



C.P.R. Liceo "La Paz"

Proyecto Fin de Ciclo

Desarrollo de Aplicaciones Multiplataforma

Autor : Diego Castreje Domínguez
Tutor : Jesús Ángel Pérez-Roca Fernández

Contenido

Resumen.....	3
Palabras clave.....	4
Motivación.....	5
Estado del arte.....	6
Viabilidad tecno-económica.....	7
Diagramas.....	8
Conclusiones.....	9

Resumen

Este proyecto es una aplicación web que simula el programa que te puedas encontrar en cualquier restaurante o bar para realizar todas las cuentas y gestiones de productos y empleados del local.

La aplicación te mostrará una ventana de login para introducir un usuario y contraseña. Si no tienes una cuenta creada no podrás acceder ya que no puedes registrarte, debe darte de alta una cuenta el jefe.

En la vista principal tienes unos botones con números que representan las mesas que tendrías en tu restaurante, a las que le puedes añadir productos una vez entres en ellas. Además también hay una barra de navegación en la parte inferior de la vista, con la cuál puedes volver a la vista principal, ver la fecha de hoy y acceder a la ventana de configuración.

En la ventana de configuración tendrás tres opciones: empleados, productos y cerrar sesión. Sólo puedes acceder a las dos primeras si eres el jefe, ya que en la primera opción se verá el listado de todos los empleados con sus respectivos datos y se podrán añadir nuevos empleados a tu plantilla; y en la segunda opción se podrá ver el listado de productos y también añadir nuevos productos para poder venderlos en las mesas.

La tercera opción de la ventana de configuración cierra sesión con el usuario con el que habías iniciado sesión y te devuelve a la ventana de login.

Palabras Clave

Java: Lenguaje de programación orientada a objetos.

Base de Datos: Conjunto de tablas y datos almacenados.

BackEnd: Término utilizado para referirse a la parte del servidor de la aplicación, el cual está conectado a la base de datos.

FrontEnd: Término utilizado para referirse a la parte visual de la aplicación, con la que el usuario tendrá contacto directo.

Spring: Framework de código abierto para la creación de aplicaciones empresariales Java.

Hibernate: Modelo de programación que permite mapear las estructuras de una base de datos relacional.

MySQL: Sistema de gestión de bases de datos relacionales de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado.

Typescript: Lenguaje de programación libre y de código abierto desarrollado, mantenido por Microsoft y que es un superconjunto de JavaScript.

Angular: Framework para aplicaciones web desarrollado en TypeScript, de código abierto y mantenido por Google.

Peticiones HTTP: Las peticiones HTTP son mensajes enviados por un cliente, para iniciar una acción en el servidor.

Código de Http Status: Son mensajes del servidor que indican el resultado de la petición que se trató de llevar a cabo.

REST: La transferencia de estado representacional (REST) es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web.

Se usa para conectar varios sistemas basados en el protocolo HTTP y nos sirve para obtener y generar datos y operaciones, devolviendo esos datos en formatos muy específicos, como XML y JSON.

HTML: lenguaje descriptivo que especifica la estructura de las páginas web.

CSS: Lenguaje para la composición y estructuración de páginas web.

Bootstrap: Framework desarrollado en CSS utilizado en el FrontEnd.

Postman: Herramienta usada para el testing de aplicaciones API REST.

Git: Es un software de control de versiones.

Controlador: Clase en el BackEnd que mediante peticiones HTTP lanzadas desde el FrontEnd realizará acciones en la base de datos o simplemente accederá a la información de la misma.

Servicio: Clase Java que sirve para comunicar mediante métodos al Controlador con el Dao.

DAO: El Data Access Object (DAO) permite separar la lógica de acceso a datos al resto de la aplicación porque implementar la información puede variar según la fuente de datos, y con el DAO ya no tienes que lidiar con la lógica de negocio en sí.

Constructor: Método de una clase Java utilizado para crear objetos de dicha clase.

Getter: Método para devolver el valor de un atributo de un objeto.

Setter: Método para dar un valor al atributo de un objeto.

POJO: Clase de Java formada por atributos, constructores y métodos getter y setter.

Empleado: Usuario el cuál debe ser registrado para poder acceder a las funcionalidades de la aplicación.

Productos: Objetos que ofertamos en nuestra aplicación.

Mesa: Entidad en la cual se almacenarán los productos.

