

EXAMEN PARCIAL – UNIDADES 9, 10, 11 y 12

TRIMESTRE: TERCERO

Fecha:

CICLO: Desarrollo de Aplicaciones Web

CURSO: 1º

CALIFICACIÓN:

MÓDULO: Programación

Turno: Mañana

Nombre:

Apellidos:

Instrucciones: Esta prueba tiene como finalidad evaluar los aprendizajes de Programación. Lee atentamente y responde escribiendo el código más adecuado.

Si las instrucciones no se siguen como se especifican el examen no será evaluado

PARTE PRÁCTICA. TIPO A

- El examen práctico tiene una puntuación máxima de 10 puntos.
- Para superar la parte práctica se requiere alcanzar un mínimo de 5 puntos.

Caso práctico: "Santuario Fauna Libre"

En algún rincón del mundo, oculto entre la neblina tropical y el murmullo de las hojas, existe un santuario único: **un refugio de animales**. No es un zoológico, ni un parque natural, sino un último bastión de esperanza para criaturas escamosas en busca de una segunda oportunidad.

Desarrollar en Java un sistema para la gestión de un santuario de reptiles llamado Santuario, que permita registrar, actualizar, eliminar, listar y filtrar animales residentes, además de identificar cuáles pueden ser adoptados, según criterios específicos.

Estructura del sistema:

◆ 1. Clase abstracta Animal

Representa cualquier animal del santuario.

Atributos:

- String nombre
- int edad
- double peso

Métodos:

- getNombre(), getEdad(), getPeso(), setters correspondientes.
- String getTipo() → devuelve el tipo concreto del animal (por ejemplo, "Camaleón").
- toString() → muestra los datos del animal.
- Otros métodos comunes que consideres útiles.

Hereda de esta clase: Camaleon, Iguana, Serpiente, Tortuga.

Implementa la interfaz Adoptable.

◆ 2. Interfaz Adoptable

Define el contrato que deben cumplir todos los animales adoptables.

Método obligatorio:

- `boolean esAdoptable();`

◆ 3. Clases concretas que heredan de Animal

Cada clase debe implementar el método `esAdoptable()` según su propio criterio:

✓ Camaleon

- Atributo adicional: `double longitudLengua`
- Criterio: adoptable si `longitudLengua > 10`

✓ Iguana

- Atributo adicional: `int escamasDaniadas`
- Criterio: adoptable si `escamasDaniadas < 5`

✓ Serpiente

- Atributo adicional: `boolean venenoControlado`
- Criterio: adoptable si `venenoControlado == true` y `edad >= 2`

✓ Tortuga

- Atributos adicionales: `boolean esTerrestre`
- Criterio: adoptable si `esTerrestre == true` y `peso < 10`

◆ 4. Clase Santuario (lógica de negocio)

Mantiene los animales en memoria utilizando un `Set<Animal>` para evitar duplicados.

Métodos:

- `boolean altaAnimal(Animal a)`
- `boolean bajaAnimal(String tipo, String nombre)`
- `boolean actualizarAnimal(String tipo, String nombre, int nuevaEdad, double nuevoPeso)`
- `List<Animal> listarTodos()`
- `List<Animal> listarAdoptables()`
- `List<Animal> listarPorTipo(String tipo)`

Valida duplicados por combinación de tipo y nombre.

◆ 5. Clase AnimalDAO

Encargada del acceso a datos. Debe contener métodos para la persistencia en una base de datos MySQL, con consultas SQL dinámicas.

Métodos:

- `void insertarAnimal(Animal a)`
- `void eliminarAnimal(String tipo, String nombre)`
- `void actualizarAnimal(Animal a)`
- `List<Animal> cargarTodos()`

Puedes añadir otros métodos auxiliares si lo consideras necesario.

◆ 6. Clase AccesoBD

Encapsula el acceso a la base de datos MySQL.

Métodos:

- `public static Connection abrirConexion()`
- `public static void cerrarConexion(Connection c)`
- `public static ResultSet consultaBD(String sql)`
- `public static int actualizaBD(String sql)`

Maneja correctamente excepciones

✍️ Requisitos del examen

Implementa en Java las clases mencionadas y asegúrate de:

1. Utilizar herencia, polimorfismo, clases abstractas e interfaces.
2. Manejar correctamente las colecciones para evitar duplicados.
3. Gestionar adecuadamente excepciones y recursos en la conexión con la base de datos.
4. Demostrar que sabes **pensar como un desarrollador que cuida reptiles... pero escribe código elegante.**

Nota: Se os proporciona el script de creación de la base de datos con la tabla necesaria y datos de un animal de cada tipo.

