

MANUAL TECNICO DE CODE N BUGS

by Juan Pablo Orizabal Gil
Carné: 201730318

MANUAL TÉCNICO

El software fue hecho, probado y ejecutado en el IDE Netbeans, es posible usarse con Netbeans 8.1 o superior y es aplicable para cualquier equipo que tenga una memoria RAM de 2 gb minimo.

El software es un gestor de proyectos, esto quiere decir que hay jerarquias entre los usuarios que existen y el fin es crear proyectos, asignar estos proyectos a administradores de proyectos, con esto crear casos y estos asignarles etapas que seran hechos por desarrolladores.

PAQUETE ENTIDADES

Esta clase contiene todas las entidades utilizadas en la base de datos.

CLASE ACCION.

Esta clase contiene varios atributos y sus respectivos setters y getters.

Asi como su constructor.

Atributos:

nombreUsuario

iDProyecto

iDCaso

iDEtapa

faseProyecto

tipoAccion

comentario

aprobado

CLASE CASO

Esta clase contiene varios atributos y sus respectivos setters y getters.

Asi como su constructor.

Atributos:

id

nombre

fechaLimite

idProyecto

porcentajeAvance

activo

tipo

CLASE CONTROL VECES

Esta clase contiene varios atributos y sus respectivos setters y getters.

Asi como su constructor.

Atributos:

nombre

veces

CLASE ETAPA

Esta clase contiene varios atributos y sus respectivos setters y getters.
Asi como su constructor.

Atributos:

nombre
desarrolladorACargo
activo
tiempoEnHoras
aprobado
fechaLimite
idCaso
total

CLASE PROYECTO

Esta clase contiene varios atributos y sus respectivos setters y getters.
Asi como su constructor.

Atributos:

id
nombre
fechaLimite
activo
administradorProyecto

CLASE TIPO CASO

Esta clase contiene varios atributos y sus respectivos setters y getters.
Asi como su constructor.

Atributos:

nombre
etapas

CLASE USUARIO

Esta clase contiene varios atributos y sus respectivos setters y getters.
Asi como su constructor.

Atributos:

usuario
contraseña
nombre
apelllido

dpi
fechaIngreso
dineroHora
tipoUsuario

PAQUETE DE INTERFAZ GRAFICA

CLASE ACCIONES REALIZADAS

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

agregarColumnas: agrega las columnas respectivas

agregarDatos: agrega los datos a las columnas

CLASE AREA ADMINISTRADOR PROYECTO

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

agregarProyectos: agrega los proyectos a la tabla

borrarDatosTabla: borra los datos de las tablas

verificarEtapa: verifica una etapa

cancelarEtapa: cancela una etapa

cancelarCaso: cancela un caso

consultarProyecto: consulta la informacion de un proyecto

consultarCaso: consulta la informacion de un caso

cambiarFechaLimite: cambia la fecha limite a un caso

CLASE AREA ADMINISTRADOR SISTEMA

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

agregarProyectos: agrega los proyectos a la tabla

borrarDatosTabla: borra los datos de los proyectos

crearCaso: crea un caso

editarProyecto: edita la informacion de un proyecto

ocultarCreacionCaso: oculta la creacion de un caso
mostrarCreacionCaso: muestra la creacion de un caso
consultarProyecto: consulta un proyecto
cambiarAdministradorProyecto: cambia de administrador de proyecto

CLASE AREA DESARROLLADOR

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

agregarFechas: agrega las fechas
crearCaso: crea un caso
terminarTrabajo: termina un trabajo de un desarrollador
agregarColumnas: agrega las columnas a los modelos de las tablas

CLASE ASIGNACION ETAPA

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

asignarDesarrolladores: asigna los desarrolladores
asignarEtapa: asigna las etapas a un caso

CLASE ASIGNAR CASO

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

asignarDatos: asigna los datos a un caso
asignarEtapa: asigna las etapas a un caso
asignarCaso: asigna los nombres de los casos

CLASE CASOS ASIGNADOS

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

asignarValores: asigna valores a las tablas

asignarDatos: asigna los datos a los modelos

CLASE CREACION PROYECTO

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

agregarDatosLista: agrega los datos a las listas y modelos de un proyecto

crearProyecto: crea un proyecto

CLASE CREACION USUARIO

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

verificarUsuario: verifica la informacion de un usuario que quiere ingresar al sistema

CLASE CREAR ETAPA

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

crearEtapa: crea una etapa

CLASE INICIO SESION

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

validarUsuario: valida un usuarios

CLASE MENU PRINCIPAL

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

agregarFrame: agrega al frame el internal frame que se esta creando

CLASE REPORTES

Esta clase contiene varios atributos entre los que se encuentran modelos de tablas.

