

## **Explicación acerca de la segunda Fase de Encuentras de todo**

**Andrés Estiven Gamboa Mena**

**Darwin Mosquera Hinestroza**

**ingeniería de sistemas**

**Martha Jiménez**

**Septiembre 2025**

## CRUD DE LA TABLA DE PROVEEDORES

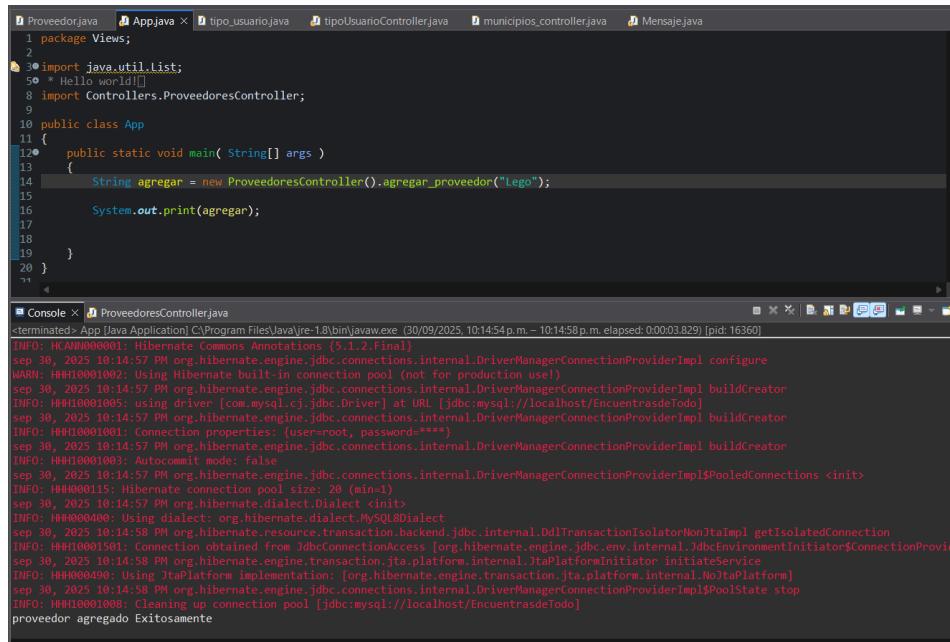
En el apartado siguiente se van a mostrar todos los controladores que por el momento se han agregado a la base de datos que hasta el momento son 3 (Municipios, Proveedores y tipos de usuario), ya mas adelante se hará el ingreso de los demás controladores para hacer la parte funcional del programa. Tambien se debe tener en cuenta que en esta fase solo se esta viendo la funcionalidad de los controladores, la parte visual se tratara mas adelante cuando ya todo este perfectamente integrado en la terminal.

Sin embargo para aclararse algunas dudas del modelo se adjuntaron en este mismo archivo, cabe resaltar que estos son padres en las relaciones, es decir que poseen una relación de 1:n con las tablas que están relacionadas a ellos, así que se le dio prioridad a crear estas tablas para que no hubiese algún tipo de conflicto con las tablas hijos de estas.

### Características de los modelos

En los modelos se puede percibir que cada uno de ellos poseen el @Id, @Column y en la llave primaria poseen un @GeneratedValue(strategy = GenerationType.IDENTITY) que hace referencia a que será un valor autoincrementable.

