

Características lenguajes



ÍNDICE

A) Java

B) Python

C) C/C++

D) Javascript

E) PHP

F) **PRINCIPALES DIFERENCIAS ENTRE JAVA Y C++**

g) Scrum y agile

A) Java

Java es un lenguaje de programación de nivel alto. Hay varias aplicaciones y páginas web que requieren tener instalado JAVA.

JAVA puede utilizarse para desarrollar diferentes tipos de aplicaciones como aplicaciones web, miniaplicaciones o programas que

se ejecutan en tu ordenador. También es preferido para desarrollar aplicaciones científicas y también herramientas de desarrollo web.

Es requerido conocer el lenguaje de JAVA para crear aplicaciones para ANDROID.

Java es un lenguaje compilado. Java se caracteriza por ser un lenguaje muy estricto y aunque pueda parecer tedioso, esto nos llevará menos

quebraderos de cabeza al analizar el código.

En cuanto a la administración de memoria, en Java está controlado por el sistema y no utiliza punteros.

Java no soporta herencia múltiple.

En Java se utiliza el compilador y el intérprete.

B) Python

PYTHON es un lenguaje que se ha ido extendiendo en uso, gran parte se debe al gran avance del campo de la "Inteligencia artificial".

Python se utiliza para las páginas web también, de hecho se utiliza un framework llamado DJANGO que está basado en código de PYTHON.

Como modo de ejemplo, NETFLIX utiliza Python para el análisis de datos en la parte del servidor.

Python es un lenguaje de nivel alto.

Simple:

Python es un lenguaje muy simple, por lo que es muy fácil iniciarse en este lenguaje.

El pseudo-código natural de Python es una de sus grandes fortalezas.

Propósito General:

Usando el lenguaje Python se puede crear todo tipo de programas; programas de propósito general y también se pueden desarrollar páginas Web.

Open Source:

Debido a la naturaleza de Python de ser Open Source; ha sido modificado para que pueda funcionar en diversas plataformas (Linux, Windows, Macintosh, Solaris, OS/2, Amiga, AROS, AS/400, BeOS, OS/390, z/OS, Palm OS, QNX, VMS, Psion, Acorn RISC OS, VxWorks, PlayStation, Sharp Zaurus, Windows CE y PocketPC).

Al ser Open Source es gratuito.

Lenguaje Orientado a Objetos:

Al ser un Lenguaje Orientado a Objetos es construido sobre objetos que combinan datos y funcionalidades.

Lenguaje de Alto Nivel:

Al programar en Python no nos debemos preocupar por detalles de bajo nivel, (como manejar la memoria empleada por el programa).

Incrustable:

Se puede insertar lenguaje Python dentro un programa C/C++ y de esta manera ofrecer las facilidades del scripting.

Extensas Librerías:

Python contiene una gran cantidad de librerías, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas comunes sin necesidad de tener que programarlas desde cero.

Las librerías pueden ayudar a hacer varias cosas como expresiones regulares, generación de documentos, evaluación de unidades, pruebas, procesos, bases de datos, navegadores web, CGI, ftp, correo electrónico, XML, XML-RPC, HTML, archivos WAV, criptografía, GUI, y también otras funciones dependientes del Sistema.

Sintaxis clara:

Python tiene una sintaxis muy visual, gracias a que maneja una sintaxis indentada (con márgenes), que es de carácter obligatorio. Para separar los bloques de código en Python se debe tabular hacia dentro. Esto ayuda a que todos los programadores adopten las mismas notaciones y que los programas hechos en Python tengan un aspecto muy similar.

C) C/C++

A día de hoy C es utilizado preferentemente como un lenguaje para escribir lenguaje de bajo nivel como kernels de sistemas operativos, Drivers, etc.

Esto se debe en parte a su apoyo universal por parte de todos los proveedores de hardware.

C++ originalmente se creó para mejorar y extender C++. Hoy en día muchos desarrolladores de aplicaciones y desarrolladores de videojuegos

aún utilizan este lenguaje. Google, Facebook y otras compañías aún la utilizan para su infraestructura central. A modo de ejemplo los antivirus

se desarrollan en base a C/C++.

En C++ la administración de la memoria es llevada a cabo por los desarrolladores usando punteros.

C++ soporta herencia múltiple y herencia singular. En c++ se utiliza solo el compilador.

