

Testing con Microservicios

Quarkus-junit5

Notas

- Soporte a CDI
- Testing HTTP simplificado
- Mocking simplificado
- Servicios containerizados
- Requisito: implementar el Surefire maven plugin, el log manager.

```
import javax.inject.Inject;
import org.junit.jupiter.api.Assertions;
import org.junit.jupiter.api.Test;
import io.quarkus.test.junit.QuarkusTest;

@QuarkusTest ❶
public class MyEngineTest {

    @Inject
    Engine engine; ❷

    @Test ❸
    public void testEngine() {
        // Given
        engine.stop();

        // When
        engine.start();

        // Then
        Assertions.assertTrue(engine.isRunning()); ❹
    }
}
```

Nota: tambien puede usar Hamcrest, AssertJ

DEV SERVICES

quarkus.datasource.db-kind=postgresql **1**

%**test**.quarkus.datasource.db-kind=h2 **2**

Recursos

<https://junit.org/junit5/>

<https://github.com/rest-assured/rest-assured>

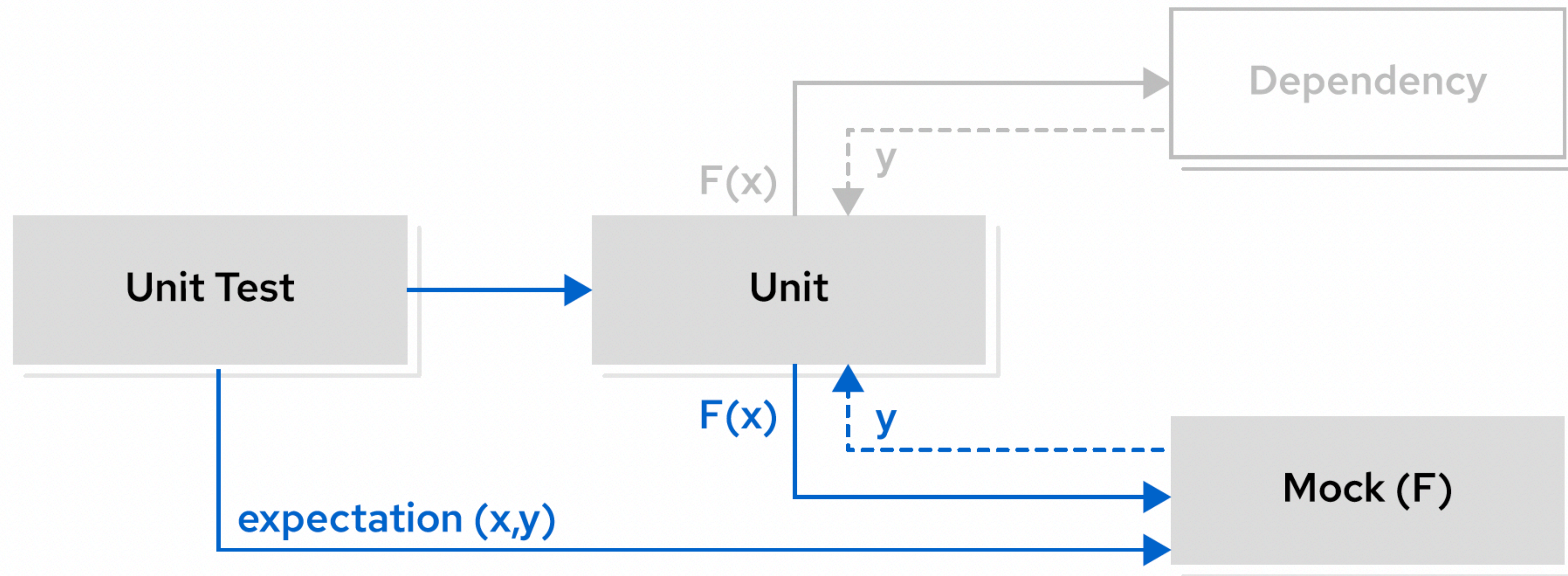


Figure 3.1: Mock type example


```
import io.quarkus.test.Mock;

import javax.enterprise.context.ApplicationScoped;

@Mock ❶
@ApplicationScoped ❷
public class OrderServiceMock extends OrderService {
    @Override
    public Order getOrder(UUID uuid) {
        return new Order(uuid, "Customer Name", 123);
    }
}
```



```
@QuarkusTest
public class OrderServiceTest {

    @Inject ❶
    OrderService orderService;

    @Test
    public void testExample() {
        UUID uuid = ...code omitted...

        Order order = orderService.getOrder(uuid); ❷

        boolean isFraud = orderService.isFraud(order); ❸

        ...code omitted...
    }
}
```



```
@QuarkusTest
public class OrderServiceTest {
    @Test
    public void testExample() {
        UUID uuid = ...code omitted...

        OrderService orderService = Mockito.mock(OrderService.class); ❶

        Mockito
            .when(orderService.getOrder(uuid)) ❷
            .thenReturn(new Order(uuid, "Customer Name", 123)); ❸

        Mockito
            .when(orderService.getOrder(null)) ❹
            .thenThrow(new NullPointerException()); ❺

        Order order = orderService.getOrder(uuid); ❻

        ...code omitted...

        Order order = orderService.getOrder(null); ❼

        ...code omitted...
    }
}
```



```
@QuarkusTest
public class OrderServiceTest {
    @Test
    public void testExample() {
        UUID uuid = ...code omitted...

        OrderService orderService = Mockito.mock(OrderService.class);

        Mockito
            .when(orderService.getOrder(Mockito.any())) ❶
            .thenReturn(new Order(uuid, "Customer Name", 123));

        Order order = orderService.getOrder(uuid); ❷
        Order order = orderService.getOrder(null); ❸

        ...code omitted...
    }
}
```



```

@QuarkusTest
public class SomeServiceTest {

    @InjectMock ❶
