

# Lenguaje de programación JAVA

## TP Final - Sistema de Forrajería

### Documentación

**Docente:**

Meca, Adrián

**Año:** 2020

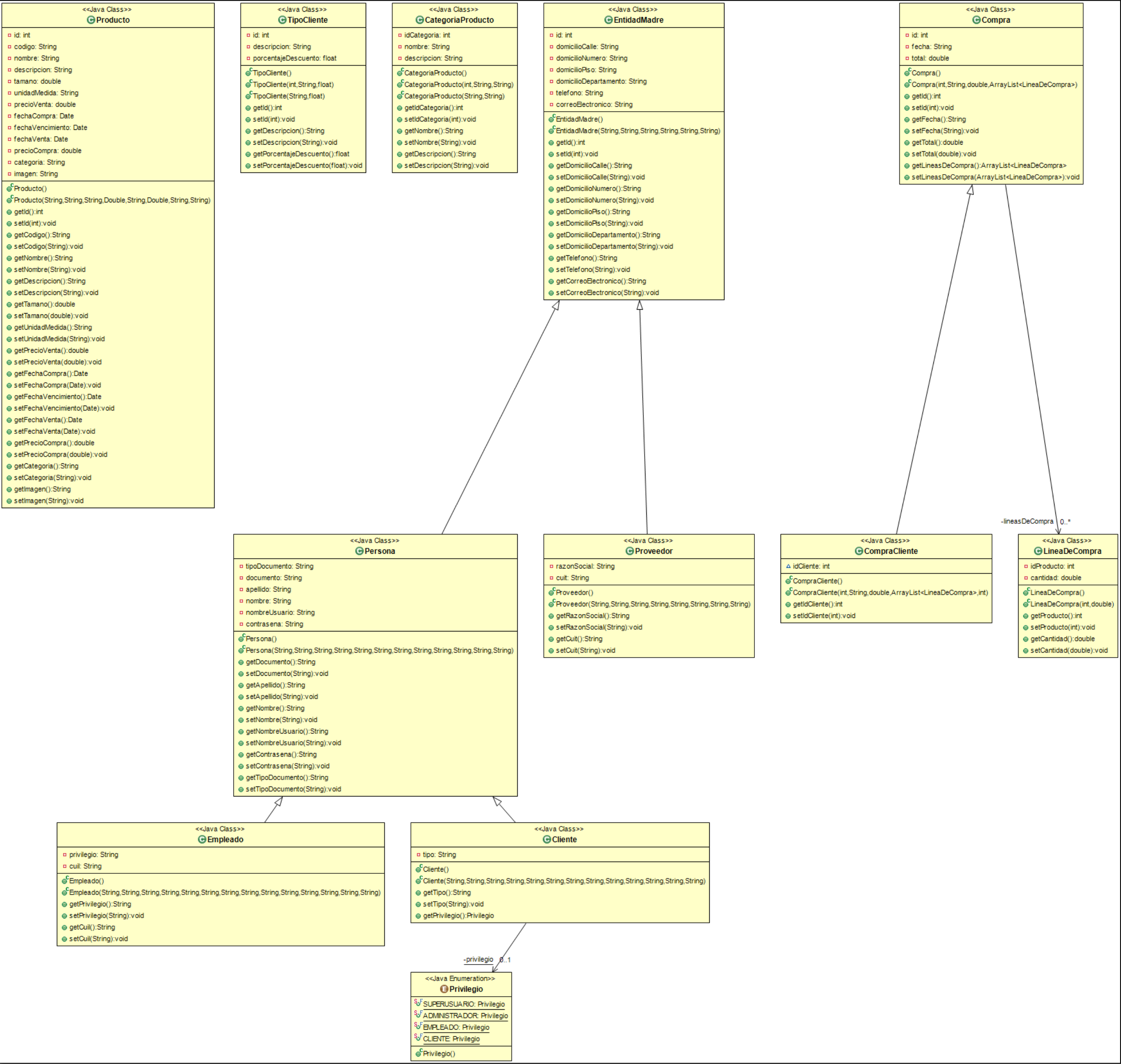
### Integrantes

Legajo	Apellido, Nombre	Correo
34.392	Arnold, Joel	ar.joelarnold@yahoo.com.ar
38.704	Mecánico, César Alberto	cesar.mecanico@gmail.com

