



Facultad de
Ciencias
UNAM

Fundamentos de Bases de Datos
Grupo 7077

PRÁCTICA 6. LENGUAJE PARA DEFINICIÓN
DE DATOS. DDL, (DATA DEFINITION
LANGUAGE)

EQUIPO: mmm xd

317088296 - Demian Oswaldo Garcia Toxqui.

317042522 - Erick Bernal Márquez.

317180321 - Karen Cristóbal Morales.

317061521 - Rosa María Robles Huerta.

317205776 - Rubén Acosta Arazate.

27 de abril de 2022



Modelo Relacional

Últimas correcciones a nuestras tablas:

- En la relación “TenerPa” entre las entidades **Duenio** y **MetodoPago**, debido a que hicimos directamente la relación con el padre, agregamos en las tablas **Fisico** e **Internet** la curp del cliente normal y la curp del cliente frecuente.
- Lo anterior pasa igual con la tabla **Mascota**, por lo tanto agregamos en dicha tabla la curp del cliente normal y la curp del cliente frecuente.
- Quitamos la tabla “Cuarto”.

Consideraciones al momento de escoger el tipo de dato y las restricciones de dominio para cada columna.

Tabla Supervisor

```
--Supervisor
CREATE TABLE Supervisor(
  curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) CHECK(curp SIMILAR TO '[a-z]{4}[0-9]{6}[a-z]{8}') UNIQUE,
  nombreSup VARCHAR(50) NOT NULL CHECK(nombreSup <> ''),
  paternoSup VARCHAR(50) NOT NULL CHECK(paternoSup <> ''),
  maternoSup VARCHAR(50) NOT NULL CHECK(maternoSup <> ''),
  calleSup VARCHAR NOT NULL CHECK(calleSup <> ''),
  numeroSup CHAR(4) NOT NULL CHECK(numeroSup SIMILAR TO '[0-9]{4}'),
  estadoSup VARCHAR NOT NULL CHECK(estadoSup <> ''),
  cpSup CHAR(5) NOT NULL CHECK(cpSup SIMILAR TO '[0-9]{5}'),
  generoSup CHAR(1) NOT NULL CHECK(generoSup = 'M' OR generoSup = 'F'),
  nacimientoSup DATE NOT NULL CHECK(nacimientoSup > '1950-01-01' AND nacimientoSup < '2022-12-31'),
  telefonoSup CHAR(10) NOT NULL CHECK(telefonoSup SIMILAR TO '[0-9]{10}'),
  salarioSup FLOAT NOT NULL CHECK(salarioSup > 0),
  rfcSup CHAR(13) NOT NULL CHECK(CHAR_LENGTH(rfcSup) = 13)
);
```

curp: Es la llave primaria de la tabla Supervisor.

- No aceptamos valor null puesto que es la llave primaria.
- La curp está obligada a tener una longitud de 18 caracteres, donde verificamos que los primeros 4 caracteres sean letras, los siguientes 6 caracteres dígitos y los 8 caracteres restantes letras.
- La curp es única para cada persona, por ello que lo limitamos a que fuera único.

nombreSup: Es el nombre del Supervisor.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 50 caracteres.
- El nombre no debe de ser una cadena vacía ya que no tendría sentido tener supervisores sin nombre, por esa misma razón no permitimos que tome el valor null.

paternoSup: Es el apellido paterno del Supervisor.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada apellido debe tener máximo 50 caracteres.
- El apellido no debe de ser una cadena vacía y no se permite que tome el valor null.

maternoSup: Es el apellido materno del Supervisor.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada apellido debe tener máximo 50 caracteres.
- El apellido no debe de ser una cadena vacía y no se permite que tome el valor null.

calleSup: Es la calle donde vive el Supervisor.

- Debido a que las calles son muy variadas establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- La calle no debe de ser una cadena vacía y no se permite que tome el valor null.

numeroSup: Es la número donde vive el Supervisor.

- El número está obligado a tener una longitud de 4 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

estadoSup: Es la Estado donde vive el Supervisor.

- Debido a que los Estados son muy variados establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El Estado no debe de ser una cadena vacía y no se permite que tome el valor null.

cpSup: Es el código postal donde vive el Supervisor.

- El código postal está obligado a tener una longitud de 5 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

generoSup: Es el género del Supervisor.

- El género está obligado a tener un solo caracter, el cual admite los valores 'M' para masculino ó 'F' para femenino.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

nacimientoSup: Es la fecha de nacimiento del Supervisor.

- Este dato es de tipo DATE, donde se verifica que las personas hayan nacido a partir del año de 1950 hasta el año 2022.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

telefonoSup: Es la número telefónico del Supervisor.

- El telefono está obligado a tener una longitud de 10 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

salarioSup: Es la remuneración que recibe el supervisor por parte de la empresa, por brindar sus servicios.

- Este dato es de tipo FLOAT debido a que un salario esta sujeto a impuestos, bonos, inflación, etc. y no siempre será un entero.

- El salario debe de ser mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

rfcSup: Es el RFC del Supervisor.

- El RFC está obligado a tener una longitud de 13 caracteres alfanuméricos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla Clinica

--Clinica

```
CREATE TABLE Clinica(
    idClinica CHAR(5) NOT NULL CHECK(idClinica SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE,
    nombreCli VARCHAR(30) NOT NULL CHECK(nombreCli <> ''),
    calleCli VARCHAR NOT NULL CHECK(calleCli <> ''),
    estadoCli VARCHAR NOT NULL CHECK(estadoCli <> ''),
    cpCli CHAR(5) NOT NULL CHECK (cpCli SIMILAR TO '[0-9]{5}'),
    telefonoCli CHAR(10) NOT NULL CHECK(telefonoCli SIMILAR TO '[0-9]{10}'),
    horarioCli TIME NOT NULL
);
```

idClinica: Es la llave primaria de la tabla Clinica.

- No aceptamos valor null puesto que es la llave primaria.
- El id está obligado a tener una longitud de 5 caracteres, donde verificamos que los primeros 3 caracteres sean letras (iniciales del nombre de la clinica), y los caracteres restantes dígitos positivos.
- El id es único para cada clínica, por ello que lo limitamos a que fuera único.

nombreCli: Es el nombre de la Clinica.

- Debido a que este valor puede ser muy variado establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 30 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

calleCli: Es la calle donde se ubica la Clinica.

- Debido a que las calles son muy variadas establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- La calle no debe de ser una cadena vacía y no se permite que tome el valor null.

estadoCli: Es la Estado donde se ubica la Clinica.

- Debido a que los Estados son muy variados establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El Estado no debe de ser una cadena vacía y no se permite que tome el valor null.

cpCli: Es el código postal donde se ubica la Clinica.

- El código postal está obligado a tener una longitud de 5 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

telefonoCLI: Es la número telefónico de la Clinica.

- El telefono está obligado a tener una longitud de 10 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

horarioCli: Es el horario en el que la Clinica da servicio.

