

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL



**PROGRAMA DE TECNOLOGÍA EN COMPUTACIÓN
Y DISEÑO GRÁFICO Y PUBLICITARIO**

PROYECTO DE GRADUACIÓN

**PREVIO A LA OBTENCIÓN DEL TÍTULO DE
ANALISTA DE SOPORTE DE MICROCOMPUTADORES**

**TEMA:
SISTEMA DE CONTROL DE ASISTENCIA DE
PERSONAL "EAGLEEYES"**

MANUAL DE DISEÑO

**AUTORES:
NARANJO MORÁN WENDY NATALIA
QUIROGA MIELES CARLOS ANTONIO
PINTO ROBAYO MARCELO RAÚL**

**DIRECTOR:
ANL. YAMILL LAMBERT**

AÑO 2006



AGRADECIMIENTO

Agradecemos primeramente a Jehová Dios, sobre todas las cosas que con su infinita bondad y misericordia ha puesto en nosotros sus bendiciones. A nuestros amorosos Padres que siempre nos han estado apoyando incondicionalmente para alcanzar una de nuestras metas en la vida, el incorporarnos de Analista de Soporte de Microcomputadores.

Al Anl. Yamill Lambert que nos ha capacitado excelentemente con mucha paciencia y siempre ha estado disponible para cualquier duda o inquietud.

DEDICATORIA

Dedicamos este trabajo y esta dirigido especialmente, a las personas que siempre nos han colaborado, han confiado y estuvieron apoyándonos siempre con ahínco en todo lo necesario quienes son nuestros Padres. A nuestro director del tópico y a todos aquellos amigos y compañeros que hicimos a lo largo de la carrera.

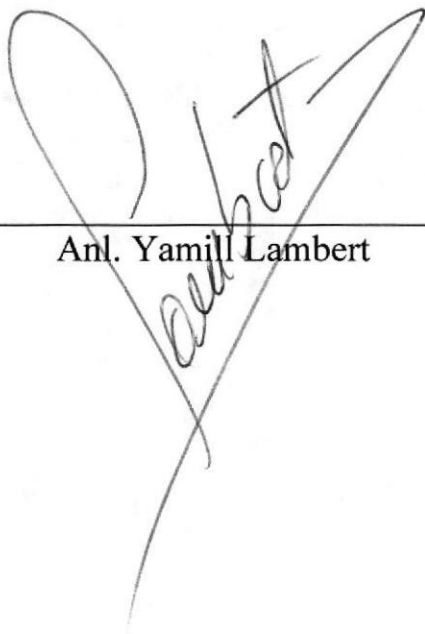


DECLARACIÓN EXPRESA

La responsabilidad por los hechos, ideas y doctrinas expuestas en este Tópico de Graduación nos corresponde exclusivamente; y el patrimonio intelectual de la misma al PROTCOM (*Programa de Tecnología en Computación y Diseño Gráfico*) de la Escuela Superior Politécnica del Litoral.

(Reglamentos de Exámenes y Títulos Profesionales de la ESPOL).

FIRMA DEL DIRECTOR DEL TÓPICO DE GRADUACIÓN



Anl. Yamill Lambert



AUTORES DEL PROYECTO

Wendy M. Naranjo Morán
Wendy Naranjo Morán

Carlos Quiroga Mieles
Carlos Quiroga Mieles

Marcelo Pinto Robayo
Marcelo Pinto Robayo

TABLA DE CONTENIDO



CAPÍTULO	PÁGINA
1. ANTECEDENTES	1
2. DEFINICIÓN DEL PROBLEMA ACTUAL	1
2.1 DEFINICIÓN DE PROBLEMAS	1
3. SOLUCIÓN PROPUESTA Y ALCANCE	1
3.1. ALCANCE DEL SISTEMA PROPUESTO	1
4. DIAGRAMA GENERAL DEL SISTEMA	1
5. DIAGRAMAS DE CASOS DE USO	1
6. DIAGRAMA DE TABLAS Y CLASES	1
6.1. DIAGRAMA GENERAL	1
6.2. DIAGRAMAS DE CLASE CLASIFICADOS POR CAPAS	2
6.2.1. <i>CAPA DATOS</i>	2
6.2.1.1 CLASE DATCARGO	2
6.2.1.2 CLASE DATCARGO	3
<i>DESCRIPCIÓN DE CLASE</i>	3
6.2.1.3 CLASE DATASISTENCIA	4
6.2.1.4 CLASE DATCIUDAD	5
6.2.1.5 CLASE DATCIUDAD	6
6.2.1.6 CLASE DATDEPARTAMENTO	7
6.2.1.7 CLASE DATDEPARTAMENTO	8
6.2.1.8 CLASE DATEMPRESA	9
6.2.1.9 CLASE DATEMPRESA	10
6.2.1.10 CLASE DATINCENTIVO	11
6.2.1.11 CLASE DATINCENTIVO	12
6.2.1.12 CLASE DATJUSTIFICA	13
6.2.1.13 CLASE DATJUSTIFICA	14
6.2.1.14 CLASE DATMULTA	15
6.2.1.15 CLASE DATMULTA	16
6.2.1.16 CLASE DATPERSONAL	17
6.2.1.17 CLASE DATPERSONAL	18
6.2.1.18 CLASE DATPERSONAL	19
6.2.1.19 CLASE DATPROFESION	20
6.2.1.20 CLASE DATPROFESION	21
6.2.1.21 CLASE DATTIPOINCENTIVO	22
6.2.1.22 CLASE DATTIPOINCENTIVO	23
6.2.1.23 CLASE DATTIPOJUSTIFICA	24
6.2.1.24 CLASE DATTIPOJUSTIFICA	25
6.2.1.25 CLASE DATTIPOMULTA	26
6.2.1.26 CLASE DATTIPOMULTA	27
6.2.1.27 CLASE DATTURNOSHORARIOS	28

6.2.1.28	CLASE DATTURNOSHORARIOS	29
6.2.1.29	CLASE DATUSUARIO	30
6.2.1.30	CLASE DATUSUARIO	31
6.2.2.	<i>CAPA CLIENTES</i>	32
6.2.2.1	CLASE CLSMANTENIMIENTOTIPO1	32
6.2.2.2	CLASE CLSMANTENIMIENTOTIPO1	33
6.2.2.3	CLASE CLSMANTENIMIENTOTIPO2	34
6.2.2.4	CLASE CLSMANTENIMIENTOTIPO3	35
6.2.2.5	CLASE CLSMANTENIMIENTOTIPO3	36
6.2.2.6	CLASE CLSPRELIMINAR	37
6.2.2.7	CLASE FRMASISTENCIA	38
6.2.2.8	CLASE FRMASISTENCIA	39
6.2.2.9	CLASE FRMCARGO	40
6.2.2.10	CLASE FRMCIUDAD	41
6.2.2.11	CLASE FRMDEPARTAMENTO	42
6.2.2.12	CLASE FRMEMPRESA	43
6.2.2.13	CLASE FRMINCENTIVO	44
6.2.2.14	CLASE FRMINCENTIVO	45
6.2.2.15	CLASE FRMINCENTIVO	46
6.2.2.16	CLASE FRMJUSTIFICA	47
6.2.2.17	CLASE FRMJUSTIFICA	48
6.2.2.18	CLASE FRMJUSTIFICA	49
6.2.2.19	CLASE FRMMULTA	50
6.2.2.20	CLASE FRMMULTA	51
6.2.2.21	CLASE FRMMULTA	52
6.2.2.22	CLASE FRMPERSONAL	53
6.2.2.23	CLASE FRMPERSONAL	54
6.2.2.24	CLASE FRMPERSONAL	55
6.2.2.25	CLASE FRMPROFESION	56
6.2.2.26	CLASE FRMTIPOINCENTIVO	57
6.2.2.27	CLASE FRMTIPOJUSTIFICA	58
6.2.2.28	CLASE FRMTIPOMULTA	59
6.2.2.29	CLASE FRMTURNOSHORARIOS	60
6.2.2.30	CLASE FRMTURNOSHORARIOS	61
6.2.3.	<i>CAPA SERVIDOR</i>	62
6.2.3.1	CLASE clsCONEXION	62
6.2.3.2	CLASE cnxPERSONAL	63
6.2.3.3	CLASE cnxPERSONAL	64
6.2.3.4	CLASE cnxEMPRESA	65
6.2.3.5	CLASE cnxUSUARIO	66
6.2.3.6	CLASE cnxDEPARTAMENTO	67
6.2.3.7	CLASE cnxCiudad	68
6.2.3.8	CLASE cnxAsistencia	69
6.2.3.9	CLASE cnxTurnosHorarios	70

7. DIAGRAMAS DE SECUENCIAS 1

7.1.	PROCESO DE INGRESO DEL PERSONAL	1
7.2.	PROCESO DE INGRESO DE PERSONAL CON CÓDIGO NO EXISTENTE	3



BIBLIOTECA
CAMPUS
PEÑAS

7.3.	PROCESO DE SALIDA DEL PERSONAL	5
7.4.	PROCESO DE INGRESO DE INCENTIVO	7
7.5.	PROCESO DE INGRESO DE JUSTIFICATIVOS	9
7.6.	PROCESO DEL INGRESO DE MULTAS	11
8.	DIAGRAMAS DE CASOS DE COLABORACIÓN	1
9.	DIAGRAMAS DE ESTADO	1
9.1.	ESTADO DEL PERSONAL	1
9.2.	ESTADO DE EMPRESA	2
9.3.	ESTADO DE DEPARTAMENTOS	3
9.4.	ESTADO DE CIUDAD	4
9.5.	ESTADO DEL CARGO	5
9.6.	ESTADO DE ADMINISTRADOR	6
10.	ESTANDARIZACIONES	1
10.1.	FORMATOS DE ALMACENAMIENTO / INFORMACIÓN	1
10.1.1.	FORMATO PARA BASE DE DATOS	1
10.1.2.	FORMATO PARA EL NOMBRE DE FORMAS Y REPORTES	2
10.1.3.	FORMATO PARA EL NOMBRE DE LAS CLASES	3
10.2.	FORMATOS DE ESTÁNDARES DE PROGRAMACIÓN	4
10.2.1.	FORMATO PARA EL NOMBRE DE LOS CONTROLES	4
10.2.2.	FORMATO PARA EL NOMBRE DE LOS CAMPOS DE LA TABLA ..	5
10.3.	FORMATOS DE ESTÁNDARES PARA PRESENTACIONES DE PANTALLAS	5
10.3.1.	FORMATOS DE FONTS	5
10.3.2.	FORMATOS DE LOCALIZACIÓN DE BOTONES	5
10.3.3.	FORMATOS DE ICONOS	6
11.	PLAN DE CÓDIGOS	1
11.1.	TABLAS MAESTRAS	1
11.1.1.	TABLA CPm_Persona	1
11.1.2.	TABLA CPm_Cargo	1
11.1.3.	TABLA CPm_Departamento	2
11.1.4.	TABLA CPm_Profesion	2
11.1.5.	TABLA CPm_Tipo incentivo	3
11.1.6.	TABLA CPm_Tipo justificativo	3
11.1.7.	TABLA CPm_Tipo multa	4
11.1.8.	TABLA CPm_Ciudad	4
11.1.9.	TABLA CPm_Empresa	5
11.1.10.	TABLA CPm_Turno	5
11.1.11.	TABLA CPm_Horario	6
11.2.	TABLA HISTÓRICA	6
11.2.1.	TABLA CPh_Horariopersona	6
11.3.	TABLAS TRANSACCIONALES	7
11.3.1.	TABLA CPt_Asistencia	7
11.3.2.	TABLA CPt_Carnets	7
11.3.3.	TABLA CPt_Multa	8
11.3.4.	TABLA CPt_Incentivo	8



12. MODELOS ENTIDAD - RELACIÓN.....	1
12.1. MODELO LÓGICO	1
12.2. MODELO FÍSICO.....	2
13. DISEÑO DE ARCHIVOS	1
13.1. HISTORIAL DE HORARIO DE PERSONAL.....	1
13.2. MANTENIMIENTO DE PERSONA	2
13.3. MANTENIMIENTO DE CARGO.....	3
13.4. MANTENIMIENTO DE DEPARTAMENTO	4
13.5. MANTENIMIENTO DE CIUDAD	5
13.6. MANTENIMIENTO DE PROFESIÓN.....	6
13.7. MANTENIMIENTO DE EMPRESA	7
13.8. MANTENIMIENTO DE HORARIO	8
13.9. MANTENIMIENTO DE TURNO.....	9
13.10. MANTENIMIENTO DE TIPO DE INCENTIVO	10
13.11. MANTENIMIENTO DE TIPO DE JUSTIFICATIVO.....	11
13.12. MANTENIMIENTO DE TIPO DE MULTA	12
13.13. TRANSACCIONAL DE INCENTIVO	13
13.14. TRANSACCIONAL DE JUSTIFICATIVO.....	14
13.15. TRANSACCIONAL DE MULTAS	15
13.16. TRANSACCIONAL DE ASISTENCIA	16
13.17. TRANSACCIONAL DE CARNETS.....	17
14. DISEÑO DE ENTRADAS Y SALIDAS	1
14.1. DISEÑO DE ENTRADAS.....	1
14.1.1. MENÚ PRINCIPAL DEL SISTEMA.....	1
14.1.2. MENÚ MANTENIMIENTO	3
14.1.3. MENÚ PROCESOS	5
14.1.4. MENÚ CONSULTAS.....	7
14.1.5. MENÚ VER.....	10
14.1.6. MANTENIMIENTO PERSONA	12
14.1.7. MANTENIMIENTO DE EMPRESA.....	14
14.1.8. MANTENIMIENTO DE DEPARTAMENTOS	16
14.1.9. MANTENIMIENTO DE CARGO.....	18
14.1.10. MANTENIMIENTO DE TIPO DE JUSTIFICACIONES.....	20
14.1.11. MANTENIMIENTO DE TIPO DE MULTAS.....	22
14.1.12. MANTENIMIENTO DE TIPO DE INCENTIVO	24
14.1.13. MANTENIMIENTO DE PROFESION.....	26
14.1.14. MANTENIMIENTO DE CIUDADES.....	28
15. EXPLICATIVA DE MODELO DE CLASES Y HERENCIAS.....	1
15.1. DIAGRAMA GENERAL DE CLASES	1
15.2. DIAGRAMA DE CLASE clsCONSULTABASE	2
15.3. CLASE clsCONSULTABASE	3
15.4. NARRATIVA DE CLASE clsCONSULTABASE.....	4
15.5. DIAGRAMA DE CLASE clsMANTENIMIENTOBASICO	5
15.6. CLASE clsMANTENIMIENTOBASICO	6
15.7. CLASE clsMANTENIMIENTOBASICO	7
15.8. NARRATIVA CLASE clsMANTENIMIENTOBASICO	8
15.9. CLASE clsMANTENIMIENTOTIPO3	9

15.10.	CLASE CLSMANTENIMIENTO TIPO3	10
15.11.	CLASE CLSMANTENIMIENTO TIPO3	11
15.12.	NARRATIVA DE CLASE CLSMANTENIMIENTO TIPO3	12
15.13.	DIAGRAMA DE CLASE CLSMANTENIMIENTO TIPO1	13
15.14.	CLASE CLSMANTENIMIENTO TIPO1	14
15.15.	CLASE CLSMANTENIMIENTO TIPO1	15
15.16.	NARRATIVA DE CLASE CLSMANTENIMIENTO TIPO1	16
15.17.	DIAGRAMA DE CLASE CLSMANTENIMIENTO TIPO2	17
15.18.	CLASE CLSMANTENIMIENTO TIPO2	18
15.19.	NARRATIVA DE CLASE CLSMANTENIMIENTO TIPO2	18
16.	SEGURIDADES DEL SISTEMA	1
16.1.	NIVELES DE SEGURIDAD DE ACCESO Y USO	1
16.2.	SEGURIDADES DE DATOS	1
16.2.1.	<i>RESPALDO DE LA BASE DE DATOS</i>	<i>1</i>
16.2.2.	<i>RESTAURACIÓN</i>	<i>1</i>





CAPÍTULO 1

ANTECEDENTES

1. ANTECEDENTES

La unidad educativa LAS AMERICAS se fundo en 1998 en la ciudad de Guayaquil. Ubicada en la ciudadela LA SAIBA calle oeste 32 y Eloy Alfaro, se encuentra su matriz principal, base para la expansión de su primera sucursal ubicada en la parte Norte de la ciudad (alborada 7º etapa calle Robles y Av. San Jorge) y sede de todos sus controles.

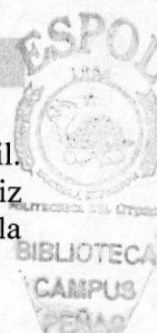
La unidad educativa LAS AMERICAS fue fundada por la fusión de capitales de dos socios visionarios, preocupados por ofrecer a la comunidad una educación de calidad a costos racionales para nuestro mercado.

LA UNIDAD EDUCATIVA LAS AMERICAS ES UNA UNIDAD ACADÉMICA QUE FORMA PROFESIONALES EN EL ÁREA RELACIONADA A INFORMÁTICA Y CONTABILIDAD A UN NIVEL SUPERIOR A TRAVÉS DE PROGRAMAS DE ESPECIFICACIÓN TÉCNICA TECNOLÓGICA.

Así también efectúa extensiones de estudio para el desarrollo de sus programas y prestar servicios a la comunidad.

La visión de la unidad educativa LAS AMERICAS es expandirse a la capital Quito para lograr una mayor cobertura con el nivel de educación impartida.

La Misión es impartir la mejor educación en sistemas de Información Computarizados a la Sociedad Ecuatoriana, apoyados en la tecnología disponible en el mundo de la computación, manteniendo los lineamientos y la filosofía de los socios fundadores.





CAPÍTULO 2

DEFINICIÓN DEL PROBLEMA ACTUAL



2. DEFINICIÓN DEL PROBLEMA ACTUAL

Gracias a la ayuda de los socios accionistas, se llegó a determinar lo siguiente:

La unidad educativa LAS AMERICAS cuenta con departamentos de Contabilidad, Secretariado, Cobranzas, Rectorado, Vicerrectorado, Ayuda Social y Recursos Humanos, departamento encargado de la contratación y control de todo el personal docente y administrativo. Los departamentos se manejan actualmente con sistemas de cómputo no apropiados para las funciones correspondientes, y nos enfocaremos en este proyecto a resolver la problemática del departamento de recursos humanos para el control de entrada y salida de su personal.

2.1 DEFINICIÓN DE PROBLEMAS

El departamento de recursos humanos para el control de entrada y salida de su personal presentan los siguientes problemas:

PROBLEMA	CAUSA	EFEECTO
Se lleva el control y registro de las salidas y entradas del personal en carpetas escritas a mano por una o mas personas autorizadas.	El sistema fue implantado a la rápida para solucionar un problema.	Frecuencia de errores en los reportes de control de asistencia.
Perdida de tiempo para encontrar datos de los ingresos a destiempo y sobre tiempo del personal	Base de datos -Excel- "en donde enlistan todos los ingresos registrados en papel.	El tiempo para realizar el traspaso es un tiempo valioso perdido.
Inconsistencia de frecuente de datos.	Manejo multiusuario no restringido a los registros en Excel. Cualquier persona autorizada (Excepto por una clave de archivo) tienen acceso a la base de datos de la empresa.	Al realizar cambios muy frecuentes de datos se genera una inconsistencia de los mismos en los reportes finales.
Traspaso manual de los registros que se llevan en carpetas escritas.	No existe el sistema de control de personal automatizado.	Causa que exista un total descontrol y parcialización de los controles de horarios al personal.
No se pueden emitir un control exacto por departamentos o por horas de retraso o sobre tiempo	Actualmente no se cuenta con un sistema de control de personal que permita dividir en los diferentes departamentos la carga de personal.	Se esta propenso a la confusión de personal por departamentos.

No existen las seguridades necesarias en el manejo de información.	No existe un sistema de verificación de usuarios o de acceso a los datos mas para la modificación de los reportes de personal, ya que los ingresos se generaran de manera automática por cada usuario.	Personal de distintos departamentos tenían acceso a información y manipulación de datos en los reportes de control de personal.
--	--	---



CAPÍTULO 3

SOLUCIÓN, PROPUESTA Y ALCANCE

3. SOLUCIÓN PROPUESTA Y ALCANCE

Soluciones:

De acuerdo a la fase de estudio realizada por el personal de desarrollo y verificando los problemas en los diferentes departamentos, enfocándonos a recursos humanos, se propone automatizar los siguientes procesos:

1. Historial del ingreso y egreso de personal.
2. Control de sobre tiempos por cada personal
3. Realizar un control detallado de horarios según sea la emisión de reportes solicitados por el departamento de Recursos Humanos.
4. Anulación o modificación de horarios, única y exclusivamente, por personal autorizado con clave de acceso restringida.
5. Reportes de control de ingresos y egresos por departamentos.
6. Registro de ingreso y egresos eventuales durante la jornada laboral.
7. Control detallado el personal contratado de manera eventual por la empresa.
8. Control de ingreso al sistema y/o a la información correspondiente.
9. Manejo de datos en una base segura y apropiada para la manipulación de los mismos.
10. Emisión de carnets de asistencia para el control de horarios al personal.

3.1. ALCANCE DEL SISTEMA PROPUESTO

BENEFICIOS.-

- Capaz de permitir al director de la empresa o gerente de recursos humanos, llevar un control sobre el ingreso y salidas del personal.
- Control de ingreso y egresos de personal por departamentos según requerimientos del departamento de recursos humanos.
- Procesos automáticos de registro de horarios de entrada y de salida sin intervención de segundos (Proceso automatizado empleado – sistema).
- Automatizar el control de ingreso de personal eventual (Asignación de carnets de ingreso de invitados controlado por el sistema).
- Llevar un control personalizado de la hoja de vida de salidas y entradas por cada uno de los miembros de la empresa.



- El personal anterior, encargado de llevar este control de manera manual, tendrá una mayor motivación en su ambiente de trabajo, ya que sus esfuerzos operacionales serán mínimos con la automatización.
- Un sistema de identificación eficiente; guardando los datos necesarios y requeridos, asignándole un código único de identificación por empleado.
- El sistema podrá actualizar módulos de control según se incrementen los requerimientos de la empresa.(Sistema con nivel de crecimiento)
- Se puede emitir (opcional) pre-nominas para el pago correspondiente al mes transcurrido del personal por departamentos o de manera total.

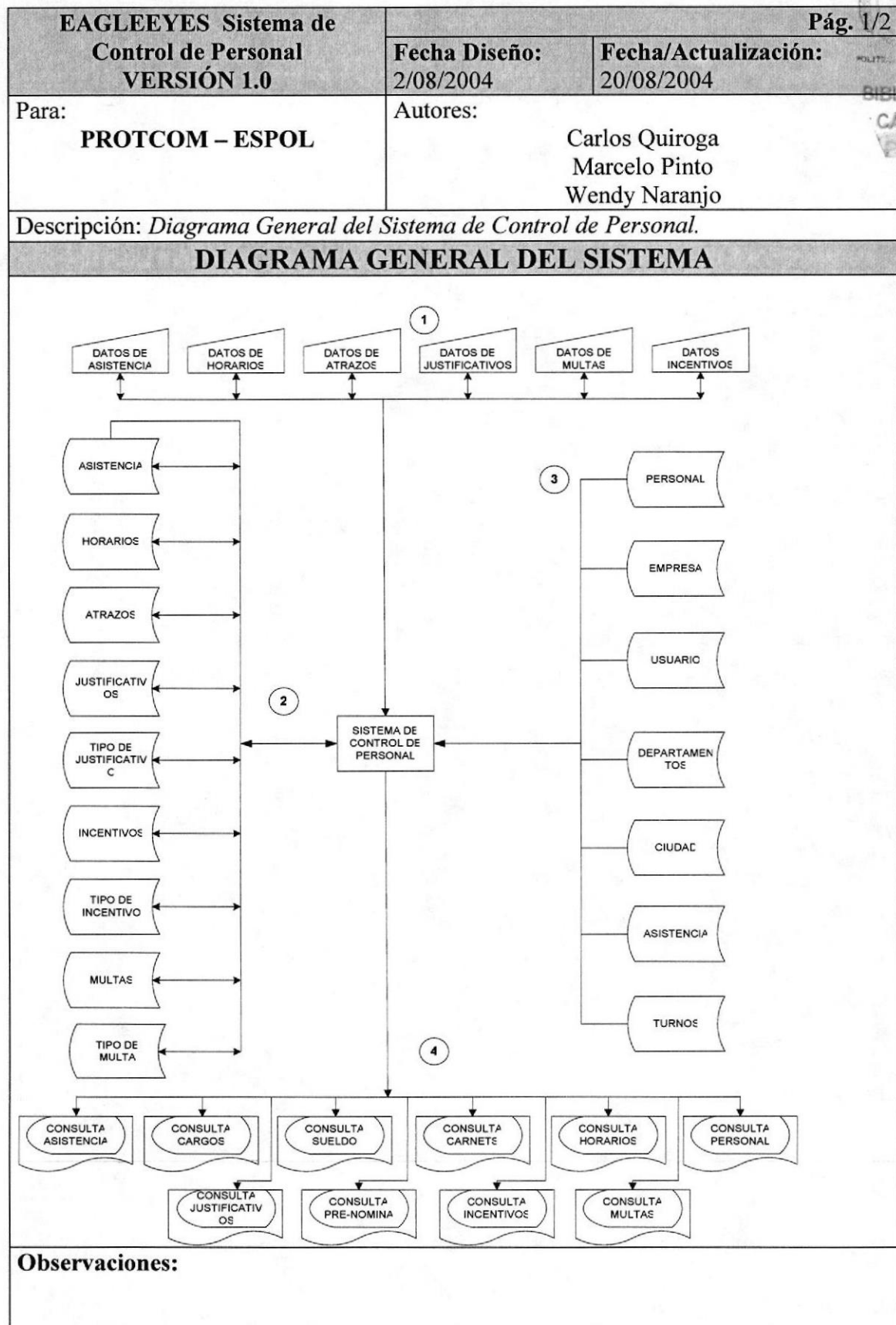




CAPÍTULO 4

DIAGRAMA GENERAL DEL SISTEMA

4. DIAGRAMA GENERAL DEL SISTEMA



EAGLEEYES Sistema de Control de Personal		Pág. 2/2	
de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Narrativa de Diagrama General del Sistema de Control de Personal.</i>			
NARRATIVA DEL DIAGRAMA GENERAL DEL SISTEMA			
1. -	Ingreso de los datos para las tablas Maestras del Sistema de Control de Personal EAGLEEYES. Genera los datos de Asistencia, Horarios, Atrasos, Justificativos, Multas e Incentivos.		
2. -	El Sistema Control de Personal EAGLEEYES realiza sus procesos en las tablas tales como Asistencia, Horarios, Atrasos, Justificativos, Tipo de Justificativo, Incentivos, Tipo de Incentivo, Multas, Tipo de Multas.		
3.-	El Sistema Control de Personal EAGLEEYES necesita consultar tablas como: Persona, Empresa, Departamentos, Usuarios Carnets, Asistencia, Turnos.		
4.-	El Sistema realiza Consultas de Asistencias, Cargos, Sueldos, Carnets, Horarios, Personal. A su vez genera informes de los Reportes de Justificativos, Pre-Nómina, Incentivos, Multas.		
Observaciones:			



CAPÍTULO 5

DIAGRAMA DE CASOS DE USO

5. DIAGRAMAS DE CASOS DE USO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
DISEÑO DE DIAGRAMA DE CASOS DE USO			
<pre>graph TD; A([1.-Proceso de Ingreso de persona]) --> B([Proceso de Ingreso de Justificativos]); A --> C([Procesos de Ingreso de multas]); A --> D([Proceso de Ingreso de Incentivos]); E([Proceso de Ingreso y Egreso de Personal]) --> B; E --> C; E --> D; E --> F([Proceso de Reporte de Faltas]); E --> G([Proceso de Reporte de pre-nominación]);</pre>			
Observaciones:			



CAPÍTULO 6

DIAGRAMAS DE TABLAS Y CLASES

6. DIAGRAMA de TABLAS Y CLASES

6.1. DIAGRAMA GENERAL

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Diagrama General de Relaciones de Tablas</i>			
DIAGRAMA GENERAL DE TABLAS			
The diagram illustrates the relationships between various tables in the system. The central table is CPm_Persona. It is connected to several other tables: CPm_Departamento, CPm_Ciudad, CPm_Cargo, CPm_Profesion, CPh_Horarioperson, CPh_Asistencia, CPh_Came, CPm_Empresa, CPm_Turno, CPm_Ciudad (repeated), CPT_Incentiv, CPm_TipoIncentiv, CPT_Justificativo, CPm_Tipojustificativo, CPT_Multas, and CPm_Tipomulta. The connections are represented by lines with dots at the ends, indicating foreign key relationships.			
Observaciones:			

6.2. DIAGRAMAS DE CLASE CLASIFICADOS POR CAPAS

6.2.1. CAPA DATOS

6.2.1.1 CLASE DATCARGO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/2	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datCargo.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Cargo a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datCargo{ private int Codigo; private String Descripcion; private double Sueldo; private boolean Estado; datCargo(){ this.Codigo=0; this.Descripcion="";this.Sueldo=0;this.Estado=false; } datCargo(String Descripcion,double Sueldo){ this.Codigo=0; this.Descripcion=Descripcion;this.Sueldo=Sueldo; this.Estado=true; } datCargo(int Codigo,String Descripcion,double Sueldo,boolean Estado){ this.Codigo=Codigo; this.Descripcion=Descripcion;this.Sueldo=Sueldo; this.Estado=Estado;} // Metodos boolean Grabar(){ cnxCargo cnx = new cnxCargo(); if (cnx.Grabar(Descripcion,Sueldo)) return true; else return false; } boolean Actualizar(){ cnxCargo cnx = new cnxCargo(); if (cnx.Modificar(Codigo,Descripcion,Sueldo,Estado)) return true; else return false;} boolean Anular(){ cnxCargo cnx = new cnxCargo(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxCargo cnx = new cnxCargo(); if (cnx.Eliminar(this.Codigo)) return true; else return false; }</pre>			
Observaciones:			



6.2.1.2 CLASE DATCARGO

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datCargo.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Cargo a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>String getUltimoCodigo(){ cnxCargo cnx = new cnxCargo(); int i = cnx.getUltimoCodigo(); return String.valueOf(i);} boolean Consultar(int lnCodigo){ datCargo a = new datCargo(); cnxCargo b = new cnxCargo(); a = b.Consultar(lnCodigo); this.Codigo = a.getCodigo(); this.Descripcion = a.getDescripcion(); this.Estado = a.getEstado(); if (this.getCodigo()<0){ return false; } return true; } // SETs void setCodigo(int Codigo){ this.Codigo=Codigo; } void setDescripcion(String Descripcion){ this.Descripcion=Descripcion; } void setSueldo(double Sueldo){ this.Sueldo=Sueldo; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs int getCodigo(){ return this.Codigo; } String getDescripcion(){ return this.Descripcion; } double getSueldo(){ return this.Sueldo; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			



6.2.1.3 CLASE DATASISTENCIA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datAsistencia.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Asistencia a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datAsistencia{ private int Numero; private int Persona; private String Carnet; datAsistencia(){ this.Numero=0; this.Persona=0; this.Carnet=""; } datAsistencia(String Carnet){ this.Carnet=Carnet; this.Persona=getPersona(Carnet); } datAsistencia(int Carnet){ this.Carnet="" + Carnet; this.Persona=Carnet; } // Metodos boolean Grabar(){ cnxAsistencia cnx = new cnxAsistencia(); if (cnx.Grabar(Persona,Carnet)) return true; else return false; } // SETs void setNumero(int Numero){ this.Numero=Numero; } void setPersona(int Persona){ this.Persona=Persona; } void setCarnet(String Carnet){ this.Carnet=Carnet; } // GETs int getNumero(){ return this.Numero; } private int getPersona(String Carnet){ clsConexion con = new clsConexion(); Statement cSQL; ResultSet rs; int per; per = 0; try{ cSQL = con.cnx.createStatement(); rs = cSQL.executeQuery("Select personal From CPt_Carnet Where codigo = "" + Carnet + "" and estado=true"); if (rs.next()){ per = rs.getInt(1); } con.cnx.close(); }catch(SQLException e){} return per; } String getCarnet(){ return this.Carnet; } }</pre>			
Observaciones:			

6.2.1.4 CLASE DATCIUDAD

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datCiudad.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Ciudad a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>class datCiudad{ private int Codigo = 0; private String Nombre = ""; private boolean Estado = false; datCiudad(){ datCiudad(String Nombre){ this.Nombre=Nombre; this.Estado=true;} datCiudad(int Codigo,String Nombre,boolean Estado){ this.Codigo=Codigo; this.Nombre=Nombre; this.Estado=Estado;} // Metodos boolean Grabar(){ cnxCiudad cnx = new cnxCiudad(); if (cnx.Grabar(Nombre)) return true; else return false; } boolean Actualizar(){ cnxCiudad cnx = new cnxCiudad(); if (cnx.Modificar(Codigo,Nombre,Estado)) return true; else return false;} boolean Anular(){ cnxCiudad cnx = new cnxCiudad(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxCiudad cnx = new cnxCiudad(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxCiudad cnx = new cnxCiudad(); int i = cnx.getUltimoCodigo(); return String.valueOf(i);} boolean Consultar(int lnCodigo){ datCiudad a = new datCiudad(); cnxCiudad b = new cnxCiudad(); a = b.Consultar(lnCodigo); this.Codigo = a.getCodigo(); this.Nombre = a.getNombre(); this.Estado = a.getEstado(); if (this.getCodigo()<0){ return false; }return true; } }</pre>			
Observaciones:			

6.2.1.5 CLASE DATCIUDAD

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datCiudad.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Ciudad a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>// SETs void setCodigo(int Codigo){ this.Codigo=Codigo;} void setNombre(String Nombre){ this.Nombre=Nombre;} void setEstado(boolean Estado){ this.Estado=Estado; } // GETs int getCodigo(){ return this.Codigo; } String getNombre(){ return this.Nombre; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			

6.2.1.6 CLASE DATDEPARTAMENTO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datDepartamento.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Departamento a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datDepartamento{ private int Codigo; private String Descripcion; private boolean Estado; datDepartamento(){ this.Codigo=0; this.Descripcion=""; this.Estado=false; } datDepartamento(String Descripcion){ this.Codigo=0; this.Descripcion=Descripcion; this.Estado=true; } datDepartamento(int Codigo,String Descripcion,boolean Estado){ this.Codigo=Codigo; this.Descripcion=Descripcion; this.Estado=Estado;} // Metodos boolean Grabar(){ cnxDepartamento cnx = new cnxDepartamento(); if (cnx.Grabar(Descripcion)) return true; else return false; } boolean Actualizar(){ cnxDepartamento cnx = new cnxDepartamento(); if (cnx.Modificar(Codigo,Descripcion,Estado)) return true; else return false;} boolean Anular(){ cnxDepartamento cnx = new cnxDepartamento(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxDepartamento cnx = new cnxDepartamento(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } }</pre>			
Observaciones:			

6.2.1.7 CLASE DATDEPARTAMENTO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datDepartamento.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Departamento a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>String getUltimoCodigo(){ cnxDepartamento cnx = new cnxDepartamento(); int i = cnx.getUltimoCodigo(); return String.valueOf(i);} boolean Consultar(int lnCodigo){ datDepartamento a = new datDepartamento(); cnxDepartamento b = new cnxDepartamento(); a = b.Consultar(lnCodigo); this.Codigo = a.getCodigo(); this.Descripcion = a.getDescripcion(); this.Estado = a.getEstado(); if (this.getCodigo()<0){ return false;} return true;} // SETs void setCodigo(int Codigo){ this.Codigo=Codigo;} void setDescripcion(String Descripcion){ this.Descripcion=Descripcion;} void setEstado(boolean Estado){ this.Estado=Estado;} // GETs int getCodigo(){ return this.Codigo; } String getDescripcion(){ return this.Descripcion;} boolean getEstado(){ return this.Estado;} }</pre>			
Observaciones:			



6.2.1.8 CLASE DATEMPRESA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datEmpresa.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Empresa a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>class datEmpresa{ private StringCodigo; private StringRuc; private StringNombre; private StringDireccion; private StringTelefono; private StringFax; private StringEmail; private StringWeb; private booleanEstado; datEmpresa(){ this.Codigo=""; this.Ruc=""; this.Nombre=""; this.Direccion=""; this.Telefono=""; this.Fax=""; this.Email=""; this.Web=""; this.Estado=false; } datEmpresa(StringNombre,StringRuc,StringDireccion,StringTelefono,StringFax,StringEmail,StringWeb){ this.Codigo=""; this.Ruc=Ruc; this.Nombre=Nombre; this.Direccion=Direccion; this.Telefono=Telefono; this.Fax=Fax; this.Email=Email; this.Web=Web; this.Estado=true; } datEmpresa(StringCodigo,StringNombre,StringRuc,StringDireccion,StringTelefono,StringFax,StringEmail,String Web,booleanEstado){ this.Codigo=Codigo; this.Ruc=Ruc; this.Nombre=Nombre; this.Direccion=Direccion; this.Telefono=Telefono; this.Fax=Fax; this.Email=Email; this.Web=Web; this.Estado=Estado; } // Metodos booleanGrabar(){ cnxEmpresa cnx = new cnxEmpresa(); if (cnx.Grabar(Nombre,Ruc,Direccion,Telefono,Fax,Email,Web)) return true; else return false; } booleanActualizar(){ cnxEmpresa cnx = new cnxEmpresa(); if (cnx.Actualizar(Codigo,Nombre,Ruc,Direccion,Telefono,Fax,Email,Web,Estado)) return true; else return false;} }</pre>			
Observaciones:			

