

Universidad Nacional Autónoma de México

Facultad de Ingeniería

Fundamentos de Programación

Tarea #2: Historia de la programación

Guevara Chávez Emiliano

02/10/2020



Breve historia del cómputo

Algunos de los antecedentes de las computadoras fueron los ábacos y la Pascalina inventada por Blaise Pascal (1623-1662), en estas máquinas, los datos se representaban mediante las posiciones de los engranajes, y los datos se introducían manualmente estableciendo dichas posiciones finales de las ruedas.

Pero la primera computadora fue la máquina analítica creada por Charles Babbage, quien fue un profesor matemático en la Universidad de Cambridge del siglo XIX, en 1823 el gobierno Británico lo apoyo para crear el proyecto de una máquina de diferencias, pero Charles Jacquard (francés), un fabricante de tejidos, él había creado un telar que podía reproducir automáticamente patrones de tejidos leyendo la información codificada en patrones de agujeros perforados en tarjetas de papel rígido. Al enterarse de este método Babbage abandonó la máquina de diferencias y se dedicó al proyecto de la máquina analítica que se pudiera programar con tarjetas perforadas para efectuar cualquier cálculo con una precisión de 20 dígitos.

En 1944 se construyó la Mark I en la Universidad de Harvard, esta fue diseñada por un equipo liderado por Howard.H.Aiken. Esta máquina no está considerada como computadora electrónica debido a que no era de propósito general y su funcionamiento estaba basado en dispositivos electromecánicos llamados relevadores.

En 1947 en la Universidad de Pennsylvania se construyó la ENIAC (Electronic Numerical Integrator And Calculator) que fue la primera computadora electrónica, el equipo de diseño fue liderado por los ingenieros John Cauty y John Eckert. Esta máquina ocupaba todo un sótano de la Universidad, tenía más de 18000 tubos de vacío, consumía 200 KW de energía eléctrica y requería todo un sistema de aire acondicionado, pero tenía la capacidad de realizar cinco mil operaciones aritméticas en un segundo. El proyecto, auspiciado por el departamento de Defensa de los Estados Unidos, culminó dos años después, cuando se integró a ese equipo el ingeniero y matemático húngaro John von Neumann (1903 - 1957). Las ideas de von Neumann resultaron tan fundamentales para su desarrollo posterior, que es considerado el padre de las computadoras. La EDVAC (Electronic Discrete Variable Automatic Computer) fue diseñada por este nuevo equipo. Tenía aproximadamente 4000 bulbos y usaba un tipo de memoria basado en tubos llenos de mercurio por donde circulaban señales eléctricas sujetas a retardos.

La idea fundamental de von Neumann fue: permitir que en la memoria coexistan datos con instrucciones, para que entonces la computadora pueda ser programada en un lenguaje, y no por medio de alambres que eléctricamente interconectaban varias secciones de control, como en la ENIAC.

Todo este desarrollo de las computadoras suele divisarse por generaciones y el criterio que se determinó para determinar el cambio de generación no está muy bien definido, pero resulta aparente que deben cumplirse al menos los siguientes requisitos: La forma en que están construidas y la forma en que el ser humano se comunica con ellas

Primera generación

Esta generación se desarrolló durante la época de los 50. Sus características eran:

- *Estaban construidas por medio de tubos de vacío

- *Eran programas de lenguaje de máquina

- *Las computadoras eran muy grandes y costosas

En 1951 aparece la UNIVAC (NIVersAl Computer), esta fue la primera computadora comercial, que disponía de mil palabras de memoria central y podían leer cintas magnéticas.

En las dos primeras generaciones, las unidades de entrada utilizaban tarjetas perforadas, retomadas por Herman Hollerith (1860 - 1929), quien además fundó una compañía que con el paso del tiempo se conocería como IBM (International Bussines Machines).