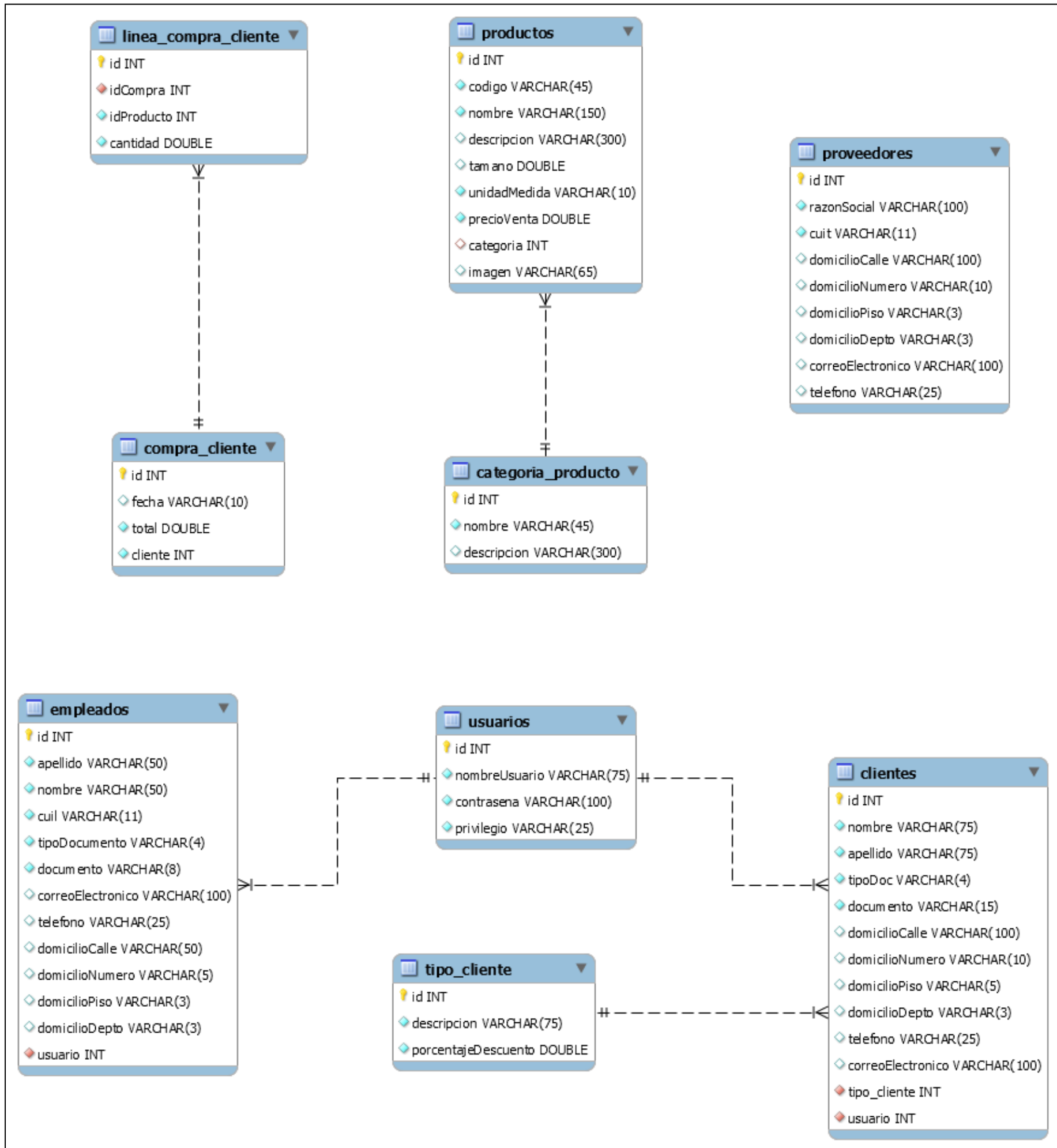
# Indice

Diagrama de Clases .....	3
Modelo de Datos .....	4
Casos de Uso .....	5
Caso de Uso Resumen re-estructurado .....	5
Código y Nombre del CASO DE USO: CUR1 - Comprar. ....	5
Casos de Uso de Usuario sin estructurar .....	6
Código y Nombre del CASO DE USO: CUU1.1 – Mostrar productos y categorías. ....	6
Código y Nombre del CASO DE USO: CUU1.2 – Elegir productos.....	6
Código y Nombre del CASO DE USO: CUU1.3 - Confirmar compra.....	7
Capturas de pantalla .....	8
Pantalla de inicio .....	8
Contacto.....	9
Registro para clientes .....	10
Compra .....	11
ABMC de Tipos de Clientes.....	12
ABMC de Proveedores .....	13
ABMC de Productos.....	14
ABMC de Categorías de Productos .....	15
ABMC de Privilegios.....	16
Perfil de usuario .....	17
Fragmentos de código .....	18

Diagrama de Clases



## Modelo de Datos (Diagrama "Crow Foot")



## Casos de Uso

### Caso de Uso Resumen re-estructurado

**Código y Nombre del CASO DE USO:** CUR1 - Comprar.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Resumen	Re-estructurado	Sistema	Negra	Real	Semántica

