

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA: INGENIERÍA EN COMPUTACIÓN



Proyecto Final

Bases de Datos

Gloria Etelbina Chavez Valenzuela

Medina López Moisés Alejandro

01155492

Salas Atondo Giovanni Alberto

01158360

Mexicali, Baja California a 4 de Diciembre del 2020

Descripción de la base de datos

Se solicita crear una base de datos para el procesado y almacenado de reservaciones de un restaurante.

El restaurante posee mesas, de las cuales se desea saber el código de la mesa y su localización. Se tienen diferentes comidas, se debe saber el código de comida, el nombre, el precio, el tipo de comida (entrada, plato fuerte, bebida o postre) y una pequeña descripción.

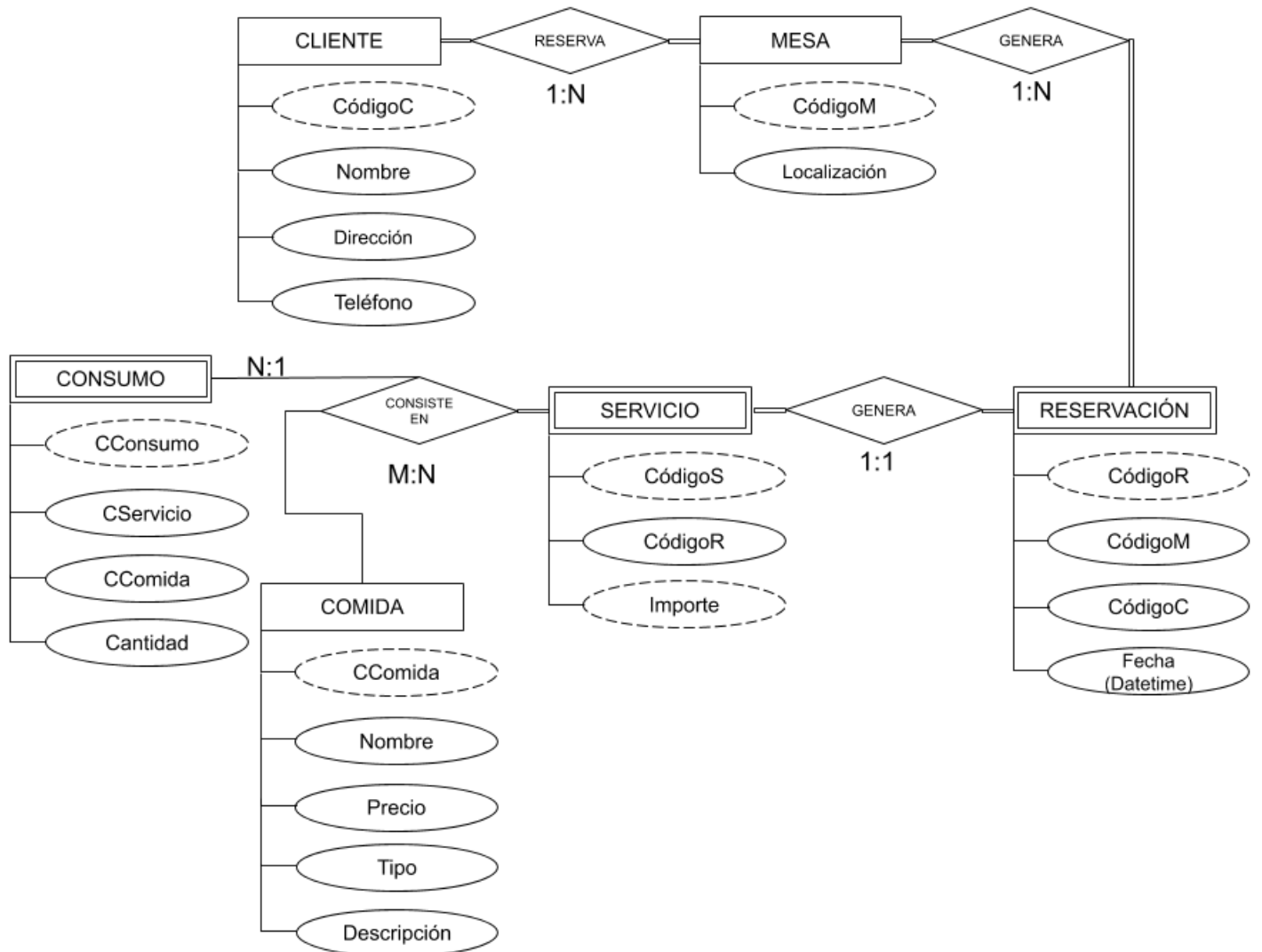
De los clientes se almacenará el código de cliente, su nombre, dirección y teléfono. Cuando un cliente reserva una mesa se crea una reservación; de las reservaciones se desea guardar el código de reservación, la fecha y hora de la reservación y qué mesa se asignó a qué cliente.

Las reservaciones generan servicios, que consisten en un código de servicio, la reservación a la que se asignó y el importe (dinero generado por ese servicio).

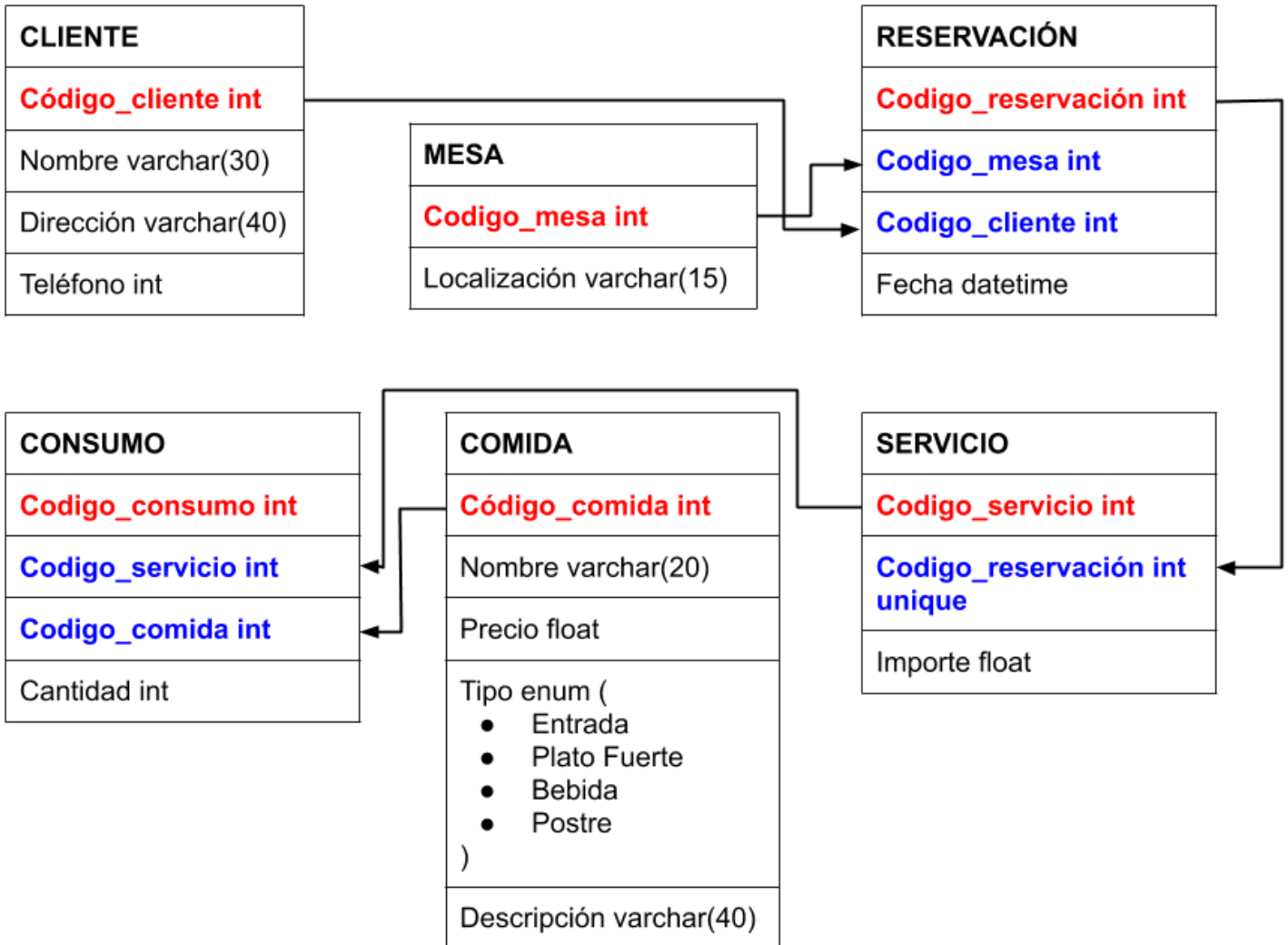
Para calcular el importe del servicio es necesario saber qué se consumió, se guardarán los datos de cada solicitud del cliente durante el servicio; dicha solicitud consistirá en el código del consumo, el servicio al que está relacionado, la comida solicitada y la cantidad de esa misma comida en caso de que el cliente solicite más de un plato o bebida; es posible solicitar la misma comida en muchos consumos diferentes aún en el mismo servicio, estos se guardarán con un código de consumo diferente, no afecta al cálculo del importe.

Considerar que el restaurante abre a las 12 del mediodía y cierra a las 9 de la noche, cada reservación dura tres horas. Se debe prevenir que no se puede reservar una mesa si su reservación actual aún no ha cumplido sus tres horas y que no se pueden hacer reservaciones que terminen después de que el restaurante cierre.

Modelo Entidad-Relación



Diseño de la base de datos



Tablas mySQL

CLIENTE

```
mysql> describe cliente;
```

Field	Type	Null	Key	Default	Extra
Codigo_cliente	int	NO	PRI	NULL	auto_increment
Nombre	varchar(30)	YES		NULL	
Direccion	varchar(40)	YES		NULL	
Telefono	int	YES		NULL	

4 rows in set (0.00 sec)

MESA

```
mysql> describe mesa;
```

Field	Type	Null	Key	Default	Extra
Codigo_mesa	int	NO	PRI	NULL	auto_increment
Localizacion	varchar(15)	YES		NULL	

RESERVACIÓN

```
mysql> describe reservation;
```

Field	Type	Null	Key	Default	Extra
Codigo_reservacion	int	NO	PRI	NULL	auto_increment
Codigo_mesa	int	YES	MUL	NULL	
Codigo_cliente	int	YES	MUL	NULL	
Fecha	datetime	YES		NULL	

SERVICIO

```
mysql> describe servicio;
```

Field	Type	Null	Key	Default	Extra
Codigo_servicio	int	NO	PRI	NULL	auto_increment
Codigo_reservacion	int	YES	UNI	NULL	
Importe	float	NO		0	

COMIDA

```
mysql> describe comida;
```

Field	Type	Null	Key	Default	Extra
Codigo_comida	int	NO	PRI	NULL	auto_increment
Nombre	varchar(30)	YES		NULL	
Precio	float	YES		NULL	
Tipo	enum('ENTRADA','PLATO FUERTE','BEBIDA','POSTRE')	YES		NULL	
Descripcion	varchar(40)	YES		NULL	

```
5 rows in set (0.00 sec)
```

CONSUMO

```
mysql> describe consumo;
```

Field	Type	Null	Key	Default	Extra
Codigo_consumo	int	NO	PRI	NULL	auto_increment
Codigo_servicio	int	YES	MUL	NULL	
Codigo_comida	int	YES	MUL	NULL	
Cantidad	int	YES		NULL	

```
4 rows in set (0.00 sec)
```

Functions y Triggers

Esta base de datos es bastante sencilla, las validaciones son tan fáciles que no necesitan función.

Sólo se necesita un Trigger para aumentar automáticamente el importe de los servicios al insertar consumos relacionados con ellos.

Dicho Trigger se coloca a continuación:

```
mysql> delimiter //
```

```
mysql> create trigger sumarImporte
```

```
-> after insert on CONSUMO
```

```
-> for each row
```

```
-> begin
```

```
-> update SERVICIO set Importe = Importe +
```

```
-> (select distinct Precio from COMIDA inner join CONSUMO
```

```
-> on new.Codigo_comida = COMIDA.Codigo_comida) *
```

```
-> new.Cantidad where Codigo_servicio = new.Codigo_servicio;
```

```
-> end //
```

```
mysql> delimiter ;
```

Descripción del programa en Java

Versión de JAVA y del Connector mySQL



DTO(Data Transfer Object)

Es un objeto común que tiene como atributos los datos del modelo, con sus correspondientes getters y setters. Son utilizados por DAO para transportar los datos desde la base de datos.

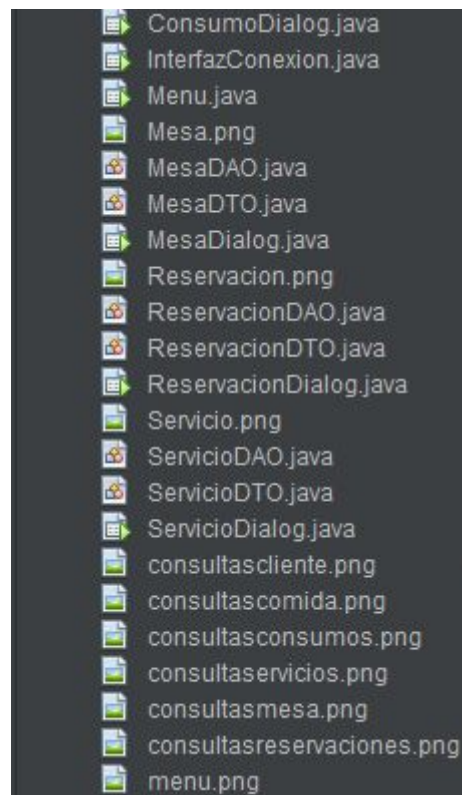
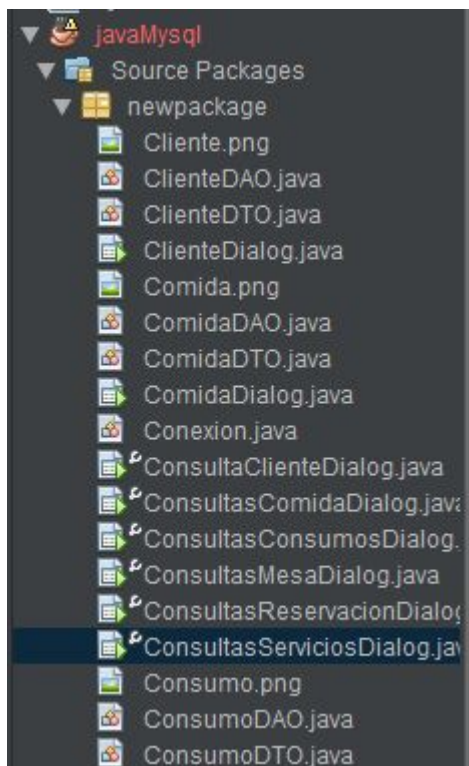
DAO(Data Access Object)

Consiste en una clase que interactúa con la base de datos. Los métodos de esta clase dependen de la aplicación y de lo que se quiere hacer. Por lo normal se implementan los método CRUD para realizar las operaciones básicas de una base de datos.

Interfaz

El diseño de cada JFrame o Jdialog fue hecho desde un programa llamado GIMP, por lo cual se tomaron medidas de cada JFrame o Jdialog y luego se pasaron al programa GIMP y posteriormente se fueron diseñando con color y títulos, cada diseño se hizo con creatividad propia de nosotros.

