

Universidad InterNaciones

Análisis de sistemas

Mgtr. Franco Nery López

Proyecto Final

José Antonio Miranda Corzo

15030016

## Fase Análisis

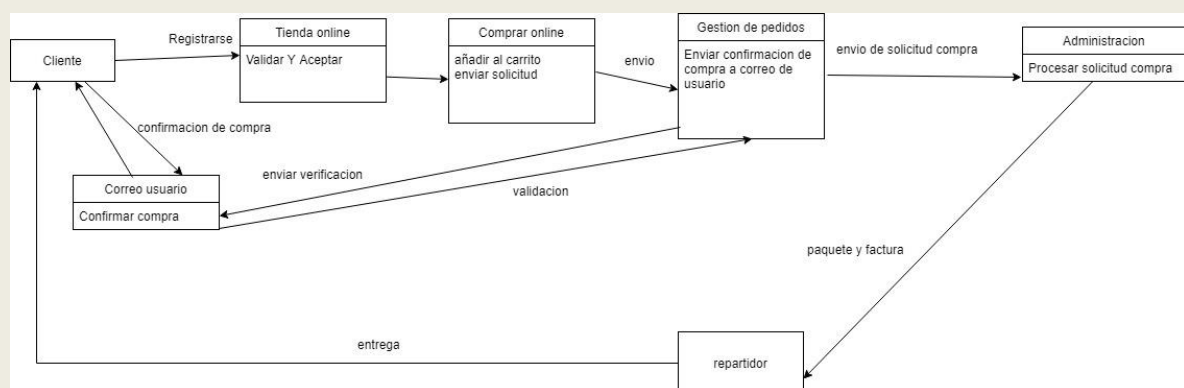
**Descripción del proyecto:** el proyecto a desarrollar es el desarrollo de una aplicación que permita a un cliente realizar una compra online para la Ferretería “La Ferre”, para agilización de gestión de pedidos y que facilite a los clientes la compra en línea de manera cómoda. La arquitectura que se utilizara para dicha solicitud será una arquitectura cliente servidor, el software solicitado estará situado en el servidor que pertenezca a la Ferretería, a la cual se tendrá acceso por internet, toda la actividad se desarrolla en el navegador. Será de mucha utilidad, pues ayudara a verificar el mantenimiento de artículos, catalogo y registro de usuarios, generación de informes, estado de pedidos, facturación.

**Funcionalidades:** se detalla el desglose de funcionalidades que tendrá dicho sistema.

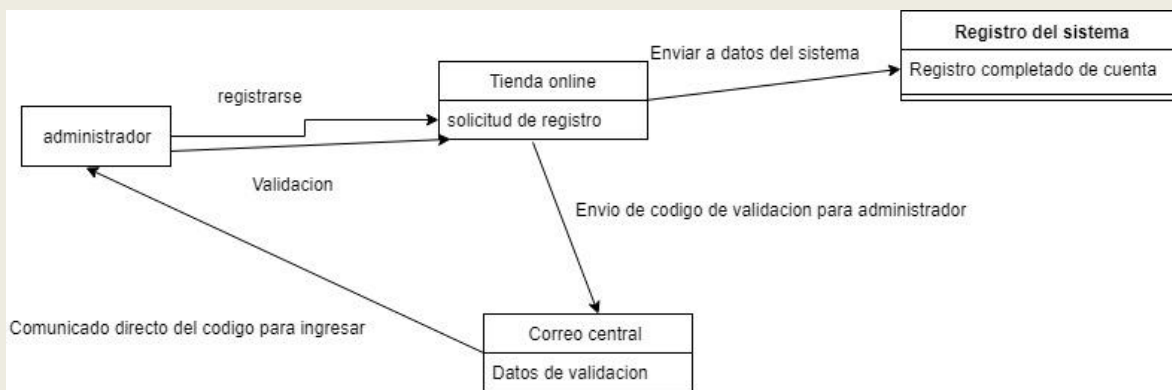
- Catálogos: jerarquía de del catálogo para incluir los artículos correspondientes por medio de navegación dentro del catálogo o búsqueda del usuario, permitiendo añadir al “carrito” para realizar el pedido.
- La compra: permite el envío del pedido del cliente, para que posteriormente se pueda verificar el estado del pedido por parte de los encargados, el pedido se entregara en un lugar específico que determine la ferretería.
- Gestión de los datos: permite que un usuario con el rol de trabajador de la empresa pueda acceder para verificar: modificar categorías, artículos, facturación y usuarios.
- Información enlistada: datos sobre pedidos en el sistema, artículos disponibles o no disponibles, usuarios y sus pedidos.

**Seguridad del sistema:** la seguridad del sistema constara de lo siguiente: los usuarios deben de identificar sus datos cuando acceden a realizar una compra, solo los usuarios con rol de trabajador tendrán acceso a la administración por obvio.

Para la seguridad del sistema de cuentas de usuario se contempló lo siguiente al momento de la realización de pedidos



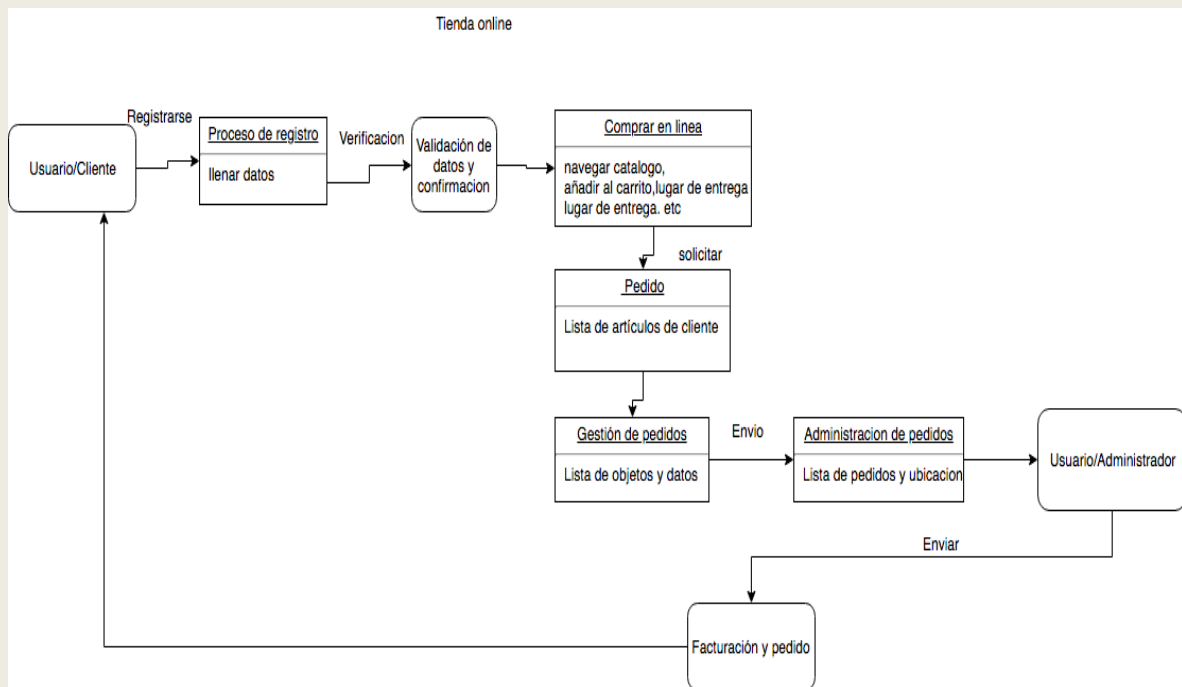
## Seguridad para administrador



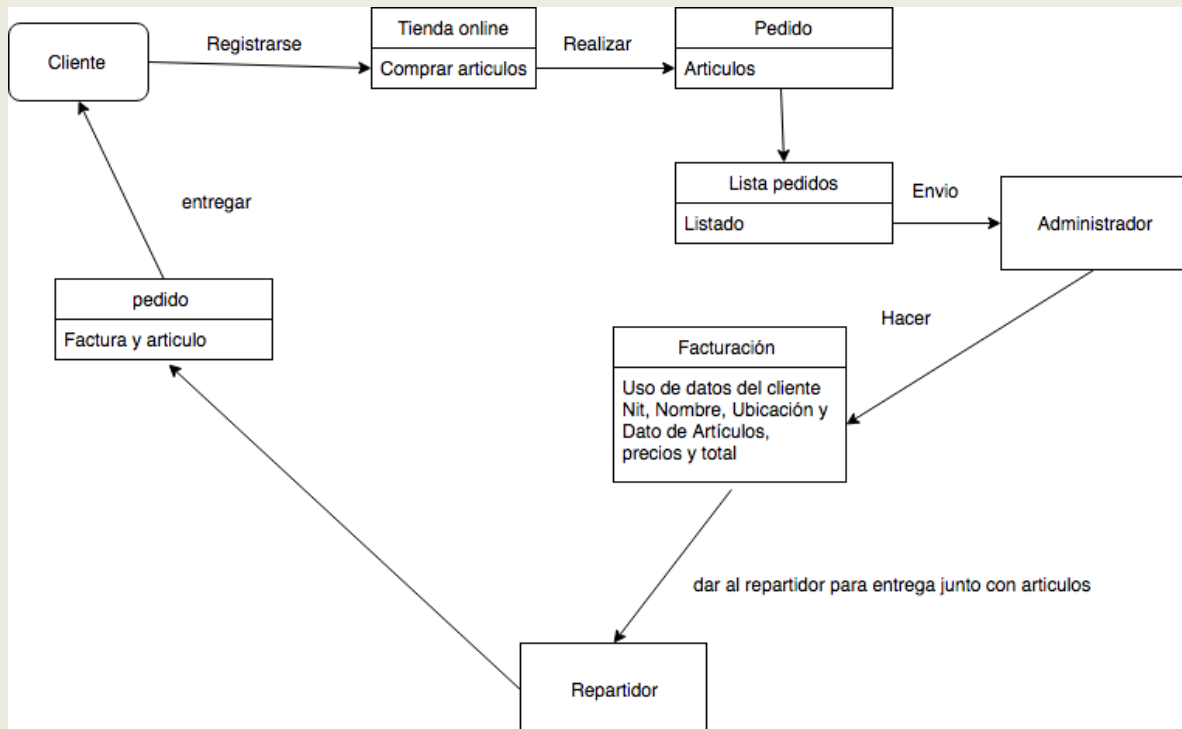
El administrador cuando es contratado y debe crear su registro en el sistema, al ingresar, se le envía un código de verificación al correo que tiene designado la empresa, para que lleguen los códigos de validación de administradores, para que después el encargado de la empresa de enviar los códigos, le indique al administrador el código, dentro de los campos que pide el sistema para registro

## Diagramas

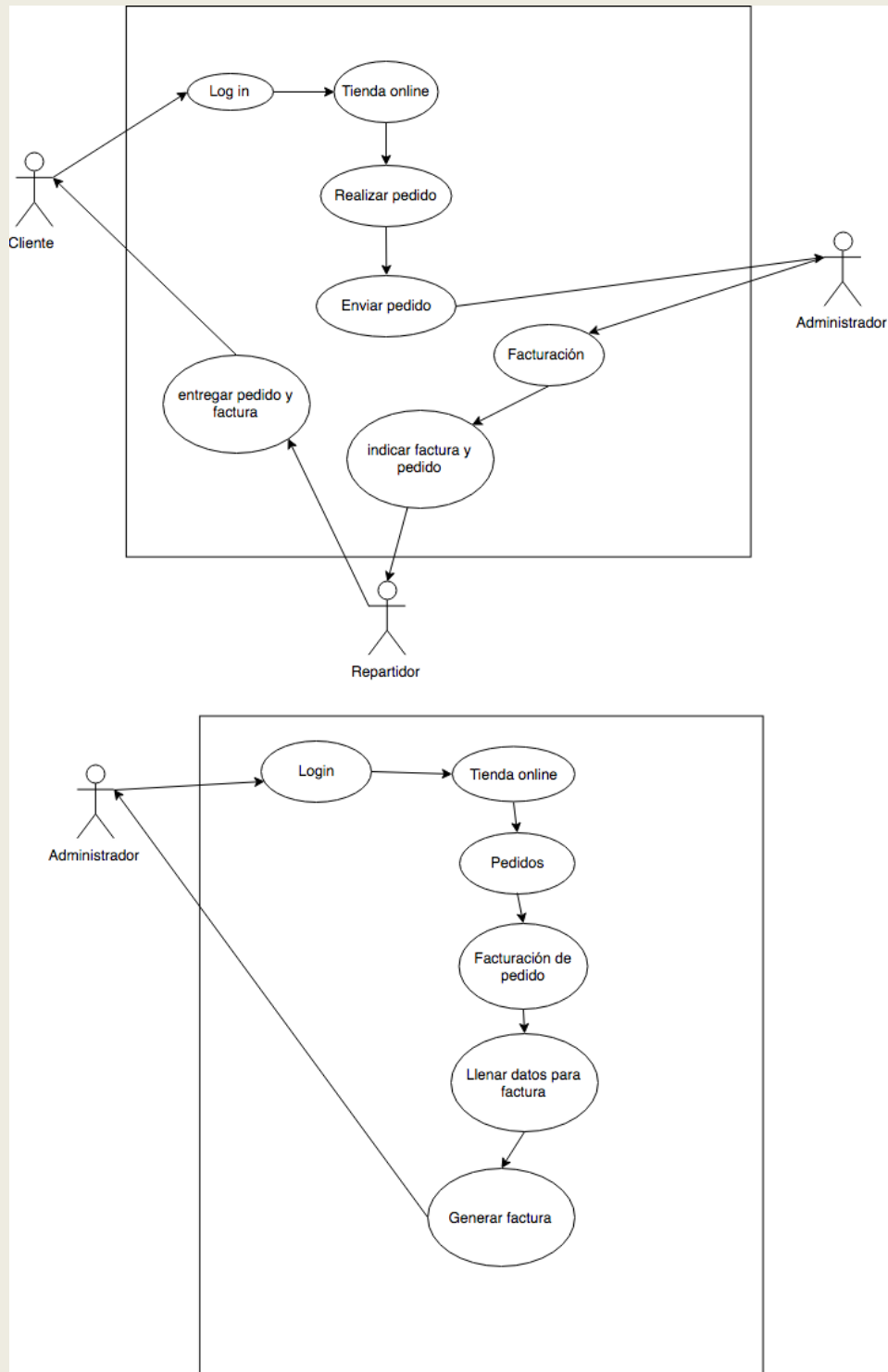
### DFD general que engloba al cliente y administrador nivel 1



