



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

SISTEMA DE ADMINISTRACIÓN Y CONTROL DE RECURSOS

TECNOLÓGICOS PARA LA COOPERATIVA DE AHORRO Y

CRÉDITO "ATUNTAQUI" LTDA.

MANUAL TÉCNICO

AUTOR

WILSON ANÍBAL CÁRDENAS HERNÁNDEZ

DIRECTOR

ING. MARCELO JURADO

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN	6
1.1. DESCRIPCIÓN GENERAL DEL SISTEMA	6
2. TECNOLOGÍA	7
3. BASE DE DATOS	7
3.1. TABLAS	7
3.1.1. TABLA SARTE_GRUPO	10
3.1.2. TABLA SARTE_TRECURSO	10
3.1.3. TABLA SARTE_STRECURSO	11
3.1.4. TABLA SARTE_MARCA	11
3.1.5. TABLA SARTE_MODELO	11
3.1.6. TABLA SARTE_ESTADO	12
3.1.7. TABLA SARTE_UBICACION	12
3.1.8. TABLA SARTE_RECURSO	13
3.1.9. TABLA SARTE_DRECURSO	13
3.1.10. TABLA SARTE_TSOPORTE	14
3.1.11. TABLA SARTE_TECNICO	14
3.1.12. TABLA SARTE_ACCION	15
3.1.13. TABLA SARTE_SOPORTE	15
3.1.14. TABLA SARTE_DSOPORTE	16
3.1.15. TABLA SARTE_MOTIVO	16
3.1.16. TABLA SARTE_MOVIMIENTO	17
3.1.17. TABLA ADM_EMPRESA	18
3.1.18. TABLA ADM_OFICINA	18
3.1.19. TABLA ADM_USUARIO	18
3.1.20. TABLA PER_DEPARTAMENTO	19
3.1.21. TABLA ADM_CARGO	19
3.1.22. TABLA PER_CATEGORIA	19
3.1.23. TABLA PER_EMPLEADO	20
3.2. VISTAS	21
3.2.1. VISTA SARTE_RECURSOS	21
3.2.2. VISTA SARTE_GRUPOS	22
3.2.3. VISTA SARTE_EMPLEADOS	23
3.2.4. VISTA SARTE_MARCAS	23
3.2.5. VISTA SARTE_DETALLES	24
3.2.6. VISTA SARTE_DRECURSOS	25
3.2.7. VISTA SARTE_SOPORTES	26
3.2.8. VISTA SARTE_TRECURSOS	27
3.2.9. VISTA SARTE_MODELOS	27

3.2.10. VISTA SARTE_STRECURSOS	28
4. PROGRAMACION JAVA	29
<hr/>	
4.1. PAQUETE "SARTE.BEANS"	29
4.1.1. CLASE ACCION	29
4.1.2. CLASE DRECURSO	29
4.1.3. CLASE DSOPORTE	30
4.1.4. CLASE ESTADO	31
4.1.5. CLASE GRUPO	32
4.1.6. CLASE MARCA	33
4.1.7. CLASE MODELO	34
4.1.8. CLASE MOVIMIENTO	35
4.1.9. CLASE RECURSO	36
4.1.10. CLASE SOPORTE	38
4.1.11. CLASE STRECURSO	39
4.1.12. CLASE TRECURSO	40
4.1.13. CLASE TSOPORTE	41
4.1.14. CLASE UBICACION	42
4.1.15. CLASE USUARIO	43
4.2. PAQUETE "SARTE.DATOS"	44
4.2.1. CLASE CONEXIÓN	44
4.3. PAQUETE "SARTE.TAGS"	45
4.3.1. CLASE ADMGRUPOS	45
4.3.2. CLASE ADMMARCAS	45
4.3.3. CLASE ADMSTRECURSOS	45
4.3.4. CLASE ADMTRECURSOS	45
4.3.5. CLASE INGOFICINA	45
4.3.6. CLASE INGTRERCURSOS	46
4.3.7. CLASE LISAREAS	46
4.3.8. CLASE LISDRECURSOS	46
4.3.9. CLASE LISDSOPORTES	46
4.3.10. CLASE LISESTADOS	46
4.3.11. CLASE LISGRUPOS	47
4.3.12. CLASE LISMARCAS	47
4.3.13. CLASE LISMODELOS	47
4.3.14. CLASE LISMOVIMIENTOS	47
4.3.15. CLASE LISOPFIN	47
4.3.16. CLASE LISOPORTES	48
4.3.17. CLASE LISSTRECURSOS	48
4.3.18. CLASE LISTRECURSOS	48
4.3.19. CLASE LISTSOPORTES	48
4.3.20. CLASE REPDRECURSOS	48
4.3.21. CLASE REPGRUPOS	48
4.3.22. CLASE REPOFICINAS	49

4.3.23.	CLASE REPTRECURSOS	49
4.3.24.	CLASE RPOFICINA	49
4.3.25.	CLASE RPRECURSOS	49
4.3.26.	CLASE RPSOPFIN	49
4.3.27.	CLASE RPSOPORTES	50
4.3.28.	CLASE RPTRECURSOS	50
4.3.29.	CLASE SELACCION	50
4.3.30.	CLASE SELDETALLE	50
4.3.31.	CLASE SELEMPLLEADO	50
4.3.32.	CLASE SELESTADO	51
4.3.33.	CLASE SELGRUPO	51
4.3.34.	CLASE SELMARCA	51
4.3.35.	CLASE SELMODELO	51
4.3.36.	CLASE SELMOTIVO	51
4.3.37.	CLASE SELOFICINA	51
4.3.38.	CLASE SELSTRECURSO	52
4.3.39.	CLASE SELTRECURSO	52
4.3.40.	CLASE SELTSOPORTE	52
4.3.41.	CLASE SELUBICACION	52
4.3.42.	CLASE SOPDRECURSOS	52
4.3.43.	CLASE SOPMDRECURSOS	53
4.3.44.	CLASE SOPOFICINA	53
4.3.45.	CLASE TRSOFICINA	53
5.	JASPERREPORTS	54

5.1.1.	REPORTE DMOVIMIENTO	54
5.1.2.	REPORTE DMOVIMIENTO	54
5.1.3.	REPORTE DSOPORTE	54
5.1.4.	REPORTE INVCUSTODIO	54
5.1.5.	REPORTE INVGENERAL	55
5.1.6.	REPORTE INVGENERALR	55
5.1.7.	REPORTE INVGRUPO	55
5.1.8.	REPORTE INVGRUPOR	55
5.1.9.	REPORTE INVMODELO	56
5.1.10.	REPORTE INVOFICINA	56
5.1.11.	REPORTE INVOFICINAR	56
5.1.12.	REPORTE INVTRECURSO	56
5.1.13.	REPORTE INVTRECURSOR	56
5.1.14.	REPORTE MODELOS	57
5.1.15.	REPORTE RECMOVIMIENTO	57
5.1.16.	REPORTE RECSOPORTES	57
5.1.17.	REPORTE RECURSO	57
5.1.18.	REPORTE SOPORTE	57
5.1.19.	REPORTE SOPORTES	58

5.1.20. REPORTE SOPORTESF	58
6. PROGRAMACION WEB	59
<hr/>	
6.1.1. INGRESO AL SISTEMA	59
6.1.2. MÓDULO DE ADMINISTRACIÓN	59
6.1.3. MODULO DE RECURSOS	61
6.1.4. MODULO DE MANTENIMIENTOS	62
6.1.5. MODULO DE TRASLADOS	66
6.1.6. MODULO DE REPORTES	67

1. INTRODUCCIÓN

Después de haber realizado un estudio de los procesos involucrados en la administración y control de los recursos tecnológicos de la Cooperativa de Ahorro y Crédito Atuntaqui Ltda., se ha visto conveniente clasificados, en cinco módulos principales. Mismo que fueron optimizados para una mejor administración de tecnología.



Figura 1.1. Esquema general del sistema (SARTE).

1.1. DESCRIPCIÓN GENERAL DEL SISTEMA

A continuación se describe brevemente cada uno de los módulos del sistema:

Administración

El sistema está basado en su mayoría en parámetros establecidos por parte del administrador del sistema; mismos que pueden ser configurables a través del sistema de información.

Ingreso de datos

Una vez realizada la configuración de los parámetros, esenciales para el correcto funcionamiento del sistema, estos nos permite ingresar, recompilar y organizar la información de una variedad de recursos tecnológicos, conforme la categorización establecida por el administrador del sistema.

Mantenimiento

Uno de las necesidades principales y motivos por el cual se planteó la implementación del sistema de administración de recursos tecnológicos, fue el poder registrar y llevar un control de todos los mantenimientos y novedades suscitadas durante el tiempo de vida útil de los recursos tecnológicos.

Traslados

Durante el estudio del proceso de control de los recursos y el proceso de recopilación de la información, pudimos determinar que existe carencia de información en lo que respecta al traslado y custodios de equipos, dificultándose el proceso de control de los mismos. Es por esto que se planteó la implementación de un módulo de traslado de equipos que nos ayuden con esta tarea.

Reportes

La razón principal por la que se planteó la implementación de este sistema en la cooperativa es el poder disponer de información clara, precisa y oportuna de todas las novedades presentadas con los recursos tecnológicos, con el fin de conocer la realidad institucional en lo que respecta a la tecnología y poder tomar con prontitud las decisiones más acertadas dentro de la administración de estos recursos.

El módulo de reportes presenta una variedad de informes que pueden ser generados instantáneamente.

2. TECNOLOGÍA

La tecnología utilizada para el desarrollo del presente proyecto es la tecnología Java.

Las herramientas de desarrollo utilizadas para la implementación de este sistema fueron:

- El IDE de programación, NetBeans IDE 7
- El servidor de aplicaciones para Java, Tomcat 7
- El sistema de bases de datos, Adaptive Sybase Enterprise 12.5
- La librería de desarrollo JasperReports 3.5
- El programa para la creación de reportes iReport 4

3. BASE DE DATOS

Durante el proceso de implementación del Sistema de Administración y Control de los Recursos Tecnológicos se creó una variedad de objetos en el Sistema Gestor de Bases de Datos existente en la Cooperativa de Ahorro y Crédito Atuntaqui Ltda., específicamente en la base de datos "*atun_riesgos*"; la misma que está destinada para la creación de los objetos de los sistemas diseñados e implementados en y para la cooperativa. Estos objetos nos permitirán almacenar y manipular toda la información del sistema. Debido a que la cooperativa dispone de una variedad de sistemas diseñados internamente y enfocados a diferentes finalidades fue necesario diferenciar los objetos del sistema implementado del resto de sistemas, para identificar los objetos pertenecientes al Sistema de Administración de Recursos Tecnológicos se utilizó en la definición de estos el prefijo "*sarte_*".

A continuación se detalla la estructura y funcionalidad de todos y cada uno de los objetos implementados durante la elaboración del proyecto.

3.1. TABLAS

Las tablas creadas y destinadas a almacenar la información correspondiente al Sistema de Administración y Control de los Recursos Tecnológicos se resumen en el siguiente diagrama y posteriormente se detalla cada una de ellas.

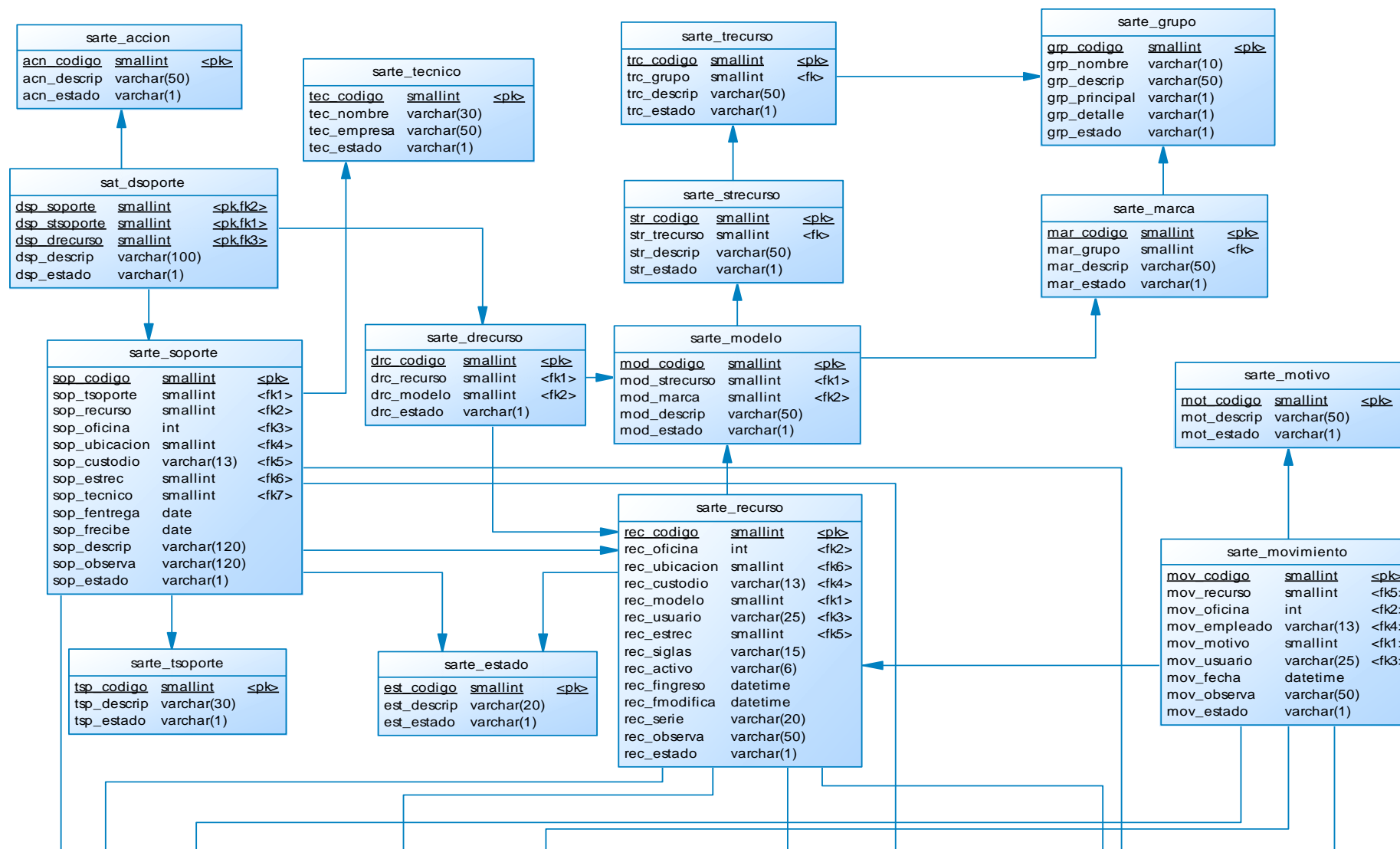


Figura 2.1. Diagrama entidad relación de la base (1ra parte).

