



**ceutec**  
de unitec  
LAUREATE INTERNATIONAL UNIVERSITIES

*Universidad Tecnológica Centroamericana*

lunes 17 de diciembre de 2018  
Diagrama Corregido



**Mediware®**

## **INFORME DE PROYECTO**

Teoría de bases de datos

MAESTRO: BIANCA ROSALES  
ALUMNO: DENNIS MOTIÑO  
SECCION: 1217  
CUENTA: 31711467

# Introducción

---

En la Actualidad, la información es la base principal de cualquier sistema, y el correcto almacenamiento y procesamiento de esta es de vital importancia para lograr sus objetivos.

Esto implica la forma en que se almacena y el medio en donde lo hacemos. Lo más común es que usemos una base de datos digital sea esta relacional o no, es muy importante saber crear el modelo de cómo se almacenaran los datos, pues de esto depende que cualquier acción que se haga la información sea de manera óptima, ahorrando costos y tiempo.

En el presente proyecto se busca crear un modelo óptimo para el almacenamiento, administración y control de la información proporcionada por pacientes de un centro de seguros médicos. Este software que se llamara Mediware será el encargado de atender los reclamos médicos de los pacientes de la manera más ágil y efectiva posible.

Se podrá almacenar registros de pacientes, llevar un historial médico y además generar reportes de información relevante a petición del usuario.

El tiempo de espera ,costos serán reducidos aumentado la calidad al cliente y por ende las ganancias de la institución.

## **Descripción del Problema a Resolver**

---

Una compañía de Seguros, desea tener un sistema automatizado de los reclamos médicos de los asegurados.

De los asegurados se desea guardar, número de identificación, el nombre completo del asegurado, ciudad de nacimiento, departamento de nacimiento, fecha de nacimiento, edad, nacionalidad, género, peso en libras, estaturas en centímetros, estado civil, nombre del conyugue, dirección de residencia, ciudad de residencia, departamento de residencia, teléfonos, correos electrónicos, nombre de empresa donde labora, cargo que desempeña, antigüedad laboral, sitio web de la empresa, posee póliza de seguros con otra aseguradora, nombre de la compañía, tiempo de poseerla, padecimientos del asegurado.

Cada asegurado posee un código interno. Los asegurados pueden incluir a sus familiares dentro de la póliza de seguros, los cuales serían dependientes.

Un asegurado puede tener muchos dependientes, pero este dependiente solo puede estar asegurado a través de un solo asegurado.

De los dependientes se requiere, nombre, parentesco, ocupación, género, estatura en cm, peso en libras, fecha de nacimientos y padecimientos del dependiente.

Un asegurado contrata una póliza de seguros, esta póliza pertenece a un solo asegurado. De cada póliza nos interesa, el número de póliza, número de certificado, tipo de póliza, fecha de inicio de cobertura, fecha final de cobertura, valor máximo de habitación, deducible, porcentaje de coaseguro, incluye atención de maternidad, porcentaje de coaseguro en farmacias, porcentaje de coaseguro en laboratorio y rayos X, red de médicos, otros servicios,

monto del seguro de vida, monto de servicios fúnebres. En una póliza un asegurado puede incluir varios dependientes, pero estos solo pueden ser cubiertos por una sola póliza

Los asegurados presentan sus reclamos médicos, de los cuales se guarda, la fecha de presentación, a quién pertenece el reclamo (dependiente), monto total del reclamo. Cada reclamo se identifica a través de un Número de reclamo.

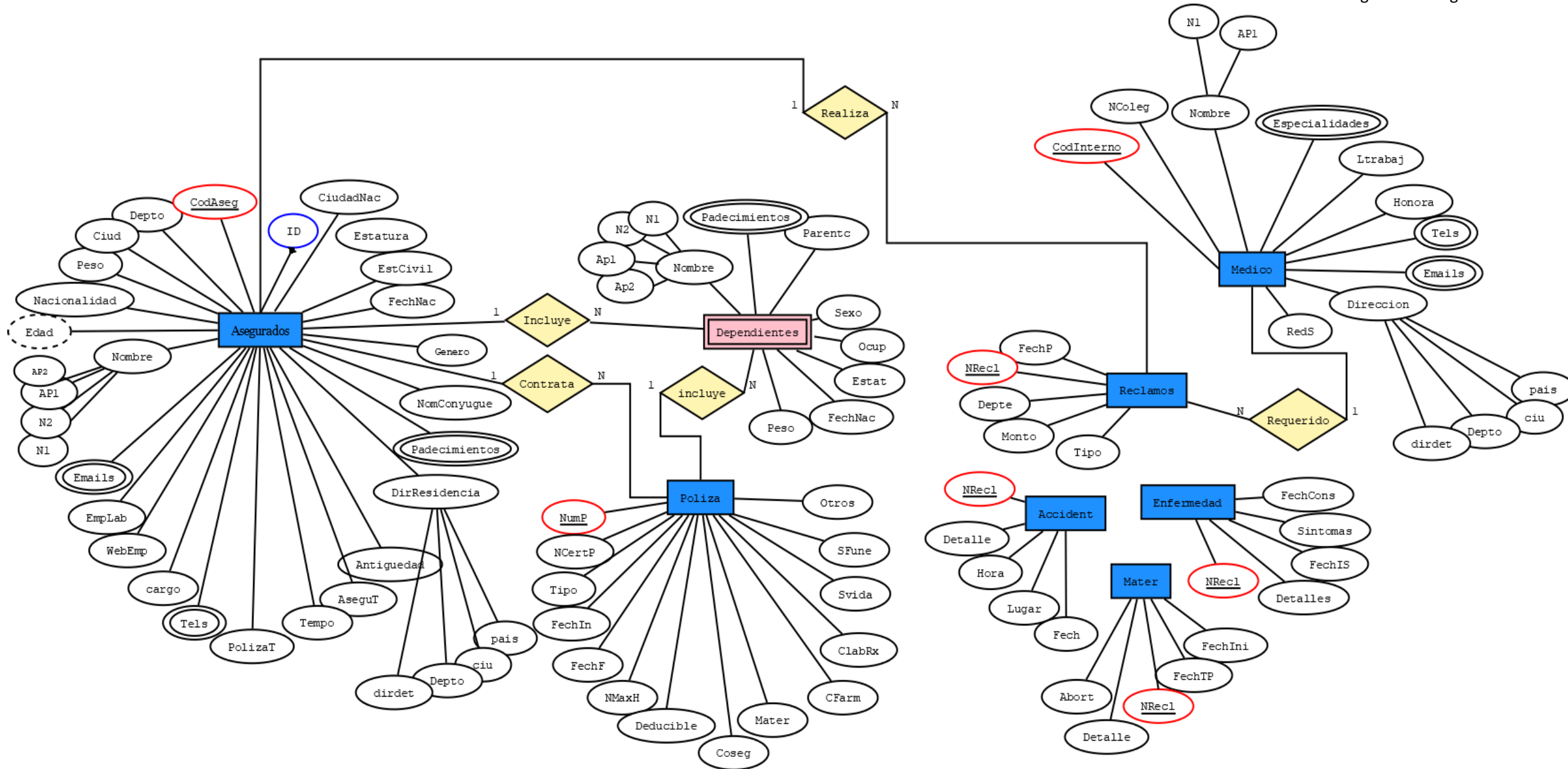
Los tipos de reclamos pueden ser: Por Accidentes, de los cuales nos interesa saber, fecha en que ocurrió, hora en que ocurrió, lugar del suceso y los detalles del mismo. Por Maternidad, de los cuales nos interesa Fecha de inicio de embarazo, fecha tentativa de parto, cantidad de abortos previos, detalles. De las enfermedades, nos interesa, los síntomas, fecha de inicio del síntoma, fecha de consulta, detalles. Recuerde que un asegurado puede realizar varios reclamos médicos y pero un reclamo médico solo pertenece a un solo asegurado. Cada reclamo se realiza para un dependiente y/o asegurado.

