

TELEPHONOS.ME

[Autores: Juan Huesa, Daniel Frías y Pablo Tarín](#)

[Tutor: José Manuel Prieto](#)

ÍNDICE

1. Objetivo	2
2. Evaluación de costes	2
3. Herramientas utilizadas.....	3
4. Configuración servidor web	4
5. Base de datos	7
6. Pagina web	10
6.1 Registro	10
6.2 Inicio y cierre de sesión	11
6.3 Productos en la web.....	12
6.4 Diseño web.....	14
7. Configuración servidor Raspberry Pi.....	18
8. Servidor de dominio	20
9. Configuración redes	22
10. Redes sociales y comunicación	23

1. Objetivo

El objetivo del proyecto es crear una tienda online especializada en la venta de teléfonos móviles con una red de tiendas físicas. La empresa constará de un servidor virtual privado (VPS) donde se alojará la base de datos y la página web, un servidor Raspberry Pi que actuará como DHCP, FTP y servidor de correo. La empresa también contará con un servidor de dominio y DNS para gestionar los usuarios de los PCs. La página web constará de un registro de usuarios para poder realizar compras en la misma.

Nuestro objetivo es conseguir una web rápida, de diseño moderno, desde la cual el usuario pueda acceder fácilmente y comprar de forma segura terminales móviles y accesorios electrónicos. Para poder realizar las compras los usuarios deberán de registrarse mediante un sencillo formulario y añadir al carrito los productos que deseen comprar. Para asegurar la seguridad de los datos del comprador se ha implementado el protocolo seguro https que provee de un cifrado de los datos del usuario.

2. Evaluación de costes

Durante este proyecto se ha adquirido los siguientes artículos para la realización del mismo:

- Alquiler VPS adquirido en digitalocean.com durante 6 meses 30€
- Compra de dominio telephonos.me durante 1 año en namecheap.com 3€
- Compra de certificado SSL en namecheap.com 9€

3. Herramientas utilizadas

- **XAMPP | LAMP:** Herramienta muy útil y versátil para la creación de la página web. Viene con un Servidor Apache, donde está la página web y un servidor de MYSQL donde se guardan las bases de datos. Esta versión viene con PHP 5.5, lenguaje con el que transmitiremos la información de la base de datos a la web y viceversa. XAMPP es la versión que hemos utilizado para trabajar con la web desde Windows, y LAMP es la versión para Linux, donde residen nuestros servidores.
- **Sublime Text 2:** Editor de texto muy potente con el que se puede escribir en varios lenguajes. Autocompletado de funciones, buena interfaz grafica y sencilla visualización del contenido.
- **Putty:** Un cliente de telnet muy sencillo y útil. Para conectarse en remoto al servidor de forma segura y configurarlo.
- **Filezilla:** Con este cliente SFTP subimos los cambios en el servidor. Herramienta de buen rendimiento para manejar los archivos.
- **Virtual Box:** Gestor de maquinas virtuales con el que simularemos la red privada de la empresa y el dominio.
- **Packet tracer:** Simulador de redes en el que configuraremos tanto la red LAN de la empresa como la red WAN y su acceso al servidor web.

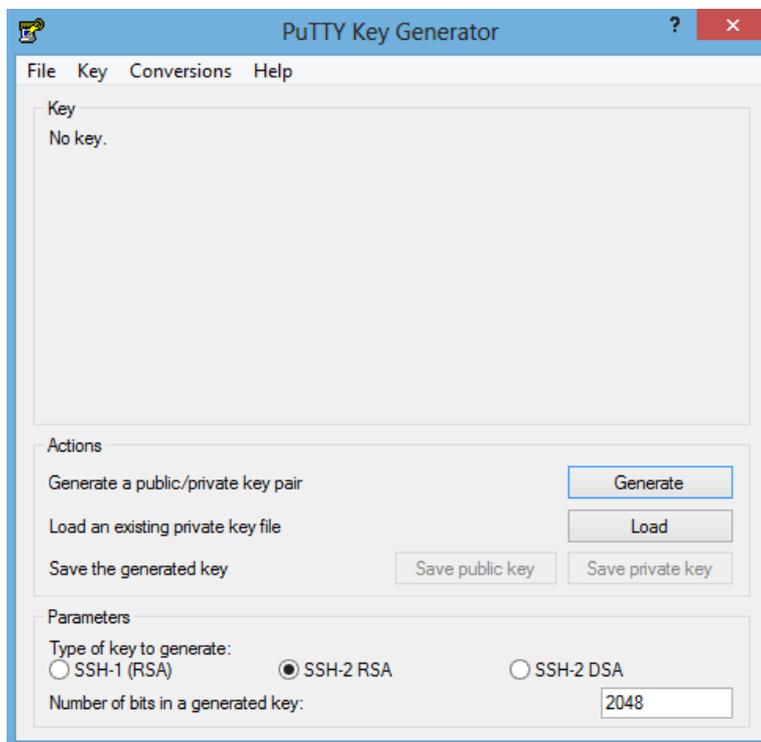
Sistemas operativos utilizados:

- Ubuntu server 14.04 LTS, donde se aloja el servidor web y base de datos de nuestra empresa.
- Raspbian, usada en la Raspberry que proporcionara varios servicios de red en las tiendas y oficinas.
- Windows Server 2012 R2, que tiene las funciones de controlador de DOMINIO y DNS.
- Novedades: Uso de Windows 10 como simulador de una sucursal.

