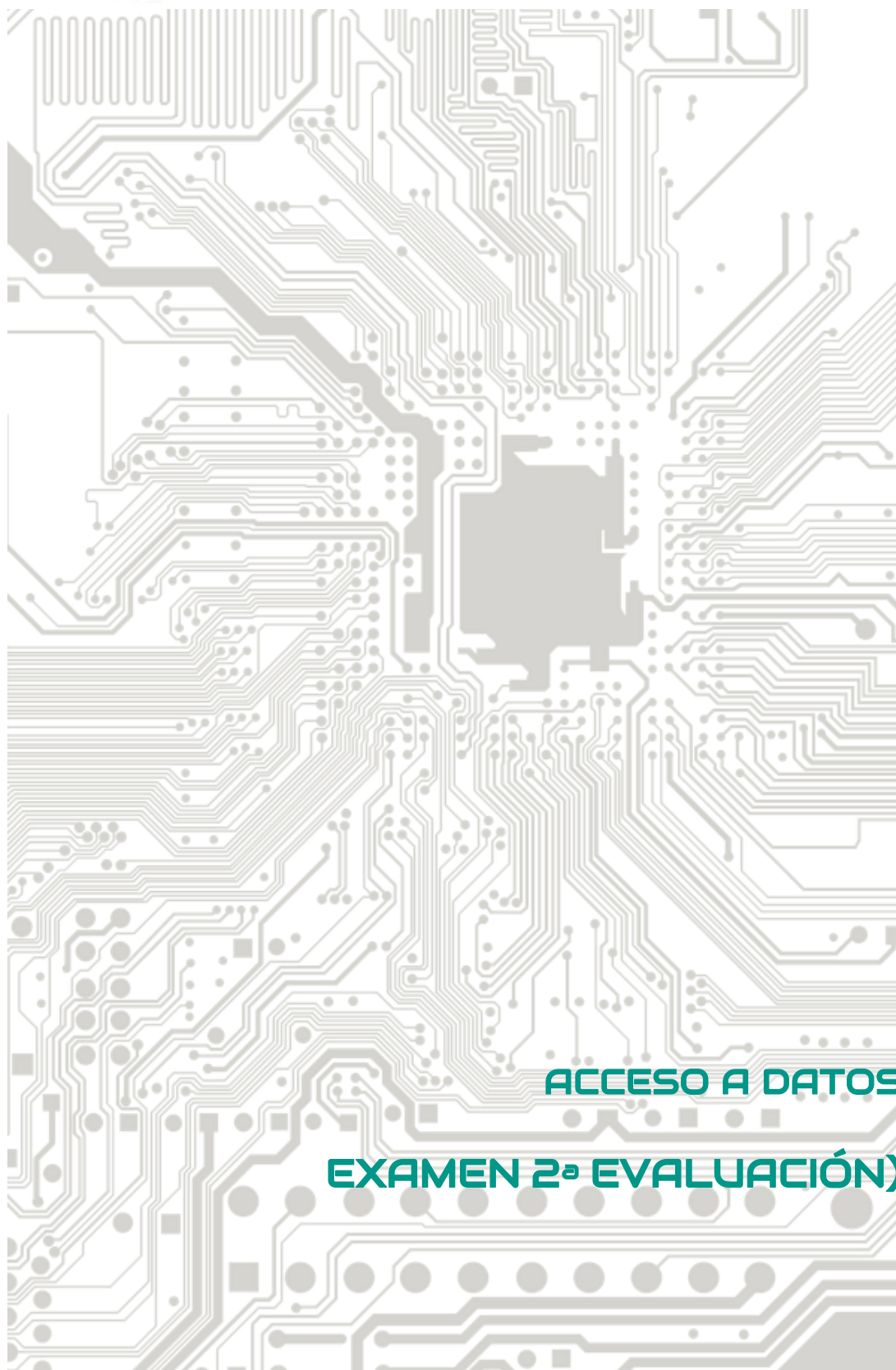




CIPFP Mislata

Centre Integrat Públic
Formació Professional Superior



ALBERTO MARTÍNEZ RUANO – 2º DAM SEMI-PRESENCIAL

CIPFP MISLATA - 2021

ÍNDICE

1.	SQLITE CONNECTION	¡Error! Marcador no definido.
1.1	Diseño.....	¡Error! Marcador no definido.
1.2	BBDD.....	¡Error! Marcador no definido.
1.3	Conexión.....	¡Error! Marcador no definido.
1.4	Inserción	¡Error! Marcador no definido.
1.5	Consulta Escalar.....	¡Error! Marcador no definido.
1.6	Consulta	¡Error! Marcador no definido.
1.7	Consulta desconectada – DataSet.....	¡Error! Marcador no definido.
1.8	Desconexión	¡Error! Marcador no definido.

BASE DE DATOS A UTILIZAR

+ Opciones

			emp_no	apellido	oficio	dir	fecha_alt	salario	comision	1	dept_no
<input type="checkbox"/>				7369	SÁNCHEZ	EMPLEADO	7902	1990-12-17	1040.00	NULL	20
<input type="checkbox"/>				7902	FERNÁNDEZ	ANALISTA	7566	1991-12-03	3000.00	NULL	20
<input type="checkbox"/>				7900	JIMENO	EMPLEADO	7698	1991-12-03	1335.00	NULL	30
<input type="checkbox"/>				7876	ALONSO	EMPLEADO	7788	1991-09-23	1430.00	NULL	20
<input type="checkbox"/>				7839	REY	PRESIDENTE	NULL	1991-11-17	4100.00	NULL	10
<input type="checkbox"/>				7788	GIL	ANALISTA	7566	1991-11-09	3000.00	NULL	20
<input type="checkbox"/>				7782	CEREZO	DIRECTOR	7839	1991-06-09	2885.00	NULL	10
<input type="checkbox"/>				7698	NEGRO	DIRECTOR	7839	1991-05-01	3005.00	NULL	30
<input type="checkbox"/>				7566	JIMÉNEZ	DIRECTOR	7839	1991-04-02	2900.00	NULL	20