- Este dato es de tipo TIME debido a las facilidades de formato que nos brinda.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla Juguete

--Juguete

```
CREATE TABLE Juguete(  
    idProducto CHAR(10) NOT NULL CHECK(CHAR_LENGTH(idProducto) = 10) UNIQUE ,  
    nombre VARCHAR(30) NOT NULL CHECK(nombre <> ''),  
    precio FLOAT NOT NULL CHECK(precio>0),  
    numUnidades INT NOT NULL CHECK(numUnidades>0),  
    descripcion VARCHAR NOT NULL CHECK(descripcion <> ''),  
    nombreArchivo VARCHAR NOT NULL CHECK(nombreArchivo <> '')  
);
```

idProducto: Es la llave primaria de la tabla Juguete.

- No aceptamos valor null puesto que es la llave primaria.
- El id está obligado a tener una longitud de 10 de caracteres alfanuméricos.
- La id es único para cada juguete, por ello que lo limitamos a que fuera único.

nombre: Es el nombre del Juguete.

- Debido a que este valor puede ser muy variado establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 30 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

precio: Es el precio del Juguete.

- Este dato es de tipo FLOAT debido a que a veces el precio se maneja con centavos.
- Verificamos que el precio sea mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

numUnidades: Es la cantidad de juguetes que hay.

- Este dato es de tipo INT debido a los juguetes se manejan como unidades enteras.
- Verificamos que el número de unidades sea mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

descripcion: Es la descripción del Juguete.

- Debido a que la descripción suele ser variada establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- La descripción no debe de ser una cadena vacía y no se permite que tome el valor null.

nombreArchivo: Es la nombre de archivo del Juguete.

- Debido a que el nombre del archivo podría ser variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- La descripción no debe de ser una cadena vacía y no se permite que tome el valor null.

Tabla Accesorio

--Accesorio

```
CREATE TABLE Accesorio(  
    idProducto CHAR(10) NOT NULL CHECK(CHAR_LENGTH(idProducto) = 10) UNIQUE,  
    nombre VARCHAR(30) NOT NULL CHECK(nombre <> ''),  
    precio FLOAT NOT NULL CHECK(precio>0),  
    numUnidades INT NOT NULL CHECK(numUnidades>0),  
    descripcion VARCHAR NOT NULL CHECK(descripcion <> ''),  
    nombreArchivo VARCHAR NOT NULL CHECK(nombreArchivo <> '')  
);
```

Tiene las mismas especificaciones y restricciones que la tabla Juguete debido a que heredan del mismo padre.

Tabla Medicamento

--Medicamento

```
CREATE TABLE Medicamento(  
    idProducto CHAR(10) NOT NULL CHECK(CHAR_LENGTH(idProducto) = 10) UNIQUE,  
    nombre VARCHAR(30) NOT NULL CHECK(nombre <> ''),  
    precio FLOAT NOT NULL CHECK(precio>0),  
    numUnidades INT NOT NULL CHECK(numUnidades>0),  
    descripcion VARCHAR NOT NULL CHECK(descripcion <> ''),  
    nombreArchivo VARCHAR NOT NULL CHECK(nombreArchivo <> ''),  
    caducidad DATE NOT NULL CHECK(caducidad > '2022-01-01')  
);
```

Tiene las mismas especificaciones y restricciones que la tabla Juguete debido a que heredan del mismo padre. Solo se agrega el atributo "caducidad"

caducidad: Es la fecha de caducidad del Medicamento.

- Este dato es de tipo DATE debido a las facilidades de formato que nos brinda. Se verifica que el año de la fecha de caducidad sea mayor al año 2022.
 - Deseamos que se especifique este dato por ello no permitimos que sea null.
-

Tabla Comida

--Comida

```
CREATE TABLE Comida(  
    idProducto CHAR(10) NOT NULL CHECK(CHAR_LENGTH(idProducto) = 10) UNIQUE,  
    nombre VARCHAR(30) NOT NULL CHECK(nombre <> ''),  
    precio FLOAT NOT NULL CHECK(precio>0),  
    numUnidades INT NOT NULL CHECK(numUnidades>0),  
    descripcion VARCHAR NOT NULL CHECK(descripcion <> ''),  
    nombreArchivo VARCHAR NOT NULL CHECK(nombreArchivo <> ''),  
    caducidad DATE NOT NULL CHECK(caducidad > '2022-01-01')  
);
```

Tiene las mismas especificaciones y restricciones que la tabla Juguete debido a que heredan del mismo padre. Solo se agrega el atributo "caducidad"

caducidad: Es la fecha de caducidad de la Comida.

- Este dato es de tipo DATE debido a las facilidades de formato que nos brinda. Se verifica que el año de la fecha de caducidad sea mayor al año 2022.
 - Deseamos que se especifique este dato por ello no permitimos que sea null.
-

Tabla Consultorio

--Consultorio

```
CREATE TABLE Consultorio(  
    idClinica CHAR(5) NOT NULL CHECK(idClinica SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE,  
    idCuarto CHAR(2) NOT NULL CHECK(idCuarto SIMILAR TO '[0-9]{2}') UNIQUE  
);
```

Es una entidad débil, por lo tanto su llave primaria está compuesta de los atributos “idClinica” y “idCuarto”

idClinica: Es la llave primaria de la entidad fuerte “Clinica”. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

idCuarto: Forma parte de la llave primaria de la tabla Consultorio

- No aceptamos valor de null puesto que forma parte de la llave primaria.
 - El id está obligado a tener una longitud de 2 caracteres, donde verificamos que sean dígitos positivos.
 - El id es único para cada Consultorio, es por ello que lo limitamos a que fuera único.
-

Tabla CuartoCuidados

--Cuarto Cuidados

```
CREATE TABLE CuartoCuidados(  
    idClinica CHAR(5) NOT NULL CHECK(idClinica SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE,  
    idCuarto CHAR(2) NOT NULL CHECK(idCuarto SIMILAR TO '[0-9]{2}') UNIQUE  
);
```

Es una entidad débil, por lo tanto su llave primaria está compuesta de los atributos “idClinica” y “idCuarto”

idClinica: Es la llave primaria de la entidad fuerte “Clinica”. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

idCuarto: Forma parte de la llave primaria de la tabla CuartoCuidados.

- No aceptamos valor de null puesto que forma parte de la llave primaria.
 - El id está obligado a tener una longitud de 2 caracteres, donde verificamos que sean dígitos positivos.
 - El id es único para cada Cuarto de cuidados, es por ello que lo limitamos a que fuera único.
-

Tabla Caja

```
--Caja
CREATE TABLE Caja(
    idClinica CHAR(5) NOT NULL CHECK(idClinica SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE,
    idCaja CHAR(2) NOT NULL CHECK(idCaja SIMILAR TO '[0-9]{2}') UNIQUE
);
```

Es una entidad débil, por lo tanto su llave primaria está compuesta de los atributos “idClinica” y “idCaja”

idClinica: Es la llave primaria de la entidad fuerte “Clinica”. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

idCaja: Forma parte de la llave primaria de la tabla Caja.