Metodos:

irReporte: va al reporte que se esta pidiendo

exportarHTML: exporta a html el reporte pertinente

reporteProyectos: reporta un proyecto y le asigna valores

mostrarFiltros: muestra los filtros pertinentes

asignarReportes: asigna valores a los reportes

asignarTotal: asigna el total al label que lo pide

agregarModelos: agrega los datos a los modelos a utilizar

PAQUETE LOGICA

Este paquete contiene clases de java normales que serviran para poder cumplir con la logica requerida del programa

CLASE HTML

Esta clase contiene todos los métodos necesarios para poder crear el lenguaje de etiquetado html

Metodos

usarFileChooser: crea un filechooser para poder elegir donde guardar el archivo

decirQueSeGeneroElArchivo: dice que se genero el archivo

generarTitulo: genera el titulo a un reporte

generarReporte0: genera el reporte 0

generarReporte1: genera el reporte 1

generarReporte2: genera el reporte 2

generarReporte3: genera el reporte 3

generarReporte4:genera el reporte 4
generarReporte5:genera el reporte 5
generarReporte6:genera el reporte 6
generarReporte7y8:genera el reporte 7 y 8
generarReporte9EnAdelante: :genera el reporte 9 en adelante a excepcion del 12
generarReporte12:genera el reporte 12

CLASE MOSTRAR INFORMACION

Esta clase contiene las cosas necesarias para poder mostrar la informacion requerida.

Metodos

mostrarInformacionCaso: muestra la informacion de un caso especifico
consultaProyectoCasos: consulta los casos de un proyecto especifico
agregarFechasLimites: agrega las fechas limites a las tablas de la interfaz
mostrarAcciones: muestra las acciones realizadas
usarEnArreglo: crea un arreglo de desarrolladores
transformarArrayListEnObject: convierte un array list de string en un arreglo de objetos
cambiarAdministrador: cambia a un administrador de proyecto
reporteProyectoActivo: crea el reporte de proyecto activo
cantidadCasos: devuelve la cantidad de casos de un proyecto
reporteDineroHorasProyecto: crea el reporte de horas y dinero de un proyecto especifico
reporteDineroHorasDesarrollador:crea el reporte de horas y dinero de un desarrollador especifico
reporteDineroHorasTipo:crea el reporte de horas y dinero de un tipo especifico
reporteDesarrollador: reporte de un desarrollador
reporteDesarrolladorMasCasos: reporte del desarrollador que mas ha participado en casos
reporteDesarrolladorMasDinero: reporte del desarrollador que mas se ha gastado dinero
reporteProyectoMasCasosFinalizados: reporte del proyecto que mas casos tiene finalizados

mostrarCasoProyectoCasos: reporte del proyecto que mas casos tiene cancelados

reporteProyectoMasCasosCancelados: reporte del proyecto que mas casos tiene cancelados

CLASE SESION

Esta clase contiene las cosas necesarias para poder cumplir la logica planteada

Metodos

verificarUsuario: verifica si existe el usuario

agregarFrame: agrega al frame una u otra interfaz

limpiar: limpia todo lo que exista en el escritorio actual

validarEspaciosUsuario: verifica los datos del usuario

validarUsuario: valida al usuario que quiere ingresar al sistema

crearUsuario: crea un usuario

arreglo reportes: crea un arreglo de reportes

Luego tiene una serie de getters que son nada mas que constantes de Strings.