<input type="checkbox"/>				7934	MUÑOZ	EMPLEADO	7782	1992-01-23	1690.00	NULL	10
<input type="checkbox"/>				7844	TOVAR	VENDEDOR	7698	1991-09-08	1350.00	0.00	30
<input type="checkbox"/>				7499	ARROYO	VENDEDOR	7698	1990-02-20	1500.00	390.00	30
<input type="checkbox"/>				7521	SALA	VENDEDOR	7698	1991-02-22	1625.00	650.00	30
<input type="checkbox"/>				7654	MARTÍN	VENDEDOR	7698	1991-09-29	1600.00	1020.00	30

+ Opciones

				dept_no	dnombre	loc
<input type="checkbox"/>				10	CONTABILIDAD	SEVILLA
<input type="checkbox"/>				20	INVESTIGACIÓN	MADRID
<input type="checkbox"/>				30	VENTAS	BARCELONA
<input type="checkbox"/>				40	PRODUCCIÓN	BILBAO

1. EJERCICIOS 1 - HIBERNATE

1.1 A - MOSTRAR TODOS LOS DATOS DE EMPLEADOS

```

public static void consultaEmpleados(Session sesion) {
    Query consulta = sesion.createQuery("from Empleados");
    List lista = consulta.list();
    Iterator it = lista.iterator();

    System.out.println("DATOS DE TODOS LOS EMPLEADOS");
    while (it.hasNext()) {
        Empleados empleado = (Empleados) it.next();
        System.out.println("NOMBRE: " + empleado.getApellido() + " - OFICIO: " + empleado.getOficio()
            + " - Nº DIRECTOR: " + empleado.getDir() + " - FECHA ALTA: " + empleado.getFechaAlt()
            + " - SALARIO: " + empleado.getSalario() + "€" + " - COMISION: " + empleado.getComision()
            + "€ - DEPARTAMENTO: " + empleado.getDepartamentos().getDnombre());
    }
    sesion.close();
    System.exit(0);
}

