

Enginyeria del Programari de Components i Sistemes Distribuïts

PROVA D'AVALUACIÓ CONTINUADA – Pràctica 3

Presentació

A la Pràctica 2 s'ha implementat tres microserveis del cas **Photo&Film4You**. En aquesta Pràctica 3 provarem (*testing*) un d'aquests tres microserveis, i validarem que el nostre codi segueix una arquitectura hexagonal.

La Pràctica 3 cobreix els continguts del llibres "Microservices Patterns" i "Fundamentals of Software Architecture" (veure a l'apartat *Recursos* els capítols que cal estudiar).

Competències

Aquesta pràctica treballa les següents competències del Grau d'Enginyeria Informàtica:

- Proposta i avaluació de diferents alternatives tecnològiques per resoldre un problema concret.
- Aplicació de les tècniques específiques de l'Enginyeria del Programari a les diferents etapes del cicle de vida d'un projecte.

Objectius

Els objectius de la Pràctica 3 són:

- Reconèixer els factors de testing i qualitat d'una especificació/disseny.
- Avaluar formalment el testing i la qualitat d'una implementació.
- Identificar els models més importants i estàndards de qualitat de programari.

Descripció de la Pràctica 3 a realitzar

La Pràctica 3 es compon de tres exercicis. Per cadascun d'ells cal descriure **com s'ha fet l'exercici** i aportar **instruccions** per a la seva correcta execució.

Exercici 1 (4 punts)

Partint de la solució de la Pràctica 2 publicada en l'enllaç:

<https://github.com/orgs/UOC-EPCSD-SPRING-2025/repositories>

Heu de desenvolupar **proves unitàries** emprant les eines **Junit**, **Spring Boot Test**, **Mockito** i **AssertJ**.

Notes: Podeu utilitzar com a referència per ajudar-vos a desenvolupar els tests:

- <https://github.com/anirban99/hexagonal-architecture/tree/master/src/test/java/com/example/hexagonal/architecture>
- <https://github.com/eugenp/tutorials/blob/master/spring-boot-modules/spring-boot-testing/src/test/java/com/baeldung/boot/>
- <https://spring.io/guides/gs/testing-web/>
- <https://docs.spring.io/spring-boot/docs/current/reference/html/features.html#features.testing.spring-boot-applications.with-mock-environment>

Consell: Per les proves unitàries no fem ús de cap infraestructura ni dependència real. No utilitzarem comunicació HTTP, servidors, bases de dades, serveis externs o altres microserveis. Tot aspecte relacionat amb dependències s'ha de simular (amb **Mockito**), excepte les classes de domini (Product, ...).

QUÈ CAL FER

Desenvolupar les següents classes i tests:

1. **DigitalItemUnitTest**. Incloure un test sobre la classe de domini *DigitalItem* que verifiqui que en crear un nou *DigitalItem*, el seu *status* sigui AVAILABLE.
2. **DigitalSessionServiceUnitTest**. Incloure tests que verifiquin:
 - Mètode Test 1: Quan cridem al mètode *findDigitalSessionByUser* amb un *userId* existent/vàlid, obtenim correctament el llistat de sessions digitals de l'usuari.

- Mètode Test 2: Quan cridem al mètode *findDigitalSessionByUser* amb un *userId* inexistent, obtenim l'excepció que indica que l'usuari no s'ha trobat.

Notes:

- Les dependències respecte el *RestTemplate* i el *DigitalSessionRepository* han de ser simulades (*mock*).

3. **DigitalItemRestControllerUnitTest.** Incloure un test que verifiqui que si cridem a l'endpoint de *findDigitalItemBySession* obtenim correctament els ítems digitals que pertanyen a la sessió amb l'identificador especificat.

Notes:

- No es pot aixecar el servidor per fer la prova. Cal simular la infraestructura web per rebre i tractar la petició REST, així com la dependència amb *DigitalItemService*.

Exercici 2 (4 punts)

Partint de la solució de la PRAC2 publicada a:

<https://github.com/UOC-EPCSD-SPRING-2025/repositories>

Heu de desenvolupar **proves d'integració**, utilitzant **JUnit**, **Spring Boot Test** i **AssertJ**.

