

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

LENGUAJE DE PROGRAMACIÓN 2

Examen 2

(Segundo 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 (1.0 punto)

Explique para qué se utiliza la palabra reservada “**lock**” en C# y escriba un pequeño ejemplo de uso en código fuente en su cuadernillo. Además, explique brevemente el ejemplo.

Pregunta 2 (1.0 punto)

Explique el funcionamiento de los semáforos en JAVA y escriba un pequeño ejemplo de uso en código fuente en su cuadernillo. Además, explique brevemente el ejemplo.

PARTE PRÁCTICA:

Pregunta 3 (18 puntos)

(Resolver empleando VISUAL STUDIO y .NET Framework 4.8)

Debido a su amplia experiencia en programación, usted ha sido contratado(a) por una empresa llamada *GameFreak* dedicada al desarrollo de videojuegos para su desempeño como programador. La temática del videojuego es aquella de un mundo ficticio habitado por criaturas misteriosas que conviven en armonía con las personas, conocidas como *Pokémon*. Estas misteriosas criaturas poseen ciertas características que las diferencia unas de otras: un identificador único de tres cifras (valor o número en la pokedex nacional), son de un tipo específico (o dos) (si solo es de un tipo, el segundo debe guardarse como tipo ninguno), tienen un nombre, un peso, una altura, una descripción, una foto y pertenece a una determinada generación. Asimismo, un *Pokémon* puede aprender movimientos a medida que sube de nivel en el videojuego. Cada movimiento tiene ciertas características que son necesarias registrar en el sistema como: el nivel en el que ese movimiento es aprendido por el *Pokémon*, el nombre del movimiento, el tipo al cual pertenece ese movimiento (por ejemplo, un movimiento puede ser de tipo Normal, Fuego, Planta, etc.), la clase de movimiento (por ejemplo, un movimiento puede ser Físico, Especial, etc.), el blanco o espectro a quien puede dirigir el movimiento (por ejemplo, un movimiento lo puede dirigir a un *Pokémon* aliado o a un Oponente Elegido o a Todos los Oponentes, etc.), el poder de ese movimiento (valor de 0 a 200), la exactitud de ese movimiento (valor de 0 a 100) y los puntos de poder de ese movimiento (valor de 0 a 100). Bajo este escenario, se le solicita realizar la programación del registro, búsqueda/listado y selección de los datos de estas criaturas misteriosas. Tener en cuenta de que para que esto sea posible, es necesario implementar previamente el listado de tipos elementales, el listado de clases de movimiento y el listado/búsqueda y selección de blancos o espectros a quienes dirigen los movimientos.

Para la programación del formulario, se le solicita trabajar con los proyectos iniciales que se encuentran en PAIDEIA tanto en Java como C#. Se le solicita programar (2 servidores en JAVA) y (1 cliente en C#). Los dos servidores en JAVA, 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, vía servicios web, los métodos de acceso a la base de datos.

Utilice una sola clase llamada "**ServicioWeb**" para definir todos los métodos del servicio en el programa web services.

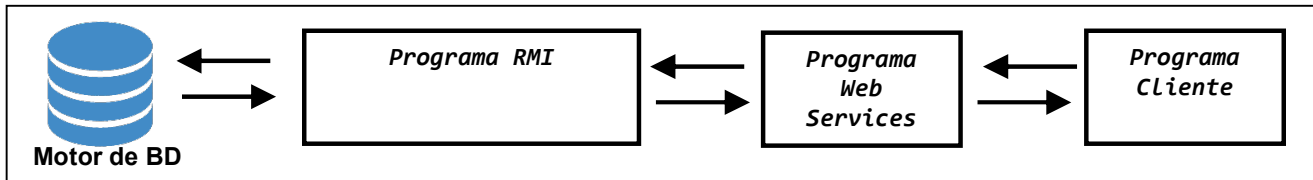


Fig. 01. Arquitectura a utilizar

Se le solicita descargar el script SQL que se encuentra en PAIDEIA y ejecutarlo en su motor de base de datos de AWS Academy. El script generará las siguientes tablas y sus relaciones:

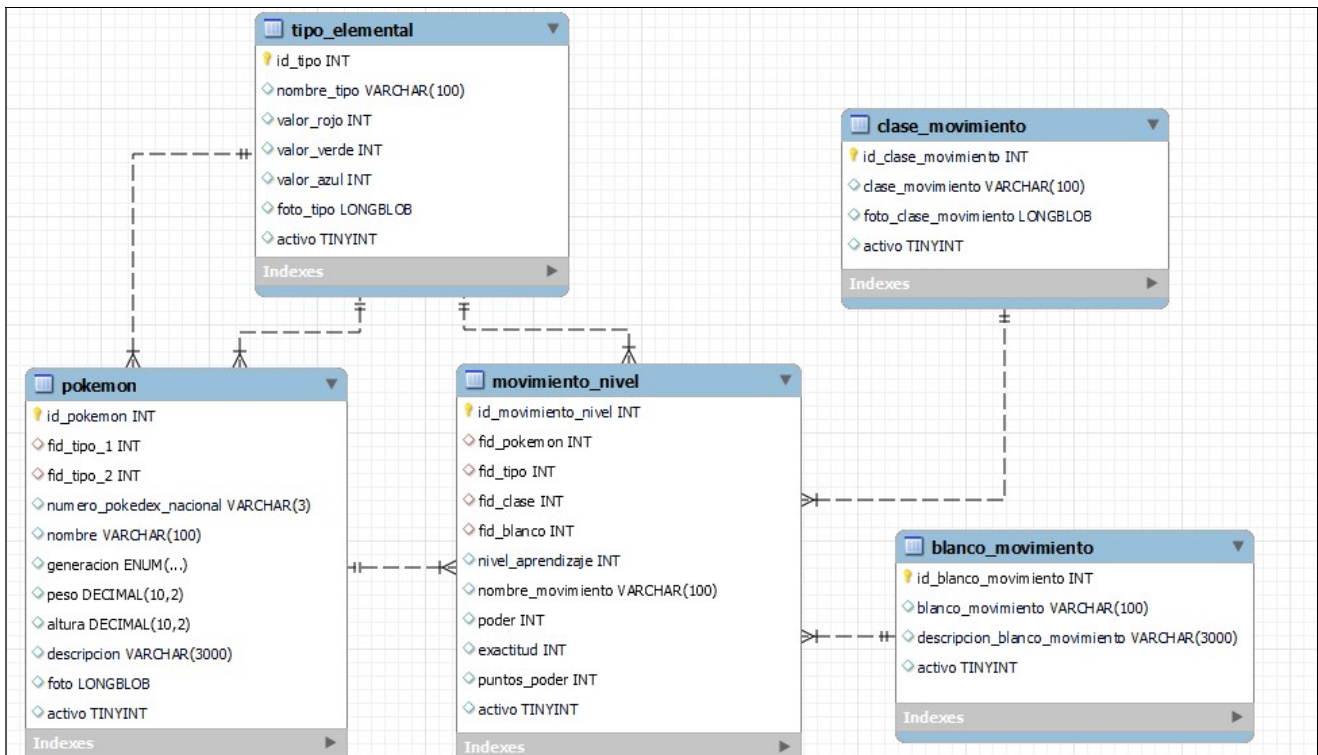


Fig. 02. Diagrama EER de la Base de datos

No es posible modificar los atributos de las clases relacionadas a la capa del modelo ni los procedimientos almacenados. Su modificación conllevará a la disminución significativa de puntaje.

Utilice todos los procedimientos almacenados provistos en el Script SQL.

En las Fig. 03 y Fig. 04 se muestran las interfaces gráficas que debe programar.

Coloque su nombre y código PUCP en las clases: frmGestionPokemons (C#) y ServicioWeb (JAVA – proyecto Web Service). No colocar su nombre conlleva a la disminución de 2 puntos en el total.

A continuación, se coloca un video sobre el funcionamiento del formulario: <https://youtu.be/0Xc4Ynj2abY>

Quando registre un Pokémon, utilizará una única función en donde se registre tanto el pokémon como sus movimientos. No utilizará dos funciones o métodos separados para este propósito.

Quando se muestre la información de un determinado Pokémon, una vez que el usuario ha seleccionado el pokémon del cual desea visualizar la información, se llamará a un método de la clase MovimientoNivelMySQL "listarPorIdPokemon" y se traerá la lista de movimientos asociados a este pokémon seleccionado. En otras palabras, cuando registre utilizará una única función "insertar(Pokemon pokemon)" en PokemonMySQL, mientras que para el listado utilizará dos funciones o métodos separados:

"listarPorNombre(String nombre)" en PokemonMySQL,

"listarPorIdPokemon (int idPokemon)" en MovimientoNivelMySQL

Cuando el usuario seleccione el primer tipo elemental del Pokémon a registrar, el BackColor tanto del panel "panelNroNombre" como del panel "panelTipo1" deben pintarse del color asociado al primer tipo elemental seleccionado.

