**CENTRO DE ENSEÑANZA TÉCNICA INDUSTRIAL**

*Organismo Público Descentralizado Federal*



**Sistema de inventario en línea Stock-Web**

**INTEGRANTE(S)**

Ayala Castillo Enrique Paul - 13100034

**ASESOR(ES)**

Ferrer Hernández Susana Elizabeth

**GUADALAJARA, JAL.** Agosto - diciembre 2018

RESUMEN

Este documento contiene las características y todas las herramientas necesarias que se necesitaron para lo que fue la realización del proyecto “Stock-Web” el cual su principal objetivo es el de proporcionar una manera efectiva y completa de controlar el inventario de un negocio.

Así mismo se incluye una descripción de lo que es el sistema en general, en el cual habla sobre el problema que se desea solucionar, así como también una justificación de cómo esta plataforma puede ayudar a una determinada empresa en el manejo de su inventario.

Se muestran también varias imágenes ilustrativas sobre el diseño del sistema en su etapa de prototipo hasta las ilustraciones de como quedo en su versión final.

Al final del documento también incluye un manual de usuario que contiene, desde los recursos necesarios para el uso del sistema, hasta las diferentes acciones que los usuarios pueden usar dentro de la plataforma.

Tabla de contenido

Contenido

[I. INTRODUCCIÓN AL PROYECTO 18](#_Toc535413598)

[I.1 DEFINICIÓN DEL PROBLEMA 18](#_Toc535413599)

[I.2 OBJETIVOS Y ALCANCES DEL PROYECTO 19](#_Toc535413600)

[I.3 JUSTIFICACIÓN 19](#_Toc535413601)

[I.4 DESCRIPCIÓN DEL PRODUCTO 20](#_Toc535413602)

[I.5 LISTA DE REQUERIMIENTOS FUNCIONALES 22](#_Toc535413603)

[I.6 LISTA DE REQUERIMIENTOS NO FUNCIONALES 26](#_Toc535413604)

[I.7 COSTO DEL PROYECTO Y ESTIMACIÓN DE COSTOS DE MATERIALES 27](#_Toc535413605)

[II. ANTECEDENTES DEL PROYECTO 32](#_Toc535413606)

[III. PLANTEAMIENTO Y MODELOS DEL PROYECTO 35](#_Toc535413607)

[III.1 OPCIONES DE SOLUCIÓN 36](#_Toc535413608)

[III.2 EVALUACIÓN Y SELECCIÓN 36](#_Toc535413609)

[III.3 MODELOS 38](#_Toc535413610)

[IV. DESARROLLO Y EVALUACIÓN DEL PROYECTO 66](#_Toc535413611)

[IV.1 Clase para subir imágenes 67](#_Toc535413612)

[IV.2 Función para actualizar imagen de usuario 68](#_Toc535413613)

[IV.3 Función para borrar imagen de fondo 68](#_Toc535413614)

[IV.4 Función para borrar imagen de perfil de usuario 69](#_Toc535413615)

[IV.5 Función para borrar el logo de la empresa 69](#_Toc535413616)

[IV.6 Función para eliminar imagen de perfil del usuario 70](#_Toc535413617)

[IV.7 Función para eliminar la imagen de producto 71](#_Toc535413618)

[IV.8 Función para checar la extensión de una imagen 72](#_Toc535413619)

[IV.9 Función para insertar imagen de fondo 72](#_Toc535413620)

[IV.10 Función para insertar el logo de la empresa 73](#_Toc535413621)

[IV.11 Función para insertar imagen de producto 74](#_Toc535413622)

[IV.12 Función para subir una imagen al servidor 74](#_Toc535413623)

[IV.13 Función para subir el logo de la empresa al servidor 75](#_Toc535413624)

[IV.14 Función para subir una imagen de productos al servidor 76](#_Toc535413625)

[IV.15 Función para subir una imagen de perfil al servidor 77](#_Toc535413626)

[IV.16 Función para subir una imagen de fondo al servidor 79](#_Toc535413627)

[IV.17 Función para revisar la extensión de la imagen 80](#_Toc535413628)

[IV.18 Clase que administra la base de datos 81](#_Toc535413629)

[IV.19 Función constructora de la clase 81](#_Toc535413630)

[IV.20 Función para saber si hubo filas actualizadas 81](#_Toc535413631)

[IV.21 Función para conectarse a la base de datos 82](#_Toc535413632)

[IV.22 Función para desconectarse de la base de datos 83](#_Toc535413633)

[IV.23 Función para filtrar una cadena de texto de caracteres extraños 83](#_Toc535413634)

[IV.24 Función que retorna el resultado de una consulta en forma de arreglo 83](#_Toc535413635)

[IV.25 Función que retorna el resultado de una consulta en forma de arreglo asociativo 84](#_Toc535413636)

[IV.26 Función que retorna el resultado de una consulta en forma de arreglo de objetos 84](#_Toc535413637)

[IV.27 Función que inserta un registro en la base de datos 84](#_Toc535413638)

[IV.28 Función que retorna el número de registros de una consulta 85](#_Toc535413639)

[IV.29 Función que realiza una consulta 85](#_Toc535413640)

[IV.30 Función que retorna los valores de una consulta en forma de un arreglo asociativo 86](#_Toc535413641)

[IV.31 Clase que administra las sesiones 86](#_Toc535413642)

[IV.32 Función constructora de la clase 87](#_Toc535413643)

[IV.33 Función para cerrar la sesión del usuario actual 87](#_Toc535413644)

[IV.34 Función para revisar la sesión del usuario actual 87](#_Toc535413645)

[IV.35 Función que retorna el estado de la sesión del usuario actual 88](#_Toc535413646)

[IV.36 Función que crea una sesión para un usuario 88](#_Toc535413647)

[IV.37 Función que muestra un mensaje de error o informativo con formato 88](#_Toc535413648)

[IV.38 Función que muestra un mensaje de error o informativo con formato instantáneo 89](#_Toc535413649)

[IV.39 Funciones utilitarias 89](#_Toc535413650)

[IV.40 Función para contar registros 90](#_Toc535413651)

[IV.41 Función para contar varios registros 90](#_Toc535413652)

[IV.42 Función para recuperar la fecha actual 90](#_Toc535413653)

[IV.43 Función para leer la fecha 90](#_Toc535413654)

[IV.44 Función para mostrar un mensaje con estilos 91](#_Toc535413655)

[IV.45 Función para calcular el precio total 91](#_Toc535413656)

[IV.46 Función para reemplazar el carácter “-“ con un espacio 92](#_Toc535413657)

[IV.47 Función para generar una cadena con caracteres aleatorios 92](#_Toc535413658)

[IV.48 Función para eliminar caracteres no convencionales de una cadena 93](#_Toc535413659)

[IV.49 Función para redirigir a otra página 93](#_Toc535413660)

[IV.50 Función para eliminar caracteres no convencionales de una consulta 94](#_Toc535413661)

[IV.51 Función para validar campos 94](#_Toc535413662)

[IV.52 Funciones SQL 95](#_Toc535413663)

[IV.53 Función para actualizar último login 95](#_Toc535413664)

[IV.54 Función para actualizar la cantidad de producto 96](#_Toc535413665)

[IV.55 Función para actualizar la llave primaria de la categoría de productos 97](#_Toc535413666)

[IV.56 Función para actualizar la llave primaria de la imagen de productos 97](#_Toc535413667)

[IV.57 Función para autenticar el inicio de sesión de un usuario 97](#_Toc535413668)

[IV.58 Función para borrar el rango de fecha de un reporte 99](#_Toc535413669)

[IV.59 Función para borrar un tipo de reporte 99](#_Toc535413670)

[IV.60 Función para borrar un registro por medio de su clave primaria 100](#_Toc535413671)

[IV.61 Función para buscar un rango de fecha 100](#_Toc535413672)

[IV.62 Función para buscar un tipo de reporte 101](#_Toc535413673)

[IV.63 Función para buscar tipos de reportes 101](#_Toc535413674)

[IV.64 Función para contar registros por medio de su clave primaria 102](#_Toc535413675)

[IV.65 Función para saber los valores de la empresa 103](#_Toc535413676)

[IV.66 Función para encontrar todos los registros de la tabla grupo 103](#_Toc535413677)

[IV.67 Función para encontrar todos los grupos con un determinado nivel 104](#_Toc535413678)

[IV.68 Función para encontrar todos los registros de la tabla categoría que coincidan con un nombre 104](#_Toc535413679)

[IV.69 Función para encontrar todos los registros de la tabla grupo que coincidan con un nombre 105](#_Toc535413680)

[IV.70 Función para encontrar todos los registros de la tabla producto que coincidan con un nombre 105](#_Toc535413681)

[IV.71 Función para encontrar todos los registros de la tabla usuario que coincidan con un nombre 106](#_Toc535413682)

[IV.72 Función para encontrar todos los registros de la tabla usuario que coincidan con un nombre 106](#_Toc535413683)

[IV.73 Función para encontrar todos los registros de una tabla de acuerdo con su llave primaria 107](#_Toc535413684)

[IV.74 Función para encontrar todos los registros de 2 tablas de acuerdo con una llave primaria 108](#_Toc535413685)

[IV.75 Función para encontrar determinados registros de la tabla usuario de acuerdo con una llave primaria 108](#_Toc535413686)

[IV.76 Función para encontrar un grupo por su nivel 109](#_Toc535413687)

[IV.77 Función para encontrar un grupo por su nombre 110](#_Toc535413688)

[IV.78 Función para encontrar por SQL una determinada consulta 110](#_Toc535413689)

[IV.79 Función para encontrar un producto por título 111](#_Toc535413690)

[IV.80 Función para encontrar todos los productos 111](#_Toc535413691)

[IV.81 Función para encontrar productos altamente vendidos 112](#_Toc535413692)

[IV.82 Función para encontrar productos con bajo inventario 112](#_Toc535413693)

[IV.83 Función para encontrar productos recientemente añadidos 113](#_Toc535413694)

[IV.84 Función para encontrar tipos de reporte por usuario 113](#_Toc535413695)

[IV.85 Función para encontrar todas las ventas 114](#_Toc535413696)

[IV.86 Función para encontrar todas las ventas agregadas recientemente 114](#_Toc535413697)

[IV.87 Función para encontrar todos los registros de una tabla 115](#_Toc535413698)

[IV.88 Función para encontrar todos los productos de acuerdo con un título 115](#_Toc535413699)

[IV.89 Función para encontrar todos los usuarios 116](#_Toc535413700)

[IV.90 Función para encontrar un usuario de acuerdo con el nombre 116](#_Toc535413701)

[IV.91 Función para encontrar ventas por un rango de fechas 117](#_Toc535413702)

[IV.92 Función para insertar un rango de fecha para un reporte 118](#_Toc535413703)

[IV.93 Función para insertar un tipo de reporte 118](#_Toc535413704)

[IV.94 Función que requiere un nivel de usuario para visualizarla 119](#_Toc535413705)

[IV.95 Función para restaurar un registro por su clave primaria 120](#_Toc535413706)

[IV.96 Función para saber si una tabla existe 121](#_Toc535413707)

[IV.97 Función que retorna el tema del usuario 122](#_Toc535413708)

[IV.98 Función que devuelve registros de la tabla producto uniendo con la tabla categoría y media 122](#_Toc535413709)

[IV.99 Función para unir la tabla de producto 123](#_Toc535413710)

[IV.100 Función para actualizar la imagen de perfil del usuario 123](#_Toc535413711)

[IV.101 Función para actualizar la imagen de fondo 124](#_Toc535413712)

[IV.102 Función para actualizar el lema de la empresa 124](#_Toc535413713)

[IV.103 Función para actualizar el logo de la empresa 125](#_Toc535413714)

[IV.104 Función para actualizar el tema del usuario 125](#_Toc535413715)

[IV.105 Función que retorna el usuario actual 126](#_Toc535413716)

[IV.106 Función que retorna las ventas diarias 126](#_Toc535413717)

[IV.107 Función que retorna las ventas mensuales 127](#_Toc535413718)

[IV.108 Funciones JQuery para agregar grupos 127](#_Toc535413719)

[IV.109 Función para revisar formulario de categoría 127](#_Toc535413720)

[IV.110 Funciones JQuery para agregar productos 129](#_Toc535413721)

[IV.111 Función que aplica márgenes a la ventana cuando se carga 129](#_Toc535413722)

[IV.112 Función que aplica márgenes al redimensionar la ventana 130](#_Toc535413723)

[IV.113 Función que revisa el formulario de producto 131](#_Toc535413724)

[IV.114 Funciones JQuery para agregar productos 132](#_Toc535413725)

[IV.115 Función que manda el formulario de productos 132](#_Toc535413726)

[IV.116 Funciones JQuery para autenticación al iniciar sesión 132](#_Toc535413727)

[IV.117 Función que autentica el usuario que inició sesión 133](#_Toc535413728)

[IV.118 Funciones JQuery para editar productos 133](#_Toc535413729)

[IV.119 Función que redimensiona la página al cargar la página 133](#_Toc535413730)

[IV.120 Función que cambia la dimensión de la página al redimensionar 134](#_Toc535413731)

[IV.121 Funciones JQuery utilitarias 135](#_Toc535413732)

[IV.122 Función que muestra un mensaje en forma de lista desplegable 135](#_Toc535413733)

[IV.123 Función que manda un formulario Ajax 136](#_Toc535413734)

[IV.124 Función que calcula el precio de venta total 137](#_Toc535413735)

[IV.125 Función que carga otras funciones al cargar la página 138](#_Toc535413736)

[IV.126 Funciones JQuery para agregar múltiples categorías 138](#_Toc535413737)

[IV.127 Función que carga otras funciones al cargar la página 139](#_Toc535413738)

[IV.128 Función para agregar múltiples categorías 140](#_Toc535413739)

[IV.129 Función para agregar un formulario de categoría con un botón 141](#_Toc535413740)

[IV.130 Función para eliminar un formulario de categoría con un botón 142](#_Toc535413741)

[IV.131 Función para mandar el o los formularios de categorías 142](#_Toc535413742)

[IV.132 Funciones JQuery para agregar múltiples productos. 143](#_Toc535413743)

[IV.133 Función que carga el resto de las funciones al cargar la página 143](#_Toc535413744)

[IV.134 Función que guarda los datos de los productos antes de mandarlos en el formulario 147](#_Toc535413745)

[IV.135 Función para mandar el o los productos por medio de los formularios 148](#_Toc535413746)

[IV.136 Función que elimina un producto de un formulario 148](#_Toc535413747)

[IV.137 Función que guarda los datos de los formularios de productos antes de enviarlos 149](#_Toc535413748)

[IV.138 Tabla Categoría 151](#_Toc535413749)

[IV.139 Campo Id (int) 151](#_Toc535413750)

[IV.140 Campo Nombre (varchar 50) 151](#_Toc535413751)

[IV.141 Tabla Empresa 151](#_Toc535413752)

[IV.142 Campo Id (int) 151](#_Toc535413753)

[IV.143 Campo Id\_usuario (int) 152](#_Toc535413754)

[IV.144 Campo Logo (varchar 255) 152](#_Toc535413755)

[IV.145 Campo Lema (varchar 255) 152](#_Toc535413756)

[IV.146 Tabla Grupo 152](#_Toc535413757)

[IV.147 Campo Id (int) 152](#_Toc535413758)

[IV.148 Campo Nombre (varchar 50) 152](#_Toc535413759)

[IV.149 Campo Nivel (int 1) 153](#_Toc535413760)

[IV.150 Campo Estado\_actual (int 1) 153](#_Toc535413761)

[IV.151 Tabla Media 153](#_Toc535413762)

[IV.152 Campo Id (int) 153](#_Toc535413763)

[IV.153 Campo Nombre\_archivo (varchar 255) 153](#_Toc535413764)

[IV.154 Campo Tipo\_archivo (varchar 100) 153](#_Toc535413765)

[IV.155 Tabla Producto 154](#_Toc535413766)

[IV.156 Campo Id (int) 154](#_Toc535413767)

[IV.157 Campo Nombre (varchar 255) 154](#_Toc535413768)

[IV.158 Campo Cantidad (int) 154](#_Toc535413769)

[IV.159 Campo Precio\_compra (decimal 25,2) 154](#_Toc535413770)

[IV.160 Campo Precio\_venta (decimal 25,2) 154](#_Toc535413771)

[IV.161 Campo Id\_categoria (int) 155](#_Toc535413772)

[IV.162 Campo Id\_media (int) 155](#_Toc535413773)

[IV.163 Campo Fecha\_actualizacion (datetime) 155](#_Toc535413774)

[IV.164 Tabla Producto\_eliminado 155](#_Toc535413775)

[IV.165 Campo Id (int) 155](#_Toc535413776)

[IV.166 Campo Nombre (varchar 255) 156](#_Toc535413777)

[IV.167 Campo Cantidad (int) 156](#_Toc535413778)

[IV.168 Campo Precio\_compra (decimal 25,2) 156](#_Toc535413779)

[IV.169 Campo Precio\_venta (decimal 25,2) 156](#_Toc535413780)

[IV.170 Campo Id\_categoria (int) 156](#_Toc535413781)

[IV.171 Campo Id\_media (int) 156](#_Toc535413782)

[IV.172 Campo Fecha\_actualizacion (datetime) 157](#_Toc535413783)

[IV.173 Tabla Producto\_respaldo 157](#_Toc535413784)

[IV.174 Campo Id (int) 157](#_Toc535413785)

[IV.175 Campo Nombre (varchar 255) 157](#_Toc535413786)

[IV.176 Campo Cantidad (int) 157](#_Toc535413787)

[IV.177 Campo Precio\_compra (decimal 25,2) 157](#_Toc535413788)

[IV.178 Campo Precio\_venta (decimal 25,2) 158](#_Toc535413789)

[IV.179 Campo Id\_categoria (int) 158](#_Toc535413790)

[IV.180 Campo Id\_media (int) 158](#_Toc535413791)

[IV.181 Campo Fecha\_actualizacion (datetime) 158](#_Toc535413792)

[IV.182 Tabla Rango\_fecha 158](#_Toc535413793)

[IV.183 Campo Id (int) 158](#_Toc535413794)

[IV.184 Campo Id\_tipos\_reporte (int) 159](#_Toc535413795)

[IV.185 Campo Fecha\_inicio (datetime) 159](#_Toc535413796)

[IV.186 Campo Fecha\_fin (datetime) 159](#_Toc535413797)

[IV.187 Tabla Reporte 159](#_Toc535413798)

[IV.188 Campo Id (int) 159](#_Toc535413799)

[IV.189 Campo Nombre (varchar 50) 159](#_Toc535413800)

[IV.190 Tabla Reporte\_usuario 160](#_Toc535413801)

[IV.191 Campo Id (int) 160](#_Toc535413802)

[IV.192 Campo Id\_usuario (int) 160](#_Toc535413803)

[IV.193 Campo Id\_reporte (int) 160](#_Toc535413804)

[IV.194 Tabla Tema 160](#_Toc535413805)

[IV.195 Campo Id (int) 161](#_Toc535413806)

[IV.196 Campo Nombre (varchar 50) 161](#_Toc535413807)

[IV.197 Tabla Tema\_personal 161](#_Toc535413808)

[IV.198 Campo Id (int) 161](#_Toc535413809)

[IV.199 Campo Id\_usuario (int) 161](#_Toc535413810)

[IV.200 Campo Id\_reporte (int) 162](#_Toc535413811)

[IV.201 Tabla Usuario 162](#_Toc535413812)

[IV.202 Campo Id (int) 162](#_Toc535413813)

[IV.203 Campo Id\_grupo\_usuario (int) 162](#_Toc535413814)

[IV.204 Campo Nombre (varchar 50) 162](#_Toc535413815)

[IV.205 Campo Username (varchar 50) 163](#_Toc535413816)

[IV.206 Campo Contrasena (varchar 255) 163](#_Toc535413817)

[IV.207 Campo Imagen (varchar 255) 163](#_Toc535413818)

[IV.208 Campo Imagen\_fondo (varchar 255) 163](#_Toc535413819)

[IV.209 Campo Estado (int 1) 163](#_Toc535413820)

[IV.210 Campo Ultimo\_acceso (datetime) 163](#_Toc535413821)

[IV.211 Tabla Venta 164](#_Toc535413822)

[IV.212 Campo Id (int) 164](#_Toc535413823)

[IV.213 Campo Id\_usuario (int) 164](#_Toc535413824)

[IV.214 Campo Id\_producto (int) 164](#_Toc535413825)

[IV.215 Campo Cantidad (int) 164](#_Toc535413826)

[IV.216 Campo Precio (decimal 25,2) 164](#_Toc535413827)

[IV.217 Campo Fecha (date) 165](#_Toc535413828)

[IV.218 Vistas complementarias 165](#_Toc535413829)

[IV.219 Cabecera de la aplicación web 165](#_Toc535413830)

[IV.220 Menú de usuarios administradores 168](#_Toc535413831)

[IV.221 Menú de usuarios especiales 170](#_Toc535413832)

[IV.222 Menú de usuarios estándares 172](#_Toc535413833)

[IV.223 Pie de la aplicación web 173](#_Toc535413834)

[IV.224 Vistas y archivos utilitarios 174](#_Toc535413835)

[IV.225 Panel de control de administración 174](#_Toc535413836)

[IV.226 Sección para agregar grupo 180](#_Toc535413837)

[IV.227 Sección para agregar múltiples registros 181](#_Toc535413838)

[IV.228 Sección para agregar producto 186](#_Toc535413839)

[IV.229 Sección para agregar usuario 189](#_Toc535413840)

[IV.230 Sección para agregar venta 191](#_Toc535413841)

[IV.231 Archivo para mandar datos de productos por medio de AJAX 194](#_Toc535413842)

[IV.232 Archivo para autenticar a los usuarios que inicien sesión 195](#_Toc535413843)

[IV.233 Archivo para borrar una categoría 197](#_Toc535413844)

[IV.234 Archivo para borrar un grupo 198](#_Toc535413845)

[IV.235 Archivo para borrar una imagen 199](#_Toc535413846)

[IV.236 Archivo para borrar un usuario 200](#_Toc535413847)

[IV.237 Archivo para borrar una venta 201](#_Toc535413848)

[IV.238 Archivo para buscar una categoría 202](#_Toc535413849)

[IV.239 Archivo para buscar un producto 205](#_Toc535413850)

[IV.240 Sección para cambiar contraseña 211](#_Toc535413851)

[IV.241 Sección principal 213](#_Toc535413852)

[IV.242 Sección de categorías 214](#_Toc535413853)

[IV.243 Archivo para cerrar la sesión de un usuario 217](#_Toc535413854)

[IV.244 Sección para editar categoría 217](#_Toc535413855)

[IV.245 Sección para editar cuenta 219](#_Toc535413856)

[IV.246 Sección para editar grupo 228](#_Toc535413857)

[IV.247 Sección para editar producto 231](#_Toc535413858)

[IV.248 Sección para editar usuario 235](#_Toc535413859)

[IV.249 Sección para editar venta 239](#_Toc535413860)

[IV.250 Archivo para eliminar un producto 242](#_Toc535413861)

[IV.251 Archivo para eliminar un producto definitivamente 243](#_Toc535413862)

[IV.252 Sección de grupos 244](#_Toc535413863)

[IV.253 Sección principal 246](#_Toc535413864)

[IV.254 Sección de imágenes de productos 249](#_Toc535413865)

[IV.255 Sección de perfil de usuario 252](#_Toc535413866)

[IV.256 Sección de productos 253](#_Toc535413867)

[IV.257 Sección de productos eliminados 255](#_Toc535413868)

[IV.258 Sección de reporte de ventas 257](#_Toc535413869)

[IV.259 Sección de reporte de ventas generado 260](#_Toc535413870)

[IV.260 Archivo para restaurar un producto 263](#_Toc535413871)

[IV.261 Sección de usuarios 265](#_Toc535413872)

[IV.262 Sección de ventas 267](#_Toc535413873)

[IV.263 Sección de ventas diarias 269](#_Toc535413874)

[IV.264 Sección de ventas mensuales 271](#_Toc535413875)

[IV.265 Archivo para verificar antes de agregar un grupo 273](#_Toc535413876)

[IV.266 Archivo para verificar antes de agregar un producto 275](#_Toc535413877)

[IV.267 Archivo para verificar antes de agregar usuario 276](#_Toc535413878)

[V. PROCESO DE EVALUACIÓN DE ALCANCE DE LOS OBJETIVOS 279](#_Toc535413879)

[VI. CONCLUSIONES Y COMENTARIOS FINALES 292](#_Toc535413880)

[VI.1 SITUACIÓN ACTUAL DEL PROYECTO. 293](#_Toc535413881)

[VI.2 FUTURO DEL PROYECTO. 293](#_Toc535413882)

[VI.3 PRINCIPALES INTERROGANTES POR RESOLVER EN ACTIVIDADES FUTURAS 293](#_Toc535413883)

[VI.4 LECCIONES APRENDIDAS SOBRE EL PROYECTO 293](#_Toc535413884)

[VI.5 LECCIONES APRENDIDAS SOBRE EL PROCESO DEL PROYECTO 295](#_Toc535413885)

[REFERENCIAS Y BIBLIOGRAFÍA 296](#_Toc535413886)

[Bibliografía 296](#_Toc535413887)

[APÉNDICE 297](#_Toc535413888)

[MANUAL DE USUARIO 1](#_Toc535413889)

[1.1. Descripción de la plataforma 3](#_Toc535413890)

[1.2. Guía de instalación 3](#_Toc535413891)

[1.3. Manual de usuario 3](#_Toc535413892)

[1.3.1 Bienvenida 3](#_Toc535413893)

[1.3.2 Ingreso a la aplicación 4](#_Toc535413894)

[1.3.3 Página principal 4](#_Toc535413895)

[1.3.4 Panel de control 6](#_Toc535413896)

[1.3.5 Sección de grupos 7](#_Toc535413897)

[1.3.6 Sección para editar grupo 8](#_Toc535413898)

[1.3.7 Sección para agregar grupo 9](#_Toc535413899)

[1.3.8 Sección de usuarios 10](#_Toc535413900)

[1.3.9 Sección para editar usuario 11](#_Toc535413901)

[1.3.10 Sección para agregar usuario 12](#_Toc535413902)

[1.3.11 Sección de categorías 13](#_Toc535413903)

[1.3.12 Sección de editar categoría 13](#_Toc535413904)

[1.3.13 Sección de productos 14](#_Toc535413905)

[1.3.14 Sección de editar producto 15](#_Toc535413906)

[1.3.15 Sección para agregar producto 16](#_Toc535413907)

[1.3.16 Sección de productos eliminados 17](#_Toc535413908)

[1.3.17 Sección de imágenes de productos 18](#_Toc535413909)

[1.3.18 Sección para agregar múltiples registros 19](#_Toc535413910)

[1.3.19 Sección de ventas 20](#_Toc535413911)

[1.3.20 Sección de editar venta 21](#_Toc535413912)

[1.3.21 Sección de agregar venta 22](#_Toc535413913)

[1.3.22 Sección de reporte de ventas 22](#_Toc535413914)

[1.3.23 Sección de reporte de ventas generado 23](#_Toc535413915)

[1.3.24 Sección de ventas mensuales 24](#_Toc535413916)

[1.3.25 Sección de ventas diarias 24](#_Toc535413917)

[1.3.26 Sección de perfil 25](#_Toc535413918)

[1.3.27 Sección de editar cuenta 26](#_Toc535413919)

[1.3.28 Sección para cambiar contraseña 27](#_Toc535413920)

[ANEXOS 29](#_Toc535413921)

Índices particulares

[Ilustración 1 Coeficientes de proyecto de software 28](#_Toc535413922)

[Ilustración 2 Tabla de conductores de coste 29](#_Toc535413923)

[Ilustración 3 Diagrama de entidad relación de la base de datos 39](#_Toc535413924)

[Ilustración 4 Diagrama de tablas de base de datos 44](#_Toc535413925)

[Ilustración 5 Interfaz de inicio de sesión 45](#_Toc535413926)

[Ilustración 6 Interfaz de página principal 46](#_Toc535413927)

[Ilustración 7 Interfaz de panel de control 47](#_Toc535413928)

[Ilustración 8 Interfaz de grupos 47](#_Toc535413929)

[Ilustración 9 Interfaz de agregar grupos 48](#_Toc535413930)

[Ilustración 10 Interfaz de editar grupos 49](#_Toc535413931)

[Ilustración 11 Interfaz de usuarios 50](#_Toc535413932)

[Ilustración 12 Interfaz de agregar usuarios 51](#_Toc535413933)

[Ilustración 13 Interfaz de editar usuarios 52](#_Toc535413934)

[Ilustración 14 Interfaz de categorías 53](#_Toc535413935)

[Ilustración 15 Interfaz de editar categoría 54](#_Toc535413936)

[Ilustración 16 Interfaz de productos 54](#_Toc535413937)

[Ilustración 17 Interfaz de agregar productos 55](#_Toc535413938)

[Ilustración 18 Interfaz de editar producto 56](#_Toc535413939)

[Ilustración 19 Interfaz de productos eliminados 57](#_Toc535413940)

[Ilustración 20 Interfaz de media 58](#_Toc535413941)

[Ilustración 21 Interfaz de agregar múltiples registros 59](#_Toc535413942)

[Ilustración 22 Interfaz de ventas 59](#_Toc535413943)

[Ilustración 23 Interfaz de agregar venta 60](#_Toc535413944)

[Ilustración 24 Interfaz de editar venta 61](#_Toc535413945)

[Ilustración 25 Interfaz de reporte de ventas 61](#_Toc535413946)

[Ilustración 26 Interfaz de ventas mensuales 62](#_Toc535413947)

[Ilustración 27 Interfaz de ventas diarias 63](#_Toc535413948)

[Ilustración 28 Interfaz de perfil 63](#_Toc535413949)

[Ilustración 29 Interfaz de editar perfil 64](#_Toc535413950)

[Ilustración 30 Interfaz de cambiar contraseña 65](#_Toc535413951)

[Ilustración 31 Cabecera 168](#_Toc535413952)

[Ilustración 32 Menú de usuarios administradores 170](#_Toc535413953)

[Ilustración 33 Menú de usuarios especiales 172](#_Toc535413954)

[Ilustración 34 Menú de usuarios estándares 173](#_Toc535413955)

[Ilustración 35 Panel de control de administración 179](#_Toc535413956)

[Ilustración 36 Sección para agregar grupo 181](#_Toc535413957)

[Ilustración 37 Sección para agregar múltiples registros 186](#_Toc535413958)

[Ilustración 38 Sección para agregar productos 189](#_Toc535413959)

[Ilustración 39 Sección para agregar usuario 191](#_Toc535413960)

[Ilustración 40 Sección para agregar venta 193](#_Toc535413961)

[Ilustración 41 Sección para cambiar contraseña 213](#_Toc535413962)

[Ilustración 42 Sección principal 214](#_Toc535413963)

[Ilustración 43 Sección de categorías 216](#_Toc535413964)

[Ilustración 44 Sección para agregar categoría 219](#_Toc535413965)

[Ilustración 45 Sección para editar cuenta 228](#_Toc535413966)

[Ilustración 46 Sección para editar grupo 231](#_Toc535413967)

[Ilustración 47 Sección para editar producto 235](#_Toc535413968)

[Ilustración 48 Sección para editar usuario 239](#_Toc535413969)

[Ilustración 49 Sección para editar venta 242](#_Toc535413970)

[Ilustración 50 Sección de grupos 246](#_Toc535413971)

[Ilustración 51 Sección de inicio de sesión 249](#_Toc535413972)

[Ilustración 52 Sección de imágenes de productos 251](#_Toc535413973)

[Ilustración 53 Sección de perfil de usuario 253](#_Toc535413974)

[Ilustración 54 Sección de productos 255](#_Toc535413975)

[Ilustración 55 Sección de productos eliminados 257](#_Toc535413976)

[Ilustración 56 Sección de reporte de ventas 259](#_Toc535413977)

[Ilustración 57 Sección de reporte de ventas generado 263](#_Toc535413978)

[Ilustración 58 Sección de usuarios 267](#_Toc535413979)

[Ilustración 59 Sección de ventas 269](#_Toc535413980)

[Ilustración 60 Sección de ventas diarias 271](#_Toc535413981)

[Ilustración 61 Sección de ventas mensuales 273](#_Toc535413982)

[Ilustración 62 Sistema de inventario en línea "Stock-Web" 280](#_Toc535413983)

[Ilustración 63 Sistema de inventario 280](#_Toc535413984)

[Ilustración 64 Programación en PHP 281](#_Toc535413985)

[Ilustración 65 Uso de NetBeans 282](#_Toc535413986)

[Ilustración 66 Programación en PHP por medio del IDE 283](#_Toc535413987)

[Ilustración 67 Programación en JavaScript por medio del IDE 284](#_Toc535413988)

[Ilustración 68 Uso de HTML por medio del IDE 285](#_Toc535413989)

[Ilustración 69 Uso de CSS por medio del IDE 286](#_Toc535413990)

[Ilustración 70 Uso del SGBD del hosting 287](#_Toc535413991)

[Ilustración 71 Programación de las interfaces por medio del IDE 288](#_Toc535413992)

[Ilustración 72 Funcionamiento de la aplicación web 289](#_Toc535413993)

[Ilustración 73 Sistema de inventario en línea Stock-Web 1](#_Toc535413994)

[Ilustración 74 Página de inicio de sesión 4](#_Toc535413995)

[Ilustración 75 Página principal del usuario administrador 5](#_Toc535413996)

[Ilustración 76 Página principal del usuario especial 6](#_Toc535413997)

[Ilustración 77 Página principal de usuario estándar 6](#_Toc535413998)

[Ilustración 78 Página de panel de control 7](#_Toc535413999)

[Ilustración 79 Página de grupos 8](#_Toc535414000)

[Ilustración 80 Página para editar grupo 9](#_Toc535414001)

[Ilustración 81 Página para agregar grupo 10](#_Toc535414002)

[Ilustración 82 Página de usuarios 11](#_Toc535414003)

[Ilustración 83 Página para editar cuenta de usuario 12](#_Toc535414004)

[Ilustración 84 Página para agregar usuario 12](#_Toc535414005)

[Ilustración 85 Página de categorías 13](#_Toc535414006)

[Ilustración 86 Página para editar categoría 14](#_Toc535414007)

[Ilustración 87 Página de productos 15](#_Toc535414008)

[Ilustración 88 Página para editar producto 16](#_Toc535414009)

[Ilustración 89 Página para agregar producto 17](#_Toc535414010)

[Ilustración 90 Página de productos eliminados 18](#_Toc535414011)

[Ilustración 91 Página de imágenes de productos 19](#_Toc535414012)

[Ilustración 92 Página para agregar múltiples registros 20](#_Toc535414013)

[Ilustración 93 Página de ventas 21](#_Toc535414014)

[Ilustración 94 Página de editar venta 21](#_Toc535414015)

[Ilustración 95 Página para agregar venta 22](#_Toc535414016)

[Ilustración 96 Página de reporte de ventas 23](#_Toc535414017)

[Ilustración 97 Página de reporte de ventas generado 23](#_Toc535414018)

[Ilustración 98 Página de ventas mensuales 24](#_Toc535414019)

[Ilustración 99 Página de ventas diarias 25](#_Toc535414020)

[Ilustración 100 Página de perfil 25](#_Toc535414021)

[Ilustración 101 Página para editar cuenta 27](#_Toc535414022)

[Ilustración 102 Página para cambiar contraseña 28](#_Toc535414023)

NOMENCLATURAS

**C#** (pronunciado si Sharp en inglés): es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.[[1]](#footnote-1)

**Disco duro**: es el dispositivo del sistema de memoria del (Iglesias, 2016) PC que usamos para almacenar todos los programas y archivos ya que es el único capaz de guardar datos incluso aunque no esté alimentado por corriente eléctrica.[[2]](#footnote-2)

**HTML**, que significa Lenguaje de Marcado para Hipertextos (Hipertexto Markup Language): es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad.[[3]](#footnote-3)

**Interfaz**: es un término que procede del vocablo inglés interface. En informática, esta noción sirve para señalar a la conexión que se da de manera física y a nivel de utilidad entre dispositivos o sistemas.[[4]](#footnote-4)

**Java**: Sun Microsystems desarrolló, en 1991, el lenguaje de programación orientado a objetos que se conoce como Java.[[5]](#footnote-5)

**Loopback**: se usa generalmente para probar (conste que no puse testear) una línea de comunicación un enlace cableado, un modem, el puerto de comunicaciones de una PC y ver que lo que se está transmitiendo es lo mismo que se recibe ya que justamente el loopback regresa como si fuera un eco la información enviada desde determinado punto del enlace.[[6]](#footnote-6)

**PHP** (acrónimo recursivo de PHP: Hypertext Preprocessor): es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.[[7]](#footnote-7)

**RAM**: es el acrónimo del concepto inglés de Random Access Memory (Memoria de Acceso Aleatorio). Se trata de la memoria que, en un equipo informático, es utilizada por un procesador para recibir instrucciones y guardar los resultados.[[8]](#footnote-8)

**SGBD**: Sistemas de gestión de bases de datos (en inglés database management system, abreviado DBMS) o SGBD son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Permiten describir los elementos de datos con su estructura, sus interrelaciones y sus validaciones.[[9]](#footnote-9)

# I. INTRODUCCIÓN AL PROYECTO

## I.1 DEFINICIÓN DEL PROBLEMA

En una empresa de productos de limpieza se desconoce la cantidad existente de cada uno de los productos a la venta en su inventario, así como las ventas realizadas. En ocasiones se necesita contar uno por uno los productos que hay en el inventario, porque no se conoce la cantidad disponible que se puede vender a un cliente. No se lleva un registro de las ventas realizadas en electrónico, por lo que se registran en papel y a veces se pierden esos papeles.

* No se conoce la cantidad existente de cada producto en el inventario.
* No se tiene un registro de las ventas.
* Se tiene que revisar la cantidad en inventario de un producto, antes de venderlo.
* Las ventas no se registran en electrónico y se consultan por medio de registros en papel.

## I.2 OBJETIVOS Y ALCANCES DEL PROYECTO

Desarrollar un sistema que administre un inventario que será usado por usuarios. Este sistema permitirá administrar y agregar productos, generar ventas y poder generar reportes de ventas. Se logrará por medio de un servidor web que contará con la aplicación web programada en PHP y compuesta de documentos HTML.

* Desarrollar una aplicación web para la administración de un inventario en línea por medio de HTML5, CSS3, PHP y JavaScript.
* Programar la sección de productos, ventas, reportes de ventas, etc., por medio de PHP.
* Utilizar el IDE NetBeans para diseñar y programar esta aplicación en una computara con el sistema operativo Windows por ser a juicio personal el IDE más vistoso.
* Hacer uso del lenguaje de programación PHP para programar desde el lado del servidor la aplicación web por medio del IDE.
* Hacer uso del lenguaje de programación JavaScript para programar desde el lado del cliente la aplicación web por medio del IDE.
* Hacer uso del lenguaje de etiquetas HTML5 para diseñar las páginas web de la aplicación web por medio del IDE.
* Hacer uso del lenguaje de estilos en cascada CSS para diseñar las páginas web de la aplicación web por medio del IDE.
* Hacer uso de un SGBD MySQL para almacenar los datos de la aplicación web por medio del hosting.
* Diseñar y programar adecuadamente todas las interfaces de la aplicación web que verá el usuario por medio del IDE.
* Hacer pruebas a la aplicación web para checar su funcionamiento de cara al usuario final por medio del documento de caso de pruebas.

## I.3 JUSTIFICACIÓN

Esta aplicación beneficiará a una empresa de limpieza que desea poder llevar el control de su inventario, así como registrar sus ventas.

Para poder usar esta aplicación bastará con tener el vínculo; ya que será web el proyecto, por lo que puede ser utilizado en cualquier dispositivo que use cualquiera de los navegadores necesarios para utilizar esta aplicación.

Esta aplicación estará diseñada en PHP con HTML5 por lo cual utilizará las últimas tecnologías y especificaciones que se necesitan para poder tener la aplicación web al día y con un posicionamiento web adecuado. La base de datos a utilizar será MySQL por la sencillez y el bajo requerimiento de nivel de seguridad de los datos.

Ventajas que ofrecerá este sistema:

* Sistema de login.
* Administrar grupos.
* Administrar usuarios.
* Administrar productos.
* Administrar ventas.
* Generar reportes de ventas.
* Personalización de apariencia.
* Personalizar imagen de la empresa con logan
* Pasar información de otro sistema a este por medio de formularios que no necesitarán recargar la página para interactuar con la base de datos.
* 3 tipos de usuario:
* Administrador: control total de los usuarios y el inventario.
* Usuario especial: control del inventario.
* Usuario: todo lo relacionado con ventas.

El nivel de complejidad de uso de esta aplicación web será bajo, bastará con que el usuario que la tenga que utilizar, sepa utilizar una computadora o un dispositivo móvil.

## I.4 DESCRIPCIÓN DEL PRODUCTO

Panel de control amigable: El sistema contará con un panel de control desde donde se contabilizan a manera de reporte los usuarios, categorías, productos y ventas registradas. Además, mostrará un pequeño reporte de los productos más vendidos, ventas recientes y un listado de los productos recientemente agregados a la base de datos del sistema. Este panel podrá ser personalizable en cuanto a fondo, logo de la empresa, logan de la empresa y el tema del panel en cualquier momento.

Administración de grupos de usuarios: El sistema manejará 3 tipos de niveles de permisos, permitiendo así; desde el módulo de administración de grupos de usuarios la gestión de dichos grupos, permitiendo las operaciones de visualización, creación, edición y eliminación de los grupos.

Administración de usuarios: Desde el módulo de administración de usuario se podrá gestionar los usuarios, permitiendo también poder visualizarlos, crear nuevos usuarios, editar los datos de los usuarios y poder eliminarlos.

Gestión de categorías: Permitirá el ingreso, edición y eliminación de categorías al sistema.

Administración de productos: Desde el módulo de administración de productos podremos gestionar el inventario. Si un producto es eliminado se guarda automáticamente en una tabla de BD aparte, para poder recuperarlo fácilmente.

Sección multimedia: En este módulo se permitirá la carga de imágenes al servidor, para que posteriormente sean asignadas a un producto. Este módulo también podrá ser usado para poder pasar información de un sistema de inventario diferente a este, hacia el nuestro. Se realizará por medio de una plantilla que genera nuestro propio sistema, el cual el usuario deberá rellenar, para después poder subirla a este propio módulo y no tener que pasar uno por uno cada uno de los datos.

Gestión de ventas: Permitirá listar las ventas realizadas, con la posibilidad de ir agregando nuevas ventas. Cuando se realicen ventas el sistema automáticamente da salida del inventario las cantidades vendidas. Cuando la cantidad de un producto es inferior al umbral que indica el usuario, se generará automáticamente una notificación de escasa cantidad de producto.

Generación de reportes: El sistema permitirá generar 3 tipos de reportes: Reporte por rango de fechas, reportes mensuales y reporte de ventas diarias. Todos estos reportes pueden ser generados automáticamente, pero indicando cuales.

## I.5 LISTA DE REQUERIMIENTOS FUNCIONALES

|  |  |
| --- | --- |
| Rqf1. | En la interfaz de inicio de sesión, todos los usuarios ingresarán su nombre de usuario y contraseña en el formulario correspondiente. |
| Rqf2. | En la interfaz de inicio de sesión, a todo usuario que ingrese datos incorrectos o inválidos, se le mostrará uno de los siguientes mensajes de error: el nombre de usuario es incorrecto o la contraseña es incorrecta o el grupo al que pertenece el usuario está cesado o el usuario en cuestión está cesado. El mensaje se mostrará en la parte superior del formulario de inicio de sesión. |
| Rqf3. | Los usuarios tendrán en el menú lateral izquierdo funcionalidades del sistema, de acuerdo con su nivel de permisos. |
| Rqf4. | El usuario tipo administrador tendrá panel de control, control de accesos (grupos de usuario y usuarios), administración de categorías, administración de productos (productos y productos eliminados), administración de media (imágenes de productos y sección para agregar múltiples registros), administración de ventas y administración de reporte de ventas. |
| Rqf5. | El usuario tipo especial tendrá panel de control, administración de categorías, administración de productos (productos y productos eliminados), administración de media (imágenes de productos), administración de ventas y administración de reporte de ventas. |
| Rqf6. | El usuario tipo estándar tendrá panel de control, administración de ventas y administración de reporte de ventas. |
| Rqf7. | En la interfaz de panel de control, todos los usuarios verán el número de usuarios registrados, número de categorías, número de productos y número de ventas. |
| Rqf8. | En la interfaz de panel de control, todos los usuarios verán los productos más vendidos, últimas ventas, productos recientemente añadidos, productos con bajo stock (en caso de que los haya) y reportes automáticos. |
| Rqf9. | En la interfaz de panel de control, en la sección de productos recientemente añadidos, los usuarios tipo administrador y especial pueden interactuar editar el producto de su selección. |
| Rqf10. | En la interfaz de grupos, el usuario tipo administrador verá los datos de los grupos existentes: nombre, nivel y estado; así como posibilidad de editarlos o eliminarlos. |
| Rqf11. | En la interfaz de agregar grupos, el usuario tipo administrador deberá ingresar el nombre del grupo, nivel del grupo (1-administrador, 2-especial, 3-estandar), estado del grupo (activo o inactivo). |
| Rqf12. | En la interfaz de agregar grupos, en caso de que el usuario tipo administrador introduzca datos incorrectos, se le mostrará un mensaje de error. El mensaje indicará si el nombre de grupo ya está siendo utilizado o el nivel de grupo también está siendo utilizado. |
| Rqf13. | En la interfaz de editar grupos, el usuario tipo administrador podrá editar el nombre del grupo, nivel del grupo, estado del grupo. |
| Rqf14. | En la interfaz de editar grupos, en caso de que el usuario tipo administrador introduzca datos incorrectos, se le mostrará un mensaje de error. El mensaje indicará si el nombre de grupo ya está siendo utilizado o el nivel de grupo también está siendo utilizado. |
| Rqf15. | En la interfaz de grupos, el usuario tipo administrador no podrá eliminar el grupo con nivel 1. |
| Rqf16. | En la interfaz de editar grupos, el usuario tipo administrador no podrá editar más que el nombre del grupo con nivel 1; para el resto de los grupos no aplica esto. |
| Rqf17. | En la interfaz de usuarios, el usuario tipo administrador verá los datos de los usuarios existentes: nombre, nombre de usuario, rol de usuario, estado, último inicio de sesión, así como posibilidad de editarlos o eliminarlos. |
| Rqf18. | En la interfaz de agregar usuarios, el usuario tipo administrador deberá ingresar el nombre completo de la persona que usará la cuenta, nombre de usuario, contraseña y rol de usuario (grupo al que pertenece). |
| Rqf19. | En la interfaz de agregar usuarios, en caso de que el usuario tipo administrador introduzca datos incorrectos, se le mostrará un mensaje de error. El mensaje indicará si el nombre de usuario ya está siendo utilizado. |
| Rqf20. | En la interfaz de editar usuarios, el usuario tipo administrador podrá editar el nombre completo de la persona que usará la cuenta, nombre de usuario, contraseña y rol de usuario. |
| Rqf21. | En la interfaz de editar usuarios, en caso de que el usuario tipo administrador introduzca datos incorrectos, se le mostrará un mensaje de error. El mensaje indicará si el nombre de usuario ya está siendo utilizado. |
| Rqf22. | En la interfaz de usuarios, el usuario tipo administrador no podrá eliminar al administrador “Administrador”. |
| Rqf23. | En la interfaz de editar usuarios, el usuario tipo administrador no podrá editar más que el nombre del usuario administrador “Administrador”; para el resto de los usuarios no aplica esto. |
| Rqf24. | En la interfaz de categoría, el usuario tipo administrador y especial verán el nombre de la categoría. |
| Rqf25. | En la interfaz de categoría, el usuario tipo administrador y especial deberán ingresar el nombre de la categoría. |
| Rqf26. | En la interfaz de categoría, en caso de que el usuario tipo administrador y especial introduzcan datos incorrectos, se les mostrará un mensaje de error. El mensaje indicará si el nombre de categoría ya está siendo utilizado. |
| Rqf27. | En la interfaz de editar categorías, el usuario tipo administrador y especial podrán editar el nombre de la categoría. |
| Rqf28. | En la interfaz de editar categorías, en caso de que el usuario tipo administrador y especial introduzcan datos incorrectos, se les mostrará un mensaje de error. El mensaje indicará si el nombre de categoría ya está siendo utilizado. |
| Rqf29. | En la interfaz de productos, el usuario tipo administrador y especial verán datos de los productos existentes: imagen del producto, descripción del producto, categoría del producto, cantidad existente de producto, precio de compra del producto, precio de venta del producto, fecha de actualización de producto, así como posibilidad de editarlos o eliminarlos. |
| Rqf30. | En la interfaz de agregar productos, el usuario tipo administrador y especial deberán ingresar la imagen del producto, descripción del producto, categoría del producto, cantidad existente de producto, precio de compra del producto y precio de venta del producto. |
| Rqf31. | En la interfaz de agregar productos, en caso de que el usuario tipo administrador y especial introduzcan datos incorrectos, se les mostrará un mensaje de error. El mensaje indicará si la descripción de producto ya está siendo utilizada. |
| Rqf32. | En la interfaz de editar productos, el usuario tipo administrador y especial podrán modificar la imagen del producto, descripción del producto, categoría del producto, cantidad existente de producto, precio de compra del producto y precio de venta del producto. |
| Rqf33. | En la interfaz de editar productos, en caso de que el usuario tipo administrador y especial introduzcan datos incorrectos, se les mostrará un mensaje de error. El mensaje indicará si la descripción de producto ya está siendo utilizada. |
| Rqf34. | En la interfaz de productos eliminados, el usuario tipo administrador y especial verán datos de los productos existentes: imagen del producto, descripción del producto, categoría del producto, cantidad existente de producto, precio de compra del producto, precio de venta del producto, fecha de actualización de producto, así como posibilidad de restaurarlos o eliminarlos definitivamente. |
| Rqf35. | En la interfaz de media, el usuario tipo administrador, especial y estándar verán los datos de las imágenes de productos: la imagen, descripción, extensión de la imagen, así como posibilidad de eliminarla. |
| Rqf36. | En la interfaz de media, el usuario tipo administrador, especial y estándar podrán subir imágenes para productos. |
| Rqf37. | En la interfaz de agregar múltiples registros el usuario tipo administrador podrá ver las categorías y productos registrados. |
| Rqf38. | En la interfaz de agregar múltiples registros el usuario tipo administrador podrá registrar varias categorías y productos sin recargar la página. |
| Rqf39. | En la interfaz de agregar múltiples registros, en caso de que el usuario tipo administrador introduzca datos incorrectos, se le mostrará un mensaje de error. El mensaje indicará si la categoría o producto ya ha sido registrado. |
| Rqf40. | En la interfaz de ventas, el usuario tipo administrador, especial y estándar verán datos de las ventas registradas: nombre del producto que se vendió, cantidad vendida, precio de venta total, fecha en que se efectuó la compra, el usuario que la registró, así como posibilidad de modificarla o eliminarla (solo usuario tipo administrador). |
| Rqf41. | En la interfaz de agregar venta, el usuario tipo administrador, especial y estándar deberán buscar y seleccionar el producto que desean vender, para posteriormente indicar la cantidad a vender y la fecha de la venta. |
| Rqf42. | En la interfaz de editar venta, el usuario tipo administrador podrá modificar la fecha de venta. |
| Rqf43. | En la interfaz de reporte de ventas, el usuario tipo administrador, especial y estándar podrán generar reportes de ventas, indicando la fecha de inicio y fin. |
| Rqf44. | En la interfaz de ventas mensuales, el usuario tipo administrador, especial y estándar podrán ver los datos de las ventas registradas del mes actual: descripción del producto, cantidad vendida del producto, precio de venta total, fecha en que se registró la venta y el usuario que realizó la venta. |
| Rqf45. | En la interfaz de ventas diarias, el usuario tipo administrador, especial y estándar podrán ver los datos de las ventas registradas del día actual: descripción del producto, cantidad vendida del producto, precio de venta total, fecha en que se registró la venta y el usuario que realizó la venta. |
| Rqf46. | En la interfaz de perfil, todos los usuarios podrán acceder a su configuración de cuenta. |
| Rqf47. | En la interfaz de editar perfil, todos los usuarios podrán cambiar su imagen de perfil, cambiar el fondo de la aplicación, el tema de la aplicación, seleccionar reportes de ventas automáticos, editar su nombre real, nombre de usuario e ir a la interfaz cambiar contraseña. |
| Rqf48. | En la interfaz de editar perfil, el usuario de tipo administrador podrá cambiar el lema de la empresa dueña de este sistema y cambiar o eliminar el logo de la empresa. |
| Rqf49. | En la interfaz de cambiar contraseña, todos los usuarios podrán cambiar su contraseña. Primero deberán ingresar su nueva contraseña y luego su antigua contraseña. Al finalizar el cambio de su contraseña, el usuario será direccionado a la interfaz de inicio de sesión y deberá iniciar sesión nuevamente. |

Requisitos de software

|  |  |
| --- | --- |
| Rqf50. | La computadora o dispositivo debe contar con el navegador Opera o Google Chrome. |

## I.6 LISTA DE REQUERIMIENTOS NO FUNCIONALES

|  |  |
| --- | --- |
| Rqnf1. | El sistema debería funcionar adecuadamente en los navegadores web Opera y Google Chrome más recientes. |
| Rqnf2. | El sistema debería poder subir imágenes correctamente al directorio correspondiente por medio de la clase PHP correspondiente. |

Requisitos de rendimiento

|  |  |
| --- | --- |
| Rqnf3. | Soportará hasta 200 consultas simultáneas al servidor. |
| Rqnf4. | Soportará hasta 30 usuarios conectados. |
| Rqnf5. | El tiempo de carga de la información desde el servidor será de alrededor de 0.45s. |
| Rqnf6. | La aplicación web estará disponible las 24 horas del día los 365 días del año. |
| Rqnf7. | La aplicación web requiere cifrado y descifrado de contraseña. |

Restricciones de diseño

|  |  |
| --- | --- |
| Rqnf8. | Las interfaces de usuario estarán diseñadas para ser responsivas. |

## I.7 COSTO DEL PROYECTO Y ESTIMACIÓN DE COSTOS DE MATERIALES

Se utiliza el COCOMO Modelo Constructivo de Costes (Constructive Cost Model) para obtener el costo de software del proyecto.

Estimación COCOMO

Líneas de código del proyecto de sistema de inventario en línea “Stock-Web”. Las líneas de código son el estimado que se cree, necesitará el proyecto para estar completo y funcional.

|  |  |
| --- | --- |
| Lenguaje de programación, etiquetas o estilos | Líneas de código |
| CSS | 1610 |
| JavaScript | 667 |
| PHP | 2720 |
| HTML | 2280 |
| SQL | 584 |
| Total | 7861 |

Se escoge el tipo de proyecto de software, en este caso el tipo orgánico será el más apropiado ya que el número de líneas de código no supera los 50 KLDC, y además el proyecto no es muy complejo, por consiguiente, los coeficientes que usaremos serán las siguientes:



Ilustración 1 Coeficientes de proyecto de software

Y por otro lado se debe hallar la variable ME (Multiplicador de Esfuerzo), la cual se obtiene mediante la multiplicación de los valores evaluados en los diferentes 15 conductores de coste que se observan en la siguiente tabla:

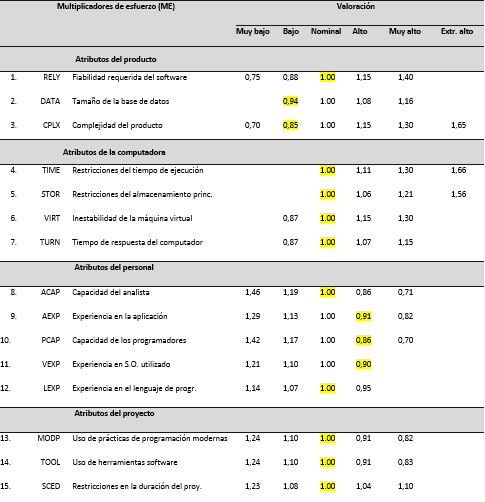


Ilustración 2 Tabla de conductores de coste

ME = 1 \* 0.94 \* 0.85 \* 1 \* 1 \* 1 \* 1 \* 1 \* 0.91 \* 0.86 \* 0.90 \* 1 \* 1 \* 1 \* 1

ME = 0.56276766

*Cálculo del esfuerzo del desarrollo*

Aquí se calcula las personas por mes necesarias para desarrollar el proyecto. Se necesita conocer las KLDC (Kilo Líneas De Código), la variable ME (Multiplicador de Esfuerzo) y los coeficientes de proyecto de software (a, e).

Esfuerzo = E

E = a KLDC e \* ME

E = 3.2 \* (7.861) ^ 1.05 \* 0.56276766

E = 15.3167 personas / mes

*Cálculo tiempo de desarrollo*

Aquí se calculan los meses necesarios para desarrollar el proyecto. Se necesita conocer el esfuerzo y los coeficientes de proyecto de software (c, d).

Meses de desarrollo = T

T = c Esfuerzo d

T = 2.5 \* (15.3167) ^ 0.38

T = 7.0518 meses

*Productividad:*

Aquí se calcula la productividad o las personas necesarias por mes para desarrollar el proyecto. Se necesita conocer las líneas de código (Líneas De Código) y el esfuerzo.

Productividad = PR

PR = LDC / Esfuerzo

PR = 7861 / 15.3167

PR = 513.2306 LDC / personas / mes

*Personal promedio*

Aquí se calcula el personal promedio para desarrollar el proyecto. Se necesita conocer el esfuerzo y el tiempo de desarrollo.

Personal promedio = P

P = E / T

P = 15.3167 / 7.0518

P = 2.1720 personas

*Conclusiones*

El proyecto se realizará en el doble de tiempo de desarrollo (14 meses), por lo que el personal promedio será de 1.086. EL costo por mes del personal será de 4000 pesos. Finalmente, la estimación de costes de proyecto será de 56000 pesos.

*Costo del proyecto*

|  |  |  |
| --- | --- | --- |
| Descripción | Cantidad | Costo |
| Costo de software | 1 | 56000 |
| Energía eléctrica | 14 | 600 |
| Servicio de agua | 14 | 400 |
| Insumos | 14 | 500 |
| Gasolina | 14 | 1000 |
| Computadora portátil | 1 | 7500 |
| Total | 58 | 98500 |

# II. ANTECEDENTES DEL PROYECTO

Antecedentes de una empresa con rol de limpieza

Es conocido que una empresa que cuenta con inventario necesita administrarlo correctamente. En la empresa es necesario saber que productos tienen baja cantidad de inventario, así como poder registrar las ventas que se realizan e ir actualizando el inventario. Se puede utilizar desde un programa dedicado a administrar el inventario o manejarse por medio de lo que se ve en los anaqueles de la empresa. Finalmente, la decisión dependerá de cada empresa.

Fundamentos del sistema de inventario en línea “Stock-Web”

Este proyecto administrará el sistema de inventario de una empresa que necesita tener el control de la cantidad de productos que tiene, así como las ventas que se realizan.

Proyectos parecidos con el sistema de inventario en línea “Stock-Web”

*ABC Inventory*

ABC Inventory es un sistema gratuito para un solo usuario, que te permite administrar almacenes múltiples, productos con números de serie, garantías y una gran cantidad de otros datos relacionados con el inventario. Sin embargo, esta versión gratuita está diseñada para ejecutarse en una computadora, ya que varias estaciones de trabajo no pueden sincronizar los mismos datos.

*Delivrd*

Delivrd es una solución para la administración de inventarios y cumplimiento de pedidos basada en la nube que lanzó su primera versión oficial a finales del 2016. La empresa ofrece 3 planes: gratis y $49.99 y $99 por mes. El plan gratuito cubre una sucursal y crear un catálogo de 25 productos. Una excelente opción para las empresas que tienen un enfoque limitado, como los minoristas en línea y los minoristas de productos especializados.

Si está utilizando WooCommerce o Shopify, Delivrd puede integrarse directamente con ellos, haciendo que la vida sea mucho más fácil en con back-end. Si usas otra plataforma de comercio electrónico, tendrá que trabajar con una transferencia de datos manual para que todo funcione.

Delivrd también admite códigos de barras, historial de transacciones realizadas, alerta de stock con existencia mínima y algunos reportes financieros. En el plan de pago, obtienes acceso a más ubicaciones y productos, así como también obtiene algunas herramientas de administración de ventas y punto de venta.[[10]](#footnote-10)

# III. PLANTEAMIENTO Y MODELOS DEL PROYECTO

## III.1 OPCIONES DE SOLUCIÓN

La primera solución sería desarrollar una aplicación de sistema de inventario de escritorio para el sistema operativo Windows. Contaría con un panel de control, una sección de productos, ventas, reporte de ventas y los accesos por medio de grupos de usuarios. La apariencia de la aplicación sería acorde a las aplicaciones de Windows y utilizaría varias ventanas de Windows para mostrar una interfaz de usuario.

La segunda solución sería desarrollar una aplicación de sistema de inventario móvil para el sistema operativo Android. Contaría con un panel de control, una sección de productos, ventas, reporte de ventas y los accesos por medio de grupos de usuarios. La apariencia de la aplicación sería personalizable y se mostrarían notificaciones en el dispositivo móvil del usuario.

## III.2 EVALUACIÓN Y SELECCIÓN

En esta sección se explican las pruebas por realizar en el sistema para confirmar si el funcionamiento es correcto o no y mostrar el procedimiento de verificación de cada prueba.

A continuación, se muestra la tabla con los casos de prueba.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Caso de prueba. | Caso de uso. | Entrada. | Condiciones de ejecución. | Resultado esperado. | Objetivo del caso de prueba. |
| CP1. | Formulario para iniciar sesión con nombre de usuario y contraseña. | Ingresar el nombre de usuario junto con la contraseña y dar clic al botón para tratar de iniciar sesión. | El usuario debe estar registrado en el sistema. | El usuario inicia sesión exitosamente. | Comprobar que el usuario que desee iniciar sesión en el sistema debe de estar registrado. |
| CP2. | Cuadro de error de formulario de inicio de sesión. | Llenar el formulario de inicio de sesión y tratar de iniciar sesión. | El nombre de usuario, contraseña o permisos no son los correctos. | Mostrar cuadro de error sobre el campo de usuario con la descripción del error. | Comprobar que se informa al usuario sobre problemas de inicio de sesión. |
| CP3. | Formulario para editar nombre de grupo de tipo administrador. | Editar el nombre de grupo y dar clic al botón actualizar. | El grupo por editar es de tipo administrador. | Editar únicamente el nombre de grupo. | Comprobar la imposibilidad de modificar datos de este grupo más allá del nombre. |
| CP4. | Formulario para editar el nombre personal del usuario administrador por defecto y con sesión activa. | Editar el nombre personal y dar clic al botón actualizar. | El usuario por editar es el administrador por defecto o el usuario con sesión activa. | Editar únicamente el nombre real de la cuenta. | Comprobar la imposibilidad de modificar datos de estos usuarios más allá del nombre personal. |
| CP5. | Acciones de usuario con sesión activa y administrador por defecto. | Ir a la sección de usuarios. | Los usuarios con sesión activa y el administrador por defecto no pueden eliminar la o su cuenta. | Poder editar la cuenta únicamente. | Comprobar la imposibilidad de eliminar el usuario con sesión actual o administrador por defecto. |
| CP6. | Formulario para restaurar producto eliminado. | Dar clic al botón de restaurar. | No debe existir otro producto no eliminado con el mismo nombre. | Restaurar el producto. | Comprobar que se puede restaurar un producto. |
| CP7. | Formulario para agregar varias categorías. | Llenar varias categorías y dar clic en el botón de insertar categorías. | Las categorías se registran en la base de datos en una sola operación sin necesidad de recargar la página. | Agregar varias categorías. | Comprobar que se puede agregar varias categorías sin recargar la página. |
| CP8. | Formulario para agregar varios productos. | Llenar varios productos y dar clic en el botón de insertar productos. | Los productos se registran en la base de datos en una sola operación sin necesidad de recargar la página. | Agregar varios productos. | Comprobar que se puede agregar varios productos sin recargar la página. |
| CP9. | Formulario para editar el nombre personal de usuario, así como el nombre de usuario. | Editar el nombre personal, así como el nombre de usuario y dar clic en el botón actualizar para guardar cambios. | El usuario debe iniciar sesión y acceder a su perfil. | Editar el nombre personal, así como el nombre de usuario. | Comprobar que se puede editar el nombre personal, así como el nombre de usuario. |
| CP10. | Formulario para cambiar la contraseña. | Ingresar la nueva contraseña, así como la antigua contraseña y dar clic al botón cambiar contraseña. | La antigua contraseña debe coincidir con la contraseña actual. | Cambiar la contraseña de la cuenta exitosamente y direccionar a la pantalla de inicio de sesión. | Comprobar que se puede cambiar la contraseña exitosamente. |

## III.3 MODELOS

Diagrama de entidad relación de la base de datos

El modelo se muestra en la ilustración 3. Se muestran las entidades (tablas) en forma de rectángulos, los atributos de estas entidades (campos) en forma de óvalos, las relaciones entre entidades se muestran en forma de un rombo que une dos entidades; las relaciones tienen en cada extremo el tipo de relación según se lea de entidad a entidad, es decir una tabla tiene uno o más registros de la otra tabla o un solo registro de la otra tabla. Las siglas 1..1 y 1..M de las relaciones significan 1 y solo 1 y 1 o muchos respectivamente.

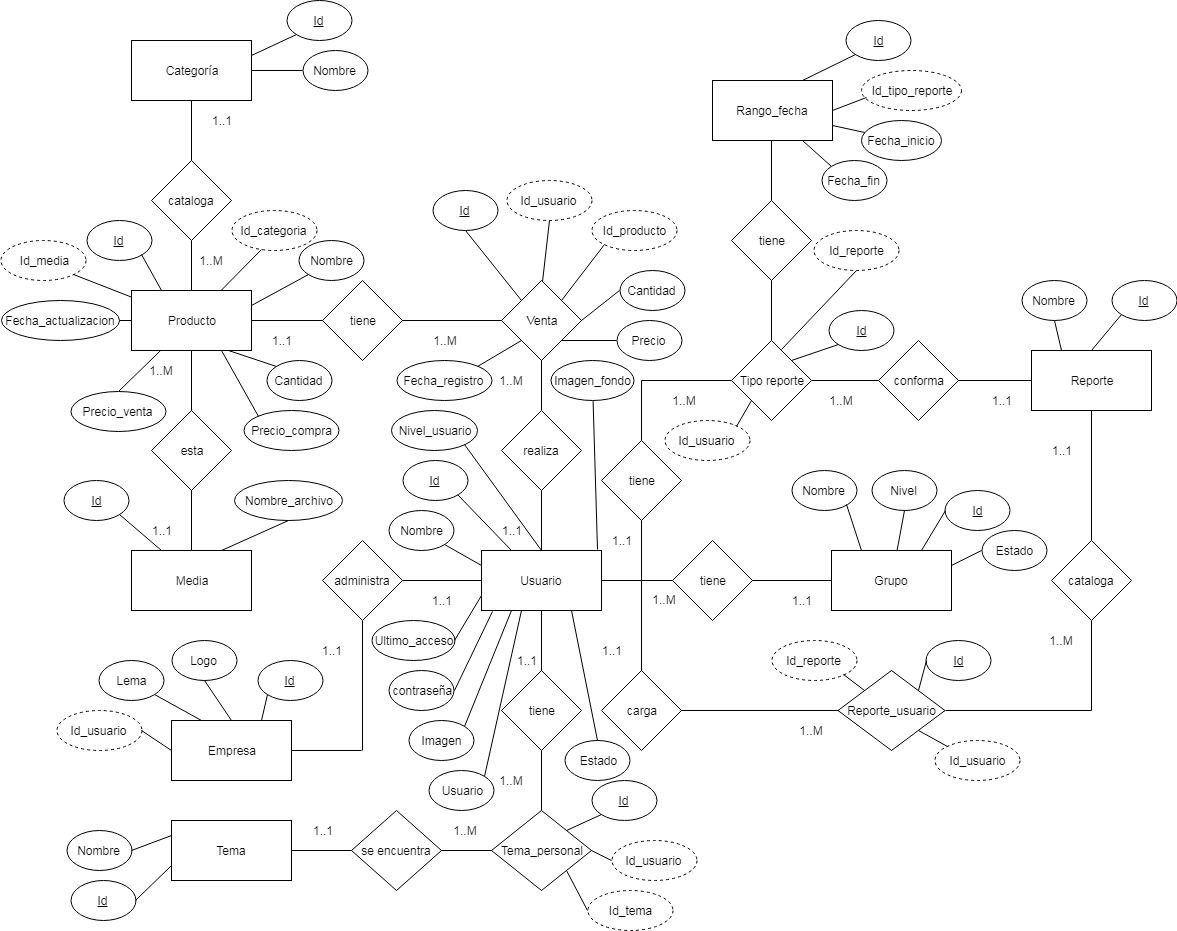


Ilustración 3 Diagrama de entidad relación de la base de datos

Normalización de base de datos

Tabla no normalizada de producto

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Id | Nombre | Categoría | Media | Cantidad | Precio | Fecha\_act. |

Primera forma normal: los datos deben ser atómicos. Se divide el campo precio en precio de venta y precio de compra.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Nombre | Categoría | Media | Cantidad | Precio\_vent | Precio\_compra | Fecha\_act. |

Segunda forma normal: se eliminan datos redundantes. Se crean tablas separadas para los campos: categoría y media, estas tablas se deben relacionar con la tabla producto por medio de una clave externa.

*Producto*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Nombre | Id\_categoría | Id\_media | Cantidad | Precio\_vent | Precio\_compra | Fecha\_act. |

*Categoría*

|  |  |
| --- | --- |
| Id | Nombre |

*Media*

|  |  |
| --- | --- |
| Id | Nombre\_archivo |

Tabla no normalizada de venta

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Usuario | Producto | Precio | Fecha\_registro |

Primera forma normal: los datos deben ser atómicos. Se divide el campo producto en producto y cantidad.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Usuario | Producto | Cantidad | Precio | Fecha\_registro |

Segunda forma normal: se eliminan datos redundantes. Se crean tablas separadas para los campos usuario y producto, estas tablas se deben relacionar con la tabla usuario por medio de una clave externa.

*Venta*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Id\_usuario | Id\_producto | Cantidad | Precio | Fecha\_registro |

Tabla no normalizada de usuario

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Nombre | Usuario | Ima. | Con | Acceso | Nivel | Imagen\_f | Tema | Emp. | Reporte |

Primera forma normal: los datos deben ser atómicos. Se divide el campo acceso en acceso y estado.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Nombre | Usuario | Ima. | Con | Acc. | Esdo. | Nivel | Imagen\_f | Tema | Emp. | Reporte |

Segunda forma normal: se eliminan datos redundantes. Se crean tablas separadas para los campos tema, tema\_personal, reporte, tipo\_reporte y reporte\_usuario, estas tablas se deben relacionar con la tabla usuario por medio de una clave externa.

*Usuario*

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | Nombre | Usuario | Imagen | Contraseña | Acceso | Estado | Nivel | Imagen\_fondo |

*Tema\_personal*

|  |  |  |
| --- | --- | --- |
| Id | Id\_usuario | Id\_tema |

*Tema*

|  |  |
| --- | --- |
| Id | Nombre |

*Tipo\_reporte*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Id\_reporte | Id\_usuario | Fecha\_inicio | Fecha\_fin |

*Reporte\_usuario*

|  |  |  |
| --- | --- | --- |
| Id | Id\_reporte | Id\_usuario |

*Reporte*

|  |  |
| --- | --- |
| Id | Nombre |

Tercera forma normal: el campo de empresa no pertenece a la clave principal por lo que se colocan en una tabla aparte: empresa y se relaciona con la tabla usuario por medio de una clave externa.

*Empresa*

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Id\_usuario | Lema | Logo |

Tabla no normalizada de grupo

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Nombre | Estado | Usuario |

Primera forma normal: los datos deben ser atómicos. Se divide el campo estado en estado y nivel.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Nombre | Nivel | Estado | Usuario |

Segunda forma normal: se eliminan datos redundantes. Se crea una tabla separada para el campo usuario, esta tabla se debe relacionar con la tabla grupo por medio de una clave externa.

*Grupo*

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Nombre | Nivel | Estado |

Tabla no normalizada tipo\_reporte

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Reporte | Usuario | Fecha |

Primera forma normal: los datos deben ser atómicos. Se divide el campo fecha en fecha de inicio y fecha de fin.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Reporte | Usuario | Fecha\_inicio | Fecha\_fin |

Segunda forma normal: se eliminan datos redundantes. Se crea una tabla separada para el campo usuario, esta tabla se debe relacionar con la tabla grupo por medio de una clave externa.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Id | Id\_reporte | Id\_usuario | Fecha\_inicio | Fecha\_fin |

Tercera forma normal: los campos fecha\_inicio y fecha\_fin no pertenecen a la clave principal por lo que se colocan en una tabla aparte: rango\_fecha y se relaciona con la tabla tipo\_reporte por medio de una clave externa.

*Rango\_fecha*

|  |  |  |  |
| --- | --- | --- | --- |
| Id | Id\_tipo\_reporte | Fecha\_inicio | Fecha\_fin |

Diagrama de tablas de base de datos

El modelo se muestra en la ilustración 4. Se muestran las tablas en forma de rectángulos con su nombre en negritas en la parte superior derecha y con el nombre de la base de datos a la izquierda. Los campos de cada tabla se encuentran debajo con su nombre; a la izquierda está un símbolo que indica si es clave primaria (llave), campo estándar (hashtag), campo de fecha (calendario) o campo de texto (circulo gris) y a la derecha se muestra el nombre del tipo de dato. Las relaciones entre tablas se muestran por medio de líneas de colores.

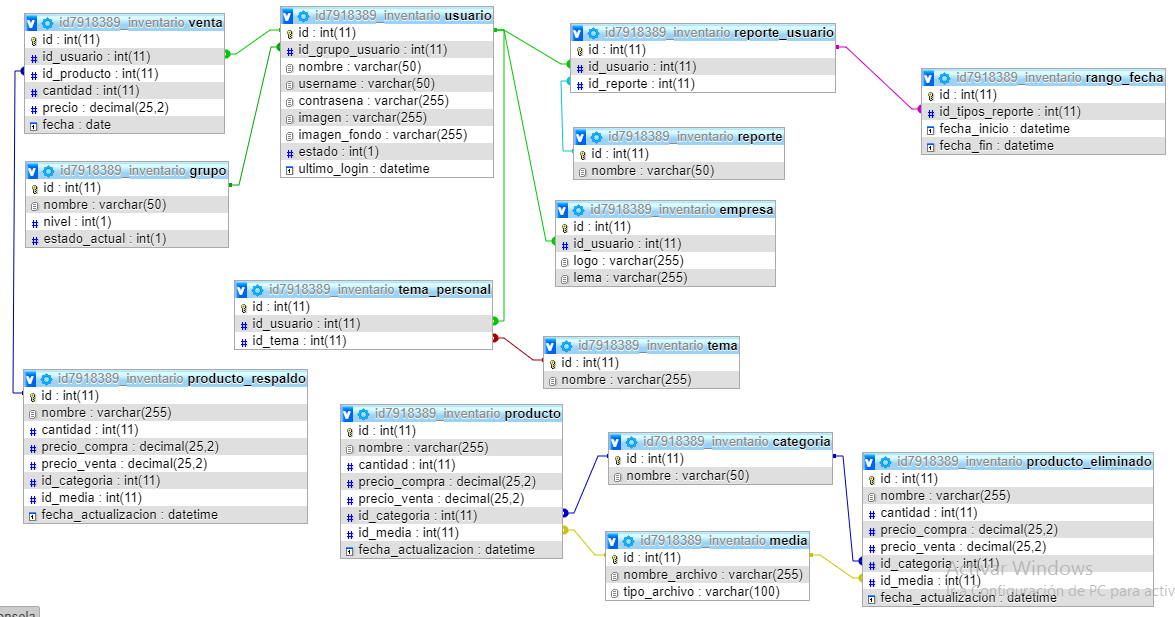


Ilustración 4 Diagrama de tablas de base de datos

Interfaces con el usuario

Pertenecen a la aplicación web.

Inicio de sesión: es un formulario centrado en la pantalla con título y subtítulo, centrados también. Debajo hay 2 campos, 1 para usuario y otro para contraseña. Hasta el final tiene un botón “Entrar” situado a la derecha. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 5.



