Informe técnico

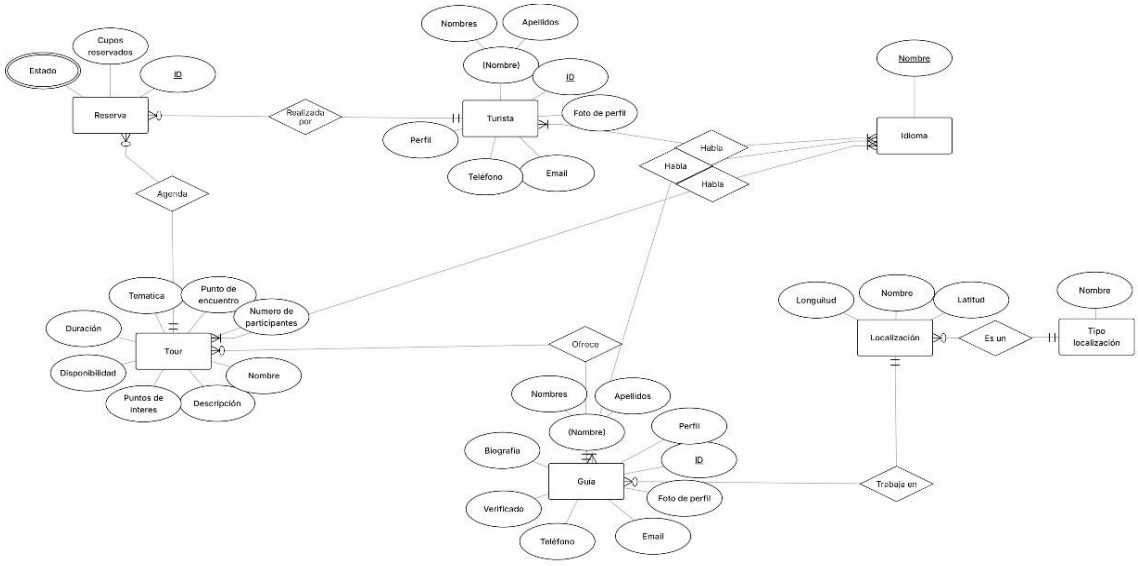
Primera iteración de la creación de la base de datos.

1. Introducción.

El presente documento tiene como objetivo explicar, de manera resumida, el proceso de diseño que se siguió en la primera versión de la base de datos de la empresa ficticia Free-Tour partiendo desde el diseño conceptual hasta el físico y detallando decisiones que se tomaron en el paso de un diseño a otro. También se explicarán y justificarán decisiones de implementación como los tipos de datos usados en el motor Microsoft SQL Server, los tamaños asociados a cada atributo (cuando corresponda), el uso de ciertas expresiones regulares y el manejo de llaves principales y foráneas.

Debido a la extensión del proyecto este documento no podrá cubrir cada singularidad del diseño, sin embargo, buscará hacer del conocimiento del lector gran parte de este para facilitar su comprensión y que pueda familiarizarse con él. En general el proyecto tiene 3 aspectos principales: el diagrama entidad-relación que representa el modelo conceptual, el diagrama de tablas que representa el diseño lógico y el script de creación de la base de datos escrito con notación propia de del motor de base de datos Microsoft SQL Server.

2. Modelo conceptual:



En esta primera iteración del modelo conceptual se realizó la implementación de las 3 primeras historias de usuario recomendadas en la guía de dominio e información sobre la base de datos, donde se propuso la creación de las entidades “guía”, “tour” y “reserva” las cuales están planteadas en el diagrama, pero realizando un análisis se logró identificar la creación de las historias de usuario #6, #7, #8 y #9 que dicen lo siguiente:

Historia de usuario 6

Como director de la plataforma **quiero** conocer quiénes son los turistas (usuarios) que utilizan mis servicios **para tener** una trazabilidad y dominio de quienes realizan las reservaciones, opiniones, calificaciones, etc.

Descripción

Los administradores serán capaces de poder registrar los datos esenciales de los turistas, como sus nombres, su número telefónico, email, fecha de nacimiento, su identificación, el o los idiomas que habla, su foto de perfil y su perfil asociado dentro de la plataforma.

Criterios de aceptación

1. El número de teléfono deber ser único y de acuerdo con los estándares de Colombia o estándares internacionales en el caso de que el turista no sea de Colombia.
2. La foto de perfil debe ser obligatoria porque permite mostrar humanidad y calidad para el guía.
3. Su perfil debe ser único dentro de la plataforma.
4. El o los idiomas debe ser un valor dentro de los ya registrados.
5. La fecha de nacimiento no debe ser mayor a la fecha actual.

Historia de usuario 7

Como administrador de la plataforma **quiero** registrar todos los idiomas presentes en: el turista, el guía y el tour **para que** cada parte (tanto el turista como el guía) se desempeñen en el lenguaje que manejan.

Descripción

La funcionalidad permitirá gestionar un catálogo de idiomas en la plataforma. Estos idiomas estarán disponibles para ser asociados a los turistas (idiomas que hablan), a los guías (idiomas que dominan) y a los tours (idiomas en los que se ofrecen). De esta manera se garantiza que la comunicación entre los actores sea clara y que el turista pueda seleccionar tours con guías que hablen un idioma en común.

Criterios de aceptación

1. El nombre deber uno real dentro de los idiomas registrados en el mundo
2. Un tour puede ser hablado en varios idiomas
3. Un turista puede hablas varios idiomas
4. Un guía puede hablar varios idiomas

Historia de usuario 8

Como administrador **quiero** tener la localización de nuestros tours ofrecidos ya sea en un municipio o en una ciudad de Colombia **para que** los turistas sean capaces de encontrar tours en sus destinos favoritos y que los guías de dichas zonas se promuevan más en el negocio.

Descripción

Se debe poner ingresar una nueva localización con el nombre de esta dentro de Colombia, el tipo de localización (municipio/ciudad), además de su latitud y longitud para mejorar la precisión y reducir las ambigüedades entre los usuarios a la hora de buscar el destino al cual van a asistir.

