Dpto. de Electrónica e Informática

Salesianos Atocha

RaspyNET

Proyecto Fin de Ciclo ASIR

Marzo de 2017

Autor: Jorge López Prieto

Tutor: Jose Manuel Prieto Gordo

Índice de contenidos

1. Introducción

1.1. Objetivo

He buscado una forma sencilla pero funcional de que pequeñas empresas puedan tener una pequeña pero funcional Intranet, orientada al servicio técnico de éstas, que más tarde pueda ser ampliada. Además, en dicho pack, se incluye una página web personalizada para que los empleados puedan administrar todo lo relacionado con el apartado técnico de la empresa.

Uno de los problemas de una pequeña empresa es que puede estar empezando en cualquier parte y necesite una rápida expansión en los próximos meses, como cambiar el lugar de trabajo o ampliar la estructura informática de ésta, necesitando sistemas de alta portabilidad y que sean fáciles de implementar más tarde en nuevos sistemas, como servidores dedicados.

Con el sistema que he diseñado de Intranet portátil basada en el novedoso miniordenador conocido como RaspBerry, ahora esto es posible. Utilizo una Raspberry como router para distinguir la intranet de la red global de la empresa que adquiera el servicio, y dicha RaspBerry se encarga de que la información de la intranet pueda salir pero que a la vez no se pueda acceder a ésta desde fuera, aumentando así la seguridad de los datos que se tienen almacenados dentro de ésta.

Luego como mínimo se implementan dentro de la intranet dos Raspberry más, una que hace como servidor WEB y otra como servidor de BBDD, cuyo objetivo es el de mantener la página web internar siempre activa e individualizar los servicios de las dos RaspBerry para que así el tiempo de consulta a la base de datos y el tiempo de acceso a la web sean lo más óptimo posible. Con el coste de las RaspBerry todo esto es posible, y barato.

La web debe ser lo más intuitiva y rápida posible para los usuarios, ya que, aunque se supone que los usuarios principales van a ser técnicos con conocimientos informáticos altos, debe ser rápida de entender tanto por ellos como por cualquier persona de la empresa que necesite consultar algo.

1.2. Justificación

He elegido este proyecto ya que principalmente se requieren conocimientos o ampliaciones de éstos que hemos dado a lo largo del año en ASIR. Lo que quería era ampliar mis conocimientos en algunos campos realizando investigación y aprendiendo nuevas cosas sobre lo que ya conocía.

Si tuviera que conocimientos he aplicado, serían:

- Por el lado de desarrollo web: **HTML5**, **CSS**, **PHP**, **JavaScript**, **AJAX**, **jQuery**.
- Por el lado de BBDD: **MySQL**.
- Luego en lo tocante a las RaspBerry me ha tocado investigar servicios y distintas formas de aplicaros, especialmente los orientados a **FTP**, **BBDD** y **WEB**. Así como nuevas formas de utilizar Raspbian para, por ejemplo, hacer una RaspBerry router.

1.3. Propuesta detallada

Aunque el objetivo del proyecto ya está claro como se ha explicado en el punto **[1.1]**, hay recalcar que la Intranet, para ver su potencial, hay que verla desde un punto de realismo en el que esté implementado en una empresa y se vea cómo funciona y sus capacidades, así como sus limitaciones.

Teniendo esto en cuenta, de decidido que voy a emular que una empresa llamada **Servitrek**, que acaba de empezar y se dedica al servicio técnico mediante llamada de problemas que pueden surgir relacionados con dicho sector. Es una empresa pequeña y solo tiene 10 técnicos que se encargan de responder a las llamadas, a parte del jefe que es el que se encarga de supervisarlos y administrar todo el sistema informático de ésta.

A grandes rasgos lo que la empresa buscaba era lo yo que puedo ofrecer, una forma barata de administrar una página web en la que se lleve a cabo el procesamiento de las incidencias relacionadas con las llamadas telefónicas de los clientes, así como algunos servicios extra para los empleados que puedan resultar útiles: Un servicio de almacenamiento personal, una agenda individual, mensajería interna, un chat para poner conocimientos en común y el apartado más importante, el de la gestión de las incidencias.

Así mismo para el administrador se buscaba que pudiera realizar las labores normales en una página web teniendo en cuenta el Back-end común. Algunas de éstas funciones son reiniciar el chat, administrar a los usuarios, supervisión mediante web de los servidores y un control global sobre las incidencias.

Al ser una empresa pequeña, se han necesitado pocos recursos, la evaluación de los costes necesarios para su implementación están explicados en el próximo punto [**2**].

2. Planificación temporal y evaluación de costes

Tras tener clara la idea de lo que iba a hacer, hice labores de investigación en diciembre para ver cómo iba a llevar a cabo el proyecto, ya que no sabía si todo iba a ser viable desde un principio, aunque en algunas cosas acabé arriesgándome o buscando alternativas, y gracias a eso descubrí nuevas formas de hacer las cosas.

En enero me puse con el apartado Hardware del proyecto, es decir los servidores (RaspBerry) y la forma en las que iba a enlazarlos. Al final me decanté por la forma más barata y eficaz de la que disponía con mi presupuesto y lo que tenía al alcance para implementar todo lo que había investigado anteriormente. Esto me llevo una semana entre que venían un par de pedidos. Todo el tiempo hasta mediados de mes lo dediqué a instalar los servicios que iba a utilizar, así como optimizar los 2 servidores y la RouterBerry (la llamaré así a partir de ahora para abreviar).

El resto del mes de enero lo utilicé para empezar a realizar plantillas de la web y pensar en un Template (css) que fuera intuitivo para los empleados, rápido a la hora de cargar y que tuviera sentido para una web de servicio técnico, así como alta usabilidad.

A la hora de evaluar los costes, si tuviera que calcular lo que le ha costado a la empresa el proyecto desde 0, había que tener en cuenta dos apartados:

- **Dispositivos necesarios para el funcionamiento:**

Dispositivo	Unidades	Precio total de unidades
RaspBerry PI 3	3	120 €
Switch de 16 puertos	1	76 €
Tarjeta SD 65GB	3	69 €
Cargador RaspBerry Standard	3	27 €
Caja Raspberry con ventilador	3	27 €
Conector USB-Ethernet	1	10 €
	Inversión Total	329 €

- **Diseño de la web y montar la intranet:**

Hay que tener en cuenta que la puesta a punto de todo el sistema, así como la elaboración de web suponen una gran cantidad de tiempo, más si se quiere todo optimizado y que no se ha utilizado nada relacionado con los CMS, así que podríamos valorarla página en unos 800 € más la puesta a punto de todos los dispositivos y sus funciones otros 200 €.

Teniendo en cuenta la suma de todos los costes, diremos que a la empresa le saldrá todo el pack por unos 1350 €, y que además si quisiera mantener un mantenimiento constante y supervisión deberá pagar al mes 20 € para conseguir la atención más rápida posible por mi parte, así como mejoras ocasionales.

