**Universidad Católica Boliviana “San Pablo”**

**Unidad Académica Cochabamba**



**PROYECTO FINAL**

**PRIMERA PRESENTACIÓN**

**Estudiantes:** Paolo Fernandez

Oscar Gonzalez

Adrian Rojas

**Materia:** Sistemas Web

Cochabamba – Bolivia  
Noviembre 2013

**Sistema de atención de un restaurante**

1. **INTRODUCCION**

El presente proyecto consta de la construcción de un sistema de gestión de un restaurant, que se denomina COMECALLADO!, por medio del uso de conceptos de programación web, utilizando HTML, PHP y JavaScript.

1. **ANTECEDENTES**

En nuestra ciudad existen bastantes restaurantes, estos son de todo tipo de comida, algunos especializados en un tipo especifico de cocina, platos o preparaciones culinarias. Cochabamba al ser la capital gastronomica de Bolivia, ostenta este titulo con al menos 300 restaurantes, no todos afiliados a la cámara hotelera, ente encargado del área tanto gastronómica como turística de nuestra ciudad.

Estos restaurants, son conocidos mas que nada en el ámbito local y algunos en el ámbito nacional de acuerdo a un ranking, a una calificación que le asigna la cámara hotelera, asi como también diversos comensales denominados críticos culinarios, que se especializan en dar calificaciones a los restaurantes, de acuerdo a la atención recibida, a la preparación de los platos, bebidas y guarniciones, de la misma manera la higiene de estos lugares influye de sobremanera en estas calificaciones.

Uno que otro restaurant, cuenta con una pagina web que permite realizar reservas online, ver los menus, las especialidades de la casa, asi como promociones y otros.

1. **PROBLEMA**

El problema que se identifica es que son pocos los comensales del interior y del exterior del país, que pueden tener un concepto acerca del restaurant, al mismo tiempo observar las ofertas que brinda este restaurant.

Esto representa un problema de sobremanera, ya que se pierde clientela de una manera enorme al no conocer mas acerca del restaurant, si es que no se estuvo ya ahí o se vio un review de este en alguna pagina de internet.

Este problema representa una perdida de miles de dólares anuales para un restaurant y en muchos casos esto lleva al cierre de estos.

1. **SOLUCIÓN**

Como estudiantes de la materia de Sistemas Web en la Universidad Católica Boliviana nuestro grupo presenta como proyecto final de esta materia, una pagina web para un restaurant, siendo esta un modelo básico, funcional y de interfaz amigable tanto para el administrador como para los clientes, que consta de varias funcionalidades que después de haber realizado una extensa investigación y una evaluación tanto cualitativa como cuantitativa de los requerimientos de varios restaurantes, se toman los puntos mas necesarios para este. Al mismo tiempo con la base del proyecto que nos brinda la docente, expandir un poco estos requerimientos y presentar un sistema completo y funcional sobre todo.

1. **DESCRIPCIÓN**

Un restaurant hoy en día está envuelto en un mercado muy competitivo, en especial en nuestra ciudad de Cochabamba, ya que existen una infinidad de restaurants q ofrecen sus servicios en esta ciudad.

La competitividad obliga cada día más y más a hacer cambios en la forma tradicional de gestionar su información y también la de sus clientes.

Es por este motivo que se necesita dar el siguiente paso tecnológicamente para ofrecer un mejor servicio a los clientes del restaurant, por lo que es necesario contar con un Sistema de Atención, el cual cubra las necesidades primordiales en el manejo de información de un restaurant.

Un Sistema de Atención de un Restaurant debería abarcar áreas de la información como la atención a las mesas en horarios de atención, los pedidos a domicilio de un cliente, el manejo de información de los menús que ofrece el restaurant así como también las promociones y combos que están siendo ofertados.

De esta manera se agiliza el proceso de atención al cliente y se obtiene un registro tangible de las operaciones primordiales del restaurant.

1. **OBJETIVO GENERAL**

Como objetivo general de nuestro sistema tenemos:

* Desarrollar un sistema web para un restaurant, utilizando todos los conocimientos adquiridos en la materia de Sistemas Web, mediante PHP, HTML y JavaScript. Esta aplicación permitirá registrar y administrar la atención realizada en un restaurant, asi mismo las reservas de mesas y la gestión de pedidos y ordenes en el restaurant.

1. **OBJETIVOS ESPECIFICOS**

* Desarrollar módulo de cuentas de usuarios.
* Desarrollar módulo de productos.
* Desarrollar módulo de atención de mesas.
* Desarrollar módulo de pedidos a domicilio.

