



Servicio Nacional de Aprendizaje SENA

**Centro Electricidad, Electrónica y
Telecomunicaciones**



**MANUAL BASE DE DATOS
SISTEMA DE INFORMACIÓN
PARA LA GESTIÓN DE CLIENTES Y COTIZACIONES DE SEGUROS**

SENA 2022



ÍNDICE

ÍNDICE	2
INTRODUCCIÓN	4
BASE DE DATOS	4
DIAGRAMA DE CASOS DE USO	5
MODELO RELACIONAL	5
DICCIONARIO DE DATOS	6
TABLAS DE LA BASE DE DATOS	6
DDL	6
Tablas	6
/* TABLA: tipo_documento	6
/* TABLA: tipo_usuario	6
/* TABLA: estado	6
/* TABLA: tipo_usuario	7
/* TABLA: usuario	7
/* TABLA: notificacion	7
/* TABLA: notificacion_usuario	7
/* TABLA: departamento	8
/* TABLA: ciudad	8
/* TABLA: cliente	8
/* TABLA: aseguradora	9
/* TABLA: seguro	9
/* TABLA: seguro_aseguradora	9
/* TABLA: cotizacion	9
/* TABLA: cobertura	10
/* TABLA: aseguradora_cotizante	10
/* TABLA: cobertura_cotizacion	10
/* TABLA: poliza	11
/* TABLA: siniestro	11
/* TABLA: reporte	11
Vistas	12
/* Vista: view_usuario	12
/* Vista: view_cliente	12
/* Vista: view_cotizacon	14



Servicio Nacional de Aprendizaje SENA

**Centro Electricidad, Electrónica y
Telecomunicaciones**



/* Vista: view_poliza	15
/* Vista: view_siniestro	16
/* Vista: view_reportes	17
Rutinas	18
/* Rutina: sp_insert_usuario	18
/* Rutina: sp_insert_cliente	19
/* Rutina: sp_update_cliente	21
/* Rutina: sp_delete_cliente	22
/* Rutina: sp_insert_cotizacion	23
/* Rutina: sp_update_cotizacion	24
/* Rutina: sp_insert_poliza	24
/* Rutina: sp_update_poliza	25
/* Rutina: sp_insert_siniestro	26
/* Rutina: sp_update_siniestro	27
CONTROL DE DOCUMENTO	28



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



INTRODUCCIÓN

En el documento se presentará la base de datos del proyecto Pérez Lara CIA, de acuerdo a esto, se realiza una descripción detallada de las tablas registradas en la base de datos, el tipo de campo contenido y sus relaciones (llaves foráneas).

se visualiza las líneas de código utilizadas para la creación de las tablas, relaciones, inserción de datos, rutinas y funcionalidades de conexión entre tablas.

BASE DE DATOS

La base de datos consta de 19 tablas que almacenan información de cadenas largas de caracteres, en campos (Text), también hay vistas que almacenan datos como información de los usuarios, campos (Varchar) con diversos tamaños dependiendo del dato y campos tipo (int).

aseguradora	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	7	InnoDB	utf8mb4_general_ci	16.0 KB	-
aseguradora_cotizante	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	17	InnoDB	utf8mb4_general_ci	48.0 KB	-
ciudad	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	1,118	InnoDB	utf8mb4_general_ci	96.0 KB	-
cliente	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	7	InnoDB	utf8mb4_general_ci	64.0 KB	-
cobertura	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	12	InnoDB	utf8mb4_general_ci	16.0 KB	-
cobertura_cotizacion	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	51	InnoDB	utf8mb4_general_ci	48.0 KB	-
cotizacion	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	7	InnoDB	utf8mb4_general_ci	64.0 KB	-
departamento	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	33	InnoDB	utf8mb4_general_ci	16.0 KB	-
estado	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	11	InnoDB	utf8mb4_general_ci	16.0 KB	-
notificacion	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_general_ci	16.0 KB	-
notificacion_usuario	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	0	InnoDB	utf8mb4_general_ci	64.0 KB	-
poliza	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	7	InnoDB	utf8mb4_general_ci	80.0 KB	-
reporte	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	7	InnoDB	utf8mb4_general_ci	32.0 KB	-
seguro	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	7	InnoDB	utf8mb4_general_ci	16.0 KB	-
seguro_aseguradora	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	15	InnoDB	utf8mb4_general_ci	48.0 KB	-
sinistro	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	7	InnoDB	utf8mb4_general_ci	48.0 KB	-
tipo_documento	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	4	InnoDB	utf8mb4_general_ci	16.0 KB	-
tipo_usuario	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	2	InnoDB	utf8mb4_general_ci	16.0 KB	-
usuario	★	Examinar	Estructura	Buscar	Insertar	Vaciar	Eliminar	5	InnoDB	utf8mb4_general_ci	80.0 KB	-

también tiene un total de 6 views o vistas las cuales se explicaran más adelante, estas son consultas de inner join que nos servirán para mostrar mejor explicado nuestros datos