### Agregar



The screenshot shows an IDE interface with two tabs: 'App.java' and 'Console'. The 'App.java' tab contains the following Java code:

```
1 package Views;
2
3 import java.util.List;
4 * Hello world!*
5 import Controllers.ProveedoresController;
6
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        String agregar = new ProveedoresController().agregar_proveedor("Lego");
12        System.out.print(agregar);
13    }
14 }
```

The 'Console' tab displays the output of the application's execution, showing Hibernate configuration logs and the successful addition of the supplier 'Lego' to the database.

```
INFO: HHH00000001: Hibernate Commons Annotations (5.1.2.Final)
sep 30, 2025 10:14:57 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH00001002: Using Hibernate built-in connection pool (not for production use!)
sep 30, 2025 10:14:57 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH00001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost/EncuentrasdeTodo]
sep 30, 2025 10:14:57 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH00001001: Connection properties: {user=root, password=''}
sep 30, 2025 10:14:57 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH00001003: Autocommit mode: false
sep 30, 2025 10:14:57 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH0000115: Hibernate connection pool size: 20 (min=1)
sep 30, 2025 10:14:57 PM org.hibernate.dialect.Dialect <init>
INFO: HHH0000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
sep 30, 2025 10:14:58 PM org.hibernate.resource.transaction.backend.jdbc.internal.OidTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH00001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProvider]
sep 30, 2025 10:14:58 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH0000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
sep 30, 2025 10:14:58 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH00001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
proveedor agregado Exitosamente
```

### Controlador

```

public String agregar_proveedor(String nom_proveedor) {
    try {
        Proveedor proveedor = new Proveedor(nom_proveedor);
        session.beginTransaction();
        session.persist(proveedor);
        session.getTransaction().commit();
        return "proveedor agregado Exitosamente";
    } catch (Exception e){
        e.printStackTrace();
    } finally {
        sessionFactory.close();
    }
    return "Error al agregar el proveedor";
}

```

## Modificar

### Controlador

```

public String actualizar_proveedor(int id_proveedor, String nom_proveedor) {
    try {
        Proveedor proveedor = session.get(Proveedor.class, id_proveedor);

        session.beginTransaction();
        proveedor.setNom_proveedor(nom_proveedor);

        session.update(proveedor);
        session.getTransaction().commit();
        return "proveedor actualizado Exitosamente";
    } catch (Exception e){
        e.printStackTrace();
        sessionFactory.close();
    } finally {
        sessionFactory.close();
    }

    return "Error al actualizar el proveedor";
}

```

## Output

The screenshot shows an IDE interface with two tabs open: 'App.java' and 'ProveedoresController.java'. The 'App.java' tab contains the following code:

```
1 package Views;
2
3• import java.util.List;
4 * Hello world[]
5
6 import Controllers.ProveedoresController;
7
8
9
10 public class App
11 {
12•     public static void main( String[] args )
13     {
14         String agregar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");
15
16         System.out.print(agregar);
17
18
19     }
20 }
21
```

The 'ProveedoresController.java' tab shows the output of the application's execution in the console:

```
<terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:19:42 p.m. – 10:19:46 p.m. elapsed: 0:00:03.161) [pid: 18136]
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use)
sep 30, 2025 10:19:44 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost/EncuentrasdeTodo]
sep 30, 2025 10:19:44 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {user=root, password=*****}
sep 30, 2025 10:19:44 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: AutoCommit mode: false
sep 30, 2025 10:19:44 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
sep 30, 2025 10:19:44 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
sep 30, 2025 10:19:45 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JDBCConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ContainerIdentifiedConnectionAccess]
sep 30, 2025 10:19:45 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000498: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
sep 30, 2025 10:19:45 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
proveedor actualizado Exitosamente
```

## Eliminar

### Controlador

```

public String EliminarProveedor(int id_proveedor) {
    try {

        Proveedor Eliminar = session.get(Proveedor.class, id_proveedor);

        session.delete(Eliminar);

        session.beginTransaction();

        session.getTransaction().commit();

        return "Proveedor eliminado exitosamente";

    } catch (Exception e) {
        e.printStackTrace();

        return "Hubo un error al eliminar al proveedor";
    } finally {
        sessionFactory.close();
    }
}

