



# EVALUACION PROCESUAL HITO 2



BASE DE DATOS II

ESTUDIANTE: NILBER MAYTA CUNO



# PARTE TEORICA

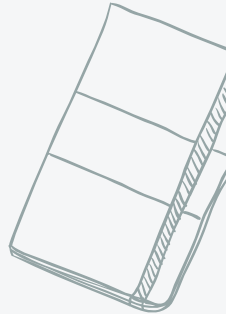




# MANEJO DE CONCEPTOS

## *1. ¿A que se refiere cuando se habla de bases de datos relacionales?*

Se refiere al tipo de base de datos la cual almacena datos mediante el uso de tablas con el uso de filas y columnas. Se usa la relación de estas para el manejo de datos.



## *2. ¿A que se refiere cuando se habla de bases de datos no relacionales?*

Son aquellas que almacenan datos pero no mediante tablas sino con una conexión directa.

### 3. ¿Qué es MySQL y MariaDB?. Explique si existen diferencias o son iguales, etc.

#### MySQL

Es el programa mediante el cual podemos crear las bases de datos y añadirle registros.



#### MariaDB

Es donde se almacena todas las bases de datos que creamos, así también los archivos que contengan.




#### 4. ¿Qué son las funciones de agregación?

Es el uso de comandos que nos permiten computar un valor mediante un conjunto de entrada como ser el count, max, sum, etc.

```
select avg(est.edad)
from estudiantes as est
    inner join inscripcion as ins on est.id_est = ins.id_est
    inner join materias as mat on ins.id_mat = mat.id_mat
where est.sexo = 'masculino' and mat.cod_mat = 'ARQ-104';
```



## 5. ¿Qué llegaría a ser XAMPP, WAMP SERVER o LAMP?



Los servidores XAMPP, WAMP SERVER son servidores en la nube o servidores web mientras que LAMP es un servidor físico que tenemos a mano como MariaDB o SQLServer

## 6. ¿Cual es la diferencia entre las funciones de agregación y funciones creados por el DBA? Es decir funciones creadas por el usuario

Las funciones de ,agregación son funciones que ya están implementadas en el mismo editor como ser el count, max, min, etc mientras que las DBA son aquellas que podemos crear nosotros mismos.

```
select avg(est.edad)
```

```
create or replace function ConcatenarParametros(par1 varchar(100), par2 varchar(100), par3 integer)
returns varchar (200)
begin
declare resp varchar(200) ;
select concat('('||par1||'),('||par2||'),('||par3||')) into resp;
return resp;
end;
```



## 7. ¿Para qué sirve el comando USE?

Este comando lo utilizamos para seleccionar la base de datos con la cual queremos trabajar ya que en un servidor pueden haber muchas bases de datos.

```
use POLLOS_COPA;
```

## 8. Que es DML y DDL?

El DDL se refiere al lenguaje a Data Definition Language (Lenguaje de definicion de datos) es decir aspectos como crear, alterar o eliminar datos.

El DML se refiere al lenguaje a Data Manipulation Language (Lenguaje de manipulaci3n de datos) es decir aspectos como mostrar, insertar o actualizar datos.

# Ejemplo

## DDL

```
create table pedido(  
  id_pedido integer auto_increment primary key not null,  
  articulo varchar(30) not null ,  
  costo integer not null ,  
  fecha date  
);
```

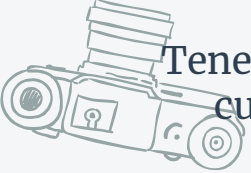
## DML

```
insert into pedido (articulo, costo, fecha)  
values  
  ('Alitas', 25, '2023-03-27'),  
  ('Balde grande', 50, '2023-03-28');
```

9. ¿Qué cosas características debe de tener una función? Explique sobre el nombre, el return, parametros, etc.

