

Session Factory

```
<session-factory>
<property name="connection.driver_class" >com.mysql.jdbc.Driver</property>
"connection.url" jdbc:mysql://localhost/hibernaterelaciones
"connection.username" root
"connection.pool_size" 1
"dialect" org.hibernate.dialect.MySQL5Dialect
"hbm2ddl.auto" create-drop
<mapping class="paquete.clase" />
</session-factory>
```

Anotaciones

@Entity

@Table(name="contactos") //si la tabla no se llama igual que la clase

```
public class Contacto implements Serializable{
```

```
    @Id
```

```
    @GeneratedValue(strategy=GenerationType.IDENTITY)
```

```
    private long id;
```

```
    @Column(name="e_mail") //si la columna no se llama igual que el campo
```

```
    private String email;
```

```
    @ManyToMany(cascade = CascadeType.ALL, fetch = FetchType.EAGER) //M a M, hacer todo en cascada
```

```
    @OneToOne(cascade = {CascadeType.PERSIST, CascadeType.REMOVE}) //1 a 1, persistencia en cascada,
    borrar en cascada
```

```
@ManyToOne //M a 1
```

```
    private List<Materia> materias = new ArrayList<>();
```

```
}
```

```
Session sesion = HibernateUtil.getSessionFactory().openSession();
```

```
    sesion.beginTransaction(); //para transacciones en cadena
```

```
    sesion.save(...);
```

```
    sesion.delete(...);
```

```
    sesion.persist(...);
```

```
    sesion.getTransaction().commit();
```

```
    sesion.close();
```

```
Session sesion = HibernateUtil.getSessionFactory().openSession();
```

```
Transaction tx = sesion.beginTransaction();
```

```
sesion.persist/update/delete //hacer lo que sea
```

```
tx.commit(); //para una única transacción
```