Criterios de aceptación

1. El nombre de la localización debe corresponder a un municipio o ciudad válida de Colombia.
2. El tipo de localización debe seleccionarse únicamente entre las opciones disponibles.
3. La localización debe contar con coordenadas de latitud y longitud válidas.
4. No se permite registrar dos veces la misma localización.
5. Un tour debe poder asociarse a una y solo una localización registrada.

Historia de usuario 9
Como usuario **quiero** saber qué tipo de localización estoy visitando, si es una ciudad o un municipio de Colombia **para poder** tener una mayor proyección y conocimiento de donde viajaré
Descripción
Se podrá clasificar a cada localización registrada con su tipo de localización dentro de los dos valores validos (municipio o ciudad), sabiendo que una localización no puede ser los dos tipos al mismo tiempo.
Criterios de aceptación
1. Los únicos tipos disponibles son: Municipio o ciudad 2. Cada localización se clasifica en uno y solo un tipo

3. Catálogo de relaciones y generalizaciones/especializaciones:

3.1 RESERVA-TOURISTA

En este caso tenemos una relación (N:1) con la conjunción “realizado por”, o sea: “una reserva es realizada por uno y solo un turista y cada turista puede hacer 0 o varias reservas”

3.2 RESERVA-TOUR

En este caso teneos una relación (N:1) con la conjunción “agenda”, o sea: “una reserva agenda a uno y solo un tour y cada tour es agendado por 0 o varias reservas”

3.3 TURISTA-IDIOMA

En este caso tenemos una relación (M:N) con la conjunción “habla”, o sea: “un turista habla uno o muchos idiomas y un idioma es hablado por uno o muchos turistas”

3.4 TOUR-IDIOMA

En este caso tenemos una relación (M:N) con la conjunción “habla” ”, o sea: “en un tour se habla uno o muchos idiomas y un idioma es hablado en uno o muchos tours”

3.5 GUÍA-IDIOMA

En este caso tenemos una relación (M:N) con la conjunción “habla”, o sea: “un guía habla uno o muchos idiomas y un idioma es hablado por uno o muchos guías”

3.6 GUÍA-LOCALIZACIÓN

En este caso tenemos una relación (N:1) con la conjunción “trabaja en”, o sea: “un guía trabaja en una y solo una localización y en una localización trabaja 0 o muchos guías”

3.7 LOCALIZACIÓN-TIPO LOCALIZACIÓN

En este caso tenemos una relación (N:1) con la conjunción “es un”, o sea: “una localización es un y solo un tipo de localización y un tipo de localización clasifica a 0 o varias localizaciones”

4. Matriz de trazabilidad y entidades:

Historia de Usuario	Entidades Relacionadas	Relaciones
H1: Registrar guía local	Guía	Guía–Idioma Guía–Idioma
H2: Registrar tours	Tours, temática	Tours-Temática **(por implementar)**
H3: Registrar reservación	Reservación, tour, guía	Reservación-Tour Reservación-Guía
H6: Registrar turistas	Turista	Guía–Idioma
H7: Registrar idiomas	Idioma, Turista, Guía, Tour	Turista–Idioma. Guía–Idioma. Tour–Idioma.
H8: Registrar localización	Localización, Tour	Tour–Localización
H9: Identificar tipo de localización	Localización, Tipo de localización	Localización–Tipo de localización

4.1 Entidades:

- Reserva:

Atributo	Definición (significado)	Dominio de valores	Ejemplos	Reglas de validación
CUPOS RESERVADOS	Denomina el total de Cupos Reservados en cantidad	ENTERO POSITIVO	4,5,6,8,10,12	No nulo, No se permiten valores menores o iguales a cero.
ID	IDENTIFICACION DE LA RESERVA, este valor será un entero Único	ENTERO POSITIVO	1,2,65,78798,9965265,564	único, No se permite valores menores o iguales a cero, No nulo.
Estado	Reconoce el estado del Tour, este valor cambia en relación a la circunstancia en la que se encuentre.	CADENA DE TEXTO (MAX 100 CHAR)	"Cancelada";"Confirmada","etc"	La cadena de texto <=100; No nulo, Puede cambiar, No se permiten solamente espacios, Todo en Mayúscula
Turista (relación)	Identificador único del Turista, realiza la reserva. Permite asociar, esta entidad con un turista existente.	ENTERO POSITIVO EXISTENTE EN LA ENTIDAD Turista	1,2,3,4,5,6,43,	No nulo, No se permiten valores menores, iguales a cero o que no exista en la entidad turista.
Tour (relación)	Identificador único del Tour, Donde la reserva se está dando. Permite asociar, esta entidad con un tour existente.	ENTERO POSITIVO EXISTENTE EN LA ENTIDAD Tour	1,2,3,4,5,6,43,	No nulo, No se permiten valores menores, iguales a cero o que no exista en la entidad Tour.

- Tour:

Atributo	Definición (significado)	Dominio de valores	Ejemplos	Reglas de validación
Nombre del Tour	identifica el nombre del Recorrido que se dará	CADENA DE TEXTO (MAX 100 CHAR)	"Recorrido al Putumayo", "Tour de Sevilla"	La cadena de texto <=100; No nulo, Puede cambiar, No se permiten solamente espacios, Todo en Mayúscula
Temática	Menciona las temáticas que tendrá el recorrido	CADENA DE TEXTO (MAX 100 CHAR)	"Prueba de Valor", "Locura extrema"	La cadena de texto <=100; No nulo, Puede cambiar, No se permiten solamente espacios, Todo en Mayúscula
Punto de Encuentro	Nos habla sobre el punto donde el guía se encontrará con el Turista	CADENA DE TEXTO (MAX 100 CHAR)	"Popayán Mercado", "Rio negro", "Rio blanco"	La cadena de texto <=100; No nulo, Puede cambiar, No se permiten solamente espacios, Todo en Mayúscula
Número de Participantes	Nos habla sobre el número total de Participantes que se pueden dar en el Tour	Entero positivo	1,2,3,4,5,6,43,	No nulo, No se permiten valores menores o iguales a cero. Debe de ser un número entero
Duración	Tiempo total estimado que dura el tour desde su inicio hasta su fin	Número Real (enteros o decimales) expresado en unidades de tiempo específicamente: HORA	2, 4.5, 72	No nulo, No se permiten valores menores o iguales a cero, Debe de ser un número Real, menor a 720.
Disponibilidad	Nos habla sobre los distintos horarios que están disponible para el inicio del Tour	CADENA DE TEXTO (MAX 100 CHAR)	"2025/11/25", "2025/12/30"	La cadena de texto <=100; No nulo, Puede cambiar, No se permiten solamente espacios, Todo en Mayúscula
Puntos de Interés	Nos mencionan sobre los distintos puntos de intereses que vamos a conocer en el tour	CADENA DE TEXTO (MAX 100 CHAR)	"T1 Sesión", "Bocagrande"	La cadena de texto <=100; No nulo, Puede cambiar, No se permiten solamente espacios, Todo en Mayúscula
Descripción	Habla sobre el recorrido, lo que se hará, y el objetivo central.	CADENA DE TEXTO (MAX 150 CHAR)	"Este recorrido te ayudara a relajarte y sentirte renovado de nuevo"	La cadena de texto <=150; No nulo, Puede cambiar, No se permiten solamente espacios, Todo en Mayúscula

