**Universidad Tecnológica de Panamá**

**Facultad de Ingeniería de Sistemas Computacionales**

**Licenciatura en Ingeniería de Sistemas de Información**

**Departamento de Sistemas de Información**

Sistemas de Bases de Datos II

Equipo #2

PROYECTO SEMESTRAL

CONTROL SANITARIO EMPRESAS

Facilitador:

Ing. Henry Lezcano

Integrantes grupo 2:

Karen Cabrera 20-14-5403

Medardo Logreira 8-879-1679

Sebastian Zamora 20-15-4392

Grupo:

1IF131

II Semestre, 2020

**UNIVERSIDAD TECNOLOGICA DE PANAMA**

**FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONES**

**LICENCIATURA EN INGENIERIA DE SISTEMAS DE INFORMACION**

**SISTEMAS DE BASE DE DATOS II**

**PROYECTO SEMESTRAL: CONTROL SANITARIO EMPRESAS**

**PRIMERA PARTE - PROYECTO BASE DE DATOS I (DISEÑO).**

**• MISIÓN DE LA BASE DE DATOS**

La base de datos CONTROL SANITARIO A EMPRESAS es creada con el fin de tener un control de las empresas que estan en proceso de reapertura o que actualmente se encuentran funcionando. Con está se espera llevar un control y seguimiento de la aplicacion de las medidas sanitarias en dichas empresas. Asi como tambien saber que empleados se encuentran trabajando en las mismas de manera presencial y llevar un control de esta informacion como medida preventiva en caso de que llegara a presentarse un brote de covid 19 en alguna de estas empresas.

**• OBJETIVO DE LA MISIÓN DE LA BASE DE DATOS**

Gestionar de manera eficiente los permisos de reapertura para las diferentes empresas, fomentando la reactivacion de la economia de manera segura y en concordancia con las medidas sanitarias.

**• DEFINICIÓN DEL PROYECTO O ÁMBITO O ALCANCE O LIMITES**

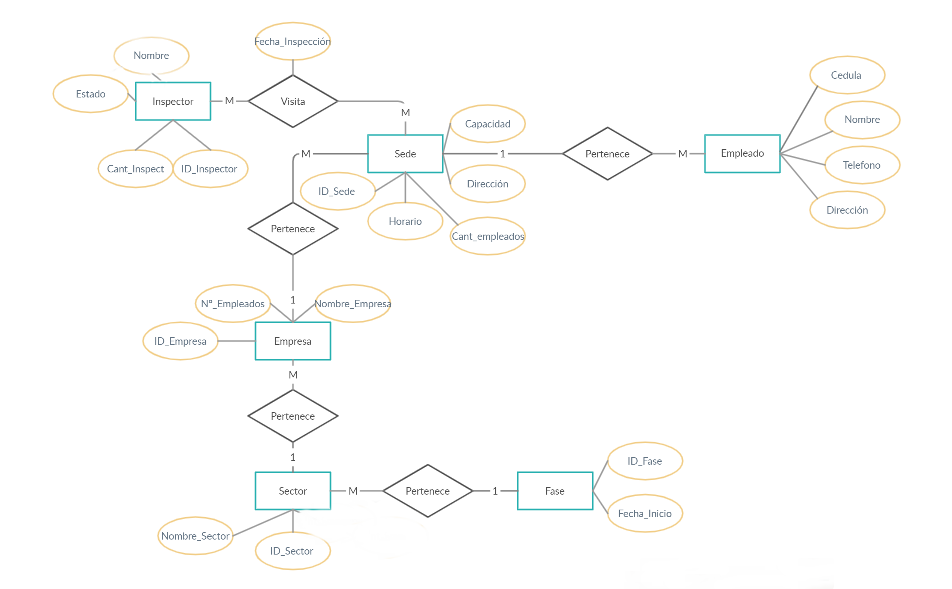
Debido a la crisis producida por el COVID-19, el gobierno nacional ha organizado un plan con el fin de realizar una reactivación de la economía en el país por medio de bloques o fases las cuales estarán compuestas de diferentes sectores económicos los mismos que a su vez están formados por diferentes empresas. El MICI como medida de control ha decidido expedir a las diferentes empresas permisos de reapertura. Sin embargo, para expedir dichos permisos se le realiza una visita de inspección a las sedes de dichas empresas en la cual se llevará control del cumplimiento de las medidas sanitarias para la contención del covid-19. Las visitas serán realizadas por un inspector del ministerio de salud que le notificará al MICI si la sede de la empresa cumple o no con las medidas para su reapertura y posteriormente se realizaran visitas seguimiento del cumplimiento de las pautas de bioseguridad.

Una vez aprobada la reapertura el MICI desea saber que empleados se encuentran trabajando en la modalidad presencial para lo cual tendrá en cuenta la identificación del empleado, nombre, cargo, empresa a la que pertenece, sede en la que trabaja, teléfono, lugar de residencia.

