



# PRUEBA TÉCNICA DE INGRESO

Ingeniero de Desarrollo

Juan Camilo Paternina Martinez  
Juank311a@gmail.com

## 1. ANALISIS Y DISEÑO

### a. Notas o aclaraciones:

- Aclarar si existe un lapso de no disponibilidad entre la salida del cliente reservado de la habitación y la nueva reservación, para que el equipo de servicios generales la prepare para una nueva reserva.

### b. Desglose de trabajo.

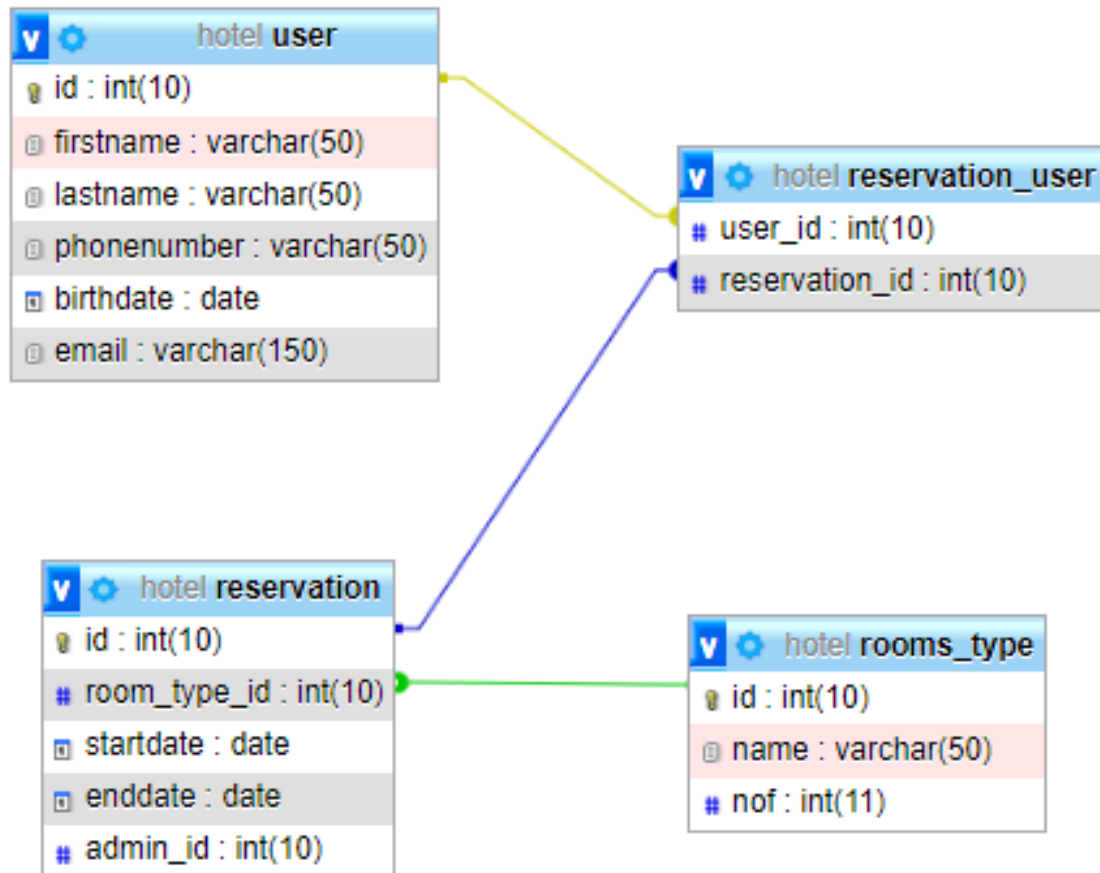
Estructura de Desglose de Trabajo EDG	
<b>Título:</b> Gestión de Reservaciones Hotel	<b>Responsable:</b> Juan Camilo Paternina Martínez
<b>Metodología:</b> SCRUM	
<b>Objetivo principal:</b> Gestionar las reservas del hotel por medio de un panel administrativo. <b>Objetivos específicos:</b> <ul style="list-style-type: none"><li>• Consultar las reservas existentes condicionadas por medio de datos de entrada.</li><li>• Permitir generar una nueva reservación teniendo en cuenta parámetros establecidos por el sistema.</li></ul>	
Nivel 1: Análisis e Identificación del Spring	
Items	Objetivo
Analizar los requerimientos.	Analizar los requerimientos de cliente de forma detallada, para optimizar las soluciones ofrecidas.
Establecer objetivos.	Establecer la trazabilidad de los objetivos con el fin de evitar cambios a último momento.
Cronograma de Actividad	Establecer un cronograma de las actividades a realizar y tiempos establecidos.
Nivel 2: Desarrollo del Spring	
1. Base de datos	Diseñar y crear una base de datos para almacenar información sobre las habitaciones, las reservaciones existentes y los clientes.
2. Diseño FROM	Crear una interfaz de usuario para que el administrador pueda consultar la disponibilidad de las habitaciones y generar reservaciones.
3. Diseño FROM	Crear una interfaz para que el administrador pueda ver las reservaciones existentes, editarlas o cancelarlas si es necesario.
4. Desarrollo de funcionalidad	Implementar un sistema para validar y gestionar la disponibilidad de las habitaciones antes de generar una reservación.
Nivel 3: Evaluaciones y Seguridad	
5.	Crear un sistema de seguridad para garantizar que solo el administrador tenga acceso al sistema de gestión de reservaciones.
6. Evaluación del software	Probar y depurar el sistema para asegurar su correcto funcionamiento.