6.2.1.9 CLASE DATEMPRESA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datEmpresa.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Empresa a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>boolean Anular(){ cnxEmpresa cnx = new cnxEmpresa(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxEmpresa cnx = new cnxEmpresa(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxEmpresa cnx = new cnxEmpresa(); return cnx.getUltimoCodigo();} boolean Consultar(String Codigo){ datEmpresa a = new datEmpresa(); cnxEmpresa b = new cnxEmpresa(); a = b.Consultar(Codigo); this.Codigo = a.getCodigo(); this.Ruc = a.getRuc(); this.Nombre = a.getNombre(); this.Direccion = a.getDireccion(); this.Telefono = a.getTelefono(); this.Fax = a.getFax(); this.Email = a.getEmail(); this.Web = a.getWeb(); if (this.getCodigo().equals("-1")){ return false;} return true;} // SETs void setCodigo(String Codigo){ this.Codigo=Codigo;} void setRuc(String Ruc){ this.Ruc=Ruc;} void setNombre(String Nombre){ this.Nombre=Nombre;} void setDireccion(String Direccion){ this.Direccion=Direccion;} void setTelefono(String Telefono){ this.Telefono=Telefono;} void setFax(String Fax){ this.Fax=Fax;} void setEmail(String Email){ this.Email=Email;} void setWeb(String Web){ this.Web=Web;} void setEstado(boolean Estado){ this.Estado=Estado;} // GETs String getCodigo(){ return this.Codigo;} String getRuc(){ return this.Ruc;} String getNombre(){ return this.Nombre;} String getDireccion(){return this.Direccion;} }</pre>			
Observaciones:			



6.2.1.10 CLASE DATINCENTIVO

EAGLEEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datIncentivo.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Incentivos a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>class datIncentivo{ private String Numero = ""; private String Fecha = " / / "; private int Personal = 0; private int Tipo = 0; private String Motivo = ""; private int Monto = 0; private boolean Estado = false; datIncentivo(){ } datIncentivo(String fecha,int personal,int tipo,String motivo,int monto){ this.Fecha=fecha; this.Personal=personal; this.Tipo=tipo; this.Motivo=motivo; this.Monto=monto; this.Estado=true; } datIncentivo(String Numero,String fecha,int personal,int tipo,String motivo,int monto,boolean estado){ this.Numero=Numero; this.Fecha=fecha; this.Personal=personal; this.Tipo=tipo; this.Motivo=motivo; this.Monto=monto; this.Estado=estado; } // Metodos boolean Grabar(){ cnxIncentivo cnx = new cnxIncentivo(); if (cnx.Grabar(Fecha,Personal,Tipo,Motivo,Monto)) return true; else return false; } boolean Actualizar(){ cnxIncentivo cnx = new cnxIncentivo(); if (cnx.Actualizar(Numero,Fecha,Personal,Tipo,Motivo,Monto,Estado)) return true; else return false;} boolean Anular(){ cnxIncentivo cnx = new cnxIncentivo(); if (cnx.Anular(Numero)) return true; else return false; } boolean Eliminar(){ cnxIncentivo cnx = new cnxIncentivo(); if (cnx.Eliminar(Numero)) return true; else return false; } String getUltimoNumero(){ cnxIncentivo cnx = new cnxIncentivo(); return cnx.getUltimoNumero(); } }</pre>			
Observaciones:			

6.2.1.11 CLASE DATINCENTIVO

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datIncentivo.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Incentivos a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>boolean Consultar(String Codigo){ datIncentivo a = new datIncentivo(); cnxIncentivo b = new cnxIncentivo(); a = b.Consultar(Codigo); this.Numero = a.getNumero(); this.Personal = a.getPersonal(); this.Tipo = a.getTipo(); this.Motivo = a.getMotivo(); this.Fecha = a.getFecha(); this.Monto = a.getMonto(); this.Estado = a.getEstado(); if (this.getNumero().equals("-1")){ return false; } return true; } // SETs void setNumero(String Numero){ this.Numero=Numero;} void setFecha(String Fecha){ this.Fecha=Fecha;} void setPersonal(int Personal){ this.Personal=Personal;} void setTipo(int Tipo){ this.Tipo=Tipo;} void setMotivo(String Motivo){ this.Motivo=Motivo;} void setMonto(int Monto){ this.Monto=Monto;} void setEstado(boolean Estado){ this.Estado=Estado;} // GETs String getNumero(){ return this.Numero;} String getFecha(){ return this.Fecha;} int getPersonal(){ return this.Personal;} int getTipo(){ return this.Tipo;} String getMotivo(){ return this.Motivo;} int getMonto(){ return this.Monto;} boolean getEstado(){ return this.Estado;} }</pre>			
Observaciones:			



6.2.1.12 CLASE DATJUSTIFICA

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datJustifica.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Justificativos a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datJustifica{ private String Numero = ""; private String Fecha = " / / "; private int Personal = 0; private int Tipo = 0; private String Motivo = ""; private boolean Estado = false; datJustifica(){} datJustifica(String fecha,int personal,int tipo,String motivo){ this.Fecha=fecha; this.Personal=personal; this.Tipo=tipo; this.Motivo=motivo; this.Estado=true;} datJustifica(String Numero,String fecha,int personal,int tipo,String motivo,boolean estado){ this.Numero=Numero; this.Fecha=fecha; this.Personal=personal; this.Tipo=tipo; this.Motivo=motivo; this.Estado=estado;} // Metodos boolean Grabar(){ cnxJustifica cnx = new cnxJustifica(); if (cnx.Grabar(Fecha,Personal,Tipo,Motivo)) return true; else return false; } boolean Actualizar(){ cnxJustifica cnx = new cnxJustifica(); if (cnx.Actualizar(Numero,Fecha,Personal,Tipo,Motivo,Estado)) return true; else return false;} boolean Anular(){ cnxJustifica cnx = new cnxJustifica(); if (cnx.Anular(Numero)) return true; else return false;} boolean Eliminar(){ cnxJustifica cnx = new cnxJustifica(); if (cnx.Eliminar(Numero)) return true; else return false; } }</pre>			
Observaciones:			

6.2.1.13 CLASE DATJUSTIFICA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datJustifica.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Justificativos a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>String getUltimoNumero(){ cnxJustifica cnx = new cnxJustifica(); return cnx.getUltimoNumero(); } boolean Consultar(String Codigo){ datJustifica a = new datJustifica(); cnxJustifica b = new cnxJustifica(); a = b.Consultar(Codigo); this.Numero = a.getNumero(); this.Personal = a.getPersonal(); this.Tipo = a.getTipo(); this.Motivo = a.getMotivo(); this.Fecha = a.getFecha(); this.Estado = a.getEstado(); if (this.getNumero().equals("-1")){ return false; } return true; } // SETs void setNumero(String Numero){ this.Numero=Numero; } void setFecha(String Fecha){ this.Fecha=Fecha; } void setPersonal(int Personal){ this.Personal=Personal; } void setTipo(int Tipo){ this.Tipo=Tipo; } void setMotivo(String Motivo){ this.Motivo=Motivo; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getNumero(){ return this.Numero; } String getFecha(){ return this.Fecha; } int getPersonal(){ return this.Personal; } int getTipo(){ return this.Tipo; } String getMotivo(){ return this.Motivo; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			



6.2.1.14 CLASE DATMULTA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datMulta.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Multas a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datMulta{ private String Numero = ""; private String Fecha = " / / "; private int Personal = 0; private int Tipo = 0; private String Motivo = ""; private int Monto = 0; private boolean Estado = false; datMulta(){ datMulta(String fecha,int personal,int tipo,String motivo,int monto){ this.Fecha=fecha; this.Personal=personal; this.Tipo=tipo; this.Motivo=motivo; this.Monto=monto; this.Estado=true;} datMulta(String Numero,String fecha,int personal,int tipo,String motivo,int monto,boolean estado){ this.Numero=Numero; this.Fecha=fecha; this.Personal=personal; this.Tipo=tipo; this.Motivo=motivo; this.Monto=monto; this.Estado=estado;} // Metodos boolean Grabar(){ cnxMulta cnx = new cnxMulta(); if (cnx.Grabar(Fecha,Personal,Tipo,Motivo,Monto)) return true; else return false; } boolean Actualizar(){ cnxMulta cnx = new cnxMulta(); if (cnx.Actualizar(Numero,Fecha,Personal,Tipo,Motivo,Monto,Estado)) return true; else return false;} boolean Anular(){ cnxMulta cnx = new cnxMulta(); if (cnx.Anular(Numero)) return true; else return false; } boolean Eliminar(){ cnxMulta cnx = new cnxMulta(); if (cnx.Eliminar(Numero)) return true; else return false; } String getUltimoNumero(){ cnxMulta cnx = new cnxMulta(); return cnx.getUltimoNumero();}</pre>			
Observaciones:			

6.2.1.15 CLASE DATMULTA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datMulta.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Multas a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre> boolean Consultar(String Codigo){ datMulta a = new datMulta(); cnxMulta b = new cnxMulta(); a = b.Consultar(Codigo); this.Numero = a.getNumero(); this.Personal = a.getPersonal(); this.Tipo = a.getTipo(); this.Motivo = a.getMotivo(); this.Fecha = a.getFecha(); this.Monto = a.getMonto(); this.Estado = a.getEstado(); if (this.getNumero().equals("-1")){ return false; } return true; } // SETs void setNumero(String Numero){ this.Numero=Numero; } void setFecha(String Fecha){ this.Fecha=Fecha; } void setPersonal(int Personal){ this.Personal=Personal; } void setTipo(int Tipo){ this.Tipo=Tipo; } void setMotivo(String Motivo){ this.Motivo=Motivo; } void setMonto(int Monto){ this.Monto=Monto; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getNumero(){ return this.Numero; } String getFecha(){ return this.Fecha; } int getPersonal(){ return this.Personal; } int getTipo(){ return this.Tipo; } String getMotivo(){ return this.Motivo; } int getMonto(){ return this.Monto; } boolean getEstado(){ return this.Estado; } } </pre>			
Observaciones:			



6.2.1.16 CLASE DATPERSONAL

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datPersonal.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Personal a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>class datPersonal{ private String Codigo = ""; private String Nombre = ""; private String Apellido = ""; private String NombreCorto = ""; private String Clave = ""; private String Direccion = ""; private String Cedula = ""; private int Cargo = 0; private int Dpto = 0; private String Telefono = ""; private String Email = ""; private String FEntrada = " / / "; private String FSalida = " / / "; private String FNacimiento = " / / "; private int LNacimiento = 0; private int EstadoCivil = 0; private int Profesion = 0; private String Foto = ""; private boolean Estado = false; datPersonal(){ } datPersonal(String Nombre,String Apellido,String NombreCorto,String Clave, String Direccion,String Cedula,int Cargo,int Dpto,String Telefono, String Email,String FEntrada,String FNacimiento,int LNacimiento, int EstadoCivil,int Profesion,String Foto){ this.Nombre=Nombre; this.Apellido=Apellido; this.NombreCorto=NombreCorto; this.Clave=Clave; this.Direccion=Direccion; this.Cedula=Cedula; this.Cargo=Cargo; this.Dpto=Dpto; this.Telefono=Telefono; this.Email=Email; this.FEntrada=FEntrada; this.FNacimiento=FNacimiento; this.LNacimiento=LNacimiento; this.EstadoCivil=EstadoCivil; this.Profesion=Profesion; this.Foto=Foto; this.Estado=true; } datPersonal(String Codigo,String Nombre,String Apellido,String NombreCorto, String Clave,String Direccion,String Cedula,int Cargo,int Dpto, String Telefono,String Email,String FEntrada,String FNacimiento, int LNacimiento,int EstadoCivil,int Profesion,String Foto,boolean Estado){ this.Codigo=Codigo; this.Nombre=Nombre; this.Apellido=Apellido; this.NombreCorto=NombreCorto; this.Clave=Clave; this.Direccion=Direccion; this.Cedula=Cedula; this.Cargo=Cargo; this.Dpto=Dpto; this.Telefono=Telefono; this.Email=Email; this.FEntrada=FEntrada; this.FNacimiento=FNacimiento; this.LNacimiento=LNacimiento; this.EstadoCivil=EstadoCivil; this.Profesion=Profesion; this.Foto=Foto; this.Estado=Estado; } }</pre>			
Observaciones:			

6.2.1.17 CLASE DATPERSONAL

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datPersonal.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Personal a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>// Metodos boolean Grabar(){ cnxPersonal cnx = new cnxPersonal(); if (cnx.Grabar(Nombre,Apellido,NombreCorto,Clave, Direccion,Cedula,Cargo,Dpto,Telefono,Email,FEntrada, FNacimiento,LNacimiento,EstadoCivil,Profesion,Foto)) return true; else return false; } boolean Actualizar(){ cnxPersonal cnx = new cnxPersonal(); if (cnx.Actualizar(Codigo,Nombre,Apellido,NombreCorto,Clave, Direccion,Cedula,Cargo,Dpto,Telefono,Email,FEntrada, FNacimiento,LNacimiento,EstadoCivil,Profesion,Foto,Estado)) return true; else return false; } boolean Anular(){ cnxPersonal cnx = new cnxPersonal(); if (cnx.Anular(Codigo)) return true; else return false; } boolean Eliminar(){ cnxPersonal cnx = new cnxPersonal(); if (cnx.Eliminar(Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxPersonal cnx = new cnxPersonal(); return cnx.getUltimoCodigo(); } boolean Consultar(String Codigo){ datPersonal a = new datPersonal(); cnxPersonal b = new cnxPersonal(); a = b.Consultar(Codigo); this.Codigo = a.getCodigo(); this.Nombre = a.getNombre(); this.Apellido = a.getApellido(); this.NombreCorto= a.getNombreCorto(); this.Clave = a.getClave(); this.Direccion = a.getDireccion(); this.Cedula = a.getCedula(); this.Cargo = a.getCargo(); this.Dpto = a.getDpto(); this.Telefono = a.getTelefono(); this.Email = a.getEmail(); this.FEntrada = a.getFEntrada(); this.FSalida = a.getFSalida(); this.FNacimiento= a.getFNacimiento(); this.LNacimiento= a.getLNacimiento(); this.EstadoCivil= a.getEstadoCivil(); this.Profesion = a.getProfesion(); this.Foto = a.getFoto(); this.Estado = a.getEstado(); if (this.getCodigo().equals("-1")){ return false; } return true; }</pre>			
Observaciones:			

6.2.1.18 CLASE DATPERSONAL

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datPersonal.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Personal a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>// SETs void setCodigo(String Codigo){ this.Codigo=Codigo;} void setNombre(String Nombre){ this.Nombre=Nombre;} void setApellido(String Apellido){ this.Apellido=Apellido;} void setNombreCorto(String NombreCorto){ this.NombreCorto=NombreCorto;} void setClave(String Clave){ this.Clave=Clave;} void setDireccion(String Direccion){this.Direccion=Direccion;} void setCedula(String Cedula){ this.Cedula=Cedula;} void setCargo(int Cargo){ this.Cargo=Cargo;} void setDpto(int Dpto){ this.Dpto=Dpto;} void setTelefono(String Telefono){ this.Telefono=Telefono;} void setEmail(String Email){this.Email=Email;} void setFEntrada(String FEntrada){this.FEntrada=FEntrada;} void setFSalida(String FSalida){this.FSalida=FSalida;} void setFNacimiento(String FNacimiento){this.FNacimiento=FNacimiento;} void setLNacimiento(int LNacimiento){this.LNacimiento=LNacimiento;} void setEstadoCivil(int EstadoCivil){this.EstadoCivil=EstadoCivil;} void setProfesion(int Profesion){ this.Profesion=Profesion;} void setFoto(String Foto){this.Foto=Foto;} void setEstado(boolean Estado){this.Estado=Estado;} // GETs String getCodigo(){return this.Codigo;} String getNombre(){return this.Nombre;} String getApellido(){return this.Apellido;} String getNombreCorto(){return this.NombreCorto;} String getClave(){return this.Clave;} String getDireccion(){return this.Direccion;} String getCedula(){return this.Cedula;} int getCargo(){return this.Cargo;} int getDpto(){return this.Dpto;} String getTelefono(){return this.Telefono;} String getEmail(){return this.Email;} String getFEntrada(){return this.FEntrada;} String getFSalida(){return this.FSalida;} String getFNacimiento(){return this.FNacimiento;} int getLNacimiento(){return this.LNacimiento;} int getEstadoCivil(){return this.EstadoCivil;} int getProfesion(){return this.Profesion;} String getFoto(){return this.Foto;} boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			

6.2.1.19 CLASE DATPROFESION

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datProfesion.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla Profesión a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datProfesion{ private String Codigo; private String Descripcion; private String Siglas; private boolean Estado; datProfesion(){ this.Codigo=""; this.Descripcion=""; this.Siglas=""; this.Estado=false; } datProfesion(String Descripcion,String Siglas){ this.Codigo=""; this.Descripcion=Descripcion; this.Siglas=Siglas; this.Estado=true; } datProfesion(String Codigo,String Descripcion,String Siglas,boolean Estado){ this.Codigo=Codigo; this.Descripcion=Descripcion; this.Siglas=Siglas; this.Estado=Estado;} // Metodos boolean Grabar(){ cnxProfesion cnx = new cnxProfesion(); if (cnx.Grabar(Descripcion,Siglas)) return true; else return false; } boolean Actualizar(){ cnxProfesion cnx = new cnxProfesion(); if (cnx.Actualizar(Codigo,Descripcion,Siglas,Estado)) return true; else return false;} boolean Anular(){ cnxProfesion cnx = new cnxProfesion(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxProfesion cnx = new cnxProfesion(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxProfesion cnx = new cnxProfesion(); return cnx.getUltimoCodigo();} boolean Consultar(String Codigo){ datProfesion a = new datProfesion(); cnxProfesion b = new cnxProfesion(); a = b.Consultar(Codigo); this.Codigo = a.getCodigo(); this.Descripcion= a.getDescripcion(); this.Siglas = a.getSiglas(); this.Estado = a.getEstado(); if (this.getCodigo().equals("-1")){ return false;} return true;} }</pre>			
Observaciones:			



6.2.1.20 CLASE DATPROFESION

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datProfesion.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Profesión a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>// SETs void setCodigo(String Codigo){ this.Codigo=Codigo; } void setDescripcion(String Descripcion){ this.Descripcion=Descripcion; } void setSiglas(String Siglas){ this.Siglas=Siglas; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getCodigo(){ return this.Codigo; } String getDescripcion(){ return this.Descripcion; } String getSiglas(){ return this.Siglas; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			

6.2.1.21 CLASE DATTIPOINCENTIVO

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTipoIncentivo.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla TipoIncentivo a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datTipoIncentivo{ private String Codigo; private String Descripcion; private double Monto; private boolean Estado; datTipoIncentivo(){ this.Codigo=""; this.Descripcion=""; this.Monto=0; this.Estado=false; } datTipoIncentivo(String Descripcion,double Monto){ this.Codigo=""; this.Descripcion=Descripcion; this.Monto=Monto; this.Estado=true; } datTipoIncentivo(String Codigo,String Descripcion,double Monto,boolean Estado){ this.Codigo=Codigo; this.Descripcion=Descripcion; this.Monto=Monto; this.Estado=Estado; } // Metodos boolean Grabar(){ cnxTipoIncentivo cnx = new cnxTipoIncentivo(); if (cnx.Grabar(Descripcion,Monto)) return true; else return false; } boolean Actualizar(){ cnxTipoIncentivo cnx = new cnxTipoIncentivo(); if (cnx.Actualizar(Codigo,Descripcion,Monto,Estado)) return true; else return false; } boolean Anular(){ cnxTipoIncentivo cnx = new cnxTipoIncentivo(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxTipoIncentivo cnx = new cnxTipoIncentivo(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxTipoIncentivo cnx = new cnxTipoIncentivo(); return cnx.getUltimoCodigo(); } boolean Consultar(String Codigo){ datTipoIncentivo a = new datTipoIncentivo(); cnxTipoIncentivo b = new cnxTipoIncentivo(); a = b.Consultar(Codigo); this.Codigo = a.getCodigo(); this.Descripcion= a.getDescripcion(); this.Monto = a.getMonto(); this.Estado = a.getEstado(); if (this.getCodigo().equals("-1")){ return false; } return true; } }</pre>			
Observaciones:			



6.2.1.22 CLASE DATTIPOINCENTIVO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTipoIncentivo.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla TipoIncentivo a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>// SETs void setCodigo(String Codigo){ this.Codigo=Codigo; } void setDescripcion(String Descripcion){ this.Descripcion=Descripcion; } void setMonto(double Monto){ this.Monto=Monto; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getCodigo(){ return this.Codigo; } String getDescripcion(){ return this.Descripcion; } double getMonto(){ return this.Monto; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			

6.2.1.23 CLASE DATTIPOJUSTIFICA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTipoJustifica.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla TipoJustificacion a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datTipoJustifica{ private String Codigo;private String Descripcion;private boolean Estado; datTipoJustifica(){ this.Codigo=""; this.Descripcion=""; this.Estado=false; } datTipoJustifica(String Descripcion){ this.Codigo=""; this.Descripcion=Descripcion; this.Estado=true; } datTipoJustifica(String Codigo,String Descripcion,boolean Estado){ this.Codigo=Codigo; this.Descripcion=Descripcion; this.Estado=Estado;} // Metodos boolean Grabar(){ cnxTipoJustifica cnx = new cnxTipoJustifica(); if (cnx.Grabar(Descripcion)) return true; else return false; } boolean Actualizar(){ cnxTipoJustifica cnx = new cnxTipoJustifica(); if (cnx.Actualizar(Codigo,Descripcion,Estado)) return true; else return false;} boolean Anular(){ cnxTipoJustifica cnx = new cnxTipoJustifica(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxTipoJustifica cnx = new cnxTipoJustifica(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxTipoJustifica cnx = new cnxTipoJustifica(); return cnx.getUltimoCodigo();} boolean Consultar(String Codigo){ datTipoJustifica a = new datTipoJustifica(); cnxTipoJustifica b = new cnxTipoJustifica(); a = b.Consultar(Codigo); this.Codigo = a.getCodigo(); this.Descripcion= a.getDescripcion(); this.Estado = a.getEstado(); if (this.getCodigo().equals("-1")){ return false; } return true;} }</pre>			
Observaciones:			

6.2.1.24 CLASE DATTIPOJUSTIFICA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTipoJustifica.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla TipoJustificacion a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>// SETs void setCodigo(String Codigo){ this.Codigo=Codigo; } void setDescripcion(String Descripcion){ this.Descripcion=Descripcion; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getCodigo(){ return this.Codigo; } String getDescripcion(){ return this.Descripcion; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			

6.2.1.25 CLASE DATTIPOMULTA

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTipoMultajava	Tipo: Public	
Descripción: Clase que contiene datos de la tabla TipoMultajava a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datTipoMultajava{ private String Codigo; private String Descripcion; private double Monto; private boolean Estado; datTipoMultajava(){ this.Codigo=""; this.Descripcion=""; this.Monto=0;this.Estado=false; } datTipoMultajava(String Descripcion,double Monto){ this.Codigo=""; this.Descripcion=Descripcion;this.Monto=Monto; this.Estado=true; } datTipoMultajava(String Codigo,String Descripcion,double Monto,boolean Estado){ this.Codigo=Codigo;this.Descripcion=Descripcion; this.Monto=Monto;this.Estado=Estado;} // Metodos boolean Grabar(){ cnxTipoMultajava cnx = new cnxTipoMultajava(); if (cnx.Grabar(Descripcion,Monto)) return true; else return false; } boolean Actualizar(){ cnxTipoMultajava cnx = new cnxTipoMultajava(); if (cnx.Actualizar(Codigo,Descripcion,Monto,Estado)) return true; else return false;} boolean Anular(){ cnxTipoMultajava cnx = new cnxTipoMultajava(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxTipoMultajava cnx = new cnxTipoMultajava(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxTipoMultajava cnx = new cnxTipoMultajava(); return cnx.getUltimoCodigo();} boolean Consultar(String Codigo){ datTipoMultajava a = new datTipoMultajava(); cnxTipoMultajava b = new cnxTipoMultajava(); a = b.Consultar(Codigo); this.Codigo = a.getCodigo(); this.Descripcion= a.getDescripcion(); this.Monto = a.getMonto(); this.Estado = a.getEstado(); if (this.getCodigo().equals("-1")){ return false;} return true;} }</pre>			
Observaciones:			



6.2.1.26 CLASE DATTIPOMULTA

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTipoMultajava	Tipo: Public	
Descripción: Clase que contiene datos de la tabla TipoMultajava a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre> // SETs void setCodigo(String Codigo){ this.Codigo=Codigo; } void setDescripcion(String Descripcion){ this.Descripcion=Descripcion; } void setMonto(double Monto){ this.Monto=Monto; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getCodigo(){ return this.Codigo; } String getDescripcion(){ return this.Descripcion; } double getMonto(){ return this.Monto; } boolean getEstado(){ return this.Estado; } } </pre>			
Observaciones:			



6.2.1.27 CLASE DATTURNOSHORARIOS

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTurnosHoraios.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla TurnosHorarios a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>class datTurnosHorarios{ private String Codigo; private String Descripcion;private boolean Estado; datTurnosHorarios(){ this.Codigo=""; this.Descripcion=""; this.Estado=false; } datTurnosHorarios(String Descripcion){ this.Codigo=""; this.Descripcion=Descripcion; this.Estado=true; } datTurnosHorarios(String Codigo,String Descripcion,boolean Estado){ this.Codigo=Codigo; this.Descripcion=Descripcion; this.Estado=Estado;} // Metodos boolean Grabar(){ cnxTurnosHorarios cnx = new cnxTurnosHorarios(); if (cnx.Grabar(Descripcion)) return true; else return false; } boolean Actualizar(){ cnxTurnosHorarios cnx = new cnxTurnosHorarios(); if (cnx.Actualizar(Codigo,Descripcion,Estado)) return true; else return false;} boolean Anular(){ cnxTurnosHorarios cnx = new cnxTurnosHorarios(); if (cnx.Anular(this.Codigo)) return true; else return false; } boolean Eliminar(){ cnxTurnosHorarios cnx = new cnxTurnosHorarios(); if (cnx.Eliminar(this.Codigo)) return true; else return false; } String getUltimoCodigo(){ cnxTurnosHorarios cnx = new cnxTurnosHorarios(); return cnx.getUltimoCodigo();} boolean Consultar(String Codigo){ return true; } }</pre>			
Observaciones:			

6.2.1.28 CLASE DATTURNOSHORARIOS

EAGLEEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datTurnosHoraios.java	Tipo: Public	
Descripción: Clase que contiene datos de la tabla TurnosHorarios a ser utilizado por su formulario respectivo			
DESCRIPCIÓN DE CLASE			
<pre>// SETs void setCodigo(String Codigo){ this.Codigo=Codigo; } void setDescripcion(String Descripcion){ this.Descripcion=Descripcion; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getCodigo(){ return this.Codigo; } String getDescripcion(){ return this.Descripcion; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			



6.2.1.29 CLASE DATUSUARIO

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datUsuario.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Usuarios a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>class datUsuario{ private String Id;private String Usuario; private String Clave;private int Nivel;private boolean Estado; datUsuario(){ this.Id="";this.Usuario=""; this.Clave="";this.Nivel=0;this.Estado=false; } datUsuario(String Usuario,String Clave,int Nivel){ this.Id=""; this.Usuario=Usuario;this.Clave=Clave; this.Nivel=Nivel;this.Estado=true; } datUsuario(String Id,String Usuario,String Clave,int Nivel,boolean Estado){ this.Id=Id; this.Usuario=Usuario;this.Clave=Clave; this.Nivel=Nivel;this.Estado=Estado;} // Metodos boolean Grabar(){ cnxUsuario cnx = new cnxUsuario(); if (cnx.Grabar(Usuario,Clave,Nivel)) return true; else return false; } boolean Actualizar(){ cnxUsuario cnx = new cnxUsuario(); if (cnx.Actualizar(Id,Usuario,Clave,Nivel,Estado)) return true; else return false;} boolean Anular(){ cnxUsuario cnx = new cnxUsuario(); if (cnx.Anular(this.Id)) return true; else return false; } boolean Eliminar(){ cnxUsuario cnx = new cnxUsuario(); if (cnx.Eliminar(this.Id)) return true; else return false; } String getUltimoCodigo(){ cnxUsuario cnx = new cnxUsuario(); return cnx.getUltimoCodigo();} boolean Consultar(String Id){ datUsuario a = new datUsuario(); cnxUsuario b = new cnxUsuario(); a = b.Consultar(Id); this.Id = a.getId(); this.Nivel = a.getNivel(); this.Usuario = a.getUsuario(); this.Clave = a.getClave(); this.Estado = a.getEstado(); if (this.getId().equals("-1")){ return false;} return true;} }</pre>			
Observaciones:			

6.2.1.30 CLASE DATUSUARIO

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Datos	Nombre de la Clase: datUsuario.java	Tipo: Public	
Descripción: <i>Clase que contiene datos de la tabla Usuarios a ser utilizado por su formulario respectivo</i>			
DESCRIPCIÓN DE CLASE			
<pre>// SETs void setId(String Id){ this.Id=Id; } void setUsuario(String Usuario){ this.Usuario=Usuario; } void setClave(String Clave){ this.Clave=Clave; } void setNivel(int Nivel){ this.Nivel=Nivel; } void setEstado(boolean Estado){ this.Estado=Estado; } // GETs String getId(){ return this.Id; } String getUsuario(){ return this.Usuario; } String getClave(){ return this.Clave; } int getNivel(){ return this.Nivel; } boolean getEstado(){ return this.Estado; } }</pre>			
Observaciones:			





6.2.2. CAPA CLIENTES

6.2.2.1 CLASE CLSMANTENIMIENTOTIPO1

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: clsMantenimientoTipo1.java	Tipo: Abstract	
Descripción: <i>Clase abstracta base para la herencia de los formularios de mantenimiento.</i>			
DESCRIPCIÓN DE CLASE			
<pre>abstract class clsMantenimientoTipo1 extends clsMantenimientoBasico{ String SQLconsulta = ""; // Etiquetas clsLabel lblCodigo = new clsLabel("Código:",10,10,80,20); clsLabel lblDescripcion = new clsLabel("Descripción:",10,32,80,20); clsEstado cntEstado = new clsEstado(10,54); // Campos clsTextField txtCodigo = new clsTextField(90,10,50,20); clsExaminar btnExaminar = new clsExaminar(145,10); clsTextField txtDescripcion = new clsTextField(90,32,250,20); clsMantenimientoTipo1(){ super("Mantenimiento Basico Tipo 1",350,170); Componentes();} clsMantenimientoTipo1(String titulo){ super(titulo,350,170); Componentes();} clsMantenimientoTipo1(String titulo,int w,int h){ super(titulo,w,h); Componentes();} private void Componentes(){ // Etiquetas getContentPane().add(lblCodigo); getContentPane().add(lblDescripcion); getContentPane().add(cntEstado); // Campos txtCodigo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) if (txtCodigo.getText().equals("")) Examinar(SQLconsulta); Else Consultar(txtCodigo.getText()); transferFocus(); } }); getContentPane().add(txtCodigo); btnExaminar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ if (txtCodigo.getText().equals("")) Examinar(SQLconsulta); Else Consultar(txtCodigo.getText()); transferFocus(); } }); getContentPane().add(btnExaminar); txtDescripcion.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus(); } }); getContentPane().add(txtDescripcion);} }</pre>			
Observaciones: Hereda de clsMantenimientoBasico.java y añade etiquetas y campos necesarios para las tablas de mantenimiento.			

6.2.2.2 CLASE CLSMANTENIMIENTOTIPO1

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: clsMantenimientoTipo1.java	Tipo: Abstract	
Descripción: <i>Clase abstracta base para la herencia de los formularios de mantenimiento.</i>			
DESCRIPCIÓN DE CLASE			
<pre>// Metodos void Nuevo(){ LimpiarEntrada(); HabilitarEntrada();} void Modificar(){ HabilitarEntrada();} void Grabar(){ HabilitarEntrada();} void Cancelar(){ LimpiarEntrada(); HabilitarEntrada();} void LimpiarEntrada(){ txtCodigo.setText(""); txtDescripcion.setText(""); cntEstado.setEstado(true); setRegistro(false); HabilitarBotones(); } void HabilitarEntrada(){ txtCodigo.Habilitar(!getEditando()); btnExaminar.setEnabled(!getEditando()); txtDescripcion.Habilitar(getEditando()); cntEstado.Habilitar(getEditando() & !getEsnuevo());} void Consultar(String Codigo){ JOptionPane.showMessageDialog(null,"Sin programación","Aviso",JOptionPane.INFORMATION_MESSAGE); }; void Examinar(String cSQL){ frmExaminar frm = new frmExaminar(cSQL); if (!frm.getCodigo().equals("")){ txtCodigo.setText(frm.getCodigo()); Consultar(txtCodigo.getText()); } } // SETs void setEstado(boolean estado){ cntEstado.setEstado(estado);} // GETs boolean getEstado(){ return cntEstado.getEstado(); } }</pre>			
Observaciones: Hereda de clsMantenimientoBasico.java y añade etiquetas y capos necesarios para las tablas de mantenimiento.			

6.2.2.3 CLASE CLSMANTENIMIENTOTIPO2

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: clsMantenimientoTipo2.java	Tipo: Abstract	
Descripción: <i>Clase abstracta base para la herencia de los formularios de mantenimiento.</i>			
DESCRIPCIÓN DE CLASE			
<pre>abstract class clsMantenimientoTipo2 extends clsMantenimientoTipo1 { // Etiquetas clsLabel lblMonto = new clsLabel("Valor:",10,54,80,20); // Campos clsTextField txtMonto = new clsTextField(90,54,80,20); clsMantenimientoTipo2(){ super("Mantenimiento Basico Tipo 2",350,170); Componentes(); } clsMantenimientoTipo2(String titulo,int w,int h){ super(titulo,w,h); Componentes();} private void Componentes(){ getContentPane().add(lblMonto); txtMonto.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus(); } }); getContentPane().add(txtMonto); cntEstado.setLocation(cntEstado.getX(),cntEstado.getY()+22); } // Metodos void LimpiarEntrada(){ super.LimpiarEntrada(); txtMonto.setText("0.00"); } void HabilitarEntrada(){ super.HabilitarEntrada(); txtMonto.Habilitar(getEditando()); } void Consultar(String Codigo){ JOptionPane.showMessageDialog(null,"Sin programación", "Aviso", JOptionPane.INFORMATION_MESSAGE); }; }</pre>			
Observaciones: Hereda de clsMantenimientoTipo1.java y añade etiquetas y campos necesarios para las tablas de mantenimiento.			



6.2.2.4 CLASE CLSMANTENIMIENTOTIPO3

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: clsMantenimientoTipo3.java	Tipo: Abstract	
Descripción: Clase abstracta base para la herencia de los formularios de mantenimiento.			
DESCRIPCIÓN DE CLASE			
<pre> abstract class clsMantenimientoTipo3 extends clsMantenimientoBasico{ String SQLconsulta = ""; // Etiquetas clsLabel lblCodigo = new clsLabel("Código:",10,10,80,20); clsLabel lblCedRuc = new clsLabel("Cédula o Ruc:",220,10,80,20); clsLabel lblNombre = new clsLabel("Nombre:",10,32,80,20); clsLabel lblDireccion = new clsLabel("Dirección:",10,54,80,20); clsLabel lblTelefono = new clsLabel("Teléfono:",10,76,80,20); clsLabel lblEmail = new clsLabel("e-mail:",10,98,80,20); clsEstado cntEstado = new clsEstado(10,120); // Campos clsTextField txtCodigo = new clsTextField(90,10,50,20); clsExaminar btnExaminarCod= new clsExaminar(145,10); clsTextField txtCedRuc = new clsTextField(260,10,80,20); clsExaminar btnExaminarCed = new clsExaminar(345,10); clsTextField txtNombre = new clsTextField(90,32,250,20); clsTextField txtDireccion= new clsTextField(90,54,400,20); clsTextField txtTelefono = new clsTextField(90,76,100,20); clsTextField txtEmail = new clsTextField(90,98,400,20); clsMantenimientoTipo3(){super("Mantenimiento Basico Tipo 3",500,350); Componentes();} clsMantenimientoTipo3(String titulo){super(titulo,350,170);Componentes();} clsMantenimientoTipo3(String titulo,int w,int h){super(titulo,w,h);Componentes();} private void Componentes(){ // Etiquetas getContentPane().add(lblCodigo);getContentPane().add(lblCedRuc); getContentPane().add(lblNombre);getContentPane().add(lblDireccion); getContentPane().add(lblTelefono);getContentPane().add(lblEmail); getContentPane().add(cntEstado); // Campos txtCodigo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) if (txtCodigo.getText().equals("")) Examinar(SQLconsulta); Else Consultar(txtCodigo.getText()); transferFocus();}); getContentPane().add(txtCodigo); btnExaminarCod.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ if (txtCodigo.getText().equals("")) Examinar(SQLconsulta); Else Consultar(txtCodigo.getText()); transferFocus();}); getContentPane().add(btnExaminarCod); txtCedRuc.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) if (txtCedRuc.getText().equals("")) Examinar(SQLconsulta); Else Consultar(txtCedRuc.getText());transferFocus();}); getContentPane().add(txtCedRuc); btnExaminarCed.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ if (txtCedRuc.getText().equals("")) Examinar(SQLconsulta); Else Consultar(txtCedRuc.getText());transferFocus();}); </pre>			
Observaciones: Hereda de clsMantenimientoBasico.java y añade etiquetas y campos necesarios para las tablas de mantenimiento.			

