Actividad: Introducción, Origen y Evolución de los Lenguajes de Programación

Octubre 21 de 2024

Debes subir la solución a este taller a tu Github, creando una carpeta que se llame

Introducción a las bases de datos y si no tienes Github, envíalo al correo

docente4@pio.edu.co con el asunto Conceptos básicos de bases de datos y tu nombre

completo

Hora de entrega: 10:00 p.m

**Objetivo:** Explorar la historia de los lenguajes de programación, entender su evolución y discutir las preferencias de los estudiantes en cuanto a los lenguajes y plataformas de desarrollo.

**Instrucciones:**

**1. Lee la siguiente introducción:** Los lenguajes de programación han evolucionado enormemente desde sus inicios. Desde los primeros lenguajes de bajo nivel como el ensamblador hasta los lenguajes de alto nivel actuales, cada uno ha sido diseñado para resolver diferentes tipos de problemas y facilitar el trabajo de los desarrolladores.

**Algunos hitos importantes incluyen la creación de lenguajes como:**

* FORTRAN (1957): uno de los primeros lenguajes de alto nivel, diseñado para cálculos científicos.
* COBOL (1959): orientado a la gestión de negocios.
* C (1972): un lenguaje estructurado que influyó en muchos lenguajes modernos.
* Python (1991): popular por su sencillez y versatilidad.
* JavaScript (1995): clave para el desarrollo web.

**Actualmente, el desarrollo de software se puede dividir en dos grandes áreas:**

* Desarrollo móvil: centrado en aplicaciones para dispositivos móviles como teléfonos y tabletas.
* Desarrollo de escritorio: aplicaciones diseñadas para sistemas operativos como Windows, macOS o Linux.

**2. Preguntas de reflexión:**

* ¿Qué lenguaje de programación te llama más la atención y por qué?

Python, porque se convirtió desde que inicié en un desafío, y me impacta que esta en lo primer línea de los lenguajes mas populares y de acceso.

* ¿Te gustaría enfocarte más en el desarrollo móvil o en el desarrollo de escritorio? Explica tu preferencia.

Desarrollo de escritorio, sueño con programar en grande para muchas empresas que a la fecha trabajan manual muchos procesos, además todo lo guardan todavía en armarios con hojas de papel, el cual sigue siendo un problema para el medio ambiente y conservación de reservas naturales.

3. **Responde las siguientes preguntas sobre el origen y la evolución de los lenguajes de programación:**

1. ¿Cuál fue uno de los primeros lenguajes de programación de alto nivel y para qué fue diseñado?

Fortran diseñado para cálculos científicos, Desarrollado originalmente por [IBM](https://es.wikipedia.org/wiki/IBM) en 1957 para el equipo [IBM 704](https://es.wikipedia.org/wiki/IBM_704), y usado para aplicaciones científicas y de ingeniería, el FORTRAN vino a dominar esta área de la programación desde el principio y ha estado en uso continuo por más de medio siglo en áreas de cómputo intensivo tales como la [predicción numérica del tiempo](https://es.wikipedia.org/w/index.php?title=Predicci%C3%B3n_num%C3%A9rica_del_tiempo&action=edit&redlink=1), [análisis de elementos finitos](https://es.wikipedia.org/wiki/M%C3%A9todo_de_los_elementos_finitos), [dinámica de fluidos computacional](https://es.wikipedia.org/wiki/Din%C3%A1mica_de_fluidos_computacional) (CFD), [física computacional](https://es.wikipedia.org/wiki/F%C3%ADsica_computacional) y [química computacional](https://es.wikipedia.org/wiki/Qu%C3%ADmica_computacional). Es uno de los lenguajes más populares en el área de la [computación de alto rendimiento](https://es.wikipedia.org/wiki/Computaci%C3%B3n_de_alto_rendimiento) y es el lenguaje usado para programas que evalúan el desempeño ([benchmark](https://es.wikipedia.org/wiki/Benchmark_(inform%C3%A1tica)" \o "Benchmark (informática))) y el ranking de los [supercomputadores](https://es.wikipedia.org/wiki/Supercomputador) [más rápidos del mundo](https://es.wikipedia.org/wiki/TOP500).[4](https://es.wikipedia.org/wiki/Fortran#cite_note-4)​

2. ¿Qué características hicieron que el lenguaje C, creado en los años 70, fuera tan influyente en la evolución de otros lenguajes?

Desde su creación, C ha tenido un impacto profundo en la informática. Se convirtió rápidamente en el lenguaje de referencia para el desarrollo de sistemas operativos y software de sistemas. UNIX, escrito casi completamente en C, demostró la viabilidad y las ventajas de este lenguaje, lo que llevó a su adopción generalizada.

C no solo ha sido fundamental en el desarrollo de sistemas operativos como UNIX y sus derivados, sino que también ha influido en la creación de muchos otros lenguajes de programación modernos, como C++, C#, Java y JavaScript. Su sintaxis clara y eficiente ha servido de base para estos lenguajes, facilitando la transición de los programadores de C a estos nuevos entornos.

3. ¿Cómo contribuyó el lenguaje COBOL, desarrollado en 1959, al mundo de los negocios y la administración?

Cobol en realidad fue diseñado para desarrollar aplicaciones de grandes negocios (por lo general, orientadas a archivos), no para desarrollar programas de sistema, como desarrollar un sistema operativo o un compilador. Hace 50 años, Cobol era los mejor para organizaciones grandes que tenían muchos datos que administrar. Sustentó el núcleo de las aplicaciones de banca, finanzas, seguros, gobierno y salud. Pero Cobol sigue siendo espléndido donde el paradigma procedimental, en lugar de orientado a objetos, es adecuado.

