

Claudio Cabrera

Programación básica

Primer borrador



Guia rapida y fundamentos de programación

Que es programar?

La programación: es el proceso de diseño, creación, prueba y mantenimiento de software. Los programadores utilizan un lenguaje de programación para escribir código, que es un conjunto de instrucciones que indican al ordenador qué hacer. El código se utiliza para crear programas, aplicaciones y sistemas que realizan una variedad de tareas, desde jugar videojuegos hasta controlar robots industriales. La programación también se utiliza para automatizar tareas y analizar datos. La programación se divide en varias áreas, incluyendo el desarrollo de aplicaciones, la programación de sistemas, la programación web y la programación de inteligencia artificial. Cada área tiene sus propios lenguajes de programación y herramientas específicas. La programación es una habilidad valiosa en la era digital actual, ya que la mayoría de los aspectos de nuestra vida están cada vez más conectados a la tecnología y los ordenadores. Los programadores son responsables de crear los programas y sistemas que utilizamos todos los días y son altamente valorados en el mercado laboral.

Que dispositivos usan programación

La programación de software, se puede utilizar en una variedad de dispositivos, incluyendo:

- 1. Computadoras de escritorio y portátiles:** se utilizan para escribir, editar y ejecutar código en sistemas operativos como Windows, MacOS y Linux.
- 2. Teléfonos móviles:** los sistemas operativos de los teléfonos móviles como Android e iOS permiten a los desarrolladores crear aplicaciones móviles.
- 3. Tablets:** similares a los teléfonos móviles, las tablets también pueden ejecutar aplicaciones móviles.
- 4. Servidores:** la programación de software se utiliza para crear y ejecutar aplicaciones en servidores que proporcionan servicios a través de una red.
- 5. Dispositivos IoT (Internet de las cosas):** los dispositivos IoT, como las cámaras de seguridad, los termostatos inteligentes, y los relojes inteligentes, utilizan software para conectarse y comunicarse con otros dispositivos y servicios en la red.
- 6. Robots y drones:** la programación de software se utiliza para controlar y automatizar robots y drones.
- 7. Automóviles:** la programación de software se utiliza para controlar y monitorear el rendimiento de los automóviles, así como para implementar características avanzadas como el control de crucero y la conducción autónoma.
- 8. placas AVR:** Es una plataforma fácil de usar y versátil para desarrollar proyectos interactivos y robóticos. Se utiliza para controlar y automatizar diferentes dispositivos y sensores mediante el uso de programación.

Editores de programación mas conocidos

- **Visual Studio Code** - es un editor de código multiplataforma desarrollado por Microsoft con una gran variedad de características y una gran comunidad de desarrolladores.
- **Sublime Text** - es un editor de código ligero pero potente con una interfaz de usuario personalizable y una gran cantidad de características avanzadas.
- **Atom** - es un editor de código de código abierto desarrollado por GitHub con una gran variedad de características y una gran comunidad de desarrolladores.
- **Notepad++** - es un editor de código para Windows con una interfaz de usuario limpia y una gran cantidad de características avanzadas.
- **IntelliJ IDEA** - es un editor de código de Java desarrollado por JetBrains con una gran variedad de características y una gran comunidad de desarrolladores.
- **Eclipse** - es un editor de código de código abierto para Java con una gran variedad de características y una gran comunidad de desarrolladores.
- **Visual Studio** - es un entorno de desarrollo integrado (IDE) de Microsoft con una gran variedad de características y una gran comunidad de desarrolladores.

- **XCode** - es el entorno de desarrollo integrado (IDE) para MacOS de Apple para desarrollar aplicaciones para iOS, iPadOS, macOS, watchOS y tvOS.
- **Brackets** - es un editor de código de código abierto desarrollado por Adobe con un enfoque en el diseño y la interfaz de usuario.
- **Vim** - es un editor de código de línea de comandos altamente personalizable y configurable que se ejecuta en varios sistemas operativos.
- **GNU emacs** - es un editor de código de código abierto altamente personalizable y configurable con una gran cantidad de características avanzadas.
- **WebStorm** - es un editor de código JavaScript desarrollado por JetBrains con una gran variedad de características y una gran comunidad de desarrolladores.
- **PyCharm** - es un editor de código Python desarrollado por JetBrains con una gran variedad de características y una gran comunidad de desarrolladores.
- **Kate** - es un editor de código para sistemas operativos de tipo Unix como Linux y MacOS, tiene una interfaz de usuario limpia y una gran cantidad de características avanzadas.
- **TextMate** - es un editor de código para MacOS con una interfaz de usuario limpia y una gran cantidad de características avanzadas.
- **Coda** - es un editor de código para MacOS con una interfaz de usuario limpia y una gran cantidad de características avanzadas.

