Enunciado:

Con el objetivo de aumentar la seguridad se pide agregar una clave de acceso a una base de datos Access, la clave a utilizar es “**oop**” y el nombre de la base de datos es “**BibliotecaDB.mdb**”.

Para agregar una clave debemos abrir la base de datos en modo exclusivo “**Open Exclusive**”, como se ve en la **Figura 1** y luego establecer la clave con el icono que muestra la **Figura 2**, una vez establecida la clave la siguiente ves que intentemos abrirla con la aplicación Microsoft Access, nos la pedirá tal como muestra la **Figura 3**.

En la **Figura 4**, podemos ver el archivo “**App.config**” de nuestra aplicación, con el atributo **connectionString** codificado en base64, para aumentar la seguridad y que no aparezca la clave “oop” en texto plano de simple lectura. Solo hemos dejado un comentario, color verde, a los fines académicos para que podamos ver el texto decodificado cosa que evidentemente no deberíamos hacer en una aplicación productiva real.

La **Figura 5**, muestra la interfaz de usuario de la aplicación, en ella podemos ver la cadena de conexión a la base de datos escrita en dos formas, **codificada** y **sin codificar**, en el archivo de configuración (“**App.config**”) se escribe solo en formato codificado, siendo la aplicación la responsable de decodificarla, más precisamente la clase “Codificador” cuyo código podemos ver en la **Figura 6**.

La clase “Codificador” solo tiene dos métodos una que codifica y otro que decodifica, ver **Figura 6**.

Existen varios sitios web que permiten codificar y decodificar base64, uno de ellos es <http://base64encode.net/> pero podría utilizarse cualquier otro para realizar el proceso de codificación de la cadena de conexión y su posterior escritura en el archivo “**App.config**”.

La **Figura 7**, muestra los 64 códigos ASCII, que conforma el código base64 aunque básicamente son a-z, A-Z y 0-9 para los primeros 62 valores. Otras variaciones Base64 comparten la misma propiedad pero utilizan diferentes símbolos en los dos últimos valores.

El resto de la aplicación utiliza la clase “LibroPersistidor” para realizar el CRUD (create, read, update and delete), de este modo es transparente o desconocido el tema de la seguridad ya que la clase se encarga de ocultar la implementación de la misma.

|  |
| --- |
| Figura 1 |
|  |

|  |
| --- |
| Figura 2 |
|  |

|  |
| --- |
| Figura 3 |
|  |

|  |
| --- |
| Figura 4 |
|  |

|  |
| --- |
| Figura 5 |
|  |

|  |
| --- |
| Figura 6 |
|  |

|  |
| --- |
| Figura 7 |
|  |