



# Tarea HIT04

09.04.20XX

—

Base de Datos II

Unifranz

Sede el Alto

Hito4

## Objetivos de la tarea.

1. Mostrar el manejo de **programación a nivel de Base de Datos**.
2. Mostrar el manejo del **lenguaje procedural**.
3. Mostrar el manejo de **funciones, vistas y triggers en MySQL**.

## Consigna.

Esta tarea se divide en dos partes:

- La primera parte corresponde a la **parte TEÓRICA necesaria**, en donde se encuentra un conglomerado de preguntas relacionadas a BASES DE DATOS RELACIONALES.
- La segunda parte corresponde a la **parte PRÁCTICA necesaria** en donde deberá realizar y crear funciones manejando el motor de base de datos MySQL manejando el concepto del lenguaje procedural.

## Consideraciones sobre la entrega.

La tarea es abierta, usted decide cómo entregar su tarea. Puede agregar imágenes, enlaces, etc. Sin embargo, considerar los siguientes enunciados.

- Deberá generar una presentación hecha en **POWER POINT**. Después este mismo archivo deberá ser convertido a un **archivo PDF**.
- Después de tener disponible la presentación, debe de **GENERAR UN VIDEO** explicando todo su contenido o lo que considere necesario e importante.
  - a. El video debe de tener como mínimo 15 minutos.
  - b. En el video debe estar presente su cámara web.
    - Si no tiene cámara web puede apoyarse en [IRIUM](#) que básicamente convierte su celular en una cámara web.
- Después de tener disponible el video, el video lo puede subir a **youtube, vimeo** o a drive u otra cualquier plataforma.
- Finalmente, todo lo generado, es decir:
  - a. El archivo **ppt** (powerpoint).
  - b. El archivo **pdf** (el powerpoint convertido a PDF).

- c. El **video** subido a alguna plataforma.
- d. IMPORTANTE:
  - Tiene que ser subido a la plataforma **GITHUB**.
  - Para hacer este proceso tiene que tener sus archivos ya disponibles.
  - Crear la carpeta **HITO4** en nuestro repositorio de github.
- En la plataforma **MOODLE** solo deberá de subir una carátula referenciando a su tarea que se encuentra en github.

**Ejemplo:**

<p><b>UNIVERSIDAD PRIVADA FRANZ TAMAYO</b></p> <p><b>DEFENSA HITO 4 - TAREA FINAL</b></p> <p><b>Estudiante:</b> Univ. Nombres Apellidos</p> <p><b>Asignatura:</b> BASE DE DATOS II</p> <p><b>Carrera:</b> INGENIERÍA DE SISTEMAS</p> <p><b>Sede:</b> El Alto</p> <p><b>Paralelo:</b> BDA (1)</p> <p><b>Docente:</b> Lic. William Barra Paredes</p> <p><b>fecha:</b> xx/xx/2022</p> <p><b>GITHUB:</b> <a href="https://github.com/dheevi/base-de-datos-ii">https://github.com/dheevi/base-de-datos-ii</a> (aquí va el enlace a su cuenta de github)</p>
--