CLASE TABLA MODELO

Esta clase contiene unicamente un método que es para poder hacer que no se puedan editar las tablas, esta clase hereda de DefaultTableModel

Metodos

isCellEditable: solo le modifiko para que ninguna celda se pueda modificar mas si seleccionar

CLASE VALIDACION CASO

Esta clase contiene las cosas necesarias para poder cumplir la logica planteada de un caso

Metodos

agregarProyectos: agrega los proyectos

verificarInformacionCaso: verifica la informacion de un caso

crearTipoCaso: crea un tipo de caso

asignarDatosTablaCasos: asigna los datos a las tablas casos

agregarProyectosTabla: agrega los datos a la tabla proyectos

verificarDatosCreacionCaso: verifica los datos de la creacion de un caso

crearCasos: crea un caso

asignarDatosListaDesarrolladores: asigna los datos de los desarrolladores que existen
asignarDatosCasos: asigna datos a la tabla de casos
idCaso: verifica el id del caso
cancelarCaso: cancela un caso
mandarComentario: abre un JOptionPane que pide al usuario el comentario a mandar
agregarAccionCancelacionCaso: agrega la accion de la cancelacion de un caso
asignarValoresAccionDesarrollador: asigna valores a las acciones de los desarrolladores
agregarCasosAsignados: agrega los datos de los casos asignados

CLASE VALIDACION ETAPA

Esta clase contiene las cosas necesarias para poder cumplir la logica planteada de un caso

Metodos

crearEtapa: crea una etapa
asignarDatosEtapa: asigna los datos a una etapa
idEtapa: verifica el id de la etapa
nombreEtapa: asigna el nombre de una etapa
asignarEtapasDesarrolladores: asigna las etapas a los desarrolladores
finalizacionCaso: finaliza un caso, y manda el comentario al administrador de proyecto
registrarHoras: registra las horas que se tardo el desarrollador
idProyectoDesdeEtapa: verifica el id del proyecto desde la etapa
agregarDatosEtapasFinalizadas: agrega los datos a las etapas que se aprobaron
agregarTotalTiempo: agrega el total de tiempo que se tardo a la etapa
agregarTotal: agrega en cuanto salio el caso en contexto de dinero
actualizarEstadoAccion: actualiza el estado de la accion a que ya se aprobo
siguienteEtapa: abre la interfaz para asignar la siguiente etapa del caso a un desarrollador especifico
asignarEtapaNueva: asigna los datos a la etapa nueva
actualizarPorcentajeAvance: actualiza el porcentaje de avance de un proyecto
introducirPorcentaje: introduce el porcentaje de avance

cancelarAccion: cancela una accion

cancelarEtapa: cancela una etapa.

revisarAccion: revisa una accion

mandarComentario: manda el comentario que se necesita al realizar algunas acciones

CLASE VALIDACION PROYECTO

Esta clase contiene las cosas necesarias para poder cumplir la logica planteada de un caso

Metodos

crearProyecto: crea un proyecto

agregarDatosLista: agrega los datos de un proyecto

agregarDatosTablaProyectos: agrega los datos a las tablas de proyecto

borrarDatos: borra los datos de las tablas del administrador de sistema

editarActivoProyecto: edita si un proyecto esta activo o no

verificarProyectoEtapa: verifica la etapa basado en el id de un proyecto

verificarProyectoCaso: verifica un caso basado en el id de un proyecto

preguntarFecha: pregunta la fecha que se desea cambiar

cambiarFecha: cambia la fecha

PAQUETE CODE N BUGS

CLASE CODE N BUGS

Esta clase unicamente contiene el main que es donde va a empezar un proyecto

ANALISIS DE REQUERIMIENTOS

Se requiere un sistema que pueda tener una jerarquia establecida asi como la creacion de diferentes etapas, basicamente se necesita que el software pueda crear proyectos, crear tipos de casos, crear casos y asignar estos casos a desarrolladores, asi como la implementacion de etapas que estan vinculadas con los casos y los tipos de casos, asi de esa forma hacer que los desarrollares tengan que trabajar en estas etapas dentro de una fecha limite, luego de esto que los administradores de proyecto revisen el trabajo realizado por dicho desarrollador y aceptar o no la etapa, si se acepta la etapa se procede con la siguiente y asi hacer que se lleve un flujo las etapas de un caso hasta la finalizacion del mismo. Esto es a rasgos generales, luego existen distintas operaciones un poco mas especificas que se entrara en detalle mas adelante.

