

Iteración 2

Mariana Díaz Arenas, Tomas Ángel Gallón
 HotelAndes
 Universidad de los Andes, Bogotá, Colombia
 {m.diaza2, t.angel}@uniandes.edu.co
 Fecha de presentación: Mayo 8 de
 2022

Tabla de contenido

1 Introducción	1
1.1 Ajustes al modelo inicial	1
1.2 Listado de tablas	2
1.3 Implementación nuevos requerimientos	3
2 Modelos	4
2.1 Modelo conceptual	4
2.2 Modelo relacional	5
3 Requerimientos funcionales de modificación	9
3.1 RF10- Registrar un consumo de un hotel por parte de un cliente	9
3.2 RF11- Registrar la salida de un cliente	9
3.3 RF12- Reservar alojamiento y consumos para una convención	10
3.4 RF13- Cancelar reservas asociadas a una convención	10
3.5 RF14- Registrar el fin de una convención	11
4 Requerimientos funcionales de consulta	12
4.1 RFC5- Mostrar el consumo de HotelAndes por un usuario dado	12
4.2 RFC6 - Analizar la operación de HotelAndes	12
5 Pruebas	13
6 Conclusiones	14

1 Introducción

El propósito de este documento es documentar la iteración número 3 de la aplicación HotelAndes para la clase de Sistemas Transaccionales. Los próximos puntos mostrarán los modelos conceptuales y relacionales con los cuales el equipo trabajó para cumplir con los requerimientos funcionales y de consulta. Estos mismos modelos fueron implementados en la base de datos *Oracle*. *(Leer el archivo instrucciones.txt en la carpeta data del jdo para claridad)*

1.1 Ajustes al modelo inicial

Para esta iteración se debía implementar la funcionalidad de poder realizar convenciones al interior del hotel, de tal manera que se pudiesen realizar las reservas respectivas a las habitaciones y a los servicios que requiriera la convención. Adicionalmente, todas estas transacciones debían ser manejadas por otro tipo de usuario que debía ser introducido “OrganizadorEventos”, el cual manejaría el plan de pago ofrecido a las personas de la convención y lo que se requeriría.

Para poder agregar esto al modelo, se debió incluir una nueva tabla al modelo relacional, denominada “Convención”, la cual, incluiría la información relevante de la convención, como su nombre, organizador y el intervalo de fechas en el que se iba a desarrollar. Así mismo, se debió modificar el check de la tabla TipoUsuario para poder incluir el nuevo rol de usuario que manejara las convenciones. Adicionalmente agregamos una nueva tabla Usuario para manejar de manera más fácil la información de los empleados y los clientes. Finalmente, para un mejor uso de los consumos de las habitaciones y los servicios que se manejan dentro del hotel, se reestructuraron las tablas de Factura y se agregó la de Consumo para poder llegar al BCNF.

Adicionalmente, se agregaron atributos a varias tablas, como las habitaciones, servicios, usuarios, etc. Para poder facilitar los requerimientos de consulta y almacenar la información de estadía de los clientes y sus gastos totales a lo largo de su estadía en HotelAndes. Así mismo, poder establecer si una habitación estaba reservada o no para poder ofrecer en disponibilidad para el hotel o mantenerla para un cliente o convención.

1.2 Listado de Tablas

El listado de las tablas incluídas en la iteración fue obtenido mediante la siguiente sentencia en SQL:

```
SELECT  tabs.TABLE_NAME  AS  NombreTabla,  tabs.COLUMN_NAME  AS  NombreColumna,
tabs.DATA_TYPE AS TipoDeDato,
        DECODE(re.CONSTRAINT_NAME,  null,  'NO TIENE',re.CONSTRAINT_NAME) AS
NombreRestriccion
FROM (SELECT *
      FROM ALL_TAB_COLUMNS
      WHERE owner = 'ISIS2304A08202210') tabs
LEFT OUTER JOIN
      (SELECT *
      FROM ALL_CONS_COLUMNS
      WHERE owner = 'ISIS2304A08202210') re
ON tabs.COLUMN_NAME = re.COLUMN_NAME
AND tabs.TABLE_NAME = re.TABLE_NAME
ORDER BY tabs.TABLE_NAME ASC, tabs.COLUMN_NAME ASC, re.CONSTRAINT_NAME ASC;
```