### DFD de facturación nivel

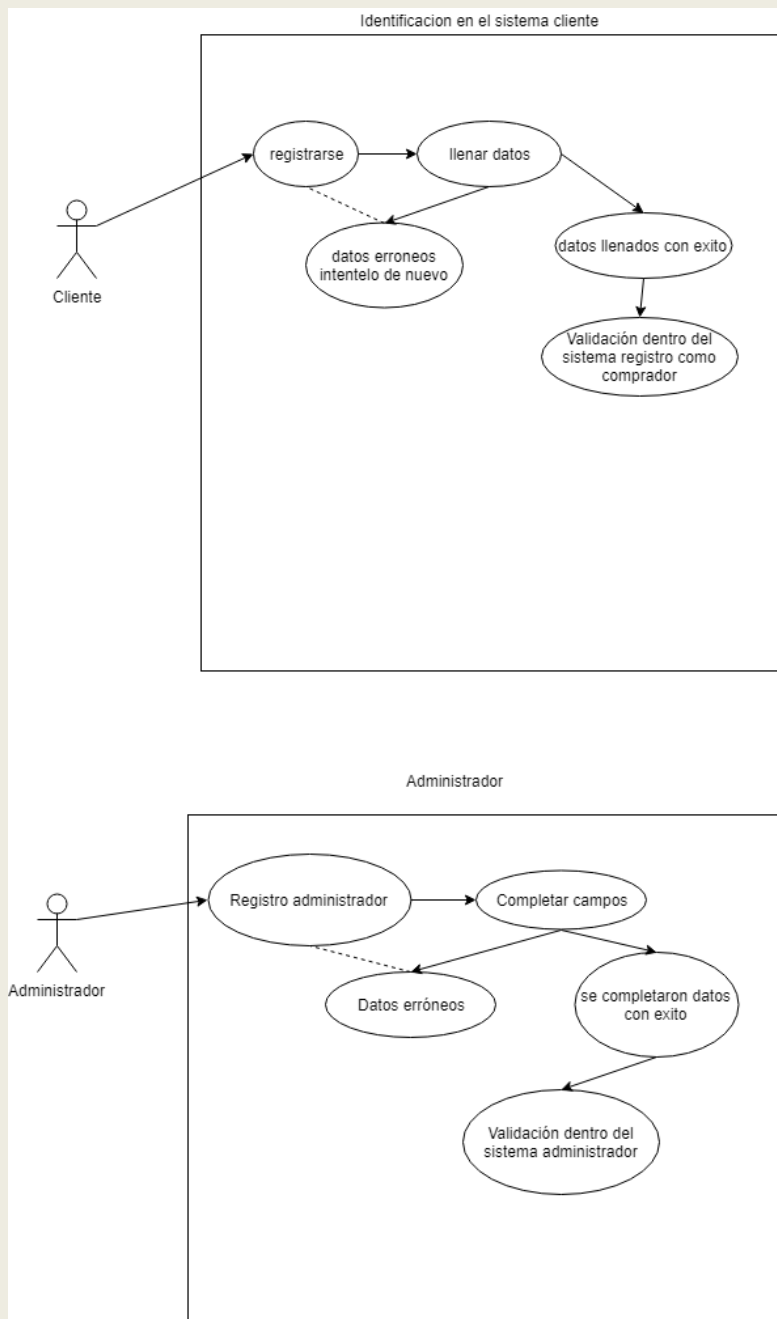


### Caso de uso de facturación desde ambos puntos de vista(cliente / administrador)

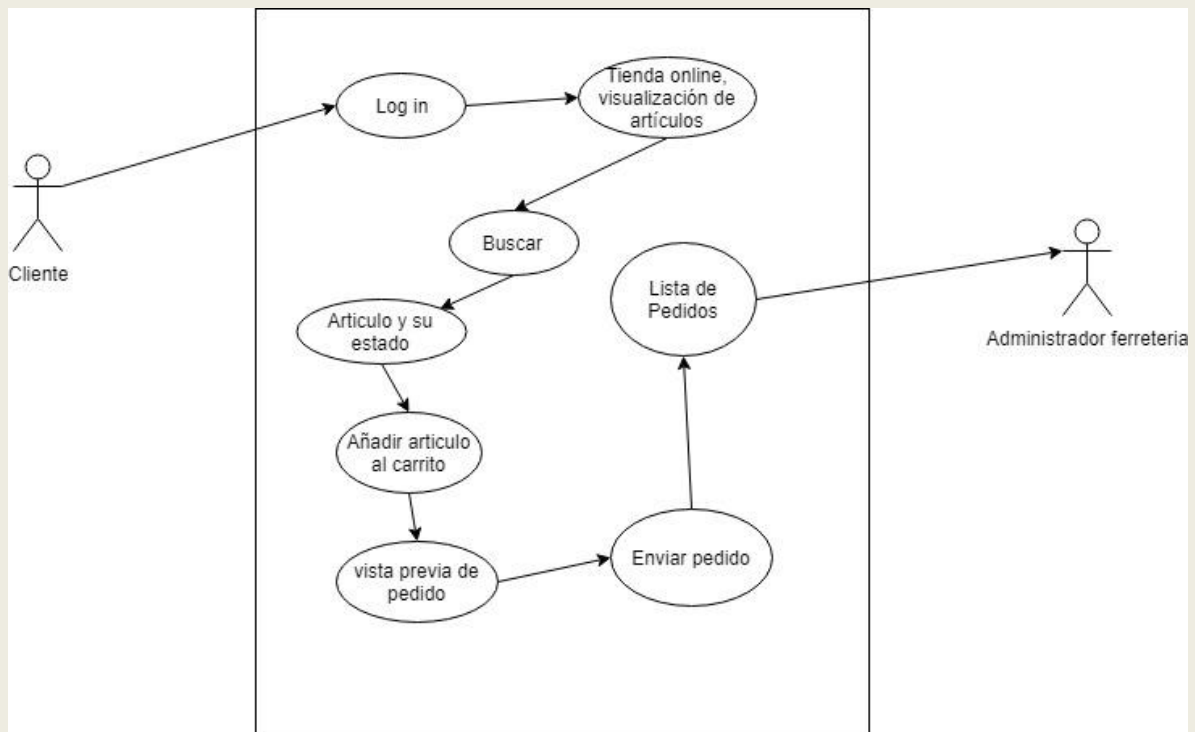


**Nota:** para la generación de factura solo se tendrán contemplados los datos del nombre, nit, ubicación, artículos y valor total

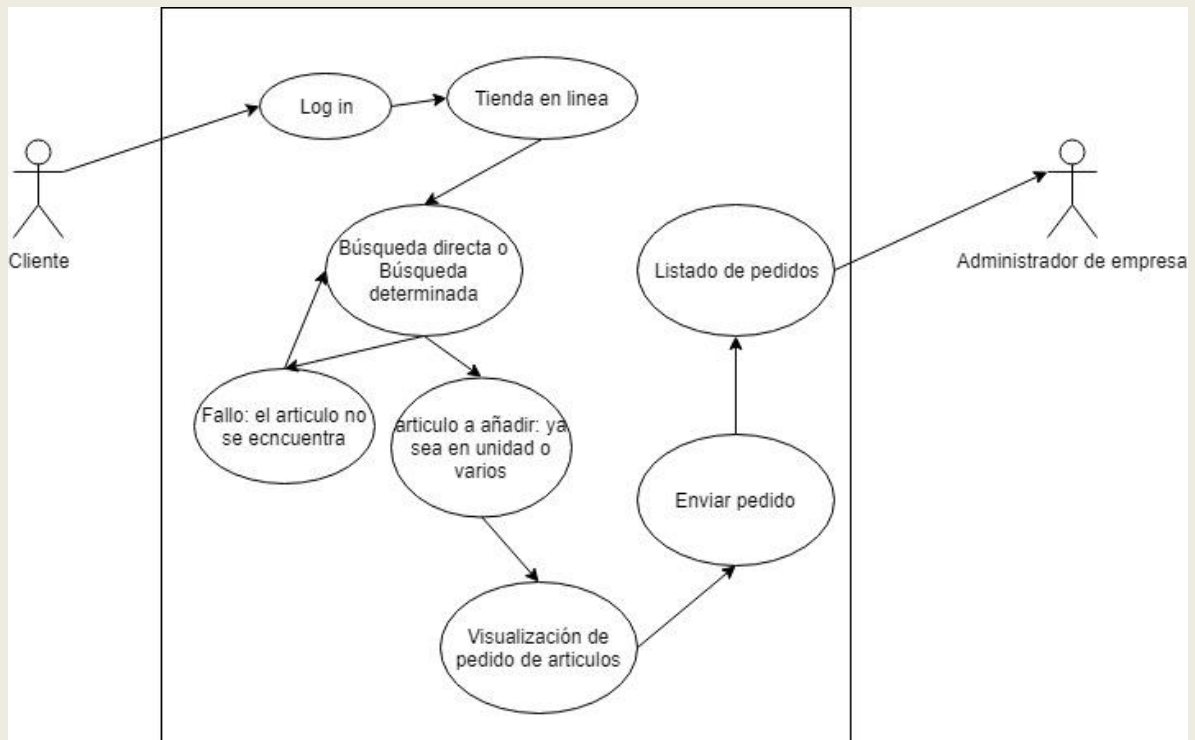
## Caso de uso identificación en el sistema (cliente)



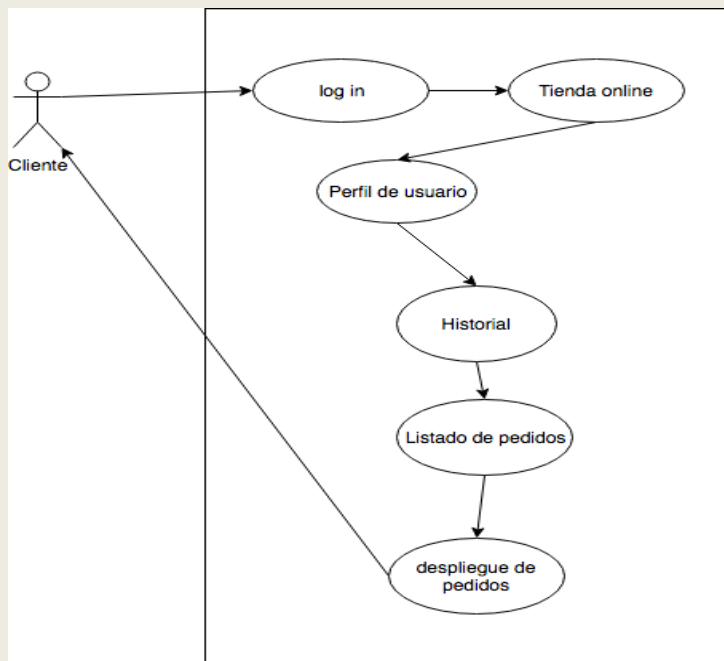
### Caso de uso búsqueda en el catalogo



### Añadir artículo a comprar y envío de pedidos

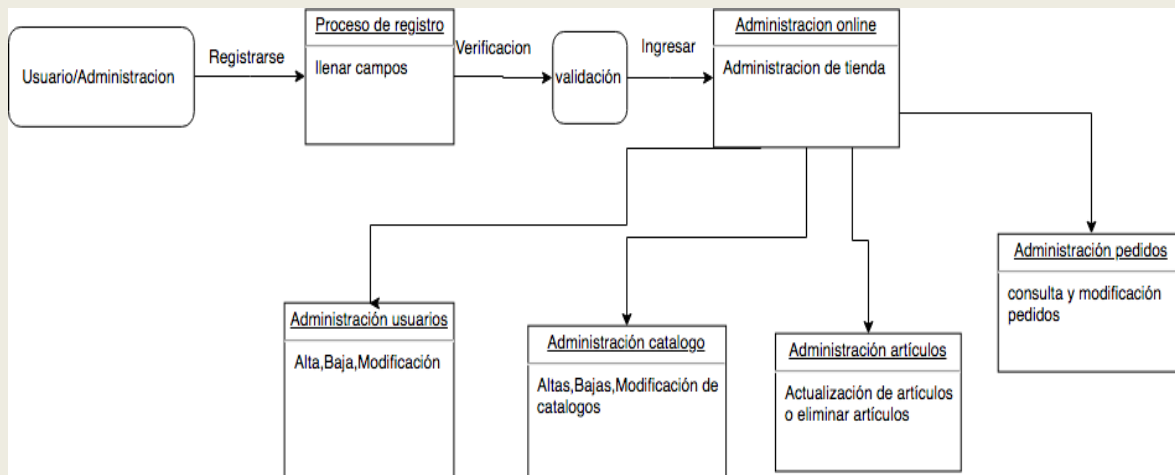


### Visualización de historial de pedidos:

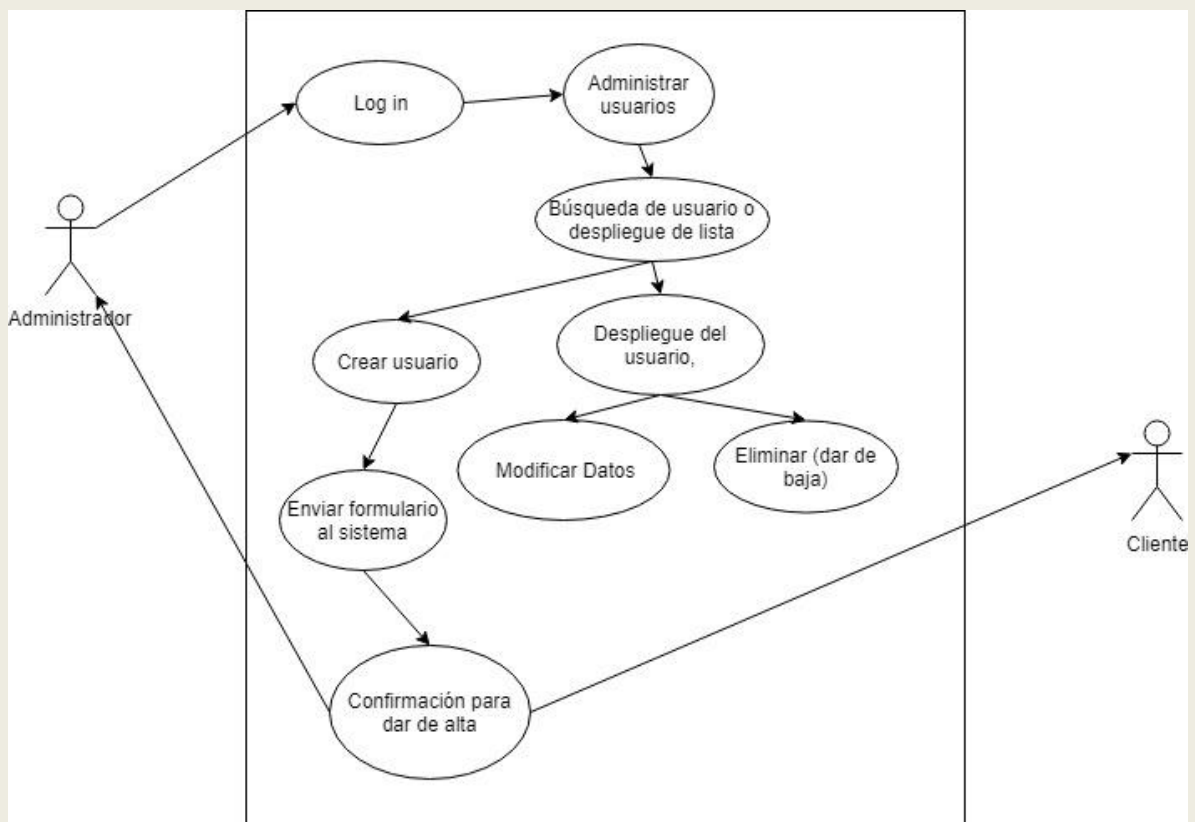




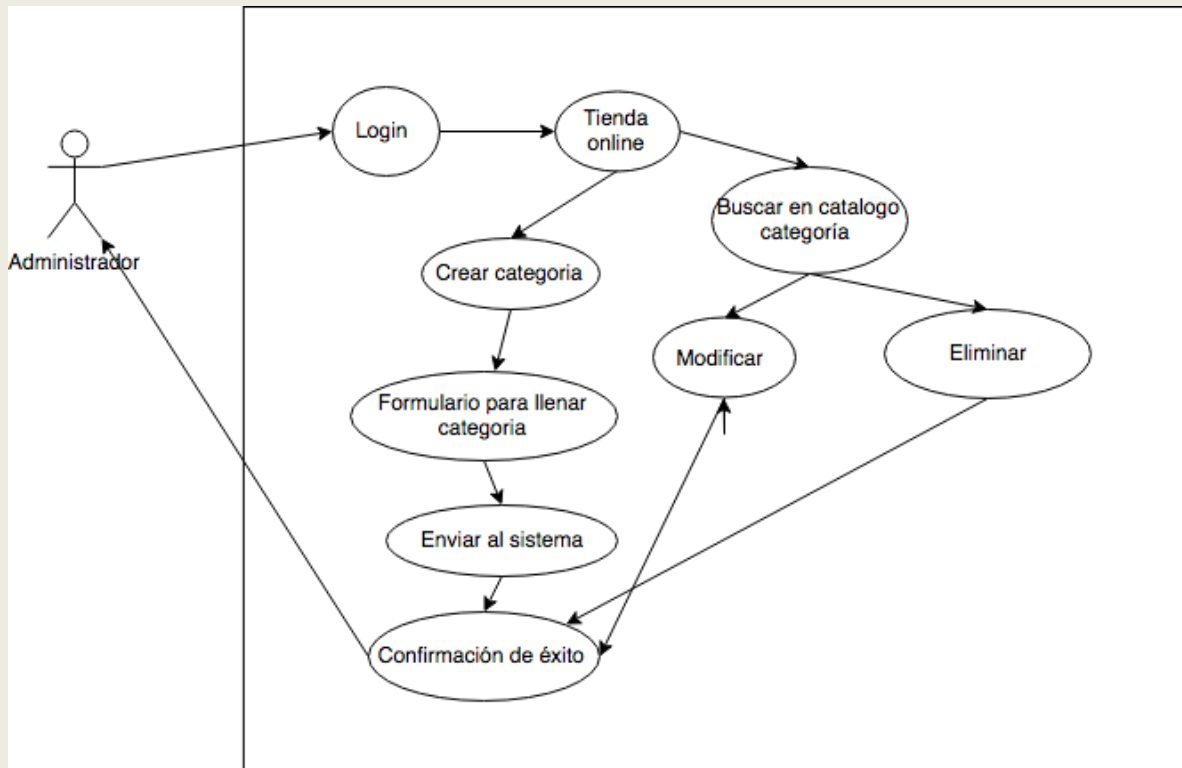
## DFD general de uso de administrador nivel 2 (visto solo desde el administrador)



