

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 2

5ta. práctica (tipo b)
(Primer Semestre 2023)

Indicaciones Generales:

- Tiempo estimado: 1h 50 minutos
- Se les recuerda que, de acuerdo al reglamento disciplinario de nuestra institución, constituye una falta grave copiar del trabajo realizado por otro estudiante o cometer plagio para el desarrollo de esta práctica.
- Puede hacer uso del entorno de desarrollo integrado de NETBEANS (a excepción de la pregunta Nro. 2 y 3, las cuales deben realizarse mediante línea de comandos y uso de editor de texto: Notepad++ o Sublime).
- Está permitido el uso de apuntes de clase, diapositivas, ejercicios de clase y código fuente. (Debe descargarlos antes de iniciar con la solución del enunciado).
- Está permitido el uso de Internet (únicamente para consultar páginas oficiales de Microsoft, Oracle y PAIDEIA). No obstante, está prohibida toda forma de comunicación con otros estudiantes o terceros.

PARTE PRÁCTICA (20 puntos)

PUEDE UTILIZAR MATERIAL DE CONSULTA.

Antes de comenzar el laboratorio, descargue todos los proyectos, apuntes, diapositivas que utilizará.

Se considerará en la calificación el uso de buenas prácticas de programación (aquellas vistas en clase).

PREGUNTA 1: Creación de un motor de base de datos MySQL en AWS Academy (3 puntos)

Se le solicita ingresar al servicio de AWS Academy (<https://awsacademy.instructure.com/login/canvas>) y crear una instancia o motor de base de datos MySQL con los siguientes parámetros:

- Método de creación de base de datos: **Creación estándar**
- Tipo de motor: **MySQL**
- Edición: **Comunidad de MySQL**
- Versión del motor: **MySQL 8.0.32**
- Plantilla: **Capa gratuita**
- Identificador del clúster de base de datos: **lab05-lengpro2-2023-1**
- Nombre de usuario maestro: **admin**
- Contraseña maestra: **inf282lp220231**
- Clase de instancia de base de datos: **db.t2.micro (1 CPU, 1 GB RAM)**
- Capacidad de disco duro para la BD: **20 GB SSD (disco sólido) sin posibilidad de escalado automático**
- Acceso público: **SI**
- Puerto de la base de datos: **3306**
- Nombre de base de datos inicial: **lp2lab05**
- **Configure las reglas de entrada para que cualquier IP o computador tenga acceso a su base de datos.**
- **IMPORTANTE: El motor de base de datos debe ser configurado para que cualquier computador o IP pueda acceder al mismo. Si esta configuración no es realizada y no es posible conectarse a su motor de base de datos desde la computadora de los jefes de práctica o docente, entonces no se considerará puntaje alguno.**

Una vez creada su base de datos, se le solicita comprobar que funciona correctamente. Se le solicita dejar la base de datos **ACTIVA** en AWS y colocar los datos de conexión en un archivo llamado datosConexion_códigoAlumno.txt (en el archivo deberá colocar el *hostname*, el *username*, el *password* y el nombre de la BD o esquema).

PREGUNTA 2: Ejercicio de paquetes (Sin uso de IDE) (4 puntos)

Como parte de un proceso de reingeniería, se ha establecido modificar la estructura de paquetes inicial de las clases del software que ha desarrollado en el Laboratorio 02.

En la Fig. 01, se muestra a través de un diagrama de clases, la nueva estructura de paquetes.

Se le solicita descargar los archivos fuente que se adjuntan a este enunciado (no aquellos que se encuentran en la Semana 03). Agregue las instrucciones **package** e **import** a todas las clases, así como la estructura de carpetas para que el programa JAVA funcione bajo la nueva estructura de paquetes solicitada. Compruebe que es posible compilar y ejecutar bajo la nueva estructura. Suba un único archivo .zip a PAIDEIA que contenga TODA esta nueva estructura (con todos los paquetes que se visualizan en el diagrama). **Está prohibido utilizar el comodín * para realizar importaciones en las clases. Para compilar sí puede utilizar el *.**

Se descontarán puntos por importaciones innecesarias.

Para un mejor entendimiento, a continuación, se detallan los paquetes y las clases que deberían contener:

- **com.pucp.eventsoft.logistica.model:** Funcion, Local.
- **com.pucp.eventsoft.colaboradores.model:** IConsultable, Artista, OrigenArtista, Agrupacion, Persona.
- **com.pucp.eventsoft.gestion.model:** Evento, Productora, Concierto, ObraTeatral.
- **com.pucp.eventsoft.main:** Principal.