Primero tenemos como modelos de función el siguiente modelo

```
create or replace function nombre()  
    returns varchar(30)  
begin  
    return 'Nilber Mayta';  
end;
```



Tenemos el nombre con el cual llamaremos a la funcion

```
create or replace function nombre()  
  returns varchar(30)  
begin  
  return 'Nilber Mayta';  
end;
```


Tenemos un inicio y un final

```
create or replace function nombre()  
  returns varchar(30)  
begin  
  return 'Nilber Mayta';  
end;
```



El parametron y tipo de variable que nos retornara

```
create or replace function nombre()  
  returns varchar(30)  
begin  
  return 'Nilber Mayta';  
end;
```

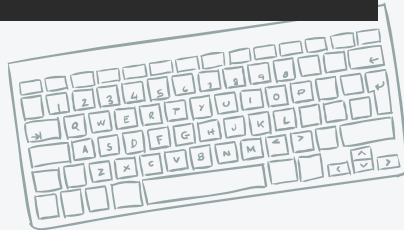


El proceso que realizara y lo que nos retornara

```
create or replace function nombre()  
  returns varchar(30)  
begin  
  return 'Nilber Mayta';  
end;
```

Y para llamarla usaremos el commando select

```
select nombre();
```



## 10. ¿Cómo crear, modificar y cómo eliminar una función?

Para la creación o modificación es recomendable usar el comando

```
create or replace function
```