IBM entre 1953 y 1957 sacó la IBM 701, posteriormente Remington Rand fabricó el modelo 1103, estas dos compitieron por el campo científico, luego IBM desarrolló la IBM 702, pero salió con problemas de memoria, por lo tanto no duró en el mercado. La computadora más exitosa de la primera generación fue la IBM 650. Esta computadora usaba un esquema de memoria secundaria llamado tambor magnético, que es el antecesor de los discos actuales.

Otros modelos de computadora que se pueden situar en los inicios de la segunda generación son: la UNIVAC 80 y 90, las IBM 704 y 709, Burroughs 220 y UNIVAC 1105.

Segunda Generación

Cerca de la década de los 60, las computadoras seguían evolucionando y estas reducían su tamaño, aumentaban su nivel de procesamiento y reducían sus costos. Además en esta época se empezó a desarrollar el sistema con el que se comunican las computadoras y estas reciben el nombre de programación de sistemas.

Las características de las segunda generación son:

- *Están construidas con circuitos de transistores

- *Se programan en nuevos lenguajes llamados lenguajes de alto nivel

Algunas de estas computadoras se programaban con cintas perforadas y otras más por medio de cableado en un tablero. Los programas eran hechos a la medida por un equipo de expertos. El usuario final de la información no tenía contacto directo con las computadoras. Esta situación en un principio se produjo en las primeras computadoras personales, pues se requería saberlas "programar" (alimentarle instrucciones) para obtener resultados; por lo tanto su uso estaba limitado. Además, para no perder el "programa" resultante había que "guardarlo" (almacenarlo) en una grabadora de astte, este procedimiento podía tomar de 10 a 45 minutos.

Aparecen los programas procesadores de palabras como el Wordstar, y la hoja de cálculo (spreadsheet) Visicalc. Aparece el concepto de human interface que es la relación entre el usuario y su computadora, además se empieza a hablar de hardware ergonómico. Se empieza a sentir que la relación usuario-PC no está acorde con los desarrollos del equipo y de la potencia de los programas. Hace falta una relación amistosa entre el usuario y la PC.

Aparecen muchas compañías y las computadoras eran bastante avanzadas para su época como la serie 5000 de Burroughs y la ATLAS de la Universidad de Manchester .Las computadoras de esta generación fueron: la Philco 212 y la UNIVAC M460, la Control Data Corporation modelo 1604, seguida por la serie 3000, la IBM mejoró la 709 y sacó al mercado la 7090, la National Cash Register empezó a producir máquinas para proceso de datos de tipo comercial, introdujo el modelo NCR 315. La Radio Corporation of America introdujo el modelo 501, que manejaba el lenguaje COBOL, para procesos administrativos y comerciales. Después salió al mercado la RCA 601.

Tercera Generación

Con los progresos de la electrónica y los avances de comunicación con las computadoras en la década de los 1960, surge la tercera generación de las computadoras. Se inauguró con la IBM 360 en abril de 1964.

Las características de esta generación fueron las siguientes:

- *Su fabricación electrónica está basada en circuitos integrados.

- *Su manejo es por medio de los lenguajes de control de los sistemas operativos.

IBM en esta generación fue la la empresa más grande del mercado. IBM produjo la serie 360 con los modelos 20, 22, 30, 40, 50, 65, 67, 75, 85, 90, 195 que utilizaban técnicas especiales del procesador, unidades de cinta de nueve canales, paquetes de discos magnéticos y otras características que ahora son estándares (no todos los modelos usaban estas técnicas, sino que estaba dividido por aplicaciones). El sistema operativo de la serie 360, se llamaba OS que contaba con varias configuraciones, incluía un conjunto de técnicas de manejo de memoria y del procesador que pronto se convirtieron en estándares. En la década de 1970, la IBM produce la serie 370 (modelos 115, 125, 135, 145, 158, 168). UNIVAC compite con los modelos 1108 y 1110, máquinas en gran escala y a finales de esta década la IBM de su serie 370 produce los modelos 3031, 3033, 4341.

