

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 2

9na. 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.
- Se le solicita hacer uso del entorno de desarrollo integrado de Apache Netbeans y JDK 8.
- 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)

PUEDEN 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: (20 puntos)

Newbury Peru S.A.C., una empresa dedicada a la venta de discos de vinilo o LPs, un antiguo formato de almacenamiento y grabación de música en disco físico que apareció en el año de 1955, los cuales eran reproducidos a través de artefactos llamados **tocadiscos** que se conectaban a **amplificadores**, está requiriendo de un producto de software que le permita gestionar los álbumes que salen a la venta actualmente debido al regreso de estos productos. En este caso específico, se ha solicitado programar únicamente el formulario de gestión de álbumes de vinilo. Para este laboratorio se está solicitando programar las funcionalidades de registro y listado o búsqueda de álbumes de vinilo, así como la selección y visualización de los datos de los mismos. Para alcanzar este propósito se ha diseñado la interfaz gráfica que se muestra en la Fig. 01.

Formulario de Gestión de Álbumes de Vinilo

Nuevo Guardar Buscar Cancelar

Gestión de Álbumes de Vinilo

ID del Álbum:

Nombre del Álbum:

Artista:

Año de Lanzamiento:

Género:

Precio:

Cantidad de Canciones:

Presentación del Disco: ☒ Edición Gatefold ☐ Acompañado de CD ☒ Edición de Color

Subir Foto

Descripción del Álbum:

Este disco contiene las canciones vocales de la serie de anime titulada Neon Genesis Evangelion, el cual se desarrolló en 1995 y se le atribuye la gran difusión mundial de la animación japonesa debido a la popularidad de la serie. Es una edición lanzada por el 25 aniversario de la emisión de la serie.

Fig. 01. Interfaz gráfica del formulario a implementar

Asimismo, se ha elaborado el diseño de la base de datos, el cual se presenta en la Figura 02. Se le solicita descargar el archivo "**ScriptSQL-Laboratorio09-INF282-2023-1.sql**" y ejecutar el script en su motor de base de datos MySQL alojado en AWS. El script generará las tablas, sus relaciones y los procedimientos almacenados (*stored procedures*) que deberá utilizar para la programación de las funcionalidades solicitadas.

Es necesario indicar que, para realizar el registro de los datos de un álbum de vinilo, se requiere previamente la implementación de la funcionalidad de listar géneros musicales y listar/buscar/seleccionar artistas.

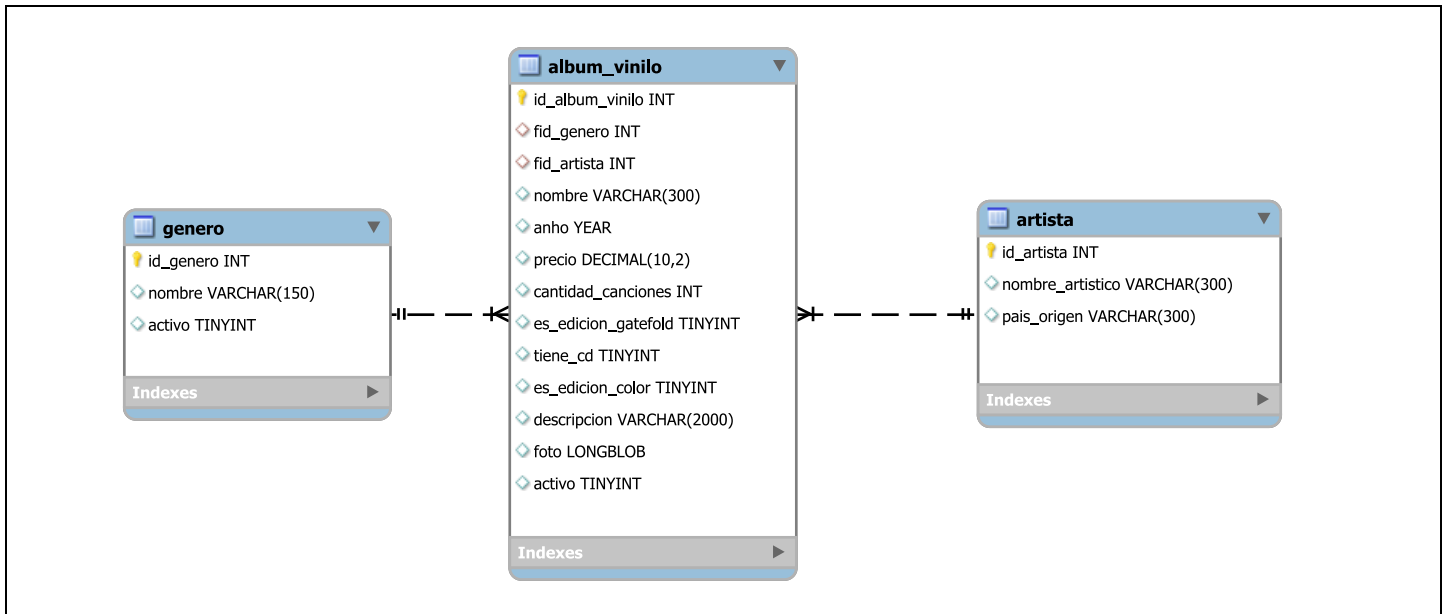


Fig. 02. Diagrama EER de la base de datos

Con respecto a la lógica del negocio, se requiere registrar de cualquier álbum de vinilo los siguientes datos: el nombre, el género musical, el artista al que pertenece el álbum, el año de lanzamiento, el precio con el que se venderá al mercado, la cantidad de canciones que contiene, una descripción, una foto, si es que se trata o no de una edición *gatefold*, si es que se trata o no de una edición a color, y es que viene con versión en CD.

