Documentación General

Manual de Cliente

Aergibide S.L.

Realizada por:

® We(App);

Web & Application Developers
By Tania, Mario & Adrián

García, Tania · Pisabarro, Adrián · Zatón, Adrián
2º DAW

Índice

Análisis de la Idea, Introducción

Reto

Desde Aergibide SL nos piden a la cooperativa WeApp la realización de un foro o aplicación como intranet de la empresa.

- **Aplicación interna o Intranet de tipo Foro.**
  Está desarrollada con las tecnologías de HTML5, CSS3, JS y PHP7.
  Es una aplicación que se basa en la gestión del contenido en la que la utilizarán 150 ingenieros aproximadamente. Para poder ayudarles en su trabajo autónomo hemos desarrollado un portal con las siguientes aplicaciones prácticas ya en pre-producción:
  - Registro/Login
  - Gestión de cuentas
  - Categorías
  - Subida de preguntas y respuestas
  - Subida de archivos en preguntas y respuestas (gestión de archivos)
  - Borrar y modificar preguntas
  - Borrar respuestas
  - Filtrado de preguntas
  - Me gusta (like) de preguntas y respuestas
  - Mis preguntas favoritas
  - Mis preguntas (o historial de mis propias preguntas)
  - Puntos obtenidos y nivel según el número de preguntas y respuestas hechas.
  - Montar un servidor basado en scripts de Apache, MySQL, PHP, FTP y SSH.

WeApp Cooperativa

WeApp ha aceptado el segundo reto para la empresa Aergibide SL, esta empresa está formada por los siguientes integrantes:

- Tania García: https://github.com/TaniaGarciaOlarte
  Funciones: Planificadora y Reportera. Socia
- Adrián Pisabarro: https://github.com/adrianpisabarrogarcia
  Función: Coordinador. Socio
- Mario Zatón: https://github.com/mariozaton01
  Funciones: Responsable de material y Armonizador. Socio.

La empresa WeApp de IkasEmpresa se centra en el desarrollo web, realizando así aplicaciones específicas para empresas como Aergibide. Realizamos mantenimiento de páginas webs, montamos servidores para webs junto con base de datos, páginas webs estáticas, dinámicas y aplicaciones web, entre otros servicios.

“WeApp what you demand”

Objetivos del reto

Este reto nos ha ayudado a practicar diferentes competencias que en una empresa real nos pueden pedir. Ha sido una oportunidad para poner en práctica aquello que hemos aprendido.

La conclusión de nuestro trabajo durante este reto ha sido un objetivo y es el aprendizaje, no todos teníamos el mismo nivel después de haber realizado las actividades previas al reto; sin embargo, ha tenido que ser una constante ayuda entre miembros del equipo para poder sacar el reto adelante, poniendo cada uno su granito de conocimiento a aquel que no se le daba bien o tenía problemas desarrollando sus tareas.

Podemos destacar que hemos trabajado de forma cooperativa y con buen ambiente de trabajo entre nosotros.

En cuanto a los objetivos técnicos creemos que también los hemos cumplido, a pesar de que hemos llegado justos partiendo de que somos 3 personas en comparación con los grupos de 4 íbamos retrasados.

Hemos dividido el reto en diferentes funcionalidades que nos hemos repartido entre las 3 personas. Todos hemos realizado scripts, css, html, js y php. (véase la página 3 para poder ver las funcionalidades que tiene nuestro reto)

Creemos firmemente que se notará a la hora de estudiar para los exámenes y mejor aún, nos sentimos muy alegres de saber cómo manejar las diferentes tecnologías (bash, php, js, …)

Por ello, nos sentimos orgullosos de nuestro proyecto, a pesar de que nos hubiera gustado estar menos agobiados o haber hecho hincapié en errores que no se tenían que haber cometido en el desarrollo de la aplicación, cosa que nos ha retrasado el desarrollo.

Planificación y organización

Antes de comenzar

Antes de empezar el reto, nos hemos organizado por funcionalidades tal y como nos explicó Jon pusimos en una hoja todas las funcionalidades que íbamos a hacer de esa manera comenzamos los días con una reunión diaria y ponernos a trabajar directamente.

Hemos realizado reuniones todos los días del reto por la mañana a primera hora respondiendo a las siguientes preguntas siempre:

- ¿Qué tal vamos?
- ¿Qué estamos haciendo bien? ¿Qué estamos haciendo mal?
- ¿Deberíamos mejorar algo como grupo?
- Tareas para hoy

- ¿Algún problema en el desarrollo del día anterior? para poder ayudar a esa persona que ha tenido un problema. Esto se ha dado bastante, con lo cual algún integrante del grupo tiene que ayudar a otro. Lo mismo cuando una persona no sabe hacer algo, se le asigna otra persona que pueda en ese momento, le explica para que aprenda.

Dentro de la aplicación “Trello” nos hemos planificado y visto que las funcionalidades cada día se iban completando.

