

MANUAL TÉCNICO

1. INTRODUCCIÓN

El manual técnico esta enfocado al personal encargado de la administración del sistema aquí constaran los diagramas de clases, Entidad Relación, creación de objetos de la base de datos y Scripts de las tablas. Se mostrará además la codificación mas relevante del Modulo de Operaciones.

2. DIAGRAMAS

Aquí se mostraran las representaciones gráficas empleadas para el desarrollo de esta aplicación.

2.1 DIAGRAMAS DE CLASES

El diagrama de clases es una descripción de las clases en un sistema y sus relaciones. El primer elemento de un diagrama de clases es la descripción de las clases individuales.

Ver referencia Tomo 1 Capitulo 3.1(Modelo de clases generales)

Ver Figura 3.1 en Anexo B

2.2 DIAGRAMAS DE ENTIDAD RELACIÓN

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos.

El modelo entidad-relación está formado por un conjunto de conceptos que permiten describir la realidad mediante un conjunto de representaciones gráficas estableciendo las relaciones entre cada una de las entidades de la base de datos.

Ver Figura en Anexo D

3. CREACION DE OBJETOS DE LA BASE DE DATOS

3.1 SCRIPTS DE LAS TABLAS.

```
CREATE TABLE oper_calle_dat
(
  codi_calle_pk int8 NOT NULL,
  desc_calle varchar(30) NOT NULL,
  codi_ciud_fk int8,
  estd_reg_calle char(1) NOT NULL ,
  CONSTRAINT calle_pk PRIMARY KEY (codi_calle_pk),
  CONSTRAINT ciud_call_fk FOREIGN KEY (codi_ciud_fk) REFERENCES
  car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)
```

```
CREATE TABLE oper_cdla_dat
(
  codi_cdla_pk int8 NOT NULL,
  desc_cdla varchar(50) NOT NULL,
  codi_ciud_fk int8 NOT NULL,
  estd_reg_cdla char(1) NOT NULL,
  CONSTRAINT cdla_pk PRIMARY KEY (codi_cdla_pk),
  CONSTRAINT ciud_cdla_fk FOREIGN KEY (codi_ciud_fk) REFERENCES
  car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)
```

```

CREATE TABLE oper_ciud_entr
(
codi_ciud_fk int8,
estd_reg_cent char(1),
CONSTRAINT codi_ciud_fkey FOREIGN KEY (codi_ciud_fk) REFERENCES
car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_deta_entr_dat
(
codi_orden_entr_fk int8 NOT NULL,
item_deta_entr_pk int8 NOT NULL ,
date_orden_entr_fk date NOT NULL,
codi_detalle_factura_fk serial NOT NULL,
codi_factura_fk int8 NOT NULL,
codi_prod int8 NOT NULL,
cant_vend_deta_entr int8 NOT NULL,
cant_entr_deta_entr int8 NOT NULL DEFAULT,
cant_pend_deta_entr int8 NOT NULL DEFAULT,
estd_reg_deta_entr char(1),
date_ulti_modi timestamp,
CONSTRAINT detalle_entr_pk PRIMARY KEY (codi_orden_entr_fk,
date_orden_entr_fk, item_deta_entr_pk),
CONSTRAINT orden_entr_fk FOREIGN KEY (codi_orden_entr_fk,
date_orden_entr_fk) REFERENCES oper_orden_entr_dat (codi_orden_entr_pk,
date_orden_entr_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
CONSTRAINT ven_det_fact_fk FOREIGN KEY (codi_detalle_factura_fk,
codi_factura_fk) REFERENCES ven_detalle_factura (codi_detalle_factura,
codi_factura) ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_deta_serv_dat
(
codi_orden_serv_fk int8 NOT NULL ,
date_orden_serv_fk date NOT NULL,
item_deta_serv_pk int8 NOT NULLt,
codi_deta_orde_fk serial NOT NULL,
codi_prod int8 NOT NULL ,
cant_deta_serv int8 NOT NULL,
estd_reg varchar(1),
CONSTRAINT detalle_serv_pk PRIMARY KEY (codi_orden_serv_fk,
date_orden_serv_fk, item_deta_serv_pk),
CONSTRAINT cmp_deta_orden_fk FOREIGN KEY (codi_deta_orde_fk)
REFERENCES cmp_detalle_orden_dat (codi_deta_orde) ON UPDATE RESTRICT
ON DELETE RESTRICT,
CONSTRAINT orden_serv_fk FOREIGN KEY (codi_orden_serv_fk,
date_orden_serv_fk) REFERENCES oper_orden_serv_dat (codi_orden_serv_pk,
date_orden_serv_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_entr_terc_dat
(
  codi_terc_fk int8 NOT NULL,
  codi_orden_entr_fk int8 NOT NULL,
  date_orden_entr_fk date NOT NULL,
  date_entr_terc_pk date NOT NULL,
  estd_reg_entr_terc char(1),
  date_sald_entr_terc timestamp,
  CONSTRAINT entr_terc PRIMARY KEY (codi_terc_fk, codi_orden_entr_fk,
date_orden_entr_fk, date_entr_terc_pk),
  CONSTRAINT orden_entr_fk FOREIGN KEY (codi_orden_entr_fk,
date_orden_entr_fk) REFERENCES oper_orden_entr_dat (codi_orden_entr_pk,
date_orden_entr_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT terc_fk FOREIGN KEY (codi_terc_fk) REFERENCES oper_terc_dat
(codi_terc_pk) ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE oper_estd_dat
(
  codi_estd_pk int8 NOT NULL,
  desc_estd varchar(25) NOT NULL DEFAULT "":character varying,
  estd_regt_estd char(1) NOT NULL DEFAULT "":bpchar,
  CONSTRAINT estd_pk PRIMARY KEY (codi_estd_pk)
)

```

```

CREATE TABLE oper_motv_dat
(
  codi_motv_pk int8 NOT NULL,
  desc_motv varchar(50) NOT NULL DEFAULT "":character varying,
  tipo_motv varchar(10) NOT NULL DEFAULT "":character varying,
  estd_reg_motv char(1) NOT NULL DEFAULT "":bpchar,
  CONSTRAINT motv_pk PRIMARY KEY (codi_motv_pk)
)

```

```

CREATE TABLE oper_sect_dat
(
  codi_sect_pk int8 NOT NULL,
  codi_ciud_fk int8 NOT NULL,
  desc_sect varchar(50) NOT NULL,
  estd_reg_sect varchar(1),
  CONSTRAINT sect_pk PRIMARY KEY (codi_sect_pk),
  CONSTRAINT ciud_sect_fk FOREIGN KEY (codi_ciud_fk) REFERENCES
car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_prio_entr
(
  codi_prio_entr_pk int8 NOT NULL,
  desc_prio_entr varchar(25) NOT NULL DEFAULT "":character varying,
  estd_regt_prio_entr char(1) NOT NULL DEFAULT "":bpchar,
  CONSTRAINT prio_entr_pk PRIMARY KEY (codi_prio_entr_pk)
)