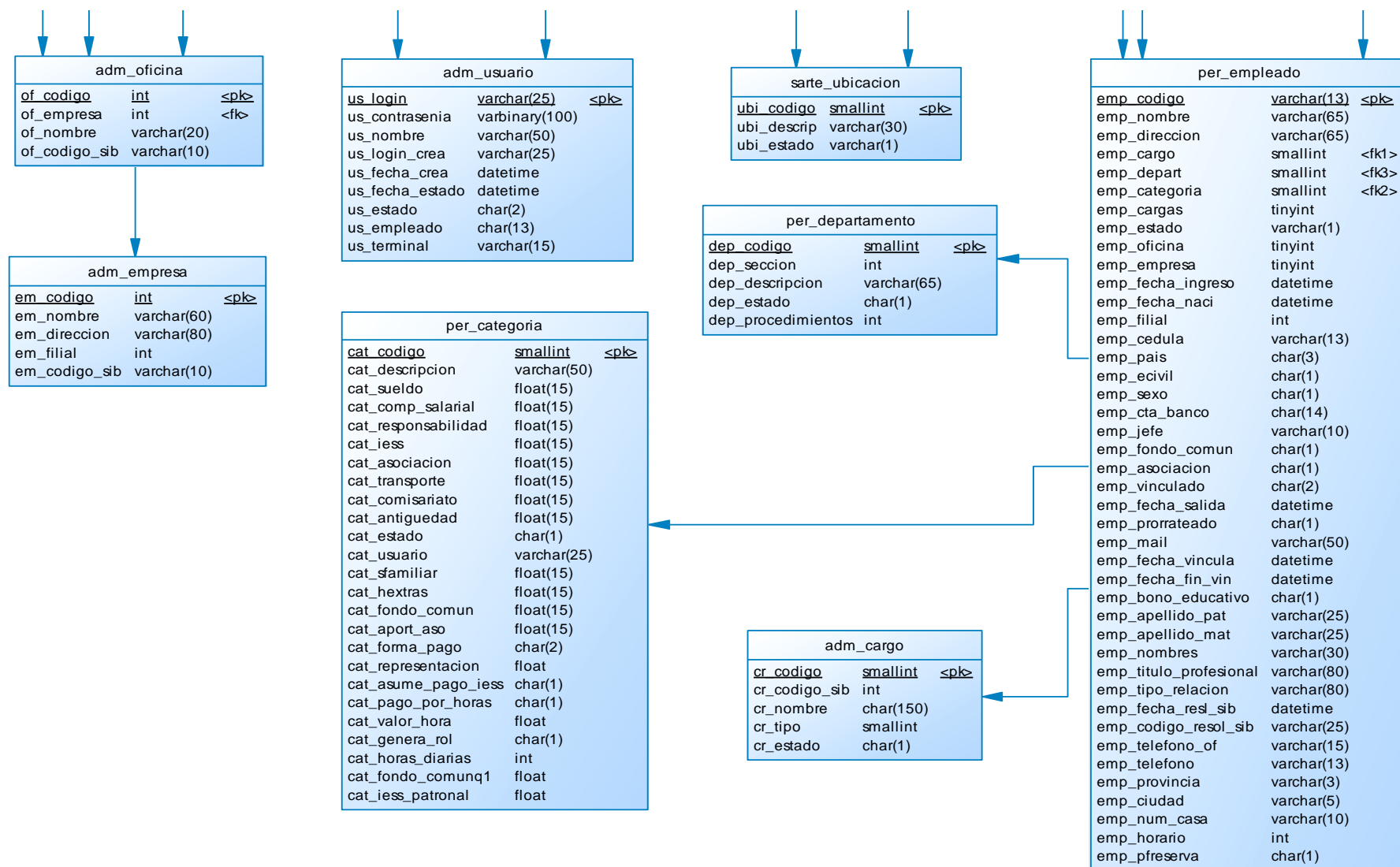


Figura 2.1. Diagrama entidad relación de la base (2da parte).

3.1.1. TABLA SARTE_GRUPO

Tabla destinada a la parametrización y agrupación de recursos tecnológicos.

CAMPO	TIPO	KEY	DESCRIPCION
grp_codigo	SMALLINT	PK	Identificador único de la tabla grupo.
grp_nombre	VARCHAR(10)		Nombre del grupo de recursos.
grp_descrip	VARCHAR(50)		Descripción larga del grupo de recursos.
grp_principal	VARCHAR(1)		Valor para determinar si el nivel del grupo de recursos es primario.
grp_detalle	VARCHAR(1)		Valor para determinar si el grupo de recursos describe las características de otro grupo de recursos.
grp_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.1. Estructura de la tabla sarte_grupo.

```
CREATE TABLE dbo.sarte_grupo(
    grp_codigo SMALLINT NOT NULL,
    grp_nombre VARCHAR (10) NULL,
    grp_descrip VARCHAR (50) NULL,
    grp_principal VARCHAR (1) NULL,
    grp_detalle VARCHAR (1) NULL,
    grp_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_grupo PRIMARY KEY (grp_codigo)
)
```

3.1.2. TABLA SARTE_TRECURSO

Permite realizar una clasificación por tipo de los recursos tecnológicos existentes en un grupo.

CAMPO	TIPO	KEY	DESCRIPCION
trc_codigo	SMALLINT	PK	Identificador único de la tabla tipo de recurso (categoría).
trc_grupo	SMALLINT	FK	Grupo al que pertenece la categoría de recursos.
trc_descrip	VARCHAR(50)		Descripción de la categoría de recursos.
trc_estado	VARCHAR(1)		Estado actual del registro.
trc_detalle	VARCHAR(1)		Parámetro para determinar si el tipo de recurso tiene características específicas

Tabla 3.2. Estructura de la tabla sarte_trecurso.

```
CREATE TABLE dbo.sarte_trecurso(
    trc_codigo SMALLINT NOT NULL,
    trc_grupo SMALLINT NULL,
    trc_descrip VARCHAR (50) NULL,
    trc_estado VARCHAR (1) NULL,
    trc_detalle VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_trecurso PRIMARY KEY (trc_codigo),
    CONSTRAINT fk_trecurso_grupo FOREIGN KEY (trc_grupo) REFERENCES dbo.sarte_grupo
    (grp_codigo)
)
```

3.1.3. TABLA SARTE_STRECURSO

Permite realizar una clasificación por subtipo de los recursos tecnológicos existentes en una categoría de tipos de recursos para una mejor organización.

CAMPO	TIPO	KEY	DESCRIPCION
str_codigo	SMALLINT	PK	Identificador único de la tabla subtipo de recurso (subcategoría).
str_trecurso	SMALLINT	FK	Categoría a la que pertenece la subcategoría de recursos.
str_descrip	VARCHAR(50)		Descripción de la subcategoría de recursos.
str_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.3. Estructura de la tabla sarte_strecurso.

```
CREATE TABLE dbo.sarte_strecurso(
    str_codigo SMALLINT NOT NULL,
    str_trecurso SMALLINT NULL,
    str_descrip VARCHAR (50) NULL,
    str_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_strecurso PRIMARY KEY (str_codigo),
    CONSTRAINT fk_strecurso_trecurso FOREIGN KEY (str_trecurso) REFERENCES
    dbo.sarte_trecurso (trc_codigo)
)
```

3.1.4. TABLA SARTE_MARCA

Almacena la información correspondiente a las marcas de los recursos, clasificadas de acuerdo a los grupos de recursos parametrizados en la tabla sarte_grupo.

CAMPO	TIPO	KEY	DESCRIPCION
mar_codigo	SMALLINT	PK	Identificador único de la tabla marca.
mar_grupo	SMALLINT	FK	Grupo al que pertenece la marca.
mar_descrip	VARCHAR(50)		Nombre de la marca.
mar_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.4. Estructura de la tabla sarte_marca.

```
CREATE TABLE dbo.sarte_marca(
    mar_codigo SMALLINT NOT NULL,
    mar_grupo SMALLINT NULL,
    mar_descrip VARCHAR (50) NULL,
    mar_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_marca PRIMARY KEY (mar_codigo),
    CONSTRAINT fk_marca_grupo FOREIGN KEY (mar_grupo) REFERENCES dbo.sarte_grupo
    (grp_codigo)
)
```

3.1.5. TABLA SARTE_MODELO

Almacena la información correspondiente a los modelos de los recursos tecnológicos, dependiendo del subtipo de recurso y la marca a la que pertenece.

CAMPO	TIPO	KEY	DESCRIPCION
mod_codigo	SMALLINT	PK	Identificador único de la tabla modelo.

mod_strecurso	SMALLINT	FK	Subcategoría de recursos al que pertenece el modelo.
mod_marca	SMALLINT	FK	Marca del modelo.
mod_descrip	VARCHAR(50)		Descripción del modelo.
mod_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.5. Estructura de la tabla sarte_modelo.

```
CREATE TABLE dbo.sarte_modelo(
    mod_codigo SMALLINT NOT NULL,
    mod_strecurso SMALLINT NULL,
    mod_marca SMALLINT NULL,
    mod_descrip VARCHAR (50) NULL,
    mod_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_modelo PRIMARY KEY (mod_codigo),
    CONSTRAINT fk modelo marca FOREIGN KEY (mod marca) REFERENCES dbo.sarte marca
    (mar_codigo),
    CONSTRAINT fk modelo strecurso FOREIGN KEY (mod strecurso) REFERENCES
    dbo.sarte_strecurso (str_codigo)
)
```

3.1.6. TABLA SARTE_ESTADO

Permite almacenar la parametrización de los posibles estados en los que puede encontrarse un recurso tecnológico.

CAMPO	TIPO	KEY	DESCRIPCION
est_codigo	SMALLINT	PK	Identificador único de la tabla estado.
est_descrip	VARCHAR(50)		Descripción del estado.
est_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.6. Estructura de la tabla sarte_estado.

```
CREATE TABLE dbo.sarte_estado(
    est_codigo SMALLINT NOT NULL,
    est_descrip VARCHAR (20) NULL,
    est_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_estado PRIMARY KEY (est_codigo)
)
```

3.1.7. TABLA SARTE_UBICACION

Guarda la información de las diferentes áreas o ubicaciones donde se puede encontrar un recurso, para su localización.

CAMPO	TIPO	KEY	DESCRIPCION
ubi_codigo	SMALLINT	PK	Identificador único de la tabla ubicación.
ubi_descrip	VARCHAR(50)		Descripción de la ubicación o área.
ubi_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.7. Estructura de la tabla sarte_ubicacion.

```
CREATE TABLE dbo.sarte_ubicacion(
    ubi_codigo SMALLINT NOT NULL,
    ubi_descrip VARCHAR (30) NULL,
    ubi_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_ubicacion PRIMARY KEY (ubi_codigo)
)
```

3.1.8. TABLA SARTE_RECURSO

Almacena la información correspondiente a los diferentes tipos de recursos tecnológicos, categorizándolos de acuerdo a la parametrización establecida en las tablas anteriores.

CAMPO	TIPO	KEY	DESCRIPCION
rec_codigo	SMALLINT	PK	Identificador único de la tabla recurso.
rec_oficina	INT	FK	Oficina donde se encuentra el recurso.
rec_ubicacion	SMALLINT	FK	Área donde se encuentra el recurso.
rec_custodio	VARCHAR(13)	FK	Custodio o responsable del recurso.
rec_modelo	SMALLINT	FK	Modelo del recurso tecnológico.
rec_usuario	VARCHAR(25)	FK	Usuario que ingreso el recurso tecnológico.
rec_estrec	SMALLINT	FK	Estado actual del recurso tecnológico.
rec_activo	VARCHAR(15)		Código de activo fijo, de no existir este campo se queda en NULL.
rec_fingreso	DATE		Fecha cuando se realizó el ingreso del recurso, esta fecha nunca cambia después del ingreso.
rec_fmodifica	DATE		Fecha de la última modificación del recurso.
rec_serie	VARCHAR(20)		Número de serie del recurso.
rec_observa	VARCHAR(50)		Breve observación referente al recurso.
rec_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.8. Estructura de la tabla sarte_recurso.

```
CREATE TABLE dbo.sarte_recurso(
    rec_codigo    SMALLINT NOT NULL,
    rec_oficina   INT NULL,
    rec_ubicacion SMALLINT NULL,
    rec_custodio  VARCHAR (13) NULL,
    rec_modelo    SMALLINT NULL,
    rec_usuario   VARCHAR (25) NULL,
    rec_estrec    SMALLINT NULL,
    rec_activo    VARCHAR (15) NULL,
    rec_fingreso  DATETIME NULL,
    rec_fmodifica DATETIME NULL,
    rec_serie     VARCHAR (20) NULL,
    rec_observa   VARCHAR (50) NULL,
    rec_estado    VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_recurso PRIMARY KEY (rec_codigo),
    CONSTRAINT fk_recurso_ubicacion FOREIGN KEY (rec ubicacion) REFERENCES
    dbo.sarte_ubicacion (ubi_codigo),
    CONSTRAINT fk_recurso_empleado FOREIGN KEY (rec_custodio) REFERENCES dbo.per_empleado
    (emp_codigo),
    CONSTRAINT fk_recurso_estado FOREIGN KEY (rec_estrec) REFERENCES dbo.sarte_estado
    (est_codigo),
    CONSTRAINT fk_recurso_modelo FOREIGN KEY (rec modelo) REFERENCES dbo.sarte modelo
    (mod_codigo),
    CONSTRAINT fk_recurso_oficina FOREIGN KEY (rec oficina) REFERENCES dbo.adm_oficina
    (of_codigo),
    CONSTRAINT fk_recurso_usuario FOREIGN KEY (rec_usuario) REFERENCES dbo.adm_usuario
    (us_login)
)
```

3.1.9. TABLA SARTE_DRECURSO

Permite detallar las características o componentes principales de los recursos tecnológicos.

