

Informe de Trabajo del Equipo

Nombre del Proyecto	TDD-Practica
Nombre del Equipo	Tidelock
Fecha de entrega	15/05/2025

1. Análisis del Trabajo

Interpretación del proyecto: Desarrollamos un gestor de credenciales seguro siguiendo la metodología S-TDD. Implementamos funciones básicas (añadir, verificar, eliminar y listar credenciales) asegurando desde el principio requisitos de seguridad como autenticación, cifrado, validación de entradas y políticas de contraseñas robustas.

Aportaciones originales:

- Aplicamos S-TDD estrictamente: primero los tests, luego el código
- Usamos Design-by-Contract con **icontract** para reforzar pre/postcondiciones
- Incorporamos fuzzing con Hypothesis para probar la robustez de la política de contraseñas
- Añadimos tests de seguridad para escenarios como acceso concurrente y manipulación de datos
- Nos centramos en protección contra inyección y almacenamiento seguro con bcrypt
- Implementamos logging para auditoría sin exponer datos sensibles

2. Seguimiento y Herramientas de Gestión

Herramientas de planificación y coordinación: Planificamos con reuniones de equipo, nos comunicamos por Discord y WhatsApp, y usamos GitHub para control de versiones y colaboración.

Frecuencia de revisión: Revisábamos el progreso diariamente por chat y semanalmente en reuniones estructuradas. Los commits y pull requests nos permitieron seguir el desarrollo continuamente.

3. Contribuciones Individuales

Ignacio Ferrer

Tarea	Creación de tests funcionales y de seguridad
Colaboradores	Ninguno
Tiempo empleado	30 minutos
Entregables	tests/gestor_credenciales/
Notas	Creó todos los tests del proyecto

Gonzalo Ramos

Tarea	Implementación del código para pasar los tests
-------	--

Tarea	Implementación del código para pasar los tests
Colaboradores	Cayetano López
Tiempo empleado	1 hora
Entregables	src/gestor_credenciales/gestor_credenciales.py
Notas	Desarrolló la clase GestorCredenciales

Jorge Varea

Tarea	Implementación de contratos con icontract
Colaboradores	Ninguno
Tiempo empleado	45 minutos
Entregables	src/gestor_credenciales/gestor_credenciales.py
Notas	Definió pre y postcondiciones con icontract

Cayetano López

Tarea	Revisión y refactorización del código
Colaboradores	Ninguno
Tiempo empleado	45 minutos
Entregables	src/gestor_credenciales/gestor_credenciales.py, tests/
Notas	Revisó el código para cumplir todos los requisitos

4. Evaluación del Trabajo en Equipo

¿Qué ha funcionado bien y qué se puede mejorar?

El equipo funcionó bien gracias a una clara distribución de tareas. La comunicación por Discord y WhatsApp resolvió dudas rápidamente, y GitHub facilitó la integración del trabajo.

Lo que funcionó:

- La especialización de cada miembro permitió avanzar en paralelo
- Las revisiones cruzadas detectaron problemas temprano
- Cumplimos con los plazos establecidos

Podemos mejorar:

- La documentación del código desde el principio
- Implementar un sistema más formal de revisión de código
- Llegar a consensos más rápidos sobre decisiones técnicas

5. Evaluación de la experiencia con S-TDD

¿Qué os ha parecido útil y qué se puede mejorar?

S-TDD fue clave para desarrollar software seguro desde su concepción. Al escribir primero los tests de seguridad, consideramos estos aspectos antes de escribir código, resultando en un diseño inherentemente más seguro.

Aspectos positivos:

- La seguridad se integró desde el inicio, no como añadido posterior
- El fuzzing con Hypothesis descubrió casos extremos difíciles de identificar
- Los contratos con icontract añadieron verificación en tiempo de ejecución
- Logramos casi 100% de cobertura de pruebas

Consideraciones:

- Requiere más trabajo inicial, pero compensa con menos vulnerabilidades
- Tiene una curva de aprendizaje pronunciada para quienes no conocen seguridad
- Exige un cambio cultural en equipos acostumbrados a métodos tradicionales

En conclusión, S-TDD mejoró significativamente la seguridad de nuestro gestor. Aunque requiere más tiempo inicial, es ideal para proyectos donde la seguridad es crítica.