

Los 10 temas de tecnologia

Kevin Camilo Muñoz Campos

Investigación - 2694667

Servicio Nacional de Aprendizaje Sena

9/12/2024

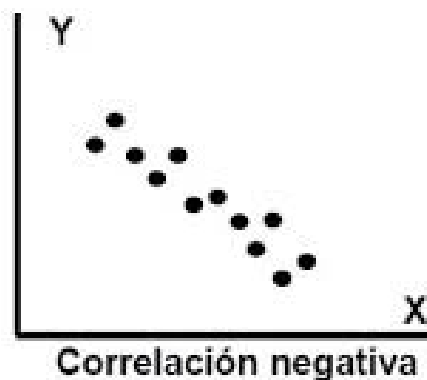
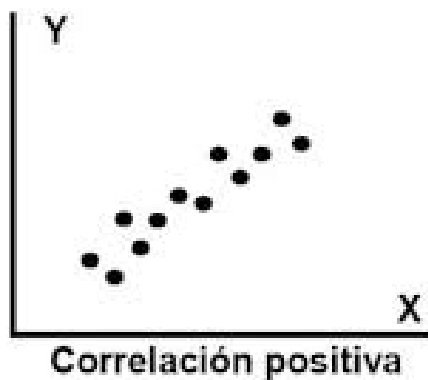
Neiva-Huila

¿Cuáles son las correlaciones positivas y negativas en programación?

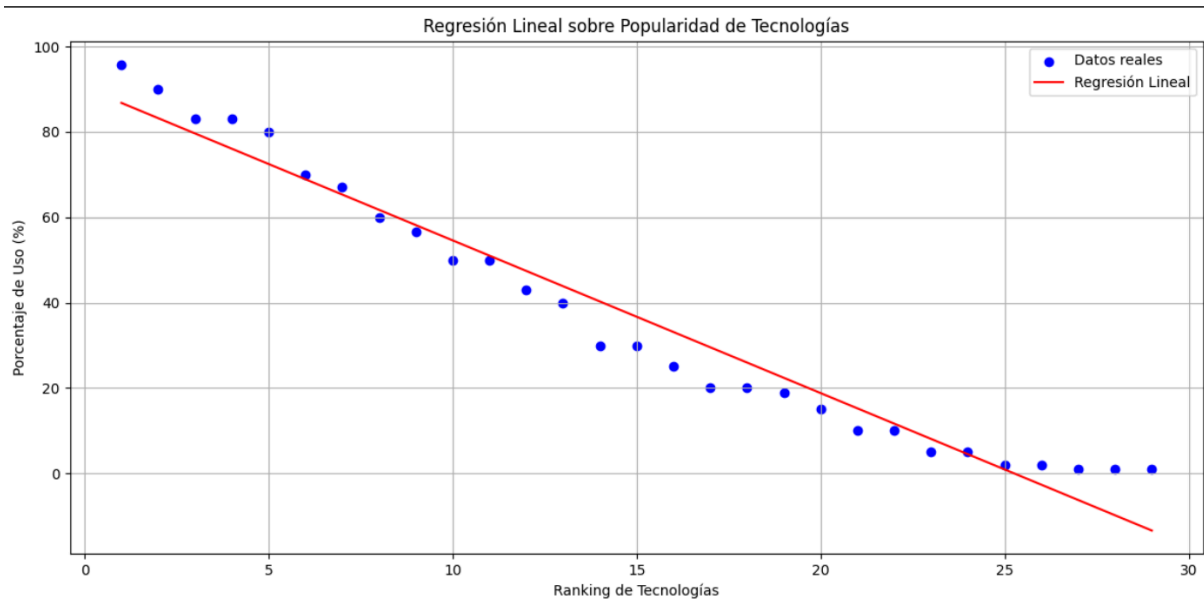
En programación, el término "correlación" generalmente no se refiere a un concepto técnico específico de código, sino a una metáfora tomada de las ciencias estadísticas que describen la relación entre dos variables o elementos. Las correlaciones en programación pueden referirse a cómo ciertas variables, procesos o comportamientos se afectan entre sí.

Correlación positiva: Se refiere a la relación entre dos elementos que se mueven en la misma dirección. Es decir, cuando una variable aumenta, la otra también tiende a aumentar. En el contexto de la programación, una activación positiva podría significar, por ejemplo, que cuando se incrementa un valor o se realiza una acción, otra acción relacionada también aumenta o mejora. Un ejemplo podría ser el caso de aumentar el rendimiento de un programa al mejorar el número de hilos de procesamiento en un sistema multihilo, lo que genera una mejora en la eficiencia.