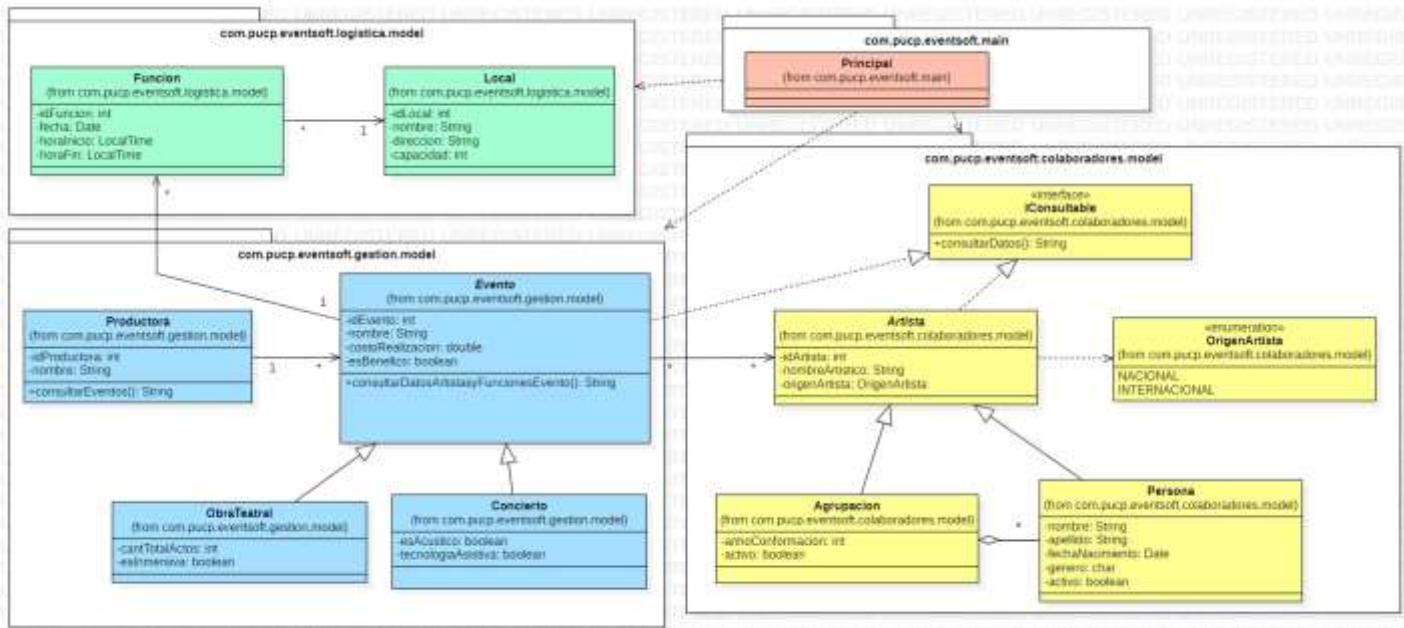


Fig. 01 Diagrama de clases – estructura de paquetes solicitada.

Con las siguientes instrucciones puede corroborar que su programa se ha estructurado y funciona bajo el nuevo esquema:

```
javac com/pucp/eventsoft/logistica/model/*.java com/pucp/eventsoft/colaboradores/model/*.java
com/pucp/eventsoft/gestion/model/*.java com/pucp/eventsoft/main/*.java
```

```
java -cp
com/pucp/eventsoft/logistica/model/*.class;com/pucp/eventsoft/colaboradores/model/*.class;com/p
ucp/eventsoft/gestion/model/*.class; com/pucp/eventsoft/main/Principal
```

PREGUNTA 3: Ejercicio de generación de librerías JAR (Sin uso de IDE) (4 puntos)

Para continuar con este ejercicio,

borre todos los archivos .class generados en el ejercicio anterior en todos los paquetes.

Realice las siguientes instrucciones en estricto orden e indique en un archivo llamado "Pregunta03_codigoAlumno.txt", las líneas de comandos utilizadas por consola para alcanzar cada actividad:

1. Compile únicamente las clases contenidas en **com.pucp.eventsoft.logistica.model** desde el directorio raíz de trabajo.
2. Genere un componente .jar llamado "eventsoftLogisticaModel.jar" que contenga la estructura y clases en Bytecode contenidas en: **com.pucp.eventsoft.logistica.model**.

Una vez creado el componente, borre todos los archivos .java, .class, y estructura de carpetas que corresponden al componente creado "eventsoftLogisticaModel.jar" (**com.pucp.eventsoft.logistica.model**). Prosiga con las siguientes instrucciones:

3. Compile las clases contenidas en **com.pucp.eventsoft.colaboradores.model** desde el directorio raíz de trabajo.
4. Genere un componente .jar llamado "eventsoftColaboradoresModel.jar" que contenga la estructura y clases en Bytecode contenidas en: **com.pucp.eventsoft.colaboradores.model**.