```

```

Problems Javadoc Declaration Console Data Hierarchy Debug
<terminated> LanzaHibernate [Java Application] C:\Program Files\Java\jre1.8.0_241\bin\javaw.exe (9 feb. 2021 16:52:57 - 16:53:03)
INFO: HH000397: Using ASTQueryTranslatorFactory
DATOS DE TODOS LOS EMPLEADOS
NOMBRE: SÁNCHEZ - OFICIO: EMPLEADO - Nº DIRECTOR: 7902 - FECHA ALTA: 1990-12-17 - SALARIO: 1040.0€ - COMISION: null€ - DEPARTAMENTO: INVESTIGACIÓN
NOMBRE: ARROYO - OFICIO: VENDEADOR - Nº DIRECTOR: 7698 - FECHA ALTA: 1990-02-20 - SALARIO: 1500.0€ - COMISION: 390.0€ - DEPARTAMENTO: VENTAS
NOMBRE: SALA - OFICIO: VENDEADOR - Nº DIRECTOR: 7698 - FECHA ALTA: 1991-02-22 - SALARIO: 1625.0€ - COMISION: 650.0€ - DEPARTAMENTO: VENTAS
NOMBRE: JIMÉNEZ - OFICIO: DIRECTOR - Nº DIRECTOR: 7839 - FECHA ALTA: 1991-04-02 - SALARIO: 2900.0€ - COMISION: null€ - DEPARTAMENTO: INVESTIGACIÓN
NOMBRE: MARTÍN - OFICIO: VENDEADOR - Nº DIRECTOR: 7698 - FECHA ALTA: 1991-09-29 - SALARIO: 1600.0€ - COMISION: 1020.0€ - DEPARTAMENTO: VENTAS
NOMBRE: NEGRO - OFICIO: DIRECTOR - Nº DIRECTOR: 7839 - FECHA ALTA: 1991-05-01 - SALARIO: 3005.0€ - COMISION: null€ - DEPARTAMENTO: VENTAS
NOMBRE: CEREZO - OFICIO: DIRECTOR - Nº DIRECTOR: 7839 - FECHA ALTA: 1991-06-09 - SALARIO: 2885.0€ - COMISION: null€ - DEPARTAMENTO: CONTABILIDAD
NOMBRE: GIL - OFICIO: ANALISTA - Nº DIRECTOR: 7566 - FECHA ALTA: 1991-11-09 - SALARIO: 3000.0€ - COMISION: null€ - DEPARTAMENTO: INVESTIGACIÓN
NOMBRE: REY - OFICIO: PRESIDENTE - Nº DIRECTOR: null - FECHA ALTA: 1991-11-17 - SALARIO: 4100.0€ - COMISION: null€ - DEPARTAMENTO: CONTABILIDAD
NOMBRE: TOVAR - OFICIO: VENDEADOR - Nº DIRECTOR: 7698 - FECHA ALTA: 1991-09-08 - SALARIO: 1350.0€ - COMISION: 0.0€ - DEPARTAMENTO: VENTAS
NOMBRE: ALONSO - OFICIO: EMPLEADO - Nº DIRECTOR: 7788 - FECHA ALTA: 1991-09-23 - SALARIO: 1430.0€ - COMISION: null€ - DEPARTAMENTO: INVESTIGACIÓN
NOMBRE: JIMENO - OFICIO: EMPLEADO - Nº DIRECTOR: 7698 - FECHA ALTA: 1991-12-03 - SALARIO: 1335.0€ - COMISION: null€ - DEPARTAMENTO: VENTAS
NOMBRE: FERNÁNDEZ - OFICIO: ANALISTA - Nº DIRECTOR: 7566 - FECHA ALTA: 1991-12-03 - SALARIO: 3000.0€ - COMISION: null€ - DEPARTAMENTO: INVESTIGACIÓN
NOMBRE: MUÑOZ - OFICIO: EMPLEADO - Nº DIRECTOR: 7782 - FECHA ALTA: 1992-01-23 - SALARIO: 1690.0€ - COMISION: null€ - DEPARTAMENTO: CONTABILIDAD

