Instalación Mongo Ubuntu: [https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-20-04](https://www.digitalocean.com/community/tutorials/how-to-install-mongodb-on-ubuntu-20-04" \t "https://mail.yahoo.com/d/folders/1/messages/_blank)

Descarga e instalación de Compass: [https://www.mongodb.com/docs/compass/current/install/](https://www.mongodb.com/docs/compass/current/install/" \t "https://mail.yahoo.com/d/folders/1/messages/_blank)

Descargar el mongo-java-driver 2.12.3

Ejercicio CRUD:

[https://www.youtube.com/watch?v=1fX7kk0DOdE](https://www.youtube.com/watch?v=1fX7kk0DOdE" \t "https://mail.yahoo.com/d/folders/1/messages/_blank)

[https://www.youtube.com/watch?v=uekx2\_sdjvg](https://www.youtube.com/watch?v=uekx2_sdjvg" \t "https://mail.yahoo.com/d/folders/1/messages/_blank)

[https://www.youtube.com/watch?v=nxqylXCPUQg](https://www.youtube.com/watch?v=nxqylXCPUQg" \t "https://mail.yahoo.com/d/folders/1/messages/_blank)

[https://www.youtube.com/watch?v=vkbPf2s8Q74&t=201s](https://www.youtube.com/watch?v=vkbPf2s8Q74&t=201s" \t "https://mail.yahoo.com/d/folders/1/messages/_blank)

Código:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

package crudmongodb;  
  
import com.mongodb.BasicDBObject;  
import com.mongodb.DB;  
import com.mongodb.DBCollection;  
import com.mongodb.DBCursor;  
import com.mongodb.MongoClient;  
import java.net.UnknownHostException;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
  
public class CRUDMongoDB {  
  
  
    public static void main(String[] args) {  
         
        MongoClient mongo = crearConexion();  
         
        if(mongo != null) {  
            // Si la base de datos no existe la crea  
            DB db = mongo.getDB("Pruebas");  
           
        // PARA EL CREATE  
        //    System.out.println("BASE DE DATOS CREADA");  
             
        //    insertarUsuario(db, "usuarios", "Sergio", "Mexico");  
        //    insertarUsuario(db, "usuarios", "Laura", "Colombia");  
        //    insertarUsuario(db, "usuarios", "Franco", "Chile");  
         
        // PARA EL READ  
        //    mostrarColeccion(db, "usuarios");  
         
        //    buscarPorNombre(db, "usuarios", "Sergio");  
         
        // PARA EL UPDATE  
        //    System.out.println("Antes del update");  
        //    mostrarColeccion(db, "usuarios");  
        //    actualizarDocumento(db, "usuarios", "Sergio");  
        //    System.out.println("Despues del update");  
        //    mostrarColeccion(db, "usuarios");  
         
        // PARA EL DELETE  
        //    System.out.println("Antes del delete");  
        //    mostrarColeccion(db, "usuarios");  
        //    borrarDocumento(db, "usuarios", "Sergio");  
        //    System.out.println("Después del delete");  
        //    mostrarColeccion(db, "usuarios");            
        }  
         
    }  
     
     
  public static MongoClient crearConexion() {  
      System.out.println("PRUEBA CONEXION MONGODB");  
      MongoClient mongo = null;  
        try {  
            mongo = new MongoClient("localhost", 27017);  
        } catch (UnknownHostException ex) {  
            Logger.getLogger(CRUDMongoDB.class.getName()).log(Level.SEVERE, null, ex);  
        }  
      return mongo;  
  }    
   
  public static void insertarUsuario(DB db, String coleccion, String nombre, String pais) {  
       
      // Inserta un documento en la colección con el 'nombre' y el 'país' que se le pasan  
      // Si la coleccion no existe la crea  
      DBCollection colec = db.getCollection(coleccion);  
       
      // Creamos un objeto con los parámetros pasados  
      BasicDBObject documento = new BasicDBObject();  
      documento.put("nombre", nombre);  
      documento.put("pais", pais);  
       
      // Insertamos el objeto (documento)  
      colec.insert(documento);  
       
  }  
   
  public static void mostrarColeccion(DB db, String coleccion) {  
       
      // Obtenemos la coleccion de la base de datos que nos pasan  
      DBCollection colec = db.getCollection(coleccion);  
       
      // Consulta la colección y obtiene un objeto 'cursor' equivalente al ResultSet de SQL  
      DBCursor cursor = colec.find();  
       
      // Recorremos el 'cursor' y pintamos los valores de los registros  
      while(cursor.hasNext()) {  
          System.out.println("\* "+ cursor.next().get("nombre") + " - " + cursor.curr().get("pais"));  
      }  
  }  
   
  public static void buscarPorNombre(DB db, String coleccion, String nombre) {  
       
      // Obtenemos la coleccion de la base de datos que nos pasan  
      DBCollection colect = db.getCollection(coleccion);  
       
      // Creamos un objeto y le pasamos el parámetro que nos llega para hacer la consulta  
      BasicDBObject consulta = new BasicDBObject();  
      consulta.put("nombre", nombre);  
       
      // Hacemos la consulta y obtenemos el registro o registros que coincidan  
      DBCursor cursor = colect.find(consulta);  
      //Los pintamos  
      while(cursor.hasNext()) {  
          System.out.println("-- " + cursor.next().get("nombre") + " - " + cursor.curr().get("pais"));  
           
      }  
       
       
  }  
   
  public static void actualizarDocumento(DB db, String coleccion, String nombre) {  
      DBCollection colec = db.getCollection(coleccion);  
       
      // Sentencia con la información a reemplazar  
      BasicDBObject actualizarPais = new BasicDBObject();  
      actualizarPais.append("$set", new BasicDBObject().append("pais", "Perú"));  
       
      // Busca el documento en la colección  
      BasicDBObject busquedaPorNombre = new BasicDBObject();  
      busquedaPorNombre.append("nombre", nombre);  
       
      // Realiza el update  
      colec.updateMulti(busquedaPorNombre, actualizarPais);  
       
     
  }  
   
  public static void borrarDocumento(DB db, String coleccion, String nombre) {  
      DBCollection colec = db.getCollection(coleccion);  
       
      colec.remove(new BasicDBObject().append("nombre", nombre));  
  }  
}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*