Ilustración 5 Interfaz de inicio de sesión

Página principal: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se tiene un recuadro verde con un mensaje de bienvenida y finalmente un recuadro con el contenido de la página. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 6.



Ilustración 6 Interfaz de página principal

Panel de control: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se hayan 4 recuadros con un icono y un agregado de color; uno a la derecha del otro con la información de los usuarios actuales, el número de categorías de productos, el número de productos registrados y el número de ventas totales. Debajo se encuentran 5 paneles uno a la derecha del otro con la información en forma de tabla de los productos más vendidos, últimas ventas, productos recientemente añadidos, productos con bajo stock y reportes de ventas automáticos. El panel del administrador puede ver todo el menú visto en la imagen, el usuario especial no puede ver la pestaña “Accesos” y el usuario solo puede ver las pestañas relacionadas a ventas. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 7.

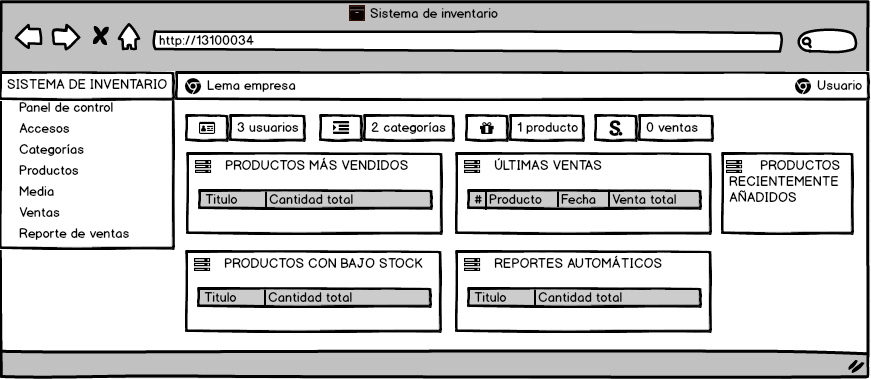


Ilustración 7 Interfaz de panel de control

Grupos: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se tiene un panel con un botón en la parte superior derecha para agregar grupo. Dentro del mismo panel hay una tabla con los datos del grupo y un par de botones por cada fila a excepción de una para editar o borrar grupos. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 8.

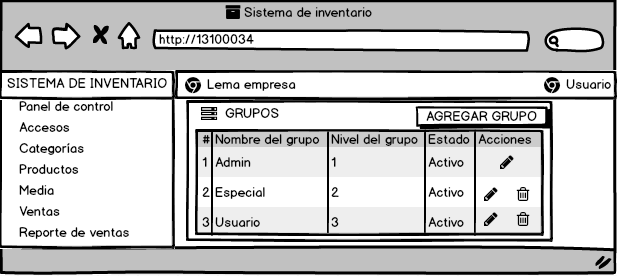


Ilustración 8 Interfaz de grupos

Agregar grupos: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se tiene un formulario situado a la izquierda y que contiene un título centrado, debajo hay un cuadro de texto para el nombre de grupo, debajo para elegir el nivel de grupo, debajo el estado del grupo y finalmente un botón azul “Guardar”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 9.



Ilustración 9 Interfaz de agregar grupos

Editar grupos: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se tienen todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se tiene un formulario situado a la izquierda y que contiene un título centrado, debajo un cuadro de texto para el nombre de grupo, debajo para elegir el nivel de grupo, debajo el estado del grupo y finalmente un botón azul “Actualizar”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 10.



Ilustración 10 Interfaz de editar grupos

Usuarios: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se tiene un panel con un botón en la parte superior derecha para agregar usuario. Dentro del mismo panel hay una tabla con los datos del usuario y un par de botones por cada fila a excepción de una para editar o borrar usuarios. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 11.

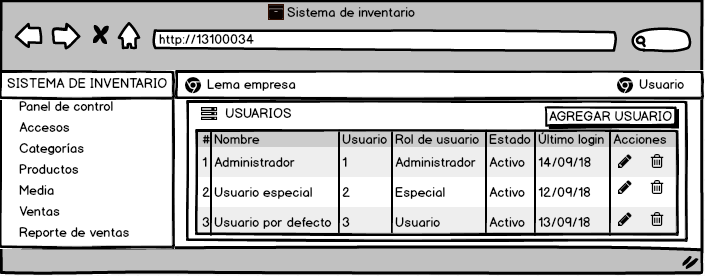


Ilustración 11 Interfaz de usuarios

Agregar usuarios: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se tienen todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se tiene un formulario situado a la izquierda y que contiene un título centrado, debajo un cuadro de texto para el nombre de la persona, debajo un cuadro de texto para el nombre de usuario, debajo un cuadro de texto para la contraseña, debajo una lista desplegable para seleccionar el rol de usuario finalmente un botón azul “Guardar”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 12.

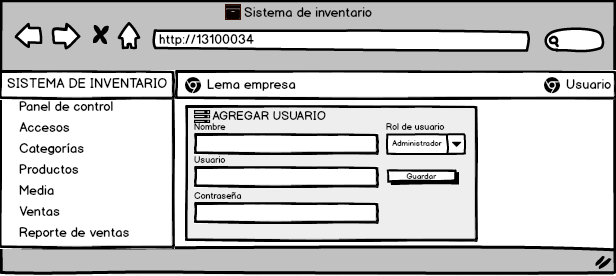


Ilustración 12 Interfaz de agregar usuarios

Editar usuarios: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo está el contenido de la página. Debajo hay un formulario situado a la izquierda y que contiene un título centrado con el nombre de la cuenta, debajo un cuadro de texto para el nombre de grupo, debajo para elegir el nivel de grupo, debajo el estado del grupo y finalmente un botón azul “Actualizar”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 13.

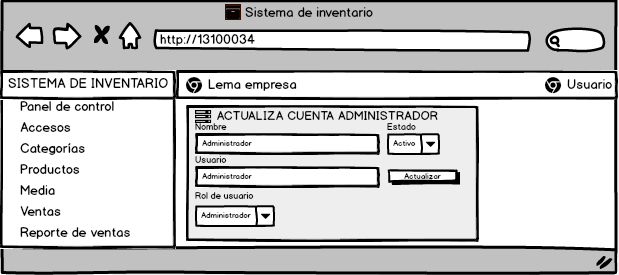


Ilustración 13 Interfaz de editar usuarios

Categoría: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se tienen todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se encuentra el contenido de la página. De izquierda a derecha hay un formulario para agregar una categoría, debajo hay un cuadro de texto para ingresar el nombre y finalmente un botón azul “Agregar categoría”. Lo siguiente es un panel donde hay una tabla con el nombre de las categorías, con cada fila con un par de botones para editar o borrar la categoría. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 14.

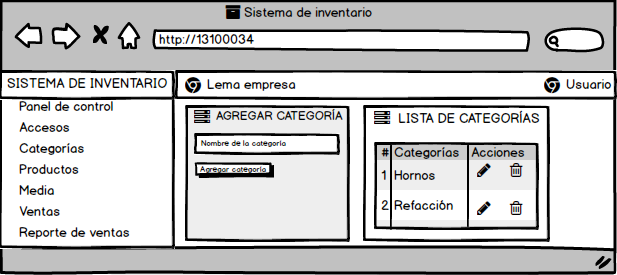


Ilustración 14 Interfaz de categorías

Editar categoría: a la izquierda se tiene un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo está el contenido de la página. Se muestra un panel orientado horizontalmente hacia la izquierda con un título que muestra que producto se va a actualizar. Debajo hay un cuadro de texto con el nombre del producto y finalmente un botón azul “Actualizar categoría”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 15.

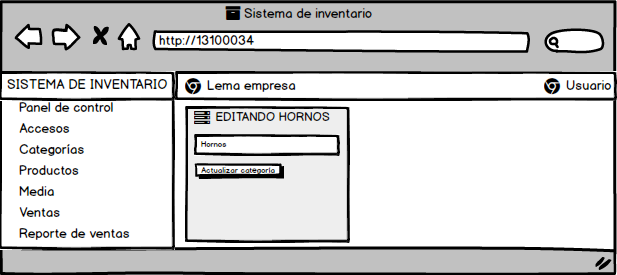


Ilustración 15 Interfaz de editar categoría

Productos: a la izquierda hay un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo se encuentran todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se localiza un panel con un botón en la parte superior derecha para agregar productos. Dentro del mismo panel hay una tabla con los datos del producto y un par de botones por cada fila a excepción de una para editar o borrar productos. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 16.



Ilustración 16 Interfaz de productos

Agregar productos: a la izquierda se encuentra un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un formulario centrado que contiene un título, debajo un cuadro de texto para la descripción del producto, debajo una lista desplegable para seleccionar una categoría, a la derecha una lista desplegable para seleccionar una imagen. Debajo un cuadro de texto para ingresar la cantidad de existencia, a la derecha 2 cuadros de texto para el precio de compra y los decimales respectivamente. Debajo 2 cuadros de texto para el precio de venta y los decimales respectivamente. Debajo un botón rojo orientado horizontalmente a la izquierda que dice “Agregar producto”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 17.

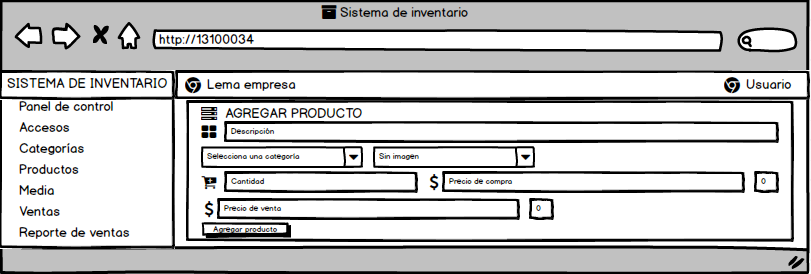


Ilustración 17 Interfaz de agregar productos

Editar productos: a la izquierda está un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se encuentra un formulario centrado que contiene un título, debajo un cuadro de texto para la descripción del producto, debajo una lista desplegable para seleccionar una categoría, a la derecha una lista desplegable para seleccionar una imagen. Debajo un cuadro de texto para editar la cantidad de existencia, a la derecha 2 cuadros de texto para el precio de compra y los decimales respectivamente. Debajo 2 cuadros de texto para el precio de venta y los decimales respectivamente. Debajo un botón rojo orientado horizontalmente a la izquierda que dice “Actualizar”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 18.



Ilustración 18 Interfaz de editar producto

Productos eliminados: a la izquierda se encuentra un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo se localiza un panel con un botón en la parte superior derecha para agregar productos. Dentro del mismo panel hay una tabla con los datos del producto eliminado y un par de botones por cada fila a excepción de una para restaurar o borrar productos eliminados definitivamente. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 19.



Ilustración 19 Interfaz de productos eliminados

Media: a la izquierda se encuentra un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un panel con un título y a la derecha un par de botones: uno azul “Ningún archivo seleccionado” y otro “Subir”. Debajo hay una tabla con los datos de cada imagen subida con un botón para eliminar alguna imagen. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 20.



Ilustración 20 Interfaz de media

Agregar múltiples registros: a la izquierda se localiza un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un panel con un título y un subtítulo con las categorías de productos existentes y debajo de eso está un cuadro de texto para agregar otra categoría con 1 par de botones para agregar o quitar cuadros de texto y finalmente un botón para agregar la o las categorías. Debajo hay un panel con un título y un subtítulo con los productos existentes y debajo de eso está un formulario para agregar otro producto con 1 par de botones para agregar o quitar formularios y finalmente un botón para agregar el o los productos. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 21.

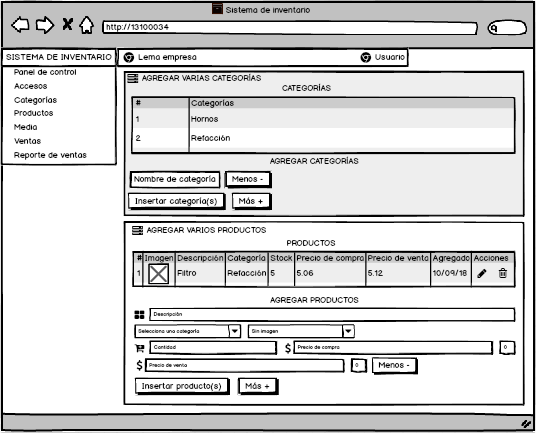


Ilustración 21 Interfaz de agregar múltiples registros

Ventas: a la izquierda está un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un panel con un título y a la derecha un botón “Agregar venta”. Debajo una tabla con los datos de la venta realizada. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 22.

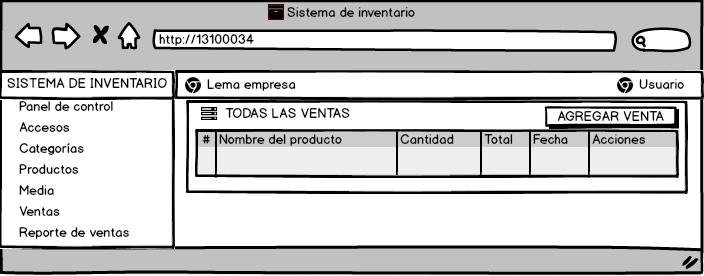


Ilustración 22 Interfaz de ventas

Agregar venta: a la izquierda se encuentra un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un botón azul “Búsqueda” con un cuadro de texto a la derecha para las búsquedas. Debajo hay un panel con un título. Debajo una tabla con los datos del producto. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 23.

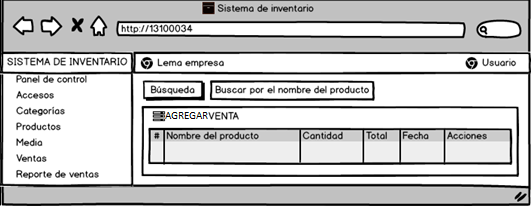


Ilustración 23 Interfaz de agregar venta

Editar venta: a la izquierda se encuentra un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un panel con un título. Debajo una tabla con los datos del producto ya vendido. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 24.

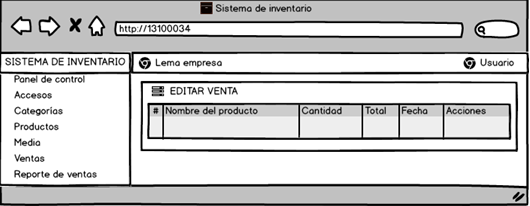


Ilustración 24 Interfaz de editar venta

Reporte de ventas: a la izquierda hay un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un recuadro con un pequeño título. Dentro del recuadro hay un par de cuadros de texto para seleccionar fechas. Por último, un botón azul “Generar reporte”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 25.



Ilustración 25 Interfaz de reporte de ventas

Ventas mensuales: a la izquierda hay un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un panel con un título y una tabla dentro con los datos de las ventas mensuales. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 26.

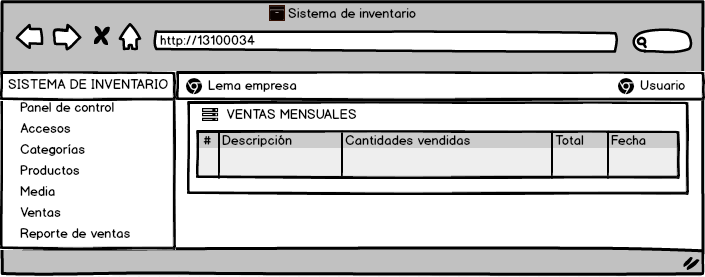


Ilustración 26 Interfaz de ventas mensuales

Ventas diarias: a la izquierda hay un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un panel con un título y una tabla dentro con los datos de las ventas diarias. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 27.

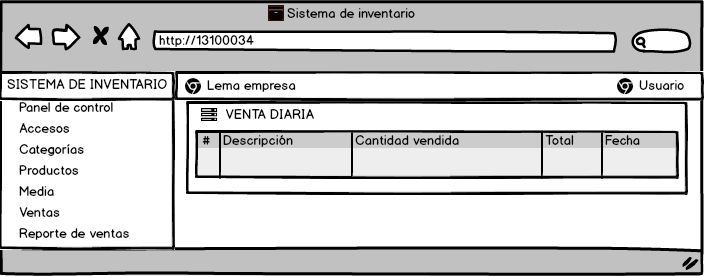


Ilustración 27 Interfaz de ventas diarias

Perfil: a la izquierda hay un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un recuadro con una imagen centrada y redondeada, que vendría ser la imagen de usuario. Justo debajo estaría el nombre de usuario y finalmente un botón con un link para editar el perfil. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 28.



Ilustración 28 Interfaz de perfil

Editar perfil: a la izquierda hay un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay 5 paneles para cambiar foto de perfil, cambiar imagen de fondo, cambiar tema, cambiar logo o lema y finalmente un panel con un formulario con un par de campos de texto para nombre y usuario, debajo un par de botones a los extremos, el de la izquierda de color azul “Actualizar” y el de la derecha “Cambiar contraseña”. Los 4 primeros paneles cuentan con un selector de imagen y debajo un par de botones amarillos a los costados para agregar o quitar. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 29.

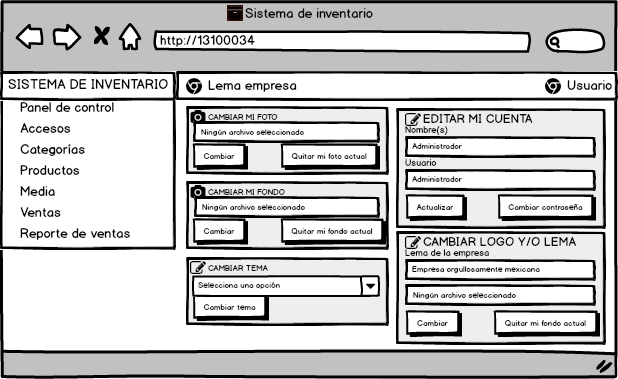


Ilustración 29 Interfaz de editar perfil

Cambiar contraseña: a la izquierda hay un menú donde en la parte superior se encuentra un título “Sistema de inventario” en letras mayúsculas y con un fondo verde, debajo están todas las opciones del panel de control. A la derecha y en la parte superior se encuentra el logo de la empresa, después el lema de la empresa, la fecha actual y por último un botón desplegable con una imagen de perfil y las opciones para el usuario. Debajo hay un cuadro orientado horizontalmente hacia la izquierda; cuenta con un título, 2 cuadros de texto para la nueva y antigua contraseña. Finalmente hay un botón azul “Cambiar”. Tiene un icono de folder en la pestaña del navegador. Se observa en la ilustración 30.

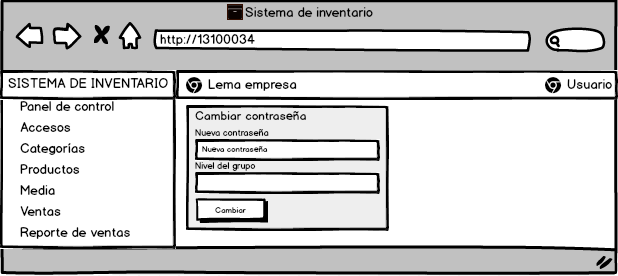


Ilustración 30 Interfaz de cambiar contraseña

# IV. DESARROLLO Y EVALUACIÓN DEL PROYECTO

## Clase para subir imágenes

Esta clase *“class Media”* se ubica en el archivo *“subir.php”* en el directorio *“incluye”*. Esta clase sirve para poder subir imágenes a un directorio y agregar la referencia a la base de datos. Tiene los siguientes atributos públicos *“$infoImagen”*, *“$nombreArchivo”*, *“$tipoArchivo”*, *“$pathTempArchivo”*, *“$pathUsuario”*, *“$pathProductos”*, *“$fondoPath”*, *“$logoPath”*, *“$errores”*, *“$subir\_errores”* con los siguientes valores: 0 => 'No hay error, el archivo se subió con éxito.', 1 => 'El archivo subido excede la carga máxima en php.ini.', 2 => 'EL archivo subido excede la carga máxima especificada en el formulario HTML.', 3 => 'EL archivo fue parcialmente subido.', 4 => 'Ningún archivo fue subido.', 6 => 'Folder temporal perdido.', 7 => 'Fallo al escribir en el disco.', 8 => 'Una extensión PHP detuvo la subida.'. Y por último la variable *“$subir\_extensiones”* con los siguientes valores: 'gif', 'jpg', 'jpeg', 'png'. En la parte de abajo se muestra el código correspondiente.

*class Media {*

*public $infoImagen;*

*public $nombreArchivo;*

*public $tipoArchivo;*

*public $pathTempArchivo;*

*public $pathUsuario = SITE\_ROOT.DS.'..'.DS.'cargas/usuarios';*

*public $pathProductos = SITE\_ROOT.DS.'..'.DS.'cargas/productos';*

*public $fondoPath = SITE\_ROOT.DS.'..'.DS.'cargas/fondos';*

*public $logoPath = SITE\_ROOT.DS.'..'.DS.'cargas/logo';*

*public $errores = array();*

*public $subir\_errores = array(*

*0 => 'No hay error, el archivo se subió con éxito.',*

*1 => 'El archivo subido excede la carga máxima en php.ini.',*

*2 => 'EL archivo subido excede la carga máxima especificada en el formulario HTML.',*

*3 => 'EL archivo fue parcialmente subido.',*

*4 => 'Ningún archivo fue subido.',*

*6 => 'Folder temporal perdido.',*

*7 => 'Fallo al escribir en el disco.',*

*8 => 'Una extensión PHP detuvo la subida.'*

*);*

*public $subir\_extensiones = array(*

*'gif',*

*'jpg',*

*'jpeg', 'png',);*

## Función para actualizar imagen de usuario

Esta función *“private function actualizar\_imagenUsuario($id)”* actualiza la imagen de perfil que escoge el usuario y que se verá en su perfil; esta función recibe como parámetro el identificador del usuario llamado *“$id”*. La variable *“$sql”* tiene como valor la sentencia SQL para actualizar el nombre de la imagen de perfil de acuerdo con el parámetro *“$id”* y se ejecuta esta sentencia SQL por medio de la variable global *“$db”* que operará la base de datos por medio de su método *“query”* de PHP. Si la actualización tuvo éxito, retorna el valor booleano verdadero, caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*private function actualizar\_imagenUsuario($id) {*

*global $db;*

*$sql = "UPDATE usuario SET";*

*$sql .=" imagen='{$db->escape($this->fileName)}'";*

*$sql .=" WHERE id='{$db->escape($id)}'";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para borrar imagen de fondo

Esta función *“public function borrar\_fondo($id)”* borra el fondo del usuario que se ve en la aplicación, para ello recibe como parámetro el identificador del usuario, conocido de ahora en adelante como la variable *“$id”*. Primero encuentra el nombre de la imagen de fondo del correspondiente usuario, por medio de la función *“encontrar\_por\_id”*. Esta función recibe como parámetro el nombre de la tabla *“usuarios”* y la variable *“$id”*;el valor que devuelve esta función es almacenado en la variable *“$image”*.Finalmente se procede a borrar la imagen de fondo del directorio por medio de la función *“unlink”*, que tomará como parámetro el atributo *“fondoPath”* y la variable *“$image”*. En la parte de abajo se muestra el código correspondiente.

*public function borrar\_fondo($id) {*

*$image = encontrar\_por\_id('usuarios',$id);*

*unlink($this->fondoPath.'/'.$image['imagen\_fondo']);*

*}*

## Función para borrar imagen de perfil de usuario

Esta función *“public function borrar\_imagen($id)”* borra la imagen de perfil del usuario que se ve en la aplicación, para ello recibe como parámetro el identificador del usuario, conocido de ahora en adelante como la variable *“$id”*. Primero encuentra el nombre de la imagen de perfil del correspondiente usuario, por medio de la función *“encontrar\_por\_id”*. Esta función recibe como parámetro el nombre de la tabla *“usuario”* y la variable *“$id”*;el valor que devuelve esta función es almacenado en la variable *“$image”*.Finalmente se procede a borrar la imagen de perfil del directorio por medio de la función *“unlink”*, que tomará como parámetro el atributo *“pathUsuario”* y la variable *“$image”*. En la parte de abajo se muestra el código correspondiente.

*public function borrar\_imagen($id) {*

*$image = encontrar\_por\_id('usuarios',$id);*

*unlink($this->pathUsuario.'/'.$image['imagen']);*

*}*

## Función para borrar el logo de la empresa

Esta función *“public function borrar\_logo()”* borra el logo de la empresa asociado a la aplicación. Primero encuentra el nombre del logo, por medio de la función *“encontrar\_por\_id”*. Esta función recibe como parámetro el nombre de la tabla *“empresa”* y el valor numérico 1;el valor que devuelve esta función es almacenado en la variable *“$image”*.Finalmente se procede a borrar el logo del directorio por medio de la función *“unlink”*, que tomará como parámetro el atributo *“fondoPath”* y la variable *“$image”*. En la parte de abajo se muestra el código correspondiente.

*public function borrar\_logo() {*

*$image = encontrar\_por\_id('empresa',1);*

*unlink($this->logoPath.'/'.$image['logo']);*

*}*

## Función para eliminar imagen de perfil del usuario

Esta función *“public function destruir\_imagen\_usuario($id)”* borra la imagen de perfil del usuario que se ve en la aplicación, para ello recibe como parámetro el identificador del usuario, conocido de ahora en adelante como la variable *“$id”*. Primero encuentra el nombre de la imagen de perfil del correspondiente usuario, por medio de la función *“encontrar\_por\_id”*. Esta función recibe como parámetro el nombre de la tabla *“usuario”* y la variable *“$id”*;el valor que devuelve esta función es almacenado en la variable *“$image”*.Finalmente se procede a borrar la imagen de perfil del directorio por medio de la función *“unlink”*, que tomará como parámetro el atributo *“userPath”* y la variable *“$image”*. En caso de que tenga una imagen de perfil con el nombre “no\_image.jpg” no se borrará la imagen. En cualquier caso, solo devolverá el valor booleano verdadero. En la parte de abajo se muestra el código correspondiente.

*public function destruir\_imagen\_usuario($id) {*

*$image = encontrar\_por\_id('usuarios',$id);*

*if($image['image'] === 'no\_image.jpg') {*

*return true;*

*} else {*

*unlink($this->userPath.'/'.$image['image']);*

*return true; } }*

## Función para eliminar la imagen de producto

Esta función *public function destruir\_media($id,$file\_name)* borra el nombre de la imagen y su extensión en la base de datos, así como la imagen dentro del directorio correspondiente. Esta función recibe como parámetro el identificador de la imagen *“$id”*, así como el nombre de la imagen a borrar *“file\_name”*. Primero se iguala el valor de la propiedad *“fileName”* con el del valor de parámetro *“file\_name”*. Si la propiedad *fileName* está vacía, agrega al arreglo con lista de errores “*errors”* el siguiente mensaje: “Falta el archivo de foto.” para inmediatamente retornar el valor booleano falso. Se checa además si el parámetro *“$id”* está vacío, si es el caso, agrega al arreglo con lista de errores *“errors”* el siguiente mensaje: “ID de foto ausente.” para inmediatamente retornar el valor booleano falso. Si no hay ningún error se procede a intentar borrar la referencia a la imagen en la base de datos por medio de la función *“borrar\_por\_id”* que recibe como parámetro el nombre de la tabla media y el parámetro *“$id”*. Si tuvo éxito la función *“borrar\_por\_id”* se procede a borrar la imagen del directorio correspondiente por medio de la función *“unlink”* de PHP que recibe como parámetro la ruta conformada por los atributos *“pathProductos”* y *“fileName”*, para inmediatamente retornar el valor booleano verdadero. Si la falla la función *“borrar\_por\_id”* agrega al arreglo con lista de errores *“errors”* el siguiente mensaje: “Se ha producido un error en la eliminación de fotografías.” para inmediatamente retornar el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*public function destruir\_media($id,$file\_name) {*

*$this->fileName = $file\_name;*

*if(empty($this->fileName)) {*

*$this->errors[] = "Falta el archivo de foto.";*

*return false;*

*}*

*if(!$id) {*

*$this->errors[] = "ID de foto ausente.";*

*return false;*

*}*

*if(borrar\_por\_id('media',$id)) {*

*unlink($this->pathProductos.'/'.$this->fileName);*

*return true;*

*} else {*

*$this->error[] = "Se ha producido un error en la eliminación de fotografías.";*

*return false;*

*}*

*}*

## Función para checar la extensión de una imagen

Esta función *“public function ext\_archivo($nombreArchivo)”* revisa que la extensión de la imagen a subir sea válida; esta es pasada por parámetro con el nombre de la imagen incluido y en su conjunto se llama *“nombreArchivo”*. Primero se crea una variable *“$ext”* cuyo valor será la extensión de la imagen. Para obtener la extensión se deberá recortar de la función *“$nombreArchivo”*. Finalmente se compara el valor de *“$ext”* con el arreglo *“subir\_extensiones”* que contiene todas las extensiones posibles y retornará el valor booleano verdadero en caso de que coincida. En la parte de abajo se muestra el código correspondiente.

*public function ext\_archivo($nombreArchivo) {*

*$ext = strtolower(substr( $nombreArchivo, strrpos( $nombreArchivo, '.' ) + 1 ) );*

*if(in\_array($ext, $this->subir\_extensiones)) {*

*return true;*

*}*

*}*

## Función para insertar imagen de fondo

Esta función *“private function insertar\_fondo($id)”* actualiza el fondo que escoge el usuario y que se verá en el fondo de la aplicación; esta función recibe como parámetro el identificador del usuario llamado *“$id”*. La variable *“$sql”* tiene como valor la sentencia SQL para actualizar el nombre de la imagen de fondo de acuerdo con el parámetro *“$id”* y se ejecuta esta sentencia SQL por medio de la variable global *“$db”* que operará la base de datos por medio su método *“query”* de PHP. Si la actualización tuvo éxito, retorna el valor booleano verdadero, caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*private function insertar\_fondo($id) {*

*global $db;*

*$sql = "UPDATE usuario SET";*

*$sql .=" imagen\_fondo='{$db->escape($this->fileName)}'";*

*$sql .=" WHERE id='{$db->escape($id)}'";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para insertar el logo de la empresa

Esta función *“private function insertar\_logo()”* actualiza el logo de la empresa que se verá en la cabecera de la aplicación. Primero encuentra el nombre del logo único que tiene la aplicación, por medio de la función *“encontrar\_por\_id”* y la guarda en la variable *“$image”*. Se procede a borrar del directorio el logo, por medio de la función *“unlink”,* pasando como parámetro el atributo *“logoPath”* y la variable *“$image”.* Una vez hecha la eliminación del logo, la variable *“$sql”* tiene como valor la sentencia SQL para actualizar el nombre del logo y se ejecuta por medio de la variable global *“$db”* que operará la base de datos por medio de su método *“query”* de PHP. Si la actualización tuvo éxito, retorna el valor booleano verdadero, caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*private function insertar\_logo() {*

*$image = encontrar\_por\_id('empresa',1);*

*unlink($this->logoPath.'/'.$image['logo']);*

*global $db;*

*$sql = "UPDATE empresa SET";*

*$sql .=" logo='{$db->escape($this->fileName)}'";*

*$sql .=" WHERE empresa.id=1";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para insertar imagen de producto

Esta función *“private function insertar\_media()”* registra el nombre de la imagen y su extensión en la base de datos. La variable *“$sql”* tiene como valor la sentencia SQL para insertar el nombre de la imagen, así como su extensión para posteriormente ejecutar esta sentencia SQL por medio de la variable global *“$db”* que operará la base de datos por medio de su método *“query”* de PHP. Si la actualización tuvo éxito, retorna el valor booleano verdadero, caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*private function insertar\_media() {*

*global $db;*

*$sql = "INSERT INTO media (nombre\_archivo,tipo\_archivo)";*

*$sql .=" VALUES ";*

*$sql .="(*

*'{$db->escape($this->fileName)}',*

*'{$db->escape($this->fileType)}'*

*)";*

*return ($db->query($sql) ? true : false);*

*}*

## Función para subir una imagen al servidor

Esta función *“public function proceso()”* supervisa que la subida de la imagen al directorio correspondiente sea correcta. Primero se checa si el atributo *“errors”* no está vacío, siendo el caso, retornara el valor booleano falso. Se checa si el atributo *“fileName”* que tiene el nombre del archivo está vacío o el atributo *“fileTempPath”* que tiene la ruta temporal de la imagen si es el caso actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “La ubicación del archivo no está disponible.” y retorna el valor booleano falso. Se checa si el atributo *“productPath”* que tiene la ruta de la imagen, tiene permisos de escritura y lectura y si no es el caso, actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “Debe tener permisos de escritura!” y retorna el valor booleano falso. Se checa por última vez si el archivo existe por medio de la función *“file\_exists”* de PHP, si no existe, actualizará el atributo de tipo arreglo *errors* de la clase actual con el mensaje de error: “El archivo *“fileName”* realmente existe” y retorna el valor booleano falso. Si no hay ningún error devolverá el valor booleano verdadero. En la parte de abajo se muestra el código correspondiente.

*public function proceso() {*

*if(!empty($this->errors)):*

*return false;*

*elseif(empty($this->fileName) || empty($this->fileTempPath)):*

*$this->errors[] = "La ubicación del archivo no está disponible.";*

*return false;*

*elseif(!is\_writable($this->productPath)):*

*$this->errors[] = $this->productPath." Debe tener permisos de escritura!!!.";*

*return false;*

*elseif(file\_exists($this->productPath."/".$this->fileName)):*

*$this->errors[] = "El archivo {$this->fileName} realmente existe.";*

*return false;*

*else:*

*return true;*

*endif;*

*}*

## Función para subir el logo de la empresa al servidor

Esta función *“public function proceso\_logo()”* supervisa que la subida de la imagen del logo de la empresa al directorio correspondiente sea correcta y colocando un registro en la base de datos por medio de la función *“insertar\_logo”*. Esta función borra el valor del atributo *“fileTempPath”* y retorna el valor booleano verdadero. Si hay algún error en la carga de la imagen se actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.” y retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*public function proceso\_logo() {*

*if(!empty($this->errors)) {*

*return false;*

*}*

*if(empty($this->fileName) || empty($this->fileTempPath)) {*

*$this->errors[] = "la ubicación del archivo no se encuentra disponible.";*

*return false;*

*}*

*if(!is\_writable($this->logoPath)) {*

*$this->errors[] = $this->logoPath." debe tener permisos de escritura!!!.";*

*return false;*

*}*

*/\**

*if(file\_exists($this->logoPath."/".$this->fileName)) {*

*$this->errors[] = "El archivo {$this->fileName} realmente existe.";*

*return false;*

*}*

*\*/*

*if(move\_uploaded\_file($this->fileTempPath,$this->logoPath.'/'.$this->fileName)) {*

*if($this->insertar\_logo()) {*

*unset($this->fileTempPath);*

*return true;*

*}*

*} else {*

*$this->errors[] = "Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.";*

*return false;*

*}*

*}*

## Función para subir una imagen de productos al servidor

Esta función *“public function proceso\_media()”* supervisa que la subida de la imagen de producto al directorio correspondiente sea correcta y colocando un registro en la base de datos por medio de la función *“insertar\_media”*. Esta función borra el valor del atributo *“fileTempPath”* y retorna el valor booleano verdadero. Si hay algún error en la carga de la imagen se actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.” y retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*public function proceso\_media() {*

*if(!empty($this->errors)) {*

*return false;*

*}*

*if(empty($this->fileName) || empty($this->fileTempPath)) {*

*$this->errors[] = "La ubicación del archivo no se encuentra disponible.";*

*return false;*

*}*

*if(!is\_writable($this->pathProductos)) {*

*$this->errors[] = $this->pathProductos." Debe tener permisos de escritura.";*

*return false;*

*}*

*if(file\_exists($this->pathProductos."/".$this->fileName)) {*

*$this->errors[] = "El archivo {$this->fileName} realmente existe.";*

*return false;*

*}*

*if(move\_uploaded\_file($this->fileTempPath,$this->pathProductos.'/'.$this->fileName)) {*

*if($this->insertar\_media()) {*

*unset($this->fileTempPath);*

*return true;*

*}*

*} else {*

*$this->errors[] = "Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.";*

*return false;*

*}*

*}*

## Función para subir una imagen de perfil al servidor

Esta función *“public function proceso\_usuario($id)”*. Esta función no sube imágenes a cualquier directorio, sino al directorio de imágenes de perfil de los usuarios. Además, revisa el siguiente posible error: si el identificador del correspondiente usuario está vacío se actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “ID de usuario ausente.” y retorna el valor booleano falso. Finalmente, si no hay errores se creará un nombre de imagen aleatorio por medio de la función *“randString”* y se guardará en la variable *“$new\_name”* y se asigna al atributo *“fileName”*. Se procede a destruir la imagen de perfil del usuario por medio de la función *“destruir\_imagen\_usuario”* pasando el parámetro que recibe la función *“proceso\_usuario”* como parámetro a la función *“destruir\_imagen\_usuario”*. Una vez borrada la imagen de perfil de usuario se procede a subir la imagen al directorio donde se encuentran las imágenes de perfil de usuarios, por medio de la función *“move\_uploaded\_file”* de PHP, luego se procede a actualizar la referencia en la base de datos por medio de la función *“actualizar\_imagenUsuario”* para después borrar el valor del atributo *“fileTempPath”* y retorna el valor booleano verdadero. Si hay algún error en la carga de la imagen se actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.” y retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*public function proceso\_usuario($id) {*

*if(!empty($this->errors)) {*

*return false;*

*}*

*if(empty($this->fileName) || empty($this->fileTempPath)) {*

*$this->errors[] = "La ubicación del archivo no estaba disponible.";*

*return false;*

*}*

*if(!is\_writable($this->pathUsuario)) {*

*$this->errors[] = $this->pathUsuario." Debe tener permisos de escritura";*

*return false;*

*}*

*if(!$id) {*

*$this->errors[] = " ID de usuario ausente.";*

*return false;*

*}*

*$ext = explode(".",$this->fileName);*

*$new\_name = randString(8).$id.'.' . end($ext);*

*$this->fileName = $new\_name;*

*if($this->destruir\_imagen\_usuario($id)) {*

*if(move\_uploaded\_file($this->fileTempPath,$this->pathUsuario.'/'.$this->fileName)) {*

*if($this->actualizar\_imagenUsuario($id)) {*

*unset($this->fileTempPath);*

*return true;*

*}*

*} else {*

*$this->errors[] = "Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.";*

*return false;*

*}*

*}*

*}*

## Función para subir una imagen de fondo al servidor

Esta función *“public function process\_fondo($id)”* supervisa que la subida de la imagen de fondo al directorio correspondiente sea correcta y colocando un registro en la base de datos por medio de la función *“insertar\_fondo”*. Esta función borra el valor del atributo *“fileTempPath”* y retorna el valor booleano verdadero. Si hay algún error en la carga de la imagen se actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.” y retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*public function process\_fondo($id) {*

*if(!empty($this->errors)) {*

*return false;*

*}*

*if(empty($this->fileName) || empty($this->fileTempPath)) {*

*$this->errors[] = "la ubicación del archivo no se encuentra disponible.";*

*return false;*

*}*

*if(!is\_writable($this->fondoPath)) {*

*$this->errors[] = $this->fondoPath." debe tener permisos de escritura!!!.";*

*return false;*

*}*

*if(move\_uploaded\_file($this->fileTempPath,$this->fondoPath.'/'.$this->fileName)) {*

*if($this->insertar\_fondo($id)) {*

*unset($this->fileTempPath);*

*return true;*

*}*

*} else {*

*$this->errors[] = "Error en la carga del archivo, posiblemente debido a permisos incorrectos en la carpeta de carga.";*

*return false;*

*}*

*}*

## Función para revisar la extensión de la imagen

Esta función *“public function subir($file)”* revisa que el nombre de la imagen, pasado por parámetro este correcto; este parámetro se llama *“$file”*. Primero se checa si *“$file”* está vacío, si es el caso actualizará el atributo de tipo arreglo *“$errors”* de la clase actual con el mensaje de error: “Ningún archivo subido.” y retornará el valor booleano falso. Si la variable *“$file”* en su atributo error tiene un valor diferente de 0 actualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error al intentar subir el archivo y retornará el valor booleano falso. Si la variable *“$file”* tiene una extensión incorrecta y esto se verifica con la función *“ext\_archivo”* seactualizará el atributo de tipo arreglo *“errors”* de la clase actual con el mensaje de error: “Formato de archivo incorrecto.” Finalmente, si no hay ningún error guarda en los siguientes atributos: *“infoImagen”*, *“nombreArchivo”*, *“tipoImagen”*, *“pathTempArchivo”* con los siguientes valores: el tamaño del archivo, el nombre del archivo, el tipo del archivo y la url completa del archivo respectivamente. En la parte de abajo se muestra el código correspondiente.

*public function subir($file) {*

*if(!$file || empty($file) || !is\_array($file)):*

*$this->errors[] = "Ningún archivo subido.";*

*return false;*

*elseif($file['error'] != 0):*

*$this->errors[] = $this->upload\_errors[$file['error']];*

*return false;*

*elseif(!$this->ext\_archivo($file['name'])):*

*$this->errors[] = 'Formato de archivo incorrecto. ';*

*return false;*

*else:*

*$this->imageInfo = getimagesize($file['tmp\_name']);*

*$this->fileName = basename($file['name']);*

*$this->fileType = $this->imageInfo['mime'];*

*$this->fileTempPath = $file['tmp\_name'];*

*return true;*

*endif;*

*}*

## Clase que administra la base de datos

Esta clase *“class MySqli\_DB”* se ubica en el archivo *“baseDatos.php”* en el directorio *“incluye”*. Esta clase modela la base de datos con las funciones más importantes de esta. Cuenta con 2 atributos: *“$con”* que es privado para establecer la conexión con la base de datos y el último que es privado *“$query\_id”* que es el identificador de la consulta. En la parte de abajo se muestra el código correspondiente.

*class MySqli\_DB {*

*private $con;*

*public $query\_id;*

*function \_\_construct() {*

*$this->db\_connect();*

*}*

## Función constructora de la clase

Este constructor *“function \_\_construct()”* utiliza una rutina pública *“db\_connect”* para realizar una conexión a la base de datos.Se invoca automáticamente al crear un objeto de la clase a la que pertenece esta función. En la parte de abajo se muestra el código correspondiente.

*function \_\_construct() {*

*$this->db\_connect();*

*}*

## Función para saber si hubo filas actualizadas

Esta función *“public function affected\_rows()”* retorna el valor booleano verdadero en caso de tener éxito al hacer una modificación a una tabla, caso contrario retorna el valor booleano falso. Mantiene la misma sintaxis que la función *“fetch\_array”*. Utiliza la función de PHP *“mysqli\_affected\_rows”*. Esta función no tiene parámetro, utiliza el atributo *“con”*. En la parte de abajo se muestra el código correspondiente.

*public function affected\_rows() {*

*return mysqli\_affected\_rows($this->con);*

*}*

## Función para conectarse a la base de datos

Esta función *“public function db\_connect()“* realiza la conexión a la base de datos y envía un mensaje de error en caso de fallo. Primero el atributo *“con“* utilizará la función de PHP *“mysqli\_connect “*para conectarse a la base de datos. A esta función de PHP se lepasa como parámetros las constantes *“DB\_HOST“*, *“DB\_USER“* y *“DB\_PASS“*. Si el atributo *“con“* está vacío entonces la conexión a la base de datos falló y se procede a mostrar un mensaje: “Database connection failed:” con el código del error por medio de la función de PHP *“mysqli\_connect\_error“*. Todo esto se muestra en código HTML con la función de PHP *“die“*. Si la conexión con la base de datos tuvo éxito se procede a guardar en la variable *“$select\_db “*el nombre de la base de datos utilizando el atributo *“con“* que a su vez selecciona el nombre de la base de datos por medio de la función de PHP *“$select\_db“* que a su vez recibe como parámetro la constante *“DB\_NAME“*. Si la variable *“$select\_db “*no tiene ningún valor y se procede a mostrar un mensaje: “Failed to Select Database” con el código del error por medio de la función de PHP *“mysqli\_connect\_error“*. Todo esto se muestra en código HTML con la función de PHP *“die“*. En la parte de abajo se muestra el código correspondiente.

*public function db\_connect()*

*{*

*$this->con = mysqli\_connect(DB\_HOST,DB\_USER,DB\_PASS);*

*if(!$this->con) {*

*die(" Database connection failed:". mysqli\_connect\_error());*

*} else {*

*$select\_db = $this->con->select\_db(DB\_NAME);*

*if(!$select\_db) {*

*die("Failed to Select Database". mysqli\_connect\_error());*

*}*

*}*

*}*

## Función para desconectarse de la base de datos

Esta función *“public function db\_disconnect()”* realiza la desconexión a la base de datos. Primero checa que el atributo *“con“* tenga algún valor, si es el caso utilizará la función de PHP *“mysqli\_close“* que a su vez recibirá como parámetro el atributo *“con“* para realizar la desconexión de la base de datos. Finalmente se borra el valor que tenga el atributo *“con“* por medio de la función de PHP *“unset“*. En la parte de abajo se muestra el código correspondiente.

*public function db\_disconnect() {*

*if(isset($this->con)) {*

*mysqli\_close($this->con);*

*unset($this->con);*

*}*

*}*

## Función para filtrar una cadena de texto de caracteres extraños

Esta función “*public function escape($str)”* retorna una cadena sin espacios tomando como valor la cadena que recibe por parámetro. El parámetro se llama *“$str“*. Todo se realiza por medio del atributo *“con“* que a su vez utiliza el parámetro *“$str “*por medio de su función *“real\_escape\_string“*. En la parte de abajo se muestra el código correspondiente.

*public function escape($str) {*

*return $this->con->real\_escape\_string($str);*

*}*

## Función que retorna el resultado de una consulta en forma de arreglo

Esta función “*public function fetch\_array($statement)”* recupera y retorna cada uno de los registros de la consulta SQL efectuada, que recibe como parámetro. El parámetro se llama *“$statement“* y es utilizado para que esta función retorne un array por medio de la función *“mysqli\_fetch\_array“*; que también utiliza el parámetro *“$statement“.* En la parte de abajo se muestra el código correspondiente.

*public function fetch\_array($statement) {*

*return mysqli\_fetch\_array($statement);*

*}*

## Función que retorna el resultado de una consulta en forma de arreglo asociativo

Esta función “*public function fetch\_assoc($statement)”* retorna los resultados de una consulta en forma de un arreglo asociativo. Utiliza la función de PHP *“mysqli\_fetch\_assoc“*. En la parte de abajo se muestra el código correspondiente.

*public function fetch\_assoc($statement) {*

*return mysqli\_fetch\_assoc($statement);*

*}*

## Función que retorna el resultado de una consulta en forma de arreglo de objetos

Esta función “*public function fetch\_object($statement)”* retorna los resultados de una consulta en forma de un objeto. Utiliza la función de PHP *“mysqli\_fetch\_object“*. En la parte de abajo se muestra el código correspondiente.

*public function fetch\_object($statement) {*

*return mysqli\_fetch\_object($statement);*

*}*

## Función que inserta un registro en la base de datos

Esta función “*public function insert\_id()”* retorna el valor booleano verdadero en caso de tener éxito al insertar un registro, caso contrario retorna el valor booleano falso. Mantiene la misma sintaxis que la función *“fetch\_array“*. Utiliza la función de PHP *“mysqli\_insert\_id“*. Esta función no tiene parámetro, utiliza el atributo *“con“*. En la parte de abajo se muestra el código correspondiente.

*public function insert\_id() {*

*return mysqli\_insert\_id($this->con);*

*}*

## Función que retorna el número de registros de una consulta

Esta función “*public function num\_rows($statement)”* retorna la cantidad de registros del resultado de una consulta que recibe por parámetro. Mantiene la misma sintaxis que la función *“fetch\_array“*. Utiliza la función de PHP *“mysqli\_num\_rows“*. En la parte de abajo se muestra el código correspondiente.

*public function num\_rows($statement) {*

*return mysqli\_num\_rows($statement);*

*}*

## Función que realiza una consulta

Esta función “*public function query($sql)”* realiza una consulta a la base de datos por medio de la sentencia SQL que recibe como parámetro llamado *“$sql“*. Lo primero que intenta hacer es limpiar los caracteres en blanco del parámetro *“$sql“* al inicio y al final, para después checar si está en blanco. Si no está en blanco utiliza el atributo *“query­\_id“* para guardar en él, el valor que devuelve la función de PHP *“query“*, donde esta utiliza como parámetro el valor de *“$sql“*; todo esto por medio del atributo *“con“*. Si el atributo *“query\_id“* está vacío se procede a mostrar un mensaje: “Error en esta consulta:” con la sentencia de la consulta *“$sql“*. Finalmente retorna el atributo *“query\_id“*. En la parte de abajo se muestra el código correspondiente.

*public function query($sql) {*

*if (trim($sql != "")) {*

*$this->query\_id = $this->con->query($sql);*

*}*

*if (!$this->query\_id) {*

*die("Error en esta consulta :<pre> " . $sql . "</pre>");*

*}*

*return $this->query\_id;*

*}*

## Función que retorna los valores de una consulta en forma de un arreglo asociativo

Esta función “*public function while\_loop($loop)”* retorna todos los valores del resultado de una consulta en forma de arreglo asociativo. Recibe como parámetro la variable *“$loop“* que es el resultado de una consulta. Primero crea un arreglo con nombre *“$results“*. La variable *“$results“* guardará en cada posición una posición de *“$loop“*; todo esto se realiza por medio del ciclo while. Finalmente se retorna la variable *“$results“*. En la parte de abajo se muestra el código correspondiente.

*public function while\_loop($loop){*

*global $db;*

*$results = array();*

*while ($resultado = $this->fetch\_array($loop)) {*

*$results[] = $resultado;*

*}*

*return $results;*

*}*

## Clase que administra las sesiones

Esta clase “*class Session”* se ubica en el archivo *“sesion.php”* en el directorio “*incluye”*. Esta clase nos sirve para administrar las sesiones de nuestra aplicación web. Tiene 2 atributos: el mensaje que recibe el usuario cuando intenta ingresar a su cuenta *“$mensaje“* y el indicador para dejarlo entrar o no al sistema *“$user\_is\_logged\_in“*. En la parte de abajo se muestra el código correspondiente.

*class Session {*

*public $mensaje;*

*private $user\_is\_logged\_in = false;*

## Función constructora de la clase

Esta función *“function \_\_construct()“* utiliza una rutina pública *“mensaje\_flash“* para mostrar un mensaje en caso de hacer un login correcto en la aplicación y otra rutina pública *“configuracionDeLoginUsuario“* para saber si ya está logeado o no.Se invoca automáticamente al crear un objeto de la clase a la que pertenece esta función. En la parte de abajo se muestra el código correspondiente.

*function \_\_construct() {*

*$this->mensaje\_flash();*

*$this->configuracionDeLoginUsuario();*

*}*

## Función para cerrar la sesión del usuario actual

Esta función “*public function cerrarSesion()”* cierra la sesión actual. Utiliza la función de PHP *“unset“* para borrar la sesión actual *“user\_id“*. En la parte de abajo se muestra el código correspondiente.

*public function cerrarSesion() {*

*unset($\_SESSION['user\_id']);*

*}*

## Función para revisar la sesión del usuario actual

Esta función “*private function configuracionDeLoginUsuario()”* revisa si hay una sesión activa en la aplicación web, en caso de que la haya actualiza las variables del servidor. Primero se checa si hay una sesión activa con el nombre *“user\_id“*, si hay, iguala el atributo *“user\_is\_logged\_in“* con el valor booleano verdadero; caso contrario igualará con el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*private function configuracionDeLoginUsuario() {*

*if(isset($\_SESSION['user\_id'])) {*

*$this->user\_is\_logged\_in = true;*

*} else {*

*$this->user\_is\_logged\_in = false;*

*}*

*}*

## Función que retorna el estado de la sesión del usuario actual

Esta función “*public function estaLogeado()”* retorna el estado actual de la sesión; si esta logeado o no. Retorna el atributo *“user\_is\_logged\_in“*. En la parte de abajo se muestra el código correspondiente.

*public function estaLogeado() {*

*return $this->user\_is\_logged\_in;*

*}*

## Función que crea una sesión para un usuario

Esta función “*public function login($id\_usuario)”* inicia la sesión del usuario correspondiente. Recibe como parámetro el identificador del usuario *“$id\_usuario“*. Crea una sesión *“user\_id“* con *“$\_SESSION“* de PHP con el valor de *“$id\_usuario“*. En la parte de abajo se muestra el código correspondiente.

*public function login($id\_usuario) {*

*$\_SESSION['user\_id'] = $id\_usuario;*

*}*

## Función que muestra un mensaje de error o informativo con formato

Esta función “*public function mensaje($type ='', $mensaje ='')”* le da formato de visualización al mensaje que recibe. Recibe como parámetro el tipo de mensaje *“$type“* y el mensaje a mostrar *“$mensaje“*. Primero revisa si el parámetro *“$mensaje“* está vacío por medio de la función de PHP *“empty“*; si el parámetro no está vacío se procede a limpiar los caracteres en blanco al inicio y al final del parámetro *“$type“* para después contar cada uno de los caracteres por medio de la función de PHP *“strlen“*. Si el conteo es de 1, reemplaza el valor de *“$type“*: d, i, w, s con los siguientes valores: danger, info, warning, success respectivamente. Se procede a actualizar el valor de la sesión actual con el mensaje correspondiente. Finalmente, si el conteo fue diferente de 1, retorna el valor del atributo *“$mensaje“*. En la parte de abajo se muestra el código correspondiente.

*public function mensaje($type ='', $mensaje ='') {*

*if(!empty($mensaje)) {*

*if(strlen(trim($type)) == 1){*

*$type = str\_replace( array('d', 'i', 'w','s'), array('danger', 'info', 'warning','success'), $type );*

*}*

*$\_SESSION['mensaje'][$type] = $mensaje;*

*} else {*

*return $this->mensaje;*

*}*

*}*

## Función que muestra un mensaje de error o informativo con formato instantáneo

Esta función *function mensaje\_flash()* muestra un mensaje en la aplicación web relacionado a la sesión actual. Primero se checa si la sesión actual tiene un mensaje que mostrar y si es el caso actualiza el atributo *“mensaje“* con el valor del mensaje de la sesión actual para después eliminar la sesión por medio de la función de PHP *“unset“*. En la parte de abajo se muestra el código correspondiente.

*private function mensaje\_flash() {*

*if(isset($\_SESSION['mensaje'])) {*

*$this->mensaje = $\_SESSION['mensaje'];*

*unset($\_SESSION['mensaje']);*

*} else {*

*$this->mensaje;*

*}*

*}*

## Funciones utilitarias

Estas funciones se ubican en el archivo *“funciones.php”* en el directorio *“incluye”*.

## Función para contar registros

Esta función *“function contar\_id()“* cuenta y retorna el valor de la variable estática *“$contador“*. En la parte de abajo se muestra el código correspondiente.

*function contar\_id() {*

*static $count = 1;*

*return $count++;*

*}*

## Función para contar varios registros

Esta función *“function contar\_ids()“* cuenta y retorna el valor de la variable estática *“$count“*. En la parte de abajo se muestra el código correspondiente.

*function contar\_ids() {*

*static $count = 1;*

*return $count++;*

*}*

## Función para recuperar la fecha actual

Esta función *“function crear\_fecha()“* retorna la fecha actual de acuerdo con la zona horaria de México. Primero establece la zona horaria de Monterrey, México, por medio de la función de PHP *“date\_default\_timezone\_set“* y retorna la fecha actual por medio de las funciones de PHP *“strftime“* y *“time“*. En la parte de abajo se muestra el código correspondiente.

*function crear\_fecha() {*

*date\_default\_timezone\_set('America/Monterrey');*

*return strftime("%Y-%m-%d %H:%M:%S", time());*

*}*

## Función para leer la fecha

Esta función *“function leer\_fecha($str)“* recibe como parámetro una fecha en formato de cadena *“$str“*. Si *“$str“* no está vacío, procede a darle formato adecuado por medio de las funciones de PHP *“date “*y *“strtotime“* y lo retorna; caso contrario retorna un valor nulo. En la parte de abajo se muestra el código correspondiente.

*function leer\_fecha($str) {*

*if ($str) {*

*return date('d/m/Y g:i:s a', strtotime($str));*

*} else {*

*return null;*

*}*

*}*

## Función para mostrar un mensaje con estilos

Esta función *“function mostrar\_mensaje($mensaje ='')“* devuelve el parámetro *“$mensaje“* con formato HTML y estilo CSS adecuado. Se checa si *“$mensaje“* no se encuentra vacío por medio de la función de PHP *“empty“*. En caso de que *“$mensaje“* no este vacío se procede a guardar en una variable de tipo arreglo *“$output“* el mensaje para después retornarlo. En caso de que *“$mensaje“* este vacío devuelve una cadena vacía. En la parte de abajo se muestra el código correspondiente.

*function mostrar\_mensaje($mensaje =''){*

*$output = array();*

*if(!empty($mensaje)) {*

*foreach ($mensaje as $key => $value) {*

*$output = "<div class=\"alert alert-{$key}\">";*

*$output .= "<a href=\"#\" class=\"close\" data-dismiss=\"alert\">&times;</a>";*

*$output .= remover\_basura(primer\_caracter($value));*

*$output .= "</div>";*

*}*

*return $output;*

*} else {*

*return "";*

*}*

*}*

## Función para calcular el precio total

Esta función *“function precio\_total($totals)“* recibe como parámetro un arreglo *“$totals“* con los valores de precio total de venta y compra. Estos valores se sumarán y se asignarán a las variables *“$sum”*, *“$sub“* correspondientemente; además se calcula los beneficios, que es el precio de venta menos el precio de compra y se guarda su valor en la variable *“$profit“*. Finalmente retorna *“$sum“* y $*“profit“* en forma de un arreglo. En la parte de abajo se muestra el código correspondiente.

*function precio\_total($totals) {*

*$sum = 0;*

*$sub = 0;*

*foreach($totals as $total ){*

*$sum += $total['total\_saleing\_price'];*

*$sub += $total['total\_buying\_price'];*

*$profit = $sum - $sub;*

*}*

*return array($sum,$profit);*

*}*

## Función para reemplazar el carácter “-“ con un espacio

Esta función *“function primer\_caracter($str)“* reemplaza el carácter “-” con un espacio en blanco por medio de la función de PHP *“str\_replace“*, además de sustituir la primera letra de la cadena por una letra mayúscula de la cadena que se le pasa por parámetro *“$str“* por medio de la función de PHP *“ucfirst“* y tras todos estos filtros se almacena en la variable *“$val“* que la devuelve ya corregida. En la parte de abajo se muestra el código correspondiente.

*function primer\_caracter($str) {*

*$val = str\_replace('-'," ",$str);*

*$val = ucfirst($val);*

*return $val;*

*}*

## Función para generar una cadena con caracteres aleatorios

Esta función *“function randString($length = 5)“* devuelve una cadena con los siguientes caracteres aleatorios: “0123456789abcdefghijklmnopqrstuvwxyz”. Recibe como parámetro la longitud de la cadena que va a retornar *“$length“* que por defecto es de 5. Primero se crea una variable cadena con valor vacío *“$str“*. Seguido de otra variable cadena *“$cha“* con los caracteres que serán utilizados para esta función. Se actualiza el valor de *“$str“* con caracteres generados aleatoriamente por medio de las funciones de PHP *“mt\_rand“* y *“strlen“* tomando como referencia *“$cha“*. Finalmente se retorna *“$str“*. En la parte de abajo se muestra el código correspondiente.

*function randString($length = 5) {*

*$str='';*

*$cha = "0123456789abcdefghijklmnopqrstuvwxyz";*

*for ($x = 0; $x < $length; $x++) {*

*$str .= $cha[mt\_rand(0, strlen($cha))];*

*}*

*return $str;*

*}*

## Función para eliminar caracteres no convencionales de una cadena

Esta función *function real\_escape($str)“* elimina los caracteres no convencionales de la cadena que se le pasa por parámetro y la devuelve ya filtrada. Tiene como variable global *“$con“*. Se crea una variable *“$escape“* que guardará como valor el contenido de la cadena pasada como parámetro *“$str“* ya filtrada por medio de la función de PHP *“mysqli\_real\_escape\_string“* pasándole como parámetros a esta función la variable *“$con“* y el parámetro *“$str“*. Finalmente retorna el valor de la variable *“$escape“*. En la parte de abajo se muestra el código correspondiente.

*function real\_escape($str){*

*global $con;*

*$escape = mysqli\_real\_escape\_string($con,$str);*

*return $escape;*

*}*

## Función para redirigir a otra página

Esta función *“function redirigir($url, $permanent = false)“* direcciona a la url que le coloquemos por parámetro. Los parámetros son la dirección web *“$url“* y el tipo de direccionamiento *“$permanent“*. Primero se revisa que la pagina no se haya direccionado ya, por medio de la función de PHP *“headers\_sent“*. Si no es el caso, utiliza la función de PHP *“header“* utilizando los parámetros que recibe la presente función. Finalmente sale de la presente función con la función de PHP *“exit“*. En la parte de abajo se muestra el código correspondiente.

*function redirigir($url, $permanent = false) {*

*if (headers\_sent() === false) {*

*header('Location: ' . $url, true, ($permanent === true) ? 301 : 302);*

*}*

*exit();*

*}*

## Función para eliminar caracteres no convencionales de una consulta

Esta función *function remover\_basura($str)“* elimina los caracteres HTML que son extraños. Recibe como parámetro la variable *“$str“*. Se sobrescribe el parámetro *“$str“* con su mismo valor, pero eliminando los saltos de línea de HTML por medio de la función de PHP *“nl2br“*. Después se sobrescribe el parámetro *“$str“* con su mismo valor, pero eliminando los caracteres especiales por medio de la función de PHP *“htmlspecialchars“*. Finalmente se retorna el valor del parámetro *“$str“*. En la parte de abajo se muestra el código correspondiente.

*function remover\_basura($str) {*

*$str = nl2br($str);*

*$str = htmlspecialchars(strip\_tags($str, ENT\_QUOTES));*

*return $str;*

*}*

## Función para validar campos

Esta función *“function validar\_campos($var)“* recibe como parámetro un arreglo *“$var“* con los valores de los campos que se desea que se validen. Por cada valor del arreglo *“$var“* se asignará su valor de acuerdo con su índice a la variable *“$val“*. Se utilizará la función *“remover\_basura“* en la variable *“$val“* y en caso de que este vacía, asignará un error con el siguiente mensaje “. No puede estar en blanco.” por medio de la variable global *“$errores“* que será retornada con todos los mensajes de error. En la parte de abajo se muestra el código correspondiente.

*function validar\_campos($var) {*

*global $errores;*

*foreach ($var as $field) {*

*$val = remover\_basura($\_POST[$field]);*

*if(isset($val) && $val=='') {*

*$errores = $field .". No puede estar en blanco.";*

*return $errores;*

*}*

*}*

*}*

## Funciones SQL

Estas funciones se ubican en el archivo *“sql.php”* en el directorio *“incluye”*.

## Función para actualizar último login

Esta función *“function actualizarUltimoLogin($id\_usuario)“* actualiza el último acceso al sistema del correspondiente usuario por medio de su clave primaria *“$id\_usuario“* que recibe como parámetro.Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se crea una variable *“$date“* que tomará como valor el resultado que devuelva la función *“crear\_fecha“*. Se crea una variable *“$sql“* con la sentencia SQL; esta actualiza la fecha del último login de acuerdo con *“$id\_usuario“*. Se crea una variable *“$resultado“* que tomará como valor el resultado que devuelva el método *“query“* de la variable *“$db“*; este método recibe como parámetro *“$sql“*. Finalmente, la presente función, retorna el valor booleano verdadero en caso de que *“$resultado“* tenga algún valor y el método *“affected\_rows“* de la variable *“$db“* también tenga algún valor; caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*function actualizarUltimoLogin($id\_usuario) {*

*global $db;*

*$date = crear\_fecha();*

*$sql = "UPDATE usuario SET ultimo\_login='{$date}' WHERE id ='{$id\_usuario}' LIMIT 1";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para actualizar la cantidad de producto

Esta función *“function actualizar\_cantidad\_producto($qty,$p\_id)“* actualiza la cantidad del correspondiente producto por medio de su clave primaria *“$p\_id“* y la nueva cantidad de producto *“$qty “*que recibe como parámetro. Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se convierten a tipo entero *“$p\_id“* y *“$qty“*. Se crea una variable *“$sql“* con la sentencia SQL; esta actualiza la cantidad de producto por medio de *“$qty“* de acuerdo con *“$p\_id“*. Se crea una variable *“$resultado“* que tomará como valor el resultado que devuelva el método *“query“* de la variable *“$db“*; este método recibe como parámetro *“$sql“*. Finalmente, la presente función, retorna el valor booleano verdadero en caso de que *“$resultado“* tenga algún valor y el método *“affected\_rows“* de la variable *“$db“* también tenga algún valor; caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*function actualizar\_cantidad\_producto($qty,$p\_id){*

*global $db;*

*$qty = (int) $qty;*

*$id = (int)$p\_id;*

*$sql = "UPDATE producto SET cantidad=cantidad -'{$qty}' WHERE id = '{$id}'";*

*$resultado = $db->query($sql);*

*return($db->affected\_rows() === 1 ? true : false);*

*}*

## Función para actualizar la llave primaria de la categoría de productos

Esta función *“function actualizar\_id\_categoria\_producto($id)“* actualiza la clave referencial de la tabla *“categoría”* del correspondiente producto eliminado por medio de su clave primaria que recibe como parámetro esta función. Utiliza la misma estructura que la función *“actualizar\_cantidad\_producto“*. En la parte de abajo se muestra el código correspondiente.

*function actualizar\_id\_categoria\_producto($id){*

*global $db;*

*$id = (int)$id;*

*$sql = "UPDATE producto\_eliminado SET id\_categoria = 10 WHERE id = '{$id}'";*

*$resultado = $db->query($sql);*

*return($db->affected\_rows() === 1 ? true : false);*

*}*

## Función para actualizar la llave primaria de la imagen de productos

Esta función *“function actualizar\_id\_media\_producto($id)“* actualiza la clave referencial de la tabla media del correspondiente producto eliminado por medio de su clave primaria que recibe como parámetro esta función. Utiliza la misma estructura que la función *“actualizar\_cantidad\_producto“*. En la parte de abajo se muestra el código correspondiente.

*function actualizar\_id\_media\_producto($id){*

*global $db;*

*$id = (int)$id;*

*$sql = "UPDATE producto\_eliminado SET id\_media = 3 WHERE id = '{$id}'";*

*$resultado = $db->query($sql);*

*return($db->affected\_rows() === 1 ? true : false);*

*}*

## Función para autenticar el inicio de sesión de un usuario

Esta función *“function autenticar($usuario='', $contrasena='')“* da acceso al sistema de inventario o no. Recibe como parámetro el nombre del usuario “*“usuario”“* y la contraseña “*“contrasena”“*. Se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se convierten a tipo entero *“$usuario“* y *“$contrasena“*. Se crea una variable *“$sql“* con la sentencia SQL filtrada por medio de la función *“sprintf“* de PHP; esta selecciona el identificador del usuario, el nombre del usuario, la contraseña, nivel de usuario, estado de la cuenta de usuario, estado del grupo de usuario por medio de *“$usuario“*. Se crea una variable *“$resultado“* que tomará como valor el resultado que devuelva el método *“query“* de la variable *“$db“*; este método recibe como parámetro *“$sql“*. Si *“$resultado“* tiene algún valor, *“$usuario “*tomará el nombre del usuario de *“$resultado “*por medio de la función *“fetch\_assoc “*de *“$db“*. Se crea una variable *“$password\_request “*que tomará el valor de *“$contrasena “*por medio de la función *“sha1 “*de PHP; que desencripta *“$contrasena“*. Finalmente, la presente función, retorna el valor booleano verdadero en caso de que *“$resultado“* tenga algún valor y el método *“affected\_rows“* de la variable *“$db“* también tenga algún valor; caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*function autenticar($usuario='', $contrasena='') {*

*global $db;*

*$usuario = $db->escape($usuario);*

*$contrasena = $db->escape($contrasena);*

*$sql = sprintf("SELECT usuario.id,username,contrasena,nivel,estado,estado\_actual FROM usuario INNER JOIN grupo ON id\_grupo\_usuario = grupo.id WHERE username = '$usuario'");*

*$resultado = $db->query($sql);*

*if($db->num\_rows($resultado)) {*

*$usuario = $db->fetch\_assoc($resultado);*

*$password\_request = sha1($contrasena);*

*if($password\_request === $usuario['contrasena'] && $usuario['estado'] == 1 && $usuario['estado\_actual'] == 1) {*

*return $usuario['id'];*

*} elseif($password\_request !== $usuario['contrasena'] && $usuario['estado'] == 1 && $usuario['estado\_actual'] == 1) {*

*return -2;*

*} elseif($password\_request === $usuario['contrasena'] && $usuario['estado'] != 1 && $usuario['estado\_actual'] == 1) {*

*return -3;*

*} elseif($password\_request === $usuario['contrasena'] && $usuario['estado'] == 1 && $usuario['estado\_actual'] != 1) {*

*return -4;*

*}*

*}*

*return -1;*

*}*

## Función para borrar el rango de fecha de un reporte

Esta función *“function borrarRangoFecha($id\_tipos\_reporte)“* borra el registro de rango de fecha que coincida con la clave foránea de tipo de reporte recibido por parámetro a esta función. Utiliza la misma estructura que la función *“actualizar\_cantidad\_producto“*. En la parte de abajo se muestra el código correspondiente.

*function borrarRangoFecha($id\_tipos\_reporte) {*

*global $db;*

*$sql = "DELETE FROM rango\_fecha WHERE id\_tipos\_reporte = $id\_tipos\_reporte";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para borrar un tipo de reporte

Esta función *“function borrarTipoReporte($id\_usuario, $id\_reporte)“* borra el registro del reporte que coincida con las claves foráneas de usuario y reporte recibidos por parámetros a esta función. Utiliza la misma estructura que la función *“actualizar\_cantidad\_producto“*. En la parte de abajo se muestra el código correspondiente.

*function borrarTipoReporte($id\_usuario, $id\_reporte) {*

*global $db;*

*$sql = "DELETE FROM reporte\_usuario WHERE id\_usuario = $id\_usuario AND id\_reporte = $id\_reporte";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para borrar un registro por medio de su clave primaria

Esta función *“function borrar\_por\_id($table,$id)“* borra el determinado registro de una tabla. El nombre de la tabla *“$table“* es pasado por parámetro, así como la clave primaria de la tabla *“$id“*. Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se checa si *“$table “*existe por medio de la función *“tablaExiste“*. Se crea una variable *“$sql“* con la sentencia SQL; esta sentencia borra el registro de la tabla *“$table “*(filtrado su valor por medio del método de *“$db“*: *“escape“*)*” “*de acuerdo con la clave primaria *“$id “*(filtrado su valor por medio del método de *“db“*: *“escape“*). Se ejecuta la consulta *“$sql“* por medio del método de *“$db“*: *“query“*. Finalmente, la presente función, retorna el valor booleano verdadero en caso el método *“affected\_rows“* de la variable *“$db“* también tenga algún valor; caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*function borrar\_por\_id($table,$id) {*

*global $db;*

*if(tablaExiste($table)) {*

*$sql = "DELETE FROM ".$db->escape($table);*

*$sql .= " WHERE id=". $db->escape($id);*

*$sql .= " LIMIT 1";*

*$db->query($sql);*

*return ($db->affected\_rows() === 1) ? true : false;*

*}*

*}*

## Función para buscar un rango de fecha

Esta función *“buscarRangoFechas ($id\_tipos\_reporte)“* busca todos los registros de la tabla rango fecha en donde coincida la clave foránea pasada por parámetro *“$id\_tipos\_reporte“*. Se crea una variable *“$resultado “*que tendrá como valor la función *“encontrar\_por\_sql“* y posteriormente retornar esta variable. En la parte de abajo se muestra el código correspondiente.

*function buscarRangoFechas ($id\_tipos\_reporte) {*

*$sql = "SELECT \* FROM rango\_fecha WHERE id\_tipos\_reporte = $id\_tipos\_reporte";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para buscar un tipo de reporte

Esta función *“function buscarTipoReporte($id\_usuario, $id\_reporte)“* busca todos los registros de la tabla reporte usuario en donde coincidan las claves foráneas de usuario *“$id\_usuario“* y reporte *“$id\_reporte“*. Primero se crea una variable global *“$db“* que operará la base de datos por medio de sus métodos. Se crea una variable *“$sql“* con la sentencia SQL; esta sentencia selecciona todos los registros de la tabla reporte\_usuario*” “*de acuerdo con *“$id\_usuario “*e *“$id\_reporte“*. Se crea una variable *“$resultado “*que utilizará el método *“query“* de *“$db“* tomando como valor *“$sql“*. Finalmente, la presente función, retorna el valor booleano verdadero en caso el método *“num\_rows“* de la variable *“$db“* también tenga algún valor; caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*function buscarTipoReporte($id\_usuario, $id\_reporte) {*

*global $db;*

*$sql = "SELECT \* FROM reporte\_usuario WHERE id\_usuario = $id\_usuario AND id\_reporte = $id\_reporte";*

*$resultado = $db->query($sql);*

*return($db->num\_rows($resultado) === 1 ? true : false);*

*}*

## Función para buscar tipos de reportes

Esta función *“function buscarTipoReportes($id\_usuario, $id\_reporte)“* busca los tipos de reportes de acuerdo a la clave primaria de usuario y reporte, solo cambia al utiliza la función *“encontrar\_por\_sql“* en vez del método *“query“*. En la parte de abajo se muestra el código correspondiente.

*function buscarTipoReportes($id\_usuario, $id\_reporte) {*

*$sql = "SELECT \* FROM reporte\_usuario WHERE id\_usuario = $id\_usuario AND id\_reporte = $id\_reporte";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para contar registros por medio de su clave primaria

Esta función *“function contar\_por\_id($table)“* cuenta el número de registros que tiene una determinada tabla *“$table“* que recibe por parámetro. Primero se crea una variable global *“$db“* que operará la base de datos por medio de sus métodos. Se utiliza la función *“tablaExiste“* para poder checar si *“$table“* tiene algún valor correcto. Se crea una variable *“$sql“* con la sentencia SQL; esta sentencia selecciona y cuenta todas las claves primarias de la tabla pasada por parámetro y filtrada por medio del método *“escape “*de *“$db“*. Se ejecuta la consulta *“$sql“* por medio del método de *“$db“*: *“query“* y se guarda en una variable *“$resultado“*. Finalmente, la presente función, retorna el array asociativo correspondiente a *“$resultado“* por medio del método *“fetch\_assoc “* de *“$db“*. En la parte de abajo se muestra el código correspondiente.

*function contar\_por\_id($table) {*

*global $db;*

*if(tablaExiste($table)) {*

*$sql = "SELECT COUNT(id) AS total FROM ".$db->escape($table);*

*$resultado = $db->query($sql);*

*return($db->fetch\_assoc($resultado));*

*}*

*}*

## Función para saber los valores de la empresa

Esta función *function empresa()* retorna los valores de la única empresa alojada en la base de datos. Se crea una variable *“$empresas“* que tendrá como valor los datos de la empresa por medio de la función *“encontrar\_por\_id“*. Finalmente se retorna el valor de *“$empresas“*. En la parte de abajo se muestra el código correspondiente.