En cada reclamo médico se hace para un médico, pero un médico puede ser requerido en muchos reclamos médicos, de cada doctor no interesa, código de colegiación, nombre, especialidad o especialidades, nombre lugar de trabajo, costo por honorarios, teléfonos, correos electrónicos, dirección completa, pertenece a la red del seguro. Cada médico se identifica a través de un código interno.



lunes 17 de diciembre de 2018

Diagrama Corregido





## Script de base de datos y registros

---

```
use master
go
```

### --\*\*\*Creación de La base de datos pura\*\*\*

```
-----
create database BDMediware
on primary
(
name='BDMediware',
filename='C:\Data\Mediware\BDMediware.mdf',
size=200MB,
maxsize=5000MB,
filegrowth=25%)
log on
(
name='MediwareLogs',
filename='C:\Data\Mediware\Logs\mdLogs.ldf',
size=100MB,
maxsize=1000MB,
filegrowth=25%
)
go
```

### --\*\*\*\*\*TABLAS\*\*\*\*\*

```
USE BDMediware
```

#### --\*\*\*\*\*TABLA PARA LOGIN\*\*\*\*\*

```
-----
create table dbo.LogUsuarios (
log_Id varchar(20) not null,
log_Clave varchar(10) not null,
log_NombreCompleto varchar(35) not null,
log_IDempleado int not null
primary key(log_Id)
)
go
```

#### --\*\*\*\*\*TABLA CIUDADES\*\*\*\*\*

```
create table dbo.Ciudades(
ciu_Id int identity(1,1),
Ciu_nombre varchar(15),
primary key(ciu_Id),
unique(Ciu_nombre)
)
```

#### --\*\*\*\*\*TABLA DEPARTAMENTOS\*\*\*\*\*

```
create table dbo.Departamento(
dep_Id int identity(1,1),
dep_nombre varchar(15),
primary key(dep_Id),
unique(dep_nombre))
```

#### --\*\*\*\*\*TABLA PAISES\*\*\*\*\*

```
create table dbo.Nacionalidad(
na_Id int identity(1,1),
na_nombre varchar(15),
primary key(na_Id),
unique(na_nombre))
```

)

--\*\*\*\*\*TABLA PADECIMIENTOS\*\*\*\*\*

```
create table dbo.Padecimientos(  
pad_Id int identity(1,1),  
pad_nombre varchar(25),  
primary key(pad_Id),  
unique(pad_nombre)  
)
```

--\*\*\*\*\*TABLA ESPECIALIDADES\*\*\*\*\*

```
create table dbo.Especialidades(  
esp_Id int identity(1,1),  
esp_nombre varchar(25) not null,  
primary key(esp_Id),  
unique(esp_nombre)  
)
```

--\*\*\*\*\*TABLA BITACORA\*\*\*\*\*

```
create table dbo.Bitacora(  
bit_Usuario varchar(60) not null,  
bit_PCname varchar(30),  
bit_Evento varchar(50),  
bit_fecha date,  
bit_Hora time  
)
```

--\*\*\*\*\*TABLA ASEGURADOS\*\*\*\*\*

```
create table dbo.AseguradosTBL(  
Aseg_Codigo int identity(1,1) primary key,  
Aseg_Id varchar (20) unique not null,  
Aseg_Nombre varchar(25) not null,  
Aseg_Apellido varchar(25) not null,  
Aseg_Nacionalidad int not null,  
Aseg_DeptoNac int not null,  
Aseg_CiudadNac int not null,  
Aseg_FechaNac date not null,  
Aseg_Estatura tinyint not null,  
Aseg_Peso int not null,  
Aseg_Genero char(1) not null,  
Aseg_EstadoCivil char (1) not null,  
Aseg_nomConyugue varchar(40),  
Aseg_DeptoRes int not null,  
Aseg_CiudadRes int not null,  
Aseg_direccionRes varchar(50)not null,  
Aseg_NomEmpresa varchar(25),  
Aseg_EmpresaWeb varchar (30),  
Aseg_AntiguedadLab tinyint ,  
Aseg_puestoTrab varchar(30),  
Aseg_nomAseguradoraT varchar(20),  
Aseg_tiempoT tinyint,  
foreign key (Aseg_Nacionalidad) references dbo.Nacionalidad(na_Id),  
foreign key (Aseg_DeptoNac) references dbo.Departamento(dep_Id),  
foreign key (Aseg_CiudadNac) references dbo.Ciudades(ciu_Id),  
foreign key (Aseg_DeptoRes) references dbo.Departamento(dep_Id),  
foreign key (Aseg_CiudadRes) references dbo.Ciudades(ciu_Id)  
)  
go
```

--\*\*\*\*\*TABLA MEDICOS\*\*\*\*\*

```
create table dbo.MedicoTBL(  
med_IntCod int identity(0,1),  
med_Colegiacion varchar(30) not null,  
med_Nombres varchar(25) not null,  
med_Apellidos varchar (25) not null,  
med_LugTrabajo varchar (40) not null,  
med_Honorarios decimal (9,2) not null,  
med_pais int not null,  
med_ciudad int not null,  
med_Depto int not null,  
med_Detalledir varchar (50)not null,  
med_redSeg char(1) not null,  
primary key (med_IntCod),  
unique(med_Colegiacion),  
foreign key (med_pais) references dbo.Nacionalidad(na_Id),  
foreign key (med_Depto) references dbo.Departamento(dep_Id),  
foreign key (med_ciudad) references dbo.Ciudades(ciu_Id)  
)  
go
```