**Meta del CASO DE USO:** Realizar una compra.

#### ACTORES

**Primario:** Cliente

**Otros:** <vacío>

**DISPARADOR:** Cliente desea realizar una compra.

#### FLUJO DE SUCESOS

##### CAMINO BÁSICO

1. El Sistema muestra los productos disponibles y sus categorías **invocando al CUU1.1 Mostrar productos y categorías.**
2. El cliente elige un producto e indica la cantidad requerida del mismo **invocando al CUU 1.2 Elegir productos.**  
*El paso 2 se repite por cada producto requerido por el cliente.*
3. El sistema muestra por cada producto elegido: nombre, descripción, y la cantidad requerida. Además, indica el total de la compra.
4. El cliente confirma la compra **invocando al CUU 1.3 Confirmar compra.**
5. El sistema registra la compra, informa el número de compra al cliente y envía un correo electrónico con los datos de la misma.

##### CAMINOS ALTERNATIVOS

- \*.a **Cliente intenta abandonar la compra.**
    - \*.a.1 Sistema pide confirmación para abandonar la página.
      - \*.a.1.a Cliente confirma.
        - \*.a.1.a.1 Sistema cancela registro. Fin CU.
        - \*.a.1.b Cliente cancela abandonar la página.
          - \*.a.1.b.1 Continúa CU en el punto donde se encontraba.
  - \*.b **Cliente elige una categoría.**
    - \*.b.1 Sistema lista productos de la categoría seleccionada.
- 2.a <durante> Cliente elimina un producto.**
- 2.a.1 Sistema elimina producto del listado de productos elegidos.

## POSTCONDICIONES

**Éxito:** Se registró la compra, se informó número de compra y se envió correo con los datos de la misma.

**Fracaso:** No se registró la compra.

**Éxito alternativo:** <vacío>.

## Casos de Uso de Usuario sin estructurar

**Código y Nombre del CASO DE USO:** CUU1.1 – Mostrar productos y categorías.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Semántica

**META DEL CASO DE USO:** Mostrar productos y categorías disponibles.

### ACTORES

**Primario:** Cliente

**Otros:** <vacío>

**PRECONDICIONES:** Cliente logueado.

Existen productos.

Existen categorías.

### FLUJO DE SUCESOS

#### CAMINO BÁSICO

1. Cliente elige la opción comprar.
2. Sistema:
  - 2.1. Valida las categorías y los productos existentes.
  - 2.2. Muestra los productos y categorías disponibles.

#### CAMINOS ALTERNATIVOS

**\*.a Cliente elige una categoría.**

**\*.a.1** Sistema lista productos de la categoría seleccionada.

**Código y Nombre del CASO DE USO:** CUU1.2 – Elegir productos.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Semántica

**META DEL CASO DE USO:** Elegir los productos requeridos.

### ACTORES

**Primario:** Cliente

**Otros:** <vacío>

**PRECONDICIONES:** Cliente logueado.

Existen productos.

Existen categorías.

## FLUJO DE SUCESOS

### CAMINO BÁSICO

1. El cliente elige un producto.
  2. Sistema:
    - 2.1. Valida si el producto se encuentra en el carrito de compra.
      - 2.1.1. Si el producto se encuentra en el carrito, el sistema le suma una unidad en la cantidad.
      - 2.1.2. Si el producto no está en el carrito, agrega el nuevo ítem al mismo.
    - 2.2. Actualiza el valor total de la compra sumando el monto del producto elegido.
- Los pasos 1 y 2 se repiten por cada producto elegido por el cliente.*

### CAMINOS ALTERNATIVOS

- \*.a Cliente elige una categoría.
  - \*.a.1 Sistema lista productos de la categoría seleccionada.
- \*.a Cliente elimina un producto.
  - \*.a.1 Sistema elimina producto del carrito de compra.

**Código y Nombre del CASO DE USO:** CUU1.3 - Confirmar compra.

Nivel	Estructura	Alcance	Caja	Instanciación	Interacción
Usuario	Sin estructurar	Sistema	Negra	Real	Semántica

**META DEL CASO DE USO:** Confirmar la compra.

### ACTORES

**Primario:** Cliente

**Otros:** <vacío>

**PRECONDICIONES:** Cliente logueado.  
Existen productos.  
Existen categorías.  
Hay productos en el carrito de compra.

