

LENGUAJES DE DESARROLLO

ANDROID STUDIO



INTRODUCCIÓN

La programación de aplicaciones móviles nunca requiere un idioma. Ya sea para iOS o Android. La compilación, por ejemplo, es un paso final y se puede hacer con una variedad de idiomas diferentes. Si su aplicación usará la web o fuera de las API (interfaz del programa de aplicación), necesitará utilizar algunos lenguajes de programación de conectividad y almacenamiento de datos.

No se preocupe demasiado por esto ahora, pero recuerde que eventualmente necesitará algunos idiomas en su haber. Los mejores y más modernos lenguajes de programación utilizan lo que se llama programación orientada a objetos o POO.

Los lenguajes de programación antiguos eran muy exhaustivos y requerían largas líneas de código para hacer cosas simples. Pero la **Programación Orientada a Objetos** divide el código en objetos que luego se conectan entre sí. ¿Cuál es el Mejor Lenguaje de Programación para Android Apps?

Los dos lenguajes de Programación Orientada a Objetos más grandes que puede usar para programar una aplicación de Android son **C++** y **Java**.

LENGUAJES

C++

- C++ no es fácil de aprender y definitivamente no es un punto de partida recomendado.
- Es un lenguaje orientado a objetos más antiguo con muchas reglas y limitaciones.
- Está estandarizado por la Organización Internacional de Normalización. Que ayuda al idioma y ralentiza su desarrollo.
- C++ ofrece el más alto nivel de control con acceso a librerías C++ y desarrollo nativo. Esto significa que su código se comunicará directamente con el dispositivo.
- C++ es un lenguaje complejo de aprender, y puede encontrarse con serios problemas como principiante. Por un lado, no hay recolección de basura automática. Esto significa que los objetos no se eliminan de la memoria del dispositivo. A menos que codifique esto manualmente, la memoria continuará llenándose y eventualmente la aplicación se bloqueará.
- C++ también es muy implacable cuando comete errores en el código. No se darán mensajes de error o excepciones. Su código se bloqueará sin explicación.

LENGUAJES

JAVA

- Java es un lenguaje un poco más amigable para principiantes. A pesar de las similitudes en el nombre, Java no está relacionado con JavaScript.
- Es un lenguaje orientado a objetos muy universal y se ejecuta en la **JVM - Máquina Virtual de Java**.
- La mayoría de los idiomas funcionan al convertir el código en un idioma que una computadora puede leer. La Máquina Virtual Java primero convierte su código en otro idioma y luego puede convertirlo para que funcione en cualquier sistema operativo, incluidos Android e iOS.
- Esto hace que la conversión de código Java para Android a iOS y otros dispositivos sea mucho más fácil. Sin embargo, la codificación en Java significa que se está comunicando con la Máquina Virtual de Java y no de forma nativa.
- A diferencia de C++, Java da mensajes de error y te ayuda a definir qué está mal con tu código. Esto hace que sea más fácil para los programadores principiantes.

LENGUAJES

JAVASCRIPT

- Antes de analizar este lenguaje de programación, es importante decir que Java nada tiene que ver con JavaScript, son dos lenguajes de programación que no tienen que ver entre sí.
- JavaScript es un **lenguaje web**; con plataformas de desarrollo como IONIC se puede compilar de forma nativa, lo que es una ventaja con respecto a Java, esto se debe a que los lenguajes web son más sencillos y ligeros que los nativos.
- Las instrucciones se suelen escribir en JavaScript e IONIC se encarga de traducir al lenguaje nativo, no sólo de Android, también iOS o Windows 10 Mobile.
- De esta manera se puede acceder a los recursos físicos del dispositivo como por ejemplo la cámara o el GPS, el micrófono, etc.
- Por todo ello JavaScript es una buena elección a la hora de programar aplicaciones en Android

