



Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

TESTING

Raúl Montes T.

Test

Prueba que permite validar una hipótesis

Resultado binario - success / fail

Estructura: Given - When - Then

¿Para qué?

Asegurar código correcto

Documentación

Previene futuros errores

Tipos de tests

Unitarios: modelos, helpers

Funcionales: controllers, views

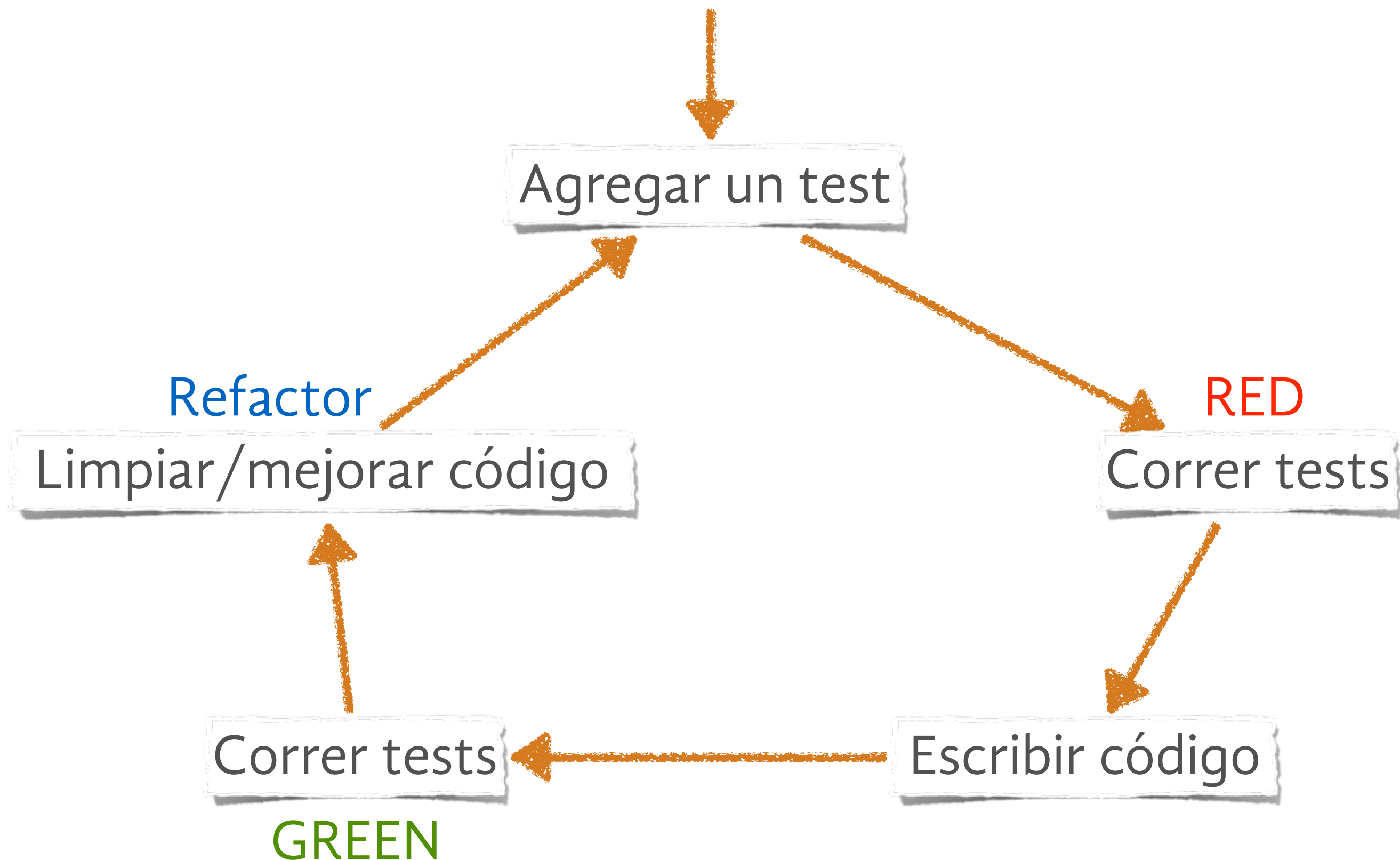
Integración

TDD

~~Tratemos de Desarrollar Decentemente~~

TDD

Test Driven Development



¿Ventajas?

¿Desventajas?

Behavior Driven Development

Foco en comportamiento deseado

Comunicación entre desarrolladores, QA, producto

Niveles de testing

Unit tests

Functional tests

Integration tests

Elementos necesarios

Test runner

Assertions/expectations

Mocks/Spies

Coverage

Diferentes opciones

mocha + chai + sinon + proxyquire + istambul

jest

otras...