1. **ALCANCES Y LIMITES**

**8.1. ALCANCES**

* Registro de productos del restaurant.
* Manejo de información de marketing.
* Gestión de menús.
* Manejo de atención de las mesas.
* Pedidos a domicilio.
* Registro de pensionados.
* Manejo de Usuarios.
* No se manejaran mapas en la aplicación.
* No se harán reservas de mesas por parte de un cliente.
* No se manejará la forma de envió en pedidos a domicilio.
* No se manejará el inventario del restaurant.

**8.2. LIMITES**

* Desarrollo en el lenguaje php.
* Base de datos en MySql.
* Manejo de JavaScripts.
* Implementación de CSS para interfaz de usuario.
* Será un sistema web, no una aplicación móvil.
* No se manejaran mapas.

1. **REQUERIMIENTOS**

**9.1. REQUERIMIENTOS FUNCIONALES**

**Como Funcionario:**

* Debe poder registrar, ver y cancelar/restablecer la atención de mesas (registrar el consumo/orden de un cliente).
* Debe poder registrar, ver y cancelar/restablecer pedidos de usuarios (registrar el pedido de un cliente).
* Puede gestionar información de productos:
  + Comida.
  + Bebidas.
  + Guarniciones.
* Puede gestionar información del restaurant:
  + Horas de atención.
  + Combos.
  + Promociones.
* Puede registrar, ver y eliminar menús del día (no modificar).
* Gestionar pensionados.

**Como Administrador:**

* Registrar un funcionario.
* Puede eliminar un funcionario/cliente.
* Puede modificar los datos de un funcionario.
* Todas las funcionalidades de un funcionario.

**Como Cliente:**

* Debe poder registrarse.
* Debe poder iniciar sesión.
* Puede modificar su contraseña.
* Puede modificar sus datos.
* Un cliente debe poder ver información de restaurant.
* Un cliente puede ver los menús del día del restaurant.
* Un cliente puede realizar pedidos por internet (registrar, eliminar y ver pedidos propios).

**9.2. REQUERIMIENTOS NO FUNCIONALES**

* Seguridad:
  + Cuentas y roles de usuario.
  + Manejo de sesiones.
  + Solo podrán eliminar y modificar información los usuarios con el debido permiso.
  + Se eliminaran los datos de forma visible para usuarios ajenos al restaurant pero no se eliminaran de base de datos.
* Implementación de CSS, cookies y javascripts para el desarrollo de interfaz de usuario del sistema.
* Velocidad en consultas a bases de datos.

1. **ANALISIS**

En cuanto al análisis de nuestro sistema, utilizaremos como se menciono antes HTML, PHP y JavaScript para la implementación del sistema, tanto en la parte del cliente, como en la interaccion con base de datos y servidor Apache.

Para la base de datos, utilizaremos MySQL, una base de datos poderosa que nos permite la creación de tablas de manera sencilla, asi mismo la creación de registros para la base de datos se realiza desde la misma aplicación de PHPMyAdmin que corre de manera local en un servidor Apache, gracias a la herramienta MAMP para Macintosh y WAMP para Windows en todas sus versiones.

Cabe resaltar que no importa el navegador que se utilice, ya sea Opera, Google Chrome, Mozilla Firefox o incluso Internet Explorer, el sistema funcionara de igual manera y con todas las funcionalidades al 100% debido a que HTML, PHP y JavaScript son universales para cualquier navegador, incluso un navegador de un dispositivo móvil.

A continuación presentamos el diccionario de datos de la Base de Datos, diagrama de base de datos y el diagrama de casos de uso:

**combo**

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_combo | int(8) | No | Codigo del Combo |
| descripcion | varchar(50) | No | Descripcion del Combo |
| imagen | int(50) | No | Imagen de un Combo |
| fecha\_inicio | date | No | Fecha de inicio de vigencia del Combo |
| fecha\_fin | date | No | Fecha fin de vigencia del combo |
| estado | tinyint(1) | No | Estado activo o inactivo del combo |

**combo\_producto**

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_combo\_producto | bigint(20) | No | Codigo de la tabla |
| cod\_combo | int(8) | No | Codigo de un combo |
| cod\_producto | varchar(8) | No | Codigo de un producto |
| Cantidad | int(1) | No | Cantidad de combos y productos disponibles |

**menu**

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_menu | int(8) | No | Codigo del menu |
| fecha\_inicio | date | No | Fecha de inicio de vigencia del menu |
| fecha\_fin | date | Sí | Fecha de fin de vigencia del menu |
| estado | tinyint(1) | No | Estado del menu activo o inactivo |

