

---

# DEFENSA HITO 4 - TAREA FINAL

---

Nombre

Kevin German Paucara Oro

Base de Datos - Ingenieria de sistemas

---

```
CREATE DATABASE Agencia de viajes
```

```
USE Agencia de viajes
```

```
CREATE TABLE Clientes (
```

```
    Cliente_ID INTEGER PRIMARY KEY NOT NULL,
```

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

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

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

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

```
);
```

```
CREATE TABLE Destinos (
```

```
    Destino_ID INTEGER PRIMARY KEY NOT NULL,
```

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

```
    Descripcion INTEGER NOT NULL,
```

```
    PrecioBase INTEGER NOT NULL,
```

```
);
```

```
CREATE TABLE Paquetes (
```

```
    Paquete_ID INTEGER PRIMARY KEY NOT NULL,
```

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

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

```
    fechainicio DATE NOT NULL,
```

```
    fechafin DATE NOT NULL,
```

```
    destino_ID INTEGER NOT NULL,
```

```
    preciototal INTEGER NOT NULL,
```

```
    FOREIGN KEY (destino_ID) REFERENCES Destinos(Destino_ID)
```

```
);
```

```
CREATE TABLE Empleados (
```

```
    Empleado_ID INTEGER PRIMARY KEY NOT NULL,
```

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

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

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

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

```
);
```

```
CREATE TABLE Reservas (
```

```
    Reserva_ID INTEGER PRIMARY KEY NOT NULL,
```

```
    Cliente_ID INTEGER NOT NULL,
```

```
    Paquete_ID INTEGER NOT NULL,
```

```
    Fechareserva DATE NOT NULL,
```

```
    Cantidadpersonas INTEGER NOT NULL,
```

```
    Preciototal INTEGER NOT NULL,
```

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

```
    FOREIGN KEY (Cliente_ID) REFERENCES Clientes(Cliente_ID),
```

```
    FOREIGN KEY (Paquete_ID) REFERENCES Paquetes(Paquete_ID)
```

```
);
```

1.2. Los registros de cada tabla deberían quedar de la siguiente forma

```
INSERT INTO Clientes (Cliente_ID, Nombre, Apellido, Email, Telefono)
VALUES
```

```
(1, 'John', 'Doe', 'john.doe@example.com', '555-1234'),
```

```
(2, 'Jane', 'Smith', 'jane.smith@example.com', '555-5678'),
```

```
(3, 'Bob', 'Jones', 'bob.jones@example.com', '555-9012'),
```

```
(4, 'Alice', 'Lee', 'alice.lee@example.com', '555-3456'),
```

```
(5, 'Charlie', 'Brown', 'charlie.brown@example.com', '555-7890');
```

```
INSERT INTO Destinos (Destino_ID, Nombredestino, Descripcion, PrecioBase)
VALUES
```

```
(1, 'París', 'La ciudad del amor y la luz', 1000.00),
(2, 'Londres', 'La capital de Inglaterra y el Reino Unido', 800.00),
(3, 'Sydney', 'La ciudad más grande y cosmopolita de Australia', 1500.00);
```

```
INSERT INTO Empleados (Empleado_ID, Nombreempleado, Apellidoempleado, EmailEmpleado, Cargo)
VALUES
```

```
(1, 'Felipe', 'González', 'felipe.gonzalez@correo.com', 'Agente de Viajes'),
(2, 'María', 'Pérez', 'maria.perez@correo.com', 'Gerente'),
(3, 'Juan', 'López', 'juan.lopez@correo.com', 'Asistente');
```

```
INSERT INTO Paquetes (Paquete_ID, nombrepaquete, descripcion, fechainicio, fechafin, destino_ID, preciototal, Estado)
VALUES
```

```
(1, 'Paquete a París', 'Explora la ciudad del amor', '2023-01-01', '2023-01-15', 1, 1500.00, 'Confirmado'),
(2, 'Tour por Londres', 'Descubre los encantos de la capital británica', '2023-02-01', '2023-02-10', 2, 1200.00, 'Pendiente'),
(3, 'Aventura en Sydney', 'Disfruta la cosmopolita ciudad australiana', '2023-03-01', '2023-03-20', 3, 1800.00, 'Cancelado');
```

```
INSERT INTO Reservas (Reserva_ID, Cliente_ID, Paquete_ID, Fechareserva, Cantidadpersonas, Preciototal, Estadoreserva)
VALUES
```

```
(1, 1, 1, '2022-12-01', 2, 3000.00, 'Confirmada'),
(2, 2, 2, '2022-12-15', 1, 1200.00, 'Pendiente'),
(3, 3, 3, '2023-01-01', 4, 7200.00, 'Cancelada');
```

## 2. Manejo de conceptos

### 2.1. Muestra un ejemplo de DDL.

Es un lenguaje de definición de datos esta se utiliza para definir la estructura de la base de datos. Define base de datos y sus objetos, como tablas, índices y procedimientos, también se utiliza para crear y eliminar objetos de base de datos, algunos ejemplos son el DROP, TRUNCATE.