*function empresa() {*

*$empresas = encontrar\_por\_id('empresa', 1);*

*return $empresas;*

*}*

## Función para encontrar todos los registros de la tabla grupo

Esta función *“function encontrar\_id\_grupo\_usuario($nivel\_grupo)“* selecciona todos los registros de la tabla grupo, de acuerdo con su nivel de grupo *“$nivel\_grupo“*.Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se crea una variable *“$sql“* que tendrá como valor lo que devuelva el método *“query “*de *“$db “*donde este utilizará una sentencia SQL; esta sentencia selecciona todos los registros de la tabla grupo de acuerdo con el nivel de grupo. Se procede a checar si *“$sql“*, tiene algún valor por medio del método *“fetch\_assoc“* de *“$db“* y se asigna a una nueva variable *“$resultado“*; se retorna la clave primaria perteneciente al array asociativo *“$resultado“*. Si no lo tiene se retorna el valor nulo. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_id\_grupo\_usuario($nivel\_grupo) {*

*global $db;*

*$sql = $db->query("SELECT \* FROM grupo where nivel = $nivel\_grupo");*

*if ($resultado = $db->fetch\_assoc($sql)) {*

*return $resultado['id'];*

*} else {*

*return null;*

*}*

*}*

## Función para encontrar todos los grupos con un determinado nivel

Esta función *“function encontrar\_nivel\_grupo($nivel)“* selecciona todos los registros de la tabla grupo de acuerdo con el nivel de grupo *“$nivel “*recibido como parámetro. Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se crea una variable *“$sql“* con la sentencia SQL; esta sentencia selecciona todos los registros de la tabla grupo de acuerdo con *“$nivel“*. Se ejecuta la consulta *“$sql“* por medio de la función *“encontrar\_por\_sql“* y se guarda en una variable *“$resultado“*. Finalmente se retorna *“$resultado“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_nivel\_grupo($nivel) {*

*global $db;*

*$sql = "SELECT \*";*

*$sql .=" FROM grupo ";*

*$sql .="WHERE nivel = $nivel";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar todos los registros de la tabla categoría que coincidan con un nombre

Esta función *“function encontrar\_nombre\_categoria($nombre)“* selecciona todos los registros de la tabla categoría de acuerdo con el nombre de categoría *“$nombre“* recibido como parámetro. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todas las categorías de acuerdo con determinado nombre y se llama a la función *“encontrar\_por\_sql”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_nombre\_categoria($nombre) {*

*global $db;*

*$sql = "SELECT \*";*

*$sql .=" FROM categoria ";*

*$sql .="WHERE nombre = '$nombre'";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar todos los registros de la tabla grupo que coincidan con un nombre

Esta función *“function encontrar\_nombre\_grupo($nombre)“* selecciona todos los registros de la tabla grupo de acuerdo con el nombre de grupo *“$nombre “*recibido como parámetro. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todas las grupos de acuerdo con determinado nombre y se llama a la función *“encontrar\_por\_sql”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_nombre\_grupo($nombre) {*

*global $db;*

*$sql = "SELECT \*";*

*$sql .=" FROM grupo ";*

*$sql .="WHERE nombre = '$nombre'";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar todos los registros de la tabla producto que coincidan con un nombre

Esta función *“function encontrar\_nombre\_producto($nombre)“* selecciona todos los registros de la tabla producto de acuerdo con el nombre de producto *“$nombre“* recibido como parámetro. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todos los productos de acuerdo con determinado nombre y se llama a la función *“encontrar\_por\_sql”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_nombre\_producto($nombre) {*

*global $db;*

*$sql = "SELECT \*";*

*$sql .=" FROM producto ";*

*$sql .="WHERE nombre = '$nombre'";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar todos los registros de la tabla usuario que coincidan con un nombre

Esta función *“function encontrar\_nombre\_usuario($nombre)“* selecciona todos los registros de la tabla usuario de acuerdo con el nombre de usuario *“$nombre“* recibido como parámetro. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todos los usuarios de acuerdo con determinado nombre y se llama a la función *“encontrar\_por\_sql”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_nombre\_usuario($nombre) {*

*global $db;*

*$sql = "SELECT \*";*

*$sql .=" FROM usuario ";*

*$sql .="WHERE username = '$nombre'";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar todos los registros de la tabla usuario que coincidan con un nombre

Esta función *“function encontrar\_nombre\_usuarios($usuario)“* selecciona todos los registros de la tabla usuario de acuerdo con el nombre de usuario *“$nombre“* recibido como parámetro. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para selecciona todos los nombres de usuario de acuerdo con determinado nombre y se llama a la función *“encontrar\_por\_sql”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_nombre\_usuarios($usuario) {*

*global $db;*

*$sql = $db->query("SELECT \* FROM usuario where username = '$usuario'");*

*if ($resultado = $db->fetch\_assoc($sql)) {*

*return $resultado['username'];*

*} else {*

*return null;*

*}*

*}*

## Función para encontrar todos los registros de una tabla de acuerdo con su llave primaria

Esta función *“function encontrar\_por\_id($table,$id)“* selecciona todos los registros de una tabla de acuerdo con su clave primaria.El nombre de la tabla *“$table“* y su clave primaria *“$id “*son pasados por parámetro. Utiliza la misma sintaxis que la función *“encontrar\_id\_grupo\_usuario“*. Retorna la variable *“$resultado“*. Se utiliza además la función *“tablaExiste “*para checar si *“$table“* existe. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_por\_id($table,$id) {*

*global $db;*

*$id = (int)$id;*

*if(tablaExiste($table)) {*

*$sql = $db->query("SELECT \* FROM {$db->escape($table)} WHERE id='{$db->escape($id)}' LIMIT 1");*

*if ($resultado = $db->fetch\_assoc($sql)) {*

*return $resultado;*

*} else {*

*return null;*

*}*

*}*

*}*

## Función para encontrar todos los registros de 2 tablas de acuerdo con una llave primaria

Esta función *function encontrar\_por\_id\_join($table,$tables,$id)“* selecciona todos los registros de 2 tablas, de acuerdo con la clave foránea *“$id“* pasada por parámetro. Primero se checa que existan con las tablas con la función *“tablaExiste“*. Retorna la variable *“$resultado“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_por\_id\_join($table,$tables,$id) {*

*global $db;*

*$id = (int)$id;*

*if(tablaExiste($table)) {*

*$sql = $db->query("SELECT \* FROM {$db->escape($table)} RIGHT JOIN {$db->escape($tables)} ON TEMA.ID = TEMA\_PERSONAL.ID WHERE ID\_USUARIO='{$db->escape($id)}' LIMIT 1");*

*if ($resultado = $db->fetch\_assoc($sql)) {*

*return $resultado;*

*} else {*

*return null;*

*}*

*}*

*}*

## Función para encontrar determinados registros de la tabla usuario de acuerdo con una llave primaria

Esta función *“function encontrar\_por\_id\_usuario($id)“* selecciona los siguientes registros: Id de usuario, id de grupo, nombre de usuario, nombre de usuario, contraseña del usuario, imagen de perfil del usuario, imagen de fondo que tiene el usuario, estado de la cuenta del usuario, último acceso del usuario, nombre del grupo, nivel del grupo, estado del grupo de 2 tablas: usuario y grupo. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para encontrar los datos de un usuario por medio de su identificador y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. Retorna la variable *“$resultado“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_por\_id\_usuario($id) {*

*global $db;*

*$id = (int)$id;*

*$sql = $db->query("SELECT usuario.id, id\_grupo\_usuario, usuario.nombre as nombre\_usuario, username, contrasena,imagen, imagen\_fondo, estado, ultimo\_login, grupo.nombre, nivel, estado\_actual FROM usuario LEFT JOIN grupo ON id\_grupo\_usuario = grupo.id WHERE usuario.id='{$db->escape($id)}' LIMIT 1");*

*if ($resultado = $db->fetch\_assoc($sql)) {*

*return $resultado;*

*} else {*

*return null;*

*}*

*}*

## Función para encontrar un grupo por su nivel

Esta función *“function encontrar\_por\_nivel\_grupo($nivel)“* selecciona campo nivel de la tabla grupo donde el nivel coincida con el valor *“$nivel“* que es pasado por parámetro.Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se crea una variable *“$sql“* con la sentencia SQL; esta selecciona el campo nivel de acuerdo con *“$nivel“*. Se crea una variable *“$resultado“* que tomará como valor el resultado que devuelva el método *“query“* de la variable *“$db“*; este método recibe como parámetro *“$sql“*. Finalmente, la presente función, retorna el valor booleano verdadero en caso de que *“$resultado“* tenga algún valor y el método *“num\_rows“* de la variable *“$db“* también tenga algún valor; caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_por\_nivel\_grupo($nivel) {*

*global $db;*

*$sql = "SELECT nivel FROM grupo WHERE nivel = '{$db->escape($nivel)}' LIMIT 1 ";*

*$resultado = $db->query($sql);*

*return($db->num\_rows($resultado) === 0 ? true : false);*

*}*

## Función para encontrar un grupo por su nombre

Esta función *“function encontrar\_por\_nombre\_grupo($val)“* busca el campo nombre de la tabla grupo donde el nombre coincida con el valor *“$nombre“* que es pasado por parámetro. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todos los nombres de grupo de acuerdo con determinado nombre y se llama a la función *“encontrar\_por\_sql”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_por\_nombre\_grupo($val) {*

*global $db;*

*$sql = "SELECT nombre FROM grupo WHERE nombre = '{$db->escape($val)}' LIMIT 1 ";*

*$resultado = $db->query($sql);*

*return($db->num\_rows($resultado) === 0 ? true : false);*

*}*

## Función para encontrar por SQL una determinada consulta

Esta función *“function encontrar\_por\_sql($sql)“* retorna una sentencia SQL en forma de un result set. Recibe como parámetro la sentencia SQL: *“$sql“*. Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se crea una variable llamada *“$resultado“* que guardará como valor el resultado que devuelva el método de *“$db“*: *“query“*, que a su vez recibe como parámetro *“$sql“*. Se crea una variable *“$result\_set“* que guardará como valor el resultado que devuelva el método de *“$db“*: *“while\_loop“*, que a su vez recibe como parámetro *“$resultado“*. Finalmente retorna *“$result­\_set“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_por\_sql($sql) {*

*global $db;*

*$resultado = $db->query($sql);*

*$result\_set = $db->while\_loop($resultado);*

*return $result\_set;*

*}*

## Función para encontrar un producto por título

Esta función *“function encontrar\_producto\_por\_titulo($product\_name)“* selecciona el nombre del producto que más coincida con el valor pasado por parámetro *“$product\_name“* a esta función. Utiliza la misma sintaxis que la función *“encontrar\_nombre\_­usuario“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_producto\_por\_titulo($product\_name){*

*global $db;*

*$p\_name = remover\_basura($db->escape($product\_name));*

*$sql = "SELECT nombre FROM producto WHERE nombre like '%$p\_name%' LIMIT 5";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar todos los productos

Esta función *“function encontrar\_productos($nombre)“* selecciona los siguientes registros: nombre de archivo, nombre de producto, nombre de categoría, cantidad de producto, precio de compra de producto, precio de venta de producto, fecha de actualización de producto de 3 tablas: producto, media y categoría de acuerdo con el nombre del producto *“$nombre“* pasado por parámetro. Utiliza la misma sintaxis que la función *“encontrar\_por\_id“*. Retorna la variable *“$resultado“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_productos($nombre) {*

*global $db;*

*$sql = $db->query("SELECT nombre\_archivo, producto.nombre, categoria.nombre as categoria, cantidad, precio\_compra, precio\_venta, fecha\_actualizacion FROM producto RIGHT JOIN media on id\_media = media.id RIGHT JOIN categoria on id\_categoria = categoria.id where producto.nombre = '$nombre'");*

*if ($resultado = $db->fetch\_assoc($sql)) {*

*return $resultado;*

*} else {*

*return null;*

*}*

*}*

## Función para encontrar productos altamente vendidos

Esta función *“function encontrar\_productos\_altamente\_vendidos($limite)“* selecciona los siguientes registros: nombre de producto, cantidad de productos, cantidad de productos vendidos de 2 tablas: producto\_respaldo y venta limitados por el valor de limite *“$limite“* pasado por parámetro a esta función. Se crea una variable global *“$db“* que operará la base de datos por medio de sus métodos. Se utiliza el método *“query “*de *“$db“* para ejecutar *“$sql“*. Finalmente se retorna el resultado del método *“query“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_productos\_altamente\_vendidos($limite){*

*global $db;*

*$sql = "SELECT p.nombre, COUNT(s.id\_producto) AS totalSold, SUM(s.cantidad) AS totalQty";*

*$sql .= " FROM venta s";*

*$sql .= " LEFT JOIN producto\_respaldo p ON p.id = s.id\_producto ";*

*$sql .= " GROUP BY s.id\_producto";*

*$sql .= " ORDER BY SUM(s.cantidad) DESC LIMIT ".$db->escape((int)$limite);*

*return $db->query($sql);*

*}*

## Función para encontrar productos con bajo inventario

Esta función *“function encontrar\_productos\_bajo\_stock($limite)“* selecciona el nombre y cantidad de la tabla de productos donde la cantidad sea menor o igual al valor *“$limite“* pasado por parámetro a esta función. Se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos y se utilizará el método *“$escape“* que limpiara *“$límite“* de cualquier carácter extraño. Utiliza la función *“encontrar\_por\_sql“* para obtener el resultado y posteriormente retornarlo. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_productos\_bajo\_stock($limite) {*

*global $db;*

*$sql = "SELECT nombre, cantidad";*

*$sql .= " FROM producto";*

*$sql .= " WHERE cantidad <= ".$db->escape((int)$limite);*

*return encontrar\_por\_sql($sql);*

*}*

## Función para encontrar productos recientemente añadidos

Esta función *“function encontrar\_productos\_recientemente\_anadidos($limite)“* selecciona los siguientes registros: id de producto, nombre de producto, precio de venta de producto, referencia de la tabla media en producto, nombre de categoría, nombre de imagen de 3 tablas: producto, categoría y media limitados por el valor *“$limite“* pasado por parámetro a esta función. Utiliza la misma sintaxis que la función *“encontrar\_productos\_con\_bajo\_stock“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_productos\_recientemente\_anadidos($limite){*

*global $db;*

*$sql = " SELECT p.id,p.nombre,p.precio\_venta,p.id\_media,c.nombre AS categorie,";*

*$sql .= "m.nombre\_archivo AS image FROM producto p";*

*$sql .= " LEFT JOIN categoria c ON c.id = p.id\_categoria";*

*$sql .= " LEFT JOIN media m ON m.id = p.id\_media";*

*$sql .= " ORDER BY p.id DESC LIMIT ".$db->escape((int)$limite);*

*return encontrar\_por\_sql($sql);*

*}*

## Función para encontrar tipos de reporte por usuario

Esta función *”function encontrar\_tipos\_reporte\_por\_usuario($id\_usuario)“* selecciona todos los registros de la tabla de reporte usuario donde el campo id\_usuario sea igual al valor *“$id\_usuario“* pasado por parámetro a esta función. Utiliza la misma sintaxis que la función *“encontrar\_productos\_con\_bajo\_stock“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_tipos\_reporte\_por\_usuario($id\_usuario) {*

*global $db;*

*$sql = "SELECT \*";*

*$sql .= " FROM reporte\_usuario";*

*$sql .= " WHERE id\_usuario = ".$db->escape((int)$id\_usuario);*

*return encontrar\_por\_sql($sql);*

*}*

## Función para encontrar todas las ventas

Esta función *“function encontrar\_todas\_ventas()“* selecciona los siguientes registros: id de venta, cantidad de la venta, precio de la venta, fecha de la venta, nombre del producto y clave foránea de producto en venta de 2 tablas: *“producto\_respaldo”* y *“venta”*. Utiliza la misma sintaxis que la función *“encontrar\_productos\_con\_bajo\_stock“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_todas\_ventas(){*

*global $db;*

*$sql = "SELECT s.id,s.cantidad,s.precio,s.fecha,p.nombre,s.id\_usuario";*

*$sql .= " FROM venta s";*

*$sql .= " LEFT JOIN producto\_respaldo p ON s.id\_producto = p.id";*

*$sql .= " ORDER BY s.fecha DESC";*

*return encontrar\_por\_sql($sql);*

*}*

## Función para encontrar todas las ventas agregadas recientemente

Esta función *“function encontrar\_todas\_ventas\_agregadas\_recientemente($limite)“* selecciona los siguientes registros: id de venta, cantidad de la venta, precio de la venta, fecha de la venta, nombre del producto y clave foránea de producto en venta de 2 tablas: producto\_respaldo y venta y limita la cantidad de resultados por medio del valor *“$limite“* pasado por parámetro. Utiliza la misma sintaxis que la función *“encontrar\_productos\_con\_bajo\_stock“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_todas\_ventas\_agregadas\_recientemente($limite){*

*global $db;*

*$sql = "SELECT s.id,s.cantidad,s.precio,s.fecha,p.nombre";*

*$sql .= " FROM venta s";*

*$sql .= " LEFT JOIN producto\_respaldo p ON s.id\_producto = p.id";*

*$sql .= " ORDER BY s.fecha DESC LIMIT ".$db->escape((int)$limite);*

*return encontrar\_por\_sql($sql);*

*}*

## Función para encontrar todos los registros de una tabla

Esta función *“function encontrar\_todos($table)“* retorna todas las columnas de la tabla que recibe por parámetro *“$table“*, previamente verificando que la tabla exista. Se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se procede a checar si la tabla *“$table“* realmente existe en la base de datos; para ello, se utiliza la función *“tablaExiste“*. Si hay una tabla en la base de datos con el nombre de *“$table“*, retorna el valor devuelto por la función *“encontrar\_por\_sql“* que recibe como parámetro *“$sql“* y el nombre de la tabla *“$table“*; previamente filtrada de caracteres extraños por la función *“escape “*de *“$db“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_todos($table) {*

*global $db;*

*if(tablaExiste($table)) {*

*return encontrar\_por\_sql("SELECT \* FROM ".$db->escape($table));*

*}*

*}*

## Función para encontrar todos los productos de acuerdo con un título

Esta función *“function encontrar\_todos\_producto\_por\_titulo($title)“* selecciona todos los registros del producto que tengan el valor del campo nombre igual al valor del parámetro *“$title“* pasado a esta función. Utiliza la misma sintaxis que la función *“encontrar\_productos\_con\_bajo\_stock“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_todos\_producto\_por\_titulo($title){*

*global $db;*

*$sql = "SELECT \* FROM producto ";*

*$sql .= " WHERE nombre ='{$title}'";*

*$sql .=" LIMIT 1";*

*return encontrar\_por\_sql($sql);*

*}*

## Función para encontrar todos los usuarios

Esta función *“function encontrar\_todos\_usuarios()“* selecciona los siguientes registros: Id de usuario, id de grupo, nombre de usuario, nombre de usuario, contraseña del usuario, imagen de perfil del usuario, imagen de fondo que tiene el usuario, estado de la cuenta del usuario, último acceso del usuario, nombre del grupo, nivel del grupo, estado del grupo de 2 tablas: usuario y grupo. Utiliza la misma sintaxis que la función *“encontrar\_productos\_con\_bajo\_stock“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_todos\_usuarios() {*

*global $db;*

*$results = array();*

*$sql = "SELECT usuario.id,usuario.nombre,username,estado,ultimo\_login,";*

*$sql .="nivel,grupo.nombre as nombre\_grupo ";*

*$sql .="FROM usuario ";*

*$sql .="LEFT JOIN grupo ";*

*$sql .="ON grupo.id = id\_grupo\_usuario ORDER BY nombre ASC";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar un usuario de acuerdo con el nombre

Esta función *“function encontrar\_usuario($nombre)“* selecciona todos los registros de la tabla usuario de acuerdo con el nombre de usuario *“$nombre“* recibido como parámetro. Utiliza la misma sintaxis que la función *“encontrar\_productos\_con\_bajo\_stock“*. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_usuario($nombre) {*

*global $db;*

*$sql = "SELECT \*";*

*$sql .=" FROM usuario ";*

*$sql .="WHERE nombre = '$nombre'";*

*$resultado = encontrar\_por\_sql($sql);*

*return $resultado;*

*}*

## Función para encontrar ventas por un rango de fechas

Esta función *“function encontrar\_ventas\_por\_fechas($start\_date,$end\_date)“* selecciona la fecha de venta, nombre de producto, precio de venta del producto, precio de compra del producto, id del producto en venta, cantidad de productos en la venta, cantidad de ventas, precio de venta total y precio de compra total de 2 tablas: producto\_respaldo y venta de acuerdo con la venta de inicio *“$start\_date“* y venta máxima *“$end\_date“* pasadas por parámetro a esta función. Primero se convierte a tipo fecha *“$start\_date“* y *“$end\_date“*. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todas las ventas de acuerdo con los parámetros recibidos y se llama a la función *“query”;* el resultado se retorna. En la parte de abajo se muestra el código correspondiente.

*function encontrar\_ventas\_por\_fechas($start\_date,$end\_date){*

*global $db;*

*$start\_date = date("Y-m-d", strtotime($start\_date));*

*$end\_date = date("Y-m-d", strtotime($end\_date));*

*$sql = "SELECT s.fecha, p.nombre,p.precio\_venta,p.precio\_compra,s.id\_usuario,";*

*$sql .= "COUNT(s.id\_producto) AS total\_records,";*

*$sql .= "SUM(s.cantidad) AS total\_sales,";*

*$sql .= "SUM(p.precio\_venta \* s.cantidad) AS total\_saleing\_price,";*

*$sql .= "SUM(p.precio\_compra \* s.cantidad) AS total\_buying\_price ";*

*$sql .= "FROM venta s ";*

*$sql .= "LEFT JOIN producto\_respaldo p ON s.id\_producto = p.id";*

*$sql .= " WHERE s.fecha BETWEEN '{$start\_date}' AND '{$end\_date}'";*

*$sql .= " GROUP BY DATE(s.fecha),p.nombre";*

*$sql .= " ORDER BY DATE(s.fecha) DESC";*

*return $db->query($sql);*

*}*

## Función para insertar un rango de fecha para un reporte

Esta función *“function insertarRangoFecha($id\_tipos\_reporte, $fecha\_inicio, $fecha\_fin)“* registra un nuevo registro de rango de fecha con los valores recibidos por parámetro a esta función. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todos los rangos de fecha de acuerdo con los parámetros recibidos y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function insertarRangoFecha($id\_tipos\_reporte, $fecha\_inicio, $fecha\_fin) {*

*global $db;*

*$sql = "INSERT INTO rango\_fecha (id\_tipos\_reporte,fecha\_inicio,fecha\_fin) VALUES ($id\_tipos\_reporte, '$fecha\_inicio', '$fecha\_fin')";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para insertar un tipo de reporte

Esta función *“function insertarTipoReporte($id\_usuario, $id\_reporte)“* registra un nuevo registro de reporte de usuario de acuerdo con los valores recibidos *“$id\_usuario“* e *“$id\_reporte“* por parámetro a esta función. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para registrar todos los tipos de reporte de acuerdo con los parámetros recibidos y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function insertarTipoReporte($id\_usuario, $id\_reporte) {*

*global $db;*

*$sql = "INSERT INTO reporte\_usuario (id\_usuario,id\_reporte) VALUES ($id\_usuario, $id\_reporte)";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función que requiere un nivel de usuario para visualizarla

Esta función *“function pagina\_requiere\_nivel($nivel\_requerido)“* se encarga de permitir el acceso o no a ciertos usuarios dependiendo del valor numérico (1-3) siendo el menor el de mayor jerarquía. Se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se procede a buscar el usuario actual y el grupo al que pertenece por medio de las funciones *“usuario\_actual“* y *“encontrar\_por\_nivel\_grupo“* respectivamente y guardar los resultados en las variables *“$usuario\_actual“* y *“$nivel\_login“*. Se checa si ha iniciado sesión en el sistema y se revisa si cuenta con los permisos necesarios para ver la página web; en caso de que no cuente con los permisos necesarios se redireccionará a la página principal por medio de la función *“redirigir “*y se muestra un mensaje de error por medio de la función *“redirigir“*. En la parte de abajo se muestra el código correspondiente.

*function pagina\_requiere\_nivel($nivel\_requerido) {*

*global $sesion;*

*$usuario\_actual = usuario\_actual();*

*$nivel\_login = encontrar\_por\_nivel\_grupo($usuario\_actual['nivel']);*

*if (!$sesion->estaLogeado(true)):*

*$sesion->mensaje('d','Por favor iniciar sesión...');*

*redirigir('index.php', false);*

*elseif($nivel\_login['estado\_grupo'] === '0'):*

*$sesion->mensaje('d','Este nivel de usuario esta inactivo!');*

*redirigir('casa.php',false);*

*elseif($usuario\_actual['nivel'] <= (int)$nivel\_requerido):*

*return true;*

*else:*

*$sesion->mensaje("d", "¡Lo siento! no tienes permiso para ver la página.");*

*redirigir('casa.php', false);*

*endif;*

*}*

## Función para restaurar un registro por su clave primaria

Esta función *“function restaurar\_por\_id($id, $categoria, $media)“* restaura un producto eliminado. Esta función recibe como parámetro la clave primaria del producto eliminado *“$id“*, la categoría del producto *“$categoría“* y la imagen del producto *“$media“*.Primero se busca que el nombre del producto eliminado no esté usado en la lista de los productos, si no está usado, lo restaura por medio de *“$db“* y retorna un valor numérico que dependiendo de su valor mostrará un mensaje u otro fuera de esta función; caso contrario no hace nada y retorna un valor numérico que se tomará como un error o fallo. En la parte de abajo se muestra el código correspondiente.

*function restaurar\_por\_id($id, $categoria, $media) {*

*global $db;*

*$sql = "SELECT \* FROM PRODUCTO\_ELIMINADO WHERE ID=$id";*

*$resultado = $db->query($sql);*

*if ($db->num\_rows($resultado)) {*

*$resultados = $resultado->fetch\_array(MYSQLI\_NUM);*

*$sqls = "SELECT \* FROM PRODUCTO WHERE NOMBRE='{$resultados[1]}'";*

*$results = $db->query($sqls);*

*if ($db->num\_rows($results)) {*

*return 3;*

*}*

*$sqlss = "DELETE FROM PRODUCTO\_ELIMINADO WHERE ID=$id";*

*$db->query($sqlss);*

*if($db->affected\_rows() === 1) {*

*$condicion = FALSE;*

*$a5 = $resultados[5];*

*if($a5 == NULL) {*

*$condicion = TRUE;*

*}*

*$a6 = $resultados[6];*

*if($a6 == NULL) {*

*$condicion = TRUE;*

*}*

*if(!$condicion) {*

*$sqlsss = "INSERT INTO PRODUCTO (NOMBRE,CANTIDAD,PRECIO\_COMPRA,PRECIO\_VENTA,ID\_CATEGORIA,ID\_MEDIA,FECHA\_ACTUALIZACION) VALUES ('$resultados[1]','$resultados[2]',$resultados[3],$resultados[4],$resultados[5],$resultados[6],'$resultados[7]')";*

*$db->query($sqlsss);*

*return ($db->affected\_rows() === 1) ? 1 : 3;*

*} else {*

*$sqlsss = "INSERT INTO PRODUCTO (NOMBRE,CANTIDAD,PRECIO\_COMPRA,PRECIO\_VENTA,ID\_CATEGORIA,ID\_MEDIA,FECHA\_ACTUALIZACION) VALUES ('$resultados[1]','$resultados[2]',$resultados[3],$resultados[4],$categoria,$media,'$resultados[7]')";*

*$db->query($sqlsss);*

*return ($db->affected\_rows() === 1) ? 1 : 3;*

*}*

*} else {*

*return 0;*

*}*

*} else {*

*return 0;*

*}*

*}*

## Función para saber si una tabla existe

Esta función *function tablaExiste($table)* revisa que una determinada tabla exista. Recibe como parámetro el nombre de la tabla *“table“*. Primero se crea una variable global *“$db “*que operará la base de datos por medio de sus métodos. Se procede a crear una variable *“$table­\_exit“* que guardará como valor lo que devuelva el método *“query“* de *“$db“*; este método se utilizará para contar las filas de *“$table“*. Si *“$table\_exit“* tiene algún valor se checa por medio del método *“num\_rows“* y si es mayor a cero, retorna el valor booleano verdadero y caso contrario retorna el valor booleano falso. En la parte de abajo se muestra el código correspondiente.

*function tablaExiste($table) {*

*global $db;*

*$table\_exit = $db->query('SHOW TABLES FROM '.DB\_NAME.' LIKE "'.$db->escape($table).'"');*

*if($table\_exit) {*

*if ($db->num\_rows($table\_exit) > 0) {*

*return true;*

*} else {*

*return false;*

*}*

*}*

*}*

## Función que retorna el tema del usuario

Esta función *“function tema()“* retorna el tema del usuario con sesión actual. En la parte de abajo se muestra el código correspondiente.

*function tema() {*

*static $tema\_actual;*

*global $db;*

*if(!$tema\_actual) {*

*if(isset($\_SESSION['user\_id'])):*

*$id\_usuario = intval($\_SESSION['user\_id']);*

*$tema\_actual = encontrar\_por\_id\_join('tema','tema\_personal',$id\_usuario);*

*endif;*

*}*

*return $tema\_actual;*

*}*

## Función que devuelve registros de la tabla producto uniendo con la tabla categoría y media

Esta función *“function unir\_tabla\_producto()“* selecciona los siguientes registros: id de producto, nombre de producto, cantidad de producto, precio de compra de producto, precio de venta de producto, clave referencial de la tabla media en producto, fecha de actualización de producto, nombre de categoría, nombre de la imagen de 3 tablas: producto, categoría y media. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para seleccionar todos los productos y se llama a la función *“encontrar\_por\_sql”;* el resultado se retorna. En la parte de abajo se muestra el código correspondiente.

*function unir\_tabla\_producto() {*

*global $db;*

*$sql =" SELECT p.id,p.nombre,p.cantidad,p.precio\_compra,p.precio\_venta,p.id\_media,p.fecha\_actualizacion,c.nombre";*

*$sql .=" AS categorie,m.nombre\_archivo AS imagen";*

*$sql .=" FROM producto p";*

*$sql .=" LEFT JOIN categoria c ON c.id = p.id\_categoria";*

*$sql .=" LEFT JOIN media m ON m.id = p.id\_media";*

*$sql .=" ORDER BY p.id ASC";*

*return encontrar\_por\_sql($sql);*

*}*

## Función para unir la tabla de producto

Esta función *“function unir\_tabla\_productos()“* selecciona los siguientes registros: id de producto, nombre de producto, cantidad de producto, precio de compra de producto, precio de venta de producto, clave referencial de la tabla media en producto, fecha de actualización de producto, nombre de categoría, nombre de la imagen de 3 tablas: producto, categoría y media. En la parte de abajo se muestra el código correspondiente.

*function unir\_tabla\_productos() {*

*global $db;*

*$sql =" SELECT p.id,p.nombre,p.cantidad,p.precio\_compra,p.precio\_venta,p.id\_media,p.fecha\_actualizacion,c.nombre";*

*$sql .=" AS categorie,m.nombre\_archivo AS imagen";*

*$sql .=" FROM producto\_eliminado p";*

*$sql .=" LEFT JOIN categoria c ON c.id = p.id\_categoria";*

*$sql .=" LEFT JOIN media m ON m.id = p.id\_media";*

*$sql .=" ORDER BY p.id ASC";*

*return encontrar\_por\_sql($sql);*

*}*

## Función para actualizar la imagen de perfil del usuario

Esta función *“function updateImagen($id\_usuario)“* actualiza la imagen de perfil del correspondiente usuario por medio de su clave primaria *“$id\_usuario“* que recibe como parámetro esta función. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para actualizar la imagen de perfil de acuerdo *“$id\_usuario”* y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function updateImagen($id\_usuario) {*

*global $db;*

*$sql = "UPDATE usuario SET imagen='no\_image.jpg' WHERE id ='{$id\_usuario}' LIMIT 1";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para actualizar la imagen de fondo

Esta función *“function updateImagenFondo($id\_usuario)“* actualiza la imagen de fondo del correspondiente usuario por medio de su clave primaria *“$id\_usuario“* que recibe como parámetro esta función. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para actualizar la imagen de fondo de acuerdo *“$id\_usuario”* y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function updateImagenFondo($id\_usuario) {*

*global $db;*

*$sql = "UPDATE usuario SET imagen\_fondo='' WHERE id ='{$id\_usuario}' LIMIT 1";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para actualizar el lema de la empresa

Esta función *“function updateLema($mensaje)“* actualiza el lema de la única empresa alojada en la base de datos por medio del mensaje “*“mensaje“*” que recibe como parámetro esta función. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para actualizar el lema de la empresa de acuerdo *“$mensaje”* y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function updateLema($mensaje) {*

*global $db;*

*$sql = "UPDATE empresa SET lema='{$mensaje}' WHERE id ='1' LIMIT 1";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para actualizar el logo de la empresa

Esta función *“function updateLogo()“* actualiza la referencia al directorio donde se aloja el logo de la correspondiente empresa. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para actualizarel logo y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function updateLogo() {*

*global $db;*

*$sql = "UPDATE empresa SET logo='' WHERE id =1 LIMIT 1";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función para actualizar el tema del usuario

Esta función *“function updateTema($idUsuario, $idTema)“* actualiza las claves referenciales de usuario *“$idUsuario“* y tema *“$idTema“* de la tabla tema personal. Se crea una variable global *“$db”*. Se crea una consulta en la variable *“$sql”* para actualizar el tema del usuario de acuerdo con *“$idUsuario” y “$idTema”* y se llama a la función *“query”;* el resultado se almacena en la variable *“$resultado”* y finalmente se retorna. En la parte de abajo se muestra el código correspondiente.

*function updateTema($idUsuario, $idTema) {*

*global $db;*

*$sql = "UPDATE tema\_personal SET id\_tema=$idTema WHERE id=$idUsuario LIMIT 1";*

*$resultado = $db->query($sql);*

*return ($resultado && $db->affected\_rows() === 1 ? true : false);*

*}*

## Función que retorna el usuario actual

Esta función *“function usuario\_actual()“* retorna el nombre del usuario actual. Se crea una variable estática *“usuario\_actual “*que tendrá el usuario actual. Se crea una variable global *“$db“* que operará la base de datos. Si *“$usuario\_actual“* no tiene algún valor, se checa si hay una sesión activa de PHP y si la hay, se crea una variable *“$id\_usuario“*, que tendrá como valor la clave primaria del usuario que actualmente inició sesión, además se crea una variable *“$usuario\_actual“* que tendrá como valor lo que retorne la función *“encontrar\_por\_id\_usuario “*utilizando *“$id\_usuario“*. Finalmente se retorna *“$usuario\_­actual“*. En la parte de abajo se muestra el código correspondiente.

*function usuario\_actual() {*

*static $usuario\_actual;*

*global $db;*

*if(!$usuario\_actual) {*

*if(isset($\_SESSION['user\_id'])):*

*$id\_usuario = intval($\_SESSION['user\_id']);*

*$usuario\_actual = encontrar\_por\_id\_usuario($id\_usuario);*

*endif;*

*}*

*return $usuario\_actual;*

*}*

## Función que retorna las ventas diarias

Esta función *“function ventasDiarias($year,$month,$day)“* selecciona la cantidad vendida, fecha de venta, nombre de producto y precio de venta total de 2 tablas: producto\_respaldo y venta de acuerdo con la fecha actual. Utiliza la misma sintaxis que la función*” unir\_tabla\_producto“*. En la parte de abajo se muestra el código correspondiente.

*function ventasDiarias($year,$month,$day){*

*global $db;*

*$sql = "SELECT s.cantidad,";*

*$sql .= " DATE\_FORMAT(s.fecha, '%Y-%m-%e') AS date,p.nombre,";*

*$sql .= "SUM(p.precio\_venta \* s.cantidad) AS total\_saleing\_price";*

*$sql .= " FROM venta s";*

*$sql .= " LEFT JOIN producto\_respaldo p ON s.id\_producto = p.id";*

*$sql .= " WHERE DATE\_FORMAT(s.fecha, '%Y-%m-%e' ) = '{$year}-{$month}-{$day}'";*

*$sql .= " GROUP BY DATE\_FORMAT( s.fecha, '%e' ),s.id\_producto";*

*return encontrar\_por\_sql($sql);*

*}*

## Función que retorna las ventas mensuales

Esta función *“function ventasMensuales($year,$month)“* selecciona la cantidad vendida, fecha de venta, nombre de producto y precio de venta total de 2 tablas: producto\_respaldo y venta de acuerdo con el mes actual. Utiliza la misma sintaxis que la función*” unir\_tabla\_producto“*. En la parte de abajo se muestra el código correspondiente.

*function ventasMensuales($year,$month){*

*global $db;*

*$sql = "SELECT s.cantidad,";*

*$sql .= " DATE\_FORMAT(s.fecha, '%Y-%m-%e') AS date,p.nombre,";*

*$sql .= "SUM(p.precio\_venta \* s.cantidad) AS total\_saleing\_price";*

*$sql .= " FROM venta s";*

*$sql .= " LEFT JOIN producto\_respaldo p ON s.id\_producto = p.id";*

*$sql .= " WHERE DATE\_FORMAT(s.fecha, '%Y-%m' ) = '{$year}-{$month}'";*

*$sql .= " GROUP BY DATE\_FORMAT( s.fecha, '%e' ),s.id\_producto";*

*return encontrar\_por\_sql($sql);*

*}*

## Funciones JQuery para agregar grupos

Estas funciones se ubican en el archivo *“agregar\_grupo.js”* en el directorio *“libs/js”*.

## Función para revisar formulario de categoría

Esta función *“$('#formulario').submit(function miForm(event)“* se encarga de mandar el formulario para agregar una categoría. Primero se crean 3 variables *“$nombre\_grupo“*, *“$nivel\_grupo“* y *“$estado“*. A cada variable se le asignan sus correspondientes valores de campos por medio de la función de JQuery *“val“*. Se utiliza la función *“post“* que recibirá las 3 variables y las enviará al archivo verificar\_agregar\_grupo.php; cuando esta función lo haya hecho, mostrará un mensaje de éxito o fallo con los datos que muestra el archivo verificar\_agregar\_grupo.php. Se utiliza la función success para saber cómo proceder con lo que regresó la función *“post“*. Se crea la variable “*$variable“* que tomará como valor el atributo de HTML con identificador “requisito” y se checará su valor: si es 0 nos direcciona a la página de grupos, si es -1 borra el valor del campo de nombre de grupo y le hace foco y si es -2 borra el valor del campo de nivel de grupo y le hace foco. Finalmente se utiliza el método *“preventDefault“* de *“e “*para evitar que esta función se dispare dos veces. En la parte de abajo se muestra el código correspondiente.

*$('#formulario').submit(function miForm(event) {*

*var $nombre\_grupo;*

*var $nivel\_grupo;*

*var $estado;*

*$nombre\_grupo = $("#nombre\_grupo").val();*

*$nivel\_grupo = $("#nivel\_grupo").val();*

*$estado = $("#estado").val();*

*$.post("verificar\_agregar\_grupo.php", {nombre\_grupo: $nombre\_grupo, nivel\_grupo: $nivel\_grupo, estado: $estado}, function(mensaje) {*

*$("#resultado").html(mensaje);*

*$("#valor").html(requisito);*

*}).success(function() {*

*if($("#valor #requisito").length) {*

*var $variable;*

*$variable = $("#requisito").val();*

*if($variable > -1){*

*location.href ="http://localhost/13100034/grupo.php";*

*} else if($variable == -1) {*

*$("#nombre\_grupo").val('').focus();*

*} else if($variable == -2) {*

*$("#nivel\_grupo").val('').focus();*

*}*

*}*

*});*

*event.preventDefault();*

*});*

## Funciones JQuery para agregar productos

Estas funciones se ubican en el archivo *“agregar\_producto.js”* en el directorio “*libs/js*”.

## Función que aplica márgenes a la ventana cuando se carga

Esta función *“$(document).ready(function ventanaLista()“* se encarga de aplicar un margen izquierdo a dos elementos al cargar la página. Se crea una variable *“$ancho“* que tendrá como valor el ancho de la ventana por medio de la función de JQuery *“width“*. Se checa si *“$ancho “*es menor a 991 pixeles, si lo es aplicara un margen izquierdo a los dos elementos mencionados. Caso contrario retira los márgenes izquierdos. En la parte de abajo se muestra el código correspondiente.

*$(document).ready(function ventanaLista() {*

*var $ancho = $(window).width();*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*}*

*$(window).resize(function ventanaCambio(){*

*//aquí el código que se ejecutara cuando se redimensione la ventana*

*var $ancho = $(window).width();*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*}*

*});*

*});*

*$('#formulario').submit(function formularioProducto(event) {*

*var $titulo;*

*var $categoria;*

*var $foto;*

*var $cantidad;*

*var $compra;*

*var $compra\_decimal;*

*var $precio;*

*var $precio\_decimal;*

*$titulo = $("#titulo").val();*

*$categoria = $("#categoria").val();*

*$foto = $("#foto").val();*

*$cantidad = $("#cantidad").val();*

*$compra = $("#compra").val();*

*$compra\_decimal = $("#compra\_decimal").val();*

*$precio = $("#precio").val();*

*$precio\_decimal = $("#precio\_decimal").val();*

*$.post("verificar\_agregar\_producto.php", {titulo: $titulo, categoria: $categoria, foto: $foto, cantidad: $cantidad, compra: $compra, compra\_decimal: $compra\_decimal, precio: $precio, precio\_decimal: $precio\_decimal}, function(mensaje) {*

*$("#resultado").html(mensaje);*

*$("#valor").html(requisito);*

*}).success(function() {*

*if($("#valor #requisito").length) {*

*var $variable;*

*$variable = $("#requisito").val();*

*if($variable > -1){*

*location.href ="http://localhost/13100034/producto.php";*

*} else if($variable == -1) {*

*$("#titulo").val('').focus();*

*}*

*}*

*});*

*event.preventDefault();*

*});*

## Función que aplica márgenes al redimensionar la ventana

Esta función *“$(window).resize(function ventanaCambio()“* se encarga de aplicar un margen izquierdo a dos elementos al redimensionar la ventana. Utiliza la misma sintaxis que la función *“ventanaLista“*. En la parte de abajo se muestra el código correspondiente.

*$(window).resize(function ventanaCambio(){*

*//aquí el código que se ejecutara cuando se redimensione la ventana*

*var $ancho = $(window).width();*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*}*

*});*

## Función que revisa el formulario de producto

Esta función *“$('#formulario').submit(function formularioProducto(event)“* se encarga de mandar el formulario para agregar un producto. Utiliza la misma sintaxis que la función *“miForm“*. En la parte de abajo se muestra el código correspondiente.

*$('#formulario').submit(function formularioProducto(event) {*

*var $titulo;*

*var $categoria;*

*var $foto;*

*var $cantidad;*

*var $compra;*

*var $compra\_decimal;*

*var $precio;*

*var $precio\_decimal;*

*$titulo = $("#titulo").val();*

*$categoria = $("#categoria").val();*

*$foto = $("#foto").val();*

*$cantidad = $("#cantidad").val();*

*$compra = $("#compra").val();*

*$compra\_decimal = $("#compra\_decimal").val();*

*$precio = $("#precio").val();*

*$precio\_decimal = $("#precio\_decimal").val();*

*$.post("verificar\_agregar\_producto.php", {titulo: $titulo, categoria: $categoria, foto: $foto, cantidad: $cantidad, compra: $compra, compra\_decimal: $compra\_decimal, precio: $precio, precio\_decimal: $precio\_decimal}, function(mensaje) {*

*$("#resultado").html(mensaje);*

*$("#valor").html(requisito);*

*}).success(function() {*

*if($("#valor #requisito").length) {*

*var $variable;*

*$variable = $("#requisito").val();*

*if($variable > -1){*

*location.href ="http://localhost/13100034/producto.php";*

*} else if($variable == -1) {*

*$("#titulo").val('').focus();*

*}*

*}*

*});*

## Funciones JQuery para agregar productos

Estas funciones se ubican en el archivo “*agregar\_usuario.js”* en el directorio *“libs/js”*.

## Función que manda el formulario de productos

Esta función *“$('#formulario').submit(function formularioUsuario(event)“* se encarga de mandar el formulario para agregar un usuario. Utiliza la misma sintaxis que la función *“miForm“*. En la parte de abajo se muestra el código correspondiente.

*$('#formulario').submit(function formularioUsuario(event) {*

*var $nombre;*

*var $usuario;*

*var $password;*

*var $rol;*

*$nombre = $("#nombre").val();*

*$usuario = $("#usuario").val();*

*$password = $("#password").val();*

*$rol = $("#rol").val();*

*$.post("verificar\_agregar\_usuario.php", {nombre: $nombre, usuario: $usuario, password: $password, rol: $rol}, function(mensaje) {*

*$("#resultado").html(mensaje);*

*$("#valor").html(requisito);*

*}).success(function() {*

*if($("#valor #requisito").length) {*

*var $variable;*

*$variable = $("#requisito").val();*

*if($variable > -1){*

*location.href ="http://localhost/13100034/usuarios.php";*

*} else if($variable == -1) {*

*$("#nombre").val('').focus();*

*}*

*}*

*});*

*event.preventDefault();*

*});*

## Funciones JQuery para autenticación al iniciar sesión

Estas funciones se ubican en el archivo *“autenticacion.js”* en el directorio *“libs/js”*.

## Función que autentica el usuario que inició sesión

Esta función *“$('#formulario').submit(function formularioAutenticacion(event)“* se encarga de mandar el formulario para autenticar un usuario. Utiliza la misma sintaxis que la función *“miForm“*. En la parte de abajo se muestra el código correspondiente.

*$('#formulario').submit(function formularioAutenticacion(event) {*

*var $usuario;*

*var $password;*

*$usuario = $("#usuario").val();*

*$password = $("#password").val();*

*$.post("autenticacion.php", {usuario: $usuario, password: $password}, function(mensaje) {*

*$("#resultado").html(mensaje);*

*$("#valor").html(requisito);*

*}).success(function() {*

*if($("#valor #requisito").length) {*

*var $variable;*

*$variable = $("#requisito").val();*

*if($variable > -1){*

*location.href ="http://localhost/13100034/casa.php";*

*} else if($variable == -1) {*

*$("#usuario").val('').focus();*

*} else if($variable == -2) {*

*$("#password").val('').focus();*

*}*

*}*

*});*

*event.preventDefault();*

*});*

## Funciones JQuery para editar productos

Estas funciones se ubican en el archivo *“editar\_producto.js”* en el directorio *“libs/js”*.

## Función que redimensiona la página al cargar la página

Esta función *“$(document).ready(function redimension()“* se encarga de cargar la función descrita a continuación al cargar la página y se encarga de aplicar un margen izquierdo a los elementos correspondientes. Utiliza la misma sintaxis que la función *“ventanaLista“*. En la parte de abajo se muestra el código correspondiente.

*$(document).ready(function redimension() {*

*var $ancho = $(window).width();*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*$(".venta-decimal").css("margin-top","0px");*

*$(".venta-compra").css("margin-top","0px");*

*$(".clase").css("margin-top","0px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*$(".venta-decimal").css("margin-top","25px");*

*$(".venta-compra").css("margin-top","25px");*

*$(".clase").css("margin-top","25px");*

*}*

*$(window).resize(function cambioRedimension() {*

*//aquí el código que se ejecutara cuando se redimensione la ventana*

*var $ancho = $(window).width();*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*$(".venta-decimal").css("margin-top","0px");*

*$(".venta-compra").css("margin-top","0px");*

*$(".clase").css("margin-top","0px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*$(".venta-decimal").css("margin-top","25px");*

*$(".venta-compra").css("margin-top","25px");*

*$(".clase").css("margin-top","25px");*

*}*

*});*

*});*

## Función que cambia la dimensión de la página al redimensionar

Esta función *“$(window).resize(function cambioRedimension()“* se encarga de aplicar un margen izquierdo a los elementos correspondientes al redimensionar la ventana. Utiliza la misma sintaxis que la función *“ventanaCambio“*. En la parte de abajo se muestra el código correspondiente.

*$(window).resize(function cambioRedimension() {*

*//aquí el código que se ejecutara cuando se redimensione la ventana*

*var $ancho = $(window).width();*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*$(".venta-decimal").css("margin-top","0px");*

*$(".venta-compra").css("margin-top","0px");*

*$(".clase").css("margin-top","0px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*$(".venta-decimal").css("margin-top","25px");*

*$(".venta-compra").css("margin-top","25px");*

*$(".clase").css("margin-top","25px");*

*}*

*});*

## Funciones JQuery utilitarias

Estas funciones se ubican en el archivo *“functions.js”* en el directorio *“libs/js”*.

## Función que muestra un mensaje en forma de lista desplegable

Esta función *“function suggetion()“* se encarga de mostrar un mensaje en forma de lista desplegable en un campo de texto en el área para agregar ventas. Primero se checa que el usuario haya dejado de teclear en el campo de texto para posteriormente obtener el valor de ese campo de texto y proceder a guardarlo en una variable llamada *“product\_name“*. Se procede a revisar si el valor de *“product\_name“* está vacío, si no es el caso se manda a un archivo PHP llamado *“Ajax “*que nos devolverá las coincidencias de ese valor en el área de productos de la base de datos, caso contrario nos mostrará un mensaje de que no hay coincidencias; las coincidencias son mostradas en forma de una lista desplegable en el cuadro de texto correspondiente, pudiendo hacer clic en algún elemento de esa lista y si se hace clic en algún elemento de esa lista, el valor de *“product\_name“* será colocado como valor en el cuadro de texto con la desaparición de la lista del cuadro de texto. Caso contrario nos indicará que no hay coincidencias con nuestra búsqueda, de igual forma habrá un mensaje desplegable pero solo desaparecerá al hacer clic. En la parte de abajo se muestra el código correspondiente.

*function suggetion() {*

*$('#sug\_input').keyup(function(e) {*

*var formData = {*

*'product\_name' : $('input[name=title]').val()*

*};*

*if(formData['product\_name'].length >= 1) {*

*$.ajax({*

*type : 'POST',*

*url : 'ajax.php',*

*data : formData,*

*dataType : 'json',*

*encode : true*

*})*

*.done(function(data) {*

*$('#result').html(data).fadeIn();*

*$('#result li').click(function() {*

*$('#sug\_input').val($(this).text());*

*$('#result').fadeOut(500);*

*});*

*$("#sug\_input").blur(function() {*

*$("#result").fadeOut(500);*

*});*

*});*

*} else {*

*$("#result").hide();*

*};*

*e.preventDefault();*

*});*

*}*

## Función que manda un formulario Ajax

Esta función *“$('#sug-form').submit(function FormularioAjax(e)“* manda los datos de un determinado formulario a un archivo PHP sin recargar la página. Recibe como parámetro *“e“* que sería el manipulador del evento de esta función. Primero se crea una variable *“formData “*que tendrá como valor el nombre de producto por medio de la función de JQuery *“val “*sobre el elemento HTML. Se utiliza la función de JQuery *“ajax “*que enviará la información *“formData“* al archivo ajax.php por medio de POST. Se utiliza la función de JQuery *“done “*para mostrar un mensaje HTML de éxito, ya que se procese la función de JQuery *“ajax“*; donde además se usará la función *“total“*. Se utiliza la función de JQuery *“fail“* para mostrar un mensaje HTML de fallo. Finalmente se utiliza el método *“preventDefault“* de *“e “*para evitar que esta función se dispare dos veces. En la parte de abajo se muestra el código correspondiente.

*$('#sug-form').submit(function FormularioAjax(e) {*

*var formData = {*

*'p\_name' : $('input[name=title]').val()*

*};*

*$.ajax({*

*type : 'POST',*

*url : 'ajax.php',*

*data : formData,*

*dataType : 'json',*

*encode : true*

*})*

*.done(function(data) {*

*$('#product\_info').html(data).show();*

*total();*

*$('.datePicker').datepicker('update', new Date());*

*}).fail(function() {*

*$('#product\_info').html(data).show();*

*});*

*e.preventDefault();*

*});*

## Función que calcula el precio de venta total

Esta función *“function total()“* se encarga de mostrar en un cuadro de texto el precio de venta total producto de la cantidad total del producto a vender y el precio de venta individual de cada producto como valores en cuadros de texto respectivamente.

Primero se checa que el usuario haya hecho cambios en alguno de los cuadros de texto que tienen los valores de cantidad de producto y precio de venta individual de cada producto. En caso de haber cambios se guarda en una variable llamada *“price“* el valor del cuadro de texto del precio individual del producto, en otra variable *“qty“* se guarda la cantidad de producto. Finalmente, la variable *“price“* se multiplica con la variable *“qty“* y se guarda en la variable *“total“* para asignar su valor al cuadro de texto que corresponde al precio de venta total, redondeando a 2 decimales previamente. En la parte de abajo se muestra el código correspondiente.

*function total() {*

*$('#product\_info input').change(function(e) {*

*var price = +$('input[name=price]').val() || 0;*

*var qty = +$('input[name=quantity]').val() || 0;*

*var total = qty \* price ;*

*$('input[name=total]').val(total.toFixed(2));*

*});*

*}*

## Función que carga otras funciones al cargar la página

Esta función *“$(document).ready(function documentoListo()“* se ejecuta automáticamente al cargar la página. Se crea un tooltip a todos los elementos HTML que lo requieran. Se crea una función interna que hará aplicará tooltip con un retraso de 200 milisegundos. Se utilizan las funciones de JQuery *“suggetion “*y *“total“*. Finalmente se utiliza el método *“datapicker“* para poder seleccionar las fechas adecuadamente en los campos necesarios. En la parte de abajo se muestra el código correspondiente.

*$(document).ready(function documentoListo() {*

*$('[data-toggle="tooltip"]').tooltip();*

*$('.submenu-toggle').click(function () {*

*$(this).parent().children('ul.submenu').toggle(200);*

*});*

*suggetion();*

*total();*

*$('.datepicker')*

*.datepicker({*

*format: 'yyyy-mm-dd',*

*todayHighlight: true,*

*autoclose: true*

*});*

*});*

## Funciones JQuery para agregar múltiples categorías

Estas funciones se ubican en el archivo *“multiples\_registros\_categorias.js”* en el directorio *“libs/js”*.

## Función que carga otras funciones al cargar la página

Esta función *“$(document).ready(function documentoCategorias()“* se encarga de cargar el resto de las funciones descritas a continuación, al cargar la página. En la parte de abajo se muestra el código correspondiente.

*$(document).ready(function documentoCategorias() {*

*var $contador = 1;*

*var $categorias;*

*var $contador\_tabla = 0;*

*// Clona la fila oculta que tiene los campos base, y la agrega al final de la tabla*

*$("#adicional").on('click', function clickCategoria() {*

*$("#tabla tbody tr:eq(0)").clone().removeClass('fila-fija').appendTo("#tabla");*

*$("#tabla tbody tr:eq(" + $contador + ") input").removeAttr("name");*

*$("#tabla tbody tr:eq(" + $contador + ") input").attr("name","nombre-categoria-" + $contador);*

*$('.boton-menos').attr("disabled", false);*

*$('#nombre-categoria').val('').focus();*

*$contador++;*

*});*

*// Evento que selecciona la fila y la elimina*

*$(document).on("click",".eliminar",function eliminarCategoria() {*

*if($contador === 2) {*

*var parent = $(this).parents().get(0);*

*$(parent).remove();*

*$contador--;*

*$('.boton-menos').attr("disabled", true);*

*} else {*

*var parent = $(this).parents().get(0);*

*$(parent).remove();*

*$contador--;*

*}*

*for (var $i = 0; $i <= $contador; $i++) {*

*$("#tabla tbody tr:eq(" + $i + ") input").removeAttr("name");*

*$("#tabla tbody tr:eq(" + $i + ") input").attr("name","nombre-categoria-" + $i);*

*}*

*});*

*$("#agregarCategoria").click(function agregarCategoria() {*

*$categorias = new Array($contador);*

*var $indicador = false;*

*for(var $i = 0; $i < $contador; $i++) {*

*$categorias[$i] = $("#tabla tbody tr:eq(" + $i + ") input").val();*

*}*

*for(var $i = 0; $i < $contador; $i++) {*

*for(var $j = 1; $j < $contador; $j++) {*

*if($categorias[$i] === $categorias[$j] && $i !== $j) {*

*if(!$("#tabla tbody tr:eq(" + $j + ")").hasClass('has-error')) {*

*$("#tabla tbody tr:eq(" + $j + ") input").val('').focus();*

*$("#tabla tbody tr:eq(" + $j + ") td:eq(" + 0 + ")").append("<span class='" + $j + "'>Procura no ingresar iguales.</span>");*

*$("#tabla tbody tr:eq(" + $j + ")").addClass('has-error');*

*$indicador = true;*

*break;*

*} else {*

*$("#tabla tbody tr:eq(" + $j + ") input").val('').focus();*

*$indicador = true;*

*break;*

*}*

*} else {*

*if($("#tabla tbody tr:eq(" + $j + ")").hasClass('has-error')) {*

*$("#tabla tbody tr:eq(" + $j + ")").removeClass('has-error');*

*$("span").remove("." + $j);*

*}*

*}*

*}*

*if($indicador) {*

*break;*

*}*

*}*

*});*

*$('#formulario').submit(function enviarCategoria(event) {*

*$contador\_tabla = $("#tabla\_categorias tr").length;*

*if($("#tabla\_categorias\_recientes tr").length) {*

*var $contador\_tabla\_recientes = 0;*

*$contador\_tabla\_recientes = $("#tabla\_categorias\_recientes tr").length;*

*$contador\_tabla\_recientes--;*

*var $categorias\_agregadas = new Array($contador\_tabla\_recientes);*

*for(var $i = 0; $i < $contador\_tabla\_recientes; $i++) {*

*$categorias\_agregadas[$i] = $("#tabla\_categorias\_recientes tbody tr:eq(" + $i + ") td")[1].innerHTML;*

*}*

*}*

*$.post("buscar\_categoria.php", {categorias: $categorias, contador: $contador\_tabla, categorias\_recientes: $categorias\_agregadas, contador\_tabla\_recientes: $contador\_tabla\_recientes}, function(mensaje) {*

*$("#resultadoBusqueda").html(mensaje);*

*$("#resultados").html(datos\_categorias);*

*});*

*event.preventDefault();*

*});*

*});*

## Función para agregar múltiples categorías

Esta función *“$("#agregarCategoria").click(function agregarCategoria()“* guarda las categorías localizadas en los campos de categorias, en una variable arreglo *“$categorias “*y las revisa e informa al usuario de cualquier error. En la parte de abajo se muestra el código correspondiente.

*$("#agregarCategoria").click(function agregarCategoria() {*

*$categorias = new Array($contador);*

*var $indicador = false;*

*for(var $i = 0; $i < $contador; $i++) {*

*$categorias[$i] = $("#tabla tbody tr:eq(" + $i + ") input").val();*

*}*

*for(var $i = 0; $i < $contador; $i++) {*

*for(var $j = 1; $j < $contador; $j++) {*

*if($categorias[$i] === $categorias[$j] && $i !== $j) {*

*if(!$("#tabla tbody tr:eq(" + $j + ")").hasClass('has-error')) {*

*$("#tabla tbody tr:eq(" + $j + ") input").val('').focus();*

*$("#tabla tbody tr:eq(" + $j + ") td:eq(" + 0 + ")").append("<span class='" + $j + "'>Procura no ingresar iguales.</span>");*

*$("#tabla tbody tr:eq(" + $j + ")").addClass('has-error');*

*$indicador = true;*

*break;*

*} else {*

*$("#tabla tbody tr:eq(" + $j + ") input").val('').focus();*

*$indicador = true;*

*break;*

*}*

*} else {*

*if($("#tabla tbody tr:eq(" + $j + ")").hasClass('has-error')) {*

*$("#tabla tbody tr:eq(" + $j + ")").removeClass('has-error');*

*$("span").remove("." + $j);*

*}*

*}*

*}*

*if($indicador) {*

*break;*

*}*

*}*

*});*

## Función para agregar un formulario de categoría con un botón

Esta función *“$("#adicional").on('click', function clickCategoria()“* agrega un campo para agregar una categoría y va contado los que hay en total de forma dinámica. En la parte de abajo se muestra el código correspondiente.

*$("#adicional").on('click', function clickCategoria() {*

*$("#tabla tbody tr:eq(0)").clone().removeClass('fila-fija').appendTo("#tabla");*

*$("#tabla tbody tr:eq(" + $contador + ") input").removeAttr("name");*

*$("#tabla tbody tr:eq(" + $contador + ") input").attr("name","nombre-categoria-" + $contador);*

*$('.boton-menos').attr("disabled", false);*

*$('#nombre-categoria').val('').focus();*

*$contador++;*

*});*

## Función para eliminar un formulario de categoría con un botón

Esta función *“$(document).on("click",".eliminar",function eliminarCategoria()“* quita un campo para agregar una categoría y va contado los que hay en total de forma dinámica. Deshabilita de forma permanente el botón para eliminar el campo de categoría si solo hay un campo. En la parte de abajo se muestra el código correspondiente.

*$(document).on("click",".eliminar",function eliminarCategoria() {*

*if($contador === 2) {*

*var parent = $(this).parents().get(0);*

*$(parent).remove();*

*$contador--;*

*$('.boton-menos').attr("disabled", true);*

*} else {*

*var parent = $(this).parents().get(0);*

*$(parent).remove();*

*$contador--;*

*}*

*for (var $i = 0; $i <= $contador; $i++) {*

*$("#tabla tbody tr:eq(" + $i + ") input").removeAttr("name");*

*$("#tabla tbody tr:eq(" + $i + ") input").attr("name","nombre-categoria-" + $i);*

*}*

*});*

## Función para mandar el o los formularios de categorías

Esta función *“$('#formulario').submit(function enviarCategoria(event)“* cuenta los campos de categorias y guarda sus valores, para posteriormente enviarlos al archivo PHP correspondiente a través de Ajax. En la parte de abajo se muestra el código correspondiente.

*$('#formulario').submit(function enviarCategoria(event) {*

*$contador\_tabla = $("#tabla\_categorias tr").length;*

*if($("#tabla\_categorias\_recientes tr").length) {*

*var $contador\_tabla\_recientes = 0;*

*$contador\_tabla\_recientes = $("#tabla\_categorias\_recientes tr").length;*

*$contador\_tabla\_recientes--;*

*var $categorias\_agregadas = new Array($contador\_tabla\_recientes);*

*for(var $i = 0; $i < $contador\_tabla\_recientes; $i++) {*

*$categorias\_agregadas[$i] = $("#tabla\_categorias\_recientes tbody tr:eq(" + $i + ") td")[1].innerHTML;*

*}*

*}*

*$.post("buscar\_categoria.php", {categorias: $categorias, contador: $contador\_tabla, categorias\_recientes: $categorias\_agregadas, contador\_tabla\_recientes: $contador\_tabla\_recientes}, function(mensaje) {*

*$("#resultadoBusqueda").html(mensaje);*

*$("#resultados").html(datos\_categorias);*

*});*

*event.preventDefault();*

*});*

## Funciones JQuery para agregar múltiples productos.

Estas funciones se ubican en el archivo *“multiples\_registros\_productos.js”* en el directorio *“libs/js”*.

## Función que carga el resto de las funciones al cargar la página

Esta función *“$(document).ready(function documentoProductos()“* se encarga de cargar el resto de las funciones descritas a continuación, al cargar la página. En la parte de abajo se muestra el código correspondiente.

*$(document).ready(function documentoProductos() {*

*var $contador\_producto = 1;*

*var $ancho = $(window).width();*

*var $indice;*

*var $indices;*

*var $titulo\_productos;*

*var $categoria\_productos;*

*var $imagen\_productos;*

*var $cantidad\_productos;*

*var $compra\_productos;*

*var $compra\_decimal\_productos;*

*var $precio\_productos;*

*var $precio\_decimal\_productos;*

*var $nombre\_imagen\_productos;*

*var $nombre\_categoria\_productos;*

*var $contador\_tabla\_productos\_recientes;*

*var $productos\_agregados;*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*}*

*$(window).resize(function redimensionProductos(){*

*//aquí el código que se ejecutara cuando se redimensione la ventana*

*var $ancho = $(window).width();*

*if($ancho < 991) {*

*$(".venta-decimal").css("margin-left","15px");*

*$(".venta-compra").css("margin-left","15px");*

*} else {*

*$(".venta-decimal").css("margin-left","0");*

*$(".venta-compra").css("margin-left","0");*

*}*

*});*

*// Clona la fila oculta que tiene los campos base, y la agrega al final de la tabla*

*$("#adicional-producto").on('click', function agregarProducto() {*

*$indice = $("#formulario-producto .form-producto");*

*$("#formulario-producto .form-producto:eq(0)").clone().appendTo("#contenedor-producto").attr("id","contenedor-producto-" + $contador\_producto);*

*$("#formulario-producto input.titulo:eq(" + $contador\_producto + ")").attr("name","titulo-producto-" + $contador\_producto);*

*$("#formulario-producto select.categoria:eq(" + $contador\_producto + ")").attr("name","categoria-" + $contador\_producto);*

*$("#formulario-producto select.foto:eq(" + $contador\_producto + ")").attr("name","product-photo-" + $contador\_producto);*

*$("#formulario-producto input.cantidad:eq(" + $contador\_producto + ")").attr("name","cantidad-" + $contador\_producto);*

*$("#formulario-producto input.compra:eq(" + $contador\_producto + ")").attr("name","compra-" + $contador\_producto);*

*$("#formulario-producto input.compra\_decimal:eq(" + $contador\_producto + ")").attr("name","compra\_decimal-" + $contador\_producto);*

*$("#formulario-producto input.precio:eq(" + $contador\_producto + ")").attr("name","precio-" + $contador\_producto);*

*$("#formulario-producto input.precio\_decimal:eq(" + $contador\_producto + ")").attr("name","precio\_decimal-" + $contador\_producto);*

*$("#formulario-producto .boton-menos-producto:eq(" + $contador\_producto + ")").attr("id","" + $contador\_producto);*

*$('.boton-menos-producto').attr("disabled", false);*

*$("#formulario-producto input.titulo:eq(" + $contador\_producto + ")").val('').focus();*

*$("#formulario-producto select.categoria:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto select.foto:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.cantidad:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.compra:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.compra\_decimal:eq(" + $contador\_producto + ")").val('0');*

*$("#formulario-producto input.precio:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.precio\_decimal:eq(" + $contador\_producto + ")").val('0');*

*$contador\_producto++;*

*});*

*// Evento que selecciona la fila y la elimina*

*$(document).on("click",".boton-menos-producto",function eliminarProducto() {*

*$indices = $(this).attr("id");*

*if($contador\_producto === 2) {*

*$("#contenedor-producto-" + $indices).remove();*

*$contador\_producto--;*

*$('.boton-menos-producto').attr("disabled", true);*

*} else {*

*$("#contenedor-producto-" + $indices).remove();*

*$contador\_producto--;*

*}*

*for(var $i = 0; $i <= $contador\_producto; $i++) {*

*$("#formulario-producto div div.form-producto:eq(" + $i + ")").attr("id","contenedor-producto-" + $i);*

*$("#formulario-producto input.titulo:eq(" + $i + ")").attr("name","titulo-producto-" + $i);*

*$("#formulario-producto select.categoria:eq(" + $i + ")").attr("name","categoria-" + $i);*

*$("#formulario-producto select.foto:eq(" + $i + ")").attr("name","product-photo-" + $i);*

*$("#formulario-producto input.cantidad:eq(" + $i + ")").attr("name","cantidad-" + $i);*

*$("#formulario-producto input.compra:eq(" + $i + ")").attr("name","compra-" + $i);*

*$("#formulario-producto input.compra\_decimal:eq(" + $i + ")").attr("name","compra\_decimal-" + $i);*

*$("#formulario-producto input.precio:eq(" + $i + ")").attr("name","precio-" + $i);*

*$("#formulario-producto input.precio\_decimal:eq(" + $i + ")").attr("name","precio\_decimal-" + $i);*

*$("#formulario-producto .boton-menos-producto:eq(" + $i + ")").attr("id","" + $i);*

*}*

*});*

*$("#agregarProducto").click(function adicionarProducto() {*

*$titulo\_productos = new Array($contador\_producto);*

*var $indicador = false;*

*for(var $i = 0; $i < $contador\_producto; $i++) {*

*$titulo\_productos[$i] = $("#formulario-producto input.titulo:eq(" + $i + ")").val();*

*}*

*for(var $i = 0; $i < $contador\_producto; $i++) {*

*for(var $j = 1; $j < $contador\_producto; $j++) {*

*if($titulo\_productos[$i] === $titulo\_productos[$j] && $i !== $j) {*

*if(!$("#formulario-producto div.clase-titulo:eq(" + $j + ")").hasClass('has-error')) {*

*$("#formulario-producto input.titulo:eq(" + $j + ")").val('').focus();*

*$("#formulario-producto div.titulo-es:eq(" + $j + ")").append("<span class='span-" + $j + "'>Procura no ingresar iguales.</span>");*

*$("#formulario-producto div.clase-titulo:eq(" + $j + ")").addClass('has-error');*

*$indicador = true;*

*break;*

*} else {*

*$("#formulario-producto input.titulo:eq(" + $j + ")").val('').focus();*

*$indicador = true;*

*break;*

*}*

*} else {*

*if($("#formulario-producto div.clase-titulo:eq(" + $j + ")").hasClass('has-error')) {*

*$("#formulario-producto div.clase-titulo:eq(" + $j + ")").removeClass('has-error');*

*$("span").remove(".span-" + $j);*

*}*

*}*

*}*

*if($indicador) {*

*break;*

*}*

*}*

*});*

*$('#formulario-producto').submit(function enviarProductos(event) {*

*var $contador\_tabla = $(".origen tr").length;*

*$categoria\_productos = new Array($contador\_producto);*

*$imagen\_productos = new Array($contador\_producto);*

*$cantidad\_productos = new Array($contador\_producto);*

*$compra\_productos = new Array($contador\_producto);*

*$compra\_decimal\_productos = new Array($contador\_producto);*

*$precio\_productos = new Array($contador\_producto);*

*$precio\_decimal\_productos = new Array($contador\_producto);*

*$nombre\_imagen\_productos = new Array($contador\_producto);*

*$nombre\_categoria\_productos = new Array($contador\_producto);*

*for(var $i = 0; $i < $contador\_producto; $i++) {*

*$categoria\_productos[$i] = $("#formulario-producto select.categoria:eq(" + $i + ")").val();*

*$imagen\_productos[$i] = $("#formulario-producto select.foto:eq(" + $i + ")").val();*

*$cantidad\_productos[$i] = $("#formulario-producto input.cantidad:eq(" + $i + ")").val();*

*$compra\_productos[$i] = $("#formulario-producto input.compra:eq(" + $i + ")").val();*

*$compra\_decimal\_productos[$i] = $("#formulario-producto input.compra\_decimal:eq(" + $i + ")").val();*

*$precio\_productos[$i] = $("#formulario-producto input.precio:eq(" + $i + ")").val();*

*$precio\_decimal\_productos[$i] = $("#formulario-producto input.precio\_decimal:eq(" + $i + ")").val();*

*$nombre\_imagen\_productos[$i] = $("#formulario-producto select.foto:eq(" + $i + ") option:selected").text();*

*$nombre\_categoria\_productos[$i] = $("#formulario-producto select.categoria:eq(" + $i + ") option:selected").text();*

*}*

*if($(".tabla-resultado-producto tbody tr").length) {*

*$contador\_tabla\_productos\_recientes = $(".tabla-resultado-producto tbody tr").length;*

*$productos\_agregados = new Array($contador\_tabla\_productos\_recientes);*

*for(var $i = 0; $i < $contador\_tabla\_productos\_recientes; $i++) {*

*$productos\_agregados[$i] = $(".tabla-resultado-producto tbody tr td.necesario")[$i].innerHTML;*

*}*

*}*

*$.post("buscar\_producto.php", {productos\_agregados: $productos\_agregados, contador\_tabla\_productos\_recientes: $contador\_tabla\_productos\_recientes, nombre\_categoria\_productos: $nombre\_categoria\_productos, nombre\_imagen\_productos: $nombre\_imagen\_productos, titulo\_productos: $titulo\_productos, categoria\_productos: $categoria\_productos, imagen\_productos: $imagen\_productos, cantidad\_productos: $cantidad\_productos, compra\_productos: $compra\_productos, compra\_decimal\_productos: $compra\_decimal\_productos, precio\_productos: $precio\_productos, precio\_decimal\_productos: $precio\_decimal\_productos, contador\_producto: $contador\_producto, contador\_tabla: $contador\_tabla}, function(mensaje) {*

*$("#resultadoBusqueda-productos").html(mensaje);*

*$("#resultados-productos").html(datos\_productos);*

*});*

*event.preventDefault();*

*});*

*});*

## Función que guarda los datos de los productos antes de mandarlos en el formulario

Esta función *“$("#agregarProducto").click(function adicionarProducto()“* guarda los productos localizados en los campos de productos, en una variable arreglo *“$productos “*y las revisa e informa al usuario de cualquier error. Utiliza la misma sintaxis que la función *“agregarCategoria“*.

*$("#agregarProducto").click(function adicionarProducto() {*

*$titulo\_productos = new Array($contador\_producto);*

*var $indicador = false;*

*for(var $i = 0; $i < $contador\_producto; $i++) {*

*$titulo\_productos[$i] = $("#formulario-producto input.titulo:eq(" + $i + ")").val();*

*}*

*for(var $i = 0; $i < $contador\_producto; $i++) {*

*for(var $j = 1; $j < $contador\_producto; $j++) {*

*if($titulo\_productos[$i] === $titulo\_productos[$j] && $i !== $j) {*

*if(!$("#formulario-producto div.clase-titulo:eq(" + $j + ")").hasClass('has-error')) {*

*$("#formulario-producto input.titulo:eq(" + $j + ")").val('').focus();*

*$("#formulario-producto div.titulo-es:eq(" + $j + ")").append("<span class='span-" + $j + "'>Procura no ingresar iguales.</span>");*

*$("#formulario-producto div.clase-titulo:eq(" + $j + ")").addClass('has-error');*

*$indicador = true;*

*break;*

*} else {*

*$("#formulario-producto input.titulo:eq(" + $j + ")").val('').focus();*

*$indicador = true;*

*break;*

*}*

*} else {*

*if($("#formulario-producto div.clase-titulo:eq(" + $j + ")").hasClass('has-error')) {*

*$("#formulario-producto div.clase-titulo:eq(" + $j + ")").removeClass('has-error');*

*$("span").remove(".span-" + $j);*

*}*

*}*

*}*

*if($indicador) {*

*break;*

*}*

*}*

*});*

## Función para mandar el o los productos por medio de los formularios

Esta función *“$("#adicional-producto").on('click', function agregarProducto()“* agrega un campo para agregar un producto y va contado los que hay en total de forma dinámica. Utiliza la misma sintaxis que la función *“clickCategoria“*. En la parte de abajo se muestra el código correspondiente.

*$("#adicional-producto").on('click', function agregarProducto() {*

*$indice = $("#formulario-producto .form-producto");*

*$("#formulario-producto .form-producto:eq(0)").clone().appendTo("#contenedor-producto").attr("id","contenedor-producto-" + $contador\_producto);*

*$("#formulario-producto input.titulo:eq(" + $contador\_producto + ")").attr("name","titulo-producto-" + $contador\_producto);*

*$("#formulario-producto select.categoria:eq(" + $contador\_producto + ")").attr("name","categoria-" + $contador\_producto);*

*$("#formulario-producto select.foto:eq(" + $contador\_producto + ")").attr("name","product-photo-" + $contador\_producto);*

*$("#formulario-producto input.cantidad:eq(" + $contador\_producto + ")").attr("name","cantidad-" + $contador\_producto);*

*$("#formulario-producto input.compra:eq(" + $contador\_producto + ")").attr("name","compra-" + $contador\_producto);*

*$("#formulario-producto input.compra\_decimal:eq(" + $contador\_producto + ")").attr("name","compra\_decimal-" + $contador\_producto);*

*$("#formulario-producto input.precio:eq(" + $contador\_producto + ")").attr("name","precio-" + $contador\_producto);*

*$("#formulario-producto input.precio\_decimal:eq(" + $contador\_producto + ")").attr("name","precio\_decimal-" + $contador\_producto);*

*$("#formulario-producto .boton-menos-producto:eq(" + $contador\_producto + ")").attr("id","" + $contador\_producto);*

*$('.boton-menos-producto').attr("disabled", false);*

*$("#formulario-producto input.titulo:eq(" + $contador\_producto + ")").val('').focus();*

*$("#formulario-producto select.categoria:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto select.foto:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.cantidad:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.compra:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.compra\_decimal:eq(" + $contador\_producto + ")").val('0');*

*$("#formulario-producto input.precio:eq(" + $contador\_producto + ")").val('');*

*$("#formulario-producto input.precio\_decimal:eq(" + $contador\_producto + ")").val('0');*

*$contador\_producto++;*

*});*

## Función que elimina un producto de un formulario

Esta función *“$(document).on("click",".boton-menos-producto",function eliminarProducto()“* quita un campo para agregar un producto y va contado los que hay en total de forma dinámica. Deshabilita de forma permanente el botón para eliminar el campo de producto si solo hay un campo. Utiliza la misma sintaxis que la función *“eliminarCategoria“*. En la parte de abajo se muestra el código correspondiente.