Frameworks de Programación

Framework de programación:

Es un conjunto de herramientas y librerías que se utilizan para desarrollar aplicaciones o programas de manera más fácil y rápida. Estos frameworks proporcionan una estructura y un conjunto de reglas predefinidas para facilitar el desarrollo de software, lo que permite a los desarrolladores centrarse en escribir código específico para sus proyectos en lugar de tener que construir todo desde cero. Los frameworks también suelen incluir componentes y funcionalidades comunes que se utilizan en muchos proyectos, como la gestión de bases de datos, la autenticación de usuarios y la gestión de rutas, lo que ayuda a reducir la cantidad de código que se tiene que escribir.

Frameworks de Programación Populares

Aquí no se encuentran todos los Frameworks, ya que su número es muy amplio y va creciendo día a día, en el listado aparecen algunos como ejemplo, siendo los que quedan afuera no menos importantes.

- **React:** Una librería de JavaScript para construir interfaces de usuario.
- **Angular:** Un framework de JavaScript para construir aplicaciones web.
- **Vue.js:** Un framework de JavaScript para construir interfaces de usuario.
- **Express:** Un framework minimalista y flexible de Node.js para aplicaciones web.
- **Laravel:** Un framework de PHP para aplicaciones web con sintaxis elegante.
- **Django:** Un framework de Python de alto nivel para construir aplicaciones web.

Sentencias comunes a todos los lenguajes

Las sentencias de programación: son las instrucciones individuales que componen un programa informático. Cada sentencia le indica al computador qué acción realizar. Hay diferentes tipos de sentencias en la programación, algunos ejemplos son:

Sentencias de asignación: asignan un valor a una variable. Ejemplo: $x = 5$

Sentencias de control de flujo: controlan el orden en que se ejecutan las instrucciones en un programa. Ejemplo:

if, else, for, while, etc.

Sentencias de llamada a funciones: llaman a una función ya definida para ejecutar un conjunto de instrucciones.

Ejemplo: `print(x)`.

Sentencias de declaración: utilizadas para declarar variables, funciones, clases, etc. Ejemplo: `var x;`

Sentencias de salida: utilizadas para mostrar información en la pantalla o en otro dispositivo de salida. Ejemplo: `console.log(x);`

Sentencias de entrada: utilizadas para recibir información del usuario o de otro dispositivo de entrada. Ejemplo: `input()`

Cada lenguaje tiene su propia sintaxis y conjunto de sentencias, pero estas son algunas de las sentencias más comunes utilizadas en la programación.

Lenguajes de programación

Un lenguaje de programación es un conjunto de reglas sintácticas y semánticas que se utilizan para escribir programas informáticos. Un lenguaje de programación de software es un lenguaje formal que se utiliza para escribir programas de computadora. Un programa de computadora es un conjunto de instrucciones que se ejecutan en una computadora para realizar una tarea específica. Los lenguajes de programación de software proporcionan una forma de expresar estas instrucciones de manera clara y precisa, de manera que la computadora pueda entenderlas y ejecutarlas. Existen muchos lenguajes de programación diferentes, cada uno con sus propias características y utilizado para diferentes propósitos. Algunos ejemplos de lenguajes de programación de software populares son C, C++, Java, Python, JavaScript y C#.

Bajo y Alto Nivel

Los lenguajes de programación se clasifican en dos categorías: lenguajes de bajo nivel y lenguajes de alto nivel. Los lenguajes de bajo nivel son aquellos que se asemejan más al lenguaje de la máquina, es decir, al código binario que entiende el procesador. Estos lenguajes son más difíciles de aprender y utilizar, ya que requieren un conocimiento detallado de la arquitectura de la computadora. Ejemplos de lenguajes de bajo nivel son el lenguaje ensamblador y el lenguaje de máquina. Por otro lado, los lenguajes de alto nivel son aquellos que se asemejan más al lenguaje humano. Son más fáciles de aprender y utilizar, ya que están diseñados para ser más intuitivos y legibles. Estos lenguajes son más abstractos y menos dependientes de la arquitectura de la computadora. Ejemplos de lenguajes de alto nivel son C, Java, Python, JavaScript, entre otros. Los lenguajes de alto nivel son más populares y comúnmente utilizados en la programación debido a su facilidad de uso y su capacidad para abstraerse de los detalles técnicos de la computadora. Sin embargo, los lenguajes de bajo nivel aún tienen sus usos en ciertas aplicaciones especializadas, como la programación de sistemas operativos y dispositivos de bajo nivel.