Notes:

- Podeu usar com a referència per ajudar-vos a desenvolupar els tests: <https://github.com/eugenp/tutorials/tree/master/spring-boot-modules/spring-boot-testing/src/test/java/com/baeldung/boot/>

Consell: Les proves d'integració han de provar que el servei es comunica amb les seves dependències reals i la infraestructura. Per tant, no es fa ús de la llibreria **Mockito** com en les proves unitàries.

QUÈ CAL FER

Desenvolupar la següent classe amb el seu test:

1. **DigitalItemRepositoryIntegrationTest.** Provar que després d'afegir una *DigitalSession* amb alguns *DigitalItems*, podem recuperar aquests correctament mitjançant el mètode *findDigitalItemBySession*.

Notes:

- En aquest cas serà necessari una base de dades real per verificar que la integració funciona correctament.

2. **DigitalItemRestControllerIntegrationTest.** Provar que després d'afegir una *DigitalSession* amb alguns *DigitalItems*, podem recuperar aquests correctament mitjançant una petició HTTP a la ruta que exposa el *DigitalItemRestController* per obtenir els *DigitalItems* de la *DigitalSession*.

Notes:

- En aquest cas NO s'ha de fer la crida al mètode del Controlador, si no que cal fer directament la petició HTTP al servei web de l'aplicació Spring.

Exercici 3 (2 punts)

Partint de la solució de la PRAC2 publicada a:

<https://github.com/orgs/UOC-EPCSD-SPRING-2025/repositories>

Volem verificar que l'arquitectura es basa correctament en una arquitectura hexagonal, i que fem servir bones regles arquitectòniques.

Lectures recomanades:

- Capítol "Measuring and Governing Architecture Characteristics" del llibre "Fundamentals of Software Architecture"
- Documentació ArchUnit: https://www.archunit.org/userguide/html/000_Index.html#_junit_4_5_support
- Exemples ArchUnit amb JUnit5: <https://github.com/TNG/ArchUnit-Examples/tree/main/example-junit5/src/test>

QUÈ CAL FER

Desenvolupar els següents dos tests automàtics de verificació de l'arquitectura sobre el microservei *UserService*:

- Es compleix l'arquitectura hexagonal (anomenada *onion* a **ArchUnit**)
- Les classes que es troben al *package domain.service* i estan anotades amb **@Service**, tenen el seu nom acabat en **ServiceImpl**.

Recursos

Recursos Bàsics

- Llibre "Microservices Patterns" (Chris Richardson): Chapter 9. "Testing microservices: Part 1". Introduction.
- Llibre "Microservices Patterns" (Chris Richardson): Chapter 10. "Testing microservices: Part 2".
- Llibre "Fundamentals of Software Architecture" (Richards & Ford):
 - o Chapter 3: "Modularity"
 - o Chapter 6: "Measuring and Governing Architecture Characteristics"

Criteris d'avaluació

- En aquesta activitat no està permès utilitzar **eines d'intel·ligència artificial**, i cal realitzar-la de manera **estrictament individual**. En cas de detectar irregularitats es penalitzarà la prova amb una D com a nota. Això inclou la reutilització de solucions d'aquesta pràctica de semestres anteriors. Al pla docent i al [web sobre integritat acadèmica i plagi de la UOC](#) trobareu informació sobre què es considera conducta irregular en l'avaluació i les conseqüències que pot tenir.
- El pes de cada exercici està indicat en l'enunciat.
- Trobareu una descripció ampliada dels criteris d'avaluació de cada exercici en l'annex.
- Cal justificar la solució a cadascun dels exercicis. Es valorarà tant la correcció de la solució com la justificació donada

Format i data de lliurament

Cal lliurar un únic **document** PDF *CognomsNom_EPCSDPRAC3.pdf* amb la descripció de com s'ha realitzat la pràctica (argumentar decisions i instruccions de com s'han d'executar les proves desenvolupades) i lliurar el **codi dels tests desenvolupats** (en un fitxer ZIP).