D) Javascript

Javascript es un lenguaje de entorno cliente, Javascript se utiliza generalmente y de manera primaria para mejorar la experiencia del usuario

en cuanto a la interacción con la página web. De manera secundaria Javascript se utiliza en el lado del servidor con Node,

en el lado del escritorio con Electron y en el móvil con React Native.

En resumidas cuentas Javascript es utilizado principalmente y casi exclusivamente como un lenguaje de desarrollo front-end.

Javascript es un lenguaje de script. Javascript es un lenguaje muy flexible y fácil de usar.

E) PHP

PHP es un lenguaje utilizado como servidor web. Un servidor recibe una solicitud HTTP y procesa la solicitud con PHP.

En este punto PHP analiza la solicitud y rellena la solicitud.

Un ejemplo común sería el inicio de sesión en Facebook, PHP se utiliza para validar la información que has introducido. La mayor especialidad de

PHP es como desarrollo web. Podemos crear páginas web estáticas o dinámicas usando PHP. PHP es open-source y es gratis.

Características

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, ya que en su desarrollo se simplificaron distintas especificaciones, como es el caso de la definición de las variables primitivas, ejemplo que se hace evidente en el uso de php arrays.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos (llamados extensiones).
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.

- Si bien PHP no obliga a quien lo usa a seguir una determinada metodología a la hora de programar, aun haciéndolo, el programador puede aplicar en su trabajo cualquier técnica de programación o de desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.
- Debido a su flexibilidad ha tenido una gran acogida como lenguaje base para las aplicaciones WEB de manejo de contenido, y es su uso principal.

F) PRINCIPALES DIFERENCIAS ENTRE JAVA Y C++

- Memoria libre no utilizada
 - Java tiene *recolección de basura automática* como un hilo con nombre que es responsable de eliminar los objetos no utilizados de la memoria y que JVM genera automáticamente. Por lo tanto, la memoria se libera automáticamente cuando ya no se usa.
 - C ++ no tiene *recolección automática de basura*. Por lo tanto, toda la memoria reservada no utilizada debe liberarse manualmente cuando ya no se usa para evitar pérdidas de memoria.
- Punteros
 - C ++ admite punteros. Puede declarar explícitamente los punteros que contienen la dirección de otra variable y puede pasar la dirección de la variable.
 - Java no admite punteros, lo que significa que en Java no puede declarar explícitamente los punteros como puede hacerlo en C ++.
- Independencia de plataforma que da portabilidad
 - C ++ es un *lenguaje dependiente de la plataforma*. Por lo tanto, los programas C ++ que se compilan en una máquina solo pueden ejecutarse en esa máquina en particular en la que se compilan, por lo que no es portátil.
 - Mientras que Java es *lenguaje independiente de la plataforma*. Cuando los programas de Java se compilan, se convierten en algún código intermedio denominado *bytecode*. Ahora, cualquier máquina que tenga instalada la aplicación JVM puede ejecutar el archivo fuente ya compilado sin preocuparse por el sistema operativo para que sea portátil.
- Seguridad
 - Dado que el programa C ++ permite el uso de punteros, permite controlar el hardware mediante punteros y tampoco necesita ninguna aplicación para ejecutarse. Entonces, podemos desarrollar virus usando C ++ que no es seguro para la computadora.
 - Java no admite punteros y los programas de Java necesitan que se ejecute la aplicación JVM. Entonces, el lenguaje Java previene este tipo de código inseguro.
- Herencia múltiple