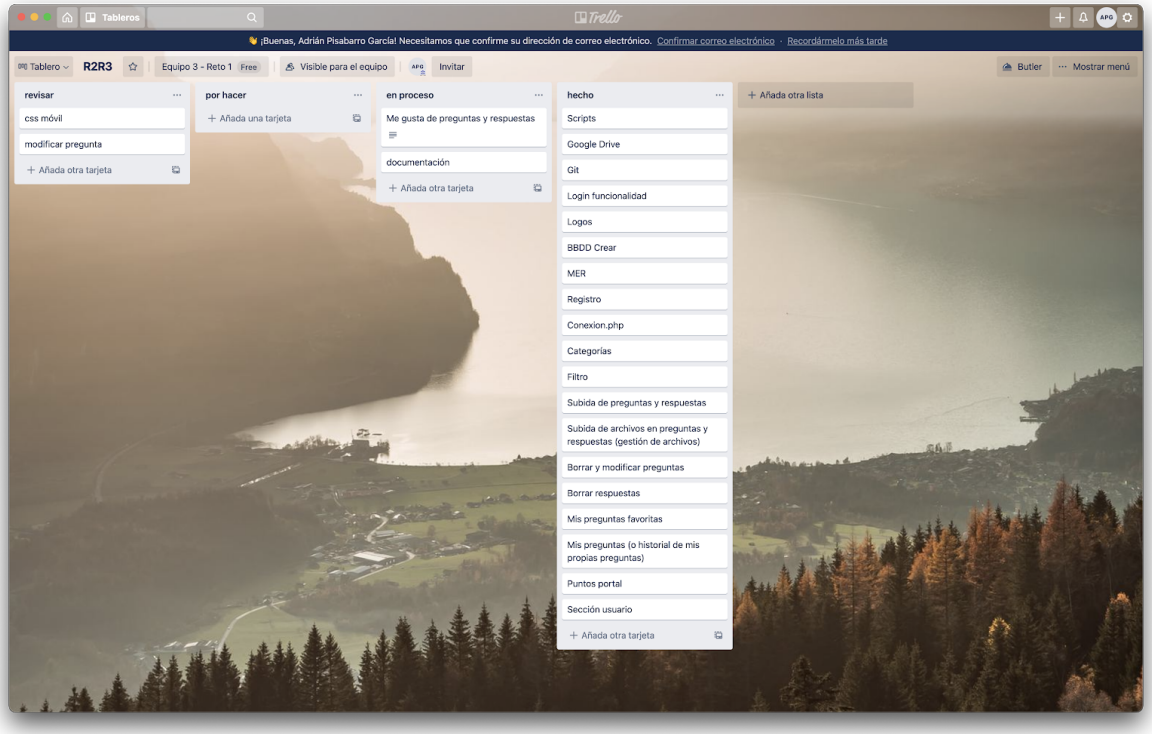


Diagrama de Gantt

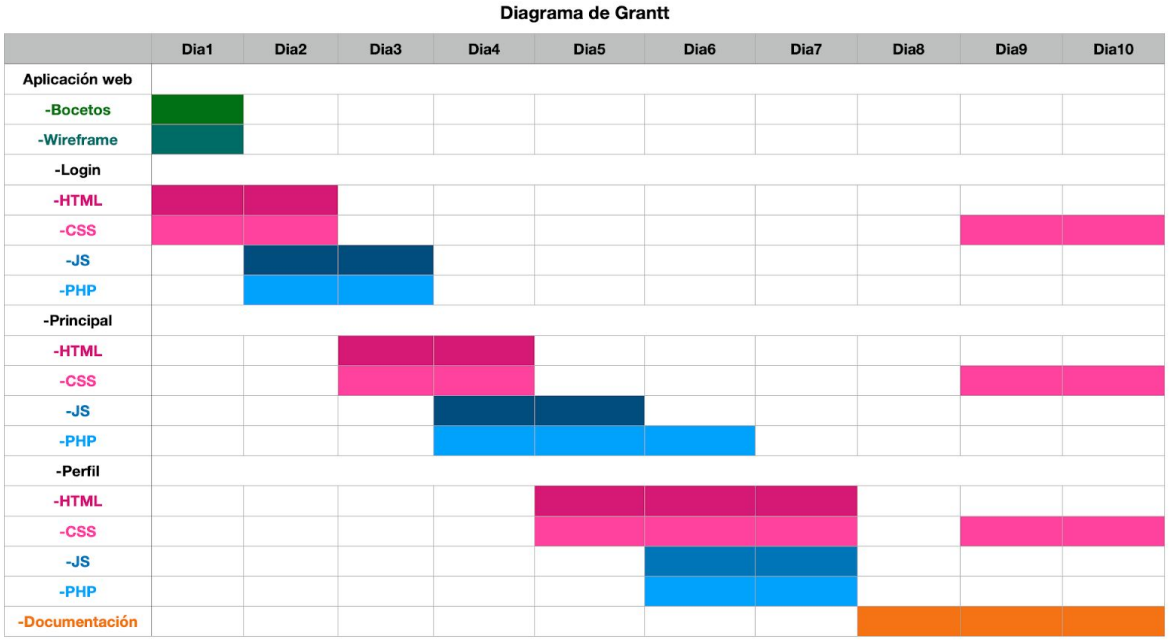


Diagrama de Gantt para planificarnos

El diagrama muestra el el tiempo que hemos tardado en hacer las diferentes funcionalidades; sin embargo, somos conscientes de que ciertas funcionalidades estamos actualmente refinándolas para poder presentar un mejor reto la próxima semana.

