

# CAPÍTULO 1

## 1. MANUAL TÉCNICO

### 1.1 Introducción

El patrón de diseño aplicado a este proyecto es MVC (Model View Controller).

**Modelo Vista Controlador (MVC)** es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML, el control es el código que provee de datos dinámicos a la página, y el modelo contiene clases representativas de la aplicación (como el mensaje de un foro, un miembro registrado, etc.).

## 1.2 Componentes MVC: Model-View-Controller

**Modelo:** Ésta es la representación específica del dominio de la información sobre la cual funciona la aplicación. El modelo es otra forma de llamar a la capa de dominio. La lógica de dominio añade significado a los datos; por ejemplo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos.

**Vista:** Éste presenta el modelo en un formato adecuado para interactuar, usualmente un elemento de interfaz de usuario.

**Controlador:** Éste responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Muchas aplicaciones utilizan un mecanismo de almacenamiento persistente (como puede ser una base de datos) para almacenar los datos. MVC no menciona específicamente esta capa de acceso a datos.

Es común pensar que una aplicación tiene tres capas principales: presentación (IU), dominio, y acceso a datos. En MVC, la capa de presentación está partida en controlador y vista. La principal separación es entre presentación y dominio; la separación entre V/C es menos clara.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)
- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los

interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.

- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

Entre los frameworks que se utilizan actualmente tenemos a los siguientes:

## Frameworks MVC

Lenguaje	Licencia	Nombre
Java / J2ee	Apache	Struts
Java / J2ee	Apache	Spring Framework
Java / J2ee	Apache	Aurora
Perl	GPL	Catalyst
Ruby	MIT	Ruby on Rails
PHP	MIT	cakePHP
.NET	Apache	CastleProject/MonoRail

Figura 1.1 Frameworks

### **1.3 Struts Framework**

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE (Java 2, Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts permite reducir el tiempo de desarrollo. Su carácter de software libre y su compatibilidad con todas las plataformas en que Java Enterprise esté disponible, lo convierte en una herramienta altamente disponible.

## **1.4 Estructura**

Cuando se programan aplicaciones Web con el patrón MVC, siempre surge la duda de usar un solo controlador o usar varios controladores, pues si consideramos mejor usar un solo controlador para tener toda nuestra lógica en un mismo lugar, nos encontramos con un grave problema, ya que nuestro controlador se convierte en lo que se conoce como "fat controller", es decir un controlador saturado de peticiones, Struts surge como la solución a este problema ya que implementa un solo controlador (ActionServlet) que evalúa las peticiones del usuario mediante un archivo configurable (struts-config.xml).

### **Con JSP's y Servlets**

El uso de "MVC" en ambientes Web para JSP's y Servlets ha empezado a generar gran interés, debido a que una vez diseñada una aplicación para ambiente Web es raro que ésta permanezca sin cambios, el uso de MVC permite realizar diseños con JSP's/Servlets que logran verdaderas soluciones a escala.

Tomemos el simple caso de un proceso de registro que después de varios meses requiere ser modificado para solicitar datos no contemplados en el diseño inicial, o bien, la modificación de un proceso de trabajo ("Work-Flow") que requiere modificar el orden de solicitud de datos; cualquiera de los casos anteriores requiere forzosamente modificación al código de JSP's y/o Servlets, observe la siguiente gráfica

que demuestra los posibles resultados de una aplicación con JSP's/Servlets :

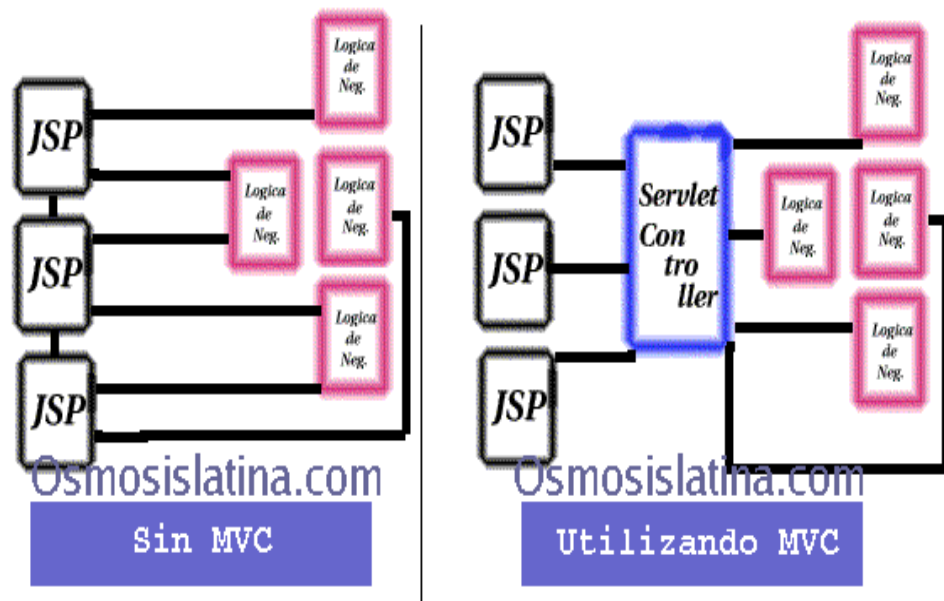


Figura 1.2 JSP's/Servlets

### Java Beans

Un Java Bean es una manera de modularizar el uso de datos en una aplicación con JSP's/Servlets a través de una *Clase*, su característica primordial es el uso de los métodos *get* y *set* los cuales permiten el acceso a los valores del Bean, el diseño de un Java Bean es relativamente sencillo ya que no posee código extenso, observe el siguiente Java Bean que es utilizado para manipular datos de usuarios.

```
public class Usuario implements java.io.Serializable
{
    private String nombre;
    private String apellido;
    private String correo;
    public Usuario() {}
    public Usuario(String nombre, String apellido, String
correo)
    { this.nombre = nombre;
      this.apellido = apellido;
      this.correo = correo;}
    public void setNombre(String nombre) {
    this.nombre = nombre;
    }
    public String getNombre()
    { return nombre;}
    public void setApellido(String apellido) {
    this.apellido = apellido;}
    public String getApellido() {
    return apellido;}
    public void setCorreo(String correo) {
    this.correo = correo;}
    public String getCorreo(){
    return correo;}
}
```

**Figura 1.3 Java Beans**



## 1.5 Características de Struts.

Entre las características de Struts se pueden mencionar:

- Configuración del control centralizada.
- Interrelaciones entre Acciones y página u otras acciones se especifican por tablas XML en lugar de codificarlas en los programas o páginas.
- Componentes de aplicación, que son el mecanismo para compartir información bidireccionalmente entre el usuario de la aplicación y las acciones del modelo.
- Librerías de entidades para facilitar la mayoría de las operaciones que generalmente realizan las páginas JSP.
- Struts contiene herramientas para validación de campos de plantillas bajo varios esquemas que van desde validaciones locales en la página (en javaScript) hasta las validaciones de fondo hechas a nivel de las acciones.
- Struts permite que el desarrollador se concentre en el diseño de aplicaciones complejas como una serie simple de componentes del Modelo y de la vista intercomunicados por un control centralizado. Diseñando de esta manera se debe obtener una aplicación más consistente y más fácil de mantener.

## 1.6 Diagrama de Clases

A continuacion la estructura de un diagrama de clases.

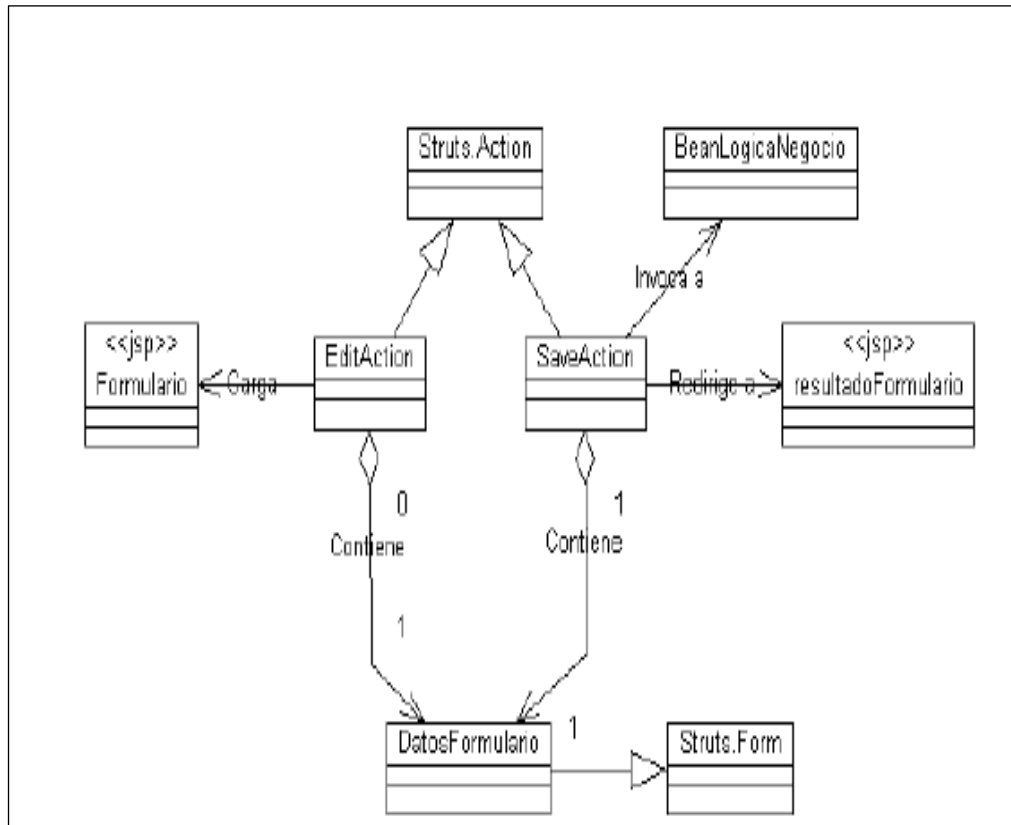


Figura 1.4 Diagrama de clases

## 1.7 Diagrama de Colaboración.

Interacción entre clases y struts.

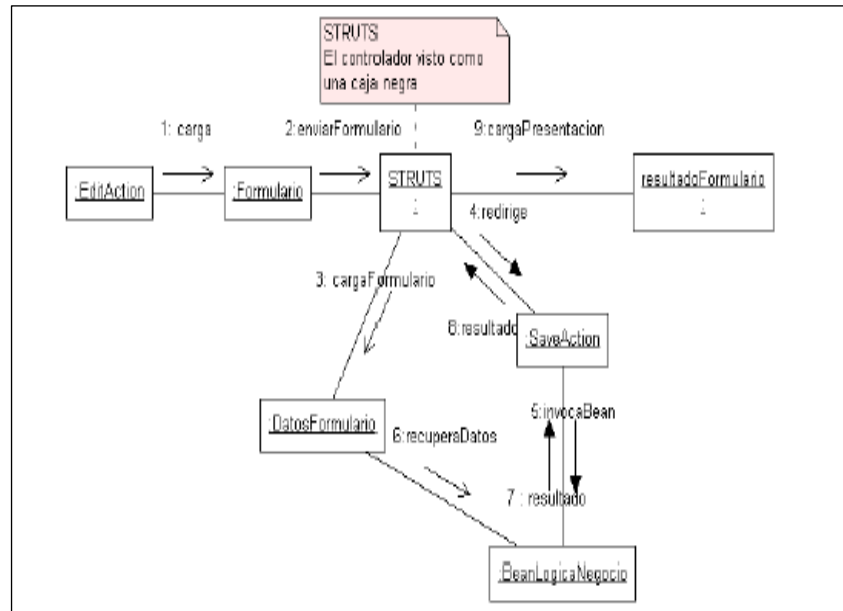
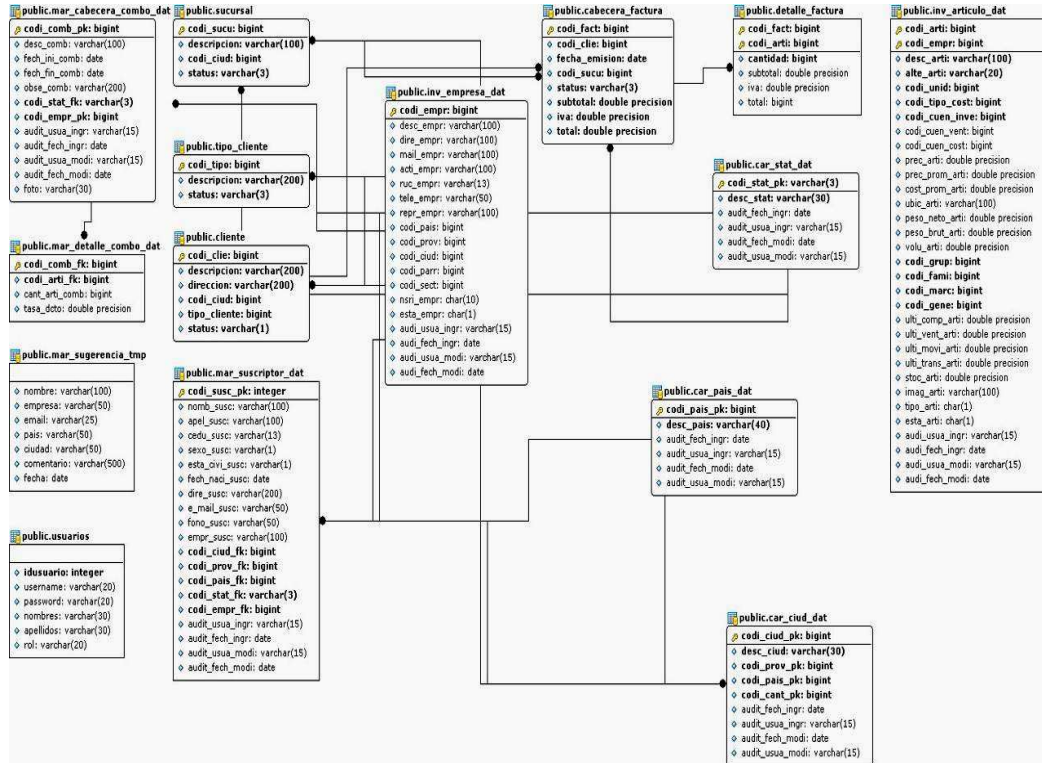