Ejemplo de ejecución

```
--- Menú Santuario Fauna Libre ---
1) Alta de animal
2) Baja de animal
3) Listar todos
4) Listar adoptables
5) Listar por tipo
6) Actualizar datos
0) Salir
Opción: 3
Serpiente - Pepa (Edad: 18, Peso: 30.0kg)
Tortuga - Micaela (Edad: 67, Peso: 25.0kg)
Camaleon - pepito (Edad: 2, Peso: 1.0kg)
Iguana - Ana (Edad: 3, Peso: 3.0kg)

--- Menú Santuario Fauna Libre ---
1) Alta de animal
2) Baja de animal
3) Listar todos
4) Listar adoptables
5) Listar por tipo
6) Actualizar datos
0) Salir
Opción: 4
Iguana - Ana (Edad: 3, Peso: 3.0kg)

--- Menú Santuario Fauna Libre ---
1) Alta de animal
2) Baja de animal
3) Listar todos
4) Listar adoptables
5) Listar por tipo
6) Actualizar datos
0) Salir
Opción: 1
Tipo (Camaleon/Iguana/Serpiente/Tortuga): Iguana
Nombre: Mariana
Edad: 8
Peso: 2
Escamas dañadas: 2
Añadido: Iguana - Mariana (Edad: 8, Peso: 2.0kg)

--- Menú Santuario Fauna Libre ---
1) Alta de animal
2) Baja de animal
3) Listar todos
4) Listar adoptables
5) Listar por tipo
6) Actualizar datos
0) Salir
Opción: 5
Tipo: Iguana
Iguana - Ana (Edad: 3, Peso: 3.0kg)
Iguana - Mariana (Edad: 8, Peso: 2.0kg)
```

--- Menú Santuario Fauna Libre ---

- 1) Alta de animal
- 2) Baja de animal
- 3) Listar todos
- 4) Listar adoptables
- 5) Listar por tipo
- 6) Actualizar datos
- 0) Salir

Opción: 6

Nombre: Mariana

Tipo: Iguana

Nueva edad: 6

Nuevo peso: 2

Actualizado: Iguana - Mariana (Edad: 6, Peso: 2.0kg)

--- Menú Santuario Fauna Libre ---

- 1) Alta de animal
- 2) Baja de animal
- 3) Listar todos
- 4) Listar adoptables
- 5) Listar por tipo
- 6) Actualizar datos
- 0) Salir

Opción: 2

Nombre: Mariana

Tipo: Iguana

Eliminado: Mariana (Iguana)

--- Menú Santuario Fauna Libre ---

- 1) Alta de animal
- 2) Baja de animal
- 3) Listar todos
- 4) Listar adoptables
- 5) Listar por tipo
- 6) Actualizar datos
- 0) Salir

Opción: 3

Serpiente - Pepa (Edad: 18, Peso: 30.0kg)

Tortuga - Micaela (Edad: 67, Peso: 25.0kg)

Camaleon - pepito (Edad: 2, Peso: 1.0kg)

Iguana - Ana (Edad: 3, Peso: 3.0kg)

--- Menú Santuario Fauna Libre ---

- 1) Alta de animal
- 2) Baja de animal
- 3) Listar todos
- 4) Listar adoptables
- 5) Listar por tipo
- 6) Actualizar datos
- 0) Salir

Opción: 0

Guardando y saliendo...

Rúbrica

Criterio de evaluación	Nivel Excelente (Puntuación completa)	Nivel Aceptable (Puntuación media)	Nivel Insuficiente (0 puntos)
1. Implementación de la clase abstracta, interfaz y subclases	Clase abstracta, interfaz y subclases correctamente diseñadas, con herencia y polimorfismo bien aplicados – 2 puntos	Estructura funcional, aunque con errores menores en jerarquía o sintaxis – 1 punto	Estructura incompleta o con errores graves conceptuales – 0 puntos
2. Gestión de animales (alta, baja, consulta)	Métodos de inserción, eliminación y consulta implementados correctamente sobre la colección o la base de datos – 2 puntos	Métodos presentes pero con errores funcionales o falta de validaciones – 1 punto	Métodos ausentes o con funcionamiento incorrecto – 0 puntos
3. Filtrado de datos (exhibibles, por tipo, etc.)	Filtrado bien implementado con lógica coherente, uso de streams o filtros apropiados – 1,5 puntos	Filtrado funcional pero con código poco optimizado o confuso – 0,75 puntos	No se implementan métodos de filtrado o fallan – 0 puntos
4. Modificación de datos en la base de datos	La actualización de datos en BD se realiza correctamente con SQL parametrizado y control de errores – 1,5 puntos	Actualización funcional pero con mala gestión de errores o malas prácticas – 0,75 puntos	No se realiza modificación o produce errores graves – 0 puntos
5. Implementación del patrón DAO	El patrón DAO está correctamente estructurado con separación de responsabilidades, clases DAO específicas, y operaciones CRUD completas – 2 puntos	DAO presente pero con mezcla de responsabilidades o diseño incompleto – 1 punto	No se aplica el patrón DAO o está mal estructurado – 0 puntos
6. Método main con menú interactivo	Menú funcional, intuitivo, con control de errores y ejecución de opciones – 1 punto	Menú básico funcional pero sin control de errores o poco amigable	Menú incompleto o no interactivo – 0 puntos