*$(document).on("click",".boton-menos-producto",function eliminarProducto() {*

*$indices = $(this).attr("id");*

*if($contador\_producto === 2) {*

*$("#contenedor-producto-" + $indices).remove();*

*$contador\_producto--;*

*$('.boton-menos-producto').attr("disabled", true);*

*} else {*

*$("#contenedor-producto-" + $indices).remove();*

*$contador\_producto--;*

*}*

*for(var $i = 0; $i <= $contador\_producto; $i++) {*

*$("#formulario-producto div div.form-producto:eq(" + $i + ")").attr("id","contenedor-producto-" + $i);*

*$("#formulario-producto input.titulo:eq(" + $i + ")").attr("name","titulo-producto-" + $i);*

*$("#formulario-producto select.categoria:eq(" + $i + ")").attr("name","categoria-" + $i);*

*$("#formulario-producto select.foto:eq(" + $i + ")").attr("name","product-photo-" + $i);*

*$("#formulario-producto input.cantidad:eq(" + $i + ")").attr("name","cantidad-" + $i);*

*$("#formulario-producto input.compra:eq(" + $i + ")").attr("name","compra-" + $i);*

*$("#formulario-producto input.compra\_decimal:eq(" + $i + ")").attr("name","compra\_decimal-" + $i);*

*$("#formulario-producto input.precio:eq(" + $i + ")").attr("name","precio-" + $i);*

*$("#formulario-producto input.precio\_decimal:eq(" + $i + ")").attr("name","precio\_decimal-" + $i);*

*$("#formulario-producto .boton-menos-producto:eq(" + $i + ")").attr("id","" + $i);*

*}*

*});*

## Función que guarda los datos de los formularios de productos antes de enviarlos

Esta función *“$('#formulario-producto').submit(function enviarProductos(event)“* cuenta los campos de productos y guarda sus valores, para posteriormente enviarlos al archivo PHP correspondiente a través de Ajax. Utiliza la misma sintaxis que la función *“enviarCategoria“*. En la parte de abajo se muestra el código correspondiente.

*$('#formulario-producto').submit(function enviarProductos(event) {*

*var $contador\_tabla = $(".origen tr").length;*

*$categoria\_productos = new Array($contador\_producto);*

*$imagen\_productos = new Array($contador\_producto);*

*$cantidad\_productos = new Array($contador\_producto);*

*$compra\_productos = new Array($contador\_producto);*

*$compra\_decimal\_productos = new Array($contador\_producto);*

*$precio\_productos = new Array($contador\_producto);*

*$precio\_decimal\_productos = new Array($contador\_producto);*

*$nombre\_imagen\_productos = new Array($contador\_producto);*

*$nombre\_categoria\_productos = new Array($contador\_producto);*

*for(var $i = 0; $i < $contador\_producto; $i++) {*

*$categoria\_productos[$i] = $("#formulario-producto select.categoria:eq(" + $i + ")").val();*

*$imagen\_productos[$i] = $("#formulario-producto select.foto:eq(" + $i + ")").val();*

*$cantidad\_productos[$i] = $("#formulario-producto input.cantidad:eq(" + $i + ")").val();*

*$compra\_productos[$i] = $("#formulario-producto input.compra:eq(" + $i + ")").val();*

*$compra\_decimal\_productos[$i] = $("#formulario-producto input.compra\_decimal:eq(" + $i + ")").val();*

*$precio\_productos[$i] = $("#formulario-producto input.precio:eq(" + $i + ")").val();*

*$precio\_decimal\_productos[$i] = $("#formulario-producto input.precio\_decimal:eq(" + $i + ")").val();*

*$nombre\_imagen\_productos[$i] = $("#formulario-producto select.foto:eq(" + $i + ") option:selected").text();*

*$nombre\_categoria\_productos[$i] = $("#formulario-producto select.categoria:eq(" + $i + ") option:selected").text();*

*}*

*if($(".tabla-resultado-producto tbody tr").length) {*

*$contador\_tabla\_productos\_recientes = $(".tabla-resultado-producto tbody tr").length;*

*$productos\_agregados = new Array($contador\_tabla\_productos\_recientes);*

*for(var $i = 0; $i < $contador\_tabla\_productos\_recientes; $i++) {*

*$productos\_agregados[$i] = $(".tabla-resultado-producto tbody tr td.necesario")[$i].innerHTML;*

*}*

*}*

*$.post("buscar\_producto.php", {productos\_agregados: $productos\_agregados, contador\_tabla\_productos\_recientes: $contador\_tabla\_productos\_recientes, nombre\_categoria\_productos: $nombre\_categoria\_productos, nombre\_imagen\_productos: $nombre\_imagen\_productos, titulo\_productos: $titulo\_productos, categoria\_productos: $categoria\_productos, imagen\_productos: $imagen\_productos, cantidad\_productos: $cantidad\_productos, compra\_productos: $compra\_productos, compra\_decimal\_productos: $compra\_decimal\_productos, precio\_productos: $precio\_productos, precio\_decimal\_productos: $precio\_decimal\_productos, contador\_producto: $contador\_producto, contador\_tabla: $contador\_tabla}, function(mensaje) {*

*$("#resultadoBusqueda-productos").html(mensaje);*

*$("#resultados-productos").html(datos\_productos);*

*});*

*event.preventDefault();*

*});*

## Tabla Categoría

En esta tabla se almacena cada una de las categorías de los productos dadas de alta por los usuarios administradores y especiales en el sistema, almacena la llave primaria de la tabla y el nombre de la categoría. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada categoría que se crea. Es autoincremental ya que se generará automáticamente cada vez que se crea una categoría.

### Campo Nombre (varchar 50)

En el campo de nombre se llevará el registro del nombre de la categoría, es de tipo varchar porque el campo se llenará con el nombre de la categoría.

## Tabla Empresa

En esta tabla se almacena la única empresa, almacena la llave primaria de la tabla, la llave foránea del usuario, la URL del logo de la empresa y el lema de la empresa. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá esta empresa. Es autoincremental ya que se generará automáticamente cada vez que se crea una empresa.

### Campo Id\_usuario (int)

Es la llave externa con la que se relaciona la primary key de la tabla usuario ya que solo un usuario tiene la opción de controlar la empresa, es de tipo entero para que sea compatible con la primary key de usuario.

### Campo Logo (varchar 255)

En el campo de logo se llevará el registro de la URL del logo de la empresa, es de tipo varchar porque el campo se llenará con la URL del logo de la empresa.

### Campo Lema (varchar 255)

En el campo de lema se llevará el registro del lema de la empresa, es de tipo varchar porque el campo se llenará con el lema de la empresa.

## Tabla Grupo

En esta tabla se almacena los grupos de usuarios dados de alta por los usuarios administradores en el sistema, almacena la llave primaria de la tabla, el nombre del grupo, el nivel del grupo y el estado actual del grupo. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada grupo. Es autoincremental ya que se generará automáticamente cada vez que se crea un grupo.

### Campo Nombre (varchar 50)

En el campo de nombre se llevará el registro del nombre del grupo, es de tipo varchar porque el campo se llenará con el nombre del grupo.

### Campo Nivel (int 1)

En el campo de nivel se llevará el registro del nivel del grupo, es de tipo int porque el campo se llenará con el número correspondiente al nivel.

### Campo Estado\_actual (int 1)

En el campo de estado\_actual se llevará el registro del estado del grupo, es de tipo int porque el campo se llenará con el número correspondiente al estado.

## Tabla Media

En esta tabla se almacena las imágenes de productos dadas de alta por los usuarios administradores y especiales en el sistema, almacena la llave primaria de la tabla, el nombre del archivo y el tipo de imagen. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada imagen. Es autoincremental ya que se generará automáticamente cada vez que se crea una imagen.

### Campo Nombre\_archivo (varchar 255)

En el campo de nombre\_archivo se llevará el registro del nombre de la imagen, es de tipo varchar porque el campo se llenará con el nombre de la imagen.

### Campo Tipo\_archivo (varchar 100)

En el campo de tipo\_archivo se llevará el registro del tipo de archivo de la imagen, es de tipo varchar porque el campo se llenará con el tipo de archivo de la imagen.

## Tabla Producto

En esta tabla se almacena los productos dados de alta por los usuarios administradores y especiales en el sistema, almacena la llave primaria de la tabla, el nombre del producto, la cantidad existente del producto, precio de compra del producto, precio de venta del producto, la llave foránea de la categoría, la llave foránea de la imagen de producto y la fecha de actualización. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada producto. Es autoincremental ya que se generará automáticamente cada vez que se crea un producto.

### Campo Nombre (varchar 255)

En el campo de nombre se llevará el registro del nombre del producto, es de tipo varchar porque el campo se llenará con el nombre del producto.

### Campo Cantidad (int)

En el campo de cantidad se llevará el registro de la cantidad de producto existente, es de tipo int porque el campo se llenará con la cantidad de producto.

### Campo Precio\_compra (decimal 25,2)

En el campo de precio\_compra se llevará el registro del precio de compra de cada producto, es de tipo decimal porque el campo se llenará con el precio de compra.

### Campo Precio\_venta (decimal 25,2)

En el campo de precio\_venta se llevará el registro del precio de venta de cada producto, es de tipo decimal porque el campo se llenará con el precio de venta.

### Campo Id\_categoria (int)

Es la llave externa con la que se relaciona la primary key de la tabla categoría ya que cada uno de los productos tiene una categoría, es de tipo entero para que sea compatible con la primary key de categoría.

### Campo Id\_media (int)

Es la llave externa con la que se relaciona la primary key de la tabla media ya que cada uno de los productos puede tener una imagen, es de tipo entero para que sea compatible con la primary key de media.

### Campo Fecha\_actualizacion (datetime)

En el campo de fecha se llevará el registro de cuándo se actualizó el producto, es de tipo datetime porque el campo se llenará con la fecha actual.

## Tabla Producto\_eliminado

En esta tabla se almacena los productos eliminados en el sistema, almacena la llave primaria de la tabla, el nombre del producto, la cantidad existente del producto, precio de compra del producto, precio de venta del producto, la llave foránea de la categoría, la llave foránea de la imagen de producto y la fecha de actualización. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada producto. Es autoincremental ya que se generará automáticamente cada vez que se crea un producto.

### Campo Nombre (varchar 255)

En el campo de nombre se llevará el registro del nombre del producto, es de tipo varchar porque el campo se llenará con el nombre del producto.

### Campo Cantidad (int)

En el campo de cantidad se llevará el registro de la cantidad de producto existente, es de tipo int porque el campo se llenará con la cantidad de producto.

### Campo Precio\_compra (decimal 25,2)

En el campo de precio\_compra se llevará el registro del precio de compra de cada producto, es de tipo decimal porque el campo se llenará con el precio de compra.

### Campo Precio\_venta (decimal 25,2)

En el campo de precio\_venta se llevará el registro del precio de venta de cada producto, es de tipo decimal porque el campo se llenará con el precio de venta.

### Campo Id\_categoria (int)

Es la llave externa con la que se relaciona la primary key de la tabla categoría ya que cada uno de los productos tiene una categoría, es de tipo entero para que sea compatible con la primary key de categoría.

### Campo Id\_media (int)

Es la llave externa con la que se relaciona la primary key de la tabla media ya que cada uno de los productos puede tener una imagen, es de tipo entero para que sea compatible con la primary key de media.

### Campo Fecha\_actualizacion (datetime)

En el campo de fecha se llevará el registro de cuándo se actualizó el producto, es de tipo datetime porque el campo se llenará con la fecha actual.

## Tabla Producto\_respaldo

En esta tabla se almacena los productos respaldados en el sistema al ser agregados por primera vez, almacena la llave primaria de la tabla, el nombre del producto, la cantidad existente del producto, precio de compra del producto, precio de venta del producto, la llave foránea de la categoría, la llave foránea de la imagen de producto y la fecha de actualización. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada producto. Es autoincremental ya que se generará automáticamente cada vez que se crea un producto.

### Campo Nombre (varchar 255)

En el campo de nombre se llevará el registro del nombre del producto, es de tipo varchar porque el campo se llenará con el nombre del producto.

### Campo Cantidad (int)

En el campo de cantidad se llevará el registro de la cantidad de producto existente, es de tipo int porque el campo se llenará con la cantidad de producto.

### Campo Precio\_compra (decimal 25,2)

En el campo de precio\_compra se llevará el registro del precio de compra de cada producto, es de tipo int porque el campo se llenará con el precio de compra.

### Campo Precio\_venta (decimal 25,2)

En el campo de precio\_venta se llevará el registro del precio de venta de cada producto, es de tipo int porque el campo se llenará con el precio de venta.

### Campo Id\_categoria (int)

Es la llave externa con la que se relaciona la primary key de la tabla categoría ya que cada uno de los productos tiene una categoría, es de tipo entero para que sea compatible con la primary key de categoría.

### Campo Id\_media (int)

Es la llave externa con la que se relaciona la primary key de la tabla media ya que cada uno de los productos puede tener una imagen, es de tipo entero para que sea compatible con la primary key de media.

### Campo Fecha\_actualizacion (datetime)

En el campo de fecha se llevará el registro de cuándo se actualizó el producto, es de tipo datetime porque el campo se llenará con la fecha actual.

## Tabla Rango\_fecha

En esta tabla se almacena los rangos de fecha para los reportes de ventas del sistema, almacena la llave primaria de la tabla, la llave foránea de reporte de usuario, la fecha de inicio y la fecha de fin. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada rango de fecha. Es autoincremental ya que se generará automáticamente cada vez que se crea un rango de fecha.

### Campo Id\_tipos\_reporte (int)

Es la llave externa con la que se relaciona la primary key de la tabla reporte de usuario ya que cada uno de los reportes tiene relación a un usuario, es de tipo entero para que sea compatible con la primary key de reporte de usuario.

### Campo Fecha\_inicio (datetime)

En el campo de fecha\_inicio se llevará el registro de cuál es la fecha de inicio, es de tipo datetime porque el campo se llenará con la fecha que escogió el usuario.

### Campo Fecha\_fin (datetime)

En el campo de fecha\_fin se llevará el registro de cuál es la fecha de fin, es de tipo datetime porque el campo se llenará con la fecha que escogió el usuario.

## Tabla Reporte

En esta tabla se almacena los nombres para los reportes de ventas del sistema, almacena la llave primaria de la tabla y el nombre del tipo de reporte. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada reporte. Es autoincremental ya que se generará automáticamente cada vez que se crea un tipo de reporte.

### Campo Nombre (varchar 50)

En el campo de nombre se llevará el registro del nombre del reporte, es de tipo varchar porque el campo se llenará con el nombre del reporte.

## Tabla Reporte\_usuario

En esta tabla se almacena las relaciones de los reportes de ventas con los usuarios del sistema, almacena la llave primaria de la tabla, la llave foránea del usuario y la llave foránea del reporte. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada reporte de usuario. Es autoincremental ya que se generará automáticamente cada vez que se crea un reporte de usuario.

### Campo Id\_usuario (int)

Es la llave externa con la que se relaciona la primary key de la tabla usuario ya que cada uno de los reportes de usuario tiene uno, es de tipo entero para que sea compatible con la primary key de usuario.

### Campo Id\_reporte (int)

Es la llave externa con la que se relaciona la primary key de la tabla reporte ya que cada uno de los reportes de usuario tiene uno, es de tipo entero para que sea compatible con la primary key de reporte.

## Tabla Tema

En esta tabla se almacena los nombres de los temas de la aplicación, almacena la llave primaria de la tabla y el nombre del tema. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada tema. Es autoincremental ya que se generará automáticamente cada vez que se crea un tema.

### Campo Nombre (varchar 50)

En el campo de nombre se llevará el registro del nombre del tema, es de tipo varchar porque el campo se llenará con el nombre del tema.

## Tabla Tema\_personal

En esta tabla se almacena el tema personal de cada usuario, almacena la llave primaria de la tabla, la llave foránea del usuario y la llave foránea del tema. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada tema con su usuario. Es autoincremental ya que se generará automáticamente cada vez que se crea un tema personal.

### Campo Id\_usuario (int)

Es la llave externa con la que se relaciona la primary key de la tabla usuario ya que cada uno de los reportes de usuario tiene uno, es de tipo entero para que sea compatible con la primary key de usuario.

### Campo Id\_reporte (int)

Es la llave externa con la que se relaciona la primary key de la tabla tema ya que cada uno de los reportes de usuario tiene uno, es de tipo entero para que sea compatible con la primary key de tema.

## Tabla Usuario

En esta tabla se almacena los usuarios dados de alta por los usuarios administradores en el sistema, almacena la llave primaria de la tabla, la llave foránea de grupo, nombre de la persona dueña de la cuenta de usuario, nombre del usuario, contraseña, imagen de perfil, imagen de fondo, estado de la cuenta de usuario y la fecha de último acceso. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada usuario. Es autoincremental ya que se generará automáticamente cada vez que se crea un usuario.

### Campo Id\_grupo\_usuario (int)

Es la llave externa con la que se relaciona la primary key de la tabla grupo ya que cada uno de los usuarios pertenece a un grupo, es de tipo entero para que sea compatible con la primary key de grupo.

### Campo Nombre (varchar 50)

En el campo de nombre se llevará el registro del nombre de la persona dueña de la cuenta de usuario, es de tipo varchar porque el campo se llenará con el nombre de la persona.

### Campo Username (varchar 50)

En el campo de username se llevará el registro del nombre de usuario, es de tipo varchar porque el campo se llenará con el nombre de usuario.

### Campo Contrasena (varchar 255)

En el campo de contrasena se llevará el registro de la contraseña de usuario, es de tipo varchar porque el campo se llenará con la contraseña del usuario.

### Campo Imagen (varchar 255)

En el campo de imagen se llevará el registro de la dirección del directorio de la imagen, es de tipo varchar porque el campo se llenará con la dirección de la imagen.

### Campo Imagen\_fondo (varchar 255)

En el campo de imagen\_fondo se llevará el registro de la dirección del directorio de la imagen de fondo, es de tipo varchar porque el campo se llenará con la dirección de la imagen de fondo.

### Campo Estado (int 1)

En el campo de estado se llevará el registro del estado de la cuenta del usuario, es de tipo int porque el campo se llenará con el número correspondiente al estado.

### Campo Ultimo\_acceso (datetime)

En el campo de ultimo\_acceso se llevará el registro de cuál fue la fecha de último inicio de sesión, es de tipo datetime porque el campo se llenará con la última fecha en la que ingresó el usuario al sistema.

## Tabla Venta

En esta tabla se almacena las ventas de productos realizadas por cualquier usuario, almacena la llave primaria de la tabla, la llave foránea del usuario, la llave foránea del producto, la cantidad de producto vendido, el precio total de la venta y la fecha en que se realizó la venta. Los campos de la tabla son:

### Campo Id (int)

Id se refiere a la *“Primary key “*o la identificación que tendrá cada venta. Es autoincremental ya que se generará automáticamente cada vez que se crea una venta.

### Campo Id\_usuario (int)

Es la llave externa con la que se relaciona la primary key de la tabla usuario que realizó la venta, es de tipo entero para que sea compatible con la primary key de usuario.

### Campo Id\_producto (int)

Es la llave externa con la que se relaciona la primary key de la tabla producto ya que en cada una de las ventas se registra un producto, es de tipo entero para que sea compatible con la primary key de producto.

### Campo Cantidad (int)

En el campo de cantidad se llevará el registro de la cantidad de producto vendido, es de tipo int porque el campo se llenará con la cantidad de producto vendido.

### Campo Precio (decimal 25,2)

En el campo de precio se llevará el registro del precio de venta de cada producto, es de tipo decimal porque el campo se llenará con el precio de venta.

### Campo Fecha (date)

En el campo de fecha se llevará el registro de cuál fue la fecha en que se registró la venta, es de tipo date porque el campo se llenará con la última fecha en que se efectuó la venta.

## Vistas complementarias

Todas estas vistas se encuentran en la carpeta *“diseno”* ubicado en la carpeta principal.

## Cabecera de la aplicación web

En esta sección de vista se observa la cabecera de las páginas web del sitio.

Se muestra el nombre del sitio seguido del nombre de la empresa, la fecha con hora y finalmente una lista desplegable perteneciente al perfil del usuario actual.

Al dar clic a la lista desplegable se podrá ver el perfil de usuario, la configuración de la cuenta o cerrar sesión.

A continuación, se muestra el código y la pantalla.

*<?php $usuario = usuario\_actual(); ?>*

*<?php $empresa = empresa(); ?>*

*<?php $tema = tema(); ?>*

*<!DOCTYPE html>*

*<html lang="es">*

*<head>*

*<meta charset="UTF-8">*

*<title>*

*<?php*

*if (!empty($titulo\_pagina)) {*

*echo remover\_basura($titulo\_pagina);*

*} elseif (!empty($usuario)) {*

*echo ucfirst($usuario['nombre\_usuario']);*

*} else {*

*echo "Sistema de inventario";*

*}*

*?>*

*</title>*

*<link rel="shortcut icon" type="image/x-icon" href="cargas/icono/stock.ico" />*

*<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css"/>*

*<?php*

*$numero = intval($tema['id\_tema']);*

*switch ($numero) {*

*case 1:*

*echo "<link rel='stylesheet' href='libs/css/main.css' />";*

*break;*

*case 2:*

*echo "<link rel='stylesheet' href='libs/css/pastel.css' />";*

*break;*

*case 3:*

*echo "<link rel='stylesheet' href='libs/css/personalizado1.css' />";*

*break;*

*case 4:*

*echo "<link rel='stylesheet' href='libs/css/personalizado2.css' />";*

*break;*

*case 5:*

*echo "<link rel='stylesheet' href='libs/css/personalizado3.css' />";*

*break;*

*default:*

*break;*

*}*

*?>*

*<link rel="stylesheet" href="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css" />*

*<script src="http://code.jquery.com/jquery-1.9.1.js"></script>*

*<script src="http://code.jquery.com/ui/1.10.1/jquery-ui.js"></script>*

*<script src="libs/js/fecha.js"></script>*

*<script>*

*$(function checarFecha() {*

*$.datepicker.setDefaults($.datepicker.regional["es"]);*

*var $ancho = $(window).width();*

*if($ancho < 890) {*

*$("div.header-date").css("display", "none");*

*} else {*

*$("div.header-date").css("display", "inline");*

*}*

*$(window).resize(function dimensionarChecarFecha(){*

*//aquí el código que se ejecutara cuando se redimensione la ventana*

*$ancho = $(window).width();*

*if($ancho < 890) {*

*$("div.header-date").css("display", "none");*

*} else {*

*$("div.header-date").css("display", "inline");*

*}*

*});*

*});*

*</script>*

*</head>*

*<body>*

*<?php if ($sesion->estaLogeado(true)): ?>*

*<header id="header">*

*<div class="logo pull-left"> Stock-Web </div>*

*<div class="header-content">*

*<div class="header-date pull-left">*

*<?php if($empresa['logo'] !== ''): ?>*

*<img src="cargas/logo/<?php echo $empresa['logo']; ?>" width="50" height="50" alt="logo"/>*

*<?php endif;?>*

*<?php if($empresa['lema'] !== ''): ?>*

*<strong><?php echo '"' . $empresa['lema'] . '"'; ?></strong>*

*<?php endif;?>*

*<strong><?php date\_default\_timezone\_set('America/Monterrey'); echo date("d/m/Y g:i a");?></strong>*

*</div>*

*<div class="pull-right clearfix">*

*<ul class="info-menu list-inline list-unstyled">*

*<li class="profile">*

*<a href="#" data-toggle="dropdown" class="toggle" aria-expanded="false">*

*<img src="cargas/usuarios/<?php echo $usuario['imagen'];?>" alt="user-image" class="img-circle img-inline">*

*<span><?php echo remover\_basura(ucfirst($usuario['username'])); ?> <i class="caret"></i></span>*

*</a>*

*<ul class="dropdown-menu">*

*<li>*

*<a href="perfil.php?id=<?php echo (int)$usuario['id'];?>">*

*<i class="glyphicon glyphicon-user"></i>*

*Perfil*

*</a>*

*</li>*

*<li>*

*<a href="editar\_cuenta.php">*

*<i class="glyphicon glyphicon-cog"></i>*

*Configuración*

*</a>*

*</li>*

*<li class="last">*

*<a href="cerrar\_sesion.php">*

*<i class="glyphicon glyphicon-off"></i>*

*Salir*

*</a>*

*</li>*

*</ul>*

*</li>*

*</ul>*

*</div>*

*</div>*

*</header>*

*<div class="sidebar">*

*<?php if($usuario['nivel'] === '1'): ?>*

*<?php include\_once('menu\_administrador.php');?>*

*<?php elseif($usuario['nivel'] === '2'): ?>*

*<?php include\_once('menu\_especial.php');?>*

*<?php elseif($usuario['nivel'] === '3'): ?>*

*<?php include\_once('menu\_usuario.php');?>*

*<?php endif;?>*

*</div>*

*<?php endif;?>*

*<?php*

*if($usuario['imagen\_fondo'] != "") {*

*echo "<img style='position: relative; height: 75em' src='cargas/fondos/$usuario[imagen\_fondo]' width='100%' />";*

*}*

*?>*

*<div class="page" style="position:absolute;display:block;padding:45px 15px 20px 263px;">*

*<div class="container-fluid">*

En la ilustración 31 se muestra el resultado de la codificación del código de arriba.

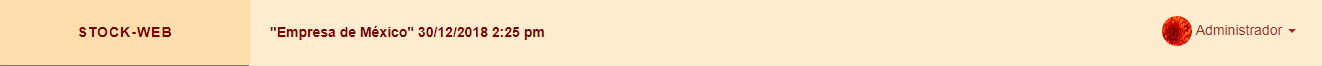


Ilustración 31 Cabecera

## Menú de usuarios administradores

En esta sección de vista se observa el menú de los usuarios administradores con vínculos a las secciones del sitio correspondientes.

Este menú se muestra debajo de la cabecera, más concretamente debajo del nombre del sitio.

Al dar clic a un vínculo, deberá desplegar una lista con más vínculos o dirigir al usuario a la correspondiente sección.

A continuación, se muestra el código y la pantalla.

*<ul>*

*<li>*

*<a href="admin.php">*

*<i class="glyphicon glyphicon-home"></i>*

*<span>Panel de control</span>*

*</a>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-user"></i>*

*<span>Accesos</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="grupo.php">Administrar grupos</a> </li>*

*<li><a href="usuarios.php">Administrar usuarios</a> </li>*

*</ul>*

*</li>*

*<li>*

*<a href="categoria.php" >*

*<i class="glyphicon glyphicon-indent-left"></i>*

*<span>Categorías</span>*

*</a>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-th-large"></i>*

*<span>Productos</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="producto.php">Administrar productos</a> </li>*

*<li><a href="agregar\_producto.php">Agregar productos</a> </li>*

*<li><a href="productos\_eliminados.php">Productos eliminados</a> </li>*

*</ul>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-picture"></i>*

*<span>Media</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="media.php">Administrar imágenes</a> </li>*

*<li><a href="agregar\_multiples\_registros.php">Agregar múltiples registros</a> </li>*

*</ul>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-th-list"></i>*

*<span>Ventas</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="ventas.php">Administrar ventas</a> </li>*

*<li><a href="agregar\_venta.php">Agregar venta</a> </li>*

*</ul>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-signal"></i>*

*<span>Reporte de ventas</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="reporte\_ventas.php">Ventas por fecha </a></li>*

*<li><a href="ventas\_mensuales.php">Ventas mensuales</a></li>*

*<li><a href="ventas\_diarias.php">Ventas diarias</a> </li>*

*</ul>*

*</li>*

*</ul>*

En la ilustración 32 se muestra el resultado de la codificación del código de arriba.

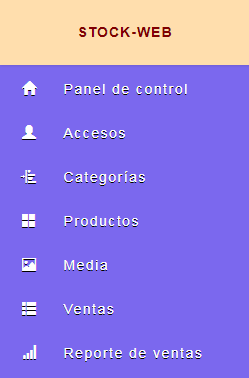


Ilustración 32 Menú de usuarios administradores

## Menú de usuarios especiales

En esta sección de vista se observa el menú de los usuarios especiales con vínculos a las secciones del sitio correspondientes.

Este menú se muestra debajo de la cabecera, más concretamente debajo del nombre del sitio.

Al dar clic a un vínculo, deberá desplegar una lista con más vínculos o dirigir al usuario a la correspondiente sección.

A continuación, se muestra el código y la pantalla.

*<ul>*

*<li>*

*<a href="admin.php">*

*<i class="glyphicon glyphicon-home"></i>*

*<span>Panel de control</span>*

*</a>*

*</li>*

*<li>*

*<a href="categoria.php" >*

*<i class="glyphicon glyphicon-indent-left"></i>*

*<span>Categorías</span>*

*</a>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-th-large"></i>*

*<span>Productos</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="producto.php">Administrar productos</a> </li>*

*<li><a href="agregar\_producto.php">Agregar producto</a> </li>*

*<li><a href="productos\_eliminados.php">Productos eliminados</a> </li>*

*</ul>*

*</li>*

*<li>*

*<a href="media.php" >*

*<i class="glyphicon glyphicon-picture"></i>*

*<span>Media</span>*

*</a>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-th-list"></i>*

*<span>Ventas</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="ventas.php">Administrar ventas</a> </li>*

*<li><a href="agregar\_venta.php">Agregar venta</a> </li>*

*</ul>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-signal"></i>*

*<span>Reporte de ventas</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="reporte\_ventas.php">Ventas por fecha </a></li>*

*<li><a href="ventas\_mensuales.php">Ventas mensuales</a></li>*

*<li><a href="ventas\_diarias.php">Ventas diarias</a> </li>*

*</ul>*

*</li>*

*</ul>*

En la ilustración 33 se muestra el resultado de la codificación del código de arriba.

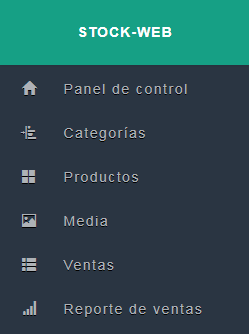


Ilustración 33 Menú de usuarios especiales

## Menú de usuarios estándares

En esta sección de vista se observa el menú de los usuarios estándares con vínculos a las secciones del sitio correspondientes.

Este menú se muestra debajo de la cabecera, más concretamente debajo del nombre del sitio.

Al dar clic a un vínculo, deberá desplegar una lista con más vínculos o dirigir al usuario a la correspondiente sección.

A continuación, se muestra el código y la pantalla.

*<ul>*

*<li>*

*<a href="admin.php">*

*<i class="glyphicon glyphicon-home"></i>*

*<span>Panel de control</span>*

*</a>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-th-list"></i>*

*<span>Ventas</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="ventas.php">Administrar ventas</a> </li>*

*<li><a href="agregar\_venta.php">Agregar venta</a> </li>*

*</ul>*

*</li>*

*<li>*

*<a href="#" class="submenu-toggle">*

*<i class="glyphicon glyphicon-signal"></i>*

*<span>Reporte de ventas</span>*

*</a>*

*<ul class="nav submenu">*

*<li><a href="reporte\_ventas.php">Ventas por fecha </a></li>*

*<li><a href="ventas\_mensuales.php">Ventas mensuales</a></li>*

*<li><a href="ventas\_diarias.php">Ventas diarias</a> </li>*

*</ul>*

*</li>*

*</ul>*

En la ilustración 34 se muestra el resultado de la codificación del código de arriba.



Ilustración 34 Menú de usuarios estándares

## Pie de la aplicación web

En esta sección de vista se cargan los archivos de JavaScript y se cierra el documento HTML de la página actual.

A continuación, se muestra el código.

*</div>*

*</div>*

*<!--<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>-->*

*<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/js/bootstrap.min.js"></script>*

*<script type="text/javascript" src="libs/js/functions.js"></script>*

*</body>*

*</html>*

*<?php if(isset($db)) { $db->db\_disconnect(); }*

## Vistas y archivos utilitarios

Todos estos archivos y vistas se encuentran en la carpeta principal.

## Panel de control de administración

En esta vista se observa en la parte superior el número de usuarios, el número de categorías de productos, el número de productos y el número de ventas. En la parte de abajo se muestran 5 paneles: productos más vendidos, últimas ventas, productos recientemente añadidos, productos con bajo stock y reportes automáticos.

Al dar clic a la imagen con vínculo en el panel de productos recientemente añadidos, deberá direccionar a la sección para editar el producto.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Admin página de inicio';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$contar\_categoria = contar\_por\_id('categoria');*

*$contar\_producto = contar\_por\_id('producto');*

*$contar\_venta = contar\_por\_id('venta');*

*$contar\_usuario = contar\_por\_id('usuario');*

*$productos\_vendidos = encontrar\_productos\_altamente\_vendidos('10');*

*$productos\_recientes = encontrar\_productos\_recientemente\_anadidos('5');*

*$ventas\_recientes = encontrar\_todas\_ventas\_agregadas\_recientemente('5');*

*$productos\_bajo = encontrar\_productos\_bajo\_stock('10');*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-3">*

*<div class="panel panel-box clearfix">*

*<div class="panel-icon pull-left bg-green">*

*<i class="glyphicon glyphicon-user" style="font-size:20px;margin-left:-5px"></i>*

*</div>*

*<div class="panel-value pull-right">*

*<h2 class="margin-top"><?php echo $contar\_usuario['total']; ?></h2>*

*<p class="text-muted">Usuarios</p>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-3">*

*<div class="panel panel-box clearfix">*

*<div class="panel-icon pull-left bg-red">*

*<i class="glyphicon glyphicon-list" style="font-size:20px;margin-left:-5px"></i>*

*</div>*

*<div class="panel-value pull-right">*

*<h2 class="margin-top"> <?php echo $contar\_categoria['total']; ?> </h2>*

*<p class="text-muted">Categorías</p>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-3">*

*<div class="panel panel-box clearfix">*

*<div class="panel-icon pull-left bg-blue">*

*<i class="glyphicon glyphicon-shopping-cart" style="font-size:20px;margin-left:-5px"></i>*

*</div>*

*<div class="panel-value pull-right">*

*<h2 class="margin-top"> <?php echo $contar\_producto['total']; ?> </h2>*

*<p class="text-muted">Productos</p>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-3">*

*<div class="panel panel-box clearfix">*

*<div class="panel-icon pull-left bg-yellow">*

*<i class="glyphicon glyphicon-usd" style="font-size:20px;margin-left:-5px"></i>*

*</div>*

*<div class="panel-value pull-right">*

*<h2 class="margin-top"> <?php echo $contar\_venta['total']; ?></h2>*

*<p class="text-muted">Ventas</p>*

*</div>*

*</div>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-4">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Productos más vendidos</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-striped table-bordered table-condensed">*

*<thead>*

*<tr>*

*<th style="vertical-align:middle">Título</th>*

*<th style="vertical-align:middle">Cantidad total</th>*

*<tr>*

*</thead>*

*<tbody>*

*<?php foreach ($productos\_vendidos as $producto\_vendido): ?>*

*<tr>*

*<td><?php echo remover\_basura(primer\_caracter($producto\_vendido['nombre'])); ?></td>*

*<td><?php echo (int)$producto\_vendido['totalQty']; ?></td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-4">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>ÚLTIMAS VENTAS</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-striped table-bordered table-condensed">*

*<thead>*

*<tr>*

*<th class="text-center" style="width:50px;vertical-align:middle;">#</th>*

*<th style="vertical-align:middle;">Producto</th>*

*<th style="vertical-align:middle;width:30%;">Fecha</th>*

*<th style="vertical-align:middle;width:30%;">Venta total</th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($ventas\_recientes as $venta\_reciente): ?>*

*<tr>*

*<td class="text-center"><?php echo contar\_id();?></td>*

*<td>*

*<a href="editar\_venta.php?id=<?php echo (int)$venta\_reciente['id']; ?>">*

*<?php echo remover\_basura(primer\_caracter($venta\_reciente['nombre'])); ?>*

*</a>*

*</td>*

*<td><?php echo remover\_basura(ucfirst($venta\_reciente['fecha'])); ?></td>*

*<td>$<?php echo remover\_basura(primer\_caracter($venta\_reciente['precio'])); ?></td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-4">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Productos recientemente añadidos</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<div class="list-group">*

*<?php foreach ($productos\_recientes as $producto\_reciente): ?>*

*<a class="list-group-item clearfix" href="editar\_producto.php?id=<?php echo (int)$producto\_reciente['id'];?>">*

*<h4 class="list-group-item-heading">*

*<?php if($producto\_reciente['id\_media'] === '0'): ?>*

*<img class="img-avatar img-circle" src="cargas/productos/no\_image.jpg" alt="">*

*<?php else: ?>*

*<img class="img-avatar img-circle" src="cargas/productos/<?php echo $producto\_reciente['image'];?>" alt="" />*

*<?php endif;?>*

*<?php echo remover\_basura(primer\_caracter($producto\_reciente['nombre']));?>*

*<span class="label label-warning pull-right">*

*$<?php echo (int)$producto\_reciente['precio\_venta']; ?>*

*</span>*

*</h4>*

*<span class="list-group-item-text pull-right">*

*<?php echo "<span style='margin-right:10px'>".remover\_basura(primer\_caracter($producto\_reciente['categorie']))."</span>"; ?>*

*</span>*

*</a>*

*<?php endforeach; ?>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php if($contar\_producto['total'] > 0): ?>*

*<div class="col-md-4">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Productos con bajo stock</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-striped table-bordered table-condensed">*

*<thead>*

*<tr>*

*<th style="vertical-align:middle;">Título</th>*

*<th style="vertical-align:middle;">Cantidad</th>*

*<tr>*

*</thead>*

*<tbody>*

*<?php foreach ($productos\_bajo as $producto): ?>*

*<tr>*

*<td><?php echo remover\_basura(primer\_caracter($producto['nombre'])); ?></td>*

*<td><?php echo (int)$producto['cantidad']; ?></td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*<?php endif; ?>*

*<?php*

*$reportes = encontrar\_tipos\_reporte\_por\_usuario($usuario['id']);*

*?>*

*<div class="col-md-4">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Reportes automáticos</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-striped table-bordered table-condensed">*

*<thead>*

*<tr>*

*<th style="vertical-align:middle;">Tipo</th>*

*<th style="vertical-align:middle;">Reporte</th>*

*<tr>*

*</thead>*

*<tbody>*

*<?php*

*function \_data\_last\_month\_day() {*

*$month = date('m');*

*$year = date('Y');*

*$day = date("d", mktime(0,0,0, $month+1, 0, $year));*

*return date('Y-m-d', mktime(0,0,0, $month, $day, $year));*

*}*

*function \_data\_first\_month\_day() {*

*$month = date('m');*

*$year = date('Y');*

*return date('Y-m-d', mktime(0,0,0, $month, 1, $year));*

*}*

*?>*

*<?php foreach ($reportes as $reporte): ?>*

*<tr>*

*<?php*

*if($reporte['id\_reporte'] == 1) {*

*echo "<td> Diario </td>";*

*$fecha\_inicio = date('Y-m-d');*

*echo "<td><a href='http://localhost/13100034/reporte\_ventas\_proceso.php?fecha-inicio=$fecha\_inicio&tipo=1'>Ver</a></td>";*

*} elseif($reporte['id\_reporte'] == 2) {*

*echo "<td> Mensual </td>";*

*$fecha\_inicio = \_data\_first\_month\_day();*

*$fecha\_fin = \_data\_last\_month\_day();*

*echo "<td><a href='http://localhost/13100034/reporte\_ventas\_proceso.php?fecha-inicio=$fecha\_inicio&fecha-fin=$fecha\_fin&tipo=2'>Ver</a></td>";*

*} else {*

*echo "<td> Por fechas </td>";*

*$resultados = buscarTipoReportes($usuario['id'], 3);*

*foreach($resultados as $resultado):*

*$id\_tipos\_reporte = $resultado['id'];*

*endforeach;*

*$rangos = buscarRangoFechas($id\_tipos\_reporte);*

*foreach($rangos as $rango):*

*$fecha\_inicio = $rango['fecha\_inicio'];*

*$fecha\_inicio = substr($fecha\_inicio, 0, -9);*

*$fecha\_fin = $rango['fecha\_fin'];*

*$fecha\_fin = substr($fecha\_fin, 0, -9);*

*endforeach;*

*echo "<td><a href='http://localhost/13100034/reporte\_ventas\_proceso.php?fecha-inicio=$fecha\_inicio&fecha-fin=$fecha\_fin&tipo=2'>Ver</a></td>";*

*}*

*?>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 35 se muestra el resultado de la codificación del código de arriba.

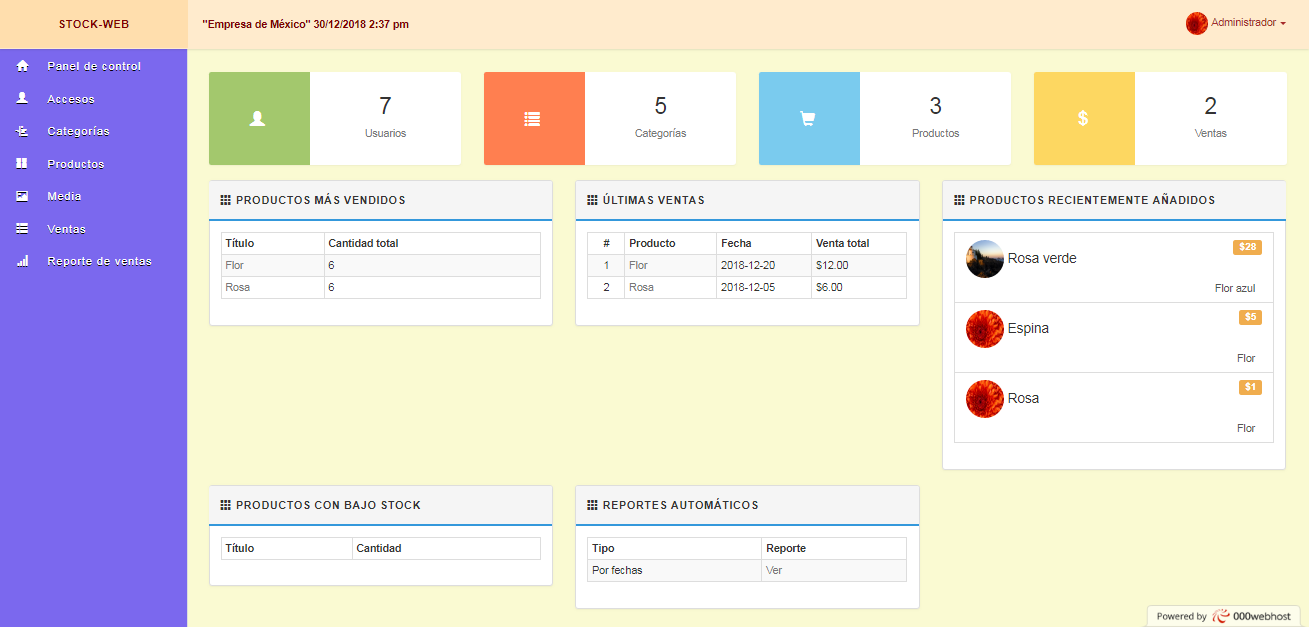


Ilustración 35 Panel de control de administración

## Sección para agregar grupo

En esta vista se observa el formulario para agregar un nuevo grupo de usuarios.

El formulario cuenta con campos para agregar el nombre de grupo, el nivel de grupo y el estado de grupo.

Al ingresar el nombre de grupo, no debe existir otro con nombre igual.

Al ingresar el nivel de grupo, este podrá tener como valor el rango 1-3.

Al ingresar el estado de grupo, este podrá tener como valor activo o inactivo.

Al dar clic en el botón de agregar grupo, deberá agregar el grupo.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Agregar grupo';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="login-page" style='margin-top:-2px'>*

*<div class="text-center">*

*<h3>Agregar nuevo grupo de usuarios</h3>*

*</div>*

*<div id="resultado"></div>*

*<div id="valor"></div>*

*<form method="POST" class="clearfix" id="formulario">*

*<div class="form-group">*

*<label for="nombre\_grupo" class="control-label">Nombre del grupo</label>*

*<input id="nombre\_grupo" autofocus="" type="name" class="form-control" name="nombre\_grupo" required>*

*</div>*

*<div class="form-group">*

*<label for="nivel\_grupo" class="control-label">Nivel del grupo</label>*

*<input id="nivel\_grupo" type="number" min="1" max="3" class="form-control" name="nivel\_grupo" required="">*

*</div>*

*<div class="form-group">*

*<label for="estado">Estado</label>*

*<select id="estado" class="form-control" name="estado">*

*<option value="1">Activo</option>*

*<option value="0">Inactivo</option>*

*</select>*

*</div>*

*<div class="form-group clearfix">*

*<button type="submit" name="agregar" class="btn btn-info">Guardar</button>*

*</div>*

*</form>*

*</div>*

*<script src="libs/js/agregar\_grupo.js"></script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 36 se muestra el resultado de la codificación del código de arriba.



Ilustración 36 Sección para agregar grupo

## Sección para agregar múltiples registros

En esta vista se observa el par de formularios para agregar una o más categorías y/o productos.

El par de formularios cuentan con los campos correspondientes para agregar una categoría y un producto.

Se da clic en el botón “Menos” de cualquier formulario para quitar un formulario.

Se da clic en el botón “Más” de cualquier formulario para agregar un formulario.

Se da clic en el botón “Agregar categoría(s)” para agregar todos los formularios de categorías llenados.

Se da clic en el botón “Agregar producto(s)” para agregar todos los formularios de productos llenados.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Múltiples registros';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$categorias = encontrar\_todos('categoria');*

*$productos = unir\_tabla\_producto();*

*$todas\_categorias = $categorias;*

*$todas\_fotos = encontrar\_todos('media');*

*include\_once('disenos/cabecera.php');*

*?>*

*<style>*

*tr:nth-child(odd) {*

*background-color:#f2f2f2;*

*}*

*tr:nth-child(even) {*

*background-color:#fbfbfb;*

*}*

*</style>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span class="glyphicon glyphicon-equalizer"></span>*

*<span>Agregar varias categorías</span>*

*</div>*

*<div class="panel-body" style="margin-top:-20px">*

*<div>*

*<h3 style="padding:10px" class="bg-primary text-center pad-basic no-btm">Categorías</h3>*

*<div class="panel panel-default">*

*<table id='tabla\_categorias' class="table table-bordered table-striped table-hover">*

*<thead>*

*<tr>*

*<th class="text-center" style="width: 50px;">#</th>*

*<th>Categorías</th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($categorias as $categoria):?>*

*<tr>*

*<td class="text-center"><?php echo contar\_id();?></td>*

*<td><?php echo remover\_basura(ucfirst($categoria['nombre'])); ?></td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*<form method="POST" id="formulario">*

*<div id="resultadoBusqueda"></div>*

*<div id="resultados"></div>*

*<h3 style="padding:10px" class="bg-primary text-center pad-basic no-btm">Agregar categorías</h3>*

*<table class="table bg-info" id="tabla">*

*<tr class="fila-fija form-group" id="fila">*

*<td><input required="" class="form-control" autofocus="" id="nombre-categoria" type="text" name="nombre\_categoria-0" placeholder="Nombre de categoría"/></td>*

*<td class="eliminar"><button style="margin-top:4px" class="boton-menos btn-info" disabled="" type="button" >Menos -</button></td>*

*</tr>*

*</table>*

*<div class="btn-der">*

*<input type="submit" id="agregarCategoria" name="insertar" value="Insertar categoría(s)" class="btn btn-info"/>*

*<button id="adicional" name="adicional\_producto" type="button" class="btn btn-warning"> Más + </button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-12 panel-default panel">*

*<div class="panel-heading clearfix" style="margin-left:-15px;margin-right:-15px">*

*<span class="glyphicon glyphicon-equalizer"></span>*

*<span>Agregar varios productos</span>*

*</div>*

*<div class="panel-body" style="margin-top:-20px;margin-left:-15px;margin-right:-15px">*

*<div>*

*<h3 style="padding:10px" class="bg-primary text-center pad-basic no-btm">Productos</h3>*

*<div class="panel panel-default">*

*<table class="table table-bordered table-striped table-hover origen">*

*<thead>*

*<tr>*

*<th class="text-center" style="width: 50px;vertical-align:middle;">#</th>*

*<th class="text-center" style="vertical-align:middle;">Imagen</th>*

*<th class="text-center" style="vertical-align:middle;">Descripción</th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Categoría </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Cantidad </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Precio de compra </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Precio de venta </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Agregado </th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($productos as $producto):?>*

*<tr>*

*<td class="text-center" style="vertical-align:middle;"><?php echo contar\_ids();?></td>*

*<td>*

*<?php if($producto['id\_media'] === NULL): ?>*

*<img class="img-avatar img-circle" src="cargas/productos/no\_image.jpg" alt="">*

*<?php else: ?>*

*<img class="img-avatar img-circle" src="cargas/productos/<?php echo $producto['imagen']; ?>" alt="">*

*<?php endif; ?>*

*</td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['nombre']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['categorie']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['cantidad']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['precio\_compra']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['precio\_venta']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo leer\_fecha($producto['fecha\_actualizacion']); ?></td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*<form method="POST" id="formulario-producto">*

*<div id="resultadoBusqueda-productos"></div>*

*<div id="resultados-productos"></div>*

*<h3 style="padding:10px" class="bg-primary text-center pad-basic no-btm">Agregar productos</h3>*

*<div id="contenedor-producto">*

*<div id="contenedor-producto-0" class="form-producto">*

*<div class="form-group titulo-es">*

*<div class="input-group clase-titulo">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-th-large"></i>*

*</span>*

*<input autofocus="" type="text" class="form-control titulo" name="titulo-producto-0" placeholder="Descripción" required="">*

*</div>*

*</div>*

*<div class="row">*

*<div class="form-group col-md-4">*

*<select class="form-control categoria" name="categoria-0" required="">*

*<option value="">Selecciona una categoría</option>*

*<?php foreach ($todas\_categorias as $categoria): ?>*

*<?php if($categoria['nombre'] != "No hay"): ?>*

*<option value="<?php echo (int)$categoria['id'] ?>">*

*<?php echo $categoria['nombre'] ?></option>*

*<?php endif; ?>*

*<?php endforeach; ?>*

*</select>*

*</div>*

*<div class="form-group col-md-4">*

*<select class="form-control foto" name="product-photo-0" required="">*

*<option value=""> Selecciona una imágen </option>*

*<?php foreach ($todas\_fotos as $photo): ?>*

*<?php if($photo['nombre\_archivo'] != "no\_image.jpg"): ?>*

*<option value="<?php echo (int)$photo['id'];?>" >*

*<?php echo $photo['nombre\_archivo'];?></option>*

*<?php endif; ?>*

*<?php endforeach; ?>*

*</select>*

*</div>*

*</div>*

*<div class="row">*

*<div class="form-group col-md-5">*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-shopping-cart"></i>*

*</span>*

*<input type="number" min="1" class="form-control cantidad" name="cantidad-0" placeholder="Cantidad" required="">*

*</div>*

*</div>*

*<div class="form-group col-md-4">*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-usd"></i>*

*</span>*

*<input type="number" min="1" class="form-control compra" name="compra-0" placeholder="Precio de compra" required="">*

*</div>*

*</div>*

*<span style="padding-bottom:10px;margin-bottom:15px" class="venta-compra col-xs-12 input-group-addon col-md-1">*

*<i>.</i>*

*</span>*

*<div class="form-group col-md-2">*

*<input type="number" min="0" max="99" class="form-control compra\_decimal" name="compra\_decimal-0" placeholder="Decimales" required="" value="0">*

*</div>*

*<div class="form-group col-md-4">*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-usd"></i>*

*</span>*

*<input type="number" min="1" class="form-control precio" name="precio-0" placeholder="Precio de venta" required="">*

*</div>*

*</div>*

*<span style="padding-bottom:10px;margin-bottom:15px" class="venta-decimal col-xs-12 form-group input-group-addon col-md-1">*

*<i>.</i>*

*</span>*

*<div class="form-group col-md-2">*

*<input type="number" min="0" max="99" class="form-control precio\_decimal" name="precio\_decimal-0" placeholder="Decimales" required="" value="0">*

*</div>*

*<div class="col-xs-2" style="padding-top:3px">*

*<button class="boton-menos-producto btn-info" id="0" disabled="" type="button" >Menos -</button>*

*</div>*

*</div>*

*<hr>*

*</div>*

*</div>*

*<div style="padding-top:10px" class="form-group btn-der">*

*<input type="submit" id="agregarProducto" name="insertar-producto" value="Insertar producto(s)" class="btn btn-info"/>*

*<button id="adicional-producto" name="adicional-producto" type="button" class="btn btn-warning"> Más + </button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="libs/js/multiples\_registros\_categorias.js"></script>*

*<script src="libs/js/multiples\_registros\_productos.js"></script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 37 se muestra el resultado de la codificación del código de arriba.

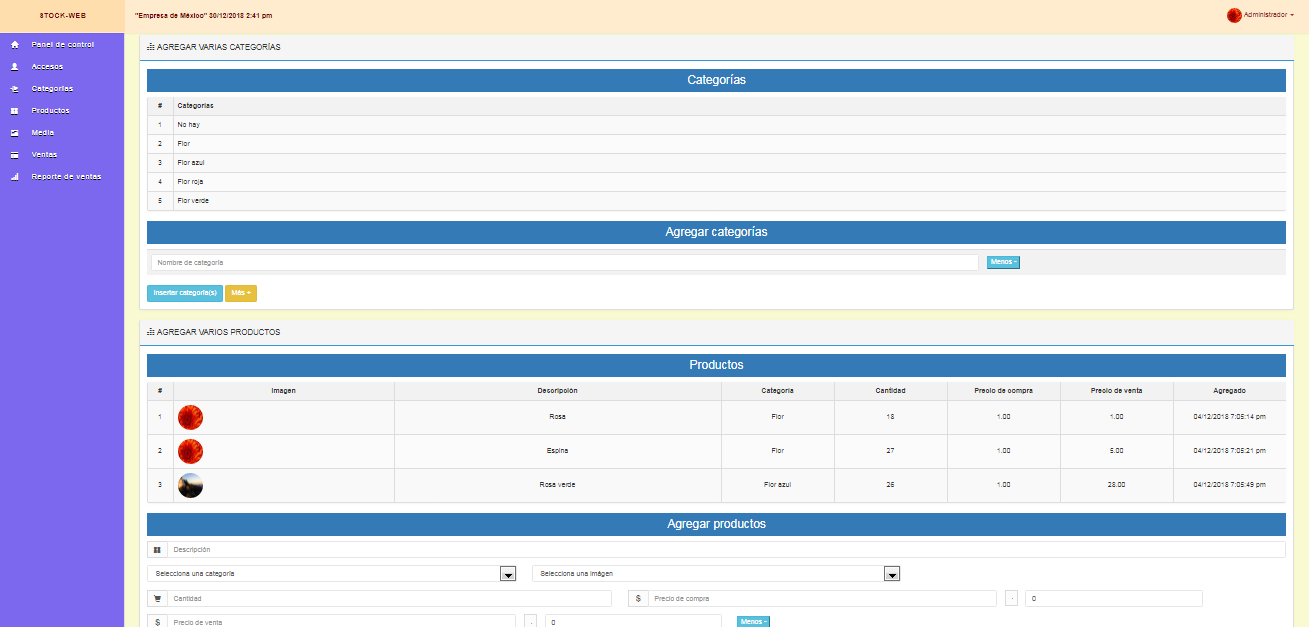


Ilustración 37 Sección para agregar múltiples registros

## Sección para agregar producto

En esta vista se observa el formulario para agregar un nuevo producto.

El formulario cuenta con campos para agregar el nombre de producto, la categoría del producto, la imagen del producto, la cantidad de producto existente, el precio de compra y el precio de venta.

Al ingresar el nombre de producto, no debe existir otro con nombre igual.

Al dar clic en el botón de agregar producto, deberá agregar el producto.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Agregar producto';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$todas\_categorias = encontrar\_todos('categoria');*

*$todas\_fotos = encontrar\_todos('media');*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Agregar producto</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<div class="col-md-12">*

*<div id="resultado"></div>*

*<div id="valor"></div>*

*<form method="POST" id="formulario" class="clearfix">*

*<div class="form-group">*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-th-large"></i>*

*</span>*

*<input id="titulo" autofocus="" type="text" class="form-control" name="titulo" placeholder="Descripción" required="">*

*</div>*

*</div>*

*<div class="">*

*<div class="row">*

*<div class="form-group col-md-4">*

*<select id="categoria" class="form-control" name="categoria" required="">*

*<option value="">Selecciona una categoría</option>*

*<?php foreach ($todas\_categorias as $categoria): ?>*

*<?php if($categoria['nombre'] != "No hay"): ?>*

*<option value="<?php echo (int)$categoria['id'] ?>">*

*<?php echo $categoria['nombre'] ?></option>*

*<?php endif; ?>*

*<?php endforeach; ?>*

*</select>*

*</div>*

*<div class="form-group col-md-4">*

*<select id="foto" class="form-control" name="foto" required="">*

*<option value=""> Selecciona una imágen </option>*

*<?php foreach ($todas\_fotos as $foto): ?>*

*<?php if($foto['nombre\_archivo'] != "no\_image.jpg"): ?>*

*<option value="<?php echo (int)$foto['id'];?>" >*

*<?php echo $foto['nombre\_archivo'] ?></option>*

*<?php endif; ?>*

*<?php endforeach; ?>*

*</select>*

*</div>*

*</div>*

*</div>*

*<div class="">*

*<div class="row">*

*<div class="col-md-5 form-group">*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-shopping-cart"></i>*

*</span>*

*<input id="cantidad" type="number" min="1" class="form-control" name="cantidad" placeholder="Cantidad" required="">*

*</div>*

*</div>*

*<div class="col-md-4 form-group">*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-usd"></i>*

*</span>*

*<input id="compra" type="number" min="1" class="form-control" name="compra" placeholder="Precio de compra" required="">*

*</div>*

*</div>*

*<span style="padding-bottom:10px" class="venta-compra col-xs-12 form-group input-group-addon col-md-1">*

*<i>.</i>*

*</span>*

*<div class="col-md-2 form-group">*

*<input id="compra\_decimal" type="number" min="0" max="99" class="form-control" name="compra\_decimal" placeholder="Decimales" required="" value="0">*

*</div>*

*<div class="col-md-4 form-group">*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-usd"></i>*

*</span>*

*<input id="precio" type="number" min="1" class="form-control" name="precio" placeholder="Precio de venta" required="">*

*</div>*

*</div>*

*<span style="padding-bottom:10px" class="venta-decimal col-xs-12 form-group input-group-addon col-md-1">*

*<i>.</i>*

*</span>*

*<div class="form-group col-md-2">*

*<input id="precio\_decimal" type="number" min="0" max="99" class="form-control" name="precio\_decimal" placeholder="Decimales" required="" value="0">*

*</div>*

*</div>*

*</div>*

*<button type="submit" name="agregar\_producto" class="btn btn-danger">Agregar producto</button>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="libs/js/agregar\_producto.js"></script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 38 se muestra el resultado de la codificación del código de arriba.

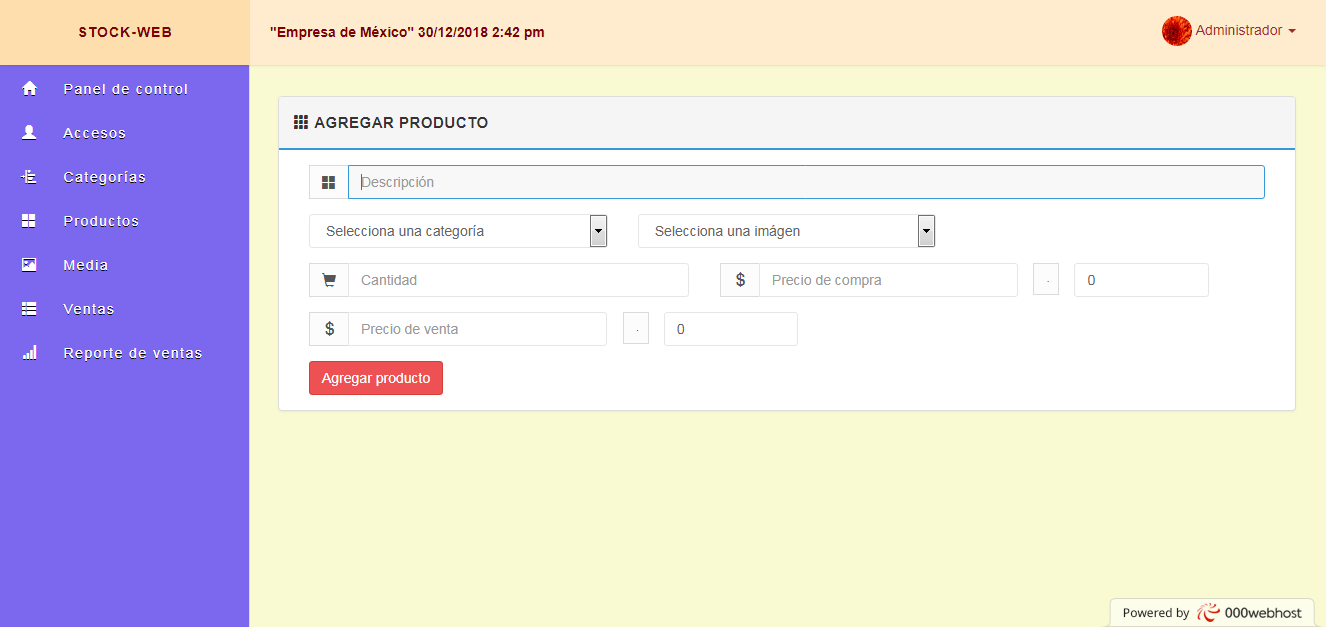


Ilustración 38 Sección para agregar productos

## Sección para agregar usuario

En esta vista se observa el formulario para agregar un nuevo usuario.

El formulario cuenta con campos para agregar el nombre completo de la persona dueña de la cuenta, el nombre de usuario, la contraseña y el rol de usuario.

Al ingresar el nombre de usuario, no debe existir otro con nombre igual.

Al ingresar el rol de usuario, este podrá tener como valor administrador, especial o estándar.

Al dar clic en el botón de agregar usuario, deberá agregar el usuario.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Agregar usuarios';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$grupos = encontrar\_todos('grupo');*

*include\_once('disenos/cabecera.php');*

*echo mostrar\_mensaje($mensaje);*

*?>*

*<div class="">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Agregar usuario</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<div class="col-md-12">*

*<div id="resultado"></div>*

*<div id="valor"></div>*

*<form method="POST" id="formulario">*

*<div class="form-group">*

*<label for="nombre">Nombre</label>*

*<input id="nombre" autofocus="" type="text" class="form-control" name="nombre" placeholder="Nombre completo" required>*

*</div>*

*<div class="form-group">*

*<label for="usuario">Usuario</label>*

*<input id="usuario" required="" type="text" class="form-control" name="usuario" placeholder="Nombre de usuario">*

*</div>*

*<div class="form-group">*

*<label for="password">Contraseña</label>*

*<input id="password" required="" type="password" class="form-control" name="password" placeholder="Contraseña">*

*</div>*

*<div class="form-group">*

*<label for="rol">Rol de usuario</label>*

*<select id="rol" class="form-control" name="rol" required="">*

*<?php foreach ($grupos as $grupo ):?>*

*<option value="<?php echo $grupo['nivel'];?>"><?php echo ucwords($grupo['nombre']);?></option>*

*<?php endforeach;?>*

*</select>*

*</div>*

*<div class="form-group clearfix">*

*<button type="submit" name="agregar\_usuario" class="btn btn-primary">Guardar</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="libs/js/agregar\_usuario.js"></script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 39 se muestra el resultado de la codificación del código de arriba.



Ilustración 39 Sección para agregar usuario

## Sección para agregar venta

En esta vista se observa el formulario para agregar una nueva venta.

El formulario cuenta con un campo para ingresar el nombre del producto a vender.

Al ingresar el nombre del producto se generará un formulario dinámicamente para ingresar la cantidad a vender y la fecha en que se efectuará la venta; esto por medio del archivo *“ajax.php”.*

Al dar clic en el botón de agregar del formulario dinámico este efectuará la venta.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Agregar venta';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$usuario = usuario\_actual();*

*if(isset($\_POST['add\_sale'])) {*

*$campos\_requeridos = array('s\_id','quantity','price','total', 'date' );*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)) {*

*$p\_id = $db->escape((int)$\_POST['s\_id']);*

*$s\_qty = $db->escape((int)$\_POST['quantity']);*

*$s\_total = $db->escape($\_POST['total']);*

*$date = $db->escape($\_POST['date']);*

*$s\_date = crear\_fecha();*

*$sql = "INSERT INTO venta (";*

*$sql .= " id\_usuario,id\_producto,cantidad,precio,fecha";*

*$sql .= ") VALUES (";*

*$sql .= "{$usuario['id']},'{$p\_id}','{$s\_qty}','{$s\_total}','{$date}'";*

*$sql .= ")";*

*if($db->query($sql)) {*

*actualizar\_cantidad\_producto($s\_qty,$p\_id);*

*$sesion->mensaje('s',"Venta agregada. ");*

*redirigir('ventas.php', false);*

*} else {*

*$sesion->mensaje('d','Lo siento el registro falló.');*

*redirigir('agregar\_venta.php', false);*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('agregar\_venta.php',false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-6">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*<form method="post" action="ajax.php" autocomplete="off" id="sug-form">*

*<div class="form-group">*

*<div class="input-group">*

*<span class="input-group-btn">*

*<button type="submit" class="btn btn-primary">Búsqueda</button>*

*</span>*

*<input autofocus="" type="text" id="sug\_input" class="form-control" name="title" placeholder="Buscar por el nombre del producto">*

*</div>*

*<div id="result" class="list-group"></div>*

*</div>*

*</form>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Agregar venta</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<form method="post" action="agregar\_venta.php">*

*<table class="table table-bordered">*

*<thead>*

*<th class="text-center"> Producto </th>*

*<th class="text-center"> Precio </th>*

*<th class="text-center"> Cantidad </th>*

*<th class="text-center"> Total </th>*

*<th class="text-center"> Agregado</th>*

*<th class="text-center"> Acciones</th>*

*</thead>*

*<tbody id="product\_info"> </tbody>*

*</table>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 40 se muestra el resultado de la codificación del código de arriba.

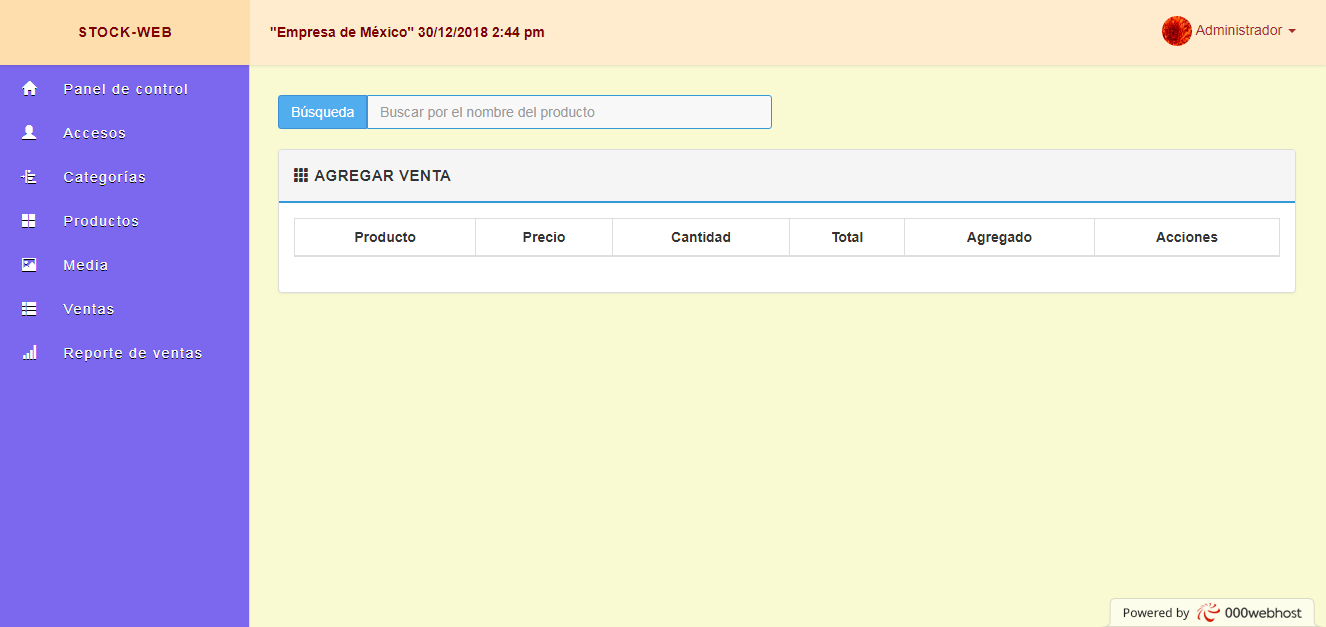


Ilustración 40 Sección para agregar venta

## Archivo para mandar datos de productos por medio de AJAX

En este archivo “*ajax.php”* se muestran una lista desplegable de los productos que más se ajusten a la descripción que recibe por medio de POST. Primero se incluye el archivo *“cargar.php”* en el directorio *“incluye”*. Se checa si hay una sesión activa o se direcciona a la página principal. Primero se checa si el nombre de producto a buscar fue recibido por medio de POST. Se procede a crear una variable *“$products“* que tendrá como valor lo que devuelva la función *“encontrar\_producto\_por\_titulo“* que recibirá como parámetro lo recibido por POST. Se procede a checar si *“$products“* tiene algún valor, si hay coincidencia muestra todas las coincidencias en forma de lista y caso contrario mostrará el mensaje “Producto no encontrado” en forma de lista. El usuario selecciona la opción que buscaba y le da clic, al hacerlo se muestra un formulario; puede haber excepciones como que el producto no cuenta con suficiente inventario. Finalmente se codifica en formato HTML por medio de JSON.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*if (!$sesion->estaLogeado(true)) { redirigir('index.php', false);}*

*$html = '';*

*if(isset($\_POST['product\_name']) && strlen($\_POST['product\_name'])) {*

*$products = encontrar\_producto\_por\_titulo($\_POST['product\_name']);*

*if($products) {*

*foreach ($products as $producto):*

*$html .= "<li class=\"list-group-item\">";*

*$html .= $producto['nombre'];*

*$html .= "</li>";*

*endforeach;*

*} else {*

*$html .= '<li onClick=\"fill(\''.addslashes().'\')\" class=\"list-group-item\">';*

*$html .= 'No encontrado';*

*$html .= "</li>";*

*}*

*echo json\_encode($html);*

*}*

*if(isset($\_POST['p\_name']) && strlen($\_POST['p\_name'])) {*

*$product\_title = remover\_basura($db->escape($\_POST['p\_name']));*

*$results = encontrar\_todos\_producto\_por\_titulo($product\_title);*

*foreach ($results as $resultado) {*

*$cantidad = $resultado['cantidad'];*

*}*

*if($results && $cantidad > 0) {*

*foreach ($results as $resultado) {*

*$html .= "<tr>";*

*$html .= "<td id=\"s\_name\">".$resultado['nombre']."</td>";*

*$html .= "<input type=\"hidden\" name=\"s\_id\" value=\"{$resultado['id']}\">";*

*$html .= "<td>";*

*$html .= "<input type=\"hidden\" name=\"price\" value=\"{$resultado['precio\_venta']}\">";*

*$html .= "<input type=\"number\" disabled min='1' class=\"form-control\" name=\"price\" value=\"{$resultado['precio\_venta']}\">";*

*$html .= "</td>";*

*$html .= "<td id=\"s\_qty\">";*

*$html .= "<input type=\"number\" min='1' max=\"{$resultado['cantidad']}\" class=\"form-control\" name=\"quantity\" value=\"1\" required>";*

*$html .= "</td>";*

*$html .= "<td>";*

*$html .= "<input type=\"hidden\" name=\"total\" value=\"{$resultado['precio\_venta']}\">";*

*$html .= "<input type=\"text\" disabled class=\"form-control\" name=\"total\" value=\"{$resultado['precio\_venta']}\">";*

*$html .= "</td>";*

*$html .= "<td>";*

*$html .= "<input type=\"date\" class=\"form-control datePicker\" name=\"date\" data-date data-date-format=\"yyyy-mm-dd\" required>";*

*$html .= "</td>";*

*$html .= "<td>";*

*$html .= "<button type=\"submit\" name=\"add\_sale\" class=\"btn btn-primary\">Agregar</button>";*

*$html .= "</td>";*

*$html .= "</tr>";*

*}*

*} elseif($results && $cantidad == 0) {*

*$html ='<tr><td>El producto no cuenta con suficiente inventario.</td></tr>';*

*} else {*

*$html ='<tr><td>El producto no se encuentra registrado en la base de datos.</td></tr>';*

*}*

*echo json\_encode($html);*

*}*

## Archivo para autenticar a los usuarios que inicien sesión

En este archivo *“autenticación.php”* se autentica a los usuarios que desean acceder al sistema. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se crea un arreglo *“$campos\_requeridos“* donde se guardarán el usuario y la contraseña, se procede a validar los campos por medio de la función *“validar\_campos“*. Se crea una variable *“$nombre\_usuario“* que tendrá como valor el nombre de usuario recibido por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se crea una variable *“$contrasena“* que tendrá como valor la contraseña del usuario recibido por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se checa si la variable *“$errors“* (que tiene cualquier error) está vacía. Se procede a guardar en la variable *“$id\_usuario “*lo que devuelva la función *“autenticar“*, donde esta recibe como parámetro las variables *“$nombre\_usuario“* y *“$contrasena“*. Se procede a checar el valor de *“$id\_usuario“*, si es mayor -1 se crea una sesión y se actualiza la última fecha de login, si es igual a -1 se muestra un mensaje “Usuario incorrecto”, si es igual a -2 se muestra un mensaje “Contraseña incorrecta”, si es igual a -3 se muestra un mensaje “Te encuentras cesado”, si es igual a -4 se muestra un mensaje “Tu grupo se encuentra cesado” y finalmente si es igual a otro valor se muestra un mensaje con el error correspondiente. Finalmente imprime el mensaje correspondiente.

A continuación, se muestra el código.

