



MEMORIA FINAL DE PROYECTO

PicSite



CICLO FORMATIVO DE GRADO SUPERIOR

Desarrollo de Aplicaciones Web

AUTOR

Eduardo Jesús Otayza Rojas

TUTOR / COORDINADOR

Manuel Retamosa Tejero

CURSO

2020/2021

I.E.S. CLARA DEL REY

Índice general

1. Título
2. Introducción
3. Alcance de proyecto y análisis previo
 - 3.1. Objetivos
 - 3.2. Requisitos
 - 3.3. Planificación previa
4. Estudio de viabilidad
5. Análisis de requisitos
 - 5.1. Pliego de requisitos
 - 5.2. Planificación previa
6. Diseño de la solución escogida
7. Implementación del diseño
8. Pruebas
9. Conclusión
 - 9.1. Valoración personal
 - 9.2. Objetivos alcanzados

Índice de ilustraciones

- 2.1. Logotipo de StigaCX: <https://stigacx.com>
- 3.2.1. Logotipos de lenguajes de programación: <https://www.cursosgis.com/como-integramos-los-lenguajes-html-css-y-javascript/>
- 3.2.2. Logotipo de XAMPP: <https://interpolados.wordpress.com/2017/06/27/xampp/>
- 3.2.3. Logotipo de AwardSpace: <https://www.awardspace.com/>
- 3.3.1. Logotipo de PLSQL: <https://www.oracle.com/database/technologies/appdev/plsql.html>
- 4.1. Logo de Symfony: <https://es.m.wikipedia.org/wiki/Archivo:Symfony2.svg>
- 4.2. Logo de PHP: <https://es.m.wikipedia.org/wiki/Archivo:PHP-logo.svg>
- 4.3. Logo AJAX: <https://www.vcomputadoras.com/usa-ajax-para-refrescar-un-tag-div/>
- 4.4. Captura de AwardSpace: <https://www.awardspace.com/>
- 4.5. Logo de Imgur:
<https://tabletzona.es/2016/02/27/sin-nada-que-hacer-comparte-contenidos-con-imgur/>
- 4.6. Logo de Pixabay: <https://www.martechforum.com/herramienta/pixabay/>
- 5.1.1. Captura de phpmyadmin desde XAMPP
- 6.1. Logotipo de 000webhost: <https://es.000webhost.com/>
- 6.2. Captura de pantalla de sesión iniciada de AwardSpace:
<https://www.awardspace.com/>
- 6.3. Captura de la API de Pixabay: <https://pixabay.com/api/docs/>
- 6.4. Página de inicio de Pixabay: <https://pixabay.com/es/>
- 6.5. Captura de API de Imgur:
<https://apidocs.imgur.com/#de179b6a-3eda-4406-a8d7-1fb06c17cb9c>
- 6.6. Captura de las tablas utilizadas para mi proyecto.
- 6.7. Logo de Font Awesome: <https://fontawesome.com/>
- 7.1. Index de PicSite “Hello World”
- 7.2. Formulario de inicio de sesión
- 7.3. Formulario de registro

- 7.4. Formulario de registro con validaciones
- 7.5. Página de sesión iniciada de PicSite
- 7.6. Búsqueda de imágenes de “mascotas”
- 7.7. Captura de imagen de la API siendo comentada
- 7.8. Formulario de subida de nueva imagen
- 7.9. Previsualización de subida de imagen.
- 7.10. Comentando la imagen subida.

1. Título y enlace a aplicación

PicSite – Observa tu Mundo, comparte el tuyo.

<http://picsite.atwebpages.com/>

2. Introducción

Mi objetivo es desarrollar una aplicación web de tipo red social, que, aunque no está orientada a mensajería, tiene como objetivo compartir y comentar imágenes de nuestra realidad.

Considero que una red social es, hoy en día, un tipo de aplicación muy completa y popular que me obligará a usar una gran parte de los conocimientos adquiridos en estos dos años como estudiante de Desarrollo de Aplicaciones Web.

Las redes sociales tienen como objetivo el ocio, conque, para ello, tiene que ser dinámica y rápida con su respuesta.

Esta aplicación será desarrollada con mis conocimientos adquiridos en este tiempo de la mano de mis profesores que tanto me han aportado y sacando a relucir la experiencia en la resolución de problemas con algoritmos por mi parte y soluciones diversas por parte de la web.

