**Documentar relaciones y reglas de integridad**

**Relación Usuarios – Turistas**

* **Tabla padre:** usuarios
* **Tabla hija:** turistas
* **Llave foránea:** turistas.id\_usuario → usuarios.id\_usuario
* **Cardinalidad:** 1 usuario puede estar vinculado a un turista.
* **Regla de integridad:** No puede existir un turista asociado a un usuario inexistente.

**Relación Grupos – Itinerarios**

* **Tabla padre:** grupos
* **Tabla hija:** itinerarios
* **Llave foránea:** itinerarios.id\_grupo → grupos.id\_grupo
* **Cardinalidad:** Un grupo puede tener varios itinerarios.
* **Regla de integridad:** Un itinerario no puede existir sin un grupo asociado.

**Relación Estados\_presupuesto – Itinerarios**

* **Tabla padre:** estados\_presupuesto
* **Tabla hija:** itinerarios
* **Llave foránea:** itinerarios.estado\_presupuesto\_id → estados\_presupuesto.id\_estado
* **Cardinalidad:** Un estado puede aplicarse a muchos itinerarios.
* **Regla de integridad:** Un itinerario siempre debe tener un estado de presupuesto válido.

**Relación Itinerarios – Itinerario\_programas**

* **Tabla padre:** itinerarios
* **Tabla hija:** itinerario\_programas
* **Llave foránea:** itinerario\_programas.id\_itinerario → itinerarios.id\_itinerario
* **Cardinalidad:** Un itinerario puede tener múltiples programas asociados.

**Relación Programas – Itinerario\_programas**

* **Tabla padre:** programas
* **Tabla hija:** itinerario\_programas
* **Llave foránea:** itinerario\_programas.id\_programa → programas.id\_programa
* **Cardinalidad:** Un programa puede ser usado en varios itinerarios.

**Relación Itinerario\_programas – Detalle\_machu\_itinerario**

* **Tabla padre:** itinerario\_programas
* **Tabla hija:** detalle\_machu\_itinerario
* **Llave foránea:** detalle\_machu\_itinerario.id\_itinerario\_programa → itinerario\_programas.id\_itinerario\_programa
* **Cardinalidad:** Relación 1:1 (cada itinerario de tipo Machu Picchu tiene un único detalle).

**Relación Itinerario\_programas – Detalle\_transporte\_itinerario**

* **Tabla padre:** itinerario\_programas
* **Tabla hija:** detalle\_transporte\_itinerario
* **Llave foránea:** detalle\_transporte\_itinerario.id\_itinerario\_programa → itinerario\_programas.id\_itinerario\_programa
* **Cardinalidad:** Un itinerario puede tener varios transportes asociados.

**Relación Transportes – Detalle\_transporte\_itinerario**

* **Tabla padre:** transportes
* **Tabla hija:** detalle\_transporte\_itinerario
* **Llave foránea:** detalle\_transporte\_itinerario.id\_transporte → transportes.id\_transporte
* **Cardinalidad:** Un transporte puede ser asignado a múltiples itinerarios.

**Relación Itinerarios – Itinerario\_turistas**

* **Tabla padre:** itinerarios
* **Tabla hija:** itinerario\_turistas
* **Llave foránea:** itinerario\_turistas.id\_itinerario → itinerarios.id\_itinerario
* **Cardinalidad:** Relación N:M (un itinerario puede tener varios turistas y un turista puede pertenecer a varios itinerarios).

**Relación Turistas – Itinerario\_turistas**

* **Tabla padre:** turistas
* **Tabla hija:** itinerario\_turistas
* **Llave foránea:** itinerario\_turistas.id\_turista → turistas.id\_turista
* **Cardinalidad:** Relación N:M, gestionada con clave primaria compuesta.

**Implementar ON DELETE y ON UPDATE**

Para asegurar consistencia se recomiendan las siguientes reglas:

* **ON DELETE CASCADE** en tablas de relación (itinerario\_turistas, itinerario\_programas, detalle\_machu\_itinerario, detalle\_transporte\_itinerario).  
  Esto asegura que al eliminar un itinerario, programa o turista, se eliminen automáticamente los registros relacionados.
* **ON DELETE RESTRICT** en entidades principales (usuarios, grupos, programas, transportes).  
  Esto evita borrar registros base si aún están en uso.

Ejemplo de implementación:

ALTER TABLE itinerario\_turistas

ADD CONSTRAINT fk\_itinerario FOREIGN KEY (id\_itinerario)

REFERENCES itinerarios(id\_itinerario)

ON DELETE CASCADE

ON UPDATE CASCADE;

ALTER TABLE itinerario\_turistas

ADD CONSTRAINT fk\_turista FOREIGN KEY (id\_turista)

REFERENCES turistas(id\_turista)

ON DELETE CASCADE

ON UPDATE CASCADE;

**Validar integridad referencial**

La integridad referencial garantiza que:

* No se pueden insertar registros en tablas hijas sin que exista la entidad en la tabla padre.
* No quedan registros huérfanos cuando se eliminan entidades principales.
* Se mantienen consistentes las claves primarias y foráneas ante actualizaciones.

**Ejemplo de inserción inválida:**

-- Error: turista inexistente

INSERT INTO itinerario\_turistas (id\_itinerario, id\_turista)

VALUES (1, 999);

**Ejemplo de eliminación en cascada:**

-- Eliminar un itinerario elimina automáticamente los turistas vinculados a ese itinerario

DELETE FROM itinerarios WHERE id\_itinerario = 1;

**Resumen de relaciones y reglas**

| **Tabla padre** | **Tabla hija** | **Cardinalidad** | **ON DELETE** | **ON UPDATE** |
| --- | --- | --- | --- | --- |
| usuarios | turistas | 1:1 | RESTRICT | CASCADE |
| grupos | itinerarios | 1:N | RESTRICT | CASCADE |
| estados\_presupuesto | itinerarios | 1:N | RESTRICT | CASCADE |
| itinerarios | itinerario\_programas | 1:N | CASCADE | CASCADE |
| programas | itinerario\_programas | 1:N | RESTRICT | CASCADE |
| itinerario\_programas | detalle\_machu\_itinerario | 1:1 | CASCADE | CASCADE |
| itinerario\_programas | detalle\_transporte\_itinerario | 1:N | CASCADE | CASCADE |
| transportes | detalle\_transporte\_itinerario | 1:N | RESTRICT | CASCADE |
| itinerarios | itinerario\_turistas | N:M | CASCADE | CASCADE |
| turistas | itinerario\_turistas | N:M | CASCADE | CASCADE |