Se necesita crear usuarios, que tengan: nombre, apellido, usuario, contraseña, dinero por hora que cobra, tipo de usuario que va a ser. Esto servira para que puedan tener acceso a las distintas partes del sistema y la base de datos.

Se necesita crear proyectos, para esto solo se necesitara darle un nombre, y un id propio.

Se necesita crear tipos de casos, especificando un nombre y el numero de etapas que va a tener

Se necesita crear casos, especificando el tipo, el nombre y la fecha limite de este caso, asi de esta forma asignar el desarrollador de la primera etapa.

Se necesita crear etapas con un nombre, un identificador unico, y la fecha limite que tendra esta etapa.

Al finalizar una etapa se necesita un mecanismo que haga que un desarrollador introduzca un comentario y el tiempo que tardo en hacerlo.

Para luego hacer que se pueda validar el trabajo hecho por el desarrollador en la interfaz del administrador de proyecto.

Se necesita poder visualizar toda la informacion de la base de datos si es un administrador de sistema, y todo los proyectos asignados en los que esta involucrado un usuario si este esta en el area de administracion de proyectos, asi como los casos a los que esta asignado un desarrollador.

Se debe tener la opcion de desactivar un proyecto

Se debe tener la opcion de cancelar un caso.

Se debe tener la opcion de cambiar la fecha limite de un proyecto

Se debe tener la opcion de cambiar la fecha limite de un caso

Se debe poder visualizar las acciones que se registren en el proyecto

Basicamente esto seria todo los requerimientos que se necesitan.