--\*\*\*\*\*TABLA POLIZAS\*\*\*\*\*

```
create table dbo.PolizaTBL(  
polz_Id int identity(1,1),  
polz_AsegCodigo int not null,  
polz_NumCert int not null,  
polz_Tipo char(2) not null,  
polz_FechIni date not null,  
polz_FechFin date not null,  
polz_MaxHab decimal(9,2) not null,  
polz_Deducible decimal (9,2) not null,  
polz_Coseguro tinyint not null,  
polz_Maternidad char(1) not null,  
polz_Farmacia tinyint not null,  
polz_Lab_Rayosx tinyint not null,  
polz_Otros tinyint not null,  
polz_segVida decimal (9,2) not null,  
polz_serFunebres decimal (9,2)not null,  
primary key(polz_Id),  
unique(polz_NumCert),  
check(polz_FechIni<polz_FechFin),  
foreign key (polz_AsegCodigo) references dbo.AseguradosTBL(Aseg_Codigo)  
)  
go
```

--\*\*\*\*\*TABLAS DEPENDIENTES\*\*\*\*\*

```
create table dbo.DependientesTBL(  
dep_CodigoD int identity(1,1) primary key,  
Aseg_CodigoD int not null,  
polz_IdD int not null,  
dep_Nombre varchar(30) not null,  
dep_Apellido varchar(25) not null,  
dep_FechaNac date not null,  
dep_Estatura int not null,  
dep_Peso int not null,  
dep_Parentesco varchar(15) not null,  
dep_genero char(1) not null,  
foreign key (Aseg_CodigoD) references dbo.AseguradosTBL(Aseg_Codigo),  
foreign key (polz_IdD) references dbo.PolizaTBL(polz_Id)  
)  
go
```



--\*\*\*\*\*TABLA RECLAMOS POR ACCIDENTE\*\*\*\*\*

```
create table dbo.RecAccidenteTBL(  
Acc_Id int identity(1,1),  
Acc_Aseg_Codigo int not null,  
Acc_med_IntCod int not null,  
Acc_Fecpresentacion date not null,  
Acc_Monto decimal(9,2) not null,  
Acc_fecha date not null,  
Acc_horaSuc time not null,  
Acc_ciudad int not null,  
Acc_direccion varchar(50) not null,  
Acc_Detalle varchar(200) not null,  
check(Acc_Fecpresentacion>=Acc_fecha),  
primary key(Acc_Id),  
foreign key (Acc_Aseg_Codigo) references dbo.AseguradosTBL(Aseg_Codigo),  
foreign key (Acc_med_IntCod) references dbo.MedicoTBL(med_IntCod),  
foreign key(Acc_ciudad) references dbo.Ciudades(ciu_Id)  
)
```

--\*\*\*\*\*TABLA RECLAMOS POR MATERIDAD\*\*\*\*\*

```
create table dbo.RecMaternidadTBL(  
mat_Id int identity(1,1),  
mat_Aseg_Codigo int not null,  
mat_med_IntCod int not null,  
mat_Fecpresentacion date not null,  
mat_Monto decimal(9,2) not null,  
mat_fechaIn date not null,  
mat_fechaParto date not null,  
mat_prevAbort tinyint not null,  
mat_detalles varchar(200) not null,  
check(mat_fechaIn<=mat_Fecpresentacion and mat_fechaIn<mat_fechaParto),  
primary key(mat_Id),  
foreign key (mat_Aseg_Codigo) references dbo.AseguradosTBL(Aseg_Codigo),  
foreign key (mat_med_IntCod) references dbo.MedicoTBL(med_IntCod)  
)
```

--\*\*\*\*\*TABLA RECLAMOS POR ENFERMEDAD\*\*\*\*\*

```
create table dbo.recEnfermedadTBL(  
enf_Id int identity(1,1),  
enf_Aseg_Codigo int not null,  
enf_med_IntCod int not null,  
enf_Fecpresentacion date not null,  
enf_Monto decimal(9,2) not null,  
enf_fechConsulta date not null,  
enf_fechIniSintomas date not null,  
enf_sintomas varchar(200) not null,  
enf_detalles varchar(200),  
check(enf_Fecpresentacion>=enf_fechIniSintomas and  
enf_Fecpresentacion>=enf_fechConsulta),  
primary key(enf_Id),  
foreign key (enf_Aseg_Codigo) references dbo.AseguradosTBL(Aseg_Codigo),  
foreign key (enf_med_IntCod) references dbo.MedicoTBL(med_IntCod)  
)
```

--\*\*\*TABLA PADECIMIENTOS PARA SEGUROS Y DEPENDIENTES\*\*\*\*\*

```
create table dbo.PadecimientosTBL(  
Pad_Aseg_Cod int not null ,  
Padecimiento int not null,  
primary key(Pad_Aseg_Cod,Padecimiento),  
foreign key (Pad_Aseg_Cod) references dbo.AseguradosTBL,  
foreign key (Padecimiento) references dbo.Padecimientos(pad_Id)  
)
```

--\*\*\*\*\*TABLA PARA TELEFONOS DE ASEGURADOS\*\*\*\*\*

```
create table dbo.TelefonosAseg(  
Aseg_Codigo int,  
Telefono varchar (10),  
primary key(Aseg_Codigo,Telefono),  
foreign key (Aseg_Codigo) references dbo.AseguradosTBL(Aseg_Codigo)  
)  
go
```

--\*\*\*\*\*TABLA PARA CORREOS DE ASEGURADOS\*\*\*\*\*

```
create table dbo.EmailsAseg(  
e_Aseg_Codigo int,  
e_mail varchar(30),  
primary key(e_Aseg_Codigo,e_mail),  
foreign key (e_Aseg_Codigo) references dbo.AseguradosTBL(Aseg_Codigo)  
)  
go
```

--\*\*\*\*\*TABLA PARA TELEFONOS DE MEDICOS\*\*\*\*\*

```
create table dbo.TelefonosMed(  
Med_Codigo int,  
Telefono varchar (10),  
primary key(Med_Codigo,Telefono),  
foreign key (Med_Codigo) references dbo.MedicoTBL(med_IntCod)  
)  
go
```

--\*\*\*\*\*TABLA PARA CORREOS DE MEDICOS\*\*\*\*\*

```
create table dbo.EmailsMed(  
e_Aseg_Codigo int,  
e_mail varchar(30),  
primary key(e_Aseg_Codigo,e_mail),  
foreign key (e_Aseg_Codigo) references dbo.MedicoTBL(med_IntCod)  
)  
go
```

