

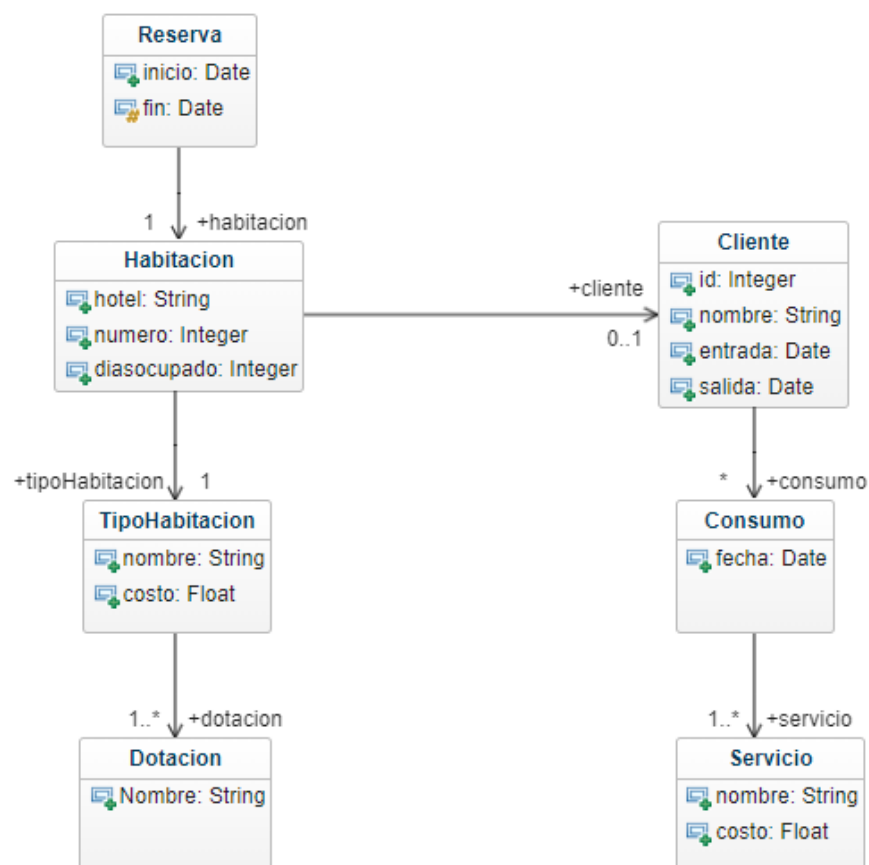
Juan Sebastián Sánchez - 202121498

Andrés Felipe Gómez - 202021189

Juan Felipe Camelo - 202122062

Análisis y modelo conceptual

- a. Proponga un modelo conceptual en UML o E/R que describa las entidades del modelo de datos para la aplicación que se quiere desarrollar.



Diseño de la base de datos

1. Análisis de la carga de trabajo (workload).

- a. Identifiquen entidades y sus atributos

Reserva:

- Inicio (Date): Es una fecha del inicio de la reserva.
- Fin (Date): Es la fecha de fin de la reserva.
- Habitación: Tiene la habitación respectiva embebida

Habitacion:

- Hotel (String): El nombre del hotel al cual pertenece la habitación.
- Número (Int): Es el número de la habitación.
- DiasOcupado (Int): Es un contador en el cual se guardará el número de días que la habitación respectiva se encuentra ocupada en un intervalo de tiempo.
- Clientes (Array<Cliente>): Tiene embebido una lista de los clientes que se hospedan en la habitación.
- TipoHabitación: Tiene el tipo de habitación respectivo a esa habitación.

TipoHabitacion:

- Nombre (String): El tipo de habitación de la habitación respectiva (suite, doble, etc).
- Precio (Float): El precio del tipo de la habitación respectiva. Mejores tipos de habitación tendrán un mayor precio.
- Dotación (Array<String>): Contiene todos los elementos que tiene la habitación.