3. Diseño

3.1. Diseño arquitectónico

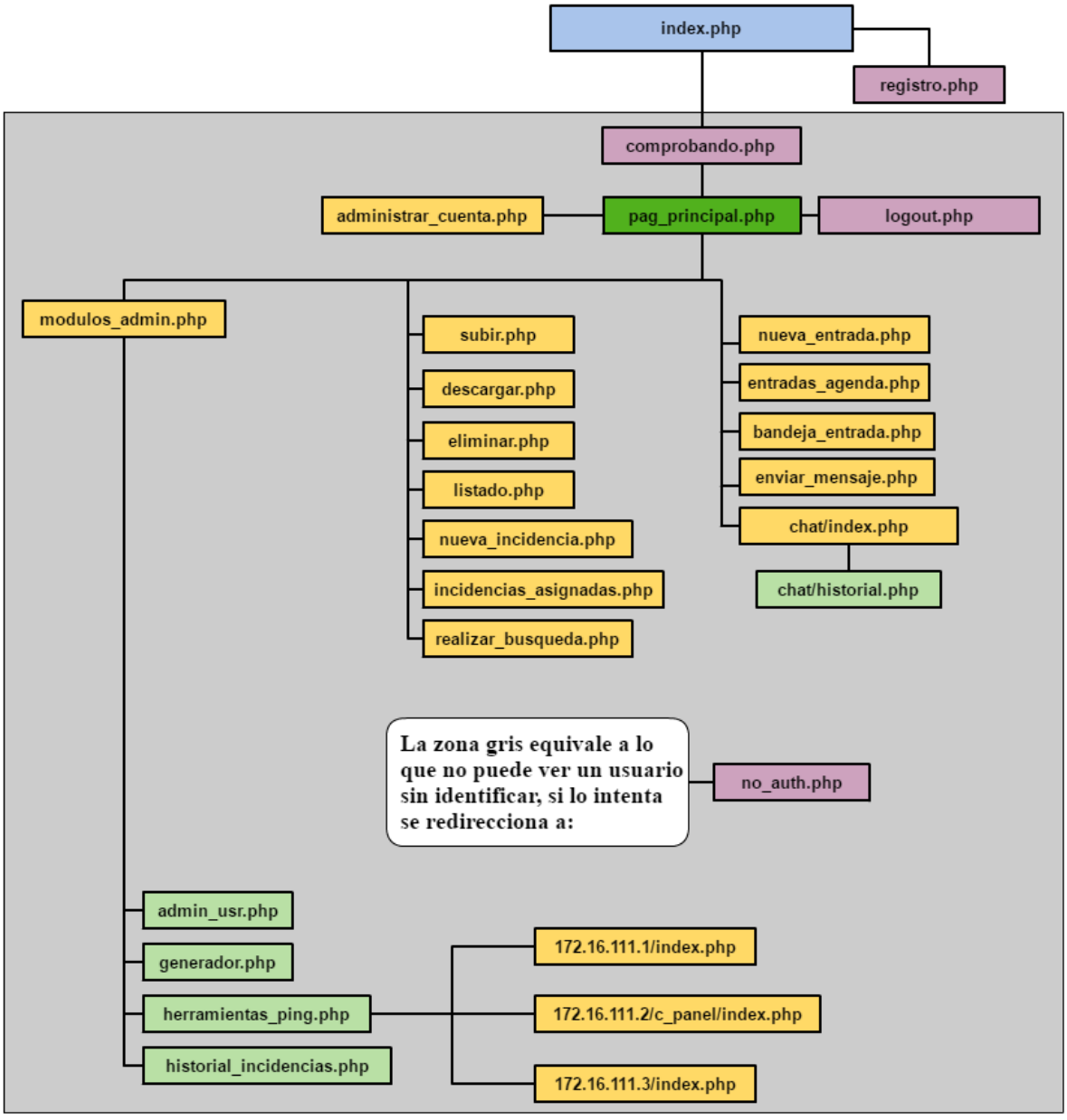
De nuevo, como en el caso anterior, se deberá observar el diseño arquitectónico que se ha utilizado en la web, así como la relación que tienen todos los dispositivos en la intranet.

Empezando por el apartado web, se pueden dar tres casos:

- **Ser un usuario sin identificar**
- **Ser un empleado sin poderes de administración**
- **Ser un empleado con poderes de administración**

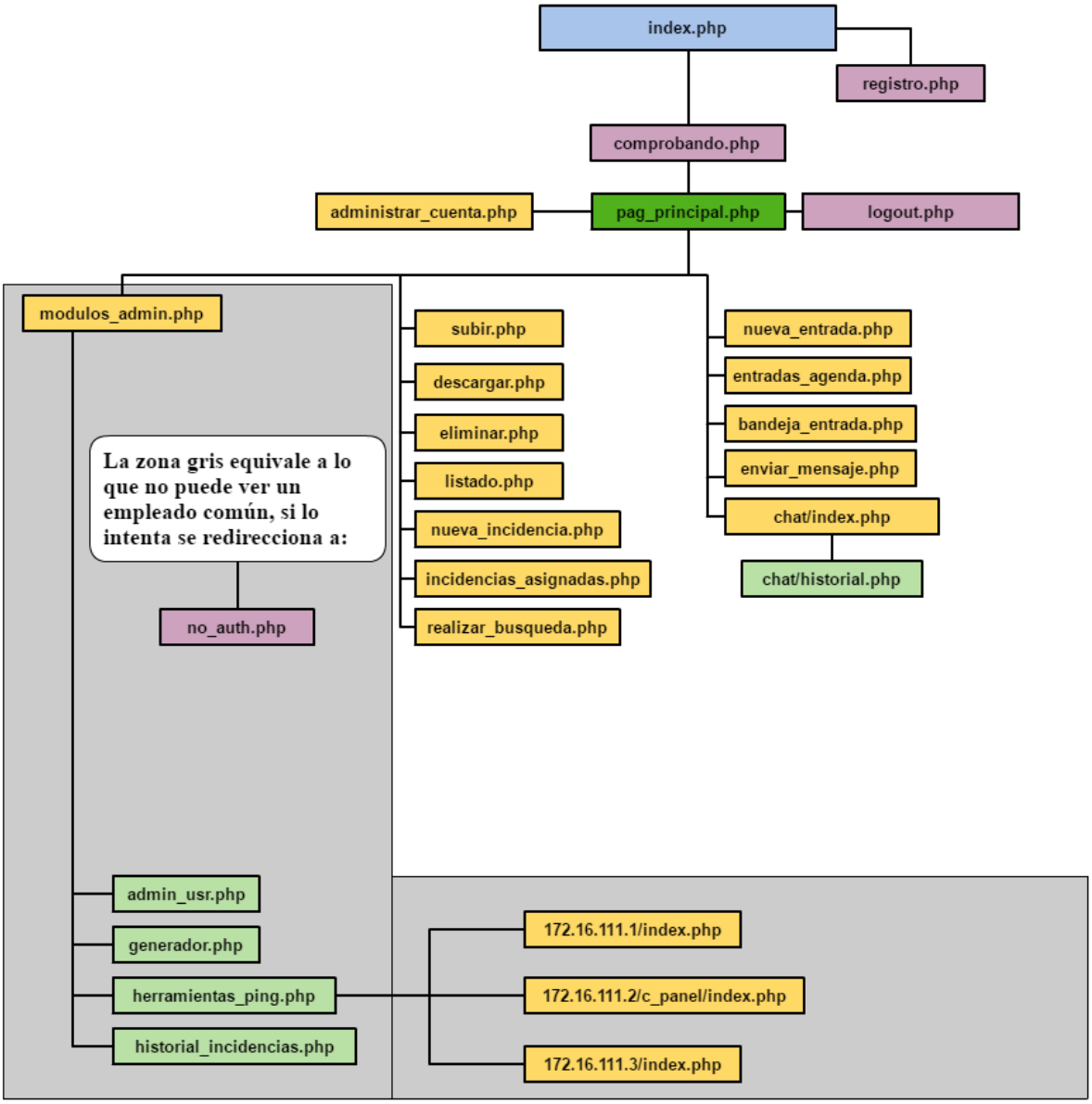
Vamos a observar lo que puede ver cada empleado si se conectara a la página web, así como lo que ocurriría si intentase entrar a una página web a la que no tiene acceso y lo que ocurriría en dicho caso.

Usuario sin identificar



Un usuario sin modificar solamente puede acceder al inicio y al apartado registro. Cualquier intento de acceso a cualquiera de los otros apartados desembocará en un aviso de acceso sin identificación que automaticamente le redirigirá a la página **no_auth.**php donde puede volver a intentar hacer login o registrarse.