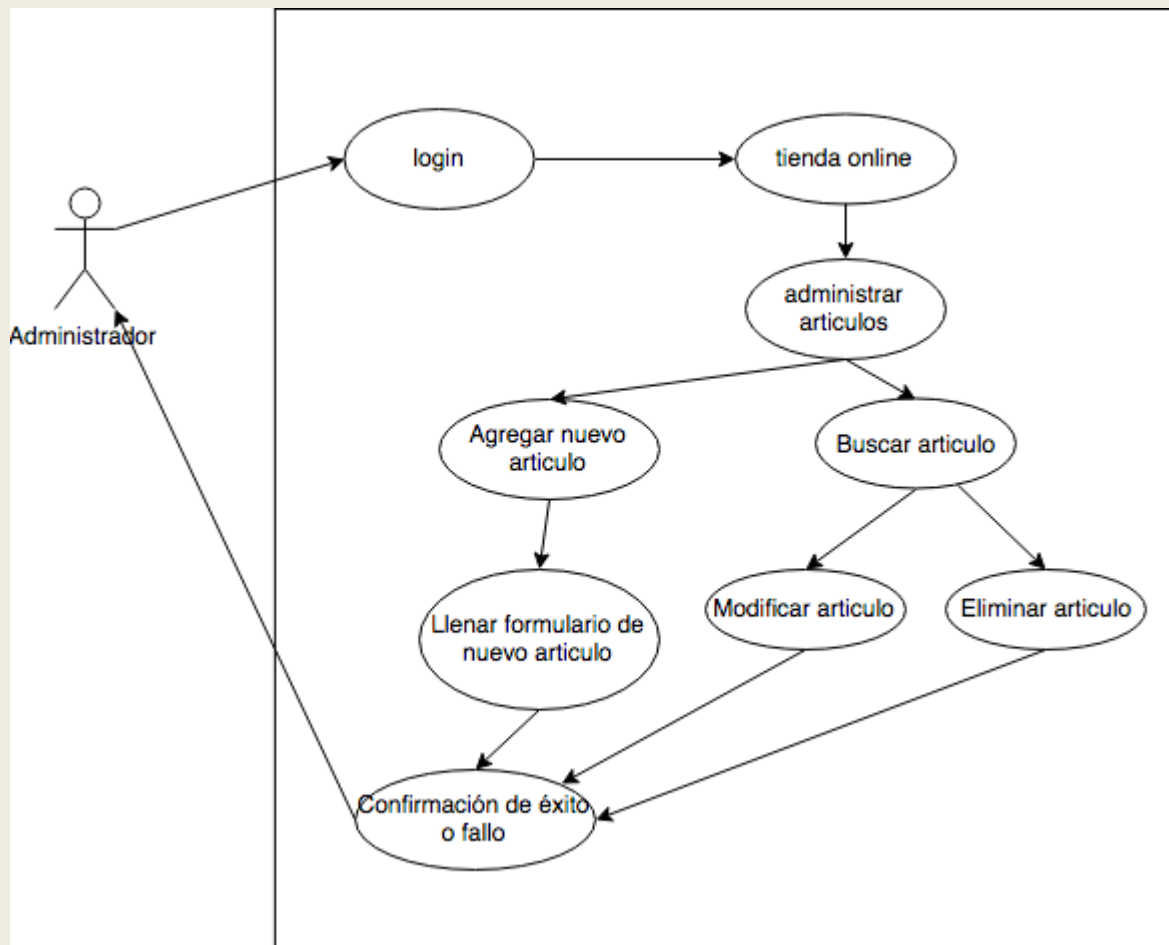
## Administración de usuarios



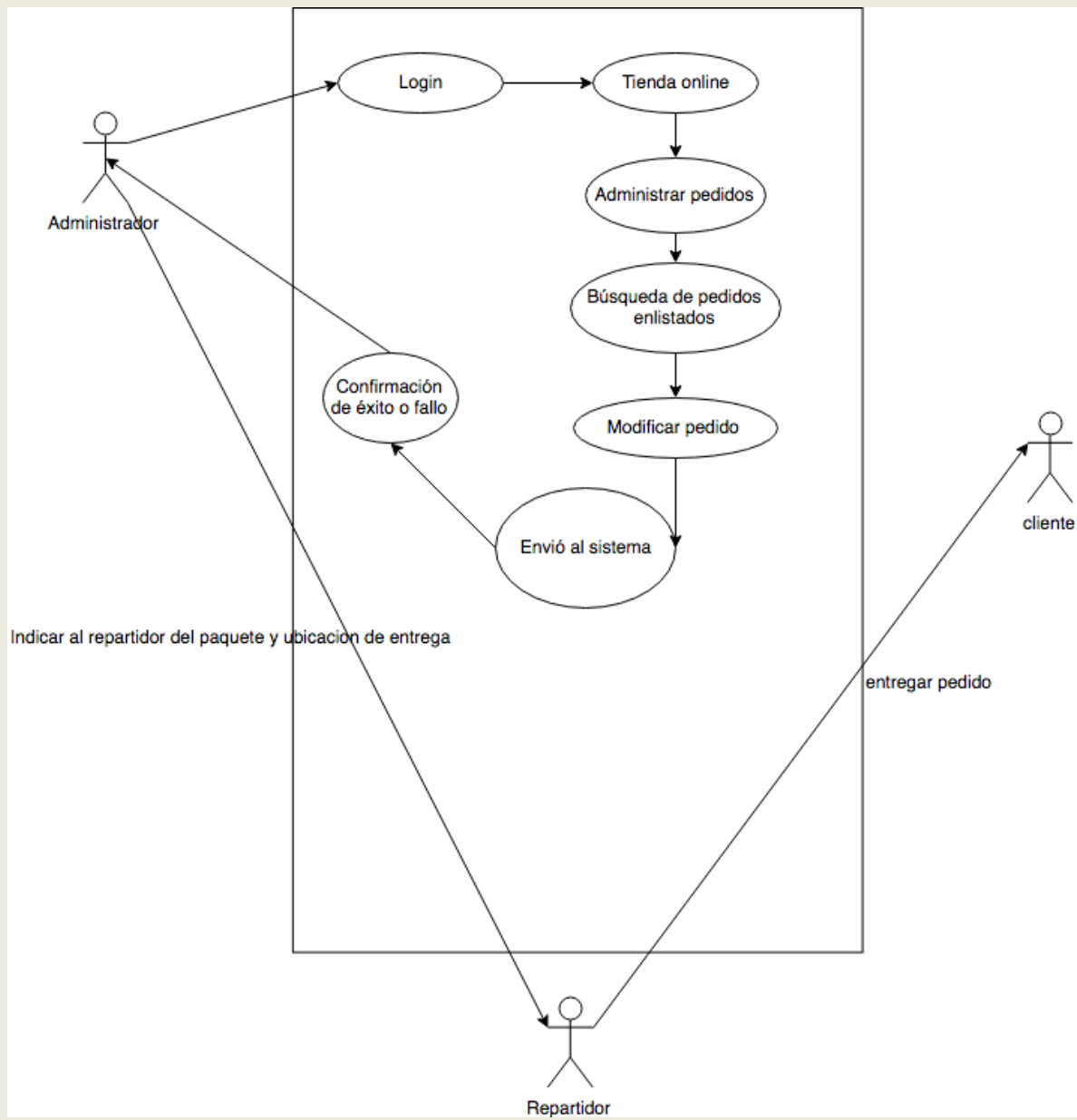
## Administrar catalogo



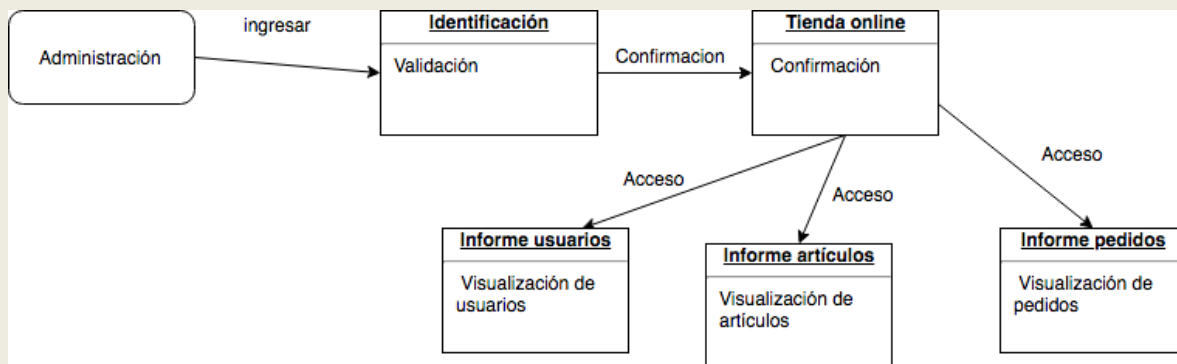
## Administrar artículos



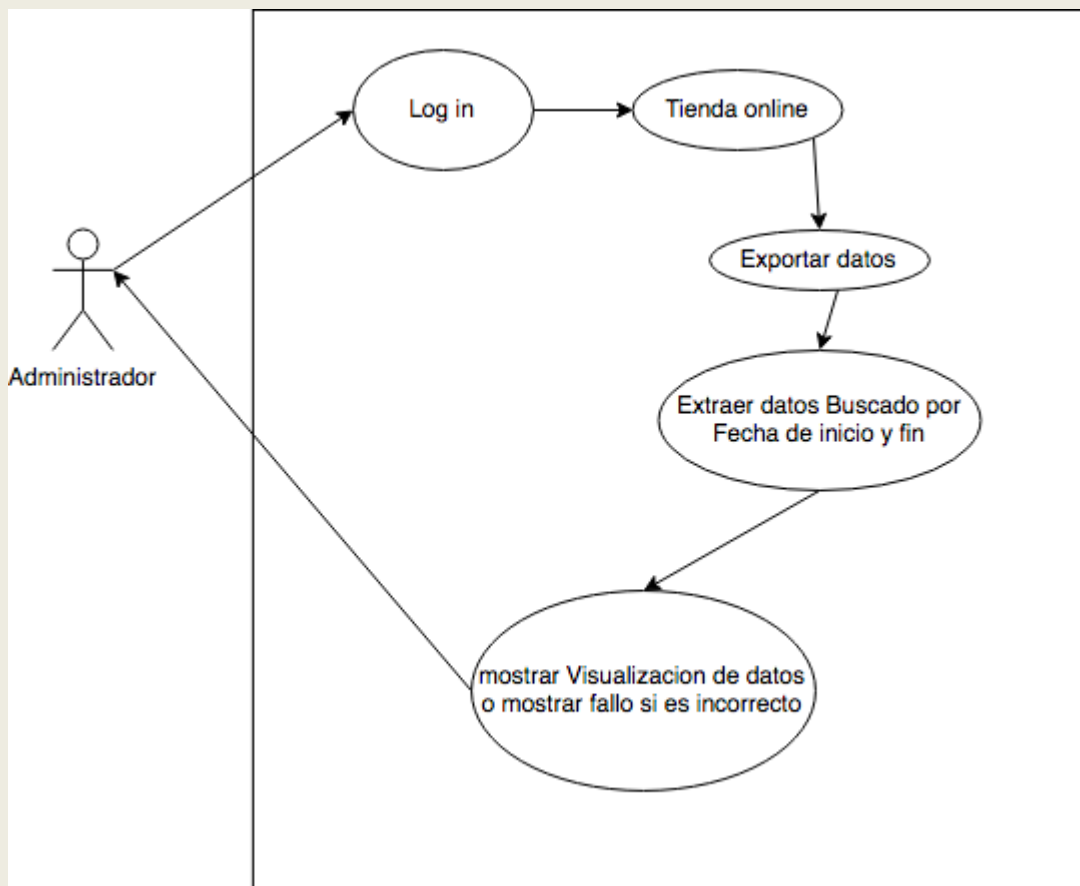
## Administrar pedidos



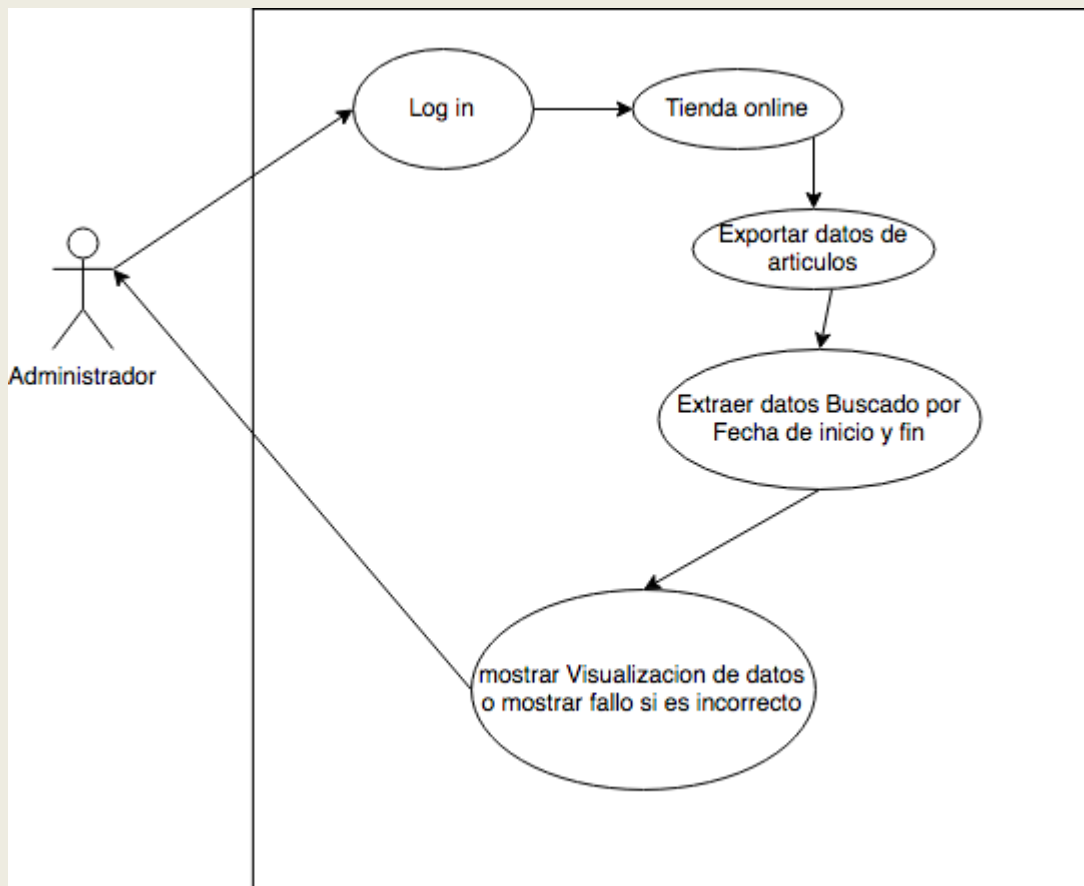
## DFD listados de la aplicación



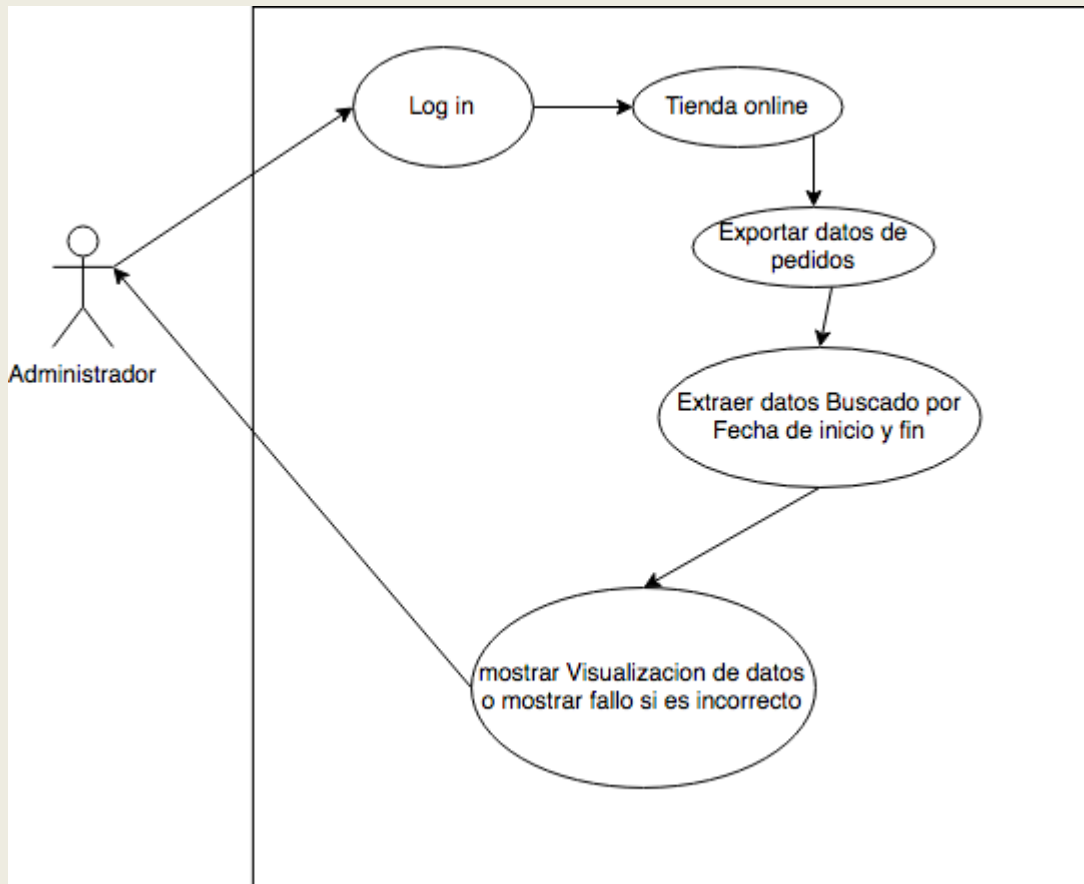
## Informe usuarios



## Informe artículos



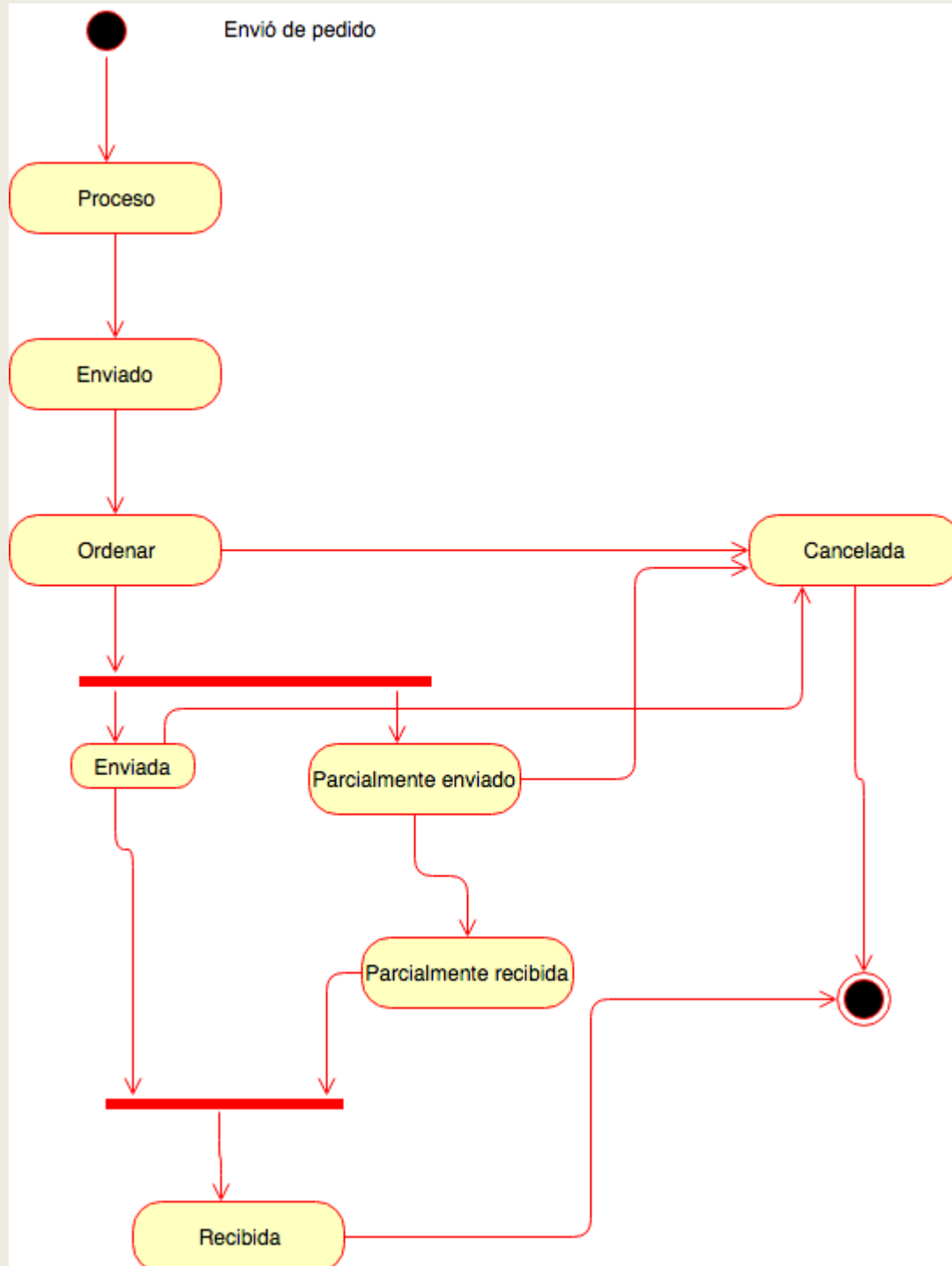
## Informe pedidos



## Diagramas de estado, colaboración y secuencia

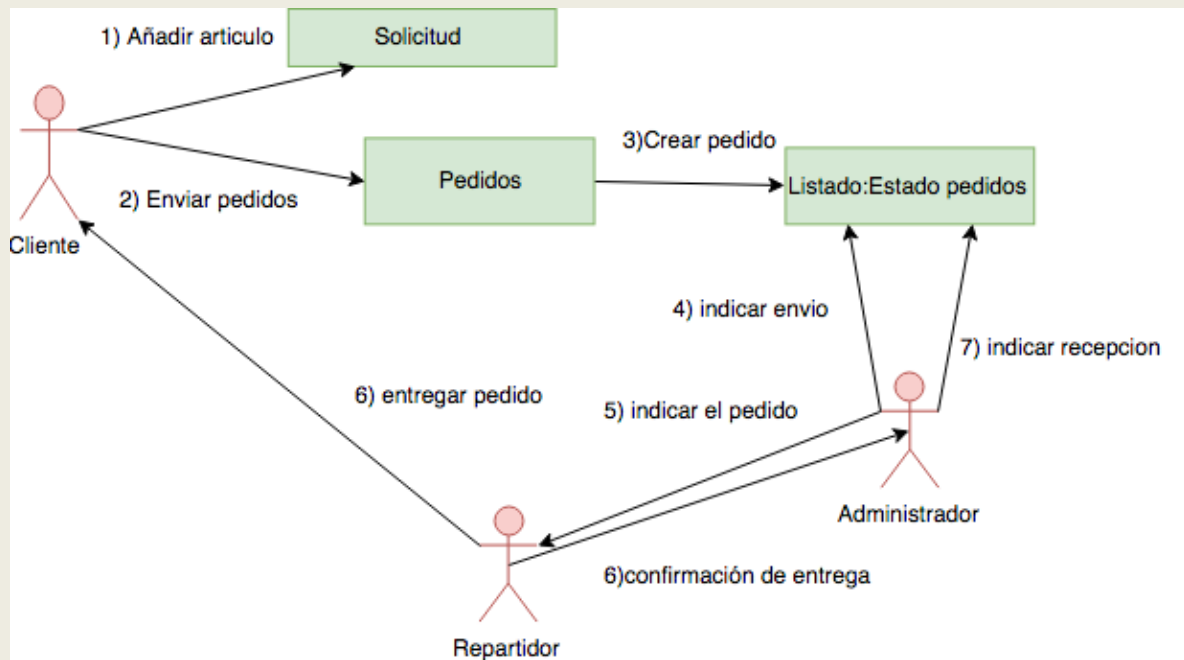
se detalla el proceso más importante dentro de la venta

### Estado

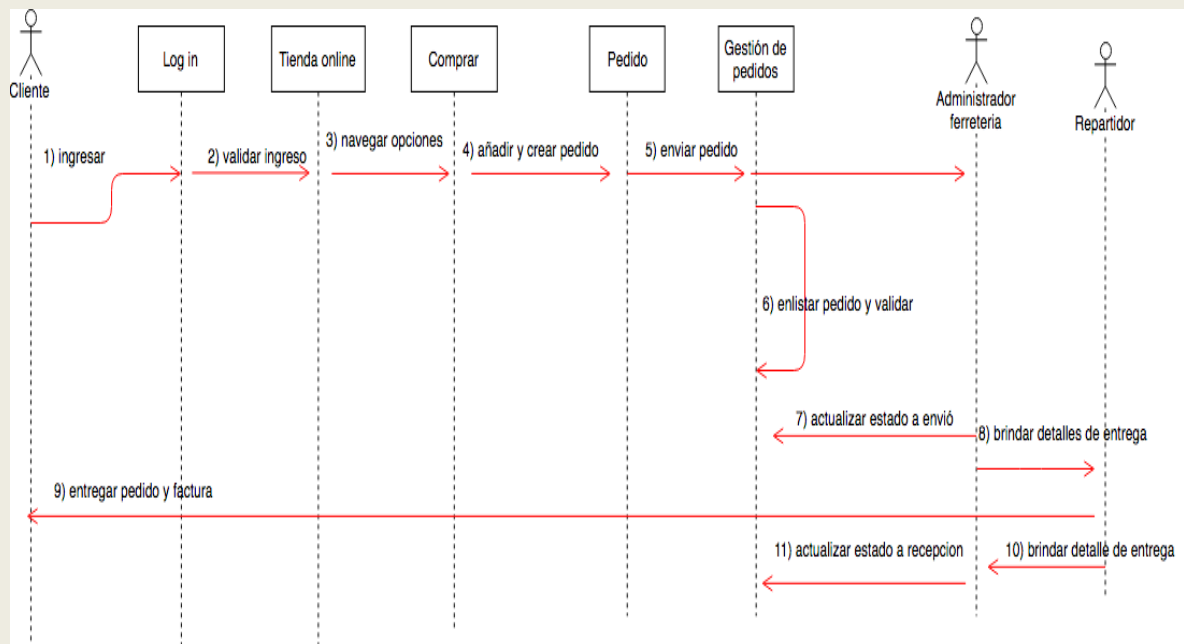




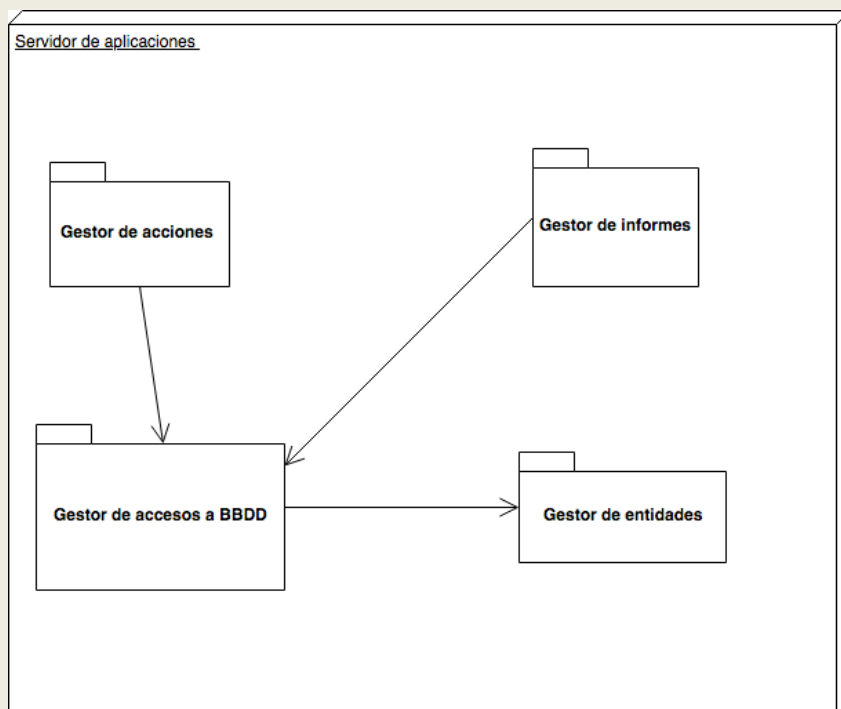
## Diagrama de colaboración



## Diagrama de secuencia



## Diagrama de paquetes



## Fase Diseño y Desarrollo

### Diseño de pantallas

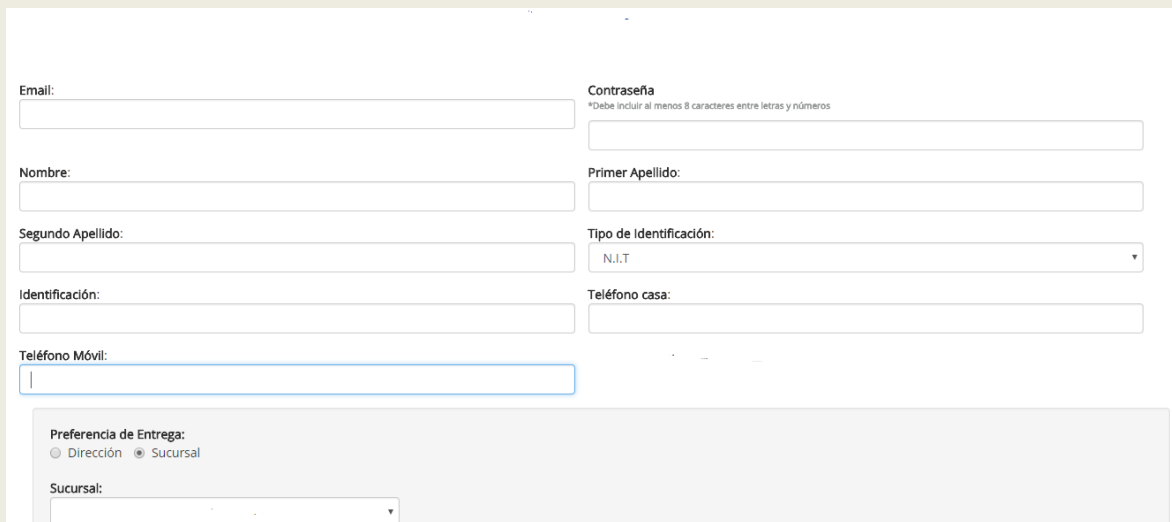
Pantalla de inicio: cuando navega un usuario por primera vez donde se visualizan artículos promociones etc, el logotipo, donde puede visualizar productos, categorías, buscar o iniciar sesión

The wireframe shows a home page layout. At the top left is a box labeled 'Logotipo'. At the top right is a blue button labeled 'Iniciar sesion'. Below the logo is a navigation bar with three tabs: 'Inicio' (highlighted in blue), 'Productos', and 'Categorías'. To the right of the tabs is a search bar labeled 'Search' with a magnifying glass icon. Below the navigation bar are four rounded rectangular boxes arranged in a 2x2 grid. The top-left box is labeled 'Articulos', the top-right box is labeled 'Articulos', the bottom-left box is labeled 'Articulos', and the bottom-right box is labeled 'Anuncios'. To the right of these boxes is a vertical ellipsis icon.