Figura 1.5 Diagrama de colaboración

## 1.8 Diagrama Objeto Relación-Módulo de Marketing



## 1.9 SCRIPT PARA CREACION DE LA BASE DE DATOS

```
-- SQL Manager 2005 for PostgreSQL 3.6.0.1
-- -----
-- Host      : localhost
-- Database  : erp_marketing
-- Version:   : PostgreSQL 8.0.0beta4

DROP TABLE mar_detalle_combo_dat;
DROP TABLE mar_cabecera_combo_dat;
DROP TABLE usuarios;
DROP TABLE detalle_factura;
DROP TABLE cabecera_factura;
DROP TABLE sucursal;
DROP TABLE tipo_cliente;
DROP TABLE cliente;
DROP TABLE usuariolor;
DROP TABLE usuario;
DROP TABLE rol;
DROP TABLE rh_tipocontrato_dat;
DROP TABLE rh_salario_dat;
DROP TABLE rh_nivelmando_dat;
DROP TABLE rh_empleado_dat;
DROP TABLE rh_departamento_dat;
DROP TABLE rh_cargo_dat;
DROP TABLE rh_area_dat;
DROP TABLE rh_agencia_dat;
DROP TABLE portal;
DROP TABLE oper_cdla_dat;
DROP TABLE oper_calle_dat;
DROP TABLE mar_suscriptor_dat;
DROP TABLE mar_sugerencia_tmp;
DROP TABLE inv_unidad_dat;
DROP TABLE inv_marca_dat;
DROP TABLE inv_grupo_dat;
DROP TABLE inv_genero_dat;
DROP TABLE inv_familia_dat;
DROP TABLE inv_empresa_dat;
DROP TABLE inv_articulo_dat;
DROP TABLE car_tipo_pago_dat;
DROP TABLE car_tipo_clie_dat;
DROP TABLE car_stat_dat;
DROP TABLE car_prov_dat;
DROP TABLE car_pais_dat;
DROP TABLE car_ciud_dat;
DROP TABLE car_cant_dat;
DROP TABLE banco;
```

```
CREATE TABLE banco (  
    idbanco bigint NOT NULL,  
    nombre varchar(100) NOT NULL,  
    idestado bigint NOT NULL,  
    idusuario bigint NOT NULL,  
    fecharegistro varchar(10) NOT NULL,  
    horaregistro varchar(8) NOT NULL  
);  
  
CREATE TABLE car_cant_dat (  
    codi_cant_pk bigint NOT NULL,  
    desc_cant varchar(30) NOT NULL,  
    codi_prov_pk bigint NOT NULL,  
    audit_fech_ingr date,  
    audit_usua_ingr varchar(15),  
    audit_fech_modi date,  
    audit_usua_modi varchar(15)  
);  
  
CREATE TABLE car_ciud_dat (  
    codi_ciud_pk bigint NOT NULL,  
    desc_ciud varchar(30) NOT NULL,  
    codi_prov_pk bigint NOT NULL,  
    codi_pais_pk bigint NOT NULL,  
    codi_cant_pk bigint NOT NULL,  
    audit_fech_ingr date,  
    audit_usua_ingr varchar(15),  
    audit_fech_modi date,  
    audit_usua_modi varchar(15)  
);  
  
CREATE TABLE car_pais_dat (  
    codi_pais_pk bigint NOT NULL,  
    desc_pais varchar(40) NOT NULL,  
    audit_fech_ingr date,  
    audit_usua_ingr varchar(15),  
    audit_fech_modi date,  
    audit_usua_modi varchar(15)  
);  
  
CREATE TABLE car_prov_dat (  
    codi_prov_pk bigint NOT NULL,  
    desc_prov varchar(30) NOT NULL,  
    codi_pais_pk bigint NOT NULL,  
    audit_fech_ingr date,  
    audit_usua_ingr varchar(15),  
    audit_fech_modi date,  
    audit_usua_modi varchar(15)  
);
```

```
CREATE TABLE car_stat_dat (
  codi_stat_pk varchar(3) NOT NULL,
  desc_stat varchar(30) NOT NULL,
  audit_fech_ingr date,
  audit_usua_ingr varchar(15),
  audit_fech_modi date,
  audit_usua_modi varchar(15)
);

CREATE TABLE car_tipo_clie_dat (
  codi_tipo_clie_pk bigint NOT NULL,
  desc_tipo_clie varchar(30),
  audit_fech_ingr date,
  audit_usua_ingr varchar(15),
  audit_fech_modi date,
  audit_usua_modi varchar(15)
);

CREATE TABLE car_tipo_pago_dat (
  codi_tipo_pago_pk bigint NOT NULL,
  desc_tipo_pago varchar(30),
  audit_fech_ingr date,
  audit_usua_ingr varchar(15),
  audit_fech_modi date,
  audit_usua_modi varchar(15)
);

CREATE TABLE inv_articulo_dat (
  codi_arti bigint NOT NULL,
  desc_arti varchar(100) NOT NULL,
  alte_arti varchar(20) NOT NULL,
  codi_unid bigint NOT NULL,
  codi_tipo_cost bigint NOT NULL,
  codi_cuen_inve bigint NOT NULL,
  codi_cuen_vent bigint,
  codi_cuen_cost bigint,
  prec_arti double precision,
  prec_prom_arti double precision,
  cost_prom_arti double precision,
  ubic_arti varchar(100),
  peso_netto_arti double precision,
  peso_brut_arti double precision,
  volu_arti double precision,
  codi_grup bigint NOT NULL,
  codi_fami bigint NOT NULL,
  codi_marc bigint NOT NULL,
  codi_gene bigint NOT NULL,
  ulti_comp_arti double precision,
  ulti_vent_arti double precision,
  ulti_movi_arti double precision,
  ulti_trans_arti double precision,
  stoc_arti double precision,
  imag_arti varchar(100),
  tipo_arti char(1),
  codi_empr bigint NOT NULL,
  esta_arti char(1),
  audi_usua_ingr varchar(15),
  audi_fech_ingr date,
  audi_usua_modi varchar(15),
  audi_fech_modi date
);
```

```
CREATE TABLE inv_empresa_dat (  
    codi_empr bigint NOT NULL,  
    desc_empr varchar(100),  
    dire_empr varchar(100),  
    mail_empr varchar(100),  
    acti_empr varchar(100),  
    ruc_empr varchar(13),  
    tele_empr varchar(50),  
    repr_empr varchar(100),  
    codi_pais bigint,  
    codi_prov bigint,  
    codi_ciud bigint,  
    codi_parr bigint,  
    codi_sect bigint,  
    nsri_empr char(10),  
    esta_empr char(1),  
    audi_usua_ingr varchar(15),  
    audi_fech_ingr date,  
    audi_usua_modi varchar(15),  
    audi_fech_modi date  
);  
  
CREATE TABLE inv_familia_dat (  
    codi_fami bigint NOT NULL,  
    codi_grup bigint NOT NULL,  
    desc_fami varchar(100),  
    codi_empr bigint NOT NULL,  
    esta_fami char(1),  
    audi_usua_ingr varchar(15),  
    audi_fech_ingr date,  
    audi_usua_modi varchar(15),  
    audi_fech_modi date  
);  
  
CREATE TABLE inv_genero_dat (  
    codi_gene bigint NOT NULL,  
    desc_gene varchar(100),  
    codi_empr bigint NOT NULL,  
    esta_gene char(1),  
    audi_usua_ingr varchar(15),  
    audi_fech_ingr date,  
    audi_usua_modi varchar(15),  
    audi_fech_modi date  
);  
  
CREATE TABLE inv_grupo_dat (  
    codi_grup bigint NOT NULL,  
    desc_grup varchar(100),  
    codi_empr bigint NOT NULL,  
    esta_grup char(1),  
    audi_usua_ingr varchar(15),  
    audi_fech_ingr date,  
    audi_usua_modi varchar(15),  
    audi_fech_modi date  
);
```



```
CREATE TABLE inv_marca_dat (  
    codi_grup bigint NOT NULL,  
    codi_fami bigint NOT NULL,  
    codi_marc bigint NOT NULL,  
    codi_empr bigint NOT NULL,  
    desc_marc varchar(100),  
    esta_marc char(1),  
    audi_usua_ingr varchar(15),  
    audi_fech_ingr date,  
    audi_usua_modi varchar(15),  
    audi_fech_modi date  
);  
  
CREATE TABLE inv_unidad_dat (  
    codi_unid bigint NOT NULL,  
    desc_unid varchar(100),  
    codi_empr bigint NOT NULL,  
    esta_unid char(1),  
    audi_usua_ingr varchar(15),  
    audi_fech_ingr date,  
    audi_usua_modi varchar(15),  
    audi_fech_modi date  
);  
  
CREATE TABLE mar_sugerencia_tmp (  
    nombre varchar(100),  
    empresa varchar(50),  
    email varchar(25),  
    pais varchar(50),  
    ciudad varchar(50),  
    comentario varchar(500),  
    fecha date  
);  
  
CREATE TABLE mar_suscriptor_dat (  
    codi_susc_pk serial NOT NULL,  
    nomb_susc varchar(100),  
    apel_susc varchar(100),  
    cedu_susc varchar(13),  
    sexo_susc varchar(1),  
    esta_civi_susc varchar(1),  
    fech_naci_susc date,  
    dire_susc varchar(200),  
    e_mail_susc varchar(50),  
    fono_susc varchar(50),  
    empr_susc varchar(100),  
    codi_ciud_fk bigint NOT NULL,  
    codi_prov_fk bigint NOT NULL,  
    codi_pais_fk bigint NOT NULL,  
    codi_stat_fk varchar(3) NOT NULL,  
    codi_empr_fk bigint NOT NULL,  
    audit_usua_ingr varchar(15),  
    audit_fech_ingr date,  
    audit_usua_modi varchar(15),  
    audit_fech_modi date  
);
```

```
CREATE TABLE oper_calle_dat (
    codi_calle_pk bigint NOT NULL,
    desc_calle varchar(30) DEFAULT ''::character varying NOT NULL,
    sent_calle varchar(30) NOT NULL,
    codi_ciud_fk bigint NOT NULL,
    estd_reg_calle char(1) DEFAULT ''::bpchar NOT NULL
);

CREATE TABLE oper_cdla_dat (
    codi_cdla_pk bigint NOT NULL,
    desc_cdla varchar(50) DEFAULT ''::character varying NOT NULL,
    codi_ciud_fk bigint NOT NULL,
    estd_reg_cdla char(1) DEFAULT ''::bpchar NOT NULL
);

CREATE TABLE portal (
    idportal bigint NOT NULL,
    nombre varchar(100) NOT NULL,
    ruta varchar(200)[] NOT NULL
);

CREATE TABLE rh_agencia_dat (
    idagencia serial NOT NULL,
    idciudad integer NOT NULL,
    descripcion varchar(30) DEFAULT ''::character varying NOT NULL,
    direccion varchar(50) DEFAULT ''::character varying NOT NULL,
    estado varchar(1) DEFAULT 'A'::character varying
);

CREATE TABLE rh_area_dat (
    idarea serial NOT NULL,
    descripcion varchar(30) DEFAULT ''::character varying NOT NULL,
    estado varchar(1) DEFAULT 'A'::character varying
);

CREATE TABLE rh_cargo_dat (
    idcargo serial NOT NULL,
    idsalario integer NOT NULL,
    idnivelmando integer NOT NULL,
    iddepartamento integer NOT NULL,
    descripcion varchar(30) DEFAULT ''::character varying NOT NULL,
    estado varchar(1) DEFAULT 'A'::character varying
);

CREATE TABLE rh_departamento_dat (
    iddepartamento serial NOT NULL,
    idarea integer NOT NULL,
    descripcion varchar(30) DEFAULT ''::character varying NOT NULL,
    estado varchar(1) DEFAULT 'A'::character varying
);
```

```
CREATE TABLE rh_empleado_dat (
    idempleado serial NOT NULL,
    idcargo integer,
    idagencia integer,
    idtipocontrato integer,
    idlicencia integer,
    idsuperior integer,
    idusuario integer,
    idbanco bigint,
    nombre varchar(50),
    apellido varchar(50),
    cedula varchar(10) NOT NULL,
    sexo varchar(1) NOT NULL,
    nacionalidad varchar(50),
    domicilio varchar(150) NOT NULL,
    telefono varchar(10),
    celular varchar(10),
    mail varchar(50),
    fechaingreso date NOT NULL,
    fechasalida date,
    fechanacimiento date,
    segurosocial varchar(30),
    pago varchar(1),
    numerocuenta varchar(50),
    tipocuenta varchar(1),
    estado varchar(1)
);

CREATE TABLE rh_nivelmando_dat (
    idnivelmando serial NOT NULL,
    descripcion varchar(30) DEFAULT ''::character varying NOT NULL,
    estado varchar(1) DEFAULT 'A'::character varying
);

CREATE TABLE rh_salario_dat (
    idsalario serial NOT NULL,
    descripcion varchar(30) DEFAULT ''::character varying NOT NULL,
    valor double precision,
    estado varchar(1) DEFAULT 'A'::character varying
);

CREATE TABLE rh_tipocontrato_dat (
    idtipocontrato serial NOT NULL,
    descripcion varchar(30) DEFAULT ''::character varying NOT NULL
);

CREATE TABLE rol (
    idrol bigint NOT NULL,
    nombre varchar(100) NOT NULL,
    idportal bigint NOT NULL
);

CREATE TABLE usuario (
    idusuario bigint NOT NULL,
    idsucursal bigint NOT NULL,
    nombre varchar(100) NOT NULL,
    clave varchar(100) NOT NULL,
    email varchar(100) NOT NULL
);
```

```
CREATE TABLE usuariorol (  
    idusuario bigint NOT NULL,  
    idrol bigint NOT NULL  
);  
  
CREATE TABLE cliente (  
    codi_clie bigint NOT NULL,  
    descripcion varchar(200) NOT NULL,  
    direccion varchar(200) NOT NULL,  
    codi_ciud bigint NOT NULL,  
    tipo_cliente bigint NOT NULL,  
    status varchar(1) NOT NULL  
) WITHOUT OIDS;  
  
CREATE TABLE tipo_cliente (  
    codi_tipo bigint NOT NULL,  
    descripcion varchar(200) NOT NULL,  
    status varchar(3) NOT NULL  
) WITHOUT OIDS;  
  
CREATE TABLE sucursal (  
    codi_sucu bigint NOT NULL,  
    descripcion varchar(100) NOT NULL,  
    codi_ciud bigint NOT NULL,  
    status varchar(3) NOT NULL  
) WITHOUT OIDS;  
  
CREATE TABLE cabecera_factura (  
    codi_fact bigint NOT NULL,  
    codi_clie bigint NOT NULL,  