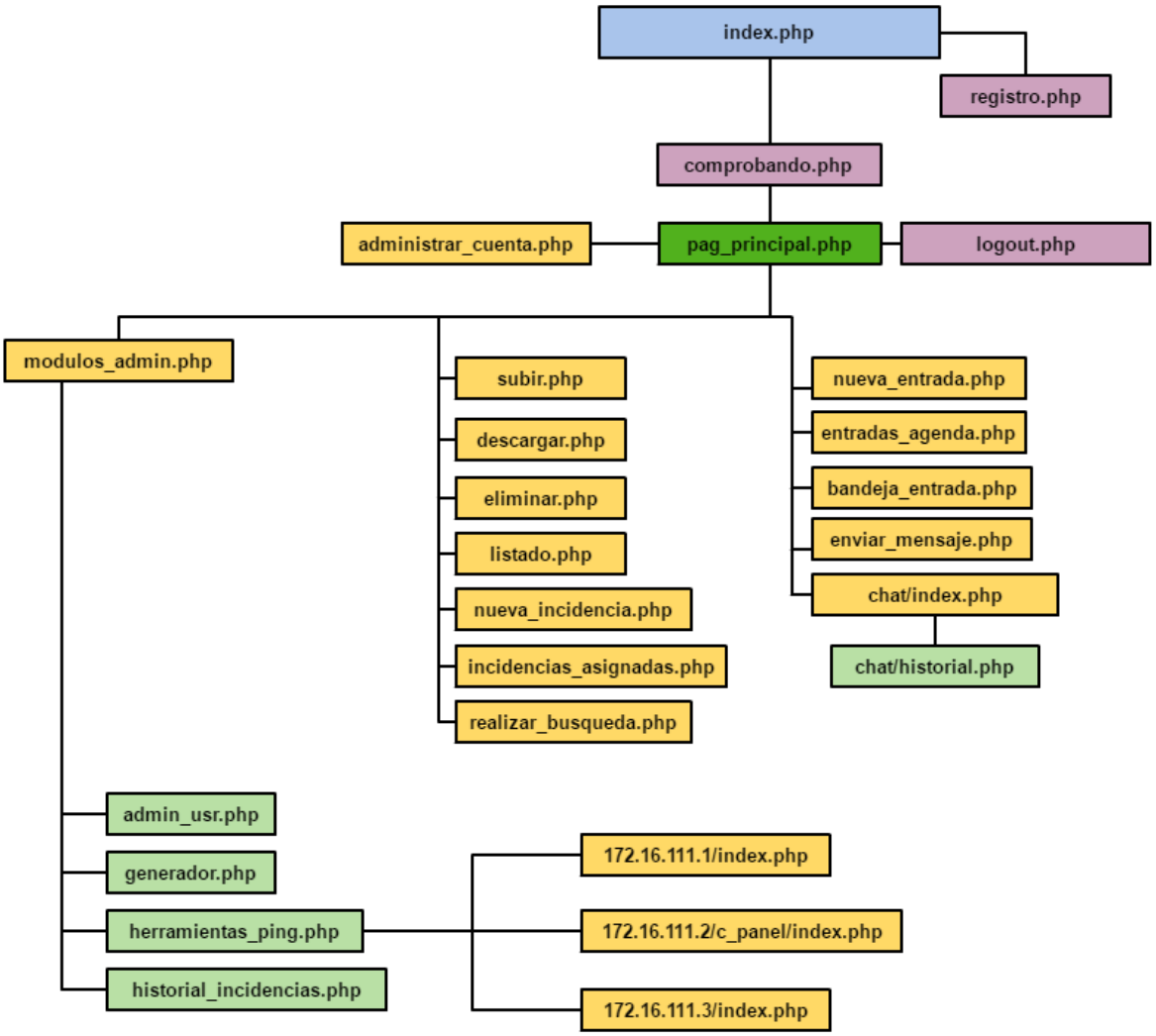
Empleado común



Cualquier empleado que no sea administrador ve todo menos el apartado de administración. Si intenta acceder a dichos apartados le saldrá un mensaje de que no tiene suficientes privilegión y un login por si quiere acceder desde otra cuenta.

Si intenta acceder a alguno de los servidores mediante herramientas_ping.php automaticamente será rederidigido a un apartado de login para comprabar los niveles de privilegio de su cuenta.

Empleado con privilegios de administrador



Puede acceder a cualquier apartado de la página web como se puede observar. Cuando accede a cualquier lado tiene permisos. En ningún momento puede acceder a la página no_auth.php debido a que tiene los privilegios mas altos posibles.

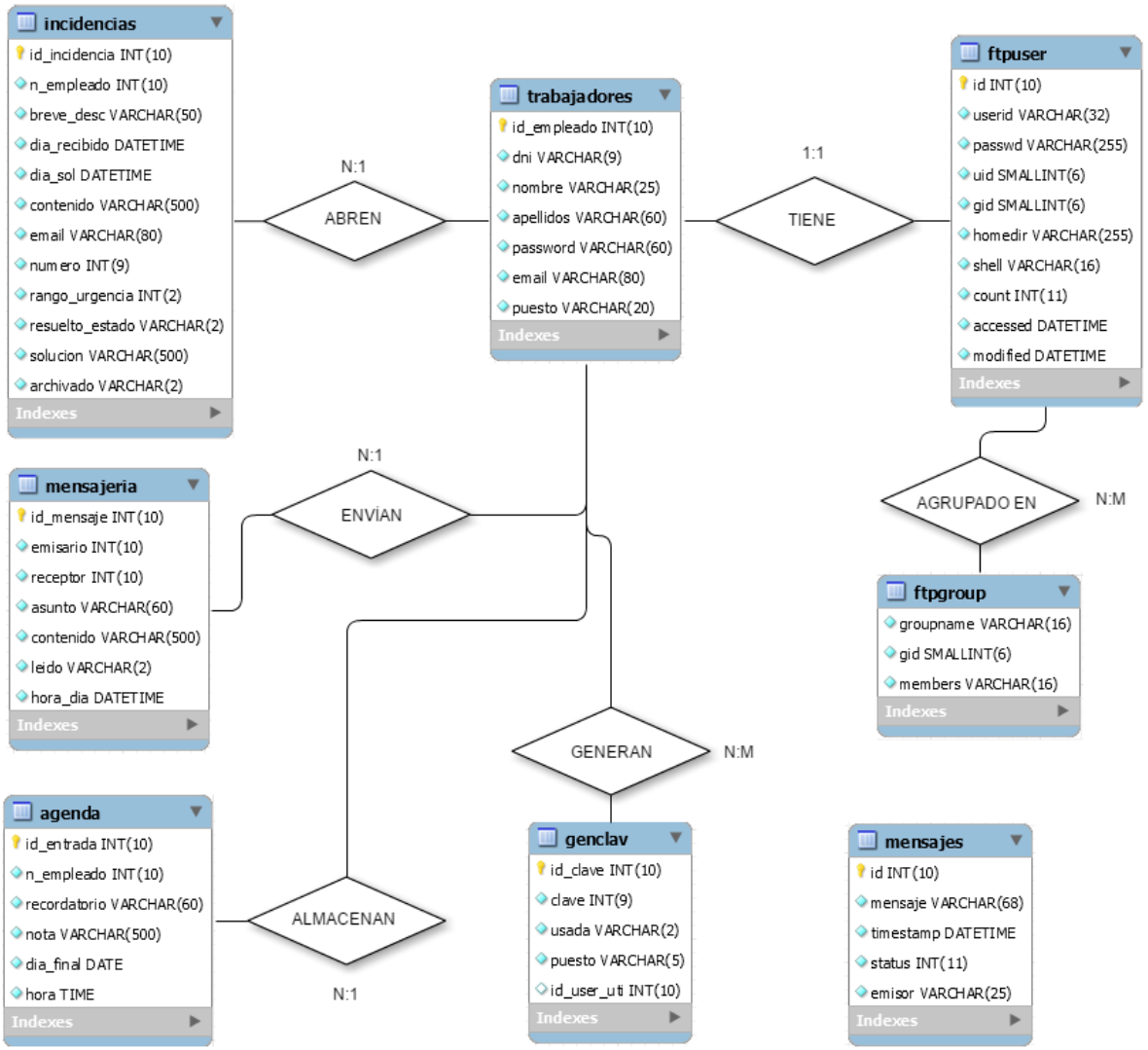
Solo tiene una pequeña limitación de seguridad y es que no puede acceder al apartado de información del servidor BBDD y la RouterBerry sin pasar de nuevo por un apartado de autentificación. Esto se ha hecho para incrementar la seguridad y aunque todos los login se basan en la misma base de datos, cada RaspBerry tiene su sesión individual.

Al abandonar cualquiera de las páginas de información de servidor automáticamente se hace log out por seguridad.

3.2. Diseño de datos

Como el proyecto está basado en una página web, obviamente se ha requerido de una base de datos para almacenar toda la información necesaria. Además, el servidor FTP lo he implementado con dicha base de datos para un mejor manejo de usuarios y de organización.

El modelo de entidad-relación utilizado se basa en 8 tablas/entidades, de las cuales 7 tienen algún tipo de relación entre ellas, es el siguiente:



A continuación, procedo a explicar la función de cada una de las tablas y sus variables:

- **Incidencias:** En esta entidad se almacenan todos los datos recogidos en lo referente a las incidencias por los empleados de la empresa **Servitrek**. La función de cada una de las variables es:

Id_incidencia: id de la incidencia.

N_empleado: id del empleado asignado a la incidencia.

Breve_desc: descripción pequeña o título identificador de la incidencia.

Dia_recibido: día en el que se abre la incidencia.

Dia_sol: día en el que se da la incidencia como resuelta.

Contenido: descripción del problema.