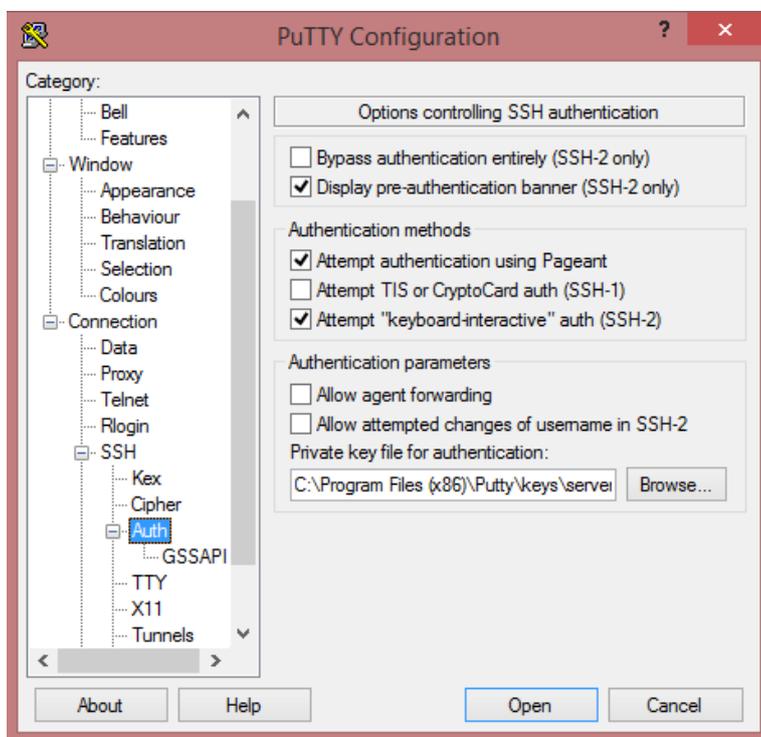
4. Configuración servidor web

Para la instalación del servidor web se ha utilizado un servidor alojado en Digitalocean.com en el cual se ha instalado un servidor Ubuntu server 14.04LTS. En él se ha instalado un LAMP (Linux, Apache, MySQL y PHP).

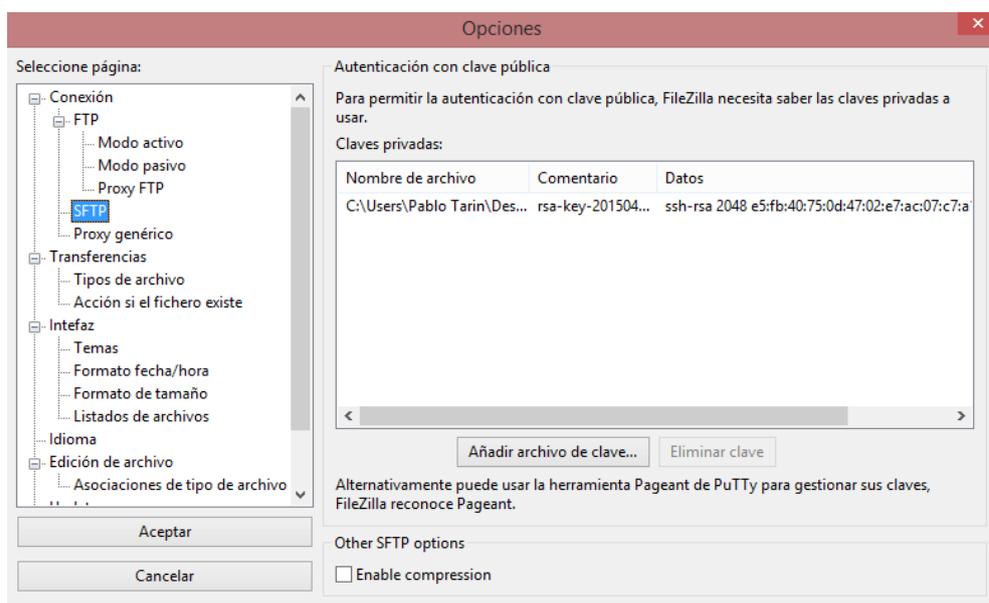
Para acceder al servidor de forma segura y lograr que nadie pueda acceder al mismo con privilegios root, se ha implementado un sistema de llaves ssh de manera que sólo se pueden lograr credenciales utilizando esta llave. Para lograrlo se ha creado una llave privada y una llave pública, la llave privada utiliza una encriptación RSA de 2048 bits que será utilizada por los administradores del servidor y una clave pública que ha de ser alojada en el hosting, en este caso Digitalocean.com.



Para realizar los cambios en la configuración se ha utilizado el protocolo SSH, el primer cambio realizado ha sido el de cambiar la contraseña por defecto del servidor mysql y posteriormente se han introducido los comandos SQL para añadir la base de datos y las tablas correspondientes a la misma.



Para añadir los ficheros de la web (htmls, phps, imágenes, etc) se ha utilizado el protocolo seguro SFTP utilizando la misma llave utilizada para acceder mediante SSH.



