

En esta tarea vamos a conocer algunos conceptos sobre Desarrollo de Software.

IDE

Estas siglas hacen referencia en inglés a “Integrated Development Environment” que viene a significar “Entorno de Desarrollo Integrado”. Es una aplicación informática que proporciona un servicio integral al programador para facilitar su tarea de desarrollo de software.

A groso modo consiste en un editor de texto donde añadir el código fuente del programa a desarrollar, una serie de herramientas de construcción y un depurador de código, aunque hay versiones de IDEs que son más completas y pueden añadir “IntelliSense”, que es un auto completador de código, distintos compiladores e intérpretes.

El fin máximo de un IDE es el de facilitar la tarea a un programador y maximizar su tiempo y esfuerzo. Esto lo logra unificando todas las tareas de creación de un software en una misma aplicación, tales como creación, modificación, compilación, implementación y depuración. Esto hace que aprender a manejar un IDE acabe siendo más rápido y productivo que llevar a cabo todas las partes por separado.

Los IDE pueden estar dedicados a un único lenguaje de programación o soportar varios de ellos, y los más modernos y potentes usan interfaz gráfica, añadiendo aún más comodidades para los programadores.

El IDE más implantado hoy día es Eclipse, por su sencillez y su soporte para varios lenguajes de programación por medio de plugins, tales como C/C++, Python, php y Java.

LIBRERÍA

Una librería es un archivo o conjunto de archivos que se utilizan para facilitar la programación. Para que un archivo pueda ser considerado una Librería deberá ser usada en varias páginas de diferentes sitios. En cuanto dicho archivo haya conseguido ese nivel de usabilidad se podrá considerar una Librería.

FRAMEWORK

Es un término inglés que viene a significar Entorno de Trabajo. Es un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problema concreto que sirve como referencia para resolver nuevos problemas de índole similar.

La definición proporcionada por Wikipedia es un poco “técnica” y en realidad explica poco y mal, así que en otras palabras un Framework es un entorno de trabajo usado por programadores para desarrollar su software. Utilizar un entorno de trabajo adecuado permite agilizar los procesos de dicho desarrollo ya que evita tareas ineficaces como repetir código, añade más seguridad al tener muchas vulnerabilidades resueltas por el propio Framework y aporta muchas herramientas al ser entornos conocidos por una gran comunidad que da continuo soporte. De ahí que elegir un buen Framework para una tarea concreta es fundamental.

Algunos de los más conocidos son .NET de Microsoft (sobretudo a nivel usuario cualquier usuario de este Sistema Operativo se ha tenido que enfrentar alguna vez a su instalación al ser requerido por infinidad de programas) y Meteor, uno de los mejores Frameworks para diseñar aplicaciones en JavaScript para Web y Móviles.

SDK

SDK son las siglas en inglés de “Software Development Kit” o Kit de Desarrollo de Software y consiste en un grupo de herramientas que permiten la programación de aplicaciones para móviles.

Estos paquetes de desarrollo se dividen en tres grandes grupos:

- SDK para entorno de programación en sistemas operativos móviles (mayormente IOS y Android).
- SDK para mantenimiento de dichas aplicaciones.
- SDK enfocado en el marketing y la publicidad.

Normalmente las SDK son proporcionadas por un fabricante de hardware, u sistema operativo o un lenguaje de programación. Estos kits de herramientas permiten a los desarrolladores de software crear aplicaciones para esa plataforma concreta (hardware, SO o lenguaje). Un buen ejemplo práctico es pensar en dichas SDK como las instrucciones y la bolsa con tornillos y llaves que acompañan a los muebles de IKEA para su montaje. Por lo general, un SDK, dependiendo del fabricante, suele incluir una serie de manuales para aprender sobre dicho SDK y aquello para lo que se va a programar, un compilador, un depurador y varias “Interfaces de Programación de Aplicaciones” (API). Algo importante a tener en cuenta si se va a trabajar con un SDK es el posible acuerdo de confidencialidad que pueda existir con el fabricante de dicha SDK, al crear software para un entorno cerrado como puede ser el sistema operativo de Apple.

CÓDIGO FUENTE

Se denomina Código Fuente a un conjunto de instrucciones escritas en un lenguaje comprensible por el ser humano usando la sintaxis de un lenguaje de programación con el fin de guiar el proceso de ejecución de un programa.

CÓDIGO OBJETO

Es el conjunto de instrucciones que se obtienen al compilar un código fuente y que pasan a ser entendidos por la máquina.

EJECUTABLE

Un ejecutable es un archivo diseñado para poder iniciar un programa. Dicho archivo contiene las instrucciones precisas que el programador ha diseñado para decirle al ordenador lo que quiere que su ejecutable lleve a cabo.

COMPILADOR

Un compilador es un programa que traduce código fuente escrito en un lenguaje de alto nivel (es aquel que está más próximo al lenguaje humano, como C, C++, Cobol...) a un lenguaje legible por la máquina llamado código objeto.

La forma de trabajar de un compilador es a través de un documento con el código fuente del programa ya completo y escrito en un lenguaje de programación de alto nivel. Dicho código fuente es escaneado y se produce un análisis del léxico y una corrección sintáctica. Una vez se ha determinado la semántica del código se pasa a crear el código entendible por la máquina optimizando u ocultando los errores presentes en el código.

INTÉRPRETE

Los intérpretes son muy parecidos a los compiladores pues su fin último es exactamente el mismo, traducir código fuente a código máquina para que los programas escritos en lenguajes de alto nivel puedan ser entendidos por las computadoras.

La principal diferencia es que los intérpretes no leen todo el código de golpe una vez el programa esté completo, si no que van leyendo cada línea del código fuente por separado y la van ejecutando en la plataforma. De esta forma se entiende que el intérprete hace dos pasos en uno. Primero lee la línea traduciéndola a lenguaje máquina y luego la ejecuta.

TRADUCTOR

Es la parte de los compiladores y los intérpretes que se encargan de la conversión de código fuente a código máquina.

ANALIZADOR SINTÁCTICO

Un analizador sintáctico o como se le conoce en inglés, parser, es un programa que forma parte de los compiladores. El parser es quien se encarga de descomponer y transformar las entradas de código para posteriormente procesarlo. En otras palabras, analiza una cadena de instrucciones que está en un lenguaje de programación y la descompone en instrucciones individuales.