De acuerdo a lo anterior, las tablas del modelo son:

1.3 Implementación Nuevos Requerimientos:

RF12: Se debe ingresar el nombre de la convención de la cual se van a realizar las reservas a las habitaciones y a los servicios. Posteriormente, se ingresan la cantidad de habitaciones de cada tipo que se quiere solicitar, junto con un listado de los servicios deseados. A partir de ello, se busca si la convención existe efectivamente en la base de datos, posteriormente, para cada tipo de habitación se verifica si hay suficientes habitaciones disponibles de cada tipo mediante una consulta sql y determinando la cantidad que hay para realizar las reservas y se le informa al usuario cuales reserva de que tipo de habitación se pudieron hacer y cuales no, se realiza el mismo proceso para los servicios.

RF13: En este paso, el proceso de quitar una reserva se realiza eliminando las reservas correspondientes. Para ello, se solicita el nombre de la convención para verificar si existe en la base de datos y se solicita el número de las habitaciones que se quieren dejar sin reserva y lo mismo para los servicios, se procede a eliminar dichas reservas.

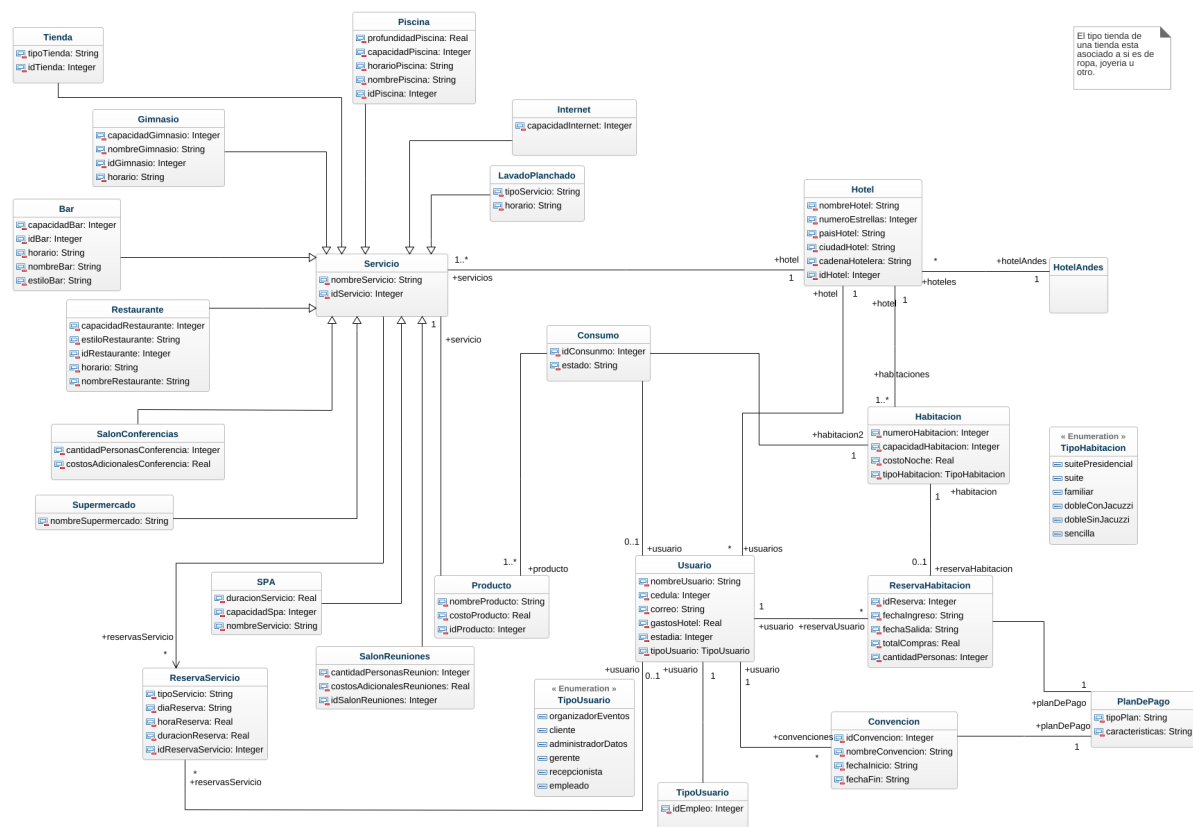
RF14: Al finalizar la convención, se solicita únicamente el nombre de la convención y se va a