Una vez creado el componente, borre todos los archivos .java, .class, y estructura de carpetas que corresponden al componente creado "eventsoftColaboradoresModel.jar" (**com.pucp.eventsoft.colaboradores.model**). Prosiga con las siguientes instrucciones:

5. Compile las clases contenidas en **com.pucp.eventsoft.gestion.model** desde el directorio raíz de trabajo utilizando los .jar previamente creados: "eventsoftLogisticaModel.jar" y "eventsoftColaboradoresModel.jar".
6. Genere un componente .jar llamado "eventsoftGestionModel.jar" que contenga la estructura y clases en Bytecode contenidas en: **com.pucp.eventsoft.gestion.model**.

Una vez creado el componente, borre todos los archivos .java, .class, y estructura de carpetas que corresponden al componente creado "eventsoftGestionModel.jar" (**com.pucp.eventsoft.gestion.model**). Prosiga con las siguientes instrucciones.

7. Compile las clases contenidas en **com.pucp.eventsoft.main** desde el directorio raíz de trabajo utilizando los .jar previamente creados: "eventsoftLogisticaModel.jar", "eventsoftColaboradoresModel.jar" y "eventsoftGestionModel.jar".
8. Genere un componente **.jar ejecutable** llamado "eventsoftPrincipal.jar" que contenga la estructura y clases en Bytecode contenidas en **com.pucp.eventsoft.main**. Comprueba que funcione. Prosiga con la siguiente instrucción:
9. Además de las instrucciones ejecutadas, incorpore en el archivo "Pregunta03_codigoAlumno.txt" el contenido que debería tener el **MANIFEST** de "eventsoftPrincipal.jar" para que pueda funcionar como componente ejecutable. Suba a PAIDEIA el archivo de texto o documento de texto con todas las instrucciones utilizadas por consola para alcanzar las actividades solicitadas.
- Además, adjunte también los archivos JARs.

PREGUNTA 4: Conexión a Base de Datos (Con uso de IDE) (9 puntos)

Desarrolle una solución de software en lenguaje JAVA utilizando Netbeans que permita **registrar y listar artistas de tipo "Persona"** en y desde una base de datos. Utilice el motor de base de datos que ha creado en la pregunta 1 y utilice el script que se encuentra en PAIDEIA (**ScriptLab05.sql**) que genera las tablas "artista", "agrupacion" y "persona" en la BD. Ud. solo utilizará las tablas "artista" y "persona". **Para el desarrollo de este ejercicio debe utilizar la clase Statement o la clase PreparedStatement.** Cuando programe el listar artistas de tipo persona, debe extraer todos los atributos de los mismos.

En la Fig. 02 se muestran las tablas en la base de datos que se han diseñado para este propósito.

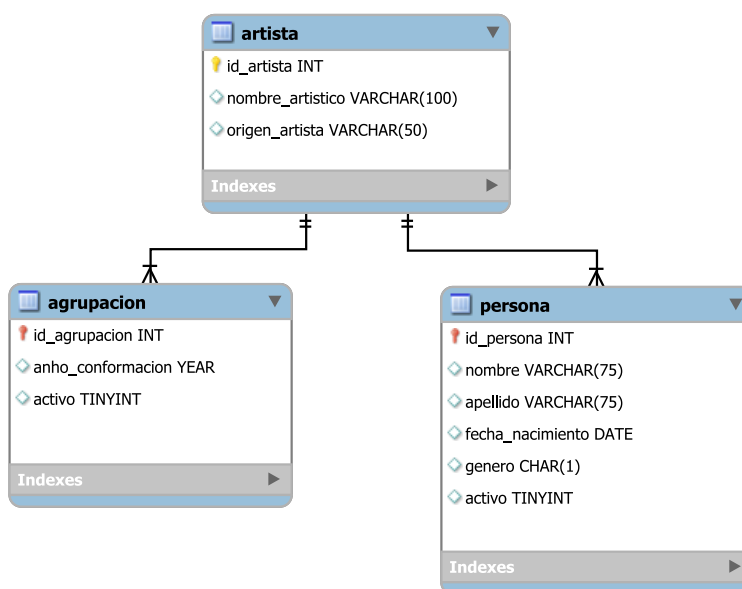


Fig. 02. Diagrama EER de la Base de Datos

Se le solicita crear 3 proyectos **desde cero** y utilizar el componente "eventsoftColaboradores.jar" que ha elaborado en la Pregunta 03:

- **EventSoftDBManager:** Que define la clase con los datos de conexión.
- **EventSoftColaboradoresController:** Que define las clases de acceso a base de datos necesarias para realizar el registro y listado de artistas de tipo persona empleando el patrón DAO.
- **EventSoft:** Que contiene la clase Principal y permitirá mostrar un ejemplo de dos registros y el listado de artistas de tipo persona en y desde la base de datos.