En 1964 CDC introdujo la serie 6000 con la computadora 6600 que se consideró durante algunos años como la más rápida. En la década de 1970, mientras que CDC produce su serie 7000 con el modelo 7600. Estas computadoras se caracterizan por ser muy potentes y veloces. Burroughs con su serie 6000 produce los modelos 6500 y 6700 de avanzado diseño, que se reemplazaron por su serie 7000. Honey - Well participa con su computadora DPS con varios modelos. A mediados de la década de 1970, aparecen en el mercado las computadoras de tamaño mediano, o minicomputadoras que no son tan costosas como las grandes (llamadas también como mainframes que significa también, gran sistema), pero disponen de gran capacidad de procesamiento. Algunas minicomputadoras fueron las siguientes: la PDP - 8 y la PDP - 11 de Digital Equipment Corporation, la VAX (Virtual Address eXtended) de la misma compañía, los modelos NOVA y ECLIPSE de Data General, la serie 3000 y 9000 de Hewlett - Packard con varios modelos el 36 y el 34, la Wang y Honey - Well -Bull, Siemens de origen alemán, la ICL fabricada en Inglaterra. En la Unión Soviética se utilizó la US (Sistema Unificado, Ryad) que ha pasado por varias generaciones.

Cuarta Generación

Aquí aparecen los microprocesadores que es un gran adelanto de la microelectrónica, son circuitos integrados de alta densidad y con una velocidad impresionante. Las microcomputadoras con base en estos circuitos son extremadamente pequeñas y baratas, por lo que su uso se extiende al mercado industrial. Aquí nacen las computadoras personales que han adquirido proporciones enormes y que han influido en la sociedad en general sobre la llamada "revolución informática". En 1976 Steve Wozniak y Steve Jobs inventan la primera microcomputadora de uso masivo y más tarde forman la compañía conocida como Apple. En 1981 se vendieron 800 00 computadoras personales, al año siguiente subió a 1 400 000. Entre 1984 y 1987 se vendieron alrededor de 60 millones de computadoras personales.

Con el surgimiento de las computadoras personales, el software y los sistemas que con ellas se manejan han tenido un considerable avance, porque han hecho más interactiva la comunicación con el usuario. Surgen otras aplicaciones como los procesadores de palabra, las hojas electrónicas de cálculo, paquetes gráficos, etc. También las industrias del Software de las computadoras personales crecieron con gran rapidez, Gary Kildall y William Gates se dedicaron durante años a la creación de sistemas operativos y métodos para lograr una utilización sencilla de las microcomputadoras (son los creadores de CP/M y de los productos de Microsoft). Aunque las microcomputadoras dominaban el mercado, se seguían produciendo grandes sistemas de computación pero estas estaban en esferas de control gubernamental, militar y de la gran industria. Las enormes computadoras de las series CDC, CRAY, Hitachi o IBM, eran capaces de atender a varios cientos de millones de operaciones por segundo

Quinta Generación

En vista de la acelerada marcha de la microelectrónica, la sociedad industrial se ha dado a la tarea de poner también a esa altura el desarrollo del software y los sistemas con que se manejan las computadoras. Surge la competencia internacional por el dominio del mercado de la computación, en la que se perfilan dos líderes que, sin embargo, no han podido alcanzar el nivel que se desea: la capacidad de comunicarse con la computadora en un lenguaje más cotidiano y no a través de códigos o lenguajes de control especializados. Japón lanzó en 1983 el llamado "programa de la quinta generación de computadoras", con los objetivos explícitos de producir máquinas con innovaciones reales en los criterios mencionados. Y en los Estados Unidos ya está en actividad un programa en desarrollo que persigue objetivos semejantes, que pueden resumirse de la siguiente manera:

- *Procesamiento en paralelo mediante arquitecturas y diseños especiales y circuitos de gran velocidad.