Email: email del cliente con el problema.

Número: número de teléfono del cliente con el problema.

Rango_urgencia: prioridad valorada del 1 al 10 del caso (1 mayor prioridad, 10 menor).

Resuelto_estado: define si está resuelto o no el problema.

Solución: solución aportada para el problema.

Archivado: define si está archivado o no el problema.

- **Mensajería:** Esta entidad se utiliza para almacenar los mensajes que los usuarios se mandan entre ellos, así como variables necesarias para el procedimiento. La función de cada una de las variables es:

Id_mensaje: id del mensaje.

Emisario: id del empleado que envía el mensaje privado.

Receptor: id del empleado que recibe el mensaje privado.

Asunto: asunto del mensaje, como en los correos electrónicos.

Contenido: contenido del mensaje.

Leído: variable utilizada para comprobar si se han leído los mensajes o no.

Hora_dia: hora en el que se ha enviado el mensaje.

- **Agenda:** Esta entidad se utiliza para almacenar los eventos que los usuarios almacenan en la agenda. La función de cada una de las variables es:

Id_entrada: id del evento.

N_empleado: id del empleado que añade el evento.

Recordatorio: asunto o breve descripción de lo que es el evento.

Nota: descripción completa del evento.

Dia_final: día en el que ocurre el evento, automáticamente pasa al historial al siguiente día.

Hora: hora en la que ocurre el evento a lo largo del día.

- **Trabajadores:** Esta entidad se utiliza para almacenar los empleados y administradores. Son los datos de los que se dispone, así como de sus datos de inicio de sesión. La función de cada una de las variables es:

Id_empleado: id del empleado.

Dni: dni del empleado utilizado para iniciar sesión, es único obviamente.

Nombre: nombre del empleado.

Apellidos: apellidos del empleado.

Password: contraseña cifrada del empleado para iniciar sesión.

Email: email del empleado.

Puesto: define si es un usuario (user) o un administrador (admin).

- **Genclav:** Esta entidad se utiliza para almacenar las claves utilizadas para registrar un nuevo empleado. La función de cada una de las variables es:

Id_clave: id de la clave generada.

Clave: números que forman la clave para pedirla en el registro.

Usada: define si la clave ha sido utilizada con un si/no.

Puesto: define si va a ser un usuario (user) o un administrador (admin).

Id_user_uti: el id del usuario que ha utilizado la clave al registrarse.

- **Mensajes**: Esta entidad es independiente y se utiliza para almacenar y procesar los mensajes del chat incluido en la página web. La función de cada una de las variables es:

Id: id del mensaje.

Mensaje: contenido del mensaje en el chat.

Timestamp: variable de tiempo en el que se ha enviado necesaria para procesar en Ajax.

Status: almacena el valor 0 (no se ha procesado) o el 1 (ya se ha procesado).

Emisor: nombre de la persona que envía el mensaje basado en id.

- **Ftpuse**r: Entidad que es utilizada para almacenar los datos de los usuarios para poder realizar las conexiones con el servidor FTP. La función de cada una de las variables es:

Id: id del usuario dentro del ftp.

Userid: id del usuario basado en la tabla trabajadores.

Passwd: contraseña cifrada para el ftp (misma que la del login, pero usando otro cifrado).

Uid: id del usuario dentro del sistema Raspbian en la RaspBerry que almacena el ftp.

Gid: id del grupo dentro del sistema Raspbian en la RaspBerry que almacena el ftp.

Homedir: dirección asignada automáticamente en el servidor, almacén/empleados/x

Shell: almacena el valor nologin para no requerir loguearse en la RaspBerry por usuario.

Count: número de veces que se ha utilizado ese usuario para acceder al ftp.

Accessed: último acceso realizado por el usuario.

Modified: última vez que el usuario ha hecho alguna acción una vez conectado al ftp.

- **Ftpgroup**: Entidad que se utilizada para almacenar los datos de los grupos para poder realizar las conexiones con el servidor FTP. La función de cada una de las variables es:

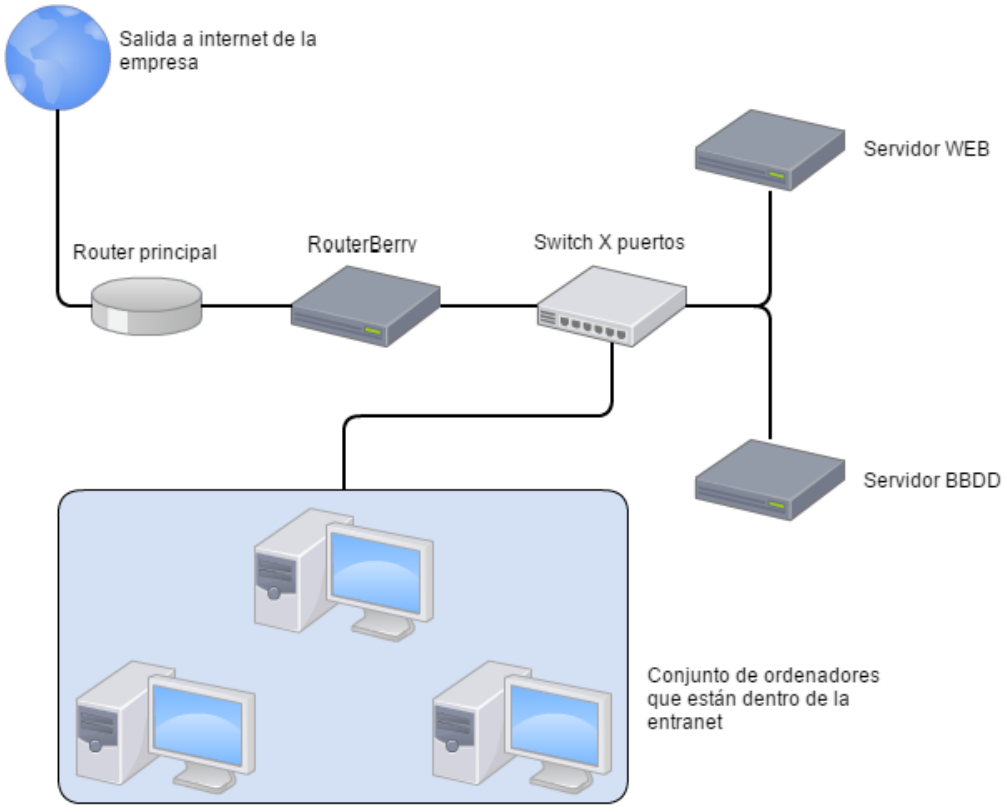
Groupname**:** nombre del grupo.

Gid**:** id del grupo.

Members**:** miembros que forman parte del grupo.

3.3. Arquitectura de la red y componentes

Aquí puesto un mapa conceptual de cómo está organizada la red y sus componentes, aunque más tarde en el punto [**5**] lo explicaré con mayor detalle.



4. Lenguajes

Voy a hablar de los lenguajes que he utilizado y las cosas para los que los he aplicado a la hora de realizar el proyecto. Haré mucho más hincapié en las cosas nuevas o que han tenido mayor dificultad a la hora de implementarlas.

HTML5

Toda la parte visual de la página está basada en HTML5. Gracias a este lenguaje de marcas se han podido realizar cosas como establecer las divisiones de la página (el 95% de la web se divide en 5 apartados, <header><nav><article><section><footer>).

En base a estas divisiones he podido aplicar posteriormente el CSS, de lo que hablaré más tarde.

Así mismo todo lo referente a mostrar imágenes, texto, tablas, encabezados, hipervínculos, llamadas a script, etc… se realizan desde el HTML. No lo he utilizado para nada fuera de lo común así que no me voy a centrar en explicar éste apartado, ya que todo lo importante está trabajado en el PHP o la parte visual (CSS).

CSS

Toda la parte visual está hecha con CSS a mano exceptuando algunas partes en las que he utilizado PureCSS. Consiste en un conjunto de librerías con diseños predefinidos de tablas, formularios, botones, etc. que he utilizado en mi página para seguir la misma temática y ahorrar trabajo en diseñar dichas cosas por mí mismo

La otra alternativa era utilizar BOOTSTRAP, pero alteraba tantas cosas que no me interesaba puesto que quería una página echa a medida por mí mismo. Casi todas las páginas se basan en la misma hoja de estilo (style_pag_principal.css), y las pocas que han necesitado algún reajuste se basan en dicha hoja clonada solo que con los cambios hechos pertinentes.

