RestauranTech



**Nombre**: Javier Sanz Rozalén

Alberto Gómez García

**Asignaturas**: [AD][DI][SGE]

**Profesores:** Javier Martín Aznar

Jose Ramón Simó Martíne



**ÍNDICE**

1. **INTRODUCCIÓN**
2. **TECNOLOGÍAS EMPLEADAS**
3. **DISEÑO DE LA BASE DE DATOS**
4. **ESTRUCTURA DE LAS CLASES**
5. **EXPLICACIÓN DEL CÓDIGO**
6. **MANUAL DE USUARIO**
7. **CONCLUSIÓN**
8. **BIBLIOGRAFÍA**

**1.INTRODUCCIÓN**

Este es un trabajo de investigación y desarrollo que contempla la idea de afianzar los conocimientos aprendidos en las asignaturas de Acceso a Datos, Sistemas de Gestión Empresarial y Diseño de Interfaces.

Consideramos que es un trabajo de investigación debido a que vamos a aprender por nuestra propia cuenta cosas nuevas sobra las tecnologías que hemos aprendido en clase para llevar a cabo la idea propuesta. También consideramos que es un trabajo de desarrollo debido a que el objetivo final del proyecto es ver un producto funcional finalizado.

El equipo de desarrollo (Alberto Gómez y Javier Sanz) decidió que la temática del proyecto se basase en un Sistema de Gestión de un restaurante debido a que conocemos la competencia y pensamos desde un inicio que podría ser un proyecto motivador.

**2.TECNOLOGÍAS EMPLEADAS**

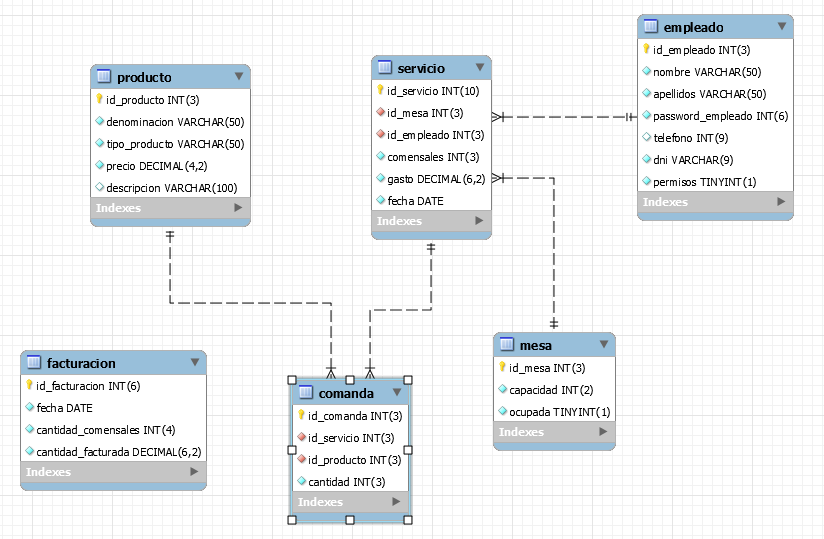
Para el desarrollo de este proyecto hemos hecho uso de las tecnologías aprendidas en clase. Consideramos que ha sido buena idea elegir estas tecnologías ya que la realización del proyecto nos permitiría aumentar nuestros conocimientos en ellas y salir al mercado laboral con una base más sólida.

|  |  |  |
| --- | --- | --- |
| **TECNOLOGÍAS** | | |
| **Nombre** | **Descripción** | **Logo** |
| **Java** | Lenguaje de programación de propósito general, concurrente, orientado a objetos que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. | Resultado de imagen de java se |
| **Java Swing** | Biblioteca gráfica para Java. Incluye widgets para interfaz gráfica de usuario tales como cajas de texto, botones, desplegables y tablas. | Resultado de imagen de java swing logo |
| **Hibernate** | ORM que facilita el mapeo de atributos entre una BD relacional y el modelo de objetos de la aplicación. | Resultado de imagen de hibernate |
| **MySQL** | Sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y considerada como la base datos open source más popular del mundo. | Resultado de imagen de mysql |
| **Git** | Software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. | http://blog.corunet.com/wp-content/uploads/2013/12/git4.jpg |

**3.DISEÑO DE LA BASE DE DATOS**

Antes de realizar cualquier parte del proyecto, hicimos el análisis y diseño previo de la Base de Datos en función de los requerimientos que nuestra aplicación nos iba a demandar.

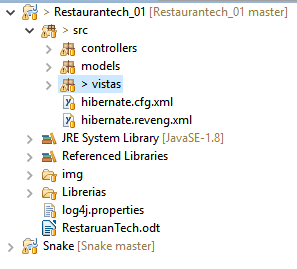
Podemos observar como las tablas Empleado, Producto y Servicio están relacionadas entre sí a través de la tabla servicio. Además, tenemos la tabla Mesa que quedará reflejada también en la tabla Servicio a través de una clave ajena y una tabla facturación que registrará los servicios de cada mesa al ser finalizados.