LENGUAJES

KOTLIN

- **Kotlin** es un lenguaje desarrollado por **Google**. Desde que fue desarrollado por Google, funciona a la perfección con Android y Java.
- Este nuevo lenguaje está inspirado en Java, C++ y todos sus precedentes.
- **Kotlin** es con mucho, el más accesible. Es un lenguaje muy limpio y relativamente simple, con menos formalidades y reglas que C++ y Java.
- Si bien aún necesita comprender conceptos y estructuras básicas de programación. Generalmente puede lograr el mismo resultado con menos líneas de código.
- Las estimaciones aproximadas indican aproximadamente un recorte del **40% en el número de líneas de código**.
- Kotlin también está en el proceso de convertirse en totalmente compatible con iOS, que es otra gran ventaja.
- Desafortunadamente, dado que este lenguaje es nuevo y no está estandarizado. Hay menos soporte y solución de problemas disponibles. Lo que puede dificultar la tarea de los principiantes.

LENGUAJES

HTML5 + CSS

- Son lenguajes completamente web, se trata de lenguajes de etiquetación creados para Internet, el paso del tiempo los ha beneficiado, desde la llegada de la versión 5 de HTML (**lenguaje de marca de hipertexto**).
- Esta versión es un salto cualitativo con respecto a la versión 4, con la nueva versión, se puede reproducir vídeo de un modo más nativo en el navegador, sin necesitar flash, esto es muy positivo, porque reduce notablemente el consumo de recursos.
- CSS (**hojas de estilo en cascada**) es un **lenguaje de incorporación**, que permite mejorar la presentación de contenido, este lenguaje trata por separado la parte estética o de presentación del HTML5, por lo que lo hace fácil de reemplazarlo y mejorar rápidamente su apariencia.

LENGUAJES

XML

- XML significa **Lenguaje de etiquetas extensible**. XML es un lenguaje de etiquetas muy parecido al HTML utilizado para describir datos.
- Las etiquetas XML no están predefinidas en XML. Debemos definir nuestras propias etiquetas (**TAG**).
- Xml ya que es bien legible tanto por humanos como por máquinas. Además, es escalable y simple de desarrollar.
- En Android usamos xml para diseñar nuestros diseños porque xml es un lenguaje liviano por lo que no hace que nuestro diseño sea pesado.
- El concepto completo de la interfaz de usuario de Android se define utilizando la jerarquía de los objetos **View** y **ViewGroup**.

CONSIDERACIONES DE LOS LENGUAJES – TIPOS DE DATOS

Todo sistema de computación es alimentado con datos. La representación de los datos incide en la elaboración de los algoritmos. El sistema de identificación de los datos se denomina "**tipos de datos**", y depende de las características de estos.

Los datos se clasifican:

- a) Datos numéricos
- b) Datos alfanuméricos
- c) Datos lógicos

CONSIDERACIONES DE LOS LENGUAJES – TIPOS DE DATOS

DATOS NUMÉRICOS

Enteros: corresponden a números "completos", o sea que no tienen parte decimal, pueden ser negativos, positivos o cero. Ejemplo: La edad de una persona (39), el total de estudiantes en un grupo (36).

Reales: son aquellos números que poseen punto decimal. Pueden ser negativos, positivos. Ejemplo: 37.0 ; 1.0

Reales decimales: son los que constan de una parte entera y una decimal (en algunos casos se puede omitir la parte entera o la decimal). Ejemplo: La nota de un estudiante (4.7), el porcentaje de algún impuesto (0.16).

Reales exponenciales: (de punto flotante); son aquellos números que se expresan en función de potencias de 10; esta expresión es conocida como notación científica. Ejemplo: -2.85E03; 0.735E-4; 0.58E-1

CONSIDERACIONES DE LOS LENGUAJES – TIPOS DE DATOS

DATOS ALFANUMÉRICOS

Corresponden al conjunto de caracteres alfabéticos, numéricos, y/o especiales; con ellos no se efectúan operaciones matemáticas.