- No aceptamos valor de null puesto que forma parte de la llave primaria.
 - El id está obligado a tener una longitud de 2 caracteres, donde verificamos que sean dígitos positivos.
 - El id es único para cada Caja, es por ello que lo limitamos a que fuera único.
-

Tabla TenerJuguete

```
--Tener Juguete
CREATE TABLE TenerJuguete(
    idClinica CHAR(5) NOT NULL,
    idProducto CHAR(10) NOT NULL
);
```

idClinica: Es una llave foránea en la tabla TenerJuguete que nos permitirá ligar a la clínica en la relación “Tener” con la entidad Juguete. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

idProducto: Es una llave foránea en la tabla TenerJuguete que nos permitirá ligar al juguete en la relación “Tener” con la entidad Clinica. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

Tabla TenerAccesorio

```
--Tener Accesorio
CREATE TABLE TenerAccesorio(
    idClinica CHAR(5) NOT NULL,
    idProducto CHAR(10) NOT NULL
);
```

idClinica: Es una llave foránea en la tabla TenerAccesorio que nos permitirá ligar a la clínica en la relación “Tener” con la entidad Accesorio. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

idProducto: Es una llave foránea en la tabla TenerAccesorio que nos permitirá ligar al accesorio en la relación “Tener” con la entidad Clinica. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

Tabla TenerMedicamento

```
--Tener Medicamento
CREATE TABLE TenerMedicamento(
    idClinica CHAR(5) NOT NULL,
    idProducto CHAR(10) NOT NULL
);
```

idClinica: Es una llave foránea en la tabla TenerMedicamento que nos permitirá ligar a la clínica en la relación “Tener” con la entidad Medicamento. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

idProducto: Es una llave foránea en la tabla TenerMedicamento que nos permitirá ligar al medicamento en la relación “Tener” con la entidad Clinica. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

Tabla TenerComida

```
--Tener Comida
CREATE TABLE TenerComida(
    idClinica CHAR(5) NOT NULL,
    idProducto CHAR(10) NOT NULL
);
```

idClinica: Es una llave foránea en la tabla TenerComida que nos permitirá ligar a la clínica en la relación “Tener” con la entidad Comida. Sus restricciones son las mismas que se definieron para

“idClinica” en la tabla **Clinica**.

idProducto: Es una llave foránea en la tabla TenerComida que nos permitirá ligar a la comida en la relación “Tener” con la entidad Clinica. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

Tabla VenderJuguete

```
--Vender Juguete
CREATE TABLE VenderJuguete(
    idProducto CHAR(10) NOT NULL,
    idCaja CHAR(2) NOT NULL,
    idClinica CHAR(5) NOT NULL
);
```

idProducto: Es una llave foránea en la tabla VenderJuguete que nos permitirá ligar al juguete en la relación “Vender” con la entidad Caja. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

idCaja, idClinica: Es una llave compuesta foránea en la tabla VenderJuguete que nos permitirá ligar a la caja en la relación “Vender” con la entidad Juguete. Sus restricciones son las mismas que se definieron en la tabla **Caja**.

Tabla VenderAccesorio

```
--Vender Accesorio
CREATE TABLE VenderAccesorio(
    idProducto CHAR(10) NOT NULL,
    idCaja CHAR(2) NOT NULL,
    idClinica CHAR(5) NOT NULL
);
```

idProducto: Es una llave foránea en la tabla VenderAccesorio que nos permitirá ligar al accesorio en la relación “Vender” con la entidad Caja. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

idCaja, idClinica: Es una llave compuesta foránea en la tabla VenderAccesorio que nos permitirá ligar a la caja en la relación “Vender” con la entidad Accesorio. Sus restricciones son las mismas que se definieron en la tabla **Caja**.

Tabla VenderMedicamento

```
--Vender Medicamento
CREATE TABLE VenderMedicamento(
    idProducto CHAR(10) NOT NULL,
    idCaja CHAR(2) NOT NULL,
    idClinica CHAR(5) NOT NULL
);
```

idProducto: Es una llave foránea en la tabla VenderMedicamento que nos permitirá ligar al medicamento en la relación “Vender” con la entidad Caja. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

idCaja, idClinica: Es una llave compuesta foránea en la tabla VenderMedicamento que nos permitirá ligar a la caja en la relación “Vender” con la entidad Medicamento. Sus restricciones son las mismas que se definieron en la tabla **Caja**.

Tabla VenderComida

```
--Vender Comida
CREATE TABLE VenderComida(
    idProducto CHAR(10) NOT NULL,
    idCaja CHAR(2) NOT NULL,
    idClinica CHAR(5) NOT NULL
);
```

idProducto: Es una llave foránea en la tabla VenderComida que nos permitirá ligar a la comida en la relación “Vender” con la entidad Caja. Sus restricciones son las mismas que se definieron para “idProducto” en la tabla **Juguete**.

idCaja, idClinica: Es una llave compuesta foránea en la tabla VenderComida que nos permitirá ligar a la caja en la relación “Vender” con la entidad Comida. Sus restricciones son las mismas que se definieron en la tabla **Caja**.

Tabla Supervisor

```
--Supervisor
CREATE TABLE Supervisor(
  curp CHAR(18) NOT NULL,
  idClinica CHAR(5) NOT NULL,
  nombreSup VARCHAR(50) NOT NULL CHECK(nombreSup <> ''),
  paternoSup VARCHAR(50) NOT NULL CHECK(paternoSup <> ''),
  maternoSup VARCHAR(50) NOT NULL CHECK(maternoSup <> ''),
  numeroSup CHAR(4) NOT NULL CHECK(numeroSup SIMILAR TO '[0-9]{4}'),
  calleSup VARCHAR NOT NULL CHECK(calleSup <> ''),
  estadoSup VARCHAR NOT NULL CHECK(estadoSup <> ''),
  cpSup CHAR(5) NOT NULL CHECK(cpSup SIMILAR TO '[0-9]{5}'),
  generoSup CHAR(1) NOT NULL CHECK(generoSup = 'M' OR generoSup = 'F'),
  nacimientoSup DATE NOT NULL,
  telefonoSup CHAR(10) NOT NULL CHECK(telefonoSup SIMILAR TO '[0-9]{10}'),
  salarioSup FLOAT NOT NULL CHECK(salarioSup > 0),
  rfcSup CHAR(13) NOT NULL CHECK(CHAR_LENGTH(rfcSup) = 13),
  nombreCli VARCHAR(30) NOT NULL CHECK(nombreCli <> ''),
  numeroCli CHAR(4) NOT NULL CHECK(numeroCli SIMILAR TO '[0-9]{4}'),
  calleCli VARCHAR NOT NULL CHECK(calleCli <> ''),
  estadoCli VARCHAR NOT NULL CHECK(estadoCli <> ''),
  cpCli CHAR(5) NOT NULL CHECK(cpCli SIMILAR TO '[0-9]{5}'),
  telefonoCli CHAR(10) NOT NULL CHECK(telefonoCli SIMILAR TO '[0-9]{10}'),
  horarioCli TIME NOT NULL,
  inicioSupervision DATE NOT NULL CHECK(inicioSupervision > '1970-01-01' AND inicioSupervision < '2022-12-31'),
  finSupervision DATE NOT NULL CHECK(finSupervision > '1970-01-01' AND finSupervision < '2022-12-31')
);
```