--\*\*\*\*\*TABLA ESPECIALIDADES DE MEDICOS\*\*\*\*\*

```
create table dbo.Especialidadesmed(  
esp_MedCodigo int,  
Especialidad int,  
primary key(esp_MedCodigo,Especialidad),  
foreign key (esp_MedCodigo) references dbo.MedicoTBL(med_IntCod),  
foreign key (Especialidad) references dbo.Especialidades(esp_Id)  
)  
go
```

--#####STORED PROCEDURES #####

--#####AUTENTICACION LOGIN#####

```
create procedure sp_LoginUsuarios  
@Clave varchar(10),  
@Usuario varchar(20) as  
select log_NombreCompleto from BDMediware.dbo.LogUsuarios where  
dbo.LogUsuarios.log_Clave=@Clave and dbo.LogUsuarios.log_Id=@Usuario  
go
```

--#####SP VER ASEGURADOS TABLA COMPLETA#####

```
create procedure sp_verAsegurados as
```

```

select
dbo.AseguradosTBL.Aseg_Codigo as Id,
dbo.AseguradosTBL.Aseg_Id as Dni,
dbo.AseguradosTBL.Aseg_Nombre as Nombre,
dbo.AseguradosTBL.Aseg_Apellido as Apellido,
dbo.Nacionalidad.na_nombre as Pais,
DN.dep_nombre as DeptoNac,
CN.Ciu_nombre as CiuNatal,
dbo.AseguradosTBL.Aseg_FechaNac as Nacimiento,
dbo.AseguradosTBL.Aseg_Estatura as Estatura,
dbo.AseguradosTBL.Aseg_Peso as Peso,
dbo.AseguradosTBL.Aseg_Genero as Genero,
dbo.AseguradosTBL.Aseg_EstadoCivil as EstadoCivil,
dbo.AseguradosTBL.Aseg_nomConyugue as NombreConyugue,
DR.dep_nombre as DeptoResidencia,
CR.Ciu_nombre as ciuResidencia,
dbo.AseguradosTBL.Aseg_direccionRes as Direccion,
dbo.AseguradosTBL.Aseg_NomEmpresa as Laburo,
dbo.AseguradosTBL.Aseg_EmpresaWeb as WebLaburo,
dbo.AseguradosTBL.Aseg_AntiguedadLab as Antiguedad,
dbo.AseguradosTBL.Aseg_puestoTrab as Puesto,
dbo.AseguradosTBL.Aseg_nomAseguradoraT as OtraAseguradora,
dbo.AseguradosTBL.Aseg_tiempoT as Tiempo from dbo.AseguradosTBL inner join
dbo.Ciudades as CN
on dbo.AseguradosTBL.Aseg_CiudadNac=CN.ciu_Id inner join dbo.Ciudades as CR
on dbo.AseguradosTBL.Aseg_CiudadRes=CR.ciu_Id inner join dbo.Departamento as DN
on dbo.AseguradosTBL.Aseg_DeptoNac=DN.dep_Id inner join dbo.Departamento as DR
on dbo.AseguradosTBL.Aseg_DeptoRes=DR.dep_Id inner join dbo.Nacionalidad
on dbo.AseguradosTBL.Aseg_Nacionalidad=dbo.Nacionalidad.na_Id order by Nombre

```

#### --##### SP BUSCAR DEPENDIENTES

```

create procedure sp_BuscaDependiente
@Nombre varchar(25)
as
select
Aseg_CodigoD as Codigo,
polz_IdD as Poliza,
dep_Nombre as nombre,
dep_Apellido as Apellido,
dep_FechaNac as Fecha_Nacimiento,
dep_Estatura as Estatura,
dep_Peso as Peso,
dep_Parentesco as Parentesco
from dbo.DependientesTBL
where Aseg_CodigoD=(select Aseg_Codigo from dbo.AseguradosTBL where
Aseg_Nombre=@Nombre)
go

```

#### #####SP VER RECLAMOS POR DEPENDIENTE#####

```

create procedure sp_verReclamos
@AsegNom varchar(25) as
select ASE.Aseg_Nombre as Asegurado, ASE.Aseg_Apellido as
Apellidos,MED.med_Nombres As Medico_Nombre,MED.med_Apellidos as
Apellido,ACC.Acc_Fecpresentacion as Presentado,ACC.Acc_Monto as
Monto,ACC.Acc_Detalle as Descripcion from dbo.RecAccidenteTBL as ACC
inner join dbo.AseguradosTBL as ASE on
ACC.Acc_Aseg_Codigo=ASE.Aseg_Codigo
inner join dbo.MedicoTBL as MED on
ACC.Acc_med_IntCod=MED.med_IntCod where ASE.Aseg_Nombre=@AsegNom
union
select ASEG.Aseg_Nombre as Asegurado,ASEG.Aseg_Apellido as
Apellidos,MEDI.med_Nombres As Medico_Nombre,MEDI.med_Apellidos as