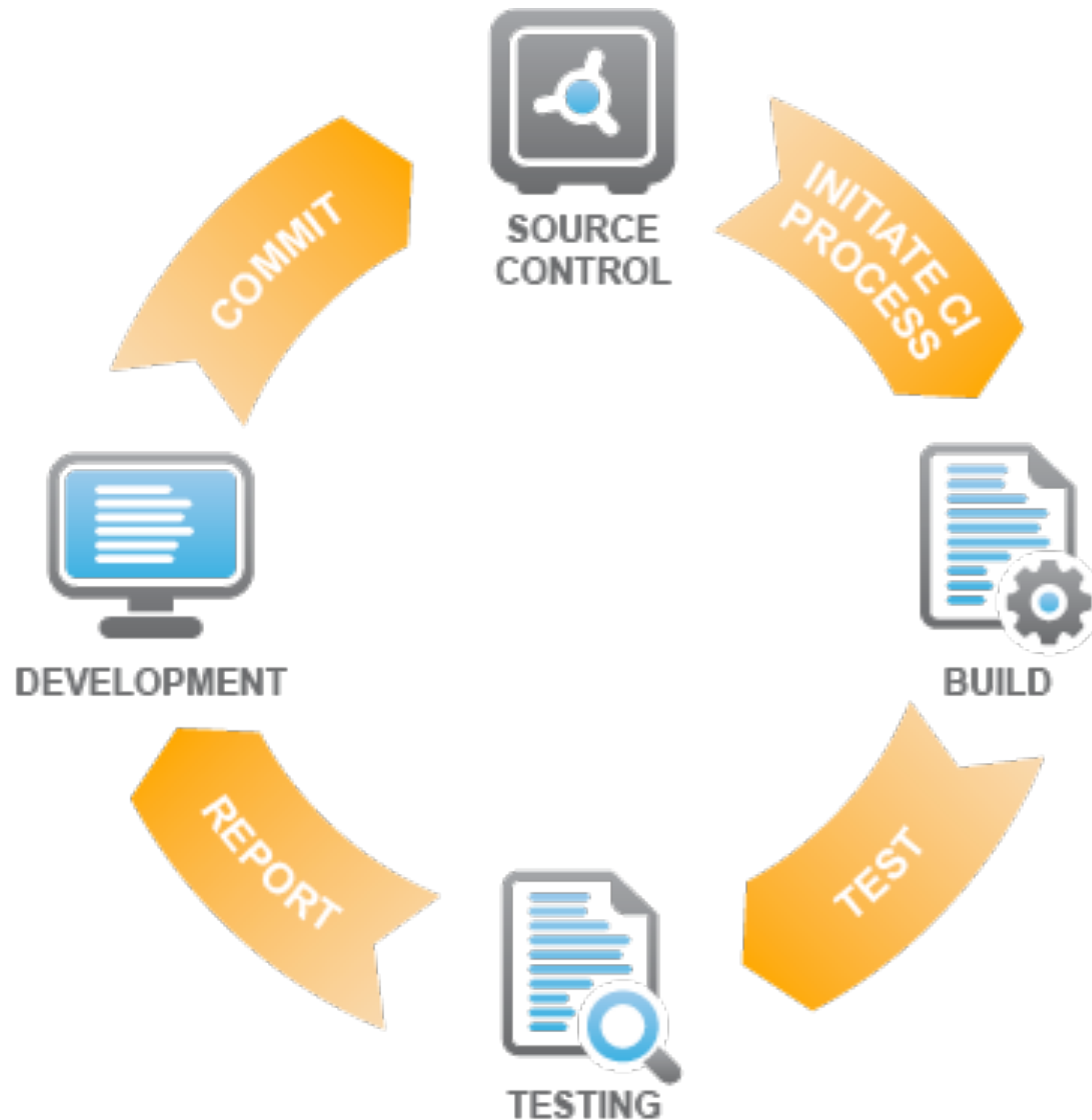
Jest

```
describe('Model', () => {
  let instance
  beforeEach(() => {
    instance = new Model();
  });
  describe('#instanceMethod', () => {
    context('when given some context', () => {
      beforeEach(() => {
        // set the context
      });
      it('returns object with propX with value 5', () => {
        expect(instance.instanceMethod()).toMatchObject({ propX: 5 });
      })
    });
    context('when given some other context', () => {
      // ...
    });
  });
});
```

Recomendaciones

- Identificar bien el *subject*
- El test siempre se refiere al *subject* más próximo
- Definir contextos complementarios
- Definir claramente las *expectations*
- Una o muy pocas *expectations* por test

Continuous Integration



Continuous Integration

Circle CI

Jenkins

Travis CI

Snap CI

Semaphore CI