
**Universidad Nacional Autónoma de México**
**Facultad de Ingeniería**
**Ingeniería en computación**


# Materia: Fundamentos de programación


## Tarea 2- Historia de la programación

**Alumno: Ángel Joel Flores Torres**

**Fecha: 04/10/2020**

# BREVE HISTORIA DEL CÓMPUTO

Ya desde la antigüedad se buscaba la forma de calcular con rapidez. La palabra cálculo significa piedrecitas por el tipo de ábaco que se usaba en roma para sumar y restar. Luego apareció el ábaco oriental, sin embargo, después de un tiempo ya no daba el ancho porque el mundo se fue complicando y las cuentas junto con él. En 1642 Blaise Pascal inventa una máquina que suma, resta y guarda los resultados. De ella surgieron otras máquinas que también sumaban y dividían.

191 años después se piensa en una computadora casi automática, fue en 1833 cuando Charles Babbage ideó un equipo gigantesco movido por vapor que recibía los datos con cartones perforados y aunque este proyecto nunca se realizó, aún estaban presentes sus ideas.

El tiempo pasó y el progreso hizo que el hombre además de calcular, necesitara guardar mucha información, por eso el ingeniero Herman Hollerith inventó una máquina para censar la población de Estados Unidos de América en 1890, funcionaba con electricidad y tarjetas perforadas. Más de medio siglo después entra en escena la primera computadora electromecánica, creada por Howard H. Aiken que pesaba dos toneladas.

Luego, en 1946 John Von Neuman inventa la primera computadora programable, la que podía memorizar datos, e instrucciones y utiliza por primera vez en lugar del sistema decimal, el binario, pero fue la UNIVAC 1 construida en 1951 una de las primeras computadoras como tal, estas eran caras, lentas, enormes y pesadas.

En 1958 se inventa el transistor. Unos años después se crea el CHIP, luego estos se unieron a otras piezas diminutas y surgieron los microprocesadores, con más memoria, velocidad y capacidad que alguna de las primeras y enormes computadoras.

En cuanto a las tarjetas perforadas, primero fueron sustituidas por las cintas magnéticas y luego por discos y disquetes. Gracias a todos estos avances y el desarrollo de los programas y software, ahora el hombre posee un equipo capaz de calcular, procesar información, así como escribir, dibujar, hacer música y muchas cosas más. Surge así una nueva ciencia llamada “Informática” y con el internet, una nueva era de comunicación entre los hombres.

## HISTORIA DE LA PROGRAMACIÓN

La historia de la Programación está relacionada directamente con la aparición de los computadores.

Lady Ada Countess of Lovelace fue la primera persona que incursionó en la programación y de quien proviene el nombre del lenguaje de programación ADA.

Los primeros programadores realizaban sus programas mediante tarjetas perforadas, esta idea se basó en el telar de Joseph Marie Jacquard en la época de la revolución industrial, donde se empezaban a automatizar procesos y en específico donde tuvo gran importancia la industria textil. Este telar hacía uso de unas tarjetas perforadas que servían para dibujar patrones sobre las telas, lo que en conjunto formarían los motivos que finalmente quedarían en las telas. En la computación estas tarjetas perforadas se usaban para traducir información e instrucciones en las computadoras de sistema binario, donde se dice que los orificios representaban al 0s, mientras que los lugares que carecían de los orificios eran interpretados como el de 1s, con estas tarjetas se cargaba información a la computadora, la cual era muy diferente a la que conocemos hoy en día, pues era tan grande que podía ocupar una habitación entera. En este momento no existía un lenguaje de programación propiamente dicho, solo se utilizaba el lenguaje binario, el cual no era más que un sistema de códigos representados por ceros y unos que la computadora interpreta como instrucciones. Programar en binario era una tarea muy tediosa, por lo cual los científicos de ese entonces decidieron fabricar un diccionario, en este diccionario se ponía la dirección u operación en binario, seguido

de lo que representaba este número, a este lenguaje se le conoció como lenguaje ensamblador.