PHP

Es el lenguaje que más he utilizado en la página web con diferencia, especialmente en la interacción de la web con la base de datos. A continuación, haré mención de las partes que me han resultado más complejas en lo referente al PHP dentro de la web:

- **Registro de un nuevo empleado**:

Hacer un registro mediante PHP y MySQL es muy sencillo, el problema viene cuando debes conseguir que simultáneamente se genere una carpeta en el servidor para dicho usuario y que a la vez se establezca un usuario para el ftp que sea único para éste y le lleve a la carpeta generada anteriormente.

Para esto he necesitado utilizar el comando mkdir (creando una carpeta con permisos totales) desde PHP y un insert automático para que el usuario a la vez que se registraba en la base de datos de los trabajadores, automáticamente también se registrara como usuario del ftp.

```php
$oldmask=umask(0);
mkdir($path, 0777);
unmask($oldmask);
```

Patch es la ruta que he generado previamente para cada usuario. Tuve problemas para dar permisos a la carpeta de 777 ya que no me dejaba hacerlo directamente, así que había que cambiar la umask previamente y lueg odejarla como estaba.

El problema que surgía tambien al insertar el usuario del FTP en la base de datos es que el método de cifrado que utilizaba en un principio para los empleados no era válido, puesto que el plugin de mysql que leía la base de datos no reconocía ese hash, así que tuve que usar otro llamado “crpyt” como se aprecia en la imagen.

```php
$pass2 = crypt($password2);
```

El resto de cosas a la hora de hacer el registro son normales y no hay nada fuera de lo común o que haya dado mas problemas.

- **La interacción del FTP con PHP:**

En un principio parecía una cosa sencilla, busqué en diciembre formas de hacerlo y no encontré problema en ello. Claro está que en ese momento yo estaba pensando en que el FTP tendría ya la carpeta creada y solo habría un usuario.

Me tuve que plantear como hacerlo y al final almacenando la id y contraseña (con hash) del usuario en la sesión mediante $_SESSION. La función que utilizo para hacer las conexiones al ftp es la siguiente:

```php
$ftp_server = '172.16.111.2';
$ftp_user_name = $login_ftp;
$ftp_user_pass = $pass_ftp;

$conn_id = ftp_connect($ftp_server);

$login_result = ftp_login($conn_id, $ftp_user_name, $ftp_user_pass);
```

Todas las conexiones al FTP se basan en las mismas condiciones y funcionan perfectamente, así respetamos que cada empleado tenga su carpeta personal y podemos contabilizar de forma independiente que han subido y cuanto espacio ocupan.

El otro problema que surgió entre el FTP y el PHP fue a la hora de descargar los archivos mediante una conexión FTP, así que utilicé otro sistema bastante eficaz que además aceleraba los procesos de conexión y a su vez la descargar.

En lugar de valerme de las funciones de conexión, me basé en los comandos de php **array_diff**, **scandir** y **readfile** para, mediante bucles, gestionar la descarga de los archivos. La idea era mediante scandir y array_dif almacenar los archivos en un array para luego mostrarlos en una lista y mediante la utilización de header forzar al php para mostrar el archivo que seleccionabamos mediante el radio (no permite descarga múltiple de archivos).

Para forzar al php a descargar el archivo utilizo el siguiente código:

```php
if(file_exists($filename)){


    $finfo = finfo_open(FILEINFO_MIME_TYPE);
    header('Content-Type: ' . finfo_file($finfo, $filename));
    finfo_close($finfo);


    header('Content-Disposition: attachment; filename='.basename($filename));

    header('Content-Transfer-Encoding: binary');
    header('Cache-Control: must-revalidate, post-check=0, pre-check=0');

    header('Expires: 0');

    header('Pragma: public');

    header('Content-Length: ' . filesize($filename));

    ob_clean();
    flush();
    readfile($filename);
    exit;
}
```

Básicamente éste trozo de código comprueba el nombre del archivo que selecciono con el array, comprueba que tip ode archivo es (png, pdf, exe…) lo develve en formato basename y lo envía al navegador mediante **redfile** para procesarlo como archivo de descarga.

He probado con muchas extensiones y ninguna me ha dado problema, así que podría decirse que funciona perfectamente.

- **Conexión al panel de información de los distintos servidores**:

Dentro de la sección de administracción, en la gestión de servidores, hay tres botones llamados “Informacción de…” cada uno corresponde a un servidor. En el Servidor WEB no tuve ningún tipo de problema, puesto que se alojan en la misma RaspBerry y comparten la misma sesión, pero no ocurre lo mismo con el servidor de BBDD ni la RouterBerry.

Al funcionar en un servidor independiente tuve que encontrar alguna forma de que solo pudiera ver la información los administradores, así que al final opté por diseñar una página simple de login para ambos servidores, que se basara en la misma base de datos que la de los trabajadores, para que solo los que fueran administradores pudieran ver dichas páginas.

Tambien tuve que instalar un plugin para que el PHP se conectase mediante SSH al servidor y así recabar información que luego se muestra en dicha página. Al abandonar la página en cualquier servidor se cierra automáticamente la sesión.

- **Mensajería y agenda**:

Como ya tenía suficiente haciendo el chat con jQuery y AJAX, el cual me había costado bastante implementarlo (hablaré mas tarde de éste), en vez de buscar actualizaciones en vivo tanto para cuando se recibiera un mensaje como para los eventos próximos, lo que he hecho ha sido realizar un select cada vez que se cambia de página, así que si se ha recibido un mensaje o un evento se ha marcado como urgente (esto ocurre automáticamente cuando al evento le quedan 7 días para cumplirse), automáticamente dicho número se vera modificado.

Además, si el número es superior a 1, éste se convertirá en un hipervínculo que te llevara a la sección correspondiente.

Los mensajes se considerarán como leídos cuando se acceda a la bandeja de entrada de éstos mediante un insert que cambia e lestado de leído "no" al estado "si", volviendo en el menú de usuario a aparecer que no hay mensajes nuevos.

- **Contraseña en blanco al modificar un usuario:**

Obviamente si queremos modificar a un usuario o un usuario quiere modificar algun apartado de su cuenta, no tiene porque querer cambiar la contraseña. En un principio no pense en ello y obviamente cuando le daba a modificar la contraseña del usuario estaba en blanco así que si que se modificaba.

Esto lo resolví haciendo que cuando el usuario dejara en blanco la contraseña, mediante un IF en php, no se tomaba en cuenta el campo "password" del usuario y se quedaba la que ya estaba introducida.

- **Prohibir el acceso del usuario a páginas escritas manualmente**:

Esto lo hablaré mas detalladamente en el apartado de seguridad, pero básicamente cuando intentas acceder por ejemplo a **http://172.16.111.2/pag_principal.php** sin estar logeado, automáticamente el php te reenvia a una página llamada **no_auth.php** en la que se te avisa que no tienes los permisos adecuados y te señala que debes logearte con privilegios mas elevados. Esto ocurre con absolutamente cualquier página del servidor.

Tambien tendo prohibido al acceso de carpetas utilizadas por el php como puede ser img, libs, style…

jQuery

El principal uso que le he dado en toda la página es a una librería suya, llamada jQuery.validate. Dicha librería se encarga de muchas funciones a la hora de enviar formularios para filtrar eventos y procesarlos en vivo.

Por ejemplo a la hora de registrar un usuario, en el formulario, si no pongo un DNI que tenga exactamente 9 digitos el formulario no se enviará, o si la contraseña y la comprobación de contraseña no coinciden ocurrirá lo mismo. Todos estos filtros los he puesto para evitar evitar equivocaciones del usuario, y evitar entradas en la base de datos no permitidas o que no sea

válidas, aumentando así un poco la seguridad de la web, ya que se basa bastante en formularios.

Un ejemplo básico de jQuery validate es la función para el login:

```
$(function() {

  $("#login_form").validate({

          rules: {
           username: {
           required: true,
           minlength: 9,
     maxlength: 9,

},

           password: {
               required: true,
               minlength: 6,
               maxlength: 20,
           },

       },

       messages: {
           username: {
           required: "<br><span class='decor'>Campo requerido</span>",
           minlength: "<br><span class='decor'>El DNI debe contener 9 carácteres</span>",
     maxlength: "<br><span class='decor'>El DNI debe contener 9 carácteres</span>",


},

           password: {
               required: "<br><span class='decor'>Campo requerido</span>",
       minlength: "<br><span class='decor'>Contraseña demasiado corta</span>",
       maxlength: "<br><span class='decor'>Contraseña demasiado larga</span>",
        },

},

       submitHandler: function(form) {
           form.submit();
       }
   });

 });
```

Aquí le estamos diciendo a la entrada del DNI que debe contener al menos 9 caracteres y que la contraseña debe tener un límite de 6 como poco y 20 caracteres como máxmimo. #login_form es la id del formulario que se ve afectado pro estas normas, **rules** las reglas que se aplican y **messages** los mensajes que se muestran a la hora de procesar el error.

