

# “Sistemas Transaccionales: Iteración 2”

Felix S. Rojas Casadiego  
Universidad de los Andes, Bogotá, Colombia  
{fs.rojas}@uniandes.edu.co  
Fecha de presentación: Abril 6 de 2022

## Tabla de contenido

1. Modelo conceptual del negocio.....	1
2. Análisis.....	¡Error! Marcador no definido.
a. Requerimientos Funcionales .....	2
3. Modelo de datos relacional .....	5
a. Nivel de normalización .....	8

## 1. Modelo conceptual del negocio

El modelo es similar al realizado en la iteración 2, solo que para un mejor desarrollo se disminuyó el número de clases uniendo aquellas que no eran totalmente necesarias para cumplir con los requerimientos funcionales. En cuanto a los roles de usuario, se puede observar que se concentran en la clase `User`, la cual tiene un atributo `rol` que permite identificar al usuario en el sistema y los permisos que tendrá.

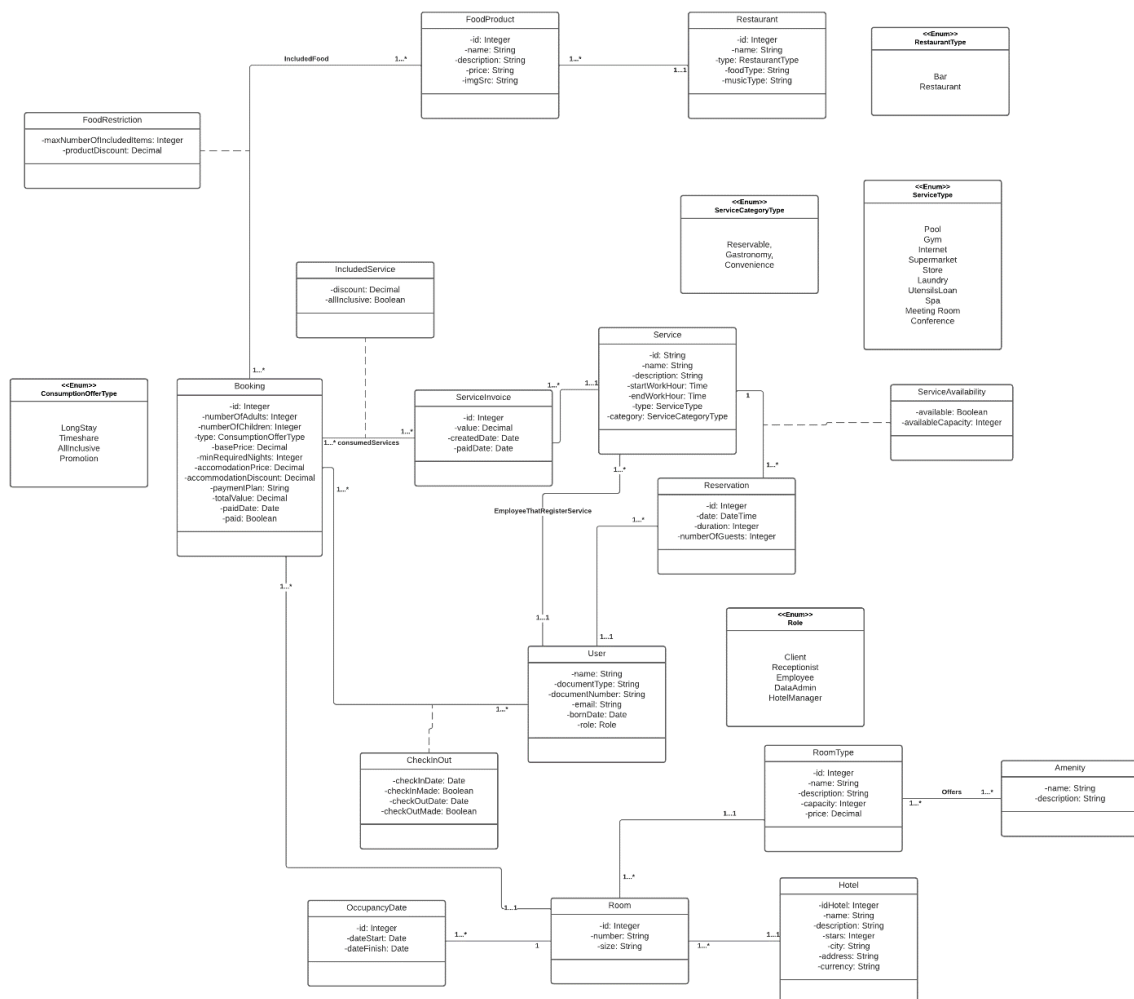


Figura 1. Modelo conceptual

## 2. Funcionalidades principales

### a. Requerimientos Funcionales

<b>Nombre</b>	RF1.
<b>Resumen</b>	Registrar roles de usuario. Esta operación es realizada por el administrador del sistema.
<b>Entradas</b>	
Rol	
<b>Resultados</b>	
Rol registrado	
<b>RNF asociados</b>	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que hay que ejecutar el registro de los roles para los usuarios y persistirlos para que el sistema pueda crear usuarios.

<b>Nombre</b>	RF2.
<b>Resumen</b>	Registrar usuario. Esta operación es realizada por el administrador del hotel o por un cliente independiente.
<b>Entradas</b>	
Tipo de documento	
Numero de documento	
Nombres	
Apellidos	
Correo	
Rol	
<b>Resultados</b>	
Usuario registrado	
<b>RNF asociados</b>	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	
RNF4. Concurrencia	

El requerimiento necesita comportamiento transaccional ya que hay que el sistema debe permitir registrar usuarios para que estos puedan acceder a la información necesaria para operar el hotel. Si es el caso de un cliente, requiere estar registrado para poder reservar entre muchas otras cosas.

<b>Nombre</b>	RF3.
<b>Resumen</b>	Registrar la descripción de los tipos de habitaciones del hotel con su dotación correspondiente. Esta operación es realizada por el administrador de datos del hotel.
<b>Entradas</b>	
Nombre	
Descripción	
Capacidad	
Precio	
Dotación correspondiente	
<b>Resultados</b>	
Tipo de habitación creada	

<b>RNF asociados</b>
RNF1. Seguridad
RNF2. Privacidad
RNF3. Persistencia

