



Escuela:

Facultad de Informática

Carrera:

Licenciatura en informática

Nombre del Alumno:

Antonio Gastelum Landeros

Grupo: 2 - 1

INDICE

Java.....	3
Net Core.....	5
PHP.....	6
Python.....	7
Node JS.....	8

Java

¿Qué es la tecnología Java y por qué la necesito?

Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean muchos servicios y aplicaciones. Los nuevos e innovadores productos y servicios digitales diseñados para el futuro también siguen basándose en Java.

Ventajas de Java

Hasta 2016 (año en el que Kotlin tuvo su primera versión oficial estable) Java era insustituible en el desarrollo Android y esto ocurría por una razón muy sencilla: Java era el único lenguaje de programación oficial para desarrollo Android. A pesar de la alta competencia con otras tecnologías, Java no se queda atrás y todavía tiene una serie de pros que ofrecer que pueden persuadir a uno para hacer la elección a su favor.

- 1.-Es un lenguaje de alto nivel. Esta noción implica que un lenguaje de programación debe parecerse mucho más a un lenguaje humano que a uno de máquina. En consecuencia, debe ser fácil y sencillo de escribir, leer y mantener.
- 2.-Su estabilidad. Se dice que las soluciones creadas con la ayuda de Java son estables. Esto ocurre en parte porque cada día se publica una nueva versión de Java con nuevas funciones avanzadas.
- 3.-Está orientado a objetos. Dado que Java pertenece a la programación orientada a objetos, permite a un desarrollador escribir programas típicos y reutilizar el código. Así, se pueden declarar clases, generar objetos dentro de las clases, trabajar y mantener la interacción entre dos objetos.
- 4.-Su mantenimiento es bastante barato. La naturaleza del trabajo de un programa Java no depende de una infraestructura de hardware única, por lo que es posible ejecutar el servidor en cualquier máquina. Resultado: su mantenimiento es barato.

- 5.-Su seguridad. Java es la primera tecnología que proporciona seguridad como parte integral del diseño. JVM posee un identificador especial que detecta el bytecode y lo comprueba antes de ejecutarlo.
- 6.-Es multihilo. Internamente un programa Java puede realizar varias tareas a la vez.
- 7.-Informática distribuida. Es un método en el que varios ordenadores trabajan juntos en una red. Esto es sin duda una ventaja, ya que permite desarrollar aplicaciones en redes que pueden contribuir tanto a la funcionalidad de la aplicación como a la de los datos.
- 8.-Es portable (independiente de la plataforma). La portabilidad implica que un desarrollador sólo tiene que escribir un código una vez y el programa puede iniciarse en cualquier plataforma. La única condición es que esta plataforma soporte JVM.
- 9.-Es robusto. Se dice que Java es un lenguaje de lo más fiable y potente. Sus compiladores son capaces de identificar cualquier tipo de error en el código. Además, Java tiene grandes características como el manejo de excepciones y la recolección de basura, que también demuestran que Java es fiable.

Inconvenientes de Java

Por desgracia, como cualquier tecnología, Java no sólo tiene ventajas, sino también un montón de inconvenientes. Los varios contras significativos de Java son:

- 1.-El rendimiento de Java. Si comparamos los programas Java con los escritos en C o C++ que se compilan de forma nativa, se nota fácilmente que son mucho más lentos.
- 2.-Su memoria. En Java, la memoria se gestiona a través de la recolección de basura, por lo que cada vez que el recolector de basura trabaja, se deteriora la forma en que la aplicación funciona. La razón principal es que el recolector de basura sólo funciona cuando todos los demás hilos no están trabajando.
- 3.-Costes de la licencia comercial. Desde 2019 Oracle espera que los usuarios paguen por Java Standard Edition 8 cuando se utiliza para fines como negocios, comercio y producción. Así que cuando necesites actualizaciones y corrección de errores, te cobrarán por el número de usuarios que tengas o por el procesador. Duele mucho cuando empiezas a contar los nuevos costes.
- 4.-El aspecto no es nativo (de escritorio). Los programadores utilizan una variedad de herramientas especiales para cada lenguaje con el fin de construir la interfaz gráfica de usuario, GUI para abreviar. Por supuesto, no debemos

olvidarnos de Android Studio, que se diseñó inicialmente para el desarrollo móvil de Android, de modo que las aplicaciones se sienten y se ven nativas. Por lo tanto, en cuanto la conversación pasa a la interfaz de usuario de los programas de escritorio, Java se queda ciertamente rezagado y tendrás que investigar a fondo para encontrar una herramienta que te sirva para crear tu interfaz gráfica de usuario.

5.-La verbosidad de Java hace que el código sea bastante complicado. Java implica que uno necesita usar muchas palabras usadas ya que se parece mucho al lenguaje natural de los humanos. Los desarrolladores escriben casi literalmente sus comandos y pensamientos por lo que el código es drásticamente enorme (especialmente cuando se compara con Python).

NET CORE

¿Qué es .Net Core?

.NET Core es la plataforma de desarrollo de Microsoft más moderna, de código fuente abierto, multiplataforma y de alto rendimiento para la creación de todo tipo de aplicaciones.

Fue liberada en el año 2016, y es el resultado de múltiples esfuerzos para hacer más ágil el desarrollo en .NET puesto que el .NET Framework, depende totalmente del sistema operativo Windows.

.NET Core, a diferencia del .NET Framework, no tiene este tipo de dependencia del sistema Windows y es modular, usando el sistema de paquetes NuGet, gracias al cual, recibiremos las diversas actualizaciones de .NET Core, a diferencia de .NET Framework que se actualiza a través de Windows Update.

