

CURSO : Lenguaje de Programación II
PROFESOR : XXX
SEMESTRE : 2025 - 3
CICLO : Cuarto
SECCION : T4XX
GRUPO : XX
FECHA : XXXXX

NOTA

ALUMNO (A) :

PRIMERA EVALUACIÓN DE LABORATORIO

Consideraciones generales:

- Se considerará el orden, la claridad de las respuestas y las buenas prácticas.
- Las preguntas deben resolverse de acuerdo con los conceptos discutidos o analizados en clase. Para ello, aplicará su propio criterio para dar una solución a los problemas planteados
- Para el desarrollo de la siguiente aplicación **crea el proyecto MAVEN** con el Group Id **com.examenT1** y Artifact Id **lp2_t1_PrimerApellido_PrimerNombre** y cree la estructura de carpetas necesarias.

LOGRO DE LA EVALUACION:

Al término de la evaluación, el alumno implementa proceso de mantenimiento de una tabla empleando JPA en un proyecto MAVEN, realizando un correcto mapeado y una vista para su funcionalidad.

Consolidado

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	06			
2	14			
Nota Recalificada				

A usted como desarrollador se le ha asignado la funcionalidad de **gestión de solicitudes** del **Instituto de Arte y Tecnologías de Lima**, para ello:

Pregunta 01

- Mediante el control de versiones con Git, **crea por lo menos 2 commit con mensajes**, al iniciar **“Proyecto iniciado PrimerApellido, Primer Nombre”** y al finalizar **“Proyecto finalizado PrimerApellido, Primer Nombre”**, agregando la información necesaria.
- Crea el proyecto Maven, agrega el archivo de **persistencia** con la configuración correspondiente y agrega las dependencias necesarias para el desarrollo en el archivo **pom.xml**.
- Cargue el archivo **databaseT1.sql** a **MySQLWorkbench** y liste las solicitudes de manera descendente.
- Crear, una por cada tabla, la entidad JPA adecuada usando anotaciones (@Entity, @Table, @Id/@GeneratedValue, @Column, etc.), marcando como obligatorios los campos NOT NULL, reflejando en la declaración de @Column cualquier cláusula DEFAULT definida en la base y mapeando las relaciones entre tablas mediante las anotaciones de asociación y @JoinColumn correspondientes.

Pregunta 02

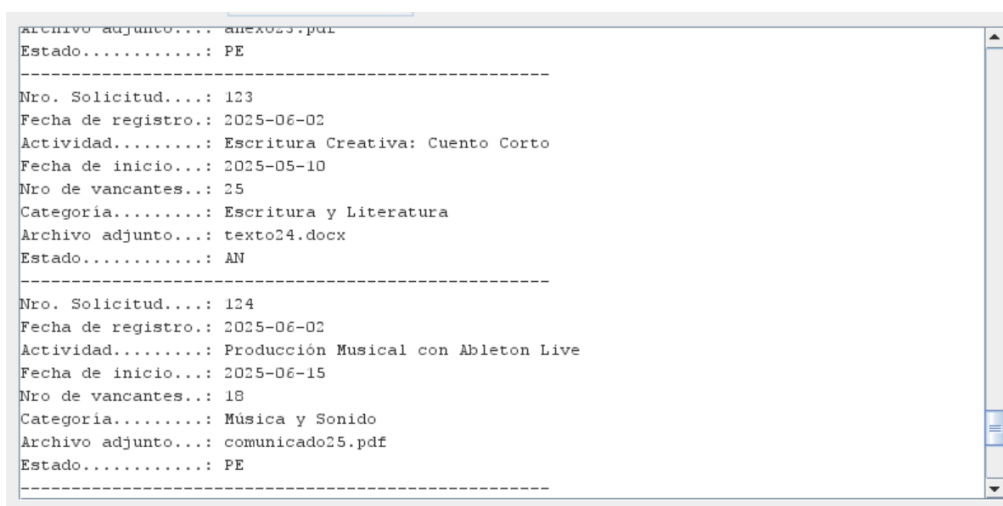
Utilice la GUI compartida **DlgSolicitud.java** que permita **dar mantenimiento** a los datos de las solicitudes implementado los siguientes métodos.