Resumen y relacion entre programación, Frameworks, editores y lenguajes

La programación es el proceso de escribir código para crear programas de computadora. Los editores de código son herramientas utilizadas por los programadores para escribir y editar el código. Los frameworks son conjuntos de herramientas y librerías que proporcionan una estructura para desarrollar programas de manera más eficiente. Los lenguajes de programación son los medios a través de los cuales los programadores expresan las instrucciones para que una computadora realice una tarea específica. La relación entre estos elementos es que los editores de código se utilizan para escribir y editar el código en un lenguaje de programación específico, mientras que los frameworks proporcionan una estructura para el desarrollo del software y pueden ser utilizados con varios lenguajes de programación. Por ejemplo, un programador puede utilizar el editor de código Visual Studio Code para escribir código en C# utilizando el framework de .NET. En resumen, los lenguajes de programación son el

medio a través del cual se expresan las instrucciones para una computadora, los editores de código son las herramientas utilizadas para escribir y editar ese código, los frameworks proporcionan una estructura para desarrollar programas de manera más eficiente y los editores de código ayudan a escribir y editar el código en un lenguaje de programación específico.

Tipos de programación

Programación para consola:

La programación para consola es una forma de programación en la que los programas interactúan con el usuario a través de la línea de comandos de un sistema operativo. Los programas de consola suelen tener una interfaz de texto simple y no requieren una interfaz gráfica de usuario (GUI). Estos programas son adecuados para tareas específicas, automatización y administración de sistemas.

Programación con ventanas:

La programación GUI se refiere a la creación de programas con una interfaz gráfica de usuario (GUI). Estos programas utilizan elementos gráficos como botones, menús, ventanas y cajas de diálogo para interactuar con el usuario. La programación GUI es adecuada para aplicaciones que requieren una interfaz fácil de usar y atractiva para el usuario final.

Programación web:

La programación web se refiere a la creación de aplicaciones y sitios web que se ejecutan en un navegador web. Estas aplicaciones utilizan lenguajes de programación como HTML, CSS y JavaScript para crear contenido y diseño, y lenguajes de servidor como PHP, Python o Ruby para acceder a bases de datos y procesar datos. La programación web es adecuada para aplicaciones que se ejecutan en un entorno distribuido y requieren acceso a datos en tiempo real.

Programación móvil:

La programación móvil se refiere a la creación de aplicaciones para dispositivos móviles como smartphones y tablets. Estas aplicaciones pueden ser nativas, es decir, escritas específicamente para un sistema operativo móvil específico como iOS o Android, o pueden ser aplicaciones web progresivas (PWA) que se ejecutan en un navegador web. Los desarrolladores utilizan lenguajes de programación específicos como Swift o Kotlin para desarrollar aplicaciones nativas y HTML, CSS y JavaScript para desarrollar aplicaciones web progresivas. La programación móvil es adecuada para aplicaciones que se ejecutan en dispositivos móviles y requieren acceso a características del dispositivo como la cámara, el GPS y los sensores.

Sistemas operativos

Windows:

es un sistema operativo desarrollado y comercializado por Microsoft. Es uno de los sistemas operativos más utilizados en computadoras de escritorio y portátiles. Algunos de los fundamentos de Windows incluyen: Interfaz gráfica de usuario: Windows utiliza una interfaz gráfica de usuario (GUI) con elementos como iconos, ventanas y menús para interactuar con el usuario. Compatibilidad con software: Windows es compatible con una amplia variedad de software, incluyendo programas de productividad, juegos y aplicaciones de negocios. Conectividad: Windows ofrece características de conectividad para conectarse a redes y dispositivos externos, como impresoras y discos duros externos. Seguridad: Windows incluye características de seguridad para proteger el sistema y los datos del usuario, como Firewall y Control de cuentas de usuario. Multitarea: Windows permite ejecutar varios programas al mismo tiempo, lo que facilita la multitarea. Personalización: Windows permite personalizar la apariencia y la configuración del sistema operativo, incluyendo el fondo de pantalla, los temas y los iconos. En resumen, Windows es un sistema operativo desarrollado y comercializado por Microsoft, es uno de los más utilizados en computadoras de escritorio y portátiles, utiliza una interfaz gráfica de usuario, es compatible con una amplia variedad de software, ofrece características de conectividad, seguridad, multitarea y permite personalizar la apariencia y configuración del sistema operativo.