**REQUERIMIENTOS**

|  |  |
| --- | --- |
| Empresa | Id\_empresa (ruc) (pk)  Nombre  Numero\_empleados  Id\_sector (fk) |
| Sector | Id\_sector (pk)  Nombre  Cantidad\_empresas |
| Fase | Id\_fase  Fecha\_inicio  Id\_sector (fk) |
| Sede | Id\_sede  Id\_empresa (fk)  Dirección  Capacidad |
| Empleado | Cedula  Nombre  Cargo  Teléfono  Dirección  Id\_sede (fk) |
| Inspector | Id\_inspector  Fecha\_realizacion  Cantidad\_inspecciones  Id\_empresa (fk)  Estado |

**DIAGRAMA ENTIDAD-RELACIÓN**



**DISEÑO LÓGICO**

Inspector (ID\_Inspector (PK), nombre, Cant\_Inspect, Estado)

Visita (ID\_Inspector (FK), ID\_Sede (FK), Fecha\_Inspección)

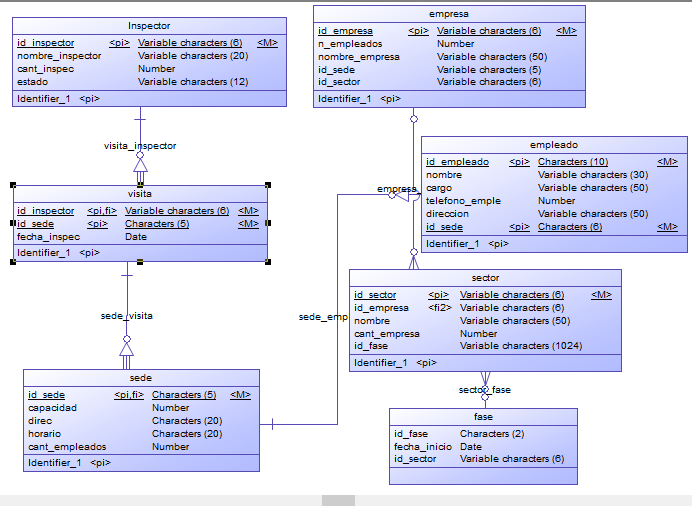
Sede (ID\_Sede (PK), Capacidad, Dirección, horario, Cant\_Empleado, id\_empresa (FK))

Empresa (ID\_Empresa (PK), Nombre\_Empresa, Nº\_Empleados, Id\_sector(FK))

Sector (ID\_Sector (PK), Nombre\_Sector, ID\_Fase(FK))

Fase (ID\_Fase (PK), Fecha\_Inicio)

Empleado (Cédula (PK), Nombre, cargo, Teléfono, Dirección, ID\_Sede (FK))



**Modelo Físico según sistema de gestión (Oracle)**

Create Table Inspector (

ID\_Inspector Varchar (6) Primary Key not null,

Nombre\_Inspector Varchar (20) ,

Cant\_Inspec Number,

Estado Varchar (12)

);

Table created.

Create Table Fase (

ID\_Fase Varchar (6) Primary Key Not Null,

Fecha\_Inicio Date

);

Create table Sector (

ID\_Sector Varchar (5) Primary Key Not Null,

Nombre\_Sector Varchar (50),

ID\_Fase Varchar (6),

Cant\_empresa Number,

Foreign Key (ID\_Fase) References Fase (ID\_Fase)

);

Create Table Empresa (

ID\_Empresa Varchar (6),

Nombre Varchar (50),

N\_Empleados Number,

ID\_Sector Varchar (5),

Foreign Key (ID\_Sector) References Sector (ID\_Sector)

);

Create Table Sede (

ID\_Sede Varchar (5) Primary Key Not Null,

Capacidad Number,

Direc Varchar (40),

Horario Varchar (25),

Cant\_Empleado Number,

ID\_Empresa Varchar (6),

Foreign Key (ID\_Empresa) References Empresa (ID\_Empresa)

);

Create Table Empleado (

ID\_Empleado Varchar (10) Primary Key Not Null,

ID\_Sede Varchar (5),

Nombre Varchar (20),

Cargo Varchar (20),

Telefono Varchar (8),

Direc Varchar (20),

Foreign Key (ID\_Sede) References Sede (ID\_Sede)

);

Create Table Visita (

ID\_Inspector Varchar (6) Primary Key Not Null,

ID\_Sede Varchar (5),

Fecha\_Inspec Date,

Foreign Key (ID\_Inspector) References Inspector (ID\_Inspector),

Foreign Key (ID\_sede) References Sede (ID\_Sede)

);

Procesos

Create Or Replace Procedure AddInspector (

P\_IDInspecT IN Inspector.ID\_Inspector%Type,

P\_NombInspe IN Inspector.Nombre\_Inspector%Type,

P\_CantInspe IN Inspector.Cant\_Inspec%Type,

P\_Estado IN Inspector.Estado%Type,

P\_Error OUT Varchar2

) AS

Begin

Insert Into Inspector

Values (P\_IDInspecT, P\_NombInspe, P\_CantInspe, P\_Estado);

Exception

When Others Then

P\_Error := 'Datos Faltantes';

Commit;

End AddInspector;