c. Escriba un caso de prueba de la historia

Caso de prueba: Gestión de reservaciones Hotel	
<b>ID Caso de Prueba:</b>	<b>CP-0001</b>
<b>Descripción de la prueba:</b>	Verificar que el sistema de gestión de las reservaciones del Hotel, permita al administrador del mismo verificar de manera adecuada la disponibilidad de habitaciones en tiempo real, dependiendo de los datos de fecha y hora ingresados, en caso de que existan reservaciones debe mostrar cuantas ocupadas hay con detalles de cantidad, hora, fecha y tipo; a su vez mostrar cuantas habitaciones están sin reserva si es el caso, si es así, debe permitir poder realizar la reservaciones a las habitaciones con disponibilidad.
<b>Restricciones:</b>	El sistema de gestión de reservas del hotel no debe permitir crear una reserva sobre habitaciones que no existe disponibilidad.
<b>Condiciones previas:</b>	<ul style="list-style-type: none"> <li>• El sistema debe contar con un acceso de tipo administrador.</li> <li>• El sistema debe contar con datos en la DB para poder contar con los diferentes eventos a los cuales se le realizarán las pruebas.</li> </ul>
<b>Datos de prueba:</b>	User_admin: admin Pass_admin: admin Start_date: ##### End_date ##### Roms_type: ##
<b>Pasos a ejecutar:</b>	<ol style="list-style-type: none"> <li>1. Logueo en la aplicación como Roll:Administrador.(externo)</li> <li>2. Ingresar a la sección de habitaciones. (externo)</li> <li>3. Presionar el botón: Disponibilidad. (externo)</li> <li>4. Ingresar datos de entrada: fecha y hora. (externo)</li> <li>5. Realizar la consulta en base de datos. (interno)</li> <li>6. Mostrar disponibilidad de habitaciones para los datos ingresados, clasificar por tipo de habitación. (interno)</li> <li>7. Presionar el botón “reservar” de una habitación disponible. (externo)</li> <li>8. Redireccionar al módulo de reservacion_usuario. (Interno)</li> <li>9. Diligenciar los datos de nuevo usuario, (externo)</li> <li>10. Presionar el botón: Reservar (Exterior)</li> </ol>
<b>Resultado esperado:</b>	Realizar exitosamente la reservación de una o varias habitaciones. <ol style="list-style-type: none"> <li>1. El sistema permite consultar las reservaciones existentes a partir de los criterios dados.</li> <li>2. El sistema permite generar una reservación con la información requerida teniendo en cuenta la disponibilidad.</li> </ol>

## 2. BASES DE DATOS.

### Modelo Relacional



### Consulta 1:

```
SELECT rt.id as 'room_type_id', startdate, enddate, rt.nof as  
'num_of_rooms',  
  
COUNT(r.id) as 'num_of_reservations', rt.nof -  
COUNT(r.id) as 'available_rooms'  
  
FROM rooms_type rt LEFT JOIN Reservation r ON rt.id = r.room_type_id  
WHERE r.startdate <= '2017-04-29' AND r.enddate >= '2017-05-01'  
  
GROUP BY rt.id, r.startdate, r.enddate;
```