```

## Output

```

1 package Views;
2
3 import java.util.List;
4 * Hello world!*
5 import Controllers.ProveedoresController;
6
7 public class App
8 {
9     public static void main( String[] args )
10    {
11        String agregar = new ProveedoresController().EliminarProveedor(3);
12        System.out.print(agregar);
13    }
14 }
15
16
17
18
19
20 }
21

```

Console > ProveedoresController.java

```

<terminated> App [Java Application] C:\Program Files\Java\jre1.8\bin\javaw.exe (30/09/2025, 10:17:44 p.m. - 10:17:47 p.m. elapsed: 0:00:03.622) [pid: 18660]
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
sep 30, 2025 10:17:46 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost/EncuentrasdeTodo]
sep 30, 2025 10:17:46 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {user=root, password=*****}
sep 30, 2025 10:17:46 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
sep 30, 2025 10:17:46 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
sep 30, 2025 10:17:46 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
sep 30, 2025 10:17:47 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProvider]
sep 30, 2025 10:17:47 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
sep 30, 2025 10:17:47 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
Proveedor eliminado exitosamente

```

## Mostrar

## Controlador

## Output

The screenshot shows an IDE interface with two main panes. The top pane displays a Java file named `App.java` containing the following code:

```
Proveedorjava App.java tipo_usuariojava tipoUsuarioController.java municipios_controller.java Mensaje.java
z
3• import java.util.List;
4
5
7• * Hello world![]
10 import Controllers.ProveedoresController;
11 import Models.Proveedor;
12
13
14 public class App
15 {
16•     public static void main( String[] args )
17     {
18         //String agregar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");
19
20         //System.out.print(agregar);
21
22         List<Proveedor> proveedores = ProveedoresController.listar_proveedores();
23
24         System.out.print(proveedores);
25
26
27 }
```

The bottom pane is a terminal window titled "Console" showing the output of running the application:

```
terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:25:19 p.m. - 10:25:25 p.m. elapsed: 0:00:05.657) [pid: 5608]
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost/EncuentrasdeTodo]
sep 30, 2025 10:25:23 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {user=root, password="***"}
sep 30, 2025 10:25:23 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
sep 30, 2025 10:25:23 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
sep 30, 2025 10:25:24 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
sep 30, 2025 10:25:24 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolated
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitia
sep 30, 2025 10:25:25 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
sep 30, 2025 10:25:25 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
[Proveedor [id_proveedor=1, nom_proveedor=Hasbro]]
```

## CRUD DE LA TABLA DE TIPOS DE USUARIOS

## Crear

### Controlador

```
public String agregar_tipo_usuario(String des_t_usuario) {  
  
    try {  
        tipo_usuario t_usuario = new tipo_usuario(des_t_usuario);  
  
        session.beginTransaction();  
  
        session.save(t_usuario);  
  
        session.getTransaction().commit();  
  
        return "Tipo de usuario agregado Exitosamente";  
  
    } catch (Exception e){  
  
        e.printStackTrace();  
        sessionFactory.close();  
  
    } finally {  
        sessionFactory.close();  
    }  
  
    return "Error al agregar el tipo de usuario";  
}
```