Cliente:

- ID (Int): El identificador único para cada cliente.
- Nombre (String): El nombre del cliente.
- Entrada (Date): La fecha de entrada al hotel del respectivo cliente.
- Salida (Date): La fecha de salida del hotel del respectivo cliente.
- Consumo (Array<Consumo>): Tiene la lista de consumos hechos por el cliente.

Consumo:

- Fecha (Date): La fecha en la que se hizo el consumo.
- Servicio: El servicio en el cual se hizo el consumo.

Servicio:

- Nombre (String): Nombre del servicio (gimnasio, lavandería, etc).
- Costo (Float): Valor del servicio respectivo.

- a. Cuantifiquen las entidades (cantidad de registros que tendría la BD para cada una de las entidades, pueden encontrar un aproximado en el enunciado).

Reserva:

- La creación de reservas de habitaciones es muy común, esta operación corresponde a la operación fundamental de la aplicación y se realiza cada vez que un cliente hace una reserva en alguno de los hoteles.
- **Cantidad Diaria:** 50 reservas/día

- **Cantidad Anual:** 18250 reservas/año

Habitacion:

- La creación de habitaciones es poco común, esta operación se hace muy ocasionalmente cuando se decide modificar las habitaciones disponibles. Evidentemente hay un número inicial de habitaciones en un hotel que se mantendrá relativamente constante en la mayoría de los casos.
- **Cantidad Promedio:** 200 habitaciones

TipoHabitacion:

- La creación de tipos de habitaciones es poco común, esta operación se hace muy ocasionalmente cuando se decide modificar los tipos de habitaciones disponibles. En términos generales se estima que la cantidad de tipos de habitaciones se mantendrá aproximadamente constante.
- **Cantidad Promedio:** 20 tipo de habitación

Cliente:

- La creación de clientes generalmente es común, la operación se realiza cada vez que un nuevo cliente se hospedaré en un hotel y, aunque puede haber clientes que se hospeden varias veces, es usual que en los períodos de tiempo entre estadías repetidas nuevos clientes se asocien al hotel. Usualmente la creación de clientes estará asociada a reservas:
- **Cantidad Anual:** 19000 clientes/año

Consumo:

- La creación de consumos es frecuente, esta sucede cada vez que un cliente consume un servicio proveído por el hotel.
- **Cantidad Diaria:** 500 consumos/día
- **Cantidad Anual:** 182500 consumos/año

Servicio:

- La creación de servicios es de muy baja frecuencia, esta sucede cada vez que se quieren modificar los servicios de un hotel, lo cual no sucede a menudo. En promedio la cantidad de servicios de un hotel permanece relativamente constante.
- **Cantidad Promedio:** 35

- b. Analicen las operaciones de lectura y escritura para cada entidad. Para ello utilicen una tabla como la del ejemplo del anexo A. Recuerden que este análisis sirve para saber qué información se accederá de manera conjunta.

Entidades	Operación	Información Necesaria	Tipo
TipoHabitacion	Consultar tipos de habitación y sus dotaciones	Nombre Tipo + Precio Tipo + Dotación Tipo	Lectura
TipoHabitacion	Registrar/Actualizar tipos de habitación	Nombre Tipo + Dotación Tipo + Precio Tipo	Escritura