# menu\_dia

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_menu\_dia | int(8) | No | Codigo del Menu del dia |
| dia | varchar(12) | No | Dia del menu |
| fecha\_inicio | date | No | Fecha inicio de vigencia del menu del dia |
| fecha\_fin | date | Sí | Fecha fin de vigencia del menu del dia |
| estado | tinyint(1) | No | Estado del menu del dia |

# orden

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_orden | bigint(20) | No | Codigo de la orden |
| num\_mesa | int(2) | No | Numero de mesa de la orden |
| nombre\_cliente | varchar(40) | No | Nombre del cliente de la orden |
| fecha | datetime | No | Fecha de la orden |
| estado | tinyint(1) | No | Estado de la Orden Activa o Inactiva |

# orden\_producto

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_orden\_producto | bigint(20) | No | Codigo de la tabla |
| cod\_orden | bigint(20) | No | Codigo de una orden |
| cod\_producto | varchar(8) | No | Codigo de un producto |
| cantidad | int(2) | No | Cantidad de Ordenes y Productos Asociados |

# pedido

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_pedido | bigint(20) | No | Codigo del pedido |
| nombre | varchar(40) | No | Nombre del cliente al cual pertenece el pedido |
| nit | bigint(12) | No | Nit del cliente |
| direccion | varchar(50) | No | Direccion del cliente para el pedido |
| telefono | bigint(12) | No | Telefono del cliente para el pedido |
| fecha | datetime | No | Fecha del pedido |
| estado | tinyint(1) | No | Estado del pedido Activo o Inactivo |
| entregado | tinyint(1) | No | Informa si se entrego el pedido o no |
| usuario | varchar(12) | No | Codigo del Usuario al que esta asociado el pedido |

# pedido\_producto

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_penido\_producto | bigint(20) | No | Codigo de la tabla |
| cod\_pedido | bigint(20) | No | Codigo de un Pedido |
| cod\_producto | varchar(8) | No | Codigo de un Producto asociado a un pedido |
| cantidad | int(2) | No | Cantidad de Productos Asociados al pedido |

# pensionados

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_pensionado | int(8) | No | Codigo de un pensionado |
| ci | int(12) | No | Carnet de Identidad de un pensionado |
| nombre | varchar(40) | No | Nombre del Pensionado |
| direccion | varchar(60) | No | Direccion del Pensionado |
| telefono | bigint(12) | No | Telefono del pensionado |
| estado | tinyint(1) | No | Estado del pensionado Activo o Inactivo |

# producto

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_producto | varchar(8) | No | Codigo de un producto |
| nombre | varchar(30) | No | Nombre del Producto |
| tipo | int(1) | No | Tipo del Producto Comida, bebida o guarnicion |
| precio | float | No | Precio del Producto |
| estado | tinyint(1) | No | Estado del producto Activo o Inactivo |

# producto\_menu

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_producto\_menu | int(8) | No | Codigo de la Tabla |
| cod\_menu | int(8) | No | Codigo del menu al que se asocia un producto |
| cod\_producto | varchar(8) | No | Codigo del producto que se asocia a un menu |

# producto\_menu\_dia

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_producto\_menu\_dia | int(8) | No | Codigo de la tabla |
| cod\_menu\_dia | int(8) | No | Codigo del menu del dia al que esta asociado un producto |
| cod\_producto | varchar(8) | No | Codigo del producto asociado a un menu del dia |

# promocion

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_promocion | int(8) | No | Codigo e una promocion |
| descripcion | varchar(50) | No | Descripcion de una promocion |
| imagen | varchar(50) | No | Imagen de una promocion |
| fecha\_inicio | date | No | Fecha de inicio de vigencia de una promocion |
| fecha\_fin | date | No | Fecha de fin de vigencia de una promocion |
| estado | tinyint(1) | No | Estado de una promocion Activa o Inactiva |