Al dar click en iniciar sesión saldrá lo siguiente

The wireframe shows a sign-in page layout. At the top is the text 'Sign In' followed by a horizontal line. Below this is the text 'User Name:' followed by a text input field containing the text 'johndoe'. Below that is the text 'Password:' followed by a password input field containing seven asterisks '\*\*\*\*\*'. Below the password field is a blue button labeled 'SIGN IN'. Below the button is a link labeled 'Forgot Password?'. Below the link is another horizontal line. Below the line is the text 'New User' followed by a blue button labeled 'SIGN UP'.

Al darle click en registrarse saldrá una ventana parecida a esta que pedirá datos como



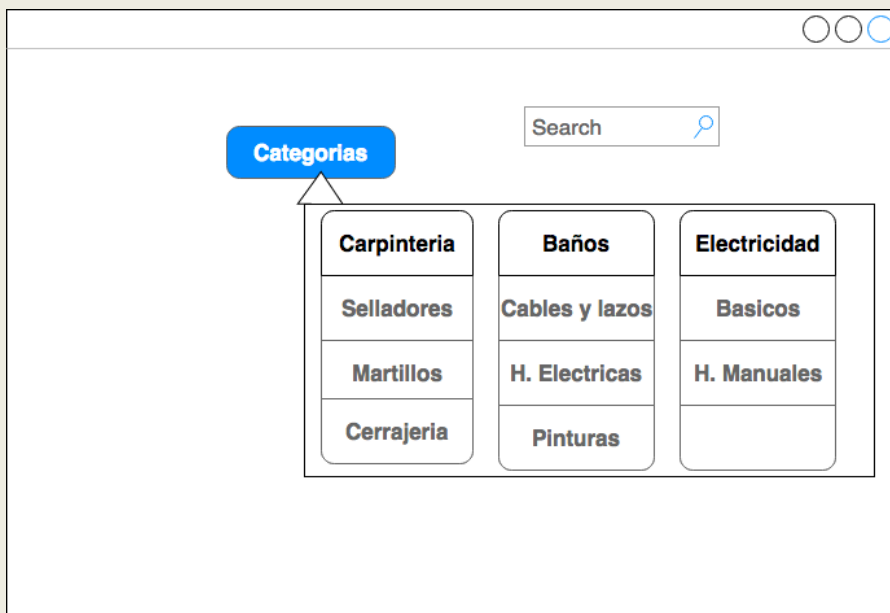
Registration form interface with the following fields:

- Email:
- Contraseña:  \*Debe incluir al menos 8 caracteres entre letras y números
- Nombre:
- Primer Apellido:
- Segundo Apellido:
- Tipo de Identificación:  ▼
- Identificación:
- Teléfono casa:
- Teléfono Móvil:
- Preferencia de Entrega: ☐ Dirección ☒ Sucursal
- Sucursal:

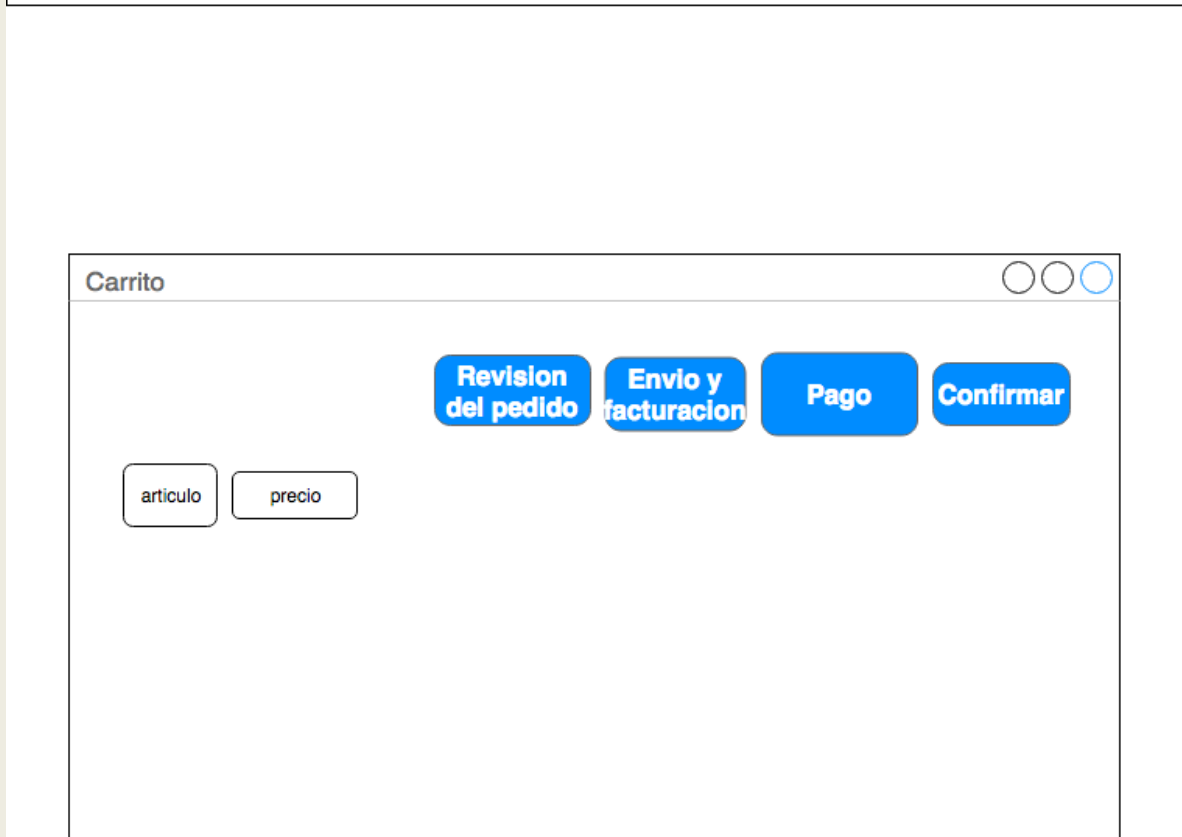
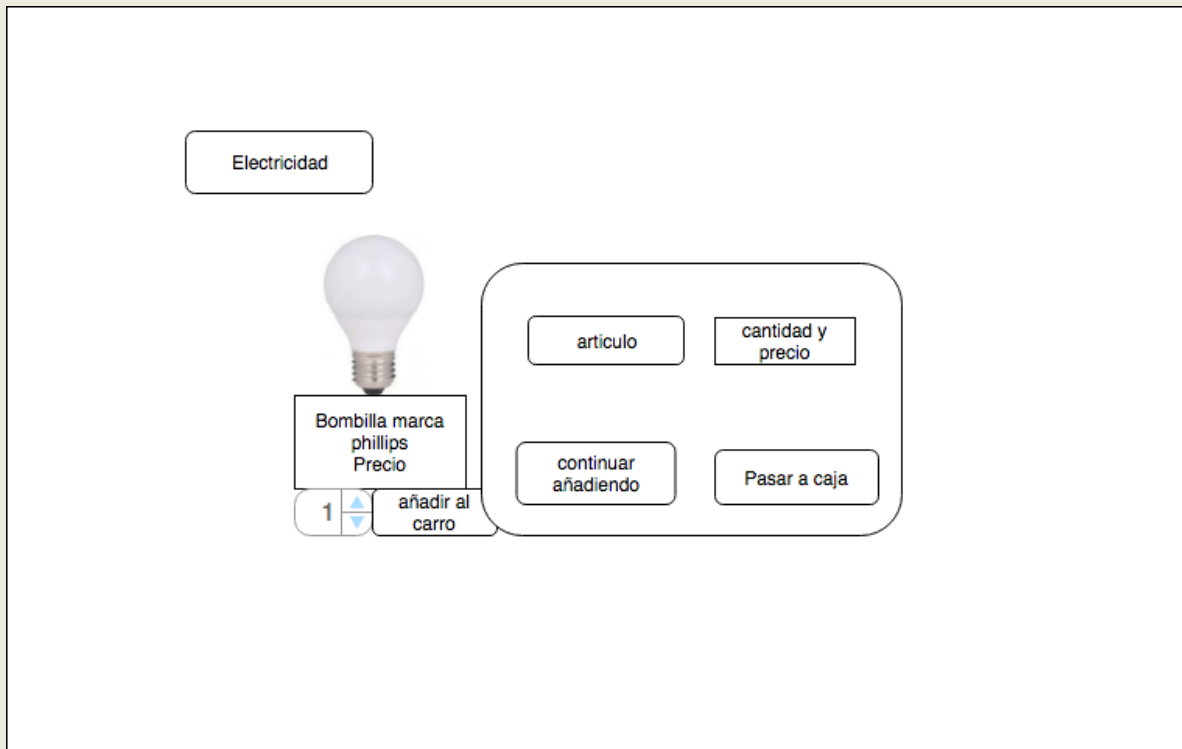
Esta imagen es una referencia, la única diferencia de esta pantalla con la del administrador, es que al administrador le solicitara un campo más para código de confirmación e ingresar al sistema.