Por ejemplo, el procesamiento de registros masivos y la reconciliación de cuentas.

Cobol puede ejecutarse en DB2, Oracle o cualquier otra base de datos. Puedes emparejarlo en cualquier momento con una interfaz de usuario atractiva o una interfaz web.

Los compiladores COBOL modernos pueden generar código nativo para Windows, .NET, UNIX, Linux, JVM y la nube. Ahora ofrece todos los beneficios de los lenguajes de programación modernos sin incurrir en costos adicionales.

4. ¿Qué impacto tuvo la aparición de lenguajes de programación como Python (1991) y JavaScript (1995) en el desarrollo de software moderno?

**Python** jugará un papel crucial en el avance de la IA y el aprendizaje profundo, facilitando el desarrollo de tecnologías como vehículos autónomos, sistemas de reconocimiento de voz e imagen y robots inteligentes.

La simplicidad y flexibilidad de Python también lo hacen ideal para el rápido prototipado en startups tecnológicas y para la implementación de soluciones innovadoras en diversas industrias.

**JavaScript** continuará impulsando la innovación en aplicaciones web y móviles.

Con la adopción de Progressive Web Apps (PWA) y el Internet de las Cosas (IoT), JavaScript expandirá su influencia más allá de los navegadores, permitiendo el desarrollo de aplicaciones web que ofrecen una experiencia de usuario similar a las aplicaciones nativas, así como la gestión de dispositivos IoT.

La interactividad, la reactividad y la capacidad de trabajar en tiempo real que ofrece JavaScript serán fundamentales para el desarrollo de aplicaciones y servicios web en la próxima década.

Como ves, ambos lenguajes tienen un futuro prometedor, lleno de innovación y avances tecnológicos.

5. En términos de evolución, ¿cuáles son las diferencias principales entre el desarrollo móvil y el desarrollo de escritorio, y cómo han influido los lenguajes en estos dos entornos?

|  |  |
| --- | --- |
| **DESARROLLO MÓVIL** | **DESARROLLO DE ESCRITORIO** |
| Se refiere a las aplicaciones instaladas y en funcionamiento en un teléfono inteligente o tableta. Estas aplicaciones suelen descargarse de una tienda de aplicaciones y se accede a ellas a través de un icono en la pantalla de inicio. | se refiere a la creación de aplicaciones y software que se instalan y ejecutan en una computadora de escritorio o portátil. Por lo general, se accede a estas aplicaciones a través de un ícono del escritorio o del menú de inicio. |
| Ventajas:  1. Mayor accesibilidad: Las aplicaciones móviles pueden utilizarse en cualquier lugar, siempre que el usuario tenga su teléfono. Esto las hace convenientes para personas cuyo ritmo de vida es relativamente rápido y requiere un acceso rápido y cómodo a tecnología, servicios y gadgets modernos. 2. Audiencia más amplia: hay miles de millones de teléfonos inteligentes y tabletas en uso en todo el mundo, lo que significa que las aplicaciones móviles tienen el potencial de llegar a una audiencia mucho más amplia que las aplicaciones de escritorio. 3. Actualizaciones fáciles: las tiendas de aplicaciones móviles facilitan la actualización y el mantenimiento de las aplicaciones, ya que los usuarios reciben una notificación cuando hay una actualización disponible y pueden instalarla con solo unos pocos toques. | Ventajas:   1. Mayor poder de procesamiento: las computadoras de escritorio generalmente tienen más poder de procesamiento que los dispositivos móviles, lo que puede beneficiar tareas que consumen muchos recursos, como la edición de video o el modelado 3D. 2. Pantallas más grandes: Las computadoras de escritorio también tienden a tener pantallas más grandes, lo que hace que sea más fácil ver e interactuar con la interfaz. 3. Más almacenamiento: las computadoras de escritorio suelen tener más almacenamiento que los dispositivos móviles, lo que puede ser útil para diseños que necesitan almacenar una gran cantidad de datos. |
| Desventajas:  1. Potencia de procesamiento limitada: los dispositivos móviles generalmente tienen menos potencia de procesamiento que las computadoras de escritorio, lo que puede ser una limitación para tareas que requieren un uso intensivo de recursos. 2. Pantallas más pequeñas: los dispositivos móviles tienen pantallas más pequeñas, lo que hace que sea más difícil ver e interactuar con la interfaz. 3. Restricciones de almacenamiento: los dispositivos móviles a menudo tienen menos almacenamiento que las computadoras de escritorio, lo que puede ser un problema para los diseños que almacenan grandes cantidades de datos. | Desventajas:  1. Accesibilidad limitada: las aplicaciones de escritorio solo se pueden usar cuando el usuario está cerca de una computadora, lo que puede resultar inconveniente para las personas que siempre están en movimiento. 2. Problemas de compatibilidad: diferentes computadoras de escritorio pueden tener diferentes sistemas operativos, hardware y software, lo que dificulta garantizar que el diseño sea compatible con todos los sistemas. 3. Tasa de adopción más baja: las aplicaciones de escritorio generalmente tienen una tasa de adopción más baja que las aplicaciones móviles, ya que es más probable que las personas descarguen una aplicación a su teléfono o tableta que a su computadora. |