Actividades – Entornos de desarrollo – Elementos de

desarrollo de software **Apuntes**

1. Elabora los apuntes de esta Unidad y guárdalos en GitHub en un repositorio

llamado 'Apuntes\_ED'.

**Introducción**

1. Haz una clasificación del software.

- El software se puede clasificar en 3 apartados.

* Software de sistema: Este tipo de software incluye al sistema operativo o los drivers.
* Software de aplicación: Incluye a las suites ofimáticas como Microsoft office o Libre Office, Navegadores como Google Chrome o Firefox o software de edición de imagen, como Photoshop o gimp.
* Software de desarrollo: Estos incluyen compiladores, editores e intérpretes.

2. Describe la relación que existe entre los componentes hardware principales de

un computador y el almacenamiento y ejecución del software.

* **Disco duro:**

Almacena de forma permanente los archivos ejecutables y de datos.

* **RAM:**

Almacena de forma temporal el código binario de los archivos ejecutables y los archivos de datos necesario.

* **CPU:**

Lee y ejecuta las instrucciones almacenadas en la memoria RAM, asi como los datos necesarios.

* **Dispositivos E/S:**

Recoge nuevos datos desde la entrada, se muestran los resultados, se leen/guardan en un disco duro, etc.

3. Define los siguientes conceptos:

o Código fuente: Conjunto de líneas de texto que indican que pasos ha de seguir una computadora para ejecutar un programa.

o Código objeto: Este tipo de código es el resultado de la compilación del código fuente, después de compilar el código fuente este pasa a estar escrito en binario o código máquina.

o Código ejecutable: Después de obtener el código objeto un “linker” se encarga de juntar todos los archivos de código objeto en un programa ejecutable. Generalmente se les conoce como archivos “.exe” en Windows o “.com” en Linux.

**Ciclo de vida del software**

1. Define "Ciclo de vida del software".

Proceso que se sigue para construir, entregar y hacer evolucionar el software desde la concepción de una idea hasta la entrega y el retiro del sistema, para garantizar que el software cumple con los requisitos se requiere que este pase por una serie de fases intermedias.

2. Nombra las fases principales del desarrollo de software y explica brevemente

que se hace en cada una de ellas.

* **Análisis:** Se contempla lo que nuestro cliente nos está pidiendo, el análisis ha de ser completo y sin omisiones, conciso y sin trivialidades, con pocos o ningún detalle de diseño, hay que separar los requisitos funcionales y los que no lo son, fijar criterios de validación y por último dividir y jerarquizar el modelo.
* **Diseño:** Se descompone y organiza el sistema de en elementos que se pueden desarrollar individualmente. Hay que ser especifico con la interrelación y funcionalidad de los elementos componentes.
* Las actividades habituales son las siguientes:
  + Diseño Arquitectónico.
  + Diseño detallado.
  + Diseño de datos.
  + Diseño de interfaz
* **Codificación:** Se escribe el código fuente de cada componente. Aquí hay que diferenciar entre el backend y el frontend, el primero es el código fuente por ejemplo y se usan lenguajes como C, C++, Python, etc. El Segundo es la parte visible y con la que puedes interactuar, se usan lenguajes como HTML o XML.
* **Pruebas:** Su principal objetivo debe ser conseguir que el programa funcione incorrectamente y corregir los problemas que encontremos. Hay que intentar hacer el número máximo de pruebas.
* **Documentación:** Es necesario documentar y explicar todo lo que podamos para que si en un futuro retomamos el programa podamos saber qué hace cada línea de código.
* **Mantenimiento:** Hay diferentes tipos de mantenimiento:
  + **Correctivo:** Se corrigen defectos.
  + **Perfectivo:** Se mejora la funcionalidad.
  + **Evolutivo:** Se añaden funcionalidades nuevas.
  + **Adaptativo:** Se adapta a nuevos entornos.

3. Explica brevemente en qué consiste el modelo en cascada cuando hablamos de

desarrollo de software.

* Es una manera de trabajar que consiste en terminar una de las fases del proyecto para poder pasar a la siguiente.