CAMPO	TIPO	KEY	DESCRIPCION
drc_codigo	SMALLINT	PK	Identificador único de la tabla detalle del recurso.
drc_recurso	SMALLINT	FK	Código del recurso al que pertenece el detalle.
drc_modelo	SMALLINT	FK	Modelo del detalle del recurso.
drc_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.9. Estructura de la tabla sarte_drecurso.

```
CREATE TABLE dbo.sarte_drecurso(
    drc_codigo SMALLINT NOT NULL,
    drc_recurso SMALLINT NULL,
    drc_modelo SMALLINT NULL,
    drc_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_drecurso PRIMARY KEY (drc_codigo),
    CONSTRAINT fk_sarte_dr_fk_drecur_sarte_mo FOREIGN KEY (drc_modelo) REFERENCES
    dbo.sarte_modelo (mod_codigo),
    CONSTRAINT fk_drecurso_recurso FOREIGN KEY (drc_recurso) REFERENCES dbo.sarte_recurso
    (rec_codigo)
)
```

3.1.10. TABLA SARTE_TSOORTE

Registra los tipos de mantenimientos que se pueden realizar a los recursos tecnológicos.

CAMPO	TIPO	KEY	DESCRIPCION
tsp_codigo	SMALLINT	PK	Identificador único de la tabla tipo de soporte.
tsp_descrip	VARCHAR(50)		Descripción del tipo de soporte.
tsp_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.10. Estructura de la tabla sarte_tsoporte.

```
CREATE TABLE dbo.sarte_tsoporte(
    tsp_codigo SMALLINT NOT NULL,
    tsp_descrip VARCHAR (50) NULL,
    tsp_detalle VARCHAR (1) NULL,
    tsp_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_tsoporte PRIMARY KEY (tsp_codigo)
)
```

3.1.11. TABLA SARTE_TECNICO

Almacena los datos del técnico que realizó el mantenimiento de un recurso.

CAMPO	TIPO	KEY	DESCRIPCION
tec_codigo	SMALLINT	PK	Identificador único de la tabla técnico.
tec_nombre	VARCHAR(30)		Nombre del técnico.
tec_empresa	VARCHAR(50)		Nombre de la empresa del técnico.
tec_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.11. Estructura de la tabla sarte_tecnico.

```
CREATE TABLE dbo.sarte_tecnico(
    tec_codigo SMALLINT NOT NULL,
    tec_nombre VARCHAR (30) NULL,
    tec_empresa VARCHAR (50) NULL,
```

```
tec_estado VARCHAR (1) NULL,
CONSTRAINT pk_sarte_tecnico PRIMARY KEY (tec_codigo)
)
```

3.1.12. TABLA SARTE_ACCION

Almacena la parametrización de las tareas o acciones a realizarse durante el mantenimiento.

CAMPO	TIPO	KEY	DESCRIPCION
acn_codigo	SMALLINT	PK	Identificador único de la tabla acción.
acn_descrip	VARCHAR(20)		Descripción de la acción o tarea.
acn_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.12. Estructura de la tabla sarte_accion.

```
CREATE TABLE dbo.sarte_accion(
    acn_codigo SMALLINT NOT NULL,
    acn_descrip VARCHAR (20) NULL,
    acn_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_accion PRIMARY KEY (acn_codigo)
)
```

3.1.13. TABLA SARTE_SOPORTE

Tabla en la cual se registran todos los mantenimientos brindados a determinado recurso.

CAMPO	TIPO	KEY	DESCRIPCION
sop_codigo	SMALLINT	PK	Identificador único de la tabla soporte.
sop_tsoporte	SMALLINT	FK	Código perteneciente al tipo de mantenimiento.
sop_recurso	SMALLINT	FK	Recurso al que se realizó el mantenimiento.
sop_oficina	SMALLINT	FK	Oficina donde se encontraba el recurso al momento del mantenimiento.
sop_ubicacion	SMALLINT	FK	Ubicación o área donde se encontraba el recurso al momento del mantenimiento.
sop_custodio	VARCHAR(13)	FK	Custodio del recurso al momento de realizar el mantenimiento.
sop_estrec	SMALLINT	FK	Estado del recurso al momento de realizar el mantenimiento.
sop_tecnico	SMALLINT	FK	Técnico que brindo el mantenimiento.
sop_fentrega	DATE		Fecha de entrega del recurso para realizar el mantenimiento.
sop_frecibe	DATE		Fecha de recepción del recurso después de realizar el mantenimiento.
sop_descrip	VARCHAR(200)		Descripción del problema o inconveniente presentado con el recurso.
sop_observa	VARCHAR(200)		Detalle de lo realizado por parte del técnico durante el proceso de mantenimiento.
sop_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.13. Estructura de la tabla sarte_soporte.

```
CREATE TABLE dbo.sarte_soporte(
    sop_codigo SMALLINT NOT NULL,
```

```

sop_tsoporte SMALLINT NULL,
sop_recurso SMALLINT NULL,
sop_oficina SMALLINT NULL,
sop_ubicacion SMALLINT NULL,
sop_custodio VARCHAR (13) NULL,
sop_estrec SMALLINT NULL,
sop_fentrega DATE NULL,
sop_frecibe DATE NULL,
sop_descrip VARCHAR (200) NULL,
sop_observa VARCHAR (200) NULL,
sop_estado VARCHAR (1) NULL,
CONSTRAINT pk_sarte_soporte PRIMARY KEY (sop_codigo),
CONSTRAINT fk_soporte_recurso FOREIGN KEY (sop_recurso) REFERENCES dbo.sarte_recurso
(rec_codigo),
CONSTRAINT fk_soporte_tsoporte FOREIGN KEY (sop_tsoporte) REFERENCES
dbo.sarte_tsoporte (tsp_codigo)
)

```

3.1.14. TABLA SARTE_DSOPORTE

Permite detallar todas las acciones tomadas durante el proceso de mantenimiento.

CAMPO	TIPO	KEY	DESCRIPCION
dsp_soporte	SMALLINT	PK FK	Número de mantenimiento realizado.
dsp_drecurso	SMALLINT	PK FK	Características o componentes del recurso afectados durante el proceso de mantenimiento.
dsp_accion	SMALLINT	FK	Modelo del detalle del recurso.
dsp_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.14. Estructura de la tabla sarte_dsoporte.

```

CREATE TABLE dbo.sarte_dsoporte(
    dsp_codigo SMALLINT NOT NULL,
    dsp_soporte SMALLINT NOT NULL,
    dsp_accion SMALLINT NOT NULL,
    dsp_drecurso SMALLINT NOT NULL,
    dsp_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_dsoporte PRIMARY KEY (dsp_codigo),
    CONSTRAINT fk_dsoporte_stsoporte FOREIGN KEY (dsp_accion) REFERENCES dbo.sarte_accion
(acn_codigo),
    CONSTRAINT fk_dsoporte_drecurso FOREIGN KEY (dsp_drecurso) REFERENCES
dbo.sarte_drecurso (drc_codigo),
    CONSTRAINT fk dsoporte soporte FOREIGN KEY (dsp soporte) REFERENCES dbo.sarte soporte
(sop_codigo)
)

```

3.1.15. TABLA SARTE_MOTIVO

Registra los motivos por los cuales se puede realizar el traslado de un recurso.

CAMPO	TIPO	KEY	DESCRIPCION
mot_codigo	SMALLINT	PK	Identificador único de la tabla motivo.
mot_descrip	VARCHAR(50)		Descripción del motivo de los traslados.
mot_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.15. Estructura de la tabla sarte_motivo.

```

CREATE TABLE dbo.sarte_motivo(
    mot_codigo SMALLINT NOT NULL,

```



```

mot_descrip VARCHAR (50) NULL,
mot_estado VARCHAR (1) NULL,
CONSTRAINT pk_sarte_motivo PRIMARY KEY (mot_codigo)
)

```

3.1.16. TABLA SARTE_MOVIMIENTO

Almacena todos los movimientos realizados con los recursos tecnológicos para registrar sus custodios y ubicaciones tenidos durante la vida útil del recurso.

CAMPO	TIPO	KEY	DESCRIPCION
mov_codigo	SMALLINT	PK	Identificador único de la tabla movimiento.
mov_recurso	SMALLINT	FK	Código del recurso que cambio de ubicación y/o custodio.
mov_oficina	INT	FK	Oficina destino donde se trasladó el recurso.
mov_ubicacion	SMALLINT	FK	Área destino donde se ubicó el recurso.
mov_custodio	VARCHAR(13)	FK	Custodio nuevo al cual se asignó el recurso.
mov_motivo	SMALLINT	FK	Motivo por el cual se realizó el traslado.
mov_estrec	SMALLINT	FK	Estado del recurso al momento de realizar el traslado.
mov_usuario	VARCHAR(25)	FK	Usuario que registro el traslado del recurso.
mov_fecha	DATE		Fecha cuando se realizó el movimiento.
mov_observa	VARCHAR(50)		Almacena alguna observación sobre el movimiento.
mov_estado	VARCHAR(1)		Estado actual del registro.

Tabla 3.16. Estructura de la tabla sarte_movimiento.

```

CREATE TABLE dbo.sarte_movimiento(
    mov_codigo SMALLINT NOT NULL,
    mov_motivo SMALLINT NOT NULL,
    mov_usuario VARCHAR (25) NOT NULL,
    mov_recurso SMALLINT NOT NULL,
    mov_oficina INT NOT NULL,
    mov_ubicacion SMALLINT NOT NULL,
    mov_custodio VARCHAR (13) NOT NULL,
    mov_ofiant INT NULL,
    mov_ubiant SMALLINT NULL,
    mov_cusant VARCHAR (13) NULL,
    mov_fecha DATETIME NULL,
    mov_observa VARCHAR (50) NULL,
    mov_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_movimiento PRIMARY KEY (mov_codigo),
    CONSTRAINT fk_movimiento_empleado FOREIGN KEY (mov_custodio) REFERENCES
    dbo.per_empleado (emp_codigo),
    CONSTRAINT fk_movimiento_motivo FOREIGN KEY (mov motivo) REFERENCES dbo.sarte motivo
    (mot_codigo),
    CONSTRAINT fk_movimiento_oficina FOREIGN KEY (mov_oficina) REFERENCES dbo.adm_oficina
    (of_codigo),
    CONSTRAINT fk_movimiento_recurso FOREIGN KEY (mov_recurso) REFERENCES
    dbo.sarte_recurso (rec_codigo),
    CONSTRAINT fk_movimiento_ubicacion FOREIGN KEY (mov ubicacion) REFERENCES
    dbo.sarte_ubicacion (ubi_codigo),
    CONSTRAINT fk_movimiento_usuario FOREIGN KEY (mov_usuario) REFERENCES dbo.adm_usuario
    (us_login)
)

```

Además, para la implementación del presente proyecto fue necesario utilizar algunas tablas existentes en la base de datos "atun_riesgos", las mismas que se detallan a continuación:

3.1.17. TABLA ADM_EMPRESA

Tabla en donde se detalla los datos principales de la cooperativa.

CAMPO	TIPO	KEY	DESCRIPCION
em_codigo	INT	PK	Identificador único de la tabla empresa.
em_nombre	VARCHAR(60)		Nombre de la empresa.
em_direccion	VARCHAR (80)		Dirección de la empresa.
em_filial	INT		Filial a la que pertenece la empresa.
em_codigo_sib	VARCHAR (10)		Código de la empresa asignado por la Superintendencia de Bancos.

Tabla 3.17. Estructura de la tabla adm_empresa.

```
CREATE TABLE dbo.sarte_marca (
    mar_codigo SMALLINT NOT NULL,
    mar_grupo SMALLINT NULL,
    mar_descrip VARCHAR (50) NULL,
    mar_estado VARCHAR (1) NULL,
    CONSTRAINT pk_sarte_marca PRIMARY KEY (mar_codigo),
    CONSTRAINT fk_marca_grupo FOREIGN KEY (mar_grupo) REFERENCES dbo.sarte_grupo
    (grp_codigo)
)
```

3.1.18. TABLA ADM_OFICINA

Tabla en donde se encuentran los datos de las oficinas existentes de la cooperativa, para ubicar el lugar donde se encuentran los recursos.

CAMPO	TIPO	KEY	DESCRIPCION
of_codigo	INT	PK	Identificador único de la tabla oficina.
of_empresa	INT	FK	Código de la empresa a la cual pertenece la oficina.
of_nombre	VARCHAR (20)		Nombre de la oficina.
of_codigo_sib	VARCHAR (10)		Código de la oficina asignado por la SBS.

Tabla 3.18. Estructura de la tabla adm_oficina.

3.1.19. TABLA ADM_USUARIO

Tabla en la que se almacenan el login y la contraseña de los usuarios para ingresar a los diferentes sistemas de la cooperativa.

CAMPO	TIPO	KEY	DESCRIPCION
us_login	VARCHAR (25)	PK	Identificador único de la tabla usuario Login de acceso al sistema.
us_contrasenia	VARBINARY(100)		Contraseña cifrada de acceso al sistema.
us_nombre	VARCHAR (50)		Nombre del usuario.
us_login_crea	VARCHAR (25)		Login del usuario que ingreso el usuario.

us_fecha_crea	DATETIME	Fecha cuando se ingresó el usuario.
us_fecha_estado	DATETIME	Fecha de la última modificación del usuario.
us_estado	CHAR (2)	Estado actual del usuario.
us_empleado	CHAR (13)	Empleado al que pertenece el login.
us_terminal	VARCHAR (15)	Estación de trabajo a la que pertenece el usuario.