    fecha_emision date NOT NULL,  
    codi_sucu bigint NOT NULL,  
    status varchar(3) NOT NULL,  
    subtotal double precision NOT NULL,  
    iva double precision NOT NULL,  
    total double precision NOT NULL  
) WITHOUT OIDS;  
  
CREATE TABLE detalle_factura (  
    codi_fact bigint NOT NULL,  
    codi_arti bigint NOT NULL,  
    cantidad bigint NOT NULL,  
    subtotal double precision,  
    iva double precision,  
    total bigint  
) WITHOUT OIDS;  
CREATE TABLE usuarios (  
    idusuario integer NOT NULL,  
    username varchar(20),  
    "password" varchar(20),  
    nombres varchar(30),  
    apellidos varchar(30),  
    rol varchar(20)  
);
```

```
CREATE TABLE mar_cabecera_combo_dat (  
    codi_comb_pk bigint NOT NULL,  
    desc_comb varchar(100),  
    fech_ini_comb date,  
    fech_fin_comb date,  
    obse_comb varchar(200),  
    codi_stat_fk varchar(3) NOT NULL,  
    codi_empr_pk bigint NOT NULL,  
    audit_usua_ingr varchar(15),  
    audit_fech_ingr date,  
    audit_usua_modi varchar(15),  
    audit_fech_modi date,  
    foto varchar(30)  
);  
  
CREATE TABLE mar_detalle_combo_dat (  
    codi_comb_fk bigint NOT NULL,  
    codi_arti_fk bigint NOT NULL,  
    cant_arti_comb bigint,  
    tasa_dcto double precision  
);  
  
ALTER TABLE ONLY inv_articulo_dat  
    ADD CONSTRAINT articulo_pk PRIMARY KEY (codi_arti, codi_empr);  
  
ALTER TABLE ONLY banco  
    ADD CONSTRAINT banco_pkey PRIMARY KEY (idbanco);  
  
ALTER TABLE ONLY oper_calle_dat  
    ADD CONSTRAINT calle_pk PRIMARY KEY (codi_calle_pk);  
  
ALTER TABLE ONLY car_cant_dat  
    ADD CONSTRAINT car_cant_dat_pkey PRIMARY KEY (codi_cant_pk);  
  
ALTER TABLE ONLY car_ciud_dat  
    ADD CONSTRAINT car_ciud_dat_pkey PRIMARY KEY (codi_ciud_pk);  
  
ALTER TABLE ONLY car_pais_dat  
    ADD CONSTRAINT car_pais_dat_pkey PRIMARY KEY (codi_pais_pk);  
  
ALTER TABLE ONLY car_prov_dat  
    ADD CONSTRAINT car_prov_dat_pkey PRIMARY KEY (codi_prov_pk);  
  
ALTER TABLE ONLY car_stat_dat  
    ADD CONSTRAINT car_stat_dat_pkey PRIMARY KEY (codi_stat_pk);  
  
ALTER TABLE ONLY car_tipo_clie_dat  
    ADD CONSTRAINT car_tipo_clie_dat_pkey PRIMARY KEY (codi_tipo_clie_pk);  
  
ALTER TABLE ONLY car_tipo_pago_dat  
    ADD CONSTRAINT car_tipo_pago_dat_pkey PRIMARY KEY (codi_tipo_pago_pk);
```

```
ALTER TABLE ONLY oper_cdla_dat
    ADD CONSTRAINT cdla_pk PRIMARY KEY (codi_cdla_pk);

ALTER TABLE ONLY inv_empresa_dat
    ADD CONSTRAINT empresa_pk PRIMARY KEY (codi_empr);

ALTER TABLE ONLY inv_familia_dat
    ADD CONSTRAINT familia_pk PRIMARY KEY (codi_grup, codi_fami, codi_empr);

ALTER TABLE ONLY inv_genero_dat
    ADD CONSTRAINT genero_pk PRIMARY KEY (codi_gene, codi_empr);

ALTER TABLE ONLY inv_grupo_dat
    ADD CONSTRAINT grupo_pk PRIMARY KEY (codi_grup, codi_empr);

ALTER TABLE ONLY mar_suscriptor_dat
    ADD CONSTRAINT mar_suscriptor_dat_pkey PRIMARY KEY (codi_susc_pk);

ALTER TABLE ONLY inv_marca_dat
    ADD CONSTRAINT marca_pk PRIMARY KEY (codi_grup, codi_fami, codi_marc,
codi_empr);

ALTER TABLE ONLY portal
    ADD CONSTRAINT portal_pkey PRIMARY KEY (idportal);

ALTER TABLE ONLY rh_agencia_dat
    ADD CONSTRAINT rh_agencia_dat_pkey PRIMARY KEY (idagencia);

ALTER TABLE ONLY rh_area_dat
    ADD CONSTRAINT rh_area_dat_pkey PRIMARY KEY (idarea);

ALTER TABLE ONLY rh_cargo_dat
    ADD CONSTRAINT rh_cargo_dat_pkey PRIMARY KEY (idcargo);
ALTER TABLE ONLY rh_departamento_dat
    ADD CONSTRAINT rh_departamento_dat_pkey PRIMARY KEY (iddepartamento);

ALTER TABLE ONLY rh_empleado_dat
    ADD CONSTRAINT rh_empleado_dat_pkey PRIMARY KEY (idempleado);

ALTER TABLE ONLY rh_nivelmando_dat
    ADD CONSTRAINT rh_nivelmando_dat_pkey PRIMARY KEY (idnivelmando);

ALTER TABLE ONLY rh_salario_dat
    ADD CONSTRAINT rh_salario_dat_pkey PRIMARY KEY (idsalario);
ALTER TABLE ONLY rh_tipocontrato_dat
    ADD CONSTRAINT rh_tipocontrato_dat_pkey PRIMARY KEY (idtipocontrato);

ALTER TABLE ONLY rol
    ADD CONSTRAINT rol_pkey PRIMARY KEY (idrol);
```

```
ALTER TABLE ONLY inv_unidad_dat
    ADD CONSTRAINT unidad_pk PRIMARY KEY (codi_unid, codi_empr);

ALTER TABLE ONLY usuario
    ADD CONSTRAINT usuario_pkey PRIMARY KEY (idusuario);

ALTER TABLE ONLY mar_suscriptor_dat
    ADD CONSTRAINT "CODI_CIUd_FK" FOREIGN KEY (codi_ciud_fk) REFERENCES
car_ciud_dat(codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY mar_suscriptor_dat
    ADD CONSTRAINT "CODI_EMPR_FK" FOREIGN KEY (codi_empr_fk) REFERENCES
inv_empresa_dat(codi_empr) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY mar_suscriptor_dat
    ADD CONSTRAINT "CODI_PAIS_FK" FOREIGN KEY (codi_pais_fk) REFERENCES
car_pais_dat(codi_pais_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY mar_suscriptor_dat
    ADD CONSTRAINT "CODI_PROV_FK" FOREIGN KEY (codi_prov_fk) REFERENCES
car_prov_dat(codi_prov_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY mar_suscriptor_dat
    ADD CONSTRAINT "CODI_STAT_FK" FOREIGN KEY (codi_stat_fk) REFERENCES
car_stat_dat(codi_stat_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY banco
    ADD CONSTRAINT banco_idusuario_fkey FOREIGN KEY (idusuario) REFERENCES
usuario(idusuario) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY car_cant_dat
    ADD CONSTRAINT car_cant_dat_codi_prov_pk_fkey FOREIGN KEY (codi_prov_pk)
REFERENCES car_prov_dat(codi_prov_pk);

ALTER TABLE ONLY car_ciud_dat
    ADD CONSTRAINT car_ciud_dat_codi_cant_pk_fkey FOREIGN KEY (codi_cant_pk)
REFERENCES car_cant_dat(codi_cant_pk);

ALTER TABLE ONLY car_ciud_dat
    ADD CONSTRAINT car_ciud_dat_codi_pais_pk_fkey FOREIGN KEY (codi_pais_pk)
REFERENCES car_pais_dat(codi_pais_pk);

ALTER TABLE ONLY car_ciud_dat
    ADD CONSTRAINT car_ciud_dat_codi_prov_pk_fkey FOREIGN KEY (codi_prov_pk)
REFERENCES car_prov_dat(codi_prov_pk);
ALTER TABLE ONLY car_prov_dat
    ADD CONSTRAINT car_prov_dat_codi_pais_pk_fkey FOREIGN KEY (codi_pais_pk)
REFERENCES car_pais_dat(codi_pais_pk);

ALTER TABLE ONLY oper_calle_dat
    ADD CONSTRAINT ciud_calle_fk FOREIGN KEY (codi_ciud_fk) REFERENCES
car_ciud_dat(codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;
```

```
ALTER TABLE ONLY oper_cdla_dat
  ADD CONSTRAINT ciud_cdla_fk FOREIGN KEY (codi_ciud_fk) REFERENCES
  car_ciud_dat(codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY inv_articulo_dat
  ADD CONSTRAINT genero_fk FOREIGN KEY (codi_gene, codi_empr) REFERENCES
  inv_genero_dat(codi_gene, codi_empr) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY inv_familia_dat
  ADD CONSTRAINT grupo_fk FOREIGN KEY (codi_grup, codi_empr) REFERENCES
  inv_grupo_dat(codi_grup, codi_empr) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY inv_articulo_dat
  ADD CONSTRAINT marca_fk FOREIGN KEY (codi_grup, codi_fami, codi_marc,
  codi_empr) REFERENCES inv_marca_dat(codi_grup, codi_fami, codi_marc, codi_empr) ON
  UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY rh_agencia_dat
  ADD CONSTRAINT rh_agencia_dat_idciudad_fkey FOREIGN KEY (idciudad) REFERENCES
  car_ciud_dat(codi_ciud_pk);

ALTER TABLE ONLY rh_cargo_dat
  ADD CONSTRAINT rh_cargo_dat_iddepartamento_fkey FOREIGN KEY (iddepartamento)
  REFERENCES rh_departamento_dat(iddepartamento);

ALTER TABLE ONLY rh_cargo_dat
  ADD CONSTRAINT rh_cargo_dat_idnivelmando_fkey FOREIGN KEY (idnivelmando)
  REFERENCES rh_nivelmando_dat(idnivelmando);

ALTER TABLE ONLY rh_cargo_dat
  ADD CONSTRAINT rh_cargo_dat_idsalario_fkey FOREIGN KEY (idsalario) REFERENCES
  rh_salario_dat(idsalario);

ALTER TABLE ONLY rh_departamento_dat
  ADD CONSTRAINT rh_departamento_dat_idarea_fkey FOREIGN KEY (idarea) REFERENCES
  rh_area_dat(idarea);
ALTER TABLE ONLY rh_empleado_dat
  ADD CONSTRAINT rh_empleado_dat_idagencia_fkey FOREIGN KEY (idagencia)
  REFERENCES rh_agencia_dat(idagencia);

ALTER TABLE ONLY rh_empleado_dat
  ADD CONSTRAINT rh_empleado_dat_idbanco_fkey FOREIGN KEY (idbanco) REFERENCES
  banco(idbanco);

ALTER TABLE ONLY rh_empleado_dat
  ADD CONSTRAINT rh_empleado_dat_idcargo_fkey FOREIGN KEY (idcargo) REFERENCES
  rh_cargo_dat(idcargo);

ALTER TABLE ONLY rh_empleado_dat
  ADD CONSTRAINT rh_empleado_dat_idtipocontrato_fkey FOREIGN KEY (idtipocontrato)
  REFERENCES rh_tipocontrato_dat(idtipocontrato);
```



```
ALTER TABLE ONLY rol
  ADD CONSTRAINT rol_idportal_fkey FOREIGN KEY (idportal) REFERENCES
portal(idportal) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY inv_articulo_dat
  ADD CONSTRAINT unidad_fk FOREIGN KEY (codi_unid, codi_empr) REFERENCES
inv_unidad_dat(codi_unid, codi_empr) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY usuariorol
  ADD CONSTRAINT usuariorol_idrol_fkey FOREIGN KEY (idrol) REFERENCES rol(idrol)
ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY usuariorol
  ADD CONSTRAINT usuariorol_idusuario_fkey FOREIGN KEY (idusuario) REFERENCES
usuario(idusuario) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY cabecera_factura
  ADD CONSTRAINT codi_fact PRIMARY KEY (codi_fact);

ALTER TABLE ONLY cliente
  ADD CONSTRAINT codi_clie PRIMARY KEY (codi_clie);

ALTER TABLE ONLY tipo_cliente
  ADD CONSTRAINT codi_tipo PRIMARY KEY (codi_tipo);

ALTER TABLE ONLY sucursal
  ADD CONSTRAINT codi_sucu PRIMARY KEY (codi_sucu);

ALTER TABLE ONLY cliente
  ADD CONSTRAINT codi_ciu FOREIGN KEY (codi_ciu) REFERENCES
car_ciud_dat(codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY cliente
  ADD CONSTRAINT tipo_cliente FOREIGN KEY (tipo_cliente) REFERENCES
tipo_cliente(codi_tipo) ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY cliente
  ADD CONSTRAINT status FOREIGN KEY (status) REFERENCES
car_stat_dat(codi_stat_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY cabecera_factura
  ADD CONSTRAINT codi_clie FOREIGN KEY (codi_clie) REFERENCES cliente(codi_clie)
ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY cabecera_factura
  ADD CONSTRAINT codi_sucu FOREIGN KEY (codi_sucu) REFERENCES sucursal(codi_sucu)
ON UPDATE RESTRICT ON DELETE RESTRICT;
ALTER TABLE ONLY cabecera_factura
  ADD CONSTRAINT status FOREIGN KEY (status) REFERENCES
car_stat_dat(codi_stat_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY detalle_factura
  ADD CONSTRAINT codi_pk PRIMARY KEY (codi_fact, codi_arti);
```

```
ALTER TABLE ONLY sucursal
  ADD CONSTRAINT codi_ciudad FOREIGN KEY (codi_ciud) REFERENCES
  car_ciud_dat(codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY sucursal
  ADD CONSTRAINT status FOREIGN KEY (status) REFERENCES
  car_stat_dat(codi_stat_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY tipo_cliente
  ADD CONSTRAINT status FOREIGN KEY (status) REFERENCES
  car_stat_dat(codi_stat_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY mar_cabecera_combo_dat
  ADD CONSTRAINT codi_comb PRIMARY KEY (codi_comb_pk);

ALTER TABLE ONLY mar_cabecera_combo_dat
  ADD CONSTRAINT estado FOREIGN KEY (codi_stat_fk) REFERENCES
  car_stat_dat(codi_stat_pk) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY mar_cabecera_combo_dat
  ADD CONSTRAINT codi_empresa FOREIGN KEY (codi_empr_pk) REFERENCES
  inv_empresa_dat(codi_empr) ON UPDATE RESTRICT ON DELETE RESTRICT;

ALTER TABLE ONLY detalle_factura
  ADD CONSTRAINT detalle_factura_fk FOREIGN KEY (codi_fact) REFERENCES
  cabecera_factura(codi_fact);

ALTER TABLE ONLY mar_detalle_combo_dat
  ADD CONSTRAINT mar_detalle_combo_dat_fk FOREIGN KEY (codi_comb_fk) REFERENCES
  mar_cabecera_combo_dat(codi_comb_pk);
```

