

Tarea: Historia de los Lenguajes de Programación

Steven Sequeira Araya

Carnet: 2022080836

Instituto Tecnológico de Costa Rica

Escuela Ingeniería en Computación

Lenguajes de Programación

Contenido

Introducción nueva	5
Línea del tiempo.....	6
Lenguajes actualizados	7
1.Fortran Antes	7
FORTRAN.....	7
Fortran Después	7
FORTRAN.....	7
2.Pascal Antes	8
PASCAL	8
Pascal Después	8
PASCAL	8
3.C++ Antes	9
C++	9
C++ Después	9
C++	9
4.Oberon Antes	10
OBERON.....	10
Oberon Después	10
OBERON.....	10
5.Delphi Antes.....	11
DELPHI	11
Delphi Después.....	11
DELPHI	11
6.Wyvern Antes.....	12
WYVERN	12
Wyvern Después.....	12
WYVERN	12
Lenguajes nuevos.....	13
Microsoft Power Fx.....	13
Carbon	13
Mojo	13
Documento completo reformado	14
Introducción	14
HISTORIA DE LOS LENGUAJES Siglo XX Década de los 50's.....	14
SHORT CODE	14
FORTRAN.....	14
JOVIAL.....	15
LISP	15
HISTORIA DE LOS LENGUAJES Siglo XX Década de los 60's.....	15
COBOL.....	15
ALGOL	16
BASIC	16

PASCAL	16
HISTORIA DE LOS LENGUAJES Siglo XX Década de los 70's.	17
SMALLTALK.....	17
C	17
PROLOG.....	18
ML	18
X10.....	18
ADA.....	18
LOGO.....	19
SMALLTALK V	19
MODULA-2.....	19
C++	19
HISTORIA DE LOS LENGUAJES Siglo XX Década de los 80's.	20
OBERON.....	20
MIRANDA.....	21
EIFFEL	21
FOXPRO	21
ERLANG	22
PERL.....	22
HISTORIA DE LOS LENGUAJES Siglo XX Década de los 90's.	22
HASKELL	22
JAVA	23
PYTHON	23
LUA	24
RUBY	25
PHP.....	26
JAVASCRIPT	27
D	28
DELPHI	29
HISTORIA DE LOS LENGUAJES Primera Década Siglo XXI.	30
LENGUAJES EN PLATAFORMAS .NET	30
PROCESSING	30
C#	31
SCALA	32
VISUALBASIC.NET	32
A++	32
F#.....	33
GROOVY	33
BOO	33
RUBY ON RAILS.....	34
VALA.....	34
CLOJURE	35
GO	35
HISTORIA DE LOS LENGUAJES Segunda Década Siglo XXI.....	36

CHAPEL.....	37
ZIMBU.....	37
RUST.....	37
CRYSTAL.....	38
CEYLON.....	38
DART (originalmente llamado Dash)	39
OPA.....	40
ELIXIR.....	41
RED.....	41
XTEND.....	41
JULIA.....	41
ELM.....	42
TYPESCRIPT.....	43
KOTLIN.....	43
WYVERN.....	44
HACK.....	44
SWIFT.....	45
PIXIE.....	46
ALBATROSS.....	46
RING.....	46
ODEN.....	46
HISTORIA DE LOS LENGUAJES Tercer Década Siglo XXI.	47
MICROSOFT POWER FX Microsoft Power Fx.....	47
CARBON.....	47
MOJO.....	47

Introducción nueva

Desde tiempos inmemorables la creación de nuevos lenguajes de programación ha sido algo constante en el mundo de la computación e informática, sin embargo, ¿cuántos de estos lenguajes se han mantenido hasta el día de hoy a flote?

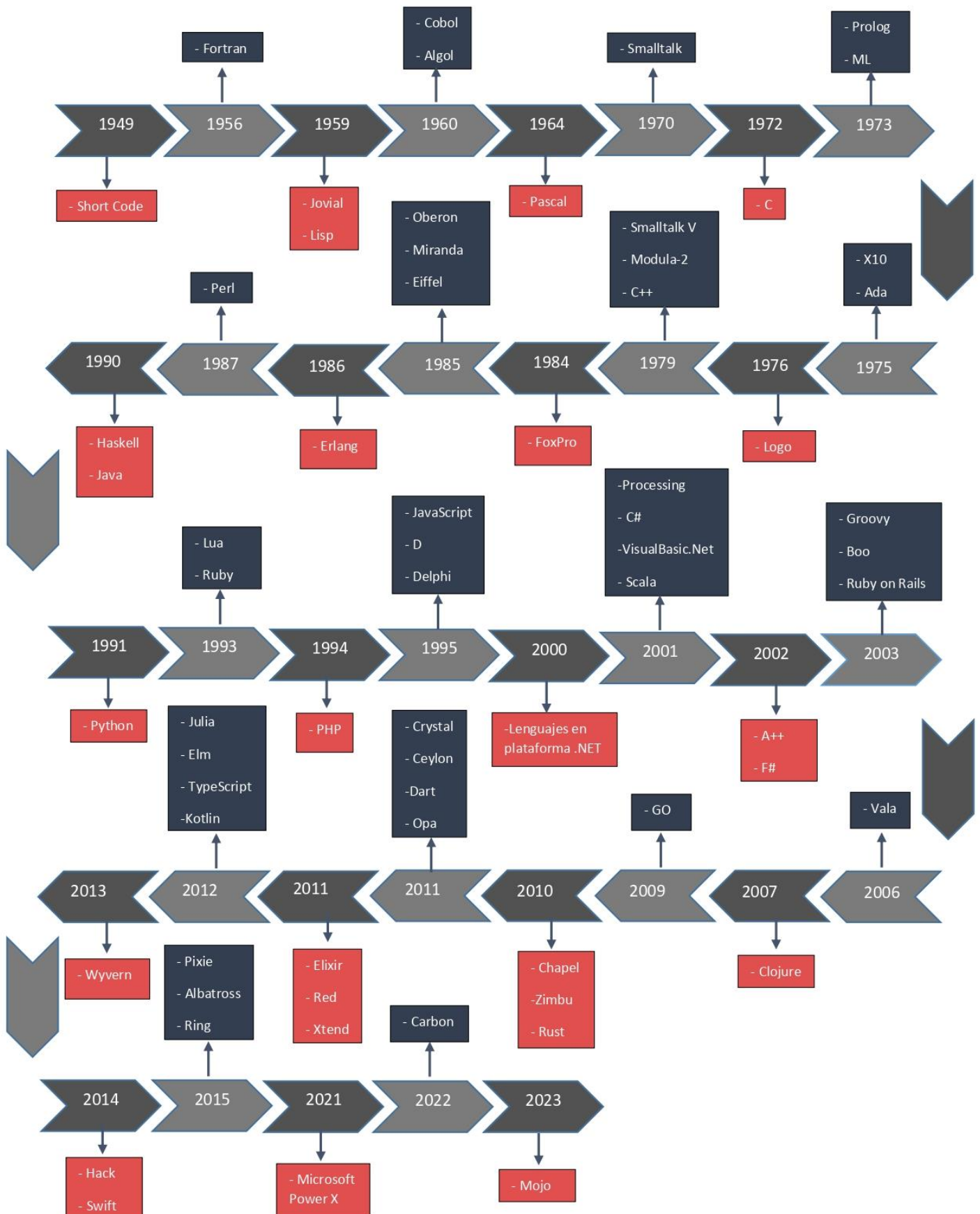
La respuesta es simple, probablemente menos del 1 por ciento de ellos se sigue considerando como un lenguaje de programación vivo. Muchos de ellos alcanzaron la cima y se hundieron con el pasar de los años, otros se quedaron como ideas prometedoras, otros no tuvieron el apoyo económico para alcanzar el éxito, otros se crearon como Hobby sin ambición de crecimiento. Como se puede ver son muchas las posibles razones que pueden explicar el poco éxito que han tenido la gran mayoría de lenguajes de programación.

Por otro lado, también es importante conocer como un lenguaje puede llegar a alcanzar su objetivo y así poder aportar grandes avances a la industria tecnológica. Algunas características vitales a considerar son:

- Facilidad de uso
- Documentación completa y clara
- Buen rendimiento
- Comunidad activa
- Amplia gama de frameworks y librerías
- Seguridad y robustez
- Actualizaciones regulares
- Innovación y aportes nuevos

Estos son solo algunos aspectos claves a tomar en cuenta, más no son suficientes para asegurar el éxito. Por ello es importante indagar y buscar más información sobre el tema. Para ello es obligatorio conocer que hay detrás de los lenguajes más importantes de la historia. Es por eso que en este escrito vamos exponer información sobre muchos de estos y así ver qué es lo que los han diferenciado de los demás que han quedado en el camino.

Línea del tiempo



Lenguajes actualizados

1. Fortran Antes

FORTTRAN

Data de 1956. Es el primer lenguaje de alto nivel. Su nombre viene de FORmula TRANslation. Fue creado por Backus. Se utiliza principalmente para aplicaciones científicas.

Fortran Después

FORTTRAN

Data de 1956. Es el primer lenguaje de alto nivel. Su nombre viene de FORmula TRANslation. Fue creado por Backus. Se utiliza principalmente para aplicaciones científicas, específicamente en el campo de la Computación de Alto Rendimiento. Algunos ejemplos de uso son: predicción numérica del tiempo y los océanos, fluidodinámica computacional, matemáticas aplicadas, estática y finanzas.

Características principales:

- **Alto rendimiento:** Fortran ha sido diseñado desde cero para realizar cálculos en el campo de la ciencia y la ingeniería computacionalmente exigentes. Los compiladores y bibliotecas disponibles están bastante desarrolladas permitiendo escribir código que se ejecuta muy rápido.
- **Tipado estático y fuerte:** Fortran utiliza un tipado estático y fuerte, lo que permite al compilador detectar inmediatamente numerosos errores de programación. Esto también le permite generar código binario eficiente.
- **Fácil de aprender y usar:** Fortran es un lenguaje relativamente pequeño y sorprendentemente fácil de aprender y usar. La mayoría de expresiones matemáticas y aritméticas aplicadas sobre matrices grandes se expresan tan fácilmente como si se escribieran las ecuaciones en papel.
- **Versátil:** Fortran permite escribir código en el estilo que mejor se adapte a su problema imperativo, procedimental, matricial, orientado a objetos o funcional.
- **Paralelo nativo:** Fortran es un lenguaje de programación nativamente paralelo con una sintaxis intuitiva de tipo matiz para comunicar datos entre procesadores. Puede ejecutar casi el mismo código en un solo procesador, en un sistema multinúcleo con memoria compartida, en un sistema HPC con memoria distribuida o en un sistema de la nube. Los conceptos de Coarrays, equipos, eventos, y subrutinas colectivas le permiten utilizar distintos modelos de programación paralela para elegir el que mejor se adapte a su problema.

2.Pascal Antes

PASCAL

Data de 1969. Su nombre es en honor al conocido matemático Blaise Pascal. Fue desarrollado por Nicklaus Wirth con el propósito de ser utilizado para la enseñanza. Sin embargo su diseño elegante y fiel a la programación estructurada le granjeó un lugar entre los favoritos de los programadores que durante tres décadas le han utilizado como una de las herramientas principales de programación. A finales de los 80's se amplió su espectro al surgir varias mezclas con orientación a objetos en lo que hoy en día llamamos Object Pascal.

Pascal Después

PASCAL

Data de 1969. Su nombre es en honor al conocido matemático Blaise Pascal. Fue desarrollado por Nicklaus Wirth con el propósito de ser utilizado para la enseñanza. Sin embargo su diseño elegante y fiel a la programación estructurada le granjeó un lugar entre los favoritos de los programadores que durante tres décadas le han utilizado como una de las herramientas principales de programación. A finales de los 80's se amplió su espectro al surgir varias mezclas con orientación a objetos en lo que hoy en día llamamos Object Pascal.

Las primeras versiones del compilador de Pascal, entre ellas la más distribuida fue UCSD Pascal, traducían el lenguaje en código para una máquina virtual llamada máquina-P. La gran ventaja de este enfoque es que para tener un compilador de Pascal en una nueva arquitectura de máquina solo hacía falta reimplementar la máquina-P. Como consecuencia de esto, solo una pequeña parte del intérprete tenía que ser reescrita hacia muchas arquitecturas.

Pascal se caracteriza por ser un lenguaje de programación estructurado fuertemente tipificado. Esto implica que:

1. El código está dividido en porciones fácilmente legibles llamadas funciones o procedimientos. De esta forma Pascal facilita la utilización de la programación estructurada en oposición al antiguo estilo de programación monolítica.
2. El tipo de dato de todas las variables debe ser declarado previamente para que su uso quede habilitado.

Otra característica importante a destacar es que en Pascal el tipo de una variable se fija en su definición; la asignación a variables de valores de tipo incompatible no está permitida. Esto previene errores comunes donde las variables son usadas incorrectamente debido a que el tipo no es conocido; y también evita la necesidad de notación húngara, que vienen a ser prefijos que se añaden a los nombres de las variables y que indican su tipo.

3.C++ Antes

C++

Es de finales de los 70's y principios de la década de los 80's. Es el claro sucesor de C. Su principal aporte es agregarle a C una mezcla con Orientación a Objetos. Tuvo un éxito arrollador principalmente debido a dos causas, primero el lenguaje en que se basó ya era un éxito y tenía un muy amplio y fiel grupo de seguidores que mantuvo al trasladarse a C++. Segundo la orientación a objetos agregada fue una muy elegante puesta en práctica de los principios de este paradigma, siendo hoy en día el lenguaje más importante que posee herencia múltiple.

