**Autoevaluación UT1.**

1. ¿Cuál es el único lenguaje que entiende directamente el ordenador?
2. Fuente.
3. **Máquina.**
4. Ensamblador.
5. Objeto.
6. ¿Qué es un programa?
7. Es un lenguaje de programación.
8. **Es un conjunto de instrucciones secuenciadas que realizan una función determinada.**
9. Es el hardware del ordenador.
10. Ninguna de las anteriores.
11. ¿Qué es un proceso?
12. **Es un programa en ejecución.**
13. Es el lenguaje con el que se desarrolla un programa.
14. Es el lenguaje virtual con el que se programa.
15. Ninguna de las anteriores.
16. Relaciona el tipo de lenguaje de programación con sus características según en la forma en que son ejecutados.
17. Lenguajes compilados. No se genera código objeto, se lee cada

instrucción, se interpreta y se ejecuta. 2

1. Lenguajes interpretados. Necesita un programa traductor para convertir

el código fuente en código máquina. 1

1. Lenguajes virtuales. Es un lenguaje más portable porque se genera

un código intermedio o bytecode interpretado

por una máquina virtual en cualquier máquina. 3

1. Relaciona los tipos de lenguajes de programación con sus características.
2. Lenguaje máquina. Es difícil de programar y trabaja con los

registros del procesador y direcciones físicas. 2

1. Lenguaje ensamblador. Tiene una forma de programar más intuitiva y

sencilla. 3

1. Lenguaje de alto nivel. Difiere para cada procesador y fue el primer

lenguaje utilizado. 1

1. Relaciona los siguientes lenguajes de programación de alto nivel con sus características.
2. Java Lenguaje virtual interpretado y orientado a objetos. 1,2
3. Python Lenguaje interpretado, multiplataforma, al principio

orientado para webs dinámicas y con buena integración con MySQL. 3

1. PHP Lenguaje de scripting que se ejecuta en el lado cliente4
2. JavaScript Lenguaje compilado y estructurado que combina

comandos de alto nivel con los de bajo nivel. 5

1. C y C++
2. ¿Cuáles son las etapas a seguir en el Desarrollo de Software?
3. Análisis, Planificación, Pruebas y Documentación.
4. Análisis, Diseño, Codificación, Pruebas, Documentación, Explotación y Mantenimiento.
5. Análisis, Codificación, Reestructuración, Pruebas, Documentación y Mantenimiento.
6. Ninguna de las anteriores.
7. Tener deficiencias en la fase de \_\_\_\_\_\_\_ es la principal causa del gran porcentaje de fracasos de los proyectos software.
8. Diseño
9. Codificación
10. Documentación
11. Análisis

1. Relaciona cada fase del desarrollo de software con lo que se hace en cada una de ellas.
2. Análisis Se implementa el software en un lenguaje de

programación.

1. Diseño. Se realizan los documentos asociados a cada

fase del desarrollo del software.

1. Codificación. Se instala el software en el entorno real.
2. Pruebas Se realizan los procedimientos correctivos y

actualizaciones de software.

1. Explotación. Se determina la arquitectura de la aplicación.
2. Mantenimiento. Se realizan las pruebas para verificar que la

aplicación realice lo que se espera.

1. Documentación. Se analiza el problema a resolver y se toman

los requisitos del cliente.

1. ¿Qué componente es vital para unir archivos en la generación del ejecutable?
2. Compilador.
3. Linker.
4. Ensamblador.
5. Intérprete.
6. En cualquier aplicación se deben generar los siguientes documentos (de usuario, técnico y de instalación). Relaciona cada uno de ellos para lo que sirve.
7. Manual de usuario. Realizado para que cualquier técnico conozca

el lenguaje con el que la aplicación ha sido desarrollada.

1. Manual técnico. Se especifica cómo se instala y se pone en

funcionamiento de la aplicación.

1. Manual de instalación. El manual que usará el usuario para manejar la

desarrollada.

1. Los roles que forman parte del desarrollo de software son los siguientes. Relaciónalos con sus funciones.
2. Arquitecto de Software. Realiza funciones de análisis y también codifica.
3. Jefe de Proyecto. Conoce en profundidad el lenguaje de programación

y realiza las tareas encomendadas por el analista-programador o analista.

1. Analista de sistemas. Dirige el proyecto.
2. Analista programador. Decide cómo va a realizarse el proyecto y cómo se

se cohesiona.

1. Programador. Realiza el análisis del problema y el diseño de todo el

Sistema.

1. Si queremos realizar una aplicación que tendrá grandes cambios a lo largo de su vida. ¿Qué modelo del ciclo de vida sería más recomendable?

* Modelo en cascada.
* Modelo en cascada con realimentación.
* Modelo evolutivo.

1. ¿Qué tipo de herramienta CASE ofrece ayuda en la fase de diseño y análisis de software:

* U-CASE.
* M-CASE.
* L-CASE.

1. De todos los tipos de software, \_\_\_\_\_\_\_\_ es el encargado de gestionar los recursos hardware de un computador.
2. La aplicación.
3. El compilador.
4. El código objeto.
5. El sistema operativo.
6. El modelo en cascada con realimentación es:
7. El modelo perfecto si los requisitos están claros y el proyecto es rígido.
8. El modelo idóneo si se prevén muchos cambios durante el desarrollo.
9. Es el modelo de vida clásico del software.
10. Es un tipo de modelo evolutivo.

1. Entre las ventajas de utilizar Frameworks tenemos:
   1. Poca dependencia del código.
   2. Consume pocos recursos.
   3. Permite la reutilización de partes de código para otras aplicaciones.
   4. Poco portable.
2. En un supuesto práctico, ¿qué tipo de requisito es el deseo del cliente de incluir en la aplicación el control de stock de productos en el almacén?
   1. No funcional.
   2. Funcional.
   3. No es un requisito.
   4. Depende de la fase de codificación.
3. ¿Cómo se llama el cambio consistente en mejorar la funcionalidad de todo software?
   1. Correctivo.
   2. Perfectivo.
   3. Adaptativo.
   4. Evolutivo.
4. ¿Cómo se llama el proceso de traducción de código fuente a código objeto?
   1. Interpretación.
   2. Compilación.
   3. Ejecución.
   4. Runtime Environment.
5. Una máquina virtual
   1. Es un tipo especial de Hardware que se instala en el ordenador para poder ejecutar los programas.
   2. Es un tipo especial de software cuya misión es separar el funcionamiento del ordenador de los componentes hardware instalados.
   3. Es un tipo de software que se utiliza para compilar los programas de código fuente a código objeto.