## Output

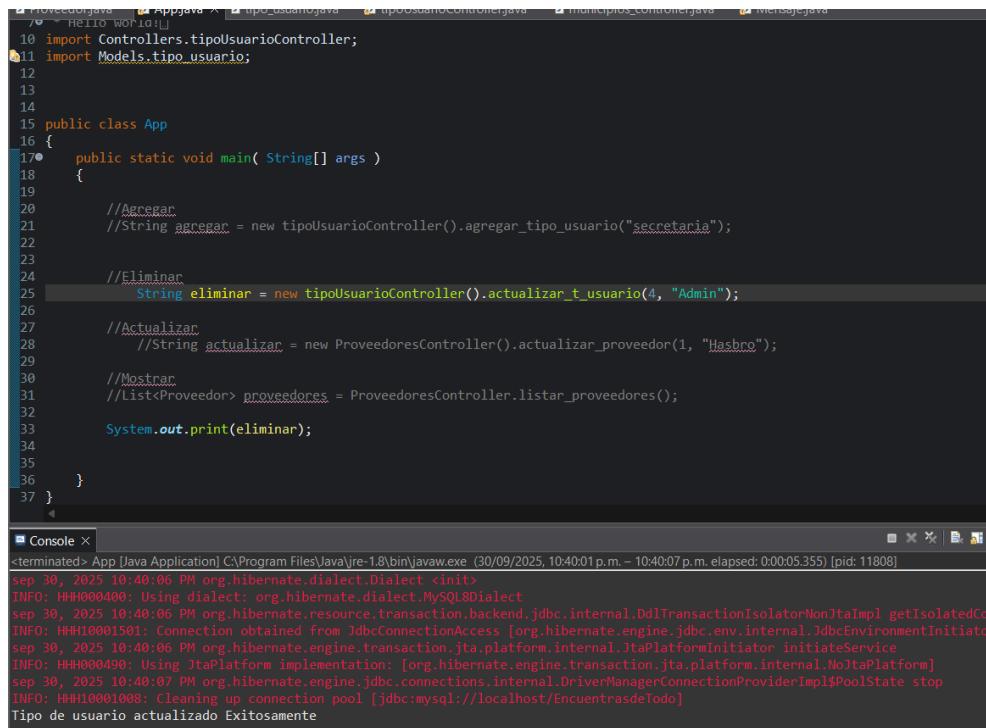
```
70  Hello World!  
10 import Controllers.tipoUsuarioController;  
11 import Models.tipo_usuario;  
12  
13  
14  
15 public class App  
16 {  
17     public static void main( String[] args )  
18     {  
19         //Agregar  
20         String agregar = new tipoUsuarioController().agregar_tipo_usuario("secretaria");  
21  
22         //Eliminar  
23         //String eliminar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");  
24  
25         //Actualizar  
26         //String actualizar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");  
27  
28         //Mostrar  
29         //List<Proveedor> proveedores = ProveedoresController.listar_proveedores();  
30  
31         System.out.print(agregar);  
32  
33     }  
34  
35  
36 }  
37 }  
  
Console X  
terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:36:12 p.m. – 10:36:15 p.m. elapsed: 0:00:03.304) [pid: 14596]  
sep 30, 2025 10:36:14 PM org.hibernate.dialect.Dialect <init>  
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect  
sep 30, 2025 10:36:15 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection  
INFO: HHH10001501: Connection obtained from 3djcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProvider]  
sep 30, 2025 10:36:15 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService  
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]  
sep 30, 2025 10:36:15 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl stop  
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]  
Tipo de usuario agregado Exitosamente
```

## Modificar

## Controlador

```
public String actualizar_t_usuario(int id, String des_tipo_usuario) {  
  
    try {  
  
        tipo_usuario t_usuario = session.get(tipo_usuario.class, id);  
  
        session.beginTransaction();  
        t_usuario.setDes_tipo_usuario(des_tipo_usuario);  
  
        session.update(t_usuario);  
  
        session.getTransaction().commit();  
  
        return "Tipo de usuario actualizado Exitosamente";  
  
    } catch (Exception e){  
  
        e.printStackTrace();  
        sessionFactory.close();  
  
    } finally {  
        sessionFactory.close();  
    }  
}
```

