

**DESARROLLO DE UN SITIO WEB PARA UNA FLORERÍA.**

**Autores:**  
Gian Carlos Tapia Fuertes  
Harold Smith Cajahuaman Esteban   
Jason Lyler Changanaqui Morillas

Rojas Segura Brayhan Orlando

**Docente:**  
Richerd Homero Rodas Cueva

**Carrera profesional:**  
Ingeniería de sistemas computacionales

**Curso:**  
Soluciones web y aplicaciones distribuidas

**Ciclo:**VIII

Lima – Perú  
2020

# INDICE

Contenido

[1. INDICE 2](#_Toc45487579)

[2. VISIÓN 3](#_Toc45487580)

[2.1 Presentación y objetivos 3](#_Toc45487581)

[2.2 Contexto 3](#_Toc45487582)

[2.3 Planteamiento del problema 3](#_Toc45487583)

[2.4 Descripción de Stakeholder 4](#_Toc45487584)

[3. REQUISITOS 4](#_Toc45487585)

[3.1 Descripción general 4](#_Toc45487586)

[3.2 Funciones del producto 4](#_Toc45487587)

[3.3 Restricciones 5](#_Toc45487588)

[3.4 Requerimientos de interfaces 5](#_Toc45487589)

[4. ANÁLISIS 10](#_Toc45487590)

[4.1 Diagrama de clases 10](#_Toc45487591)

[4.2 Diagrama de casos de uso 10](#_Toc45487592)

[4.3 Actores 11](#_Toc45487593)

[4.4 Diagrama general de caso de uso 11](#_Toc45487594)

[5. DESCRIPCIÓN 12](#_Toc45487595)

[5.1 Capa de presentación 12](#_Toc45487596)

# VISIÓN

## 2.1 Presentación y objetivos

Este documento describe el trabajado realizado para la carrera de ingeniería de sistemas computacionales. El proyecto consiste en el desarrollo de un sitio web para la florería corporación “Karumi”, ubicada en la avenida Metropolitana 2450 en el distrito de Comas.   
Su misión “Somos una empresa dedicada a la decoración floral, nos especializamos en el diseño y armado de arreglos florales para cualquier evento social y bellos diseños con flores para transmitir todo tipo de sentimientos humanos”.  
El objetivo del sitio web es brindar el servicio de compra al cliente que lo visite, así como proporcionar toda la información sobre los productos que ofrece.  
En cuanto a los usuarios, la página tendrá dos tipos. Los registrados obtendrán descuentos a medida que compran con el tiempo. Los no registrados obtendrán los mismos precios de siempre, sujeto a cambio de la tienda. Ambos tipos de usuarios podrán gozar de todos los productos disponibles en la tienda.

## 2.2 Contexto

El proyecto realizado para la florería “Karumi” ubicada en el distrito de Comas. Fue realizado con previas reuniones para conocer el proceso de compra y su gran lista de productos.

Hoy en día los sitios web de ventas pueden ser utilizados en cualquier negocio que tenga fines por ventas productos, por ejemplo: Farmacias, Tiendas por departamento.

Un sitio web de venta se encarga de tener un stock de sus productos, poder venderlos por medio de una interfaz, solicitar a los proveedores indicados los productos que hagan falta, generar un reporte de venta y administrar los clientes que tengan más compras online.

Debido a que en cualquier sucursal se puede utilizar un sitio web de venta, está interfaz tiene una gran oportunidad de negocio, de igual forma tiene muchos contrincantes. Para poder lograr su éxito haremos un sitio web innovador que no se base en los ya existentes.

## 2.3 Planteamiento del problema

El problema planteado para la creación del sitio web responde a las siguientes funcionalidades:

* Mostrar información general de la florería.
* Permitir a los clientes consultar precios y productos de la florería.
* Permitir a los clientes comprar productos que ofrece la florería.
* Permitir a los clientes generar orden de comprar de productos de la florería.
* Permitir al administrador del sitio web, administrar la base de datos del sistema.

## 2.4 Descripción de Stakeholder

Para proveer de una forma efectiva productos y servicios que se ajusten a las necesidades de los usuarios, es necesario identificar e involucrar a todos los participantes en el proyecto como parte del proceso de modelado de requerimientos. También es necesario identificar a los usuarios del sistema y asegurarse de que el conjunto de participantes en el proyecto los representa adecuadamente. Esta sección muestra un perfil de los participantes y de los usuarios involucrados en el proyecto, así como los problemas más importantes que estos perciben para enfocar la solución propuesta hacia ellos. No describe sus requisitos específicos ya que estos se capturan mediante otro artefacto. En lugar de esto proporciona la justificación de por qué estos requisitos son necesarios.

# REQUISITOS

## 3.1 Descripción general

El sitio web desarrollado pretende dar información general de la tienda así como información sobre sus productos y precios respectivos que ofrece.