```

```

Apellido,MAT.mat_Fecpresentacion as Presentado,MAT.mat_Monto as
Monto,MAT.mat_detalles as Descripcion from dbo.RecMaternidadTBL as MAT
inner join dbo.AseguradosTBL as ASEG on
MAT.mat_Aseg_Codigo=ASEG.Aseg_Codigo
inner join dbo.MedicoTBL as MEDI on
MAT.mat_med_IntCod=MEDI.med_IntCod where ASEG.Aseg_Nombre=@AsegNom
union
select ASEGU.Aseg_Nombre as Asegurado,ASEGU.Aseg_Apellido as
Apellidos,MEDIC.med_Nombres as Medico_Nombre,MEDIC.med_Apellidos as
Apellido,ENF.enf_Fecpresentacion as Presentado,ENF.enf_Monto as
Monto,ENF.enf_detalles as Descripcion from dbo.recEnfermedadTBL as ENF
inner join dbo.AseguradosTBL as ASEGU on
ENF.enf_Aseg_Codigo=ASEGU.Aseg_Codigo
inner join dbo.MedicoTBL as MEDIC on
ENF.enf_med_IntCod=MEDIC.med_IntCod where ASEGU.Aseg_Nombre=@AsegNom

--exec sp_verReclamos 'Keylin Celena'
--select * from dbo.RecAccidenteTBL where Acc_Monto=(select max(Acc_Monto) from
dbo.RecAccidenteTBL)
--select max(Acc_Monto) from dbo.RecAccidenteTBL

--#####SP VER POLIZAS POR NUMERO
create procedure sp_Poliza
@numerop int as
select P.polz_Id as Numero,CONCAT(ASE.Aseg_Nombre,' ',ASE.Aseg_Apellido) as
Asegurado,P.polz_NumCert as Certificado,p.polz_Tipo as Tipo,
P.polz_FechIni as Inicio,P.polz_FechFin Finaliza,P.polz_MaxHab as [Maximo
Habitacion],
P.polz_Deducible as Deducible,P.polz_Maternidad [Incluye Maternidad],
p.polz_Coseguro as [% Coseguro],P.polz_Farmacia as [% Farmacia],
P.polz_Lab_Rayosx as [% Rayos X],P.polz_Otros as Otros,P.polz_segVida as [Seguro
de Vida],
P.polz_serFunebres as [Servicios Funebres] from dbo.PolizaTBL as P
inner join dbo.AseguradosTBL as ASE on
P.polz_AsegCodigo=ASE.Aseg_Codigo where P.polz_Id=@numerop

#####SP VER RECLAMOS POR FECHA
create procedure sp_RecXfecha
@fecha date as
begin
select * from v_reclamoMax where Presentado=@fecha
end
exec sp_RecXfecha '2018-5-1'

--#####SP INSERTAR DATOS EN TABALA ASEGURADOS ###
create procedure sp_insertarAsegurados
@Aseg_Id varchar (20) ,
@Aseg_Nombre varchar(25) ,
@Aseg_Apellido varchar(25),
@Aseg_Nacionalidad int ,
@Aseg_DeptoNac int ,
@Aseg_CiudadNac int,
@Aseg_FechaNac date ,
@Aseg_Estatura tinyint ,
@Aseg_Peso int ,
@Aseg_Genero char(1) ,
@Aseg_EstadoCivil char (1),
@Aseg_nomConyugue varchar(40),
@Aseg_DeptoRes int ,
@Aseg_CiudadRes int ,
@Aseg_direccionRes varchar(50),
@Aseg_NomEmpresa varchar(25),
@Aseg_EmpresaWeb varchar (30),

```

```

@Aseg_AntiguedadLab tinyint,
@Aseg_puestoTrab varchar(30),
@Aseg_nomAseguradoraT varchar(20),
@Aseg_tiempoT tinyint
as
insert into dbo.AseguradosTBL(
Aseg_Id,Aseg_Nombre,Aseg_Apellido,Aseg_Nacionalidad,Aseg_DeptoNac,Aseg_CiudadNac
,Aseg_FechaNac,
Aseg_Estatura,Aseg_Peso,Aseg_Genero,Aseg_EstadoCivil,Aseg_nomConyugue,Aseg_Depto
Res,Aseg_CiudadRes,
Aseg_direccionRes,Aseg_NomEmpresa
,Aseg_EmpresaWeb,Aseg_AntiguedadLab,Aseg_puestoTrab,Aseg_nomAseguradoraT,Aseg_ti
empoT)
values(
@Aseg_Id,@Aseg_Nombre,@Aseg_Apellido,@Aseg_Nacionalidad,@Aseg_DeptoNac,@Aseg_Ciu
dadNac,@Aseg_FechaNac,
@Aseg_Estatura,@Aseg_Peso,@Aseg_Genero,@Aseg_EstadoCivil,@Aseg_nomConyugue,@Aseg
_DeptoRes,@Aseg_CiudadRes,
@Aseg_direccionRes,@Aseg_NomEmpresa
,@Aseg_EmpresaWeb,@Aseg_AntiguedadLab,@Aseg_puestoTrab,@Aseg_nomAseguradoraT,@As
eg_tiempoT)

```

## --##### VISTAS #####

### ##### V. RECLAMOS

```

create view v_reclamoMax as
select ASE.Aseg_Nombre as Asegurado, ASE.Aseg_Apellido as
Apellidos,MED.med_Nombres As Medico_Nombre,MED.med_Apellidos as
Apellido,ACC.Acc_Fecpresentacion as Presentado,ACC.Acc_Monto as
Monto,ACC.Acc_Detalle as Descripcion from dbo.RecAccidenteTBL as ACC
inner join dbo.AseguradosTBL as ASE on
ACC.Acc_Aseg_Codigo=ASE.Aseg_Codigo
inner join dbo.MedicoTBL as MED on
ACC.Acc_med_IntCod=MED.med_IntCod
union
select ASEG.Aseg_Nombre as Asegurado,ASEG.Aseg_Apellido as
Apellidos,MEDI.med_Nombres As Medico_Nombre,MEDI.med_Apellidos as
Apellido,MAT.mat_Fecpresentacion as Presentado,MAT.mat_Monto as
Monto,MAT.mat_detalles as Descripcion from dbo.RecMaternidadTBL as MAT
inner join dbo.AseguradosTBL as ASEG on
MAT.mat_Aseg_Codigo=ASEG.Aseg_Codigo
inner join dbo.MedicoTBL as MEDI on
MAT.mat_med_IntCod=MEDI.med_IntCod
union
select ASEGU.Aseg_Nombre as Asegurado,ASEGU.Aseg_Apellido as
Apellidos,MEDIC.med_Nombres as Medico_Nombre,MEDIC.med_Apellidos as
Apellido,ENF.enf_Fecpresentacion as Presentado,ENF.enf_Monto as
Monto,ENF.enf_detalles as Descripcion from dbo.recEnfermedadTBL as ENF
inner join dbo.AseguradosTBL as ASEGU on
ENF.enf_Aseg_Codigo=ASEGU.Aseg_Codigo
inner join dbo.MedicoTBL as MEDIC on
ENF.enf_med_IntCod=MEDIC.med_IntCod

```

```
-##### V. RECLAMO CON MAYOR VALOR
create view MaximoValorReclamo as
select * from v_reclamoMax where Monto=(select max(Monto) from v_reclamoMax)
```

