FLAKY TESTS

SAMUEL BUSTAMANTE LARRIET – UO289689

MARCO QUINTANA GARCÍA - UO287872

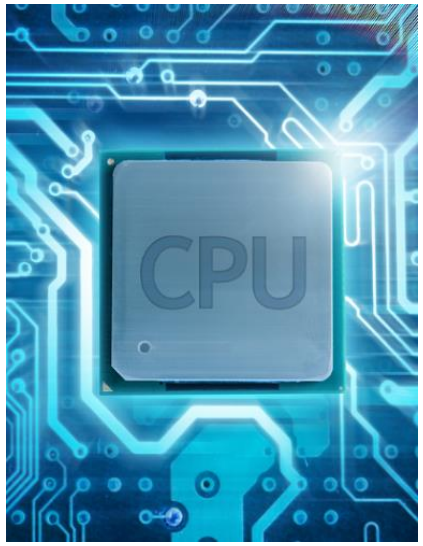
TERESA GONZÁLEZ GARCÍA - UO288347

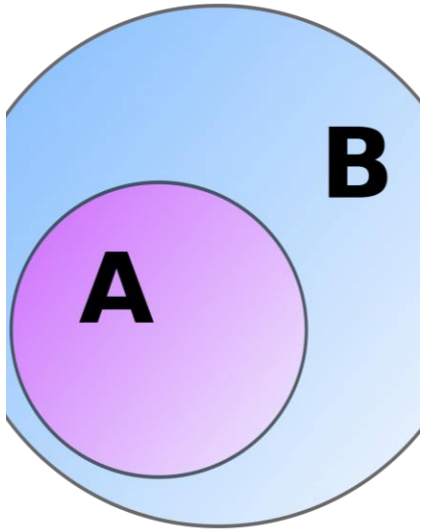
ÍNDICE

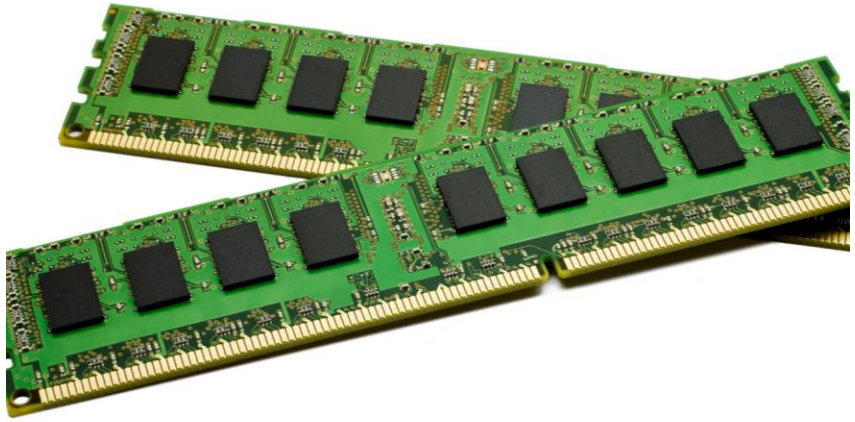
1.1 Introducción: ¿Qué es un Flaky test?

Caso de prueba que puede pasar unas veces y fallar otras sin modificar el código fuente.

```
//PR13. Loguearse como estudiante y ver los detalles de la nota con Descripcion = Nota A4.
@Test
@Order(15)
public void PR13() {
    //Iniciamos sesión como estudiante y comprobamos inicio correcto
    PO_PrivateView.login(driver,dni:"99999990A",password:"123456", rol:"estudiante");
    //Contamos las notas
    By enlace = By.xpath("//td[contains(text(), 'Nota A4')]/following-sibling::*[2]");
    driver.findElement(enlace).click();
    //Esperamos por la ventana de detalle
    String checkText = "Detalles de la nota";
    List<WebElement> result = PO_View.checkElementBy(driver, type:"text", checkText);
    Assertions.assertEquals(checkText, result.get(0).getText());
    //Ahora nos desconectamos comprobamas que aparece el menu de registrarse
    PO_PrivateView.logout(driver);
}
```







1.2 Introducción: Tipos de Flaky test

- Order-dependent flaky test
  Dependientes del orden en que se ejecutan.
- Non-order-dependent flaky test:
  Independientes del orden en que se ejecutan. Podrían estar relacionados con:
  - Problemas de temporización
  - Problemas de asincronía
  - Problemas de red
  - Problemas relacionados con la CPU, la memoria o el sistema de archivos

2.1 Cómo detectarlos: Procedimiento a seguir

1. Ejecutar todos los casos de prueba.
2. Si falla, ejecutar ese caso de prueba de manera aislada.
3. Si funciona, ejecutar gradualmente de forma conjunta.
4. Variar el contexto en el que se ejecutó.

Recomendable: ejecutar las pruebas de forma aleatoria

2.2 Cómo detectarlos: Herramientas

- Contenedores Docker y máquinas virtuales
  - Agregan restricciones a diferentes recursos.
- Frameworks
  - Adjuntan marcadores a los casos de prueba.
- Surefire
  - Plugin para Maven.
  - Reejecuta tests al detectar inestabilidad.

