

UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

MANUAL TÉCNICO

AUTOR: CHRISTIAN FELIPE REINOSO CHAMBA

DIRECTOR: ING. JOSÉ LUIS RODRÍGUEZ

IBARRA – ECUADOR 2015

ÍNDICE DE CONTENIDOS

ÍNDICE DE CONTENIDOS.....	2
ÍNDICE DE GRÁFICOS.....	2
ÍNDICE DE TABLAS.....	2
ANEXO 1 MANUAL TÉCNICO	3
1.1. DIMFRA WEB	3
1.2. DIMFRA MÓVIL.....	17

ÍNDICE DE GRÁFICOS

Gráfico 1: Raíz del sistema.....	3
Gráfico 2: Estructura del sistema	4
Gráfico 3: Configuración del sistema.....	5
Gráfico 4: Modelo del sistema	7
Gráfico 5: Controlador del sistema.....	10
Gráfico 6: Vistas del sistema.....	14
Gráfico 7: Sistema móvil.....	17
Gráfico 8: Servicios.php móvil.....	18
Gráfico 9: Phonegap.php móvil.....	19

ÍNDICE DE TABLAS

Tabla 1: Configuración del sistema.....	6
Tabla 2: Estructura del modelo.....	9
Tabla 3: Estructura del controlador	13
Tabla 4: Estructura de la vista	17
Tabla 5: Servicios web	18
Tabla 6: Desarrollo en Phonegap	23

ANEXO 1 MANUAL TÉCNICO

El manual técnico dispondrá de la información referencial mínima para el entendimiento del funcionamiento interno del sistema, el mismo que está compuesto de dos partes: DIMFRA WEB y DIMFRA MOVIL.

1.1. DIMFRA WEB

Plataforma:

- LAMP (Linux – Apache – Mysql – PHP -)

Herramientas:

- Eclipse IDE de desarrollo (PHP – HTML-CSS)
- Workbench Gestor de Base de datos de MYSQL
- Framework Code Igniter
- Patrón de diseño.- MVC (Modelo, Vista, Controlador)
- Framework PhoneGap,
- Framework JQuery Mobile, Bootstrap
- JDK de Java, SDK y plugin ADT de Android,
- librerías de Cordova

Directorios

Las carpetas principales del sistema son:

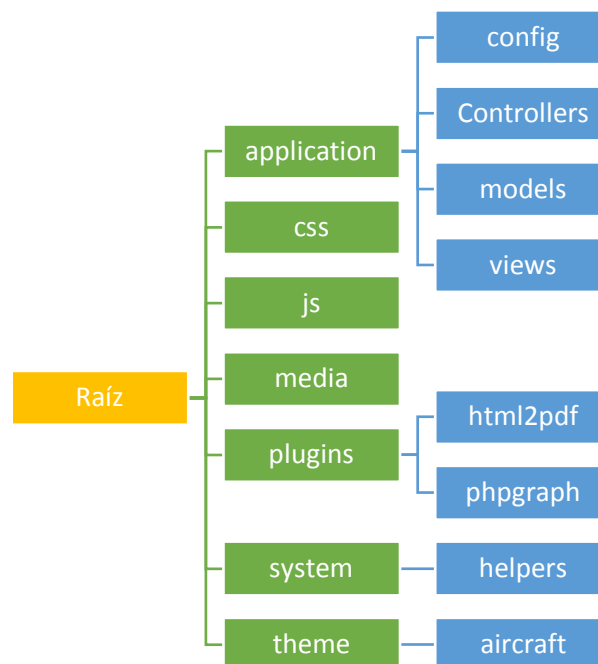


Gráfico 1: Raíz del sistema

Fuente: Propia

Application: Contiene todo el código fuente de la aplicación dentro de esta se encuentran lo subdirectorios:

Config: Contiene los archivos de configuración del sistema como la conexión a la base de datos.

Controllers: Contiene los controladores que controlan las acciones del sistema:

Models: Contiene los modelos de acceso a datos

Views: Contiene las vistas de usuario del sistema, es decir el HTML

Css: Contiene los estilos del sistema para los controles

Js: Contiene las librerías de javascript que utilizan los controles del sistema

Media: Ubicación donde se guardan las imágenes de los productos

Plugins: Contiene plugins para los reportes

System: Contiene el núcleo del CodeIgniter y principalmente la carpeta Helpers, que son librerías para controles como botones, formularios, ayuda de direcciones url

Theme: Contiene los archivos del Tema principal del sistema que se llama Aircraft