## 3.2 Funciones del producto

Cliente anónimo  
- Autenticación: Se muestran 2 cajas de texto vacías y el botón con la opción de login.

Cliente registrado  
- Cerrar sesión: Se muestra la opción para cerrar sesión, cuando se pulsa la página lo redirige a la página principal.

- Pedidos: Se muestra el catálogo de productos que ofrece la tienda con las opciones para agregar al carrito de compras. Están presentes las opciones de búsqueda por categoría y fabricantes.

- Carrito de compras: Se muestra la lista de productos seleccionados por el cliente, con su respectivo precio, cantidad y total. La opción de eliminar está disponible al costado de cada producto en caso se desee quitar del carrito de compras.

- Agregar categoría: Se muestran los campos respectivos para la creación de una nueva categoría en el catálogo de productos.

- Agregar Fabricante: Se muestran los campos respectivos para la creación de un nuevo fabricante en el catálogo de productos.

- Orden de compra: Generará la relación de productos con sus respectivos datos: cantidad, fabricante, precio y total..

-Registrar reporte: Generará un reporte sobre cualquier inquietud o comentario que los cliente deseen hacer sobre la tienda.

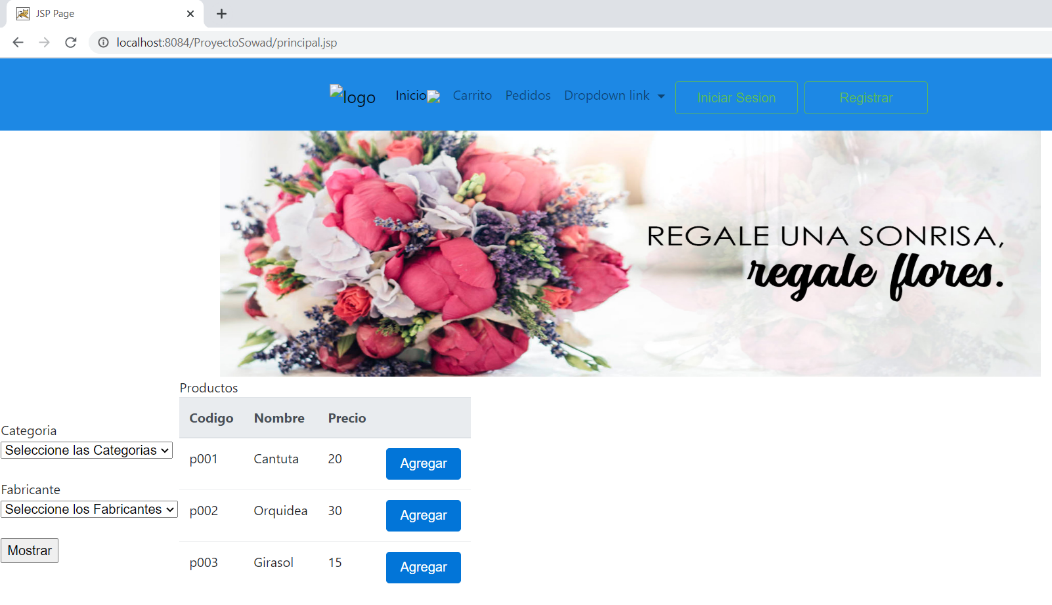
## 3.3 Restricciones

Debido a la naturaleza del sitio web, se requiere de un ordenador con un navegador convencional y una conexión a internet.

