Introducción al proceso de desarrollo.

Para meternos en el mundo de la programación, debemos tener presentes los elementos iniciales que forman parte de los procesos ligados al desarrollo de aplicaciones.

Como futuros programadores, nos encontraremos con problemas a resolver, y para encontrar la solución adecuada iremos confeccionando aplicaciones informáticas.

En el mundo del desarrollo, veremos que cada problema es único, pero son únicas las soluciones posibles y es parte de nuestro trabajo diseñar la respuesta más eficiente para cada problema al que nos enfrentamos.

En el contexto actual nos encontramos a diario con diferentes tipos de sistemas y aplicaciones, como, por ejemplo, sistemas operativos para controlar dispositivos, juegos, gestores de información, plataformas de compras, páginas web, etc.

Detrás de todo este mundo de soluciones se encuentran los programadores, analistas, diseñadores y muchos profesionales más, cuyo trabajo conjunto hace posible que existan estos elementos que ya forman parte de nuestras tareas diarias, sin importar a que nos dediquemos.

Una definición bastante aceptada de lo que significa en el mundo IT el desarrollo, es confeccionar, probar y gestionar errores en un programa destinado a dar solución a un determinado problema.

Cuando nos proponemos aprender a programar, lo hacemos con el fin de cubrir determinadas necesidades, ya sean personales o de terceros (clientes) a cambio de un rédito económico.

El primer paso que tendremos que dar como futuros programadores, es aprender lo que llamamos, programación lógica, muy importante porque si bien cada lenguaje tiene sus particularidades, los problemas son analizados desde un punto de vista lógico único y de un mismo modo sin importar las herramientas que utilicemos para diseñar y desarrollar nuestras soluciones. Esto es sumamente conveniente ya que nos permite migrar nuestra solución a cualquier lenguaje de programación que necesitemos implementar.

Resolución de problemas.

Cuando desarrollamos una solución, ya sea por trabajo o con fines académicos, partimos siempre de un problema a resolver, al cual debemos encontrarle la mejor solución.

El problema a resolver o la necesidad para la cual programamos, puede ser de índole personal, para realizar pequeñas tareas con un fin determinado que nos beneficie, como puede ser una aplicación para nuestro móvil que nos informe nuestras tareas pendientes, una web o un CV en línea o bien puede ser del ámbito empresarial, donde podríamos necesitar realizar sistemas, partes de sistemas o nuevos módulos, incluso arreglar un código escrito por alguien más, o automatizar procesos que se estén llevando a cabo de manera manual.

Sea cual fuere el caso, debemos tener siempre presente nuestras metas, determinar fehacientemente el objetivo al cual apuntamos, analizar las posibles etapas e incumbencias del proyecto y evaluar correctamente la viabilidad del mismo.

Cuando programamos podemos hacerlo para diferentes entornos o ecosistemas, trabajar con diferentes arquitecturas (formas de estructurar nuestros desarrollos) y de esa manera hacer aplicaciones móviles, web, de escritorio, sistemas operativos, SCRIPTS (porciones de código con un fin específico) o procesos.

Aplicaciones web.

Todos conocemos y utilizamos, cada día más, las aplicaciones desarrolladas para el mundo web. En general las agrupamos en dos grandes categorías, las aplicaciones estáticas, más orientadas a mostrar imágenes y datos estáticos o animaciones simples y las dinámicas, que son aquellas más complejas que, además de toda parte que se ve, cuenta además con toda una estructura por detrás, lo que llamamos BACKEND que, entre otras cosas, incluyen un servidor, un intérprete del lenguaje de programación y un sistema de gestión de bases de datos.

Muy populares en el mundo empresarial, les permiten a las compañías trascender las limitaciones de la actividad presencial y llegar literalmente a todo un mundo de potenciales clientes.

Trabajan sobre lo que llamamos arquitectura cliente – servidor, siendo quienes solicitan ver o acceder a determinada aplicación web, los clientes, y dentro del servidor está implementada toda la infraestructura de proceso y bases de datos.