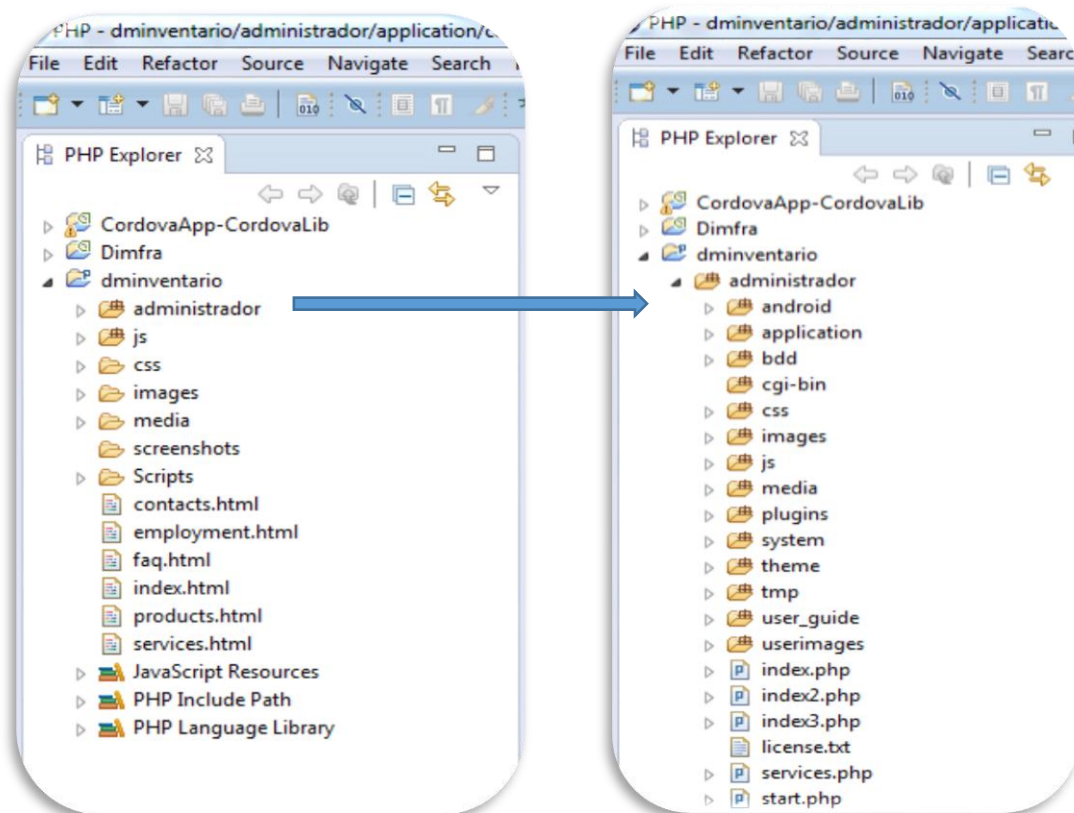


Gráfico 2: Estructura del sistema

Fuente: Propia

Configuración

El archivo de configuración permite establecer los parámetros de conexión con la base de datos es database.php, que se encuentra dentro de la carpeta config.

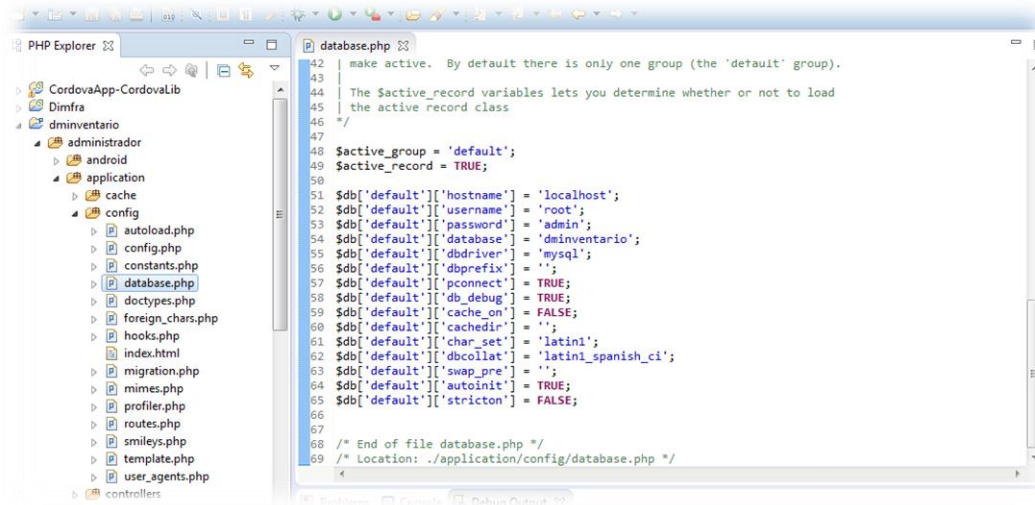


Gráfico 3: Configuración del sistema

Fuente: Propia

Nombre	Database.php
Tipo	Script php
Descripción	Permite el ingreso de la configuración del servidor de base de datos.
Código Fuente:	<pre>\$active_group = 'default'; \$active_record = TRUE; \$db['default']['hostname'] = 'localhost'; \$db['default']['username'] = 'root'; \$db['default']['password'] = 'xxxxxxx'; \$db['default']['database'] = 'dminventario'; \$db['default']['dbdriver'] = 'mysql'; \$db['default']['dbprefix'] = ""; \$db['default']['pconnect'] = TRUE; \$db['default']['db_debug'] = TRUE; \$db['default']['cache_on'] = FALSE; \$db['default']['cachedir'] = ""; \$db['default']['char_set'] = 'latin1';</pre>

```
$db['default']['dbcollat'] = 'latin1_spanish_ci';  
$db['default']['swap_pre'] = '';  
$db['default']['autoinit'] = TRUE;  
$db['default']['stricton'] = FALSE;
```

Tabla 1: Configuración del sistema
Fuente: Propia

Modelos

Los modelos permiten el acceso a la base de datos ya sea para buscar, insertar, actualizar o eliminar.