```

1.2 B - EMPLEADOS CON COMISIÓN

```

public static void consultaEmpleadosConComision(Session session) {
    Query consulta = session.createQuery("from Empleados");
    List lista = consulta.list();
    Iterator it = lista.iterator();

    System.out.println("DATOS DE TODOS LOS EMPLEADOS CON COMISION");
    while (it.hasNext()) {
        Empleados empleado = (Empleados) it.next();
        if (empleado.getComision() != null && empleado.getComision() != 0) {
            System.out.println("NOMBRE: " + empleado.getApellido() + " - OFICIO: " + empleado.getOficio()
                + " - Nº DIRECTOR: " + empleado.getDir() + " - FECHA ALTA: " + empleado.getFechaAlt()
                + " - SALARIO: " + empleado.getSalario() + "€" + " - COMISION: " + empleado.getComision()
                + "€ - DEPARTAMENTO: " + empleado.getDepartamentos().getDnombre());
        }
    }
    session.close();
    System.exit(0);
}

```

```

INFO: HH000397: Using ASTQueryTranslatorFactory
DATOS DE TODOS LOS EMPLEADOS CON COMISION
NOMBRE: ARROYO - OFICIO: VENDEADOR - Nº DIRECTOR: 7698 - FECHA ALTA: 1990-02-20 - SALARIO: 1500.0€ - COMISION: 390.0€ - DEPARTAMENTO: VENTAS
NOMBRE: SALA - OFICIO: VENDEADOR - Nº DIRECTOR: 7698 - FECHA ALTA: 1991-02-22 - SALARIO: 1625.0€ - COMISION: 650.0€ - DEPARTAMENTO: VENTAS
NOMBRE: MARTÍN - OFICIO: VENDEADOR - Nº DIRECTOR: 7698 - FECHA ALTA: 1991-09-29 - SALARIO: 1600.0€ - COMISION: 1020.0€ - DEPARTAMENTO: VENTAS

```

1.3 C - ACTUALIZAR DEPARTAMENTO 10

```

public static void cambiarCiudadDpto(Session session) {
    // Creación de la transacción
    Transaction tx = session.beginTransaction();
















    // Consulta para modificar
    String hqlModificacion = "update Departamentos d set d.loc = :nuevaLoc where d.deptNo = :dpto";
    Query consulta1 = session.createQuery(hqlModificacion);
    // Seteado de parametros para la consulta1
    consulta1.setParameter("nuevaLoc", "PARIS");
    consulta1.setParameter("dpto", (byte) 10);
    // Ejecución de la consulta, devuelve filas afectadas
    int filasMod = consulta1.executeUpdate();
    System.out.println("Filas modificadas: " + filasMod);

    // Validación de la transaccion
    tx.commit();

    // Cerrado de recursos
    session.close();
    System.exit(0);
}

```

+ Opciones

  		dept_no	dnombre	loc
<input type="checkbox"/>  Editar  Copiar  Borrar		10	CONTABILIDAD	PARIS
<input type="checkbox"/>  Editar  Copiar  Borrar		20	INVESTIGACIÓN	MADRID
<input type="checkbox"/>  Editar  Copiar  Borrar		30	VENTAS	BARCELONA
<input type="checkbox"/>  Editar  Copiar  Borrar		40	PRODUCCIÓN	BILBAO

1.4 DEPARTAMENTO NUEVO



















TED 09, 2021 5:22:08 PM 01
 WARN: HHH9000004: embed-:
 Nuevo dpto creado...

```
private static void nuevoDpto(Session sesion2, byte noDpto, String nombreDpto, String localizacion) {
    // Nuevo departamento con constructor por defecto
    Departamentos nuevoDpto = new Departamentos();
    // Transaccion
    Transaction tx = sesion.beginTransaction();

    // Seteado de los parametros al Departamento
    nuevoDpto.setDeptNo(noDpto);
    nuevoDpto.setDnombre(nombreDpto);
    nuevoDpto.setLoc(localizacion);

    // Almacenamiento del objeto Departamentos
    sesion.save(nuevoDpto);
    // Validación de la transaccion
    tx.commit();
    System.out.println("Nuevo dpto creado...");
}
```

