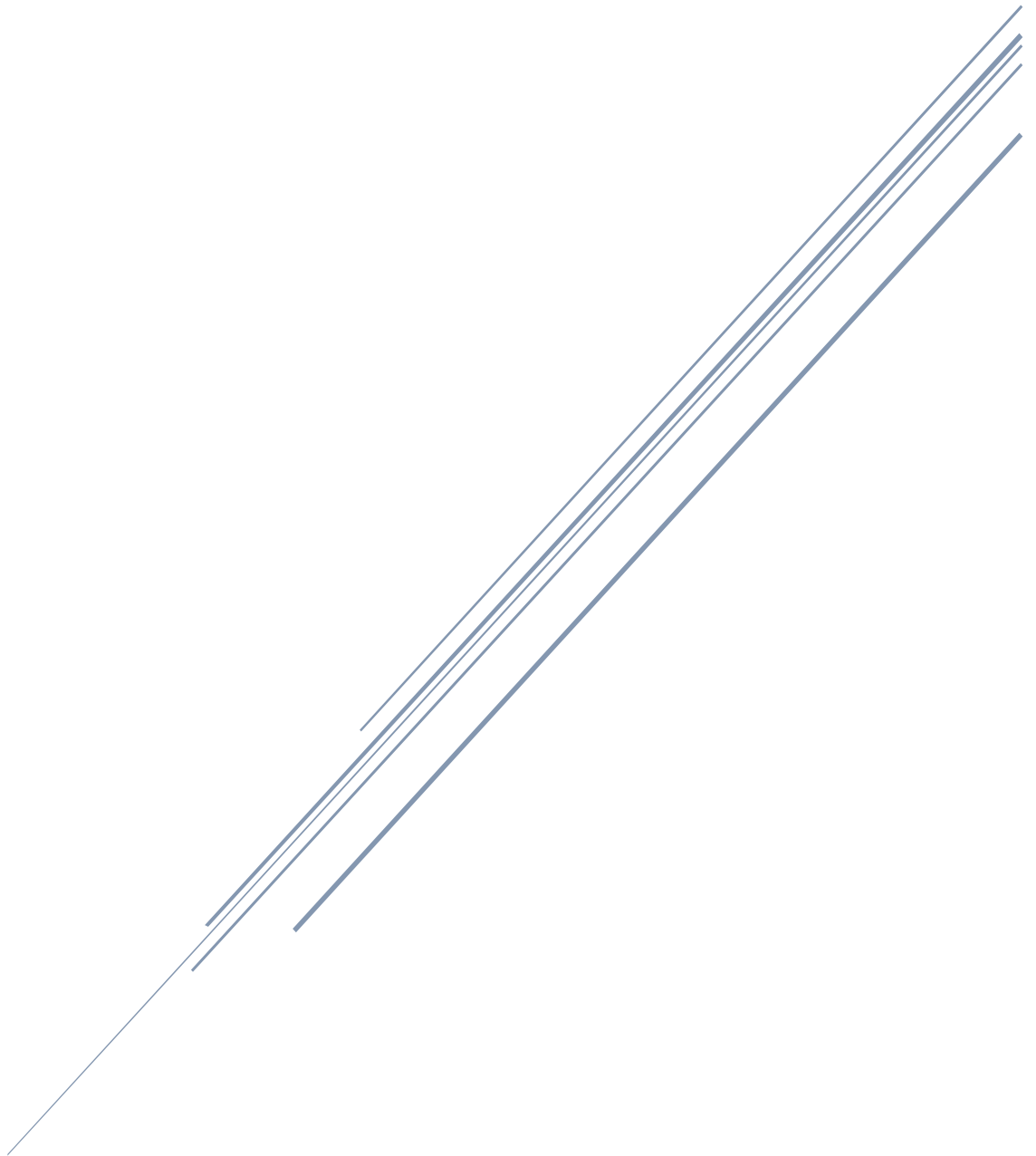


MANUAL TECNICO

Proyecto 2

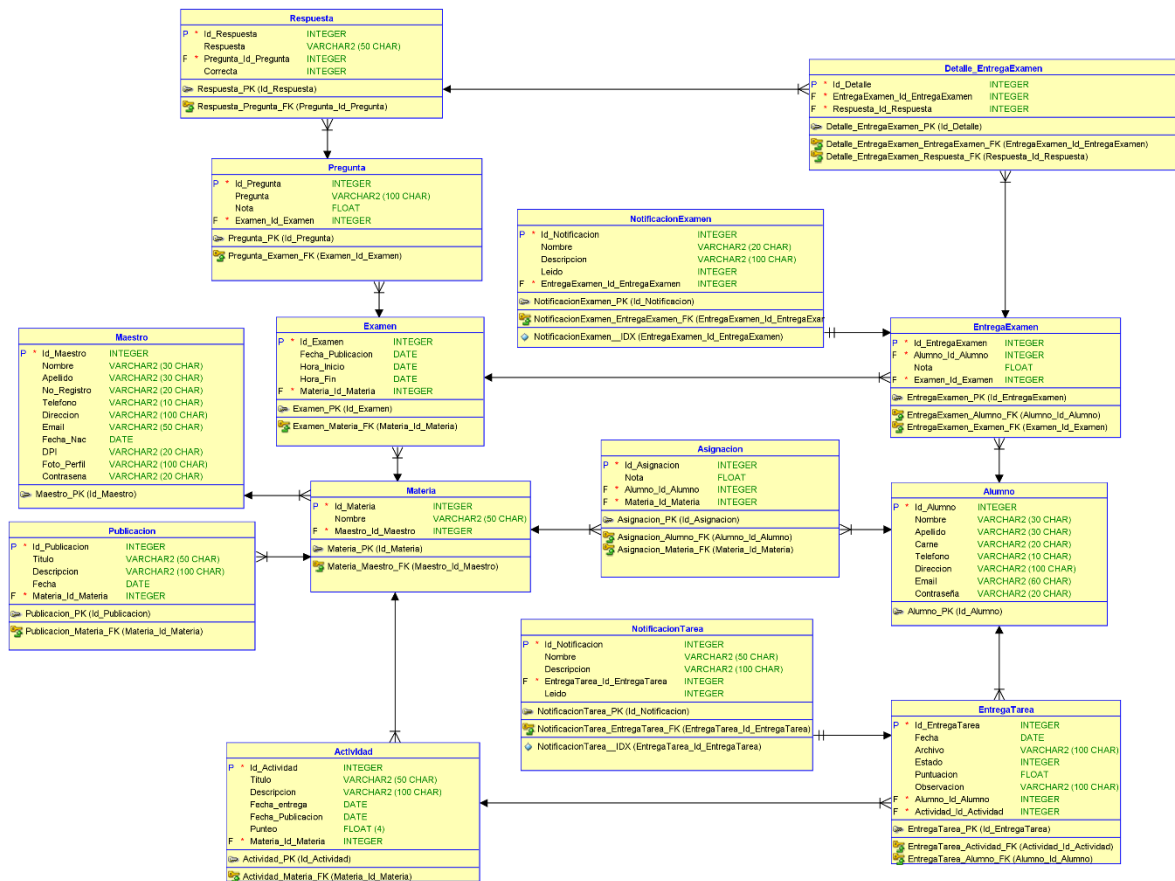


Universidad de San Carlos de Guatemala
Laboratorio de Bases de datos 1.

Índice.

Entidad Relación.....	2
Tablas.	2
Tecnologías usadas.....	3
GCP.	3
MYSQL.	3
Node js.	4
React js.	4
Script.	5
Creación base de datos.....	5
Procedimientos.....	13

Entidad Relación.



Tablas.

La figura anterior es el diagrama entidad relación que se usó para el proyecto a continuación se listaran las tablas del ER.

