

# M6 Calidad de los servicios informáticos LFIN A

# Proyecto modular integrador

|  |  |
| --- | --- |
| **Tutor:** | Virginia Araguz Lara |
| **Estudiante:** | José Ramón Ibáñez Posadas |
| **Matricula:** | BNL098377 |

|  |  |
| --- | --- |
| San Nicolás de los Garza, Nuevo León | Viernes, 4 de Julio de 2025 |

# Introducción

En la era digital, la calidad del software es un pilar fundamental para garantizar sistemas eficientes, seguros y accesibles. WordPress, como uno de los CMS más utilizados globalmente, sirve como caso de estudio ideal para evaluar los estándares de calidad en desarrollo de software.

Este proyecto analizará su alineación con la norma **ISO 25000**, examinando atributos como usabilidad, seguridad y rendimiento.

Además, se investigará su ciclo de vida de desarrollo y cómo este impacto en su producto final. La finalidad es identificar áreas de mejora aplicando modelos como **CMMI** o **COMPETISOFT**, así como métricas de calidad clave.

Con este análisis, se demostrará la importancia de gestionar procesos de desarrollo con enfoque en la mejora continua, asegurando entregar software que cumpla con las expectativas del usuario y los estándares internacionales.

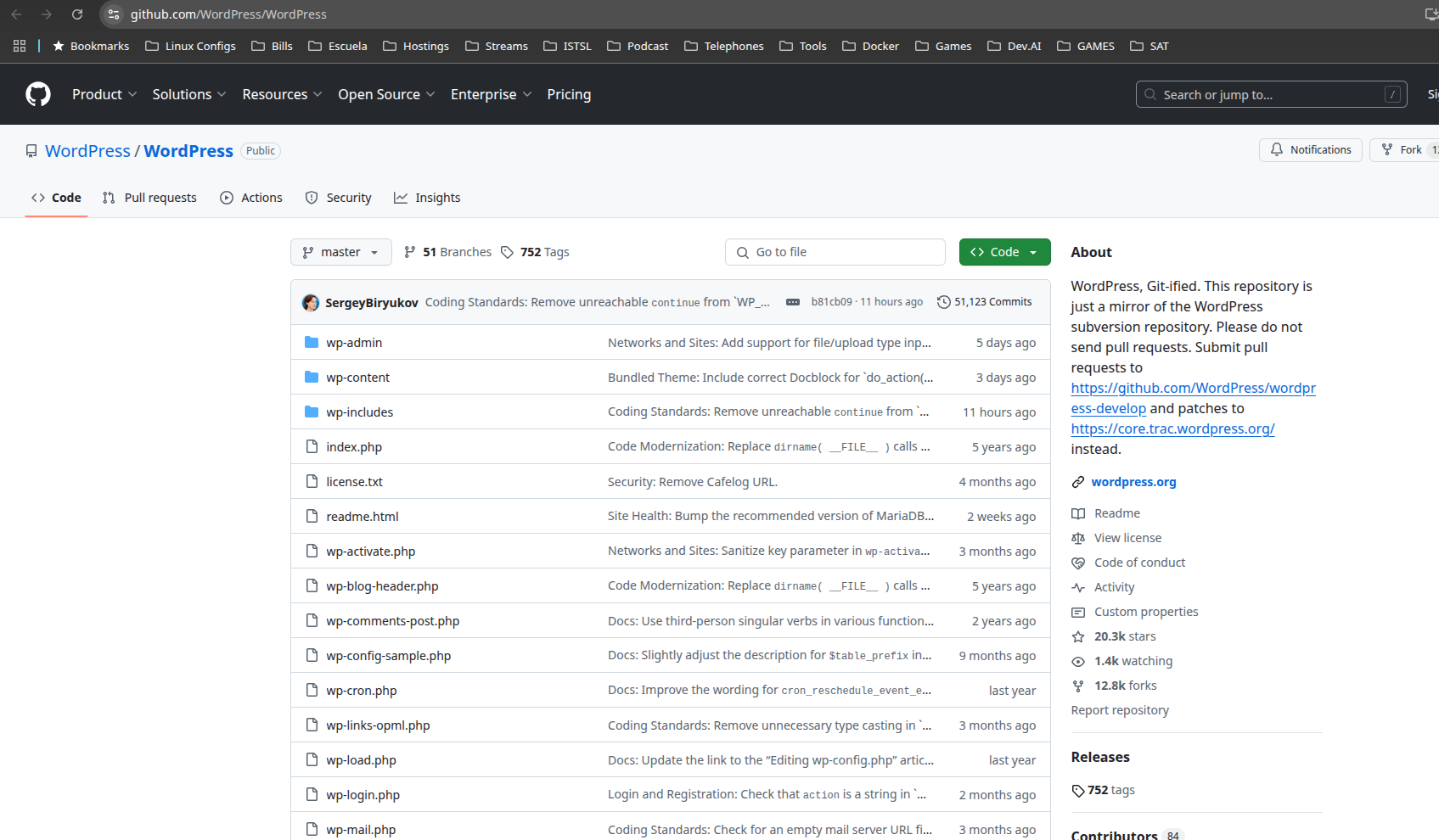
# Desarrollo

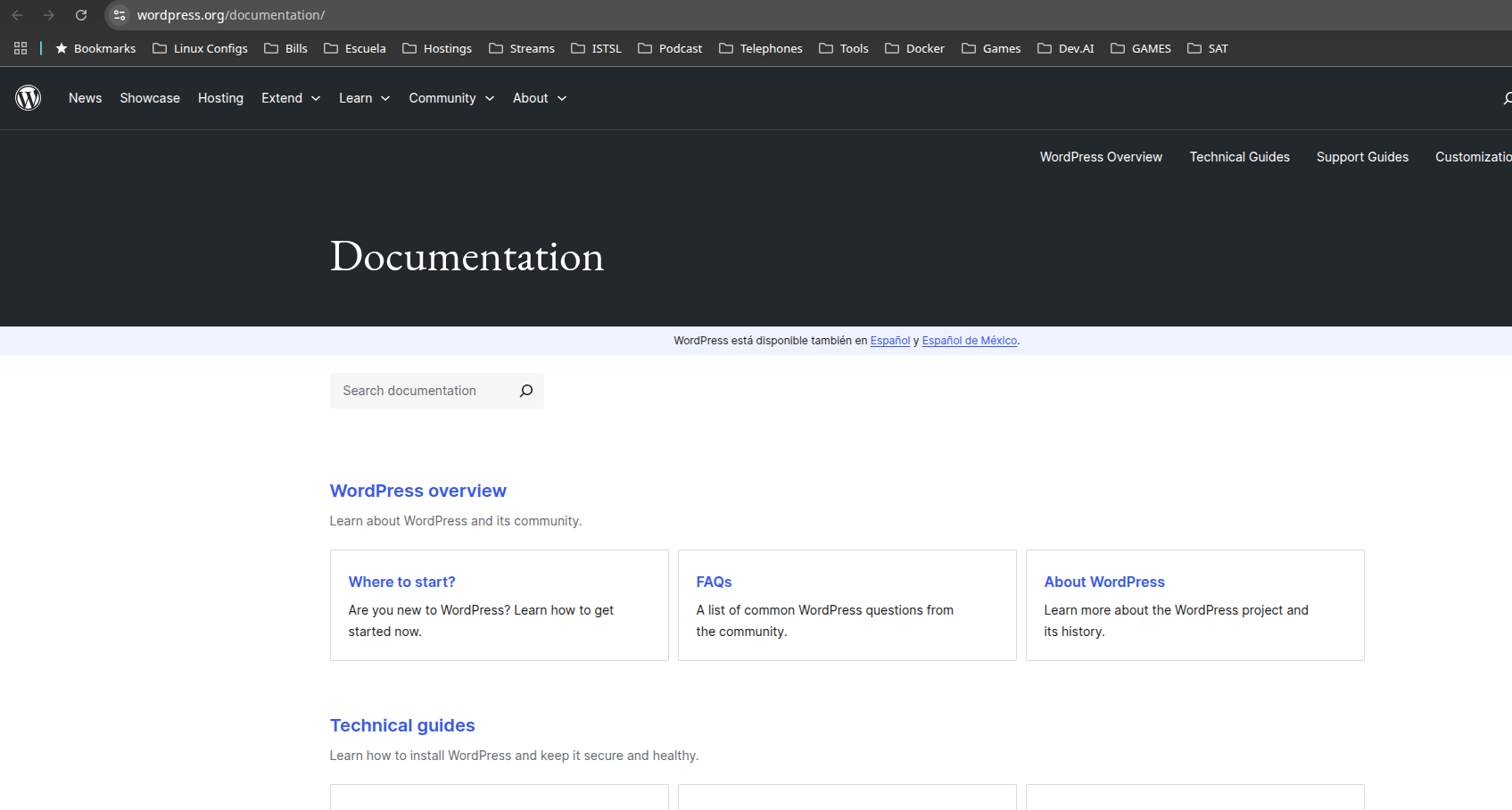
## Análisis de WordPress

### Cumplimiento de estandares de calidad (ISO 25000)

WordPress no esta certiicado oficialmente bajo ISO 25000, pero muchos de sus atributos de calidad cumplen con los criterios de ISO 25010.