## FLUJO DE SUCESOS

### CAMINO BÁSICO

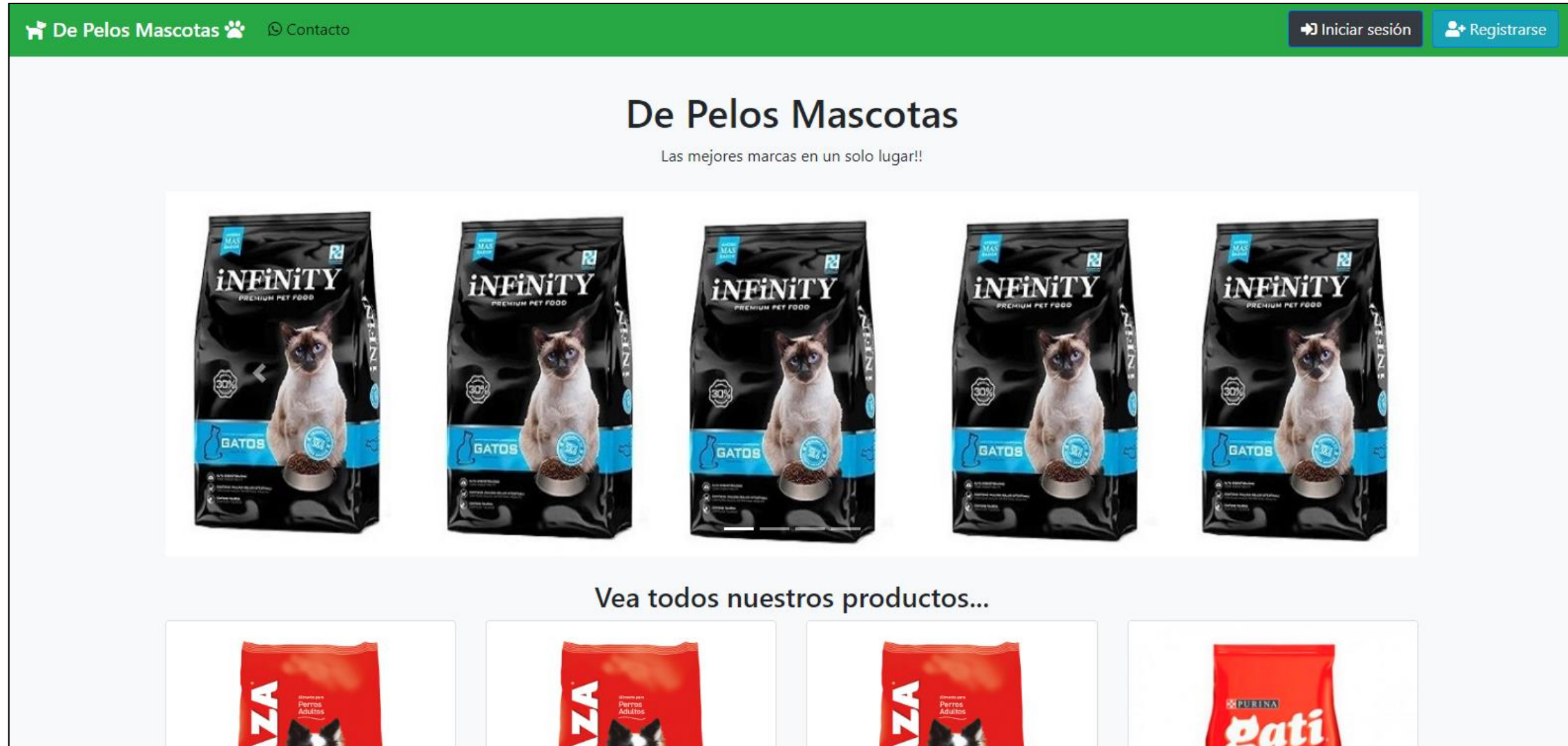
1. Cliente confirma la compra.
2. Sistema:
  - 2.1. Registra fecha de compra, total, número de compra, el cliente y cada uno de los productos seleccionados.
  - 2.2. Confirma compra e informa número de la misma.
  - 2.3. Envía constancia al correo electrónico del cliente con los datos de la compra.

### CAMINOS ALTERNATIVOS

<vacío>

## Capturas de pantalla

### Pantalla de inicio





### Contactanos


Escribinos acá por cualquier consulta o reclamo. También podés escribirnos por Whatsapp, haciendo clic [acá](#).

Tu nombre

Tu correo electrónico

Motivo de la consulta

Tu mensaje

 Enviar consulta



J. J. Paso 2000, Rosario,  
Santa Fe



(0341) 153-366073



depelasmascotas@gmail.com

### ¿Dónde estamos?




### Alta de cliente

#### Usuario y contraseña

#### Datos personales



## Compra

[Comprar](#) [Tipos de cliente](#) [Proveedores](#) [Productos](#) [Categorías de Producto](#) [Privilegios](#) [Contacto](#) [Super Usuario](#)

### Filtrar por categoría


**Todas las categorías**

Perros

Gatos


Aves

Pipetas y accesorios




**Raza**  
Alimento para perros  
\$175

Agregar






**Doggy**  
Alimento para perros  
\$200

Agregar



**Eukanuba**  
Alimento para perros  
\$300



Agregar





### Carrito de compra


Aún no agregó productos al carrito


Total: **\$0.00** [Comprar](#)


