

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**  
**FACULTAD DE CIENCIAS E INGENIERÍA**

**LENGUAJE DE PROGRAMACIÓN 2**

**Examen 2**  
**(Primer semestre 2023)**

**Indicaciones generales:**

- **Duración:** 3 horas.
- **Materiales o equipos a utilizar:**  
 La sección teórica se realizará sin material de apoyo.  
 En la sección práctica podrá utilizar material de apoyo: diapositivas, ejemplos de clase, grabaciones de clase, código fuente y documentación oficial de Microsoft u Oracle.
- No está permitido el uso de ningún material o equipo electrónico adicional al indicado.
- **La presentación, la ortografía y la gramática influirán en la calificación.**

Puntaje total: 20 puntos

**PARTE TEÓRICA:** (Resolver en el cuadernillo)

**Pregunta 1** (1 punto)

- ¿Cuál es el resultado de compilar y la salida de ejecutar el siguiente código en JAVA? Explique.  
 No se considerará puntaje alguno por este criterio si la respuesta no es correcta en su totalidad.

```

1 class Base{
2     public void imprimir(){
3         System.out.println("Impresion por defecto");
4     }
5 }
6 class DerivadaA extends Base{
7     @Override
8     public void imprimir(){
9         System.out.println("Impresion de DerivadaA");
10    }
11 }
12
13 class Ejemplo extends Base{
14     public Ejemplo(){System.out.println("Objeto ejemplo");}
15     public static void main(String[] args){
16         Base objBase = new DerivadaA();
17         DerivadaA objDerivadaA = new DerivadaA();
18         Ejemplo objEjemplo = new Ejemplo();
19         objBase.imprimir();
20         objDerivadaA.imprimir();
21         objEjemplo.imprimir();
22     }
23 }
24 
```

**PARTE PRÁCTICA:**

**Pregunta 3** (19 puntos)

(Resolver empleando VISUAL STUDIO y .NET Framework 4.8)

Una nueva plataforma de *streaming* desea llevar la gestión de las series de televisión que se producen y emiten de las diferentes cadenas para proporcionar posteriormente información adecuada a sus consumidores.

Para resumir la lógica de negocio, una serie de televisión tiene una productora como HBO, Warner Bros, Fox, etc. Asimismo, la serie de televisión tiene varios capítulos que están a cargo de un director.

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.