## Output



The screenshot shows an IDE interface with several tabs open. The main tab contains Java code for an 'App' class. The code includes imports for 'Controller' and 'Model', defines a main method, and contains several commented-out code blocks for adding, deleting, updating, and listing data from databases. Below the IDE is a terminal window showing the execution of the application and its output.

```
10 import Controllers.tipoUsuarioController;  
11 import Models.tipo_usuario;  
12  
13  
14  
15 public class App  
16 {  
17     public static void main( String[] args )  
18     {  
19         //Aregar  
20         //String agregar = new tipoUsuarioController().agregar_tipo_usuario("secretaria");  
21  
22         //Eliminar  
23         //String eliminar = new tipoUsuarioController().actualizar_t_usuario(4, "Admin");  
24  
25         //Actualizar  
26         //String actualizar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");  
27  
28         //Mostrar  
29         //List<Proveedor> proveedores = ProveedoresController.listar_proveedores();  
30  
31         System.out.print(eliminar);  
32  
33     }  
34  
35 }  
36 }  
37 }  
  
Console <terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:40:01 p.m. - 10:40:07 p.m. elapsed: 0:00:05.355) [pid: 11808]  
sep 30, 2025 10:40:06 PM org.hibernate.dialect.Dialect <init>  
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect  
sep 30, 2025 10:40:06 PM org.hibernate.resource.transaction.backend.jdbc.internal.JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator] INFO: HHH000150: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator] inititateService [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator] inititateService  
sep 30, 2025 10:40:06 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]  
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]  
sep 30, 2025 10:40:07 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop  
INFO: HHH0001000108: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]  
Tipo de usuario actualizado Exitosamente
```

## Eliminar

## Controlador

```
public String EliminarTipo_usuario(int id_tipo_usuario) {
    try {

        tipo_usuario Eliminar = session.get(tipo_usuario.class, id_tipo_usuario);

        session.delete(Eliminar);

        session.beginTransaction();

        session.getTransaction().commit();

        return "tipo de usuario eliminado exitosamente";

    } catch (Exception e) {
        e.printStackTrace();

        return "Hubo un error al eliminar al tipo de usuario";
    } finally {
        sessionFactory.close();
    }

}
```

## Output

```
/* Hello World! */
10 import Controllers.tipoUsuarioController;
11 import Models.tipo_usuario;
12
13
14
15 public class App
16 {
17     public static void main( String[] args )
18     {
19
20         //Agregar
21         //String agregar = new tipoUsuarioController().agregar_tipo_usuario("secretaria");
22
23
24         //Actualizar
25         String eliminar = new tipoUsuarioController().EliminarTipo_usuario();
26
27         //Eliminar
28         //String actualizar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");
29
30         //Mostrar
31         //List<Proveedor> proveedores = ProveedoresController.listar_proveedores();
32
33         System.out.print(eliminar);
34
35
36     }
37 }
```

Console X

```
<terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:43:31 p.m. – 10:43:36 p.m. elapsed: 0:00:04.482) [pid: 5400]
sep 30, 2025 10:43:34 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQLDialect
sep 30, 2025 10:43:35 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolated
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitia
sep 30, 2025 10:43:35 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
sep 30, 2025 10:43:35 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001088: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
tipo de usuario eliminado exitosamente
```

## Mostrar

## Controlador

```
public static List<tipo_usuario> Mostrar_tipos_usuarios(){

    SessionFactory sessionFactory = new Configuration().configure("hibernate.cfg.xml")
        .addAnnotatedClass(tipo_usuario.class).buildSessionFactory();

    Session session = sessionFactory.openSession();

    List<tipo_usuario> t_usuarios = null;

    try {

        t_usuarios = session.createQuery("FROM tipo_usuario", tipo_usuario.class).getResultList();

        return t_usuarios;

    } catch (Exception e){
        e.printStackTrace();

    }finally {
        sessionFactory.close();
    }
    return t_usuarios;

}
```

## Output

```
/* Hello World! */
10 import Controllers.tipoUsuarioController;
11 import Models.tipo_usuario;
12
13
14
15 public class App
16 {
17     public static void main( String[] args )
18     {
19
20         //Agregar
21         //String agregar = new tipoUsuarioController().agregar_tipo_usuario("secretaria");
22
23
24         //Actualizar
25         //String eliminar = new tipoUsuarioController().EliminarTipo_usuario(5);
26
27         //Eliminar
28         //String actualizar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");
29
30
31         //Mostrar
32         List<tipo_usuario> lista = tipoUsuarioController.Mostrar_tipos_usuarios();
33
34         System.out.print(lista);
35
36     }
37 }
```

