

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJE DE PROGRAMACIÓN 2

Examen 2
(Segundo semestre 2022)

Indicaciones generales:

- Duración: 3 horas.
 - Materiales o equipos a utilizar:
ESTÁ PERMITIDO USAR APUNTES Y DIAPOSITIVAS DE CLASE.
El Internet será utilizado solo para consultar documentación de programación y PAIDEIA.
Está prohibido cualquier forma de conversación con otras personas.
 - El uso apropiado de buenas prácticas de programación (sobre todo aquellas vistas en clase) será tomado en cuenta para la calificación.
- Puntaje total: 20 puntos
-

Parte Teórica (1 punto)

Observe el siguiente código fuente e indique cuáles son las posibles salidas que puede mostrar el programa. Para cada salida del programa, indique el orden de las sentencias (apóyese del número de línea).

1	package pe.edu.pucp.lp2soft.ejemplo;
2	public class Ejemplo implements Runnable{
3	public static void main(String[] args) throws Exception{
4	Runnable e = new Ejemplo();
5	Thread t = new Thread(e,"Hilo 1");
6	t.start();
7	System.out.print(t.getName());
8	System.out.println(", estado = " + t.isAlive());
9	}
10	@Override
11	public void run(){
12	Thread t = Thread.currentThread();
13	System.out.print(t.getName());
14	System.out.println(", estado = " + t.isAlive());
15	}
16	}

Parte Práctica: (19 puntos)

El departamento de Ingeniería de la universidad requiere de un producto de software que le permita gestionar los múltiples eventos que organizan las distintas especialidades. Para resumir la lógica de negocio, un evento es organizado por una especialidad y tiene como responsables a uno o varios docentes de esa especialidad. Asimismo, en estos eventos se realizan un conjunto de actividades.

En PAIDEIA se encuentra el Script SQL que deberá utilizar para la programación de lo solicitado. Ejecute el script en su motor de base de datos de **AWS Academy**, el cual generará las tablas, relaciones y procedimientos almacenados que se visualizan en la Figura 01.

