Conceptos en un ejemplo práctico.

Modelo de capaz básicas

Lógica

Persistencia

IGU(vista)

3 paquetes.

Esto es la lógica el main!!! Y desde aquí se llama al IGU

package com.sertecdev.peluqueriacanina;

import com.sertecdev.peluqueriacanina.igu.Principal;

public class PeluqueriaCanina {

public static void main(String[] args) {

Principal p = new Principal();

p.setVisible(true);

p.setLocationRelativeTo(null);

}

}

Abrir con un botón una ventana nueva la ventana es una clase y se llama CargaDatos y creo una instancia de ella.

private void btnCargarDatosActionPerformed(java.awt.event.ActionEvent evt) {

CargaDatos cd = new CargaDatos();

cd.setVisible(true);

cd.setLocationRelativeTo(null);

}

Lógica del botón limpiar

private void btnLimpiarActionPerformed(java.awt.event.ActionEvent evt) {

txtNombre.setText("");

txtRaza.setText("");

txtNombreDuenio.setText("");

txtCelDuenio.setText("");

txtAreaObser.setText("");

cboxAlergico.setSelectedIndex(0);

cboxAtencionEsp.setSelectedIndex(0);

}

En la lógica va el modelado:

**Dueño**

package com.sertecdev.peluqueriacanina.logica;

public class Duenio {

private int idDuenio;

private String nombreDuenio;

private String celDuenio;

public Duenio(int idDuenio, String nombreDuenio, String celDuenio) {

this.idDuenio = idDuenio;

this.nombreDuenio = nombreDuenio;

this.celDuenio = celDuenio;

}

public Duenio() {

}

public int getIdDuenio() {

return idDuenio;

}

public void setIdDuenio(int idDuenio) {

this.idDuenio = idDuenio;

}

public String getNombreDuenio() {

return nombreDuenio;

}

public void setNombreDuenio(String nombreDuenio) {

this.nombreDuenio = nombreDuenio;

}

public String getCelDuenio() {

return celDuenio;

}

public void setCelDuenio(String celDuenio) {

this.celDuenio = celDuenio;

}

}

package com.sertecdev.peluqueriacanina.logica;

public class Mascota {

private int munCliente;

private String nombreMascota;

private String raza;

private String alergico;

private String atencionEspecial;

private String observaciones;

private Duenio duenio;

**Mascota**

public Mascota() {

}

public Mascota(int munCliente, String nombreMascota, String raza, String alergico, String atencionEspecial, String observaciones, Duenio duenio) {

this.munCliente = munCliente;

this.nombreMascota = nombreMascota;

this.raza = raza;

this.alergico = alergico;

this.atencionEspecial = atencionEspecial;

this.observaciones = observaciones;

this.duenio = duenio;

}

public int getMunCliente() {

return munCliente;

}

public void setMunCliente(int munCliente) {

this.munCliente = munCliente;

}

public String getNombreMascota() {

return nombreMascota;

}

public void setNombreMascota(String nombreMascota) {

this.nombreMascota = nombreMascota;

}

public String getRaza() {

return raza;

}

public void setRaza(String raza) {

this.raza = raza;

}

public String getAlergico() {

return alergico;

}

public void setAlergico(String alergico) {

this.alergico = alergico;

}

public String getAtencionEspecial() {

return atencionEspecial;

}

public void setAtencionEspecial(String atencionEspecial) {

this.atencionEspecial = atencionEspecial;

}

public String getObservaciones() {

return observaciones;

}

public void setObservaciones(String observaciones) {

this.observaciones = observaciones;

}

public Duenio getDuenio() {

return duenio;

}

public void setDuenio(Duenio duenio) {

this.duenio = duenio;

}

}

Creo clases controladoras

En la lógica y la persistencia.

Creo la conexión con la base de datos -> services ->Databases, clic derecho new connection, elijo mysql y su drivers (ya esta configurado en esta pc) en database coloco el nombre de la base de datos y la contraseña.