****

Para la generación de este diagrama he hecho uso de la herramienta MySQL Workbench, ya que me proporciona una solución de diseño que auto-genera este diagrama y me permite editarlo y añadir anotaciones.

**4.ESTRUCTURA DE LAS CLASES**

Para la estructuración de este proyecto decidimos establecer desde un inicio la arquitectura MVC (Modelo Vista Controlador), afianzando de esa manera el acoplamiento y la modularidad.



El MVC es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.

Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Diagrama de clases de los modelos:

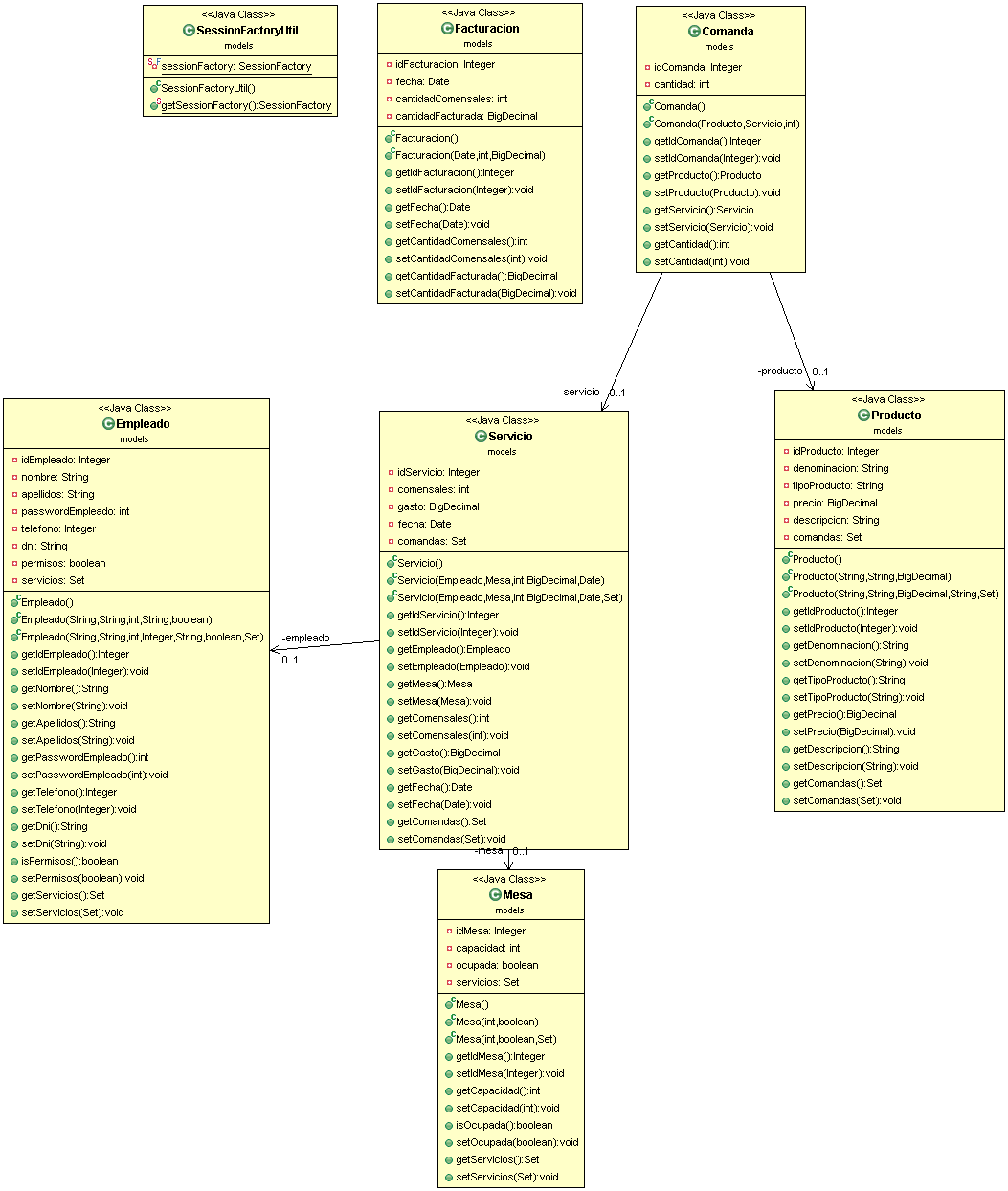
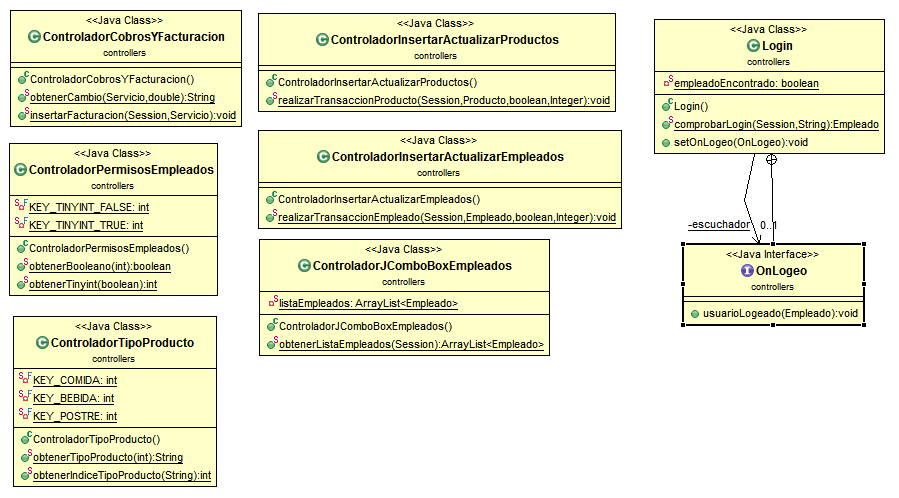


