



UNIVERSIDAD NACIONAL DE ASUNCIÓN
FACULTAD DE INGENIERÍA



DIPLOMADO DE PROGRAMACIÓN EN PYTHON
ESPECIALIZADO PARA INGENIEROS

Informe de Proyecto Final

**“Explorador de Tendencia de Tecnologías mediante el
análisis de datos en la Ingeniería Informática”**

Profesores: - Ing. Diego Stalder
 - Ing. Christian Torres
 - Ing. José María Rivas

Alumna: Samira López

2025

Índice

| | |
|----------------------------------|---|
| Introducción..... | 1 |
| Objetivos..... | 1 |
| - Objetivo General: | 1 |
| - Objetivos Específicos: | 1 |
| Metodología..... | 1 |
| Análisis de Resultados..... | 3 |
| - Tendencias de Adopción | 4 |
| - Distribución del Mercado | 4 |
| - Dinámicas Evolutivas | 5 |
| Conclusiones..... | 5 |

Introducción

El presente informe tiene como objetivo analizar la evolución histórica del interés y uso de diversas tecnologías de programación en los últimos años. A través del cálculo de métricas de crecimiento, volatilidad y participación, se busca identificar patrones de adopción, estabilidad y posibles tendencias emergentes.

El análisis se realizó utilizando un entorno de desarrollo en Python, combinando bibliotecas especializadas en manipulación y visualización de datos. Esto permitió procesar información histórica y construir indicadores que reflejan el comportamiento relativo de cada tecnología dentro del ecosistema actual del desarrollo de software.

Objetivos

- Objetivo General

Desarrollar un sistema capaz de examinar la evolución y popularidad de tecnologías informáticas utilizando datos adquiridos a partir de fuentes abiertas, facilitando la toma de decisiones sobre qué lenguajes o frameworks priorizar en el desarrollo profesional o en la formación.

- Objetivos Específicos
 - Obtención de datos acerca de las tendencias tecnológicas a través de APIs como GitHub Trending y Stack Overflow.
 - Implementar procesos de limpieza y normalización de datos para estandarizar las tecnologías analizadas.
 - Ejecutar un análisis estadístico y temporal para evaluar el incremento o la disminución de la popularidad.
 - Crear visualizaciones que permitan comparaciones, tales como líneas de tiempo y listas clasificatorias.
 - Elaborar reportes que orienten sobre las tecnologías más importantes o emergentes del momento.

Metodología

El estudio siguió un enfoque cuantitativo, apoyado en la recopilación, procesamiento y análisis de datos de popularidad tecnológica. El procedimiento se estructuró en las siguientes etapas:

1. Recolección de datos históricos: se obtuvieron registros de menciones y actividad asociada a distintas tecnologías de programación a lo largo de varios años mediante el uso de APIs para la obtención de los mismos.
2. Limpieza y normalización: los datos se depuraron para eliminar inconsistencias, valores nulos y variaciones no representativas.
3. Cálculo de métricas: se estimaron indicadores como el CAGR (tasa de crecimiento anual compuesta), la volatilidad y un ranking comparativo de desempeño histórico.

4. Visualización de resultados: se generaron gráficos de tendencias, participación y estabilidad que facilitaron la interpretación.
5. Síntesis analítica: se identificaron patrones comunes, anomalías y posibles direcciones de cambio en la popularidad de las tecnologías.

Fuentes de datos:

- Stack Overflow API (tags más populares por fecha).
- GitHub Trending (repositorios y lenguajes más populares).

Bibliotecas:

- Análisis de Datos y Cálculos

`pandas` - Manipulación y análisis de datos estructurados

`numpy` - Cálculos numéricos y operaciones con arrays/matrices

- APIs y Conexiones Externas

`requests` - Hace peticiones HTTP a APIs web

`os + dotenv` - Gestionar variables de entorno y claves API de forma segura

- Visualización de Datos

`matplotlib.pyplot` - Gráficos estáticos 2D

`plotly.express` - Gráficos interactivos de alto nivel

`plotly.graph_objects` - Gráficos interactivos más personalizables

- Manejo de Tiempo y Fechas

`datetime` - Trabaja con fechas, horas y períodos temporales

`time` - Funciones relacionadas con tiempo del sistema

- Utilidades

`random` - Genera números aleatorios y muestreos

`warnings` - Controla mensajes de advertencia

Análisis de Resultados

Luego del análisis de datos, los siguientes gráficos fueron obtenidos:

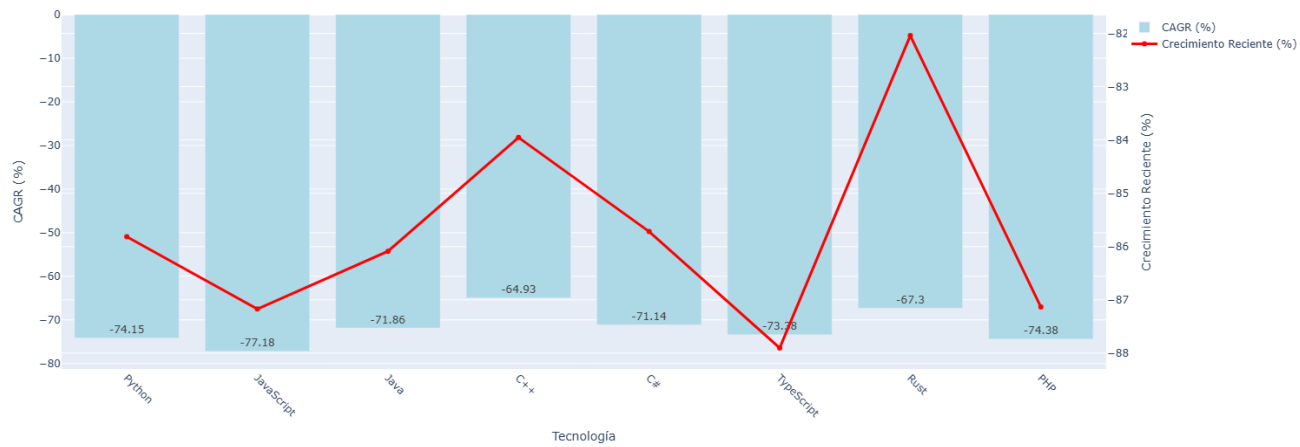


Figura 1. (CAGR vs. Crecimiento Reciente)

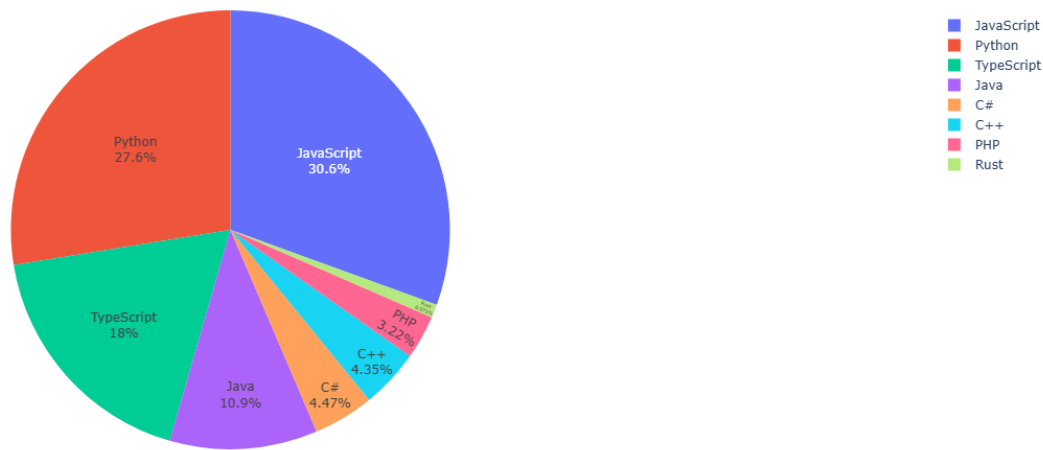


Figura 2. (Gráfico de distribución de tecnologías -2025)