La llamada a jQuery.validate se hace en el head de la siguiente forma (requiere jQuery):

```
<script src="//ajax.aspnetcdn.com/ajax/jquery.validate/1.9/jquery.validate.min.js"></script>
```

Tambien he utilizado jQuery junto a ajax para realizar el chat, que es de lo que hablaré a continuación:

AJAX

El chat está basado principalmente en AJAX, aunque gran parte lo he hecho siguiendo varios tutoriales en internet sobre hacer un chat básico, me ha costado implementarlo en la página web de la forma mas eficaz posible.

Dicho chat consta de 5 archivos principales:

- **Conect.php:**

Almacena la información para conectarsme al servidor MySQL desde los archivos del chat haciendo solamente llamadas con el include, me habría gustado aplicarlo a toda la página pero me enteré tarde de que se podía hacer.

- **Insertar.php**

Se encarga de insertar en la base de datos el mensaje y los valores para que éste funcione con el chat (**timestamp**, que almacena la hora de lenvío del mensaje y el valor **status**, que es 0 si no se ha enviado, 1 si se ha enviado). Estos valores vienen ordenados por el archivo **mensajes.php** del que hablare a continuación.

- **Mensajes.php**

Parte que se ve del chat y es la que interactua directamente con el usuario junto a **index.php**. Es la que recoge el contenido del mensaje así como del usuario que lo envía. Una vez dado al botón enviar, se procesa a la siguiente función en AJAX:

```
$("#en_men").click(function() {

    var url = "insertar.php";

    $.ajax({
            type: "POST",
            url: url,
            data: $("#mensaje").serialize(),
            success: function(data)
            {

            }
          });

    return false;
});
```

Dicha función se encarga de enviar los datos a insertar.php, explicado en el punto anterior. El contenido que envia es “mensaje”, que es la ID del campo que contiene el mensaje. Además mensajes.php se encarga de mostrar el chat en vivo y de procesar visualmente los cambios junto a **index.php**

- **Httpush.php**

Probablemente una de las partes mas importantes del chat, ya que se encarga de hacer comprobaciones solicitadas por **index.php** para ver si se ha enviado algun mensaje mediante este while en php.

Mientras el día y hora del último mensaje de la base de datos sea menor que el día y hora actual, si el servidor ha solicitado una comprobación (cosa que hace cada segundo), se enviará el último mensaje manejado por el chat.

Dicho mensaje se devuelve a **index.php** en forma de json con un echo, con las variables que interesan, para que lo procese adecuadamente.

- **Index.php**

El centro de trabajo del chat. Cada segundo comprueba si ha habido un mensaje nuevo enviando una petición a **httpush.php** haciendo lo que he mencionado anteriormente. Se pueden dar dos casos, que devuelva null (en cuyo caso no hay mensaje nuevo y no se realiza nada mas hasta la próxima activación) o que devuelva un valor, en cuyo caso se activa la segunda parte de la función.

Dicha parte de la función (dentro de un else) se encarga de mandar un aviso a **mensajes.php** de que debe modificar su código html, en éste caso el div "contenido" para que muestre los nuevos mensajes. Una vez echa la llamada a mensajes se activa un select en la base de datos que muestra dichos mensajes (el chat tiene un límite en el que solo muestra los 15 últimos mensajes).

En resumidas cuentas, dentro de **mensajes.php** se procesa el envío del mensaje que es tratado por **insertar.php**. **Index.php** está enviando peticiones continuas a **httpush.php** para ver si hay mensajes y si es el caso se procesa y se envía **a index.php** para que éste lo procese al div contenido dentro de **mensajes.php**.

5. Herramientas utilizadas

He utilizado herramientas de varios ámbitos a la hora de realizar el proyecto.

Para el desarrollo de la web me he valido de **Notepad++,** herramienta ligera para programar perfecta para los lenguajes de marcas. Además, para agilizar todo lo relacionado con subir la web a la base de datos, he utilizado **Filezilla** para actualizar la página web subiéndola mediante ftp cada vez que realizaba algún cambio y trabajar siempre directamente sobre el proyecto, no me valía hacerlo en una máquina virtual y exportarlo luego ya que quizá podría dar errores y así los solucionaba mejor sobre la marcha.

En el apartado de la BBDD he utilizado **MySQL Workbench** tanto para el manejo de tablas como para la modificación de datos.

Para el control de los servidores **VNC View** y **Putty** como formas de manejarlos, puesto que una vez instalados no he requerido del cable HDMI en ningún momento.

6. Hardware utilizado y servicios

6.1. Raspberry como servidor

Se ha buscado hacer una Intranet que sea capaz de transmitir la resolución de DNS, así como de asignar IPs automáticamente mediante el servidor DHCP. También se han bloqueado todas las entradas de fuera hacia dentro para evitar el intento de acciones maliciosas mediante el uso de IPtables, no obstante, de dentro hacia afuera no hay ningún tipo de limitación.

Al buscar que sea portable, se ha requerido automatizar el proceso completamente, así que los únicos conocimientos necesarios para que funcione por parte de un usuario normal son los de conectar los cables de red con los dispositivos (RaspBerry, switch) y encenderlos. Del resto se encarga la RouterBerry.

Teniendo en cuenta el esquema explicado en el punto [**3.3**], podemos diferenciar 3 servidores, uno que funciona como servidor WEB, otro como servidor de BBDD y el que se encarga de organizarlo todo, la RouterBerry, que funciona como pasarela entre la intranet y el router central de la empresa.

En el siguiente punto procedo a explicar los servicios que contiene cada servidor y el porqué de éstos.

6.2. Servicios implementados

Hablaré de los servicios desde el punto de vista de cada servidor, obviando los que vienen por defecto como el de SSH:

- **RouterBerry**:

El servidor que menos sobrecargado debe de estar puesto que se encarga de manejar muchos datos al efectuar de enlace entre el router principal de la empresa y el switch que maneja la intranet.

Tiene instalado es el de **apache2** junto al de **php5**, para poder administrar la página de información a la que se accede desde la página web.

Además, hace de servidor DHCP mediante el servicio **isc-dhcp-server** que se encarga de administrar IPs fijas a los servidores web y de bbdd, así como de asignar IPs dinámicas a los ordenadores de los empleados.

Para que funcione como router se ha utilizado la siguiente IPtable, que habilita la utilización de NAT:

iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE

iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT

- **Servidor WEB:**

Es el que más servicios tiene instalados. Son los siguientes:

Apache2 y php5: Encargados de mantener la web siempre activa y mantener a flote el apartado web del servidor.

Proftpd: Servidor ftp para administrar tanto el almacenamiento de los usuarios como como la página web mediante clientes como Filezilla.

proftpd-mod-mysql: Une ftp con MySQL y establecer el inicio de sesión mediante consulta de base de datos.

Además de estos servicios, es en ella donde se almacenan todos los archivos que el usuario sube previamente al almacén privado mediante la página web.

- **Servidor BBDD**:

Al igual que la RouterBerry, tiene instalado el **apache2** junto al de **php5**, para poder administrar la página de información a la que se accede desde la página web.

MySQL: La única base de datos del proyecto. En ella se almacenan todos los datos.