También comentar, que estos dos últimos meses con mi empresa de prácticas, Stiga, he podido obtener mayor visión a la hora de hacer webs más dinámicas.

stiga

2.1. Logotipo de StigaCX: <https://stigacx.com>

3. Alcance proyecto y análisis previo

3.1. Objetivos:

El objetivo de mi proyecto es crear una red social dinámica, sin redirecciones, visual e interactiva con el usuario. Para ello, el usuario podrá navegar con rapidez sin usar el famoso botón “retroceder” del navegador y enfocando más su atención al contenido de la aplicación.

3.2. Requisitos:

Para ello usaré JavaScript, PHP, PLSQL, CSS y HTML para lograr mi cometido. Para el intercambio cliente servidor, usaré AJAX en su forma más básica para obtener o registrar información con la mayor rapidez posible de diferentes APIs y de mi propio servidor gratuito.



3.2.1. Logotipos de lenguajes de programación: <https://www.cursosgis.com/como-integramos-los-lenguajes-html-css-y-javascript/>

Para cumplir mi propósito crearé bases de datos. Una para usuarios, otra para imágenes (o referencias a imágenes) y una última para los comentarios. Todas ellas tendrán un identificador para conectarlas y poder realizar consultas de manera más efectiva.

Como todos sabemos, una página web es una carpeta de ficheros. Por eso, intentaré reducir la cantidad de archivos para obtener una aplicación rápida y accesible. Se tratará de utilizar un único documento por herramienta para el desarrollo web.

Trabajar en el hosting gratuito es complicado, pues no posee un editor de texto para facilitar el desarrollo del programador, conque usaré XAMPP y su base de datos local para la mayor parte del proyecto. Comprobaré las diferencias entre trabajar en local y compartirlo con el mundo a través de internet de la mano de AwardSpace, el sitio web que nos proveerá el alojamiento gratuito.



3.2.2. Logotipo de XAMPP: <https://interpolados.wordpress.com/2017/06/27/xampp/>

3.2.3. Logotipo de AwardSpace: <https://www.awardspace.com/>

3.3. Planificación previa:

Empezaremos con pruebas en el servidor local donde verificaremos la conexión de este con nuestros archivos de código. Las tablas serán creadas y modificadas a lo largo del proyecto, para mejorar la legibilidad de consultas y facilitar el intercambio de información cliente-servidor.



3.3.1. Logotipo de PLSQL:

<https://www.oracle.com/database/technologies/appdev/plsql.html>

Dado que usaremos información de dos APIs distintas (tres si contamos con el script externo para obtener los iconos...) aprenderemos a utilizarlas y crearemos las funciones pertinentes para obtener o enviar los datos deseados.

Se intentará no mezclar mucho los lenguajes utilizados en la página web para distinguir la función de cada uno de ellos y que el código sea lo más legible que se pueda.

Me gusta pensar en los lenguajes de programación como piezas para una máquina compleja, no como ingredientes de un puré en el que se mezcla todo tipo de alimentos.

4. Estudio de viabilidad:

En un inicio quería usar el framework de Symfony para agilizar el proceso inicial de desarrollo. Pero me daba problemas a la hora de implementarlo en el alojamiento gratuito conque me decidí a usar PHP puro, que controlo mucho mejor y quiero mostrar mis avances con esta tecnología.



4.1. Logo de Symfony: <https://es.m.wikipedia.org/wiki/Archivo:Symfony2.svg>

4.2. Logo de PHP: <https://es.m.wikipedia.org/wiki/Archivo:PHP-logo.svg>

Este proyecto me ayudó aún más el concepto de modularización. Cada función, cada pieza, que funcione de forma independiente a sus hermanos, de modo que si falla algún componente o elemento sea más fácil corregirlo.

En mi sincera opinión, hacen falta más proyectos de gran tamaño (lógicamente adaptado a nosotros, los estudiantes que aprendemos desde cero) para poner a prueba el desempeño de cada uno a la hora de programar pequeños retos.

Por ejemplo, en mi caso, tuve que hacer todo el código desde cero (con gran parte iniciada) para poder modularizar y optimizar componentes y funciones de este proyecto. Fue algo tedioso, pero vale la pena el ver como se le facilita a uno corregir los errores con un proyecto más ordenado.

