

Documentación de proyecto de SW

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

Desarrollo:

Para empezar, hay que entender que previamente fue necesario crear un modelo relacional, el cual fue guía para la construcción de las clases (ver informe o carpeta modelos/modelo-relacional). Ya con esto, se inició el desarrollo en 4 pasos principales:

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

1. Modelos

En este paso, en la carpeta *modelo*, se, valga la redundancia, modelan las mismas tablas que aparecen en el modelo relacional, con sus atributos y relaciones. Algo importante para notar es que, para aquellas tablas que cuentan con una PK que contiene más de un atributo o que cuenta con FKs como parte de ella, fue necesario hacer también una clase para esta PK (ejemplo, la PK de Acompañantes tiene un número de documento y una reserva, por lo que se crea una clase AcompañantePK). Por otro lado, para aquellas tablas que tienen FKs que no son parte de la PK, simplemente se hizo una relación mediante las anotaciones `@ManyToOne` y `@OneToOne`. También 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 SQL para acceder a/manipular la base de datos. Para ello, por cada tabla en el modelo relacional se definió un repositorio y, dentro de él, sentencias básicas del CRUD (insert, select, update, delete). Aquí también se definieron consultas más avanzadas para aquellos requerimientos que lo necesitaran. Estas sentencias CRUD son las que permiten el correcto funcionamiento de la mayoría de RFs.

3. Controladores

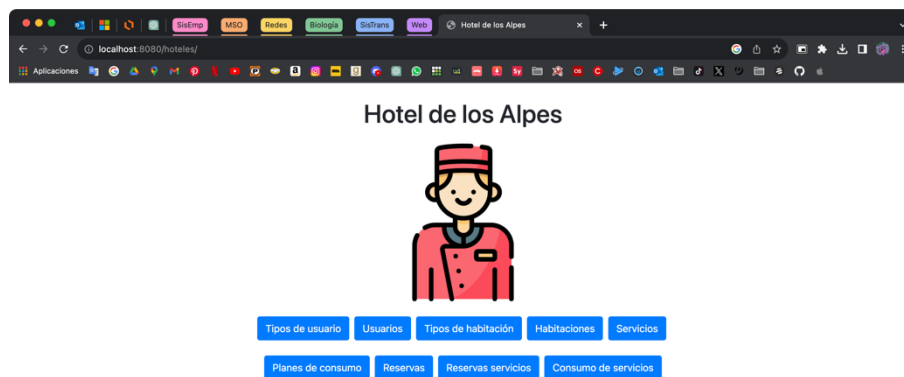
Ya con el back terminado (modelos y repositorios), se definen los controladores para cada clase que es parte de un RF. Aquí se definen las rutas y se hace un llamado a los repositorios para ejecutar las sentencias SQL que se verán reflejadas en el front dependiendo de la acción seleccionada, como ver, editar, borrar o agregar (CRUD).

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 controlador 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 usuarios, aunque un usuario 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.

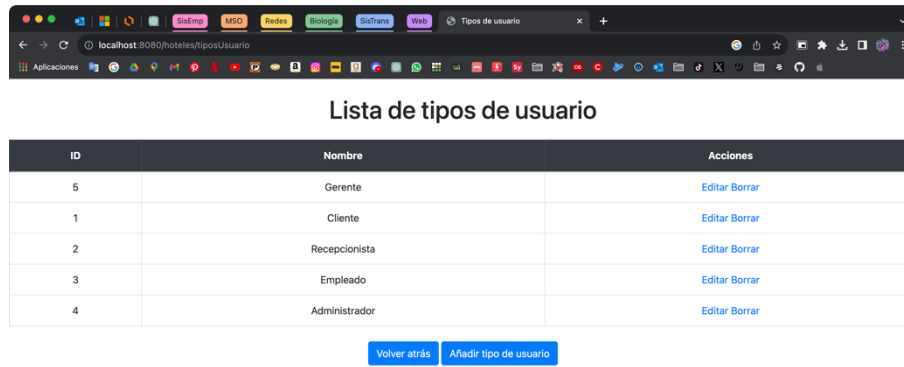
Funcionamiento:

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



Página principal

Como se puede ver, cada botón representa, para casi todos los casos, un RF. El RF1 indica el CRUD de los tipos de usuario, por lo que al dar click en el botón correspondiente se observa lo siguiente:



Lista de tipos de usuario

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

Crear tipo de usuario

Nombre

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

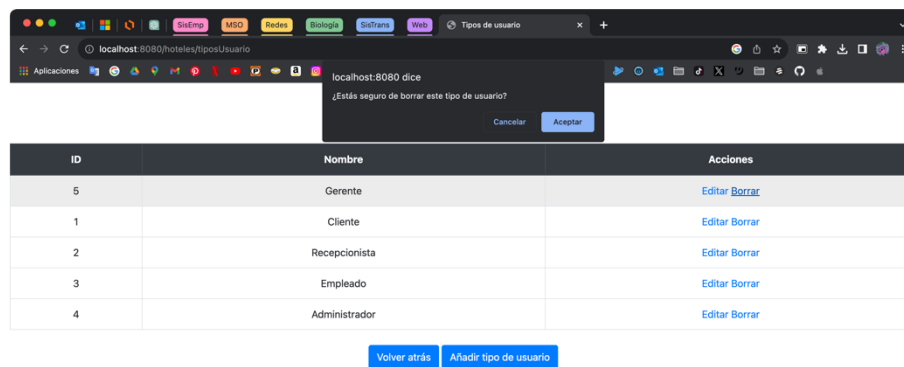
Crear tipo de usuario

Una vez se llenen los datos, se creará un nuevo tipo de usuario 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:



Editar tipo de usuario

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/tiposUsuario/5/delete

Eliminar tipo de usuario

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

Número de documento	Tipo de documento	Nombre	Correo	Tipo de usuario	Hotel	Acciones
1009	CC	Valentina	v.arenas@uniandes.edu.co	Cliente	732757320-8	Editar Borrar
1006	CC	Laura	l.restrepop@uniandes.edu.co	Cliente	732757320-8	Editar Borrar
1007	CC	Catalina	c.espita@uniandes.edu.co	Cliente	732757320-8	Editar Borrar
1008	CC	Sergio	s.cifuentes@uniandes.edu.co	Cliente	732757320-8	Editar Borrar
1010	CC	Juan	j.yepes@uniandes.edu.co	Gerente	732757320-8	Editar Borrar
1011	CC	Lucía	l.posada@uniandes.edu.co	Administrador	732757320-8	Editar Borrar
1012	CC	Lindsay	l.pinto@uniandes.edu.co	Recepcionista	732757320-8	Editar Borrar
1013	CC	Yei	y.zhang@uniandes.edu.co	Recepcionista	732757320-8	Editar Borrar
1016	CC	Juan	j.fajardo@uniandes.edu.co	Empleado	732757320-8	Editar Borrar
1014	CC	Leidy	l.lozano@uniandes.edu.co	Empleado	732757320-8	Editar Borrar
1015	CC	Tatiana	t.vera@uniandes.edu.co	Empleado	732757320-8	Editar Borrar

Lista de usuarios

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

ID	Nombre	Dotación	Capacidad	Acciones
3	Familiar	Comedor y cocina	6	Editar Borrar
4	Doble jacuzzi	Jacuzzi	3	Editar Borrar
5	Doble	Sin jacuzzi	3	Editar Borrar
2	Suite	Comedor y cocina	7	Editar Borrar
1	Suite presidencial	Comedor y cocina	8	Editar Borrar

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

Lista de tipos de habitación

El RF4 es lo mismo que el RF1, pero con habitaciones, por lo que solo se mostrará la tabla principal (igual el resto del CRUD funciona):

Lista de habitaciones

Número de habitación	Costo noche	Tipo	Hotel	Acciones
103	300.0	Familiar	732757320-8	Editar Borrar
101	500.0	Suite presidencial	732757320-8	Editar Borrar
102	250.0	Doble	732757320-8	Editar Borrar

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

Lista de habitaciones

El RF5 es lo mismo que el RF1, pero con servicios, por lo que solo se mostrará la tabla principal (igual el resto del CRUD funciona):