Para lograr esta funcionalidad puede utilizar el evento "SelectedIndexChanged" verificando que no se den errores por null y utilizando la siguiente sentencia:

Color color = Color.FromArgb(tipo.valorRojo,tipo.valorVerde,tipo.valorAzul);

Asimismo, se debe visualizar la foto asociada al primer tipo elemental seleccionado en "pbTipo1".

Cuando el usuario seleccione el segundo tipo elemental del Pokémon a registrar, el BackColor del panel "panelTipo2" debe pintarse del color asociado al segundo tipo elemental seleccionado. Asimismo, se debe visualizar la foto asociada al segundo tipo elemental seleccionado en "pbTipo2".

Finalmente, cuando se seleccione el tipo y la clase del movimiento a agregar, se debe visualizar las fotos asociadas al tipo y la clase seleccionada en "pbTipoMovimiento" y "pbClaseMovimiento" respectivamente.

Suba a PAIDEIA su propuesta de solución.

No es necesario realizar validaciones. Asuma que el usuario ingresa correctamente la información.

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. Asimismo, debe desarrollar el *back-end*, antes de desarrollar el *front-end*. No se considerará puntaje alguno por implementación del *front-end* si es que el *back-end* presenta errores.

Recuerde que para la generación a la cual pertenece el Pokémon (enumerado) debe colocar el atributo specified en true, además de asignar el valor.

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 y fuera de los plazos establecidos para la evaluación.

Pokémon y GameFreak son marcas registradas y solo se están utilizando con fines académicos.

Rúbrica de calificación:

- 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 del listado de tipos elementales a nivel de RMI, servicio web e interfaz gráfica: 3 puntos.

B. Implementación del listado de clases de movimiento a nivel de RMI, servicio web e interfaz gráfica: 2 puntos.

C. Implementación de la búsqueda/listado y selección de blancos o espectros a los que se pueden dirigir los movimientos a nivel de RMI, servicio web e interfaz gráfica: 4 puntos.

D. Implementación del registro de Pokémon a nivel de RMI, servicio web e interfaz gráfica (considerando registro de movimientos asociados y funcionamiento de los botones agregar/eliminar movimiento del front-end): 5 puntos.

E. Implementación de la búsqueda/listado y selección de Pokémon (considerando búsqueda/listado de los movimientos asociados): 4 puntos.

Prof. FREDDY PAZ

San Miguel, 27 de Noviembre del 2023

ANOTACIONES DE PROGRAMACIÓN

Para asignar la foto a la variable de tipo byte[] (C#)

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

Para mostrar una imagen en el PictureBox desde una variable de tipo byte[] (C#)

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

Para enviar la generación (enumerado) y el arreglo de bytes (foto) como parámetro mediante un comando a un procedimiento almacenado (JAVA)

```
cs.setString("_generacion", pokemon.getGeneracion().toString());  
cs.setBytes("_foto", pokemon.getFoto());
```

En caso se desee leer la generación (enumerado) y el arreglo de bytes (foto) a una variable pokemon (JAVA)

```
pokemon.setGeneracion(Generacion.valueOf(rs.getString("generacion")));  
pokemon.setFoto(rs.getBytes("foto"));
```

En caso de ocurrir algún desbordamiento de memoria en caso de trabajar con un cliente C#, modificar las siguientes líneas relacionadas al servicio en el App.config:

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

Fig. 03. Formulario de Gestión de Pokémon – Parte 01

Formulario de Gestión de Pokémon


Nuevo
Guardar
Buscar
Cancelar


Datos Generales
Movimientos por Nivel

Agregar Nuevo Movimiento por Nivel

LVL: 15

Nombre: POLVO VENENO

Tipo:  VENENO

Clase:  ESTADO

Poder: 0

Exactitud: 75

PP: 35

Blanco: ELEGIDO

Su efecto es contra un único Pokémon, oponente o aliado, a elegir.

	LV	NOMBRE	TIPO	CLASE	PODER	EXACT	PP	BLANCO
	1	PLACAJE	NORMAL	FISICO	35	95	35	OPONENTE ELEGIDO
▶	3	GRUÑIDO	NORMAL	ESTADO	0	100	40	ELEGIDO
	3	LATIGO CEPA	PLANTA	FISICO	45	100	25	ELEGIDO
	9	DRENADORAS	PLANTA	ESTADO	0	90	10	ELEGIDO
	12	HOJA AFILADA	PLANTA	FISICO	55	95	25	OPONENTES ADYACENTES

Fig. 04. Formulario de Gestión de Pokémon – Parte 02