+ Opciones

  		dept_no	dnombre	loc
<input type="checkbox"/>  Editar  Copiar  Borrar		10	CONTABILIDAD	PARIS
<input type="checkbox"/>  Editar  Copiar  Borrar		20	INVESTIGACIÓN	MADRID
<input type="checkbox"/>  Editar  Copiar  Borrar		30	VENTAS	BARCELONA
<input type="checkbox"/>  Editar  Copiar  Borrar		40	PRODUCCIÓN	BILBAO
<input type="checkbox"/>  Editar  Copiar  Borrar		100	NUEVO DPTO	VALENCIA

2. EJERCICIOS ORACLE

3. EJERCICIOS MONGODB

New Connection localhost:27017 examen

```
use examen
empleado1={_id: 'emp1', nombre: 'Juan', salario: 1000, fechaalta: '10/10/1999'};
empleado2={_id: 'emp2', nombre: 'Alicia', salario: 1400, fechaalta: '07/08/2000'};
empleado3={_id: 'emp3', nombre: 'Maria Jesus', salario: 1500, fechaalta: '05/01/2005'};
empleado4={_id: 'emp4', nombre: 'Alberto', salario: 1100, fechaalta: '15/11/2001'};
db.empleados.save(empleado1);
db.empleados.save(empleado2);
db.empleados.save(empleado3);
db.empleados.save(empleado4);
```

New Connection localhost:27017 examen

```
db.getCollection('empleados').find({})
```

empleados

0.002 sec.

	_id	nombre	salario	fechaalta
1	<div>emp1</div>	<div>Juan</div>	<div>1000.0</div>	<div>10/10/1999</div>
2	<div>emp2</div>	<div>Alicia</div>	<div>1400.0</div>	<div>07/08/2000</div>
3	<div>emp3</div>	<div>Maria Jesus</div>	<div>1500.0</div>	<div>05/01/2005</div>
4	<div>emp4</div>	<div>Alberto</div>	<div>1100.0</div>	<div>15/11/2001</div>

3.1 DATOS DE LOS EMPLEADOS CON SUELDO ENTRE 1050 Y 1450

```
public static void empleSalarioEntre1050Y1450() {
    MongoClient cliente = new MongoClient();
    MongoDBDatabase db = cliente.getDatabase("examen");
    MongoCollection<Document> coleccion = db.getCollection("empleados");

    System.out.println("EMPLEADOS CON SALARIO ENTRE 1050 Y 1450");
    // MongoClient que recibe el find de la collection
    MongoCursor<Document> listaEmpleados = coleccion.find(and(gte("salario", 1050), lte("salario", 1450)))
        .iterator();

    // Se recorre todo el MongoClient de Documents y se va imprimiendo por pantalla
    // Con los objetos BSON, se utilizan los filtros
    while (listaEmpleados.hasNext()) {
        Document emple = listaEmpleados.next();
        System.out.println(emple);
        System.out.println("#####");
        System.out.println("Nº empleado: " + emple.getString("_id"));
        System.out.println("Nombre: " + emple.getString("nombre"));
        System.out.println("Salario: " + emple.getDouble("salario"));
        System.out.println("Fecha alta: " + emple.getString("fechaalta"));
    }
}
```

```
EMPLEADOS CON SALARIO ENTRE 1050 Y 1450
feb 09, 2021 5:49:00 PM com.mongodb.diagnostics.log:
INFORMACIÓN: Cluster description not yet available.
feb 09, 2021 5:49:00 PM com.mongodb.diagnostics.log:
INFORMACIÓN: Opened connection [connectionId{localVa:
feb 09, 2021 5:49:00 PM com.mongodb.diagnostics.log:
INFORMACIÓN: Monitor thread successfully connected to
feb 09, 2021 5:49:00 PM com.mongodb.diagnostics.log:
INFORMACIÓN: Opened connection [connectionId{localVa:
#####
Nº empleado: emp2
Nombre: Alicia
Salario: 1400.0
Fecha alta: 07/08/2000
#####
Nº empleado: emp4
Nombre: Alberto
Salario: 1100.0
Fecha alta: 15/11/2001
```



```

private static void bajarSalario() {
    MongoClient cliente = new MongoClient();
    MongoDBDatabase db = cliente.getDatabase("examen");
    MongoCollection<Document> coleccion = db.getCollection("empleados");

    System.out.println("EMPLEADOS CON SALARIO MAYOR DE 1300");
    // MongoCursor que recibe el find de la collection
    MongoCursor<Document> listaEmpleados = coleccion.find(gte("salario", 1300)).iterator();

    // Se recorre todo el MongoCursor de Documents y se va imprimiendo por pantalla
    // Con los objetos BSON, se utilizan los filtros
    while (listaEmpleados.hasNext()) {
        Document emple = listaEmpleados.next();
        System.out.println(emple);
        System.out.println("#####");
        System.out.println("EMPLEADOS A MODIFICAR");
        System.out.println("Nº empleado: " + emple.getString("_id"));
        System.out.println("Nombre: " + emple.getString("nombre"));
        System.out.println("Salario: " + emple.getDouble("salario"));
        System.out.println("Fecha alta: " + emple.getString("fechaalta"));
    }

    UpdateResult update2 = coleccion.updateMany(gte("salario", 1300), inc("salario", 500));
    // Impresión de cantidad datos actualizados
    System.out.println("ACTUALIZACIÓN DE SALARIO A EMPLEADOS");
    System.out.println("Se han modificado: " + update2.getModifiedCount());
    System.out.println("ACTUALIZACIÓN DE SALARIO A EMPLEADOS");
    System.out.println("Se han modificado: " + update2.getModifiedCount());
}

