Ejercicio Práctico 1: Testing Top Down Documentación Análisis de ventajas

Estudiante Daniel Alejandro Arevalo Guecha

Docente
HELY SUAREZ MARIN

Codificación y Pruebas de Software
Ingeniería de Software
Fundación de Estudios Superiores Comfanorte
FESC

ANÁLISIS DE VENTAJAS

Confiabilidad desde la base

Al probar primero los módulos más pequeños (cálculo de impuestos, bonos y deducciones), se garantiza que las partes críticas del sistema funcionan correctamente antes de integrarlas. Esto reduce la posibilidad de errores grandes en etapas avanzadas.

Pruebas inmediatas

Como los módulos básicos no dependen de otros, se pueden probar tan pronto son desarrollados. Esto permite detectar y corregir errores rápidamente sin tener que esperar a que esté listo todo el sistema.

Menor dependencia de stubs

A diferencia del enfoque Top-Down, aquí no es necesario crear "simuladores" de módulos inferiores. Los módulos reales ya están disponibles y se integran directamente con drivers sencillos.

Integración progresiva y controlada

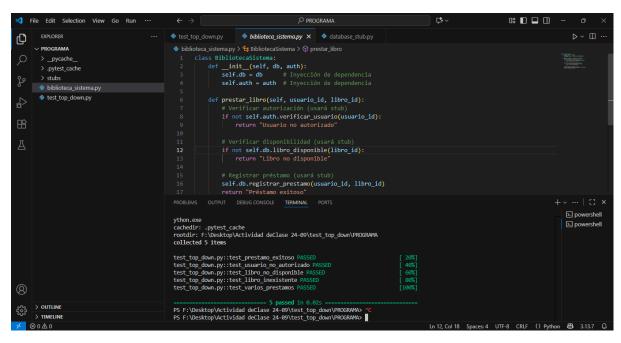
El sistema se construye de abajo hacia arriba, lo que permite verificar en cada nivel que la integración sea correcta y que los cálculos de la nómina coincidan con lo esperado.

Reutilización de componentes

Los módulos atómicos, al estar bien probados, se pueden reutilizar en otros sistemas o versiones futuras sin necesidad de volver a validarlos en detalle.

CAPTURAS

RATIO DE PRUEBAS



PRUFBAS

```
def test_libro_no_disponible():
def test_prestamo_exitoso():
   db_stub = DatabaseStub()
                                                                           db_stub = DatabaseStub()
   auth_stub = AuthStub()
                                                                           auth_stub = AuthStub()
   sistema = BibliotecaSistema(db_stub, auth_stub)
                                                                           sistema = BibliotecaSistema(db stub, auth stub)
   resultado = sistema.prestar_libro(usuario_id=1, libro_id=2)
                                                                           resultado = sistema.prestar_libro(usuario_id=1, libro_id=3)
                                                                           assert resultado == "Libro no disponible
   assert resultado == "Préstamo exitoso"
                                                                           assert len(db_stub.prestamos_registrados) == 0
   assert (1, 2) in db_stub.prestamos_registrados
                                                                      def test_libro_inexistente():
def test usuario no autorizado():
                                                                           db_stub = DatabaseStub()
   db_stub = DatabaseStub()
                                                                           auth_stub = AuthStub()
   auth_stub = AuthStub()
                                                                           sistema = BibliotecaSistema(db_stub, auth_stub)
   sistema = BibliotecaSistema(db_stub, auth_stub)
                                                                           resultado = sistema.prestar_libro(usuario_id=1, libro_id=99)
   resultado = sistema.prestar_libro(usuario_id=0, libro_id=2)
                                                                           assert resultado == "Libro no disponible
   assert resultado == "Usuario no autorizado
                                                                           assert len(db_stub.prestamos_registrados) == 0
   assert len(db_stub.prestamos_registrados) == 0
def test_varios_prestamos():
   db_stub = DatabaseStub()
    sistema = BibliotecaSistema(db_stub, auth_stub)
   resultado1 = sistema.prestar_libro(usuario_id=1, libro_id=1)
   resultado2 = sistema.prestar_libro(usuario_id=2, libro_id=2) resultado3 = sistema.prestar_libro(usuario_id=3, libro_id=3)
    assert resultado1 == "Préstamo exitoso"
    assert resultado2 == "Préstamo exitoso"
    assert resultado3 == "Libro no disponible"
    assert (1, 1) in db_stub.prestamos_registrados
    assert (2, 2) in db_stub.prestamos_registrados
    assert len(db_stub.prestamos_registrados) == 2
```