El requerimiento necesita comportamiento transaccional ya que es necesario poder registrar con que tipo de habitaciones cuenta el hotel para poder estructurar la oferta comercial. Además, esta información debe quedar persistida para su posterior uso.

<b>Nombre</b>	RF4.
<b>Resumen</b>	Registra las habitaciones del hotel. Esta operación es realizada por el administrador de datos del hotel.
<b>Entradas</b>	
Número de habitación	
Piso	
Metros cuadrados	
<b>Resultados</b>	
Habitación registrada	
<b>RNF asociados</b>	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que hay que registrar las habitaciones del hotel, lo que significa que toca darle una instrucción al sistema para que persista esa información y se puedan reservar habitaciones en el futuro.

<b>Nombre</b>	RF5.
<b>Resumen</b>	Registra los servicios del hotel, junto con su descripción. Esta operación es realizada por el administrador de datos del hotel.
<b>Entradas</b>	
Nombre	
Descripción	
Hora de inicio	
Hora de cierre	
Características del servicio	
<b>Resultados</b>	
Servicio registrado	
<b>RNF asociados</b>	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que para registrar los servicios del hotel, es necesario ejecutar una operación que guarde esos servicios y lo haga de manera correcta y confiable.

<b>Nombre</b>	RF6.
<b>Resumen</b>	Registra los planes de consumo disponibles en el hotel. Esta operación es realizada por el administrador de datos del hotel
<b>Entradas</b>	
Nombre	
Precio base	
Noches requeridas	
Servicios incluidos	
Tipo de plan	

Resultados
Plan de consumo registrado
RNF asociados
RNF1. Seguridad
RNF2. Privacidad
RNF3. Persistencia

El requerimiento necesita comportamiento transaccional ya que para registrar los planes de consumo del hotel, el sistema debe permitir guardar esa información y utilizarla después.

Nombre	RF7.
Resumen	Reserva una habitación por un período de tiempo, por parte de un cliente, siempre y cuando esté disponible. Esta operación es realizada por un cliente.
Entradas	
Cliente que reserva	
Número de adultos	
Número de niños	
Fecha de llegada	
Fecha de salida	
Plan de consumo asociado	
Método de pago	
Resultados	
Reserva agendada	
RNF asociados	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que se necesitan poder agendar reservas y que estas queden guardadas para que el cliente pueda acceder al hotel en el momento estipulado en la reserva. Si esto no se persiste, es como si nunca hubiera existido reserva.