Clases del programa



VISTA DEL MENÚ PRINCIPAL



Altas

Para insertar un dato en una tabla, se accede desde los Registros



Se abrirá el menú de inserción

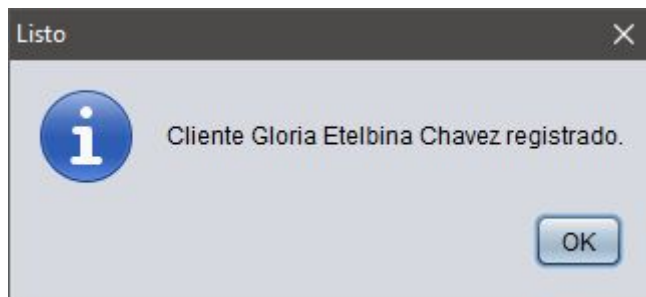
The screenshot shows the 'Cliente' form, which has a green header with the title 'Cliente'. The form contains three input fields: 'Nombre' with the value 'Gloria Etelbina Chavez', 'Direccion' with the value 'Mexicali??', and 'Telefono' with the value '1100110'. A red rectangle highlights these three input fields. At the bottom right of the form, there is a button labeled 'Agregar', also highlighted with a red rectangle.

Nombre: Gloria Etelbina Chavez

Direccion: Mexicali??

Telefono: 1100110

Agregar



Al llenar los datos y dar *Agregar* habremos insertado un dato a nuestra tabla.

NOTAS

- Hay datos que en lugar de cuadros de texto son *JComboBox*. Esto con la intención de generar una comprensión más clara de lo que hay que insertar.
- Los códigos o ID de las tablas son auto incrementables, por lo que en ningún momento se colocan a mano en el programa de Java.

CASOS DONDE NO SE PUEDE INSERTAR

- No se puede insertar una Reservación si no hay antes al menos un cliente y una mesa registrados; los *JComboBox* de selección de mesa y cliente estarán vacíos y esto no permitirá insertar reservaciones.
- No se puede registrar una reservación en una mesa si dicha mesa está reservada y la diferencia de la fecha de los registros es de menos de tres horas.
- No se pueden registrar servicios si no hay reservaciones registradas.
- No se pueden registrar consumos si no hay servicios registrados.
- No se pueden registrar Consumos si no hay comidas registradas.

Consultas

Para consultar una tabla, se accede desde las consultas



Al presionar el botón *Buscar* se mostrarán todos los datos de la tabla según el modo seleccionado en el comboBox



Aún cuando se seleccione un modo de consulta, el ID de la Tabla siempre será visible en la primera columna.

NOTA

- No todas las consultas tienen opción de búsqueda por tipo; en estos casos no habrá un botón *Buscar* y los datos se cargarán automáticamente.

Modificaciones

Para modificar un campo, se accede desde las consultas



Después hay que tocar el dato que necesitamos modificar.