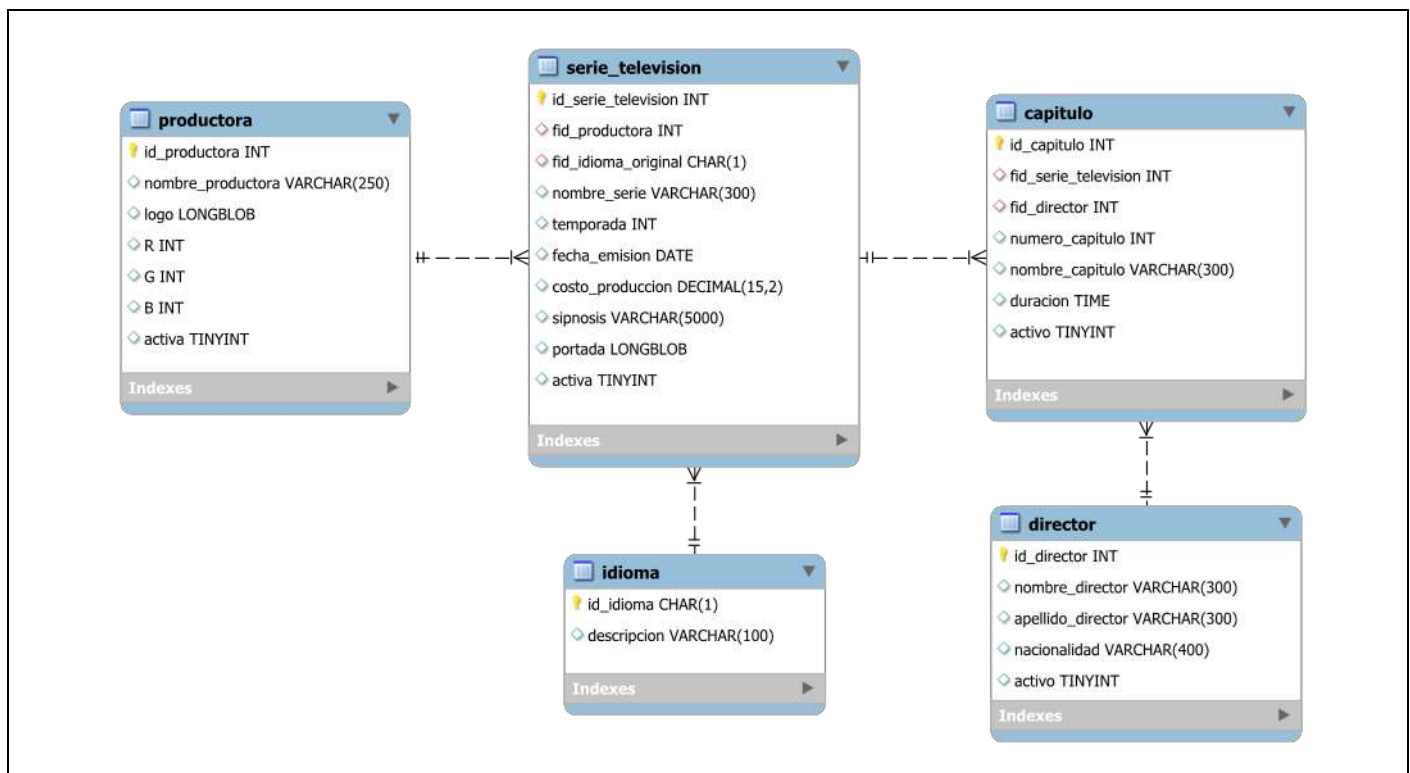


Fig. 01. Diagrama EER de la Base de datos

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 series de televisión. No obstante, debe tener en cuenta que, para programar las funcionalidades anteriormente mencionadas, se requiere además implementar el listado de productoras y la búsqueda y selección de directores (responsables de la dirección de cada capítulo de la serie), para poder registrar correctamente toda la información.

Para la programación de este formulario, se le solicita utilizar una arquitectura en particular. 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. Además, todos los proyectos debe trabajarlos en localhost. Asimismo, debe utilizar la versión 5.0 del Glassfish, JDK 8 y la versión del Framework 4.8.

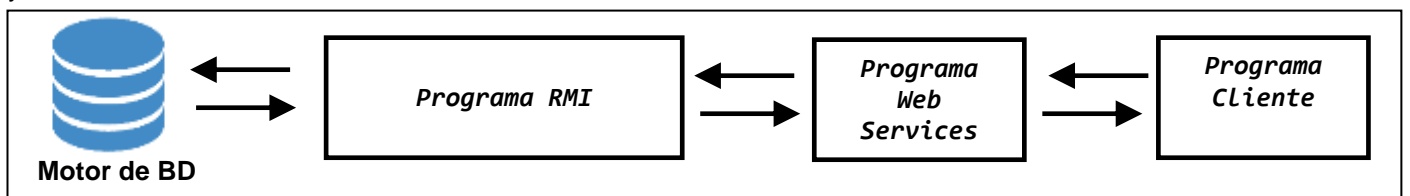


Fig. 02. Arquitectura a utilizar

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

Formulario de Gestión de Series de Televisión

Nuevo Guardar Buscar Cancelar

**Datos Generales** Capítulos de la Serie

**GAME OF THRONES**

ID de la Serie:

Nombre de la Serie:

Productora:

Temporada:

Fecha de Emisión:

Costo de Producción: \$

Idioma Original: ☒ Inglés ☐ Español

**HBO**

**Sipnosis:**

Es una serie que relata las vivencias de un grupo de personajes de distintas casas nobiliarias en el continente ficticio de Poniente para tener el control del Trono de Hierro y gobernar los siete reinos que conforman el territorio.

Fig. 03. Formulario de Gestión de Series de Televisión.

Formulario de Gestión de Series de Televisión

Nuevo Guardar Buscar Cancelar

**Datos Generales** **Capítulos de la Serie**

N° del Capítulo:

Nombre del Capítulo:

Director del Capítulo:

Duración del Capítulo:

N° del Capítulo	Nombre del Capítulo	Director	Duración
1	WINTER IS COMING	T. VAN PATTEN	55:08
2	THE KINGSROAD	T. VAN PATTEN	59:33
3	LORD SNOW	B. KIRK	43:59
4	CRIPPLES, BASTARDS AND BROKEN THINGS	B. KIRK	56:50

Fig. 04. Formulario de Gestión de Series de Televisión – Parte 02.

