

Proyecto creación bases de datos para Tadeo Tours.

Alaix Andrés

Correa Adriana

Universidad de Bogotá Jorge Tadeo Lozano.

Bases de Datos

Jun Gabriel Villamil

Mayo 2024

## 1. INTRODUCCIÓN:

### **Contexto del negocio:**

Tadeo Tours es una empresa dedicada al turismo, y sus servicios son la gestión con hoteles y aerolíneas. Sus servicios pueden ser tomados dentro del territorio nacional o fuera de él.

### **Antecedentes:**

Tadeo tours actualmente hace sus procesos de reserva en un formato preestablecido. Ahora quiere implementar una base de datos unificada para ofrecer mejores servicios a sus clientes.

### **Justificación:**

Este proyecto es importante ya que contribuye al crecimiento de la empresa, esto debido a que en los últimos tiempos ofrece servicios a más personas por fuera de la universidad.

Adicional a lo anterior, también automatiza procesos y los hace más rápidos y eficientes para los colaboradores.

## 2. Generalidades del proyecto:

Este proyecto se centrará en dos tablas principales: la de clientes y la de reservas. Estas dos tablas son fundamentales debido a su relación con las otras tablas.

Se contemplan 13 tablas: cliente, contacto cliente, viajeros, reservas, pagos, reservas hotel, reservas avión, vuelos, aerolínea, hotel cancelaciones, cambios, fecha hora vuelo.

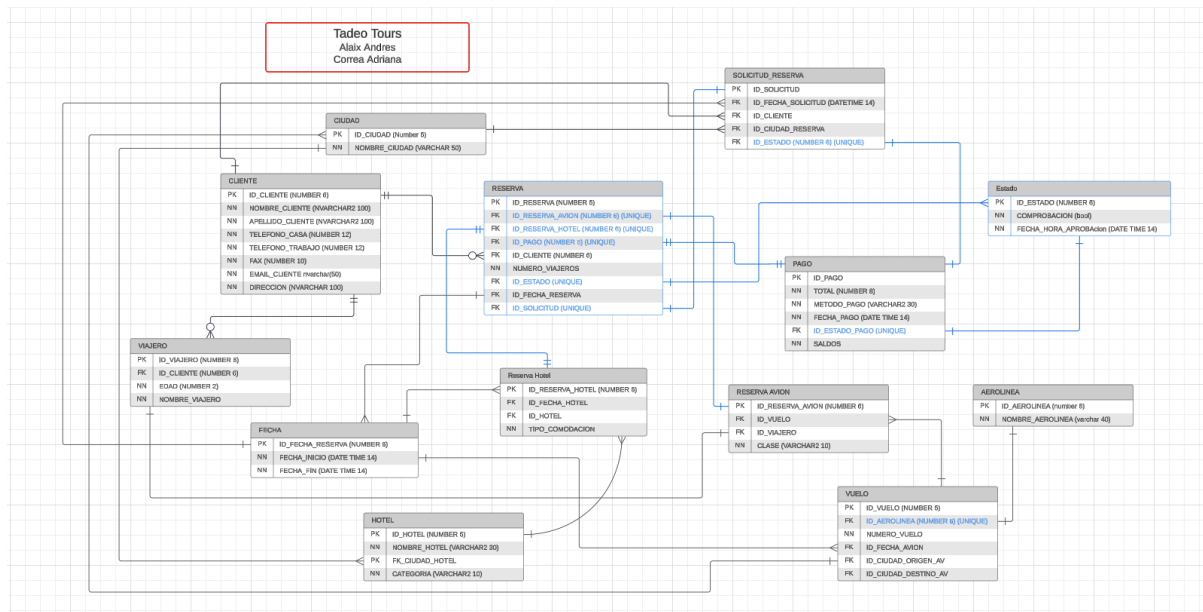
La idea es que cuando un cliente solicite una reserva, tengamos su información de contacto y generemos un ID único para ese cliente. A partir de las preferencias del cliente, se generará la reserva tanto en la tabla general como en las tablas específicas de reserva de hotel y de avión. Posteriormente, una vez creada la reserva, se completará la información del hotel, del avión, de la aerolínea y de la fecha y hora del vuelo.

