

```
<!--Grupo 106-->
```

# Flaky Tests {

```
<ul>
```

```
<li>Miguel Méndez Murias</li>
```

```
<li>Martín Cancio Barrera</li>
```

```
<li>Pablo García-Ovies Pérez</li>
```

```
<li>Marcos Barril Villaverde</li>
```

```
</ul>
```

}



# Contenidos

- 01 ¿Qué son?
- 02 Problemas
- 03 Tipos y causas
- 04 Cómo Detectarlos
- 05 Cómo Evitarlos

¿Que es? {

```
<h1>¿Qué es un Flaky Test?</h1>
```

}

Problemas {

- Deterioro en la Calidad de las Pruebas
- Desconfianza en el Conjunto de Pruebas
- Falsa Sensación de Seguridad
- Desperdicio de Tiempo y Recursos
- Repercusiones en la Integración y Despliegue Continuos
- Impacto Adverso en la Productividad

}

# Tipos {

## Dependientes del orden;

Fallan de manera intermitente dependiendo del orden en el que se ejecuten.

**Ejemplo:** Herramienta de evaluación automatizada.

## No dependientes del orden;

Dependen de factores ajenos del orden de ejecución.

**Ejemplo:** Problemas de zona horaria en pruebas de logística.

}

Causas {

- Código no determinista
- Errores de asincronía
- Problemas de concurrencia
- Problemas de red
- Recursos del sistema
- Pruebas deficientes

}

Causas: Código no determinista {

```
@Test
public void methodThatUsesNow() {
    String fixedTime = "2022-01-01T12:00:00Z";
    Clock clock = Clock.fixed(Instant.parse(fixedTime), ZoneId.of("UTC"));

    // Ahora el cronómetro usa una fecha conocida
    Instant now = Instant.now(clock);

    // El resto del test...
}
```

}

## Causas: Errores de asincronía {

```
function someAsyncFunction(myObject) {  
  // cuerpo del método  
  // ....  
  
  // valor de retorno por referencia  
  myObject.return_value = "some string";  
}  
  
// esto produce inestabilidad  
let testObject = {};  
someAsyncFunction(testObject);  
assertEqual(testObject.return_value == "some string");
```

}

Causas: Problemas de concurrencia {

```
function testAccountTransfer(fromAccount, toAccount) {  
  lockFrom=fromAccount.lock()  
  lockTo=toAccount.lock()  
  
  beforeBalanceFrom = getBalance(fromAccount)  
  beforeBalanceTo = getBalance(toAccount)  
  
  transfer(fromAccount,toAccount,100)  
  
  assert(beforeBalanceFrom - getBalance(fromAccount) == 100)  
  assert(getBalance(toAccount) - beforeBalanceTo == 100)  
  
  lockTo.release()  
  lockFrom.release()  
}
```

}

Causas: Problemas de red {

```
async function testNetworkIssues(){  
  var data = await fetch("https://www.example.com/data");  
  
  expected = "Some string"  
  assert(data.prop === expected);  
}
```

}

Causas: Recursos del sistema {

- CPU
- Memoria
- Filesystem

}

Causas: Pruebas deficientes {

```
@AfterEach
public void tearDown() {
    friendsRepository.deleteAll();
    recommendationsRepository.deleteAll();
    postsRepository.deleteAll();
    usersRepository.deleteAll();

    insertSampleDataService.init();
}
```

}

# Cómo detectarlos {

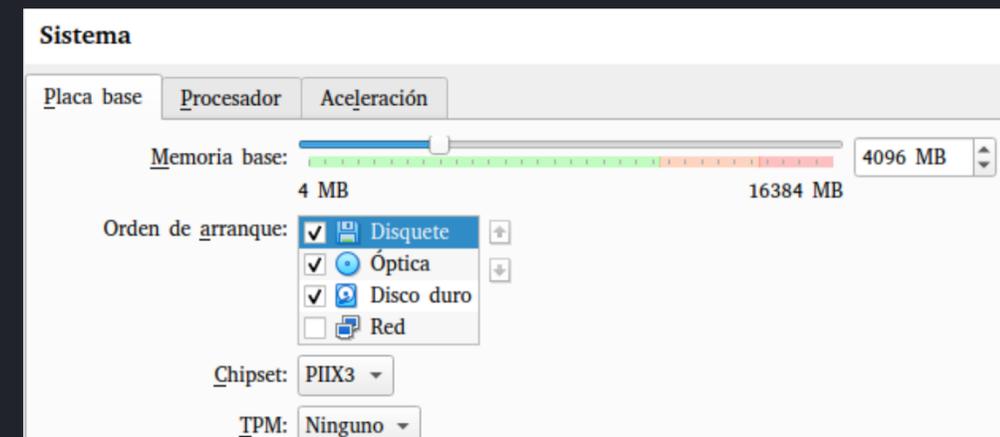
## Ejecución repetida

### De forma aislada

- Uso de contenedores Docker o MV
- Distintas configuraciones
  - Tiempo
  - Memoria

