

# UNIVERSIDAD PRIVADA FRANZ TAMAYO

## DEFENSA HITO 4 - TAREA FINAL

Estudiante: Sharon Geraldine Limachi Conde

**Asignatura: BASE DE DATOS I**

**Carrera: INGENIERÍA DE SISTEMAS**

**Paralelo: BDA (1)**

**Docente: Lic. William Barra Paredes**

Enlace github <https://github.com/bungoustray>

```

create database hito4;
use hito4;
create table campeonato
(
    id_camp varchar (12) primary key,
    nombrecampeonato varchar (30),
    sede varchar(30)
)
create table equipo
(
    id_eq varchar (12) primary key not null,
    id_camp varchar(12),
    nombre_equipo varchar(30),
    categoria varchar (8),

    foreign key (id_camp) references campeonato(id_camp)
);
create table jugador
(
    id_jug varchar (12)primary key ,
    nombres varchar(30),
    apellidos varchar (50),
    ci varchar(15),
    edad integer,
    id_equipo varchar (12) ,
    foreign key (id_equipo) references equipo(id_eq)
);
insert into campeonato(id_camp,nombrecampeonato,sede)
values ('camp-111','campeonato unifranz','El alto');
insert into campeonato(id_camp,nombrecampeonato,sede)
values ('camp-222','campeonato unifranz','Cochabamba');
-- para equipo
insert into equipo(id_eq, id_camp, nombre_equipo, categoria)
values ('equ-111','camp-111','googlo','varones');
insert into equipo(id_eq, id_camp, nombre_equipo, categoria)
values ('equ-222','camp-111','040 not found','varones');
insert into equipo(id_eq, id_camp, nombre_equipo, categoria)
values ('equ-333','camp-111','girls unifranz','mujeres');

--jugador
insert into jugador(id_jug, nombres, apellidos, ci, edad, id_equipo)
values ('jug_111','carlos','villa','8997811LP',19,'equ-222');
insert into jugador(id_jug, nombres, apellidos, ci, edad, id_equipo)
values ('jug_222','pedro','salas','8997812LP',19,'equ-222');
insert into jugador(id_jug, nombres, apellidos, ci, edad, id_equipo)
values ('jug_333','saul ','araj','8997813LP',19,'equ-222');
insert into jugador(id_jug, nombres, apellidos, ci, edad, id_equipo)
values ('jug_444','sandra','solis','8997814LP',19,'equ-333');
insert into jugador(id_jug, nombres, apellidos, ci, edad, id_equipo)
values ('jug_555','ana','mica','8997815LP',19,'equ-333');

--2 manejo de conceptos
--2.1 mestre un ejemplo de DDL
--es un ddl es el lenguaje de definicion de datos

```

```
--muestra un ejemplo de DML
--un ejemplo aplicado a la base de datos UNIFRANZITOS. el DML es el
lenguaje de manipulación de datos
-- por ejemplo selec
select*
from campeonato as camp;
```

|   | id_camp  | nombrecampeonato    | sede       |
|---|----------|---------------------|------------|
| 1 | camp-111 | campeonato unifranz | El alto    |
| 2 | camp-222 | campeonato unifranz | Cochabamba |

```
--2.3. Para que drive INNER JOIN
-- inner join sirve para te permite consultar datos de dos o más tablas
relacionadas cuando necesitas unir dos consultas
```

```
select ju.nombres, equ.id_camp
from jugador as ju
inner join equipo as equ on equ.id_eq = ju.id_equipo;
```

|   | nombres | id_camp  |
|---|---------|----------|
| 1 | carlos  | camp-111 |
| 2 | pedro   | camp-111 |
| 3 | saul    | camp-111 |
| 4 | sandra  | camp-111 |
| 5 | ana     | camp-111 |

```
--2.4. Defina que es una función de agregación
-- nos permiten obtener medias, los máximos, etc... sobre un conjunto de
valores.
```

```
--2.7. Para qué sirve la función CONCAT en SQL-Server
-- devuelve el número total de filas seleccionadas por la consulta que
elijamos
```

```
--2.8. Muestra un ejemplo del uso de COUNT
```

```
SELECT COUNT(*)
FROM JUGADOR AS JU
```

|   | <anonymous> |
|---|-------------|
| 1 | 5           |

```
--2.9. Muestra un ejemplo del usos de AVG
```

```
SELECT *
FROM jugador
WHERE edad > (SELECT AVG(edad) FROM jugador)
```

| id_jug | nombres | apellidos | ci | edad | id_equipo |
|--------|---------|-----------|----|------|-----------|
|--------|---------|-----------|----|------|-----------|

