



**UNIVERSIDAD POLITÉCNICA SALESIANA  
SEDE GUAYAQUIL**

**CARRERA: INGENIERÍA EN SISTEMAS**

**Tesis previa a la obtención del título de: INGENIERO DE SISTEMAS**

**TEMA:**

**“DISEÑO, VALIDACIÓN E IMPLEMENTACIÓN DE SOFTWARE PARA LA GESTIÓN DE SERVICIOS DE LA HOSTERÍA COMUNITARIA DE LA AGRUPACIÓN MUJERES PROGRESISTAS ÁFRICA MIA UBICADA EN LA COOP. INDEPENDENCIA II, DEL BARRIO NIGERIA, SECTOR ISLA TRINITARIA”**

**AUTORES:**

**MARIANA LISSETTE PINTADO CUJI  
RICARDO FRANCISCO MESÍAS OCHOA**

**DIRECTOR:**

**ING. MÁXIMO GIOVANI TANDAZO ESPINOZA**

**Guayaquil, abril de 2015**

**DECLARATORIA DE RESPONSABILIDAD Y AUTORIZACIÓN DE  
USO DEL TRABAJO DE GRADO**

Nosotros Mesías Ochoa Ricardo Francisco y Pintado Cuji Mariana Lisette autorizamos a la Universidad Politécnica Salesiana la publicación total o parcial de este trabajo de grado y su reproducción sin fines de lucro.

Además declaramos que los conceptos y análisis desarrollados y las conclusiones del presente trabajo son de exclusiva responsabilidad de los autores.

-----  
**MESÍAS OCHOA RICARDO FRANCISCO**  
**C.C. 0925064982**

-----  
**PINTADO CUJI MARIANA LISSETTE**  
**C.C. 0927064451**

## **DEDICATORIA**

Dedico esta tesis a mis padres Mariana Cuji e Iván Pintado y a mi hermanita Karen Pintado quienes me han ayudado a ser la persona que soy.

**Mariana Lissette Pintado Cuji.**

## **DEDICATORIA**

Dedico esta tesis a mis padres y hermanos quienes confiaron en mí, y a mi hija Paulette quien ha sido mi más grande inspiración.

**Ricardo Francisco Mesías Ochoa.**

## **AGRADECIMIENTO**

Agradezco a la Universidad Politécnica Salesiana por ofrecerme una educación de calidad, a mis profesores los cuales me brindaron los conocimientos necesarios para ser un profesional.

A mis padres, a mis hermanos por apoyarme siempre y confiar en mis capacidades.

Agradezco a mis compañeros y amigos por enseñarme tantas cosas y compartir conmigo momentos inolvidables.

**Ricardo Francisco Mesías Ochoa.**

## **AGRADECIMIENTO**

Agradezco a la Universidad Politécnica Salesiana por brindarme una educación de calidad y una formación con excelencia humana y académica

Al economista Andrés Bayolo quien me dió la oportunidad de culminar mis estudios confiando en mis capacidades y compromiso con lo institución.

A la Ing. Ángela Flores quien me ha brindado su apoyo desde el día que la conocí.

Al master Roberto Rangel Donoso director de la maestría de desarrollo local y gran amigo quien siempre me ha enseñado desde que lo conocí en Pastoral

A todos mis profesores que día a día durante toda mi etapa estudiantil me brindaron muchos conocimientos y experiencias maravillosas, en especial a la Ec. Liliana Aguirre por sus consejos y enseñanzas y sobre todo por su amistad.

A mi tutor Ing. Máximo Tandazo quien me ha asesorado en el desarrollo de la tesis.

A mis padres y hermana quienes siempre me han inspirado para terminar mi carrera.

Y principalmente a Dios quien lo ha hecho todo posible.

**Mariana Lissette Pintado Cuji.**

## ÍNDICE

|  |     |
|--|-----|
| RESUMEN .....  | xiv |
| ABSTRACT .....   | xv  |
| CAPÍTULO I.....  | 3   |
| DISEÑO DE LA INVESTIGACIÓN.....                            | 3   |
| 1. Planteamiento del Problema .....                        | 3   |
| 1.1. Antecedentes .....                                    | 3   |
| 1.2. Enunciado del problema. ....                          | 4   |
| 1.3. Formulación del problema.....                         | 5   |
| 1.4. Objetivos.....  | 5   |
| 1.4.1. Objetivo general.....                               | 5   |
| 1.4.2. Objetivos específicos. ....                         | 5   |
| 1.5. Justificación. ....                                   | 5   |
| 1.5.1. Importancia. ....                                   | 6   |
| 1.5.2. Necesidad .....                                     | 6   |
| 1.5.3. Beneficios que aporta.....                          | 7   |
| 1.5.4. Beneficiarios. ....                                 | 7   |
| CAPITULO 2 .....   | 8   |
| MARCO REFERENCIAL DE LA INVESTIGACIÓN .....                | 8   |
| 2.1. Marco teórico.....                                    | 8   |
| 2.1.1. Fundamentos de la web.....                          | 8   |
| 2.1.2. El protocolo HTTP.....                              | 8   |
| 2.1.2.1. Peticiones en el protocolo HTTP: GET y POST ..... | 10  |
| 2.1.2.2. Línea de petición .....                           | 10  |
| 2.1.2.3. Cabecera de la petición.....                      | 11  |
| 2.1.2.4. Parámetros de la petición.....                    | 11  |
| 2.1.2.5. Respuestas en el protocolo HTTP .....             | 14  |
| 2.1.3. El lenguaje HTML .....                              | 14  |
| 2.1.3.1. ¿Qué es html? .....                               | 14  |
| 2.1.3.2. HTML5 .....                                       | 15  |
| 2.1.4. ¿Qué es un servidor FTP (FTP Servers)?.....         | 15  |
| 2.1.5. ¿Qué es XAMPP? .....                                | 16  |

|   |    |
|---|----|
| 2.1.6. Definición de php .....                                      | 16 |
| 2.1.6.1. Variables.....   | 17 |
| 2.1.7. ¿Qué es una Base de Datos?.....                              | 18 |
| 2.1.7.1. Importancia de las Bases de Datos .....                    | 19 |
| 2.1.7.2. Interacción con PHP .....                                  | 19 |
| 2.1.7.3. Ventajas de MySQL. ....                                    | 20 |
| 2.1.7.4. Conectarse con PHP a MySQL. ....                           | 20 |
| 2.1.8. Sublime text 2 .....   | 21 |
| 2.1.9. AJAX .....   | 21 |
| 2.1.10. Hostería .....  | 22 |
| 2.1.10.1. Servicios que ofrecerá la Hostería.....                   | 22 |
| 2.1.11. Sistema de información.....                                 | 23 |
| 2.1.12. ¿Qué es Intranet?.....                                      | 23 |
| 2.2. Matriz Causa y efecto .....                                    | 25 |
| 2.3. Marco Legal .....  | 26 |
| 2.4. Formulación de hipótesis .....                                 | 26 |
| 2.5. Operacionalización de variables, e indicadores.....            | 27 |
| 2.5.1. Variables Independientes .....                               | 27 |
| 2.5.2. Variables Dependientes.....                                  | 27 |
| CAPITULO 3 .....  | 28 |
| MARCO METODOLÓGICO DE LA INVESTIGACIÓN .....                        | 28 |
| 3.1. Tipo de estudio.....   | 28 |
| 3.1.1. Investigación exploratoria.....                              | 28 |
| 3.1.2. Investigación descriptiva.....                               | 29 |
| 3.2. Método de investigación.....                                   | 29 |
| 3.2.1. Método Experimental.....                                     | 30 |
| 3.2.2. Método de Análisis .....                                     | 30 |
| 3.3. Fuentes y Técnicas para la recolección de la Información ..... | 30 |
| CAPITULO 4 .....  | 32 |
| ANÁLISIS Y DISEÑO DE LA PROPUESTA.....                              | 32 |
| 4.1. Resultado de la Investigación .....                            | 32 |
| 4.2. Arquitectura de Aplicación.....                                | 32 |
| 4.3. Diagrama de escenarios .....                                   | 33 |
| 4.4. Diagrama de Estados .....                                      | 34 |

|  |     |
|--|-----|
| 4.5. Toma de Requerimientos .....                | 35  |
| 4.6. Casos de Uso.....                           | 36  |
| 4.6.1. Caso de uso de Restaurante.....           | 36  |
| 4.6.2. Caso de uso de Hospedaje.....             | 39  |
| 4.6.3. Caso de uso de Ventas.....                | 41  |
| 4.6.4. Caso de uso de Clientes.....              | 42  |
| 4.6.5. Caso de uso de Eventos.....               | 43  |
| 4.6.6. Caso de uso de Personal.....              | 44  |
| 4.7. Requisitos de Hardware y Software.....      | 44  |
| 4.8. Diagrama Entidad – Relación .....           | 43  |
| CAPITULO 5 .....                                 | 44  |
| DESARROLLO E IMPLEMENTACION DE LA PROPUESTA..... | 44  |
| 5.1. Script de creación de base de datos .....   | 44  |
| 5.3. Código de la Aplicación .....               | 78  |
| 5.1. Pruebas y Resultados .....                  | 104 |
| CAPITULO 6.....                                  | 117 |
| CONCLUSIONES Y RECOMENDACIONES .....             | 117 |
| 6.1. CONCLUSIONES.....                           | 117 |
| 6.2. RECOMENDACIONES.....                        | 118 |
| REFERENCIAS.....                                 | 119 |
| ANEXOS .....                                     | 115 |

## ÍNDICE DE TABLAS

|   |    |
|---|----|
| Tabla 1 Ingresando Productos o servicios.....           | 37 |
| Tabla 2 Visualizando productos y servicios .....        | 37 |
| Tabla 3 Creando nueva mesa .....                        | 37 |
| Tabla 4 Visualizando mesas.....                         | 38 |
| Tabla 5 Creando nueva habitación.....                   | 39 |
| Tabla 6 Visualizando Habitaciones.....                  | 40 |
| Tabla 7 Registro de Huéspedes .....                     | 40 |
| Tabla 8 Visualizando Huéspedes registrados.....         | 40 |
| Tabla 9 Creando Facturas.....                           | 41 |
| Tabla 10 Visualizando Facturas .....                    | 41 |
| Tabla 11 Creando nuevo cliente.....                     | 42 |
| Tabla 12 Visualizando Clientes .....                    | 42 |
| Tabla 13 Creando eventos.....                           | 43 |
| Tabla 14 Visualizando eventos .....                     | 43 |
| Tabla 15 Visualizando y creando meseros o cajeros ..... | 44 |

## ÍNDICE DE GRÁFICOS

|  |     |
|--|-----|
| Figura 1: Arquitectura de Aplicación.....                  | 32  |
| Figura 2: Diagrama de Escenarios de la Aplicación Web..... | 33  |
| Figura 3: Diagrama de Estados .....                        | 34  |
| Figura 4: Diagrama de caso de uso de Restaurante.....      | 36  |
| Figura 5: Diagrama de Casos de Uso de Hospedaje .....      | 39  |
| Figura 6: Diagrama de Casos de Uso de Ventas .....         | 41  |
| Figura 8: Diagrama de casos de uso de Eventos .....        | 43  |
| Figura 9: Diagrama de Casos de Uso de Personal .....       | 44  |
| Figura 10: Diagrama Entidad Relación.....                  | 43  |
| Figura 11: Pantalla de Bienvenida .....                    | 104 |
| Figura 12: Pantalla de Ingreso a usuarios .....            | 105 |
| Figura 13: Disponibilidad de habitaciones.....             | 106 |
| Figura 14: Listado de habitaciones .....                   | 106 |
| Figura 15: Pantalla de nueva Reservación .....             | 107 |
| Figura 16: Listado de Reservaciones .....                  | 107 |
| Figura 17: Reservar habitación .....                       | 108 |
| Figura 18: Huéspedes Registrados .....                     | 108 |
| Figura 19: Pantalla de Ingreso de productos .....          | 109 |
| Figura 20: Pantalla de Listado de productos .....          | 109 |
| Figura 21: Creación de mesas .....                         | 110 |
| Figura 22: Disponibilidad de mesas .....                   | 110 |
| Figura 23: Vista de estado de las mesas.....               | 111 |
| Figura 24: Pantalla de creación de eventos .....           | 112 |
| Figura 25: Pantalla de listado de eventos.....             | 112 |
| Figura 26: Pantalla de Nueva factura .....                 | 113 |
| Figura 27: Listado de Facturas .....                       | 113 |
| Figura 28: Pantalla de Creación de nuevo servicio.....     | 114 |
| Figura 29: Pantalla de listado de servicios.....           | 114 |
| Figura 30: Pantalla de creación de nuevo cliente.....      | 115 |
| Figura 31: Listado de los clientes.....                    | 115 |

|  |     |
|--|-----|
| Figura 32: Pantalla de creación de empleados ..... | 116 |
| Figura 33: Listado de empleados .....              | 116 |
| Figura 34: Listado de empleados .....              | 116 |

## INDICE DE ANEXOS

|  |     |
|--|-----|
| Anexo 1: La encuesta: Grupo afroecuatoriano y público en general ..... | 115 |
| Anexo 2: Guía diseñada: Grupo focal.....                               | 116 |
| Anexo 3: La entrevista: Experto en hostería .....                      | 117 |
| Anexo 4: Diccionario de datos.....                                     | 118 |

## RESUMEN

En este trabajo de grado se creará un sistema para la gestión de servicios de la Hostería Comunitaria África Mía, luego de conocer las necesidades y requerimientos que existe dentro de la hostería y haber analizado el diseño más adecuado para así automatizar los procesos de negocio y logren mantener un control. El beneficio tiene un impacto directo en las mujeres asociadas y cabeza de hogar y de economía prioritaria, con niveles significativos de vulnerabilidad, por situaciones de violencia de género, así como sus hijos menores de edad. Beneficio que se extiende a los adultos mayores familiares. Además el impacto indirecto sobre la comunidad del entorno de los mismos huéspedes.

El sistema de gestión de servicios es muy fácil de utilizar, con él se mantendrá un mejor control en el negocio además de la automatización de los procesos evitando así realizarlos de forma manual. El proceso para el diseño y desarrollo de esta aplicación web fue conocer las necesidades informáticas que actualmente posee la hostería por medio de entrevistas y reuniones en la comunidad afroecuatoriana de la Isla trinitaria donde se encuentra situada la hostería África Mía.

Para el análisis se conoció y se utilizó el estándar UML, en el cual se obtuvo las necesidades y requerimientos de la hostería.

En cuanto al diseño se presenta el diagrama de Entidad Relación, en el que se encuentra la estructura que tiene la Base de Datos en cada módulo del Sistema de gestión de servicios.

Para el desarrollo del sistema, se trabajó con XAMP Server. En el lenguaje PHP, con la base de datos de MySQL Server.

Se presenta el Diccionario de datos con información de cada tabla utilizada en el software.

Además se presentan las pruebas y los resultados de la aplicación web.

## **ABSTRACT**

In this paper grade a system for service management of the Community Hosteria Africa Mia, after meeting the needs and requirements that exist within the hostel and having analyzed the most suitable design to automate business processes will be created and achieve maintain control and Profit has a direct impact on associated and headed households and women priority economy, with significant levels of vulnerability, for situations of domestic violence and their minor children. Benefit extending the elderly relatives. Furthermore, the indirect impact on the surrounding community of the same guests.

The service management system is easy to use and it will keep better control in addition to automating business processes avoiding manual execution. The process for the design and development of this Web application was to meet the computing needs that currently owns the inn through interviews and meetings in the Afro-Ecuadorian community of Trinitarian Island where is located the inn Africa Mia.

For the analysis was met and the UML standard was used, in which the needs and requirements of the inn was obtained.

In terms of design Entity Relationship diagram is presented, which is the structure that has the database in each module management system services.

To develop the system, worked with XAMP Server. In the PHP language with MySQL Database Server.

The Data Dictionary is presented with information for each table used in the software.

Further tests and the results of the Web application are presented.

## INTRODUCCIÓN

En la actualidad se utilizan los sistemas de información a partir de la necesidad de las diferentes tareas que se realizan en las organizaciones y negocios ya que gracias a ellos los negocios manejan la información de manera confiable, cómoda y facilitando los procesos y de esa manera permiten obtener información rápida de los registros diarios que se necesitan en su debido momento. Con los sistemas informáticos esta tarea resulta más fácil de controlar.

La Hostería escogida para el desarrollo e implementación del software es África Mía por iniciativa del director de la maestría de Desarrollo Local Msc. Roberto Rangel Donoso y la Asociación afroecuatoriana “Mujeres Progresistas” quienes propusieron el tema de tesis demostrando disponibilidad desde el inicio para el desarrollo del proyecto.

En el Primer Capítulo se detalla el motivo de la investigación en la Hostería África Mía y la Asociación de Mujeres Progresistas; el planteamiento del problema, la formulación del problema, delimitación del problema, los objetivos que ayudarán a implementar el software, la justificación de la investigación, la importancia, necesidad, beneficios que aporta y los beneficiarios.

En el Segundo Capítulo se encuentra el marco teórico, conceptualizaciones de las fundamentaciones y leyes que respaldan la investigación, marco conceptual, base legal, la formulación de hipótesis, operacionalización de las variables e indicadores.

El tercer capítulo da a conocer el marco metodológico donde se detallan los recursos a utilizar, contiene los datos informativos, requerimientos funcionales, requerimientos no funcionales, definición de roles en los módulos.

El cuarto capítulo puntualiza el diseño de la arquitectura del sistema, los módulos con los que cuenta el sistema, diagrama de clases del sistema, modelo lógico de la base de datos y el análisis e interpretación de resultados, además de las pruebas, resultados

El quinto capítulo comprende las conclusiones, recomendaciones, bibliografía y anexos.

# CAPÍTULO I

## DISEÑO DE LA INVESTIGACIÓN

### 1. Planteamiento del Problema

#### 1.1. Antecedentes

Guayaquil es la ciudad más densamente poblada del Ecuador y posiblemente la de mayor inequidad. Según la encuesta de condiciones de vida, la población aproximadamente es de 2 millones 291 mil 158, de los cuales el 7.2 % son afrodescendientes. (INEC, 2011). El 44% de población afrodescendiente vive en áreas no consolidadas, lo que, de acuerdo al Municipio de Guayaquil, quiere decir, con alta condición de pobreza. Una de las áreas no consolidadas es la Isla Trinitaria, en la cual vive el 21% de población afro guayaquileña en los sectores Nigeria, Cenepa, entre otros. (DASE, 2006). Se desenvuelven con poca disponibilidad de obras de infraestructura vial, baja cobertura de servicios básicos, escaso equipamiento social y urbano y socioeconómicamente corresponde al quintil 1, o sea, tienen ingresos económicos cerca de un salario mínimo vital, en otras palabras, en los sectores en mención, los hogares en que hay jefatura femenina, en su mayoría viven en pobreza.

Entre los sectores más populares de la Isla Trinitaria de la ciudad de Guayaquil se encuentra un sector conocido como barrio Nigeria, que está rodeada por el Estero Mogollón y Estero del Muerto, debido a esto su acceso se lo puede realizar tanto por vía fluvial como por vía terrestre. Conforme se iba poblando el lugar se comenzaron a palpar las necesidades, entre ellas estaba el transporte, el cual se lo realiza a través de tricimotos por vía terrestre y canoas por vía marítima.

Como respuesta a las necesidades la comunidad en el año 2009 creó la Asociación de Mujeres Progresistas conformada por mujeres afro-ecuatorianas y sus familias, migrantes de la zona Norte de la Costa de Ecuador, jefas de hogar y en violencia intrafamiliar por la dependencia económica a sus parejas.

En el año 2012 para mejorar la situación económica de sus integrantes crearon el “Centro Empresarial Mujeres Emprendedoras” teniendo como proyecto la Hostería Comunitaria “África Mía”.

Este proyecto esta visualizado para poder cambiar la imagen negativa que tienen de los barrios de la Isla Trinitaria, dando un nuevo modelo de desarrollo turístico urbano donde sean beneficiados todos los habitantes de la comunidad. Además, se pretende promover y desarrollar actividades turísticas que pongan en valor las riquezas culturales y naturales que el sector posee y de esta forma promocionarlo para que sea conocido a nivel nacional e internacional.

Su misión es prestar servicio de alojamiento, alimentación, organización de eventos, entre otros, promoviendo la participación activa de las mujeres de la comunidad al servicio del cliente teniendo como garantía su satisfacción.

Su visión es lograr en 5 años ser la primera empresa modelo de desarrollo comunitario y trabajo asociativo en la provincia del Guayas.

Como objetivos principales están:

- Fortalecer aspectos culturales de los afroecuatorianos.
- Generar empleos sostenidos para mujeres y jóvenes del sector.
- Brindar servicios para visitantes tanto nacionales como extranjeros.

## **1.2. Enunciado del problema.**

El desconocimiento de la automatización de los procesos de gestión limita de la administración de servicios de la hostería comunitaria de la agrupación mujeres progresistas África Mía ubicada en la coop. Independencia II, del Barrio Nigeria, sector Isla Trinitaria.

### **1.3. Formulación del problema.**

¿Cuáles son las consecuencias de llevar los procesos de forma manual y de no contar con un software de gestión de servicios en la hostería comunitaria África Mía ubicada en la coop. Independencia II, del Barrio Nigeria, sector Isla Trinitaria?

### **1.4. Objetivos.**

#### **1.4.1. Objetivo general.**

Desarrollar un software para la gestión de servicios de la hostería comunitaria de la agrupación mujeres progresistas de la coop. Independencia II, del barrio Nigeria, sector Isla Trinitaria que permita la automatización y control de los procesos y reglas del negocio.

#### **1.4.2. Objetivos específicos.**

- ⊙ Analizar las necesidades tecnológicas y recursos de la hostería comunitaria.
- ⊙ Definir el diseño de Software de gestión de servicios de acuerdo a sus necesidades y requerimientos.
- ⊙ Capacitar a los usuarios en el manejo del software para la Hostería Comunitaria.

### **1.5. Justificación.**

El propósito de este proyecto nace de la necesidad existente de procesos y actividades que se requieren realizar en el negocio de una hostería tales como: almacenar la información de los clientes de forma segura y confiable, un sistema que soporte el registro y control sistemático de las reservas que se gestionan en la Hostería “África Mía”, Ubicada en el Cantón: Guayaquil, Parroquia Ximena Distrito Uno, Sector: Isla Trinitaria, Perimetral Sur Cooperativa Independencia II, Barrio Nigeria Dirección: Código 1042, solar 15. Y así poder brindar los servicios de: Hospedaje, Salón de eventos, Restaurante, Salón de Imagen, artesanías y trajes.

Además de contar con información actualizada y la disponibilidad de servicios y productos.

Una de las razones por lo que es necesario que la hostería cuente con el software de gestión de servicios es que de esta forma no se retrasarán los pedidos y se llevará un control de todos los servicios que se ofrecen en la hostería y que no permitirán que la misma alcance logros de ventajas competitivas.

### **1.5.1. Importancia.**

La hostería comunitaria es una obra emblemática de una asociación étnica de mujeres afrodescendientes, localizadas en un sector socioeconómico prioritario. El beneficio tiene un impacto directo en las mujeres asociadas y cabeza de hogar y de economía prioritaria, con niveles significativos de vulnerabilidad, por situaciones de violencia de género, así como sus hijos menores de edad. Beneficio que se extiende a los adultos mayores familiares. Además el impacto indirecto sobre la comunidad del entorno de los mismos huéspedes.

La importancia de esta presente tesis es el de estar actuando dentro del marco de un convenio de cooperación interinstitucional entre la universidad Politécnica Salesiana y la asociación afroecuatoriana “Mujeres Progresistas”, en el cual se impulsará el desarrollo social en el cual dice: Contribuir en el diseño, gestión, ejecución y evaluación de planes, programas y proyectos sociales y académicos de interés mutuo, enmarcados dentro de su misión y visión interinstitucional y que respondan a las necesidades y expectativas del desarrollo de sectores sociales específicos, local y nacional.

### **1.5.2. Necesidad**

El desarrollo de esta aplicación web surge debido a la necesidad de automatizar los procesos de negocio que actualmente maneja la Hostería Comunitaria África Mía y darle un valor agregado a los servicios que ofrece.

### **1.5.3. Beneficios que aporta**

- ✓ Mejorar la eficiencia en las diversas áreas de trabajo.
- ✓ Reducir tiempo en la ejecución de los procesos.
- ✓ Obtener mayor disponibilidad del recurso humano.
- ✓ Mejorar la planeación, control y uso de recursos.
- ✓ Alcanzar logros de ventajas competitivas.
- ✓ Tomar decisiones acertadamente.
- ✓ Disponer de información actualizada
- ✓ Obtener Información Confiable
- ✓ Mejorar el servicio al cliente

### **1.5.4. Beneficiarios.**

Los beneficiarios de este proyecto son la Asociación afroecuatoriana de Mujeres Progresistas y la Hostería Comunitaria África Mía.

## CAPITULO 2

### MARCO REFERENCIAL DE LA INVESTIGACIÓN

#### 2.1. Marco teórico.

##### 2.1.1. Fundamentos de la web.

El éxito de la web se basa en dos factores fundamentales: el protocolo HTTP y el lenguaje HTML. El primero permite una implementación sencilla de un sistema de comunicaciones que permite enviar cualquier fichero de forma fácil, simplificando el funcionamiento del servidor y posibilitando que servidores poco potentes atiendan cientos o miles de peticiones y reduzcan de este modo los costes de despliegue. El segundo, el lenguaje HTML, proporciona un mecanismo sencillo y muy eficiente de creación de páginas enlazadas. (Cibernética, s.f.)

##### 2.1.2. El protocolo HTTP.

El protocolo HTTP (Hypertext Transfer Protocol) es el protocolo principal de la World Wide Web. Es un protocolo simple, orientado a conexión y sin estado. Está orientado a conexión porque emplea para su funcionamiento un protocolo de comunicaciones (TCP, o Transport Control Protocol) de modo conectado, que establece un canal de comunicaciones entre el cliente y el servidor, por el cual pasan los bytes que constituyen los datos de la transferencia, en contraposición a los protocolos denominados de datagrama (o no orientados a conexión) que dividen la serie de datos en pequeños paquetes (o datagramas) antes de enviarlos, pudiendo llegar por diversas vías del servidor al cliente. Este protocolo no mantiene estado o, dicho de otro modo, cada transferencia de datos es una conexión diferente a la anterior, sin relación entre ellas. Por ejemplo, para transferir una página web se debe enviar el código HTML del texto, por un lado, y las imágenes que la componen por otro, pues en la especificación inicial 1.0 del protocolo HTTP se abrían y utilizaban tantas conexiones como componentes tenía la página, transfiriendo cada componente por conexiones diferentes. (Cibernética, s.f.)

Existe una variante de HTTP denominada HTTPS (S significa "secure", o "seguro") que utiliza el protocolo de seguridad SSL (o "Secure Socket Layer") para cifrar y autenticar el tráfico de datos, muy utilizada por los servidores web orientados al comercio electrónico o por aquellos que albergan información de tipo personal o confidencial. De forma esquemática, el funcionamiento de HTTP es como sigue: el cliente establece una conexión TCP con el servidor, hacia el puerto por defecto para el protocolo HTTP (o el indicado expresamente en la conexión), envía una orden HTTP de solicitud de un recurso (añadiendo algunas cabeceras con información) y, utilizando la misma conexión, el servidor responde enviando los datos solicitados y, además, añadiendo algunas cabeceras con información. (Cibernética, s.f.)

El protocolo también define la forma de codificar el paso de parámetros entre dos páginas, la tunelización de las conexiones (para sistemas de firewall), define la posible existencia de servidores intermedios de caché, etc. Las directivas que se utilizan para la petición de información según se definen en HTTP 1.1 (la versión considerada estable y ampliamente utilizada) son: GET: Petición de un recurso.

- POST: Petición de un recurso enviando parámetros.
- HEAD: Petición de algunos datos sobre un recurso.
- PUT: Creación o envío de un recurso.
- DELETE: Borrado de un recurso.
- TRACE: Devolución de la petición a su origen tal como la había recibido el receptor, para la depuración de errores.
- OPTIONS: Comprueba la capacidad del servidor.
- CONNECT: Se reserva para su uso en servidores intermedios con capacidad para funcionar como túneles entre otros servidores.

Se detallan a continuación algunos de entre estos comandos, ya que su comprensión es básica para desarrollar aplicaciones web. Se debe destacar que todos los recursos servidos mediante HTTP deberán ser referenciados utilizando una URL ("Universal Resource Locators"). (Cibernética, s.f.)

### **2.1.2.1. Peticiones en el protocolo HTTP: GET y POST**

Las peticiones en HTTP se pueden realizar utilizando 2 métodos. El método GET, cuando envía parámetros con la petición, lo hace codificándolos en la URL. El método POST lo hace como parte del cuerpo de la petición. Una petición GET tiene este formato:

```
GET /index.html HTTP/1.1
Host: www.unejemplo.com
User-Agent: Mozilla/4.5 [en]
Accept: image/jpeg, image/gif, text/html
Accept-language: en
Accept-Charset: iso-8859-1
```

La petición está formada por:

1. Línea de la petición: contiene el recurso que se solicita.
2. Cabecera de la petición: contiene la información adicional sobre el cliente que hace la solicitud.
3. Cuerpo de la petición: en peticiones de tipo POST y otras contiene más información adicional. (Cibernética, s.f.)

### **2.1.2.2. Línea de petición**

La línea de la petición está formada por estos elementos:

1. Método: nombre del método HTTP utilizado (GET, POST, etc.).
2. Identificador del recurso: URL ("Uniform Resource Locator").
3. Versión del protocolo utilizado. (Cibernética, s.f.)

### **2.1.2.3. Cabecera de la petición**

Contiene información adicional que ayuda al servidor a procesar correctamente la petición.