Herramientas utilizadas

Estas han sido algunas de las herramientas que hemos utilizado, sin embargo, pasamos a mencionar cuáles han sido.

- Trello
  Nos ha ayudado a visualizar las tareas y a ir completando nuestro tablero kanban

- GitHub
  Hemos seguido el mismo modelo de desarrollo de Git que en el anterior reto.
  Visualizar aquí:
  https://drive.google.com/file/d/1moQMAYIelT9nnNBsAmhj6PlOFs80t9ZE/view?usp=sharing

- JetBrains: PhpStorm y WebStorm
  Han sido los principales IDEs para poder trabajar con la aplicación. Hemos tenido que añadir un archivo .gitignore a nuestro repo para poder excluir los archivos ocultos que generan estas aplicaciones, sin embargo, estamos encantados con estos editores de código.

- Notion y Todoist
  Como bloc de notas para el reto organización de tareas con calendarización.

- Google Suit for Education
  Proporcionado por Egibide. Herramienta para compartir archivos y editarlos online, como este documento.

- Ubuntu Desktop, Server, Vagrant y VirtualBox
  Para la realización del servidor.

y otras herramientas más...

Aplicación web

Guía de Estilo

Nuestra guía de estilo es la siguiente, se encuentra en el ANEXO 1 de la documentación entregada:

https://drive.google.com/file/d/1QCp8ImupGROg34Ga6ZK8YHDxnQyl6U4i/view?usp=sharing

Scripts para Servidor

Para automatizar la instalación y configuración de los servicios necesarios para hacer un LAMP hemos creado los siguientes scripts:

- **configurarip.sh** → Sirve para configurar la ip de nuestro servidor en caso de que sea un ordenador que esté red de clase ya que en caso de que sea en un portátil si no estamos en la misma red nos daría problemas a la hora de conectarnos a nuestro servidor
- **apache.sh** → Sirve para realizar la instalación y configuracion del servicio de apache en el cual realizamos todo tipo de comprobaciones y copiamos los archivos de nuestra app web

- **ftp.sh** → Sirve para realizar la instalacion y configuracion del servicio de ftp y así poder acceder mediante la transferencia de archivos poder modificar nuestra app web más cómodamente
- **mysql.sh** → Sirve para realizar la instalación y configuración de nuestra base de datos
- **ssh.sh** → Sirve para realizar la instalacion y configuración del servicio de ssh para poder conectarnos remotamente a dicho servidor

Servidor LAMP

Para montar el servidor LAMP hemos utilizado vagrant y para ello hemos generado un VagrantFile en el cual indicamos lo siguiente:

- El sistema operativo que va a tener esa maquina
- La configuración de la tarjeta de red que en este caso va a ser una pública para poder conectarnos
- La carpeta compartida que vamos a tener con nuestro equipo, en nuestro caso las cosas que están en la carpeta de la máquina se guardarán en el apartado de carpeta-compartida del servidor la cual estará en la raíz del servidor

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/bionic64"
  config.vm.network :public_network
  config.vm.synced_folder "./", "/carpeta-compartida"
end
```

Para crear la máquina deberemos ir al terminal y situarnos en la carpeta donde tenemos el vagrantfile y hacer un **vagrant up**, en el caso de utilicemos un ordenador que no sea de la red de clase, una de las opciones que se puede utilizar es la “1” la cual nos dará una ip del rango de nuestra wifi, pero en nuestro caso no nos va a salir el apartado de la imagen inferior ya que utilizaremos un equipo que está conectado a una red concreta (clase)

