



Asignación 20

Base de datos antes:

restaurantes > nombre				
_id	nombre	rating	fechainauguracion	categorias
[id] 672aec854c3489...	 Restaurante 1	[j32] 5	[T1] 2023-10-10T00:...	[C] [2 elements]
[id] 672aec854c3489...	 Sushi ltto	[j32] 5	[T1] 2023-05-15T00:...	[C] [3 elements]

1. Inserta 3 documentos (restaurantes) más con al menos 2 categorías cada uno.

```

public static void main(String[] args) {
    // Crear instancia de RestaurantesDAO para insertar los restaurantes
    RestaurantesDAO restaurantesDAO = new RestaurantesDAO();

    // Crear 3 restaurantes
    Restaurante restaurante1 = new Restaurante("Sushilito", 5, new Date(), null);
    Restaurante restaurante2 = new Restaurante("PastaMania", 4, new Date(), null);
    Restaurante restaurante3 = new Restaurante("BurgerKing", 4, new Date(), null);

    // Insertar los restaurantes para obtener sus ObjectId
    ObjectId idRestaurante1 = restaurantesDAO.agregar(restaurante1);
    ObjectId idRestaurante2 = restaurantesDAO.agregar(restaurante2);
    ObjectId idRestaurante3 = restaurantesDAO.agregar(restaurante3);

    // Crear categorías y asociarles el ObjectId del restaurante
    Categoria sushi = new Categoria("Sushi", idRestaurante1);
    Categoria tempura = new Categoria("Tempura", idRestaurante1);
    Categoria pasta = new Categoria("Pasta", idRestaurante2);
    Categoria pizza = new Categoria("Pizza", idRestaurante2);
    Categoria hamburguesa = new Categoria("Hamburguesa", idRestaurante3);
    Categoria frita = new Categoria("Frita", idRestaurante3);

    // Asignar las categorías a cada restaurante
    restaurante1.setCategorias(Arrays.asList(sushi, tempura));
    restaurante2.setCategorias(Arrays.asList(pasta, pizza));
    restaurante3.setCategorias(Arrays.asList(hamburguesa, frita));

    // Actualizar los restaurantes con las categorías asociadas
    restaurantesDAO.actualizar(restaurante1);
    restaurantesDAO.actualizar(restaurante2);
    restaurantesDAO.actualizar(restaurante3);

    System.out.println("Tres restaurantes insertados correctamente con sus categorías.");
}

```

Resultados:

Output

Debugger Console

Run (Ej1)

```
--- exec:3.1.0:exec (default-cli) @ RestaurantesMongo ---
nov 10, 2024 8:11:38 P. M. org.bson.diagnostics.Loggers shouldUseSLF4J
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the 'org.bs
nov 10, 2024 8:11:38 P. M. com.mongodb.internal.diagnostics.logging.Loggers should
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the 'org.m
Tres restaurantes insertados correctamente con sus categorias.

-----
BUILD SUCCESS
-----

Total time: 4.619 s
Finished at: 2024-11-10T20:11:39-07:00
-----
```

restaurantes > nombre				
_id	nombre	rating	fechainauguracion	categorias
[id] 672aec854c3489...	Restaurante 1	[i32] 5	[T1] 2023-10-10T00:...	[C] [2 elements]
[id] 672aec854c3489...	Sushi Itto	[i32] 5	[T1] 2023-05-15T00:...	[C] [3 elements]
[id] 673175eab24025...	Sushilito	[i32] 5	[T1] 2024-11-11T03:...	[C] [2 elements]
[id] 673175eab24025...	PastaMania	[i32] 4	[T1] 2024-11-11T03:...	[C] [2 elements]
[id] 673175eab24025...	BurgerKing	[i32] 4	[T1] 2024-11-11T03:...	[C] [2 elements]

2. Consulta los restaurantes con más de 4 estrellas de rating.

```
public static void main(String[] args) {
    RestaurantesDAO restaurantesDAO = new RestaurantesDAO();

    // Consultar restaurantes con más de 4 estrellas
    int ratingMinimo = 4;
    List<Restaurante> restaurantesConMasDe4Estrellas = restaurantesDAO.consultarRestaurantesPorRatingMayorA(ratingMinimo);

    // Mostrar los resultados
    for (Restaurante restaurante : restaurantesConMasDe4Estrellas) {
        System.out.println("Restaurante: " + restaurante.getNombre() + " | Rating: " + restaurante.getRating());
    }
}
```

