

BOLETÍN TEÓRICO

By Miladys V. Barreto Lara

Ourense, Octubre de 2020



CFP A Carballeira
Entornos de Desarrollo

Tabla de contenido

Introducción	3
1. Nombres comerciales de software de sistemas y de aplicación.	4
2. Aplicaciones disponibles para el usuario en versión alfa, beta o RC o RTM.	4
3. Últimas versiones de Debian, Ubuntu, Gentoo, NetBeans, Eclipse, Java y Windows Server.	4
4. fases típicas del proceso de desarrollo de SW.	5
5. Ventajas y desventajas del modelo en cascada.	5
Ventajas.	5
Desventajas.	5
6. Manifiesto para el desarrollo ágil.	6
7. Historial de usuario. Metodología en la cual se aplica.	6
8. Metodología Scrum. Principales pautas.	6
9. Metodología Kanban. Principales pautas.	6
10. Terminología Scrum. Algunos términos que se emplean como sinónimos.	7
11. Diferencia existe entre los lenguajes declarativos y los imperativos.	7
12. ¿Qué es compilar? ¿Qué es interpretar?	7
13. Ventajas de los lenguajes compilados.	8
14. Ventajas de los lenguajes interpretados.	8
15. Ejemplo de 2 lenguajes compilados y 2 interpretados.	8
16. ¿Puede considerarse código objeto el bytecode generado en Java tras la compilación? Explique.	9
17. Ejemplos de lenguaje de Bajo, Medio y Alto nivel.	9
18. Paradigma de programación de algunos lenguajes.	9
19. Definición de conceptos.	10
Conclusión	11

Introducción

A lo largo del documento iremos respondiendo a una serie de cuestiones básicas relacionadas al desarrollo de software, con lo cual se pretende que el estudiante aprenda a:

- Identificar los tipos de software.
- Identificar y reconocer las características principales de los modelos mas destacados empleados en el desarrollo de software.
- Reconocer las diferencias entre los diversos tipos de lenguaje de programación.
- Familiarizarse con la terminología usada en el proceso de generación de código.

1. Nombres comerciales de software de sistemas y de aplicación.

- Software de sistemas: Windows, Mac OS, Solaris, Android.
- Software de aplicación: Notepad++, NetBeans, (horizontal: Microsoft Word, Power Point, Excel, Numbers).

2. Aplicaciones disponibles para el usuario en versión alfa, beta o RC o RTM.

Opera beta 36.0.2130.29, Google Chrome – RC 1.5.849, SQL Server 2016 RTM - KB4037357

Fuentes consultadas: <https://opera-beta.uptodown.com/windows>

<https://chrome-beta.uptodown.com/android>

<https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

3. Últimas versiones de Debian, Ubuntu, Gentoo, NetBeans, Eclipse, Java y Windows Server.

Debian 10.6, Ubuntu 20.10, Getnoo (Sep15,2020), NetBeans 12.1(LTS), Eclipse IDE 2020-09, Java15, Windows Server 2019.

Fuentes consultadas:

<https://www.debian.org/index.es.html>

<https://ubuntu.com/download>

<https://www.gentoo.org/>

<https://netbeans.org/>

<https://www.eclipse.org/downloads/>

<https://www.oracle.com/java/technologies/javase/15-relnote-issues.html>

<https://www.microsoft.com/es-es/sql-server/sql-server-downloads>

4. fases típicas del proceso de desarrollo de SW.

Análisis, diseño, codificación, pruebas, instalación y mantenimiento, **la documentación** es un aspecto implícito en todas las etapas.

5. Ventajas y desventajas del modelo en cascada.

Ventajas.

- La fase de documentación resulta en información extra y mas accesible a los programadores, parte de la misma esta dirigida al usuario lo cual facilita la comprensión del software.
- La fase de pruebas es incluyente pues se recomienda que en ella participen especialistas externos y usuarios.

Desventajas.

- Al ser un modelo lineal de haber fallas en una fase se irían acumulando en las siguientes.
- Debido a que el cliente no puede ver el software ejecutándose hasta las últimas etapas, se necesitaría mucho trabajo para corregir un error que se detecta en las últimas fases, pero afectando a las primeras.

6. Manifiesto para el desarrollo ágil.

Resumen de los principios sobre los que se basan los **métodos alternativos de desarrollo de software** en cuatro postulados, en estos se valora a los individuos e interacciones, a la colaboración con el cliente y al software en función y cambiante.

Fuentes consultadas: https://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil

7. Historial de usuario. Metodología en la cual se aplica.

Es una representación de un requisito escrito en una o dos frases utilizando el lenguaje común del usuario. Permiten responder rápidamente a los requisitos cambiantes.

Son utilizadas en las metodologías de desarrollo ágiles.

Fuentes consultadas: https://es.wikipedia.org/wiki/Historias_de_usuario

8. Metodología Scrum. Principales pautas.

Metodología de desarrollo ágil.

