**Script de instrucciones.**

create database institucion;

use institucion;

**Creación de tablas.**

create table alumno( noControl int primary key auto\_increment,

nombre varchar(30), apellidos varchar(60),

fechaNac date);

create table materia (clave varchar(10)primary key,

nombre varchar(150),

descripcion tinyBlob);

create table cursar (noControl int, clave varchar(10),

calificacion decimal (10,2), fecha\_reporte DATE,

primary key (noControl, clave),

constraint fk\_IDENTIFICADOR foreign key(noControl) references alumno(noControl)

on delete cascade on update cascade,

constraint fk\_MATERIA foreign key(clave) references materia (clave)

on delete cascade on update cascade);

Realizará lo siguiente:

1. **Un procedimiento para Insertar Alumnos.**

delimiter //

create procedure procesoAgregaAlumnos(in nomAlum varchar(30),

in apellidoAlum varchar(60), fecha date)

begin

insert into Alumno (nombre,apellidos,fechaNac)

values (nomAlum,apellidoAlum,fecha);

end //

call procAgregaAlumnos(”mesino”,’10-11-12’);

1. **Un procedimiento para Eliminar Alumnos.**

delimiter //

create procedure procEliminaAlumno( in numC int)

begin

delete from alumno where noControl=numC;

end //

1. **Un procedimiento para Modificar Alumnos.**

delimiter //

create procedure procModificaAlumno( in numC int,

in nomAlum Varchar(30), in apellidoAlum Varchar(60), fecha date)

begin

update Alumno

set nombre=nomAlum, apellidos=apellidoAlum, fechaNac=fecha where noControl=numC;

end //

1. **Procedimiento que realice la búsqueda con patrones(LIKE) de alumnos**

delimiter //

create procedure probusquedaalumno(letra varchar(100))

begin

select \* from alumno where nombre like letra;

end//

call proBusquedaAlumno('%o%');

1. **Función que permite calcular la edad de un alumno**

delimiter //

create function f\_calEdad ( fechaNac date, nombre varchar(30))

returns int

deterministic

begin

declare edad int;

select year(CURDATE())-YEAR(`alumno`.`fechaNac`)+

IF(DATE\_FORMAT(CURDATE(),'%m-%d') > DATE\_FORMAT(`alumno`.`fechaNac`,'%m-%d'), 0 , -1 )

into edad FROM Alumno where `alumno`.`nombre`=nombre and `alumno`.`fechaNac`=fechaNac ;

return edad;

end //

Creación de Bitácora Alumno.

CREATE TABLE bitacoraAlumno(noControl int primary key, nombreAlumno varchar(25), nombreModificar varchar(25), apellido varchar(25), apellidoModificar varchar(25), materia varchar(20), materiaModificar varchar(20), calificacion int, califModif int, fecha date, fechaModif date, accion varchar(12));

DELIMITER//

create trigger EliminarAlumno after delete on materia on cursarfor each rowinsert into bitacoraAlumno (noControl,nombreAlumno,apellido,apellidoModificar,materia,materiaModificar,calificacion,califModif,fecha,fechaModif,accion) values (old.noControl, old.nombre,new.nombreAct ,old.apellidos,new.apellidosAct,old.materia,new.materia,old.calificacion,new.califModif,old.fechaReporte, now(),'Eliminado');

END//

Cursar o nel

delimeter //

create trigger va before insert on cursar for each row

begin

if new.calificacion <= 60 thensignal sqlstate '45000' set message\_text = 'debe volver a cursar la materia';

end if;end//