# RETO2



# INTRODUCCIÓN

Para el reto de la segunda evaluación hemos decidido hacer nuestro proyecto sobre un restaurante cuyo nombre es “El Rincón de Oliva”. En este documento hablaremos sobre los pasos que hemos realizado para llevar a cabo el proyecto mencionando las partes que corresponden a cada asignatura en base a lo que hemos hecho cada día.

El servicio de la página web consiste en una gestión de reservas de un restaurante en forma de formulario que se tendrá que rellenar en dicha web y dicha información se guardará en un servidor de AWS que estará conectada a la base de datos.

Los procesos que hemos llevado en este trabajo los mostraremos en el mismo orden en el que los hemos ido realizando y, a su vez, iremos explicando los apartados al mismo tiempo que hemos ido realizándolo en estas tres semanas de trabajo.

Pero antes de empezar, la página web contiene un total de 6 páginas las cuales están divididas de la siguiente manera:1 Portada (index.html), 1 formulario (reservas) y cuatro páginas más que su objetivo es dar información del restaurante.

El contenido de la portada de la página está constituido en un Titulo en el que contiene el nombre del restaurante (Rincón de Oliva), seguido por un menú que te permite navegar por las diferentes páginas de la web. Además, contiene una pequeña descripción del restaurante aportándonos información útil del restaurante. Para finalizar, la parte inferior de la página se puede observar el logo del restaurante y unas pequeñas fotos de su establecimiento. (Las páginas sobre la política de privacidad y de los avisos legales se sitúan a pie de página, aunque su funcionalidad es igual a la del menú.)

El formulario está compuesto por 7 input donde tenemos 3 inputs de tipo Varchar, 2 de tipo Integer, 1 de tipo Time y 1 de tipo Date. Donde el cliente deberá insertar los datos necesarios para realizar la reserva en el restaurante. A su vez, tenemos que añadir el botón de enviar (EVIDENTEMENTE)

Índice:

[RETO2 1](#_Toc158366324)

[INTRODUCCIÓN 2](#_Toc158366325)

[FOL Formación y Orientación Laboral 5](#_Toc158366326)

[ENGLISH 6](#_Toc158366327)

[Lenguaje de Marcas 7](#_Toc158366328)

[HTML: 7](#_Toc158366329)

[XSD: 8](#_Toc158366330)

[XML: 8](#_Toc158366331)

[Bases de Datos 10](#_Toc158366332)

[Introducción: 10](#_Toc158366333)

[Proceso llevado a cabo: 10](#_Toc158366334)

[Imágenes: 11](#_Toc158366335)

[Sistemas de Informáticos 13](#_Toc158366336)

[Instalación del servidor: 13](#_Toc158366337)

[Instalación del PHP: 14](#_Toc158366338)

[Programación 16](#_Toc158366339)

[Introducción: 16](#_Toc158366340)

[Proceso llevado a cabo:(Página 1) 16](#_Toc158366341)

[Página 2: 17](#_Toc158366342)

[Página 2 19](#_Toc158366343)

[Página 3 20](#_Toc158366344)

[Entornos de desarrollo: 21](#_Toc158366345)

[Introducción: 21](#_Toc158366346)

[Imágenes: 21](#_Toc158366347)

[Mejoras en un futuro 22](#_Toc158366348)

[Introducción 22](#_Toc158366349)

[Cosas a mejorar 22](#_Toc158366350)

[-Optimizar el programa Java (que la fecha actual salga automáticamente) 22](#_Toc158366351)

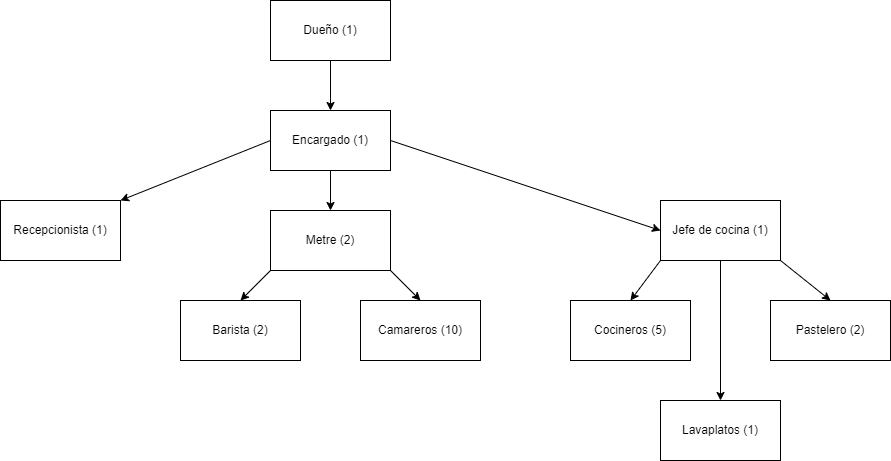
[Fallos y complicaciones 24](#_Toc158366352)

[Introducción: 24](#_Toc158366353)

[Problemas: 24](#_Toc158366354)

# FOL Formación y Orientación Laboral

Esta parte del trabajo fue el inicio del proyecto, una vez que pensamos cuál sería nuestra empresa (Restaurante), empezamos a investigar en varias páginas web cuántos empleados serían necesarios para una capacidad de unas 150 personas, en dichas páginas indican que serían necesarios alrededor de unas 21 personas (2 metres, 2 recepcionistas, 10 camareros y 5 chefs) sin contar al dueño y al encargado.

Posteriormente a los empleados les asignamos un organigrama profesional, es decir, asignamos una jerarquía del restaurante para tener un orden (en modo gráfica). 

Después de asignar los empleos del restaurante iniciamos el plan de prevención de riesgos laborales del restaurante gracias a la plantilla que teníamos en el classroom.

Para finalizar, realizamos una nómina a uno de los empleados del restaurante, gracias a una nueva plantilla que teníamos en classroom, a su vez, miramos en internet los convenios laborales que tiene en base al sector de la hostelería.

# ENGLISH

En la parte de inglés, debíamos realizar un video explicando los servicios de nuestra página web, es decir, de cara al cliente debemos hacer una presentación en inglés.

Alejandro Rebollo realizó la parte de la portada de la página web, Jon Zorzano presento la manera de realizar una reserva, Julen Lopez nos enseñó el menú del restaurante y, por último, Aarón Oliva y Mario Hernandez presentaron la pagina donde presentamos a nuestros chefs con una pequeña biografía.

# Lenguaje de Marcas

## HTML:

Para este apartado del trabajo comenzamos realizando en VSCodium las páginas web del restaurante (en total 4: Portada, Reservas, Menú y Chefs). Para el tema de enlazar las páginas unas con otras usamos en HTML: href y button, el primero hay que indicar a qué página debe llevar (para eso debemos tener los distintos html’s en la misma carpeta guardados) y luego usamos button para indicar que mensaje queremos que salga en la página.

Para tener organizado el HTML y el CSS hemos usado los div’s con el objetivo de dividir los objetos diferentes que aparecen en la página y de esta manera tenerlo más ordenado. Más tarde, hemos creado un Main con el objetivo de poder usar en el CSS los grid-row y grid-column (que ayudan mucho para poder colocar los objetos de la página).

Para las imágenes debemos de guardar en el mismo sitio que tenemos guardado los HTML y CSS para que el ordenador pueda localizar las imágenes a su vez en el HTML debes usar “img src” y la dirección donde está la imagen para que cargue dicha imagen.

También añadimos para las reservas del restaurante un formulario para dichas reservas, la estructura para hacer el formulario es sencilla. Por cada dato que pusimos añadimos un input, indicando el nombre del input, de que tipo es e indicamos que era obligatorio “required”. Al final del formulario añadimos el “button” enviar.

****

## XSD:

A parte del HTML, otra parte del trabajo era realizar un xsd y un xml en base a lo puesto en el formulario.

El xsd no nos llevó mucho tiempo, lo más importante, indicamos que era una secuencia, es decir, se tiene que respetar el orden de los elementos. A su vez, abrimos dos elementos, el elemento “restaurante” que sería el primer elemento y un segundo elemento llamado reserva con el objetivo de poder añadir un “maxoccurs:unbounded”, de esta manera podemos añadir las reservas que quieras.

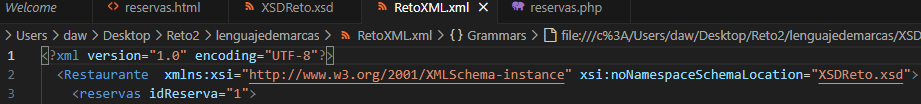
Dentro del elemento restaurante pusimos que era complexType porque hay varios elementos dentro al igual que el elemento reserva. En el elemento nombre pusimos un “whitespace” por la posibilidad de un nombre compuesto. Más adelante, para el elemento de correo electrónico os vimos en la obligación de poner un pattern, gracias a la profesora nos proporcionó dicho código del correo electrónico. El teléfono la única restricción que pusimos fue que como máximo solo acepta 9 números. Para los elementos fecha y hora indicamos como tipo date y time. Por último, añadimos el atributo de id ya que estos datos son sacados de la información insertada en el formulario.



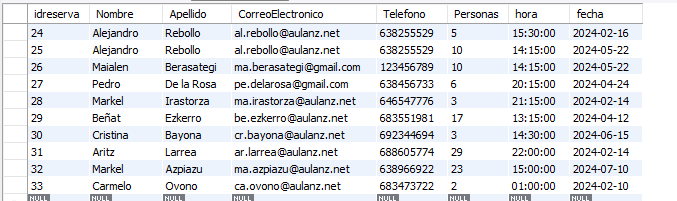
## XML:

Esta parte fue la más sencilla ya que al estar bien estructurado el xsd solo teníamos que indicar los datos que se habían hecho por el formulario.

Lo única posible complicación era conectar bien el xsd y el xml. Para ello, guardamos ambos archivos en la misma carpeta e indicamos en las dos primeras filas del XML la conexión con el xsd.



En la siguiente imagen se muestra la concordancia entre los datos del XML y la base de datos WORKBENCH: (Los datos son los más realistas que hemos podido obtener).



# Bases de Datos

## Introducción:

En este apartado del trabajo se nos pedía realizar una base de datos para las reservas y que estuviese la base de datos en un servidor web (es decir, fuera de Localhost).

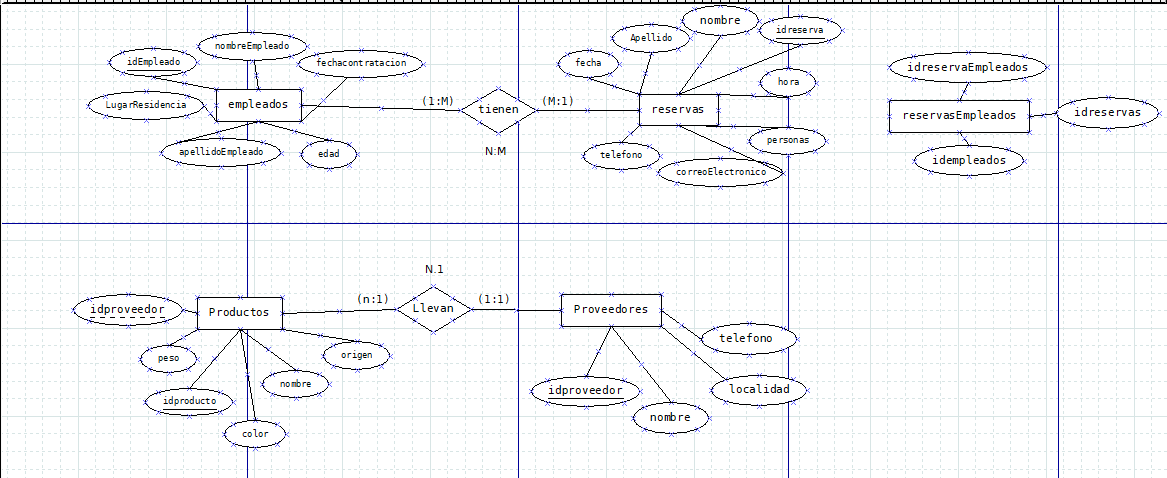
## Proceso llevado a cabo:

La parte de bases de datos ha sido una de las partes más complicadas del reto ya que debíamos realizar la conexión con el formulario de la página web y ambas cosas deben estar subidas en una máquina virtual (algo que no habíamos dado).

Nuestro grupo decidió crear la máquina virtual en AWS ya que en las últimas semanas habíamos trabajado en ello y la base de datos decidimos realizarla con PHP porque vimos que la conexión con PHP era más sencilla.

Pero antes de hacer la conexión debíamos crear la base de datos. Empezamos creando el diagrama entidad-relación de nuestra base de datos para asignar las tablas que queríamos y sus relaciones, respetando las cardinalidades para que luego en el PHP no nos de error.

Al diagrama le acabamos asignando 4 tablas (Reservas, Empleados, Productos y Proveedores) de dichas tablas sacamos las relaciones como: N:M (es decir, debíamos crear una tabla más añadiendo los 2 id de dichas tablas) y 1: N. Y a cada tabla le asignamos sus atributos.

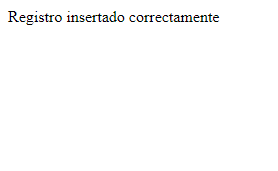


Más tarde, iniciamos XAMPP para poder acceder al PHPmyAdmin y desde ahí insertamos las tablas, atributos y por último las relaciones. Comprobamos que todo iba bien cuando en el formulario de la página e indicamos la ip de la base de datos y que cuando la enviábamos vemos que la información se había guardado bien.