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y Telecomunicaciones

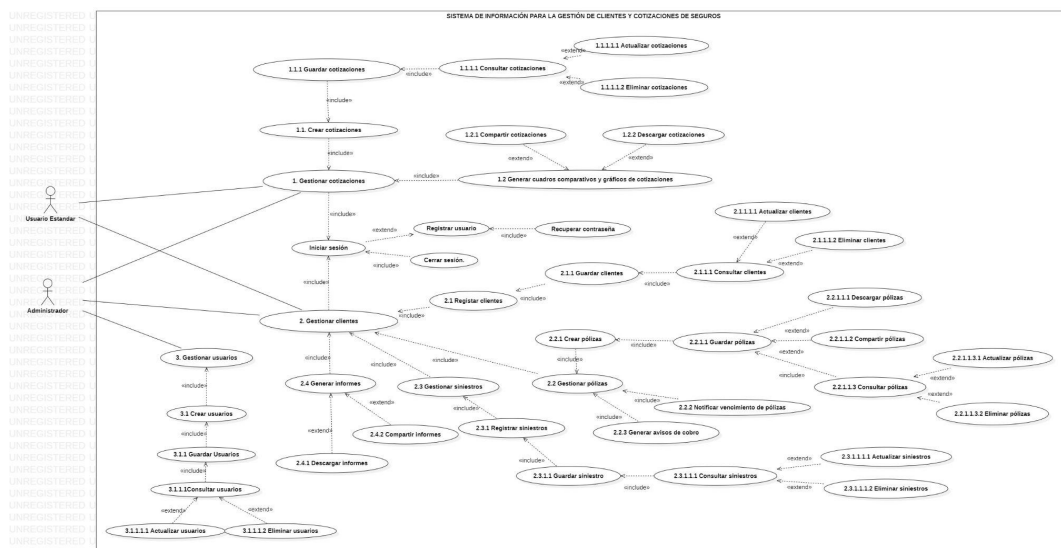


Tabla	Acción	Filas	Tipo
view_cliente	★ Examinar Estructura Buscar Insertar Editar Eliminar ~0 Visualizar		
view_cotizacion	★ Examinar Estructura Buscar Insertar Editar Eliminar ~0 Visualizar		
view_poliza	★ Examinar Estructura Buscar Insertar Editar Eliminar ~0 Visualizar		
view_reporte	★ Examinar Estructura Buscar Insertar Editar Eliminar ~0 Visualizar		
view_siniestro	★ Examinar Estructura Buscar Insertar Editar Eliminar ~0 Visualizar		
view_usuario	★ Examinar Estructura Buscar Insertar Editar Eliminar ~0 Visualizar		

también tiene por el momento un total de 10 rutinas almacenadas lo que nos mejora la velocidad para hacer el Crud desde nuestro proyecto

Nombre	Acción	Tipo	Retorna
sp_delete_cliente	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_insert_cliente	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_insert_cotizacion	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_insert_poliza	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_insert_siniestro	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_insert_usuario	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_update_cliente	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_update_cotizacion	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_update_poliza	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	
sp_update_siniestro	✎ Editar ▶ Ejecutar 📄 Exportar 🗑 Eliminar	PROCEDURE	

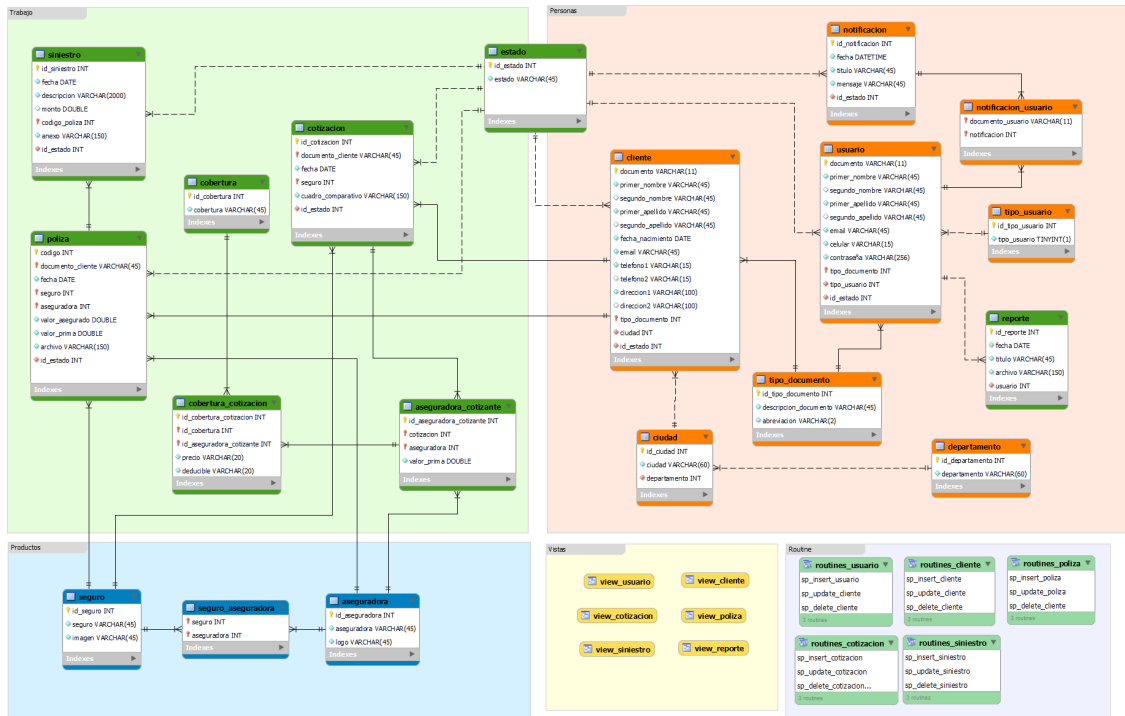
DIAGRAMA DE CASOS DE USO



Enlace:

<https://drive.google.com/file/d/1AXSsl23ve9HIQX3D93DJxH7Fye3RQzU4/view?usp=sharing>

MODELO RELACIONAL



Enlace:

https://drive.google.com/file/d/1ZwKJeFk_2776tvck8QLuNu_rjSille7W/view?usp=sharing

DICCIONARIO DE DATOS

El Diccionario de Datos es un listado de datos organizado que define de manera rigurosa, las características lógicas y puntuales tales como nombre, tipo de contenido y descripción de negocio, de tal forma que se cuente con un elemento común, para el entendimiento de la base de datos, y se utiliza en un sistema de información.

Enlace:

https://drive.google.com/file/d/1u0k8JRE5JlSt1SCF_VpAbj3FaW_mRwC9/view?usp=sharing



TABLAS DE LA BASE DE DATOS

DDL

Tablas

```
/* TABLA: tipo_documento
===== */
drop database if exists Perez_Lara_Cia_LTDA;
create database if not exists Perez_Lara_Cia_LTDA;
use Perez_Lara_Cia_LTDA;
/* TABLA: tipo_usuario
===== */
drop table if exists tipo_documento;
create table if not exists tipo_documento
(
    id_tipo_documento    int auto_increment primary key,
    descripcion_documento varchar(45) not null,
    abreviacion          varchar(2)
);

/* TABLA: estado
===== */
drop table if exists estado;
create table if not exists estado
(
    id_estado int auto_increment primary key,
    estado    varchar(45) not null
);

/* Usuarios
===== */
/* TABLA: tipo_usuario
===== */
drop table if exists tipo_usuario;
create table if not exists tipo_usuario
(
    id_tipo_usuario int auto_increment primary key,
    tipo_usuario    tinyint(1) not null
);

/* TABLA: usuario
===== */
drop table if exists usuario;
create table if not exists usuario
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



```
(
    documento      varchar(20) primary key,
    primer_nombre   varchar(45) not null,
    segundo_nombre  varchar(45),
    primer_apellido varchar(45) not null,
    segundo_apellido varchar(45),
    celular         varchar(15) not null,
    email           varchar(45) not null unique,
    clave           varchar(256) not null,
    foto_perfil     varchar(150) not null,
    tipo_documento  int      not null,
    tipo_usuario    int      not null,
    estado          int      not null,
    foreign key (tipo_documento) references tipo_documento (id_tipo_documento),
    foreign key (tipo_usuario) references tipo_usuario (id_tipo_usuario),
    foreign key (estado) references estado (id_estado)
);

/* TABLA: notificacion
===== */
drop table if exists notificacion;
create table if not exists notificacion
(
    id_notificacion int auto_increment primary key,
    fecha          datetime not null,
    titulo         varchar(45),
    mensaje        varchar(45)
);

/* TABLA: notificacion_usuario
===== */
drop table if exists notificacion_usuario;
create table if not exists notificacion_usuario
(
    id_notificacion_usuario int auto_increment primary key,
    documento_usuario      varchar(11) not null,
    notificacion            int      not null,
    estado                  int      not null,
    foreign key (documento_usuario) references usuario (documento) on update cascade,
    foreign key (notificacion) references notificacion (id_notificacion),
    foreign key (estado) references estado (id_estado)
);

/* Clientes
===== */

/* TABLA: departamento
===== */
drop table if exists departamento;
```




Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



create table if not exists departamento

```
(  
    id_departamento int auto_increment primary key,  
    departamento varchar(60) not null  
);
```

```
/* TABLA: ciudad  
===== */
```

drop table if exists ciudad;

create table if not exists ciudad

```
(  
    id_ciudad int auto_increment primary key,  
    ciudad varchar(60) not null,  
    departamento int not null,  
    foreign key (departamento) references departamento (id_departamento)  
);
```

```
/* TABLA: cliente  
===== */
```

drop table if exists cliente;

create table if not exists cliente

```
(  
    documento varchar(20) primary key,  
    primer_nombre varchar(45) not null,  
    segundo_nombre varchar(45),  
    primer_apellido varchar(45) not null,  
    segundo_apellido varchar(45),  
    fecha_nacimiento date not null,  
    email varchar(45) not null,  
    telefono1 varchar(15) not null,  
    telefono2 varchar(15),  
    direccion1 varchar(100) not null,  
    direccion2 varchar(100),  
    tipo_documento int not null,  
    ciudad int not null,  
    estado int not null,  
    fecha_ingreso date not null,  
    foreign key (tipo_documento) references tipo_documento (id_tipo_documento),  
    foreign key (ciudad) references ciudad (id_ciudad),  
    foreign key (estado) references estado (id_estado)  
);
```

```
/* Productos  
===== */
```

```
/* TABLA: aseguradora  
===== */
```

drop table if exists aseguradora;

create table if not exists aseguradora



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



```
(
    id_aseguradora int auto_increment primary key,
    aseguradora    varchar(45) not null,
    logo           varchar(150) not null
);

/* TABLA: seguro
===== */
drop table if exists seguro;
create table if not exists seguro
(
    id_seguro int auto_increment primary key,
    seguro    varchar(45) not null,
    imagen    varchar(150) not null
);

/* TABLA: seguro_aseguradora
===== */
drop table if exists seguro_aseguradora;
create table if not exists seguro_aseguradora
(
    id_seguro_aseguradora int auto_increment primary key,
    seguro                int not null,
    aseguradora            int not null,
    foreign key (seguro) references seguro (id_seguro),
    foreign key (aseguradora) references aseguradora (id_aseguradora)
);

/* Trabajo
===== */
/* TABLA: cotizacion
===== */
drop table if exists cotizacion;
create table if not exists cotizacion
(
    id_cotizacion    int auto_increment primary key,
    documento_cliente varchar(20) not null,
    fecha            date          not null,
    seguro           int           not null,
    cuadro_comparativo varchar(150) not null,
    estado           int           not null,
    fecha_actualizacion date        null,
    foreign key (documento_cliente) references cliente (documento) on update cascade,
    foreign key (seguro) references seguro (id_seguro),
    foreign key (estado) references estado (id_estado)
);

/* TABLA: cobertura
===== */
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



drop table if exists cobertura;

create table if not exists cobertura

```
(
  id_cobertura int auto_increment primary key,
  cobertura varchar(130)
);

/* TABLA: aseguradora_cotizante
===== */
```

drop table if exists aseguradora_cotizante;

create table if not exists aseguradora_cotizante

```
(
  id_aseguradora_cotizante int auto_increment primary key,
  cotizacion int not null,
  aseguradora int not null,
  valor_prima double not null,
  foreign key (cotizacion) references cotizacion (id_cotizacion),
  foreign key (aseguradora) references aseguradora (id_aseguradora)
);

/* TABLA: cobertura_cotizacion
===== */
```

drop table if exists cobertura_cotizacion;

create table if not exists cobertura_cotizacion

```
(
  id_cobertura_cotizacion int auto_increment primary key,
  id_aseguradora_cotizante int not null,
  id_cobertura int not null,
  precio varchar(20) not null,
  deducible varchar(20) not null,
  foreign key (id_aseguradora_cotizante) references
aseguradora_cotizante(id_aseguradora_cotizante),
  foreign key (id_cobertura) references cobertura(id_cobertura)
);

/* TABLA: poliza
===== */
```

drop table if exists poliza;

create table if not exists poliza

```
(
  codigo int primary key,
  documento_cliente varchar(20) not null,
  fecha date not null,
  seguro int not null,
  aseguradora int not null,
  valor_asegurado double not null,
  valor_prima double not null,
  archivo varchar(150) not null,
  estado int not null,
```



```
fecha_actualizacion date      null,  
foreign key (documento_cliente) references cliente (documento) on update cascade,  
foreign key (seguro) references seguro (id_seguro),  
foreign key (aseguradora) references aseguradora (id_aseguradora),  
foreign key (estado) references estado (id_estado)  
);  
  
/* TABLA: siniestro  
===== */  
  
drop table if exists siniestro;  
create table if not exists siniestro  
(  
    id_siniestro int auto_increment primary key,  
    fecha      date      not null,  
    descripcion varchar(2000) not null,  
    monto      DOUBLE,  
    codigo_poliza int      not null,  
    anexo      varchar(150) not null,  
    estado     int        not null,  
    foreign key (codigo_poliza) references poliza (codigo) on update cascade,  
    foreign key (estado) references estado (id_estado)  
);
```

```
/* TABLA: reporte  
===== */  
  
drop table if exists reporte;  
create table if not exists reporte  
(  
    id_reporte int auto_increment primary key,  
    fecha     date      not null,  
    titulo    varchar(45) not null,  
    archivo   varchar(150) not null,  
    usuario   varchar(20) not null,  
    foreign key (usuario) references usuario (documento)  
);
```

Vistas

```
/* Vista: view_usuario  
===== */  
  
drop view if exists view_usuario;  
create view view_usuario as  
select concat(primer_nombre, ' ', segundo_nombre, ' ', primer_apellido, ' ',  
segundo_apellido) as Nombre,  
    email as Email,  
    celular as Celular,
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



```
t2.tipo_usuario as Rol
from usuario
inner join tipo_documento t on usuario.tipo_documento = t.id_tipo_documento
inner join tipo_usuario t2 on usuario.tipo_usuario = t2.id_tipo_usuario
inner join estado e on usuario.estado = e.id_estado
order by Nombre;
```

```
SELECT * FROM `view_usuario`
```

Nombre	Email	Celular	Rol
Admin Perez Lara	admin@perezlara.com	0	0

esta vista nos trae los campos relacionados
usuario.tipo_documento = t.id_tipo_documento es la llave foránea que une las tablas tipo_documento y usuario.

usuario.tipo_usuario = t2.id_tipo_usuario es la llave foránea que une las tablas tipo_usuario y usuario.

usuario.estado_usuario = e.id_estado_usuario es la llave foránea que une las tablas estado_usuario t usuario

```
/* Vista: view_cliente
===== */
drop view if exists view_cliente;
create view view_cliente as
select concat(t.abreviacion, ' ', documento) as `No. Documento`,
concat(primer_nombre, ' ', segundo_nombre, ' ', primer_apellido, ' ', segundo_apellido) as Cliente,
d.departamento as Departamento,
c.ciudad as Ciudad,
concat(direccion1, ' ', direccion2) as Direccion,
fecha_nacimiento as `Fecha de Nacimiento`,
email as Email,
concat(telefono1, ' ', telefono2) as Telefono,
fecha_ingreso as `Fecha ingreso`,
e.estado as Estado
from cliente
inner join tipo_documento t on cliente.tipo_documento = t.id_tipo_documento
inner join ciudad c on cliente.ciudad = c.id_ciudad
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



inner join departamento d on c.departamento = d.id_departamento
inner join estado e on cliente.estado = e.id_estado
where cliente.estado <> 2;

```
SELECT * FROM `view_cliente`
```

No. Documento	Cliente	Departamento	Ciudad	Direccion	Fecha de Nacimiento	Email	Telefono	Estado
CC 1010760139	David Santiago caro escobar	Amazonas	La Pedrera	cra 200 calle 100	2004-02-24	santiagocar0816@gmail.com	31182215 5109816	Activo
CC 51658418	oscar gyovanny lopez	Amazonas	El Encanto	cra 15 calle 100	2022-03-19	oscar@gmail.com	31528454	Activo
CC 5196149	katherine Cárdenas	Amazonas	El Encanto	cra 125	2003-07-16	khaterine@gmail.com	846848812	Activo
CC 564684	camila andrea sanchez	Amazonas	El Encanto	cra 98	2022-03-04	camila@gmail.com	51814984	Activo
CE 8486414	juanes fernando urrea	Amazonas	La Pedrera	cra 200 calle 100	2022-03-03	daniel@gmail.com	31182215	Activo
CE 5184184	juan sebastian alarcon	Amazonas	El Encanto	cra 15 calle 100	2022-03-01	juan@gmail.com	31182215	Inactivo
CC 585818	David fernando urrea daza	Amazonas	El Encanto	cra 98b	2022-03-18	juan@gmail.com	2198259	Inactivo

En esta vista nos trae los campos relacionados con llaves foráneas
cliente.tipo_documento = t.id_tipo_documento es es la llave foránea que une
las tablas tipo_documento y cliente.

cliente.ciudad = c.id_ciudad es la llave foránea que une las tablas
ciudad y cliente.

c.departamento = d.id_departamento es la llave foránea que une las tablas
departamento y ciudad

cliente.estado_cliente = e.id_estado_cliente es la llave foránea que une las
tablas estado_cliente y cliente.

```
/* Vista: view_cotizacon
===== */
drop view if exists view_cotizacion;
create view view_cotizacion as
select id_cotizacion                                as `No. Cotizacion`,
       fecha                                          as Fecha,
       concat(c.primer_nombre, ' ', c.segundo_nombre, ' ', c.primer_apellido, ' ',
c.segundo_apellido) as Cliente,
       s.seguro                                       as Seguro,
       group_concat(a2.logo)                         as Aseguradora,
       cuadro_comparativo                           as `Cuadro
comparativo`,
       e.estado                                       as Estado
from cotizacion
       inner join cliente c on cotizacion.documento_cliente = c.documento
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



```
inner join seguro s on cotizacion.seguro = s.id_seguro
inner join aseguradora_cotizante a on cotizacion.id_cotizacion = a.cotizacion
inner join aseguradora a2 on a.aseguradora = a2.id_aseguradora
inner join estado e on cotizacion.estado = e.id_estado
group by id_cotizacion;
```

```
SELECT * FROM `view_cotizacion`
```

No. Cotizacion	Fecha	Cliente	Seguro	Aseguradora	Cuadro comparativo	Estado
1	2022-04-01	camila andrea sanchez	Arrendamiento	http://localhost/Perez%20Lara/icons/aseguradoras/A...	cuadro comparativo	Aceptada
2	2022-03-02	David fernando urrea daza	Autos	http://localhost/Perez%20Lara/icons/aseguradoras/A...	cuadro comparativo	No vigente
3	2022-03-23	juan sebastian alarcon	Copropiedades	http://localhost/Perez%20Lara/icons/aseguradoras/S...	cuadro comparativo	Rechazada
4	2022-02-04	katherine Cárdenas	Hogar	http://localhost/Perez%20Lara/icons/aseguradoras/H...	cuadro comparativo	Aceptada
5	2022-02-05	juanes fernando urrea	Mascotas	http://localhost/Perez%20Lara/icons/aseguradoras/S...	cuadro comparativo	No vigente
6	2022-01-06	oscar gyovanny lopez	Responsabilidad Civil	http://localhost/Perez%20Lara/icons/aseguradoras/A...	cuadro comparativo	Rechazada
7	2021-01-08	David Santiago caro escobar	Vida	http://localhost/Perez%20Lara/icons/aseguradoras/H...	cuadro comparativo	Vigente

En esta vista nos trae los campos relacionados con llaves foráneas
cotizacion.documento_cliente = c.documento Documento es la llave foránea
que une las tablas cotizacion y cliente.

cotizacion.seguro = s.id_seguro id_seguro es la llave foránea que une las
tablas cotización y seguro.

cotizacion.id_cotizacion = a.cotizacion id_cotizacion es la llave foranea que
une las tablas aseguradora_cotizante y cotizacion.

a.aseguradora = a2.id_aseguradora id_aseguradora es es la llave foránea
que une las tablas aseguradora y aseguradora.

cotizacion.estado_cotizacion = e.id_estado_cotizacion es es la llave foránea
que une las tablas es es la llave foránea que une las tablas estado_cotizacion y
cotizacion.

```
/* Vista: view_poliza
===== */
```

```
drop view if exists view_poliza;
create view view_poliza as
select codigo                                as `Codigo poliza`,
       fecha                                as Fecha,
       concat(primer_nombre, ' ', segundo_nombre, ' ', primer_apellido, ' ', segundo_apellido)
as Cliente,
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



```
s.seguro as Seguro,
a.logo as Aseguradora,
archivo as Archivo,
valor_asegurado as `Valor asegurado`,
valor_prima as `Valor prima`,
e.estado as Estado
from poliza
inner join cliente c on poliza.documento_cliente = c.documento
inner join seguro s on poliza.seguro = s.id_seguro
inner join aseguradora a on poliza.aseguradora = a.id_aseguradora
inner join estado e on poliza.estado = e.id_estado
order by `Codigo poliza`;
```

```
SELECT * FROM `view_poliza`
```

Codigo poliza	Fecha	Cliente	Seguro	Aseguradora	Archivo	Valor asegurado	Valor prima	Estado
13	2022-04-01	David fernando urrea daza	Vida	http://localhost/Perez%20Lara/icons/aseguradoras/H...	Póliza	1000000	10000	Vigente
213	2022-03-02	juan sebastian alarcon	Responsabilidad Civil	http://localhost/Perez%20Lara/icons/aseguradoras/L...	Póliza	323123	4544	Vigente
214	2022-03-23	katherine Cárdenas	Mascotas	http://localhost/Perez%20Lara/icons/aseguradoras/S...	Póliza	86766587	13123	Proxima a vencer
219	2021-01-08	camila andrea sanchez	Arrendamiento	http://localhost/Perez%20Lara/icons/aseguradoras/M...	Póliza	3545435453	5454545	Vigente
456	2022-02-04	juanes fernando urrea	Hogar	http://localhost/Perez%20Lara/icons/aseguradoras/A...	Póliza	90000000	3456000	Vigente
476	2022-02-05	oscar gyovanny lopez	Copropiedades	http://localhost/Perez%20Lara/icons/aseguradoras/S...	Póliza	2300000	100000	No vigente
643	2022-01-06	David Santiago caro escobar	Autos	http://localhost/Perez%20Lara/icons/aseguradoras/Z...	Póliza	100000000	3000000	No vigente

En esta vista nos trae los campos relacionados con llaves foráneas

poliza.documento_cliente = c.documento es la llave foránea que une las tablas cliente y poliza.

poliza.seguro = s.id_seguro Es la llave foránea que une las tablas seguro y poliza.

poliza.aseguradora = a.id_aseguradora Es la llave foránea que une las tablas aseguradora y poliza.

poliza.estado_poliza = e.id_estado_poliza Es la llave foránea que une las tablas estado_poliza y poliza.

```
/* Vista: view_siniestro
===== */
```




Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



```
drop view if exists view_siniestro;
create view view_siniestro as
select id_siniestro, siniestro.fecha, siniestro.codigo,
concat(primer_nombre, ' ', segundo_nombre, ' ', primer_apellido, ' ', segundo_apellido)
as Cliente,
s.seguro, a.logo, anexo, descripcion, e.estado
from siniestro
inner join poliza p on siniestro.codigo_poliza = p.codigo
inner join seguro s on p.seguro = s.id_seguro
inner join aseguradora a on p.aseguradora = a.id_aseguradora
inner join cliente c on p.documento_cliente = c.documento
inner join estado e on siniestro.estado = e.id_estado
order by `No. Referencia`;
```

SELECT * FROM `view_siniestro`

No. Referencia	Fecha	Codigo poliza	Cliente	Seguro	Aseguradora	anexo	Observaciones	Estado
1	2022-01-05	13	David fernando urrea daza	Vida	http://localhost/Perez%20Lara/icons/aseguradoras/H...	Anexo	Accidente automovilistico	Pendiente
2	2022-02-05	213	juan sebastian alarcon	Responsabilidad Civil	http://localhost/Perez%20Lara/icons/aseguradoras/L...	Anexo	Accidente automovilistico	Pendiente
3	2022-03-05	214	katherine Cárdenas	Mascotas	http://localhost/Perez%20Lara/icons/aseguradoras/S...	Anexo	Accidente automovilistico	Completado
4	2022-04-05	456	juanes fernando urrea	Hogar	http://localhost/Perez%20Lara/icons/aseguradoras/A...	Anexo	Accidente automovilistico	Pendiente
5	2022-05-05	476	oscar gyovanny lopez	Copropiedades	http://localhost/Perez%20Lara/icons/aseguradoras/S...	Anexo	Accidente automovilistico	Urgente
6	2022-06-05	643	David Santiago caro escobar	Autos	http://localhost/Perez%20Lara/icons/aseguradoras/Z...	Anexo	Accidente automovilistico	Completado
7	2022-07-05	219	camila andrea sanchez	Arrendamiento	http://localhost/Perez%20Lara/icons/aseguradoras/M...	Anexo	Accidente automovilistico	Pendiente

En esta vista nos trae los campos relacionados con llaves foráneas

`siniestro.codigo_poliza = p.codigo` Es la llave foránea que une las tablas poliza y siniestro.

`p.seguro = s.id_seguro` Es la llave foránea que une las tablas seguro y poliza.

`p.aseguradora = a.id_aseguradora` Es la llave foránea que une las tablas aseguradora y poliza.



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



p.documento_cliente = c.documento Es la llave foránea que une las tablas cliente y poliza.

siniestro.estado_siniestro = e.id_estado_siniestro Es la llave foránea que une las tablas estado_siniestro y siniestro.

```
/* Vista: view_reportes  
===== */
```

```
drop view if exists view_reporte;  
create view view_reporte as  
select id_reporte, fecha, titulo, archivo, concat(u.primer_nombre, ' ', u.segundo_nombre, ' ', u.primer_apellido, ' ',  
u.segundo_apellido) as `No. Reporte`,  
as Fecha,  
as Nombre,  
as `Reporte`,  
from reporte  
inner join usuario u on reporte.usuario = u.documento  
order by `No. Reporte`;
```

```
SELECT * FROM `view_reporte`
```

No. Reporte	Fecha	Nombre	Reporte	Generado por
1	2022-01-05	Informe mensual	archivo	Admin Perez Lara
2	2022-02-05	Informe mensual	archivo	Admin Perez Lara
3	2022-03-05	Informe mensual	archivo	Admin Perez Lara
4	2022-04-05	Informe mensual	archivo	Admin Perez Lara
5	2022-05-05	Informe mensual	archivo	Admin Perez Lara
6	2022-06-05	Informe mensual	archivo	Admin Perez Lara
7	2022-07-05	Informe mensual	archivo	Admin Perez Lara

En esta vista nos trae los campos relacionados con llaves foráneas
reporte.usuario = u.documento Es la llave foránea que une las tablas usuario y
reporte.



Rutinas

```
use Perez_Lara_Cia_LTDA;
```

```
/* Rutina: sp_insert_usuario  
===== */
```

```
drop procedure if exists `sp_insert_usuario`;  
create procedure `sp_insert_usuario`(in `_documento` varchar(20), in `_primer_nombre`  
varchar(45),  
in `_segundo_nombre` varchar(45), in `_primer_apellido`  
varchar(45),  
in `_segundo_apellido` varchar(45), in `_celular` varchar(15),  
in `_email` varchar(76), in `_clave` varchar(256), in `_foto_perfil`  
varchar(45),  
in `_tipo_documento` int(3),  
in `_tipo_usuario` int(3), in `_estado_usuario` int(3))  
insert into usuario (documento, primer_nombre, segundo_nombre, primer_apellido,  
segundo_apellido, celular, email, clave,  
foto_perfil, tipo_documento, tipo_usuario, estado)  
values (_documento, _primer_nombre, _segundo_nombre, _primer_apellido,  
_segundo_apellido, _celular, _email,  
AES_ENCRYPT(_clave, 'colombia123'), _foto_perfil, _tipo_documento, _tipo_usuario,  
_estado_usuario);
```

Nombre	Tipo	Función	Valor
_documento	VARCHAR	<input type="text"/>	<input type="text"/>
_primer_nombre	VARCHAR	<input type="text"/>	<input type="text"/>
_segundo_nombre	VARCHAR	<input type="text"/>	<input type="text"/>
_primer_apellido	VARCHAR	<input type="text"/>	<input type="text"/>
_segundo_apellido	VARCHAR	<input type="text"/>	<input type="text"/>
_celular	VARCHAR	<input type="text"/>	<input type="text"/>
_email	VARCHAR	<input type="text"/>	<input type="text"/>
_clave	VARCHAR	<input type="text"/>	<input type="text"/>
_foto_perfil	VARCHAR	<input type="text"/>	<input type="text"/>
_tipo_documento	INT	<input type="text"/>	<input type="text"/>
_tipo_usuario	INT	<input type="text"/>	<input type="text"/>
_estado_usuario	INT	<input type="text"/>	<input type="text"/>

Continuar

Cerrar

Procedimiento almacenado con 12 parámetros para insertar un nuevo Usuario



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



```
/* Rutina: sp_insert_cliente
===== */
drop procedure if exists `sp_insert_cliente`;
create procedure sp_insert_cliente(in `_documento` varchar(20), in
`_primer_nombre` varchar(45),
                                in `_segundo_nombre` varchar(45), in `_primer_apellido`
varchar(45),
                                in `_segundo_apellido` varchar(45), in `_fecha_nacimiento`
date,
                                in `_email` varchar(76), in `_telefono1` varchar(15), in
`_telefono2` varchar(15),
                                in `_direccion1` varchar(100), in `_direccion2` varchar(100),
                                in `_tipo_documento` int,
                                in `_ciudad` int, in `_estado_cliente` int, in `_fecha_ingreso`
date)
insert into cliente (documento, primer_nombre, segundo_nombre, primer_apellido,
segundo_apellido, fecha_nacimiento,
                    email, telefono1, telefono2, direccion1, direccion2, tipo_documento,
ciudad, estado, fecha_ingreso)
values (_documento, _primer_nombre, _segundo_nombre, _primer_apellido,
_segundo_apellido, _fecha_nacimiento, _email,
        _telefono1, _telefono2, _direccion1, _direccion2, _tipo_documento, _ciudad,
_estado_cliente, _fecha_ingreso);
```



Nombre	Tipo	Función	Valor
_documento	VARCHAR	<input type="text"/>	<input type="text"/>
_primer_nombre	VARCHAR	<input type="text"/>	<input type="text"/>
_segundo_nombre	VARCHAR	<input type="text"/>	<input type="text"/>
_primer_apellido	VARCHAR	<input type="text"/>	<input type="text"/>
_segundo_apellido	VARCHAR	<input type="text"/>	<input type="text"/>
_fecha_nacimiento	DATE	<input type="text"/>	<input type="text"/>
_email	VARCHAR	<input type="text"/>	<input type="text"/>
_telefono1	VARCHAR	<input type="text"/>	<input type="text"/>
_telefono2	VARCHAR	<input type="text"/>	<input type="text"/>
_direccion1	VARCHAR	<input type="text"/>	<input type="text"/>
_direccion2	VARCHAR	<input type="text"/>	<input type="text"/>
_tipo_documento	INT	<input type="text"/>	<input type="text"/>
_ciudad	INT	<input type="text"/>	<input type="text"/>
_estado_cliente	INT	<input type="text"/>	<input type="text"/>
_fecha_ingreso	DATE	<input type="text"/>	<input type="text"/>