*<?php*

*include\_once('incluye/cargar.php');*

*$campos\_requeridos = array('usuario','password' );*

*validar\_campos($campos\_requeridos);*

*$nombre\_usuario = remover\_basura($\_POST['usuario']);*

*$contrasena = remover\_basura($\_POST['password']);*

*if(empty($errores)) {*

*$id\_usuario = autenticar($nombre\_usuario, $contrasena);*

*if($id\_usuario > -1) {*

*$sesion->login($id\_usuario);*

*actualizarUltimoLogin($id\_usuario);*

*$sesion->mensaje("s", "Bienvenido al inventario.");*

*} elseif($id\_usuario == -1) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Usuario incorrecto.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*} elseif($id\_usuario == -2) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Contraseña incorrecta.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*} elseif($id\_usuario == -3) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Te encuentras cesado.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*} elseif($id\_usuario == -4) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Tu grupo se encuentra cesado.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*}*

*} else {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>$errores.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*}*

*$requisito = "<input type='hidden' id='requisito' value='$id\_usuario' />";*

*echo $mensaje;*

*echo $requisito;*

## Archivo para borrar una categoría

En este archivo *“borrar\_categoria.php”* se borra la categoría correspondiente por medio de su identificador. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se establece el nivel de usuario para este archivo *“pagina\_requiere\_nivel“*; en este caso es de estándar a administrador. Se procede a crear una variable *“$categoría“* que tendrá todos los datos de la categoría por medio de su identificador pasado a este archivo por medio de GET; esto se realiza por medio de la función *“encontrar\_por\_id“* donde esta recibe el nombre de la tabla categoría y el identificador de la categoría. Se procede a checar si *“$categoria“* tiene algún valor, si no es el caso muestra un mensaje “Falta id de categoría” por medio del objeto *“sesión“* por medio de su método *“mensaje“* y finalmente redirige a la sección de categorías por medio de la función *“redirigir“*. Si *“$categoria“* tiene un valor se crea una variable *“borrar\_id“* que guardará el resultado de intentar borrar la categoría por medio de *“borrar\_por\_id“*. Finalmente se checa el valor de *“borrar\_id“*, si es verdadero se muestra el mensaje “Categoría eliminada” y se redirige a la sección de categorías por medio de las funciones *“mensaje “*y *“redirigir; si es falso “*se muestra el mensaje “La eliminación falló” y se redirige a la sección de categorías por medio de las funciones *“mensaje “*y *“redirigir“*.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$categoria = encontrar\_por\_id('categoria',(int)$\_GET['id']);*

*if(!$categoria) {*

*$sesion->mensaje("d","Falta id de categoría.");*

*redirigir('categoria.php');*

*}*

*$borrar\_id = borrar\_por\_id('categoria',(int)$categoria['id']);*

*if($borrar\_id) {*

*$sesion->mensaje("s","Categoría eliminada.");*

*redirigir('categoria.php');*

*} else {*

*$sesion->mensaje("d","La eliminación falló.");*

*redirigir('categoria.php');*

*}*

## Archivo para borrar un grupo

En este archivo *“borrar\_grupo.php”* se borra el grupo correspondiente por medio de su identificador. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se establece el nivel de usuario para este archivo *“pagina\_requiere\_nivel“*; en este caso es de estándar a administrador. Se procede a crear una variable *“$borrar\_id“* que tendrá todos los datos del grupo por medio de su identificador pasado a este archivo por medio de GET; esto se realiza por medio de la función *“encontrar\_por\_id“* donde esta recibe el nombre de la tabla grupo y el identificador del grupo. Se procede a checar si *“$borrar\_id“* tiene algún valor, si no es el caso muestra un mensaje “Falta id de grupo” por medio del objeto *“sesión“* por medio de su método *“mensaje“* y finalmente redirige a la sección de grupos por medio de la función *“redirigir“*. Si *“$borrar\_id“* tiene un valor se crea una variable *“borrar\_id“* que guardará el resultado de intentar borrar el grupo por medio de *“borrar\_por\_id“*. Finalmente se checa el valor de *“borrar\_id“*, si es verdadero se muestra el mensaje “Grupo eliminado” y se redirige a la sección de grupos por medio de las funciones *“mensaje “*y *“redirigir; si es falso “*se muestra el mensaje “La eliminación falló” y se redirige a la sección de grupos por medio de las funciones *“mensaje “*y *“redirigir“*.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$borrar\_id = borrar\_por\_id('grupo',(int)$\_GET['id']);*

*if($borrar\_id) {*

*$sesion->mensaje("s","Grupo eliminado.");*

*redirigir('grupo.php');*

*} else {*

*$sesion->mensaje("d","La eliminación falló.");*

*redirigir('grupo.php');*

*}*

## Archivo para borrar una imagen

En este archivo *“borrar\_media.php”* se borra la imagen correspondiente por medio de su identificador. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se establece el nivel de usuario para este archivo *“pagina\_requiere\_nivel“*; en este caso es de estándar a administrador. Se procede a crear una variable *“$find\_media “* que tendrá todos los datos de la imagen por medio de su identificador pasado a este archivo por medio de GET; esto se realiza por medio de la función *“encontrar\_por\_id“* donde esta recibe el nombre de la tabla media y el identificador de la imagen. Se procede a checar si *“$find\_media “* tiene algún valor, si no es el caso muestra un mensaje “Falta id de imagen” por medio del objeto *“sesión“* por medio de su método *“mensaje“* y finalmente redirige a la sección de media por medio de la función *“redirigir“*. Si *“$borrar\_id“* tiene un valor se crea una variable *“borrar\_id“* que guardará el resultado de intentar borrar la imagen por medio de *“borrar\_por\_id“*. Finalmente se checa el valor de *“borrar\_id“*, si es verdadero se muestra el mensaje “Imagen eliminada” y se redirige a la sección de media por medio de las funciones *“mensaje “*y *“redirigir; si es falso “*se muestra el mensaje “La eliminación falló” y se redirige a la sección de media por medio de las funciones *“mensaje “*y *“redirigir“*.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$find\_media = encontrar\_por\_id('media',(int)$\_GET['id']);*

*$photo = new Media();*

*if($photo->destruir\_media($find\_media['id'],$find\_media['nombre\_archivo'])) {*

*$sesion->mensaje("s","Se ha eliminado la foto.");*

*redirigir('media.php');*

*} else {*

*$sesion->mensaje("d","Se ha producido un error en la eliminación de fotografías.");*

*redirigir('media.php');*

*}*

## Archivo para borrar un usuario

En este archivo *“borrar\_usuario.php”* se borra el usuario correspondiente por medio de su identificador. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se establece el nivel de usuario para este archivo *“pagina\_requiere\_nivel“*; en este caso es de estándar a administrador. Se procede a crear una variable *“$borrar\_id“* que tendrá todos los datos del usuario por medio de su identificador pasado a este archivo por medio de GET; esto se realiza por medio de la función *“encontrar\_por\_id“* donde esta recibe el nombre de la tabla usuario y el identificador del usuario. Se procede a checar si *“$borrar\_id“* tiene algún valor, si no es el caso muestra un mensaje “Falta id de usuario” por medio del objeto *“sesión“* por medio de su método *“mensaje“* y finalmente redirige a la sección de usuarios por medio de la función *“redirigir“*. Si *“$borrar\_id“* tiene un valor se crea una variable *“borrar\_id“* que guardará el resultado de intentar borrar el usuario por medio de *“borrar\_por\_id“*. Finalmente se checa el valor de *“borrar\_id“*, si es verdadero se muestra el mensaje “Usuario eliminado” y se redirige a la sección de grupos por medio de las funciones *“mensaje “*y *“redirigir; si es falso “*se muestra el mensaje “La eliminación falló” y se redirige a la sección de usuarios por medio de las funciones *“mensaje “*y *“redirigir“*.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$borrar\_id = borrar\_por\_id('usuario',(int)$\_GET['id']);*

*if($borrar\_id) {*

*$sesion->mensaje("s","Usuario eliminado.");*

*redirigir('usuarios.php');*

*} else {*

*$sesion->mensaje("d","Se ha producido un error en la eliminación del usuario.");*

*redirigir('usuarios.php');*

*}*

## Archivo para borrar una venta

En este archivo *“borrar\_venta.php”* se borra la venta correspondiente por medio de su identificador. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se establece el nivel de usuario para este archivo *“pagina\_requiere\_nivel“*; en este caso es de estándar a administrador. Se procede a crear una variable *“$borrar\_venta “* que tendrá todos los datos de el grupo por medio de su identificador pasado a este archivo por medio de GET; esto se realiza por medio de la función *“encontrar\_por\_id“* donde esta recibe el nombre de la tabla grupo y el identificador del grupo. Se procede a checar si *“$borrar\_venta “* tiene algún valor, si no es el caso muestra un mensaje “Falta id de grupo” por medio del objeto *“sesión“* por medio de su método *“mensaje“* y finalmente redirige a la sección de grupos por medio de la función *“redirigir“*. Si *“$borrar\_venta “* tiene un valor se crea una variable *“borrar\_id“* que guardará el resultado de intentar borrar el grupo por medio de *“borrar\_por\_id“*. Finalmente se checa el valor de *“borrar\_id“*, si es verdadero se muestra el mensaje “Grupo eliminado” y se redirige a la sección de grupos por medio de las funciones *“mensaje “*y *“redirigir; si es falso “*se muestra el mensaje “La eliminación falló” y se redirige a la sección de grupos por medio de las funciones *“mensaje “*y *“redirigir“*.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$borrar\_venta = encontrar\_por\_id('venta',(int)$\_GET['id']);*

*if(!$borrar\_venta) {*

*$sesion->mensaje("d","ID vacío.");*

*redirigir('ventas.php');*

*}*

*$borrar\_id = borrar\_por\_id('venta',(int)$borrar\_venta['id']);*

*if($borrar\_id) {*

*$sesion->mensaje("s","Venta eliminada.");*

*redirigir('ventas.php');*

*} else {*

*$sesion->mensaje("d","La eliminación falló.");*

*redirigir('ventas.php');*

*}*

## Archivo para buscar una categoría

En este archivo “*buscar\_categoria.php”* se busca una categoría para saber si ya se registró previamente. Primero se incluye el archivo *“cargar.php”* en el directorio *“incluye”*. Se checa si hay una sesión activa o se direcciona a la página principal. Primero se checa si el nombre de la categoría a buscar fue recibido por medio de POST. Se procede a crear una variable *“$categorias“* que tendrá como valor lo que devuelva la función *“encontrar\_categorias\_por\_titulo“* que recibirá como parámetro lo recibido por POST. Se procede a checar si *“$categorias“* tiene algún valor, si hay coincidencia muestra todas las coincidencias en forma de lista y caso contrario mostrará el mensaje “Categoría no encontrado” en forma de lista. El usuario selecciona la opción que buscaba y le da clic, al hacerlo se muestra un formulario; puede haber excepciones como que el producto no cuenta con suficiente inventario. Finalmente se codifica en formato HTML por medio de JSON.

A continuación, se muestra el código.

*<?php*

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

*require\_once('incluye/cargar.php');*

*$categorias = $\_REQUEST['categorias'];*

*$caracteres\_malos = array("<", ">", "\"", "'", "/", "<", ">", "'", "/");*

*$caracteres\_buenos = array("& lt;", "& gt;", "& quot;", "& #x27;", "& #x2F;", "& #060;", "& #062;", "& #039;", "& #047;");*

*$numero = sizeof($categorias);*

*$mensaje = "";*

*$categorias\_error = "";*

*$indicador = FALSE;*

*$indicadores = TRUE;*

*$resultado = array($numero);*

*$datos\_categorias = "";*

*$categorias\_recientes;*

*$numero\_recientes = -1;*

*if(isset($\_POST['categorias\_recientes'])) {*

*$categorias\_recientes = $\_POST['categorias\_recientes'];*

*}*

*if(isset($\_POST['contador\_tabla\_recientes'])) {*

*$numero\_recientes = $\_POST['contador\_tabla\_recientes'];*

*}*

*if(isset($categorias)) {*

*for($contador = 0; $contador < $numero; $contador++) {*

*$categoria\_filtrada = $categorias[$contador];*

*$categoria\_filtrada = str\_replace($caracteres\_malos, $caracteres\_buenos, $categoria\_filtrada);*

*$categoria\_filtrada = trim($categoria\_filtrada);*

*$consulta\_categoria = encontrar\_nombre\_categoria($categoria\_filtrada);*

*foreach ($consulta\_categoria as $nombre) {*

*$resultado[$contador] = $nombre['nombre'];*

*if(isset($resultado[$contador])) {*

*$categorias\_error = $categorias\_error . $resultado[$contador] . ", ";*

*$indicador = TRUE;*

*}*

*}*

*}*

*if($indicador && $numero == 1) {*

*$contar\_id = $\_POST['contador'];*

*$categorias\_error = substr($categorias\_error, 0, -2);*

*$mensaje = "<div class='alert alert-warning'><span style='padding-right:5px'>La siguiente categoría ya está utilizada:" . " " . $categorias\_error . ".</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*if($numero\_recientes != -1) {*

*$datos\_categorias = "<h3 style='padding:10px' class='bg-primary text-center pad-basic no-btm'>Categorías recien agregadas</h3><div class='panel panel-default'><table id='tabla\_categorias\_recientes' class='table table-bordered table-striped table-hover'><thead><tr><th class='text-center' style='width: 50px;'>#</th><th>Categorías</th></tr></thead><tbody>";*

*for($j = 0; $j < $numero\_recientes; $j++) {*

*$c = $j + $contar\_id;*

*$datos\_categorias .= "<tr><td class='text-center'>$c</td><td>$categorias\_recientes[$j]</td></tr>";*

*}*

*$datos\_categorias .= "</tbody></table></div>";*

*}*

*} else if($indicador && $numero != 1) {*

*$contar\_id = $\_POST['contador'];*

*$categorias\_error = substr($categorias\_error, 0, -2);*

*$mensaje = "<div class='alert alert-warning'><span style='padding-right:5px'>Las siguientes categorías ya están utilizadas:" . " " . $categorias\_error . ".</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*if($numero\_recientes != -1) {*

*$datos\_categorias = "<h3 style='padding:10px' class='bg-primary text-center pad-basic no-btm'>Categorías recien agregadas</h3><div class='panel panel-default'><table id='tabla\_categorias\_recientes' class='table table-bordered table-striped table-hover'><thead><tr><th class='text-center' style='width: 50px;'>#</th><th>Categorías</th></tr></thead><tbody>";*

*for($j = 0; $j < $numero\_recientes; $j++) {*

*$c = $j + $contar\_id;*

*$datos\_categorias .= "<tr><td class='text-center'>$c</td><td>$categorias\_recientes[$j]</td></tr>";*

*}*

*$datos\_categorias .= "</tbody></table></div>";*

*}*

*} else if(!$indicador) {*

*$nombre\_categoria = array($numero);*

*for($contador = 0; $contador < $numero; $contador++) {*

*$nombre\_categoria[$contador] = remover\_basura($db->escape($categorias[$contador]));*

*$sql = "INSERT INTO categoria (nombre)";*

*$sql .= " VALUES ('{$nombre\_categoria[$contador]}')";*

*if($db->query($sql)){*

*$indicadores = FALSE;*

*} else {*

*$indicadores = TRUE;*

*}*

*}*

*if(!$indicadores) {*

*$mensaje = "<div class='alert alert-success'><span style='padding-right:5px'>Los registros fueron agregados exitosamente.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$contar\_id = $\_POST['contador'];*

*$datos\_categorias = "<h3 style='padding:10px' class='bg-primary text-center pad-basic no-btm'>Categorías recien agregadas</h3><div class='panel panel-default'><table id='tabla\_categorias\_recientes' class='table table-bordered table-striped table-hover'><thead><tr><th class='text-center' style='width: 50px;'>#</th><th>Categorías</th></tr></thead><tbody>";*

*if($numero\_recientes != -1) {*

*for($j = 0; $j < $numero\_recientes; $j++) {*

*$c = $j + $contar\_id;*

*$datos\_categorias .= "<tr><td class='text-center'>$c</td><td>$categorias\_recientes[$j]</td></tr>";*

*}*

*$numero += $numero\_recientes;*

*for($counter = $numero\_recientes; $counter < $numero; $counter++) {*

*$total = $counter + $contar\_id;*

*$indice = $counter - $numero\_recientes;*

*$datos\_categorias .= "<tr><td class='text-center'>$total</td><td>$nombre\_categoria[$indice]</td></tr>";*

*}*

*} else {*

*for($counter = 0; $counter < $numero; $counter++) {*

*$total = $counter + $contar\_id;*

*$datos\_categorias .= "<tr><td class='text-center'>$total</td><td>$nombre\_categoria[$counter]</td></tr>";*

*}*

*}*

*$datos\_categorias .= "</tbody></table></div>";*

*} else {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>La inserción de registros falló.:" . " " . $categorias\_error . "</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*}*

*}*

*}*

*echo $mensaje;*

*echo $datos\_categorias;*

## Archivo para buscar un producto

En este archivo *“buscar\_producto.php”* se busca un producto para saber si ya se registró previamente. Primero se incluye el archivo *“cargar.php”* en el directorio *“incluye”*. Se checa si hay una sesión activa o se direcciona a la página principal. Primero se checa si el nombre de producto a buscar fue recibido por medio de POST. Se procede a crear una variable *“$productos“* que tendrá como valor lo que devuelva la función *“encontrar\_producto\_por\_titulo“* que recibirá como parámetro lo recibido por POST. Se procede a checar si *“$productos“* tiene algún valor, si hay coincidencia muestra todas las coincidencias en forma de lista y caso contrario mostrará el mensaje “Producto no encontrado” en forma de lista. El usuario selecciona la opción que buscaba y le da clic, al hacerlo se muestra un formulario; puede haber excepciones como que el producto no cuenta con suficiente inventario. Finalmente se codifica en formato HTML por medio de JSON.

A continuación, se muestra el código.

*<?php*

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

*require\_once('incluye/cargar.php');*

*$titulo\_productos = $\_REQUEST['titulo\_productos'];*

*$categoria\_productos = $\_REQUEST['categoria\_productos'];*

*$imagen\_productos = $\_REQUEST['imagen\_productos'];*

*$cantidad\_productos = $\_REQUEST['cantidad\_productos'];*

*$compra\_productos = $\_REQUEST['compra\_productos'];*

*$compra\_decimal\_productos = $\_REQUEST['compra\_decimal\_productos'];*

*$precio\_productos = $\_REQUEST['precio\_productos'];*

*$precio\_decimal\_productos = $\_REQUEST['precio\_decimal\_productos'];*

*$numero = $\_REQUEST['contador\_producto'];*

*$contador\_tabla = $\_REQUEST['contador\_tabla'];*

*$nombre\_imagen\_productos = $\_REQUEST['nombre\_imagen\_productos'];*

*$nombre\_categoria\_productos = $\_REQUEST['nombre\_categoria\_productos'];*

*if(isset($nombre\_imagen\_productos)) {*

*for($i = 0; $i < $numero; $i++) {*

*$nombre\_imagen\_productos[$i] = trim($nombre\_imagen\_productos[$i]);*

*}*

*}*

*$caracteres\_malos = array("<", ">", "\"", "'", "/", "<", ">", "'", "/");*

*$caracteres\_buenos = array("& lt;", "& gt;", "& quot;", "& #x27;", "& #x2F;", "& #060;", "& #062;", "& #039;", "& #047;");*

*$mensaje = "";*

*$titulos\_productos\_error = "";*

*$indicador = FALSE;*

*$indicadores = TRUE;*

*$resultado = array($numero);*

*$datos\_productos = "";*

*$numero\_recientes = -1;*

*$arreglo\_productos\_agregados;*

*if(isset($\_REQUEST['contador\_tabla\_productos\_recientes'])) {*

*$numero\_recientes = $\_REQUEST['contador\_tabla\_productos\_recientes'];*

*if(isset($\_REQUEST['productos\_agregados'])) {*

*$productos\_agregados = $\_REQUEST['productos\_agregados'];*

*}*

*}*

*if(isset($titulo\_productos)) {*

*for($contador = 0; $contador < $numero; $contador++) {*

*$titulo\_producto\_filtrado = $titulo\_productos[$contador];*

*$titulo\_producto\_filtrado = str\_replace($caracteres\_malos, $caracteres\_buenos, $titulo\_producto\_filtrado);*

*$titulo\_producto\_filtrado = trim($titulo\_producto\_filtrado);*

*$consulta\_titulo\_producto = encontrar\_nombre\_producto($titulo\_producto\_filtrado);*

*foreach ($consulta\_titulo\_producto as $nombre) {*

*$resultado[$contador] = $nombre['nombre'];*

*if(isset($resultado[$contador])) {*

*$titulos\_productos\_error = $titulos\_productos\_error . $resultado[$contador] . ", ";*

*$indicador = TRUE;*

*}*

*}*

*}*

*if($indicador && $numero == 1) {*

*$contar\_id = $contador\_tabla;*

*$titulos\_productos\_error = substr($titulos\_productos\_error, 0, -2);*

*$mensaje = "<div class='alert alert-warning'><span style='padding-right:5px'>El siguiente nombre de producto ya existe:" . " " . $titulos\_productos\_error . ".</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*if($numero\_recientes != -1) {*

*$datos\_productos = "<h3 style='padding:10px' class='bg-primary text-center pad-basic no-btm'>Productos recien agregados</h3><div class='panel panel-default'><table class='tabla-resultado-producto table table-bordered table-striped table-hover'><thead><tr><th class='text-center' style='width: 50px;vertical-align:middle;'>#</th><th class='text-center' style='vertical-align:middle;'>Imagen</th><th class='text-center' style='vertical-align:middle;'>Descripción</th><th class='text-center' style='width: 10%;vertical-align:middle;'> Categoría </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Cantidad </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Precio de compra </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Precio de venta </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Agregado </th></thead><tbody>";*

*for($j = 0; $j < $numero\_recientes; $j++) {*

*$c = $j + $contar\_id;*

*$arreglo\_productos\_agregados = encontrar\_productos($productos\_agregados[$j]);*

*$datos\_productos .= "<tr><td class='text-center' style='vertical-align:middle;'>$c</td>";*

*$datos\_productos .= "<td><img class='img-avatar img-circle' src='cargas/productos/".$arreglo\_productos\_agregados['nombre\_archivo']."' alt=''></td>";*

*$datos\_productos .= "<td class='text-center necesario' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['nombre']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['categoria']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['cantidad']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['precio\_compra']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['precio\_venta']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['fecha\_actualizacion']."</td></tr>";*

*}*

*$datos\_productos .= "</tbody></table></div>";*

*}*

*} else if($indicador && $numero != 1) {*

*$contar\_id = $contador\_tabla;*

*$titulos\_productos\_error = substr($titulos\_productos\_error, 0, -2);*

*$mensaje = "<div class='alert alert-warning'><span style='padding-right:5px'>Los siguentes productos ya están creados:" . " " . $titulos\_productos\_error . ".</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*if($numero\_recientes != -1) {*

*$datos\_productos = "<h3 style='padding:10px' class='bg-primary text-center pad-basic no-btm'>Categorías recien agregadas</h3><div class='panel panel-default'><table class='tabla-resultado-producto table table-bordered table-striped table-hover'><thead><tr><th class='text-center' style='width: 50px;vertical-align:middle;'>#</th><th class='text-center' style='vertical-align:middle;'>Imagen</th><th class='text-center' style='vertical-align:middle;'>Descripción</th><th class='text-center' style='width: 10%;vertical-align:middle;'> Categoría </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Cantidad </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Precio de compra </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Precio de venta </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Agregado </th></thead><tbody>";*

*for($j = 0; $j < $numero\_recientes; $j++) {*

*$c = $j + $contar\_id;*

*$arreglo\_productos\_agregados = encontrar\_productos($productos\_agregados[$j]);*

*$datos\_productos .= "<tr><td class='text-center' style='vertical-align:middle;'>$c</td>";*

*$datos\_productos .= "<td><img class='img-avatar img-circle' src='cargas/productos/".$arreglo\_productos\_agregados['nombre\_archivo']."' alt=''></td>";*

*$datos\_productos .= "<td class='text-center necesario' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['nombre']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['categoria']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['cantidad']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['precio\_compra']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['precio\_venta']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['fecha\_actualizacion']."</td></tr>";*

*}*

*$datos\_productos .= "</tbody></table></div>";*

*}*

*} else if(!$indicador) {*

*$nombre\_producto = array($numero);*

*$cantidad\_producto = array($numero);*

*$categoria\_producto = array($numero);*

*$compra\_producto = array($numero);*

*$venta\_producto = array($numero);*

*$media\_id = array($numero);*

*$compra\_decimal\_producto = array($numero);*

*$venta\_decimal\_producto = array($numero);*

*for($contador = 0; $contador < $numero; $contador++) {*

*$nombre\_producto[$contador] = remover\_basura($db->escape($titulo\_productos[$contador]));*

*$cantidad\_producto[$contador] = remover\_basura($db->escape($cantidad\_productos[$contador]));*

*$compra\_producto[$contador] = remover\_basura($db->escape($compra\_productos[$contador]));*

*$venta\_producto[$contador] = remover\_basura($db->escape($precio\_productos[$contador]));*

*$categoria\_producto[$contador] = remover\_basura($db->escape($categoria\_productos[$contador]));*

*$media\_id[$contador] = remover\_basura($db->escape($imagen\_productos[$contador]));*

*$compra\_decimal\_producto[$contador] = remover\_basura($db->escape($compra\_decimal\_productos[$contador]));*

*$compra\_decimal\_producto[$contador] += 0.00;*

*$compra\_decimal\_producto[$contador] /= 100.00;*

*$compra\_producto[$contador] += $compra\_decimal\_producto[$contador];*

*$compra\_producto[$contador] += 0.00;*

*$venta\_decimal\_producto[$contador] = remover\_basura($db->escape($precio\_decimal\_productos[$contador]));*

*$venta\_decimal\_producto[$contador] += 0.00;*

*$venta\_decimal\_producto[$contador] /= 100.00;*

*$venta\_producto[$contador] += $venta\_decimal\_producto[$contador];*

*$venta\_producto[$contador] += 0.00;*

*$date = crear\_fecha();*

*$sql = "INSERT INTO producto (";*

*$sql .=" nombre,cantidad,precio\_compra,precio\_venta,id\_categoria,id\_media,fecha\_actualizacion";*

*$sql .=") VALUES (";*

*$sql .=" '{$nombre\_producto[$contador]}', '{$cantidad\_producto[$contador]}', '{$compra\_producto[$contador]}', '{$venta\_producto[$contador]}', '{$categoria\_producto[$contador]}', {$media\_id[$contador]}, '{$date}'";*

*$sql .=")";*

*if($db->query($sql)) {*

*$indicadores = FALSE;*

*} else {*

*$indicadores = TRUE;*

*}*

*}*

*if(!$indicadores) {*

*$mensaje = "<div class='alert alert-success'><span style='padding-right:5px'>Los registros fueron agregados exitosamente.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$contar\_id = $contador\_tabla;*

*$datos\_productos = "<h3 style='padding:10px' class='bg-primary text-center pad-basic no-btm'>Productos recien agregados</h3><div class='panel panel-default'><table class='tabla-resultado-producto table table-bordered table-striped table-hover'><thead><tr><th class='text-center' style='width: 50px;vertical-align:middle;'>#</th><th class='text-center' style='vertical-align:middle;'>Imagen</th><th class='text-center' style='vertical-align:middle;'>Descripción</th><th class='text-center' style='width: 10%;vertical-align:middle;'> Categoría </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Cantidad </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Precio de compra </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Precio de venta </th><th class='text-center' style='width: 10%;vertical-align:middle;'> Agregado </th></thead><tbody>";*

*if($numero\_recientes != -1) {*

*for($j = 0; $j < $numero\_recientes; $j++) {*

*$c = $j + $contar\_id;*

*$arreglo\_productos\_agregados = encontrar\_productos($productos\_agregados[$j]);*

*$datos\_productos .= "<tr><td class='text-center' style='vertical-align:middle;'>$c</td>";*

*$datos\_productos .= "<td><img class='img-avatar img-circle' src='cargas/productos/".$arreglo\_productos\_agregados['nombre\_archivo']."' alt=''></td>";*

*$datos\_productos .= "<td class='text-center necesario' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['nombre']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['categoria']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['cantidad']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['precio\_compra']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['precio\_venta']."</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>".$arreglo\_productos\_agregados['fecha\_actualizacion']."</td></tr>";*

*}*

*for($counter = 0; $counter < $numero; $counter++) {*

*$total = $counter + $contar\_id + $numero\_recientes;*

*$mostrar\_fecha = leer\_fecha($date);*

*$datos\_productos .= "<tr><td class='text-center' style='vertical-align:middle;'>$total</td>";*

*$datos\_productos .= "<td><img class='img-avatar img-circle' src='cargas/productos/$nombre\_imagen\_productos[$counter]' alt=''></td>";*

*$datos\_productos .= "<td class='text-center necesario' style='vertical-align:middle;'>$nombre\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$nombre\_categoria\_productos[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$cantidad\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$compra\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$venta\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$mostrar\_fecha</td></tr>";*

*}*

*}*

*else {*

*for($counter = 0; $counter < $numero; $counter++) {*

*$total = $counter + $contar\_id;*

*$mostrar\_fecha = leer\_fecha($date);*

*$datos\_productos .= "<tr><td class='text-center' style='vertical-align:middle;'>$total</td>";*

*$datos\_productos .= "<td><img class='img-avatar img-circle' src='cargas/productos/$nombre\_imagen\_productos[$counter]' alt=''></td>";*

*$datos\_productos .= "<td class='text-center necesario' style='vertical-align:middle;'>$nombre\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$nombre\_categoria\_productos[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$cantidad\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$compra\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$venta\_producto[$counter]</td>";*

*$datos\_productos .= "<td class='text-center' style='vertical-align:middle;'>$mostrar\_fecha</td></tr>";*

*}*

*}*

*$datos\_productos .= "</tbody></table></div>";*

*} else {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>La inserción de registros falló.:" . " " . $titulos\_productos\_error . "</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*}*

*}*

*}*

*echo $mensaje;*

*echo $datos\_productos;*

## Sección para cambiar contraseña

En esta vista se observa el formulario para cambiar la contraseña del usuario con sesión actual.

El formulario cuenta con campos para agregar la nueva contraseña y la vieja contraseña.

Al dar clic en el botón “Cambiar” deberá cambiar la contraseña y direccionar al usuario a la página principal para que pueda volver a iniciar sesión nuevamente.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Cambiar contraseña';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$usuario = usuario\_actual();*

*if(isset($\_POST['update'])) {*

*$campos\_requeridos = array('new-password','old-password','id');*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)) {*

*if(sha1($\_POST['old-password']) !== usuario\_actual()['contrasena'] ){*

*$sesion->mensaje('d', "Tu antigua contraseña no coincide");*

*redirigir('cambiar\_contrasena.php',false);*

*}*

*$id = (int)$\_POST['id'];*

*$new = remover\_basura($db->escape(sha1($\_POST['new-password'])));*

*$sql = "UPDATE usuario SET contrasena ='{$new}' WHERE id='{$db->escape($id)}'";*

*$resultado = $db->query($sql);*

*if($resultado && $db->affected\_rows() === 1):*

*$sesion->cerrarSesion();*

*$sesion->mensaje('s',"Inicia sesión con tu nueva contraseña.");*

*redirigir('index.php', false);*

*else:*

*$sesion->mensaje('d',' Lo siento, actualización falló.');*

*redirigir('cambiar\_contrasena.php', false);*

*endif;*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('cambiar\_contrasena.php',false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="login-page">*

*<div class="text-center">*

*<h3>Cambiar contraseña</h3>*

*</div>*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*<form method="post" action="cambiar\_contrasena.php" class="clearfix">*

*<div class="form-group">*

*<label for="newPassword" class="control-label">Nueva contraseña</label>*

*<input autofocus="" type="password" class="form-control" name="new-password" placeholder="Nueva contraseña" required="">*

*</div>*

*<div class="form-group">*

*<label for="oldPassword" class="control-label">Antigua contraseña</label>*

*<input type="password" class="form-control" name="old-password" placeholder="Antigua contraseña" required="">*

*</div>*

*<div class="form-group clearfix">*

*<input type="hidden" name="id" value="<?php echo (int)$usuario['id'];?>">*

*<button type="submit" name="update" class="btn btn-info">Cambiar</button>*

*</div>*

*</form>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 41 se muestra el resultado de la codificación del código de arriba.



Ilustración 41 Sección para cambiar contraseña

## Sección principal

En esta vista se observa un cuadro con un mensaje de bienvenida al iniciar sesión.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Página principal';*

*require\_once('incluye/cargar.php');*

*if (!$sesion->estaLogeado(true)) { redirigir('index.php', false);}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*<div class="col-md-12">*

*<div class="panel">*

*<div class="jumbotron text-center">*

*<h1>Esta es su nueva página de inicio</h1>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 42 se muestra el resultado de la codificación del código de arriba.

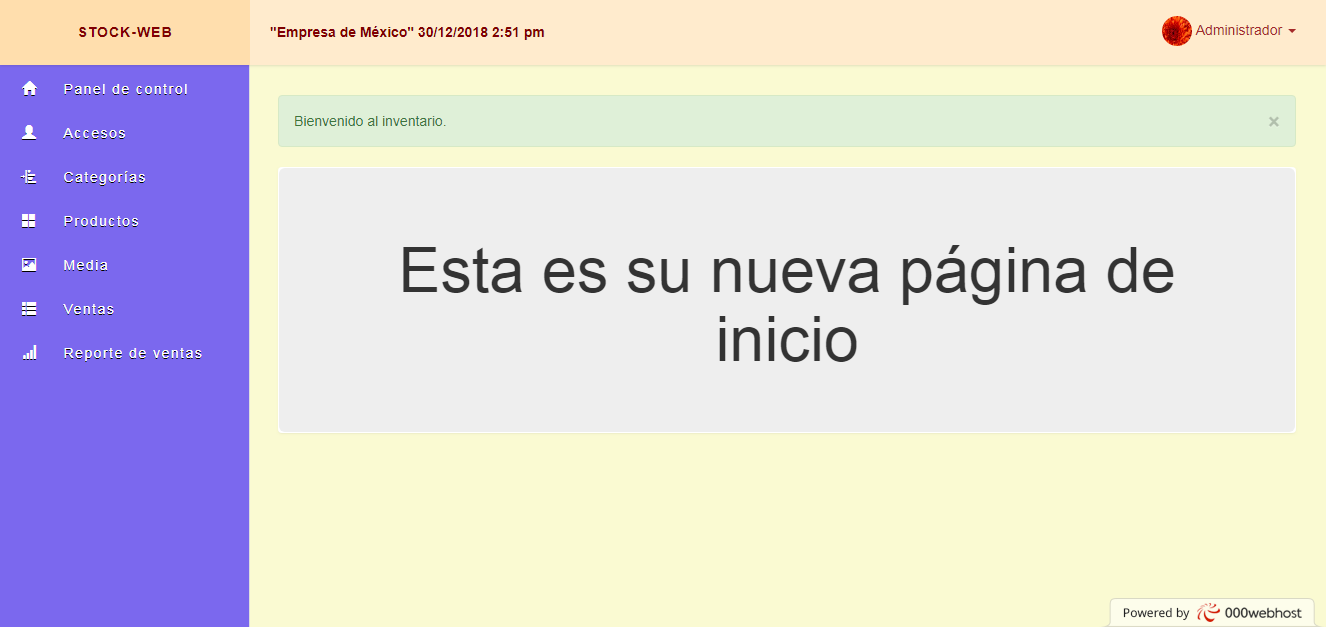


Ilustración 42 Sección principal

## Sección de categorías

En esta vista se observa un formulario para agregar una categoría y un panel con la lista de categorías registradas con opciones de editarlas o eliminarlas.

El formulario cuenta con un campo para agregar una categoría y el botón para registrarla.

Al registrar una categoría, esta no deberá estar registrada previamente.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Lista de categorías';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$todas\_categorias = encontrar\_todos('categoria');*

*if(isset($\_POST['agregar\_categoria'])) {*

*$campo\_requerido = array('nombre-categoria');*

*validar\_campos($campo\_requerido);*

*$nombre\_categoria = remover\_basura($db->escape($\_POST['nombre-categoria']));*

*if(empty($errores)) {*

*$comprobar = encontrar\_nombre\_categoria($nombre\_categoria);*

*$variable = null;*

*foreach($comprobar as $buscar\_categoria) {*

*$variable = $buscar\_categoria['nombre'];*

*}*

*if($variable == null) {*

*$sql = "INSERT INTO categoria (nombre)";*

*$sql .= " VALUES ('{$nombre\_categoria}')";*

*if($db->query($sql)){*

*$sesion->mensaje("s", "Categoría agregada exitosamente.");*

*redirigir('categoria.php',false);*

*}*

*} else {*

*$sesion->mensaje("d", "Lo siento, ya hay una categoría con ese nombre.");*

*redirigir('categoria.php',false);*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('categoria.php',false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Agregar categoría</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<form method="post" action="categoria.php">*

*<div class="form-group">*

*<input autofocus="" type="text" class="form-control" name="nombre-categoria" placeholder="Nombre de la categoría" required>*

*</div>*

*<button type="submit" name="agregar\_categoria" class="btn btn-primary">Agregar categoría</button>*

*</form>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Lista de categorías</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered table-striped table-hover">*

*<thead>*

*<tr>*

*<th class="text-center" style="width: 50px;">#</th>*

*<th>Categorías</th>*

*<th class="text-center" style="width: 100px;">Acciones</th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($todas\_categorias as $categoria): ?>*

*<?php if($categoria['nombre'] != "No hay"): ?>*

*<tr>*

*<td class="text-center"><?php echo contar\_id();?></td>*

*<td><?php echo remover\_basura(ucfirst($categoria['nombre'])); ?></td>*

*<td class="text-center">*

*<div class="btn-group">*

*<a href="editar\_categoria.php?id=<?php echo (int)$categoria['id'];?>" class="btn btn-xs btn-warning" data-toggle="tooltip" title="Editar">*

*<span class="glyphicon glyphicon-edit"></span>*

*</a>*

*<a href="borrar\_categoria.php?id=<?php echo (int)$categoria['id'];?>" class="btn btn-xs btn-danger" data-toggle="tooltip" title="Eliminar">*

*<span class="glyphicon glyphicon-trash"></span>*

*</a>*

*</div>*

*</td>*

*</tr>*

*<?php endif; ?>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 43 se muestra el resultado de la codificación del código de arriba.

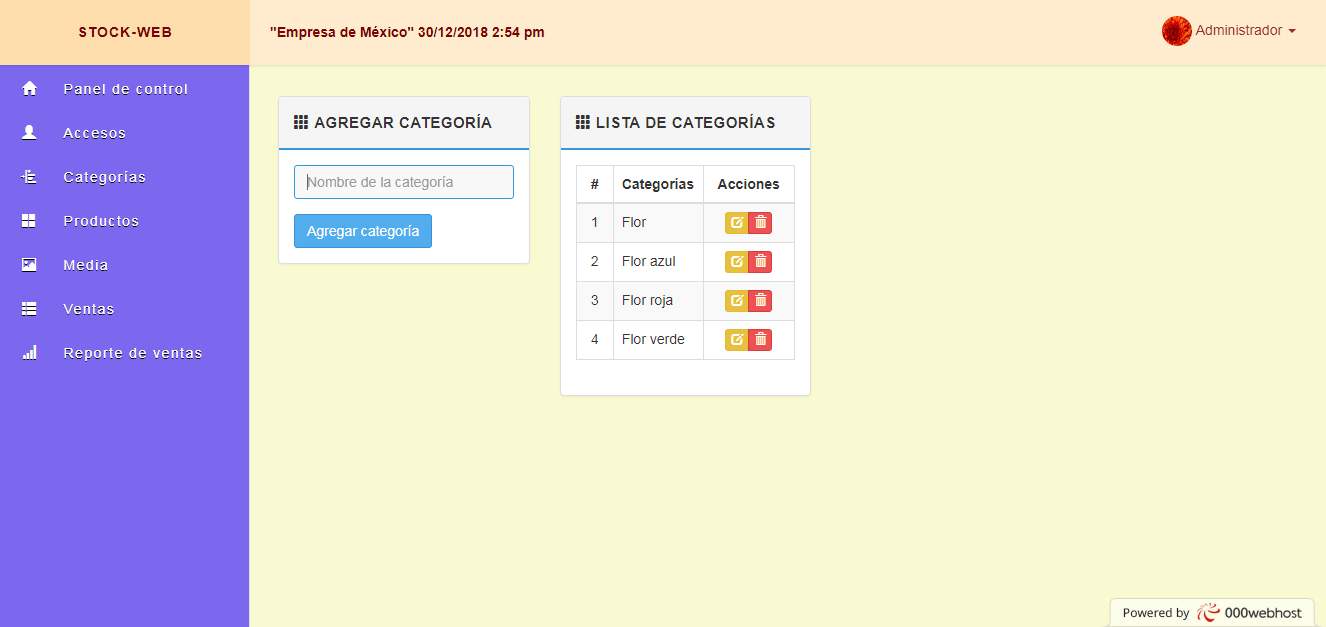


Ilustración 43 Sección de categorías

## Archivo para cerrar la sesión de un usuario

En este archivo *“cerrar\_sesion.php”* se cierra la sesión del usuario actual. Primero se carga el archivo cargar.php ubicado en el directorio incluye. Finalmente se checa si ya se cerró sesión por medio del objeto *“sesión“* y su método *“cerrarSesion“*, si es el caso se cierra la sesión y se redirige a la página principal.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*if(!$sesion->cerrarSesion()) {redirigir("index.php");}*

## Sección para editar categoría

En esta vista se observa el formulario para editar una categoría.

El formulario cuenta con el campo de nombre de categoría

Al editar el nombre de grupo, no debe existir otro con nombre igual.

Al dar clic en el botón de editar categoría, deberá editar la categoría.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Editar categoría';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$categoria = encontrar\_por\_id('categoria',(int)$\_GET['id']);*

*if(!$categoria) {*

*$sesion->mensaje("d","Id de categoría perdido.");*

*redirigir('categoria.php');*

*}*

*if(isset($\_POST['editar\_categoria'])) {*

*$req\_field = array('nombre-categoria');*

*validar\_campos($req\_field);*

*$cat\_name = remover\_basura($db->escape($\_POST['nombre-categoria']));*

*$categorias = encontrar\_nombre\_categoria($cat\_name);*

*foreach($categorias as $categoria):*

*$nombreCategoria = $categoria['nombre'];*

*endforeach;*

*$categoriaActual = encontrar\_por\_id("categorias", $\_GET['id']);*

*if(empty($errores)){*

*$sql = "UPDATE categoria SET nombre='{$cat\_name}'";*

*$sql .= " WHERE id='{$categoria['id']}'";*

*$resultado = $db->query($sql);*

*if($resultado && $db->affected\_rows() === 1) {*

*$sesion->mensaje("s", "Categoría actualizada.");*

*redirigir('categoria.php',false);*

*} else {*

*if($categoriaActual['nombre'] != $nombreCategoria) {*

*$sesion->mensaje("d", "Lo siento, ya hay una categoría con ese nombre.");*

*redirigir("editar\_categoria.php?id=$\_GET[id]",false);*

*} else {*

*if($categoriaActual['nombre'] == $nombreCategoria) {*

*redirigir("categoria.php",false);*

*} else {*

*$sesion->mensaje("d", "Lo siento, no se pudo actualizar.");*

*redirigir("editar\_categoria.php?id=$\_GET[id]",false);*

*}*

*}*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir("editar\_categoria.php?id=$\_GET[id]",false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Editando <?php echo remover\_basura(ucfirst($categoria['nombre']));?></span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<form method="post" action="editar\_categoria.php?id=<?php echo (int)$categoria['id'];?>">*

*<div class="form-group">*

*<input type="text" class="form-control" name="nombre-categoria" value="<?php echo remover\_basura(ucfirst($categoria['nombre']));?>" required="">*

*</div>*

*<button type="submit" name="editar\_categoria" class="btn btn-primary">Actualizar categoría</button>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 44 se muestra el resultado de la codificación del código de arriba.

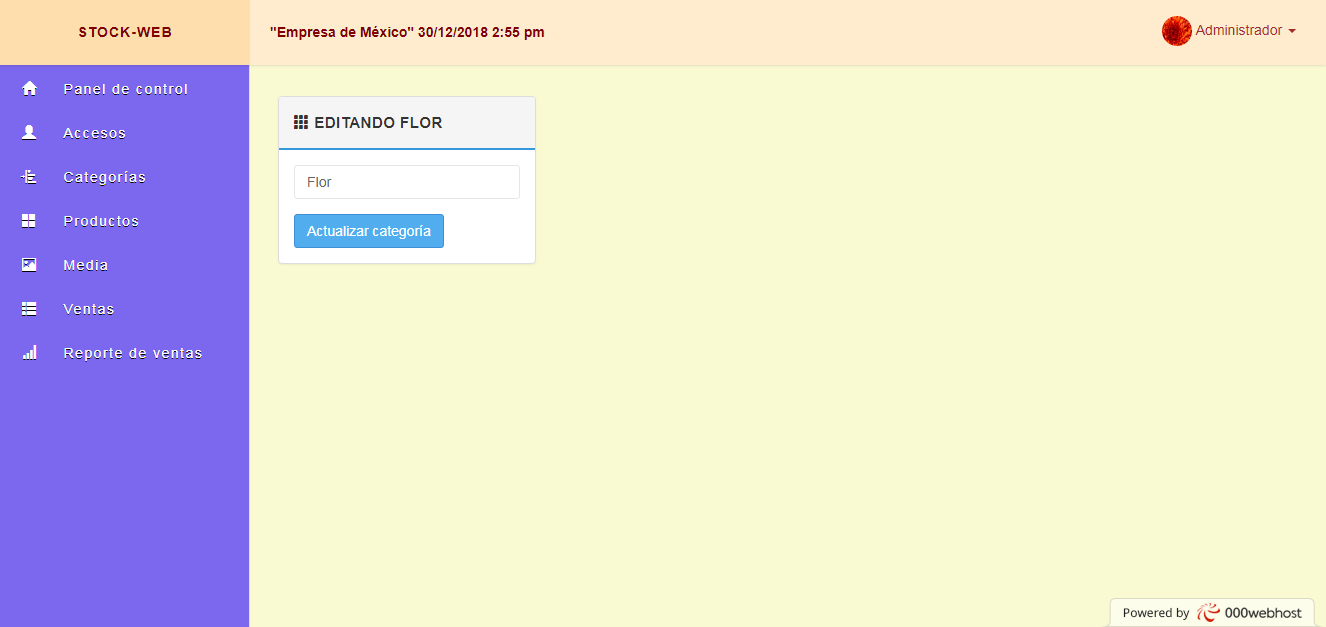


Ilustración 44 Sección para agregar categoría

## Sección para editar cuenta

En esta vista se observa el formulario para editar la cuenta y 4 paneles para todos los usuarios como cambiar la foto de perfil, cambiar la foto de fondo, cambiar tema de la aplicación, seleccionar reportes de ventas automáticos y un último panel para el usuario tipo administrador donde se puede cambiar el lema y logo de la empresa.

En el formulario para editar la cuenta, se podrá cambiar el nombre de usuario (no puede haber otro igual) así como el nombre de la persona dueña de la cuenta; finalmente hay un botón con enlace a la sección para cambiar la contraseña.

En los paneles para cambiar foto de perfil, de fondo y el logo de la empresa se hace por medio de un selector de imagen.

En el resto de los paneles se ingresa la información por medio de cuadros de texto y listas desplegables.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Editar cuenta';*

*require\_once('incluye/cargar.php');*

*/\**

*$editar\_usuario = encontrar\_por\_id('usuarios',(int)$\_POST['user\_id']);*

*$grupos = encontrar\_todos('grupo\_usuario');*

*\*/*

*pagina\_requiere\_nivel(3);*

*if(isset($\_POST['submit'])) {*

*$photo = new Media();*

*$id\_usuario = (int)$\_POST['user\_id'];*

*$photo->subir($\_FILES['usuario\_upload']);*

*if($photo->proceso\_usuario($id\_usuario)){*

*$sesion->mensaje('s','La foto fue subida al servidor.');*

*redirigir('editar\_cuenta.php');*

*} else{*

*$sesion->mensaje('d',join($photo->errors));*

*redirigir('editar\_cuenta.php');*

*}*

*}*

*if(isset($\_POST['not'])) {*

*$id\_usuario = (int)$\_POST['user\_id'];*

*$holas = new Media();*

*$holas->borrar\_imagen($id\_usuario);*

*$booleano = updateImagen($id\_usuario);*

*if($booleano){*

*$sesion->mensaje('s','La foto por defecto fue colocada.');*

*redirigir('editar\_cuenta.php');*

*} else{*

*$sesion->mensaje('d',join($photo->errors));*

*redirigir('editar\_cuenta.php');*

*}*

*}*

*if(isset($\_POST['fondo'])) {*

*$photo = new Media();*

*$id\_usuario = (int)$\_POST['user\_id'];*

*$photo->subir($\_FILES['fondo\_upload']);*

*if($photo->process\_fondo($id\_usuario)){*

*$sesion->mensaje('s','La foto fue subida al servidor.');*

*redirigir('editar\_cuenta.php');*

*} else{*

*$sesion->mensaje('d',join($photo->errors));*

*redirigir('editar\_cuenta.php');*

*}*

*}*

*if(isset($\_POST['not\_fondo'])) {*

*$id\_usuario = (int)$\_POST['user\_id'];*

*$hola = new Media();*

*$hola->borrar\_fondo($id\_usuario);*

*$booleano = updateImagenFondo($id\_usuario);*

*if($booleano){*

*$sesion->mensaje('s','La foto de fondo fue retirada.');*

*redirigir('editar\_cuenta.php');*

*} else{*

*$sesion->mensaje('d',join($photo->errors));*

*redirigir('editar\_cuenta.php');*

*}*

*}*

*if(isset($\_POST['update'])) {*

*$campos\_requeridos = array('name','username' );*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)) {*

*$id = (int)$\_SESSION['user\_id'];*

*$nombre = remover\_basura($db->escape($\_POST['name']));*

*$nombre\_usuario = remover\_basura($db->escape($\_POST['username']));*

*$sql = "UPDATE usuario SET nombre ='{$nombre}', username ='{$nombre\_usuario}' WHERE id='{$id}'";*

*$resultado = $db->query($sql);*

*if($resultado && $db->affected\_rows() === 1) {*

*$sesion->mensaje('s',"Cuenta actualizada. ");*

*redirigir('editar\_cuenta.php', false);*

*} else {*

*#$sesion->mensaje('d',' Lo siento, la actualización no se efectuó.');*

*redirigir('editar\_cuenta.php', false);*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('editar\_cuenta.php',false);*

*}*

*}*

*if(isset($\_POST['empresa'])) {*

*$lema = $\_POST['lema'];*

*$comprobarLema = updateLema($lema);*

*$photo = new Media();*

*$photo->subir($\_FILES['logo\_upload']);*

*$comprobarLogo = $photo->proceso\_logo();*

*if($comprobarLema && $comprobarLogo){*

*$sesion->mensaje('s','El logo fue subido al servidor y el lema fue actualizado.');*

*redirigir('editar\_cuenta.php');*

*}*

*if($comprobarLema && !$comprobarLogo) {*

*$sesion->mensaje('s','El lema fue actualizado.');*

*redirigir('editar\_cuenta.php');*

*}*

*if(!$comprobarLema && $comprobarLogo) {*

*$sesion->mensaje('s','El logo fue subido al servidor.');*

*redirigir('editar\_cuenta.php');*

*}*

*else {*

*$sesion->mensaje('d',join($photo->errors));*

*redirigir('editar\_cuenta.php');*

*}*

*}*

*if(isset($\_POST['not\_empresa'])) {*

*$hola = new Media();*

*$hola->borrar\_logo();*

*$booleano = updateLogo($id\_usuario);*

*if($booleano){*

*$sesion->mensaje('s','El logo fue retirado.');*

*redirigir('editar\_cuenta.php');*

*} else{*

*$sesion->mensaje('d',join($photo->errors));*

*redirigir('editar\_cuenta.php');*

*}*

*}*

*if(isset($\_POST['tema'])) {*

*$id\_usuario = (int)$\_POST['user\_id'];*

*$idTema = intval($\_POST['tema-seleccion']);*

*if($idTema > 0) {*

*$comprobarTema = updateTema($id\_usuario,$idTema);*

*if($comprobarTema) {*

*$sesion->mensaje('s','El tema fue actualizado.');*

*redirigir('editar\_cuenta.php');*

*}*

*}*

*}*

*if(isset($\_POST['reporte'])) {*

*$id\_usuario = (int)$\_POST['user\_id'];*

*if(isset($\_POST['diario'])) {*

*if(!buscarTipoReporte($id\_usuario, 1)) {*

*insertarTipoReporte($id\_usuario, 1);*

*$sesion->mensaje('s','Se realizaron cambios en reportes de ventas.');*

*}*

*} if(isset($\_POST['mensual'])) {*

*if(!buscarTipoReporte($id\_usuario, 2)) {*

*insertarTipoReporte($id\_usuario, 2);*

*$sesion->mensaje('s','Se realizaron cambios en reportes de ventas.');*

*}*

*} if(isset($\_POST['fecha'])) {*

*if(!buscarTipoReporte($id\_usuario, 3)) {*

*insertarTipoReporte($id\_usuario, 3);*

*$fecha\_inicio = $\_POST['fecha-inicio'];*

*$fecha\_fin = $\_POST['fecha-fin'];*

*$resultados = buscarTipoReportes($id\_usuario, 3);*

*foreach($resultados as $resultado):*

*$id\_tipos\_reporte = $resultado['id'];*

*endforeach;*

*insertarRangoFecha($id\_tipos\_reporte, $fecha\_inicio, $fecha\_fin);*

*$sesion->mensaje('s','Se realizaron cambios en reportes de ventas.');*

*}*

*} if(!isset($\_POST['diario'])) {*

*if(buscarTipoReporte($id\_usuario, 1)) {*

*borrarTipoReporte($id\_usuario, 1);*

*$sesion->mensaje('s','Se realizaron cambios en reportes de ventas.');*

*}*

*} if(!isset($\_POST['mensual'])) {*

*if(buscarTipoReporte($id\_usuario, 2)) {*

*borrarTipoReporte($id\_usuario, 2);*

*$sesion->mensaje('s','Se realizaron cambios en reportes de ventas.');*

*}*

*} if(!isset($\_POST['fecha'])) {*

*if(buscarTipoReporte($id\_usuario, 3)) {*

*$resultados = buscarTipoReportes($id\_usuario, 3);*

*foreach($resultados as $resultado):*

*$id\_tipos\_reporte = $resultado['id'];*

*endforeach;*

*borrarRangoFecha($id\_tipos\_reporte);*

*borrarTipoReporte($id\_usuario, 3);*

*$sesion->mensaje('s','Se realizaron cambios en reportes de ventas.');*

*}*

*}*

*redirigir('editar\_cuenta.php');*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span class="glyphicon glyphicon-camera"></span>*

*<span>Cambiar mi foto</span>*

*</div>*

*<div class="container-fluid">*

*<div class="row">*

*<div class="col-md-12 panel-body">*

*<form class="form" action="editar\_cuenta.php" method="POST" enctype="multipart/form-data">*

*<div class="form-group">*

*<input style="width:100.5%" type="file" name="usuario\_upload" multiple="multiple" class="btn btn-default btn-file"/>*

*</div>*

*<div class="form-group">*

*<input type="hidden" name="user\_id" value="<?php echo $usuario['id'];?>">*

*<button type="submit" name="submit" style="margin-bottom:10px" class="btn btn-warning col-xs-12 col-md-5">Cambiar</button>*

*<button type="submit" name="not" class="btn btn-warning pull-right col-xs-12 col-md-6">Quitar mi foto actual</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span class="glyphicon glyphicon-edit"></span>*

*<span>Editar mi cuenta</span>*

*</div>*

*<div class="panel-body">*

*<form method="post" action="editar\_cuenta.php?id=<?php echo (int)$usuario['id'];?>" class="clearfix">*

*<div class="form-group">*

*<label for="name" class="control-label">Nombre(s)</label>*

*<input type="name" class="form-control" required="" name="name" value="<?php echo remover\_basura(ucwords($usuario['nombre\_usuario'])); ?>">*

*</div>*

*<div class="form-group">*

*<label for="username" class="control-label">Usuario</label>*

*<input type="text" class="form-control" required="" name="username" value="<?php echo remover\_basura(ucwords($usuario['username'])); ?>">*

*</div>*

*<div class="form-group clearfix">*

*<a href="cambiar\_contrasena.php" title="change password" style="margin-bottom:10px" class="col-xs-12 col-md-6 btn btn-danger pull-right">Cambiar contraseña</a>*

*<button type="submit" name="update" class="btn btn-info col-xs-12 col-md-5">Actualizar</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span class="glyphicon glyphicon-camera"></span>*

*<span>Cambiar mi fondo</span>*

*</div>*

*<div class="container-fluid">*

*<div class="row">*

*<div class="col-md-12 panel-body">*

*<form class="form" action="editar\_cuenta.php" method="POST" enctype="multipart/form-data">*

*<div class="form-group">*

*<input style="width:100.5%" type="file" name="fondo\_upload" multiple="multiple" class="btn btn-default btn-file"/>*

*</div>*

*<div class="form-group">*

*<input type="hidden" name="user\_id" value="<?php echo $usuario['id'];?>">*

*<button type="submit" name="fondo" style="margin-bottom:10px" class="col-xs-12 col-md-4 btn btn-warning">Cambiar</button>*

*<button type="submit" name="not\_fondo" class="col-xs-12 col-md-7 btn btn-warning pull-right">Quitar mi fondo actual</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php if($usuario['nivel'] === '1'): ?>*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span class="glyphicon glyphicon-edit"></span>*

*<span>Cambiar logo y/o lema</span>*

*</div>*

*<div class="container-fluid">*

*<div class="row">*

*<div class="col-md-12 panel-body">*

*<form class="form" action="editar\_cuenta.php" method="POST" enctype="multipart/form-data">*

*<div class="form-group">*

*<label for="lema" class="control-label">Lema de la empresa</label>*

*<input type="text" class="form-control" name="lema" value="<?php echo remover\_basura($empresa['lema']); ?>">*

*</div>*

*<div class="form-group">*

*<input style="width:100.5%" type="file" name="logo\_upload" multiple="multiple" class="btn btn-default btn-file"/>*

*</div>*

*<div class="form-group">*

*<button type="submit" name="empresa" class="btn btn-warning">Cambiar</button>*

*<button type="submit" name="not\_empresa" class="btn btn-warning pull-right">Quitar logo</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php endif;?>*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span class="glyphicon glyphicon-edit"></span>*

*<span>Cambiar tema</span>*

*</div>*

*<div class="container-fluid">*

*<div class="row">*

*<div class="col-md-12 panel-body" >*

*<form class="form" action="" method="POST">*

*<div class="form-group">*

*<label for="tema" class="control-label">Tema</label>*

*<input type="hidden" name="user\_id" value="<?php echo $usuario['id'];?>">*

*<select class="form-control" id="seleccion" name="tema-seleccion">*

*<option value="0" selected="">Selecciona una opción</option>*

*<option value="1">Por defecto</option>*

*<option value="2">Pastel</option>*

*<option value="3">Personalizado 1</option>*

*<option value="4">Personalizado 2</option>*

*<option value="5">Personalizado 3</option>*

*</select>*

*</div>*

*<div class="form-group">*

*<button type="submit" name="tema" id="tema" class="btn btn-warning">Cambiar tema</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<div class="col-md-6">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span class="glyphicon glyphicon-list-alt"></span>*

*<span>Reportes de ventas automáticos</span>*

*</div>*

*<div class="container-fluid">*

*<div class="row">*

*<div class="col-md-12 panel-body" >*

*<form class="form" action="" method="POST">*

*<div class="form-group">*

*<input type="hidden" name="user\_id" value="<?php echo $usuario['id'];?>">*

*<div class="col-md-4">*

*<label for="diario" class="control-label">Diarias</label>*

*<input style="transform:scale(1.1);" type="checkbox" name="diario" value="ON" />*

*</div>*

*<div class="col-md-4">*

*<label for="mensual" class="control-label">Mensuales</label>*

*<input style="transform:scale(1.1);" type="checkbox" name="mensual" value="ON" />*

*</div>*

*<div class="col-md-4">*

*<label for="fecha" class="control-label">Entre fechas</label>*

*<input id="fecha" style="transform:scale(1.1);" type="checkbox" name="fecha" value="ON" />*

*</div>*

*<div>*

*<label class="form-label">Rango de fechas</label>*

*<div class="input-group">*

*<input required="" disabled type="text" class="datepicker form-control fecha-inicio" name="fecha-inicio" id="fecha-inicio" placeholder="De" autocomplete="off">*

*<span class="input-group-addon"><i class="glyphicon glyphicon-menu-right"></i></span>*

*<input required="" disabled type="text" class="datepicker form-control fecha-fin" name="fecha-fin" id="fecha-fin" placeholder="Hasta" autocomplete="off">*

*</div>*

*</div>*

*</div>*

*<div class="form-group">*

*<button type="submit" name="reporte" id="tema" class="btn btn-warning">Actualizar reportes automáticos</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="libs/js/moment.js"></script>*

*<script>*

*$(document).ready(function fecha() {*

*$('#fecha').on('change', function() {*

*if ($(this).is(':checked')) {*

*$("#fecha-inicio").prop("disabled", false);*

*$("#fecha-fin").prop("disabled", false);*

*} else {*

*$("#fecha-inicio").prop("disabled", true);*

*$("#fecha-fin").prop("disabled", true);*

*}*

*});*

*$(".fecha-inicio").change(function fechaVar() {*

*var valor = $(".fecha-fin").val();*

*if(valor === '') {*

*var fecha1;*

*var fecha2;*

*var fecha;*

*var diferencia;*

*fecha = $(".fecha-inicio").val();*

*fecha1 = moment(fecha);*

*fecha2 = moment(new Date());*

*diferencia = fecha1.diff(fecha2,'days');*

*if(diferencia > 0) {*

*$(".fecha-fin").datepicker('option', 'minDate', diferencia + 1);*

*} else {*

*$(".fecha-fin").datepicker('option', 'minDate', diferencia);*

*}*

*}*

*});*

*$(".fecha-fin").change(function fechaObj() {*

*var valor = $(".fecha-inicio").val();*

*if(valor === '') {*

*var fecha1;*

*var fecha2;*

*var fecha;*

*var diferencia;*

*fecha = $(".fecha-fin").val();*

*fecha1 = moment(fecha);*

*fecha2 = moment(new Date());*

*diferencia = fecha1.diff(fecha2,'days');*

*alert(diferencia);*

*if(diferencia > 0) {*

*$(".fecha-inicio").datepicker('option', 'maxDate', diferencia + 1);*

*} else {*

*$(".fecha-inicio").datepicker('option', 'maxDate', diferencia);*

*}*

*}*

*});*

*});*

*</script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 45 se muestra el resultado de la codificación del código de arriba

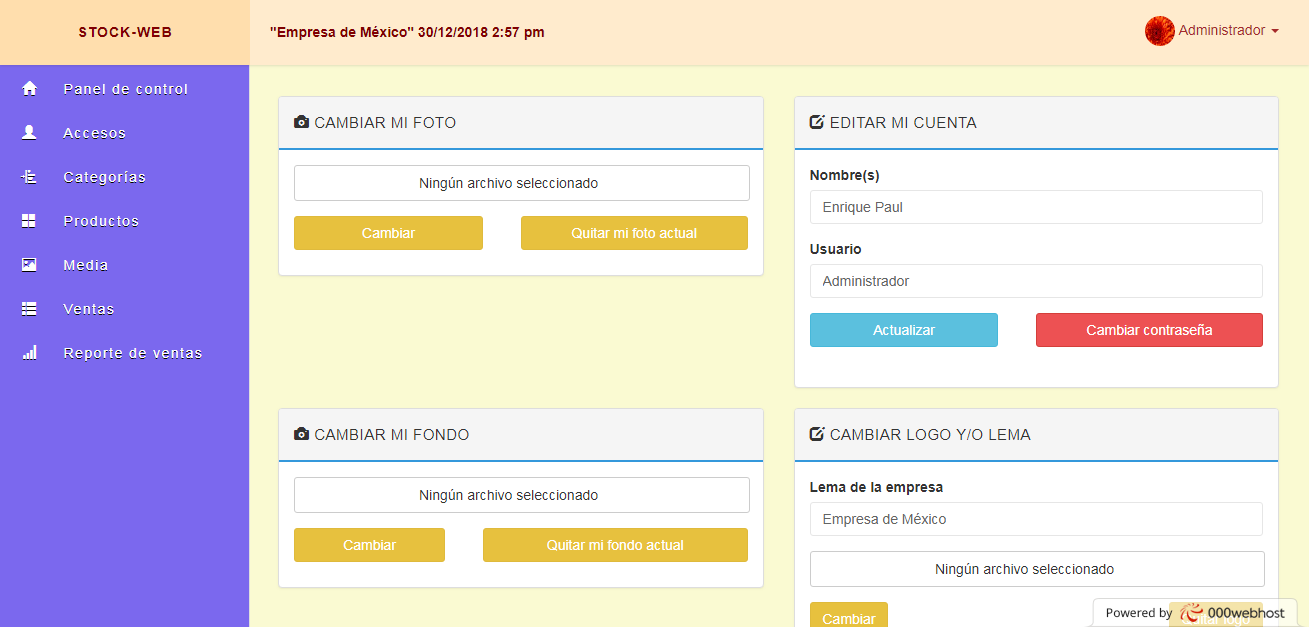


Ilustración 45 Sección para editar cuenta

## Sección para editar grupo

En esta vista se observa el formulario para editar un grupo de usuarios.

El formulario cuenta con campos para editar el nombre de grupo, el nivel de grupo y el estado de grupo.

Al ingresar el nombre de grupo, no debe existir otro con nombre igual.

Al ingresar el nivel de grupo, este podrá tener como valor el rango 1-3.

Al ingresar el estado de grupo, este podrá tener como valor activo o inactivo.

En el caso del grupo de administradores solo se podrá editar el nombre de grupo.

Al dar clic en el botón de actualizar, deberá editar el grupo.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Editar grupo';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$editar\_grupo = encontrar\_por\_id('grupo',(int)$\_GET['id']);*

*if(isset($\_POST['actualizar'])){*

*$campos\_requeridos = array('nombre-grupo','nivel-grupo');*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)) {*

*$nombre = remover\_basura($db->escape($\_POST['nombre-grupo']));*

*$nivelIngresado = remover\_basura($db->escape($\_POST['nivel-grupo']));*

*$estado = remover\_basura($db->escape($\_POST['estado']));*

*$grupos = encontrar\_nombre\_grupo($nombre);*

*foreach($grupos as $grupo):*

*$nombreGrupo = $grupo['nombre'];*

*endforeach;*

*$niveles = encontrar\_nivel\_grupo($nivelIngresado);*

*foreach($niveles as $nivel):*

*$nivelGrupo = $nivel['nivel'];*

*endforeach;*

*$grupoActual = encontrar\_por\_id("grupo", $\_GET['id']);*

*$condicionA = false;*

*$condicionB = false;*

*if(isset($nombreGrupo)) {*

*if($nombreGrupo != $grupoActual['nombre']) {*

*$sesion->mensaje('d',' Ese nombre de grupo ya existe. ');*

*redirigir("editar\_grupo.php?id=$\_GET[id]", false);*

*} else {*

*$condicionA = true;*

*}*

*}*

*if(isset($nivelGrupo)) {*

*if($nivelGrupo != $grupoActual['nivel']) {*

*$sesion->mensaje('d',' Ese nivel de grupo ya existe. ');*

*redirigir("editar\_grupo.php?id=$\_GET[id]", false);*

*} else {*

*$condicionB = true;*

*}*

*}*

*$query = "UPDATE grupo SET ";*

*$query .= "nombre='{$nombre}',nivel=$nivelIngresado,estado\_actual='{$estado}' ";*

*$query .= "WHERE ID='{$db->escape($editar\_grupo['id'])}'";*

*$resultado = $db->query($query);*

*if($resultado && $db->affected\_rows() === 1){*

*$sesion->mensaje('s',"El grupo se ha actualizado.");*

*redirigir('grupo.php?id='.(int)$editar\_grupo['id'], false);*

*} elseif(!$condicionA || !$condicionB) {*

*$sesion->mensaje('d','Lamentablemente no se ha actualizado el grupo.');*

*redirigir('editar\_grupo.php?id='.(int)$editar\_grupo['id'], false);*

*} else {*

*redirigir('grupo.php?id='.(int)$editar\_grupo['id'], false);*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('editar\_grupo.php?id='.(int)$editar\_grupo['id'], false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="login-page" style='margin-top:-2px'>*

*<div class="text-center">*

*<h3>Editar grupo</h3>*

*</div>*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*<form method="post" action="editar\_grupo.php?id=<?php echo (int)$editar\_grupo['id'];?>" class="clearfix">*

*<div class="form-group">*

*<label for="name" class="control-label">Nombre del grupo</label>*

*<input type="name" class="form-control" name="nombre-grupo" required="" value="<?php echo remover\_basura(ucwords($editar\_grupo['nombre'])); ?>">*

*</div>*

*<div class="form-group">*

*<label for="level" class="control-label">Nivel del grupo</label>*

*<input type="hidden" name="nivel-grupo" value="<?php echo (int)$editar\_grupo['nivel']; ?>" />*

*<?php if($editar\_grupo['nivel'] !== '1'): ?>*

*<input type="number" min="1" max="3" required="" class="form-control" name="nivel-grupo" value="<?php echo (int)$editar\_grupo['nivel']; ?>">*

*<?php else: ?>*

*<input type="number" disabled min="1" max="3" class="form-control" name="nivel-grupo" value="<?php echo (int)$editar\_grupo['nivel']; ?>">*

*<?php endif;?>*

*</div>*

*<div class="form-group">*

*<label for="estado">Estado</label>*

*<input type="hidden" name="estado" value="<?php echo (int)$editar\_grupo['estado\_actual']; ?>" />*

*<?php if($editar\_grupo['estado\_actual'] !== '1' || $editar\_grupo['nivel'] !== '1'): ?>*

*<select class="form-control" name="estado" required="">*

*<?php else: ?>*

*<select disabled class="form-control" name="estado">*

*<?php endif;?>*

*<option <?php if ($editar\_grupo['estado\_actual'] === '1') {*

*echo 'selected="selected"';*

*}*

*?> value="1"> Activo </option>*

*<option <?php if ($editar\_grupo['estado\_actual'] === '0') {*

*echo 'selected="selected"';*

*}*

*?> value="0">Inactivo</option>*

*</select>*

*</div>*

*<div class="form-group clearfix">*

*<button type="submit" name="actualizar" class="btn btn-info">Actualizar</button>*

*</div>*

*</form>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 46 se muestra el resultado de la codificación del código de arriba.

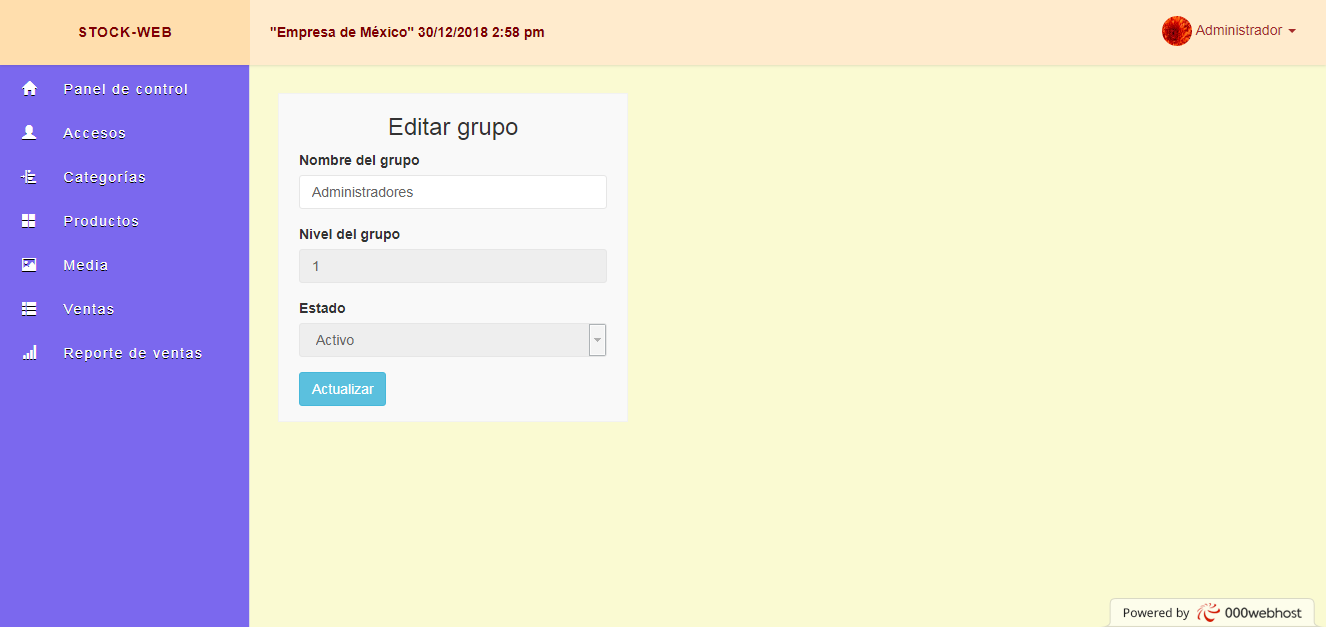


Ilustración 46 Sección para editar grupo

## Sección para editar producto

En esta vista se observa el formulario para editar un producto.

El formulario cuenta con campos para editar el nombre de producto, la categoría del producto, la imagen del producto, la cantidad de producto existente, el precio de compra y el precio de venta.

Al editar el nombre de producto, no debe existir otro con nombre igual.

Al dar clic en el botón de editar producto, deberá editar el producto.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Editar producto';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$producto = encontrar\_por\_id('producto',(int)$\_GET['id']);*

*$todas\_categorias = encontrar\_todos('categoria');*

*$todas\_foto = encontrar\_todos('media');*

*if(!$producto) {*

*$sesion->mensaje("d","Id de producto pérdida.");*

*redirigir('producto.php');*

*}*

*if(isset($\_POST['producto'])) {*

*$campos\_requeridos = array('titulo\_producto','categoria\_producto','cantidad\_producto','precio\_compra', 'precio\_venta' );*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)) {*

*$nombre\_producto = remover\_basura($db->escape($\_POST['titulo\_producto']));*

*$categoria\_producto = (int)$\_POST['categoria\_producto'];*

*$cantidad\_producto = remover\_basura($db->escape($\_POST['cantidad\_producto']));*

*$compra\_producto = remover\_basura($db->escape($\_POST['precio\_compra']));*

*$compra\_decimal\_producto = remover\_basura($db->escape($\_POST['compra\_decimal']));*

*$compra\_decimal\_producto += 0.00;*

*$compra\_decimal\_producto /= 100.00;*

*$compra\_producto += $compra\_decimal\_producto;*

*$venta\_producto = remover\_basura($db->escape($\_POST['precio\_venta']));*

*$venta\_decimal\_producto = remover\_basura($db->escape($\_POST['precio\_decimal']));*

*$venta\_decimal\_producto += 0.00;*

*$venta\_decimal\_producto /= 100.00;*

*$venta\_producto += $venta\_decimal\_producto;*

*$date = crear\_fecha();*

*if (is\_null($\_POST['foto-producto']) || $\_POST['foto-producto'] === "") {*

*$media\_id = "NULL";*

*} else {*

*$media\_id = remover\_basura($db->escape($\_POST['foto-producto']));*

*}*

*$comprobar = encontrar\_nombre\_producto($nombre\_producto);*

*foreach ($comprobar as $buscar\_producto) {*

*$variable = $buscar\_producto['nombre'];*

*}*

*$productoActual = encontrar\_por\_id("producto", $\_GET['id']);*

*if($variable != $productoActual['nombre'] && $variable != null) {*

*$sesion->mensaje('d',"Ya hay un nombre de producto igual.");*

*redirigir('editar\_producto.php?id='.(int)$productoActual['id'], false);*

*}*

*$query = "UPDATE producto SET";*

*$query .=" nombre ='{$nombre\_producto}', cantidad ='{$cantidad\_producto}',";*

*$query .=" precio\_compra ='{$compra\_producto}', precio\_venta ='{$venta\_producto}', id\_categoria ='{$categoria\_producto}',id\_media={$media\_id},fecha\_actualizacion='{$date}'";*

*$query .=" WHERE id ='{$producto['id']}'";*

*$resultado = $db->query($query);*

*if($resultado && $db->affected\_rows() === 1){*

*$sesion->mensaje('s',"El producto ha sido actualizado. ");*

*redirigir('producto.php', false);*

*} else {*

*if(isset($variable)) {*

*redirigir('producto.php', false);*

*} else {*

*$sesion->mensaje('d','Ya hay un producto con ese nombre.');*

*redirigir('editar\_producto.php?id='.$producto['id'], false);*

*}*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('editar\_producto.php?id='.$producto['id'], false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Editar producto</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<div class="col-md-12">*

*<form method="POST" action="editar\_producto.php?id=<?php echo (int)$producto['id'] ?>">*

*<div class="form-group">*

*<label for="titulo\_producto">Descripción</label>*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-th-large"></i>*

*</span>*

*<input type="text" class="form-control" name="titulo\_producto" value="<?php echo remover\_basura($producto['nombre']);?>" required="">*

*</div>*

*</div>*

*<div>*

*<div class="row">*

*<div class="col-md-6 form-group">*

*<label for="categoria">Categoría</label>*

*<select class="form-control" name="categoria\_producto" required="">*

*<option value="">Selecciona una categoría</option>*

*<?php foreach ($todas\_categorias as $categoria): ?>*

*<?php if($categoria['nombre'] != "No hay"): ?>*

*<option value="<?php echo (int)$categoria['id']; ?>" <?php if($producto['id\_categoria'] === $categoria['id']): echo "selected"; endif; ?> >*

*<?php echo remover\_basura($categoria['nombre']); ?></option>*

*<?php endif; ?>*

*<?php endforeach; ?>*

*</select>*

*</div>*

*<div class="col-md-6 form-group">*

*<label for="imagen">Imágen</label>*

*<select class="form-control" name="foto-producto">*

*<option value=""> Selecciona una imágen </option>*

*<?php foreach ($todas\_foto as $foto): ?>*

*<?php if($foto['nombre\_archivo'] != "no\_image.jpg"): ?>*

*<option value="<?php echo (int)$foto['id'];?>" <?php if($producto['id\_media'] === $foto['id']): echo "selected"; endif; ?> >*

*<?php echo $foto['nombre\_archivo'] ?></option>*

*<?php endif; ?>*

*<?php endforeach; ?>*

*</select>*

*</div>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-5">*

*<div class="form-group">*

*<label for="cantidad\_producto">Cantidad</label>*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-shopping-cart"></i>*

*</span>*

*<input type="number" min="0" class="form-control" name="cantidad\_producto" value="<?php echo remover\_basura($producto['cantidad']); ?>" required="">*

*</div>*

*</div>*

*</div>*

*<div class="col-md-4">*

*<div class="form-group">*

*<label for="precio">Precio de compra</label>*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-usd"></i>*

*</span>*

*<?php*

*$precio\_compra = remover\_basura($producto['precio\_compra']);*

*$precio\_compra\_decimal = $precio\_compra;*

*$precio\_compra = intval($precio\_compra);*

*$precio\_compra\_decimal -= $precio\_compra;*

*$precio\_compra\_decimal \*= 100;*

*$precio\_compra\_decimal = intval($precio\_compra\_decimal);*

*?>*

*<input type="number" class="form-control" name="precio\_compra" value="<?php echo $precio\_compra;?>" required="">*

*</div>*

*</div>*

*</div>*

*<span style="padding-bottom:10px" class="venta-compra col-xs-12 form-group input-group-addon col-md-1">*

*<i>.</i>*

*</span>*

*<div class="form-group col-md-2 clase">*

*<input type="number" min="0" max="99" class="form-control" name="compra\_decimal" placeholder="Decimales" required="" value="<?php echo $precio\_compra\_decimal;?>">*

*</div>*

*<div class="col-md-4">*

*<div class="form-group">*

*<label for="venta">Precio de venta</label>*

*<div class="input-group">*

*<span class="input-group-addon">*

*<i class="glyphicon glyphicon-usd"></i>*

*</span>*

*<?php*

*$precio\_venta = remover\_basura($producto['precio\_venta']);*

*$precio\_venta\_decimal = $precio\_venta;*

*$precio\_venta = intval($precio\_venta);*

*$precio\_venta\_decimal -= $precio\_venta;*

*$precio\_venta\_decimal \*= 100;*

*$precio\_venta\_decimal = intval($precio\_venta\_decimal);*

*?>*

*<input type="number" class="form-control" name="precio\_venta" value="<?php echo $precio\_venta;?>" required="">*

*</div>*

*</div>*

*</div>*

*<span style="padding-bottom:10px" class="venta-decimal form-group col-xs-12 input-group-addon col-md-1">*

*<i>.</i>*

*</span>*

*<div class="col-md-2 form-group clase">*

*<input type="number" min="0" max="99" class="form-control" name="precio\_decimal" placeholder="Decimales" required="" value="<?php echo $precio\_venta\_decimal;?>">*

*</div>*

*</div>*

*<button type="submit" name="producto" class="btn btn-danger">Actualizar</button>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="libs/js/editar\_producto.js"></script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 47 se muestra el resultado de la codificación del código de arriba.