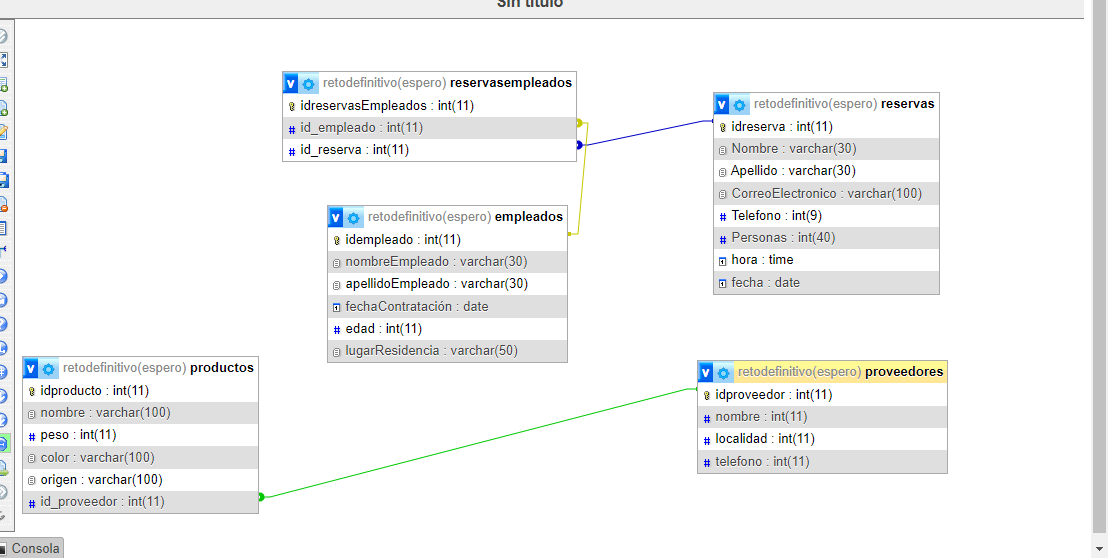
## Imágenes:







Por último, en la siguiente imagen se mostrará las tablas y sus relaciones en el PHPmyadmin. Los datos que no pertenecen a la tabla de reservas han sido insertados manualmente desde el PHP, sin embargo, los datos de la reserva han sido insertados desde el formulario.



Sin embargo, hubo complicaciones a la hora de conectar la base de datos con el programa Java y nos vimos obligados a realizar la base de datos en workbench.

Para este proceso tuvimos que crear un servidor de base de datos en AWS (RDS), una vez creada la instancia se nos abre el programa de Workbench que tendrá un usuario, contraseña y la ip pública en programa. Por último, teníamos que exportar la base de datos de PHPmyAdmin al workbench para no tener que volver a insertar todos los datos.

# Sistemas de Informáticos

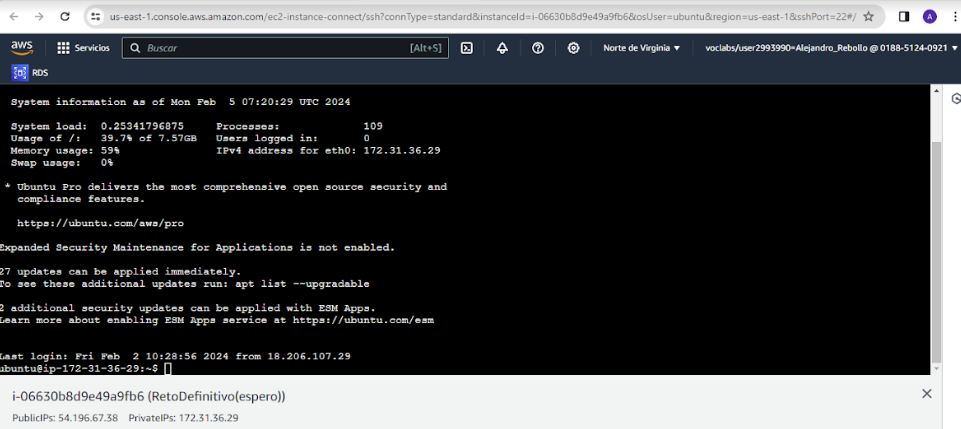
## Instalación del servidor:

La parte que corresponde a sistemas es la manera que hemos conectado la página web y la base de datos al servidor. Pero primero explicaré cómo creamos la máquina virtual en AWS Academy ya que veníamos trabajando en ello las últimas semanas.