Para ayudarme a desarrollar la aplicación, usé XAMPP para depurar y hacer evolucionar mi proyecto, que, como comenté anteriormente, me permitía trabajar con mis propios editores de texto y poner a prueba mi código de manera más rápida y accesible. Pero el objetivo final es subirlo a un hosting externo gratuito y, si sale bien, tenerlo allí para presentarlo en mi futuro como programador.

He notado que XAMP y el hosting gratuito tienen diferencias en función de desarrollo, siendo el hosting más sensible y estricto respecto a la sintaxis y errores en código. Pero nada que no se pueda arreglar.

También apuntar, que un entorno local como XAMPP ha tenido problemas para interactuar con alguna API y eso se llega a conocer a base de prueba y error. Estuve dos horas peleando con mi propio código para corregir un fallo de petición de información a una API externa. Un fallo que no existía en el hosting gratuito.

Para AJAX y el intercambio de datos cliente-servidor, en un inicio opté por usar peticiones “GET”, pero fallaba el enviar un exceso de cabeceras y la información del usuario era más visible.



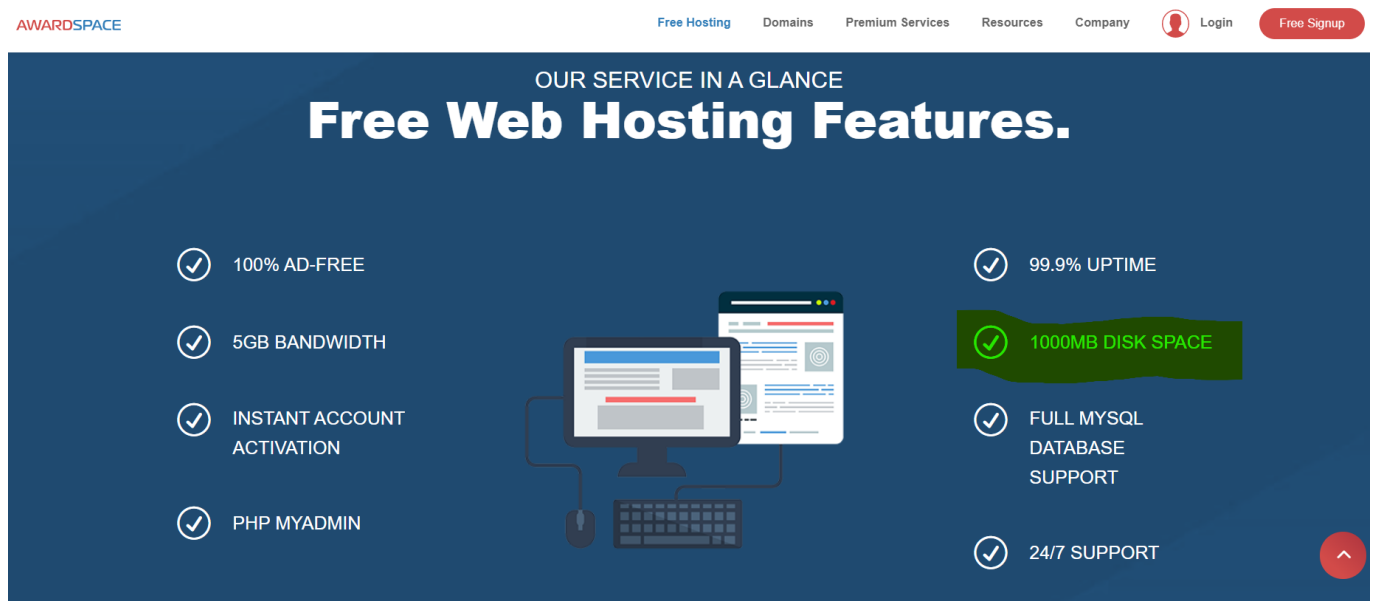
4.3. Logo AJAX: <https://www.vcomputadoras.com/usa-ajax-para-refrescar-un-tag-div/>

Aunque las peticiones “POST” son menos eficientes, preferí usarlas por la sencillez y manejo de peticiones, respuestas y la seguridad que aporta, que, aunque no es mucha, menos es nada.