Este tipo de tecnología no solo es aplicable en los negocios, también son muy útiles desde hace ya bastante tiempo, en la educación, medicina y todo tipo de actividades que requieran interacción entre personas o de personas con sistemas complejos.

Existe una muy amplia gama de sitios web dedicadas a casi todas las actividades que se nos puedan ocurrir, es una rama de la programación muy requerida en el mundo laboral y que continuamente está incorporando nuevas tecnologías y métodos de trabajo, pasando a ser desde hace ya un tiempo, una forma de relacionamiento cada vez más presente y aceptada como alternativa de trabajo.

Aplicaciones de escritorio.

Las aplicaciones de escritorio, son aquellas que funcionan sobre un sistema operativo, de dispositivos tales como un servidor, pc de escritorio o notebook.

Este tipo de desarrollos suelen ser bastante más grandes que sus pares de otra arquitectura, como la web, y a menudo más costosos, ya que están orientas a un público determinado, generalmente a demanda y con requerimientos específicos.

Muchos de estos programas cuentan con versiones de prueba (TRIAL) destinadas a hacer conocido el proyecto o al testeo de errores con usuarios reales, lo que llamamos versión BETA, y desde luego, las versiones finales, no gratuitas con diferentes configuraciones dependiendo del grupo de usuarios y objetivos al que esté destinado.

Aplicaciones móviles.

Son los programas que se desarrollan para dispositivos móviles como teléfonos celulares o tabletas.

A simple vista puede que nos resulten similares a las aplicaciones de escritorio, pero desde su construcción y diseño, se basan en paradigmas de programación muy distintos a los del software de escritorio.

Existen muchos sistemas operativos para estos dispositivos móviles, siendo tal vez el más conocido, Android, cuya arquitectura tiene entre sus características, la posibilidad de desarrollar aplicaciones seguras, portables y con una simpleza de acceso al hardware del dispositivo, que nos permite un nuevo mundo de posibilidades a la hora de programar.

En particular, el mundo de las aplicaciones móviles y nuestra orientación dentro del mismo, estará determinado por el público objetivo al cual deseamos llegar, siendo el sistema operativo Android uno de los más populares y que no requiere demasiada inversión a la hora de iniciar un proyecto.

Se trabaja con herramientas de uso gratuito, en cuanto a lenguajes de programación, los dos lenguajes oficiales aceptados por Google son Java y Kotlin y como herramienta para el desarrollo integral de aplicaciones, Android Studio, que, si bien es gratuito y cuenta con todo lo necesario para desarrollar cualquier tipo de actividad, tiene algunos requerimientos de hardware bastante más exigentes que la mayoría de los entornos.

Sistema.

En programación entendemos como sistema a todo conjunto de instrucciones, trabajando en conjunto con un fin común.

Puede tratarse de un complejo sistema de información, hasta un simple tratamiento de datos, que, ya que lo mencionamos, siempre contará con los mismos elementos básicos, una entrada de datos, un proceso de los mismos, y una salida.

Una entrada es un ingreso de datos o comandos, ejecutados sobre un dispositivo, como, por ejemplo, un teclado, una pantalla táctil, un comando de inserción de datos desde un origen, como puede ser un archivo que se cargue en una memoria, una cámara web o un micrófono.

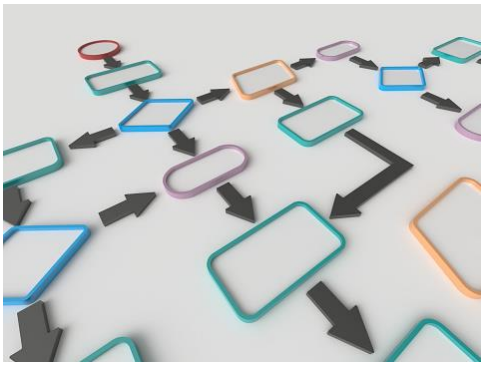
El trabajo que hacemos con los datos que fueron ingresados, se llama proceso y va desde un simple ordenamiento hasta un complejo sistema de cálculos o transformaciones, diferentes pasos todos programados en algún lenguaje informático, donde entran en acción elementos tales como el procesador y la memoria del dispositivo donde se esté ejecutando este procesamiento.