TipoHabitacion	Borrar tipos de habitación	Nombre Tipo	Escritura
Habitacion	Consultar habitaciones y sus características	Hotel + Número + Nombre Tipo + Precio Tipo	Lectura
Habitacion	Registrar/Actualizar habitaciones	Hotel + Número + Días Ocupado + Info Tipo	Escritura
Habitacion	Borrar habitaciones	Hotel + Número	Escritura
Servicio	Consultar servicios y sus descripciones	Nombre + Precio	Lectura
Servicio	Registrar/Actualizar servicios	Nombre + Precio	Escritura
Servicio	Borrar servicios	Nombre	Escritura
Reserva	Consultar reservas y sus características	Id Reserva + Inicio + Fin	Lectura
Reserva	Registrar/Actualizar reservas	Inicio + Fin + Habitacion (Incluye Info Tipo) + Clientes	Escritura
Reserva	Borrar reservas	Id Reserva	Escritura
Cliente, Reserva	Consultar llegada de un cliente	Entrada + Id Cliente + Nombre Cliente	Lectura
Cliente, Reserva	Registrar llegada de un cliente	ID Reserva + ID Cliente + Nombre Cliente + Entrada	Escritura
Cliente, Consumo, Servicio	Consultar consumos de un cliente	Id Cliente + Nombre Cliente + Fecha Consumo + Nombre Servicio + Costo Consumo	Lectura
Consumo	Registrar/Actualizar consumos	Id Reserva + Id Cliente + Fecha Consumo + Nombre Servicio + Costo Consumo	Escritura
Consumo	Borrar consumos	Id Reserva + Id Cliente + Fecha Consumo	Escritura
Cliente, Reserva	Consultar salida de un cliente	Id Cliente + Nombre + Fin	Lectura
Cliente, Reserva	Registrar salida de un cliente	ID Reserva + ID Cliente + Nombre Cliente + Salida	Escritura
Consumo, Habitacion, Servicio	Mostrar el dinero recolectado por servicios en cada habitación en el último año corrido	Hotel + Numero Habitación + Costo Total Consumos (Sacado de Precio Consumo Individual) + Fecha Consumo	Lectura

Reserva, Habitacion	Mostrar el índice de ocupación de cada una de las habitaciones del hotel en el último año corrido	Hotel + Numero Habitación + Índice Ocupación (Sacado de Inicio Reserva y Fin Reserva)	Lectura
Cliente, Consumo	Mostrar el consumo en hotelandes por un cliente, en un rango de fechas indicado	Id Cliente + Fecha Consumo + Gasto Total Consumos (Usando Precio Consumo)	Lectura
Cliente	Consultar los clientes excelentes	Entrada Cliente + Salida Cliente + Id Cliente + Nombre Cliente	Lectura

c. Cuantifiquen las operaciones de lectura y escritura para cada entidad. Para ello utilicen una tabla como la del ejemplo del anexo B.

Entidades	Operación	Información Necesaria	Tipo	Tasa
TipoHabitacion	Consultar tipos de habitación y sus dotaciones	Nombre Tipo + Precio Tipo + Dotación Tipo	Lectura	10/día
TipoHabitacion	Registrar/Actualizar tipos de habitación	Nombre Tipo + Dotación Tipo + Precio Tipo	Escritura	1/semana
TipoHabitacion	Borrar tipos de habitación	Nombre Tipo	Escritura	2/año
Habitacion	Consultar habitaciones y sus características	Hotel + Número + Nombre Tipo + Precio Tipo	Lectura	50/día
Habitacion	Registrar/Actualizar habitaciones	Hotel + Número + Días Ocupado + Info Tipo	Escritura	2/semana
Habitacion	Borrar habitaciones	Hotel + Número	Escritura	2/año
Servicio	Consultar servicios y sus descripciones	Nombre + Precio	Lectura	2/semana
Servicio	Registrar/Actualizar servicios	Nombre + Precio	Escritura	2/semana
Servicio	Borrar servicios	Nombre	Escritura	4/año
Reserva	Consultar reservas y sus características	Id Reserva + Inicio + Fin	Lectura	40/día
Reserva	Registrar/Actualizar reservas	Inicio + Fin + Habitacion (Incluye Info Tipo) + Clientes	Escritura	50/día
Reserva	Borrar reservas	Id Reserva	Escritura	10/semana
Cliente, Reserva	Consultar llegada de un cliente	Entrada + Id Cliente + Nombre Cliente	Lectura	25/día

