Testing con Microservicios

Quarkus-junit5 Notas

- Soporte a CDI
- Testing HTTP simplificado
- Mocking simplificado
- Servicios contenerizados
- 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
public class MyEngineTest {
    @Inject
    Engine engine;
   @Test 3
    public void testEngine() {
        // Given
        engine.stop();
        // When
        engine.start();
        // Then
        Assertions.assertTrue(engine.isRunning());
```

Nota: tambien puede usar Hamcrest, AssertJ

DEV SERVICES

Recursos

https://junit.org/junit5/

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

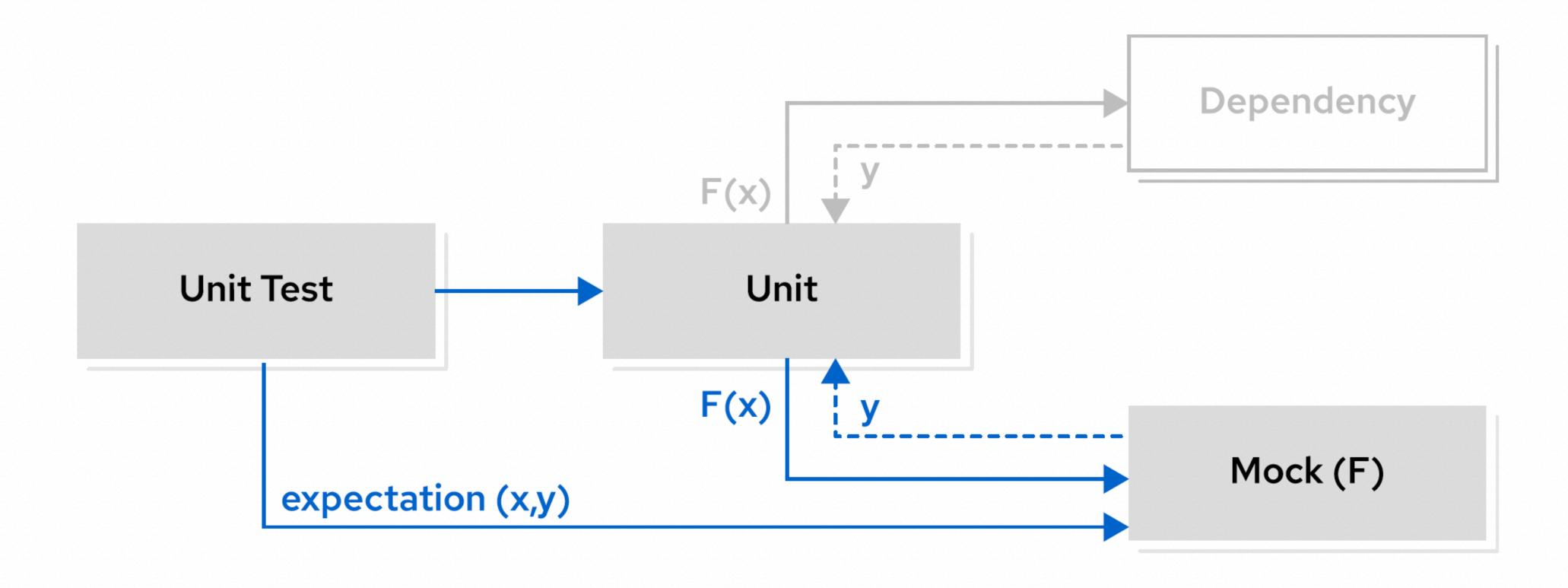


Figure 3.1: Mock type example

```
import io.quarkus.test.Mock;
import javax.enterprise.context.ApplicationScoped;
@Mock
@ApplicationScoped 2
public class OrderServiceMock extends OrderService {
    @Override
    public Order getOrder(UUID uuid) {
        return new Order(uuid, "Customer Name", 123);
```

```
@QuarkusTest
public class OrderServiceTest {
   @Inject 1
    OrderService orderService;
    @Test
    public void testExample() {
       UUID uuid = ...code omitted...
       Order order = orderService.getOrder(uuid);
       boolean isFraud = orderService.isFraud(order);
        ...code omitted...
```

```
@QuarkusTest
public class OrderServiceTest {
   @Test
    public void testExample() {
       UUID uuid = ...code omitted...
       OrderService orderService = Mockito.mock(OrderService.class);
       Mockito
           .when(orderService.getOrder(uuid))
            .thenReturn(new Order(uuid, "Customer Name", 123)); 🤨
       Mockito
           .when(orderService.getOrder(null))
           .thenThrow(new NullPointerException());
       Order order = orderService.getOrder(uuid);
        ...code omitted...
       Order order = orderService.getOrder(null);
        ...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()))
            .thenReturn(new Order(uuid, "Customer Name", 123));
       Order order = orderService.getOrder(uuid);
       Order order = orderService.getOrder(null);
        ...code omitted...
```