C++ Después

C++

Es de finales de los 70's y principios de la década de los 80's. Es el claro sucesor de C. Su principal aporte es agregarle a C una mezcla con Orientación a Objetos. Tuvo un éxito arrollador principalmente debido a dos causas, primero el lenguaje en que se basó ya era un éxito y tenía un muy amplio y fiel grupo de seguidores que mantuvo al trasladarse a C++. Segundo la orientación a objetos agregada fue una muy elegante puesta en práctica de los principios de este paradigma, siendo hoy en día el lenguaje más importante que posee herencia múltiple.

Se utiliza principalmente para el desarrollo de aplicaciones y softwares más complejos que usualmente requieren de una intervención visual. Algunos ejemplos de uso son: sistemas para el almacenamiento y procesamiento de datos a gran escala, videojuegos, compilación de lenguajes de programación y sistemas operativos como Windows, macOS y Linux.

Principales características:

- Es compatible con bibliotecas que ayudan a escribir código rápidamente.
- Es un lenguaje de programación centrado en objetos y en su manipulación y configuración bajo diversos parámetros.
- La ejecución de los programas desarrollados con C++ suele ser mucho más rápida.
- Al programar con C++ es necesario compilar el código de bajo nivel antes de que sea ejecutado, cabe mencionar que esto no es necesario con otros lenguajes.
- Aprender lenguaje C++ facilita el aprendizaje de otros lenguajes como C#, Java, JavaScript, entre otros.

C++ es un lenguaje que ha tenido un gran éxito con el pasar del tiempo. Ésto gracias a que cuenta con una gran capacidad de rendimiento, poder operar directamente con el sistema operativo, ser compatible con cualquier hardware, ser multiplataforma, y estar en constante actualización. Todo eso y más lo han convertido en un lenguaje robusto y con un buen porcentaje de oportunidades laborales.

4.Oberon Antes

OBERON

Fue creado a mediados de la década de los 80's con el propósito de servir como una plataforma más completa de desarrollo (tratando de seguir un poco el espíritu de C-Unc). Fue creado por Wirth siguiendo la línea Pascal-Modula2 pero no tuvo el mismo éxito que sus antecesores.

Oberon Después

OBERON

Fue creado a mediados de la década de los 80's con el propósito de servir como una plataforma más completa de desarrollo (tratando de seguir un poco el espíritu de C-Unc). Fue creado por Wirth siguiendo la línea Pascal-Modula2 pero no tuvo el mismo éxito que sus antecesores.

Oberon se enfocó bastante en omitir temas complejos, esto con la intención de producir un lenguaje con una mejor curva de aprendizaje, más sencillo de implementar y muy eficiente. Los compiladores de Oberon han sido catalogados por muchos como compactos y rápidos, al tiempo que proporcionan una calidad de código comparable a la de los compiladores comerciales.

Oberon permite la extensión de tipos de registro para la elaboración de abstracciones y estructuras heterogéneas. A diferencia de los dialectos anteriores, Oberon-2 y Active Oberon, la primera versión de Oberon carece de un mecanismo de envío como parte del lenguaje, pero la tiene como técnica de programación o patrón de diseño, permitiéndole obtener una gran flexibilidad en OPP. En el sistema operativo de Oberon, se suelen usar dos técnicas de programación para la llamada de despacho: el conjunto de métodos y el controlador de mensajes.

Algunas de sus principales características son:

- Sintaxis sensible de caso con palabras clave de mayúsculas.
- Tipo-extensión con prueba tipo.
- Módulos y compilación separada
- Operaciones de enlace
- Apoyo a la programación del sistema.

5.Delphi Antes

DELPHI

Data del año 1995. Fue creado originalmente por la empresa Borland y actualmente es comercializado por la empresa estadounidense CodeGear, adquirida en mayo del año 2008 por Embarcadero Technologies. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal.

Es un entorno de desarrollo de software que fue diseñado para la programación de propósito general, dándole énfasis a la programación visual. Es muy usado para el desarrollo de aplicaciones administrativas, debido a su gran flexibilidad y a que incorpora un ambiente de trabajo de interfaz gráfica muy completo.

En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET.

Delphi Después

DELPHI

Data del año 1995. Fue creado originalmente por la empresa Borland y actualmente es comercializado por la empresa estadounidense CodeGear, adquirida en mayo del año 2008 por Embarcadero Technologies. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal.

Es un entorno de desarrollo de software que fue diseñado para la programación de propósito general, dándole énfasis a la programación visual. Es muy usado para el desarrollo de aplicaciones administrativas, debido a su gran flexibilidad y a que incorpora un ambiente de trabajo de interfaz gráfica muy completo.

En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET.

Delphi es un lenguaje muy versátil se utiliza casi para cualquier tipo de proyecto, como por ejemplo servicios del sistema operativo, establecer comunicación entre un servidor web y un programa, aplicaciones de consola, conectividad con bases de datos, para realizar aplicaciones visuales, entre muchas otras.

Genera aplicaciones de código máquina, por lo que la computadora las interpreta inmediatamente y no precisa de un lenguaje que lo haga, como si pasa en otros lenguajes de programación.

Una de las principales ventajas que se puede encontrar en Delphi, es el hecho de que no importa la versión que se tenga, se puede programar DLLs, para poder conectarlos con otros lenguajes de programación. También, los componentes gráficos de Delphi por lo general son más efectivos con respecto a otros lenguajes, haciendo posible que se pueda

programar directamente en dichos componentes visuales e incluso crear nuevos controles que heredan características de los ya existentes.

6.Wyvern Antes

WYVERN

Es un lenguaje de programación para aplicaciones web y móviles creado en el año 2013 desarrollado por Jonathan Aldrich. Su principal objetivo es proveer una forma de usar múltiples lenguajes en el mismo programa de forma segura para la creación de aplicaciones web. Así se puede utilizar el lenguaje que más convenga dependiendo de la función. Se centra mucho en solucionar problemas de seguridad y así proteger las aplicaciones de los ataques de inyección de código. Aún se encuentra en un nivel básico ya que sigue en desarrollo.

Wyvern Después

WYVERN

Es un lenguaje de programación para aplicaciones web y móviles creado en el año 2013 desarrollado por Jonathan Aldrich. Su principal objetivo es proveer una forma de usar múltiples lenguajes en el mismo programa de forma segura para la creación de aplicaciones web. Así se puede utilizar el lenguaje que más convenga dependiendo de la función. Se centra mucho en solucionar problemas de seguridad y así proteger las aplicaciones de los ataques de inyección de código. Aún se encuentra en un nivel básico ya que sigue en desarrollo.

Wyvern determina el sublenguaje que se está usando en el programa mediante los tipos de datos que el programador está manipulando. Los tipos especifican el formato de los datos, ya sean caracteres alfanuméricos o estructurados de datos más complejos, como podría ser una petición de datos a una base de datos de una página Web. Con ello se evitan cosas como tener que pasar cadenas de texto como variables entre lenguajes y así poder aumentar la seguridad.

Este concepto que se usa Wyvern no es sencillo, y desde luego no es algo que los usuarios vayan a poder adaptarse fácilmente, sin embargo, puede llegar a ser un avance muy importante para el desarrollo de lenguajes y prácticas seguras que cierren la puerta a hackers.

Uno de los puntos que juegan en contra, es el hecho de que Wyvern está siendo financiado por NSA, quien ha sido protagonista en los medios debido a muchos escándalos relacionados al espionaje electrónico. Por ello, muchos han salido con especulaciones nada favorables para el lenguaje. También, el hecho de tener una documentación poco amigable, hace que muchos no sean capaces de aportar al proyecto, recordando que este es de código abierto.

Lenguajes nuevos

Microsoft Power Fx

Es un lenguaje de programación funcional, declarativo, de tipo fuerte y de uso general. Fué anunciado en el 2021 y se basa principalmente en el bajo uso de código, ya que se puede trabajar directamente en una barra de fórmulas similar a Excel o la ventana de texto de Visual Studio Code. De esa manera se consigue que este lenguaje pueda ser usado por personas que tengan nula o poca experiencia en el desarrollo de código, así como por personas bastante experimentadas en este aspecto. Está altamente influenciado por lenguajes de programación como Pascal y Miranda.

Carbon

Anunciado por primera vez en la conferencia de Cpp North en Toronto, Canadá, en julio de 2022, por el ingeniero de Google Chandler Carruth. Es un lenguaje experimental, de código abierto, impulsado por Google y diseñado para ser el sucesor del lenguaje de programación C++. Tiene como objetivo principal corregir algunas deficiencias percibidas en C++, como lo son la legibilidad, la interoperabilidad bidireccional y la curva de aprendizaje.

Mojo

Presentado en el 2023. Mojo es un lenguaje de programación que surge principalmente para abordar problemas relacionados a la Inteligencia Artificial. Destaca por tener una velocidad de ejecución alta, gran compatibilidad con Python y ser de tipado fuerte, así como poder gestionar la memoria de forma avanzada. En resumen, se puede ver como una combinación entre la facilidad de uso de Python y la eficiencia de C++.

Documento completo reformado

Introducción

Desde tiempos inmemorables la creación de nuevos lenguajes de programación ha sido algo constante en el mundo de la computación e informática, sin embargo, ¿cuántos de estos lenguajes se han mantenido hasta el día de hoy a flote?

La respuesta es simple, probablemente menos del 1 por ciento de ellos se sigue considerando como un lenguaje de programación vivo. Muchos de ellos alcanzaron la cima y se hundieron con el pasar de los años, otros se quedaron como ideas prometedoras, otros no tuvieron el apoyo económico para alcanzar el éxito, otros se crearon como Hobby sin ambición de crecimiento. Como se puede ver son muchas las posibles razones que pueden explicar el poco éxito que han tenido la gran mayoría de lenguajes de programación.

Por otro lado, también es importante conocer como un lenguaje puede llegar a alcanzar su objetivo y así poder aportar grandes avances a la industria tecnológica. Algunas características vitales a considerar son:

- Facilidad de uso
- Documentación completa y clara
- Buen rendimiento
- Comunidad activa
- Amplia gama de frameworks y librerías
- Seguridad y robustez
- Actualizaciones regulares
- Innovación y aportes nuevos

Estos son solo algunos aspectos claves a tomar en cuenta, más no son suficientes para asegurar el éxito. Por ello es importante indagar y buscar más información sobre el tema. Para dicho fin, es obligatorio conocer que hay detrás de los lenguajes más importantes de la historia. Es por eso que en este escrito vamos exponer información sobre muchos de estos y así ver qué es lo que los han diferenciado de los demás que han quedado en el camino.

HISTORIA DE LOS LENGUAJES Siglo XX Década de los 50's.

SHORT CODE

Es de 1949. Es el lenguaje ensamblador de la UNIVAC. Primer lenguaje ensamblador no binario de la historia.

FORTRAN

Data de 1956. Es el primer lenguaje de alto nivel. Su nombre viene de FORmula TRANslation. Fue creado por Backus. Se utiliza principalmente para aplicaciones científicas, específicamente en el campo de la Computación de Alto Rendimiento. Algunos ejemplos de uso son: predicción numérica del tiempo y los océanos, fluidodinámica computacional, matemáticas aplicadas, estática y finanzas.

Características principales:

- **Alto rendimiento:** Fortran ha sido diseñado desde cero para realizar cálculos en el campo de la ciencia y la ingeniería computacionalmente exigentes. Los compiladores y bibliotecas disponibles están bastante desarrolladas permitiendo escribir código que se ejecuta muy rápido.
- **Tipado estático y fuerte:** Fortran utiliza un tipado estático y fuerte, lo que permite al compilador detectar inmediatamente numerosos errores de programación. Esto también le permite generar código binario eficiente.
- **Fácil de aprender y usar:** Fortran es un lenguaje relativamente pequeño y sorprendentemente fácil de aprender y usar. La mayoría de expresiones matemáticas y aritméticas aplicadas sobre matrices grandes se expresan tan fácilmente como si se escribieran las ecuaciones en papel.
- **Versátil:** Fortran permite escribir código en el estilo que mejor se adapte a su problema imperativo, procedimental, matricial, orientado a objetos o funcional.
- **Paralelo nativo:** Fortran es un lenguaje de programación nativamente paralelo con una sintaxis intuitiva de tipo matiz para comunicar datos entre procesadores. Puede ejecutar casi el mismo código en un solo procesador, en un sistema multinúcleo con memoria compartida, en un sistema HPC con memoria distribuida o en un sistema de la nube. Los conceptos de Coarrays, equipos, eventos, y subrutinas colectivas le permiten utilizar distintos modelos de programación paralela para elegir el que mejor se adapte a su problema.

JOVIAL

Es de 1959. Fue el primer lenguaje cuyo propósito fue el uso general. Es decir no tenía un tipo de aplicaciones objetivo.

LISP

También es de 1959. Su nombre proviene de LISt Processing. Fue creado en el MIT como una herramienta para facilitar el trabajo e investigación en la inteligencia artificial. Perteneció al paradigma funcional y es uno de sus mayores representantes. Se considera como uno de los lenguajes más fáciles de aprender para personas novicias en el arte de la programación. A mediados de la década anterior hubo un fuerte movimiento en varias universidades de nuestro continente por utilizarlo como primer lenguaje de programación.

HISTORIA DE LOS LENGUAJES Siglo XX Década de los 60's.

COBOL

Es de 1960. En ese año el Departamento de Defensa de los EEUU estaba muy preocupado por la aparición de múltiples lenguajes de programación, pues cada dependencia programaba en el lenguaje de su preferencia. Con el fin de mantener un estándar se realizó un concurso de diseño de un lenguaje de programación. El requisito principal: ser un lenguaje orientado a aplicaciones comerciales. El ganador Cobol. Su nombre proviene de Common Business Oriented Language y fue el estándar de la industria durante casi 30 años. A principios de la década pasada se calculaba que en Costa Rica más del 90% del

software empresarial estaba desarrollado en Cobol. En la actualidad el número ha decrecido bastante pero no es raro encontrar aplicaciones desarrolladas en Cobol.

