

PRUEBA TÉCNICA DESARROLLADOR FULLSTACK

Lee la prueba y teniendo en cuenta tu disponibilidad de tiempo, infórmalos el día y hora en la que máximo enviarás las respuestas. Nota que en el transcurso del tiempo podrás enviar preguntas que serán respondidas con la mayor prontitud. Si por cualquier motivo no puedes terminar toda la prueba en el tiempo estipulado, envía las partes que tengas y el motivo por el cual no pudiste completarla. Una adición al tiempo de entrega podría ser otorgada.

El resultado de la prueba debe ser subido a un repositorio como GitHub. Solo debes enviarnos el enlace.

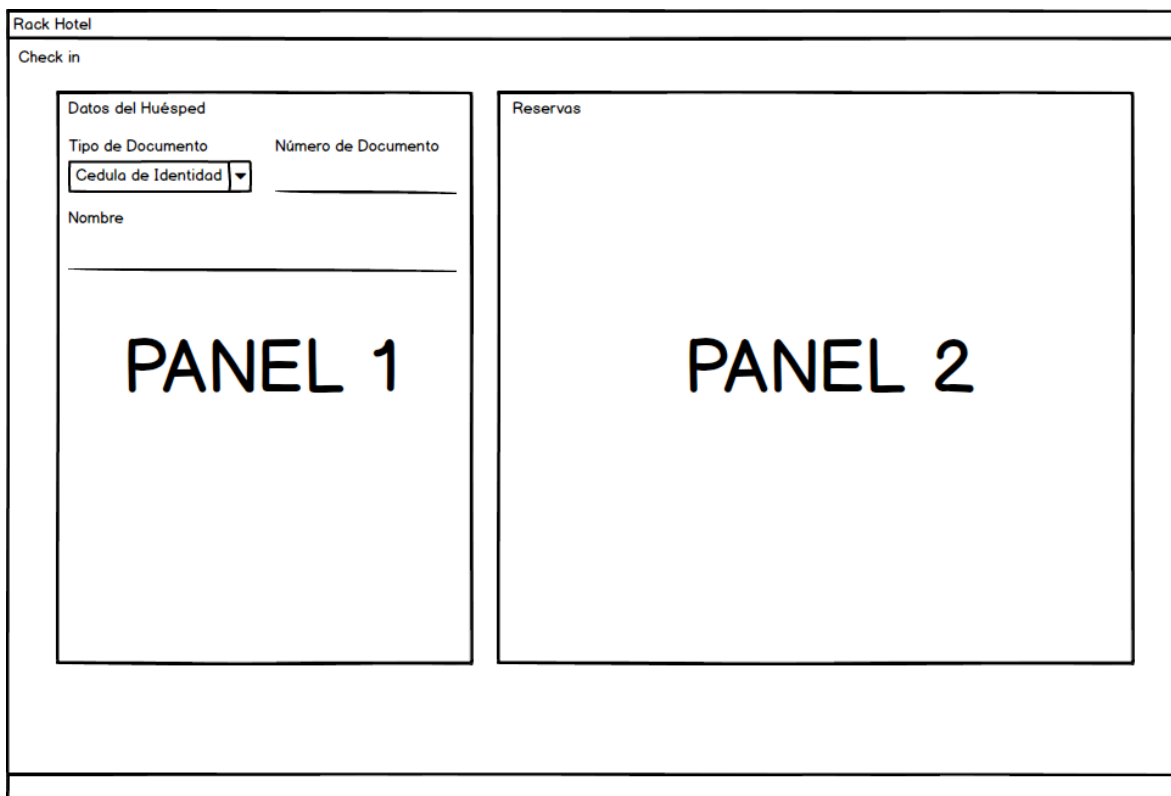
Debe estar desarrollado con el framework Laravel. Esta prueba técnica es un problema que ya hemos resuelto. Si ves alguna mejora en cuanto estructura u otra cosa, tus comentarios son bien recibidos.