Ejemplo de activación positiva: Supongamos que en una aplicación se tiene un contador de visitas a una página web. Si el número de visitas aumenta, el contador de visitas también aumentará de manera positiva. En programación, esta relación podría verse reflejada en el código como una acción que aumenta una variable cada vez que ocurre el evento.



Gráfica 1 Tecnologías más usadas



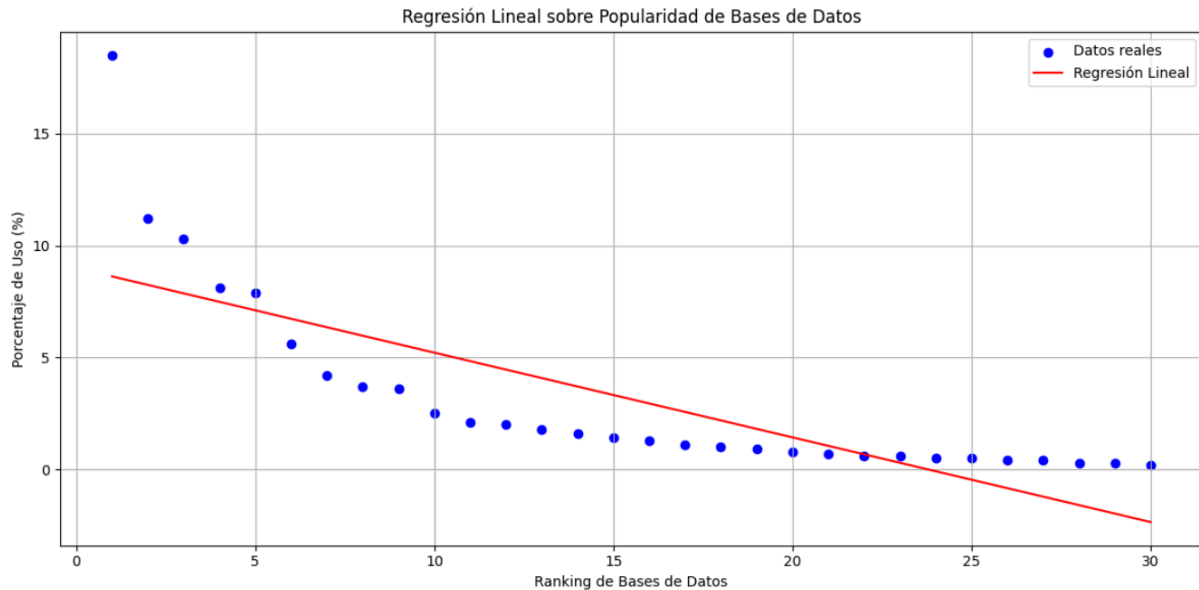
Hipótesis:

El internet sigue siendo la tecnología más utilizada, seguida de las computadoras y dispositivos móviles, lo que subraya la dependencia digital en la vida diaria. Los servicios como el correo electrónico y la búsqueda en línea mantienen una alta prevalencia debido a su utilidad básica, mientras que actividades como el comercio electrónico y la banca en línea reflejan un cambio hacia una economía digital. Sin embargo, tecnologías emergentes como blockchain, criptomonedas y realidad virtual tienen una adopción limitada debido a barreras como la falta de accesibilidad, conocimiento técnico o aplicaciones inmediatas.

Conclusion:

La alta adopción de tecnologías fundamentales como Internet y teléfonos móviles demuestra su relevancia en la vida diaria. Por otro lado, la baja aceptación de tecnologías emergentes como blockchain y realidad virtual sugiere que aún se enfrentan desafíos relacionados con costos y usabilidad para masificarse.