## 1.10 CODIFICACION DE LOS COMPONENTES

### CLASE Reporte.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.HashMap;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class Reporte extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws
        ServletException, IOException {
        String BandReport="/reporte/sugerencia.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

                parametros.put("desde",request.getParameter("desde"));
                parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteCompoClientes.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import javax.naming.NamingException;
import javax.servlet.*;
import javax.servlet.http.*;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class ReporteCompoClientes extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws
        ServletException, IOException {

        String BandReport="/reporte/comporta_clientes.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

                parametros.put("tipo_cliente",request.getParameter("tipo_cliente"));
                parametros.put("cliente",request.getParameter("cliente"));
                parametros.put("pais",request.getParameter("pais"));
                parametros.put("ciudad",request.getParameter("ciudad"));
                parametros.put("sucursal",request.getParameter("sucursal"));
                parametros.put("desde",request.getParameter("desde"));
                parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteGraficoCaract.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class ReporteGraficoCaract extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {

        String BandReport="/reporte/ventas_producto_completo02_grafico.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

                parametros.put("grupo",request.getParameter("grupo"));
                parametros.put("familia",request.getParameter("familia"));
                parametros.put("marca",request.getParameter("marca"));
                parametros.put("desde",request.getParameter("desde"));
                parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteGraficoVtasU.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.HashMap;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class ReporteGraficoVtasU extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws
        ServletException, IOException {

        String BandReport="/reporte/ventas_producto_completo01_grafico.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

                parametros.put("pais",request.getParameter("pais"));
                parametros.put("ciudad",request.getParameter("ciudad"));
                parametros.put("sucursal",request.getParameter("sucursal"));
                parametros.put("desde",request.getParameter("desde"));
                parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteGraficoVtas.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.HashMap;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class ReporteGraficoVtas extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {

        String BandReport="/reporte/ventas_producto_completo01.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

            parametros.put("pais",request.getParameter("pais"));
            parametros.put("ciudad",request.getParameter("ciudad"));
            parametros.put("sucursal",request.getParameter("sucursal"));
            parametros.put("desde",request.getParameter("desde"));
            parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteInvFlash.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import java.util.HashMap;
import java.util.Map;
import javax.naming.NamingException;
import javax.servlet.*;
import javax.servlet.http.*;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class ReporteInvFlash extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    public void service(HttpServletRequest request,
        HttpServletResponse response) throws
        ServletException, IOException {

        String BandReport="/reporte/productos_bodega.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

            parametros.put("status",request.getParameter("status"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```



## CLASE ReporteProdNoVendido.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import javax.naming.NamingException;
import javax.servlet.*;
import javax.servlet.http.*;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;
public class ReporteProdNoVendido extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-
1252";

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    public void service(HttpServletRequest request,
        HttpServletResponse response) throws
        ServletException, IOException {

        String BandReport="/reporte/productos_no_vendidos.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

            parametros.put("desde",request.getParameter("desde"));
            parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteProdVendido.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import javax.servlet.http.*;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;
public class ReporteProdVendido extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-
1252";

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
                        HttpServletResponse response) throws
        ServletException, IOException {

        String BandReport="/reporte/grafico_total_vtas_articulo.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

            parametros.put("desde",request.getParameter("desde"));
            parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteSusc.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.sql.SQLException;
import javax.servlet.http.*;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class ReporteSusc extends HttpServlet{

    public ReporteSusc() {

    }

    private static final String CONTENT_TYPE = "text/html; charset=windows-
1252";

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }

    public void service(HttpServletRequest request,
                        HttpServletResponse response) throws
        ServletException, IOException {

        String BandReport="/reporte/suscriptores.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

            parametros.put("estado",request.getParameter("estado"));
            parametros.put("sexo",request.getParameter("sexo"));
            parametros.put("status",request.getParameter("status"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteVtas.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import javax.servlet.http.*;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;
public class ReporteVtas extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-
1252";

    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
                        HttpServletResponse response) throws
        ServletException, IOException {

        String BandReport="/reporte/ventas_producto.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

            parametros.put("desde",request.getParameter("desde"));
            parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE ReporteVtasCaracteristicas.java

```

package com.ug.cisc.erp.marketing.view;
import com.ug.cisc.erp.marketing.dao.util.Base;
import java.io.File;
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.SQLException;
import javax.servlet.http.*;
import net.sf.jasperreports.engine.*;
import net.sf.jasperreports.engine.JRExporterParameter;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.JasperPrint;
import net.sf.jasperreports.engine.JasperReport;
import net.sf.jasperreports.engine.JasperRunManager;
import net.sf.jasperreports.engine.export.JRHtmlExporter;
import net.sf.jasperreports.engine.export.JRHtmlExporterParameter;
import net.sf.jasperreports.engine.util.JRLoader;

public class ReporteVtasCaracteristicas extends HttpServlet {
    private static final String CONTENT_TYPE = "text/html; charset=windows-1252";
    public void init(ServletConfig config) throws ServletException {
        super.init(config);
    }
    public void service(HttpServletRequest request,
        HttpServletResponse response) throws ServletException,
        IOException {

        String BandReport="/reporte/ventas_producto_completo02.jasper";
        response.setContentType("text/html");
        ServletContext sc;
        sc= this.getServletContext();
        try {
            File reportFile = new File(sc.getRealPath(BandReport));
            JasperReport jasperReport
            =(JasperReport)JRLoader.loadObject(reportFile.getPath());
            Map parametros = new HashMap();

                parametros.put("grupo",request.getParameter("grupo"));
                parametros.put("familia",request.getParameter("familia"));
                parametros.put("marca",request.getParameter("marca"));
                parametros.put("desde",request.getParameter("desde"));
                parametros.put("hasta",request.getParameter("hasta"));

            Connection unCon=Base.getConexion();
            byte[] bytes = null;
            bytes =
                JasperRunManager.runReportToPdf(
                    jasperReport,
                    parametros,
                    unCon);
            response.setContentType("application/pdf");
            response.setContentLength(bytes.length);
            ServletOutputStream ouputStream = response.getOutputStream();
            ouputStream.write(bytes, 0, bytes.length);
            ouputStream.flush();
            ouputStream.close();
        }catch(JRException j){
            System.out.println(j);
        }
    }
}

```

## CLASE FechaTag.java

```
package com.ug.cisc.erp.marketing.tags;
import java.util.List;
import java.util.ArrayList;
import javax.servlet.jsp.tagext.TagSupport;
public class FechaTag extends TagSupport {
public int doStartTag() {
    int i=0;
    int j=0;
    List diaValue = new ArrayList();
    for (i=1;i<=31;i++)
    {
    diaValue.add(Integer.toString( i ));
    }
    List mesValue = new ArrayList();
    mesValue.add("1");
    mesValue.add("2");
    mesValue.add("3");
    mesValue.add("4");
    mesValue.add("5");
    mesValue.add("6");
    mesValue.add("7");
    mesValue.add("8");
    mesValue.add("9");
    mesValue.add("10");
    mesValue.add("11");
    mesValue.add("12");
    List mesLabel = new ArrayList();
    mesLabel.add("Enero");
    mesLabel.add("Febrero");
    mesLabel.add("Marzo");
    mesLabel.add("Abril");
    mesLabel.add("Mayo");
    mesLabel.add("Junio");
    mesLabel.add("Julio");
    mesLabel.add("Agosto");
    mesLabel.add("Septiembre");
    mesLabel.add("Octubre");
    mesLabel.add("Noviembre");
    mesLabel.add("Diciembre");
    List anioValue = new ArrayList();
    for (j=1950;j<=2006;j++)
    {
    anioValue.add(Integer.toString( j ));
    }
    pageContext.setAttribute("diaValue",diaValue);
    pageContext.setAttribute("mesLabel",mesLabel);
    pageContext.setAttribute("mesValue",mesValue);
    pageContext.setAttribute("anioValue",anioValue);
    return SKIP_BODY;
}
}
```