Al finalizar al dar click en aceptar confirma mediante el correo y puede ingresar a las secciones del usuario

Navegación en catalogo: al posicionar el puntero en el catalogo como en la primera interfaz desplegara una pequeña ventana que desplegara las categorías o bien se podrá buscar el articulo que desea

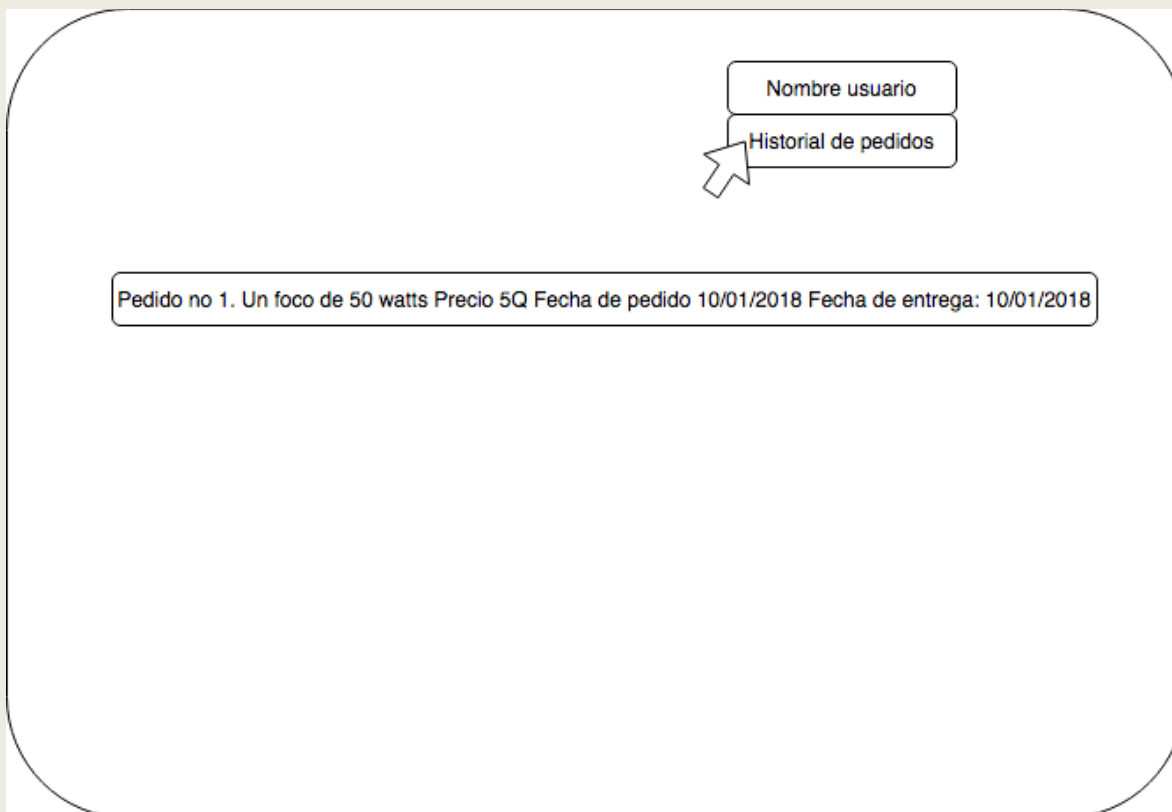


Muestra de cómo se ve la visualización de los productos dentro de las categorías



Al darle click a envío y facturación muestra la ubicación y los datos de su factura para enviar, en pago selecciona el método de pago y despliega las opciones y al darle confirmar se envía, en revisión del pedido se revisa y se añade o elimina un artículo.

Para visualizar el historia bajo el nombre del usuario podrá ver su historial que se enlista



Si el usuario es administrador los artículos tendrán también la opción administrar artículos que permitirá editar o eliminar

## Edición de los artículos

Edición de artículos y categorías

Nueva categoría Crear

**Categorías** Modificar Borrar

**Categorías** Modificar Borrar

**Categorías** Modificar Borrar

**Categorías** Modificar Borrar

Mensaje


Esta seguro de que el cambio es correcto

Cancel OK

Edición artículos y categorías

**Categoría**

Añadir artículo a esta categoría



Sierra eléctrica Precio

Modificar Dar de baja

añadir foto de artículo

Nombre y marca Precio

Guardar Cancelar

Descripción de edición de categoría y artículos de cómo añadir y eliminar

## Pedidos

# Pedidos

**1. Información y Facturación**

\*Nombre Completo

\*Correo Electrónico

\*Teléfono

Nombre de la Compañía

NIT

\*Calle

\*Código Postal (requiere para el envío)

\*País

\*Ciudad

☒ Enviar a la misma dirección

**2. Método de Envío**


☒ Recoger en Sucursal Insignia  
CL 50 CL 22A 28B 1000046


☐ Recoger en Sucursal Wynd  
Z 10 10 90 2 8

☐ ¿Demando la entrega a domicilio?

Pag. La suma debe ser por internet o efectivo

**3. Método de Pago**

☒  **Visa/Master**

☐  **Transferencia**

Paga con tarjeta de Crédito Visa o MasterCard

**4. Confirmar Orden**

Cant.  Q 21.00 X

Entrega:	Q 0.00
Promoción:	Q 21.00
Tarifa:	Q 21.00

☒ Obtener promociones y descuentos vía correo electrónico

**Número de Tarjeta**

**Nombre de Titular/tituyente**

**Fecha de expiración**

MM / AAAA    VVVV    YY Y

**CVC**

Vinculos

**Pagar ahora**



En el usuario de administrador los pedidos se irán enlistando de la siguiente manera

Administrar pedidos

Usuario admin

Administrar pedidos

Lista de pedidos

No pedido	Pedido	Precio	Fecha	Mostrar detalles del pedido
-----------	--------	--------	-------	-----------------------------

Datos del usuario para factura

Nombre

Apellido

Nit

Telefono

Ubicación de entrega

Forma de pago

Detalles de forma de pago

Imprimir factura

articulos del pedido y total

Martillo

2

Precio 15 Q

Total 30 Q

actualizar estado del pedido

No enviado

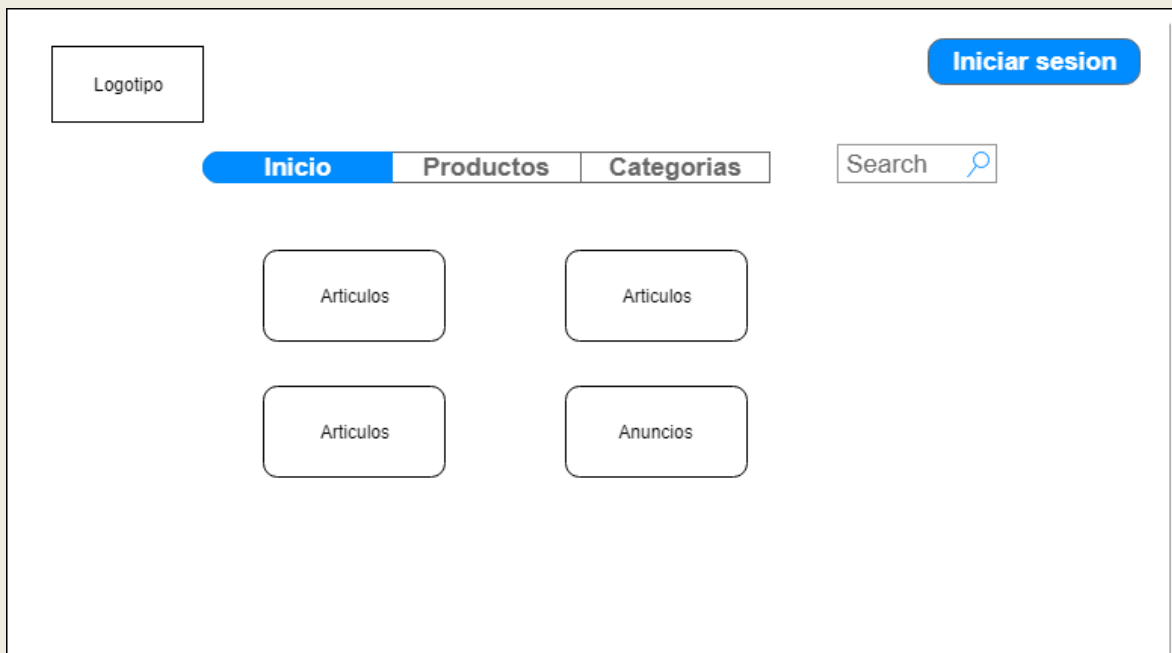
Recibido

Guardar

Cancelar

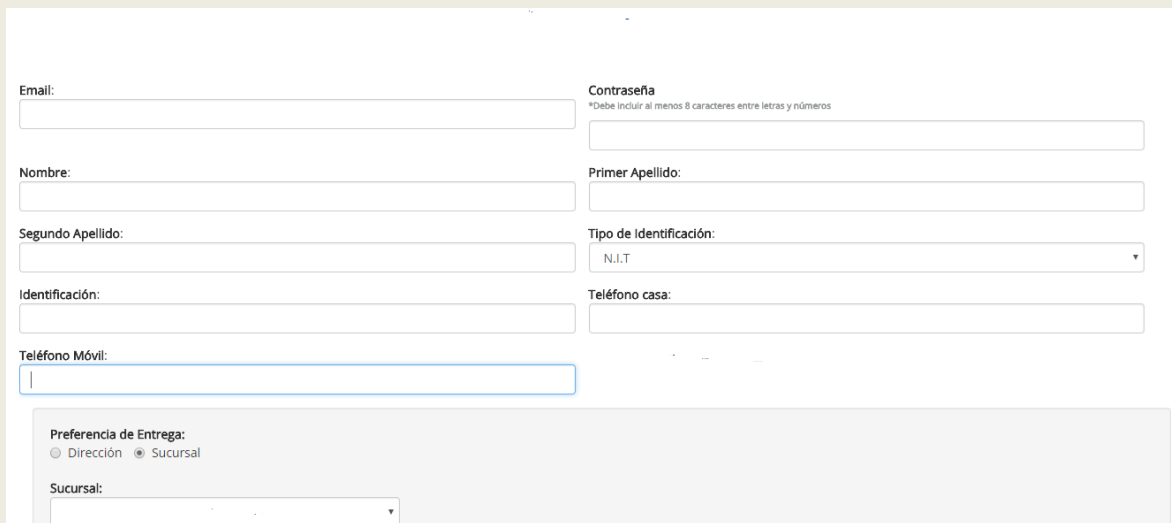
## Screens de diseños

### Pantalla principal



The main screen design features a white background with a blue header bar. On the left, there is a placeholder for a logo labeled 'Logotipo'. On the right, there is a blue button labeled 'Iniciar sesion'. Below the header, there is a navigation bar with three tabs: 'Inicio' (highlighted in blue), 'Productos', and 'Categorias'. To the right of the navigation bar is a search bar with the text 'Search' and a magnifying glass icon. Below the navigation bar, there are four rounded rectangular buttons arranged in a 2x2 grid. The top-left button is labeled 'Articulos', the top-right button is labeled 'Articulos', the bottom-left button is labeled 'Articulos', and the bottom-right button is labeled 'Anuncios'. On the far right, there is a vertical ellipsis menu icon.

### Registrarse



The registration form is a white rectangular area with a light gray border. It contains several input fields and a section for delivery preferences. The fields are arranged in two columns. The left column contains: 'Email:' with a text input field, 'Nombre:' with a text input field, 'Segundo Apellido:' with a text input field, 'Identificación:' with a text input field, and 'Teléfono Móvil:' with a text input field. The right column contains: 'Contraseña' with a text input field and a note '\*Debe incluir al menos 8 caracteres entre letras y números', 'Primer Apellido:' with a text input field, 'Tipo de Identificación:' with a dropdown menu showing 'N.I.T.', and 'Teléfono casa:' with a text input field. At the bottom, there is a section titled 'Preferencia de Entrega:' with two radio buttons: 'Dirección' and 'Sucursal' (selected). Below this is a dropdown menu labeled 'Sucursal:'.

Para realizar pedidos

Pedidos

1

Facturación

\*Nombre Completo

\*Correo Electrónico

\*Teléfono

Nombre de la Empresa

RUT

\*Calle

\*Código Postal (requiere para el envío)

\*País

Argentina

\*Ciudad

☐ Enviar a la misma dirección

2

Método de Envío

☒ Recoger en Sucursal (envío gratis)  
Calle 19, Ciudad Autónoma de Buenos Aires

☐ Recoger en Sucursal (envío gratis)  
2 de Mayo 48

☐ Demorar el envío (seleccionar ciudad)  
País, La Plata (envío gratis)  
seleccionar Sucursal

3

Método de Pago

☒ Mover a Monero

☐ Transferencia

Paga con tarjeta de Crédito  
Visa o MasterCard

4

Confirmar Orden

Cant. 1 Q 21.000

Entrega: Q 0.00

Productos: Q 21.00

Impuestos: Q 21.00

☐ Obtener promociones y descuentos vía correo electrónico

Número de Tarjeta

Tarj. 1234 5678 1234 5678

Nombre de Tarjetahabiente

Mr. Pedro Perez

Fecha de expiración

MM/YY

YY/YY

12%

CVC

123

Monedas

Cardano

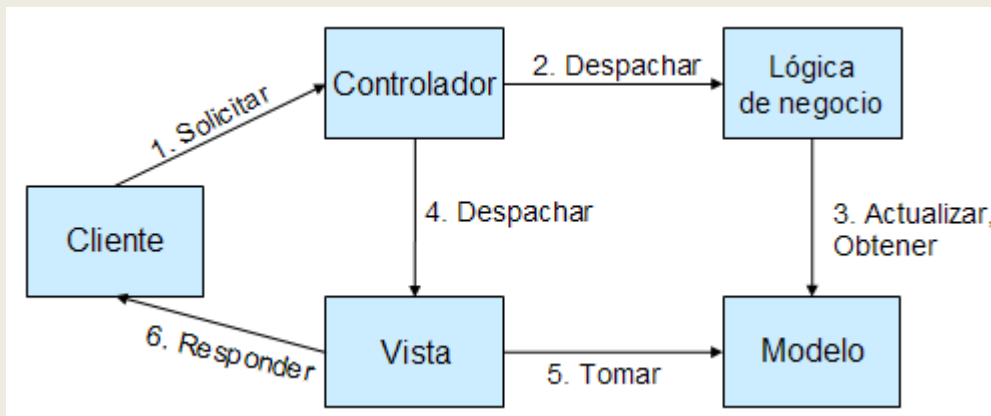
Paga ahora

**Detalle arquitectónico:** Struts es un framework, que nos muestra un estandar para poder trabajar el desarrollo de una aplicación web.