La información se facilita en forma de:

Identificador: valor

Los identificadores más importantes son:

- Host: nombre del servidor.
- User-Agent: nombre del navegador o del programa usado para acceder al recurso solicitado.
- Accept: se indican los formatos de texto e imagen aceptados por el User-Agent.
- Accept-Language: idiomas que soporta (preferentemente) el cliente. (Cibernética, s.f.)

### **2.1.2.4. Parámetros de la petición**

Una petición HTTP puede contener parámetros, por ejemplo, como respuesta a un formulario de registro o a una selección de entre los productos en una tienda virtual. Tales parámetros pueden pasarse de 2 formas:

- Formando parte de la propia cadena de la petición, codificados como parte de la misma URL.
- Como datos añadidos a la petición. (Cibernética, s.f.)

Para codificar los parámetros como parte incluida en la URL, éstos deben añadirse a la URL detrás del nombre del recurso, separándolos de éste mediante el caracter "?". Los parámetros se separan entre sí mediante el caracter "&". Los espacios se sustituyen por el caracter "+". Los caracteres especiales (los mencionados antes de "&", "+" y "?", y los caracteres que no son imprimibles, etc.) se representan mediante "%xx", donde "xx" representa el código en codificación ASCII en hexadecimal del caracter en cuestión. (Cibernética, s.f.)

Por ejemplo:

```
http://www.ejemplo.com/indice.jsp?
nombre=Fulano+Mengano&OK=1
```

Que en la petición HTTP quedaría:

```
GET /indice.jsp?nombre=Fulano+Mengano&OK=1 HTTP/1.0
Host: www.unejemplo.com
User-Agent: Mozilla/4.5 [en]
Accept: image/jpeg, image/gif, text/html
Accept-language: en
Accept-Charset: iso-8859-1
```

Para pasar los parámetros como datos añadidos, se envían al servidor en el cuerpo del mensaje de la petición. Por ejemplo, siguiendo con la petición:

```
POST /indice.jsp HTTP/1.0
Host: www.unejemplo.com
http://www.unejemplo.com/indice.jsp?
nombre=Fulano+Mengano&OK=1
User-Agent: Mozilla/4.5 [en]
```

Accept: image/jpeg, image/gif, text/html  
Accept-language: en  
Accept-Charset: iso-8859-1  
nombre=Perico+Palotes&OK=1 (Cibernética, s.f.)

Se debe destacar que para pasar los parámetros en el cuerpo de la petición, ésta se debe realizar como POST (no como GET), aunque una petición POST puede llevar parámetros en la línea de petición (igual que una GET). Los parámetros pasados en el cuerpo de la petición están codificados igual que si los pasamos mediante la URL, o pueden usar una codificación específica derivada del formato MIME ("Multipurpose Internet Mail Extensions"), conocida como codificación multiparte. (Cibernética, s.f.)

El ejemplo anterior en formato multiparte quedaría:

```
POST /indice.jsp HTTP/1.0
Host: www.unejemplo.com
User-Agent: Mozilla/4.5 [en]
Accept: image/jpeg, image/gif, text/html
Accept-language: en
Accept-Charset: iso-8859-1
Content-Type: multipart/form-data,
delimiter="----ALEATORIO----"

----ALEATORIO----
Content-Disposition: form-data; name="nombre"
Fulano Mengano
----ALEATORIO----
Content-Disposition: form-data; name="OK"
1
----ALEATORIO-----
```

Esta codificación es propia y exclusiva del método POST. Se emplea para el envío de ficheros al servidor. (Cibernética, s.f.)

### 2.1.2.5. Respuestas en el protocolo HTTP

Las respuestas en el protocolo HTTP son similares a las peticiones. Una respuesta estándar sería similar a esto:

```
HTTP/1.1 200 OK
Date: Mon, 04 Aug 2003 16:25:10 GMT
Server: Apache/2.0.40 (Red Hat Linux)
Last-Modified: Tue, 26 Mar 2004 08:53:53 GMT
Accept-Ranges: bytes
Content-Length: 428
Connection: close
```

```
<HTML>
```

...

Podemos observar que la primera línea responde con la versión del protocolo utilizada para enviar la página, seguida por un código de retorno y lo que se denomina una frase de retorno.

Después del estatus aparecen unos campos de control, que tienen el mismo formato que en las cabeceras de la petición. (Cibernética, s.f.)

### 2.1.3. El lenguaje HTML

#### 2.1.3.1. ¿Qué es html?

HTML es el acrónimo de HyperText Markup Language (Lenguaje de Marcado de Hipertexto) y es el lenguaje que se utiliza para crear las páginas web. Este lenguaje indica a los navegadores cómo deben mostrar el contenido de una página web.

El lenguaje HTML contiene dos partes:

- El contenido, que es el texto que se verá en la pantalla de un ordenador, y las etiquetas y atributos que estructuran el texto de la página web en encabezados, párrafos, listas, enlaces, etc. y normalmente no se muestra en pantalla.

-Las etiquetas, que son un conjunto de caracteres que rodean partes del documento, están formadas por el símbolo. (Ruiz, 1999 - 2015)

### **2.1.3.2. HTML5**

(HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML.<sup>1 2</sup> Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

La versión definitiva de la quinta revisión del estándar se publicó en octubre de 2014.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5.

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C. (Wales, 2001)

### **2.1.4. ¿Qué es un servidor FTP (FTP Servers)?**

Uno de los servicios más antiguos de Internet, File Transfer Protocol permite mover uno o más archivos con seguridad entre distintos ordenadores proporcionando seguridad y organización de los archivos así como control de la transferencia.

La seguridad se ha convertido en un tema candente. Durante años, los servidores ftp comunicaban con los clientes "en abierto," es decir, que la información de la conexión y de la contraseña era vulnerables a la interceptación. Ahora, los servidores ftp, tales como BulletProof FTP, SecureFTP, SurgeFTP, TitanFTP, y WS\_FTP, soportan SSL/TLS y utilizan el mismo tipo de cifrado presente en los sitios web seguros. Con SSL/TLS, los servidores ftp pueden cifrar los comandos de control entre los clientes del ftp y el servidor, así como los datos del archivo. Con

la ayuda del PGP, como en WS\_FTP pro, los datos del archivo se aseguran todavía más con el cifrado público. (Ruiz, 1999 - 2015)

### **2.1.5. ¿Qué es XAMPP?**

XAMPP es el entorno más popular de desarrollo con PHP y una distribución de Apache completamente gratuita y fácil de instalar que contiene MySQL, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar. (Vogelgesang, 2002)

### **2.1.6. Definición de php**

PHP es un lenguaje de programación muy potente que, junto con HTML, permite crear sitios web dinámicos. Php se instala en el servidor y funciona con versiones de Apache, Microsoft IIS, Netscape Enterprise Server y otros.

La forma de usar php es insertando código php dentro del código HTML de un sitio web. Cuando un cliente (cualquier persona en la web) visita la página web que contiene éste código, el servidor lo ejecuta y el cliente sólo recibe el resultado. Su ejecución, es por tanto en el servidor, a diferencia de otros lenguajes de programación que se ejecutan en el navegador.

Php permite la conexión a numerosas bases de datos, incluyendo MySQL, Oracle, ODBC, etc. Y puede ser ejecutado en la mayoría de los sistemas operativos (Windows, Mac OS, Linux, Unix. (elwebmaster.com, 2007-2008)

Una de las grandes ventajas que tiene PHP es su gran versatilidad a la hora de trabajar con bases de datos, permitiendo realizar páginas con información almacenada en forma persistente en bases de datos y mostrarla según la petición de los usuarios. (elwebmaster.com, 2007-2008)

Dentro de un archivo PHP se puede alternar entre código html y código PHP. Esto nos permite realizar páginas dinámicas de forma fácil: realizando toda la estructura con html y luego agregando el contenido dinámico.

Imaginemos una página html común, pero en lugar de ser index.html es index.php. Cuando se quiera agregar código PHP habrá que hacerlo dentro de las etiquetas `<?php y ?>` o `<? y ?>`, por ejemplo:

```
<html>
<head>
<title>hola mundo</title>
</head>
<body>
<h1>incluir código PHP en html</h1>
<p><? acá va el código PHP ?></p>
</body>
</html>
```

(elwebmaster.com, 2007-2008)

### 2.1.6.1. Variables

Las variables son elementos que creamos para asignarles un valor o dato específico (que puede cambiar a lo largo de nuestro archivo). En PHP, las variables se definen anteponiendo el signo \$, por ejemplo:

```
$hola = "hola mundo";
```

En el código anterior definimos la variable hola la cual contiene la información hola mundo. Noten que al final agregué un punto y coma (;). En PHP, siempre que se termine una instrucción hay que aclararlo de esta forma. (elwebmaster.com, 2007-2008)

Lo ideal a la hora de programar es definir todas las variables y realizar las operaciones antes del código HTML y en ese código mostrar los resultados de nuestras variables. Por ejemplo, un código completo de hola mundo sería:

```
<? $hola = "hola mundo"; ?>
<html>
<head>
<title><? echo $hola; ?></title>
</head>
<body>
<h1>incluir código PHP en html</h1>
<p><? echo $hola; ?></p>
</body>
</html>
```

La función echo lo que hace es imprimir una o más cadenas de texto. En nuestro ejemplo anterior, como la variable contenía una cadena de texto (string), lo que el navegador hubiera mostrado es:

```
<html>
<head>
<title>hola mundo</title>
</head>
<body>
<h1>incluir código PHP en html</h1>
<p>hola mundo</p>
</body>
</html>
```

(elwebmaster.com, 2007-2008)

### 2.1.7. ¿Qué es una Base de Datos?

Las bases de datos son un conjunto de datos almacenados sistemáticamente para su uso posterior. Gracias al avance de la informática, la mayoría de las bases de

datos se encuentran en formato digital, por lo que una de las formas para acceder a la información guardada en ellas es a través de una computadora. (elwebmaster.com, 2007-2008)

#### **2.1.7.1. Importancia de las Bases de Datos**

Php ofrece interfaces para el acceso a la mayoría de las bases de datos, entonces de esta forma podremos almacenar y acceder a estos datos (se dice que los datos guardados de esta forma son “datos persistentes”) a través de una página web realizada en PHP.

Por ejemplo, un sitio como este, almacena todos los talleres en una base de datos, de modo que, cuando ustedes acceden a la clase 10, el archivo encargado de mostrar todas las clases, busca en la base de datos la clase 10 y muestra todo el contenido guardado.

Es siempre el mismo archivo el encargado de interactuar con la base de datos y es por eso que se llaman páginas dinámicas, ya que es siempre la misma y sólo cambia el contenido que se muestra en ella.

La base de datos más común para utilizar con PHP es MySQL, debido a que es muy potente, gratuito y se encuentra en la mayoría de los servicios de hosting de páginas webs. Otras bases utilizadas son PostgreSQL, ODBC, Oracle, IBM DB2, etc. (elwebmaster.com, 2007-2008)

#### **2.1.7.2. Interacción con PHP**

La interacción entre php y las base de datos se realiza a través de herramientas propias del lenguaje Php, por un lado, y escribiendo los pedidos a la base de datos en un idioma universal,SQL (Structured Query Language) por otro, que podremos encontrar en todas las bases de datos.

MySQL es un sistema de base de datos open source (gratuito y modificable), y es gracias a ello que se convirtió en la base más popular del mundo. Incluso su continuo desarrollo y su impresionante popularidad están haciendo que MySQL sea un competidor directo de las grandes empresas como Oracle o IBM. (elwebmaster.com, 2007-2008)

Este sistema de base de datos muy utilizado sobre todo con PHP es relacional, ya que utiliza múltiples tablas para almacenar y organizar la información. También es multihilo porque se pueden realizar varias tareas concurrentemente. Y como última característica es multiusuario, pudiendo conectarse varios usuarios simultáneamente. (elwebmaster.com, 2007-2008)

#### **2.1.7.3. Ventajas de MySQL.**

- Es una base de datos muy rápida en la lectura en aplicaciones web, convirtiéndola en la herramienta ideal para este tipo de aplicaciones.
- Por otro lado está disponible en gran cantidad de plataformas y sistemas, brindando a su vez, una conectividad muy segura.
- Nos da también una potencia sin igual, ya que utiliza SQL como lenguaje para consultar la base de datos (usar ese motor nos ahorrará una gran cantidad de trabajo).
- También es muy portable por la misma característica anterior, SQL es fácilmente portable a otras plataformas y nos brinda una gran escalabilidad, pudiendo manipular base de datos enormes con más de 50 millones de registros. (elwebmaster.com, 2007-2008)

#### **2.1.7.4. Conectarse con PHP a MySQL.**

PHP nos brinda muchísimas herramientas a la hora de conectarnos con MySQL, pudiendo realizar conexiones simultáneas y todo tipo de consultas para obtener

información (a medida que avance este curso, vamos ir viendo todas estas posibilidades).

Básicamente es necesario tener la IP del servidor de base de datos (por lo general se encuentra en el mismo lugar físico que el servidor apache con PHP, utilizando como IP para conexión la palabra "localhost"). El nombre de usuario y contraseña de conexión y el nombre de la base de datos a la que queremos conectarnos.

Con esos elementos ya podemos crear un script de conexión a la base de datos y poder utilizar las funciones de PHP predefinidas para interactuar con la misma. (elwebmaster.com, 2007-2008)

### **2.1.8. Sublime text 2**

Sublime Text 2 es un editor de texto pensado para escribir código en la mayoría de lenguajes de programación y formatos documentales de texto, utilizados en la actualidad: Java, Python, Perl, HTML, JavaScript, CSS, HTML, XML, PHP, C, C++, etc. Permite escribir todo tipo de documentos de código en formato de texto y es capaz de colorear el código, ayudarnos a la escritura, corregir mientras escribimos, usar abreviaturas (snippets), ampliar sus posibilidades, personalizar hasta los últimos detalles, casi cualquier cosa que le podamos pedir a un editor. (Sánchez, 2012)

### **2.1.9. AJAX**

El término AJAX se presentó por primera vez en el artículo "Ajax: A New Approach to Web Applications" publicado por Jesse James Garrett el 18 de Febrero de 2005. Hasta ese momento, no existía un término normalizado que hiciera referencia a un nuevo tipo de aplicación web que estaba apareciendo.

En realidad, el término AJAX es un acrónimo de *Asynchronous JavaScript + XML*, que se puede traducir como "JavaScript asíncrono + XML".

El artículo define AJAX de la siguiente forma:

Ajax no es una tecnología en sí mismo. En realidad, se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes.

Las tecnologías que forman AJAX son:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías. (LIBROSWEB, 2015)

#### **2.1.10. Hostería**

Hostería.- Es Hostería todo establecimiento hotelero, situado fuera de los núcleos urbanos, preferentemente en las proximidades de las carreteras, que esté dotado de jardines, zonas de recreación y deportes y en el que, mediante precio, se preste servicios de alojamiento y alimentación al público en general, con una capacidad no menor de seis habitaciones. (Ejecutivo, 2012)

##### **2.1.10.1. Servicios que ofrecerá la Hostería**

###### ☉ Servicio de Alojamiento

La hostería contará con 5 habitaciones con capacidad de 5 pax. Cada una. Su equipamiento cuenta con baño privado, ventilador y TV.

###### ☉ Servicio de Alimentación

Dentro de la hostería se encuentra un restaurante, donde se podrá degustar gastronomía tanto Nacional como Internacional, teniendo como especialidades de la casa los platos típicos de los Afroecuatorianos.

###### ☉ Alquiler de Salón de Eventos

El alquiler de los salones, también incluye la organización total o parcial de los servicios, tales como:

Banquetería  
Diskjockey y Animación  
Mantelería, Cristalería  
Bebidas, decoración, etc.

⊙ Locales comerciales con venta de artesanías, sandalias y centros de asesoramiento de imagen.

El visitante tendrá la opción de adquirir souvenirs representativos y hechos por las mujeres Afroecuatorianas, tales como: camisetas, pulseras, sandalias, etc.

#### **2.1.11. Sistema de información**

Según Peralta: Define sistema de información como el conjunto de elementos que interactúan entre sí con el fin de apoyar las actividades de una empresa o negocio. Teniendo muy en cuenta el equipo computacional necesario para que el sistema de información pueda operar y el recurso humano que interactúa con el Sistema de Información, el cual está formado por las personas que utilizan el sistema.

Un sistema de información realiza cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información”.

Al momento que empieza la interacción con el sistema informático, éste cumple con cuatro actividades básicas: entrada, almacenamiento, procesamiento y salida de información. (Peralta, 2008)

#### **2.1.12. ¿Qué es Intranet?**

Una Intranet es una red de ordenadores privada basada en los estándares de Internet.

Las Intranets utilizan tecnologías de Internet para enlazar los recursos informativos de una organización, desde documentos de texto a documentos multimedia, desde bases de datos legales a sistemas de gestión de documentos. Las Intranets pueden incluir sistemas de seguridad para la red, tableros de anuncios y motores de búsqueda.

Una Intranet puede extenderse a través de Internet. Esto se hace generalmente usando una red privada virtual (VPN). (Ruiz, 1999 - 2015)

## 2.2. Matriz Causa y efecto

| <b>Problema General</b>   | <b>Objetivo General</b>  | <b>Hipótesis General</b>   |
|---|--|--|
| ¿Por qué es necesario la creación e implementación de un software de gestión de servicios en la hostería? | Creación de un software para la automatización de procesos y reglas de negocios de la hostería       | Al automatizar los procesos mediante un sistema de gestión de servicios permitirá mejorar la planeación, control y uso de recursos en la hostería. |
| <b>Problemas Específicos</b>  | <b>Objetivos Específicos</b>   | <b>Hipótesis Específicas</b>   |
| ¿Por qué la hostería necesita herramientas tecnológicas para optimizar los procesos dentro del negocio?   | Analizar las necesidades tecnológicas de la hostería comunitaria                                     | Al no poseer herramientas tecnológicas para realizar los procesos dentro del negocio la hostería no podría lograr ventajas competitivas            |
| ¿Por qué el desconocimiento de los sistemas de información puede retrasar el negocio?                     | Definir el diseño de Software de gestión de servicios de acuerdo a sus necesidades y requerimientos. | Al no conocer mucho sobre herramientas informáticas, las personas que trabajan en la hostería deberían tener una interfaz amigable y fácil de usar |
| ¿Por qué la ejecución de los procesos dentro de la Hostería Comunitaria no es eficiente?                  | Capacitar a las mujeres progresistas en el manejo del software para la Hostería Comunitaria          | El desconocimiento del manejo de las herramientas tecnológicas impide la reducción de tiempos en la ejecución de los procesos                      |

### **2.3. Marco Legal**

La Constitución de la República del Ecuador (2008) respalda el proyecto a presentarse, debido a que mencionan la necesidad de que toda actividad como la comercial tenga fines lucrativos para la sociedad. Recalcando el uso de las tecnologías, en donde el software a implementarse sería considerado como un valor favorable en el proyecto. A continuación el Artículo 16 con sus respectivos incisos mostrarán la importancia que tiene las actividades interculturales junto con el uso de las TICs en el gobierno ecuatoriano:

Art. 16.- Todas las personas, en forma individual o colectiva, tienen derecho:

1. Una comunicación libre, intercultural, incluyente, diversa y participativa, en todos los ámbitos de la interacción social, por cualquier medio y forma, en su propia lengua y con sus propios símbolos.

2. El acceso universal a las tecnologías de información y comunicación (Constitución de la República del Ecuador, 2008)

Adicionalmente dentro del plan nacional del buen vivir que es considerado como la herramienta del Gobierno Nacional para modular las políticas públicas con la gestión y la inversión pública (Plan Nacional del Buen vivir) se encuentra el objetivo 3 que impulsa la mejora de la calidad de vida de la población, apoyando de esta manera la iniciativa de este proyecto.

### **2.4. Formulación de hipótesis**

El desarrollo de un software de gestión de servicios para la hostería comunitaria de la agrupación “Mujeres Progresistas” de La Coop. Independencia II, sector Isla Trinitaria Distrito 3 de la región 8 del barrio Nigeria”, permitirá que los servicios de hostería mejoren la planeación, control y uso de recursos, reduzcan el tiempo en la ejecución de los procesos, dispongan de información actualizada, obtengan Información confiable, mejorando el servicio al cliente para así alcanzar eficiencia en las diversas áreas de trabajo y logros de ventajas competitivas.

## **2.5. Operacionalización de variables, e indicadores**

### **2.5.1. Variables Independientes**

Creación e implementación de un software de gestión de servicios para la hostería comunitaria “África Mía”.

### **2.5.2. Variables Dependientes**

Automatización de los procesos de negocio en la hostería comunitaria África Mía

- Integra y consolida la información.
- Simplifica y agiliza cada operación.
- Optimiza organiza y actualiza la gestión de servicios.
- Aumenta el control y flexibilidad de la hostería.
- Se adapta al desenvolvimiento del mercado y el negocio.

## CAPITULO 3

### MARCO METODOLÓGICO DE LA INVESTIGACIÓN

#### 3.1. Tipo de estudio

La investigación exploratoria impulsará a determinar el mejor diseño de la investigación, los métodos de recopilación de datos para cumplir con nuestro objetivo.

La investigación descriptiva ayudará a especificar el porqué de la falta de un sistema de gestión de servicios en la Hostería.

##### 3.1.1. Investigación exploratoria

Este tipo de investigaciones presenta una visión general del tema de estudio, sin llegar a desarrollar más que conceptos básicos o caracterizar una situación o problemática determinada. Dos son las características generales de este tipo de investigaciones:

1) El poco estudio del tema elegido, lo cual limita la formulación de hipótesis precisas o elaborar una descripción detallada; y,

2) La escasa contribución de la teoría existente a la comprensión del fenómeno o conjunto de fenómenos estudiados.

Ander - Egg (1977: 35) apunta, citando a Selltiz y otros, que las finalidades de este tipo de estudios son, entre otras:

- formular problemas;
- desarrollar hipótesis;
- familiarizar al investigador con el fenómeno que desea estudiar;
- aclarar conceptos;

- establecer preferencias para posteriores clasificaciones; y, reunir información acerca de posibilidades prácticas para llevar a cabo investigaciones en marcos de vida actual.

Estas investigaciones, si bien interesantes, exigen del investigador una extraordinaria creatividad y capacidad de improvisación, ya que implica la ausencia de guías teóricas que faciliten la comprensión del tema de estudio, aparte de la incertidumbre respecto a los resultados que seguramente provocará. De todos modos, no es el deseo del investigador lo que determina el tipo de estudio que se vaya a efectuar, sino la existencia previa del material referido al tema y el enfoque que quiera dársele. (Henández)

### **3.1.2. Investigación descriptiva**

En un estudio descriptivo se seleccionan una serie de conceptos o variables y se mide cada una de ellas independientemente de las otras, con el fin, precisamente, de describirlas.

Estos estudios buscan especificar las propiedades importantes de personas, grupos, comunidades o cualquier otro fenómeno. El énfasis está en el estudio independiente de cada característica, es posible que de alguna manera se integren las mediciones de dos o más características con el fin de determinar cómo es o cómo se manifiesta el fenómeno. Pero en ningún momento se pretende establecer la forma de relación entre estas características.

Su propósito es la delimitación de los hechos que conforman el problema de investigación (Vásquez, 2005).

### **3.2. Método de investigación**

Para llevar a cabo la ejecución del proyecto se utilizará como técnicas de investigación el método de análisis y el método experimental los cuales permitirán encontrar soluciones a los problemas planteados.

### **3.2.1. Método Experimental**

El método experimental permitirá mediante pruebas la demostración de las hipótesis antes y después de su implementación del sistema comprobando la efectividad con el cual se llevaran los procesos en la hostería.

### **3.2.2. Método de Análisis**

El método de análisis permitirá comparar la información obtenida mediante el uso del sistema y se demostrará que se ha cumplido con los objetivos planteados.

### **3.3. Fuentes y Técnicas para la recolección de la Información**

Para la obtención de la información necesaria para conocer las necesidades que existen en la Hostería África mía, se hará uso de los siguientes métodos de investigación:

Primaria: Se refiere al entorno social que brinda la información básica, de primera mano, para ir recabando información en base a entrevistas y encuestas.

Secundaria: Se refiere a las industrias e instituciones que aportan datos a través de publicaciones sobre el tema, La característica fundamental de la investigación en estos dos pasos, es el descubrimiento de principios generales.

Terciaria: Son aquellas que brindan información de investigaciones hechas por terceras personas. Estas se encuentran recopiladas en informes, tesis o tesinas, que se consultan como guía para el desarrollo de la investigación.

Mixta: Se refiere a todas las anteriores incluyendo las empresas de servicios con las que se harán estudios para analizar la información recabada finalmente. (Henández)

Dentro de las técnicas a aplicarse para la recopilación de información que permitirá el desarrollo de este proyecto, se tiene:

- La encuesta, dirigida a las asociaciones y entidades étnicas afrodescendientes, además del público en general, para conocer su aceptación frente al proyecto. Se utilizará un cuestionario de preguntas abiertas, de manera presencial, y mediante el uso del Internet (Anexo 1).

- Grupo focal, guiado a los trabajadores de la hostería con el fin de captar sus necesidades y mostrarles la propuesta de un acceso amigable y comprensible que agilite los procesos administrativos y operacionales. Se manejará una guía diseñada que contendrá preguntas específicas (Anexo 2).

- Método Delphi, enfocado a expertos en sistemas para agilizar y optimizar los procesos operacionales de la hostería. Se contará con un bosquejo inicial del software.

- Entrevista a un experto en el manejo administrativo de hostería, para conocer el funcionamiento de los softwares que implementan, con el fin de pulir el bosquejo preliminar. Se utilizará cuestionario de preguntas (Anexo 3).

## CAPITULO 4

### ANÁLISIS Y DISEÑO DE LA PROPUESTA

#### 4.1. Resultado de la Investigación

El principal propósito de este software a desarrollar en el presente proyecto de tesis es automatizar los procesos de negocios mediante el uso de una aplicación web, el comportamiento adecuado por parte del usuario. El alcance de esta aplicación es demostrar que se puede mantener un control y mayor eficiencia en los procesos que se realicen dentro de la hostería.

#### 4.2. Arquitectura de Aplicación



Figura 1: Arquitectura de Aplicación

Elaborado por: Autores

### 4.3. Diagrama de escenarios



Figura 2: Diagrama de Escenarios de la Aplicación Web

Elaborado por: Autores

Aquí se observa el diagrama de escenarios de la aplicación en donde el usuario inicia sesión en la plataforma web, realiza el registro de clientes, empleados huéspedes así como también puede generar reportes de eventos, huéspedes, productos, clientes empleados y facturas emitidas, los cuales a su vez se pueden modificar o eliminar. Por ultimo está el proceso de emisión de la factura, el cual es visualizado por el cliente.

#### 4.4. Diagrama de Estados

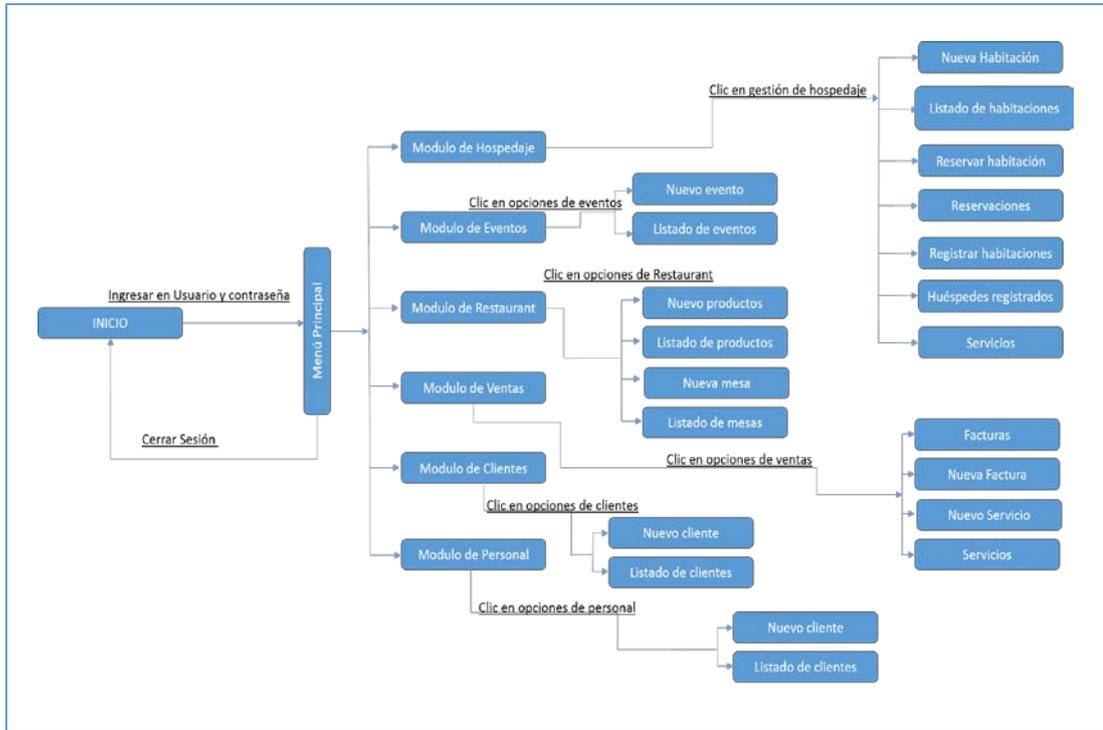


Figura 3: Diagrama de Estados

Elaborado por: Autores

Un evento es un conjunto de acontecimientos que suceden en un determinado tiempo y que pueden ocurrir durante la manipulación de la plataforma web.

