

ACCESO A DATOS - T2 - HITO GRUPAL



Isla P, Sergio C, Mario L y Mario V

17/01/2023

ÍNDICE

Tecnologías utilizadas.	2
Interfaz de la aplicación (modo consola)	3
Demostración de la ejecución del programa y funcionamiento.	5
Explicación de las partes más relevantes del código.	11
Rúbrica	18

Tecnologías utilizadas.

- Base de datos: Para este trabajo se nos pedía utilizar MongoDB. Hay 2 formas de utilizarlo por consola o por interfaz gráfica, nosotros hemos elegido la última opción ya que nos pareció que nos facilitaba mucho más.
- Lenguaje: Como lenguaje nos requieren utilizar Java con el gestor de dependencias 'Maven'. Hemos decidido utilizar IntelliJ IDEA Ultimate ya que las prestaciones de este son mejores y nos permite utilizar "code with me" para que más de uno pueda trabajar en el mismo proyecto.

Versiones :

- JDK : 18
- Mongo-Java-Driver : 3.12.11
- Maven : 17

Dependencia 'mongo-java-driver' para establecer conexión con una base de datos MongoDB.

Clases:

- 'mongoClient' para establecer una conexión entre un cliente y el servidor con la base de datos,
- 'mongoDatabase' para obtener datos orientados a conexión,
- 'mongoCollection' para obtener la colección de la cual queremos utilizar sus datos.
- 'Document' para introducir los datos en la base de datos.

Interfaz de la aplicación (modo consola)

Primero mostramos un menú con el cual el usuario puede identificarse como 'Técnico', 'Dirección' o salir del programa.

```
-----  
Elige el usuario  
1 : Tecnico  
2 : Direccion  
3 : Salir
```

En cuanto escoges una opción de tipo de usuario te pedirá una credenciales. Si las credenciales son correctas, se inicia sesión y muestra las opciones de las que dispone el tipo de usuario, si las credenciales fallan nos da una opción de seguir intentando, si decimos 'n' (no) salimos del programa.

```
Escribe el nombre :  
tecnico  
Escribe la contraseña :  
curso123  
Sesion iniciada
```

```
Escribe el nombre :  
a  
Escribe la contraseña :  
a  
Usuario tecnico incorrecto  
¿Desea continuar? - s/n  
s  
Escribe el nombre :
```

Las opciones de las cuales dispone cada tipo de usuario son las siguientes :

```
Bienvenido a tecnicos
-----
Operaciones disponibles :
Elige el numero de la opci
1 : Añadir cliente
2 : Consultar cliente
3 : Modificar cliente
4 : Eliminar cliente
5 : Añadir incidencia
6 : Consultar incidencia
7 : Modificar incidencia
8 : Eliminar incidencia
9 : Salir de tecnicos
```

```
Bienvenido a direccion
Operaciones disponibles :
1 : Añadir incidencia
2 : Consultar incidencia
3 : Modificar incidencia
4 : Eliminar incidencia
5 : Salir
```

Demostración de la ejecución del programa y funcionamiento.

- Identificación del usuario

```
¿Especifica tu puesto : (técnicos o dirección)?  
Elige el numero de la opcion  
1 : 'Tecnicos'  
2 : 'Dirección'  
3 : 'Salir'
```

- Inicio sesión

```
Escribe el nombre :  
tecnico  
Escribe la contraseña :  
curso123  
Sesion iniciada  
Bienvenido a tecnicos
```

- Opciones de los Técnicos

```
Operaciones disponibles : (Añadir - Consultar - Modificar - Eliminar)  
Elige el numero de la opcion  
1 : Añadir cliente  
2 : Consultar cliente  
3 : Modificar cliente  
4 : Eliminar cliente  
5 : Añadir incidencia  
6 : Consultar incidencia  
7 : Modificar incidencia  
8 : Eliminar incidencia  
9 : Salir de tecnicos
```

Técnicos 1. Añadir cliente

- Programa

```
Nombre de la persona :  
Jaun  
Apellidos de la persona :  
Martínez  
Producto de la persona :  
Movil  
¿Añadir incidencia? (si / no)  
si  
Dime la fecha de la incidencia :  
Formato : dd/mm/aaaa  
12/12/2012  
Dime el motivo de la incidencia :  
Pantalla rota  
Tipo de la incidencia :  
Ruptura  
¿Necesito reparacion fisica el producto?  
Si  
¿La llamada soluciono su problema?  
No  
Operador que le atendio :  
Luis Garcia  
INCIDENCIA AÑADIDA CORRECTAMENTE
```