Diagrama de clases del controlador:



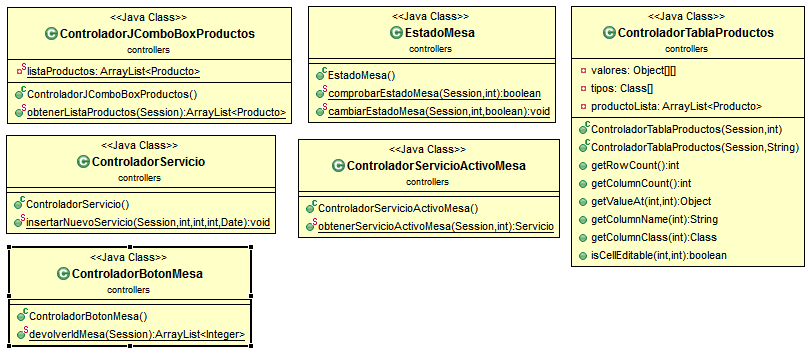
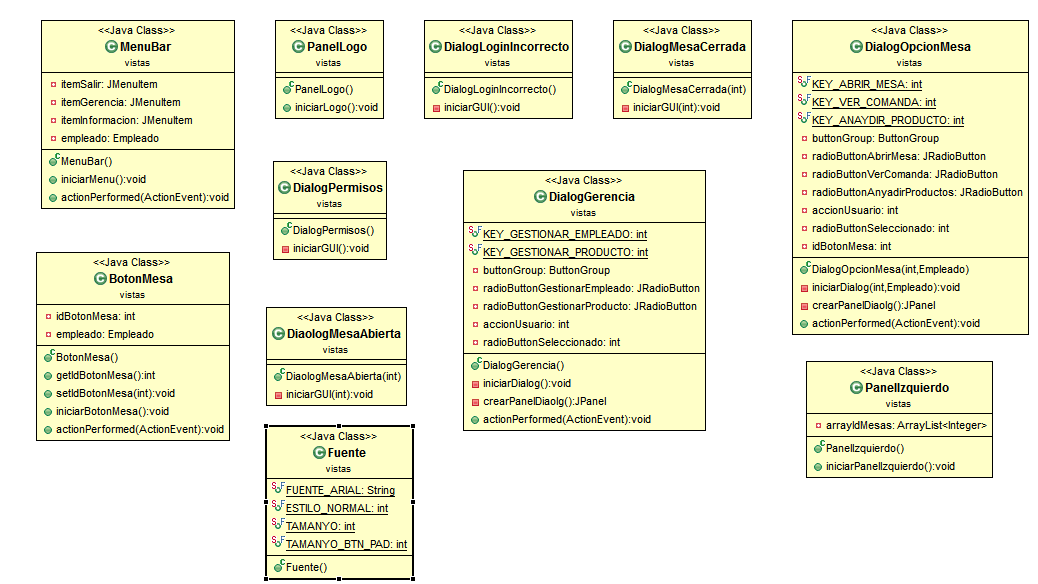
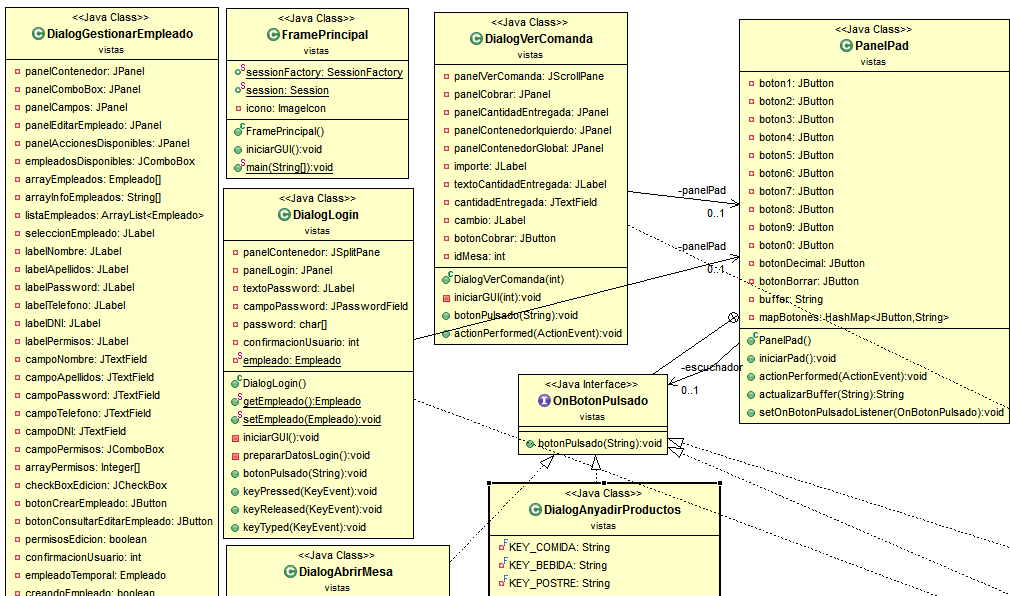


Diagrama de clases de la vista:



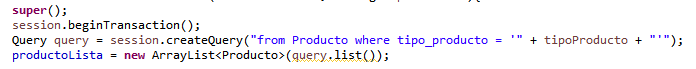


**5.EXPLICACIÓN DEL CÓDIGO**

En cuanto al uso y acceso a la base de datos, hay cuatro ciertas partes muy importantes en nuestro proyecto:

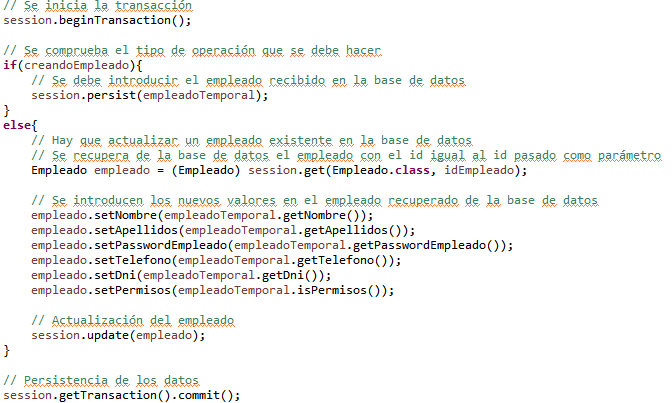
Inserción de datos en una tabla con Hibernate

Para ello deberemos hacer uso de un objeto Query que hará la consulta y almacenar el resultado en una Lista para que nuestra tabla rellene cada fila con la lista:



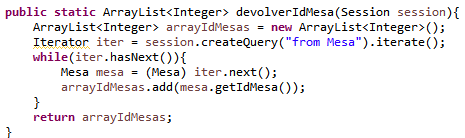
Inserción/modificación de empleados y productos

Para ello deberemos de persistir un nuevo objeto (en caso de inserción) o obtener el objeto que queremos modificar y hacer un set para aquellos atributos que deseemos modificar:



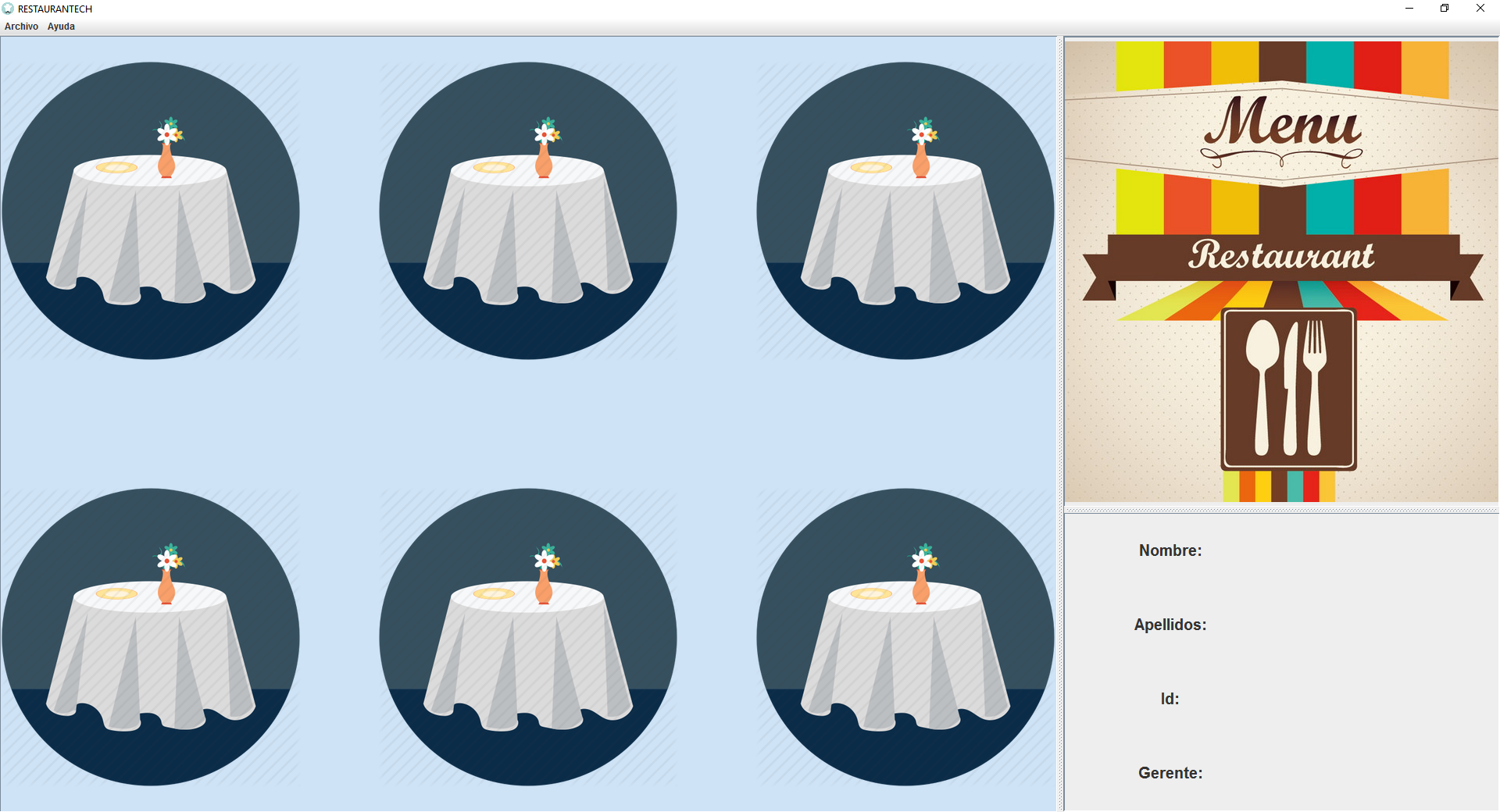
Generador de mesas

Crearemos nuevos objetos mesa que recogerán sus atributos desde la base de datos. Estos objetos mesa será utilizados como JButton personalizados:

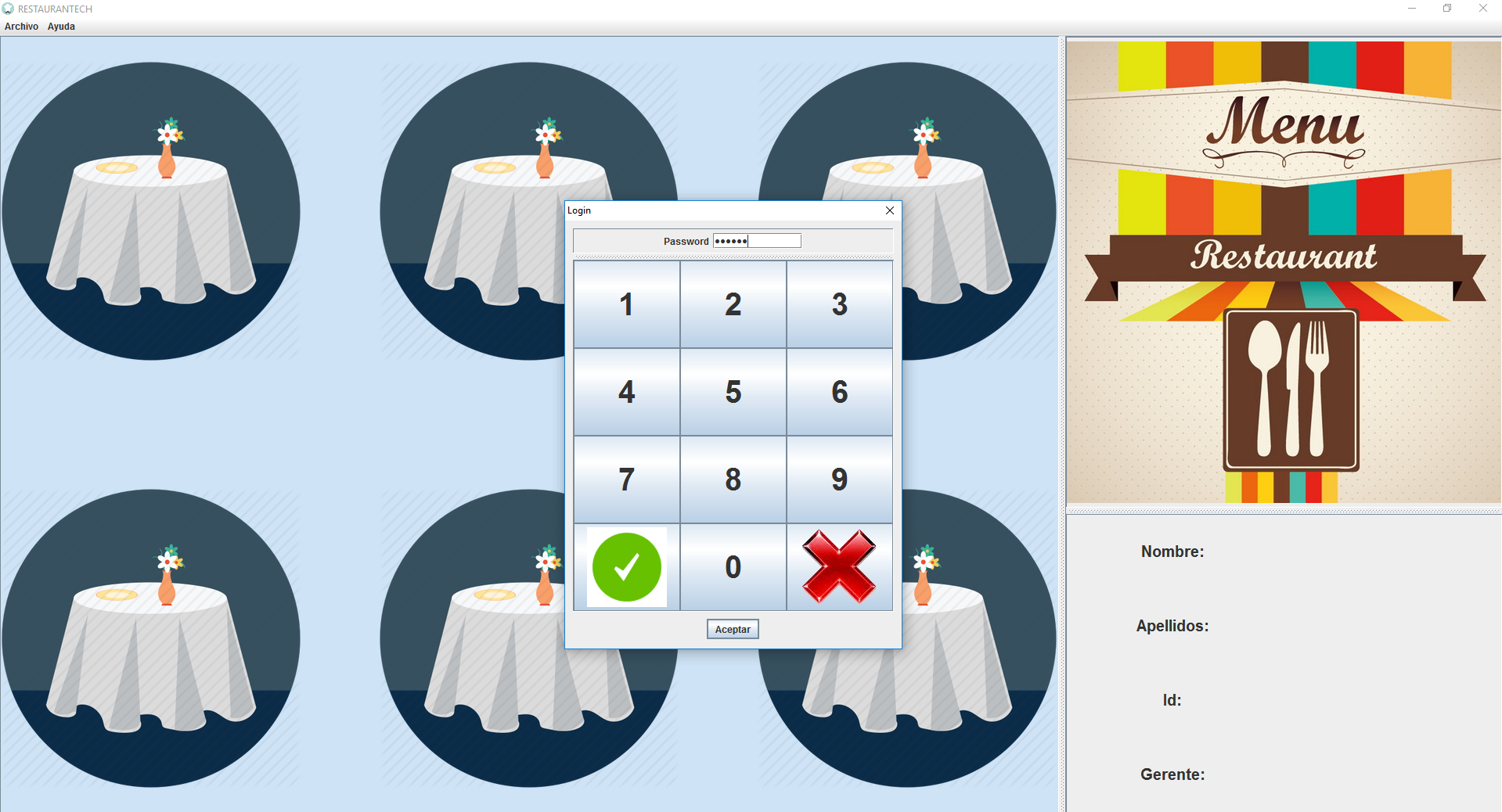


**6.MANUAL DE USUARIO**

Al lanzar nuestro programa podremos observar la pantalla inicial en la cual podremos seleccionar una mesa.