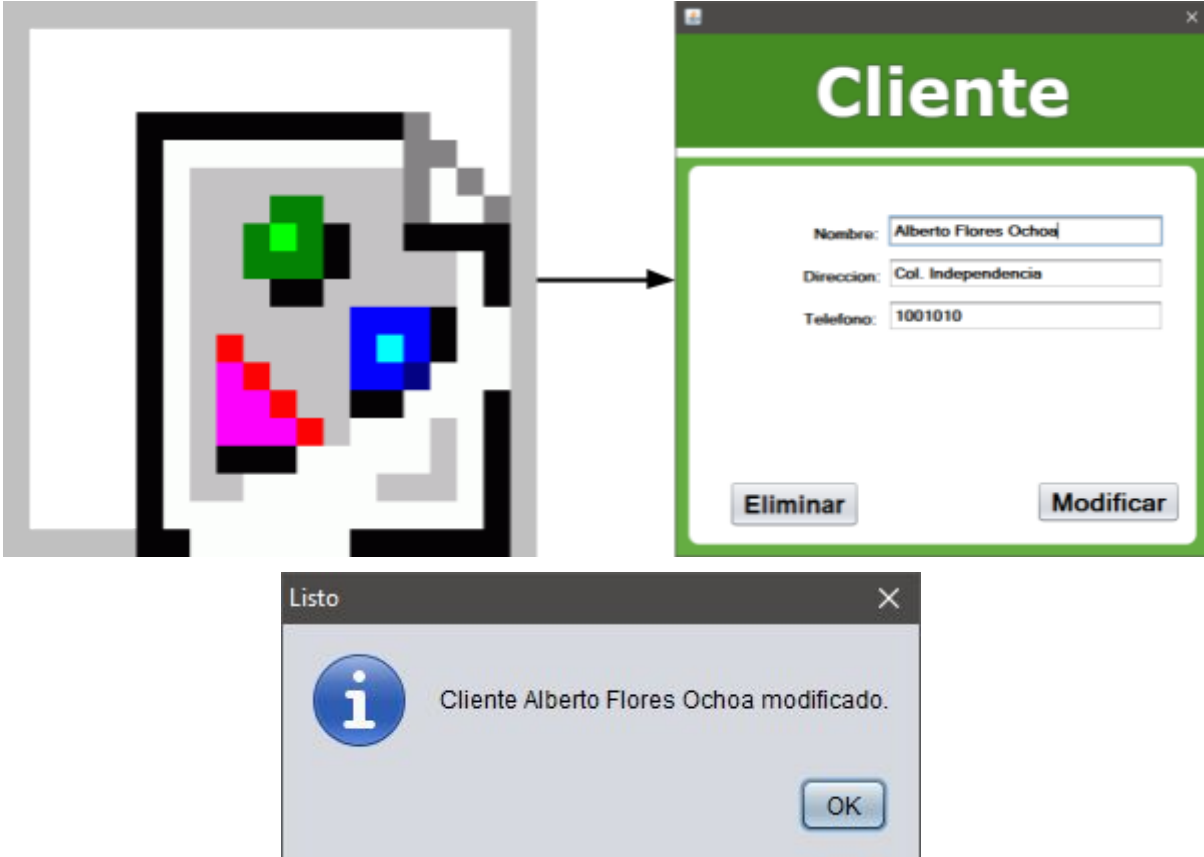


Consulta Clientes

Todo Buscar Cerrar

ID Cliente	Nombre	Dirección	Teléfono
1	Moisés Alejandro Medina López	Villas del Colorado	3236753
2	Rocío Paola Payan López	Some Place in SLRC	1234567
3	Giovanni Alberto Salas Alondo	Who knows?	7654321
4	Alberto	Col. Independencia	1001010
6	Sergio De Jesús Medina Lopez	Villas del Colorado	1010101

Se abrirá la ventana de modificaciones, modificamos los campos y damos **Modificar** y listo.




The diagram illustrates the process of modifying a client record. It starts with a pixelated icon representing a database or list, which points to a 'Cliente' form. The form contains fields for 'Nombre', 'Direccion', and 'Telefono', with the values 'Alberto Flores Ochoa', 'Col. Independencia', and '1001010' respectively. Below the form are 'Eliminar' and 'Modificar' buttons. An arrow points from the 'Modificar' button to a 'Listo' (Ready) dialog box. The dialog box features an information icon and the message 'Cliente Alberto Flores Ochoa modificado.' with an 'OK' button.

Cliente

Nombre: Alberto Flores Ochoa
Direccion: Col. Independencia
Telefono: 1001010

Eliminar Modificar

Listo

 Cliente Alberto Flores Ochoa modificado.

OK

Se mostrará el JOptionPane exitoso si lo hicimos bien.

Consulta Clientes			
<input type="button" value="Todo"/> <input type="button" value="Buscar"/>		<input type="button" value="Cerrar"/>	
ID Cliente	Nombre	Dirección	Teléfono
1	Moisés Alejandro Medina López	Villas del Colorado	3236753
2	Rocío Paola Payan López	Some Place in SLRC	1234567
3	Giovanni Alberto Salas Alonzo	Who knows?	7654321
4	Alberto Flores Ochoa	Col. Independencia	1001010
5	Sergio De Jesús Medina Lopez	Villas del Colorado	1010101

Datos actualizados.

```
mysql> select * from CLIENTE where nombre like 'Alberto%';
+-----+-----+-----+-----+
| Codigo_cliente | Nombre           | Direccion         | Telefono |
+-----+-----+-----+-----+
| 4              | Alberto Flores Ochoa | Col. Independencia | 1001010 |
+-----+-----+-----+-----+
```

DATOS NO MODIFICABLES

- No es posible modificar la tabla CONSUMO; los platillos ya fueron cargados al servicio y por lo tanto al importe, no hay devoluciones.
- No es posible modificar los servicios porque ya están ligados a una Reservación y el importe no es editable, sino calculable.
- No es posible modificar reservación de una mesa si la nueva reservación tiene una diferencia de menos de tres horas con otra reservación en la misma mesa.