```
==> default: Setting the name of the VM: MaquinaScripts_default_1606418193855_10
622
==> default: Clearing any previously set network interfaces...
==> default: Available bridged network interfaces:
1) en0: Wi-Fi (Wireless)
2) en3: USB Ethernet(?)
3) en1: Thunderbolt 1
4) en2: Thunderbolt 2
5) bridge0
6) awdl0
7) llw0
==> default: When choosing an interface, it is usually the one that is
==> default: being used to connect to the internet.
==> default:
    default: Which interface should the network bridge to? 1
```

Montando la máquina en un portátil u ordenador que no es de la red

Montando la máquina en un un ordenador de la red

Al terminar de montar la máquina nos situaremos en la carpeta compartida y en caso de que el usuario utilice un ordenador que no está en la red el primer script que deberá ejecutar es el de **apache.sh**,en caso contrario, el primer script que se deberá ejecutar es el de **configurarip.sh** para poder asignar una ip de dicha red.
La ip de nuestro servidor en este caso será la de 172.20.224.204.
En nuestro caso como vamos a utilizar un equipo que está en la red el primer script que ejecutaremos será el de **configurarip.sh**

***Importante: Antes de ejecutar el script haremos sudo su para tener privilegios de administrador y no tener ningún problema a la hora de modificar archivos entre otras cosas**

Al finalizar la ejecución del script anterior *(configurarip.sh)*, automáticamente se asignan los permisos de ejecución al siguiente script y este se ejecutará automáticamente y así sucesivamente hasta terminar. El orden de ejecución de los scripts es:

1. configurarip.sh
2. apache.sh
3. ftp.sh
4. mysql.sh
5. ssh.sh

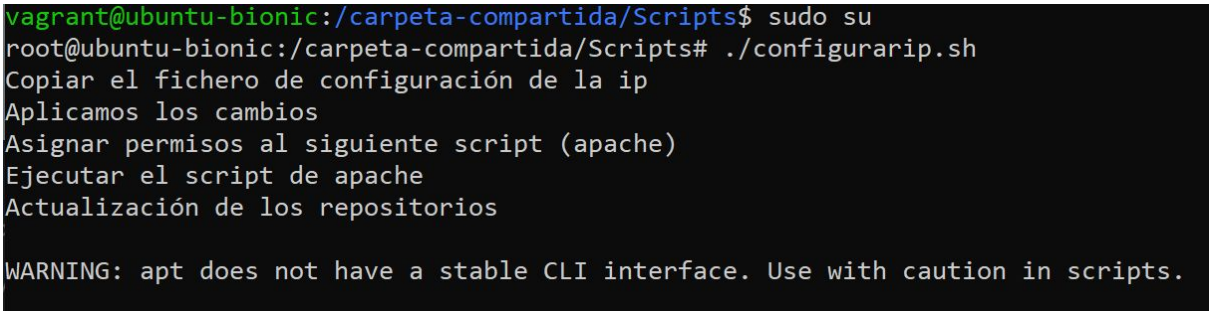


Imagen de la ejecución de los scripts

Al terminar de ejecutarse todos los scripts nos podremos conectar al servidor mediante el servicio de FTP y el servicio de SSH para ello podemos utilizar los siguientes detalles:

- FTP
  - **Observación: El usuario ftp está enjaulado en el directorio /var/www/html/aergibide-develop/ para que solo pueda hacer modificaciones de la app web y no de los demás datos del servidor**
  - **Usuario:** equipo3
  - **Contraseña:** 12345Abcde
  - **Puerto:** 21

- SSH
    - **Usuario:** sshequipo3
    - **Contraseña:** 12345Abcde
    - **Puerto:** 22

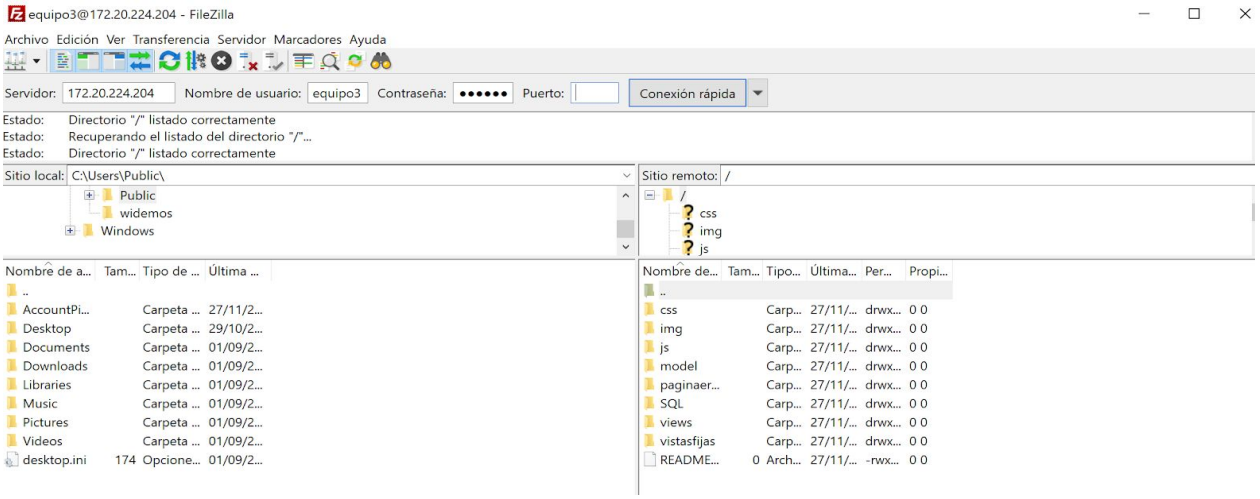


Imagen conexión al servidor mediante ftp (herramienta gráfica)

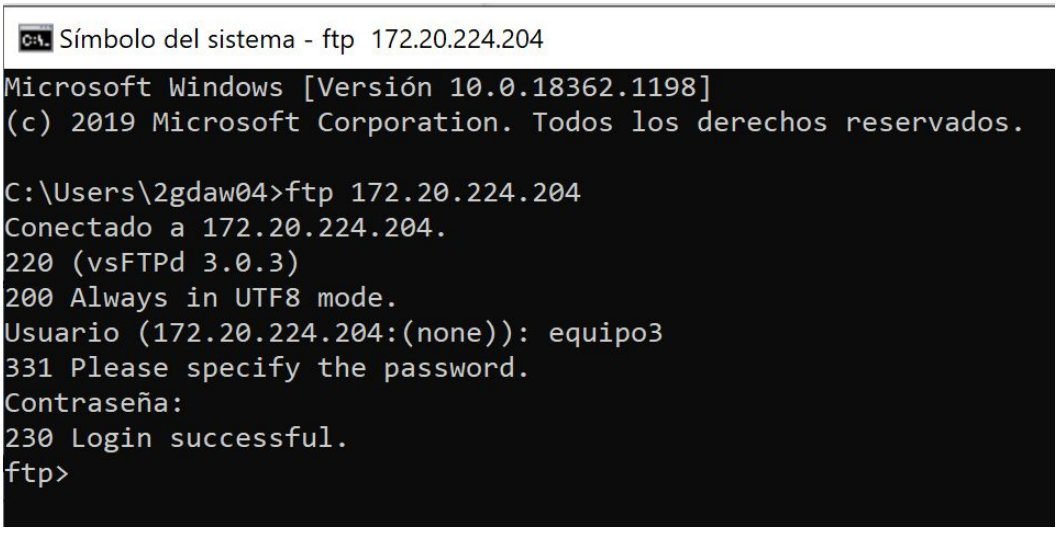


Imagen conexión al servidor mediante ftp (terminal)

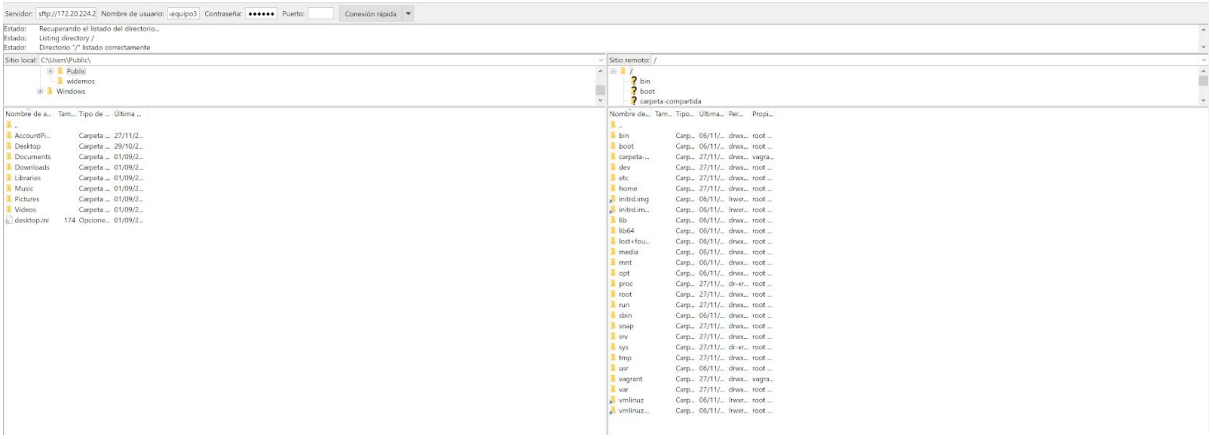


Imagen conexión al servidor mediante ssh (herramienta gráfica)