Console > App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:45:40 p.m. – 10:45:43 p.m. elapsed: 0:00:03.392) [pid: 11252]
sep 30, 2025 10:45:42 PM org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
sep 30, 2025 10:45:43 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolated
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator\$Platform]
sep 30, 2025 10:45:43 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
sep 30, 2025 10:45:43 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl\$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
[tipo\_usuario [id\_tipo\_usuario=3, des\_tipo\_usuario=empleado], tipo\_usuario [id\_tipo\_usuario=4, des\_tipo\_usuario=Admin]]

## Municipios

## Crear

### Controlador

```
public String agregar_municipio(String nom_municipio) {  
  
    try {  
        Municipio municipio = new Municipio(nom_municipio);  
        session.beginTransaction();  
  
        session.save(municipio);  
        session.getTransaction().commit();  
  
        return "municipio agregado Exitosamente";  
    } catch (Exception e){  
  
        e.printStackTrace();  
        sessionFactory.close();  
  
    } finally {  
        sessionFactory.close();  
    }  
  
    return "Error al agregar el municipio";  
}
```

## Output

```
5  
6 * Hello world!□  
7  
8 import Controllers.municipios_controller;  
9 import Models.Municipio;  
10  
11  
12  
13  
14  
15 public class App  
16 {  
17     public static void main( String[] args )  
18     {  
19         //Agregar  
20         String agregar = new municipios_controller().agregar_municipio("Girardota");  
21  
22         //Actualizar  
23         //String eliminar = new tipoUsuarioController().EliminarTipo_usuario(5);  
24  
25         //Eliminar  
26         //String actualizar = new ProveedoresController().actualizar_proveedor(1, "Hasbro");  
27  
28         //Mostrar  
29         //List<tipo_usuario> lista = tipoUsuarioController.Mostrar_tipos_usuarios();  
30  
31         System.out.print(agregar);  
32  
33     }  
34  
35 }  
36  
Console <terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:51:57 p.m. - 10:52:01 p.m. elapsed: 0:00:03.956) [pid: 2572]  
sep 30, 2025 10:52:00 PM org.hibernate.dialect.Dialect <init>  
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect  
sep 30, 2025 10:52:01 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getLog  
INFO: HHH0001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentImpl@  
sep 30, 2025 10:52:01 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService  
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]  
sep 30, 2025 10:52:01 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledStat  
INFO: HHH0001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]  
municipio agregado Exitosamente
```

## Modificar

## Controlador

```
public String actualizar_municipio(int id, String nombre_nuevo) {
    try {

        Municipio municipio = session.get(Municipio.class, id);

        municipio.setNom_municipio(nombre_nuevo);

        session.beginTransaction();

        session.update(municipio);

        session.getTransaction().commit();

    }

    return "Municipio actualizado exitosamente";

} catch (Exception e){

    e.printStackTrace();
    return "Error al actualizar el municipio";

} finally {
    sessionFactory.close();
}
```

]

## Output

```
3● import java.util.List;
4
5
7● * Hello world![]
10 import Controllers.municipios_controller;
11 import Models.Municipio;
12
13
14
15 public class App
16 {
17●     public static void main( String[] args )
18     {
19
20         //Agregar
21         //String agregar = new municipios_controller().agregar_municipio("Girardota");
22
23
24         //Actualizar
25         //String eliminar = new tipoUsuarioController().EliminarTipo_usuario(5);
26
27         //Eliminar
28         String actualizar = new municipios_controller().actualizar_municipio(3, "Churilo");
29
30         //Mostrar
31         //List<tipo_usuario> lista = tipoUsuarioController.Mostrar_tipos_usuarios();
32
33         System.out.print(actualizar);
34
35
36     }
37 }
38
```