- C ++ soporta la herencia múltiple, ya que tiene *el operador de resolución de ámbito (:)* mediante el cual podemos superar "la ambigüedad" sucediendo durante la herencia múltiple.
- Java no admite la herencia múltiple porque un constructor de subclase solo puede llamar a un constructor de superclase y solo podemos tener un *super ()*; declaración en un cuerpo de constructor que conduce a una ambigüedad.
- Con respecto al uso
 - Java es más probable que se encuentre en aplicaciones Web y Enterprise, C ++ es más probable que se encuentre en System y otras aplicaciones de bajo nivel.
- Valor verdadero / falso
 - C ++ considera todos los valores distintos de cero como VERDADERO y 0 como FALSO.
 - A diferencia de C ++, una expresión de valor entero no se puede colocar donde se espera un BOOLEAN en Java.
- Función virtual
 - A diferencia de C ++, las funciones son virtuales por defecto en Java. Por lo tanto, el polimorfismo en tiempo de ejecución está presente de forma predeterminada en Java.
- Método derivado de la clase base
 - C ++ permite dar un especificador de nivel de acceso más restrictivo a los métodos derivados de superclase / clase base.
 - Java no permite dar un especificador de nivel de acceso más restrictivo a los métodos derivados de la superclase / clase base.
- Clase abstracta
 - En C ++, si una clase tiene al menos una función virtual pura, entonces la clase se vuelve abstracta.
 - A diferencia de C ++, en Java, se utiliza un *resumen de palabras clave* por separado para hacer un resumen de clase.
- Excepciones
 - En C ++, todas las excepciones están desmarcadas, por lo que el compilador no lo obliga a manejar o especificar la excepción. Depende de los programadores ser civilizados y especificar o capturar las excepciones.
 - En las excepciones de Java en las clases Error y RuntimeException son excepciones no marcadas, se verifica todo lo demás bajo arrojable.

SCRUM

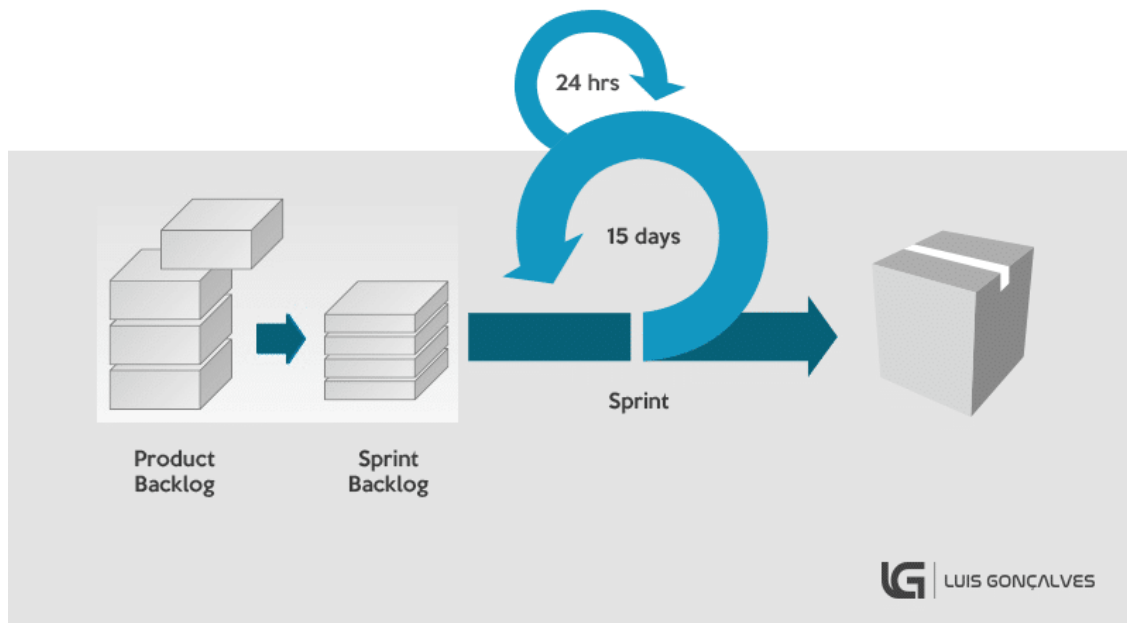
Scrum es un proceso en el cuál se deben aplicar de manera regular buenas prácticas para trabajar colaborativamente y obtener el mejor resultado posible. Se basa en equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final. Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable.

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

Scrum Process - Overview



En Scrum el sprint es un espacio de tiempo de 1 mes o menos en el que un potencial lanzamiento es creado. Los Sprints tienen una duración continua, ya que un sprint comienza inmediatamente

después de la finalización del previo sprint. Cada sprint deberá considerarse como con un tiempo inferior a 1 mes.

El Scrum diario es un tiempo diario de 15 minutos dedicados a planear el trabajo de desarrollo del día siguiente durante un sprint.

El sprint backlog es una vista general del trabajo de desarrollo para realizar el objetivo del sprint, y el trabajo necesario para cumplir con la meta. Se encarga el equipo de desarrollo.

AGILE

Agile es un enfoque para el desarrollo de productos de software.