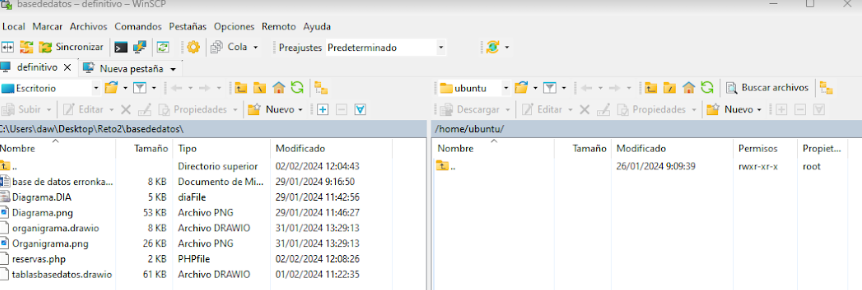
En AWS lanzamos una nueva instancia cuya aplicación de inicio sea UBUNTU, a su vez, pedimos una clave “PERM” para tener algo de seguridad y no pueda acceder fácilmente en el servidor. Posteriormente indicamos que tuviese acceso de tráfico de HTTPS para que se pueda acceder desde internet y no solo desde Localhost.

https://lh7-us.googleusercontent.com/s5ooP-wngFEZi_mT0VbkP8MD47hLnyEQx6VmJA5Qj3mY4CRRbdOWmQ6WasfLE8XU68R1Ci0l7ngulXwmg800_yp21mFxT6bV82uiZa9GMVI9t_zyGPyNsCZnUH7PTkRupcLWlRNkD_JiVaj65MRpAu0

A partir de aquí ya tenemos creado el servidor y tenemos que realizar ahora los pasos para subir la página web al servidor. Para comenzar, debemos instalar en el servidor APACHE2 con el comando “sudo apt install apache2” con el objetivo de poder asignar las páginas de la página web. (al instalarlo se nos crea en /VAR las carpetas /WWW y /HTML en esta última debemos meter las páginas web cuya página principal se debe llamar index.html.



Sin embargo, no es tan fácil, debemos descargarnos WinSCP para poder subir los archivos al servidor para ello debemos indicar en dicho programa la ip publica del servidor e indicar el usuario y la clave. Una vez que hemos accedido, podemos subir los archivos que queramos al servidor y hacer los pasos dichos anteriormente.



## Instalación del PHP:

Por otra parte, falta el proceso de instalación del PHP para poder subir la base de datos (ya creada) al servidor, nuestro grupo realizo dicha base de datos en PHP lo cual explicare de la manera para bases de PHP, la otra manera es con workbench.

Para subir la base de datos debemos instalar el PHP en el servidor de AWS siguiendo uno por uno los siguientes comandos:(importante hacer **SUDO -SU** para ser root).

Instalar el lenguaje de programación PHP

**apt install php**

Instalar el servidor de bases de datos MySQL

**apt install mysql-server**

Instalar el conector entre PHP y MySQL

**apt install php-mysql**

Instalar phpMyAdmin

**apt install phpmyadmin** (pide escoger servidor web=> Apache, y establecer contraseña de acceso)

Una vez realizado estos pasos ya tenemos instalado lo necesario para subir la base de datos. Lo primero, debemos añadir el siguiente comando que nos proporcionará acceder a una base de datos PHP creado por el propio servidor, donde tendremos que indicar un usuario y una contraseña.

**mysql -u user1 -p(Este comando nos abre el php)**

**create user 'user1'@'localhost' identified by 'pwuser1';**

**create database db1;**

**grant all on db1.\* to 'user1'@'localhost';**

Cuando accedemos al PHP ya simplemente debemos exportar desde localhost de la página PHP al servidor.

# Programación

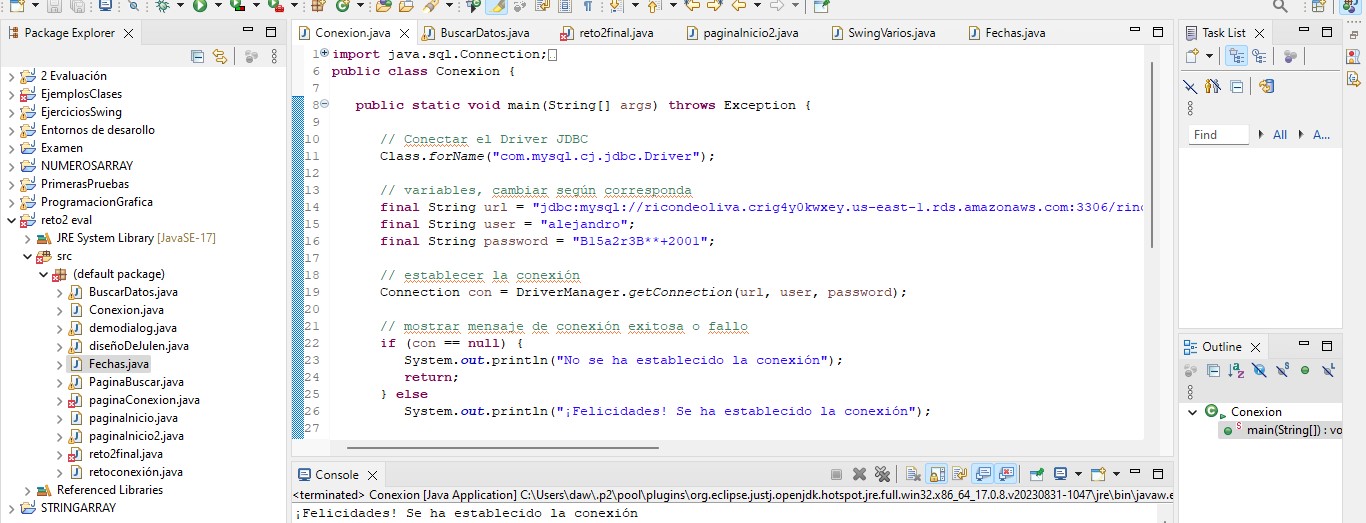
## Introducción:

Para la parte de programación se nos pedía crear un proyecto Java donde debíamos crear un programa gráfico donde se muestre las reservas de la base de datos del restaurante para que de cara al empleado puedan gestionar el tema de las reservas. A su vez, debíamos indicar el programa con comentarios(JavaDoc).

## Proceso llevado a cabo:(Página 1)

Nuestro trabajo se basa en la creación de 4 páginas Java, dichas páginas empezamos a desarrollarlas una vez que conseguimos subir la base de datos al servidor, en este punto, con el código que se nos proporcionó en el classroom creamos la primera página en la que el objetivo es que la conexión del programa Java con la base de datos es exitosa, donde en caso de que la conexión es buena se nos mostraría el siguiente mensaje:

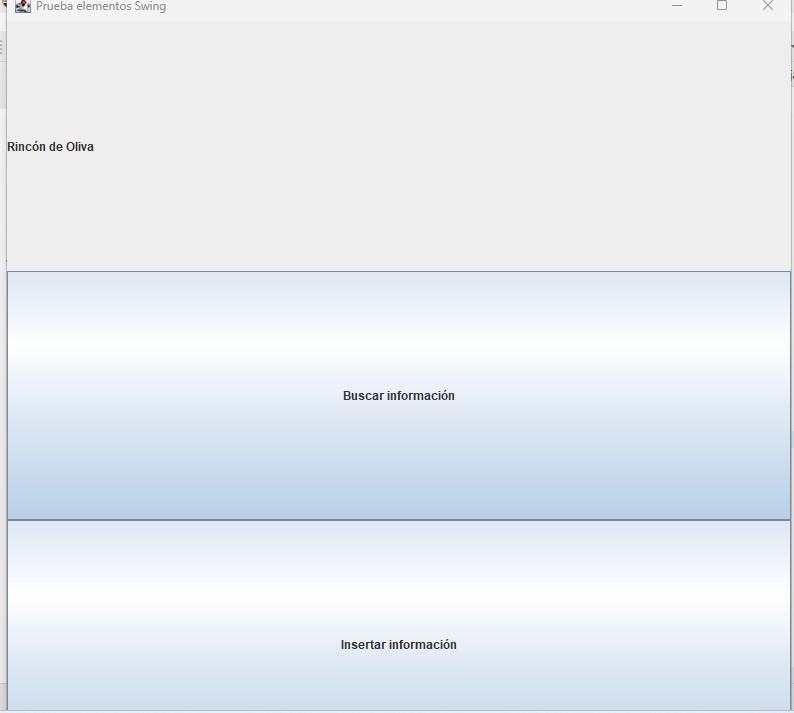
“! Felicidades ¡Se ha establecido conexión”

––––

### Página 2:

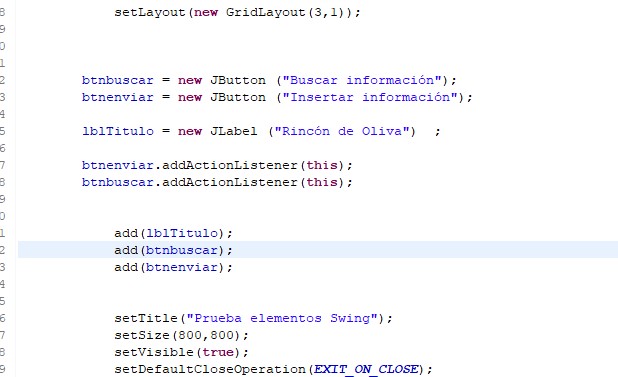
Una vez comprobada que la conexión es exitosa nos reunimos para saber cómo sería nuestro programa Java. La idea que decidimos fue añadir 3 páginas más donde la página 1 sería la siguiente:

Una página donde le mostraría al empleado, a parte del nombre del restaurante, dos botones donde saldrían una frase en cada botón (“Buscar información” y “Insertar información”.