Tabla 3.19. Estructura de la tabla adm_usuario.

3.1.20. TABLA PER_DEPARTAMENTO

Especifica los datos de los departamentos existentes en la cooperativa.

CAMPO	TIPO	KEY	DESCRIPCION
dep_codigo	SMALLINT	PK	Identificador único de la tabla departamento, código del departamento.
dep_seccion	INT		Código de la sección a la que pertenece el departamento.
dep_descripcion	VARCHAR (65)		Descripción del departamento.
dep_estado	CHAR (1)		Estado actual del departamento.
dep_procedimientos	INT		Código de los procedimientos correspondientes al departamento.

Tabla 3.20. Estructura de la tabla per_departamento.

3.1.21. TABLA ADM_CARGO

Almacena la parametrización correspondiente a los cargos de los empleados de la cooperativa.

CAMPO	TIPO	KEY	DESCRIPCION
of_codigo	INT	PK	Identificador único de la tabla oficina.
of_empresa	INT	FK	Código de la empresa a la cual pertenece la oficina.
of_nombre	VARCHAR (20)		Nombre de la oficina.
of_codigo_sib	VARCHAR (10)		Código de la oficina asignado por la Superintendencia de Bancos.

Tabla 3.21. Estructura de la tabla adm_cargo.

3.1.22. TABLA PER_CATEGORIA

Tabla donde se ingresan las categorías de los empleados y se establecen los parámetros correspondientes para la gestión de las mismas.

CAMPO	TIPO	KEY	DESCRIPCION
cat_codigo	SMALLINT	PK	Identificador único de la tabla categoría.
cat_descripcion	VARCHAR (50)		Descripción o nombre de la categoría.
cat_sueldo	FLOAT		Valor de la remuneración mensual correspondiente a la categoría.
cat_responsabilidad	FLOAT		Cantidad asignada por responsabilidad.
cat_ies	FLOAT		Porcentaje de aportación al IESS.
cat_asociacion	FLOAT		Identifica las categorías que pertenecen a la asociación de empleados.

cat_estado	CHAR (1)	Estado actual de la categoría.
cat_usuario	VARCHAR (25)	Usuario que ingreso la categoría.
cat_fondo_comun	FLOAT	Valor debitado para el fondo común.
cat_forma_pago	CHAR (2)	Indica si la forma de pago es Quincenal o Mensual.
cat_representacion	FLOAT	Valor asignado por representación.
cat_asume_pago_ies	CHAR (1)	Indica si el patrono asume el pago al IESS.
cat_valor_hora	FLOAT	Indica el valor de la hora extra.
cat_genera_rol	CHAR (1)	Indica si la categoría general rol de pagos.
cat_horas_diarias	INT	Indica el número de horas diarias a laborar por parte del empleado.
cat_fondo_comunq1	FLOAT	Valor debitado para el fondo común en la primera quincena.
cat_ies_patronal	FLOAT	Porcentaje del IESS correspondiente al empleador.

Tabla 3.22. Estructura de la tabla per_categoria.

3.1.23. TABLA PER_EMPLEADO

Tabla donde se registran todos los datos correspondientes a los empleados de la cooperativa, y utilizados para asignar los custodios responsables de los recursos.

CAMPO	TIPO	KEY	DESCRIPCION
emp_codigo	VARCHAR (13)	PK	Identificador único de la tabla empleado.
emp_nombre	VARCHAR (65)		Nombre completo del empleado.
emp_direccion	VARCHAR (65)		Dirección domiciliaria del empleado.
emp_cargo	SMALLINT	FK	Cargo actual del empleado.
emp_depart	SMALLINT	FK	Departamento actual del empleado.
emp_categoria	SMALLINT	FK	Categoría actual del empleado.
emp_cargas	TINYINT		Número de cargar familiares.
emp_estado	VARCHAR (1)		Estado actual del empleado.
emp_oficina	TINYINT		Oficina a la que pertenece el empleado.
emp_empresa	TINYINT		Empresa a la que pertenece el empleado.
emp_fecha_ingreso	DATETIME		Fecha de ingreso a la cooperativa.
emp_fecha_naci	DATETIME		Fecha de nacimiento del empleado.
emp_filial	INT		Filial a la que pertenece el empleado.
emp_cedula	VARCHAR (13)		Número de cédula de identidad.
emp_pais	CHAR (3)		Nacionalidad del empleado.
emp_ecivil	CHAR (1)		Estado civil actual del empleado.
empsexo	CHAR (1)		Genero del empleado.
emp_cta_banco	CHAR (14)		Número de cuenta del empleado.
emp_jefe	VARCHAR (10)		Código del jefe inmediato.
emp_fondo_comun	CHAR (1)		Indica si aporta a no al fondo común.
emp_asociacion	CHAR (1)		Indica si es o no parte de la asociación de empleados.
emp_vinculado	CHAR (2)		Indica si el empleado es vinculado.
emp_fecha_salida	DATETIME		Fecha de salida de la institución.
emp_mail	VARCHAR (50)		Correo electrónico del empleado.
emp_fecha_vincula	DATETIME		Fecha de vinculación del empleado.
emp_fecha_fin_vin	DATETIME		Fecha de finalización de la vinculación.
emp_apellido_pat	VARCHAR (25)		Apellido paterno del empleado.
emp_apellido_mat	VARCHAR (25)		Apellido materno del empleado.

emp_nombres	VARCHAR (30)	Nombres del empleado.
emp_titulo_profesional	VARCHAR (80)	Título profesional que tiene el empleado.
emp_telefono_of	VARCHAR (15)	Teléfono de la oficina.
emp_telefono	VARCHAR (13)	Teléfono personal.
emp_provincia	VARCHAR (3)	Provincia donde vive el empleado.
emp_ciudad	VARCHAR (5)	Ciudad donde vive el empleado.
emp_num_casa	VARCHAR (10)	Número de la casa donde vive.
emp_horario	INT	Horario asignado al empleado.
emp_pfreserva	CHAR (1)	Indica si se le deposita o no los fondos de reserva en la cuenta.
emp_imp_renta	INT	Indica si paga o no el impuesto a la renta.

Tabla 3.23. Estructura de la tabla per_empleado.

3.2. VISTAS

Las vistas creadas y destinadas a visualizar la información detallada del Sistema de Administración y Control de los Recursos Tecnológicos se resumen en el siguiente diagrama y posteriormente se detalla cada una de ellas.

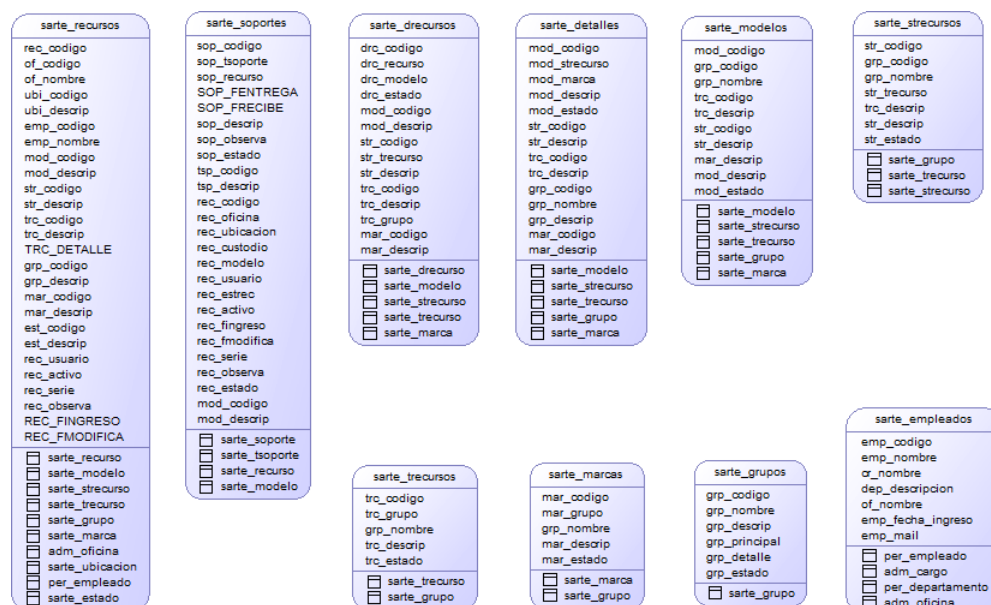


Figura 3.24. Diagrama vistas implementadas en el proyecto.

3.2.1. VISTA SARTE_RECURSOS

Vista generada a partir de las tablas "sarte_recurso", "sarte_modelo", "sarte_strecurso", "sarte_trecurso", "sarte_grupo", "sarte_marca", "adm_oficina", "sarte_ubicacion", "per_empleado" y "sarte_estado" para visualizar en una sola consulta todos los datos correspondientes a los recursos tecnológicos activos del sistema y generar los informes.

CAMPO	TABLA
rec_codigo	sarte_recurso
rec_usuario	sarte_recurso
rec_activo	sarte_recurso
rec_serie	sarte_recurso
rec_fingreso	sarte_recurso

rec_fmodifica	sarte_recurso
rec_observa	sarte_recurso
of_codigo	adm_oficina
of_nombre	adm_oficina
ubi_codigo	sarte_ubicacion
ubi_descrip	sarte_ubicacion
emp_codigo	per_empleado
emp_nombre	per_empleado
grp_codigo	sarte_grupo
grp_descrip	sarte_grupo
trc_codigo	sarte_trecurso
trc_descrip	sarte_trecurso
trc_detalle	sarte_trecurso
str_codigo	sarte_strecurso
str_descrip	sarte_strecurso
mod_codigo	sarte_modelo
mod_descrip	sarte_modelo
mar_codigo	sarte_marca
mar_descrip	sarte_marca
est_codigo	sarte_estado
est_descrip	sarte_estado

Tabla 3.25. Estructura lógica de la vista sarte_recursos.

```

CREATE VIEW sarte_recursos AS
SELECT rec_codigo, of_codigo, of_nombre, ubi_codigo,
       ubi_descrip, emp_codigo, emp_nombre, cr_nombre,
       mod_codigo, mod_descrip, str_codigo, str_descrip,
       trc_codigo, trc_descrip, trc_detalle, grp_codigo,
       grp_nombre, grp_descrip, mar_codigo, mar_descrip, est_codigo,
       est_descrip, rec_usuario, rec_activo, rec_serie,
       rec_observa, trc_estado, str_estado, rec_estado,
       CONVERT(VARCHAR(10),rec_fingreso,101) rec_fingreso,
       CONVERT(VARCHAR(10),rec_fmodifica,101) rec_fmodifica
FROM sarte_recurso, sarte_modelo, sarte_strecurso,
     sarte_trecurso, sarte_grupo, sarte_marca, adm_oficina,
     sarte_ubicacion, sarte_estado, per_empleado, adm_cargo
WHERE rec_modelo = mod_codigo
   AND mod_strecurso = str_codigo
   AND str_trecurso = trc_codigo
   AND trc_grupo = grp_codigo
   AND mod_marca = mar_codigo
   AND rec_oficina = of_codigo
   AND rec_ubicacion = ubi_codigo
   AND rec_estrec = est_codigo
   AND rec_custodio = emp_codigo
   AND emp_cargo = cr_codigo
GO

```

3.2.2. VISTA SARTE_GRUPOS

Muestra un listado de todos los grupos de recursos activos dentro del sistema, para indicar con que grupos de recursos se debe interactuar.

CAMPO	TABLA
grp_codigo	sarte_grupo
grp_nombre	sarte_grupo
grp_descrip	sarte_grupo
grp_principal	sarte_grupo
grp_detalle	sarte_grupo
grp_estado	sarte_grupo

Tabla 3.26. Estructura lógica de la vista sarte_grupos.

```
CREATE VIEW sarte_grupos AS
SELECT sarte_grupo.grp_codigo,
       sarte_grupo.grp_nombre,
       sarte_grupo.grp_descrip,
       sarte_grupo.grp_principal,
       sarte_grupo.grp_detalle,
       sarte_grupo.grp_estado
FROM sarte_grupo
WHERE grp_estado = 'A'
GO
```

3.2.3. VISTA SARTE_EMPLEADOS

Vista generada a partir de las tablas "*per_empleado*", "*adm_cargo*", "*per_departamento*", "*adm_oficina*" del sistema de personal; con el fin de visualizar la información más importante de los empleados de la Cooperativa y requerida por el sistema SARTE.

CAMPO	TABLA
emp_codigo	per_empleado
emp_nombre	per_empleado
emp_cargo	adm_cargo
emp_depart	per_departamento
emp_oficina	adm_oficina
emp_fingreso	per_empleado
emp_mail	per_empleado

Tabla 3.27. Estructura lógica de la vista sarte_empleados.

```
CREATE VIEW sarte_empleados AS
SELECT emp_codigo, emp_nombre, cr_nombre emp_cargo, dep_descripcion emp_depart, of_nombre
emp_oficina, emp_fecha_ingreso emp_fingreso, emp_mail
FROM per_empleado, adm_cargo, per_departamento, adm_oficina
WHERE emp_cargo = cr_codigo
AND emp_depart = dep_codigo
AND emp_oficina = of_codigo
AND emp_estado = 'V'
AND emp_codigo <> '1'
AND emp_categoria <> 18 -- 18:Asesores Externos - 13:Consejo Administracion
GO
```

3.2.4. VISTA SARTE_MARCAS

Vista generada a partir de las tablas "*sarte_marca*" y "*sarte_grupo*" para visualizar todas las marcas disponibles en el sistema y clasificadas por grupo de recursos.

CAMPO	TABLA
mar_codigo	sarte_marca
mar_grupo	sarte_marca
grp_nombre	sarte_grupo
mar_descrip	sarte_marca
mar_estado	sarte_marca

Tabla 3.28. Estructura lógica de la vista sarte_marcas.

```
CREATE VIEW sarte_marcas AS
SELECT sarte_marca.mar_codigo,
       sarte_marca.mar_grupo,
       sarte_grupo.grp_nombre,
       sarte_marca.mar_descrip,
       sarte_marca.mar_estado
FROM sarte_marca, sarte_grupo
WHERE sarte_grupo.grp_codigo = sarte_marca.mar_grupo
      AND sarte_marca.mar_estado = 'A'
GO
```

3.2.5. VISTA SARTE_DETALLES

Vista generada a partir de las tablas "sarte_modelo", "sarte_strecurso", "sarte_trecurso", "sarte_grupo" y "sarte_marca" para mostrar un listado de todas las características de los recursos parametrizadas y que pueden ser asignadas a un recurso.