Gráfica 2 Bases de datos



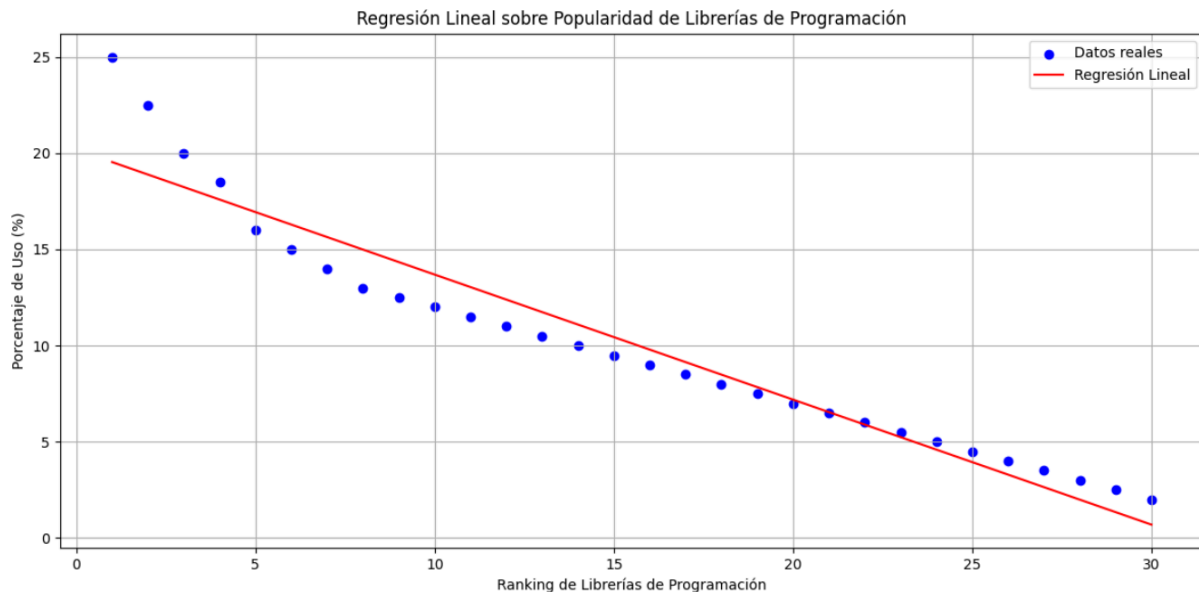
Hipótesis:

La gráfica sugiere una relación inversamente proporcional entre el ranking de popularidad de una base de datos y su porcentaje de uso actual. Es decir, a medida que una base de datos se vuelve más popular y asciende en el ranking, su porcentaje de uso tiende a disminuir con el tiempo. Esta tendencia podría explicarse por factores como el efecto de novedad, la saturación del mercado y la competencia. En otras palabras, la popularidad inicial suele ser seguida por una disminución gradual en el uso a medida que la base de datos madura y surgen nuevas alternativas. Este patrón sugiere un ciclo de vida típico para las tecnologías de bases de datos, donde la adopción inicial es seguida por una fase de madurez y, eventualmente, un declive.

Conclusion:

El predominio de bases de datos como MySQL y PostgreSQL indica su confiabilidad y robustez en aplicaciones empresariales. MongoDB lidera entre las bases de datos NoSQL, destacando su flexibilidad para aplicaciones modernas. Por otro lado, tecnologías como Redis y SQLite responden a necesidades específicas de almacenamiento en caché y desarrollo móvil, consolidando su relevancia en nichos particulares.

Gráfica 3 Librerías de programación



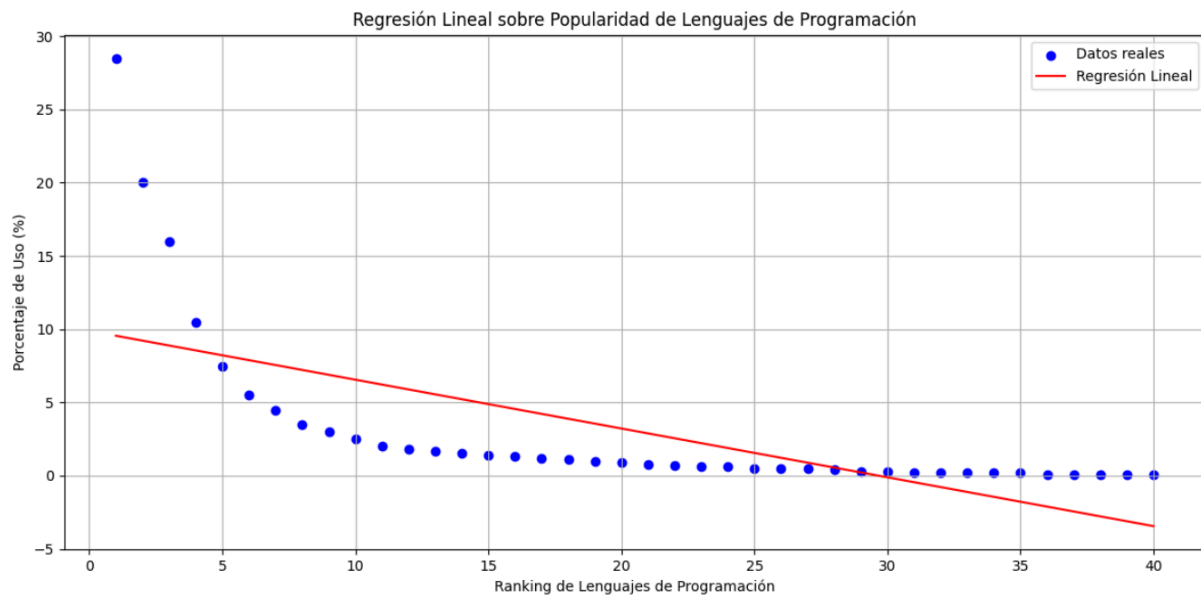
Hipótesis:

La gráfica sugiere una relación inversamente proporcional entre el ranking de popularidad de una librería de programación y su porcentaje de uso actual. Es decir, a medida que una librería se vuelve más popular y asciende en el ranking, su porcentaje de uso tiende a disminuir con el tiempo. Esta tendencia podría explicarse por factores como el efecto de novedad, la saturación del mercado y la competencia. En otras palabras, la popularidad inicial suele ser seguida por una disminución gradual en el uso a medida que la librería madura y surgen nuevas alternativas. Este patrón sugiere un ciclo de vida típico para las tecnologías de librerías de programación, donde la adopción inicial es seguida por una fase de madurez y, eventualmente, un declive.

Conclusion:

La popularidad de librerías como NumPy y Pandas confirma que Python sigue siendo la opción preferida para la manipulación de datos y análisis. Sin embargo, herramientas avanzadas como TensorFlow y PyTorch reflejan el crecimiento del aprendizaje automático, consolidando su papel en la inteligencia artificial y en la investigación tecnológica.

Gráfica 4 Lenguajes de programación



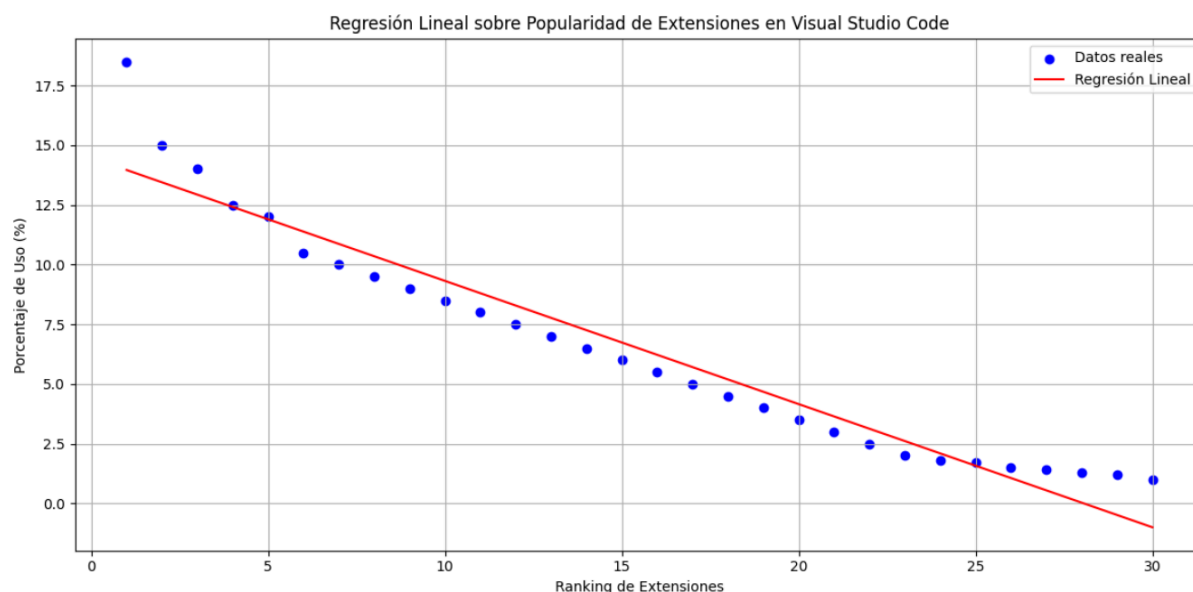
Hipótesis:

La gráfica sugiere una relación inversamente proporcional entre el ranking de popularidad de un lenguaje de programación y su porcentaje de uso actual. Es decir, a medida que un lenguaje se vuelve más popular y asciende en el ranking, su porcentaje de uso tiende a disminuir con el tiempo. Esta tendencia podría explicarse por factores como el efecto de novedad, la saturación del mercado y la competencia. En otras palabras, la popularidad inicial suele ser seguida por una disminución gradual en el uso a medida que el lenguaje madura y surgen nuevas alternativas. Este patrón sugiere un ciclo de vida típico para los lenguajes de programación, donde la adopción inicial es seguida por una fase de madurez y, eventualmente, un declive.