Linux:

es un sistema operativo de código abierto, similar a Unix. Fue creado por Linus Torvalds en 1991 mientras estudiaba en la Universidad de Helsinki, Finlandia. A diferencia de sistemas operativos como Windows y macOS, Linux es un sistema operativo que puede ser modificado y distribuido libremente por cualquier persona. Esto ha llevado a la creación de muchas distribuciones diferentes de Linux, como Ubuntu, Fedora y Debian. Linux se ejecuta en una variedad de dispositivos, incluyendo computadoras de escritorio, servidores, teléfonos inteligentes y dispositivos de Internet de las cosas (IoT). Es popular entre los desarrolladores debido a su estabilidad, seguridad y flexibilidad, y también es ampliamente utilizado en la industria y en el sector gubernamental. En resumen Linux es un sistema operativo de código abierto, gratuito, estable, seguro y flexible que se puede utilizar en una variedad de dispositivos, es popular entre desarrolladores y ampliamente utilizado en la industria y el sector gubernamental.

MacOs:

Anteriormente conocido como OS X, es un sistema operativo desarrollado y distribuido por Apple para sus computadoras de escritorio y portátiles Macintosh. Es un sistema operativo basado en Unix que se ejecuta en el hardware de Apple. Incluye características como una interfaz de usuario gráfica de escritorio, integración con otros dispositivos de Apple, así como aplicaciones de productividad y creatividad, como iPhoto, iMovie y GarageBand. Además, macOS también ofrece compatibilidad con software de terceros mediante su tienda en línea, la Mac App Store. Es compatible con procesadores Intel y ARM. Hay varias características adicionales que hacen de macOS un sistema operativo popular entre los usuarios de Mac. Algunas de estas características incluyen: Interfaz de usuario intuitiva y fácil de usar: La interfaz de usuario de macOS es conocida por su diseño limpio y sencillo, lo que hace que sea fácil de navegar para los usuarios. Integración con otros dispositivos de Apple: macOS se integra perfectamente con otros dispositivos de Apple, como iPhone, iPad y Apple Watch, lo que permite a los usuarios sincronizar fácilmente sus datos y configuraciones entre dispositivos.

Android:

Es un sistema operativo móvil de código abierto desarrollado por Google. Fue diseñado principalmente para dispositivos móviles como smartphones y tabletas, pero también se utiliza en televisores inteligentes, automóviles y otros dispositivos electrónicos. Algunas de las características clave de Android incluyen: Interfaz de usuario personalizable: Android permite a los usuarios personalizar la apariencia y la configuración de su dispositivo mediante la instalación de temas y aplicaciones de terceros. Aplicaciones de Google: Android viene con una serie de aplicaciones de Google preinstaladas, como Gmail, Maps y YouTube, lo que permite a los usuarios acceder fácilmente a estos servicios. Tienda de aplicaciones: Android cuenta con una tienda de aplicaciones oficial, Google Play Store, donde los usuarios pueden descargar y comprar aplicaciones de terceros. Integración con Google Assistant: Android se integra con Google Assistant, un asistente de voz inteligente desarrollado por Google, lo que permite a los usuarios interactuar con su dispositivo mediante comandos de voz. Amplia variedad de dispositivos: Android se utiliza en una amplia variedad de dispositivos de diferentes fabricantes, lo que significa que hay una gran variedad de opciones de dispositivos disponibles para los usuarios. Actualizaciones regulares: Google libera actualizaciones regulares para Android, lo que garantiza que el sistema operativo esté siempre actualizado con las últimas características y mejoras de seguridad. Sistema abierto: El sistema es de código abierto, lo que permite a los desarrolladores de terceros modificar y mejorar el sistema operativo. Multitarea: es posible ejecutar varias aplicaciones al mismo tiempo, y se pueden alternar entre ellas fácilmente, permitiendo una mejor productividad en el dispositivo.