Resultados:

Output - Run (Ej2)

--- exec:3.1.0:exec (default-cli) @ RestaurantesMongo ---

nov 10, 2024 8:21:00 P. M. org.bson.diagnostics.Loggers shouldUseSLF4J

ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the 'org.l

nov 10, 2024 8:21:00 P. M. com.mongodb.internal.diagnostics.logging.Loggers shou

ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the 'org

Restaurante: Restaurante 1 | Rating: 5

Restaurante: Sushi Itto | Rating: 5

Restaurante: Sushilito | Rating: 5

BUILD SUCCESS

Total time: 4.072 s

Finished at: 2024-11-10T20:21:01-07:00

restaurantes > nombre

_id	nombre	rating	fechainauguracion	categorias
<div>id</div> 672aec854c3489...	<div>Restaurante 1</div>	<div>5</div>	<div>2023-10-10T00:...</div>	<div>[2 elements]</div>
<div>id</div> 672aec854c3489...	<div>Sushi Itto</div>	<div>5</div>	<div>2023-05-15T00:...</div>	<div>[3 elements]</div>
<div>id</div> 673175eab24025...	<div>Sushilito</div>	<div>5</div>	<div>2024-11-11T03:...</div>	<div>[2 elements]</div>

3. Consulta los restaurantes que incluyan la categoría pizza.

```
public static void main(String[] args) {  
    // Usar la clase RestaurantesDAO para consultar restaurantes por categoría  
    RestaurantesDAO restaurantesDAO = new RestaurantesDAO();  
    List<Restaurante> restaurantesPizza = restaurantesDAO.consultarPorCategoria("Pizza");  
  
    // Imprimir los restaurantes encontrados  
    if (restaurantesPizza.isEmpty()) {  
        System.out.println("No se encontraron restaurantes con la categoría 'Pizza'.");  
    } else {  
        for (Restaurante restaurante : restaurantesPizza) {  
            System.out.println("Restaurante: " + restaurante.getNombre());  
            System.out.println("Rating: " + restaurante.getRating());  
            System.out.println("Fecha de inauguración: " + restaurante.getFechainauguracion());  
            System.out.println("Categorías: ");  
            for (Categoria categoria : restaurante.getCategorias()) {  
                System.out.println(" - " + categoria.getNombre());  
            }  
            System.out.println("-----");  
        }  
    }  
}
```

Resultados:

```
Output - Run (Ej3)
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the '
nov 10, 2024 8:24:03 P. M. com.mongodb.internal.diagnostics.logging.Loggers :
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the
Restaurante: PastaMania
Rating: 4
Fecha de inauguración: Sun Nov 10 20:11:38 MST 2024
Categorías:
- Pasta
- Pizza
-----
BUILD SUCCESS
-----
Total time: 5.086 s
Finished at: 2024-11-10T20:24:04-07:00
-----
```

restaurantes > fechainauguracion				
_id	fechainauguracion	nombre	rating	categorias
[id] 673175eab24025...	[T] 2024-11-11T03:...	[N] PastaMania	[i32] 4	[C] [2 elements]

4. Consulta los restaurantes que incluyan la palabra sushi en su nombre.

```
public static void main(String[] args) {
    RestaurantesDAO restaurantesDAO = new RestaurantesDAO();

    // Consultar restaurantes cuyo nombre contiene "sushi"
    String palabra = "sushi";
    List<Restaurante> restaurantesConSushiEnNombre = restaurantesDAO.consultarRestaurantesPorNombre(palabra);

    // Mostrar los resultados
    for (Restaurante restaurante : restaurantesConSushiEnNombre) {
        System.out.println("Restaurante: " + restaurante.getNombre() + " | Rating: " + restaurante.getRating());
    }
}
```

Resultados:

Output - Run (Ej4)

```

/C:/Users/pauli/OneDrive/Documentos/NetBeansProjects/RestaurantesMongo/sr
/C:/Users/pauli/OneDrive/Documentos/NetBeansProjects/RestaurantesMongo/sr

--- exec:3.1.0:exec (default-cli) @ RestaurantesMongo ---
nov 10, 2024 8:28:51 P. M. org.bson.diagnostics.Loggers shouldUseSLF4J
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for th
nov 10, 2024 8:28:51 P. M. com.mongodb.internal.diagnostics.logging.Logge
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for t
Restaurante: Sushi Itto | Rating: 5
Restaurante: Sushilito | Rating: 5

-----
BUILD SUCCESS
-----

Total time: 4.324 s
Finished at: 2024-11-10T20:28:52-07:00
-----