- Maestro
- Publicación
- Materia
- Examen
- Asignacion
- Alumno
- EntregaTarea
- NotificacionTarea
- Actividad
- EntregaExamen
- NotificacionExamen
- Detalle_EntregaExamen
- Pregunta
- Respuesta

Tecnologías usadas.

Las tecnologías que se usaron el proyecto se listan a continuación.

- GCP
- MYSQL
- Node JS
- React.

GCP.

Google Cloud Platform se trata de la suite de infraestructuras y servicios que Google utiliza a nivel interno y, ahora, disponible para cualquier empresa, de tal forma que sea aplicable a multitud de procesos empresariales.

GCP se usó para el desarrollo ya que a la hora de realizar insert en la base de datos los datos sean compartidos y así llevar un mejor control de ellos, también para realizar la base de datos en una máquina virtual.



MYSQL.

MySQL es el sistema de gestión de bases de datos relacional más extendido en la actualidad al estar basada en código abierto. Desarrollado originalmente por MySQL AB, fue adquirida por Sun Microsystems en 2008 y esta su vez comprada por Oracle Corporation en 2010, la cual ya era dueña de un motor propio InnoDB para MySQL.

Mysql se usó en el proyecto para la creación de nuestra base de datos ya que es relacional y es de fácil aprendizaje ya que usa el lenguaje sql conocido por los desarrolladores.



Node js.

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Node se usó para levantar nuestro backend ya que es muy sencillo hacer uno de ello, se usaron diferentes librerías para el desarrollo del mismo a continuación se listara algunas de ellas:

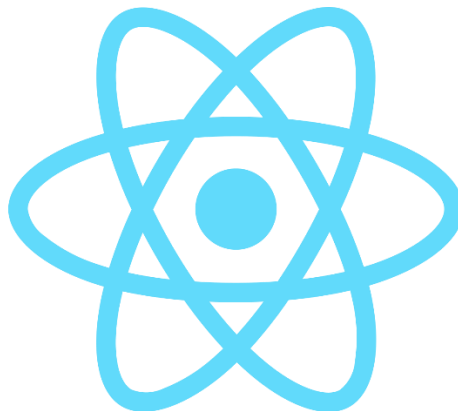
- Express
- Body-Parser
- CsvToJson
- Mysql



React js.

React es una biblioteca Javascript de código abierto diseñada para crear interfaces de usuario con el objetivo de facilitar el desarrollo de aplicaciones en una sola página. Es mantenido por Facebook y la comunidad de software libre. En el proyecto hay más de mil desarrolladores libres.

React en nuestro proyecto se usó para levantar una interfaz amigable al usuario también se usó para aprender una nueva tecnología.



Script.

Creación base de datos.

```
CREATE DATABASE proyecto2;
```

```
USE proyecto2;
```

```
DROP DATABASE proyecto2;
```

```
SHOW DATABASES;
```

```
SHOW TABLES;
```

```
-- ----- creacion de tablas -----  
-----
```

```
CREATE TABLE maestro(id_maestro INT AUTO_INCREMENT NO  
T NULL,
```

no

```
mbre VARCHAR(30) NOT NULL,
```

```
apellido VARCHAR(30) NOT NULL,
```

```
no_registro VARCHAR(20) NOT NULL,
```

```
telefono VARCHAR(10) NOT NULL,
```

```
direccion VARCHAR(100) NOT NULL,
```

```
email VARCHAR(50) NOT NULL,
```

fecha_nac DATE NOT NULL,

dpi VARCHAR(20) NOT NULL,

foto_perfil VARCHAR(100),

contrasenia VARCHAR(20) NOT NULL,

PRIMARY KEY (id_maestro));

CREATE TABLE materia (id_materia INT AUTO_INCREMENT NOT NULL,

n

ombre VARCHAR(50) NOT NULL,

id_maestro INT NOT NULL,

PRIMARY KEY (id_materia),

FOREIGN KEY (id_maestro) REFERENCES maestro(id_maestro));