ALGOL

También es de 1960. Fue el primer lenguaje cuya gramática se definió formalmente mediante la herramienta BNF. Esta es un lenguaje para definir gramáticas de lenguajes de programación conocido como Backus-Naur Form en honor a sus dos autores. La forma de programación propuesta por Algol sirvió como base para muchos lenguajes que han sido muy utilizados a través de la historia y que aún hoy son muy apetecidos por los programadores, tales como Pascal, Modula-2, etc.

BASIC

Es de mediados de la década de los 60's. Inicialmente se desarrolló como un pequeño lenguaje con el que fuese fácil introducir a la programación imperativa a los programadores novicios, pero con el paso de los años se tornó en una importante herramienta para el desarrollo de aplicaciones al ser muy apetecido por los programadores, teniendo un lugar de privilegio entre los programadores empíricos. A través de los años ha mantenido su importancia gracias a herramientas como Business Basic en los 80's y Visual Basic en los 90's. No es un lenguaje muy querido en los ambientes académicos principalmente debido a que no cumple algunos principios de programación estructurada y a que las mezclas con orientación a objetos poseen serias deficiencias conceptuales.

PASCAL

Data de 1969. Su nombre es en honor al conocido matemático Blaise Pascal. Fue desarrollado por Nicklaus Wirth con el propósito de ser utilizado para la enseñanza. Sin embargo su diseño elegante y fiel a la programación estructurada le granjeó un lugar entre los favoritos de los programadores que durante tres décadas le han utilizado como una de las herramientas principales de programación. A finales de los 80's se amplió su espectro al surgir varias mezclas con orientación a objetos en lo que hoy en día llamamos Object Pascal.

Las primeras versiones del compilador de Pascal, entre ellas la más distribuida fue UCSD Pascal, traducen el lenguaje en código para una máquina virtual llamada máquina-P. La gran ventaja de este enfoque es que para tener un compilador de Pascal en una nueva arquitectura de máquina solo hacía falta reimplementar la máquina-P. Como consecuencia de esto, solo una pequeña parte del intérprete tenía que ser reescrita hacia muchas arquitecturas.

Pascal se caracteriza por ser un lenguaje de programación estructurado fuertemente tipificado. Esto implica que:

3. El código está dividido en porciones fácilmente legibles llamadas funciones o procedimientos. De esta forma Pascal facilita la utilización de la programación estructurada en oposición al antiguo estilo de programación monolítica.
4. El tipo de dato de todas las variables debe ser declarado previamente para que su uso quede habilitado.

Otra característica importante a destacar es que en Pascal el tipo de una variable se fija en su definición; la asignación a variables de valores de tipo incompatible no está permitida. Esto previene errores comunes donde las variables son usadas incorrectamente debido a que el tipo no es conocido; y también evita la necesidad de notación húngara, que vienen a ser prefijos que se añaden a los nombres de las variables y que indican su tipo.

HISTORIA DE LOS LENGUAJES Siglo XX Década de los 70's.

SMALLTALK

Es un lenguaje de programación que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Metafóricamente, se puede considerar que un Smalltalk es un mundo virtual donde viven objetos que se comunican mediante el envío de mensajes. Un sistema Smalltalk está compuesto por:

- Máquina virtual
- Imagen virtual que contiene todos los objetos del sistema
- Lenguaje de programación (también conocido como Smalltalk)
- Biblioteca de Objetos reusables
- Opcionalmente un entorno de desarrollo que funciona como un sistema en tiempo de ejecución.

Un programa Smalltalk consiste únicamente de objetos, un concepto que se utiliza universalmente dentro de todo sistema Smalltalk. Prácticamente todo, desde un número natural como el 4 hasta un servidor web es considerado un objeto. Los objetos Smalltalk presentan características comunes:

- Tienen una memoria propia.
- Poseen capacidad para comunicarse con otros objetos.
- Poseen la capacidad de heredar características de objetos ancestros.
- Tienen capacidad de procesamiento.

Se busca que todo lo que se pueda realizar con un ordenador se pueda realizar mediante un Sistema Smalltalk, por lo que en algunos casos se puede considerar que incluso sirve como Sistema Operativo. Smalltalk es considerado el primero de los lenguajes orientados a objetos (OOP).

C

Es de 1972. Originalmente fue hecho como parte de un proyecto académico de Dennis Ritchie y debe su nombre a que era el tercer prototipo del lenguaje que estaba desarrollando (por supuesto los primeros dos eran A y B). Posteriormente fue comprado por los laboratorios Bell junto con el sistema operativo que estaba desarrollando Ritchie. Este sistema operativo llegaría a ser Unix. Es un lenguaje muy poderoso que permite al programador el nivel de trabajo con la máquina que provee un ensamblador pero con las facilidades que provee un lenguaje de alto nivel, por esta causa se le suele llamar de forma cariñosa ensamblador de alto nivel. En la actualidad es el lenguaje favorito de los ingenieros en computación, entre otras razones debido a que es un lenguaje que permite la escritura de programas complejos con muy poco código. Por la misma razón se le ha acusado de ser

un lenguaje oscuro pero eficiente. Sin embargo debe recordarse que la programación oscura depende del uso que el programador le da al lenguaje.

PROLOG

Data de 1973. Surgió como la forma más natural de representar los conceptos de lógica en un lenguaje de programación. Es por esta razón que los lenguajes del paradigma lógico siempre tienen cierta semejanza con Prolog. Existen varios dialectos entre los que se pueden resaltar el de Turbo Prolog de Borland, el prolog de Edimburgo y Prolog III. Este último aporta el trabajo con restricciones a la programación lógica. Para poder comprender Prolog es necesario comprender muy claramente los principios del paradigma de programación lógica y para comprender estos es necesario tener claridad en el trabajo con la lógica y semántica formal.

ML

Tuvo su primera aparición en el año 1973. Es uno de los lenguajes favoritos en Europa donde en la mayoría de universidades lo utilizan como primer lenguaje de programación. Perteneció primordialmente al paradigma funcional pero mantiene algo de sabor del paradigma imperativo. Fue desarrollado por Robin Milner y su objetivo original era servir como una herramienta que facilitara la creación de otros lenguajes de programación, de hecho su nombre es una abreviatura de MetaLanguage (Es decir un lenguaje para definir lenguajes). Existen varios dialectos entre los que podemos citar el ML puro propiamente y el bien conocido Caml Light.

X10

El lenguaje de programación X10 está siendo desarrollado por IBM en el Centro de Investigación Thomas J. Watson, para apoyar la programación en paralelo. Se asemeja mucho al lenguaje de programación Java, pero ofrece soporte adicional para arreglos y concurrencia.

X10 usa el modelo de programación paralela Partitioned Global Address Space (PGAS). Es básicamente orientado a objetos, pero posee elementos de otros paradigmas. Utiliza clases, interfaces y estructuras, aunque también agrega nuevos elementos, como places (multiprocesador virtual de memoria compartida), actividades (hilos que se ejecutan para actuar sobre los objetos de datos), y relojes (clocks).

ADA

Fue diseñado en 1975 por el equipo japonés de JMishra a petición del Departamento de Defensa de los EEUU, quien sacó a concurso en ese año el diseño de un lenguaje de programación para todas sus aplicaciones científicas. Es un lenguaje bastante árido que sin embargo es una buena opción para el manejo de concurrencia (requisito importante para los intereses del Departamento de Defensa). En la actualidad la gran mayoría de aplicaciones militares están programadas en Ada. Se bautizó en honor a Ada Lovelace quien fue la primera programadora de la historia.

LOGO

Es de 1976. Basó sus ideas en Lisp agregándole como distintivo el manejo de gráficos de tortuga. Fue desarrollado por el MIT para servir el doble propósito de la enseñanza y de facilitar proyectos en el área de inteligencia artificial. Durante varios años en nuestro país se le utilizó para introducir a los niños en el mundo de la computación por lo que tiene cierta fama infundada e injusta de ser un juguete cuando en realidad es una herramienta poderosa de programación que puede respaldar proyectos de alta complejidad.

SMALLTALK V

Esta versión del lenguaje SmallTalk es de 1979, sin embargo se basa en versiones anteriores de los años 1970 y 1976. Es un lenguaje muy importante por dos causas principales. Primero es uno de los pioneros de la Orientación a Objetos siendo el representante del paradigma puro más fiel que existe. Segundo fue el primer lenguaje de programación en ser un lenguaje cien por ciento gráfico, incluyendo un ambiente de desarrollo como parte de la definición del lenguaje. Fue desarrollado inicialmente por la Xerox (cuando no eran The Document Company :) quienes introdujeron en la computación conceptos y herramientas que hoy en día son en extremo comunes (como mouse, ventanas, íconos, etc.).

MODULA-2

Data de finales de la década de los 70's. Fue creado por Nicklaus Wirth como el sucesor de Pascal. Su principal aporte fue el concepto de modularización (ya que la definición original de Pascal no tiene el concepto de módulos que algunas implementaciones le suelen agregar por su cuenta). Algunas implementaciones de Modula-2 agregan a la vez un manejo interesante de la concurrencia. Durante algún tiempo substituyó a Pascal en el puesto de primer lenguaje de programación en algunas Universidades por lo que hay varias generaciones de amantes y defensores de Modula-2. Uno de sus principales problemas fue la falta de una implementación que contara con el respaldo de una casa grande de software.

C++

Es de finales de los 70's y principios de la década de los 80's. Es el claro sucesor de C. Su principal aporte es agregarle a C una mezcla con Orientación a Objetos. Tuvo un éxito arrollador principalmente debido a dos causas, primero el lenguaje en que se basó ya era un éxito y tenía un muy amplio y fiel grupo de seguidores que mantuvo al trasladarse a C++. Segundo la orientación a objetos agregada fue una muy elegante puesta en práctica de los principios de este paradigma, siendo hoy en día el lenguaje más importante que posee herencia múltiple.

Se utiliza principalmente para el desarrollo de aplicaciones y softwares más complejos que usualmente requieren de una intervención visual. Algunos ejemplos de uso son: sistemas para el almacenamiento y procesamiento de datos a gran escala, videojuegos, compilación de lenguajes de programación y sistemas operativos como Windows, macOS y Linux.

Principales características:

- Es compatible con bibliotecas que ayudan a escribir código rápidamente.

- Es un lenguaje de programación centrado en objetos y en su manipulación y configuración bajo diversos parámetros.
- La ejecución de los programas desarrollados con C++ suele ser mucho más rápida.
- Al programar con C++ es necesario compilar el código de bajo nivel antes de que sea ejecutado, cabe mencionar que esto no es necesario con otros lenguajes.
- Aprender lenguaje C++ facilita el aprendizaje de otros lenguajes como C#, Java, JavaScript, entre otros.

C++ es un lenguaje que ha tenido un gran éxito con el pasar del tiempo. Ésto gracias a que cuenta con una gran capacidad de rendimiento, poder operar directamente con el sistema operativo, ser compatible con cualquier hardware, ser multiplataforma, y estar en constante actualización. Todo eso y más lo han convertido en un lenguaje robusto y con un buen porcentaje de oportunidades laborales.

HISTORIA DE LOS LENGUAJES Siglo XX Década de los 80's.

OBERON

Fue creado a mediados de la década de los 80's con el propósito de servir como una plataforma más completa de desarrollo (tratando de seguir un poco el espíritu de C-Unix). Fue creado por Wirth siguiendo la línea Pascal-Modula2 pero no tuvo el mismo éxito que sus antecesores.

Oberon se enfocó bastante en omitir temas complejos, esto con la intención de producir un lenguaje con una mejor curva de aprendizaje, más sencillo de implementar y muy eficiente. Los compiladores de Oberon han sido catalogados por muchos como compactos y rápidos, al tiempo que proporcionan una calidad de código comparable a la de los compiladores comerciales.

Oberon permite la extensión de tipos de registro para la elaboración de abstracciones y estructuras heterogéneas. A diferencia de los dialectos anteriores, Oberon-2 y Active Oberon, la primera versión de Oberon carece de un mecanismo de envío como parte del lenguaje, pero la tiene como técnica de programación o patrón de diseño, permitiéndole obtener una gran flexibilidad en OPP. En el sistema operativo de Oberon, se suelen usar dos técnicas de programación para la llamada de despacho: el conjunto de métodos y el controlador de mensajes.

Algunas de sus principales características son:

- Sintaxis sensible de caso con palabras clave de mayúsculas.
- Tipo-extensión con prueba tipo.
- Módulos y compilación separada
- Operaciones de enlace
- Apoyo a la programación del sistema.

MIRANDA

Creado en 1985 por David Turner, su objetivo era crear un lenguaje puramente funcional comercial por esta razón la palabra Miranda es una marca registrada por la compañía Research Software de Inglaterra.

Fue el primer lenguaje funcional comercial y no con fines educativos.

Se caracteriza porque los argumentos se pasan a las funciones sin ser evaluados, estos se evalúan en la función cuando esta los va a utilizar, a esto se le conoce como lazy evaluation.

Los tipos básicos son char, num, bool y "nada"; donde "nada" se asigna por default a los parámetros que en los que el tipo no afecta su interpretación.

Las listas en Miranda permiten una cantidad de elementos infinitos.

Aparte tiene un mecanismo para crear y manejar módulos de programa, además de un mecanismo para que las funciones internas de estos sean transparentes para otros programas que los utilicen.

Miranda es un lenguaje fuertemente tipado, sin embargo la declaración de tipos es opcional, esto quiere decir que si no se coloca de manera explícita el tipo, el intérprete lo deduce directamente de los parámetros y de su uso.

EIFFEL

Historia: El lenguaje de programación Eiffel fue creado por Bertrand Meyer y desarrollado por su empresa, Interactive Software Engineering (ISE) de Goleta, California, en 1985.