Cliente, Reserva	Registrar llegada de un cliente	ID Reserva + ID Cliente + Nombre Cliente + Entrada	Escritura	30/día
Cliente, Consumo, Servicio	Consultar consumos de un cliente	Id Cliente + Nombre Cliente + Fecha Consumo + Nombre Servicio + Costo Consumo	Lectura	250/día
Consumo	Registrar/Actualizar consumos	Id Reserva + Id Cliente + Fecha Consumo + Nombre Servicio + Costo Consumo	Escritura	500/día (Con todos los clientes) 7/día (Por cliente particular)
Consumo	Borrar consumos	Id Reserva + Id Cliente + Fecha Consumo	Escritura	50/día
Cliente, Reserva	Consultar salida de un cliente	Id Cliente + Nombre + Fin	Lectura	25/día
Cliente, Reserva	Registrar salida de un cliente	ID Reserva + ID Cliente + Nombre Cliente + Salida	Escritura	30/día
Consumo, Habitación, Servicio	Mostrar el dinero recolectado por servicios en cada habitación en el último año corrido	Hotel + Numero Habitación + Costo Total Consumos (Sacado de Precio Consumo Individual) + Fecha Consumo	Lectura	1/semana
Reserva, Habitación	Mostrar el índice de ocupación de cada una de las habitaciones del hotel en el último año corrido	Hotel + Numero Habitación + Índice Ocupación (Sacado de Inicio Reserva y Fin Reserva)	Lectura	1/semana
Cliente, Consumo	Mostrar el consumo en hotelandes por un cliente, en un rango de fechas indicado	Id Cliente + Fecha Consumo + Gasto Total Consumos (Usando Precio Consumo)	Lectura	1/semana
Cliente	Consultar los clientes excelentes	Entrada Cliente + Salida Cliente + Id Cliente + Nombre Cliente	Lectura	1/semana

2. Describan las entidades de datos y las relaciones entre ellas que corresponden al modelo conceptual UML propuesto.

a. La lista de entidades con la descripción de cada una de ellas.

Reserva	Nuestra entidad de reserva es la que contiene toda la información que tendremos en nuestra base de datos, esto principalmente contiene una
---------	--

	fecha de inicio de la reserva, una fecha de fin y una habitación asociada a la reserva.
Habitación	La entidad de habitación es aquella con la cual accedemos a los clientes. Todas las habitaciones tienen un hotel al cual pertenecen y un número de habitación que sirven para diferenciarlas (estos dos atributos hacen a cada habitación única). Además de esto, una habitación tiene un tipo de habitación asociado, los clientes que se quedaron en la habitación durante la reserva y un atributo extra que indica el número de días que lleva ocupada una habitación, que la utilizamos para simplificar las consultas.
Tipo habitación	El tipo de habitación es una entidad con la cual podemos identificar nuestras habitaciones, el tipo de habitación tiene un nombre que la identifica, un costo y una lista donde se guardan la dotación que tiene asociada dicho tipo de habitación, por ejemplo, minibar, jacuzzi, etc.
Cliente	Cliente es una entidad que guardará la información de los clientes al igual que los consumos asociados al mismo, cada cliente tiene un id y nombre asociado para identificarlo, al igual que una fecha de entrada al hotel y de salida. Además, guardamos un arreglo de objetos de sus consumos.
Consumo	Cada consumo tiene una fecha cuando ocurrió dicho consumo y tiene un servicio asociado, donde se guarda el nombre y precio del consumo. Por lo cual cada consumo tiene una fecha, servicio y costo.

- b. Las relaciones entre entidades y su cardinalidad (uno a uno, uno a muchos, o muchos a muchos).