- Base de datos

```
_id: ObjectId('63d8fa29a11df536423b5f2f')  
id_cliente: 5  
nombre_cliente: "Jaun"  
apellidos_cliente: "Martínez"  
producto_cliente: "Movil"  
id_incidencia: 4  
idCliente_incidencia: 5  
fecha_incidencia: "12/12/2012"  
motivo_incidencia: "Pantalla rota"  
tipo_incidencia: "Ruptura"  
reparacionFisica_incidencia: "Si"  
solucionDeLlamada: "No"  
operador_incidencia: "Luis Garcia"
```

Técnicos 2. Consultar clientes

```
Document({_id=63d25ca88979a039c0903756, id_cliente=1, nombre_cliente=pepe,
apellidos_cliente=Perez Sanchez, producto_cliente=Xiaomi Poco X3,
indicencia_cliente=Document({_id_incidencia=1, fecha_incidencia=19/01/2023,
motivo_incidencia=A mi dispositivo movil no le funciona la pantalla,
tipo_incidencia=Hardware, reparacionFisica_incidencia=Si,
solucionDeLlamada=Si, operador_incidencia=Mario
L}}})Document({_id=63d2654ea2799036ab9c5fc9, id_cliente=2,
nombre_cliente=luis, apellidos_cliente=martinez, producto_cliente=weiu,
id_incidencia=2, fecha_incidencia=08/08/2008, motivo_incidencia=ewuf,
tipo_incidencia=piwe, reparacionFisica_incidencia=si, solucionDeLlamada=no,
operador_incidencia=manolo})Document({_id=63d8fa29a11df536423b5f2f,
id_cliente=5, nombre_cliente=Jaun, apellidos_cliente=Martinez,
producto_cliente=Movil, id_incidencia=4, idCliente_incidencia=5,
fecha_incidencia=12/12/2012, motivo_incidencia=Pantalla rota,
tipo_incidencia=Ruptura, reparacionFisica_incidencia=Si, solucionDeLlamada=No,
operador_incidencia=Luis Garcia}}
```


Técnicos 3 : Modificar cliente

- Base de datos antes

```
_id: ObjectId('63d8fa29a11df536423b5f2f')
id_cliente: 5
nombre_cliente: "Jaun"
apellidos_cliente: "Martinez"
producto_cliente: "Movil"
id_incidencia: 4
idCliente_incidencia: 5
fecha_incidencia: "12/12/2012"
motivo_incidencia: "Pantalla rota"
tipo_incidencia: "Ruptura"
reparacionFisica_incidencia: "Si"
solucionDeLlamada: "No"
operador_incidencia: "Luis Garcia"
```

- Programa

```
Se ha conectado correctamente
Dime el ID del cliente a modificar :
5
Dime el nombre del cliente :
Juan
Dime los apellidos del cliente :
Martínez
Dime el producto del cliente :
Movil: iphone 12 pro
Dime el ID de la incidencia :
4
Dime el ID del cliente
5
Dime la fecha de la incidencia :
Formato : dd/mm/aaaa
12/12/2012
Dime el motivo de la incidencia :
Pantalla rota
Dime el tipo de la incidencia :
Rotura
¿Necesita reparacion fisica el producto?
SI
¿La llamada soluciono el problema?
No
Operador que le atendio :
Luis García
```

- Base de datos después

```
_id: ObjectId('63d8fa29a11df536423b5f2f')
id_cliente: 5
nombre_cliente: "Juan"
apellidos_cliente: "Martínez"
producto_cliente: "Movil: iphone 12 pro"
id_incidencia: 4
idCliente_incidencia: 5
fecha_incidencia: "12/12/2012"
motivo_incidencia: "Pantalla rota"
tipo_incidencia: "Rotura"
reparacionFisica_incidencia: "SI"
solucionDeLlamada: "No"
operador_incidencia: "Luis García"
```

Técnicos 4 : Eliminar cliente

- Base de datos antes

```
_id: ObjectId('63d8fa29a11df536423b5f2f')
id_cliente: 5
nombre_cliente: "Jaun"
apellidos_cliente: "Martinez"
producto_cliente: "Movil"
id_incidencia: 4
idCliente_incidencia: 5
fecha_incidencia: "12/12/2012"
motivo_incidencia: "Pantalla rota"
tipo_incidencia: "Ruptura"
reparacionFisica_incidencia: "Si"
solucionDeLlamada: "No"
operador_incidencia: "Luis Garcia"
```