## 3.4 Requerimientos de interfaces

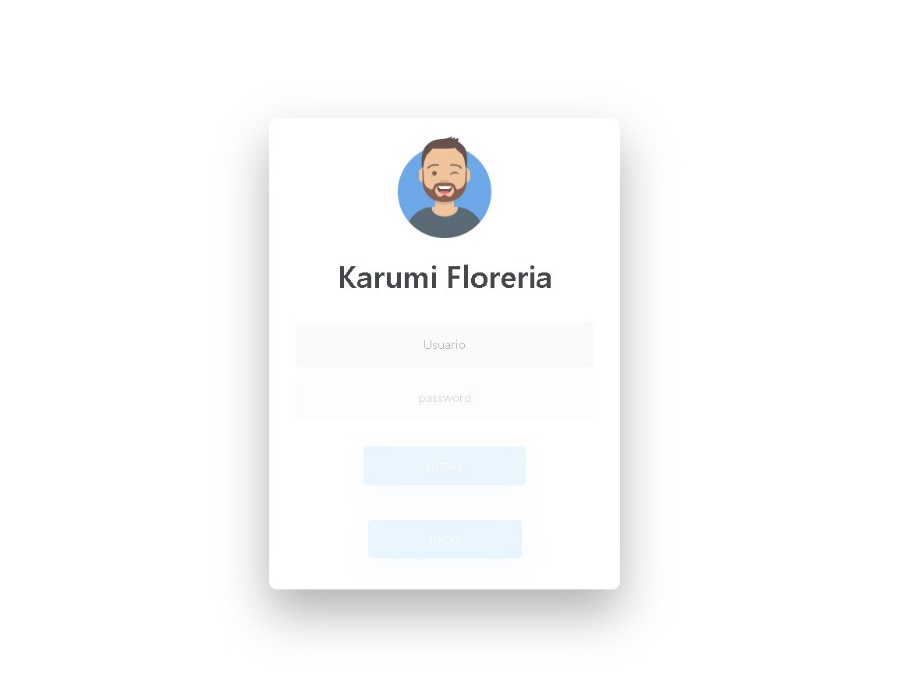
* + 1. **Interface principal**

Captura de la interface principal del sitio web.



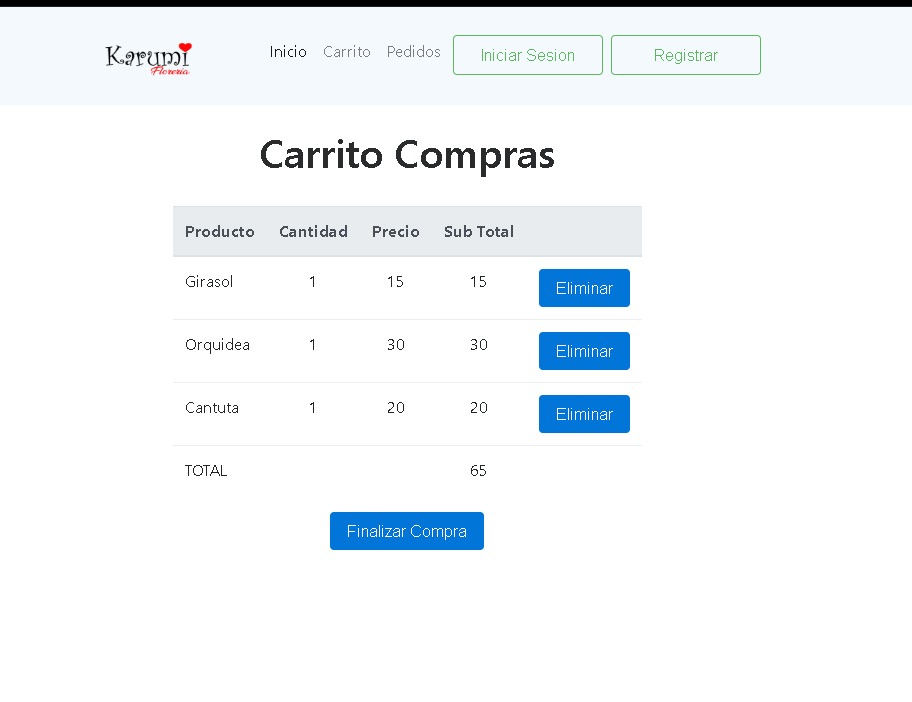
* + 1. **Interface de login**

Captura de la interface de login del sitio web. Con los campos de usuario y contraseña para acceder a los beneficios del cliente registrado.



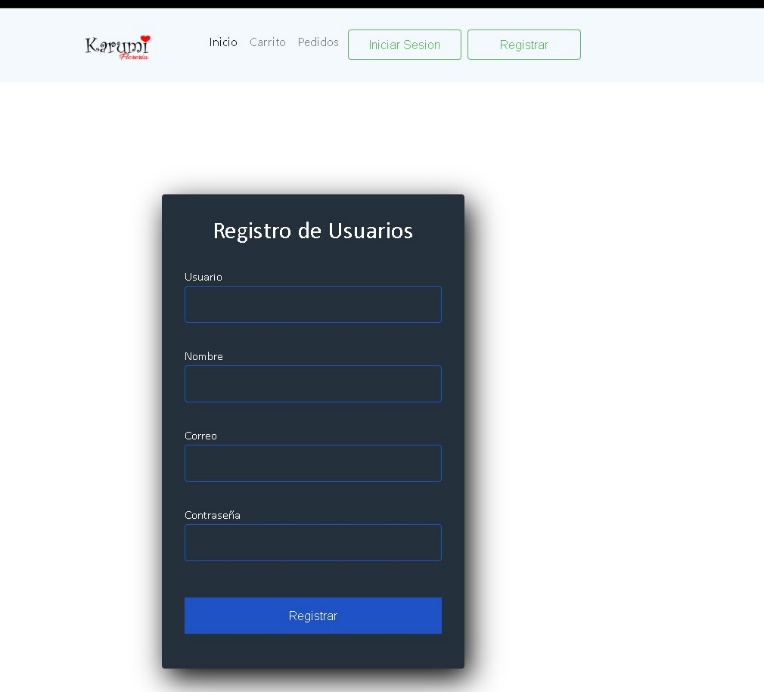
* + 1. **Interface carrito de compras**

Captura de la interface de carrito de compra. Con las opciones de eliminar y finalizar comprar para generar la orden respectiva.