Esta página tiene una estructura sencilla, se basa en la creación de un JLabel donde se muestra el nombre del restaurante y de dos JButton para que nos de la opción de pulsar y que nos lleve a las otras páginas.

En el constructor hemos indicado que sea GridLayout, es decir, que nosotros indicamos donde queremos que vaya el JLabel y los dos JButton. Después indicamos como queremos que sea la gráfica (Tamaño, sea visible…). Por último, indicamos con ADD en el orden que queremos que salga la información (en este caso no importa)



### Página 2

La segunda página de Java se basa en que muestre en forma gráfica los datos de la base de datos del servidor.

El diseño es sencillo, hemos necesitado un JButton, JTable para indicar los campos, String[] para indicar la dimensión de los datos que se muestren. En el constructor hemos indicado que es FlowLayout y los demás elementos.

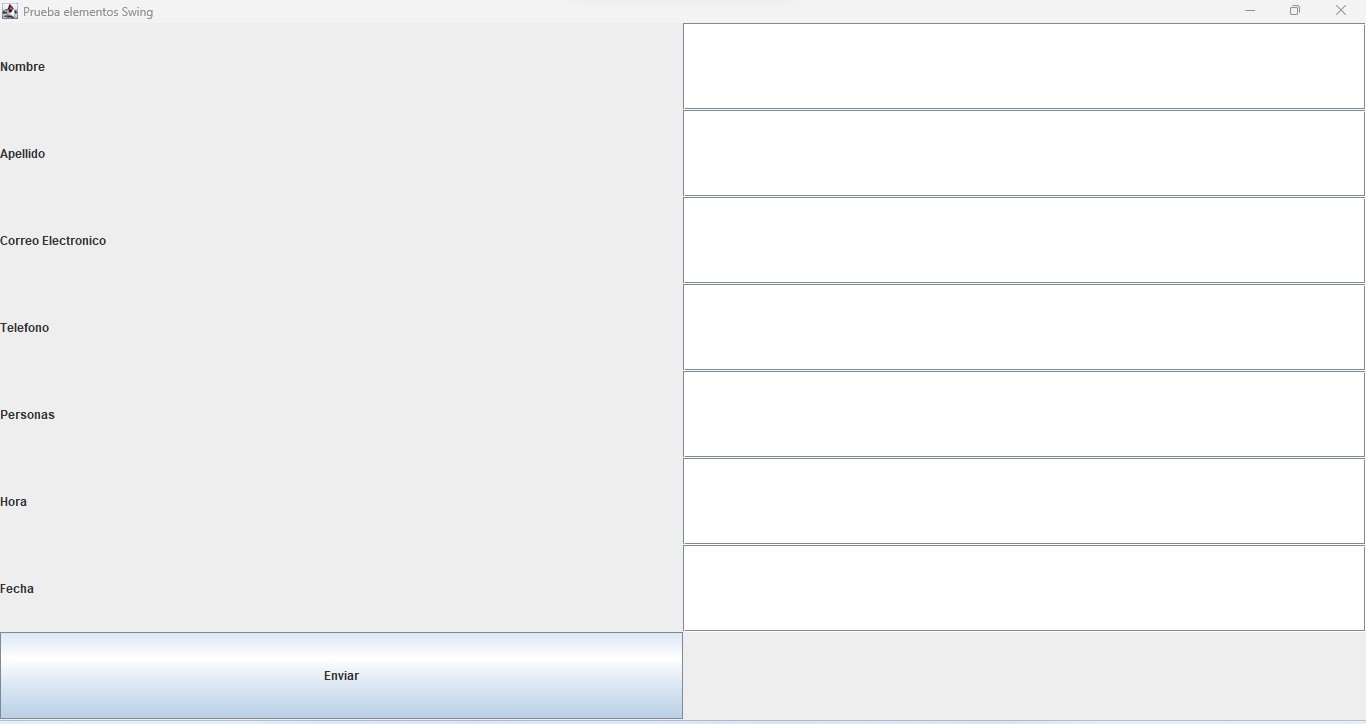
Sin embargo, la parte clave de la página está a la hora de la conexión ya que indicamos en el main el código de la conexión y tenemos que indicar los campos de que tipo son y qué campo tiene que coger de la base de datos.



### Página 3

En esta página hemos intentado insertar los datos, pero a falta de tiempo e información no hemos podido completar el código. Aunque, dejamos una captura del diseño donde se ve la estructura de la gráfica.





# Entornos de desarrollo:

## Introducción:

Lo que se nos pedía era realizar el trabajo en conjunto mediante Git bash y añadir a los programas Java comentarios (JAVADOC).

