

CURSO DE HIBERNATE Y JPA

EJERCICIO

PERSISTENCIA EN CASCADA CON HIBERNATE/JPA



Experiencia y Conocimiento para tu vida

CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

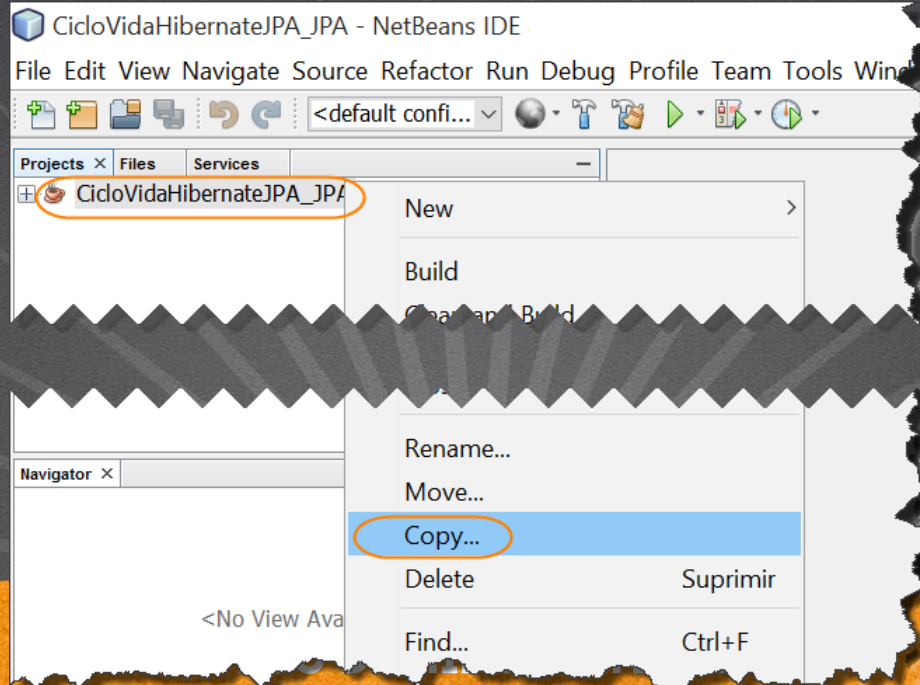
OBJETIVO DEL EJERCICIO

Crear un proyecto para poner en práctica la persistencia en cascada utilizando Hibernate y JPA. Al finalizar deberemos observar lo siguiente:

```
Output - Run (PersistenciaCascadaHibernateJPA) X
16:16:47 [main] DEBUG org.hibernate.SQL - insert into contacto (celular, deleted, email, email2, extension, oficina, telefono, version) values (?, ?, ?, ?, ?, ?, ?, ?)
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [0]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [VARCHAR] - [contacto@mail.com]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [7] as [VARCHAR] - [55717189]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [8] as [INTEGER] - [0]
16:16:47 [main] DEBUG org.hibernate.id.IdentifierGeneratorHelper - Natively generated identity: 24
16:16:47 [main] DEBUG org.hibernate.SQL - insert into domicilio (calle, cp, deleted, estado, municipio, no_externo, no_interno, pais, version) values (?, ?, ?, ?, ?, ?, ?, ?, ?)
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [Allende]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [INTEGER] - [0]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [VARCHAR] - [115]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [7] as [VARCHAR] - [A-101]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [8] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [9] as [INTEGER] - [0]
16:16:47 [main] DEBUG org.hibernate.id.IdentifierGeneratorHelper - Natively generated identity: 23
16:16:47 [main] DEBUG org.hibernate.SQL - insert into alumno (apellido_materno, apellido_paterno, id_contacto, deleted, id_domicilio, fecha_nacimiento, genero, nombre, id_usuario, version) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [Garcia]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [VARCHAR] - [Martinez]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [INTEGER] - [24]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [INTEGER] - [0]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [INTEGER] - [23]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [TIMESTAMP] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [7] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [8] as [VARCHAR] - [Martha]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [9] as [INTEGER] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [10] as [INTEGER] - [0]
16:16:47 [main] DEBUG org.hibernate.id.IdentifierGeneratorHelper - Natively generated identity: 17
Alumno insertado: Alumno{idAlumno=17, nombre=Martha, apellidoPaterno=Martinez, apellidoMaterno=Garcia, genero=null, fechaNacimiento=null, version=0, deleted=0, contacto=Contacto{idC
```

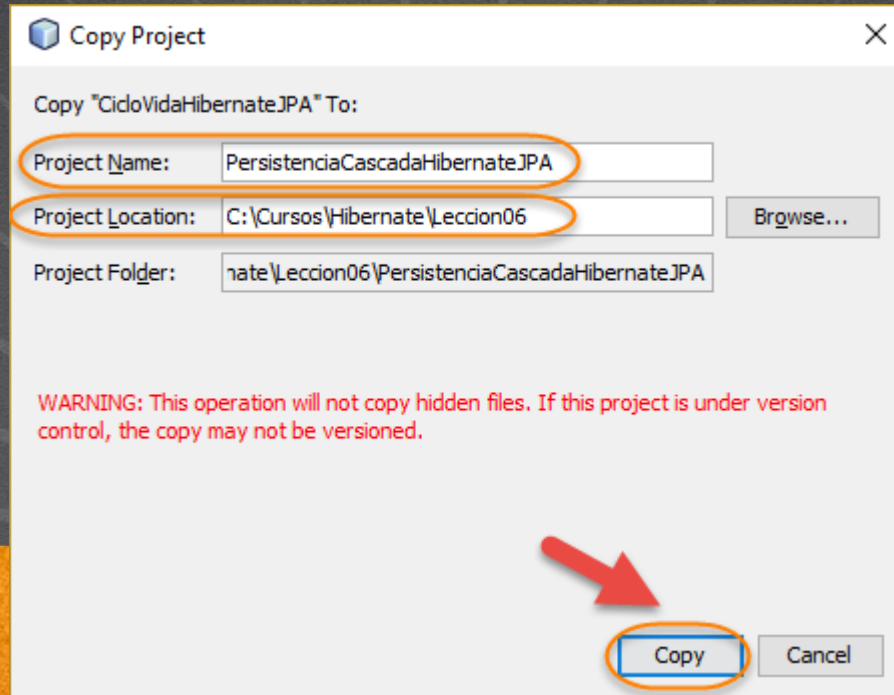

PASO 1. CREAMOS EL PROYECTO

Copiamos y pegamos el proyecto anterior:



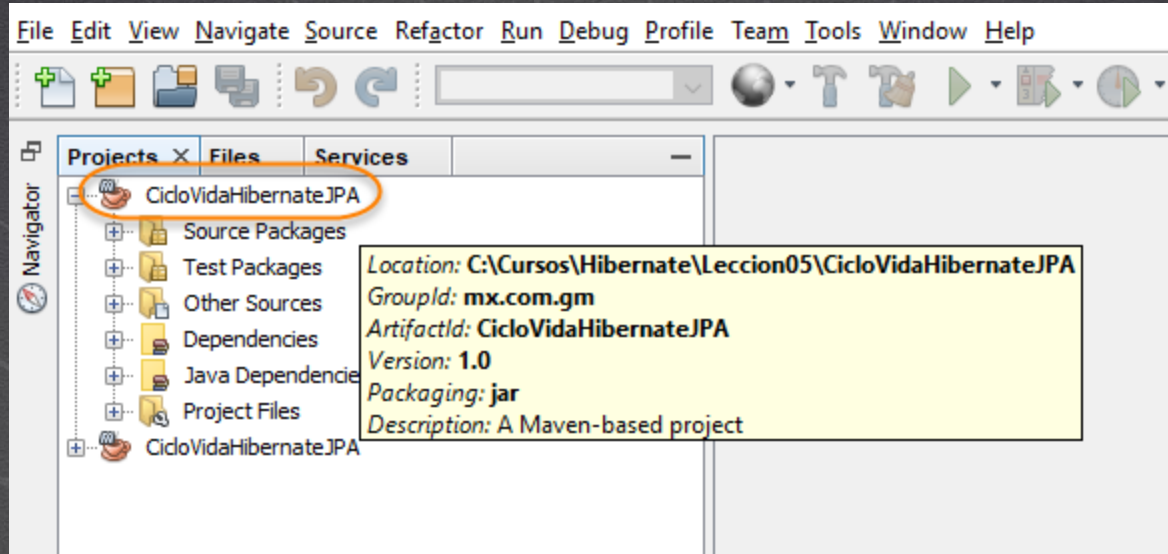
PASO 1. CREAMOS EL PROYECTO

Copiamos y pegamos el proyecto anterior, y lo renombramos a PersistenciaCascadaHibernateJPA:



PASO 2. CERRAMOS EL PROYECTO

Ubicamos el proyecto que vamos a cerrar:

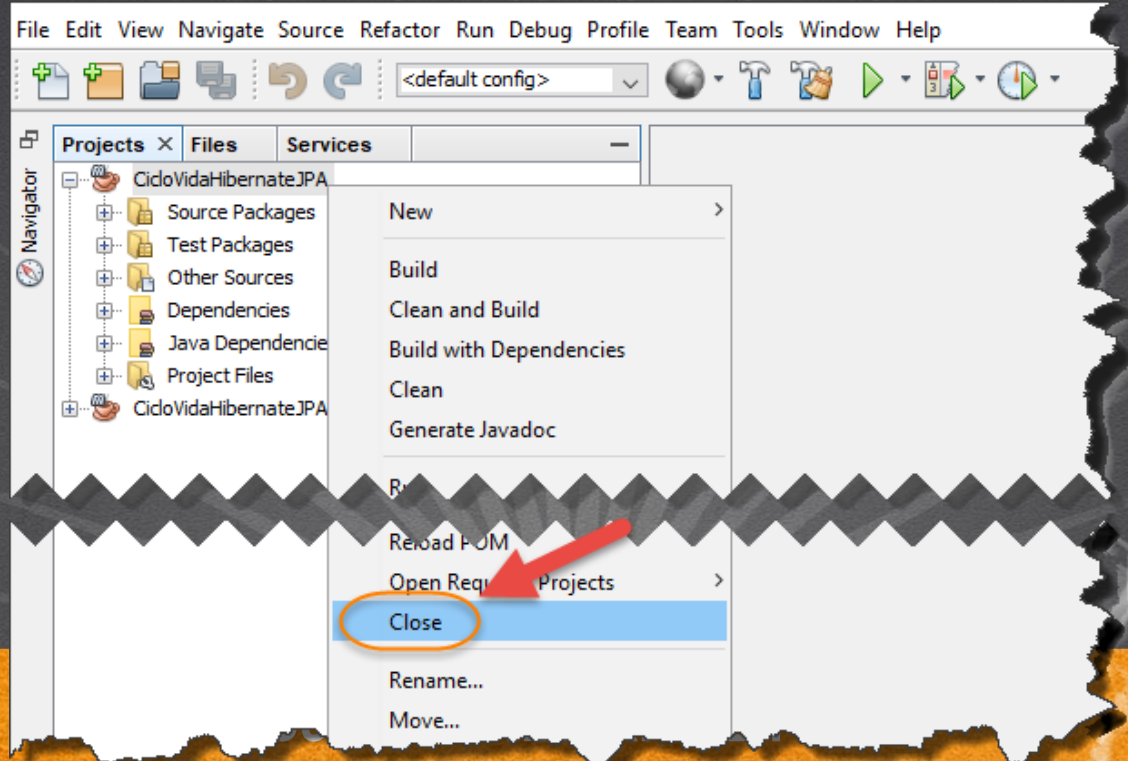


CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

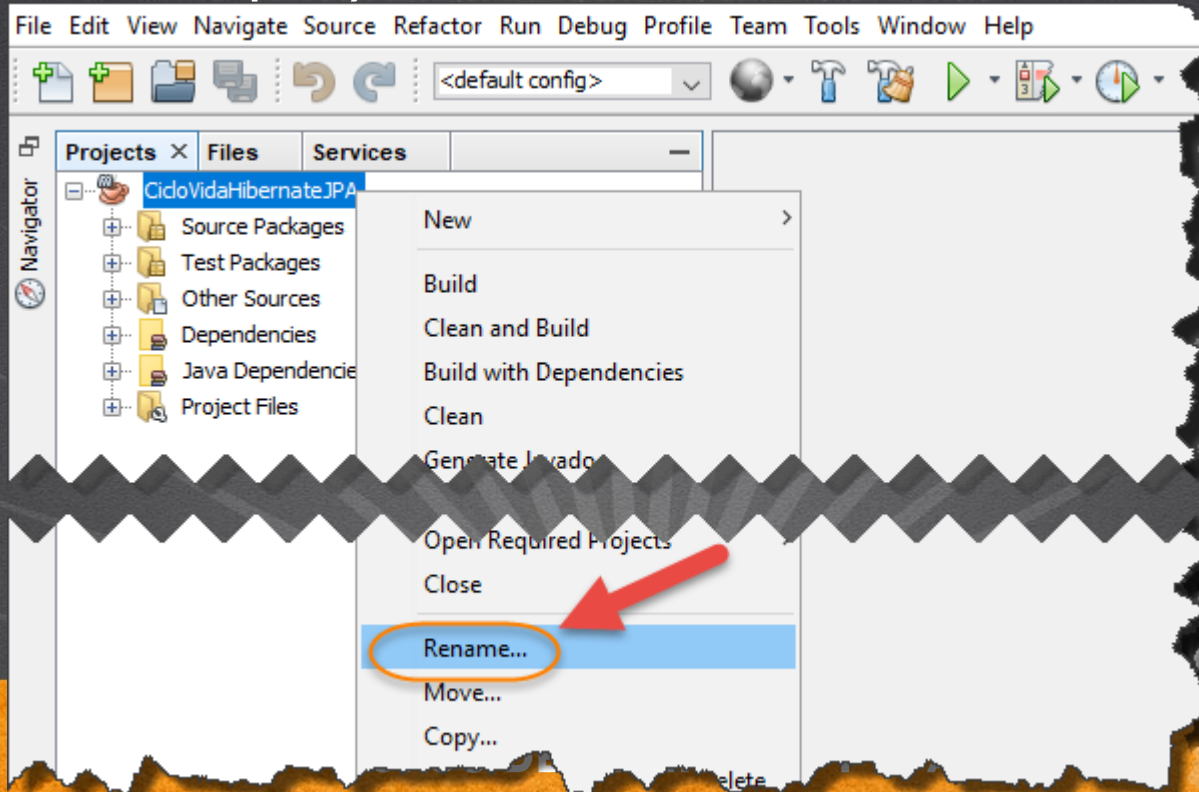
PASO 2. CERRAMOS EL PROYECTO

Cerramos el proyecto que ya no utilizaremos:



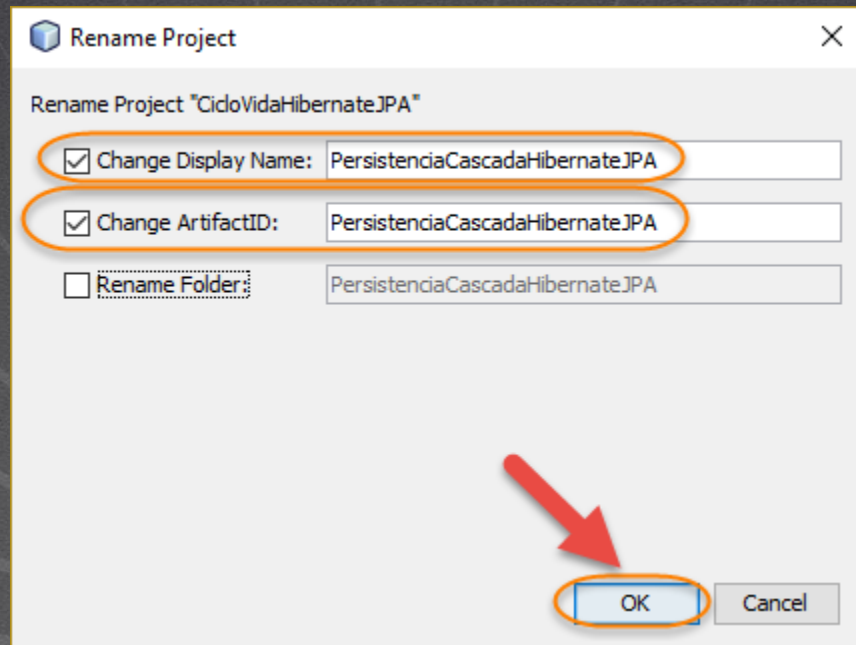
PASO 3. RENOMBRAMOS EL PROYECTO

Renombramos el proyecto:



PASO 3. RENOMBAMOS EL PROYECTO

Renombramos el proyecto:



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 4. CREAMOS UNA CLASE

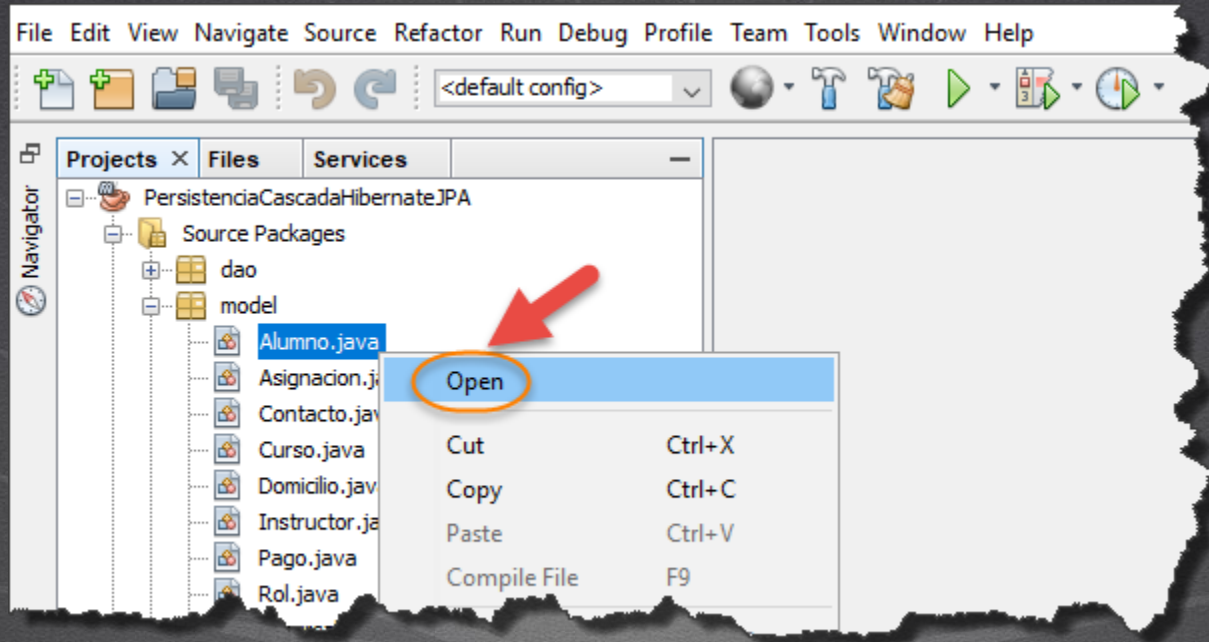
Modificamos la clase `model.Alumno.java` para agregar el concepto de persistencia en cascada en el atributo de contacto y domicilio.

Es decir que al guardar un objeto de tipo `Alumno`, también guardará en automático los objetos `contacto` y `domicilio` relacionados.

Veamos cómo queda nuestra clase de `Alumno.java`:

PASO 4. CREAMOS UNA CLASE

Modificamos la clase Alumno.java:



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Dar click para ir al código

```
package model;

import java.io.Serializable;
import java.util.*;
import javax.persistence.*;

@Entity
@Table(name = "alumno")
@NamedQueries({
    @NamedQuery(name = "Alumno.findAll", query = "SELECT a FROM Alumno a")})
public class Alumno implements Serializable {

    private static final long serialVersionUID = 1L;
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Basic(optional = false)
    @Column(name = "id_alumno")
    private Integer idAlumno;

    @Basic(optional = false)
    @Column(name = "nombre")
    private String nombre;

    @Basic(optional = false)
    @Column(name = "apellido_paterno")
    private String apellidoPaterno;
```


PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Dar click para ir al código

```
@Column(name = "apellido_materno")
private String apellidoMaterno;

@Column(name = "genero")
private String genero;

@Column(name = "fecha_nacimiento")
@Temporal(TemporalType.TIMESTAMP)
private Date fechaNacimiento;

@Basic(optional = false)
@Column(name = "version")
private int version;

@Basic(optional = false)
@Column(name = "deleted")
private int deleted;

@OneToMany(mappedBy = "alumno")
private List<Asignacion> asignacionList;

@JoinColumn(name = "id_contacto", referencedColumnName = "id_contacto")
@ManyToOne(cascade = CascadeType.ALL)
private Contacto contacto;

@JoinColumn(name = "id_domicilio", referencedColumnName = "id_domicilio")
@ManyToOne(cascade = CascadeType.ALL)
private Domicilio domicilio;
```

PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Dar click para ir al código

```
@JoinColumn(name = "id_usuario", referencedColumnName = "id_usuario")
@ManyToOne
private Usuario usuario;

public Alumno() {
}

public Alumno(Integer idAlumno) {
    this.idAlumno = idAlumno;
}

public Alumno(Integer idAlumno, String nombre, String apellidoPaterno, int version, int deleted) {
    this.idAlumno = idAlumno;
    this.nombre = nombre;
    this.apellidoPaterno = apellidoPaterno;
    this.version = version;
    this.deleted = deleted;
}

public Integer getIdAlumno() {
    return idAlumno;
}

public void setIdAlumno(Integer idAlumno) {
    this.idAlumno = idAlumno;
}
```

PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Dar click para ir al código

```
public String getNombre() {  
    return nombre;  
}  
  
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}  
  
public String getApellidoPaterno() {  
    return apellidoPaterno;  
}  
  
public void setApellidoPaterno(String apellidoPaterno) {  
    this.apellidoPaterno = apellidoPaterno;  
}  
  
public String getApellidoMaterno() {  
    return apellidoMaterno;  
}  
  
public void setApellidoMaterno(String apellidoMaterno) {  
    this.apellidoMaterno = apellidoMaterno;  
}  
  
public String getGenero() {  
    return genero;  
}
```


PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Dar click para ir al código

```
public void setGenero(String genero) {
    this.genero = genero;
}

public Date getFechaNacimiento() {
    return fechaNacimiento;
}

public void setFechaNacimiento(Date fechaNacimiento) {
    this.fechaNacimiento = fechaNacimiento;
}

public int getVersion() {
    return version;
}

public void setVersion(int version) {
    this.version = version;
}

public int getDeleted() {
    return deleted;
}

public void setDeleted(int deleted) {
    this.deleted = deleted;
}
```

PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Dar click para ir al código

```
public List<Asignacion> getAsignacionList() {  
    return asignacionList;  
}  
  
public void setAsignacionList(List<Asignacion> asignacionList) {  
    this.asignacionList = asignacionList;  
}  
  
public Contacto getContacto() {  
    return contacto;  
}  
  
public void setContacto(Contacto contacto) {  
    this.contacto = contacto;  
}  
  
public Domicilio getDomicilio() {  
    return domicilio;  
}  
  
public void setDomicilio(Domicilio domicilio) {  
    this.domicilio = domicilio;  
}  
  
public Usuario getUsuario() {  
    return usuario;  
}
```

PASO 4. MODIFICAMOS EL CÓDIGO

Archivo Alumno.java:

Dar click para ir al código

```
public void setUsuario(Usuario usuario) {
    this.usuario = usuario;
}

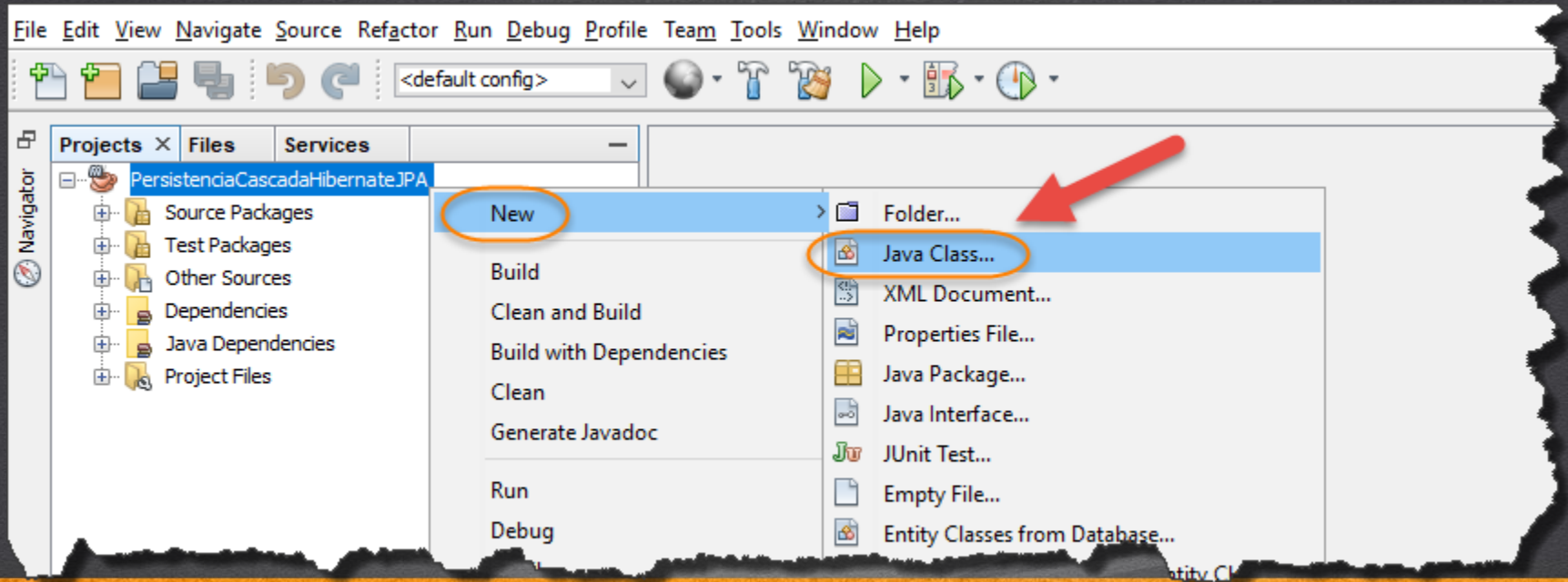
@Override
public int hashCode() {
    int hash = 0;
    hash += (idAlumno != null ? idAlumno.hashCode() : 0);
    return hash;
}

@Override
public boolean equals(Object object) {
    if (!(object instanceof Alumno)) {
        return false;
    }
    Alumno other = (Alumno) object;
    if ((this.idAlumno == null && other.idAlumno != null) || (this.idAlumno != null && !this.idAlumno.equals(other.idAlumno)))
    {
        return false;
    }
    return true;
}

@Override
public String toString() {
    return "Alumno{" + "idAlumno=" + idAlumno + ", nombre=" + nombre + ", apellidoPaterno=" + apellidoPaterno + ",
apellidoMaterno=" + apellidoMaterno + ", genero=" + genero + ", fechaNacimiento=" + fechaNacimiento + ", version=" + version + ",
deleted=" + deleted + ", contacto=" + contacto + ", domicilio=" + domicilio + ", usuario=" + usuario + '}';
}
```


PASO 5. CREAMOS UNA CLASE

Creamos una clase PersistenciaCascadaTest.java:



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 5. CREAMOS UNA CLASE

Creamos una clase PersistenciaCascadaTest.java:

New Java Class

Steps

1. Choose File Type
2. **Name and Location**

Name and Location

Class Name:

Project:

Location:

Package:

Created File:

< Back Next > **Finish** Cancel Help

PASO 6. MODIFICAMOS EL CÓDIGO

Archivo PersistenciaCascadaTest.java

Dar click para ir al código

```
package test.persistenciacascada;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.Persistence;
import model.Contacto;
import model.Domicilio;
import model.Alumno;

public class PersistenciaCascadaTest {

    public static void main(String[] args) {
        /*Utilizamos la Unidad de Persistencia de JPA*/
        EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = fabrica.createEntityManager();

        // Creamos un objeto Domicilio
        Domicilio domicilio = new Domicilio();
        domicilio.setCalle("Allende");
        domicilio.setNoExterno("115");
        domicilio.setNoInterno("A-101");
        domicilio.setVersion(0);
        domicilio.setDeleted(0);
    }
}
```