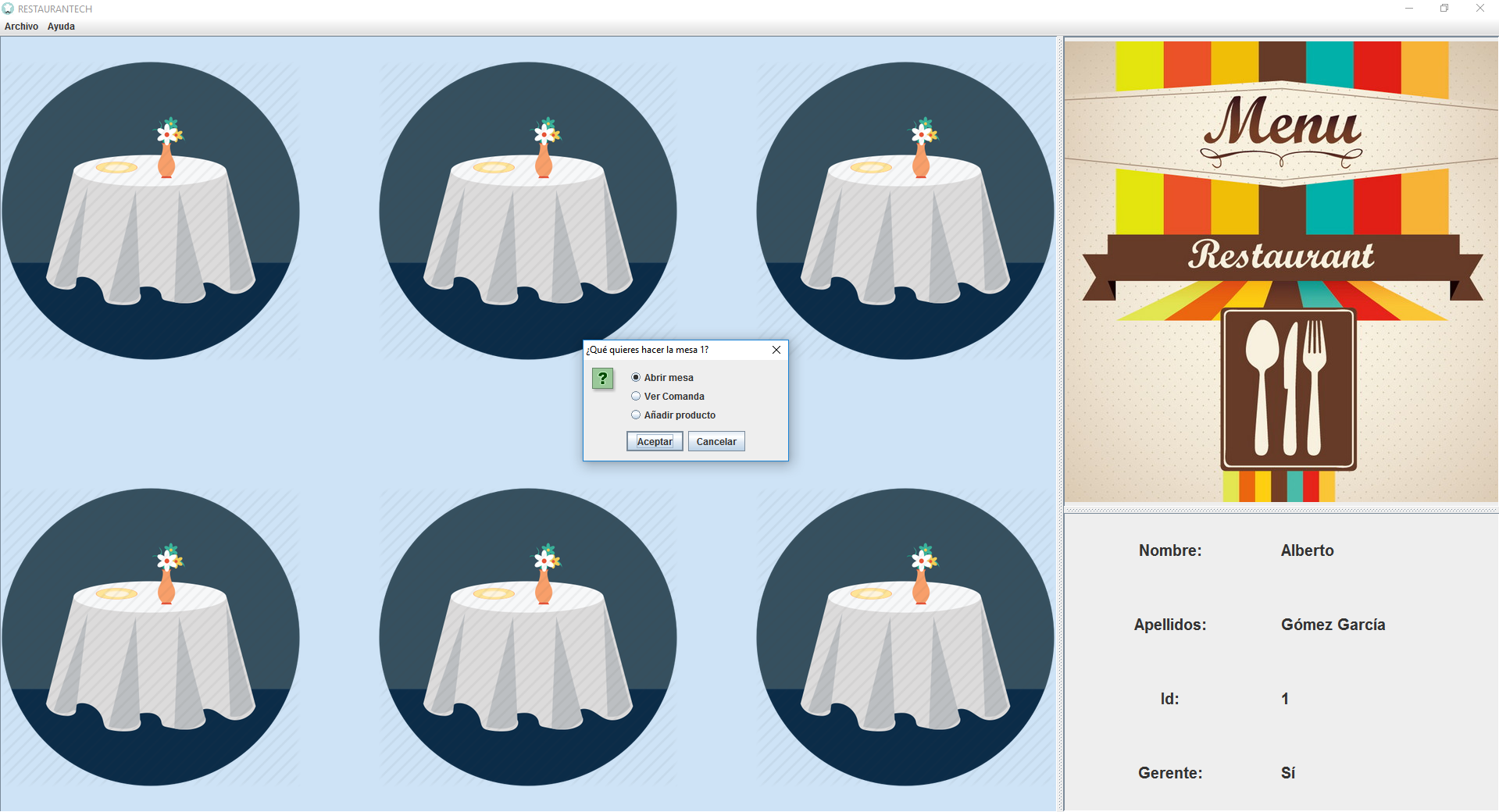


Al seleccionar una mesa, deberemos logearnos como usuarios para ver las opciones que nos brinda la aplicación. El programa obtendrá directamente los datos del camarero que va a servir a esa mesa y los mostrará en el panel inferior-derecho:



Una vez hayamos introducido la contraseña, en caso de ser correcta mostraremos los datos y las opciones para la mesa:

* Abrir mesa: Podremos trabajar en ella, se nos pedirá el número de comensales.
* Ver comanda: Observaremos qué se ha pedido hasta el momento.
* Añadir producto: Añadiremos productos a la comanda.



Opciones para la mesa:

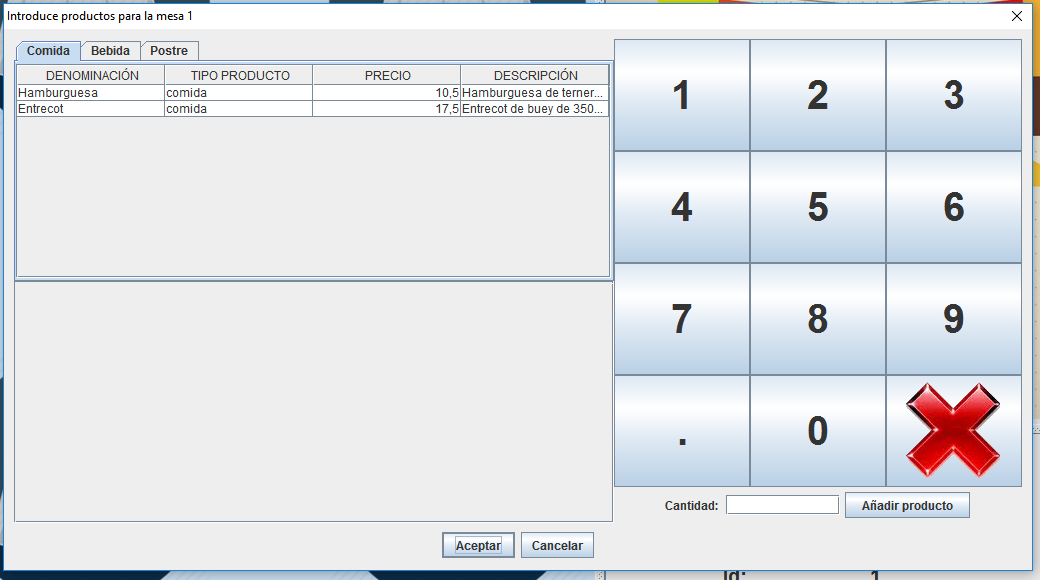
**Abrir mesa**: La aplicación nos pedirá el nº de comensales y al aceptar, se podrá trabajar en ella. Si no la hemos abierto previamente, no podremos trabajar en esa mesa. Al abrirla, se registrará quién la está sirviendo.



**Ver comanda**: Se mostrará los productos que se han pedido así como el precio acumulado hasta el momento, número de la mesa, número de comensales, etc.

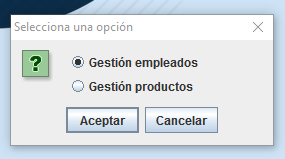


**Añadir producto**: Podremos elegir un producto de la BD a la comanda y se añadirá a la comanda, de modo que el precio acumulado aumentará a razón del producto.