- **curp:** Es una llave foránea en la tabla Supervisor que nos permitirá ligar al supervisor en la relación “Supervisor” con la entidad Clinica. Sus restricciones son las mismas que se definieron para “curp” en la tabla **Supervisor**.
- **idClinica:** Es una llave foránea en la tabla Supervisor que nos permitirá ligar a la clínica en la relación “Supervisor” con la entidad Supervisor. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

Todos los atributos en la tabla **Supervisor** y todos los atributos en la tabla **Clinica** se incluyen en la tabla **Supervisor**, por lo tanto tiene las mismas restricciones que en dichas tablas. Solo se agregan los siguientes atributos:

- **inicioSupervision:** Este dato es de tipo DATE debido a las facilidades de formato que nos brinda. Se verifica que el año sea mayor a 1970 y menor a 2022. Deseamos que se especifique este dato por ello no permitimos que sea null.
- **finSupervision:** Este dato es de tipo DATE debido a las facilidades de formato que nos brinda. Se verifica que el año sea mayor a 1970 y menor a 2022. Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla GenerarReciboNormal

```
--Generar Recibo Normal
CREATE TABLE GenerarReciboNormal(
    idRecibo CHAR(4) NOT NULL,
    idConsulta CHAR(5) NOT NULL,
    idMascota CHAR(5) NOT NULL,
    servicioBrindado VARCHAR NOT NULL CHECK(servicioBrindado <> ''),
    nombreTrabajador VARCHAR(50) NOT NULL CHECK(nombreTrabajador <> ''),
    nombreMascota VARCHAR(20) NOT NULL CHECK(nombreMascota <> ''),
    nombreDuenio VARCHAR(50) NOT NULL CHECK(nombreDuenio <> ''),
    precio FLOAT NOT NULL CHECK(precio > 0),
    fechaProxCita DATE NOT NULL CHECK(fechaProxCita > '2021-01-01'),
    medicamento VARCHAR NOT NULL,
    motivoConsulta VARCHAR NOT NULL CHECK(motivoConsulta <> ''),
    estadoMascota VARCHAR NOT NULL CHECK(estadoMascota <> '')
);
```

idRecibo: Es una llave foránea en la tabla GenerarReciboNormal que nos permitirá ligar un recibo en la relación “GenerarReciboNormal” con la entidad Normal (que hace referencia a una consulta normal). Sus restricciones son las mismas que se definieron para “idRecibo” en la tabla **Recibo**.

idConsulta, idMascota: Es una llave compuesta foránea en la tabla GenerarReciboNormal que nos permitirá ligar a la consulta normal en la relación “GenerarReciboNormal” con la entidad Recibo. Sus restricciones son las mismas que se definieron en la tabla **Normal**.

servicioBrindado: Es el servicio brindado a la Mascota.

- Debido a que el servicio puede ser muy variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El servicio brindado no debe de ser una cadena vacía y no se permite que tome el valor null.

nombreTrabajador: Es el nombre del trabajador que atendió a la mascota.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 50 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

nombreMascota: Es el nombre de la mascota que se atendió.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 20 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

nombreDuenio: Es el nombre del dueño de la mascota que se atendió.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 50 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

precio: Es el costo total de la consulta.

- Este dato es de tipo FLOAT debido a que a veces el precio se maneja con centavos.
- Verificamos que el precio sea mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

fechaProxCita: Es la fecha de la próxima consulta para la mascota.

- Este dato es de tipo DATE debido a las facilidades de formato que nos brinda. Se verifica que el año de la próxima cita sea mayor al año 2021.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

medicamento: Es el medicamento recetado a la Mascota.

- Debido a que el medicamento puede ser muy variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El medicamento brindado sí puede ser una cadena vacía (esto en caso de que la mascota haya ido solo a revisión y no requiere de más medicamento).
- No se permite que tome valor null.

motivoConsulta: Es el motivo por el cual la mascota fue a consulta.

- Debido a que el motivo puede ser muy variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El motivo de consulta no debe de ser una cadena vacía y no se permite que tome el valor null.

estadoMascota: Es el estado de la mascota que fue a consulta.

- Debido a que el estado puede ser muy variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El estado de la mascota no debe de ser una cadena vacía y no se permite que tome el valor null.

Tabla GenerarReciboEmergencia

```
--Generar Recibo Emergencia
CREATE TABLE GenerarReciboEmergencia(
  idRecibo CHAR(4) NOT NULL,
  idConsulta CHAR(5) NOT NULL,
  idMascota CHAR(5) NOT NULL,
  servicioBrindado VARCHAR NOT NULL CHECK(servicioBrindado <> ''),
  nombreTrabajador VARCHAR(50) NOT NULL CHECK(nombreTrabajador <> ''),
  nombreMascota VARCHAR(20) NOT NULL CHECK(nombreMascota <> ''),
  nombreDuenio VARCHAR(50) NOT NULL CHECK(nombreDuenio <> ''),
  precio FLOAT NOT NULL CHECK(precio > 0),
  procedimientoEmitido VARCHAR NOT NULL CHECK(procedimientoEmitido <> ''),
  sintomasMascota VARCHAR NOT NULL CHECK(sintomasMascota <> ''),
  codigoEmergencia VARCHAR NOT NULL CHECK(codigoEmergencia = 'verde' OR codigoEmergencia = 'amarillo' OR codigoEmergencia = 'rojo')
);
```

idRecibo: Es una llave foránea en la tabla GenerarReciboEmergencia que nos permitirá ligar un recibo en la relación “GenerarReciboEmergencia” con la entidad Emergencia (que hace referencia a una consulta de emergencia). Sus restricciones son las mismas que se definieron para “idRecibo” en la tabla **Recibo**.

idConsulta, idMascota: Es una llave compuesta foránea en la tabla GenerarReciboEmergencia que nos permitirá ligar a la consulta de emergencia en la relación “GenerarReciboEmergencia” con la entidad Recibo. Sus restricciones son las mismas que se definieron en la tabla **Emergencia**.

Tiene las mismas especificaciones y restricciones que la tabla GenerarReciboNormal. De la tabla GenerarReciboNormal solo se excluyen los atributos “fechaProxCita”, “medicamento”, “motivoConsulta” y “estadoMascota”.

Se agregan los siguientes atributos:

procedimientoEmitido: Es el procedimiento que se hizo para tratar a la Mascota.

- Debido a que el procedimiento puede ser muy variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El procedimiento no debe de ser una cadena vacía y no se permite que tome el valor null.

sintomasMascota: Son los síntomas que presenta la Mascota.

- Debido a que los síntomas puede ser muy variados establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- Los síntomas no deben de ser una cadena vacía y no se permite que tome el valor null.

codigoEmergencia: Es el estado de la mascota que fue a consulta.