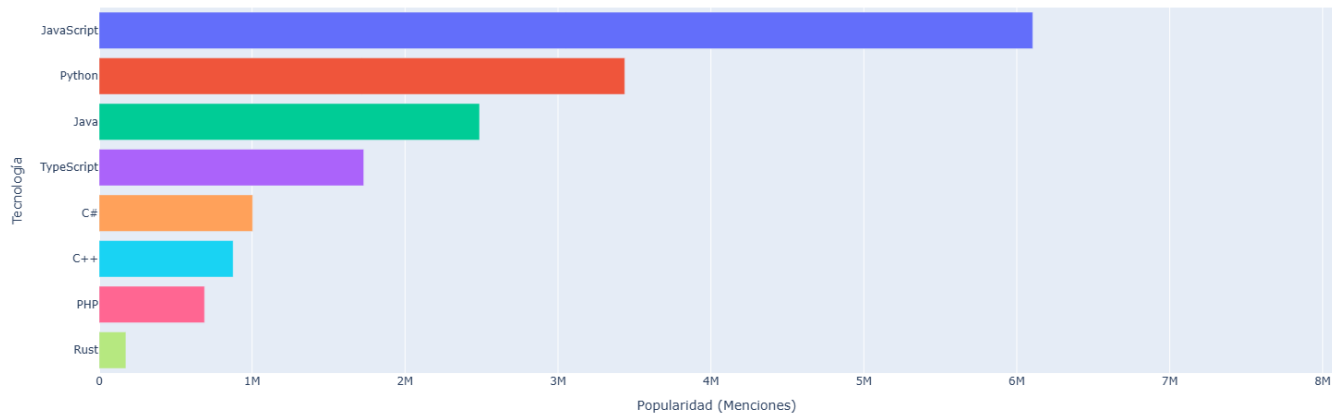


Figura 3. (Ranking anual -2023)

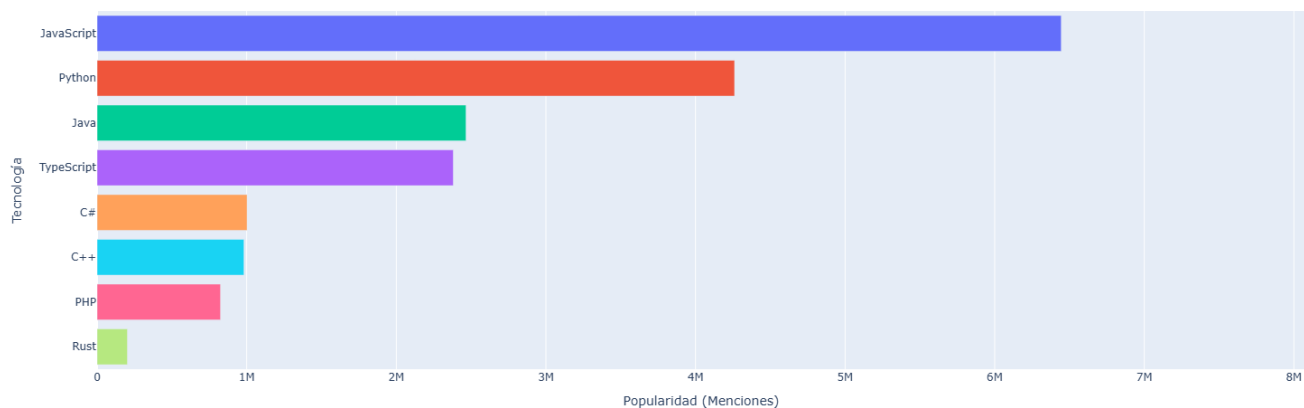


Figura 4. (Ranking anual -2024)