    OrderService orderService;

    @BeforeEach
    public void setup() {
        Mockito
            .when(orderService.getOrder(null))
            .thenThrow(new NullPointerException()); ❷
    }

    @Test
    public void testExample() {
        UUID uuid = ...code omitted...

        Mockito
            .when(orderService.getOrder(Mockito.any()))
            .thenReturn(new Order(uuid, "Customer Name", 123)); ❸

        Data data = orderService.getOrder(uuid);

        ...output omitted...
    }
}

```

@InjectMock

Testing Indirect Interactions

```
@QuarkusTest
public class SpyTest {
    @InjectMock
    OrderService orderService;

    @Test
    public void testExample() {
        given()
            .when()
                .get("/orders")
            .then()
                .statusCode(200)
                .body("$.size()", is(0));

        Mockito.verify(orderService, Mockito.times(1)).list(); ❶
        Mockito.verify(expenseService).list(); ❷
        Mockito.verifyNoMoreInteractions(orderService, expenseService); ❸
    }
}
```


Mocking de Panache Entities

```
@QuarkusTest
public class SomeResourceTest {

    @BeforeAll
    public static void setup() {
        PanacheMock.mock(Order.class); ❶
    }

    @Test
    public void testExample() {
        Mockito
            .when(Order.listAll()) ❷
            .thenReturn(Collections.emptyList()); ❸

        given() ❹
            .when()
                .get("/orders")
            .then()
                .statusCode(200)
                .body("$.size()", is(0));
    }
}
```

```
/dependency/
<dependency>
  <groupId>io.quarkus</groupId>
  <artifactId>quarkus-panache-mock</artifactId>
  <scope>test</scope>
</dependency>
```


Mocking de REST Clients

```
@QuarkusTest
public class SomeResourceTest {

    @InjectMock
    @RestClient
    FraudScoreService scoringService; ❶

    @Test
    public void testExample() {
        UUID uuid = ...code omitted...

        Mockito
            .when(scoringService.getById(Mockito.anyString())) ❷
            .thenReturn(new FraudScore(uuid, 9999)); ❸

        given() ❹
            .when()
                .get("/orders/score/an-order-uuid")
            .then()
                .statusCode(200)
                .body("$.size()", is(1));
    }
}
```


Spies

```
@QuarkusTest
public class SomeResourceTest {

    @InjectSpy
    OrderService orderService; ❶

    @Test
    public void testExample() {
        given() ❷
            .when()
                .get("/orders")
            .then()
                .statusCode(200)
                .body("$.size()", is(0));

        Mockito.verify(orderService, Mockito.times(1)).list(); ❸
    }
}
```

Recursos

<https://jakarta.ee/specifications/cdi/2.0/cdi-spec-2.0.html>

<https://site.mockito.org/>