```

```

CREATE TABLE oper_orden_entr_dat
(
  codi_orden_entr_pk int8 NOT NULL,
  date_orden_entr_pk date NOT NULL,
  codi_fact_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_clie_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_estd_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_prio_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_sect_fk int8 NOT NULL DEFAULT (0)::bigint,
  scan_orden_entr char(1) DEFAULT "":bpchar,
  codi_motv_fk int8 DEFAULT (0)::bigint,
  codi_ciud_fk int8 NOT NULL DEFAULT (0)::bigint,
  imag_orden_entr varchar(50),
  codi_callep_fk int8,
  codi_callei_fk int8,
  codi_cdla_fk int8,
  num_calle varchar(5),
  comen_dir varchar(30),
  estd_reg_orden_entr char(1),
  num_entr int2,
  date_ulti_modi timestamp,
  CONSTRAINT orden_entr_pk PRIMARY KEY (codi_orden_entr_pk,
date_orden_entr_pk),
  CONSTRAINT ciud_orden_entr_fk FOREIGN KEY (codi_ciud_fk) REFERENCES
car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT clie_orden_entr_fk FOREIGN KEY (codi_clie_fk) REFERENCES
car_clie_dat (codi_clie_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT estd_orden_entr_fk FOREIGN KEY (codi_estd_fk) REFERENCES
oper_estd_dat (codi_estd_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT fact_orden_entr_fk FOREIGN KEY (codi_fact_fk) REFERENCES
ven_factura (codi_factura) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT motv_orden_entr_fk FOREIGN KEY (codi_motv_fk) REFERENCES
oper_motv_dat (codi_motv_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT prio_orden_entr_fk FOREIGN KEY (codi_prio_fk) REFERENCES
oper_prio_entr (codi_prio_entr_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT sect_orden_entr_fk FOREIGN KEY (codi_sect_fk) REFERENCES
oper_sect_dat (codi_sect_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)

CREATE TABLE oper_trans_dat
(
  codi_trans_pk int8 NOT NULL,
  desc_trans varchar(25),
  mode_trans varchar(25) NOT NULL DEFAULT "":character varying,
  placa_trans varchar(8) NOT NULL DEFAULT "":character varying,
  marca_trans varchar(20) NOT NULL DEFAULT "":character varying,
  color_trans varchar(20) NOT NULL DEFAULT "":character varying,
  capa_trans int8 NOT NULL DEFAULT (0)::bigint,
  estd_reg_trans char(1) NOT NULL DEFAULT "":bpchar,
  volu_trans int8,
  codi_empl_fk int4,
  dispon_trans varchar(1),
  CONSTRAINT trans_pk PRIMARY KEY (codi_trans_pk),
  CONSTRAINT trans_empl FOREIGN KEY (codi_empl_fk) REFERENCES
rh_empleado_dat (idempleado) ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_orden_serv_dat
(
  codi_orden_serv_pk int8 NOT NULL,
  date_orden_serv_pk date NOT NULL,
  codi_orden_comp_fk int8 NOT NULL,
  codi_prov_fk int8,
  codi_estd_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_ciud_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_sect_fk int8,
  dire_serv varchar(60),
  comen_dir varchar(60),
  codi_empr_fk int8 NOT NULL,
  estd_reg varchar(1),
  CONSTRAINT orden_serv_pk PRIMARY KEY (codi_orden_serv_pk,
  date_orden_serv_pk),
  CONSTRAINT ciud_orden_serv_fk FOREIGN KEY (codi_ciud_fk) REFERENCES
  car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT compras_servicio FOREIGN KEY (codi_orden_comp_fk,
  codi_empr_fk) REFERENCES cmp_orden_dat (codi_orde, codi_empr) ON UPDATE
  RESTRICT ON DELETE RESTRICT,
  CONSTRAINT estd_orden_serv_fk FOREIGN KEY (codi_estd_fk) REFERENCES
  oper_estd_dat (codi_estd_pk) ON UPDATE NO ACTION ON DELETE NO ACTION,
  CONSTRAINT sector_servicio FOREIGN KEY (codi_sect_fk) REFERENCES
  oper_sect_dat (codi_sect_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_terc_ciud_ref
(
  codi_terc_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_ciud_fk int8 NOT NULL DEFAULT (0)::bigint,
  CONSTRAINT terc_ciud_pk PRIMARY KEY (codi_terc_fk, codi_ciud_fk),
  CONSTRAINT ciud_fk FOREIGN KEY (codi_ciud_fk) REFERENCES car_ciud_dat
  (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT terc_fk FOREIGN KEY (codi_terc_fk) REFERENCES oper_terc_dat
  (codi_terc_pk) ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE oper_terc_dat
(
  codi_terc_pk int8 NOT NULL,
  desc_terc varchar(50) NOT NULL DEFAULT "":character varying,
  dire_terc varchar(50) NOT NULL DEFAULT "":character varying,
  telf_terc varchar(10),
  estd_reg_terc char(1),
  CONSTRAINT oper_terc_dat_pkey PRIMARY KEY (codi_terc_pk)
)

```

```

CREATE TABLE oper_orden_serv_inv_dat
(
  codi_orden_serv_inv_pk int8 NOT NULL,
  date_orden_serv_inv_pk date NOT NULL,
  codi_transfer_fk int8 NOT NULL,
  codi_empr_fk int8 NOT NULL,
  codi_estd_fk int8 NOT NULL DEFAULT (0)::bigint,
  codi_ciud_orig int8 NOT NULL DEFAULT (0)::bigint,
  dire_serv_desti varchar(60),
  dire_serv_orig varchar(60),
  sucur_orig int8 NOT NULL,
  sucur_dest int8 NOT NULL,
  bode_orig int8 NOT NULL,
  bode_desti int8 NOT NULL,
  codi_ciud_desti int8 NOT NULL,
  estd_reg varchar(1),
  CONSTRAINT orden_serv_inv_pk PRIMARY KEY (codi_orden_serv_inv_pk,
  date_orden_serv_inv_pk),
  CONSTRAINT ciud2_transfer FOREIGN KEY (codi_ciud_desti) REFERENCES
  car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT cuid1_transfer FOREIGN KEY (codi_ciud_orig) REFERENCES
  car_ciud_dat (codi_ciud_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT inv_serv_fk FOREIGN KEY (codi_transfer_fk, codi_empr_fk)
  REFERENCES inv_transaccion_log (codi_tran, codi_empr) ON UPDATE RESTRICT
  ON DELETE RESTRICT,
  CONSTRAINT sucur1_transfer FOREIGN KEY (sucur_orig, codi_empr_fk)
  REFERENCES inv_sucursal_dat (codi_sucu, codi_empr) ON UPDATE RESTRICT
  ON DELETE RESTRICT,
  CONSTRAINT sucur2_transfer FOREIGN KEY (sucur_dest, codi_empr_fk)
  REFERENCES inv_sucursal_dat (codi_sucu, codi_empr) ON UPDATE RESTRICT
  ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_trans_entr_dat
(
  codi_trans_fk int8 NOT NULL,
  codi_orden_entr_fk int8 NOT NULL,
  date_orden_entr_fk date NOT NULL,
  num_viaje_trans_entr int8 NOT NULL,
  time_stim_trans_entr varchar(10),
  peso_actual_trans int8 DEFAULT 0,
  hora_sald_trans_entr timestamp,
  volu_actual_trans int8 DEFAULT 0,
  estd_reg_trans_entr char(1) NOT NULL,
  CONSTRAINT orden_entr FOREIGN KEY (codi_orden_entr_fk, date_orden_entr_fk)
  REFERENCES oper_orden_entr_dat (codi_orden_entr_pk, date_orden_entr_pk) ON
  UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT trans_fk FOREIGN KEY (codi_trans_fk) REFERENCES
  oper_trans_dat (codi_trans_pk) ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE oper_deta_serv_inv_dat
(
  codi_orden_serv_inv_fk int8 NOT NULL DEFAULT (0)::bigint,
  date_orden_serv_inv_fk date NOT NULL,
  codi_empr_fk int8,
  item_deta_serv_inv_pk int8 NOT NULL DEFAULT (0)::bigint,
  codi_prod_fk int8 NOT NULL DEFAULT (0)::bigint,
  cant_deta_serv_inv int8 NOT NULL DEFAULT (0)::bigint,
  codi_deta_trans_fk int8,
  codi_trans_fk int8,
  estd_reg varchar(1),
  CONSTRAINT deta_serv_inv PRIMARY KEY (codi_orden_serv_inv_fk,
date_orden_serv_inv_fk, item_deta_serv_inv_pk),
  CONSTRAINT deta_inv_deta_transf FOREIGN KEY (codi_deta_trans_fk, codi_trans_fk,
codi_empr_fk) REFERENCES inv_detalle_transaccion_log (codi_deta, codi_tran, codi_empr)
ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT orde_serv_inv_deta FOREIGN KEY (codi_orden_serv_inv_fk,
date_orden_serv_inv_fk) REFERENCES oper_orden_serv_inv_dat (codi_orden_serv_inv_pk,
date_orden_serv_inv_pk) ON UPDATE RESTRICT ON DELETE RESTRICT
)

```

```

CREATE TABLE oper_trans_sect_dat
(
  codi_trans_fk int8 NOT NULL,
  codi_prio_sect int8 NOT NULL,
  codi_alte_sect int8 NOT NULL,
  CONSTRAINT trans_sect PRIMARY KEY (codi_trans_fk),
  CONSTRAINT sector_fk FOREIGN KEY (codi_prio_sect) REFERENCES
oper_sect_dat (codi_sect_pk) ON UPDATE RESTRICT ON DELETE RESTRICT,
  CONSTRAINT trans_fk FOREIGN KEY (codi_trans_fk) REFERENCES
oper_trans_dat (codi_trans_pk) ON UPDATE NO ACTION ON DELETE NO ACTION
)

```

```

CREATE TABLE oper_trans_serv_dat
(
  codi_trans_fk int8 NOT NULL,
  codi_orden_serv_fk int8 NOT NULL,
  date_orden_serv_fk date NOT NULL,
  date_trans_serv date,
  tipo_trans_serv varchar(1),
  estd_serv varchar(1),
  codi_trans_serv_pk int8 NOT NULL,
  CONSTRAINT trans_serv_pk PRIMARY KEY (codi_trans_serv_pk,
codi_orden_serv_fk, date_orden_serv_fk, codi_trans_fk),
  CONSTRAINT trans_fk FOREIGN KEY (codi_trans_fk) REFERENCES
oper_trans_dat (codi_trans_pk) ON UPDATE NO ACTION ON DELETE NO ACTION
)

```


4. DICCIONARIO DE DATOS

Un diccionario de datos es un catálogo, un depósito, de los elementos de un sistema. Estos elementos se centran alrededor de los datos y la forma en que están estructurados para satisfacer los requerimientos y las necesidades de la organización. En él se encuentran la lista de todos los elementos que forman parte del flujo de datos en todo el sistema.