---

**A partir de este punto están las preguntas TEÓRICAS y PRÁCTICAS a resolverse.**

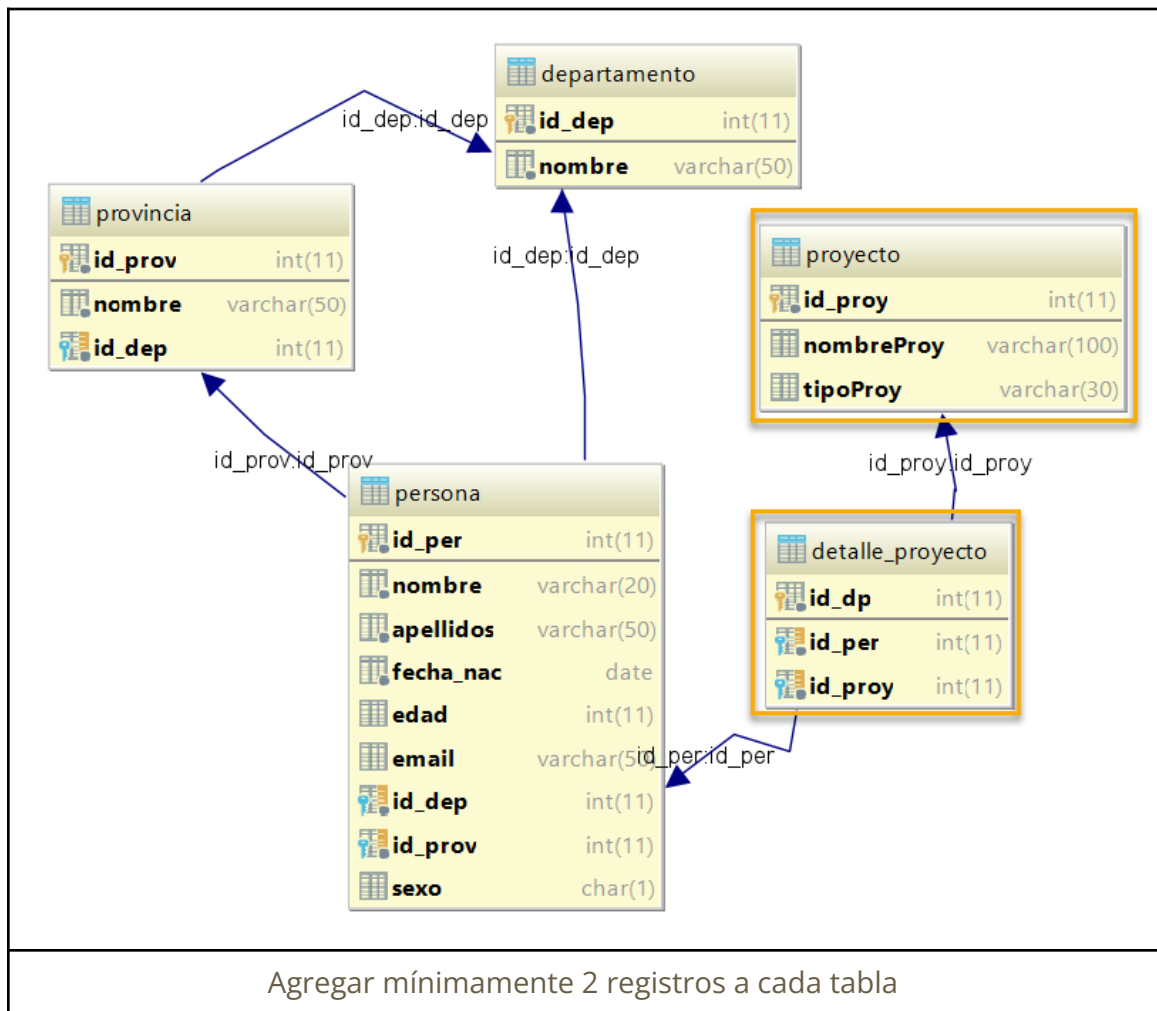
---

## Manejo de conceptos.

1. Defina que es lenguaje procedural en MySQL.
2. Defina que es una **FUCNTION** en MySQL.
- 3.Cuál es la diferencia entre funciones y procedimientos almacenados.
4. Cómo se ejecuta una función y un procedimiento almacenado.
5. Defina que es una **TRIGGER** en MySQL.
6. En un trigger que papel juega las variables **OLD** y **NEW**
7. En un trigger que papel juega los conceptos(cláusulas) **BEFORE** o **AFTER**
8. A que se refiere cuando se habla de eventos en **TRIGGERS**