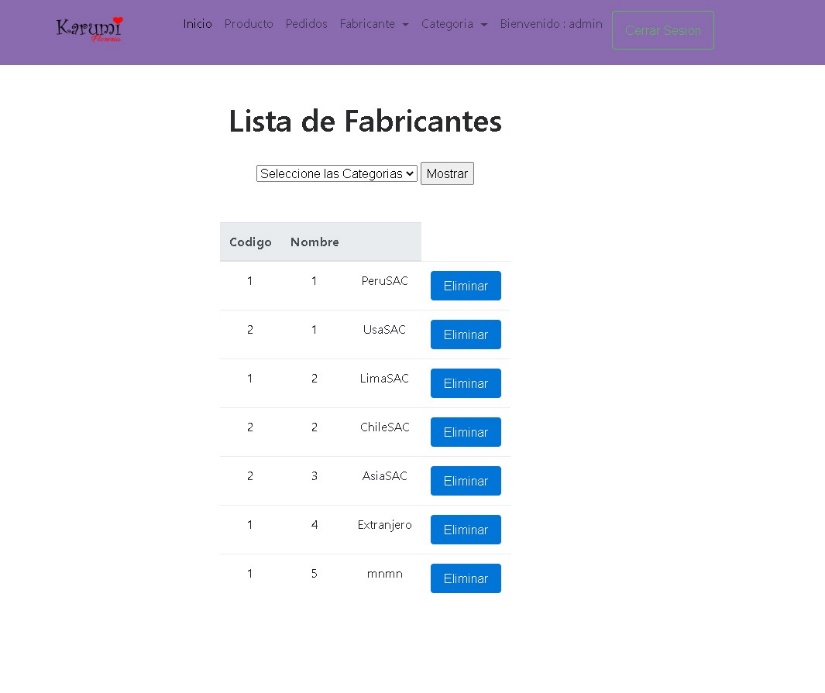
* + 1. **Interface registrar**

Captura de la interface registro de usuario con los campos respectivos para la creación de un usuario nuevo.



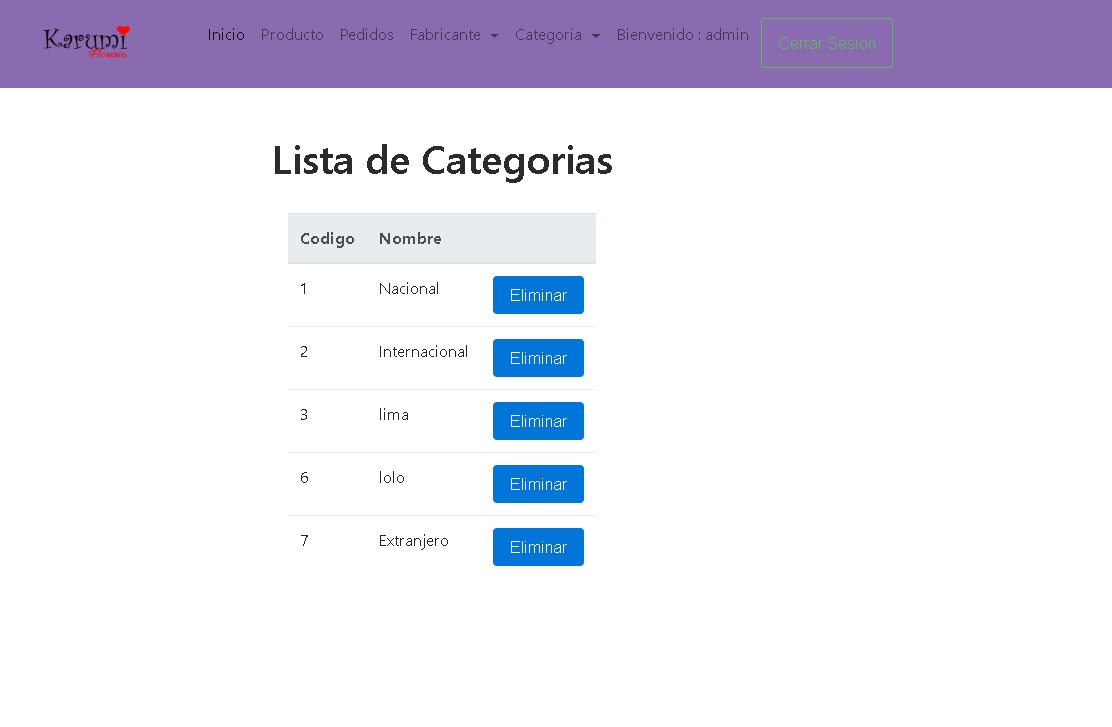
* + 1. **Interface lista de fabricantes**

Captura de la interface lista de fabricantes con la relación respectiva de cada proveedor disponible en la tienda.



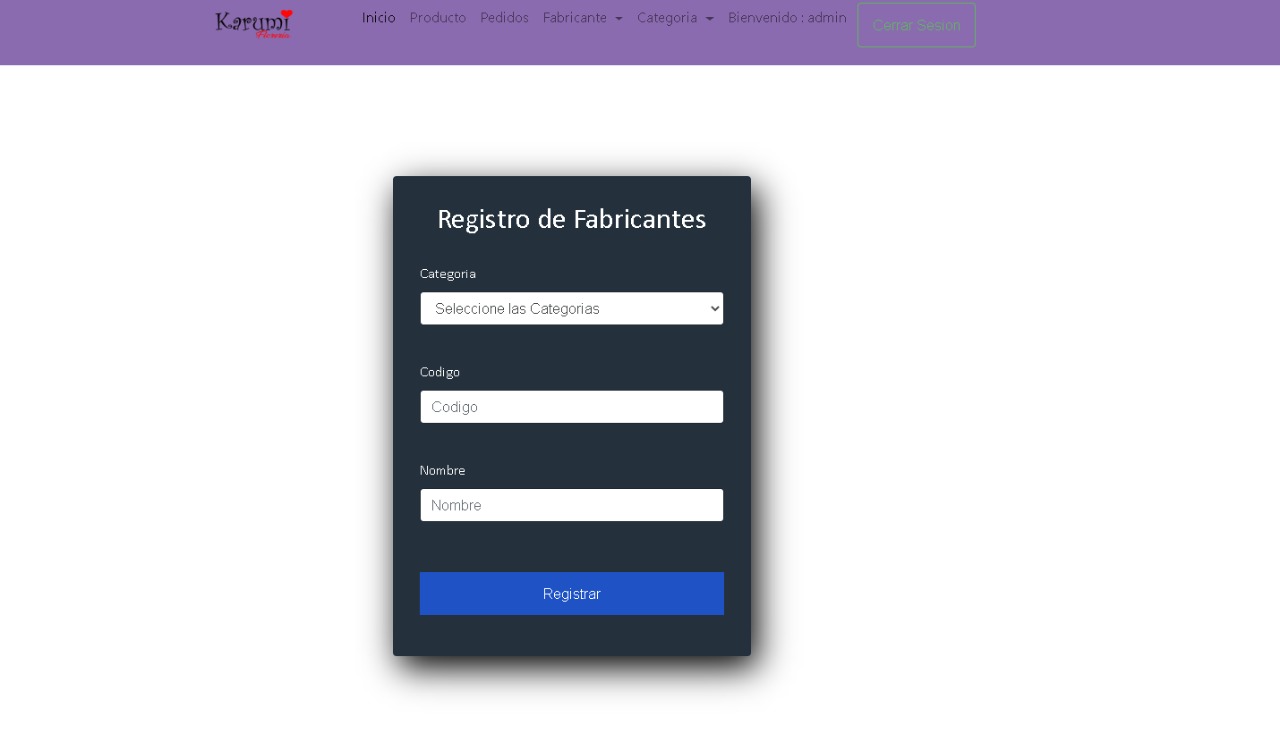
* + 1. **Interface lista de categorías**

Captura de la interface lista de categorías con la relación de categorías respectivas de cada productos disponible en la tienda.



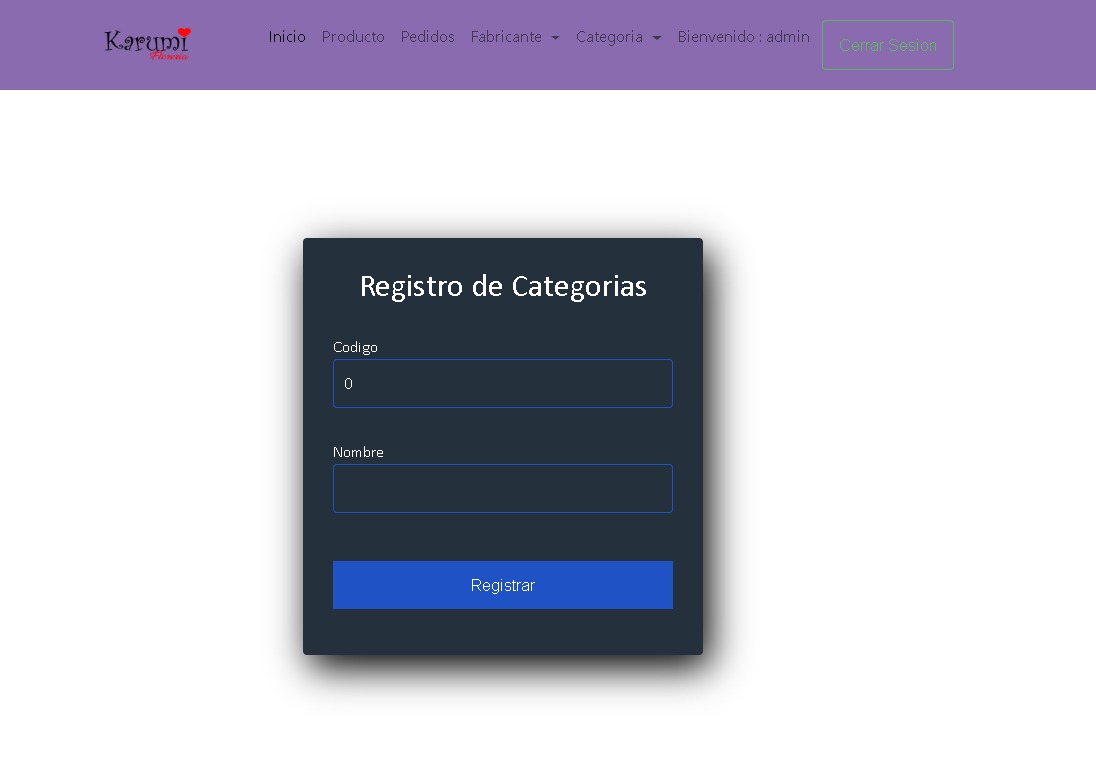
* + 1. **Interface registro de fabricantes**