```

restaurantes > nombre				
_id	nombre	rating	fechainauguracion	categorias
[id] 672aec854c3489...	[icon] Sushi Itto	[icon] 5	[icon] 2023-05-15T00:...	[icon] [3 elements]
[id] 673175eab24025...	[icon] Sushilito	[icon] 5	[icon] 2024-11-11T03:...	[icon] [2 elements]

5. Agrega una categoría extra al restaurantsushilito (agregalo si no lo tienes).

```

public static void main(String[] args) {
    RestaurantesDAO restaurantesDAO = new RestaurantesDAO();
    CategoriasDAO categoriasDAO = new CategoriasDAO();

    // Crear una nueva categoria (por ejemplo, "Comida Asiática")
    Categoria nuevaCategoria = new Categoria("Comida Asiática", new ObjectId());

    // Buscar el restaurante "Sushilito"
    List<Restaurante> restaurantesSushilito = restaurantesDAO.consultarRestaurantesPorNombre("Sushilito");

    if (!restaurantesSushilito.isEmpty()) {
        Restaurante restauranteSushilito = restaurantesSushilito.get(0); // Obtener el primer restaurante de la lista
        // Agregar la categoría al restaurante
        boolean exito = restaurantesDAO.agregarCategoriaARestaurante(restauranteSushilito, nuevaCategoria);

        if (exito) {
            System.out.println("Categoría 'Comida Asiática' agregada correctamente a 'Sushilito'.");
        } else {
            System.out.println("No se pudo agregar la categoría a 'Sushilito'.");
        }
    } else {
        // Si no existe, lo creamos
        Restaurante nuevoRestaurante = new Restaurante("Sushilito", 5, new Date(), new ArrayList<>());
        nuevoRestaurante.getCategorias().add(nuevaCategoria); // Agregar la categoría inicial al crear el restaurante

        // Agregar el nuevo restaurante a la base de datos
        ObjectId idRestaurante = restaurantesDAO.agregar(nuevoRestaurante);

        if (idRestaurante != null) {
            System.out.println("Restaurante 'Sushilito' creado y categoría 'Comida Asiática' agregada.");
        } else {
            System.out.println("Hubo un problema al crear el restaurante 'Sushilito'.");
        }
    }
}

```

Resultados:

```
Output - Run (Ej5)
--- resources:3.3.1:resources (default-resources) @ RestaurantesMongo ---
skip non existing resourceDirectory C:\Users\pauli\OneDrive\Documentos\NetBe
--- compiler:3.11.0:compile (default-compile) @ RestaurantesMongo ---
Changes detected - recompiling the module! :source
Compiling 16 source files with javac [debug target 21] to target\classes
/C:/Users/pauli/OneDrive/Documentos/NetBeansProjects/RestaurantesMongo/src/m
/C:/Users/pauli/OneDrive/Documentos/NetBeansProjects/RestaurantesMongo/src/m
--- exec:3.1.0:exec (default-cli) @ RestaurantesMongo ---
nov 10, 2024 8:41:54 P. M. org.bson.diagnostics.Loggers shouldUseSLF4J
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the '
nov 10, 2024 8:41:54 P. M. com.mongodb.internal.diagnostics.logging.Loggers
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the
Categoria 'Comida Asiática' agregada correctamente a 'Sushilito'.

BUILD SUCCESS

Total time: 4.456 s
Finished at: 2024-11-10T20:41:55-07:00
```

restaurantes > categorias > 2 > idRestaurante	
idRestaurante	nombre
id 67317d0202669e...	Comida Asiática

6. Eliminar un restaurante utilizando su identificador.

Restaurante a eliminar:

restaurantes > _id				
_id	nombre	rating	fechainauguracion	categorias
id 672aec854c3489...	Restaurante 1	5	2023-10-10T00:...	[2 elements]
id 672aec854c3489...	Sushi Itto	5	2023-05-15T00:...	[3 elements]
id 673175eab24025...	Sushilito	5	2024-11-11T03:...	[3 elements]
id 673175eab24025...	PastaMania	4	2024-11-11T03:...	[2 elements]
id 673175eab24025...	BurgerKing	4	2024-11-11T03:...	[2 elements]

```
public static void main(String[] args) {
    RestaurantesDAO restaurantesDAO = new RestaurantesDAO();

    ObjectId idRestaurante = new ObjectId("672aec854c34890ba3072f8c");

    // Eliminar el restaurante por su identificador
    boolean exito = restaurantesDAO.eliminar(idRestaurante);

    if (exito) {
        System.out.println("El restaurante fue eliminado exitosamente.");
    } else {
        System.out.println("Hubo un error al eliminar el restaurante.");
    }
}
```