* **Funcionalidad**
  + WordPress cumple con su propósito principal: permitir la creación y gestión de sitios web sin necesidad de programación avanzada.
  + Soporta plugins y temas que amplían sus funcionalidades.
* **Usabilidad**
  + Su interfaz es intuitiva, con un editor visual (Gutenberg) que facilita la creación de contenido.
  + Tiene una gran comunidad de soporte (foros, documentación oficial).
* **Eficiencia**
  + Puede ser lento si no se optimiza correctamente (requiere caching y optimización de bases de datos).
  + Consumo de recursos varía según los plugins instalados.
* **Mantenibilidad**
  + Actualizaciones frecuentes de seguridad y funcionalidad.
  + Código abierto (GitHub) permite contribuciones de la comunidad.
* **Seguridad**
  + Vulnerabilidades comunes en plugins mal desarrollados.
  + Requiere configuración adicional (firewalls, HTTPS) para mayor protección.

Figura : Repositorio oficial de WordPress - Alojado en GitHub

Figura : Sitio oficial de WordPress - Documentación oficial

### Ciclo de vida del software

WordPress sigue un modelo de desarrollo ágil (Scrum) con iteraciones continuas:

* Lanzamientos frecuentes (versiones menores cada mes, mayores cada 3-6 meses).
* Retroalimentación constante de la comunidad.
* Pruebas automatizadas (PHPUnit, integración continua).

Impacto en la calidad:

* Mejora continua gracias a actualizaciones rápidas.
* Mayor adaptabilidad a nuevas tecnologías.