- Turista:

Atributo	Definición (significado)	Dominio de valores	Ejemplos	Reglas de validación
ID	Identificador único de cada Turista que hace referencia al # identificación (cedula, T.I, etc.)	Entero positivo	1657567, 2556757, 10356757	No nulo, único, su longitud debe de ser de al menos 7 dígitos y máximo 18
Nombres	Nombre(s) propios del Turista.	Texto (máx. 50 caracteres)	"Juan", "María Fernanda"	No nulo, solo letras y espacios, sin números. Sin solo espacios
Apellidos	Apellido(s) del Turista.	Texto (máx. 50 caracteres)	"Pérez", "González Ramírez"	No nulo, solo letras y espacios, sin números. Y sin solo espacios
(Nombre)	Representa el nombre completo del Turista. Está formado por los atributos Nombres y Apellidos.	Texto (máx. 100 caracteres)	"Juan Pérez"	No nulo, se construye a partir de Nombres + Apellidos.
Foto de perfil	Imagen asociada al Turista.	Archivo en formato JPG/PNG	"foto123.png"	No nulo, formato válido JPG/PNG, tamaño ≤ 5MB.
Perfil	Información asociada al perfil del Turista (preferencias, intereses, etc.).	Texto estructurado	"Perfil #001", "Aventura"	Puede estar vacío al inicio, debe corresponder a un perfil existente si aplica.
Teléfono	Número de contacto del Turista.	Numérico (máx. 15 dígitos)	"+57 3104567890"	No nulo, debe cumplir formato de teléfono internacional. UNICO
Email	**Dirección de correo electrónico del Turista.**	**Texto (formato email)**	**"juan@mail.com"**	**No nulo, único, debe cumplir formato estándar de email.**

- Guía:

Atributo	Definición (significado)	Dominio de valores	Ejemplos	Reglas de validación
ID	Representa el código único que identifica al guía dentro del sistema. Este valor corresponde al número oficial de identificación de la persona	Entero positivo	1082G45247,21466120	No nulo, único, solo dígitos numéricos, mayor que 0 y su longitud >7
Nombre *(compuesto)*	Nombre completo del guía, formado por Nombres y Apellidos.	Texto (máx. 100)	"Carlos Pérez"	No nulo, derivado de Nombres + Apellidos.
Nombres	Nombre(s) del guía.	Texto (máx. 50)	"Carlos", "Ana María"	No nulo, solo letras y espacios.
Apellidos	Apellidos del guía.	Texto (máx. 50)	"Pérez López"	No nulo, solo letras y espacios.
Biografía	Descripción personal del guía.	Texto (máx. 500)	"Guía con 10 años de experiencia..."	Opcional, longitud máxima definida.
Perfil	Información del perfil del guía (ej.: experiencia, idiomas).	Texto estructurado o referencia	"Perfil #G01"	Puede ser opcional al inicio.
Foto de perfil	Imagen de identificación del guía.	Archivo JPG/PNG	"guia1.png"	No nula, tamaño ≤ 5MB. Se da con formato de dominio valido.
Teléfono	Número de contacto del guía.	Numérico (máx. 15)	"+57 31045678G0"	No nulo, debe cumplir formato internacional.
Email	Correo del guía.	Texto (formato email)	"guia@mail.com"	No nulo, único, debe cumplir formato email.
Verificado	Indica si el guía ha sido validado por la plataforma.	Booleano (Sí/No)	"Sí", "No"	No nulo, valor binario, default = "No".

- Localización:

Atributo	Significado	Dominio de valores	Ejemplos	Reglas de validación
Nombre	Nombre que identifica la localización (ciudad, municipio, pueblo o lugar donde se desarrollan tours).	Cadena de texto (máx. 100 caracteres).	“Bogotá”, “Medellín”, “Cartagena”.	Obligatorio (no nulo); no solo espacios; longitud ≤ 100 caracteres.
Latitud	Coordenada geográfica que indica la posición norte-sur de la localización en el mapa.	Número decimal entre -4.6067 y +13.5310.	4.7109, -2.4567	Obligatorio; debe ser un valor numérico decimal válido dentro del rango -4.6067 a +13.5310.
Longitud	Coordenada geográfica que indica la posición este-oeste de la localización en el mapa.	Número decimal entre -79.0474 y -66.8512.	-75.474, -74.0817	Obligatorio; debe ser un valor numérico decimal válido dentro del rango -79.0474 a -66.8512.

- Tipo de localización:

Atributo	Significado	Dominio de valores	Ejemplos	Reglas de validación
Nombre	Categoría o clase de la localización.	Cadena de texto (máx. 50 caracteres).	"Ciudad", "Municipio", "Área rural", "Parque natural"	Obligatorio (no nulo); no solo espacios; longitud ≤ 50.

- Idioma:

Atributo	Significado	Dominio de valores	Ejemplos	Reglas de validación
Nombre	Nombre del idioma que puede hablar un turista, guía o que se asocia a un tour.	Cadena de texto (máx. 50 caracteres).	"español", "inglés", "francés", "alemán"	Obligatorio (no nulo); no solo espacios; longitud ≤ 50; debe contener ú nicamente letras y espacios.