Agile nos proporciona flexibilidad, transparencia y un sistema rápido de entrega de trabajo.

Los principios generales de AGILE son los siguientes:

- ◆ Satisfacer los deseos del cliente y desarrollar continuamente un software.
- ◆ Los cambios son bienvenidos, esto es una ventaja para el cliente indeciso y avaricioso.
- ◆ Concentración en un desarrollo de software continuo, se espera lanzar el proyecto en un espacio corto de tiempo.
- ◆ Los desarrolladores y los hombres del sector de negocio deben trabajar conjuntos durante la duración entera del proyecto.
- ◆ La comunicación cara a cara es la mejor manera de transmitir información a y para un equipo.
- ◆ Equipos autoorganizados suelen ser los mejores.

Agile nació en el año 2001, se reunieron los CEOs de las principales empresas de 'software' en Utah. Allí pusieron en común las mejores prácticas de cada compañía y crearon el "Manifiesto agile";

Antes de 'agile', cuando una empresa quería desarrollar un proyecto nuevo, se ponía en marcha un proceso lineal (en cascada o 'waterfall') que podía tardar uno o dos años en entregar un producto, con un alto riesgo de no adaptarse a la demanda final del cliente. Cuando esto sucedía, se optaba por soluciones de urgencia, o incluso empezar de cero el proyecto.

Otro de los rasgos más característicos de las metodologías ágiles es el de emplear equipos multidisciplinares que trabajen juntos, codo con codo, durante todo el proceso. Así, por ejemplo, el responsable del 'marketing' puede dar 'feedback' de cómo está resultando el producto final al técnico que está realizando una tarea concreta. De esta manera, y junto con las entregas más rápidas, tempranas y frecuentes, el producto resultante es exactamente lo que el mercado está demandando.

Los equipos ágiles realizan sesiones diarias en las que cada miembro tiene que explicar tres cosas: qué tareas ha completado, cuáles va a hacer y, si los hay, señalar los impedimentos que no le han permitido avanzar. Estas reuniones llamadas 'dailys' sirven para que todos los integrantes del equipo sepan en qué punto de trabajo está cada uno y sean conscientes de cómo pueden colaborar para que el proyecto siga avanzando.

Aquí se encuentran los 12 principios del manifiesto Agile:

1. **Satisfacción del cliente.** Es la base de todo. Se alcanza a través de la entrega de productos de valor que cubran una necesidad.
2. **Bienvenidos los nuevos requisitos.** Cambiar sobre la marcha no es dar un paso atrás. Cualquier sugerencia o solución es bienvenida si se trata de mejorar el producto.
3. **Entregas por semanas.** La división del trabajo en fases productivas es la base de la metodología. En lo posible, ejecutar una cada semana.
4. **Es posible medir el progreso.** La evolución de los procesos no es un elemento subjetivo. Se puede medir con indicadores concretos.
5. **Desarrollo sostenible.** La forma de ejecutar los proyectos debe garantizar en sí misma su continuidad. No es una cuestión de hacer por hacer.
6. **Trabajo cercano.** Los líderes de los proyectos deben ejercer su labor en el mismo terreno donde tienen lugar las tareas y no desde los despachos.
7. **Conversación cara a cara.** El gestor responsable debe comunicar de forma eficaz sus mensajes, mejor si se hace de forma presencial. Se recomiendan reuniones periódicas tanto con el cliente como con sus colaboradores.
8. **Motivación y confianza.** Los procesos sólo tendrán éxito si quienes los llevan a cabo son personas motivadas y que interactúan en climas de confianza y solidaridad.
9. **Excelencia técnica y buen diseño.** Las formas nunca deben perderse, así como tampoco la calidad del trabajo. Todo es un conjunto.
10. **Simplicidad.** Las tareas han de ser lo más sencillas posible. Si alguna no puede ser ejecutada en esos términos, debe ser dividida en iteraciones hasta que se reduzca su nivel de complejidad.
11. **Autogestión de los equipos.** Si bien debe existir una figura que monitorice los equipos de trabajo, éstos deben ser capaces de organizarse por sí mismos. El exceso de jerarquías crea dependencia entre los colaboradores.
12. **Adaptación circunstancias cambiantes.** Los proyectos no suelen terminar de la misma forma en que empezaron. Es indispensable que quienes los ejecutan puedan adaptarse a las distintas circunstancias que puedan surgir.