Si los analistas desean conocer cuántos caracteres hay en un dato, con qué otros nombres se le conoce en el sistema, o en donde se utilizan dentro del sistema deben ser capaces de encontrar las respuesta en un diccionario de datos desarrollado apropiadamente.

4.1 Importancia del diccionario:

Los analistas usan los diccionarios de datos por 5 razones principales:

- Manejar los detalles en sistemas grandes
- Comunicar un significado común para todos los elementos del sistema
- Documentar las características del sistema
- Facilitar el análisis de los detalles con la finalidad de evaluar las características y determinar donde efectuar cambios en el sistema

- Localizar errores y omisiones en el sistema

NOMBRE: oper_calle_dat

DESCRIPCIÓN.-

Especifica toda la información referente a las calles según la ciudad respectiva a la que pertenecen.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_calle_pk	Int	8	No	Indica el código único de la calle.
desc_calle	Varchar	30	No	Indica el nombre de la calle.
codi_ciud_fk	Int	8	No	Indica el código de la ciudad a la que pertenece la calle.
estd_reg_calle	Varchar	1	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_cdla_dat

DESCRIPCIÓN.-

Especifica toda la información referente a las ciudadelas según la ciudad respectiva a la que pertenecen.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_cdla_pk	Int	8	No	Indica el código único de la ciudadela.
desc_cdla	Varchar	30	No	Indica el nombre de la ciudadela.
codi_ciud_fk	Int	8	No	Indica el código de la ciudad a la que pertenece la ciudadela.
estd_reg_cdla	Varchar	1	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_ciud_entr

DESCRIPCIÓN.-

Especifica toda la información referente a las ciudades de entrega y de servicio en las que trabajan nuestros transportes.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_ciud_fk	Int	8	No	Indica el código de la ciudad de entrega o servicio.
estd_reg_cent	Int	8	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_prio_entr

DESCRIPCIÓN.-

Especifica toda la información referente a las prioridades que manejamos en el proceso de entrega. Actualmente la prioridad con que trabajamos es por fecha de emisión de factura. Estas prioridades dependerán de las políticas de la empresa.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_prio_entr_pk	Int	8	No	Indica el código de la prioridad de entrega.
desc_prio_entr	Varchar	30	No	Indica la descripción de la prioridad.
estd_regt_prio_entr	Int	8	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_orden_entr_dat

DESCRIPCIÓN.-

Especifica toda la información referente a las órdenes de entrega generadas en base a facturas que han cumplido con los requerimientos necesarios para proceder a la ejecución del proceso de entrega.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_orden_entr_pk	Int	8	No	Indica el código único de la orden de entrega
date_orden_entr_pk	Int	8	No	Indica la fecha en que se genera la orden.
codi_fact_fk	Int	8	No	Indica el código de la factura.
codi_clie_fk	Int	8	No	Indica el código del cliente.
codi_estd_fk	Int	8	No	Indica el código de la prioridad.
codi_prio_fk	Int	8	No	Indica el código de la estado de la orden de entrega.
codi_sect_fk	Int	8	No	Indica el código del sector.
scan_orden_entr	Char	1	No	Indica si la orden ha sido scaneada (S) o no (N).
codi_motv_fk	Int	8	No	Indica el código del motivo.
codi_ciud_fk	Int	8	No	Indica el código de la ciudad.
imag_orden_entr	Varchar	50	Si	Indica la ruta donde ha sido guardada la imagen de la orden de entrega.
codi_callep_fk	Int	8	Si	Indica el código de la calle principal.
codi_callei_fk	Int	8	Si	Indica el código de la calle de intersección.
codi_cdla_fk	Int	8	Si	Indica el código de la ciudadela en caso que no tenga calles.
num_calle	Int	8	Si	Indica la numeración de la calle.
comen_dir	Varchar	30	Si	Indicaciones específicas de la dirección de entrega.
estd_reg_orden_entr	Char	1	No	Indica si el registro esta activo(A) o inactivo (I).
num_entr	Int	2	No	Indica las veces que se ha intentado la entrega, teniendo un máximo de tres intentos. Si se excede este valor la orden será dada de baja.
date_ulti_modi	TimeStamp		Si	Indica la última fecha en que se modificó la orden de entrega.

NOMBRE: oper_deta_entr_dat

DESCRIPCIÓN.-

Especifica toda la información referente a los detalles respetivos de las órdenes de entrega generadas en base a facturas que han cumplido con los requerimientos necesarios para proceder a la ejecución del proceso de entrega.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_orden_entr_fk	Int	8	No	Indica el código único de la orden de entrega
date_orden_entr_fk	Date		No	Indica la fecha en que se genera la orden.
codi_factura_fk	Int	8	No	Indica el código de la factura.
codi_detalle_factura_fk	Int	8	No	Indica el código del detalle de la factura.
codi_prod	Int	8	No	Indica el código del producto.
item_deta_entr_pk	Int	8	No	Indica el código del item.
cant_vend_deta_entr	Int	8	No	Indica las unidades a entregar.
cant_entr_deta_entr	Int	8	No	Indica las unidades entregadas.
cant_pend_deta_entr	Int	8	Si	Indica las unidades pendientes.
estd_reg_deta_entr	Char	1	No	Indica si el registro esta activo(A) o inactivo (I).
date_ulti_modi	TimeStamp		Si	Indica la última fecha en que se modificó la orden de entrega.

NOMBRE: oper_estd_dat

DESCRIPCIÓN.-

Especifica toda la información referente al estado en que se encuentra la orden estos pueden ser pendiente, entregado, por courier, etc

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_estd_pk	Int	8	No	Indica el código del estado de la orden.
desc_estd	Varchar	25	No	Indica la descripción del estado.
estd_regt_estd	Int	8	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_deta_serv_dat

DESCRIPCIÓN.-

Especifica toda la información referente a los detalles respectivos de las órdenes de servicio emitidas por compras que han cumplido con todos los requerimientos necesarios para proceder a la ejecución del proceso de prestación de servicio.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_orden_serv_fk	Int	8	No	Indica el código único de la orden de servicio
date_orden_serv_fk	date	25	No	Indica la fecha en que se genera la orden se servicio.
item_deta_serv_pk	Int	8	No	Indica el código del item.
codi_deta_orde_fk	int			Indica el código del detalle de la compra.
codi_prod	int			Indica el código del producto
cant_deta_serv	int			Indica las unidades a retirar.
estd_reg	char	1	no	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_orden_serv_dat

DESCRIPCIÓN.-

Especifica toda la información referente a las órdenes de servicio emitidas por compras que han cumplido con todos los requerimientos necesarios para proceder a la ejecución del proceso de prestación de servicio.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_orden_serv_pk	Int	8	No	Indica el código único de la orden de servicio
date_orden_serv_pk	date		No	Indica la fecha en que se genera la orden de servicio.
codi_orden_comp_fk	Int	8	No	Indica el código del la orden de compra.
codi_prov_fk	int	8	No	Indica el código del proveedor.
codi_estd_fk	int	8	No	Indica el código del estado de la orden de servicio.
codi_ciud_fk	int	8	No	Indica el código de la ciudad
codi_sect_fk	Int	8	No	Indica el código del sector
dire_serv	varchar	60	No	Indica la dirección de prestación de servicio
comen_dir	varchar	60	Si	Indicaciones específicas de la dirección de prestación de servicio.
codi_empr_fk			No	Indica el código de la empresa
estd_reg	char	1	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_entr_terc_dat

DESCRIPCIÓN.-

Especifica toda la información referente a las órdenes de entrega que serán efectuadas por medio de las tercerizadoras.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_terc_fk	Int	8	No	Indica el código único de la terceriadora
codi_orden_entr_fk	Int	8	No	Indica el código de la orden.
date_orden_entr_fk	Date		No	Indica la fecha en que se genero la orden de entrega.
date_entr_terc_pk	Int	8	No	Indica la fecha en que se genera el registro.
estd_reg_entr_terc	Char	1	No	Indica si el registro esta activo(A) o inactivo (I).
date_sald_entr_terc	timestamp	8	No	Indica la fecha en que se despacho la orden

NOMBRE: oper_trans_sect_dat

DESCRIPCIÓN.-

Especifica toda la información referente a los transportes a quienes se les ha establecido un sector alternativo y un sector prioritario para la entrega o prestación de servicios.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_trans_fk	Int	8	No	Indica el código del transporte.
codi_prio_sect	Int	8	No	Indica el código del sector prioritario asignado al transporte.
codi_alte_sect	Int	8	No	Indica el código del sector alternativo asignado al transporte.

