**Hoja de Actividades 1.1**

|  |  |
| --- | --- |
| **Alumno/a** | Antonio Serrano Herruzo |

**1. Localiza distintos ejemplos de aplicaciones informáticas y completa la tabla aportando alguna información relevante sobre ellas**

|  |  |  |  |
| --- | --- | --- | --- |
| Aplicación | Tipo | Año | Información |
| Microsoft Windows 10 | Sistemas operativos | 2015 | Windows 10 es un sistema operativo de licencia privada desarrollado por Microsoft. Su primera versión apareció en 1985 y estaba desarrollada en C y C++ y lenguaje ensamblador. |
| Linux | Sistemas operativos | 1991 | Linux es un sistema operativo gratuito y de open source. Su primera versión fue desarrollada por Linus Torvalds en 1991. Este sistema es muy popular entre desarrolladores. |
| LibreOffice  The Document Foundation libera LibreOffice 3.5.2 | Contabilidad y ofimática | 2010 | LibreOffice es un paquete ofimático libre y abierto. Este paquete ofimático está disponible para múltiples sistemas operativos como macOS, Windows o Linux. Este incluye varias herramientas, como writer, calc, impress entre otras |
| Oracle Database | Gestión de Bases de datos | 1979 | Oracle es un sistema de gestión de base de datos relacional, desarrollado por Oracle Corporation. Es uno de los más potentes para gestionar grandes cantidades de datos |
| Adobe Photoshop | Diseño gráfico | 1990 | Es la herramienta de diseño gráfico más conocida. Esta herramienta permite retocar fotos, crear gráficos y realizar fotomontajes |
| Gmail | Correo Electrónico | 2004 | Se trata de un servicio de correo electrónico gratuito y de código abierto desarrollado por mozilla. Te permite gestionar tu correo de manera óptima y sencilla |
| Eclipse | Entorno de desarrollo | 2001 | Se trata de un entorno de desarrollo, creado por Microsoft, para el desarrollo de aplicaciones web, móviles, etc. Soporta gran cantidad de lenguajes como C#, C++, Python y Java Script |
| Visual Studio | Entorno de desarrollo | 1997 | Se trata de un entorno de desarrollo, diseñado por Java, pero que actualmente es compatible con otros lenguajes de programación como C++, Python y PHP. |
| NetBeans | Entorno de desarrollo | 2000 | Se trata de un IDE de código abierto que soporta varios lenguajes de programación como Java, PHP, C++ y JavaScript. Ofrece herramientas para la creación de aplicaciones móviles, web y empresariales. |
| Telegram | Mensajería instantánea | 2013 | Se trata de una aplicación de mensajería basada en la nube con un enfoque en la velocidad y seguridad. Ofrece chats secretos con cifrado de extremo a extremo, canales y grupos grandes. |
| Epic Games Store | Videojuegos | 1991 | Se trata de una Plataforma de distribución de videojuegos que ofrece una selección de juegos para PC y Mac, con un enfoque en títulos independientes y exclusivas. |

**2. Reflexiona sobre las ventajas e inconvenientes del software estándar y del software a medida.**

|  |  |  |
| --- | --- | --- |
|  | Ventajas | Inconvenientes |
| Software Estándar | * Es más barato, ya que esta desarrollado para un amplio mercado * Actualizaciones constantes para mejorar el rendimiento y la seguridad * Resuelve múltiples tareas esenciales en la gestión. | * Al estar diseñado para un mercado amplio, el software estándar puede no ajustarse perfectamente a las necesidades específicas de una organización. Es decir, carece de personalización * Dado que el software estándar busca cubrir un amplio rango de usuarios, a menudo incluye características que no son relevantes para todas las empresas o usuarios. * Algunas actualizaciones pueden ser obligatorias y no siempre deseables para el usuario. |
| Software  a medida | * Exclusivo: el desarrollador diseñara el software para que funcione exclusivamente en empresa contratante y bajo y las necesidades de implementación * Algunas actualizaciones pueden ser obligatorias y no siempre deseables para el usuario. * Adaptado: Todos los requerimientos del análisis se ven plasmados en el software por lo que las necesidades del cliente se ven completamente cubiertas. | * Tiene un coste elevado respecto al software estándar, debido a que el diseño de software personalizado requiere importantes recursos financieros. * Desarrollar una solución a medida de forma profesional puede llevar meses, e incluso años en proyectos de mayor envergadura, mientras que adquirir una solución estándar toma apenas unos minutos o unas pocas horas. * La empresa que desarrolla el software puede volverse dependiente del equipo de desarrollo o de la empresa que lo creó para futuras modificaciones, correcciones o mejoras. |