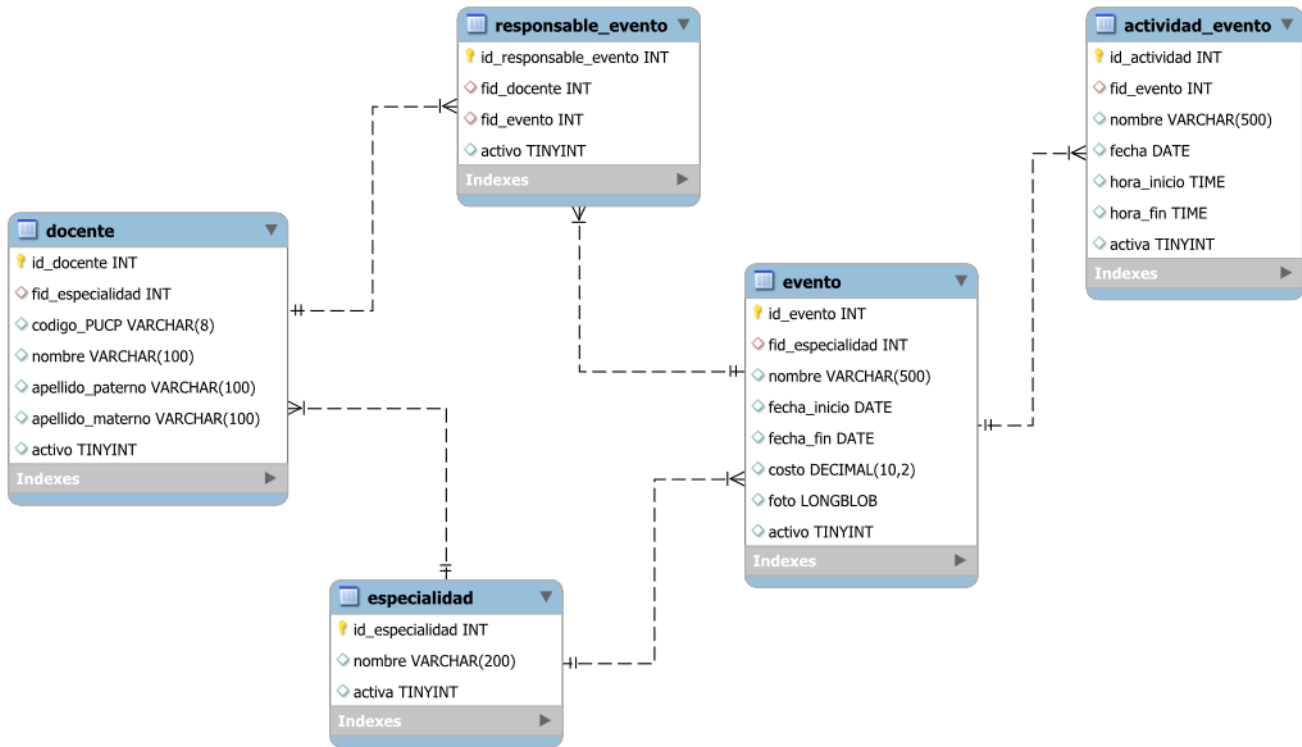


Fig. 01. Diagrama EER de la Base de datos

Se solicita también descargar los proyectos base (**JAVA para el back-end y C# para el front-end**) con los cuales trabajará la propuesta de solución. Se solicita implementar las funcionalidades de registro, búsqueda, selección y visualización de eventos. No obstante, debe tener en cuenta que, para programar las funcionalidades anteriormente mencionadas, se requiere además implementar el listado de especialidades y la búsqueda y selección de docentes (responsables del evento), para poder registrar correctamente la información del evento.

Para la programación de este formulario, se le solicita utilizar una arquitectura en participar. Se le solicita programar (2 servidores en JAVA) y (1 cliente en C#). Los servidores se implementarán en JAVA, estos serán: un servidor RMI y un servidor Web desplegado en Glassfish. El servidor RMI será el encargado de conectarse a la base de datos y de realizar las consultas SQL. El servidor Web consumirá los métodos de los objetos daos que el servidor RMI coloque a disposición (es decir, trabajará como un cliente del servidor RMI) y pondrá a disposición de otro cliente (la interfaz gráfica en C#), vía servicios web, los métodos de acceso a la base de datos.

Utilice una sola clase llamada "**ServicioWS**" para definir todos los servicios en el programa web services.

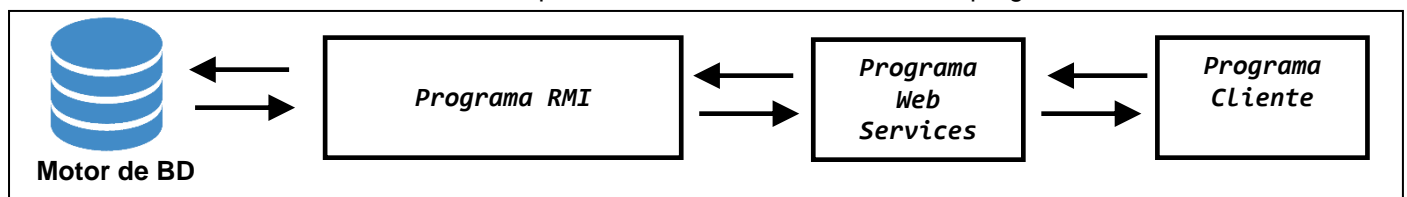


Fig. 02. Arquitectura a utilizar

En la Figura 03 y 04, se muestra el formulario solicitado.

Formulario de Gestión de Eventos

Nuevo Guardar Buscar Cancelar

Gestión de Eventos PUCP

Datos Generales del Evento **Actividades del Evento**

Reconocimiento a la Investigación

ID del Evento:

Nombre del Evento:

Fecha de Inicio:

Fecha Fin:

Especialidad que organiza el evento:

Costo del Evento:

Profesores Responsables

Código PUCP:

Nombre Completo:

	Código PUCP	Nombre Completo	Especialidad
▶	19931850	MANUEL TUPIA ANTICONA	INGENIERIA INFORMATICA
	19960275	LUIS FLORES GARCIA	INGENIERIA INFORMATICA

Fig. 03. Formulario de Gestión de Eventos

Formulario de Gestión de Eventos

Nuevo Guardar Buscar Cancelar

Gestión de Eventos PUCP

Datos Generales del Evento **Actividades del Evento**

Nombre de la Actividad:

Fecha:

Hora Inicio: Hora Fin:

	Actividad	Fecha	Hora Inicio	Hora Fin
▶	Inauguración del evento	06-12-2022	10:00	11:00
	Premiación de docentes	06-12-2022	11:00	13:00
	Clausura del evento	06-12-2022	13:00	13:30