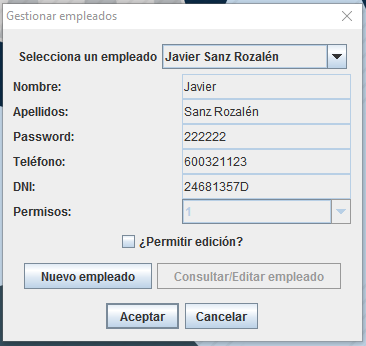


Opciones para el gerente:

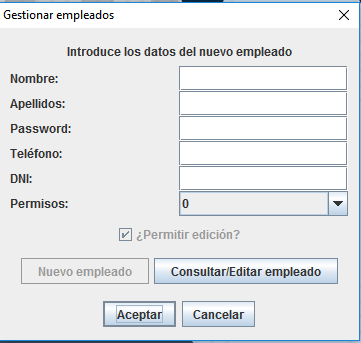
En el JMenuBar podremos hacer click en Archivo->Gerencia para realizar modificaciones/inserciones en la base de datos. Para entrar en esta opción deberemos de volver a introducir nuestra contraseña de gerente, la cual se comprobará en la BD. Las opciones que podremos elegir son las siguientes:



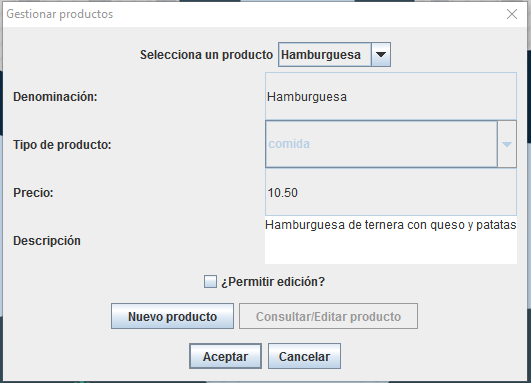
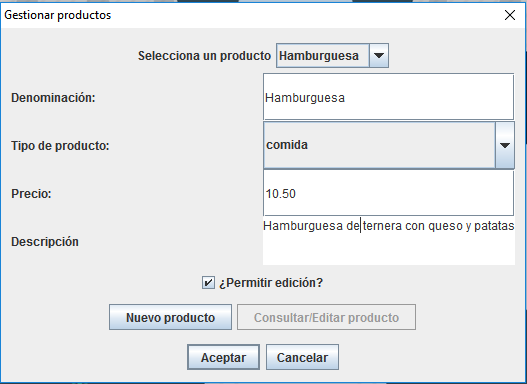
**Gestión de empleado**: En esta opción podremos consultar los datos de nuestros empleados y modificarlos en caso de tener clicado el “¿Permitir edición?”:

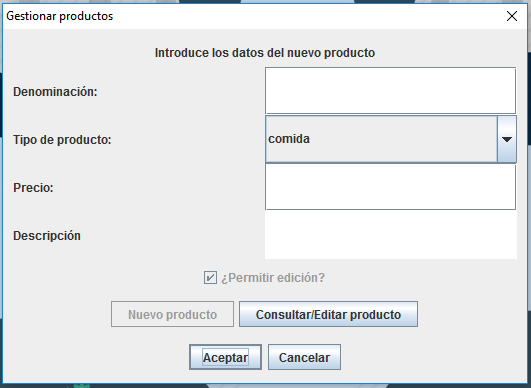
También podremos insertar uno nuevo si presionamos el botón “Nuevo empleado”:



**Gestión de productos**: En esta opción seguimos la misma filosofía que en el apartado de gestión de empleados. Podremos ver los productos y editarlos si tenemos habilitada la opción “¿Permitir edición?”:

Podremos insertar un nuevo producto en la base de datos pulsando el botón “Nuevo producto”:



**7.CONCLUSIÓN**

Tras realizar este trabajo hemos podido afianzar conceptos tanto de Java, como de Hibernate y Java Swing. Nos hemos dado cuenta de que hemos sido capaces de afrontar un gran reto como es combinar lo impartido en distintas asignaturas para realizar una aplicación.

Resulta satisfactorio ver un resultado final, y sobretodo, ver el progreso que hemos tenido como desarrolladores en el transcurso de sólo un año y medio que ha durado este ciclo formativo.

**8.BIBLIOGRAFÍA**

<http://dalila.sip.ucm.es/~manuel/JSW1/Slides/Swing.pdf>

<http://hibernate.org/>

<https://git-scm.com/>

<https://github.com/>

<https://docs.jboss.org/hibernate/orm/3.3/reference/en/html/tutorial.html>