iterativo Incremental

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

Modelo en Cascada.

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

La diferencia es que el modelo cascada se pasa de una etapa a otra sin que sea posible volver hacia atrás y el modelo cascada con retroalimentación proviene del modelo anterior

**Pero 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?

requisitos funcionales y no funcionales del sistema.

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

Es la fase del proyecto más complicada, en ella se establece los fundamentos del software

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

Planificación conjunta de requisitos.

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

Al terminar esta fase obtenemos el documento ERS (Especificación de Requisitos Software).

**Donde estará especificado todo lo que será el software.**

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

En la técnica de Desarrollo conjunto de aplicaciones y en la de Brainstorming.

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?

es un requisito funcional

1. Enumera las fases del desarrollo del software

1.Análisis de requisitos

**2. Diseño**

**3. Codificación**

**4. Pruebas**

**5. Documentación**

**6. Explotación**

**7. Mantenimiento**

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

Es un requisito no funcional.

1. ¿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

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

representa el movimiento de datos del sistema (flechas)

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

Entrevistas.

1. ¿Qué técnicas se utilizan en el análisis?

1 Entrevistas.

2 Desarrollo conjunto de aplicaciones.

3 Planificación conjunta de requisitos.

4 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?

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?

Diagramas de flujo de datos DFD.

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. (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?

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 arquitectónico

• Diseño de la interfaz

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

Diseño arquitectónico.

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

Diseño a nivel de componentes (procedimental).

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 a nivel de componentes (procedimental).

### Diseño orientado a objetos

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

Para llevar a cabo un diseño de software orientado a objetos (DOO) es necesario partir de 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?

Modularidad

Corrección

Fácil de leer

Eficiencia

Portabilidad

1. ¿Qué implica la programación eficiente?

La cual usa las especificaciones procedentes del diseño y las transforma a lenguaje de programación y almacenadas dentro de un programa.

1. ¿Qué permite un código portable?

Permite moverse fácilmente de un tipo de máquina a otra, aunque cambien las características internas de la misma.

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

La 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

Es una secuencia de pasos finitos bien definidos que resuelven un problema.

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

En JavaScript. Y el ordenador en binario

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?

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

Interpretación

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

Un compilador genera un programa “stand-alone” es decir que tiene sentido y se puede ejecutar por sí solo, mientras que un programa interpretado siempre necesita su intérprete correspondiente para poder ejecutarse.

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

El código 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

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, 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?

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?

formado por la máquina virtual y los API's (bibliotecas de clases estándar

1. ¿Qué es el runtime?

Se denomina runtime al 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.

• 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?

para que la aplicación, escrita en algún Lenguaje de Programación pueda ser ejecutada

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.

### Java RuntimeEnviroment

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

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?

**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 caja blanca**

**Pruebas de caja negra**

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

**la validación y verificación**

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 negra**

## Documentación

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

GUÍA DE USO

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?

Tenemos que ir documentando el proyecto en todas las fases del mismo para dar toda la información a los usuarios de nuestro software y poder acometer futuras revisiones del proyecto.

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?

fase de producción normal

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

Suelen ser los propios usuarios.

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

Si el software es "a medida", lo más aconsejable es que la hagan aquellos que la han fabricado

## 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:

Lenguaje ensamblador:

Lenguaje de alto nivel basados en código

Lenguajes visuales:

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.

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?

Lenguajes interpretados

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

Lenguajes de Programación de Bajo nivel

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?

Concepto.

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

Sintaxis

Semántica

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

Lenguajes compilados

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

Lenguajes declarativos

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?

Lenguajes interpretados

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 de programación de sistemas

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

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

Lenguajes de propósito genera

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

Lenguajes de propósito general

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 denominaa 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 de sistemas

## Lenguajes de programación estructurados

1. ¿Qué tres tipos de sentencia permite la programación estructurada?
2. ¿Cuáles son las ventajas de la programación estructurada?
3. ¿Cuáles son las desventajas de la programación estructurada?
4. ¿Hacía que evolucionó la programación estructurada?

