

Evaluación de Metodologías de Desarrollo de Software

Resumen

En este informe se desarrolla una evaluación comparativa de las metodologías de desarrollo de software **Scrum**, **Test-Driven Development (TDD)** y **Extreme Programming (XP)**. El análisis se basa en los criterios establecidos por Tinoco Gómez et al. (2010), complementados con un referente adicional académico. La finalidad es asignar un puntaje por criterio a cada metodología y determinar cuál obtiene la mayor valoración global.

1. Introducción

La selección de una metodología de desarrollo de software requiere considerar múltiples factores, desde documentación disponible hasta adopción empresarial. Para estandarizar este análisis, Tinoco Gómez et al. (2010) proponen una serie de criterios comparativos que sirven como base para evaluar Scrum, TDD y XP. En este informe se aplican dichos criterios con el fin de identificar la metodología que presenta mejores condiciones generales.

2. Referente principal

El artículo de Tinoco Gómez et al. (2010), titulado *Criterios de selección de metodologías de desarrollo de software*, establece siete criterios fundamentales: presencia en internet, documentación, certificación, entrenamiento, comunidades, presencia empresarial y proyectos de software. Estos criterios proporcionan un marco sistemático para evaluar la madurez y pertinencia de una metodología.

3. Referente adicional

Como complemento, se incluyó la tesis de Flórez Marín y Grisales Tobón (2014), la cual profundiza en factores relevantes para la selección de metodologías, tales como adaptabilidad, costo de capacitación y madurez organizacional. Este referente permite interpretar con mayor claridad el impacto práctico de los criterios evaluados.

4. Metodología de evaluación

El proceso desarrollado para esta evaluación consistió en:

1. Analizar cada criterio según las definiciones de Tinoco Gómez et al. (2010).
2. Realizar una búsqueda documental y bibliográfica sobre las metodologías evaluadas.

3. Asignar puntajes entre 1 y 5 a cada metodología por criterio.
4. Construir una tabla comparativa y obtener un puntaje total para cada metodología.

5. Evaluación de metodologías

La Tabla 1 presenta los puntajes asignados a Scrum, TDD y XP de acuerdo con los criterios establecidos.

Criterio	Scrum	TDD	XP
Presencia en internet	5	2	2
Documentación	2	5	4
Certificación	5	2	3
Training (entrenamiento)	5	5	3
Comunidades	5	4	4
Presencia empresarial	5	3	3
Proyectos de software	5	3	2
Total	32	24	21

Tabla 1: Puntajes asignados a Scrum, TDD y XP según los criterios evaluados.

6. Evidencia y justificación

Los puntajes asignados se fundamentan en datos recopilados mediante búsquedas bibliográficas, repositorios académicos y documentación oficial de cada metodología. Además, se integró evidencia proveniente de estudios académicos sobre la adopción real de Scrum, TDD y XP.

Diversas revisiones sistemáticas documentan que Scrum es ampliamente utilizado en la industria debido a su enfoque iterativo y adaptable. Khan et al. (2020) indican que Scrum se ha convertido en uno de los marcos ágiles más difundidos en proyectos con alta incertidumbre. De manera similar, Ramírez M. y Vega L. (2021) destacan que Scrum facilita la autoorganización y la mejora progresiva mediante iteraciones.

En contraste, TDD y XP presentan una adopción más limitada. Aunque Rodríguez et al. (2018) señalan los beneficios de TDD, su aplicación suele encontrarse en equipos con alta madurez técnica. XP, aunque bien documentado, enfrenta menor presencia debido a su alto nivel de disciplina y menor difusión institucional (Pérez & Álvarez, 2020).

Atendiendo a los criterios de Tinoco Gómez et al. (2010) especialmente presencia empresarial, certificación y comunidades Scrum obtiene una ventaja notable, en coherencia con lo observado por González y Muñoz (2022) respecto a su amplio ecosistema de formación y soporte.

7. Justificación de la selección de Scrum

Además de la valoración numérica, la selección de Scrum como metodología recomendada se basa en evidencia empírica, adecuación al contexto del equipo y facilidad de implementación.

Scrum ha demostrado ser eficaz en proyectos que requieren entregas incrementales, validación frecuente y flexibilidad, como argumentan Khan et al. (2020) y Ramírez M. y Vega L. (2021). Su estructura ligera y adaptable favorece el trabajo en equipos pequeños, facilitando organización y seguimiento.

Asimismo, su ecosistema maduro con certificaciones, guías oficiales, herramientas y comunidades activas reduce la curva de aprendizaje y permite una implementación estructurada, como señala González y Muñoz (2022).