En la primera ronda de mi trabajo en el proyecto final (si, el que tuve que empezar desde cero) tenía la intención de crear y eliminar elementos del DOM con ayuda de JavaScript, añadiendo eventos, quitándolos, borrando nodos, creando nodos o modificándolos. Pero el trabajo se complicaba y la cantidad de recursos que utilizaría, tanto en entorno cliente y servidor, no valía la pena.

Al final opté por ayudarme con los lenguajes sencillos pero fiables de HTML y CSS para dotar a la aplicación de dinamismo con un simple “display: none/block”. Debo reconocer, que esta estrategia no sé si será la más adecuada, pero la aprendí gracias a mi empresa de prácticas, Stiga, a la que agradezco por mostrarme aplicaciones en un entorno más profesional.

Para subir imágenes, al principio quería limitar el potencial de mi aplicación al usar un hosting gratuito, pues este tipo de servicios tienen muy poco espacio de almacenamiento y, al ser un servidor con limitaciones, ralentizaría mi propio proyecto.



The screenshot shows the AwardSpace website with a dark blue background. At the top, there is a navigation bar with links: 'Free Hosting', 'Domains', 'Premium Services', 'Resources', 'Company', 'Login', and a 'Free Signup' button. Below the navigation bar, the main heading reads 'OUR SERVICE IN A GLANCE' followed by 'Free Web Hosting Features.' in large white text. In the center, there is an illustration of a computer monitor, a laptop, and a keyboard. Surrounding this illustration are eight features, each with a checkmark icon: '100% AD-FREE', '5GB BANDWIDTH', 'INSTANT ACCOUNT ACTIVATION', 'PHP MYADMIN', '99.9% UPTIME', '1000MB DISK SPACE' (highlighted with a green background), 'FULL MYSQL DATABASE SUPPORT', and '24/7 SUPPORT'. A red circular button with an upward arrow is located in the bottom right corner.

4.4. Captura de AwardSpace: <https://www.awardspace.com/>

Por eso, recurrí a alojar las imágenes en un sitio web de alojamiento gratuito del que hablaré más adelante. Se llama “Imgur” y también usaré su API para cumplir el propósito de mi aplicación.



4.5. Logo de Imgur:

<https://tabletzona.es/2016/02/27/sin-nada-que-hacer-comparte-contenidos-con-imgur/>

¿De donde saco esas imágenes tan vistosas y como las muestro?

Pixabay es un sitio web con un banco de imágenes libres que cuenta con una API que podemos utilizar. En un inicio opté por solamente coger las imágenes desde la respuesta JSON del servidor, pero quería personalizar más mi aplicación y creé objetos a partir de ellas. Estos objetos contienen información que generará dinámicamente registros en nuestra base de datos.



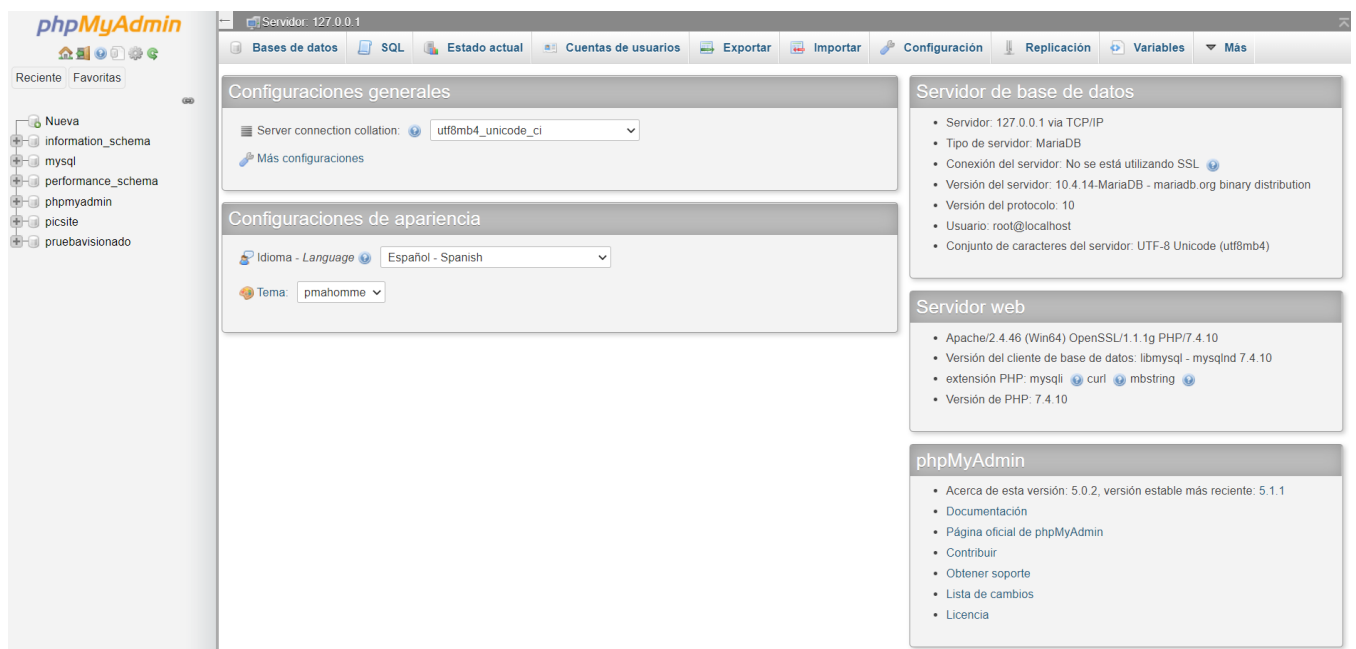
4.6. Logo de Pixabay: <https://www.martechforum.com/herramienta/pixabay/>

