

CFGS Desenvolupament d'Aplicacions Web
Mòdul 7 B2 – Desenvolupament web en entorn servidor
UF3 – Tècniques d'accés a dades
Unitat 4 – Tècniques d'accés a dades
EAC4
(Curs 2022–23 / 2n semestre)

Presentació i resultats d'aprenentatge

Aquest exercici d'avaluació continuada (EAC) es correspon amb els continguts treballats a la unitat 4 “Tècniques d'accés a dades”

Els **resultats d'aprenentatge** que es plantegen són:

- Desenvolupa aplicacions d'accés a magatzems de dades, aplicant mesures per mantenir la seguretat i la integritat de la informació.

Criteris d'avaluació

La puntuació màxima assignada a cada activitat s'indica a l'enunciat.


Els criteris que es tindran en compte per avaluar el treball de l'alumnat són els següents:

- S'han analitzat les tecnologies que permeten l'accés mitjançant programació a la informació disponible en magatzems de dades.
- S'han creat aplicacions que estableixin connexions amb bases de dades.
- S'ha recuperat informació emmagatzemada en bases de dades.
- S'ha publicat en aplicacions web la informació recuperada.
- S'han utilitzat conjunts de dades per emmagatzemar la informació.
- S'han creat aplicacions web que permetin l'actualització i l'eliminació d'informació disponible en una base de dades.
- S'han utilitzat transaccions per mantenir la consistència de la informació.
- S'han provat i subsidiat les aplicacions.

Forma i data de lliurament

Un cop finalitzat l'exercici d'avaluació continuada heu d'enviar la resposta des de l'apartat **Lliurament EAC4** de l'aula, dins del termini establert. Tingueu en compte que el sistema no permetrà fer lliuraments després de la data i hora indicades.

Important

	Codi: I71	Exercici d'avaluació continuada 4	Pàgina 1 de 10
	Versió: 03	DAW_M07B2_EAC4_Enunciat_2223S2	Lliurament: 23/03/2023

Formació professional


Heu d'entregar la **carpeta completa del projecte de Netbeans**, conjuntament amb aquest document amb les respostes degudament emplenades, si escau, i comprimir-ho tot amb un **compressor lliure (zip, 7z o tar.gz)**. RAR no és un compressor lliure i no es pot fer servir per lliurar fitxers.

Esborreu la carpeta «target» del projecte abans de lliurar.

El nom del fitxer serà el següent: **DAWM07B2_EAC4_Cognom1_Inicial del cognom2.zip**. Els cognoms s'escriuran sense accents. Per exemple, l'estudiant *Joan García Santos* posaria el següent nom al seu fitxer de l'EAC4: **DAWM07B2_EAC4_Garcia_S.zip**.

Substituïu **Nom i cognoms** de la capçalera per les vostres dades personals.

El termini de lliurament finalitza a les **23:55 h** del dia **23/03/2023**.

	Codi: I71	Exercici d'avaluació continuada 1	Pàgina 2 de 10
	Versió: 03	DAW_M07B2_EAC1_Enunciat_2223S2	Lliurament: 23/03/2023

Enunciat

General

L'aplicació

Partiu d'una senzilla aplicació de gestió. Fa referència a una botiga on reservar *xollos* online, ja desenvolupada amb Spring MVC. Aquesta aplicació té finalitat didàctica i per tant, es dona prioritat a la utilització dels coneixements assolits, sobre l'exhaustivitat de les funcionalitats a desenvolupar.

L'aplicació de partida manega *xollos* amb codi, numeroUnitats, numeroReserves, títol i descripció. Els números d'unitat i reserves son enters, la resta cadenes de caràcters.

Gràfic dels casos d'ús

El gràfic següent mostra els casos d'ús de l'aplicació. Es pot afegir un nou xollo, es pot consultar per codi per veure'n el detall i es pot reservar un article (fins al numeroUnitats, on ja no es podrà reservar més). Aquesta reserva (o venda) es farà mitjançant una URL concreta que s'explica a la pantalla de compra.