**3. Busca en Internet imágenes que muestren algún fragmento de código programado en los siguientes lenguajes de programación:**

|  |  |
| --- | --- |
| **Visual Basic .NET** | **C++** |
| **Java** | **JavaScript** |
| **C#** | IRIS y Python en la práctica - ¡con ejemplos! | InterSystems Developer  **Python** |
| Ensamblador?, ¿Lenguaje maquina?, Escribiendo un Hola mundo 🔥 | by Jimmy  Vasquez | Medium  **Lenguaje Ensamblador** | **Kotlin** |
| **PHP** | C (lenguaje de programación) - Wikipedia, la enciclopedia libre  **C** |

**4. Busca documentación en Internet y describe con tus propias palabras en qué se diferencia el modo de ejecución de un lenguaje de programación compilado frente a un lenguaje de programación interpretado.**

Los lenguajes de programación compilados traducen el código para lo entienda el ordenador antes de que se ejecute el programa, para así hacer que el programa sea más rápido.

Por otro lado, los lenguajes de programación interpretado el código se traduce línea por línea mientras se ejecuta.

**5. Busca información sobre, al menos, 5 lenguajes de programación y rellena la siguiente tabla indicando sus características el tipo (compilado/interpretado), su año de aparición y el paradigma/s que utilizan.**

|  |  |  |  |
| --- | --- | --- | --- |
| Logo | Tipo | Aparición | Paradigma |
|  | Compilado | 1972 | Imperativo, estructurado |
|  | Interpretado | 1991 | Orientado a objetos, Imperativo, Funcional |
|  | Compilado e interpretado | 1995 | Orientado a objetos, Imperativo |
|  | Interpretado | 1995 | Imperativo, Funcional, Orientado a objetos |
|  | Compilado | 2010 | Imperativo, Funcional, Orientado a objetos |
|  | Interpretado | 1995 | Orientado a objetos, Funcional, Imperativo |

**6. Visualiza la presentación** [**https://prezi.com/cqq7pc8xhy45/coodigo-fuente-codigo-objeto-y-codigo-ejecutable/**](https://prezi.com/cqq7pc8xhy45/coodigo-fuente-codigo-objeto-y-codigo-ejecutable/) **y define con tus propias palabras los siguientes conceptos:**

* **Algoritmo:** Es un conjunto de pasos o instrucciones que le dices a un ordenador para que haga algo.
* **Código fuente:** El código fuente es como el lenguaje en el que los programadores hablan con el ordenador.
* **Código objeto:** es el resultado de convertir el código fuente en un formato que el ordenador puede ejecutar.
* **Código ejecutable:** es como la versión final de tu programa que realmente puedes usar en el ordenador

**7. Consulta el enlace que se muestra a continuación y responde a las siguientes cuestiones:** [**http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm**](http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm)

* **¿Es Java un lenguaje compilado o interpretado? ¿Por qué?**

Es un lenguaje compilado e interpretado. Con el compilador se convierte el código fuente que reside en archivos cuya extensión es .java, a un conjunto de instrucciones que recibe el nombre de bytecodes que se guardan en un archivo cuya extensión es .class. El intérprete ejecuta cada una de estas instrucciones en un ordenador específico

* **¿Qué elementos intervienen cuando se ejecuta un código fuente en Java?**

Los elementos que intervienen cuando se ejecuta el código fuente de java son la Máquina Virtual Java y el API de Java

* **¿Qué es el bytecode?**

El bytecode permite la posibilidad de escribir el programa una vez y que se pueda correr en cualquier plataforma que disponga de una implementación de la Máquina Virtual Java

* **¿Qué función tiene la Máquina Virtual de Java MVJ (o Java Virtual Machine JVM)?**

Se encarga de ejecutar los programas creados a través del lenguaje utilizado. su misión principal es la de garantizar la portabilidad de las aplicaciones Java. Define esencialmente un ordenador abstracto y especifica las instrucciones que este ordenador puede ejecutar.

**8. Lee los siguientes artículos y describe con tus propias palabras cuál es la funcionalidad y características de los siguientes componentes:**

* **Ensambladores:** [**https://es.slideshare.net/luisrudeboy/14-ensambladores-y-compiladores**](https://es.slideshare.net/luisrudeboy/14-ensambladores-y-compiladores)

La función del ensamblador es traducir el lenguaje ensamblador a un código que el ordenador pueda entender, es decir, el lenguaje de máquina. Sus características son:

* Es demasiado complejo
* Te permite mayor control del ordenador
* La mayoría de ordenadores pueden ensamblar
* Es difícil de portar
* **Preprocesadores:** [**http://www.it.uc3m.es/pbasanta/asng/course\_notes/ch13s11.html**](http://www.it.uc3m.es/pbasanta/asng/course_notes/ch13s11.html)

El preprocesador en C es un programa que actúa como una etapa previa al proceso de compilación. Su función es preparar el código fuente antes de que sea traducido a código binario, modificando el archivo original a partir de las líneas de código que comienzan con el símbolo #. Este programa puede procesar un amplio catálogo de directivas