Lenguajes de programación mas conocidos

- [Ada](#)

- Assembly
- Assembly Web
- C
- C#
- C++
- COBOL
- CSS
- Dart
- Elixir
- Erlang
- F#
- Fortran
- Go
- HTML
- Haskell
- Java
- JavaScript
- Json
- Kotlin
- Lisp
- Lua
- NodeJS
- Objective-C
- Pascal
- Perl
- PHP
- Python
- R
- Ruby
- Rust
- Scala
- Smalltalk
- SQL
- Swift
- TypeScript
- Visual Basic

Algunos Lenguajes menos usados en la actualidad

- ActionScript
- Bash
- Basic
- CoffeeScript
- Clipper
- Crystal
- D
- Elm
- Emacs Lisp
- Forth
- Groovy
- Julia
- Logo
- Lua

- Nim
- OCaml
- Pascal
- Prolog
- Racket
- Scheme
- Shell
- Standard ML
- Tcl
- VHDL
- XML

Ada

Ada:

es un lenguaje de programación de alto nivel, desarrollado por el Departamento de Defensa de los Estados Unidos (DoD) en los años 80. Fue diseñado para ser utilizado en sistemas críticos, como sistemas de control de vuelo y sistemas militares. Los fundamentos del lenguaje Ada incluyen: Seguridad: Ada fue diseñado para ser seguro y confiable, con características como tipos de datos seguros y control de acceso a los datos. Modularidad: Ada permite dividir el código en módulos y paquetes, lo que facilita la lectura, la depuración y la reutilización del código. Orientado a objetos: Ada es un lenguaje orientado a objetos, lo que significa que permite la creación de objetos y la herencia de características entre ellos. Sistema de tiempo real: Ada tiene un sistema de tiempo real incorporado, lo que permite la programación de tareas en tiempo real, como el control de sistemas de control de vuelo. Interfaz de usuario: Ada tiene una interfaz de usuario incorporada, lo que permite la creación de programas con una interfaz gráfica de usuario (GUI) fácil de usar. Concurrente: Ada tiene un sistema de concurrencia incorporado, lo que permite la creación de programas concurrentes que pueden ejecutar varias tareas al mismo tiempo. En resumen.

Ada es un lenguaje de programación de alto nivel, desarrollado específicamente para sistemas críticos y seguros, con características como seguridad, modularidad, orientación a objetos, sistema de tiempo real, interfaz de usuario y concurrencia. Fue diseñado para ser utilizado en proyectos del Departamento de Defensa de los Estados Unidos, pero también se utiliza en otros campos, como la aviación y la medicina. Su enfoque en la seguridad y la confiabilidad lo convierten en una excelente opción para proyectos críticos y de alto rendimiento.

Assembly (Ensamblador)

Assembly (Ensamblador):

El lenguaje ensamblador es un lenguaje de programación de bajo nivel utilizado para programar en el nivel de la arquitectura del procesador. Los fundamentos del lenguaje ensamblador incluyen: Control de bajo nivel: El lenguaje ensamblador permite acceder y controlar directamente los registros y la memoria del procesador, lo que lo hace muy cercano al lenguaje de máquina. Instrucciones simples: El lenguaje ensamblador se basa en un conjunto reducido de instrucciones simples, como sumar, restar, desplazamiento de bits, etc. Direccionamiento de memoria: El lenguaje ensamblador permite acceder y operar con los diferentes tipos de direccionamiento de memoria, como absoluto, relativo y indirecto. Optimización: El lenguaje ensamblador permite un control fino del rendimiento y una optimización directa del código, lo que lo hace adecuado para aplicaciones de tiempo crítico. Depuración: El lenguaje ensamblador permite una depuración detallada y controlada del código, lo que facilita el diagnóstico de problemas. Portabilidad: El lenguaje ensamblador es específico para cada arquitectura de procesador, por lo que el código escrito para un procesador no es portable para otro. En general, el lenguaje ensamblador es utilizado para escribir programas de sistema básicos, control de dispositivos de bajo nivel y programas que requieren rendimiento y control directo del hardware. En resumen.

El lenguaje ensamblador es un lenguaje de programación de bajo nivel utilizado para programar directamente en el

nivel de la arquitectura del procesador. Se basa en un conjunto reducido de instrucciones simples y permite un control fino del rendimiento y una optimización directa del código. Es específico para cada arquitectura de procesador, por lo que el código escrito para un procesador no es portable para otro. Se utiliza para escribir programas de sistema básicos, control de dispositivos de bajo nivel y programas que requieren rendimiento y control directo del hardware

Web Assembly (Wasm)