realizar una búsqueda de todas las habitaciones de la convención, todos los consumos hechos por los clientes en cada habitación y se va a realizar una factura para cada uno, para mostrar al final el total que se debe pagar por todos los miembros de la convención.

** Los requerimientos 15 y 16 se implementaron parcialmente

2 Modelo conceptual y relacional

2.1 Modelo conceptual



2.2 Modelo relacional

Bar

idServicio	idBar	nombreBar	horario	capacidadBar	estiloBar
FK(servicio.id)	PK, SA	NN, UA, ND	CK('DIURNO','NOCTURNO','COMPLETO'), NN	UA	UA

Consumo

idConsumo	idServicio	habitacion	producto	estado
PK, SA	FK(servicio.id)	FK(hotel.idHotel), UA	FK(producto.idProducto)	NN, UA, CK('PAGO','PENDIENTE')

Convención

idConvencion	nombreConvencion	organizador	fechaInicio	fechaFin	planPago
PK, SA	NN, UA	FK(usuario.cedula)	NN< UA	NN, UA	FK(planPago.tipoPlan)

Factura

idFactura	habitacion	montoTotal
PK, SA	FK(habitacion.numHabitacion)	NN

Gimnasio

idServicio	idGimnasio	capacidadGimnasio	nombreGimnasio	horario
FK(servicio.id)	PK	NN, UA	UA, NN, ND	CK('DIURNO','NOCTURNO','COMPLETO'), NN

TipoHabitacion

idTipoHabitacion	tipoHabitacion
PK	NN, ND

Habitacion

numHabitacion	capacidadHabitacion	costoNoche	tipoHabitacion	idHotel	estado	cliente
PK	NN, UA	UA, NN	FK(tipoHabitacion.idTipoHabitacion), NN, CK('SUITEPRESIDENCIAL', 'SUITE', 'FAMILIAR', 'DOBLECONJACUZZI', 'SENCILLACONJACUZZI', 'SENCILLA')	FK(hotel.idHotel), NN	CK('OCUPADA','VACIA','RESERVADA'), NN	FK(usuario.cedula)

Hotel

idHotel	nombreHotel	numeroEstrellas	paisHotel	ciudadHotel	cadenaHotel
PK	UA, NN, ND	UA	UA, NN	UA, NN	UA, NN

Internet

capacidadInternet	idServicio
UA, NN	FK(servicio.id)

Lavado/Planchado

tipoServicio	horario	idServicio
PK	CK('DIURNO','NOCTURNO','COMPLETO'), NN	FK(servicio.id)

Piscina

idServicio	idPiscina	capacidadPiscina	profundidad	horario	nombrePiscina
FK(servicio.id)	PK	NN, UA	NN, UA	CK('DIURNO','NOCTURNO','COMPLETO'), NN	UA, NN, ND

PlanDePago

tipoPlan	caracteristicas
PK	NN, UA

Producto

id	idServicio	costoProducto	nombreProducto
PK	FK(servicio.id)	UA, NN	NN

ReservaHabitacion

idReserva	fechaIngreso	fechaSalida	cantidadPersonas	planPago	cedulaCliente	numHabitacion	totalCompras
PK	UA, NN	UA, NN	UA, NN	FK(planDePago.tipoPlan)	FK(usuario.cedula)	FK(habitacion.numHabitacion)	NN