NOMBRE: oper_orden_serv_inv_dat**DESCRIPCIÓN.-**

Especifica toda la información referente a las órdenes de servicio solicitadas por concepto de transferencias de productos (Inventario) las cuales han cumplido con todos los requerimientos necesarios para proceder a la ejecución del proceso de prestación de servicio.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_orden_serv_inv_pk	Int	8	No	Indica el código único de la orden de servicio de transferencia
date_orden_serv_inv_pk	Date		No	Indica la fecha en que se genera la solicitud de transferencia
codi_transfer_fk	Int	8	No	Indica el código de solicitud de transferencia.
codi_empr_fk	Int	8	No	Indica el código de la empresa.
codi_estd_fk	Int	8	No	Indica el código del estado de la orden.
codi_ciud_orig	Int	8	No	Indica el código de la ciudad de origen
dire_serv_desti	Varchar	60	No	Indica al dirección destino.
dire_serv_orig	Varchar	60	No	Indica al dirección origen
sucur_orig	Int	8	No	Indica el código de Suc. Origen.
sucur_dest	Int	8	No	Indica el código de Suc. Destino.
bode_orig	Int	8	No	Indica el codigo de bodega origen.
bode_desti	Int	8	No	Indica el codigo de bodega destino
codi_ciud_desti	Int	8	No	Indica el código de la ciudad destino
estd_reg	char	1	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_sect_dat

DESCRIPCIÓN.-

Especifica toda la información referente a los sectores en los que se ha dividido la ciudad para el proceso de entrega a domicilio así como también para la prestación del servicio de transporte.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
oper_sect_dat	Int	8	No	Indica el código único del sector.
codi_ciud_fk	Int	8	No	Indica el código de la ciudad.
desc_sect	Varchar	30	No	Indica la descripción del sector.
estd_reg_sect	Char	1	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_terc_dat

DESCRIPCIÓN.-

Especifica toda la información referente a los sectores en los que se ha dividido la ciudad para el proceso de entrega a domicilio así como también para la prestación de transporte.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_terc_pk	Int	8	No	Indica el código único de la tercerizadora.
desc_terc	Varchar	50	No	Indica el la descripción de la tercerizadora.
dire_terc	Varchar	50	No	Indica la dirección de la tercerizadora.
telf_terc	Varchar	10	Si	Indica el teléfono de la tercerizadora.
estd_reg_terc	char	1	No	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_trans_dat

DESCRIPCIÓN.-

Especifica toda la información referente a los transportes con los que cuenta la empresa para la realización de entrega y prestación de servicio.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_trans_pk	Int	8	No	Indica el código único del transporte.
desc_trans	Varchar	25	No	Indica el la descripción del transporte.
mode_trans	Varchar	25	No	Indica el modelo del transporte.
placa_trans	Varchar	8	Si	Indica la placa del transporte
marca_trans	Varchar	20	No	Indica la marca del transporte
color_trans	Varchar	20	No	Indica el color del Transporte
capa_trans	Int	8	No	Indica la capacidad del transporte
estd_reg_trans	char	1	No	Indica si el registro esta activo(A) o inactivo (I).
volu_trans	int	8	No	Indica el Volumen del Transporte
codi_empl_fk	int	4	No	Indica el código del chofer asignado al transporte
dispon_trans	char	1	No	Indica si el transporte esta disponible (D) caso contrario (O)

NOMBRE: oper_motv_dat

DESCRIPCIÓN.-

Especifica toda la información referente a los motivos que se puedan presentar al momento de la entrega o prestación. Como por ejemplo carro dañado, estado defectuoso, etc.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_motv_pk	Int	8	No	Indica el código único del motivo
desc_motv	Varchar	50	No	Indica el la descripción del motivo.
tipo_motv	Varchar	50	No	Indica el tipo de motivo sea servicio o entrega.
estd_reg_motv	Varchar	10	Si	Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_terc_ciud_ref

DESCRIPCIÓN.-

Especifica toda la información referente a las ciudades que cubren o abarcan las tercerizadoras con las cuales trabajamos para facilitar el proceso de entrega en distintos lugares del país.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_terc_fk	Int	8	No	Indica el código único del la tercerizadora
codi_ciud_fk	Int	8	No	Indica el código de la ciudad

NOMBRE: oper_trans_entr_dat

DESCRIPCIÓN.-

Especifica toda la información referente al proceso de entrega. Esta entidad enlaza o asocia el transporte con la orden de entrega que se le haya asignado.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_trans_fk	Int	8	No	Indica el código del transporte asignado a la orden de entrega.
codi_orden_entr_fk	Int	8	No	Indica el código de la ciudad.
date_orden_entr_fk	Date		No	Indica la fecha en que se generó la orden de entrega.
num_viaje_trans_entr	Int	8		Indica el número de viaje que ha realizado el transporte en el día.
peso_actual_trans	Int	8		Indica peso actual del transporte.
hora_sald_trans_entr	Timestamp		No	Indica la fecha y hora de salida del transporte.
volu_actual_trans	Int	8	No	Indica el volumen actual del transporte.
estd_reg_trans_entr	Char	1		Indica si el registro esta activo(A) o inactivo (I).

NOMBRE: oper_trans_serv_dat

DESCRIPCIÓN.-

Especifica toda la información referente al proceso de prestación de servicio (por compras o por transferencias). Enlaza o asocia a un transporte determinado con la orden de servicio respectiva según las necesidades del caso.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_trans_fk	Int	8	No	Indica el código del transporte asignado a la orden de entrega.
codi_orden_serv_fk	Int	8	No	Indica el código de la orden de servicio por compra o por transferencia..
date_orden_serv_fk	Date		No	Indica la fecha en que se generó la orden.
date_trans_serv	TimeStamp		SI	Indica la fecha y hora de salida del transporte.
tipo_trans_serv	Varchar	1	No	Indica si el registro es por compra (C) o por transferencia (T).
estd_serv	Timestamp		No	Indica si el registro esta activo(A) o inactivo (I).
codi_trans_serv_pk	Int	8	No	Indica el código único del registro.

NOMBRE: oper_deta_serv_inv_dat

DESCRIPCIÓN.-

Especifica toda la información referente al detalle de la orden de transferencia (solicitada por inventario) para la prestación del servicio de transportación.

Nombre	Tipo de dato	Longitud	Nulo	Descripción
codi_orden_serv_inv_fk	Int	8	No	Indica el código único de la orden de servicio de transferencia
date_orden_serv_inv_fk	Date		No	Indica la fecha en que se genera la solicitud de transferencia
codi_empr_fk	Int	8	No	Indica el código de la empresa.
item_deta_serv_inv_pk	Int	8	No	Indica el código del item del detalle.
codi_prod_fk	Int	8	No	Indica el código del producto.
cant_deta_serv_inv	Int	8	No	Indica la cantidad del producto a ser transferida.
codi_deta_trans_fk	Int	8	No	Indica el código del detalle de la transferencia.
codi_trans_fk	Int	8	No	Indica el código de la transferencia.
estd_reg	Varchar	1	No	Indica si el registro esta activo(A) o inactivo (I).

