

QUÉ VEUREM


```
@Entity
public class Persona {

    @Id
    private Long id;
    private String nom;

    public Long getId() {
        return id;
    }
    public void setId(Long id) {
        this.id = id;
    }
    public String getNom() {
        return nom;
    }
    public void setNombre(String nom) {
        this.nom = nom;
    }
}
```

MAPPING

insert into persona (nom, id) values (?, ?)

API

```
@Component
public class Util {

    @PersistenceContext
    private EntityManager em;

    @Transactional
    public void alta (long id, String nom) {
        Persona nova = new Persona();
        nova.setId(id);
        nova.setNom(nom);
        em.persist(nova);
    }
}
```

EntityManager : punt d'accés a la major part de les operacions que emprarem

persist: fa un objecte "persistent" (guardat a la base de dades)

Quins tipus de components tendrem?

```
@Component
public class Util {

    @PersistenceContext
    private EntityManager em;

    @Transactional
    public void alta (long id, String nom) {
        Persona nova = new Persona();
        nova.setId(id);
        nova.setNombre(nom);
        em.persist(nova);
    }
}
```

ARQUITECTURA

Com delimitam les transaccions?

Com organitzem el codi?

Com se integra amb els
@Controler?

CONSULTES

- En relacional tenim l'SQL: Com serà el llenguatge de consultes en JPA?
- Com afectarà els mappings que feim?
- Quina API tindrem per accedir a aquestes consultes?
- Quines eines tenim per millorar el rendiment?

CONFIGURACIÓ

```
<dependencies>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-jpa</artifactId>
</dependency>

<dependency>
  <groupId>com.h2database</groupId>
  <artifactId>h2</artifactId>
  <scope>runtime</scope>
</dependency>
</dependencies>
```

Crea un BBDD en memòria

Crea automàticament
l'esquema de base de dades