- Cadena que verifica que se reciba como entrada 'verde' en caso de una emergencia leve, 'amarillo' en caso de una emergencia considerable ó 'rojo' en caso de una emergencia grave.
- El código de emergencia no debe de ser una cadena vacía y no se permite que tome el valor null.

Tabla GenerarReciboTratamiento

```
--Generar Recibo Tratamiento
CREATE TABLE GenerarReciboTratamiento(
    idRecibo CHAR(4) NOT NULL,
    idConsulta CHAR(5) NOT NULL,
    idMascota CHAR(5) NOT NULL,
    servicioBrindado VARCHAR NOT NULL CHECK(servicioBrindado <> ''),
    nombreTrabajador VARCHAR(50) NOT NULL CHECK(nombreTrabajador <> ''),
    nombreMascota VARCHAR(20) NOT NULL CHECK(nombreMascota <> ''),
    nombreDuenio VARCHAR(50) NOT NULL CHECK(nombreDuenio <> ''),
    precio FLOAT NOT NULL CHECK(precio > 0),
    servicioDado VARCHAR NOT NULL CHECK(servicioDado <> '')
);
```


idRecibo: Es una llave foránea en la tabla GenerarReciboTratamiento que nos permitirá ligar un recibo en la relación “GenerarReciboTratamiento” con la entidad Tratamiento (que hace referencia a una consulta estética). Sus restricciones son las mismas que se definieron para “idRecibo” en la tabla **Recibo**.

idConsulta, idMascota: Es una llave compuesta foránea en la tabla GenerarReciboTratamiento que nos permitirá ligar a la consulta de tratamiento en la relación “GenerarReciboTratamiento” con la entidad Recibo. Sus restricciones son las mismas que se definieron en la tabla **Tratamiento**.

Tiene las mismas especificaciones y restricciones que la tabla GenerarReciboNormal. De la tabla GenerarReciboNormal solo se excluyen los atributos “fechaProxCita”, “medicamento”, “motivoConsulta” y “estadoMascota”.

Se agrega el siguiente atributo:

servicioDado: Es el servicio brindado a la Mascota.

- Debido a que el servicio puede ser muy variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El servicio brindado no debe de ser una cadena vacía y no se permite que tome el valor null.

Tabla ClienteNormal

```
--Cliente Normal
CREATE TABLE ClienteNormal(
  curpClienteNormal CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) CHECK(curp SIMILAR TO '[a-z]{4}[0-9]{6}[a-z]{8}') UNIQUE,
  idCaja CHAR(2) NOT NULL,
  idClinica CHAR(5) NOT NULL,
  nombre VARCHAR(50) NOT NULL CHECK(nombre <> ''),
  paterno VARCHAR(50) NOT NULL CHECK(paterno <> ''),
  materno VARCHAR(50) NOT NULL CHECK(materno <> ''),
  numero CHAR(4) NOT NULL CHECK(numero SIMILAR TO '[0-9]{4}'),
  calle VARCHAR NOT NULL CHECK(calle <> ''),
  estado VARCHAR NOT NULL CHECK(estado <> ''),
  cp CHAR(5) NOT NULL CHECK(cp SIMILAR TO '[0-9]{5}'),
  genero CHAR(1) NOT NULL CHECK(genero = 'M' OR genero = 'F'),
  nacimiento DATE NOT NULL CHECK(nacimiento > '1950-01-01' AND nacimiento < '2022-12-31'),
  telefono CHAR(10) NOT NULL CHECK(telefono SIMILAR TO '[0-9]{10}')
);
```

curpClienteNormal: Es la llave primaria de la tabla ClienteNormal.

- No aceptamos valor null puesto que es la llave primaria.
- La curp está obligada a tener una longitud de 18 caracteres, donde verificamos que los primeros 4 caracteres sean letras, los siguientes 6 caracteres dígitos y los 8 caracteres restantes letras.
- La curp es única para cada persona, por ello que lo limitamos a que fuera único.

idCaja, idClinica: Es una llave compuesta foránea en la tabla ClienteNormal que nos permitirá ligar a la caja en la relación “Cobrar” con la entidad ClienteNormal. Sus restricciones son las mismas que se definieron en la tabla **Caja**.

nombre: Es el nombre del Cliente.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 50 caracteres.
- El nombre no debe de ser una cadena vacía y no permitimos que tome el valor null.

paterno: Es el apellido paterno del Cliente.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada apellido debe tener máximo 50 caracteres.
- El apellido no debe de ser una cadena vacía y no se permite que tome el valor null.

materno: Es el apellido materno del Cliente.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada apellido debe tener máximo 50 caracteres.
- El apellido no debe de ser una cadena vacía y no se permite que tome el valor null.

numero: Es la número donde vive el Cliente.

- El número está obligado a tener una logitud de 4 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

calle: Es la calle donde vive el Cliente.

- Debido a que las calles son muy variadas establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- La calle no debe de ser una cadena vacía y no se permite que tome el valor null.

estado: Es la Estado donde vive el Cliente.

- Debido a que los Estados son muy variados establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El Estado no debe de ser una cadena vacía y no se permite que tome el valor null.

cp: Es el código postal donde vive el Cliente.

- El código postal está obligado a tener una logitud de 5 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

genero: Es el género del Cliente.

- El género está obligado a tener un solo caracter, el cual admite los valores 'M' para masculino ó 'F' para femenino.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

nacimiento: Es la fecha de nacimiento del Cliente.

- Este dato es de tipo DATE, donde se verifica que las personas hayan nacido a partir del año de 1950 hasta el año 2022.

- Deseamos que se especifique este dato por ello no permitimos que sea null.

telefono: Es la número telefónico del Cliente.

- El telefono está obligado a tener una longitud de 10 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla ClienteFrecuente

```
--Cliente Frecuente
CREATE TABLE ClienteFrecuente(
  curpClienteFrecuente CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) CHECK(curp SIMILAR TO '[a-z]{4}[0-9]{6}[a-z]{8}') UNIQUE,
  idCaja CHAR(2) NOT NULL,
  idClinica CHAR(5) NOT NULL,
  nombre VARCHAR(50) NOT NULL CHECK(nombre <> ''),
  paterno VARCHAR(50) NOT NULL CHECK(paterno <> ''),
  materno VARCHAR(50) NOT NULL CHECK(materno <> ''),
  numero CHAR(4) NOT NULL CHECK(numero SIMILAR TO '[0-9]{4}'),
  calle VARCHAR NOT NULL CHECK(calle <> ''),
  estado VARCHAR NOT NULL CHECK(estado <> ''),
  cp CHAR(5) NOT NULL CHECK(cp SIMILAR TO '[0-9]{5}'),
  genero CHAR(1) NOT NULL CHECK(genero = 'M' OR genero = 'F'),
  nacimiento DATE NOT NULL CHECK(nacimiento > '1950-01-01' AND nacimiento < '2022-12-31'),
  telefono CHAR(10) NOT NULL CHECK(telefono SIMILAR TO '[0-9]{10}'),
  email VARCHAR(50) NOT NULL CHECK(email SIMILAR TO '[0-z]{0,30}[@]{1}[a-z]{0,10}[.]{1}[a-z]{2,5}')
);
```

curpClienteFrecuente: Es la llave primaria de la tabla ClienteFrecuente.