## 4.5. Toma de Requerimientos

### a. Requerimientos del Cliente

Tabla 4.5.1 Requerimientos del cliente

|  |     |                   |                        |
|--|-----|-------------------|------------------------|
| <b>Responsables:</b> Ricardo Mesías y Mariana Pintado  |     |                   |                        |
| <b>ID:</b>   | 001 | <b>Prioridad:</b> | 1 Gestión de servicios |
| <b>Descripción:</b> Proceso de gestión de servicios de la Hostería Comunitaria África mía  |     |                   |                        |
| 1. Registro de Huéspedes<br>2. Datos personales de los Huéspedes<br>3. Impresión de la factura<br>4. Registro de servicios<br>5. Reportes  |     |                   |                        |
| <b>Dependencias:</b>   |     |                   |                        |
| El huésped no podrá hospedarse en la hostería sin haber hecho el check in y haber registrado la información necesaria, requisito que se necesita para poder hospedarse en la hostería. |     |                   |                        |
| El cliente podrá acceder a cualquier servicio que brinde la hostería sin necesidad de estar registrado como huésped.   |     |                   |                        |

Elaborado por: Autores

### b. Requerimientos del Desarrollador

- Requerimientos no funcionales

Tabla 4.5.2 Requerimientos No funcionales

|  |                        |                  |     |
|--|------------------------|------------------|-----|
| <b>ID:</b>   | 002                    | <b>Relación:</b> | 001 |
| <b>Descripción:</b>                                    | Colaboración de Tareas |                  |     |
| Pedir información de parte de las mujeres progresistas |                        |                  |     |

Elaborado por: Autores

- Requerimientos Funcionales

Tabla 4.5.3 Requerimientos Funcionales

|  |   |                  |     |
|--|---|------------------|-----|
| <b>ID:</b>   | 003   | <b>Relación:</b> | 001 |
| <b>Prioridad:</b>  | 1. Registro de huéspedes<br>2. Registro de servicios<br>3. Datos personales de los huéspedes<br>4. Reportes |                  |     |
| <b>Descripción:</b>  | Requerimientos de datos para el sistema   |                  |     |
| Pedir datos de servicios, platos, productos y procedimientos de registro, procedimientos de facturas, reportes, que reportes se necesita para el control de pedidos, mesas ocupadas, listado de huéspedes registrados. |   |                  |     |

Elaborado por: Autores

## 4.6. Casos de Uso

### 4.6.1. Caso de uso de Restaurante

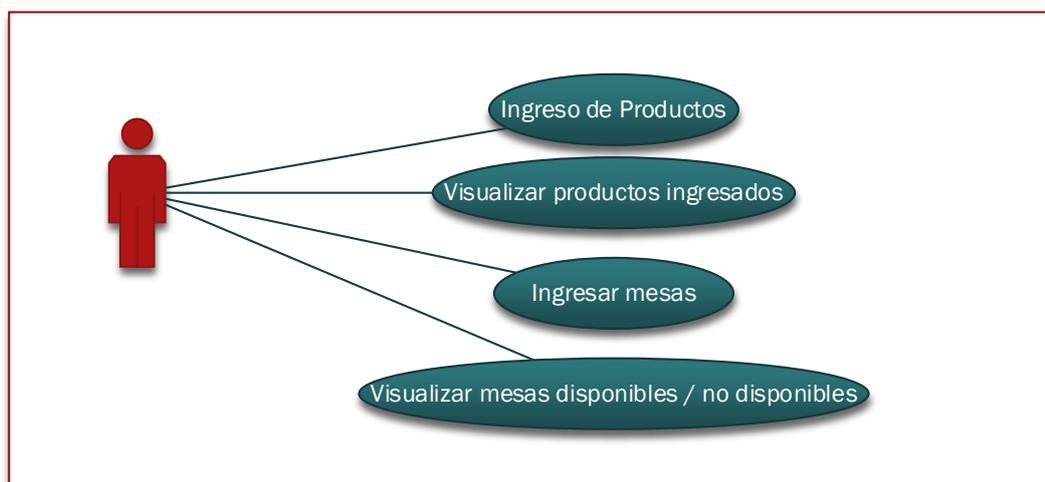


Figura 4: Diagrama de caso de uso de Restaurante

Elaborado por: Autores

Tabla 1 Ingresando Productos o servicios

|  |                                       |
|--|---------------------------------------|
| <b>Caso de Uso:</b> Ingresando Productos o servicios               |                                       |
| <b>Actor:</b> Cajero / Contadora                                   |                                       |
| <b>Curso:</b> Normal   | <b>Alternativas</b>                   |
| 1. El cajero podrá seleccionar la opción nuevo producto o servicio |                                       |
| 2. Se ingresa una descripción corta y una descripción larga        |                                       |
| 3. El cajero ingresa el Precio del producto o servicio             |                                       |
| 4. El cajero ingresa la cantidad del producto                      | El producto ya podría estar ingresado |
| 5. El sistema almacena el nuevo producto o servicio                | El servicio ya podría estar ingresado |

Elaborado por: Autores

Tabla 2 Visualizando productos y servicios

|  |   |
|--|---|
| <b>Caso de Uso:</b> Visualizando productos y servicios   |   |
| <b>Actor:</b> Cajero / Contadora   |   |
| <b>Curso:</b> Normal   | <b>Alternativas</b>                                 |
| 1. En la sección Restaurante el cajero selecciona la opción listado de productos o servicio              |   |
|  | El cajero podrá modificar algún producto o servicio |
|  | El cajero podrá eliminar algún producto o servicio  |
| 2. En la página de stock que se visualiza luego del paso 1, el cajero podrá buscar el producto que desee |   |
| 3. El sistema almacenará los cambios que se realizaron en el nuevo producto o servicio                   |   |

Elaborado por: Autores

Tabla 3 Creando nueva mesa

|  |                     |
|--|---------------------|
| <b>Caso de Uso:</b> Creando nueva mesa |                     |
| <b>Actor:</b> Cajero / Contadora       |                     |
| <b>Curso:</b> Normal                   | <b>Alternativas</b> |

|  |  |
|--|--|
| 1. En la sección Restaurante el cajero podrá seleccionar la opción “Nueva mesa”                            |  |
| 2. El cajero al crear una mesa deberá poner la descripción, una observación y deberá seleccionar el estado | El cajero podrá seleccionar el estado de la mesa, es decir si esta libre, ocupada o reservada. |
| 3. El cajero deberá presionar guardar  |  |
| 4. El sistema almacenará el estado de la nueva mesa  |  |

Elaborado por: Autores

Tabla 4 Visualizando mesas

|  |
|--|
| <b>Caso de Uso:</b> Visualizando mesas   |
| <b>Actor:</b> Cajero / Contadora   |
| <b>Curso:</b> Normal   |
| 1. En la sección Restaurante el cajero podrá seleccionar la opción “Listado mesas”     |
| 2. El cajero podrá visualizar el estado de las mesas (libre, ocupada o reservada)      |
| 3. El cajero podrá cambiar el estado de la mesa  |
| 4. El cajero podrá modificar las mesas   |
| 5. El cajero podrá eliminar alguna mesa creada   |
| 6. El sistema almacenará los cambios que se realizaron en el nuevo producto o servicio |

Elaborado por: Autores

## 4.6.2. Caso de uso de Hospedaje

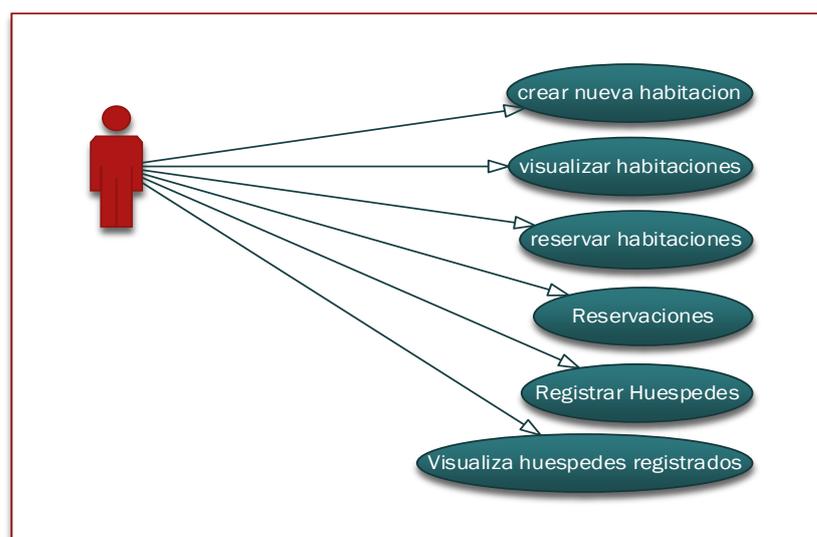


Figura 5: Diagrama de Casos de Uso de Hospedaje

Elaborado por: Autores

Tabla 5 Creando nueva habitación

| <b>Caso de Uso:</b> Creando nueva habitación   |  |
|--|--|
| <b>Actor:</b> Cajero / Contadora   |  |
| <b>Curso:</b> Normal   | <b>Alternativas</b>  |
| 1. El cajero podrá seleccionar la opción “Nueva Habitación”  |  |
| 2. El cajero al crear una habitación deberá poner la descripción de la habitación y el número de piso donde se encuentra | El cajero podrá seleccionar el estado de la mesa, es decir si está libre, ocupada o reservada.   |
| 3. El cajero deberá seleccionar el tipo de habitación  | Sencilla, doble, matrimonial y familiar  |
| 4. El cajero deberá digitar el número de baños con el que cuenta la habitación   |  |
| 5. El cajero deberá seleccionar los servicios adicionales con los que cuenta la habitación                               | El cajero podrá seleccionar de entre las siguientes opciones: teléfono, aire acondicionado, ventilador, ventana, televisión, tv pagada, wifi |
| 6. El cajero deberá seleccionar el estado de la habitación   | El cajero podrá seleccionar de entre las siguientes opciones: Libre, ocupada, reservada  |
| 7. El cajero deberá ingresar el costo de la habitación   |  |
| 8. El sistema almacenará la nueva habitación creada.   |  |

Elaborado por: Autores

Tabla 6 Visualizando Habitaciones

|  |
|--|
| <b>Caso de Uso:</b> Visualizando Habitaciones                                      |
| <b>Actor:</b> Cajero / Contadora   |
| <b>Curso:</b> Normal   |
| 1. El cajero podrá seleccionar la opción “Listado habitaciones”                    |
| 2. El cajero podrá visualizar el estado, el nombre y el tipo de habitación         |
| 3. El cajero podrá modificar la habitación que desee                               |
| 4. El cajero podrá eliminar alguna habitación                                      |
| 5. El sistema almacenará los cambios que se realizaron en la habitación modificada |

Elaborado por: Autores

Tabla 7 Registro de Huéspedes

|  |  |
|--|--|
| <b>Caso de Uso:</b> Registro de Huéspedes  |  |
| <b>Actores:</b>  | Recepcionista  |
| <b>Tipo:</b>   | Esencial   |
| <b>Precondición:</b>   | Si no se registra no puede pasar a las habitaciones  |
| <b>Post condición:</b>   | El huésped podrá acceder a cualquier servicio luego de estar registrado y cancelar en ese momento. |
| <b>Propósito:</b>  |  |
| Registrar los datos de las personas que se hospeden en la hostería   |  |
| <b>Resumen:</b>  |  |
| <p>Un huésped, llega a la hostería, se registra y para eso proporciona sus datos personales escoge el paquete turístico o la habitación, selecciona servicios adicionales en caso de requerirlos, se le ingresa al sistema todos los datos y el Sistema los guardará para asegurar que la información esté disponible cuando se la requiera.</p> |  |

Elaborado por: Autores

Tabla 8 Visualizando Huéspedes registrados

|   |
|---|
| <b>Caso de Uso:</b> Visualizando Huéspedes registrados  |
| <b>Actor:</b> Cajero / Contadora  |
| <b>Curso:</b> Normal  |
| 1. El cajero selecciona la opción “Listado huéspedes”   |
| 2. El cajero puede visualizar el listado de huéspedes, el nombre y el tipo de habitación en la que se encuentran registrados. |
| 3. El cajero podrá modificar los datos del huésped que desee  |
| 4. El sistema almacena los cambios que se realizaron en la habitación modificada  |

Elaborado por: Autores

### 4.6.3. Caso de uso de Ventas

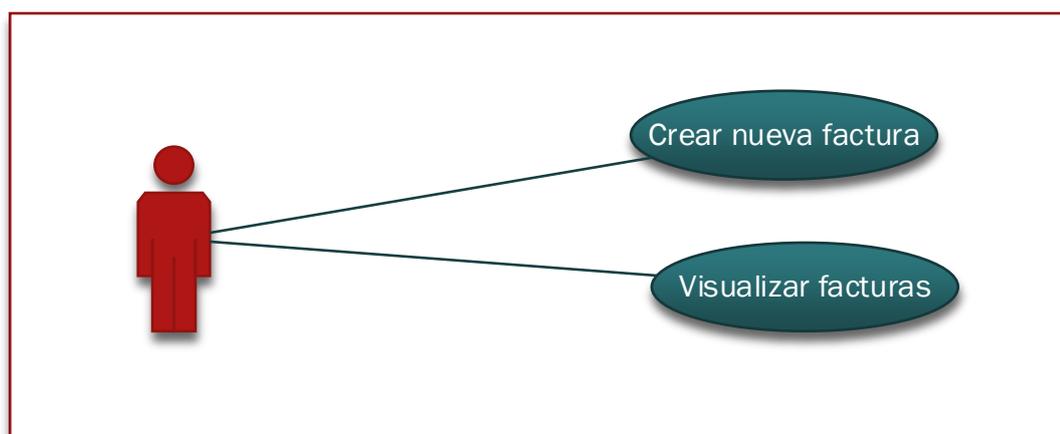


Figura 6: Diagrama de Casos de Uso de Ventas

Elaborado por: Autores

Tabla 9 Creando Facturas

|  |
|--|
| <b>Caso de Uso:</b> Creando Facturas   |
| <b>Actor:</b> Cajero / Contadora   |
| <b>Curso:</b> Normal   |
| 1. El cliente deberá proporcionar las credenciales al cajero   |
| 2. El cajero deberá ingresar el número de factura, fecha de emisión, cliente, Cédula de identidad o Ruc, dirección, mesa, forma de pago, estado del documento, producto, cantidad. |
| 3. El cajero deberá presionar agregar el producto o servicio.  |
| 4. El sistema emitirá la factura.  |

Elaborado por: Autores

Tabla 10 Visualizando Facturas

|   |
|---|
| <b>Caso de Uso:</b> Visualizando Facturas   |
| <b>Actor:</b> Cajero / Contadora  |
| <b>Curso:</b> Normal  |
| 1. El cajero podrá seleccionar la opción "Facturas"   |
| 2. El cajero podrá visualizar por fecha todas las facturas que se hayan realizado               |
| 3. El cajero también podrá buscar por número de factura   |
| 4. El cajero podrá generar un tipo de reporte diario o mensual desde y hasta una fecha indicada |
| 5. El cajero podrá realizar una nueva factura   |

Elaborado por: Autores

#### 4.6.4. Caso de uso de Clientes

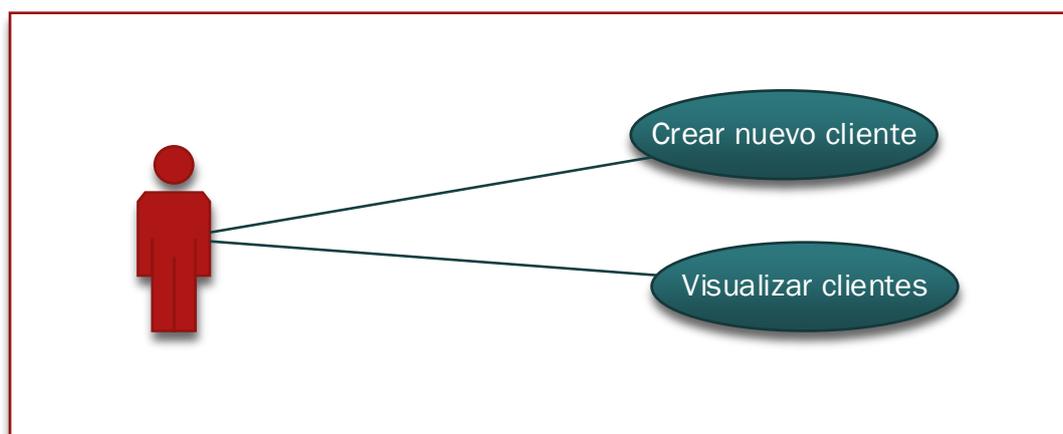


Figura 7: Diagrama de Casos de Uso de Clientes

Elaborado por: Autores

Tabla 11 Creando nuevo cliente

|   |
|---|
| <b>Caso de Uso:</b> Creando nuevo cliente                                       |
| <b>Actor:</b> Cajero / Contadora  |
| <b>Curso:</b> Normal  |
| 1. El cajero deberá ingresar el nombre, apellido, cedula, dirección y teléfono. |
| 2. El cajero deberá presionar guardar   |
| 3. El sistema generará la factura   |

Elaborado por: Autores

Tabla 12 Visualizando Clientes

|   |
|---|
| <b>Caso de Uso:</b> Visualizando Clientes                                     |
| <b>Actor:</b> Cajero / Contadora  |
| <b>Curso:</b> Normal  |
| 1. El cajero podrá visualizar todos los clientes ingresados.                  |
| 2. Se puede podrá visualizar la cedula, nombre, apellido, dirección, teléfono |
| 3. El cajero podrá realizar una búsqueda por cliente                          |

Elaborado por: Autores

#### 4.6.5. Caso de uso de Eventos

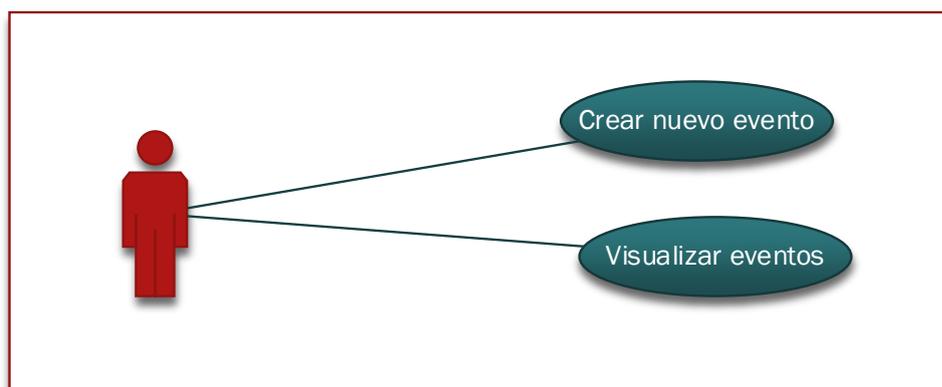


Figura 8: Diagrama de casos de uso de Eventos

Elaborado por: Autores

Tabla 13 Creando eventos

|   |
|---|
| <b>Caso de Uso:</b> Creando eventos   |
| <b>Actor:</b> Cajero / Contadora  |
| <b>Curso:</b> Normal  |
| 1. El cajero deberá ingresar la fecha del evento, el nombre de quien reserva, el tipo de evento, el nombre del evento, la duración, el número de personas, costo, si cuenta con animación, si cuenta con buffet, y el costo total del evento. |
| 2. El sistema almacenará el evento creado.  |

Elaborado por: Autores

Tabla 14 Visualizando eventos

|   |
|---|
| <b>Caso de Uso:</b> Visualizando eventos  |
| <b>Actor:</b> Cajero / Contadora  |
| <b>Curso:</b> Normal  |
| 3. El cajero podrá visualizar todos los eventos.  |
| 4. Se podrá visualizar los eventos por fecha, tipo de evento, descripción, servicios adicionales. |
| 5. El cajero podrá facturar, modificar o eliminar algún evento.                                   |

Elaborado por: Autores

#### 4.6.6. Caso de uso de Personal

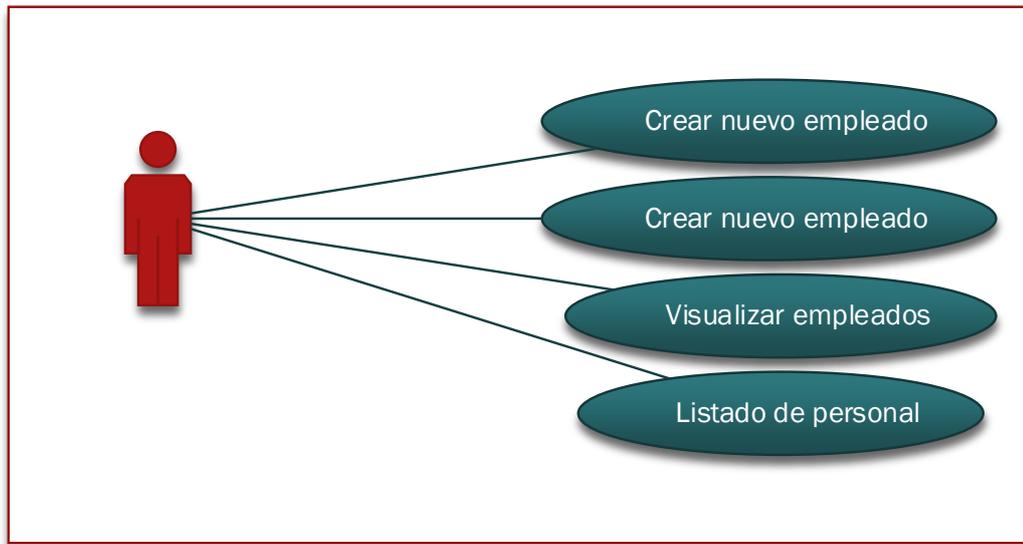


Figura 9: Diagrama de Casos de Uso de Personal

Elaborado por: Autores

Tabla 15 Visualizando y creando meseros o cajeros

|   |
|---|
| <b>Caso de Uso:</b> Visualizando y creando meseros o cajeros            |
| <b>Actor:</b> Cajero / Contadora  |
| <b>Curso:</b> Normal  |
| 1. El cajero podrá visualizar todos los meseros o cajeros.              |
| 2. Se podrá visualizar la cedula, nombre, apellido, dirección, teléfono |
| 3. El cajero podrá crear un nuevo mesero o cajero                       |

Elaborado por: Autores

#### 4.7. Requisitos de Hardware y Software

Para las pruebas del de la aplicación web se utilizará lo siguiente:

Hardware:

Portátil AMD 4GB RAM, 1.3GHz 64 bits

Software:

XAMP SERVER

Sublime tex2

## 4.8. Diagrama Entidad – Relación

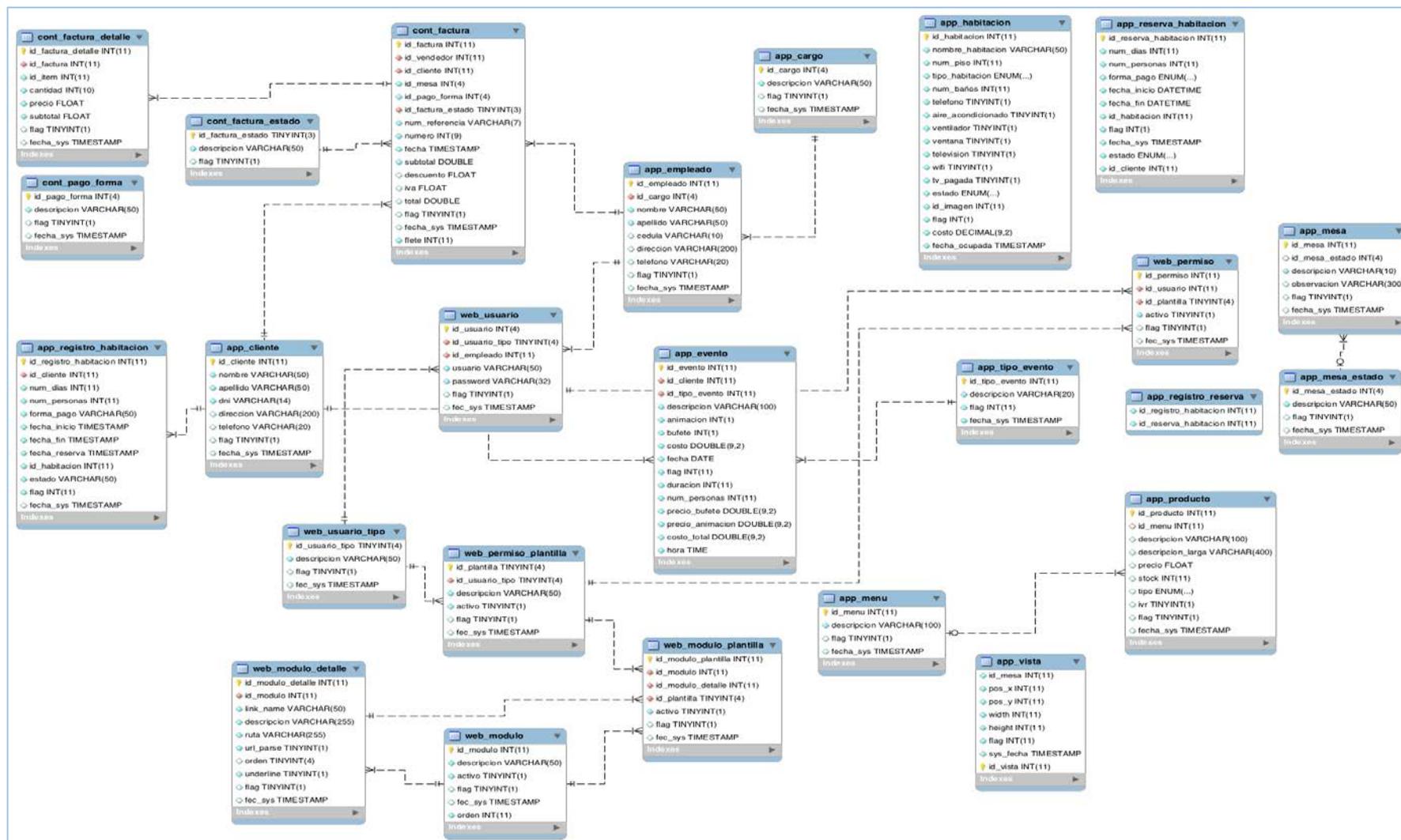


Figura 10: Diagrama Entidad Relación  
Elaborado por: Autores

## CAPITULO 5

### DESARROLLO E IMPLEMENTACION DE LA PROPUESTA

#### 5.1. Script de creación de base de datos

La Base de Datos funciona con el Motor MYSQL especializado para aplicaciones web, con gran popularidad y buen rendimiento a los requerimientos del tipo de aplicación que se ha brindado.

```
# *****  
  
# Sequel Pro SQL dump  
# Versión 4096  
#  
# http://www.sequelpro.com/  
# http://code.google.com/p/sequel-pro/  
#  
# Host: localhost (MySQL 5.5.38)  
# Base de datos: hospedaje  
# Tiempo de Generación: 2015-03-01 11:58:10 p.m. +0000  
# *****  
  
/*!40101 SET  
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;  
/*!40101 SET  
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS  
*/;  
/*!40101 SET  
@OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;  
/*!40101 SET NAMES utf8 */;  
/*!40014 SET  
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0 */;  
/*!40101 SET  
@OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
```

```

/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;

# Volcado de tabla app_cargo
# -----

DROP TABLE IF EXISTS `app_cargo`;

CREATE TABLE `app_cargo` (
  `id_cargo` int(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(50) NOT NULL DEFAULT "",
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_cargo`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `app_cargo` WRITE;
/*!40000 ALTER TABLE `app_cargo` DISABLE KEYS */;

INSERT INTO `app_cargo` (`id_cargo`, `descripcion`, `flag`, `fecha_sys`)
VALUES
  (1,'administrador',1,'2013-01-02 00:33:52'),
  (2,'mesero',1,'2013-01-02 00:33:56'),
  (3,'cajero',1,'2013-02-11 13:02:38'),
  (4,'despachador',1,'2013-09-29 03:49:21');

/*!40000 ALTER TABLE `app_cargo` ENABLE KEYS */;
UNLOCK TABLES;

```

```

# Volcado de tabla app_cliente
# -----

DROP TABLE IF EXISTS `app_cliente`;

CREATE TABLE `app_cliente` (
  `id_cliente` int(11) NOT NULL AUTO_INCREMENT,
  `nombre` varchar(50) NOT NULL DEFAULT "",
  `apellido` varchar(50) NOT NULL DEFAULT "",
  `dni` varchar(14) NOT NULL DEFAULT "",
  `direccion` varchar(200) DEFAULT NULL,
  `telefono` varchar(20) DEFAULT NULL,
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_cliente`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `app_cliente` WRITE;
/*!40000 ALTER TABLE `app_cliente` DISABLE KEYS */;

INSERT INTO `app_cliente` (`id_cliente`, `nombre`, `apellido`, `dni`, `direccion`,
`telefono`, `flag`, `fecha_sys`)
VALUES
  (1,'Johnny','Jaramillo','032568945420','El oro y
chimborazo','1213123112',1,'2013-01-21 12:20:57'),
  (2,'Tatiana','Cacinelli','092887373848','Av.
Américas','9182938392',1,'2013-01-21 12:21:23'),
  (3,'María','Alcívar','0928392019','El Oro y Chile','0922837281',1,'2013-01-
30 02:17:26'),
  (4,'Rosario','Tijeras','09776655444444','9 de Octubre y Boyacá',",1,'2013-
01-30 02:22:40'),
  (5,'Manuel','Quinteros','0928382919','Garzota II, Mz
53','042348938',1,'2013-01-30 02:23:13'),

```

```

(6,'Steffi','Zoller','0931485296','pradera      1','0982415374',1,'2014-12-15
12:09:36'),
(7,'Heinzel','Zoller','0931485288','Orquideas','2586632',1,'2015-01-05
08:58:00');

/*!40000 ALTER TABLE `app_cliente` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla app_empleado
# -----

DROP TABLE IF EXISTS `app_empleado`;

CREATE TABLE `app_empleado` (
  `id_empleado` int(11) NOT NULL AUTO_INCREMENT,
  `id_cargo` int(4) NOT NULL,
  `nombre` varchar(50) NOT NULL DEFAULT "",
  `apellido` varchar(50) NOT NULL DEFAULT "",
  `cedula` varchar(10) DEFAULT NULL,
  `direccion` varchar(200) DEFAULT NULL,
  `telefono` varchar(20) DEFAULT NULL,
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_empleado`),
  KEY `id_cargo` (`id_cargo`),
  CONSTRAINT `app_empleado_ibfk_1` FOREIGN KEY (`id_cargo`)
REFERENCES `app_cargo` (`id_cargo`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `app_empleado` WRITE;