Nombre	RF8.
Resumen	Reserva la prestación de un servicio por parte de un cliente, siempre y cuando haya disponibilidad. Esta operación es realizada por un cliente.
Entradas	
Cliente que reserva	
Fecha	
Hora	
Duración	
Número de invitados	
Resultados	
Reserva agendada	
RNF asociados	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que se necesitan poder agendar reservas y que estas queden guardadas para que el cliente pueda acceder a los servicios del hotel en el momento estipulado en las reservas. Si esto no se persiste, es como si nunca hubiera existido reserva y el cliente no puede disfrutar de los servicios.

Nombre	RF9.
--------	------

<b>Resumen</b>	Registra la llegada de un cliente al hotel, correspondiente a una reserva ya registrada. Esta operación es realizada por un recepcionista del hotel.
<b>Entradas</b>	
Reserva	
Cliente de la reserva	
Fecha de check in	
<b>Resultados</b>	
Llegada registrada	
<b>RNF asociados</b>	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que se necesitan poder guardar la fecha y hora exacta en la que un cliente llegó al hotel por temas operativos y registro futuro de información.

<b>Nombre</b>	RF10.
<b>Resumen</b>	Registra un consumo de un servicio por parte de un cliente o sus acompañantes. Esta operación es realizada por un empleado del hotel.
<b>Entradas</b>	
Reserva asociada al consumo	
Servicio asociado al consumo	
Valor	
Fecha	
<b>Resultados</b>	
Consumo registrado	
<b>RNF asociados</b>	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que el hotel necesita persistir esta información para poder realizar el cobro adecuado del consumo realizado por el cliente durante su estadía. Sin esto, el cliente puede consumir sin pagar, SIEMPRE.

<b>Nombre</b>	RF11.
<b>Resumen</b>	Registra la salida de un cliente al hotel, con todo lo que eso implica. Esta operación es realizada por un recepcionista del hotel.
<b>Entradas</b>	
Reserva	
Cliente de la reserva	
Fecha de check out	
<b>Resultados</b>	
Salida registrada	
<b>RNF asociados</b>	
RNF1. Seguridad	
RNF2. Privacidad	
RNF3. Persistencia	

El requerimiento necesita comportamiento transaccional ya que se necesitan poder guardar la fecha y hora exacta en la que un cliente salió del hotel por temas operativos y registro futuro de información.

### 3. Modelo de datos relacional

**Hotel**

hotelId	name	description	stars	city	address
PK, SA	NN	NN	NN	FKCity.cityId	NN

**Room**

number	floor	size	hotel	roomType
PK	PK	NN	FKHotel.hotelId	FKRoomType.name

**RoomType**

name	description	capacity	price
PK	NN	NN, CK [price >= 0]	NN, CK [price >= 0]

**Amenity**

name	description
PK	NN

**Offers**

amenity	roomType
PK, FKAmenity.name	PK, FKRoomType.name

**OccupancyDate**

id	date	CK1 [date > 01/01/1850]
PK	NN, CK1	

**AvailableDate**

occupancyDate	room	available
PK, FKOccupancyDate.id	PK, FKRoomType.name	PK

**Booking**

id	numberOfAdults	numberOfChildren	roomNumber	roomFloor	consumptionOffer
PK	NN, CK > 0	NN, CK >= 0	FKRoom.number	FKRoom.floor	FKConsumptionOffer.id

**Payment**

booking	paymentMethod
PK, FKBooking.id	PK, FKPaymentMethod.name

**User**

firstName	lastName	documentType	documentNumber	email	bornDate	role	group
NN	NN	NN	PK	NN	NN, CK1	NN, CK("Cliente", "Receptionist", "Employee",	FKGroup.id

						"DataAdmin", "HotelManager")	
--	--	--	--	--	--	---------------------------------	--

### CheckIn/CheckOut

booking	user	checkInDate	checkInMade	checkOutDate	checkOutMade
PK, FKBooking.id	PK, FKUser.documentNumber	PK, CK1	PK	PK, CK1	PK

### Reservation

id	date	startHour	duration	numberOfGuests	CK2 [ >= 00:00:00]
PK	NN, CK1	NN, CK2	NN, CK2	NN, CK > 0	