Lo anterior simplificó un poco la tarea de programar, sin embargo, seguía siendo una actividad tediosa y es por esto que surgen lenguajes de alto nivel, que daban las órdenes a seguir, este leguaje fue llamado “código fuente” y básicamente, conjunto de líneas de texto con los pasos que debe seguir la computadora para ejecutar un programa este proceso puede ser un compilador.
Un compilador lee todas las instrucciones y genera un resultado; un intérprete ejecuta y genera resultados línea a línea. En cualquier caso, han aparecido nuevos lenguajes de programación, unos denominados estructurados y en la actualidad en cambio los lenguajes orientados a objetos y los lenguajes orientados a eventos.

## LISTA DE LOS LENGUAJES DE PROGRAMACIÓN

Un lenguaje de programación es un lenguaje formal que consiste en todos los símbolos, caracteres y reglas de uso que permiten a las personas “comunicarse” con las computadoras y expresar los procesos que debe ejecutar la máquina. Cada lenguaje de programación tiene su propia sintaxis y su propio vocabulario

Actualmente existen muchos lenguajes de programación diferentes. Algunos son creados con un fin en específico, mientras que otros son herramientas de uso general más flexibles que son apropiadas para muchos tipos de aplicaciones.

A pesar de existir cientos de lenguajes de programación no todos estos son populares, al decir populares nos referimos a que cumpla ciertos requisitos para ser considerado de esa forma, para ello debe existir una gran cantidad de software actual que haga uso de estos lenguajes de programación, que sea utilizado por un gran número de personas, que exista una comunidad activa y un importante mercado de trabajo.

Dentro de los lenguajes de programación más populares se encuentran C, C++, C#, Java, JavaScript, Perl, PHP, Python, Objetive-C, Ruby y Visual Basic.

## A

- A# .NET
- A# (Axiom)
- A-0 System
- A+
- A++
- ABAP
- ABC
- ABC ALGOL
- ABLE
- ABEL
- ABSET
- ABSYS
- Abundance
- ACC
- Accent
- Ace DASL
- ACT-III
- Action!
- ActionScript
- Ada
- Adenine
- Agda
- Agora
- AIMMS
- Alef
- ALF
- ALGOL 58
- ALGOL 60
- ALGOL 68
- Alice
- Alma-0
- AmbientTalk
- Amiga E
- AMOS
- AMPL
- APL
- AppleScript
- Arc
- Arden Syntax[1]
- ARexx
- Argus
- AspectJ
- ASP.NET
- Assembly language
- ATS
- Ateji PX
- AutoHotkey
- Autocoder
- AutoIt
- Averest
- AWK
- Axum

## B

- B
- Babbage
- Bash
- BASIC
- bc
- BCPL
- BeanShell
- Bertrand
- BETA
- Bigwig
- Bistro
- BitC
- BLISS
- Blue
- Boo
- Boomerang
- Bourne shell (incluye bash y ksh)
- B.R.E.W.
- Brainfuck
- BPEL
- BUGSYS

- Batch (Windows/Dos)
- Bon
- BuildProfessional

## C

- C
- C--
- C++ - ISO/IEC 14882
- C# - ISO/IEC 23270
- C/AL
- Caché ObjectScript
- C Shell
- Caml
- Candle
- Cayenne
- CDuce
- Cecil
- Cel
- Cesil
- Ceylon
- CFML
- Cg
- Chapel
- CHAIN
- Charity
- Charm
- Chef
- CHILL
- CHIP-8
- chomski
- Chrome (Ahora Oxygene)
- ChucK
- CICS
- CIL
- Cilk
- CL (IBM)
- Claire
- Clarion
- Clean
- Clipper
- CLIST
- Clojure
- CLU
- CMS-2
- COBOL - ISO/IEC 1989
- Cobra
- CODE
- CoffeeScript
- Cola
- ColdC
- ColdFusion
- Cool
- COMAL
- Common Lisp (también conocido como CL)
- COMPASS
- Component Pascal
- COMIT
- Constraint Handling Rules (CHR)
- Converge
- CORAL66
- Corn
- CorVision
- Coq
- COWSEL
- CPL
- csh
- CSP
- Csound
- Curl
- Curry
- Cyclone
- Cython