Características:

- Portable: funciona en plataformas importantes de la industria(Windows, OS / 2, etc Linux) Open System: está basado en C,C++.
- "Melting Ice Technology" : combina la compilación, la interpretación de bytecode.
- "Design by Contract" : aplicados a través de afirmaciones como invariantes de clase, condiciones previas y postcondiciones.
- Automatic Documentation ("Short Form"): posee un generador de documentación.
- Multiple Inheritance : la herencia
- Repeated Inheritance: una clase hereda de otra a través de dos o más padres.
- Statically Typed : errores son capturados en tiempo de compilación.
- Dynamically Bound: garantiza que la versión correcta de una operación se aplicarán en función del objeto de destino.

FOXPRO

(FoxBASE Professional) Lenguaje de Orientación a Objetos, a la vez era un sistema de gestión de base de datos. Tenía un motor propio de base de datos, muy popular en el desarrollo de software comercial. Originalmente publicado por Fox Software, para luego ser publicado por Microsoft.

ERLANG

Erlang es un lenguaje de programación concurrente y un sistema de ejecución que incluye una máquina virtual y bibliotecas.

El subconjunto de programación secuencial de Erlang es un lenguaje funcional, con evaluación estricta, asignación única, y tipado dinámico. Fue diseñado en la compañía Ericsson para realizar aplicaciones distribuidas, tolerantes a fallos, soft-real-time y de funcionamiento ininterrumpido. Proporciona el cambio en caliente de código de forma que éste se puede cambiar sin parar el sistema. Originalmente, Erlang era un lenguaje propietario de Ericsson, pero fue cedido como software de código abierto en 1998. La implementación de Ericsson es, principalmente interpretada pero también incluye un compilador HiPE.

La creación y gestión de procesos es trivial en Erlang, mientras que, en muchos lenguajes, los hilos se consideran un apartado complicado y propenso a errores. En Erlang toda concurrencia es explícita.

PERL

Larry Wall comenzó a trabajar en Perl en 1987. Perl es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de GUI y más.

Se previó que fuera práctico (facilidad de uso, eficiente, completo) en lugar de hermoso (pequeño, elegante, mínimo). Sus principales características son que es fácil de usar, soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles.

Larry Wall comenzó a trabajar en Perl en 1987. Perl es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de GUI y más.

Se previó que fuera práctico (facilidad de uso, eficiente, completo) en lugar de hermoso (pequeño, elegante, mínimo). Sus principales características son que es fácil de usar, soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles.

HISTORIA DE LOS LENGUAJES Siglo XX Década de los 90's.

HASKELL

Su nombre proviene del lógico Haskell Curry. Apareció en 1990, y fue diseñado por: Universidad de Yale y la Universidad de Glasgow. Es un lenguaje de programación

puramente funcional de propósito general y fuertemente tipado, que incluye muchas de las últimas innovaciones en el desarrollo de los lenguajes de programación funcional, como son las funciones de orden superior, tipos polimórficos estáticos, tipos definidos por el usuario, encaje por patrones, y definiciones de listas por comprensión. Incorpora, además, otras características interesantes como el tratamiento sistemático de la sobrecarga, la facilidad en la definición de tipos abstractos de datos, el sistema de entrada/salida puramente funcional y la posibilidad de utilización de módulos.

JAVA

Es el último de los lenguajes que ya se considera como parte importante en la historia. Data de principios de la década de los 90's. Inicialmente fue desarrollado como un estandarte para tratar de mantener un estándar en ese mar infinito que es Internet. Aunque no ha llegado a cumplir su objetivo inicial del todo poco a poco ha ido ganando su nicho en el mundo de los lenguajes de programación. Está basado como lenguaje en C++, pero tratando de mantener la simplicidad, por lo que tiene varios cambios importantes. Se le redujo el acceso al nivel más bajo de programación con el fin de mantener los programas alejados de plataformas específicas y se le trató de sacar del lado de programación oscura tomando prestados algunos conceptos de Pascal para darle al programador una herramienta mucho más sencilla de usar. Con ese mismo fin se trató de aislar al programador de problemas de implementación con la memoria dinámica por lo que esos procesos están automatizados. Además se eliminó el concepto de herencia múltiple de su orientación a objetos. Para poder servir de estándar en Internet se diseñó como un compilador-interpretativo que aislara los programas escritos en Java de la plataforma final donde correrían, mediante el uso de una máquina virtual intermedia, cuyo lenguaje de programación (conocido como Bytecode) fuese sumamente sencillo para que en cualquier plataforma sin mayor exigencia de recursos se pudiese crear dicha máquina virtual. Sin embargo muchas de las implementaciones actuales de Java han cambiado esa austeridad por el poder brindarle al programador un conjunto muy amplio de herramientas con las cuales se pueda desenvolver y competir de esta forma con otros productos actuales del mercado. Esto último ha hecho que se gane la fama de ser un lenguaje muy acaparador de recursos.

PYTHON

Es un lenguaje de propósito general creado en 1991. Diseñado por Guido van Rossum, su filosofía hace énfasis en la facilidad de lectura y escritura de código. Es un lenguaje multiparadigma, más que todo funcional, imperativo y orientado a objetos. Posee dialectos como Stackless Python y Rpython.

Tiene manejo de memoria automático y debido a su condición de lenguaje dinámico también es usado como lenguaje de scripts. Posee un recolector de Basura y tiene soporte para Unicode. Permite a los usuarios la escogencia entre paradigmas a la hora de programar. Tiene alto soporte en programación orientada a objetos y en programación estructurada. Muchos otros paradigmas son soportados mediante extensiones. La programación funcional es soportada de manera limitada, pero tiene bastantes similitudes con lenguajes basados en LISP tales como Scheme.

Python es un lenguaje de programación interpretado creado por Guido van Rossum en el año 1991. Se compara habitualmente con Tcl, Perl, Scheme, Java y Ruby. La última versión estable del lenguaje es la 3.1.1. Los usuarios de Python consideran a éste mucho más limpio y elegante para programar.

Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas. También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, sockets y hasta interfaces a GUI (interfaz gráfica con el usuario) como Tk, GTK, Qt entre otros.

Python se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El nombre del lenguaje proviene de la afición de su creador original, Guido van Rossum, por los humoristas británicos Monty Python. El principal objetivo que persigue este lenguaje es la facilidad, tanto de lectura, como de diseño.

Fue creado en 1991 por Guido van Rossum. Su nombre se debe a la afición de su creador por el grupo humorístico británico Monty Python. Se creó con la idea de que fuera el sucesor del lenguaje de programación ABC. Tiene características de lenguajes funcionales tales como lambda, map, filter y reduce. También tomó características de lenguajes funcionales, tal como Haskell, del cual tomó los list comprehensions. También tiene varias influencias de lenguajes orientados a objetos como Java, del cual toma al uso de SAX, algunos paquetes, y hasta cierto punto su sintaxis. De hecho se dice que Python todo es un objeto, también soporta herencia múltiple y polimorfismo. Además incorpora un recolector de basura, capaz de tomar referencias cíclicas. Python es un lenguaje interpretado, multiplataforma y multiparadigma. Entre otras propiedades está que consta de la propiedad de ligadura dinámica de métodos. Actualmente Python se desarrolla como un proyecto de código abierto, bajo la tutela de Python Software Foundation.

LUA

Data de 1993, creado por Roberto Ierusalimsky, Luis Henrique de Figueiredo y Waldemar Celes, como un lenguaje orientado a objetos, imperativo, y con una semántica entendible que se basa en scripts.

Está inspirado en Scheme, SNOBOL, Modula, CLU y C++.

En las versiones inferiores a la 5.0 fue distribuido bajo una licencia similar a la BSD, posterior a esta versión se comenzó a distribuir bajo la licencia del MIT compatible con GPL.

Esto implica que LUA es un lenguaje que puede ser utilizado para crear aplicación con distintos propósitos de manera gratuita, sin embargo LUA no es open-source.

En LUA las variables no tienen tipo además las estructuras de datos como listas, tablas hash, matrices y conjuntos se pueden implementar por medio de la única estructura de datos: la tabla.

El código creado en LUA no es ejecutado directamente, primero es compilado a ByteCode y luego se ejecuta en una maquina virtual.

El lenguaje LUA es pequeño pero poderoso, en su versión 5.1.4 el código fuente, documentación y ejemplos solo ocupan 860kb descomprimido.

Es ampliamente utilizado en distintas aplicaciones comerciales como juegos y programas de diseño, algunos ejemplos de aplicaciones que utilizan LUA son Adobe Photoshop lighthroom, World of Warcraft , S.T.A.L.K.E.R, Half-Life 2 en el Garry's Mod.

RUBY

El lenguaje fue creado por Yukihiro "Matz" Matsumoto, quien empezó a trabajar en Ruby el 24 de febrero de 1993, y lo presentó al público en el año 1995. Ha dicho que Ruby está diseñado para la productividad y la diversión del desarrollador, siguiendo los principios de una buena interfaz de usuario. Sostiene que el diseño de sistemas necesita enfatizar las necesidades humanas más que las de la máquina.

Ruby es orientado a objetos: todos los tipos de datos son un objeto, incluidas las clases y tipos que otros lenguajes definen como primitivas, (como enteros, booleanos, y "nil"). Toda función es un método. Las variables siempre son referencias a objetos, no los objetos mismos. Ruby soporta herencia con enlace dinámico, mixins y métodos singleton (pertenecientes y definidos por una sola instancia más que definidos por la clase). A pesar de que Ruby no soporta herencia múltiple, las clases pueden importar módulos como mixins. La sintaxis procedural está soportada, pero todos los métodos definidos fuera del ámbito de un objeto son realmente métodos de la clase Object. Como esta clase es padre de todas las demás, los cambios son visibles para todas las clases y objetos.

Ruby ha sido descrito como un lenguaje de programación multiparadigma: permite programación procedural (definiendo funciones y variables fuera de las clases haciéndolas parte del objeto raíz Object), con orientación a objetos, (todo es un objeto) o funcionalmente (tiene funciones anónimas, clausuras o closures, y continuations; todas las sentencias tiene valores, y las funciones devuelven la última evaluación). Soporta introspección, reflexión y metaprogramación, además de soporte para hilos de ejecución gestionados por el intérprete. Ruby tiene tipificado dinámico, y soporta polimorfismo de tipos (permite tratar a subclases utilizando el interfaz de la clase padre). Ruby no requiere de polimorfismo de funciones al no ser fuertemente tipado.

Ruby es un lenguaje de programación interpretado, reflexivo y orientado a objetos, creado por el programador japonés Yukihiro "Matz" Matsumoto. Este empezó a trabajar en Ruby en 1993, y lo presentó públicamente en 1995. Su sintaxis está inspirada en Python, Perl con características de programación orientada a objetos similares a Smalltalk. También presenta características de programación funcional y es un lenguaje interpretado.

El lenguaje se comporta de tal manera que minimiza la confusión de los usuarios experimentados. Matz ha dicho que su principal objetivo era hacer un lenguaje que le divertiera él mismo, minimizando el trabajo de programación y la posible confusión. En este lenguaje todos los tipos de datos son objetos, inclusive las clases que otros lenguajes

definen como primitivas y todas las funciones son métodos. Ruby es considerado como un lenguaje multiparadigma. Además, cabe destacar, que se distribuye bajo una licencia de software libre.

Este lenguaje salió a inicios de los 90, salió en 1993 y fue presentado públicamente en 1995. Este es un lenguaje interpretado, reflexivo y orientado a objetos. Fue creado por el programador japonés Yukihiro Mat. La sintaxis es una mezcla entre python y perl con características de programación similares a smalltalk. Además comparte funcionalidades con otros lenguajes de programación como lo son Lisp, Lua, Dylan y CLU.

Ruby es un lenguaje interpretado de una sola pasada y es implementado bajo una licencia de software libre.

Existe una versión de Ruby llamada JRuby que fue el intento de llevar Ruby a la plataforma Java y la versión Rubines, un intérprete modelado basado en las máquinas virtuales de Smalltalk.

El diseñador de Ruby menciona que el propósito de Ruby es el énfasis en productividad y la diversión de desarrollar, siguiendo los propósitos de una buena interfaz de usuario.

Si hablamos un poco de la semántica, podemos decir que todos los tipos de datos son un objeto, incluyendo las clases y tipos que otros lenguajes definen como primitivas. Define operaciones como métodos. Las variables siempre son referencias a objetos, no los objetos. Si hablamos de la herencia tenemos que soportar herencia con enlace dinámico, minix y métodos Singleton; y no soporta la herencia múltiple, pero a pesar de ello las clases pueden importar módulos como minix.

PHP

Hypertext Preprocessor, es un lenguaje de programación interpretado de alto nivel embebido en páginas HTML y ejecutado en un servidor. El PHP inició como una modificación a Perl escrita por Rasmus Lerdorf a finales de 1994, desarrollado por la política de código abierto. Características: soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML. Su sintaxis es muy similar a la del ASP, pues el código PHP va incrustado dentro del código HTML. PHP es un potente lenguaje y el intérprete, tanto incluido en el servidor Web como módulo o ejecutado como un binario CGI, puede acceder a ficheros, ejecutar comandos y abrir comunicaciones de red en el servidor. Todas estas características hacen que lo que se ejecute en el servidor Web sea seguro por defecto. El modo de operación del PHP es el siguiente: El Navegador realiza una petición al servidor (se escribe la URL), después el servidor ejecuta el código PHP solicitado y retorna el código HTML generado al Navegador, por último el Navegador muestra la respuesta del servidor. En caso de estar sobre un servidor Linux y Unix, es más rápido que ASP, dado que se ejecuta en un único espacio de memoria.

Fue creado originalmente por Rasmus Lerdorf en 1994, es un lenguaje de programación interpretado de código abierto, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta.