- No aceptamos valor null puesto que es la llave primaria.
- La curp está obligada a tener una longitud de 18 caracteres, donde verificamos que los primeros 4 caracteres sean letras, los siguientes 6 caracteres dígitos y los 8 caracteres restantes letras.
- La curp es única para cada persona, por ello que lo limitamos a que fuera único.

idCaja, idClinica: Es una llave compuesta foránea en la tabla ClienteFrecuente que nos permitirá ligar a la caja en la relación “Cobrar” con la entidad ClienteFrecuente. Sus restricciones son las mismas que se definieron en la tabla **Caja**.

Tiene las mismas especificaciones y restricciones que la tabla ClienteNormal debido a que heredan del mismo padre. Sólo se agrega el atributo “email”

email: Es el correo electrónico del cliente frecuente.

- Debido a que el email puede ser muy variado establecimos su tipo de dato como varchar donde debe tener máximo 50 caracteres.
- Verificamos que el email tenga el caracter '@' y un '.' en su estructura.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla Fisico

```
--Fisico
CREATE TABLE Fisico(
    idFisico CHAR(8) NOT NULL CHECK(idFisico <> '') CHECK(idFisico SIMILAR TO '[0-9]{4}[a-z]{4}') UNIQUE,
    curpClienteNormal CHAR(18) NOT NULL,
    curpClienteFrecuente CHAR(18) NOT NULL
);
```

idFisico: Es la llave primaria de la tabla Fisico.

- No aceptamos valor null puesto que es la llave primaria.
- El id está obligado a tener una longitud de 8 caracteres, donde verificamos que los primeros 4 caracteres sean dígitos (representan el día DD y el mes MM del pago) y los 4 caracteres restantes son letras (2 representan las primeras letras del nombre del cliente y 2 del apellido paterno del cliente).
- El id es único para cada pago físico, por ello que lo limitamos a que fuera único.

curpClienteNormal: Es una llave foránea en la tabla Fisico que nos permitirá ligar a un cliente Normal en la relación “Tener” con la entidad Fisico (el cual hace referencia a tener un método de pago físico). Sus restricciones son las mismas que se definieron para “curpClienteNormal” en la tabla **ClienteNormal**.

curpClienteFrecuente: Es una llave foránea en la tabla Fisico que nos permitirá ligar a un cliente Frecuente en la relación “Tener” con la entidad Fisico (el cual hace referencia a tener un método de pago físico). Sus restricciones son las mismas que se definieron para “curpClienteFrecuente” en la tabla **ClienteFrecuente**.

Tabla Internet

```
--Internet
CREATE TABLE Internet(
    numTarjeta CHAR(16) NOT NULL CHECK(numTarjeta SIMILAR TO '[0-9]{16}') UNIQUE,
    curpClienteNormal CHAR(18) NOT NULL,
    curpClienteFrecuente CHAR(18) NOT NULL,
    nombreTitular VARCHAR(50) NOT NULL CHECK(nombreTitular <> ''),
    vencimiento DATE NOT NULL,
    cvv CHAR(4) NOT NULL CHECK(cvv SIMILAR TO '[0-9]{4}')
);
```

numTarjeta: Es la llave primaria de la tabla Internet.

- No aceptamos valor null puesto que es la llave primaria.
- El número de tarjeta está obligado a tener una longitud de 16 caracteres, donde verificamos que los caracteres sean dígitos positivos.
- El número de tarjeta es único para cada pago en internet, por ello que lo limitamos a que fuera único.

curpClienteNormal: Es una llave foránea en la tabla Internet que nos permitirá ligar a un cliente Normal en la relación “Tener” con la entidad Internet (el cual hace referencia a tener un método de

pago por internet). Sus restricciones son las mismas que se definieron para “curpClienteNormal” en la tabla **ClienteNormal**.

curpClienteFrecuente: Es una llave foránea en la tabla Internet que nos permitirá ligar a un cliente Frecuente en la relación “Tener” con la entidad Internet (el cual hace referencia a tener un método de pago por internet). Sus restricciones son las mismas que se definieron para “curpClienteFrecuente” en la tabla **ClienteFrecuente**.

nombreTitular: Es el nombre del titular de la tarjeta.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 50 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

vencimiento: Es la fecha de vencimiento de la tarjeta.

- Este dato es de tipo DATE debido a las facilidades de formato que nos brinda.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

cvv: Es el código de seguridad de la Tarjeta.

- El código está obligado a tener una longitud de 4 caracteres. Se verifica que estos caracteres sean dígitos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla Veterinario

```
--Veterinario
CREATE TABLE Veterinario(
  curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) CHECK(curp SIMILAR TO '[a-z]{4}[0-9]{6}[a-z]{8}') UNIQUE,
  idClinica CHAR(5) NOT NULL,
  nombre VARCHAR(50) NOT NULL CHECK(nombre <> ''),
  paterno VARCHAR(50) NOT NULL CHECK(paterno <> ''),
  materno VARCHAR(50) NOT NULL CHECK(materno <> ''),
  calle VARCHAR NOT NULL CHECK(calle <> ''),
  numero CHAR(4) NOT NULL CHECK(numero SIMILAR TO '[0-9]{4}'),
  estado VARCHAR NOT NULL CHECK(estado <> ''),
  cp CHAR(5) NOT NULL CHECK(cp SIMILAR TO '[0-9]{5}'),
  genero CHAR(1) NOT NULL CHECK(genero = 'M' OR genero = 'F'),
  nacimiento DATE NOT NULL CHECK(nacimiento > '1950-01-01' AND nacimiento < '2022-12-31'),
  telefono CHAR(10) NOT NULL CHECK(telefono SIMILAR TO '[0-9]{10}'),
  salario FLOAT NOT NULL CHECK(salario > 0),
  rfc CHAR(13) NOT NULL CHECK(CHAR_LENGTH(rfc) = 13),
  numPacientes INT NOT NULL CHECK(numPacientes>0),
  horarioInicio TIME NOT NULL,
  horarioFin TIME NOT NULL
);
```

curp: Es la llave primaria de la tabla Veterinario.

- No aceptamos valor null puesto que es la llave primaria.
- La curp está obligada a tener una longitud de 18 caracteres, donde verificamos que los primeros 4 caracteres sean letras, los siguientes 6 caracteres dígitos y los 8 caracteres restantes letras.

- La curp es única para cada persona, por ello que lo limitamos a que fuera único.

idClinica: Es una llave foránea en la tabla Veterinario que nos permitirá ligar a la clínica en la relación “Trabajar” con la entidad Veterinario. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

Tiene las mismas especificaciones y restricciones que la tabla ClienteNormal debido a que heredan del mismo padre.