### 2.2. Muestra un ejemplo de DML

Se utiliza para manipular los datos de una base de datos

Algunos ejemplos son:

SELECT: Donde nos sirve para recuperar datos de una o varias tablas como: `SELECT*FROM clientes`

INSERT: Esta se utiliza para insertar nuevos datos en una tabla Ejemplo: `INSERT INTO customers (id, name, address) VALUES (1, 'John Smith,' '123 Main St');`

### 2.3. Para que sirve INNER JOIN.

Esta nos sirve para combinar registros de dos o mas tablas, osea donde hay valores que coincidan en un campo comun

### 2.4. Defina que es una función de agregación

Es una herramienta que te permite hacer cálculos en un conjunto de datos, como sumar valores, encontrar promedios o contar elementos.

### 2.5. Liste funciones de agregación que conozca.

COUNT : Que cuenta el numero de filas

SUM: Esta suma valores

MIN: Esta nos ayuda a devolver el valor minimo

MAX: .Esta nos ayuda a devolver el valor máximo

AVG: Esta Calcula el promedio

2.6. Para qué sirve la función CONCAT en SQL-Server

Esta nos sirve para concatenar dos o más cadenas de texto en una sola cadena, ejemplo:

```
SELECT CONCAT(Nombre, ' ', Apellido) AS NombreCompleto
FROM Clientes;
```

Donde nos devolvera el nombre completo combinando las columnas Nombre y Apellido en la tabla Cliente

2.7. Muestra un ejemplo del uso de COUNT

```
SELECT COUNT(*) AS TotalClientes
FROM Clientes;
```

En esta consulta devolverá el número total de filas en la tabla Cliente que representa el total de clientes registrados

2.8. Muestra un ejemplo del usos de AVG

```
SELECT AVG(preciototal) AS PromedioPrecio
FROM Paquetes;
```

Al ejecutar esta consulta devolverá el promedio de los precios totales de los paquetes turísticos registrados en la tabla Paquetes

2.9. Muestra un ejemplo del uso de MIN-MAX

```
SELECT MIN(preciototal) AS PrecioMinimo, MAX(preciototal) AS PrecioMaximo
FROM Paquetes;
```

Al ejecutar esta consulta devolverá el precio mínimo y máximo de los paquetes turísticos registrados en la respectiva tabla de Paquetes

3. Manejo de consultas

3.1. ¿Cuáles son los empleados que tienen el título "Agente de Ventas"?

```
SELECT Nombreempleado, Apellidoempleado
FROM Empleados
WHERE Cargo = 'Agente de Ventas';
```

La consulta tendría que busca en la tabla de empleados aquellos que tienen el título de Agente de Ventas y muestra sus nombres y apellidos.

3.2. ¿Cuáles son los destinos de los paquetes turísticos reservados por el cliente con ID 1?

```
SELECT d.Nombredestino
FROM Reservas r
JOIN Destinos d ON r.idDestino = d.Destino_ID
```

WHERE r.idCliente = 1;

En aquí se selecciona el nombre del destino de las reservas hechas por el cliente con ID 1, relacionando la tabla de reservas con la tabla de destinos

3.3. ¿Cuáles son los paquetes turísticos reservados por el cliente llamado "Jane Smith"?

```
SELECT p.nombrepaquete
FROM Paquetes p
JOIN Reservas r ON p.paquete_ID = r.idReserva
JOIN Clientes c ON r.idCliente = c.Cliente_ID
WHERE c.Nombre = 'Jane' AND c.Apellido = 'Smith';
```

En esta consulta obtenemos los nombres de los paquetes reservados por el cliente llamado Jane Smith cruzando la información de clientes, reservas y paquetes

3.4. ¿Cuáles son los nombres de los clientes, los destinos de sus paquetes turísticos y los nombres de los empleados que gestionaron esas reservas?

```
SELECT c.Nombre, c.Apellido, d.Nombredestino, e.Nombreempleado, e.Apellidoempleado
FROM Reservas r
JOIN Clientes c ON r.idCliente = c.Cliente_ID
JOIN Paquetes p ON r.idReserva = p.paquete_ID
JOIN Destinos d ON p.destino_ID = d.Destino_ID
JOIN Empleados e ON p.idEmpleado = e.Empleado_ID;
```

En aquí se muestra el nombre y apellido del cliente, el destino del paquete reservado y los nombres de los empleados que gestionaron esas reservas, uniendo todas las tablas necesarias mediante sus claves.

N	A	D	N.E.	A.E.
John	Doe	París	Felipe	González
Jane	Smith	Londres	María	Pérez
Bob	Jones	Sydney	Juan	López

3.5. ¿Cuál es la cantidad total de personas que han reservado paquetes turísticos en la agencia?

```
SELECT SUM(Cantidadpersonas) AS TotalPersonas
FROM Reservas;
```

Aquí se calcula la suma total de personas que han realizado reservas, obteniendo la suma de la columna Cantidadpersonas de la tabla de Reservas.

3.6. ¿Cuántas reservas se han realizado para el paquete turístico llamado "Aventura en Tokio"?

```
SELECT COUNT(*) AS TotalReservas
FROM Paquetes p
JOIN Reservas r ON p.paquete_ID = r.idReserva
WHERE p.nombrepaquete = 'Aventura en Tokio';
```

En esta consulta cuenta cuántas reservas se han hecho para el paquete turístico llamado "Aventura en Tokio", vinculando la tabla de paquetes con la tabla de reservas.

# GRACIAS POR SU ATENCIÓN.



**INNOVACIÓN  
EN EDUCACIÓN**