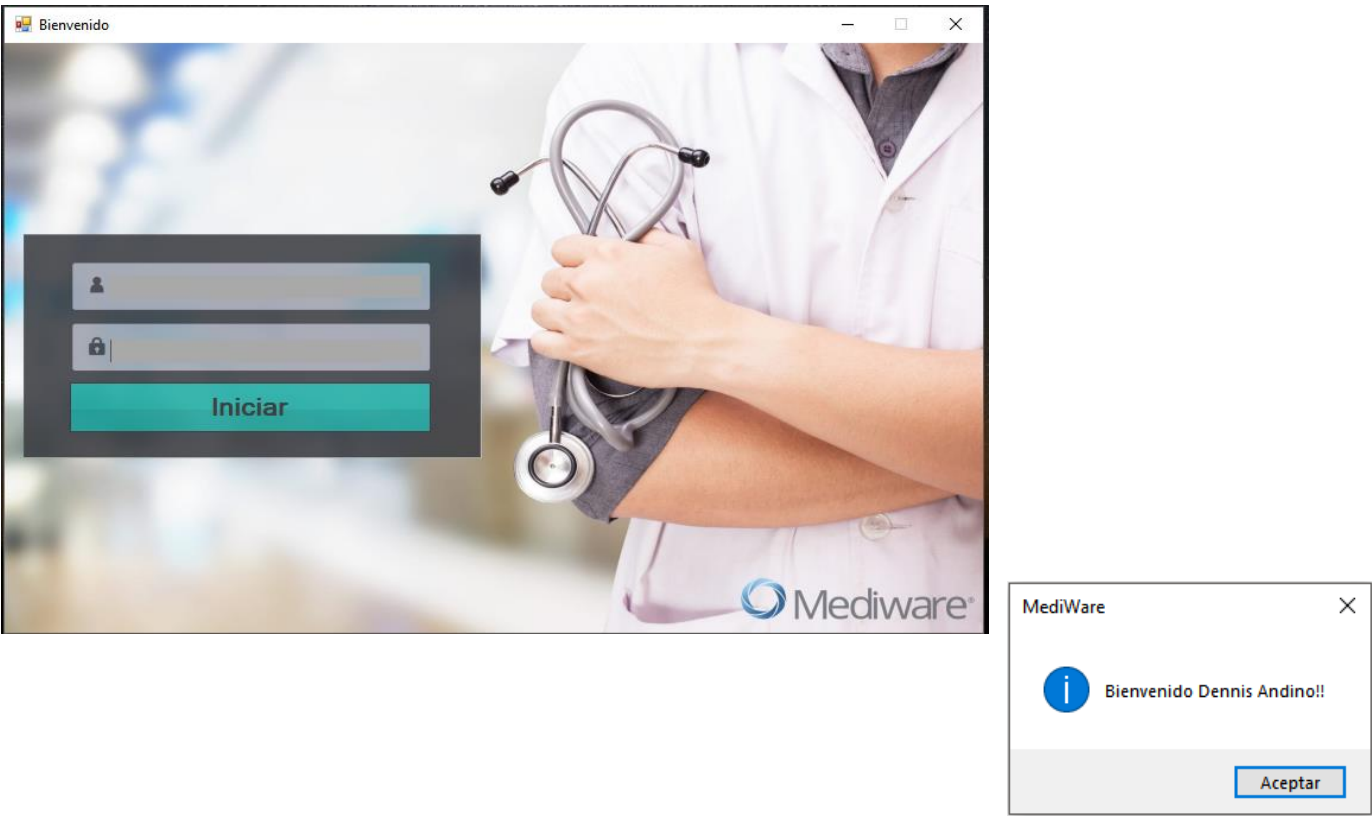
```
##### V. VER BITACORA
create view VerBtacara as
select bit_usuario as Usuario, bit_PCname as Computadora,bit_Evento as
Evento,bit_fecha as Fecha,bit_Hora as Hora from Bitacora
```

```
##### TRIGGER ASEGURADOS BITACORA

create trigger dbo.TG_Asegurados on  dbo.AseguradosTBL
after insert,update,delete
as
begin
if exists(select 1 from inserted) and not exists(select 1 from deleted)
insert into dbo.Bitacora(bit_usuario,bit_PCname,bit_Evento,bit_fecha,bit_Hora)
values(SYSTEM_USER,HOST_NAME(),'Insercion',CONVERT (date, SYSDATETIME()),CONVERT
(time, SYSDATETIME()))
else if exists(select 1 from inserted) and exists(select 1 from deleted)
insert into dbo.Bitacora(bit_usuario,bit_PCname,bit_Evento,bit_fecha,bit_Hora)
values(SYSTEM_USER,HOST_NAME(),'Actualizacion',CONVERT (date,
SYSDATETIME()),CONVERT (time, SYSDATETIME()))
else if not exists(select 1 from inserted) and exists (select 1 from deleted)
insert into dbo.Bitacora(bit_usuario,bit_PCname,bit_Evento,bit_fecha,bit_Hora)
values(SYSTEM_USER,HOST_NAME(),'Borrado',CONVERT (date, SYSDATETIME()),CONVERT
(time, SYSDATETIME()))
end
```