```

```

/*!40000 ALTER TABLE `app_empleado` DISABLE KEYS */;

INSERT INTO `app_empleado` (`id_empleado`, `id_cargo`, `nombre`, `apellido`,
`cedula`, `direccion`, `telefono`, `flag`, `fecha_sys`)
VALUES
    (1,1,'Administrador','Local','','9 de Octubre y Boyacá','1,2013-01-02
00:35:48'),
    (2,2,'José','Cárdenas','0928374882','0982837483',1,2013-01-30
01:58:48'),
    (3,2,'Raúl','Cevilla','0039288382','Av. Quito y El
Oro','0982733333',1,2013-01-30 01:59:22'),
    (4,2,'Ruddy','Aragón','0928388484','0912883822',1,2013-01-30
02:01:36'),
    (5,3,'Joel','Macías','','0900012001',1,2013-02-11 13:05:04'),
    (6,4,'RONNY','Bonilla','','0019291291',1,2013-09-29 03:51:57'),
    (7,4,'Carlos','Maute','','0981827172',1,2013-09-29 03:53:22'),
    (8,4,'Roberto','Espinel','','1213131121',1,2013-09-29 03:53:29);

/*!40000 ALTER TABLE `app_empleado` ENABLE KEYS */;

UNLOCK TABLES;

# Volcado de tabla app_evento
# -----

DROP TABLE IF EXISTS `app_evento`;

CREATE TABLE `app_evento` (
    `id_evento` int(11) NOT NULL AUTO_INCREMENT,
    `id_cliente` int(11) NOT NULL,
    `id_tipo_evento` int(11) NOT NULL,
    `descripcion` varchar(100) NOT NULL,

```

```

`animacion` int(1) NOT NULL,
`bufete` int(1) NOT NULL,
`costo` double(9,2) NOT NULL,
`fecha` date NOT NULL,
`flag` int(11) NOT NULL DEFAULT '1',
`duracion` int(11) NOT NULL,
`num_personas` int(11) NOT NULL,
PRIMARY KEY (`id_evento`),
KEY `fk_evento` (`id_tipo_evento`),
CONSTRAINT `fk_evento_tipo` FOREIGN KEY (`id_tipo_evento`)
REFERENCES `app_tipo_evento` (`id_tipo_evento`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `app_evento` WRITE;
/*!40000 ALTER TABLE `app_evento` DISABLE KEYS */;

INSERT INTO `app_evento` (`id_evento`, `id_cliente`, `id_tipo_evento`,
`descripcion`, `animacion`, `bufete`, `costo`, `fecha`, `flag`, `duracion`,
`num_personas`)
VALUES
(1,4,1,'Quincianera Karlita',1,1,777.00,'2015-02-19',1,4,200),
(2,5,2,'Manuel\'s birthday',0,0,382.00,'2015-02-28',1,5,100),
(8,5,2,'seminario de contabilidad',0,1,57.75,'2015-02-12',1,2,300),
(9,7,3,'seminario de contabilidad 2',1,0,928.00,'2015-02-28',1,4,45);

/*!40000 ALTER TABLE `app_evento` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla app_habitacion
# -----

```

```

DROP TABLE IF EXISTS `app_habitacion`;

CREATE TABLE `app_habitacion` (
  `id_habitacion` int(11) NOT NULL AUTO_INCREMENT,
  `nombre_habitacion` varchar(50) NOT NULL,
  `num_piso` int(11) NOT NULL,
  `tipo_habitacion` enum('sencilla','doble','matrimonial','familiar') CHARACTER
SET utf8 NOT NULL DEFAULT 'sencilla',
  `num_baños` int(11) NOT NULL,
  `telefono` tinyint(1) NOT NULL,
  `aire_acondicionado` tinyint(1) NOT NULL,
  `ventilador` tinyint(1) NOT NULL,
  `ventana` tinyint(1) NOT NULL,
  `television` tinyint(1) NOT NULL,
  `wifi` tinyint(1) NOT NULL,
  `tv_pagada` tinyint(1) NOT NULL,
  `estado` enum('libre','ocupada','reservada') CHARACTER SET utf8 NOT NULL
DEFAULT 'libre',
  `id_imagen` int(11) NOT NULL,
  `flag` int(1) NOT NULL DEFAULT '1',
  `costo` decimal(9,2) NOT NULL,
  PRIMARY KEY (`id_habitacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `app_habitacion` WRITE;

/*!40000 ALTER TABLE `app_habitacion` DISABLE KEYS */;

INSERT INTO `app_habitacion` (`id_habitacion`, `nombre_habitacion`,
`num_piso`, `tipo_habitacion`, `num_baños`, `telefono`, `aire_acondicionado`,
`ventilador`, `ventana`, `television`, `wifi`, `tv_pagada`, `estado`, `id_imagen`,
`flag`, `costo`)
VALUES

```

```

(1,'imperial',3,'doble',2,1,1,0,0,1,1,1,'reservada',0,1,55.80),
(3,'suit',1,'doble',2,1,1,1,0,1,1,1,'ocupada',0,1,59.00),
(4,'suit junior',2,'doble',2,1,0,1,0,1,0,1,'reservada',0,1,42.00),
(5,'penthouse',7,'matrimonial',3,1,1,1,1,1,1,1,'ocupada',0,1,125.00),
(9,'suit',1,'familiar',3,0,0,1,1,1,0,0,'reservada',0,1,75.20),
(10,'Corfu Suite',4,'matrimonial',2,0,0,0,1,1,1,1,'ocupada',0,1,80.22),
(11,'presidencial',3,'sencilla',1,1,1,1,0,1,1,1,'ocupada',0,1,80.00),
(12,'suit',1,'doble',2,1,1,0,0,1,1,0,'libre',0,1,59.00),
(13,'presidencial',3,'familiar',2,1,1,1,1,1,1,1,'ocupada',0,1,100.00),
(14,'MILLENIUM',3,'doble',2,1,1,0,0,1,1,1,'ocupada',0,1,62.30);

```

```

/*!40000 ALTER TABLE `app_habitacion` ENABLE KEYS */;

```

```

UNLOCK TABLES;

```

```

# Volcado de tabla app_menu

```

```

# -----

```

```

DROP TABLE IF EXISTS `app_menu`;

```

```

CREATE TABLE `app_menu` (
  `id_menu` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(100) NOT NULL DEFAULT "",
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_menu`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

LOCK TABLES `app_menu` WRITE;

```

```

/*!40000 ALTER TABLE `app_menu` DISABLE KEYS */;

```

```

INSERT INTO `app_menu` (`id_menu`, `descripcion`, `flag`, `fecha_sys`)
VALUES
    (1,'BEBIDAS',1,'2013-01-02 00:36:47'),
    (2,'ENTRADAS',1,'2013-01-02 00:37:08'),
    (3,'PLATOS FUERTES',1,'2013-01-02 00:37:14'),
    (4,'SOPAS',1,'2013-01-02 00:37:19'),
    (5,'POSTRES',1,'2013-01-02 00:37:26'),
    (6,'ESPECIALES',1,'2013-01-02 00:37:33'),
    (7,'ENSALADAS',1,'2013-01-02 00:37:51'),
    (8,'SERVICIOS',1,'2014-12-30 16:00:50');

/*!40000 ALTER TABLE `app_menu` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla app_mesa
# -----

DROP TABLE IF EXISTS `app_mesa`;

CREATE TABLE `app_mesa` (
  `id_mesa` int(11) NOT NULL AUTO_INCREMENT,
  `id_mesa_estado` int(4) DEFAULT '1',
  `descripcion` varchar(10) NOT NULL DEFAULT "",
  `observacion` varchar(300) DEFAULT NULL,
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_mesa`),
  KEY `id_mesa_estado` (`id_mesa_estado`),

```

```

CONSTRAINT `app_mesa_ibfk_1` FOREIGN KEY (`id_mesa_estado`)
REFERENCES `app_mesa_estado` (`id_mesa_estado`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `app_mesa` WRITE;

/*!40000 ALTER TABLE `app_mesa` DISABLE KEYS */;

INSERT INTO `app_mesa` (`id_mesa`, `id_mesa_estado`, `descripcion`,
`observacion`, `flag`, `fecha_sys`)
VALUES
    (1,2,'Mesa 1',"1,2013-01-02 00:12:20'),
    (2,2,'Mesa 2',"1,2013-01-02 00:12:25'),
    (3,2,'Mesa 3',"1,2013-01-02 00:12:29'),
    (4,2,'Mesa 4',"1,2013-01-02 00:12:32'),
    (5,2,'Mesa 5',"1,2013-01-02 00:12:36'),
    (6,2,'Mesa 6',"1,2013-01-02 00:12:40'),
    (7,1,'Mesa VIP-1',"1,2013-01-02 00:12:47'),
    (8,2,'Mesa VIP-2',"1,2013-01-02 00:13:06'),
    (9,1,'Mesa Alfa',"1,2013-01-21 11:54:09');

/*!40000 ALTER TABLE `app_mesa` ENABLE KEYS */;

UNLOCK TABLES;

# Volcado de tabla app_mesa_estado
# -----

DROP TABLE IF EXISTS `app_mesa_estado`;

CREATE TABLE `app_mesa_estado` (
    `id_mesa_estado` int(4) NOT NULL AUTO_INCREMENT,

```

```

`descripcion` varchar(50) NOT NULL DEFAULT "",
`flag` tinyint(1) DEFAULT '1',
`fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (`id_mesa_estado`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `app_mesa_estado` WRITE;
/*!40000 ALTER TABLE `app_mesa_estado` DISABLE KEYS */;

INSERT INTO `app_mesa_estado` (`id_mesa_estado`, `descripcion`, `flag`,
`fecha_sys`)
VALUES
    (1,'LIBRE',1,'2013-01-02 00:11:33'),
    (2,'OCUPADA',1,'2013-01-02 00:11:38'),
    (3,'RESERVADA',1,'2013-01-02 00:11:42');

/*!40000 ALTER TABLE `app_mesa_estado` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla app_producto
# -----

DROP TABLE IF EXISTS `app_producto`;

CREATE TABLE `app_producto` (
    `id_producto` int(11) NOT NULL AUTO_INCREMENT,
    `id_menu` int(11) DEFAULT NULL,
    `descripcion` varchar(100) DEFAULT NULL,
    `descripcion_larga` varchar(400) DEFAULT NULL,
    `precio` float DEFAULT NULL,

```

```

`stock` int(11) DEFAULT NULL,
`tipo` enum('solido','liquido') DEFAULT 'solido',
`ivr` tinyint(1) DEFAULT '0',
`flag` tinyint(1) DEFAULT '1',
`fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (`id_producto`),
KEY `id_menu` (`id_menu`),
CONSTRAINT `app_producto_ibfk_1` FOREIGN KEY (`id_menu`)
REFERENCES `app_menu` (`id_menu`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `app_producto` WRITE;
/*!40000 ALTER TABLE `app_producto` DISABLE KEYS */;

INSERT INTO `app_producto` (`id_producto`, `id_menu`, `descripcion`,
`descripcion_larga`, `precio`, `stock`, `tipo`, `ivr`, `flag`, `fecha_sys`)
VALUES
(2,1,'cocacola','bebida negra',1,200,'liquido',1,1,'2013-01-02 18:44:37'),
(3,1,'inca cola','soda',1,200,'liquido',1,1,'2013-01-02 18:48:31'),
(4,3,'Chuleta con salsa','Chuleta aderezada con salsa
bbq',8.5,12,'solido',1,1,'2013-01-02 18:51:17'),
(5,3,'Pollo al blue','pollo relleno con queso y
mortadela',5.5,0,'solido',0,1,'2013-01-02 18:51:51'),
(6,5,'Helado Óreo','Helado sabor Óreo',1.5,0,'solido',1,1,'2013-01-02
18:52:21'),
(7,6,'Langostino','langosta con salsa marinera',22,0,'solido',1,1,'2013-01-02
18:53:17'),
(8,4,'Sopa Marinera','pescado, camarón',6.8,12,'solido',1,1,'2013-01-02
18:56:13'),
(9,5,'Dulce tres leches','',1.5,39,'solido',1,1,'2013-01-02 18:56:36'),
(10,1,'Agua','Agua',1,1000,'liquido',1,1,'2013-09-29 03:57:39'),
(11,1,'Gelatina','Gelatina',1,1000,'liquido',0,1,'2013-09-29 03:58:05'),
(12,1,'Helado','Helado',1.5,1000,'liquido',0,1,'2013-09-29 03:58:21'),

```

```

(13,1,'Yogurt','Yogurt',1.4,1000,'liquido',0,1,'2013-09-29 03:58:26'),
(14,8,'SERVICIOS','servicios',NULL,NULL,'solido',0,1,'2014-12-30
16:01:57'),
(15,8,'Servicio Alquiler de Instalaciones por evento','Servicio Alquiler de
Instalaciones por evento',NULL,NULL,'solido',0,1,'2015-02-24 15:36:52');

/*!40000 ALTER TABLE `app_producto` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla app_registro_habitacion
# -----

DROP TABLE IF EXISTS `app_registro_habitacion`;

CREATE TABLE `app_registro_habitacion` (
  `id_registro_habitacion` int(11) NOT NULL AUTO_INCREMENT,
  `id_cliente` int(11) NOT NULL,
  `num_dias` int(11) NOT NULL,
  `num_personas` int(11) NOT NULL,
  `forma_pago` varchar(50) NOT NULL,
  `fecha_inicio` date NOT NULL,
  `fecha_fin` date NOT NULL,
  `fecha_reserva` date NOT NULL,
  `id_habitacion` int(11) NOT NULL,
  `estado` varchar(50) NOT NULL,
  `flag` int(11) NOT NULL DEFAULT '1',
  `fecha_sys` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_registro_habitacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

LOCK TABLES `app_registro_habitacion` WRITE;

/*!40000 ALTER TABLE `app_registro_habitacion` DISABLE KEYS */;

INSERT INTO `app_registro_habitacion` (`id_registro_habitacion`, `id_cliente`,
`num_dias`, `num_personas`, `forma_pago`, `fecha_inicio`, `fecha_fin`,
`fecha_reserva`, `id_habitacion`, `estado`, `flag`, `fecha_sys`)
VALUES
(15,6,1,2,'efectivo','2014-12-30','2014-12-31','2014-12-
22',10,'Despachada',1,'2014-12-30 13:15:50'),
(16,2,2,1,'tarjeta de credito','2014-12-21','2014-12-22','2014-12-
20',11,'Despachada',1,'2014-12-31 09:17:24'),
(17,7,2,2,'efectivo','2015-01-06','2015-01-08','2015-01-
06',5,'Despachada',1,'2015-01-05 10:07:36'),
(22,6,5,2,'tarjeta de credito','2015-01-08','2015-01-13','0000-00-
00',3,'Despachada',1,'2015-01-05 12:12:28'),
(23,5,8,2,'efectivo','2015-01-07','2015-01-15','2015-01-
05',14,'Despachada',1,'2015-01-05 12:16:35'),
(24,2,5,2,'efectivo','2015-01-07','2015-01-12','2015-01-
09',13,'Despachada',1,'2015-01-05 12:35:06');

/*!40000 ALTER TABLE `app_registro_habitacion` ENABLE KEYS */;

UNLOCK TABLES;

# Volcado de tabla app_registro_reserva
# -----

DROP TABLE IF EXISTS `app_registro_reserva`;

CREATE TABLE `app_registro_reserva` (
`id_registro_habitacion` int(11) NOT NULL,
`id_reserva_habitacion` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

```

```

LOCK TABLES `app_registro_reserva` WRITE;

/*!40000 ALTER TABLE `app_registro_reserva` DISABLE KEYS */;

INSERT INTO `app_registro_reserva` (`id_registro_habitacion`,
`id_reserva_habitacion`)
VALUES
    (0,0),
    (15,29),
    (16,30),
    (17,32),
    (0,0),
    (0,0),
    (0,0),
    (21,0),
    (22,33),
    (23,34),
    (24,35);

/*!40000 ALTER TABLE `app_registro_reserva` ENABLE KEYS */;

UNLOCK TABLES;

# Volcado de tabla app_reserva_habitacion
# -----

DROP TABLE IF EXISTS `app_reserva_habitacion`;

CREATE TABLE `app_reserva_habitacion` (
    `id_reserva_habitacion` int(11) NOT NULL AUTO_INCREMENT,
    `num_dias` int(11) NOT NULL,

```

```

`num_personas` int(11) NOT NULL,
`forma_pago` enum('tarjeta de credito','efectivo') NOT NULL DEFAULT
'efectivo',
`fecha_inicio` datetime NOT NULL,
`fecha_fin` datetime NOT NULL,
`id_habitacion` int(11) NOT NULL,
`flag` int(1) NOT NULL DEFAULT '1',
`fecha_sys` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
`estado` enum('Creada','Despachada','Anulada') NOT NULL DEFAULT 'Creada',
`id_cliente` int(11) NOT NULL,
PRIMARY KEY (`id_reserva_habitacion`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `app_reserva_habitacion` WRITE;
/*!40000 ALTER TABLE `app_reserva_habitacion` DISABLE KEYS */;

INSERT INTO `app_reserva_habitacion` (`id_reserva_habitacion`, `num_dias`,
`num_personas`, `forma_pago`, `fecha_inicio`, `fecha_fin`, `id_habitacion`, `flag`,
`fecha_sys`, `estado`, `id_cliente`)
VALUES
(29,1,2,'efectivo','2014-12-30 00:00:00','2014-12-31 00:00:00',10,1,'2014-
12-30 13:15:11','Despachada',6),
(30,2,1,'tarjeta de credito','2014-12-21 00:00:00','2014-12-22
00:00:00',1,1,'2014-12-30 13:17:55','Despachada',2),
(31,2,2,'tarjeta de credito','2014-12-22 00:00:00','2014-12-24
00:00:00',9,1,'2014-12-30 14:49:05','Creada',6),
(32,2,2,'tarjeta de credito','2015-01-06 00:00:00','2015-01-08
00:00:00',5,1,'2015-01-05 08:58:41','Despachada',7),
(33,5,2,'tarjeta de credito','2015-01-08 00:00:00','2015-01-13
00:00:00',3,1,'2015-01-05 11:48:16','Despachada',6),
(34,8,2,'efectivo','2015-01-07 00:00:00','2015-01-15 00:00:00',14,1,'2015-
01-05 12:16:06','Despachada',5),
(35,5,2,'efectivo','2015-01-07 00:00:00','2015-01-12 00:00:00',13,1,'2015-
01-05 12:34:02','Despachada',2),

```

```

(36,4,2,'efectivo','2015-03-02 00:00:00','2015-03-03 00:00:00',4,1,'2015-
03-01 10:48:54','Creada',7);

/*!40000 ALTER TABLE `app_reserva_habitacion` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla app_tipo_evento
# -----

DROP TABLE IF EXISTS `app_tipo_evento`;

CREATE TABLE `app_tipo_evento` (
  `id_tipo_evento` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(20) NOT NULL,
  `flag` int(11) NOT NULL DEFAULT '1',
  `fecha_sys` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_tipo_evento`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `app_tipo_evento` WRITE;
/*!40000 ALTER TABLE `app_tipo_evento` DISABLE KEYS */;

INSERT INTO `app_tipo_evento` (`id_tipo_evento`, `descripcion`, `flag`,
`fecha_sys`)
VALUES
  (1,'Quincianera',1,'2015-02-24 15:11:50'),
  (2,'Fiesta',1,'2015-02-24 15:11:50'),
  (3,'Seminario',1,'2015-02-24 15:11:50');

/*!40000 ALTER TABLE `app_tipo_evento` ENABLE KEYS */;

```

```

UNLOCK TABLES;

# Volcado de tabla cont_factura
# -----

DROP TABLE IF EXISTS `cont_factura`;

CREATE TABLE `cont_factura` (
  `id_factura` int(11) NOT NULL AUTO_INCREMENT,
  `id_vendedor` int(11) NOT NULL COMMENT 'ID de Empleado',
  `id_cliente` int(11) NOT NULL,
  `id_mesa` int(4) NOT NULL,
  `id_pago_forma` int(4) NOT NULL,
  `id_factura_estado` tinyint(3) NOT NULL,
  `num_referencia` varchar(7) NOT NULL DEFAULT '000-000',
  `numero` int(9) NOT NULL DEFAULT '0',
  `fecha` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `subtotal` double NOT NULL DEFAULT '0',
  `descuento` float DEFAULT '0',
  `iva` float DEFAULT '0',
  `total` double DEFAULT NULL,
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_factura`),
  KEY `id_cliente` (`id_cliente`),
  KEY `id_pago_forma` (`id_pago_forma`),
  KEY `id_factura_estado` (`id_factura_estado`),
  KEY `id_vendedor` (`id_vendedor`),
  KEY `id_mesa` (`id_mesa`),

```

```

CONSTRAINT `cont_factura_ibfk_4` FOREIGN KEY (`id_factura_estado`)
REFERENCES `cont_factura_estado` (`id_factura_estado`) ON UPDATE
CASCADE,

CONSTRAINT `cont_factura_ibfk_5` FOREIGN KEY (`id_vendedor`)
REFERENCES `app_empleado` (`id_empleado`) ON UPDATE CASCADE,

CONSTRAINT `cont_factura_ibfk_6` FOREIGN KEY (`id_mesa`)
REFERENCES `app_mesa` (`id_mesa`) ON UPDATE CASCADE,

CONSTRAINT `cont_factura_ibfk_7` FOREIGN KEY (`id_cliente`)
REFERENCES `app_cliente` (`id_cliente`) ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `cont_factura` WRITE;

/*!40000 ALTER TABLE `cont_factura` DISABLE KEYS */;

INSERT INTO `cont_factura` (`id_factura`, `id_vendedor`, `id_cliente`, `id_mesa`,
`id_pago_forma`, `id_factura_estado`, `num_referencia`, `numero`, `fecha`,
`subtotal`, `descuento`, `iva`, `total`, `flag`, `fecha_sys`)
VALUES
(7,1,1,1,1,1,'001-001',1,'2013-01-21 19:00:00',12.3,0,1.48,13.78,1,'2013-
01-21 13:28:12'),
(8,1,3,6,1,1,'001-001',2,'2013-01-29 19:00:00',19,0,2.28,21.28,1,'2013-01-
30 02:45:29'),
(9,1,2,2,1,1,'001-001',3,'2013-02-10 19:00:00',12,0,1.44,13.44,1,'2013-02-
11 11:57:10'),
(10,1,4,1,1,1,'001-001',4,'2013-02-10 19:00:00',4.5,0,0.54,5.04,1,'2013-02-
11 11:59:26'),
(11,1,3,2,1,1,'001-001',5,'2013-02-10 19:00:00',22,0,2.64,24.64,1,'2013-02-
11 12:02:06'),
(12,1,2,6,1,1,'001-001',6,'2013-02-10 19:00:00',15,1.3,1.64,15.34,1,'2013-
02-11 12:05:38'),
(13,1,1,3,1,1,'001-001',7,'2013-02-11 19:00:00',22,0,2.64,24.64,1,'2013-02-
11 16:23:49'),
(14,1,7,4,1,1,'001-001',8,'2015-01-07
00:00:00',251.5,0,30.18,281.68,1,'2015-01-07 11:06:18');

/*!40000 ALTER TABLE `cont_factura` ENABLE KEYS */;

```

```

UNLOCK TABLES;

# Volcado de tabla cont_factura_detalle
# -----

DROP TABLE IF EXISTS `cont_factura_detalle`;

CREATE TABLE `cont_factura_detalle` (
  `id_factura_detalle` int(11) NOT NULL AUTO_INCREMENT,
  `id_factura` int(11) NOT NULL,
  `id_item` int(11) NOT NULL,
  `cantidad` int(10) NOT NULL DEFAULT '1',
  `precio` float NOT NULL,
  `subtotal` float NOT NULL,
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_factura_detalle`),
  KEY `id_factura` (`id_factura`),
  KEY `id_item` (`id_item`),
  CONSTRAINT `cont_factura_detalle_ibfk_1` FOREIGN KEY (`id_factura`)
REFERENCES `cont_factura` (`id_factura`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `cont_factura_detalle` WRITE;
/*!40000 ALTER TABLE `cont_factura_detalle` DISABLE KEYS */;

INSERT INTO `cont_factura_detalle` (`id_factura_detalle`, `id_factura`, `id_item`,
`cantidad`, `precio`, `subtotal`, `flag`, `fecha_sys`)
VALUES
(1,7,6,1,1.5,1.5,1,'2013-01-21 13:28:12'),

```

```

(2,7,8,1,6.8,6.8,1,'2013-01-21 13:28:12'),
(3,7,2,4,1,4,1,'2013-01-21 13:28:12'),
(4,8,4,2,8.5,17,0,'2013-01-30 02:45:29'),
(5,8,3,2,1,2,0,'2013-01-30 02:45:29'),
(6,8,4,2,8.5,17,1,'2013-01-30 02:54:39'),
(7,8,3,2,1,2,1,'2013-01-30 02:54:39'),
(8,9,5,1,5.5,5.5,1,'2013-02-11 11:57:10'),
(9,9,9,1,1.5,1.5,1,'2013-02-11 11:57:10'),
(10,9,2,5,1,5,1,'2013-02-11 11:57:10'),
(11,10,9,3,1.5,4.5,1,'2013-02-11 11:59:26'),
(12,11,7,1,22,22,1,'2013-02-11 12:02:06'),
(13,12,6,10,1.5,15,1,'2013-02-11 12:05:38'),
(14,13,7,1,22,22,1,'2013-02-11 16:23:49'),
(15,14,9,1,1.5,1.5,1,'2015-01-07 11:06:18'),
(16,14,14,2,125,250,1,'2015-01-07 11:06:18');

```

```

/*!40000 ALTER TABLE `cont_factura_detalle` ENABLE KEYS */;

```

```

UNLOCK TABLES;

```

```

# Volcado de tabla cont_factura_estado

```

```

# -----

```

```

DROP TABLE IF EXISTS `cont_factura_estado`;

```

```

CREATE TABLE `cont_factura_estado` (
  `id_factura_estado` tinyint(3) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(50) CHARACTER SET latin1 COLLATE latin1_spanish_ci
  NOT NULL,
  `flag` tinyint(1) DEFAULT '1',
  PRIMARY KEY (`id_factura_estado`)

```

```

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

LOCK TABLES `cont_factura_estado` WRITE;

/*!40000 ALTER TABLE `cont_factura_estado` DISABLE KEYS */;

INSERT INTO `cont_factura_estado` (`id_factura_estado`, `descripcion`, `flag`)
VALUES
    (1,'emitida',1),
    (2,'cancelada',1),
    (3,'anulada',1),
    (4,'devolucion',1),
    (6,'pago parcial',1),
    (7,'PEDIDO',1);

/*!40000 ALTER TABLE `cont_factura_estado` ENABLE KEYS */;

UNLOCK TABLES;

# Volcado de tabla cont_pago_forma
# -----

DROP TABLE IF EXISTS `cont_pago_forma`;

CREATE TABLE `cont_pago_forma` (
  `id_pago_forma` int(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(50) NOT NULL DEFAULT "",
  `flag` tinyint(1) DEFAULT '1',
  `fecha_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_pago_forma`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

LOCK TABLES `cont_pago_forma` WRITE;

/*!40000 ALTER TABLE `cont_pago_forma` DISABLE KEYS */;

INSERT INTO `cont_pago_forma` (`id_pago_forma`, `descripcion`, `flag`,
`fecha_sys`)
VALUES
    (1,'EFECTIVO',1,'2013-01-02 00:21:22'),
    (2,'TARJETA CREDITO',1,'2013-01-02 00:21:29');

/*!40000 ALTER TABLE `cont_pago_forma` ENABLE KEYS */;

UNLOCK TABLES;

# Volcado de tabla web_modulo
# -----

DROP TABLE IF EXISTS `web_modulo`;

CREATE TABLE `web_modulo` (
  `id_modulo` int(11) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
  `activo` tinyint(1) NOT NULL DEFAULT '1',
  `flag` tinyint(1) DEFAULT '1',
  `fec_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_modulo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

LOCK TABLES `web_modulo` WRITE;

/*!40000 ALTER TABLE `web_modulo` DISABLE KEYS */;

```

```

INSERT INTO `web_modulo` (`id_modulo`, `descripcion`, `activo`, `flag`,
`fec_sys`)
VALUES
    (1,'Restaurant',1,1,'2012-11-21 23:00:48'),
    (2,'Hospedaje',1,1,'2014-12-15 11:13:25'),
    (3,'Ventas',1,1,'2012-11-14 11:56:16'),
    (4,'Clientes',1,1,'2014-12-15 11:00:44'),
    (5,'Otros',1,1,'2012-11-14 11:56:27'),
    (6,'Personal',1,1,'2013-01-30 01:48:35'),
    (7,'Mesas',1,1,'2013-01-17 23:55:57'),
    (8,'Eventos',1,1,'2015-02-24 10:28:18');

/*!40000 ALTER TABLE `web_modulo` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla web_modulo_detalle
# -----

DROP TABLE IF EXISTS `web_modulo_detalle`;

CREATE TABLE `web_modulo_detalle` (
  `id_modulo_detalle` int(11) NOT NULL AUTO_INCREMENT,
  `id_modulo` int(11) NOT NULL,
  `link_name` varchar(50) COLLATE latin1_spanish_ci NOT NULL,
  `descripcion` varchar(255) COLLATE latin1_spanish_ci NOT NULL,
  `ruta` varchar(255) COLLATE latin1_spanish_ci NOT NULL,
  `url_parse` tinyint(1) NOT NULL DEFAULT '0',
  `orden` tinyint(4) DEFAULT NULL,
  `underline` tinyint(1) NOT NULL DEFAULT '0',
  `flag` tinyint(1) DEFAULT '1',

