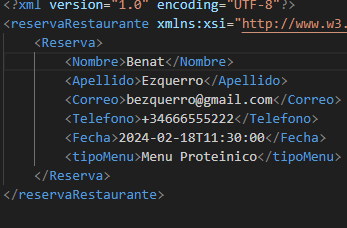
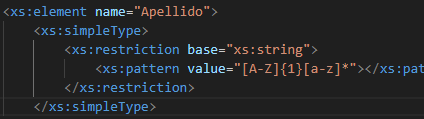
DOCUMENTACIÓN

# **XML,XSD: JERARQUIZACION DE GUARDADO DE DATOS**

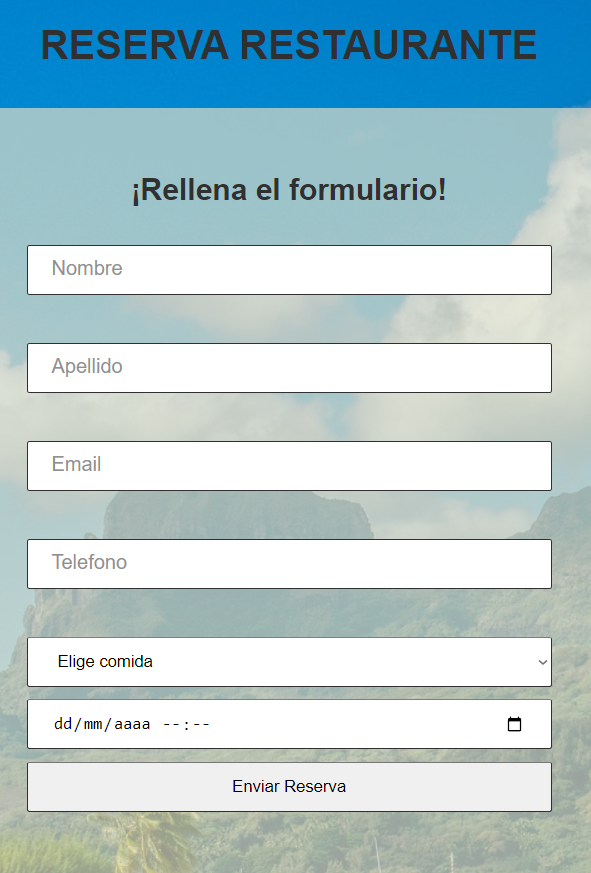


Hemos hecho un esquema de guardado de datos con respecto a las tablas "Deportistas" y "Comensales" en las cuales hemos puesto patrones y restricciones para que el guardado de datos sea correcto y fiel a el orden y contenido de las tablas

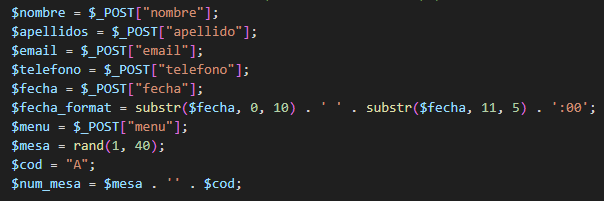


# FORMULARIOS DE RESERVA A GIMNASIO Y RESTAURANTE Y PHP, INSERCIONES A LA BASE DE DATOS

Cuando comenzamos a crear los formularios, íbamos a crear un solo formulario con un desplegable y según lo que se elija en el desplegable, se Reserva el Restaurante o el Gimnasio. Al final se optó por crear un formulario individual para cada tipo de reserva

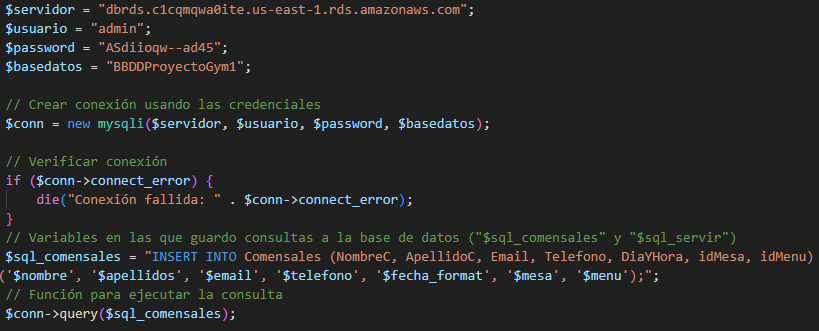
Nos quedamos con tres formularios los cuales son el formulario de Reservas a Restaurante, el de Suscripciones al Gimnasio y uno para Aplicar a un trabajo en el restaurante o el gimnasio.