Console X

```
<terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:55:26 p.m. - 10:55:30 p.m. elapsed: 0:00:04)
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.JtaPlatform]
sep 30, 2025 10:55:30 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProvider
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
Municipio actualizado exitosamente
```

## Eliminar

## Controlador

```
public String eliminar_municipio(int id) {
    try {
        Municipio municipio = session.get(Municipio.class, id);
        session.delete(municipio);
        session.beginTransaction();

        session.getTransaction().commit();

        return "Municipio eliminado exitosamente";
    } catch (Exception e){
        e.printStackTrace();
        return "Error al eliminar el municipio";
    } finally {
        sessionFactory.close();
    }
}
```

## Output

The screenshot shows an IDE interface with several tabs at the top: Proveedor.java, App.java (selected), tipo\_usuario.java, tipoUsuarioController.java, municipios\_controller.java, and Mensaje.java. Below the tabs is a code editor with the following Java code:

```
1 import java.util.List;
2
3 * Hello world[]
4
5
6 * @author
7 * Hello world[]
8
9
10 import Controllers.municipios_controller;
11 import Models.Municipio;
12
13
14
15 public class App
16 {
17     public static void main( String[] args )
18     {
19
20         //agregar
21         //String agregar = new municipios_controller().agregar_municipio("Gicardota");
22
23
24         //actualizar
25         String eliminar = new municipios_controller().eliminar_municipio[0];
26
27         //eliminar
28         //String actualizar = new municipios_controller().actualizar_municipio(3, "Churido");
29
30         //mostrar
31         //List<tipo_usuario> lista = tipoUsuarioController.Mostrar_tipos_usuarios();
32
33         System.out.print(eliminar);
34
35
36     }
37 }
```

Below the code editor is a terminal window titled "Console" showing the application's output:

```
<terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:57:35 p.m. – 10:57:38 p.m. elapsed: 0:00:03.053) [pid: 18]
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
sep 30, 2025 10:57:38 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolImpl INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]
Municipio eliminado exitosamente
```

## Mostrar

## Controlador

```
public static List<Municipio> lista_municipios() {  
  
    SessionFactory sessionFactory = new Configuration().configure("hibernate.cfg.xml")  
        .addAnnotatedClass(Municipio.class).buildSessionFactory();  
    List<Municipio> lista_municipios = null;  
    Session session = sessionFactory.openSession();  
  
    try {  
        lista_municipios = session.createQuery("FROM Municipio", Municipio.class).getResultList();  
  
        return lista_municipios;  
    } catch (Exception e){  
        e.printStackTrace();  
        return lista_municipios;  
    } finally {  
        sessionFactory.close();  
    }  
}
```

]

## Output

```
3• import java.util.List;  
4  
5  
7• * Hello world!□  
10 import Controllers.municipios_controller;  
11 import Models.Municipio;  
12  
13  
14  
15 public class App  
16 {  
17•     public static void main( String[] args )  
18     {  
19         //Añadir  
20         //String añadir = new municipios_controller().añadir_municipio("Girardota");  
21  
22  
23         //Actualizar  
24         //String actualizar = new municipios_controller().actualizar_municipio(3);  
25  
26         //Eliminar  
27         //String eliminar = new municipios_controller().eliminar_municipio(3);  
28  
29         //Mostrar  
30         List<Municipio> lista = municipios_controller.lista_municipios();  
31  
32         System.out.print(lista);  
33         |  
34     }  
35  
36 }
```

Console ×  
<terminated> App [Java Application] C:\Program Files\Java\jre-1.8\bin\javaw.exe (30/09/2025, 10:59:20 p.m. – 10:59:23 p.m. elapsed: 0:00:03.035  
sep 30, 2025 10:59:22 PM org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorInfo  
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentImpl@40d1f3c]  
sep 30, 2025 10:59:22 PM org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initia  
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.  
sep 30, 2025 10:59:23 PM org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImp  
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost/EncuentrasdeTodo]  
[Municipio: id\_municipio=1, nombre=Turbo]