7. Página WEB

7.1. Usabilidad

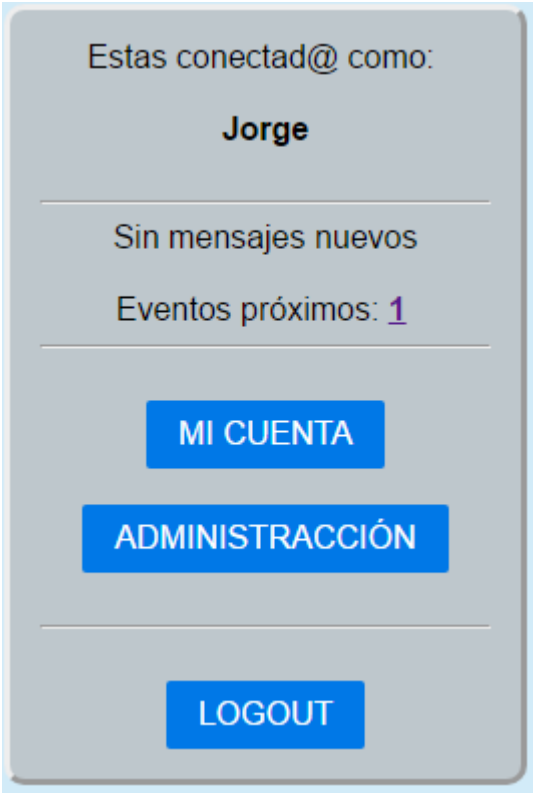
Como se ha comentado en puntos anteriores, se ha buscado que la web sea sencilla de entender para cualquier persona que se encuentre ante ella.

El **index.php**, que es la primera parte que se observa nada más conectarnos, es bastante simple como se puede observar en la siguiente foto:

No se ha buscado unos colores que atraigan al usuario ni nada por el estilo, solo que sea simple, eficaz y ordenada. El logo ha sido elaborado con photoshop a petición de la empresa solicitante y la foto que aparece en grande ha sido buscada en un almacén de fotos online de libre distribución.

Una vez logueados veremos la misma página y se ha optado por un menú de navegación desplegable basado en categorías, las cuales son:

Además disponemos de un panel a la izquierda que muestra apartados relevantes para el usuario el cual permanece igual a lo largo de la sesión y en el mismo sitio:



El **footer** es adaptable a la página. Así se evitan problemas si los datos sobrepasan el tamaño de la página o si se queda corta, poniéndose siempre al final de panel o en su defecto del cuerpo de la página.

Al ser el menú superior y el panel de la izquierda fijos en todas las páginas, el usuario puede acceder a donde quiera en cualquier momento sin verse envuelto en un laberinto de hiperenlaces o algo similar.

7.2. Funciones disponibles para los usuarios:

Se va a dividir entre el apartado Back-end y Front-end:

- **Back-end**: Lo que solo el administrador puede hacer:

- Gestionar los empleados: En este apartado se pueden ver todos los usuarios registrados, así como modificarlos y eliminarlos. Tiene controles de seguridad para no rellenar datos sin sentido y que si se diera el caso de introducir una id de usuario inexistente a la hora de modificarlos, se retorna un error de que no existe.

- Gestionar las claves de registro para nuevos empleados: Esta parte de la web se ha realizado para gestionar dos cosas simultáneamente: La asignación de los privilegios de los nuevos usuarios (**admin** o **user**) y limitar los registros a 1 por persona para evitar cuentas duplicadas y similares. Desde esta sección, mediante los botones, se puede generar una clave que se utilizará a la hora del registro, que si se da el caso de que no coincide se devolverá error y si coincide se establecerá como utilizada y se almacenará en el historial).

- Información sobre los servidores y comprobar la conexión: En esta sección se accede a los paneles de información de servidor. Son independientes de la página principal y son tratados como tales. **Hay 3**: para el servidor web, para el servidor bbdd y para la RouterBerry, cada uno con su sesión independiente. En estas páginas se puede consultar información del servidor (temperatura, espacio, servicios activos, procesos y el archivo phpinfo). Luego aparte de los botones de información de servidor, podemos ver las comprobaciones de conexión. Su utilidad es la de realizar un ping a los distintos servidores o a internet en general, para ver si hay conexión de una forma rápida y eficaz.

- Un historial de las últimas 20 incidencias: Para llevar un control de las incidencias los administradores tienen un historial de las últimas 20 incidencias que han ocurrido.

- Vaciar el chat: botón que al pulsarlo reestablece totalmente el chat, dejándolo por defecto y vacío. Recomendable darle cada semana al menos.

- **Front-end**: Lo que cualquier usuario puede hacer:

- Iniciar sesión: Conectarse a la página web mediante la base de datos para utilizar el contenido disponible solo para los empleados registrados.

- Registrarse como empleado: Se le solicita el empleado a la hora de registrarse una clave, así como una serie de datos. Tiene control de datos y validación mediante jQuery.validate.

- Modificar los datos personales de su cuenta: Apartado que todo empleado conectado tiene donde puede modificar su nombre, apellidos, email o contraseña de forma sencilla y rápida.

- Un espacio de almacenamiento online: Utilizado para para subir, descargar y gestionar los archivos que deseen subir. Este espacio es personal y no se puede tener acceso de otra forma a excepción de un administrador. Cuando un empleado se registra se crea una carpeta para él solo, y si se da de baja dicho empleado, la carpeta se borra con todo su contenido, sin forma de recuperarlo.
- El área de la gestión de incidencias: Desde aquí podemos abrir las incidencias sobre los casos de los clientes, gestionarlas y además disponemos de un buscador para poder comprobar casos antiguos y ver si se puede utilizar la misma solución.
- Una agenda personal para los empleados: El objetivo de ésta es gestionar cualquier tipo de evento. Aunque está pensado especialmente para cosas relacionadas con el trabajo puede utilizarse como guste por parte del empleado. Dentro se puede ver tanto los eventos próximos, como los que van a ocurrir en un futuro y los que ya han pasado, disponiendo de la opción de eliminarlos en cualquier momento.
- Mensajería privada: Si se da el caso de que se quiere mandar un mensaje a cualquier otro empleado, con un sistema de aviso ante mensajes nuevos que se actualiza cada vez que el usuario recarga o cambia de página. Si el usuario entra en la bandeja de entrada, todos los mensajes se dan por leídos. Hay una lista para mostrar los distintos empleados puesto que para mandar un mensaje se requiere conocer la id del destinatario.
- Un chat: entre los empleados para comunicarse en caso de necesidad inmediata. Se recomienda tenerlo abierto en una pestaña para estar siempre al tanto de lo que sucede. Dicho chat tiene un historial para consultar mensajes antiguos puesto que el chat solo muestra los últimos 15 mensajes.

8. Seguridad implementada

8.1. Seguridad Web

A pesar de ser una intranet y por defecto ser bastante segura, no he querido omitir un aspecto tan importante como puede ser la seguridad de los datos. Todas las contraseñas se han almacenado en algún tipo de cifrado para que a la hora de su almacenamiento en la base de datos no suponga ningún problema a largo plazo.

En todos los formularios se ha establecido un control de datos mediante **jQuery.validate** para no utilizar caracteres inadecuados o que sean muy largos.

Además, se ha establecido seguridad en todas las páginas con control de usuario. Si se intenta acceder a cualquier página de la web sin estar con la sesión iniciada, automáticamente se reenviará al invitado a una página donde se le indica que no puede hacerlo y que debe conectarse con permisos más elevados. Esto lo he hecho mediante el uso de condicionales basándome en el valor de $_SESSION que previamente he almacenado cuando el usuario

pasa la fase de comprobación. En dicho valor se almacena su puesto, es decir si es **user** o **admin**.

Mediante una modificación en la configuración de **apache2** en el servidor he anulado el acceso a los directorios de forma manual, saliendo un mensaje de permiso denegado al intentarlo. Así se evita que comprueben las imágenes o archivos que puedan ser comprometedores.

Otra cosa que he tenido en cuenta es el tamaño de los archivos para subir al almacén interno. Aunque el empleado no tengo un tope personal, cada archivo no puede tener un tamaño superior a los 100mb, o dará error al subirlo, ya que el servidor lo denegará con un aviso de error.

Mediante CobianBackup he ido haciendo copias de seguridad de la página web cada hora por si cometía algún tipo de equivocación poder solucionarla fácilmente.

8.2. Seguridad de MySQL (BBDD).

Solamente permite el acceso a dos usuarios, **global** y **root.** En el caso de **root** obviamente se le ha modificado la contraseña por defecto. Global tiene permiso para conectarse desde cualquier ordenador de la intranet, es decir 172.16.111.0 mientras que **root** solo tiene permisos por parte de localhost y 172.16.111.2 (servidor web).

Los privilegios de **root** son más elevados para así evitar problemas.