[Continuar](#) [Cerrar](#)

Se crearon 15 parámetros que van a insertar en la tabla cliente que serán insertados desde variables de php.

```
/* Rutina: sp_update_cliente
===== */
drop procedure if exists `sp_update_cliente`;
create procedure sp_update_cliente(in `_documento` varchar(20), in `_documento2`
varchar(20),
in `_primer_nombre` varchar(45),
in `_segundo_nombre` varchar(45), in `_primer_apellido` varchar(45),
in `_segundo_apellido` varchar(45), in `_fecha_nacimiento` date,
in `_email` varchar(76), in `_telefono1` varchar(15), in `_telefono2`
varchar(15),
in `_direccion1` varchar(100), in `_direccion2` varchar(100),
in `_tipo_documento` int,
in `_ciudad` int)
update cliente
set documento = `_documento2`,
```



```
primer_nombre = _primer_nombre,  
segundo_nombre = _segundo_nombre,  
primer_apellido = _primer_apellido,  
segundo_apellido = _segundo_apellido,  
fecha_nacimiento = _fecha_nacimiento,  
email = _email,  
telefono1 = _telefono1,  
telefono2 = _telefono2,  
direccion1 = _direccion1,  
direccion2 = _direccion2,  
tipo_documento = _tipo_documento,  
ciudad = _ciudad  
where documento = _documento;
```

Ejecutar rutina `sp_update_cliente`

Parámetros de rutina

Nombre	Tipo	Función	Valor
_documento	VARCHAR	<input type="text"/>	<input type="text"/>
_documento2	VARCHAR	<input type="text"/>	<input type="text"/>
_primer_nombre	VARCHAR	<input type="text"/>	<input type="text"/>
_segundo_nombre	VARCHAR	<input type="text"/>	<input type="text"/>
_primer_apellido	VARCHAR	<input type="text"/>	<input type="text"/>
_segundo_apellido	VARCHAR	<input type="text"/>	<input type="text"/>
_fecha_nacimiento	DATE	<input type="text"/>	<input type="text"/>
_email	VARCHAR	<input type="text"/>	<input type="text"/>
_telefono1	VARCHAR	<input type="text"/>	<input type="text"/>
_telefono2	VARCHAR	<input type="text"/>	<input type="text"/>
_direccion1	VARCHAR	<input type="text"/>	<input type="text"/>
_direccion2	VARCHAR	<input type="text"/>	<input type="text"/>
_tipo_documento	INT	<input type="text"/>	<input type="text"/>
_ciudad	INT	<input type="text"/>	<input type="text"/>
_estado_cliente	INT	<input type="text"/>	<input type="text"/>

Continuar

Cerrar

Se crearon 15 parámetros que van a actualizar datos en la tabla cliente y serán insertados desde variables de php.



```
/* Rutina: sp_delete_cliente
===== */
drop procedure if exists `sp_delete_cliente`;
create procedure `sp_delete_cliente`(in _documento varchar(20))
update cliente
set estado = 2
where documento = _documento;
```

Ejecutar rutina `sp_delete_cliente`

Parámetros de rutina

Nombre	Tipo	Función	Valor
_documento	VARCHAR		

Continuar Cerrar

rutina de un parametro que cambia el estado al cliente a inactivo

```
/* Rutina: sp_insert_cotizacion
===== */
drop procedure if exists `sp_insert_cotizacion`;
create procedure sp_insert_cotizacion(in `_documento_cliente` varchar(11), in `_fecha` date,
in `_seguro` int,
in `_cuadro_comparativo` varchar(150), in `_estado_cotizacion` int)
insert into cotizacion (documento_cliente, fecha, seguro, cuadro_comparativo, estado)
values (_documento_cliente, _fecha, _seguro, _cuadro_comparativo, _estado_cotizacion);
```

Ejecutar rutina `sp_insert_cotizacion`

Parámetros de rutina

Nombre	Tipo	Función	Valor
_documento_cliente	VARCHAR		
_fecha	DATE		
_seguro	INT		
_cuadro_comparativo	VARCHAR		
_estado_cotizacion	INT		

Continuar Cerrar

Se crearon 5 parámetros que van a insertar datos en la tabla cotizacion y serán incluidos desde variables de php.



```
/* Rutina: sp_update_cotizacion
===== */
drop procedure if exists `sp_update_cotizacion`;
create procedure sp_update_cotizacion(in `_id_cotizacion` int, in `_documento_cliente`
varchar(11), in `_fecha` date,
                                     in `_seguro` int, in `_cuadro_comparativo` varchar(150),
                                     in `_estado_cotizacion` int, in `_fecha_actualizacion` date)
update cotizacion
SET documento_cliente = _documento_cliente,
    fecha              = _fecha,
    seguro             = _seguro,
    cuadro_comparativo = _cuadro_comparativo,
    estado             = _estado_cotizacion,
    fecha_actualizacion = _fecha_actualizacion
where id_cotizacion = _id_cotizacion;
```

Ejecutar rutina `sp_update_cotizacion`

Parámetros de rutina			
Nombre	Tipo	Función	Valor
<code>_id_cotizacion</code>	INT	<input type="text"/>	<input type="text"/>
<code>_documento_cliente</code>	VARCHAR	<input type="text"/>	<input type="text"/>
<code>_fecha</code>	DATE	<input type="text"/>	<input type="text"/>
<code>_seguro</code>	INT	<input type="text"/>	<input type="text"/>
<code>_cuadro_comparativo</code>	VARCHAR	<input type="text"/>	<input type="text"/>
<code>_estado_cotizacion</code>	INT	<input type="text"/>	<input type="text"/>