Creado originalmente por el programador danés-canadiense Rasmus Lerdorf en 1994. La implementación principal de PHP es producida ahora por The PHP Group. Su nombre viene de PHP Hypertext Pre-processor. Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas, puede ser incrustado dentro de código HTML. Además, permite crear aplicaciones con una interfaz gráfica para el usuario. La versión PHP CLI (Command Line Interface) permite ser usado desde la línea de comandos. Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite. Tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos como UNIX (Linux o Mac OS X) y Windows. Puede interactuar con los servidores de web más populares. Usa una mezcla entre interpretación y compilación para intentar ofrecer a los programadores rendimiento y flexibilidad. Permite aplicar técnicas de programación orientada a objetos y el uso de una biblioteca nativa de funciones sumamente amplia e incluida. Tiene una manera diferente a la convencional de compilación de lenguajes como C++ donde el código es compilado a código ejecutable, pues PHP compila para el código una serie de instrucciones (llamadas opcodes) siempre que estas son accedidas. Estas instrucciones son entonces ejecutadas una por una hasta que el script termina. Tiene gran parecido con lenguajes muy comunes como C y Perl, por lo que la mayoría de programadores pueden crear aplicaciones e involucrarse sin tener que aprender todo un nuevo grupo de funciones. Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.

JAVASCRIPT

Fue creado en el año 1995 pero no es sino hasta el año 2000 cuando éste se hace compatible con otros navegadores aparte de Netscape y consolida su popularidad como código integrado en los navegadores web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. El bautizo del nombre JavaScript coincidió aproximadamente con el momento en que Netscape (empresa que desarrolló JavaScript) agregó soporte para la tecnología Java en su navegador web Netscape Navigator en la versión 2.0B3 en diciembre de 1995. La denominación produjo confusión, dando la impresión de que el lenguaje es una prolongación de Java pero hay que denotar que Java no tiene nada que ver con JavaScript (aparte del nombre y su parecida sintaxis derivada de C); Java tiene tipos estáticos, JavaScript los tiene dinámicos; Java es corrido desde un bytecode compilado, JavaScript es corrido desde el código fuente; y las diferencias siguen.

El uso primario del JavaScript es escribir funciones que estén incorporadas en las páginas HTML e interactúen con el DOM (Document Object Model).

La Historia de JavaScript se inicia en el año de 1995 cuando Netscape introduce la versión 2.0 de Navigator e incluye JavaScript bajo el nombre de Mocha, cuando aparece esta versión de Navigator se le llamaba LiveScript. Lo que quería Netscape es que JavaScript fuera un lenguaje de guiones, fácil de usar y que cualquier persona pudiera utilizarlo. Después de 2 años JavaScript se convirtió en una de las herramientas más utilizadas por los desarrolladores web, incluso se utiliza más que el propio Java y ActiveX. JavaScript es un lenguaje de scripting basado en objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript es un dialecto de ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan.

Nace en 1995 bajo el nombre de Mocha, luego pasa a llamarse LiveScript y finalmente es bautizado como JavaScript. Desarrollado inicialmente por Brendan Eiche en la empresa Netscape que luego firmo una alianza junto Sun Microsystems para su finalización. Su objetivo principal era ser un lenguaje fácil de usar por cualquier persona especial para desarrollar aplicaciones Web. Viene incorporada junto al navegador y no necesita un kit de desarrollo. Es un lenguaje de Scripting por excelencia y el más popular entre los programadores Web, con casi un 90% de las aplicaciones desarrolladas.

D

La primera versión apareció en el año 1995, aunque una versión estable fue lanzada en enero de 2007 y otra versión 2,0 en junio del mismo año.

Nace como un rediseño de C++ donde toma algunas características de este y desecha otras, por lo que no son compatibles entre sí, además de que han agregado varios conceptos de otros lenguajes como Java, C# y Eiffel.

Tiene soporte de lenguaje ensamblador, con lo que se pueden realizar tareas de muy bajo nivel. Se puede anotar que posee Orientación a objetos basada en herencia simple. Posee un recolector de basura, pero también se puede acceder a la memoria directamente.

Historia: D es un lenguaje de programación de uso general desarrollado por Walter Bright cuya primera versión apareció en 1995. Se origina como un rediseño de C++, con un enfoque más pragmático, pero no es un lenguaje puramente derivado del anterior. D ha mantenido algunas características de C++ y también está influido por otros conceptos de otros lenguajes como Java, C# y Eiffel.

Características: D retiene la habilidad de C++ de hacer código de bajo nivel, permitiendo incluir código en ensamblador. La herencia múltiple de C++ es reemplazada por herencia simple e interfaces y 'mixins'. La sintaxis de declaración y expresiones se parecen a las de C++.

Manejo de memoria: La memoria es usualmente manejada por un recolector de basura, pero objetos específicos pueden ser finalizados inmediatamente cuando salen del ámbito. Las implementaciones actuales de D compilan el código directamente a código máquina para lograr una ejecución eficiente.

- DMD: El compilador de Digital Mars, el compilador oficial de D. El código es abierto.
- GDC: Un front end para el back end GCC, construido usando las fuentes del compilador DMD.
- LDC: Un compilador basado en DMD y usa LLVM como back end.

DELPHI

Data del año 1995. Fue creado originalmente por la empresa Borland y actualmente es comercializado por la empresa estadounidense CodeGear, adquirida en mayo del año 2008 por Embarcadero Technologies. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal.

Es un entorno de desarrollo de software que fue diseñado para la programación de propósito general, dándole énfasis a la programación visual. Es muy usado para el desarrollo de aplicaciones administrativas, debido a su gran flexibilidad y a que incorpora un ambiente de trabajo de interfaz gráfica muy completo.

En sus diferentes variantes, permite producir archivos ejecutables para Windows, Linux y la plataforma .NET.

Delphi es un lenguaje muy versátil se utiliza casi para cualquier tipo de proyecto, como por ejemplo servicios del sistema operativo, establecer comunicación entre un servidor web y un programa, aplicaciones de consola, conectividad con bases de datos, para realizar aplicaciones visuales, entre muchas otras.

Genera aplicaciones de código máquina, por lo que la computadora las interpreta inmediatamente y no precisa de un lenguaje que lo haga, como si pasa en otros lenguajes de programación.

Una de las principales ventajas que se puede encontrar en Delphi, es el hecho de que no importa la versión que se tenga, se puede programar DLLs, para poder conectarlos con otros lenguajes de programación. También, los componentes gráficos de Delphi por lo general son más efectivos con respecto a otros lenguajes, haciendo posible que se pueda programar directamente en dichos componentes visuales e incluso crear nuevos controles que heredan características de los ya existentes.

HISTORIA DE LOS LENGUAJES Primera Década Siglo XXI.

LENGUAJES EN PLATAFORMAS .NET

Es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. La plataforma .NET de Microsoft es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma. Esta solución es el producto principal en la oferta de Microsoft, y pretende ser utilizada por la mayoría de las aplicaciones creadas para la plataforma Windows. La norma (incluido en ECMA-335, ISO/IEC 23271) que define el conjunto de funciones que debe implementar la biblioteca de clases base (BCL por sus siglas en inglés, tal vez el más importante de los componentes de la plataforma), define un conjunto funcional mínimo que debe implementarse para que el marco de trabajo sea soportado por un sistema operativo. Aunque Microsoft implementó esta norma para su sistema operativo Windows, la publicación de la norma abre la posibilidad de que sea implementada para cualquier otro sistema operativo existente o futuro, permitiendo que las aplicaciones corran sobre la plataforma independientemente del sistema operativo para el cual haya sido implementada. El Proyecto Mono emprendido por Ximian pretende realizar la implementación de la norma para varios sistemas operativos adicionales bajo el marco del código abierto.

Esta plataforma es de las más usadas en los últimos años por la mayoría de empresas, debido a su facilidad para funcionar en redes tanto locales como en Internet. Además presenta un soporte para bases de datos muy sencillo.

La plataforma .NET debe su éxito en su mayoría gracias a su IDE llamado Microsoft Visual Studio, que otorga a los programadores un ambiente agradable, aunque un poco pesado en lo que a recursos se refiere, para programar en los diferentes lenguajes que facilita dicha plataforma. Los lenguajes más usados de la plataforma .NET son Visual Basic y Visual C#.

PROCESSING

Lenguaje de programación que nació con este nuevo milenio. Basado en la licencia GNU GPL, Processing integra un lenguaje de programación y entorno de desarrollo que se caracteriza por su fácil utilización y aprendizaje. Fue creado para usarse en el ámbito de la enseñanza escolar, para que personas sin conocimiento de desarrollo de aplicaciones puedan aprender los fundamentos de la programación basándose en un contexto visual.

Creado bajo el modelo Java, Processing hereda todas las funcionalidades de éste, pudiendo convertirse entonces en una herramienta importante cuando se trata de encarar proyectos complejos. El punto fuerte de este lenguaje es que nos permite desarrollar proyectos multimedia e interactivos, que abarquen tecnología 2D y 3D.

A su vez, en el ámbito 3D, Processing posee una integración total para la aceleración basada en OpenGL, y también puede ejecutarse sin problema bajo las plataformas

Windows, Linux y OS-X. Gracias a todas estas prestaciones, más su gran comunidad en línea que se ocupa de actualizar constantemente las mejoras desarrolladas para este lenguaje, Processing ha ganado mucha popularidad. A tal punto que ha conseguido obtener un lugar dentro de los torneos de verano organizados por Google para chicos y adolescentes (Google Summer Code y Computer Science Summer Institute). Fue iniciado por Ben Fry y Casey Reas a partir de reflexiones en el Aesthetics and Computation Group del MIT Media Lab dirigido por John Maeda. Se distribuye bajo la licencia GNU GPL.

C#

Lenguaje del paradigma orientado a objetos desarrollado por Microsoft, es una evolución de los lenguajes de programación C y C++, fue diseñado para construir aplicaciones que se ejecutan en .NET Framework. Este ofrece una interfaz común para trabajar de manera cómoda y visual con cualquiera de los lenguajes de la plataforma .NET. Su compilador produce archivos ejecutables (.exe), librerías dinámicas (.dll) o módulos de código (.netmodule). Algunas características y ventajas de c# son: La colección de basura automáticamente reclama la memoria ocupada por objetos no usados; el manejo de excepción proporciona un acercamiento estructurado y extensible a la detección de errores y la recuperación; y el diseño seguro de tipo de lenguaje hace imposible de leer de variables inicializadas, no importa el orden en que haya sido definidas las clases o funciones, todos los valores son inicializados antes de ser usados, no se pueden utilizar valores no booleanos, eliminación del uso de los punteros, no existen las dependencias circulares, entre otras. Existen otros compiladores además de Microsoft tales como MONO, #develop, delphi, dotGNU y otros.

C # o C# (pronunciado si sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA e ISO. Nace en el año 2001 y según las fuentes es un lenguaje multiplataforma, gracias a la implementación Mono que permite programar en dicho lenguaje en sistemas operativos Linux y Mac. (Wikipedia, Wikipedia La Enciclopedia Libre) Fue diseñado por Anders Hejlsberg. La creación del nombre del lenguaje, C#, proviene de dibujar dos signos positivos encima de los dos signos positivos de "C++", queriendo dar una imagen de salto evolutivo del mismo modo que ocurrió con el paso de C a C++. C# es un híbrido de C++ y Java que fusiona, principalmente, la capacidad de combinar operadores propia del primero (no incorpora la herencia múltiple) con la plena orientación a objetos del segundo. La orientación a objetos es tal que el propio programa está encapsulado en una clase. Actualmente C# se encuentra entre los 10 lenguajes más utilizados. (Saavedra).

Este lenguaje hizo su aparición en diciembre del 2001. Es orientado a objetos e imperativo, fue desarrollado y estandarizado por Microsoft como parte de su plataforma .NET. Su sintaxis se deriva de C/C++ y utiliza el modelo de objetos de .NET, aunque incluye mejoras derivadas de otros lenguajes (entre ellos Delphi). C# es un lenguaje de programación independiente diseñado para generar programas sobre la plataforma .NET. Hejlsberg lideró el proyecto de desarrollo de C#.

SCALA

Scala salió a la luz en el año 2003. Fue diseñado por Martin Odersky. Es un lenguaje de programación multi-paradigma. Del paradigma orientado a objetos se reconoce el hecho de que todo valor es un objeto, se utilizan clases (incluyendo las del tipo genérico) para definir tipos y el comportamiento de los objetos, la abstracción se realiza extendiendo clases y se utiliza un mecanismo de composición basado en mixins para reemplazar la herencia múltiple.

En cuanto al paradigma funcional, toda función es un valor, y es posible crear funciones anónimas, de orden superior, y anidadas; además soporta currying. Scala utiliza la tipificación estática para asegurar la coherencia y seguridad de las abstracciones. Los programas en Scala corren en la Máquina Virtual de Java, ya que los compiladores de estos lenguajes producen byte code prácticamente idéntico. Esta alta compatibilidad permite a Scala acceder a código y bibliotecas Java. Más allá de esto, Scala es capaz de interactuar no sólo con Java, sino con la plataforma .NET. Por último, el lenguaje Scala posee mecanismos que facilitan el extender el lenguaje a través de bibliotecas.

Fue creado en 2001. Es un lenguaje de programación de propósito general que incluye características de lenguajes orientados a objetos y funcionales; reduce drásticamente el tamaño de un programa comparado con su contraparte en Java. Los programas desarrollados en este lenguaje corren sobre la máquina virtual de Java y son compatibles en cuanto a plataforma, además se pueden hacer llamadas a una sección de código escrito en Java desde uno escrito en Scala y viceversa; estas similitudes hacen que se puedan utilizar bibliotecas de Java e incluso las mismas herramientas de desarrollo. El compilador de Scala fue creado por Martin Odersky, el mismo que escribió la referencia para el compilador de Java; este compilador genera el byte code de forma igual que Java.