WebAssembly (WASM):

es un lenguaje de programación binario de bajo nivel, diseñado para ser ejecutado en el navegador web. Los fundamentos de WebAssembly incluyen: Velocidad: WebAssembly es un lenguaje binario de bajo nivel, lo que lo hace muy rápido y eficiente en la ejecución. Portabilidad: WebAssembly es un formato binario independiente de la plataforma, lo que permite que el código escrito en una plataforma se ejecute en otra plataforma. Interoperabilidad: WebAssembly permite la interoperabilidad con otros lenguajes de programación, como JavaScript, lo que permite el acceso a las funciones y bibliotecas de JavaScript desde WebAssembly. Seguridad: WebAssembly se ejecuta en un ambiente aislado y seguro en el navegador, lo que lo hace seguro para su uso en aplicaciones web críticas. Acceso al sistema: WebAssembly tiene acceso a características del sistema, como el acceso a la memoria y los registros del procesador, lo que permite un mejor control y rendimiento. Compacto: El tamaño del código en WebAssembly es generalmente más pequeño que el código equivalente en otros lenguajes de programación, lo que lo hace adecuado para aplicaciones web que requieren una descarga rápida. En resumen, WebAssembly es un lenguaje de programación binario de bajo nivel, diseñado para ser ejecutado en el navegador web, es muy rápido y eficiente, independiente de la plataforma, seguro, tiene acceso al sistema y es compacto, lo que lo hace adecuado para aplicaciones web críticas y de alto rendimiento.

C:

C:

Es un lenguaje de programación de alto nivel, de uso general, desarrollado por Dennis Ritchie en los años 1972 y 1973 en Bell Labs. Es un lenguaje de programación estructurado, de tipado fuerte y de bajo nivel, lo que permite un gran control sobre el sistema. C es ampliamente utilizado para el desarrollo de sistemas operativos, compiladores, controladores, aplicaciones empresariales y juegos. Además, C es un lenguaje de programación estándar, lo que significa que su sintaxis es reconocida y utilizada en muchas plataformas y sistemas operativos diferentes. Es un lenguaje de programación compilado, lo que significa que el código fuente se traduce a código objeto antes de ser ejecutado. Es un lenguaje de programación de bajo nivel, lo que significa que tiene una gran cercanía con el lenguaje de máquina y permite un gran control sobre el hardware. Es un lenguaje de programación estándar, lo que significa que su sintaxis es reconocida y utilizada en muchas plataformas y sistemas operativos diferentes. Es utilizado para desarrollar muchos sistemas operativos, como UNIX, Linux y Windows. Es utilizado para desarrollar muchos compiladores, como GCC y Clang. Es utilizado para desarrollar muchas aplicaciones empresariales y juegos. Es un lenguaje de programación muy flexible, ya que permite trabajar con diversos tipos de datos y estructuras de control de flujo. Tiene un gran ecosistema de bibliotecas y frameworks que lo hace muy poderoso.

C#

C#:

Es un lenguaje de programación desarrollado por Microsoft en el año 2000. Es un lenguaje de programación orientado a objetos que se utiliza principalmente en el desarrollo de aplicaciones para la plataforma Microsoft .Net. Es similar en sintaxis a C++ y Java, y es utilizado en una variedad de aplicaciones, desde aplicaciones de escritorio

y juegos hasta aplicaciones web y móviles. C# es un lenguaje potente y versátil que permite a los desarrolladores crear aplicaciones de alta calidad de forma rápida y eficiente. C# es un lenguaje de programación moderno y estándar, con características como la gestión automática de memoria, tipos seguros y un sistema de tipos fuertemente tipado. También tiene un gran apoyo de la comunidad de desarrolladores, con una amplia variedad de bibliotecas y frameworks disponibles para ayudar en el desarrollo de aplicaciones. Además, C# es compatible con una variedad de plataformas, como Windows, Linux y MacOS. Por lo tanto, se considera uno de los lenguajes de programación más populares y utilizados en el mundo.

C++

C++:

A mi entender es lenguaje mas importante desde la creacion de las PC. por medio de el, se escribieron la mayoría de los sistemas operativos y demas lenguajes vigentes en la actualidad. Es un lenguaje de programación de alto nivel, desarrollado por Bjarne Stroustrup a principios de los años 80. Utilizado en aplicaciones de sistemas, videojuegos, aplicaciones de escritorio y aplicaciones de alto rendimiento. C++ también cuenta con una gran cantidad de características avanzadas, como la sobrecarga de operadores, la herencia múltiple, la templates y el manejo de excepciones, lo que permite una mayor flexibilidad y expresividad en el código. Además, C++ es un lenguaje estandarizado, lo que significa que sigue un conjunto establecido de reglas y convenciones, lo que facilita la portabilidad del código a diferentes plataformas. Es un lenguaje de programación de alto rendimiento, lo que significa que es adecuado para desarrollar aplicaciones de alta velocidad y computación intensiva. También es un lenguaje de programación de bajo nivel, lo que significa que permite el acceso directo a los recursos del sistema, como la memoria y los registros. Es también un lenguaje de programación orientado a objetos, lo que permite la creación de programas modularizados y fácilmente extensibles. Además, C++ es un lenguaje estandarizado, lo que significa que se rige por un estándar internacional, lo que garantiza la portabilidad y la compatibilidad entre diferentes sistemas y plataformas.