Solo se agregan los siguientes atributos:

salario: Es la remuneración que recibe el veterinario por parte de la empresa, por brindar sus servicios.

- Este dato es de tipo FLOAT debido a que un salario esta sujeto a impuestos, bonos, inflación, etc. y no siempre será un entero.
- El salario debe de ser mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

rfc: Es el RFC del Veterinario.

- El RFC está obligado a tener una logitud de 13 caracteres alfanuméricos.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

numPacientes: Es el número de pacientes que ha atendido el veterinario.

- Este dato es de tipo INT debido a los pacientes se manejan como unidades enteras.
- Verificamos que el número de unidades sea mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

horarioInicio: Es el horario en el que el veterinario empieza a laborar.

- Este dato es de tipo TIME debido a las facilidades de formato que nos brinda.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

horarioFin: Es el horario en el que el veterinario termina de laborar.

- Este dato es de tipo TIME debido a las facilidades de formato que nos brinda.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla Estilista

```
--Estilista
CREATE TABLE Estilista(
  curp CHAR(18) NOT NULL CHECK(CHAR_LENGTH(curp) = 18) CHECK(curp SIMILAR TO '[a-z]{4}[0-9]{6}[a-z]{8}') UNIQUE,
  idClinica CHAR(5) NOT NULL,
  nombre VARCHAR(50) NOT NULL CHECK(nombre <> ''),
  paterno VARCHAR(50) NOT NULL CHECK(paterno <> ''),
  materno VARCHAR(50) NOT NULL CHECK(materno <> ''),
  calle VARCHAR NOT NULL CHECK(calle <> ''),
  numero CHAR(4) NOT NULL CHECK(numero SIMILAR TO '[0-9]{4}'),
  estado VARCHAR NOT NULL CHECK(estado <> ''),
  cp CHAR(5) NOT NULL CHECK(cp SIMILAR TO '[0-9]{5}'),
  genero CHAR(1) CHECK(genero = 'M' OR genero = 'F'),
  nacimiento DATE NOT NULL CHECK(nacimiento > '1950-01-01' AND nacimiento < '2022-12-31'),
  telefono CHAR(10) NOT NULL CHECK(telefono SIMILAR TO '[0-9]{10}'),
  salario FLOAT NOT NULL CHECK(salario > 0),
  numCertificado INT NOT NULL CHECK(numCertificado>0)
);
```

curp: Es la llave primaria de la tabla Estilista.

- No aceptamos valor null puesto que es la llave primaria.
- La curp está obligada a tener una longitud de 18 caracteres, donde verificamos que los primeros 4 caracteres sean letras, los siguientes 6 caracteres dígitos y los 8 caracteres restantes letras.
- La curp es única para cada persona, por ello que lo limitamos a que fuera único.

idClinica: Es una llave foránea en la tabla Estilista que nos permitirá ligar a la clínica en la relación “Trabajar” con la entidad Estilista. Sus restricciones son las mismas que se definieron para “idClinica” en la tabla **Clinica**.

Tiene las mismas especificaciones y restricciones que la tabla ClienteNormal debido a que heredan del mismo padre.

Solo se agregan los siguientes atributos:

salario: Es la remuneración que recibe el estilista por parte de la empresa, por brindar sus servicios.

- Este dato es de tipo FLOAT debido a que un salario esta sujeto a impuestos, bonos, inflación, etc. y no siempre será un entero.
- El salario debe de ser mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

numCertificado: Es el número de certificado del Estilista.

- Es de tipo INT debido a que debe de ser un entero positivo (no especificamos tamaño de dígitos).
- Verificamos que sea mayor a cero.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla Recibo

```
--Recibo
CREATE TABLE Recibo(
    idRecibo CHAR(4) NOT NULL CHECK(idRecibo SIMILAR TO '[N,E,T]{1}[0-9]{3}') UNIQUE,
    idCaja CHAR(2) NOT NULL,
    idClinica CHAR(5) NOT NULL,
    servicioBrindado VARCHAR NOT NULL CHECK(servicioBrindado <> ''),
    nombreTrabajador VARCHAR(50) NOT NULL CHECK(nombreTrabajador <> ''),
    nombreMascota VARCHAR(20) NOT NULL CHECK(nombreMascota <> ''),
    nombreDuenio VARCHAR(50) NOT NULL CHECK(nombreDuenio <> '')
);
```

idRecibo: Es una llave primaria en la tabla Recibo.

- No aceptamos valor null puesto que es la llave primaria.
- El id está obligada a tener una longitud de 4 caracteres, donde verificamos que el primer caracter sea una letra 'N' para consulta normal, 'E' para consulta de emergencia ó 'T' para tratamiento estético. Los últimos 3 caracteres son dígitos.
- La curp es única para cada persona, por ello que lo limitamos a que fuera único.

idConsulta, idMascota: Es una llave compuesta foránea en la tabla GenerarReciboNormal que nos permitirá ligar a la consulta normal en la relación “GenerarReciboNormal” con la entidad Recibo. Sus restricciones son las mismas que se definieron en la tabla **Normal**.

idCaja, idClinica: Es una llave compuesta foránea en la tabla Recibo que nos permitirá ligar a la caja en la relación “Recibir” con la entidad Recibo. Sus restricciones son las mismas que se definieron en la tabla **Caja**.

servicioBrindado: Es el servicio brindado a la Mascota.

- Debido a que el servicio puede ser muy variado establecimos su tipo de dato como varchar (no especificamos límite de caracteres).
- El servicio brindado no debe de ser una cadena vacía y no se permite que tome el valor null.

nombreTrabajador: Es el nombre del trabajador que atendió a la mascota.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 50 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

nombreMascota: Es el nombre de la mascota que se atendió.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 20 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

nombreDuenio: Es el nombre del dueño de la mascota que se atendió.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 50 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

Tabla Mascota

--Mascota

```
CREATE TABLE Mascota(  
    idMascota CHAR(5) NOT NULL CHECK(idMascota SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE,  
    curpClienteNormal CHAR(18) NOT NULL,  
    curpClienteFrecuente CHAR(18) NOT NULL,  
    nombre VARCHAR(15) NOT NULL CHECK(nombre <> ''),  
    nacimiento DATE NOT NULL CHECK(nacimiento > '1995-01-01' AND nacimiento < '2022-12-31'),  
    peso FLOAT NOT NULL CHECK(peso > 0 AND peso < 150),  
    especie VARCHAR NOT NULL CHECK(especie <> ''),  
    raza VARCHAR NOT NULL CHECK(raza <> ''),  
    nombreDeDuenio VARCHAR(50) NOT NULL CHECK(nombreDeDuenio <> '')  
);
```

idMascota: Es la llave primaria de la tabla Mascota.