6.2.2.5 CLASE CLSMANTENIMIENTOTIPO3

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: clsMantenimientoTipo3.java	Tipo: Abstract	
Descripción: <i>Clase abstracta base para la herencia de los formularios de mantenimiento.</i>			
DESCRIPCIÓN DE CLASE			
<pre>// Metodos void Nuevo(){ LimpiarEntrada(); HabilitarEntrada(); } void Modificar(){ HabilitarEntrada(); } void Grabar(){ HabilitarEntrada(); } void Cancelar(){ LimpiarEntrada(); HabilitarEntrada(); } void LimpiarEntrada(){ txtCodigo.setText(""); txtCedRuc.setText(""); txtNombre.setText(""); txtDireccion.setText(""); txtTelefono.setText(""); txtEmail.setText(""); cntEstado.setEstado(true); setRegistro(false); HabilitarBotones(); } void HabilitarEntrada(){ txtCodigo.Habilitar(!getEditando()); btnExaminarCod.setEnabled(!getEditando()); txtCedRuc.Habilitar(getEditando()); btnExaminarCed.setEnabled(getEditando()); txtNombre.Habilitar(getEditando()); txtDireccion.Habilitar(getEditando()); txtTelefono.Habilitar(getEditando()); txtEmail.Habilitar(getEditando()); cntEstado.Habilitar(getEditando() & !getEsNuevo()); } void Consultar(String Codigo){ JOptionPane.showMessageDialog(null,"Sin programación","Aviso",JOptionPane.INFORMATION_MESSAGE); }; void Examinar(String cSQL){ frmExaminar frm = new frmExaminar(cSQL); if (!frm.getCodigo().equals("")){ txtCodigo.setText(frm.getCodigo()); Consultar(txtCodigo.getText()); } } // SETs void setEstado(boolean estado){ cntEstado.setEstado(estado); } // GETs boolean getEstado(){ return cntEstado.getEstado(); } }}</pre>			
Observaciones: Hereda de clsMantenimientoBasico.java y añade etiquetas y campos necesarios para las tablas de mantenimiento.			

6.2.2.6 CLASE CLSPRELIMINAR

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: clsPreliminar.java	Tipo: Public	
Descripción: <i>Clase encargada de recoger los datos y mandar mediante parámetro a su respectivo reporte.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class clsPreliminar{ public clsPreliminar(String archivo, HashMap mapa){ run(archivo, mapa); } public void run(String archivo,HashMap mapa){ try{ //JDialog viewer = new JDialog(new JDialog(),"Vista preliminar", true); JFrame viewer = new JFrame("Vista preliminar"); //Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize(); //viewer.setSize(screenSize.width,screenSize.height); viewer.setSize(600,450); viewer.setLocationRelativeTo(null); // Cargar el archivo .jasper JasperReport template = JasperManager.loadReport("reportes/" + archivo + ".jasper"); // En mapa se especifican los parametros del reporte clsConexion con = new clsConexion(); JasperPrint report = JasperManager.fillReport(template, mapa,con.cnx); // Mostrar el reporte JRViewer jrv = new JRViewer(report); viewer.getContentPane().add(jrv); viewer.show(); } catch(Exception ex){ ex.printStackTrace(); } } }</pre>			
Observaciones:			

6.2.2.7 CLASE FRMASISTENCIA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmAsistencia.java	Tipo: Public	
Descripción: Clase encargada de recoger la información de asistencia del personal por código de ingreso.			
DESCRIPCIÓN DE CLASE			
<pre>class frmAsistencia extends JFrame{ // Etiquetas private JPanel pnlFechaHora = new JPanel(null); private JLabel lblFecha = new JLabel(); private JLabel lblHora = new JLabel(); private JPanel pnlRegistro= new JPanel(null); private JLabel lblCarnet = new JLabel("Ingrese su Clave:",5,25,180,30); private JPasswordField txtClave= new JPasswordField(); private JPanel pnlSaludo = new JPanel(null); private JLabel lblSaludo = new JLabel("Hola:");private JLabel lblMensaje = new JLabel(""); private JLabel lblRegistro = new JLabel(""); private JButton btnAceptar = new JButton("Aceptar","A","Registrar","aceptar",160,170); frmAsistencia(){ super("Registro de Asistencia",false,true,false,true); Entorno(); Componentes(); show();PonerHora(); } void Centrar(int w,int h){ setLocation((w-getWidth())/2,(h-getHeight())/2);} //METODOS private void Entorno(){ setBounds(10,10,400,300);getContentPane().setLayout(null); setFrameIcon(new ImageIcon("imagenes/tmp.gif")); } private void Componentes(){ pnlFechaHora.setBounds(5,5,getWidth()-20,70); pnlFechaHora.setBorder(new TitledBorder("")); //pnlFechaHora.setBorder(new javax.swing.border.BevelBorder(1)); pnlFechaHora.setBackground(Color.black); getContentPane().add(pnlFechaHora); lblFecha.setBounds(0,0,pnlFechaHora.getWidth(),30); lblFecha.setHorizontalAlignment(JLabel.CENTER); lblFecha.setBackground(Color.black); lblFecha.setForeground(Color.red); lblFecha.setFont(new Font("Times New Roman", 1, 30)); pnlFechaHora.add(lblFecha); lblHora.setBounds(0,30,pnlFechaHora.getWidth(),40); lblHora.setHorizontalAlignment(JLabel.CENTER); lblHora.setBackground(Color.black); lblHora.setForeground(Color.red); lblHora.setFont(new Font("Arial Black", 1, 40)); pnlFechaHora.add(lblHora); pnlRegistro.setBounds(5,80,getWidth()-20,70); //pnlRegistro.setBackground(Color.black); pnlRegistro.setBorder(new TitledBorder("Ingrese su código personal para registrar su ingreso o salida de la oficina")); //pnlRegistro.setBorder(new javax.swing.border.BevelBorder(1)); getContentPane().add(pnlRegistro); lblCarnet.setFont(new Font("Arial",0,20)); pnlRegistro.add(lblCarnet); txtClave.setBounds(200,25,100,30); txtClave.setEchoChar('*'); txtClave.setFont(new Font("Arial",0,20)); txtClave.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) Grabar(); } }); } }</pre>			
Observaciones:			



6.2.2.8 CLASE FRMASISTENCIA

EAGLEEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmAsistencia.java	Tipo: Public	
Descripción: <i>Clase encargada de recoger la información de asistencia del personal por código de ingreso.</i>			
DESCRIPCIÓN DE CLASE			
<pre>pnlRegistro.add(txtClave); pnlSaludo.setBounds(5,140,getWidth()-20,60);pnlSaludo.setVisible(false); pnlSaludo.setBorder(new TitledBorder("")); pnlSaludo.setBorder(new javax.swing.border.BevelBorder(1)); getContentPane().add(pnlSaludo); lblSaludo.setBounds(20,5,60,16);lblSaludo.setForeground(Color.blue); lblSaludo.setFont(new Font("Arial Black", 0, 16));pnlSaludo.add(lblSaludo); lblMensaje.setBounds(0,20,pnlSaludo.getWidth(),20); lblMensaje.setHorizontalAlignment(JLabel.CENTER);lblMensaje.setForeground(Color.blue); lblMensaje.setFont(new Font("Arial", 0, 20));pnlSaludo.add(lblMensaje); lblRegistro.setBounds(0,40,pnlSaludo.getWidth(),20); lblRegistro.setHorizontalAlignment(JLabel.CENTER);lblRegistro.setForeground(Color.red); lblRegistro.setFont(new Font("Arial Black", 0, 20));pnlSaludo.add(lblRegistro); btnAceptar.addActionListener(new ActionListener() { public void actionPerformed(ActionEvent evt) { Grabar();}); getContentPane().add(btnAceptar); } private void PonerHora(){ final SimpleDateFormat formatoFecha = new SimpleDateFormat("EEEE dd MMMM yyyy"); Date fecha = new Date(); final SimpleDateFormat formatoHora = new SimpleDateFormat("HH:mm:ss"); java.util.Timer timer = new java.util.Timer(); TimerTask horaActual = new TimerTask(){ public void run(){ Date fecha = new Date();lblFecha.setText(formatoFecha.format(fecha)); lblHora.setText(formatoHora.format(fecha));}; timer.schedule(horaActual,0,1000);} private void Grabar(){ if (txtClave.getText().equals("")) {JOptionPane.showMessageDialog(null,"Especifique Código","Error", JOptionPane.ERROR_MESSAGE);return;} datPersonal per = new datPersonal(); if (per.Consultar(txtClave.getText())){ if (per.getEstado() == false){ JOptionPane.showMessageDialog(null,"El Usuario Esta Inactivo","Error", JOptionPane.ERROR_MESSAGE);return;}} else {JOptionPane.showMessageDialog(null,"Codigo no existe. ","Mensaje", JOptionPane.INFORMATION_MESSAGE);return;} datAsistencia dat = new datAsistencia(Integer.parseInt(txtClave.getText())); if (dat.Grabar()) { btnAceptar.setVisible(false);pnlSaludo.setVisible(true); lblMensaje.setText(per.getNombreCorto()); lblRegistro.setText("Se registró a las " + dat.getUltimaMarca()); int nSegundos = 2000; Date timeToRun = new Date(System.currentTimeMillis() + nSegundos); java.util.Timer timer = new java.util.Timer(); timer.schedule(new TimerTask(){ public void run(){ pnlSaludo.setVisible(false);btnAceptar.setVisible(true); txtClave.setText("");txtClave.requestFocus();},timeToRun); } }</pre>			
Observaciones:			

6.2.2.9 CLASE FRMCARGO

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1
		Fecha Diseño: 2/08/2004
		Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo
Capa: Clientes	Nombre de la Clase: frmCargo.java	Tipo: Public
Descripción: Clase encargada de ingresar modificar o eliminar los cargos existentes.		
DESCRIPCIÓN DE CLASE		
<pre> class frmCargo extends clsMantenimientoTipo2{ frmCargo(){ super("Mantenimiento Cargos",420,170); SQLconsulta = "Select codigo,descripcion From CPM_Cargo Where estado = true"; Reporte = "cargo";HabilitarEntrada();HabilitarBotones(); } //METODOS void Grabar(){ super.Grabar();if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre", "Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones(); return;} if (txtMonto.getText().equals("")) txtMonto.setText("0"); datCargo dat = new datCargo();dat.setDescripcion(txtDescripcion.getText()); dat.setSueldo(Double.parseDouble(txtMonto.getText()));if (getEsNuevo()) { if (dat.Grabar())txtCodigo.setText(dat.getUltimoCodigo()); } else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();} else { dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado()); if (!dat.Actualizar()){ JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones(); }} void Consultar(String Codigo) { if (Codigo.equals("")) return; datCargo dat = new datCargo();if (dat.Consultar(Codigo)){ txtCodigo.setText("" + dat.getCodigo()); txtDescripcion.setText("" + dat.getDescripcion());txtMonto.setText("" + dat.getSueldo()); setEstado(dat.getEstado());setRegistro(true);HabilitarBotones();} else{LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE);} void Eliminar(){ int res = JOptionPane.showConfirmDialog(null,"¿Está seguro de eliminar el registro?", "Mensaje",JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION){ datCargo dat = new datCargo(); dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar())LimpiarEntrada();}} } </pre>		
Observaciones: Hereda de clsMantenimientoTipo2.java		



6.2.2.10 CLASE FRMCIUDAD

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmCiudad.java	Tipo: Public	
Descripción: Clase encargada de ingresar modificar o eliminar las ciudades existentes.			
DESCRIPCIÓN DE CLASE			
<pre>class frmCiudad extends clsMantenimientoTipo1 { frmCiudad(){ super("Mantenimiento Ciudad",420,170); lblDescripcion.setText("Nombre:"); SQLconsulta = "Select codigo,nombre From Cpm_Ciudad Where estado = true"; Reporte = "ciudad"; HabilitarEntrada();HabilitarBotones(); } //METODOS void Grabar(){ super.Grabar(); if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre","Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada(); HabilitarBotones();return;} datCiudad dat = new datCiudad(); dat.setNombre(txtDescripcion.getText()); if (getEsnuevo()) {if (dat.Grabar()) txtCodigo.setText(dat.getUltimoCodigo()); else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();} else { dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado());if (!dat.Actualizar()){ JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones(); }}} } void Consultar(String Codigo) { if (Codigo.equals(""))return; datCiudad dat = new datCiudad(); if (dat.Consultar(Codigo)){ txtCodigo.setText("" + dat.getCodigo());txtDescripcion.setText("" + dat.getNombre()); setEstado(dat.getEstado());setRegistro(true);HabilitarBotones();} else{ LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE); } } void Eliminar(){ int res = JOptionPane.showConfirmDialog(null,"¿Está seguro de eliminar el registro?","Mensaje",JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION){ datCiudad dat = new datCiudad();dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar()) LimpiarEntrada();}}} }</pre>			
Observaciones: Hereda de clsMantenimientoTipo1.java			



6.2.2.11 CLASE FRMDEPARTAMENTO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmDepartamento.java	Tipo: Public	
Descripción: Clase encargada de ingresar modificar o eliminar los departamentos existentes.			
DESCRIPCIÓN DE CLASE			
<pre>class frmDepartamento extends clsMantenimientoTipo1 { frmDepartamento() { super("Mantenimiento Departamento",420,170); lblDescripcion.setText("Nombre:"); SQLconsulta = "Select codigo,descripcion From CPM_Departamento Where estado = true"; Reporte = "departamento";HabilitarEntrada();HabilitarBotones();} //METODOS void Grabar(){super.Grabar(); if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre", "Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();return;} datDepartamento dat = new datDepartamento(); dat.setDescripcion(txtDescripcion.getText()); if (getEsNuevo()) {if (dat.Grabar()) txtCodigo.setText(dat.getUltimoCodigo()); } else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido", "Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();} else { dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado()); if (!dat.Actualizar()) { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido", "Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones(); } } void Consultar(String Codigo) {if (Codigo.equals(""))return; datDepartamento dat = new datDepartamento(); if (dat.Consultar(Codigo)){ txtCodigo.setText("" + dat.getCodigo());txtDescripcion.setText("" + dat.getDescripcion()); setEstado(dat.getEstado());setRegistro(true);HabilitarBotones();} else{LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE);} void Eliminar(){ int res = JOptionPane.showConfirmDialog(null,"¿Está seguro de eliminar el registro?", "Mensaje",JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION){ datDepartamento dat = new datDepartamento(); dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar()) LimpiarEntrada(); } } } }</pre>			
Observaciones: Hereda de clsMantenimientoTipo1.java			

6.2.2.12 CLASE FRMEMPRESA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmEmpresa.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar las empresas existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre> class frmEmpresa extends clsMantenimientoTipo3 { // Etiquetas private clsLabel lblFax = new clsLabel("Fax:",lblCedRuc.getX(),txtTelefono.getY(),80,20); private clsLabel lblWeb = new clsLabel("Página WEB:",lblEmail.getX(),lblEmail.getY()+22,80,20); // Campos private clsTextField txtFax = new clsTextField(txtCedRuc.getX(),lblFax.getY(),80,20); private clsTextField txtWeb = new clsTextField(txtNombre.getX(),lblWeb.getY(),400,20); frmEmpresa() { super("Mantenimiento Empresa",510,250); SQLconsulta = "Select codigo,nombre From CPm_Empresa Where estado = true"; Componentes();HabilitarEntrada();HabilitarBotones();Reporte = "empresa"; } //METODOS private void Componentes(){ lblCedRuc.setText("Ruc:"); cntEstado.setLocation(cntEstado.getX(),cntEstado.getY()+22);btnExaminarCed.setVisible(false); getContentPane().add(lblFax);getContentPane().add(lblWeb); txtFax.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER)transferFocus();}); getContentPane().add(txtFax); txtWeb.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER)transferFocus();}); getContentPane().add(txtWeb); } void LimpiarEntrada(){super.LimpiarEntrada();txtFax.setText("");txtWeb.setText("");} void HabilitarEntrada(){ super.HabilitarEntrada();txtFax.Habilitar(getEditando());txtWeb.Habilitar(getEditando());} void Grabar(){super.Grabar(); if (txtNombre.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre","Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();return;} datEmpresa dat = new datEmpresa();dat.setRuc(txtCedRuc.getText()); dat.setNombre(txtNombre.getText());dat.setDireccion(txtDireccion.getText()); dat.setTelefono(txtTelefono.getText());dat.setFax(txtFax.getText()); dat.setEmail(txtEmail.getText());dat.setWeb(txtWeb.getText()); if (getEsNuevo()) { if (dat.Grabar()) txtCodigo.setText(dat.getUltimoCodigo()); } else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar",JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();} else { dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado()); if (!dat.Actualizar()){ JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar",JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();}}} </pre>			
Observaciones: Hereda de clsMantenimientoTipo3.java			



6.2.2.13 CLASE FRMINCENTIVO

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmIncentivo.java	Tipo: Public	
Descripción: Clase encargada de ingresar modificar o eliminar los incentivos existentes.			
DESCRIPCIÓN DE CLASE			
<pre>class frmIncentivo extends JFrame{ // Propiedades // Etiquetas private clsLabel lblNumero = new clsLabel("Numero:",10,10,80,20); private clsLabel lblFecha = new clsLabel("Fecha:",10,35,80,20); private clsLabel lblPersonal = new clsLabel("Cod.Personal:",10,60,80,20); private clsLabel lblTipo = new clsLabel("Tipo Incentivo:",10,85,180,20); private clsLabel lblMonto = new clsLabel("Monto:",10,110,80,20); private clsLabel lblMotivo = new clsLabel("Motivo:",10,135,80,20); private clsEstado cntEstado = new clsEstado(10,200); // Campos private clsTextField txtNumero = new clsTextField(90,10,40,20); private clsCampoFecha txtFecha = new clsCampoFecha(90,35,80,20); private clsTextField txtCodigo = new clsTextField(90,60,40,20); private clsTextField txtNombre = new clsTextField(140,60,200,20); private clsCombo cmbTipo = new clsCombo("TipoIncentivo",90,85,250,20); private clsTextField txtMonto = new clsTextField(90,110,80,20); private JTextArea txtMotivo = new JTextArea(); // Botones JPanel pnlBotones = new JPanel(null,true); private clsBoton btnNuevo = new clsBoton("Nuevo","N","Nuevo registro","new",0,0); private clsBoton btnGrabar = new clsBoton("Grabar","G","Guardar registro","save",0,0); private clsBoton btnModificar = new clsBoton("Modificar","M","Modificar registro","edit24",0,0); private clsBoton btnCancelar = new clsBoton("Cancelar","C","Cancelar edición","undo",0,0); private clsBoton btnCerrar = new clsBoton("Cerrar","C","Cerrar formulario","close",0,0); frmIncentivo(){ super("Incentivos",false,true,false,true);Entorno();Componentes(); HabilitarEntrada(false);show();} //METODOS private void Entorno(){ setBounds(10,10,370,300);getContentPane().setLayout(null); setFrameIcon(new ImageIcon("imagenes/face01.gif"));} private void Componentes(){ // Etiquetas getContentPane().add(lblNumero);getContentPane().add(lblFecha); getContentPane().add(lblPersonal);getContentPane().add(lblTipo); getContentPane().add(lblMotivo);getContentPane().add(lblMonto); getContentPane().add(cntEstado); txtNumero.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) Consultar(Integer.parseInt(txtNumero.getText()));}); getContentPane().add(txtNumero); txtFecha.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) txtCodigo.requestFocus(); } }); getContentPane().add(txtFecha); } }</pre>			
Observaciones:			

6.2.2.14 CLASE FRMINCENTIVO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: frmIncentivo.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar los incentivos existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>txtCodigo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) ConsultarPersona(Integer.parseInt(txtCodigo.getText())); cmbTipo.requestFocus(); } }); getContentPane().add(txtCodigo);txtNombre.setEnabled(false);getContentPane().add(txtNombre); cmbTipo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) txtMonto.requestFocus(); } }); getContentPane().add(cmbTipo);txtMonto.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) txtMotivo.requestFocus(); } }); getContentPane().add(txtMonto);txtMotivo.setLineWrap(true); txtMotivo.setWrapStyleWord(true);JScrollPane scrMotivo = new JScrollPane(txtMotivo); scrMotivo.setBounds(90,135,250,60);getContentPane().add(scrMotivo); getContentPane().add(btnNuevo); btnNuevo.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Nuevo(); } }); getContentPane().add(btnGrabar);btnGrabar.setVisible(false); btnGrabar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Grabar(); } }); getContentPane().add(btnModificar); btnModificar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Modificar(); } }); getContentPane().add(btnCancelar); btnCancelar.setVisible(false);btnCancelar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Cancelar(); } }); getContentPane().add(btnCerrar);btnCerrar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ dispose();setVisible(false); } }); UbicarBotones(10,getHeight()-btnNuevo.getHeight()-40); private void UbicarBotones(int x,int y){ btnNuevo.setLocation(x,y);btnGrabar.setLocation(x,y); btnModificar.setLocation(btnNuevo.getX() + btnNuevo.getWidth(),y); btnCancelar.setLocation(btnModificar.getX(),btnModificar.getY()); btnCerrar.setLocation(btnModificar.getX() + btnModificar.getWidth(),y); private void VerBotones(boolean lEditando){btnGrabar.setVisible(lEditando); btnCancelar.setVisible(!lEditando);btnNuevo.setVisible(!lEditando); btnModificar.setVisible(!lEditando);btnCerrar.setEnabled(!lEditando); btnModificar.setEnabled(false); private void LimpiarEntrada(){txtNumero.setText("");txtFecha.setText("");txtCodigo.setText(""); txtNombre.setText("");cmbTipo.setSelectedIndex(-1);txtMotivo.setText("");txtMonto.setText(""); cntEstado.setEstado(true);} }</pre>			
Observaciones:			

6.2.2.15 CLASE FRMINCENTIVO

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmIncentivo.java	Tipo: Public	
Descripción: Clase encargada de ingresar modificar o eliminar los incentivos existentes.			
DESCRIPCIÓN DE CLASE			
<pre>private void HabilitarEntrada(boolean lEstado){ txtNumero.Habilitar(!lEstado);txtFecha.Habilitar(lEstado); txtCodigo.Habilitar(lEstado);txtNombre.Habilitar(lEstado);cmbTipo.Habilitar(lEstado); txtMonto.Habilitar(lEstado);cntEstado.Habilitar(lEstado);} private void Nuevo(){ VerBotones(true);LimpiarEntrada();HabilitarEntrada(true);cntEstado.Habilitar(false);} private void Modificar(){ VerBotones(true);HabilitarEntrada(true);} private void Cancelar(){ VerBotones(false);HabilitarEntrada(false);LimpiarEntrada();} private void Grabar(){ if (txtCodigo.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre","Error", JOptionPane.ERROR_MESSAGE);return;} if (cmbTipo.getSelectedIndex() < 0) { JOptionPane.showMessageDialog(null,"Especifique un Tipo de Incentivo","Error", JOptionPane.ERROR_MESSAGE);return;} datIncentivo dat = new datIncentivo();dat.setFecha(txtFecha.getText()); dat.setPersonal(Integer.parseInt(txtCodigo.getText())); dat.setTipo(cmbTipo.getId(cmbTipo.getSelectedItem()+""));dat.setMotivo(txtMotivo.getText()); dat.setMonto(Integer.parseInt(txtMonto.getText())); if (cntEstado.getHabilitar()) { dat.setNumero(Integer.parseInt(txtNumero.getText()));dat.setEstado(cntEstado.getEstado()); if (dat.Actualizar()) { HabilitarEntrada(false);VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE);} else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar",JOptionPane.ERROR_MESSAGE); } } else {if (dat.Grabar()) { txtNumero.setText(dat.getUltimoNumero());HabilitarEntrada(false); VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE);} else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE);} } } private void Consultar(int lnCodigo) {if (lnCodigo < 1){return;} datIncentivo dat = new datIncentivo(); if (dat.Consultar(lnCodigo)){ txtNumero.setText(""+ dat.getNumero());txtFecha.setText(dat.getFecha()); txtCodigo.setText(""+ dat.getPersonal());cmbTipo.setSelectedIndex(dat.getTipo()-1); txtMotivo.setText(""+ dat.getMotivo());txtMonto.setText(""+ dat.getMonto()); cntEstado.setEstado(dat.getEstado()); } else{LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", private void ConsultarPersona(int lnCodigo) {</pre>			
Observaciones:			

6.2.2.16 CLASE FRMJUSTIFICA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: frmJustifica.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar las jstificaciones existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre> class frmJustifica extends JInternalFrame{ // Etiquetas private clsLabel lblNumero = new clsLabel("Numero:",10,10,80,20); private clsLabel lblFecha = new clsLabel("Fecha:",10,35,80,20); private clsLabel lblPersonal = new clsLabel("Cod.Personal:",10,60,80,20); private clsLabel lblTipo = new clsLabel("Tipo Justifi.:",10,85,180,20); private clsLabel lblMotivo = new clsLabel("Motivo:",10,110,80,20); private clsEstado cntEstado = new clsEstado(10,175); // Campos private clsTextField txtNumero = new clsTextField(90,10,40,20); private clsCampoFecha txtFecha = new clsCampoFecha(90,35,80,20); private clsTextField txtCodigo = new clsTextField(90,60,40,20); private clsTextField txtNombre = new clsTextField(140,60,200,20); private clsCombo cmbTipo = new clsCombo("TipoJustifica",90,85,250,20); private JTextArea txtMotivo = new JTextArea(); // Botones JPanel pnlBotones = new JPanel(null,true); private clsBoton btnNuevo = new clsBoton("Nuevo","N","Nuevo registro","new",0,0); private clsBoton btnGrabar = new clsBoton("Grabar","G","Guardar registro","save",0,0); private clsBoton btnModificar = new clsBoton("Modificar","M","Modificar registro","edit24",0,0); private clsBoton btnCancelar = new clsBoton("Cancelar","C","Cancelar edición","undo",0,0); private clsBoton btnCerrar = new clsBoton("Cerrar","C","Cerrar formulario","close",0,0); frmJustifica(){ super("Justificativos",false,true,false,true);Entorno();Componentes(); HabilitarEntrada(false);show();} //METODOS private void Entorno(){ setBounds(10,10,370,300);getContentPane().setLayout(null); setFrameIcon(new ImageIcon("imagenes/face01.gif"));} private void Componentes(){ // Etiquetas getContentPane().add(lblNumero);getContentPane().add(lblFecha); getContentPane().add(lblPersonal);getContentPane().add(lblTipo); getContentPane().add(lblMotivo);getContentPane().add(cntEstado); // Campos txtNumero.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) Consultar(Integer.parseInt(txtNumero.getText()));} }); getContentPane().add(txtNumero);txtFecha.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) txtCodigo.requestFocus();} }); getContentPane().add(txtFecha);txtCodigo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) ConsultarPersona(Integer.parseInt(txtCodigo.getText())); cmbTipo.requestFocus();} }); getContentPane().add(txtCodigo);txtNombre.setEnabled(false);getContentPane().add(txtNombre); } } </pre>			
Observaciones:			

6.2.2.17 CLASE FRMJUSTIFICA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmJustifica.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar las jstificaciones existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>cmbTipo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) txtMotivo.requestFocus();}}); getContentPane().add(cmbTipo);txtMotivo.setLineWrap(true); txtMotivo.setWrapStyleWord(true);JScrollPane scrMotivo = new JScrollPane(txtMotivo); scrMotivo.setBounds(90,110,250,60);getContentPane().add(scrMotivo); getContentPane().add(btnNuevo); btnNuevo.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){Nuevo();}}); getContentPane().add(btnGrabar);btnGrabar.setVisible(false); btnGrabar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){Grabar();}}); getContentPane().add(btnModificar);btnModificar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Modificar();}}); getContentPane().add(btnCancelar);btnCancelar.setVisible(false); btnCancelar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Cancelar();}}); getContentPane().add(btnCerrar);btnCerrar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ dispose();setVisible(false);}}); UbicarBotones(10,getHeight()-btnNuevo.getHeight()-40);} private void UbicarBotones(int x,int y){btnNuevo.setLocation(x,y); btnGrabar.setLocation(x,y);btnModificar.setLocation(btnNuevo.getX() + btnNuevo.getWidth(),y); btnCancelar.setLocation(btnModificar.getX(),btnModificar.getY()); btnCerrar.setLocation(btnModificar.getX() + btnModificar.getWidth(),y);} private void VerBotones(boolean lEditando){btnGrabar.setVisible(!lEditando); btnCancelar.setVisible(!lEditando);btnNuevo.setVisible(!lEditando); btnModificar.setVisible(!lEditando);btnCerrar.setEnabled(!lEditando); btnModificar.setEnabled(false);} private void LimpiarEntrada(){ txtNumero.setText("");txtFecha.setText("");txtCodigo.setText("");txtNombre.setText(""); cmbTipo.setSelectedIndex(-1);txtMotivo.setText("");cntEstado.setEstado(true);} private void HabilitarEntrada(boolean lEstado){ txtNumero.Habilitar(!lEstado);txtFecha.Habilitar(!lEstado);txtCodigo.Habilitar(!lEstado); txtNombre.Habilitar(!lEstado);cmbTipo.Habilitar(!lEstado);cntEstado.Habilitar(!lEstado);} private void Nuevo(){ VerBotones(true);LimpiarEntrada();HabilitarEntrada(true);cntEstado.Habilitar(false);} private void Modificar(){ VerBotones(true); HabilitarEntrada(true); }</pre>			
Observaciones:			



6.2.2.18 CLASE FRM.JUSTIFICA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: frmJustifica.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar las jstificaciones existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>private void Cancelar(){ VerBotones(false);HabilitarEntrada(false);LimpiarEntrada();} private void Grabar(){ if (txtCodigo.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre","Error", JOptionPane.ERROR_MESSAGE);return;} if (cmbTipo.getSelectedIndex() < 0) { JOptionPane.showMessageDialog(null,"Especifique un Tipo de Justificativos","Error", JOptionPane.ERROR_MESSAGE);return; } datJustifica dat = new datJustifica();dat.setFecha(txtFecha.getText()); dat.setPersonal(Integer.parseInt(txtCodigo.getText())); dat.setTipo(cmbTipo.getId(cmbTipo.getSelectedItem()+""));dat.setMotivo(txtMotivo.getText()); if (cntEstado.getHabilitar()) { dat.setNumero(Integer.parseInt(txtNumero.getText()));dat.setEstado(cntEstado.getEstado()); if (dat.Actualizar()) { HabilitarEntrada(false);VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE);} else {JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); } else {if (dat.Grabar()) {txtNumero.setText(dat.getUltimoNumero()); HabilitarEntrada(false);VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE); }else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); } } private void Consultar(int lnCodigo) {if (lnCodigo < 1){return; } datJustifica dat = new datJustifica(); if (dat.Consultar(lnCodigo)){ txtNumero.setText("" + dat.getNumero());txtFecha.setText(dat.getFecha()); txtCodigo.setText("" + dat.getPersonal());cmbTipo.setSelectedIndex(dat.getTipo()-1); txtMotivo.setText("" + dat.getMotivo());cntEstado.setEstado(dat.getEstado());} else {LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe. ", "Mensaje", JOptionPane.INFORMATION_MESSAGE);} } private void ConsultarPersona(int lnCodigo) { if (lnCodigo < 1){return;} datPersonal dat = new datPersonal();if (dat.Consultar(lnCodigo)){ if (dat.getEstado()) { txtCodigo.setText("" + dat.getCodigo());txtNombre.setText("" + dat.getNombre());} else {txtCodigo.setText("");txtNombre.setText(""); JOptionPane.showMessageDialog(null,dat.getNombre() + " Esta Anulado","Error", JOptionPane.ERROR_MESSAGE); } } else{ txtNombre.setText(""); } }</pre>			
Observaciones:			

6.2.2.19 CLASE FRMMULTA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmMulta.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar las multas existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre> class frmMulta extends JFrame{ private clsLabel lblNumero = new clsLabel("Numero:",10,10,80,20); private clsLabel lblFecha = new clsLabel("Fecha:",10,35,80,20); private clsLabel lblPersonal = new clsLabel("Cod.Personal:",10,60,80,20); private clsLabel lblTipo = new clsLabel("Tipo de Multa:",10,85,180,20); private clsLabel lblMonto = new clsLabel("Monto:",10,110,80,20); private clsLabel lblMotivo = new clsLabel("Motivo:",10,135,80,20); private clsEstado cntEstado = new clsEstado(10,200); private clsTextField txtNumero = new clsTextField(90,10,40,20); private clsCampoFecha txtFecha = new clsCampoFecha(90,35,80,20); private clsTextField txtCodigo = new clsTextField(90,60,40,20); private clsTextField txtNombre = new clsTextField(140,60,200,20); private clsCombo cmbTipo = new clsCombo("TipoMulta",90,85,250,20); private clsTextField txtMonto = new clsTextField(90,110,80,20); private JTextArea txtMotivo = new JTextArea(); JPanel pnlBotones = new JPanel(null,true); private clsBoton btnNuevo = new clsBoton("Nuevo","N","Nuevo registro","new",0,0); private clsBoton btnGrabar = new clsBoton("Grabar","G","Guardar registro","save",0,0); private clsBoton btnModificar = new clsBoton("Modificar","M","Modificar registro","edit24",0,0); private clsBoton btnCancelar = new clsBoton("Cancelar","C","Cancelar edición","undo",0,0); private clsBoton btnCerrar = new clsBoton("Cerrar","C","Cerrar formulario","close",0,0); frmMulta(){ super("Multas",false,true,false,true);Entorno();Componentes();HabilitarEntrada(false);show(); } //METODOS private void Entorno(){ setBounds(10,10,370,300);getContentPane().setLayout(null);setFrameIcon(new ImageIcon("imagenes/face01.gif")); private void Componentes(){ getContentPane().add(lblNumero);getContentPane().add(lblFecha); getContentPane().add(lblPersonal);getContentPane().add(lblTipo);getContentPane().add(lblMotivo); getContentPane().add(lblMonto);getContentPane().add(cntEstado); txtNumero.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) Consultar(Integer.parseInt(txtNumero.getText()));}); getContentPane().add(txtNumero);txtFecha.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) txtCodigo.requestFocus();});getContentPane().add(txtFecha); txtCodigo.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) ConsultarPersona(Integer.parseInt(txtCodigo.getText())); cmbTipo.requestFocus();}); getContentPane().add(txtCodigo);txtNombre.setEnabled(false);getContentPane().add(txtNombre); cmbTipo.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER)txtMonto.requestFocus();}); getContentPane().add(cmbTipo);txtMonto.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER)txtMotivo.requestFocus();}); getContentPane().add(txtMonto);txtMotivo.setLineWrap(true);txtMotivo.setWrapStyleWord(true); JScrollPane scrMotivo = new JScrollPane(txtMotivo); scrMotivo.setBounds(90,135,250,60); getContentPane().add(scrMotivo); </pre>			
Observaciones:			

6.2.2.20 CLASE FRMMULTA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmMulta.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar las multas existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>// Botones getContentPane().add(btnNuevo);btnNuevo.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Nuevo();}}); getContentPane().add(btnGrabar);btnGrabar.setVisible(false); btnGrabar.addActionListener(new ActionListener(){public void actionPerformed(ActionEvent e){ Grabar();}}); getContentPane().add(btnModificar);btnModificar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Modificar();}}); getContentPane().add(btnCancelar);btnCancelar.setVisible(false); btnCancelar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Cancelar();}}); getContentPane().add(btnCerrar);btnCerrar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ dispose();setVisible(false);});UbicarBotones(10,getHeight()-btnNuevo.getHeight()-40);} private void UbicarBotones(int x,int y){btnNuevo.setLocation(x,y);btnGrabar.setLocation(x,y); btnModificar.setLocation(btnNuevo.getX() + btnNuevo.getWidth(),y); btnCancelar.setLocation(btnModificar.getX(),btnModificar.getY()); btnCerrar.setLocation(btnModificar.getX() + btnModificar.getWidth(),y);} private void VerBotones(boolean lEditando){btnGrabar.setVisible(lEditando);btnCancelar.setVisible(lEditando); btnNuevo.setVisible(!lEditando);btnModificar.setVisible(!lEditando);btnCerrar.setEnabled(!lEditando);btnModificar.setEnabled(false);} private void LimpiarEntrada(){txtNumero.setText(""); txtFecha.setText("");txtCodigo.setText(""); txtNombre.setText("");cmbTipo.setSelectedIndex(-1); txtMotivo.setText("");txtMonto.setText("");cntEstado.setEstado(true);} private void HabilitarEntrada(boolean lEstado){ txtNumero.Habilitar(!lEstado);txtFecha.Habilitar(!lEstado); txtCodigo.Habilitar(!lEstado);txtNombre.Habilitar(!lEstado); cmbTipo.Habilitar(!lEstado);txtMonto.Habilitar(!lEstado);cntEstado.Habilitar(!lEstado);} private void Nuevo(){ VerBotones(true); LimpiarEntrada(); HabilitarEntrada(true); cntEstado.Habilitar(false); } private void Modificar(){ VerBotones(true); HabilitarEntrada(true); }</pre>			
Observaciones:			





6.2.2.21 CLASE FRMMULTA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmMulta.java	Tipo: Public	
Descripción: Clase encargada de ingresar modificar o eliminar las multas existentes.			
DESCRIPCIÓN DE CLASE			
<pre>private void Cancelar(){ VerBotones(false);HabilitarEntrada(false);LimpiarEntrada();} private void Grabar(){if (txtCodigo.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre","Error", JOptionPane.ERROR_MESSAGE);return;} if (cmbTipo.getSelectedIndex() < 0) { JOptionPane.showMessageDialog(null,"Especifique un Tipo de Multa","Error", JOptionPane.ERROR_MESSAGE);return;} datMulta dat = new datMulta();dat.setFecha(txtFecha.getText()); dat.setPersonal(Integer.parseInt(txtCodigo.getText())); dat.setTipo(cmbTipo.getId(cmbTipo.getSelectedIndex()+""));dat.setMotivo(txtMotivo.getText()); dat.setMonto(Integer.parseInt(txtMonto.getText()));if (cntEstado.getHabilitar()) { dat.setNumero(Integer.parseInt(txtNumero.getText()));dat.setEstado(cntEstado.getEstado()); if (dat.Actualizar()) { HabilitarEntrada(false);VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE);} else {JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); }} else {if (dat.Grabar()) { txtNumero.setText(dat.getUltimoNumero());HabilitarEntrada(false);VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE); }else {JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); }}} private void Consultar(int lnCodigo) { if (lnCodigo < 1){return;} datMulta dat = new datMulta(); if (dat.Consultar(lnCodigo)){ txtNumero.setText(""+ dat.getNumero());txtFecha.setText(dat.getFecha()); txtCodigo.setText(""+ dat.getPersonal());cmbTipo.setSelectedIndex(dat.getTipo()-1); txtMotivo.setText(""+ dat.getMotivo());txtMonto.setText(""+ dat.getMonto()); cntEstado.setEstado(dat.getEstado());} else{LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE); }} private void ConsultarPersona(int lnCodigo) {if (lnCodigo < 1){return;} datPersonal dat = new datPersonal(); if (dat.Consultar(lnCodigo)){ if (dat.getEstado()) { txtCodigo.setText(""+ dat.getCodigo()); txtNombre.setText(""+ dat.getNombre()); }else{ txtCodigo.setText("");txtNombre.setText(""); JOptionPane.showMessageDialog(null,dat.getNombre() + " Esta Anulado","Error", JOptionPane.ERROR_MESSAGE); } } else{txtNombre.setText("");}}</pre>			
Observaciones:			