ESTRICTAMENTE CONFIDENCIAL

La prueba consiste en efectuar el Check-in en un hotel. Para realizar un Check-in, debe existir un huésped que tenga una reserva para el día en que hace Check-in o en caso de que no tenga reserva, el hotel debe contar con habitaciones disponibles para crear una nueva reserva para alojar al huésped.

Esta vista se presenta en dos paneles, tal y como se muestra a continuación:



The screenshot shows a web application titled "Rack Hotel" with a "Check in" section. It is divided into two main panels:

- PANEL 1 (Left):** Contains the "Datos del Huésped" (Guest Data) form. It includes a "Tipo de Documento" (Document Type) dropdown menu with "Cedula de Identidad" selected, a "Número de Documento" (Document Number) input field, and a "Nombre" (Name) input field.
- PANEL 2 (Right):** Contains the "Reservas" (Reservations) section, which is currently empty.

Panel 1

Panel 1 o panel izquierdo contiene la información del huésped.

Este panel se utiliza para buscar a un huésped almacenado en base de datos, en caso de no encontrarlo, entonces se registra.

ESTRICTAMENTE CONFIDENCIAL

La búsqueda se realiza por documento de identidad: Se debe seleccionar el tipo de documento de identidad y se debe escribir el número de documento, luego al hacer click en otro lugar que no sea la caja de texto donde se escribe el número de documento, el formulario debe cargar los datos del huésped si se encuentra registrado. En caso de que el huésped se encuentre registrado, solo se muestran los datos, no se debe poder modificar sus datos.

En caso de que el huésped no esté registrado, se debe cargar el formulario para introducir los datos.


Los huéspedes se clasifican en dos tipos: huéspedes personales (personas naturales, cuyo tipo de documento de identidad es cédula de identidad o cédula de extranjería) y huéspedes empresariales (cuyo tipo de documento de identidad es compañía). Para cada tipo de huésped se muestra un formulario diferente, como se ilustra a continuación:



Datos del Huésped

Tipo de Documento Número de Documento
 Cedula de Identidad _____

Título Nombre Apellido
 Sr. _____

Sexo Fecha de Nacimiento
☐ Femenino 28/02/1919 
☐ Masculino

Ocupación Correo Electrónico

Tipo de Teléfono Número de Teléfono
 Celular _____

Dirección

País Estado Ciudad
 Venezuela Merida Merida

Datos del Huésped

Tipo de Documento Número de Documento
 Compañía _____

Nombre

Correo Electrónico

Tipo de Teléfono Número de Teléfono
 Celular _____

Dirección

País Estado Ciudad
 Venezuela Merida Merida

ESTRICTAMENTE CONFIDENCIAL

El formulario en el panel izquierdo pertenece a los huéspedes personales y el formulario en el panel derecho pertenece a los huéspedes empresariales, en resumen, dependiendo del tipo de identificación que se seleccione se muestra uno u otro formulario. (No registrar el número de teléfono en ambos casos).

Panel 2

El panel 2 o panel derecho contiene la información de la reserva. Este panel depende del panel 1. Se muestra un formulario para generar una nueva reserva.

Rack Hotel

Check in

Datos del Huésped

Tipo de Documento Número de Documento

Cedula de Identidad

Título Nombre Apellido

Sr.

Sexo Fecha de Nacimiento

☐ Femenino
☐ Masculino

28/02/1919

Ocupación Correo Electrónico

Tipo de Teléfono Número de Teléfono

Celular

Dirección

País Estado Ciudad




Venezuela

Merida

Merida

Reservas

Fecha de Entrada 01/02/2017 Fecha de Salida 28/02/2017

Habitaciones	Acciones
 Nombre de Habitación Descripción de la Habitación Número Mínimo de Personas: 1 Número Máximo de Personas: 2	Cantidad 3 Agregar
 Nombre de Habitación Descripción de la Habitación Número Mínimo de Personas: 1 Número Máximo de Personas: 2	Cantidad 1 Agregar
 Nombre de Habitación Descripción de la Habitación Número Mínimo de Personas: 1 Número Máximo de Personas: 2	No hay costos Registrados