## Lenguajes de programación orientados a objetos

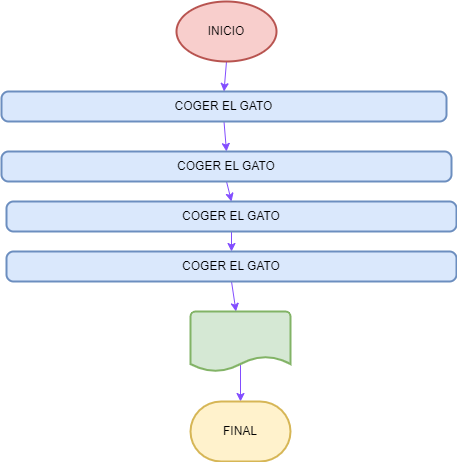
1. ¿El lenguaje Pascal es estructurado u orientado a objetos?
2. ¿De qué están compuestos los programas orientados a objetos?
3. ¿El lenguaje Java es estructurado u orientado a objetos?
4. ¿Qué ventajas presenta la programación orientada a objetos?
5. ¿El lenguaje C es estructurado u orientado a objetos?
6. ¿Qué nos permite diferenciar los objetos entre sí en un programa orientado a objetos?
7. ¿Qué nombre recibe una colección de objetos con características similares?
8. ¿Cómo se comunican los objetos con otros produciéndose un cambio de estado de los mismos?
9. ¿El lenguaje C++ es estructurado u orientado a objetos?

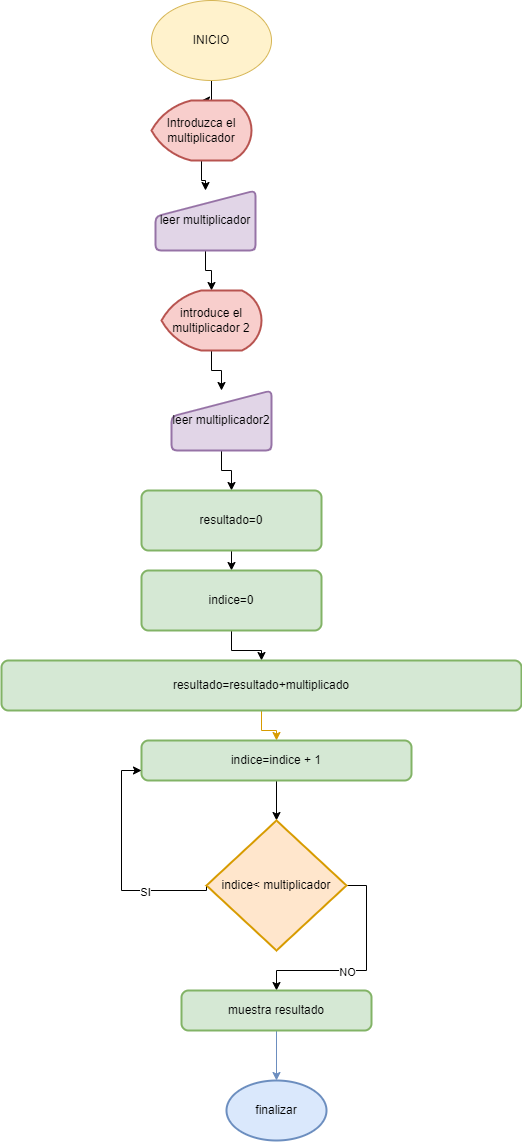
# Herramientas de apoyo al desarrollo del software

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?
4. ¿Qué ventajas presenta el uso de un framework?
5. ¿Qué desventajas presenta el uso de un framework?
6. Pon dos ejemplos de frameworks