Login: Pantalla en la que el empleado debe registrarse para acceder a las funcionalidades de la aplicación.

Motivación

La idea de este proyecto surgió a raíz de mi experiencia en la hostelería y de los diferentes programas que había tratado. Tuve en cuenta las diferentes funciones que tenían y me pareció que podría ser algo muy interesante de realizar. Además de que vi la posibilidad de hacer un proyecto más completo al poder hacer una base de datos algo más compleja que las realizadas durante el curso.

Las tecnologías que uso en el proyecto las aprendí por cursos llevados a cabo durante mi etapa como estudiante en formación en Atos, las cuáles a pesar de ser completamente nuevas para mí en su mayoría me gustaron mucho por ver en ejemplos lo útiles que podían llegar a ser, ya que lo vi más cercano a un proyecto profesional que lo que había realizado anteriormente.

Estado del Arte

Se encuentran muchas opciones similares a esta aplicación en el mercado, cualquiera que puedas ver en el ordenador de un bar o restaurante seguramente se parezca en grandes rasgos a esta. La diferencia más notable a destacar es que esta es una versión web, lo cuál puede facilitar el acceso a la aplicación desde diferentes dispositivos, ya que si los empleados tienen tablets para registrar pedidos y demás funciones les será más cómodo, porque normalmente el resto de aplicaciones de bares suelen tener que validarse con claves y sincronizarse, pero al estar como diferentes usuarios en la web ya están sincronizados en todo momento.

Además estas aplicaciones no suelen tener un inicio de sesión antes de entrar, por lo que cualquier empleado que entre en el ordenador podría acceder a funciones que puedan causar problemas en la administración del negocio. Añadiendo que este tipo de aplicaciones tampoco tienen diferenciación de roles dentro de ella, lo cuál da muchas opciones para posibles actualizaciones a este proyecto.

En cuanto a las tecnologías usadas:

-Base de Datos: Para la administración de bases de datos usé Hibernate en Java con el lenguaje de MySQL, el cuál me permitió crear las tablas y sus relaciones mediante anotaciones en clases POJO en el proyecto. Una alternativa a esto podría ser el uso de JDBC, pero preferí Hibernate porque me pareció más cómodo y evitaba el uso de código MySQL. Además de MySQL existen muchos tipos de lenguajes de bases de datos para poder usar como PostgreSQL o OracleSQL.

-BackEnd: En cuanto al funcionamiento del servidor para este proyecto decidí usar Java ya que es el lenguaje que más he empleado y con el que tengo más soltura y conocimientos a estas alturas del ciclo. Además también usé el framework Spring que me facilitó el uso de controladores y de servicios mediante anotaciones en el proyecto, a parte de toda la seguridad con Spring Security. En vez de este método se podría usar otro con lenguajes como Python o C++.

-FrontEnd: Para hacer la parte del cliente del servidor usé en la estructura de las vistas HTML, y para su estética empleé CSS y librerías como Bootstrap. Para su lógica hice uso del lenguaje TypeScript, el cuál con sus clases service me permitió una fácil comunicación con la parte del servidor.

-Ayudas: Para facilitar el comprobar el funcionamiento del BackEnd utilicé Postman, y para guardar el progreso y poder volver atrás cuando ocurría algún problema hice uso de Git.

Viabilidad tecno-económica

La aplicación es realizada por una sola persona a cargo, lo cuál reduce mucho los costes de producción.

Es necesario comprar un host para el servidor de la página, y su precio dependerá de los usuarios que lo usen a la vez, ya que cuantos más usuarios mayor rendimiento necesitará, pero de momento este no superará los 10€ al mes. Además quien desee acceder a la aplicación deberá tener un equipo informático y acceso a internet.

Este es un análisis para determinar las partes fuertes y débiles de la aplicación:

Análisis DAFO

-Debilidades: Escasos recursos para promocionar la aplicación web.

-Fortalezas: Al ser todo creado por una única persona existe un mayor control sobre la página web, no necesita inversión inicial y por ello se puede ofertar a buen precio.

-Amenazas: Existen muchas otras aplicaciones similares más extensas en el mercado.