Tipo de habitación	Costo	Cantidad
Suite Junior ✕	01/02/2017 - 13/02/2017 Estandar - 150.000 COP	3
	14/02/2017 Amor - 250.000 COP	
	15/02/2017 - 26/02/2017 Estandar - 150.000 COP	
	27/02/2017 - 28/02/2017 Carnaval - 170.000 COP	
Total: 4'020.000 COP / Habitación		12'060.000 COP

Reservar

Tabla 1 (Habitaciones disponibles, tabla superior)

Se muestra un formulario con la fecha de entrada y salida (La fecha de entrada debe ser el día actual y no puede cambiarse, la fecha de salida debe mostrar un selector de fecha para seleccionar, por defecto estará el día siguiente al día actual, se debe validar que la fecha de salida sea después que la fecha de entrada). Al presionar el botón buscar (lupa) se muestra una tabla con los tipos de habitación disponibles para la fecha seleccionada.

Por cada tipo de habitación se muestra:

- Foto del tipo de habitación: Se debe colocar la primera foto seleccionada.
- Nombre del tipo de habitación.
- Descripción del tipo de habitación.
- Número mínimo y máximo de personas que pueden ocupar esa habitación.
- En acciones: Cantidad disponible del tipo de habitación (Seleccionador) y el botón agregar. En caso de que la habitación no posea costos asociados a la fecha requerida, no se debe mostrar las acciones, si no un mensaje de advertencia "No hay costos registrados".
- El botón agregar añadirá el tipo de habitación y la cantidad seleccionada a la tabla de inferior. Una vez agregado el tipo de habitación se debe descontar la cantidad del tipo de habitación que se añadió, en caso de que la cantidad sea cero, el tipo de habitación debe desaparecer de la tabla de habitaciones disponibles. Si se añade de nuevo un mismo tipo de habitación, en la tabla 2 no debe repetirse sino aumentar su cantidad.

Tabla 2 (Habitaciones seleccionadas, tabla inferior)

En la segunda tabla se muestran las habitaciones seleccionadas para la reserva.

Mientras no hayan registros en la tabla, no se debe mostrar el esqueleto de la tabla y el campo Total debe ir en cero.

La tabla tiene las siguientes columnas:

Columna	Descripción
Habitación	Nombre del tipo de habitación, seguido de un icono "eliminar". Esta opción permite eliminar el tipo de habitación seleccionado. Al eliminar se debe reestablecer la cantidad del tipo de habitación eliminada en la tabla superior.

Costo	Indica el costo por habitación
Cantidad	Cantidad del tipo de habitación seleccionado

- Campo total: Se muestran dos cantidades totales. Una debajo de la columna costos que indica la sumatoria de todos los costos seleccionados en dicha columna. Este total hace alusión al dinero a pagar por UNA habitación durante el periodo de tiempo seleccionado, lo que quiere decir que cada costo seleccionado debe multiplicarse por la cantidad de días incluidos dentro del rango al cual pertenece el costo. El segundo total mostrado aparece debajo de la columna cantidad, el cuál no es más que la multiplicación del costo total por la cantidad.
- Para mostrar el costo por cada tipo de habitación se recomienda ver con detenimiento la siguiente ilustración:

Costos para tipo de habitación: Suite Junior

Fecha Inicio	Fecha Fin	Nombre	Precio
01/02/2017	28/02/2017	Estandar	150.000
14/02/2017	14/02/2017	Amor	250.000
27/02/2017	28/02/2017	Carnaval	170.000