Ciclo de struts cuando procesa una petición HTTP

- El controlador es un servlet de una clase proporcionada por struts. Será necesario configurar la app (xml) para que todas las peticiones se dirijan a este servlet
- El controlador despacha las peticiones para la ejecución de esa acción
- La vista por lo general se implementa mediante un jsp (ayuda a los desarrolladores a realizar paginas web dinamicas)
- El modelo es por cuenta del desarrollador ya que es propio de la capa de negocio y no esta dentro del ámbito de struts

**Diagrama que explica el funcionamiento de struts**



La separación en estas tres capas se realiza de la siguiente manera: La capa de Modelo es aquella que se encarga de la relación de la aplicación con la Base Datos y el diseño de los objetos que va a utilizar la aplicación. Esto va a permitir que en ningún momento, desde la capa de vista o negocio se hagan llamadas directas a la BBDD para conseguir información, sino que siempre pasará por llamadas a estos objetos que son los que se encargan en si mismos de realizar las acciones concretas sobre los datos. En cuanto a la capa de Controlador o Negocio, será donde se defina cómo debe funcionar la aplicación en cuanto a qué cosas se pueden o no hacer, cómo deben ser los flujos de información, etc. Esto se consigue mediante las llamadas Actions, clases que se encargan de encapsular estos elementos, totalmente aislados de cómo se van a mostrar por pantalla o de cómo obtienen los datos reales. Y finalmente, la capa de Vista es aquella que se encarga de mostrar la información necesaria en cada momento según la acción que se esté realizando. En estas vistas, no se realiza ningún tipo de operación, sino que simplemente muestra los datos que internamente está manejando el negocio, sin actuar sobre ellos.

la decisión de tomar este modelo para trabajo es porque se contempló la facilidad a la hora del mantenimiento, pues este modelo esta descrito como uno de los diseños que proporciona más facilidad a la hora de realizar mejoras, añadidos o arreglo en la aplicación y no afectan al resto de componentes del trabajo.

## Diagramas de clases

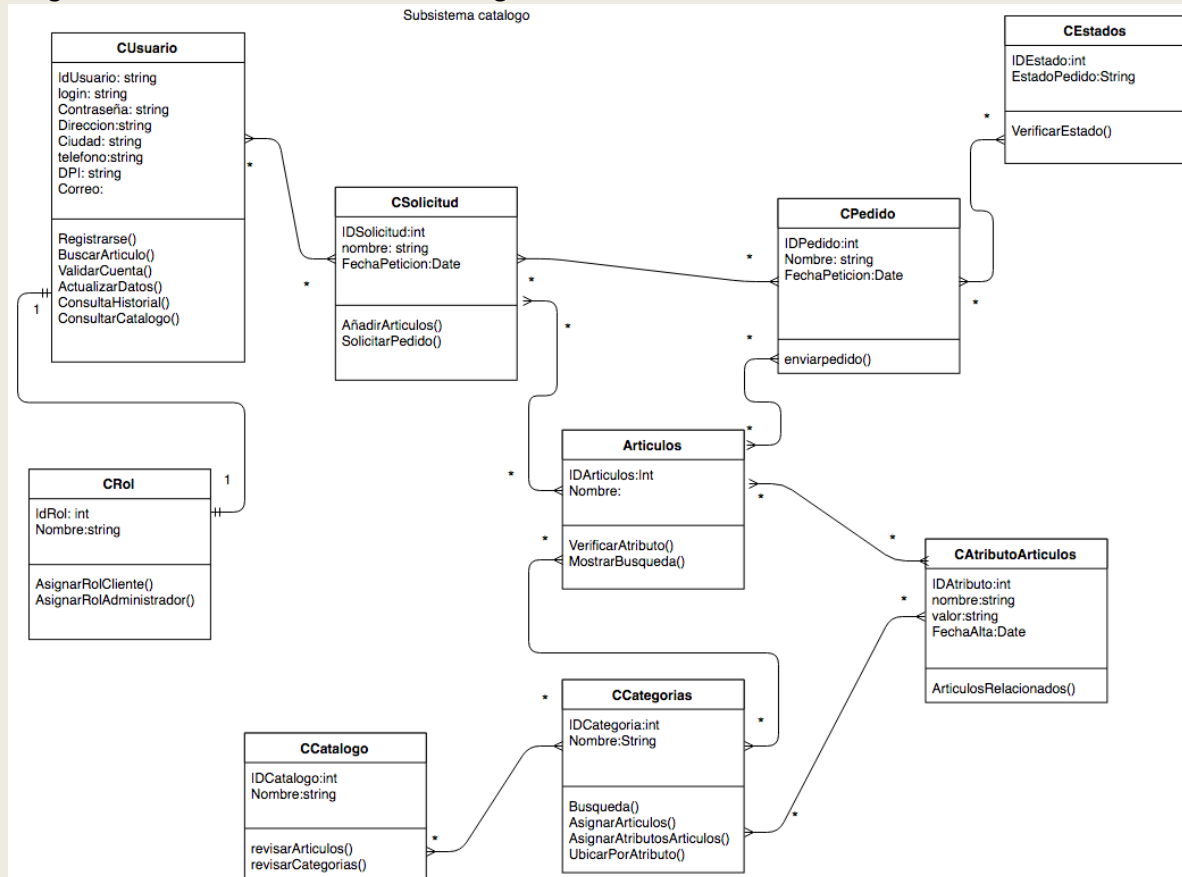
Tendrá 3 subsistemas para poder trabajar con la tienda online, que son el catalogo, administración e informes.

**Catalogo:** en esta sección se detallara y enlistara de una manera general las acciones que los usuarios pueden realizar una vez hayan ingresado en la aplicación.

- Consultar el catálogo de la ferretería
- Consultar el historial de pedidos
- Búsqueda de articulas de manera directa
- Añadir artículos al carrito para
- Alta de usuario para hacer pedidos
- Actualizar los datos de usuario

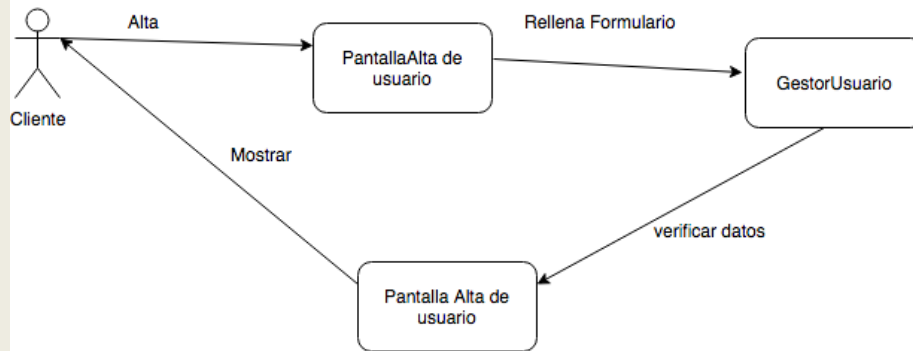
**Nota:** el usuario si no está registrado igual puede acceder a revisar el catalogo, la solicitud es el proceso en donde solicita el pedido, al solicitarlo debe realizar una confirmación después de haber añadido los articulos

### Diagrama de clases subsistema de catálogos

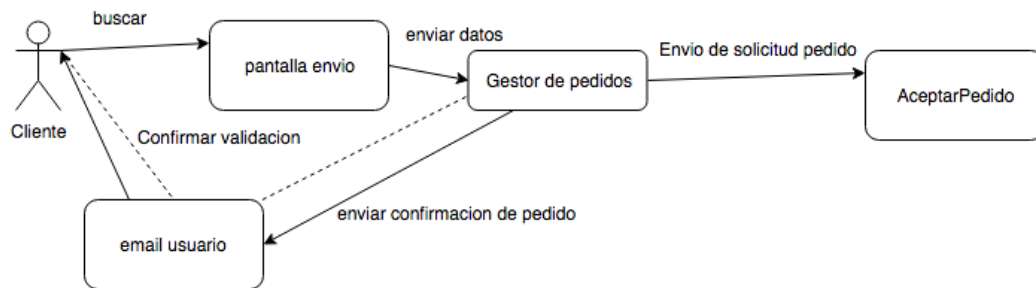


## Diagrama de colaboración

### Alta de usuario



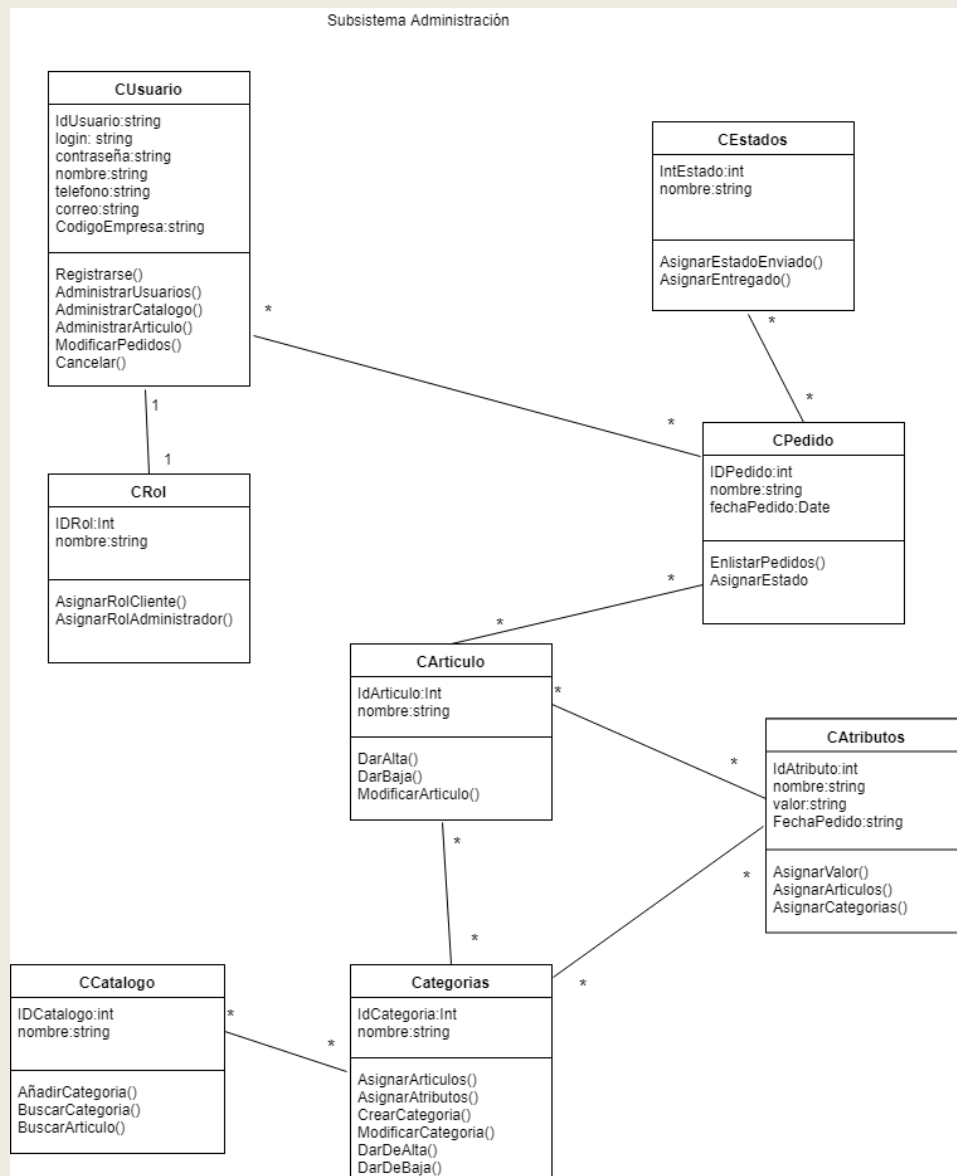
### Pedido



Administración: acciones que los administradores pueden realizar una vez dentro de la aplicación.

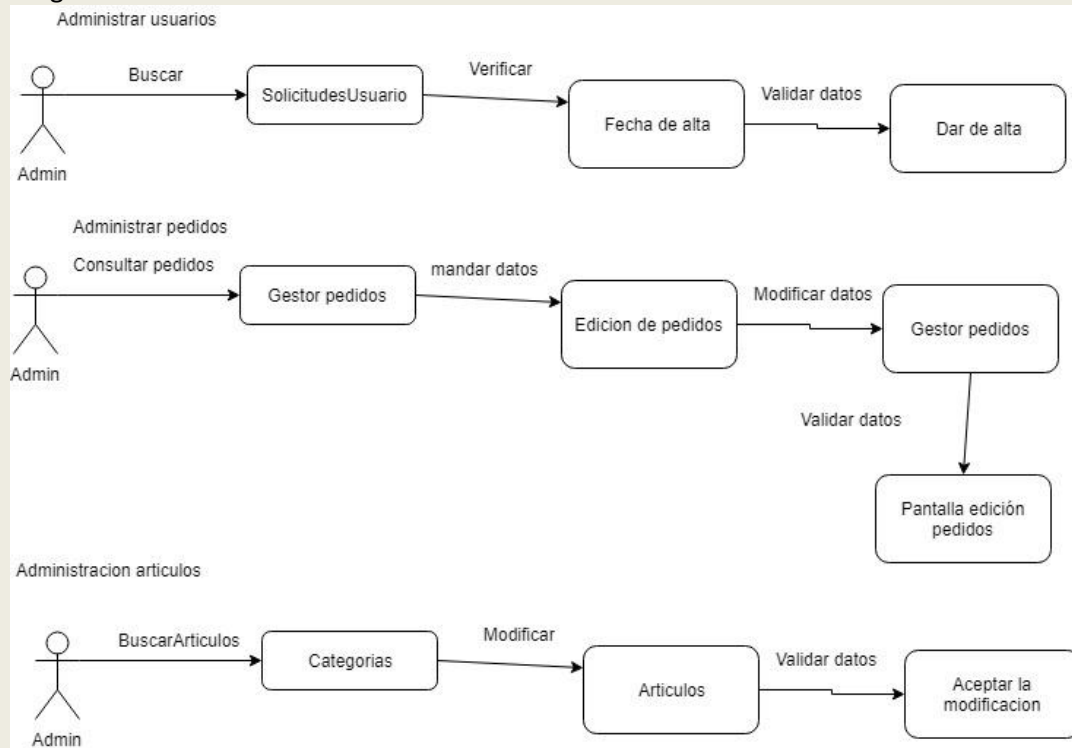
- Alta, baja, modificación usuario.
- ..... categorías
- ..... artículos
- Cancelar y modificar pedidos

**Nota:** tomar en cuenta que solo los administradores pueden acceder a esto y si se desea cancelar un pedido se debe de llamar a la central para modificación o eliminación de dicho pedido





## Diagramas de colaboración

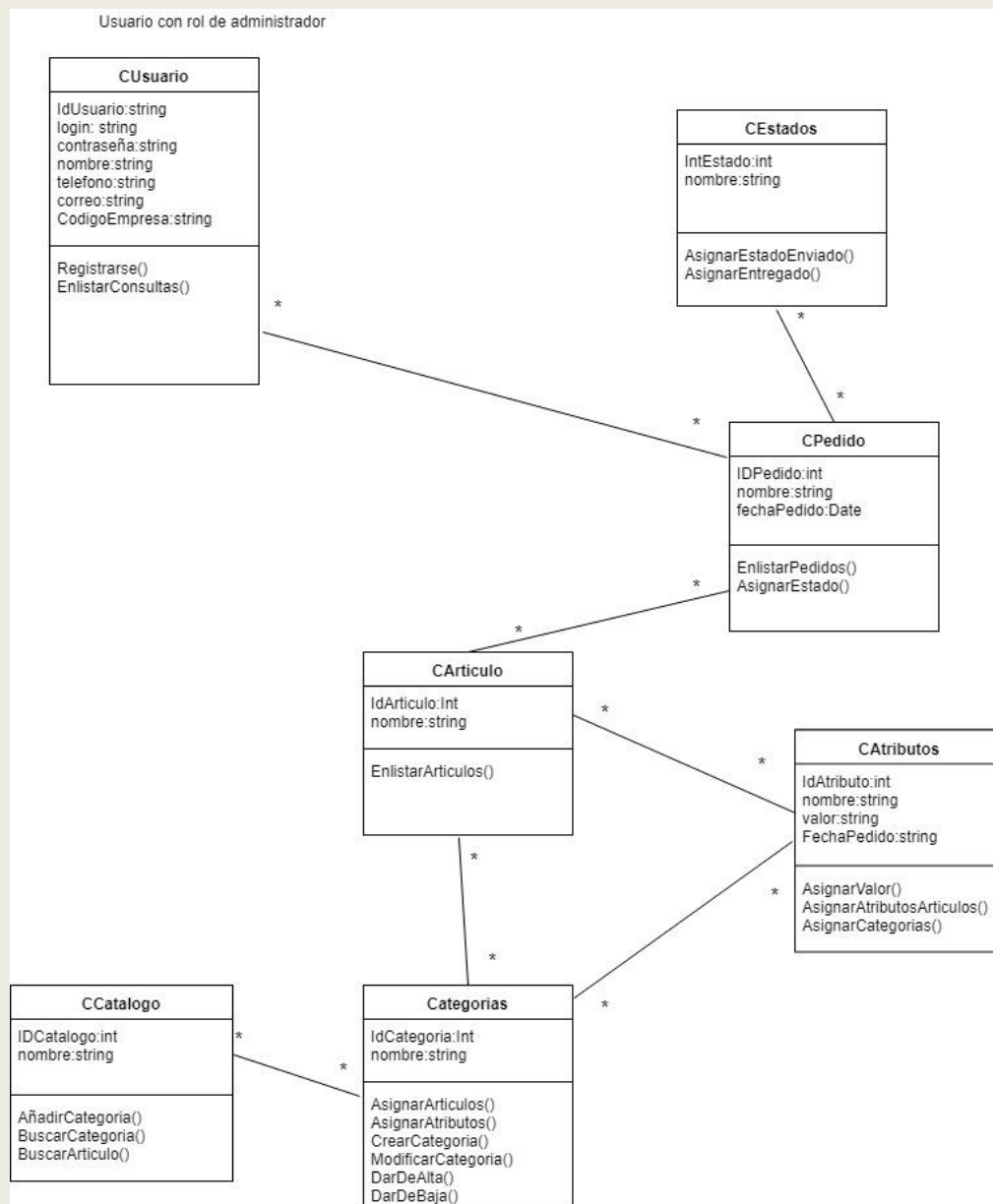


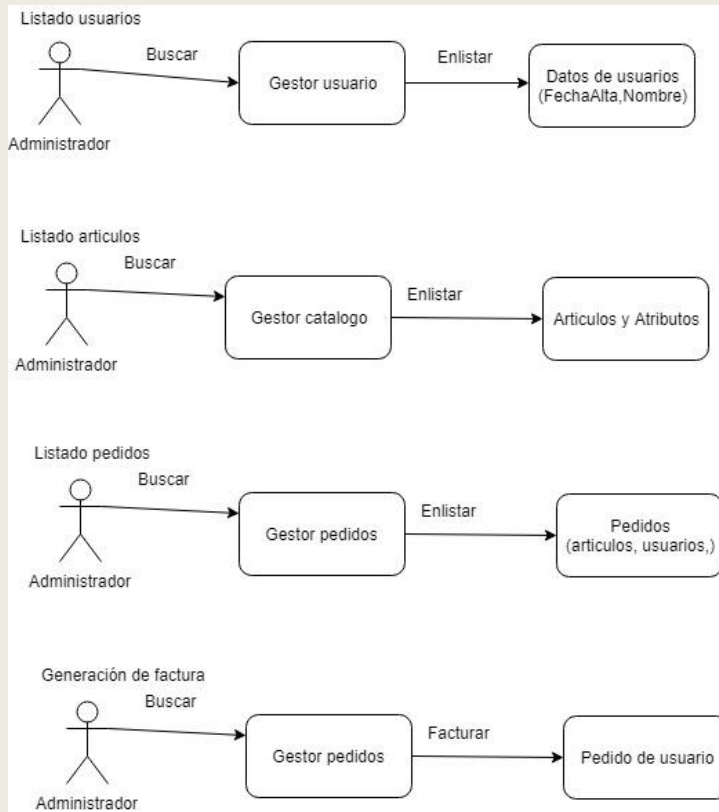
Listados: engloba todo lo que los administradores pueden realizar al momento de querer consultar las estadísticas de la tienda online y poder generar sus facturas