## D

- D
- DaVinci Concurrente
- Datalog
- DATATRIEVE
- DinkC
- DIBOL

- DASL (*Datapoint's Advanced Systems Language*)
- DASL
- Dart
- DataFlex
- DAVID SASTRE
- dBase
- DC
- DCL
- Deesel (formalmente G)
- Delphi
- DL/I
- Draco
- Dylan
- DYNAMO

## E

- E
- E#
- Ease
- Easy PL/I
- EASYTRIEVE PLUS
- ECMAScript
- Edinburgh IMP
- EGL
- Eiffel
- ELAN
- Emacs Lisp
- Emerald
- Epigram
- Erlang
- es
- Escapade
- Escher
- ESPOL
- Esterel
- Etoys
- Euclid
- Euler
- EUPHORIA
- EusLisp Robot Lenguaje de programación
- CMS EXEC
- EXEC 2
- EXCEL

## F

- F
- F#
- Factor
- Falcon
- Fancy
- Fantom
- FAUST
- Felix
- Ferite
- FFP
- Fjölnir
- FL
- Flavors
- Flex
- FLOW-MATIC
- FOCAL
- FOCUS
- FOIL
- FORMAC
- @Formula
- Forth
- Fortran - ISO/IEC 1539
- Fortress
- FoxBase
- FoxPro
- FP
- FPr
- Franz Lisp
- Frink
- F-Script
- FSProg
- Fuxi

## G

- G
- Gambas
- Game Maker Language
- GameMonkey Script
- GarGar
- GAMS
- GAP
- G-code
- GEN, lenguaje Inteligencia artificial.
- Genie
- GDL
- Gibiane
- GJ
- GEORGE (lenguaje de programación)
- GLSL
- GNU E
- Go
- Go!
- GOAL
- Gödel
- Godiva
- Goo
- GOTRAN
- GPSS
- GraphTalk
- GRASS
- Groovy

## H

- Hack (lenguaje de programación)
- HAL/S
- Hamilton C shell
- Harbour
- Haskell
- Haxe
- High Level Assembly
- HLSL
- Hop
- Hope
- Hugo
- Hume
- HyperTalk

## I

- IBM Basic assembly language
- IBM HAScript
- IBM Informix-4GL
- ICI
- Icon
- Id
- IDL
- IMP
- Inform
- Io
- Ioke
- IPL
- IPTSCRAE
- ISLISP
- ISPF
- ISWIM

## J

- J
- J#
- J++
- JADE
- Jako
- JAL
- Janus
- JASS
- Java
- JavaScript
- JCL
- JEAN
- Join Java
- JOSS
- Joule
- JOVIAL
- Joy
- JScript
- JavaFX Script
- Julia

## K

- K
- Kaleidoscope
- Karel
- Karel++
- Kaya
- KEE
- KIF
- KRC
- KRL
- KRL (KUKA Lenguaje Robot)
- KRYPTON
- ksh
- Kotlin

## L

- L
- L# .NET
- LabVIEW
- Ladder
- Lagoona
- LANSA
- Lasso
- LaTeX
- Lava
- LC-3
- Leadwerks Script
- Leda
- Legoscript
- LIL
- LilyPond
- Limbo
- Limnor
- LINC
- Lingo
- Linoleum
- LIS
- LISA
- Lisaac
- Lisp - ISO/IEC 13816
- Lite-C
- Lithe
- Little b
- Logo
- Logtalk
- LPC
- LSE
- LSL
- LiveCode
- Lua
- Lucid
- Lustre
- LYaPAS
- Lynx