VISUALBASIC.NET

2001, Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que puede ser visto como la evolución de Visual Basic de Microsoft, este se encuentra implementado en el Framework .NET de Microsoft y la sintaxis es prácticamente igual que la de Visual Basic. Su introducción resultó controvertida pues los cambios realizados en cuanto a la metodología de programación y bibliotecas hacen que no sean compatibles con Visual Basic, sin embargo los entusiastas aseguran que tales cambios valen la pena. Microsoft ha puesto a la disposición de los usuarios herramientas que permiten la migración de VB a VB.NET. Adicionalmente, Microsoft pone a disposición Visual Basic sin costo alguno.

A++

A++ fue desarrollado en el 2002 principalmente para servir a la educación más que para servir a resolver problemas prácticos de programación. Es un lenguaje multiparadigma, funciona para los paradigmas: funcionales, imperativo, orientado a objetos y lógico. Este lenguaje se supone ser muy bueno como una herramienta para comenzar el aprendizaje en programación, ya que tiene patrones que pueden ser aplicados a la programación a la hora de enfrentar el terreno laboral. A + +, con su intérprete en Scheme, Java, C, C + + y Python, ofrece un ambiente ideal para la formación básica en la programación, la aplicación de la

confrontación rigurosa con los elementos esenciales de los lenguajes de programación. Aún cuando este lenguaje es principalmente educativo, es posible escribir programas de aplicación simple en A++ como implementaciones orientadas a objetos de una cuenta de manejo simple y un sistema de gestión de la librería.

F#

Nació en el año 2002, F# (pronunciado F Sharp) es un lenguaje de programación multiparadigma, para la plataforma .NET, que conjunta la programación funcional con las disciplinas imperativas y orientada a objetos. Es una variante del lenguaje de programación ML y es compatible con la implementación Objective Caml. F# fue inicialmente desarrollado por Don Syme de Microsoft Research, pero actualmente está siendo desarrollado por la División de Desarrolladores de Microsoft y será distribuido como un lenguaje totalmente soportado en la plataforma .NET y Visual Studio 2010. (Wikipedia, Wikipedia La enciclopedia Libre).

F# (pronunciado F Sharp) es un lenguaje de programación multiparadigma, para la plataforma .NET, que conjunta la programación funcional con las disciplinas imperativas y orientada a objetos. Es una variante del lenguaje de programación ML y es compatible con la implementación Objective Caml. F# fue inicialmente desarrollado por Don Syme de Microsoft Research, pero actualmente está siendo desarrollado por la División de Desarrolladores de Microsoft y será distribuido como un lenguaje totalmente soportado en la plataforma .NET y Visual Studio 2010. F# es un lenguaje fuertemente tipificado que utiliza inferencia de tipos. Como resultado, los tipos no necesitan estar declarados explícitamente por el programador; estos serán deducidos por el compilador durante el proceso de compilación. Sin embargo, F# también permite la declaración explícita de tipos de datos. Los objetos de .Net son clasificados como tipos mutables y son usados para proveer un modelo de programación orientada a objetos. Los tipos inmutables son usados principalmente por la programación funcional.

GROOVY

Apareció en el 2003. Es un lenguaje de programación orientado a objetos implementado sobre la plataforma Java. Tiene características similares a Python, Ruby, Perl y Smalltalk. Usa una sintaxis muy parecida a Java, comparte el mismo modelo de objetos, de hilos y de seguridad. Desde Groovy se puede acceder directamente a todas las API existentes en Java. El bytecode generado en el proceso de compilación es totalmente compatible con el generado por el lenguaje Java para la Java Virtual Machine (JVM), por tanto puede usarse directamente en cualquier aplicación Java.

BOO

El nacimiento de Boo

- Boo es un lenguaje nacido en el 2003, creado por Rodrigo Barreto di Oliveira. Responde al paradigma de orientación a objetos, y está basado en .NET framework. [1] Es de código abierto, bajo la licencia MIT/BSD. Su sintaxis está basada en Python. Di Oliveira creó este lenguaje de programación para enfrentar un problema

que tuvo en su trabajo donde no podía trabajar con su lenguaje preferido (Python) en el framework establecido por su empresa.

Características básicas de Boo

- **Sintaxis limpia y amigable a la muñeca:** Similar a Python, espolvoreada con azúcar sintáctico, sin necesidad de declarar clases, y automatiza la declaración de variables, la inferencia de tipos y el “casting” de tipos.
- **Expresividad:** Las funciones pueden ser valores de retorno, argumentos, estar en variables y funcionar como objetos. Se trabaja con generadores que son constructores del lenguaje capaces de producir más de un valor cuando están en una iteración. Utiliza duck typing.
- **Extensibilidad:** Utilizando atributos y macros sintácticos, el compilador de Boo reemplaza ciertas líneas con bloques de código que son tediosos para el programador.

Predecesor de Python, C#.

- Lenguaje con orientación a objetos de tipo estático con una sintaxis originaria de Python. Su aporte al avance de la programación es la interferencia de tipo, multimétodos y duck typing.
- Boo forma parte de los lenguajes de código abierto.

RUBY ON RAILS

Este framework de aplicaciones web open source fue escrito utilizando el lenguaje de programación Ruby, permitiendo extender sobre éste la arquitectura Model View Controller, muy utilizada en estos días por los desarrolladores de prácticamente cualquier lenguaje orientado a objetos. Ruby On Rails permite combinar dentro de las aplicaciones escritas en Ruby las conocidas RubyGems, que son los paquetes o bibliotecas creados para Ruby, siguiendo el paradigma del patrón Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. El lenguaje de programación Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby.

Aunque se piensa que es el mismo lenguaje que Ruby o solo una extensión de este, hay quienes lo consideran como un nuevo lenguaje.

VALA

Lanzado en el 2006. Desarrollado bajo varios paradigmas: Orientado a objetos, imperativo y estructurado. Este lenguaje trata de incorporar las características modernas de los lenguajes a C. Su sintaxis es muy similar a la de C#, aunque también permite otras sintaxis de entrada, como el caso de Genie, la cual es muy similar a la sintaxis de Python. Este lenguaje no necesita ser compilado directamente en ensamblador o algún lenguaje intermedio, sino que se compila a C, que luego se compila con la plataforma del compilador

C estándar, para esto usa Valac (Compilador de Vala). En Vala la gestión de adición o eliminación de referencias es automática. En caso a los sistemas operativos, puede ser utilizado en cualquier plataforma que tenga GLib. Hoy en día compite con Mono (implementación abierta de C#) por formar parte activa en el núcleo de Gnome.

Vala es un lenguaje reciente considerado dentro de los paradigmas: orientado a objetos e imperativo. Su principal aporte es la combinación de características de lenguajes modernos con el lenguaje C y hacerlo sin producir costos adicionales. La sintaxis de Vala es muy similar a C# y está diseñado para utilizar las bibliotecas de C, sobre todo basadas en GObject, y sin aumentar los tiempos de ejecución. Otra particularidad de este lenguaje es que su compilador (valac) compila el código a código en C y luego se compila como código C. El compilador de Vala permite escribir código orientado a objetos con requisitos bajos de memoria y con una gran rapidez. Aunque es todavía un trabajo en proceso, el apoyo y uso del lenguaje Vala está creciendo rápidamente.

El nuevo lenguaje de programación VALA, fue creado con el fin de facilitar características de programación modernas a los desarrolladores de GNOME sin hacer un gasto excesivo en los recursos, y sin utilizar una interfaz de aplicación binaria diferente en comparación de las aplicaciones y bibliotecas escritas en el lenguaje C. Su sintaxis es muy similar a la de C#, sin embargo ha sido modificada para que calce de una mejor manera con las librerías basadas en GObject de C. A diferencia de las librerías escritas en C# y Java que no son aceptadas como parte de la plataforma GNOME.

Algunas de las características que fueron agregadas son: expresiones lambda, gestión de memoria asistida, gestión de excepciones, tipos no nulos, entre otras más. Una de sus principales características, y que además ayuda mucho a los programadores, es el hecho de que su compilador, valac, soporta no solamente el código fuente en Vala, sino también el código fuente escrito en Genie(para programadores de python), transformado ambos en código fuente de C. De esta manera, los programadores de python no tendrán que aprender este nuevo lenguaje si lo desearan usar.

CLOJURE

Es un lenguaje funcional moderno, concurrente, basado en Lisp e implementado sobre la Máquina Virtual de JAVA. Fue creado a finales de 2007 por Rich Hickey. Propone unir las ventajas que tienen los lenguajes funcionales y los orientados a objetos para otorgar al lenguaje estabilidad y concurrencia. Sigue la filosofía de que los lenguajes deben ser definidos sobre una plataforma existente, en vez que el lenguaje sea su propia plataforma.

Aunque el lenguaje está basado en Lisp, su diseño no toma en cuenta la compatibilidad con los distintos dialectos de Lisp. Además, no realiza optimizaciones en llamadas de c.

GO

Una nueva creación desarrollada por la compañía Google, el lenguaje de programación está inspirado en los ya muy conocidos lenguajes: C y C++, basándose en su sintaxis y muchas otras características. Sin embargo, GO! Incluye muchas ventajas de los lenguajes Pascal, Modula y Oberon, entre ellas, las formas de trabajar las declaraciones y los paquetes extra.

Actualmente, GO! Sólo se encuentra disponible para los sistemas operativos GNU/Linux (en la mayoría de sus distribuciones) y Mac OS. Los usuarios de Windows tendrán que esperar un tiempo para que se libere este lenguaje en dicho sistema.

Uno de los principales puntos a favor de GO!, es la velocidad con la que compila los programas, menos de un segundo, y la velocidad en la que dura al cargar todas sus librerías, menos de 10 segundos. Otro de sus puntos a favor, para algunos usuarios, es que es un lenguaje OPEN SOURCE, lo cual permite a los usuarios ver el código fuente del lenguaje y posiblemente, crear nuevas versiones como se hizo con Linux. También se creó como un lenguaje multi-paradigma, incorporando características de todos los paradigmas existentes: imperativo, lógico, funcional y orientado a objetos. Además que soporta la programación concurrente.

Una nueva creación desarrollada por la compañía Google, el lenguaje de programación está inspirado en los ya muy conocidos lenguajes: C y C++, basándose en su sintaxis y muchas otras características. Sin embargo, GO! Incluye muchas ventajas de los lenguajes Pascal, Modula y Oberon, entre ellas, las formas de trabajar las declaraciones y los paquetes extra.

Actualmente, GO! Sólo se encuentra disponible para los sistemas operativos GNU/Linux (en la mayoría de sus distribuciones) y Mac OS. Los usuarios de Windows tendrán que esperar un tiempo para que se libere este lenguaje en dicho sistema. Uno de los principales puntos a favor de GO!, es la velocidad con la que compila los programas, menos de un segundo, y la velocidad en la que dura al cargar todas sus librerías, menos de 10 segundos. Otro de sus puntos a favor, para algunos usuarios, es que es un lenguaje OPEN SOURCE, lo cual permite a los usuarios ver el código fuente del lenguaje y posiblemente, crear nuevas versiones como se hizo con Linux. También se creó como un lenguaje multi-paradigma, incorporando características de todos los paradigmas existentes: imperativo, lógico, funcional y orientado a objetos. Además que soporta la programación concurrente.

Lanzado en el 2009. Es un lenguaje de programación de código abierto, concurrente y compilado, desarrollado por Google para incursionar en el campo de los lenguajes de programación. Es muy similar a los lenguajes C y C++, pero con características modernas de la computación. También incorpora elementos de Python, así como propiedades de los lenguajes Pascal, Modula y Oberon. Go se convierte en uno de los pocos lenguajes que se han creado en los últimos años. Diseñado principalmente para el desarrollo de aplicaciones web y para ser utilizado en los servidores de Google, con lo cual se garantiza obtener el mejor rendimiento de este, para así ofrecer un mejor servicio a sus clientes. Dentro de sus características se encuentra la capacidad de sacar provecho de los procesadores de múltiples núcleos para realizar múltiples tareas en paralelo, así como características modernas como el recolector de basura. Go está disponible solo para los sistemas operativos GNU/Linux y Mac OS.

HISTORIA DE LOS LENGUAJES Segunda Década Siglo XXI.

Los lenguajes de programación son un medio utilizado para comunicarse con las computadoras y están conformados por reglas de sintaxis y símbolos. A través de los años han existido gran cantidad de lenguajes que se diseñan para ser utilizados en distintas ramas de la computación y es de suma importancia conocer las características más sobresalientes que poseen, por lo tanto, el siguiente documento explicará características generales sobre distintos lenguajes creados en la segunda década del siglo XXI. Presentará información como, por ejemplo: el año de creación, sus creadores, lenguajes en los que se basaron para crearlo, propiedades que lo distinguen, entre otras.

CHAPEL

Es un lenguaje de programación en paralelo creado en el año 2010 por Cray Inc. Fue creado como un proyecto para incrementar la productividad de las super-computadoras. Se basó en HPF, ZPL y Cray MTA para sus ideas de paralelismo. Su principal objetivo es que programar en paralelo sea tan fácil como en secuencial. Sus influencias fueron Ada, C, Fortran, HPF, ZPL y Cray MTA.

ZIMBU

Es un nuevo lenguaje que se encuentra en una etapa experimental, creado por Bram Moolenaar, creador del editor de texto Vim. Moolenaar pretende que el lenguaje que se escriba sea sencillo, conciso, fácil de leer y rápido de escribir y que no sean necesarias tantas herramientas gráficas. Según Moolenaar "La programación cuanto más breve, mejor".

La portabilidad es una consideración clave. Aunque Zimbu es un lenguaje compilado, Zimbu y el compilador expresan las salidas en código ANSI C, permitiendo que los binarios que se construyan corran sólo en plataformas con un compilador C nativo. Por otra parte, es muy rico en funciones aunque no están desarrolladas todas las especificadas ya que, en la actualidad, este proyecto se encuentra en una fase Alfa.