Reserva: Desde 01/02/2017 Hasta el 28/02/2017

01/02/2017 - 13/02/2017

14/02/2017

15/02/2017 - 26/02/2017

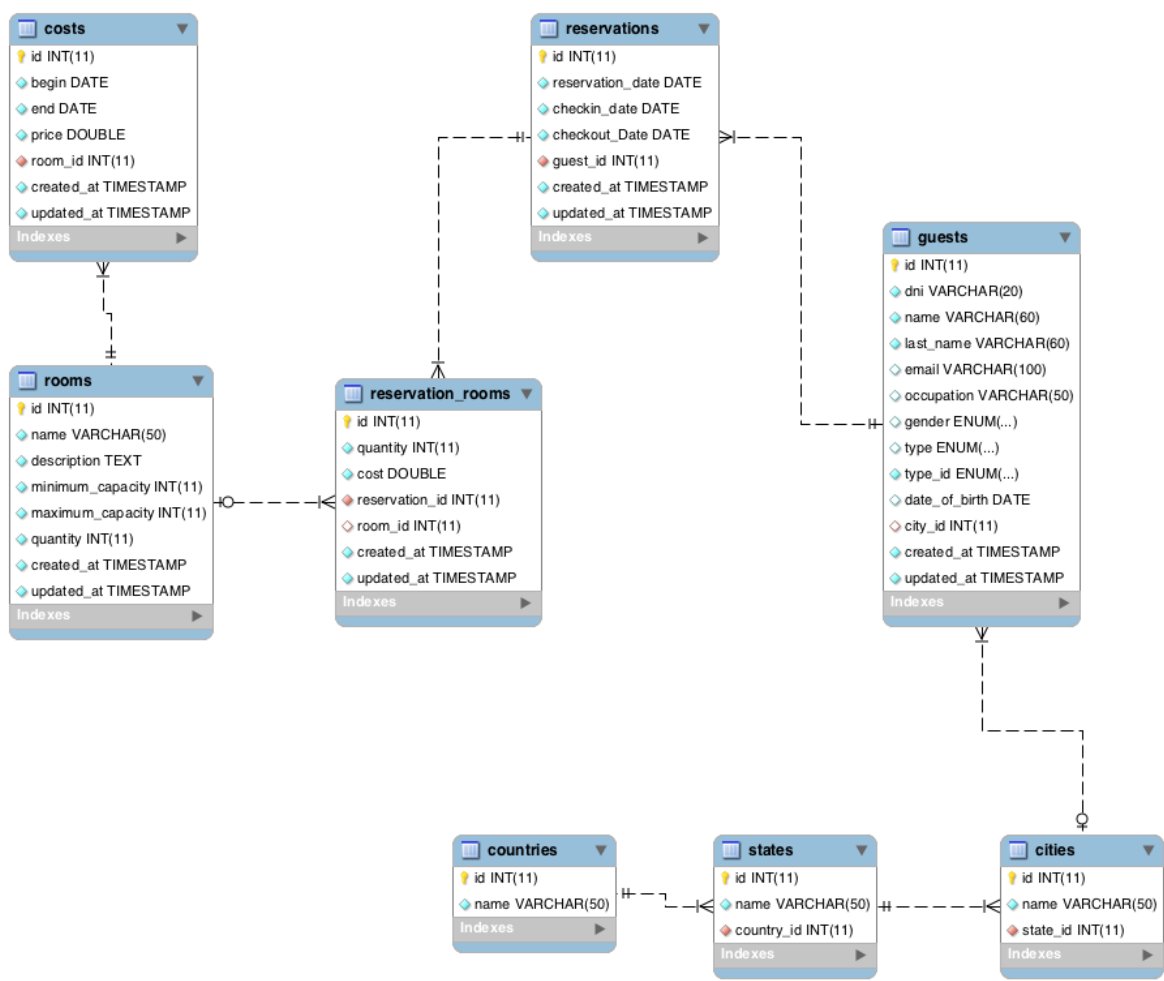
27/02/2017 - 28/02/2017

- Para entender mejor este apartado, daremos un ejemplo: El tipo de habitación suite junior tiene 3 costos. Uno para todo el mes de febrero, uno para el 14 de febrero celebrando el día del amor y amistad y otro para el 27 y 28 de febrero

celebrando en carnaval. Tenemos una reserva que va desde el 01 de febrero al 28 de febrero. Se deben mostrar todos los costos que tiene esa habitación en ese rango de fechas. Dado que hay varios costos para esa fecha, se deben mostrar segmentados. Del 01 al 13 solo existe el costo estándar. El 14 está disponible el costo estandar y el de amor, del 15 al 26 solo el costo estándar y el 27 y 28 el costo estandar y el costo carnavales. El usuario debe escoger el costo que requiere para cada fecha.