- Programa

```
ID del cliente a eliminar :
5
Cliente borrado correctamente
```

- Base de datos después:

```
▶ _id: ObjectId('63d25ca88979a039c0903756')  
id_cliente: "1"  
nombre_cliente: "pepe"  
apellidos_cliente: "Perez Sanchez"  
producto_cliente: "Xiaomi Poco X3"  
▸ incidencia_cliente: Object
```

```
_id: ObjectId('63d2654ea2799036ab9c5fc9')  
id_cliente: "2"  
nombre_cliente: "luis"  
apellidos_cliente: "martinez"  
producto_cliente: "weiu"  
id_incidencia: 2  
fecha_incidencia: "08/08/2008"  
motivo_incidencia: "ewuf"  
tipo_incidencia: "piwe"  
reparacionFisica_incidencia: "si"  
solucionDeLlamada: "no"  
operador_incidencia: "manolo"
```

Explicación de las partes más relevantes del código.

En este apartado **se mostrarán solo las funciones más importantes** del código CRUD. El programa completo estará adjunto en la entrega de la tarea. La lógica seguida por los técnicos y la dirección.

Los JSON con los que hemos creado las colecciones en MongoDB están situados en una carpeta dentro del proyecto.

- Conexión con MongoDB :

Abrir conexión con la base de datos de datos MongoDB, en caso de no poder encontrar el puerto enviará mensaje de error. Este método lo utilizamos cada vez que queremos realizar una operación.

```
private static MongoClient crearConexion() {
    MongoClient mongoClient = null;
    String servidor="localhost";
    Integer puerto = 27017;
    try{
        mongoClient = new MongoClient(servidor,puerto);
        System.out.println("Se ha conectado correctamente");
    }catch(Exception e){
        System.out.println("Error en la conexión");
        e.printStackTrace();
    }
    return mongoClient;
} //CIERRA CREAR CONEXIÓN
```

- Añadir una nueva incidencia :

Hemos creado un método llamado "insertarIncidencia". Este método utiliza un objeto de tipo "Scanner" para leer datos de entrada del usuario a través de la consola. Luego, se conecta a una base de datos MongoDB llamada "watermelon" y recupera la colección "incidencias".

A continuación, el programa le pregunta al usuario por un "ID de incidencia" y almacena el valor en una variable llamada "id_incidencia". Luego, se crea un nuevo objeto de la interfaz "Document" y se agrega un campo "id_incidencia" con el valor almacenado en la variable "id_incidencia". Para el resto de valores de la base de datos empleamos la misma estructura mencionada anteriormente.

```
public static void insertarIncidencia(){
    Scanner lector = new Scanner(System.in);
    //insertar incidencia
    MongoClient mongoClient = crearConexion();
    MongoDBDatabase database = mongoClient.getDatabase("watermelon");
    MongoCollection<Document> collection =
database.getCollection("incidencias");
    System.out.println("ID de incidencia : ");
    int id_incidencia= Integer.parseInt(lector.nextLine());
    Document doc = new Document("id_incidencia",id_incidencia);
    doc.append("id_incidencia",id_incidencia);
    System.out.println("ID del cliente : ");
    int idCliente_incidencia= Integer.parseInt(lector.nextLine());
    doc.append("idCliente_incidencia",idCliente_incidencia);
    System.out.println("Dime la fecha de la incidencia : "+"\\n"+"Formato :
dd/mm/aaaa");
    String fecha_incidencia=lector.nextLine();
    doc.append("fecha_incidencia",fecha_incidencia);
    System.out.println("Dime el motivo de la incidencia : ");
    String motivo_incidencia=lector.nextLine();
    doc.append("motivo_incidencia",motivo_incidencia);
    System.out.println("Tipo de la incidencia : ");
    String tipo_incidencia=lector.nextLine();
    doc.append("tipo_incidencia",tipo_incidencia);
    System.out.println("¿Necesito reparacion fisica el producto?");
    String reparacionFisica_incidencia=lector.nextLine();
    doc.append("reparacionFisica_incidencia",reparacionFisica_incidencia);
    System.out.println("¿La llamada soluciono su problema?");
    String solucion_incidencia=lector.nextLine();
    doc.append("solucionDeLlamada",solucion_incidencia);
    System.out.println("Operador que le atendio : ");
    String operador_incidencia=lector.nextLine();
    doc.append("operador_incidencia",operador_incidencia);
    collection.insertOne(doc);
    System.out.println("Incidencia insertada correctamente");
    mongoClient.close();
}
```

- Consultar los datos de la base de datos.