 De Pelos Mascotas 


 Comprar


 Tipos de cliente


 Proveedores

 Productos









 Categorías de Producto


 Privilegios

 Contacto






 Super Usuario ▾


Tipos de cliente

Descripción	Porcentaje descuento	Acción
Común	3.0	 
Super	5.0	 
Especial	10.0	 
Super especial	20.0	 

 Agregar

## Proveedores

Razón social	CUIT	Domicilio	Correo electrónico	Teléfono	Acción
Proveedor 1	3055555559	Calle 1	mail@proveedor.com.ar	(0341) 4300155	 
Proveedor 2	3022222224	Callecita 123	correo@electronico.com.ar	(0341) 4305203	 
Proveedor 3	3099999999	Calle 90210	unpro@veedor.com.ar	03 03 456	 










 Agregar

[Comprar](#)
[Tipos de cliente](#)
[Proveedores](#)
[Productos](#)
[Categorías de Producto](#)
[Privilegios](#)
[Contacto](#)
Super Usuario

## Productos









Código	Marca	Descripción	Tamaño	Precio	Categoría	Acción
91	Raza	Alimento para perros	1.0 kg	175.0	Alimento para perros	
93	Doggy	Alimento para perros	10.0 kg	200.0	Alimento para perros	
94	Eukanuba	Alimento para perros	15.0 kg	300.0	Alimento para perros	
90	Gati	Alimento para gatos	0.5 kg	150.0	Alimento para gatos	
99	Avy	Alimento balanceado pajaritos	5.0 gr	355.0	Alimento para aves	
92	Pajarito	Balanceado para pajaritos	2.0 kg	300.0	Alimento para aves	
101	Birdy	Alimento para aves	2.0 kg	555.0	Alimento para aves	
142	Pipetita para gatos	Pipeta para gatos pulgosos	1.0 litro	167.0	Pipetas	
143	Pipeta para perros	Pipeta para perros pulgosos	1.0 litro	120.0	Pipetas	


Agregar

 De Pelos Mascotas   Comprar  Tipos de cliente  Proveedores  Productos  Categorías de Producto  Privilegios  Contacto

Super Usuario ▾

## Categorías de Producto

Nombre	Descripción	Acción
Alimento para perros	Todo lo que se coma que sea exclusivamente para perritos	 
Alimento para gatos	Todo lo que se coma que sea exclusivamente para gatos	 
Alimento para aves	Todo lo comestible para aves	 
Pipetas	Pipetas y accesorios	 



 Agregar








 [Comprar](#) [Tipos de cliente](#) [Proveedores](#) [Productos](#) [Categorías de Producto](#) [Privilegios](#) [Contacto](#) Super Usuario ▾


## Administración de privilegios de usuarios

Empleado	Privilegio	Editar
ADO, Emple	EMPLEADO	
MECA, Adrián	SUPERUSUARIO	
ISTRADOR, Admin	ADMINISTRADOR	
USUARIO, Super	SUPERUSUARIO	



 De Pelos Mascotas 

 Comprar  Tipos de cliente  Proveedores  Productos  Categorías de Producto  Privilegios  Contacto

 Super Usuario ▾

# Mi perfil

---

- Usuario: susuario
- Nombre y apellido: Super Usuario
- Documento: DNI 34567894
- CUIL: 20345678947
- Domicilio: Calle 123 - -
- Teléfono: 3416897845
- Correo electrónico: cesar.mecanico@gmail.com
- Nivel de acceso: SUPERUSUARIO

## Fragmentos de código

### C.U. Elegido: Consulta de Proveedores

#### Código página JSP (proveedoresAdmin.jsp)

```
<!doctype html>
<html lang="en">
<head>
    <jsp:include page="vistas/general/referenciasEncabezado.jsp"></jsp:include>
    <title>Proveedores</title>
    <script type="text/javascript"
src="vistas/administracion/proveedores.Controlador.js"></script>
</head>

<body class="bg-light">
    <% session.setAttribute("huboError", null); %>
    <jsp:include page="vistas/general/barraNavegacion.jsp"></jsp:include>

    <!-- ACÁ ABAJO VA EL CONTENIDO -->
    <% if(session.getAttribute("privilegio") == ("ADMINISTRADOR") ||
        session.getAttribute("privilegio") == ("EMPLEADO") ||
        session.getAttribute("privilegio") == ("SUPERUSUARIO")){ %>
    <jsp:include page="vistas/administracion/proveedores.jsp"></jsp:include>
    <% } else{
        session.setAttribute("huboError", "si");
        session.setAttribute("causaError", null);
        session.setAttribute("mensajeError", "Usted no cuenta con los privilegios de acceso
necesarios para esta sección");
        response.sendRedirect("error.jsp");
    } %>
    <!-- ACÁ ARRIBA VA EL CONTENIDO -->

    <jsp:include page="vistas/general/referenciasPie.jsp"></jsp:include>
</body>

</html>
```