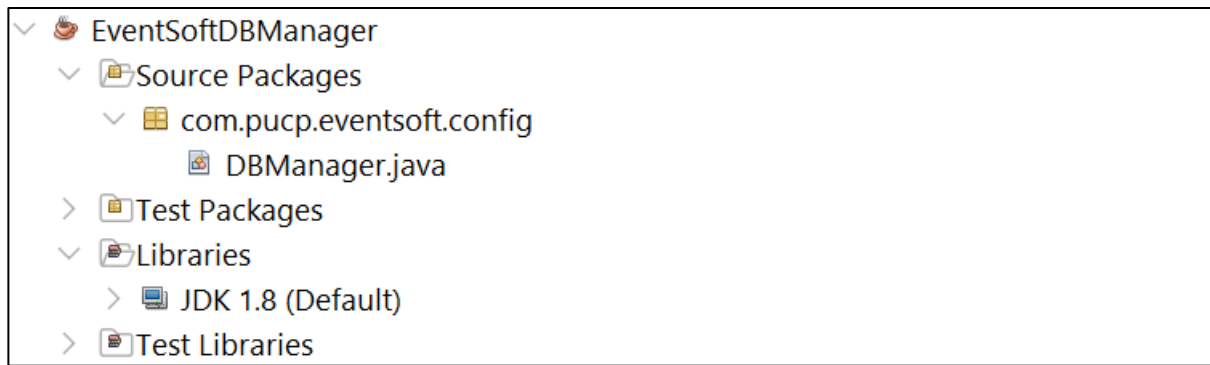


Fig. 03. Estructura de proyectos en JAVA

No es necesario implementar el modificar ni el eliminar artista de tipo persona.
El componente "eventsoftColaboradores.jar" formará parte de los archivos de los proyectos "EventSoftColaboradoresController" y "EventSoft". Asimismo, estará referenciado con ruta relativa.

La clase **Principal** sería la siguiente:

```
package com.pucp.eventsoft.main;
import com.pucp.eventsoft.colaboradores.dao.PersonaDAO;
import com.pucp.eventsoft.colaboradores.model.OriginArtista;
import com.pucp.eventsoft.colaboradores.model.Persona;
import com.pucp.eventsoft.colaboradores.mysql.PersonaMySQL;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
public class Principal {
    public static void main(String[] args) throws Exception{
        SimpleDateFormat sdf = new SimpleDateFormat("dd-MM-yyyy");
        Persona p1 = new Persona("GIANMARCO", OriginArtista.NACIONAL, "JAVIER", "ZIGNAGO",
sdf.parse("17-08-1970"), 'M');
        Persona p2 = new Persona("SHAKIRA", OriginArtista.INTERNACIONAL, "ISABEL", "MEBARAK",
sdf.parse("02-02-1977"), 'F');
        //Se registran los artistas de tipo persona
        PersonaDAO daoPersona = new PersonaMySQL();
        evaluarResultado(daoPersona.insertar(p1), p1.getNombreArtistico());
        evaluarResultado(daoPersona.insertar(p2), p2.getNombreArtistico());
        //Se listan los artistas de tipo persona
        ArrayList<Persona> artistasPersonas = daoPersona.listarTodasPersonas();
        for(Persona p : artistasPersonas){
            p.imprimirDatos();
        }
    }
    public static void evaluarResultado(int resultado, String nombreArtistico){
        if(resultado == 1)
            System.out.println("Se ha registrado con éxito al artista: " + nombreArtistico);
        else
            System.out.println("Ha fallado el registro de: " + nombreArtistico);
    }
}
```

La ejecución del programa genera la siguiente salida:

```
Output - EventSoft (run) x
run:
Se ha registrado con éxito al artista: GIANMARCO
Se ha registrado con éxito al artista: SHAKIRA
1. GIANMARCO - NACIONAL - JAVIER ZIGNAGO - 17-08-1970 - Genero: M
2. SHAKIRA - INTERNACIONAL - ISABEL MEBARAK - 02-02-1977 - Genero: F
BUILD SUCCESSFUL (total time: 5 seconds)
```

Suba en un archivo .zip todos los proyectos desarrollados a PAIDEIA.

Los siguientes aspectos pueden conllevar a un descuento significativo en su propuesta de solución:

- No seguir las instrucciones establecidas en el enunciado.
- Que los programas no compilen.
- Que los programas presenten errores en tiempo de ejecución.
- Que los proyectos tengan referencias innecesarias.
- Que el driver de conexión a base de datos no se haya adjuntado al proyecto.
- Que el driver de conexión a base de datos no se encuentre referenciado con ruta relativa en el proyecto.
- No seguir las prácticas vistas en clase: patrón DAO, desarrollo por componentes en capas, etc.

Profesor del Curso:
Dr. Freddy Paz

20 de abril del 2023