



SEMESTRE MAYO- AGOSTO

**Name:** Jefferson David Yépez Morán

**NRC:** 14575

**Activity:** HW17 – projectMongoImplementation

Create:

The screenshot displays a terminal window on the left and the MongoDB Compass interface on the right. The terminal shows a menu-driven application for managing employees. The user has selected '1' to create an employee, entered 'Juan' as the name, '20' as the hourly wage, and 'Juan@espe.com' as the email. The application confirms the successful creation. The MongoDB Compass interface shows the 'Organivent.Employee' collection with two documents. The first document is for 'Juan' with ID 1 and a wage of 20. The second document is for 'Felipe' with ID 2 and a wage of 19.

```
Output
Organivent1 (run) x Organivent1 (run) #2 x
1. Create employee
2. Read employee
3. Update employee
4.- Delete employee
5.- Exit
Enter the option:
1
Enter id:
1
Enter name:
Juan
Enter hourly wage:
20

Enter user email:
Juan@espe.com
Employee create successfully.
----- Employee -----
1.- Create employee
2.- Read employee
3.- Update employee
4.- Delete employee
5.- Exit
Enter the option:
1
Enter id:
2
Enter name:
Felipe
Enter hourly wage:
19

Enter user email:
Felipe@espe.com
Employee create successfully.
```

MongoDB Compass - cluster0.6gfecg.mongodb.net/Organivent.Employee

cluster0.6gfecg...

My Queries Performance Databases Search

Organivent

Employee

admin config local sample\_mflix

My Queries

Organivent.Employee

Documents Aggregations Schema Indexes

Filter Type a query: { field: 'value' }

ADD DATA EXPORT DATA

```
{ "_id": ObjectId("65adf2fcec5db578561fc7d4"),
  "ID": 1,
  "Name": "Juan",
  "Hourly Wage": "20",
  "Email": "Juan@espe.com" }
{ "_id": ObjectId("65adf30eec5db578561fc7d5"),
  "ID": 2,
  "Name": "Felipe",
  "Hourly Wage": "19",
  "Email": "Felipe@espe.com" }
```

> MONGOSH

Read:



## SEMESTRE MAYO- AGOSTO

Output

Organivent1 (run) x Organivent1 (run) #2 x

U.- EXIT

Enter the option:  
1  
Enter id:  
2  
Enter name:  
Felipe  
Enter hourly wage:  
19

Enter user email:  
Felipe@espe.com  
Employee create successfully

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:  
2

1	Juan	20	Juan@espe.com
2	Felipe	19	Felipe@espe.com

MongoDB Compass - cluster0.6gfeclg.mongodb.net/Organivent.Employee

Documents

Filter Type a query: { field: 'value' } or Generate query Explain Reset Find

ADD DATA EXPORT DATA

1 - 2 of 2

1

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:  
2

Enter user ID to update:  
1  
Enter updated user name:  
Jefferson  
Enter updated hourly wage:  
27  
Enter updated user email:  
Jefferson@espe.ec.com  
User updated successfully.

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit

User updated successfully.

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:  
4  
Enter user ID to delete:  
2  
User deleted successfully.

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:

### Update:

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:  
3  
Enter user ID to update:  
1  
Enter updated user name:  
Jefferson  
Enter updated hourly wage:  
27  
Enter updated user email:  
Jefferson@espe.ec.com  
User updated successfully.

MongoDB Compass - cluster0.6gfeclg.mongodb.net/Organivent.Employee

Documents

Filter Type a query: { field: 'value' } or Generate query Explain Reset Find

ADD DATA EXPORT DATA

1 - 2 of 2

1

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit

### Delete:

User updated successfully.

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:  
4  
Enter user ID to delete:  
2  
User deleted successfully.

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:

Documents

Filter Type a query: { field: 'value' } or Generate query Explain

ADD DATA EXPORT DATA

1

----- Employee -----  
1.- Create employee  
2.- Read employee  
3.- Update employee  
4.- Delete employee  
5.- Exit  
Enter the option:

### Code:



The image shows a Java IDE with two windows open, both displaying the 'Organvent.java' file. The top window shows the initial setup, including imports and the main method's menu structure. The bottom window shows the implementation of the menu options, including creating, reading, updating, and deleting employees.

```

import java.util.function.Consumer;
import org.apache.commons.lang.StringUtils;
import org.bson.Document;
import java.util.Scanner;

/**
 * Author: Jefferson David Yepes Moran, Bit Coders, DDDO - ESSE
 */

public class Organvent {

    private static final String FILE_PATH = "employees.json";
    private static final Gson gson = new GsonBuilder().setPrettyPrinting().create();
    private static final List<Employee> users = loadUsersFromFile();

    private static final String DATABASE_URI = "mongodb+srv://Jefferson:Hola_135@cluster0-6gfeclq.mongodb.net/";
    private static final String DATABASE_NAME = "Organvent";
    private static final String COLLECTION_NAME = "Employee";

    private static final MongoClient mongoClient = new MongoClient(new MongoClientURI(uri: DATABASE_URI));
    private static final MongoDBDatabase database = mongoClient.getDatabase(databaseName: DATABASE_NAME);
    private static final MongoCollection<Document> usersCollection = database.getCollection(string: COLLECTION_NAME);

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        int option = 0;
        do {
            System.out.println("----- Employee -----");
            System.out.println("1.- Create employee");
            System.out.println("2.- Read employee");
            System.out.println("3.- Update employee");
            System.out.println("4.- Delete employee");
            System.out.println("5.- Exit");
            System.out.println("Enter the options: ");
            option = scanner.nextInt();
            switch (option) {
                case 1:
                    createEmployee();
                    break;
                case 2:
                    readEmployee();
                    break;
                case 3:
                    updateEmployee();
                    break;
                case 4:
                    deleteEmployee();
                    break;
                case 5:
                    System.exit(status: 0);
                    break;
                default:
                    System.out.println("Option invalid");
                    break;
            }
        } while (option != 5);
    }

    private static void createEmployee() {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter id: ");
        int id = scanner.nextInt();
        System.out.println("Enter name: ");
        scanner.nextline();
        String name = scanner.nextline();
        System.out.println("Enter hourly wage: ");
        String hourlywage = scanner.nextline();
        scanner.nextline();
        System.out.println("Enter user email: ");
        String email = scanner.nextline();

        Document newUserDocument = new Document()
            .append(key: "ID", value: id)
            .append(key: "Name", value: name)
            .append(key: "Hourly Wage", value: hourlywage)
            .append(key: "Email", value: email);

        usersCollection.insertOne(doc: newUserDocument);

        System.out.println("Employee create successfully.");
    }
}

```