Estos formularios están montados en archivos de extensión PHP, pero inicialmente iba a haber un archivo PHP para cada formulario, ya que era más sencillo de implementar. Finalmente se decidió integrar el PHP en los distintos formularios porque cuando estaban en archivos separados, al enviar el formulario, salía una ventana del PHP. Al meter el HTML del formulario en el archivo PHP, el código de PHP se ejecutaba dentro del formulario, así logrando evitar redirecciones.

Dentro del código de PHP se empieza extrayendo los valores de los campos del formulario, dependiendo del método usado en el formulario(POST y GET).

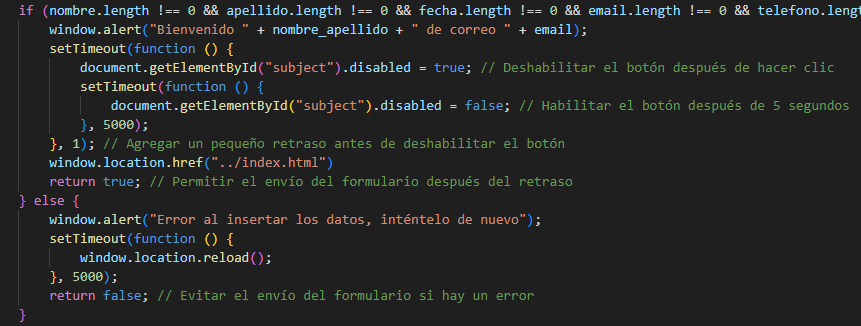
Luego se crea la conexión a la base de datos, creando variables que guardan los parámetros para la conexión y usándolos en otra variable para la conexión:

**$conn = new mysqli($servidor, $usuario, $password, $basedatos);** y luego se verifica la conexión por si hay errores al conectarse. Tras verificar la conexión, se crean variables con la query a ejecutar en la base de datos y esas variable se ejecutan llamando a la variable de la conexión: **$conn->query($consulta).** Tras haber hecho todas las query necesarias, se cierra la conexión: **$conn->close();**



Tras eso se hizo un código JavaScript para que al enviar el formulario, saliera un Pop-Up en la parte superior de la pantalla indicando si se ha hecho la reserva correctamente o no. Si la reserva se guarda sin problemas en la base de datos, en el Pop-Up saldrá el Nombre y apellido del/de la solicitante además de su correo electrónico y tras 5 segundos, será redirigido/a a la página principal. En caso de haber cualquier error en la solicitud por campos dejados en blanco, se notificará mediante un Pop-Up con un mensaje de error que dice "Todos los campos deben de ser rellenados, inténtelo de nuevo" y la página se recargará para que el formulario se vuelva a rellenar

Eso se hizo con una función de JavaScript llamada "Mensaje()" que se ejecuta tras enviar el formulario. Esta función extrae los datos del formulario y los guarda en variables, para luego usarlas en la condición. Si la condición se cumple, se crea un window.alert() dentro de un timeOut para que haya 5 segundos de espera tras el Pop-Up y se redirija a la página principal



# JAVA, CONSULTAS A LA BASE DE DATOS(SWING)

Al empezar, pensábamos hacer todas las peticiones(las de ambos sectores) en una misma ventana y solo usar una ventana pero al final pensamos que sería mejor dividirlo en sectores para optimizar el código y facilitar su lectura al estar dividido.