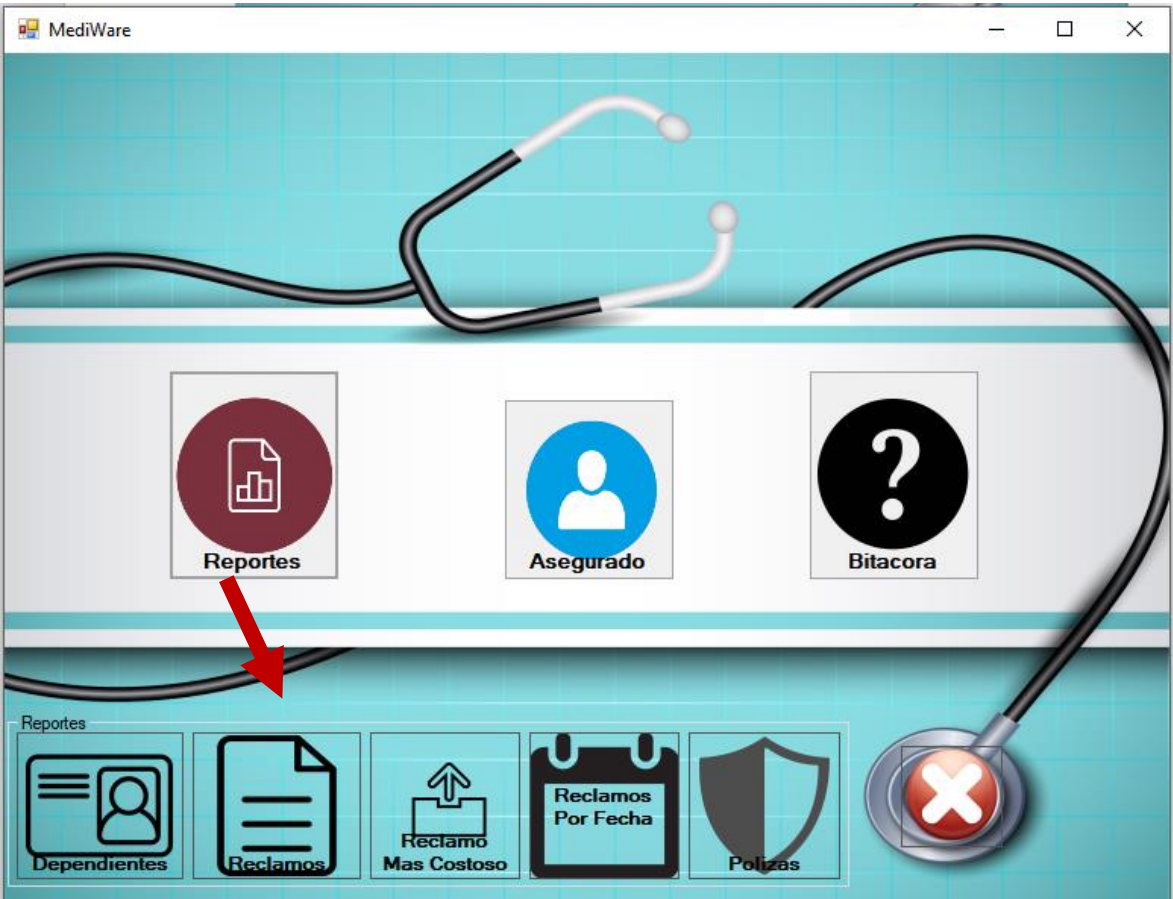
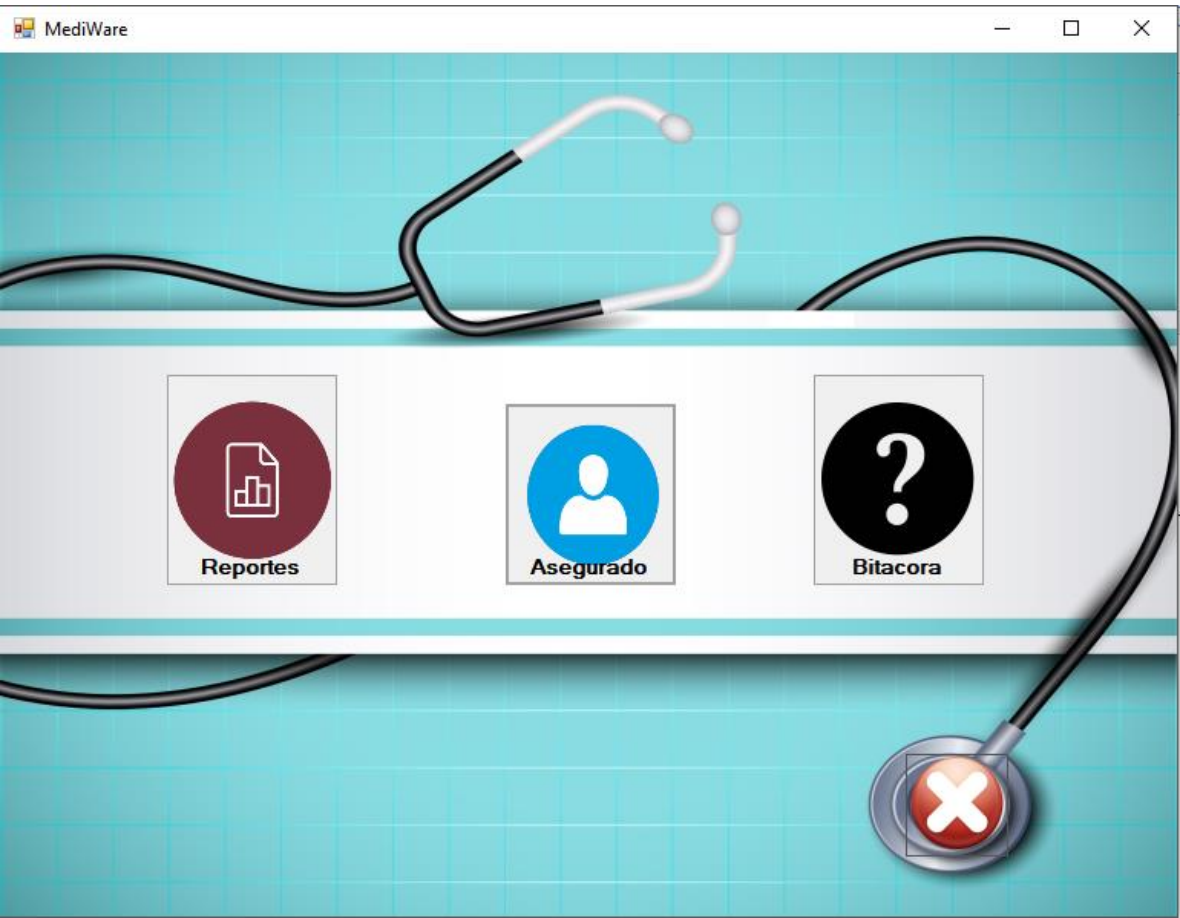
# Manual de Aplicación

## Login





Pantalla principal



Reportes de dependientes

Reportes

×

Ingrese nombre de Asegurado

Aceptar

Cancelar

Keylin Celena

Keylin Celena es la Asegurada principal y el reporte nos mostrara los que están aseguradas con la póliza de ella.

Resultado

Reporte Asegurados Dep							
Codigo	Poliza	nombre	Apellido	Fecha_Nacimiento	Estatura	Peso	Parentesco
1	2	Maria Berta	Castillo Rodriguez	12/3/1950	165	150	Madre
1	2	Juan Tomas	Palma Rodriguez	10/5/1940	175	180	Padre

Reporte de reclamos

Se nos solicitara el nombre del titular o asegurado principal:

Reportes Reclamos

×

Ingrse el Nombre del Asegurado :

Aceptar

Cancelar

Nombre 1 Nombre2

Esta muestra todos los reclamos que pertenecen a este asegurado, incluyendo las propias y la de sus dependientes.

Resultado

Reclamos Medicos						
Asegurado	Apellidos	Medico_Nombre	Apellido	Presentado	Monto	Descripcion
Keylin Celena	Castillo Palma	Berta Caceres	Espinoza Martinez	10/1/2018	2,000.00	control prenatal
Keylin Celena	Castillo Palma	Berta Caceres	Espinoza Martinez	1/5/2018	3,000.00	diabetes en familia
Keylin Celena	Castillo Palma	Berta Caceres	Espinoza Martinez	7/5/2018	90,000.00	choque automovilistico



### Reclamo más costoso

Este nos muestra el reclamo de mayor valor hasta el momento.

Reclamo Con Valor Mas Alto

Asegurado	Tatiana Marina	Gomez Gomez	Presentado 13/10/2018	Monto 280,000.00
Medico_Nombre	Berta Caceres      Espinoza Martinez			
Descripcion	choque automovilistico,por ir viendo chicas			

### Reclamos por fecha

Nos mostrara un resumen de todos los reclamos realizados en la fecha dada.