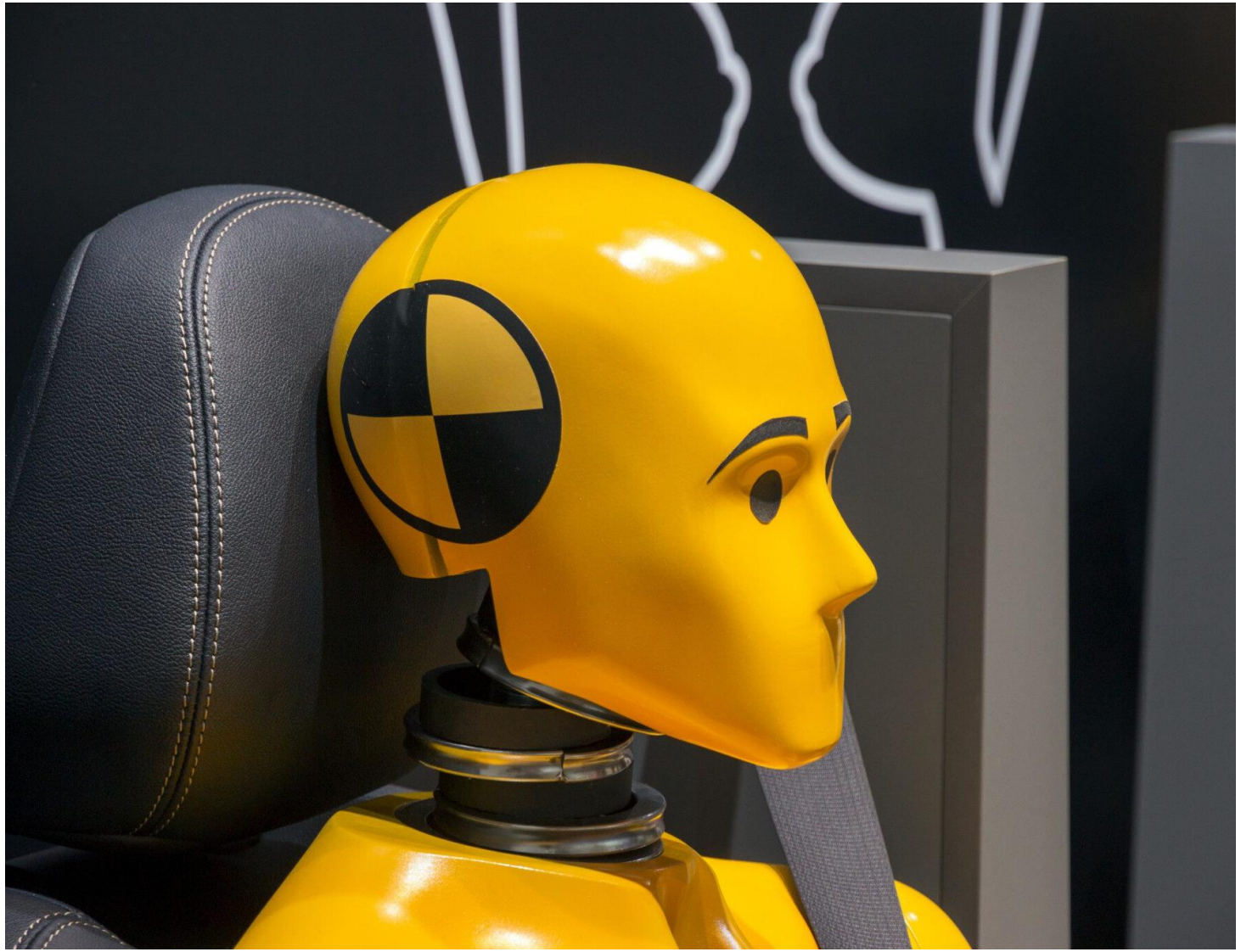
2.2 Cómo detectarlos: Herramientas.

- Herramientas de Machine Learning
  - Información sobre el programa y la Test Suite
    - Características estáticas: AST, estructura sintáctica, etc.
    - Características dinámicas: sobrecarga de la memoria, información extra sobre el código, etc.
  - Más información => aproximación más precisa.
- PyTest
  - Ejecuta los tests de forma aleatoria

3.1 Cómo solucionarlos: Como abordar el problema.

Investigar cómo interactúa el caso de prueba con el entorno de ejecución:

- Observar cómo se ejecuta el código.
- Medir el tiempo de ejecución.
- Rastrear el uso de memoria.
- Cómo interactúa el programa con el sistema de archivos.
- Herramienta => Datadog



3.2 Cómo solucionarlos: ¿Qué pasa si la estabilidad del test está fuera de nuestro control?

- Uso frecuente de servicios de terceros
- Soluciones
  - Mock objects
  - Aislar el test

3.3 Cómo solucionarlos. ¿Es rentable invertir tiempo en arreglar tests inestables?

- Tres opciones
  - Eliminar el test
  - Intentar arreglarlo
  - Aislarlo

4.1 Evitar Flaky tests: Buenas prácticas

- Utilizar eventos.
  - Aporta estabilidad y rendimiento
  - Problema del evento inexistente
- Priorizar existencia de un elemento y no su localización
  - Aporta estabilidad
- Casos de prueba simples
  - Evitar bucles y condicionales
  - Pocos asertos
- Componentes y funciones con responsabilidad única

4.2 Evitar Flaky tests: Métodos setUp y tearDown

- Se ejecutan antes y después de cada test
- Dotan de estabilidad a las pruebas
- Pueden producir problemas de rendimiento

```
Marco Quintana
@BeforeEach
public void setUp() { driver.navigate().to(URL); }

// Despues de cada prueba se borran las cookies del navegador
Marco Quintana
@AfterEach
public void tearDown() { driver.manage().deleteAllCookies(); }
```

FLAKY TESTS

SAMUEL BUSTAMANTE LARRIET – UO289689

MARCO QUINTANA GARCÍA - UO287872

TERESA GONZÁLEZ GARCÍA - UO288347