Programa Construido en Java SWING, este programa está hecho para que los empleados de MIIB Restauración & Deporte puedan consultar registros de la base de datos sobre el Restaurante y el Gimnasio. Cuenta con tres ventanas, una en la que eliges el sector de el cual quieres extraer datos, usando un JComboBox para elegir el sector y un JButton para confirmar la elección. Además cuenta con un fondo azul y el logo que representa nuestra marca. Al confirmar la elección de sector, se abrirá la respectiva ventana por encima.

Por otro lado tenemos las ventanas de los sectores, están conformadas de un JComboBox que te da cuatro opciones. En el caso del restaurante son: Mostrar todos los comensales,  Filtrar por nombre, Mostrar todas las mesas, Mostrar todos los menús. En el gimnasio son: Mostrar todos los Deportistas, Filtrar por nombre, Mostrar todas las Suscripciones, Mostrar todos las Zonas. Al elegir la opción que comparten ambos ("Filtrar por nombre"), se mostrará un cuadro de texto para filtrar los resultados por el nombre escrito y un botón para confirmar esos cambios. Al elegir cualquiera de las opciones y confirmar la elección, el programa te mostrará una tabla (hecha con JTable) con la información pedida. Estas ventanas tienen un fondo azul y botones de color turquesa oscuro, ademas de el logo de MIIB como icono de aplicacion

# BASE DE DATOS

Inicialmente iba a haber un hotel, con lo cual iba a haber tres tablas más (ZonaDescanso, Suites, Descansa[Tabla N:M] y Huéspedes) pero tras haber hecho la base de datos completa, se decidió que sería muy complicado hacerlo asi, con lo que descartamos la idea del hotel y seguimos con el resto de tablas.

Tras rehacer toda la base de datos y ponerse a hacer los formularios, nos dimos cuenta de que la base de datos no estaba bien diseñada así que la volvimos a rehacer y al final nos quedamos con ese diseño.

Nuestra base de datos está principalmente compuesta por dos tablas centrales, ZonaDeporte y ZonaRestauracion. Ambas tablas están conectadas a la tabla de empleados con relación 1:N[Trabajan] (de Zona a Empleado) en ambos casos, pues los empleados trabajan en ambas zonas. La tabla "ZonaRestauracion(idMesa, MesaNum)" tiene conectada la tabla "Menu(idMenu, Precio, ObjetoMenu)" mediante una relación N:M[SeSirve] y tiene conectada la tabla "Comensales(idComensal, NombreC, ApellidoC, Email, Telefono, DiaYHora)" mediante una relación N:1 (Hacia ZonaRestauracion). A su vez, la tabla Menú está conectada a la tabla Comensales con una relación de 1:N[EsComido] (de menú a Comensal).

La tabla "ZonaDeporte(idZona, NumTarjeta, Zona)" está conectada a la tabla "Deportistas(idDeportista, NombreD, ApellidoD, Email, Telefono, idZona)" con una relación N:M[Entrenan] y la tabla Deportistas está conectada a la tabla

"Suscripcion(idSubs, TipoSubs, PrecioSubs)" usando una relación de 1:N[Tienen] (de Suscripción a Deportistas).

# SERVIDOR AWS(BASE DE DATOS) y SERVIDOR DE NAZARET(SERVIDOR WEB)

Al inicio pensábamos alojar tanto la base de datos como la página web dentro del servidor EC2 de AWS Academy, pero tuvimos problemas con Java así que terminamos creando un servidor RDS para la base de datos y usando el servidor de Nazaret como servidor web.

El servidor RDS de Amazon Web Services lo usamos para alojar la base de datos. Es un tipo de instancia específica para montar bases de datos y en el que se pueden crear varias bases de datos. Por otro lado está el servidor de Nazaret, el

cual lo usamos para alojar la página web y que pueda ser visible tanto desde la misma red como desde una red externa. Se puede acceder desde internet poniendo la ip del router al que está conectado el servidor (85.50.79.98) y el puerto ssh junto con el puerto HTTP (23 y 80).

La URL sería la siguiente: http://       85.50.79.98 :2380

        (protocolo http) (ip del router)          (puertos ssh y http)