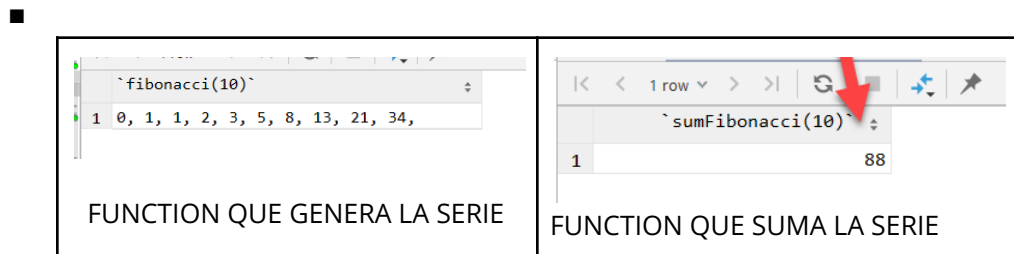
## Parte practica

9. Crear la siguiente Base de datos y sus registros.



## 10. Crear una función que sume los valores de la serie Fibonacci.

- **El objetivo es sumar todos los números de la serie fibonacci desde una cadena.**
- Es decir usted tendrá solo la cadena generada con los primeros N números de la serie fibonacci y a partir de ellos deberá sumar los números de esa serie.
- Ejemplo: **suma\_serie\_fibonacci(mi\_metodo\_que\_retorna\_la\_serie(10))**
  - Note que previamente deberá crear una función que retorne una cadena con la serie fibonacci hasta un cierto valor.
    1. Ejemplo: 0,1,1,2,3,5,8,.....
  - Luego esta función se deberá pasar como parámetro a la función que suma todos los valores de esa serie generada.



- Adjuntar el **código SQL generado y una imagen de su correcto funcionamiento.**

## 11. Manejo de vistas.

- Crear una consulta SQL para lo siguiente.
  - La consulta de la vista debe reflejar como campos:
    1. nombres y apellidos **concatenados**
    2. la edad
    3. fecha de nacimiento.
    4. Nombre del proyecto
- Obtener todas las personas del sexo femenino que hayan nacido en el departamento de El Alto en donde la fecha de nacimiento sea:
  1. fecha\_nac = '2000-10-10'

**LA CONSULTA GENERADA PREVIAMENTE CONVERTIR EN UNA VISTA**

## 12. Manejo de TRIGGERS I.

- Crear TRIGGERS Before or After para INSERT y UPDATE aplicado a la tabla PROYECTO
  - Debera de crear 2 triggers minimamente.
- Agregar un nuevo campo a la tabla PROYECTO.
  - El campo debe llamarse **ESTADO**

- Actualmente solo se tiene habilitados ciertos tipos de proyectos.
  - EDUCACION, FORESTACION y CULTURA
- Si al hacer insert o update en el campo **tipoProy** llega los valores EDUCACION, FORESTACIÓN o CULTURA, en el campo ESTADO colocar el valor **ACTIVO**. Sin embargo se llegat un tipo de proyecto distinto colocar **INACTIVO**
- Adjuntar el **código SQL generado y una imagen de su correcto funcionamiento**.

### 13. Manejo de Triggers II.

- El trigger debe de llamarse **calculaEdad**.
- El evento debe de ejecutarse en un **BEFORE INSERT**.
- Cada vez que se inserta un registro en la tabla **PERSONA**, el trigger debe de calcular la edad en función a la fecha de nacimiento.
- Adjuntar el **código SQL generado y una imagen de su correcto funcionamiento**.

### 14. Manejo de TRIGGERS III.

- Crear otra tabla con los mismos campos de la tabla persona(Excepto el primary key **id\_per**).
  - No es necesario que tenga **PRIMARY KEY**.
- Cada vez que se haga un **INSERT** a la tabla persona estos mismos valores deben insertarse a la tabla copia.
- Para resolver esto deberá de crear un **trigger before insert para la tabla PERSONA**.
- Adjuntar el **código SQL generado y una imagen de su correcto funcionamiento**.

### 15. Crear una consulta SQL que haga uso de todas las tablas.

- La consulta generada convertirlo a VISTA