CREATE TABLE publicacion (id_publicacion INT AUTO_INCREMENT NOT NULL,

titulo VARCHAR(50) NOT NULL,

descripcion VARCHAR(100) NOT NULL,

fecha DATE NOT NULL,

```
id_materia INT NOT NULL,
```

```
PRIMARY KEY (id_publicacion),
```

```
FOREIGN KEY (id_materia) REFERENCES materia(id_materia));
```

```
CREATE TABLE examen (id_examen INT AUTO_INCREMENT NOT NULL,
```

nom

```
breExamen VARCHAR(50) UNIQUE NOT NULL,
```

```
fecha_publicacion DATE NOT NULL,
```

```
hora_inicio DATETIME NOT NULL,
```

```
hora_fin DATETIME NOT NULL,
```

```
id_materia INT NOT NULL,
```

```
PRIMARY KEY (id_examen),
```

```
FOREIGN KEY (id_materia) REFERENCES materia(id_materia));
```

```
ALTER TABLE examen MODIFY COLUMN hora_inicio VARCHAR(10);
```



```
ALTER TABLE examen MODIFY COLUMN hora_fin VARCHAR(10)  
;
```

```
CREATE TABLE pregunta (id_pregunta INT AUTO_INCREMENT  
NOT NULL,
```

```
pregunta VARCHAR(100) NOT NULL,
```

```
nota FLOAT NOT NULL,
```

```
id_examen INT NOT NULL,
```

```
PRIMARY KEY (id_pregunta),
```

```
FOREIGN KEY (id_examen) REFERENCES examen(id_examen));
```

```
CREATE TABLE respuesta (id_respuesta INT AUTO_INCREMENT  
NOT NULL,
```

```
respuesta VARCHAR(50) NOT NULL,
```

```
correcta INT NOT NULL,
```

```
id_pregunta INT NOT NULL,
```

```
PRIMARY KEY (id_respuesta),
```

```
FOREIGN KEY (id_pregunta) REFERENCES pregunta(id_pregunta));
```

```
CREATE TABLE actividad (id_actividad INT AUTO_INCREMENT NOT NULL,

titulo VARCHAR(50) NOT NULL,

descripcion VARCHAR(100) NOT NULL,

fecha_entrega DATETIME NOT NULL,

fecha_publicacion DATETIME NOT NULL,

punteo FLOAT NOT NULL,

id_materia INT NOT NULL,

PRIMARY KEY(id_actividad),

FOREIGN KEY(id_materia) REFERENCES materia(id_materia));
```

```
CREATE TABLE alumno (id_alumno INT AUTO_INCREMENT NOT NULL,

nom
bre VARCHAR(30) NOT NULL,

apellido VARCHAR(30) NOT NULL,

carne VARCHAR(10) NOT NULL,

telefono VARCHAR(10) NOT NULL,
```

direccion VARCHAR(100) NOT NULL,

email VARCHAR(60) NOT NULL,

contrasenia VARCHAR(20) NOT NULL,

PRIMARY KEY (id_alumno));