	Codi: I71	Exercici d'avaluació continuada 1	Pàgina 3 de 10
	Versió: 03	DAW_M07B2_EAC1_Enunciat_2223S2	Lliurament: 23/03/2023

Model de la Gestió de Xollos

Només hi ha un component al model. Es tracta de Xollo amb codi, numeroUnitats, numeroReserves, títol i descripció. Els números d'unitat i reserves son enters, la resta cadenes de caràcters.

Part Spring MVC

En aquesta pràctica es donen fetes les capes de presentació (vistes i controladors) i de servei.

Paquet / Directori	Fitxer / Classe	Objectiu
WEB-INF/views	xxx.jsp	Pàgines jsp que corresponen a les vistes de l'aplicació.
WEB-INF	web.xml	Configuració de l'aplicació. En el nostre cas només el fem servir per dir al servidor d'aplicacions que carregui el DispatcherServlet.
WEB-INF/spring	DispatcherServlet-servlet.xml	Conté tota la configuració Spring i Spring MVC i determina els beans que s'han de carregar.
cat.xtec.ioc.controller cat.xtec.ioc.service cat.xtec.ioc.service.impl	Classes controladores i de servei.	Classes ja creades seguint el model DDD (Domain Driven Design) de Spring MVC. És a dir, una petició de l'usuari crida al controlador i aquest al servei (a la implementació del servei). El servei crida al repositori que accedirà a les dades.
cat.xtec.ioc.repository	XolloRepository	És la interfície del repositori que han d'implementar les classes que interactuen amb les dades. En realitat el servei injecta una implementació concreta de repositori. La sentència: @Autowired private XolloRepository xolloRepository; injecta la implementació del repositori anotada amb @Repository

Les preguntes de la pràctica són diverses implementacions del repositori per desenvolupar diversos mètodes d'accés a dades per accedir a dues bases de dades.

Pregunta 1. Accés a dades amb JDBC (3 punts)

Partint del projecte **dawm07eac4TotxolloEnunciat** que podeu descarregar de l'aula, l'objectiu d'aquesta pregunta és crear una implementació del repositori per accedir / actualitzar els xollos amb JDBC. A més fareu servir els test JUnit que us permetran provar la funcionalitat independentment de la part vista.

Si heu fet el primer bloc del mòdul, haureu practicat amb aplicacions Spring MVC on el repositori era simplement una llista en memòria. En aquest exercici, tota la part d'Spring MVC es dona feta, i només haureu d'implementar la part de JDBC.

La base de dades amb la que treballarem en aquesta pregunta és H2 Database, tal i com es fa servir als materials.

El que heu de fer és afegir les classes, dependències i fitxers de configuració necessaris per poder desenvolupar el codi de la classe XolloDAO.

Per situar una mica, penseu que XolloDAO implementa a XolloRepository, i pretén proporcionar el diàleg amb la base de dades H2 «xollos_db». Per fer això, es dona feta tota la part de connexió i configuració de la base de dades. Reviseu la graella següent per saber quins fitxers teniu i quins cal crear de nou. Aquesta hauria de ser la foto del final del procés

Paquet / Directori	Fitxer / Classe	Objectiu
src/main/resources	db.properties	Conté els paràmetres de connexió a la base de dades.
src/main/resources	init.sql	Es crea la taula xollos a la base de dades i s'afegeix diversos xollos. Això no caldria en una aplicació real, a no ser que fos una migració de dades.
src/test/resources	db_test.properties	Conté els paràmetres de connexió a la base de dades a fer servir en els tests.
	db_wrong_driver.properties	Conté els paràmetres de connexió erronis a fer servir en els tests.
src/test/resources	init.sql	Es crea la taula xollos a la base de dades i s'afegeix diversos xollos per fer-los servir en els tests.
cat.xtec.ioc.repository.impl	DBConnection	És la classe que realitza la connexió a la base de dades. Us heu de fixar que el constructor: <pre>public DBConnection(String connectionFile)</pre> accepta el paràmetre connectionFile i llavors a producció farem servir db.properties i a test db_test.properties que definiran els paràmetres de connexió.
Test Packages cat.xtec.ioc.repository.impl	DBConnTest	Realitza els tests de connexió. Fixeu-vos que la connexió la fa amb els paràmetres de “db_test.properties”. S’han de realitzar dos tests, un amb els paràmetres db_test.properties i un altre amb db_wrong_driver.properties, que revisi que la connexió es faci i no respectivament.
Test Packages cat.xtec.ioc.repository.impl	XollosDAOTest	Realitza els tests de classe que heu de desenvolupar (XolloDAO). Heu de provar cada un dels mètodes de la nova implementació de XolloRepository.