Este nos crea la función y si es que existiera la modifica

Y para poder eliminarla solo usamos el comando

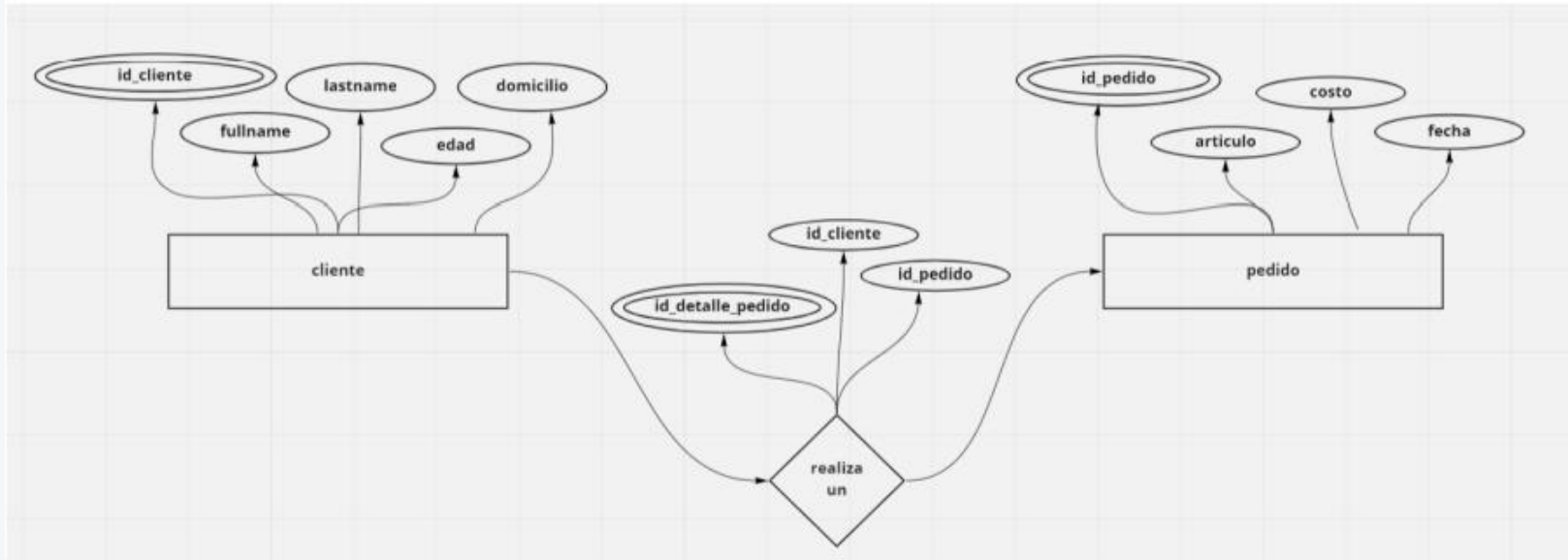
```
drop function
```

Seguido del nombre de la función

# PARTE PRACTICA



## 11. Crear las tablas y 2 registros para cada tabla para el siguiente modelo ER





Se sugiere crear una base de datos de nombre  
POLLOS\_COPA y en ella crear las tablas: ■ cliente ■  
detalle\_pedido ■ pedido

```
create database POLLOS_COPA;
```

```
use POLLOS_COPA;
```

```
create table cliente(  
  id_cliente integer auto_increment primary key not null,  
  fullname varchar(20) not null,  
  lastname varchar(20) not null,  
  edad integer not null,  
  domicilio varchar(50) not null  
);
```

```
create table pedido(  
  id_pedido integer auto_increment primary key not null,  
  articulo varchar(30) not null ,  
  costo integer not null ,  
  fecha date  
);
```

```
create table detalle_pedido(  
  id_detalle_pedido integer auto_increment primary key not null ,  
  id_cliente integer not null ,  
  id_pedido integer not null ,  
  foreign key (id_cliente) references cliente(id_cliente),  
  foreign key (id_pedido) references pedido(id_pedido)  
);
```

## Agregamos registros a las tablas

```
insert into cliente (fullname, lastname, edad, domicilio)
values
    ('Juan', 'Pérez', 35, 'Calle 123, Ciudad'),
    ('María', 'García', 28, 'Av. Principal, Pueblo');
```

```
insert into pedido (articulo, costo, fecha)
values
    ('Alitas', 25, '2023-03-27'),
    ('Balde grande', 50, '2023-03-28');
```

```
insert into detalle_pedido (id_cliente, id_pedido)
values
    (1, 1),
    (2, 2);
```

## 12. Crear una consulta SQL en base al ejercicio anterior

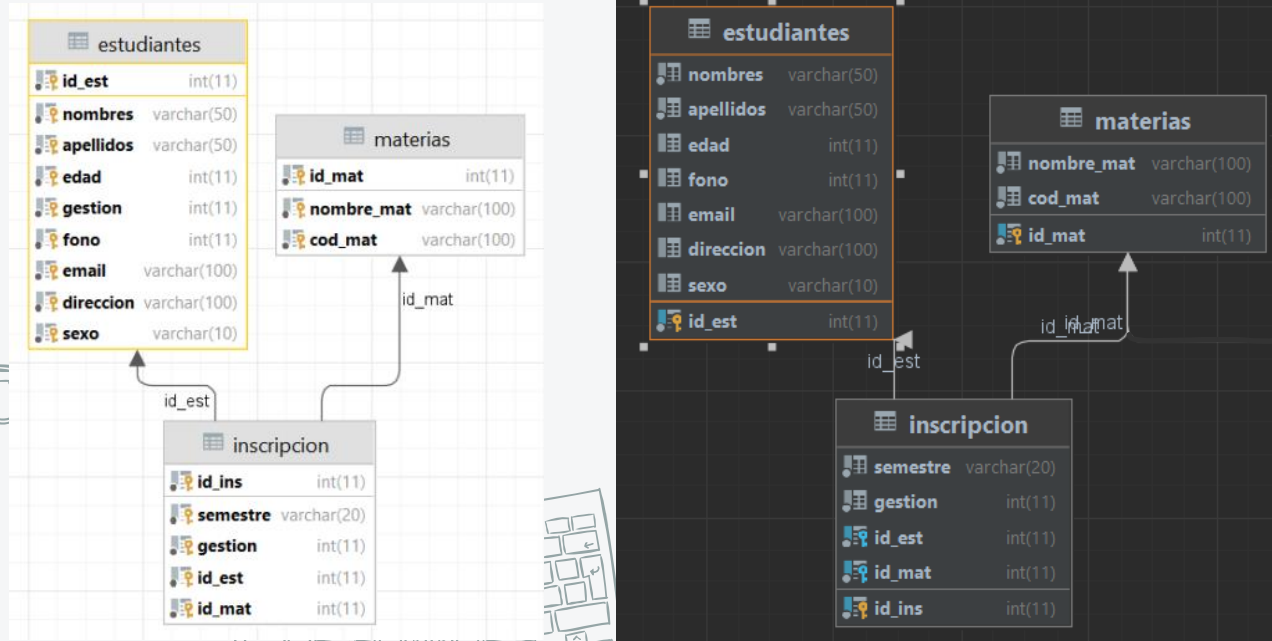
- Debe de utilizar las 3 tablas creadas anteriormente.
- Para relacionar las tablas utilizar JOINS.
- Adjuntar el código SQL generado.

```
select cli.fullname, pe.id_pedido, cli.domicilio
from pedido as pe
inner join detalle_pedido as det on det.id_pedido = pe.id_pedido
inner join cliente as cli on det.id_cliente = cli.id_cliente
where cli.fullname = 'Juan'
```

|   | fullname | id_pedido | domicilio         |
|---|----------|-----------|-------------------|
| 1 | Juan     | 1         | Calle 123, Ciudad |