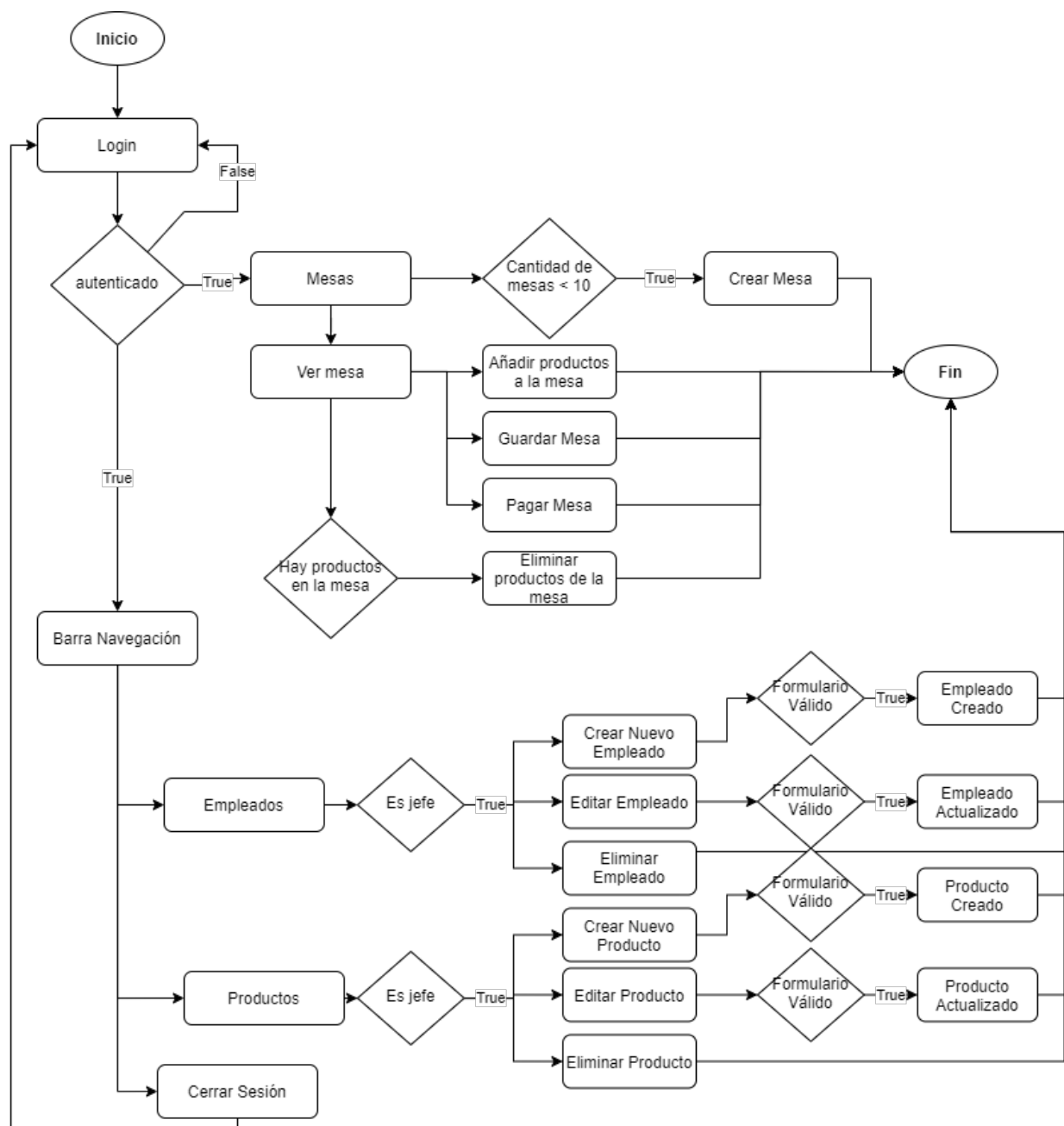
-Oportunidades: Después de la reapertura de la hostelería y los pocos fondos económicos que tendrán la mayoría de autónomos, el bajo precio de la aplicación puede resultar atractivo.

Para terminar, este proyecto no incumple ninguna ley y su uso puede ser aprobado sin ningún inconveniente por la fiscalidad, normativa legal, derecho mercantil ni la contabilidad.

Diagramas

Diagrama de Flujo

[Link de descarga](#)



Mockups

Estos son los mockups que tomé como modelo para realizar la vista de la aplicación.

Mockup Vista Login

Mi Restaurante

Inicio de Sesión

Usuario

Contraseña

Entrar



Para acceder al uso de la aplicación usted debe trabajar aquí.
Si es así y aún no tiene cuenta comuníquese a su jefe.