PASO 6. MODIFICAMOS EL CÓDIGO

[Archivo PersistenciaCascadaTest.java](#)

Dar click para ir al código

```
// Creamos un objeto Contacto
Contacto contacto = new Contacto();
contacto.setTelefono("55717189");
contacto.setEmail("contacto@mail.com");
contacto.setVersion(0);
contacto.setDeleted(0);

// Creamos un objeto Alumno (tiene varios valores requeridos, ej.
// id_domicilio y id_contacto)
Alumno alumno = new Alumno();
alumno.setNombre("Martha");
alumno.setApellidoPaterno("Martinez");
alumno.setApellidoMaterno("Garcia");
//Agregamos las relaciones y su persistencia en cascada
alumno.setDomicilio(domicilio);
alumno.setContacto(contacto);
alumno.setVersion(0);
alumno.setDeleted(0);
```

PASO 6. MODIFICAMOS EL CÓDIGO

[Archivo PersistenciaCascadaTest.java](#)

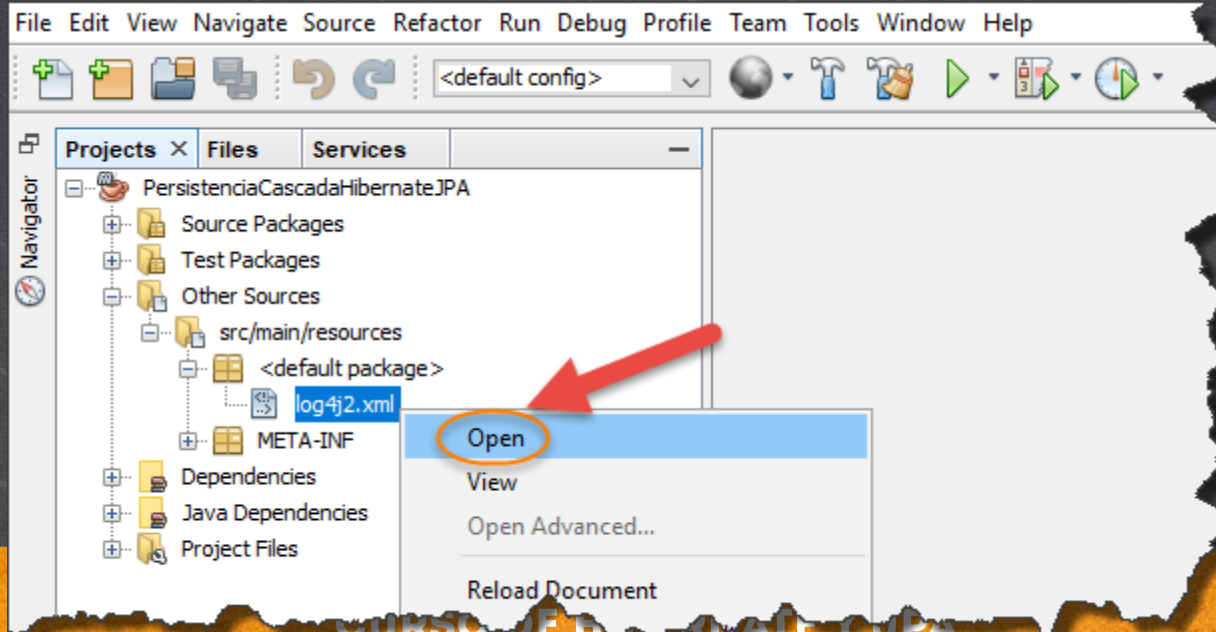
Dar click para ir al código

```
try {
    em.getTransaction().begin();
    //Solo persistimos el alumno, y se agregan en automatico
    //las relaciones asociadas marcadas como persistencia en cascada
    em.persist(alumno);
    em.getTransaction().commit();
} catch (Exception e) {
    em.getTransaction().rollback();
    e.printStackTrace();
} finally {
    if (em != null) {
        em.close();
    }
}

// Objetos insertados
System.out.println("Alumno insertado:" + alumno);
}
```

PASO 7. MODIFICAMOS EL ARCHIVO LOG4J2.XML

Si no existe el archivo log4j2 lo creamos en la ruta que se observa, si ya existe lo modificamos:



PASO 7. MODIFICAMOS EL CÓDIGO

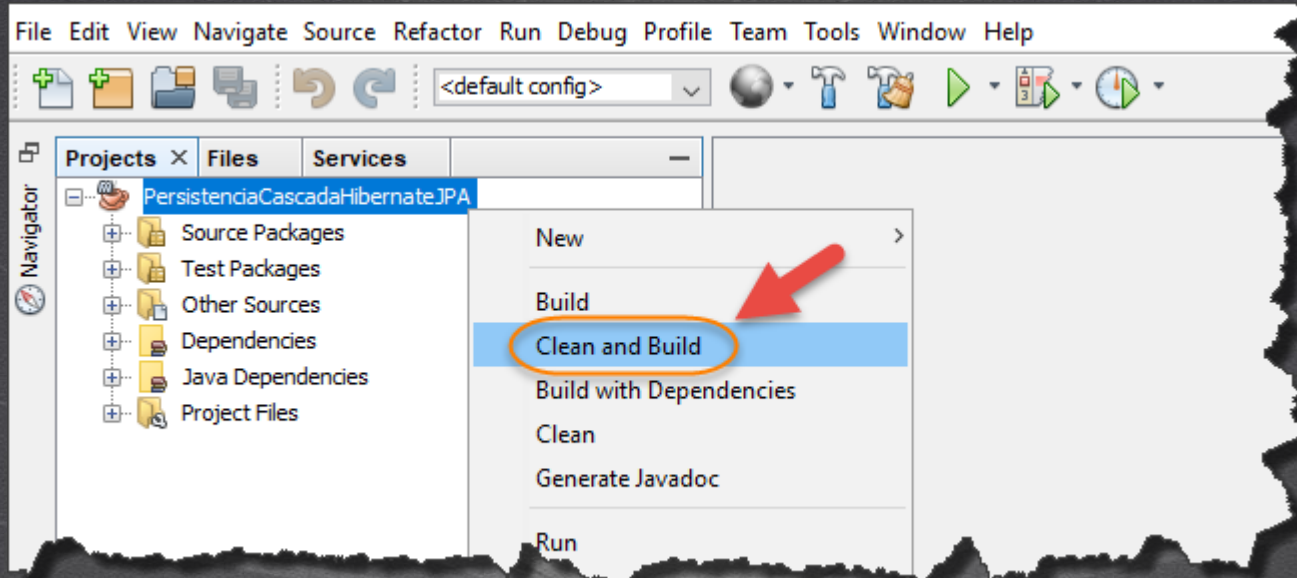
Archivo log4j2.xml:

Dar click para ir al código

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
  <Appenders>
    <Console name="Console" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{HH:mm:ss} [%t] %-5level %logger{36} - %msg%n" />
    </Console>
  </Appenders>
  <Loggers>
    <Logger name="org.hibernate.SQL" level="debug" additivity="false">
      <AppenderRef ref="Console"/>
    </Logger>
    <Logger name="org.hibernate.type.descriptor.sql" level="trace" additivity="false">
      <AppenderRef ref="Console" />
    </Logger>
    <Logger name="org.hibernate.id" level="debug" additivity="false">
      <AppenderRef ref="Console" />
    </Logger>
    <Root level="info">
      <AppenderRef ref="Console" />
    </Root>
  </Loggers>
</Configuration>
```


PASO 8. HACEMOS CLEAN & BUILD

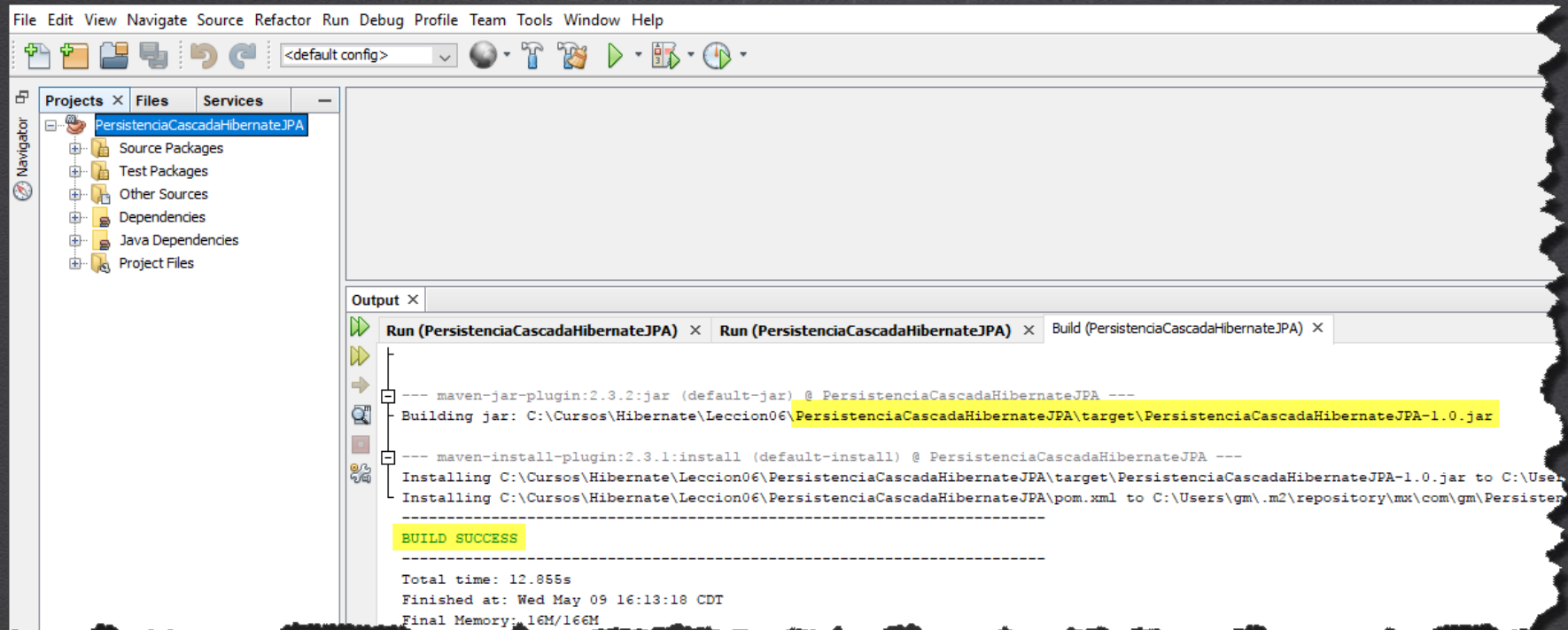
Hacemos Clean & Build para tener las últimas versiones de cada archivo:



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

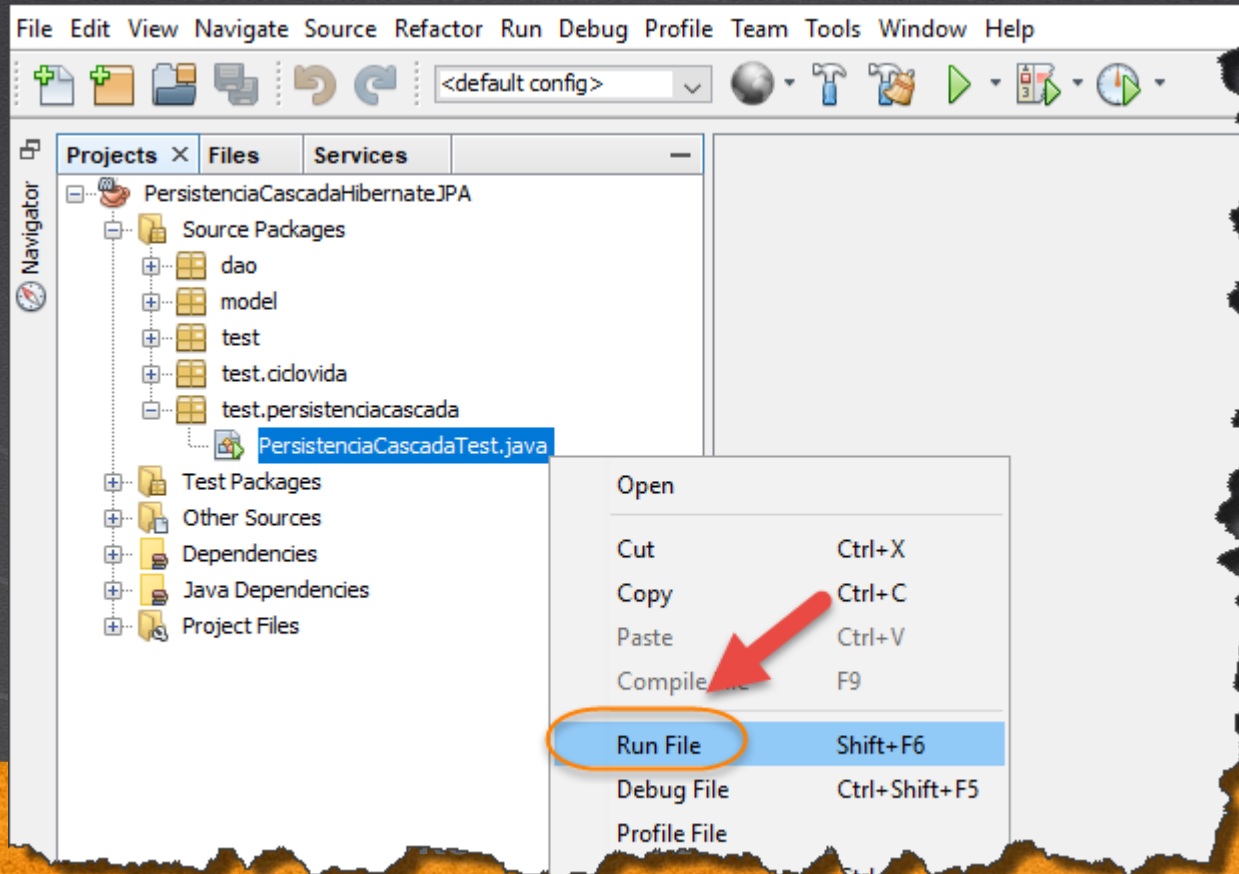
PASO 8. HACEMOS CLEAN & BUILD



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 9. EJECUTAMOS EL CÓDIGO



PASO 9. EJECUTAMOS EL CÓDIGO

Output - Run (PersistenciaCascadaHibernateJPA) x