/

Create OR Replace Procedure AddFase (

P\_IDFase IN Fase.ID\_Fase%Type,

P\_FechaIni IN Fase.Fecha\_Inicio%Type,

P\_Error OUT Varchar

) AS

Begin

Insert INto Fase

Values (P\_IDFase, P\_FechaIni);

Exception

When Others Then

P\_Error := 'Datos Faltantes';

Commit;

End AddFase;

/

Create Or Replace Procedure AddSector (

P\_IDSector IN Sector.ID\_Sector%Type,

P\_NombSect IN Sector.Nombre\_Sector%Type,

P\_IDFase IN Sector.ID\_Fase%Type,

P\_CantEmrp IN Sector.Cant\_Empresa%Type,

P\_Error OUT Varchar2

) AS

Begin

Insert Into Sector

Values (P\_IDSector, P\_NombSect, P\_IDFase, P\_CantEmrp);

Exception

When Others Then

P\_Error := 'Datos Faltantes';

Commit;

End AddSector;

/

Create Or Replace Procedure AddEmpresa (

P\_IDEmpresa IN Empresa.ID\_Empresa%Type,

P\_Nombre IN Empresa.Nombre%Type,

P\_NEmpleado IN Empresa.N\_Empleados%Type,

P\_IDSector IN Empresa.ID\_Sector%Type,

P\_Error OUT Varchar2

) AS

Begin

Insert Into Empresa

Values (P\_IDEmpresa, P\_Nombre, P\_NEmpleado, P\_IDSector);

Exception

When Others Then

P\_Error := 'Datos Faltantes';

Commit;

End AddEmpresa;

/

Create or Replace Procedure AddSede (

P\_IDSede IN Sede.ID\_Sede%Type,

P\_Capacidad IN Sede.Capacidad%Type,

P\_Direccion IN Sede.Direc%Type,

P\_Horario IN Sede.Horario%Type,

P\_CantEmple IN Sede.Cant\_Empleado%Type,

P\_IDEmpresa IN Sede.ID\_Empresa%Type,

P\_Error OUT Varchar2

) AS

Begin

Insert Into Sede

Values (P\_IDSede, P\_Capacidad, P\_Direccion, P\_Horario, P\_CantEmple, P\_IDEmpresa);

Exception

When Others Then

P\_Error := 'Datos Faltantes';

Commit;

End AddSede;

/

Create Or Replace Procedure AddEmpleado (

P\_IDEmpleado IN Empleado.ID\_Empleado%Type,

P\_IDSede IN Empleado.ID\_Sede%Type,

P\_Nombre IN Empleado.Nombre%Type,

P\_Cargo IN Empleado.Cargo%Type,

P\_Telefono IN Empleado.Telefono%Type,

P\_Direc IN Empleado.Direc%Type,

P\_Error OUT Varchar2

) as

Begin

Insert INto Empleado

Values (P\_IDEmpleado, P\_IDSede, P\_Nombre, P\_Cargo, P\_Telefono, P\_Direc);

Exception

When Others Then

P\_Error := 'Datos Faltantes';

Commit;

End AddEmpleado;

/

Create or Replace Procedure AddVisita (

P\_IDInspect IN Visita.ID\_Inspector%Type,

P\_IDSede IN Visita.ID\_Sede%Type,

P\_FechaInsp IN Visita.Fecha\_Inspec%Type,

P\_Error OUT Varchar2

)As

Begin

Insert Into Visita

Values (P\_IDInspect, P\_IDSede, P\_FechaInsp);

Exception

When Others Then

P\_Error := 'Datos Faltantes';

Commit;

End AddVisita;

/

CREATE OR REPLACE TRIGGER ins\_trig AFTER UPDATE OR INSERT OR DELETE

ON Inspector FOR EACH ROW

BEGIN

IF INSERTING THEN

INSERT INTO Temp\_ins (upd\_ins, ID\_Inspector, Nombre\_Inspector , Cant\_Inspec\_v , Cant\_Inspec\_n, Estado)

VALUES (trig\_ins.nextval, :new.ID\_Inspector, :new.Nombre\_Inspector , :old.Cant\_Inspec, :new.Cant\_Inspec, :new.Estado);

ELSIF UPDATING THEN

UPDATE Temp\_ins

SET

Upd\_ins= trig\_ins.nextval,

ID\_Inspector = :new.ID\_Inspector,

Nombre\_Inspector = :new.Nombre\_Inspector,

Cant\_Inspec\_n = :OLD.Cant\_Inspec +(new.Cant\_Inspec- :OLD.Cant\_Inspec),

Cant\_Inspec\_v = :OLD.Cant\_Inspec,

Estado = :new.Estado

where Nombre\_Inspector = :new.Nombre\_Inspector AND ID\_Inspector = :new.ID\_Inspector;

END IF;

END ins\_trig;

/

create sequence trig\_ins

start with 1

increment by 1;

create table temp\_ins(

upd\_ins number,

ID\_Inspector varchar(6),

Nombre\_Inspector varchar(20),

Cant\_Inspec\_v number,

Cant\_Inspec\_n number,

Estado varchar(12)

);

Anexos