5. Decisiones y planeación estimadas:

En esta parte del documento vamos a explicar más a fondo cómo fue nuestra planeación sobre las primeras entidades y las posibles relaciones.

Primera Iteración.

En el modelo conceptual hemos decidido establecer un total de 7 entidades:

- Tour
- Guía
- Turista
- Localización
- Tipo de Localización
- Reserva
- Idioma

Estas entidades fueron elegidas porque representan la base de nuestra base de datos y resultan necesarias para gran parte del proyecto.

- **Tipo de Localización:** se creó para diferenciar de forma adecuada las distintas localizaciones registradas.
- **Reserva:** fue implementada porque necesitamos llevar un control de todas las reservas realizadas, así como del estado en que se encuentra cada una.
- **Idioma:** permite registrar todos los idiomas disponibles y vigentes en la base de datos.
- **Localización:** además de sus datos básicos, le agregamos los atributos de longitud y latitud, lo que nos dará una dirección más precisa.
- **Turista:** se le añadió el atributo perfil con el fin de identificar el tipo de perfil que corresponde a cada turista.
- **Guía:** se implementó porque es fundamental identificar la persona que va a dirigir y crear el tour.
- **Tour:** se implementó ya que su función principal es registrar y gestionar la información de los recorridos ofrecidos en el sistema.

Hasta el momento, hemos plasmado las historias de usuario 1, 2 y 3. Las historias 4 y 5 se estarán optimizando más adelante.

6. Planeación Estimada (Iteración 2).

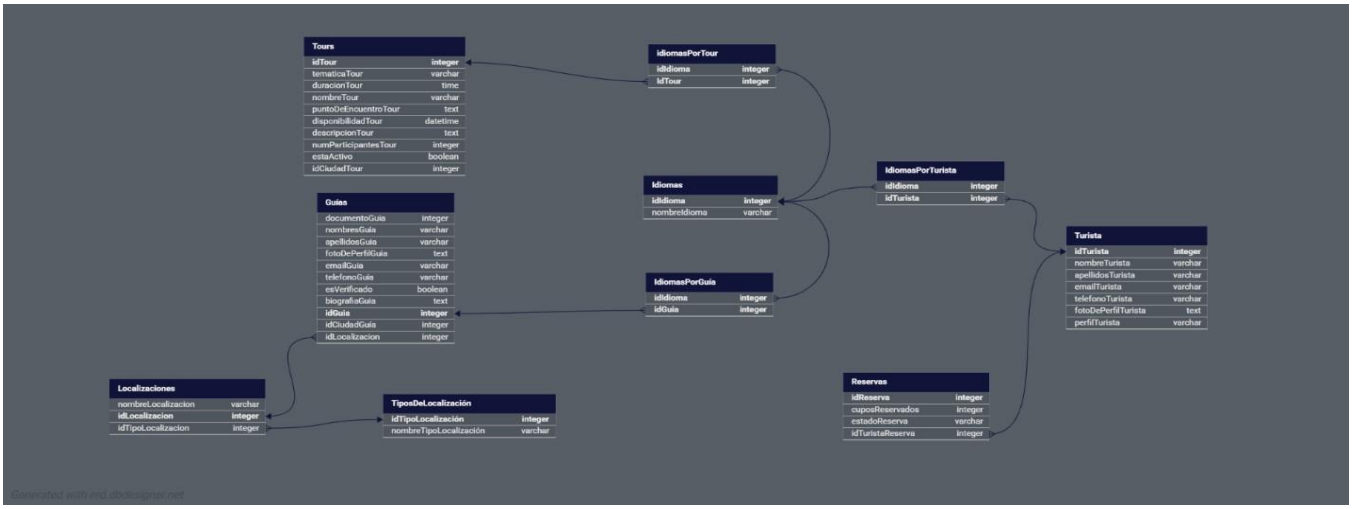
Para la versión 2 o iteración 2, hemos definido los siguientes aspectos a trabajar:

1. Generalización de una nueva entidad llamada Perfil de Usuario, ya que en Free Tour es clave manejar distintos perfiles.
2. Creación de la entidad Usuario, donde se podrán almacenar los datos personales de la persona que representa al perfil.
3. Puntos de Interés (una nueva entidad), la cual contará con sus propios atributos específicos.
4. Mejorar las relaciones entre las entidades ya existentes, con el fin de optimizar la estructura del modelo.
5. Mantener en stand by las historias de usuario 4 y 5, hasta contar con una correcta implementación de todo lo anterior.

7. Modelo lógico:

En esta etapa, se realiza la extrapolación del modelo entidad-relación previamente definido hacia un modelo relacional, con el fin de establecer las tablas resultantes, sus relaciones y las políticas de restricción necesarias, que posteriormente servirán de base para el diseño físico de la base de datos.

Es importante señalar que el modelo descrito en este documento, y los derivados de este, mantendrán un diseño consistente e independiente de cualquier DBMS. Esto permitirá no solo una implementación correcta en distintos sistemas, sino también garantizar la integridad y coherencia de los datos.



En este podemos observar las entidades *Reserva, Turista, Idioma, Tour, Guía, Localización* y *Tipo Localización* y las relaciones entre ellas. Para realizar su conversión al modelo relacional tenemos en cuenta que:

- Toda entidad se convierte en una tabla, y sus atributos en columnas.
- El identificador de la entidad se convierte en la clave primaria de la tabla.
- En las relaciones de uno a muchos la clave primaria de la entidad con cardinalidad "1" pasa a la tabla de la entidad cuya cardinalidad es muchos.
- Toda relación de muchos a muchos se convierte en una tabla que tendrá como columnas las llaves primarias de las entidades que intervienen en la relación.
- Los atributos asociados a una relación de muchos a muchos pasan a ser columnas de la nueva tabla.

Cumpliendo con estos criterios, tendremos el siguiente modelo relacional:

Es menester explicar que por convención nombraremos en plural y en PascalCase las tablas, y haremos uso de camelCase para el nombre de las columnas. Los criterios de desarrollo se explicarán en los siguientes apartados.