Finalmente, la salida es el resultado de estas acciones, efectuadas durante el proceso, sobre los datos ingresados.

Puede ser una salida por pantalla, una impresión o directamente guardar los datos resultantes en una base de datos.

Algoritmo.

Si bien existen múltiples definiciones de lo que es y lo que no es un algoritmo, vamos a enfocarnos y trabajar sobre la definición formal que nos da la RAE respecto a los mismos, a los que define como un conjunto ordenado y finito de operaciones que nos permite hallar la solución a un problema.

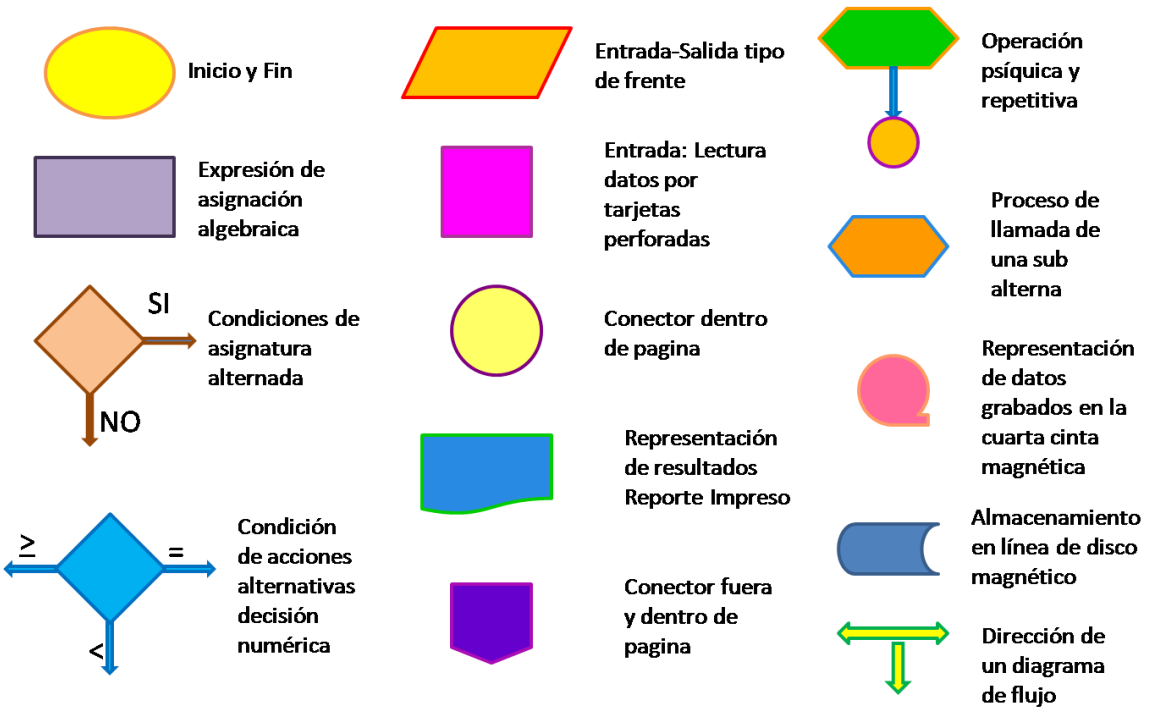


Las personas tenemos incorporados nuestros propios algoritmos de decisión y ejecución de procesos cotidianos.

Desde decidir con que ropa vestirnos hasta armar una lista de compras del supermercado. Y, como bien dijimos, todo esto lo hacemos en base a un número finito de operaciones, que, conociendo la simbología adecuada, podríamos expresarlos como un algoritmo.

En general, conociendo solo unos pocos símbolos de algoritmos o diagramas de flujo de datos como también se los conoce, podemos armar un diagrama tan complejo como lo necesitemos.

Veamos primero cuales son estos elementos y cómo encarar un problema mediante el uso de estos diagramas.