- * Manejo de lenguaje natural y sistemas de inteligencia artificial.

Historia de la programación

Los primeros lenguajes de programación preceden a la computadora moderna. En un inicio los lenguajes eran códigos.

Algunos de los antecedentes de la programación fueron:

En un período de nueve meses entre 1842 y 1843, Ada Lovelance tradujo las memorias del matemático italiano Luigi Menabrea acerca de la nueva máquina propuesta por la máquina de Charles Babbage, la Máquina Analítica. Con estos escritos, ella añadió unas notas en las cuales especificaba en detalle un método para calcular los números de Bernoulli con la utilización de las tarjetas perforadas en esta máquina.

Herman Hollerith se percató de que podía codificar la información en tarjetas perforadas cuando observó a los conductores de trenes que identificaban a los pasajeros según el orificio que hacían en su respectivo ticket. En 1890 Hollerith codificó los datos del censo en tarjetas perforadas.

Los primeros códigos de computadora estaban especializados según sus aplicaciones. En las primeras décadas del siglo 20, los cálculos numéricos estaban basados en los números decimales. Con el paso del tiempo, se dieron cuenta de que la lógica podía ser representada con números, no sólo con palabras. Por ejemplo, Alonzo Church fue capaz de expresar el cálculo lambda a través de fórmulas. La máquina de Turing estableció las bases para almacenar programas como datos en la arquitectura de Von Neuman de una computadora. Sin embargo, a diferencia del cálculo lambda, el código de Turing no serviría satisfactoriamente como base para lenguajes de más alto nivel, su principal uso es en el análisis riguroso en la complejidad algorítmica.

En la década de 1940 fueron creadas las primeras computadoras modernas, con alimentación eléctrica. La velocidad y capacidad de memoria limitadas forzaron a los programadores a escribir programas, en lenguaje ensamblador muy afinados. Finalmente se dieron cuenta de que la programación en lenguaje ensamblador requería de un gran esfuerzo intelectual y era muy propensa a errores.

En 1948, Konrad Zuse publicó un artículo acerca de su lenguaje de programación Plankalkül. Sin embargo, no fue implementado en su vida y sus contribuciones fueron aisladas de otros desarrollos.

Entre algunos lenguajes importantes que fueron desarrollados en este período se encuentran:

*1943 - Plankalkül (Respetad a Conrado), diseñado, pero sin implementar durante medio siglo

*1943 - se crea el sistema de codificación ENIAC

*1949 - 1954 - una serie de conjuntos de instrucciones nemotécnicas, como la de ENIAC, comenzando en 1949 con C-10 para BINAC (que luego evolucionaría en UNIVAC). Cada conjunto de instrucciones estaba destinado a un fabricante específico

En los cincuenta, los tres primeros lenguajes de programación modernos, cuyos descendientes aún continúan siendo utilizados, son:

*FORTRAN (1955) creado por John Backus, este fue el primer lenguaje de alto nivel y surgió como vía de escape de Backus para "trabajar menos" en su tarea de calcular trayectorias de misiles, ya que el lenguaje, y el compilador, eran capaces de hacer el trabajo mucho más rápido.

*COBOL (1959), creado por el Short Range Commitee, altamente influenciado por Grace Hopper. Tras Fortran como primer compilador, llegó COBOL como la culminación de varios otros sistemas de programación. El objetivo era crear un lenguaje de programación universal que pudiera ser usado en cualquier ordenador. Además, permitió modernizar la informática de gestión.

Otro hito a finales de 1950 fue la publicación, por un comité Americano y Europeo de científicos de la computación, de un nuevo "lenguaje para algoritmos"; el Reporte de ALGOL 60 ("ALGOritmic Language"). Este reporte consolidó muchas ideas que estaban circulando en aquel entonces, y proporcionó dos innovaciones importantes para los lenguajes de programación:

*Estructuras de bloques anidadas: las secuencias de código y las declaraciones asociadas se pueden agrupar en bloque sin tener que pertenecer explícitamente a procedimientos separados;

*Ámbito léxico: un bloque puede tener sus propias variables, procedimientos y funciones, invisible al código fuera de dicho bloque, por ejemplo, ocultamiento de información.