Diego Castreje

Mockup Vista Principal

Mi Restaurante

1	2	3	4
5	6	7	8
9	10	11	12

Home

Fecha

Configuración

Mockup Vista Mesa

Mi Restaurante

Mesa

Volver

Empleado

Nombre Empleado

Añadir producto

Producto	Cantidad	Precio
Producto	Cantidad	Precio

Precio Total

Pagar

Home

Fecha

Configuración

Mockup Vista Menú Configuración

Mi Restaurante

Empleados

Productos

Cerrar Sesión

Home

Fecha

Configuración

Mockup Vista Listado Empleados

Mi Restaurante

Listado de empleados

Nuevo empleado

Nombre	Apellido	Horas	Sueldo	Editar	Eliminar

Home

Fecha

Configuración

Mockup Vista Formulario Empleados

Mi Restaurante

Añadir Empleado

Usuario

Nombre

Email

Contraseña

Apellido

Salario

Crear

Home

Fecha

Configuración

Mockup Vista Listado Productos

Mi Restaurante

Listado de productos

Nuevo producto

Nombre	Tipo	Precio	Stock	Editar	Eliminar

Home

Fecha

Configuración

Mockup Vista Formulario Productos

Mi Restaurante

Añadir Producto

Nombre

Precio

Stock

Tipo de Producto

Comida

Bebida

¿El producto lleva alcohol?

Alcohol

Crear

Home

Fecha

Configuración

Diagrama Entidad-Relación

[Link de descarga](#)

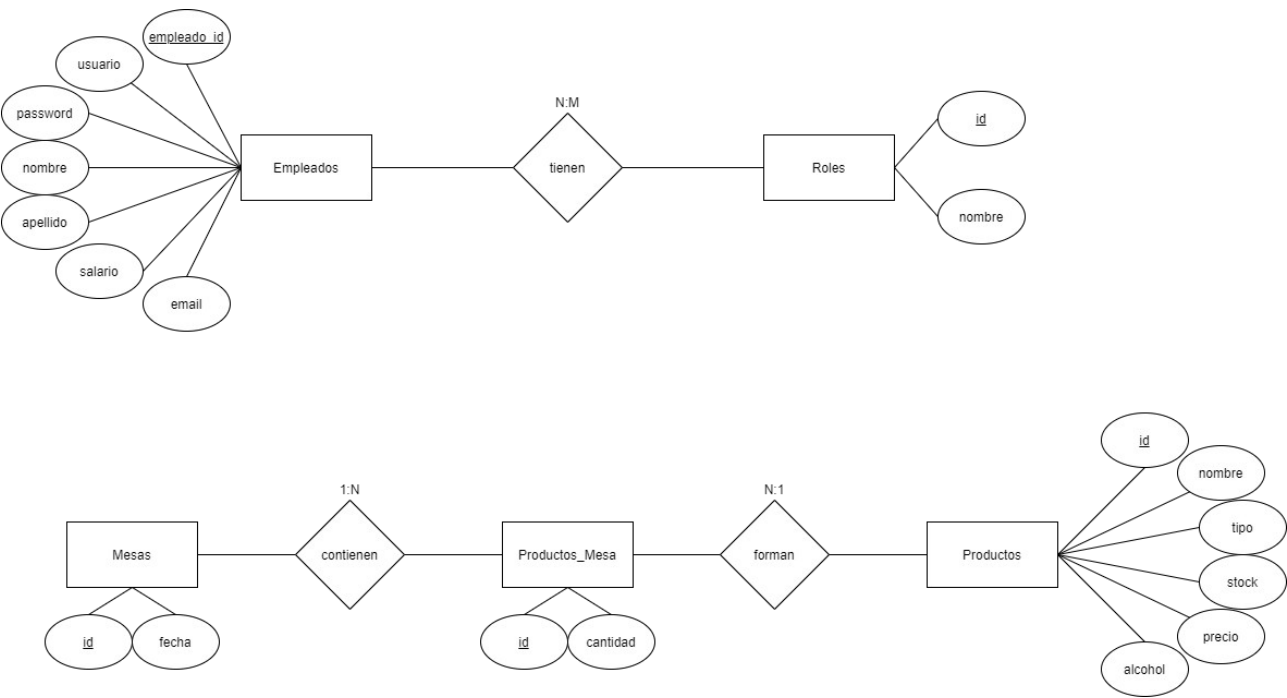


Diagrama de Clases

[Link de descarga](#)