CREATE TABLE entrega_tarea(id_entrega_tarea INT AUTO_INCREMENT NOT NULL,

fecha DATETIME NOT NULL,

archivo VARCHAR(100) NOT NULL,

estado INT NOT NULL,

puntuacion FLOAT NOT NULL,

observacion VARCHAR(100) NOT NULL,

id_alumno INT NOT NULL,

id_actividad INT NOT NULL,

PRIMARY KEY (id_entrega_tarea),

FOREIGN KEY (id_alumno) **REFERENCES** alumno(id_alumno),

FOREIGN KEY (id_actividad) **REFERENCES** actividad(id_actividad));

```
CREATE TABLE notificacion_tarea(id_notificacion_tarea
INT AUTO_INCREMENT NOT NULL,

nombre VARCHAR(50) NOT NULL,

descripcion VARCHAR(100) NOT NULL,

leido INT NOT NULL,

id_entrega_tarea INT NOT NULL,

PRIMARY KEY (id_notificacion_tarea),
FOREIGN KEY (id_entrega_tarea) REFERENCES entrega_tarea(id_entrega_tarea)
;
```

```
CREATE TABLE entrega_examen (id_entrega_examen INT AU
TO_INCREMENT NOT NULL,

nota FLOAT NOT NULL,

id_alumno INT NOT NULL,

id_examen INT NOT NULL,

PRIMARY KEY (id_entrega_examen),
FOREIGN KEY (id_alumno)
REFERENCES alumno(id_alumno),
```

```
FOREIGN KEY (id_examen)  
REFERENCES examen(id_examen));
```

```
CREATE TABLE detalle_entrega_examen (id_detalle_entrega_examen INT AUTO_INCREMENT NOT NULL,
```

```
id_entrega_examen INT NOT NULL,
```

```
id_respuesta INT NOT NULL,
```

```
PRIMARY KEY (id_detalle_entrega_examen),
```

```
FOREIGN KEY (id_entrega_examen) REFERENCES entrega_examen(id_entrega_examen),
```

```
FOREIGN KEY (id_respuesta) REFERENCES respuesta(id_respuesta));
```

```
CREATE TABLE notificacion_examen (id_notificacion_examen INT AUTO_INCREMENT NOT NULL,
```

```
nombre VARCHAR(20) NOT NULL,
```

```
descripcion VARCHAR(100) NOT NULL,
```

```
leido INT NOT NULL,
```

```
id_entrega_examen INT NOT NULL,
```

```
PRIMARY KEY (id_notificacion_examen),
```

```
FOREIGN KEY (id_entrega_examen) REFERENCES entrega_examen(id_entrega_examen));
```

```
CREATE TABLE asignacion (id_asignacion INT AUTO_INCREMENT NOT NULL,
```

```
nota FLOAT NOT NULL,
```

```
id_alumno INT NOT NULL,
```

```
id_materia INT NOT NULL,
```

```
PRIMARY KEY (id_asignacion),
```

```
FOREIGN KEY (id_alumno) REFERENCES alumno(id_alumno),
```

```
FOREIGN KEY (id_materia) REFERENCES materia(id_materia));
```