La idea inicial de poder contar con unas pocas imágenes a mostrar en mi pequeño proyecto se convertirá en una aplicación cuyo espacio de almacenamiento no será mayor de 50 megabytes y aportará gran cantidad de imágenes sin temor a perder almacenamiento en nuestro hosting gratuito.

5. Análisis de solución escogida:

5.1. Pliego de requisitos:

Tanto el hosting externo como XAMPP utilizan MySQL, conque la solución final fue trabajar con SQL para el intercambio de datos cliente-servidor.



5.1.1. Captura de phpmyadmin desde XAMPP

Para recibir y enviar datos del servidor usé PHP, con un numero reducido de funciones que facilitan el proceso de manipulación de datos de la Base de datos.

Para animaciones, JavaScript fue mi solución. Desarrollé animaciones con funciones en función de los eventos ejecutados por parte del cliente. El redireccionamiento y accesibilidad de la aplicación las creé en el mismo JavaScript para no cargar el índice de tipo HTML.

Formularios y peticiones a servidor también las hice con JavaScript, mayormente, con métodos POST, para controlar el flujo de datos desde el entorno cliente y evitar cargar al servidor. Cuando llamo al servidor de alguna API o de mi propio servidor, procuro recibir toda la información posible para evitar la repetición de llamadas. Esta respuesta llena de contenido la transporto en un array como JSON.

Como las peticiones las hice desde JavaScript, opté por usar AJAX de forma sencilla para evitar el redireccionamiento constante de la página junto con técnicas aprendidas en la experiencia de trabajar con mi empresa de prácticas, Stiga. Una consultora especializada en UX (experiencia de usuario) cuyos formularios enviaban y recibían datos del servidor cada vez que ellos respondían alguna pregunta de una forma eficiente e ideal para la experiencia de usuario. Que de menos para una empresa a la que le interesa la buena opinión del usuario tanto de la red, como del comercio local.

5.2. Planificación de tareas:

Para empezar a trabajar con el proyecto, tenía una idea clara, una única carpeta con pocos ficheros.

Tengo un único documento HTML, donde estructuro casi todos los elementos necesarios para evitar su creación en el fichero JavaScript. Este fichero también abarca los distintos ficheros a utilizar, como su CSS para diseños y estilos, ficheros JS para la creación de objetos, eventos o funciones y scripts externos para personalizar más mi aplicación.

Para las funciones y eventos por parte del cliente, tengo también un único fichero, que al cargarse (window.onload) lo único que aporta son eventos con funciones a casi todos los elementos del HTML.

Estos eventos vienen acompañados de procesos de intercambio de información cliente servidor. Recepción de imágenes e información por parte de las APIs y ocultar o mostrar elementos del HTML en determinados momentos. Aunque son pocos eventos, esto demuestra el dinamismo que podemos lograr con poco código si sintetizamos bien un fichero JS.

Otro fichero JS es para los objetos. El elemento más importante de mi aplicación, son las imágenes y su información. Para poder manejar esta necesidad, recurrí a la creación de un objeto llamado “Foto” que mostrará las imágenes recibidas de la API externa de Pixabay acompañado de unos atributos que vendrán de parte de nuestro pequeño servidor.