8. Política de identificadores:

Entenderemos en este apartado el proceso mediante el cual se definieron los atributos identificadores de cada tabla, distinguiéndolos en dos tipos:

- Naturales, los cuales son características propias y existentes de la entidad descrita.
- Sustitutas, no existentes en la realidad, por lo que son creadas únicamente para la base de datos. En caso de ser necesarias, las definiremos como un entero autoincremental no nulo.

Entendiendo esto, podremos explicar qué tipo de identificador se adoptó para cada tabla y su motivo.

- **Tours:** Debido a que ninguno de los atributos propios de la entidad cumple con el criterio de unicidad, creamos un identificador sustituto *idTour,* el cual será un entero autoincremental no nulo.
- **Idiomas:** Haremos uso de un identificador sustituto *idIdioma*. No usamos el nombre como identificador porque este puede presentar problemas en su unicidad al haber distintas variaciones de un mismo idioma, además de ser sensible a cambios según el idioma definido para la base de datos (ej. "Inglés" podría cambiar a "English", "Inglese", "Englisch", etc.)
- **Guías:** Se contempló la posibilidad de usar *documentoGuía* (ID en el modelo ER) ya que este cumple con el criterio de unicidad; sin embargo, se descartó esta opción debido a que presenta problemas al variar la longitud de este si incluimos documentos internacionales. Siendo este el caso, definimos una nueva columna *idGuía*.
- **Localizaciones:** Ya que el nombre de la localización no es único, debemos definir un identificador sustituto *idLocalización*.
- **TiposDeLocalización:** Ya que esta tabla tiene como función clasificar los distintos tipos de localización, observamos que esta será parte de *Localizaciones;* luego, para evitar la redundancia en los datos, usaremos un identificador sustituto *idTipoLocalización.*

- **Turistas:** No observamos que haya columnas que cumplan con el criterio de unicidad y se mantengan constantes, por lo que definimos el identificador sustituto *idTurista.*
- **Reservas:** Ya que no existe un atributo que cumpla con la unicidad, definimos el identificador sustituto *idReserva.*

También tenemos las tablas intermedias generadas por las relaciones N:M existentes. Sabiendo que estas tienen como columnas las llaves primarias de las involucradas en la relación, tendremos que estas serán posibles llaves primarias compuestas. Analizando cada tabla:

- **IdiomasPorTour:** Ya que esta no cuenta con ningún atributo ni propósito propios en la base de datos, sino el de relacionar *Idiomas* y *Tours*, usaremos como llave primaria la compuesta por *idIdioma* y *idTour.*
- **IdiomasPorGuía:** Ya que esta no cuenta con ningún atributo ni propósito propios en la base de datos, sino el de relacionar *Idiomas* y *Guías*, usaremos como llave primaria la compuesta por *idIdioma* y *idGuía.*
- **IdiomasPorTurista:** Ya que esta no cuenta con ningún atributo ni propósito propios en la base de datos, sino el de relacionar *Idiomas* y *Turistas*, usaremos como llave primaria la compuesta por *idIdioma* y *idTurista.*

9. Normalización:

Observaremos ahora de qué manera las tablas cumplen con los siguientes niveles de normalización:

1FN:

- Nombre y apellidos de Guía y Turista: en lugar de un atributo único “nombreCompleto”, se separaron en columnas distintas: nombres Guía, apellidos Guía, nombreTurista y apellidosTurista. Esto evita el problema de almacenar un valor compuesto que requeriría procesamiento adicional para separar sus partes.
- Correos electrónicos y teléfonos: se definieron como atributos individuales (email Guía, telefonoGuía, emailTurista, telefonoTurista) y no como listas o conjuntos. De esta manera, cada fila almacena un único valor en cada columna, lo cual garantiza la atomicidad.

- Idiomas: el hecho de que un guía, un turista o un tour puedan manejar varios idiomas podría violar la 1FN si se guardaran todos en una misma columna. Para resolver esto, se crearon tablas intermedias (IdiomasPorGuía, IdiomasPorTurista, IdiomasPorTour) en las que cada fila registra un solo idioma asociado a un identificador. Así se evita el almacenamiento de múltiples valores en una misma celda.

2FN: En el modelo lógico, las tablas con clave primaria simple (Tours, Turistas, Guías, Reservas, Idiomas, Localizaciones, TiposDeLocalización) no presentan problemas de dependencias parciales, ya que todos sus atributos dependen directamente de su identificador único.
El análisis se centra en las tablas intermedias generadas por relaciones N:M, que tienen claves primarias compuestas:

- IdiomasPorTour (idIdioma, idTour): no posee atributos adicionales; la combinación de idIdioma y idTour es la única que determina la existencia del registro. No hay dependencia parcial, ya que ningún atributo depende solo de idIdioma o solo de idTour.
- IdiomasPorGuía (idIdioma, idGuía): al igual que en el caso anterior, la PK compuesta no presenta atributos no clave, por lo que no hay dependencias parciales.
- IdiomasPorTurista (idIdioma, idTurista): cumple con la misma condición: todos los atributos dependen de la totalidad de la PK compuesta, sin redundancias.

3FN: Para esta parte fue necesario verificar que no existan dependencias transitivas que puedan generar redundancia en las actualizaciones de datos.
En la tabla Localizaciones, el atributo idTipoLocalización es clave foránea hacia la tabla TiposDeLocalización, donde se encuentra el atributo nombreTipoLocalización. De esta manera, el valor de nombreTipoLocalización no depende directamente de idLocalización sino de idTipoLocalización. Por lo tanto, existe una dependencia transitiva: idLocalización hacia nombreTipoLocalización.
Para resolver esta situación y cumplir con la 3FN, el modelo separa los atributos dependientes transitivos en una nueva tabla (TiposDeLocalización) que contiene la PK idTipoLocalización y su atributo dependiente nombreTipoLocalización.
Es recomendable que aquellos atributos que pueden tener más de un valor se representen como una nueva tabla, evitando así que existen tantos registros como cantidad de valores del atributo existan. Por ejemplo, es posible representar el atributo *teléfono* como una nueva tabla, la cual será relacionada con la tabla *Usuario,* existiendo entre ellas una relación 1:N.