DIAGRAMA DE TABLAS

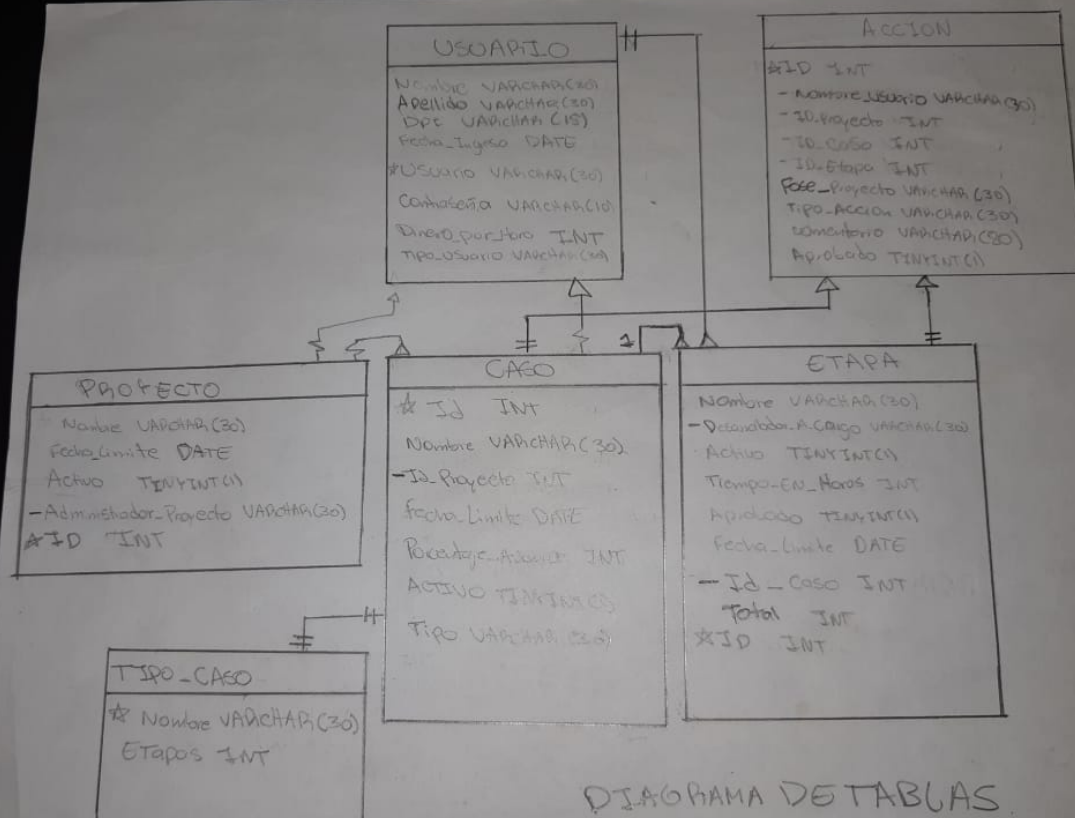
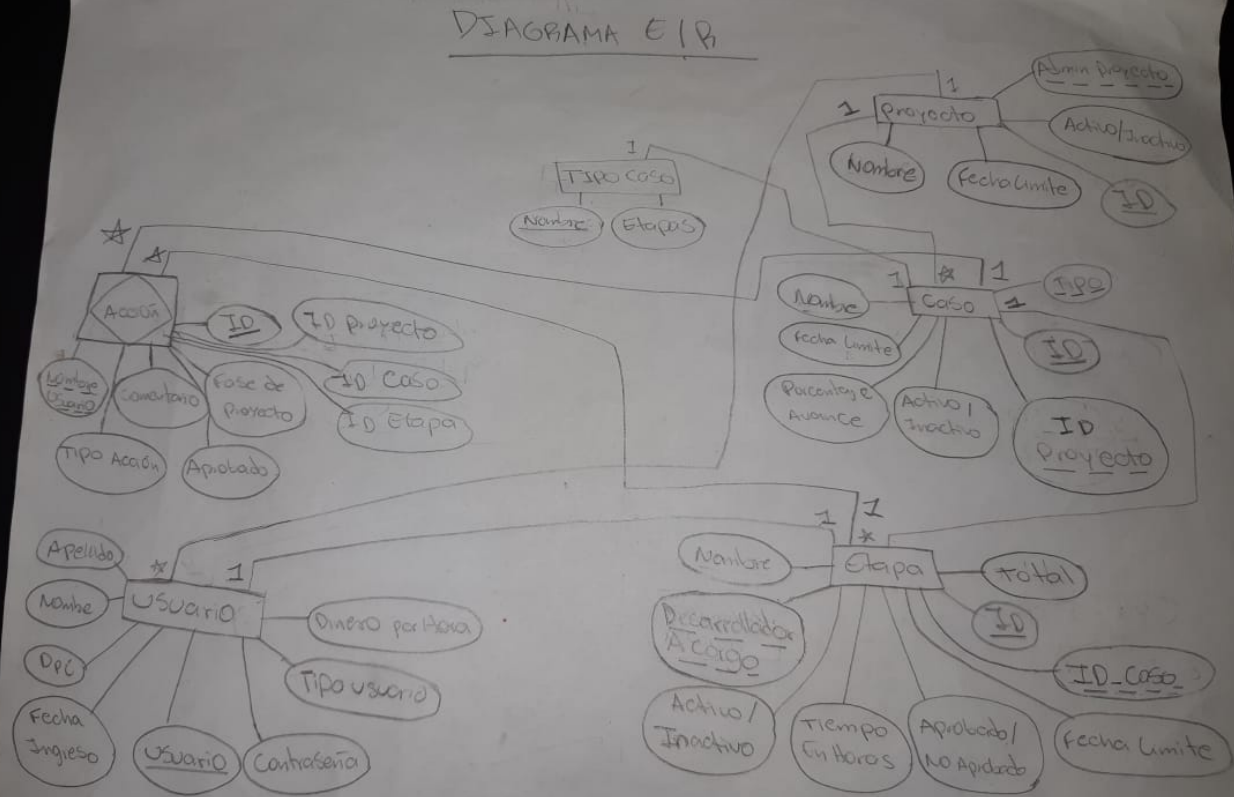