El fichero que recibirá datos y enviará respuestas al cliente y servidor es un fichero pequeño con muy pocas funciones, suficientes para poder manejar el intercambio de información de una manera sencilla.

Un fichero que me causa gracia es el famoso fichero “conexión.php” con el cual, con no más de diez líneas de código, podremos acceder tanto a nuestro servidor local (XAMPP) como a nuestro hosting gratuito (AwardSpace).

Para el CSS también utilice un único fichero, que, aunque pueda parecer muy sobrecargado y bastante repetitivo, es esencial para dotar a nuestra pagina web de la vistosidad que requiere mi aplicación.

Como comenté en apartados anteriores, también creé las funciones necesarias para utilizar las APIs. Unas funciones muy sencillas que requerían una clave por usuario y unos detalles concretos dependiendo de cada API, como los parámetros a utilizar en la petición o la forma de verificar mi propia clave.

6. Diseño de la solución escogida:

En primer lugar, hablaré del sitio web que me aporta el hosting gratuito. Pero antes comentaré como aprendí a usar un hosting web.

Gracias a mi profesor Carlos, de la asignatura entornos de desarrollo, descubrí 000webhost. Un sitio web de alojamiento gratuito con poco espacio, pero eficaz para aprender a dominar un hosting y subir aplicaciones sencillas como las nuestras.

Pero este alojamiento tenía muchas más limitaciones que el hosting que acabé utilizando. Como por ejemplo menos espacio de almacenamiento y los anuncios en la página web. Además de que era muy estricto con el contenido que subíamos, habiendo cerrado varios de mis proyectos por licencias o derechos de autor.

Aún así, agradezco mucho a mi profesor Carlos por enseñarme este sitio y abrirme la puerta para encontrar otros sitios web que me aporten alojamiento gratuito.