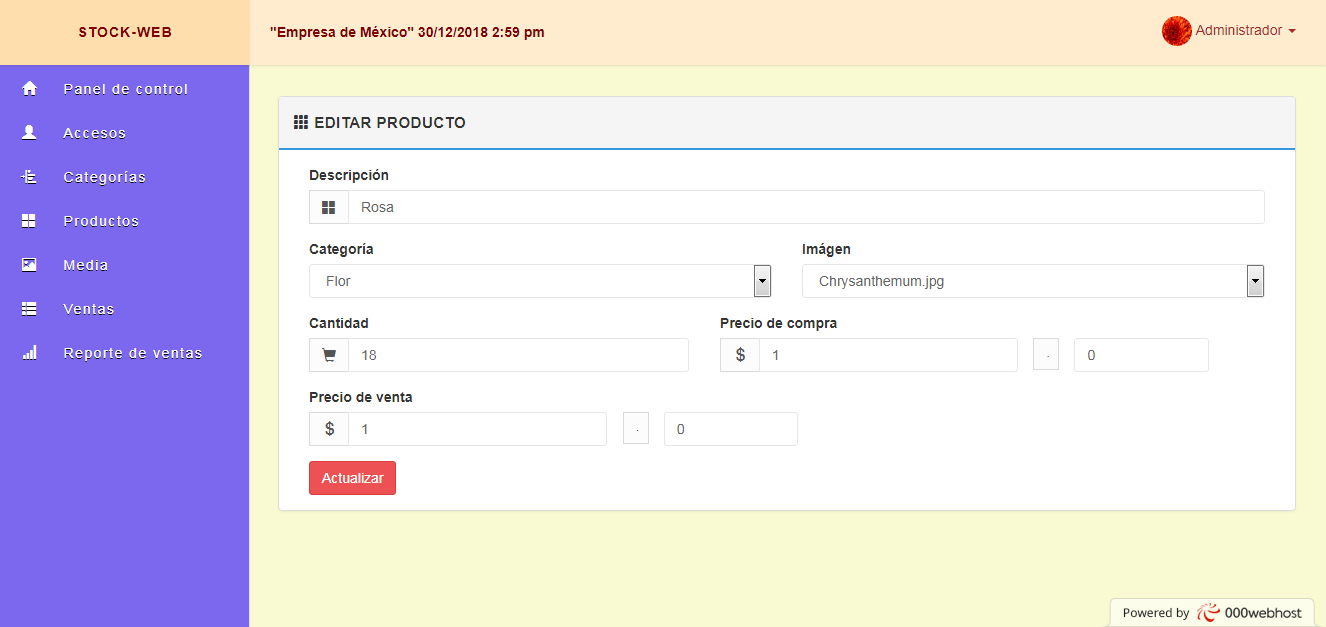


Ilustración 47 Sección para editar producto

## Sección para editar usuario

En esta vista se observa el formulario para editar un usuario.

El formulario cuenta con campos para agregar el nombre completo de la persona dueña de la cuenta, el nombre de usuario, la contraseña y el rol de usuario.

Al editar el nombre de usuario, no debe existir otro con nombre igual.

Al editar el rol de usuario, este podrá tener como valor administrador, especial o estándar.

Al dar clic en el botón de editar usuario, deberá editar el usuario.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Editar usuario';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$editar\_usuario = encontrar\_por\_id\_usuario((int)$\_GET['id']);*

*$grupos = encontrar\_todos('grupo');*

*if(!$editar\_usuario) {*

*$sesion->mensaje("d","Id de usuario no encontrado.");*

*redirigir('usuarios.php');*

*}*

*if(isset($\_POST['actualizar'])) {*

*$campos\_requeridos = array('nombre','usuario','nivel');*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)){*

*$id = (int)$editar\_usuario['id'];*

*$nombre = remover\_basura($db->escape($\_POST['nombre']));*

*$nombre\_usuario = remover\_basura($db->escape($\_POST['usuario']));*

*$nivel = (int)$db->escape($\_POST['nivel']);*

*$estado = remover\_basura($db->escape($\_POST['estado']));*

*$niveles\_grupos = encontrar\_id\_grupo\_usuario($nivel);*

*$comprobar = encontrar\_nombre\_usuarios($nombre\_usuario);*

*$usuarioActual = encontrar\_por\_id("usuario", $\_GET['id']);*

*if($comprobar != $usuarioActual['username'] && $comprobar != null) {*

*$sesion->mensaje('d',"Ya hay un nombre de usuario igual. ");*

*redirigir('editar\_usuario.php?id='.(int)$editar\_usuario['id'], false);*

*}*

*$sql = "UPDATE usuario SET nombre = '{$nombre}', username = '{$nombre\_usuario}', id\_grupo\_usuario = {$niveles\_grupos}, estado = {$estado} WHERE id = {$db->escape($id)}";*

*$resultado = $db->query($sql);*

*if($resultado && $db->affected\_rows() === 1) {*

*$sesion->mensaje('s',"Cuenta actualizada. ");*

*redirigir('usuarios.php?id='.(int)$editar\_usuario['id'], false);*

*} else {*

*redirigir('usuarios.php', false);*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('editar\_usuario.php?id='.(int)$editar\_usuario['id'],false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12"> <?php echo mostrar\_mensaje($mensaje); ?> </div>*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*Actualiza cuenta <?php echo remover\_basura(ucwords($editar\_usuario['nombre'])); ?>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<form method="post" action="editar\_usuario.php?id=<?php echo (int)$editar\_usuario['id'];?>" class="clearfix">*

*<div class="form-group">*

*<label for="nombre" class="control-label">Nombre(s)</label>*

*<input type="name" required="" class="form-control" name="nombre" value="<?php echo remover\_basura(ucwords($editar\_usuario['nombre\_usuario'])); ?>">*

*</div>*

*<div class="form-group">*

*<label for="usuario" class="control-label">Usuario</label>*

*<input type="text" required="" class="form-control" name="usuario" value="<?php echo remover\_basura(ucwords($editar\_usuario['username'])); ?>">*

*</div>*

*<div class="form-group">*

*<label for="nivel">Rol de usuario</label>*

*<?php if($editar\_usuario['id'] !== '1'): ?>*

*<select class="form-control" name="nivel" required="">*

*<?php foreach($grupos as $grupo ):?>*

*<option*

*<?php if ($grupo['nivel'] === $editar\_usuario['nivel']) {*

*echo 'selected="selected"';*

*}*

*?> value="<?php echo $grupo['nivel'];?>"><?php echo ucwords($grupo['nombre']);?></option>*

*<?php endforeach;?>*

*</select>*

*<?php else: ?>*

*<?php foreach ($grupos as $grupo):?>*

*<?php if ($grupo['nivel'] === $editar\_usuario['nivel']) :?>*

*<input type="hidden" name="nivel" value="<?php echo $grupo['nivel']; ?>" />*

*<?php endif;?>*

*<?php endforeach;?>*

*<select class="form-control" name="nivel" disabled required="">*

*<?php foreach ($grupos as $grupo):?>*

*<option*

*<?php if ($grupo['nivel'] === $editar\_usuario['nivel']) {*

*echo 'selected="selected"';*

*}*

*?> value="<?php echo $grupo['nivel'];?>"><?php echo ucwords($grupo['nombre']);?></option>*

*<?php endforeach;?>*

*</select>*

*<?php endif;?>*

*</div>*

*<div class="form-group">*

*<label for="estado">Estado</label>*

*<?php if($editar\_usuario['id'] !== '1'): ?>*

*<select class="form-control" name="estado" required="">*

*<option <?php if ($editar\_usuario['estado'] === '1') {*

*echo 'selected="selected"';*

*}*

*?>value="1">Activo</option>*

*<option <?php if ($editar\_usuario['estado'] === '0') {*

*echo 'selected="selected"';*

*}*

*?> value="0">Inactivo</option>*

*</select>*

*<?php else: ?>*

*<?php foreach ($grupos as $grupo):?>*

*<?php if ($grupo['nivel'] === $editar\_usuario['nivel']) :?>*

*<input type="hidden" name="estado" value="<?php echo $editar\_usuario['estado']; ?>" />*

*<?php endif;?>*

*<?php endforeach;?>*

*<select class="form-control" name="estado" disabled="" required="">*

*<option <?php if ($editar\_usuario['estado'] === '1') {*

*echo 'selected="selected"';*

*}*

*?>value="1">Activo</option>*

*<option <?php if ($editar\_usuario['estado'] === '0') {*

*echo 'selected="selected"';*

*}*

*?> value="0">Inactivo</option>*

*</select>*

*<?php endif;?>*

*</div>*

*<div class="form-group clearfix">*

*<button type="submit" name="actualizar" class="btn btn-info">Actualizar</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 48 se muestra el resultado de la codificación del código de arriba.

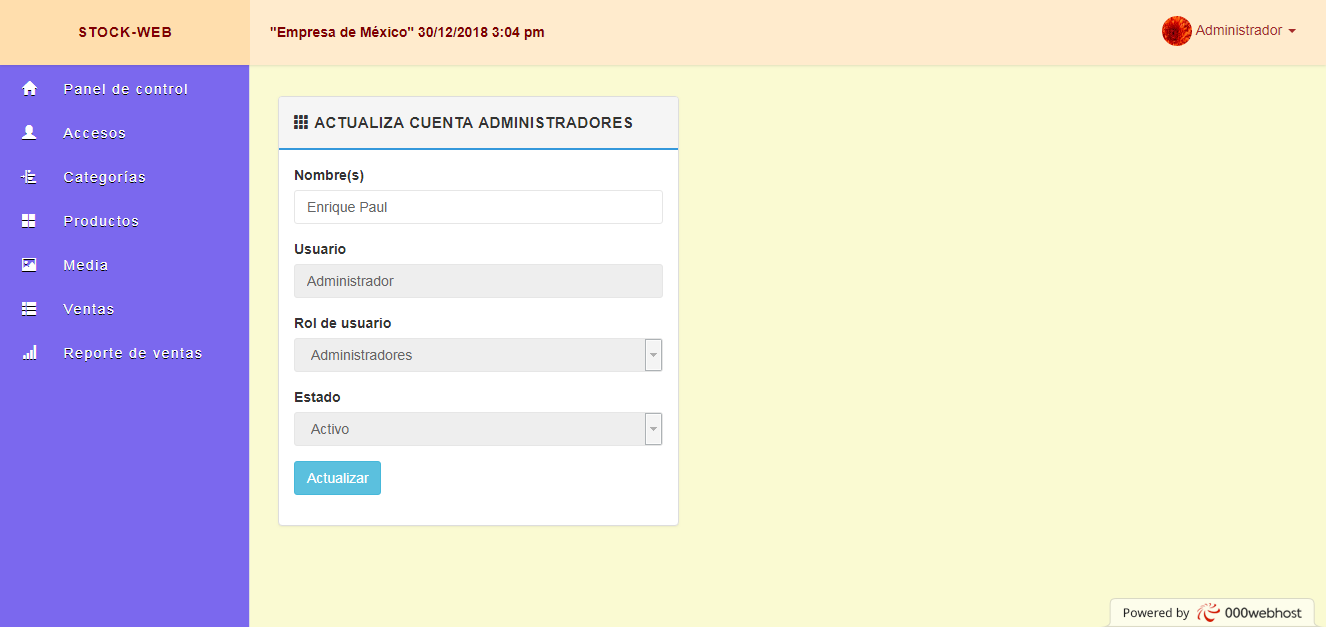


Ilustración 48 Sección para editar usuario

## Sección para editar venta

En esta vista se observa el formulario para editar una venta.

El formulario cuenta con un campo para editar el nombre del producto a vender, además de la fecha en que se efectuó la venta.

Al dar clic en el botón de editar la venta, deberá editar la venta.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Editar venta';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$venta = encontrar\_por\_id('venta',(int)$\_GET['id']);*

*if(!$venta) {*

*$sesion->mensaje("d","Id de producto no encontrada.");*

*redirigir('ventas.php');*

*}*

*$producto = encontrar\_por\_id('producto\_respaldo',$venta['id\_producto']);*

*if(isset($\_POST['actualizar\_venta'])) {*

*$campos\_requeridos = array('titulo','cantidad','precio','totales', 'fecha');*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)) {*

*$id\_producto = $db->escape((int)$producto['id']);*

*$cantidad\_venta = $db->escape((int)$\_POST['cantidad']);*

*$total\_venta = $db->escape($\_POST['totales']);*

*$fecha = $db->escape($\_POST['fecha']);*

*$venta\_fecha = date("Y-m-d", strtotime($fecha));*

*$sql = "UPDATE venta SET";*

*$sql .= " id\_producto= '{$id\_producto}',cantidad={$cantidad\_venta},precio='{$total\_venta}',fecha='{$venta\_fecha}'";*

*$sql .= " WHERE id ='{$venta['id']}'";*

*$resultado = $db->query($sql);*

*if($resultado && $db->affected\_rows() === 1) {*

*//actualizar\_cantidad\_producto($cantidad\_venta,$id\_producto);*

*$sesion->mensaje('s',"Venta actualizada.");*

*redirigir('ventas.php', false);*

*} else {*

*redirigir('ventas.php', false);*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*redirigir('editar\_venta.php?id='.(int)$venta['id'],false);*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-6">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong style="vertical-align:middle">*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Editar venta</span>*

*</strong>*

*<div class="pull-right">*

*<a href="ventas.php" class="btn btn-primary btn-xs">Mostrar todas las ventas</a>*

*</div>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered">*

*<thead>*

*<th> Título </th>*

*<th> Cantidad </th>*

*<th> Precio </th>*

*<th> Total </th>*

*<th> Fecha </th>*

*<th> Acción </th>*

*</thead>*

*<tbody id="product\_info">*

*<tr>*

*<form method="POST" action="editar\_venta.php?id=<?php echo (int)$venta['id']; ?>">*

*<td id="s\_name">*

*<input type="hidden" name="titulo" value="<?php echo remover\_basura($producto['nombre']); ?>" />*

*<input type="text" disabled="" class="form-control" id="sug\_input" name="titulo" value="<?php echo remover\_basura($producto['nombre']); ?>">*

*<div id="result" class="list-group"></div>*

*</td>*

*<td id="s\_qty">*

*<input type="hidden" name="cantidad" value="<?php echo (int)$venta['cantidad']; ?>" />*

*<input type="number" disabled="" class="form-control" name="cantidad" value="<?php echo (int)$venta['cantidad']; ?>" min="1" max="<?php echo (int)$producto['cantidad']; ?>">*

*</td>*

*<td id="s\_price">*

*<input type="hidden" name="precio" value="<?php echo remover\_basura($producto['precio\_venta']); ?>" />*

*<input disabled="" type="text" class="form-control" name="precio" value="<?php echo remover\_basura($producto['precio\_venta']); ?>" >*

*</td>*

*<td>*

*<input type="hidden" name="totales" value="<?php echo remover\_basura($venta['precio']); ?>" />*

*<input disabled type="text" class="form-control" name="totales" value="<?php echo remover\_basura($venta['precio']); ?>">*

*</td>*

*<td id="s\_date">*

*<input type="text" required="" class="form-control datepicker" name="fecha" data-date-format="" value="<?php echo remover\_basura($venta['fecha']); ?>">*

*</td>*

*<td>*

*<button type="submit" name="actualizar\_venta" class="btn btn-primary">Actualizar venta</button>*

*</td>*

*</form>*

*</tr>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 49 se muestra el resultado de la codificación del código de arriba.

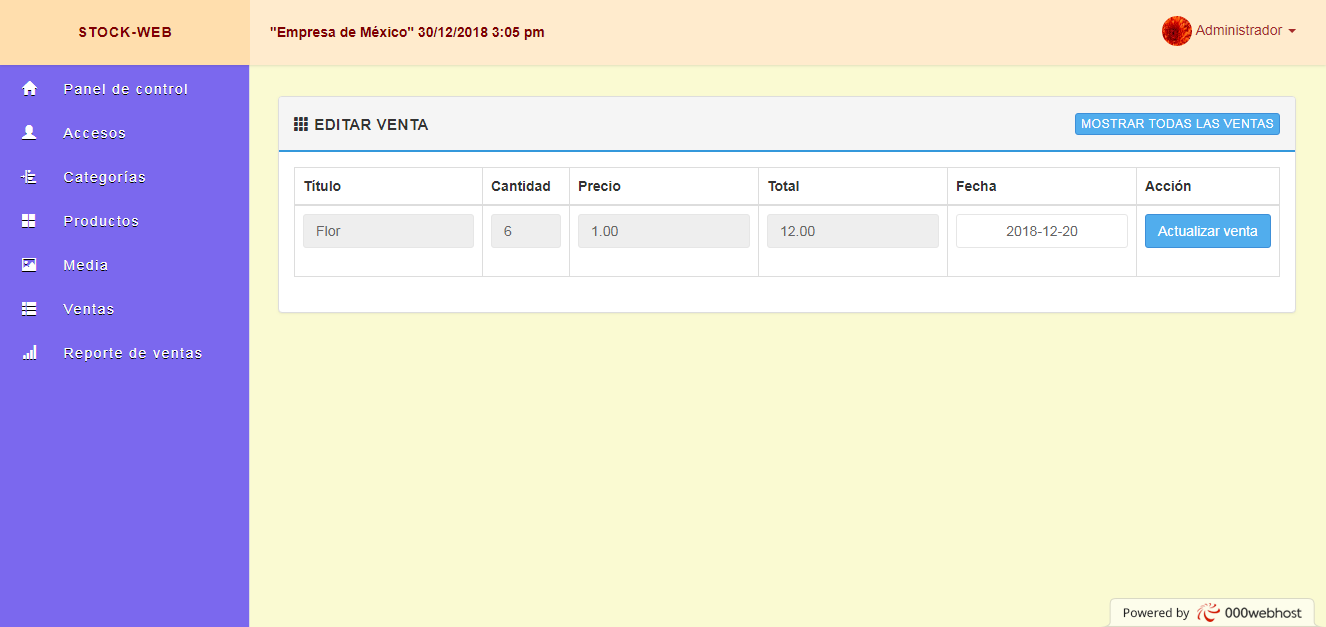


Ilustración 49 Sección para editar venta

## Archivo para eliminar un producto

En este archivo *“eliminar\_producto.php”* se borra el producto correspondiente por medio de su identificador y se manda a otra sección para poder ser recuperado después. Primero se incluye el archivo *“cargar.php”* en el directorio *“incluye”*. Se checa si hay una sesión activa o se direcciona a la página principal. Primero se checa si el nombre del producto a eliminar fue recibido por medio de POST. Se procede a crear una variable *“$producto“* que tendrá como valor lo que devuelva la función *“encontrar\_por\_id“* que recibirá como parámetro lo recibido por POST. Se procede a checar si *“$producto“* tiene algún valor, si hay coincidencia muestra todas las coincidencias en forma de lista y caso contrario mostrará el mensaje “Producto no encontrado” en forma de lista. Se borra el producto

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$producto = encontrar\_por\_id('producto',(int)$\_GET['id']);*

*if(!$producto) {*

*$sesion->mensaje("d","Id de producto pérdida.");*

*redirigir('producto.php');*

*}*

*$borrar\_id = borrar\_por\_id('producto',(int)$producto['id']);*

*if($borrar\_id){*

*$sesion->mensaje("s","El producto fue eliminado.");*

*redirigir('producto.php');*

*} else {*

*$sesion->mensaje("d","La eliminación falló.");*

*redirigir('producto.php');*

*}*

## Archivo para eliminar un producto definitivamente

En este archivo *“eliminar\_producto\_definitivamente.php”* se borra el producto correspondiente definitivamente por medio de su identificador. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se establece el nivel de usuario para este archivo *“pagina\_requiere\_nivel“*; en este caso es de estándar a administrador. Se procede a crear una variable *“$borrar\_id“* que tendrá todos los datos del usuario por medio de su identificador pasado a este archivo por medio de GET; esto se realiza por medio de la función *“encontrar\_por\_id“* donde esta recibe el nombre de la tabla usuario y el identificador del usuario. Se procede a checar si *“$borrar\_id“* tiene algún valor, si no es el caso muestra un mensaje “Falta id de usuario” por medio del objeto *“sesión“* por medio de su método *“mensaje“* y finalmente redirige a la sección de usuarios por medio de la función *“redirigir“*. Si *“$borrar\_id“* tiene un valor se crea una variable *“borrar\_id“* que guardará el resultado de intentar borrar el usuario por medio de *“borrar\_por\_id“*. Finalmente se checa el valor de *“borrar\_id“*, si es verdadero se muestra el mensaje “Usuario eliminado” y se redirige a la sección de grupos por medio de las funciones *“mensaje “*y *“redirigir; si es falso “*se muestra el mensaje “La eliminación falló” y se redirige a la sección de usuarios por medio de las funciones *“mensaje “*y *“redirigir“*.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$borrar\_id = borrar\_por\_id('producto\_eliminado',(int)$\_GET['id']);*

*if($borrar\_id){*

*$sesion->mensaje("s","El producto fue eliminado definitivamente.");*

*redirigir('productos\_eliminados.php');*

*} else {*

*$sesion->mensaje("d","La eliminación falló.");*

*redirigir('productos\_eliminados.php');*

*}*

## Sección de grupos

En esta vista se observa un panel con la lista de grupos registrados con opciones de editarlos o eliminarlos.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Lista de grupos';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$todos\_grupos = encontrar\_todos('grupo');*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong style="vertical-align:middle;">*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Grupos</span>*

*</strong>*

*<a href="agregar\_grupo.php" class="btn btn-info pull-right btn-xs"> Agregar grupo</a>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered">*

*<thead>*

*<tr>*

*<th class="text-center" style="width:50px;vertical-align:middle">#</th>*

*<th class="text-center" style="vertical-align:middle">Nombre del grupo</th>*

*<th class="text-center" style="width:20%;vertical-align:middle">Nivel del grupo</th>*

*<th class="text-center" style="width:15%;vertical-align:middle">Estado</th>*

*<th class="text-center" style="width:100px;vertical-align:middle">Acciones</th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach($todos\_grupos as $grupo): ?>*

*<tr>*

*<td class="text-center" style="vertical-align:middle"><?php echo contar\_id();?></td>*

*<td><?php echo remover\_basura(ucwords($grupo['nombre']))?></td>*

*<td class="text-center" style="vertical-align:middle">*

*<?php echo remover\_basura(ucwords($grupo['nivel']))?>*

*</td>*

*<td class="text-center" style="vertical-align:middle">*

*<?php if($grupo['estado\_actual'] === '1'): ?>*

*<span class="label label-success"><?php echo "Activo"; ?></span>*

*<?php else: ?>*

*<span class="label label-danger"><?php echo "Inactivo"; ?></span>*

*<?php endif;?>*

*</td>*

*<td class="text-center" style="vertical-align:middle">*

*<div class="btn-group">*

*<a href="editar\_grupo.php?id=<?php echo (int)$grupo['id'];?>" class="btn btn-xs btn-warning" data-toggle="tooltip" title="Editar">*

*<i class="glyphicon glyphicon-pencil"></i>*

*</a>*

*<?php if($grupo['nivel'] !== '1'): ?>*

*<a href="borrar\_grupo.php?id=<?php echo (int)$grupo['id'];?>" class="btn btn-xs btn-danger" data-toggle="tooltip" title="Eliminar">*

*<i class="glyphicon glyphicon-remove"></i>*

*</a>*

*<?php endif;?>*

*</div>*

*</td>*

*</tr>*

*<?php endforeach;?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 50 se muestra el resultado de la codificación del código de arriba.

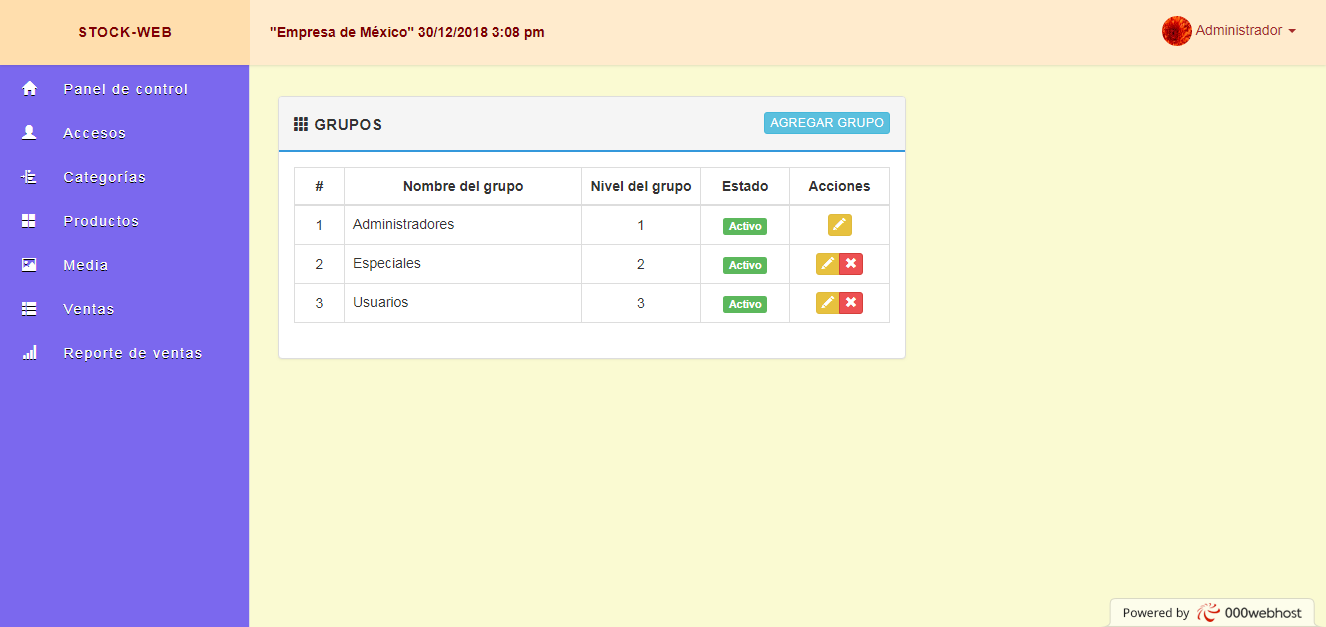


Ilustración 50 Sección de grupos

## Sección principal

En esta vista se observa el formulario para iniciar sesión con los campos de nombre de usuario y contraseña con un botón para iniciar sesión.

Si falla el inicio de sesión, se mostrará un mensaje de acuerdo con la situación.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*ob\_start();*

*require\_once('incluye/cargar.php');*

*if($sesion->estaLogeado(true)) { redirigir('casa.php', false);}*

*$usuario = usuario\_actual();*

*$empresa = empresa();*

*$tema = tema();*

*?>*

*<!DOCTYPE html>*

*<html lang="es">*

*<head>*

*<meta charset="UTF-8">*

*<title>*

*<?php*

*if (!empty($titulo\_pagina)) {*

*echo remover\_basura($titulo\_pagina);*

*} elseif (!empty($usuario)) {*

*echo ucfirst($usuario['nombre']);*

*} else {*

*echo "Sistema de inventario";*

*}*

*?>*

*</title>*

*<link rel="shortcut icon" type="image/x-icon" href="cargas/icono/stock.ico" />*

*<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css"/>*

*<?php*

*$numero = intval($tema['id\_tema']);*

*switch ($numero) {*

*case 1:*

*echo "<link rel='stylesheet' href='libs/css/main.css' />";*

*break;*

*case 2:*

*echo "<link rel='stylesheet' href='libs/css/pastel.css' />";*

*break;*

*default:*

*break;*

*}*

*?>*

*<link rel="stylesheet" href="http://code.jquery.com/ui/1.10.1/themes/base/jquery-ui.css" />*

*<script src="http://code.jquery.com/jquery-1.9.1.js"></script>*

*<script src="http://code.jquery.com/ui/1.10.1/jquery-ui.js"></script>*

*<script src="libs/js/fecha.js"></script>*

*<script>*

*$(function date() {*

*$.datepicker.setDefaults($.datepicker.regional["es"]);*

*});*

*</script>*

*</head>*

*<body>*

*<?php if ($sesion->estaLogeado(true)): ?>*

*<header id="header">*

*<div class="logo pull-left"> Sistema de inventario </div>*

*<div class="header-content">*

*<div class="header-date pull-left">*

*<?php if($empresa['logo'] !== ''): ?>*

*<img src="cargas/logo/<?php echo $empresa['logo']; ?>" width="50" height="50" alt="logo"/>*

*<?php endif;?>*

*<?php if($empresa['lema'] !== ''): ?>*

*<strong><?php echo '"' . $empresa['lema'] . '"'; ?></strong>*

*<?php endif;?>*

*<strong><?php date\_default\_timezone\_set('America/Monterrey'); echo date("d/m/Y g:i a");?></strong>*

*</div>*

*<div class="pull-right clearfix">*

*<ul class="info-menu list-inline list-unstyled">*

*<li class="profile">*

*<a href="#" data-toggle="dropdown" class="toggle" aria-expanded="false">*

*<img src="cargas/usuarios/<?php echo $usuario['imagen'];?>" alt="user-image" class="img-circle img-inline">*

*<span><?php echo remover\_basura(ucfirst($usuario['nombre'])); ?> <i class="caret"></i></span>*

*</a>*

*<ul class="dropdown-menu">*

*<li>*

*<a href="perfil.php?id=<?php echo (int)$usuario['id'];?>">*

*<i class="glyphicon glyphicon-user"></i>*

*Perfil*

*</a>*

*</li>*

*<li>*

*<a href="editar\_cuenta.php">*

*<i class="glyphicon glyphicon-cog"></i>*

*Configuración*

*</a>*

*</li>*

*<li class="last">*

*<a href="cerrar\_sesion.php">*

*<i class="glyphicon glyphicon-off"></i>*

*Salir*

*</a>*

*</li>*

*</ul>*

*</li>*

*</ul>*

*</div>*

*</div>*

*</header>*

*<div class="sidebar">*

*<?php if($usuario['nivel\_grupo'] === '1'): ?>*

*<?php include\_once('menu\_administrador.php');?>*

*<?php elseif($usuario['nivel\_grupo'] === '2'): ?>*

*<?php include\_once('menu\_especial.php');?>*

*<?php elseif($usuario['nivel\_grupo'] === '3'): ?>*

*<?php include\_once('menu\_usuario.php');?>*

*<?php endif;?>*

*</div>*

*<?php endif;?>*

*<?php*

*if($usuario['imagen\_fondo'] != "") {*

*echo "<img style='position: relative; height: 75em' src='cargas/fondos/$usuario[imagen\_fondo]' width='100%' />";*

*}*

*?>*

*<link rel='stylesheet' href='libs/css/main.css' />*

*<div style="padding-top:70px">*

*<div class="login-page" id="inicio-sesion" style="padding-bottom:20px">*

*<div class="text-center">*

*<h1>Bienvenido</h1>*

*<p>Iniciar sesión </p>*

*</div>*

*<div id="resultado"></div>*

*<div id="valor"></div>*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*<form method="post" class="clearfix" id="formulario">*

*<div class="form-group">*

*<label for="usuario" class="control-label">Usuario</label>*

*<input id="usuario" autofocus="" type="name" class="form-control" name="usuario" placeholder="Usuario" required="">*

*</div>*

*<div class="form-group">*

*<label for="Password" class="control-label">Contraseña</label>*

*<input id="password" type="password" name= "password" class="form-control" placeholder="Contraseña" required>*

*</div>*

*<div class="form-group">*

*<button type="submit" class="btn btn-info pull-right">Entrar</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*<script src="libs/js/autenticacion.js"></script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 51 se muestra el resultado de la codificación del código de arriba.



Ilustración 51 Sección de inicio de sesión

## Sección de imágenes de productos

En esta vista se observa el formulario para subir imágenes de productos.

Se suben imágenes por medio de un botón para seleccionar la imagen.

Se pueden borrar imágenes por medio del botón ubicado en el lado extremo derecho de cada imagen.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Lista de imágenes';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$media\_files = encontrar\_todos('media');*

*if(isset($\_POST['submit'])) {*

*$foto = new Media();*

*$foto->subir($\_FILES['file\_upload']);*

*if($foto->proceso\_media()){*

*$sesion->mensaje('s','Imagen subida al servidor.');*

*redirigir('media.php');*

*} else{*

*$sesion->mensaje('d',join($foto->errors));*

*redirigir('media.php');*

*}*

*}*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class='row'>*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<span style="vertical-align:middle" class="glyphicon glyphicon-camera"></span>*

*<span style="vertical-align:middle">Imagenes</span>*

*<div class="pull-right">*

*<form class="form-inline" action="media.php" method="POST" enctype="multipart/form-data">*

*<div class="form-group row" style='padding-left:10px;padding-right:10px'>*

*<div class="input-group">*

*<span class="input-group-btn">*

*<input type="file" name="file\_upload" multiple="multiple" class="btn btn-primary btn-file btn-xs" required=""/>*

*</span>*

*<button style="padding-bottom:3px" type="submit" name="submit" class="btn btn-default btn-xs">Subir</button>*

*</div>*

*</div>*

*</form>*

*</div>*

*</div>*

*<div class="panel-body">*

*<table class="table">*

*<thead>*

*<tr>*

*<th style="vertical-align:middle;" class="text-center" style="width: 50px;">#</th>*

*<th style="vertical-align:middle;" class="text-center">Imagen</th>*

*<th style="vertical-align:middle;" class="text-center">Descripción</th>*

*<th style="vertical-align:middle;" class="text-center" style="width: 20%;">Tipo</th>*

*<th style="vertical-align:middle;" class="text-center" style="width: 50px;">Acciones</th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($media\_files as $media\_file): ?>*

*<?php if($media\_file['nombre\_archivo'] != "no\_image.jpg"): ?>*

*<tr class="list-inline">*

*<td style="vertical-align:middle;" class="text-center"><?php echo contar\_id();?></td>*

*<td class="text-center">*

*<img src="cargas/productos/<?php echo $media\_file['nombre\_archivo'];?>" class="img-thumbnail" />*

*</td>*

*<td style="vertical-align:middle;" class="text-center">*

*<?php echo $media\_file['nombre\_archivo'];?>*

*</td>*

*<td style="vertical-align:middle;" class="text-center">*

*<?php echo $media\_file['tipo\_archivo'];?>*

*</td>*

*<td style="vertical-align:middle;" class="text-center">*

*<a href="borrar\_media.php?id=<?php echo (int) $media\_file['id'];?>" class="btn btn-danger btn-xs" title="Eliminar">*

*<span class="glyphicon glyphicon-trash"></span>*

*</a>*

*</td>*

*</tr>*

*<?php endif; ?>*

*<?php endforeach;?>*

*</tbody>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 52 se muestra el resultado de la codificación del código de arriba.

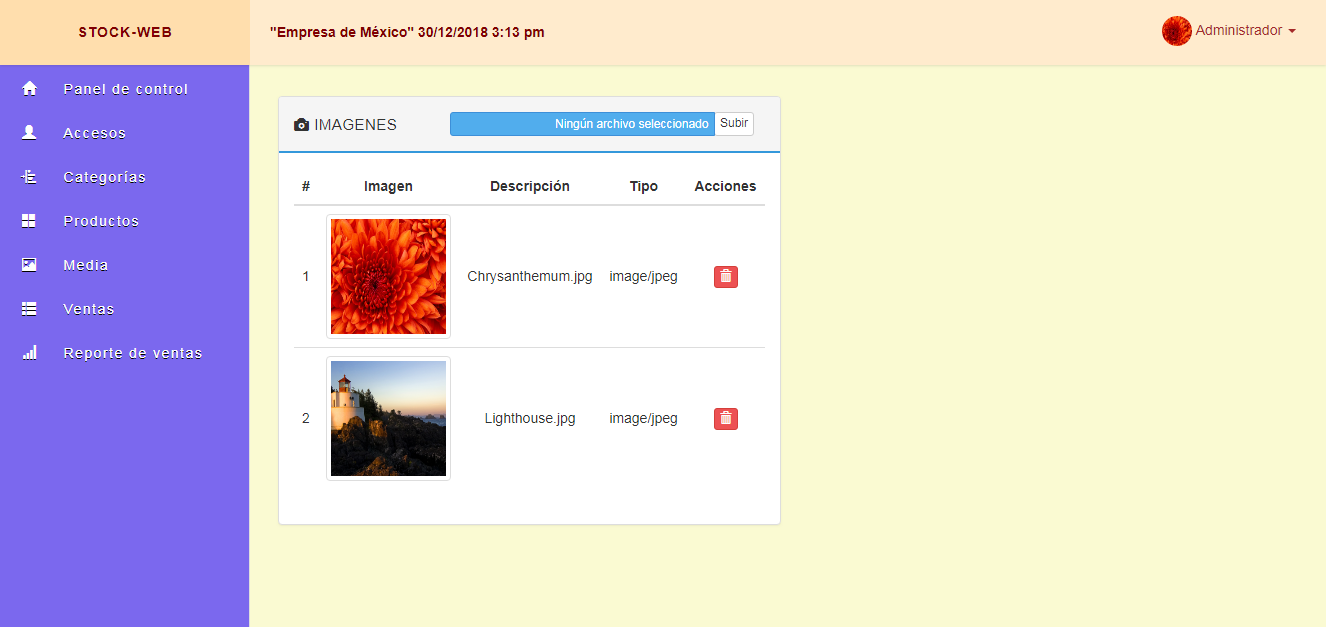


Ilustración 52 Sección de imágenes de productos

## Sección de perfil de usuario

En esta vista se observa la imagen de perfil del usuario con un enlace a la configuración de la cuenta debajo.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Mi perfil';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$id\_usuario = (int)$\_GET['id'];*

*if(empty($id\_usuario)):*

*redirigir('casa.php',false);*

*else:*

*$user\_p = encontrar\_por\_id('usuario',$id\_usuario);*

*endif;*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel profile">*

*<div class="jumbotron text-center bg-red">*

*<img class="img-circle img-size-2" src="cargas/usuarios/<?php echo $user\_p['imagen'];?>" alt="">*

*<h3><?php echo primer\_caracter($user\_p['nombre']); ?></h3>*

*</div>*

*<?php if( $user\_p['id'] === $usuario['id']):?>*

*<ul class="nav nav-pills nav-stacked">*

*<li><a href="editar\_cuenta.php"> <i class="glyphicon glyphicon-edit"></i> Editar perfil</a></li>*

*</ul>*

*<?php endif;?>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 53 se muestra el resultado de la codificación del código de arriba.

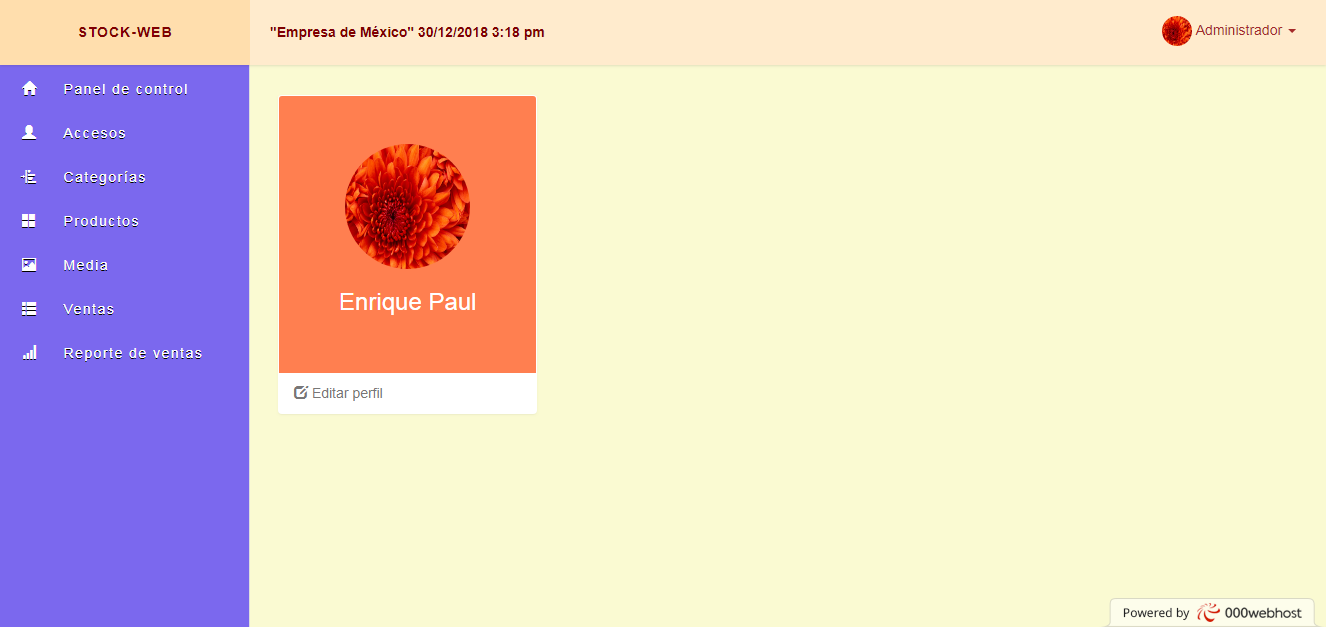


Ilustración 53 Sección de perfil de usuario

## Sección de productos

En esta vista se observa un panel con la lista de productos registrados con opciones de editarlos o eliminarlos.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Lista de productos';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$productos = unir\_tabla\_producto();*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong style="vertical-align:middle;">*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Productos</span>*

*</strong>*

*<div class="pull-right">*

*<a href="agregar\_producto.php" class="btn btn-primary btn-xs">Agregar producto</a>*

*</div>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered">*

*<thead>*

*<tr>*

*<th class="text-center" style="width: 50px;vertical-align:middle;">#</th>*

*<th class="text-center" style="vertical-align:middle;">Imagen</th>*

*<th class="text-center" style="vertical-align:middle;">Descripción</th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Categoría </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Cantidad </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Precio de compra </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Precio de venta </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Fecha actualizado </th>*

*<th class="text-center" style="width: 100px;vertical-align:middle;"> Acciones </th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($productos as $producto):?>*

*<tr>*

*<td class="text-center" style="vertical-align:middle;"><?php echo contar\_id();?></td>*

*<td>*

*<?php if($producto['id\_media'] === NULL): ?>*

*<img class="img-avatar img-circle" src="cargas/productos/no\_image.jpg" alt="">*

*<?php else: ?>*

*<img class="img-avatar img-circle" src="cargas/productos/<?php echo $producto['imagen']; ?>" alt="">*

*<?php endif; ?>*

*</td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['nombre']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['categorie']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['cantidad']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['precio\_compra']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['precio\_venta']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo leer\_fecha($producto['fecha\_actualizacion']); ?></td>*

*<td class="text-center" style="vertical-align:middle;">*

*<div class="btn-group">*

*<a href="editar\_producto.php?id=<?php echo (int)$producto['id'];?>" class="btn btn-info btn-xs" title="Editar" data-toggle="tooltip">*

*<span class="glyphicon glyphicon-edit"></span>*

*</a>*

*<a href="eliminar\_producto.php?id=<?php echo (int)$producto['id'];?>" class="btn btn-danger btn-xs" title="Eliminar" data-toggle="tooltip">*

*<span class="glyphicon glyphicon-trash"></span>*

*</a>*

*</div>*

*</td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 54 se muestra el resultado de la codificación del código de arriba.

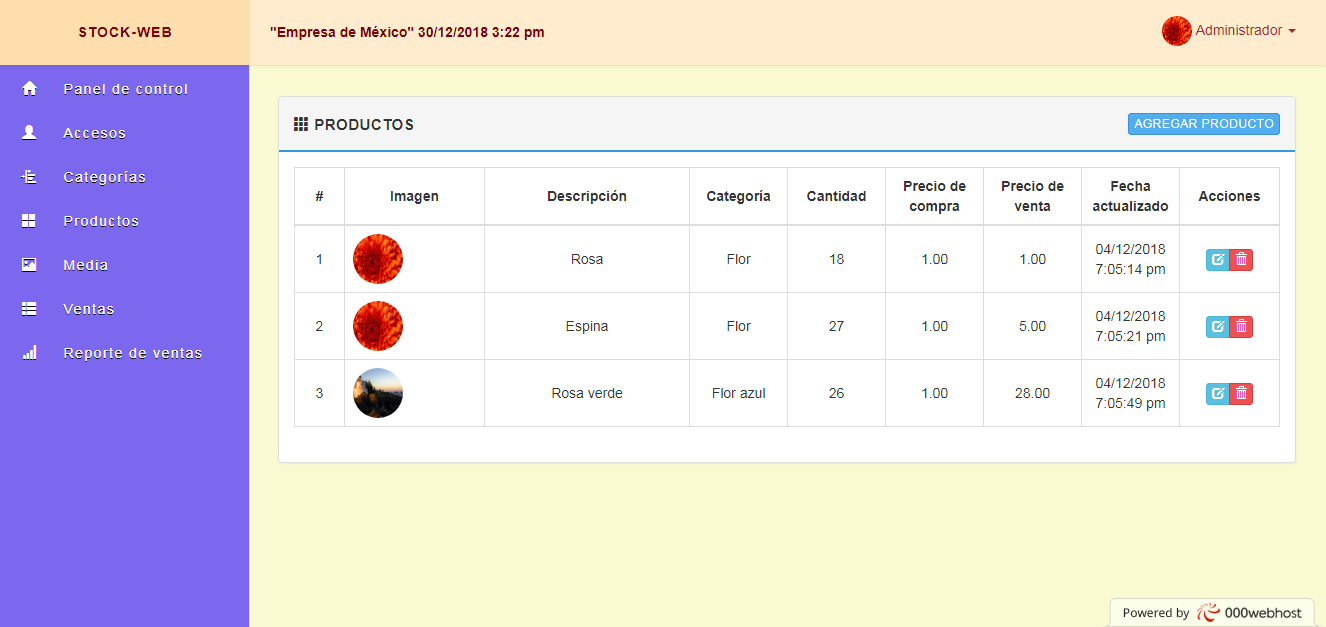


Ilustración 54 Sección de productos

## Sección de productos eliminados

En esta vista se observa un panel con la lista de productos eliminados con opciones de restaurarlos o eliminarlos definitivamente.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Lista de productos';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$productos = unir\_tabla\_productos();*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Productos eliminados</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered">*

*<thead>*

*<tr>*

*<th class="text-center" style="width: 50px;vertical-align:middle;">#</th>*

*<th class="text-center" style="vertical-align:middle;">Imagen</th>*

*<th class="text-center" style="vertical-align:middle;">Descripción</th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Categoría </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Cantidad </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Precio de compra </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Precio de venta </th>*

*<th class="text-center" style="width: 10%;vertical-align:middle;"> Fecha actualizado </th>*

*<th class="text-center" style="width: 100px;vertical-align:middle;"> Acciones </th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($productos as $producto):?>*

*<tr>*

*<td class="text-center" style="vertical-align:middle;"><?php echo contar\_id();?></td>*

*<td>*

*<?php if($producto['id\_media'] === NULL): ?>*

*<img class="img-avatar img-circle" src="cargas/productos/no\_image.jpg" alt="">*

*<?php else: ?>*

*<img class="img-avatar img-circle" src="cargas/productos/<?php echo $producto['imagen']; ?>" alt="">*

*<?php endif; ?>*

*</td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['nombre']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['categorie']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['cantidad']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['precio\_compra']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo remover\_basura($producto['precio\_venta']); ?></td>*

*<td class="text-center" style="vertical-align:middle;"> <?php echo leer\_fecha($producto['fecha\_actualizacion']); ?></td>*

*<td class="text-center" style="vertical-align:middle;">*

*<div class="btn-group">*

*<a href="restaurar\_producto.php?id=<?php echo (int)$producto['id'];?>" class="btn btn-info btn-xs" title="Restaurar" data-toggle="tooltip">*

*<span class="glyphicon glyphicon-repeat"></span>*

*</a>*

*<a href="eliminar\_producto\_definitivamente.php?id=<?php echo (int)$producto['id'];?>" class="btn btn-danger btn-xs" title="Eliminar" data-toggle="tooltip">*

*<span class="glyphicon glyphicon-trash"></span>*

*</a>*

*</div>*

*</td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 55 se muestra el resultado de la codificación del código de arriba.

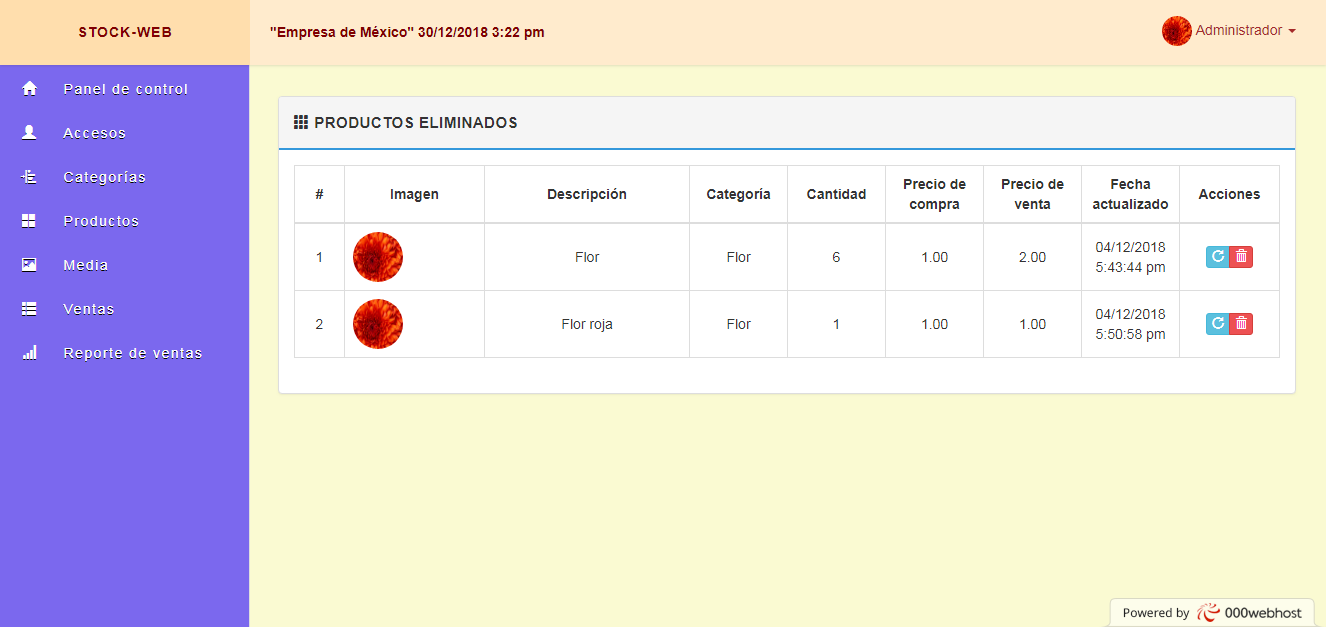


Ilustración 55 Sección de productos eliminados

## Sección de reporte de ventas

En esta vista se observa un formulario para seleccionar la fecha de inicio y final para poder generar el reporte.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Reporte de ventas';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-6">*

*<div class="panel">*

*<div class="panel-heading">*

*</div>*

*<div class="panel-body">*

*<form class="clearfix" method="post" action="reporte\_ventas\_proceso.php">*

*<div class="form-group">*

*<label class="form-label">Rango de fechas</label>*

*<div class="input-group">*

*<input type="text" class="datepicker form-control fecha-inicio" name="fecha-inicio" placeholder="De" autocomplete="off">*

*<span class="input-group-addon"><i class="glyphicon glyphicon-menu-right"></i></span>*

*<input type="text" class="datepicker form-control fecha-fin" name="fecha-fin" placeholder="Hasta" autocomplete="off">*

*</div>*

*</div>*

*<div class="form-group">*

*<button type="submit" name="submit" class="btn btn-primary">Generar reporte</button>*

*</div>*

*</form>*

*</div>*

*</div>*

*</div>*

*</div>*

*<script src="libs/js/moment.js"></script>*

*<script>*

*$(document).ready(function() {*

*$(".fecha-inicio").change(function dateInicio() {*

*var valor = $(".fecha-fin").val();*

*if(valor === '') {*

*var fecha1;*

*var fecha2;*

*var fecha;*

*var diferencia;*

*fecha = $(".fecha-inicio").val();*

*fecha1 = moment(fecha);*

*fecha2 = moment(new Date());*

*diferencia = fecha1.diff(fecha2,'days');*

*if(diferencia > 0) {*

*$(".fecha-fin").datepicker('option', 'minDate', diferencia + 1);*

*} else {*

*$(".fecha-fin").datepicker('option', 'minDate', diferencia);*

*}*

*}*

*});*

*$(".fecha-fin").change(function dateFin() {*

*var valor = $(".fecha-inicio").val();*

*if(valor === '') {*

*var fecha1;*

*var fecha2;*

*var fecha;*

*var diferencia;*

*fecha = $(".fecha-fin").val();*

*fecha1 = moment(fecha);*

*fecha2 = moment(new Date());*

*diferencia = fecha1.diff(fecha2,'days');*

*alert(diferencia);*

*if(diferencia > 0) {*

*$(".fecha-inicio").datepicker('option', 'maxDate', diferencia + 1);*

*} else {*

*$(".fecha-inicio").datepicker('option', 'maxDate', diferencia);*

*}*

*}*

*});*

*});*

*</script>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 56 se muestra el resultado de la codificación del código de arriba.

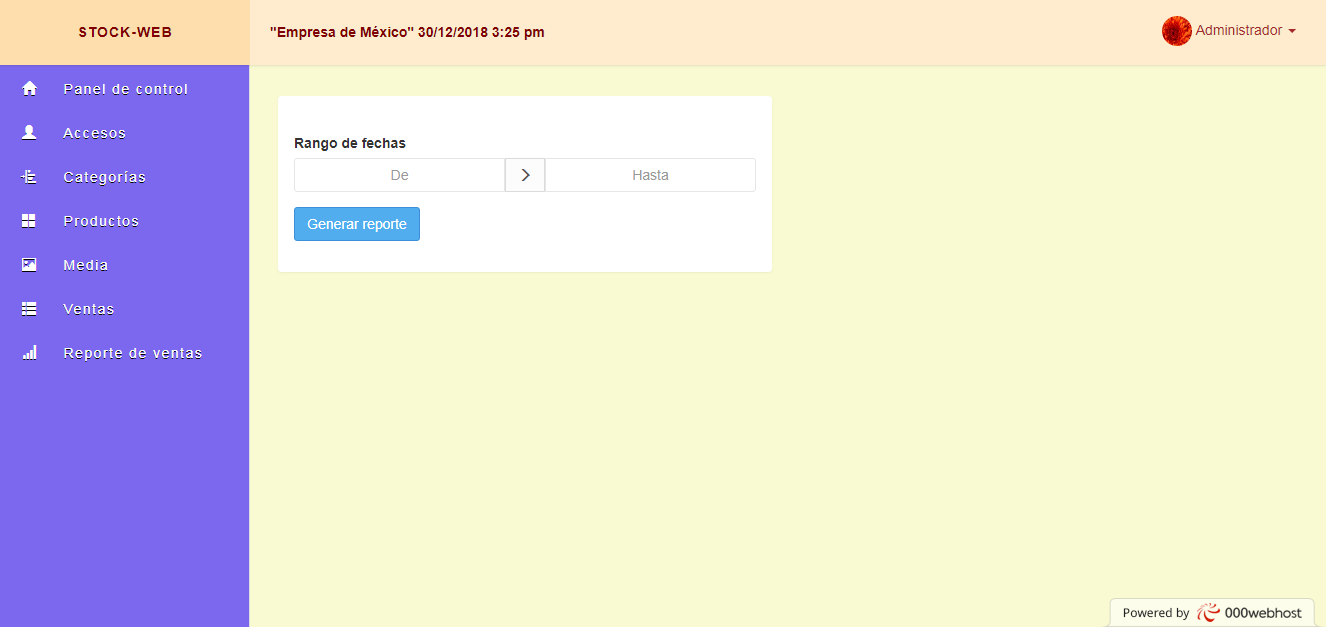


Ilustración 56 Sección de reporte de ventas

## Sección de reporte de ventas generado

En esta vista se observa una tabla con los datos del reporte: fecha, descripción, precio de compra, precio de venta, cantidad total, total, el usuario que registró la venta.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Reporte de ventas';*

*$resultados = '';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$usuario = usuario\_actual();*

*if(isset($\_POST['submit'])){*

*$fechas\_requeridas = array('fecha-inicio','fecha-fin');*

*validar\_campos($fechas\_requeridas);*

*if(empty($errores)):*

*$fecha\_inicio = remover\_basura($db->escape($\_POST['fecha-inicio']));*

*$fecha\_fin = remover\_basura($db->escape($\_POST['fecha-fin']));*

*$resultados = encontrar\_ventas\_por\_fechas($fecha\_inicio,$fecha\_fin);*

*else:*

*$sesion->mensaje("d", $errores);*

*redirigir('reporte\_ventas.php', false);*

*endif;*

*} elseif(isset($\_GET['fecha-inicio'])) {*

*$tipo = $\_GET['tipo'];*

*if(intval($tipo) == 1) {*

*$fecha\_inicio = $\_GET['fecha-inicio'];*

*$fecha\_fin = $fecha\_inicio;*

*$resultados = encontrar\_ventas\_por\_fechas($fecha\_inicio,$fecha\_fin);*

*} elseif(intval($tipo) == 2) {*

*$fecha\_inicio = $\_GET['fecha-inicio'];*

*$fecha\_fin = $\_GET['fecha-fin'];*

*$resultados = encontrar\_ventas\_por\_fechas($fecha\_inicio,$fecha\_fin);*

*} else {*

*$fecha\_inicio = $\_GET['fecha-inicio'];*

*$fecha\_fin = $\_GET['fecha-fin'];*

*$fecha\_inicio = remover\_basura($fecha\_inicio);*

*$fecha\_fin = remover\_basura($fecha\_fin);*

*$resultados = encontrar\_ventas\_por\_fechas($fecha\_inicio,$fecha\_fin);*

*}*

*} else {*

*$sesion->mensaje("d", "Selecciona fechas.");*

*redirigir('reporte\_ventas.php', false);*

*}*

*?>*

*<!doctype html>*

*<html lang="es">*

*<head>*

*<meta http-equiv="Content-Type" content="text/html; charset=utf-8">*

*<title>Reporte de ventas</title>*

*<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.4/css/bootstrap.min.css"/>*

*<style>*

*@media print {*

*html,body {*

*font-size: 9.5pt;*

*margin: 0;*

*padding: 0;*

*}.page-break {*

*page-break-before:always;*

*width: auto;*

*margin: auto;*

*}*

*}*

*.page-break {*

*width: 980px;*

*margin: 0 auto;*

*}*

*.sale-head {*

*margin: 40px 0;*

*text-align: center;*

*}.sale-head h1,.sale-head strong {*

*padding: 10px 20px;*

*display: block;*

*}.sale-head h1 {*

*margin: 0;*

*border-bottom: 1px solid #212121;*

*}.table>thead:first-child>tr:first-child>th {*

*border-top: 1px solid #000;*

*}*

*table thead tr th {*

*text-align: center;*

*border: 1px solid #ededed;*

*}table tbody tr td {*

*vertical-align: middle;*

*}.sale-head,table.table thead tr th,table tbody tr td,table tfoot tr td {*

*border: 1px solid #212121;*

*white-space: nowrap;*

*}.sale-head h1,table thead tr th,table tfoot tr td {*

*background-color: #f8f8f8;*

*}tfoot {*

*color:#000;*

*text-transform: uppercase;*

*font-weight: 500;*

*}*

*</style>*

*</head>*

*<body>*

*<?php if($resultados): ?>*

*<div class="page-break">*

*<div class="sale-head pull-right">*

*<h1>Reporte de ventas</h1>*

*<strong><?php if(isset($fecha\_inicio)){ echo $fecha\_inicio;}?> a <?php if(isset($fecha\_fin)){echo $fecha\_fin;}?> </strong>*

*</div>*

*<table class="table table-border">*

*<thead>*

*<tr>*

*<th>Fecha</th>*

*<th>Descripción</th>*

*<th>Precio de compra</th>*

*<th>Precio de venta</th>*

*<th>Cantidad total</th>*

*<th>TOTAL</th>*

*<th>Registró usuario</th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach($resultados as $resultado): ?>*

*<tr>*

*<td class=""><?php echo remover\_basura($resultado['fecha']);?></td>*

*<td class="desc">*

*<h6><?php echo remover\_basura(ucfirst($resultado['nombre']));?></h6>*

*</td>*

*<td class="text-right"><?php echo remover\_basura($resultado['precio\_compra']);?></td>*

*<td class="text-right"><?php echo remover\_basura($resultado['precio\_venta']);?></td>*

*<td class="text-right"><?php echo remover\_basura($resultado['total\_sales']);?></td>*

*<td class="text-right"><?php echo remover\_basura($resultado['total\_saleing\_price']);?></td>*

*<?php $usuario\_venta = encontrar\_por\_id("usuario", $resultado['id\_usuario']) ?>*

*<td class="text-center"><?php echo $usuario\_venta['username'] ?></td>*

*</tr>*

*<?php endforeach; ?>*

*</tbody>*

*<tfoot>*

*<tr class="text-right">*

*<td colspan="4"></td>*

*<td colspan="1"> Total </td>*

*<td> $*

*<?php echo number\_format(@precio\_total($resultados)[0], 2);?>*

*</td>*

*</tr>*

*<tr class="text-right">*

*<td colspan="4"></td>*

*<td colspan="1">Utilidad</td>*

*<td> $<?php echo number\_format(@precio\_total($resultados)[1], 2);?></td>*

*</tr>*

*</tfoot>*

*</table>*

*</div>*

*<?php*

*/\**

*else:*

*$sesion->mensaje("d", "No se encontraron ventas. ");*

*redirigir('reporte\_ventas.php', false);*

*\*/*

*endif;*

*?>*

*</body>*

*</html>*

*<?php if(isset($db)) { $db->db\_disconnect(); }*

En la ilustración 57 se muestra el resultado de la codificación del código de arriba.

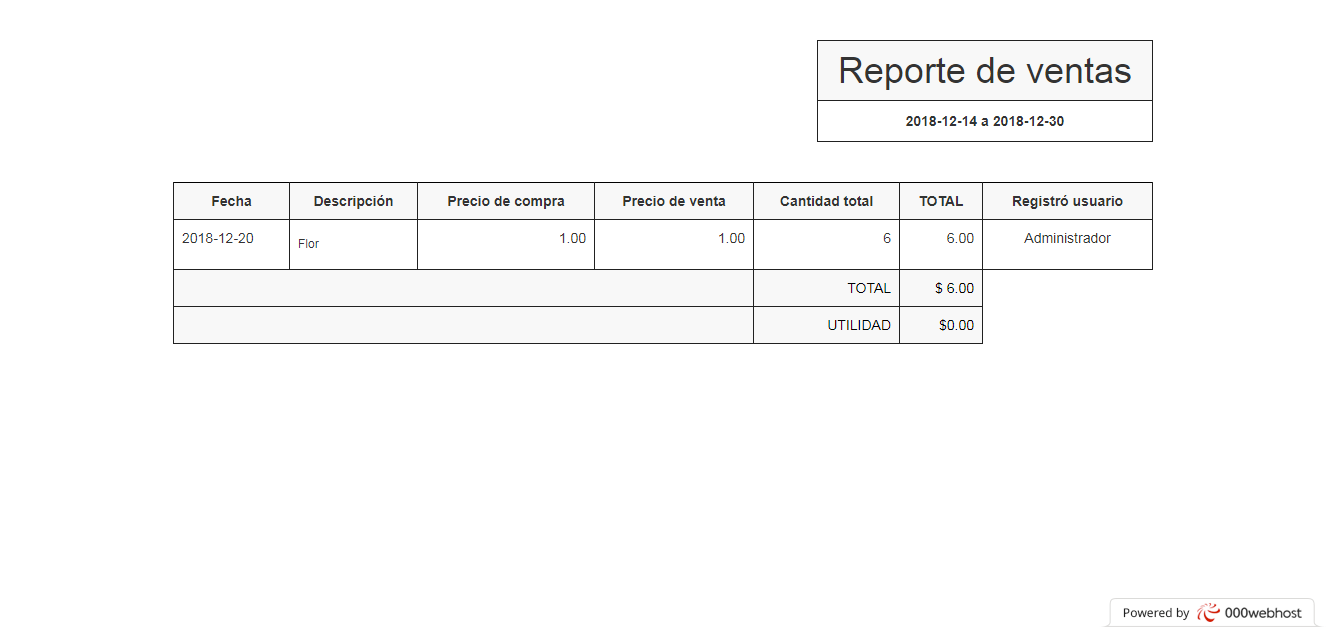


Ilustración 57 Sección de reporte de ventas generado

## Archivo para restaurar un producto

En este archivo *“restaurar\_producto.php”* se restaura el producto correspondiente por medio de su identificador. Primero se carga el archivo cargar.php ubicado en el directorio incluye. Se establece el nivel de usuario para este archivo *“pagina\_requiere\_nivel“*; en este caso es de especial a administrador. Se procede a crear una variable *“producto“* que tendrá todos los datos del producto por medio de su identificador pasado a este archivo por medio de GET; esto se realiza por medio de la función *“encontrar\_por\_id“* donde esta recibe el nombre de la tabla “producto\_eliminado” y el identificador del producto eliminado. Se procede a checar si *“producto“* tiene algún valor, si no es el caso muestra un mensaje “Id de producto perdida” por medio del objeto *“sesión“* por medio de su método *“mensaje“* y finalmente redirige a la sección de productos eliminados por medio de la función *“redirigir“*. Si *“producto“* tiene un valor se crea una variable *“categoria“* que guardará la identificación de la categoría, así como una variable *“media“* que guardará la identificación de la imagen. Se checa si alguna de las 2 variables antes mencionadas tiene algún valor, caso contrario se les asigna un valor adecuado. Se crea una variable *“restore\_id“* que tendrá como valor el resultado que devuelva la función encargada de restaurar el producto *“restaurar\_por\_id“* por medio de las variables *“categoria“*, *“media “*y la clave del producto a restaurar. Finalmente se checa el valor de *“restore\_id“*: si es 0 se muestra el mensaje “La eliminación falló” y se redirige a la sección de productos eliminados por medio de las funciones *“mensaje “*y *“redirigir“*; si es 1 se muestra el mensaje “El producto fue restaurado” y se redirige a la sección de productos por medio de las funciones *“mensaje “*y *“redirigir“*; si es 2 es el mismo mensaje que el caso 0; si es 3 se muestra el mensaje “Ya hay un producto no eliminado con ese nombre” y se redirige a la sección de productos eliminados por medio de las funciones *“mensaje “*y *“redirigir“*.

A continuación, se muestra el código.

*<?php*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(2);*

*$producto = encontrar\_por\_id('producto\_eliminado',(int)$\_GET['id']);*

*if(!$producto) {*

*$sesion->mensaje("d","Id de producto pérdida.");*

*redirigir('productos\_eliminados.php');*

*} else {*

*$categoria = $producto['id\_categoria'];*

*if($categoria == NULL) {*

*$categoria = 10;*

*}*

*$media = $producto['id\_media'];*

*if($media == NULL) {*

*$media = 16;*

*}*

*$restore\_id = restaurar\_por\_id((int)$producto['id'], $categoria, $media);*

*switch ($restore\_id) {*

*case 0:*

*$sesion->mensaje("d","La eliminación falló.");*

*redirigir('productos\_eliminados.php');*

*break;*

*case 1:*

*$sesion->mensaje("s","El producto fue restaurado.");*

*redirigir('producto.php');*

*break;*

*case 2:*

*$sesion->mensaje("d","La eliminación falló.");*

*redirigir('productos\_eliminados.php');*

*case 3:*

*$sesion->mensaje("d","Ya hay un producto no eliminado con ese nombre.");*

*redirigir('productos\_eliminados.php');*

*break;*

*default:*

*break;*

*}*

*}*

## Sección de usuarios

En esta vista se observa un panel con la lista de usuarios registrados con opciones de editarlos o eliminarlos.

Al usuario administrador que esta por defecto se le puede modificar únicamente el nombre más no eliminar al usuario.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Lista de usuarios';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(1);*

*$todos\_usuarios = encontrar\_todos\_usuarios();*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong style="vertical-align:middle;">*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Usuarios</span>*

*</strong>*

*<a href="agregar\_usuario.php" class="btn btn-info pull-right btn-xs">Agregar usuario</a>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered table-striped">*

*<thead>*

*<tr>*

*<th class="text-center" style="width:50px;vertical-align:middle">#</th>*

*<th class="text-center" style="vertical-align:middle">Nombre </th>*

*<th class="text-center" style="vertical-align:middle">Usuario</th>*

*<th style="vertical-align:middle;" class="text-center" style="width: 15%;">Rol de usuario</th>*

*<th style="vertical-align:middle;" class="text-center" style="width: 10%;">Estado</th>*

*<th class="text-center" style="vertical-align:middle;" style="width: 20%;">Último login</th>*

*<th style="vertical-align:middle;" class="text-center" style="width: 100px;">Acciones</th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach($todos\_usuarios as $agregar\_usuario): ?>*

*<tr>*

*<td class="text-center" style="vertical-align:middle"><?php echo contar\_id();?></td>*

*<td class="text-center" style="vertical-align:middle"><?php echo remover\_basura(ucwords($agregar\_usuario['nombre']))?></td>*

*<td class="text-center" style="vertical-align:middle"><?php echo remover\_basura(ucwords($agregar\_usuario['username']))?></td>*

*<td class="text-center" style="vertical-align:middle"><?php echo remover\_basura(ucwords($agregar\_usuario['nombre\_grupo']))?></td>*

*<td class="text-center" style="vertical-align:middle">*

*<?php if($agregar\_usuario['estado'] === '1'): ?>*

*<span class="label label-success"><?php echo "Activo"; ?></span>*

*<?php else: ?>*

*<span class="label label-danger"><?php echo "Inactivo"; ?></span>*

*<?php endif;?>*

*</td>*

*<td class="text-center" style="vertical-align:middle"><?php echo leer\_fecha($agregar\_usuario['ultimo\_login'])?></td>*

*<td class="text-center" style="vertical-align:middle">*

*<div class="btn-group">*

*<a href="editar\_usuario.php?id=<?php echo (int)$agregar\_usuario['id'];?>" class="btn btn-xs btn-warning" data-toggle="tooltip" title="Editar">*

*<i class="glyphicon glyphicon-pencil"></i>*

*</a>*

*<?php if($agregar\_usuario['id'] !== '1'): ?>*

*<a href="borrar\_usuario.php?id=<?php echo (int)$agregar\_usuario['id'];?>" class="btn btn-xs btn-danger" data-toggle="tooltip" title="Eliminar">*

*<i class="glyphicon glyphicon-remove"></i>*

*</a>*

*<?php endif;?>*

*</div>*

*</td>*

*</tr>*

*<?php endforeach;?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 58 se muestra el resultado de la codificación del código de arriba.