Por defecto se genera :

jdbc:mysql://localhost:3306/peluqueria\_canina?zeroDateTimeBehavior=CONVERT\_TO\_NULL

pero lo cambiamos para no tener problemas por el horario del server.

jdbc:mysql://localhost:3306/peluqueria\_canina?ServerTimezone=UTC

next y finish

se crea la conexión a la base de datos, clic derecho y desconectamos

dentro del paquete persistencia creamos una unidad de persistencia

new Persistence Unit

cambiamos el nombre por defecto!

¡VER SIEMPRE QUE ESTE EN CREATE TABLE COMO STRATEGY

Se crea lo unidad de persistencia, y agregamos las entidades, para ello se realiza el mapeo primero.

Mapeo

package com.sertecdev.peluqueriacanina.logica;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.OneToOne;

@Entity

public class Mascota {

@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE)

private int munCliente;

private String nombreMascota;

private String raza;

private String alergico;

private String atencionEspecial;

private String observaciones;

@OneToOne

private Duenio duenio;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

@Entity

public class Duenio {

@Id

@GeneratedValue(strategy = GenerationType.SEQUENCE)

private int idDuenio;

private String nombreDuenio;

private String celDuenio;

ahora agregamos las entidades a la unidad de persistencia

add class🡪 y ahí estarán para agregar las clases mapeadas

creamos ahora las controladoras de persistencia , dentro del paquete de persistencia

new JPA controller from entity seleccione entidades y agregue, next elegimos el paquete y finalizamos.

Agregamos un constructor a los jpa:

public MascotaJpaController() {

emf = Persistence.createEntityManagerFactory("PeluCaninaPU");

}

En general se crea este constructor:

Public NombreDeLaClaseJpaController() {

emf = Persistence.createEntityManagerFactory("NombreDeLaUnidadDePersistencia");

}

Importante: agregar Mysql controlador!! Con Maven

Dependencies 🡪 add dependencias🡪 busco mysql connector y descargar

Mysql: mysql-connector-java-8.0.30 jar

Creo las capas controladoraPersistencia(persistencia) y controladora(lógica)

package com.sertecdev.peluqueriacanina.persistencia;

public class ControladoraPersistencia {

DuenioJpaController duenioJpa = new DuenioJpaController();

MascotaJpaController mascoJpa= new MascotaJpaController();

}

package com.sertecdev.peluqueriacanina.logica;

import com.sertecdev.peluqueriacanina.persistencia.ControladoraPersistencia;

public class Controladora {

ControladoraPersistencia cp= new ControladoraPersistencia();

}

Boton guardar lógica interna:

private void btnGuardarActionPerformed(java.awt.event.ActionEvent evt) {

String nombreMasco = txtNombre.getText();

String raza= txtRaza.getText();

String observaciones= txtAreaObser.getText();

String nombreDuenio = txtNombreDuenio.getText();

String celDuenio = txtCelDuenio.getText();

String alergias = (String) cboxAlergico.getSelectedItem();

String atencionEs = (String) cboxAtencionEsp.getSelectedItem();

control.guardar(nombreMasco,raza,observaciones,nombreDuenio,celDuenio,alergias,atencionEs);

}

Creamos la función guardar() en la controladora:

public class Controladora {

ControladoraPersistencia cp= new ControladoraPersistencia();

public void guardar(String nombreMasco, String raza, String observaciones,

String nombreDuenio, String celDuenio, String alergias, String atencionEs) {

Duenio duenio= new Duenio();

duenio.setNombreDuenio(nombreDuenio);

duenio.setCelDuenio(celDuenio);

Mascota mascota= new Mascota();

mascota.setNombreMascota(nombreMasco);

mascota.setRaza(raza);

mascota.setObservaciones(observaciones);

mascota.setAlergico(alergias);

mascota.setAtencionEspecial(atencionEs);

mascota.setDuenio(duenio);

cp.guardar(duenio,mascota);

}

}

Luego la capa de persistencia:

public class ControladoraPersistencia {

DuenioJpaController duenioJpa = new DuenioJpaController();

MascotaJpaController mascoJpa= new MascotaJpaController();

public void guardar(Duenio duenio, Mascota mascota) {

duenioJpa.create(duenio);

mascoJpa.create(mascota);

}

}

Fuimos creando el método guardar en las capas de la aplicación