Otra innovación, relacionada con esto, fue cómo el lenguaje fue descrito:

*Una notación matemática exacta, Backus-Naur Form (BNF), fue utilizada para describir la sintaxis del lenguaje. Todos los subsecuentes lenguajes de programación han utilizado una variante de BNF para describir la porción libre del contexto de su sintaxis.

El período comprendido entre finales de 1960 y finales de 1970 trajo un gran florecimiento de lenguajes de programación. La mayoría de los paradigmas de lenguajes de programación más importantes y actualmente en uso se inventaron en este período:

*SIMULA, inventado en la década de 1960 por Nygaard y Dahl como un superconjunto de Algol 60, fue el primer lenguaje diseñado para apoyar la programación orientada a objetos.

*C, en principio un lenguaje de programación de sistemas, fue desarrollado por Dennis Ritchie y Ken Thompson en los Laboratorios Bell entre 1969 y 1973.

*SMALLTALK (mediados de los 70) proporcionaron un completo diseño de un lenguaje orientado a objetos.

*PROLOG, diseñado en 1972 por Alain Colmerauer, Philippe Roussel y Robert Kowalski, fue el primer lenguaje de programación lógica.

*ML construyó un sistema de tipos polimórfico (inventado por Robin Milner en 1973) en el tope de Lisp, pionero en los lenguajes funcionales de programación con tipado estático. Cada uno de estos lenguajes generó toda una familia de descendientes, y los lenguajes más modernos cuentan al menos uno de ellos en su ascendencia.

La década de 1980 fueron años de consolidación relativa en los lenguajes imperativos. En vez de inventar nuevos paradigmas, se comenzó a trabajar a partir de las ideas inventadas en la década anterior. C++ combinaba la programación orientada a objetos y la programación de sistemas. El gobierno de Estados Unidos estandarizó Ada, un lenguaje de programación de sistemas destinado a ser utilizado por contratistas de defensa. En Japón y en otras partes, se gastaron enormes sumas investigando los llamados lenguajes de programación de quinta generación que incorporan construcciones de la programación lógica. La comunidad de los lenguajes funcionales llevaron a cabo la estandarización de ML y Lisp. La investigación en Miranda, un lenguaje funcional, con evaluación perezosa, comenzó a tomar fuerza en esta década.

Una nueva tendencia importante en el diseño de lenguajes era un mayor enfoque en la programación de sistemas a gran escala a través del uso de módulos, una organización a gran escala de unidades de código. Modula, Ada, y ML desarrollaron sistemas de módulos notables en la década de 1980. Los sistemas de módulos eran relacionados con frecuencia con construcciones de programación genéricas, que serían, en esencia, módulos parametrizados

Aunque no aparecían nuevos paradigmas de lenguajes de programación imperativos, muchos investigadores extendieron las ideas de los lenguajes anteriores y las adaptaron a nuevos contextos.

La década de 1980 también trajo avances en la implementación de lenguajes de programación. El movimiento de RISC en la arquitectura de computadoras postulaba que el hardware debía estar diseñado para los compiladores más que para los programadores de ensamblador humanos. Con la ayuda de las mejoras en la velocidad del procesador, permitiendo técnicas de compilación cada vez más agresivas, el movimiento RISC despertó un mayor interés en la tecnología de compilación de los lenguajes de alto nivel.

Las tecnologías de los lenguajes continuaron sobre estas líneas entrando en la década de 1990.

