

Proyecto – Community

Sergio Mora Martínez

M12 – Informe Proyecto

Índice

Introducción.....	3
Rutas.	3
Descripción.	3
Página Principal	3
Cpanel	3
BBDD.....	5
Plugins y librerías.	6
SCSS.....	6
DataTables.....	7
Alertify	8
DomPDF.....	8
Ficheros Clave	9
Evolución del código.....	11
Funcionalidades	13

Introducción

El proyecto se basa en la creación de un entorno gráfico en la que proveedores y vendedores se puedan encontrar con mayor facilidad y comunicar entre ellos. Por lo que un proveedor puede buscar vendedores que distribuyan sus productos y un vendedor puede buscar los productos que vende al mejor precio.

Rutas

Web: <http://146.255.101.232/p2daw6/>

Github: <https://github.com/kokoke/Community>

Descripción

Página principal

La página principal sirve para las funciones básicas, atraer nuevos usuarios, un contactar con su formulario y una pequeña API de google con la localización de la sede principal. Un formulario de registro con los detalles de usuario, datos personales y datos de la empresa en el que se incluye si es un proveedor o vendedor. Se incluye un formulario de login, para quien no conozca el login principal.

Cpanel

Se dispone de dos tipos distinto de usuarios.

Proveedores

Dispone de las siguientes secciones propias:

- **Productos:** En la que podrá crear, editar y modificar sus propios productos. También incorpora la sección categorías, en la que podrá ver las categorías insertadas por el administrador, que no podrá eliminar, y las que el mismo se cree. Una vez creada una categoría propia, este la podrán eliminar.

- **Ventas:** Se podrán ver todos los pedidos que se han realizado. Un listado general de todos las ventas, que clicando sobre dicha venta, se podrá ver un listado detallado de quien le ha realizado el pedido y que productos y cantidad le han solicitado.

Podrán exportar un XML, de la Venta. Se elije **XML** a causa de que es **openSource**, el fichero que se descarga es texto plano y además es la representación básica del XLS,.

Vendedores

Dispone de las siguientes secciones propias:

- **Pos:** Es un terminal de punto de venta, en el que podrán comprar todos aquellos productos que dispongan sus contactos (sección de Preferencias) y ver un listado detallado del producto que compra y a quien se lo compra.

- **Compras:** Sección en la que podrán ver todas las compras realizadas, y crear un **PDF** de la factura, detallando las cantidades de los productos comprados, ha quien se los ha comprado y el importe final de dicha factura.

Comunes

Los dos usuarios dispondrán de secciones comunes:

- **Mensajes:** En esta sección se podrá hablar con todos aquellos usuarios que en contacto (Preferencias) se hayan aceptado y con el administrador, que ya esta agregado por defecto.

- **Preferencias:** Sección en la que podrán modificar todos los datos del usuario, como la imagen utilizada como avatar. También podrán buscar a otros proveedores, vendedores y productos que se hayan registrado con sus diferentes precios.

Si encuentran a alguien interesante, pueden mandar la solicitud para que lo acepten, una vez enviada, al que le han solicitado, podrá ver en la sección de "contactos pendientes" aquellos usuarios que le han enviado la solicitud. Si la aceptar, en la sección de contactos aparecerán ambos. Si el que manda la petición es un vendedor, una vez aceptada la petición, este podrá ver todos los productos que vende el proveedor.

- **Contactar:** Le envía a la página principal de contactar.

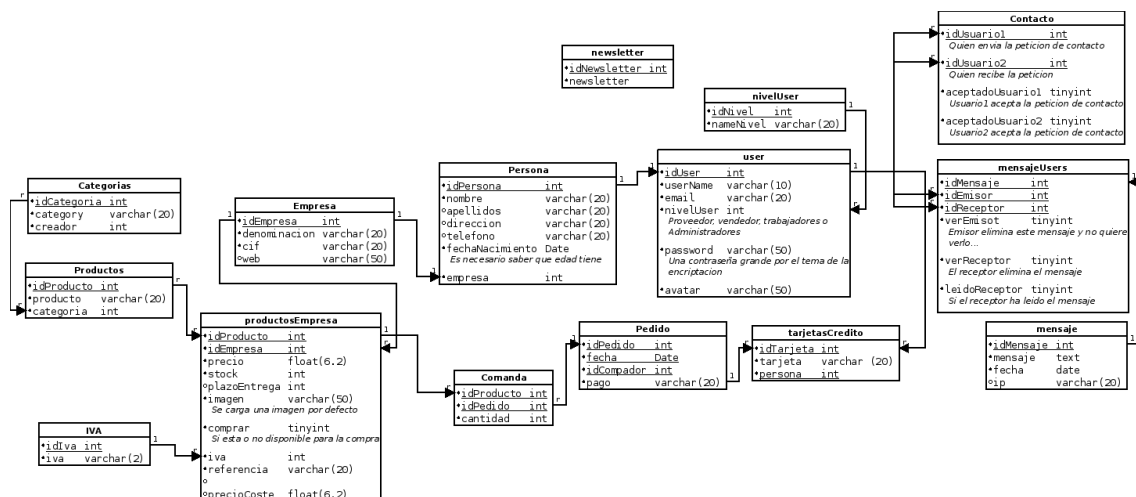
BBDD

Los diferentes usuarios que se registran disponen de un nivel distinto, dependiendo del nivel que disponga se les permitirá entrar en unas secciones u otra. Se utiliza la encriptación **"sha1"**, para que las contraseñas queden almacenadas de una forma segura. Se almacenan los datos personales y empresariales.

Los productos nuevos son independientes de la empresa, en el momento que alguien lo crea, este se guarda y cualquier otra persona que utilice el mismo producto comparte la "id" de dicho producto. En la tabla "productosEmpresa" permite configurar al gusto del empresario dicho producto, puede configurar el stock que dispone, el precio, IVA, imagen propia... Cuando se crea un producto se puede elegir la categoría, tanto las que hay por defecto como las creadas por el propio creador.

En “contacto” se almacenan las peticiones entre usuarios para que puedan interactuar entre ellos. Hasta que los dos no han aceptado la petición, no pueden hablar entre ellos ni comprarse producto.

Diagrama



Plugins/Librerías

SCSS

Es una extensión de CSS3, que te permite la creación de **variables**, **mixins** y la creación de una **estructura de herencia** más organizada. Con la creación de variables... me permite definir colores, alturas, fuentes...

SCSS	CSS
<pre> \$colordeFondo: #3A3R21; \$colorTextoHeader: #431232; \$fuenteTituloCabecera: 'Arial'; \$anchoElemento: "150px"; body { background: \$colordeFondo; header { font-family: \$fuenteTituloCabecera; img { width: \$anchoElemento; } div { width: \$anchoElemento; color: \$colorTextoHeader; } } } #contenedor { background: \$colordeFondo; }</pre>	<pre> body { background: #3A3R21; } body header { font-family: 'Arial'; } body header img { width: "150px"; } body header div { width: "150px"; color: #431232; } #contenedor { background: #3A3R21; }</pre>

Mientras que en SCSS, la estructura jerárquica es más clara y precisa, en CSS tienes que estar repitiendo elementos para llegar al mismo fin y puede ser más confuso.

Llegado un momento en el que prefieres el cambiar "#3A3R21" por "#3A3R22" en SCSS únicamente debes cambiar el valor de la variable mientras que en CSS, tienes que ir elemento por elemento cambiando el valor.

Mixins	CSS
<pre> @mixin prefixr(\$que, \$arguments) { #{\$que}: \$arguments; -webkit-#{\$que}: \$arguments; -ms-#{\$que}: \$arguments; -moz-#{\$que}: \$arguments; -o-#{\$que}: \$arguments; } @include prefixr(box-shadow, 0 1px 2px 0 #cd8787 inset); </pre>	<pre> body { -webkit-box-shadow: 0 1px 2px 0 #cd8787 inset; -moz-box-shadow: 0 1px 2px 0 #cd8787 inset;; } #contenedor { -webkit-box-shadow: 0 1px 2px 0 #cd8787 inset; -moz-box-shadow: 0 1px 2px 0 #cd8787 inset;; } </pre>

*código proporcionado por deCampus.es

A través de SCSS se puede crear un mixin (funciones) que te permita tener configuraciones. Por ejemplo, este ejemplo te permite adaptar tu código llamando únicamente a este mixin a cualquier navegador, mientras que en CSS tienes que ir incorporándolo en cada tag la adaptación.