```
C:\Users\2gdaw04>ssh -p 22 sshequipo3@172.20.224.204
The authenticity of host '172.20.224.204 (172.20.224.204)' can't be established.
ECDSA key fingerprint is SHA256:n24ecb21qaDoyVCl+amnoWxdUy+bq7Qh4rYS3vRppH4.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.20.224.204' (ECDSA) to the list of known hosts.
sshequipo3@172.20.224.204's password:
Welcome to Ubuntu 18.04.5 LTS (GNU/Linux 4.15.0-122-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri Nov 27 08:17:21 UTC 2020

  System load:  0.0               Processes:             116
  Usage of /:   16.3% of 9.63GB   Users logged in:       1
  Memory usage: 36%               IP address for enp0s3: 10.0.2.15
  Swap usage:   0%                IP address for enp0s8: 172.20.224.204


29 packages can be updated.
14 updates are security updates.

New release '20.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

Imagen conexión al servidor mediante ssh (terminal)

Para poder ver nuestra app web introduciremos la **ipservidor/model/login.php** y en caso de que metamos una página web errónea nos mostrará la página de error que hemos generado con el script

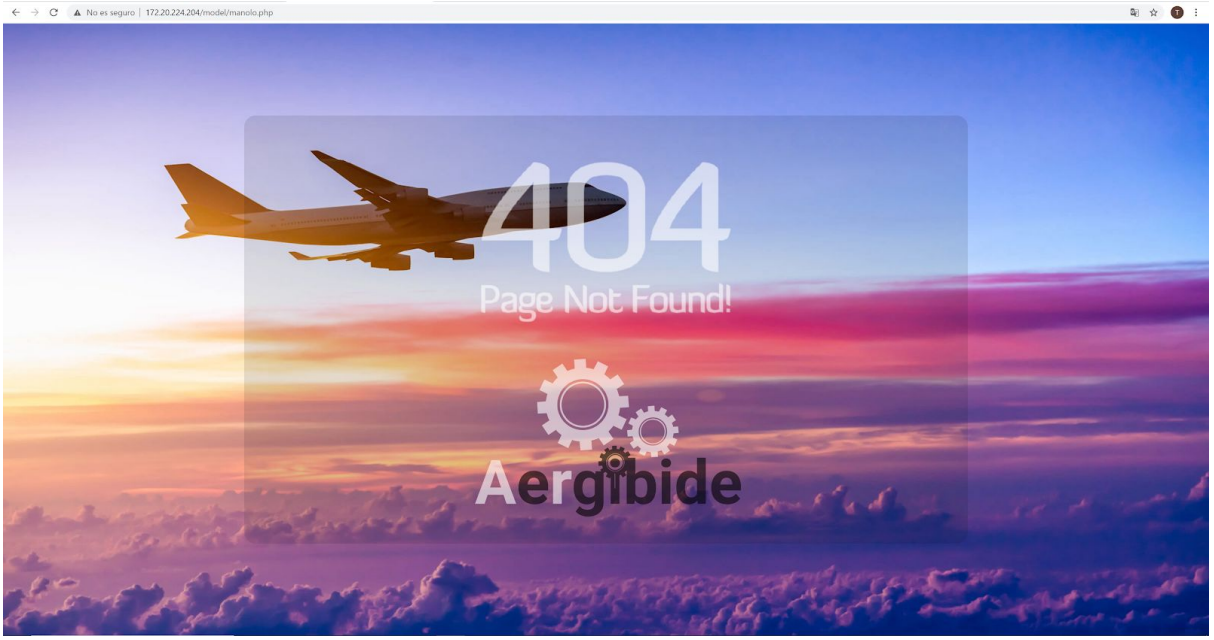
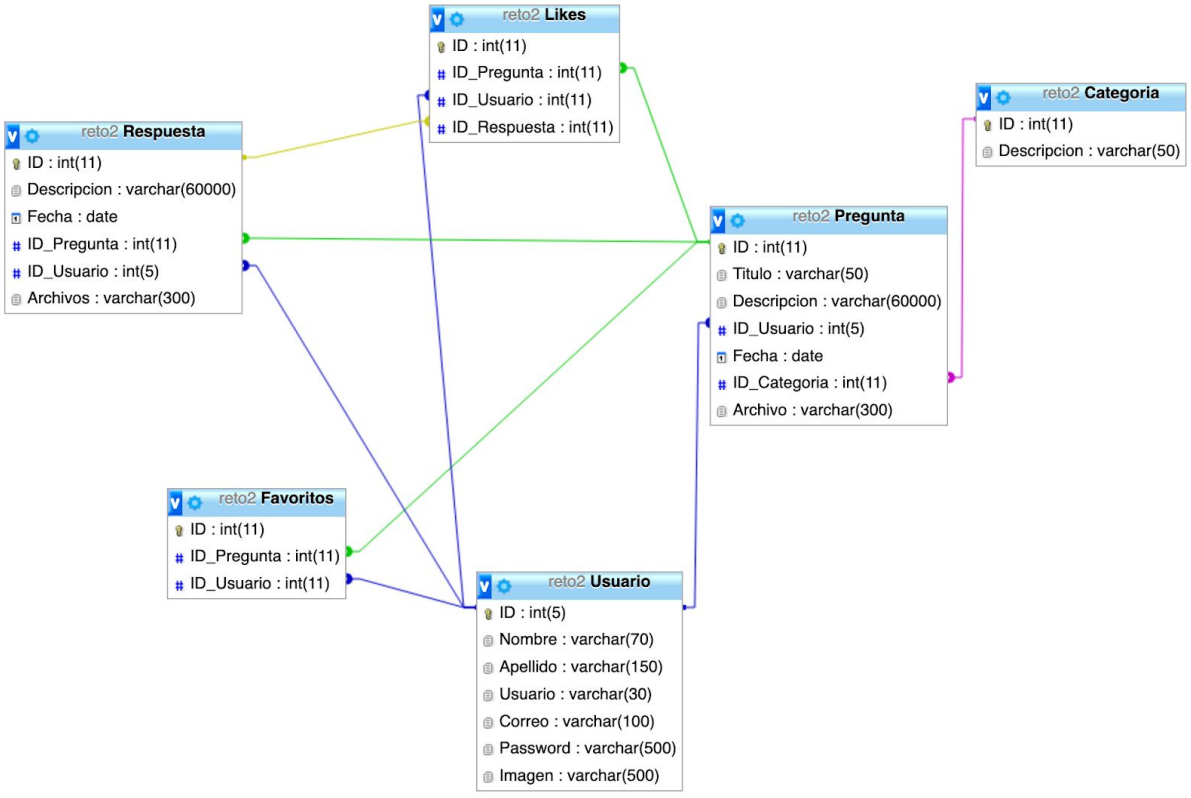


Imagen de la página de error

Imagen del login

Modelo de BBDD MySQL

El esquema modelo-entidad-relación de la base de datos, con su correspondientes foreign keys y claves primarias, es el siguiente:



Explicaciones importantes de las tablas:

- **Likes**: es una tabla que se utiliza para almacenar el like de una pregunta o de una respuesta.
- **Favorito**: únicamente podrás marcar como favorita una pregunta, nunca una respuesta para que el propio usuario vaya al menú y haga clic en “Mis favoritos” y le muestre sus publicaciones favoritas, accedes más rápidamente a ciertas preguntas que quiera marcar.
- **Pregunta**: es una tabla en la que se guarda todo el contenido de una pregunta y que hace referencia al usuario que la ha escrito y a la categoría a la que pertenece. Hemos decidido no guardar la hora de cada publicación o respuesta, sólo la fecha. En un desarrollo más profundo o próxima versión, podría ser algo a tener en cuenta. La pregunta puede ser borrada por el creador, así como la posibilidad de editarla. Tanto las preguntas como las respuestas pueden guardar un archivo por cada, no es un campo obligatorio.
- **Respuesta**: hace referencia a una pregunta (N:1) para poder visualizar todas las respuestas de una pregunta. El propio usuario dueño de la pregunta puede eliminar respuestas, así como el usuario que ha creado la respuesta.
- **Categoría**: todas las preguntas deben de tener una categoría ya insertada en base de datos. En caso de querer añadir una categoría más, lo tendría que hacer el administrador de la bbdd creando una fila y automáticamente se crearían los botones y los radiobuttons de las categorías en las respectivas páginas.
- **Usuario**: cada pregunta y respuesta pertenece a un usuario concreto y cada ingeniero se tiene que registrar en la página con su propia cuenta, desde el login. En ningún caso podrás entrar al portal si no está la sesión en servidor activada. Atención: se puede iniciar sesión con un usuario o con un correo electrónico.

Datos de acceso a la bbdd de pre-producción:

```
$host = "127.0.0.1"
$dbname = "reto2"
$user = "root"
$pass = ""
```

Código JavaScript

Utilización

En general JS lo hemos utilizado siempre que no queríamos sobrecargar al servidor, es preferible que se carguen todas las funciones que se puedan dentro del cliente. Tales como:

- Llamadas ajax para evitar volver a cargar toda la página, a pesar de que en alguna ocasión hemos utilizado un location.reload. Como parámetros principalmente les hemos pasado *valores* o *actions* para que php interprete que es lo que tiene que hacer.
- Hemos utilizado expresiones regulares para evitar que se comprobaran los datos en el servidor.
- Hemos utilizado DOM para poder crear botones simplemente visualizar algo que dando clic a un botón se muestre con el método css de JQuery.

- La utilización de JQuery para desarrollar acciones recogida de valores y otros muchos métodos que si utilizaramos JS puro tardaríamos más tiempo.

En este reto no hemos utilizando un modelo como el ApiFetch o un Local Storage al tener un modelo de datos a través de MySql.

Expresiones regulares o comentarios del acceso a datos

Usuarios

- Usuario, contraseña: caracteres alfanuméricos latinos; de 5 a 20 caracteres.
- Correo electrónico.
- Nombre, apellidos: caracteres del alfabeto latinos y blancos.
- Contraseña: caracteres alfanuméricos sin blancos y un mínimo de 8 caracteres. En la BBDD se guarda un HASH para evitar que si piratean la base de datos puedan acceder.

Preguntas y Respuestas

- Título: caracteres del alfabeto y blancos
- Fecha: (automática en PHP a la hora de guardar, formato YYYY-mm-dd)
- Descripción: cualquier carácter comprendido con una descripción de entre 10 a 60.000 caracteres.
- Archivo: sólo se puede insertar un archivo en el caso de que un usuario quiera almacenar más de un archivo en el servidor tiene que comprimir los archivos. En un posterior desarrollo se podría

Otros elementos a tener en cuenta

A modo de referencia hemos utilizado mayoritariamente estas páginas webs https://developer.mozilla.org/es/ y https://es.stackoverflow.com para dudas que anteriormente a otros usuarios les habían surgido.

Hemos añadido dos **librerías** JS: el famoso editor de texto TinyMCE (https://www.tiny.cloud) y Splide (https://splidejs.com) . Lógicamente para poder utilizarlos hemos entendido su funcionamiento interno investigando.

Código PHP

Utilización

Para él contenido dinámico que proviene de la base de datos hemos utilizado PHP haciendo la diferencia entre las “views”, vistas que imprimen el contenido y la capa lógica que se puede encontrar dentro de la carpeta “model” que esta realiza toda la lógica y llama a “conexion.php” para poder traer de vuelta datos, borrar, insertar o modificar. Además, hemos añadido algunas vistas fijas que se encuentran en la carpeta “vistasfijas”.

Tenemos en cuenta que en algunas llamadas ajax al servidor hemos hecho un location.reload para que cargara toda la página del servidor, lo adecuado sería hacer un append o trabajar aún más de lo que hemos trabajado con DOM en la función :success de javascript. También tenemos que ser conscientes de que somos 3 personas y el tiempo es que es, siempre hay cosas que mejorar.