➤ **CARGAR ACTIVIDADES:**

Obtiene las actividades y carga la data en el comboBox **cboActividades**.

➤ **LISTAR:**

Recupere todas las solicitudes y cargue su información en el **txtSalida** con el formato mostrado en la imagen (utilice el método **imprimir**).



```
Archivo adjunto...: anexo23.pdf
Estado.....: PE
-----
Nro. Solicitud...: 123
Fecha de registro.: 2025-06-02
Actividad.....: Escritura Creativa: Cuento Corto
Fecha de inicio...: 2025-05-10
Nro de vacantes..: 25
Categoría.....: Escritura y Literatura
Archivo adjunto...: texto24.docx
Estado.....: AN
-----
Nro. Solicitud...: 124
Fecha de registro.: 2025-06-02
Actividad.....: Producción Musical con Ableton Live
Fecha de inicio...: 2025-06-15
Nro de vacantes..: 18
Categoría.....: Música y Sonido
Archivo adjunto...: comunicado25.pdf
Estado.....: PE
-----
```

➤ **ADICIONAR:**

Cuando el usuario pulse el botón **“Adicionar”**, ingrese los datos en el formulario y, al hacer clic en el botón **OK**, se debe de guarda la solicitud en la base de datos; también muestre el mensaje de éxito **“Solicitud registrada”** (usando el método **mensajeInfo**) y, finalmente, borre todos los campos del formulario (usando el método **limpiar**).

➤ **BUSCAR:**

Al ingresar el número de solicitud y presionar “**Buscar**”, recupere sus datos y pueble el formulario. Si no existe, muestre el mensaje “Solicitud no encontrada” (usando el método **mensajeError**); en caso contrario, habilite el formulario (usando el método **habilitarOk**).

➤ **MODIFICAR:**

Cuando el usuario pulse el botón “**Modificar**” y luego de haber obtenido los datos de la solicitud con el botón “**Buscar**”, al hacer clic en el botón **OK**, se debe modificar los datos en la base de datos; también muestre el mensaje de éxito “Solicitud actualizada” (usando **mensajeInfo**) y borre todos los campos del formulario (usando el método **limpiar**).

➤ **ELIMINAR:**

Cuando el usuario pulse el botón “**Eliminar**” y luego de haber obtenido los datos de la solicitud con el botón “**Buscar**”, al hacer clic en el botón **OK**, se debe eliminar la solicitud en la base de datos; también muestre el mensaje de éxito “Solicitud eliminada” (usando el método **mensajeInfo**) y borre todos los campos del formulario (usando el método **limpiar**).

RÚBRICA DE LA PRIMERA EVALUACIÓN

Rúbrica					
Pregunta	puntaje	6 puntos	4 puntos	3 puntos	2 puntos
1	6 puntos	Realiza el control de versiones al inicio y final, la estructura del proyecto y mapea correctamente las clases.	Realiza el control de versiones solo al inicio, la estructura del proyecto y mapea correctamente al menos dos clases.	Realiza el control de versiones solo al inicio, la estructura del proyecto y mapea correctamente al menos una clase.	No <u>realiza</u> el control de versiones, pero sí la estructura del proyecto y mapea correctamente al menos 1 clase.
		14 puntos	11 puntos	8 puntos	5 puntos
2	14 puntos	Carga las actividades, logra las 4 acciones: ADICIONAR, MODIFICAR, ELIMINAR Y LISTAR.	Carga las actividades, logra al menos 3 de las acciones: ADICIONAR, MODIFICAR, ELIMINAR y/o LISTAR.	Carga las actividades, logra al menos 2 de las acciones: ADICIONAR, MODIFICAR, ELIMINAR y/o LISTAR.	Solo Carga las actividades y logra 1 acción: ADICIONAR, MODIFICAR, ELIMINAR y/o LISTAR.