Conclusion:

La amplia aceptación de lenguajes como Python y Java resalta su versatilidad y relevancia tanto en el desarrollo empresarial como en la inteligencia artificial. Mientras tanto, lenguajes como C# y JavaScript mantienen su posición debido a su aplicabilidad en desarrollo de software y web, respectivamente. La adopción de lenguajes como TypeScript refleja un interés creciente en el desarrollo más estructurado y eficiente.

Gráfica 5 Extensiones en Visual Studio Code



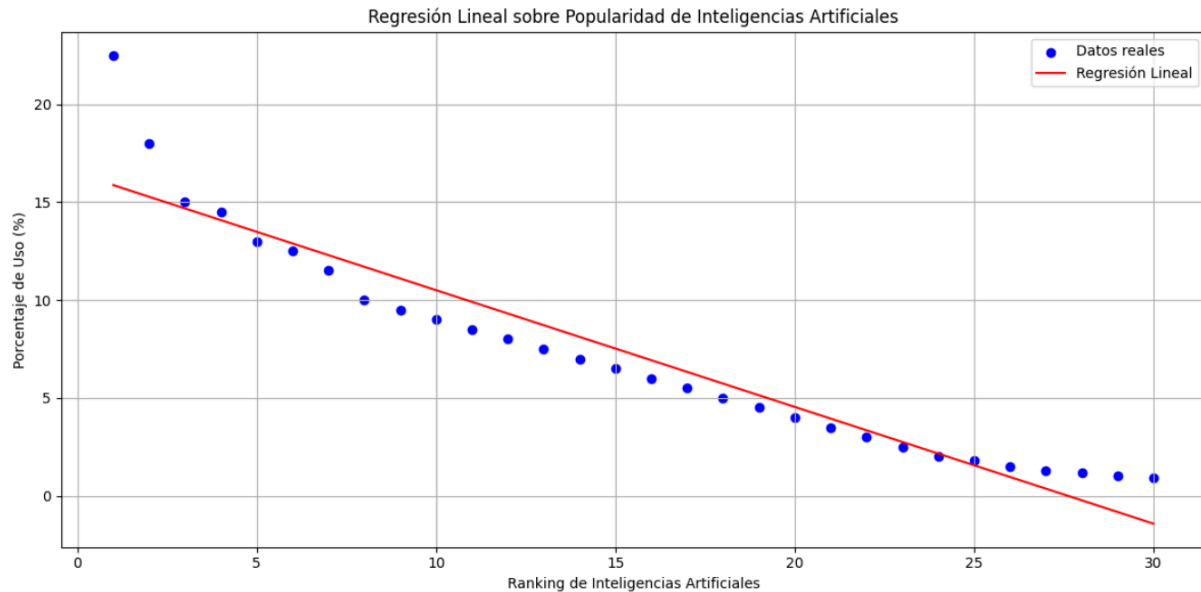
Hipótesis:

Extensiones como Prettier y ESLint son indispensables en el flujo de desarrollo debido a su capacidad para garantizar la calidad y el formato del código. Live Server es ampliamente adoptado por desarrolladores web por su funcionalidad de recarga en tiempo real. Herramientas como GitLens y Docker reflejan la importancia del control de versiones y la virtualización, mientras que otras, como Remote - SSH y WSL, son populares entre desarrolladores que necesitan trabajar en Múltiples entornos. Las extensiones para lenguajes específicos como Python o C# tienen menor adopción en comparación con las herramientas generalistas que abarcan múltiples lenguajes.

Conclusion:

La preferencia por extensiones como Prettier y ESLint muestra el interés de los desarrolladores en mantener un código limpio y bien estructurado. Por otro lado, herramientas como GitLens y Remote - SSH refuerzan la tendencia hacia entornos colaborativos y remotos, adaptándose a nuevas formas de trabajo distribuidos.

Gráfica 6 Inteligencias Artificiales



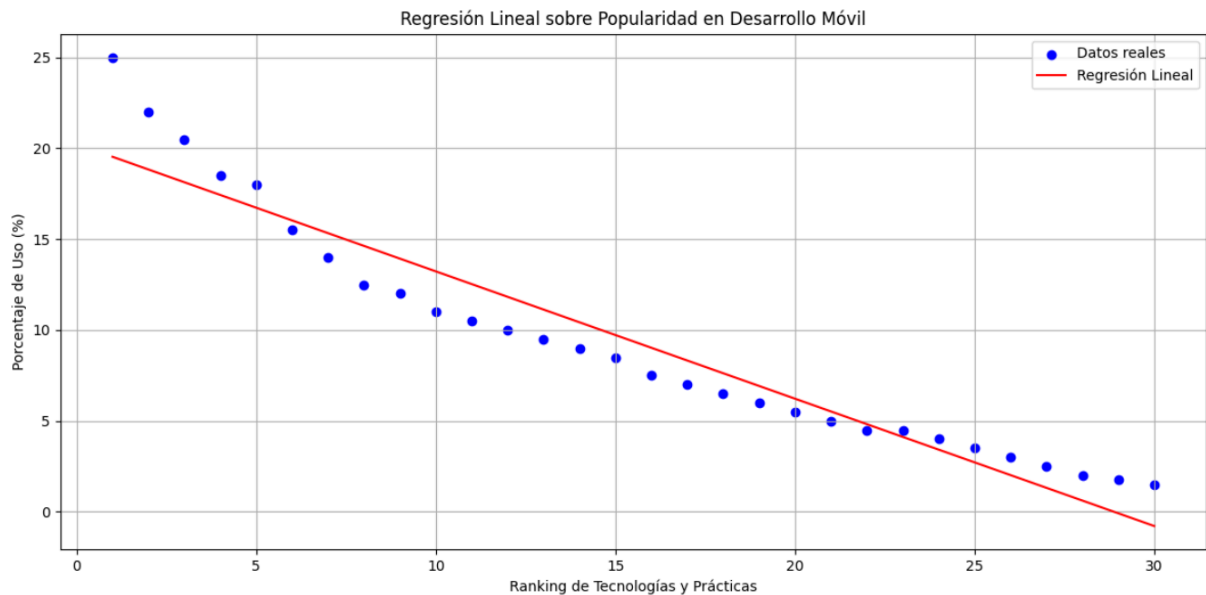
Hipótesis:

Modelos como ChatGPT y BERT lideran el uso debido a su versatilidad en tareas de procesamiento del lenguaje natural y generación de contenido. Herramientas de generación de imágenes como DALL-E, MidJourney y Stable Diffusion reflejan un crecimiento en aplicaciones creativas. Modelos avanzados como GPT-4 y Claude AI son preferidos por desarrolladores y empresas que requieren soluciones de alta precisión. Aunque herramientas tradicionales como IBM Watson o SageMaker tienen presencia en sectores específicos, su adopción es menor debido a su complejidad y costos asociados. Las plataformas como PyTorch Lightning y FastAI están ganando tracción entre investigadores y desarrolladores de ML.

Conclusion:

El uso de herramientas como ChatGPT, DALL-E y GPT-4 destaca el interés en capacidades avanzadas de generación de contenido e interacción humano-máquina. Mientras tanto, tecnologías como Amazon SageMaker y Hugging Face Transformers fortalecen el desarrollo de modelos personalizados en diversos sectores.

Gráfica 7 Desarrollo Movil



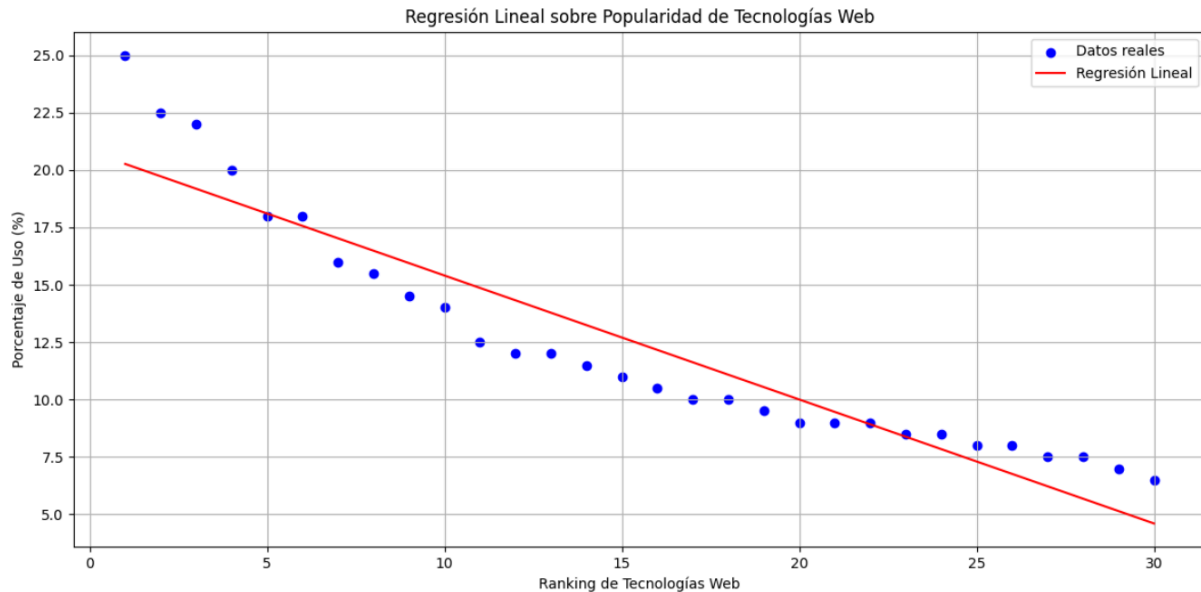
Hipótesis:

Las plataformas Android e iOS dominan el desarrollo móvil debido a su alta cuota de mercado, pero tecnologías multiplataforma como Flutter y React Native ganan terreno rápidamente gracias a su capacidad para reducir costos y tiempos de desarrollo. Las bases de datos como Firebase y SQLite son preferidas por su flexibilidad y facilidad de integración, mientras que los desarrolladores también priorizan la implementación de API REST y notificaciones push como funcionalidades esenciales. La adopción de tecnologías más avanzadas como ARKit o Core ML es limitada debido a su nicho especializado y los recursos necesarios para implementarlas.

Conclusion:

La predominancia de Kotlin/Java y Swift en sus respectivos ecosistemas indica una fuerte preferencia por el desarrollo nativo. Al mismo tiempo, Flutter y React Native emergen como soluciones efectivas para aplicaciones multiplataforma, demostrando que los desarrolladores buscan eficiencia y cobertura de mercado con un solo código base.

Gráfica 8 Desarrollo Web



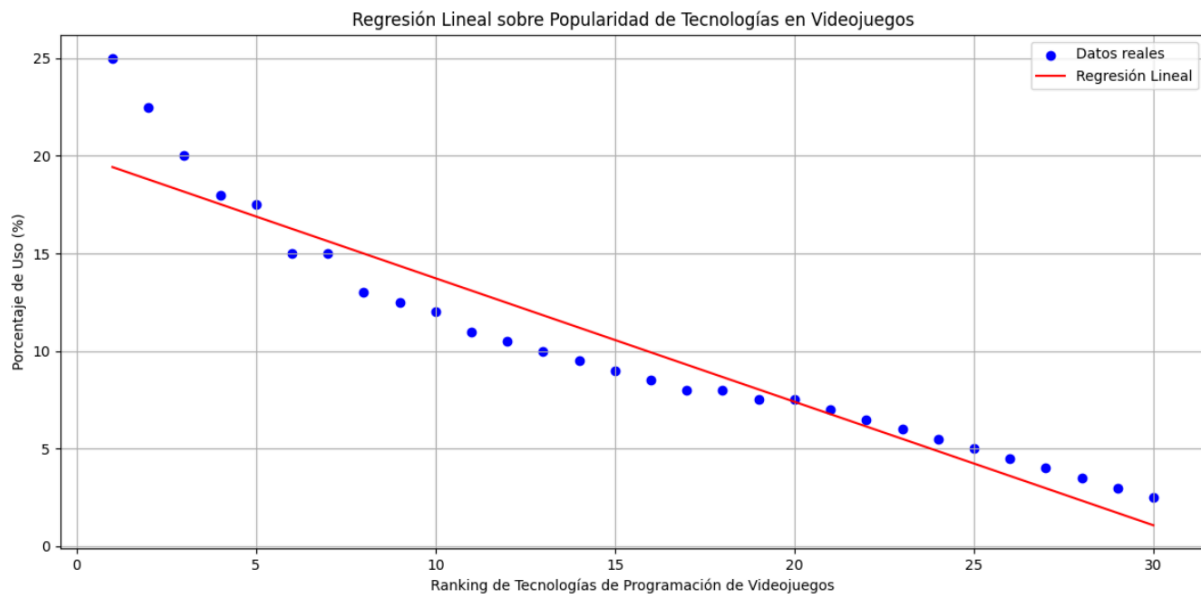
Hipótesis:

HTML, CSS y JavaScript forman la base del desarrollo web, mientras que frameworks como React.js, Angular y Vue.js son esenciales para crear interfaces de usuario dinámicas. Las API REST y GraphQL son ampliamente adoptadas debido a su capacidad para manejar datos en tiempo real. Bases de datos como MySQL y PostgreSQL tienen una presencia significativa, aunque soluciones más modernas como MongoDB crecen en popularidad gracias a su enfoque en documentos. Herramientas como Webpack y Babel destacan entre desarrolladores avanzados para optimizar flujos de trabajo, mientras que las tecnologías relacionadas con OAuth 2.0 y JWT reflejan la necesidad de seguridad en aplicaciones modernas.