5. CODIFICACIÓN RELEVANTE

grabarOrdenEntrega() .- Método que nos permite realizar la generación de ordenes de entrega a domicilio que se efectúan en base a las facturas que emite Ventas las cuales deben cumplir con los requerimientos necesarios para que se formalice la entrega de la mercadería. Así como también la asignación del respectivo transporte según el sector de entrega, la capacidad y disponibilidad del mismo. Una vez que se haya encontrado el transporte indicado se procederá a la generación de la orden de servicio así como también a la asignación de orden de entrega al transporte indicado.

```

public boolean grabarOrdenEntrega(RegistroEntregaForm entrega,ArrayList listUniEntre) throws
createException {
    PreparedStatement ps = null;
    PreparedStatement ps1 = null;
    Date fecha = new Date();
    //SimpleDateFormat fechaConFormato1 = new SimpleDateFormat("d/MMMM/yyyy");
    SimpleDateFormat fechaConFormato = new SimpleDateFormat("yyyy-MM-dd");
    boolean descriptor = false;

    try {

if (con.isClosed()) {
    throw new IllegalStateException("error.unexpected");
}
if (entrega.getEntregaDirecta() == 1){
    PreparedStatement ps13 = con.prepareStatement("SELECT num_entr
FROM oper_orden_entr_dat WHERE codi_orden_entr_pk = ?;");
    ps13.setInt(1,entrega.getIdEntrega());
    ResultSet rs = ps13.executeQuery();
    int numEntre = 1;
    if (rs.next()){
        numEntre = rs.getInt(1);
    }
    String sql = "INSERT INTO oper_orden_entr_dat (codi_orden_entr_pk,
date_orden_entr_pk, codi_fact_fk, codi_clie_fk, codi_estd_fk,+
"codi_prio_fk,
codi_sect_fk,codi_motv_fk,codi_ciud_fk,codi_callep_fk,codi_callei_fk,codi_c
dla_fk,num_calle,comen_dir,num_entr,estd_reg_orden_entr) "+
"VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?);";
    ps = con.prepareStatement(sql);
    ps.setInt(1,entrega.getIdEntrega());
    ps.setString(2,fechaConFormato.format(fecha));
    ps.setInt(3,entrega.getIdFactura());
    ps.setInt(4,entrega.getIdCliente());
    ps.setInt(5,3);//estado
    ps.setInt(6,1);//prioridad
    ps.setInt(7,entrega.getSector());
    ps.setInt(8,entrega.getMotivo());
    ps.setInt(9,entrega.getCiudad());
    ps.setInt(10,entrega.getCallep());
    ps.setInt(11,entrega.getCallei());
    ps.setInt(12,entrega.getCdla());
    ps.setInt(13,entrega.getNumCalle());
    ps.setString(14,entrega.getComentDir());
    ps.setInt(15,numEntre);
    ps.setString(16,"A");
    if (ps.executeUpdate() != 1) {
        throw new CreateException("error.create.anuncio");
    }
}
}

```



```

String sql4 = "SELECT codi_detalle_factura,c.codi_arti,desc_arti,peso_brut_arti,
volu_arti,b.cantidad,b.precio,b.total "+
"FROM ven_factura a,ven_detalle_factura b,inv_articulo_dat c "+
"WHERE a.codi_factura = b.codi_factura AND b.codi_arti = c.codi_arti AND
a.codi_factura = ?;";

PreparedStatement ps7 = con.prepareStatement(sql4);
ps7.setInt(1,entrega.getIdFactura());
ResultSet rs3 = ps7.executeQuery();

while (rs3.next()){
    PreparedStatement ps2 = con.prepareStatement("INSERT INTO
oper_deta_entr_dat
(codi_orden_entr_fk,item_deta_entr_pk,date_orden_entr_fk," +
detalle_factura_fk,coodi_factura_fk,coodi_prod,cant_vend_deta_entr,"+
cant_entr_deta_entr,cant_pend_deta_entr,estd_reg_deta_entr)" +
"VALUES(?,?,?,?,?,?,?,?);");

    int uniEntregadas = rs3.getInt(6);
    int idDetaFact = rs3.getInt(1);
    if (listUniEntre!=null){
        Iterator iter = listUniEntre.iterator();
        while (iter.hasNext()) {
            Factura fact = (Factura)iter.next();

            if (fact.getIdDetaFactura() == idDetaFact) {
                uniEntregadas = fact.getUniEntregadas();
                break;
            }
        }
    }
    int UniPendientes = (rs3.getInt(6) - uniEntregadas);
    if (UniPendientes > 0){
        PreparedStatement ps14 = con.prepareStatement("UPDATE oper_orden_entr_dat
SET codi_estd_fk = 4,codi_motv_fk = 4 WHERE codi_orden_entr_pk = ?;");
        ps14.setInt(1,entrega.getIdEntrega());
        ps14.executeUpdate();
    }
    ps2.setInt(1,entrega.getIdEntrega());
    ps2.setInt(2,idDetalleOrdenEntrega());
    ps2.setString(3,fechaConFormato.format(fecha));
    ps2.setInt(4,rs3.getInt(1));
    ps2.setInt(5,entrega.getIdFactura());
    ps2.setInt(6,rs3.getInt(2));
    ps2.setInt(7,rs3.getInt(6));
    ps2.setInt(8,uniEntregadas);
    ps2.setInt(9,UniPendientes);
    ps2.setString(10,"A");
    ps2.executeUpdate();
}
PreparedStatement ps3 = null;
ps3 = con.prepareStatement("UPDATE ven_factura SET estado = 'G' WHERE
codi_factura = ?;");
ps3.setInt(1,entrega.getIdFactura());
ps3.executeUpdate();
}else{
    String sql1 = "SELECT
codi_detalle_factura,c.codi_arti,desc_arti,peso_brut_arti,volu_arti,b.cantidad,b.pr
ecio,b.total FROM ven_factura a,ven_detalle_factura b,inv_articulo_dat c "+
"WHERE a.codi_factura = b.codi_factura AND b.codi_arti = c.codi_arti AND
a.codi_factura = ?;";
    ps1 = con.prepareStatement(sql1);
    ps1.setInt(1,entrega.getIdFactura());
    ResultSet rs = ps1.executeQuery();
}

```

```

int pesoTotalFact = 0;
int volumenTotalFact = 0;
while (rs.next()){
    pesoTotalFact = pesoTotalFact + (rs.getInt(4) * rs.getInt(6));
    volumenTotalFact = volumenTotalFact + (rs.getInt(5) *
rs.getInt(6));
}

String sql2 = "SELECT codi_trans_pk,desc_trans,codi_prio_sect,codi_alte_sect,
"capa_trans,volu_trans " +
"FROM oper_trans_sect_dat a, oper_trans_dat b " +
"WHERE a.codi_trans_fk = b.codi_trans_pk AND b.estd_reg_trans = 'A'";
Statement ps4 = con.createStatement();
ResultSet rs1 = ps4.executeQuery(sql2);
int codTransPrio = 0;
int pesoTransPrio = 0;
int voluTransPrio = 0;
int codTransAlte = 0;
int pesoTransAlte = 0;
int voluTransAlte = 0;

while (rs1.next()){
    if (entrega.getSector() == rs1.getInt(3)){
        codTransPrio = rs1.getInt(1);
        pesoTransPrio = rs1.getInt(5);
        voluTransPrio = rs1.getInt(6);
    }
    if (entrega.getSector() == rs1.getInt(4)){
        codTransAlte = rs1.getInt(1);
        pesoTransAlte = rs1.getInt(5);
        voluTransAlte = rs1.getInt(6);
    }
}
String sql3 = "SELECT codi_trans_fk,peso_actual_trans,
volu_actual_trans,num_viaje_trans_entr "+
"FROM oper_trans_entr_dat a, oper_trans_dat b "+
"WHERE a.codi_trans_fk = b.codi_trans_pk AND estd_reg_trans_entr='A'
AND codi_trans_pk = ?";
PreparedStatement ps5 = con.prepareStatement(sql3);
ps5.setInt(1,codTransPrio);
ResultSet rs2 = ps5.executeQuery();
//int idTrans = 0;
int pesoActual = 0;
int volumenActual = 0;
int idViajes = 0;
while (rs2.next()){
    //idTrans = rs2.getInt(1);
    pesoActual = rs2.getInt(2);
    volumenActual = rs2.getInt(3);
    idViajes = rs2.getInt(4);
}
pesoActual = pesoTotalFact + pesoActual;
volumenActual = volumenTotalFact + volumenActual;
if (pesoActual <= pesoTransPrio && volumenActual <= voluTransPrio){

    PreparedStatement ps13 = con.prepareStatement("SELECT num_entr FROM
oper_orden_entr_dat WHERE codi_orden_entr_pk = ?");
ps13.setInt(1,entrega.getIdEntrega());
ResultSet rs6 = ps13.executeQuery();
int numEntre = 1;
if (rs6.next()){
    numEntre = rs6.getInt(1);
}
}