Para la programación de la funcionalidad solicitada debe utilizar el proyecto en JAVA que se proporciona en PAIDEIA. No se está solicitando implementar ni el modificar ni el eliminar álbumes de vinilo. NO son requeridas validaciones por lo que debe asumir que el usuario ingresa correctamente la información al formulario. A continuación, se adjunta un video que muestra la forma en que debería funcionar el formulario: https://youtu.be/Kj5u_dmCtX0

Se le solicita colocar su nombre completo y su código a modo de comentario en las líneas 18-24 de la clase llamada: "frmGestionarAlbumes". No colocar conllevará a una disminución de 2 puntos en la nota final obtenida.

Está prohibido modificar el proyecto "MusicSoftModel" a nivel de programación y también está prohibido modificar la estructura de tablas y los procedimientos almacenados.

Suba su propuesta de solución en un único .zip a PAIDEIA. (Realice un proceso *clean* antes de empaquetar todos los proyectos [solución] en el archivo .zip). El formato de nombre del archivo es: codigoPUCP_Lab09.zip Verifique que haya subido correctamente su archivo a PAIDEIA. Es responsabilidad de cada estudiante asegurarse de haber subido correctamente a PAIDEIA su propuesta de solución a los ejercicios propuestos. No se aceptarán reclamos por equivocaciones respecto a los archivos entregados. Tampoco se aceptarán archivos entregados por otros medios.

Anotaciones de Programación:

Para asignar la foto a la variable de tipo byte[]

```
try{
    albumVinilo.setFoto(Files.readAllBytes(archivoFoto.toPath()));
}catch(Exception ex){
    System.out.println(ex.getMessage());
}
```

Para enviar un arreglo de bytes como parámetro mediante un comando a un procedimiento almacenado

```
cs.setBytes("_foto", albumVinilo.getFoto());
```

En caso se desee leer un LONGBLOB a una variable de tipo byte[]

```
if (!lector.IsDBNull(lector.GetOrdinal("foto"))) local.Foto = (byte[])lector["foto"];
```

Para mostrar una imagen en el JLabel desde una variable de tipo byte[]

```
Image img = new ImageIcon(albumVinilo.getFoto()).getImage();
Image newImg = img.getScaledInstance(pbFoto.getWidth(), pbFoto.getHeight(),
java.awt.Image.SCALE_SMOOTH);
pbFoto.setIcon(new ImageIcon(newImg));
```

Rúbrica de calificación:

- No se considerará puntaje alguno si es que el aspecto a evaluar no funciona a nivel de interfaz gráfica, por lo que se solicita que, a medida que va avanzando el *back-end*, realice la programación del *front-end*. Asimismo, la corrección de los apartados será progresiva. Para obtener el puntaje de un apartado en específico, todos los anteriores aspectos a evaluar deben estar correctamente desarrollados.

A continuación, se citan algunos ejemplos:

- “no se considerará puntaje por la programación de la funcionalidad de registro, si es que para ese registro son necesarias algunas búsquedas y selecciones en el formulario y estas no se encuentran correctamente implementadas.”
 - “no se considerará puntaje por programar la funcionalidad de búsqueda si es que el registro no se encuentra correctamente implementado”.
 - “no se considerará puntaje por la programación de un formulario si es que los métodos de acceso a base de datos que son requeridos para su funcionamiento no se encuentran correctamente implementados”.
- El driver de conexión a base de datos debe estar referenciado con ruta relativa (no absoluta).
 - Se descontarán puntos significativamente si es que no se siguen las indicaciones del enunciado.
 - Se descontarán puntos significativamente si es que existen errores de compilación.
 - Se descontarán puntos por declaración de variables que admiten cualquier tipo de dato como "Object" (JAVA).
 - Se descontarán puntos si el código no está optimizado, por ejemplo, repetición innecesaria de código.
 - Se descontarán puntos si es que no sigue las instrucciones en referencia a los métodos que deberían implementarse en cada clase.
 - Se descontarán puntos por referencias innecesarias entre proyectos.
 - Debe utilizar la estructura vista en clase, proyectos relacionados a: DBManager, a la capa de modelo, a la capa de acceso a base de datos y a la vista.

- A. Desarrollo del proyecto "MusicSoftDBManager" y sus clases asociadas. (1.5 puntos).
- B. Implementación de la funcionalidad que permite listar los géneros musicales (*back-end*): (2.0 puntos).
- C. Implementación de la funcionalidad que permite listar los géneros musicales (*front-end*): (1.0 puntos).
- D. Implementación de la funcionalidad que permite listar/buscar los artistas (*back-end*): (2.5 puntos).
- E. Implementación de la funcionalidad que permite listar/buscar/seleccionar los artistas (*front-end*): (2.5 puntos).
- F. Implementación de la funcionalidad que permite registrar un álbum de vinilo (*back-end*): (3.0 puntos).
- G. Implementación de la funcionalidad que permite registrar un álbum de vinilo (*front-end*): (2.0 puntos).
- H. Implementación de la funcionalidad que permite listar/buscar un álbum de vinilo (*back-end*): (2.5 puntos).
- I. Implementación de la funcionalidad que permite listar/buscar/seleccionar/visualizar los datos un álbum de vinilo (*front-end*): (3.0 puntos).

Profesor del Curso:
Dr. Freddy Paz

25 de mayo del 2023