4. Ventajas e inconvenientes del modelo en cascada.

* Al ser un modelo de trabajo lineal es el más fácil de implementar entre los trabajadores, la cantidad de recursos que hay que invertir son mínimos. Después de cada etapa se realizan pruebas para comprobar el funcionamiento del código.
* Una de sus principales desventajas es que no se puede volver atrás, por ejemplo, si la fase de diseño ha sido mala, la fase de ejecución va a ser más difícil de desarrollar. Un cambio en el software puede ocasionar varios problemas.

5. ¿Qué se entiende por verificación? ¿Y por validación?

* **Validación:** La validación consiste en comprobar que hemos completado todos los objetivos que teníamos marcados.
* **Verificación:** La verificación es comprobar que cada una de las fases funciona correctamente

6. Explica cómo funciona el modelo de desarrollo mediante creación de prototipos.

Lo primero se crea un plan de trabajo que le parezca bien al cliente y al programador, no se pasa de una fase a otra hasta que se alcancen los requisitos y estos sean testeados. A continuación, pasamos a la fase de construcción y después del desarrollo, se presenta el prototipo y se ve si se desecha (como prototipo rapido) o si lo usamos como referencia para un proyecto.

7. Explica cómo funciona el modelo espiral cuando se aplica al desarrollo

orientado a objetos.

8. ¿Qué cuatro principios rigen el desarrollo ágil expresados en el Manifiesto Ágil?

9. ¿Qué es una **historia de usuario**? Consulta el siguiente enlace:

o https://es.wikipedia.org/wiki/Historias\_de\_usuario

10. Haz un resumen sobre que se entiende por **Lean software** y qué principios lo

rigen. Consulta el siguiente enlace:

o https://es.wikipedia.org/wiki/Lean\_software\_development

11. KANBAN. Estudia las ventajas e inconvenientes de tener una pizarra web digital

para la metodología Kanban. Puedes consultar los siguientes enlaces:

o https://leankit.com/learn/kanban/kanban-board/ o https://trello.com/es o https://taiga.io/ o https://kanbantool.com/es/

1. KANBAN. Haz un resumen de la metodología Kanban e indica sus diferencias

frente a SCRUM. Puedes consultar el siguiente enlace:

o https://es.atlassian.com/agile/kanban

2. SCRUM. Explica como funciona Scrum. Consulta los siguientes enlaces:

o https://proyectosagiles.org/que-es-scrum/ o https://proyectosagiles.org/como-funciona-scrum/

3. SCRUM. Define los siguientes términos:

o Product backlog. o Sprint backlog.

4. SCRUM. En la terminología Scrum qué terminos se utilizan como sinónimo de:

o Jefe de proyecto. o Cliente. o Equipo de desarrollo.

5. SCRUM. Haz un resumen de los requisitos para poder utilizar Scrum. Consulta

el siguiente enlace:

o https://proyectosagiles.org/requisitos-de-scrum/

6. XP. Explica los 5 valores de la Programación Extrema.

7. XP. ¿Cuáles son las características distintivas de XP frente a otras metodologías

ágiles? Explícalas. Puedes consultar el siguiente enlace:

o http://www.davidvalverde.com/blog/introduccion-a-la-programacion-

extrema-xp/

**Lenguajes de programación**

1. ¿Qué diferencia existe entre los lenguajes declarativos y los imperativos?.

Nombra al menos 2 de cada tipo.

2. ¿Explica qué es compilar? ¿Explica qué es interpretar?

3. Ventajas de los lenguajes compilados.

4. Ventajas de los lenguajes interpretados.

5. Nombra 2 lenguajes compilados y otros 2 interpretados.

6. ¿Puede considerarse código objeto el **bytecode** generado en Java tras la

compilación? Explica la respuesta.

7. Pon un ejemplo de lenguaje de los siguientes tipos:

o Bajo nivel. o Nivel medio. o Alto nivel.

8. ¿Qué paradigma de programación siguen los siguientes lenguajes?

o C o C++ o SQL o Java o Javascript o Lisp o Prolog

Puedes consultar el siguiente enlace:

o https://es.wikipedia.org/wiki/Paradigma\_de\_programaci%C3%B3n

9. Explica qué criterios pueden seguirse a la hora de elegir un lenguade de

programación para el desarrollo software.