* Riesgo de introducir errores en actualizaciones apresuradas.

### **Propuestas de mejora en la gestión del desarrollo**

* **Implementar más pruebas de seguridad automatizadas para plugins (ej: integración con SonarQube).**
* **Mejorar la documentación para desarrolladores (estándares de codificación más claros).**
* **Adoptar métricas de calidad más estrictas (ej: reducir deuda técnica con herramientas como CodeClimate).**
* **Si WordPress fuera una PYME, aplicar COMPETISOFT para estandarizar procesos de desarrollo.**

# ****Enlace al vídeo****

<https://youtu.be/sdhhul5B9d8>

# Conclusión

El análisis de WordPress permitió comprender cómo un software de código abierto puede mantener altos estándares de calidad mediante comunidad activa y metodologías ágiles.

Si bien no está certificado bajo ISO 25000, cumple con atributos clave como funcionalidad y mantenibilidad, aunque requiere mejoras en seguridad y optimización.

Su ciclo de vida ágil facilita adaptabilidad, pero también exige rigurosidad en pruebas. Las propuestas presentadas, como integración de métricas automatizadas y adopción de modelos como COMPETISOFT, buscan fortalecer su gestión de calidad.

Este proyecto refuerza que la evaluación constante, el cumplimiento de normas y la retroalimentación del usuario son esenciales para desarrollar software confiable.

WordPress, como caso de éxito, demuestra que la calidad no es un destino, sino un proceso de mejora continua.

# Bibliografía