## CLASE FechaTagInf.java

```
package com.ug.cisc.erp.marketing.tags;
import java.util.List;
import java.util.ArrayList;
import javax.servlet.jsp.tagext.TagSupport;
public class FechaTagInf extends TagSupport{
    public FechaTagInf() {
        super();
    }
    public int doStartTag() {

        int i=0;
        int j=0;
        List diaValueInf = new ArrayList();
        for (i=1;i<=31;i++)
        {
            diaValueInf.add(Integer.toString( i ));
        }
        List mesValueInf = new ArrayList();
        mesValueInf.add("1");
        mesValueInf.add("2");
        mesValueInf.add("3");
        mesValueInf.add("4");
        mesValueInf.add("5");
        mesValueInf.add("6");
        mesValueInf.add("7");
        mesValueInf.add("8");
        mesValueInf.add("9");
        mesValueInf.add("10");
        mesValueInf.add("11");
        mesValueInf.add("12");

        List mesLabelInf = new ArrayList();

        mesLabelInf.add("Enero");
        mesLabelInf.add("Febrero");
        mesLabelInf.add("Marzo");
        mesLabelInf.add("Abril");
        mesLabelInf.add("Mayo");
        mesLabelInf.add("Junio");
        mesLabelInf.add("Julio");
        mesLabelInf.add("Agosto");
        mesLabelInf.add("Septiembre");
        mesLabelInf.add("Octubre");
        mesLabelInf.add("Noviembre");
        mesLabelInf.add("Diciembre");

        List anioValueInf = new ArrayList();

        for (j=2004;j<=2020;j++)
        {

            anioValueInf.add(Integer.toString( j ));

        }

        pageContext.setAttribute("diaValueInf",diaValueInf);
        pageContext.setAttribute("mesLabelInf",mesLabelInf);
        pageContext.setAttribute("mesValueInf",mesValueInf);
        pageContext.setAttribute("anioValueInf",anioValueInf);
        return SKIP_BODY;
    }
}
```

## CLASE car\_ciu\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;
public class car_ciu_dat {
    private Integer codi_ciud_pk;
    private Integer codi_empr_pk;
    private String desc_ciud;
    private Integer codi_prov_pk;
    private Integer codi_pais_pk;
    private Integer codi_cant_pk;
    private Date audit_fech_ingr;
    private String audit_usua_ingr;
    private Date audit_fech_modi;
    private String audit_usua_modi;
    public car_ciu_dat() {}
    public Date getAudit_fech_ingr() {
        return audit_fech_ingr;
    }
    public void setAudit_fech_ingr(Date audit_fech_ingr) {
        this.audit_fech_ingr = audit_fech_ingr;
    }
    public Date getAudit_fech_modi() {
        return audit_fech_modi;
    }
    public void setAudit_fech_modi(Date audit_fech_modi) {
        this.audit_fech_modi = audit_fech_modi;
    }
    public String getAudit_usua_ingr() {
        return audit_usua_ingr;
    }
    public void setAudit_usua_ingr(String audit_usua_ingr) {
        this.audit_usua_ingr = audit_usua_ingr;
    }
    public String getAudit_usua_modi() {
        return audit_usua_modi;
    }
    public void setAudit_usua_modi(String audit_usua_modi) {
        this.audit_usua_modi = audit_usua_modi;
    }
    public Integer getCodi_cant_pk() {
        return codi_cant_pk;
    }
    public void setCodi_cant_pk(Integer codi_cant_pk) {
        this.codi_cant_pk = codi_cant_pk;
    }
    public Integer getCodi_ciud_pk() {
        return codi_ciud_pk;
    }
    public void setCodi_ciud_pk(Integer codi_ciud_pk) {
        this.codi_ciud_pk = codi_ciud_pk;
    }
    public Integer getCodi_empr_pk() {
        return codi_empr_pk;
    }
    public void setCodi_empr_pk(Integer codi_empr_pk) {
        this.codi_empr_pk = codi_empr_pk;
    }
    public Integer getCodi_pais_pk() {
        return codi_pais_pk;
    }
    public void setCodi_pais_pk(Integer codi_pais_pk) {
        this.codi_pais_pk = codi_pais_pk;
    }
    public Integer getCodi_prov_pk() {
        return codi_prov_pk;
    }
    public void setCodi_prov_pk(Integer codi_prov_pk) {
        this.codi_prov_pk = codi_prov_pk;
    }
    public String getDesc_ciud() {
        return desc_ciud;
    }
    public void setDesc_ciud(String desc_ciud) {
        this.desc_ciud = desc_ciud;
    }
}
```



**CLASE Cliente.java**

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;
public class Cliente {
    private Integer codi_clie;
    private String descripcion;
    private String direccion;
    private Integer codi_ciud;
    private Integer tipo_cliente;
    private String status;
    public Cliente() {

    }
    public Integer getCodi_clie() {
        return codi_clie;
    }
    public void setCodi_clie(Integer codi_clie) {
        this.codi_clie = codi_clie;
    }
    public Integer getCodi_ciud() {
        return codi_ciud;
    }
    public void setCodi_ciud(Integer codi_ciud) {
        this.codi_ciud = codi_ciud;
    }
    public Integer getTipo_cliente() {
        return tipo_cliente;
    }
    public void setTipo_cliente(Integer tipo_cliente) {
        this.tipo_cliente = tipo_cliente;
    }
    public String getDescripcion() {
        return descripcion;
    }
    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }
    public String getDireccion() {
        return direccion;
    }
    public void setDireccion(String direccion) {
        this.direccion = direccion;
    }
    public String getStatus() {
        return status;
    }
    public void setStatus(String status) {
        this.status = status;
    }
}
```

## CLASE inv\_articulo\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;
public class inv_articulo_dat {
    private Integer codi_cuen_cost;
    private Float prec_arti;
    private Float prec_prom_arti;
    private Float cost_prom_arti;
    private String ubic_arti;
    private Float peso_netto_arti;
    private Float peso_brut_arti;
    private Float volu_arti;
    private Integer codi_grup;
    private Integer codi_fami;
    private Integer codi_marc;
    private Integer codi_gene;
    private Float ulti_comp_arti;
    private Float ulti_vent_arti;
    private Float ulti_movi_arti;
    private Float ulti_trans_arti;
    private Float stoc_arti;
    private String imag_arti;
    private String tipo_arti;
    private Integer codi_empr;
    private String esta_arti;
    private String audi_usua_ingr;
    private Date audi_fech_ingr;
    private String audi_usua_modi;
    private Date audi_fech_modi;

    public inv_articulo_dat() {
    }

    public String getAlte_arti() {
        return alte_arti;
    }
    public void setAlte_arti(String alte_arti) {
        this.alte_arti = alte_arti;
    }
    public Date getAudi_fech_ingr() {
        return audi_fech_ingr;
    }
    public void setAudi_fech_ingr(Date audi_fech_ingr) {
        this.audi_fech_ingr = audi_fech_ingr;
    }
    public Date getAudi_fech_modi() {
        return audi_fech_modi;
    }
    public void setAudi_fech_modi(Date audi_fech_modi) {
        this.audi_fech_modi = audi_fech_modi;
    }
    public String getAudi_usua_ingr() {
        return audi_usua_ingr;
    }
    public void setAudi_usua_ingr(String audi_usua_ingr) {
        this.audi_usua_ingr = audi_usua_ingr;
    }
}
}
```

## CLASE inv\_empresa\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;
public class inv_empresa_dat {
    private Integer codi_empr;
    private String desc_empr;
    private String dire_empr;
    private String mail_empr;
    private Integer codi_pais;
    private Integer codi_prov;
    private Integer codi_ciud;
    private Integer codi_parr;
    private Integer codi_sect;
    private String nsri_empr;
    private String esta_empr;
    private String audi_usua_ingr;
    private Date audi_fech_ingr;
    private String audi_usua_modi;
    private Date audi_fech_modi;

    public inv_empresa_dat() {}
    public String getActi_empr() {
        return acti_empr;
    }
    public void setActi_empr(String acti_empr) {
        this.acti_empr = acti_empr;
    }
    public Date getAudi_fech_ingr() {
        return audi_fech_ingr;
    }
    public void setAudi_fech_ingr(Date audi_fech_ingr) {
        this.audi_fech_ingr = audi_fech_ingr;
    }
    public Date getAudi_fech_modi() {
        return audi_fech_modi;
    }
    public void setAudi_fech_modi(Date audi_fech_modi) {
        this.audi_fech_modi = audi_fech_modi;
    }
    public String getAudi_usua_ingr() {
        return audi_usua_ingr;
    }
    public void setAudi_usua_ingr(String audi_usua_ingr) {
        this.audi_usua_ingr = audi_usua_ingr;
    }
    public String getAudi_usua_modi() {
        return audi_usua_modi;
    }
    public void setAudi_usua_modi(String audi_usua_modi) {
        this.audi_usua_modi = audi_usua_modi;
    }
    public Integer getCodi_ciud() {
        return codi_ciud;
    }
    public void setCodi_ciud(Integer codi_ciud) {
        this.codi_ciud = codi_ciud;
    }
    public Integer getCodi_empr() {
        return codi_empr;
    }
    public void setCodi_empr(Integer codi_empr) {
        this.codi_empr = codi_empr;
    }
    public Integer getCodi_pais() {
        return codi_pais;
    }
    public void setCodi_pais(Integer codi_pais) {
        this.codi_pais = codi_pais;
    }
    public Integer getCodi_parr() {
        return codi_parr;
    }
    public void setCodi_parr(Integer codi_parr) {
        this.codi_parr = codi_parr;
    }
    public Integer getCodi_prov() {
        return codi_prov;
    }
    public void setCodi_prov(Integer codi_prov) {
        this.codi_prov = codi_prov;
    }
}
}
```

## CLASE inv\_familia\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;
public class inv_familia_dat {
    private String esta_fami;
    private String audi_usua_ingr;
    private Date audi_fech_ingr;
    private String audi_usua_modi;
    private Date audi_fech_modi;

    public inv_familia_dat() {}
    public Date getAudi_fech_ingr() {
        return audi_fech_ingr;
    }
    public void setAudi_fech_ingr(Date audi_fech_ingr) {
        this.audi_fech_ingr = audi_fech_ingr;
    }
    public Date getAudi_fech_modi() {
        return audi_fech_modi;
    }
    public void setAudi_fech_modi(Date audi_fech_modi) {
        this.audi_fech_modi = audi_fech_modi;
    }
    public String getAudi_usua_ingr() {
        return audi_usua_ingr;
    }
    public void setAudi_usua_ingr(String audi_usua_ingr) {
        this.audi_usua_ingr = audi_usua_ingr;
    }
    public String getAudi_usua_modi() {
        return audi_usua_modi;
    }
    public void setAudi_usua_modi(String audi_usua_modi) {
        this.audi_usua_modi = audi_usua_modi;
    }
    public Integer getCodi_empr() {
        return codi_empr;
    }
    public void setCodi_empr(Integer codi_empr) {
        this.codi_empr = codi_empr;
    }
    public Integer getCodi_fami() {
        return codi_fami;
    }
    public void setCodi_fami(Integer codi_fami) {
        this.codi_fami = codi_fami;
    }
    public Integer getCodi_grup() {
        return codi_grup;
    }
    public void setCodi_grup(Integer codi_grup) {
        this.codi_grup = codi_grup;
    }
    public String getDesc_fami() {
        return desc_fami;
    }
    public void setDesc_fami(String desc_fami) {
        this.desc_fami = desc_fami;
    }
}
}
```

## CLASE inv\_grupo\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;
public class inv_grupo_dat {
    private Integer codi_grup;
    private String desc_grup;
    private Integer codi_empr;
    private String esta_grup;
    private String audi_usua_ingr;
    private Date audi_fech_ingr;
    private String audi_usua_modi;
    private Date audi_fech_modi;
    public inv_grupo_dat() {}
    public Date getAudi_fech_ingr() {
        return audi_fech_ingr;
    }
    public void setAudi_fech_ingr(Date audi_fech_ingr) {
        this.audi_fech_ingr = audi_fech_ingr;
    }
    public Date getAudi_fech_modi() {
        return audi_fech_modi;
    }
    public void setAudi_fech_modi(Date audi_fech_modi) {
        this.audi_fech_modi = audi_fech_modi;
    }
    public String getAudi_usua_ingr() {
        return audi_usua_ingr;
    }
    public void setAudi_usua_ingr(String audi_usua_ingr) {
        this.audi_usua_ingr = audi_usua_ingr;
    }
    public String getAudi_usua_modi() {
        return audi_usua_modi;
    }
    public void setAudi_usua_modi(String audi_usua_modi) {
        this.audi_usua_modi = audi_usua_modi;
    }
    public Integer getCodi_empr() {
        return codi_empr;
    }
    public void setCodi_empr(Integer codi_empr) {
        this.codi_empr = codi_empr;
    }
    public Integer getCodi_grup() {
        return codi_grup;
    }
    public void setCodi_grup(Integer codi_grup) {
        this.codi_grup = codi_grup;
    }
    public String getDesc_grup() {
        return desc_grup;
    }
    public void setDesc_grup(String desc_grup) {
        this.desc_grup = desc_grup;
    }
    public String getEsta_grup() {
        return esta_grup;
    }
    public void setEsta_grup(String esta_grup) {
        this.esta_grup = esta_grup;
    }
}
}
```