6.2.2.22 CLASE FRMPERSONAL

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmPersonal.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar el personal existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre> class frmPersonal extends clsMantenimientoTipo3{ private JLabel lblFoto = new JLabel(); private clsLabel lblApellido = new clsLabel("Apellido:",10,54,80,20); private clsLabel lblNombreCorto = new clsLabel("Nombre Corto:",10,76,80,20); private clsLabel lblClave = new clsLabel("Clave:",260,76,60,20); private clsLabel lblEstadoCivil = new clsLabel("E.Civil:",220,120,80,20); private clsLabel lblFNacimiento = new clsLabel("F.Nacimiento:",10,144,80,20); private clsLabel lblLNacimiento = new clsLabel("Lugar Nac.:",220,144,80,20); private clsLabel lblProfesion = new clsLabel("Profesión:",10,166,80,20); private clsLabel lblDpto = new clsLabel("Depto.:",220,166,60,20); private clsLabel lblCargo = new clsLabel("Cargo:",240,188,60,20); private clsLabel lblFEntrada = new clsLabel("Fec.Entrada:",10,188,80,20); private clsLabel lblFSalida = new clsLabel("Salida:",150,188,60,20); private clsTextField txtApellido = new clsTextField(90,54,275,20); private clsTextField txtNombreCorto = new clsTextField(90,76,150,20); private clsTextField txtClave = new clsTextField(305,76,60,20); private clsEstadoCivil cmbEstadoCivil= new clsEstadoCivil(265,120,100,20); private clsCampoFecha txtFNacimiento= new clsCampoFecha(90,144,60,20); private clsCombo cmbLNacimiento = new clsCombo("Ciudad",265,144,100,20); private clsCombo cmbProfesion = new clsCombo("Profesion",90,166,100,20); private clsCombo cmbDpto = new clsCombo("Departamento",265,166,100,20); private clsCombo cmbCargo = new clsCombo("Cargo",265,188,100,20); private clsCampoFecha txtFEntrada = new clsCampoFecha(90,188,60,20); private clsCampoFecha txtFSalida = new clsCampoFecha(200,188,60,20); frmPersonal(){ super("Mantenimiento Personal",510,330);Componentes(); SQLconsulta = "Select codigo,nombrecorto From Cpm_Persona Where estado = true"; Reporte = "personal";HabilitarEntrada();HabilitarBotones();} private void Componentes(){ lblCedRuc.setText("Cédula:");txtNombre.setSize(275,20); lblDireccion.setLocation(10,98);txtDireccion.setBounds(90,98,275,20); lblTelefono.setLocation(10,120);txtTelefono.setLocation(90,120); lblEmail.setLocation(10,210);txtEmail.setBounds(90,210,275,20); cntEstado.setLocation(cntEstado.getX(),232); lblFoto.setIcon(new ImageIcon(""));lblFoto.setBounds(getWidth()-100,5, 100, 100); getContentPane().add(lblFoto);getContentPane().add(lblApellido); getContentPane().add(lblNombreCorto);getContentPane().add(lblClave); getContentPane().add(lblEstadoCivil);getContentPane().add(lblProfesion); getContentPane().add(lblFNacimiento);getContentPane().add(lblLNacimiento); getContentPane().add(lblDpto);getContentPane().add(lblCargo); getContentPane().add(lblFEntrada);getContentPane().add(lblFSalida); txtApellido.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(txtApellido); txtNombreCorto.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(txtNombreCorto); txtClave.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(txtClave); </pre>			
Observaciones: Hereda de clsMantenimientoTipo3.java			

6.2.2.23 CLASE FRMPERSONAL

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: frmPersonal.java	Tipo: Public	
Descripción: Clase encargada de ingresar modificar o eliminar el personal existentes.			
DESCRIPCIÓN DE CLASE			
<pre> cmbEstadoCivil.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(cmbEstadoCivil); cmbProfesion.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(cmbProfesion); txtFNacimiento.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(txtFNacimiento); cmbLNacimiento.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(cmbLNacimiento); cmbDpto.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(cmbDpto); cmbCargo.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(cmbCargo); txtFEntrada.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(txtFEntrada); txtFSalida.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(txtFSalida);} void LimpiarEntrada(){ super.LimpiarEntrada();txtApellido.setText("");txtNombreCorto.setText("");txtClave.setText(""); cmbEstadoCivil.setSelectedIndex(-1);cmbProfesion.setSelectedIndex(-1);txtFNacimiento.setText(""); cmbLNacimiento.setSelectedIndex(-1);cmbDpto.setSelectedIndex(-1); cmbCargo.setSelectedIndex(-1);txtFEntrada.setText("");txtFSalida.setText("");} void HabilitarEntrada(){ super.HabilitarEntrada();txtApellido.Habilitar(getEditando());txtNombreCorto.Habilitar(getEditando()); txtClave.Habilitar(getEditando());cmbEstadoCivil.Habilitar(getEditando());cmbProfesion.Habilitar(getEditando()); txtFNacimiento.Habilitar(getEditando());cmbLNacimiento.Habilitar(getEditando()); cmbDpto.Habilitar(getEditando());cmbCargo.Habilitar(getEditando());txtFEntrada.Habilitar(getEditando()); txtFSalida.Habilitar(getEditando());txtFSalida.setVisible(!getEsnuevo());txtFSalida.setVisible(!getEsnuevo());} void Grabar(){ super.Grabar();if (txtClave.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique una clave","Error",JOptionPane.ERROR_MESSAGE); txtClave.requestFocus(); setEditando(true); HabilitarEntrada(); HabilitarBotones(); return; } } </pre>			
Observaciones: Hereda de clsMantenimientoTipo3.java			



6.2.2.24 CLASE FRMPERSONAL

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmPersonal.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar el personal existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre> if (txtApellido.getText().equals("")) { JOptionPane.showMessageDialog(null, "Especifique un apellido", "Error", JOptionPane.ERROR_MESSAGE); txtApellido.requestFocus(); setEditando(true); HabilitarEntrada(); HabilitarBotones(); return; } if (txtNombreCorto.getText().equals("")) { JOptionPane.showMessageDialog(null, "Especifique un nombre corto", "Error", JOptionPane.ERROR_MESSAGE); txtNombreCorto.requestFocus(); setEditando(true); HabilitarEntrada(); HabilitarBotones(); return; } datPersonal dat = new datPersonal(); dat.setCedula(txtCedRuc.getText()); dat.setNombre(txtNombre.getText()); dat.setApellido(txtApellido.getText()); dat.setNombreCorto(txtNombreCorto.getText()); dat.setClave(txtClave.getText()); dat.setEstadoCivil(cmbEstadoCivil.getId()); dat.setDireccion(txtDireccion.getText()); dat.setTelefono(txtTelefono.getText()); dat.setEmail(txtEmail.getText()); dat.setFNacimiento(txtFNacimiento.getText()); dat.setFEntrada(txtFEntrada.getText()); dat.setProfesion(cmbProfesion.getId(cmbProfesion.getSelectedIndex())); dat.setCargo(cmbCargo.getId(cmbCargo.getSelectedIndex())); dat.setDpto(cmbDpto.getId(cmbDpto.getSelectedIndex())); dat.setLNacimiento(cmbLNacimiento.getId(cmbLNacimiento.getSelectedIndex())); if (getEsNuevo()) { if (dat.Grabar()) txtCodigo.setText(dat.getUltimoCodigo()); else { JOptionPane.showMessageDialog(null, "Intento de Grabación Fallido", "Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true); HabilitarEntrada(); HabilitarBotones(); } } else { dat.setCodigo(txtCodigo.getText()); dat.setEstado(getEstado()); if (!dat.Actualizar()) { JOptionPane.showMessageDialog(null, "Intento de Grabación Fallido", "Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true); HabilitarEntrada(); HabilitarBotones(); } } void Consultar(String Codigo) { if (Codigo.equals("")) return; datPersonal dat = new datPersonal(); if (dat.Consultar(Codigo)) { txtCodigo.setText(dat.getCodigo()); txtNombre.setText(dat.getNombre()); txtApellido.setText(dat.getApellido()); txtNombreCorto.setText(dat.getNombreCorto()); txtClave.setText(dat.getClave()); txtDireccion.setText(dat.getDireccion()); txtCedRuc.setText(dat.getCedula()); txtTelefono.setText(dat.getTelefono()); txtEmail.setText(dat.getEmail()); txtFEntrada.setText(dat.getFEntrada()); txtFSalida.setText(dat.getFSalida()); txtFNacimiento.setText(dat.getFNacimiento()); cmbEstadoCivil.setSelectedIndex(dat.getEstadoCivil()-1); cmbLNacimiento.setSelectedIndex(dat.getLNacimiento()); cmbDpto.setSelectedIndex(dat.getDpto()); cmbCargo.setSelectedIndex(dat.getCargo()); cmbProfesion.setSelectedIndex(dat.getProfesion()); cntEstado.setSelectedIndex(dat.getEstado()); lblFoto.setIcon(new ImageIcon(dat.getFoto())); setRegistro(true); HabilitarBotones(); } else { LimpiarEntrada(); JOptionPane.showMessageDialog(null, "Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE); } void Eliminar() { int res = JOptionPane.showConfirmDialog(null, "¿Está seguro de eliminar el registro?", "Mensaje", JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION) { datPersonal dat = new datPersonal(); dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar()) LimpiarEntrada(); } } </pre>			
Observaciones: Hereda de clsMantenimientoTipo3.java			



6.2.2.25 CLASE FRMPROFESION

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: frmProfesion.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar las profesiones existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre> class frmProfesion extends clsMantenimientoTipo1 { private clsLabel lblSiglas = new clsLabel("Siglas:",10,54,80,20); private clsTextField txtSiglas = new clsTextField(90,54,80,20); frmProfesion(){ super("Mantenimiento Profesión",420,170);Componentes(); SQLconsulta = "Select codigo,descripcion From Cpm_Profesion Where estado = true"; Reporte = "profesion";HabilitarEntrada();HabilitarBotones();} private void Componentes(){ lblDescripcion.setText("Nombre:");getContentPane().add(lblSiglas); txtSiglas.addKeyListener(new KeyAdapter(){public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();});getContentPane().add(txtSiglas); cntEstado.setLocation(cntEstado.getX(),cntEstado.getY()+22);} void LimpiarEntrada(){ super.LimpiarEntrada();txtSiglas.setText("");} void HabilitarEntrada(){ super.HabilitarEntrada();txtSiglas.Habilitar(getEditando());} void Grabar(){ super.Grabar(); if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre","Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();return;} datProfesion dat = new datProfesion();dat.setDescripcion(txtDescripcion.getText()); dat.setSiglas(txtSiglas.getText());if (getEsNuevo()) {if (dat.Grabar()) txtCodigo.setText(dat.getUltimoCodigo());} else {JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar",JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();} else {dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado());if (!dat.Actualizar()){ JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar",JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();}} void Consultar(String Codigo) {if (Codigo.equals(""))return; datProfesion dat = new datProfesion(); if (dat.Consultar(Codigo)){ txtCodigo.setText(" "+ dat.getCodigo());txtDescripcion.setText(" "+ dat.getDescripcion()); txtSiglas.setText(" "+ dat.getSiglas());setEstado(dat.getEstado());setRegistro(true);HabilitarBotones();} else {LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE);} void Eliminar(){ int res = JOptionPane.showConfirmDialog(null,"¿Está seguro de eliminar el registro?", "Mensaje",JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION){ datProfesion dat = new datProfesion(); dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar()) LimpiarEntrada();}}} </pre>			
Observaciones: Hereda de clsMantenimientoTipo1.java			

6.2.2.26 CLASE FRMTIPOINCENTIVO

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmTipoIncentivo.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar los Tipos de Incentivo existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class frmTipoIncentivo extends clsMantenimientoTipo2{ frmTipoIncentivo(){ super("Mantenimiento Tipo de Incentivos",420,170); SQLconsulta = "Select codigo,descripcion From CPm_TipoIncentivo Where estado = true"; Reporte = "tipoincentivo";HabilitarEntrada();HabilitarBotones(); } void Grabar(){super.Grabar(); if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique una descripción","Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();return;} if (txtMonto.getText().equals("")) txtMonto.setText("0"); datTipoIncentivo dat = new datTipoIncentivo();dat.setDescripcion(txtDescripcion.getText()); dat.setMonto(Double.parseDouble(txtMonto.getText())); if (getEsnuevo()) {if (dat.Grabar()) txtCodigo.setText(dat.getUltimoCodigo()); else {JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar",JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();}} else { dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado());if (!dat.Actualizar()){ JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar",JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();}} } void Consultar(String Codigo) { if (Codigo.equals(""))return; datTipoIncentivo dat = new datTipoIncentivo(); if (dat.Consultar(Codigo)){ txtCodigo.setText(" "+ dat.getCodigo());txtDescripcion.setText(" "+ dat.getDescripcion()); txtMonto.setText(" "+ dat.getMonto());setEstado(dat.getEstado()); setRegistro(true);HabilitarBotones(); } else {LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE);} void Eliminar(){int res = JOptionPane.showConfirmDialog(null,"¿Está seguro de eliminar el registro?", "Mensaje",JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION){ datTipoIncentivo dat = new datTipoIncentivo(); dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar()) LimpiarEntrada();}} } }</pre>			
Observaciones: Hereda de clsMantenimientoTipo2.java			



6.2.2.27 CLASE FRMTIPOJUSTIFICA

EAGLEEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Cientes	Nombre de la Clase: frmTipoJustifica.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar los tipos de justificación existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class frmTipoJustifica extends clsMantenimientoTipo1 { frmTipoJustifica(){ super("Mantenimiento Tipo de Justificativo",420,170); SQLconsulta = "Select codigo,descripcion From CPM_TipoJustificativo Where estado = true"; Reporte = "tipojustificativo";HabilitarEntrada();HabilitarBotones();} void Grabar(){ super.Grabar();if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un descripción", "Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();return;} datTipoJustifica dat = new datTipoJustifica();dat.setDescripcion(txtDescripcion.getText()); if (getEsNuevo()) {if (dat.Grabar())txtCodigo.setText(dat.getUltimoCodigo()); else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();}} else { dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado());if (!dat.Actualizar()) { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();}} } void Consultar(String Codigo) { if (Codigo.equals(""))return; datTipoJustifica dat = new datTipoJustifica();if (dat.Consultar(Codigo)){ txtCodigo.setText("" + dat.getCodigo());txtDescripcion.setText("" + dat.getDescripcion()); setEstado(dat.getEstado());setRegistro(true);HabilitarBotones();} else {LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE);} void Eliminar(){ int res = JOptionPane.showConfirmDialog(null,"¿Está seguro de eliminar el registro?", "Mensaje",JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION){ datTipoJustifica dat = new datTipoJustifica(); dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar()) LimpiarEntrada(); } } } }</pre>			
Observaciones: Hereda de clsMantenimientoTipo1.java			

6.2.2.28 CLASE FRMTIPOMULTA



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmTipoMulta.java	Tipo: Public	
Descripción: Clase encargada de ingresar modificar o eliminar los tipos de multa existentes.			
DESCRIPCIÓN DE CLASE			
<pre> class frmTipoMulta extends clsMantenimientoTipo2{ frmTipoMulta(){ super("Mantenimiento Tipo de Multas",420,170); SQLconsulta = "Select codigo,descripcion From CPm_TipoMulta Where estado = true"; Reporte = "tipomulta";HabilitarEntrada();HabilitarBotones(); void Grabar(){super.Grabar(); if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un descripción","Error", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();return;} if (txtMonto.getText().equals("")) txtMonto.setText("0"); datTipoMulta dat = new datTipoMulta();dat.setDescripcion(txtDescripcion.getText()); dat.setMonto(Double.parseDouble(txtMonto.getText())); if (getEsNuevo()) { if (dat.Grabar())txtCodigo.setText(dat.getUltimoCodigo()); } else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();} else { dat.setCodigo(txtCodigo.getText());dat.setEstado(getEstado());if (!dat.Actualizar()){ JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); setEditando(true);HabilitarEntrada();HabilitarBotones();}} void Consultar(String Codigo) { if (Codigo.equals(""))return; datTipoMulta dat = new datTipoMulta(); if (dat.Consultar(Codigo)){ txtCodigo.setText(" "+ dat.getCodigo());txtDescripcion.setText(" "+ dat.getDescripcion()); txtMonto.setText(" "+ dat.getMonto());setEstado(dat.getEstado()); setRegistro(true);HabilitarBotones(); } else{LimpiarEntrada(); JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE); } void Eliminar(){ int res = JOptionPane.showConfirmDialog(null,"¿Está seguro de eliminar el registro?", "Mensaje",JOptionPane.YES_NO_OPTION); if (res == JOptionPane.YES_OPTION){ datTipoMulta dat = new datTipoMulta(); dat.setCodigo(txtCodigo.getText()); if (dat.Eliminar()) LimpiarEntrada(); } } } } } </pre>			
Observaciones: Hereda de clsMantenimientoTipo2.java			

6.2.2.29 CLASE FRMTURNOSHORARIOS

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmTurnosHorarios.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar los horarios y turnos del personal existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class frmTurnosHorarios extends JInternalFrame{ private clsLabel lblCodigo = new clsLabel("Código:",10,10,80,20); private clsLabel lblDescripcion = new clsLabel("Descripción:",10,30,80,20); private clsEstado cntEstado = new clsEstado(10,70); private clsTextField txtCodigo = new clsTextField(90,10,50,20); private clsTextField txtDescripcion= new clsTextField(90,30,250,20); JPanel pnlBotones = new JPanel(null,true); private clsBoton btnNuevo = new clsBoton("Nuevo","N","Nuevo registro","new",0,0); private clsBoton btnGrabar = new clsBoton("Grabar","G","Guardar registro","save",0,0); private clsBoton btnModificar = new clsBoton("Modificar","M","Modificar registro","edit24",0,0); private clsBoton btnCancelar = new clsBoton("Cancelar","C","Cancelar edición","undo",0,0); private clsBoton btnCerrar = new clsBoton("Cerrar","C","Cerrar formulario","close",0,0); frmTurnosHorarios(){ Entorno();Componentes();HabilitarEntrada(false);show(); } private void Entorno(){ setTitle("Mantenimiento Turnos & Horarios");setBounds(10,10,350,200); setResizable(false);setMaximizable(false);setClosable(true); getContentPane().setLayout(null);setFrameIcon(new ImageIcon("imagenes/icono06.gif")); } private void Componentes(){ getContentPane().add(lblCodigo);getContentPane().add(lblDescripcion); getContentPane().add(cntEstado);getContentPane().add(txtCodigo);getContentPane().add(txtDescripcion); getContentPane().add(btnNuevo);btnNuevo.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Nuevo();}); getContentPane().add(btnGrabar);btnGrabar.setVisible(false);btnGrabar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Grabar();}); getContentPane().add(btnModificar);btnModificar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Modificar();}); getContentPane().add(btnCancelar);btnCancelar.setVisible(false);btnCancelar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Cancelar();}); getContentPane().add(btnCerrar); btnCerrar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ dispose(); setVisible(false); } }); UbicarBotones(10,getHeight()-btnNuevo.getHeight()-40); } }</pre>			
Observaciones:			

6.2.2.30 CLASE FRMTURNOSHORARIOS

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Clientes	Nombre de la Clase: frmTurnosHorarios.java	Tipo: Public	
Descripción: <i>Clase encargada de ingresar modificar o eliminar los horarios y turnos del personal existentes.</i>			
DESCRIPCIÓN DE CLASE			
<pre>private void UbicarBotones(int x,int y){ btnNuevo.setLocation(x,y);btnGrabar.setLocation(x,y);btnModificar.setLocation(btnNuevo.getX() + btnNuevo.getWidth(),y);btnCancelar.setLocation(btnModificar.getX(),btnModificar.getY());btnCerrar.setLocation(btnModificar.getX() + btnModificar.getWidth(),y);} private void VerBotones(boolean !Editando){btnGrabar.setVisible(!Editando);btnCancelar.setVisible(!Editando);btnNuevo.setVisible(!Editando);btnModificar.setVisible(! Editando);btnCerrar.setEnabled(!Editando);} private void LimpiarEntrada(){ txtCodigo.setText("");txtDescripcion.setText("");cntEstado.setEstado(true);} private void HabilitarEntrada(boolean !Estado){ txtCodigo.Habilitar(!Estado);txtDescripcion.Habilitar(!Estado);cntEstado.Habilitar(!Estado);} private void Nuevo(){ VerBotones(true);LimpiarEntrada();HabilitarEntrada(true);cntEstado.Habilitar(false);} private void Modificar(){ VerBotones(true);HabilitarEntrada(true);} private void Cancelar(){ VerBotones(false);HabilitarEntrada(false);LimpiarEntrada();} private void Grabar(){ if (txtDescripcion.getText().equals("")) { JOptionPane.showMessageDialog(null,"Especifique un nombre","Error", JOptionPane.ERROR_MESSAGE);return;} datTurnosHorarios dat = new datTurnosHorarios();dat.setDescripcion(txtDescripcion.getText()); if (cntEstado.getHabilitar()) { dat.setCodigo(Integer.parseInt(txtCodigo.getText()));dat.setEstado(cntEstado.getEstado()); if (dat.Actualizar()) { HabilitarEntrada(false);VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE);} else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); } else {if (dat.Grabar()) {JOptionPane.showMessageDialog(null,"turno 3");txtCodigo.setText(dat.getUltimoCodigo()); HabilitarEntrada(false);VerBotones(false); JOptionPane.showMessageDialog(null,"Registro Guardado Con Éxito","Grabar", JOptionPane.INFORMATION_MESSAGE);} else { JOptionPane.showMessageDialog(null,"Intento de Grabación Fallido","Grabar", JOptionPane.ERROR_MESSAGE); } } private void Consultar(String lcCodigo) {if (lcCodigo.equals("")){return; } datTurnosHorarios dat = new datTurnosHorarios(); if (dat.Consultar(lcCodigo)){ txtCodigo.setText(txtCodigo.getText() + dat.getCodigo());txtDescripcion.setText(dat.getDescripcion());cntEstado.setEstado(dat.getEstado()); else { JOptionPane.showMessageDialog(null,"Codigo no existe.", "Mensaje", JOptionPane.INFORMATION_MESSAGE); } } } }</pre>			
Observaciones:			



6.2.3. CAPA SERVIDOR.

6.2.3.1 CLASE clsCONEXION

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: clsConexion.java	Tipo: Public	
Descripción: <i>Clase encargada de realizar la conexión con la BD mediante el ODBC destinado.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class clsConexion{ Connection cnx; clsConexion(){ try { Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); } catch(ClassNotFoundException e){ JOptionPane.showMessageDialog(null,"No se pudo cargar el controlador JDBC-ODBC"); } try { cnx=DriverManager.getConnection("jdbc:odbc:Proyecto","",""); } catch(Exception e){ JOptionPane.showMessageDialog(null,"No se pudo cargar el ODBC Proyecto"); } } boolean Ejecutar(String lcSQL){ Statement cSQL; try{ cSQL=cnx.createStatement(); cSQL.executeUpdate(lcSQL); cnx.close(); return true; } catch(SQLException e){ return false; } } String getUltimold(String lcSQL){ Statement cSQL; ResultSet rs; String id; id = "0"; try{ cSQL = cnx.createStatement(); rs = cSQL.executeQuery(lcSQL); if (rs.next()){ id = String.valueOf(rs.getInt(1)); } cnx.close(); return id; } catch(SQLException e){ return "0"; } } }</pre>			
Observaciones:			

6.2.3.2 CLASE cnxPERSONAL

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1
		Fecha Diseño: 2/08/2004
		Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo
Capa: Servidor	Nombre de la Clase: cnxPersonal.java	Tipo: Public
Descripción: Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla personal.		
DESCRIPCIÓN DE CLASE		
<pre> class cnxPersonal extends clsConexion{ cnxPersonal(){ boolean Grabar(String Nombre,String Apellido,String NombreCorto,String Clave, String Direccion,String Cedula,int Cargo,int Dpto,String Telefono, String Email,String FEntrada,String FNacimiento,int LNacimiento, int EstadoCivil,int Profesion,String Foto){ return Ejecutar("Insert Into CPm_Persona(nombre,apellido,nombrecorto,clave,direccion" + ",cedula,cargo,dpto,telefono,email,fechaentrada,fechanacimiento" + ",lugarnacimiento,estadocivil,profesion,foto)" + " Values(" + Nombre + "," + Apellido + "," + NombreCorto + "," + Clave + "," + Direccion + "," + Cedula + "," + Cargo + "," + Dpto + "," + Telefono + "," + Email + "," + FEntrada + "," + FNacimiento + "," + LNacimiento + "," + EstadoCivil + "," + Profesion + "," + Foto + ")"); } boolean Actualizar(String Codigo,String Nombre,String Apellido,String NombreCorto, String Clave,String Direccion,String Cedula,int Cargo,int Dpto,String Telefono, String Email,String FEntrada,String FNacimiento,int LNacimiento,int EstadoCivil, int Profesion,String Foto,boolean Estado){ return Ejecutar("Update CPm_Persona Set " + " nombre=" + Nombre + ",apellido=" + Apellido + ",nombrecorto=" + NombreCorto + ",clave=" + Clave + ",direccion=" + Direccion + ",cedula=" + Cedula + ",cargo=" + Cargo + ",dpto=" + Dpto + ",telefono=" + Telefono + ",email=" + Email + ",fechaentrada=" + FEntrada + ",fechanacimiento=" + FNacimiento + ",lugarnacimiento=" + LNacimiento + ",estadocivil=" + EstadoCivil + ",profesion=" + Profesion + ",foto=" + Foto + ",estado=" + Estado + " Where codigo=" + Codigo); } boolean Anular(String Codigo){ return Ejecutar("Update CPm_Persona Set estado=false Where codigo=" + Codigo); } boolean Eliminar(String Codigo){ return Ejecutar("Delete * From CPm_Persona Where Codigo=" + Codigo); } String getUltimoCodigo(){ return this.getUltimoId("Select Max(codigo) From CPm_Persona"); } datPersonal Consultar(String Codigo){ Statement cSQL; ResultSet rs; datPersonal dat = new datPersonal(); } </pre>		
Observaciones: Hereda de clsConexion.java		

6.2.3.3 CLASE cnxPERSONAL

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: cnxPersonal.java	Tipo: Public	
Descripción: <i>Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla personal.</i>			
DESCRIPCIÓN DE CLASE			
<pre>try{ cSQL = cnx.createStatement(); rs = cSQL.executeQuery("Select * From CPm_Persona Where Codigo =" + Codigo); if (rs.next()){ dat.setCodigo(rs.getString("Codigo")); dat.setNombre(rs.getString("Nombre")); dat.setApellido(rs.getString("Apellido")); dat.setNombreCorto(rs.getString("NombreCorto")); dat.setClave(rs.getString("Clave")); dat.setDireccion(rs.getString("Direccion")); dat.setCedula(rs.getString("Cedula")); dat.setCargo(rs.getInt("Cargo")); dat.setDpto(rs.getInt("Dpto")); dat.setTelefono(rs.getString("Telefono")); dat.setEmail(rs.getString("Email")); dat.setFEntrada(rs.getString("FechaEntrada")); dat.setFSalida(rs.getString("FechaSalida")); dat.setFNacimiento(rs.getString("FechaNacimiento")); dat.setLNacimiento(rs.getInt("LugarNacimiento")); dat.setEstadoCivil(rs.getInt("EstadoCivil")); dat.setProfesion(rs.getInt("Profesion")); dat.setFoto(rs.getString("Foto")); dat.setEstado(rs.getBoolean("Estado")); cnx.close(); } else { dat.setCodigo("-1"); } return dat; } catch(SQLException e){ dat.setCodigo("-1"); return dat; } }</pre>			
Observaciones: Hereda de clsConexion.java			



6.2.3.4 CLASE cnxEMPRESA

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: cnxEmpresa.java	Tipo: Public	
Descripción: Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla Empresa.			
DESCRIPCIÓN DE CLASE			
<pre>class cnxEmpresa extends clsConexion{ cnxEmpresa(){ boolean Grabar(String Nombre,String Ruc,String Direccion,String Telefono,String Fax,String Email,String Web){ return Ejecutar("Insert Into CPm_Empresa(nombre,ruc,direccion,telefono,fax,email,web)" + " Values('"+ Nombre + "','" + Ruc + "','" + Direccion + "','" + Telefono + "','" + Fax + "','" + Email + "','" + Web + "')");} boolean Actualizar(String Codigo,String Ruc,String Nombre,String Direccion,String Telefono,String Fax,String Email,String Web,boolean Estado){ return Ejecutar("Update CPm_Empresa Set" + " ruc='"+ Ruc + "',nombre='"+ Nombre + "',direccion='"+ Direccion + "',telefono='"+ Telefono + "',fax='"+ Fax + "',email='"+ Email + "',Web='"+ Web + " Where codigo=" + Codigo);} boolean Anular(String Codigo){ return Ejecutar("Update CPm_Empresa Set estado=false Where codigo=" + Codigo); } boolean Eliminar(String Codigo){ return Ejecutar("Delete * From CPm_Empresa Where Codigo=" + Codigo); } String getUltimoCodigo(){ return this.getUltimoId("Select Max(codigo) From CPm_Empresa"); } datEmpresa Consultar(String Codigo){ Statement cSQL; ResultSet rs; datEmpresa dat = new datEmpresa(); try{ cSQL = cnx.createStatement(); rs = cSQL.executeQuery("Select * From CPm_Empresa Where Codigo =" + Codigo); if (rs.next()){ dat.setCodigo(rs.getString("Codigo")); dat.setRuc(rs.getString("Ruc")); dat.setNombre(rs.getString("Nombre")); dat.setDireccion(rs.getString("Direccion")); dat.setTelefono(rs.getString("Telefono")); dat.setFax(rs.getString("Fax")); dat.setEmail(rs.getString("Email")); dat.setWeb(rs.getString("Web")); cnx.close(); } else { dat.setCodigo("-1"); } } return dat; } catch(SQLException e){ dat.setCodigo("-1"); return dat; } }</pre>			
Observaciones: Hereda de clsConexion.java			

6.2.3.5 CLASE cnxUSUARIO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: cnxUsuario.java	Tipo: Public	
Descripción: Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla usuario.			
DESCRIPCIÓN DE CLASE			
<pre>class cnxUsuario extends clsConexion{ cnxUsuario(){ } boolean Grabar(String Usuario,String Clave,int Nivel){ return Ejecutar("Insert Into Cpm_Cargo(usuario,clave,nivel)" + " Values('" + Usuario + "','" + Clave + "','" + Nivel + "')"); } boolean Actualizar(String Id,String Usuario,String Clave,int Nivel,boolean Estado){ return Ejecutar("Update Cpm_Usuario Set" + " usuario='" + Usuario + "','" + Clave + "','" + Nivel + "',estado='" + Estado + " Where id=" + Id); } boolean Anular(String Id){ return Ejecutar("Update Cpm_Usuario Set estado=false Where id=" + Id); } boolean Eliminar(String Id){ return Ejecutar("Delete * From Cpm_Usuario Where id=" + Id); } String getUltimoCodigo(){ return this.getUltimoId("Select Max(id) From Cpm_Usuario"); } } datUsuario Consultar(String Id){ Statement cSQL; ResultSet rs; datUsuario dat = new datUsuario(); try{ cSQL = cnx.createStatement(); rs = cSQL.executeQuery("Select * From Cpm_Usuario Where estado = true and id=" + Id); if (rs.next()){ dat.setId(rs.getString(1)); dat.setUsuario(rs.getString("Usuario")); dat.setClave(rs.getString("Clave")); dat.setNivel(rs.getInt("Nivel")); dat.setEstado(rs.getBoolean("Estado")); cnx.close(); } else dat.setId("-1"); return dat; } catch(SQLException e){ dat.setId("-1"); dat.setNivel(0); return dat; } } }</pre>			
Observaciones: Hereda de clsConexion.java			

6.2.3.6 CLASE cnxDEPARTAMENTO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: cnxDepartamento.java	Tipo: Public	
Descripción: Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla departamento.			
DESCRIPCIÓN DE CLASE			
<pre>class cnxDepartamento extends clsConexion{ cnxDepartamento(){ } boolean Grabar(String Descripcion){ return Ejecutar("Insert Into CPm_Departamento(descripcion)" + " Values('" + Descripcion + "')"); } boolean Actualizar(String Codigo,String Descripcion,boolean Estado){ return Ejecutar("Update CPm_Departamento Set" + " descripcion='"+ Descripcion + "',estado='"+ Estado + " Where codigo=" + Codigo); } boolean Anular(String Codigo){ return Ejecutar("Update CPm_Departamento Set estado=false Where codigo=" + Codigo); } boolean Eliminar(String Codigo){ return Ejecutar("Delete * From CPm_Departamento Where Codigo=" + Codigo); } String getUltimoCodigo(){ return this.getUltimoId("Select Max(codigo) From CPm_Departamento"); } datDepartamento Consultar(String Codigo){ Statement cSQL; ResultSet rs; datDepartamento dat = new datDepartamento(); try{ cSQL = cnx.createStatement(); rs = cSQL.executeQuery("Select * From CPm_Departamento Where codigo=" + Codigo); if (rs.next()){ dat.setCodigo(rs.getString(1)); dat.setDescripcion(rs.getString("Descripcion")); dat.setEstado(rs.getBoolean("Estado")); cnx.close(); } else { dat.setCodigo("-1"); } return dat; } catch(SQLException e){ dat.setCodigo("-1"); return dat; } } }</pre>			
Observaciones: Hereda de clsConexion.java			

6.2.3.7 CLASE cnxCiudad

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: cnxCiudad.java	Tipo: Public	
Descripción: <i>Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla ciudad.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class cnxCiudad extends clsConexion{ cnxCiudad(){ } boolean Grabar(String Nombre){ return Ejecutar("Insert Into CPm_Ciudad(nombre)" + " Values(" + Nombre + ")"); } boolean Actualizar(String Codigo,String Nombre,boolean Estado){ return Ejecutar("Update CPm_Ciudad Set " + " nombre="+ Nombre + ",estado="+ Estado + " Where codigo=" + Codigo); } boolean Anular(String Codigo){ return Ejecutar("Update CPm_Ciudad Set estado=false Where codigo=" + Codigo); } boolean Eliminar(String Codigo){ return Ejecutar("Delete * From CPm_Ciudad Where Codigo=" + Codigo); } String getUltimoCodigo(){ return this.getUltimold("Select Max(codigo) From CPm_Ciudad"); } datCiudad Consultar(String Codigo){ Statement cSQL; ResultSet rs; datCiudad dat = new datCiudad(); try{ cSQL = cnx.createStatement(); rs = cSQL.executeQuery("Select * From CPm_Ciudad Where codigo=" + Codigo); if (rs.next()){ dat.setCodigo(rs.getString(1)); dat.setNombre(rs.getString("Nombre")); dat.setEstado(rs.getBoolean("Estado")); cnx.close(); } else { dat.setCodigo("-1"); } return dat;} catch(SQLException e){ dat.setCodigo("-1"); return dat;}}}</pre>			
Observaciones: Hereda de clsConexion.java			

6.2.3.8 CLASE cnxAsistencia

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: cnxPersonal.java	Tipo: Public	
Descripción: <i>Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla asistencia.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class cnxAsistencia extends clsConexion{ cnxAsistencia(){ } boolean Grabar(int Persona,String Carnet){ return Ejecutar("Insert Into CPT_Asistencia(persona,carnet)" + " Values(" + Persona + "," + Carnet + ")"); } String getUltimoNumero(){ return this.getUltimoid("Select Max(numero) From CPT_Asistencia"); } String getUltimaMarca(){ return this.getUltimoid("Select Max(marca) From CPT_Asistencia"); } datAsistencia Consultar(String Numero){ Statement cSQL; ResultSet rs; datAsistencia dat = new datAsistencia(); try{ cSQL = cnx.createStatement(); rs = cSQL.executeQuery("Select * From CPT_Asistencia Where numero=" + Numero); if (rs.next()){ dat.setNumero(rs.getString("Numero")); dat.setFecha(rs.getDate("Fecha")); dat.setPersona(rs.getInt("Persona")); dat.setCarnet(rs.getString("Carnet")); dat.setMarca(rs.getDate("Marca")); dat.setEstado(rs.getBoolean("Estado")); cnx.close(); } else { dat.setNumero("-1"); } } return dat; } catch(SQLException e){ dat.setNumero("-1"); return dat; } }</pre>			
Observaciones: Hereda de clsConexion.java			

6.2.3.9 CLASE cnxTurnosHorarios

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Capa: Servidor	Nombre de la Clase: cnxTurnosHorarios.java	Tipo: Public	
Descripción: <i>Clase encargada mediante el ODBC de modificar, ingresar o Eliminar datos físicamente en la base de datos en la tabla TurnosHorarios.</i>			
DESCRIPCIÓN DE CLASE			
<pre>class cnxTurnosHorarios extends clsConexion{ cnxTurnosHorarios(){ } boolean Grabar(String Descripcion){ return Ejecutar("Insert Into CPm_Cargo(descripcion)" + " Values('"+ Descripcion + "')"); } boolean Actualizar(String Codigo,String Descripcion,boolean Estado){ return Ejecutar("Update CPm_Cargo Set" + " descripcion='"+ Descripcion + ",estado='"+ Estado + " Where codigo=" + Codigo); } boolean Anular(String Codigo){ return Ejecutar("Update CPm_Cargo Set estado=false Where codigo=" + Codigo); } boolean Eliminar(String Codigo){ return Ejecutar("Delete * From CPm_Cargo Where Codigo=" + Codigo); } String getUltimoCodigo(){ return this.getUltimold("Select Max(codigo) From CPm_Cargo"); } datTurnosHorarios Consultar(String Codigo){ Statement cSQL; ResultSet rs; datTurnosHorarios dat = new datTurnosHorarios(); return dat; } }</pre>			
Observaciones: Hereda de clsConexion.java			