ReservaServicio

id	tipoServicio	diaReserva	horaRegreso	duracion	cedulaCliente	idServicio
PK	NN, UA	NN, UA	NN, UA	UA, NN	FK(usuario.cedula)	FK(servicio.id)

Restaurante

idServicio	idRestaurante	nombreRestaurante	capacidadRestaurante	horario	estiloRestaurante
FK(servicio.id)	PK	NN, UA, ND	NN, UA	CK('DIURNO','NOCTURNO','COMPLETO'), NN	NN, UA

SalonesReuniones

idSalonReuniones	nombreSalonReuniones	capacidadSalonReuniones	idServicio
PK	NN, UA, ND	NN, UA	FK(servicio.id)

SalonesConferencias

idSalonConferencias	nombreSalonConferencias	capacidadSalonConferencias	idServicio
PK	NN, UA, ND	NN, UA	FK(servicio.id)

Servicios

id	nombreServicio	demanda
PK	NN, UA, ND	

Spa

idSpa	idServicio	horario	nombreSpa	capacidadSpa
PK	FK(servicio.id)	CK('DIURNO','NOCTURNO','COMPLETO'), NN	NN, UA, ND	NN, UA

Supermercado

idSupermercado	idServicio	horario	nombreSupermercado
PK	FK(servicio.id)	CK('DIURNO','NOCTURNO','COMPLETO'), NN	NN, UA, ND

Tienda

idTienda	idServicio	nombreTienda	tipoTienda	horario
PK	FK(servicio.id)	NN, UA, ND	NN, UA	CK('DIURNO','NOCTURNO','COMPLETO'), NN

TipoUsuario

idEmpleo	tipoEmpleo
PK	NN, UA, ND, CK('RECEPCIONISTA', 'EMPLEADO', 'ADMINISTRADORDATOS', 'GERENTE', 'ORGANIZADOREVENTOS', 'CLIENTE')

Usuario

tipoUsuario	nombreUsuario	cedulaUsuario	correoUsuario	idHotel	gastosHotel	estadia
FK(tipoUsuario.idEmpleo)	NN, UA	PK, UA	NN, ND, UA	FK(hotel.idHotel), UA	NN	UA

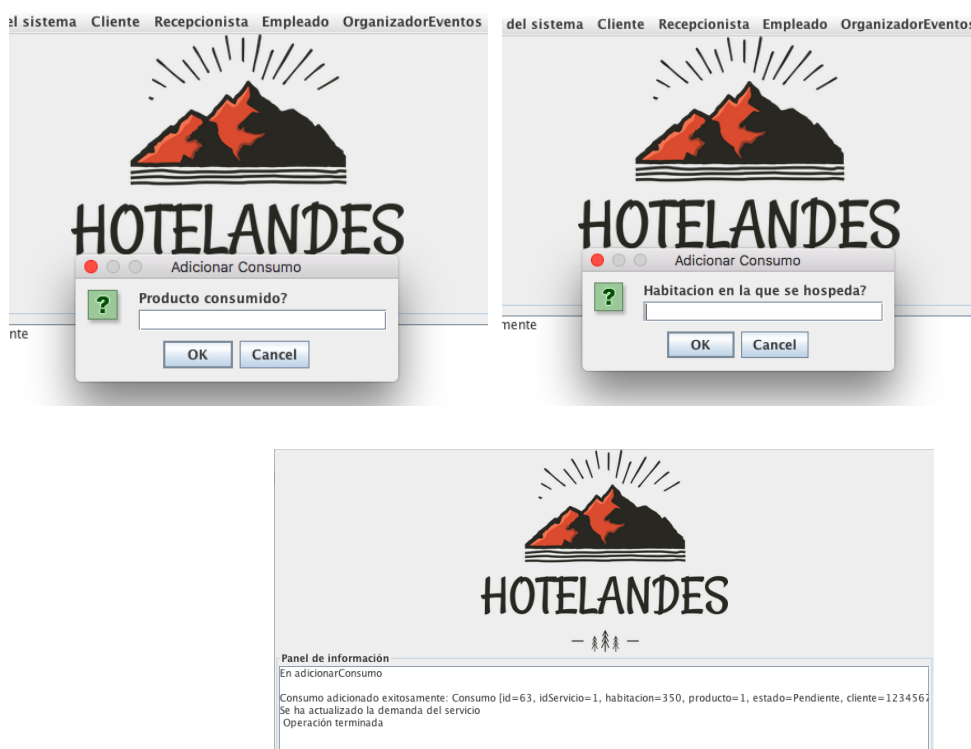
3 Requerimientos funcionales de modificación