- No aceptamos valor null puesto que es la llave primaria.
- El id está obligado a tener una longitud de 5 caracteres, donde verificamos que los primeros 3 caracteres son letras (son las primeras letras del nombre de la mascota), y los 2 caracteres restantes son dígitos.
- El id es único para cada mascota, por ello que lo limitamos a que fuera único.

curpClienteNormal: Es una llave foránea en la tabla Mascota que nos permitirá ligar a un cliente Normal en la relación “Tener” con la entidad Mascota (el cual hace referencia a que un cliente tiene una o más mascotas). Sus restricciones son las mismas que se definieron para “curpClienteNormal” en la tabla **ClienteNormal**.

curpClienteFrecuente: Es una llave foránea en la tabla Mascota que nos permitirá ligar a un cliente Frecuente en la relación “Tener” con la entidad Mascota (el cual hace referencia a que un cliente tiene una o más mascotas). Sus restricciones son las mismas que se definieron para “curpClienteFrecuente” en la tabla **ClienteFrecuente**.

nombre: Es el nombre de la Mascota.

- Debido a que son muy variados establecimos su tipo de dato como varchar donde cada nombre debe tener máximo 20 caracteres.
- El nombre no debe de ser una cadena vacía y no se permite que tome el valor null.

nacimiento: Es la fecha de nacimiento de la Mascota.

- Este dato es de tipo DATE, donde se verifica que las mascotas hayan nacido a partir del año de 1995 hasta el año 2022.
- Deseamos que se especifique este dato por ello no permitimos que sea null.

peso: Es el peso de la Mascota.

- Este dato es de tipo FLOAT debido a que un el peso es requerido exacto.
- Se verifica que el peso sea mayor a 0 y menor a 150.

- Deseamos que se especifique este dato por ello no permitimos que sea null.

Tabla Normal

```
--Normal
CREATE TABLE Normal(
    idConsulta CHAR(5) NOT NULL CHECK(idConsulta SIMILAR TO '[0-9]{2}') UNIQUE,
    idMascota CHAR(5) NOT NULL CHECK(idMascota SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE,
    curp CHAR(18) NOT NULL,
    precio FLOAT NOT NULL CHECK(precio > 0),
    fechaProxCita DATE NOT NULL CHECK(fechaProxCita > '2022-01-01'),
    medicamento VARCHAR NOT NULL CHECK(medicamento <> ''),
    motivoConsulta VARCHAR NOT NULL CHECK(motivoConsulta <> ''),
    estadoMascota VARCHAR NOT NULL CHECK(estadoMascota <> '')
);
```

Es una entidad débil, por lo tanto su llave primaria está compuesta de los atributos “idConsulta” y “idMascota”

idConsulta: Forma parte de la llave primaria de la tabla Normal.

- No aceptamos valor de null puesto que forma parte de la llave primaria.
- El id está obligado a tener una longitud de 2 caracteres, donde verificamos que sean dígitos positivos.
- El id es único para cada Consulta Normal, es por ello que lo limitamos a que fuera único.

idMascota: Es la llave primaria de la entidad fuerte “Mascota”. Sus restricciones son las mismas que se definieron para “idMascota” en la tabla **Mascota**.

curp: Es una llave foránea en la tabla Normal que nos permitirá ligar un veterinario en la relación “DarConsulta” con la entidad Normal (que hace referencia a una consulta normal). Sus restricciones son las mismas que se definieron para “curp” en la tabla **Veterinario**.

Tiene las mismas especificaciones y restricciones que la tabla GenerarReciboNormal.

De la tabla GenerarReciboNormal solo se excluyen los atributos “servicioDado”, “nombreTrabajador”, “nombreMascota” y “nombreDuenio”.

Tabla Emergencia

```
--Emergencia
CREATE TABLE Emergencia(
  idConsulta CHAR(5) NOT NULL CHECK(idConsulta SIMILAR TO '[0-9]{2}') UNIQUE,
  idMascota CHAR(5) NOT NULL CHECK(idMascota SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE,
  curp CHAR(18) NOT NULL,
  precio FLOAT NOT NULL CHECK(precio > 0),
  procedimientoEmitido VARCHAR NOT NULL CHECK(procedimientoEmitido <> ''),
  sintomasMascota VARCHAR NOT NULL CHECK(sintomasMascota <> ''),
  codigoEmergencia VARCHAR NOT NULL CHECK(codigoEmergencia = 'verde' OR codigoEmergencia = 'amarillo' OR codigoEmergencia = 'rojo')
);
```

Es una entidad débil, por lo tanto su llave primaria está compuesta de los atributos “idConsulta” y “idMascota”

idConsulta: Forma parte de la llave primaria de la tabla Emergencia.

- No aceptamos valor de null puesto que forma parte de la llave primaria.
- El id está obligado a tener una longitud de 2 caracteres, donde verificamos que sean dígitos positivos.
- El id es único para cada Consulta de emergencia, es por ello que lo limitamos a que fuera único.

idMascota: Es la llave primaria de la entidad fuerte “Mascota”. Sus restricciones son las mismas que se definieron para “idMascota” en la tabla **Mascota**.

curp: Es una llave foránea en la tabla Emergencia que nos permitirá ligar un veterinario en la relación “DarConsulta” con la entidad Emergencia (que hace referencia a una consulta de emergencia). Sus restricciones son las mismas que se definieron para “curp” en la tabla **Veterinario**.

Tiene las mismas especificaciones y restricciones que la tabla GenerarReciboEmergencia. De la tabla GenerarReciboEmergencia solo se excluyen los atributos “servicioDado”, “nombreTrabajador”, “nombreMascota” y “nombreDuenio”.

Tabla Tratamiento

```
--Tratamiento
CREATE TABLE Tratamiento(
  idConsulta CHAR(5) NOT NULL CHECK(idConsulta SIMILAR TO '[0-9]{2}') UNIQUE,
  idMascota CHAR(5) NOT NULL CHECK(idMascota SIMILAR TO '[A-Z]{3}[0-9]{2}') UNIQUE ,
  curp CHAR(18) NOT NULL,
  precio FLOAT NOT NULL CHECK(precio > 0),
  servicioDado VARCHAR NOT NULL CHECK(servicioDado <> '')
);
```

Es una entidad débil, por lo tanto su llave primaria está compuesta de los atributos “idConsulta” y “idMascota”

idConsulta: Forma parte de la llave primaria de la tabla Tratamiento.

- No aceptamos valor de null puesto que forma parte de la llave primaria.

- El id está obligado a tener una longitud de 2 caracteres, donde verificamos que sean dígitos positivos.
- El id es único para cada Consulta de tratamiento, es por ello que lo limitamos a que fuera único.

idMascota: Es la llave primaria de la entidad fuerte “Mascota”. Sus restricciones son las mismas que se definieron para “idMascota” en la tabla **Mascota**.

curp: Es una llave foránea en la tabla Tratamiento que nos permitirá ligar un estilista en la relación “DarConsulta” con la entidad Tratamiento (que hace referencia a una consulta de tratamiento estético). Sus restricciones son las mismas que se definieron para “curp” en la tabla **Estilista**.

Tiene las mismas especificaciones y restricciones que la tabla GenerarReciboTratamiento.

De la tabla GenerarReciboTratamiento solo se excluyen los atributos “servicioDado”, “nombreTrabajador”, “nombreMascota” y “nombreDuenio”.