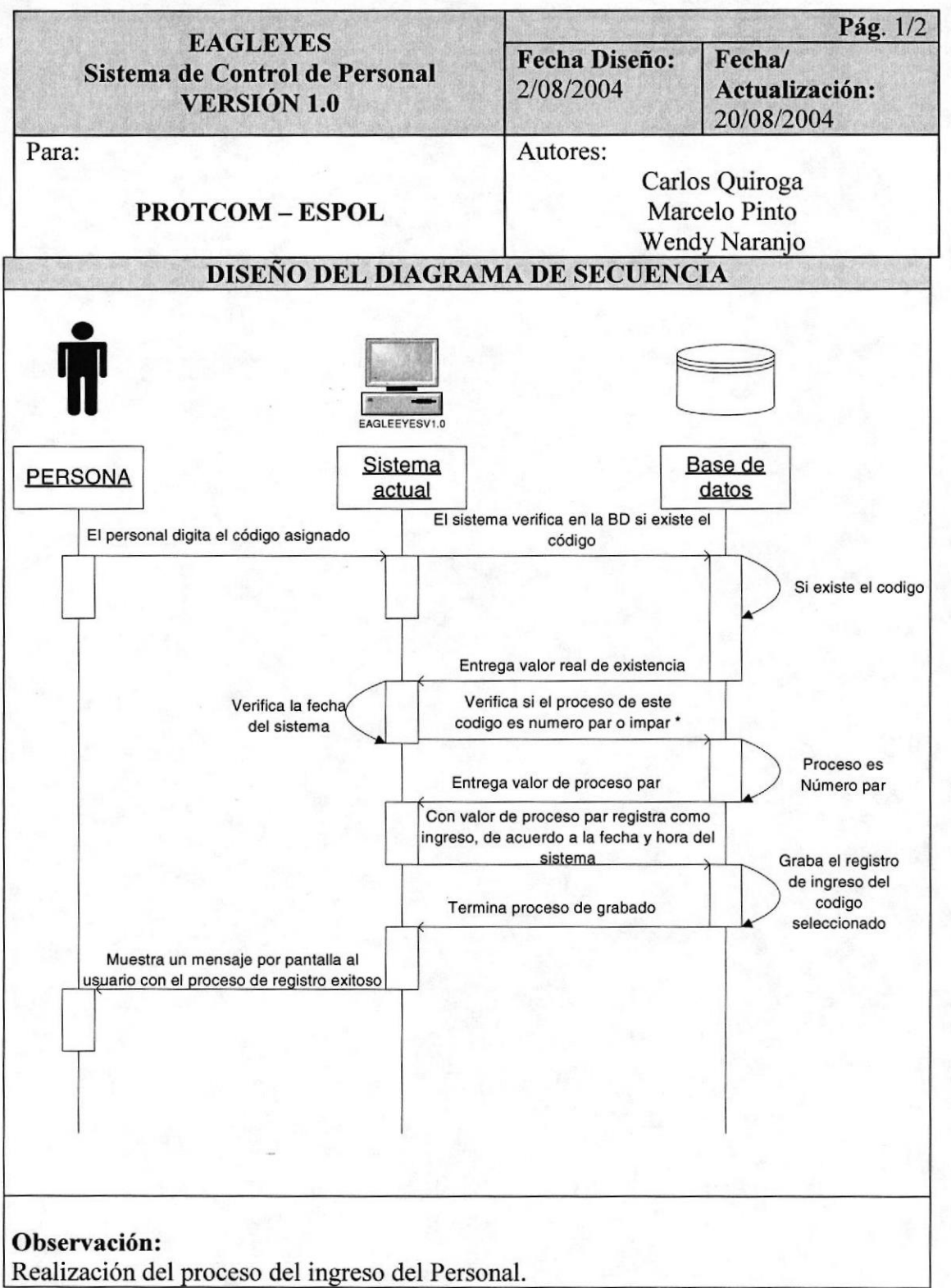


CAPÍTULO 7

DIAGRAMAS DE SECUENCIAS

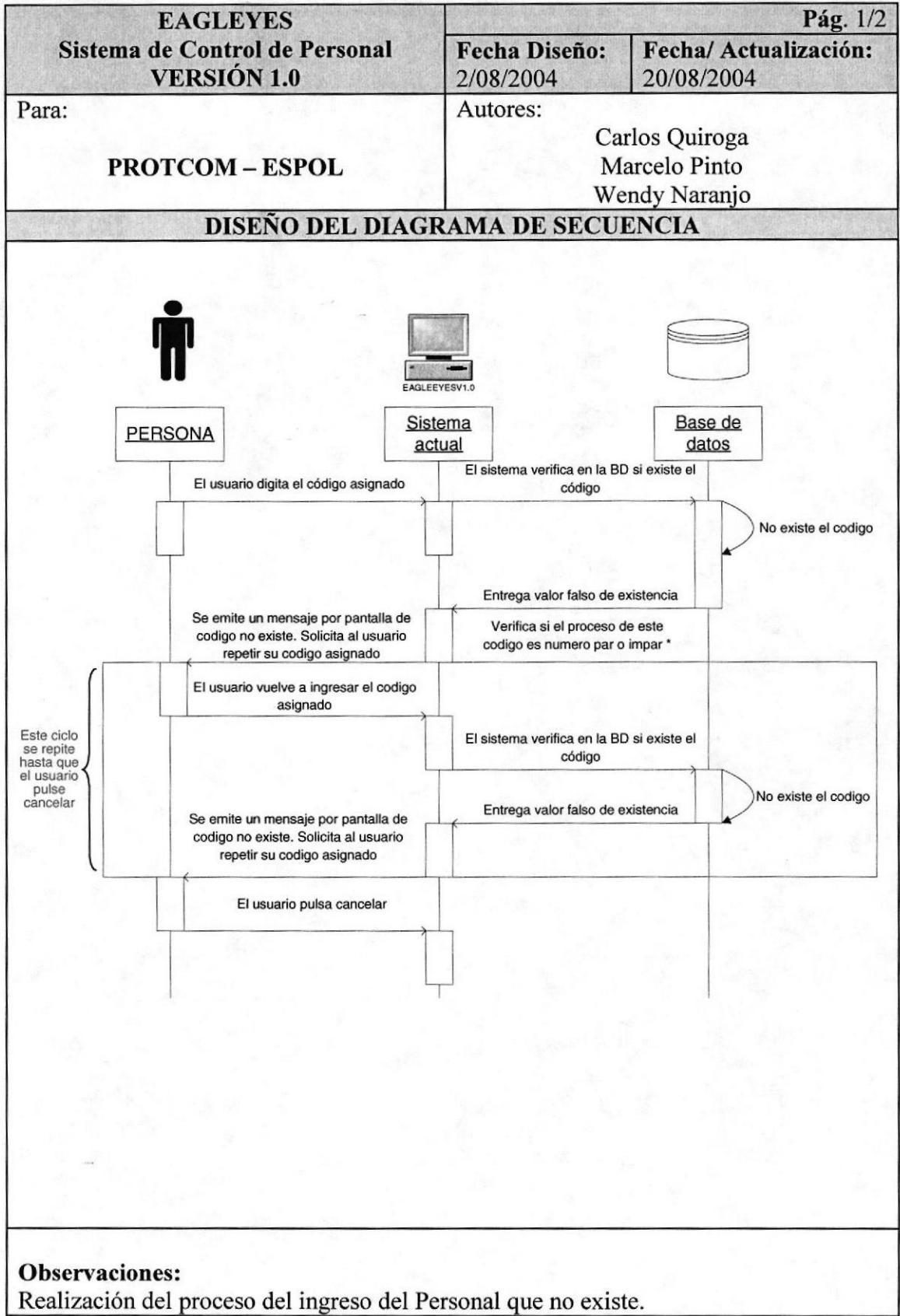
7. DIAGRAMAS DE SECUENCIAS

7.1. PROCESO DE INGRESO DEL PERSONAL



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2	
		Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Narrativa de Diagrama de Secuencia de Ingreso del Personal.</i>			
NARRATIVA DE DIAGRAMA DE SECUENCIA DE INGRESO DEL PERSONAL			
1. -	La persona ingresa al lugar de Trabajo y digita su código o clave asignado al Sistema de Control de Personal EAGLEYES .		
2. -	El Sistema Control de Personal EAGLEYES verifica en la Base de Datos que exista dicho código.		
3.-	Si el código existe entonces entrega un valor real de existencia y verifica la fecha del sistema.		
4.-	El Sistema verifica si el proceso de este código es número par o impar. Si el proceso es par entonces registra el proceso de ingreso de la persona de acuerdo con la fecha y la hora del sistema.		
5.-	Entonces graba el registro de ingreso del código seleccionado.		
6.-	Termina el proceso de grabado.		
7.-	El Sistema Control de Personal EAGLEYES muestra un mensaje por pantalla al usuario con el proceso de registro exitoso.		
Observaciones:			

7.2. PROCESO DE INGRESO DE PERSONAL CON CÓDIGO NO EXISTENTE

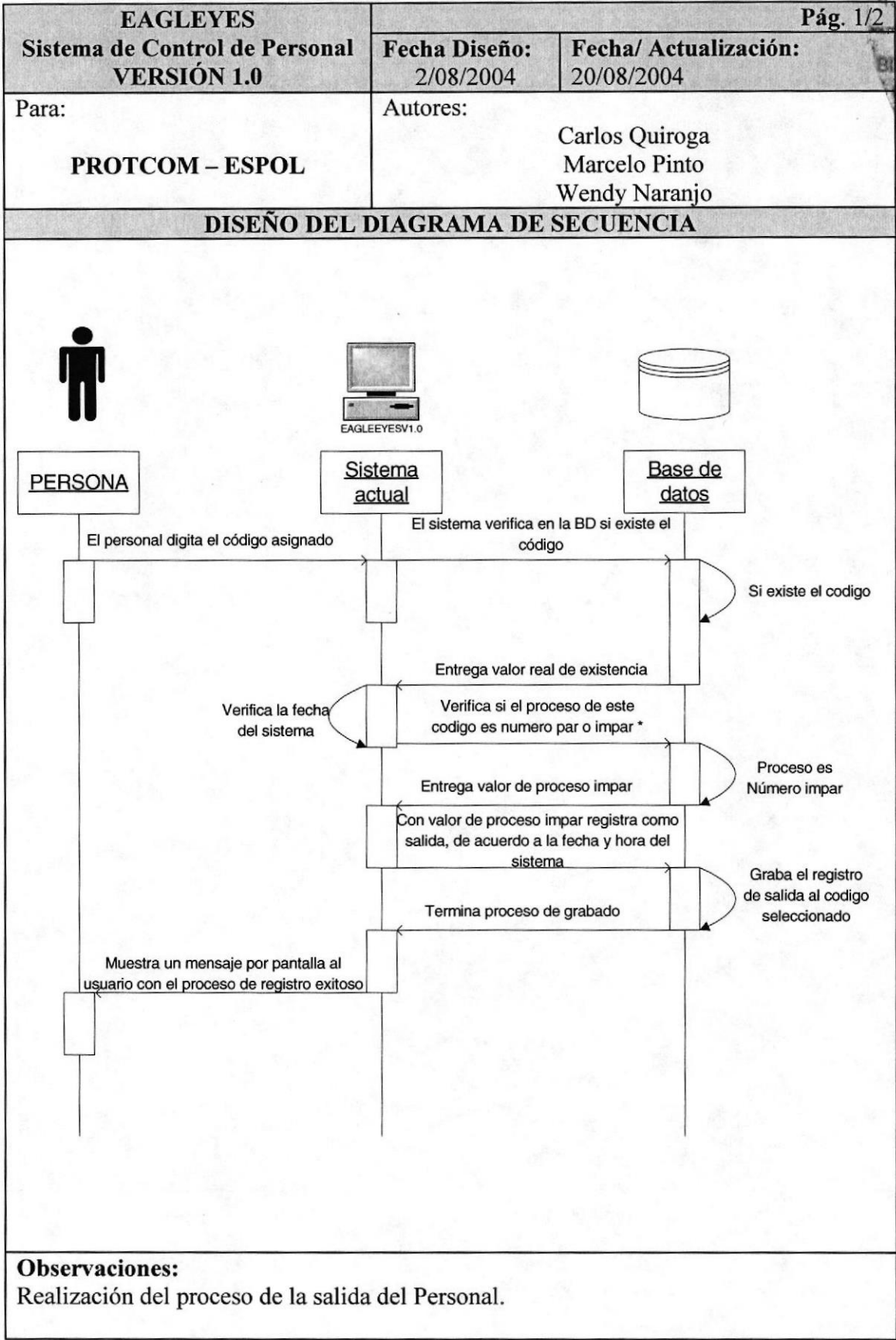




EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2	
		Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Narrativa de Diagrama de Secuencia de Ingreso de Personal con Código no existente.</i>			
NARRATIVA DE DIAGRAMA DE SECUENCIA DE INGRESO DE PERSONAL CON CÓDIGO NO EXISTENTE			
1. -	La Usuario digita su código o clave asignado al Sistema de Control de Personal EAGLEYES .		
2. -	El Sistema Control de Personal EAGLEYES verifica en la Base de Datos que exista dicho código.		
3.-	El Sistema verifica si el proceso de este código es número par o impar. Si el código NO existe entonces entrega un valor falso de existencia.		
4.-	Por pantalla se emite un mensaje de que el código NO existe. Y se solicita al usuario a que vuelva a repetir el código asignado.		
5.-	El Sistema Control de Personal EAGLEYES verifica en la Base de Datos que exista dicho código. Si el código NO existe entonces entrega un valor falso de existencia.		
5.-	Por pantalla se emite un mensaje de que el código NO existe. Y se solicita al usuario a que vuelva a repetir el código asignado.		
6.-	Y este ciclo se repetirá sucesivamente hasta que el usuario pulse cancelar.		
6.-	Una vez que digite cancelar termina el proceso.		
Observaciones:			

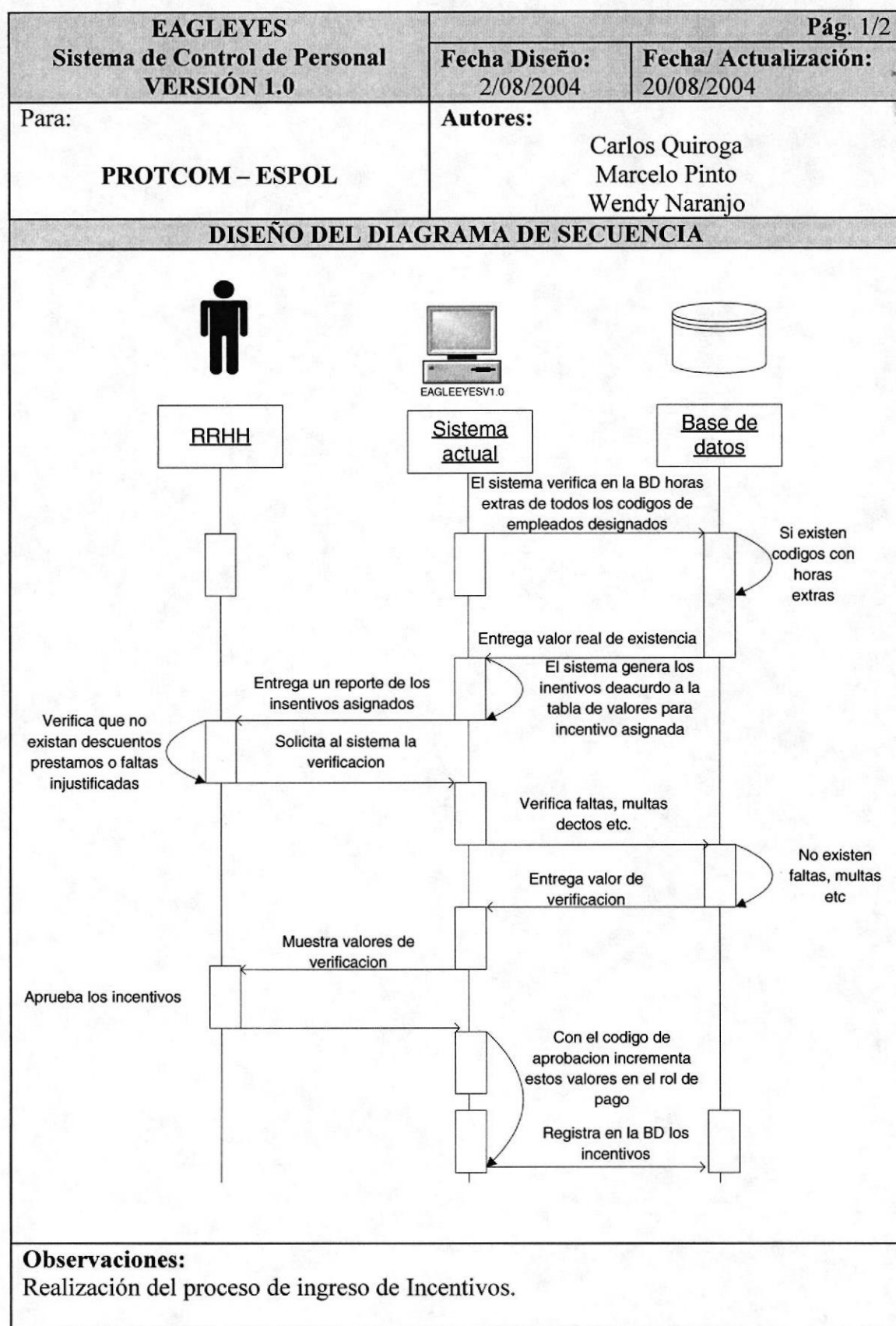


7.3. PROCESO DE SALIDA DEL PERSONAL



EAGLEYES		Pág. 2/2	
Sistema de Control de Personal		Fecha Diseño:	Fecha/ Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para:		Autores:	
PROTCOM – ESPOL		Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Narrativa de Diagrama de Secuencia de Salida del Personal.</i>			
NARRATIVA DE DIAGRAMA DE SECUENCIA DE SALIDA DEL PERSONAL			
1. -	El personal digita su código o clave asignado al Sistema de Control de Personal EAGLEYES .		
2. -	El Sistema Control de Personal EAGLEYES verifica en la Base de Datos que exista dicho código.		
3.-	Si el código existe entonces entrega un valor real de existencia y verifica la fecha del sistema.		
4.-	El Sistema verifica si el proceso de este código es número par o impar. Si el proceso es impar entonces entrega valor de proceso impar. Con el valor impar registra el proceso de salida de acuerdo con la fecha y la hora del sistema.		
5.-	Entonces graba el registro de salida del código seleccionado.		
6.-	Termina el proceso de grabado.		
7.-	El Sistema Control de Personal EAGLEYES muestra un mensaje por pantalla al usuario con el proceso de registro exitoso.		
Observaciones:			

7.4. PROCESO DE INGRESO DE INCENTIVO



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2 Fecha Diseño: 2/08/2004		Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Descripción: Narrativa de Diagrama de Secuencia de Ingreso del Incentivos.				
NARRATIVA DE DIAGRAMA DE SECUENCIA DE INGRESO DEL INCENTIVOS				
1. -	El Sistema Control de Personal EAGLEYES verifica en la Base de Datos las Horas Extras de todos los códigos de empleados designados.			
2. -	Si existen códigos con Horas Extras entonces se entrega un valor real de existencia y el sistema genera los incentivos de acuerdo a la tabla de valores para incentivos asignada.			
3.-	El Sistema Control de Personal EAGLEYES entrega un reporte de los incentivos asignados y solicita la verificación.			
4.-	Entonces el Sistema verifica que no existan descuentos, préstamos o faltas injustificadas, el sistema solicita dicha verificación.			
5.-	El Sistema entrega el valor de verificación. Si no existen ningún tipo de descuentos entonces entrega el valor de verificación.			
6.-	Entonces muestra los valores de verificación aprueba los incentivos.			
7.-	El Sistema Control de Personal EAGLEYES con el código de aprobación incrementa estos valores en el rol de pago de los empleados. Y registra en la base de datos dichos incentivos.			
Observaciones:				

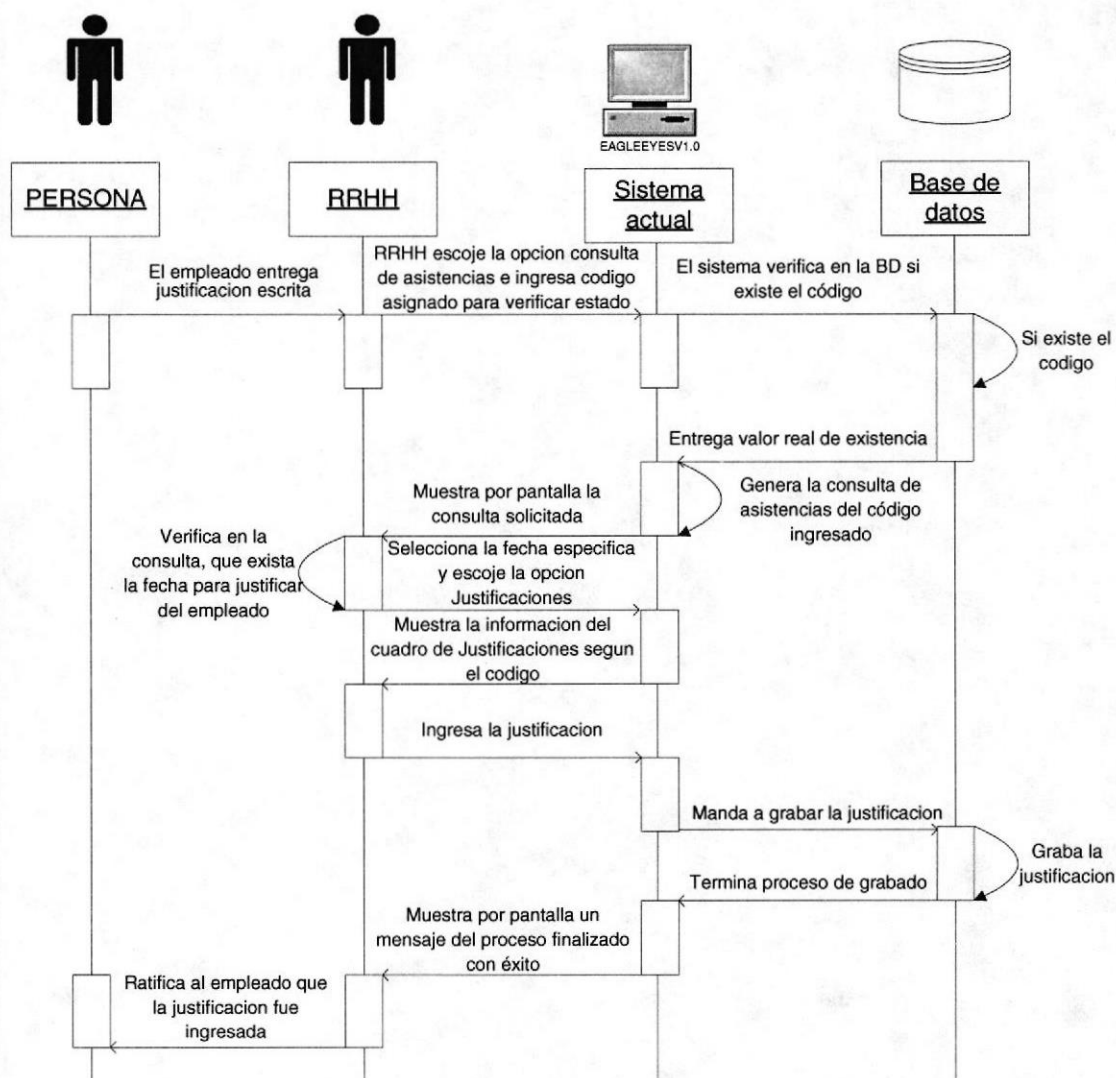


7.5. PROCESO DE INGRESO DE JUSTIFICATIVOS

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/2	
	Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	



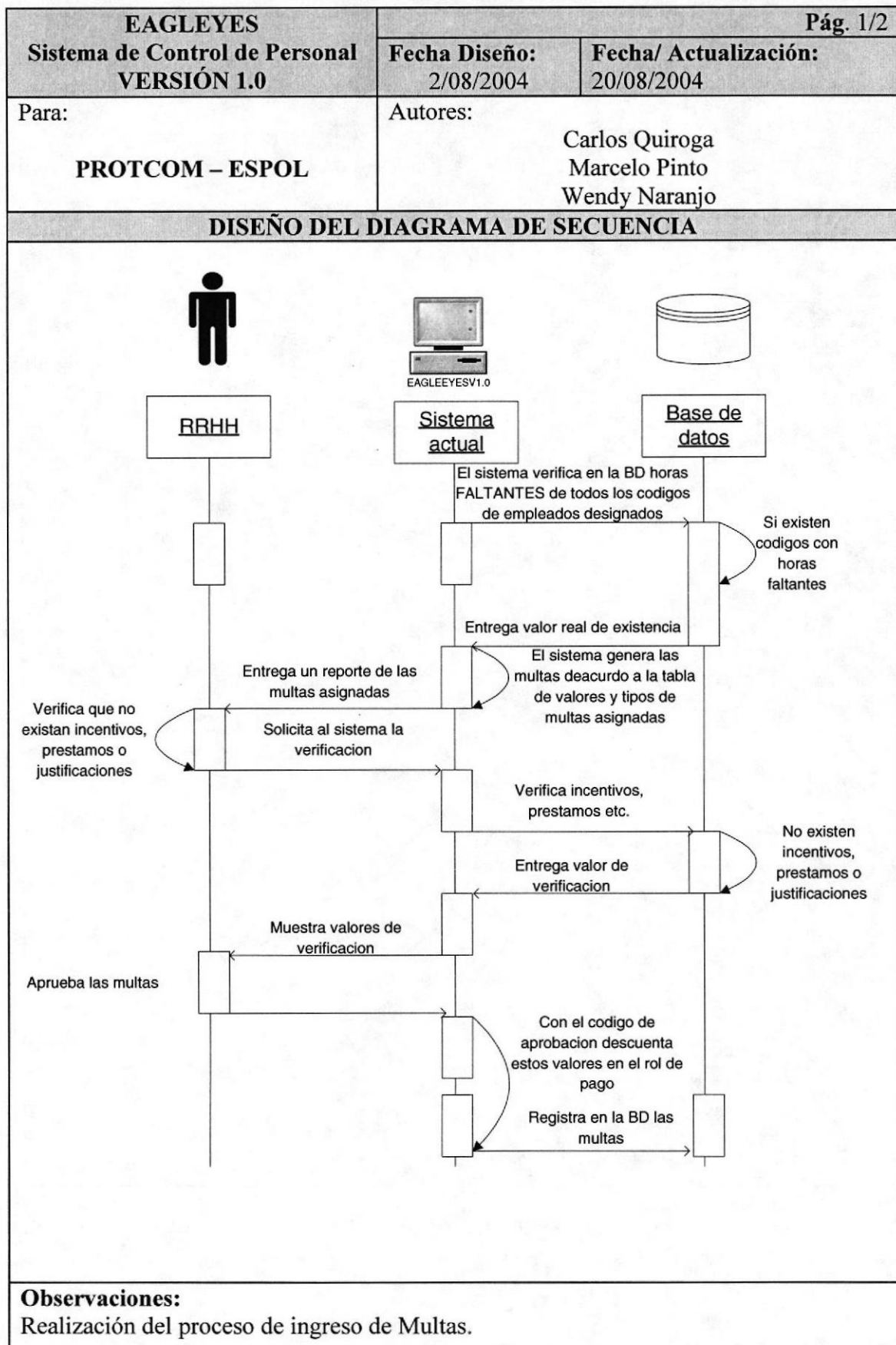
DISEÑO DEL DIAGRAMA DE SECUENCIA

**Observaciones:**

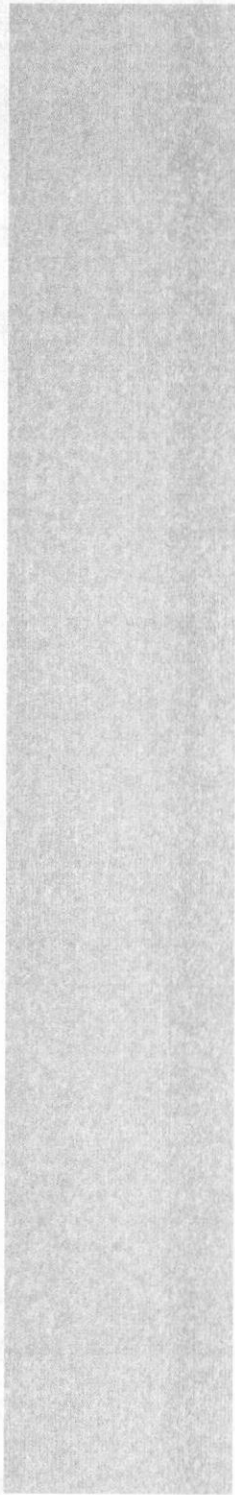
Realización del proceso de ingreso de Justificativos.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2 Fecha Diseño: 2/08/2004		Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Descripción: <i>Narrativa de Diagrama de Secuencia de Ingreso de Justificativos.</i>				
NARRATIVA DE DIAGRAMA DE SECUENCIA DE INGRESO DE JUSTIFICATIVOS				
1. -	El empleado entrega la justificación escrita. El Departamento de RRHH escoge la opción de consulta de asistencias e ingresa el código asignando para verificar el estado.			
2. -	El Sistema Control de Personal EAGLEYES verifica en la Base de Datos si existe el código.			
3.-	Si existe el código el Sistema entrega el valor real de existencia y procede a generar la consulta de asistencias del código ingresado.			
4.-	Entonces el Sistema muestra por pantalla la consulta solicitada. Se selecciona la fecha específica y escoge la opción justificaciones. Verifica en la consulta, que exista la fecha para justificar del empleado.			
5.-	El sistema muestra la información del cuadro de Justificaciones según el código. Y procede con el ingreso de la justificación.			
6.-	Entonces envía a grabar la justificación. Graba la justificación y termina con el proceso de grabado.			
7.-	El Sistema Control de Personal EAGLEYES muestra por pantalla un mensaje del proceso de finalizado con éxito.			
8.-	Ratifica al empleado que la justificación fue ingresada con éxito.			
Observaciones:				

7.6. PROCESO DEL INGRESO DE MULTAS



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2	
		Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Narrativa de Diagrama de Secuencia de Ingreso de Multas.</i>			
NARRATIVA DE DIAGRAMA DE SECUENCIA DE INGRESO DE MULTAS			
1. -	El Sistema Control de Personal EAGLEYES verifica en la Base de Datos las Horas Faltantes de todo el código de empleado designado.		
2. -	Si existen códigos con horas faltantes entonces entrega un valor real de existencia.		
3.-	El Sistema Genera las multas de acuerdo a la tabla de valores y tipos de multas asignadas. Y entrega un reporte de las multas asignadas a dicho empleado.		
4.-	Verifica que no existan incentivos, préstamos o justificaciones. RRHH solicita al Sistema la verificación de las multas.		
5.-	El sistema verifica los valores de préstamos, incentivos o justificaciones. Si no existen ninguno de los valores en la Base de Datos entrega el valor de la verificación.		
6.-	El Sistema le muestra los valores de verificaron a RRHH. El cual aprueba las multas.		
7.-	El Sistema Control de Personal EAGLEYES con el código de aprobación de RRHH descuenta estos valores en el rol de pagos de dicho empleado.		
8.-	El sistema procede a registrar en la Base de Datos las multas de aquel empleado.		
Observaciones:			



CAPÍTULO 8

DIAGRAMAS DE CASOS DE COLABORACIÓN

8. DIAGRAMAS DE CASOS DE COLABORACIÓN

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
DIAGRAMA DE COLABORACIÓN			
<pre> graph TD A([1.-Proceso de Ingreso de persona]) --> B([Procesos de Ingreso de multas]) A --> C([Proceso de Ingreso de Incentivos]) A --> D([Proceso de Ingreso de Justificativos]) E([Proceso de Ingreso y Egreso de Personal]) --> B E --> C E --> D E --> F([Proceso de Reporte de Faltas]) E --> G([Proceso de Reporte de pre-nomina]) </pre>			
Observación:			



CAPÍTULO 9

DIAGRAMAS DE ESTADOS

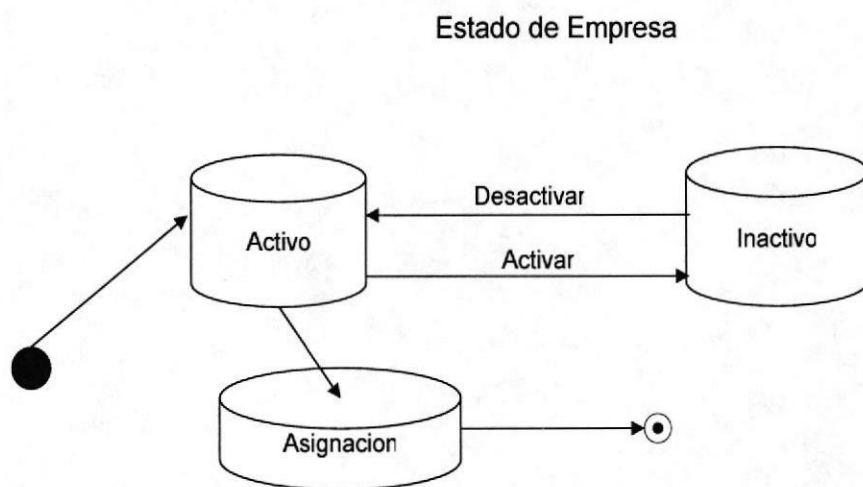
9. DIAGRAMAS DE ESTADO

9.1. ESTADO DEL PERSONAL

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
DISEÑO DEL DIAGRAMA DE ESTADO			
Estado de Personal <pre>stateDiagram-v2 [*] --> Activo Activo --> Inactivo : Desactivar Inactivo --> Activo : Activar Activo --> Asistencia Asistencia --> [*]</pre>			
Observaciones: El estado del personal cambia cuando este tiene un horario libre o franco. Si esta activo puede ser llamado por asistencia.			

9.2. ESTADO DE EMPRESA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	

DISEÑO DEL DIAGRAMA DE ESTADO**Narrativa:**

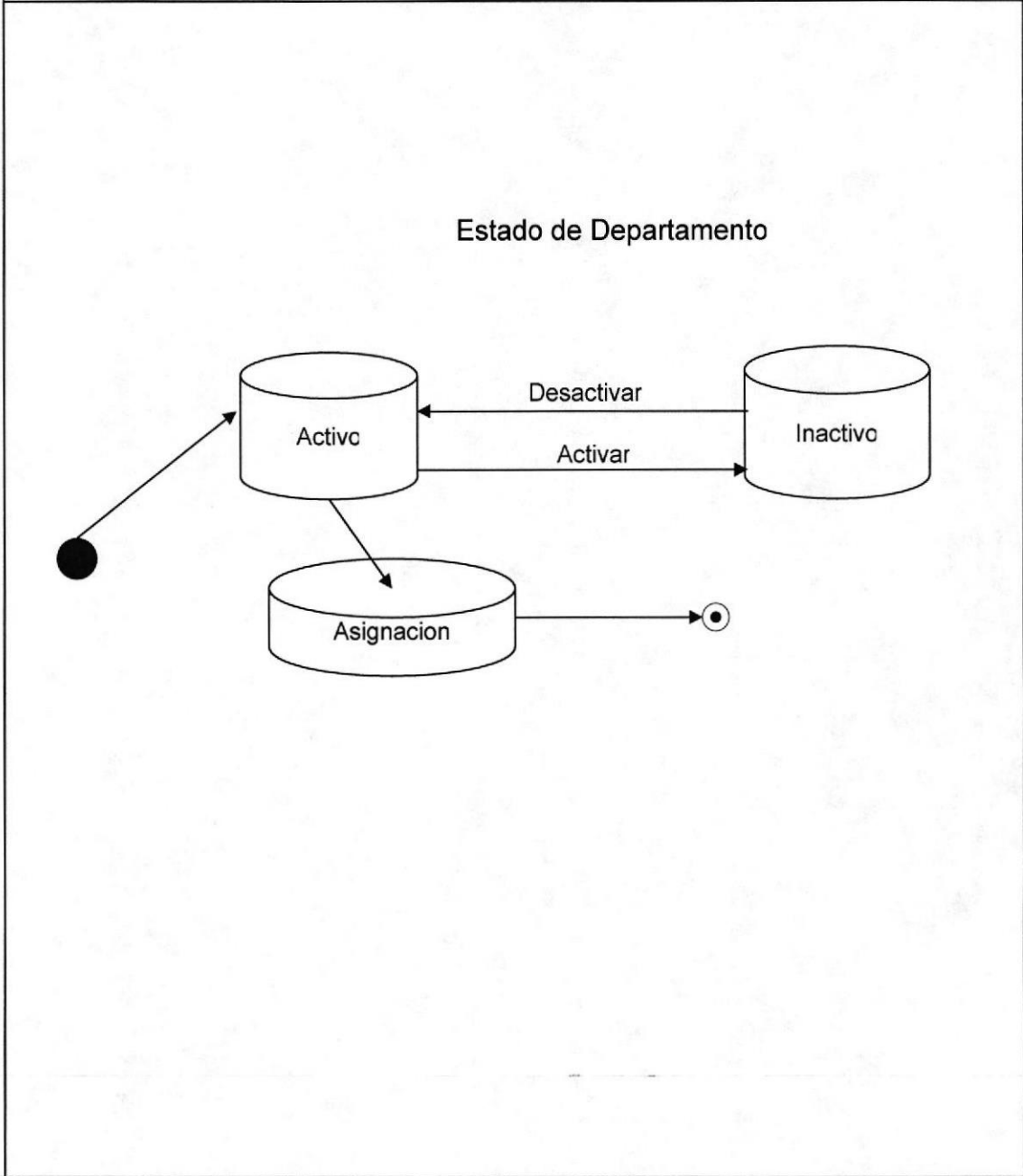
1. El estado activo de una empresa permite ser asignada al personal o a las distintas funciones del sistema.