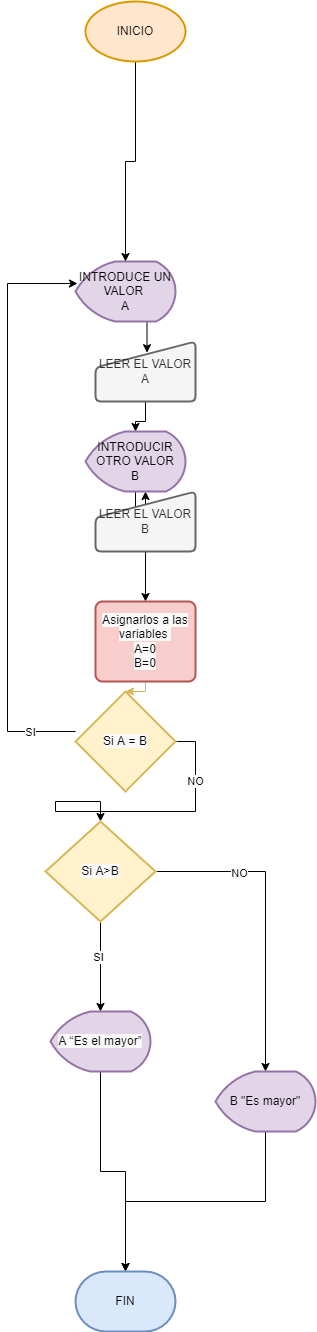
# DIAGRAMA DE FLUJO

PRIMER DIAGRAMA DE FLUJO

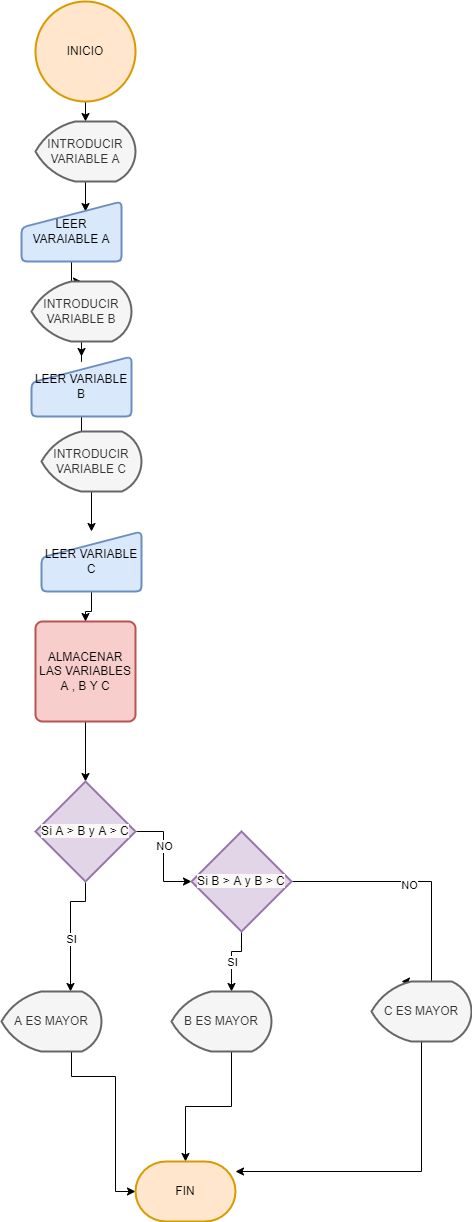




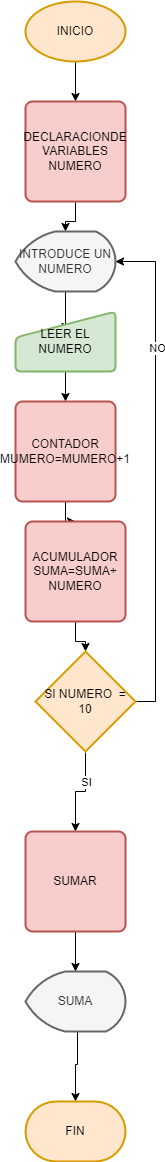
Ejercicio 1 DIAGRAMA DE FLUJO



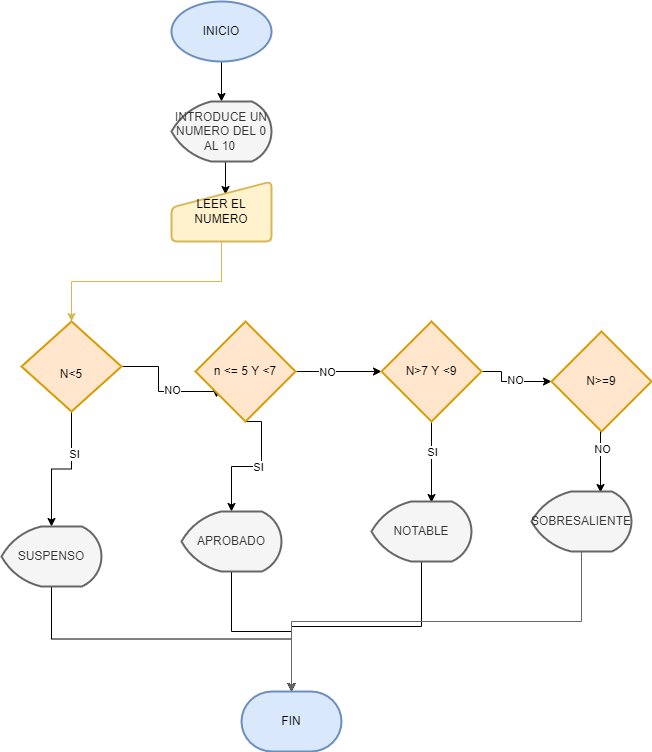
Digrama de flujo A B C



EJERCICIO 3 SUMA

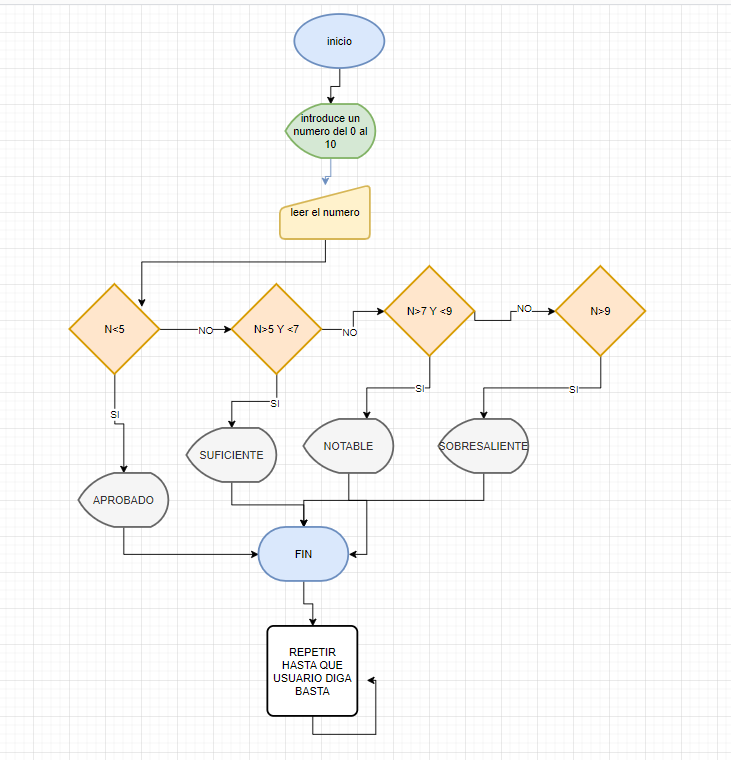


EJERCISIO NOTA 1

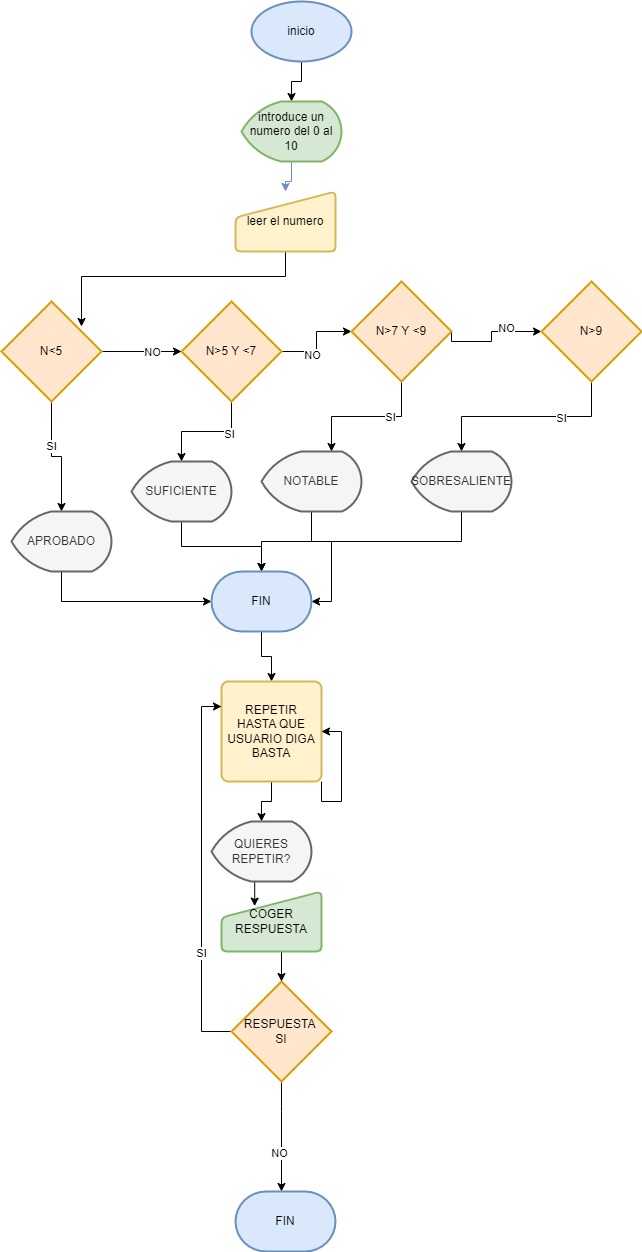


Ejercicio 2

Modifica el diagrama de flujo del ejercicio1 para que se repita el proceso anterior el hasta que el usuario indique que no se quieren introducir más notas (el número mínimo de notas que se introducirán será 0)



Ejercicio 3 Modifica el diagrama de flujo del ejercicio1 para que se repita el proceso anterior el hasta que el usuario indique que no se quieren introducir más notas (el número mínimo de notas que se introducirán será una)



Ejercicio 4

Modifica el diagrama de flujo del ejercicio1 para que se repita el proceso anterior el número de veces que introduzca el usuario. Es decir se preguntará cuantas notas se quieren introducir y se repetirá el proceso del ejercicio anterior ese número de veces.

