

ACTIVITAT
Objectius: <ul style="list-style-type: none">- Obtenir i processar dades a partir de documents XML i guardar-los en una base de dades JSON
Instruccions: <ul style="list-style-type: none">- Responen a l'espai de cada pregunta, si ho feu amb diapositives enganxeu la diapositiva en aquest mateix espai.- Es valorarà la cura en la presentació del document i que segueixi l'estructura indicada.
Criteris d'avaluació: <ul style="list-style-type: none">- Cada pregunta té el mateix pes- Es valorarà la presentació i els comentaris al codi
Entrega: <ul style="list-style-type: none">- Aquest document anomenat memoria.pdf i el codi corresponent

Repositori de referència: <https://github.com/jpala4-ieti/DAM2-MP06-UF03-Base>

Noms i Cognoms: Joel Berzal Álamo

Repositori GitHub amb exercicis resolts:

<https://github.com/joelberzalgithub/AMS2-MP06-PR3.2-BerzalJoel>

Preparació de l'activitat

El punt de partida és la feina realitzada a la pràctica PR3.1

Heu de posar en marxa un servidor mongodb amb docker-compose (o d'alguna altra forma).

Veure aquesta guia

<https://docs.google.com/presentation/d/1Iw6DURrQvVhAn2VCRB8w99qNWxFWhepidzfyeaZ3r1E/edit#slide=id.p>

Exercicis

Exercici 1. Inserció de dades (5 punts)

Crea un programa Java anomenat **PR32CreateMain.java**. Ha d'inserir a la BD de mongodb les 10000 preguntes amb més "ViewCount" extretes del fitxer xml d'stack exchange que vas escollir per la pràctica anterior.

Requisits

- Podeu usar els procediments que vulgueu per obtenir les dades (consulta directa a BaseX, aprofitar el desenvolupament de PR3.1 per generar un XML i processar-lo, etc...) **(1.75 punts)**
- Heu d'inserir a mongodb documents en el següent format **(1.75 punts)**:

```
{
  "question": {
    "Id": "10",
    "PostTypeId": "1",
    "AcceptedAnswerId": "18",
    "CreationDate": "2010-10-19T20:43:29.497",
    "Score": "15",
    "ViewCount": "5541",
    "Body": "<p>In Dominion: Alchemy, you can play Possession to",
    "OwnerUserId": "32",
    "LastActivityDate": "2016-11-09T23:48:17.017",
    "Title": "If Possession is played while executing another Po",
    "Tags": "<dominion><dominion-alchemy>",
    "AnswerCount": "3",
    "CommentCount": "4",
    "ContentLicense": "CC BY-SA 2.5"
  }
}
```

- Abans d'inserir, heu de tractar les entitats HTML per convertir-les en els caràcters corresponents. **(1 punt)** Per exemple:

```
"Body": "&lt;p&gt;Are there any dictionaries specific
```

en

```
"Body": "<p>Are there any dictionaries specific to the subject
```

- Al directory **./data/logs** caldrà generar el fitxer de log **PR32CreateMain.java.log**. Usar algun dels loggers que ofereix java per guardar missatges de l'aplicació, a banda de treure'ls per pantalla. **(1 punt)**

Exercici 2. Realització de consultes contra mongodb i informes (5 punts)

Crea un programa Java anomenat **PR32QueryMain.java** que consulti el servidor mongo.

Cal fer dues consultes i retornar per pantalla el nombre de resultats coincidents:

- Obtenir les preguntes amb ViewCount més gran que la mitjana de ViewCounts que tenim a la col·lecció **(1.75 punts)**
- Obtenir les preguntes que contenen en el títol qualsevol d'aquestes tres lletres ["pug", "wig", "yak", "nap", "jig", "mug", "zap", "gag", "oaf", o "elf"] **(1.75 punts)**

A **./data/out/** genera 2 fitxers pdf anomenats **informe1.pdf** i **informe2.pdf** amb els títols de les preguntes obtingudes com a resultats de les consultes. Cal usar Apache PDFbox **(1.50 punts)**

```
<!-- https://mvnrepository.com/artifact/org.apache.pdfbox/pdfbox -->
<dependency>
  <groupId>org.apache.pdfbox</groupId>
  <artifactId>pdfbox</artifactId>
  <version>3.0.2</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.pdfbox/fontbox -->
<dependency>
  <groupId>org.apache.pdfbox</groupId>
  <artifactId>fontbox</artifactId>
  <version>3.0.2</version>
</dependency>
```

Annexos

docker-compose.yml

```
version: '3.1'
services:
  mongo:
    image: mongo:latest
    container_name: mongo-m06-uf3
    ports:
      - "27017:27017"
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: example
    volumes:
      - mongo-data:/data/db

  mongo-express:
    image: mongo-express:latest
    container_name: mongo-express-m06-uf3
    depends_on:
      - mongo
    ports:
      - "8081:8081"
    environment:
      ME_CONFIG_MONGODB_ADMINUSERNAME: root
      ME_CONFIG_MONGODB_ADMINPASSWORD: example
      ME_CONFIG_MONGODB_URL: "mongodb://root:example@mongo-m06-uf3:27017/"
    volumes:
      mongo-data:
```

Exemple ús PDFbox

```
package cat.iesesteveterradas.exemples;

import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.pdmodel.PDPage;
import org.apache.pdfbox.pdmodel.PDPageContentStream;
import org.apache.pdfbox.pdmodel.font.PDType1Font;
import org.apache.pdfbox.pdmodel.font.Standard14Fonts;

import java.io.IOException;

public class PdfOutputExample {

    public static void main(String[] args) {

        try (PDDocument document = new PDDocument()) {
            PDPage page = new PDPage();
            document.addPage(page);
            PDPageContentStream contentStream = null;

            try {
                contentStream = new PDPageContentStream(document, page);
                contentStream.beginText();
                contentStream.setFont(new PDType1Font(Standard14Fonts.FontName.HELVETICA), 12);
                contentStream.setLeading(14.5f);
                contentStream.newLineAtOffset(25, 750);

                for (int i = 1; i <= 1000; i++) {
                    if (i % 45 == 0 && i != 1000) {
                        contentStream.endText();
                        contentStream.close();

                        page = new PDPage();
                        document.addPage(page);
                        contentStream = new PDPageContentStream(document, page);
                        contentStream.beginText();
                        contentStream.setFont(new PDType1Font(Standard14Fonts.FontName.HELVETICA), 12);
                        contentStream.setLeading(14.5f);
                        contentStream.newLineAtOffset(25, 750);
                    }

                    String line = "Línia " + i + ": Exemple de text";
                    contentStream.showText(line);
                    contentStream.newLine();
                }

                contentStream.endText();
            } finally {
                if (contentStream != null) {
                    contentStream.close();
                }
            }

            String outputPath = System.getProperty("user.dir") + "/data/output/testoutput.pdf";
            document.save(outputPath);
            System.out.println("PDF creat a: " + outputPath);
        } catch (IOException e) {
            System.err.println("Error al crear el PDF: " + e.getMessage());
        }
    }
}
```