CAMPO	TABLA
mod_codigo	sarte_modelo
mod_strecurso	sarte_modelo
mod_marca	sarte_modelo
mod_descrip	sarte_modelo
mod_estado	sarte_modelo
str_codigo	sarte_strecurso
str_descrip	sarte_strecurso
trc_codigo	sarte_trecurso
trc_descrip	sarte_trecurso
grp_codigo	sarte_grupo
grp_nombre	sarte_grupo
grp_descrip	sarte_grupo
mar_codigo	sarte_marca
mar_descrip	sarte_marca

Tabla 3.29. Estructura lógica de la vista sarte_detalles.

```
CREATE VIEW sarte_detalles AS
SELECT sarte_modelo.mod_codigo,
       sarte_modelo.mod_strecurso,
       sarte_modelo.mod_marca,
       sarte_modelo.mod_descrip,
       sarte_modelo.mod_estado,
       sarte_strecurso.str_codigo,
       sarte_strecurso.str_descrip,
       sarte_trecurso.trc_codigo,
       sarte_trecurso.trc_descrip,
       sarte_grupo.grp_codigo,
       sarte_grupo.grp_nombre,
```



```

    sarte_grupo.grp_descrip,
    sarte_marca.mar_codigo,
    sarte_marca.mar_descrip
FROM sarte_modelo, sarte_strecurso, sarte_trecurso, sarte_grupo, sarte_marca
WHERE mod_strecurso = str_codigo
    AND str_trecurso = trc_codigo
    AND trc_grupo = grp_codigo
    AND mod_marca = mar_codigo
    AND grp_detalle = 'S'
GO

```

3.2.6. VISTA SARTE_DRECURSOS

Vista generada a partir de las tablas "sarte_drecurso", "sarte_modelo", "sarte_strecurso", "sarte_trecurso" y "sarte_marca" que nos permite visualizar todos los recursos ingresados en el sistema con sus respectivas características.

CAMPO	TABLA
drc_codigo	sarte_drecurso
drc_recurso	sarte_drecurso
drc_modelo	sarte_drecurso
drc_estado	sarte_drecurso
mod_codigo	sarte_modelo
mod_descrip	sarte_modelo
str_codigo	sarte_strecurso
str_trecurso	sarte_strecurso
str_descrip	sarte_strecurso
trc_codigo	sarte_trecurso
trc_descrip	sarte_trecurso
trc_grupo	sarte_trecurso
mar_codigo	sarte_marca
mar_descrip	sarte_marca

Tabla 3.30. Estructura lógica de la vista sarte_drecursos.

```

CREATE VIEW sarte_drecursos AS
SELECT sarte_drecurso.drc_codigo,
    sarte_drecurso.drc_recurso,
    sarte_drecurso.drc_modelo,
    sarte_drecurso.drc_estado,
    sarte_modelo.mod_codigo,
    sarte_modelo.mod_descrip,
    sarte_strecurso.str_codigo,
    sarte_strecurso.str_trecurso,
    sarte_strecurso.str_descrip,
    sarte_trecurso.trc_codigo,
    sarte_trecurso.trc_descrip,
    sarte_trecurso.trc_grupo,
    sarte_marca.mar_codigo,
    sarte_marca.mar_descrip
FROM sarte_drecurso, sarte_modelo, sarte_strecurso, sarte_trecurso, sarte_marca
WHERE drc_modelo = mod_codigo
    AND mod_strecurso = str_codigo
    AND str_trecurso = trc_codigo
    AND mod_marca = mar_codigo
    AND drc_estado IN ('A','R')
GO

```

3.2.7. VISTA SARTE_SOPORTES

Vista generada a partir de las tablas "sarte_soporte", "sarte_tsoporte", "sarte_recurso" y "sarte_modelo" para mostrar un listado de todos los mantenimientos realizados a los recursos.

CAMPO	TABLA
sop_codigo	sarte_soporte
sop_tsoporte	sarte_soporte
sop_recurso	sarte_soporte
sop_fentrega	sarte_soporte
sop_frecibe	sarte_soporte
sop_descrip	sarte_soporte
sop_observa	sarte_soporte
sop_estado	sarte_soporte
tsp_codigo	sarte_tsoporte
tsp_descrip	sarte_tsoporte
rec_codigo	sarte_recurso
rec_oficina	sarte_recurso
rec_ubicacion	sarte_recurso
rec_custodio	sarte_recurso
rec_modelo	sarte_recurso
rec_usuario	sarte_recurso
rec_estrec	sarte_recurso
rec_activo	sarte_recurso
rec_fingreso	sarte_recurso
rec_fmodifica	sarte_recurso
rec_serie	sarte_recurso
rec_observa	sarte_recurso
rec_estado	sarte_recurso
mod_codigo	mod_codigo
mod_descrip	mod_codigo

Tabla 3.31. Estructura lógica de la vista sarte_soportes.

```
CREATE VIEW sarte_soportes AS
SELECT sarte_soporte.sop_codigo,
       sarte_soporte.sop_tsoporte,
       sarte_soporte.sop_recurso,
       CONVERT(VARCHAR(10), sarte_soporte.sop_fentrega, 101) sop_fentrega,
       CONVERT(VARCHAR(10), sarte_soporte.sop_frecibe, 101) sop_frecibe,
       sarte_soporte.sop_descrip,
       sarte_soporte.sop_observa,
       sarte_soporte.sop_estado,
       sarte_tsoporte.tsp_codigo,
       sarte_tsoporte.tsp_descrip,
       sarte_strecurso.str_codigo,
       sarte_strecurso.str_descrip,
       sarte_modelo.mod_codigo,
       sarte_modelo.mod_descrip,
       sarte_recurso.rec_codigo,
       sarte_recurso.rec_activo,
       sarte_recurso.rec_serie,
       sarte_recurso.rec_observa,
       adm_oficina.of_nombre,
```

```

    sarte_ubicacion.ubi_descrip,
    per_empleado.emp_nombre,
    sarte_estado.est_descrip
FROM sarte_soporte, sarte_tsoporte, sarte_recurso, sarte_modelo, sarte_strecurso,
adm_oficina, sarte_ubicacion, per_empleado, sarte_estado
WHERE sarte_soporte.sop_tsoporte = sarte_tsoporte.tsp_codigo
    AND sarte_soporte.sop_recurso = sarte_recurso.rec_codigo
    AND sarte_recurso.rec_modelo = sarte_modelo.mod_codigo
    AND sarte_modelo.mod_strecurso = sarte_strecurso.str_codigo
    AND sarte_soporte.sop_oficina = adm_oficina.of_codigo
    AND sarte_soporte.sop_ubicacion = sarte_ubicacion.ubi_codigo
    AND sarte_soporte.sop_custodio = per_empleado.emp_codigo
    AND sarte_recurso.rec_estrec = sarte_estado.est_codigo
GO

```

3.2.8. VISTA SARTE_TRECURSOS

Vista generada a partir de las tablas "*sarte_grupo*" y "*sarte_trecurso*" que nos permite visualizar un listado de todas las categorías parametrizadas en el sistema.

CAMPO	TABLA
grp_codigo	sarte_grupo
grp_nombre	sarte_grupo
trc_codigo	sarte_trecurso
trc_descrip	sarte_trecurso
trc_estado	sarte_trecurso

Tabla 3.32. Estructura lógica de la vista *sarte_trecursos*.

```

CREATE VIEW sarte_trecursos AS
SELECT sarte_trecurso.trc_codigo,
    sarte_trecurso.trc_grupo,
    sarte_grupo.grp_nombre,
    sarte_trecurso.trc_descrip,
    sarte_trecurso.trc_estado
FROM sarte_trecurso, sarte_grupo
WHERE sarte_grupo.grp_codigo = sarte_trecurso.trc_grupo
    AND trc_estado = 'A'
GO

```

3.2.9. VISTA SARTE_MODELOS

Vista generada a partir de las tablas "*sarte_modelo*", "*sarte_strecurso*", "*sarte_trecurso*", "*sarte_grupo*" y "*sarte_marca*" para mostrar un listado de todos los modelos de recursos disponibles para el ingreso de nuevos recursos al sistema.

CAMPO	TABLA
mod_codigo	sarte_modelo
mod_strecurso	sarte_modelo
mod_marca	sarte_modelo
mod_descrip	sarte_modelo
mod_estado	sarte_modelo
str_codigo	sarte_strecurso
str_descrip	sarte_strecurso
trc_codigo	sarte_trecurso

trc_descrip	sarte_trecurso
grp_codigo	sarte_grupo
grp_nombre	sarte_grupo
grp_descrip	sarte_grupo
mar_codigo	sarte_marca
mar_descrip	sarte_marca

Tabla 3.33. Estructura lógica de la vista sarte_modelos.

```

CREATE VIEW sarte_modelos AS
SELECT sarte_modelo.mod_codigo,
       sarte_modelo.mod_descrip,
       sarte_modelo.mod_estado,
       sarte_grupo.grp_codigo,
       sarte_grupo.grp_nombre,
       sarte_trecurso.trc_codigo,
       sarte_trecurso.trc_descrip,
       sarte_strecurso.str_codigo,
       sarte_strecurso.str_descrip,
       sarte_marca.mar_codigo,
       sarte_marca.mar_descrip
FROM sarte_modelo, sarte_strecurso, sarte_trecurso, sarte_grupo, sarte_marca
WHERE sarte_strecurso.str_codigo = sarte_modelo.mod_strecurso
      AND sarte_trecurso.trc_codigo = sarte_strecurso.str_trecurso
      AND sarte_grupo.grp_codigo = sarte_trecurso.trc_grupo
      AND sarte_marca.mar_codigo = sarte_modelo.mod_marca
GO

```

3.2.10. VISTA SARTE_STRECURSOS

Vista generada a partir de las tablas "sarte_grupo", "sarte_trecurso" y "sarte_strecurso" que muestra el detalle de todas las subcategorías de recursos parametrizadas en el sistema.

CAMPO	TABLA
grp_codigo	sarte_grupo
grp_nombre	sarte_grupo
trc_codigo	sarte_trecurso
trc_descrip	sarte_trecurso
str_trecurso	sarte_strecurso
str_descrip	sarte_strecurso
str_estado	sarte_strecurso

Tabla 3.34. Estructura lógica de la vista sarte_strecursos.

```

CREATE VIEW sarte_strecursos AS
SELECT sarte_strecurso.str_codigo,
       sarte_grupo.grp_codigo,
       sarte_grupo.grp_nombre,
       sarte_strecurso.str_trecurso,
       sarte_trecurso.trc_descrip,
       sarte_strecurso.str_descrip,
       sarte_strecurso.str_estado
FROM sarte_grupo, sarte_trecurso, sarte_strecurso
WHERE sarte_grupo.grp_codigo = sarte_trecurso.trc_grupo
      AND sarte_trecurso.trc_codigo = sarte_strecurso.str_trecurso
      AND str_estado = 'A'
GO

```

4. PROGRAMACION JAVA

Durante el tiempo de programación del sistema se desarrolló una variedad de clases que interactúan con la base de datos y con la interfaz del sistema. Con la finalidad de optimizar el código fuente del proyecto, hemos clasificado a las clases en tres paquetes.

4.1. PAQUETE "SARTE.BEANS"

En el paquete "sarte.beans" encontramos los objetos que interactúan directamente con las tablas de la base de datos y en este paquete tenemos las siguientes clases:

4.1.1. CLASE ACCION

Clase utilizada para interactuar con la tabla "sarte_accion" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *acn_codigo* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *acn_estado* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *acn_descrip* de la tabla.

Métodos

- **Accion:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el valor del atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getSiguiete:** método que retorna el máximo valor del campo *acn_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.

Código Fuente



Accion.java

4.1.2. CLASE DRECURSO

Clase utilizada para interactuar con la tabla "sarte_drecurso" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *drc_codigo* de la tabla.
- **recurso:** atributo tipo *int* que interactua con el campo *drc_recurso* de la tabla.
- **modelo:** atributo tipo *int* que interactua con el campo *drc_modelo* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *drc_estado* de la tabla.

Métodos

- **Drecurso:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el valor del atributo código del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getModelo:** método que retorna el valor del atributo modelo del objeto.
- **getRecurso:** método que retorna el valor del atributo recurso del objeto.
- **getSiguiente:** método que retorna el máximo valor del campo *drc_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setModelo:** método que asigna un valor al atributo modelo del objeto.
- **setRecurso:** método que asigna un valor al atributo recurso del objeto.

Código Fuente



Drecurso.java

4.1.3. CLASE DSOPORTE

Clase utilizada para interactuar con la tabla "*sarte_dsoporte*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *dsp_codigo* de la tabla.

- **soporte:** atributo tipo *int* que interactua con el campo *dsp_soporte* de la tabla.
- **accion:** atributo tipo *int* que interactua con el campo *dsp_accion* de la tabla.
- **drecurso:** atributo tipo *int* que interactua con el campo *dsp_drecurso* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *dsp_estado* de la tabla.

Métodos

- **Dsoporte:** constructor del objeto e inicia la conexión con la base de datos.
- **getAccion:** método que retorna el valor del atributo accion del objeto.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDrecurso:** método que retorna el valor del atributo descripción del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getSiguiete:** método que retorna el máximo valor del campo *dsp_codigo* de la tabla.
- **getSoporte:** método que retorna el valor del atributo soporte del objeto.
- **setAccion:** método que asigna un valor al atributo accion del objeto.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDrecurso:** método que asigna un valor al atributo drecurso del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setSoporte:** método que asigna un valor al atributo soporte del objeto.

Código Fuente



Dsoporte.java

4.1.4. CLASE ESTADO

Clase utilizada para interactuar con la tabla "*sarte_estado*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *est_codigo* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *est_descrip* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *est_estado* de la tabla.