DIAGRAMA DE TABLAS

DIAGRAMA ENTIDAD RELACION



MAPEO FISICO DE LA DB

```
CREATE SCHEMA CODE_N_BUGS;  
USE CODE_N_BUGS;
```

```
CREATE TABLE USUARIO(  
  Usuario VARCHAR(30) NOT NULL,  
  Contraseña VARCHAR(10) NOT NULL,  
  Nombre VARCHAR(30) NOT NULL,  
  Apellido VARCHAR(30) NOT NULL,  
  Dpi VARCHAR(15) NOT NULL,  
  Fecha_Ingreso DATE NOT NULL,  
  Dinero_Hora INT NOT NULL,  
  Tipo_Usuario VARCHAR(30) NOT NULL,  
  CONSTRAINT PK_USUARIO PRIMARY KEY(Usuario)  
);
```

```
CREATE TABLE PROYECTO(  
  Id INT NOT NULL AUTO_INCREMENT,  
  Nombre VARCHAR(30) NOT NULL,  
  Fecha_Limite DATE NOT NULL,  
  Activo TINYINT(1) NOT NULL,  
  Administrador_Proyecto VARCHAR(30) NOT NULL,  
  CONSTRAINT PK_PROYECTO PRIMARY KEY(Id),  
  CONSTRAINT FK_USUARIO_IN_ADMINISTRADOR_PROYECTO  
  FOREIGN KEY (Administrador_Proyecto) REFERENCES  
  USUARIO(Usuario)  
);
```



```
CREATE TABLE CASO(  
Id INT NOT NULL AUTO_INCREMENT,  
Nombre VARCHAR(30) NOT NULL,  
Id_Proyecto INT NOT NULL,  
Fecha_Limite DATE NOT NULL,  
Porcentaje_Avance INT,  
Activo TINYINT(1),  
Tipo VARCHAR(30),  
CONSTRAINT PK_CASO PRIMARY KEY(Id),  
CONSTRAINT FK_PROYECTO_IN_ID_PROYECTO FOREIGN KEY  
(Id_Proyecto) REFERENCES PROYECTO(Id)  
);
```

```
CREATE TABLE ETAPA(  
Id INT NOT NULL AUTO_INCREMENT,  
Nombre VARCHAR(30) NOT NULL,  
Desarrollador_A_Cargo VARCHAR(30),  
Activo TINYINT(1),  
Tiempo_En_Horas INT,  
Aprobado TINYINT(1),  
Fecha_Limite DATE,  
Id_Caso INT NOT NULL,  
Total INT,  
CONSTRAINT PK_PASO PRIMARY KEY(Id),  
CONSTRAINT FK_USUARIO_IN_DESARROLLADOR_A_CARGO  
FOREIGN KEY (Desarrollador_A_Cargo) REFERENCES  
USUARIO(Usuario),  
CONSTRAINT FK_CASO_IN_ID_CASO FOREIGN KEY (Id_Caso)  
REFERENCES CASO(Id)  
);
```

```
CREATE TABLE TIPO_CASO(  
Nombre VARCHAR(30) NOT NULL,  
Etapas INT NOT NULL,  
CONSTRAINT PK_TIPO_CASO PRIMARY KEY(Nombre)  
);
```

```
CREATE TABLE ACCION(  
Id INT NOT NULL AUTO_INCREMENT,  
Nombre_Usuario VARCHAR(30) NOT NULL,  
Id_Proyecto INT NOT NULL,  
Id_Caso INT,  
Id_Etapa INT,  
Fase_Proyecto VARCHAR(30) NOT NULL,  
Tipo_Accion VARCHAR(30) NOT NULL,  
Comentario VARCHAR(80) NOT NULL,  
Aprobado TINYINT(1),  
Revisada TINYINT(1),  
CONSTRAINT PK_ACCION PRIMARY KEY(Id),  
CONSTRAINT FK_PROYECTO_IN_ID_PROYECTO_2 FOREIGN KEY  
(Id_Proyecto) REFERENCES PROYECTO(Id),  
CONSTRAINT FK_CASO_IN_ID_CASO_2 FOREIGN KEY (Id_Caso)  
REFERENCES CASO(Id),  
CONSTRAINT FK_ETAPA_IN_ID_ETAPA_2 FOREIGN KEY (Id_Etapa)  
REFERENCES ETAPA(Id),  
CONSTRAINT FK_USUARIO_IN_NOMBRE_USUARIO_2 FOREIGN  
KEY (Nombre_Usuario) REFERENCES USUARIO(Usuario)  
);
```

```
INSERT INTO  
USUARIO(Usuario,Contraseña,Nombre,Apellido,Dpi,Fecha_Ingreso,Dinero  
_Hora,Tipo_Usuario) VALUES  
( 'admin','1','administrador','base','123456789','2018-09-03',300,'administrador  
sistema');
```

DIAGRAMA DE CLASES