Lista de servicios

Tipo de servicio	Capacidad	Tipo cobro	Acciones
Piscina	10	Alojamiento	Editar Borrar
Gimnasio	9	Alojamiento	Editar Borrar
Bar	7	Habitacion	Editar Borrar
Internet	7	Alojamiento	Editar Borrar
Restaurante	18	Habitacion	Editar Borrar
Supermercado	9	Habitacion	Editar Borrar
Tienda joyeria	21	Habitacion	Editar Borrar
Tienda ropa	25	Habitacion	Editar Borrar
Spa	23	Habitacion	Editar Borrar
Lavado	7	Alojamiento	Editar Borrar
Planchado	7	Alojamiento	Editar Borrar

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

Planes de consumo

localhost:8080/hoteles/planesConsumo

Listado de planes de consumo

Tipo	¿Génera descuento?	Tiempo de estadía	¿Aplica descuentos a servicios comerciales?	Valor de descuento servicio comercial	Acciones
Plan secundario	N	1	Y	0.2	Editar Borrar
Plan familiar	Y	3	N	0.0	Editar Borrar
Plan todo incluido	N	5	N	0.0	Editar Borrar
Plan larga estadua	Y	7	Y	0.15	Editar Borrar
Plan principal	Y	3	Y	0.4	Editar Borrar

[Volver atrás](#) [Añadir plan de consumo](#)

Lista de planes de consumo

El RF7 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 indicadas. Para ello, en el repositorio correspondiente se definió una sentencia SQL avanzada que verificara esto. Si la reserva se puede hacer, se va a ver agregada a la tabla, si no, no. Ambos resultados devuelven a la lista de reservas:

Reservas

localhost:8080/hoteles/reservas

Listado de reservas

Código	Fecha entrada	Fecha salida	Número de huéspedes	¿Entró?	¿Salió?	Paz y salvo	Cliente	Hotel	Habitacion	Plan consumo	Acciones
1	2001-01-01	2001-02-01	10	Y	N	N	1006	732757320-8	101	Plan principal	Editar Borrar

[Volver atrás](#) [Añadir reserva](#)

Lista de reservas antes de crear

Crear reserva

Fecha entrada
25/09/2023

Fecha salida
01/10/2023

Número de huéspedes
2

¿Entró?
No

¿Salió?
No

Paz y salvo
No

Cliente
1007

Hotel
732757320-8

Crear reserva

Lista de reservas

Código	Fecha entrada	Fecha salida	Número de huéspedes	¿Entró?	¿Salió?	Paz y salvo	Cliente	Hotel	Habitacion	Plan consumo	Acciones
1	2001-01-01	2001-02-01	10	Y	N	N	1006	732757320-8	101	Plan principal	Editar Borrar
23	2023-09-25	2023-10-01	2	N	N	N	1007	732757320-8	101	Plan secundario	Editar Borrar

[Volver atrás](#) [Añadir reserva](#)

Lista de reservas después de crear

El RF8 es lo mismo que el RF1, pero con reservas de un servicio (de renta o comercial), por lo que solo se mostrará la tabla principal:

Lista de reservas de servicios

ID	Fecha	Hora inicio	Hora fin	Servicio comercial	Servicio renta	Cliente	Acciones
6	2023-10-27	14:30:00	15:00:00		Salon de reunion	1006	Editar Borrar
3	2001-01-02	15:00:00	16:00:00		Salon de conferencia	1006	Editar Borrar

[Volver atrás](#) [Añadir reserva de servicio](#)

Lista de reservas de servicios

El RF9 registra la entrada de un cliente. Para esto, simplemente se puede editar una reserva y cambiar el valor de “Entro” a verdadero, poniendo “Sí” en el campo “¿Entró?”.

El RF10 es lo mismo que el RF1, pero con consumos de servicios, por lo que solo se mostrará la tabla principal (igual el resto del CRUD funciona):

Lista de consumo de servicios

Cliente	Servicio	Fecha	Hora	¿Cuenta pagada?	Acciones
1006	Piscina	2001-01-25	16:00:00	N	Editar Borrar

[Volver atrás](#) [Añadir consumo](#)

Lista de consumo de servicios

El RF11 registra la salida de un cliente. Para esto, simplemente se puede editar una reserva y cambiar el valor de “Salio” y “Paz y salvo” a verdadero, poniendo “Sí” en los campos “¿Salió?” y “Paz y salvo”.