9.3. ESTADO DE DEPARTAMENTOS



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	

DISEÑO DEL DIAGRAMA DE ESTADO



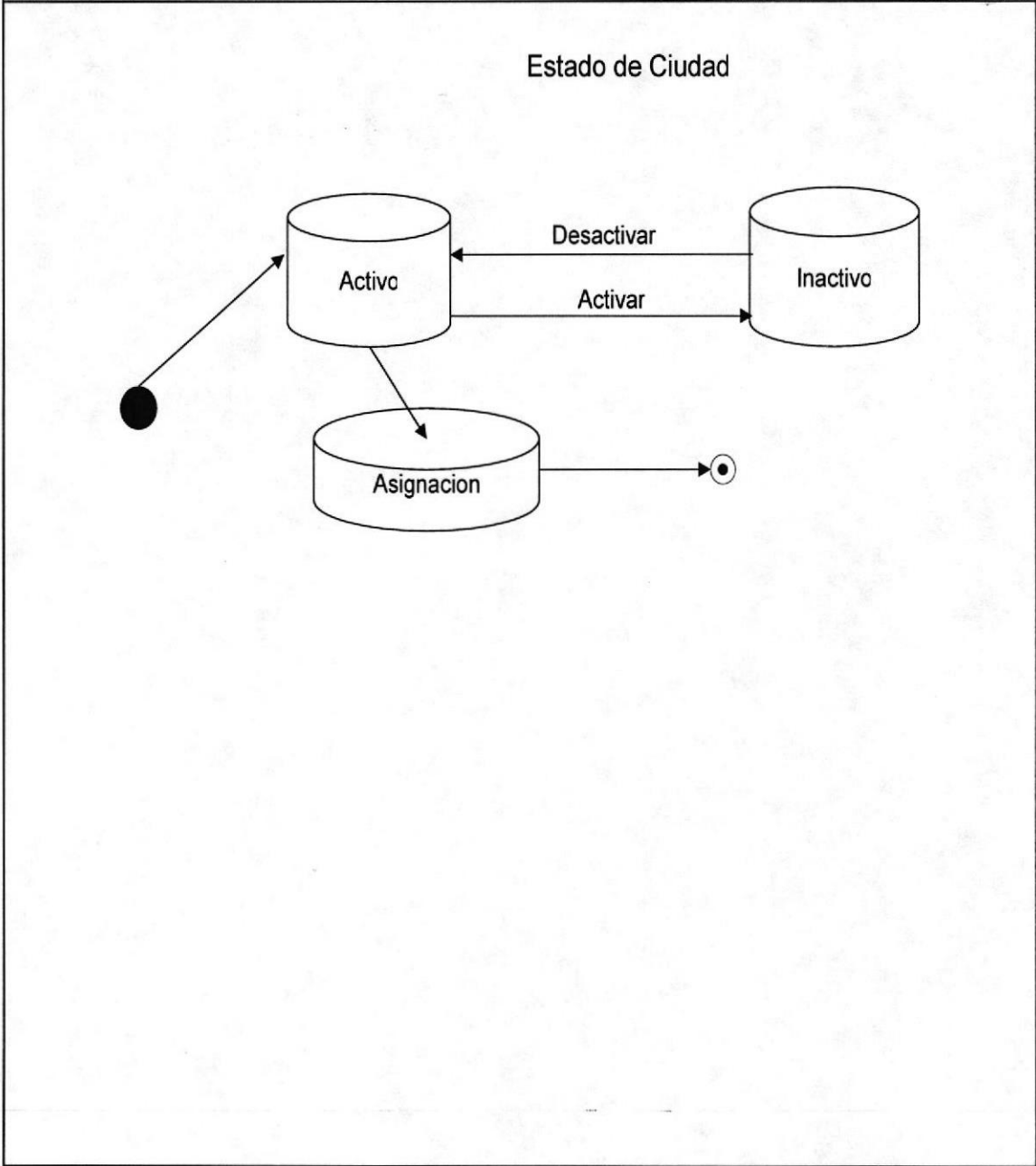
Narrativa:
1. El estado activo de un departamento permite ser asignado al personal o a las distintas funciones del sistema

9.4. ESTADO DE CIUDAD

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	



DISEÑO DEL DIAGRAMA DE ESTADO

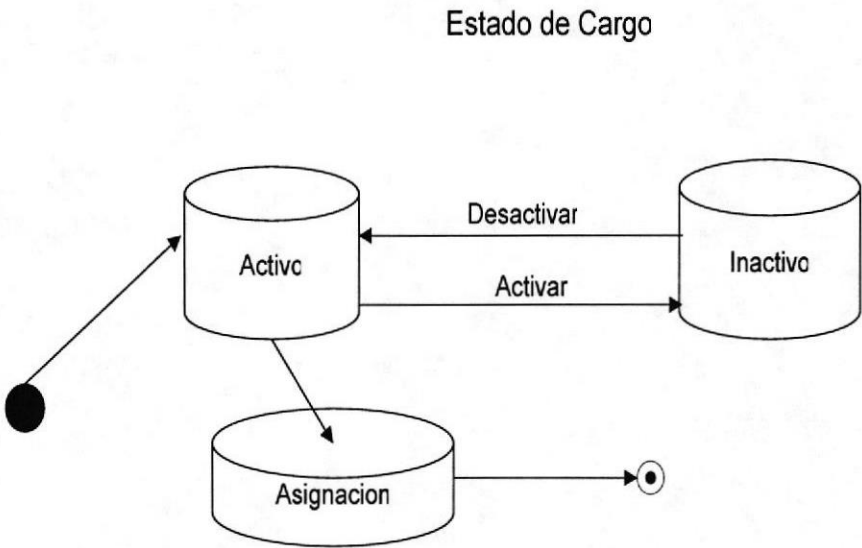


Observaciones:

- 1. El estado de ciudad se utiliza para activar o desactivar la opción de código de una ciudad especificada.

9.5. ESTADO DEL CARGO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/ Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
DISEÑO DEL DIAGRAMA DE ESTADO		



Narrativa:

- 1. El estado de cargo cambia a Inactivo cuando el cupo o vacante para ese cargo esta ocupado o no esta disponible.

9.6. ESTADO DE ADMINISTRADOR

EAGLEYES		Pág. 1/1	
Sistema de Control de Personal		Fecha Diseño:	Fecha/ Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para:	Autores:		
PROTCOM – ESPOL	Carlos Quiroga Marcelo Pinto Wendy Naranjo		
DISEÑO DEL DIAGRAMA DE ESTADO			
<pre> stateDiagram-v2 [*] --> TodosLosPermisos TodosLosPermisos --> Activo TodosLosPermisos --> Inactivo Activo --> Inactivo : Desactivar Inactivo --> Activo : Activar Activo --> ControlTotal ControlTotal --> [*] </pre> The diagram illustrates the state transitions for an administrator. It begins with a start node (solid black circle) pointing to the 'Todos los permisos' state. From 'Todos los permisos', there are transitions to 'Activo' and 'Inactivo' states. A 'Seguridac' state is shown as a separate entity. Between 'Activo' and 'Inactivo', there are two transitions: 'Desactivar' (from 'Activo' to 'Inactivo') and 'Activar' (from 'Inactivo' to 'Activo'). From the 'Activo' state, a transition leads to the 'Control Total' state, which then leads to an end node (circle with a dot).			
Observaciones			
1. El usuario Administrador tiene libre permiso para todas las opciones del sistema(Activas) El nivel de usuario o invitado están restringidas(Inactivo)			



CAPÍTULO 10

ESTANDARIZACIONES DE LOS ELEMENTOS DEL SISTEMA



10. ESTANDARIZACIONES

10.1. FORMATOS DE ALMACENAMIENTO / INFORMACIÓN

Para facilitar la identificación de los elementos del sistema, como son:

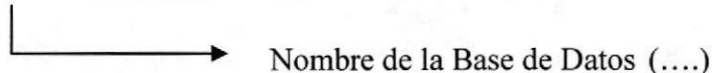
- ✓ Nombre del sistema.
- ✓ Nombre de la base de datos.
- ✓ Nombres de programas.
- ✓ Menús.
- ✓ Formatos de pantallas y reportes.

Se han establecido los siguientes estándares:

10.1.1. FORMATO PARA BASE DE DATOS

ESTÁNDAR DE LA BASE DE DATOS

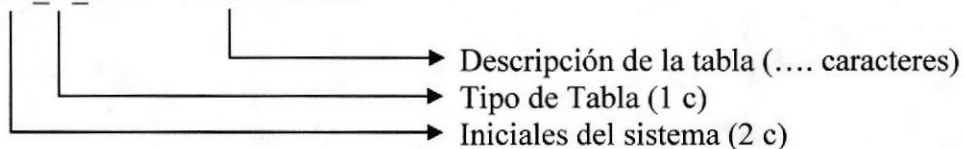
XXXXXXXXXXXXX...



Nombre de la base de Datos: Proyecto.

ESTÁNDAR DE LAS TABLAS

XX X XXXXXXXXXXXXXXXX





INICIALES DEL MÓDULO
Control de personal CP

TIPOS DE TABLAS

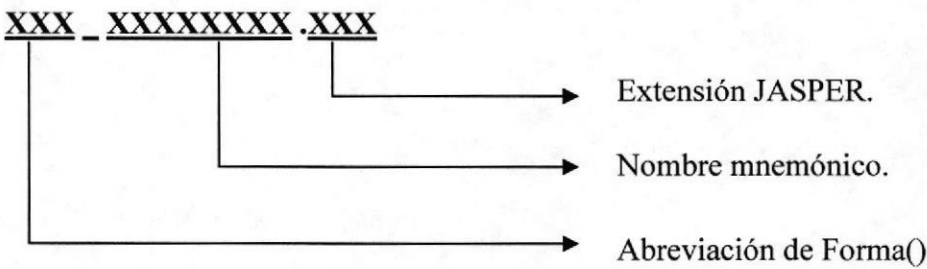
- M Maestra
- H Histórico
- T Transaccional

Tipos de archivos de JAVA y sus extensiones más comunes.

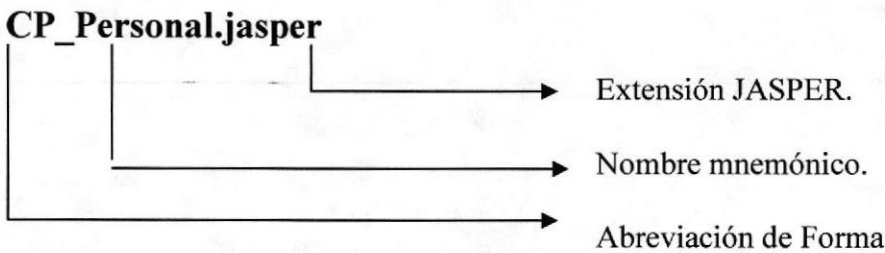
ARCHIVO	EXTENSIÓN
PROYECTO	jcp
BASE DE DATOS	mdb
FORMULARIO	frm
ETIQUETA	lbl
MENÚ	mnu
PROGRAMA	java
REPORTE	cls
CLASE	class

10.1.2. FORMATO PARA EL NOMBRE DE FORMAS Y REPORTE

El formato del nombre de formas y reportes tanto físico como lógico consta de 13 caracteres, el cual se describe a continuación:

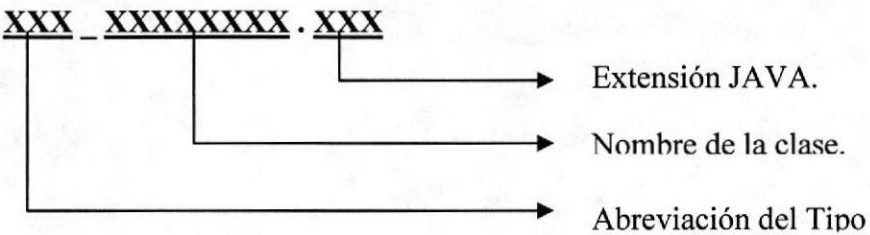


Ejemplo:

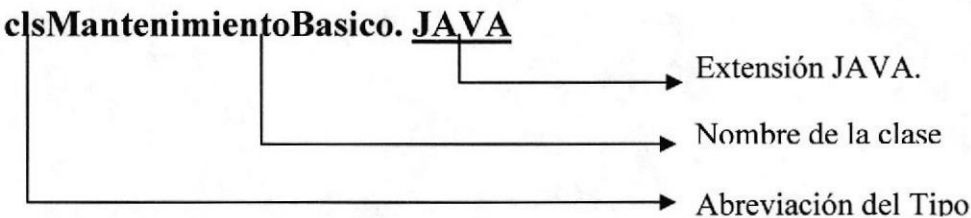


10.1.3. FORMATO PARA EL NOMBRE DE LAS CLASES

El formato de nombres para clases hasta 12 caracteres, el mismo que llevara el siguiente formato:



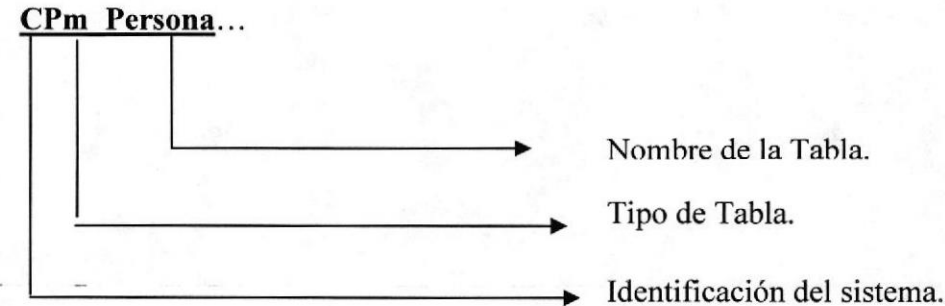
Ejemplo:



IDENTIFICACIÓN DE TABLAS POR TIPO

ID	TABLA
M	Maestra
T	Transaccional
H	Histórica

Ejemplo:

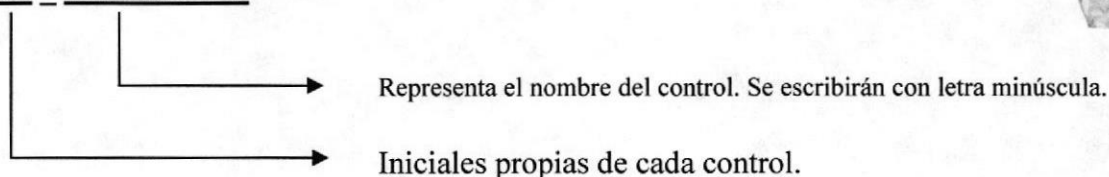


10.2. FORMATOS DE ESTÁNDARES DE PROGRAMACIÓN

10.2.1. FORMATO PARA EL NOMBRE DE LOS CONTROLES

El formato para los nombres de controles es el siguiente:

CCC _ XXXXXXXX



LAS INICIALES DE CADA CONTROL SON LAS SIGUIENTES:

<i>CONTROLES</i>	<i>INICIALES</i>
JLABEL	lbl
JBUTTON	btn
JPANEL	pnl
JTEXTFIELD	txt
JTEXTAREA	txa
JSCROLLPANEL	scr
JCOMBOBOX	cmb
JTABLE	tbl
JCHECKBOX	chk
JRADIOBUTTON	opt
JMENU	mnu
JMENUITEM	smn
JSEPARATOR	spr
JDATEFIELD	txt
JLIST	lst
JTOOLBAR	tbl

Ejemplos:

Un Combo para editar estado civil: cmb_estado civil hereda de la clase clsCombo

Un Textbox para presentar un código: txt_código hereda de la clase clsTextField

Los botones de uso exclusivo de la forma serán ubicados en la parte inferior de la misma de izquierda a derecha.

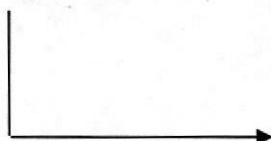


Nota: Con respecto las propiedades de los controles, estas deberán llevar las mismas características que presentan los controles en la pantalla de estándares.

10.2.2. FORMATO PARA EL NOMBRE DE LOS CAMPOS DE LA TABLA

El formato del nombre para los campos de las tablas que forman parte de una base de datos consta hasta 16 letras:

XXXXXXXX....



Nombre del Campo en la Tabla.

Ejemplo:

CPm_Persona : Tabla		
	Nombre del campo	Tipo de datos
	codigo	Autonumérico
	nombre	Texto
	apellido	Texto
	nombrecorto	Texto
	direccion	Texto
	cedula	Texto
	cargo	Número
	dpto	Número
	telefono	Texto

10.3. FORMATOS DE ESTÁNDARES PARA PRESENTACIONES DE PANTALLAS

10.3.1. FORMATOS DE FONTS

El hecho de variar el tamaño y ancho de los fonts, pueda darle o quitarle importancia al texto, evite utilizar italic y sheriff fonts, ya que usualmente son difíciles de leer.

- ✓ Utilice negrita para el texto en los botones y cuando quiera resaltar alguna información.

10.3.2. FORMATOS DE LOCALIZACIÓN DE BOTONES

Ubique los botones principales en la parte inferior de izquierda a derecha. Si necesita botones específicos para cada grupo, ubíquelos internamente.

El tamaño de los botones debe ser aproximadamente de 30 de alto por 135 de ancho.

10.3.3. FORMATOS DE ICONOS

Utilice iconos para representar objetos o tareas. Cuando diseñe un icono tenga en cuenta que debe representar su función.



<i>ICONOS</i>	<i>DESCRIPCIÓN</i>
	Ingresar nuevo registro
	Para aceptar registro consultado
	Para guardar
	Modificar registro
	Cancelar proceso
	Imprimir registro
	Cerrar proceso
	Salir del sistema



CAPÍTULO 11

PLAN DE CÓDIGOS



11. PLAN DE CÓDIGOS

11.1. TABLAS MAESTRAS

11.1.1. TABLA CPm_Persona

TABLA MAESTRAS CPm_Persona			
TIPO	NUMERICO	NOMBRE LÓGICO	CODIGO DE PERSONA
LONGITUD	6	NOMBRE FÍSICO	Per_codigo
FORMATO			
9999999 Ejemplo: 123456 → Código secuencial de 6 dígitos.			

11.1.2. TABLA CPm_Cargo

TABLA MAESTRAS CPm_Cargo			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE CARGO
LONGITUD	4	NOMBRE FÍSICO	Crg_codigo
FORMATO			
9999 Ejemplo: 1456 → Número Secuencial de 4 dígitos.			

11.1.3. TABLA CPm_Departamento

TABLA MAESTRAS CPm Departamento			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE DEPARTAMENTO
LONGITUD	4	NOMBRE FÍSICO	Dpt_codigo
FORMATO			
<u>9999</u> Ejemplo: <u>9856</u> └───────────▶ Número Secuencial de 4 dígitos.			

11.1.4. TABLA CPm Profesion

TABLA MAESTRAS CPM Profesion			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE PROFESION
LONGITUD	4	NOMBRE FÍSICO	Prf_código
FORMATO			
<u>9999</u> Ejemplo: <u>0001</u> └───────────▶ Número Secuencial de 4 dígitos.			



11.1.5. TABLA CPm_TipoIncentivo

TABLA MAESTRAS CPm_TipoIncentivo			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE TIPO DE INCENTIVO
LONGITUD	4	NOMBRE FÍSICO	Tin_Código
FORMATO			
9999 Ejemplo: 0001 Número Secuencial de 4 dígitos.			

11.1.6. TABLA CPm_TipoJustificativo

TABLA MAESTRAS CPm_TipoJustificativo			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE TIPO DE JUSTIFICATIVO
LONGITUD	3	NOMBRE FÍSICO	Tju_codigo
FORMATO			
999 Ejemplo: 156 Número Secuencial de 3 dígitos.			

11.1.7. TABLA CPm_Tipomulta

TABLA MAESTRAS CPm Tipomulta			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE TIPO DE MULTA
LONGITUD	4	NOMBRE FÍSICO	Tmu_código
FORMATO			
<u>9999</u> Ejemplo: <u>8477</u> └───────────▶ Número Secuencial de 4 dígitos.			

11.1.8. TABLA CPm_Ciudad

TABLA MAESTRAS CPm_Ciudad			
TIPO	NUMERICO	NOMBRE LÓGICO	CODIGO DE CIUDAD
LONGITUD	2	NOMBRE FÍSICO	Ciu_código
FORMATO			
99 Ejemplo: 21 └──────────→ Código secuencial de 2 dígitos.			



11.1.9. TABLA CPm_Empresa

TABLA MAESTRAS CPm_Empresa			
TIPO	NUMERICO	NOMBRE LÓGICO	CODIGO DE EMPRESA
LONGITUD	3	NOMBRE FÍSICO	emp_código
FORMATO			
999 Ejemplo: 145 → Código secuencial de 3 dígitos.			

11.1.10. TABLA CPm_Turno

TABLA MAESTRAS CPm_Turno			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE TURNO
LONGITUD	4	NOMBRE FÍSICO	tur_código
FORMATO			
<u>9999</u> Ejemplo: <u>0012</u> → Número Secuencial de 4 dígitos.			

11.1.11. TABLA CPm Horario

TABLA MAESTRAS CPm Horario			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE HORARIO
LONGITUD	2	NOMBRE FÍSICO	hor_código
FORMATO			
<u>99</u> Ejemplo: <u>16</u> └───────────▶ Número Secuencial de 2 dígitos.			

11.2. TABLA HISTORICA

11.2.1. TABLA CPh_Horariopersona

TABLA HISTORICAS CPh Horariopersona			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE HORARIO DE PERSONA
LONGITUD	4	NOMBRE FÍSICO	Hpe_código
FORMATO			
<u>9999</u> Ejemplo: <u>9856</u> Número Secuencial de 4 dígitos.			



1 1.3 TABLAS TRANSACCIONALES

11.3.1. TABLA CPt_Asisistencia

TABLA MAESTRAS CPm Asistencia			
TIPO	TEXTO	NOMBRE LÓGICO	CÓDIGO DE ASISTENCIA
LONGITUD	8	NOMBRE FÍSICO	String
FORMATO			
<u>XXXXXXXXXX</u> Ejemplo: <u>QWE123QWE</u> Número Secuencial de 8 caracteres.			

11.3.2. TABLA CPt_Carnets

TABLA MAESTRAS CPm Carnets			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE CARNETS
LONGITUD	4	NOMBRE FÍSICO	cnt_código
FORMATO			
<u>9999</u> Ejemplo: <u>9856</u> Número Secuencial de 4 dígitos.			

11.3.3. TABLA CPt_Multa

TABLA MAESTRAS CPm_Multa			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE MULTA
LONGITUD	4	NOMBRE FÍSICO	Mta_numero
FORMATO			
9999 Ejemplo: 9856 Número Secuencial de 4 dígitos.			

11.3.4. TABLA CPt_Incentivo

TABLA MAESTRAS CPm_Incentivo			
TIPO	NUMÉRICO	NOMBRE LÓGICO	CÓDIGO DE INCENTIVO
LONGITUD	4	NOMBRE FÍSICO	Inc_numero
FORMATO			
9999 Ejemplo: 0001 Número Secuencial de 4 dígitos.			

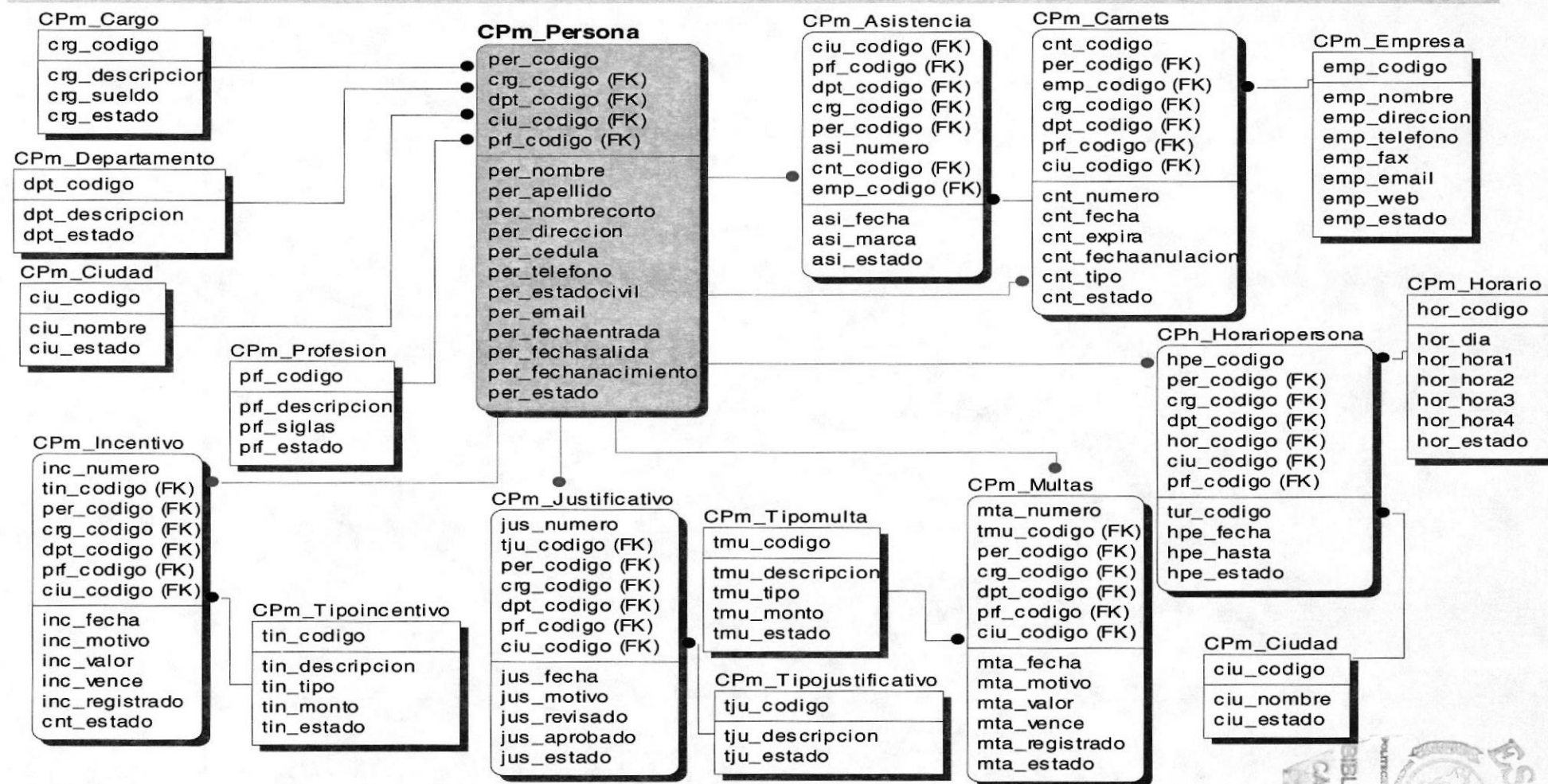


CAPÍTULO 12

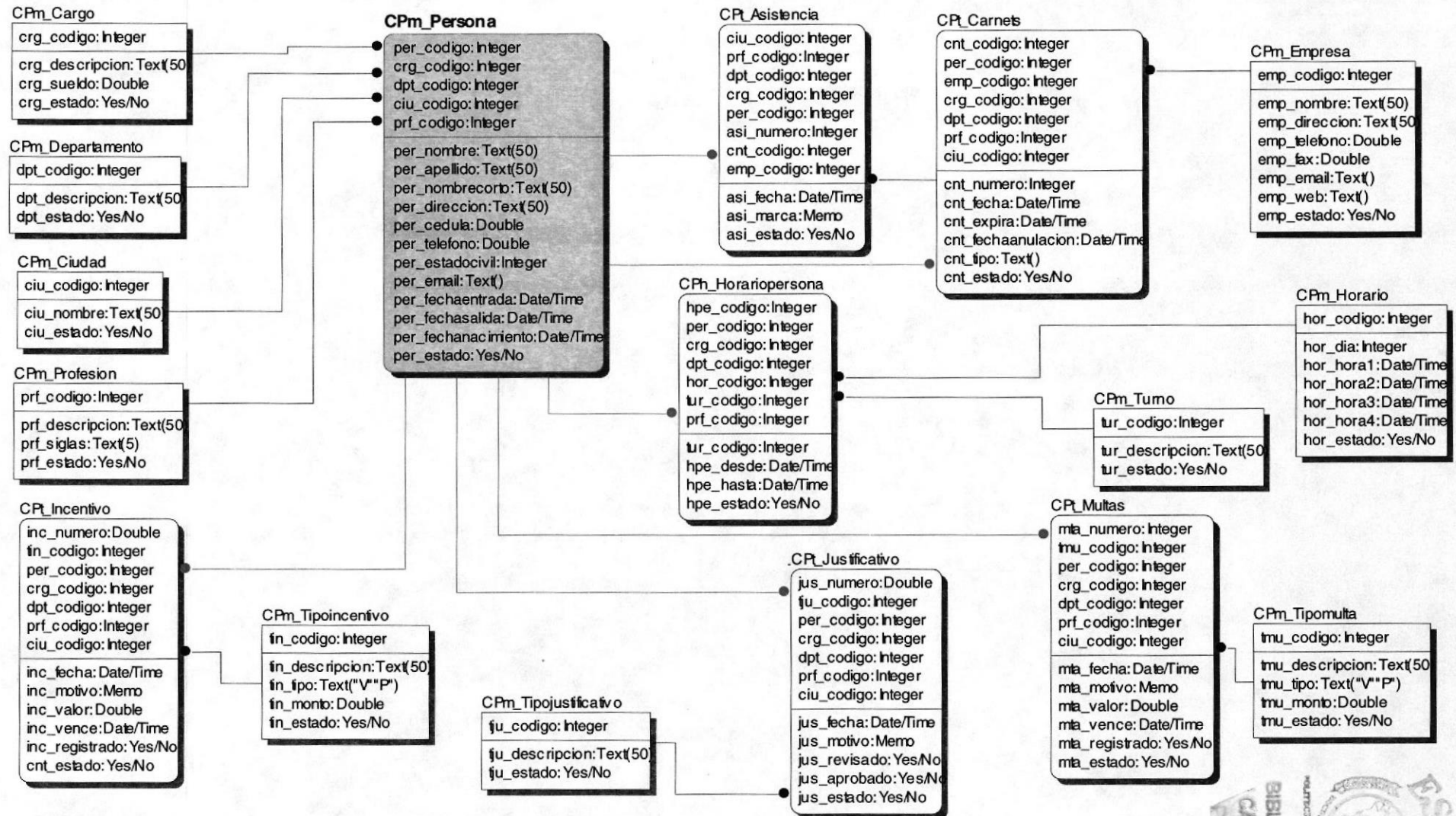
MODELOS DE ENTIDAD-RELACIÓN

12. MODELOS ENTIDAD - RELACIÓN

12.1. MODELO LÓGICO



12.2. MODELO FÍSICO





CAPÍTULO 13

DISEÑO DE LOS ARCHIVOS

13. DISEÑO DE ARCHIVOS

13.1. HISTORIAL DE HORARIO DE PERSONAL

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPh_Horariopersona				Tipo: HISTÓRICA		LONG. REG: 74 bytes	
Descripción: historial horario de personal							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	hpe_código	N	4	Código del horario establecido	✓		
2	per_código	N	4	Código de persona	✓	✓	
3	crg_código	N	4	Código de cargo	✓	✓	
4	dpt_código	N	4	Código del departamento	✓	✓	
5	hor_código	N	4	Código del horario	✓	✓	
6	tur_código	N	4	Código del turno	✓	✓	
7	prf_código	N	4	Código de profesión	✓	✓	
8	tur_código	N	4	Código del turno	✓	✓	
9	hpe_desde	F	20	Fecha desde			
10	hpe_hasta	F	20	Fecha hasta			
11	hpe_estado	B	2	Estado de la tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.2. MANTENIMIENTO DE PERSONA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004	Fecha/Actualización : 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
NOMBRE DE LA TABLA: CPm_Persona				Tipo: MAESTRA	LONG. REG: 177 bytes		
Descripción: mantenimiento de persona							
N o.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NU LL
1	per_código	N	4	Código de persona	✓		
2	crg_código	N	4	Código de cargo	✓	✓	
3	dpt_código	N	4	Código de departamento	✓	✓	
4	ciu_código	N	4	Código de ciudad	✓	✓	
5	prf_código	N	4	Código de profesión	✓	✓	
6	per_nombre	S	30	Nombre			
7	per_apellido	S	30	Apellido			
8	per_nombrecorto	S	20	Pseudónimo			
9	per_dirección	S	30	Dirección			
10	per_cédula	N	10	Cédula			
11	per_teléfono	N	10	Teléfono			
12	per_estadocivil	N	1	Estado civil			✓
13	per_email	S	15	Correo electrónico			
14	per_fechaentrada	D	10	Fecha de entrada			
15	per_fechasalida	D	10	Fecha de salida			
16	per_fechanacimiento	D	10	Fecha nacimiento			
17	per_estado	B	1	Estado de tabla			

Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial.

Per_codigo FK Viene de la tabla persona.



13.3. MANTENIMIENTO DE CARGO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004	Fecha/Actualización : 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
NOMBRE DE LA TABLA: CPM_CARGO				Tipo: MAESTRA		LONG. REG: 45 bytes	
Descripción: mantenimiento de cargo							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	N UL L
1	crg_codigo	N	4	Código de cargo	✓		
2	crg_descripción	S	30	Descripción			
3	crg_sueldo	N	10	Sueldo			
4	crg_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.4. MANTENIMIENTO DE DEPARTAMENTO

EAGLEYES				Pág. 1/1			
Sistema de Control de Personal				Fecha Diseño:	Fecha/Actualización		
VERSIÓN 1.0				2/08/2004	: 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPm_Departamento				Tipo: MAESTRA	LONG. REG: 35 bytes		
Descripción: mantenimiento de departamento							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NUL L
1	dpt_código	N	4	Código del departamento	✓		
2	dpt_descripción	S	30	Descripción			
3	dpt_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							



13.5. MANTENIMIENTO DE CIUDAD

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
NOMBRE DE LA TABLA: CPM_CIUDAD				Tipo: MAESTRA	LONG. REG: 35 bytes		
Descripción: mantenimiento de ciudad							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	ciu_codigo	N	4	Código de ciudad	✓		
2	ciu_nombre	S	30	Nombre			
3	ciu_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.6. MANTENIMIENTO DE PROFESIÓN

EAGLEYES		Pág. 1/1					
Sistema de Control de Personal		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004				
VERSIÓN 1.0							
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo					
NOMBRE DE LA TABLA: CPM_PROFESION		Tipo: MAESTRA	LONG. REG: 37 bytes				
Descripción: mantenimiento de profesión							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	prf_codigo	N	4	Código de profesión	✓		
2	prf_descripción	S	30	Descripción			
3	prf_siglas	S	2	Siglas			
4	prf_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							



13.7. MANTENIMIENTO DE EMPRESA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
NOMBRE DE LA TABLA: CPM_EMPRESA				Tipo: MAESTRA		LONG. REG: 125 bytes	
Descripción: mantenimiento de empresa							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	emp_código	N	4	Código de la empresa	✓		
2	emp_nombre	S	30	Nombre			
3	emp_dirección	S	30	Dirección			
4	emp_teléfono	N	10	Teléfono			
5	emp_fax	N	10	Fax			
6	emp_email	S	20	Correo electrónico			
7	emp_web	S	20	Página web			
8	emp_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.8. MANTENIMIENTO DE HORARIO

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
NOMBRE DE LA TABLA: CPM_HORARIO				Tipo: MAESTRA		LONG. REG: 49 bytes	
Descripción: mantenimiento de horario							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	hor_código	N	4	Código del horaio.	✓		
2	hor_día	N	4	Horario diario			
3	hor_hora1	D	10	Horario			
4	hor_hora2	D	10	Horario			
5	hor_hora3	D	10	Horario			
6	hor_hora4	D	10	Horario			
7	hor_estado	B	1	Estado			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.9. MANTENIMIENTO DE TURNO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPm_Turno				Tipo: MAESTRA		LONG. REG: 35 bytes	
Descripción: mantenimiento de turno							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	tur_código	N	4	Código del turno	✓		
2	tur_descripción	S	30	Descripción.			
3	tur_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							



13.10.MANTENIMIENTO DE TIPO DE INCENTIVO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPm_TipoIncentivo				Tipo: MAESTRA		LONG. REG: 75 bytes	
Descripción: mantenimiento de tipo de incentivo							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	tin_código	N	4	Código del tipo de incentivo	✓		
2	tin_descripción	S	30	Descripción			
3	tin_tipo	S	30	Tipo			
4	tin_monto	N	10	Monto			
5	tin_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.11.MANTENIMIENTO DE TIPO DE JUSTIFICATIVO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1					
		Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004			
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo					
Nombre de la Tabla: CPm_Tipojustificativo		Tipo: MAESTRA		LONG. REG: 35 bytes			
Descripción: mantenimiento de tipo de justificativo							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	tju_código	N	4	Código del tipo de justificativo	✓		
2	tju_descripción	S	30	Descripción			
3	tju_estado	B	1	Estado			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.12. MANTENIMIENTO DE TIPO DE MULTA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1					
		Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004			
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo					
Nombre de la Tabla: CPm_Tipomulta		Tipo: MAESTRA		LONG. REG: 75 bytes			
Descripción: mantenimiento de tipo de multa							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	tmu_código	N	4	Código del tipo de multa	✓		
2	tmu_descripción	S	30	Descripción			
3	tmu_tipo	S	30	Tipo			
4	tmu_monto	N	10	Monto			
5	tmu_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.)							