--2.10. Muestra un ejemplo del uso de MIN-MAX

```
Select min(edad)
from jugador
```

| <anonymous> |    |
|-------------|----|
| 1           | 19 |

```
-- max
select max(edad)
from jugador
```

--3. Manejo de consultas

--3.1. Mostrar que jugadores que formen parte del equipo equ-333

```
select jug.nombres as jugadores
from jugador as jug
inner join EQUIPO as equi on jug.ID_EQUIPO = equi.ID_EQ
WHERE equi.id_eq = 'equip-333';
```

|   | nombres | id_equipo |
|---|---------|-----------|
| 1 | sandra  | equ-333   |
| 2 | ana     | equ-333   |

--3.2. Crear una función que permita saber cuántos jugadores están inscritos.

-- La función debe llamarse Crear una función que permita saber cuántos jugadores están inscritos.

--La función debe llamarse F1\_CantidadJugadores()

```
create or alter function F1_Cantidad_de_Jugadores()
returns integer as
begin
    declare @Cantidad_de_Jugadores integer;

    select @Cantidad_de_Jugadores = Count(jug.nombres)
    from jugador as jug
    return @Cantidad_de_Jugadores
end;
```

```
select dbo.F1_Cantidad_de_Jugadores() as jugadores_inscritos
```

---3.3crear una funcion que permita saber cuantos jugadores estan inscritos.

--la funcion debe llamarse Crear una funcionn que permita saber cuantos jugadores estan inscritos.

--La funcion debe llamarse f1\_CantidadJugadores()()

```

create function F2_Cantidad_de_JugadoresParam(@categoria varchar(30))
returns integer as
begin
    declare @Cantidad_de_Jugadores integer;

    select @Cantidad_de_Jugadores = Count(jug.nombres)
    from jugador as jug
    inner join EQUIPO as equi on jug.ID_EQUIPO = equi.id_eq
    where equi.CATEGORIA = @categoria
    return @Cantidad_de_Jugadores
end;

select dbo.F2_Cantidad_de_JugadoresParam('varones') as
jugadores_inscritos

-- 3.5 Crear una función que permita concatenar 3 parámetros.
-- a) La función debe llamarse F4_ConcatItems()
-- b) La función debe de recibir 3 parámetros.
-- c) La función debe de concatenar los 3 valores.
-- d) Para verificar la correcta creación de la función debe mostrar lo
siguiente.
-- f) Mostrar los nombres de los jugadores, el nombre del equipo y la
sede concatenada, utilizando la función
-- que acaba de crear.
CREATE FUNCTION F4_ConcatItems(
@par1 VARCHAR(30),
@par2 VARCHAR(30),
@par3 VARCHAR(30)
)
RETURNS VARCHAR(30) AS
BEGIN
DECLARE @RESPUESTA VARCHAR(80)
DECLARE @CONCATENADO1 VARCHAR(80)
DECLARE @CONCATENADO2 VARCHAR(80)
DECLARE @CONCATENADO3 VARCHAR(80)
SET @CONCATENADO1 = CONCAT(' NOMBRES DE LOS JUGADORES: ',@PAR1);
SET @CONCATENADO2 = CONCAT(' NOMBRE DE EQUIPO: ',@PAR2);
SET @CONCATENADO3 = CONCAT(' SEDE: ',@PAR3);
SET @RESPUESTA = @CONCATENADO1 + @CONCATENADO2 +@CONCATENADO3

RETURN @RESPUESTA;
END;

--3.6 3.6. Generar la serie fibonacci.
--a) El objetivo es generar una función que retorne una cadena con la
serie de la fibonacci.
--b) La función solo recibe el valor N.
--c) Comportamiento esperado

```

```
return @numbers table (number int)
AS
BEGIN
    Declare @n1 int = 0,@n2 int =1,@i int=0,@temp int
    Insert Into @numbers Values (@n1), (@n2)
    WHILE (@i<=@max-2)
    BEGIN
        Insert Into @numbers Values (@n2+@n1)
        set @temp = @n2
        Set @n2 = @n2 + @n1
        Set @n1 = @temp
        Set @i += 1
    END
RETURN
END
Select * from dbo.Fibonacci(5);
```