RUST

Este lenguaje fue creado por Mozilla en 2014 como alternativa a C++. Ellos mismos lo definieron como un lenguaje centrado en el rendimiento y la programación paralela. Fue creado a partir de scratch, emplea un diseño moderno de programación y lo soportan un gran número de librerías. En la última encuesta realizada por StackOverflow fue elegido como el lenguaje preferido por sus usuarios. Es un lenguaje de programación compilado, de propósito general y multiparadigma que está siendo desarrollado por Mozilla. Ha sido diseñado para ser un lenguaje seguro, concurrente y práctico". Soporta programación funcional pura, por procedimientos, imperativa y orientada a objetos.

El lenguaje surgió de un proyecto personal desarrollado por Graydon Hoare (trabajador de Mozilla), quién empezó a trabajar en él en 2006; y Mozilla se involucró en este proyecto en 2009 y lo dio a conocer oficialmente en 2010. El mismo año, el trabajo pasó del compilador inicial (escrito en OCaml) al compilador autocontenido, escrito en sí mismo, en Rust. Conocido como rustc, se compiló a sí mismo en 2011. El compilador auto contenido usa LLVM como su back-end. La primera versión alfa numerada del compilador de Rust

apareció en enero de 2012. La versión actual es la 1.0, que fue lanzada el 9 de enero de 2015.

Según la política de Mozilla, Rust es desarrollado de forma totalmente abierta y solicita la opinión y contribución de la comunidad. El diseño del lenguaje se ha ido perfeccionando a través de las experiencias en el desarrollo del motor de navegador, Servo y el propio compilador de Rust. Aunque es desarrollado y patrocinado por Mozilla y Samsung, es un proyecto comunitario. Una gran parte de las contribuciones proceden de los miembros de la comunidad.

Es un lenguaje de programación enfocado en tres objetivos: seguridad, velocidad y concurrencia, además fue diseñado con un propósito general y multiparadigma. Desarrollado por Graydon Hoare, junto con la colaboración de Mozilla, lo cual dio al traste con su lanzamiento en 2010. Rust es desarrollado de forma totalmente abierta y solicita la opinión y contribución de la comunidad asociada.

La sintaxis de este lenguaje es muy similar a la de C y C++, con bloques de código delimitados con llaves y estructuras de control de flujo tales como if, while, for, etc. Además de esto es un lenguaje multiplataforma, desarrollado para Windows, Linux, OSX.

Es un lenguaje de propósito general creado en 2010 por Graydon Hoare y luego Rust Project Developers. Se caracteriza por ser seguro, concurrente, soportar paradigma imperativo y funcional, código abierto, posee un buen control de distribución de memoria, no permite punteros nulos, inferencia de tipos y soporta los traits y typestate; los traits se utilizan para proporcionar herencia. Su sintaxis se asimila a la de C y C++ y fue influenciado por Common Lisp, C++, OCaml, Ruby, Swift y Haskell.

CRYSTAL

Fue desarrollado por Ary Borenszweig, según este, este nuevo lenguaje de programación no solo posee la riqueza de lenguajes de programación existentes, sino que construye en base a sus mejores características.

“Sí, hay muchos lenguajes. Sin embargo, Crystal combina algunas características presentes en otros lenguajes de una forma que ningún otro lenguaje lo hace”, dijo Borenszweig, quien además añade “Tiene macros para evitar códigos repetitivos. Y tiene una forma muy fácil para usar las bibliotecas existentes de C”.

Una particularidad especial de este lenguaje es que muy a menudo se utiliza para construir compiladores. Se cree que este lenguaje en un futuro llegará a reemplazar a Ruby.

Es un lenguaje orientado a objetos creado por Ary Borenszweig en 2011. Su sintaxis se basó en la de Ruby, posee inferencia de tipos, es de código abierto, corre bajo LLVM y posee un sistema macro. El objetivo era crear un lenguaje con la productividad de Ruby, pero con la rapidez y eficiencia de un lenguaje compilado. Sus influencias fueron Ruby, C, Rust, Go, C# y Python.

CEYLON

Es un lenguaje de programación orientado a objetos, desarrollado por Red Hat. El lenguaje en sí mismo toma mucha de su sintaxis de Java. Gavin escribe en su blog que el Lenguaje Java ha quedado un poco viejo; por lo tanto el diseño un lenguaje llamado Ceylon que tiene como principal objetivo actualizar el lenguaje Java.

Los demás objetivos de Ceylon incluyen:

Ser fácil de aprender tanto para desarrolladores Java como C#, eliminar la verbosidad de Java pero manteniendo la legibilidad, mejorar el tipado estático de Java, proveer una expresión declarativa para eliminar dependencias con xml, apoyar y fomentar un estilo de programación más funcional, proveer soporte para meta-programación y proveer construcción modular.

Es un nuevo lenguaje de programación concebido como un proyecto de Red Hat, liderado por Gavin King. Es un lenguaje Orientado a Objetos, para muchos se le tiene gran respeto ya que la expectativa es que destrone a Java en un par de años, sin embargo en palabras de Gavin King, "Ceylon no es Java, sino un nuevo lenguaje profundamente influenciado por Java, proyectado por personas que son fanáticos declarados de Java. Java no morirá en breve; así que nada lo está matando".

Ceylon además de ser Orientado a Objetos soporta funciones de orden superior con un estilo de sintaxis típico de C.

Es un lenguaje de programación creado en 2011 por Gavin King y Red Hat. Es orientado a objetos y fue influenciado mayormente por Java, pero también por C++, Smalltalk y Eiffel. Su objetivo principal era actualizar el lenguaje Java, sus sintaxis son muy parecidas, sólo que Ceylon le agrega más características. También querían que fuera un lenguaje fácil de aprender y además poder fomentar la programación funcional. Su sistema de tipos permite a una variable tener más de un tipo, por ejemplo, que pueda ser entero o string.

DART (originalmente llamado Dash)

Es un lenguaje de programación nacido en los laboratorios de Google. Este fue creado en el año 2011, bajo los estándares de código abierto, con el objetivo de brindar un lenguaje de programación más moderno y rápido dentro de los navegadores web. Al igual que Processing, Dart es muy utilizado en los torneos y cursos de verano impartidos en las oficinas de Googleplex.

Fue desarrollado para ser el sustituto de JavaScript. El objetivo es disponer de un lenguaje más moderno, rápido dentro de los navegadores web, flexible y fácil de aprender, como JavaScript, pero que sea lo suficientemente potente y que no presente los problemas que tiene JavaScript cuando lo que se está programando tiene miles y miles de líneas de código.

Dart nació como un lenguaje estructurado y permite desarrollar aplicación web, tanto en formato de páginas como webapps puras, muy similares a las que corre Google Chrome

dentro del navegador web. La creciente plataforma de desarrollo Google App Engine, nos da todas las herramientas necesarias para aprender el lenguaje de programación Dart.

Si bien JavaScript lleva 20 años dentro del mundo web, este lenguaje todavía posee algunos resultados no muy favorables. El objetivo de Dart es crecer a la par del avance de los navegadores web, teniendo a éstos como único objetivo y así poder brindar, bajo una herramienta sencilla, soporte para proyectos de toda envergadura que estén orientados a la gran red de redes. El navegador Chromium incluye un compilador específico para Dart, y a su vez posee todo el soporte necesario para poder compilar Dart bajo el paradigma ECMAScript 3 (propio de JavaScript).

Dart-SDK es un kit de desarrollo que incluye todas las librerías que existen actualmente para Dart, y posee una ayuda completa para la ejecución de comandos de compilación hacia Dart, y Dart-toJavaScript.

Es un lenguaje de programación desarrollado por Google, como un lenguaje fácil de aprender y de escalar. Originalmente fue llamado Dash, el principal objetivo de este lenguaje es ofrecer mejores resultados mediante alternativas a algunos problemas que se presentan con JavaScript (aunque esto no se dio a conocer directamente).

En su desarrollo Dart también se pensó en que fuera óptimo para desarrollar proyectos grandes de Google, con muchísima más seguridad pero acompañado de mucha simpleza. Es importante mencionar que Dart es un lenguaje de código abierto, estructurado y flexible, orientado a objetos, basado en clases, con herencia simple y soporte de interfaces, clases abstractas y tipado opcional de datos. Fue desarrollado en especial por Lars Bak y Kasper Lund.

Es un lenguaje de propósito general desarrollado por Google y diseñado por Lars Bak y Kasper Lund en 2011. Es usado para el desarrollo de aplicaciones móviles y programación web. Dart usa el estilo de sintaxis de C# y tuvo influencias de JavaScript, Java, CoffeeScript y Go. Fue creado para arreglar varios problemas que presentaba JavaScript. Este lenguaje es orientado a objetos con herencia múltiple. La máquina virtual de Dart tiene archivos esenciales llamados snapshots que se usan para almacenar objetos.

OPA

Creado en el 2011 es un lenguaje de programación diseñado para el desarrollo web, que promete el desarrollo ágil de aplicaciones web seguras, altamente escalables y distribuidas. Hoy en día es muy difícil el desarrollo web debido a la cantidad de tecnologías que hace falta conocer: HTML, CSS, JavaScript y DOM en el cliente; Java o PHP y SQL en el servidor. Opa promete ofrecer un entorno único de programación que generará el código necesario para la parte cliente o para la parte servidor.

Es un lenguaje que funciona primordialmente para el desarrollo web en la nube, es funcional y fue desarrollado por MLstate en 2011. Es de tipado estático, de código abierto, ejecución distribuida, posee base de datos propia y posee gran seguridad. Sus influencias son OCaml, Erlang y JavaScript. Se destaca por funcionar de manera compilado y no interpretado, que

es como todos los demás lenguajes web funcionan. No está diseñado para aplicaciones no-web.

ELIXIR

Nace como respuesta a una necesidad especial, aparece como la solución para aquellas personas que argumentan que Erlang es un lenguaje de los más difíciles para programar. Este nuevo lenguaje corre sobre la Máquina Virtual de Erlang, está clasificado como un lenguaje perteneciente al paradigma de la programación funcional y que posee una sintaxis parecida a la Ruby.

Este lenguaje fue desarrollado principalmente por José Valim y su sencillez permite la extensibilidad a través de la construcción de “DSLs” o Lenguajes de Dominio Específico (lenguajes creados con el único objetivo de solucionar problemas en específico, no pensados de manera general).

Fue creado en 2011 con el objetivo de crear un lenguaje de programación parecido a Erlang, pero con una sintaxis menos complicada. Es un lenguaje concurrente, con aislamiento de procesos, de propósito general y funcional. Usa la plataforma Erlang (BEAM) y fue mayormente influenciado por Erlang, así mismo también por Clojure, LFE y Ruby; su sintaxis es muy parecida a la de este último. Elixir se caracteriza por ser fuerte en la rama de las telecomunicaciones, aplicaciones como Pinterest y Moz lo utilizan. También es conocido por su utilización en desarrollo web. Además, provee herramientas para búsqueda de patrones y supervisión de procesos. Aunque se empezó a crear en 2011 su versión 1.0 salió hasta el año 2014.

RED

Es un lenguaje creado en el 2011 por Nenad Rakocevic para superar ciertas limitaciones que presenta el lenguaje Rebol. Se caracteriza por ser imperativo y funcional, puede ser utilizado tanto para programación de alto nivel como de bajo nivel y soporta fuertemente la metaprogramación y lenguajes específicos de dominio. El lenguaje contiene dos partes: Red/System y Red; Red/System es muy parecido a C, pero en una estructura como la de Rebol y Red es un lenguaje en donde su representación primaria es una estructura de datos de un tipo primitivo del lenguaje; ambos son de código abierto. Sus influencias fueron Rebol, Scala y Lua.

XTEND

Es un lenguaje de alto nivel creado en 2011 por Sven Efftinge y Sebastian Zarnekow como parte de los proyectos de Eclipse. Soporta paradigmas orientado a objetos, imperativo y funcional, es tipado estático, es compilado en código Java y corre bajo la máquina virtual de Java, posee código abierto y tiene inferencia de tipos. Xtend trata de tomar lo mejor que posee Java y quita partes tediosas de su sintaxis, así mismo, le agrega nuevas características y código más fuerte. Es un lenguaje totalmente compatible con Java. Fue influenciado mayoritariamente por Java, y también por Scala, Groovy y Smalltalk.

JULIA

Julia fue desarrollado para ayudar a los científicos de datos (ese perfil profesional tan necesario en los CPD y que une conocimientos matemáticos, informáticos y de estadística). Se define como un lenguaje de alto nivel y dinamismo para necesidades de informática técnica.

Aunque más lentamente que otros lenguajes de esta lista, Julia va poco a poco engrosando su base de usuarios. De hecho dominarlo es una de las habilidades más valoradas y mejor pagadas en algunos mercados laborales como el americano. Es un lenguaje de programación homoicónico, multiplataforma y multiparadigma de tipado dinámico de alto nivel y alto desempeño para la computación genérica, técnica y científica, con una sintaxis similar a la de otros entornos de computación similares. El nombre del lenguaje Julia fue puesto en honor a Gaston Julia, un matemático francés que descubrió los fractales.

Dispone de un compilador avanzado (JIT), mecanismos para la ejecución en paralelo y distribuida, además de una extensa biblioteca de funciones matemáticas. La biblioteca, desarrollada fundamentalmente en Julia, también contiene código desarrollado en C o Fortran, para el álgebra lineal, generación de números aleatorios, procesamiento de señales, y procesamiento de cadenas. Adicionalmente, la comunidad de desarrolladores de Julia contribuye con la creación y distribución de paquetes externos a través del gestor de paquetes integrado de Julia a un paso acelerado. Julia es el resultado de la colaboración entre las comunidades de IPython y Julia, provee de una poderosa interfaz gráfica basada en el navegador para Julia.

