**Treball de Recerca**

Pablo Fraile Alonso

**Índex:**

**1. Introducció al llenguatge de programació:**

1.1 Definició de programar

1.2 Definició de programa.

1.2.1 Explicació del codi font

1.2.1.1 Explicació de les paraules reservades

1.2.1.2 Explicació de les anotacions

1.2.2 Executable

1.2.3 Explicació de la variant d'un programa anomenat aplicació

1.3 Diferents tipus de llenguatges de programació i la seva utilitat

1.4 Introducció a les llicències dels llenguatges de programació i de software

1.4.1 Llicències de Software Lliure

1.4.2 Llicències de Software Propietari

**2. Introducció a Java:**

2.1 Definició i característiques del llenguatge de programació Java

2.1.1 Programació dedicada a objectes (POO)

2.1.2 ByteCode Java

2.1.3 Java virtual Machine

2.1.4 Utilitzacions del llenguatge de programació Java

2.2 Funcionament de Java

2.3 Historia de Java

2.4 Versions del Java JDK

**3. Aplicació del llenguatge Java en diferents ecosistemes:**

3.1 Packages de Java en Android

3.2 Adaptacions de Java en Android

**5. Android Studio:**

5.1 Introducció de Android Studio

5.2 Introducció al compilador Gradle i la substitució de JVM

**6. Android App ( Part Visual ):**

Introducció del llenguatge XML adaptat a Android Studio

Icones (mesures, adaptació a les noves versions de Android)

Notificacions ( Barra d’estat, pantalla de bloqueig i app icon badge)

**1. Introducció al llenguatge de programació:**

**1.1 Definició de programar:**

El verb programar es refereix a l'art de fer que un dispositiu electrònic ( com per exemple un telèfon mòbil ) dugui a terme unes determinades accions que l'usuari anomenat programador li ha encomanat anteriorment. Una vegada explicat de mode general què és programar, cal recalcar que tal i com es veurà posteriorment, hi ha diferents tipus de llenguatges de programació que poden oferir més accions que altres, però finalment tots tenen com a funció realitzar accions encomanades anteriorment per el programador.

**1.2 Definició de programa:**

Per a poder definir la paraula programa, s'ha decidit citar la definició que dona la Universitat Internacional d' Atlanta ja que explica correctament què és un programa:

"Un programa es un conjunt d'instruccions o ordres basades en un llenguatge de programació que un ordinador interpreta per a resoldre un problema o una funció específica."

El conjunt de programes per a fer diferents funcions d'un dispositiu electrònic s'anomena Software.

**1.2.1 Explicació del codi font:**

El codi font és una de les parts més importants d'un programa. El codi font d'un programa és el conjunt de línies que un programador escriu amb un llenguatge de programació amb els passos que ha de seguir un dispositiu per a executar dit programa.

Un exemple de codi font és el de la imatge 1.1, que com es pot apreciar és un conjunt de línies escrites en un llenguatge de programació ( en aquest cas el llenguatge utilitzat és Java). Aquestes línies de codi són essencials per a que el programa, quan sigui executat tingui les ordres que ha de fer.



1.1 Exemple de codi font. Font: Imatge pròpia

**1.2.1.1 Explicació de les paraules reservades en el codi font:**

Dintre de el codi font, encara que aquest depengui del llenguatge de programació, es poden trobar ítems que es trobaran en tots els llenguatges de programació. Un d'aquests ítems son les paraules reservades.

Les paraules reservades, son identificadors dintre d'un codi font, predefinits per el propi llenguatge de programació que tenen un significat "especial" assignat i que no es poden utilitzar com identificadors de qualsevol element en el seus programes.

Per a entendre millor aquest fet, s'introduirà una paraula reservada en el codi font d'un programa de java (però podria ser aplicat a qualsevol llenguatge de programació, amb qualsevol paraula reservada d'aquest.)

En java, una paraula reservada es per exemple la paraula "int". Per tant, is li assignem nosaltres un identificador, el programa ens donarà error a l'hora de executar l'acció.

1.1.1 Exemple de paraules reservades Font: Pròpia

Com es pot apreciar en la imatge 1.1.1, si fiquem una la paraula int en qualsevol lloc de el programa (exceptuant algunes funcions, que seran explicades a la part pràctica) el llenguatge la detectarà com a una paraula reservada.