El rápido crecimiento de Internet de la década de 1990 fue el siguiente gran acontecimiento histórico para los lenguajes de programación. Con la apertura de una plataforma totalmente nueva para los sistemas informáticos, Internet creó una oportunidad adoptar nuevos lenguajes. En particular, el lenguaje de programación JavaScript se hizo popular debido a su pronta integración con el navegador web Netscape Navigator, y varios lenguajes de scripting alcanzaron un amplio uso en el desarrollo de aplicaciones personalizadas para servidores web. La década de 1990 no vio ninguna novedad fundamental en los lenguajes imperativos, pero sí mucha recombinación y la maduración de viejas ideas. Esta era comenzó la difusión de los

lenguajes funcionales. Una filosofía de conducción grande era la productividad del programador. Surgieron muchos lenguajes de “aplicaciones de desarrollo rápido” (RAD), los cuales usualmente venían con un IDE, recolector de basura, y eran descendientes de lenguajes anteriores. Todos estos lenguajes están orientados al objeto. Estos incluían Object Pascal, Visual Basic y Java, en particular, recibió mucha atención. Pero más radicales e innovadores que los lenguajes de RAD eran los nuevos lenguajes de script. Estos no descendían directamente de otros lenguajes y ofrecieron nuevas sintaxis e incorporación más liberal de otras características. Muchos consideran estos lenguajes de script más productivos que los lenguajes de RAD, aunque esto se debe a menudo a que es más difícil escribir y mantener largos programas que pequeños programas simples.

La evolución de los lenguajes de programación continúa, tanto en la industria como en investigación. Algunas de las tendencias actuales incluyen:

- *Aumentar el soporte para la programación funcional en lenguajes importantes utilizados comercialmente, incluida la programación funcional pura para hacer el código más fácil de razonar y de paralelizar (tanto en macro como en micro-niveles).

- *Construir lenguajes para apoyar la programación concurrente y distribuida.

- *Mecanismos para añadir al lenguaje verificación en cuanto a seguridad y confiabilidad: chequeo sintáctico extendido, control de flujo de información, seguridad de hilos.

- *Desarrollo de software orientado a componentes.

- *metaprogramación, la reflexión o el acceso al árbol de sintaxis abstracta.

- *Mayor énfasis en cuanto a distribución y movilidad.

- *El código abierto como una filosofía de desarrollo de lenguajes, incluyendo la colección de compiladores de GNU y lenguajes recientes, como Python, Ruby, y Squeak .

- *Programación Orientada a Aspectos (AOP).

Lista de los lenguajes de programación

- *Java

- *JavaScript

- *Go

- *Kotlin

- *Python

- *Lenguaje C/C++

- *Scala

- *Ruby

*Swift

*Dart

Clasificación de los lenguajes de programación

Hay tres tipos de lenguaje de programación que se pueden utilizar, las cuales son:

Lenguaje máquina

El lenguaje máquina es el sistema de códigos interpretables directamente por un circuito microprogramable. Este lenguaje se compone de un conjunto de instrucciones que serán realizadas por la computadora. Y un programa de computadora consiste en una cadena de estas instrucciones de lenguaje de máquina. 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.

Lenguaje de programación de bajo nivel

Un lenguaje de programación de bajo nivel es lo que proporciona poca o ninguna abstracción del microprocesador de una computadora. Consecuentemente, su traslado al lenguaje máquina es fácil. El término ensamblador 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 programación 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.

Uno de los problemas que tienen los lenguajes de programación de alto nivel son que hay una gran cantidad de ellos, que además de tener diferentes versiones y dialectos. Es difícil establecer una clasificación general de ellos, ya que cualquiera que se realice habrá lenguajes que pertenezcan a más de uno de los grupos definidos. Estos grupos definidos es una clasificación muy extendida, atendiendo a la forma de trabajar de los programas y la filosofía en la que fueron concebidos. Estos grupos definidos son los siguientes:

*Lenguajes Imperativos: Emplean instrucciones como unidad de trabajo de los programas (Cobol, Pascal, C, Ada).