DataTables

Se ha utilizado este plugin principalmente para el listado de elementos como facturas, compras... Las principales funcionalidades por la que se eligió este :

- **Buscador:** Te permite buscar cualquier contenido (texto) en cualquiera de las columnas y filas.
- **Ordenar:** Te permite ordenar cualquier columna ascendente o descendientemente, haciendo que si quieres ordenar las facturas de más caras a menos , se una tarea fácil.
- **Mostrar:** Se pueden mostrar tantos elementos como se desee. Por defecto lo máximo que se muestra es 100 elementos por paginación, pudiéndose modificar a través de código.
- **Paginación:** Para evitar que se muestren tantos elementos que sea imposible encontrar algo, este plugin dispone de una paginación evitando que la tabla se haga enorme, únicamente los que le indicas que se muestren.

Alertify

Plugin utilizado para las notificaciones u advertencias que van ocurriendo como por ejemplo que introduce un email ya existente. Las diferentes funcionalidades que se han utilizado para este plugin son :

```
alertify.success( "Inscripción correcta" );
```

Para notificar sucesos que han ocurrido satisfactoriamente.

```
alertify.error( "Usuario ya inscrito" );
```

Para notificar sucesos que se han terminado de ejecutar por algún error.

```
alertify.confirm("Eliminar?", function (e) {  
    if (e) {  
        alertify.success( "Eliminado." );  
    } else {  
        alertify.error( "No se ha eliminado" );  
    }  
});
```

Para solicitar que acepte antes de realizar cualquier acción.

DomPDF

Librería utilizada para generar PDF de las facturas. A diferencia de otras librerías, esta te permite generar css para darle estilo a tu plantilla de PDF.

Ficheros clave

La mayor parte de los ficheros clave de mi proyecto se encuentran en el directorio "core". Desde un principio, index.php llama al fichero principal de mi proyecto "core/core.php":

Core

Este fichero es el encargado de trabajar con la URL, de aquí sacamos ha que controlador tenemos que llamar y que acción efectuar. Su trabajo consiste en fragmentar la URL, una vez fragmentada, la envía a "**Route**", que se encargara de verificar si la ruta es correcta. Una vez le haya retornado un valor dependiendo del valor devuelto, llamara al controlador adecuado y a su acción o lo enviar al controlador de "**error 404**".

Controller

Es el encargado de cargar el controlador que le envía core.

Database

Es la clase que se encarga de conectar con la base de datos, y el encargado de realizar las consultas de inserción u obtención de datos. Se utiliza el PDO para poder trabajar en diferentes BBDD como por ejemplo "**Matisse**".

Define

Este fichero es donde se almacenan el diccionario, en mi caso se utiliza para definir todas las rutas.

Helpers

Este fichero almacena unas pequeñas funciones que te permiten mostrar el contenido de las variables, las cuales te pueden ayudar a saber si los valores que contiene el objeto son los adecuados. `pd($array)`, te mostraría en pantalla el contenido de la variable "\$array" el cual puedes saber si contiene lo que debe contener.

/config

Es el encargado de configurar los valores por defecto, en mi caso almaceno la array de los idiomas disponibles y la url principal del dominio 'http://146.255.101.232/p2daw6/'.

Lang

Esta clase se encarga de recoger y cargar el diccionario de los idiomas por: `setLanguage`, es el encargado de cargar en una sesión el idioma seleccionado, `Lang::p("Language:")`, es quien a través de la sesión creada anteriormente recoge la traducción de una `array` (`"Language:" => "Idioma"`);

Redirect

Clase que proporciona las funciones de redicciones; `to()`, se utiliza para redireccionar a una página en concreto; `to_prev()`, para redireccionar a la página anterior; `to_route()`, se utiliza para redirecciona a través del `Route` especificando el controlador, la acción y argumento (productos).

Route

Clase que se encarga de recoger el controlador y acción que le envía `Core`, y trabajar con esto. Sigue el mismo proceso que el `Redirect`, con la diferencia que esta clase no redirecciona, únicamente obtiene la ruta del controlador. Se utiliza principalmente para los enlaces.