\*\*\*\*\*

#### Código página JSP (proveedores.jsp)

```
<%@page import="entidades.*"%>
<%@page import="java.util.ArrayList"%>
<%@page import="logica.*"%>
<div class="container-fluid" style="text-align:center">
    <br><br>
    <h1>Proveedores</h1>
```



```

        <%=}%>
    </tbody>
</table>
<button type="button" class="btn btn-primary" data-toggle="modal" data-
target="#modalAlta"><i class="fas fa-user-plus"></i> Agregar</button>
</div>

<!-- Modal para el alta -->
<div class="modal fade" id="modalAlta" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
<div class="modal-dialog" role="document">
    <div class="modal-content">
        <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Nuevo proveedor</h5>
            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                <span aria-
                    hidden="true">&times;</span>
            </button>
        </div>
        <div class="modal-body text-left" style="margin-left:10%; margin-right:10%">
        <form role="form" id="altaProveedor" method="post" action="ProveedorAlta">
            <div class="form-group">
                <input type="text" class="form-control" id="razonSocial" name="razonSocial"
                placeholder="Razón social" required="true">
            </div>
            <div class="form-group">
                <input type="number" min="0" class="form-control text-left" id="cuit"
                name="cuit" placeholder="CUIT" required="true">
            </div>
            <div class="form-group">
                <div class="row">
                    <input type="text" class="form-control text-left w-75" id="calle" name="calle"
                    placeholder="Calle" required="true">
                    <input type="text" class="form-control text-left w-25" id="numero"
                    name="numero" placeholder="Nro." required="true">
                </div>
            </div>
            <div class="form-group">
                <div class="row">
                    <input type="number" class="form-control text-left w-50" id="piso"
                    name="piso" placeholder="Piso">
                    <input type="text" class="form-control text-left w-50" id="departamento"
                    name="departamento" placeholder="Depto.">
                </div>
            </div>
        </form>
    </div>
</div>

```

```

    <div class="form-group">
      <input type="tel" class="form-control text-left" id="telefono" name="telefono"
placeholder="Teléfono" required="true">
    </div>
    <div class="form-group">
      <input type="email" class="form-control text-left" id="correoElectronico"
name="correoElectronico" placeholder="Correo electrónico" required="true">
    </div>
    <div class="text-right">
      <br>
      <button type="submit" class="btn btn-primary">Guardar</button>
      <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cerrar</button>
    </div>
  </form>
</div>
</div>
</div>
</div>

```

```

<!-- Modal para la baja -->
<div class="modal fade" id="modalBaja" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Confirmar
eliminación</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-
hidden="true">&times;</span>
      </div>
      <div class="modal-body text-left" style="margin-left:10%; margin-right:10%">
        <form role="form" id="eliminaProveedor" method="post"
action="ProveedorElimina">
          <div class="form-group d-none">
            <input type="number" class="form-control" id="campo_idEliminar"
name="idEliminar">
          </div>
          <div class="form-group">
            <label for="Confirmar">¿Está seguro que desea eliminar este
proveedor?</label>
          </div>
          <div class="text-right">

```

```

        <br>
        <button type="submit" class="btn btn-danger">Confirmar</button>
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cerrar</button>
    </div>
</form>
</div>
</div>
</div>
</div>

<!-- Modal para la modificaci3n -->
<div class="modal fade" id="modalModificacion" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
    <div class="modal-dialog" role="document">
        <div class="modal-content">
            <div class="modal-header">
                <h5 class="modal-title" id="exampleModalLabel">Modificar datos del
proveedor</h5>
                <button type="button" class="close" data-dismiss="modal" aria-label="Close">
                    <span aria-
hidden="true">&times;</span>
            </div>
            <div class="modal-body text-left" style="margin-left:10%; margin-right:10%">
                <form role="form" id="modificaProveedor" method="post"
action="ProveedorModifica">
                    <div class="form-group d-none">
                        <label for="campo_idModificar">ID:</label>
                        <input type="number" class="form-control" id="campo_idModificar"
name="idModificar">
                    </div>
                    <div class="form-group">
                        <input type="text" class="form-control" id="campo_razonSocial"
name="razonSocial" placeholder="Raz3n social" required="true">
                    </div>
                    <div class="form-group">
                        <input type="number" min="0" class="form-control text-left" id="campo_cuit"
name="cuit" placeholder="CUIT" required="true">
                    </div>
                    <div class="form-group">
                        <div class="row">
                            <input type="text" class="form-control text-left w-75" id="campo_calle"
name="calle" placeholder="Calle" required="true">