Captura de la interface registro de fabricantes con los campos respectivos para la creación de un fabricante nuevo.



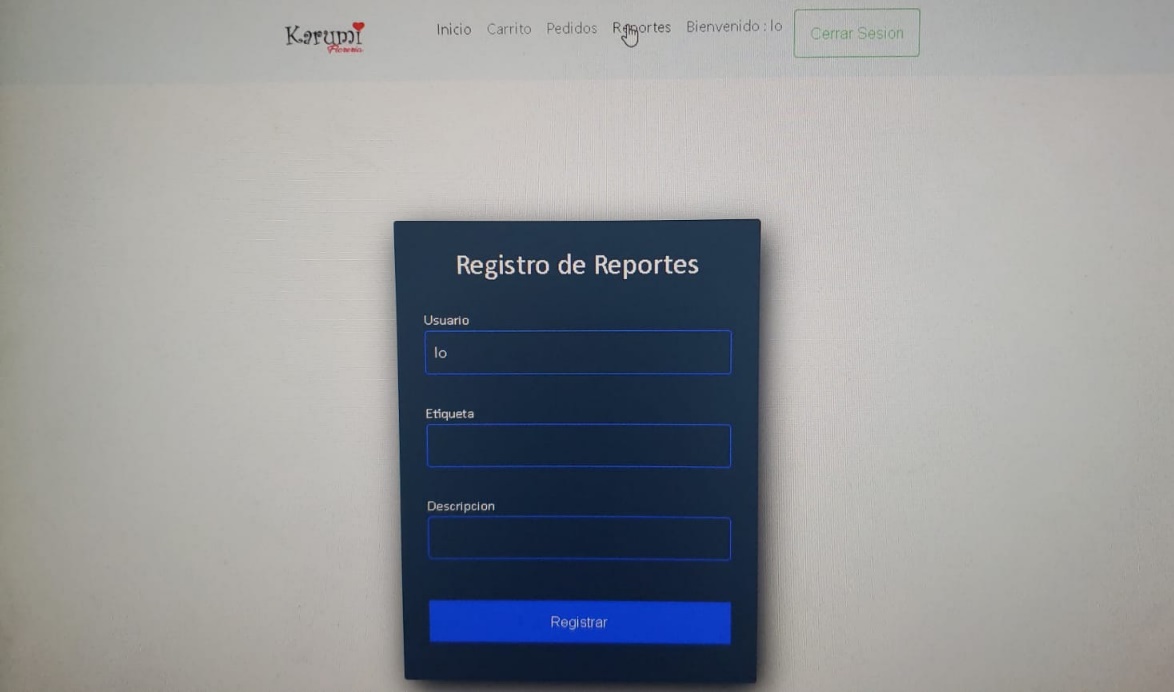
* + 1. **Interface registro de categorías**

Captura de la interface registro de categorías con los campos respectivos para la creación de una categoría nueva.



* + 1. **Interface registro de reportes**

Captura de la interface registro de reportes con los campos respectivos para la creación de reportes por parte de los clientes.



* + 1. **Interface hardware y software**

El sitio web estará disponible para funcionar en cualquier sistema operativo con conexión a internet.

