Actividad 1 – Antes de comenzar

**Nombre y apellidos:** Rubén García Ruiz

**Realiza una búsqueda contrastada** entre diferentes sitios web para responder a las siguientes preguntas:

* ¿Qué es JavaScript y cuál es su papel en el desarrollo web? (1pto.)

**JavaScript** es un lenguaje de programación de alto nivel que se utiliza principalmente en el desarrollo web. Su papel en el desarrollo web es fundamental, ya que permite agregar interactividad y dinamismo a las páginas web.

Permite lo siguiente:

* **Lenguaje de programación**
* **Interacción del usuario**
* **Dinamismo**
* **Validación de datos**
* **Comunicación con el servidor**
* **Creación de aplicaciones web**

En resumen, JavaScript desempeña un papel esencial en el desarrollo web al permitir la creación de sitios web interactivos y dinámicos. Es un lenguaje versátil que se ejecuta en el navegador del usuario y puede interactuar con el servidor, lo que lo convierte en una herramienta poderosa para crear una amplia variedad de aplicaciones web.

* Explique la diferencia entre JavaScript y Java. ¿Son lo mismo? (2ptos.)

JavaScript y Java son dos lenguajes de programación que, aunque parezcan que tienen nombres similares, son muy diferentes en términos de características, aparte de su uso y propósito. Por lo que puede sonar parecido pero no son lo mismo, por eso lo explicaremos con los siguientes casos:

**1- Su origen, es decir, su creación y evolución:**

**JavaScript** y Java, aunque fuesen desarrollados en la misma época, alrededor de los años 90. Y como hemos detallado en la pregunta del ejercicio anterior, JavaScript se utiliza principalmente en el desarrollo web para agregar interactividad a las páginas en el navegador, pero es un lenguaje independiente de la web. **Java** se utiliza en una variedad de aplicaciones, incluyendo aplicaciones de escritorio, aplicaciones móviles, servidores y sistemas embebidos.

**2- Tipo de Lenguaje:**

**JavaScript** es un lenguaje de programación de alto nivel **interpretado** que se ejecuta en el navegador del cliente. Conocido como un lenguaje de scripting del lado del cliente. En el otro margen, aunque **Java** también es un lenguaje de programación de alto nivel, es **compilado** y que se ejecuta en una máquina virtual Java (JVM). Java se considera un lenguaje de programación del lado del servidor, aunque también se utiliza en otros contextos.

**3- Sintaxis y Estilo**:

**JavaScript**: Tiene una sintaxis más similar a otros lenguajes de programación como C++ y C#. Es más flexible y permite la tipificación dinámica.

**Java**: Tiene una sintaxis más estricta y verbosa en comparación con JavaScript. Requiere una tipificación estática, lo que significa que debes declarar el tipo de datos de las variables antes de usarlas.

**4- Uso y Aplicación:**

**JavaScript**: Se utiliza principalmente en el desarrollo web para agregar interactividad a las páginas, validar formularios, realizar peticiones a servidores, etc.

**Java**: Se utiliza en una amplia variedad de aplicaciones, desde desarrollo de aplicaciones de escritorio (Java SE) hasta aplicaciones empresariales en servidores (Java EE) y desarrollo móvil (Android).

**5- Ecosistema y Bibliotecas:**

**JavaScript**: Tiene un ecosistema rico, es decir, muy amplio y con muchas bibliotecas, además de frameworks populares como React, Angular, y Vue.js para el desarrollo web.

**Java**: También tiene un ecosistema sólido con una amplia gama de bibliotecas y frameworks para diferentes aplicaciones, como Spring para desarrollo empresarial y Android para desarrollo móvil.

En resumen, JavaScript y Java son dos lenguajes de programación distintos con propósitos y características diferentes. La única similitud significativa es el nombre, que puede causar confusión, pero no están relacionados en términos de diseño o uso.

* Encuentra uno o dos rankings de popularidad o uso de lenguajes de programación y pega los pantallazos correspondientes de los primeros lenguajes que los encabezan. ¿Qué posición ocupa JavaScript? (1pto.)

Los 10 lenguajes de programación más solicitados y que a su vez se ofertan en LinkedIn. Como vemos, **JavaScript ocupa la tercera posición**.

1. Java: 21% | 16.238 ofertas de trabajo.
2. Python: 20% | 15.360 ofertas de trabajo.
3. **JavaScript**: 19% | 14.840 ofertas de trabajo.
4. C++: 17% | 13.439 ofertas de trabajo.
5. C#: 5% | 3.534 ofertas de trabajo.
6. PHP: 4% | 3.314 ofertas de trabajo.
7. C: 3% | 2.392 ofertas de trabajo.
8. Swift developer: 2% | 1.781 ofertas de trabajo.
9. Typescript: 2% | 1.607 ofertas de trabajo.
10. R: 1% | 989 ofertas de trabajo.



En cuanto a uso de lenguajes de programación este es un gráfico detallado, dónde podemos ver que **JavaScrip ocupa la primera posición**:



Fuente: <https://www.hackaboss.com/blog/ranking-de-lenguajes-de-programacion>

Nota: No olvides citar la fuente de donde proviene la imagen (ideal formato APA7)

* ¿Qué es el back-end? ¿y el front-end? ¿Podrías asociarlos a cliente o servidor? (2ptos.)

El término "back-end" y "front-end" se utilizan comúnmente en el desarrollo web para describir dos aspectos diferentes de una aplicación o sitio web, y están **asociados a las funciones de servidor y cliente** de una aplicación web.