```

```

        <input type="text" class="form-control text-left w-25" id="campo_numero"
name="numero" placeholder="Nro." required="true">
    </div>
</div>
<div class="form-group">
    <div class="row">
        <input type="number" class="form-control text-left w-50" id="campo_piso"
name="piso" placeholder="Piso">
        <input type="text" class="form-control text-left w-50" id="campo_depto"
name="departamento" placeholder="Depto.">
    </div>
</div>
<div class="form-group">
    <input type="tel" class="form-control text-left" id="campo_telefono"
name="telefono" placeholder="Teléfono" required="true">
</div>
<div class="form-group">
    <input type="email" class="form-control text-left"
id="campo_correoElectronico" name="correoElectronico" placeholder="Correo electrónico"
required="true">
</div>
<div class="text-right">
    <br>
    <button type="submit" class="btn btn-primary">Guardar</button>
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cerrar</button>
</div>
</form>
</div>
</div>
</div>
</div>

```

\*\*\*\*\*

## Controlador Proveedor

```
package logica;
```

```
import java.sql.SQLException;
```

```
import java.util.ArrayList;
```

```
import datos.DatosProveedor;
```

```
import entidades.Proveedor;
```

```
import extras.Excepcion;
```

```

public class ControladorProveedor {

    DatosProveedor datos = new DatosProveedor();

    public ArrayList<Proveedor> buscarTodos() throws SQLException, Excepcion {
        datos = new DatosProveedor();
        return datos.buscarTodos();
    }

    public void alta(Proveedor proveedor) throws SQLException, Excepcion{
        datos.alta(proveedor);
    }

    public void modificacion(Proveedor proveedor) throws SQLException, Excepcion{
        datos.modificacion(proveedor);
    }

    public void baja(int id) throws SQLException, Excepcion{
        datos.baja(id);
    }

}

```

\*\*\*\*\*

### **Datos Proveedor**

```

package datos;

import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

import entidades.Proveedor;
import extras.Excepcion;

public class DatosProveedor {

    public ArrayList<Proveedor> buscarTodos() throws SQLException, Excepcion {
        ResultSet resultado = null;
        PreparedStatement sentenciaSQL = null;
        ArrayList<Proveedor> listado = new ArrayList<Proveedor>();

        try {

```



```

        sentenciaSQL =
Conexion.crearInstancia().abrirConexion().prepareStatement("SELECT * FROM proveedores");
        resultado = sentenciaSQL.executeQuery();

        while (resultado.next()) {
            Proveedor proveedor = new Proveedor();
            proveedor.setId(resultado.getInt("id"));
            proveedor.setRazonSocial(resultado.getString("razonSocial"));
            proveedor.setCuit(resultado.getString("cuit"));

            proveedor.setDomicilioCalle(resultado.getString("domicilioCalle"));

            proveedor.setDomicilioNumero(resultado.getString("domicilioNumero"));

            proveedor.setDomicilioPiso(resultado.getString("domicilioPiso"));

            proveedor.setDomicilioDepartamento(resultado.getString("domicilioDepto"));

            proveedor.setCorreoElectronico(resultado.getString("correoElectronico"));
            proveedor.setTelefono(resultado.getString("telefono"));
            listado.add(proveedor);
        }
    }

    // Estos 2 "Catch" son para el "Try" principal (donde está la consulta a la base
    // de datos)
    catch (SQLException excepcion) {
        throw new SQLException("Algo salió mal intentando buscar los
proveedores en la base de datos", excepcion);
    }

    catch (Excepcion excepcion) {
        throw new Excepcion(excepcion, "Algo salió mal intentando buscar los
proveedores");
    }

    // Este "Try-Catch" es para cerrar la conexión y sus resultados.
    try {
        if (resultado != null)
            resultado.close();
        if (sentenciaSQL != null)
            sentenciaSQL.close();
        Conexion.crearInstancia().cerrarConexion();
    }

```

```

        catch (SQLException excepcion) {
            throw new SQLException("Error intentando cerrar la conexion a la
base de datos", excepcion);
        }
        return listado;
    }

    public void alta(Proveedor proveedor) throws SQLException, Excepcion {
        PreparedStatement sentenciaSQL = null;

        try {
            sentenciaSQL = Conexion.crearInstancia().abrirConexion()
                .prepareStatement("INSERT INTO proveedores
(razonSocial, cuit, domicilioCalle, "
                                + "domicilioNumero, domicilioPiso,
domicilioDepto, telefono, correoElectronico)"
                                + "values (?, ?, ?, ?, ?, ?, ?, ?)");
            sentenciaSQL.setString(1, proveedor.getRazonSocial());
            sentenciaSQL.setString(2, proveedor.getCuit());
            sentenciaSQL.setString(3, proveedor.getDomicilioCalle());
            sentenciaSQL.setString(4, proveedor.getDomicilioNumero());
            sentenciaSQL.setString(5, proveedor.getDomicilioPiso());
            sentenciaSQL.setString(6, proveedor.getDomicilioDepartamento());
            sentenciaSQL.setString(7, proveedor.getTelefono());
            sentenciaSQL.setString(8, proveedor.getCorreoElectronico());
            sentenciaSQL.executeUpdate();
        }

        // Estos 2 "Catch" son para el "Try" principal (donde está la consulta a la base
        // de datos)
        catch (SQLException excepcion) {
            throw new SQLException("Algo salió mal intentando grabar el
proveedor en la base de datos", excepcion);
        }

        catch (Excepcion excepcion) {
            throw new Excepcion(excepcion, "Algo salió mal intentando dar de
alta el proveedor");
        }

        // Este Try-Catch es para cerrar la conexión y sus resultados.
        try {
            if (sentenciaSQL != null)
                sentenciaSQL.close();
            Conexion.crearInstancia().cerrarConexion();
        }
    }

