



# Especialista en Aplicaciones y Servicios Web con Java Enterprise

## Enterprise JavaBeans

### Sesión 8:

### Introducción a JPA



# Java Persistence API

- Uno de los cambios más importantes en EJB 3.0
- Basado en el enfoque de Hibernate de clases persistentes
- Uso intensivo de anotaciones
- Disponible en Java EE 5 y en Java SE 6.



# Entidades

- Las clases entidad son clases normales Java (POJO) con campos y métodos de acceso (set) y lectura (get).
- Persistencia
- Identidad
- Transaccionalidad
- Relaciones entre entidades



# Ejemplo de entidad JPA

```
@Entity
public class Empleado {
    @Id private int id;
    private int id;
    private String nombre;
    private long sueldo;

    public Empleado() {}
    public Empleado(int id) { this.id = id; }

    public getId() { return id; }
    public void setId(int id) { this.id = id; }
    public String getNombre() { return nombre; }
    public void setNombre(String nombre) { this.nombre = nombre; }
    public long getSueldo() { return sueldo; }
    public void setSueldo(long sueldo) { this.sueldo; }
}
```



# Entity Manager (1)

- Se encarga de gestionar las clases y objetos persistentes
- Se define como una interfaz
- Se configura con los parámetros definidos en una *unidad de persistencia* (fichero XML)
- Se obtiene a partir de un EntityManagerFactory (en Java SE):

```
EntityManagerFactory emf =  
    Persistence.createEntityManagerFactory("NombreUP");  
EntityManager em = emf.createEntityManager();
```



## Entity Manager (2)

- Se utiliza inyección de dependencias para obtener el gestor de entidades

```
@PersistenceContext(name="NombrePU")  
private EntityManager em;
```



# Operaciones sobre entidades (1)

- Hacer persistente una entidad

```
Empleado emp = new Empleado(158);  
em.persist(emp);
```

- Modificar una entidad

```
emp.setNombre("Isabel");  
emp.setSueldo(40000);
```

- Buscar una entidad

```
emp = em.find(158);
```



## Operaciones sobre entidades (2)

- Borrar una entidad

```
emp = em.find(158);  
em.remove(emp)
```

- Consultas SQL

```
Query query = em.createQuery("SELECT e FROM Empleado e");  
Collection<Empleado> = query.getResultList();
```



# Demostración: Saludo con Mensajes

- Creación de la unidad de persistencia
- Implementación de la entidad Mensaje
- Modificación de la aplicación cliente para poder añadir mensajes
- Examen de la tabla y la base de datos



# ¿Preguntas?