Los modelos dentro del sistema son:

- adquisicionModel.php
- auditoriaModel.php
- autoModel.php
- bodegaModel.php
- cargoModel.php
- categoriaModel.php
- clienteModel.php
- contenidoModel.php
- detalleadquisicionModel.php
- detalledevolucionModel.php
- detalleencomiendaModel.php
- detallepedidoModel.php
- detalletransferenciaModel.php
- detalleventaModel.php
- devolucionventaModel.php
- empleadoModel.php
- encomiendaModel.php
- estadovehiculoModel.php
- formapagoModel.php
- guiaModel.php
- kardexModel.php
- kardexbodegaModel.php
- marcaModel.php
- moduloModel.php
- pagofacturacompraModel.php
- pagofacturaventaModel.php
- pedidoModel.php
- pedidoventaModel.php
- privilegioModel.php
- productoModel.php
- proveedorModel.php
- reporteModel.php
- rolModel.php
- rolprivilegioModel.php
- transferenciaModel.php
- unidadmedidaModel.php
- usuarioModel.php
- ventaModel.php

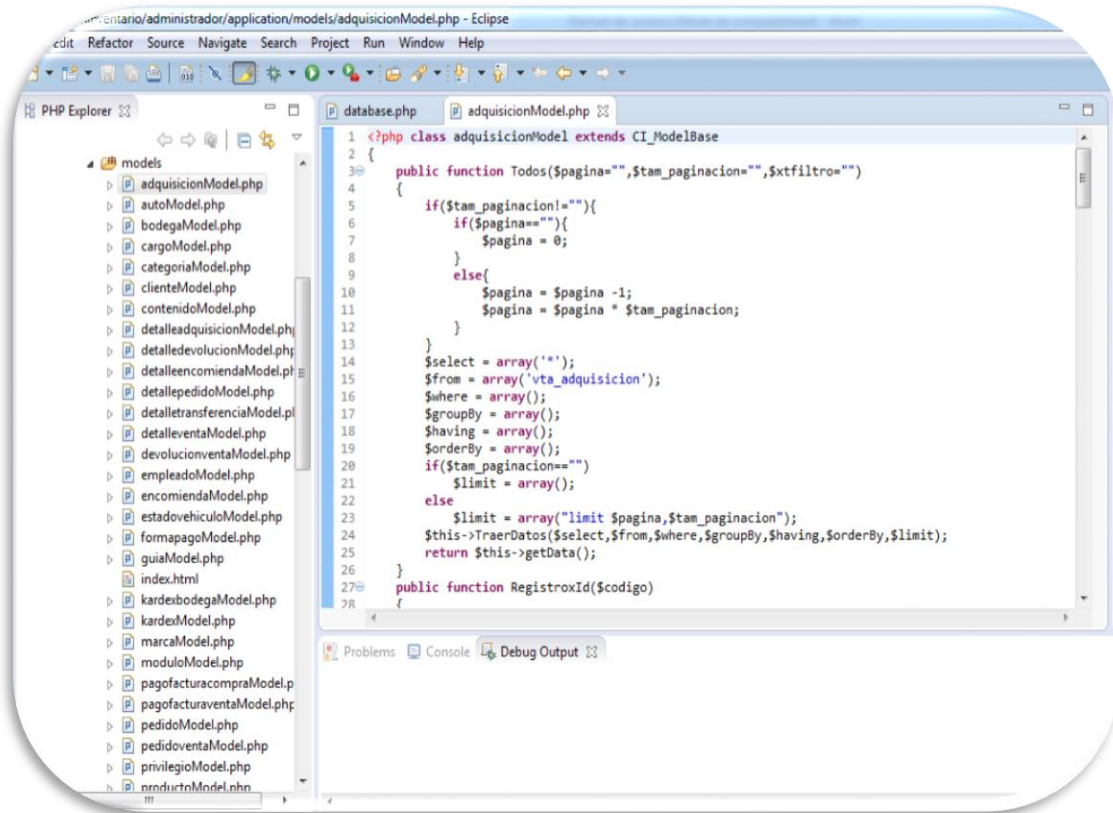


Gráfico 4: Modelo del sistema
Fuente: Propia

Estructura del Modelo

La estructura del modelo en casi todos es similar a la que será descrita a continuación:

Nombre	autoModel
Tipo	Clase heredada de CI_ModelBase
Métodos y Funciones	<p>Todos: Obtiene todos los registros de una consulta de una tabla.</p> <p>RegistroId: Obtiene un registro de la tabla a través de la clave primaria.</p> <p>Insertar: Inserta un nuevo registro</p> <p>Actualizar: Actualiza un registro</p> <p>Eliminar: Elimina un registro</p>
Descripción	Esta clase se encarga de la administración de acceso a datos de la tabla auto.
Código Fuente:	

```

class autoModel extends CI_ModelBase
{
    public function Todos()
    {
        $select = array('*');
        $from = array('vta_auto');
        $where = array();
        $groupBy = array();
        $having = array();
        $orderBy = array();
        $limit = array();
        $this->TraerDatos($select,$from,$where,$groupBy,$having,$orderBy,$limit);
        return $this->getData();
    }
    public function RegistroId($codigo)
    {
        $select = array('*');
        $from = array('vta_auto');
        $where = array("id_auto=$codigo");
        $groupBy = array();
        $having = array();
        $orderBy = array();
        $limit = array();
        $this->TraerDatos($select,$from,$where,$groupBy,$having,$orderBy,$limit);
        $temporal = $this->getData();
        return $temporal[0];}
    publicfunction
    Insertar($id_estado_vehiculo,$placa_auto,$modelo_auto,$matricula_auto,$capacidad
    _auto)
    {
        $this->InsertarRegistro('id_auto',"insert into tbl_auto(id_estado_vehiculo,
    placa_auto, modelo_auto, matricula_auto, capacidad_auto)
    values($id_estado_vehiculo,$placa_auto,$modelo_auto,$matricula_auto,$capacida
    d_auto)");
        if($this->getErrorExiste()==true){
            return false;
        }
        else{
            return true;
        }
    }
    publicfunction
    Actualizar($id_auto,$id_estado_vehiculo,$placa_auto,$modelo_auto,$matricula_auto,
    $capacidad_auto)
    {
        $this->EjecutarSQL("update tbl_auto set id_estado_vehiculo=$id_estado_vehiculo,
    placa_auto=$placa_auto, modelo_auto=$modelo_auto,
    matricula_auto=$matricula_auto, capacidad_auto=$capacidad_auto where
    id_auto=$id_auto");
        if($this->getErrorExiste()==true){
            return false;
        }
    }
}

```



```

        else{
            return true;
        }
    }
    public function Eliminar($codigo)
    {
        $this->EjecutarSQL("delete from tbl_auto where id_auto=$codigo");
        if($this->getErrorExiste()==true){
            return false;
        }
        else{
            return true;}
    }
}

```

Tabla 2: Estructura del modelo
Fuente: Propia

Controladores

Los controladores gestionan las acciones que realizara el sistema, se encargan de llamar a los modelos y las vistas.

Los controladores del sistema son:

- adquisicion.php
- auditoria.php
- auto.php
- bodega.php
- cargo.php
- categoria.php
- cliente.php
- contenido.php
- detalleadquisicion.php
- detalledevolucion.php
- detalleencomienda.php
- detallepedido.php
- detalletransferencia.php
- detalleventa.php
- devolucionventa.php
- empleado.php
- encomienda.php
- estadovehiculo.php
- formapago.php
- guia.php
- index.html
- kardex.php
- kardexbodega.php
- marca.php
- pagofacturacompra.php
- pagofacturaventa.php
- pedido.php
- pedidoventa.php
- phonegap.php
- producto.php
- proveedor.php
- reporte.php

- rol.php
- rolprivilegio.php
- transferencia.php
- unidadmedida.php
- usuario.php
- venta.php

```

1 <?php
2
3 class adquisicion extends CI_Controller
4 {
5     public function index()
6     {
7         $this->listar();
8     }
9     public function listar()
10    {
11        $this->load->model('adquisicionModel','adquisicion');
12        extract(XTUtils::ALLVars());
13        $listado = $this->adquisicion->Todos($this->pagina, $this->tam_paginacion,$xtfiltro);
14        $data['listado'] = $listado;
15        $data['total_registros'] = $this->adquisicion->getContadorfilas();
16        $this->load->view("adquisicion/listar", $data);
17    }
18    public function agregar()
19    {
20        $xtformaction = XTUtils::Request("xtformaction");
21        if($xtformaction=="")
22        {
23            //inicia modo
24            $this->load->model('empleadoModel','listado_empleado');
25            $this->load->model('proveedorModel','listado_proveedor');
26            $this->load->model('formapagoModel','listado_formapago');
27            $data['listado_empleado'] = $this->listado_empleado->Todos();
28            $data['listado_proveedor'] = $this->listado_proveedor->Todos();
29            $data['listado_formapago'] = $this->listado_formapago->Todos();
30
31            $this->load->model('bodegaModel','listado_bodega');
32            $data['listado_bodega'] = $this->listado_bodega->Todos();

```

Gráfico 5: Controlador del sistema
Fuente: Propia

Estructura de Controlador

Casi todos los controladores tienen la estructura que se describe a continuación para auto.php:

Nombre	auto
Tipo	Clase heredada de CI_Controller
Métodos y Funciones	<p>Index: Acción por defecto</p> <p>Listar: Acción que llama los datos de auto y presenta en un listado</p> <p>Agregar: Acción que tiene dos etapas la primera muestra el formulario de agregado y la segunda que lo guarda.</p> <p>Actualizar: Acción que tiene dos etapas la primera muestra el formulario de edición y la segunda que lo actualiza.</p> <p>Eliminar: Acción que elimina un dato</p>