* **Intérpretes:** [**https://programar.best/programacion/que-es-un-interprete/**](https://programar.best/programacion/que-es-un-interprete/)

Un interprete es un programa cuya función es ejecutar un código en un lenguaje de alto nivel, traduciendo cada instrucción. A diferencia de los compiladores, los interpretes no guardan la traducción. Sus características son:

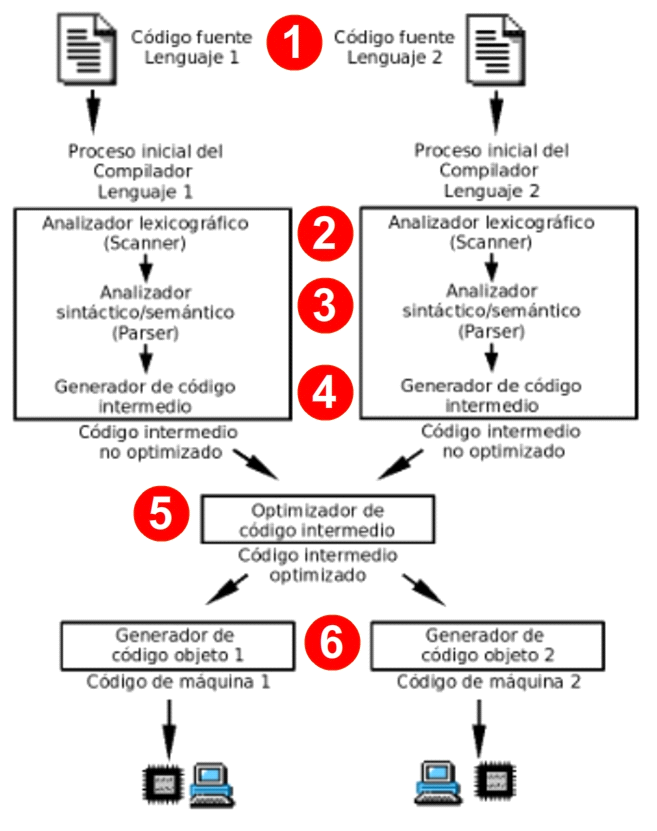
* Traduce y ejecuta una línea del programa a la vez.
* Si hay error, detiene el programa
* Programas más lentos, pero más flexibles
* **Compiladores.** [**https://programar.best/programacion/que-es-un-compilador/**](https://programar.best/programacion/que-es-un-compilador/)

Los compiladores son programas o herramientas que se encargan de compilar. La función del compilador es traducir un programa de un lenguaje de alto nivel a otro de nivel más bajo, normalmente es lenguaje de máquina. Este está compuesto por varias fases y hay diferentes tipos de compiladores

**9. Lee los siguientes artículos:**

* [**https://www.europeanvalley.es/noticias/que-es-un-compilador-en-programacion/**](https://www.europeanvalley.es/noticias/que-es-un-compilador-en-programacion/)
* [**https://tech.tribalyte.eu/blog-programa-o-compilador**](https://tech.tribalyte.eu/blog-programa-o-compilador)

**A continuación, explica qué está ocurriendo en las distintas fases del proceso de compilación que se muestra en la siguiente imagen:**

**Fase 1.-** Partimos del código fuente de un programa desarrollado en cualquier lenguaje de programación.

**Fase 2.-**

En la segunda fase se produce el analizador lexicográfico, es la primera etapa del proceso de compilación y se encarga de dividir el programa en Tokens. Cada token del programa es clasificado según su significado, para más tarde, ser procesados

**Fase 3.-**

En la tercera fase se produce el analizador sintáctico y el analizador semántico, es la segunda y tercera fase de compilación y genera un árbol sintáctico, que consiste en una estructura de datos compleja que permite representar de una forma más simple al programa fuente. A continuación, se validan los puntos más finos del programa. El analizador semántico es el que analiza que todo el programa tenga un significado exacto y que este no pueda fallar en tiempo de ejecución.

**Fase 4.-**

En esta fase aparece el generador de código intermedio, Después del análisis sintáctico y semántico, se genera un código intermedio no optimizado, que es una representación abstracta del programa, más cercana a un lenguaje de bajo nivel.

**Fase 5.-**

En esta fase se produce la optimización del código intermedio, esta fase toma el código intermedio generado y lo optimiza para mejorar la eficiencia y el rendimiento del código resultante, reduciendo instrucciones innecesarias o redundantes.

**Fase 6.-**

Por último, en esta fase se produce el generador de código objeto. Finalmente, el código intermedio optimizado se convierte en código de máquina específico para la plataforma de destino.

**Final.-** El código máquina generado ya puede ser ejecutado directamente por el microprocesador de la máquina destino.