Lisp

Lisp:

Es un lenguaje de programación de alto nivel, de tipo interpretado y dinámico, creado en 1958 por John McCarthy. Es uno de los lenguajes de programación más antiguos que sigue siendo ampliamente utilizado hoy en día. Lisp se caracteriza por su sintaxis basada en listas, lo que permite una gran flexibilidad en la manipulación de estructuras de datos. Además, Lisp es conocido por su capacidad para manipular fácilmente metaprogramación, lo que permite a los programadores extender fácilmente el lenguaje. Lisp ha sido utilizado en una amplia variedad de campos, desde inteligencia artificial hasta la programación de videojuegos.

Cobol

COBOL (Common Business-Oriented Language)

Es un lenguaje de programación de alto nivel diseñado específicamente para el uso en aplicaciones comerciales y administrativas. Fue desarrollado en el año 1959 y se ha utilizado ampliamente en aplicaciones de gran envergadura en el sector financiero, gubernamental y de negocios. Una de las características más notables de COBOL es su enfoque en el uso de un lenguaje legible y fácil de entender para los humanos, con la intención de hacer que los programas sean fácilmente comprensibles y modificables por los usuarios no técnicos. COBOL también tiene un fuerte enfoque en la gestión de archivos, lo que lo convierte en una excelente opción para aplicaciones que manejan grandes cantidades de datos. Sin embargo, debido a su enfoque en la legibilidad humana, COBOL tiende a ser menos eficiente que otros lenguajes en términos de rendimiento y se ha quedado atrás en comparación con otros lenguajes en términos de funcionalidad y características modernas. COBOL se sigue

usando en algunas aplicaciones legadas y aun se utiliza en algunos proyectos nuevos, especialmente en el sector financiero.

CSS

CSS (Cascading Style Sheets):

Basicamente CSS sirve para darle orden,color y formato a los archivos html, que son los hacen que veamos las pagina web tal como son. Es un lenguaje de diseño utilizado para describir la presentación de documentos escritos en un lenguaje de marcas como HTML o XML. Es utilizado principalmente para aplicar estilos visuales, como colores, tamaños de fuente, márgenes y posicionamiento, a los elementos de una página web. Con CSS, los desarrolladores pueden separar la estructura de contenido de una página web de su diseño y presentación visual, lo que permite un mayor control y flexibilidad en la creación de páginas web. CSS tiene una variedad de características, incluyendo selectores, reglas de estilo y cascada. Los selectores permiten seleccionar elementos específicos de una página web para aplicar estilos. Las reglas de estilo especifican cómo deben verse los elementos seleccionados. La cascada permite a los desarrolladores especificar reglas de estilo para varios elementos y determinar cómo se aplican y se combinan estas reglas. CSS también tiene varias extensiones y versiones, como CSS2, CSS3 y CSS4, cada una con características y funciones adicionales. CSS ha evolucionado significativamente desde su creación en 1996 y sigue siendo una herramienta esencial para la creación de páginas web modernas y atractivas.

Dart

Dart:

Es un lenguaje de programación de código abierto desarrollado por Google. Fue lanzado en 2011 y es utilizado para desarrollar aplicaciones para dispositivos móviles, web y escritorio. Dart es similar a otros lenguajes de programación como Java y C#, y tiene una sintaxis fácil de aprender para desarrolladores que ya tienen experiencia en otros lenguajes de programación. Dart también incluye características modernas como tipado estático, programación orientada a objetos y una biblioteca estándar completa. Es utilizado para desarrollar aplicaciones para la plataforma Flutter de Google, que se utiliza para desarrollar aplicaciones para dispositivos móviles, web y desktop.