```

```

`fec_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
PRIMARY KEY (`id_modulo_detalle`),
KEY `id_modulo` (`id_modulo`),
CONSTRAINT `web_modulo_detalle_ibfk_1` FOREIGN KEY (`id_modulo`)
REFERENCES `web_modulo` (`id_modulo`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

LOCK TABLES `web_modulo_detalle` WRITE;

/*!40000 ALTER TABLE `web_modulo_detalle` DISABLE KEYS */;

INSERT INTO `web_modulo_detalle` (`id_modulo_detalle`, `id_modulo`,
`link_name`, `descripcion`, `ruta`, `url_parse`, `orden`, `underline`, `flag`,
`fec_sys`)
VALUES
(21,1,'Listado
Productos','Listado','modulos/inventario/index.php',1,2,0,1,'2013-01-02 18:31:13'),
(22,1,'Nuevo
Producto','Nuevo
Producto','modulos/inventario/productos.php',0,1,0,1,'2013-01-02 18:31:13'),
(23,3,'Facturas','Facturas','modulos/ventas/facturas.php',0,1,0,1,'2013-01-02
18:32:44'),
(24,3,'Nueva
Factura','Nueva
Factura','modulos/ventas/facturas.php?ID=',1,2,0,1,'2013-01-02 18:32:44'),
(25,1,'Listado
Mesas','Listado
Mesas','modulos/servicio/mesas.php',0,4,0,1,'2013-01-17 23:57:30'),
(26,1,'Nueva
Mesa','Nueva
Mesa','modulos/servicio/mesas.php?ID=',0,3,0,1,'2013-01-17 23:58:40'),
(27,6,'Meseros','Meseros','modulos/empresa/meseros.php',0,1,0,1,'2013-01-
30 01:49:33'),
(28,6,'Clientes','Clientes','modulos/empresa/clientes.php',0,2,0,0,'2013-01-
30 01:49:52'),
(29,3,'Pedidos','Pedidos','modulos/ventas/pedidos.php',0,3,0,1,'2013-01-30
02:38:10'),
(30,6,'Cajeros','Cajeros','modulos/empresa/cajeros.php',0,3,0,1,'2013-02-11
12:58:02'),

```

```

(31,6,'Despachadores','Despachadores','modulos/empresa/despachadores.php',0,4,0,0,'2013-09-29 03:48:14'),
(34,4,'Nuevo Cliente','Nuevo Cliente','modulos/clientes/nuevo_cliente.php',0,NULL,0,1,'2014-12-15 11:06:47'),
(35,4,'Listado Clientes','Listado Clientes','modulos/clientes/clientes.php',0,NULL,0,1,'2014-12-15 11:06:47'),
(36,2,'Nueva Habitacion','Nueva Habitacion','modulos/hospedaje/habitacion.php?ID=',0,1,0,1,'2014-12-15 11:19:19'),
(37,2,'Reservar Habitacion','Reservar Habitacion','modulos/hospedaje/reserva_habitacion.php?ID=&rad=reserva',0,3,0,1,'2014-12-15 11:19:19'),
(40,2,'Servicios','Servicios','modulos/hospedaje/servicios.php',0,7,0,1,'2014-12-15 11:30:25'),
(41,2,'Listado Habitaciones','Listado Habitaciones','modulos/hospedaje/habitacion.php',0,2,0,1,'2014-12-16 10:57:12'),
(43,2,'Reservaciones','Reservaciones','modulos/hospedaje/reserva_habitacion.php',0,4,0,1,'2014-12-18 16:46:11'),
(44,2,'Registrar Huesped','Registrar Huesped','modulos/hospedaje/reserva_habitacion.php?ID=&rad=registro',0,5,0,1,'2014-12-23 11:32:22'),
(45,2,'Huespedes Registrados','Huespedes Registrados','modulos/hospedaje/reserva_huesped.php',0,6,0,1,'2014-12-30 15:11:54'),
(46,8,'Nuevo Evento','Nuevo Evento','modulos/eventos/evento.php?ID=',0,1,0,1,'2015-02-24 10:30:02'),
(47,8,'Listado de eventos','Listado de eventos','modulos/eventos/evento.php',0,2,0,1,'2015-02-24 10:34:05');

/*!40000 ALTER TABLE `web_modulo_detalle` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla web_modulo_plantilla
# -----

```

```

DROP TABLE IF EXISTS `web_modulo_plantilla`;

CREATE TABLE `web_modulo_plantilla` (
  `id_modulo_plantilla` int(11) NOT NULL AUTO_INCREMENT,
  `id_modulo` int(11) NOT NULL,
  `id_modulo_detalle` int(11) NOT NULL,
  `id_plantilla` tinyint(4) NOT NULL,
  `activo` tinyint(1) NOT NULL DEFAULT '1',
  `flag` tinyint(1) DEFAULT '1',
  `fec_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_modulo_plantilla`),
  KEY `id_modulo` (`id_modulo`),
  KEY `id_modulo_detalle` (`id_modulo_detalle`),
  KEY `id_plantilla` (`id_plantilla`),
  CONSTRAINT `web_modulo_plantilla_ibfk_1` FOREIGN KEY (`id_modulo`)
REFERENCES `web_modulo` (`id_modulo`) ON UPDATE CASCADE,
  CONSTRAINT `web_modulo_plantilla_ibfk_2` FOREIGN KEY
(`id_modulo_detalle`) REFERENCES `web_modulo_detalle`
(`id_modulo_detalle`) ON UPDATE CASCADE,
  CONSTRAINT `web_modulo_plantilla_ibfk_3` FOREIGN KEY (`id_plantilla`)
REFERENCES `web_permiso_plantilla` (`id_plantilla`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

LOCK TABLES `web_modulo_plantilla` WRITE;

/*140000 ALTER TABLE `web_modulo_plantilla` DISABLE KEYS */;

INSERT INTO `web_modulo_plantilla` (`id_modulo_plantilla`, `id_modulo`,
`id_modulo_detalle`, `id_plantilla`, `activo`, `flag`, `fec_sys`)
VALUES
  (21,1,21,1,1,1,'2013-01-02 18:34:14'),
  (22,1,22,1,1,1,'2013-01-02 18:34:14'),
  (23,3,23,1,1,1,'2013-01-02 18:34:14'),

```

```
(24,3,24,1,1,1,'2013-01-02 18:34:14'),  
(28,1,25,1,1,1,'2013-01-17 23:59:05'),  
(29,1,26,1,1,1,'2013-01-17 23:59:12'),  
(30,6,27,1,1,1,'2013-01-30 01:50:12'),  
(31,6,28,1,0,1,'2013-01-30 01:50:20'),  
(32,3,29,1,1,1,'2013-01-30 02:38:22'),  
(33,6,30,1,1,1,'2013-02-11 12:58:22'),  
(36,1,21,2,1,1,'2013-02-11 13:26:20'),  
(37,1,25,2,1,1,'2013-02-11 13:26:28'),  
(38,3,23,2,1,1,'2013-02-11 13:26:54'),  
(39,3,24,2,1,1,'2013-02-11 13:27:01'),  
(40,3,29,2,1,1,'2013-02-11 13:27:13'),  
(41,6,31,1,0,1,'2013-09-29 03:48:25'),  
(42,4,34,1,1,1,'2014-12-15 11:09:40'),  
(43,4,35,1,1,1,'2014-12-15 11:09:40'),  
(46,2,36,1,1,1,'2014-12-15 11:25:36'),  
(47,2,37,1,1,1,'2014-12-15 11:25:40'),  
(48,2,40,1,1,1,'2014-12-15 11:32:08'),  
(49,2,41,1,1,1,'2014-12-16 10:59:45'),  
(51,2,43,1,1,1,'2014-12-18 16:47:20'),  
(52,2,44,1,1,1,'2014-12-23 11:33:50'),  
(53,2,45,1,1,1,'2014-12-30 15:13:45'),  
(54,8,46,1,1,1,'2015-02-24 10:34:51'),  
(56,8,47,1,1,1,'2015-02-24 10:35:25');
```

```
/*!40000 ALTER TABLE `web_modulo_plantilla` ENABLE KEYS */;
```

```
UNLOCK TABLES;
```

```
# Volcado de tabla web_permiso
```

```

# -----

DROP TABLE IF EXISTS `web_permiso`;

CREATE TABLE `web_permiso` (
  `id_permiso` int(11) NOT NULL AUTO_INCREMENT,
  `id_usuario` int(11) NOT NULL,
  `id_plantilla` tinyint(4) NOT NULL,
  `activo` tinyint(1) NOT NULL DEFAULT '1',
  `flag` tinyint(1) DEFAULT '1',
  `fec_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_permiso`),
  KEY `id_usuario` (`id_usuario`),
  KEY `id_plantilla` (`id_plantilla`),
  CONSTRAINT `web_permiso_ibfk_1` FOREIGN KEY (`id_usuario`)
REFERENCES `web_usuario` (`id_usuario`) ON UPDATE CASCADE,
  CONSTRAINT `web_permiso_ibfk_2` FOREIGN KEY (`id_plantilla`)
REFERENCES `web_permiso_plantilla` (`id_plantilla`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COLLATE=latin1_spanish_ci;

LOCK TABLES `web_permiso` WRITE;

/*!40000 ALTER TABLE `web_permiso` DISABLE KEYS */;

INSERT INTO `web_permiso` (`id_permiso`, `id_usuario`, `id_plantilla`, `activo`,
`flag`, `fec_sys`)
VALUES
  (1,1,1,1,1,'2012-11-14 12:25:49'),
  (2,2,2,1,1,'2013-02-11 14:12:31');

/*!40000 ALTER TABLE `web_permiso` ENABLE KEYS */;

UNLOCK TABLES;

```

```

# Volcado de tabla web_permiso_plantilla
# -----

DROP TABLE IF EXISTS `web_permiso_plantilla`;

CREATE TABLE `web_permiso_plantilla` (
  `id_plantilla` tinyint(4) NOT NULL AUTO_INCREMENT,
  `id_usuario_tipo` tinyint(4) NOT NULL,
  `descripcion` varchar(50) NOT NULL,
  `activo` tinyint(1) NOT NULL DEFAULT '1',
  `flag` tinyint(1) DEFAULT '1',
  `fec_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_plantilla`),
  KEY `id_usuario_tipo` (`id_usuario_tipo`),
  CONSTRAINT `web_permiso_plantilla_ibfk_1` FOREIGN KEY
(`id_usuario_tipo`) REFERENCES `web_usuario_tipo` (`id_usuario_tipo`) ON
UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `web_permiso_plantilla` WRITE;

/*!40000 ALTER TABLE `web_permiso_plantilla` DISABLE KEYS */;

INSERT INTO `web_permiso_plantilla` (`id_plantilla`, `id_usuario_tipo`,
`descripcion`, `activo`, `flag`, `fec_sys`)
VALUES
  (1,1,'Administrador',1,1,'2012-11-14 12:25:31'),
  (2,2,'Mesero',1,1,'2012-11-14 12:25:38'),
  (4,3,'Cajero',1,1,'2013-02-11 14:16:15');

```

```

/*!40000 ALTER TABLE `web_permiso_plantilla` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla web_usuario
# -----

DROP TABLE IF EXISTS `web_usuario`;

CREATE TABLE `web_usuario` (
  `id_usuario` int(4) NOT NULL AUTO_INCREMENT,
  `id_usuario_tipo` tinyint(4) NOT NULL,
  `id_empleado` int(11) NOT NULL,
  `usuario` varchar(50) NOT NULL,
  `password` varchar(32) NOT NULL,
  `flag` tinyint(1) DEFAULT '1',
  `fec_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_usuario`),
  KEY `id_usuario_tipo` (`id_usuario_tipo`),
  KEY `id_empleado` (`id_empleado`),
  CONSTRAINT `web_usuario_ibfk_1` FOREIGN KEY (`id_usuario_tipo`)
REFERENCES `web_usuario_tipo` (`id_usuario_tipo`) ON UPDATE CASCADE,
  CONSTRAINT `web_usuario_ibfk_2` FOREIGN KEY (`id_empleado`)
REFERENCES `app_empleado` (`id_empleado`) ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `web_usuario` WRITE;
/*!40000 ALTER TABLE `web_usuario` DISABLE KEYS */;

INSERT INTO `web_usuario` (`id_usuario`, `id_usuario_tipo`, `id_empleado`,
`usuario`, `password`, `flag`, `fec_sys`)

```

```

VALUES
    (1,1,1,'admin','21232f297a57a5a743894a0e4a801fc3',1,'2012-11-13
15:36:26'),
    (2,2,2,'mesero','21232f297a57a5a743894a0e4a801fc3',1,'2013-02-11
13:28:30'),
    (3,3,5,'cajero','21232f297a57a5a743894a0e4a801fc3',1,'2013-02-11
13:28:47'),
    (4,4,6,'despacho','21232f297a57a5a743894a0e4a801fc3',1,'2014-03-30
08:17:33');

/*!40000 ALTER TABLE `web_usuario` ENABLE KEYS */;
UNLOCK TABLES;

# Volcado de tabla web_usuario_tipo
# -----

DROP TABLE IF EXISTS `web_usuario_tipo`;

CREATE TABLE `web_usuario_tipo` (
  `id_usuario_tipo` tinyint(4) NOT NULL AUTO_INCREMENT,
  `descripcion` varchar(50) NOT NULL,
  `flag` tinyint(1) DEFAULT '1',
  `fec_sys` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_usuario_tipo`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

LOCK TABLES `web_usuario_tipo` WRITE;
/*!40000 ALTER TABLE `web_usuario_tipo` DISABLE KEYS */;

INSERT INTO `web_usuario_tipo` (`id_usuario_tipo`, `descripcion`, `flag`,
`fec_sys`)

```

```

VALUES
    (1,'administrador',1,'2012-11-13 15:35:52'),
    (2,'mesero',1,'2012-11-13 15:35:58'),
    (3,'cajero',1,'2013-02-11 13:28:06'),
    (4,'despachador',1,'2014-03-30 08:14:32');

/*!40000 ALTER TABLE `web_usuario_tipo` ENABLE KEYS */;
UNLOCK TABLES;

--
-- Dumping routines (FUNCTION) for database 'hospedaje'
--
DELIMITER ;;

# Dump of FUNCTION ADDCERO
# -----

/*!50003 DROP FUNCTION IF EXISTS `ADDCERO` */;;
/*!50003 SET SESSION SQL_MODE="NO_AUTO_VALUE_ON_ZERO"*/;;
/*!50003 CREATE*/ /*!50020 DEFINER=`root`@`localhost`*/ /*!50003
FUNCTION `ADDCERO`(variable INTEGER(11), numero INTEGER(11))
RETURNS varchar(11) CHARSET utf8
BEGIN
DECLARE conversion VARCHAR(11);
DECLARE init INT DEFAULT LENGTH(variable);
DECLARE i INT DEFAULT init;

SET conversion = variable;
WHILE i < numero DO

```

```

SET conversion = CONCAT('0',conversion);

SET i = i + 1;

END WHILE;

RETURN conversion;

END *;;

/*!50003 SET SESSION SQL_MODE=@OLD_SQL_MODE *;;

# Dump of FUNCTION NUMBER_FORMAT
# -----
/*!50003 DROP FUNCTION IF EXISTS `NUMBER_FORMAT` *;;
/*!50003 SET SESSION SQL_MODE="NO_AUTO_VALUE_ON_ZERO"*;;
/*!50003 CREATE*/ /*!50020 DEFINER=`root`@`localhost`*/ /*!50003
FUNCTION `NUMBER_FORMAT`(numero FLOAT, decimales
INT,separador_dec VARCHAR(1), separador_mil VARCHAR(1)) RETURNS
varchar(20) CHARSET latin1
BEGIN
DECLARE resultado VARCHAR(20);
SET resultado = numero;
IF separador_dec = "," THEN
SET resultado = REPLACE(REPLACE(REPLACE( FORMAT( numero ,
decimales ), ',', '@'), ',', '.'), '@', ',');
END IF;
IF separador_dec = "." THEN
SET resultado = FORMAT(numero,decimales);
END IF;
RETURN resultado;
END *;;

/*!50003 SET SESSION SQL_MODE=@OLD_SQL_MODE *;;

DELIMITER ;

```

```
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;

/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS
*/;

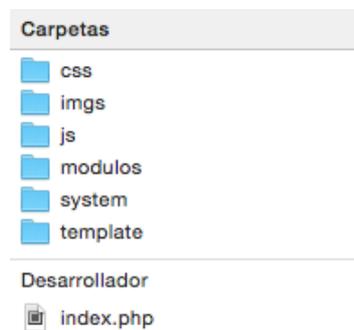
/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;

/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;

/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

## 5.2. Código de la Aplicación

- Organización de Archivos



La aplicación al ser web se desarrolla bajo la estructura convencional de un página web, encontrándose en ella las diferentes carpetas mencionadas a continuación:

### CSS

Es donde se guardan los archivos con extensión .css que sirven para darle estilo a las etiquetas HTML.

## **IMGS**

Se guardan las imágenes de la aplicación, tales como logotipo, íconos, y avatares para el diseño de la misma.

## **JS**

Se almacenan los archivos en lenguaje Javascript cuya extensión es .js

## **MODULOS**

Aquí se guardan los diferentes archivos que permiten mostrar los distintos módulos que posee la aplicación, entre ellos formularios y listado de información ingresada y guardada en la Base de Datos.

## **SYSTEM**

Contiene archivos de configuración del aplicativo, clases y librerías que sirven a la aplicación para realizar todas las funciones y características que este posee.

## **TEMPLATE**

Guarda archivos que sean comunes de incluir en el aplicativo, tal como cabeceras y pie de página.

## **Archivo index.php**

Es el que muestra el formulario de autenticación y es el que se ejecuta al momento de abrir la aplicación en un navegador.

## Configuración

Para configurar la aplicación por primera vez, se necesita un servidor apache. Una vez ubicados dentro del repositorio de aplicativos del servidor Apache por lo general denominada “WWW” se agrega todos los ficheros mostrados anteriormente y luego se procede a encontrar el archivo config.php dentro de la carpeta “system”.

```
1 <?php
2 ini_set('session.gc_maxlifetime','28800');
3 session_start(); // 8 horas de sesión
4 /**
5 ARCHIVO DE CONFIGURACION GLOBAL
6 */
7 Configurar $jedarz de la siguiente manera
8 - local: Para ambiente de desarrollo
9 - web: Para albergar en el servidor donde se hospedará
10 */
11 $config = 'local';
12
13 /**
14 Especifica Host o IP para la BD y el puerto
15 Por default siempre 'localhost' y puerto 80
16 */
17 $Host = 'localhost';
18 $Port = '';
19 //$IP = '192.168.1.4';
20 $IP = $Host;
21
22 /**
23 AMBIENTE DE DESARROLLO
24 Especifica los datos de Conexión a la BD
25 */
26 // Conexion BD Desarrollo
27 $Local_BD = 'hospedaje';
28 $Local_User = 'root';
29 $Local_Pass = 'root';
30
31 // Especifica una Cookie para sesión de usuario
32 $Cookie = 'RESTLogin';
33
34 // Carpeta principal de todo el sitio
35 $Carpeta_raiz = 'www/hospedaje';
36
37 /**
38 ARCHIVOS DE CONFIGURACION REQUERIDOS
39 */
40 // Requerido Archivo de Constantes Globales
41 require_once(dirname(__FILE__).'config/constantes.php');
42
43 // Requerido Archivo de Depuración
44 require_once(dirname(__FILE__).'config/debug.php');
45
46 // Requerido para Archivos Comunes
47 require_once(dirname(__FILE__).'config/common.php');
48
49
50
51
52 ?>
```

Dentro del archivo config.php existen las variables \$Local\_BD que es donde se asignará el nombre de la Base de Datos a la cual se debe conectar la aplicación.

También están \$Local\_User y \$Local\_Pass que son para asignar el usuario y contraseña de la base de datos respectivamente.

Se le configura un Host, un puerto y una dirección IP en caso de ser necesario para el funcionamiento de la Aplicación.

La variable `$Cookie` guarda el nombre de la cookie que se almacena en el navegador para las autenticaciones automáticas.

La variable `$Carpeta_raiz` guarda la ruta dentro de la carpeta “www” del servidor Apache desde la cual se han guardado los archivos y se piensa hacer correr la aplicación.

## **CLASES**

Las clases son las encargadas de hacer funcionar la aplicación desde su autenticación hasta el modelo de negocio de la empresa dentro del Aplicativo. Entre ellas destacaremos las más importantes junto a otros archivos que relacionen con ellas. Todas están ubicadas dentro de la ruta “system/clases/” en el aplicativo.

### **Permisos.class.php**

Verifica la autenticación de los usuarios que tratan de ingresar al aplicativo, dependiendo el nivel de usuario será destinado a su respectivo módulo de acceso.

También verifica que si un usuario trata de poner una ruta en la URL la cual no está autorizado a ingresar, éste será redireccionado a una página de ERROR 404 indicándole que dicho enlace no existe en el servidor.

Todo el proceso de autenticación se encuentra en el constructor de la clase asignando datos del usuario en el caso de ser una autenticación correcta.

```

1 <?php
2 class Permisos()
3
4     var $User;
5     var $bd;
6     //var $bd_func;
7     var $Login;
8     var $Usuario;
9     var $ID_Usuario;
10    var $Tipo_Usuario;
11    var $Persona;
12    var $img_user;
13    var $plantilla;
14
15    function __construct(){
16        $usuario = $_SESSION['usuario'];
17        $password = $_SESSION['password'];
18
19        $this->bd = new BD_Func();
20        $this->Login = new LOGIN();
21
22        if(!empty($usuario)){
23
24            $Usuario = $this->Login->Usuario($usuario,$password);
25            if(!$Usuario) $this->Error404();
26            elseif(!empty($Usuario)) {
27                $this->Usuario = $Usuario;
28                $this->Persona = $this->bd->identifica($Usuario['id_empleado'],'app_empleado');
29                $this->User = $usuario;
30                $this->Tipo_Usuario = $Usuario['id_tipo'];
31                $this->ID_Usuario = $Usuario['id_usuario'];
32                $this->plantilla = $this->Plantilla($this->ID_Usuario);
33
34                // $img_user = $this->bd->data_img($Usuario['id_imagen']);
35                // $this->img_user = $img_user['thumball'];
36            }
37        } else {
38            $this->Error404();
39        }
40    }
41
42    function Error404(){
43        header('Location: '.URL_LOCAL.'index.php');
44        die;
45    }
46

```

## Método Módulos

Se encuentra dentro de la clase de permisos.class.php y se encarga de consultar los diferentes módulos asignados en la Base de datos dependiendo el tipo de usuario que trate de acceder al aplicativo.

```

103
104 function|Modulos(){
105
106     $vista = func_get_arg(0);
107     if(func_num_args()==2) $identificador = func_get_arg(1);
108
109     $Modulos = array();
110     $Print = '';
111     $plantilla = ($vista=='plantilla') ? $identificador : $this->Plantilla($identificador);
112     $Modulos = $this->List_Modulos($plantilla);
113
114     $i = 0;
115     if(!$Modulos) return false;
116
117     foreach($Modulos as $item => $valor){
118         extract($valor);
119
120         //$data_img = $this->bd->data_img($id_imagen);
121         //$uri_imagen = $data_img['uri_img'];
122         //$tag_img = '';
123         //$img = ($vista=='menu') ? $tag_img : '';
124
125         $cod_mn = AddCero(($i+1),2);
126
127         $tag_cab = '<li><big>';
128         $cabecera = ($vista=='modulo' || $vista=='menu' || $vista=='nuevo' || $vista=='plantilla') ? strtoupper($descripcion) : '';
129         $cabecera = $tag_cab.$cabecera.</big>';
130
131         $tag_open = '<ul>';
132         $tag_close = '</ul>';
133
134         if($vista == 'modulo' || $vista == 'menu' || $vista == 'nuevo' || $vista == 'plantilla'){
135             $id_modulo = $id_modulo;
136         } else {
137             $id_modulo = ($id_modulo == $id_mnmod) ? $id_modulo : 0;
138             $tag_open = $tag_close = '';
139         }
140
141         $Secciones = $this->Secciones($id_modulo, $vista, $identificador);
142         //var_dump($Secciones);
143         if(!$Secciones) $data = '';
144         else if(!empty($Secciones)) $data = $cabecera.$tag_open.$Secciones.$tag_close.</li>';
145
146         $Print .= $data;
147
148         $i++;
149     }
150
151     return utf8_encode($Print);
152 }
153

```

## Interacción Base de Datos

Para la interacción con la Base de Datos existe la clase `bd.class.php` la cual contiene funciones para la administración de datos como consultas `select`, `insert`, `update` y `delete`, así como la generación de `querys sql`.

```

1 <?php
2 // Conexion y autentificacion dentro de la BD
3 class BD {
4
5     private $tb_join;
6     private $link_join;
7
8     function conectar(){
9         $error1 = 'No se puede conectar a la BD.<br>> Server: '.DB_SERVER.'<br>> User: '.DB_USER.'<br>> Pass: '.DB_PASS;
10        $error2 = 'La Base de Datos: '.DB_NAME.' , al parecer no existe';
11        //conexion MYSQL
12        $db_con = mysql_pconnect(DB_SERVER,DB_USER,DB_PASS) or die(MSG_ERROR($error1));
13        mysql_select_db(DB_NAME, $db_con) or die(MSG_ERROR($error2));
14        return $db_con;
15    }
16
17    function query(){
18
19        $vars;
20        $args = func_num_args();
21
22        if($args == 1){
23            $n_arg = func_get_arg(0);
24            $vars = is_array($n_arg) ? count($n_arg) : false;
25            $tabla = ($vars > 1) ? $n_arg[0] : false;
26        }
27
28        if($args > 1){
29            $tabla = func_get_arg(0);
30            $campos = func_get_arg(1);
31            $where = func_get_arg(2);
32            $query = "SELECT $campos FROM $tabla WHERE $where";
33        }
34
35        if($vars == 1) { $query = $n_arg[0];/*Sentencia SQL */ if(empty($query)) die(MSG_ERROR("<b>EL Query est&aacute; vac&iacute;"));
36        if($vars == 2) {
37            $campos = $n_arg[1];
38            $query = "SELECT $campos FROM $tabla WHERE 1";
39        }
40        if($vars == 3) {
41            if(is_numeric($n_arg[1])){
42                $i = $n_arg[1];/* inicio de pagina
43                $k = $n_arg[2];/* Items por página
44                $query = "SELECT * FROM $tabla WHERE 1 LIMIT $i,$k";
45            } else {
46                $campos = $n_arg[1]; $condicion = $n_arg[2];
47                $query = "SELECT $campos FROM $tabla WHERE $condicion";
48            }
49        }
50    }
51 }

```

Esta clase es la más usada dentro de otras clases siempre que se quiera acceder a los datos del sistema, a continuación un ejemplo de conexión:

```

// Objeto que conecta a la Base de Datos.
$BD = new BD();

// generamos un query sencillo y lo guardamos en una variable
$query = $BD->query("mi_tabla");

```

El parámetro que se le indica es el nombre de la tabla, si solo tiene un parámetro automáticamente traerá todos los campos existentes en la tabla ejecutando la siguiente instrucción:

```
$resultado = $BD->select($query);
```

Así mismo existen otros métodos como insert(), update() cuyo primer parámetro recibe el nombre de la tabla, el segundo, los campos y el tercero la condición la cual afecta la instrucción en el caso de update(), mientras que insert() en el tercer parámetro recibe un valor booleano para indicarle si TRUE para que devuelva el identificador de la tabla registrado en el caso de tener un PRIMARY autonumérico.

```
// Asignamos los campos a modificar
$campos = array('nombre'=>'Ricardo', 'apellido'=>'Mesías');
$id_cliente = $BD->insert('cliente',$campos,true);

// Actualización de datos con condición
$BD->update("cliente",$campos,"id_cliente=1");
```

## Clase Login.class.php

Esta clase se encarga de permitir el acceso de los usuarios al sistema por medio de un usuario y contraseña.

```
function __construct(){
    $this->BD = new BD_Func();
}

function Usuario( $usuario, $password ) {
    if ($usuario==' ' || $password==' ') return false;
    $query = "
        SELECT id_usuario,usuario,password, id_empleado,t.descripcion as tipo, nombre, apellido
        FROM web_usuario INNER JOIN web_usuario_tipo t USING(id_usuario_tipo)
        INNER JOIN app_empleado USING(id_empleado)
        WHERE usuario = '$usuario'
    ";
    $data = $this->BD->select($query);
    $password_from_db = $data ['password'];
    if ( $password_from_db == $password ) return $data;
    else return false;
}

function Entrar($usuario,$password,$recordar){
    $msg1 = '* Debe ingresar su Usuario y Contrase&ntilde;a';
    $msg2 = '* El Usuario o Contrase&ntilde;a no coinciden';
    if (empty($usuario) || empty($password)) $error = $msg1;
    else {
        $Usuario = $this->Usuario($usuario,md5($password));
        if(!$Usuario) $error = $msg2;
    }
    if(empty($error)) $this->Ingresar($Usuario,$recordar);
    else return $error;
}
```

## Clase Habitación.class.php

Esta clase se encarga de administrar las habitaciones del hostel, aquí existen procedimientos que asignan una habitación al huésped hasta la reserva de una habitación por periodos.