Métodos

- **Estado:** constructor del objeto e inicia la conexión con la base de datos.

- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getSiguiete:** método que retorna el máximo valor del campo *est_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.

Código Fuente



Estado.java

4.1.5. CLASE GRUPO

Clase utilizada para interactuar con la tabla "sarte_grupo" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *grp_codigo* de la tabla.
- **nombre:** atributo tipo *String* que interactua con el campo *grp_nombre* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *grp_descrip* de la tabla.
- **principal:** atributo tipo *String* que interactua con el campo *grp_principal* de la tabla.
- **detalle:** atributo tipo *String* que interactua con el campo *grp_detalle* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *grp_estado* de la tabla.

Métodos

- **Grupo:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getDetalle:** método que retorna el valor del atributo detalle del objeto.
- **getEstado:** método que retorna el calor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getNombre:** método que retorna el valor del atributo nombre del objeto.

- **getPrincipal:** método que retorna el valor del atributo principal del objeto.
- **getSiguiente:** método que retorna el máximo valor del campo *grp_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setDetalle:** método que asigna un valor al atributo detalle del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setNombre:** método que asigna un valor al atributo nombre del objeto.
- **setPrincipal:** método que asigna un valor al atributo principal del objeto.

Código Fuente



Grupo.java

4.1.6. CLASE MARCA

Clase utilizada para interactuar con la tabla "*sarte_marca*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *mar_codigo* de la tabla.
- **grupo:** atributo tipo *int* que interactua con el campo *mar_grupo* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *mar_descrip* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *mar_estado* de la tabla.

Métodos

- **Marca:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getGrupo:** método que retorna el valor del atributo grupo del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getNgrupo:** método que retorna el nombre del grupo del objeto.
- **getSiguiente:** método que retorna el máximo valor del campo *mar_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.

- **setGrupo:** método que asigna un valor al atributo grupo del objeto.

Código Fuente



Marca.java

4.1.7. CLASE MODELO

Clase utilizada para interactuar con la tabla "sarte_modelo" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *mod_codigo* de la tabla.
- **marca:** atributo tipo *int* que interactua con el campo *mod_marca* de la tabla.
- **strecurso:** atributo tipo *String* que interactua con el campo *mod_strecurso* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *mod_descrip* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *mod_estado* de la tabla.

Métodos

- **Modelo:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getMarca:** método que retorna el valor del atributo marca del objeto.
- **getNstrecurso:** método que retorna el nombre del strecurso del objeto.
- **getSiguiente:** método que retorna el máximo valor del campo *mod_codigo* de la tabla.
- **getStrecurso:** método que retorna el valor del atributo strecurso del objeto.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setMarca:** método que asigna un valor al atributo marca del objeto.
- **setStrecurso:** método que asigna un valor al atributo strecurso del objeto.

Código Fuente



Modelo.java

4.1.8. CLASE MOVIMIENTO

Clase utilizada para interactuar con la tabla "sarte_movimiento" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *mov_codigo* de la tabla.
- **motivo:** atributo tipo *int* que interactua con el campo *mov_motivo* de la tabla.
- **recurso:** atributo tipo *int* que interactua con el campo *mov_recurso* de la tabla.
- **oficina:** atributo tipo *int* que interactua con el campo *mov_oficina* de la tabla.
- **ubicacion:** atributo tipo *int* que interactua con el campo *mov_ubica* de la tabla.
- **ofiant:** atributo tipo *int* que interactua con el campo *mov_ofiant* de la tabla.
- **ubiant:** atributo tipo *int* que interactua con el campo *mov_ubiant* de la tabla.
- **usuario:** atributo tipo *String* que interactua con el campo *mov_usuario* de la tabla.
- **custodio:** atributo tipo *String* que interactua con el campo *mov_custodio* de la tabla.
- **cusant:** atributo tipo *String* que interactua con el campo *mov_cusant* de la tabla.
- **fecha:** atributo tipo *String* que interactua con el campo *mov_fecha* de la tabla.
- **observa:** atributo tipo *String* que interactua con el campo *mov_observa* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *mov_estado* de la tabla.

Métodos

- **Movimiento:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getCusant:** método que retorna el atributo cusant del objeto.
- **getCustodio:** método que retorna el atributo custodio del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getFecha:** método que retorna el valor del atributo fecha del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getMotivo:** método que retorna el valor del atributo motivo del objeto.
- **getObserva:** método que retorna el valor del atributo observa del objeto.
- **getOfiant:** método que retorna el valor del atributo ofiant del objeto.
- **getOficina:** método que retorna el valor del atributo oficina del objeto.

- **getRecurso:** método que retorna el valor del atributo recurso del objeto.
- **getSiguiete:** método que retorna el máximo valor del campo *mov_codigo* de la tabla.
- **getUbiant:** método que retorna el valor del atributo ubiant del objeto.
- **getUbicacion:** método que retorna el valor del atributo ubicacion del objeto.
- **getUsuario:** método que retorna el valor del atributo usuario del objeto.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setCusant:** método que asigna un valor al atributo cusant del objeto.
- **setCustodio:** método que asigna un valor al atributo custodio del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setFecha:** método que asigna un valor al atributo fecha del objeto.
- **setMotivo:** método que asigna un valor al atributo motivo del objeto.
- **setObserva:** método que asigna un valor al atributo observa del objeto.
- **setOfiant:** método que asigna un valor al atributo ofiant del objeto.
- **setOficina:** método que asigna un valor al atributo oficina del objeto.
- **setRecurso:** método que asigna un valor al atributo recurso del objeto.
- **setUbiant:** método que asigna un valor al atributo ubiant del objeto.
- **setUbicacion:** método que asigna un valor al atributo ubicacion del objeto.
- **setUsuario:** método que asigna un valor al atributo usuario del objeto.

Código Fuente



Movimiento.java

4.1.9. CLASE RECURSO

Clase utilizada para interactuar con la tabla "*sarte_recurso*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *rec_codigo* de la tabla.
- **oficina:** atributo tipo *int* que interactua con el campo *rec_oficina* de la tabla.
- **ubicacion:** atributo tipo *int* que interactua con el campo *rec_ubica* de la tabla.
- **modelo:** atributo tipo *int* que interactua con el campo *rec_modelo* de la tabla.
- **estrec:** atributo tipo *int* que interactua con el campo *rec_estrec* de la tabla.
- **custodio:** atributo tipo *String* que interactua con el campo *rec_custodio* de la tabla.
- **usuario:** atributo tipo *String* que interactua con el campo *rec_usuario* de la tabla.
- **activo:** atributo tipo *String* que interactua con el campo *rec_activo* de la tabla.

- **fingreso:** atributo tipo *String* que interactua con el campo *rec_fingreso* de la tabla.
- **fmodifica:** atributo tipo *String* que interactua con el campo *rec_fmodifica* de la tabla.
- **serie:** atributo tipo *String* que interactua con el campo *rec_serie* de la tabla.
- **observa:** atributo tipo *String* que interactua con el campo *rec_observa* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *rec_estado* de la tabla.

Métodos

- **Recurso:** constructor del objeto e inicia la conexión con la base de datos.
- **getActivo:** método que retorna el atributo activo del objeto.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getCustodio:** método que retorna el atributo custodio del objeto.
- **getDetalles:** método que determina si un tiene o no tiene descripción de características.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getEstrec:** método que retorna el valor del atributo estrec del objeto.
- **getFingreso:** método que retorna el valor del atributo fingreso del objeto.
- **getFmodifica:** método que retorna el valor del atributo fmodifica del objeto.
- **getGrupo:** método que retorna el valor del atributo grupo del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getModelo:** método que retorna el valor del atributo modelo del objeto.
- **getNcustodio:** método que retorna el nombre del custodio del recurso.
- **getNestrec:** método que retorna el nombre del estado del recurso.
- **getNmarca:** método que retorna el nombre de la marca del recurso.
- **getNmodelo:** método que retorna el nombre del modelo del recurso.
- **getNoficina:** método que retorna el nombre de la oficina del recurso.
- **getNstrecurso:** método que retorna el nombre de la subcategoría del recurso.
- **getNubicacion:** método que retorna el nombre de la ubicación del recurso.
- **getObserva:** método que retorna el valor del atributo observa del objeto.
- **getOficina:** método que retorna el valor del atributo oficina del objeto.
- **getSerie:** método que retorna el valor del atributo serie del objeto.
- **getSiguiete:** método que retorna el máximo valor del campo *rec_codigo* de la tabla.
- **getTrecurso:** método que retorna el valor del atributo trecurso del objeto.
- **getUbicacion:** método que retorna el valor del atributo ubicacion del objeto.
- **getUsuario:** método que retorna el valor del atributo usuario del objeto.
- **setActivo:** método que asigna un valor al atributo activo del objeto.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setCustodio:** método que asigna un valor al atributo custodio del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setEstrec:** método que asigna un valor al atributo estrec del objeto.

- **setFingreso:** método que asigna un valor al atributo `fingreso` del objeto.
- **setFmodifica:** método que asigna un valor al atributo `fmodifica` del objeto.
- **setModelo:** método que asigna un valor al atributo `modelo` del objeto.
- **setObserva:** método que asigna un valor al atributo `observa` del objeto.
- **setOficina:** método que asigna un valor al atributo `oficina` del objeto.
- **setRecurso:** método que asigna un valor al atributo `recurso` del objeto.
- **setSerie:** método que asigna un valor al atributo `serie` del objeto.
- **setUbicacion:** método que asigna un valor al atributo `ubicacion` del objeto.
- **setUsuario:** método que asigna un valor al atributo `usuario` del objeto.

Código Fuente



Recurso.java

4.1.10. CLASE SOPORTE

Clase utilizada para interactuar con la tabla "`sarte_soporte`" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo `sop_codigo` de la tabla.
- **recurso:** atributo tipo *int* que interactua con el campo `sop_recurso` de la tabla.
- **oficina:** atributo tipo *int* que interactua con el campo `sop_oficina` de la tabla.
- **ubicacion:** atributo tipo *int* que interactua con el campo `sop_ubica` de la tabla.
- **tsoporte:** atributo tipo *int* que interactua con el campo `sop_tsoporte` de la tabla.
- **estrec:** atributo tipo *int* que interactua con el campo `sop_estrec` de la tabla.
- **custodio:** atributo tipo *String* que interactua con el campo `sop_custodio` de la tabla.
- **fentrega:** atributo tipo *String* que interactua con el campo `sop_fentrega` de la tabla.
- **frecibe:** atributo tipo *String* que interactua con el campo `sop_frecibe` de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo `sop_descrip` de la tabla.
- **observa:** atributo tipo *String* que interactua con el campo `sop_observa` de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo `sop_estado` de la tabla.

Métodos

- **Soporte:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.

- **getCustodio:** método que retorna el atributo custodio del objeto.
- **getDescrip:** método que retorna el atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getEstrec:** método que retorna el valor del atributo estrec del objeto.
- **getFecha:** método que retorna la fecha actual del sistema.
- **getFentrega:** método que retorna el valor del atributo fentrega del objeto.
- **getFrecibe:** método que retorna el valor del atributo frecibe del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getNcustodio:** método que retorna el nombre del custodio del recurso.
- **getNestrec:** método que retorna el nombre del estado del recurso.
- **getNoficina:** método que retorna el nombre de la oficina del recurso.
- **getNtsoporte:** método que retorna el nombre del tipo de soporte dado al recurso.
- **getNubicacion:** método que retorna el nombre de la ubicación del recurso.
- **getObserva:** método que retorna el valor del atributo observa del objeto.
- **getOficina:** método que retorna el valor del atributo oficina del objeto.
- **getRecurso:** método que retorna el valor del atributo recurso del objeto.
- **getSiguiete:** método que retorna el máximo valor del campo *sop_codigo* de la tabla.
- **getTsoporte:** método que retorna el valor del atributo tsoporte del objeto.
- **getUbicacion:** método que retorna el valor del atributo ubicacion del objeto.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setCustodio:** método que asigna un valor al atributo custodio del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setEstrec:** método que asigna un valor al atributo estrec del objeto.
- **setFentrega:** método que asigna un valor al atributo fentrega del objeto.
- **setFrecibe:** método que asigna un valor al atributo frecibe del objeto.
- **setObserva:** método que asigna un valor al atributo observa del objeto.
- **setOficina:** método que asigna un valor al atributo oficina del objeto.
- **setRecurso:** método que asigna un valor al atributo recurso del objeto.
- **setTsoporte:** método que asigna un valor al atributo tsoporte del objeto.
- **setUbicacion:** método que asigna un valor al atributo ubicacion del objeto.

Código Fuente



Soporte.java

4.1.11. CLASE STRECURSO

Clase utilizada para interactuar con la tabla "*sarte_strecurso*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *str_codigo* de la tabla.
- **trecurso:** atributo tipo *int* que interactua con el campo *str_trecurso* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *str_descrip* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *str_estado* de la tabla.

Métodos

- **Strecurso:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getNtrecurso:** método que retorna el nombre del tipo de recurso del objeto.
- **getSiguiete:** método que retorna el máximo valor del campo *str_codigo* de la tabla.
- **getTrecurso:** método que retorna el valor del atributo trecurso del objeto.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setTrecurso:** método que asigna un valor al atributo trecurso del objeto.

Código Fuente



Strecurso.java

4.1.12. CLASE TRECURSO

Clase utilizada para interactuar con la tabla "*sarte_trecurso*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *trc_codigo* de la tabla.
- **grupo:** atributo tipo *String* que interactua con el campo *trc_grupo* de la tabla.

- **descrip:** atributo tipo *String* que interactua con el campo *trc_descrip* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *trc_estado* de la tabla.

Métodos

- **Trecurso:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getGrupo:** método que retorna el valor del atributo grupo del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getNgrupo:** método que retorna el nombre del grupo del objeto.
- **getSiguiete:** método que retorna el máximo valor del campo *trc_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setGrupo:** método que asigna un valor al atributo grupo del objeto.