```
16:16:47 [main] DEBUG org.hibernate.SQL - insert into contacto (celular, deleted, email, email2, extension, oficina, telefono, version) values (?, ?, ?, ?, ?, ?, ?, ?)
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [0]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [VARCHAR] - [contacto@mail.com]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [7] as [VARCHAR] - [55717189]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [8] as [INTEGER] - [0]
16:16:47 [main] DEBUG org.hibernate.id.IdentifierGeneratorHelper - Natively generated identity: 24
16:16:47 [main] DEBUG org.hibernate.SQL - insert into domicilio (calle, cp, deleted, estado, municipio, no_externo, no_interno, pais, version) values (?, ?, ?, ?, ?, ?, ?, ?, ?)
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [Allende]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [INTEGER] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [INTEGER] - [0]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [VARCHAR] - [115]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [7] as [VARCHAR] - [A-101]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [8] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [9] as [INTEGER] - [0]
16:16:47 [main] DEBUG org.hibernate.id.IdentifierGeneratorHelper - Natively generated identity: 23
16:16:47 [main] DEBUG org.hibernate.SQL - insert into alumno (apellido_materno, apellido_paterno, id_contacto, deleted, id_domicilio, fecha_nacimiento, genero, nombre, id_usuario, version) values (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [1] as [VARCHAR] - [Garcia]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [2] as [VARCHAR] - [Martinez]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [3] as [INTEGER] - [24]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [4] as [INTEGER] - [0]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [5] as [INTEGER] - [23]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [6] as [TIMESTAMP] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [7] as [VARCHAR] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [8] as [VARCHAR] - [Martha]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [9] as [INTEGER] - [null]
16:16:47 [main] TRACE org.hibernate.type.descriptor.sql.BasicBinder - binding parameter [10] as [INTEGER] - [0]
16:16:47 [main] DEBUG org.hibernate.id.IdentifierGeneratorHelper - Natively generated identity: 17
Alumno insertado: Alumno{idAlumno=17, nombre=Martha, apellidoPaterno=Martinez, apellidoMaterno=Garcia, genero=null, fechaNacimiento=null, version=0, deleted=0, contacto=Contacto{idC
```


CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos visto como persistir en cascada un objeto y sus relaciones con Hibernate/JPA.
- Al configurar el objeto de Entidad, y sus relaciones es posible guardar de manera automática el objeto de Entidad junto con sus relaciones, según hayamos especificado en la configuración del objeto.
- También utilizamos el API de log4j para poder observar los SQL que se ejecutan al momento de persistir en cascada los objetos de entidad relacionados.

CURSO ONLINE

HIBERNATE & JPA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx