

# Documentación de proyecto de SW

En este documento se explicará el desarrollo de la aplicación para la tercera entrega y su funcionamiento de manera básica.

## Desarrollo:

Para empezar, hay que entender que previamente fue necesario crear un modelo UML, el cual fue guía para la construcción de las colecciones y sus validaciones (ver docs/modelos y docs/archivos-json/validaciones). Ya con esto, se inició el desarrollo en 4 pasos principales:

1. Modelos
2. Repositorios
3. Controladores
4. Front

### 1. Modelos

En este paso se modelan, mediante clases, las mismas colecciones que aparecen en el diagrama de entidades, con sus atributos y documentos embebidos. Para definir las colecciones, se usó la anotación `@Document`, y para definir los documentos embebidos se crearon clases aparte que fueron llamadas como `List<Clase>` si son un arreglo o simplemente `Clase` si son un objeto. Algo importante para notar es que, para aquellas colecciones que cuentan con referencias a otras, simplemente se referenció su id (que es de tipo entero). Por último, en cada clase se definieron los constructores, getters y setters.

### 2. Repositorios

Una vez los modelos fueron creados, se definieron los repositorios que contienen las sentencias de Mongo para acceder a/manipular la base de datos. Para ello, por cada colección se definió un repositorio y, dentro de él, sentencias básicas del CRUD (insert, select, update, delete) que permiten el correcto funcionamiento de la mayoría de RFs. Aquí también se definieron consultas las consultas de los RFCs.

Es importante mencionar que los repositorios fueron creados por colección y no por clase, es decir, si un RF o RFC se debe hacer sobre un documento embebido, sus sentencias fueron definidas dentro del repositorio de la colección de la cual son parte.

Por último, para aquellas consultas que devolvían listas u objetos que no estaban modelados como clase (en las proyecciones, por ejemplo), fue necesario crear clases aparte en la carpeta `src/.../grupos`.

### 3. Controladores

Ya con los modelos y los repositorios terminados, se definen los controladores para cada colección, documento embebido y RFC. Aquí se definen las rutas y se hace un llamado

a los repositorios para ejecutar las sentencias de Mongo que se verán reflejadas en el front dependiendo de la acción seleccionada, como ver, editar, borrar o agregar (CRUD) o consultar.

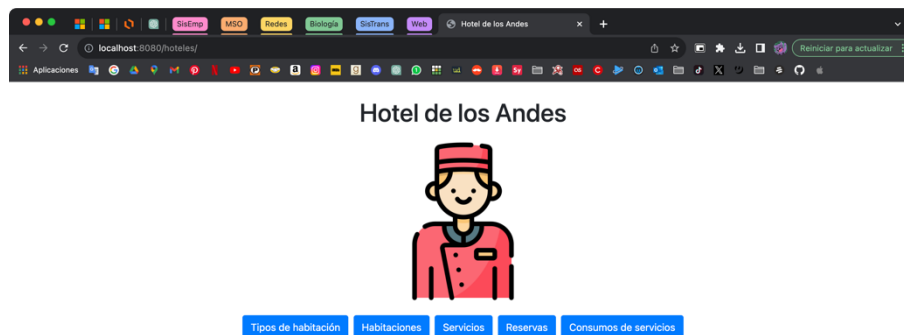
#### 4. Front

Con todo lo anterior creado, se definen los archivos HTML que harán uso de las rutas definidas en los controladores y mostrará la información almacenada en la base de datos. Para esto, por cada colección se crean 4 archivos: el que muestra la tabla con todos los datos, el formulario para editar, el formulario para crear, y el formulario base para los dos formularios anteriores. Además, también se crean los botones que permiten la navegación. Es importante mencionar que en las tablas muchas veces no se muestran los datos literalmente. Ejemplo: para la tabla de habitaciones, aunque una habitación tenga un tipo que hace referencia al id de ese tipo, se muestra el nombre del tipo y no el id. Esto se hizo con la intención de hacer la información de las tablas más amigable.

Por otro lado, para cada RFC se crea un archivo: el que muestra la tabla con todos los datos de la consulta y, en algunos casos, contiene el formulario para realizar búsquedas avanzadas.

#### Funcionamiento:

Por el lado del funcionamiento, para correr la aplicación se debe acceder al archivo principal HotelesApplication.java, y ejecutarlo. Cuando esté corriendo se debe ingresar a la dirección <http://localhost:8080/hoteles>, que muestra lo siguiente:



*Fig. 1: Página principal*

Por parte de los RFs, estos pueden ser encontrados dentro de las vistas de cada colección o documento embebido, dando click en los botones de la página principal.

El RF1 indica el CRUD de los tipos de habitación, por lo que al dar click en el botón correspondiente se observa lo siguiente:

**Tipos de habitación**

ID	Nombre	Dotación incluida	Capacidad	Acciones
1	Familiar	<a href="#">Ver</a>	8	<a href="#">Editar Borrar</a>
2	Suite	<a href="#">Ver</a>	6	<a href="#">Editar Borrar</a>
3	Doble	<a href="#">Ver</a>	2	<a href="#">Editar Borrar</a>

[Volver atrás](#)
[Añadir tipo de habitación](#)

*Fig. 2: Lista de tipos de habitación*

Entonces, aquí se cumple R. Para C, se debe dar click a *Añadir tipo de habitación*, lo que mostrará el siguiente formulario vacío:

**Crear tipo de habitación**

Nombre

Capacidad

[Cancelar](#)
[Guardar](#)

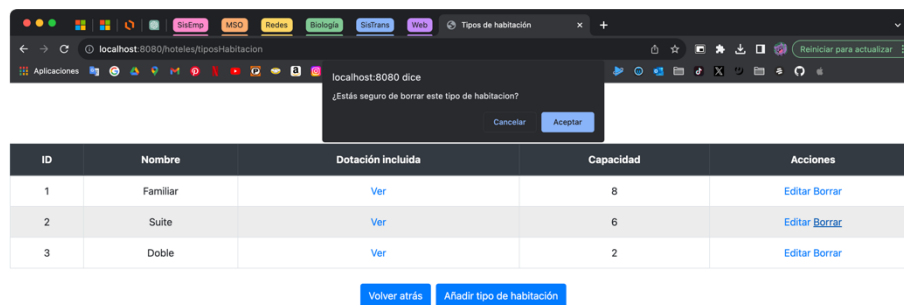
*Fig. 3: Crear tipo de habitación*

Una vez se llenen los datos, se creará un nuevo tipo de habitación con la información proporcionada. Para U, se da click en *Editar*, lo que muestra el mismo formulario pero con los datos de la fila seleccionada, que, una vez se editan, también cambiarán su valor en la BD:



Fig. 4: Editar tipo de habitación

Por último, para D, se puede dar click en *Borrar*, lo que lanzará un mensaje de confirmación y, una vez se acepte, eliminará el tipo de la base de datos:



localhost:8080/hoteles/tiposhabitacion/2/delete

Fig. 5: Borrar tipo de habitación

El RF2 es lo mismo que el RF1, pero con habitaciones, por lo que solo se mostrará la tabla principal (igual el resto del CRUD funciona):

**Habitaciones**

[Dinero recolectado](#)
[Índice de ocupación](#)

Número de habitación	Tipo	Costo noche	Acciones
101	Familiar	\$100000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
102	Doble	\$150000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
103	Suite	\$200000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
201	Familiar	\$200000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
202	Doble	\$250000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
203	Suite	\$300000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
301	Familiar	\$300000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
302	Doble	\$350000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
303	Suite	\$400000.0	<a href="#">Editar</a> <a href="#">Borrar</a>
401	Familiar	\$10.0	<a href="#">Editar</a> <a href="#">Borrar</a>

Fig. 6: Lista de habitaciones

El RF3 es lo mismo que el RF1, pero con servicios, por lo que solo se mostrará la tabla principal (igual el resto del CRUD funciona):

**Servicios**

[Reiniciar para actualizar](#)

ID	Nombre	Tipo cobro	Precio	Capacidad	Estilo	Productos	Costo adicional	Acciones
1	Gimnasio	Alojamiento	\$50000.0	100		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
2	Piscina	Gratuito	\$0.0	150		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
3	Internet	Habitacion	\$10000.0	50		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
4	Bar	Habitacion	\$1000.0	50	Inglés	<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
5	Restaurante	Alojamiento	\$5000.0	40	Gourmet	<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
6	Supermercado	Gratuito	\$0.0	200		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
7	Tienda joyería	Gratuito	\$0.0	20		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
8	Tienda ropa	Gratuito	\$0.0	30		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
9	Spa	Alojamiento	\$150000.0	15		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
10	Lavado	Habitacion	\$20000.0	10		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>
11	Planchado	Habitacion	\$15000.0	10		<a href="#">Ver</a>	\$0.0	<a href="#">Editar</a> <a href="#">Borrar</a>

Fig. 7: Lista de servicios

El RF4 es lo mismo que el RF1, pero con reservas. Sin embargo, acá está la adición de que solo se puede crear una reserva si la habitación no está ocupada en las fechas dadas. Para ello, en el repositorio correspondiente se definió una sentencia de Mongo avanzada que verificara esto. Si la reserva se puede hacer, se va a ver agregada a la tabla, si no, no. El caso exitoso devuelve a la lista de reservas:



Código	Fecha entrada	Fecha salida	Cliente	Habitación	Número de huéspedes	Estado	Acciones
1	2023-11-19	2023-12-02	Ver	101	1	Entro	Editar Borrar
2	2022-11-19	2022-12-02	Ver	201	1	Salio sin deuda	Editar Borrar
3	2024-11-19	2024-12-02	Ver	301	1	Por entrar	Editar Borrar
4	2023-01-01	2023-12-01	Ver	102	2	Entro	Editar Borrar
5	2023-01-01	2023-01-08	Ver	303	1	Entro	Editar Borrar
6	2022-11-19	2022-12-02	Ver	303	1	Salio sin deuda	Editar Borrar
7	2023-11-19	2023-12-02	Ver	303	1	Por entrar	Editar Borrar
8	2023-11-19	2023-12-02	Ver	202	1	Entro	Editar Borrar

Volver atrás Añadir reserva

Fig. 8: Lista de reservas

El caso fallido, sin embargo, muestra la siguiente página:

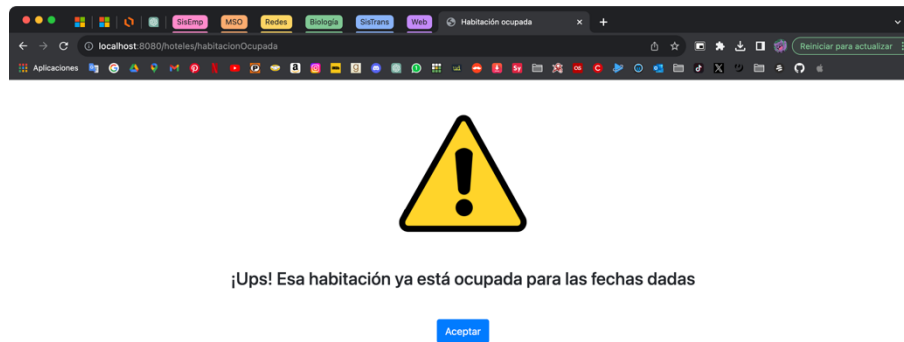


Fig. 9: Creación de reserva fallida

El RF5 registra la entrada de un cliente. Para esto, simplemente se puede editar una reserva y cambiar el valor de estado a “Entró”:

**Editar reserva**

Fecha entrada  
19/11/2024

Fecha salida  
02/12/2024

Habitación  
301

Número de huéspedes  
1

Estado  
Entró

► Cliente:

[Cancelar](#) [Guardar](#)

*Fig. 10: Registro de llegada de cliente*

El RF6 es lo mismo que el RF1, pero con consumos, por lo que solo se mostrará la tabla principal (igual el resto del CRUD funciona):

**Consumos de servicios**

[Consumos cliente](#) [Clientes consumo](#)

ID	Servicio	Cliente	Habitación	Costo	Fecha	Hora	Acciones
1	Bar	1006	101	\$1000.0	2023-11-26	14:00	<a href="#">Editar</a> <a href="#">Borrar</a>
2	Gimnasio	1007	201	\$50000.0	2022-11-20	15:00	<a href="#">Editar</a> <a href="#">Borrar</a>
3	Piscina	1008	301	\$0.0	2023-11-21	16:00	<a href="#">Editar</a> <a href="#">Borrar</a>
4	Gimnasio	1007	202	\$50000.0	2023-11-20	15:00	<a href="#">Editar</a> <a href="#">Borrar</a>
5	Gimnasio	1007	202	\$50000.0	2023-11-20	16:00	<a href="#">Editar</a> <a href="#">Borrar</a>

[Volver atrás](#) [Añadir consumo](#)

*Fig. 11: Lista de consumos de servicios*

El RF7 registra la salida de un cliente. Para esto, simplemente se puede editar una reserva y cambiar el valor de estado a “Salió con deuda” o “Salió sin deuda”:

**Editar reserva**

Fecha entrada  
19/11/2023

Fecha salida  
02/12/2023

Habitación  
101

Número de huéspedes  
1

Estado  
Salido con deuda

► Cliente:

Cancelar Guardar

Fig. 12: Registro de salida de cliente

Ahora, por parte de los RFCs, estos se encuentran dentro de las vistas de habitaciones y consumos, y pueden ser accedidos dando click en los botones de color **azul verdoso**.

El RFC1 muestra el dinero recolectado por cada habitación en el último año corrido. Para acceder a esta información, hay que dirigirse a la vista de habitaciones, se debe dar click al botón *Dinero recolectado*, lo que mostrará lo siguiente:

**Dinero recolectado por servicios**

Número de habitación	Dinero
101	\$1000.0
102	\$0.0
103	\$0.0
201	\$0.0
202	\$100000.0
203	\$0.0
301	\$0.0
302	\$0.0
303	\$0.0

Volver atrás

Fig. 13: Dinero recolectado

Aquí se observa el dinero recolectado por cada habitación que existe, incluso cuando no ha tenido consumos.

El RFC2 muestra el índice de ocupación de cada habitación en el último año corrido. Para acceder a esta información, hay que dirigirse a la vista de habitaciones y dar click en el botón *Índice de ocupación*, lo que mostrará lo siguiente:





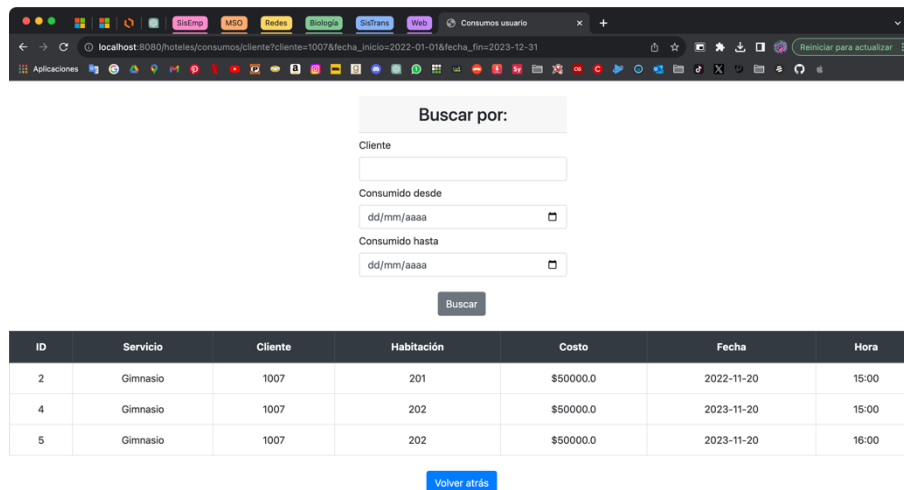
Número de habitación	Porcentaje de ocupación
101	3.88%
102	99.7%
103	0.0%
201	0.0%
202	3.88%
203	0.0%
301	0.0%
302	0.0%
303	5.97%

Volver atrás

Fig. 14: Índice de ocupación

Aquí se observa, en porcentaje, el índice de ocupación para cada habitación, incluso si nunca fue ocupada.

El RFC3, muestra los consumos realizados por un cliente. Para acceder a esta información, hay que dirigirse a la vista de consumos de servicios y dar click en el botón *Consumos cliente*, lo que mostrará un formulario colapsado para escoger un cliente por su número de documento, y un rango de fechas. Una vez se haya llenado, se muestra lo siguiente:



Buscar por:

Cliente

Consumido desde

Consumido hasta

Buscar

ID	Servicio	Cliente	Habitación	Costo	Fecha	Hora
2	Gimnasio	1007	201	\$50000.0	2022-11-20	15:00
4	Gimnasio	1007	202	\$50000.0	2023-11-20	15:00
5	Gimnasio	1007	202	\$50000.0	2023-11-20	16:00

Volver atrás

Fig. 15: Consumos cliente

En este ejemplo se buscaron los consumos del cliente con número de documento 1007 entre el 1ro de enero de 2022 y el 31 de diciembre de 2023.

El RFC4, muestra los datos de los clientes que consumieron un servicio al menos una vez. Para acceder a esta información, hay que dirigirse a la vista de consumos de servicios y dar click en el botón *Clientes consumo*, lo que mostrará un formulario colapsado para escoger entre un servicio, fecha de consumo, un mínimo de veces consumido, el número de

documento del cliente, su tipo de documento, su nombre y su correo. Estos filtros pueden aplicarse todos al tiempo o individualmente. Una vez se haya llenado, se muestra lo siguiente:

The screenshot shows a web browser window with the address bar displaying a URL with query parameters. Below the browser window is a search form with the following fields:

- dd/mm/aaaa (date input)
- Veces consumido (text input)
- Número de documento (text input)
- Tipo de documento (dropdown menu)
- Nombre cliente (text input)
- Correo (text input)

Below the form is a "Buscar" button. Below the button is a table with the following data:

Número documento	Tipo de documento	Nombre	Correo	Fecha	Veces consumido
1006	CC	Laura	Lrestrepop@uniandes.edu.co	2023-11-26	1

Below the table is a "Volver atrás" button.

Fig. 16: Clientes consumo

En este ejemplo se realizó un filtro solamente por el servicio, habiendo seleccionado el de bar.