Descripción	Esta clase se encarga de gestionar las acciones para auto
Código Fuente:	
<pre> class auto extends CI_Controller { public function index() { \$this->listar(); } public function listar() { \$this->load->model('autoModel','auto'); extract(XTUtils::AllVars()); \$listado = \$this->auto->Todos(\$this->pagina, \$this-> tam_paginacion,\$xtfiltro); \$data['listado'] = \$listado; \$data['total_registros'] = \$this->auto->getContadorfilas(); \$this->load->view("auto/listar", \$data); } public function agregar() { \$xtformaction = XTUtils::Request("xtformaction"); if(\$xtformaction=="") { //inicia modo extract(XTUtils::AllVars()); \$this->load->model('estadovehiculoModel','listado_estadovehiculo'); \$this->load->model('estadovehiculoModel','listado_estadovehiculo'); \$data["listado_estadovehiculo"] = \$this->listado_estadovehiculo->Todos(); \$data["listado_estadovehiculo"] = \$this->listado_estadovehiculo->Todos(); \$data["XTFMODE"] = 'ADD'; \$this->load->view("auto/form",\$data); } elseif(\$xtformaction=="GUARDAR"){ //inicia modo de guardado \$this->load->model('autoModel','auto'); extract(XTUtils::AllVars()); \$tranret = \$this->auto-> Insertar(\$sid_estado_vehiculo,\$placa_auto,\$modelo_auto,\$matricula_auto,\$capacida d_auto); </pre>	

```

        if($stranret==true){
            $datareturn[]=array('estado'=>'OK','parametro'=>$this->auto->last_insert);
            echo array2json($datareturn);
        }
        else{
            $datareturn[]=array('estado'=>'ERROR','parametro'=>$this->auto->getErrorMensaje());
            echo array2json($datareturn);
        }
    }
}

public function editar()
{
    $this->load->model('autoModel','auto');
    $xtformaction = XTUtils::Request("xtformaction");
    if($xtformaction=="")
    {
        $sid_auto = XTUtils::Request("codigo");
        if($sid_auto==null){
            die('No se ha seleccionado ningun registro');
        }
        extract(XTUtils::AllVars());
        $localfila = $this->auto->RegistroId($sid_auto);
        $data['localfila'] = $localfila;
    }
    $this->load->model('estadovehiculoModel','listado_estadovehiculo');
    $this->load->model('estadovehiculoModel','listado_estadovehiculo');
    $data[listado_estadovehiculo] = $this->listado_estadovehiculo->Todos();
    $data[listado_estadovehiculo] = $this->listado_estadovehiculo->Todos();
    $data["XTFMODE"] = 'EDIT';
    $this->load->view("auto/form",$data);
    //inicia modo
    }
    elseif($xtformaction=="GUARDAR"){
        //inicia modo de guardado
        extract(XTUtils::AllVars());
        $stranret = $this->auto->Actualizar($sid_auto,$sid_estado_vehiculo,$placa_auto,$modelo_auto,$matricula_auto,$capacidad_auto);
        if($stranret==true){
            $datareturn[]=array('estado'=>'OK','parametro'=>");
            echo array2json($datareturn);
        }
    }
}

```

```

        else{

            $datareturn[]=array('estado'=>'ERROR','parametro'=>$this->auto-
>getErrorMensaje());

                echo array2json($datareturn);
            }
        }
    }
    public function eliminar()
    {
        $this->load->model('autoModel','auto');
        $eliminarcodigo = XTUtils::Request("eliminarcodigo");
        $stranret = $this->auto->Eliminar($eliminarcodigo);
        if($stranret==true){
            $datareturn[]=array('estado'=>'OK','parametro'=>");
            echo array2json($datareturn);
        }
        else{
            $datareturn[]=array('estado'=>'ERROR','parametro'=>$this->auto-
>getErrorMensaje());
            echo array2json($datareturn);
        }
    }
}

```

Tabla 3: Estructura del controlador

Fuente: Propia

Vistas

Al igual que en modelos y controladores existe una carpeta individual para cada controlador de las vistas que en su mayoría están compuestos de 4 archivos:

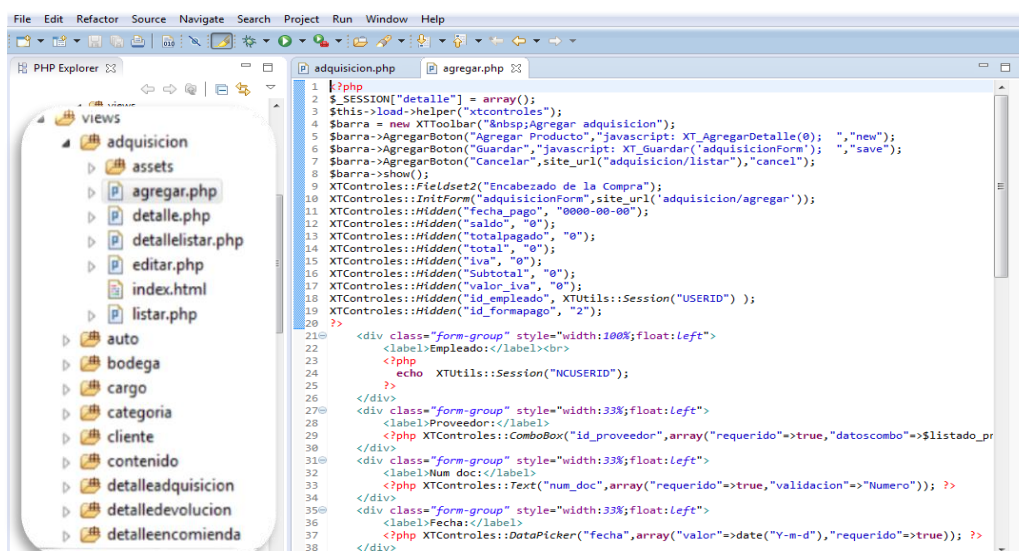


Gráfico 6: Vistas del sistema
Fuente: Propia

Listar.php	Muestra un listado con filtro de búsqueda
	<pre> <?php \$this->load->helper("xtcontroles"); \$barra = new XTToolbar("Categorías de Productos"); \$barra->AgregarBoton("Agregar Categoria",site_url("categoria/agregar"),"new"); XTControles::FSDDataTable(\$barra); ?> <table id="rounded-corner"> <thead> <tr> <th style="width:10px"></th> <th>Categoria</th> <th style="width:25px">Editar</th> <th style="width:25px">Eliminar</th> </tr> </thead> </pre>

	<pre> <?php for(\$i=0;\$i<count(\$listado);\$i++): if(is_array(\$listado[\$i])) extract(\$listado[\$i], EXTR_PREFIX_ALL, 'var'); \$stilolocalfila = (\$i % 2 == 0)?'odd':'even'; \$i_item_pos = (XTUtils::Request("xtpagcon") == "?1:XTUtils::Request("xtpagcon"); \$i_item_pos = (\$i+1+ ((\$i_item_pos-1)*XTRECORD_x_PAGES)); ?> <tr class="<?php echo \$stilolocalfila?>"> <td><?echo \$i_item_pos?></td> <td><?php echo \$var_nombre_categoria?></td> <td><?php XT_LinkEditImage(\$this,array("accion"=>"editar","params"=>"co digo=\$var_id_categoria")) ?></td> <td><a href="javascript: XT_Eliminar('<?php echo base_url()."index3.php/".\$this->router->class."/eliminar"?>',<?php echo \$var_id_categoria?>)" ><img src="<?php echo base_url()?>images/trash.gif" alt="" title="" border="0"/> </td> </tr> <?php endfor; ?> </table> <?php XTControles::EndFSDataTable(); ?> </pre>
Agregar.php	Crear la barra de herramientas para agregar
	<pre> <?php \$this->load->helper("xtcontroles"); \$barra = new XTToolbar("&nbsp;Agregar categoria"); </pre>

	<pre> \$barra->AgregarBoton("Guardar","javascript: XT_Guardar('categoriaForm'); ","save"); \$barra>AgregarBoton("Cancelar",site_url("categoria/listar"),"cancel"); XTControles::InitForm("categoriaForm",site_url('categoria/agregar')); callformview(get_defined_vars()); XTControles::EndForm(); \$barra->showrecord(); ?> </pre>
Editar.php	Crea la barra de herramientas para editar
	<pre> <?php \$this->load->helper("xtcontroles"); \$barra = new XTToolbar("&nbsp;Editar categoria"); \$barra->AgregarBoton("Actualizar","javascript: XT_Guardar('categoriaForm'); ","save"); \$barra>AgregarBoton("Cancelar",site_url("categoria/listar"),"cancel"); XTControles::InitForm("categoriaForm",site_url('categoria/editar')); callformview(get_defined_vars()); XTControles::EndForm(); \$barra->showrecord(); ?> </pre>
Form.php	Formulario con los objetos de controles para la inserción de información.
	<pre> <?php XTControles::Hidden("id_categoria",\$var_id_categoria); ?> <div class="form-group"> <label>Nombre categoria:</label> </pre>

	<pre><?php XTControles::Text("nombre_categoria",array("valor"=>\$var_nombre_categoria,"requerido"=>true)); ?> </div></pre>
--	--

Tabla 4: Estructura de la vista
Fuente: Propia

1.2. DIMFRA MÓVIL

Para el desarrollo del aplicativo móvil se tiene dos secciones:

- Creación de servicios web en la aplicación web
- Creación de la aplicación Móvil con phonegap