### En conjunto

- Uso de distintas configuraciones
- Problemas en el orden
- Problemas en el contexto



}

# Como detectarlos {

## Uso de herramientas

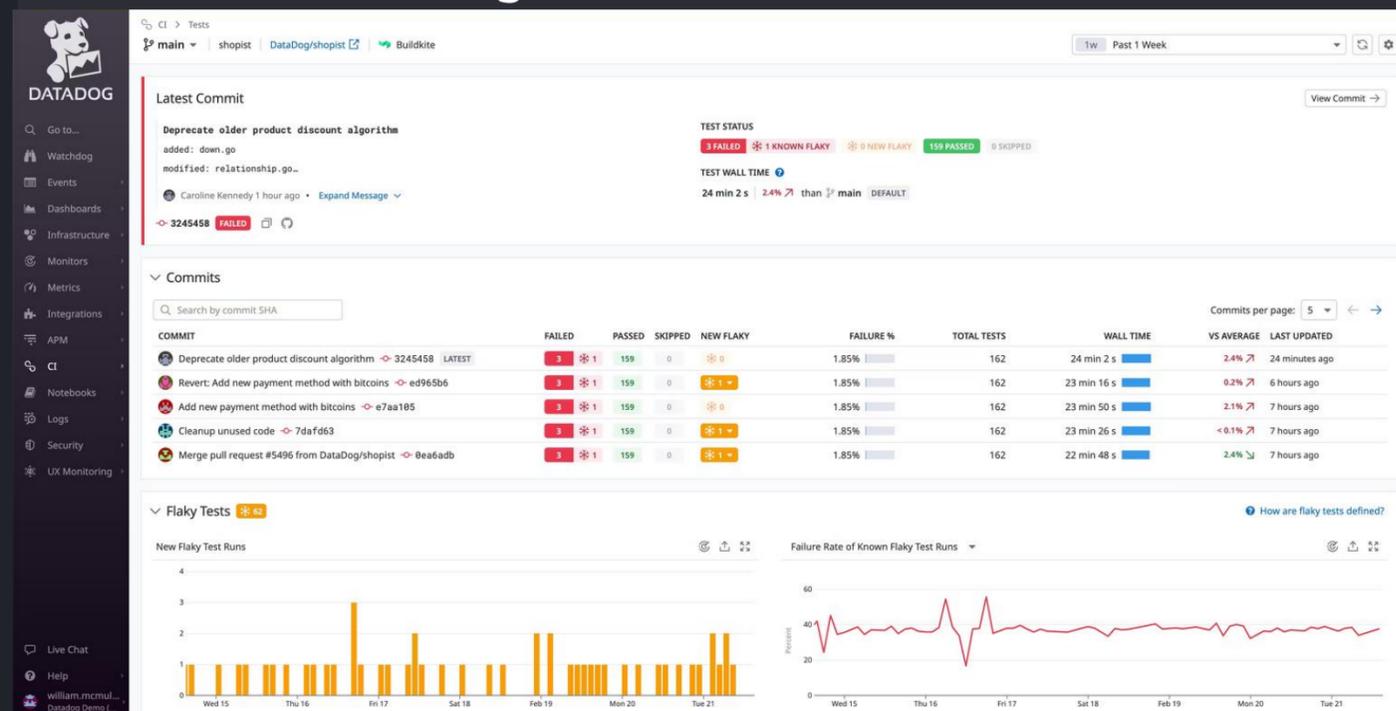
### Herramientas tradicionales

- De ejecución repetida
  - Surefire plugin
- De monitorización
  - Datadog

### Herramientas de machine learning

Obtienen y analizan información

- Estática
  - Relacionado con el árbol AST del programa o los tests
- Dinámica
  - Uso del sistema de archivos
  - Uso de memoria
  - ...



Evitarlos {

¿Cómo evitar los Flaky Tests?

}

## Evitarlos I: Corrección {

```
1- function testNetworkIssues() {  
2-     var data = fetch("https://example.com/data");  
3-  
4-     expected = "Some string"  
5-     assert(data.prop === expected);  
6- }
```

}

## Evitarlos II: Prevención {

- Mocks (estabilidad > realismo)
- Independencia de la UI
- Cubrir la latencia
- Simplicidad en los casos de pruebas:
  - Evitar bucles y lógica condicional
  - Reducir número de asserts

}

```
<!--Grupo 106-->
```

# Gracias {

```
<form>
```

```
  <label for="pregunta">Preguntas: <label>
```

```
  <input id="pregunta" type="text"/>
```

```
</form>
```

# }