Procedimientos.

```
DELIMITER $$
```

```
CREATE PROCEDURE add_alumno(IN nombre VARCHAR(30), IN apellido VARCHAR(30), IN carne VARCHAR(10), IN telefono VARCHAR(10), IN direccion VARCHAR(100), IN email VARCHAR(60), IN contrasenia VARCHAR(20))
```

```
BEGIN
```

```
INSERT INTO alumno (nombre, apellido, carne, telefono, direccion, email, contrasenia)
```

```
VALUES (nombre, apellido, carne, telefono, direccion, email, contrasenia);
```

```
END$$
```

```
DELIMITER ;
```

```
CALL add_alumno('Ariel', 'Macario', '201905837', '310  
03016', 'El por venir', 'arielmacario11@gmail.com', '  
katiteamoxd');
```

```
SELECT *
```

```
FROM alumno;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_maestro(IN nombre VARCHAR(30), I  
N apellido VARCHAR(30), IN no_registro VARCHAR(20), I  
N telefono VARCHAR(100), IN direccion VARCHAR(100), I  
N email VARCHAR(50), IN fecha_nac varchar(50) , IN dp  
i VARCHAR(20), IN foto_perfil VARCHAR(100), IN contra  
senia VARCHAR(20))
```

```
BEGIN
```

```
INSERT INTO maestro (nombre, apellido, no_  
registro, telefono, direccion, email, fecha_nac , dpi  
, foto_perfil, contrasenia)
```

```
VALUES (nombre, apellido, no_registro, tel  
efono, direccion, email, str_to_date(fecha_nac, '%m/%d  
/%Y') , dpi, foto_perfil, contrasenia);
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_materia(IN nombre VARCHAR(50), IN id_maestro INT)
```

```
    BEGIN
```

```
        INSERT INTO materia (nombre, id_maestro)
```

```
        VALUES (nombre, id_maestro);
```

```
END $$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_publicacion(IN titulo VARCHAR(50), IN descripcion VARCHAR(100), IN fecha varchar(50), IN id_materia INT)
```

```
    BEGIN
```

```
        INSERT INTO publicacion (titulo, descripcion, fecha, id_materia)
```

```
        VALUES (titulo, descripcion, str_to_date(fecha, '%m/%d/%Y'), id_materia);
```

```
END $$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_examen(IN nombreExamen VARCHAR(50), IN fecha_publicacion DATE, IN hora_inicio varchar(10), IN hora_fin varchar(10), IN id_materia INT)
```

```
    BEGIN
```

```
        INSERT INTO examen (nombreExamen, fecha_publicacion, hora_inicio, hora_fin, id_materia)
```



```
VALUES (nombreExamen, fecha_publicacion, h  
ora_inicio, hora_fin, id_materia);
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_pregunta(IN pregunta VARCHAR(100  
) , IN nota FLOAT, IN id_examen INT)
```

```
BEGIN
```

```
INSERT INTO pregunta (pregunta, nota, id_e  
xamen)
```

```
VALUES (pregunta, nota, id_examen);
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_respuesta(IN respuesta VARCHAR(5  
0), IN correcta INT, IN id_pregunta INT)
```

```
BEGIN
```

```
INSERT INTO respuesta (respuesta, correcta  
, id_pregunta)
```

```
VALUES (respuesta, correcta, id_pregunta);
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_actividad(IN titulo VARCHAR(50),  
IN descripcion VARCHAR(100), IN fecha_entrega DATETIME,  
IN fecha_publicacion DATETIME, IN punteo FLOAT, IN  
id_materia INT)
```

```
BEGIN
```

```
INSERT INTO actividad (titulo, descripcion  
, fecha_entrega, fecha_publicacion, punteo, id_materia)
```

```
VALUES (titulo, descripcion, fecha_entrega  
, fecha_publicacion, punteo, id_materia);
```

```
END$$
```

```
DELIMITER ;
```

```
DROP PROCEDURE add_entrega_tarea;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_entrega_tarea(IN archivo VARCHAR  
(100), IN estado INT, IN puntuacion FLOAT, IN observacion  
VARCHAR(100), IN id_alumno INT, IN id_actividad  
INT)
```

```
BEGIN
```

```
INSERT INTO entrega_tarea (fecha, archivo,  
estado, puntuacion, observacion, id_alumno, id_actividad)
```

```
VALUES (now(), archivo, estado, puntuacion  
, observacion, id_alumno, id_actividad);
```

```
END$$$
```

```
DELIMITER ;
```

DELIMITER \$\$

CREATE PROCEDURE add_notificacion_tarea(**IN** nombre **VAR**
CHAR(50), **IN** descripcion **VARCHAR**(100), **IN** leido **INT**,
IN id_entrega_tarea **INT**)

BEGIN

INSERT INTO notificacion_tarea (nombre, de
scripcion, leido, id_entrega_tarea)

VALUES (nombre, descripcion, leido, id_ent
rega_tarea);

END\$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE add_entrega_examen(**IN** nota **FLOAT**, **IN**
id_alumno **INT**, **IN** id_examen **INT**)

BEGIN

INSERT INTO entrega_examen (nota, id_alumn
o, id_examen)

VALUES (nota, id_alumno, id_examen);

END\$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE add_detalle_entrega_examen(**IN** id_ent
rega_examen **INT**, **IN** id_respuesta **INT**)

BEGIN

```
        INSERT INTO detalle_entrega_examen (id_entrega_examen, id_respuesta)
        VALUES (id_entrega_examen, id_respuesta);
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_notificacion_examen(IN nombre VARCHAR(20), IN descripcion VARCHAR(100), IN leido INT, IN id_entrega_examen INT)
```

```
    BEGIN
```

```
        INSERT INTO notificacion_examen (nombre, descripcion, leido , id_entrega_examen)
```

```
        VALUES (nombre, descripcion, leido , id_entrega_examen);
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE add_asignacion(IN nota FLOAT, IN id_alumno INT, IN id_materia INT)
```

```
    BEGIN
```

```
        INSERT INTO asignacion (nota, id_alumno, id_materia)
```

```
        VALUES (nota, id_alumno, id_materia);
```

```
END$$
```

```
DELIMITER ;
```