En canvi com la paraula "noerror" no es una paraula reservada, el codi s'executarà sense cap error.

**1.2.1.2 Explicació de les anotacions:**

En gairebé tots els llenguatges de programació, podem trobar una eina que s'utilitza per a organitzar el codi font i explicar a altres programadors per a què serveix i que fa aquell codi font. Aquesta eina són els anomenats símbols d'anotacions. En cada llenguatge s'associen dos o tres símbols que son ignorats per el programa i que el únic que fan és que el programador que escrigui el codi pugui explicar per a què serveix qualsevol línia de el codi font, i per tant, donar informació a altres programadors o usuaris que entenguin el llenguatge de programació.

En el cas de java, c o c++ les anotacions simples són donades amb els símbols "// "



1.1.2 Anotacions Font: Pròpia

**1.2.2 Executable**

Un executable és el conjunt d'arxius que formen un programa. Això vol dir que un executable inclou a el codi font i a més a més altres ítems que poden ajudar a fer el programa més extens o comprensible. Un exemple d'arxius que es poden trobar dintre d'un executable poden ser imatges o arxius en llenguatge de programació XML que no executen cap acció sinó que donen una interfície gràfica que pot ajudar a l'usuari que executi el programa (un exemple pot ser el arxiu icon.png de la imatge 1.2)

Aquests arxius addicionals es solen ficar en carpetes per a diferenciar-ho d'altres arxius i per a tindre un executable organitzat (com es pot apreciar en la imatge 1.2, en la el·lipse de color vermell).



1.2 Exemple d'executable F ont de la imatge: pròpia, codi: aliè Font de Simple-calculator: <https://github.com/SimpleMobileTools/Simple-Calculator>

**1.2.3 Definició d'aplicació**

Per a poder explicar el terme aplicació, s'ha de tornar a explicar la definició de programa per a poder entendre completament certes diferencies.

Tal i com s'ha mencionat anteriorment, un programa es un conjunt d'instruccions escrites llenguatge de programació que un aparell interpreta per a resoldre un problema o una funció específica.

Una aplicació no és més que un estil de programa dissenyat únicament per facilitar al usuari la realització d'un determinat treball. Això marca una gran diferencia amb altres programes que estan pensats per realitzar treballs mes avançats i que no afecten de manera comú al usuari (com podrien ser per exemple programes que fan funcionar un sistema operatiu).

Un exemple d'aplicació pot ser un programa que fa el treball de calcular les notes finals dels treballs de recerca a partir de cert percentatges donats.

**1.3 Diferents tipus de llenguatge de programació i la seva utilitat:**

Tipus de llenguatges de programació segons la seva dificultat:

.- Llenguatges de programació de nivell baix: Es programen tenint en compte les característiques del processador al qual se li donen les ordres. Solen ser ordres lògiques i no tenen molta complexitat.

Exemples de llenguatges de programació de nivell baix: Basic, python o bash.

.- Llenguatges de programació de nivell mitjà: Són llenguatges de programació que ajuden als llenguatges de programació de nivell alt. Són molt útils implementant algoritmes, taules de codis hash, etc.

Exemples de llenguatges de programació de nivell mitjà: C o Java-script.

.-Llenguatges de programació de nivell alt: Són els llenguatges de programació que més s'aproximen a el llenguatge dels éssers humans. Això vol dir que a vegades ignoren el funcionament de la màquina. Són molt utilitzats en Intel·ligència Artificial.

Exemples de llenguatge de programació de nivell alt: Java, Ruby o Kotlin.

**1.4 Tipus de llicències en els llenguatges de programació i la importància d'aquestes:**

Per a poder explicar les llicències en els llenguatges de programació, s'ha de entendre primer el significat de la paraula llicència. Segons l' IEC, la paraula llicència té el següent significat: "Llibertat de fer o de dir alguna cosa en virtut d’una permissió donada." Per tant, si apliquem la paraula llicència al món de la programació, trobem que el significat variarà, ja que una llicència en el món de programació i la informàtica és un contracte entre el creador d'un programa, software o un llenguatge de programació i un programador o un usuari comú. Dintre d'aquesta llicència, que ha d'existir, en podem trobar de diferents tipus, les llicències de software lliure (on el "contracte" explicat anteriorment seria el idoni per a l'usuari o programador) i les llicències de software propietari (on el " contracte explicat anteriorment, seria idoni per a els creadors del llenguatge de programació).