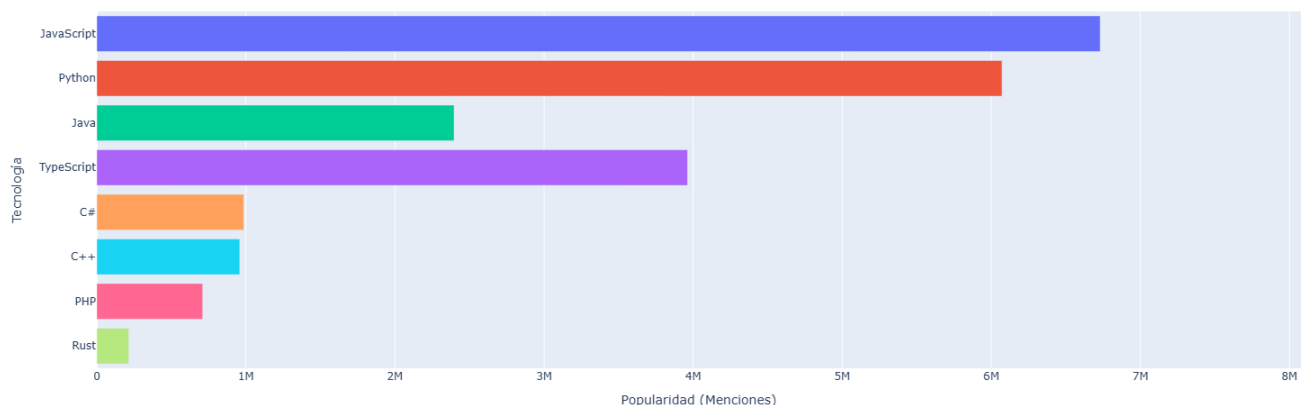


Figura 5. (Ranking anual -2025)

- Tendencias de Adopción

Se observa un crecimiento negativo generalizado en todas las tecnologías analizadas, reflejando una posible contracción en las menciones o implementaciones durante los últimos periodos. Sin embargo, este patrón no es uniforme:

- C++ emerge como la tecnología más resiliente, presentando una desaceleración menos marcada y mostrando indicios de recuperación en su curva de tendencia.
- Python y C# mantienen una base estable de adopción, con caídas moderadas que sugieren una comunidad de usuarios consolidada.
- TypeScript y JavaScript experimentan las reducciones más significativas, posiblemente debido a la aparición de alternativas emergentes o maduración del mercado.
- Rust, aunque con baja penetración relativa, evidencia un momentum positivo que podría significar una expansión futura.

- Distribución del Mercado

El panorama actual está dominado por dos actores principales:

- JavaScript (30.6%) y Python (27.6%) concentran conjuntamente el 58.2% del ecosistema, confirmando su posición hegemónica.

- TypeScript se consolida en tercer lugar con el 18%, afianzándose como estándar para desarrollo frontend contemporáneo.

- Java mantiene una presencia relevante del 10.9%, sustentada en entornos corporativos y académicos.

- El resto de tecnologías (C#, C++, PHP, Rust) representan colectivamente menos del 13%, indicando nichos de especialización o mercados más focalizados.

- **Dinámicas Evolutivas**

- JavaScript preserva su liderazgo absoluto en términos de volumen total de adopción.

- Python refuerza su posición como tecnología de referencia para inteligencia artificial, ciencia de datos y automatización.

- Java muestra estabilidad con tendencia decreciente leve, característica de tecnologías maduras y consolidadas.

- C#, C++ y PHP mantienen volúmenes constantes, asociados frecuentemente con sistemas legacy y desarrollos empresariales.

- Rust evidencia una trayectoria ascendente sostenida a pesar de su menor cuota actual.

- TypeScript continúa su proceso de maduración como evolución natural del desarrollo web frontend.

Esta distribución refleja un ecosistema en transición, donde tecnologías establecidas coexisten con alternativas emergentes, marcando diferentes etapas en el ciclo de adopción tecnológica.

Conclusiones

El análisis evidencia que el ecosistema de lenguajes de programación se encuentra en una etapa de madurez, donde JavaScript y Python dominan ampliamente el panorama, representando más de la mitad de la actividad total.

Sin embargo, el decrecimiento general del CAGR sugiere una estabilización o desaceleración en la adopción de tecnologías ya consolidadas. Esta tendencia podría deberse a la saturación del mercado o a la migración hacia nuevos paradigmas y herramientas especializadas.

Lenguajes como Rust y TypeScript destacan como alternativas emergentes con potencial de crecimiento. Rust, especialmente, se perfila como una opción prometedora en el ámbito de sistemas de alto rendimiento y desarrollo seguro.

En contraste, tecnologías como Java, C#, C++ y PHP mantienen su relevancia en nichos específicos, sosteniendo comunidades activas y roles importantes en entornos corporativos y educativos.