Si el cliente desea realizar una cancelación o un cambio, se consultará la tabla de fecha y hora del vuelo para determinar la viabilidad de la modificación.

Además, contamos con una tabla de viajeros donde se solicita la edad para determinar si se debe cobrar el pasaje de avión. También se dispone de una columna para asignar un ID a cada viajero, lo que facilita el seguimiento y registro preciso de la cantidad de viajeros.

### 3. MODELO ENTIDAD RELACIÓN:

a.



Enlace de acceso: [https://lucid.app/lucidchart/a89819a2-194d-4f92-90d4-b1585deb0d8d/edit?viewport\\_loc=-711%2C-943%2C2725%2C1322%2C0\\_0&invitationId=inv\\_79dfa6bb-16ef-459f-9456-858e0ef0a976](https://lucid.app/lucidchart/a89819a2-194d-4f92-90d4-b1585deb0d8d/edit?viewport_loc=-711%2C-943%2C2725%2C1322%2C0_0&invitationId=inv_79dfa6bb-16ef-459f-9456-858e0ef0a976)

#### Explicación del modelo:

Tomaremos la tabla de cliente como tabla inicial; para explicar su proceso de creación. Este enfoque lógico es aplicable a todas las demás tablas. Inicialmente creamos dos tablas: una llamada "clientes" y otra llamada "información cliente".

En la tabla de "clientes", teníamos un campo de identificación de cliente (ID cliente) y una relación con la clave foránea de reserva.

Por otro lado, en la tabla "información cliente", se almacenaba la información de contacto y dirección del cliente. En esta tabla, se establecía nuevamente una relación con el campo de ID cliente.

Esto está mal por principio de normalización, ya que esto es redundante, por eso, unimos estas dos tablas en una sola.

Otro de los retos al momento de crear el modelo fue la creación de la tabla reservas, ya que en un inicio teníamos una tabla con toda la información de las reservas, tanto de del avión como de hotel. Por eso decidimos crear otras dos tablas (adicional a la de reservas), una para la reserva de hotel y otra para la reserva de avión, aquí se almacena información específica de la reserva (las categorías tanto de avión como de hotel). Como ya dijimos anteriormente la principal tabla es la de reservas; aquí también hay una tabla llamada estado, que nos dice si el

pago de la reserva está aprobado o no. Esta última tabla (estado), se relaciona con la tabla de pago, y ahí tenemos información del medio de pago y el valor del pago.

En general, para todo el modelo, seguimos este principio: los datos de una tabla no deben repetirse en otra tabla. Además, si es necesario, creamos tablas adicionales para almacenar información relacionada con una tabla principal.

- b. Diccionario de datos:** La información del resto de las tablas se encuentra dentro del enlace

Nombre: CLIENTE

Fecha:02-04-24

Descripción: tabla con la información del cliente y su id dentro de la tabla

	nombre de la columna	tipo de dato	tamaño máximo	s de integridad	descripción
	id_cliente	Number	6	PK	cedula de ciudadanía del cliente que contrata nuestros servicios
	nombre_cliente	varchar	100	NN	nombre del cliente que contrata nuestros servicios
	apellido_cliente				
	dirección	varchar	100	FK	direccion del cliente que contrata nuestros servicios
	telefono_casa	Number	12	NN	telefono de la casa del cliente
	telefono_trabajo	Number	12	NN	telefono del trabajo del cliente
	fax	Number	10	NN	fax del cliente
	email	varchar	50	NN	correo del cliente

Enlace de acceso: [BD diccionario Tours.xlsx](#)

#### 4. CATALOGACIÓN DE SCRIPTS:

NOMBRE SCRIPT	DESCRIPCIÓN	FECHA	VERSIÓN
Create tables	permite la creación de tablas necesarisa pra resolver las preguntas del proyecto	11/5/2024	2
Insert into	permite la insecion de datos dentro de las tablas	2/4/2024	1
alter table	permite realizar cambios en la estructura. Podemos eliminar, agregar o modificar columnas	10/5/2024	1
Create Sequence	es una sentencia SQL que se utiliza para crear un objeto de secuencia en una base de datos relacional. Una secuencia es un objeto que genera números secuenciales automáticamente en orden ascendente o descendente	2/4/2024	1
Constraints	las "constraints" (restricciones) son reglas que se aplican a las columnas de una tabla para garantizar la integridad de los datos almacenados en la base de datos. Estas restricciones se utilizan para imponer reglas o condiciones específicas sobre los datos que pueden ser insertados, actualizados o eliminados en una tabla	12/5/2024	1