El botón "Reservar" almacena la reserva en la tabla reservation, si el huésped no está registrado, lo almacena en la tabla guests, lo asocia a la reserva, relaciona la reserva con los tipos de habitación reservados mediante la tabla reservation_rooms cuyo campo costo es el costo total del tipo de habitación (la sumatoria del costo durante todo el rango de la estadía) y muestra un mensaje "Reserva guardada con éxito".

Diagrama Entidad Relación



SQL para crear la base de datos (MySQL)

-- Table `rooms`

```
CREATE TABLE IF NOT EXISTS `rooms` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(50) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL ,  
  `description` TEXT CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL ,  
  `minimum_capacity` INT(11) NOT NULL ,  
  `maximum_capacity` INT(11) NOT NULL ,  
  `quantity` INT(11) NOT NULL ,  
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,  
  `updated_at` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00' ,  
  PRIMARY KEY (`id`) )  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

-- Table `countries`

```
CREATE TABLE IF NOT EXISTS `countries` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(50) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL ,  
  PRIMARY KEY (`id`) )  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

-- Table `states`

```
CREATE TABLE IF NOT EXISTS `states` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(50) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL ,  
  `country_id` INT(11) NOT NULL ,  
  PRIMARY KEY (`id`) ,  
  INDEX `fk_state_country` (`country_id` ASC) ,  
  CONSTRAINT `fk_state_country`  
    FOREIGN KEY (`country_id` )  
      REFERENCES `countries` (`id` )  
      ON DELETE CASCADE  
      ON UPDATE CASCADE)  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8  
COLLATE = utf8_unicode_ci;
```

-- Table `cities`

```
CREATE TABLE IF NOT EXISTS `cities` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `name` VARCHAR(50) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL ,  
  `state_id` INT(11) NOT NULL ,
```

```
PRIMARY KEY (`id`),
INDEX `fk_city_state` (`state_id` ASC),
CONSTRAINT `fk_city_state`
  FOREIGN KEY (`state_id`)
    REFERENCES `states` (`id`)
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;
```

```
-----
-- Table `guests`
-----
```

```
CREATE TABLE IF NOT EXISTS `guests` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `dni` VARCHAR(20) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL,
  `name` VARCHAR(60) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL,
  `last_name` VARCHAR(60) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT NULL,
  `email` VARCHAR(100) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NULL DEFAULT NULL,
  `occupation` VARCHAR(50) CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NULL DEFAULT NULL,
  `gender` ENUM('female','male') CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NULL DEFAULT NULL,
  `type` ENUM('n/a','miss','mr','mrs') CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NULL DEFAULT
NULL,
  `type_id` ENUM('passport','dni','dne','company') CHARACTER SET 'utf8' COLLATE 'utf8_unicode_ci' NOT
NULL,
  `date_of_birth` DATE NULL DEFAULT NULL,
  `city_id` INT(11) NULL DEFAULT NULL,
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updated_at` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`id`),
  UNIQUE INDEX `dni` (`dni` ASC),
  INDEX `fk_guest_city` (`city_id` ASC),
  CONSTRAINT `fk_guest_city`
    FOREIGN KEY (`city_id`)
      REFERENCES `cities` (`id`)
      ON DELETE CASCADE
      ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;
```

```
-----
-- Table `costs`
-----
```

```
CREATE TABLE IF NOT EXISTS `costs` (
  `id` INT(11) NOT NULL AUTO_INCREMENT,
  `begin` DATE NOT NULL,
  `end` DATE NOT NULL,
  `price` DOUBLE NOT NULL,
  `room_id` INT(11) NOT NULL,
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `updated_at` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`id`),
  INDEX `fk_cost_room` (`room_id` ASC),
  CONSTRAINT `fk_cost_room`
    FOREIGN KEY (`room_id`)
```

```
REFERENCES `rooms` (`id` )
ON DELETE CASCADE
ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;
```

```
-----
-- Table `reservations`
-----
```

```
CREATE TABLE IF NOT EXISTS `reservations` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `reservation_date` DATE NOT NULL ,
  `checkin_date` DATE NOT NULL ,
  `checkout_date` DATE NOT NULL ,
  `guest_id` INT(11) NOT NULL ,
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,
  `updated_at` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00' ,
  PRIMARY KEY (`id`) ,
  INDEX `fk_reservation_guest` (`guest_id` ASC) ,
  CONSTRAINT `fk_reservation_guest`
    FOREIGN KEY (`guest_id` )
      REFERENCES `guests` (`id` )
        ON DELETE CASCADE
        ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;
```

```
-----
-- Table `reservation_rooms`
-----
```

```
CREATE TABLE IF NOT EXISTS `reservation_rooms` (
  `id` INT(11) NOT NULL AUTO_INCREMENT ,
  `quantity` INT(11) NOT NULL ,
  `cost` DOUBLE NOT NULL ,
  `reservation_id` INT(11) NOT NULL ,
  `room_id` INT(11) NULL DEFAULT NULL ,
  `created_at` TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ,
  `updated_at` TIMESTAMP NOT NULL DEFAULT '0000-00-00 00:00:00' ,
  PRIMARY KEY (`id`) ,
  INDEX `fk_reservation_romm_reservation` (`reservation_id` ASC) ,
  INDEX `fk_reservation_room_room` (`room_id` ASC) ,
  CONSTRAINT `fk_reservation_room_reservation`
    FOREIGN KEY (`reservation_id` )
      REFERENCES `reservations` (`id` )
        ON DELETE CASCADE
        ON UPDATE CASCADE,
  CONSTRAINT `fk_reservation_room_room`
    FOREIGN KEY (`room_id` )
      REFERENCES `rooms` (`id` )
        ON DELETE CASCADE
        ON UPDATE CASCADE)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_unicode_ci;
```