Primero, se llama al método "crearConexion" para conectarse a la base de datos "watermelon". Luego, se recupera la colección "incidencias" y se utiliza un cursor de MongoDB para recorrer todos los 'Document' en la colección. Finalmente, se imprime los datos que hemos obtenido de la base de datos. En caso de que ocurra algún error, se muestra un error en consola.

```
public static void consultarIncidencia(){
    MongoClient mongoClient = crearConexion();
    try{
        MongoDB database = mongoClient.getDatabase("watermelon");
        MongoCollection<Document> coleccion =
        database.getCollection("incidencias");
        try (MongoCursor< Document > cur = coleccion.find().iterator()) {
            while (cur.hasNext()) {
                System.out.printf(String.valueOf(cur.next()));
            }
        }
        System.out.println("Esos son los datos de las incidencias");
    }catch(Exception e){
        e.printStackTrace();
    }//CIERRA TRY CATCH
}
```

- Actualizar incidencia.

Usa un 'Scanner' para leer los campos que utiliza los objetos de la colección 'incidencia'.

Luego, se crea una operación de actualización utilizando el método 'Updates.combine' que nos permite ejecutar varios "Update" a la vez y establecer los valores de los

campos leídos. Se establecen las opciones de actualización para permitir la inserción (upsert) si no existe el documento con la "id_incidencia" especificada.

Finalmente, se llama a 'updateOne' en la colección para aplicar la operación de actualización y se maneja cualquier excepción de MongoDB que pueda ocurrir. Si la actualización funciona, se imprime un mensaje y el número de documentos modificados.

```
public static void updateIncidencia() {
    MongoDB database = crearConexion().getDatabase("watermelon");
    MongoCollection<Document> collection =
database.getCollection("incidencias");
    Scanner lector = new Scanner(System.in);
    System.out.println("ID de la incidencia a actualizar : ");
    int id_incidencia=Integer.parseInt(lector.nextLine());
    System.out.println("ID del Cliente a actualizar : ");
    int idCliente_incidencia=Integer.parseInt(lector.nextLine());
    System.out.println("Dime la fecha de la incidencia : "+"\\n"+"Formato :
dd/mm/aaaa");
    String fecha_incidencia=lector.nextLine();
    System.out.println("Dime el motivo de la incidencia : ");
    String motivo_incidencia=lector.nextLine();
    System.out.println("Tipo de la incidencia : ");
    String tipo_incidencia=lector.nextLine();
    System.out.println("¿Necesito reparacion fisica el producto?");
    String reparacionFisica_incidencia=lector.nextLine();
    System.out.println("¿La llamada soluciono su problema?");
    String solucion_incidencia=lector.nextLine();
    System.out.println("Operador que le atendio : ");
    String operador_incidencia=lector.nextLine();

    Bson updates = Updates.combine(
        Updates.set("idCliente_incidencia", idCliente_incidencia),
        Updates.set("fecha_incidencia", fecha_incidencia),
        Updates.set("motivo_incidencia", motivo_incidencia),
        Updates.set("tipo_incidencia", tipo_incidencia),
        Updates.set("reparacion Fisica incidencia", reparación Física
incidencia),
        Updates.set("solución Llamada", solución incidencia),
        Updates.set("operador incidencia", operador incidencia)
    );

    UpdateOptions options = new UpdateOptions().upsert(true);

    try {
        UpdateResult result = collection.updateOne(eq("id_incidencia",
id_incidencia), updates, options);
        System.out.println("Incidencia actualizada correctamente");
    }
```

```

        System.out.println(result.getModifiedCount());
    } catch (MongoException e) {
        System.out.println("Error al actualizar la incidencia");
    }
}

```

- Borrar incidencia

Este código crea un método público llamado "borrarIncidencia". Este método se conecta a la base de datos "watermelon" y recupera la colección "incidencias". Luego, utiliza un objeto "Scanner" para leer el "ID de la incidencia a eliminar" del usuario a través de la consola y lo almacena en una variable "id_incidencia".