## Imágenes:



## Javadoc:

También hemos añadido a los programas java comentarios javadoc para explicar lo que hemos hecho a lo largo de cada programa java.

Y después hemos comprobado mediante eclipse que genere un javadoc he aquí una prueba del resultado.



# Mejoras en un futuro

## Introducción

Hemos tomado la decisión de crear un apartado dedicado a la formulación de ideas y objetivos futuros, fundamental para un posible avance en este proyecto. Esta adición no solo servirá como documento archivado, sino también como una guía organizada para facilitar el progreso coherente en caso de retomar el proyecto en el futuro. La idea es tener a mano un documento que facilite el proceso y nos sirva de ayuda en nuestro desarrollo.

Creemos que actualmente está bastante bien realizado el trabajo, pero por el tiempo, se nos han quedado algunas ideas en el aire y que sí podríamos añadir en un futuro tanto como en mejorar la página web, mejorar la base de datos, ...

## Cosas a mejorar

### -Optimizar el programa Java (que la fecha actual salga automáticamente)

Este objetivo se refiere a que cuando los empleados quieran ver las reservas de ese mismo día no tengan que escribir dicha fecha, es decir, que la fecha salga de forma automática.

-Separar las bases de datos (comidas y cenas)

Con este objetivo intentamos mantener un mayor orden en la base de datos al separar las comidas con las cenas.

-Ordenar las reservas, añadir en qué mesa se sitúa cada cliente

Con este objetivo queremos asignar un número a cada mesa con lo cual los empleados sepan a qué mesas atienden.

-Diseñar mejor el programa HTML

Simplemente queremos mejorar el diseño HTML de nuestra página ya que creemos que nuestra página resulta en cierta manera vulgar.

Ya que queremos que el cliente pueda apreciar el diseño de la página y que digamos en cierta manera la gente le sorprenda nuestro diseño. Estos detalles suelen ser diferenciales a la hora que la gente tenga que elegir que página quiera revisar.

-Hacer un apartado especial para que la gente pueda poner sus fotos y reseñas

Añadir una nueva página donde los clientes puedan subir fotos dentro del establecimiento comiendo o cenando y que además puedan añadir una pequeña reseña explicando su experiencia.

-Mejorar la accesibilidad

Intentar optimizar la página lo máximo posible para que al cliente y empleado les resulte sencillo navegar tanto por la página como por el programa Java.

# Fallos y complicaciones

## Introducción:

En este apartado vamos a explicar todos los problemas que hemos tenido a lo largo de estas 3 semanas y de qué manera los hemos solucionado (no hemos podido resolver todos los problemas).

## Problemas:

La conexión de phpmyadmin al servidor AWS, al insertar las paginas en el servidor solo nos quedaba añadir la base de datos, sin embargo, el servidor requería la instalación de varios paquetes de mysql y php, comandos que desconocíamos por completo.

Una vez consultado con el profesor nuestro problema nos proporcionó dichos comandos para la instalación de phpmyadmin y de esta manera conseguimos solucionar dicho retraso.

Otro problema que hemos tenido ha sido a la hora de intentar conectar el programa java con la base de datos situada en phpmyadmin. EL profesor nos proporcionó un código java para comprobar que la conexión era exitosa, al realizar todos los pasos que nos había indicado el profesor, sin embargo, no se efectuaba la conexión. Nos dimos cuenta gracias a una página de internet al añadir nuestra ip publica nos indicaba que nuestro puerto 3306 estaba cerrado, sin embargo, habíamos indicado en la instancia del AWS que permitiese todo el tráfico. Este problema se lo comentamos al profesor y no encontró donde estaba el error y que la única solución era hacerla en un servidor RDS.

Los pasos de la instalación del RDS ya han sido comentados en el apartado de bases de datos.

Otra complicación que tuvimos fue añadir el código de la conexión con la base de datos al programa grafico de java, ya que no sabíamos dónde teníamos que insertar dicho código en el programa.

Intentamos insertarla como una función, pero no lo conseguimos después lo intentamos añadir en el actionPerformed, sin embargo, el programa nos indicaba errores que no podíamos solucionarlos. Finalmente, gracias a un compañero de clase nos ayudó diciéndonos que lo debíamos poner en el main, a parte nos indicó que debíamos poner el actionPerformed.

Por último, el profesor nos indicó que era un buen detalle añadir un programa java para insertar datos desde java en la base de datos. Tras muchos intentos no conseguimos conectar con la base de datos ya que la estructura pensábamos que era similar a la de extraer datos de la base de datos, ni con las explicaciones del profesor, no conseguimos que el programa funcionase a pesar de entender cómo sería la estructura del programa.

Y a su vez, por la falta de tiempo y sabiendo que este programa no era obligatorio crearlo decidimos abandonarlo por ahora.