## CLASE inv\_marca\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;

public class inv_marca_dat {
    private Integer codi_grup;
    private String esta_marc;
    private String audi_usua_ingr;
    private Date audi_fech_ingr;
    private String audi_usua_modi;
    private Date audi_fech_modi;

    public inv_marca_dat() {}
    public Date getAudi_fech_ingr() {
        return audi_fech_ingr;
    }
    public void setAudi_fech_ingr(Date audi_fech_ingr) {
        this.audi_fech_ingr = audi_fech_ingr;
    }
    public Date getAudi_fech_modi() {
        return audi_fech_modi;
    }
    public void setAudi_fech_modi(Date audi_fech_modi) {
        this.audi_fech_modi = audi_fech_modi;
    }
    public String getAudi_usua_ingr() {
        return audi_usua_ingr;
    }
    public void setAudi_usua_ingr(String audi_usua_ingr) {
        this.audi_usua_ingr = audi_usua_ingr;
    }
    public String getAudi_usua_modi() {
        return audi_usua_modi;
    }
    public void setAudi_usua_modi(String audi_usua_modi) {
        this.audi_usua_modi = audi_usua_modi;
    }
    public Integer getCodi_empr() {
        return codi_empr;
    }
    public void setCodi_empr(Integer codi_empr) {
        this.codi_empr = codi_empr;
    }
    public Integer getCodi_fami() {
        return codi_fami;
    }
    public void setCodi_fami(Integer codi_fami) {
        this.codi_fami = codi_fami;
    }
    public Integer getCodi_grup() {
        return codi_grup;
    }
    public void setCodi_grup(Integer codi_grup) {
        this.codi_grup = codi_grup;
    }
    public Integer getCodi_marc() {
        return codi_marc;
    }
    public void setCodi_marc(Integer codi_marc) {
        this.codi_marc = codi_marc;
    }
}
```

## CLASE inv\_unidad\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;

public class inv_unidad_dat {
    private Integer codi_unid;
    private String desc_unid;
    private Date audi_fech_ingr;
    private String audi_usua_modi;
    private Date audi_fech_modi;

    public inv_unidad_dat() {}

    public Date getAudi_fech_ingr() {
        return audi_fech_ingr;
    }
    public void setAudi_fech_ingr(Date audi_fech_ingr) {
        this.audi_fech_ingr = audi_fech_ingr;
    }
    public Date getAudi_fech_modi() {
        return audi_fech_modi;
    }
    public void setAudi_fech_modi(Date audi_fech_modi) {
        this.audi_fech_modi = audi_fech_modi;
    }
    public String getAudi_usua_ingr() {
        return audi_usua_ingr;
    }
    public void setAudi_usua_ingr(String audi_usua_ingr) {
        this.audi_usua_ingr = audi_usua_ingr;
    }
    public String getAudi_usua_modi() {
        return audi_usua_modi;
    }
    public void setAudi_usua_modi(String audi_usua_modi) {
        this.audi_usua_modi = audi_usua_modi;
    }
    public Integer getCodi_empr() {
        return codi_empr;
    }
    public void setCodi_empr(Integer codi_empr) {
        this.codi_empr = codi_empr;
    }
    public Integer getCodi_unid() {
        return codi_unid;
    }
    public void setCodi_unid(Integer codi_unid) {
        this.codi_unid = codi_unid;
    }
    public String getDesc_unid() {
        return desc_unid;
    }
    public void setDesc_unid(String desc_unid) {
        this.desc_unid = desc_unid;
    }
    public String getEsta_unid() {
        return esta_unid;
    }
    public void setEsta_unid(String esta_unid) {
        this.esta_unid = esta_unid;
    }
}
}
```

## CLASE mar\_cabecera\_combo\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionMapping;
import java.util.*;

public class mar_cabecera_combo_dat {
    private Integer codi_comb_pk;
    private String desc_comb;
    private Date fech_ini_comb;
    private Date audit_fech_ingr;
    private String audit_usua_modi;
    private Date audit_fech_modi;
    private Integer n_item;

    public mar_cabecera_combo_dat() {
    }
    public Date getAudit_fech_ingr() {
        return audit_fech_ingr;
    }
    public void setAudit_fech_ingr(Date audit_fech_ingr) {
        this.audit_fech_ingr = audit_fech_ingr;
    }
    public Date getAudit_fech_modi() {
        return audit_fech_modi;
    }
    public void setAudit_fech_modi(Date audit_fech_modi) {
        this.audit_fech_modi = audit_fech_modi;
    }
    public String getAudit_usua_ingr() {
        return audit_usua_ingr;
    }
    public void setAudit_usua_ingr(String audit_usua_ingr) {
        this.audit_usua_ingr = audit_usua_ingr;
    }
    public String getAudit_usua_modi() {
        return audit_usua_modi;
    }
    public Integer getCodi_comb_pk() {
        return codi_comb_pk;
    }
    public void setCodi_comb_pk(Integer codi_comb_pk) {
        this.codi_comb_pk = codi_comb_pk;
    }
    public Integer getCodi_empr_fk() {
        return codi_empr_fk;
    }
    public void setCodi_empr_fk(Integer codi_empr_fk) {
        this.codi_empr_fk = codi_empr_fk;
    }
    public String getCodi_stat_fk() {
        return codi_stat_fk;
    }
    public void setCodi_stat_fk(String codi_stat_fk) {
        this.codi_stat_fk = codi_stat_fk;
    }
    public String getDesc_comb() {
        return desc_comb;
    }
    public void setDesc_comb(String desc_comb) {
        this.desc_comb = desc_comb;
    }
}
}
```



**CLASE mar\_detalle\_combo\_dat.java**

```
package com.ug.cisc.erp.marketing.entity;

public class mar_detalle_combo_dat {
    private Integer codi_comb_fk;
    private Integer codi_arti_fk;
    private Integer cant_arti_comb;
    private Float tasa_dcto;

    public mar_detalle_combo_dat() {

    }
    public Integer getCant_arti_comb() {
        return cant_arti_comb;
    }
    public void setCant_arti_comb(Integer cant_arti_comb) {
        this.cant_arti_comb = cant_arti_comb;
    }
    public Integer getCodi_arti_fk() {
        return codi_arti_fk;
    }
    public void setCodi_arti_fk(Integer codi_arti_fk) {
        this.codi_arti_fk = codi_arti_fk;
    }
    public Integer getCodi_comb_fk() {
        return codi_comb_fk;
    }
    public void setCodi_comb_fk(Integer codi_comb_fk) {
        this.codi_comb_fk = codi_comb_fk;
    }
    public Float getTasa_dcto() {
        return tasa_dcto;
    }
    public void setTasa_dcto(Float tasa_dcto) {
        this.tasa_dcto = tasa_dcto;
    }
}
```

## CLASE mar\_sugerencia\_tmp.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.*;

public class mar_sugerencia_tmp {

    private String nombre;
    private String empresa;
    private String e_mail;
    private String pais;
    private String ciudad;
    private String comentario;
    private Date fecha;

    public mar_sugerencia_tmp() {

    }

    public String getCiudad() {
        return ciudad;
    }

    public void setCiudad(String ciudad) {
        this.ciudad = ciudad;
    }

    public String getComentario() {
        return comentario;
    }

    public void setComentario(String comentario) {
        this.comentario = comentario;
    }

    public String getE_mail() {
        return e_mail;
    }

    public void setE_mail(String e_mail) {
        this.e_mail = e_mail;
    }

    public String getEmpresa() {
        return empresa;
    }

    public void setEmpresa(String empresa) {
        this.empresa = empresa;
    }

    public Date getFecha() {
        return fecha;
    }

    public void setFecha(Date fecha) {
        this.fecha = fecha;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getPais() {
        return pais;
    }

    public void setPais(String pais) {
        this.pais = pais;
    }

}
```

## CLASE mar\_suscriptor\_dat.java

```
package com.ug.cisc.erp.marketing.entity;
import java.util.GregorianCalendar;
import org.apache.struts.action.ActionForm;
import java.util.*;
public class mar_suscriptor_dat {
    private Integer codi_susc_pk;
    private String nomb susc;
    private String apel_susc;
    private String cedu_susc;
    private String sexo_susc;
    private String dire_susc;
    private String e_mail_susc;
    private String fono_susc;
    private String empr_susc;
    private String audit_usua_ingr;
    private Date audit_fech_ingr;
    private String audit_usua_modi;
    private Date audit_fech_modi;
    private Integer codi_ciud_fk;
    private String ciudad;
    private Integer codi_prov_fk;
    private String provincia;
    private Integer codi_pais_fk;
    private String pais;
    private String codi_stat_fk;
    private Integer codi_empr_fk;

    public mar_suscriptor_dat() {

    }
    public Integer getAnio_nac() {
        return anio_nac;
    }
    public void setAnio_nac(int i) {
        this.anio_nac = anio_nac;
    }
    public String getApel_susc() {
        return apel_susc;
    }
    public void setApel_susc(String apel_susc) {
        this.apel_susc = apel_susc;
    }
    public Date getAudit_fech_ingr() {
        return audit_fech_ingr;
    }
    public void setAudit_fech_ingr(Date audit_fech_ingr) {
        this.audit_fech_ingr = audit_fech_ingr;
    }
    public Date getAudit_fech_modi() {
        return audit_fech_modi;
    }
    public void setAudit_fech_modi(Date audit_fech_modi) {
        this.audit_fech_modi = audit_fech_modi;
    }
    public String getAudit_usua_ingr() {
        return audit_usua_ingr;
    }
}
}
```

## CLASE TipoCliente.java

```
package com.ug.cisc.erp.marketing.entity;

public class TipoCliente {
    private Integer codi_tipo;
    private String descripcion;
    private String status;

    public TipoCliente() {
    }

    public Integer getCodi_tipo() {
        return codi_tipo;
    }

    public void setCodi_tipo(Integer codi_tipo) {
        this.codi_tipo = codi_tipo;
    }

    public String getDescripcion() {
        return descripcion;
    }

    public void setDescripcion(String descripcion) {
        this.descripcion = descripcion;
    }

    public String getStatus() {
        return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }
}
```

## CLASE User.java

```
package com.ug.cisc.erp.marketing.entity;

public class User {
    int userId;
    String userName;
    String password;
    String rol;
    String name;
    String apellido;

    public User() {
    }

    public void setUserId(int userId) {
        this.userId = userId;
    }
    public int getUserId() {
        return this.userId;
    }

    public void setUserName(String userName) {
        this.userName = userName;
    }
    public String getUserName() {
        return this.userName;
    }

    public void setPassword(String password) {
        this.password = password;
    }
    public String getPassword() {
        return this.password;
    }

    public void setRol(String rol) {
        this.rol = rol;
    }
    public String getRol() {
        return this.rol;
    }

    public void setName(String name) {
        this.name = name;
    }
    public String getName() {
        return this.name;
    }

    public void setApellido(String apellido) {
        this.apellido = apellido;
    }
    public String getApellido() {
        return this.apellido;
    }
}
```

## CLASE Base.java

```
package com.ug.cisc.erp.marketing.dao.util;
import java.sql.Connection;
import java.sql.DriverManager;

public class Base {
private static Connection unCon= null;
    public static Connection getConexion(){
        if (unCon== null){
            try {
                Class.forName("org.postgresql.Driver");
                unCon=
DriverManager.getConnection("jdbc:postgresql://localhost:5432/erp_marketing",
"admin", "admin");
                unCon.setAutoCommit(false);
            }catch(Throwable e){
                System.out.println(e);
            }
        }
        return unCon;
    }
}
```

## CLASE ConnectionPool.java

```
package com.ug.cisc.erp.marketing.dao.util;

import java.sql.SQLException;
import java.sql.Connection;
import javax.sql.DataSource;

public class ConnectionPool {

    private DataSource ds;

    private static ConnectionPool mySelf;

    private ConnectionPool(DataSource ds) {
        this.ds = ds;
    }

    public static void init(DataSource ds) {
        mySelf = new ConnectionPool(ds);
    }

    public static ConnectionPool getInstance() {

        if (mySelf == null) {
            throw new IllegalStateException("Pool No Inicializado.");
        }
        return mySelf;
    }

    public Connection getConnection() throws SQLException {
        return ds.getConnection();
    }
}
```