DELIMITER \$\$

CREATE PROCEDURE consultarAlumnos()

BEGIN

Select * **from** alumno **where** carne = 2019096
75;

END \$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE consultarMaestros()

BEGIN

Select * **from** maestro;

END \$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE consultaActividadxAlumno(**in** id **int**)

BEGIN

Select p.id_actividad, p.titulo, p.descripcion, DATE_FORMAT(p.fecha_entrega, '%Y-%m-%d') **as** fechaEntrega, DATE_FORMAT(p.fecha_publicacion, '%Y-%m-%d') **as** fechaPublicacion, p.punteo, (0) **as** Entregado, p.id_materia, m.nombre **from** actividad p

inner join asignacion a **on** a.id_materia = p.id_materia

inner join materia m **on** a.id_materia = m.id_materia

```
-- inner join entrega_tarea e on e.id_actividad  
ad = p.id_actividad
```

```
where a.id_alumno = id
```

```
;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultaEntregaActividadxAlumno(in id  
int)
```

```
BEGIN
```

```
    Select e.id_entrega_tarea, DATE_FORMAT(e.  
fecha, '%Y-%m-%d') as fecha, e.estado, e.puntuacion, e  
.id_actividad from entrega_tarea e
```

```
where e.id_alumno = id
```

```
;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultarPublicacionxAlumno(in id in  
int)
```

```
BEGIN
```

```
    Select p.id_publicacion, p.titulo, p.desc  
ripcion, DATE_FORMAT(p.fecha, '%Y-%m-%d') as fecha, p.  
id_materia from publicacion p
```

```
        inner join asignacion a on a.id_materia = p.id_materia
        where a.id_alumno = id
    ;
END $$
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultaPublicacionxMaestro(in id int)
```

```
    BEGIN
        if(id <> 0) then
            begin
                Select p.id_publicacion, p.titulo, p.descripcion, DATE_FORMAT(p.fecha, '%Y-%m-%d') as fecha, p.id_materia, m.nombre from publicacion p
                inner join materia m on m.id_materia = p.id_materia
                where m.id_maestro = id;
            end;
        else
            begin
                Select p.id_publicacion, p.titulo, p.descripcion, DATE_FORMAT(p.fecha, '%Y-%m-%d') as fecha, p.id_materia from publicacion p;
            end;
        END IF;
```

END \$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE consultaActividadxMaestro(**in** id **in**
t)

BEGIN

if(id <> 0) **then**

begin

Select p.id_actividad, p.titulo, p.descripcion,
DATE_FORMAT(p.fecha_entrega, '%Y-%m-%d') **as** fecha_entrega,
DATE_FORMAT(p.fecha_publicacion, '%Y-%m-%d') **as** fecha_publicacion,
p.punteo, p.id_materia, m.nombre **from** actividad p

inner join materia m **on** m.id_materia = p.id_materia

where m.id_maestro = id;

end;

else

begin

Select p.id_actividad, p.titulo, p.descripcion,
DATE_FORMAT(p.fecha_entrega, '%Y-%m-%d') **as** fecha_entrega,
DATE_FORMAT(p.fecha_publicacion, '%Y-%m-%d') **as** fecha_publicacion,
p.punteo, p.id_materia, m.nombre **from** actividad p

inner join materia m **on** m.id_materia = p.id_materia;

end;

END IF;

END \$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE consultaExamenxMaestro(**in** id **int**)

