

FRAMEWORK PARA PRUEBAS UNITARIAS

UNIDAD 4: PRUEBAS DE SOFTWARE



Temario

- Elementos del framework para pruebas unitarias
- Automatización de pruebas unitarias

Elementos del framework para pruebas unitarias

JUNIT: FRAMEWORK FOR UNIT TESTING

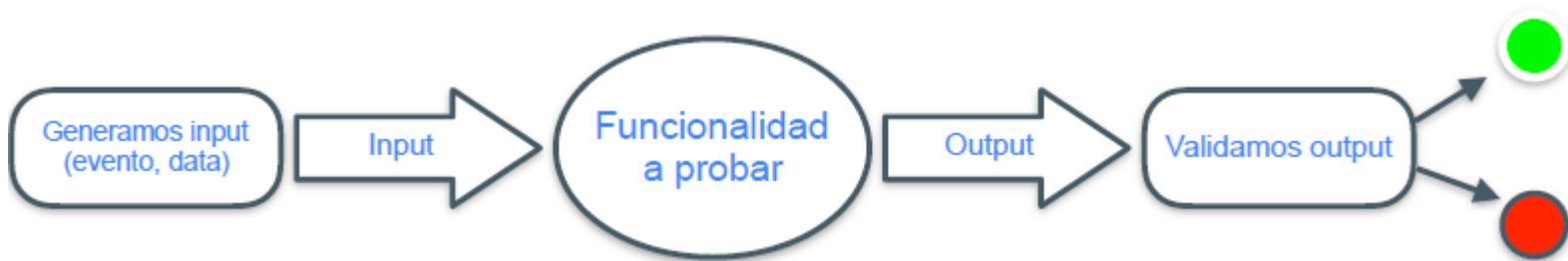
The logo for JUnit 4, featuring the word "JUnit" in a bold, sans-serif font. The "J" is green, and the "Unit" is red. The number "4" is in a dark gray, bold, sans-serif font. The entire logo is set against a light gray rectangular background.

JUnit 4

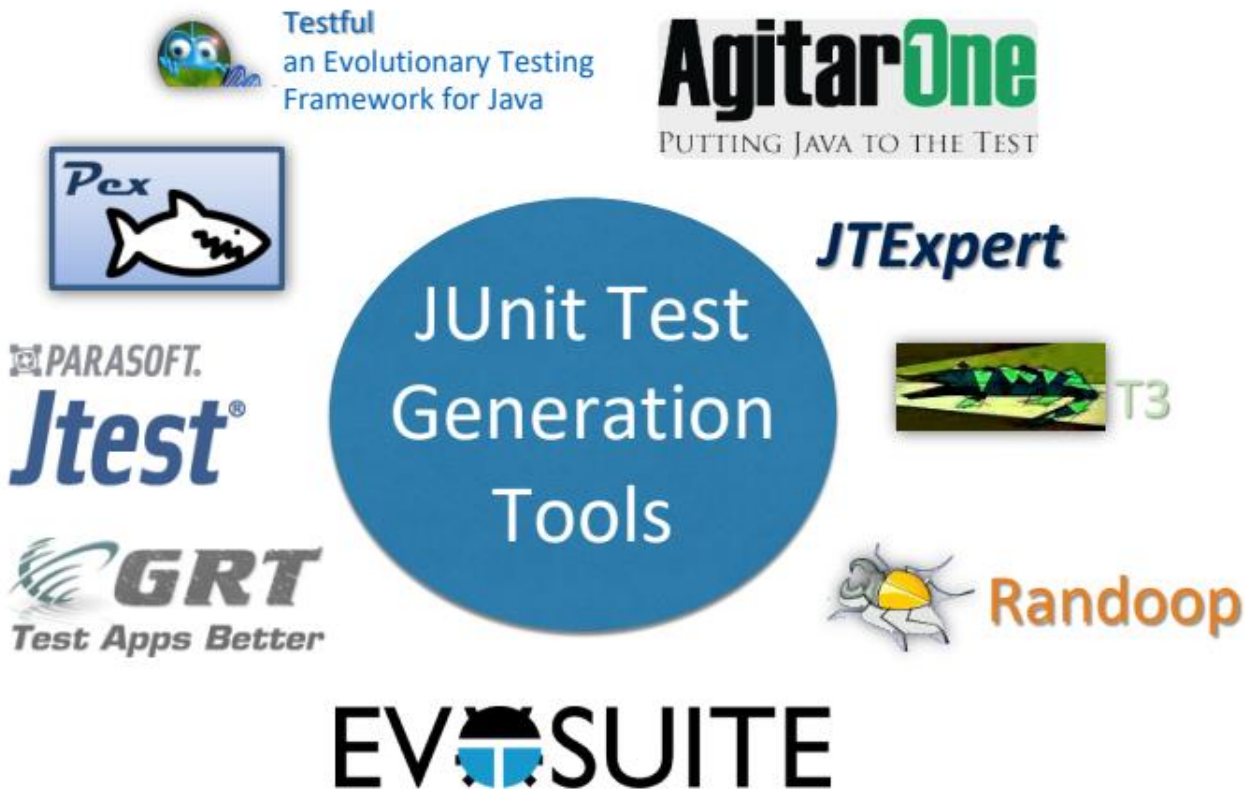
Automatización de pruebas unitarias

Test Automáticos

- La idea es poder simular la interacción de un usuario con el software implementado.
- Sigue el flujo básico de cualquier prueba de test automático:



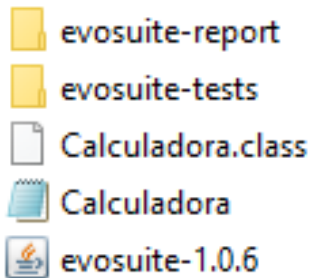
Generación de pruebas automáticas



Generación de pruebas automáticas

```
public class Calculadora{  
    public double suma(double a, double b){  
        return a+b;  
    }  
    public double divide(double a, double b){  
        if (b==0){  
            throw new ArithmeticException("No es posible la division entre cero");  
        }  
        return a/b;  
    }  
}
```

java -jar evosuite-1.0.6.jar -projectCP ./ -class Calculadora



Generación de pruebas automáticas

```
public class Calculadora_ESTest extends Calculadora_ESTest_scaffolding {
```

```
    @Test(timeout = 4000)
```

```
    public void testSumaReturningZero() throws Throwable {
```

```
        Calculadora calculadora0 = new Calculadora();
```

```
        double double0 = calculadora0.suma(0.0, 0.0);
```

```
        assertEquals(0.0, double0, 0.01);
```

```
    }
```

```
    @Test(timeout = 4000)
```

```
    public void testSumaReturningPositive() throws Throwable {
```

```
        Calculadora calculadora0 = new Calculadora();
```

```
        double double0 = calculadora0.suma(0.0, 2366.63002468);
```

```
        assertEquals(2366.63002468, double0, 0.01);
```

```
    }
```

```
java -jar evosuite-1.0.6.jar -projectCP ./ -class Calculadora -Dtest_naming_strategy=COVERAGE
```



UNIVERSIDAD
DE LIMA

Referencias

- Proceso de Construcción de Software 2, Mag. Natalí Flores Lafosse, Maestría en Informática de la PUCP, 2018.
- Sommerville, I. (2005). Ingeniería del software. Pearson educación.
- Bourque, P., & Fairley, R. E. (2014). Guide to the software engineering body of knowledge (SWEBOK (R)): Version 3.0. IEEE Computer Society Press.