

find, persist, merge,  
remove



# EntityManager

```
public class Util {  
  
    @PersistenceContext  
    private EntityManager em;  
  
    @Transactional  
    public Persona metode (long id) {  
        return em.find(Persona.class, id);  
    }  
}
```



```
<T> T find(Class<T> entityClass, Object primaryKey)
```

*Carrega l'objecte del tipus especificat a **entityClass** amb clau primària **primaryKey***

### Exemple

EntityManager



```
Persona persona = em.find(Persona.class, 1L);
```

*Retorna l'objecte del tipus Persona (entitat) amb clau primària 1  
Si no existeix, retorna null*



```
void persist(Object entity)
```

*Fa persistent l'objecte passat com a paràmetre*

Exemple

```
Persona persona = new Persona ();  
// inicialitzacions  
em.persist(persona);
```



`<T> T merge(T entity)`

*Guarda de forma persistent l'estat actual del paràmetre entity*



Si hem de continuar treballant sobre l'objecte, ho hem de fer amb la referencia retornada per el mètode !

Exemple

```
persona = em.merge(persona);
```



```
void remove(Object entity)
```

*Elimina un objecte de la base de dades*

`IllegalArgumentException` - if the instance is not an entity or is a detached entity



No hem vist que es un "detached entity".

Per ara: l'objecte que passem per paràmetre al remove ha de ser carregat per el mateix entity manager

```
Persona eliminada = em.find(Persona.class, id);  
em.remove(eliminada);
```

clau primària de l'objecte que volem eliminar