DAM. UNIT 2. ACCESS TO DATABASES. PART 2. NON ASSESSABLE EXERCISES

DAM. Acceso a Datos (ADA) (a distancia en inglés)

Unit 2. ACCESS TO DATABASES

Part 2. Working with Non-Relational Databases. Non assessable exercises

Abelardo Martínez

Based and modified from Sergio Badal (www.sergiobadal.com)
Year 2024-2025

Aspects to bear in mind

Important

> **If you look for the solutions surfing the Internet or asking the oracle of ChatGPT you will be fooling yourself**. Keep in mind that **ChatGPT is not infallible or all-powerful**.
>
> It is a great tool to speed up your work once you have mastered a subject, but using it as a shortcut when acquiring basic skills and knowledge seriously undermines your learning. If you use it to get solutions or advice on your own, check the proposed solutions carefully as well. Try to solve the activities using the resources we have seen and the extended documentation you will find in the "Virtual Classroom".

Tips for programming

We advice to follow the next coding standards:

- One instruction per line.
- Add comments to make your code clearer and more readable.
- Use the Hungarian notation to recognise the type of variables at first sight.
- If necessary, we strongly recommend using buffer-based solutions.
- Remember that there are several ways to implement a solution, so choose the one you like best.

1. Console mode. Managing a non-relational database via JDBC in MongoDB

Activity (non assessable)

Create a program in Java to manage PRODUCTS in a shopping cart by printing and using a specific menu. After each option, the user should see the same menu until option zero is pressed. Feel free to share your doubts at the UNIT forum. **You can duplicate this project from UNIT 1**.

ATTENTION: Use the proper exceptions when accessing to databases.

Menu options:

- **Press 0 to “Exit”**
- **Press 1 to “Ask for products until user enters zero as Product name“**
    - For every product we need the Name (String), Price (Float) and Units (Integer), added to the ArrayList of PRODUCTS.
    - Once zero is entered as a Product name, all PRODUCTS will be saved in a MongoDB database using JDBC. You decide columns and collection name.
    - **ATTENTION**: Price must be float and units must be integer!
    - **ATTENTION**: The collection has to be dropped and created again every time you add the ArrayList.
- **Press 2 to “List all products”**
    - Just read the **MongoDB database using JDBC** and print every product information.
- **Press 3 to “Remove all products”**
    - Just delete the products stored at the **MongoDB database using JDBC**.

Menu example:

```
*****
MENU
*****
================================================
  0. Exit
  1. Add products
  2. List all products
  3. Remove all products
================================================
    Select an option:
```