Es un lenguaje de programación conocido principalmente por su amplia gama de bibliotecas matemáticas, siendo a la vez un lenguaje de alto nivel y de alta performance. Desarrollado por Stefan Karpinski, quien a través de la necesidad de mejoras en la investigación en el MIT se vio en la tarea de crear un lenguaje único que “haga todo bien”. Es un lenguaje multiplataforma y multiparadigma.

Una de sus características más importantes es que se pueden utilizar Macros como en LISP, y otras funcionalidades de metaprogramación (programación que consiste en escribir programas que escriben o manipulan otros programas).

Es un lenguaje creado por Jeff Bezanson y Stefan Karpinski en honor a Gaston Julia, en el año 2012. Se caracteriza por ser tipado dinámico de alto nivel, su amplia librería de funciones matemáticas que integra librerías de C y Fortran, por su buena comunicación con otros lenguajes, por poseer una Shell muy poderosa al ser diseñado en paralelo y distribuida, tener corrutinas, tener soporte para Unicode y ser libre y de código abierto. Está fuertemente influenciado por C y Fortran. Sus creadores querían que funcionara para resolver prácticamente cualquier tarea. Su compilador Jit es bastante rápido y supera muchos lenguajes como Python. Posee un entorno gráfico llamado IJulia para crear gráficos de funciones matemáticas.

ELM

Su primera aparición fue en el año 2012 y fue diseñado por Evan Czaplicki. Es un lenguaje puramente funcional con características específicas para la creación de aplicaciones web, ya que fue pensado para ejecutarse en el navegador. Su mayor influencia fue Haskell, aunque también se basó en Standard ML, OCaml y F#. Todos los valores en Elm son inmutables, quiere decir que un valor no puede cambiarse cuando ya fue creado. También, sus tipos son estáticos y posee un sistema que permite al usuario partir el código en módulos; partes más pequeñas. Además, permite la comunicación entre Elm y JavaScript y posee una librería que permite escribir HTML y CSS dentro de Elm.

TYPESCRIPT

Es un lenguaje de programación perteneciente a la gama de lenguajes de código abierto, posee herramientas de Programación Orientada a Objetos. Este lenguaje fue desarrollado por Microsoft, en especial por el arquitecto Anders Hejlsberg.

TypeScript convierte su código en JavaScript común, esta técnica es llamada Superset (es cuando un lenguaje se escribe sobre otro). Además este mismo se incluye como lenguaje de programación de primer nivel en Microsoft Visual Studio 2013 Update 2 y posteriores, junto a C# y otros lenguajes de Microsoft. Una extensión oficial permite a Visual Studio 2012 soportar también TypeScript.

Es un lenguaje de código abierto desarrollado por la compañía Microsoft en el año 2012. Se caracteriza por ser un superset de JavaScript y por extender su sintaxis, el compilador está escrito en TypeScript y es compilado a JavaScript, es uno de los lenguajes de programación de alto nivel en Microsoft Visual Studio 2013, soporta programación genérica, convierte el código en JavaScript común, fue diseñado para desarrollar aplicaciones de gran escala y soporta paradigmas: imperativo, funcional, genérico, multi-paradigma y orientado a objetos. Fue creado para arreglar algunos problemas complejos que tenía JavaScript y como un adelanto de lo que sería ECMAScript 2015. Fue influenciado por JavaScript, Java y C#.

KOTLIN

Se define como un lenguaje de programación de propósito general para la máquina virtual de Java y Android. Según sus desarrolladores; "Kotlin se diseñó con la idea de ser un lenguaje de programación pragmático para JVM y Android, que combina la Orientación a Objetos y características de los lenguajes funcionales. Su enfoque es la interoperabilidad, la seguridad, la claridad y el soporte de las herramientas".

Este lenguaje cabe mencionar pertenece al grupo de lenguajes de código abierto, y su empresa desarrolladora (Jetbrains) tiene soporte a tiempo completo del lenguaje y una comunidad de colaboradores que crece día con día.

Es un lenguaje de código abierto creado en el año 2012 por la empresa JetBrains. Es de propósito general, tipado estático y corre bajo la máquina virtual de Java. Este lenguaje fue creado para combinar la orientación a objetos y las características que poseen los lenguajes funcionales. Permite operar con código ya escrito en Java, es compatible con Android y posee integración de Maven. Se usa mayormente para la creación de aplicaciones Android.

Sus influencias fueron Java, Scala, Groovy y C#. También se quería que fuera parecido a Scala, pero con la rapidez de Java. Soporta procedimientos con funciones y la inferencia de tipos.

WYVERN

Es un lenguaje de programación para aplicaciones web y móviles creado en el año 2013 desarrollado por Jonathan Aldrich. Su principal objetivo es proveer una forma de usar múltiples lenguajes en el mismo programa de forma segura para la creación de aplicaciones web. Así se puede utilizar el lenguaje que más convenga dependiendo de la función. Se centra mucho en solucionar problemas de seguridad y así proteger las aplicaciones de los ataques de inyección de código. Aún se encuentra en un nivel básico ya que sigue en desarrollo.

Wyvern determina el sublenguaje que se está usando en el programa mediante los tipos de datos que el programador está manipulando. Los tipos especifican el formato de los datos, ya sean caracteres alfanuméricos o estructurados de datos más complejos, como podría ser una petición de datos a una base de datos de una página Web. Con ello se evitan cosas como tener que pasar cadenas de texto como variables entre lenguajes y así poder aumentar la seguridad.

Este concepto que se usa Wyvern no es sencillo, y desde luego no es algo que los usuarios vayan a poder adaptarse fácilmente, sin embargo, puede llegar a ser un avance muy importante para el desarrollo de lenguajes y prácticas seguras que cierren la puerta a hackers.

Uno de los puntos que juegan en contra, es el hecho de que Wyvern está siendo financiado por NSA, quien ha sido protagonista en los medios debido a muchos escándalos relacionados al espionaje electrónico. Por ello, muchos han salido con especulaciones nada favorables para el lenguaje. También, el hecho de tener una documentación poco amigable, hace que muchos no sean capaces de aportar al proyecto, recordando que este es de código abierto.

HACK

Hack también es obra de otro gigante de internet, en este caso Facebook. Creado en 2014, en estos momentos la compañía de Zuckerberg está implementando y migrando todo su código PHP a este lenguaje.

Facebook también ha lanzado una versión de Hack de código abierto como parte de su proyecto HHVM, una tecnología para alcanzar velocidades superiores utilizando scripts PHP.

Es un lenguaje de programación de alto nivel para la Máquina Virtual HipHop (HHVM), aunque se dice que fue creado como un dialecto del PHP, actualmente se dice que es un lenguaje aparte, debido a varios cambios realizados en él. La implementación del lenguaje es de código abierto, licenciado bajo las licencias BSD y Creative Commons Attribution 3.0.234 Fue introducido en marzo del 2014.

Hack permite a los programadores utilizar tanto tipado dinámico y tipados estáticos. Este tipo de sistema de tipos se le llama tipificación gradual, y también se aplica en otros lenguajes de programación como ActionScript. El Sistema de tipeo Hacks permite tipeos que se especifican para argumentos de la función, los valores de retorno de funciones y propiedades de la clase; Sin embargo, los tipos de variables locales no se pueden especificar.

Es un lenguaje de programación desarrollado por Facebook como un dialecto de PHP, con características que lo hacen más sencillo y más rápido, Hack nace de la necesidad de Facebook de aumentar su capacidad de atención a los usuarios. Una de las principales diferencias con PHP es el uso de tipos, por ejemplo en los parámetros si un dato posee tipo estamos utilizando la sintaxis de Hack, pero sino, se estaría realizando un manejo prácticamente igual al de PHP, por lo demás se puede asegurar que un código en Hack muy posiblemente se ejecute sin problemas en PHP. La estructura de este lenguaje también toma referencias de otros lenguajes un poco más conocidos como C y C++.

Cabe mencionar que Hack también es un lenguaje de código abierto.

Es un lenguaje de código abierto creado en el año 2014 por la compañía Facebook para la HHVM, entre sus creadores se destacan Julien Verlaquet, Bryan O'Sullivan, Alok Menghrajani y Drew Paroski. Fue creado como un dialecto de PHP; influenciado por el mismo, Java y C# y diseñado principalmente para el desarrollo de páginas web complejas. Permite a los usuarios usar un sistema de tipos estático y dinámico; se le llama tipificación gradual. Las funciones en Hack se declaran como tipos y poseen herencia múltiple.

SWIFT

Swift es un lenguaje de programación multiparadigma creado por Apple para sustituir a Objective-C enfocado en el desarrollo de aplicaciones para iOS y Mac OS X. Fue presentado en WWDC2014 y está diseñado para integrarse con los Frameworks Cocoa y Cocoa Touch, puede usar cualquier biblioteca programada en Objective-C y llamar a funciones de C. También es posible desarrollar código en Swift compatible con Objective-C bajo ciertas condiciones. Swift tiene la intención de ser un lenguaje seguro, sencillo, de desarrollo rápido y conciso. Usa el compilador LLVM incluido en Xcode6. En el año 2015 pasó a ser de código abierto.

Swift es un lenguaje fuertemente tipado, aunque su declaración no siempre es necesaria gracias a su capacidad de inferir tipos. Los tipos de datos se dividen principalmente en dos grupos. Los Tipos de valor, y los Tipos por referencia, se diferencian principalmente en cómo son asignados. Ya sea por valor, donde se realiza una copia a la información o a su referencia que usa una instancia compartida mutable aún si se usa en constantes.

Es un lenguaje potente creado por Apple. Pertenece al grupo de lenguajes de código abierto. Fue creado con el objetivo de ser un lenguaje multiparadigma enfocado al desarrollo de aplicaciones para iOS y Mac. Este lenguaje toma provecho de varios lenguajes como Objective-C (principalmente trata de suplantarlos), Haskell, Rust, Ruby, Python.

Swift es un lenguaje que está enfocado para todos aquellos que están empezando en el mundo de la programación y también para ser la base del futuro desarrollo de apps móviles.

Es un lenguaje para aplicaciones iOS y Mac OS X creado en 2014 por Chris Lattner y la compañía Apple. Está diseñado para los frameworks Cocoa y Cocoa Touch, puede llamar a funciones de C, usa compilador LLVM, en 2015 pasó a ser de código abierto, tiene un sistema para encontrar errores comunes de programación como los punteros nulos y soporta los paradigmas: imperativo, funcional, bloque de código y multi-paradigma. Swift es una alternativa al lenguaje de Objective-C, es un lenguaje enfocado en aquellos que quieren empezar a programar, usa una contabilidad de referencia automática para el manejo de memoria, utiliza inferencia de tipos y posee programación genérica. Sus tipos de datos se dividen en tipos de valor y de referencia. Se basó en C#, D, Haskell, Object-C, Python, Rust y Ruby; se asimila en muchas formas a C.

PIXIE

Es un lenguaje funcional creado por Timothy Baldridge en el año 2015. Fue fuertemente influenciado por Clojure y Lisp; su librería estándar está fuertemente inspirada por Clojure. Algunas características que posee son: que sus objetos son inmutables e incluye Jit de rastreo.

ALBATROSS

Es un lenguaje de programación de uso general con verificación estática. Fue creado en 2015 y se basó en Eiffel y Coq. Sirve como asistente de pruebas y probador de teoremas. Se diferencia de los demás lenguajes ya que tiene la posibilidad de escribir arbitrariamente en código imperativo y funcional, así mismo lo verifica, compila y ejecuta.

RING

Es un lenguaje dinámico creado en 2015. Es de propósito general y soporta los paradigmas de programación imperativo, orientado a objetos, funcional y metaprogramación. El lenguaje es bastante portable y se puede utilizar para crear aplicaciones de consola, web o móviles. Es un lenguaje bastante simple, rápido y puede ser incorporado en proyectos de C o C++.

ODEN

Es un lenguaje de programación que nace por parte de un aficionado de Go, que obtuvo algunas complicaciones con este prometedor lenguaje y que decidió por su parte agarrar todo lo bueno de Go, y agregar sus nuevas características.

Según Wickström, quien es el creador de Oden, su lenguaje es bueno en las tareas dentro del ámbito de Go: servidores web, servicios Backend, y herramientas de línea de comandos, por nombrar algunas. Es bueno también para construir bibliotecas que provean estructuras de datos definidas por el usuario general, algoritmos genéricos, y abstracciones de control-flow.

HISTORIA DE LOS LENGUAJES Tercer Década Siglo XXI.

MICROSOFT POWER FX Microsoft Power Fx

Es un lenguaje de programación funcional, declarativo, de tipo fuerte y de uso general. Fué anunciado en el 2021 y se basa principalmente en el bajo uso de código, ya que se puede trabajar directamente en una barra de fórmulas similar a Excel o la ventana de texto de Visual Studio Code. De esa manera se consigue que este lenguaje pueda ser usado por personas que tengan nula o poca experiencia en el desarrollo de código, así como por personas bastante experimentadas en este aspecto. Está altamente influenciado por lenguajes de programación como Pascal y Miranda.

CARBON

Anunciado por primera vez en la conferencia de Cpp North en Toronto, Canadá, en julio de 2022, por el ingeniero de Google Chandler Carruth. Es un lenguaje experimental, de código abierto, impulsado por Google y diseñado para ser el sucesor del lenguaje de programación C++. Tiene como objetivo principal corregir algunas deficiencias percibidas en C++, como lo son la legibilidad, la interoperabilidad bidireccional y la curva de aprendizaje.

MOJO

Presentado en el 2023. Mojo es un lenguaje de programación que surge principalmente para abordar problemas relacionados a la Inteligencia Artificial. Destaca por tener una velocidad de ejecución alta, gran compatibilidad con Python y ser de tipado fuerte, así como poder gestionar la memoria de forma avanzada. En resumen, se puede ver como una combinación entre la facilidad de uso de Python y la eficiencia de C++.