
De: José Carlos Manjón Carrasco

Enviado: lunes, 17 de octubre de 2022 17:27

Para: José Carlos Manjón Carrasco

Asunto: entornos de desarrollo

1. Según la cercanía que esté del lenguaje humano, ¿sería adecuado utilizar un lenguaje de programación de bajo nivel? Justifica tu respuesta. Según la técnica de programación utilizada, ¿sería adecuado utilizar un lenguaje de programación estructurado? Justifica tu respuesta. En ambos casos, ¿cuáles serían las elecciones correctas?

Un lenguaje de bajo nivel es aquel que está más cercano al cómo funciona internamente un ordenador, se podrían algún lenguaje ensamblador o lenguaje máquina, esta opción aunque podría ser válida, no es recomendable, sobre todo por las inconveniencias que plantearía, como podría ser encontrar al autor del código, estos lenguajes no son muy populares, es único para cada procesador, por lo que si hubiera ordenadores con diversos procesadores, se multiplicaría el trabajo, otro inconveniente es que el lenguaje ensamblador en el código se hace referencia a la ubicación física de los archivos, por lo que no se podría reutilizar código, multiplicando el trabajo a realizar y, por tanto, el coste, por lo que en mi opinión se debería usar un lenguaje de alto nivel o visual, ya que los inconvenientes anteriores los tienen eliminados.

¿Sería adecuado utilizar un lenguaje de programación estructurado? En mi opinión, no sería adecuado, porque en un lenguaje estructurado el código está en un único bloque, complicando la programación y el mantenimiento, como sería el caso de la notaría, al ser un desarrollo con bastantes retos.

La elección correcta, sin entrar al lenguaje concreto, recomendaría un lenguaje de alto nivel o visual, ya que sería más fácil el desarrollo y mantenimiento, puesto que son tipos de lenguaje más populares, con uno de esos lenguajes evitaríamos el realizar trabajo múltiple y, por tanto, el coste al ser independiente del procesador y poder usar un mismo código para diversos ordenadores, eso sí, debemos tener en cuenta el sistema operativo sobre el que nos vayamos a basar aunque Windows es el más popular, podríamos perfectamente basarnos en Linux o Apple y esto nos puede plantear diversos retos.

2. El software que se va a instalar en los equipos de la notaría, ¿sería código objeto? Justifica tu respuesta. ¿Cuál sería el tipo de código correcto?

Un código abierto es aquel que está disponible para que cualquier usuario con conocimientos puede modificarlo a su antojo, teniendo en cuenta la naturaleza del proyecto, que es un programa para una notaría (empresa privada), y la naturaleza del trabajo, como podrían ser contratos, herencias..., lo referente a seguridad es de vital importancia, por lo que solo ciertas personas con conocimientos y debidamente autorizados podrían tocar el código del programa, ya que de no ser así, esto podría crear unas brechas de seguridad muy importantes, constitutivas incluso de delito

Es código objeto, no código abierto.

3. ¿Qué tipo de lenguaje deberíamos utilizar si necesitáramos independencia de nuestro software con el hardware concreto del equipo? Justifica tu respuesta. Pon un ejemplo concreto del lenguaje de programación.

Ver apuntes, se trataría de un lenguaje compilado, que genera un archivo a un código objeto, en un lenguaje que un intérprete se encargaría a traducir a lenguaje máquina, con máquinas virtuales, es independiente del hardware y sistema operativo, 5,3 del tema PDF profesora; just in time programa; ;
en los apuntes los llama lenguaje compilado

4. Previa instalación de nuestra aplicación en los equipos de la notaría, ¿qué requisitos hardware deben cumplir estos equipos? ¿Qué software deberemos tener instalado previamente?
5. Para el proceso de traducción de código fuente a código objeto, ¿usaremos un compilador o un intérprete? Justifica tu respuesta.

rerere

El proceso de traducción de código fuente a código objeto puede realizarse de dos formas:

Compilación: El proceso de traducción se realiza sobre la totalidad del 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.

6. Realiza en ejemplo de documento ERS, según las especificaciones dadas por la notaría. ¿En qué fase del desarrollo se encuentra este documento?
7. ¿Qué ventajas e inconvenientes tendría la utilización de un framework para desarrollar el proyecto solicitado? ¿Qué ventajas e inconvenientes tendría la utilización de un entorno de desarrollo para desarrollar el proyecto solicitado? ¿Cuál de las anteriores opciones utilizarías? Justifica tu respuesta.