Conclusion:

Frameworks modernos como React.js y Angular destacan por su eficiencia en aplicaciones dinámicas, mientras que herramientas como PHP y Laravel siguen teniendo un espacio importante en proyectos más tradicionales. La adopción de APIs REST y GraphQL indica una evolución hacia una mejor gestión y entrega de datos.

Gráfica 9 Desarrollo en Videojuegos



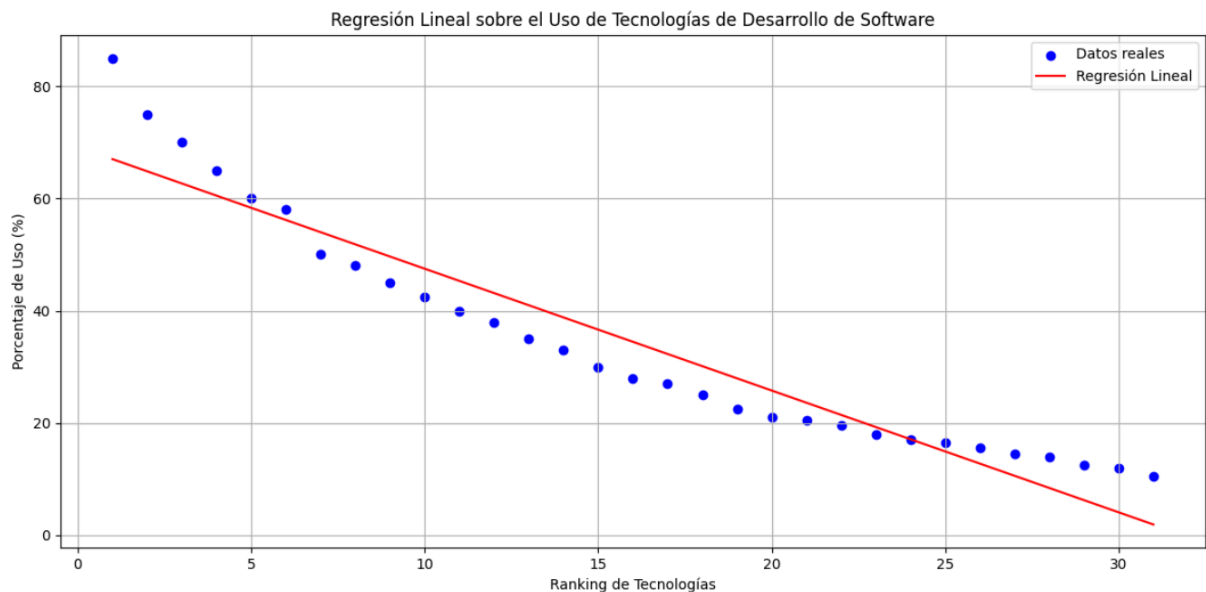
Hipótesis:

Unity y Unreal Engine son los motores más utilizados debido a su versatilidad y potencia, especialmente en juegos AAA y proyectos independientes. Herramientas más accesibles como Godot y GameMaker Studio son populares entre desarrolladores principiantes y pequeños estudios gracias a su simplicidad. Tecnologías complementarias como Blender para modelado 3D y físicas como Havok refuerzan el ecosistema de desarrollo. Sin embargo, la adopción de herramientas específicas como bibliotecas de redes o optimización de rendimiento es menor, ya que suelen ser utilizadas por desarrolladores avanzados o en proyectos con necesidades específicas.

Conclusion:

Unity y Unreal Engine lideran el desarrollo de videojuegos gracias a su potencia y flexibilidad. Sin embargo, motores como Godot y GameMaker Studio ganan popularidad por su accesibilidad, fomentando una mayor participación de pequeños desarrolladores y estudios independientes en el mercado.

Gráfica 10 Desarrollo de Software



Hipótesis:

Git y Docker lideran las herramientas más utilizadas, lo que refleja la importancia del control de versiones y la virtualización en los flujos de desarrollo modernos. Kubernetes tiene una gran adopción en empresas que buscan escalabilidad en la gestión de contenedores, mientras que herramientas como Jenkins y CI/CD demuestran la tendencia hacia la automatización. Aunque lenguajes como Python, Java y JavaScript son comunes, frameworks como React y Angular tienen mayor protagonismo debido a su uso en aplicaciones web dinámicas. Por otro lado, tecnologías más tradicionales como PHP y Laravel tienen menor adopción en comparación con soluciones más modernas.

Conclusion:

El uso extendido de herramientas como Git, Docker y Jenkins refleja un enfoque robusto hacia la integración continua y la contenedorización. Esto evidencia la prioridad en la colaboración efectiva, la escalabilidad y el manejo de proyectos modernos, donde la eficiencia operativa es clave.