```

```

String sql = "INSERT INTO oper_orden_entr_dat (codi_orden_entr_pk, date_orden_entr_pk,
codi_fact_fk, codi_clie_fk, codi_estd_fk, ccdi_prio_fk, codi_sect_fk,
codi_motv_fk, codi_ciud_fk, codi_callep_fk, codi_callei_fk, codi_cdla_fk, num_calle,
comen_dir, num_entr, estd_reg_orden_entr) VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?);";
ps = con.prepareStatement(sql);
ps.setInt(1, entrega.getIdEntrega());
ps.setString(2, fechaConFormato.format(fecha));
ps.setInt(3, entrega.getIdFactura());
ps.setInt(4, entrega.getIdCliente());
ps.setInt(5, entrega.getEstado()); //estado
ps.setInt(6, 1); //prioridad
ps.setInt(7, entrega.getSector());
ps.setInt(8, entrega.getMotivo()); //Motivo
ps.setInt(9, entrega.getCiudad());
ps.setInt(10, entrega.getCallep());
ps.setInt(11, entrega.getCallei());
ps.setInt(12, entrega.getCdla());
ps.setInt(13, entrega.getNumCalle());
ps.setString(14, entrega.getComentDir());
ps.setInt(15, numEntre);
ps.setString(16, "A");

if (ps.executeUpdate() != 1) {
    throw new CreateException("error.create.anuncio");
}
String sql4 = "SELECT codi_detalle_factura, c.codi_arti, desc_arti,
peso_brut_arti, volu_arti, b.cantidad, b.precio, b.total "+
"FROM ven_factura a, ven_detalle_factura b, inv_articulo_dat c "+
"WHERE a.codi_factura = b.codi_factura AND b.codi_arti = c.codi_arti AND
a.codi_factura = ?;";
PreparedStatement ps7 = con.prepareStatement(sql4);
ps7.setInt(1, entrega.getIdFactura());
ResultSet rs3 = ps7.executeQuery();
while (rs3.next()){
    PreparedStatement ps2 = con.prepareStatement("INSERT INTO oper_deta_entr_dat
(codi_orden_entr_fk, item_deta_entr_pk, date_orden_entr_fk, "+
"codi_detalle_factura_fk, codi_factura_fk, codi_prod, cant_vend_deta_entr, cant_entr_deta_entr,
cant_pend_deta_entr, estd_reg_deta_entr) VALUES(?,?,?,?,?,?,?,?,?,?);");

    int uniEntregadas = rs3.getInt(6);
    int idDetaFact = rs3.getInt(1);
    if (listUniEntre != null){
        Iterator iter = listUniEntre.iterator();
        While (iter.hasNext()) {
            Factura fact = (Factura)iter.next();
            if (fact.getIdDetaFactura() == idDetaFact) {
                uniEntregadas =
                    fact.getUniEntregadas();
                break;
            }
        }
    }
    int UniPendientes = (rs3.getInt(6) - uniEntregadas);
    if (UniPendientes > 0) {
        PreparedStatement ps14 = con.prepareStatement("UPDATE
oper_orden_entr_dat SET codi_estd_fk = 4, codi_motv_fk = 4
WHERE codi_orden_entr_pk = ?;");
        ps14.setInt(1, entrega.getIdEntrega());
        ps14.executeUpdate();
    }
}

```

```

        ps2.setInt(1, entrega.getIdEntrega());
        ps2.setInt(2, idDetalleOrdenEntrega());
        ps2.setString(3, fechaConFormato.format(fecha));
        ps2.setInt(4, rs3.getInt(1));
        ps2.setInt(5, entrega.getIdFactura());
        ps2.setInt(6, rs3.getInt(2));
        ps2.setInt(7, rs3.getInt(6));
        ps2.setInt(8, uniEntregadas);
        ps2.setInt(9, UniPendientes);
        ps2.setString(10, "A");
        ps2.executeUpdate();
    }
    PreparedStatement ps3 = null;
    ps3 = con.prepareStatement("UPDATE ven_factura SET estado = 'G' WHERE
codi_factura = ?;");
    ps3.setInt(1, entrega.getIdFactura());
    ps3.executeUpdate();
    PreparedStatement ps6 = con.prepareStatement("INSERT INTO
oper_trans_entr_dat(codi_trans_fk, codi_orden_entr_fk, " +
"date_orden_entr_fk, num_viaje_trans_entr, peso_actual_trans, volu_actual_trans, estd_reg_tran
s_entr) VALUES(?,?,?,?,?,?);");
        ps6.setInt(1, codTransPrio);
        ps6.setInt(2, entrega.getIdEntrega());
        ps6.setString(3, fechaConFormato.format(fecha));
        ps6.setInt(4, (idViajes + 1));
        ps6.setInt(5, pesoActual);
        ps6.setInt(6, volumenActual);
        ps6.setString(7, "A");
        ps6.executeUpdate();
    PreparedStatement ps10 = null;
    ps10 = con.prepareStatement("UPDATE oper_trans_dat SET dispon_trans = 'O' WHERE
codi_trans_pk = ?;");
    ps10.setInt(1, codTransPrio);
    ps10.executeUpdate();

    }else{
    String sql5 = "SELECT codi_trans_fk, peso_actual_trans,
volu_actual_trans, num_viaje_trans_entr FROM oper_trans_entr_dat a,
oper_trans_dat b WHERE a.codi_trans_fk = b.codi_trans_pk AND
estd_reg_trans_entr='A' AND codi_trans_pk = ?;";
    PreparedStatement ps8 = con.prepareStatement(sql5);
    ps8.setInt(1, codTransAlte);
    ResultSet rs4 = ps8.executeQuery();
    int pesoActualAlte = 0;
    int voluActualAlte = 0;
    int idViajesAlte = 0;
    while (rs4.next()){
        pesoActualAlte = rs4.getInt(2);
        voluActualAlte = rs4.getInt(3);
        idViajesAlte = rs4.getInt(4);
    }
}

pesoActualAlte = pesoTotalFact + pesoActualAlte;
voluActualAlte = volumenTotalFact + voluActualAlte;

```

```

if (pesoActualAlte <= pesoTransAlte && voluActualAlte <= voluTransAlte){

        PreparedStatement ps15 = con.prepareStatement("SELECT num_entr
FROM oper_orden_entr_dat WHERE codi_orden_entr_pk = ?;");
        ps15.setInt(1, entrega.getIdEntrega());
        ResultSet rs6 = ps15.executeQuery();
        int numEntre = 1;
        if (rs6.next()){
            numEntre = rs6.getInt(1);
        }

        String sql6 = "INSERT INTO oper_orden_entr_dat
(codi_orden_entr_pk, date_orden_entr_pk, codi_fact_fk, codi_clie_fk, codi_estd_fk,"+
        "codi_prio_fk,
codi_sect_fk,codi_motv_fk,codi_ciud_fk,codi_callep_fk,codi_callei_fk,codi_cdla_fk,num_calle,comen_
dir,num_entr,estd_reg_orden_entr) "+
        "VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?,?)";

        PreparedStatement ps9 = con.prepareStatement(sql6);

        ps9.setInt(1, entrega.getIdEntrega());
        ps9.setString(2, fechaConFormato.format(fecha));
        ps9.setInt(3, entrega.getIdFactura());
        ps9.setInt(4, entrega.getIdCliente());
        ps9.setInt(5, entrega.getEstado());
        ps9.setInt(6, 1); //prioridad
        ps9.setInt(7, entrega.getSector());
        ps9.setInt(8, entrega.getMotivo());
        ps9.setInt(9, entrega.getCiudad());
        ps9.setInt(10, entrega.getCallep());
        ps9.setInt(11, entrega.getCallei());
        ps9.setInt(12, entrega.getCdla());
        ps9.setInt(13, entrega.getNumCalle());
        ps9.setString(14, entrega.getComentDir());
        ps9.setInt(15, numEntre);
        ps9.setString(16, "A");
        ps9.executeUpdate();

String sql7 = "SELECT codi_detalle_factura,c.codi_arti,desc_arti,peso_brut_arti,
"volu_arti,b.cantidad,b.precio,b.total FROM ven_factura a,ven_detalle_factura b,
"inv_articulo_dat c WHERE a.codi_factura = b.codi_factura "
"AND b.codi_arti = c.codi_arti AND a.codi_factura = ?;";
PreparedStatement ps10 = con.prepareStatement(sql7);
ps10.setInt(1, entrega.getIdFactura());
ResultSet rs5 = ps10.executeQuery();

        while (rs5.next()){
            PreparedStatement ps11 =
con.prepareStatement("INSERT INTO oper_deta_entr_dat
(codi_orden_entr_fk,item_deta_entr_pk,date_orden_entr_fk," +
        "codi_detalle_factura_fk,codi_factura_fk,codi_prod,cant_vend_deta_entr,cant_entr_deta_entr,
"cant_pend_deta_entr,estd_reg_deta_entr) VALUES(?,?,?,?,?,?,?,?,?)");
            int uniEntregadas = rs5.getInt(6);
            int idDetaFact = rs5.getInt(1);
            if (listUniEntre!=null){
                Iterator iter = listUniEntre.iterator();
                while (iter.hasNext()) {
                    Factura fact = (Factura)iter.next();
                    if (fact.getIdDetaFactura() == idDetaFact) {
                        uniEntregadas = fact.getUniEntregadas();
                        break;
                    }
                }
            }
        }
        int UniPendientes = (rs5.getInt(6) - uniEntregadas);