# M

- M
- M2001
- M4
- Machine code
- MAD (Michigan Algorithm Decoder)
- MAD/I
- Magik
- Magma
- make
- Maple
- MAPPER (Unisys/Sperry) ahora parte de BIS
- MARK-IV (Sterling/Informatics)
- Mary
- MASM Microsoft Assembly x86
- Mathematica
- MATLAB
- Maxima (ver también Macsyma)
- Max (Max Msp - Entorno de programación gráfico)
- MaxScript lenguaje interno de 3D Studio Max
- Maya (MEL)
- MDL
- Mercury
- Mesa
- Metacard
- Metafont
- MetaL
- MetaQuotes Language (MQL4/MQL5)
- Microcode
- MicroScript
- MIIS
- MillScript
- MIMIC
- Mirah
- Miranda
- MIVA Script
- ML
- Moby
- Model 204
- Modelica
- Modula
- Modula-2
- Modula-3
- Mohol
- Monkey X
- MOO
- Mortran
- Mouse
- MPD
- MSIL - nombre deprecado por CIL
- MSL
- MUMPS

# N

- Napier88
- NASM
- NATURAL
- Neko
- Nemerle
- NESL
- Net.Data
- NetLogo
- NetRexx
- NewLISP
- NEWP
- Newspeak
- NewtonScript
- NGL
- Nial
- Nice
- Nickle
- Nodejs
- NPL
- Not eXactly C (NXC)
- Not Quite C (NQC)
- Nu
- NSIS
- NoSQL (NoSQL)

## O

- o:XML
- Oak
- Oberon
- Obix
- OBJ2
- Object Lisp
- ObjectLOGO
- Object REXX
- Object Pascal
- Objective-C
- Objective-J
- Obliq
- Obol
- OCaml
- occam
- occam-π
- Octave
- OmniMark
- Onyx
- Opa
- Opal
- OpenEdge ABL
- OPL
- OPS5
- OptimJ
- Orc
- ORCA/Modula-2
- Oriel
- Orwell
- Oxygene
- Oz

## P

- P#
- PARI/GP
- Pascal - ISO 7185
- Pauscal en español
- Pawn
- PCASTL
- PCF
- PEARL
- PeopleCode
- Perl
- PDL
- PHP
- Phrogram
- Pico
- Pict
- Pike
- Pizza
- PL-11
- PL/0
- PL/B
- PL/C
- PL/I - ISO 6160
- PL/M
- PL/P
- PL/SQL
- PL360
- PLANC
- Plankalkül
- PLEX
- PLEXIL
- Plus
- POP-11
- PostScript
- PortablE
- Powerhouse
- PowerBuilder
- PowerShell
- PPL
- Processing
- Processing.js
- Prograph
- PROIV
- Prolog
- Visual Prolog
- Promela
- PROTEL
- ProvideX
- Pro*C

- PIKT
- PILOT
- Pure
- Python

## Q

- Q (lenguaje de programación ecuacional)
- Q (lenguaje de programación de Kx Systems)
- QBasic
- Qi
- QtScript
- QuakeC
- QPL

## R

- R
- R++
- Racket
- RAPID
- Rapira
- Ratfiv
- Ratfor
- rc
- Realbasic
- REBOL
- Redcode
- REFAL
- Reia
- Revolution
- rex
- REXX
- Rlab
- ROOP
- RPG
- RPL
- RSL
- RTL/2
- Ruby
- Rust

## S

- S
- S2
- S3
- S-Lang
- mIRC scripting
- S-PLUS
- SA-C
- SabreTalk
- SAIL
- SALSA
- SAM76
- SAS
- Sed
- Seed7
- Segismundo
- Self
- SenseTalk
- SETL
- Shift Script
- SiMPLE
- SIMPOL
- SIMSCRIPT
- Simula
- Simulink
- SOL
- Span
- SPARK
- SPIN
- SP/k
- SPS
- Squeak
- Squirrel
- SQL
- SR
- S/SL
- Starlogo