## Modelos

## Proveedores

```
4 import javax.persistence.*;
5
6 @Entity
7 @Table(name="tbl_proveedores")
8 public class Proveedor {
9
10     @GeneratedValue(strategy = GenerationType.IDENTITY)
11     @Id
12     @Column(name = "id_proveedor")
13     public int id_proveedor;
14
15
16     @Column(name= "nom_proveedor")
17     public String nom_proveedor;
18
19
20     public int getId_proveedor() {
21         return id_proveedor;
22     }
23
24
25     public void setId_proveedor(int id_proveedor) {
26         this.id_proveedor = id_proveedor;
27     }
28
29
30     public String getNom_proveedor() {
31         return nom_proveedor;
32     }
33
34
35     public void setNom_proveedor(String nom_proveedor) {
36         this.nom_proveedor = nom_proveedor;
37     }
38
39
40
41     @Override
42     public String toString() {
43         return "Proveedores [id_proveedor=" + id_proveedor + ", nom_proveedor=" + nom_proveedor + "]";
44     }
45
46
47     public Proveedor() {
48         //Constructor vacio
49     }
50
51
52
53
54
55
56
57
58
59
59
```

## Tipos de usuario

```
1 package Models;
2
3 import javax.persistence.*;
4
5 @Entity
6 @Table(name="tbl_tipo_usuario")
7 public class tipo_usuario {
8
9     @GeneratedValue(strategy= GenerationType.IDENTITY)
10    @Id
11    @Column(name="id_tipo_usuario")
12    public int id_tipo_usuario;
13
14    @Column(name="des_tipo_usuario")
15    public String des_tipo_usuario;
16
17
18    public int getId_tipo_usuario() {
19        return id_tipo_usuario;
20    }
21
22    public void setId_tipo_usuario(int id_tipo_usuario) {
23        this.id_tipo_usuario = id_tipo_usuario;
24    }
25
26    public String getDes_tipo_usuario() {
27        return des_tipo_usuario;
28    }
29
30    public void setDes_tipo_usuario(String des_tipo_usuario) {
31        this.des_tipo_usuario = des_tipo_usuario;
32    }
33
34
35    public tipo_usuario() {
36
37    }
38
39    public tipo_usuario(String des_tipo_usuario) {
40
41        this.des_tipo_usuario = des_tipo_usuario;
42    }
43
44    @Override
45    public String toString() {
46        return "tipo_usuario [id_tipo_usuario=" + id_tipo_usuario + ", des_tipo_usuario=" + des_tipo_usuario + "]";
47    }

```

## Municipios

```
1 package Models;
2
3 import javax.persistence.*;
4
5 @Entity
6 @Table(name="tbl_municipios")
7 public class Municipio {
8
9     @GeneratedValue(strategy= GenerationType.IDENTITY)
10    @Id
11    @Column(name ="id_municipio")
12    public int id_municipio;
13
14    @Column (name="nom_municipio")
15    public String nom_municipio;
16
17    public int getId_municipio() {
18        return id_municipio;
19    }
20
21    public void setId_municipio(int id_municipio) {
22        this.id_municipio = id_municipio;
23    }
24
25    public String getNom_munisipio() {
26        return nom_municipio;
27    }
28
29    public void setNom_munisipio(String nom_municipio) {
30        this.nom_municipio = nom_municipio;
31    }
32
33    public Municipio() {
34        //Contructor vacio
35    }
36
37
38    public Municipio(String nom_municipio) {
39        this.nom_municipio = nom_municipio;
40    }
41
42    @Override
43    public String toString() {
44        return "Municipio: id_municipio=" + id_municipio + ", nombre=" + nom_municipio;
45    }
46}
```