/\*

\* Pueda usar esta plantilla para la carga del reto a iMaster

\* Copie las clases de los paquetes Modelo, Vista, Controlador y Util

\* No incluya los import de los archivos .java solo las clases

\*/

// Importaciones necesarias en iMaster

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.PreparedStatement;

import java.util.ArrayList;

// Util (No modificar)

class JDBCUtilities {

private static final String DATABASE\_LOCATION = "ProyectosConstruccion.db";

public static Connection getConnection() throws SQLException {

String url = "jdbc:sqlite:"+DATABASE\_LOCATION;

return DriverManager.getConnection(url);

}

}

// Remplace en adelante por las clases de sus archivos .java

// Vista

public class ViewRequerimientos {

public static final ControllerRequerimientos controlador = new ControllerRequerimientos();

public static void requerimiento1() {

try {

ArrayList<Requerimiento\_1> listaRequerimiento\_1;

listaRequerimiento\_1 = controlador.consultarRequerimiento1();

for(Requerimiento\_1 requerimiento\_1: listaRequerimiento\_1){

System.out.println(

requerimiento\_1.getFechaCompra() + " " +

requerimiento\_1.getCodCompra() + " " +

requerimiento\_1.getNombreProveedor() + " " +

requerimiento\_1.getPagado()

);

}

} catch (Exception e){

System.err.println(e);

}

}

public static void requerimiento2() {

try {

ArrayList<Requerimiento\_2> listaRequerimiento\_2;

listaRequerimiento\_2 = controlador.consultarRequerimiento2();

for(Requerimiento\_2 requerimiento\_2: listaRequerimiento\_2){

System.out.println(

requerimiento\_2.getFecIni() + " " +

requerimiento\_2.getNombreCiudad() + " " +

requerimiento\_2.getConstructora() + " " +

requerimiento\_2.getNombreLider() + " " +

requerimiento\_2.getCodigoTipo() + " " +

requerimiento\_2.getEstrato()

);

}

} catch (Exception e){

System.err.println(e);

}

}

public static void requerimiento3() {

try {

ArrayList<Requerimiento\_3> listaRequerimiento\_3;

listaRequerimiento\_3 = controlador.consultarRequerimiento3();

for(Requerimiento\_3 requerimiento\_3: listaRequerimiento\_3){

System.out.println(

requerimiento\_3.getAbrev()

);

}

} catch (Exception e){

System.err.println(e);

}

}

}

// Controlador

public class ControllerRequerimientos {

// Su código

private final RequerimientoDao\_1 requerimientoDao\_1 = new RequerimientoDao\_1();

private final RequerimientoDao\_2 requerimientoDao\_2 = new RequerimientoDao\_2();

private final RequerimientoDao\_3 requerimientoDao\_3 = new RequerimientoDao\_3();

public ArrayList<Requerimiento\_1> consultarRequerimiento1() throws SQLException {

return this.requerimientoDao\_1.requerimiento1();

}

public ArrayList<Requerimiento\_2> consultarRequerimiento2() throws SQLException {

// Su código

return this.requerimientoDao\_2.requerimiento2();

}

public ArrayList<Requerimiento\_3> consultarRequerimiento3() throws SQLException {

// Su código

return this.requerimientoDao\_3.requerimiento3();

}

}

// Modelo

// VO

public class Requerimiento\_1 {

// Operaciones de la clase requerimiento 1 -- su código

//Atributos

private String fechaCompra;

private Integer codCompra;

private String nombreProveedor;

private String pagado;

public Requerimiento\_1(){

}

public Requerimiento\_1(String fechaCompra, Integer codCompra, String nombreProveedor, String pagado){

this.fechaCompra = fechaCompra;

this.codCompra = codCompra;

this.nombreProveedor = nombreProveedor;

this.pagado = pagado;

}

public String getPagado() {

return pagado;

}

public String getFechaCompra() {

return fechaCompra;

}

public void setFechaCompra(String fechaCompra) {

this.fechaCompra = fechaCompra;

}

public Integer getCodCompra() {

return codCompra;

}

public void setCodCompra(Integer codCompra) {

this.codCompra = codCompra;

}

public String getNombreProveedor() {

return nombreProveedor;

}

public void setNombreProveedor(String nombreProveedor) {

this.nombreProveedor = nombreProveedor;

}

public void setPagado(String pagado) {

this.pagado = pagado;

}

}

public class Requerimiento\_2 {

// Operaciones de la clase requerimiento 2 -- su código

// Atributos

private String fecIni;

private String nombreCiudad;

private String constructora;

private String nombreLider;

private Integer codigoTipo;

private Integer estrato;

public Requerimiento\_2(){

}

public Requerimiento\_2(String fecIni, String nombreCiudad, String constructora, String nombreLider, Integer codigoTipo, Integer estrato){

this.fecIni = fecIni;

this.nombreCiudad = nombreCiudad;

this.constructora = constructora;

this.nombreLider = nombreLider;

this.codigoTipo = codigoTipo;

this.estrato = estrato;

}

public String getConstructora() {

return constructora;

}

public void setConstructora(String constructora) {

this.constructora = constructora;

}

public Integer getEstrato() {

return estrato;

}

public void setEstrato(Integer estrato) {

this.estrato = estrato;