Un cop desenvolupeu tots els mètodes XolloDAO segons la interfície XolloRepository i passin els tests sense errors, podeu engegar Glassfish i executar l'aplicació al navegador. Us ha de funcionar. **Procureu refactoritzar el codi del DAO que implementeu.**

Pregunta 2. Accés a dades amb Java Enterprise Edition (4 punts)

Continuem amb el projecte dawm07eac4TotxolloEnunciat on teniu totes les dependències necessàries ja creades a pom.xml.

L'objectiu d'aquesta pregunta és accedir a les dades de xollos amb MySQL i ajudats per JPA, concretament amb EclipseLink.

El que heu de fer és crear una base de dades MySQL, configurar els recursos de Glassfish, les unitats de persistència i una implementació del repositori que en comptes d'anar a la base de dades H2, vagi a la MySQL. En els paràgrafs següents s'explica detalladament el que heu de fer.

Sobre versions

Aquesta pràctica té un caràcter didàctic. Es podria haver fet tot amb «testing» i sense part vista, però he entès que és més aclaridor i pedagògic fer-la així perquè veureu que sense canviar res de la lògica de l'aplicació, fareu servir tres implementacions diferents per accedir a les dades, una per a cada pregunta. A més, la primera pregunta fa servir una base de dades en memòria i la resta MySQL.

A tal situació, posarem en una mateixa aplicació dues implementacions de JPA: EclipseLink per a la segona pregunta i Hibernate per a la tercera. I ambdues convivint amb el framework Spring MVC i tot dins del servidor d'aplicacions Glassfish (o Payara <https://www.payara.fish/>)

Això ens porta a que en el mateix projecte hi haurà diversos entorns amb diverses versions que depenen entre ells. Per això, us recomano fer servir les versions on l'aplicació ja està testejada.

Per a Glassfish, heu de fer servir la versió **Glassfish 5.0** exactament. La podeu descarregar de: <http://download.oracle.com/glassfish/5.0/release/index.html>

També heu de baixar el connector de MySQL des de: <https://dev.mysql.com/downloads/connector/j/>

Feu servir el connector independent de plataforma i heu de copiar el fitxer mysql-connector-java-8.0.12.jar contingut sobre ...\\domains\\domain1\\lib. Als materials es fa servir una versió anterior però aquesta aplicació està provada amb aquest i en la pregunta concreta explicaré alguns petits canvis.

Si Glassfish és en execució, no oblideu de reiniciar.


Base de dades xollos_db

Heu de crear la base de dades «xollos_db». Us podeu ajudar de MySQL WorkBench o phpMyAdmin.

Doneu permisos totals a l'usuari «root» amb contrasenya «root».

Heu de crear la taula xollos. Podeu crear-la amb la sentència dels init.sql que teniu al projecte Maven.

Heu d'afegir alguns xollos. Per fer això, també podeu executar les sentències de init.sql. Canvieu els codis dels productes i així us assegureu que al executar l'aplicació esteu veient els elements de MySQL i no de H2.

	Codi: I71	Exercici d'avaluació continuada 1	Pàgina 6 de 10
	Versió: 03	DAW_M07B2_EAC1_Enunciat_2223S2	Lliurament: 23/03/2023

Configuració a Glassfish

Creeu un Connection Pool amb nom «xollos_db» que apunti a la base de dades «xollos_db».

Només us heu de quedar amb les propietats addicionals següents:

User	root
password	root
Database Name	xollos_db
ServerName	localhost
Port	3306
url	jdbc:mysql://localhost:3306/xollos_db? useUnicode=true&useJDBCCompliantTimezoneShift=true&useLegacyDatetimeCode=false&serverTime zone=UTC&autoReconnect=true&useSSL=false

Heu d'eliminar la resta de propietats que us hagi creat Glassfish. Sobretot, la url ha de ser exactament la que mostra la taula.