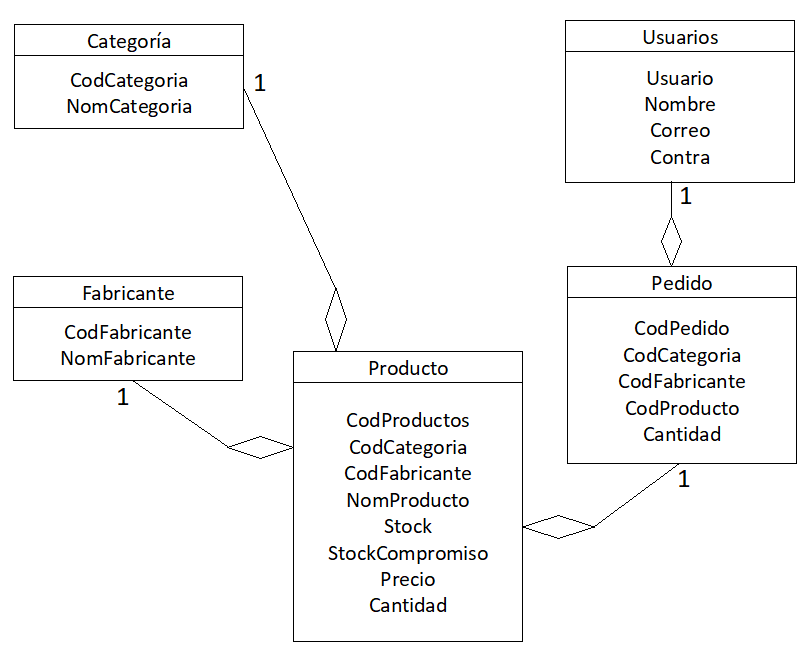
* + - 1. **Atributos**

El sitio web usa una base de datos MySQL donde se almacenan todos los datos con respecto a usuarios, categorías, fabricantes, productos, etc. Todas las consultas a la base de datos se realizan por parte del servidor web mediante PHP.

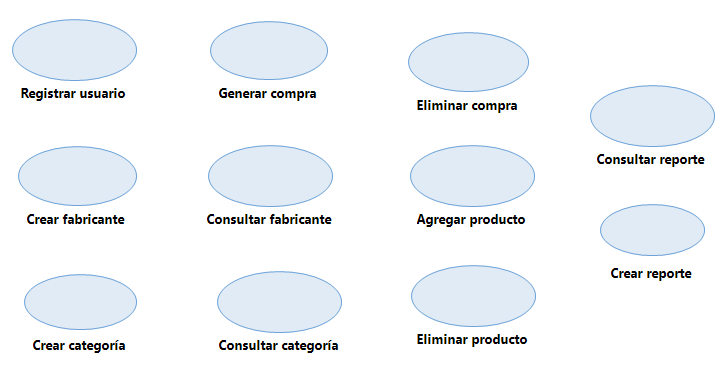
# ANÁLISIS

## 4.1 Diagrama de clases

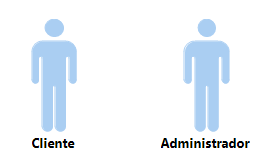
Usuario: Representa a los clientes que se crean una cuenta. Accediendo al sistema con el nombre y contraseña respectiva.  
Pedido: Representa todos los productos seleccionados por el usuario en su compra.   
Producto: Contiene toda la información del producto seleccionado.  
Categoría: Contiene el código y nombre de la categoría para su fácil selección.  
Fabricante: Contiene el código y nombre del fabricante para su fácil selección.



## 4.2 Diagrama de casos de uso



## 4.3 Actores



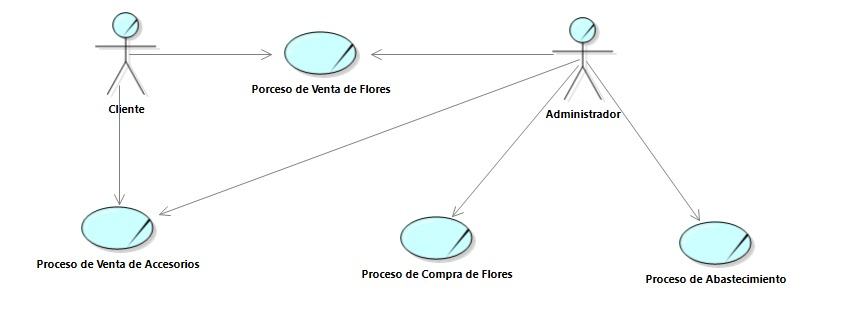
## 4.4 Diagrama general de caso de uso

Proceso de venta de flores: El cliente es quien se beneficia de dicho proceso al adquirir los productos que desee. El administrador es quien se encarga de actualizar los productos en la tienda.

Proceso de venta de Accesorios: El cliente tiene la opción de comprar accesorios además de flores.

Proceso de compra de flores: El administrador se encarga de las compras internas de la tienda.

Proceso de abastecimiento: El administrador se encarga de actualizar el stock y productos en el sitio web.