Características de .NET Core

Las principales características de .NET Core son las siguientes:

- Es multiplataforma y viene con soporte para su uso con contenedores Docker.
- Alto rendimiento. Se ha desarrollado desde cero y se le ha dado una alta importancia a esta característica.
- Asincronía con el uso de async/await. Se ha implementado este patrón en todas las librerías comunes para mejorar el rendimiento en las llamadas I/O
- Es Open Source.

PHP

¿Qué es PHP?

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies.

Existen principalmente tres campos principales donde se usan scripts de PHP.

- 1.- Scripts del lado del servidor. Este es el campo más tradicional y el foco principal.
- 2.- Scripts desde la línea de comandos. Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera.
- 3.- Escribir aplicaciones de escritorio. Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas.

¿Cuáles son los beneficios de usar PHP?

- 1.-Evoluciona a un ritmo más que satisfactorio en comparación con otras tecnologías.
- 2.-Se caracteriza por ser fácil, potente y profesional.
- 3.-Tiene excelentes frameworks para desarrollar en la web, como Laravel y Symfony.
- 4.-Sigue prácticas de desarrollo ejemplares.
- 5.-Posee una comunidad enorme y muy activa.
- 6.-Cuenta con una variedad de herramientas para los desarrolladores.
- 7.-Ha mejorado la performance notablemente desde su versión 7.
- 8.-Soporta una cantidad infinita de bases de datos y de peticiones, es muy rápido.
- 9.-Permite, al ser de código abierto, que cualquiera pueda aplicar modificaciones para mejorarlo.

PYTHON

¿Qué es Python?

Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML). Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes. El software Python se puede descargar gratis, se integra bien a todos los tipos de sistemas y aumenta la velocidad del desarrollo.

¿Qué beneficios ofrece Python?

Los beneficios de Python incluyen los siguientes:

- Los desarrolladores pueden leer y comprender fácilmente los programas de Python debido a su sintaxis básica similar a la del inglés.
- Python permite que los desarrolladores sean más productivos, ya que pueden escribir un programa de Python con menos líneas de código en comparación con muchos otros lenguajes.
- Python cuenta con una gran biblioteca estándar que contiene códigos reutilizables para casi cualquier tarea. De esta manera, los desarrolladores no tienen que escribir el código desde cero.
- Los desarrolladores pueden utilizar Python fácilmente con otros lenguajes de programación conocidos, como Java, C y C++.
- La comunidad activa de Python incluye millones de desarrolladores alrededor del mundo que prestan su apoyo. Si se presenta un problema, puede obtener soporte rápido de la comunidad.
- Hay muchos recursos útiles disponibles en Internet si desea aprender Python.
- Por ejemplo, puede encontrar con facilidad videos, tutoriales, documentación y guías para desarrolladores.
- Python se puede trasladar a través de diferentes sistemas operativos de computadora, como Windows, macOS, Linux y Unix.

¿Cómo se utiliza Python?

El lenguaje Python se aplica a varios casos de uso en el desarrollo de aplicaciones, incluidos los ejemplos siguientes:

- Desarrollo web del lado del servidor
- Automatización con scripts de Python
- Realizar tareas de ciencia de datos y machine learning
- Desarrollo de software
- Automatización de pruebas de software

NODE JS

Concebido como un entorno de ejecución JavaScript asíncrono basado en eventos, Node.js está diseñado para construir aplicaciones de red escalables. En el siguiente ejemplo de 'hola mundo', se pueden manejar muchas conexiones de manera concurrente. Por cada conexión el callback sera ejecutado, sin embargo, si no hay trabajo que hacer, Node.js entrará en reposo.

Ventajas de Node js

- Eficiencia y escalabilidad: Utiliza un modelo de operaciones de entrada/salida sin bloqueo y basado en eventos, lo que permite manejar una alta concurrencia y un gran número de solicitudes simultáneas de manera eficiente. Esto lo hace ideal para aplicaciones web en tiempo real y de alto rendimiento.
- Unificación del lenguaje: Tanto en el servidor como en el cliente (con herramientas como React o Angular), se puede escribir todo el código en JavaScript. Esto facilita la comunicación y el intercambio de código entre el frontend y el backend, lo que puede mejorar la productividad del equipo de desarrollo.
- Amplio ecosistema de módulos: Cuenta con un gran repositorio de paquetes y módulos disponibles a través de npm. Esto facilita la reutilización de código y acelera el desarrollo de aplicaciones, ya que no es necesario crear todo desde cero.

- Comunidad activa: Hay una gran cantidad de recursos, documentación y soporte disponibles en línea. Además, hay una amplia variedad de herramientas y frameworks contruidos sobre Node.js que pueden facilitar el desarrollo de aplicaciones web.

Inconvenientes de Node js

- Uso intensivo de CPU: Puede no ser la mejor opción si tu aplicación realiza operaciones intensivas en el uso de la CPU, como procesamiento de imágenes o cálculos complejos. Dado que Node.js utiliza un solo hilo de ejecución, estas operaciones pueden bloquear el rendimiento general de la aplicación.
- Madurez de algunas bibliotecas: Aunque tiene un ecosistema de módulos y paquetes muy amplio, algunas bibliotecas pueden ser menos maduras o estables en comparación con otros lenguajes o frameworks más establecidos. Es importante investigar y evaluar las bibliotecas antes de utilizarlas en un proyecto.
- Curva de aprendizaje: Si el equipo de desarrollo no está familiarizado con JavaScript o con el paradigma de programación asíncrona, puede haber una curva de aprendizaje para comprender y aprovechar completamente las características de Nodejs.