### ReservationOfService

reservation	serviceOccupancyHour
PK, FKReservation.id	PK, FKServiceOccupancyHour.id

### Service

id	name	description	startWorkHour	endWorkHour	EmployeeThatRegisterService
PK	NN	NN	NN, CK2	NN, CK > 0	PK, FKUser.documentNumber

### ServiceInvoice

id	value	createdDate	paidDate	paid	service	consumptionAccount
PK	NN	NN, CK1	CK1	NN	FKService.id	FKConsumptionAccount.id

### FoodProduct

id	name	description	price	imgSrc	restaurant
PK	NN	NN	NN, CK >= 0	NN	FKRestaurant.id

### IncludedFood

consumptionOffer	foodProduct	maxNumberOfIncludedItems	productDiscount
PK, FKConsumptionOffer.id	PK, FKFoodProduct.id	PK	PK

### Restaurant

id	name	type	foodType	musicType
PK	NN	NN, CK("Bar", "Restaurant")	NN	NN

### ConvenienceService

ConveniencelsService	type
PK, FKService.id	NN, CK("Pool", "Gym", "Internet", "Supermarket", "Store", "Laundry", "UtensilsLoan")

### GastronomyService

GastronomyIsService	description	restaurant
PK, FKService.id	NN	FKRestaurant.id

Figura 2. Tablas del modelo de datos relacional (2)

#### a. Nivel de normalización

La normalización sirve para eliminar los datos redundantes y garantizar que los datos se almacenen de forma lógica. Para encontrar el nivel de normalización se analizan las entidades:

Primera forma normal: Se cumple porque los atributos son atómicos, los valores en cada columna son del mismo dominio, los nombres de las columnas son únicos y el orden en el que se guardan los datos no importa.

**Hotel** (hotelId, name, description, stars, city, address)

Llaves: {hotelId}

Primos: {hotelId}

No Primos: {name, description, stars, city, address}

Segunda forma normal: Se cumple porque está en la primera forma normal y no tiene dependencia parcial desde los atributos primos. Es decir, no dependen de los no primos

Tercera forma normal: Se cumple porque está en la segunda forma normal y no tiene dependencia transitiva. Es decir, ningún atributo no primo depende de otro atributo no primo.

Forma normal Boyce-Codd: Encontramos que se cumple que por cada dependencia  $A \rightarrow B$ ,  $A$  sea una superllave. Por esto, el modelo cumple BCNF.

Para todas las entidades del modelo se cumple, como mínimo, BCNF.

## 4. Resultados logrados

Se logró implementar satisfactoriamente todos los requerimientos funcionales y no funcionales.

Las entidades y las reglas de negocio se cumplen en el modelo. Por ejemplo, un hotel tiene distintos tipos de habitaciones que cuentan con distintas comodidades. Por otro lado, se tiene los detalles de los servicios adicionales prestados y sus restricciones mediante ServiceCategoryType y ServiceType. Cuando se requiera agregar a la cuenta de la habitación un consumo de un servicio, se puede hacer mediante la relación entre Room, Booking y ServiceInvoice. También, se puede observar que los hoteles ofrecen la opción de que en sus habitaciones haya varios tipos de planes de consumo, de acuerdo al enunciado.

#### a. Plan de pruebas

Al insertar un registro con la misma PK sale error:



```
INSERT ALL
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (1, 1, 1, '505', '40 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (2, 1, 2, '607', '50 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (3, 1, 3, '509', '70 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (4, 1, 3, '302', '70 m2')
SELECT * FROM DUAL;

INSERT ALL
  INTO A_OCCUPANCYDATE (ID, "DATE") VALUES (1, TO_DATE('2022/05/03', 'yyyy/mm/dd'))
  INTO A_OCCUPANCYDATE (ID, "DATE") VALUES (2, TO_DATE('2022/05/04', 'yyyy/mm/dd'))
  INTO A_OCCUPANCYDATE (ID, "DATE") VALUES (3, TO_DATE('2022/05/05', 'yyyy/mm/dd'))
  INTO A_OCCUPANCYDATE (ID, "DATE") VALUES (4, TO_DATE('2022/05/06', 'yyyy/mm/dd'))
SELECT * FROM DUAL;

INSERT ALL
```