```

3.2 BAJAR SALARIO

no se cual es la función de decrementar de bson. he hecho un incremento.

```

private static void bajarSalario() {
    MongoClient cliente = new MongoClient();
    MongoDBDatabase db = cliente.getDatabase("examen");
    MongoCollection<Document> coleccion = db.getCollection("empleados");

    System.out.println("EMPLEADOS CON SALARIO MAYOR DE 1300");
    // MongoCursor que recibe el find de la collection
    MongoCursor<Document> listaEmpleados = coleccion.find(gte("salario", 1300)).iterator();

    // Se recorre todo el MongoCursor de Documents y se va imprimiendo por pantalla
    // Con los objetos BSON, se utilizan los filtros
    while (listaEmpleados.hasNext()) {
        Document emple = listaEmpleados.next();
        System.out.println(emple);
        System.out.println("#####");
        System.out.println("EMPLEADOS A MODIFICAR");
        System.out.println("Nº empleado: " + emple.getString("_id"));
        System.out.println("Nombre: " + emple.getString("nombre"));
        System.out.println("Salario: " + emple.getDouble("salario"));
        System.out.println("Fecha alta: " + emple.getString("fechaalta"));
    }

    UpdateResult update2 = coleccion.updateMany(gte("salario", 1300), inc("salario", 500));
    // Impresión de cantidad datos actualizados
    System.out.println("ACTUALIZACIÓN DE SALARIO A EMPLEADOS");
    System.out.println("Se han modificado: " + update2.getModifiedCount());
    System.out.println("ACTUALIZACIÓN DE SALARIO A EMPLEADOS");
    System.out.println("Se han modificado: " + update2.getModifiedCount());
}

```

```
#####
EMPLEADOS CON SALARIO MAYOR DE 1300
feb 09, 2021 6:08:19 PM com.mongodb.diagnostics.logging.J
INFORMACIÓN: Opened connection [connectionId{localValue:1
feb 09, 2021 6:08:19 PM com.mongodb.diagnostics.logging.J
INFORMACIÓN: Monitor thread successfully connected to ser
feb 09, 2021 6:08:19 PM com.mongodb.diagnostics.logging.J
INFORMACIÓN: Opened connection [connectionId{localValue:2
#####
EMPLEADOS A MODIFICAR
Nº empleado: emp2
Nombre: Alicia
Salario: 1400.0
Fecha alta: 07/08/2000
#####
EMPLEADOS A MODIFICAR
Nº empleado: emp3
Nombre: Maria Jesus
Salario: 1500.0
Fecha alta: 05/01/2005
ACTUALIZACIÓN DE SALARIO A EMPLEADOS
Se han modificado: 2
ACTUALIZACIÓN DE SALARIO A EMPLEADOS
Se han modificado: 2
```