Un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas) Cada iteración tiene que proporcionar un incremento de producto final que pueda ser entregado al cliente.

9. Metodología Kanban. Principales pautas.

Metodología de desarrollo ágil.

Prioriza la visualización del flujo de trabajo por demanda real, por lo cual limita el número de tareas en curso, las tareas únicamente se arrastran cuando el equipo tiene capacidad para asumir y se dividen de manera eficiente.

10. Terminología Scrum. Algunos términos que se emplean como sinónimos.

- a. Jefe de proyecto: Product owner.
- b. Cliente: Stake holder.
- c. Equipo de desarrollo: Development team.

11. Diferencia existe entre los lenguajes declarativos y los imperativos.

En la programación declarativa:

- Se escribe directamente el resultado final deseado.
- Código fuente muy abstracto en relación al procedimiento concreto por lo que los programas se escriben más rápido y las aplicaciones se optimizan de forma muy sencilla.

En cambio, en la programación imperativa:

- Se emplean instrucciones paso a paso redactadas para el ordenador.
- Son muy concretos y trabajan cerca del sistema. De esta forma, el código es, por un lado, fácilmente comprensible, pero, por el otro, requiere muchas líneas de texto fuente para describir lo que en los lenguajes de la programación declarativa se consigue con solo una parte de las instrucciones.

12. ¿Qué es compilar? ¿Qué es interpretar?

Compilar es traducir el código fuente a código de máquina creando el archivo ejecutable en tiempo de compilación, esto a través de un compilador o programa. Interpretar consiste en traducir el código fuente en lenguaje de máquina en tiempo de ejecución a través de un intérprete o programa en memoria.

13. Ventajas de los lenguajes compilados.

- El proceso de traducción se hace en tiempo de compilación y una sola vez.
- La ejecución es muy rápida ya que el ejecutable está en lenguaje máquina.
- El usuario que ejecuta no tiene que conocer el código fuente, solo tiene el código ejecutable que no es manipulable fácilmente para obtener el código fuente, y como consecuencia el programador tiene el código fuente más protegido.

14. Ventajas de los lenguajes interpretados.

- No existe un archivo ejecutable.
- Interrumpir la ejecución sólo afecta normalmente al intérprete y no a la plataforma.
- Puede ejecutarse en cualquier plataforma siempre que exista su intérprete.
- La modificación del código fuente no requiere ninguna operación extra antes de ejecutar el programa.

15. Ejemplo de 2 lenguajes compilados y 2 interpretados.

- Compilados: Java, C++.
- Interpretados: Ruby, Python.

16. ¿Puede considerarse código objeto el bytecode generado en Java tras la compilación? Explique.

Sí, este código intermedio sería considerado objeto porque es resultado de un proceso de compilación. Sin embargo, no es ejecutable por sí mismo pues necesitaría de una librería.

17. Ejemplos de lenguaje de Bajo, Medio y Alto nivel.

- Bajo nivel: Lenguaje ensamblador.
- Nivel medio: C.
- Alto nivel: Java, PHP, Python.

18. Paradigma de programación de algunos lenguajes.

1. C : Programación imperativa.
2. C++: Programación imperativa.
3. SQL: Programación declarativa.
4. Java: Programación imperativa.
5. JS: Programación imperativa.
6. Lisp: Programación declarativa.

7. Prolog: Programación declarativa.

Fuentes consultadas: https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n

19. Definición de conceptos.

Código fuente: Código escrito en un lenguaje de programación simbólico mediante una herramienta de edición. Este código se guarda en un archivo conocido como archivo fuente y tendrá que traducirse a lenguaje máquina para poder ejecutarse. La traducción puede hacerse mediante compiladores y/o intérpretes.

Código objeto: Conjunto de instrucciones y datos escritos en un lenguaje que entiende el ordenador directamente: binario o código máquina. Proviene de la traducción de cierto código fuente, es un fragmento del programa final y es específico de la plataforma de ejecución

Código ejecutable: Lenguaje creado por máquina que se corresponde a las unidades de programas, donde el ordenador realiza las instrucciones compiladas por medio del enlazador de librerías y el compilador.

Conclusión

Los programas informáticos son imprescindibles para que los ordenadores funcionen, de allí la utilidad de los diferentes tipos de software. En cuanto al desarrollo de este último, el concepto de metodologías ágiles ha tenido una implementación a escala mundial, debido a que suple las deficiencias del modelo tradicional y vincula la evaluación permanente del trabajo realizado por parte de todos los empleados, previniendo así de manera más efectiva el malgasto de tiempo, dinero y recurso en un proyecto.

En la misma línea de tiempo es factible indicar que el lenguaje de programación señalado como el más utilizado en los diferentes portales web es Java, es decir, un lenguaje compilado.