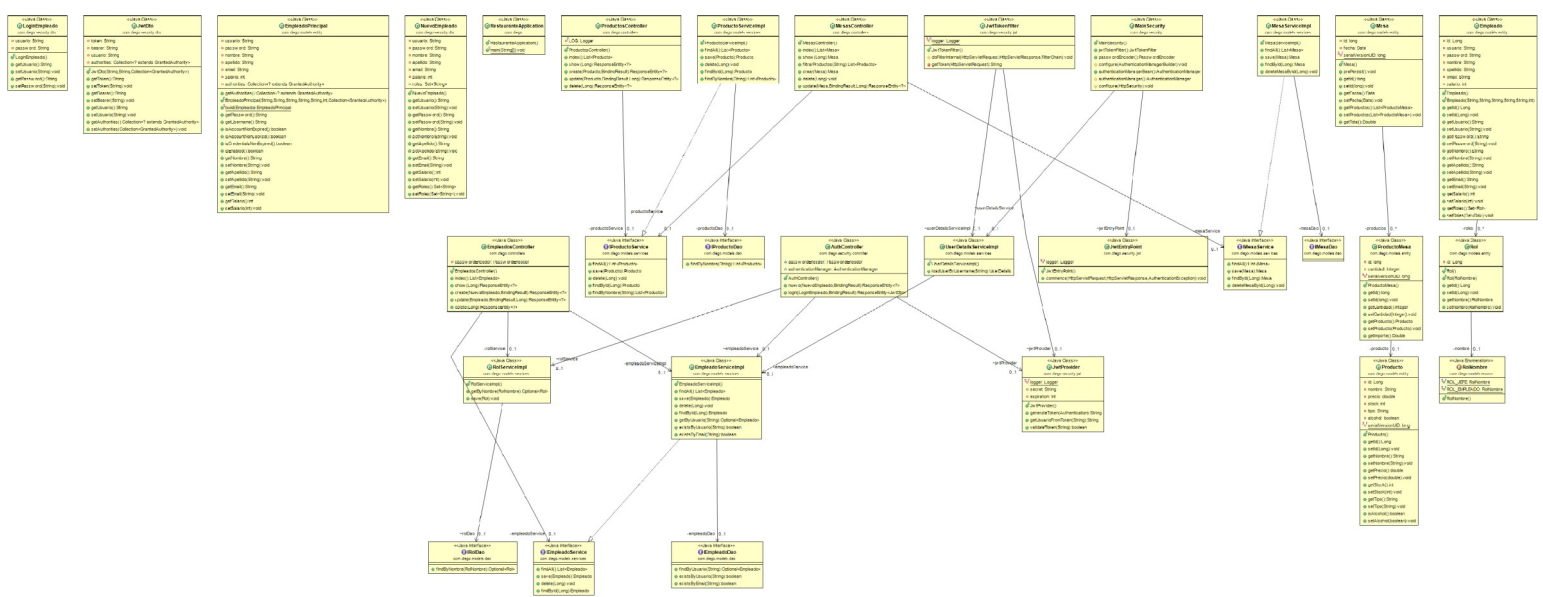
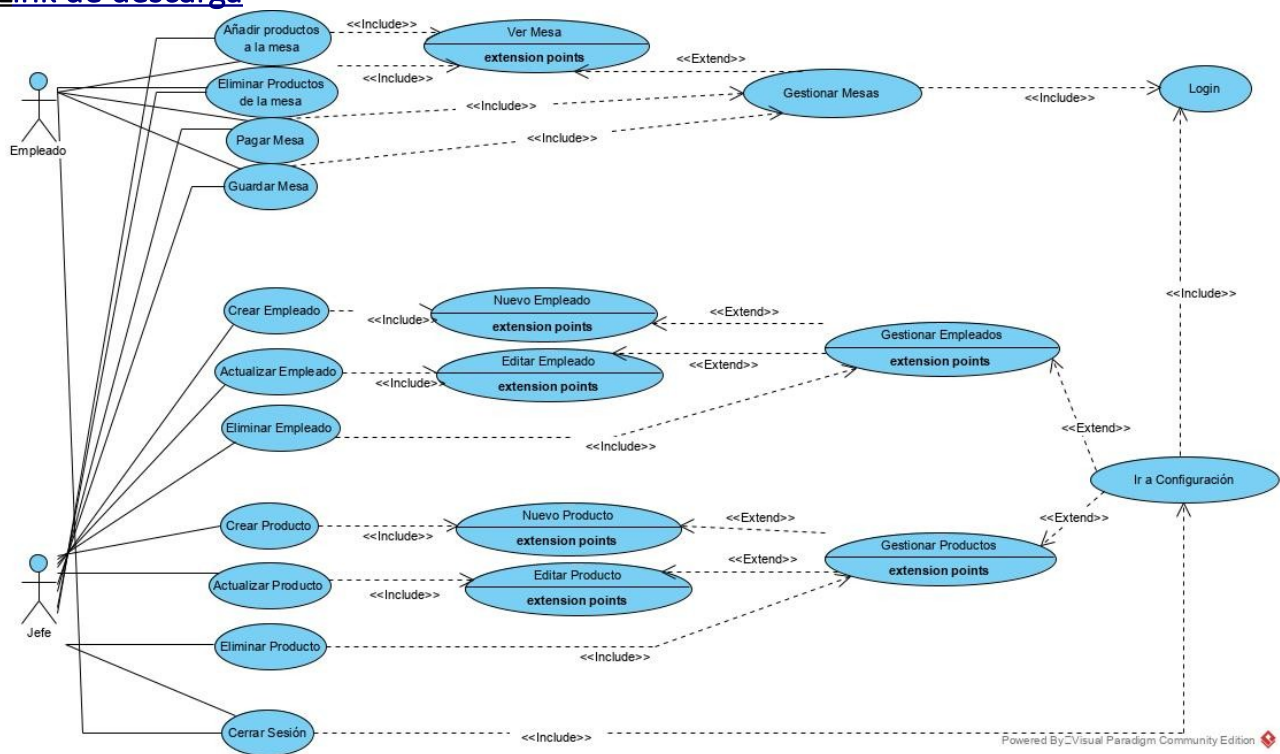