BEGIN

if(id <> 0) **then**

begin

Select p.id_examen,p.nombreExamen,DATE_FORMAT(p.fecha_publicacion, '%Y-%m-%d') **as** fecha_publicacion,DATE_FORMAT(p.hora_inicio, '%Y-%m-%d %H:%i') **as** horaInicio,DATE_FORMAT(p.hora_fin, '%Y-%m-%d %H:%i') **as** horaFin,p.id_materia ,m.nombre **from** examen p

inner join materia m **on** m.id_materia = p.id_materia

where m.id_maestro = id;

end;

else

begin

Select p.id_examen,p.nombreExamen,DATE_FORMAT(p.fecha_publicacion, '%Y-%m-%d') **as** fecha_publicacion,DATE_FORMAT(p.hora_inicio, '%Y-%m-%d %H:%i') **as** horaInicio,DATE_FORMAT(p.hora_fin, '%Y-%m-%d %H:%i') **as** horaFin,p.id_materia ,m.nombre **from** examen p

inner join materia m **on** m.id_materia = p.id_materia;

end;

END IF;

END \$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE consultarAlumnosxMateria(**in** id **int**)

BEGIN

Select alum.carne,alum.nombre,act.titulo,a
.puntuacion **from** entrega_tarea a

inner join alumno alum **on** alum.id_alumno = a.
id_alumno

inner join actividad act **on** act.id_actividad
= a.id_actividad

inner join materia mat **on** act.id_materia = ma
t.id_materia

where mat.id_materia = id

group by alum.carne,alum.nombre,act.titulo,a.
puntuacion

;

END \$\$

DELIMITER ;

DELIMITER \$\$

CREATE PROCEDURE consultarAlumnosAsigxMateria(**in** id **i**
nt)

BEGIN

Select alum.carne,alum.nombre,alum.apellid
o **from** asignacion a

```
        inner join alumno alum on alum.id_alumno = a.  
id_alumno  
        where a.id_materia = id  
    ;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultarMateria()
```

```
    BEGIN
```

```
        Select * from materia;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultarActividad()
```

```
    BEGIN
```

```
        Select * from actividad;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultarmateriamaestro(in id int)
```

```
    BEGIN
```

```

        Select mat.id_materia,mat.nombre from materia mat
        inner join maestro maes on mat.id_maestro = maes.id_maestro
        where maes.id_maestro = id;
END $$
DELIMITER ;

```

```

DELIMITER $$
CREATE PROCEDURE consultaentregaTarea(in idEstudiante
int,in id_Actividad int)
    BEGIN
        Select mat.id_materia,mat.nombre from entrega_tarea ent
        -- inner join alumno a on a.id_alumno = ent.id_alumno
        -- inner join actividad ac on ac.id_actividad = a.id_actividad
        where ent.id_alumno = idEstudiante AND ent.id_
actividad = id_Actividad;
END $$
DELIMITER ;

```

```

DELIMITER $$
CREATE PROCEDURE notificaciones(in idEstudiante int)
    BEGIN
        Select * from notificacion_tarea notf

```

```
        inner join entrega_tarea e on e.id_entrega_tarea = notf.id_entrega_tarea
        where e.id_alumno = id
    ;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultaactividadxmateria(in id int)
```

```
    BEGIN
```

```
        Select * from actividad
        where id_materia = id
    ;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultaMateriAxAlumnos(in id int)
```

```
    BEGIN
```

```
        Select m.id_materia,m.nombre from asignacion a
        inner join materia m on a.id_materia = m.id_materia
        where a.id_alumno = id
    ;
```

```
END$$
```

```
DELIMITER ;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE consultarActividadxMateria(in id int  
, in idALumno int)
```

```
BEGIN
```

```
Select act.titulo,a.puntuacion from entreg  
a_tarea a
```

```
inner join actividad act on act.id_actividad  
= a.id_actividad
```

```
inner join materia mat on act.id_materia = ma  
t.id_materia
```

```
where mat.id_materia = id AND a.id_alumno = i  
dALumno;
```

```
END$$
```

```
DELIMITER ;
```