10. Modelo Físico:

11. Objetivo del Script:

El script de creación tiene como objetivo trasladar el modelo relacional del diseño lógico, construyendo las tablas, definiendo los tipos de datos apropiados y estableciendo las relaciones y restricciones necesarias para garantizar la integridad de los datos.

12. Estructura General del Script:

El script se encuentra estructurado de forma secuencial, comenzando por la creación de la base de datos y el uso de esta, seguido por la creación de las tablas base y finalmente, las tablas intermedias que representan las relaciones de muchos a muchos.

13. Convenciones y Normas de Nomenclatura:

Para mantener la coherencia con el diseño lógico, se emplearon las siguientes convenciones de nomenclatura:

- Las tablas se nombran en plural y en PascalCase, respetando el modelo lógico (por ejemplo, Tours, Guías, Localizaciones).
- Los atributos o columnas utilizan la notación camelCase (por ejemplo, idTour, nombreGuía, fechaReserva).
- Las claves foráneas y primarias se nombran con el prefijo “id” seguido del nombre de la tabla referenciada.

14. Descripción de las Tablas:

14.1 Tours: contiene la información general de los tours ofrecidos. Su clave primaria es idTour. Incluye columnas como nombre, descripción, punto de encuentro, y también, si el tour está activo o no. Se relaciona con las tablas Localizaciones, Guías e IdiomasPorTour.

Constraints: aparte de las condiciones esenciales, como definir idTour como clave primaria y establecer que varios atributos (por ejemplo, nombre, temática, puntoDeEncuentro, entre otros) no acepten valores vacíos, se incluyen los siguientes constraints particulares:

- **NumeroDeParticipantesPositivos:** el nombre lo indica, numDeParticipantes tendrá que ser mayor que 0.
- **DuraciónVálida:** la duración tendrá que estar entre el rango de 0 a 720 minutos (12 horas).
- **FKLocalizaciónTour:** es la clave foránea idLocalización que no permite nulos por el diseño del modelo lógico.

```
CREATE TABLE Tours (
    idTour INT,
    temática VARCHAR(100) NOT NULL,
    duración INT NOT NULL,
    nombre VARCHAR(100) NOT NULL,
    puntoDeEncuentro VARCHAR(150),
    puntosDeInterés VARCHAR(1000) NOT NULL,
    disponibilidad DATETIME2 NOT NULL,
    descripción VARCHAR(1000) NOT NULL,
    numParticipantes INT,
    estáActivo BIT NOT NULL,
    idLocalización INT NOT NULL,
    CONSTRAINT PKTours PRIMARY KEY (idTour),
    CONSTRAINT NombreTourVálido CHECK (TRIM(nombre) <> ''),
    CONSTRAINT TemáticaVálida CHECK (TRIM(temática) <> ''),
    CONSTRAINT PuntoVálida CHECK (TRIM(puntoDeEncuentro) <> ''),
    CONSTRAINT NumeroParticipantesPositivos CHECK (numParticipantes > 0),
    CONSTRAINT DuraciónVálida CHECK (duración > 0 AND duración <= 720),
    CONSTRAINT PuntosDeInterésVálidos CHECK (TRIM(puntosDeInterés) <> ''),
    CONSTRAINT DescripciónVálida CHECK (TRIM(descripción) <> ''),
    CONSTRAINT FKLocalizaciónTour FOREIGN KEY (idLocalización) REFERENCES Localizaciones(idLocalización)
);
```

14.2 Idiomas: define los distintos idiomas disponibles para la interacción dentro del sistema. Su clave primaria es idIdioma. La columna nombre almacena el nombre del idioma.

Constraints:

- **PKIdiomas:** define **idIdioma** como clave primaria de la tabla.
- **NombreIdiomaÚnico:** establece que el campo **nombre** no puede repetirse entre registros, evitando duplicidades.
- **NombreIdiomaVálido:** impide que se registren nombres vacíos o con espacios en blanco.

```
CREATE TABLE Idiomas (
    idIdioma INT,
    nombre VARCHAR(30) NOT NULL,
    CONSTRAINT PKIdiomas PRIMARY KEY (idIdioma),
    CONSTRAINT nombreIdiomaÚnico UNIQUE (nombre),
    CONSTRAINT NombreIdiomaVálido CHECK (TRIM(nombre) <> '')
);
```

14.3 Guías: representa a los guías turísticos registrados. Posee como clave primaria idGuía. Incluye columnas como nombreGuía, documentoGuía y teléfonoGuía. Está relacionada con IdiomasPorGuía y con Tours mediante las reservas.

Constraints:

- **PKGuías:** establece **idGuía** como clave primaria.
- **FKLocalizaciónGuía:** define una relación con la tabla **Localizaciones** mediante la clave foránea **idLocalización.**
- **TeléfonoGuíaÚnico:** asegura que el número telefónico de cada guía sea único dentro del sistema.
- **DocumentoGuíaÚnico:** evita la repetición de documentos de identidad entre guías.
- **DocumentoGuíaVálido:** controla que el documento posea una longitud válida entre 7 y 18 caracteres.
- **NúmeroGuíaVálido:** verifica que el número telefónico cumpla con el formato esperado de tener un prefijo +.
- **EmailGuíaVálido:** valida que el correo electrónico contenga una estructura básica con un "@" y un caracter.
- **NombresGuíaVálidos, ApellidosGuíaVálidos, BiografíaGuíaVálida:** garantiza que el campo no esté vacío o formado únicamente por espacios.

```sql
CREATE TABLE Guías (
    idGuía INT,
    nombres VARCHAR(50) NOT NULL,
    apellidos VARCHAR(50) NOT NULL,
    documento BIGINT NOT NULL,
    foroDePerfil VARCHAR(MAX) NOT NULL,
    email VARCHAR(254) NOT NULL,
    teléfono VARCHAR(20) NOT NULL,
    esVerificado BIT NOT NULL,
    biografía VARCHAR(1000),
    idLocalización INT,
    CONSTRAINT PKGuías PRIMARY KEY (idGuía),
    CONSTRAINT FKLocalizaciónGuía FOREIGN KEY (idLocalización) REFERENCES Localizaciones(idLocalización),
    CONSTRAINT TelefonoGuíaÚnico UNIQUE (teléfono),
    CONSTRAINT DocumentoGuíaÚnico UNIQUE (documento),
    CONSTRAINT documentoGuíaVálido CHECK (LEN(CAST(documento AS VARCHAR(20))) >= 7
                                AND LEN(CAST(documento AS VARCHAR(20))) <= 18),
    CONSTRAINT NombresGuíaVálidos CHECK (TRIM(nombres) <> ''),
    CONSTRAINT ApellidosGuíaVálidos CHECK (TRIM(apellidos) <> ''),
    CONSTRAINT NúmeroGuíaVálido CHECK (teléfono LIKE '+%'),
    CONSTRAINT EmailGuíaVálido CHECK (email LIKE '_%@_%._%'),
    CONSTRAINT BiografíaGuíaVálida CHECK (TRIM(biografía) <> '')
);
```