Comparamos un dominio e introducimos la dirección IP del servidor al que queremos que apunten las DNS para poder acceder fácilmente a nuestro servidor y poder asignarle posteriormente el certificado SSL

Por último se ha añadido el certificado SSL, para ello primero es necesario obtener un fichero .csr y enviarlo a la entidad que nos va a certificar. Para obtenerlo hemos instalado openssl y utilizado el comando:

```
openssl req -new -newkey rsa:2048 -nodes -keyout telefonos.me.key -out telefonos.csr
```

Una vez recibido el certificado se ha copiado en el servidor y modificado la configuración de apache y utilizando el comando a2ensite para habilitar el acceso https

```
# SSL Engine Switch:
# Enable/Disable SSL for this virtual host.
SSLEngine on

# A self-signed (snakeoil) certificate can be created by installing
# the ssl-cert package. See
# /usr/share/doc/apache2/README.Debian.gz for more info.
# If both key and certificate are stored in the same file, only the
# SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/ssl/telefonos_me.crt
SSLCertificateKeyFile  /etc/ssl/telefonos.me.key
```

```
a2ensite default-ssl
```

<https://www.telefonos.me>

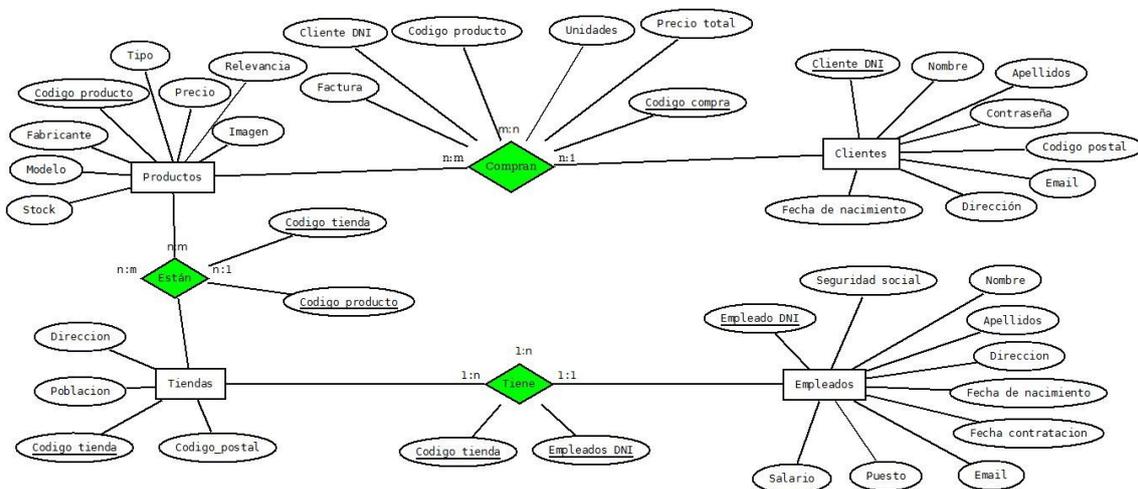


5. Base de datos

TelePhonos dispone de una base de datos MYSQL, donde se guardará la información referente a los productos, clientes, empleados y operaciones realizadas de la empresa.

La base de datos está alojada junto al servidor web. Los administradores pueden conectarse de forma remota para organizar y gestionar la base de datos. Los administradores pueden acceder a la base de datos y al servidor web de forma segura desde las diferentes tiendas y oficinas de la empresa.

Modelo entidad/relación:



Modelo relacional:

Entidades:

Empleados(Empleado DNI, nombre, apellidos, dirección, seguridad social, fecha nacimiento, fecha contratación, email, puesto, salario)

Tiendas(código tienda, dirección, población, código postal)

Productos(código producto, precio, tipo, fabricante, modelo, stock, relevancia, imagen)

Clientes(cliente DNI, nombre, apellidos, dirección, email, contraseña, fecha de nacimiento, código postal)

Relaciones:

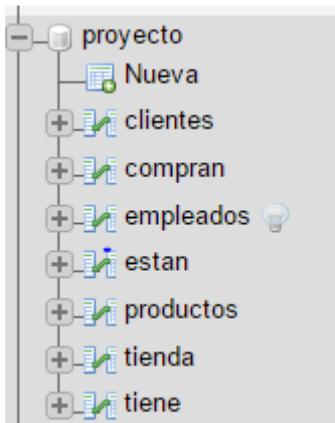
Tiene(código tienda, empleado DNI)

Están(código tienda, código producto)

Compran(cliente DNI, código compra, código producto, factura, unidades, Precio total)

Como nuestra base de datos va conectada a una página web, desde la que se realizan consultas e inserciones, todas nuestras tablas usan el motor InnoDB, pues lo consideramos el más apropiado para esta tarea.

Tablas de la base de datos:



Empleados: Almacena la información de los empleados de la empresa.

```
create table empleados (
  empleado_dni varchar(9) primary key,
  nombre varchar(40) not null,
  apellidos varchar(60) not null,
  direccion varchar(50) not null,
  seguridad_social varchar(13) not null,
  fecha_nacimiento date not null,
  fecha_contratacion date not null,
  email varchar(30) not null,
  puesto varchar(40) not null,
  salario int not null);
```

Tiendas: Almacena la ubicación de las tiendas de la empresa.

```
create table tienda(
  codigo_tienda varchar(40) primary key,
  poblacion varchar(30) not null,
  direccion varchar(30) not null,
  codigo_postal int not null);
```

Productos: Almacena todos los datos de los productos, para su venta y registro.

```
create table productos(
  codigo_producto int primary key auto_increment,
  precio int not null,
  fabricante varchar(40) not null,
  modelo varchar(40) not null,
  stock int not null,
  relevancia int,
  tipo varchar(40),
  imagen varchar(40) not null);
```

Cientes: Guarda la información de registro de los clientes.

```
create table clientes(  
cliente_dni varchar(9) primary key,  
nombre varchar (40) not null,  
apellidos varchar (60) not null,  
email varchar(30) not null,  
fecha_nacimiento date not null,  
password varchar(30) not null,  
direccion varchar(40) not null,  
codigo_postal int not null);
```

Tiene y Están. Tablas que relacionan las mencionadas anteriormente.

```
create table tiene(  
codigo_tienda varchar(40) references tienda,  
empleado_dni varchar(9) references empleados,  
primary key ( codigo_tienda,empleado_dni));
```

```
create table estan(  
codigo_tienda varchar(40) references tienda,  
codigo_producto varchar(40) references productos,  
primary key (codigo_tienda, codigo_producto));
```

Compran: Relaciona productos con los clientes. Es la tabla donde se guardan las operaciones realizadas por los clientes.

```
create table compran(  
factura varchar(30) not null,  
cliente_dni varchar(9) not null references clientes,  
codigo_producto varchar(40) not null references productos,  
unidades int not null,  
precio_total int not null,  
codigo_compra varchar(30) primary key);
```

La base de datos esta automatizada mediante unos triggers, para tramitar las compras y acciones automatizadas; y procesos, para tareas de mantenimiento y crear copias de seguridad.

Por ejemplo:

Un trigger que guarda todas las acciones que se ejecutan sobre la tabla productos y compran en otra tabla del sistema. De esta forma, se crea una nueva tabla en forma de registro de las operaciones realizadas en la base de datos.

Un proceso que crea copia de seguridad de la base de datos y que muestra el estado/información de la base de datos al administrador.

La base de datos está conectada con la web mediante código PHP. Los usuarios registrados en la web y guardados en la base de datos pueden comprar los productos de la empresa a través de un carrito de la compra que conecta con la base de datos.

6. Página web

En la página web se puede comprar los productos. Solo los usuarios registrados y conectados pueden comprar productos.

Para registrarse en la web hay que rellenar un formulario, que esta revisado con código Javascript.

6.1 Registro

Javascript sirve para poner una serie de requisitos a la hora de enviar el formulario. Sirve como sistema de seguridad para la veracidad de los datos que ingresan los clientes en nuestra plataforma.

Formulario para registrar nuevos usuarios:

Nombre

Apellidos

Contraseña

Repita la contraseña

DNI

Email

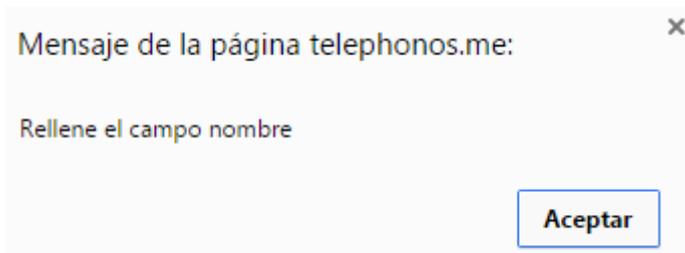
Fecha de nacimiento

Dirección

Código Postal

La validación de los campos son parecidas, pero algún campo tiene reglas adicionales para asegurarnos de que los datos sean correctos.

En el caso de que algún campo no cumpla los requisitos, la página mostrará una advertencia diciendo cual es el campo que esta erróneo. Así los clientes pueden rellenar de forma correcta su registro en nuestra web.



Pequeño extracto del código Javascript.

Ejemplo de cómo funciona la validez del campo DNI

```
// Verificación de si el dni esta vacío y si es correcto
if (dni==null || dni.length<9) {
    alert('Rellene el campo dni');
    return false;}
var letra=dni.charAt(8);
var letra=letra.toUpperCase();
var numero=dni.substring(0,8);
var numero=parseInt(numero);
var resto=numero%23;
var cadena='TRWAGMYFPDXBNJZSQVHLCKE';
var resultado=cadena.charAt(resto);
if (letra!=resultado){
    alert('El dni no es correcto');
    return false;}
```

El proceso de registro consta de una segunda parte, la validación de los datos con PHP. Una vez validado con el Javascript, el formulario se envía a datosregistro.php donde las variables recogidas en el formulario se procesan.

Entonces se comprueba si los campos DNI ó email están en uso. De ser así el registro no se realizará y mostrara una advertencia al usuario.

```
if ( $numfilas == 0 && $numfilass == 0 ){
    // Insertar en la tabla
    $consulta="insert into $tabla (cliente_dni, nombre, apellidos,
mysql_query($consulta);
    exit();}

    print("Error, ya existe ese correo electronico d dni");
```

En el caso de que se hayan cumplido las dos partes, el usuario ya estará registrado en la web y tras la activación de la cuenta podrá usarla.

6.2 Inicio y cierre de sesión

Una vez tengamos una cuenta disponible en la web, podremos iniciar sesión.

Mediante php iniciamos nuestra sesión en la página. Se nos abre un formulario desplegable para iniciar la sesión. Solo son necesarios los campos de correo y contraseña para acceder a la web.

[Regístrate](#) | [Iniciar sesión](#)
 Correo electrónico:
 Contraseña:

Si los datos son correctos se podrá acceder a la cuenta. Si no se encuentra esa cuenta asociada a esa contraseña nos aparecerá un error y no se completará el inicio de la sesión.

```
// Si no existe el usuario vuelve a la página principal
if ( !$numfilas) {
  print("No existe el usuario o la contraseña es errónea<br>");
  print("<a id=link href='iniciarsesion.html'>Clic para volver si tu navegador no te redirecciona automáticamente</a>");
  exit();}

```

Una vez iniciada la sesión desaparecen las opciones anteriores por las siguientes:

Hola Juan
[Cerrar sesión](#)
[Ver carrito](#)

Para mejorar la efectividad al cerrar la sesión nos aseguramos de destruir todas las variables de sesión

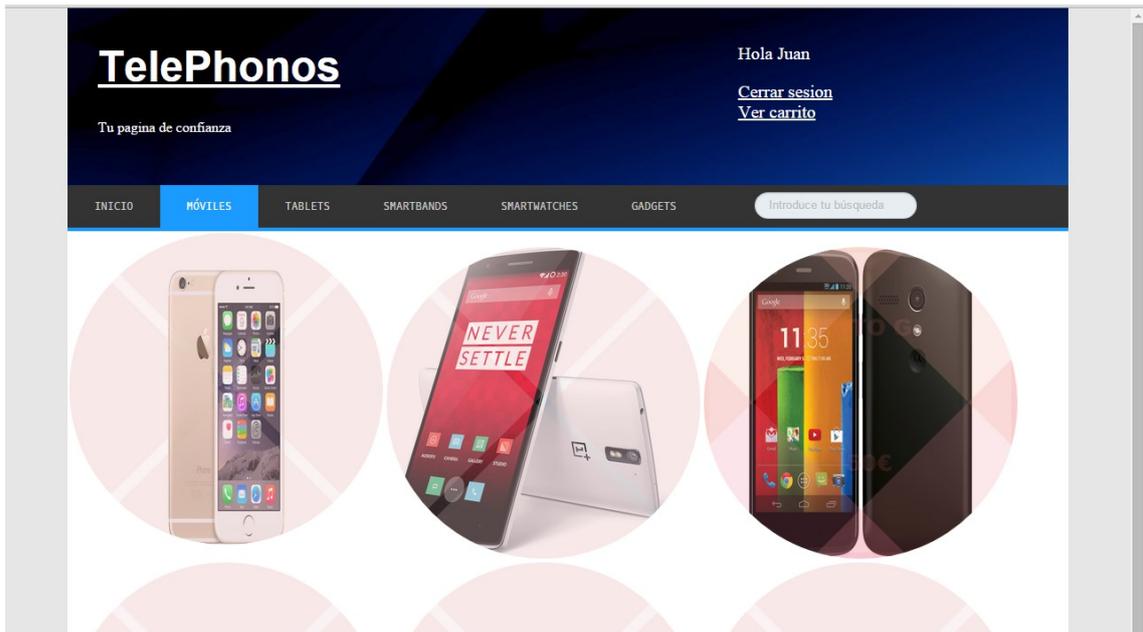
```
$_SESSION = array();
session_destroy();
```

Tanto al iniciar sesión como al cerrarla, hay un script de Javascript que redirige la página a index.php pasados unos segundos. Todo sea por la comodidad del usuario.

```
function redireccionar(){
  window.location="http://www.telefonos.me";}
setTimeout ("redireccionar()", 5000); //tiempo expresado en milisegundos
```

6.3 Productos en la web

Con el menú de la página web distribuimos los productos a mostrar según su tipo en varias páginas mediante código PHP. Cada página muestra una selección diferente.



Cada producto es un enlace que nos lleva a la pagina del producto, donde veremos más información, y podremos añadirlo al carrito o, si ya está en el, borrarlo.

¡Artículo añadido al carrito!

Este artículo ya esta en tu compra

[Borrar de la compra](#)

En el carrito de la compra podemos ver los productos que vamos a comprar. Podemos modificar las unidades, añadir o borrar más productos y ver la información propia de una factura. Al final debemos de confirmar nuestro carrito para que proceda a la compra.

The screenshot shows the 'TelePhonos' website. The header features the logo 'TelePhonos' on the left, 'Hola Juan' on the right, and links for 'Cerrar sesion' and 'Ver carrito'. Below the header is a navigation bar with categories: INICIO, MÓVILES, TABLETS, SMARTBANDS, SMARTWATCHES, and GADGETS. A search bar is located on the right of the navigation bar. The main content area displays a table of products:

	Modelo	Precio	Unidades.		Precio total	
	One Plus One	300€	1	1	300	<input type="button" value="Borrar"/>
	Iphone 6	800€	1	3	800	<input type="button" value="Borrar"/>
	Moto G	150€	1	2	150	<input type="button" value="Borrar"/>

Y mediante el recuadro de búsqueda podemos buscar una palabra clave en nuestra tabla de productos. Si la palabra coincide con algo en la base de datos mostrara el resultado. De lo contrario avisara de que no se han encontrado el producto.

6.4 Diseño web

El diseño de la web está realizado mediante PHP, HTML y CSS para darle formato. También en algún caso utilizamos librerías de jQuery ya que es software libre y de código abierto, por lo que algunas las hemos modificado para que se ajustaran a nuestra necesidad.

Todas las páginas PHP y HTML hacen referencia al menos a un archivo CSS en el que se da forma tanto al texto como a las imágenes que hay.

Desde la página inicial de la web (index.php) se hace una llamada a index.css, para recoger el estilo de toda la página. Al ser esta página en la que se encuentra el slider, también hay que tener en cuenta que los estilos para este son diferentes al resto.

Todas las páginas siguen el mismo formato, en la parte superior una cabecera en la que está incluido el nombre de la web en la parte izquierda, y en la parte derecha de la misma el login para iniciar sesión o el enlace al formulario de registro para nuevos usuarios. Para iniciar sesión se despliega en la misma parte de la web 2 campos de texto

en los que el usuario debe introducir su correo electrónico y contraseña para poder loguearse en la web. Esto se consigue haciendo una llamada al archivo desplegar.js

Debajo de esta cabecera hay una lista con todos los tipos de artículos que tenemos a la venta en la web (móviles, tablets, smartbands, smartwatches y gadgets) accesible mediante un menú desplegable incluido en todas las páginas de la web. Con la etiqueta *cssmenu* se hace referencia a cada campo del menú desplegable en el archivo *index.css* para que cambie de color la categoría en la que hemos pinchado y se ponga en un azul diferente al del resto.

En esta lista también tenemos en la parte derecha el buscador cuyo CSS también se encuentra en *index.css* y cuyo funcionamiento es básico para ayudar al usuario a que la web le sea más intuitiva y fácil de utilizar. Este buscador que hemos realizado es muy cómodo para su utilización, ya que al pinchar sobre él para realizar una búsqueda se agranda su tamaño por si el nombre del artículo es demasiado grande.



Posteriormente, en la página inicial tenemos el slider con los artículos más destacados de la web. Para el slider empleamos dos nuevas páginas CSS (*flexslider.css* y *style.css*) de las que se nutre el slider para ir pasando por cada imagen cuando el usuario lo decida o cada tiempo determinado (en nuestro caso cada 6 segundos, tiempo que hemos pensado es más que suficiente para que el usuario decida si le interesa esa oferta o no, y para que pueda ver todas las novedades del slider sin eternizarse en caso de que decida el movimiento automático).

También el slider redirige a la página de cada producto, tiene botón de avance y retroceso y debajo tiene unas bolas para que el usuario sepa el número de imágenes que contiene el slider y cuál de ellas está viendo en ese momento. También se nutre de dos librerías de jQuery, una de ellas mediante referencia a un enlace externo y otra a un enlace alojado en nuestro servidor, para demostrar que se puede realizar de ambas formas.

Posteriormente en cada una de las páginas que hemos hecho para cada producto además de tener la misma cabecera que el *index.php*, los productos tienen el mismo diseño realizado mediante pseudoelementos de CSS para conseguir que al colocar el puntero del ratón sobre cada producto una especie de cortinilla se despliegue y aparezca el nombre y precio del producto. Hemos utilizado este tipo de diseño porque pensamos que es muy visual de cara a los usuarios, a los que primero entra la imagen del producto y si les parece atractiva con solo colocar el puntero encima de la imagen conocen el

nombre del producto y el precio. Obviamente, esta cortinilla no es en ningún caso impedimento para que el producto se pueda observar sin necesidad de que se despliegue la cortinilla.



Estado inicial



Movimiento intermedio



Estado final

En cada página en la que tenemos productos con este formato, hemos añadido un enlace al archivo *seleccion.css* en el que se recoge el diseño que hemos decidido dar al octógono que rodea cada producto, tanto con sus colores como con sus movimientos explicados anteriormente.

La página de cada producto se divide en dos partes, a la izquierda la imagen, nombre y precio del producto, y a la derecha la descripción y características. También en esta página se puede añadir el producto al carrito para su posterior compra, únicamente en caso de que el usuario se haya logueado como hemos comentado previamente.

Una vez visualizado el nombre y precio del producto, si el usuario no desea más información con quitar el puntero de encima de la imagen esta vuelve a su estado inicial, pero si desea más información o añadir el producto al carrito de la compra con pinchar sobre la imagen redirige automáticamente a la página de cada producto.

En la página de cada producto además de poder comprobar sus características, el usuario puede añadirlo al carrito en caso de que haya decidido comprarlo. Esta opción hemos decidido que no aparezca cuando el usuario no está registrado o logueado en la web, algo que nos diferencia de otras webs similares del mercado, y es que pensamos que no tiene sentido dejar a cualquier usuario que pueda añadir productos al carrito sin que se haya dado de alta en nuestra web previamente.