Nuestro modelamiento funciona tal que las relaciones se ven así:

Reserva – 1..* a 1..* --> habitación

Habitación -- 1..* a 1--> tipo habitación

Habitación -- 1 a 1..* --> cliente

Cliente – 1 a 1..* --> consumos

Sin embargo, para nuestra base de datos y uml lo modelamos diferentemente para dar a entender cómo se estructuran las consultas y se ve de esta manera:

Reserva – 1 a 1--> habitación

Habitación -- 1 a 1--> tipo habitación

Habitación -- 1 a 1..* --> cliente

Cliente – 1 a 1..* --> consumos

- c. El análisis de selección de esquema de asociación (referenciado o embebido) para cada relación entre entidades. Para ello use la tabla de análisis vista en clase, la cual se retoma en el anexo C, junto con los resultados del análisis de la carga de trabajo (workload), descrita antes.

Habitacion-Reservas

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces in information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information a “has-a,” “contains,” or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	No	Yes
Archival	Should the pieces of information be archived at the same time?	No	Yes
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	Yes	No
Data Duplication	Would data duplication be too complicated to manage and undesired?	Yes	No
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	Yes	No
Document Growth	Would the embedded piece grow without bound?	Yes	No
Workload	Are the pieces of information written at different in a write-heavy workload?	Yes	No
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

TipoHabitacion-Habitacion

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces in information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information a “has-a,” “contains,” or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	No	Yes
Archival	Should the pieces of information be archived at the same time?	No	Yes
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	Yes	No
Data Duplication	Would data duplication be too complicated to manage and undesired?	Yes	No
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	Yes	No
Document Growth	Would the embedded piece grow without bound?	Yes	No
Workload	Are the pieces of information written at different in a write-heavy workload?	Yes	No
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

Dotacion-TipoHabitacion

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces in information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information a “has-a,” “contains,” or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	No	Yes
Archival	Should the pieces of information be archived at the same time?	No	Yes
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	Yes	No
Data Duplication	Would data duplication be too complicated to manage and undesired?	Yes	No

Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	Yes	No
Document Growth	Would the embedded piece grow without bound?	Yes	No
Workload	Are the pieces of information written at different in a write-heavy workload?	Yes	No
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

Cliente-Habitacion

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces in information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information a "has-a," "contains," or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	No	Yes
Archival	Should the pieces of information be archived at the same time?	No	Yes
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	Yes	No
Data Duplication	Would data duplication be too complicated to manage and undesired?	Yes	No
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	Yes	No
Document Growth	Would the embedded piece grow without bound?	Yes	No
Workload	Are the pieces of information written at different in a write-heavy workload?	Yes	No
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

Consumo-Cliente

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces in information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information a "has-a," "contains," or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	No	Yes
Archival	Should the pieces of information be archived at the same time?	No	Yes
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	Yes	No
Data Duplication	Would data duplication be too complicated to manage and undesired?	Yes	No
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	Yes	No
Document Growth	Would the embedded piece grow without bound?	Yes	No
Workload	Are the pieces of information written at different in a write-heavy workload?	Yes	No
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

Servicio-Consumo

Guideline Name	Question	Embed	Reference
Simplicity	Would keeping the pieces in information together lead to a simpler data model and code?	Yes	No
Go Together	Do the pieces of information a “has-a,” “contains,” or similar relationship?	Yes	No
Query Atomicity	Does the application query the pieces of information together?	Yes	No
Update Complexity	Are the pieces of information updated together?	No	Yes
Archival	Should the pieces of information be archived at the same time?	No	Yes
Cardinality	Is there a high cardinality (current or growing) in the child side of the relationship?	Yes	No
Data Duplication	Would data duplication be too complicated to manage and undesired?	Yes	No
Document Size	Would the combined size of the pieces of information take too much memory or transfer bandwidth for the application?	Yes	No
Document Growth	Would the embedded piece grow without bound?	Yes	No
Workload	Are the pieces of information written at different in a write-heavy workload?	Yes	No
Individuality	For the children side of the relationship, can the pieces exist by themselves without a parent?	No	Yes

- d. Una descripción gráfica usando Json de cada relación entre entidades en donde presente un ejemplo de datos junto con el esquema de asociación usado (referenciado o embebido). En el anexo D se muestra un ejemplo de lo que se requiere.