Más tarde, se crea un objeto de "Document" con el valor "id_incidencia" y se usa el método "deleteOne" de la colección para eliminar la incidencia con ese "ID". Por último, si los datos introducidos son correctos se borra el objeto de la colección 'incidencia', si no, no se borra.

```

public static void borrar Incidencia(){
    //borrar un usuario por id
    MongoClient mongoClient = crearConexion();
    MongoDBDatabase database = mongoClient.getDatabase("watermelon");
    MongoCollection<Document> collection =
    database.getCollection("incidencias");
    Scanner lector = new Scanner(System.in);
    System.out.println("ID de la incidencia a eliminar : ");
    int id_incidencia=Integer.parseInt(lector.nextLine());

    Document query = new Document();
    query.append("id_incidencia", id_incidencia);
    collection.deleteOne(query);
    System.out.println("Incidencia borrado correctamente");
    mongoClient.close();
}

```


Estos son los métodos CRUD que hemos utilizado. Para el usuario 'Direccion' hemos utilizado métodos con la misma estructura ya que tiene que acceder a la misma colección (incidencia) que 'Técnico'. Para la colección de 'Clientes' utilizamos la misma estructura que en la colección 'Incidencia', tan solo tendremos que cambiar la conexión a MongoDB (cambiar el nombre de la colección que establecemos en la clase 'mongoCollection').

- Login de usuarios :

Para el login de los usuarios que acceden a la base de datos, hemos utilizado unas credenciales guardadas en un archivo de texto. Mediante una 'Scanner' leemos los datos que el usuario introduce por teclado. Con una estructura de control comprobamos si los datos introducidos por teclado corresponden al usuario que puede iniciar sesión, si son correctos inicia sesión, si no son correctos le da la opción de volver a intentarlo o salir del programa.

```
public String LoginUsuario() throws FileNotFoundException {
    //leer fichero
    Scanner sc = new Scanner(System.in);
    System.out.println("Escribe el nombre : ");
    String correo = sc.nextLine();
    System.out.println("Escribe la contraseña : ");
    String password = sc.nextLine();
    File fr = new File("usuarios.txt");
    Scanner scFichero = new Scanner(fr);
    String[] linea;
    String nombreUsuario = "";
    String passUsuario = "";
    String resultado = null;
    while (scFichero.hasNext()) {
        linea = scFichero.nextLine().split(",");
        nombreUsuario = linea[0];
        passUsuario = linea[1];
        resultado = nombreUsuario + "," + passUsuario;
        try {
            if (correo.equals(nombreUsuario) && password.equals(passUsuario)) {
                System.out.println("Sesion iniciada");
                break;
            }
        }
    }
}
```

```
    } catch (Exception e) {  
        System.out.println("Usuario no encontrado");  
    }  
}  
return resultado;  
}
```

Rúbrica

Estructura de la base de datos.	No se ha aportado SCRIPT SQL de creación de la base de datos.	Se ha aportado SCRIPT SQL para crear una BD de una sola tabla.	Se ha aportado SCRIPT SQL para crear una BD de al menos 2 tablas asociadas correctamente.
	0 puntos	0.5 puntos	1 punto
Pantalla de acceso y de presentación de datos estadísticos	No se ha implementado.	Se ha implementado la primera pero sin validación. O se ha implementado únicamente una de las dos	Se han implementado completamente las dos pantallas.
	0 puntos	0.5 puntos	1 puntos
Modulo de gestión de usuarios	Se han realizado solamente las operaciones de CRUD ó solamente la pantalla.	Pantalla simple y lógica de CRUD.	Lógica de CRUD completa y pantalla con listado de usuarios y botones de modificación y eliminación
	0 puntos	1.5 puntos	3 punto
Módulo de gestión de tareas.	Se han realizado solamente las operaciones de CRUD ó solamente la pantalla.	Pantalla simple y lógica de CRUD.	Lógica de CRUD completa y pantalla con listado de tareas del usuario y botones de modificación y eliminación
	0 puntos	1.5 puntos	3 puntos
Memoria del proyecto y presentación.	No se ha redactado la memoria del proyecto.	Se ha redactado una memoria del proyecto, pero no se han cubierto todos los apartados o contiene falta de ortografía y/o gramática. No se ha añadido a la memoria el estudio sobre sistemas de bases de datos solicitado.	Se ha redactado una memoria del proyecto completa, clara, precisa y libre de faltas de ortografía y/o gramática.
	0 puntos	1 punto	2 puntos