Ej: `" Preferencias "`

Application/routes.php

Diccionario que almacena todas las rutas disponibles, si no se encuentra la ruta en este diccionario lo envía al **error 404**. Por lo contrario, si encuentra la ruta, llama al controlador y dicha acción asignada a esta ruta.

Evolución del código

A lo largo de la creación del proyecto, se ha ido evolucionando el código y se añade un elemento característico que lo diferencia de las otras generaciones.

Primera generación

- + HTML
- + CSS
- + PHP

Creación básica del proyecto con estructura base de HTML y con elementos de PHP. Los tags, no se les incorpora las identificaciones, únicamente un "name" para enviar formularios.

Segunda generación

- + HTML
- + PHP
- + CSS
- + JavaScript

Se comienza a trabajar con objetos y a disponer de una jerarquía de ficheros.

Tercera generación

- + HTML
- + PHP
- + CSS
- + JavaScript
- + MVC (prototipo)

En esta generación se caracteriza por la primera versión creada del MVC (anterior a la agencia).

Cuarta generación

- + HTML
- + PHP
- + CSS
- + JQuery
- + MVC

Ya se dispone un MVC funcional y con su estructura ya definida, a demás, se comienza a trabajar con JQuery.

Quinta generación

- + HTML
- + SCSS
- + PHP
- + MVC
- + JQuery
- + Plugins JQuery
- + Librerías.

Ultima generación en la que se implementa el SCSS para trabajar con estilos y la utilización de plugins para facilitar el trabajo. Las variables y funciones pasan de llamarse "botón_guardar" a "inputFormularioActualizarDatosUsuario"

Funcionalidades

Email

En la sección de contactar, en el momento de enviar un mensaje, se envía dos correos electrónicos, uno que va directamente al cliente indicándole que hemos recibido el mensaje y que nos pondremos en contacto con ellos y otro el administrador con el mensaje en cuestión, más los datos para contactar con el solicitante.

Traducciones

Implementado en el login del cpanel y su página principal, esta la opción de cambiar de idioma. El funcionamiento es fácil, se envía a:

```
<a href="/p2daw6/lang/index?lang=es_es" class="aColorBlack">Es</a>
```

Donde interviene la clase **Lang** y recoge los valores de las traducción.

Controles

A lo largo de la creación del proyecto, se han ido implementando controles para su correcto funcionamiento. Algunos como que no se repitan cuentas de correo electrónicos al registrarse o que no entre en secciones en las que hace falta hacer login. Algunos controles son:

- **Login:** Comprueba que el usuario introducido existe en nuestra BBDD y si ha introducido valores antes de enviar la petición.
- **Newsletter:** Comprueba que el email existe antes de insertarlo.
- **Registro:** Comprueba que todos los datos están introducidos, que el usuario no exista (email) y la contraseña cumpla con unos requisitos de seguridad (Mayusculas, Minusculas y numeros).
- **Cpanel:** No puede acceder a ninguna ventana perteneciente del Cpanel si no ha iniciado sesión, si lo intenta, automáticamente es redireccionado al login del Cpanel.

Los Proveedores, aparte de las secciones comunes, solo pueden entrar en Catalogo (donde crean los productos) y Pedidos (para ver los pedidos), mientras que, los Vendedores únicamente pueden entrar en Pos (terminal punto de venta) y Compras (para ver las compras realizadas). Si alguno de estos usuarios intenta entrar en una de las secciones que no le pertenece, este automáticamente es enviado a la página similar Catalogo -> Pos y Compras -> Ventas.

Cualquier elemento que se intente eliminar o modificar, antes es comprobado. Para poner un ejemplo, si el Proveedor intenta eliminar un pedido que no es suyo (modificando el código), esto se comprueba y es automáticamente redireccionado a la página en la que se situaba.

RSS

Se ha creado un mini RSS, con los clientes que se han ido registrando, para que se puedan ver todos aquellos a los que proporcionamos servicios y atraiga más usuarios interesados.

Social

Se han creado cuentas en redes sociales como Facebook, Twitter... para fomentar la distribución del "Proyecto" a través de dichas redes sociales.