14.4 Localizaciones: describe los lugares en los que se desarrollan los tours. Incluye atributos como idLocalización, nombre y dirección. Latitud y longitud son DECIMAL por almacenar valores reales de ángulos, por lo tanto, tienen precisiones de 6 decimales y (2 o 3 digitos enteros, por ejemplo: 20,021234).

Constraints:

- **PKLocalizaciones:** establece **idLocalización** como clave primaria.
- **FKTipoLocalización:** define la relación con la tabla **TiposDeLocalización** a través de la clave foránea **idTipoLocalización.**
- **NombreLocalizaciónVálida:** valida que el campo **nombre** no esté vacío o compuesto únicamente por espacio.
- **RangoGeográficoColombia:** restringe las coordenadas de **latitud** y **longitud** a los rangos geográficos reales de Colombia (latitud entre -4.2 y 13.5, longitud entre -79.0 y -66.0).

```sql
CREATE TABLE Localizaciones (
    idLocalización INT,
    nombre VARCHAR(100) NOT NULL,
    latitud DECIMAL(8, 6) NOT NULL,
    longitud DECIMAL(9, 6) NOT NULL,
    idTipoLocalización INT NOT NULL,
    CONSTRAINT PKLocalizaciones PRIMARY KEY (idLocalización),
    CONSTRAINT FKTipoLocalización FOREIGN KEY (idTipoLocalización) REFERENCES TiposDeLocalización(idTipoLocalización),
    CONSTRAINT NombreLocalizaciónVálida CHECK (TRIM(nombre) <> ''),
    CONSTRAINT RangoGeograficoColombia CHECK (latitud BETWEEN -4.2 AND 13.5
                                    AND longitud BETWEEN -79.0 AND -66.0)
);
```

14.5 TiposDeLocalización: categoriza las localizaciones según su tipo (por ejemplo, playa, museo, parque).

Constraints:

- **PKTiposDeLocalización:** establece **idTipoLocalización** como clave primaria.

```sql
CREATE TABLE TiposDeLocalización (
    idTipoLocalización INT,
    nombreTipoLocalización VARCHAR(100),
    CONSTRAINT PKTiposDeLocalización PRIMARY KEY (idTipoLocalización)
);
```

14.6 Turistas: contiene los datos de los turistas registrados en el sistema. Su clave primaria es idTurista. Incluye columnas como nombres, apellidos, email, documento teléfono (siendo todos VARCHAR). Ahora bien, fotoDePerfil es VARCHAR(MAX) para manejar tamaños muy grandes que son fotos de altas resoluciones, igual que documento con BIGINT solo que con números.

Constraints:

- **PKTurista:** establece idTurista como clave primaria.
- **TeléfonoTuristaÚnico:** garantiza que el número de `teléfono` sea único para cada turista, evitando duplicados.
- **documentoTuristaVálido:** verifica que la longitud del documento esté entre 7 y 18 caracteres, validando que no sea demasiado corto ni excesivamente largo.
- **NombresTuristaVálidos, ApellidosTuristaVálidos:** asegura que el campo no esté vacío ni contenga solo espacios en blanco.
- **NúmeroTuristaVálido:** comprueba que el `teléfono` comience con un "+" (formato internacional).
- **EmailTuristaVálido:** valida que el correo electrónico contenga una estructura básica con un "@" y un caracter

```
CREATE TABLE Turistas (
    idTurista INT,
    nombres VARCHAR(50) NOT NULL,
    apellidos VARCHAR(50) NOT NULL,
    email VARCHAR(254) NOT NULL,
    teléfono VARCHAR(20) NOT NULL,
    fotoDePerfil VARCHAR(MAX) NOT NULL,
    documento BIGINT NOT NULL,
    perfil VARCHAR(MAX),
    CONSTRAINT PKTurista PRIMARY KEY (idTurista),
    CONSTRAINT TeléfonoTuristaÚnico UNIQUE (teléfono),
    CONSTRAINT documentoTuristaVálido CHECK (LEN(CAST(documento AS VARCHAR(20))) >= 7
                                AND LEN(CAST(documento AS VARCHAR(20))) <= 18),
    CONSTRAINT NombresTuristaVálidos CHECK (TRIM(nombres) <> ''),
    CONSTRAINT ApellidosTuristaVálidos CHECK (TRIM(apellidos) <> ''),
    CONSTRAINT NúmeroTuristaVálido CHECK (teléfono LIKE '+%'),
    CONSTRAINT EmailTuristaVálido CHECK (email LIKE '_%@_%._%')
);
```

14.7 Reservas: registra la información de las reservas realizadas por los turistas. Su clave primaria es idReserva. Contiene las claves foráneas idTurista, idTour y fechaReserva. Garantiza la integridad referencial entre las tablas Turistas y Tours.

Constraints:

- **PKReservas:** establece `idReserva` como clave primaria.
- **PKPositiva:** obliga a que `idReserva` sea mayor a 0, evitando identificadores negativos o nulos.