Últimas Novedades



7. Configuración servidor Raspberry Pi

Para la instalación del servidor de DHCP, FTP y correo se ha utilizado el sistema operativo Raspbian, una variante de Debian adaptada para Raspberry Pi.

Para la instalación del servidor DHCP se ha utilizado dnsmasq, para su instalación se han utilizado los siguientes comandos:

```
sudo apt-get install dnsmasq
```

Una vez instalado procederemos a configurarlo, para ello debemos acceder al fichero de configuración en /etc/dnsmasq.conf

```
interface=eth0
```

```
dhcp-range=172.16.0.10, 172.16.0.100,255.255.0.0,24h
```

```
dhcp-option=3, 172.16.0.1
```

```
#Añadimos los host que queremos que tengan siempre la misma IP
```

```
dhcp-host=28:d2:44:16:cd: 172.16.0.11
```

Para la instalación del servidor ftp en el que se almacenarán los ficheros de la empresa hemos utilizado vsftpd, para su instalación se han utilizado los siguientes comandos:

```
sudo apt-get install vsftpd
```

Creamos la carpeta ftp en el directorio /bin

Añadimos la siguiente línea al fichero cat /etc/shells:

```
/bin/ftp
```

Creamos los usuarios, los añadimos al grupo ftp y les otorgamos permisos.

Creamos el fichero vsftpd.chroot_list en el directorio /etc y creamos contraseñas para los usuarios

Modificamos el fichero /etc/vsftpd.conf y habilitamos la opción para que los usuarios se puedan loguear y escribir en el servidor.

```
#
# Run standalone? vsftpd can run either from an inetd or as
# daemon started from an initscript.
listen=YES
#
# Run standalone with IPv6?
# Like the listen parameter, except vsftpd will listen on an
# instead of an IPv4 one. This parameter and the listen param
# exclusive.
#listen_ipv6=YES
#
# Allow anonymous FTP? (Beware - allowed by default if you co
anonymous_enable=YES
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
```

Para la instalación del servidor de correo hemos decidido utilizar el servidor de correo postfix:

Sudo apt-get install postfix

Instalamos el paquete dovecot con:

Sudo apt-get install dovecot-pop3

Configuramos el fichero /etc/postfix/main.cf con los datos de nuestra red, la carpeta donde queremos que se almacenen los usuarios y la carpeta donde se almacenen los emails.

Instalamos el paquete para habilitar imap

Sudo apt-get install dovecot-imapd

Configuramos el fichero /etc/dovecot/dovecot.conf con los datos del puerto en el que queremos que escuche y habilitamos algunas opciones de seguridad como deshabilitar la autenticación en texto plano.

8. Servidor de dominio

Con el objetivo de aumentar la seguridad y facilitar la gestión de usuarios se ha instalado un servidor con Windows Server 2012 R2.

Para instalar el controlador de dominio activo hemos utilizado el apartado agregar roles y características y hemos añadido los servicios AD DS y DNS. Para configurar el DC (Controlador de Dominio) hemos creado un nuevo bosque de dominio con el nombre telephonos.local y hemos seleccionado la opción de añadir a las DNS.

Posteriormente hemos creado los usuarios con el formato nombre.apellido en el dominio y hemos unido los PCs al mismo.

Finalmente se han realizado dos scripts en Powershell, uno para listar los usuarios del dominio y otro para crear carpetas personalizadas para cada usuario del dominio.

Listar usuarios

```
import-module activedirectory
get-ADuser -filter 'samaccountname -like "*.*"' | select SAMAccountName > usuarios.txt
```

