REPORTE DE VULNERABILIDADES SQLi

Resumen del análisis

archivos analizados: 5

lineas afectadas: 8

tiempo de analisis: 1.13

Archivo: src\businesslayer\AppData.java

Línea: 54

```
try {
   conn.setAutoCommit(false);
```

Detalles:

- [CRÍTICO] SQLi por uso de parámetro no validado: Se usa la variable 'person' directamente en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL si no se valida correctamente.
- [CRÍTICO] SQLi por concatenación: La variable 'person' contaminada se concatena en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL.
- [CRÍTICO] Violación de arquitectura N-capas: En la capa 'LOGICA' no está permitido ejecutar sentencias SQL directamente. Las operaciones SQL deben realizarse solo en la capa de datos.

Línea: 57

```
stmt = conn.createStatement();
                String info = "INSERT INTO PERSON (FIRSTNAME,LASTNAME,DATEOFBIRTH,AGE,EMAIL,PHONENUMBER,ADDRESS1,ADDRESS2,CITY,ZIP,STATE,COUNTRY) "
     + "VALUES ('" + person.getFirstName().toString() + "','" + person.getLastName().toString() + "','"
```

Detalles:

- [CRÍTICO] SQLi por uso de parámetro no validado: Se usa la variable 'person' directamente en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL si no se valida correctamente.

Línea: 78

```
Person person = null;
  try {
   conn = DatabaseConnection.getConnection();
```

Detalles:

- [CRÍTICO] SQLi por uso de parámetro no validado: Se usa la variable 'person' directamente en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL si no se valida correctamente.
- [CRÍTICO] SQLi por concatenación: La variable 'person' contaminada se concatena en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL.
- [CRÍTICO] Violación de arquitectura N-capas: En la capa 'LOGICA' no está permitido ejecutar sentencias SQL directamente. Las operaciones SQL deben realizarse solo en la capa de datos.

Línea: 83

```
ResultSet result = stmt.executeQuery(
   "SELECT * FROM PERSON WHERE (FIRSTNAME = '" + firstName + "') and (LASTNAME = '" + lastName + "')");
```

Detalles:

- [CRÍTICO] SQLi por uso de parámetro no validado: Se usa la variable 'firstName' directamente en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL si no se valida correctamente.

Línea: 103

```
Statement stmt = null;
  try {
   conn = DatabaseConnection.getConnection();
```

Detalles:

- [CRÍTICO] SQLi por uso de parámetro no validado: Se usa la variable 'firstName' directamente en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL si no se valida correctamente.
- [CRÍTICO] SQLi por concatenación: La variable 'firstName' contaminada se concatena en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL.
- [CRÍTICO] Violación de arquitectura N-capas: En la capa 'LOGICA' no está permitido ejecutar sentencias SQL directamente. Las operaciones SQL deben realizarse solo en la capa de datos.

Línea: 108

```
String deleteQuery = "DELETE FROM PERSON WHERE (FIRSTNAME = '" + firstName + "') and (LASTNAME = '"
    + lastName + "');";
```

Detalles:

- [CRÍTICO] SQLi por uso de parámetro no validado: Se usa la variable 'firstName' directamente en una sentencia SQL en la capa 'LOGICA'. Esto puede permitir inyección SQL si no se valida correctamente.

Línea: 123

```
Statement stmnt = null;
  try {
   conn = DatabaseConnection.getConnection();
```

Detalles:

- [CRÍTICO] Violación de arquitectura N-capas: En la capa 'LOGICA' no está permitido ejecutar sentencias SQL directamente. Las operaciones SQL deben realizarse solo en la capa de datos.

Archivo: src\datalayer\DatabaseConnection.java

Línea: 45

```
Statement stmt = conn.createStatement();
      String sql = "SELECT * FROM PERSON WHERE FIRSTNAME = '" + userInput + "'";
      stmt.executeQuery(sql);
```

Detalles:

- [CRÍTICO] SQLi por uso de parámetro no validado: Se usa la variable 'userInput' directamente en una sentencia SQL en la capa 'DATOS'. Esto puede permitir inyección SQL si no se valida correctamente.