```

```

if (UniPendientes > 0 ){
    PreparedStatement ps14 = con.prepareStatement("UPDATE oper_orden_entr_dat SET
        codi_estd_fk = 4,codi_motv_fk = 4 WHERE codi_orden_entr_pk = ?;");
    ps14.setInt(1,entrega.getIdEntrega());
    ps14.executeUpdate();
}
ps11.setInt(1,entrega.getIdEntrega());
ps11.setInt(2,idDetalleOrdenEntrega());
ps11.setString(3,fechaConFormato.format(fechar));
ps11.setInt(4,rs5.getInt(1));
ps11.setInt(5,entrega.getIdFactura());
ps11.setInt(6,rs5.getInt(2));
ps11.setInt(7,rs5.getInt(6));
ps11.setInt(8,uniEntregadas);
ps11.setInt(9,UniPendientes);
ps11.setString(10,"A");
ps11.executeUpdate();
}
PreparedStatement ps12 = null;
ps12 = con.prepareStatement("UPDATE ven_factura SET estado = 'G' WHERE codi_factura = ?;");
ps12.setInt(1,entrega.getIdFactura());
ps12.executeUpdate();
PreparedStatement ps13 = con.prepareStatement("INSERT INTO
oper_trans_entr_dat(codi_trans_fk,codi_orden_entr_fk," +
"date_orden_entr_fk,num_viaje_trans_entr,peso_actual_trans,volu_actual_trans,estd_reg_trans_entr)
VALUES(?,?,?,?,?,?);");
ps13.setInt(1,codTransAlte);
ps13.setInt(2,entrega.getIdEntrega());
ps13.setString(3,fechaConFormato.format(fechar));
ps13.setInt(4,(idViajesAlte + 1));
ps13.setInt(5,pesoActualAlte);
ps13.setInt(6,voluActualAlte);
ps13.setString(7,"A");
ps13.executeUpdate();

PreparedStatement ps11 = null;
ps11 = con.prepareStatement("UPDATE oper_trans_dat SET dispon_trans = 'O' WHERE
codi_trans_pk = ?;");
ps11.setInt(1,codTransAlte);
ps11.executeUpdate();

}else{
    descriptor = true;
}
}
return descriptor;
} catch (SQLException e) {
    e.printStackTrace();
    throw new RuntimeException(e.getMessage());
} finally {
    try {
        if (ps != null)
            ps.close();
        if (con != null)
            con.close();
    } catch (SQLException e) {
        e.printStackTrace();
        throw new RuntimeException(e.getMessage());
    }
}
}
}

```

grabarServicioComp () .- Método que nos permite efectuar la emisión de ordenes de servicio que se generan en base a **ordenes de compra** que dicho departamento (Compras) efectúa y solicita la prestación de nuestros servicios de transportación para el traslado de la mercadería adquirida a los diversos proveedores hasta las bodegas de la empresa. Todo este proceso parte desde la selección de un transporte adecuado que cumpla con los requerimientos de la solicitud de servicio como son: disponibilidad inmediata del transporte, capacidad (peso) y espacio físico (volumen). Una vez que se haya encontrado el transporte indicado se procederá a la generación de la orden de servicio así como también a la asignación de orden al transporte elegido.

```
public boolean grabarServicioComp(RegistroServicioForm servicio) throws CreateException {
    PreparedStatement ps = null;
    PreparedStatement ps1 = null;
    Date fecha = new Date();
    ArrayList listtransportes = new ArrayList();
    int codTrans;
    float pesoTrans;
    float voluTrans;
    int sectorPrio;
    int sectorAlt;
    Transporte transporte=new Transporte();
    SimpleDateFormat fechaConFormato = new SimpleDateFormat("yyyy-MM-dd");
    boolean descriptor = false;

    try {

        if (con.isClosed()) {
            throw new IllegalStateException("error.unexpected");
        }
        String sql1 = "SELECT
codi_deta_orde,c.codi_arti,c.desc_arti,peso_brut_arti,volu_arti,b.cant_arti "+
"FROM cmp_orden1_dat a,cmp_detalle_orden_dat b,inv_articulo_dat c "+
"WHERE a.codi_orde = b.codi_orde AND b.codi_arti = c.codi_arti AND a.codi_orde = ?";

        ps1 = con.prepareStatement(sql1);
        ps1.setInt(1,servicio.getIdOrdenComp());
        ResultSet rs = ps1.executeQuery();
        int pesoTotalCmp = 0;
        int volumenTotalCmp = 0;
```

```

while (rs.next()){
pesoTotalCmp = pesoTotalCmp + (rs.getInt(4) * rs.getInt(6));
volumenTotalCmp = volumenTotalCmp + (rs.getInt(5) * rs.getInt(6));
}

String sql2 = "SELECT
codi_trans_pk,desc_trans,codi_prio_sect,codi_alte_sect,capa_trans,volu_trans " +
"FROM oper_trans_sect_dat a, oper_trans_dat b " +
"WHERE a.codi_trans_fk = b.codi_trans_pk AND b.estd_reg_trans = 'A' AND
dispon_trans='D'";
Statement ps3 = con.createStatement();
ResultSet rs1 = ps3.executeQuery(sql2);

while (rs1.next()){
pesoTrans=rs1.getFloat(5);
voluTrans=rs1.getFloat(6);
sectorPrio=rs1.getInt(3);
sectorAlt=rs1.getInt(4);

if ((servicio.getSector() == sectorPrio) ||(servicio.getSector() == sectorAlt) &&
(pesoTrans>=pesoTotalCmp)&&(voluTrans>=volumenTotalCmp)){
transporte.setldtrans(rs1.getInt(1));
listtransportes.add(transporte);
}
}

if (listtransportes.size() > 0){
Transporte transp= (Transporte) listtransportes.get(0);
codTrans = transp.getldtrans();
String sql = "INSERT INTO oper_orden_serv_dat (codi_orden_serv_pk, date_orden_serv_pk,
codi_orden_comp_fk, codi_prov_fk,codi_estd_fk ,codi_ciud_fk,codi_sect_fk,dire_serv, "
comen_dir,codi_empr_fk,estd_reg) VALUES(?,?,?,?,?,?,?,?,?,?,?)";
ps = con.prepareStatement(sql);
ps.setInt(1,servicio.getIdServicio());
ps.setString(2,fechaConFormato.format(fecha));
ps.setInt(3,servicio.getIdOrdenComp());
ps.setInt(4,servicio.getIdProveedor());
ps.setInt(5,1);//estado
ps.setInt(6,servicio.getCiudad());
ps.setInt(7,servicio.getSector());
ps.setString(8,servicio.getDireccion());
ps.setString(9,servicio.getComentDir());
ps.setInt(10,servicio.getEmpresa());
ps.setString(11,"A");

if (ps.executeUpdate() != 1) {
throw new CreateException("error.create.anuncio");
}

String sql4 = "SELECT codi_deta_orde,c.codi_arti,c.desc_arti,peso_brut_arti,
"volu_arti,b.cant_arti FROM cmp_orden_dat a,cmp_detalle_orden_dat b,inv_articulo_dat c "+
"WHERE a.codi_orde = b.codi_orde AND b.codi_arti = c.codi_arti AND a.codi_orde = ?";
PreparedStatement ps4 = con.prepareStatement(sql4);
ps4.setInt(1,servicio.getIdOrdenComp());
ResultSet rs3 = ps4.executeQuery();

```



```

while (rs3.next()){
    preparedStatement ps2 = con.prepareStatement("INSERT INTO oper_deta_serv_dat
(codi_orden_serv_fk,date_orden_serv_fk,item_deta_serv_pk," +
"codi_deta_orde_fk, codi_prod,cant_deta_serv,estd_reg) VALUES(?,?,?,?,?,?);");
    ps2.setInt(1,servicio.getIdServicio());
    ps2.setString(2,fechaConFormato.format(fecha));
    ps2.setInt(3,idDetalleOrdenServicio());
    ps2.setInt(4,rs3.getInt(1));
    ps2.setInt(5,rs3.getInt(2));
    ps2.setInt(6,rs3.getInt(6));
    ps2.setString(7,"A");
    ps2.executeUpdate();
}
PreparedStatement ps5 = null;
ps5 = con.prepareStatement("UPDATE cmp_orden_dat SET cmp_estado_dat = 'G' WHERE
codi_orde = ?;");
ps5.setInt(1,servicio.getIdOrdenComp());
ps5.executeUpdate();
PreparedStatement ps6 = con.prepareStatement("INSERT INTO
oper_trans_serv_dat(codi_trans_fk,codi_orden_serv_fk," +
"date_orden_serv_fk,tipo_trans_serv,estd_serv,codi_trans_serv_pk) VALUES(?,?,?,?,?,?);");
    ps6.setInt(1,codTrans);
    ps6.setInt(2,servicio.getIdServicio());
    ps6.setString(3,fechaConFormato.format(fecha));
    ps6.setString(4,"C");
    ps6.setString(5,"A");
    ps6.setInt(6,idTransServicio());
    ps6.executeUpdate();
    PreparedStatement ps7 = null;
    ps7 = con.prepareStatement("UPDATE oper_trans_dat SET dispon_trans='O'
WHERE codi_trans_pk = ?;");
    ps7.setInt(1,codTrans);
    ps7.executeUpdate();
}else{
    //System.out.print("No existen carros disponibles para el sector");
    descriptor = true;
}

return descriptor;
} catch (SQLException e) {
    e.printStackTrace();
    throw new RuntimeException(e.getMessage());
} finally {
    try {
        if (ps != null)
            ps.close();
    } catch (SQLException e) {
        e.printStackTrace();
        throw new RuntimeException(e.getMessage());
    }
}
}
}