Código Fuente



Trecurso.java

4.1.13. CLASE TSOPORTE

Clase utilizada para interactuar con la tabla "*sarte_tsoporte*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *tsp_codigo* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *tsp_descrip* de la tabla.
- **detalle:** atributo tipo *String* que interactua con el campo *tsp_detalle* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *tsp_estado* de la tabla.

Métodos

- **Tsoporte:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.

- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getDetalle:** método que retorna el valor del atributo detalle del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getSiguiete:** método que retorna el máximo valor del campo *tsp_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setDetalle:** método que asigna un valor al atributo detalle del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.

Código Fuente



Tsoporte.java

4.1.14. CLASE UBICACION

Clase utilizada para interactuar con la tabla "*sarte_ubicacion*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **codigo:** atributo tipo *int* que interactua con el campo *ubi_codigo* de la tabla.
- **descrip:** atributo tipo *String* que interactua con el campo *ubi_descrip* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *ubi_estado* de la tabla.

Métodos

- **Ubicacion:** constructor del objeto e inicia la conexión con la base de datos.
- **getActualizar:** método que modifica los datos de un registro de la tabla.
- **getCargar:** método que recupera los datos de un registro de la tabla.
- **getCodigo:** método que retorna el atributo código del objeto.
- **getDescrip:** método que retorna el valor del atributo descrip del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getInsertar:** método que inserta un registro a la tabla.
- **getSiguiete:** método que retorna el máximo valor del campo *tsp_codigo* de la tabla.
- **setCodigo:** método que asigna un valor al atributo código del objeto.
- **setDescrip:** método que asigna un valor al atributo descrip del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.

Código Fuente



Ubicacion.java

4.1.15. CLASE USUARIO

Clase utilizada para interactuar con la tabla "*adm_usuario*" de la base de datos.

Atributos

- **con.-** atributo tipo *Conexion* a la conexión a la base de datos.
- **rs.-** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sql.-** atributo tipo *String* para la sentencia sql a ejecutarse en la base de datos.
- **msg1:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **msg2:** atributo tipo *String* para capturar los mensajes de la conexión a la base de datos.
- **usuario:** atributo tipo *String* que interactua con el campo *us_login* de la tabla.
- **clave:** atributo tipo *String* que interactua con el campo *us_clave* de la tabla.
- **nombre:** atributo tipo *String* que interactua con el campo *us_nombre* de la tabla.
- **estado:** atributo tipo *String* que interactua con el campo *us_estado* de la tabla.
- **empleado:** atributo tipo *String* que interactua con el campo *us_empleado* de la tabla.

Métodos

- **Usuario:** constructor del objeto e inicia la conexión con la base de datos.
- **getClave:** método que retorna el atributo clave del objeto.
- **getEmpleado:** método que retorna el valor del atributo empleado del objeto.
- **getEstado:** método que retorna el valor del atributo estado del objeto.
- **getNombre:** método que retorna el valor del atributo nombre del objeto.
- **getUsuario:** método que retorna el valor del atributo usuario del objeto.
- **getVerificar:** método que verifica si el usuario puede acceder al sistema.
- **setClave:** método que asigna un valor al atributo clave del objeto.
- **setEmpleado:** método que asigna un valor al atributo empleado del objeto.
- **setEstado:** método que asigna un valor al atributo estado del objeto.
- **setNombre:** método que asigna un valor al atributo nombre del objeto.
- **setUsuario:** método que asigna un valor al atributo usuario del objeto.

Código Fuente



Usuario.java

4.2. PAQUETE "SARTE.DATOS"

En este paquete se encuentra la programación necesaria para conectar el sistema con la base de datos. En este paquete existe una única clase llamada conexión y que la pasamos a detallar a continuación:

4.2.1. CLASE CONEXIÓN

La clase conexión es la encargada de interactuar entre el sistema y la base de datos, en esta clase se encuentran las funciones necesarias para conectar, ingresar y modificar la información de la base de datos.

Atributos

- **conexion:** atributo tipo *Connection* a la conexión a la base de datos.
- **status:** atributo tipo *Statement* para ejecutar la consulta sql en la base de datos.
- **result:** atributo tipo *ResultSet* para recuperar los datos de la base de datos.
- **sqle:** atributo tipo *String* para capturar las excepciones producidas por la consulta sql.
- **cnfe:** atributo tipo *String* para capturar las excepciones producidas por el objeto.
- **usuario:** atributo tipo *String* que indica el usuario para conectarse a la base de datos.
- **clave:** atributo tipo *String* que indica la clave para conectarse a la base de datos.
- **base:** atributo tipo *String* que indica la dirección de la base de datos.

Métodos

- **Conexion:** constructor del objeto que inicia la conexión con la base de datos.
- **Actualiza:** método utilizado para realizar modificación a los datos de la base.
- **Cerrar:** método empleado para cerrar la conexión con la base de datos.
- **Consulta:** método utilizado para realizar una consulta sql a la base de datos.

Código Fuente



Conexion.java

4.3. PAQUETE "SARTE.TAGS"

En el paquete `sarte.tags` se encuentran las clases de las etiquetas personalizadas, para la generación de reportes en el sistema y que interactúan con las páginas JSP por medio de etiquetas.

Las clases que pertenecen a este paquete son:

4.3.1. CLASE `ADMGRUPOS`

Esta clase genera el menú lateral izquierdo de grupos para el módulo de parámetros.

Código Fuente



`admGrupos.java`

4.3.2. CLASE `ADM Marcas`

Esta clase genera un reporte de las marcas de un determinado grupo de recursos para el módulo de parámetros.

Código Fuente



`admMarcas.java`

4.3.3. CLASE `ADMSTRECURSOS`

Esta clase genera el menú lateral izquierdo de subcategorías para el módulo de parámetros.

Código Fuente



`admStrecursos.java`

4.3.4. CLASE `ADM TRECURSOS`

Esta clase genera el menú lateral izquierdo de categorías para el módulo de parámetros.

Código Fuente



`admTrecursos.java`

4.3.5. CLASE `INGOFICINA`

Esta clase genera un reporte de los recursos tecnológicos existentes en una determinada opción y organizados por grupo, categoría, subcategoría, modelo, oficina y custodio; dependiendo del filtrado y la búsqueda de recursos realizada.

Código Fuente

ingOficina.java

4.3.6. CLASE INGTRECURSOS

Esta clase genera el menú lateral izquierdo de categorías para el módulo de recursos.

Código Fuente

ingTrecursos.java

4.3.7. CLASE LISAREAS

Esta clase genera un reporte de las áreas o ubicaciones parametrizadas en el sistema.

Código Fuente

lisAreas.java

4.3.8. CLASE LISDRECURSOS

Esta clase genera un reporte de las características internas de los recursos.

Código Fuente

lisDrecursos.java

4.3.9. CLASE LISDSOPORTES

Esta clase genera un reporte de los componentes de los recursos afectados durante un mantenimiento.

Código Fuente

lisDsoportes.java

4.3.10. CLASE LISESTADOS

Esta clase genera un reporte de los posibles estados de los recursos, establecidos en el sistema.

Código Fuente



lisEstados.java

4.3.11. CLASE LISGRUPOS

Esta clase genera un reporte de los grupos de recursos tecnológicos establecidos en el sistema.

Código Fuente



lisGrupos.java

4.3.12. CLASE LISMARCAS

Esta clase genera un reporte de las marcas establecidas en el sistema y organizadas por grupo de recursos.

Código Fuente



lisMarcas.java

4.3.13. CLASE LISMODELOS

Esta clase genera un reporte de los modelos de recursos parametrizados en el sistema y organizados por categorías y subcategorías de recursos.

Código Fuente



lisModelos.java

4.3.14. CLASE LISMOVIMIENTOS

Esta clase genera un reporte de los movimientos realizados con un determinado recurso.

Código Fuente



lisMovimientos.java

4.3.15. CLASE LISSOPFIN

Esta clase genera un reporte de los mantenimientos realizados a os recursos tecnológicos y terminados satisfactoriamente.

Código Fuente



lisSopfin.java

4.3.16. CLASE LISOPORTES

Esta clase genera un reporte de solicitudes de mantenimientos vigentes.

Código Fuente



lisSoportes.java

4.3.17. CLASE LISSTRECURSOS

Esta clase genera un reporte de subcategorías de recursos establecidas en el sistema.

Código Fuente



lisStrecursos.java

4.3.18. CLASE LISTRECURSOS

Esta clase genera un reporte de categorías de recursos establecidas en el sistema.

Código Fuente



lisTrecursos.java

4.3.19. CLASE LISTSOPORTES

Esta clase genera un reporte de los tipos de mantenimientos parametrizados en el sistema.

Código Fuente



lisTsoportes.java

4.3.20. CLASE REPDRECURSOS

Esta clase genera un reporte de las características internas de un determinado recurso para el módulo de reportes.

Código Fuente



repDrecursos.java

4.3.21. CLASE REPGRUPOS

Esta clase genera el menú lateral izquierdo de grupos para el módulo de reportes.

Código Fuente



repGrupos.java

4.3.22. CLASE REPOFICINAS

Esta clase genera el menú lateral izquierdo de oficinas para el módulo de reportes.

Código Fuente



repOficinas.java

4.3.23. CLASE REPTRECURSOS

Esta clase genera el menú lateral izquierdo de categorías para el módulo de reportes.

Código Fuente



repTrecursos.java

4.3.24. CLASE RPOFICINA

Esta clase genera un reporte de recursos tecnológicos por oficina, clasificado por grupo, categoría y subcategoría de recursos.

Código Fuente



rpOficina.java

4.3.25. CLASE RPRECURSOS

Esta clase genera un reporte de recursos tecnológicos de una determinada categoría de recursos.

Código Fuente



rpRecursos.java

4.3.26. CLASE RPSOPFIN

Esta clase genera un reporte de mantenimiento realizados y finalizados sin novedades de un determinado recurso tecnológico.

Código Fuente



rpSopfin.java

4.3.27. CLASE RPSOPORTES

Esta clase genera un reporte de solicitudes de mantenimientos vigentes de un determinado recurso tecnológico.

Código Fuente



rpSoportes.java

4.3.28. CLASE RPTRECURSOS

Esta clase genera un reporte de las categorías de recursos de un determinado grupo de recursos.

Código Fuente



rpTrecursos.java

4.3.29. CLASE SELACCION

Esta clase genera un elemento de formulario tipo select para elegir el tipo de acción a realizar con un determinado componente de un recurso.

Código Fuente



selAccion.java

4.3.30. CLASE SELDETALLE

Esta clase genera un elemento de formulario tipo select para elegir el modelo del componente a ingresar.

Código Fuente



selDetalle.java

4.3.31. CLASE SELEMPLEADO

Esta clase genera un elemento de formulario tipo select para elegir el empleado correspondiente a un determinado recurso.

Código Fuente



selEmpleado.java

4.3.32. CLASE SELESTADO

Esta clase genera un elemento de formulario tipo select para elegir el estado actual de un recurso.

Código Fuente



selEstado.java

4.3.33. CLASE SELGRUPO

Esta clase genera un elemento de formulario tipo select para elegir el grupo.

Código Fuente



selGrupo.java

4.3.34. CLASE SELMARCA

Esta clase genera un elemento de formulario tipo select para elegir la marca.

Código Fuente



selMarca.java

4.3.35. CLASE SELMODELO

Esta clase genera un elemento de formulario tipo select para elegir el modelo.

Código Fuente



selModelo.java

4.3.36. CLASE SELMOTIVO

Esta clase genera un elemento de formulario tipo select para elegir el motivo del traslado.

Código Fuente



selMotivo.java

4.3.37. CLASE SELOFICINA

Esta clase genera un elemento de formulario tipo select para elegir la oficina.

Código Fuente



selOficina.java

4.3.38. CLASE SELSTRECURSO

Esta clase genera un elemento de formulario tipo select para elegir la subcategoría del recurso.

Código Fuente



selStrecurso.java

4.3.39. CLASE SELTRECORSO

Esta clase genera un elemento de formulario tipo select para elegir la categoría del recurso.

Código Fuente



selTrecorso.java

4.3.40. CLASE SELTSOPORTE

Esta clase genera un elemento de formulario tipo select para elegir el tipo de soporte.

Código Fuente



selTsoporte.java

4.3.41. CLASE SELUBICACION

Esta clase genera un elemento de formulario tipo select para elegir la ubicación del recurso.

Código Fuente



selUbicacion.java

4.3.42. CLASE SOPDRECURSOS

Esta clase genera un reporte de las características de un determinado recurso para el módulo de mantenimientos.

Código Fuente



sopDrecursos.java

4.3.43. CLASE SOPMDRECURSOS

Esta clase genera un reporte de las características de un determinado recurso para el módulo de mantenimientos.

Código Fuente



sopMdrecursos.java

4.3.44. CLASE SOPOFICINA

Esta clase genera emite un reporte de los recursos tecnológicos existentes en una determinada oficina y clasificados por grupo, categoría, subcategoría y modelo de recurso para el módulo de mantenimientos del sistema.

Código Fuente



sopOficina.java

4.3.45. CLASE TRSOFICINA

Esta clase genera emite un reporte de los recursos tecnológicos existentes en una determinada oficina y clasificados por grupo, categoría, subcategoría y modelo de recurso para el módulo de traslados del sistema.

Código Fuente



trsOficina.java

5. JASPERREPORTS

El objetivo principal del Sistema de Administración y Control de Recursos Tecnológicos es visualizar la información del sistema de manera organizada, detallada y/o resumida al usuario para evaluar el desempeño de los recursos y tomar las decisiones más acertadas en el proceso de administración.

En el sistema se pueden generar una variedad de reportes que están diseñados en la herramienta de desarrollo JasperReports y que se explica más detalladamente a continuación:

5.1.1. REPORTE DMOVIMIENTO

Este reporte genera un listado detallado de movimientos realizados con un determinado recurso y recibe como parámetro el código del recurso del cual se desea consultar sus traslados.

Código Fuente



dmovimiento.jrxml

5.1.2. REPORTE DMOVIMIENTO

Este reporte genera un listado detallado de las características de un determinado recurso y recibe como parámetro el código del recurso del cual se desea consultar sus características.