Elixir

Elixir:

Es un lenguaje de programación de código abierto basado en Erlang. Fue desarrollado para mejorar la productividad y la escalabilidad en aplicaciones distribuidas y concurrentes. Es similar a otros lenguajes funcionales como Haskell y Lisp, y tiene un enfoque en la programación funcional, la programación concurrente y la programación orientada a mensajes. Elixir utiliza el modelo de actores de Erlang para manejar la concurrencia, lo que permite a las aplicaciones manejar un gran número de conexiones simultáneas de forma eficiente. También tiene un sistema de macro, que permite a los desarrolladores crear nuevas funciones y estructuras de datos personalizadas. Elixir es utilizado para desarrollar aplicaciones web escalables y de alta disponibilidad y soporta tanto la programación síncrona como asíncrona. Además, es utilizado en aplicaciones IoT, aplicaciones de juegos y aplicaciones de redes sociales. En resumen, Elixir es un lenguaje de programación moderno y escalable, diseñado para resolver problemas de concurrencia y alta disponibilidad en aplicaciones distribuidas.

Erlang

Erlang:

Es un lenguaje de programación de código abierto desarrollado por Ericsson en 1986. Es un lenguaje funcional concurrente diseñado para desarrollar aplicaciones distribuidas y altamente disponibles. Erlang tiene un enfoque en la programación concurrente y la programación orientada a mensajes, utilizando el modelo de actores para manejar la concurrencia. El sistema de mensajería de Erlang permite a las aplicaciones automáticamente de errores y continuar funcionando, lo que aumenta la confiabilidad de las aplicaciones. Erlang es utilizado en sistemas telefónicos, sistemas de mensajería, sistemas de redes sociales y aplicaciones web escalables. También es utilizado en aplicaciones de IoT y en el desarrollo de juegos. En resumen, Erlang es un lenguaje de programación concurrente y distribuido, diseñado para desarrollar aplicaciones altamente disponibles y escalables. Su sistema de mensajería y control de fallos lo hacen ideal para aplicaciones en sistemas telefónicos, redes sociales y aplicaciones web escalables.

F#

F# :

Es un lenguaje de programación funcional y multiparadigma desarrollado por Microsoft. Es similar a otros lenguajes funcionales como Haskell y OCaml, y tiene características de lenguajes de programación estructurada y orientada a objetos. Algunas de las características de F# incluyen:

Uso de funciones puras y evitando el estado mutable

Sintaxis similar a ML

Integración con el ecosistema de .NET

Soporte para programación concurrente y paralela

Interoperabilidad con C# y otros lenguajes de .NET

F# es utilizado en aplicaciones empresariales, ciencia de datos, inteligencia artificial y visualización de datos.

Fortran

Fortran:

Es un lenguaje de programación de alto rendimiento desarrollado originalmente por IBM en los años 50. Es uno de los lenguajes de programación más antiguos que aún se utilizan en la actualidad. Algunas de las características de Fortran incluyen:

Enfoque en cálculos numéricos y científicos.

Sintaxis similar a la matemática.

Uso de arreglos y matrices de forma nativa.

Fortran 77 es el estándar antiguo y Fortran 90/95 incluyen características modernas como OOP, generics, etc

Fortran es especialmente utilizado en el ámbito científico y de ingeniería, como en la simulación y análisis numérico, y en aplicaciones de alto rendimiento, como la física de partículas y la meteorología. Es importante mencionar que existen varias versiones de Fortran, Fortran 77, 90, 95, 2003, 2008 y 2018, y cada una tiene características diferentes.

Go

Go:

Es un lenguaje de programación desarrollado por Google en 2007. Es un lenguaje de programación multiparadigma, con un enfoque en la simplicidad, la eficiencia y la escalabilidad. Algunas de las **características de Go incluyen:**

Sintaxis simple y fácil de aprender, Go es realmente sencillo, lo que lo hace ideal para principiantes y desarrolladores que buscan una alternativa a lenguajes más complejos.

Concurrencia incorporada:

Go tiene soporte para concurrencia incorporado a través de "goroutines", lo que permite a los desarrolladores crear programas concurrentes y escalables de forma fácil.

Garbage collection:

Go tiene un recolector de basura incorporado, lo que significa que los desarrolladores no tienen que preocuparse por liberar la memoria manualmente.

Compilación rápida:

Go se compila rápidamente, lo que permite a los desarrolladores probar y depurar sus programas de forma eficiente.

Interoperabilidad:

Go tiene una interfaz C, lo que permite a los desarrolladores llamar código C desde Go y viceversa.

Go es utilizado en sistemas distribuidos, aplicaciones web, aplicaciones de redes, sistemas de monitoreo y en general en aplicaciones donde se requiera alto rendimiento y escalabilidad.

En resumen, Go es un lenguaje de programación moderno y poderoso, diseñado para facilitar el desarrollo de programas concurrentes y escalables. Su enfoque en la simplicidad y la eficiencia lo convierten en una excelente opción para una variedad de proyectos, especialmente en el ámbito de la tecnología.