Los listados que serán parte de este subsistema serán:

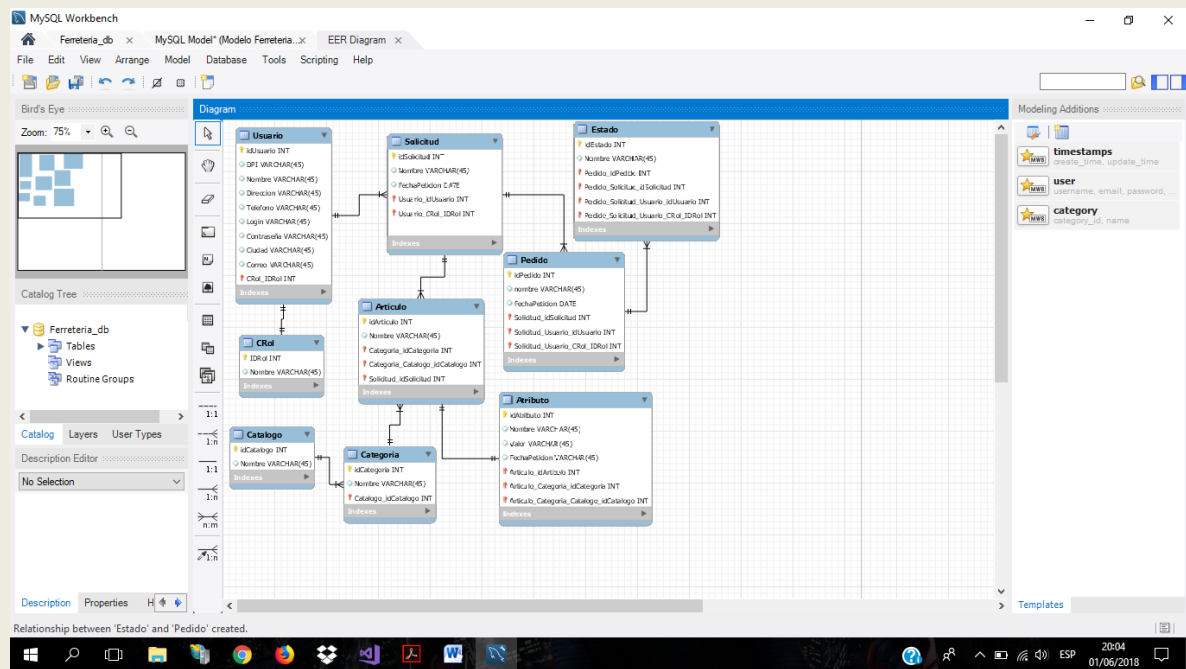
- Listado de usuarios
- Listad de artículos
- Listado de pedidos

**Nota:** todos los administradores podrán ver los listados, esto se aclara para que saber que se esta trabajando con administradores únicamente, no hay roles distintos para otros “tipos de administradores”.





## Diagrama Entidad Relación Cliente



## Script Generado Cliente

-- MySQL Script generated by MySQL Workbench

-- Fri Jun 1 20:14:25 2018

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-----

-- Schema Ferreteria\_db

-- Schema Ferreteria\_db

-----

CREATE SCHEMA IF NOT EXISTS `Ferreteria\_db` DEFAULT CHARACTER SET utf8 ;

USE `Ferreteria\_db` ;

-----

-- Table `Ferreteria\_db`.`CRol`

-----

CREATE TABLE IF NOT EXISTS `Ferreteria\_db`.`CRol` (

`IDRol` INT NOT NULL,

`Nombre` VARCHAR(45) NULL,

PRIMARY KEY (`IDRol`))

ENGINE = InnoDB;

-----

-- Table `Ferreteria\_db`.`Usuario`

-----  
CREATE TABLE IF NOT EXISTS `Ferreteria\_db`.`Usuario` (

  `idUsuario` INT NOT NULL,

  `DPI` VARCHAR(45) NULL,

  `Nombre` VARCHAR(45) NULL,

  `Direccion` VARCHAR(45) NULL,

  `Telefono` VARCHAR(45) NULL,

  `Login` VARCHAR(45) NULL,

  `Contraseña` VARCHAR(45) NULL,

  `Ciudad` VARCHAR(45) NULL,

  `Correo` VARCHAR(45) NULL,

  `CRol\_IDRol` INT NOT NULL,

  PRIMARY KEY (`idUsuario`, `CRol\_IDRol`),

  INDEX `fk\_Usuario\_CRol1\_idx` (`CRol\_IDRol` ASC),

  CONSTRAINT `fk\_Usuario\_CRol1`

    FOREIGN KEY (`CRol\_IDRol`)

      REFERENCES `Ferreteria\_db`.`CRol` (`IDRol`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----  
-- Table `Ferreteria\_db`.`Solicitud`  
-----

CREATE TABLE IF NOT EXISTS `Ferreteria\_db`.`Solicitud` (

  `idSolicitud` INT NOT NULL,

  `Nombre` VARCHAR(45) NULL,

  `FechaPeticon` DATE NULL,

```

`Usuario_idUsuario` INT NOT NULL,

`Usuario_CRol_IDRol` INT NOT NULL,

`Usuario_idUsuario1` INT NOT NULL,

`Usuario_CRol_IDRol1` INT NOT NULL,

PRIMARY KEY (`idSolicitud`, `Usuario_idUsuario`, `Usuario_CRol_IDRol`, `Usuario_idUsuario1`, `Usuario_CRol_IDRol1`),

INDEX `fk_Solicitud_Usuario1_idx` (`Usuario_idUsuario1` ASC, `Usuario_CRol_IDRol1` ASC),

CONSTRAINT `fk_Solicitud_Usuario1`

FOREIGN KEY (`Usuario_idUsuario1`, `Usuario_CRol_IDRol1`)

REFERENCES `Ferreteria_db`.`Usuario` (`idUsuario`, `CRol_IDRol`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `Ferreteria_db`.`Pedido`

-----

CREATE TABLE IF NOT EXISTS `Ferreteria_db`.`Pedido` (

`idPedido` INT NOT NULL,

`nombre` VARCHAR(45) NULL,

`FechaPeticion` DATE NULL,

`Solicitud_idSolicitud` INT NOT NULL,

`Solicitud_Usuario_idUsuario` INT NOT NULL,

`Solicitud_Usuario_CRol_IDRol` INT NOT NULL,

PRIMARY KEY (`idPedido`, `Solicitud_idSolicitud`, `Solicitud_Usuario_idUsuario`, `Solicitud_Usuario_CRol_IDRol`),

INDEX `fk_Pedido_Solicitud1_idx` (`Solicitud_idSolicitud` ASC, `Solicitud_Usuario_idUsuario` ASC,

`Solicitud_Usuario_CRol_IDRol` ASC),

CONSTRAINT `fk_Pedido_Solicitud1`

FOREIGN KEY (`Solicitud_idSolicitud`, `Solicitud_Usuario_idUsuario`, `Solicitud_Usuario_CRol_IDRol`)

REFERENCES `Ferreteria_db`.`Solicitud` (`idSolicitud`, `Usuario_idUsuario`, `Usuario_CRol_IDRol`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

```

```
ENGINE = InnoDB;
```

```
-- Table `Ferreteria_db`.`Estado`
```

```
CREATE TABLE IF NOT EXISTS `Ferreteria_db`.`Estado` (
```

```
  `idEstado` INT NOT NULL,
```

```
  `Nombre` VARCHAR(45) NULL,
```

```
  `Pedido_idPedido` INT NOT NULL,
```

```
  `Pedido_Solicitud_idSolicitud` INT NOT NULL,
```

```
  `Pedido_Solicitud_Usuario_idUsuario` INT NOT NULL,
```

```
  `Pedido_Solicitud_Usuario_CRol_IDRol` INT NOT NULL,
```

```
  PRIMARY KEY (`idEstado`, `Pedido_idPedido`, `Pedido_Solicitud_idSolicitud`, `Pedido_Solicitud_Usuario_idUsuario`,  
  `Pedido_Solicitud_Usuario_CRol_IDRol`),
```

```
  INDEX `fk_Estado_Pedido1_idx` (`Pedido_idPedido` ASC, `Pedido_Solicitud_idSolicitud` ASC,  
  `Pedido_Solicitud_Usuario_idUsuario` ASC, `Pedido_Solicitud_Usuario_CRol_IDRol` ASC),
```

```
  CONSTRAINT `fk_Estado_Pedido1`
```

```
    FOREIGN KEY (`Pedido_idPedido`, `Pedido_Solicitud_idSolicitud`, `Pedido_Solicitud_Usuario_idUsuario`,  
  `Pedido_Solicitud_Usuario_CRol_IDRol`)
```

```
    REFERENCES `Ferreteria_db`.`Pedido` (`idPedido`, `Solicitud_idSolicitud`, `Solicitud_Usuario_idUsuario`,  
  `Solicitud_Usuario_CRol_IDRol`)
```

```
    ON DELETE NO ACTION
```

```
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

```
-- Table `Ferreteria_db`.`Catalogo`
```

```
CREATE TABLE IF NOT EXISTS `Ferreteria_db`.`Catalogo` (
```

```
  `idCatalogo` INT NOT NULL,
```

```
  `Nombre` VARCHAR(45) NULL,
```

```
  PRIMARY KEY (`idCatalogo`))
```



```
ENGINE = InnoDB;
```

```
-- Table `Ferreteria_db`.`Categoria`
```

```
CREATE TABLE IF NOT EXISTS `Ferreteria_db`.`Categoria` (  
  `idCategoria` INT NOT NULL,  
  `Nombre` VARCHAR(45) NULL,  
  `Catalogo_idCatalogo` INT NOT NULL,  
  PRIMARY KEY (`idCategoria`, `Catalogo_idCatalogo`),  
  INDEX `fk_Categoria_Catalogo1_idx` (`Catalogo_idCatalogo` ASC),  
  CONSTRAINT `fk_Categoria_Catalogo1`  
    FOREIGN KEY (`Catalogo_idCatalogo`)  
      REFERENCES `Ferreteria_db`.`Catalogo` (`idCatalogo`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)
```

```
ENGINE = InnoDB;
```

```
-- Table `Ferreteria_db`.`Articulo`
```

```
CREATE TABLE IF NOT EXISTS `Ferreteria_db`.`Articulo` (  
  `idArticulo` INT NOT NULL,  
  `Nombre` VARCHAR(45) NULL,  
  `Categoria_idCategoria` INT NOT NULL,  
  `Categoria_Catalogo_idCatalogo` INT NOT NULL,  
  `Solicitud_idSolicitud` INT NOT NULL,  
  PRIMARY KEY (`idArticulo`, `Categoria_idCategoria`, `Categoria_Catalogo_idCatalogo`, `Solicitud_idSolicitud`),  
  INDEX `fk_Articulo_Categoria1_idx` (`Categoria_idCategoria` ASC, `Categoria_Catalogo_idCatalogo` ASC),  
  INDEX `fk_Articulo_Solicitud1_idx` (`Solicitud_idSolicitud` ASC),  
  CONSTRAINT `fk_Articulo_Categoria1`  
    FOREIGN KEY (`Categoria_idCategoria`, `Categoria_Catalogo_idCatalogo`)
```

```

REFERENCES `Ferreteria_db`.`Categoria` (`idCategoria`, `Catalogo_idCatalogo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk_Articulo_Solicitud1`

FOREIGN KEY (`Solicitud_idSolicitud`)

REFERENCES `Ferreteria_db`.`Solicitud` (`idSolicitud`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `Ferreteria_db`.`Atributo`

-----

CREATE TABLE IF NOT EXISTS `Ferreteria_db`.`Atributo` (

`idAtributo` INT NOT NULL,

`Nombre` VARCHAR(45) NULL,

`Valor` VARCHAR(45) NULL,

`FechaPeticion` VARCHAR(45) NULL,

`Articulo_idArticulo` INT NOT NULL,

`Articulo_Categoria_idCategoria` INT NOT NULL,

`Articulo_Categoria_Catalogo_idCatalogo` INT NOT NULL,

PRIMARY KEY (`idAtributo`, `Articulo_idArticulo`, `Articulo_Categoria_idCategoria`,

`Articulo_Categoria_Catalogo_idCatalogo`),

INDEX `fk_Atributo_Articulo1_idx` (`Articulo_idArticulo` ASC, `Articulo_Categoria_idCategoria` ASC,

`Articulo_Categoria_Catalogo_idCatalogo` ASC),

CONSTRAINT `fk_Atributo_Articulo1`

FOREIGN KEY (`Articulo_idArticulo`, `Articulo_Categoria_idCategoria`, `Articulo_Categoria_Catalogo_idCatalogo`)


REFERENCES `Ferreteria_db`.`Articulo` (`idArticulo`, `Categoria_idCategoria`, `Categoria_Catalogo_idCatalogo`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

```

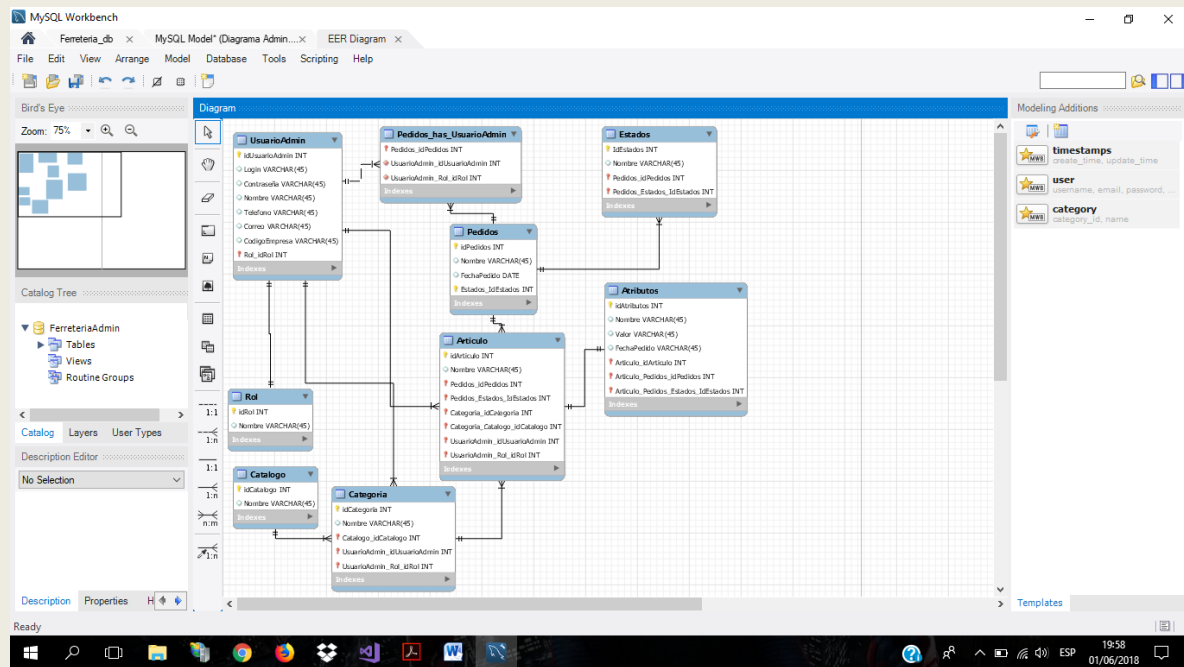


```
SET SQL_MODE=@OLD_SQL_MODE;
```

```
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
```

```
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## Diagrama Entidad Relacion Administrador



### Script Generado Administrador

-- MySQL Script generated by MySQL Workbench

-- Fri Jun 1 19:54:15 2018

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- Schema FerreteriaAdmin

-----

-----

-- Schema FerreteriaAdmin

-----

CREATE SCHEMA IF NOT EXISTS `FerreteriaAdmin` DEFAULT CHARACTER SET utf8 ;