Fig. 04. Formulario de Gestión de Eventos

Finalmente, cuando registre un evento, utilizará una única función en donde se registre tanto el evento como los responsables y actividades asociadas al mismo. No utilizará tres funciones separadas para este propósito. Cuando se muestre la información de un determinado evento, una vez que el usuario ha seleccionado el evento del cual se desea visualizar la información, se llamará a un método de la clase DocenteMySQL "listarPorIdEvento" y se traerá la lista de docentes (responsables) asociados a ese evento seleccionado. Lo mismo ocurrirá con las actividades, se llamará a un método de la clase ActividadMySQL "listarPorIdEvento" y se traerá la lista de actividades asociadas a ese evento. En otras palabras, cuando registre utilizará una única función "insertar(Evento evento)" en EventoMySQL, mientras que para el listado utilizará tres funciones separadas: "listarPorNombre(String nombre)" en EventoMySQL, "listarPorIdEvento(int idEvento)" en DocenteMySQL y "listarPorIdEvento(int idEvento)" en ActividadMySQL.

No está permitido agregar, eliminar o modificar atributos a las clases que corresponden a la capa del modelo.

Aquí se adjunta un video de lo solicitado: <https://youtu.be/lsyH50iDaMA>

No es necesario considerar validaciones. Asuma que el usuario no comete errores al momento de utilizar el software. Antes de comenzar a trabajar, configure el NETBEANS para que sea posible trabajar con el Glassfish Server.

Una vez que ha culminado el ejercicio, suba a PAIDEIA su propuesta de solución.

Rúbrica de Calificación:

- 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 existen errores de compilación.
- Se descontarán puntos si es que modifica el proyecto del modelo a excepción de la serialización.
- Se descontarán puntos por declaración de variables que admiten cualquier tipo de dato como "var" (C#).
- 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, por ejemplo: DBManager, al modelo, a las interfaces, al acceso a base de datos, al RMI, al web services, y a la vista.

ANOTACIONES DE PROGRAMACIÓN

La lista de métodos a implementar es:

- **ActividadMySQL:**
 - `ArrayList<Actividad> listarPorIdEvento(int idEvento)`
- **DocenteMySQL:**
 - `ArrayList<Docente> listarPorCodigoPUCPNombreIdEspecialidad(String codigoPUCPNombre, int idEspecialidad)`
 - `ArrayList<Docente> listarPorIdEvento(int idEvento)`
- **EspecialidadMySQL:**
 - `ArrayList<Especialidad> listarTodas()`
- **EventoMySQL:**
 - `int insertar(Evento evento)`
 - `ArrayList<Evento> listarPorNombre(String nombre)`

(En el caso de C#) Para obtener la hora de un datetimepicker utilizar:

```
actividad.horaInicio = dtpHoraInicio.Value.ToString("HH:mm");
```

(En el caso de JAVA) Para enviar un arreglo de bytes a un CallableStatement puede utilizar:

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

En caso desee recibir bytes:

```
evento.setFoto(rs.getBytes("foto"));
```

(En el caso de C#) Para asignar la foto a la variable de tipo byte[]:

```
FileStream fs = new FileStream(_rutaFoto, FileMode.Open, FileAccess.Read);  
BinaryReader br = new BinaryReader(fs);  
_evento.foto = br.ReadBytes((int)fs.Length);
```

(En el caso de C#) Para mostrar la imagen desde el arreglo:

```
MemoryStream ms = new MemoryStream(_evento.foto);  
pbFoto.Image = new Bitmap(ms);
```

Para aumentar el tamaño de la cuota de transferencia en C#, se debe modificar el archivo "app.config":

```
<basicHttpBinding>  
    <binding name="ServicioWSPortBinding" receiveTimeout="00:1:00"  
    sendTimeout="00:1:00"  
    maxBufferSize="2147483647" maxReceivedMessageSize="2147483647"/>  
</basicHttpBinding>
```

RÚBRICA DE CALIFICACIÓN

- A. Proyecto DBManager. (1 punto)
- B. Modificación del proyecto Modelo para dar soporte al RMI. (0.5 punto)
- C. Proyecto Interfaces. (1 punto)
- D. Proyecto Controller. (6 puntos)
- E. Proyecto RMI. (1 punto)
- F. Proyecto Web Services. (1 punto)
- G. Interfaz gráfica – Funcionamiento del listado de especialidades. (1 punto)
- H. Interfaz gráfica – Funcionamiento de la búsqueda, selección y asignación de docentes responsables a un evento. (2 puntos)
- I. Interfaz gráfica – Funcionamiento de asignación de actividades. (2 puntos)
- J. Interfaz gráfica – Registro del evento. (1.5 puntos)
- K. Interfaz gráfica – Búsqueda de eventos. (1 punto)
- L. Interfaz gráfica – Visualización de todos los datos del evento. (1 punto)

Profesor del curso: **Dr. Freddy Paz**

San Miguel, 05 de diciembre del 2022