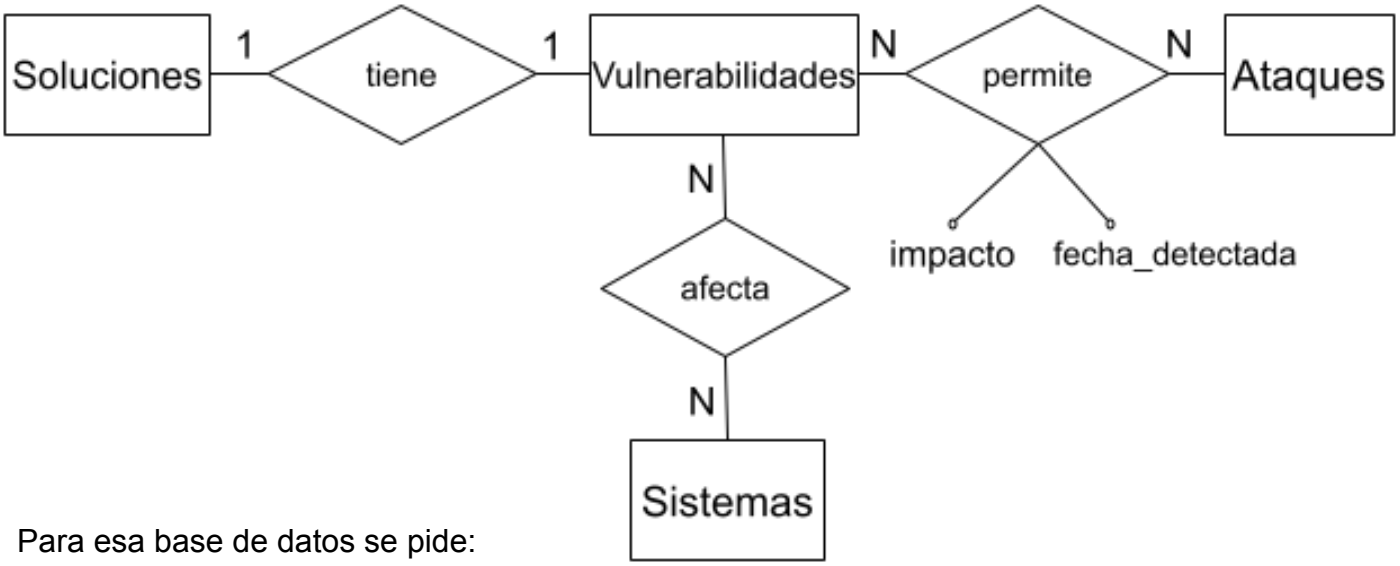
Acceso a Datos Ciclo Superior de Desarrollo de Aplicaciones Multiplataforma	2ª Evaluación	Fecha:	31/03/2025	Hora:	19:00

Apellidos:		Nombre:	
DNI:		Hora salida:	

EJERCICIO 1 (10 puntos)

Realiza un programa Java que utilice la base de datos (*vulnerabilidades_db.sql*) la cual tiene el siguiente diagrama Entidad-Relación:



Para esa base de datos se pide:

a. Validar la estructura de la base de datos, representando todas las relaciones y sus atributos **(2.5 puntos)**

b. Realizar un menú que permite las siguientes opciones **(7.5 puntos)**:

1. Crear una vulnerabilidad. **(0.5 puntos)**

2. Asigna una vulnerabilidad a un sistema. **(1 punto)**
   - La asignación será bidireccional.
   - Ej.: Vulnerabilidad id 5 (Phishing) afecta a Sistema id 1 (Windows 10)

3. Asigna una vulnerabilidad a un vector de ataque para un día concreto. **(1 punto)**
   - La asignación será bidireccional.
   - La fecha puede usarse **Localdate.now()** o cualquier otra opción que conozcas
   - Ej.: Vulnerabilidad 2 (Cross-Site Scripting) permite Ataque 4 (Phishing Masivo)

c. **Consultas**: El resultado de las siguientes consultas tendrá que obtenerse directamente de la propia consulta. No se podrá usar java para filtrar los datos.

1. **Consulta 1:** Obtener el nombre, descripción y nivel de riesgo de una vulnerabilidad a partir de su id. **(1.25 puntos)**

2. **Consulta 2**: Listar la descripción de la solución de la vulnerabilidad "Ransomware" *(Solución: Implementar copias de …).* **(1.25 puntos)**

3. **Consulta 3**: Contar cuántas vulnerabilidades de *nivel_riesgo* 4 tiene el sistema "Windows 10" *(Solución: 2)* **(1.25 puntos)**

4. **Consulta 4**: Listar todas las vulnerabilidades con sus ataques permitidos, pero sólo si el impacto del ataque es mayor o igual a 3 *(Solución: 4 elementos)* **(1.25 puntos)**

EJERCICIO 2 (4 puntos)

Dada la siguiente base de datos XML denominada *universidad (universidad.xml)* se pide realizar los siguientes apartados en Java. Para ello se utilizará la plantilla denominada *ejercicio2_3.zip* **NOTA: el resultado de las consultas debe mostrar lo solicitado de forma directa sin necesidad de realizar ningún tipo de filtrado con Java:**

1. **Consulta 1**: Contar el número de profesores que hay en un departamento dado.. *Solución: para el departamento Psicología → 2 profesores* **(1.25 punto)**

2. **Consulta 2**: Media de antigüedad de los profesores de la facultad de *Ingeniería* que trabajan algún *Lunes*. *Solución: 7.666…* **(1.25 punto)**

3. **Consulta 3**: Número medio de cursos ofrecidos por cada facultad. *Solución: 3.66…* **(1.5 puntos)**

EJERCICIO 3 (6 puntos)

Dada la siguiente base de datos MongoDB denominada *biblioteca (biblioteca.js*) se pide realizar en Java los siguientes apartados. **NOTA: el resultado de las consultas debe mostrar lo solicitado de forma directa sin necesidad de realizar ningún tipo de filtrado con Java**:

1. **Consulta 1:** Lista los autores que han escrito libros del género *Ficción histórica* y devolver el nombre, la edad y los premios pero no el ID. *Solución: Isabel Allende* **(1.5 puntos)**

2. **Consulta 2:** Lista el nombre de los autores mayores de 60 años que han recibido el *"Premio Nobel de Literatura". Solución: Gabriel García Márquez* **(1.75 puntos)**

3. **Consulta 3:** Crea una agregación que permita obtener la cantidad total de ventas de libros de todos los autores. *Solución: {"total_ventas": 324000000}* **(1.25 puntos)**

4. Eliminar los autores cuya nacionalidad sea *"Japonesa"* y que no hayan recibido ningún premio. *Solución: un elemento eliminado* **(1.5 puntos)**

ANEXO:

Mensaje de menú (Ejercicio 1)

```
String mensaje = "1. Crear vulnerabilidad\n" +
        "2. Asignar vulnerabilidad - sistema\n" +
        "3. Asignar vulnerabilidad - ataque\n" +
        "4. Consulta 1\n" +
        "5. Consulta 2\n" +
        "6. Consulta 3\n" +
        "7. Consulta 4\n" +
        "8. Salir";
```

Función de lectura de un String:

```
public static String pedirString(String mensaje){
        System.out.println(mensaje);
        return sc.next();
}
```