Reclamos por fecha

Ingrese Fecha de Reporte [y-m-d] :

Aceptar

Cancelar

2018-10-10

### resultados

17/12/2018						
Asegurado	Apellidos	Medico_Nombre	Apellido	Presentado	Monto	Descripcion
Bryan Tomas	Aquiles Brinco	Alejandro Magno	Andino Vasque	10/1/2018	2,000.00	control prenatal
Bryan Tomas	Aquiles Brinco	Alejandro Magno	Andino Vasque	10/1/2018	2,000.00	operacion Uterina de
Keylin Celena	Castillo Palma	Berta Caceres	Espinoza Marti	10/1/2018	2,000.00	control prenatal
Keylin Mabel	Rodriguez Baca	Jorge Mauricio	Turcios Cacere	10/1/2018	2,000.00	Aborto espontaneo
Mia Alejandra	Elvir Godoy	Carlos Alberto	elvir Godot	10/1/2018	2,000.00	Cesarea

# Pólizas

Nos Solicitara el número de póliza para mostrarnos todo respecto a la misma:

Polizas

Ingrese el numero de Poliza

Aceptar

Cancelar

2

## resultado

Poliza de Seguro						
<u>Numero</u>	2	<u>Tipo</u>	B			
<u>Asegurado</u>	Jose Alberto Castillo Rodriguez					
<u>Certificado</u>	15	<u>Inicio</u>	1/11/2017	<u>Finaliza</u>	30/12/2018	
<u>Maximo Habitacion</u>	<u>Deducible</u>	<u>Incluye Maternidad</u>	<u>% Coseguro</u>	<u>% Farmacia</u>	<u>% Rayos X</u>	<u>Otros</u>
50,000.00	#####	S	40	25	25	10
<u>Seguro de Vida</u>	600,000.00					
<u>Servicios Funebres</u>	70,000.00					

## Mantenimiento de tabla principal de asegurados

Podemos ingresar nuevos asegurados, modificar los actuales o eliminarlos. Todo esto quedara guardado en una bitácora.

[illegible]

# Bitácora

Esta nos muestra todo lo que sucede con los registros que tenemos, si fueron modificados, quedara guardado el momento, la hora, que usuario fue.

## Resultado

Bitacora					
	Usuario	Computadora	Evento	Fecha	Hora
	DENNISELITEBOOK\Dennis...	DENNISELITEBOOK	Insercion	17/12/2018	04:45:05.6275326
▶	DENNISELITEBOOK\Dennis...	DENNISELITEBOOK	Borrado	17/12/2018	04:46:57.9212053

# Conclusiones

Un correcto análisis del problema llevara a una correcta solución, es importante que podamos determinar que necesitamos para almacenar la información de manera que podamos obtener los resultados deseados.

Esto lo logramos aplicando las mejores estrategias adaptadas a nuestro contexto, el uso de herramientas correctas y siempre pensando en rendimiento, Escabilidad, seguridad e integridad de la base de datos.

lunes 17 de diciembre de 2018

Diagrama Corregido

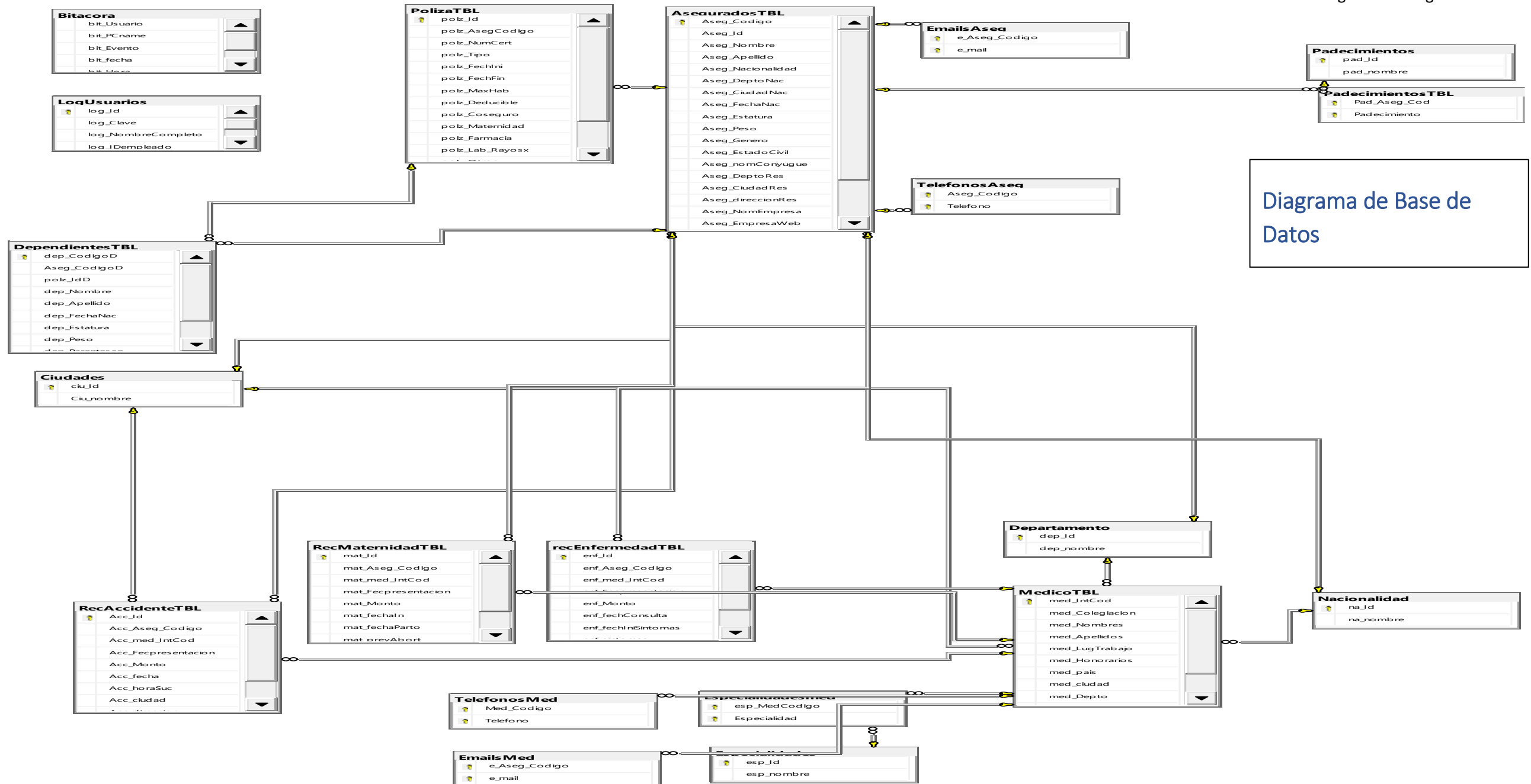


Diagrama de Base de  
Datos