Enlace de acceso: [CATALOGACION tadeo tours](#)

#### 5. GUIA DE INSTALACIÓN:

En el siguiente enlace se encontrarán el repositorio de GitHub, con archivos requeridos y mencionados en esta guía de instalación

Repositorio de GitHub de Tadeo Tours: <https://github.com/alaixgg/Tadeo-Tours-Ferreteria/tree/main/Tadeo%20Tours>

Pasos para la instalar la base de datos:

1. Cree una nueva conexión a base de datos, esta llámela db\_Tadeo\_Tours

2. Ejecute el archivo "1. **CREATE TABLE.sql**" (<https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/1.%20CREATE%20TABLE.sql> )

En este archivo, se definen las estructuras de las tablas que compondrán tu base de datos. Cada tabla tiene un conjunto de columnas con sus tipos de datos y restricciones.

Por ejemplo, la tabla CIUDAD tiene tres columnas: id\_ciudad, nombre\_ciudad, y id\_hotel. La columna id\_ciudad es un número que actúa como clave primaria, es decir, cada fila en esta tabla tiene un identificador único de ciudad. La columna nombre\_ciudad almacena el nombre de la ciudad, y id\_hotel se usa para relacionar las ciudades con los hoteles que tienen en ellas.

Cada tabla tiene su propia estructura y restricciones definidas, como claves primarias y restricciones de no nulo.

3. Ejecute el archivo "2. **Alter Table (constraints FK).sql**" ([https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/2.%20Alter%20Table%20\(constraints%20FK\).sql](https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/2.%20Alter%20Table%20(constraints%20FK).sql))

En este archivo, se añaden restricciones de clave externa (foreign key constraints, FK) a las tablas ya creadas. Estas restricciones definen relaciones entre las tablas y aseguran la integridad referencial de la base de datos.

4. Ejecute el archivo "3. **Index.sql**" (<https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/3.%20index.sql>)

Los índices se utilizan para mejorar el rendimiento de las consultas en la base de datos.

Por ejemplo, el índice idx\_nombre\_ciudad se crea en la tabla CIUDAD en la columna nombre\_ciudad. Lo que consigue que cuando realices consultas que buscan por nombre de ciudad, la base de datos puede utilizar este índice para encontrar las filas más rápidamente.

5. Ejecute el archivo "4. **SEQUENCE.sql**" (<https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/4.%20secuencias.sql>)

Las secuencias se utilizan para generar valores únicos de forma automática, como claves primarias o valores de identificación.

Por ejemplo, la secuencia seq\_id\_ciudad se utiliza para generar valores únicos para la columna id\_ciudad en la tabla CIUDAD. Cada vez que se inserta una nueva ciudad, la secuencia proporcionará un nuevo número de identificación único para esa ciudad.

6. Ejecute el archivo "5. **Inserccion de datos .sql**" (<https://github.com/alaixgg/Tadeo-Tours->

[Ferreteria/blob/main/Tadeo%20Tours/5.%20Inserccion%20de%20datos%20.sql](https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/5.%20Inserccion%20de%20datos%20.sql))

#### 6. Consultas:

Enlace de las consultas, repositorio de GitHub: <https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/6.%20Consultas%20.sql>

#### 7. Creación de paquetes

Enlace del script para la creación de paquetes, repositorio de GitHub:

<https://github.com/alaixgg/Tadeo-Tours-Ferreteria/blob/main/Tadeo%20Tours/7.%20Paquetes.sql>

#### 8. Conclusiones:

Este proyecto como parte de un proceso pedagógico da relevancia a la prueba de nuestros conocimientos aplicados.

El poder hacer todo el proceso que haría un arquitecto de datos da importancia al proceso educativo.