## 13. Crear un función que compare dos códigos de materia

- *Recrear la siguiente base de datos:*



# Creamos tablas

```
create table estudiantes(
  id_est integer auto_increment primary key not null,
  nombres varchar (50) not null ,
  apellidos varchar (50) not null ,
  edad integer ,
  fono integer ,
  email varchar(100),
  direccion varchar(100),
  sexo varchar(10)
);
```

```
create table materias(
  id_mat integer auto_increment primary key not null ,
  nombre_mat varchar(100) not null ,
  cod_mat varchar (100) not null
);
```

```
create table inscripcion (
  id_ins integer auto_increment primary key not null ,
  semestre varchar(20) not null ,
  gestion integer not null ,
  id_est integer not null ,
  id_mat integer not null ,
  foreign key (id_est) references estudiantes(id_est),
  foreign key (id_mat) references materias (id_mat)
);
```

# Insertamos los registros

```
INSERT INTO estudiantes (nombres, apellidos, edad, fono, email,
                        direccion, sexo)
VALUES ('Miguel', 'Gonzales Veliz', 20, 2832115, 'miguel@gmail.com', 'Av. 6 de Agosto', 'masculino'),
       ('Sandra', 'Mavir Uria', 25, 2832116, 'sandra@gmail.com', 'Av. 6 de Agosto', 'femenino'),
       ('Joel', 'Adubiri Mondar', 30, 2832117, 'joel@gmail.com', 'Av. 6 de Agosto', 'masculino'),
       ('Andrea', 'Arias Ballesteros', 21, 2832118, 'andrea@gmail.com', 'Av. 6 de Agosto', 'femenino'),
       ('Santos', 'Montes Valenzuela', 24, 2832119, 'santos@gmail.com', 'Av. 6 de Agosto', 'masculino');
```

```
INSERT INTO materias (nombre_mat, cod_mat)
VALUES ('Introduccion a la Arquitectura', 'ARQ-101'),
       ('Urbanismo y Diseno', 'ARQ-102'),
       ('Dibujo y Pintura Arquitectonico', 'ARQ-103'),
       ('Matematica discreta', 'ARQ-104'),
       ('Fisica Basica', 'ARQ-105');
```

```
INSERT INTO inscripcion (id_est, id_mat, semestre, gestion)
VALUES (1, 1, '1er Semestre', 2018),
       (1, 2, '2do Semestre', 2018),
       (2, 4, '1er Semestre', 2019),
       (2, 3, '2do Semestre', 2019),
       (3, 3, '2do Semestre', 2020),
       (3, 1, '3er Semestre', 2020),
       (4, 4, '4to Semestre', 2021),
       (5, 5, '5to Semestre', 2021);
```

○ Resolver lo siguiente:

- Mostrar los nombres y apellidos de los estudiantes inscritos en la materia ARQ-105, adicionalmente mostrar el nombre de la materia.
- Deberá de crear una función que reciba dos parámetros y esta función deberá ser utilizada en la cláusula WHERE.

Primero hacemos una consulta

```
select est.id_est, est.nombres, est.apellidos, mat.nombre_mat, cod_mat
from estudiantes as est
inner join inscripcion as ins on est.id_est = ins.id_est
inner join materias as mat on ins.id_mat = mat.id_mat
where mat.cod_mat = 'ARQ-105';
```

|   | id_est | nombres | apellidos         | nombre_mat    | cod_mat |
|---|--------|---------|-------------------|---------------|---------|
| 1 | 5      | Santos  | Montes Valenzuela | Fisica Basica | ARQ-105 |

Creamos la función que recibe dos parámetros y compara dos materias

```
create or replace function comparaMaterias(codmat varchar(30), veri varchar(30))  
    returns bool  
begin  
    declare respuesta bool default false;  
    if codmat = veri then  
        set respuesta = true;  
    end if;  
    return respuesta;  
end;
```

Aplicamos la función a nuestra consulta

```
select est.id_est, est.nombres, est.apellidos, mat.nombre_mat, cod_mat  
from estudiantes as est  
    inner join inscripcion as ins on est.id_est = ins.id_est  
    inner join materias as mat on ins.id_mat = mat.id_mat  
where comparaMaterias( codmat: mat.cod_mat, veri: 'ARQ-105');
```

|   | id_est | nombres | apellidos         | nombre_mat    | cod_mat |
|---|--------|---------|-------------------|---------------|---------|
| 1 | 5      | Santos  | Montes Valenzuela | Fisica Basica | ARQ-105 |



## 14. Crear una función que permita obtener el promedio de las edades del género masculino o femenino de los estudiantes inscritos en la asignatura ARQ-104.