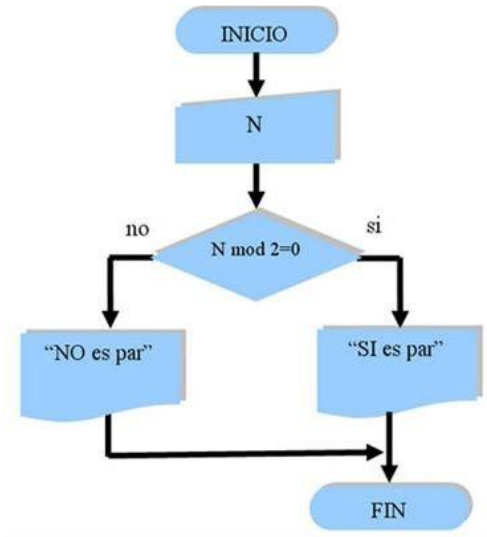


Fuente:

https://www.diagramasdeflujo.com/flujograma/diagrama-de-flujo-de-procesos/

Veamos ahora un ejemplo sencillo para expresar un proceso mediante el uso de un algoritmo o diagrama de flujo que nos permita tomar un número

cualquiera y analizar su condición de paridad, es decir, determinar si es par o impar.



Comenzamos ingresando o leyendo un valor numérico. A continuación, lo dividimos sobre dos y verificamos si el resto de dicha división es cero, de ser así, determinamos que el número es par, caso contrario, el número leído será impar.

Este tipo de diagramas son muy útiles para el programador y nos permiten trasladar rápidamente una solución lógica a un lenguaje de programación.

Lenguajes de programación.

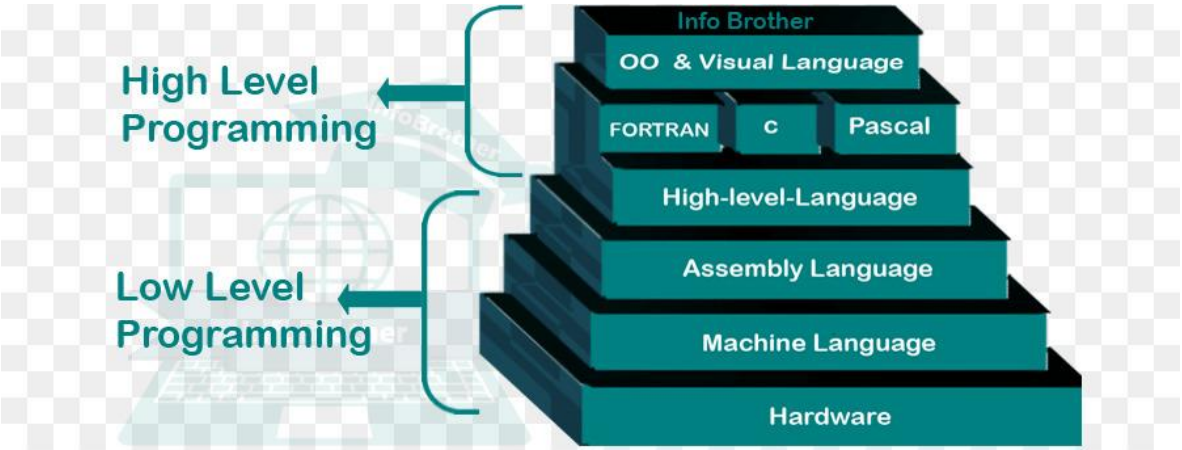
Los dispositivos para los cuales escribimos nuestros programas, como por ejemplo una computadora o un teléfono móvil, solo entienden un lenguaje propio, conocido como lenguaje de máquina.

A los programadores nos resulta algo complejo desarrollar nuestros trabajos en dicho lenguaje, por lo que necesitamos de alguna herramienta, que nos permita plasmar nuestras ideas para luego entregarlas al dispositivo y que ejecute nuestras órdenes en la manera en que nosotros queremos que sean ejecutadas.

Para ello existen los lenguajes de programación.