Para poder realizar los requerimientos funcionales que se verán a continuación, fue importante implementar a la aplicación la posibilidad de crear una convención por parte de un organizador de eventos. Para hacer esto, la persona tiene que ingresar a la misma dando su nombre y contraseña y el rol que ocupa dentro del negocio.

IMPORTANTE: Se puede crear una convención desde el jdo para poder hacer los otros RF

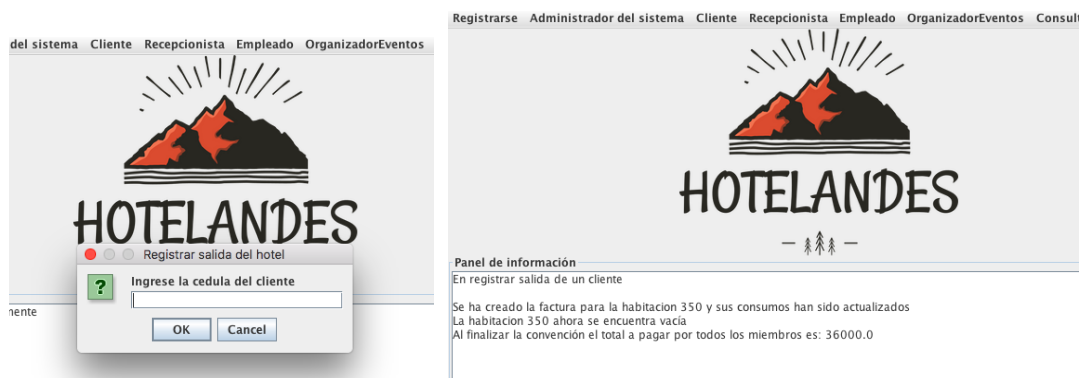
3.1) RF10- Registrar un consumo de un hotel por parte de un cliente o sus acompañantes

La implicación de este requerimiento fue satisfactoria ya que en primera instancia, tiene que iniciar sesión como un Empleado (tiene que existir un usuario en la base de datos con este tipo), posteriormente solicita el producto consumido (puede ser de cualquier servicio) y la habitación en la que se está hospedando. Finalmente, va a mostrar la información del consumo realizado. Se va a verificar la existencia de la habitación y del producto en la base de datos (La demanda del servicio se actualiza cada vez que se hace un consumo)



3.2) RF11- Registrar la salida de un cliente

La implicación de este requerimiento fue satisfactoria ya que en primera instancia, tiene que iniciar sesión como un Recepcionista (tiene que existir un usuario en la base de datos con este tipo), posteriormente solicita únicamente la cédula del cliente y va a buscar todas las habitaciones a nombre de este cliente y coleccionar los consumos e informar el costo total al cliente