Código Fuente



drecurso.jrxml

5.1.3. REPORTE DSOORTE

Este reporte genera un listado detallado de las actividades realizadas en un determinado mantenimiento y recibe como parámetro el código del mantenimiento del cual se desea consultar su detalle.

Código Fuente



dsoporte.jrxml

5.1.4. REPORTE INVCUSTODIO

Este reporte genera el inventario general detallado de recursos para un determinado custodio organizados por grupo, categoría y subcategoría de recursos; recibe como parámetro el código del custodio del cual se desea conocer los recursos a cargo.

Código Fuente



invcustodio.jrxml

5.1.5. REPORTE INVGENERAL

Este reporte genera el inventario general detallado de recursos donde constan todos los recursos del sistema organizados por grupos, categorías y subcategorías de recursos, no recibe ningún parámetro pues realiza la consulta de todo el inventario.

Código Fuente



invgeneral.jrxml

5.1.6. REPORTE INVGENERALR

Este reporte genera el inventario general resumido de recursos donde constan todos los recursos del sistema organizados por grupos, categorías, subcategorías y modelos de recursos, no recibe ningún parámetro pues realiza la consulta de todo el inventario.

Código Fuente



invgeneralr.jrxml

5.1.7. REPORTE INVGRUPO

Este reporte genera el inventario general detallado de recursos para un determinado grupo de recursos organizado por categoría y subcategoría de recursos; recibe como parámetro el código del grupo de recursos que se desea generar el reporte.

Código Fuente



invgrupo.jrxml

5.1.8. REPORTE INVGRUPOR

Este reporte genera el inventario general resumido de recursos para un determinado grupo de recursos organizado por categoría y subcategoría de recursos; recibe como parámetro el código del grupo de recursos que se desea generar el reporte.

Código Fuente



invgrupor.jrxml

5.1.9. REPORTE INVMODELO

Este reporte genera el inventario general detallado de recursos para un determinado tipo de modelo; recibe como parámetro el código del modelo del cual se desea generar el reporte.

Código Fuente



invmodelo.jrxml

5.1.10. REPORTE INVOFICINA

Este reporte genera el inventario general detallado de recursos para una determinada oficina; recibe como parámetro el código de la oficina de la cual se desea generar el reporte.

Código Fuente



invoficina.jrxml

5.1.11. REPORTE INVOFICINAR

Este reporte genera el inventario general resumido de recursos para una determinada oficina; recibe como parámetro el código de la oficina de la cual se desea generar el reporte.

Código Fuente



invoficinar.jrxml

5.1.12. REPORTE INVTRECURSO

Este reporte genera el inventario general detallado de recursos para una determinada categoría de recursos; recibe como parámetro el código de la categoría de la cual se desea generar el reporte.

Código Fuente



invtrecurso.jrxml

5.1.13. REPORTE INVTRECURSOR

Este reporte genera el inventario general resumido de recursos para una determinada categoría de recursos; recibe como parámetro el código de la categoría de la cual se desea generar el reporte.

Código Fuente



invtrecurso.jrxml

5.1.14. REPORTE MODELOS

Este reporte genera el listado de general de modelos existentes en el sistema y recibe como parámetro el código del modelo del cual se desea generar el reporte.

Código Fuente



modelos.jrxml

5.1.15. REPORTE RECMOVIMIENTO

Este reporte genera un reporte de los movimientos o traslados realizados con un determinado recurso y recibe como único parámetro el código de recurso del cual se desea generar el reporte.

Código Fuente



recmovimiento.jrxml

5.1.16. REPORTE RECSOPORTES

Este reporte genera un reporte de las solicitudes de mantenimientos vigentes y no recibe ningún parámetro.

Código Fuente



recsoportes.jrxml

5.1.17. REPORTE RECURSO

Este reporte emite un informe general de recurso para un determinado recurso y recibe como parámetro el código de recurso del cual se desea generar el reporte.

Código Fuente



recurso.jrxml

5.1.18. REPORTE SOPORTE

Este reporte emite un informe detallado de mantenimiento para una determinada solicitud de mantenimiento y recibe como parámetro el código de la solicitud.

Código Fuente



soporte.jrxml

5.1.19. REPORTE SOPORTES

Este reporte emite un reporte general de las solicitudes de mantenimientos vigentes en el sistema, este reporte no recibe ningún parámetro.

Código Fuente



soportes.jrxml

5.1.20. REPORTE SOPORTESF

Este reporte emite un reporte general de los mantenimientos realizados en los diferentes recursos del sistema, este reporte no recibe ningún parámetro.

Código Fuente



soportesf.jrxml

6. PROGRAMACION WEB

La interfaz del sistema es a través de página web JSP que utilizan los paquetes de clases y reportes detallados anteriormente. Al estar el sistema dividido en módulos, la programación de las páginas también se dividen en grupos y estos grupos de páginas son:

6.1.1. INGRESO AL SISTEMA

Para ingresar al sistema el usuario debe registrarse en la página de ingreso, para ello existe una página de inicio que permiten al usuario interactuar con el sistema.

Página index.jsp

La página index.jsp tiene el formulario de ingreso de usuario y contraseña para la autenticación del usuario; si los datos proporcionados son correctos el sistema direcciona el control a la página de bienvenida, caso contrario el sistema vuelve a solicitar la información hasta un máximo de 3 veces antes de bloquear al usuario.

Código fuente



index.jsp

6.1.2. MÓDULO DE ADMINISTRACIÓN

En el Módulo de Administración encontramos las páginas necesarias para realizar la parametrización del sistema, estas páginas interactúan con las tablas de parámetros del sistema y nos permiten crear o modificar datos en sus diferentes categorías.

Página areas.jsp

Esta página nos permite manipular toda la información correspondiente a las áreas o ubicaciones del sistema, en esta página se encuentra el formulario para ingresar o modificar una determinada ubicación.

Código fuente



areas.jsp

Página estados.jsp

Esta página nos permite manipular toda la información correspondiente a los estados de los recursos del sistema, en esta página se encuentra el formulario para ingresar o modificar un determinado estado.

Código fuente



estados.jsp

Página grupos.jsp

Esta página nos permite manipular toda la información correspondiente a lgrupos de recursos del sistema, en esta página se encuentra el formulario para ingresar o modificar un determinado grupo de recursos.

Código fuente



grupos.jsp

Página marcas.jsp

Esta página nos permite manipular toda la información correspondiente a las marcas de los recursos, en esta página se encuentra el formulario para ingresar o modificar una determinada marca.

Código fuente



marcas.jsp

Página modelos.jsp

Esta página nos permite manipular toda la información correspondiente a los modelos de los recursos, en esta página se encuentra el formulario para ingresar o modificar un determinado modelo de recursos.

Código fuente



modelos.jsp

Página recursos.jsp

Esta página nos permite manipular toda la información correspondiente a los recursos del sistema, en esta página se encuentra el formulario para ingresar o modificar un determinado recurso.

Código fuente



recursos.jsp

Página tiposRecursos.jsp

Esta página nos permite manipular toda la información correspondiente a los tipos de recursos existentes en el sistema, en esta página se encuentra el formulario correspondiente para ingresar o modificar un determinado tipo de recurso.

Código fuente



tiposRecursos.jsp

Página tiposSoportes.jsp

Esta página nos permite manipular toda la información correspondiente a los tipos de mantenimientos del sistema, en esta página se encuentra el formulario correspondiente para ingresar o modificar un determinado tipo de mantenimiento.

Código fuente



tiposSoportes.jsp

6.1.3. MODULO DE RECURSOS

El módulo de recursos es el que le permiten al usuario registrar todas las propiedades y características de los recursos, pudiendo el usuario ingresar, modificar y eliminar información de los recursos tecnológicos. Para estas actividades el módulo consta de las siguientes páginas.

Página eliRecurso.jsp

La página eliRecurso.jsp permite eliminar un recurso del inventario general de recursos, esta página tiene el formulario de confirmación para la eliminación del recurso si la respuesta es afirmativa el recurso es eliminado del inventario, caso contrario se cancela la eliminación y se retorna al listado general de recursos.

Código fuente



eliRecurso.jsp

Página imprecurso.jsp

La página imprecurso.jsp permite imprimir el reporte de un determinado recurso, donde constan las características y propiedades de un recurso.

Código fuente



imprecurso.jsp

Página ingreso.jsp

La página ingreso.jsp permite ingresar un nuevo recurso al inventario general de recursos, en esta página se encuentra el formulario para ingresar las diferentes propiedades del recurso.

Código fuente



ingreso.jsp

Página recurso.jsp

La página recurso.jsp permite emitir una página de las características y propiedades de un determinado recurso.

Código fuente



recurso.jsp

Página recursos.jsp

La página recursos.jsp visualiza un listado general de recursos clasificado por grupo, categoría y subcategoría de recursos.

Código fuente



recursos.jsp

Página verRecurso.jsp

La página verRecurso.jsp visualiza las propiedades y características de un determinado recurso.

Código fuente



verRecurso.jsp

6.1.4. MODULO DE MANTENIMIENTOS

El módulo de mantenimientos posee toda la lógica para registrar las actividades realizadas cuando se realiza un mantenimiento a un determinado recurso. Para interactuar con el usuario el módulo implementa una variedad de páginas que se pasan a describir a continuación.

Página detSoporte.jsp

La página detSoporte.jsp visualiza un listado donde se detallan todas las actividades realizadas en un determinado mantenimiento.

Código fuente



detSoporte.jsp

Página eliSoporte.jsp

La página eliSoporte.jsp visualiza el formulario de confirmación para la eliminación de una solicitud de mantenimiento; si se confirma la eliminación la solicitud es eliminada, caso contrario la página se direcciona al listado general de solicitudes vigentes.

Código fuente



eliSoporte.jsp

Página finSoporte.jsp

La página finSoporte.jsp visualiza el formulario para detallar las actividades de un mantenimiento y finalizar la solicitud de mantenimiento.

Código fuente



finSoporte.jsp

Página finSoportes.jsp

La página finSoportes.jsp visualiza un listado de todos los mantenimientos realizados y registrados en el sistema.

Código fuente



finSoportes.jsp

Página imprecurso.jsp

La página imprecurso.jsp permite al usuario imprimir un reporte con la información correspondiente a los mantenimientos realizados en un determinado recurso.

Código fuente



imprecurso.jsp

Página impsORTE.jsp

La página impsORTE.jsp permite al usuario imprimir un reporte con todas las actividades realizadas en un mantenimiento de recurso en particular.

Código fuente



impsORTE.jsp

Página impsORTes.jsp

La página impsORTes.jsp permite imprimir un listado de solicitudes de mantenimientos vigentes.

Código fuente



impsORTes.jsp

Página impsORTesf.jsp

La página impsORTesf.jsp permite imprimir un listado de los mantenimientos realizados y registrados en el sistema.

Código fuente



impsORTesf.jsp

Página modSoporte.jsp

La página modSoporte.jsp permite al usuario modificar las propiedades de una solicitud de mantenimiento.

Código fuente



modSoporte.jsp

Página recSoportes.jsp

La página recSoportes.jsp permite al usuario emitir un listado de las solicitudes de mantenimiento vigentes y los mantenimientos realizados en un determinado recurso.

Código fuente



recSoportes.jsp

Página recurso.jsp

La página recurso.jsp permite al usuario visualizar un listado de las características y propiedades de un determinado recurso.

Código fuente



recurso.jsp

Página recursos.jsp

La página recursos.jsp permite al usuario visualizar un listado general de recursos clasificado por grupo, categoría y subcategoría de recursos.

Código fuente



recursos.jsp

Página soportes.jsp

La página soportes.jsp permite al usuario visualizar un listado general de las solicitudes de mantenimiento vigentes en el sistema.

Código fuente



soportes.jsp

Página sopRecurso.jsp

La página sopRecurso.jsp permite al usuario ingresar una nueva solicitud de mantenimiento, posee el formulario para ingresar los parámetros necesarios para la solicitud.

Código fuente



sopRecurso.jsp

Página verSoporte.jsp

La página verSoporte.jsp permite al usuario visualizar toda la información correspondiente a un determinado mantenimiento.

Código fuente



verSoporte.jsp

6.1.5. MODULO DE TRASLADOS

El módulo de traslados permite al usuario registrar todos los cambios de oficina, áreas y/o custodio de los diferentes tipos de recursos. Para realizar todas estas actividades el módulo consta de las siguientes páginas:

Página imprecurso.jsp

La página imprecurso.jsp imprime el reporte de los movimientos y traslados realizados con un determinado recurso.

Código fuente



imprecurso.jsp

Página ingMovimiento.jsp

La página ingMovimiento.jsp permite al usuario registrar un nuevo cambio de oficina, área y/o custodio de un determinado recurso.

Código fuente



ingMovimiento.jsp

Página movimientos.jsp

La página movimientos.jsp emite un listado de los traslados de oficina, área y/o custodio realizados en un determinado recurso..

Código fuente



movimientos.jsp

Página recursos.jsp

La página recursos.jsp permite al usuario visualizar un listado general de recursos clasificado por grupo, categoría y subcategoría de recursos.

Código fuente



recursos.jsp

6.1.6. MODULO DE REPORTE

El módulo de reportes permite al usuario emitir una variedad de reportes correspondiente a la información almacenada en el sistema, este módulo consta de diferentes páginas que ayudan a cumplir con este objetivo. Las páginas que encontramos en este módulo son:

Página imprecursos.jsp

La página imprecursos.jsp permite al usuario imprimir un reporte general detallado de recursos clasificado por grupo, categoría y subcategoría de recursos.

Código fuente



imprecursos.jsp

Página impresumen.jsp

La página impresumen.jsp permite al usuario imprimir un reporte general resumido de recursos clasificado por grupo, categoría, subcategoría y modelo de recurso.

Código fuente



impresumen.jsp

Página recurso.jsp

La página recurso.jsp permite al usuario visualizar toda la información correspondiente a las características, mantenimientos y traslados realizados con un determinado recurso.

Código fuente



recurso.jsp

Página recursos.jsp

La página recursos.jsp permite al usuario visualizar un listado general de recursos clasificado por grupo, categoría y subcategoría de recursos.

Código fuente