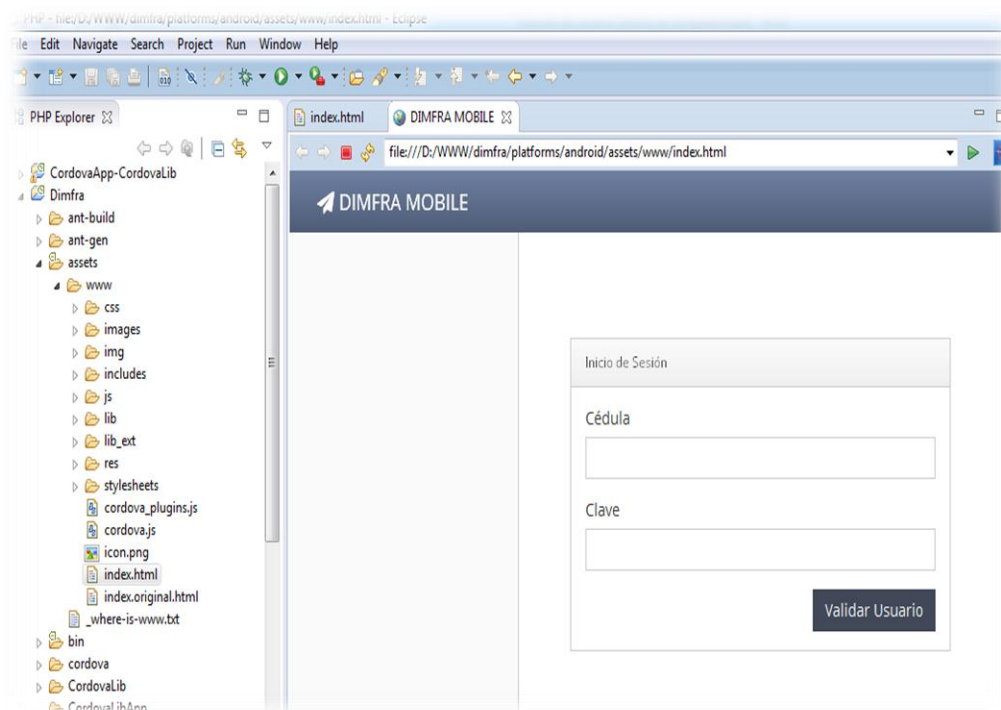


Gráfico 7: Sistema móvil
Fuente: Propia

Servicios web

Para la creación de un repositorio de servicios existen 2 archivos:

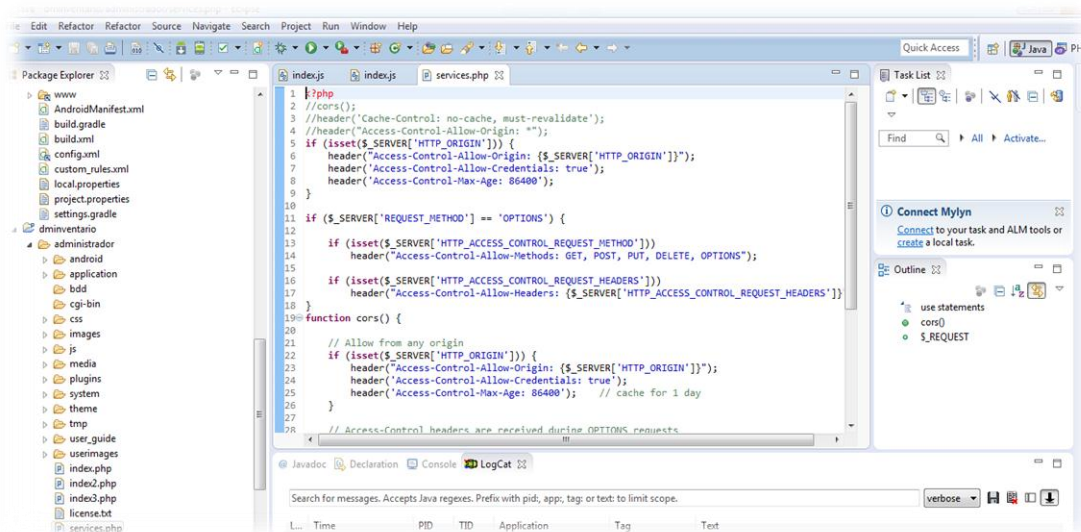


Gráfico 8: Servicios.php móvil

Fuente: Propia

Services.php Se encuentra en la raíz de la aplicación la misma que permite solicitudes CORS.

```
if (isset($_SERVER['HTTP_ORIGIN'])) {
    header("Access-Control-Allow-Origin: {$_SERVER['HTTP_ORIGIN']}");
    header('Access-Control-Allow-Credentials: true');
    header('Access-Control-Max-Age: 86400');
}
if ($_SERVER['REQUEST_METHOD'] == 'OPTIONS') {
if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_METHOD']))
header("Access-Control-Allow-Methods: GET, POST, PUT, DELETE, OPTIONS");
if (isset($_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']))
header("Access-Control-Allow-Headers:
{$_SERVER['HTTP_ACCESS_CONTROL_REQUEST_HEADERS']}");
}
$_REQUEST["notheme"] = "Y";
require("index.php");
```

Tabla 5: Servicios web

Fuente: Propia

Creación del controlador Phonegap.

```

1  <?php
2  class phonegap extends CI_Controller{
3  public function index(){
4
5  }
6
7  public function nuevopedido(){
8      $this->load->helper("url");
9      $this->load->model('clienteModel', 'listado_cliente');
10     $data['listado_cliente'] = $this->listado_cliente->Todos();
11     $this->load->view("phonegap/agregar",$data);
12 }
13
14 public function getcliente(){
15     $this->load->helper("url");
16     $this->load->model('clienteModel', 'listado_cliente');
17     $listado = $this->listado_cliente->Todos();
18     if(count($listado)==0){
19         echo "<option value=''>NO EXISTE CLIENTES</option>";
20     }
21     else{
22         echo "<option value=''>SELECCIONAR...</option>";
23         foreach ($listado as $fila){
24             $obj = (Object)$fila;
25             echo "<option value='".$obj->id_cliente.">$obj->razon_social</option>";
26         }
27     }
28 }
    
```

Gráfico 9: Phonegap.php móvil
Fuente: Propia

Nombre	Phonegap.php
Tipo	Clase heredada de CI_Controller
Métodos y Funciones	<p>getcliente: Obtiene la información de cliente</p> <p>getdatesystem: Obtiene la fecha del servidor</p> <p>guardarpedido: Guarda un pedido</p> <p>guardarcliente: Agrega un nuevo cliente</p> <p>validausuario: Valida la existencia de un usuario</p> <p>getproductos_json: Obtiene un listado de productos</p> <p>getclientes_json: Obtiene un listado de clientes</p>