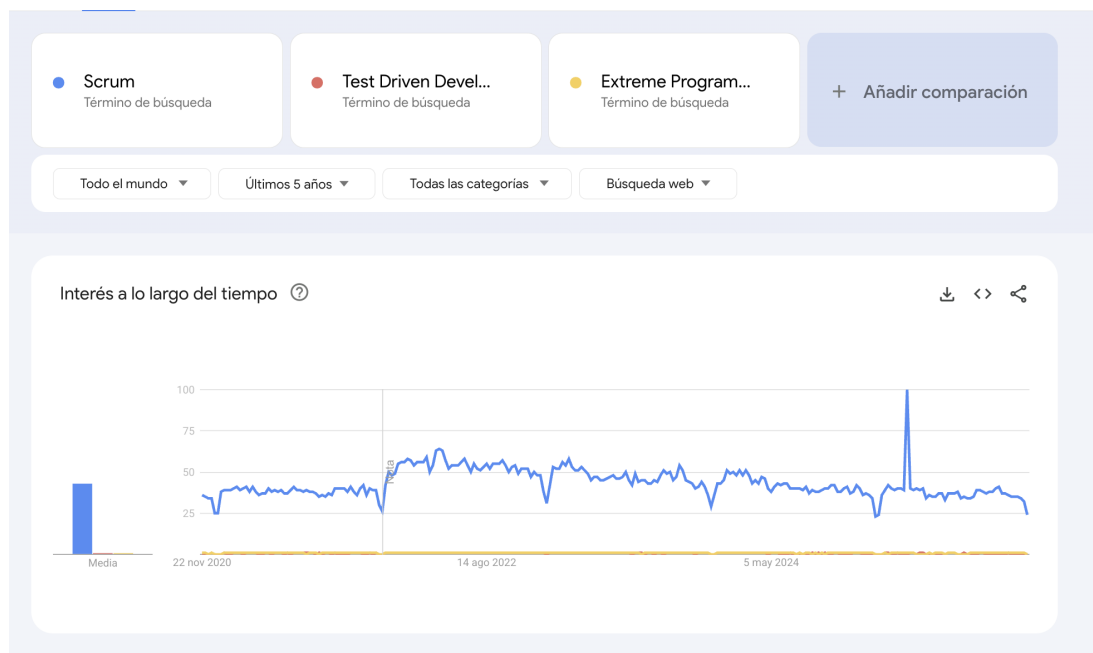
Referencias

- Flórez Marín, L., & Grisales Tobón, F. (2014). *Formulación de criterios para la selección de metodologías de desarrollo de software*.
- González, A., & Muñoz, F. (2022). SCRUM: Un enfoque práctico de metodología ágil para la ingeniería de software. *Tecnología, Investigación y Academia*, 10(2), 45-60. <https://revistas.udistrital.edu.co/index.php/tia/article/view/15702>
- Khan, M. W., Basri, S., & Dom, M. (2020). A Systematic Review of SCRUM in Software Development. *Journal of Informatics and Innovative Ventures (JOIV)*, 2(1), 1-12. <https://joiv.org/index.php/joiv/article/view/167>
- Pérez, D., & Álvarez, C. (2020). Mejora de procesos de software en equipos Scrum: Una revisión de la literatura. *Ingeniare. Revista Chilena de Ingeniería*, 28(2), 259-270. <https://ingeniare.uta.cl/index.php/inge/article/view/1867>
- Ramírez M., S., & Vega L., J. (2021). Revisión Sistemática de la Metodología Scrum para el Desarrollo de Software. *Dominio de las Ciencias*, 7(2), 434-457. <https://dominiodelasciencias.com/ojs/index.php/es/article/view/2429>
- Rodríguez, J., Soria, A., & Campo, M. (2018). Factors Models of Scrum Adoption in the Software Development Process: A Systematic Literature Review. *Revista Facultad de Ingeniería*, 27(49), 1-17. <https://www.redalyc.org/journal/5122/512255650002>
- Tinoco Gómez, O., Rosales López, P. P., & Salas Bacalla, J. (2010). Criterios de selección de metodologías de desarrollo de software. *Industrial Data*, 13(1), 70-74.