La función recibe como parámetro el género y el código de materia.

Realizamos una consulta para poder darnos una idea de la función.

```
select avg(est.edad)
from estudiantes as est
    inner join inscripcion as ins on est.id_est = ins.id_est
    inner join materias as mat on ins.id_mat = mat.id_mat
where est.sexo = 'femenino' and mat.cod_mat = 'ARQ-104';

select avg(est.edad)
from estudiantes as est
    inner join inscripcion as ins on est.id_est = ins.id_est
    inner join materias as mat on ins.id_mat = mat.id_mat
where est.sexo = 'masculino' and mat.cod_mat = 'ARQ-104';
```

| `avg(est.edad)` |         |
|-----------------|---------|
| 1               | 23.0000 |

| `avg(est.edad)` |        |
|-----------------|--------|
| 1               | <null> |

## Realizamos la función

```
create or replace function PromedioEdades(genero varchar(30), codmat varchar(30))  
  returns float  
begin  
  declare resp float ;  
  select avg(est.edad) into resp  
  from estudiantes as est  
    inner join inscripcion as ins on est.id_est = ins.id_est  
    inner join materias as mat on ins.id_mat = mat.id_mat  
  where est.sexo = genero and mat.cod_mat = codmat;  
  return resp;  
end;
```

```
select PromedioEdades( genero: 'femenino', codmat: 'ARQ-104') as Femenino;  
select PromedioEdades( genero: 'masculino', codmat: 'ARQ-104') as Masculino;
```

|   | Femenino ▾ |
|---|------------|
| 1 | 23         |

|   | Masculino ▾ |
|---|-------------|
| 1 | <null>      |

## 15. Crear una función que permita concatenar 3 cadenas.

- La función recibe 3 parámetros.
- Si las cadenas fuesen:
  - Pepito                      ■ Pep                      ■ 50
- La salida debería ser: (Pepito), (Pep), (50)
- La función creada utilizarlo en una consulta SQL. ■ Es decir podría mostrar el nombre, apellidos y la edad de los estudiantes.

```
create or replace function ConcatenarParametros(par1 varchar(100), par2 varchar(100), par3 integer)
returns varchar (200)
begin
    declare resp varchar(200) ;
    select concat('('||par1||'), ('||par2||'), ('||par3||')') into resp;
    return resp;
end;
```

```
select CondatenarParametros( par1: 'Pepito', par2: 'pep', par3: 50);
```

```
`CondatenarParametros('Pepito','pep',50)`  
1 (Pepito),(pep),(50)
```

Ya creada la funcion podemos implementar a nuestra consulta

```
SELECT CondatenarParametros( par1: est.nombres, par2: est.apellidos, par3: est.edad)  
FROM estudiantes as est;
```

```
`CondatenarParametros(est.nombres, est.apellidos, est.edad)`  
1 (Miguel),(Gonzales Veliz),(20)  
2 (Sandra),(Mavir Uria),(25)  
3 (Joel),(Adubiri Mondar),(30)  
4 (Andrea),(Arias Ballesteros),(21)  
5 (Santos),(Montes Valenzuela),(24)
```