```

```

    }

    catch (SQLException excepcion) {
        throw new SQLException("Error intentando cerrar la conexion a la
base de datos", excepcion);
    }
}

public void baja(int id) throws SQLException, Excepcion {
    PreparedStatement sentenciaSQL = null;

    try {
        sentenciaSQL = Conexion.crearInstancia().abrirConexion()
            .prepareStatement("DELETE FROM proveedores
WHERE id = ?");

        sentenciaSQL.setInt(1, id);
        sentenciaSQL.executeUpdate();
    }

    // Estos 2 "Catch" son para el "Try" principal (donde está la consulta a la base
// de datos)
    catch (SQLException excepcion) {
        throw new SQLException("No se pudo borrar el proveedor de la base
de datos", excepcion);
    }

    catch (Excepcion excepcion) {
        throw new Excepcion(excepcion, "Algo salió mal intentando borrar el
proveedor");
    }

    // Este Try-Catch es para cerrar la conexión y sus resultados.
    try {
        if (sentenciaSQL != null)
            sentenciaSQL.close();
        Conexion.crearInstancia().cerrarConexion();
    }

    catch (SQLException excepcion) {
        throw new SQLException("Error intentando cerrar la conexion a la
base de datos", excepcion);
    }
}

public void modificacion(Proveedor proveedor) throws SQLException, Excepcion {

```

```

PreparedStatement sentenciaSQL = null;

try {
    sentenciaSQL = Conexion.crearInstancia().abrirConexion()
        .prepareStatement("UPDATE proveedores SET
razonSocial = ?, cuit = ?, domicilioCalle = ?, "
                                + "domicilioNumero = ?, domicilioPiso
= ?, domicilioDepto = ?, telefono = ?, "
                                + "correoElectronico = ? WHERE id = ?
");

    sentenciaSQL.setString(1, proveedor.getRazonSocial());
    sentenciaSQL.setString(2, proveedor.getCuit());
    sentenciaSQL.setString(3, proveedor.getDomicilioCalle());
    sentenciaSQL.setString(4, proveedor.getDomicilioNumero());
    sentenciaSQL.setString(5, proveedor.getDomicilioPiso());
    sentenciaSQL.setString(6, proveedor.getDomicilioDepartamento());
    sentenciaSQL.setString(7, proveedor.getTelefono());
    sentenciaSQL.setString(8, proveedor.getCorreoElectronico());
    sentenciaSQL.setInt(9, proveedor.getId());
    sentenciaSQL.executeUpdate();
}

catch (SQLException excepcion) {
    throw new SQLException("Algo salió mal intentando modificar el
proveedor en la base de datos", excepcion);
}

catch (Excepcion excepcion) {
    throw new Excepcion(excepcion, "Algo salió mal intentando modificar
el proveedor");
}

try {
    if (sentenciaSQL != null)
        sentenciaSQL.close();
    Conexion.crearInstancia().cerrarConexion();
}

catch (SQLException excepcion) {
    throw new SQLException("Error intentando cerrar la conexion a la
base de datos", excepcion);
}

}

```

\*\*\*\*\*

## Entidad Proveedor

```
package entidades;
```

```
public class Proveedor extends EntidadMadre {

    private String razonSocial;
    private String cuit;

    public Proveedor() {

    }

    public Proveedor(String domicilioCalle, String domicilioNumero, String domicilioPiso,
String domicilioDepartamento,
                        String telefono, String correoElectronico, String razonSocial, String
cuit) {
        super(domicilioCalle, domicilioNumero, domicilioPiso, domicilioDepartamento,
telefono, correoElectronico);
        this.razonSocial = razonSocial;
        this.cuit = cuit;
    }

    public String getRazonSocial() {
        return razonSocial;
    }

    public void setRazonSocial(String razonSocial) {
        this.razonSocial = razonSocial;
    }

    public String getCuit() {
        return cuit;
    }

    public void setCuit(String cuit) {
        this.cuit = cuit;
    }

}
```