Script Output x

Task completed in 0.051 seconds

```
Error starting at line : 38 in command -
INSERT ALL
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (1, 1, 1, '505', '40 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (2, 1, 2, '607', '50 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (3, 1, 3, '509', '70 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (4, 1, 3, '302', '70 m2')
SELECT * FROM DUAL
Error report -
ORA-00001: unique constraint (ISIS2304C17202210.A_ROOM_PK) violated
```

Pruebas de integridad con FK sin la tabla referenciada:

```
INSERT ALL
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (1, 1, 1, '505', '40 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (2, 1, 2, '607', '50 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (3, 1, 3, '509', '70 m2')
  INTO A_ROOM (ID, IDHOTEL, IDROOMTYPE, "NUMBER", "SIZE") VALUES (4, 1, 3, '302', '70 m2')
SELECT * FROM DUAL;

INSERT INTO A_BOOKING (ID, IDROOM, NUMBEROFADULTS, NUMBEROFCHILDREN, "TYPE", BASEPRICE, MINREQUIREDNIGHTS, ACCOMODATIONPRICE, ACCOMMODATIONDISCOUNT, PAYMENTPLAN, TOTALVALUE, PAIDDATE, PAID)
VALUES (7, 5, 2, 2, 'ALLINCLUSIVE', 350, 3, 100, 40, 'Payment plan', 410, TO_DATE('2022/05/06', 'yyyy/mm/dd'), 1)
```

Script Output x

Task completed in 0.041 seconds

```
Error starting at line : 8 in command -
INSERT INTO A_BOOKING (ID, IDROOM, NUMBEROFADULTS, NUMBEROFCHILDREN, "TYPE", BASEPRICE, MINREQUIREDNIGHTS, ACCOMODATIONPRICE, ACCOMMODATIONDISCOUNT, PAYMENTPLAN, TOTALVALUE, PAIDDATE, PAID)
VALUES (7, 5, 2, 2, 'ALLINCLUSIVE', 350, 3, 100, 40, 'Payment plan', 410, TO_DATE('2022/05/06', 'yyyy/mm/dd'), 1)
Error report -
ORA-00001: unique constraint (ISIS2304C17202210.A_BOOKING_PK) violated

Error starting at line : 8 in command -
INSERT INTO A_BOOKING (ID, IDROOM, NUMBEROFADULTS, NUMBEROFCHILDREN, "TYPE", BASEPRICE, MINREQUIREDNIGHTS, ACCOMODATIONPRICE, ACCOMMODATIONDISCOUNT, PAYMENTPLAN, TOTALVALUE, PAIDDATE, PAID)
VALUES (7, 5, 2, 2, 'ALLINCLUSIVE', 350, 3, 100, 40, 'Payment plan', 410, TO_DATE('2022/05/06', 'yyyy/mm/dd'), 1)
Error report -
ORA-02291: integrity constraint (ISIS2304C17202210.FK_A_ROOM) violated - parent key not found
```

Cumple con restricciones de chequeo:

```
INSERT INTO A_USER ("DOCUMENT", DOCUMENTTYPE, "NAME", EMAIL, BORNDATE, "ROLE")
VALUES (10028263505, 'Cedula de Ciudadania', 'Pedro Gonzales', 'email@email.com', TO_DATE('2000/05/03', 'yyyy/mm/dd'), 'CLIENT');
```

Script Output x

Task completed in 0.023 seconds

1 row inserted.

No cumple con restricciones de chequeo:

```
INSERT INTO A_USER ("DOCUMENT", DOCUMENTTYPE, "NAME", EMAIL, BORNDATE, "ROLE")
VALUES (10028263505, 'Cedula de Ciudadania', 'Pedro Gonzales', 'email@email.com', TO_DATE('2000/05/03', 'yyyy/mm/dd'), 'CLIENT');
```

Script Output x

Task completed in 0.031 seconds

1 row inserted.

Error starting at line : 11 in command -

```
INSERT INTO A_USER ("DOCUMENT", DOCUMENTTYPE, "NAME", EMAIL, BORNDATE, "ROLE")
VALUES (10028263505, 'Cedula de Ciudadania', 'Pedro Gonzales', 'email@email.com', TO_DATE('2000/05/03', 'yyyy/mm/dd'), 'CLIENT')
```

Error report -

ORA-02290: check constraint (ISIS2304C17202210.CK\_U\_ROLE) violated