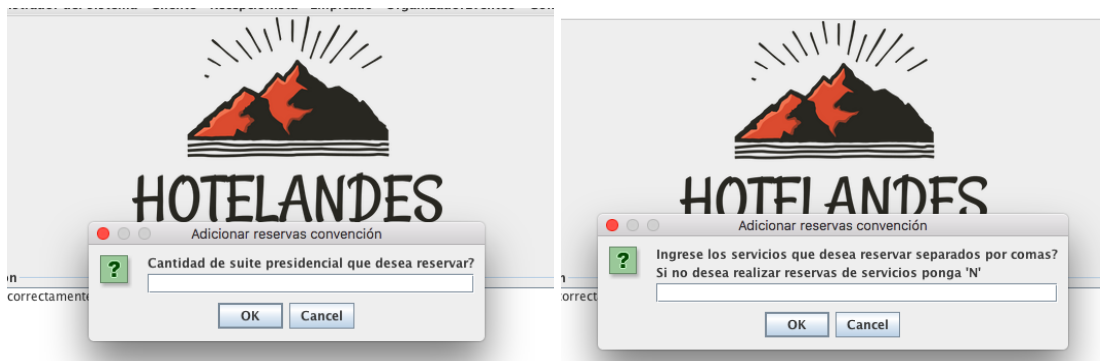


3.3) RF12- Reservar alojamiento y consumos para una convención

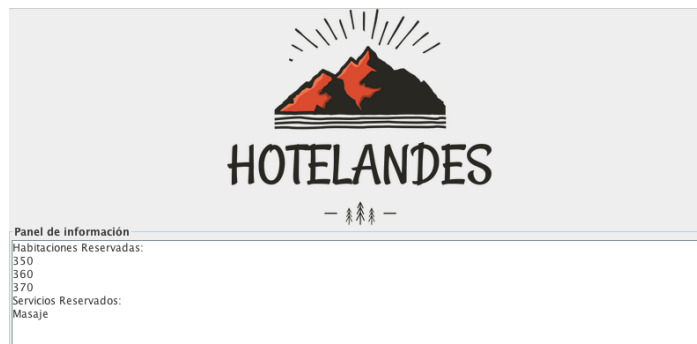
La implicación de este requerimiento fue satisfactoria ya que en primera instancia, tiene que iniciar sesión como un OrganizadorEventos (tiene que existir un usuario en la base de datos con este tipo), posteriormente solicita el nombre de la convención sobre la que se va a trabajar



A partir de ello, se busca en la base de datos y se va a solicitar la cantidad de cada tipo de habitación que se desea y una lista de cuales son los servicios que se desea reservar (estos son productos dentro de la base de datos y se verifica su existencia)



Finalmente, mostrará una lista de las habitaciones y servicios reservados (solo se solicita por cada tipo de habitación si hay las suficientes de cada tipo)

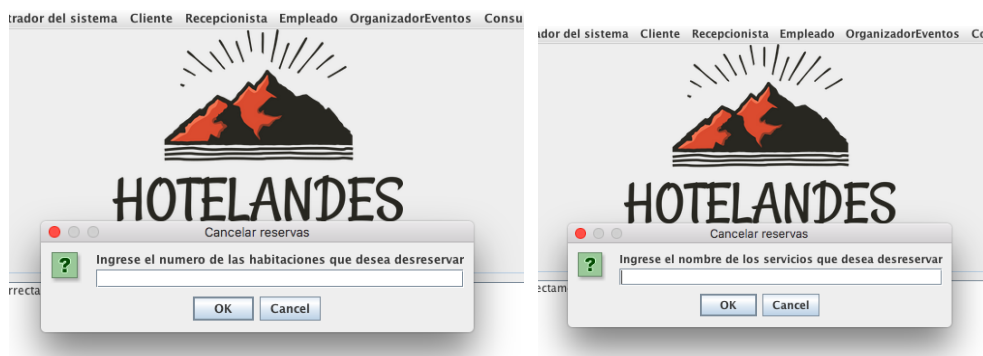


3.1) RF13- Cancelar reservas asociadas a una convención

La implicación de este requerimiento fue satisfactoria ya que en primera instancia, tiene que iniciar sesión como un OrganizadorEventos (tiene que existir un usuario en la base de datos con este tipo), posteriormente solicita el nombre de la convención sobre la que se va a trabajar



Después se solicita cuáles son las habitaciones y servicios específicos de los que se quiere eliminar la reserva de la convención



Finalmente, se realiza la eliminación de las reservas solicitadas



3.1) RF14- Registrar el fin de una convención

La implicación de este requerimiento fue satisfactoria ya que en primera instancia, tiene que iniciar sesión como un OrganizadorEventos (tiene que existir un usuario en la base de datos con este tipo), posteriormente solicita el nombre de la convención sobre la que se va a trabajar



Para este requerimiento no se solicita más información y únicamente para todas las habitaciones que estén a nombre del organizador de eventos de la convención, revisa los consumos por habitación, genera las facturas individuales y muestra el total a pagar, actualizando el estado de las habitaciones y los consumos.