USE `FerreteriaAdmin` ;

-- Table `FerreteriaAdmin`.`Rol`

-----

CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Rol` (

    `idRol` INT NOT NULL,

    `Nombre` VARCHAR(45) NULL,

    PRIMARY KEY (`idRol`))

ENGINE = InnoDB;

-----

-- Table `FerreteriaAdmin`.`UsuarioAdmin`

-----

CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`UsuarioAdmin` (

  `idUsuarioAdmin` INT NOT NULL,

  `Login` VARCHAR(45) NULL,

  `Contraseña` VARCHAR(45) NULL,

  `Nombre` VARCHAR(45) NULL,

  `Telefono` VARCHAR(45) NULL,

  `Correo` VARCHAR(45) NULL,

  `CodigoEmpresa` VARCHAR(45) NULL,

  `Rol\_idRol` INT NOT NULL,

  PRIMARY KEY (`idUsuarioAdmin`, `Rol\_idRol`),

  INDEX `fk\_UsuarioAdmin\_Rol\_idx` (`Rol\_idRol` ASC),

  CONSTRAINT `fk\_UsuarioAdmin\_Rol`

    FOREIGN KEY (`Rol\_idRol`)

      REFERENCES `FerreteriaAdmin`.`Rol` (`idRol`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----

-- Table `FerreteriaAdmin`.`Pedidos`

-----

CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Pedidos` (

  `idPedidos` INT NOT NULL,

  `Nombre` VARCHAR(45) NULL,

```

`FechaPedido` DATE NULL,

`Estados_IdEstados` INT NOT NULL,

PRIMARY KEY (`idPedidos`, `Estados_IdEstados`))

ENGINE = InnoDB;

-- Table `FerreteriaAdmin`.`Estados`
-----

CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Estados` (

`IdEstados` INT NOT NULL,

`Nombre` VARCHAR(45) NULL,

`Pedidos_idPedidos` INT NOT NULL,

`Pedidos_Estados_IdEstados` INT NOT NULL,

PRIMARY KEY (`IdEstados`, `Pedidos_idPedidos`, `Pedidos_Estados_IdEstados`),

INDEX `fk_Estados_Pedidos1_idx` (`Pedidos_idPedidos` ASC, `Pedidos_Estados_IdEstados` ASC),

CONSTRAINT `fk_Estados_Pedidos1`

FOREIGN KEY (`Pedidos_idPedidos`, `Pedidos_Estados_IdEstados`)

REFERENCES `FerreteriaAdmin`.`Pedidos` (`idPedidos`, `Estados_IdEstados`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-----

-- Table `FerreteriaAdmin`.`Catalogo`
-----

CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Catalogo` (

`idCatalogo` INT NOT NULL,

`Nombre` VARCHAR(45) NULL,

PRIMARY KEY (`idCatalogo`))

ENGINE = InnoDB;

```

-----  
-- Table `FerreteriaAdmin`.`Categoria`  
-----

```
CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Categoria` (  
  `idCategoria` INT NOT NULL,  
  `Nombre` VARCHAR(45) NULL,  
  `Catalogo_idCatalogo` INT NOT NULL,  
  `UsuarioAdmin_idUsuarioAdmin` INT NOT NULL,  
  `UsuarioAdmin_Rol_idRol` INT NOT NULL,  
  PRIMARY KEY (`idCategoria`, `Catalogo_idCatalogo`, `UsuarioAdmin_idUsuarioAdmin`,  
  `UsuarioAdmin_Rol_idRol`),  
  INDEX `fk_Categoria_Catalogo1_idx` (`Catalogo_idCatalogo` ASC),  
  INDEX `fk_Categoria_UsuarioAdmin1_idx` (`UsuarioAdmin_idUsuarioAdmin` ASC,  
  `UsuarioAdmin_Rol_idRol` ASC),  
  CONSTRAINT `fk_Categoria_Catalogo1`  
    FOREIGN KEY (`Catalogo_idCatalogo`)  
    REFERENCES `FerreteriaAdmin`.`Catalogo` (`idCatalogo`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT `fk_Categoria_UsuarioAdmin1`  
    FOREIGN KEY (`UsuarioAdmin_idUsuarioAdmin`, `UsuarioAdmin_Rol_idRol`)  
    REFERENCES `FerreteriaAdmin`.`UsuarioAdmin` (`idUsuarioAdmin`, `Rol_idRol`)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

-- Table `FerreteriaAdmin`.`Articulo`  
-----



```

CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Articulo` (

  `idArticulo` INT NOT NULL,

  `Nombre` VARCHAR(45) NULL,

  `Pedidos_idPedidos` INT NOT NULL,

  `Pedidos_Estados_IdEstados` INT NOT NULL,

  `Categoria_idCategoria` INT NOT NULL,

  `Categoria_Catalogo_idCatalogo` INT NOT NULL,

  `UsuarioAdmin_idUsuarioAdmin` INT NOT NULL,

  `UsuarioAdmin_Rol_idRol` INT NOT NULL,

  PRIMARY KEY (`idArticulo`, `Pedidos_idPedidos`, `Pedidos_Estados_IdEstados`, `Categoria_idCategoria`,
  `Categoria_Catalogo_idCatalogo`, `UsuarioAdmin_idUsuarioAdmin`, `UsuarioAdmin_Rol_idRol`),

  INDEX `fk_Articulo_Pedidos1_idx` (`Pedidos_idPedidos` ASC, `Pedidos_Estados_IdEstados` ASC),

  INDEX `fk_Articulo_Categoria1_idx` (`Categoria_idCategoria` ASC, `Categoria_Catalogo_idCatalogo` ASC),

  INDEX `fk_Articulo_UsuarioAdmin1_idx` (`UsuarioAdmin_idUsuarioAdmin` ASC, `UsuarioAdmin_Rol_idRol`
  ASC),

  CONSTRAINT `fk_Articulo_Pedidos1`

    FOREIGN KEY (`Pedidos_idPedidos`, `Pedidos_Estados_IdEstados`)

    REFERENCES `FerreteriaAdmin`.`Pedidos` (`idPedidos`, `Estados_IdEstados`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `fk_Articulo_Categoria1`

    FOREIGN KEY (`Categoria_idCategoria`, `Categoria_Catalogo_idCatalogo`)

    REFERENCES `FerreteriaAdmin`.`Categoria` (`idCategoria`, `Catalogo_idCatalogo`)

    ON DELETE NO ACTION

    ON UPDATE NO ACTION,

  CONSTRAINT `fk_Articulo_UsuarioAdmin1`

    FOREIGN KEY (`UsuarioAdmin_idUsuarioAdmin`, `UsuarioAdmin_Rol_idRol`)

    REFERENCES `FerreteriaAdmin`.`UsuarioAdmin` (`idUsuarioAdmin`, `Rol_idRol`)

```

```

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `FerreteriaAdmin`.`Atributos`
-----

CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Atributos` (

  `idAtributos` INT NOT NULL,

  `Nombre` VARCHAR(45) NULL,

  `Valor` VARCHAR(45) NULL,

  `FechaPedido` VARCHAR(45) NULL,

  `Articulo_idArticulo` INT NOT NULL,

  `Articulo_Pedidos_idPedidos` INT NOT NULL,

  `Articulo_Pedidos_Estados_IdEstados` INT NOT NULL,

  PRIMARY KEY (`idAtributos`, `Articulo_idArticulo`, `Articulo_Pedidos_idPedidos`,
  `Articulo_Pedidos_Estados_IdEstados`),

  INDEX `fk_Atributos_Articulo1_idx` (`Articulo_idArticulo` ASC, `Articulo_Pedidos_idPedidos` ASC,
  `Articulo_Pedidos_Estados_IdEstados` ASC),

  CONSTRAINT `fk_Atributos_Articulo1`

  FOREIGN KEY (`Articulo_idArticulo`, `Articulo_Pedidos_idPedidos`,
  `Articulo_Pedidos_Estados_IdEstados`)

  REFERENCES `FerreteriaAdmin`.`Articulo` (`idArticulo`, `Pedidos_idPedidos`,
  `Pedidos_Estados_IdEstados`)

  ON DELETE NO ACTION

  ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- Table `FerreteriaAdmin`.`Pedidos_has_UsuarioAdmin`
-----

```

```
CREATE TABLE IF NOT EXISTS `FerreteriaAdmin`.`Pedidos_has_UsuarioAdmin` (  
  
  `Pedidos_idPedidos` INT NOT NULL,  
  
  `UsuarioAdmin_idUsuarioAdmin` INT NOT NULL,  
  
  `UsuarioAdmin_Rol_idRol` INT NOT NULL,  
  
  PRIMARY KEY (`Pedidos_idPedidos`),  
  
  INDEX `fk_Pedidos_has_UsuarioAdmin_UsuarioAdmin1_idx` (`UsuarioAdmin_idUsuarioAdmin` ASC,  
  `UsuarioAdmin_Rol_idRol` ASC),  
  
  INDEX `fk_Pedidos_has_UsuarioAdmin_Pedidos1_idx` (`Pedidos_idPedidos` ASC),  
  
  CONSTRAINT `fk_Pedidos_has_UsuarioAdmin_Pedidos1`  
  
    FOREIGN KEY (`Pedidos_idPedidos`)  
  
    REFERENCES `FerreteriaAdmin`.`Pedidos` (`idPedidos`)  
  
    ON DELETE NO ACTION  
  
    ON UPDATE NO ACTION,  
  
  CONSTRAINT `fk_Pedidos_has_UsuarioAdmin_UsuarioAdmin1`  
  
    FOREIGN KEY (`UsuarioAdmin_idUsuarioAdmin`, `UsuarioAdmin_Rol_idRol`)  
  
    REFERENCES `FerreteriaAdmin`.`UsuarioAdmin` (`idUsuarioAdmin`, `Rol_idRol`)  
  
    ON DELETE NO ACTION  
  
    ON UPDATE NO ACTION)  
  
ENGINE = InnoDB;  
  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

### **Fase Implementación**

Se requiere de las siguientes aplicaciones para comenzar con la fase que se menciona en esta sección

- Framework de desarrollo: struts que fue el que mencionamos durante el desarrollo que sería con el que se iba a diseñar la aplicación siguiendo el modelo de 3 capas (Modelo, vista, controlador)
- Gestor de base de datos MySQL
- Creacion de directorios para la estructura web
- TOMCAT contenedor de servlets
- JBoss
- Máquina virtual Java

Con los elementos anteriores que fueron descritos se comienza la implementación, donde se empieza a desarrollar plantillas, interfaces, ficheros de generación de reportes, librerías de código y clases de la capa de negocio de struts.

El software se entregara en un disco y se entregara una copia extra del mismo.

### **Fase Despliegue**

Una vez ya creada la aplicación se procede a realizar las pruebas para verificar el funcionamiento de la misma, la base de datos ya específica es MySQL, y se desarrolla en la máquina virtual, dentro del entorno del sistema operativo, utilizando el navegador para acceder a la aplicación web y ver las pruebas del entorno del funcionamiento.

## **Fase De Mantenimiento**

### **Plan de Mantenimiento**

#### **1) Definición**

Definición del convenio: la definición de cómo estará estipulado el convenio en el acuerdo del contrato, el mantenimiento se llevara a cabo con nuestra empresa y no con terceros, en este acuerdo también se especifica el tipo de soporte que se brindara una vez entregado el producto y puesto en marcha, se especifica también y se hace constar con el cliente, que se brindara soporte técnico en línea, el mantenimiento con el que se iniciara la fase es mantenimiento correctivo y una breve capacitación que constara de 3 días para instruir al personal que será responsable de manejar el software.

#### **2) Tipo de mantenimiento que se brindara**

El mantenimiento que se brindara es un **mantenimiento preventivo**, este es el primer mantenimiento que se estará realizando, una vez entregado el producto y puesto en marcha, se estará revisando constantes revisiones para ver el estado del producto, en busca de posibles fallas que puedan ocurrir y analizar todo una vez ya en trabajo.

#### **3) Designación de encargado de mantenimiento**

Durante el convenio estipulado con el cliente se designó a un miembro de nuestro equipo de trabajo para llevar a cabo el mantenimiento, puesto que no ocurra ningún suceso o anomalía extraña con el producto solamente se designara a uno y conforme avance a otro tipo de mantenimiento se designaran otras personas que le acompañen. Para el mantenimiento preventivo mediante revisiones constantes se designó como principal encargado del mantenimiento a **José Miranda**

#### 4) Definición de costos

En esta sección se detallara la estimación de costos:

Recursos a utilizar para la capacitación: recursos de la empresa, transporte y movilización, tiempo requerido que constara de 3 días

El costo estimado en la disponibilidad de recursos y el tiempo, para empezar se estima un costo aproximado de 2500 para la capacitación que se realizara durante tres días una vez ya entregado el producto para empezar a realizar la misma capacitación, utilizando recursos de nuestra empresa para poder presentar el funcionamiento del producto.

Posteriormente para el mantenimiento preventivo solamente se cobrara el tiempo estipulado para el mantenimiento preventivo que sería aproximadamente 2700 por revisiones que se harán 2 por semana durante 1 mes y 10 días.

**Nota a tomar en cuenta:** si durante la fase de mantenimiento ocurre algún fallo por alguna anomalía ajena a código o fallo de aplicación, como por ejemplo daño del equipo de la empresa o daño al software causado indirectamente por un empleado y se nos solicita la verificación de dicho problema se cobrara por el tiempo que tome montar nuevamente la aplicación.

## **Detalle específico del plan de mantenimiento**

### **Porque de este mantenimiento**

El plan de mantenimiento que solicito el cliente, es solamente para la revisión constante durante un tiempo establecido, para verificar que esté funcionando correctamente y si acaso se presenta un problema o se detecta algo que se podría corregir, además de la capacitación a los empleados.

Más adelante el cliente podría solicitar otro tipo de mantenimiento en cualquier otro de los casos que se presenten, ya sea adaptativo, correctivo o perfectivo.

### **Responsables de realizar este trabajo:**

Como se especificó en el convenio el designado para la realización de dicho trabajo es **Jose Miranda** y si más adelante se requiere otro mantenimiento se designaran más personas.

### **Responsables e involucrados**

- Jose Miranda designado del mantenimiento y capacitación
- Empleados para la capacitación
- Cliente con el que se acordó

**Nota:** la capacitación constara de 3 dias y el mantenimiento preventivo se desarrollara 2 días por semana durante 1 mes y 10 días

Tareas	Actores
Capacitación primer día: exposición y uso básico	Jose Miranda, Empleados y cliente
Capacitación segundo día: procedimientos de uso y detalle avanzado del funcionamiento del sistema	Jose Miranda, Empleados y cliente
Capacitación tercer día: para el tercer día se realizara practica individual y evaluación para saber si comprenden el manejo del sistema	Jose Miranda, Empleados y cliente
Mantenimiento preventivo martes y jueves	Jose Miranda, Empleados
Mantenimiento preventivo martes y jueves	Jose Miranda, Empleados
Mantenimiento preventivo martes y jueves	Jose Miranda, Empleados
Mantenimiento preventivo martes y jueves	Jose Miranda, Empleados
Mantenimiento preventivo martes y jueves	Jose Miranda, Empleados
Mantenimiento preventivo martes y jueves	Jose Miranda, Empleados
Mantenimiento preventivo martes y jueves	Jose Miranda, Empleados

#### **Recursos disponibles:**

- material de la empresa (proyector, presentación, hojas de evaluación, usb, material de enseñanza)
- transporte
- recurso humano
- equipo de la empresa a la que se brinda mantenimiento
- esquema del reporte de avances
- lugar de realización: empresa



**Esquema de reporte de avances entrega a cliente**

Nombre del designado\_\_\_\_\_

ID software\_\_\_\_\_

Nombre de institución\_\_\_\_\_

Numero reporte\_\_\_\_\_

Fecha\_\_\_\_\_

Detalles del estado del sistema:

Comentario\_\_\_\_\_

\_\_\_\_\_  
Firma del designado

\_\_\_\_\_  
Firma del cliente

### **Fechas de mantenimiento**

Actividad	Fechas designadas
Revisión del trabajo del software durante jornada laboral	Martes 1 y Jueves 3 de mayo
Revisión del trabajo del software durante jornada laboral	Martes 8 y Jueves 10 de mayo
Revisión del trabajo del software durante jornada laboral	Martes 15 y Jueves 17 de mayo
Revisión del trabajo del software durante jornada laboral	Martes 22 y Jueves 24 de mayo
Revisión del trabajo del software durante jornada laboral	Martes 29 y Jueves 31 de mayo
Revisión del trabajo del software durante jornada laboral	Martes 5 y Jueves 7 de junio
Revisión del trabajo del software durante jornada laboral	Martes 12 y Jueves 14 de junio

### **Fechas de capacitación**

26 27 y 28 de abril 2018
--------------------------

### **Detalles del soporte técnico**

El soporte técnico se dará en línea, se enviara a la dirección de correo de nuestra empresa e inmediatamente se acudirá, para los fallos se deberá de enviar en un formato específico proporcionado para enviar las anomalías, después del periodo de finalización de soporte existe posibilidad de renovación de soporte y que se decida hacer un cambio, modificación enfocada a un mantenimiento especifico. La firma del cliente constara de que el soporte se realizó, se deben de imprimir 2 hojas una se queda en la empresa y la otra se la lleva el cliente

### Documento de soporte técnico

Nombre del software\_\_\_\_\_

ID del producto\_\_\_\_\_

Nombre de la empresa\_\_\_\_\_

Nombre del designado del soporte\_\_\_\_\_

Descripción de lo que solicita:

Fecha de solicitud\_\_\_\_\_

\_\_\_\_\_  
Firma del designado

\_\_\_\_\_  
Firma del Encargado de la empresa