El document PDF i el fitxer ZIP s'han de lliurar a l'espai "Lliurament de la PRAC3" de l'aula abans de les **23:59 hores del dia 9 de juny de 2025**. No s'acceptaran lliuraments fora de termini.

Annex

Criteris d'avaluació de la Pràctica 3

	Excel·lent (10 punts)	Notable (7 punts)	Suficient (5 punts)	Insuficient (2,5 punts)	Sense resposta / Tot malament (0 punts)
Ex. 1	<p>4 tests implementats i justificats correctament als exercicis 1.1, 1.2 i 1.3.</p> <p>Ús correcte de Junit a tots els 4 tests dels exercicis 1.1, 1.2 i 1.3.</p> <p>Ús correcte de Mockito als 3 tests dels exercicis 1.2 i 1.3.</p> <p>Separar en 2 mètodes els dos tests de l'exercici 1.2.</p> <p>Qualitat general de la implementació <u>perfecte</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.).</p> <p>(4 punts)</p>	<p>3 tests implementats i justificats correctament als exercicis 1.1, 1.2 i 1.3.</p> <p>Ús correcte de Junit a tots els 4 tests dels exercicis 1.1, 1.2 i 1.3.</p> <p>Ús correcte de Mockito en 2 dels 3 tests dels exercicis 1.2 i 1.3.</p> <p>Separar en 2 mètodes els dos tests de l'exercici 1.2.</p> <p>Qualitat general de la implementació <u>bona</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.).</p> <p>(3 punts)</p>	<p>2 tests implementats i justificats correctament als exercicis 1.1, 1.2 i 1.3.</p> <p>Ús correcte de Junit en 2 dels 4 tests dels exercicis 1.1, 1.2 i 1.3.</p> <p>Ús correcte de Mockito en 1 dels 3 tests dels exercicis 1.2 i 1.3.</p> <p>No separar en 2 mètodes els dos tests de l'exercici 1.2.</p> <p>Qualitat general de la implementació <u>suficient</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.).</p> <p>(2 punts)</p>	<p>1 test implementat i justificat correctament als exercicis 1.1, 1.2 i 1.3.</p> <p>Ús incorrecte de Junit en els 4 tests dels exercicis 1.1, 1.2 i 1.3.</p> <p>No ús de Mockito en els 3 tests dels exercicis 1.2 i 1.3. No separar en 2 mètodes els dos tests de l'exercici 1.2.</p> <p>Qualitat general de la implementació <u>insuficient</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.).</p> <p>(1 punt)</p>	<p>Sense resposta, no hi ha cap justificació o tot incorrecte.</p> <p>(0 punts)</p>
Ex. 2	<p>2 tests implementats i justificats correctament en els ex 2.1 i 2.2</p> <p>Qualitat general de la implementació <u>perfecte</u> (criteris:</p>	<p>2 tests implementats correctament, però justificats incorrectament.</p> <p>Qualitat general de la implementació <u>bona</u> (criteris:</p>	<p>1 test implementat i justificat correctament.</p> <p>Qualitat general de la implementació <u>suficient</u> (criteris: llegibilitat, formatat, sense warnings,</p>	<p>1 test implementat però justificat incorrectament.</p> <p>Qualitat general de la implementació <u>insuficient</u> (criteris: llegibilitat, formatat, sense warnings,</p>	<p>Sense resposta, no hi ha cap justificació o tot incorrecte.</p> <p>(0 punts)</p>

	llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (4 punts)	llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (3 punts)	utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (2 punts)	utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (1 punt)	
Ex. 3	2 tests implementats i justificats correctament. Qualitat general de la implementació <u>perfecte</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (2 punts)	2 tests implementats i però justificats incorrectament. Qualitat general de la implementació <u>bona</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (1,5 punts)	1 test implementat i justificat correctament. Qualitat general de la implementació <u>suficient</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (1 punts)	Tests mal implementats i no justificats. Qualitat general de la implementació <u>insuficient</u> (criteris: llegibilitat, formatat, sense warnings, utilitza constants en comptes de hardcode en les crides als mètodes, etc.). (0,5 punts)	Sense resposta, no hi ha cap justificació o tot incorrecte. (0 punts)