13.13. TRANSACCIONAL DE INCENTIVO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004	Fecha/Actualización : 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: Cpt_Incentivo				Tipo: TRANSACCIÓN	LONG. REG: 90 bytes		
Descripción: transaccional cabecera de asientos automáticos							
No .	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NUL L
1	inc_número	N	4	Código de incentivo	✓		
2	tin_código	N	4	Código tipo de incentivo	✓		
3	per_código	N	4	Código persona			
4	crg_código	N	4	Código de cargo			
5	dpt_código	N	4	Código de dept			
6	prf_código	N	4	Código de profesión			
7	ciu_código	N	4	Código de ciudad			
8	inc_fecha	D	10	Fecha			
9	inc_motivo	S	30	Motivo			
10	inc_valor	N	10	Valor			
11	inc_vence	D	10	Vencimiento			
12	inc_registrado	B	1	Registro si/no			
13	cnt_estado	B	1	Esatdo de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.14. TRANSACCIONAL DE JUSTIFICATIVO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
Para: PROTCOM – ESPOL				Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004	
				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPt_Justificativo				Tipo: TRANSACCIÓN		LONG. REG: 71 bytes	
Descripción: transaccional cabecera de asientos diarios							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	jus_número	N	4	Código de justificativo	✓		
2	tju_código	N	4	Código tipo justificativo	✓	✓	
3	per_código	N	4	Código persona	✓	✓	
4	crg_código	N	4	Código de cargo	✓		
5	dpt_código	N	4	Código de departamento	✓	✓	
6	prf_código	N	4	Código de profesión	✓	✓	
7	ciu_código	N	4	Código de ciudad	✓	✓	
8	jus_fecha	D	10	Fecha			
9	jus_motivo	S	30	Motivo			
10	jus_revisado	B	1	Revisado si/no			
11	jus_aprobado	B	1	Aprobado si/no			
12	jus_estado	B	1	Estado			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.15. TRANSACCIONAL DE MULTAS



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004		
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPT_Multas				Tipo: TRANSACCIÓN	LONG. REG: 90 bytes		
Descripción: transaccional cabecera de asientos repetitivos							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	mta_número	N	4	Código de multas	✓		
2	tmu_código	N	4	Código tipo multas	✓	✓	
3	per_código	N	4	Código persona	✓	✓	
4	crg_código	N	4	Código cargo	✓	✓	
5	dpt_código	N	4	Código departamento	✓	✓	
6	prf_código	N	4	Código profesión	✓	✓	
7	ciu_código	N	4	Código ciudad	✓	✓	
8	mta_fecha	D	10	Fecha			
9	mta_motivo	S	30	Motivo			
10	mta_valor	N	10	Valor			
11	mta_vence	D	10	Vencimiento			
12	mta_registrado	B	1	Registrado si/no			
13	mta_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.16. TRANSACCIONAL DE ASISTENCIA



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPt_Asiencia				Tipo: TRANSACCIÓN		LONG .REG: 53 bytes	
Descripción: transaccional detalle de asientos automáticos							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	asi_número	N	4	Código de asistencia	✓		
2	prf_código	N	4	Código de profesión	✓	✓	
3	dpt_código	N	4	Código de departamento	✓	✓	
4	crg_código	N	4	Código de cargo	✓	✓	
5	per_código	N	4	Código de persona	✓	✓	
6	ciu_código	N	4	Código de ciudad	✓	✓	
7	cnt_código	N	4	Código de carnets	✓	✓	
8	emp_código	N	4	Código de empresa	✓	✓	
9	asi_fecha	D	10	Fecha			
10	asi_marca	D	10	Marca			
11	asi_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							

13.17. TRANSACCIONAL DE CARNETS

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0				Pág. 1/1			
				Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL				Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo			
Nombre de la Tabla: CPT_Carnets				Tipo: TRANSACCIÓN		LONG.REG: 73 bytes	
Descripción: transaccional detalle de asientos repetitivos							
No.	NOMBRE	TIPO	LONG	DESCRIPCIÓN	PK	FK	NULL
1	cnt_código	N	4	Código de carnets	✓		
2	per_código	N	4	Código de persona	✓	✓	
3	emp_código	N	4	Código de empresa	✓	✓	
4	crg_código	N	4	Código de cargo	✓	✓	
5	dpt_código	N	4	Código de departamento	✓	✓	
6	prf_código	N	4	Código de profesión	✓	✓	
7	ciu_código	N	4	Código de ciudad	✓	✓	
8	cnt_número	N	4	Número de carnet			
9	cnt_fecha	D	10	Fecha			
10	cnt_expira	D	10	Fecha de expiración			
11	cnt_fechaanulacion	D	10	Fecha de anulación			
12	cnt_tipo	N	10	Tipo			
13	cnt_estado	B	1	Estado de tabla			
Observación: todos los FK tienen de la tabla que representan con su pseudónimo inicial. Per_codigo FK Viene de la tabla persona.							



CAPÍTULO 14

DISEÑO DE LAS ENTRADAS Y SALIDAS

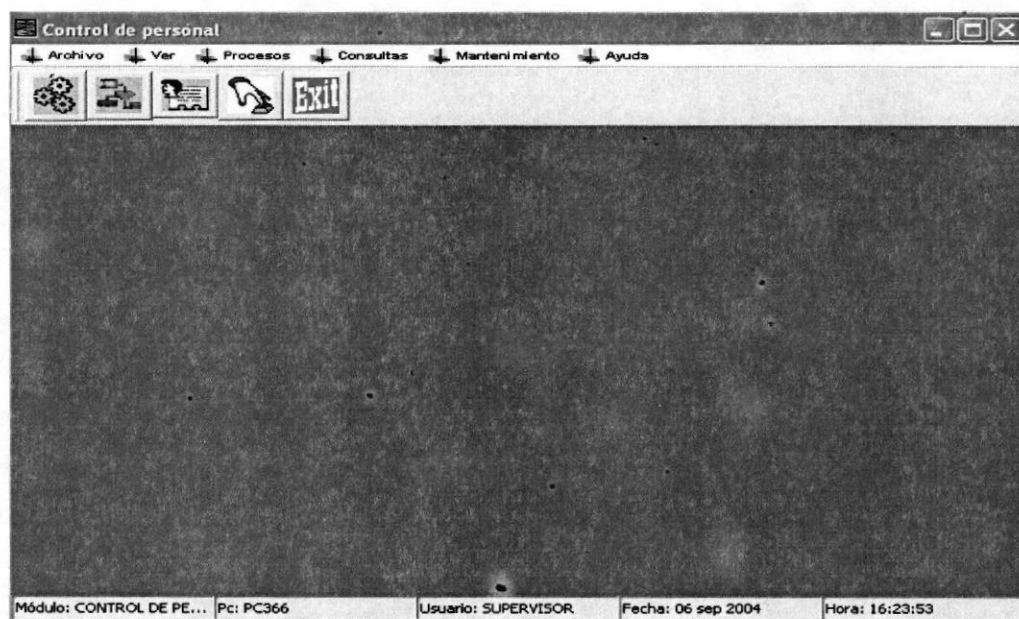
14. DISEÑO DE ENTRADAS Y SALIDAS

14.1. DISEÑO DE ENTRADAS

14.1.1. MENÚ PRINCIPAL DEL SISTEMA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/2	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal.java	Nombre Lógico: frmPrincipal	
Descripción: <i>Menú Principal del Sistema de Control de Asistencia de Personal</i>		

DISEÑO DE PANTALLA

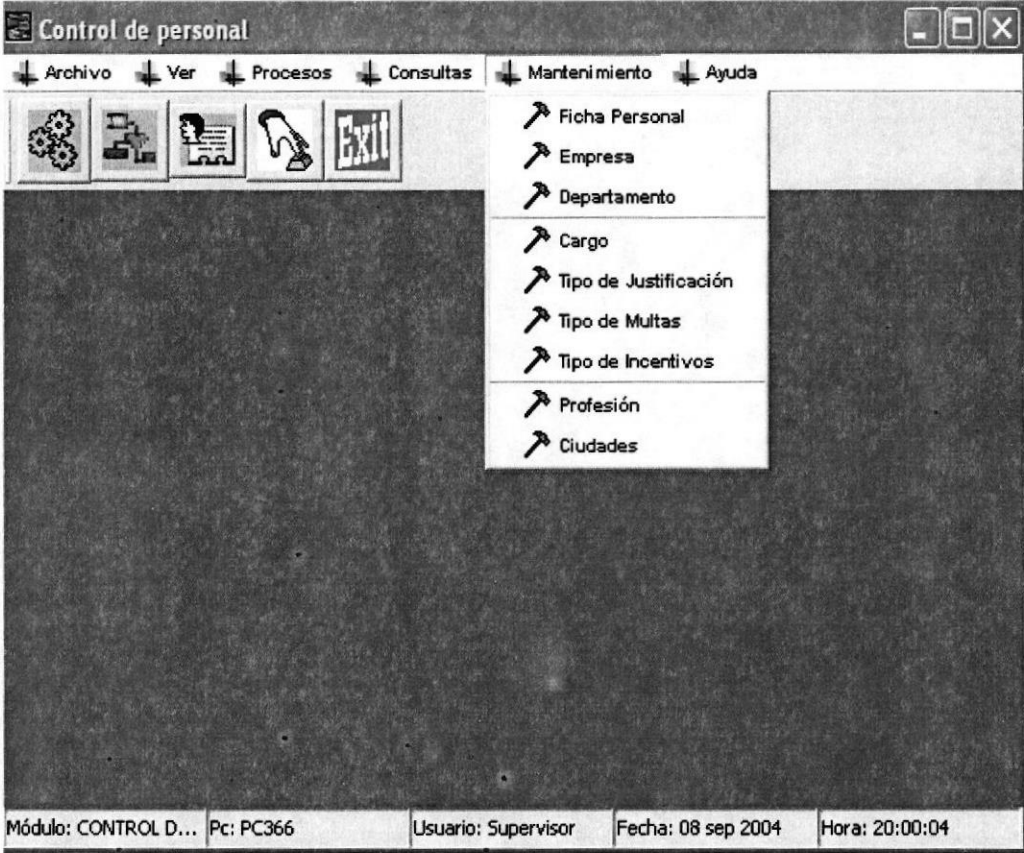

Función:

Este formulario presenta el Menú Principal del Sistema de Control de Asistencia de personal para iniciar las operaciones.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal.java		Nombre Lógico: frmPrincipal	
Descripción: <i>Menú Principal del Sistema de Control de Asistencia de personal</i>			
LISTA DE ELEMENTOS DE DATOS			
NÚM	NOMBRE DEL CONTROL	CONTENIDO	FORMATO DE EDICIÓN
1	Mnu_Barra (Archivo)	Al dar clic se despliega el submenú de Archivo, el cual contiene: Nueva Sesión, y Salir.	
2	Mnu_Barra (Ver)	Al dar clic se despliega un submenú que contiene: Barra de Herramientas, Barra de Estado, y Estilo de Interfase.	
3	Mnu_Barra (Procesos)	Al dar clic se despliega un submenú que contiene: Asistencia, Justificación, Multas e Incentivos.	
4	Mnu_Barra (Consultas)	Al dar clic se despliega el submenú de Consultas, el cual contiene: Asistencia, Personal. Justificación, Multas, Incentivos, Tipo de justificación, Tipo de Multas, Tipo de Incentivos, Cargo, Profesión, Ciudades.	
5	Mnu_Barra (Mantenimiento)	Al dar clic se despliega el submenú de Mantenimientos, el cual contiene: Ficha Personal, Empresa, Departamento, Cargo, Tipo de justificación, Tipo de Multas, Tipo de Incentivos, Profesión, Ciudades.	
6	Mnu_Barra (Ayuda)	Al dar clic se despliega el submenú de Ayuda, el cual contiene: Acerca de:	



14.1.2. MENÚ MANTENIMIENTO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal.java	Nombre Lógico: frmPrincipal		
Descripción: <i>Menú Mantenimiento del Sistema de Control de Asistencia de Personal</i>			
DISEÑO DE PANTALLA			
 The screenshot shows a software window titled 'Control de personal'. It has a menu bar with 'Archivo', 'Ver', 'Procesos', 'Consultas', 'Mantenimiento', and 'Ayuda'. Below the menu bar is a toolbar with icons for settings, a printer, a person, a hand, and an 'Exit' button. The 'Mantenimiento' menu is open, displaying a list of options: 'Ficha Personal', 'Empresa', 'Departamento', 'Cargo', 'Tipo de Justificación', 'Tipo de Multas', 'Tipo de Incentivos', 'Profesión', and 'Ciudades'. At the bottom of the window, a status bar displays: 'Módulo: CONTROL D...', 'Pc: PC366', 'Usuario: Supervisor', 'Fecha: 08 sep 2004', and 'Hora: 20:00:04'.			
Función: Este formulario presenta el Menú Mantenimiento del Sistema de Control de Asistencia de personal para iniciar las operaciones de mantenimiento.			

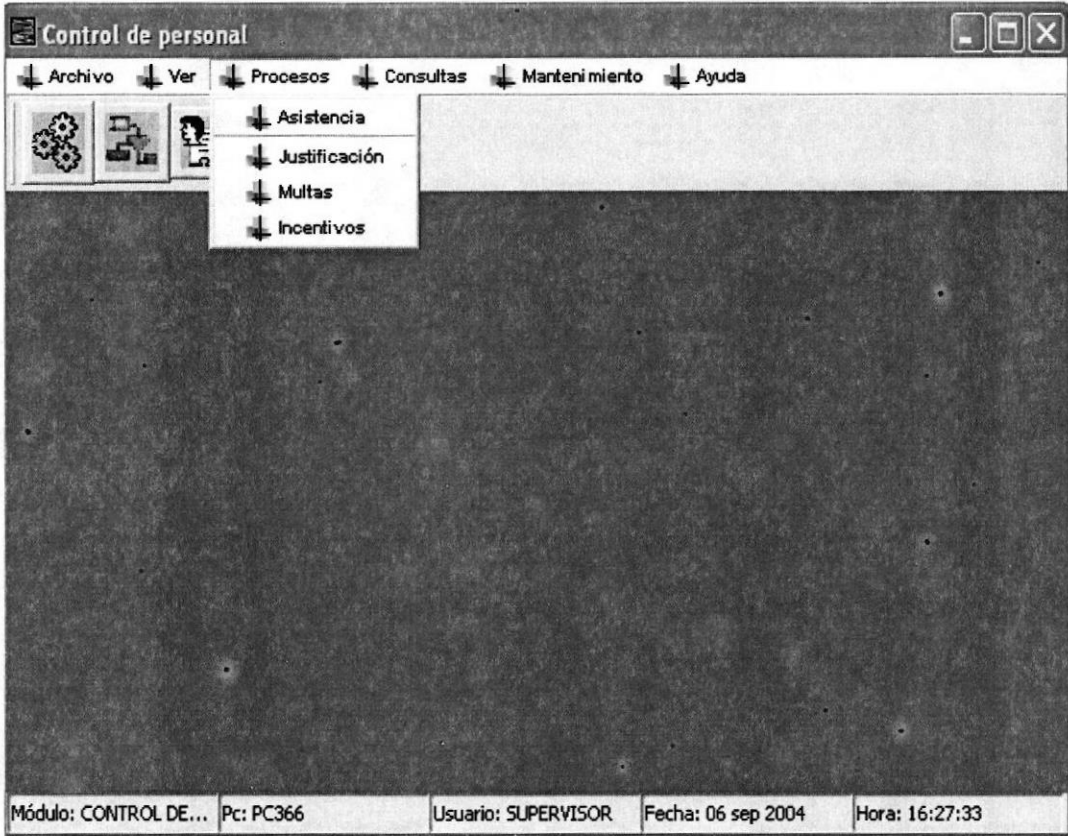
EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal.java		Nombre Lógico: frmPrincipal	
Descripción: <i>Menú Principal del Sistema de Contabilidad</i>			
LISTA DE ELEMENTOS DE DATOS			
NÚM	NOMBRE DEL CONTROL	CONTENIDO	FORMATO DE EDICIÓN
1	itMantenimientoPersonal	Al dar clic sobre esta opción se muestra la ventana de Personal	
2	itMantenimientoEmpresa	Al dar clic sobre esta opción se muestra la ventana de Empresa	
3	itMantenimientoDpto	Al dar clic sobre esta opción se muestra la ventana de departamento	
4	itMantenimientoCargo	Al dar clic sobre esta opción se muestra la ventana de.Cargo	
5	itMantenimientoHorarios	Al dar clic sobre esta opción se muestra la ventana de Horarios	
6	itMantenimientoJustifica	Al dar clic sobre esta opción se muestra la ventana de Justificación	
7	itMantenimientoMulta	Al dar clic sobre esta opción se muestra la ventana de Multas	
8	itMantenimientoIncentivo	Al dar clic sobre esta opción se muestra la ventana de Incentivos	
9	itMantenimientoProfesion	Al dar clic sobre esta opción se muestra la ventana de Profesión	
10	itMantenimientoCiudad	Al dar clic sobre esta opción se muestra la ventana de Ciudad	
Observaciones: En la esquina superior derecha se encuentran los botones estandarizados de Windows: que realizan las operaciones de maximizar, minimizar y cerrar.			

14.1.3. MENÚ PROCESOS



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal		Nombre Lógico: frmPrincipal	
Descripción: <i>Muestra la ventana desplegable de Procesos</i>			

DISEÑO DE PANTALLA

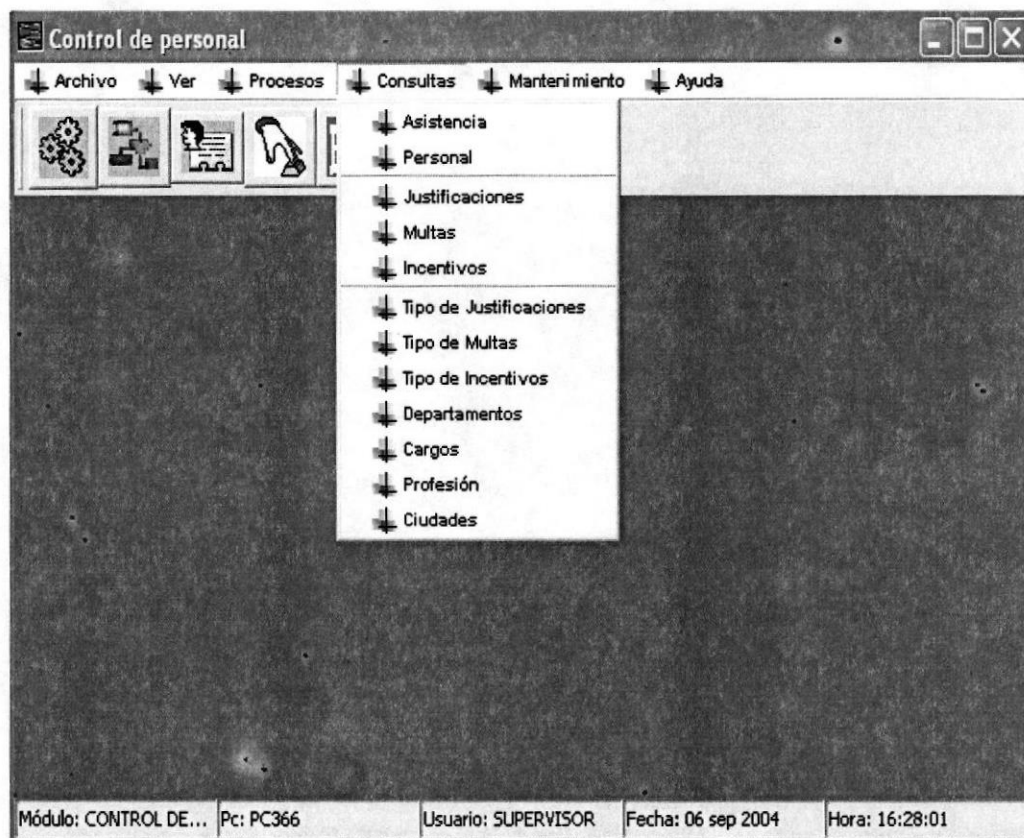


Función:
Este formulario presenta el Menú de Procesos del Sistema de Control de Asistencia de personal en el que se pueden ingresar las asistencias, las justificaciones, multas e incentivos.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización : 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal		Nombre Lógico: frmPrincipal	
Descripción: <i>Muestra la ventana desplegable de Procesos</i>			
LISTA DE ELEMENTOS DE DATOS			
NÚ M	NOMBRE DEL CONTROL	CONTENIDO	FORMATO DE EDICIÓN
1	itProcesoAsistencia	Al dar clic sobre esta opción se muestra la ventana de proceso de Ingreso de Asistencia	
2	itProcesoJustifica	Al dar clic sobre esta opción se muestra la ventana de Proceso de Ingreso de Justificación	
3	itProcesoMulta	Al dar clic sobre esta opción se muestra la ventana de Proceso de Ingreso de multas	
4	itProcesoIncentivo	Al dar clic sobre esta opción se muestra la ventana de Proceso de ingreso de Incentivos	
Observaciones: En la esquina superior derecha se encuentran los botones estandarizados de Windows: que realizan las operaciones de maximizar, minimizar y cerrar.			

14.1.4. MENÚ CONSULTAS

EAGLEYES Sistema de Control de personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal	Nombre Lógico: frmPrincipal	

Descripción: *Muestra la ventana desplegable de Consultas***DISEÑO DE PANTALLA****Función:**

Este formulario presenta el Menú de Consultas del Sistema de Control de Asistencia de personal en el que se pueden ingresar las asistencias, las justificaciones, multas e incentivos y además los diferentes tipos que existen.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	Pág. 1/2
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmPrincipal		Nombre Lógico: frmPrincipal		
Descripción: <i>Muestra la ventana desplegable de Consultas</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	CONTENIDO	FORMATO DE EDICIÓN	
1	itConsultaAsistencia	Al dar clic sobre esta opción se muestra la ventana de Consulta de Asistencia		
2	itConsultaPersonal	Al dar clic sobre esta opción se muestra la ventana de Consulta de Personal		
3	itConsultaJustifica	Al dar clic sobre esta opción se muestra la ventana de Consulta de Justificaciones		
4	itConsultaMulta	Al dar clic sobre esta opción se muestra la ventana de Consulta de Multas		
5	itConsultaIncentivo	Al dar clic sobre esta opción se muestra la ventana de Consulta de Incentivo		
6	itConsultaTipoJustifica	Al dar clic sobre esta opción se muestra la ventana de Consulta de Tipo de Justificación		
7	itConsultaTipoMulta	Al dar clic sobre esta opción se muestra la ventana de Consulta de Tipo de Multa		
8	itConsultaTipoIncentivo	Al dar clic sobre esta opción se muestra la ventana de Consulta de Tipo de Incentivo		



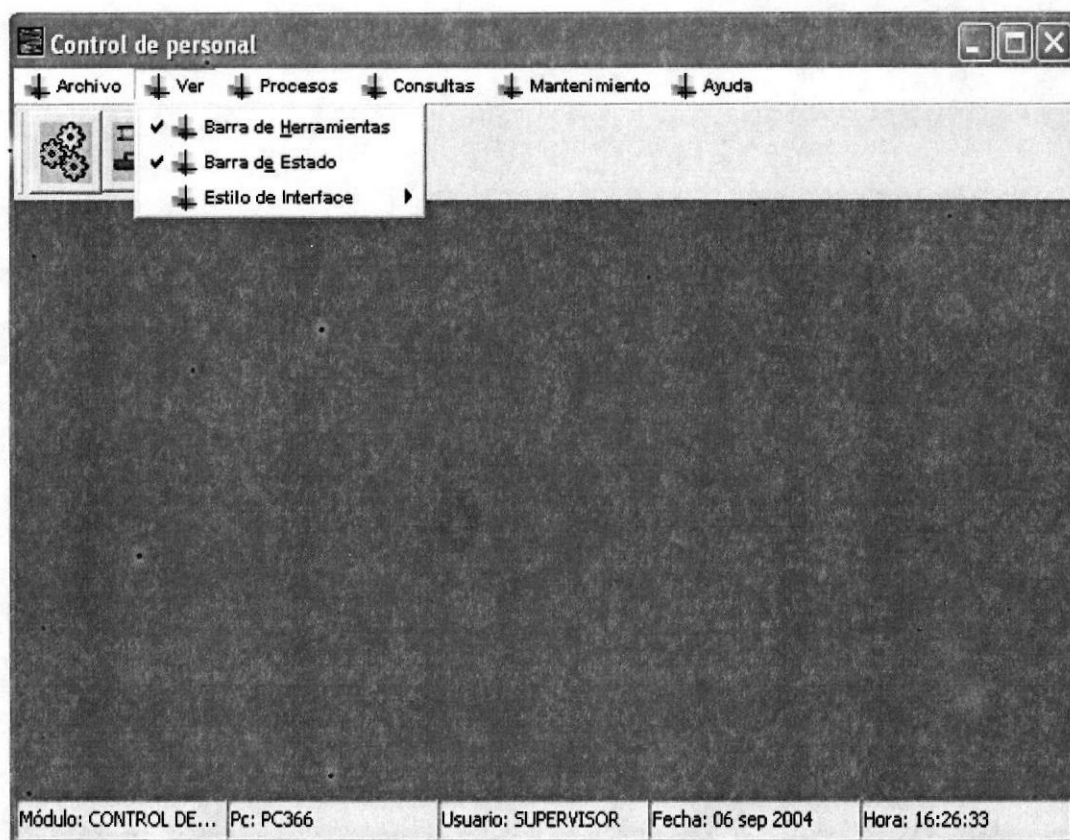
EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	Pág. 2/2
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmPrincipal		Nombre Lógico: frmPrincipal		
Descripción: <i>Muestra la ventana desplegable de Consultas</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	CONTENIDO	FORMATO DE EDICIÓN	
9	itConsultaDpto	Al dar clic sobre esta opción se muestra la ventana de Consulta de Departamento		
10	itConsultaCargo	Al dar clic sobre esta opción se muestra la ventana de Consulta de Cargo		
11	itConsultaProfesion	Al dar clic sobre esta opción se muestra la ventana de Consulta de Profesión		
12	itConsultaCiudad	Al dar clic sobre esta opción se muestra la ventana de Consulta de Ciudad		
Observaciones: En la esquina superior derecha se encuentran los botones estandarizados de Windows: que realizan las operaciones de maximizar, minimizar y cerrar.				



14.1.5. MENÚ VER

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal	Nombre Lógico: frmPrincipal	
Descripción: Descripción: <i>Muestra la ventana desplegable de menú Ver</i>		

DISEÑO DE PANTALLA

**Función:**

Este formulario presenta el Menú Ver del Sistema de Control de Asistencia de personal en el que puedes escoger si deseas ver la barra de herramientas o no, verificar su estado y escoger el estilo de interfase.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPrincipal		Nombre Lógico: frmPrincipal	
Descripción: <i>Muestra la ventana desplegable de menú Ver</i>			
LISTA DE ELEMENTOS DE DATOS			
NÚ M	NOMBRE DEL CONTROL	CONTENIDO	FORMATO DE EDICIÓN
1	itVerBarraHerramientas	Al dar clic sobre esta opción se habilita o deshabilita la barra de Herramientas	
2	itVerBarraEstado	Al dar clic sobre esta opción se habilita o deshabilita la barra de Estado	
3	itVerInterface	Al dar clic sobre esta opción se muestra el submenú para escoger la interfase grafica del sistema	
Observaciones: En la esquina superior derecha se encuentran los botones estandarizados de Windows: que realizan las operaciones de maximizar, minimizar y cerrar.			

14.1.6. MANTENIMIENTO PERSONA



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 20/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmPersonal.java	Nombre Lógico: frmPersonal	

Descripción: *Muestra la ventana de mantenimiento de Persona.*

DISEÑO DE PANTALLA

Función:

Este formulario presenta la pantalla de Mantenimiento de Personal del Sistema de Control de Asistencia.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1		
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmPersonal.java		Nombre Lógico: frmPersonal		
Descripción: <i>Muestra la ventana de mantenimiento de Persona.</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚ M	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
clsMantenimientoTipo3				
1	txtCódigo	txtCodigo	Código de persona	9(4)
2	txtCedRuc	txtCedRuc	Cédula O Ruc	9(10)
3	txtNombre	txtNombre	Nombre de persona	X(30)
4	txtDirección	txtDireccion	Dirección de persona	X(30)
5	txtTeléfono	txtTelefono	Teléfono de persona	9(10)
6	txtEmail	txtEmail	Email de persona	X(20)
7	btnExaminarCod	btnExaminarCod	Botón de Examinar código	
8	btnExaminarCed	btnExaminarCed	Botón de Examinar Cedula	
frmPersonal				
9	txtApellido	txtApellido	Descripción del Módulo	X(30)
10	txtNombreCorto	txtNombreCorto		X(10)
11	txtFNacimiento	txtFNacimiento		dd/mm/yyyy
12	txtFEntrada	txtFEntrada		dd/mm/yyyy
13	txtFSalida	txtFSalida		dd/mm/yyyy
14	cmbEstadoCivil	cmbEstadoCivil		
15	cmbLNacimiento	cmbLNacimiento		
16	cmbProfesión	cmbProfesion		
17	cmbDpto	cmbDpto		
18	cmbCargo	cmbCargo		
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

14.1.7. MANTENIMIENTO DE EMPRESA

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: FrmEmpresa.java	Nombre Lógico: frmEmpresa	

Descripción: *Muestra la ventana de mantenimiento de Empresa*

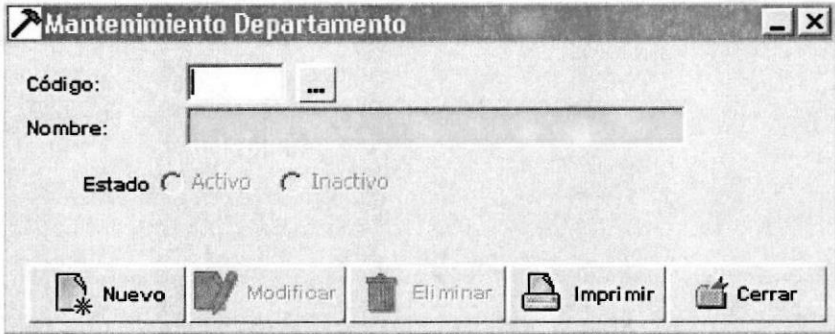
DISEÑO DE PANTALLA

Función:

Este formulario presenta la pantalla de Mantenimiento de Empresa del Sistema de Control de Asistencia.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1		
		Fecha Diseño: 2/08/2004	Fecha/Actualizaci n: 20/08/2004	
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmEmpresa.java		Nombre Lógico: frmEmpresa		
Descripción: Muestra la ventana de mantenimiento de Empresa				
clsMantenimientoTipo3				
NÚM	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
1	txtCodigo	txtCodigo	Código de Empresa	9(4)
2	txtCedRuc	txtCedRuc	Cedula O RUC	9(10)
3	txtNombre	txtNombre	Nombre de Empresa	X(30)
4	txtDireccion	txtDireccion	Dirección de Empresa	X(30)
5	txtTelefono	txtTelefono	Teléfono de Empresa	9(10)
6	txtEmail	txtEmail	Email de Empresa	X(10)
7	btnExaminarCod	btnExaminarCod	Botón Examinar Código	
8	btnExaminarCed	btnExaminarCed	Botón Examinar Cedula	
frmEmpresa				
9	txtFax	txtFax	Fax de la Empresa	9(10)
10	txtWeb	txtWeb	Web de la Empresa	X(20)
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

14.1.8. MANTENIMIENTO DE DEPARTAMENTOS

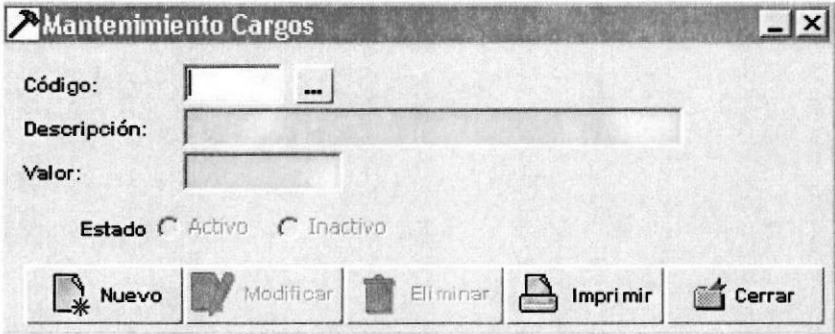
EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmDepartamento.java	Nombre Lógico: frmDepartamento		
Descripción: <i>Muestra la ventana de mantenimiento de Departamento</i>			
DISEÑO DE PANTALLA			
			
Función: Este formulario presenta la pantalla de Mantenimiento de Departamento del Sistema de Control de Asistencia.			



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1		
		Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmDepartamento.java		Nombre Lógico: frmDepartamento		
Descripción: <i>Muestra la ventana de mantenimiento de Departamento</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
clsMantenimientoTipol				
	txtCodigo	txtCodigo	Código de Departamento	9(4)
	txtDescripcion	txtDescripcion	Descripción de Departamento	X(30)
	btnExaminar	btnExaminar	Botón Examinar	
frmDepartamento				
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

14.1.9. MANTENIMIENTO DE CARGO



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmCargo.java	Nombre Lógico: frmCargo	
Descripción: <i>Muestra la ventana de mantenimiento de cargo</i>		
DISEÑO DE PANTALLA		
		
Función: Este formulario presenta la pantalla de Mantenimiento de Cargos del Sistema de Control de Asistencia.		



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	Pág. 1/1
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmCargo.java		Nombre Lógico: frmCargo		
Descripción: <i>Muestra la ventana de mantenimiento de cargo</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
clsMantenimientoTipo2				
1	txtMonto	txtMonto	Monto de Cargo	9(4)
clsMantenimientoTipo1				
2	txtCodigo	txtCodigo	Código de Cargo	9(4)
3	txtDescripcion	txtDescripcion	Descripcion de Cargo	X(30)
4	btnExaminar	btnExaminar	Botón Examinar	
frmCargo				
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

14.1.10. MANTENIMIENTO DE TIPO DE JUSTIFICACIONES

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño:	Fecha/Actualización:
		2/08/2004	20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico: frmJustifica.java		Nombre Lógico: frmJustifica	

Descripción: Muestra la ventana de mantenimiento de Justificativo.

DISEÑO DE PANTALLA

Función:

Este formulario presenta la pantalla de Mantenimiento de Tipo de Justificativo del Sistema de Control de Asistencia.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	Pág. 1/1
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: frmJustifica.java		Nombre Lógico: frmJustifica		
Descripción: <i>Muestra la ventana de mantenimiento de Justificativo</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
clsConsultaBase				
1	btnConsultar	btnConsultar	Botón de Consultar	
2	btnImprimir	btnImprimir	Botón de Imprimir	
3	btnCerrar	btnCerrar	Botón de Cerrar	
clsMantenimientoTipo1				
4	txtCodigo	txtCodigo	Código de Cargo	9(4)
5	txtDescripcion3	txtDescripcion	Descripcion de Cargo	X(30)
6	btnExaminar	btnExaminar	Botón Examinar	
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

14.1.11. MANTENIMIENTO DE TIPO DE MULTAS

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico:	Nombre Lógico:	

Descripción: Muestra la ventana de mantenimiento de Tipo de Multas.

DISEÑO DE PANTALLA

Función:

Este formulario presenta la pantalla de Mantenimiento de Tipo de Multas del Sistema de Control de Asistencia.



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	Pág. 1/1
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: CON M AsientosPredeterminados.java		Nombre Lógico: Frm MantenimientoPredeterminados		
Descripción: <i>Mantenimiento Asientos Predeterminados</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚ M	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMAT O DE EDICIÓN
CON T CABASTREPETITIVO				
1	txt_Codaplicacion	Mod_codigo	Código Aplicación	9(4)
2	txt_NoAsiento	Cat_codigo	Nº de asiento	9(4)
4	txt_Glosa	Car_Glosa	Glosa	X(30)
5	txt_CodTransa	Car_tipo_transacion	Código Transacción	9(4)
6	cmb_Estado	Car_estado	Estado	X(1)
SEG M MODULOS				
3	txt_Nomaplicacion	Mod_Descripcion	Nombre Aplicación	X(30)
CON M TRANSACCIÓN				
7	txt_NomTransa	Tra_Codigo	Nombre Transacción	X(30)
CON M CUENTAS				
8	txt_Codcuenta	Cta_codigo	Código Cuenta contable	X(13)
9	cmb_TipoSaldo	Cta_naturaleza	Tipo saldo	X(1)
10	txt_Nomcuenta	Cta_nombre	Nombre Cuenta	X(30)
13	txt_Nomcuenta	Cta_nombre	Nombre cuenta	X(30)
CON T DETASTREPETITIVO				
11	tbl_Tabla	Dar_secuencia	Secuencia	9(3)
12	tbl_Tabla	Cta_codigo	Código cuenta	X(50)
14	tbl_Tabla	Dar_Saldo	Tipo saldo	X(25)
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

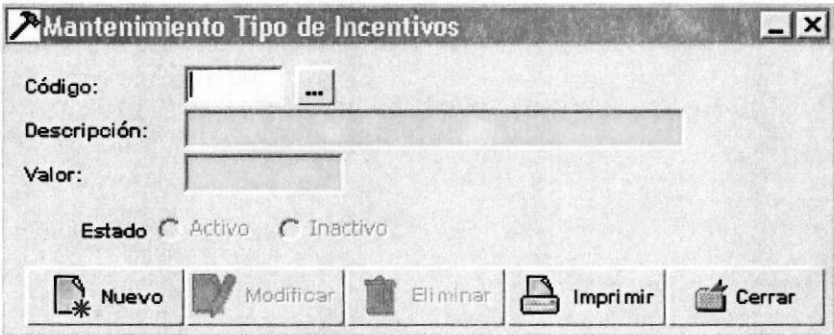


14.1.12. MANTENIMIENTO DE TIPO DE INCENTIVO

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico:	Nombre Lógico:	

Descripción: Muestra la ventana de mantenimiento de Tipo de Incentivos.