### Explicación:

La consulta realiza una relación entre las dos tablas mediante un JOIN LEFT entre las tablas "rooms\_type" y "reservation" en el campo "id" de "room\_type" y "room\_type\_id" de "reservation". Además, se utiliza el COUNT para contar el numero de filas de la tabla "reservation" por lo tanto cuenta el numero de reservas que cumplen la cláusula WHERE filtradas entre las fechas "startdate" y "enddate" especificadas. La consulta también calcula el número de habitaciones disponibles utilizando una expresión matemática, "rt.nof - COUNT(r.id)" con el alias "available\_rooms", que es el resultado de restar el número de habitaciones totales (nof) a las habitaciones reservadas (COUNT(r.id))

La consulta agrupa los resultados por "room\_type\_id", "startdate" y "enddate".

### Resultados gráficos de la consulta:

Mostrando filas 0 - 2 (total de 3, La consulta tardó 0,0097 segundos.)

```
SELECT rt.id as 'room_type_id', startdate, enddate, rt.nof as 'num_of_rooms', COUNT(r.id) as 'num_of_reservations', rt.nof - COUNT(r.id) as 'available_rooms' FROM  
rooms_type rt LEFT JOIN Reservation r ON rt.id = r.room_type_id WHERE r.startdate <= '2017-04-29' AND r.enddate >= '2017-05-01' GROUP BY rt.id, r.startdate, r.enddate;
```

☐ Perfilando [\[ Editar en línea \]](#) [\[ Editar \]](#) [\[ Explicar SQL \]](#) [\[ Crear código PHP \]](#) [\[ Actualizar \]](#)

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

room_type_id	startdate	enddate	num_of_rooms	num_of_reservations	available_rooms
1	2017-04-29	2017-05-01	2	1	1
2	2017-04-29	2017-05-01	1	1	0
3	2017-04-29	2017-05-01	2	1	1

## Consulta 2:

```
SELECT r.admin_id as 'admin_id', COUNT(r.id) as 'num_of_reservations', u.  
firstname, u.lastname  
FROM Reservation r  
LEFT JOIN Reservation_user ru  
ON r.id = ru.reservation_id  
LEFT JOIN User u  
ON ru.user_id = u.id  
GROUP BY r.admin_id, u.id ORDER BY r.admin_id, u.firstname ASC;
```

## Explicación:

La consulta selecciona información de tres tablas, "Reservation", "Reservation\_user" y "User", uniéndolas mediante dos JOIN LEFT y utiliza la función COUNT() para contar el número de reservaciones realizadas por administradores. También selecciona información del nombre y apellido de los usuarios que realizaron las reservas, y utiliza la cláusula GROUP BY para agrupar los resultados por "admin\_id" y "u.id" y la cláusula ORDER BY para ordenar los resultados por "admin\_id" y "u.firstname" en orden ascendente.

Con esta consulta, se pudo obtener una información detallada sobre las reservas realizadas por administradores, incluyendo el número de reservas realizadas, el nombre y apellido de los usuarios que realizaron las reservas.

## Resultados gráficos de la consulta:

✓ Mostrando filas 0 - 5 (total de 6, La consulta tardó 0,0141 segundos.) [admin\_id: 1... - 3...] [firstname: JANE... - JOHANA...]

SELECT r.admin\_id as 'admin\_id', COUNT(r.id) as 'num\_of\_reservations', u.firstname, u.lastname FROM Reservation r LEFT JOIN Reservation\_user ru ON r.id = ru.reservation\_id LEFT JOIN User u ON ru.user\_id = u.id GROUP BY r.admin\_id, u.id ORDER BY r.admin\_id, u.firstname ASC;

☐ Perfilando [ Editar en línea ] [ Editar ] [ Explicar SQL ] [ Crear código PHP ] [ Actualizar ]

☐ Mostrar todo | Número de filas: 25 | Filtrar filas:

+ Opciones

admin_id	num_of_reservations	firstname	lastname
1	1	Jane	Jackson
1	2	Jhon	Doe
2	2	Alex	Smith
2	1	Jane	Jackson
2	1	Johana	Roll
3	1	Johana	Roll