Los hay de sintaxis algo compleja y difícil de aprender, pero muy eficientes respecto al uso de recursos (lenguajes de bajo nivel) y los hay muy sencillos de entender e implementar, pero en ocasiones, no lo suficientemente rápidos, (lenguajes de alto nivel).

Todos los lenguajes fueron diseñados para un propósito específico, y nosotros como programadores, tenemos que tener presente el objetivo de nuestro desarrollo, entender bien el problema y determinar que lenguaje, en base a sus características, es mejor para nuestra solución.



Dentro de las diversas que clasificaciones que usamos para los lenguajes de programación, no es esta (alto o bajo nivel) una de las que más nos interesa.

Pero es importante saber, resumiendo, que hay lenguajes cuya sintaxis es más parecida al lenguaje nativo de las máquinas (bajo nivel) y lenguajes cuya sintaxis es más parecida al lenguaje natural del ser humano (alto nivel).

Otro factor a tener en cuenta a la hora de clasificar a los lenguajes de programación, es la forma en que son traducidos, desde el código que nosotros escribimos, al lenguaje de máquina.

Existen dos formas de hacer esta traducción.

La primera es mediante un intérprete, que vaya leyendo línea a línea nuestro programa y a medida que las lee, las traduce para la máquina. A estos lenguajes se los conoce como interpretados, como, por ejemplo, Python y PHP.

La segunda opción, es mediante un compilador, que es un programa encargado de leer todo nuestro código y traducirlo por completo a lenguaje máquina, dejando como resultado, uno o más archivos ejecutables que son directamente ejecutados por el dispositivo, por ejemplo, Java.

Por último, podemos ir un paso más allá y ver en qué lugar será ejecutado nuestro código, ya que hay lenguajes (en especial en el mundo de la programación web) que se ejecutan del lado del cliente, como JAVASCRIPT y lenguajes que se ejecutan del lado del servidor, como PHP, y ambos son interpretados (traducidos a código máquina mediante un programa intérprete).

Entornos de trabajo.

Dentro de las herramientas que utilizaremos como programadores, tenemos editores de texto, IDE y FRAMEWORW.

Cada uno está reservado para un tipo de tarea particular.

Al programar, necesitamos escribir nuestro código de manera simple y organizada, por lo que, además de contar con algún sistema propio de planificación y orden, necesitaremos de algún programa que nos permita escribir código de manera eficiente.

Dentro de esta categoría tenemos a los editores de texto, que van desde simples programas para escribir y entender código, hasta algunos algo más complejos que permiten ejecutar nuestro código, conectarnos con bases de datos y hasta nos marcan los errores de sintaxis, errores de estructura y

nos dan sugerencias de que instrucciones deberíamos utilizar o auto completan lo que vamos escribiendo.

Hay muchos editores de este tipo y la mayoría absolutamente gratuitos, cómo, por ejemplo, VISUAL STUDIO CODE, NOTEPAD++ y SUBLIME TEXT.

Páginas de descarga:

- https://code.visualstudio.com/download
- https://www.sublimetext.com/
- https://notepad-plus-plus.org/downloads/

Todas estas opciones son igualmente útiles y completas a la hora de decidirse, solamente habrá diferencias más ligadas con la comodidad y la experiencia que con las funcionalidades de cada uno.

Otra de las herramientas infaltables, siempre que la necesitemos por el lenguaje que decidamos aprender o utilizar, es el IDE, entorno integral de desarrollo por sus siglas en inglés.

El IDE es un programa complejo que trae todas las funcionalidades necesarias para desarrollar e implementar nuestras creaciones directamente desde dicho IDE y sin necesidad de descargas o herramientas complementarias.

Ejemplos de estos programas son Visual Studio (no confundir con Visual Studio Code) y Android Studio.

Por último, un FRAMEWORK es un conjunto de herramientas y utilidades que, en conjunto, conforman un marco de trabajo que nos permite desarrollar en determinadas tecnologías o entornos sin necesidad de escribir código de más y en donde la mayoría de las necesidades ya se encuentran resueltas, por ejemplo, .NET, ANGULAR y BOOTSTRAP.