- SASL
- Sather
- Sawzall
- SBL
- Scala
- Scheme
- Scilab
- Script.NET
- SISAL
- SLIP
- SMALL
- Smalltalk
- Small Basic
- SML
- SNOBOL(SPITBOL)
- Snowball
- Swift
- Strand
- STATA
- Stateflow
- Subtext
- Suneido
- SuperCollider
- SuperTalk
- SYMPL
- SyncCharts
- SystemVerilog

## T

- T
- TACL
- TACPOL
- TADS
- TAL
- Tcl
- Tea
- TECO
- TELCOMP
- TeX
- TEX
- TIE
- Timber
- TMG
- Tom
- TOM
- Topspeed
- TPU
- Trac
- T-SQL
- TTCN
- Turing
- TUTOR
- TXL
- TypeScript

## U

- Ubercode
- UCSD Pascal
- Unicon
- Uniface
- UNITY
- Unix shell
- UnrealScript

## V

- Vala
- VBA
- Visual Basic .NET
- Visual C#
- Visual J++
- Visual J#

- VBScript
- Verilog
- VHDL
- Visual Basic
- Visual DataFlex
- Visual DialogScript
- Visual Fortran
- Visual FoxPro
- Visual Objects
- VSXu
- Vvvv

## W

- WATFIV, WATFOR
- WebDNA
- WebQL
- Windows PowerShell
- Winbatch
- Wollok

## X

- X++
- X10
- XBL
- XC (aprovecha XMOS architecture)
- xHarbour
- XL
- XOTcl
- XPL
- XPL0
- XQuery
- XSB
- XSLT - Ver XPath

## Y

- Yorick
- YQL

## Z

- Z notation
- Zeno
- ZOPL
- ZPL

# CLASIFICACIÓN DE LOS LENGUAJES DE PROGRAMACIÓN

Clasificar nunca es fácil puesto que requiere establecer unos criterios concretos que permitan dividir en clases o tipos el conjunto de elementos a clasificar, en especial cuando nos referimos a los lenguajes de programación, ya que existen varias clasificaciones de los lenguajes de programación en función de diferentes criterios, estos pueden clasificarse de la siguiente manera: Clasificación histórica, clasificación según su nivel, clasificación por propósito, clasificación según su orientación e interpretados y compilados.

Clasificación según su nivel:

1. **Lenguaje máquina:** Es el lenguaje de programación que entiende directamente la computadora. Este lenguaje de programación utiliza el alfabeto binario, es decir, el 0 y el 1. El lenguaje máquina es específico de cada máquina o arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre ellas. Este lenguaje de programación dejo de utilizarse por su gran dificultad y por la facilidad para cometer errores al escribir las cadenas binarias.

2. **Lenguajes de bajo nivel**: Un lenguaje de programación de bajo nivel es el que proporciona poca o ninguna abstracción del microprocesador de una computadora. Consecuentemente, su trasladado al lenguaje máquina es fácil. Son mucho más fáciles de utilizar que el lenguaje máquina, pero dependen mucho de la máquina o computadora como sucedía con el lenguaje máquina. El lenguaje ensamblador fue el primer lenguaje de programación que trato de sustituir el lenguaje máquina por otro lenguaje que fuese más parecido al de los seres humanos.

3. **Lenguaje de alto nivel:** Los lenguajes de programación de alto nivel se caracterizan porque su estructura semántica es muy similar a la forma como escriben los humanos, lo que permite codificar los algoritmos de manera más

natural, en lugar de codificarlos en el lenguaje binario de las máquinas, o a nivel de lenguaje ensamblador. Este tipo de lenguajes de programación son independientes de la máquina, los podemos usar en cualquier computador con muy pocas modificaciones o sin ellas, pero precisan de un programa interprete o compilador que traduzca este lenguaje de programación de alto nivel a uno de bajo nivel como el lenguaje de máquina que la computadora pueda entender.