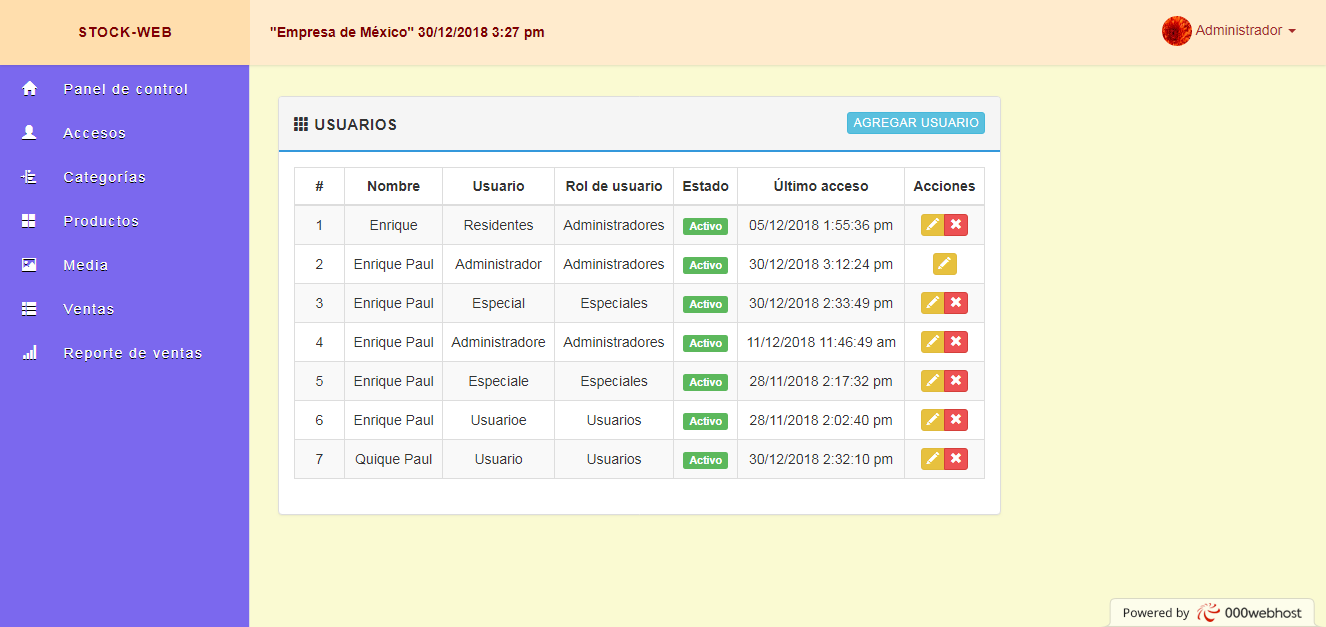


Ilustración 58 Sección de usuarios

## Sección de ventas

En esta vista se observa un panel con la lista de ventas registradas con opciones de editarlas (solo usuario tipo administrador) o eliminarlas.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Lista de ventas';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$ventas = encontrar\_todas\_ventas();*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong style="vertical-align:middle">*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Todas la ventas</span>*

*</strong>*

*<div class="pull-right">*

*<a href="agregar\_venta.php" class="btn btn-primary btn-xs">Agregar venta</a>*

*</div>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered table-striped">*

*<thead>*

*<tr>*

*<th style="vertical-align:middle" class="text-center">#</th>*

*<th style="vertical-align:middle" class="text-center"> Nombre del producto </th>*

*<th style="vertical-align:middle" class="text-center"> Cantidad</th>*

*<th style="vertical-align:middle" class="text-center"> Total </th>*

*<th style="vertical-align:middle" class="text-center"> Fecha </th>*

*<?php if($usuario['nivel'] == 1): ?>*

*<th style="vertical-align:middle" class="text-center"> Acciones </th>*

*<?php endif; ?>*

*<th style="vertical-align:middle" class="text-center"> Registró usuario </th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($ventas as $venta):?>*

*<tr>*

*<td style="vertical-align:middle" class="text-center"><?php echo contar\_id();?></td>*

*<td style="vertical-align:middle" class="text-center"><?php echo remover\_basura($venta['nombre']); ?></td>*

*<td style="vertical-align:middle" class="text-center"><?php echo (int)$venta['cantidad']; ?></td>*

*<td style="vertical-align:middle" class="text-center"><?php echo remover\_basura($venta['precio']); ?></td>*

*<td style="vertical-align:middle" class="text-center"><?php echo $venta['fecha']; ?></td>*

*<?php if($usuario['nivel'] == 1): ?>*

*<td style="vertical-align:middle" class="text-center">*

*<div class="btn-group">*

*<a href="editar\_venta.php?id=<?php echo (int)$venta['id'];?>" class="btn btn-warning btn-xs" title="Editar" data-toggle="tooltip">*

*<span class="glyphicon glyphicon-edit"></span>*

*</a>*

*<a href="borrar\_venta.php?id=<?php echo (int)$venta['id'];?>" class="btn btn-danger btn-xs" title="Borrar" data-toggle="tooltip">*

*<span class="glyphicon glyphicon-trash"></span>*

*</a>*

*</div>*

*</td>*

*<?php endif; ?>*

*<?php $usuario\_venta = encontrar\_por\_id("usuario", $venta['id\_usuario']) ?>*

*<td style="vertical-align:middle" class="text-center"><?php echo $usuario\_venta['username'] ?></td>*

*</tr>*

*<?php endforeach;?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 59 se muestra el resultado de la codificación del código de arriba.



Ilustración 59 Sección de ventas

## Sección de ventas diarias

En esta vista se observa un panel con la lista de ventas diarias registradas.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Venta diaria';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$agno = date('Y');*

*$mes = date('m');*

*date\_default\_timezone\_set('America/Monterrey');*

*$dia = intval(date('j'));*

*$ventas = ventasDiarias($agno,$mes,$dia);*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Venta diaria</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered table-striped">*

*<thead>*

*<tr>*

*<th style="vertical-align: middle" class="text-center" style="width: 50px;">#</th>*

*<th style="vertical-align: middle" class="text-center"> Descripción </th>*

*<th style="vertical-align: middle" class="text-center" style="width: 15%;"> Cantidad vendida</th>*

*<th style="vertical-align: middle" class="text-center" style="width: 15%;"> Total </th>*

*<th style="vertical-align: middle" class="text-center" style="width: 15%;"> Fecha </th>*

*<th style="vertical-align: middle" class="text-center"> Registró usuario </th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($ventas as $venta):?>*

*<tr>*

*<td class="text-center"><?php echo contar\_id();?></td>*

*<td><?php echo remover\_basura($venta['nombre']); ?></td>*

*<td class="text-center"><?php echo (int)$venta['cantidad']; ?></td>*

*<td class="text-center"><?php echo remover\_basura($venta['total\_saleing\_price']); ?></td>*

*<td class="text-center"><?php echo date("d/m/Y", strtotime ($venta['date'])); ?></td>*

*<td class="text-center"><?php echo $usuario['username'] ?></td>*

*</tr>*

*<?php endforeach;?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 60 se muestra el resultado de la codificación del código de arriba.

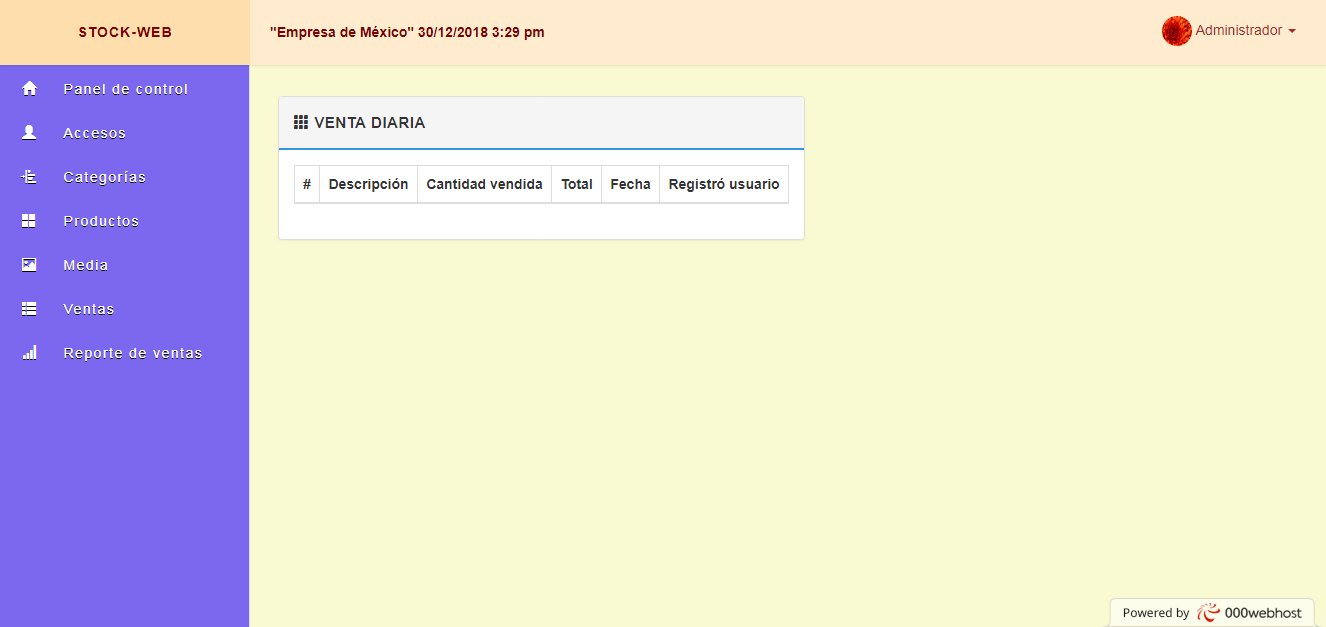


Ilustración 60 Sección de ventas diarias

## Sección de ventas mensuales

En esta vista se observa un panel con la lista de ventas mensuales registradas.

Se utilizan las secciones de cabecera, pie y el menú de usuario correspondiente.

A continuación, se muestra el código y la pantalla.

*<?php*

*$titulo\_pagina = 'Ventas mensuales';*

*require\_once('incluye/cargar.php');*

*pagina\_requiere\_nivel(3);*

*$agno = date('Y');*

*$mes = date('m');*

*$ventas = ventasMensuales($agno,$mes);*

*include\_once('disenos/cabecera.php');*

*?>*

*<div class="row">*

*<div class="col-md-12">*

*<?php echo mostrar\_mensaje($mensaje); ?>*

*</div>*

*</div>*

*<div class="row">*

*<div class="col-md-12">*

*<div class="panel panel-default">*

*<div class="panel-heading clearfix">*

*<strong>*

*<span class="glyphicon glyphicon-th"></span>*

*<span>Ventas mensuales</span>*

*</strong>*

*</div>*

*<div class="panel-body">*

*<table class="table table-bordered table-striped">*

*<thead>*

*<tr>*

*<th style="vertical-align: middle" class="text-center">#</th>*

*<th style="vertical-align: middle" class="text-center"> Descripción </th>*

*<th style="vertical-align: middle" class="text-center"> Cantidad vendida</th>*

*<th style="vertical-align: middle" class="text-center"> Total </th>*

*<th style="vertical-align: middle" class="text-center"> Fecha </th>*

*<th style="vertical-align: middle" class="text-center"> Registró usuario </th>*

*</tr>*

*</thead>*

*<tbody>*

*<?php foreach ($ventas as $venta):?>*

*<tr>*

*<td class="text-center"><?php echo contar\_id();?></td>*

*<td><?php echo remover\_basura($venta['nombre']); ?></td>*

*<td class="text-center"><?php echo (int)$venta['cantidad']; ?></td>*

*<td class="text-center"><?php echo remover\_basura($venta['total\_saleing\_price']); ?></td>*

*<td class="text-center"><?php echo date("d/m/Y", strtotime ($venta['date'])); ?></td>*

*<td class="text-center"><?php echo $usuario['username'] ?></td>*

*</tr>*

*<?php endforeach;?>*

*</tbody>*

*</table>*

*</div>*

*</div>*

*</div>*

*</div>*

*<?php include\_once('disenos/pie.php');*

En la ilustración 61 se muestra el resultado de la codificación del código de arriba.

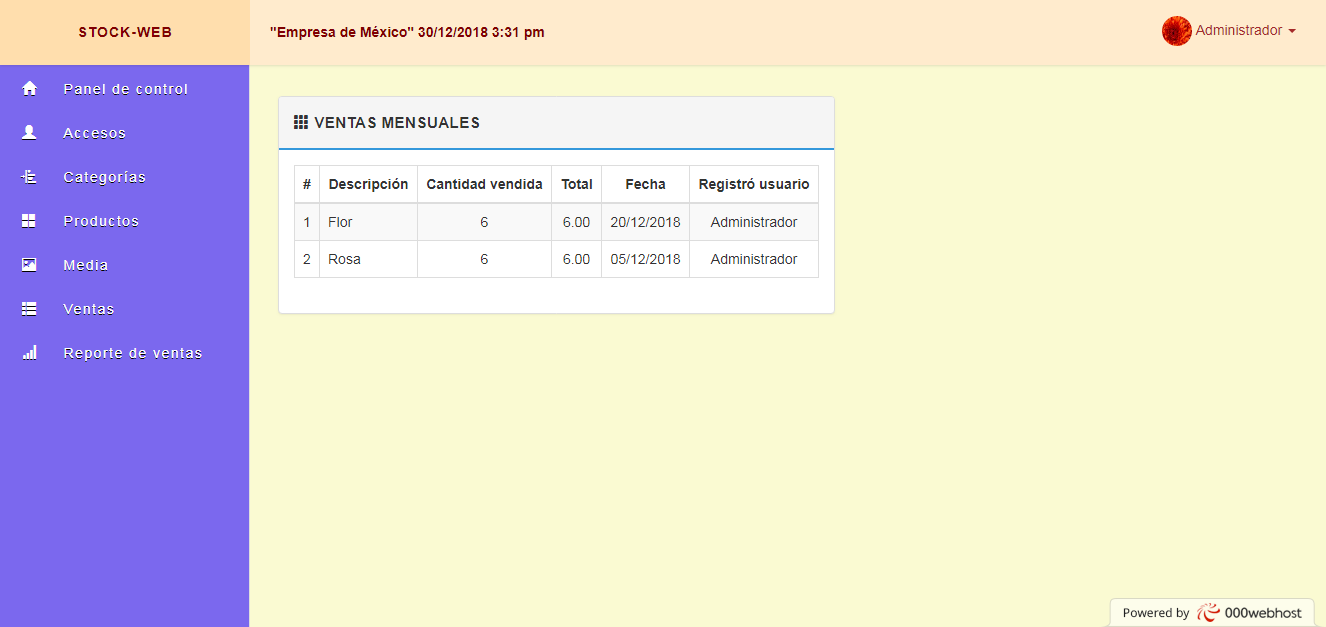


Ilustración 61 Sección de ventas mensuales

## Archivo para verificar antes de agregar un grupo

En este archivo *“verificar\_agregar\_grupo.php”* se verifica que se pueda agregar un grupo. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se crea un arreglo *“$campos\_requeridos“* donde se guarda el nombre de grupo, se procede a validar los campos por medio de la función *“validar\_campos“*. Se crea una variable *“$nombre\_grupo“* que tendrá como valor el nombre de grupo recibido por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se crea una variable *“$grupo“* que tendrá como valor el nombre de grupo por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se checa si la variable *“$errors“* (que tiene cualquier error) está vacía. Se procede a guardar en la variable *“$id\_usuario “*lo que devuelva la función *“autenticar“*, donde esta recibe como parámetro la variable *“$grupo“*. Se procede a checar el valor de *“grupo “* en caso de que ya se encuentre registrado. Se informa al usuario por medio de un mensaje.

A continuación, se muestra el código.

*<?php*

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

*require\_once('incluye/cargar.php');*

*$campos\_requeridos = array('nombre\_grupo','nivel\_grupo');*

*validar\_campos($campos\_requeridos);*

*if(isset($\_POST['nombre\_grupo']) && isset($\_POST['nivel\_grupo']) && isset($\_POST['estado'])) {*

*if(encontrar\_por\_nombre\_grupo($\_POST['nombre\_grupo']) === false ) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>El nombre de grupo ya existe.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$verificacion = -1;*

*} elseif(encontrar\_por\_nivel\_grupo($\_POST['nivel\_grupo']) === false) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Este nivel de grupo ya existe.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$verificacion = -2;*

*}*

*elseif(empty($errores)) {*

*$nombre = remover\_basura($db->escape($\_POST['nombre\_grupo']));*

*$nivel = remover\_basura($db->escape($\_POST['nivel\_grupo']));*

*$estado = remover\_basura($db->escape($\_POST['estado']));*

*$query = "INSERT INTO grupo (";*

*$query .="nombre,nivel,estado\_actual";*

*$query .=") VALUES (";*

*$query .=" '{$nombre}', '{$nivel}','{$estado}'";*

*$query .=")";*

*if($db->query($query)) {*

*$sesion->mensaje('s',"El grupo ha sido creado.");*

*$verificacion = 0;*

*} else {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Lamentablemente no se pudo crear el grupo.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*}*

*} else {*

*$sesion->mensaje("d", $errores);*

*}*

*}*

*$requisito = "<input type='hidden' id='requisito' value='$verificacion' />";*

*echo $mensaje;*

*echo $requisito;*

## Archivo para verificar antes de agregar un producto

En este archivo *“verificar\_agregar\_producto.php”* se verifica que se pueda agregar un producto. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se crea un arreglo *“$campos\_requeridos“* donde se guarda el nombre de producto, se procede a validar los campos por medio de la función *“validar\_campos“*. Se crea una variable *“$nombre\_producto“* que tendrá como valor el nombre de producto recibido por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se crea una variable *“$producto“* que tendrá como valor el nombre de producto por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se checa si la variable *“$errors“* (que tiene cualquier error) está vacía. Se procede a guardar en la variable *“$id\_usuario “*lo que devuelva la función *“autenticar“*, donde esta recibe como parámetro la variable *“$producto“*. Se procede a checar el valor de *“$producto“* en caso de que ya se encuentre registrado. Se informa al usuario por medio de un mensaje.

A continuación, se muestra el código.

*<?php*

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

*require\_once('incluye/cargar.php');*

*$campos\_requeridos = array('titulo','categoria','cantidad','compra','precio');*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)){*

*$nombre\_producto = remover\_basura($db->escape($\_POST['titulo']));*

*$categoria\_producto = remover\_basura($db->escape($\_POST['categoria']));*

*$cantidad\_producto = remover\_basura($db->escape($\_POST['cantidad']));*

*$compra\_producto = remover\_basura($db->escape($\_POST['compra']));*

*$compra\_decimal\_producto = remover\_basura($db->escape($\_POST['compra\_decimal']));*

*$compra\_decimal\_producto += 0.00;*

*$compra\_decimal\_producto /= 100.00;*

*$compra\_producto += $compra\_decimal\_producto;*

*$vental\_producto = remover\_basura($db->escape($\_POST['precio']));*

*$venta\_decimal\_producto = remover\_basura($db->escape($\_POST['precio\_decimal']));*

*$venta\_decimal\_producto += 0.00;*

*$venta\_decimal\_producto /= 100.00;*

*$vental\_producto += $venta\_decimal\_producto;*

*$media\_id = remover\_basura($db->escape($\_POST['foto']));*

*$date = crear\_fecha();*

*$comprobar = encontrar\_nombre\_producto($nombre\_producto);*

*$variable = null;*

*foreach($comprobar as $buscar\_producto) {*

*$variable = $buscar\_producto['nombre'];*

*}*

*if($variable != null) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Ya hay un nombre de producto igual.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$verificacion = -1;*

*} else {*

*$query = "INSERT INTO producto (";*

*$query .=" nombre,cantidad,precio\_compra,precio\_venta,id\_categoria,id\_media,fecha\_actualizacion";*

*$query .=") VALUES (";*

*$query .=" '{$nombre\_producto}', '{$cantidad\_producto}', '{$compra\_producto}', '{$vental\_producto}', '{$categoria\_producto}', {$media\_id}, '{$date}'";*

*$query .=")";*

*if($db->query($query)){*

*$sesion->mensaje('s',"El producto fue agregado exitosamente. ");*

*$verificacion = 0;*

*} else {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Ya hay un nombre de producto igual.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$verificacion = -1;*

*}*

*}*

*} else {*

*$mensaje = $errores;*

*$verificacion = -1;*

*}*

*$requisito = "<input type='hidden' id='requisito' value='$verificacion' />";*

*echo $mensaje;*

*echo $requisito;*

## Archivo para verificar antes de agregar usuario

En este archivo *“verificar\_agregar\_usuario.php”* se verifica que se pueda agregar un usuario. Primero se carga el archivo *“cargar.php”* ubicado en el directorio *“incluye”*. Se crea un arreglo *“$campos\_requeridos“* donde se guarda el nombre de usuario, se procede a validar los campos por medio de la función *“validar\_campos“*. Se crea una variable *“$nombre\_usuario“* que tendrá como valor el nombre de usuario recibido por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se crea una variable *“$usuario“* que tendrá como valor el nombre de usuario por POST y filtrado de caracteres extraños por medio de la función *“remover\_basura“*. Se checa si la variable *“$errors“* (que tiene cualquier error) está vacía. Se procede a guardar en la variable *“$id\_usuario “*lo que devuelva la función *“autenticar“*, donde esta recibe como parámetro la variable *“$usuario“*. Se procede a checar el valor de *“$usuario“* en caso de que ya se encuentre registrado. Se informa al usuario por medio de un mensaje.

A continuación, se muestra el código.

*<?php*

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

*require\_once('incluye/cargar.php');*

*$campos\_requeridos = array('nombre','usuario','password','rol');*

*validar\_campos($campos\_requeridos);*

*if(empty($errores)){*

*$nombre = remover\_basura($db->escape($\_POST['nombre']));*

*$nombre\_usuario = remover\_basura($db->escape($\_POST['usuario']));*

*$contrasena = remover\_basura($db->escape($\_POST['password']));*

*$rol = (int)$db->escape($\_POST['rol']);*

*$contrasena = sha1($contrasena);*

*$usuarios = encontrar\_nombre\_usuario($nombre\_usuario);*

*foreach($usuarios as $usuario):*

*$nombreUsuario = $usuario['usuario'];*

*endforeach;*

*$comprobar = encontrar\_nombre\_usuarios($nombre\_usuario);*

*if($comprobar != null) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Ya hay un nombre de usuario igual.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$verificacion = -1;*

*} else {*

*$query = "INSERT INTO usuario (";*

*$query .="id\_grupo\_usuario,nombre,username,contrasena,estado";*

*$query .=") VALUES (";*

*$query .="$rol, '{$nombre}', '{$nombre\_usuario}', '{$contrasena}','1'";*

*$query .=")";*

*if($db->query($query)) {*

*$valores = encontrar\_usuario($nombre);*

*foreach($valores as $valor):*

*$idUsuario = $valor['id'];*

*endforeach;*

*$query = "INSERT INTO tema\_personal (";*

*$query .="id\_usuario,id\_tema";*

*$query .=") VALUES (";*

*$query .=" '{$idUsuario}' ,'1'";*

*$query .=")";*

*if($db->query($query)) {*

*$sesion->mensaje('s'," La cuenta de usuario ha sido creada.");*

*$verificacion = 0;*

*}*

*} else {*

*if(isset($nombreUsuario)) {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>Ya hay un nombre de usuario igual.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$verificacion = -1;*

*} else {*

*$mensaje = "<div class='alert alert-danger'><span style='padding-right:5px'>No se pudo crear la cuenta.</span><button type='button' class='close' data-dismiss='alert' aria-label='Close'><span aria-hidden='true'>&times;</span></button></div>";*

*$verificacion = -1;*

*}*

*}*

*}*

*} else {*

*$mensaje = $errores;*

*$verificacion = -1;*

*}*

*$requisito = "<input type='hidden' id='requisito' value='$verificacion' />";*

*echo $mensaje;*

*echo $requisito;*

# V. PROCESO DE EVALUACIÓN DE ALCANCE DE LOS OBJETIVOS

El objetivo principal, el cual consistía en crear un sistema de inventario se cumplió como se muestra en la ilustración 62.

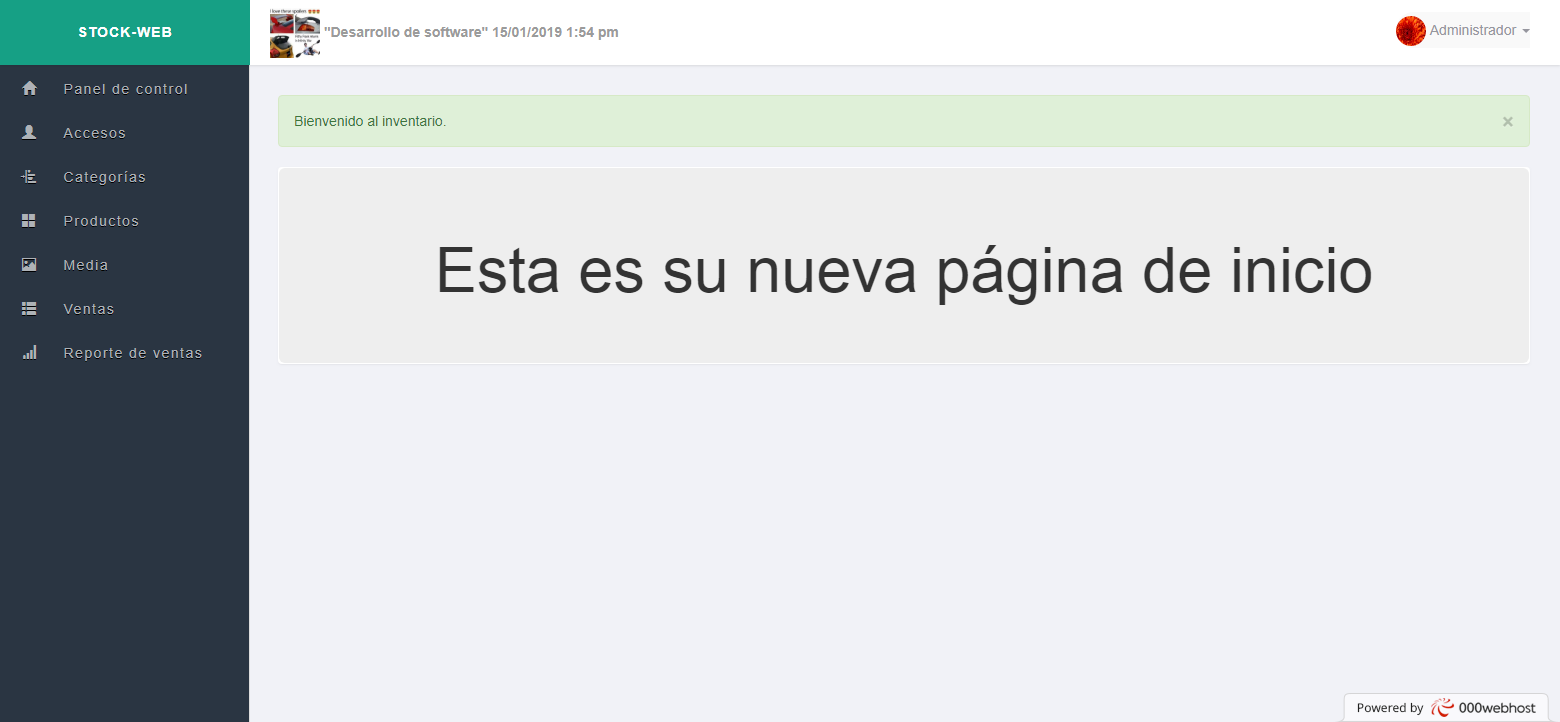


Ilustración 62 Sistema de inventario en línea "Stock-Web"

Objetivos específicos

Se cumplió desarrollar una aplicación web para la administración de un inventario en línea por medio de HTML5, CSS3, PHP y JavaScript como se muestra en la ilustración 63.

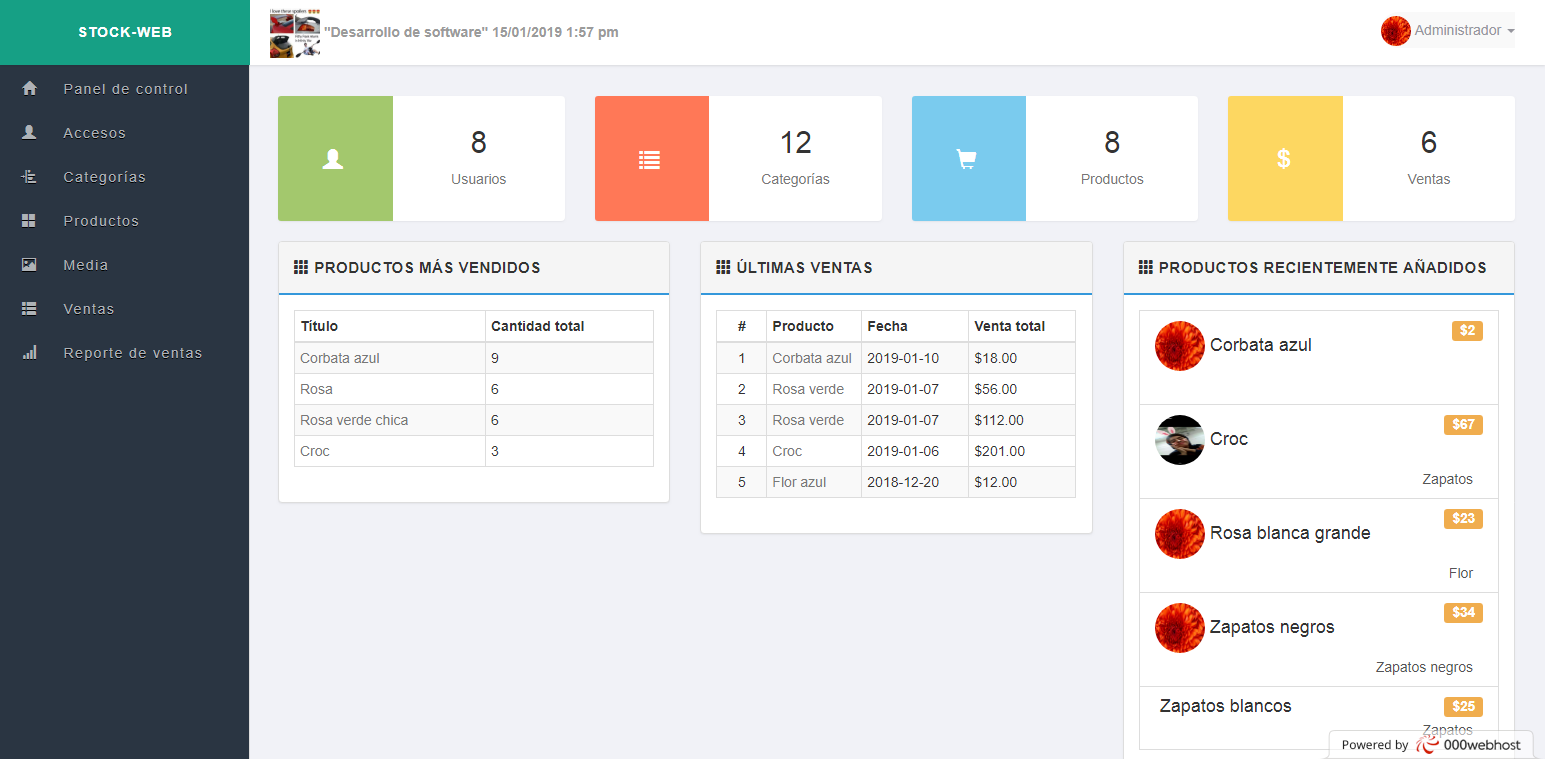


Ilustración 63 Sistema de inventario

Se cumplió programar la sección de productos, ventas, reportes de ventas, etc., por medio de PHP como se muestra en la ilustración 64.

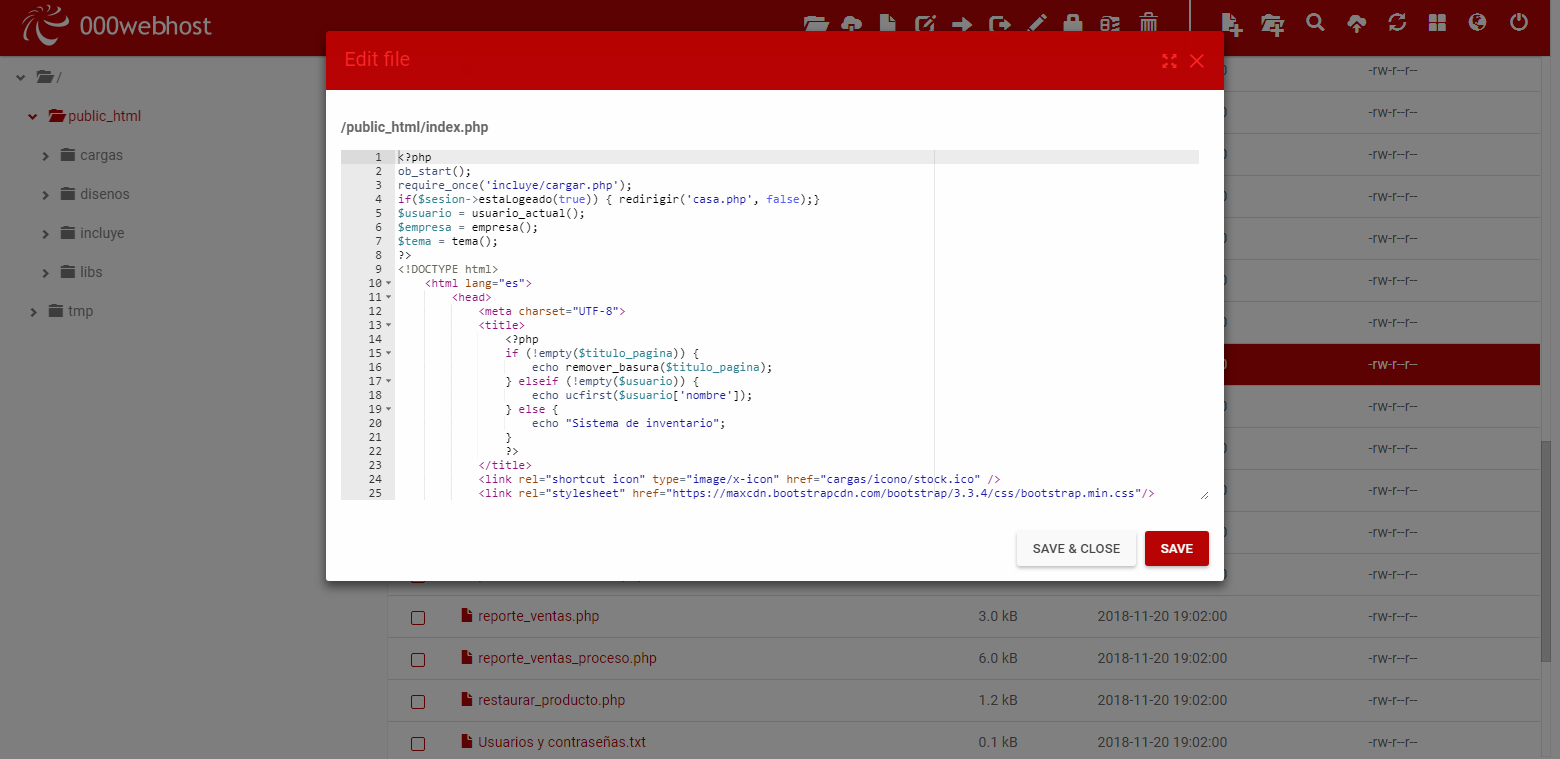


Ilustración 64 Programación en PHP

Se cumplió utilizar el IDE NetBeans para diseñar y programar esta aplicación en una computara con el sistema operativo Windows por ser a juicio personal el IDE más vistoso como se muestra en la ilustración 65.

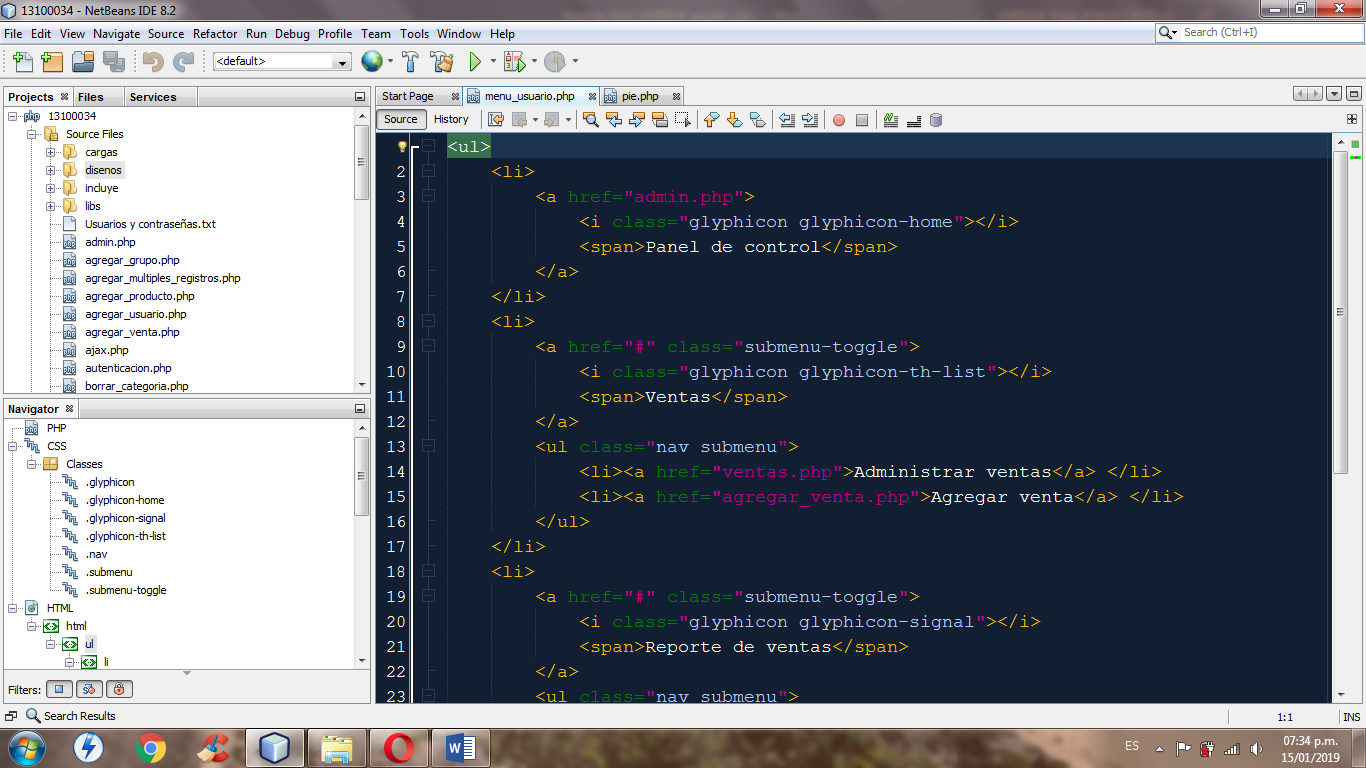


Ilustración 65 Uso de NetBeans

Se cumplió hacer uso del lenguaje de programación PHP para programar desde el lado del servidor la aplicación web por medio del IDE como se muestra en la ilustración 66.

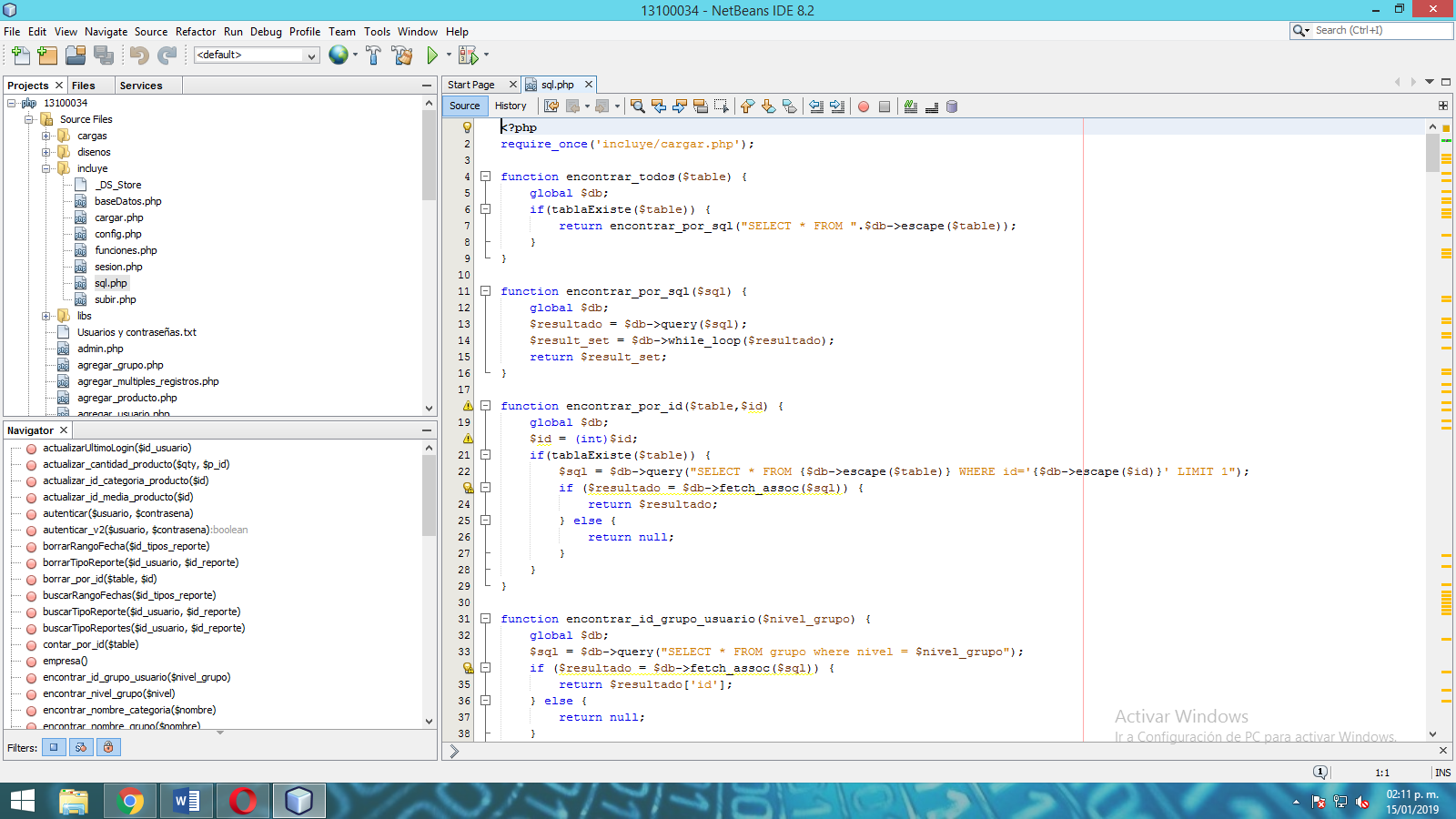


Ilustración 66 Programación en PHP por medio del IDE

Se cumplió hacer uso del lenguaje de programación JavaScript para programar desde el lado del cliente la aplicación web por medio del IDE como se muestra en la ilustración 67.

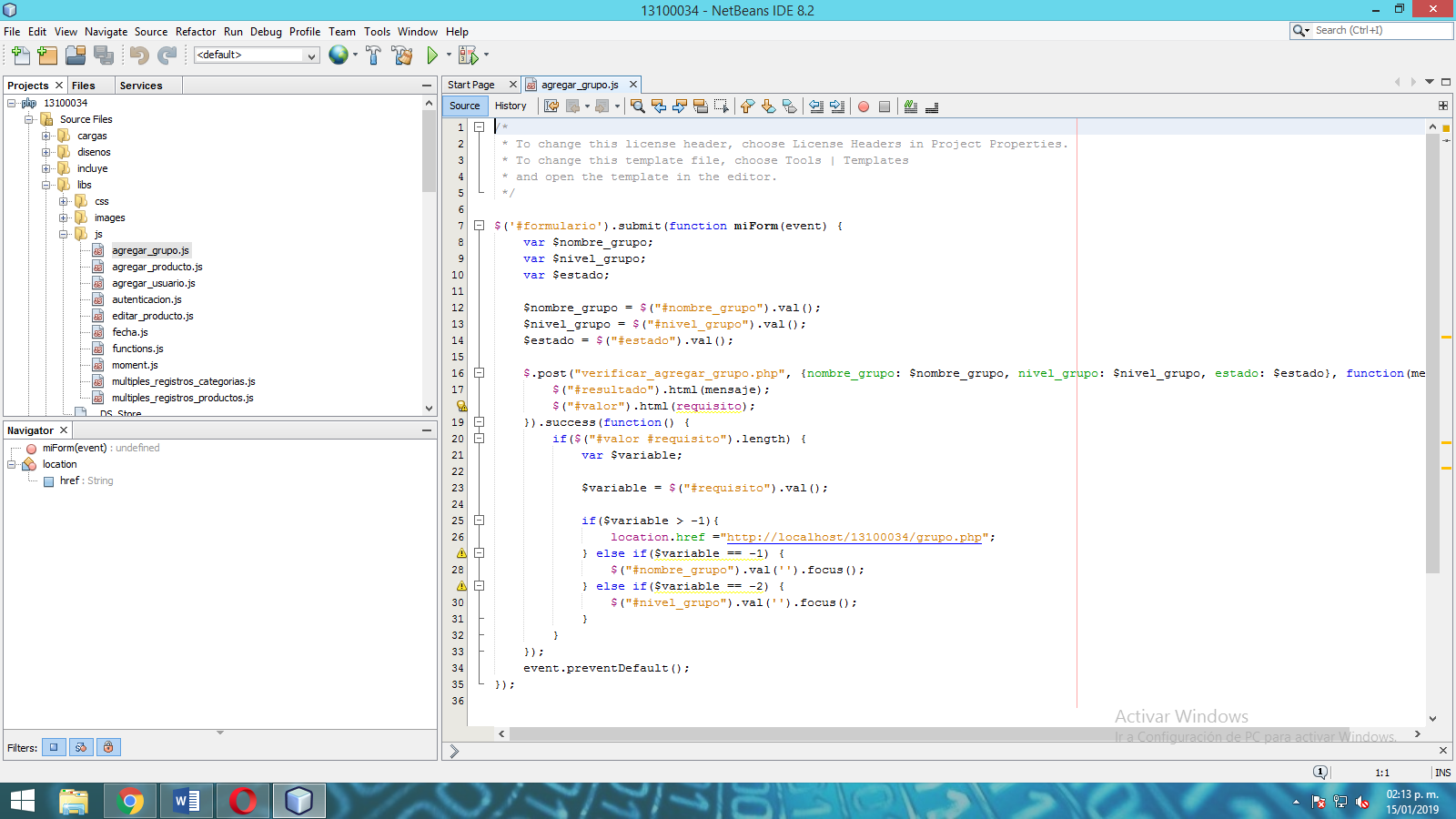


Ilustración 67 Programación en JavaScript por medio del IDE

Se cumplió hacer uso del lenguaje de etiquetas HTML5 para diseñar las páginas web de la aplicación web por medio del IDE como se muestra en la ilustración 68.

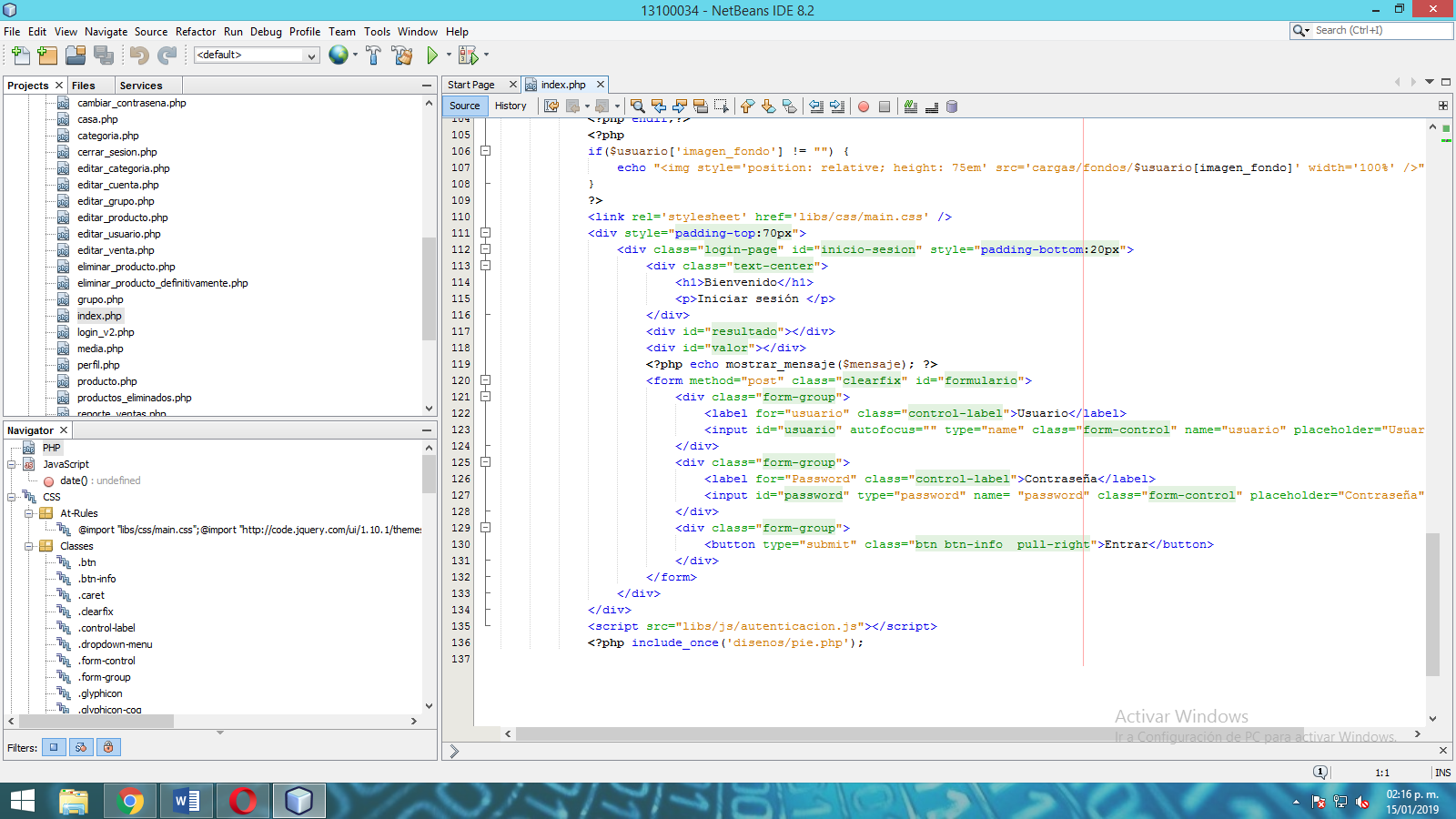


Ilustración 68 Uso de HTML por medio del IDE

Se cumplió hacer uso del lenguaje de estilos en cascada CSS para diseñar las páginas web de la aplicación web por medio del IDE como se muestra en la ilustración 69.

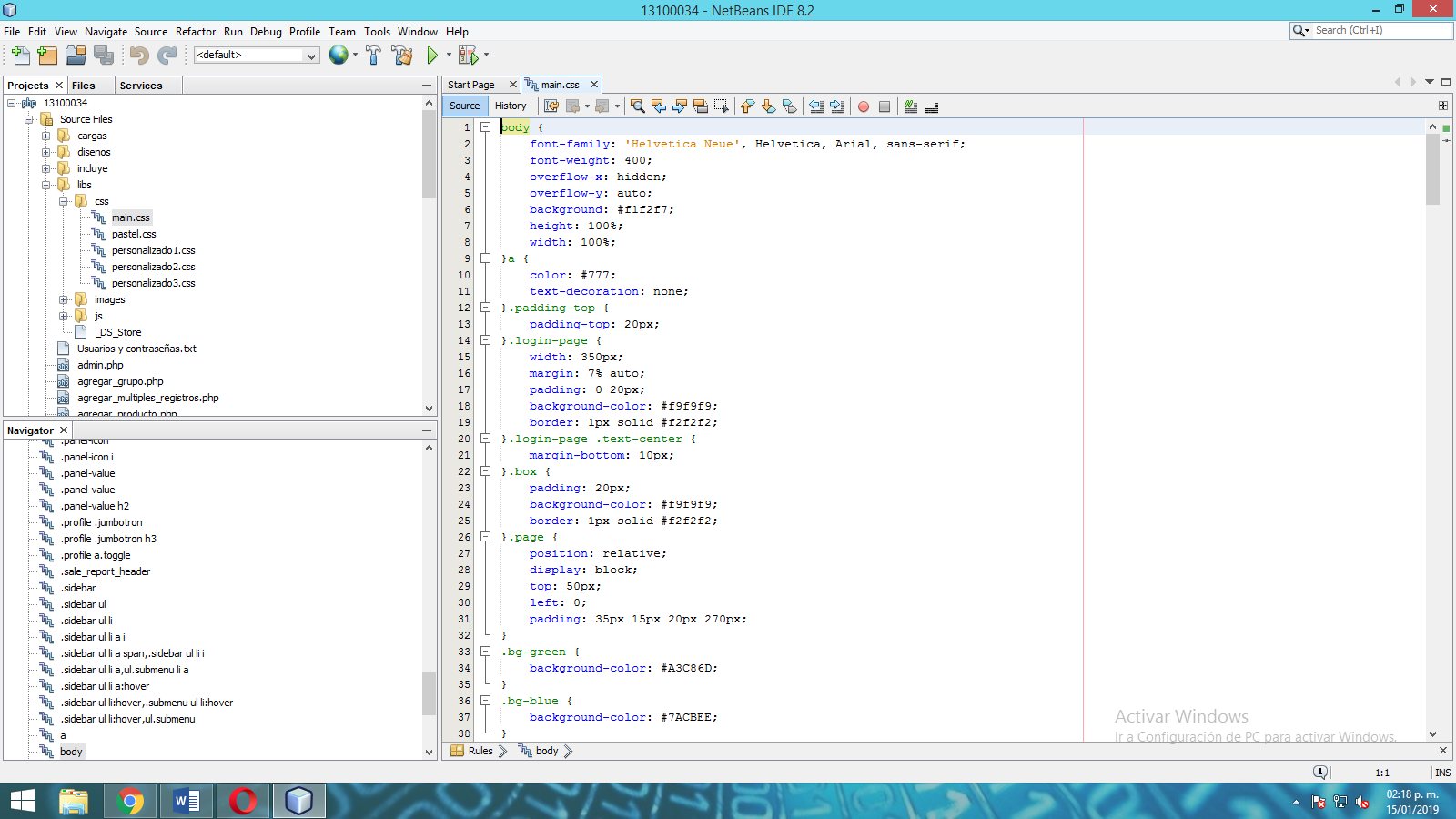


Ilustración 69 Uso de CSS por medio del IDE

Se cumplió hacer uso de un SGBD MySQL para almacenar los datos de la aplicación web por medio del hosting como se muestra en la ilustración 70.

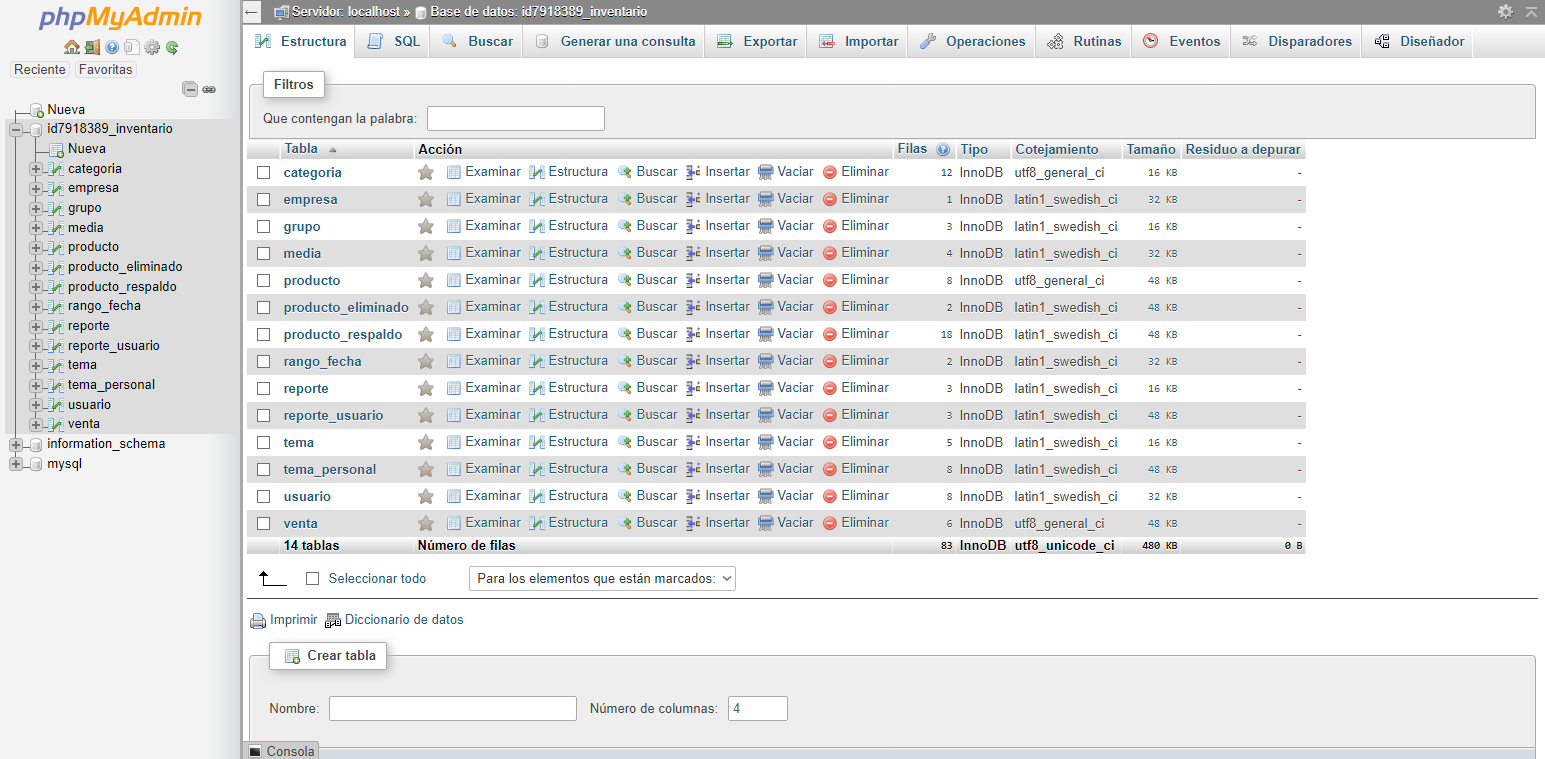


Ilustración 70 Uso del SGBD del hosting

Se cumplió diseñar y programar adecuadamente todas las interfaces de la aplicación web que verá el usuario por medio del IDE como se muestra en la imagen 71.



Ilustración 71 Programación de las interfaces por medio del IDE

Se cumplió hacer pruebas a la aplicación web para checar su funcionamiento de cara al usuario final como se muestra en la ilustración 72.

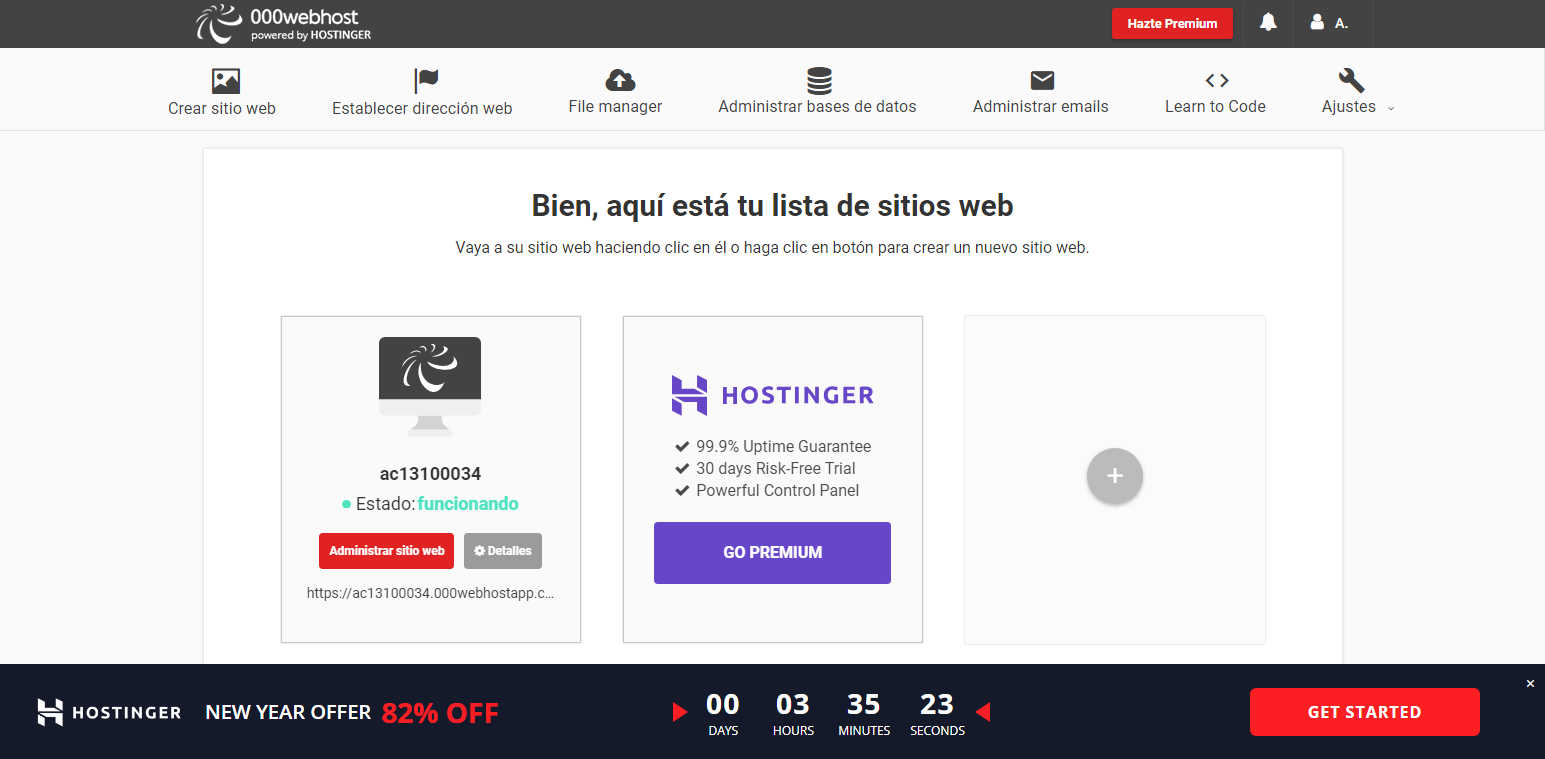


Ilustración 72 Funcionamiento de la aplicación web

Resultados de pruebas del sistema

Se muestran dentro de una tabla los casos de pruebas realizadas al sistema para comprobar su funcionalidad.

| Caso de prueba | Estado de prueba | Error | Tiempo que demoró la corrección | ¿Cómo se resolvió? |
| --- | --- | --- | --- | --- |
| CP1. Formulario para iniciar sesión con nombre de usuario y contraseña. | Éxito. | El nombre de usuario era inválido si se ingresaba con alguna letra en minúscula o mayúscula. | 2 horas. | El nombre de usuario que se ingresa se convierte a minúsculas antes de compararlo con el que se encuentra en la base de datos. |
| CP2. Cuadro de error de formulario de inicio de sesión. | Éxito. | Se mostraba información incorrecta al usuario sobre su inicio de sesión. | 1 día. | Se comprueba minuciosamente el origen del error. |
| CP3. Formulario para editar nombre de grupo de tipo administrador | Éxito. | Se podía editar el nombre de grupo, su estado y su tipo. | 2 horas. | Se comprueba que el usuario tipo administrador únicamente puede modificar el nombre del grupo de administradores. |
| CP4. Formulario para editar el nombre personal del usuario administrador por defecto y con sesión activa. | Éxito. | Se permitía modificar el nombre personal de usuario, nombre de usuario, tipo y estado de usuarios con sesión activa. | 3 horas. | Se comprueba que los usuarios a editar no sean aquellos que tienen sesión activa ni el administrador por defecto. |
| CP5. Acciones de usuario con sesión activa y administrador por defecto. | Éxito. | Se permitía borrar el usuario con sesión activa y administrador por defecto. | 2 horas. | El usuario con sesión activa y administrador por defecto |
| CP6. Formulario para restaurar producto eliminado. | Éxito. | El producto se eliminaba en vez de restaurarse. | 5 horas. | El nombre de producto no se elimina antes de intentar restaurarlo. |
| CP7. Formulario para agregar varias categorías. | Éxito. | Los nombres de categorías podían repetirse y deben ser diferentes. | 2 días. | Los nombres de categorías se deben checar antes de mandarlos a la base de datos. |
| CP8. Formulario para agregar varios productos. | Éxito. | Las descripciones de productos podían repetirse y deben ser diferentes. | 2 días. | Las descripciones de categorías se deben checar antes de mandarlos a la base de datos. |
| CP9. Formulario para editar el nombre personal de usuario, así como el nombre de usuario. | Éxito. | El nombre de usuario no estaba correctamente definido en el formulario | 1 hora. | Se colocó correctamente el nombre de usuario en el formulario. |
| CP10. Formulario para cambiar la contraseña. | Éxito. | La nueva contraseña no se guardaba en la base de datos. | 1 día. | Se guarda la nueva contraseña antes de redirigir a la sección de inicio de sesión. |

# VI. CONCLUSIONES Y COMENTARIOS FINALES

## VI.1 SITUACIÓN ACTUAL DEL PROYECTO.

El proyecto fue concluido en un 95%, lo único que falta es actualizar correctamente vínculos a los productos que se actualizan en la sección de panel de control.

## VI.2 FUTURO DEL PROYECTO.

Actualmente se tiene planeado que esta plataforma sea utilizada por una empresa de limpieza, que espera que su inventario y sus ventas sean registradas de manera correcta por este sistema.

## VI.3 PRINCIPALES INTERROGANTES POR RESOLVER EN ACTIVIDADES FUTURAS

Una de las interrogantes principales será lidiar con la información que almacenará el sistema cuando un usuario se da de baja, como poder almacenar esa información y poder recuperarla en caso de que la persona desee poder volver utilizar su cuenta.

Debido a que el servidor no soporta mucha información dentro del mismo, puede que cuando se da de baja a un usuario, se tendrá que eliminar también su información para que no esté ocupando espacio que otro usuario pueda utilizar.

## VI.4 LECCIONES APRENDIDAS SOBRE EL PROYECTO

Se aprendió a utilizar la tecnología AJAX para intercambiar variables entre los lenguajes JavaScript y PHP; para ello se utilizó la función de JQuery $.post para realizar el traspaso de valores, ya que PHP se ejecuta en el servidor y JavaScript en el navegador y eso los hace incompatibles entre ellos, con lo cual necesitó de la función $. post para hacer traspaso de variables.

Se aprendió a reutilizar archivos con funciones de PHP para no mezclar funciones entre sí y así tener en un solo archivo la configuración. De esta manera solo se incluye este archivo de configuración en cada vista que se requiera.

Se aprendió a utilizar la función sha1 de PHP para la codificación y decodificación de las contraseñas que ingresa el usuario. Esto debido a que sin esta función el administrador de la base de datos podría ver las contraseñas con solo ingresar al SGBD y modificarlas.

Se aprendió a utilizar la función fetch\_array de PHP para obtener el resultado de una consulta y por medio de un ciclo iterativo, obtener los registros que se necesitaban mostrar al usuario.

Se aprendió a utilizar clases y objetos en PHP para mejorar la legibilidad del código programado para evitar mezclar código y repetir funciones.

Se reestructuró la planeación del proyecto ya que no se conocía en su totalidad las tecnologías para crear una aplicación web comenzando con HTML, CSS, JavaScript y PHP. De HTML se aprendió la estructura correcta de una página web, de CSS se aprendió a utilizar un entorno de desarrollo, de JavaScript se aprendió a utilizar una librería de funciones y de PHP se aprendió a utilizar clases para subir imágenes a la base de datos. En la programación del proyecto se aprendió a no mezclar código de diferentes tecnologías para no empeorar la legibilidad de los archivos del proyecto, por ejemplo, colocando código JavaScript, CSS y/o PHP en las páginas web escritas en HTML.

## VI.5 LECCIONES APRENDIDAS SOBRE EL PROCESO DEL PROYECTO

Se aprendió a seguir un proceso para desarrollar el proyecto de software. Se desarrolló el proyecto codificando primero, por lo cual se alargo considerablemente el tiempo de desarrollo del proyecto. Se aprendió que lo primero que se debe hacer para desarrollar software es analizar el problema a resolver, el o los diagramas necesarios para tener una visión de como se va a resolver el problema, la elección de la o las tecnologías más idóneas para resolver el problema, así como las variables que se necesitan y por último la codificación del proyecto de software. En la codificación del proyecto se aprendió a nombrar las variables con nombres claros, para asociar el nombre con el papel que desempeña la variable. Se aprendió a utilizar un entorno de desarrollo exclusivo para codificar, que mejora considerablemente leer el código y codificar de manera más eficiente. Se aprendió a guardar avances del proyecto y manejar un control de versiones de estos avances en un repositorio en línea como OneDrive. En la publicación del proyecto en línea se aprendió a realizar un respaldo de la información publicada en el servidor del proveedor por cualquier tipo de error.

# REFERENCIAS Y BIBLIOGRAFÍA

# Bibliografía

Blog. (10 de Febrero de 2010). *BLOG*. Obtenido de Usando AJAX: Un simple ejemplo con PHP y jQuery.: https://www.imaginanet.com/blog/usando-ajax-un-simple-ejemplo-con-php-y-jquery.html

Bootstrap. (s.f.). *Bootstrap*. Obtenido de Bootstrap: https://getbootstrap.com

Foros del web. (17 de Abri de 2010). *PHP*. Obtenido de Como poner imagenes en php: http://www.forosdelweb.com/f18/como-poner-imagenes-php-798424/

guebs. (s.f.). *guebs*. Obtenido de Capítulo 20. Disparadores (triggers): https://manuales.guebs.com/mysql-5.0/triggers.html

jQuery. (s.f.). *jQuery user interface*. Obtenido de jQuery: https://jqueryui.com/datepicker/

jQuery. (s.f.). *jQuery write less, do more*. Obtenido de jQuery API: https://api.jquery.com

Lázaro, D. (2018). *Libro de PHP*. Obtenido de Sesiones en PHP: https://diego.com.es/sesiones-en-php

Millán, M. E. (2012). Fundamentos de base de datos. En M. E. Millán, *Fundamentos de base de datos* (pág. 154). Colombia: Universidad del Valle.

Moment. (s.f.). *Moment.js*. Obtenido de Moment.js Documentation: https://momentjs.com/docs/

php.net. (s.f.). *Manual de PHP*. Obtenido de Referencia del lenguaje: http://php.net/manual/es/language.oop5.php

php.net. (s.f.). *Manual de PHP*. Obtenido de API original de MySQL: http://php.net/manual/es/book.mysql.php

php.net. (s.f.). *Manual de PHP*. Obtenido de Variables desde fuentes externas: http://php.net/manual/es/language.variables.external.php

# APÉNDICE

# MANUAL DE USUARIO

**Manual de Usuario**



Ilustración 73 Sistema de inventario en línea Stock-Web

**Sistema de inventario en línea Stock-Web**

**INTEGRANTE(S)**

Ayala Castillo Enrique Paul 13100034

**GUADALAJARA, JAL.** Agosto – diciembre 2018

Tabla de contenido

[MANUAL DE USUARIO 1](#_Toc535351083)

[1.1. Descripción de la plataforma 3](#_Toc535351084)

[1.2. Guía de instalación 3](#_Toc535351085)

[1.3. Manual de usuario 3](#_Toc535351086)

[1.3.1 Bienvenida 3](#_Toc535351087)

[1.3.2 Ingreso a la aplicación 4](#_Toc535351088)

[1.3.3 Página principal 4](#_Toc535351089)

[1.3.4 Panel de control 6](#_Toc535351090)

[1.3.5 Sección de grupos 7](#_Toc535351091)

[1.3.6 Sección para editar grupo 8](#_Toc535351092)

[1.3.7 Sección para agregar grupo 9](#_Toc535351093)

[1.3.8 Sección de usuarios 10](#_Toc535351094)

[1.3.9 Sección para editar usuario 11](#_Toc535351095)

[1.3.10 Sección para agregar usuario 12](#_Toc535351096)

[1.3.11 Sección de categorías 13](#_Toc535351097)

[1.3.12 Sección de editar categoría 13](#_Toc535351098)

[1.3.13 Sección de productos 14](#_Toc535351099)

[1.3.14 Sección de editar producto 15](#_Toc535351100)

[1.3.15 Sección para agregar producto 16](#_Toc535351101)

[1.3.16 Sección de productos eliminados 17](#_Toc535351102)

[1.3.17 Sección de imágenes de productos 18](#_Toc535351103)

[1.3.18 Sección para agregar múltiples registros 19](#_Toc535351104)

[1.3.19 Sección de ventas 20](#_Toc535351105)

[1.3.20 Sección de editar venta 21](#_Toc535351106)

[1.3.21 Sección de agregar venta 22](#_Toc535351107)

[1.3.22 Sección de reporte de ventas 22](#_Toc535351108)

[1.3.23 Sección de reporte de ventas generado 23](#_Toc535351109)

[1.3.24 Sección de ventas mensuales 24](#_Toc535351110)

[1.3.25 Sección de ventas diarias 24](#_Toc535351111)

[1.3.26 Sección de perfil 25](#_Toc535351112)

[1.3.27 Sección de editar cuenta 26](#_Toc535351113)

[1.3.28 Sección para cambiar contraseña 27](#_Toc535351114)

[ANEXOS 29](#_Toc535351115)

## 1.1. Descripción de la plataforma

El sistema de inventario “Stock-Web” proporciona una manera efectiva y completa de controlar el inventario de un negocio. Puede administrar cuentas de usuario, agregar productos, ventas y/o generar reportes de ventas. Su limitante es el solo registrar ventas más no hacerlas por medio de este sistema. El beneficio es su fácil utilización y su interfaz interactiva para poder administrar el inventario del negocio. El objetivo es ofrecer a las empresas un sistema que administré el inventario del negocio. La meta es proveer un sistema que resuelva el problema de inventario de una empresa a la inmediatez.

En las siguientes páginas se detalla el uso y funcionamiento de la aplicación para un eficaz manejo.

## 1.2. Guía de instalación

En cuanto a instalación no será necesario instalar ningún software adicional a tu computadora debido a que se podrá acceder a la plataforma mediante tu navegador con conexión a internet.

## 1.3. Manual de usuario

### 1.3.1 Bienvenida

Es la página inicial y se muestra nada más al entrar al sitio web. Esta página tiene en el centro el formulario para iniciar sesión. El usuario deberá completar los campos de usuario y contraseña. Se mostrarán mensajes de error en la parte superior del campo de usuario si el nombre de usuario o contraseña son incorrectos, además de indicar si la cuenta del usuario esta deshabilitada o el grupo de usuarios al que pertenece esta deshabilitado. Una vez lleno el formulario se procede a dar clic en entrar. Una vez iniciada la sesión se direccionará al usuario a la página principal.



Ilustración 74 Página de inicio de sesión

### 1.3.2 Ingreso a la aplicación

Para poder ingresar a la aplicación web solo es necesario ingresar el siguiente link en el navegador Google Chrome y Opera para que funcione correctamente a ac13100034.000webhostapp.com.

IMPORTANTE.

La plataforma web fue diseñada para trabajar con el navegador Google Chrome y Opera, si se utiliza en otros navegadores se corre el riesgo de que algunas funciones puedan no funcionar correctamente.

### 1.3.3 Página principal

Es la primera página que se le muestra al usuario con sesión iniciada. A continuación, se muestran 3 imágenes donde se ven las páginas principales de los 3 tipos de usuario: administrador, especial y estándar. La cabecera está compuesta de el nombre de la aplicación en la parte superior izquierda y a su derecha el logo de la empresa, el lema de la empresa, la fecha, hora actual y una lista desplegable con el nombre del usuario actual que tendrá como elementos el perfil del usuario, su configuración y poder cerrar sesión. Debajo del nombre de la aplicación, se encuentra un menú para accesar a diferentes secciones; para el usuario administrador: panel de control, accesos (administrar grupos y administrar usuarios), categorías, productos (administrar productos, agregar productos y productos eliminados), media (administrar imágenes y agregar múltiples registros), ventas (administrar ventas y agregar venta) y reporte de ventas (ventas por fecha, ventas mensuales y ventas diarias); para el usuario especial: panel de control, categorías, productos (administrar productos, agregar productos y productos eliminados), media (administrar imágenes), ventas (administrar ventas y agregar venta) y reporte de ventas (ventas por fecha, ventas mensuales y ventas diarias); para el usuario estándar: panel de control, ventas (administrar ventas y agregar venta) y reporte de ventas (ventas por fecha, ventas mensuales y ventas diarias).