CSS y HTML

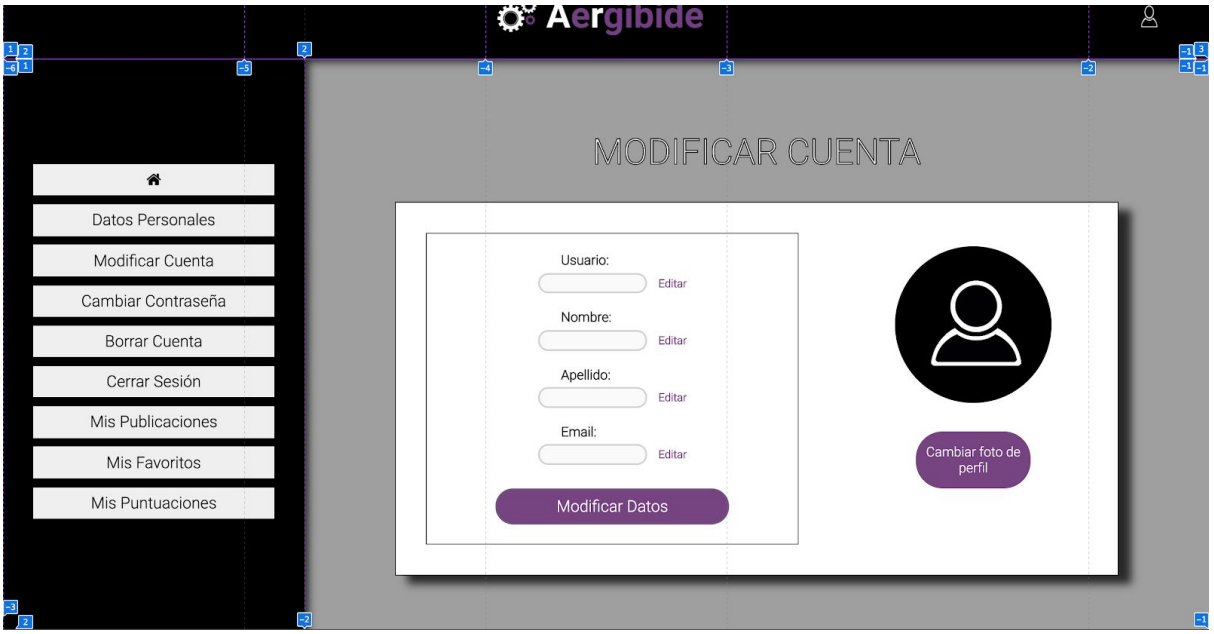
Elementos CSS3

La utilización de elementos CSS3 tales como vistas de móviles y tablets o elementos tan simples como sombras entre otros muchos elementos que corresponden a la versión nº3 de CSS.

Nuestro portal está adaptado a móvil y tablet. Además hemos tratado también de que en la orientación: landscape también se visualizará correctamente. Diseño “responsive”. Cabe a destacar que

Creemos que es una aplicación sencilla, visual y adaptada a la accesibilidad de cualquier usuario, siempre hemos pensado en nuestros padres, que no tienen mucha idea, al fin y al cabo serían los usuarios modelo que se podrían encontrar la aplicación en su trabajo.

Grid



Además, hemos utilizado una mezcla entre Flex y Grid, siendo este último lo que hemos utilizado mayoritariamente ya que lo habíamos dado en clase y había que ponerlo en práctica.

Ideas que si hubiéramos tenido tiempo nos hubiera gustado implementar

Creemos que hemos cumplido con las funcionalidades que nos pedían en el reto.

Propuestas para mejorar en un futuro de cara a la producción:

- Más filtros en publicaciones.
- La utilización de cookies (esto quizás nos dé tiempo a implementar este fin de semana)
- Subida de más de un archivo.
- Diferentes roles de usuarios.
- Mobile first para evitar mayores cambios al pasarlo a tablet o desktop.
- Utilizar más la función success de ajax para evitar que cargue toda la información desde servidor, dar aún menos carga al servidor.

Conclusión

Hemos sido conscientes de que en este reto hemos necesitado ayuda de nuestros compañeros de equipo, mucho más que en el primer reto para que todos entendamos bien los conceptos y hayamos podido llegar a lo que hemos hecho, también contando que somos 3 personas y hemos ido retrasados comparando los grupos de 4 personas. En general, un reto más, estamos contentos de formar el grupo que formamos.

Bibliografía

Agradecemos toda documentación y ayuda que nos ha prestado:

- Apuntes de Ikas: https://ikas.egibide.org
- Ayuda de diferentes profesores que han probado la aplicación
- https://stackoverflow.com
- https://developer.mozilla.org/es/
- https://www.w3schools.com
- https://www.php.net/
- https://jquery.com
- Apuntes de Jaime y Jon de CSS del curso pasado
- Apuntes de Jon de PHP
- Ejercicios resueltos de Nieves de JS

Pre-Producción

Te dejamos los enlaces relevantes de este proyecto para echarle un vistazo:

- **Github**: https://github.com/adrianpisabarrogarcia/aergibide
  Dispondrás de la rama master con todo el proyecto el mismo día de la presentación, ahora ahora estamos utilizando la rama develop.

- **Scripts** para la asignatura Despligue de Aplicaciones Web para montar el servidor: https://drive.google.com/drive/folders/1bk1MRx_Kr40bksTwK0jtTYdT_c0xIiFd?usp=sharing

- **Aplicación web**: 172.20.224.204 dentro de una de las IPs de la red que tenemos en clase. Hemos querido que podáis ver desde casa o desde cualquier lugar el portal, entonces una hora previa a la presentación, el martes, tendréis subida la aplicación terminada tal y como hemos expuesto en este documento en este enlace público: https://adrianpisabarro.com/aergibide