Carácter: cuándo se hace referencia a un solo carácter, se debe encerrar entre apostrofes o comillas simples (' '). Ej: Sexo de una persona ('M' o 'F'), estado civil ('S', 'C', 'V').

Cadena: Cuándo se hace referencia a más de un carácter, siempre debe encerrarse entre comillas dobles (" "). Ej: código de un estudiante ("2048"), nombre de una persona ("Victor Hugo Torres"), dirección donde reside una persona ("Calle 100 # 50 – 20"), número telefónico ("300-2125820").

CONSIDERACIONES DE LOS LENGUAJES – TIPOS DE DATOS

DATOS LÓGICOS

Son aquellos datos que solo pueden tomar dos valores:

Valor verdadero – **true**
Valor falso – **false**

Muchas veces también se hace referencia a los números binarios ya que estos solo pueden tener 2 estados:

Valor verdadero – **1**
Valor falso – **0**

CONSIDERACIONES DE LOS LENGUAJES - OPERADORES

Un **operador** es un símbolo que se utiliza para manipular datos.
En matemáticas, el término **operador** puede usarse con varios significados. Algunas veces, un **operador** es un símbolo matemático que indica que debe ser llevado a cabo una operación especificada sobre un cierto número o números. Los operadores suelen interpretarse como funciones, por ejemplo la suma + o el producto X.

Los operadores se clasifican en:
Aritméticos.
Comparación
Relacionales Lógicos.

CONSIDERACIONES DE LOS LENGUAJES - OPERADORES

OPERADORES ARITMÉTICOS

- + Suma
- Resta
- * Multiplicación
- / División (div)
- Mod **Módulo o residuo (%)** $15 \bmod 5 = 0$,
ya que $15/5=3$ $15 - (5*3)=0$
- mod=0
- ^ ó (**) Potencia
- ← ó (=) Asignación

CONSIDERACIONES DE LOS LENGUAJES - OPERADORES

OPERADORES COMPARACIÓN

- > Mayor que
- < Menor que
- == Igual a
- != Diferente de
- >= Mayor o igual que
- <= Menor o igual que

CONSIDERACIONES DE LOS LENGUAJES - OPERADORES

OPERADORES RELACIONALES

- * ó && ó and Conjunción (y)
- + ó || ó or Disyunción (o)
- ~ ó ! ó not Negación (no)

CONSIDERACIONES DE LOS LENGUAJES - IDENTIFICADORES

Un **identificador** es el nombre que se le da a un elemento de un algoritmo, pseudocódigo, programa. Por ejemplo, el tipo de dato entero hace referencia a un tipo de dato que es distinto a todos los demás tipos de datos, es decir, los valores que puede tomar un dato de tipo entero, no son los mismos que los que puede tomar un dato de otro tipo como por ejemplo un carácter.

La mayoría de identificadores son definidos por el programador según su necesidad, su nombre puede ser desde tan solo una letra hasta palabras e incluso frases (*variables y constantes*). A la hora de asignar un nombre a un elemento, se debe de tener en cuenta que todo identificador debe cumplir unas reglas de sintaxis.

CONSIDERACIONES DE LOS LENGUAJES - IDENTIFICADORES

REGLAS DE SINTAXIS - CONFORMACIÓN PARA NOMBRAR IDENTIFICADORES

1. Consta de uno o más caracteres
2. El primer carácter debe ser una letra o el carácter subrayado (), mientras que, todos los demás pueden ser letras, dígitos o el carácter subrayado ().
3. No deben comenzar con un número
4. Las letras pueden ser minúsculas o mayúsculas del alfabeto inglés.
5. No está permitido el uso de las letras 'ñ' y 'Ñ'; y peor aun letras tildadas.
6. No pueden existir dos identificadores iguales, es decir, con el mismo nombre. Lo cual no quiere decir que un identificador no pueda aparecer más de una vez en un algoritmo.



LENGUAJES DE DESARROLLO

ANDROID STUDIO