Una clasificación muy extendida desde el punto de vista de trabajar de los programas y la filosofía de su creación es la siguiente:

- **Lenguajes imperativos:** Emplean instrucciones como unidad de trabajo de los programas

- **Lenguajes declarativos:** Los programas se construyen mediante descripciones de funciones o expresiones lógicas

- **Lenguajes orientados a objetos**: El diseño de los programas se basa más en los datos y su estructura. La unidad de proceso es el objeto y en él se incluyen los datos (variables) y operaciones que actúan sobre ellos

- **Lenguajes orientados al problema:** Están diseñados para problemas específicos, principalmente de gestión; suelen ser generadores de aplicaciones.

- **Lenguajes naturales:** Están desarrollándose nuevos lenguajes con el objetivo de aproximar el diseño y construcción de programas al lenguaje de las personas.

Otra clasificación de los lenguajes de programación, es teniendo en cuenta el desarrollo de las computadoras según sus diferentes generaciones:

1. **Primera generación:** Lenguaje máquina y ensamblador.

2. **Segunda generación:** Los primeros lenguajes de programación de alto nivel imperativo (FROTRAN, COBOL).

3. **Tercera generació**n: Son lenguajes de programación de alto nivel imperativo, pero mucho más utilizados y vigentes en la actualidad (ALGOL 8, PL/I, PASCAL, MODULA).

4. **Cuarta generación:** Usados en aplicaciones de gestión y manejo de bases de datos (NATURAL, SQL).

5. **Quinta generación:** Creados para la inteligencia artificial y para el procesamiento de lenguajes naturales (LISP, PROLOG).

Según el propósito del lenguaje:

- **Lenguajes de propósito general:** Como su nombre lo dice no se enfoca en un solo ámbito como Pascal o C

- **Lenguajes de propósito específico:** Se centra en una sola área, ejemplo, php orientado al desarrollo de aplicaciones Web

Otra gran clasificación importante se da entre los lenguajes interpretados y los lenguajes compilados.

- **Lenguajes interpretados**: La máquina solo entiende el lenguaje binario. Los lenguajes interpretados son aquellos que por definición no están escritos en código binario y que requieren de un programa auxiliar (el intérprete), que traduce el lenguaje para que la máquina lo pueda procesar y ejecutar.

- **Lenguajes compilados**: En los lenguajes compilados, un programa anexo llamado compilador hace el proceso de transformación a un lenguaje inteligible por la máquina, antes de la finalización del programa. El archivo resultante se puede ejecutar sin la necesidad de ningún otro programa intermediario, es lo que se denomina archivo ejecutable.

Los lenguajes compilados tienen la gran ventaja de no necesitar de ningún apoyo externo para ejecutarse; no obstante, son menos flexibles, puesto que cada modificación en el código original, o en las fuentes del programa, implica necesariamente una recompilación del programa para aplicar los cambios.

## TIPOS DE PARADIGMAS DE PROGRAMACIÓN

Un paradigma de programación es un modelo básico de diseño y desarrollo de programas, que permite producir programas con un conjunto de normas específicas, tales como: estructura modular, alta rentabilidad, etc.
Los paradigmas pueden ser considerados como patrones de pensamiento para la resolución de problemas. Desde luego siempre teniendo en cuenta los lenguajes de programación, según nuestro interés de estudio.

Una de las clasificaciones más útiles es aquella que clasifica los lenguajes según el paradigma de programación. A continuación, se enumeran algunos de los paradigmas más importantes.

**Paradigma imperativo:** Los programas consisten en una sucesión de instrucciones o conjunto de sentencias, como si el programador diera órdenes concretas. El desarrollador describe en el código paso por paso todo lo que hará su programa.