## CLASE DBInitServlet.java

```
package com.ug.cisc.erp.marketing.dao.util;

import javax.servlet.http.HttpServlet;
import javax.servlet.ServletConfig;
import javax.servlet.ServletException;

import java.sql.SQLException;
import java.sql.Connection;
import javax.sql.DataSource;

import org.apache.struts.util.GenericDataSource;

public class DBInitServlet extends HttpServlet {

    public void init(ServletConfig config) throws ServletException {

        super.init(config);

        try {

            GenericDataSource ds = new GenericDataSource();

            ds.setDriverClass(getInitParameter("driverClass"));
            ds.setUrl(getInitParameter("jdbcURL"));
            ds.setUser(getInitParameter("user"));
            ds.setPassword(getInitParameter("password"));
            ds.setMinCount(Integer.parseInt(getInitParameter("minCount")));
            ds.setMaxCount(Integer.parseInt(getInitParameter("maxCount")));

            ds.open();

            ConnectionPool.init(ds);

        } catch (SQLException e) {
            e.printStackTrace();
            throw new ServletException("Unable to open datasource.");
        }

    }

}
```



## CLASE CiudadDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import com.ug.cisc.erp.marketing.entity.*;
import com.ug.cisc.erp.marketing.dao.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class CiudadDAO {
    String error;
    Connection con;

    public CiudadDAO() {}

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
        try {
            Class.forName("org.postgresql.Driver").newInstance();
            con = DriverManager.getConnection(
                "jdbc:postgresql://localhost/erp_marketing", "admin", "admin");
        } catch (ClassNotFoundException cnfe) {
            error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
            throw new ClassNotFoundException(error);
        } catch (SQLException cnfe) {
            error = cnfe.toString();
            throw new SQLException(error);
        } catch (Exception e) {
            error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
            throw new Exception(error);
        }
    }

    public void disconnect() throws SQLException {
        try {
            if ( con != null ) {
                con.close();
            }
        } catch (SQLException sqle) {
            error = ("SQLException: Unable to close the database connection.");
            throw new SQLException(error);
        }
    }

    public ResultSet list() throws SQLException, Exception {
        ResultSet rs = null;
        try {
            String queryString = ("SELECT * FROM car_ciud_dat");
            Statement stmt = con.createStatement();
            rs = stmt.executeQuery(queryString);
        } catch (SQLException sqle) {
            error = "SQLException: Could not execute the query.";
            throw new SQLException(error);
        } catch (Exception e) {
            error = "An exception occured while retrieving books.";
            throw new Exception(error);
        }
        return rs;
    }
}

```

## CLASE ClienteDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import com.ug.cisc.erp.marketing.entity.*;
import com.ug.cisc.erp.marketing.dao.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class ClienteDAO {
    String error;
    Connection con;
    public ClienteDAO() {}

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
    try {
    Class.forName("org.postgresql.Driver").newInstance();
    con = DriverManager.getConnection(
    "jdbc:postgresql://localhost/erp_marketing","admin","admin");
    } catch (ClassNotFoundException cnfe) {
    error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
    throw new ClassNotFoundException(error);
    } catch (SQLException cnfe) {
    error = cnfe.toString();
    throw new SQLException(error);
    } catch (Exception e) {
    error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
    throw new Exception(error);
    }
    }

    public void disconnect() throws SQLException {
    try {
    if ( con != null ) {
    con.close();
    }
    } catch (SQLException sqle) {
    error = ("SQLException: Unable to close the database connection.");
    throw new SQLException(error);
    }
    }

    public ResultSet list() throws SQLException, Exception {
    ResultSet rs = null;
    try {
    String queryString = ("SELECT * FROM cliente;");
    Statement stmt = con.createStatement();
    rs = stmt.executeQuery(queryString);
    } catch (SQLException sqle) {
    error = "SQLException: Could not execute the query.";
    throw new SQLException(error);
    } catch (Exception e) {
    error = "An exception occured while retrieving books.";
    throw new Exception(error);
    }
    return rs;
    }
}

```

## CLASE GrupoDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class GrupoDAO {
String error;
Connection con;

    public GrupoDAO() {    }

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
try {
Class.forName("org.postgresql.Driver").newInstance();
con = DriverManager.getConnection(
"jdbc:postgresql://localhost/erp_marketing","admin","admin");
} catch (ClassNotFoundException cnfe) {
error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
throw new ClassNotFoundException(error);
} catch (SQLException cnfe) {
error = cnfe.toString();
throw new SQLException(error);
} catch (Exception e) {
error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
throw new Exception(error);
}
}

    public void disconnect() throws SQLException {
try {
if ( con != null ) {
con.close();
}
} catch (SQLException sqle) {
error = ("SQLException: Unable to close the database connection.");
throw new SQLException(error);
}
}

    public ResultSet list() throws SQLException, Exception {
ResultSet rs = null;
try {
String queryString = ("SELECT * FROM inv_grupo_dat;");
Statement stmt = con.createStatement();
rs = stmt.executeQuery(queryString);
} catch (SQLException sqle) {
error = "SQLException: Could not execute the query.";
throw new SQLException(error);
} catch (Exception e) {
error = "An exception occured while retrieving books.";
throw new Exception(error);
}
return rs;
}
}

```

## CLASE FamiliaDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class FamiliaDAO {
String error;
Connection con;

    public FamiliaDAO() {}

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
try {
Class.forName("org.postgresql.Driver").newInstance();
con = DriverManager.getConnection(
"jdbc:postgresql://localhost/erp_marketing","admin","admin");
} catch (ClassNotFoundException cnfe) {
error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
throw new ClassNotFoundException(error);
} catch (SQLException cnfe) {
error = cnfe.toString();
throw new SQLException(error);
} catch (Exception e) {
error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
throw new Exception(error);
}
}

    public void disconnect() throws SQLException {
try {
if ( con != null ) {
con.close();
}
} catch (SQLException sqle) {
error = ("SQLException: Unable to close the database connection.");
throw new SQLException(error);
}
}

    public ResultSet list() throws SQLException, Exception {
ResultSet rs = null;
try {
String queryString = ("SELECT * FROM inv_familia_dat;");
Statement stmt = con.createStatement();
rs = stmt.executeQuery(queryString);
} catch (SQLException sqle) {
error = "SQLException: Could not execute the query.";
throw new SQLException(error);
} catch (Exception e) {
error = "An exception ocurred while retrieving books.";
throw new Exception(error);
}
return rs;
}
}

```

## CLASE MarcaDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class MarcaDAO {

    String error;
    Connection con;

    public MarcaDAO() { }

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
    try {
    Class.forName("org.postgresql.Driver").newInstance();
    con = DriverManager.getConnection(
    "jdbc:postgresql://localhost/erp_marketing","admin","admin");
    } catch (ClassNotFoundException cnfe) {
    error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
    throw new ClassNotFoundException(error);
    } catch (SQLException cnfe) {
    error = cnfe.toString();
    throw new SQLException(error);
    } catch (Exception e) {
    error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
    throw new Exception(error);
    }
    }

    public void disconnect() throws SQLException {
    try {
    if ( con != null ) {
    con.close();
    }
    } catch (SQLException sqle) {
    error = ("SQLException: Unable to close the database connection.");
    throw new SQLException(error);
    }
    }

    public ResultSet list() throws SQLException, Exception {
    ResultSet rs = null;
    try {
    String queryString = ("SELECT * FROM inv_marca_dat;");
    Statement stmt = con.createStatement();
    rs = stmt.executeQuery(queryString);
    } catch (SQLException sqle) {
    error = "SQLException: Could not execute the query.";
    throw new SQLException(error);
    } catch (Exception e) {
    error = "An exception occured while retrieving books.";
    throw new Exception(error);
    }
    return rs;
    }
}

```

## CLASE PaisDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import com.ug.cisc.erp.marketing.entity.*;
import com.ug.cisc.erp.marketing.dao.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class PaisDAO {
    String error;
    Connection con;
    public PaisDAO() {}

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
    try {
    Class.forName("org.postgresql.Driver").newInstance();
    con = DriverManager.getConnection(
    "jdbc:postgresql://localhost/erp_marketing","admin","admin");
    } catch (ClassNotFoundException cnfe) {
    error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
    throw new ClassNotFoundException(error);
    } catch (SQLException cnfe) {
    error = cnfe.toString();
    throw new SQLException(error);
    } catch (Exception e) {
    error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
    throw new Exception(error);
    }
    }

    public void disconnect() throws SQLException {
    try {
    if ( con != null ) {
    con.close();
    }
    } catch (SQLException sqle) {
    error = ("SQLException: Unable to close the database connection.");
    throw new SQLException(error);
    }
    }

    public ResultSet list() throws SQLException, Exception {
    ResultSet rs = null;
    try {
    String queryString = ("SELECT * FROM car_pais_dat;");
    Statement stmt = con.createStatement();
    rs = stmt.executeQuery(queryString);
    } catch (SQLException sqle) {
    error = "SQLException: Could not execute the query.";
    throw new SQLException(error);
    } catch (Exception e) {
    error = "An exception occured while retrieving books.";
    throw new Exception(error);
    }
    return rs;
    }
}

```

## CLASE SucursalDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class SucursalDAO {
String error;
Connection con;

    public SucursalDAO() {}

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
try {
Class.forName("org.postgresql.Driver").newInstance();
con = DriverManager.getConnection(
"jdbc:postgresql://localhost/erp_marketing","admin","admin");
} catch (ClassNotFoundException cnfe) {
error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
throw new ClassNotFoundException(error);
} catch (SQLException cnfe) {
error = cnfe.toString();
throw new SQLException(error);
} catch (Exception e) {
error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
throw new Exception(error);
}
}

    public void disconnect() throws SQLException {
try {
if ( con != null ) {
con.close();
}
} catch (SQLException sqle) {
error = ("SQLException: Unable to close the database connection.");
throw new SQLException(error);
}
}

    public ResultSet list() throws SQLException, Exception {
ResultSet rs = null;
try {
String queryString = ("SELECT * FROM sucursal;");
Statement stmt = con.createStatement();
rs = stmt.executeQuery(queryString);
} catch (SQLException sqle) {
error = "SQLException: Could not execute the query.";
throw new SQLException(error);
} catch (Exception e) {
error = "An exception occured while retrieving books.";
throw new Exception(error);
}
return rs;
}
}

```

## CLASE SugerenciaDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import com.ug.cisc.erp.marketing.entity.mar_sugerencia_tmp;
import java.util.*;
import java.sql.SQLException;
import java.sql.Statement;
public class SugerenciaDAO {
    String error;
    Connection con;
    public SugerenciaDAO() { }
    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
        try {
            Class.forName("org.postgresql.Driver").newInstance();
            con = DriverManager.getConnection(
                "jdbc:postgresql://localhost/marketing", "admin", "admin");
        } catch (ClassNotFoundException cnfe) {
            error = "ClassNotFoundException: No se pudo localizar el
driver de conexión.";
            throw new ClassNotFoundException(error);
        } catch (SQLException cnfe) {
            error = cnfe.toString();
            throw new SQLException(error);
        } catch (Exception e) {
            error = "Exception: ha ocurrido una excepcion mientras se
conectaba a la base ";
            throw new Exception(error);
        }
    }
    public void addSugerencia(String NOMBRE,String EMPRESA,String E_MAIL,String
CIUDAD,String PAIS,String COMENTARIO,Date FECHA)
        throws SQLException, Exception {
        if (con != null) {
            try {
                PreparedStatement updatecabecomb;
                updatecabecomb = con.prepareStatement(
                    "INSERT INTO mar_sugerencia_tmp
(NOMBRE,EMPRESA,E_MAIL,CIUDAD,PAIS,COMENTARIO,FECHA) VALUES(?,?,?,?,?,?,?);");
                updatecabecomb.setString(1, NOMBRE);
                updatecabecomb.setString(2, EMPRESA);
                updatecabecomb.setString(3, E_MAIL);
                updatecabecomb.setString(4, CIUDAD);
                updatecabecomb.setString(5, PAIS);
                updatecabecomb.setString(6, COMENTARIO);
                updatecabecomb.execute();
            } catch (SQLException sqle) {
                error = sqle.toString();//"SQLException: update failed, possible
duplicate entry";
                throw new SQLException(error);}
            }
            else {
                error = "Exception: Connection to database was lost.";
                throw new Exception(error);}
            }
        }
    }
}