# promocion\_producto

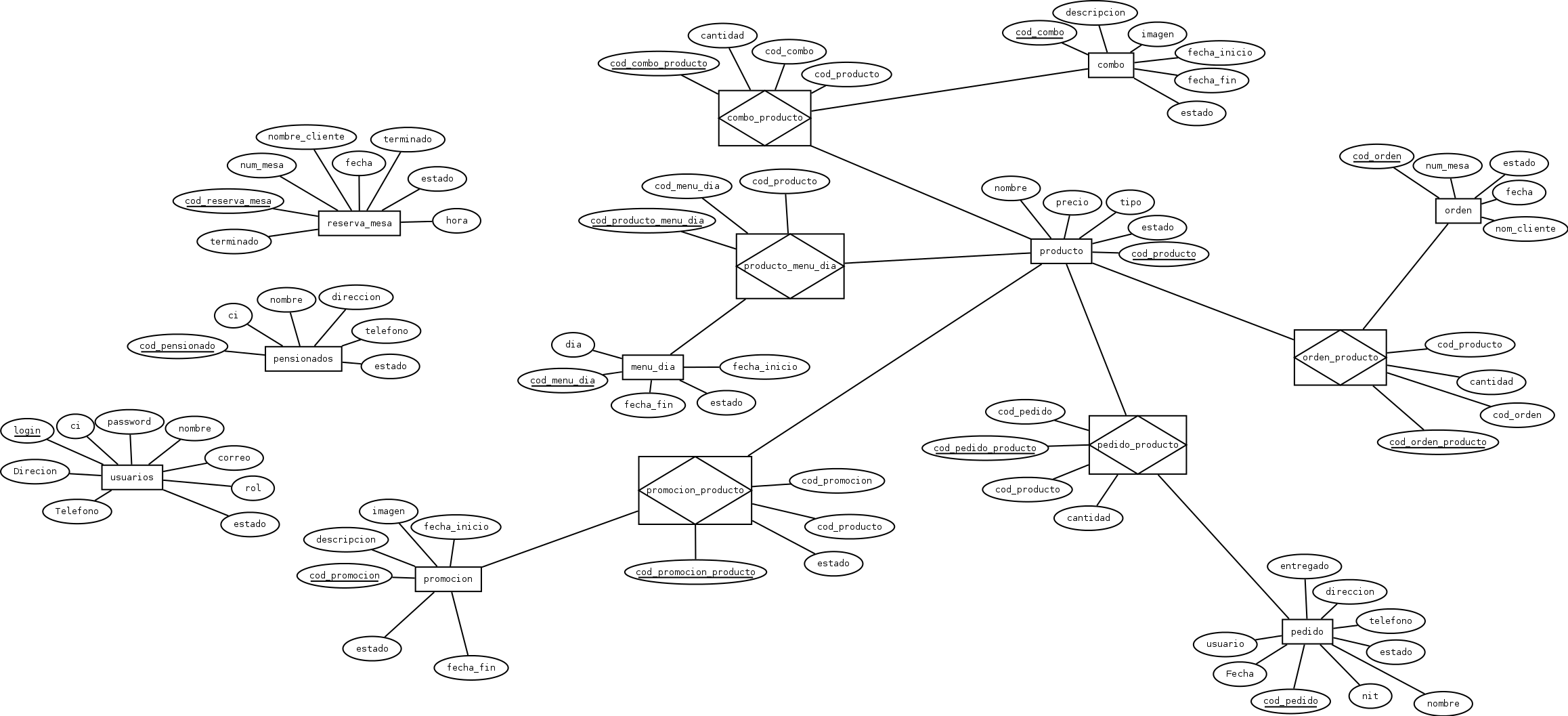
| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_promocion\_producto | bigint(20) | No | Codigo de la tabla |
| cod\_promocion | int(8) | No | Codigo de una promocion a la que esta asociado un producto |
| cod\_producto | varchar(8) | No | Codigo de un producto asociado a una promocion |
| cantidad | int(1) | No | Cantidad de productos asociados a una promocion |