- **FKTuristaReserva:** define `idTuristaReserva` como clave foránea que referencia `idTurista` en la tabla **Turistas.**
- **FKTourReserva:** define `idTour` como clave foránea que referencia `idTour` en la tabla **Tours.**
- **CuposPositivos:** garantiza que el número de `cuposReservados` sea mayor a 0 (no se permiten reservas con cero cupos).
- **EstadoVálido:** asegura que el campo `estadoReserva` no esté vacío ni contenga únicamente espacios en blanco.

```
CREATE TABLE Reservas (
    idReserva INT,
    cuposReservados INT NOT NULL,
    estadoReserva VARCHAR(20) NOT NULL,
    idTuristaReserva INT,
    idTour INT
    CONSTRAINT PKReservas PRIMARY KEY (idReserva),
    CONSTRAINT PKPositiva CHECK (idReserva > 0),
    CONSTRAINT FKTuristaReserva FOREIGN KEY (idTuristaReserva) REFERENCES Turistas(idTurista),
    CONSTRAINT FKTourReserva FOREIGN KEY (idTour) REFERENCES Tours(idTour),
    CONSTRAINT CuposPositivos CHECK (cuposReservados > 0),
    CONSTRAINT estadoVálido CHECK (TRIM(estadoReserva) <> '')
);
```

14.8 IdiomasPorTour: tabla intermedia que relaciona los idiomas con los tours disponibles. Su clave primaria compuesta está conformada por idIdioma y idTour.

Constraints:

En esta tabla solo se cuentan con 2 claves foráneas, idIdioma e idTour, siendo ambas pertenecientes a sus mismas tablas como Idiomas y Tours (por la relación de muchos a muchos).

```
CREATE TABLE IdiomasPorTour (
    idIdioma INT,
    idTour INT,
    CONSTRAINT FKIdiomaTour FOREIGN KEY (idIdioma) REFERENCES Idiomas(idIdioma),
    CONSTRAINT FKGuíaTour FOREIGN KEY (idGuía) REFERENCES Guías(idGuía),
);
```

14.9 IdiomasPorGuía: relaciona los idiomas que domina cada guía. Su clave primaria compuesta está conformada por idIdioma y idGuía.

Constraints:

En esta tabla solo se cuentan con 2 claves foráneas, idIdioma e idGuía, siendo ambas pertenecientes a sus mismas tablas como Idiomas y Guías (por la relación de muchos a muchos).

```
CREATE TABLE IdiomasPorGuia (
    idIdioma INT,
    idGuía INT,
    CONSTRAINT FKIdiomaGuía FOREIGN KEY (idIdioma) REFERENCES Idiomas(idIdioma),
    CONSTRAINT FKGuía FOREIGN KEY (idGuía) REFERENCES Guías(idGuía),
);
```

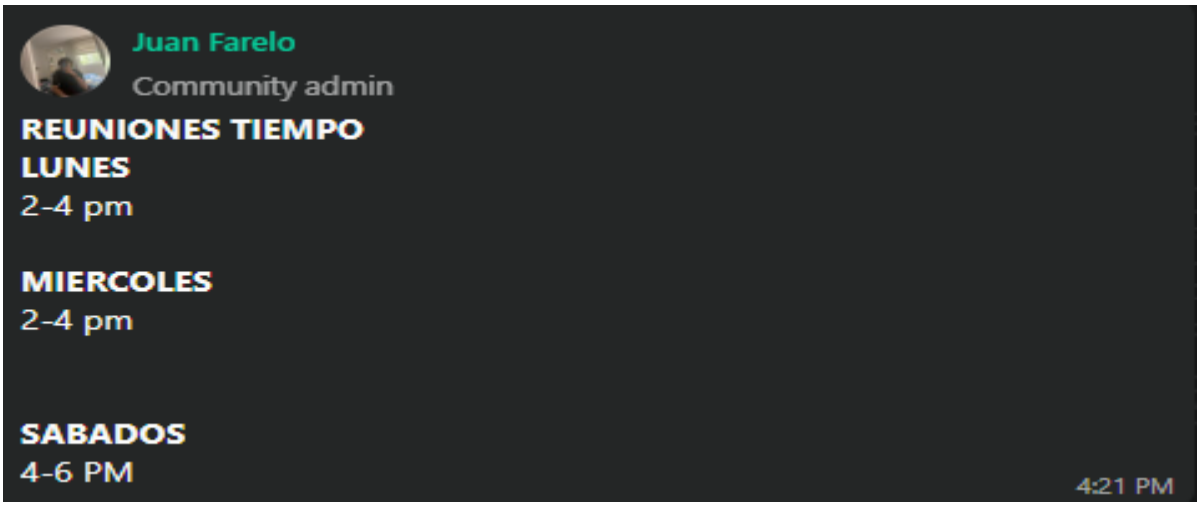
14.10 IdiomasPorTurista: asocia los idiomas que domina cada turista. Su clave primaria compuesta está conformada por idIdioma y idTurista.

Constraints:

En esta tabla solo se cuentan con 2 claves foráneas, idIdioma e idTurista, siendo ambas pertenecientes a sus mismas tablas como Idiomas y Turistas (por la relación de muchos a muchos).

```
CREATE TABLE IdiomasPorTurista (
    idIdioma INT,
    idTurista INT,
    CONSTRAINT FKIdiomaPorTurista FOREIGN KEY (idTurista) REFERENCES Turistas(idTurista),
    CONSTRAINT FKTuristaPorIdioma FOREIGN KEY (idIdioma) REFERENCES Idiomas(idIdioma)
);
```

15. **Cronograma de trabajo:** El cronograma de trabajo fue una herramienta fundamental para la organización y el progreso del proyecto. Además, facilitó la evaluación continua de los avances y la identificación temprana de posibles retrasos o dificultades. Gracias a este esquema, el equipo mantuvo una comunicación constante, mejoró la coordinación general y logró cumplir con los objetivos propuestos en cada fase del desarrollo de manera eficiente y ordenada.



16. Traslado de modelo lógico a físico: se hicieron conversiones entre el modelo lógico para el físico, tales como el uso de tipos de datos más eficaces como de INT a **BIGINT** para números muy grandes, text a **VARCHAR(MAX)** para las fotos de perfil que tienen un tamaño grande y de Boolean a **BIT.** Además de lo antes mencionado, se pudo hacer el traspaso de forma sencilla gracias a la correcta normalización de tablas que se manejó durante esta primera iteración.