```



## CLASE SuscriptorDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import com.ug.cisc.erp.marketing.entity.mar_suscriptor_dat;
import java.sql.*;
import java.util.*;
public class SuscriptorDAO {
    String error;
    Connection con;
    public SuscriptorDAO() { }
    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
        try {
            Class.forName("org.postgresql.Driver").newInstance();
            con = DriverManager.getConnection(
                "jdbc:postgresql://localhost/marketing", "admin", "admin");
        } catch (ClassNotFoundException cnfe) {
            error = "ClassNotFoundException: No se pudo localizar el driver de
conexión.";
            throw new ClassNotFoundException(error);
        } catch (SQLException cnfe) {
            error = cnfe.toString();
            throw new SQLException(error);
        } catch (Exception e) {
            error = "Exception: ha ocurrido una excepcion mientras se conectaba a
la base ";
            throw new Exception(error);
        }
    }
    public void addSuscriptor(int CODI_SUSC_PK,String NOMB_SUSC,String
APEL_SUSC,String CEDU_SUSC,String SEXO_SUSC,String ESTA_CIVI_SUSC,Date
FECH_NACI_SUSC,String DIRE_SUSC,String E_MAIL_SUSC,String FONO_SUSC,String
EMPR_SUSC,int CODI_CIUUD_FK,int CODI_PROV_FK,int CODI_PAIS_FK,String
CODI_STAT_FK,int CODI_EMPR_FK,String AUDIT_USUA_INGR,Date AUDIT_FECH_INGR)
        throws SQLException, Exception {
        if (con != null) {
            try {
                PreparedStatement updatecabecomb;
                updatecabecomb = con.prepareStatement(
                    "INSERT INTO mar_suscriptor_dat
(CODI_SUSC_PK,NOMB_SUSC,APEL_SUSC,CEDU_SUSC,SEXO_SUSC,ESTA_CIVI_SUSC,FECH_NACI_SUSC
,DIRE_SUSC,E_MAIL_SUSC,FONO_SUSC,EMPR_SUSC,CODI_CIUUD_FK,CODI_PROV_FK,CODI_PAIS_FK,C
ODI_STAT_FK,CODI_EMPR_FK,AUDIT_USUA_INGR,AUDIT_FECH_INGR)
VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
                updatecabecomb.setInt(1, CODI_SUSC_PK);
                updatecabecomb.setString(2, NOMB_SUSC);
                updatecabecomb.setInt(16, CODI_EMPR_FK);
                updatecabecomb.setString(17, AUDIT_USUA_INGR);
                updatecabecomb.execute();
            } catch (SQLException sqle) {
                error = sqle.toString();//"SQLException: update failed, possible
duplicate entry";
                throw new SQLException(error);
            }
        } else {
            error = "Exception: Connection to database was lost.";
            throw new Exception(error);
        }
    }
}

```

## CLASE TipoClienteDAO.java

```
package com.ug.cisc.erp.marketing.dao;
import com.ug.cisc.erp.marketing.entity.*;
import com.ug.cisc.erp.marketing.dao.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class TipoClienteDAO {
    String error;
    Connection con;
    public TipoClienteDAO() {}

    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
        try {
            Class.forName("org.postgresql.Driver").newInstance();
            con = DriverManager.getConnection(
                "jdbc:postgresql://localhost/erp_marketing", "admin", "admin");
        } catch (ClassNotFoundException cnfe) {
            error = "ClassNotFoundException: No se pudo localizar el driver de conexión.";
            throw new ClassNotFoundException(error);
        } catch (SQLException cnfe) {
            error = cnfe.toString();
            throw new SQLException(error);
        } catch (Exception e) {
            error = "Exception: ha ocurrido una excepcion mientras se conectaba a la base ";
            throw new Exception(error);
        }
    }

    public void disconnect() throws SQLException {
        try {
            if ( con != null ) {
                con.close();
            }
        } catch (SQLException sqle) {
            error = ("SQLException: Unable to close the database connection.");
            throw new SQLException(error);
        }
    }

    public ResultSet list() throws SQLException, Exception {
        ResultSet rs = null;
        try {
            String queryString = ("SELECT * FROM tipo_cliente;");
            Statement stmt = con.createStatement();
            rs = stmt.executeQuery(queryString);
        } catch (SQLException sqle) {
            error = "SQLException: Could not execute the query.";
            throw new SQLException(error);
        } catch (Exception e) {
            error = "An exception occurred while retrieving books.";
            throw new Exception(error);
        }
        return rs;
    }
}
```

## CLASE UserDAO.java

```

package com.ug.cisc.erp.marketing.dao;
import com.ug.cisc.erp.marketing.entity.User;
import java.sql.*;
import java.util.*;
public class UserDAO {
    String error;
    Connection con;
    public UserDAO() { }
    public void connect() throws ClassNotFoundException,
        SQLException,
        Exception {
        try {
            Class.forName("org.postgresql.Driver").newInstance();
            con = DriverManager.getConnection(
                "jdbc:postgresql://localhost/erp_marketing","admin","admin");
        } catch (ClassNotFoundException cnfe) {
            error = "ClassNotFoundException: No se pudo localizar el driver de
conexión.";
            throw new ClassNotFoundException(error);
        } catch (SQLException cnfe) {
            error = cnfe.toString();
            throw new SQLException(error);
        } catch (Exception e) {
            error = "Exception: ha ocurrido una excepcion mientras se conectaba a
la base ";
            throw new Exception(error);
        }
    }
    public ResultSet findUser(int userId) throws SQLException, Exception {
        ResultSet rs = null;
        try {
            String queryString = ("SELECT * FROM usuarios WHERE idUsuario=?");
            PreparedStatement ps = con.prepareStatement(queryString);
            ps.setInt(1,userId);
            rs = ps.executeQuery();
        } catch (SQLException sqle) {
            error = "SQLException: No se ha podido ejecutar esta petición.
Contacte con el webmaster";
        } else {
            error = "Exception: Connection to database was lost.";
            throw new Exception(error);
        }
    }
    public void updateUser(User user) throws SQLException, Exception {
        if (con != null) {
            try {
                error = sqle.toString();//"SQLException: update failed, possible
duplicate entry";
                throw new SQLException(error);
            } catch (Exception e) {
                error = "An exception occured while deleting books.";
                throw new Exception(error);
            }
        } else {
            error = "Exception: Connection to database was lost.";
            throw new Exception(error);
        }
    }
}
}

```

## CLASE RegistroSugerenciaAction.java

```

package com.ug.cisc.erp.marketing.Action;
import java.sql.Connection;
import java.sql.PreparedStatement;
import javax.sql.DataSource;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class RegistroSugerenciaAction extends Action {
    private ConnectionPool pool ;
    public RegistroSugerenciaAction() {
        pool = ConnectionPool.getInstance();
    }
    public ActionForward execute(
        ActionMapping mapping,
        ActionForm form,
        HttpServletRequest request,
        HttpServletResponse response)
        throws Exception {
        if (isCancelled(request)){

            return mapping.findForward("Welcome");
        }
        RegistroSugerenciaForm sugerForm = (RegistroSugerenciaForm)
form;

        Connection conn = pool.getConnection();
        PreparedStatement statement = conn.prepareStatement(
            "insert into MAR_SUGERENCIA_TMP" +
            "(NOMBRE,EMPRESA,EMAIL,PAIS,CIUDAD,COMENTARIO)" +
            " values (?, ?, ?, ?, ?, ?)");
        statement.setString(1,sugerForm.getNombre());
        statement.setString(2,sugerForm.getEmpresa());
        statement.setString(3,sugerForm.getE_mail());
        statement.setString(4,sugerForm.getPais());
        statement.setString(5,sugerForm.getCiudad());
        statement.setString(6,sugerForm.getComentario());
        statement.executeUpdate();
        return mapping.findForward("success");
    }
}

```

## CLASE RegistroSugerenciaForm.java

```

package com.ug.cisc.erp.marketing.Action;
import javax.servlet.http.HttpServletRequest;
import com.ug.cisc.erp.marketing.entity.*;
import org.apache.struts.action.ActionError;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
import java.util.*;
public class RegistroSugerenciaForm extends ActionForm {
    private String nombre;
    private String empresa;
    private String e_mail;
    private String pais;
    private String ciudad;
    private String comentario;
    private boolean registrado;
    public boolean isRegistrado() {
        return registrado;
    }
    public void setRegistrado(boolean registrado) {
        this.registrado = registrado;
    }
    public String getCiudad() {
        return ciudad;
    }
    public ActionErrors validate(ActionMapping mapping, HttpServletRequest
request) {
        ActionErrors errors = new ActionErrors();
        if (nombre==null
            || nombre.trim().equals("")){
            errors.add("nombre",new
ActionError("registroSugerencia.error.nombre"));
        }
        if (empresa==null
            || empresa.trim().equals("")){
            errors.add("empresa",new
ActionError("registroSugerencia.error.empresa"));
        }
        if (e_mail==null
            || e_mail.trim().equals("")){
            errors.add("e_mail",new
ActionError("registroSugerencia.error.e_mail"));
        }
        if (pais==null
            || pais.trim().equals("")){
            errors.add("pais",new
ActionError("registroSugerencia.error.pais"));
        }
        if (ciudad==null
            || ciudad.trim().equals("")){
            errors.add("ciudad",new
ActionError("registroSugerencia.error.ciudad"));
        }
        if (comentario==null
            || comentario.trim().equals("")){
            errors.add("comentario",new
ActionError("registroSugerencia.error.comentario"));
        }
        return errors;
    }
}

```

## CLASE RegistroSuscripcionAction.java

```

package com.ug.cisc.erp.marketing.Action;
import java.sql.Date;
import java.sql.Connection;
import java.sql.PreparedStatement;
import javax.sql.DataSource;
import java.util.Calendar;
import java.util.ArrayList;
import com.ug.cisc.erp.marketing.dao.util.ConnectionPool;
public class RegistroSuscripcionAction extends Action {
    private ConnectionPool pool ;
public RegistroSuscripcionAction() {
    pool = ConnectionPool.getInstance();
}
public ActionForward execute(
    ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws Exception {
    if (isCancelled(request)){
        return mapping.findForward("Welcome");
    }
    RegistroSuscripcionForm suscripcionForm = (RegistroSuscripcionForm)
form;
    String fecha = "";
    String cod_stat="A";
    int cod=1;
    Connection conn = pool.getConnection();
    PreparedStatement statement = conn.prepareStatement(
        "insert into mar_suscriptor_dat" +
        "(nomb_susc,apel_susc,cedu_susc,sexo_susc," +
"esta_civi_susc,fech_naci_susc,dire_susc,e_mail_susc,fono_susc," +
"empr_susc,codi_ciud_fk,codi_prov_fk,codi_pais_fk,codi_stat_fk,codi_empr_fk)" +
        " values (?,?,?,?,?,?,?,?,?,?,?,?,?,?)");
    fecha = suscripcionForm.getAnio_nac() + "-" +
suscripcionForm.getMes_nac()+ "-" + suscripcionForm.getDia_nac();
    statement.setString(1,suscripcionForm.getNomb_susc());
    statement.setString(2,suscripcionForm.getApel_susc());
    statement.setString(3,suscripcionForm.getCedu_susc());
    statement.setString(4,suscripcionForm.getSexo_susc());
    statement.setString(5,suscripcionForm.getEsta_civi_susc());
    statement.setString(6,fecha);
    statement.setString(7,suscripcionForm.getDire_susc());
    statement.setString(8,suscripcionForm.getE_mail_susc());
    statement.setString(9,suscripcionForm.getFono_susc());
    statement.setString(10,suscripcionForm.getEmpr_susc());
    statement.setInt(11,cod);
    statement.setInt(12,cod);
    statement.setInt(13,cod);
    statement.setString(14,cod_stat);
    statement.setInt(15,cod);
    statement.executeUpdate();
    return mapping.findForward("success");
}
}

```

## CLASE RegistroSuscripcionForm.java

```

package com.ug.cisc.erp.marketing.Action;
public class RegistroSuscripcionForm extends ActionForm {
    private Integer codi_susc_pk ;
    private String nomb_susc;
    private String apel_susc;
    private String cedu_susc;
    private String sexo_susc;
    private String esta_civi_susc;
    private Date fech_naci_susc;
    private String fecha;
    private Integer dia_nac;
    private Integer mes_nac;
    private Integer anio_nac;
    private String dire_susc;
    private String e_mail_susc;
    private String fono_susc;
    private String empr_susc;
    private int codi_ciud_fk;
    private int codi_prov_fk;
    private int codi_codi_pais_fk;
    private String codi_stat_fk;
    private int codi_empr_fk;
    private boolean registrado;
    public String getApel_susc() {
        return apel_susc;
    }
    public void setApel_susc(String apel_susc) {
        this.apel_susc = apel_susc;
    }
    public String getCedu_susc() {
        return cedu_susc;
    }
    public void reset(ActionMapping mapping, HttpServletRequest request) {
        codi_susc_pk=null;
        nomb_susc=null;
        apel_susc=null;
        cedu_susc=null;
        sexo_susc=null;
        esta_civi_susc=null;
        fech_naci_susc=null;
        dire_susc=null;
        e_mail_susc=null;
        fono_susc=null;
        empr_susc=null;
        codi_stat_fk=null;
    }
    ActionError("registroSuscripcion.error.dire_susc");
    }
    if (e_mail_susc==null
        || e_mail_susc.trim().equals("")){
        errors.add("e_mail_susc",new
    }
    return errors;
    }
    public String getFecha() {
        return fecha;
    }
    public void setFecha(String fecha) {
        this.fecha = fecha;
    }
}

```