**Programación estructurada:** La programación estructurada es un tipo de programación imperativa donde el flujo de control se define mediante bucles anidados, condicionales y subrutinas, en lugar de a través de GOTO.

**Programación procedimental**: Este paradigma de programación consiste en basarse en un número muy bajo de expresiones repetidas, englobarlas todas en un procedimiento o función y llamarlo cada vez que tenga que ejecutarse.

**Programación modular:** consiste en dividir un programa en módulos o subprogramas con el fin de hacerlo más manejable y legible. Se trata de una evolución de la programación estructurada para resolver problemas de programación más complejos.

**Paradigma declarativo:** Este paradigma no necesita definir algoritmos puesto que describe el problema en lugar de encontrar una solución al mismo. Este paradigma utiliza el principio del razonamiento lógico para responder a las preguntas o cuestiones consultadas.

**Programación orientada a objetos:** En este modelo de paradigma se construyen modelos de objetos que representan elementos (objetos) del problema a resolver, que tienen características y funciones. Permite separar los diferentes componentes de un programa, simplificando así su creación, depuración y posteriores mejoras. La programación orientada a objetos disminuye los errores y promociona la reutilización del código. Es una manera especial de programar, que se acerca de alguna manera a cómo expresaríamos las cosas en la vida real.

**Programación reactiva:** Este Paradigma se basa en escuchar lo que emite un evento o cambios en el flujo de datos, en donde los objetos reaccionan a los valores que reciben de dicho cambio. Las librerías más conocidas son Project Reactor, y RxJava. React/Angular usan RxJs para hacer uso de la programación reactiva.

# BIBLIOGRAFÍA


Uriarte, J. M. (s. f.). — *RESUMEN — HISTORIA DE LA COMPUTADORA*. Características. Recuperado 4 de octubre de 2020, de https://www.caracteristicas.co/historia-de-la-computadora/#ixzz6ZyXaPYF8

*Clasificación de los lenguajes de programación*. (s. f.). La Revista Informática.com. Recuperado 4 de octubre de 2020, de http://www.larevistainformatica.com/clasificacion-de-los-lenguajes-de-programacion.html

UNAM. (s. f.). *Lenguajes de Programación*. Programas CUAED. Recuperado 4 de octubre de 2020, de https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html

Adell, F. (2013, 8 marzo). *Lenguajes de programación: clasificación, tipos y recursos de aprendizaje*. Fundamentos y evolución de la multimedia. http://multimedia.uoc.edu/blogs/fem/es/lenguajes-de-programacion-clasificacion-tipos-y-recursos-de-aprendizaje/

Colaboradores de Wikipedia. (s. f.). *Anexo: Lenguajes de programación*. Wikipedia. Recuperado 4 de octubre de 2020, de https://es.wikipedia.org/wiki/Anexo:Lenguajes_de_programaci%C3%B3n

Sanchez, A. (2017, 28 enero). *Estos son todos los lenguajes de programación que existen hasta la actualidad.* Azul Web. https://www.azulweb.net/estos-todos-los-lenguajes-programacion-existen-la-actualidad/

Cervantes, N., & Pineda, C. (s. f.). *Fundamentos de Programación- Un poco de historia*. Fundamentos de programación. Recuperado 4 de octubre de 2020, de http://www.utn.edu.ec/reduca/programacion/fundamentos/un_poco_de_historia.html

Colaboradores de Wikipedia. (s. f.-b). *Lenguaje de programación*. Wikipedia, la enciclopedia libre. Recuperado 4 de octubre de 2020, de https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n#Clasificaci%C3%B3n_por_paradigmas

Canelo, M. M. (2020, 9 junio). *¿Qué son los paradigmas de programación?* Profile. https://profile.es/blog/que-son-los-paradigmas-de-programacion/