

Ejercicios Tema 1

1. Realiza una comparativa entre dos entornos de desarrollo que habrás de elegir entre los que ofrece el mercado actual. Elige un Entorno de desarrollo libre y otro propietario para desarrollo software. De cada uno has de comparar distintas características:

IntelliJ IDEA	Eclipse
Java, Groovy, XML/XSL, Kotlin, ActionScript, CoffeeScript, HTML/XHTML/CSS, JavaScript, Ruby/Jruby/SQL, ...(Plugins)	Java, ANSI C, C++, JSP, sh, perl, php, sed
499€ primer año, 399€ segundo año, 299€ tercer año en adelante	Totalmente gratuito
Posibilidad de instalar plugins	Posibilidad de instalar plugins
Con soporte	Sin soporte empresarial ya que es código abierto
Tiene una versión gratuita, de la comunidad	Eclipse dispone de un Editor de texto con un analizador sintáctico. La compilación es en tiempo real.

2. Describe las diferencias entre lenguaje alto y bajo nivel. Realiza una comparativa (principales lenguajes y sus características, usos, ventajas, inconvenientes...).

Lenguaje Alto Nivel	Lenguaje Bajo Nivel
High Level Language es un lenguaje amigable para el programador que proporciona un alto nivel de abstracción del hardware.	El lenguaje de bajo nivel es el lenguaje que es fácil de usar para la máquina y no proporciona ni menos abstracción del hardware.
C, Java, Python	Binario
Un lenguaje de alto nivel es más lento que un lenguaje de bajo nivel.	Un lenguaje de bajo nivel es más rápido que un lenguaje de alto nivel.
Un lenguaje de alto nivel no es eficiente en memoria.	Un lenguaje de bajo nivel es más eficiente en memoria.
Un lenguaje de alto nivel requiere un compilador o un intérprete para convertir el programa en código de máquina.	El lenguaje ensamblador requiere que un ensamblador convierta el programa a un código de máquina mientras que la computadora ejecuta el lenguaje de máquina directamente.
Un lenguaje de alto nivel es fácilmente comprensible por el programador.	Un lenguaje de bajo nivel es fácilmente comprensible por la computadora.
Un lenguaje de alto nivel es independiente de la máquina.	Un lenguaje de bajo nivel depende de la máquina.
Un lenguaje de alto nivel puede ejecutarse en múltiples plataformas, por lo que es portátil.	Un lenguaje de bajo nivel no es portátil.
Un programa escrito utilizando un lenguaje de alto nivel es fácil de depurar y mantener.	Un programa escrito con un lenguaje de bajo nivel es difícil de depurar y mantener.
Los idiomas de alto nivel tienen más apoyo de la comunidad.	Los idiomas de bajo nivel no tienen mucho apoyo de la comunidad.

3. Busca información sobre el lenguaje C (origen, característica, sintaxis, lenguajes derivantes...).

C es un lenguaje de programación de propósito general originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

Lenguajes derivados:

- Objective-C
- C++
- Java
- JavaScript
- C#

La sintaxis en C es una serie de reglas y procesos que lideran la estructura de un programa. Estas reglas tienen que ser entendidas por el compilador para que se pueda crear un programa en C válido, es decir, tienen que establecer cómo comienza una línea de código en C, cómo termina o cuándo usar, por ejemplo, comillas o llaves. El lenguaje en C hace distinción entre mayúsculas y minúsculas siendo este el motivo por el que se programa en minúsculas.

4. ¿Qué es el modelado ágil? Consulta información en internet. ¿Qué opinión te merece? ¿Cuáles son sus ventajas y desventajas?.

La metodología de desarrollo ágil de software se refiere a métodos de desarrollo iterativo e incremental de acuerdo a la necesidad y complejidad del producto que se quiere desarrollar. El trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, que en conjunto toman decisiones a corto plazo.

Ventajas	Desventajas
Rápida respuesta a los cambios	Fuerte dependencia de los líderes
Intervención del cliente en el proceso	Falta de documentación
Entregas del producto a intervalos	Soluciones erróneas en etapas largas
Eliminación de tareas innecesarias	

5. Programación extrema (XP) ¿Cuál es su filosofía de desarrollo de aplicaciones? ¿Puedes relacionar este tipo de ciclo de vida con las herramientas CASE de alguna forma?.

La programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Se puede considerar la programación extrema como la adopción de las mejores metodologías de desarrollo de acuerdo a lo que se pretende llevar a cabo con el proyecto, y aplicarlo de manera dinámica durante el ciclo de vida del software.

Ambas quieren mejorar las metodologías de la programación.