Cuando el usuario seleccione una productora o cuando se seleccione una productora, el sistema deberá colorear los paneles superior e inferior con el color representativo de la productora, el cual se encuentra almacenado en la base de datos con los valores RGB. Asimismo, cuando seleccione la productora, deberá aparecer el logo de la productora en el componente **pictureBox** correspondiente al logo de la productora.

Para lograr esta funcionalidad puede utilizar:

**Color color = Color.FromArgb(prod.r, prod.g, prod.b);**

Con respecto al nombre del director, en el formulario de búsqueda de directores, debe aparecer su nombre completo. Sin embargo, en el formulario de gestión de series de televisión, aparecerá la primera letra de su nombre seguido de un punto y el apellido como se muestra en la Fig. 04.

Finalmente, cuando registre una serie de televisión, utilizará una única función en donde se registre tanto la serie como los capítulos de la misma. No utilizará dos funciones o métodos separados para este propósito. Cuando se muestre la información de una determinada serie de televisión, una vez que el usuario ha seleccionado la serie de la cual se desea visualizar la información, se llamará a un método de la clase **CapituloMySQL** "listarPorIdSerieTelevision" y se traerá la lista de capítulos asociados a esa serie seleccionada. En otras palabras, cuando registre utilizará una única función "insertar(SerieTelevision

serieTelevision)" en SerieTelevisionMySQL, mientras que para el listado utilizará dos funciones o métodos separados:

" listarPorNombre(String nombre)" en SerieTelevisionMySQL,

" listarPorIdSerieTelevision(int idSerieTelevision)" en CapituloMySQL

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

A continuación se muestra un video que muestra el funcionamiento de lo solicitado: [https://youtu.be/YpxoU\\_DWOvw](https://youtu.be/YpxoU_DWOvw)

No es necesario considerar validaciones. Asuma que el usuario no comete errores al momento de utilizar el software. Igualmente, asuma que todos los capítulos duran menos de 60 min y que el formato ingresado siempre es "mm:ss". Antes de comenzar a trabajar, configure el NETBEANS para que sea posible trabajar con el Glassfish Server. Asimismo, agregue la programación necesaria para que la capa modelo pueda ser utilizada con RMI.

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

### Anotaciones de programación:

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

```
cs.setBytes("_portada", serieTelevision.getPortada());
```

En caso desee recibir bytes:

```
serieTelevision.setPortada(rs.getBytes("portada"));
```

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

```
FileStream fs = new FileStream(_rutaFotoLocal, FileMode.Open, FileAccess.Read);  
BinaryReader br = new BinaryReader(fs);  
_serieTelevision.portada = br.ReadBytes((int)fs.Length);  
fs.Close();
```

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

```
MemoryStream ms = new MemoryStream(_serieTelevision.portada);  
pbPortada.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:

- 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 existen errores de compilación.
- Se descontarán puntos por declaración de variables que admiten cualquier tipo de dato como "var" (C#) y "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 del enunciado y 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, las interfaces y a la vista.
  - La implementación correcta del RMI es obligatoria para la obtención de puntaje asociado al servicio web.
- A. Implementación de los métodos que permiten listar productoras (en RMI y WebServices): 2 puntos.
- B. Implementación de la visualización/selección de productoras con logo y colores (en Front): 2 puntos.
- C. Implementación de los métodos que permiten listar directores (en RMI y WebServices): 2 puntos.
- D. Implementación de la búsqueda, listado y selección del director de un capítulo (en Front): 2 puntos.
- E. Implementación de los botones agregar/eliminar capítulo (en Front): 1 punto.
- F. Implementación de los métodos que permiten registrar una serie de televisión (en RMI y WS): 3 puntos.
- G. Implementación del registro de serie de televisión (en Front): 2 puntos.
- H. Implementación de los métodos que permiten listar series de televisión (en RMI y WS) – se incluye listarSeriesTelevisionPorNombre y listarCapitulosPorIdSerieTelevision: 3 puntos.
- I. Implementación de la búsqueda, listado y selección de series de televisión (en Front): 2 puntos.

Profesor del curso: Prof. Freddy Paz

San Miguel, 03 de julio de 2023