## 16. Crear la siguiente VISTA:

- La vista deberá llamarse ARQUITECTURA\_DIA\_LIBRE
- El día viernes tendrán libre los estudiantes de la carrera de ARQUITECTURA debido a su aniversario
  - Este permiso es solo para aquellos estudiantes inscritos en el año 2021.
  - La vista deberá tener los siguientes campos.
    1. Nombres y apellidos concatenados = FULLNAME
    2. La edad del estudiante = EDAD
    3. El año de inscripción = GESTION
    4. Generar una columna de nombre DIA\_LIBRE
      - a. Si tiene libre mostrar LIBRE
      - b. Caso contrario mostrar NO LIBRE

```
create or replace view ARQUITECTURA_DIA_LIBRE as
select concat(est.nombres,est.apellidos) as FULL_NAME,
       est.edad as EDAD,
       ins.gestion as GESTION,
       (
         case
           when ins.gestion = 2021 then 'LIBRE'
           else 'NO LIBRE'
         end
       ) as DIA_LIBRE
from estudiantes as est inner join inscripcion ins on est.id_est = ins.id_est;
```

```
select *
from arquitectura_dia_libre;
```

|   | FULL_NAME               | EDAD | GESTION | DIA_LIBRE |
|---|-------------------------|------|---------|-----------|
| 1 | MiguelGonzales Veliz    | 20   | 2018    | NO LIBRE  |
| 2 | MiguelGonzales Veliz    | 20   | 2018    | NO LIBRE  |
| 3 | SandraMavir Uria        | 25   | 2019    | NO LIBRE  |
| 4 | SandraMavir Uria        | 25   | 2019    | NO LIBRE  |
| 5 | JoelAdubiri Mondar      | 30   | 2020    | NO LIBRE  |
| 6 | JoelAdubiri Mondar      | 30   | 2020    | NO LIBRE  |
| 7 | AndreaArias Ballesteros | 21   | 2021    | LIBRE     |
| 8 | SantosMontes Valenzuela | 24   | 2021    | LIBRE     |

## 17. Crear la siguiente VISTA:

- Agregar una tabla cualquiera al modelo de base de datos.
- Después generar una vista que maneje las 4 tablas
- La vista deberá llamarse PARALELO\_DBA\_I

```
create table docentes (  
    id_doc integer auto_increment primary key not null ,  
    nombre varchar(30) not null ,  
    apellido varchar(30) not null ,  
    correo_electronico varchar(100) not null ,  
    telefono integer not null ,  
    id_mat integer not null ,  
    foreign key (id_mat) references materias(id_mat)  
);
```

```
INSERT INTO docentes (nombre, apellido, correo_electronico, telefono, id_mat)  
VALUES  
    ( 'Juan', 'Pérez', 'jperez@example.com', 12345, 1),  
    ( 'María', 'González', 'mgonzalez@example.com', 98765, 2),  
    ( 'Pedro', 'Rodríguez', 'prodriguez@example.com', 55555, 3),  
    ( 'Ana', 'Sánchez', 'asanchez@example.com', 11124, 4);
```

```

create or replace view PARALELO_DBA_I as
select concat(est.nombres, ' ', est.apellidos) as NOMBRE_ESTUDIANTE,
       concat(doc.nombre, ' ', doc.apellido) as NOMBRE_DOCENTE,
       mat.id_mat as ID_MATERIA,
       mat.nombre_mat as NOMBRE_MATERIA
from estudiantes as est
       inner join inscripcion as ins on est.id_est = ins.id_est
       inner join materias as mat on ins.id_mat = mat.id_mat
       inner join docentes as doc on doc.id_mat = mat.id_mat;

```

```

select *
from PARALELO_DBA_I

```

|   | NOMBRE_ESTUDIANTE        | NOMBRE_DOCENTE  | ID_MATERIA | NOMBRE_MATERIA                  |
|---|--------------------------|-----------------|------------|---------------------------------|
| 1 | Miguel Gonzales Veliz    | Juan Pérez      | 1          | Introduccion a la Arquitectura  |
| 2 | Miguel Gonzales Veliz    | María González  | 2          | Urbanismo y Diseno              |
| 3 | Sandra Mavir Uria        | Ana Sánchez     | 4          | Matematica discreta             |
| 4 | Sandra Mavir Uria        | Pedro Rodríguez | 3          | Dibujo y Pintura Arquitectonico |
| 5 | Joel Adubiri Mondar      | Pedro Rodríguez | 3          | Dibujo y Pintura Arquitectonico |
| 6 | Joel Adubiri Mondar      | Juan Pérez      | 1          | Introduccion a la Arquitectura  |
| 7 | Andrea Arias Ballesteros | Ana Sánchez     | 4          | Matematica discreta             |



GRACIAS POR SU ATENCION!!

SIS9210838

nilberrmayta@gmail.com