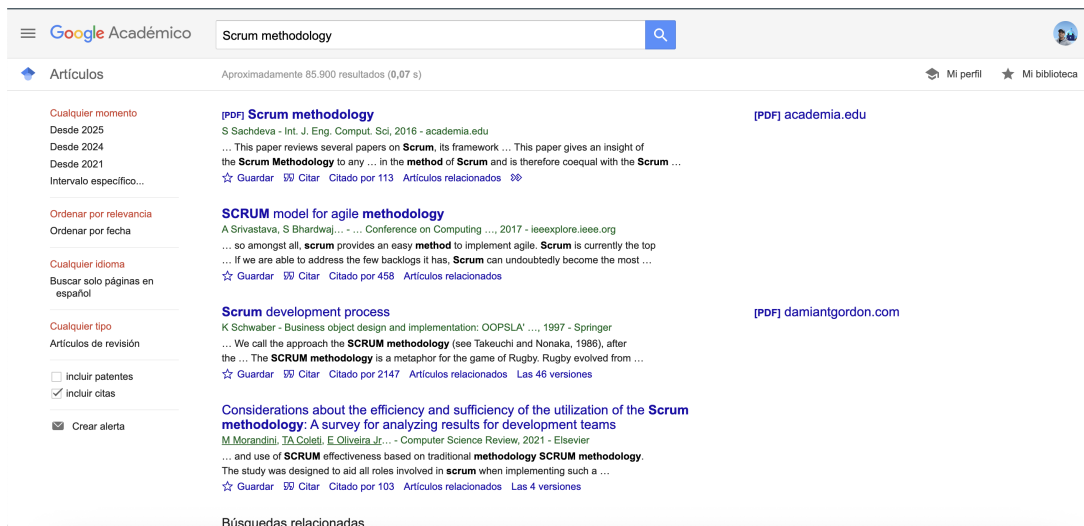
Anexos

The screenshot shows the Google Académico search results for the query "Extreme Programming". The search bar at the top displays the query and a magnifying glass icon. Below the search bar, the results are organized into a list of articles. On the left side, there are filters for "Artículos", "Cualquier momento" (with sub-options: Desde 2025, Desde 2024, Desde 2021, Intervalo específico...), "Ordenar por relevancia" and "Ordenar por fecha", "Cualquier idioma" (with sub-option: Buscar solo páginas en español), "Cualquier tipo" (with sub-option: Artículos de revisión), and checkboxes for "incluir patentes" and "incluir citas". The main list of results includes three articles: 1. "Reglas y prácticas en eXtreme Programming" by J. J. Joskowicz, Universidad de Vigo, 2008, from academia.edu. 2. "Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)" by P. Letellier, MC. Penadés, 2006, from cyta.com.ar. 3. "Desarrollo de software ágil: Extreme Programming y Scrum" by J. R. Fuentes, 2015, from books.google.com. Each article entry includes a brief description, citation information, and links to "Guardar", "Citar", "Citado por", "Artículos relacionados", and "Las 3 versiones". A fourth article, "A systematic review on extreme programming" by A. Shrivastava, I. Jaggi, N. Katoh, 2021, from iopscience.iop.org, is also visible. At the bottom, there is a section for "Búsquedas relacionadas".

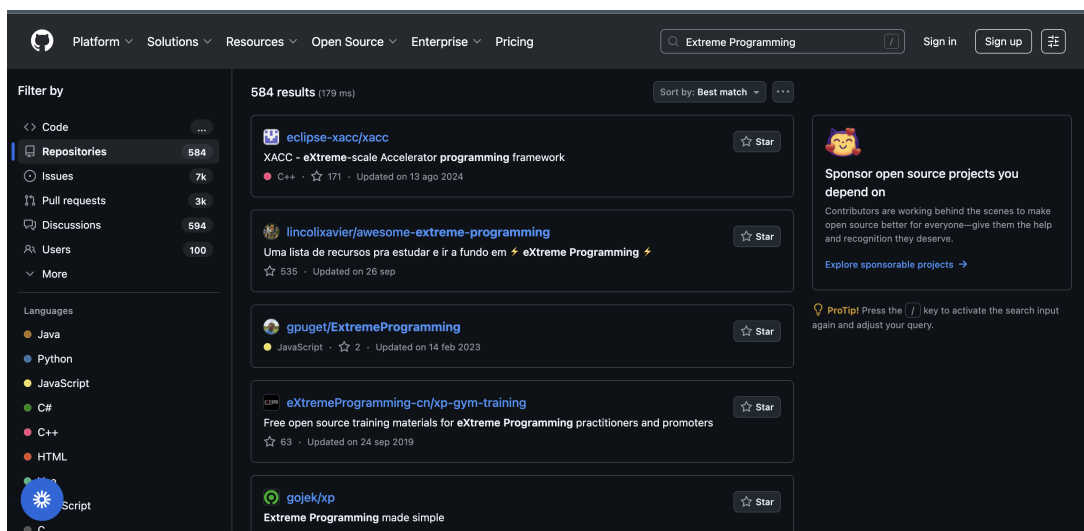
Evidencia 1: Evidencia 1.