```
63
64 function Habitaciones($filtro){
65     $Data = array();
66     $data = $this->getData('habitaciones',$filtro,true);
67
68     $cabecera = array('ESTADO','Nombre de Habitación','Tipo de habitación','Fecha Ocupada','Hora Ocupada');
69     $config = array(
70         'tb_class' => 'tb_list',
71         'width' => array(1=>150,2=>200),
72         'pos_dataview' => 2,
73         'crypt' => true,
74         'button' => array('edit'=>'habitacion.php?ID=', 'del'=>'habitacion.php?del=')
75     );
76     $DataList = $this->Tabla->crear($cabecera,$data,$config);
77     $Paginacion = '<div id="paginacion">'. $this->paginar. '</div>';
78
79     return utf8_encode($DataList).$Paginacion;
80 }
81
82 function reserva_Habitaciones($filtro){
83     $Data = array();
84     $data = $this->getData('reserva',$filtro,true);
85
86     if(empty($data)){
87         foreach ($data as $item) {
88             $i++;
89             $id = $item['id'];
90             $sql= 'select * from app_registro_reserva where id_reserva_habitacion='.$item['id_reserva_habitacion'];
91             $res=$this->BD->select($sql);
92
93             foreach ($item as $key => $value){
94                 if(empty($res)){
95                     if($key=='0'){
96                         $item[$key] = '<button><a href="reserva_habitacion.php?reserva='.base64_encode($item['id_reserva_habitacion'])."&rad=registro">Registrar H
97                     }
98                 }else{
99                     if($key=='0'){
100                         $item[$key] = '<p style="width:100%; text-align:center;">Huesped Registrado</p>';
101                     }
102                 }
103             }
104             array_push($Data,$item);
105         }
106     }
107     $cabecera = array('Fecha de Reserva','Hora de Reserva','Cedula','Nombre','Apellido','Estado','Nombre de Habitación','');
108     $config = array(
109         'tb_class' => 'tb_list',
110         'width' => array(1=>75,2=>75),
111         'pos_dataview' => 2,
112         'crypt' => true,
113         'button' => array('edit'=>'reserva_habitacion.php?rad=reserva&ID=', 'del'=>'reserva_habitacion.php?del=')
114     );
115     $DataList = $this->Tabla->crear($cabecera,$Data,$config);
116     $Paginacion = '<div id="paginacion">'. $this->paginar. '</div>';
```

## Clase Evento.class.php

Esta clase se encarga de administrar los eventos del hostel, aquí existen procedimientos que muestran los diferentes servicios adicionales que pide el cliente al momento de contratar el servicio de eventos.

```

function Evento($filtro){
    $config=array();
    $data = array();
    $data = $this->getData('evento',$filtro,true);
    if(!empty($data)){
        foreach ($data as $item) {
            foreach ($item as $key => $value){
                if($key=='servicio'){
                    $servicios=$this->BD->select('app_evento','animacion,bufete','id_evento='.$item['id_evento']);
                    if($servicios['animacion']==1 && $servicios['bufete']==1){
                        $bufete='Bufete y';
                        $animacion='Animaci&oacute;n';
                    }else if($servicios['animacion']==1 && $servicios['bufete']==0 ){
                        $bufete='';
                        $animacion='Animaci&oacute;n';
                    }elseif($servicios['animacion']==0 && $servicios['bufete']==1 ){
                        $bufete='Bufete';
                        $animacion='';
                    }else{
                        $bufete='No tiene servicios adicionales';
                        $animacion='';
                    }
                }
                $item[$key] = $bufete.$animacion;
            }
            if($key=='0'){
                $item[$key] = '<button><a href=" ../ventas/facturas.php?Id_eve='.base64_encode($item['id_evento']).'">Facturar<
            }
            if($key=='1'){
                $servicios=$this->BD->select('app_evento','animacion,bufete','id_evento='.$item['id_evento']);
                if($servicios['animacion']==1 && $servicios['bufete']==1 ){
                    $url='evento.php?rad=1&rad1=2&ID='.base64_encode($item['id_evento']);
                    $item[$key] = '<a href='.$url.'" class="cursor bt_edit" title="revisar" </a>
                                <a href="evento.php?del='.base64_encode($item['id_evento']).'" class="cursor bt_del" title=
                }else if($servicios['animacion']==1 && $servicios['bufete']==0 ){

```

La tabla se genera con la siguiente línea de código y como es un método retornamos la tabla para llamarla en el formulario de esta forma:

### Crear la tabla:

```

$cabecera = array('FECHA','TIPO DE EVENTO','DESCRIPCIO&oacute;n','SERVICIOS ADICIONALES','','');

$config = array(
    'tb_class' => 'tb_list',
    'width' => array(1=>150,2=>200,6=>50),
    'pos_dataview' => 2,
    'crypt' => false
);
$DataList = $this->Tabla->crear($cabecera,$Data,$config);
$Paginacion = '<div id="paginacion">'.$this->paginar.'</div>';
return utf8_encode($DataList).$Paginacion;

```

### Llamada de la tabla en el formulario:

```

$BD = new BD_Func();
$Evento = new Evento();
$Form = new Form();
$AlertXSS;

$filtro = empty($filtro) ? 1 : $filtro;
$DataList = $Evento->Evento($filtro);

```

## Clase Compania.class.php

Esta clase se encarga de administrar los proveedores, clientes, empleados del hostel, aquí existen procedimientos que ayudan al funcionamiento de la misma como eliminación de proveedores, clientes y empleados, entre otros.

```
function Clientes($filtro){
    $Data = $this->getData('clientes',$filtro,true);
    $cabecera = array('CLIENTE','C&Eacute;DULA','TELEFONOS','DIRECCI&Oacute;N');
    $config = array(
        'tb_class' => 'tb_list',
        'width' => array(1=>240,2=>100,3=>100),
        'crypt' => true,
        'button' => array('edit'=>'clientes.php?ID=', 'del'=>'clientes.php?del=')
    );
    $DataList = $this->Tabla->crear($cabecera,$Data,$config);
    $Paginacion = '<div id="paginacion">'.$this->paginar.'</div>';

    return $DataList.$Paginacion;
}

function Proveedores($filtro){
    $Data = $this->getData('proveedores',$filtro,true);
    $cabecera = array('EMPRESA','RAZ&Oacute;N SOCIAL','R.U.C.','DIRECCION','TELEFONOS','EMAIL');
    $config = array(
        'tb_class' => 'tb_list',
        'width' => array(3=>100,4=>240),
        'crypt' => true,
        'button' => array('edit'=>set_Url('ID',''), 'del'=>set_Url('del',''))
    );
    $DataList = $this->Tabla->crear($cabecera,$Data,$config);
    $Paginacion = '<div id="paginacion">'.$this->paginar.'</div>';

    return $DataList.$Paginacion;
}

function Empleados($filtro){
    $Data = $this->getData('empleados',$filtro,true);
    $cabecera = array('NOMBRES Y APELLIDOS','CARGO','C&Eacute;DULA','TELEFONO','DIRECCION');
```

## Clase Servicio.class.php

Esta clase se encarga de administrar las mesas del restaurante, aquí existen procedimientos que ayudan a cambiar el estado de las mismas de libre-ocupada-reservada.

```

function Mesas($filtro){
    $Data = array();
    $Estados = $this->BD->list_items('app_mesa_estado');
    $data = $this->getData('mesas',$filtro,true);
    if(!empty($data)){
        foreach ($data as $item) {
            $i++;
            $id = $item['id'];
            foreach ($item as $key => $value){
                if($key=='estado'){
                    $item[$key] = '<select name="id_mesa_estado'.$i.'" id="id_mesa_estado'.$i.'" onChange="CambiarEstado(\''.$id
                }
            }
            unset($item['id_mesa_estado']);
            array_push($Data,$item);
        }
    }

    $cabecera = array('ESTADO','MESA','OBSERVACION');
    $config = array(
        'tb_class' => 'tb_list',
        'width' => array(1=>150,2=>200),
        'pos_dataview' => 2,
        'crypt' => true,
        'button' => array('edit'=>'mesas.php?ID=', 'del'=>'mesas.php?del=')
    );
    $DataList = $this->Tabla->crear($cabecera,$Data,$config);
    $Paginacion = '<div id="paginacion">'.$this->paginar.'</div>';

    return utf8_encode($DataList).$Paginacion;
}

function Mesa($ID){
    if(empty($ID)) return false;
    $campos = "*";
    return $this->BD->select('app_mesa',$campos,"id_mesa='$ID'");
}

```

## Clase Stock.class.php

Esta clase se encarga de administrar los servicios del hostel y los productos del restaurante, aquí existen procedimientos que ayudan a manejar las existencias en cuanto a productos se refiere y las bodegas.

```

function Bodegas($filtro){

    $Data = $this->getData('bodegas',$filtro,true);

    $cabecera = array('BODEGA','CIUDAD','RESPONSABLE','MONTETIZACION');
    $config = array(
        'tb_class' => 'tb_list',
        'width' => array(1=>300,3=>320,4=>100),
        'class' => array(4=>'txt_right'),
        'url' => array(1=>'bodegas.php?'),
        'crypt' => true,
        'button' => array('edit'=>'bodegas.php?ID=', 'del'=>'bodegas.php?del=')
    );

    $DataList = $this->Tabla->crear($cabecera,$Data,$config);
    $Paginacion = '<div id="paginacion">'.$this->paginar.'</div>';

    return utf8_encode($DataList).$Paginacion;
}

```

## Método existencia

Controla el stock de los productos actualizando la cantidad existente de productos.

```

function existencia($id_producto,$cantidad){
    $data = $this->BD->select('app_producto','stock',"id_producto='$id_producto'");
    $stock = $data['stock'];
    if($stock>0){
        $new_stock = $stock - $cantidad;
        $this->BD->update('app_producto',array('stock'=>$new_stock),"id_producto='$id_producto'");
        return true;
    }
}

```

## Clase Ventas.class.php

Esta clase se encarga de la realización de la factura, aquí existen procedimientos tanto para manejar el encabezado, detalle y totales de la factura, manejar las fechas de emisión y demás, y manejar el total del mes de ingresos y egresos del mismo.

```

function Facturas($filtro){
    $i=0;
    $Data = array();
    $data = $this->getData('facturas',$filtro,true);

    $excepto = array('id');
    $total_registros = count($data);

    $resumen_tipo = $_SESSION['resumen_tipo'];
    $por_mes = ($resumen_tipo==2) ? true : false;

    if(!empty($data)){
        foreach ($data as $item) {
            $i++;

            $fecha = $item['fecha'];
            $iva = $item['iva'];
            $total = $item['total'];

            // LISTADO NORMAL

            foreach ($item as $key => $value){
                if(is_numeric($value)&&!in_array($key,$excepto)) $item[$key] = '$ '.number_format($value,2,',','');
                if($key=='cliente'){
                    $item[$key] .= '<br><small class="c_gris">Pago: '.$item['pago_forma'].'</small>';
                }
            }

            unset($item['num_proforma'],$item['pago_forma']);

            // DESGLOCE POR DIA

```

## Método DetalleSession

Este método guarda el detalle en una variable de sesión o temporal en este caso llamada “\$session = \$\_SESSION['factura\_detalle];” este método permite agregar editar y eliminar del detalle de la factura , también va anidado a la carpeta js del directorio en donde es llamado mediante el archivo “factura.js”.

```

function DetalleSession($cmd,$campos,$identificador=false,$return=true,$tipo_detalle){

    $session = $_SESSION['factura_detalle'];
    switch ($cmd) {
        case 'add':
            extract($campos);

            $key = count($session);
            $cod = empty($identificador) ? rand(0,9999).$key : $identificador;

            if(!empty($total_costo)) $total = $total_costo;

            $campos = array(
                'id'           => $cod,
                'cantidad'    => $cantidad,
                'id_producto' => $id_producto,
                'producto'    => $producto,
                'precio'      => $precio,
                'total'       => $total
            );

            $session[$cod] = $campos;
            // $calcular = empty($ID) ? true : false;
            break;
        case 'edit':
            $data = $session[$identificador];
            return $data;
            break;
        case 'del':
            unset($session[$identificador]);
            break;
    }

    $_SESSION['factura_detalle'] = $session;

    if($return){
        $data['DataGrid'] = $this->Detalle('', $tipo_detalle);
        return $data;
    }
}

```

## Método detalle

Este método permite crear la sesión “\$\_SESSION['factura\_detalle']” pero con los datos extraídos de la base de datos, ideal para editar el detalle de la factura.

```

function Detalle($ID,$documento,$create_session=false){
    $Data = array();

    if($documento!='factura'){
        return false;
    }

    $tb_documento = 'cont_factura';
    $show_buttons = true;

    if(!empty($ID)){
        unset($_SESSION['factura_detalle']);

        $filtro = "pd.id_$documento = '$ID' AND pd.flag=1";
        $data = $this->getData($documento.'_detalle',$filtro,false);

        $reg = $this->BD->select($tb_documento,'descuento',"id_$documento = '$ID'");
        $descuento = $reg['descuento'];

        // Habilito edición de detalle
        if($documento=='factura'){
            $reg = $this->BD->select('cont_factura','id_factura_estado',"id_factura='$ID'");
            $id_estado = $reg['id_factura_estado'];
            // Cancelada o Anulada
            $show_buttons = ($id_estado==2 || $id_estado==3) ? false : true;
        }
        $sess_ = true;
    } else {
        $data = $_SESSION['factura_detalle'];
        $sess_ = false;
    }

    // Formato de impresión de pantalla
    if(!empty($data)){

```

## Método GuardarFactura

Permite guardar la factura en la base de datos. Se crea un arreglo en donde se encuentran todos los campos de la tabla de la base de datos y se insertan de esta forma:

```

function GuardarFactura($form){
    $data = $_SESSION['factura_detalle'];
    if(empty($data)){
        $result['msg'] = " * * * E R R O R * * \n\nDebe de ingresar items en el Detalle para poder guardar su documento.";
        $result['box'] = 'alert';
        return $result;
    }

    $data = deserializa($form);
    extract($data);
    $result['box'] = 'alert';

    $ID = ($por_factorar) ? '' : $ID;

    if(!is_numeric($ID)&&!empty($ID)){
        $result['msg'] = 'Hay un problema con la Factura, no hemos podido guardar los datos. ERROR XSSID-001';
        return $result;
    }

    $requeridos = array($num_referencia,$numero,$fecha,$id_cliente,$nif,$id_pago_forma,$id_estado);
    $MSG = $this->Form->Verificar($requeridos);

    if(!$MSG){
        // GUARDAR RUC Y DIRECCION DEL CLIENTE DE SER NECESARIO
        $Compania = new Compania();
        $Compania->CheckData($id_cliente,$nif,$direccion);

        $campos = array(
            'fecha' => $fecha,
            'num_referencia' => $num_referencia,
            'numero' => $numero,
            'id_cliente' => $id_cliente,
            'id_vendedor' => 1, // PEANDRES S.A., quemado por La BD
            'id_pago_forma' => $id_pago_forma,
            'id_mesa' => $id_mesa,
        );
    }
}

```

Y por último el método que almacena el detalle de la factura, para ello guarda la variable de sesión que antes se creó.

```

private function GuardaDetalleFactura($ID,$id_bodega){
    $session = $_SESSION['factura_detalle'];
    $data = $this->BD->select('cont_factura_detalle','id_factura',"id_factura='$ID' and flag=1");
    $new = empty($data) ? true : false;

    if(!$new){
        $this->BD->delete('cont_factura_detalle',"id_factura='$ID'",false);
        //$this->BD->delete('app_naviera_item',"id_transaccion='$ID' AND id_documento = 1",false);
    }

    $Stock = new Stock();

    foreach ($session as $item) {
        extract($item);
        $subtotal = $cantidad * $precio;
        $campos = array(
            'id_factura' => $ID,
            'id_item' => $id_producto,
            'cantidad' => $cantidad,
            'precio' => $precio,
            'subtotal' => $subtotal
        );
        $id_factura_detalle = $this->BD->insert('cont_factura_detalle',$campos,true);

        //$Stock->existencia($id_producto,$cantidad);
        // Proforma, documento = 1
    }

    return true;
}

```

## Librería para la vista de las mesas

Se utilizó una carpeta llamada dist dentro módulos en donde se encuentra la librería a usar en este caso “gridstack”. Esta librería permite arrastrar, posicionar, cambiar el estado de las mesas del restaurante, además de guardar dicha interfaz.

```
ko.components.register('dashboard-grid', {
  viewModel: {
    createViewModel: function (controller, componentInfo) {
      var ViewModel = function (controller, componentInfo) {
        var grid = null;

        this.widgets = controller.widgets;

        this.afterAddWidget = function (items) {
          _.each(items, function (item) {
            item = $(item);

            if (item.data('_gridstack_node') || item[0].nodeType != 1)
              return;

            if (grid == null) {
              grid = $(componentInfo.element).find('.grid-stack').gridstack({
                auto: false
              }).data('gridstack');
            }

            grid.add_widget(item);
            ko.utils.domNodeDisposal.addDisposeCallback(item[0], function () {
              grid.remove_widget(item);
            });
          }, this);
        };
      };
      return new ViewModel(controller, componentInfo);
    }
  }
});
```

Por medio de la sección template se incrusta el código html en nuestra página.

```
},
template:
[
  '<div class="grid-stack" data-bind="foreach: {data: widgets, afterRender: afterAddWidget}">',
  '  <div class="grid-stack-item " data-bind="attr: {\`data-gs-x\`: $data.x, \`data-gs-y\`: $data.y, \`data-gs-wid',
  '  <b style="vertical-align:middle;"><div class="grid-stack-item-content mesas"><div data-bind="text: $data.',
  '  </div>',
  '  </div>',
  ].join('\n')
```

Y en la función donde se agregan los “widgets” se accede al template antes mencionado con los datos extraídos de la base de datos.

```

$(function () {
    var Controller = function (widgets) {
        this.widgets = ko.observableArray(widgets);
    };

    var widgets = [];
    <?php
    $i=0;
    foreach ($data as $key => $value) {
        extract($value);
        $contenido = '<select id="id_mesa_estado'.$id_mesa.'" name="id_mesa_estado'.$id_mesa.'" style="width:110px;" oncha
        echo "widgets[".$i."] = {x: $pos_x, y:$pos_y, width:$width, height:$height, id_mesa:$id_mesa, id:'mesa$'. $i', descripcion
        $i++;
    }
    ?>

    ko.applyBindings(new Controller(widgets));
    $('#grid-stack').on('dragstop', function (event, ui) {
        alert(grid.ui);
    });
});
});

```

## Método cambiar estado de la mesa

Se encuentra en la carpeta js del directorio en el archivo gui.js, este método permite actualizar los estados de las mesas si esta cambia a libre estando ocupada entonces tendrá que facturar, caso contrario actualiza el estado en cualquier momento.

```

function CambiarEstado(id,estado,id_estado_anterior){
    url = URL_LOCAL+'modulos/assets/data.grid.php';
    $.post(url,{ 'id':id,'estado':estado,'identificador':'mesa_estado','id_estado_anterior':id_estado_anterior},function(data){
        console.log(data);
        if(data=='facturar'){
            var r =confirm('Antes de cambiar el estado de la mesa a libre, debe de facturar. ¿Desea facturar?');
            if (r == true) {
                location.href = "../modulos/ventas/facturas.php?id_mesa="+id;
            }
        }
    });
}

```

El archivo gui.js hace uso del archivo data.grid.php contenido en la carpeta modulos/assets/.

En este método se realiza la actualización.

```

        break;
    case 'mesa_estado':
        if($estado==1){
            if($id_estado_anterior!=3){
                echo 'facturar';
            }else{
                $BD->update('app_mesa',array('id_mesa_estado'=>$estado),"id_mesa = '$id'");
                echo 'ok';
            }
        }else{
            $BD->update('app_mesa',array('id_mesa_estado'=>$estado),"id_mesa = '$id'");
            echo 'ok';
            die;
        }
        break;

```

## Módulos

En la carpeta módulos se encuentran los diferentes formularios que permitirán el ingreso y actualización de la información.

Carpeta: archivo.php

En este archivo se encuentra tanto el formulario para el ingreso de la información como el listado de la misma. Esta separación se realiza por medio de la variable “?ID=” en la url , si existe tal variable en la url se mostrara el formulario, caso contrario el listado respectivo.

```
$visible_list = !isset($ID) ? 'visible' : 'oculto';  
$visible_form = isset($ID) ? 'visible' : 'oculto';
```

El formulario guarda y actualiza la información, y el listado muestra la información guardada en la base de datos por medio de una tabla.

Cientes: Formulario nuevo\_cliente.php

Permite agregar y actualizar al cliente. En el arreglo campos se encuentran todos los campos de la tabla donde se almacenan los clientes, si deseamos agregar un nuevo campo a la tabla es en ese arreglo debemos agregar el campo de la siguiente manera:

```
'campo'=>utf8_decode($campo) ,
```

Para poder obtener la variable php (\$campo) debe agregar en el formulario una caja de texto de donde se extrae ese valor de la siguiente manera:

```
<input type="text" name="campo" id="campo" value="<?= Campo($campo)  
?>">
```

Código que me permite guardar el cliente en la base de datos:

```
if(isset($Save)){
    $requeridos = array($dni);
    $MSG = $Form->Verificar($requeridos);

    $campos = array(
        'nombre' => utf8_decode($nombre),
        'apellido' => utf8_decode($apellido),
        'dni' => utf8_decode($dni),
        'direccion' => utf8_decode($direccion),
        'telefono' => utf8_decode($telefono)
    );

    if(!$MSG){
        if(!$ID){
            $existen = array('nombre'=>$nombre,'apellido'=>$apellido,'dni'=>$dni);
            $existe = $BD->existe('app_cliente',$existen);

            if(empty($existe)){
                $MSG = 'Ya existe un app_cliente con mismo el c&oacute;digo o mismas características.';
                $Box = 'info';
            } else if(!$existe){

                // $id_imagen = $Imagen->Guardar($img_servidor,$folder_upload,280,250,$Permisos->ID_Usuario);
                // $campos['id_imagen'] = $id_imagen;
                $BD->insert('app_cliente',$campos);
                $process = 'guardado';
            }
        } else {

            if($img_bd != $imagen){
```

## Cientes: clientes.php

Muestra el listado de los clientes accediendo a la clase compania.class.php de la siguiente manera:

```
require_once('../system/config.php');
OpenClass('bd,login,permisos,compania,form');
OpenLibrary('page,table');

$Permisos = new Permisos();
// $Permisos->Usuario(getUrl()); // Verifico Permisos
$id_usuario = $Permisos->ID_Usuario;

$BD = new BD_Func();
$Compania = new Compania();

if(isset($del) && !empty($del)){
}

// Filtro Fechas
if(isset($search_date)){
    $f1 = empty($f1) ? date('Y-m-d') : $f1;
    $f2 = empty($f2) ? date('Y-m-d') : $f2;
    $data = base64_encode($f1.'_'.$f2);
    $url = quit_Url('camp');
    header('Location:'.set_Url('date',$data,$url));
}
if(!empty($date)){
    $data = explode('_',base64_decode($date));
    $f1 = $data[0];
    $f2 = $data[1];
    $filtro = "r.fecha >= '$f1' AND r.fecha <= '$f2'";
}

$filtro = empty($filtro) ? 1 : $filtro;

$Datalist = $Compania->Clientes($filtro);
>>
```

## Hospedaje: habitacion.php

El formulario guarda y actualiza la información de las habitaciones del hostal.

¿Cómo diferenciamos el guardar del actualizar? Pues se lo hace por medio del ID en la url si esta variable es diferente de vacía se actualiza, caso contrario se inserta el registro, todo esto se hace accediendo al arreglo campos donde se tiene todos los campos de la tabla “app\_habitación” de la base de datos.

```
if(isset($Save)){  
  
    $requeridos = array($nombre_habitacion,$tipo_habitacion,$estado);  
    $MSG = $Form->Verificar($requeridos);  
  
    $campos = array(  
        'nombre_habitacion' => utf8_decode($nombre_habitacion),  
        'num_piso' => utf8_decode($num_piso),  
        'tipo_habitacion' => utf8_decode($tipo_habitacion),  
        'num_baños' => utf8_decode($num_baños),  
        'telefono' => utf8_decode($telefono),  
        'aire_acondicionado' => utf8_decode($aire_acondicionado),  
        'ventilador' => utf8_decode($ventilador),  
        'ventana' => utf8_decode($ventana),  
        'television' => utf8_decode($television),  
        'wifi' => utf8_decode($wifi),  
        'tv_pagada' => utf8_decode($tv_pagada),  
        'estado' => utf8_decode($estado),  
        'costo' => utf8_decode($costo),  
        'id_imagen' => utf8_decode($id_imagen)  
    );  
  
    if(!$MSG){  
        if(!ID){  
            $existen = array('nombre_habitacion'=>$descripcion, 'estado'=>$estado);  
            $existe = $BD->existe('app_habitacion',$existen);  
  
            if(empty($existe)){  
                $MSG = 'Ya existe una Habitacion con la misma descripción.';  
                $Box = 'info';  
            } else if(!$existe){  
  
                $BD->insert('app_habitacion',$campos);  
                $process = 'guardada';  
            }  
        } else {
```

Y el listado muestra la información guardada de las habitaciones en la base de datos por medio de una tabla extraída de un método “habitacion” de la clase habitación del archivo habitacion.class.php .

```

<?php
require_once('../system/config.php');
OpenClass('bd,login,permisos,habitacion,form');
OpenLibrary('table,page');

$Permisos = new Permisos();
$Permisos->Usuario(getUrl()); // Verifico Permisos
$id_usuario = $Permisos->ID_Usuario;

$BD = new BD_Func();
$Habitacion = new Habitacion();
$Form = new Form();
$AlertXSS;

$filtero = empty($filtero) ? 1 : $filtero;
$dataList = $Habitacion->Habitaciones($filtero);

```

El mismo procedimiento es para todos los archivos de la carpeta módulos.

Hospedaje: reserva\_habitacion.php

Este formulario lo usamos tanto para guardar la reserva de la habitación como para el registro del huésped, separamos estas dos funcionalidades mediante radios buttons

Acción a realizar  Reservar Habitación  Registro Huésped

Para poder diferenciar y saber qué es lo que queremos guardar usamos dos botones uno que permita guardar la reserva y otro el registro, estos botones los controlamos poniéndolos visibles y ocultándolos mediante los radios buttons antes mencionados.

## El método guardar reserva habitación:

```
if(isset($Save)){
    $requeridos = array($cedula,$nombre,$apellido);
    $MSG = $Form->Verificar($requeridos);

    //defino fecha 1
    $ano1 = substr($fecha_inicio, 0,4);
    $mes1 = substr($fecha_inicio, 5,2);
    $dia1 = substr($fecha_inicio, 8,2);
    $hora1= substr($hora_inicio, 0,2);
    $min1= substr($hora_inicio, 3,2);
    //defino fecha 2
    $ano2 = substr($fecha_fin, 0,4);
    $mes2 = substr($fecha_fin, 5,2);
    $dia2 = substr($fecha_fin, 8,2);
    $hora2= substr($hora_fin, 0,2);
    $min2= substr($hora_fin, 3,2);
    //calculo timestam de Las dos fechas
    $timestamp1 = mktime($hora1,$min1,0,$mes1,$dia1,$ano1);
    $timestamp2 = mktime($hora2,$min2,0,$mes2,$dia2,$ano2);
    //resto a una fecha La otra
    $segundos_diferencia = $timestamp1 - $timestamp2;
    //echo $segundos_diferencia;

    //convierto segundos en días
    $num_dias = $segundos_diferencia / (60 * 60 * 24);
    //obtengo el valor absoluto de Los días (quito el posible signo negativo)
    $num_dias = abs($num_dias);

    //quito los decimales a los días de diferencia
    $num_dias = floor($num_dias);

    $campos = array(
        'id_cliente' => utf8_decode($id_cliente),
        'num_dias' => utf8_decode($num_dias),
        'num_personas' => utf8_decode($num_personas),
        'forma_pago' => utf8_decode($forma_pago),
        'fecha_inicio' => utf8_decode($fecha_inicio.' '.$hora_inicio),
```

En este método además de guardar el registro también se maneja el estado de la habitación mediante la actualización de la tabla de la misma, si el estado de la reserva es creada entonces la habitación estará reservada más la fecha en que será ocupada, Si es despachada entonces el estado de la habitación será ocupada más la fecha en que será ocupada, caso contrario la habitación estará disponible.

```

if(!empty($existe)){
    $MSG = 'Ya existe una reservación con la misma descripción.';
} else if(!$existe){

    $BD->insert('app_reserva_habitacion',$campos);

    if($estado=='Creada'){
        $campo = array('estado'=>utf8_decode('reservada'),'fecha_ocupada'=>utf8_decode($fecha_inicio.' '.$hora_inicio));
        $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
    }else if($estado=='Despachada'){
        $campo = array('estado'=>utf8_decode('ocupada'),'fecha_ocupada'=>utf8_decode($fecha_inicio.' '.$hora_inicio));
        $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
    }else{
        $campo = array('estado'=>utf8_decode('libre'));
        $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
    }

    $process = 'ingresada';
}

} else {

    $BD->update('app_reserva_habitacion',$campos,"id_reserva_habitacion='$ID'");
    if($estado=='Creada'){
        $campo = array('estado'=>utf8_decode('reservada'),'fecha_ocupada'=>utf8_decode($fecha_inicio.' '.$hora_inicio));
        $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
    }else if($estado=='Despachada'){
        $campo = array('estado'=>utf8_decode('ocupada'),'fecha_ocupada'=>utf8_decode($fecha_inicio.' '.$hora_inicio));
        $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
    }else{
        $campo = array('estado'=>utf8_decode('libre'));
        $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
    }
}

```

El método para guardar el registro del huésped: En este método aparte de registrar el huésped manejamos el estado de la habitación si esta es despachada, osea el huésped se ha registrado se actualiza la habitación con un estado de ocupada y la fecha en que se ocupó dicha habitación.