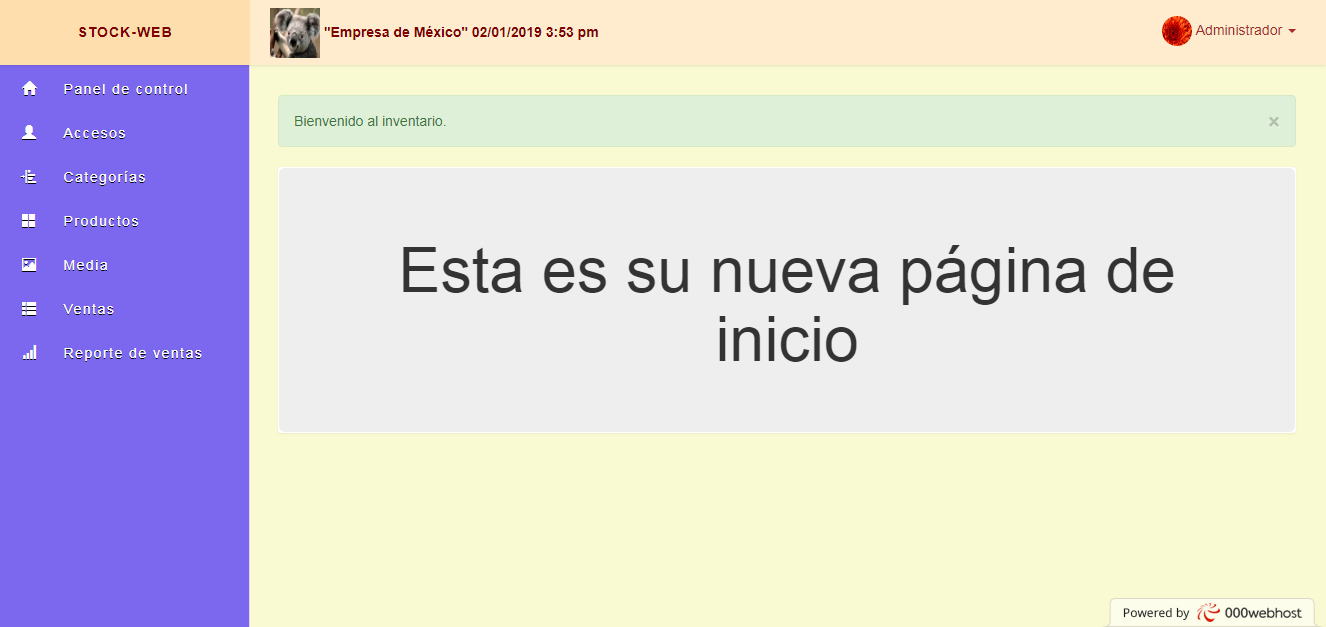


Ilustración 75 Página principal del usuario administrador

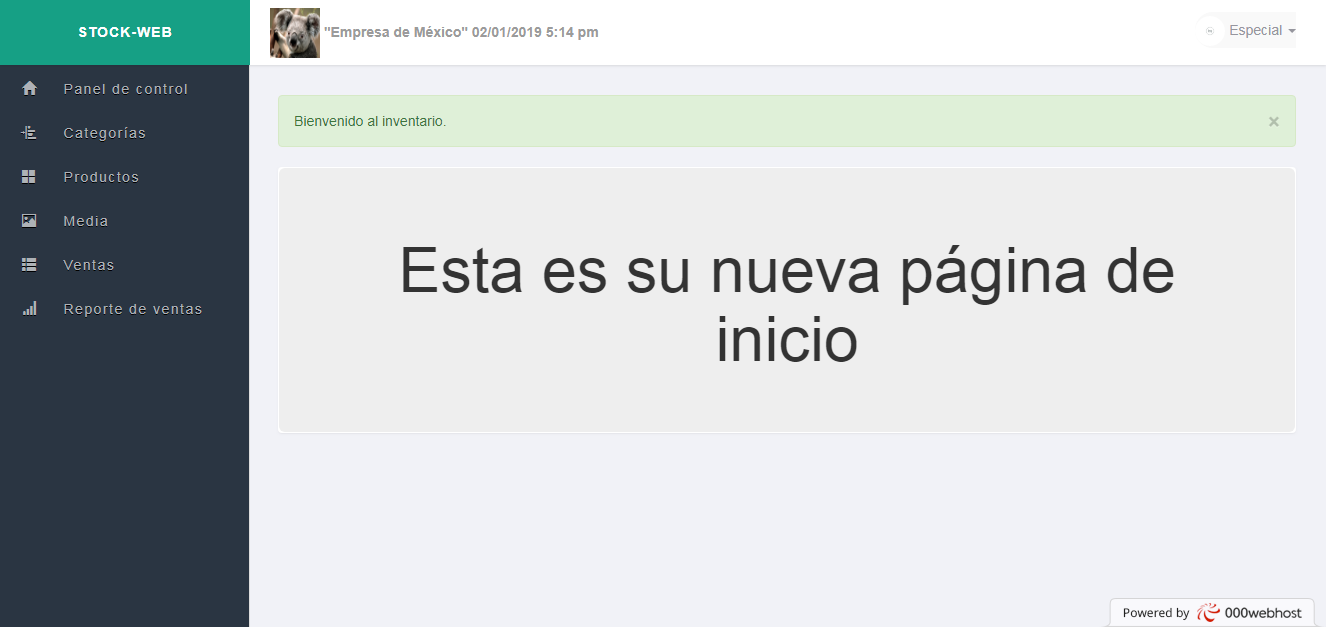


Ilustración 76 Página principal del usuario especial

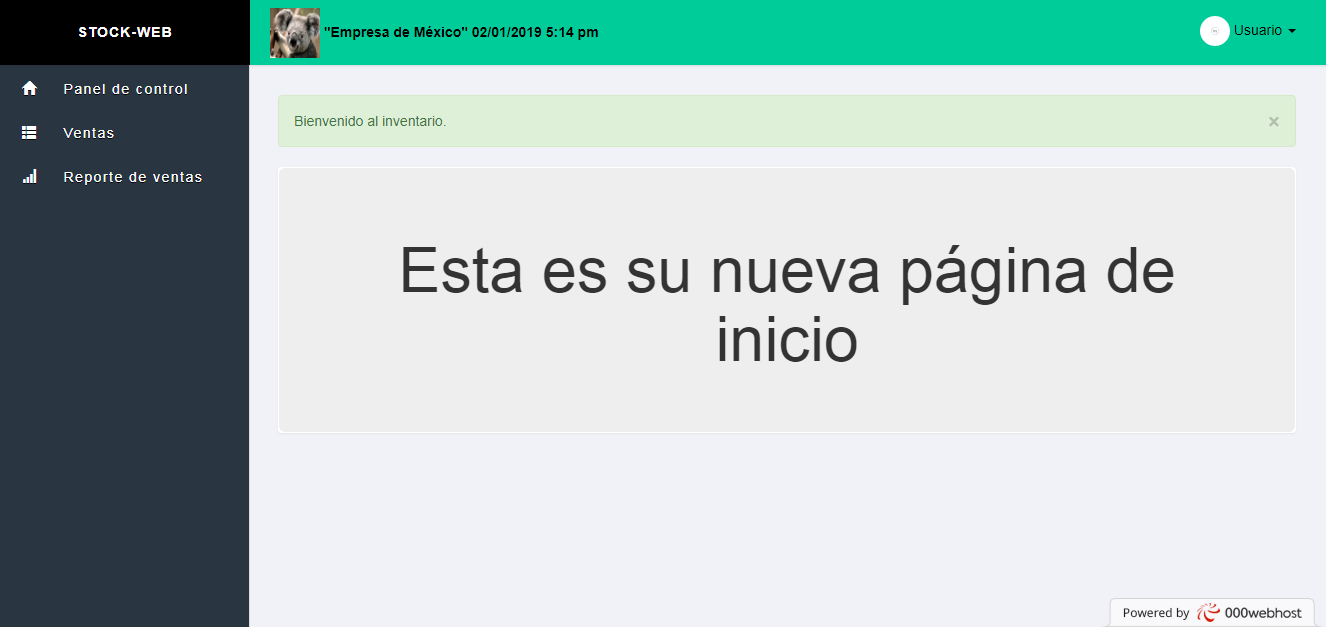


Ilustración 77 Página principal de usuario estándar

### 1.3.4 Panel de control

Esta página tiene en la parte superior en forma de secciones el número de usuarios, el número de categorías, el número de productos y el número de ventas. Debajo los paneles de productos más vendidos, últimas ventas, productos recientemente añadidos, productos con bajo stock y los reportes de ventas automáticos que el usuario desea ver. La sección de productos más vendidos (10 últimos) muestra la información en forma de tabla con el título del producto con enlace para editarlo, así como la cantidad total vendida. La sección de últimas ventas (5 últimas ventas) muestra la información en forma de tabla con el título del producto con enlace para editarlo, la fecha en que se efectuó la venta y el monto de venta total. La sección de productos con bajo stock muestra la información en forma de tabla con el título del producto con enlace para editarlo, así como la cantidad existente del producto; la cantidad existente de umbral debe ser igual a 10 o menor. La sección de productos recientemente añadidos es una lista de los últimos 5 productos añadidos al sistema. Cada fila muestra una imagen del producto, su nombre, el precio y su categoría; además de contener un enlace para editar el producto. Esta página puede ser vista por todos los usuarios, pero los enlaces no pueden ser vistos por los usuarios tipo estándar.

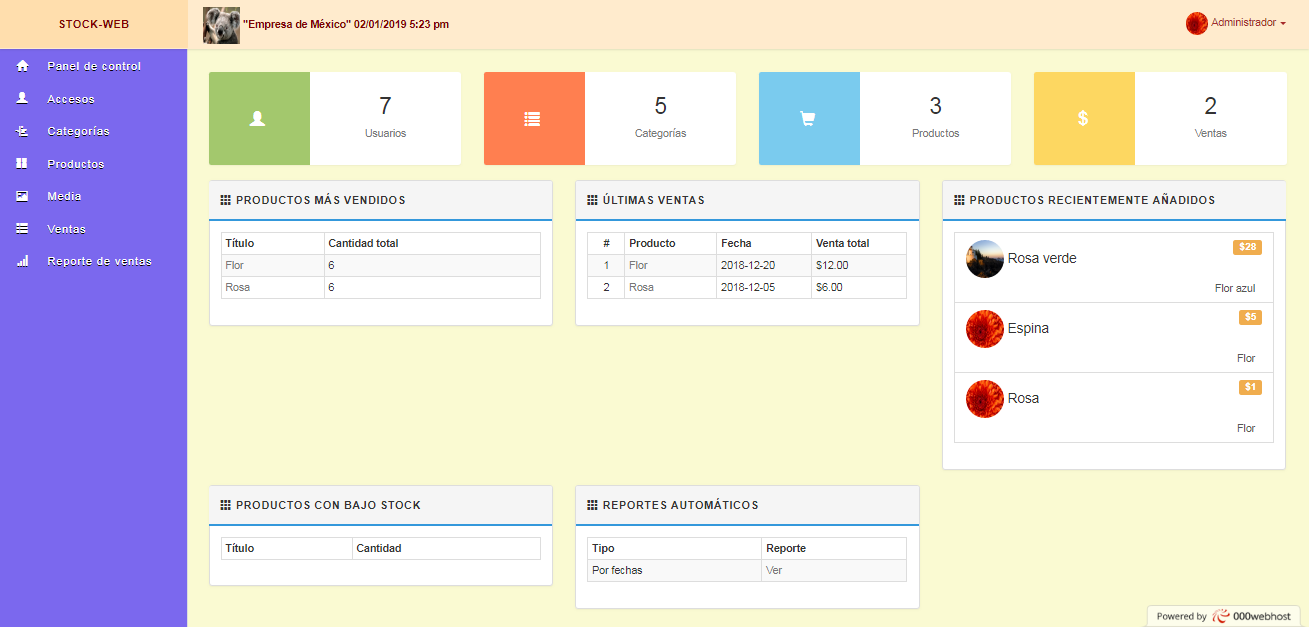


Ilustración 78 Página de panel de control

### 1.3.5 Sección de grupos

Esta página muestra un panel con la tabla con los únicos 3 grupos posibles: administradores, especiales y estándares. La tabla muestra el número de fila, el nombre del grupo, el nivel del grupo (1 es para administradores), estado (activo o inactivo) y acciones (editar o borrar). Para el grupo de administradores solo se cuenta con la acción de editar. En la parte superior derecha del panel se encuentra un botón para agregar un grupo. Esta página solo puede ser vista por los usuarios de tipo administrador. Al momento de dar clic al botón de editar, deberá direccionar a la sección para editar el grupo y al momento de dar clic al botón de eliminar, deberá eliminar el grupo.

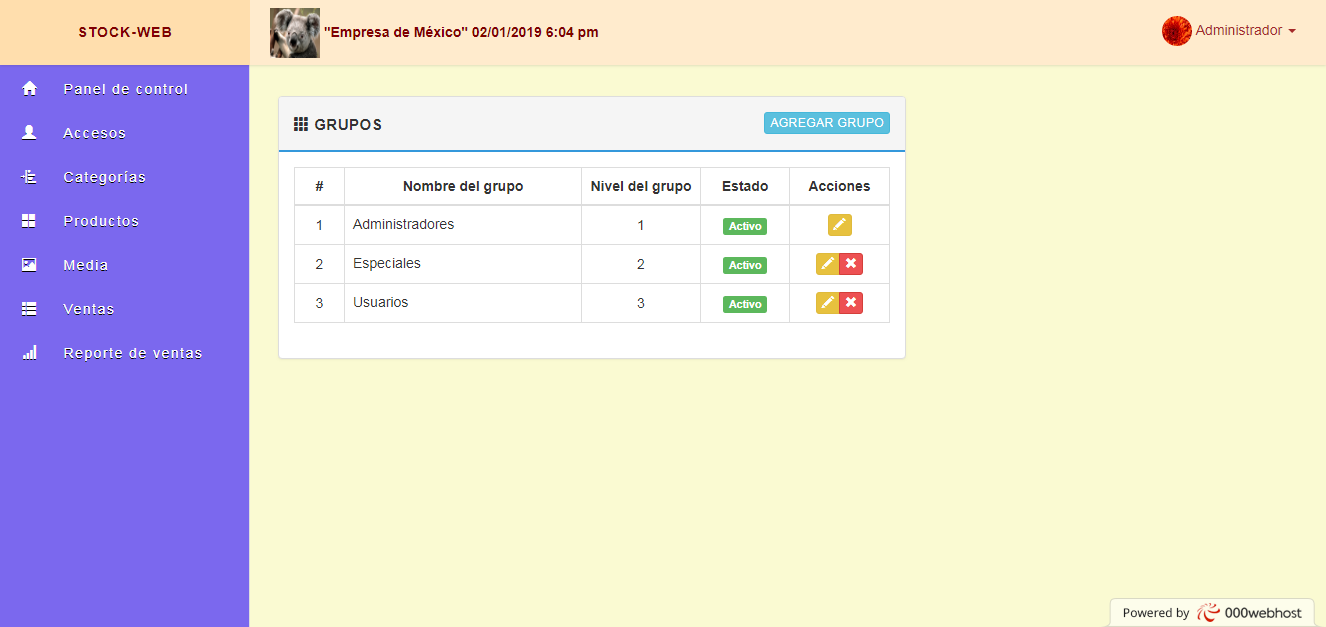


Ilustración 79 Página de grupos

### 1.3.6 Sección para editar grupo

Esta página muestra el formulario para editar el grupo, cuenta con los campos de nombre de grupo, el nivel de grupo y el estado de grupo. Cuando se han hecho las ediciones correspondientes, se da clic en el botón de actualizar; el nombre de grupo no debe estar utilizado ni el nivel de grupo. En el caso del grupo de administradores está prohibida la edición de nivel de grupo y estado de grupo. Esta página solo puede ser vista por los usuarios de tipo administrador.

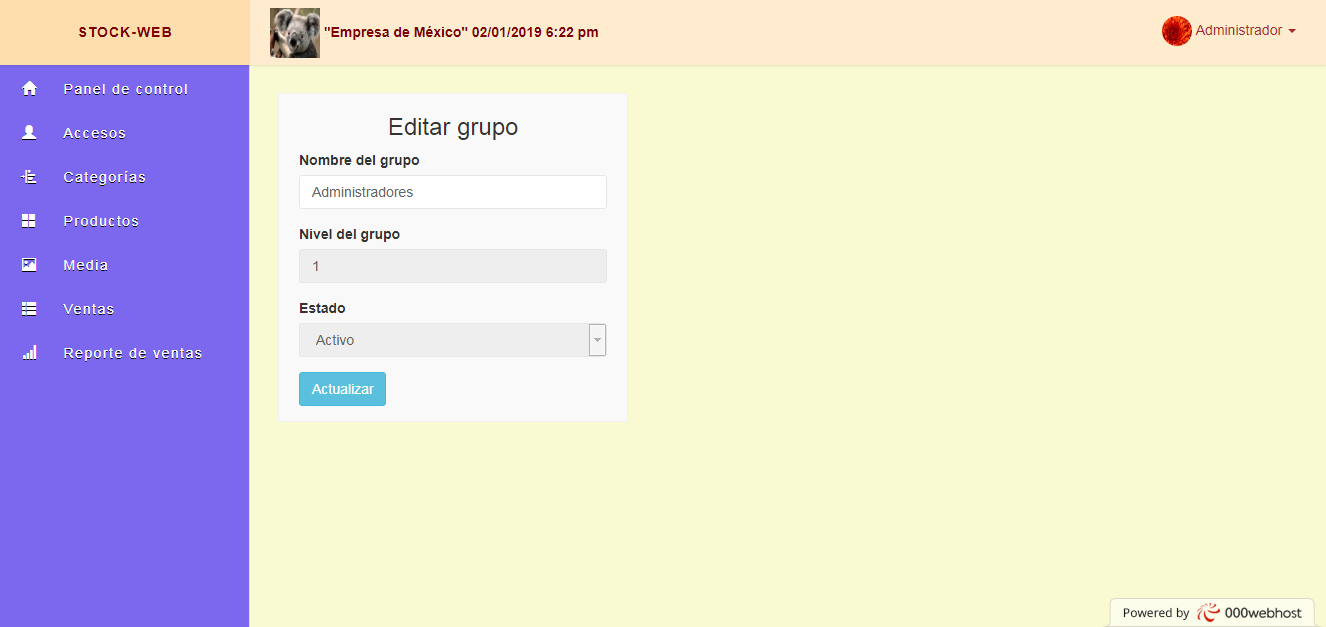


Ilustración 80 Página para editar grupo

### 1.3.7 Sección para agregar grupo

Esta página muestra el formulario para agregar el grupo, cuenta con los campos de nombre de grupo, el nivel de grupo y el estado de grupo. Cuando se han hecho las ediciones correspondientes, se da clic en el botón de guardar; el nombre de grupo no debe estar utilizado ni el nivel de grupo. En el caso del grupo de administradores está prohibida la edición de nivel de grupo y estado de grupo. Esta página solo puede ser vista por los usuarios de tipo administrador.



Ilustración 81 Página para agregar grupo

### 1.3.8 Sección de usuarios

Esta página muestra un panel con la tabla con los 3 tipos de usuarios: administradores, especiales y estándares. La tabla muestra el número de fila, el nombre de la persona dueña de la cuenta, el nombre de usuario, el rol de usuario (administrador, especial y estándar), el estado de la cuenta (activo o inactivo), último acceso (fecha y hora) y acciones (editar o borrar). Para el usuario administrador por defecto y el usuario con sección actual solo se cuenta con la acción de editar. En la parte superior derecha del panel se encuentra un botón para agregar un usuario. Esta página solo puede ser vista por los usuarios de tipo administrador. Al momento de dar clic al botón de editar, deberá direccionar a la sección para editar el usuario y al momento de dar clic al botón de eliminar, deberá eliminar el usuario.

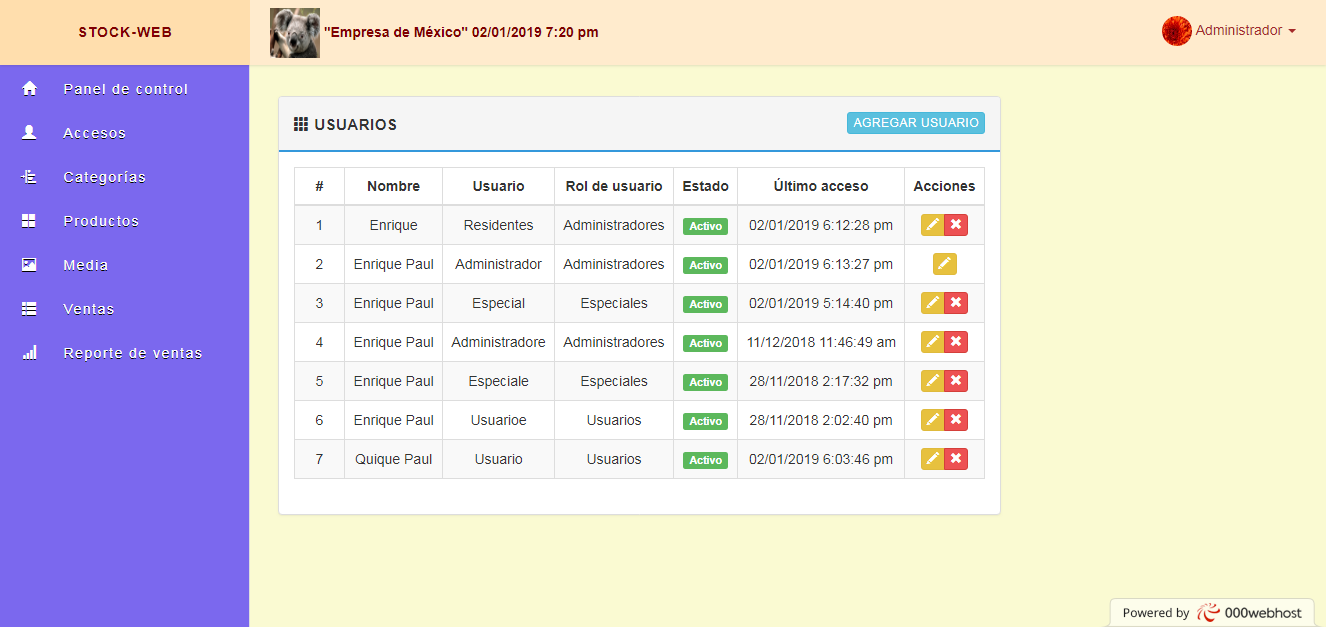


Ilustración 82 Página de usuarios

### 1.3.9 Sección para editar usuario

Esta página muestra el formulario para editar el usuario, cuenta con los campos del nombre de la persona dueña de la cuenta, el nombre de usuario, el rol de usuario y el estado de la cuenta. Cuando se han hecho las ediciones correspondientes, se da clic en el botón de actualizar; el nombre de usuario no debe estar utilizado. En el caso del administrador por defecto y el usuario con sesión actual, está prohibida la edición de el nombre de grupo, nivel de grupo y estado de grupo. Esta página solo puede ser vista por los usuarios de tipo administrador.

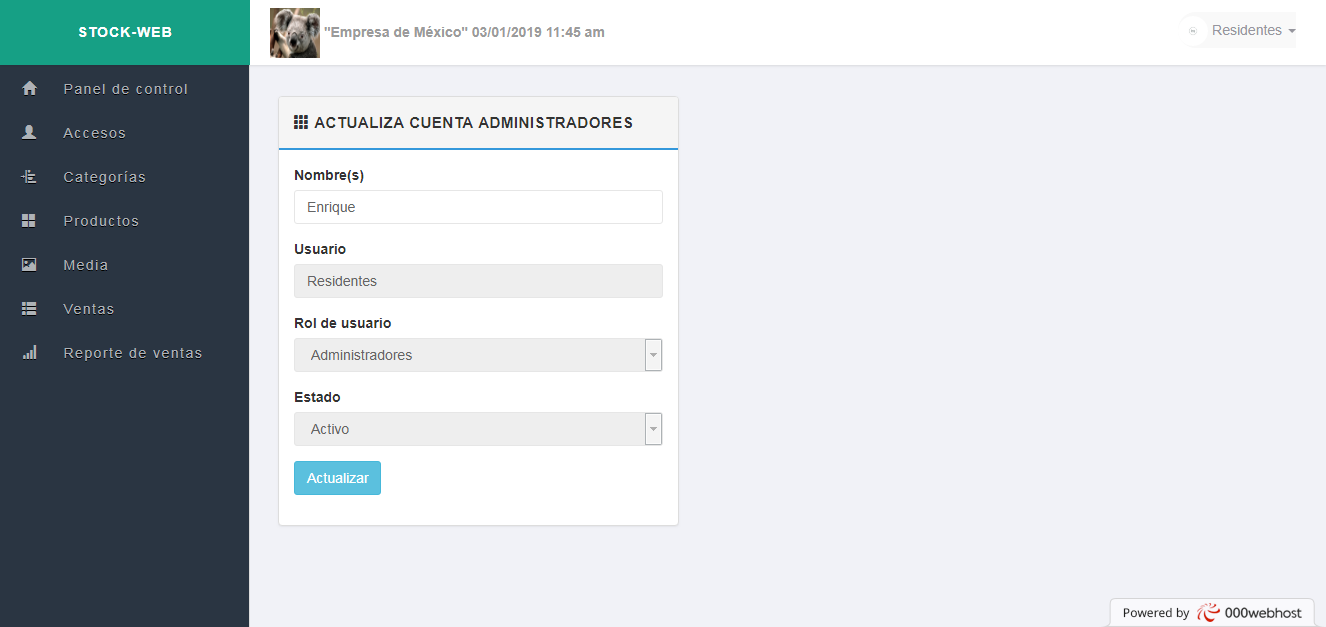


Ilustración 83 Página para editar cuenta de usuario

### 1.3.10 Sección para agregar usuario

Esta página muestra el formulario para agregar un usuario, cuenta con los campos del nombre de la persona dueña de la cuenta, el nombre de usuario, el rol de usuario y el estado de la cuenta. Cuando se han completado los campos, se da clic en el botón de guardar; el nombre de usuario no debe estar utilizado. Esta página solo puede ser vista por los usuarios de tipo administrador.

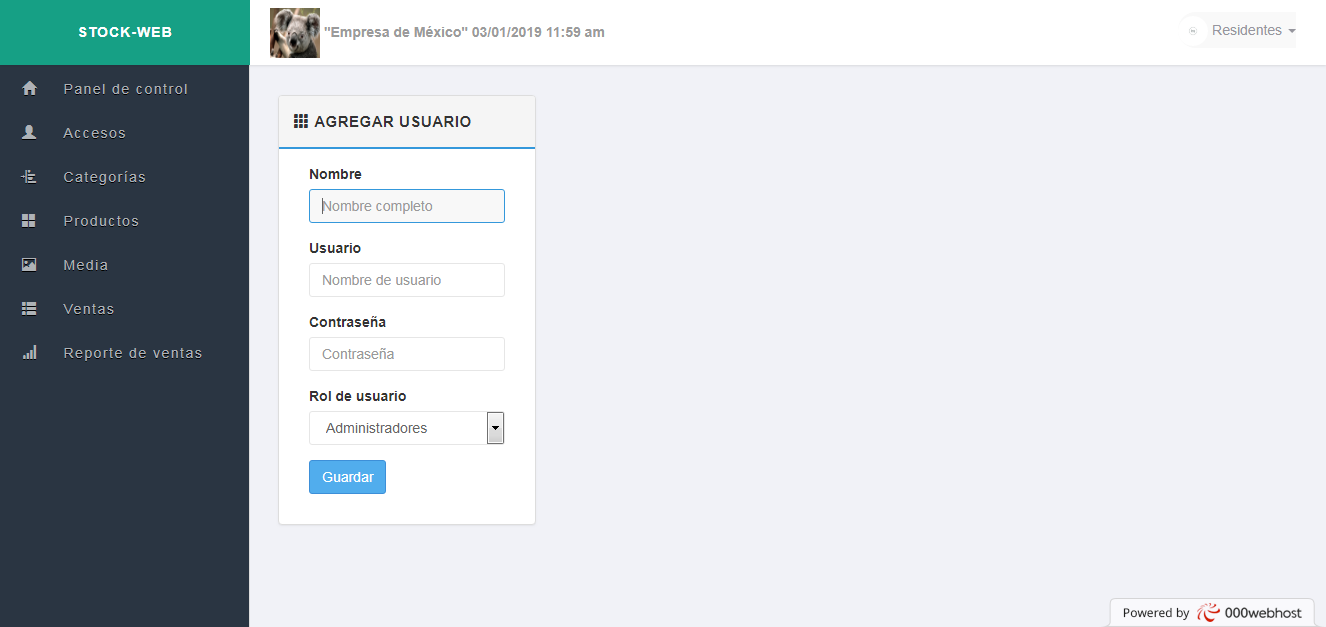


Ilustración 84 Página para agregar usuario

### 1.3.11 Sección de categorías

Esta página muestra un panel con la tabla con las categorías de los productos. La tabla muestra el número de fila, el nombre de la categoría y acciones (editar o borrar). Al momento de dar clic al botón de editar, deberá direccionar a la sección para editar la categoría y al momento de dar clic al botón de eliminar, deberá eliminar la categoría. En la parte izquierda del panel se encuentra otro panel con un formulario para agregar una categoría, cuenta con el campo del nombre de categoría; el nombre de categoría no deberá estar utilizado. Esta página solo puede ser vista por los usuarios de tipo administrador y especial.



Ilustración 85 Página de categorías

### 1.3.12 Sección de editar categoría

Esta página muestra el formulario para editar la categoría, cuenta con el campo del nombre de la categoría. Cuando se ha hecho la edición correspondiente, se da clic en el botón de actualizar categoría; el nombre de categoría no debe estar utilizado. Esta página solo puede ser vista por los usuarios de tipo administrador y especial.

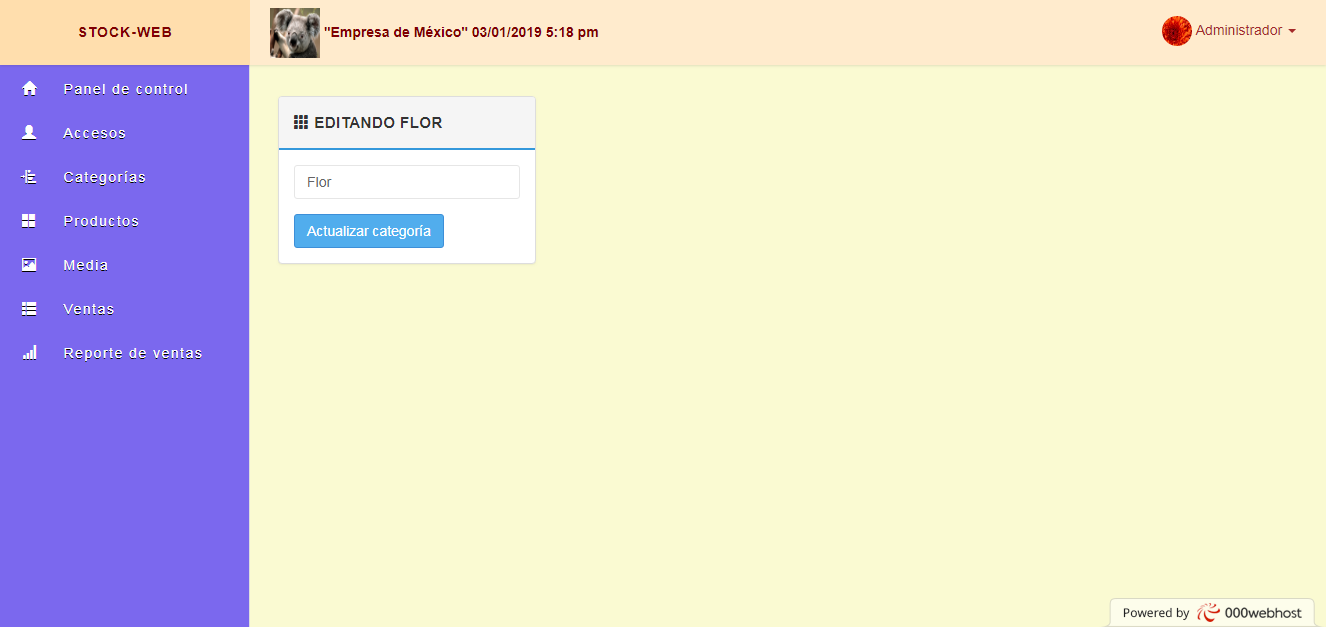


Ilustración 86 Página para editar categoría

### 1.3.13 Sección de productos

Esta página muestra un panel con la tabla de los productos. La tabla muestra el número de fila, la imagen del producto, la descripción del producto, la categoría del producto, precio de compra del producto, precio de venta del producto, fecha y hora en que se actualizó y acciones (editar o borrar). En la parte superior derecha del panel se encuentra un botón para agregar un producto. Esta página solo puede ser vista por los usuarios de tipo administrador y especial. Al momento de dar clic al botón de editar, deberá direccionar a la sección para editar el producto y al momento de dar clic al botón de eliminar, deberá eliminar el producto y mandarlo a la sección de productos eliminados.

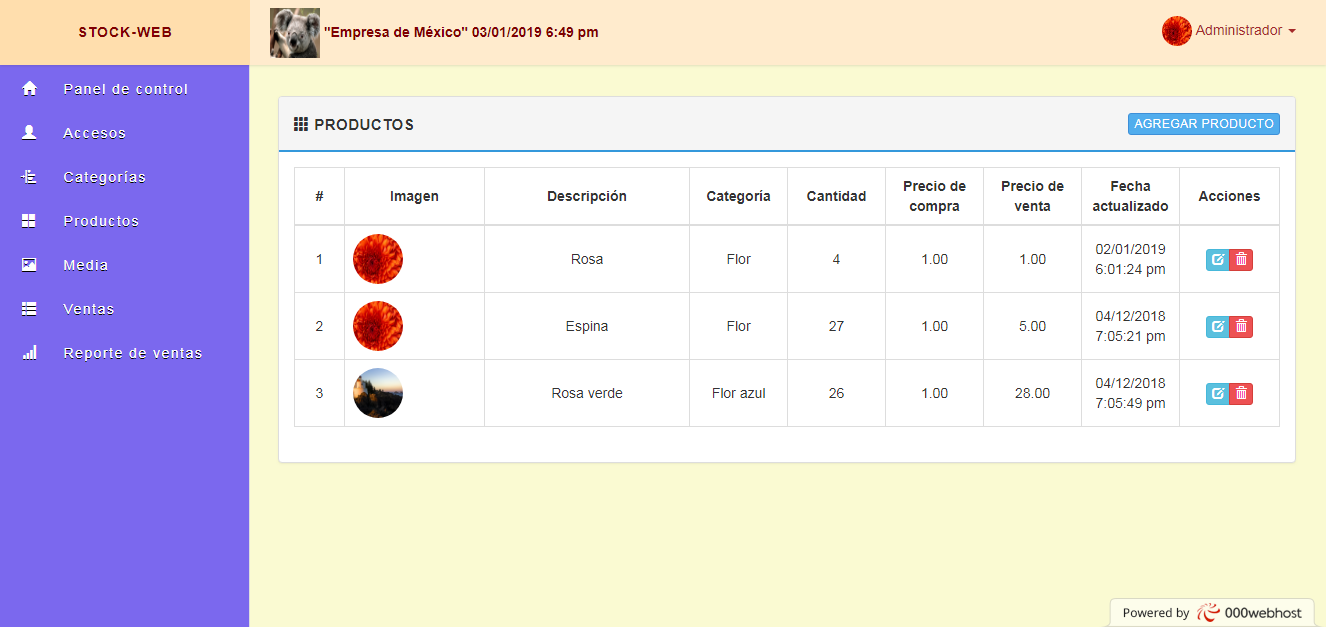


Ilustración 87 Página de productos

### 1.3.14 Sección de editar producto

Esta página muestra el formulario para editar el producto, cuenta con los campos de imagen del producto, la descripción del producto, la categoría del producto, precio de compra del producto, precio de venta del producto. Cuando se han hecho las ediciones correspondientes, se da clic en el botón de actualizar; la descripción de producto no debe estar utilizada. Esta página solo puede ser vista por los usuarios de tipo administrador y especial.

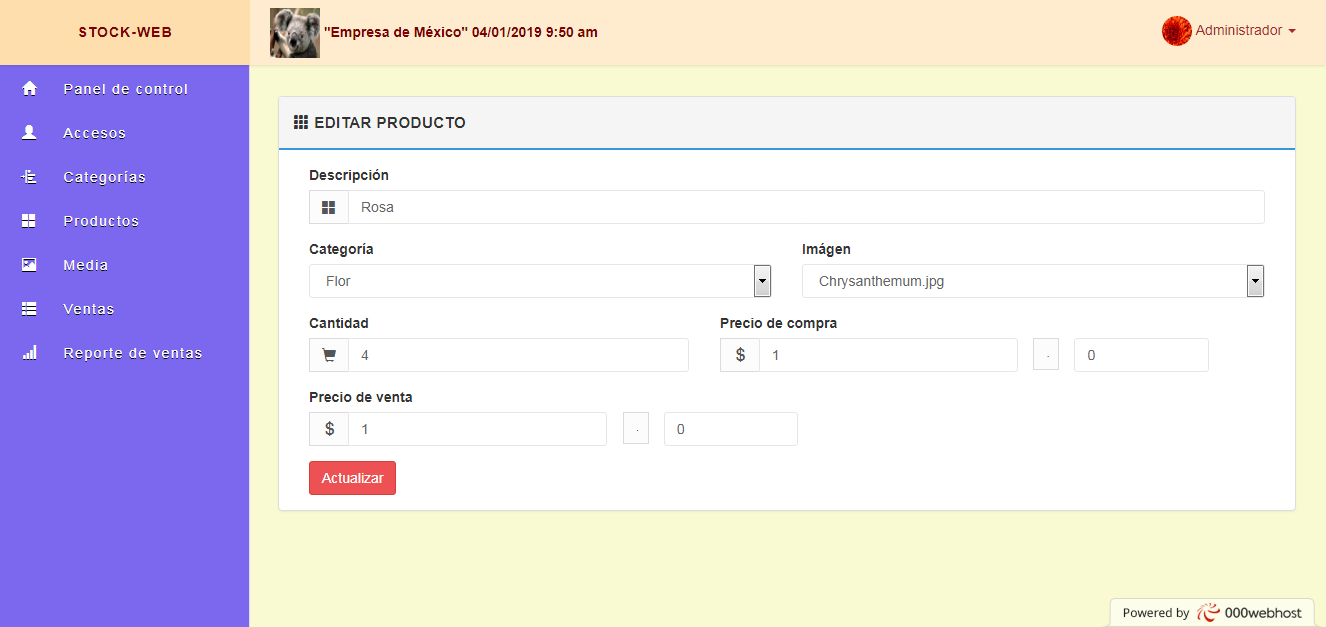


Ilustración 88 Página para editar producto

### 1.3.15 Sección para agregar producto

Esta página muestra el formulario para agregar un producto, cuenta con los campos de imagen del producto, la descripción del producto, la categoría del producto, precio de compra del producto, precio de venta del producto. Cuando se han completado los campos, se da clic en el botón de guardar producto; el nombre de usuario no debe estar utilizado. Esta página solo puede ser vista por los usuarios de tipo administrador y especial.

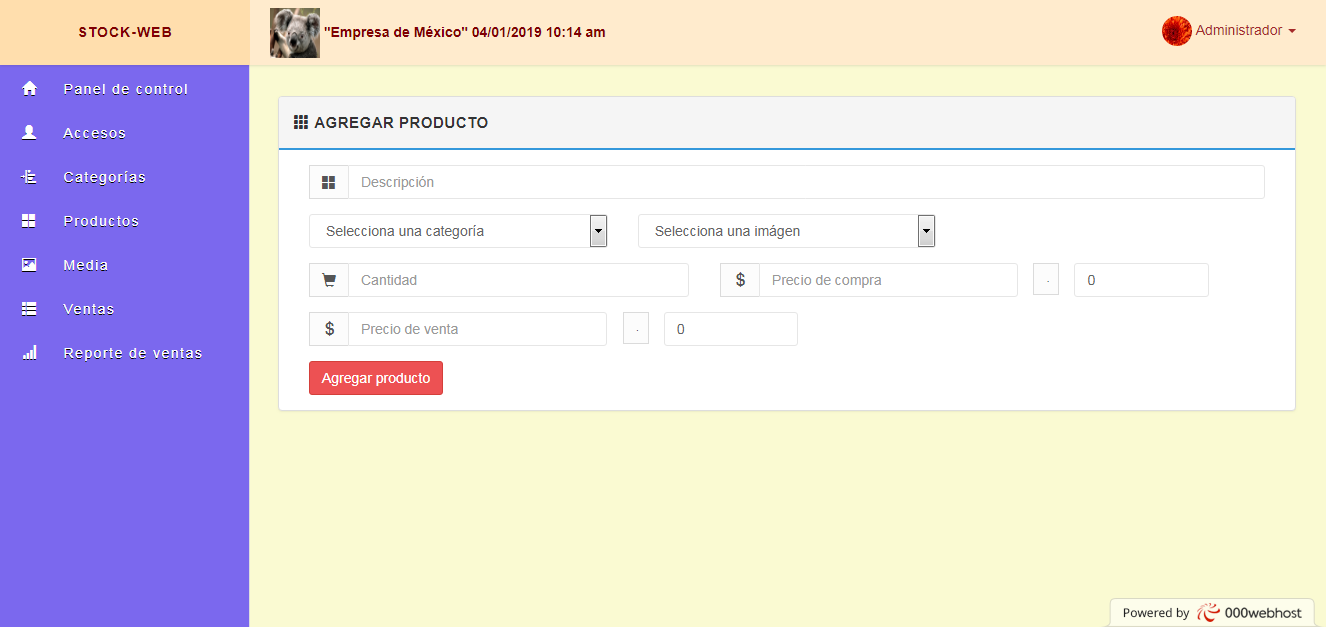


Ilustración 89 Página para agregar producto

### 1.3.16 Sección de productos eliminados

Esta página muestra un panel con la tabla de los productos eliminados. La tabla muestra el número de fila, la imagen del producto, la descripción del producto, la categoría del producto, precio de compra del producto, precio de venta del producto, fecha y hora en que se actualizó y acciones (restaurar o borrar definitivamente). Esta página solo puede ser vista por los usuarios de tipo administrador y especial. Al momento de dar clic al botón de restaurar, deberá restaurar el producto y direccionar a la sección de productos; hay una excepción, si se intenta restaurar un producto, pero hay otro no eliminado con el mismo nombre, no se podrá restaurar el producto y mostrará un mensaje de error en la parte superior del panel. Al momento de dar clic al botón de eliminar, deberá eliminar el producto definitivamente.

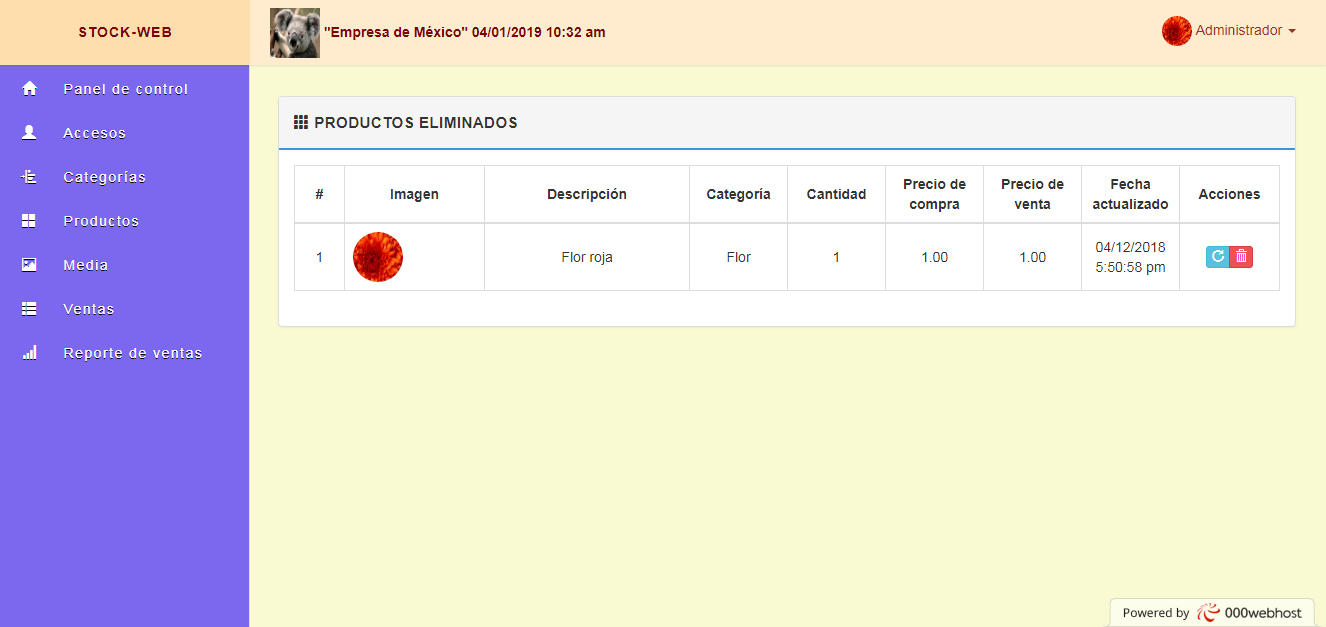


Ilustración 90 Página de productos eliminados

### 1.3.17 Sección de imágenes de productos

Esta página muestra un panel con la tabla de las imágenes de productos. La tabla muestra el número de fila, la imagen, la descripción, el tipo de imagen y poder borrar la imagen. Esta página solo puede ser vista por los usuarios de tipo administrador y especial. En la parte superior derecha se encuentra un par de botones, el primero es para seleccionar una imagen de la computadora del usuario, el segundo es para poder subir dicho archivo. Pueden mostrarse diversos errores al subir una imagen, como que la extensión de la imagen no sea válida (diferente de jpg, png o gif) o algún error del servidor.

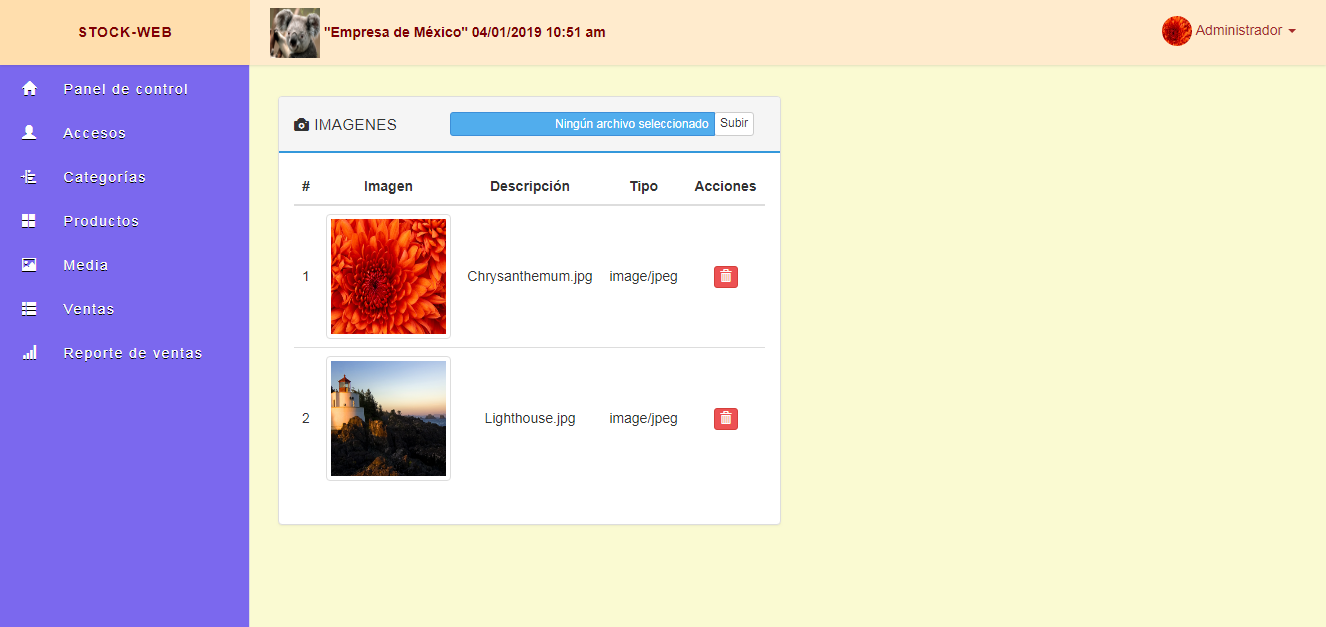


Ilustración 91 Página de imágenes de productos

### 1.3.18 Sección para agregar múltiples registros

Esta página muestra dos paneles para agregar varias categorías y productos. Cada panel muestra los registros actuales y cuenta con un formulario debajo en donde se cuenta con 3 botones, un botón para agregar otro formulario igual y poder llenar varios registros a la vez, otro botón para quitar un formulario y el último botón para mandar el o los formularios para registrar los registros. Puede haber excepciones como que, al momento de agregar un registro, este ya se encuentre registrado. Esta página solo puede ser vista por los usuarios de tipo administrador.

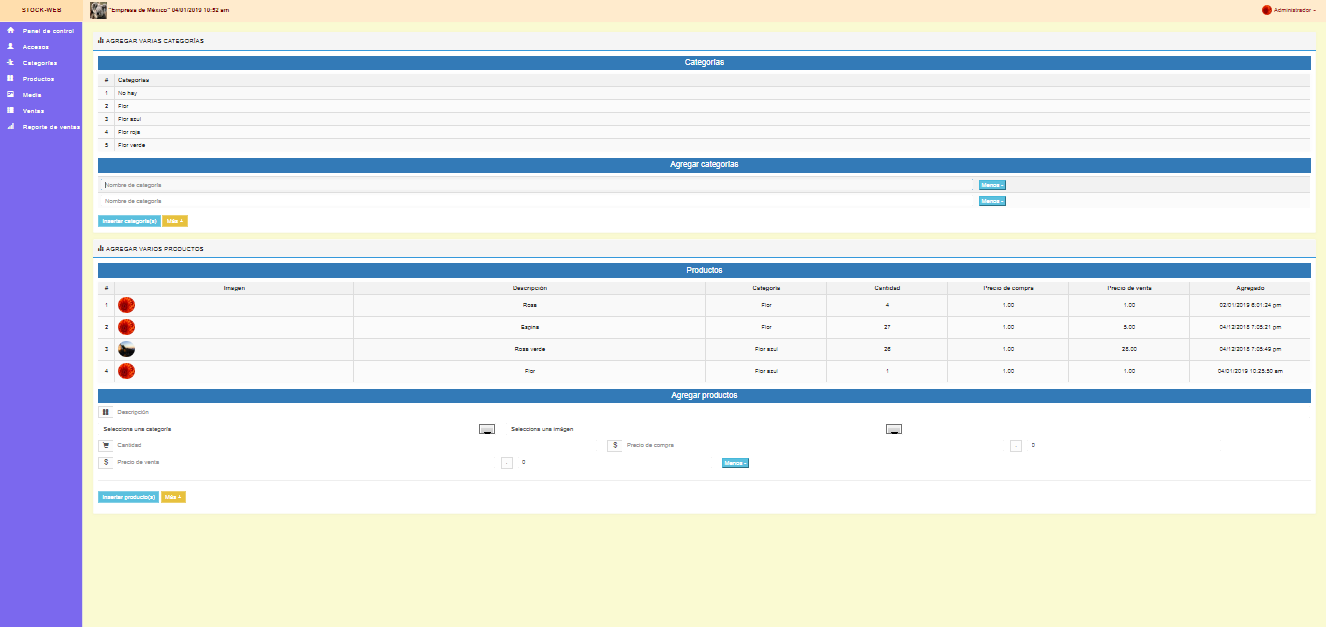


Ilustración 92 Página para agregar múltiples registros

### 1.3.19 Sección de ventas

Esta página muestra un panel con la tabla de las ventas. La tabla muestra el número de fila, el nombre del producto, la cantidad vendida, total de la venta, la fecha en que se realizó la venta, acciones para el usuario tipo administrador (editar o borrar) y el nombre del usuario que registró la venta. En la parte superior derecha del panel se encuentra un botón para agregar una venta. Esta página puede ser vista por todos los usuarios. Al momento de dar clic al botón de editar, deberá direccionar a la sección para editar la venta y al momento de dar clic al botón de eliminar, deberá eliminar la venta.



Ilustración 93 Página de ventas

### 1.3.20 Sección de editar venta

Esta página muestra el formulario para editar la venta, cuenta con el campo de fecha. Cuando se ha hecho la edición correspondiente, se da clic en el botón de actualizar venta. En la parte superior derecha del panel se encuentra un botón para mostrar todas las ventas. Esta página solo puede ser vista por los usuarios de tipo administrador.

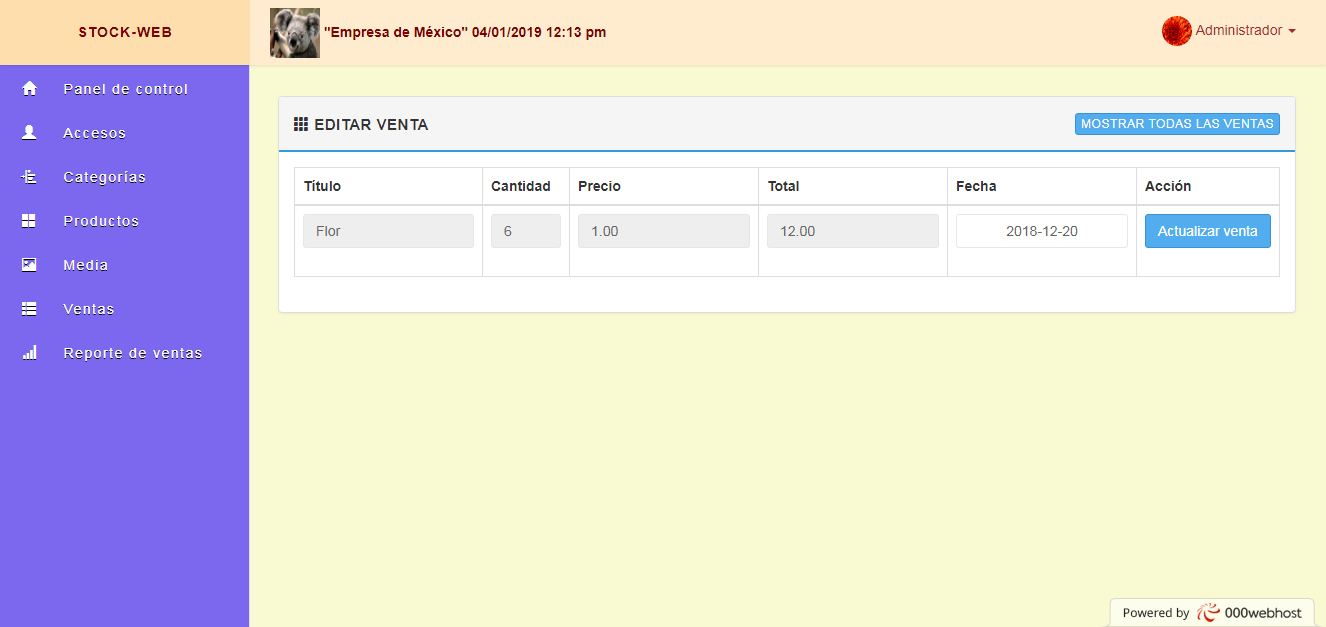


Ilustración 94 Página de editar venta

### 1.3.21 Sección de agregar venta

Esta página muestra el formulario para agregar una venta. Primero se ingresa el nombre del producto en el cuadro de texto que tiene un botón a su izquierda “búsqueda” y si hay una coincidencia muestra los resultados en forma de lista, a continuación, el usuario deberá seleccionar alguno. Al seleccionar un producto deberá ingresar en el formulario de la parte de abajo, la cantidad a vender y la fecha de la venta. Finalmente se da clic en el botón agregar para agregar la venta. Esta página puede ser utilizada por todos los usuarios.

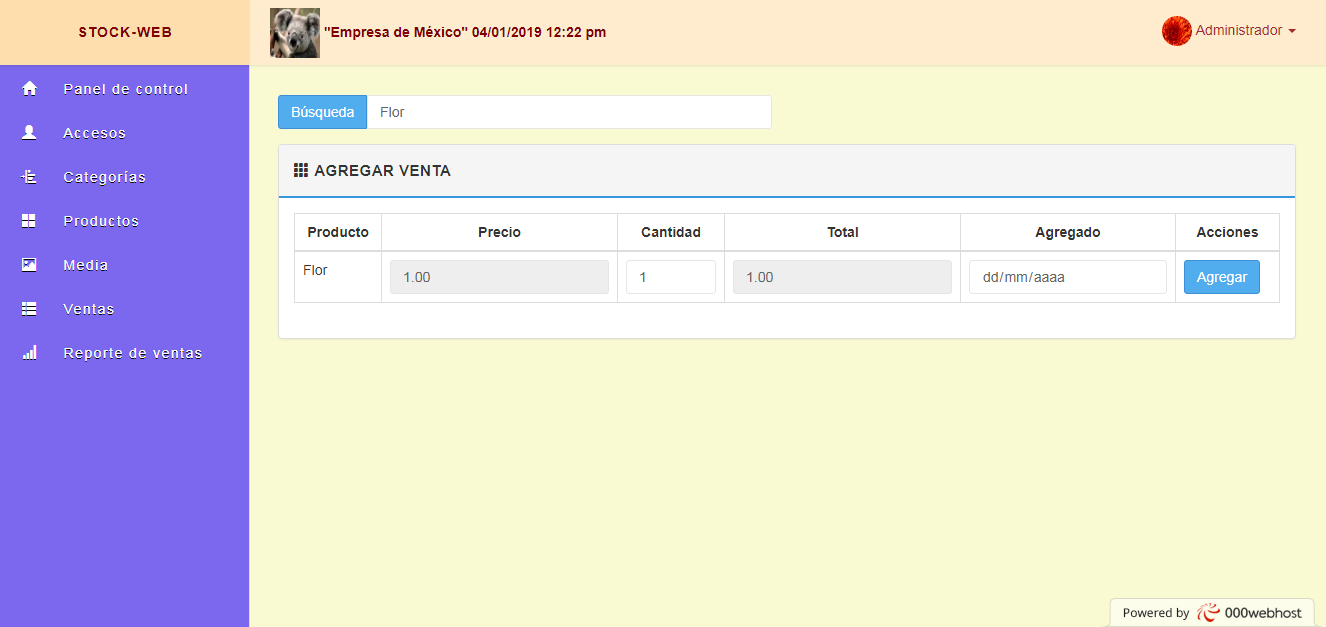


Ilustración 95 Página para agregar venta

### 1.3.22 Sección de reporte de ventas

Esta página muestra un formulario para generar un reporte de ventas. Primero se selecciona una fecha de inicio, después la fecha de fin y se da clic en el botón de generar reporte. Al generar el reporte se nos mostrará una página con el reporte ya generado. Esta página puede ser vista por todos los usuarios.

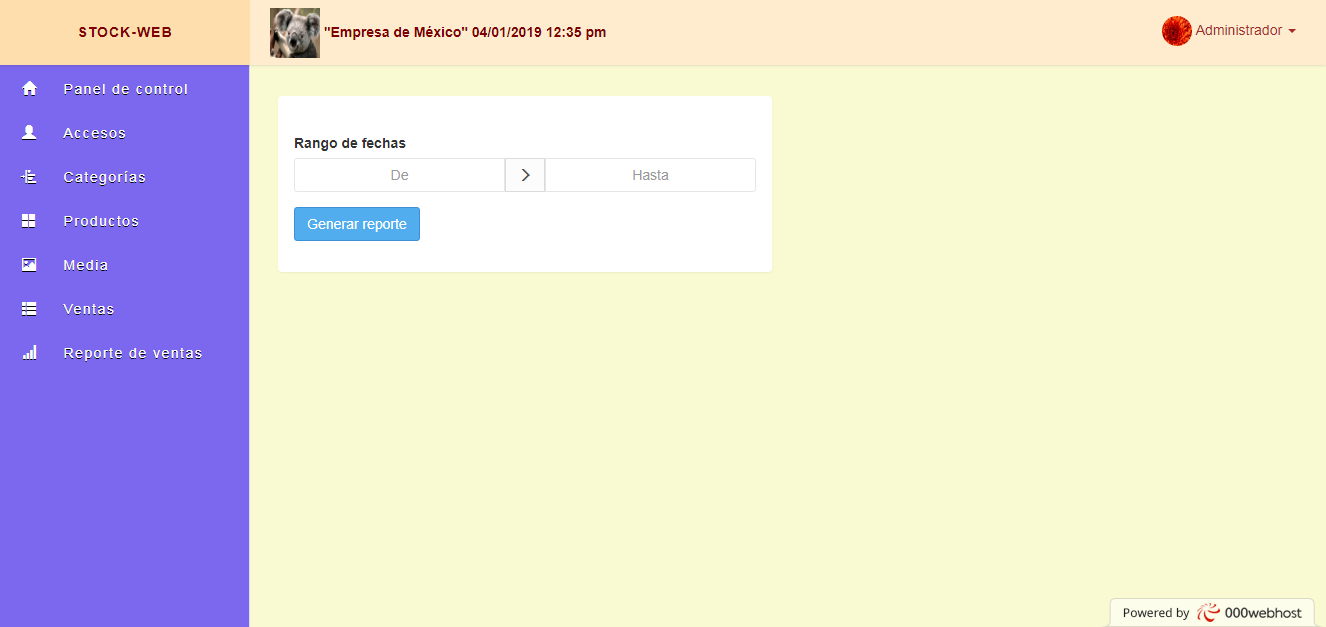


Ilustración 96 Página de reporte de ventas

### 1.3.23 Sección de reporte de ventas generado

Esta página muestra una tabla con las ventas realizadas en el intervalo que el usuario especificó. Por cada venta que se muestra en la tabla, se da la siguiente información: la fecha en que se realizó, la descripción del producto, precio de compra, precio de venta, cantidad total, total y el usuario que la registró. Finalmente se da a conocer el total de todas las ventas, así como su utilidad total.

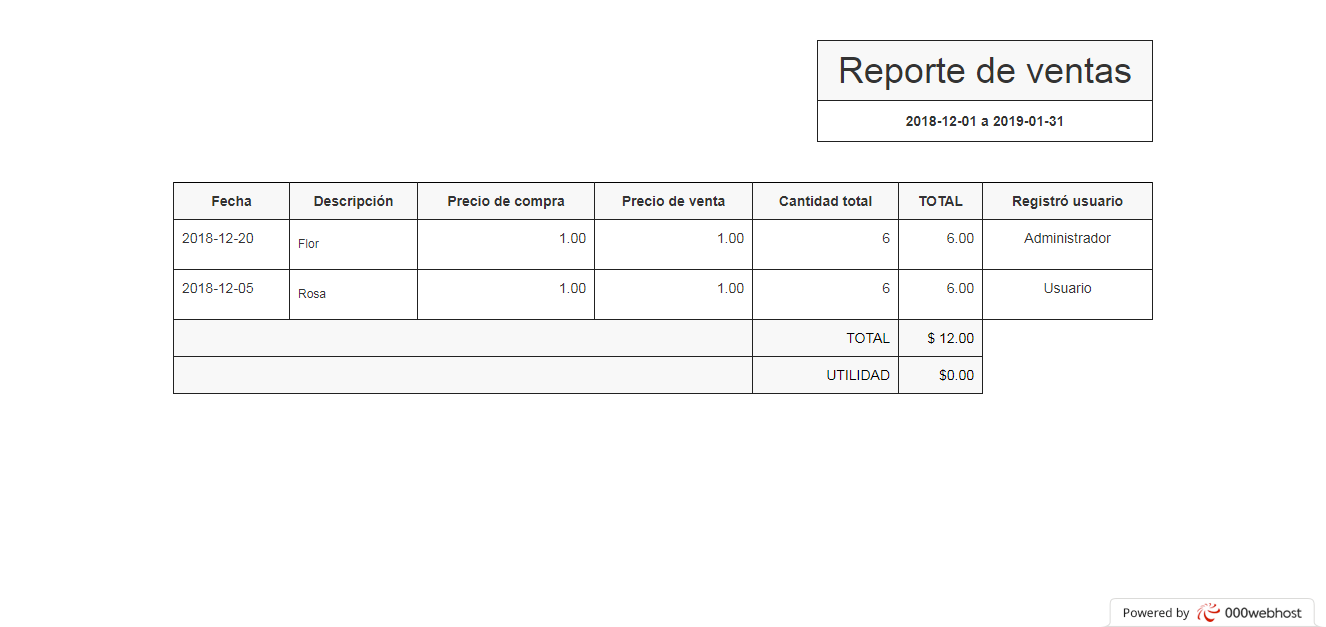


Ilustración 97 Página de reporte de ventas generado

### 1.3.24 Sección de ventas mensuales

Esta página muestra un panel con la tabla de las ventas mensuales. La tabla muestra el número de fila, el nombre del producto, la cantidad vendida, total de la venta, la fecha en que se realizó la venta y el nombre del usuario que registró la venta. Esta página puede ser vista por todos los usuarios.

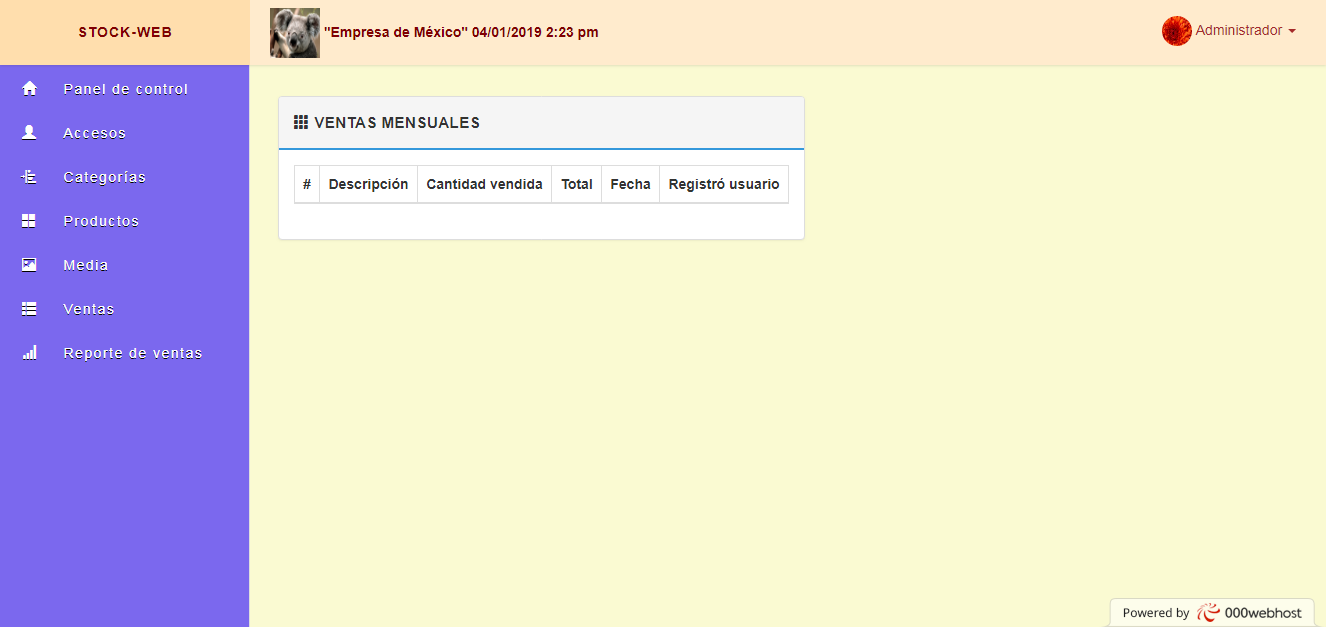


Ilustración 98 Página de ventas mensuales

### 1.3.25 Sección de ventas diarias

Esta página muestra un panel con la tabla de las ventas diarias. La tabla muestra el número de fila, el nombre del producto, la cantidad vendida, total de la venta, la fecha en que se realizó la venta y el nombre del usuario que registró la venta. Esta página puede ser vista por todos los usuarios.

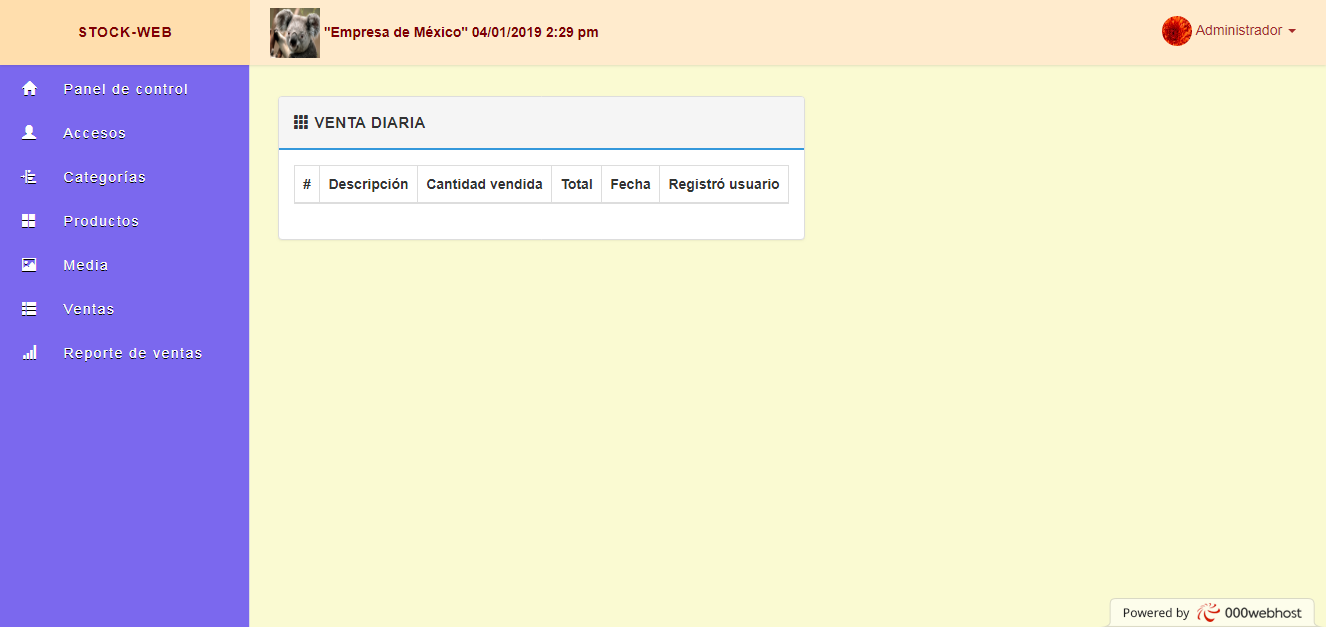


Ilustración 99 Página de ventas diarias

### 1.3.26 Sección de perfil

Esta página muestra un recuadro con la imagen de perfil redondeada sobre un recuadro de color; debajo de la imagen se encuentra el nombre de la persona dueña de la cuenta. Bajo el recuadro se encuentra un enlace para editar la cuenta del usuario. Esta página puede ser utilizada por todos los usuarios.



Ilustración 100 Página de perfil

### 1.3.27 Sección de editar cuenta

Esta página muestra 5 paneles con formularios para editar la foto de perfil, editar el fondo de la aplicación, seleccionar el tema de la aplicación, generar reportes de ventas automáticos y editar la cuenta. Estos 5 paneles pueden ser utilizados por todos los usuarios. El panel para editar la foto de perfil contiene un formulario que requiere que el usuario seleccione una imagen de su computador y la suba por medio del botón “cambiar”, caso contrario si no se quiere esa foto de perfil, se procede a dar clic en el botón “quitar mi foto actual”. El panel para editar el fondo contiene un formulario que requiere que el usuario seleccione una imagen de su computador y la suba por medio del botón “cambiar”, caso contrario si no se quiere esa foto de perfil, se procede a dar clic en el botón “quitar mi fondo actual”. El panel para seleccionar el tema contiene un formulario que requiere que el usuario seleccione un tema de los 5 disponibles y le dé clic al botón “cambiar tema”; por defecto, al crear un usuario, este ya posee un tema por defecto. El panel para generar reportes de ventas automáticos contiene un formulario que requiere que el usuario palomeé los tipos de reportes que necesité. El panel para editar la cuenta contiene un formulario que permite cambiar el nombre de la persona dueña de la cuenta, así como el nombre de usuario y finalmente hay un par de botones, uno es para guardar los cambios y otro para direccionar a la sección de cambiar contraseña. Finalmente hay un panel para cambiar el lema y logo de la empresa exclusivo para los usuarios de tipo administrador. Este panel es la misma estructura que los anteriores paneles.

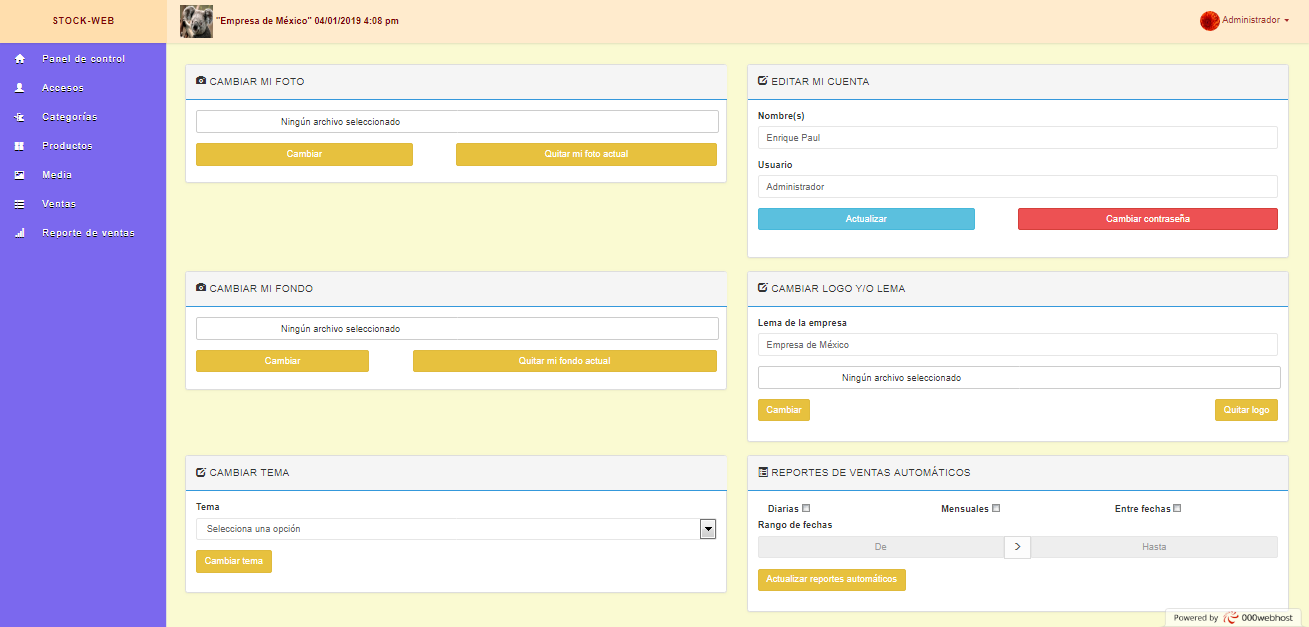


Ilustración 101 Página para editar cuenta

### 1.3.28 Sección para cambiar contraseña

Es la página para cambiar la contraseña de la cuenta. Esta página muestra el formulario para cambiar la contraseña. El usuario deberá completar los campos de antigua y nueva contraseña. Se mostrarán mensajes de error en la parte superior del formulario, si la antigua contraseña no coincide. Una vez lleno el formulario se procede a dar clic en el botón cambiar. Una vez cambiada la contraseña, se le direccionará al usuario a la página principal que vuelva a iniciar sesión.



Ilustración 102 Página para cambiar contraseña

# ANEXOS

1. C# Ya. (s.f.). Obtenido de Tutoriales Ya: https://www.tutorialesprogramacionya.com/csharpya/ [↑](#footnote-ref-1)
2. Iglesias, A. L. (18 de febrero de 2016). Disco duro, ¿Qué es? Obtenido de About Español: https://www.aboutespanol.com/disco-duro-que-es-3866824 [↑](#footnote-ref-2)
3. Gabriel. (14 de noviembre de 2014). *MDN*. Obtenido de HTML: https://developer.mozilla.org/es/docs/Web/HTML [↑](#footnote-ref-3)
4. Merino, J. P. (2011). Definicion.de. Obtenido de Definición de interfaz: https://definicion.de/interfaz/ [↑](#footnote-ref-4)
5. Gardey, J. P. (2010). *Definición de Java*. Obtenido de Definición.de: https://definicion.de/java/ [↑](#footnote-ref-5)
6. Yahoo. (2008). Obtenido de Que significa Loopback en español?: https://mx.answers.yahoo.com/question/index?qid=20061203122018AAQUKjN [↑](#footnote-ref-6)
7. PHP. (s.f.). Obtenido de ¿Qué es PHP?: http://php.net/manual/es/intro-whatis.php [↑](#footnote-ref-7)
8. Gardey, J. P. (2010). *Definicion.de*. Obtenido de Definición de RAM: https://definicion.de/ram/ [↑](#footnote-ref-8)
9. *EcuRed*. (s.f.). Obtenido de SGBD: https://www.ecured.cu/SGBD [↑](#footnote-ref-9)
10. Facturación Web. (21 de abril de 2018). *Facturación Web*. Obtenido de LOS 5 MEJORES SISTEMAS DE GESTIÓN DE INVENTARIO OPEN SOURCE: http://facturacionweb.site/blog/los-5-mejores-sistemas-gestion-inventario-open-source/ [↑](#footnote-ref-10)