

EXAMEN PRESENCIAL

PRIMER TRIMESTRE

Fecha: 12/12/2022

| | | |
|---|-------------------------------|---------------|
| C.S. DESARROLLO DE APLICACIONES MULTIPLATAFORMA | | CALIFICACIÓN: |
| MÓDULO: ACCESO A DATOS | | |
| Turno: 2º CURSO | | |
| Nombre: Juan Antonio | Apellidos: Enriquez Bustos | |

PARTE TEÓRICA

Instrucciones: Esta prueba tiene como finalidad evaluar los aprendizajes de la asignatura de Acceso a datos

- El examen consiste en 22 preguntas de tipo test.
 - Cada pregunta correcta se puntúa con 1 punto.
 - Las preguntas incorrectas se puntúan con 0 puntos.
 - Las preguntas no contestadas se puntúan con 0 puntos.
- La fórmula de cálculo es: **nota = 0.45 * acertadas**
- Se dispone de 30 min como máximo para poder realizar el examen, tras los cuales se deberá entregar automáticamente.
- Responder en esta misma hoja, marcando con un círculo la respuesta seleccionada.
- Para rectificar una respuesta, tachar la incorrecta y seleccionar con un círculo la correcta.

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

| | | | | | | | | | |
|----|---|---|---|---|----|---|---|---|---|
| 1 | A | B | C | D | 12 | A | B | C | D |
| 2 | A | B | C | D | 13 | A | B | C | D |
| 3 | A | B | C | D | 14 | A | B | C | D |
| 4 | A | B | C | D | 15 | A | B | C | D |
| 5 | A | B | C | D | 16 | A | B | C | D |
| 6 | A | B | C | D | 17 | A | B | C | D |
| 7 | A | B | C | D | 18 | A | B | C | D |
| 8 | A | B | C | D | 19 | A | B | C | D |
| 9 | A | B | C | D | 20 | A | B | C | D |
| 10 | A | B | C | D | 21 | A | B | C | D |
| 11 | A | B | C | D | 22 | A | B | C | D |

PARTE PRÁCTICA

Clona el repositorio <https://github.com/FRomero999/ExamenAD2022.git> en local y completa los siguientes ejercicios:

1) Completar el método **estadísticasDeArchivo**, que lea un archivo de texto con un texto y lo analice, mostrando por pantalla el número de caracteres, el número de líneas y el número de palabras. Incluir todos los paquetes que sean necesarios, pero no cambiar la definición del método.

(2,5 puntos)

```
Estadísticas de pom.xml:
14 líneas
603 caracteres
20 palabras
```

```
EJERCICIO 1: Archivos
```

```
-----

Estadísticas de pom.xml:
27 líneas
966 caracteres
193 Palabras
--Ejercicio resuelto--
```

2) Queremos gestionar una pequeña biblioteca compartida usando una base de datos MySQL gestionada por un programa en Java (se adjunta el script de creación de la base de datos).

| examen usuario |
|-------------------------|
| id : int(11) |
| nombre : varchar(64) |
| apellidos : varchar(64) |
| dni : varchar(16) |

| examen libro |
|-----------------------|
| id : int(11) |
| titulo : varchar(128) |
| autor : varchar(128) |

| examen ejemplar |
|----------------------|
| id : int(11) |
| estado : varchar(16) |
| # edicion : int(11) |
| # libro_id : int(11) |

Completa los métodos que aparecen en la clase **UsuarioDAO** para gestionar el listado de usuarios de la misma, usando exclusivamente JDBC.

```
public void connect() {...11 lines }

public void close() {...7 lines }

public void save( Usuario user) {...14 lines }

public ArrayList<Usuario> list() {...10 lines }

public Usuario getByDNI(String dni) {...13 lines }
```

(3,25 puntos)

EJERCICIO 2: JDBC

```
-----

Conexión establecida correctamente.
El usuario Francisco Romero
con DNI: 42352343F
ha sido añadido satisfactoriamente
--Método save completado--
```

El usuario con DNI 42352343F es:
 Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}

```
--
-----
El total de usuarios es: 16
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
Usuario{id=null, nombre=Francisco, apellidos=Romero, dni=42352343F}
```

El usuario con DNI 42352343F es:
 Usuario{id=null, nombre=Francisco, apellidos=apellidos, dni=42352343F}

El usuario con DNI 00000000F es:
 No hay un registro para ese dni
 null

3) Queremos gestionar los libros en la biblioteca teniendo en cuenta que **no es lo mismo un libro que un ejemplar**.

De un mismo libro podemos tener varios ejemplares, cada uno en un estado diferente, o de una edición diferente. Por ello vamos a utilizar Hibernate para realizar la gestión.

Debéis modificar la clase **BibliotecaDAO** para implementar los métodos incompletos, las clases POJO para añadir la relación y el archivo de configuración de Hibernate.

```
15 public class BibliotecaDAO {
16
17     private static SessionFactory sessionFactory;
18
19     {...13 lines }
32
33     public void saveLibro( Libro e ) {...13 lines }
46
47     public HashSet<Libro> findByEstado( String estado ) {...19 lines }
66
67     public void printInfo() {...19 lines }
86
87 }
```

También debéis modificar la clase Ejercicio3 para poder probar el correcto funcionamiento del programa y asociar los ejemplares creados a un Libro.

(4,25 puntos)

Subid vuestra solución a Github en un nuevo repositorio y adjuntar exclusivamente su enlace (nada de archivos zip, rar o 7z).

Añadid también varias capturas de pantalla donde se vea la salida del programa.

Además del correcto funcionamiento, se valorará en vuestra solución la documentación y calidad del código propuesto.

Si se detecta el más mínimo indicio de plagio, copia o uso de herramientas como CoPilot de forma indebida, los exámenes involucrados quedarán descalificados.

Se prohíbe el uso del chat de teams durante el transcurso del examen. Si se detecta su uso, se retirará el examen de los alumnos implicados.

Autoevaluación

Nota esperada del
ejercicio práctico:

5,75

Examen.sql

```
CREATE TABLE `ejemplar` (  
  `id` int(11) NOT NULL,  
  `estado` varchar(16) NOT NULL,  
  `edicion` int(11) NOT NULL,  
  `libro_id` int(11) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
CREATE TABLE `libro` (  
  `id` int(11) NOT NULL,  
  `titulo` varchar(128) NOT NULL,  
  `autor` varchar(128) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
CREATE TABLE `usuario` (  
  `id` int(11) NOT NULL,  
  `nombre` varchar(64) NOT NULL,  
  `apellidos` varchar(64) NOT NULL,  
  `dni` varchar(16) NOT NULL  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
  
ALTER TABLE `ejemplar`  
ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `libro`  
ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `usuario`  
ADD PRIMARY KEY (`id`);  
  
ALTER TABLE `ejemplar`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
  
ALTER TABLE `libro`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
  
ALTER TABLE `usuario`  
MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;  
COMMIT;
```