Si feu ping, no us connectarà perquè no troba la classe del DataSource. Això és perquè fem servir la versió 8.0.12 del connector de MySQL. Aneu a la consola d'administració de Glassfish, editeu el Connection Pool «xollos_db» i canvieu la propietat «Datasource Classname» (a General Settings) per:
`com.mysql.cj.jdbc.MysqlDataSource`

Ara ja podeu fer ping i us ha de connectar.

Creeu un JDBC Resource amb nom «xollos_db__pm» (VIGILEU les dobles _) apuntant al pool anterior.

Unitat de persistència

Creeu una Unitat de persistència amb nom «XollosPersistenceUnit» en un nou fitxer «persistence.xml» a `src/main/resources/META-INF` (el directori també serà nou)

El tipus de transacció ha de ser «JTA» i ha de connectar el JDBC Resource creat abans amb la classe «`cat.xtec.ioc.domain.Xollo`».

Entitat Xollo

Heu de canviar la classe «Xollo» per tal de què sigui una entitat associada a la taula xollos de la base de dades MySQL. Noteu que necessàriament ha d'implementar «Serializable».

Feu les anotacions sobre cada propietat de tal manera que sigui coherent amb la definició de la taula «xollos» de MySQL.

Repositori XolloMySQL

Creeu la classe «XolloMySQL» al paquet «cat.xtec.ioc.repository.impl» i implementant «XolloRepository».

Heu d'implementar els mètodes que ens obliga la interfície XolloRepository. Heu d'aconseguir connectar-vos a la base de dades de MySQL mitjançant la unitat de persistència, i a més, les sentències per accedir / actualitzar la taula de la base de dades han de construir-se amb Java Persistence Query Language (JPQL).

Recordeu que JPQL són les sentències del tipus «select object(o) from ...» que podeu veure als materials.

En el cas de la connexió, com estem fent servir Spring MVC i l' anotació @Repository, no tenim tantes ajudes com mostren els materials quan fem servir un Bean tipus @Stateless. A tal situació, en comptes d'injectar l'EntityManager per dialogar amb la base de dades, el que heu de fer és:

1) Crear una propietat a XolloMySQL

```
private EntityManager entityManager;
```

2) Crear un constructor a XolloMySQL sense paràmetres

```
public XolloMySQL(){  
    entityManager = Persistence.createEntityManagerFactory("XollosPersistenceUnit").createEntityManager();  
}
```

Això us permetrà fer servir «entityManager» per executar els querys

Notes

- L'anotació @Stateless fa que podem desentendre'ns dels commit, però @Repository no. Tampoc feu servir l'anotació de classe @Transactional que faria això, perquè no hem configurat res per fer-la servir.
- Al posar @Repository a XolloMySQL, hi ha un encavalcament amb el @Repository de XolloDAO. Per això, per provar-ne un, heu de treure el @Repository de l'altre.

Pregunta 3. Accés a dades amb Spring i Hibernate (3 punts)

Continuem amb el projecte **dawm07eac4TotxolloEnunciat** on teniu totes les dependències necessàries ja creades a pom.xml.

L'objectiu d'aquesta pregunta és accedir a les dades de xollos amb Spring i Hibernate. Recordeu que Hibernate és una implementació de JPA.

En definitiva, heu de desenvolupar una nova implementació del repositori que faci que l'aplicació es comuniqui amb la base de dades xollos_db amb l'ajuda d'Spring i d'Hibernate.

L'estratègia és la mateixa que podeu veure als materials de la unitat 4 / apartat 3. Però un cop heu seguit els materials, per ajudar-vos en aquesta pregunta, podeu consultar el projecte final de l'apartat dels materials. Això és perquè la solució d'aquesta pregunta ha de ser amb el codi refactoritzat, amb els fitxers / classes que se us indica a continuació.