Crear carpetas

```
$folder="\172.16.0.11\Personal";
$txtFile="C:\Users\administrador.telephonos\usuarios.txt";
$pattern="*.*";
```

```
get-content $txtFile | %{
```

```
    if($_ -like $pattern)
```

```
    {
```

```
        try
```

```
        {
```

```

    $NewFolder = New-Item -Path $folder -Name $_ -ItemType "Directory"

    $Rights =
[System.Security.AccessControl.FileSystemRights]"FullControl,Modify,ReadAndExecute,ListDirectory,Read,Write"

    $InheritanceFlag =
@([System.Security.AccessControl.InheritanceFlags]::ContainerInherit,[System.Security.AccessControl.InheritanceFlags]::ObjectInherit)

    $PropagationFlag = [System.Security.AccessControl.PropagationFlags]::None

    $objType = [System.Security.AccessControl.AccessControlType]::Allow

    $user = "telephonos.local\"$_"

$ObjACE = New-Object System.Security.AccessControl.FileSystemAccessRule("$user",
$Rights, $InheritanceFlag, $PropagationFlag, $objType)

    $ACL = Get-Acl -Path $folder

    $ACL.AddAccessRule($ObjACE)

    $ObjACE =
[System.Security.AccessControl.FileSystemAccessRule("kdes.local\IT", $Rights,
$InheritanceFlag, $PropagationFlag, $objType)

    $ACL.AddAccessRule($ObjACE)

    Set-ACL -Path $NewFolder.FullName -AclObject $ACL

    # $ObjACE = New-Object System.Security.AccessControl.FileSystemAccessRule `
    # ("telephonos.local\Administradores", $Rights, $InheritanceFlag,
$PropagationFlag, $objType)

    # $ACL.AddAccessRule($ObjACE)

    # Set-ACL -Path $NewFolder.FullName -AclObject $ACL
}

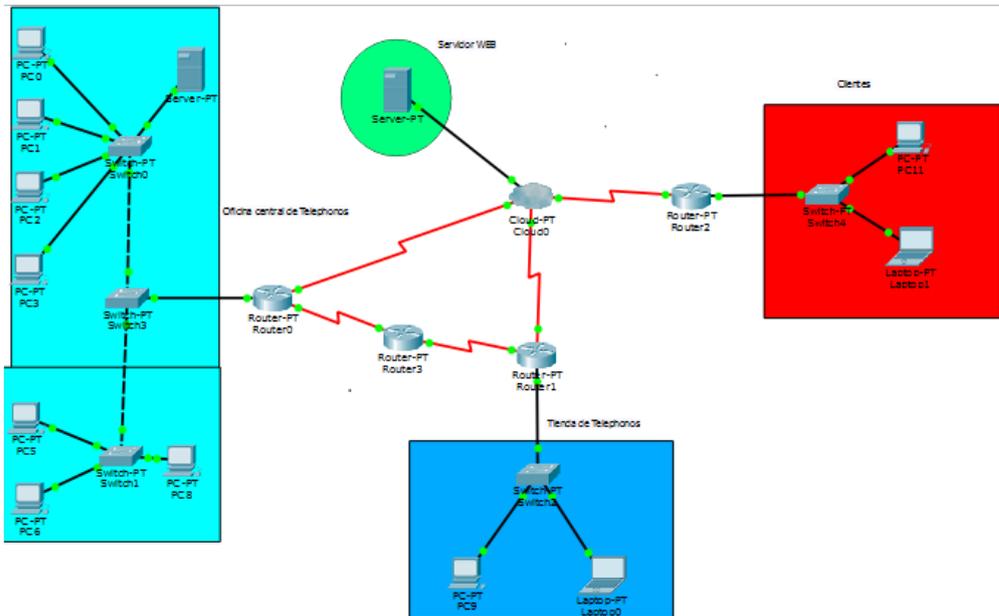
catch
{
    $msg = $_
    echo $msg
}
}
}

```

9. Configuración redes

Tenemos una red empresarial que consta de varias tiendas y oficinas. Una que es la oficina/tienda principal y otras dos que son las tiendas más pequeñas.

Mientras que las tiendas son muy pequeñas y constan solo de varios ordenadores, el eje de la compañía tiene un gran tamaño, con muchas salas y ordenadores. Están divididos en VLANS.



La red WAN tiene enrutamiento OSPF con una nube Frame Relay. El servidor de la web y mysql están online y todos los empleados pueden acceder a ellos desde sus respectivos puestos de trabajo.

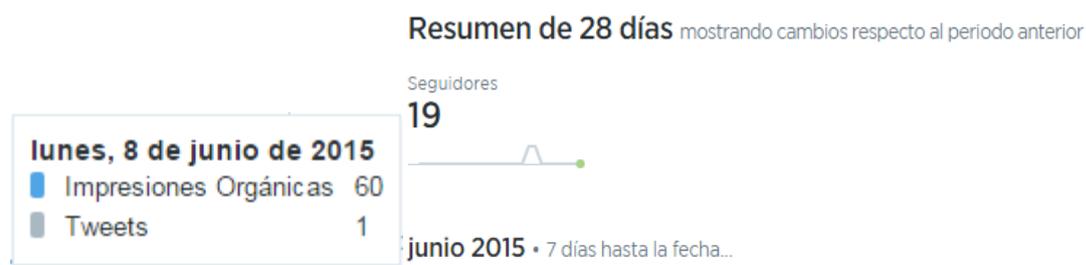
10. Redes sociales y comunicación

En cuanto a la interacción con los usuarios, hemos creado diversas cuentas online para dar a los usuarios asistencia y que puedan conocer de manera instantánea las últimas novedades de la web. Para ello hemos creado cuentas en redes sociales como twitter ([@telephonos_es](https://twitter.com/telephonos_es)) y en Facebook (<https://www.Facebook.com/telephonos>), lugares en las que además de dar soporte comentaremos las últimas novedades de nuestra tienda y noticias tecnológicas que vayan saliendo en el día a día.



Hemos decidido fijar el tweet que aparece en la imagen para que sea el primero que vea cualquier twittero que acceda a nuestra página (https://twitter.com/telephonos_es)

Para conocer la repercusión y los potenciales usuarios a los que llegan nuestros mensajes en twitter, hemos empezado a colaborar con twitter y su aplicación analytics que monitoriza el número de seguidores que tenemos en nuestra cuenta.



En cuanto a Facebook, hemos creado una página catalogándola en la categoría de Electrónica, y hemos utilizado el mismo estilo que en twitter para generarnos una marca

digital global dejando aparte la red social en la que el usuario esté interactuando con nosotros.



En un futuro para la expansión de la empresa en redes sociales tenemos previsto destinar una parte del presupuesto a lanzar alguna campaña en twitter y Facebook, principalmente en fechas navideñas que es cuando el consumo de los productos que vendemos en la web se dispara.