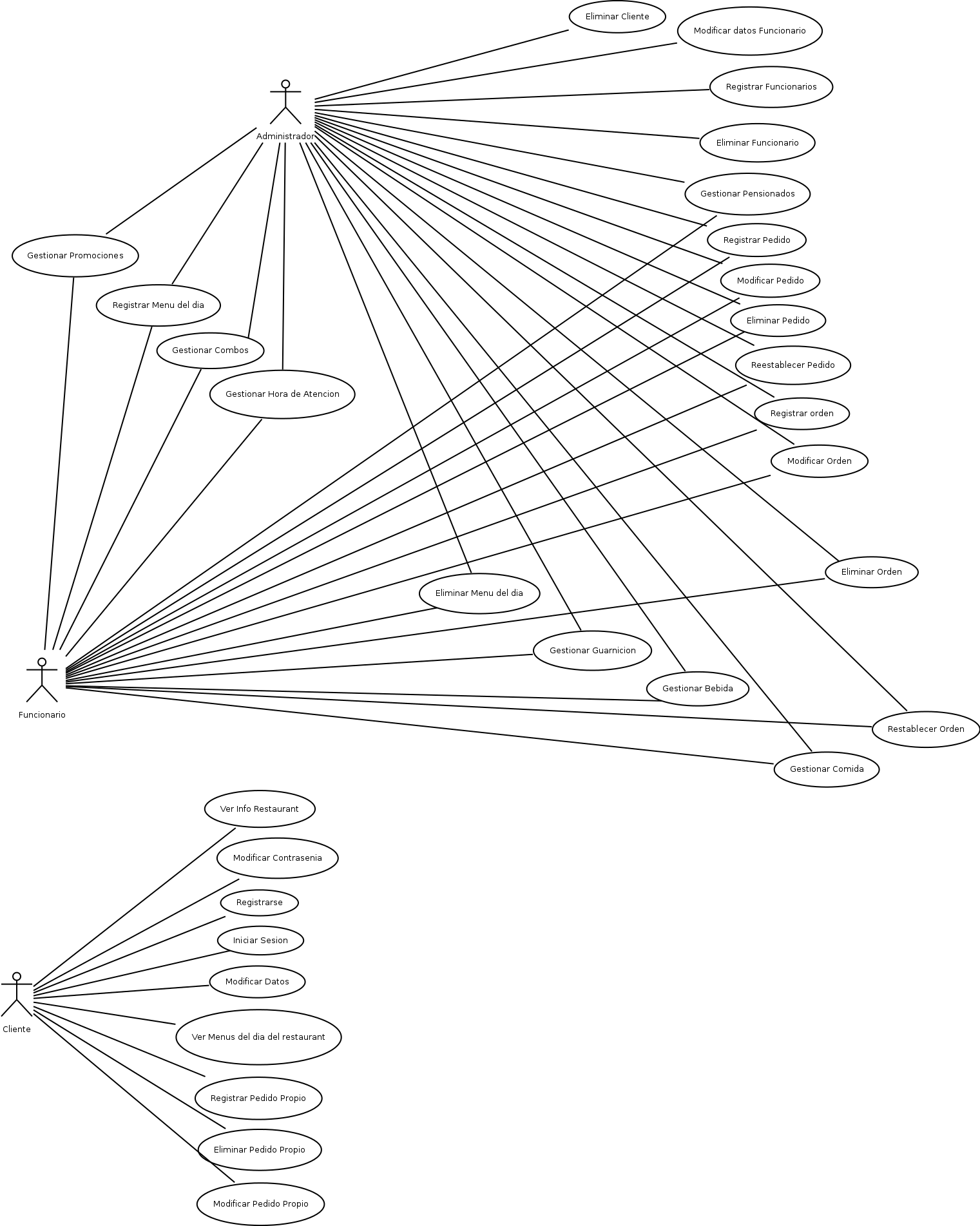
# reserva\_mesa

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| cod\_reserva\_mesa | bigint(12) | No | Codigo de reserva de una mesa |
| num\_mesa | int(2) | No | Numero de mesa reservada |
| nombre\_cliente | varchar(40) | No | Nombre del cliente para la reserva |
| fecha | date | No | Fecha de la reserva |
| hora | time | No | Hora de la reserva |
| estado | tinyint(1) | No | Estado de la reserva Activa o Inactiva |
| terminada | tinyint(1) | No | Indicador de finalizacion de la reserva |

# usuarios

| **Columna** | **Tipo** | **Nulo** | **Comentarios** |
| --- | --- | --- | --- |
| login | varchar(12) | No | Login del Usuario |
| password | varchar(100) | No | Password del usuario |
| ci | bigint(12) | No | Carnet de Identidad del usuario |
| nombre | varchar(40) | No | Nombre del usuario |
| correo | varchar(30) | No | Correo del Usuario |
| rol | int(1) | No | Rol del usuario |
| direccion | varchar(50) | Sí | Direccion del usuario |
| telefono | bigint(12) | Sí | Telefono del usuario |
| estado | tinyint(1) | No | Estado del usuario Activo o Inactivo |

****

****

1. **CONCLUSIONES**

PHP y Ruby on Rails no son directamente comparables. PHP es un lenguaje, Ruby (a secas) también es un lenguaje, y Ruby on Rails es un framework de desarrollo de aplicaciones web.

Como lenguajes, PHP y Ruby distan mucho entre sí. La mayor ventaja de PHP es que es muy conocido en el ámbito laboral. La cualidad más destacable de Ruby es que es un lenguaje mucho más limpio, conciso y meta-programable. Aunque ambos son lenguajes de propósito general, existen pocos proyectos donde PHP se use fuera del ámbito web, y en cambio Ruby se utiliza para otros muchos casos, como la programación de aplicaciones de escritorio.

En nuestro caso no hicimos uso de un framework, por lo que el desarrollo nos llevo mayor tiempo. Debido a las consultas a la base de datos mayormente y al hacer los CRUD’s para las tablas.

El sistema podría ampliarse en un futuro para que se pueda gestionar de mejor manera el uso y la conexión a base de datos.