4 Requerimientos funcionales de consulta

4.1) RFC5- Mostrar el consumo de HotelAndes por un usuario dado, en un rango de fechas indicado:

Este requerimiento fue posible de hacer, realizando la consulta en una fecha dada y un usuario dado, sin embargo, no se implementó en el jdo debido a que no existía una clase con los parámetros dados con la que fuese posible retornar el valor, por lo tanto, se presenta la sentencia SQL usada, con los valores de prueba y se muestra el valor retornado

```
SELECT nombre, SUM(COSTOPRODUCTO) AS TOTALCONSUMO
FROM consumo, producto, usuario
WHERE consumo.producto = producto.id
  AND fechaConsumo = '2022-02-04'
  AND cedula = 12345678
GROUP BY nombre;
```

	NOMBRE	TOTALCONSUMO
1	Pedro	42000

4.2) RFC6- Analizar la operación de HotelAndes:

Este requerimiento fue posible de hacer realizando la consulta con dos sentencias diferentes dado que se quiere encontrar el máximo y mínimo valor de servicios adquiridos para una fecha dada. La unidad de tiempo que se escogió para este requerimiento fue día dado que las fechas en la aplicación están dadas por strings y es difícil su manejo. Las sentencias son las mismas a diferencia de la forma en cómo se organizan; ascendente o descendente, dependiendo de lo que se busque. En este caso, se escogió una fecha en específico para mostrar cual fue el mínimo consumo de un servicio.

```
SELECT nombreservicio, sum(servicio.demanda)
FROM servicio, consumo
WHERE servicio.id = consumo.idservicio
  AND consumo.fechaconsumo = '2022-02-04'
GROUP BY servicio.nombreservicio
ORDER BY sum(servicio.demanda) ASC;
```

```
SELECT nombreservicio, sum(servicio.demanda)
FROM servicio, consumo
WHERE servicio.id = consumo.idservicio
  AND consumo.fechaconsumo = '2022-02-04'
GROUP BY servicio.nombreservicio
```

```
ORDER BY sum(servicio.demanda) DESC;
```

	NOMBRESERVICIO		SUM(SERVICIO.DEMANDA)	
1	Bar		16	

5 Pruebas

*El archivo de pruebas en excel está incluido en la carpeta data del jdo

Registrarse		
Datos entrados:	Tipo de usuario:	OrganizadorEventos
	Nombre:	Pablo
	Contraseña:	pablo

Requerimiento RF12		
Realizada por:	OrganizadorEventos(Pablo)	
Datos de entrada:	Convencion a la que pertenece:	Marvel
	Suite Presidenciales:	3
	Suite:	2
	Familiar:	0
	Doble con jacuzzi:	1
	Sencilla sin jacuzzi:	0
	Sencillas:	3
	Servicios:	Restaurante,Spa
Requerimiento RF13		
Realizada por:	OrganizadorEventos(Pablo)	
Datos de entrada:	Convencion a la que pertenece:	Marvel
	Habitaciones por desreservar:	3
	Servicios por desreservar:	Masaje
Requerimiento RF14		
Realizada por:	OrganizadorEventos(Pablo)	
Datos de entrada:	Convencion a la que pertenece:	Marvel

6 Conclusiones

1. Mantener una buena arquitectura de proyecto nos permite dividirlo por capas y asignar una responsabilidad a cada una.
2. Es importante desarrollar un buen modelo relacional para asegurar que la aplicación funcione de la manera más óptima y la implementación de los requerimientos funcionales sea sencilla.
3. La integridad y persistencia de los datos son importantes para mantener una base de datos funcional y con datos de calidad.
4. Fue necesario alterar tablas pero facilitaron la implementación de los requerimientos, es decir, que se logró llegar al BCNF
5. Se agregaron las nuevas reglas de negocio correspondientes para agregar la clase de convención