ADVERTENCIAS

- Al modificar el precio de una comida **NO** se actualizan los importes de los servicios. Sí se guardarán los nuevos consumos con precios actualizados.
- Los JComboBox no guardan la información del dato seleccionado, estarán en su valor por defecto al abrir la ventana de modificaciones.

Bajas

Para eliminar un dato, se accede desde las consultas



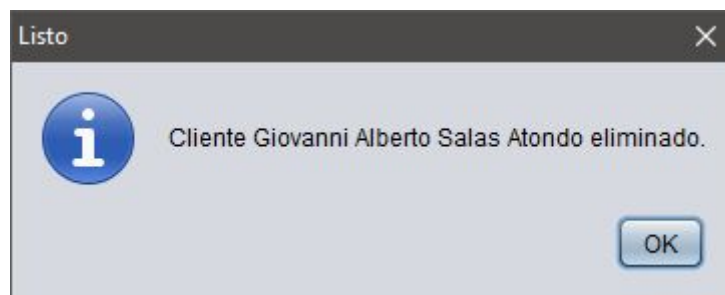
Después hay que tocar el dato que necesitamos eliminar.



Se abrirá la ventana de modificaciones, seleccionamos eliminar y listo.



The screenshot shows a window titled "Cliente" with a green header. Below the header is a white form area with three input fields: "Nombre:" containing "Giovanni Alberto Salas Atondo", "Direccion:" containing "Who knows?", and "Telefono:" containing "7654321". At the bottom of the form are two buttons: "Eliminar" (highlighted with a red rectangle) and "Modificar".



DATOS NO ELIMINABLES

- No se pueden eliminar clientes que tengan una reservación, sin embargo, si la reservación aún no tiene un servicio asignado, se puede eliminar la reservación y luego el cliente.
- No se pueden eliminar mesas que tengan una reservación, sin embargo, si la reservación aún no tiene un servicio asignado, se puede eliminar la reservación y luego la mesa.
- No se pueden eliminar reservaciones que ya tengan un servicio asignado.
- No se puede eliminar ningún servicio.
- No se puede eliminar la comida asignada a un consumo.
- No se puede eliminar ningún consumo.

NOTA

- Al eliminar un dato, no se reasignan los ID, significa que el *auto_increment* de la tabla no se reducirá. El funcionamiento del programa y la base de datos no se ven afectados por esto, se contempló y se codificó la captura de datos de manera que no afecten los huecos entre los ID.

Conclusiones

Medina López Moisés Alejandro

- Ha sido uno de los proyectos finales más divertidos que he tenido en toda la carrera.
- Fue bastante desafiante, hacer un programa en Java con clases que no conozco para nada, aunque eran pocas, así que aprendí a manejarlas rápido.
- Creo que hemos creado un proyecto bastante bueno, estoy muy satisfecho con nuestro trabajo.
- Si bien las funciones en MySQL son muy complicadas para mí aún, tener la oportunidad de liberar algo de la carga usando Java hizo que no me fuera imposible trabajar este proyecto.
- Aprendí bastante con este proyecto, y en todo el semestre, además, es la primera vez que trabajo en un programa en java donde no tengo que rellenar todos los datos desde el principio en cada ejecución, es un gran alivio mental y agiliza muchísimo la codificación del proyecto.

Salas Atondo Giovanni Alberto

- Se aprendió mucho acerca de cómo interactuar con una base de datos con mysql dentro de una interfaz en java.
- Al principio fue algo complicado pero al investigar y con ayuda de mis compañeros pude lograr implementar el código de cada identidad de nuestra base de datos.
- Al tener todo en una interfaz gráfica se mira más cómodo para el usuario que lo está utilizando, no tiene que recordar comandos ni nada por el estilo es solo presionar los botones y dejar todo el código que se haga por sí mismo.
- Fue algo interesante y a la vez divertido tener que diseñar todo en otro programa para luego importarlo y mirar que todo el diseño de la interfaz se mira bien, incluso es algo que al cliente le gusta mucho visualizar.