*Lenguajes declarativos: Los programas se construyen mediante descripciones de funciones lógicas (Lisp, Prolog).

*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 (*Smalltalk*, C++).

*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.

Tipos de paradigmas de programación

Para desarrollar este tema primero se necesita saber que es un paradigma, por lo tanto “Un paradigma de programación indica un método de realizar cálculos y la manera en que se deben estructurar y organizar las tareas que debe llevar a cabo un programa”. Y como hay distintos métodos de realizar y de estructurar estos proyectos, hay distintos tipos de paradigmas los cuales son:

Paradigma Imperativo

*Describe cómo realizar el cálculo, no el porqué

*Un cálculo consiste en una serie de sentencias, ejecutadas según un control de flujo explícito, que modifican el estado del programa.

*Las variables son celdas de memoria que contienen datos (o referencias), pueden ser modificadas, y representan el estado del programa.

*La sentencia principal es la asignación.

*Es el estándar ‘de facto’.

Paradigma Declarativo

*Describe qué se debe calcular, sin explicitar el cómo.

*No existe un orden de evaluación prefijado.

*Las variables son nombres asociados a definiciones, y una vez instanciadas son inmutables

*No existe sentencia de asignación

*El control de flujo suele estar asociado a la composición funcional, la recursividad y/o técnicas de reescritura y unificación.

+Existen distintos grados de pureza en las variantes del paradigma.

+Las principales variantes son los paradigmas funcional, lógico, la programación reactiva y los lenguajes descriptivos.

Programación Funcional

*Basado en los modelos de cómputo cálculo lambda (Lisp, Scheme) y lógica combinatoria (familia ML, Haskell).

*Las funciones son elementos de primer orden

*Evaluación por reducción funcional. Técnicas: recursividad, parámetros acumuladores, CPS, Mónadas.

Familia LISP (Common-Lisp, Scheme):

+Basados en s-expresiones.

+Tipado débil.

+Meta-programación

*Familia ML (Miranda, Haskell, Scala):

+Sistema estricto de tipos (tipado algebraico)

+Concordancia de patrones.

+ Transparencia referencial

+Evaluación perezosa (estruct. de datos infinitas)

Programación Lógica

*Basado en la lógica de predicados de primer orden

*Los programas se componen de hechos, predicados y relaciones.

*Evaluación basada en resolución SLD: unificación + backtracking

*La ejecución consiste en la resolución de un problema de decisión, los resultados se obtienen mediante la instanciación de las variables libres.

*Lenguaje representativo: PROLOG

Programación Reactiva (Dataflow)

*Basado en la teoría de grafos.

*Un programa consiste en la especificación del flujo de datos entre operaciones.

*Las variables se encuentran ligadas a las operaciones que proporcionan sus valores. Un cambio de valor de una variable se propaga a todas las operaciones en que participa.

*Las hojas de cálculo se basan en este modelo.

*Lenguajes representativos: Simulink, LabView.

BIBLIOGRAFÍA

Wikipedia. 2020. *Historia De Los Lenguajes De Programación*. [online] Disponible en: <https://es.wikipedia.org/wiki/Historia_de_los_lenguajes_de_programaci%C3%B3n> [Visto el 4 Octubre 2020].

Vaca Rodríguez, C., 2020. *Paradigmas De Programación*. [ebook] Valladolid, pp.2-9. Disponible en: <<https://www.infor.uva.es/~cvaca/asigs/docpar/intro.pdf>> [Visto 4 Octubre 2020].

Hernández García, G., 2020. *HISTORIA DE LAS COMPUTADORAS*. [ebook] pp.4-21. Disponible: <<https://www.uv.mx/personal/gerhernandez/files/2011/04/historia-compuesta.pdf>> [Visto 4 Octubre 2020].

Monterde, U., 2020. *Lenguajes De Programación*. [online]
Programas.cuaed.unam.mx. Disponible en:

<https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html#introduccion > [Visto 4 Octubre 2020].