Así mismo mirando formas de mejorar la seguridad en internet he eliminado algunas bases de datos que se recomiendan por ser inseguras. Como he mencionado antes todos los datos almacenados que sean peligrosos están cifrados en diferentes formatos de hash, ninguno md5 debido a su facilidad de des encriptación dados los métodos actuales. En el caso de la contraseña del login se utiliza el cifrado **BCRYPT** mientras que para los usuarios del ftp se utiliza **crypt**, que junto a md5 son los únicos compatibles, y crypt es de lejos mucho más seguro ya que está basado en **DES.**

A medida que hacía el proyecto he ido generando backups manualmente cada día de todos los datos y tablas por si surgieran problemas mediante MySQL WorkBench

8.3. Seguridad de las RaspBerry

Ambas hay que mantenerlas actualizadas periódicamente. Se han modificado todas las contraseñas por defecto a unas mucho más complejas y en todas hay un usuario alternativo por si surge algún tipo de error.

Por prevención no se ha realizado el uso de keys públicas SSH así que siempre va a solicitar la contraseña a la hora de conectarse.

He estado investigando sobre los virus en Raspbian y son tan pocos casos y tan remotos que no me he planteado en ningún momento el uso de antivirus o algún servicio extra por temas de seguridad.

9. Fase de testeo

Poniendo la página en internet mediante mi IP pública, he pedido a gente cercana que entrara en la página para testear los diferentes servicios que ofrece ésta.

Gracias a eso pude pulir varios aspectos como la gestión de las incidencias, problemas en el chat como por ejemplo que cuando enviabas no te devolvía automáticamente al input para enviar otro mensaje, sino tenías que hacerlo manualmente, la agenda no funcionaba correctamente ya que almacenaba mal en el historial los eventos acabados, y cosas similares.

También me encontraron varios errores cuando introducías datos aleatorios en los campos de algunos formularios, así que mediante el uso de condicionales y otros métodos conseguí anularlos todos con mensajes como “El dato introducido no es viable” o similares.

También se localizaron fallos en el CSS. Yo estaba trabajando sobre un espacio de 1200x800px, pero en pantallas más grandes se desordenaban varias cosas, así que aplique los conocimientos de responsive que tenía para adecuar la página a pantallas más grandes que la mía, solucionando el problema completamente. Si es más pequeña de 1000x700px se descuadra, pero esto es así porque la web está pensada para trabajar solo sobre ordenadores de una dimensión mínima y no se ha observado el caso de móviles o Tablet de dimensiones más pequeñas.

Además, cuando fui a validar el proyecto, Carlos y Jose me recomendaron un par de cambios como poner hipervínculos en el panel de la izquierda que llevaran a los eventos urgentes o a los mensajes nuevos en el caso de que hubiera, así como omitir en el apartado de gestión de incidencias que el usuario tuviera que introducir la id de la incidencia manualmente, poniendo botones en cada una de ellas para acelerar el proceso.

10. Ampliaciones y mejoras posibles

Se pueden realizar varias mejoras sobre el proyecto. Lo más claro y directo es, que cuando saquen nuevas versiones de RaspBerry, cambiarlas para aprovechar la última tecnología y así aumentar los tiempos de acceso y optimizar los resultados. Esto se recomienda especialmente en el caso de la RouterBerry debido a la cantidad de datos que puede llegar a manejar en ciertos momentos si los usuarios empiezan a descargar cosas de forma simultánea.

Si la empresa creciera y quisiese situarse en un lugar concreto, y si ya se tienen unos altos beneficios debería observarse la opción de actualizarse a algo más potente, es decir, servidores dedicados de alta capacidad en el caso de los servidores web y bbdd y un router CISCO o similar de gama media que venga con firewall y un kit de seguridad bueno para aumentar la seguridad de la intranet, así como hacer de las funciones de las que se está ocupando actualmente la RouterBerry.

Realmente con 64GB en ambos servidores puede parecer o dar la sensación de que es una cantidad aceptable para una pequeña empresa, pero llega un momento en el que si el volumen

de clientes crece muy rápido puede verse excedido rápidamente, así que dicho servidor dedicado debería de tener unos espacios de almacenamiento rápidos.

Otra cosa que hay que observar es la página web, puede que la empresa quiera llegando el momento disponer de una red inalámbrica con inhibidores en los exteriores para tener acceso también mediante dispositivos inalámbricos a la intranet. En dicho caso se buscaría hacer la web responsive en ese aspecto ya que en el estado actual es totalmente inviable.

Otra cosa que me habría gustado hacer y no he podido por falta de tiempo es implementar el servidor FTP con algún servicio de grabación para que los empleados graben y suban automáticamente las llamadas a una zona común y que se almacenasen por fecha y hora, para así tener un registro en voz de lo que ha ocurrido en esos casos y puede ser algo realmente útil más tarde.

También con unos conocimientos de AJAX y jQuery superiores, habría sido buena idea poner notificaciones en vivo para cuando los usuarios recibieran un mensaje o cuando alguien hablara por el chat, siendo así una web mucho más dinámica.

En la agenda habría estado bien buscar alguna librería que ayudase a generar un calendario y marcar con colores los días en los que se tuviera algún evento, y de otro color los urgentes, en vez de tener solamente el formato de tablas.

11. Conclusiones

Me ha gustado bastante realizar este proyecto. Gracias a él he conseguido aprender bastante de desarrollo web, especialmente de jQuery y de AJAX. Además, debido a que he estado utilizando muchísimo PHP los exámenes a lo largo del curso relacionados me han dado mejores resultados debido a que he tenido que ir siempre un paso por delante.

Si bien es cierto que me habría gustado desarrollar algunos apartados más, debido al tiempo y a que tenía que prestar atención a las distintas asignaturas, no he podido realizar algunas ampliaciones que me habrían gustado como indico en el punto [**10**].

Más que un trabajo obligatorio, me lo he tomado como un hobby y he llegado a estar algunos días más de 6 horas seguidas dándole caña debido a que el tiempo se me pasaba volando, ya fuera investigando a utilizando lo que había aprendido, pero siempre había algo que hacer. En internet hay muchísima información de todo y solo hay que saber dónde buscar y luego aplicarlo al proyecto.

Si tuviera que remarcar algo, lo que más me ha impresionado de lo que he hecho ha sido el apartado visual de la web. Siempre he pensado que no sería capaz llegando el momento, pero al final que conseguido hacer un template basado en CSS simple, aunque con sentido y fácil de entender para los usuarios.

Si tuviera que dejar en constancia lo que peor he llevado, ha sido el tema del chat, aunque al final conseguí terminarlo a tiempo, me parece que me ha quedado un poco siempre, pero al menos realiza las funciones que yo buscaba en un principio.

En resumen, estoy bastante contento con lo que ha resultado el trabajo y no me arrepiento para nada del tiempo invertido, incluso no descarto retomarlo en algún día para implementarles las cosas que no fui capaz en su momento por falta de tiempo y conocimientos.

12. Bibliografía

- http://librosweb.es/libro/fundamentos_jquery/ **Manual de jQuery**

Casi todos los conocimientos relacionados con jQuery los he sacado de aquí. Además, a la hora de hacer el chat, me he basado en varios videos de Youtube en los que explicaban como administrar correctamente el jQuery.

- http://librosweb.es/libro/ajax/ **Manual de AJAX**

Para saber hacer el chat necesitaba ciertos conocimientos de AJAX, y gracias a ciertos apartados de este manual entendí en gran parte lo que estaba utilizando y de qué forma, para posteriormente guiarme mejor por las guías de cómo hacerlo.

- https://stackoverflow.com/ **Stack Overflow**

Muchas dudas que buscaba en ingles están resueltas en este maravilloso foro de programadores, si tuviera que hacer hincapié en el apartado de FTP y en la búsqueda para solucionar errores de mi chat todas las respuestas se encontraban en dicha página.

- https://www.ochobitshacenunbyte.com/2014/10/13/como-instalar-proftpd-con-soporte-mysql/

Gracias a la persona que posteó este artículo entendí como implementar proftpd con MySQL. Aunque yo lo tuve que adaptar a mi proyecto y variar bastantes cosas, me sirvió como base para poder desarrollar este apartado.

- http://php.net **PHP.net**

Muchas de las funciones que estaba buscando se encontraban dentro de esta página, además en los comentarios la gente sube ejemplos que me servían de inspiración para aplicar yo luego en la página web.

- https://github.com/gumslone/GumCP **C.Panel**

Gracias a este proyecto open source que ésta persona subió a GitHUB, pude encontrar una forma de manejar la información de los servidores. Lo adapté a la página de la empresa para que englobara los campos necesarios.

Y para finalizar, como no, todo el material didáctico dado por los profesores, así como los tomados por mí mismo en lo referente a PHP, MySQL, Javascript, CSS, Linux, Redes, Servicios y Hardware.