	<p>getpagos_json: Obtiene los pagos efectuados</p> <p>getpedidos_json: Obtiene un listado de pedido</p>
Descripción	Se encarga de recibir las solicitudes de la aplicación móvil y enviar las respuestas.
Código Fuente:	
<pre> class phonegap extends CI_Controller{ public function index(){ } public function nuevopedido(){ \$this->load->helper("url"); \$this->load->model('clienteModel','listado_cliente'); \$data['listado_cliente'] = \$this->listado_cliente->Todos(); \$this->load->view("phonegap/agregar",\$data); } public function getcliente(){ \$this->load->helper("url"); \$this->load->model('clienteModel','listado_cliente'); \$listado = \$this->listado_cliente->Todos(); if(count(\$listado)==0){ echo "<option value="">NO EXISTE CLIENTES</option>"; } else{ echo "<option value="">SELECCIONAR...</option>"; foreach (\$listado as \$fila){ \$obj = (Object)\$fila; echo "<option value='{\$obj->id_cliente}'>\$obj->razon_social</option>"; } } } public function getproductos(){ \$this->load->helper("url"); \$this->load->model('productoModel','listado_producto'); \$listado = \$this->listado_producto->Todos(); if(count(\$listado)==0){ echo "<option value="">NO EXISTE CLIENTES</option>"; } else{ echo "<option value="">SELECCIONAR...</option>"; } } } </pre>	

```

        foreach ($listado as $fila){
            $obj = (Object)$fila;
            echo "<option value={$obj->id_producto}>{$obj->nombre}</option>";
        }}
    public function getdatesystem(){
        echo date("Y-m-d");
    }
    public function guardarpedido(){
        $entrada = $_REQUEST["pedidoremoto"];
        $localpedido = explode("<sep|sep>", $entrada);
        $encabezado = json_decode($localpedido[0]);
        $detalle_json = json_decode("[\".$localpedido[1].\"]");
        $this->load->model('pedidoModel','pedido');
        $detalle = array();
        foreach ($detalle_json as $fila){
            $localfila = array();
            $localfila["cantidad"] = $fila->cantidad;
            $localfila["precio"] = $fila->precio;
            $localfila["total"] = $fila->total;
            $localfila["id_producto"] = $fila->id_producto;
            $detalle[] = $localfila;
        }
        $stranret = $this->pedido->Insertar($encabezado->id_cliente,date("Y-m-d"),"GENERADO",$detalle);
        if($stranret==true){
            echo "OK";
        }
        else{
            echo "ERROR";
        }
    }
    public function guardarcliente()
    {
        $this->load->model('clienteModel','cliente');
        extract($_REQUEST);
        $stranret = $this->cliente-
>Insertar($razon_social,$ci_ruc,$nombre_contacto,$apellido_contacto,$direccion,$d
omicilio,$telefono,$celular,$ubicacion,$eliminado);
        if($stranret==true){
            echo "OK";
        }
        else{

```

```

        echo "ERROR";
    }
}

public function validausuario(){
    $this->load->model("usuarioModel","usuario");
    extract(XTUUtils::AllVars());
    $localfila = $this->usuario->validausuariomobile($cedula,$clave);
    if(count($localfila)!=0){
        if(count($localfila["privilegios"]==0){
            echo "SP";
        }
        else{
            echo "[". json_encode($localfila) ."]";
        }
    }
    else{
        echo "NO";
    }
}

public function getproductos_json(){
    $this->load->helper("url");
    $this->load->model('productoModel','listado_producto');
    $listado = $this->listado_producto->Todos();

    $tmp = array();
    foreach ($listado as $fila){
        $tmp[] = array_map('utf8_encode', $fila);
    }
    echo json_encode($tmp, JSON_PRETTY_PRINT);}

public function getclientes_json(){
    $this->load->helper("url");
    $this->load->model('clienteModel','listado_cliente');
    $listado = $this->listado_cliente->Todos();
    $tmp = array();
    foreach ($listado as $fila){
        $tmp[] = array_map('utf8_encode', $fila); }
    echo json_encode($tmp, JSON_PRETTY_PRINT);}

public function getpagos_json(){
    $this->load->helper("url");
    $this->load->model('pagofacturaventaModel','pagofacturaventa');
    $listado = $this->pagofacturaventa->Todos();
    $tmp = array();

```

```

foreach ($listado as $fila){
    $tmp[] = array_map('utf8_encode', $fila);
}
echo json_encode($tmp, JSON_PRETTY_PRINT);
}
public function getpedidos_json()
{
    $this->load->model('pedidoModel','pedido');
    extract(XTUtils::AllVars());
    $listado = $this->pedido->Todos($this->pagina, $this->tam_paginacion,$xtfiltro);
    $tmp = array();
    foreach ($listado as $fila){
        $tmp[] = array_map('utf8_encode', $fila);
    }
    echo json_encode($tmp, JSON_PRETTY_PRINT);
}
}

```

Tabla 6: Desarrollo en Phonegap
Fuente: Propia