**1.4.1 Les Llicències de Software Lliure:**

Les llicències de software lliure, tal i com s'han comentat abans, son aquelles llicències les quals el seu "contracte entre el usuari/programador i els creadors del llenguatge de programació o software beneficien sobretot als usuaris. Aquest benefici sorgeix dels següents motius:

.- El codi del llenguatge/software és accessible per tothom.

.- No té cost per utilitzar ell llenguatge o software

.- Dona total llibertat a l'usuari.

.- No requereix actualitzacions.

.- Es pot trobar orientacions a fòrums, blogs i wikis ( que solen ser més properes a els usuaris).

Un exemple de llicència de Software lliure és la llicencia anomenada GNU LGPL , què és la que utilitzen la majoria de llenguatges de programació lliures, com ara Java.



Llicència GNU i la seva col·laboració amb Java Font: Pròpia

**1.4.1 Les Llicències de Software Propietari:**

Les llicències de software propietari, com ja s'havia introduït anteriorment, són les llicències que beneficien sobretot a els creadors d'un llenguatge de programació o software. Els motius d'aquest benefici són els següents:

.- El usuari no té accés al codi.

.- Requereix actualitzacions

.- Té un determinat cost

.- Limita l'ús de l'usuari

.- Orientació únicament mitjançant pàgines oficials y manuals autoritzats.

Un exemple de software propietari pot ser Windows, que compleix tots els punts anomenats anteriorment.

**2. Introducció a Java:**

**2.1 Definició i característiques del llenguatge de programació Java:**

Si ens plantegem definir que es Java, podem dir que Java és un llenguatge de programació multi plataforma, amb un propòsit general orientat a objectes.

En una única frase, s'han pogut introduir tres nous conceptes: el llenguatge de programació multi plataforma , un propòsit general i la programació orientada a objectes. Ara s'explicaran breument aquests conceptes i després a altres apartats del 2.1 els explicarem més detalladament.

.- El llenguatge de programació multi plataforma: És aquell llenguatge de programació on el seu codi pot ser executat en qualsevol màquina mitjançant un compilador.

.-Llenguatge de propòsit general: Són aquells llenguatges de programació que poden ser utilitzats per a diferents propòsits, com podrien ser la creació d'aplicacions o les bases de dades fins a càlculs matemàtics.

.-Programació orientada a objectes: Es una nova filosofia de programar, que canvia el tipus de programació lògica de les màquines a una altra més adaptada al pensament humà.

**2.1.2 Programació orientada a objectes:**

La programació orientada a objectes, tal y com s'ha introduït en el apartat 2.1, es una filosofia/forma de programar que consisteix en apropar el llenguatge de programació a un llenguatge més humà , però per entendre bé aquesta manera de programar, caldrà analitzar primer en què consisteix i comparar-la amb la seva antecessora, la lògica.

Per a començar l'explicació d'aquesta filosofia, començarem amb un exemple de com solucionaria un problema un llenguatge de programació el qual no utilitzés la Programació Orientada a Objectes. Un exemple pot ser un programa que estigui compilat amb el llenguatge basic, que serveixi per a solucionar equacions de segon grau. En aquest programa, el codi estaria descompost en problemes que anirien ordenats per al final aconseguir resoldre un problema senzill final. Per a entendre-ho millor, el codi faria el següent:

1.- Primer, el codi font demana els coeficients dels valors a, b y c al usuari y els hi assignarà una lletra a cadascuna.

2.- Després, gracies a la fórmula de les equacions de segon grau que tindrà guardada a una constant del codi font, el problema resoldrà la arrel quadrada per a veure si té 2,1 o cap solució.

3.- Per finalitzar, calcularà l'operació restant i ho traurà per pantalla.

En canvi, un llenguatge de programació orientat a objectes trenca aquest esquema lògic, per a invertir-se en un sistema on cada objecte té definits uns paràmetres i els programadors intenten adaptar l'entorn a aquests.

Exemple de codi de la programació a objectes:

**2.2 Funcionament de Java:**

**Història de Java:**