}

public String getFecIni() {

return fecIni;

}

public void setFecIni(String fecIni) {

this.fecIni = fecIni;

}

public String getNombreCiudad() {

return nombreCiudad;

}

public void setNombreCiudad(String nombreCiudad) {

this.nombreCiudad = nombreCiudad;

}

public String getNombreLider() {

return nombreLider;

}

public void setNombreLider(String nombreLider) {

this.nombreLider = nombreLider;

}

public Integer getCodigoTipo() {

return codigoTipo;

}

public void setCodigoTipo(Integer codigoTipo) {

this.codigoTipo = codigoTipo;

}

}

public class Requerimiento\_3 {

// Operaciones de la clase requerimiento 3 -- su código

private String abrev;

public Requerimiento\_3(){

}

public Requerimiento\_3(String abrev){

this.setAbrev(abrev);

}

public String getAbrev(){

return this.abrev;

}

public void setAbrev(String abrev){

this.abrev = abrev;

}

}

// DAO

public class RequerimientoDao\_1 {

public ArrayList<Requerimiento\_1> requerimiento1() throws SQLException {

// Su código

ArrayList<Requerimiento\_1> respuesta = new ArrayList<Requerimiento\_1>();

Connection conexion = JDBCUtilities.getConnection();

try {

String consulta = "SELECT c.Fecha as 'Fecha', c.ID\_Compra, c.Proveedor, c.Pagado " +

"FROM Compra c ORDER BY c.ID\_Compra DESC LIMIT 10";

PreparedStatement statement = conexion.prepareStatement(consulta);

ResultSet resultSet = statement.executeQuery();

// Recorrer los registros en los VO especificos

while(resultSet.next()){

Requerimiento\_1 requerimiento\_1 = new Requerimiento\_1();

requerimiento\_1.setFechaCompra(resultSet.getString("Fecha"));

requerimiento\_1.setCodCompra(resultSet.getInt("ID\_Compra"));

requerimiento\_1.setNombreProveedor(resultSet.getString("Proveedor"));

requerimiento\_1.setPagado(resultSet.getString("Pagado"));

respuesta.add(requerimiento\_1);

}

resultSet.close();

statement.close();

} catch (SQLException e) {

System.out.println("Error de consulta SQL Requerimiento 1 -> " + e);

} finally {

if (conexion != null){

conexion.close();

}

}

return respuesta;

}

}

public class RequerimientoDao\_2 {

public ArrayList<Requerimiento\_2> requerimiento2() throws SQLException {

// Su código

ArrayList<Requerimiento\_2> respuesta = new ArrayList<Requerimiento\_2>();

Connection conexion = JDBCUtilities.getConnection();

try {

String consulta = "SELECT p.Fecha\_Inicio, p.Ciudad, p.Constructora, " +

"l.Nombre || ' ' || l.Segundo\_Apellido as 'Nombre Lider', " +

"t.Codigo\_Tipo, t.Estrato " +

"FROM Proyecto p INNER JOIN Lider l ON p.ID\_Lider = l.ID\_Lider " +

"INNER JOIN Tipo t ON p.ID\_Tipo = t.ID\_Tipo " +

"WHERE (p.Fecha\_Inicio >= '2019-09-01' AND p.Fecha\_Inicio <= '2019-09-09' AND p.Ciudad = 'Pereira')";

PreparedStatement statement = conexion.prepareStatement(consulta);

ResultSet resultSet = statement.executeQuery();

while(resultSet.next()){

Requerimiento\_2 requerimiento\_2 = new Requerimiento\_2(

resultSet.getString("Fecha\_Inicio"),

resultSet.getString("Ciudad"),

resultSet.getString("Constructora"),

resultSet.getString("Nombre Lider"),

resultSet.getInt("Codigo\_Tipo"),

resultSet.getInt("Estrato")

);

respuesta.add(requerimiento\_2);

}

resultSet.close();

statement.close();

} catch (SQLException e) {

System.err.println("Error en la consulta SQL Requerimiento 2 -> " + e);

} finally {

if (conexion != null){

conexion.close();

}

}

return respuesta;

}

}

public class RequerimientoDao\_3 {

public ArrayList<Requerimiento\_3> requerimiento3() throws SQLException {

// Su código

ArrayList<Requerimiento\_3> respuesta = new ArrayList<Requerimiento\_3>();

Connection conexion = JDBCUtilities.getConnection();

try {

String consulta = "SELECT SUBSTR(l.Nombre,1,3) || SUBSTR(l.Primer\_Apellido,1,3) "

+ "|| SUBSTR(l.Segundo\_Apellido,1,3) AS 'Abrev.' "

+ "FROM Lider l";

PreparedStatement statement = conexion.prepareStatement(consulta);

ResultSet rs = statement.executeQuery();

while (rs.next()) {

Requerimiento\_3 requerimiento\_3 = new Requerimiento\_3(rs.getString("Abrev."));

respuesta.add(requerimiento\_3);

}

rs.close();

statement.close();

} catch (SQLException e) {

System.err.println("Error en la consulta SQL Requerimiento\_3 -> " + e);

} finally {

if (conexion != null) {

conexion.close();

}

}

return respuesta;

}

}