



UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO



FACULTAD DE INGENIERIA

Fundamentos de Programación

Tarea 1 | Proyecto 1 | Ejercicios 1 | Historia de la  
programación

López Martínez Diana

02/09/2020

## BREVE HISTORIA DEL CÓMPUTO

Se dice que el antecesor del computador fue el ábaco, del cual se cree que pudo haber tenido su origen hace cinco mil años y por su funcionalidad fue utilizado hasta mediados del siglo pasado en Oriente Medio y Asia como instrumento de cálculo. En nuestro medio, el ábaco, aún se utiliza en las salas de billar.

Después del ábaco se hicieron múltiples intentos hasta lograr el computador como hoy lo conocemos.

John Napier (1550-1617) un matemático inventó un dispositivo consistente en unos palillos con números impresos que mediante un ingenioso y complicado mecanismo le permitía realizar operaciones de multiplicación y división.

Blaise Pascal (1.623-1.662), filósofo francés, en 1642 presentó una máquina que sumaba y restaba, ésta funcionaba con 8 ruedas giratorias, dos para los decimales y seis para los enteros y que podía manejar números entre 000.000 01 y 999.999 99.

Leibnitz (1646-1716) en 1672 presentó una máquina que podía, además de sumar y restar, multiplicar, dividir y calcular la raíz cuadrada.

Joseph Jacquard (1.752-1.834) utilizó un mecanismo de tarjetas perforadas para controlar el dibujo formado por los hilos de las telas confeccionadas por una máquina de tejer.

Charles Babbage, matemático e inventor inglés, en 1822 diseñó su máquina diferencial para el cálculo de polinomios, que fue utilizada con éxito para el cálculo de tablas de navegación y artillería.

Herman Hollerith, inventó un sistema de cómputo automático para manipular los datos del censo de Estados Unidos en 1880, su máquina funcionaba con tarjetas perforadas en las que mediante agujeros se representaba el sexo, la edad, la raza etc. En 1900 había desarrollado una máquina que podía clasificar 300 tarjetas por minuto, una perforadora de tarjetas y una máquina de cómputo semiautomática.

El 9 de Abril de 1943, John Mauchly y Lieutenant Herman Goldstine recibieron aprobación para adelantar un proyecto de construcción de la primera computadora, llamada ENIAC (Electronic Numerical integrator and Computer). Lo que caracterizaba al ENIAC como a los ordenadores modernos no era simplemente su velocidad de cálculo sino el hecho de que combinando operaciones permitía realizar tareas que antes eran imposibles.

Entre 1939 y 1944 Howard Aiken de la universidad de Harvard en colaboración con IBM desarrolló el Mark 1 también conocido como calculador Automático de Secuencia Controlada. Este podía multiplicar tres números de 8 dígitos en 1 segundo y operaba con números de hasta 23 dígitos.

En 1946 el matemático húngaro John Von Neumann propuso una versión modificada del Eniac a la que le llamó Edvac (Electronic Discrete Variable Automatic Computer) que se construyó en 1952. A diferencia con el ENIAC, esta maquina empleaba aritmética binaria, lo que simplificaba los circuitos electrónicos de cálculo, y trabajaba con programas almacenados.

En 1953 IBM fabricó su primer computador para aplicaciones científicas el 701 y seguidamente 702 y el 705, este último fue un sistema revolucionario, el primero en emplear memorias de núcleos de ferrita. Con esta producción y mediante una adecuada estrategia comercial IBM tomo la delantera en las ventas de tecnología en todo el mundo.

A partir de esta época se siguieron fabricando y comercializando computadores cada vez más sofisticados, evolucionando a tal grado la tecnología de la información, hasta convertirse en lo que es hoy en día.

## **HISTORIA DE LA PROGRAMACION**

La historia de la Programación está relacionada directamente con la aparición de los computadores, que ya desde el siglo XV tuvo sus inicios con la construcción de una máquina que realizaba operaciones básicas y raíces cuadradas (Gottfried Wilheml von Leibniz); aunque en realidad la primera gran influencia hacia la creación de los computadores fue la máquina diferencial para el cálculo de polinomios, proyecto no concluido de Charles Babbage (1793-1871) con el apoyo de Lady Ada Countess of Lovelace (1815-1852), primera persona que incursionó en la programación y de quien proviene el nombre del lenguaje de programación ADA creado por el DoD (Departamento de defensa de Estados Unidos) en la década de 1970.

Luego los avances en las ciencias informáticas han sido muy acelerados, se reemplazó los tubos al vacío por transistores en 1958 y en el mismo año, se sustituyeron por circuitos integrados, y en 1961 se miniaturizaron en chips de silicio. En 1971 apareció el primer microprocesador de Intel; y en 1973 el primer sistema operativo CP/M. El primer computador personal es comercializado por IBM en el año 1980.

De acuerdo a este breve viaje por la historia, la programación está vinculada a la aparición de los computadores, y los lenguajes tuvieron también su evolución. Inicialmente se programaba en código binario, es decir en cadenas de 0s y 1s, que es el lenguaje que entiende directamente el computador, tarea extremadamente difícil; luego se creó el lenguaje ensamblador, que aunque era lo mismo que programar en binario, al estar en letras era más fácil de recordar. Posteriormente aparecieron lenguajes de alto nivel, que en general, utilizan palabras en inglés, para dar las órdenes a seguir, para lo cual utilizan un proceso intermedio entre el lenguaje máquina y el nuevo código llamado código fuente, este proceso puede ser un compilador o un intérprete.

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 PROGRAMACION

A	Abundance	bc
A# .NET	ACC	BCPL
A# (Axiom)	Accent	BeanShell
A-0 System	Ace DASL	Batch (Windows/Dos)
A+	ACT-III	Bertrand
A++	Action!	BETA
ABAP	ActionScript	Bigwig
ABC	Ada	Bistro
ABC ALGOL	Adenine	BitC
ABLE	Babbage	BLISS
ABSET	Bash	Blue
ABSYS	BASIC	Bon

Boo	CFML	COBOL – ISO/IEC 1989
Boomerang	Cg	Cobra
Bourne shell (incluye bash y ksh)	Chapel	CODE
B.R.E.W.	CHAIN	CoffeeScript
Brainfuck	Charity	Cola
BPEL	Charm	ColdC
BUGSYS	Chef	ColdFusion
BuildProfessional	CHILL	Cool
C	CHIP-8	COMAL
	chomski	Common Lisp (también conocido como CL)
C–	Chrome (Ahora Oxygene)	COMPASS
C++ – ISO/IEC 14882	Chuck	Component Pascal
C# – ISO/IEC 23270	CICS	COMIT
C/AL	CIL	Constraint Handling Rules (CHR)
Caché ObjectScript	Cilk	Converge
C Shell	CL (IBM)	CORAL66
Caml	Claire	Corn
Candle	Clarion	CorVision
Cayenne	Clean	Coq
CDuce	Clipper	COWSEL
Cecil	CLIST	CPL
Cel	Clojure	csh
Cesil	CLU	CSP
Ceylon	CMS-2	

Csound	Draco	Euler
Curl	Dylan	EUPHORIA
Curry	DYNAMO	EusLisp Robot Lenguaje de programación
Cyclone	E	CMS EXEC
Cython	E	EXEC 2
D	E#	EXCEL
	Ease	
DaVinci Concurrente	Easy PL/I	F
DASL (Datapoint's Advanced Systems Language)	EASYTRIEVE PLUS	
DASL	ECMAScript	F#
Dart	Edinburgh IMP	Factor
DataFlex	EGL	Falcon
Datalog	Eiffel	Fancy
DATATRIEVE	ELAN	Fantom
DAVID SASTRE	Emacs Lisp	FAUST
dBase	Emerald	Felix
DC	Epigram	Ferite
DCL	Erlang	FFP
	es	Fjölnir
Deesel (formalmente G)	Escapade	FL
Delphi	Escher	Flavors
DinkC	ESPOL	Flex
DIBOL	Esterel	FLOW-MATIC
DL/I	Etoys	FOCAL
	Euclid	FOCUS

FOIL	Genie	Haxe
FORMAC	GDL	High Level Assembly
@Formula	Gibiane	HLSL
Forth	GJ	Hop
Fortran – ISO/IEC 1539	GEORGE (lenguaje de programación)	Hope
Fortress	GLSL	Hugo
FoxBase	GNU E	Hume
FoxPro	Go	HyperTalk
FP	Go!	I
FPr	GOAL	IBM Basic assembly language
Franz Lisp	Gödel	IBM HAScript
Frink	Godiva	IBM Informix-4GL
F-Script	Goo	ICI
FSProg	GOTRAN	Icon
Fuxi	GPSS	Id
	GraphTalk	IDL
G	GRASS	IMP
Gambas	Groovy	Inform
Game Maker Language	H	Io
GameMonkey Script	Hack (lenguaje de programación)	Ioke
GarGar	HAL/S	IPL
GAMS	Hamilton C shell	IPTSCRAE
GAP	Harbour	ISLISP
G-code	Haskell	ISPF

ISWIM	Karel	LIL
J	Karel++	LilyPond
J	Kaya	Limbo
J#	KEE	Limnor
J++	KIF	LINC
JADE	KRC	Lingo
Jako	KRL	Linoleum
JAL	KRL (KUKA Lenguaje Robot)	LIS
Janus	KRYPTON	LISA
JASS	ksh	Lisaac
Java	L	Lisp – ISO/IEC 13816
JavaScript	L	Lite-C
JCL	L# .NET	Lithe
JEAN	LabVIEW	Little b
Join Java	Ladder	Logo
JOSS	Lagoona	Logtalk
Joule	LANSA	LPC
JOVIAL	Lasso	LSE
Joy	LaTeX	LSL
JScript	Lava	LiveCode
JavaFX Script	LC-3	Lua
Julia	Leadwerks Script	Lucid
K	Leda	Lustre
K	Legoscript	LYaPAS
Kaleidoscope		Lynx



M	MaxScript lenguaje interno de 3D Studio	Monkey X
M	Max	MOO
M2001	Maya (MEL)	Mortran
M4	MDL	Mouse
Machine code	Mercury	MPD
MAD (Michigan Algorithm Decoder)	Mesa	MSIL – nombre deprecado por CIL
MAD/I	Metacard	MSL
Magik	Metafont	MUMPS
Magma	MetaL	N
make	Microcode	Napier88
Maple	MicroScript	NASM
MAPPER (Unisys/Sperry) ahora parte de BIS	MIIS	NATURAL
MARK-IV (Sterling/Informatics)	MillScript	Neko
Mary	MIMIC	Nemerle
MASM Microsoft Assembly x86	Mirah	NESL
Mathematica	Miranda	Net.Data
MATLAB	MIVA Script	NetLogo
Maxima (ver también Macsyma)	ML	NetRexx
Max (Max Msp – Entorno de programación gráfico)	Moby	NewLISP
	Model 204	NEWP
	Modelica	Newspeak
	Modula	NewtonScript
	Modula-2	NGL
	Modula-3	Nial
	Mohol	

Nice	occam- $\pi$	PeopleCode
Nickle	Octave	Perl
Nodejs	OmniMark	PDL
NPL	Onyx	PHP
Not eXactly C (NXC)	Opa	Phrogram
Not Quite C (NQC)	Opal	Pico
Nu	OpenEdge ABL	Pict
NSIS	OPL	Pike
NoSQL (NoSQL)	OPS5	PIKT
O	OptimJ	PILOT
o:XML	Orc	Pizza
Oak	ORCA/Modula-2	PL-11
Oberon	Oriel	PL/0
Obix	Orwell	PL/B
OBJ2	Oxygene	PL/C
Object Lisp	Oz	PL/I – ISO 6160
ObjectLOGO	P	PL/M
Object REXX	P#	PL/P
Object Pascal	PARI/GP	PL/SQL
Objective-C	Pascal – ISO 7185	PL360
Objective-J	Pauscal en español	PLANC
Obliq	Pawn	Plankalkül
Obol	PCASTL	PLEX
OCaml	PCF	PLEXIL
occam	PEARL	Plus

POP-11	Qi	RSL
PostScript	QtScript	RTL/2
Portable	QuakeC	Ruby
Powerhouse	QPL	Rust
PowerBuilder	R	S
PowerShell	R	S
PPL	R++	S2
Processing	Racket	S3
Processing.js	RAPID	S-Lang
Prograph	Rapira	mIRC scripting
PROIV	Ratfiv	S-PLUS
Prolog	Ratfor	SA-C
Visual Prolog	rc	SabreTalk
Promela	Realbasic	SAIL
PROTEL	REBOL	SALSA
ProvideX	Redcode	SAM76
Pro*C	REFAL	SAS
Pure	Reia	SASL
Python	Revolution	Sather
Q	rex	Sawzall
Q (lenguaje de programación ecuacional)	REXX	SBL
	Rlab	Scala
Q (lenguaje de programación de Kx Systems)	ROOP	Scheme
	RPG	Scilab
QBasic	RPL	Scratch

Script.NET	SP/k	TECO
Sed	SPS	TELCOMP
Seed7	Squeak	TeX
Self	Squirrel	TEX
SenseTalk	SR	TIE
SETL	S/SL	Timber
Shift Script	Starlogo	TMG
SiMPLE	Strand	Tom
SIMPOL	STATA	TOM
SIMSCRIPT	Stateflow	Topspeed
Simula	Subtext	TPU
Simulink	Suneido	Trac
SISAL	SuperCollider	T-SQL
SLIP	SuperTalk	TTCN
SMALL	SYMPL	Turing
Smalltalk	SyncCharts	TUTOR
Small Basic	SystemVerilog	TXL
SML	T	TypeScript
SNOBOL(SPITBOL)	T	U
Snowball	TACL	Ubercode
Swift	TACPOL	UCSD Pascal
SOL	TADS	Unicon
Span	TAL	Uniface
SPARK	Tcl	UNITY
SPIN	Tea	Unix shell

UnrealScript	Visual Objects	XL
V	VSXu	XOTcl
Vala	Vwv	XPL
VBA	W	XPL0
VBScript	WATFIV, WATFOR	XQuery
Verilog	WebDNA	XSB
VHDL	WebQL	XSLT – Ver XPath
Visual Basic	Windows PowerShell	Y
Visual Basic .NET	Winbatch	Yorick
Visual C#	X	YQL
Visual DataFlex	X++	Z
Visual DialogScript	X10	Z notation
Visual Fortran	XBL	Zeno
Visual FoxPro	XC (aprovecha XMOS architecture)	ZOPL
Visual J++		ZPL
Visual J#	xHarbour	

## CLASIFICACION DE LOS LENGUAJES DE PROGRAMACION

### Lenguaje máquina

Es el sistema de códigos interpretable directamente por un circuito microprogramable, como el microprocesador de una computadora. Este lenguaje se compone de un conjunto de instrucciones que determinan acciones que serán realizadas por la máquina. Y un programa de computadora consiste en una cadena de estas instrucciones de lenguaje de máquina (más los datos). Normalmente estas instrucciones son ejecutadas en secuencia, con eventuales cambios de flujo causados por el propio programa o eventos externos. El lenguaje

máquina es específico de cada máquina o arquitectura de la máquina, aunque el conjunto de instrucciones disponibles pueda ser similar entre ellas.

### 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 traslado al lenguaje máquina es fácil. El término ensamblador (del inglés assembler) se refiere a un tipo de programa informático encargado de traducir un archivo fuente, escrito en un lenguaje ensamblador, a un archivo objeto que contiene código máquina ejecutable directamente por la máquina para la que se ha generado.

### 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.

Otra clasificación se da con base en el desarrollo de los lenguajes desde la aparición de las computadoras, que sigue un cierto paralelismo con las generaciones establecidas en la evolución de las aquéllas.

Otra clasificación se da con base en el desarrollo de los lenguajes desde la aparición de las computadoras, que sigue un cierto paralelismo con las generaciones establecidas en la evolución de las aquéllas.

Primera generación: Lenguajes máquina y ensambladores.

Segunda generación: Primeros lenguajes de alto nivel imperativo.

Tercera generación: Lenguajes de alto nivel imperativo. Son los más utilizados en la actualidad.

Cuarta generación: Orientados básicamente a las aplicaciones de gestión y manejo de bases de datos.

Quinta generación: Orientados a la inteligencia artificial y al procesamiento de los lenguajes naturales.

## TIPOS DE PARADIGMAS DE LA PROGRAMACION

Paradigma imperativo: Son aquellos que facilitan los cálculos por medio de cambios de estado, entendiendo como estado la condición de una memoria de almacenamiento.

Paradigma heurístico: Define un modelo de resolución de problemas en el que se incorpora algún componente heurístico, sobre la base de una representación más apropiada de la estructura del problema, para su resolución con técnicas heurísticas.

Denominadas heurísticas, las cuales proporcionan entre varios cursos de acción uno que presenta visos de ser el más prometedor, pero no garantiza necesariamente el curso de acción más efectivo."

La Programación Heurística implica una forma de modernizar el problema en lo que respecta a la representación de su estructura, estrategias de búsqueda y métodos de resolución, que configuran el Paradigma Heurístico.

Paradigma funcional: Está basado en el modelo matemático de composición funcional. En este modelo, el resultado de un cálculo es la entrada del siguiente, y así sucesivamente hasta que una composición produce el valor deseado.

Paradigma lógico: Los programas construidos un lenguaje lógico están contruidos únicamente por expresiones lógicas, es decir, que son ciertas o falsas, en oposición a un expresión interrogativa (una pregunta) o expresiones imperativas (una orden). Un ejemplo de lenguaje lógico es Prolog (Programación lógica).

Paradigma basado en objetos: La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto).

La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar

## BIBLIOGRAFIA

[https://www.ciberaula.com/cursos/java/historia\\_programacion.php](https://www.ciberaula.com/cursos/java/historia_programacion.php)

[https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod\\_resource/content/1/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html)

<https://www.tokioschool.com/noticias/clasificacion-lenguajes-programacion/>

<https://es.slideshare.net/KatherineParra5/t-5-41728392>

<https://www.azulweb.net/estos-todos-los-lenguajes-programacion-existen-la-actualidad/>