DISEÑO DE PANTALLA



Función:
Este formulario presenta la pantalla de Mantenimiento de Tipo de Incentivos del Sistema de Control de Asistencia.

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	Pág. 1/1
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: CON M AsientosPredeterminados.java		Nombre Lógico: Frm MantenimientoPredeterminados		
Descripción: <i>Mantenimiento Asientos Predeterminados</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
CON T CABASTREPETITIVO				
1	txt_Codaplicacion	Mod_codigo	Código Aplicación	9(4)
2	txt_NoAsiento	Cat_codigo	N° de asiento	9(4)
4	txt_Glosa	Car_Glosa	Glosa	X(30)
5	txt_CodTransa	Car_tipo_transacion	Código Transacción	9(4)
6	cmb_Estado	Car_estado	Estado	X(1)
SEG M MODULOS				
3	txt_Nomaplicacion	Mod_Descripcion	Nombre Aplicación	X(30)
CON M TRANSACCIÓN				
7	txt_NomTransa	Tra_Codigo	Nombre Transacción	X(30)
CON M CUENTAS				
8	txt_Codcuenta	Cta_codigo	Código Cuenta contable	X(13)
9	cmb_TipoSaldo	Cta_naturaleza	Tipo saldo	X(1)
10	txt_Nomcuenta	Cta_nombre	Nombre Cuenta	X(30)
13	txt_Nomcuenta	Cta_nombre	Nombre cuenta	X(30)
CON T DETASTREPETITIVO				
11	tbl_Tabla	Dar_secuencia	Secuencia	9(3)
12	tbl_Tabla	Cta_codigo	Código cuenta	X(50)
14	tbl_Tabla	Dar_Saldo	Tipo saldo	X(25)
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

14.1.13. MANTENIMIENTO DE PROFESION

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico:	Nombre Lógico:	

Descripción: Muestra la ventana de mantenimiento de Profesión.

DISEÑO DE PANTALLA

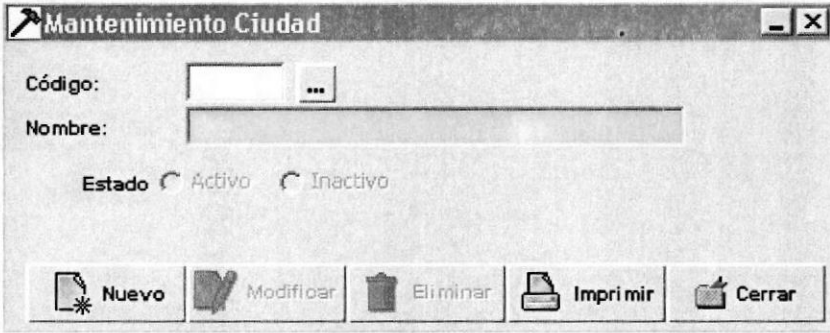
Función:

Este formulario presenta la pantalla de Mantenimiento de Profesión del Sistema de Control de Asistencia.



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1		
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: CON M AsientosPredeterminados.java		Nombre Lógico: Frm MantenimientoPredeterminados		
Descripción: <i>Mantenimiento Asientos Predeterminados</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
CON T CABASTREPETITIVO				
1	txt_Codaplicacion	Mod_codigo	Código Aplicación	9(4)
2	txt_NoAsiento	Cat_codigo	N° de asiento	9(4)
4	txt_Glosa	Car_Glosa	Glosa	X(30)
5	txt_CodTransa	Car_tipo_transacion	Código Transacción	9(4)
6	cmb_Estado	Car_estado	Estado	X(1)
SEG M MODULOS				
3	txt_Nomaplicacion	Mod_Descripcion	Nombre Aplicación	X(30)
CON M TRANSACCIÓN				
7	txt_NomTransa	Tra_Codigo	Nombre Transacción	X(30)
CON M CUENTAS				
8	txt_Codcuenta	Cta_codigo	Código Cuenta contable	X(13)
9	cmb_TipoSaldo	Cta_naturaleza	Tipo saldo	X(1)
10	txt_Nomcuenta	Cta_nombre	Nombre Cuenta	X(30)
13	txt_Nomcuenta	Cta_nombre	Nombre cuenta	X(30)
CON T DETASTREPETITIVO				
11	tbl_Tabla	Dar_secuencia	Secuencia	9(3)
12	tbl_Tabla	Cta_codigo	Código cuenta	X(50)
14	tbl_Tabla	Dar_Saldo	Tipo saldo	X(25)
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				

14.1.14. MANTENIMIENTO DE CIUDADES

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Nombre Físico:	Nombre Lógico:	
Descripción: Muestra la ventana de mantenimiento de Ciudad.		
DISEÑO DE PANTALLA		
		
Función: Este formulario presenta la pantalla de Mantenimiento de Ciudad del Sistema de Control de Asistencia.		

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1		
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Nombre Físico: CON M AsientosPredeterminados.java		Nombre Lógico: Frm MantenimientoPredeterminados		
Descripción: <i>Mantenimiento Asientos Predeterminados</i>				
LISTA DE ELEMENTOS DE DATOS				
NÚM	NOMBRE DEL CONTROL	NOMBRE DEL CAMPO	CONTENIDO	FORMATO DE EDICIÓN
CON T CABASTREPETITIVO				
1	txt_Codaplicacion	Mod_codigo	Código Aplicación	9(4)
2	txt_NoAsiento	Cat_codigo	N° de asiento	9(4)
4	txt_Glosa	Car_Glosa	Glosa	X(30)
5	txt_CodTransa	Car_tipo_transacion	Código Transacción	9(4)
6	cmb_Estado	Car_estado	Estado	X(1)
SEG M MODULOS				
3	txt_Nomaplicacion	Mod_Descripcion	Nombre Aplicación	X(30)
CON M TRANSACCIÓN				
7	txt_NomTransa	Tra_Codigo	Nombre Transacción	X(30)
CON M CUENTAS				
8	txt_Codcuenta	Cta_codigo	Código Cuenta contable	X(13)
9	cmb_TipoSaldo	Cta_naturaleza	Tipo saldo	X(1)
10	txt_Nomcuenta	Cta_nombre	Nombre Cuenta	X(30)
13	txt_Nomcuenta	Cta_nombre	Nombre cuenta	X(30)
CON T DETASTREPETITIVO				
11	tbl_Tabla	Dar_secuencia	Secuencia	9(3)
12	tbl_Tabla	Cta_codigo	Código cuenta	X(50)
14	tbl_Tabla	Dar_Saldo	Tipo saldo	X(25)
Observaciones: Las cajas de texto son obligatorias, ya que son necesarios para manipular correctamente los datos.				



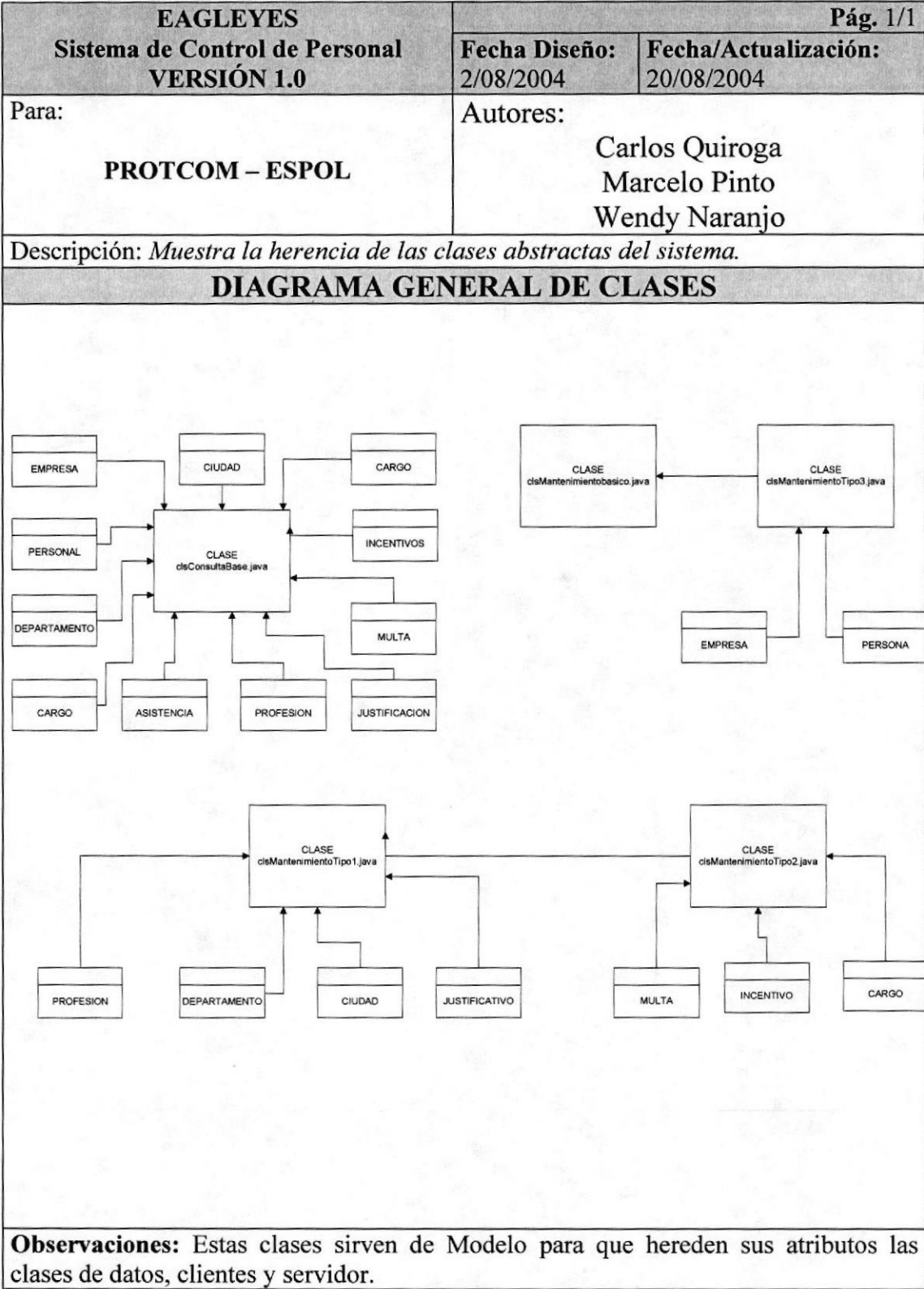
CAPÍTULO 15

EXPLICATIVA DE MODELO DE CLASES Y HERENCIAS



15. EXPLICATIVA DE MODELO DE CLASES Y HERENCIAS

15.1. DIAGRAMA GENERAL DE CLASES



15.2. DIAGRAMA DE CLASE clsConsultaBase

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Muestra la herencia de las clases para las tablas que poseen reportes.</i>			
DIAGRAMA DE CLASE DE HERENCIA DE CLSCONSULTABASE			
<pre>classDiagram class CLASE["CLASE clsConsultaBase.java"] class EMPRESA class CIUDAD class CARGO class PERSONAL class DEPARTAMENTO class ASISTENCIA class PROFESION class INCENTIVOS class MULTA class JUSTIFICACION CLASE < -- EMPRESA CLASE < -- CIUDAD CLASE < -- CARGO CLASE < -- PERSONAL CLASE < -- DEPARTAMENTO CLASE < -- ASISTENCIA CLASE < -- PROFESION CLASE < -- INCENTIVOS CLASE < -- MULTA CLASE < -- JUSTIFICACION</pre>			
Observaciones: Esta es la clase básica de la cual se hereda para todos los reportes del Sistema.			

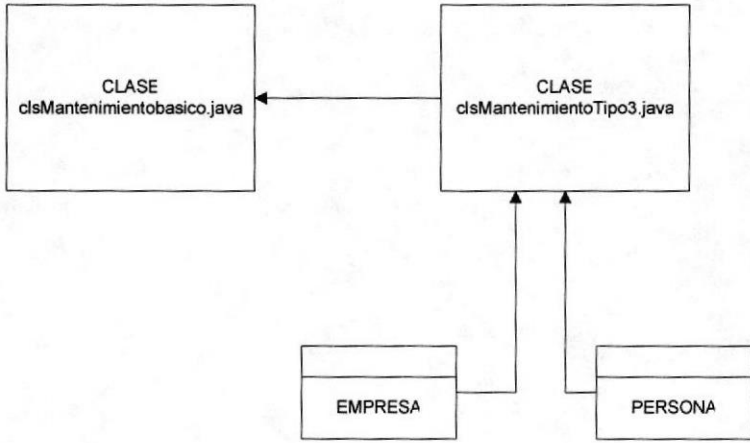
15.3. CLASE clsConsultaBase

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Clase clsConsultaBase.java			
<pre> class clsConsultaBase extends JFrame{ // Etiquetas private JPanel panel = new JPanel(null); private JTable tabla; private DefaultTableModel cnt; // Botones private clsBoton btnImprimir; private clsBoton btnCerrar; clsConsultaBase(String cTitulo,String cSQL){ super(cTitulo,false,true,false,true); getContentPane().setLayout(null); setSize(500,250); setFrameIcon(new ImageIcon("imagenes/tmp.gif")); Componentes(cSQL); show(); } //METODOS void Centrar(int w,int h){ setLocation((w-getWidth())/2,(h-getHeight())/2); } private void Componentes(String cSQL){ panel.setBorder(new TitledBorder("")); panel.setBorder(new javax.swing.border.BevelBorder(1)); panel.setBounds(2, 2, getWidth()-10,getHeight()-64); VerDatos(cSQL); tabla.setShowHorizontalLines(false); JScrollPane scrTabla = new JScrollPane(tabla); scrTabla.setBounds(5,5,panel.getWidth()-10,panel.getHeight()-10); getContentPane().add(scrTabla); btnImprimir = new clsBoton("Imprimir","I","Impresión preliminar","print",5,getHeight()-62); btnCerrar = new clsBoton("Cerrar","C","Cerrar formulario","close",5+btnImprimir.getWidth(),getHeight()- 62); // Botones // Nuevo getContentPane().add(btnImprimir); btnImprimir.setVisible(true); btnImprimir.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ Imprimir(); } // Cerrar getContentPane().add(btnCerrar); btnCerrar.setVisible(true); btnCerrar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ dispose(); setVisible(false); } // Metodos private void VerDatos(String cSQL) { clsConexion con = new clsConexion(); Statement sm; ResultSet rs; cnt = new DefaultTableModel(); int i=0; try{ sm = con.cnx.createStatement(); rs = sm.executeQuery(cSQL); int nCampos = rs.getMetaData().getColumnCount(); for (i=1;i<=nCampos;i++){ </pre>			
Observaciones:			

15.4. NARRATIVA DE CLASE clsConsultaBase

EAGLEEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 2/2	
	Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Narrativa de Clase clsConsultaBase del Sistema de Control de Personal.		
NARRATIVA DE LA CLASE CLSCONSULTABASE DEL SISTEMA		
✓	Esta clase es una de las clases básicas que nos facilita gracias a la herencia a no tener código repetido y nos disminuye cualquier cantidad de programación.	
✓	Es clase es la que nos da a nuestro Panel contenedor con la tabla modelo solo con los botones de Imprimir y cerrar.	
✓	Después cada consulta que la herede deberá solo contener el Query del Reporte.	
Observaciones:		

15.5. DIAGRAMA DE CLASE clsMantenimientoBasico

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	Pág. 1/1 BIBLIOTECA CAMPUS VENAS
Descripción: Son las clases que dan la herencia para los Mantenimientos de Empresa y Persona.		
DIAGRAMA DE CLASE clsMantenimientoBasico		
 <pre> classDiagram class CLASE_clsMantenimientoBasico_java["CLASE clsMantenimientoBasico.java"] class CLASE_clsMantenimientoTipo3_java["CLASE clsMantenimientoTipo3.java"] class EMPRESA class PERSONA CLASE_clsMantenimientoTipo3_java --> CLASE_clsMantenimientoBasico_java EMPRESA --> CLASE_clsMantenimientoTipo3_java PERSONA --> CLASE_clsMantenimientoTipo3_java </pre>		
Observaciones:		

15.6. CLASE clsMantenimientoBasico

EAGLEYES		Pág. 1/2
Sistema de Control de Personal	Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
VERSIÓN 1.0		
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Clase clsMantenimientoBasico		
<pre> abstract class clsMantenimientoBasico extends JFrame{ // Propiedades private boolean lEditando = false; private boolean lNuevo = false; // Botones JPanel pnlBotones = new JPanel(null,true); private clsBoton btnNuevo = new clsBoton("Nuevo","N","Nuevo registro","new",0,0); private clsBoton btnGrabar = new clsBoton("Grabar","G","Guardar registro","save",0,0); private clsBoton btnModificar = new clsBoton("Modificar","M","Modificar registro","edit24",0,0); private clsBoton btnCancelar = new clsBoton("Cancelar","C","Cancelar edición","undo",0,0); private clsBoton btnCerrar = new clsBoton("Cerrar","C","Cerrar formulario","close",0,0); clsMantenimientoBasico(String cTitulo,int w,int h,String icono){ super(cTitulo,false,true,false,true); setSize(w,h); getContentPane().setLayout(null); if (icono.equals("")) setFrameIcon(new ImageIcon("imagenes/icono06.gif")); else setFrameIcon(new ImageIcon("imagenes/" + icono + ".gif")); PonerBotones(); show(); } void Centrar(int w,int h){ setLocation((w-getWidth())/2,(h-getHeight())/2);} private void PonerBotones(){ pnlBotones.setBounds(0,(this.getHeight() - btnNuevo.getHeight() 40),(btnNuevo.getWidth()*3+20),(btnNuevo.getHeight())); getContentPane().add(pnlBotones); // Nuevo btnNuevo.setLocation(10,0); pnlBotones.add(btnNuevo); btnNuevo.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ lNuevo=true; lEditando = true; HabilitarBotones(); Nuevo();}); // Grabar btnGrabar.setLocation(btnNuevo.getX(),btnNuevo.getY()); pnlBotones.add(btnGrabar); btnGrabar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){lEditando = false; HabilitarBotones(); Grabar();}); // Modificar btnModificar.setLocation(btnNuevo.getX() + btnNuevo.getWidth(),btnNuevo.getY()); pnlBotones.add(btnModificar); btnModificar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ lNuevo=false; lEditando = true; HabilitarBotones(); Modificar();}); // Cancelar btnCancelar.setLocation(btnModificar.getX(),btnModificar.getY()); pnlBotones.add(btnCancelar); btnCancelar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){lEditando = false; HabilitarBotones(); Cancelar();}); // Cerrar btnCerrar.setLocation(btnModificar.getX() + btnModificar.getWidth(),btnModificar.getY()); pnlBotones.add(btnCerrar); btnCerrar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ setVisible(false);dispose();}); </pre>		
Observaciones:		

15.7. CLASE clsMantenimientoBasico

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/2 Fecha Diseño: 2/08/2004		Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		BIBLIOTEC CAMPUS PENAS	
Descripción: Clase clsMantenimientoBasico					
<pre> // METODOS // Visualizacion de botones void HabilitarBotones(){ btnGrabar.setVisible(!Editando); btnCancelar.setVisible(!Editando); btnNuevo.setVisible(!Editando); btnModificar.setVisible(!Editando); btnCerrar.setEnabled(!Editando);} abstract void Nuevo(); abstract void Modificar(); abstract void Grabar(); abstract void Cancelar(); // SETs void setEsnuevo(boolean estado){ setEsnuevo(estado);} void setEditando(boolean estado){ setEditando(estado);} // GETs boolean getEsnuevo(){ return this.lNuevo;} boolean getEditando(){ return this.lEditando;} } </pre>					
Observaciones:					

15.8. NARRATIVA CLASE clsMantenimientoBasico

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 2/2	
		Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: <i>Narrativa de Clase clsMantenimientoBasico del Sistema de Control de Personal.</i>			
NARRATIVA DE LA CLASE CLSMANTENIMIENTOBASICO DEL SISTEMA			
✓	Esta clase es una de las clases básicas que por herencia nos facilita la programación.		
✓	Es clase es la que nos da nuestro Panel contenedor con los botones Nuevo, Grabar, Modificar, Eliminar, Imprimir, Cerrar, Cancelar.		
✓	Y después todas las clases la heredan cuando necesiten los botones.		
Observaciones:			

15.9. CLASE clsMantenimientoTipo3

EAGLEYES		Pág. 1/3	
Sistema de Control de Personal		Fecha Diseño:	Fecha Última Actualización:
VERSIÓN 1.0		2/08/2004	20/08/2004
Para:	Autores:		
PROTCOM – ESPOL		Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Clase clsMantenimientoTipo3.java			
<pre>abstract class clsMantenimientoTipo3 extends clsMantenimientoBasico{ // Etiquetas clsLabel lblCodigo = new clsLabel("Código:",10,10,80,20); clsLabel lblCedRuc = new clsLabel("Cédula o Ruc:",220,10,80,20); clsLabel lblNombre = new clsLabel("Nombre:",10,32,80,20); clsLabel lblDireccion = new clsLabel("Dirección:",10,54,80,20); clsLabel lblTelefono = new clsLabel("Teléfono:",10,76,80,20); clsLabel lblEmail = new clsLabel("e-mail:",10,98,80,20); clsEstado cntEstado = new clsEstado(10,120); // Campos clsTextField txtCodigo = new clsTextField(90,10,50,20); clsExaminar btnExaminarCod = new clsExaminar(145,10); clsTextField txtCedRuc = new clsTextField(260,10,80,20); clsExaminar btnExaminarCed = new clsExaminar(345,10); clsTextField txtNombre = new clsTextField(90,32,250,20); clsTextField txtDireccion = new clsTextField(90,54,400,20); clsTextField txtTelefono = new clsTextField(90,76,100,20); clsTextField txtEmail = new clsTextField(90,98,400,20); clsMantenimientoTipo3(){ super("Mantenimiento Basico Tipo 3",500,350,""); Componentes(); } clsMantenimientoTipo3(String titulo){ super(titulo,350,170,""); Componentes(); } clsMantenimientoTipo3(String titulo,int w,int h,String icono){ super(titulo,w,h,icono); Componentes(); } private void Componentes(){ // Etiquetas getContentPane().add(lblCodigo); getContentPane().add(lblCedRuc); getContentPane().add(lblNombre); getContentPane().add(lblDireccion); getContentPane().add(lblTelefono); getContentPane().add(lblEmail); getContentPane().add(cntEstado); // Campos txtCodigo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) if (txtCodigo.getText().equals("")) Examinar(); else Consultar(Integer.parseInt(txtCodigo.getText())); transferFocus(); } }); } }</pre>			
Observaciones:			



15.10. CLASE clsMantenimientoTipo3

EAGLEYES		Pág. 2/3	
Sistema de Control de Personal		Fecha Diseño:	Fecha Última
VERSIÓN 1.0		2/08/2004	Actualización:
		20/08/2004	
Para:		Autores:	
PROTCOM – ESPOL		Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Clase clsMantenimientoTipo3.java			
<pre>getContentPane().add(txtCodigo); btnExaminarCod.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ if (txtCodigo.getText().equals("")) Examinar(); else Consultar(Integer.parseInt(txtCodigo.getText())); transferFocus();}); getContentPane().add(btnExaminarCod); txtCedRuc.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) if (txtCedRuc.getText().equals("")) Examinar(); else Consultar(Integer.parseInt(txtCedRuc.getText())); transferFocus();}); getContentPane().add(txtCedRuc); btnExaminarCed.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ if (txtCedRuc.getText().equals("")) Examinar(); else Consultar(Integer.parseInt(txtCedRuc.getText())); transferFocus();}); getContentPane().add(btnExaminarCed); txtNombre.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();}); getContentPane().add(txtNombre); txtDireccion.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();}); getContentPane().add(txtDireccion); txtTelefono.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();}); getContentPane().add(txtTelefono); txtEmail.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();}); getContentPane().add(txtEmail);}</pre>			
Observaciones:			

15.11. CLASE clsMantenimientoTipo3

EAGLEYES		Pág. 1/2	
Sistema de Control de Personal		Fecha Diseño:	Fecha Última
VERSIÓN 1.0		2/08/2004	Actualización:
		20/08/2004	
Para:		Autores:	
PROTCOM – ESPOL		Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Clase clsMantenimientoTipo3.java			
<pre>// Metodos void Nuevo(){ LimpiarEntrada(); HabilitarEntrada(); } void Modificar(){ HabilitarEntrada(); } void Grabar(){ HabilitarEntrada(); } void Cancelar(){ LimpiarEntrada(); HabilitarEntrada(); } void LimpiarEntrada(){ txtCodigo.setText(""); txtCedRuc.setText(""); txtNombre.setText(""); txtDireccion.setText(""); txtTelefono.setText(""); txtEmail.setText(""); cntEstado.setEstado(true); } void HabilitarEntrada(){ txtCodigo.Habilitar(!getEditando()); btnExaminarCod.setEnabled(!getEditando()); txtCedRuc.Habilitar(getEditando()); btnExaminarCed.setEnabled(getEditando()); txtNombre.Habilitar(getEditando()); txtDireccion.Habilitar(getEditando()); txtTelefono.Habilitar(getEditando()); txtEmail.Habilitar(getEditando()); cntEstado.Habilitar(getEditando() & !getEsnuevo()); } void Consultar(int lnCodigo){ JOptionPane.showMessageDialog(null,"Sin programación","Aviso",JOptionPane.INFORMATION_MESSAGE); }; void Examinar(){ JOptionPane.showMessageDialog(null,"Sin programación","Aviso",JOptionPane.INFORMATION_MESSAGE); }; // SETs void setEstado(boolean estado){ cntEstado.setEstado(estado); } // GETs boolean getEstado(){ return cntEstado.getEstado(); } }</pre>			
Observaciones:			

15.12. NARRATIVA DE CLASE clsMantenimientoTipo3

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004	Pág. 2/2
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Descripción: Narrativa de Clase clsMantenimientoTipo3del Sistema de Control de Personal.				
NARRATIVA DE LA CLASE CLSMANTENIMIENTOTIPO3 DEL SISTEMA				
✓	Esta clase es una de las clases básicas que por herencia nos falcita la programación.			
✓	Es clase es la que nos da nuestro Panel contenedor solo con los botones Nuevo, Grabar, Modificar, Eliminar, Imprimir, Cerrar, Cancelar que la hereda de la clase clsMantenimientoBasico.			
✓	Y además nos da el Panel contenedor con las etiquetas y cajas de texto que son los siguientes: Código, Nombre, cedula, dirección, teléfono, mail.			
	Y también nos da los botones de búsqueda de código, cedula y después todas las clases la heredan cuando necesiten los botones.			

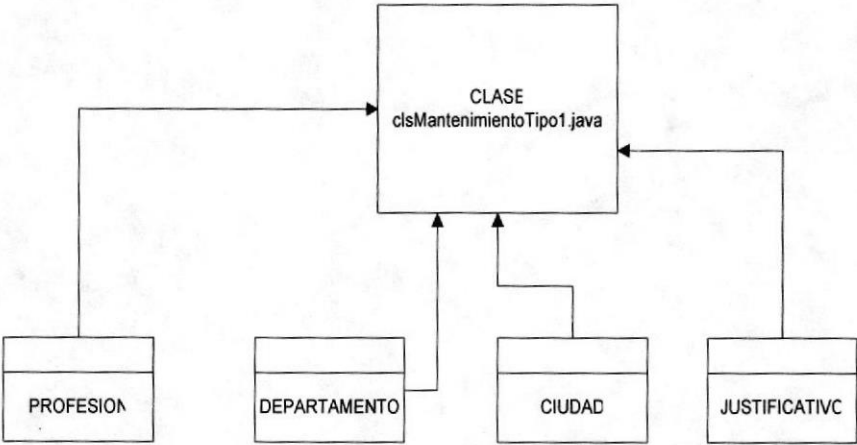


15.13.DIAGRAMA DE CLASE clsMantenimientoTipo1

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0	Pág. 1/1	
	Fecha Diseño: 2/08/2004	BIBLIOTECA CAMPUS PEÑAS
Para: PROTCOM – ESPOL	Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	

Descripción: Clase clsMantenimientoTipo1.java que da herencia a las tablas Profesión, Departamento, Ciudad, Justificativo.

DIAGRAMA DE CLASES CLSMANTENIMIENTOTIPO1



Observaciones:



15.14. CLASE clsMantenimientoTipo1

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/2 FECHA DE DISEÑO FECHA ÚLTIMA ACTUALIZACIÓN	
		Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Clase clsMantenimientoTipo1.java			
<pre>abstract class clsMantenimientoTipo1 extends clsMantenimientoBasico{ // Etiquetas clsLabel lblCodigo = new clsLabel("Código:",10,10,80,20); clsLabel lblDescripcion = new clsLabel("Descripción:",10,32,80,20); clsEstado cntEstado = new clsEstado(10,54); // Campos clsTextField txtCodigo = new clsTextField(90,10,50,20); clsExaminar btnExaminar = new clsExaminar(145,10); clsTextField txtDescripcion = new clsTextField(90,32,250,20); clsMantenimientoTipo1(){ super("Mantenimiento Basico Tipo 1",350,170,""); Componentes(); } clsMantenimientoTipo1(String titulo){ super(titulo,350,170,""); Componentes(); } clsMantenimientoTipo1(String titulo,int w,int h,String icono){ super(titulo,w,h,icono); Componentes(); } } private void Componentes(){ // Etiquetas getContentPane().add(lblCodigo); getContentPane().add(lblDescripcion); getContentPane().add(cntEstado); // Campos txtCodigo.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) if (txtCodigo.getText().equals("")) Examinar(); else Consultar(Integer.parseInt(txtCodigo.getText())); transferFocus(); } }); getContentPane().add(txtCodigo); btnExaminar.addActionListener(new ActionListener(){ public void actionPerformed(ActionEvent e){ if (txtCodigo.getText().equals("")) Examinar(); else Consultar(Integer.parseInt(txtCodigo.getText())); transferFocus(); } }); getContentPane().add(btnExaminar); txtDescripcion.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus(); } }); getContentPane().add(txtDescripcion); }</pre>			
Observaciones:			

15.15. CLASE clsMantenimientoTipo1

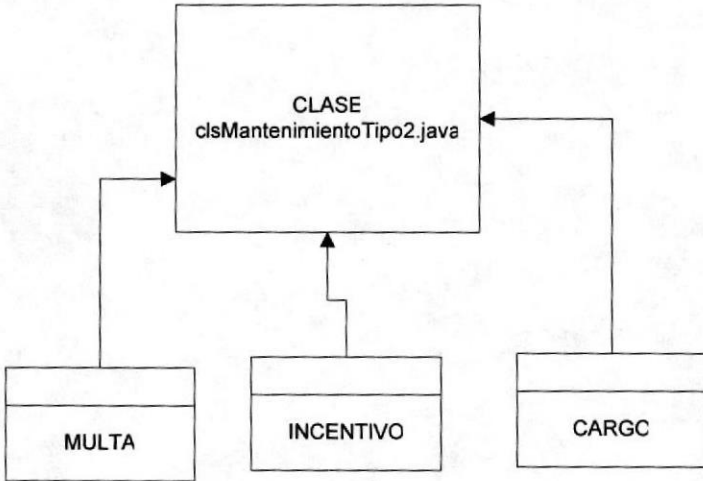
EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/2 FECHA DE ELABORACIÓN DEL DOCUMENTO FECHA DE ACTUALIZACIÓN DEL DOCUMENTO	
		Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Clase clsMantenimientoTipo1.java			
// Metodos void Nuevo(){ LimpiarEntrada(); HabilitarEntrada(); } void Modificar(){ HabilitarEntrada(); } void Grabar(){ HabilitarEntrada(); } void Cancelar(){ LimpiarEntrada(); HabilitarEntrada(); } void LimpiarEntrada(){ txtCodigo.setText(""); txtDescripcion.setText(""); cntEstado.setEstado(true); } void HabilitarEntrada(){ txtCodigo.Habilitar(!getEditando()); btnExaminar.setEnabled(!getEditando()); txtDescripcion.Habilitar(getEditando()); cntEstado.Habilitar(getEditando() & !getEsnuevo()); } void Consultar(int lnCodigo){ JOptionPane.showMessageDialog(null,"Sin programación","Aviso",JOptionPane.INFORMATION_MESSAGE); }; void Examinar(){ JOptionPane.showMessageDialog(null,"Sin programación","Aviso",JOptionPane.INFORMATION_MESSAGE); }; // SETs void setEstado(boolean estado){ cntEstado.setEstado(estado); } // GETs boolean getEstado(){ return cntEstado.getEstado(); }}			
Observaciones:			

15.16. NARRATIVA DE CLASE clsMantenimientoTipo1



EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/2	
		Fecha Diseño: 2/08/2004	Fecha Última Actualización: 20/08/2004
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Narrativa de Clase clsMantenimientoTipo1 del Sistema de Control de Personal.			
NARRATIVA DE LA CLASE CLSMANTENIMIENTOTIPO1 DEL SISTEMA			
✓	Esta clase es una de las clases básicas que por herencia nos facilita la programación.		
✓	Es clase es la que nos da nuestro Panel contenedor solo con los botones Nuevo, Grabar, Modificar, Eliminar, Imprimir, Cerrar, Cancelar que la hereda de la clase clsMantenimientoBasico.		
✓	Y además nos da el Panel contenedor con las etiquetas y cajas de texto que son los siguientes: Código, Descripción, Estado		
✓	Y también nos da al botón de búsqueda de código.		
Observaciones:			

15.17.DIAGRAMA DE CLASE clsMantenimientoTipo2

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1	
		Fecha Diseño: 2/08/2004	Fecha/Actualización: 20/08/2004
Para:	Autores:		
PROTCOM – ESPOL	Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Descripción: Clase clsMantenimientoTipo2.java que da herencia a Multa, Incentivo y Cargo.			
DIAGRAMA DE CLASES CLSMANTENIMIENTOTIPO2			
 <pre>classDiagram class CLASE_clsMantenimientoTipo2_java["CLASE clsMantenimientoTipo2.java"] class MULTA class INCENTIVO class CARGO MULTA -- > CLASE_clsMantenimientoTipo2_java INCENTIVO -- > CLASE_clsMantenimientoTipo2_java CARGO -- > CLASE_clsMantenimientoTipo2_java</pre>			
Observaciones:			



15.18. CLASE clsMantenimientoTipo2

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/1 Fecha Diseño: 20/08/2004		Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL			Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo		
Descripción: Clase clsMantenimientoTipo2.java					
<pre> abstract class clsMantenimientoTipo2 extends clsMantenimientoTipo1 { // Etiquetas clsLabel lblMonto = new clsLabel("Valor:",10,54,80,20); // Campos clsTextField txtMonto = new clsTextField(90,54,80,20); clsMantenimientoTipo2(){ super("Mantenimiento Basico Tipo 2",350,170,""); Componentes(); } clsMantenimientoTipo2(String titulo,int w,int h,String icono){ super(titulo,w,h,icono); Componentes(); } private void Componentes(){ getContentPane().add(lblMonto); txtMonto.addKeyListener(new KeyAdapter(){ public void keyPressed(KeyEvent evt){ if(evt.getKeyCode() == KeyEvent.VK_ENTER) transferFocus();}); getContentPane().add(txtMonto); cntEstado.setLocation(cntEstado.getX(),cntEstado.getY()+22);} // Metodos void LimpiarEntrada(){ super.LimpiarEntrada(); txtMonto.setText("0.00"); } void HabilitarEntrada(){ super.HabilitarEntrada(); txtMonto.Habilitar(getEditando()); } void Consultar(int lnCodigo){ JOptionPane.showMessageDialog(null,"Sin programación","Aviso",JOptionPane.INFORMATION_MESSAGE); }; } </pre>					
Observaciones:					

15.19. NARRATIVA DE CLASE clsMantenimientoTipo2

EAGLEYES Sistema de Control de Personal VERSIÓN 1.0		Pág. 1/2 Fecha Diseño: 2/08/2004 Fecha/Actualización: 20/08/2004	
Para: PROTCOM – ESPOL		Autores: Carlos Quiroga Marcelo Pinto Wendy Naranjo	
Descripción: Narrativa de Clase clsMantenimientoTipo2 del Sistema de Control de Personal.			
NARRATIVA DE LA CLASE CLSMANTENIMIENTOTIPO2 DEL SISTEMA			
✓	Esta clase es una de las clases básicas que por herencia nos facilita la programación.		
✓	Esa clase es la que nos da nuestro Panel contenedor con solo las Etiquetas Nuevo, Grabar, Modificar, Eliminar, Imprimir, Cerrar, Cancelar que la hereda de la clase clsMantenimientoBasico.		
✓	Y además nos da el Panel contenedor con las etiquetas y cajas de texto que son los siguientes: Código, Descripción, Estado		
✓	Y también nos da al botón de búsqueda de código.		
Observaciones:			



CAPÍTULO 16

SEGURIDADES DEL SISTEMA

16. SEGURIDADES DEL SISTEMA

16.1. NIVELES DE SEGURIDAD DE ACCESO Y USO

La entrada a **EAGLEEYES** se realizará mediante el menú Archivo, al momento de elegir esta opción <Nueva Sesión> le pedirá la clave y el usuario con el que va a trabajar.

Además, una vez dentro del sistema tendrá acceso a determinadas opciones del Sistema, dependiendo de la función que vaya a cumplir cada usuario del Módulo.

16.2. SEGURIDADES DE DATOS

La persona encargada del Sistema se asegurará de respaldar la Base de Datos a diario.

16.2.1. RESPALDO DE LA BASE DE DATOS

Ítem a Evaluar	Explicación
Período en que se realiza el respaldo	Todos los días
Períodos en que se borrarán los datos	No se borran
Persona que lo realiza	Encargado del Sistema
Tipo de respaldo	Toda la Base
Tipo de dispositivo	Unidad de ZIP
Ubicación de respaldo	Habitación segura a prueba de fuego

16.2.2. RESTAURACIÓN

Este proceso se da cuando se ha perdido información y se desea recuperar información respaldada. El encargado de la Base de Datos es la única persona autorizada para realizar este proceso