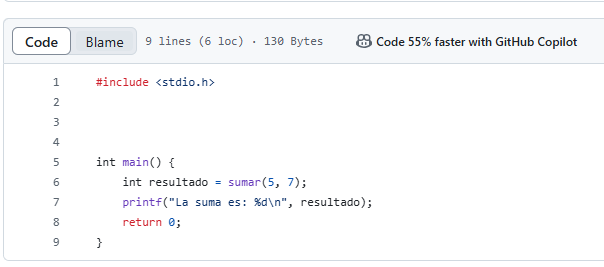
| **ID** | **Requisito Funcional** |
| --- | --- |
| R1 | Monitorización en tiempo real de amenazas. |
| R2 | Análisis automático de vulnerabilidades. |
| R3 | Generación automática de alertas. |
| R4 | Ejecución automática de respuestas a incidentes. |
| R5 | Pruebas automatizadas en cada integración. |
| R6 | Adaptabilidad a amenazas emergentes. |
| R7 | Reducción del tiempo de respuesta a incidentes. |
| ID | Historia de Usuario |
| HU1 | Como analista de seguridad, quiero ver alertas en tiempo real para reaccionar rápidamente ante incidentes. |
| HU2 | Como experto en redes, quiero analizar las vulnerabilidades detectadas automáticamente para priorizar su solución. |
| HU3 | Como desarrollador, quiero que el sistema ejecute respuestas automáticas a amenazas para minimizar el daño. |
| HU4 | Como QA, quiero que se ejecuten pruebas automatizadas al integrar código nuevo para asegurar calidad continua. |
| HU5 | Como responsable de ciberseguridad, quiero adaptar las reglas del sistema a nuevas amenazas para mantener la efectividad del software. |
| HU6 | Como stakeholder, quiero que el tiempo de respuesta ante incidentes críticos sea menor a 5 minutos para evitar pérdidas mayores. |

| **Historia de Usuario** | **Prioridad MoSCoW** | **Justificación** |
| --- | --- | --- |

|  |  |  |  |
| --- | --- | --- | --- |
| HU1 | Alertas en tiempo real | Obligatorio | Fundamental para detectar amenazas de inmediato |
| HU3 | Respuesta automática | Obligatorio | Reduce impacto y acelera la reacción |
| HU6 | Tiempo de respuesta | Obligatorio | Meta clave del proyecto. |
| HU2 | Análisis de vulnerabilidades | Importante | Ayuda a prevenir futuros ataques, pero no es inmediato |
| HU4 | Pruebas automatizadas | Importante | Mejora la calidad del software en el tiempo. |
| HU5 | Adaptación a nuevas amenazas | Deseable | Útil a largo plazo, pero se puede implementar después. |

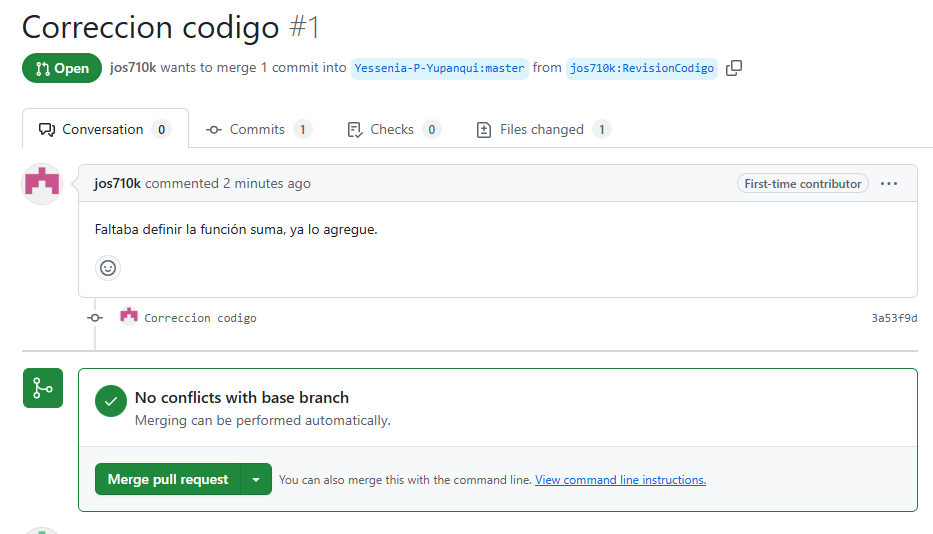
TDD:

1. **Red (Fallo)**: Escribir una prueba que falla (el código aún no existe).

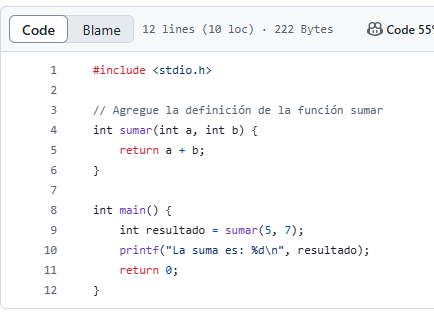


La función suma no está definida

1. **Green (Éxito)**: Escribir el **mínimo código necesario** para que la prueba pase.



Pull requests



Código sin error

1. **Refactor**: El código es simple, así que no se necesita refactor por ahora.

**INFORME DE RETROSPECTIVA:**

**Aprendizajes**

1. **Colaboración en Pair Programming:**Trabajar en pareja permitió identificar errores de forma más temprana y optimizar el desarrollo del código.
2. **Integración Continua y TDD:**  
   La adopción de Test Driven Development (TDD) y la integración continua contribuyó a mantener la estabilidad del sistema durante todo el sprint. Las pruebas unitarias realizadas antes de escribir el código facilitaron la detección de errores y ayudaron a asegurar que las nuevas funcionalidades no afectaran las existentes.
3. **Priorización Eficaz:**Se logró priorizar correctamente las historias de usuario, enfocándose en las funcionalidades más críticas, como la monitorización en tiempo real de amenazas y el análisis automático de vulnerabilidades, que son fundamentales para el éxito del sistema de ciberseguridad.

**Dificultades**

1. estimación de tiempos, algunas tareas pueden resultar más complejas de lo previsto, lo que impacta en el cumplimiento de ciertos plazos.
2. dificultades en la sincronización de horarios entre los miembros del equipo,

**Mejora a implementar:**

1. Mejor estimación y planificación: Incorporar sesiones de planificación más detalladas y usar métricas de sprints anteriores para estimar con mayor precisión.
2. Comunicación más estructurada: Establecer reuniones cortas y regulares (dailys) y canales de comunicación más efectivos.
3. Documentación continua: Implementar el hábito de documentar procesos y decisiones clave durante el desarrollo del proyecto.

* Inicio del Proyecto: Documentar los objetivos, expectativas, roles y metodología a seguir.
* Reuniones de Planificación: Registrar los puntos acordados, las tareas asignadas, los plazos establecidos y los riesgos identificados.
* Cambios Importantes en el Proyecto: Cualquier desviación en el alcance, los plazos o las prioridades debe quedar registrada con una justificación clara.