```

if(isset($SaveRegistro)){
    $requeridos = array($cedula,$nombre,$apellido);
    $MSG = $Form->Verificar($requeridos);
    $campos = array(
        'id_cliente' => utf8_decode($id_cliente),
        'num_dias' => utf8_decode($num_dias),
        'num_personas' => utf8_decode($num_personas),
        'forma_pago' => utf8_decode($forma_pago),
        'fecha_inicio' => utf8_decode($fecha_inicio.' '.$hora_inicio),
        'fecha_fin' => utf8_decode($fecha_fin.' '.$hora_fin),
        'fecha_reserva' => utf8_decode($fecha_reserva),
        'id_habitacion' => utf8_decode($id_habitacion),
        'estado' => utf8_decode($estado)
    );
}

if(!$MSG){
    if(!$registro){
        $existen = array('id_cliente'=>$id_cliente, 'fecha_inicio'=>$fecha_inicio, 'fecha_fin'=>$fecha_fin);
        $existe = $BD->existe('app_registro_habitacion',$existen);

        if(!empty($existe)){
            $MSG = 'Ya existe una reservación con la misma descripción.';
            $Box = 'error';
        } else if(!$existe){

            $BD->insert('app_registro_habitacion',$campos);

            if($estado=='Creada'){
                $campo = array('estado'=>utf8_decode('reservada'));
                $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
            }else if($estado=='Despachada'){
                $campo = array('estado'=>utf8_decode('ocupada'),'fecha_ocupada'=>utf8_decode($fecha_inicio.' '.$hora_inicio));
                $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
            }else{
                $campo = array('estado'=>utf8_decode('libre'));
                $BD->Update('app_habitacion',$campo,"id_habitacion='$id_habitacion' ");
            }
        }
    }
}

```

## Ventas: facturas.php

Aquí usamos una ventana emergente tanto para buscar el cliente como para buscar el producto.

```
<div class="PanelBox" id="ListaClientes">
  <span class="close" title="Cerrar" onClick="Cerrar('ListaClientes');"></span>
  <h2 class="txt_light line_bottom">Listado de Clientes</h2>
  <span class="search_box">
    <b>Buscar</b><input type="text" id="campo_cliente" onKeyUp="Consultar(this.value)">
  </span>
  <table><tr class="title_grid"><td>Cliente</td></tr></table>
  <ul class="DataList borde"></ul>
  <footer>Seleccione un elemento de la Lista</footer>
</div>

<div class="PanelBox" id="ListaProductos">
  <span class="close" title="Cerrar" onClick="Cerrar('ListaProductos');"></span>
  <h2 class="txt_light line_bottom">Listado de Productos</h2>
  <span class="search_box">
    <b>Buscar</b><input type="text" id="campo_producto" onKeyUp="Consultar('todo',this.value)">
  </span>
  <table><tr class="title_grid"><td>Producto</td></tr></table>
  <ul class="DataList borde"></ul>
  <footer class="right"><div style="position:absolute; right:7px; bottom:5px;"><b style="margin:10px 5px 5px 5px;">Filtrar por:</b><
    <span style="margin:5px;"><input type="radio" name="rad" id="rad_servicio" <?= Campo($rad,'servicio','radio') ?> </div>
  </div>
```

Esta ventana hace uso de algunos métodos contenidos en la carpeta js en el archivo gui.js como:

## Consultas AJAX

AJAX es una tecnología que permite traer datos de forma asíncrona lo que ayuda en las funcionalidades de consultas tales como seleccionar un producto al momento de Facturar.

## Método consultar

Este método sirve para consultar por nombre los productos o clientes por medio de la caja de texto “campo\_cliente” en el caso de los clientes y “campo\_producto” en el caso de los productos.

Quien abre la ventana modal es el método abrir contenido en la carpeta js, en el archivo core.js .

```

function Consultar(cmd,valor){
    var url = URL_LOCAL+'modulos/assets/data.grid.php';

    if(cmd=='todo'){
        var cmd= $('input[name=rad]:checked').attr('value');
    }

    $.post(url,{'identificador':cmd, 'buscar': valor},function Recibe(data){
        Consola(data);
        $(".DataList").html(data);
    });
}

```

### Método Abrir