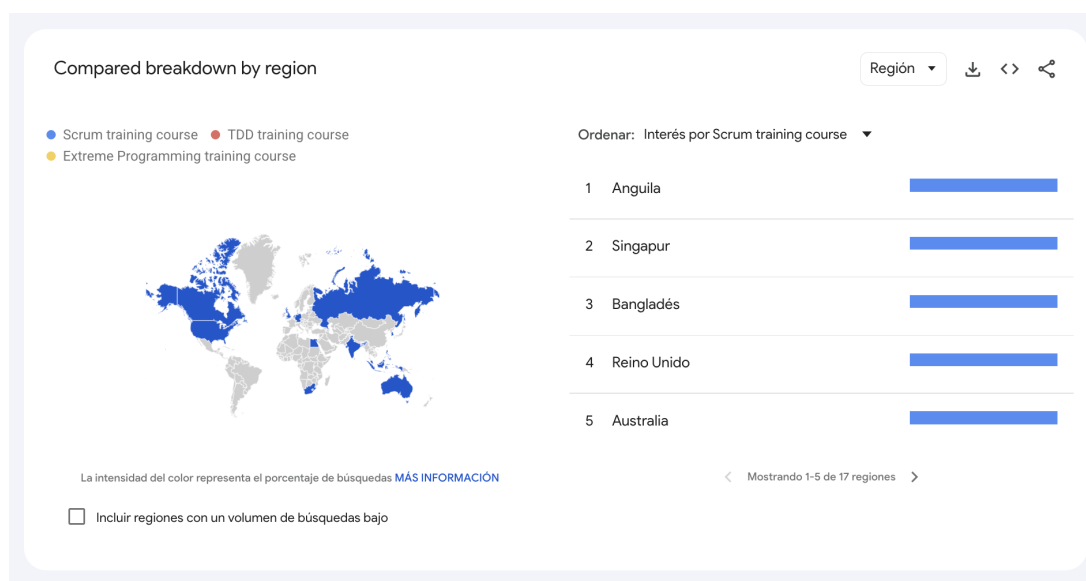
Evidencia 2: Evidencia 2.



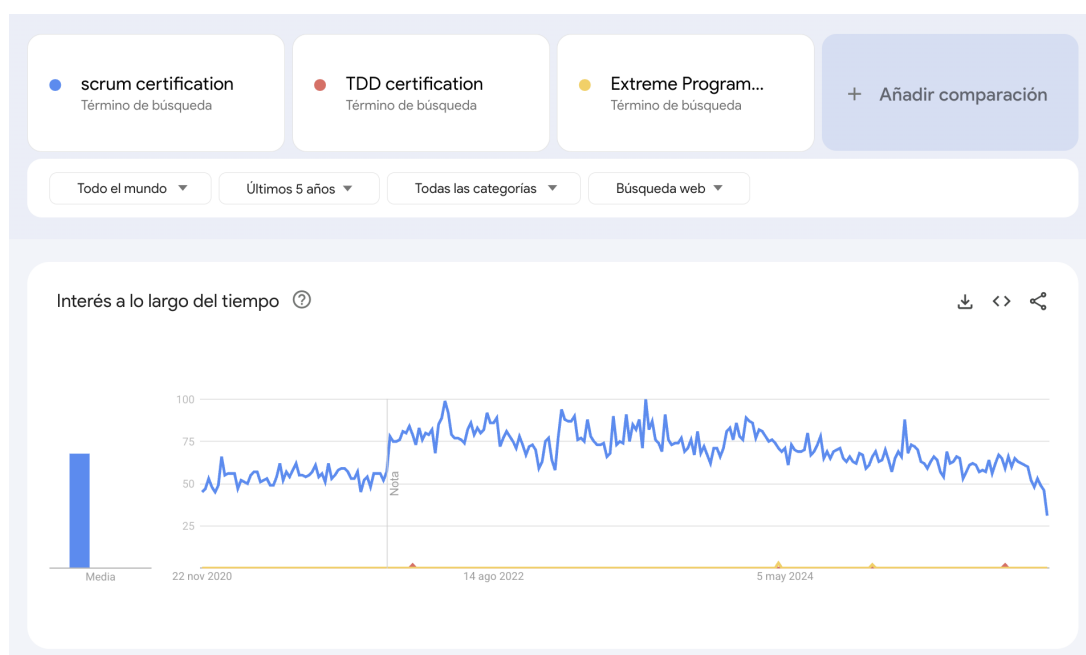
Evidencia 3: Evidencia 3.



Evidencia 4: Evidencia 4.



Evidencia 5: Evidencia 5.



Evidencia 6: Evidencia 6.