# DESCRIPCIÓN

## 5.1 Capa de presentación

Datos de las tablas

Datos de la tabla categoría:

public class categoriabe {

private int codigo;

private String nomcodigo;

public categoriabe() {

}

public categoriabe(int codigo, String nomcodigo) {

this.codigo = codigo;

this.nomcodigo = nomcodigo;

}

public categoriabe(categoriabe e) {

throw new UnsupportedOperationException("Not supported yet.");

}

public int getCodigo() {

return codigo;

}

public void setCodigo(int codigo) {

this.codigo = codigo;

}

public String getNomcodigo() {

return nomcodigo;

}

public void setNomcodigo(String nomcodigo) {

this.nomcodigo = nomcodigo;

}

}

Datos de la tabla fabricante:

public class fabricantebe {

private int categoria;

private int codfabricante;

private String nomfabricante;

public fabricantebe() {

}

public fabricantebe(int categoria, int codfabricante, String nomfabricante) {

this.categoria = categoria;

this.codfabricante = codfabricante;

this.nomfabricante = nomfabricante;

}

public int getCategoria() {

return categoria;

}

public void setCategoria(int categoria) {

this.categoria = categoria;

}

public int getCodfabricante() {

return codfabricante;

}

public void setCodfabricante(int codfabricante) {

this.codfabricante = codfabricante;

}

public String getNomfabricante() {

return nomfabricante;

}

public void setNomfabricante(String nomfabricante) {

this.nomfabricante = nomfabricante;

}

}

Datos de la tabla usuario:

public class usuario {

private String nombre;

private String contra;

public usuario() {

}

public usuario(String nombre, String contra) {

this.nombre = nombre;

this.contra = contra;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public String getContra() {

return contra;

}

public void setContra(String contra) {

this.contra = contra;

}

}

Datos de la tabla producto:

public class productobe {

private String codigo;

private String nombre;

private int stock;

private int compromiso;

private int precio;

private int cantidad;

private int fabricante;

private int categoria;

private InputStream img;

public productobe() {

}

public productobe(String codigo, String nombre, int stock, int compromiso, int precio, int cantidad, int fabricante, int categoria, InputStream img) {

this.codigo = codigo;

this.nombre = nombre;

this.stock = stock;

this.compromiso = compromiso;

this.precio = precio;

this.cantidad = cantidad;

this.fabricante = fabricante;

this.categoria = categoria;

this.img = img;

}

public String getCodigo() {

return codigo;

}

public void setCodigo(String codigo) {

this.codigo = codigo;

}

public String getNombre() {

return nombre;

}

public void setNombre(String nombre) {

this.nombre = nombre;

}

public int getStock() {

return stock;

}

public void setStock(int stock) {

this.stock = stock;

}

public int getCompromiso() {

return compromiso;

}

public void setCompromiso(int compromiso) {

this.compromiso = compromiso;

}

public int getPrecio() {

return precio;

}

public void setPrecio(int precio) {

this.precio = precio;

}

public int getCantidad() {

return cantidad;

}

public void setCantidad(int cantidad) {

this.cantidad = cantidad;

}

public int getFabricante() {

return fabricante;

}

public void setFabricante(int fabricante) {

this.fabricante = fabricante;

}

public int getCategoria() {

return categoria;

}

public void setCategoria(int categoria) {

this.categoria = categoria;

}

public InputStream getImg() {

return img;

}

public void setImg(InputStream img) {

this.img = img;

}

}

Datos de la tabla pedido:

public class Pedido {

int codpedido;

int codcategoria;

int codfabricante;

int codproducto;

int cantidad;

public Pedido() {

}

public Pedido(int codpedido, int codcategoria, int codfabricante, int codproducto, int cantidad) {

this.codpedido = codpedido;

this.codcategoria = codcategoria;

this.codfabricante = codfabricante;

this.codproducto = codproducto;

this.cantidad = cantidad;

}

public int getCodpedido() {

return codpedido;

}

public void setCodpedido(int codpedido) {

this.codpedido = codpedido;

}

public int getCodcategoria() {

return codcategoria;

}

public void setCodcategoria(int codcategoria) {

this.codcategoria = codcategoria;

}

public int getCodfabricante() {

return codfabricante;

}

public void setCodfabricante(int codfabricante) {

this.codfabricante = codfabricante;

}

public int getCodproducto() {

return codproducto;

}

public void setCodproducto(int codproducto) {

this.codproducto = codproducto;

}

public int getCantidad() {

return cantidad;

}

public void setCantidad(int cantidad) {

this.cantidad = cantidad;

}

}

Conexión con la base de datos:

public class bd {

Connection c=null;

Statement sta =null;

Result rs = null;

public bd() {

try{

Class.forName("com.mysql.jdbc.Driver");

c=DriverManager.getConnection("jdbc:mysql://localhost:3306/proyecto?user=root&password=root");

System.err.println("Bueno");

}catch(Exception e){

System.err.println("Error"+e);

}

}

public Connection getConnection(){

return c;

}

public static void main(String[] args) {

bd con = new bd();

}

}