**Front-end**:

El front-end se refiere a la parte de una aplicación o sitio web con la que interactúa directamente el usuario final. Incluye la interfaz de usuario, el diseño, la presentación y la lógica que se ejecuta en el navegador web del usuario.

**Asociación con el cliente**: El front-end está relacionado con el cliente de la aplicación, es decir, el navegador web del usuario. Por decirlo de alguna manera, es la cara visible de la aplicación con la que el usuario interactúa directamente. Se encarga de la presentación de la información y la interacción con el usuario, como formularios, botones, animaciones y elementos visuales en general.

**Back-end**:

El back-end se refiere a la parte de una aplicación o sitio web que se encuentra en el servidor. Incluye la lógica de negocio, el procesamiento de datos, el acceso a bases de datos y otras operaciones que no son visibles para el usuario final.

**Asociación con el servidor**: El back-end está relacionado con el servidor de la aplicación. Se encarga de procesar las solicitudes que provienen del front-end, interactuar con bases de datos, autenticar usuarios, realizar cálculos complejos y llevar a cabo todas las operaciones que no son directamente visibles para el usuario.

Para hacernos una idea de la **relación entre el front-end y el back-end** en una aplicación web, puedes imaginar un sitio de comercio electrónico. El front-end sería la parte de la aplicación que el usuario ve y utiliza para buscar productos, agregarlos al carrito de compras y realizar el pago. El back-end sería la parte que gestiona el inventario de productos, procesa los pagos, administra las cuentas de usuario y realiza todas las operaciones necesarias para que el sitio funcione correctamente.

En resumen, el front-end se relaciona con la interfaz de usuario y la interacción directa del usuario en el navegador (cliente), mientras que el back-end se relaciona con la lógica de negocio y el procesamiento en el servidor (servidor). Ambos trabajan juntos para proporcionar una experiencia completa en una aplicación web.

* Define Framework y librería. ¿En qué se diferencian? Añade tres ejemplos de cada tipo con una breve descripción de cada uno. (2ptos.)

Framework y librería son dos conceptos relacionados en el mundo de la programación que se utilizan para simplificar y agilizar el desarrollo de software. Aunque tienen similitudes, también tienen diferencias clave, pero antes vamos a describirlos.

**FRAMEWORK**: Un framework es una estructura o conjunto de herramientas que proporciona una arquitectura y un conjunto de reglas para el desarrollo de aplicaciones. Los frameworks suelen definir la estructura de la aplicación, las relaciones entre los componentes y las convenciones de nomenclatura.

Los desarrolladores trabajan dentro de las restricciones del framework y utilizan sus funcionalidades para construir aplicaciones. Un framework suele ser más invasivo en el código de la aplicación y dicta la organización general.

**Ejemplos de frameworks incluyen**:

**- Angular**: Un framework de JavaScript desarrollado por Google para la creación de aplicaciones web de una sola página (SPA). Proporciona una estructura sólida y herramientas para la creación de componentes reutilizables y la gestión del enrutamiento.

**- Ruby on Rails**: Un framework de desarrollo web basado en el lenguaje Ruby. Rails sigue el principio de convención sobre configuración (CoC) y hace hincapié en la rapidez de desarrollo y la legibilidad del código.

**- Django**: Un framework de desarrollo web de Python que se centra en la facilidad de uso y la productividad. Proporciona muchas características listas para usar, como un sistema de administración de bases de datos y autenticación de usuarios.

**LIBRERÍA**: Una librería es un conjunto de funciones, módulos o clases predefinidos que los desarrolladores pueden utilizar en sus aplicaciones para realizar tareas específicas. Las librerías no dictan la estructura de la aplicación y generalmente son menos invasivas en el código.

Los desarrolladores pueden elegir usar librerías de acuerdo con sus necesidades y pueden incorporarlas selectivamente en su aplicación. No se ven obligados a seguir una estructura específica.

**Ejemplos de librerías incluyen**:

**- jQuery**: Una librería de JavaScript que simplifica la manipulación del DOM, la gestión de eventos y las llamadas AJAX. Facilita la escritura de código JavaScript más conciso y compatible con diferentes navegadores.

**- React**: Una librería de JavaScript desarrollada por Facebook para la creación de interfaces de usuario interactivas. React se utiliza comúnmente junto con otras bibliotecas y herramientas para construir aplicaciones de una sola página (SPA).

**- NumPy**: Una librería de Python para realizar cálculos numéricos y matemáticos de alta calidad. Proporciona estructuras de datos eficientes para trabajar con matrices y arreglos multidimensionales.

En resumen, **la diferencia clave entre un framework y una librería** radica en el grado de control que otorgan y en la estructura que imponen en una aplicación.

Los frameworks son más invasivos y proporcionan una estructura más completa, mientras que las librerías ofrecen funcionalidades específicas que los desarrolladores pueden utilizar según sus necesidades sin imponer una estructura particular.

**NOTA**: *Se tendrán en cuenta los formatos de presentación: páginas numeradas, formato de cita de la figura o tabla, faltas de ortografía, redacción, cohesión en el uso del mismo formato de letra (responder en Arial 12) y justificación (hasta dos puntos). El resto de los puntos se repartirán a razón de un punto por respuesta correcta (hasta ocho puntos).*

**\*REALIZADO CON WordPad en Pc clase, no deja numerar las páginas. Por lo demás puesto todo correcto, formato Arial 12 y párrafos justificados.**

**\*ABRIR CON WordPad, si se abre con Word se desajusta los párrafos.**

Rubén García Ruiz.