Fitxers de propietats

Els heu de crear a «src/main/resources».

jdbc.properties

Heu d'establir les propietats de connexió amb la base de dades xollos_db amb el driver jdbc de MySQL.

hibernate.properties

Heu d'establir les propietats mínimes per fer servir Hibernate.

Classes de configuració

Fixeu-vos que podríem fer la configuració amb fitxers XML, però se us demana fer la configuració amb classes, tal i com podeu veure als materials.

Les heu de crear a un nou paquet «cat.xtec.ioc.config».

HibernateConfiguration

Heu de desenvolupar aquesta classe de configuració amb els següents requeriments:

- Ha d'habilitar la capacitat de gestionar transaccions amb la base de dades.

- Ha de crear el Bean sessionFactory de tipus LocalSessionFactoryBean establint el DataSource del paràmetre.

- Ha de crear el Bean transactionManager de tipus HibernateTransactionManager assignant-li el sessionFactory del paràmetre.


HibernateMysqlConfiguration

Heu de desenvolupar aquesta classe de configuració amb els següents requeriments:

- Ha d'habilitar la capacitat de gestionar transaccions amb la base de dades.

- Ha de fer servir els fitxers de propietats jdbc.properties i hibernate.properties

- Ha d'importar la classe de configuració HibernateConfiguration

	Codi: I71	Exercici d'avaluació continuada 1	Pàgina 9 de 10
	Versió: 03	DAW_M07B2_EAC1_Enunciat_2223S2	Lliurament: 23/03/2023

Ha de crear el Bean dataSource de tipus DataSource amb les propietats del jdbc

Implementació del repositori. XolloHibernate

Al paquet «cat.xtec.ioc.repository.impl» creeu la classe XolloHibernate implementant a XolloRepository.

Els requeriments per aquesta classe són:

Ha de ser un Repository (mireu la nota al peu).

Heu de fer servir SessionFactory sessionFactory injectat en aquesta classe (per la configuració feta, aquesta injecció serà del Bean d'Hibernate i per tant, obtindreu una sessió d'Hibernate).

Els mètodes de la classe han de fer servir transaccions.

Heu d'implementar els mètodes que defineix l'interfície XolloRepository amb el llenguatge d'Hibernate, és a dir, fent servir: sessió, Criteria, ...

Notes:

Al posar @Repository a XolloHibernate, hi ha un encavalcament amb el @Repository de XolloDAO o de XolloMySQL. Per això, per provar-ne un, heu de treure el @Repository als altres.

A HibernateMysqlConfiguration, segons la vostra instal·lació, si no funciona

«@PropertySource("jdbc.properties")», feu servir «@PropertySource("classpath:jdbc.properties")».

Si teniu problemes perquè el vostre IDE (NetBeans) i concretament Glassfish està fent servir JDK7 i voleu fer servir per defecte JDK8, llavors l'heu de configurar des de fora de NetBeans, actualitzant els fitxers:

- o asenv.bat de «directori d'instal·lació de Glassfish»/glassfish/config (pot variar la localització depenent del SO)

- afegiu-hi al final la sentència (és un exemple, l'heu d'adaptar a la vostra instal·lació)
set AS_JAVA=C:\Program Files\Java\jdk1.8.0_74

- domain-info.xml de «directori d'instal·lació de Glassfish»/glassfish/domains/«el vostre domini»/init-info

- o canvieu el valor de javahome a (és un exemple, l'heu d'adaptar a la vostra instal·lació)
javahome="C:\Program Files\Java\jdk1.8.0_74"


Si us trobeu amb l'error

java.lang.NoSuchMethodError: org.jboss.logging.Logger.debugf(Ljava/lang/String;I)V

llavors heu de descarregar el fitxer jboss-logging-3.3.0.Final.jar de

<https://mvnrepository.com/artifact/org.jboss.logging/jboss-logging/3.3.0.Final>

i substituir el que trobareu a «directori d'instal·lació de Glassfish»/glassfish/modules (segurament serà jboss-logging.jar) pel descarregat

	Codi: I71	Exercici d'avaluació continuada 1	Pàgina 10 de 10
	Versió: 03	DAW_M07B2_EAC1_Enunciat_2223S2	Lliurament: 23/03/2023