```

grabarServicioTransf () .- Método que nos permite efectuar la emisión de ordenes de transferencia que se generan en base a solicitudes que realiza Inventario en las cuales requieren la prestación de nuestros servicios de transportación para la movilización de la mercadería de una bodega a otra. Donde la bodega que solicita es el origen y la bodega que requiere es el destino. Todo este proceso parte desde la selección de un transporte adecuado que cumpla con los requerimientos de la solicitud de transferencia como son: disponibilidad inmediata de transporte, capacidad (peso) y espacio físico (volumen). Una vez que se haya encontrado el transporte indicado se procederá a la generación de la orden de transferencia así como también a la asignación de la orden al transporte elegido.

```
public boolean grabarServicioTransf(RegistroServicioInveTransfForm servicio) throws CreateException {
    PreparedStatement ps = null;
    PreparedStatement ps1 = null;
    PreparedStatement ps2 = null;
    Date fecha = new Date();
    ArrayList listtransportes = new ArrayList();
    int codiTransp=0;

    SimpleDateFormat fechaConFormato = new SimpleDateFormat("yyyy-MM-dd");
    boolean descriptor = false;

    try {

        if (con.isClosed()) {
            throw new IllegalStateException("error.unexpected");
        }
        String sql1 = "SELECT codi_deta,c.codi_arti,c.desc_arti,peso_brut_arti,"+
            "volu_arti,b.cant_deta FROM inv_transaccion_log a,inv_detalle_transaccion_log"+
            "b,inv_articulo_dat c WHERE a.codi_tran = b.codi_tran"+
            "AND b.codi_arti = c.codi_arti AND a.codi_tran = ?";
        ps1 = con.prepareStatement(sql1);
        ps1.setInt(1,servicio.getIdServicio());
        ResultSet rs = ps1.executeQuery();
        float pesoTotalTransf = 0;
        float volumenTotalTransf = 0;
```

```

while (rs.next()){
    pesoTotalTransf = pesoTotalTransf + (rs.getFloat(4) * rs.getFloat(6));
    volumenTotalTransf = volumenTotalTransf + (rs.getFloat(5) * rs.getFloat(6));
}
String sql2 = "SELECT codi_trans_pk,desc_trans,capa_trans,volu_trans FROM oper_trans_dat a " +
"WHERE a.estd_reg_trans = 'A' AND dispon_trans='D';";

float pesoTrans = 0;
float voluTrans = 0;
ps2 = con.prepareStatement(sql2);
ResultSet rs1 = ps2.executeQuery(sql2);
Transporte transporte=new Transporte();

while(rs1.next()){
    pesoTrans=rs1.getFloat(3);
    voluTrans=rs1.getFloat(4);
    if ((pesoTrans>=pesoTotalTransf)&&(voluTrans>=volumenTotalTransf)){
        transporte.setldtrans(rs1.getInt(1));
        listtransportes.add(transporte);
    }
}

if (listtransportes.size() > 0){
    Transporte transp= (Transporte) listtransportes.get(0);
    codiTransp = transp.getldtrans();
    String sql = "INSERT INTO oper_orden_serv_inv_dat(codi_orden_serv_inv_pk, "+
        "date_orden_serv_inv_pk, codi_transfer_fk, codi_empr_fk, "+
        "codi_estd_fk, codi_ciud_orig ,dire_serv_desti,dire_serv_orig,sucur_orig,"
        "sucur_dest,bode_orig,bode_desti,codi_ciud_desti,estd_reg) " +
        "VALUES(?,?,?,?,?,?,?,?,?,?,?,?,?)";
    ps = con.prepareStatement(sql);
    ps.setInt(1,servicio.getIdServicio());
    ps.setString(2,fechaConFormato.format(fecha));
    ps.setInt(3,servicio.getIdInveTranf());
    //ps.setInt(3,1);
    ps.setInt(4,servicio.getEmpresa());
    //ps.setInt(4,1);
    ps.setInt(5,2);
    ps.setInt(6,servicio.getCiudadSucur1());
    ps.setString(7,servicio.getDireSucur2());
    ps.setString(8,servicio.getDireSucur1());
    ps.setInt(9,servicio.getIdSucur1());
    ps.setInt(10,servicio.getIdSucur2());
    ps.setInt(11,servicio.getBodega1());
    ps.setInt(12,servicio.getBodega2());
    ps.setInt(13,servicio.getCiudadSucur2());
    ps.setString(14,"A");
    if (ps.executeUpdate() != 1) {
        throw new CreateException("error.create.anuncio");
    }
}
String sql4 = "SELECT b.codi_deta,c.codi_arti,c.desc_arti,b.cant_deta " +
"FROM inv_transaccion_log a,inv_detalle_transaccion_log b,inv_articulo_dat c " +
"WHERE a.codi_tran = b.codi_tran AND b.codi_arti = c.codi_arti AND a.codi_tran=?";
PreparedStatement ps3 = con.prepareStatement(sql4);
ps3.setInt(1,servicio.getIdInveTranf());
ResultSet rs2 = ps3.executeQuery();

```

```

while (rs2.next()){
    PreparedStatement ps4 = con.prepareStatement("INSERT INTO oper_deta_serv_inv_dat " +
        " (codi_orden_serv_inv_fk,date_orden_serv_inv_fk,codi_empr_fk," +
        "item_deta_serv_inv_pk,codi_prod_fk,cant_deta_serv_inv,codi_deta_trans_fk,"+
        "codi_trans_fk,estd_reg) VALUES(?,?,?,?,?,?,?,?);");
    ps4.setInt(1,servicio.getIdServicio());
    ps4.setString(2,fechaConFormato.format(fechar));
    ps4.setInt(3,servicio.getEmpresa());
    ps4.setInt(4,idDetalleServTransf());
    ps4.setInt(5,rs2.getInt(2));
    ps4.setInt(6,rs2.getInt(4));
    ps4.setInt(7,rs2.getInt(1));
    ps4.setInt(8,servicio.getIdInveTranf());
    ps4.setString(9,"A");
    ps4.executeUpdate();
}
PreparedStatement ps5 = null;
ps5 = con.prepareStatement("UPDATE inv_transaccion_log SET tipo_trans = 'G' "+
    "WHERE codi_tran = ?;");
ps5.setInt(1,servicio.getIdInveTranf());
ps5.executeUpdate();
PreparedStatement ps6 = con.prepareStatement("INSERT INTO"+
    "oper_trans_serv_dat(codi_trans_fk,codi_orden_serv_fk," +
    "date_orden_serv_fk,tipo_trans_serv,estd_serv,codi_trans_serv_pk) VALUES(?,?,?,?,?);");
ps6.setInt(1,codiTransp);
ps6.setInt(2,servicio.getIdServicio());
ps6.setString(3,fechaConFormato.format(fechar));
ps6.setString(4,"T");
ps6.setString(5,"A");
ps6.setInt(6,idTransServicio());
ps6.executeUpdate();
PreparedStatement ps7 = null;
ps7 = con.prepareStatement("UPDATE oper_trans_dat SET dispon_trans='O' "+
    "WHERE codi_trans_pk = ?;");
ps7.setInt(1,codiTransp);
ps7.executeUpdate();
}
else{
    //System.out.print("No existen carros disponibles para el sector");
    descriptor = true;
}
    return descriptor;
} catch (SQLException e) {
    e.printStackTrace();
    throw new RuntimeException(e.getMessage());
} finally {
    try {
        if (ps != null)
            ps.close();
    } catch (SQLException e) {
        e.printStackTrace();
        throw new RuntimeException(e.getMessage());
    }
}
}
}

```