Diagrama Casos de Uso

Link de descarga



Conclusiones

Este proyecto me ha sido muy útil para aprender mucho sobre varias tecnologías que me podrán ser útiles en el futuro, y a entender mejor el funcionamiento de las peticiones http y la comunicación entre BackEnd y FrontEnd. He conseguido cumplir casi todo lo que me propuse con el proyecto, solo me han faltado tres detalles que no he conseguido hacer diferente:

-Mesas: Me habría gustado haber adaptado las mesas con botones pero he tenido problemas a la hora de listar las mesas de la base de datos si lo quería hacer de esa forma, no conseguí que mantuvieran un orden determinado para mostrarse. La solución que planteé fue hacerlo en una tabla normal y con un botón en una de las columnas de la tabla para poder acceder a ella.

-Detalle Mesa: Al añadir productos a la mesa no se actualiza el total de la cuenta porque no conseguí que accediera al método de la clase de calcularTotal, por lo que la solución que tomé fue añadir un botón de Guardar Mesa que sirve para actualizar la mesa en la base de datos y se actualizará también el precio así.

-Edición de Mesas: Quería añadir un modo de edición para las mesas pero al no poder listarlas en unos botones y hacerlo en una tabla tuve que descartar esta opción.

Por el resto de cosas he conseguido implementar todo lo que me propuse, y me si el tiempo me lo hubiera permitido me habría gustado añadir más funciones que aprendí durante las prácticas, como el filtrado por campos en las listas o que al hacer click en el nombre de la columna ordene la lista en funcion de ese campo, en orden ascendente o descendente. También tenía ideas como añadir paginación o fotos para cada producto, pero el tiempo no me lo permitió.

En cuanto al aspecto gráfico estoy muy satisfecho con la estética de la aplicación, he tratado de contar con opiniones externas sobre ella y también han sido positivas, por lo que no querría añadir cambios en este aspecto.

El rendimiento al tan solo contar con unos pocos datos en la base de datos y un solo cliente conectado al servidor no es necesaria una mejora del rendimiento, si en un futuro se amplia la aplicación y se debe mejorar el rendimiento intentaré eliminar métodos que puedan provocar lentitud en el programa como pueden ser algunos bucles, implementar la paginación para no acceder a todos los datos de una tabla de golpe, etc.

En conclusión estoy muy contento con el resultado de la aplicación, y con la facilidad de expansión que tiene, por lo que espero poder hacer una presentación decente y que también agrade a quienes la conozcan.