SQL para introducir datos básicos (MySQL)

```
INSERT INTO `rooms` (name,description,minimum_capacity,maximum_capacity,quantity)
VALUES ('Suite Junior','Suite con muchos lujos',1,4,5), ('Habitación
Doble','Habitación muy cómoda para parejas',1,2,3);
```

```
INSERT INTO `countries` (name) VALUES ('Venezuela'),('Colombia');
```

```
INSERT INTO `states` (name,country_id) VALUES ('Bogotá D.C.',2) , ('Distrito
Capital',1);
```

```
INSERT INTO `cities` (name,state_id) VALUES ('Bogotá',1) , ('Caracas',2);
```

```
INSERT INTO `guests`
(dni,name,last_name,email,occupation,gender,type,type_id,date_of_birth) VALUES
('12345678','Pedro Andrés','Pérez
Pérez','pedro_perez@gmail.com','Maestro','male','mr','dni','1990-07-26');
```

```
INSERT INTO `guests` (dni,name,email,type_id) VALUES
('987654321','Máximoss INC','maximossinc@gmail.com','company');
```

```
INSERT INTO `costs` (begin,end,price,room_id) VALUES
(CURRENT_DATE,ADDDATE(CURRENT_DATE,30),150000,1),
(ADDDATE(CURRENT_DATE,10),ADDDATE(CURRENT_DATE,11),250000,1),
(ADDDATE(CURRENT_DATE,28),ADDDATE(CURRENT_DATE,30),170000,1),
(ADDDATE(CURRENT_DATE,10),ADDDATE(CURRENT_DATE,11),100000,1);
```

¿Como saber si una habitación está disponible en un rango de fecha?

La tabla 'rooms' posee un campo 'quantity'. Este campo indica la cantidad de habitaciones que hay (en físico en el hotel, es decir, número total de habitaciones disponibles de ese tipo). Ahora para saber cuántas de esas habitaciones están disponibles se deben verificar las reservas existentes. Si existe alguna reserva que tenga asociada en la tabla 'reservation_rooms' la habitación en el rango de fecha a buscar (quiere decir que la habitación está ocupada, ya alguien la reservó), se descuenta la cantidad reservada de esa habitación de las disponibles.