Continuar

Cerrar

Se crearon 6 parámetros que van a actualizar datos en la tabla cotizacion y serán insertados desde variables de php.

```
/* Rutina: sp_insert_poliza
===== */
drop procedure if exists `sp_insert_poliza`;
create procedure sp_insert_poliza(IN _codigo int(45), IN _documento_cliente varchar(11),
IN _fecha date, IN _seguro int, IN _aseguradora int,
```




```
IN _valor_asegurado double, IN _valor_prima double,  
IN _archivo varchar(150), IN _estado_poliza int)  
insert into poliza (codigo, documento_cliente, fecha, seguro,  
aseguradora, valor_asegurado, valor_prima, archivo, estado)  
values (_codigo, _documento_cliente, _fecha, _seguro,  
_aseguradora, _valor_asegurado, _valor_prima, _archivo, _estado_poliza);
```

Ejecutar rutina `sp_insert_poliza`

Parámetros de rutina

Nombre	Tipo	Función	Valor
_codigo	INT	<input type="text"/>	<input type="text"/>
_documento_cliente	VARCHAR	<input type="text"/>	<input type="text"/>
_fecha	DATE	<input type="text"/>	<input type="text"/>
_seguro	INT	<input type="text"/>	<input type="text"/>
_aseguradora	INT	<input type="text"/>	<input type="text"/>
_archivo	VARCHAR	<input type="text"/>	<input type="text"/>
_estado_poliza	INT	<input type="text"/>	<input type="text"/>

Continuar Cerrar

Se crearon 7 parámetros que van a insertar datos en la tabla poliza y serán incluidos desde variables de php.

```
/* Rutina: sp_update_poliza  
===== */  
drop procedure if exists `sp_update_poliza`;  
create procedure sp_update_poliza(IN _codigo int(45), IN _codigo2 int(45), IN  
_documento_cliente varchar(11),  
IN _seguro int, IN _aseguradora int,  
IN _valor_asegurado double, IN _valor_prima double,  
IN _archivo varchar(150), IN _estado_poliza int,  
IN _fecha_actualizacion date)  
update poliza  
SET codigo = _codigo2,  
documento_cliente = _documento_cliente,  
seguro = _seguro,  
aseguradora = _aseguradora,
```



```
valor_asegurado = _valor_asegurado,  
valor_prima     = _valor_prima,  
archivo         = _archivo,  
estado          = _estado_poliza,  
fecha_actualizacion = _fecha_actualizacion  
where codigo = _codigo;
```

Ejecutar rutina `sp_update_poliza`

Parámetros de rutina

Nombre	Tipo	Función	Valor
_codigo	INT	<input type="text"/>	<input type="text"/>
_documento_cliente	VARCHAR	<input type="text"/>	<input type="text"/>
_fecha	DATE	<input type="text"/>	<input type="text"/>
_seguro	INT	<input type="text"/>	<input type="text"/>
_aseguradora	INT	<input type="text"/>	<input type="text"/>
_archivo	VARCHAR	<input type="text"/>	<input type="text"/>
_estado_poliza	INT	<input type="text"/>	<input type="text"/>

Se crearon 7 parámetros que van a actualizar datos en la tabla poliza y serán insertados desde variables de php.

```
/* Rutina: sp_insert_siniestro  
===== */  
drop procedure if exists `sp_insert_siniestro`;  
create procedure sp_insert_siniestro(in `_fecha` date, in `_descripcion` varchar(2000), in  
`_monto` double,  
in `_codigo_poliza` int, in `_anexo` varchar(150), in  
`_estado_siniestro` int)  
insert into siniestro (fecha, descripcion, monto, codigo_poliza, anexo, estado)  
values (_fecha, _descripcion, _monto, _codigo_poliza, _anexo, _estado_siniestro);
```



Ejecutar rutina `sp_insert_siniestro`

Parámetros de rutina

Nombre	Tipo	Función	Valor
_fecha	DATE	<input type="text"/>	<input type="text"/>
_descripcion	VARCHAR	<input type="text"/>	<input type="text"/>
_monto	DOUBLE	<input type="text"/>	<input type="text"/>
_codigo_poliza	INT	<input type="text"/>	<input type="text"/>
_anexo	VARCHAR	<input type="text"/>	<input type="text"/>
_estado_siniestro	INT	<input type="text"/>	<input type="text"/>

Continuar

Cerrar

Se crearon 6 parámetros que van a insertar datos en la tabla siniestro y serán incluidos desde variables de php.

```
/* Rutina: sp_update_siniestro
===== */
drop procedure if exists `sp_update_siniestro`;
create procedure sp_update_siniestro(in `_id_siniestro` int, in `_fecha` date, in
`_descripcion` varchar(2000),
in `_monto` double, in `_codigo_poliza` int, in `_anexo` varchar(150),
in `_estado_siniestro` int)
update siniestro
SET fecha      = _fecha,
    descripcion = _descripcion,
    monto      = _monto,
    codigo_poliza = _codigo_poliza,
    anexo      = _anexo,
    estado     = _estado_siniestro
where id_siniestro = _id_siniestro;
```



Servicio Nacional de Aprendizaje SENA

Centro Electricidad, Electrónica y
Telecomunicaciones



Ejecutar rutina `sp_update_siniestro`

Parámetros de rutina

Nombre	Tipo	Función	Valor
_id_siniestro	INT	<input type="text"/>	<input type="text"/>
_fecha	DATE	<input type="text"/>	<input type="text"/>
_descripcion	VARCHAR	<input type="text"/>	<input type="text"/>
_monto	DOUBLE	<input type="text"/>	<input type="text"/>
_codigo_poliza	INT	<input type="text"/>	<input type="text"/>
_anexo	VARCHAR	<input type="text"/>	<input type="text"/>
_estado_siniestro	INT	<input type="text"/>	<input type="text"/>

Continuar

Cerrar

Se crearon 7 parámetros que van a actualizar datos en la tabla siniestro y serán insertados desde variables de php.

CONTROL DE DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor(es))	Camila Andrea Torres	Backend	Teleinformática	28/07/2022
	David Santiago Caro	Frontend	Teleinformática	28/07/2022
	David Andrey Sánchez	Tester	Teleinformática	29/07/2022
	Juan Sebastian Alarcon Diaz	Administrador del proyecto	Teleinformática	4/08/2022
	Daniel Fernando Urrea Daza	Analista	Teleinformática	5/08/2022
	Oscar Giovany Lopez Leal	Documentador	Teleinformática	5/08/2022



Servicio Nacional de Aprendizaje SENA

**Centro Electricidad, Electrónica y
Telecomunicaciones**



CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía).

	Nombre	Cargo	Dependencia	Fecha
Autor(es)				