Resultados:

```
Output - Run (Ej6)
/C:/Users/pauli/OneDrive/Documentos/NetBeansProjects/RestaurantesMongo/src/ma
/C:/Users/pauli/OneDrive/Documentos/NetBeansProjects/RestaurantesMongo/src/ma
--- exec:3.1.0:exec (default-cli) @ RestaurantesMongo ---
nov 10, 2024 8:53:01 P. M. org.bson.diagnostics.Loggers shouldUseSLF4J
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the 'o
nov 10, 2024 8:53:02 P. M. com.mongodb.internal.diagnostics.logging.Loggers s
ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the '
El restaurante fue eliminado exitosamente.

BUILD SUCCESS

Total time: 4.505 s
Finished at: 2024-11-10T20:53:03-07:00
```

restaurantes > nombre				
_id	nombre	rating	fechainauguracion	categorias
[id] 672aec854c3489...	Restaurante 1	[32] 5	[T1] 2023-10-10T00:...	[C1] [2 elements]
[id] 673175eab24025...	Sushilito	[32] 5	[T1] 2024-11-11T03:...	[C1] [3 elements]
[id] 673175eab24025...	PastaMania	[32] 4	[T1] 2024-11-11T03:...	[C1] [2 elements]
[id] 673175eab24025...	BurgerKing	[32] 4	[T1] 2024-11-11T03:...	[C1] [2 elements]

7. Eliminar los restaurantes con 3 estrellas o menos.

```
public static void main(String[] args) {
    RestaurantesDAO restaurantesDAO = new RestaurantesDAO();

    // Consultar los restaurantes con rating menor a 3
    List<Restaurante> restaurantesAEliminar = restaurantesDAO.consultarRestaurantesPorRatingMenorA(3);

    // Verificar si no se encontraron restaurantes con rating menor a 3
    if (restaurantesAEliminar.isEmpty()) {
        System.out.println("No se encontraron restaurantes con rating menor a 3.");
    } else {
        // Eliminar cada restaurante con rating menor a 3
        for (Restaurante restaurante : restaurantesAEliminar) {
            // Llamar al método eliminar usando el ObjectId del restaurante
            boolean exito = restaurantesDAO.eliminar(restaurante.getId());

            if (exito) {
                System.out.println("El restaurante " + restaurante.getNombre() + " fue eliminado exitosamente.");
            } else {
                System.out.println("Hubo un error al eliminar el restaurante " + restaurante.getNombre());
            }
        }
    }
}
```





Resultados:

No hay restaurantes con menos de 3 estrellas:

```
Output - Run (Ej7)
  /C:/Users/pauli/OneDrive/Documents/NetBeansProjects/RestaurantesMongo/src,
  --- exec:3.1.0:exec (default-cli) @ RestaurantesMongo ---
  nov 10, 2024 9:00:22 P. M. org.bson.diagnostics.Loggers shouldUseSLF4J
  ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the
  nov 10, 2024 9:00:22 P. M. com.mongodb.internal.diagnostics.logging.Logger:
  ADVERTENCIA: SLF4J not found on the classpath. Logging is disabled for the
  No se encontraron restaurantes con rating menor a 3.

  -----
  BUILD SUCCESS
  -----

  Total time: 4.957 s
  Finished at: 2024-11-10T21:00:23-07:00
  -----
```

restaurantes > nombre				
_id	nombre	rating	fechainauguracion	categorias
[id] 672aec854c3489...	 Restaurante 1	[3.2] 5	[11] 2023-10-10T00:...	[1] [2 elements]
[id] 673175eab24025...	 Sushilito	[3.2] 5	[11] 2024-11-11T03:...	[1] [3 elements]
[id] 673175eab24025...	 PastaMania	[3.2] 4	[11] 2024-11-11T03:...	[1] [2 elements]
[id] 673175eab24025...	 BurgerKing	[3.2] 4	[11] 2024-11-11T03:...	[1] [2 elements]