```
@QuarkusTest
public class SomeServiceTest {
   @InjectMock •
   OrderService orderService;
   @BeforeEach
    public void setup() {
        Mockito
            .when(orderService.getOrder(null))
            .thenThrow(new NullPointerException());
   @Test
    public void testExample() {
       UUID uuid = ...code omitted...
        Mockito
            .when(orderService.getOrder(Mockito.any()))
            .thenReturn(new Order(uuid, "Customer Name", 123)); 🤨
        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();
       Mockito.verify(expenseService).list();
       Mockito.verifyNoMoreInteractions(orderService, expenseService); 3
```

Mocking de Panache Entities

```
@QuarkusTest
public class SomeResourceTest {
    @BeforeAll
    public static void setup() {
        PanacheMock.mock(Order.class);
    @Test
    public void testExample() {
       Mockito
            .when(Order.listAll())
            .thenReturn(Collections.emptyList());
       given()
        .when()
            .get("/orders")
        .then()
            .statusCode(200)
            .body("$.size()", is(0));
```

Mocking de REST Clients

```
@QuarkusTest
public class SomeResourceTest {
    @InjectMock
    @RestClient
    FraudScoreService scoringService;
    @Test
    public void testExample() {
        UUID uuid = ...code omitted...
        Mockito
            .when(scoringService.getById(Mockito.anyString()))
            .thenReturn(new FraudScore(uuid, 9999)); 3
        given() 4
        .when()
            .get("/orders/score/an-order-uuid")
        .then()
            .statusCode(200)
            .body("$.size()", is(1));
}
```

Spies

```
@QuarkusTest
public class SomeResourceTest {
   @InjectSpy
   OrderService orderService;
   @Test
   public void testExample() {
       given() 2
        .when()
            .get("/orders")
        .then()
            .statusCode(200)
            .body("$.size()", is(0));
       Mockito.verify(orderService, Mockito.times(1)).list();
```

Recursos

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

https://site.mockito.org/

Dev Services

quarkus.devservices.enabled = false

quarkus.keycloak.devservices.enabled = false

quarkus.[dev_service].devservices.image-name

quarkus.datasource.devservices.image-name = registry.example.com/my-user/mysql:8

```
@QuarkusTest
@QuarkusTestResource(CustomBrokerResource.class)
public class MyTest {

    @Test
    public someTest() {
        // Perform the test that uses your external service
    }
}
```

```
public class CustomBrokerResource
   implements QuarkusTestResourceLifecycleManager {
   DockerImageName IMAGE = DockerImageName.parse(
       "quay.io/myorg/custom-broker:v1.0");
   GenericContainer<?> SERVICE = new GenericContainer<>( IMAGE )
       .withExposedPorts(9092)
      .withEnv("API_URL", "localhost:9092");
   @Override
   public Map<String, String> start() {
       SERVICE.start(); 5
       return Map.of("broker.internal.port", SERVICE.getPort()); 6
   @Override
   public void stop() {
```

```
@QuarkusTestResource(
   CustomBrokerResource.class 2
@Retention(RetentionPolicy.RUNTIME) [3]
@Target(ElementType.TYPE)
public @interface WithCustomBroker {
   String myPort() default ""; 🤨 🕡
```

```
public class CustomBrokerResource implements
QuarkusTestResourceConfigurableLifecycleManager<WithCustomBroker> { 12
    DockerImageName IMAGE = DockerImageName.parse(
        "quay.io/myorg/custom-broker:v1.0");
    GenericContainer<?> SERVICE = new GenericContainer<>( IMAGE );
    private String port;
    @Override
    public void init( WithCustomBroker params ) { 3
       port = params.myPort();
    @Override
    public Map<String, String> start() {
       SERVICE
            .withExposedPorts( port )
            .start();
         return Map.of("some.custom.broker.port", port );
    @Override
     public void stop() {
         SERVICE.stop();
```

```
@QuarkusTest
@WithMyCustomBroker( myPort = "9090" )
public class MyTest {

    @Test
    public someTest() {
        // Perform the test that uses your external service
    }
}
```

Recursos

https://www.testcontainers.org/

https://www.testcontainers.org/supported_docker_environment/

https://quarkus.io/version/2.13/guides/dev-services