

# Alma Mater Studiorum *Università di Bologna*

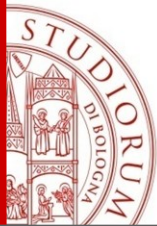
---

## Proposta di esercizio: EJB

*con uso di JPA/Hibernate*

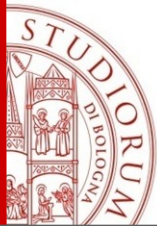
### Sistemi Distribuiti M

CdS Laurea Magistrale in Ingegneria Informatica  
I Ciclo - A.A. 2021/2022



# Obiettivi e skill

- Mostrare come Java EE semplifichi la programmazione di una applicazione commerciale, introducendo componenti modulari e riusabili, ***gestendo automaticamente molti aspetti del loro sviluppo e supporto runtime***
- Valutare in maniera critica la realizzazione di servizi *Enterprise-ready* e moderni appoggiandosi ad una piattaforma di supporto completa, come può essere un Application Server (AS) come Jboss (aka Wildfly)
- Utilizzo di metodologie moderne come REST per creare servizi Web altamente scalabili, eterogenei e portabili
- Utilizzo di ***JPA/Hibernate come piattaforma di ORM*** per semplificare la gestione della persistenza

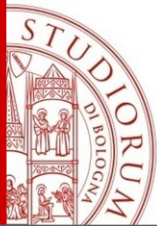


# Progetto della logica applicativa

Si sviluppi un'applicazione Web JEE per la *gestione del magazzino di un ipotetico negozio* di componenti informatiche. Tale applicazione deve avere l'aspetto di un sito e-commerce, con possibilità per un utente di mettere le componenti elettroniche nel *carrello* e successivamente di inviare l'ordine d'acquisto. Gli ordini poi (mediante Hibernate e/o JPA) dovranno essere salvati permanentemente in un DB

Per realizzare l'applicazione si consideri il seguente modello:

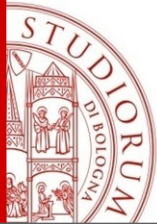
- Prodotti (componenti per pc: processori, ram, schede video ecc..)
- Produttori (amd, intel, nvidia ecc..)
- Carrello (SessionBean in attesa di essere ordinati)
- Ordini (EntityBean che devono essere salvati in memoria persistente)



# Progetto della logica applicativa

## *Si richiede di:*

- realizzare la parte di logica di accesso al db (pattern DAO) mediante componenti EJB 3.0:
  - *Utilizzare mapping Object-Relational tramite componenti Entity Bean*
  - *Realizzare DAO tramite opportuni componenti Session Bean*
- inoltre, estendere la logica applicativa già descritta, aggiungendo un meccanismo di **logging** (potenzialmente) remoto:
  - Ciascun metodo che preveda scritture su DB (aggiunta di nuovi componenti, produttori, ordini ecc...) deve inviare un messaggio JMS a un opportuno componente di logging
  - Componente di logging (realizzato come Message Driven Bean) scrive su opportuno log (anche solo stdout) il messaggio ricevuto



# Parte opzionale dell'esercitazione

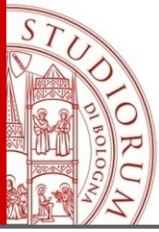
---

*Si richiede inoltre di:*

- eseguire il deployment della applicazione su un'installazione dell'application server Wildfly (in versione  $\geq 17.0.0$ ) mediante Gradle
- di eseguire testing e continuous delivery dell'applicazione tramite Travis

**Nota:** al seguente link è possibile trovare lo starter-kit

<https://github.com/unibo-distributedsystemsm/Ex-01-BusinessLogic-EJB>



# Riferimenti e possibili approfondimenti

---

- <https://www.oracle.com/technetwork/java/javaee/appmodel-135059.html>
- <https://docs.oracle.com/javaee/7/tutorial/>
- <https://www.journaldev.com/java-tutorial-java-ee-tutorials>