```

function Abrir(box){Modal(Set_Modal);$("#"+box).fadeIn();}

```

Este método en cambio se puede usar para especificar qué tipo de ventana queremos abrir, como en el caso de las consultas. Este método muestra y oculta las ventanas modales con JavaScript.

### 5.3. Pruebas y Resultados

Se realizan pruebas de la aplicación Web y se utiliza una computadora por medio de una intranet con un servidor y un computador. A continuación se muestra los resultados de la prueba:



Figura 11: Pantalla de Bienvenida

Elaborado por: Autores

## Página de Inicio

A continuación se presenta la pantalla de Inicio en donde se muestra una bienvenida amigable a la aplicación además se pueden observar las opciones principales acompañadas de imágenes que representan los servicios que brinda la Hostería Comunitaria África Mía

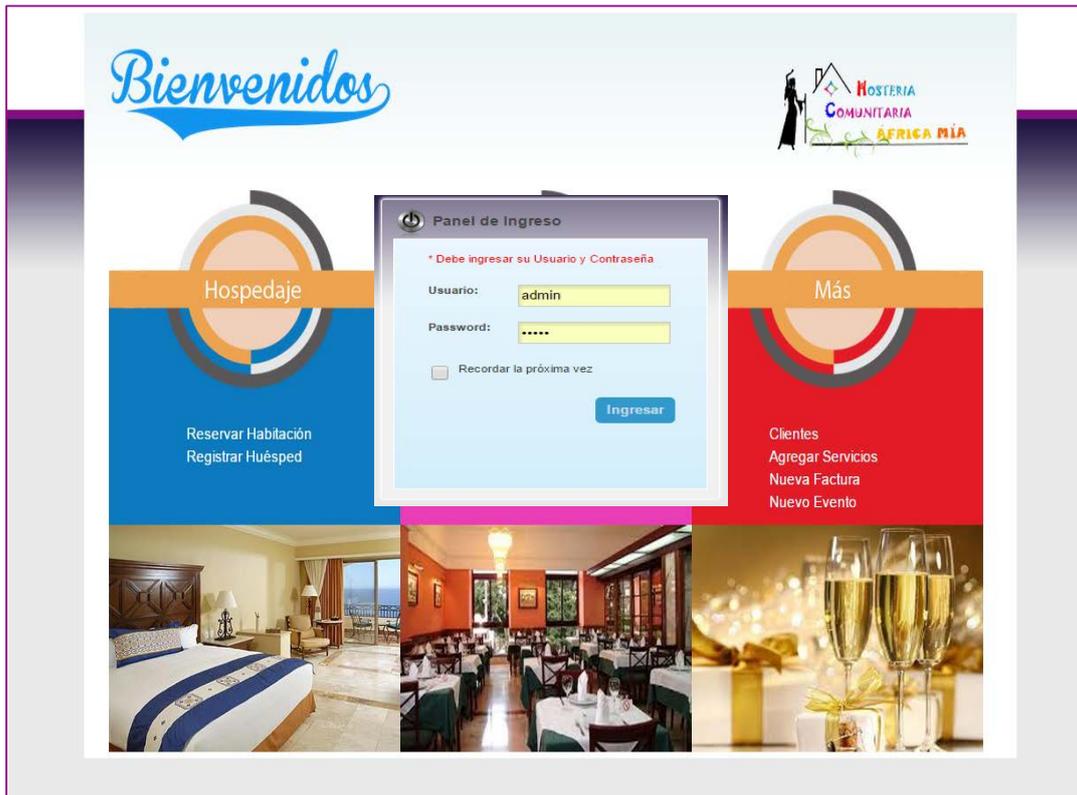


Figura 12: Pantalla de Ingreso a usuarios

Elaborado por: Autores

## Habitaciones

En esta sección se puede crear las habitaciones, tipo, sus características y servicios además del costo del alquiler de la misma.

**ADMINISTRADOR LOCAL** **SALIR**  
 INICIO HOSPEDAJE EVENTOS RESTAURANT VENTAS CLIENTES PERSONAL

**DISPONIBILIDAD DE HABITACIONES** **Nuevo**

**NUEVA HABITACIÓN**

Nombre de habitación   
 Número de piso   
 Tipo de Habitación   
 Número de baños   
 Teléfono  Sí  No  
 Aire Acondicionado  Sí  No  
 Ventilador  Sí  No  
 Ventana  Sí  No  
 Televisión  Sí  No  
 Wifi  Sí  No  
 Tv pagada  Sí  No  
 Estado   
 Costo

**LIMPIAR** **GUARDAR**

Figura 13: Disponibilidad de habitaciones

Elaborado por: Autores

### Listado de Habitaciones

Aquí se pueden visualizar las Habitaciones existentes en la Hostería además del tipo de habitación, estado (disponible / ocupada) la fecha y hora en que se realizó el proceso y también se pueden modificar las características y servicios que poseen las habitaciones ya creadas

**ADMINISTRADOR LOCAL** **SALIR**  
 INICIO HOSPEDAJE EVENTOS RESTAURANT VENTAS CLIENTES PERSONAL

**LISTADO DE HABITACIONES**

| ESTADO    | NOMBRE DE HABITACIÓN | TIPO DE HABITACIÓN | FECHA OCUPADA | HORA OCUPADA |  |
|-----------|----------------------|--------------------|---------------|--------------|--|
| reservada | Africa Matrimonial   | matrimonial        | 2015-03-19    | 11:42:00     |  |
| reservada | Corfu Suite          | matrimonial        | 2014-12-22    | 12:00:00     |  |
| reservada | imperial             | doble              | ---           | ---          |  |
| ocupada   | MILLENIUM            | doble              | 2015-01-05    | 12:00:00     |  |
| reservada | penthouse            | matrimonial        | 2015-01-06    | 12:00:00     |  |
| ocupada   | presidencial         | familiar           | 2015-01-09    | 12:00:00     |  |
| reservada | presidencial         | sencilla           | 2014-12-20    | 12:00:00     |  |
| libre     | Simple               | sencilla           | ---           | ---          |  |
| reservada | suit                 | familiar           | ---           | ---          |  |
| ocupada   | suit                 | doble              | 2015-01-10    | 12:00:00     |  |
| ocupada   | suit                 | doble              | 2015-03-19    | 11:49:00     |  |
| reservada | suit junior          | doble              | 2015-03-12    | 12:00:00     |  |

Figura 14: Listado de habitaciones

Elaborado por: Autores

## Reservaciones

En esta sección se puede reservar una habitación.

The screenshot shows a web interface for making a reservation. The header includes the logo for 'HOSTERIA COMUNITARIA AFRICA MIA' and a navigation menu with options like 'INICIO', 'HOSPEDAJE', 'EVENTOS', 'RESTAURANT', 'VENTAS', 'CLIENTES', and 'PERSONAL'. The main content area is titled 'NUEVA RESERVACION' and contains a form for 'RESERVAR HABITACION'. The form has several input fields and dropdown menus, along with a 'Buscar cliente' button. At the bottom of the form, there are 'LIMPIAR' and 'GUARDAR' buttons.

Figura 15: Pantalla de nueva Reservación

Elaborado por: Autores

## Listado de Reservaciones

Aquí se pueden visualizar las reservaciones, la fecha, hora de reserva, estado además se pueden editar y eliminar las reservaciones.

The screenshot shows a table listing reservations. The table has columns for reservation date, time, ID, name, surname, status, and room name. Each row includes a 'Registrar Huesped' button and a delete icon. The data is as follows:

| FECHA DE RESERVA | HORA DE RESERVA | CEDULA       | NOMBRE            | APELLIDO     | ESTADO     | NOMBRE DE HABITACIÓN |                    |
|------------------|-----------------|--------------|-------------------|--------------|------------|----------------------|--------------------|
| 2015-03-24       | 09:26:58        | 012345       | Jorge             | miRANDA      | Creada     | imperial             | Registrar Huesped  |
| 2015-03-19       | 11:43:25        | 0925094982   | Ricardo Francisco | Mesias Ochoa | Creada     | Africa Matrimonial   | Registrar Huesped  |
| 2015-03-10       | 12:04:21        | 0931485296   | Steffi            | Zoller       | Creada     | penthouse            | Registrar Huesped  |
| 2015-03-02       | 11:33:35        | 092887373848 | Tatiana           | Cacinelli    | Creada     | suit junior          | Huesped Registrado |
| 2015-01-05       | 12:34:02        | 092887373848 | Tatiana           | Cacinelli    | Despachada | presidencial         | Huesped Registrado |
| 2015-01-05       | 12:16:06        | 0928382919   | Manuel            | Quinteros    | Despachada | MILLENIUM            | Huesped Registrado |
| 2015-01-05       | 11:48:16        | 0931485296   | Steffi            | Zoller       | Despachada | suit                 | Huesped Registrado |
| 2015-01-05       | 08:58:41        | 0931485288   | Heinzel           | Zoller       | Despachada | penthouse            | Huesped Registrado |
| 2014-12-30       | 02:49:05        | 0931485296   | Steffi            | Zoller       | Creada     | suit                 | Huesped Registrado |
| 2014-12-30       | 01:17:55        | 092887373848 | Tatiana           | Cacinelli    | Despachada | imperial             | Huesped Registrado |
| 2014-12-30       | 01:15:11        | 0931485296   | Steffi            | Zoller       | Despachada | Corfu Suite          | Huesped Registrado |

Figura 16: Listado de Reservaciones

Elaborado por: Autores

## Huéspedes

En esta sección se realiza el registro de los huéspedes y se puede ir a la sección reservaciones en el lado derecho de la pantalla.

The screenshot shows a web interface for a hostel. At the top, there is a navigation bar with links: INICIO, HOSPEDAJE, EVENTOS, RESTAURANT, VENTAS, CLIENTES, PERSONAL. A user is logged in as 'ADMINISTRADOR LOCAL' with a 'SALIR' button. The main section is titled 'REGISTRO HUESPEDES' and contains a 'Nuevo' button. Below this is a form titled 'RESERVAR HABITACION'. The form has two radio buttons: 'Reservar Habitación' (selected) and 'Registro Huésped'. The form fields include: 'Fecha Reserva' (calendar icon), 'Cédula' (text input), 'Nombre' (text input), 'Apellido' (text input), 'Dirección' (text input), 'Teléfono' (text input), 'Fecha y hora inicio' (date and time pickers), 'Fecha y hora fin' (date and time pickers), 'Habitaciones Disponibles' (dropdown menu), 'Número de personas' (text input), 'Número de días' (text input), and 'Forma de Pago' (dropdown menu). There are two buttons: 'Buscar cliente' and 'Buscar Reservacion'. At the bottom of the form are 'LIMPIAR' and 'GUARDAR' buttons. To the right of the form is a sidebar titled 'Reservas' with a logo and the text 'Encuentra las reservaciones hechas en este espacio' and a button 'Ir a reservaciones'.

Figura 17: Reservar habitación

Elaborado por: Autores

## Huéspedes Registrados

Aquí se puede visualizar todos los huéspedes registrados y se puede ver la fecha y hora del registro.

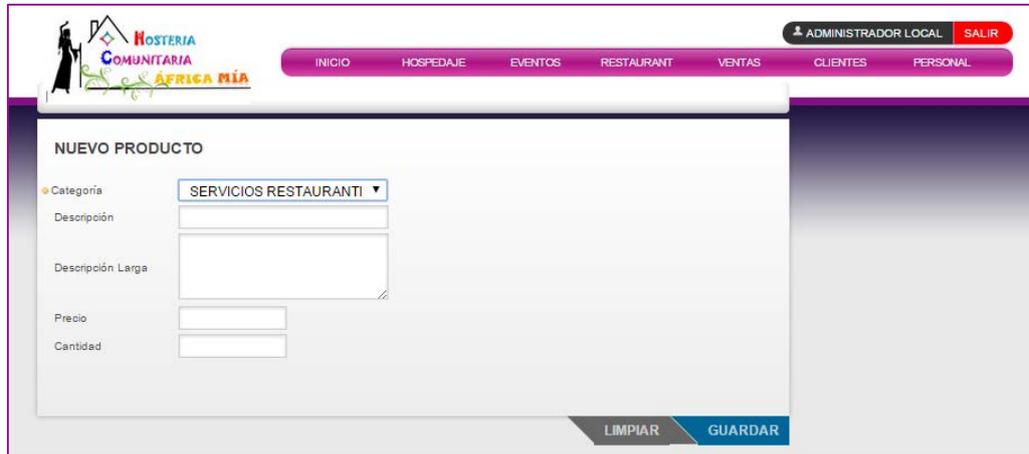
| CÉDULA       | NOMBRE  | APELLIDO  | ESTADO     | NOMBRE DE HABITACIÓN |  |
|--------------|---------|-----------|------------|----------------------|--|
| 0931485298   | Steffi  | Zoller    | Despachada | Corfu Suite          | Facturar   |
| 0928382919   | Manuel  | Quinteros | Despachada | MILLENIUM            | Facturar   |
| 0931485288   | Heinzel | Zoller    | Despachada | penthouse            | Facturar   |
| 092887373848 | Tatiana | Cacinelli | Despachada | presidencial         | Facturar   |

Figura 18: Huéspedes Registrados

Elaborado por: Autores

## Ingreso de Productos o Servicios

En esta sección se puede Ingresar los productos o servicios clasificándolos por Categoría.



The screenshot shows a web interface for adding a new product. The header includes the logo 'HOSTERIA COMUNITARIA AFRICA RIA' and navigation links: INICIO, HOSPEDAJE, EVENTOS, RESTAURANT, VENTAS, CLIENTES, PERSONAL, ADMINISTRADOR LOCAL, and SALIR. The main form is titled 'NUEVO PRODUCTO' and contains the following fields:

- Categoría: SERVICIOS RESTAURANTI (dropdown menu)
- Descripción: (text input)
- Descripción Larga: (text area)
- Precio: (text input)
- Cantidad: (text input)

Buttons at the bottom right: LIMPIAR and GUARDAR.

Figura 19: Pantalla de Ingreso de productos

Elaborado por: Autores

## Listado de Productos

En esta sección se pueden visualizar los productos y servicios ingresados, el tipo de producto o servicio, el precio además de la existencia en stock. También se puede realizar búsquedas del producto deseado.



The screenshot shows a table titled 'STOCK' with a search bar labeled 'PRODUCTO' and a 'Buscar' button. The table contains the following data:

| CÓDIGO | PRODUCTO   | TIPO DE PRODUCTO      | EXISTENCIA | PRECIO UNIT. |   |
|--------|--|-----------------------|------------|--------------|---|
| 10     | Agua<br>Agua   | BEBIDAS               | 1000       | \$ 1,00      |   |
| 4      | Chuleta con salsa<br>Chuleta aderezada con salsa bbq     | PLATOS FUERTES        | 12         | \$ 8,50      |   |
| 2      | cocacola<br>bebida negra                                 | BEBIDAS               | 200        | \$ 1,00      |   |
| 18     | Desayuno a habitacion<br>Llevar desayuno a la habitacion | SERVICIOS RESTAURANTE | ---        | \$ 5,50      |   |
| 9      | Dulce tres leches  | POSTRES               | 39         | \$ 1,50      |   |
| 11     | Gelatina<br>Gelatina                                     | BEBIDAS               | 1000       | \$ 1,00      |   |

Figura 20: Pantalla de Listado de productos

Elaborado por: Autores

## Creación y disponibilidad de Mesas

En esta sección se puede crear las mesas que tendrá el Restaurant.



DISPONIBILIDAD DE MESAS Nuevo

**NUEVA MESA**

Descripción

Observación

Estado

LIMPIAR GUARDAR

Figura 21: Creación de mesas

Elaborado por: Autores



DISPONIBILIDAD DE MESAS Nuevo Vista

| ESTADO  | MESA       | OBSERVACIÓN |
|---------|------------|-------------|
| OCUPADA | Mesa 1     |             |
| OCUPADA | Mesa 2     |             |
| OCUPADA | Mesa 3     |             |
| LIBRE   | Mesa 4     |             |
| OCUPADA | Mesa 5     |             |
| OCUPADA | Mesa 6     |             |
| LIBRE   | Mesa Alfa  |             |
| LIBRE   | Mesa VIP-1 |             |
| LIBRE   | Mesa VIP-2 |             |

Figura 22: Disponibilidad de mesas

Elaborado por: Autores

## Visualización de Mesas

En esta sección se puede Visualizar la disponibilidad de las mesas además se puede observar una gráfica de cómo están organizadas las mesas y se puede cambiar el orden de las mismas.

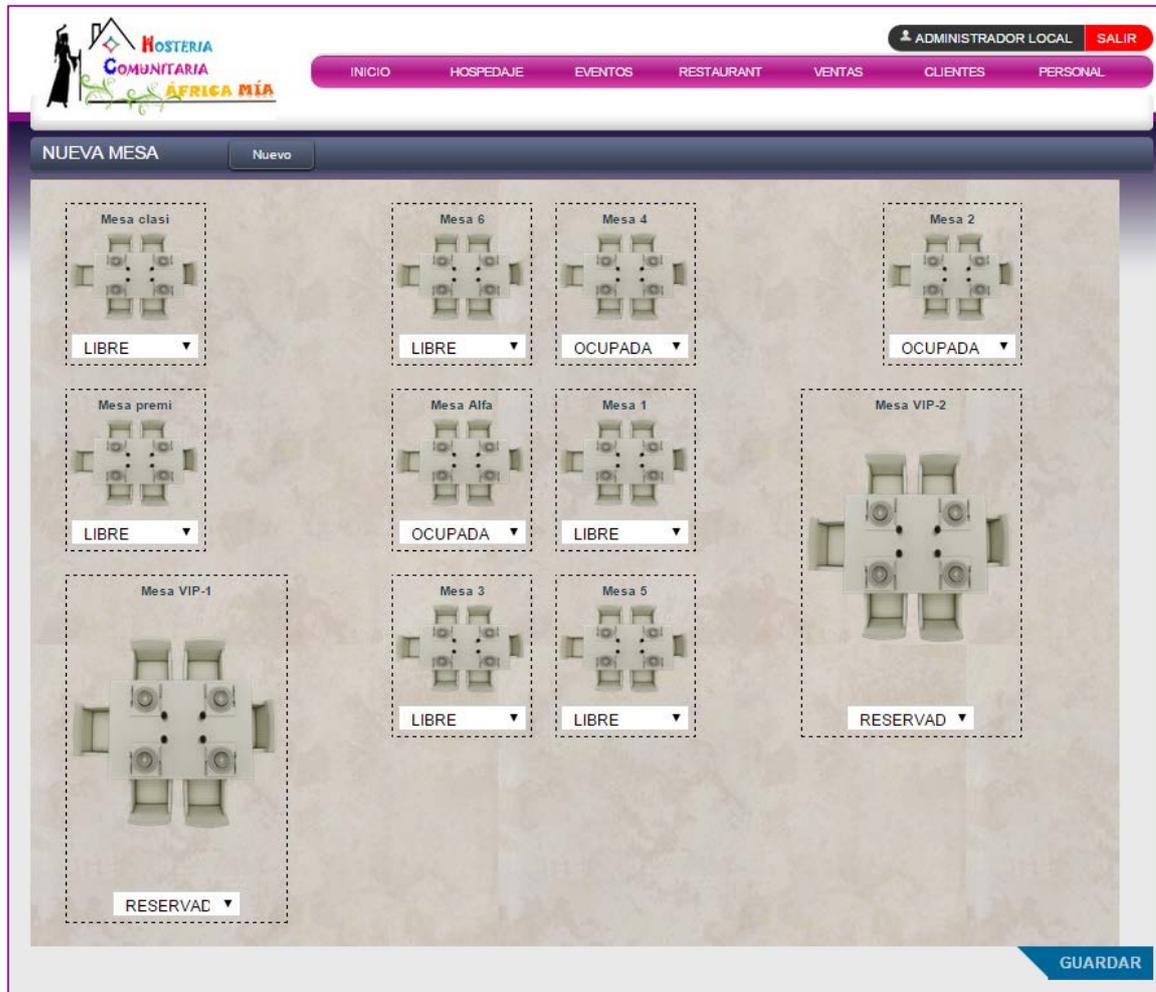
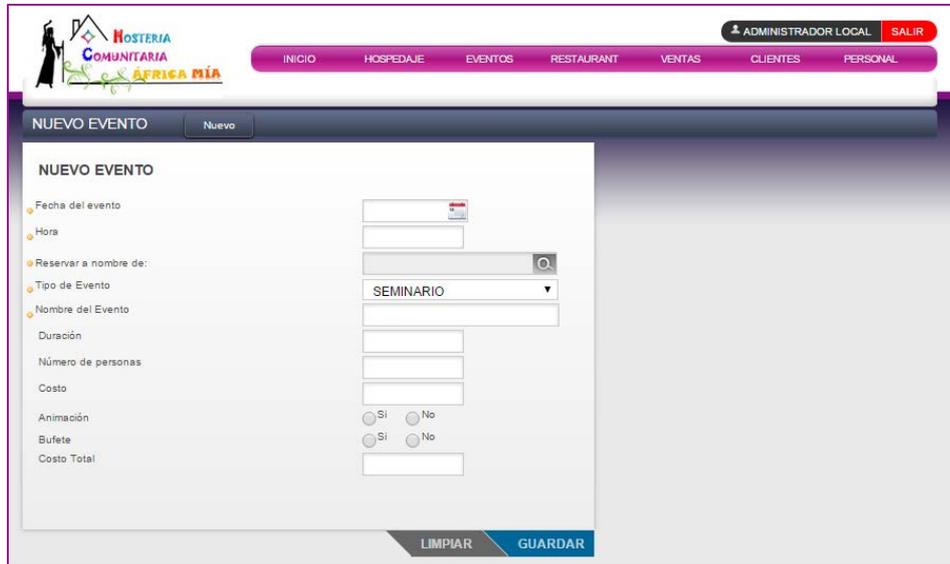


Figura 23: Vista de estado de las mesas

Elaborado por: Autores

## Creación de Eventos

En esta sección se puede ingresar los eventos a realizarse en el salón de eventos de la Hostería Comunitaria África Mía.



The screenshot shows a web interface for creating a new event. The header includes the logo for 'HOSTERIA COMUNITARIA AFRICA MIA' and a navigation menu with options: INICIO, HOSPEDAJE, EVENTOS, RESTAURANT, VENTAS, CLIENTES, PERSONAL. A user role indicator shows 'ADMINISTRADOR LOCAL' and a 'SALIR' button. The main form is titled 'NUEVO EVENTO' and contains the following fields:

- Fecha del evento: Date input field.
- Hora: Time input field.
- Reservar a nombre de: Searchable text input.
- Tipo de Evento: Dropdown menu with 'SEMINARIO' selected.
- Nombre del Evento: Text input field.
- Duración: Text input field.
- Número de personas: Text input field.
- Costo: Text input field.
- Animación: Radio buttons for 'Si' and 'No'.
- Bufete: Radio buttons for 'Si' and 'No'.
- Costo Total: Text input field.

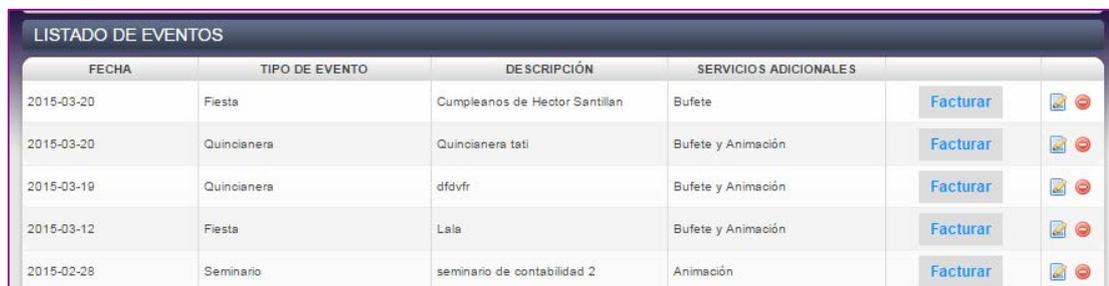
At the bottom of the form are two buttons: 'LIMPIAR' (Clear) and 'GUARDAR' (Save).

Figura 24: Pantalla de creación de eventos

Elaborado por: Autores

## Listado de Eventos

En esta sección se puede Visualizar los eventos a realizarse en el salón de eventos de la Hostería Comunitaria África Mía así como también se puede editar o eliminar los mismos.



| FECHA      | TIPO DE EVENTO | DESCRIPCIÓN                    | SERVICIOS ADICIONALES |  |
|------------|----------------|--------------------------------|-----------------------|--|
| 2015-03-20 | Fiesta         | Cumpleaños de Hector Santillan | Bufete                | Facturar   |
| 2015-03-20 | Quincianera    | Quincianera tati               | Bufete y Animación    | Facturar   |
| 2015-03-19 | Quincianera    | dfdvfr                         | Bufete y Animación    | Facturar   |
| 2015-03-12 | Fiesta         | Lala                           | Bufete y Animación    | Facturar   |
| 2015-02-28 | Seminario      | seminario de contabilidad 2    | Animación             | Facturar   |

Figura 25: Pantalla de listado de eventos

Elaborado por: Autores

## Nueva Factura

En esta sección se puede generar nuevas facturas.

Figura 26: Pantalla de Nueva factura

Elaborado por: Autores

## Listado de Facturas

En esta sección se puede crear facturas además se pueden realizar búsquedas por número de factura, por tipo de reporte y por fecha.

| FECHA       | NUM. FACTURA      | CLIENTE  | ESTADO    | SUBTOTAL  | DESCUENTO | IVA       | TOTAL USD |
|-------------|-------------------|--|-----------|-----------|-----------|-----------|-----------|
| 2015-Mar-23 | 001-001-000000017 | Rosario Tijeras<br>Pago: TARJETA CREDITO         | CANCELADA | \$ 28,80  | \$ 0,00   | \$ 3,46   | \$ 32,26  |
|             |                   |  |           | TOTAL IVA |           | TOTAL MES | \$ 32,26  |
| 2015-Mar-19 | 001-001-000000016 | Mariana Pintado<br>Pago: EFECTIVO                | CANCELADA | \$ 25,00  | \$ 0,00   | \$ 3,00   | \$ 28,00  |
| 2015-Mar-19 | 001-001-000000015 | Ricardo Francisco Mesias Ochoa<br>Pago: EFECTIVO | CANCELADA | \$ 2,90   | \$ 0,00   | \$ 0,35   | \$ 3,25   |

Figura 27: Listado de Facturas

Elaborado por: Autores

## Nuevo Servicio

Esta sección permite crear un nuevo servicio.

The screenshot shows a web interface for creating a new service. At the top, there is a navigation menu with items: INICIO, HOSPEDAJE, EVENTOS, RESTAURANT, VENTAS, CLIENTES, PERSONAL. A user profile dropdown shows 'ADMINISTRADOR LOCAL' and a 'SALIR' button. The main content area is titled 'NUEVO SERVICIO' and contains a form with the following fields:

- Descripción: A single-line text input field.
- Descripción Larga: A multi-line text area.
- Precio: A single-line text input field.

At the bottom of the form, there are two buttons: 'LIMPIAR' (Clear) and 'GUARDAR' (Save).

Figura 28: Pantalla de Creación de nuevo servicio

Elaborado por: Autores

## Servicios

En esta sección se pueden visualizar los servicios, el código y su precio también se pueden modificar y eliminar cualquier servicio según se requiera.

The screenshot shows a table titled 'LISTADO DE SERVICIOS'. The table has the following columns: 'CÓDIGO', 'SERVICIO', and 'PRECIO'. Each row also includes a small icon for editing and a red circle with a minus sign for deleting the service.

| CÓDIGO | SERVICIO   | PRECIO    |  |
|--------|--|-----------|--|
| 20     | Collar afro<br>collar con perlas esmeraldas  | \$ 25,00  |  |
| 15     | Servicio Alquiler de Instalaciones por evento<br>Servicio Alquiler de Instalaciones por evento | ---       |  |
| 17     | Servicio de buffete<br>Servicio de buffete   | \$ 240,00 |  |
| 16     | Servicio de habitacion<br>Servicio de habitacion 5 estrellas                                   | \$ 34,00  |  |
|        |  |           |  |

Figura 29: Pantalla de listado de servicios

Elaborado por: Autores

## Nuevo Cliente

En esta sección se puede crear un nuevo cliente.

The screenshot shows a web application interface for creating a new client. At the top, there is a navigation menu with options: INICIO, HOSPEDAJE, EVENTOS, RESTAURANT, VENTAS, CLIENTES, and PERSONAL. A user profile dropdown shows 'ADMINISTRADOR LOCAL' and a 'SALIR' button. The main content area is titled 'NUEVO CLIENTE' and contains the following form fields:

- Nombre
- Apellido
- Cédula
- Dirección
- Teléfono

At the bottom of the form are two buttons: 'LIMPIAR' and 'GUARDAR'.

Figura 30: Pantalla de creación de nuevo cliente

Elaborado por: Autores

## Listado de clientes

En esta sección se puede visualizar todos los clientes que han sido previamente registrados.

The screenshot displays the 'Listado de clientes' page. At the top, there is a search bar labeled 'STOCK' and 'PRODUCTO' with a 'Buscar' button. Below the search bar is a table listing registered clients. The table has the following columns: CLIENTE, CÉDULA, TELEFONOS, and DIRECCIÓN. Each row includes a small icon for editing or deleting the record.

| CLIENTE                        | CÉDULA         | TELEFONOS  | DIRECCIÓN                       |
|--------------------------------|----------------|------------|---------------------------------|
| Heinzel Zoller                 | 0931485288     | 2589932    | Orquideas                       |
| Johnny Jaramillo               | 032568945420   | 1213123112 | El oro y chimborazo             |
| Manuel Quinteros               | 0928382919     | 042348938  | Garzota II, Mz 53               |
| María Alcívar                  | 0928392019     | 0922837281 | El Oro y Chile                  |
| Mariana Pintado                | 099999999      |            | casa                            |
| Ricardo Francisco Mesías Ochoa | 0925064982     | 0988993770 | Cdla. 9 de Octubre Mz. 26 Sl. 8 |
| Rosario Tijeras                | 09776955444444 |            | 9 de Octubre y Boyacá           |
| Steffi Zoller                  | 0931485296     | 0982415374 | pradera 1                       |
| Tatiana Cacinelli              | 092887373848   | 9182938392 | Av. Américas                    |

Figura 31: Listado de los clientes

Elaborado por: Autores

## Personal, Meseros, Cajeros

Aquí se pueden ingresar los empleados de la hostería.

The screenshot shows a web application interface for managing employees. The main header includes the logo and navigation tabs: INICIO, HOSPEDAJE, EVENTOS, RESTAURANT, VENTAS, CLIENTES, PERSONAL. The 'PERSONAL' tab is active. Below the header, there's a section for 'EMPLEADOS' with a 'Nuevo' button. The 'NUEVO EMPLEADO' form is displayed, featuring several input fields: 'Nombres', 'Apellidos', 'Cargo' (a dropdown menu currently showing 'Administrador', 'Cajero', 'Despachador', and 'Mesero'), 'Cédula', 'Teléfono', and 'Dirección'. At the bottom of the form are 'LIMPIAR' and 'GUARDAR' buttons.

Figura 32: Pantalla de creación de empleados

Elaborado por: Autores

## Listado de empleados

Aquí se pueden visualizar los empleados que hay en la hostería

The screenshot displays a list of employees under the 'CAJEROS' category. The interface includes a 'Nuevo' button and a search bar labeled 'Empleados' with a 'Buscar' button. The table below has the following data:

| NOMBRES Y APELLIDOS | CARGO  | CÉDULA     | TELEFONO   | DIRECCION                 |
|---------------------|--------|------------|------------|---------------------------|
| Juan Macias         | cajero | 0999999998 | 0990043796 | Barrio Cuba               |
| Roberto Espinel     | cajero | 0924563314 | 2568863    | Chambers 227 y 5 de junio |

Figura 33: Listado de empleados

Elaborado por: Autores

The screenshot displays a list of employees under the 'EMPLEADOS' category. The interface includes a 'Nuevo' button and a search bar labeled 'Empleados' with a 'Buscar' button. The table below has the following data:

| NOMBRES Y APELLIDOS | CARGO       | CÉDULA     | TELEFONO   | DIRECCION                 |
|---------------------|-------------|------------|------------|---------------------------|
| Dario Vasconez      | despachador | 0952415895 | 24589624   | Cda Guayacanes            |
| José Cárdenas       | mesero      | 0928374882 | 0982837483 |                           |
| Juan Macias         | cajero      | 0999999998 | 0990043796 | Barrio Cuba               |
| Raúl Cevilla        | mesero      | 0039288382 | 0982733333 | Av. Quito y El Oro        |
| Ricardo Arjona      | despachador | 0999999999 | 0990043799 | Las Vegas                 |
| Roberto Espinel     | cajero      | 0924563314 | 2568863    | Chambers 227 y 5 de junio |
| Ricardo Arjona      | despachador | 0999999999 | 0990043799 | Las Vegas                 |

Figura 34: Listado de empleados

Elaborado por: Autores

## **CAPITULO 6**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **6.1. CONCLUSIONES**

En la actualidad se cuenta con gran diversidad de emprendimientos tecnológicos que cubren muchas de las necesidades de las empresas hoy en día, desde migraciones de sistemas ya establecidos a nuevas plataformas con estándares actuales, hasta la creación de nuevos aplicativos que solucionen o complementen las necesidades dentro de empresas que ya tienen establecidos sus procedimientos en base a la experiencia.

La experiencia es una clave fundamental en la creación de herramientas tecnológicas puesto que estas se crean en base a procedimientos ya definidos y cuando no es así se recurre a la adquisición de herramientas especializadas que aporten una visión al modelo de negocio que se desea emprender.

Se puede comprobar que para el desarrollo de este proyecto se tuvo que pensar abiertamente en ambas opciones debido a que la organización cuenta con un proyecto en desarrollo donde existen muchas variables al momento de tratar de cubrir todos los servicios por medio del uso de un software.

Conocer cada procedimiento y como solucionarlo es parte fundamental, es por esto que buscamos solucionar en un alto grado la integración de cada uno de estos servicios.

Esperamos poder ser parte esencial en los pilares base de este proyecto de servicios comunitarios y con la ayuda de la herramienta puedan desenvolverse tal cual lo planificado.

## 6.2. RECOMENDACIONES

Ahora que la organización posee una herramienta sólida que le permitirá gestionar algunos de sus servicios y atención al cliente es importante también destacar algunas recomendaciones por lo que detallamos las siguientes:

- Definir y establecer roles para las personas que se encargarán de atender cada uno de los servicios tales como el restaurante, eventos y hospedaje puesto que en el tiempo en el que se realizó el levantamiento de la información estos cargos no estaban definidos y por lo cual no fue posible capacitar al personal adecuado para las tareas en función.
- Adquirir los equipos con requerimientos de hardware específicos para la ejecución del aplicativo recomendados en nuestra propuesta ya que así se evitarán futuros contratiempos en vista de que su proyecto está en fases iniciales.
- Al llegar a tener el personal adecuado en cada área, ya establecido y haciendo un uso avanzado de nuestra herramienta, pudiendo ser a corto o mediano plazo, realizar un segundo análisis sobre los procedimientos definidos en este aplicativo ya que de esta forma se logran identificar algunas mejoras prudentes para el crecimiento de cada una de las áreas de su organización.
- Teniendo en cuenta que por medio del aplicativo se pueden realizar reservaciones de hospedaje y al mismo tiempo la infraestructura tecnológica usada orientada a la web, recomendaríamos la creación de una página web para que pueda integrar de forma automatizada los requerimientos de reservas de habitación e incluso de eventos. Además se puede ofrecer una cartilla web sobre los diferentes menús que se ofrecen en el Restaurant.

## REFERENCIAS

- Cibernética. (s.f.). Cibernética. Obtenido de  
[http://www.cibernetia.com/manuales/introduccion\\_aplicaciones\\_web/2\\_1\\_fundamentos\\_web.php](http://www.cibernetia.com/manuales/introduccion_aplicaciones_web/2_1_fundamentos_web.php)
- Constitución de la República del Ecuador. (2008)., (pág. 13).
- Ejecutivo, 3. D. (2012). Reglamento de actividades turísticas. Obtenido de  
<http://www.turismo.gob.ec/>: <http://www.turismo.gob.ec/wp-content/uploads/downloads/2014/02/Reglamento-Actividades-Turi%CC%81sticas.pdf>
- elwebmaster.com. (2007-2008).
- Henández, R. (s.f.). Metodología de la Investigación. Colombia: Mc Graw Hill.
- LIBROSWEB. (2015). Obtenido de [http://librosweb.es/libro/ajax/capitulo\\_1.html](http://librosweb.es/libro/ajax/capitulo_1.html)
- Peralta. (2008). Los Sistemas automatizados. Madrid: Anabad.
- Ruiz. (1999 - 2015). masadelante.com. Obtenido de  
[www.masadelante.com/faqs/intranet](http://www.masadelante.com/faqs/intranet)
- Sánchez, J. (2012). Obtenido de  
<http://www.jorgesanchez.net/programacion/manuales/sublimeText2.pdf>
- Vogelgesang. (2002). apachefriends.org. Obtenido de  
[www.apachefriends.org/es/index.html](http://www.apachefriends.org/es/index.html)
- Wales. (2001). Wikipedia. Obtenido de [es.wikipedia.org/wiki/HTML5](http://es.wikipedia.org/wiki/HTML5)

## ANEXOS

### **Anexo 1: La encuesta: Grupo afroecuatoriano y público en general**

OBJETIVO: Medir la aceptación del posible mercado objetivo, a quienes va orientado los resultados del proyecto titulado HOSTERIA COMUNITARIA DE LA AGRUPACION MUJERES PROGRESISTAS DE LA COOP. Independencia II, del Barrio Nigeria, sector Isla Trinitaria

Buenas días/tardes.

Somos estudiantes de la Universidad Politécnica Salesiana, estamos realizando una encuesta como parte de nuestro trabajo de titulación para obtener el título de Ingenieros en Sistemas. Agradeceríamos su gentil ayuda, llenando la presente encuesta para conocer qué tan aceptable es nuestro proyecto.

1. ¿Ha podido hospedarse en una hostería anteriormente?
2. ¿Pertenece a algún grupo étnico afroamericano en nuestro país?
3. ¿Qué es lo que más le agrada de los servicios que prestan las hosterías?
4. ¿Visitaría una hostería comunitaria afrodescendiente ubicada en la Isla Trinitaria, que ofrezca servicios de imagen o belleza, vestuario, platos típicos de la etnia, etc.?
5. ¿Le gustaría contar con un acceso en línea para ver la disponibilidad de un hostel en cuanto a su área de eventos, habitaciones, servicios y demás?
6. ¿Le gustaría tener información actualizada referente a los eventos a realizarse en la hostería en sus correos electrónicos y/o redes sociales?

## **Anexo 2: Guía diseñada: Grupo focal**

1. ¿Cuál es su nivel de conocimiento con respecto a los sistemas computarizados?
2. ¿Cuáles son sus necesidades en cuanto al área en que se desenvuelve?
3. ¿Ha utilizado sistemas operativos anteriormente en algún trabajo?
4. ¿Cree factible el uso de un software que facilite sus funciones en cuanto a la gestión y administración de la hostería?

### **Anexo 3: La entrevista: Experto en hostería**

#### ENTREVISTA No. 1

**OBJETIVO:** Recopilar información con el fin de conocer el grado de aceptabilidad para la adaptación de un Software inclinada a los servicios de la **HOSTERIA COMUNITARIA DE LA AGRUPACION MUJERES PROGRESISTAS DE LA COOP. Independencia II, del Barrio Nigeria, sector Isla Trinitaria**

1. ¿Nos podría contar como funciona su negocio de Hostería actualmente?
  
2. ¿Qué opina Usted sobre la utilización de un software de gestión para los servicios y productos de la hostería?
  
3. ¿Qué piensa Usted de la creación de un software para la gestión de servicios en la hostería?
  
4. ¿Nos podría mencionar las actividades y procesos que realiza en la hostería?
  
5. ¿Le gustaría diseñar alguna estrategia para que el software a desarrollar tenga un buen uso?

## Anexo 4: Diccionario de datos

A continuación se adhiere el diccionario de datos de las tablas y respectivos campos que intervienen en la solución del Aplicativo Web.

**Tabla: app\_cargo**

| COLUMNA     | TIPO        | NULO | PREDETERMINADO    | COMENTARIOS |
|-------------|-------------|------|-------------------|-------------|
| id_cargo    | int(4)      | No   |                   |             |
| Descripción | varchar(50) | No   |                   |             |
| Flag        | tinyint(1)  | Sí   | 1                 |             |
| fecha_sys   | timestamp   | Sí   | CURRENT_TIMESTAMP |             |

**Tabla: app\_cliente**

| COLUMNA    | TIPO         | NULO | PREDETERMINADO    | COMENTARIOS |
|------------|--------------|------|-------------------|-------------|
| id_cliente | int(11)      | No   |                   |             |
| nombre     | varchar(50)  | No   |                   |             |
| apellido   | varchar(50)  | No   |                   |             |
| Dni        | varchar(14)  | No   |                   |             |
| direccion  | varchar(200) | Sí   | NULL              |             |
| telefono   | varchar(20)  | Sí   | NULL              |             |
| flag       | tinyint(1)   | Sí   | 1                 |             |
| fecha_sys  | timestamp    | Sí   | CURRENT_TIMESTAMP |             |

**Tabla: app\_empleado**

| COLUMNA     | TIPO        | NULO | PREDETERMINADO    | COMENTARIOS |
|-------------|-------------|------|-------------------|-------------|
| id_empleado | int(11)     | No   |                   |             |
| id_cargo    | int(4)      | No   |                   |             |
| nombre      | varchar(50) | No   |                   |             |
| apellido    | varchar(50) | No   |                   |             |
| cedula      | varchar(10) | Sí   | NULL              |             |
| direccion   | varchar(20) | Sí   | NULL              |             |
| telefono    | varchar(20) | Sí   | NULL              |             |
| Flag        | tinyint(1)  | Sí   |                   | 1           |
| fecha_sys   | timestamp   | Sí   | CURRENT_TIMESTAMP |             |

**Tabla: app\_menu**

| COLUMNA     | TIPO         | NULO | PREDETERMINADO    | COMENTARIOS |
|-------------|--------------|------|-------------------|-------------|
| id_menu     | int(11)      | No   |                   |             |
| descripcion | varchar(100) | No   |                   |             |
| Flag        | tinyint(1)   | Sí   |                   | 1           |
| fecha_sys   | timestamp    | Sí   | CURRENT_TIMESTAMP |             |

**Tabla: app\_evento**

| COLUMNA        | TIPO         | NULO | PREDETERMINADO | COMENTARIOS |
|----------------|--------------|------|----------------|-------------|
| id_evento      | int(11)      | No   |                |             |
| id_cliente     | int(11)      | No   |                |             |
| id_tipo_evento | int(11)      | No   |                |             |
| descripcion    | varchar(100) | No   |                |             |
| animacion      | int(1)       | No   |                |             |
| bufete         | int(1)       | No   |                |             |
| costo          | double(9,2)  | No   |                |             |
| fecha          | date         | No   |                |             |
| flag           | int(11)      | No   |                | 1           |
| duracion       | int(11)      | No   |                |             |
| num_personas   | int(11)      | No   |                |             |

**Tabla: app\_habitacion**

| COLUMNA           | TIPO  | NULO | PREDETERMINADO | COMENTARIOS |
|-------------------|---|------|----------------|-------------|
| id_habitacion     | int(11)   | No   |                |             |
| nombre_habitacion | varchar(50)   | No   |                |             |
| num_piso          | int(11)   | No   |                |             |
| tipo_habitacion   | enum('sencilla', 'doble', 'matrimonio', 'familiar') | No   |                | sencilla    |
| num_baños         | int(11)   | No   |                |             |
| telefono          | tinyint(1)  | No   |                |             |

|                    |   |    |       |   |
|--------------------|---|----|-------|---|
| aire_acondicionado | tinyint(1)                                  | No |       |   |
| ventilador         | tinyint(1)                                  | No |       |   |
| ventana            | tinyint(1)                                  | No |       |   |
| television         | tinyint(1)                                  | No |       |   |
| wifi               | tinyint(1)                                  | No |       |   |
| tv_pagada          | tinyint(1)                                  | No |       |   |
| estado             | enum('libre',<br>'ocupada',<br>'reservada') | No | libre |   |
| id_imagen          | int(11)                                     | No |       |   |
| flag               | int(1)                                      | No |       | 1 |
| costo              | decimal(9,2)<br>)                           | No |       |   |

---

**Tabla: app\_mesa**

| COLUMNA        | TIPO        | NULO | PREDETERMINADO    | COMENTARIOS |
|----------------|-------------|------|-------------------|-------------|
| id_mesa        | int(11)     | No   |                   |             |
| id_mesa_estado | int(4)      | Sí   |                   | 1           |
| descripcion    | varchar(10) | No   |                   |             |
| observacion    | varchar(30) | Sí   | NULL              |             |
| flag           | tinyint(1)  | Sí   |                   | 1           |
| fecha_sys      | timestamp   | Sí   | CURRENT_TIMESTAMP |             |

---

**Tabla: app\_mesa**

| COLUMNA        | TIPO        | NULO | PREDETERMINADO | COMENTARIOS       |
|----------------|-------------|------|----------------|-------------------|
| id_mesa_estado | int(4)      | No   |                |                   |
| descripcion    | varchar(50) | No   |                |                   |
| flag           | tinyint(1)  | Sí   |                | 1                 |
| fecha_sys      | timestamp   | Sí   |                | CURRENT_TIMESTAMP |

**Tabla: app\_producto**

| COLUMNA           | TIPO                         | NULO | PREDETERMINADO | COMENTARIOS       |
|-------------------|------------------------------|------|----------------|-------------------|
| id_producto       | int(11)                      | No   |                |                   |
| id_menu           | int(11)                      | Sí   |                | NULL              |
| descripcion       | varchar(100)                 | Sí   |                | NULL              |
| descripcion_larga | varchar(400)                 | Sí   |                | NULL              |
| precio            | float                        | Sí   |                | NULL              |
| stock             | int(11)                      | Sí   |                | NULL              |
| tipo              | enum('solido',<br>'liquido') | Sí   |                | solido            |
| ivr               | tinyint(1)                   | Sí   |                | 0                 |
| flag              | tinyint(1)                   | Sí   |                | 1                 |
| fecha_sys         | timestamp                    | Sí   |                | CURRENT_TIMESTAMP |

**Tabla: app\_registro\_habitacion**

| COLUMNA                | TIPO    | NULO | PREDETERMINADO | COMENTARIOS |
|------------------------|---------|------|----------------|-------------|
| id_registro_habitacion | int(11) | No   |                |             |
| id_cliente             | int(11) | No   |                |             |

|               |             |    |                   |
|---------------|-------------|----|-------------------|
| num_dias      | int(11)     | No |                   |
| num_personas  | int(11)     | No |                   |
| forma_pago    | varchar(50) | No |                   |
| fecha_inicio  | date        | No |                   |
| fecha_fin     | date        | No |                   |
| fecha_reserva | date        | No |                   |
| id_habitacion | int(11)     | No |                   |
| estado        | varchar(50) | No |                   |
| flag          | int(11)     | No | 1                 |
| fecha_sys     | timestamp   | No | CURRENT_TIMESTAMP |

---

**Tabla: app\_registro\_reserva**

| COLUMNA                | TIPO    | NULO | PREDETERMINADO | COMENTARIOS |
|------------------------|---------|------|----------------|-------------|
| id_registro_habitacion | int(11) | No   |                |             |
| id_reserva_habitacion  | int(11) | No   |                |             |

---

**Tabla: app\_reserva\_habitación**

| COLUMNA               | TIPO                                    | NULO | PREDETERM.        | COMENT |
|-----------------------|---|------|-------------------|--------|
| id_reserva_habitacion | int(11)                                 | No   |                   |        |
| num_dias              | int(11)                                 | No   |                   |        |
| num_personas          | int(11)                                 | No   |                   |        |
| forma_pago            | enum('tarjeta de credito', 'efectivo')  | No   | efectivo          |        |
| fecha_inicio          | datetime                                | No   |                   |        |
| fecha_fin             | datetime                                | No   |                   |        |
| id_habitacion         | int(11)                                 | No   |                   |        |
| flag                  | int(1)                                  | No   |                   | 1      |
| fecha_sys             | timestamp                               | No   | CURRENT_TIMESTAMP |        |
| estado                | enum('Creada', 'Despachada', 'Anulada') | No   | Creada            |        |
| id_cliente            | int(11)                                 | No   |                   |        |

**Tabla: app\_tipo\_evento**

| COLUMNA        | TIPO        | NULO | PREDETERMINADO    | COMENTARIOS |
|----------------|-------------|------|-------------------|-------------|
| id_tipo_evento | int(11)     | No   |                   |             |
| descripcion    | varchar(20) | No   |                   |             |
| flag           | int(11)     | No   |                   | 1           |
| fecha_sys      | timestamp   | No   | CURRENT_TIMESTAMP |             |

**Tabla: cont\_fatura**

| COLUMNA           | TIPO       | NULO | PREDETERMINADO      | COMENTARIOS    |
|-------------------|------------|------|---------------------|----------------|
| id_factura        | int(11)    | No   |                     |                |
| id_vendedor       | int(11)    | No   |                     | ID de Empleado |
| id_cliente        | int(11)    | No   |                     |                |
| id_mesa           | int(4)     | No   |                     |                |
| id_pago_forma     | int(4)     | No   |                     |                |
| id_factura_estado | tinyint(3) | No   |                     |                |
| num_referencia    | varchar(7) | No   | 000-000             |                |
| numero            | int(9)     | No   |                     | 0              |
| fecha             | timestamp  | No   | 0000-00-00 00:00:00 |                |
| subtotal          | double     | No   |                     | 0              |
| descuento         | float      | Sí   |                     | 0              |
| iva               | float      | Sí   |                     | 0              |
| total             | double     | Sí   | NULL                |                |
| flag              | tinyint(1) | Sí   |                     | 1              |
| fecha_sys         | timestamp  | Sí   | CURRENT_TIMESTAMP   |                |

**Tabla: cont\_factura\_detalle**

| COLUMNA            | TIPO       | NULO | PREDETERM. | COMENT. |
|--------------------|------------|------|------------|---------|
| id_factura_detalle | int(11)    | No   |            |         |
| id_factura         | int(11)    | No   |            |         |
| id_item            | int(11)    | No   |            |         |
| cantidad           | int(10)    | No   |            | 1       |
| precio             | float      | No   |            |         |
| subtotal           | float      | No   |            |         |
| flag               | tinyint(1) | Sí   |            | 1       |

|           |           |    |                   |
|-----------|-----------|----|-------------------|
| fecha_sys | timestamp | Sí | CURRENT_TIMESTAMP |
|-----------|-----------|----|-------------------|

---

**Tabla: cont\_factura\_estado**

| COLUMNA           | TIPO        | NULO | PREDETERMINADO | COMENTARIOS |
|-------------------|-------------|------|----------------|-------------|
| id_factura_estado | tinyint(3)  | No   |                |             |
| descripcion       | varchar(50) | No   |                |             |
| flag              | tinyint(1)  | Sí   |                | 1           |

---

**Tabla: cont\_pago\_forma**

| COLUMNA       | TIPO        | NULO | PREDETERMINADO    | COMENTARIOS |
|---------------|-------------|------|-------------------|-------------|
| id_pago_forma | int(4)      | No   |                   |             |
| descripcion   | varchar(50) | No   |                   |             |
| flag          | tinyint(1)  | Sí   |                   | 1           |
| fecha_sys     | timestamp   | Sí   | CURRENT_TIMESTAMP |             |

---

**Tabla: web\_modulo**

| COLUMNA     | TIPO        | NULO | PREDETERMINADO    | COMENTARIOS |
|-------------|-------------|------|-------------------|-------------|
| id_modulo   | int(11)     | No   |                   |             |
| descripcion | varchar(50) | No   |                   |             |
| activo      | tinyint(1)  | No   |                   | 1           |
| flag        | tinyint(1)  | Sí   |                   | 1           |
| fec_sys     | timestamp   | Sí   | CURRENT_TIMESTAMP |             |

---

**Tabla: web\_modulo\_detalle**

| COLUMNA           | TIPO         | NULO | PREDETERM.        | COMENT. |
|-------------------|--------------|------|-------------------|---------|
| id_modulo_detalle | int(11)      | No   |                   |         |
| id_modulo         | int(11)      | No   |                   |         |
| link_name         | varchar(50)  | No   |                   |         |
| descripcion       | varchar(255) | No   |                   |         |
| ruta              | varchar(255) | No   |                   |         |
| url_parse         | tinyint(1)   | No   |                   | 0       |
| orden             | tinyint(4)   | Sí   | NULL              |         |
| underline         | tinyint(1)   | No   |                   | 0       |
| flag              | tinyint(1)   | Sí   |                   | 1       |
| fec_sys           | timestamp    | Sí   | CURRENT_TIMESTAMP |         |

**Tabla: web\_modulo\_plantilla**

| COLUMNA             | TIPO       | NULO | PREDETERM.        | COMEN. |
|---------------------|------------|------|-------------------|--------|
| id_modulo_plantilla | int(11)    | No   |                   |        |
| id_modulo           | int(11)    | No   |                   |        |
| id_modulo_detalle   | int(11)    | No   |                   |        |
| id_plantilla        | tinyint(4) | No   |                   |        |
| activo              | tinyint(1) | No   |                   | 1      |
| flag                | tinyint(1) | Sí   |                   | 1      |
| fec_sys             | timestamp  | Sí   | CURRENT_TIMESTAMP |        |

**Tabla: web\_permiso**

| COLUMNA      | TIPO       | NULO | PREDETER.         | COMEN. |
|--------------|------------|------|-------------------|--------|
| id_permiso   | int(11)    | No   |                   |        |
| id_usuario   | int(11)    | No   |                   |        |
| id_plantilla | tinyint(4) | No   |                   |        |
| activo       | tinyint(1) | No   |                   | 1      |
| flag         | tinyint(1) | Sí   |                   | 1      |
| fec_sys      | timestamp  | Sí   | CURRENT_TIMESTAMP |        |

**Tabla: web\_permiso\_planitlla**

| COLUMNA         | TIPO            | NULO | PREDETERM.        | COMEN |
|-----------------|-----------------|------|-------------------|-------|
| id_plantilla    | tinyint(4)      | No   |                   |       |
| id_usuario_tipo | tinyint(4)      | No   |                   |       |
| descripcion     | varchar(50<br>) | No   |                   |       |
| activo          | tinyint(1)      | No   |                   | 1     |
| flag            | tinyint(1)      | Sí   |                   | 1     |
| fec_sys         | timestamp       | Sí   | CURRENT_TIMESTAMP |       |

**Tabla: web\_usuario**

| COLUMNA         | TIPO            | NULO | PREDETERM. | COMENT |
|-----------------|-----------------|------|------------|--------|
| id_usuario      | int(4)          | No   |            |        |
| id_usuario_tipo | tinyint(4)      | No   |            |        |
| id_empleado     | int(11)         | No   |            |        |
| usuario         | varchar(50<br>) | No   |            |        |
| password        | varchar(32<br>) | No   |            |        |
| flag            | tinyint(1)      | Sí   |            | 1      |

|         |           |    |                   |
|---------|-----------|----|-------------------|
| fec_sys | timestamp | Sí | CURRENT_TIMESTAMP |
|---------|-----------|----|-------------------|

---

**Tabla: web\_usuario\_tipo**

| COLUMNA         | TIPO        | NULO | PREDETERM.        | COMEN. |
|-----------------|-------------|------|-------------------|--------|
| id_usuario_tipo | tinyint(4)  | No   |                   |        |
| descripcion     | varchar(50) | No   |                   |        |
| flag            | tinyint(1)  | Sí   |                   | 1      |
| fec_sys         | timestamp   | Sí   | CURRENT_TIMESTAMP |        |

---