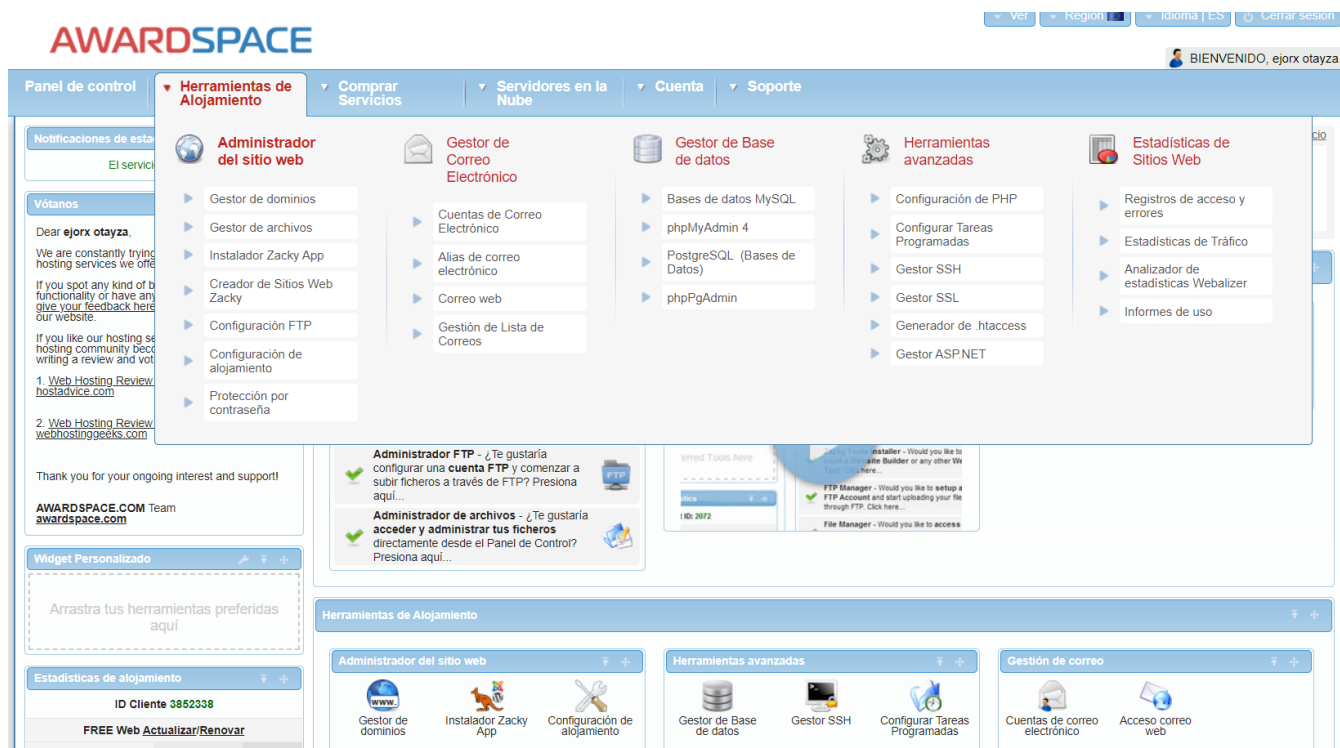


6.1. Logotipo de 000webhost: <https://es.000webhost.com/>

Por eso descubrí AwardSpace, un sitio web que ofrece alojamiento gratuito con mayores prestaciones.

AwardSpace es una empresa de hosting fundada en el año 2003, que se caracteriza por ser una de las pocas empresas que ofrece alojamiento web gratuito, dando a la posibilidad de tener un servicio de hosting sin tener que pagar nada para alojar sus webs. Aunque obviamente los hostings gratuitos están más limitados que los Premium, es algo de agradecer y un compromiso con las personas con menos recursos.

Este hosting me aporta la base de datos, una velocidad mayor, más almacenamiento (aunque mi aplicación no lo necesite) y les quita la publicidad a mis pequeños proyectos. También hay que comentar que ofrece un dominio web gratuito a elegir entre sus algunas opciones.



6.2. Captura de pantalla de sesión iniciada de AwardSpace:

<https://www.awardspace.com/>

El contenido de la aplicación visual de la aplicación viene de parte de la API de pixabay.

Pixabay es una comunidad dinámica de creativos que comparten imágenes y videos sin derechos de autor. Todos los contenidos se publican bajo la Licencia Pixabay, que los hace seguros para usar sin pedir permiso o dar crédito al artista, incluso con fines comerciales.

Para utilizar su API, tuve que crearme una cuenta y obtener la clave que me permitirá obtener cientos de imágenes a disposición de mi aplicación.

Search Images

<https://pixabay.com/api/>

GET

Parameters

key (required)	<i>str</i>	Please login to see your API key here. Login Sign up
q	<i>str</i>	A URL encoded search term. If omitted, <i>all images</i> are returned. This value may not exceed 100 characters. Example: "yellow+flower"
lang	<i>str</i>	Language code of the language to be searched in. Accepted values: cs, da, de, en, es, fr, id, it, hu, nl, no, pl, pt, ro, sk, fi, sv, tr, vi, th, bg, ru, el, ja, ko, zh Default: "en"
id	<i>str</i>	Retrieve individual images by ID.
image_type	<i>str</i>	Filter results by image type. Accepted values: "all", "photo", "illustration", "vector" Default: "all"
orientation	<i>str</i>	Whether an image is wider than it is tall, or taller than it is wide. Accepted values: "all", "horizontal", "vertical" Default: "all"
category	<i>str</i>	Filter results by category. Accepted values: backgrounds, fashion, nature, science, education, feelings, health, people, religion, places, animals, industry, computer, food, sports, transportation, travel, buildings, business, music
min_width	<i>int</i>	Minimum image width. Default: "0"

6.3. Captura de la API de Pixabay: <https://pixabay.com/api/docs/>

Con una petición GET y fijando una serie de parámetros obtenemos una respuesta JSON con cantidad de datos a recibir por parte de la imagen. Además de obtener la foto, también obtenemos datos como el autor, los likes o un identificador único de imagen.

En un inicio quería usar la URL de la imagen como identificador único, pero ésta cambiaba con cada petición. Me pareció algo curioso, pero estuve trabajando con ello durante mucho tiempo.

Con este ejercicio de descubrir una nueva API y usar su información, aprendí a revisarlas lentamente para poderlas utilizar. Pues estuve trabajando mucho tiempo con esa URL de imagen dinámica que no aportaba nada a mi aplicación y, tiempo después, me di cuenta de que podría haber usado el identificador único del JSON desde el inicio.



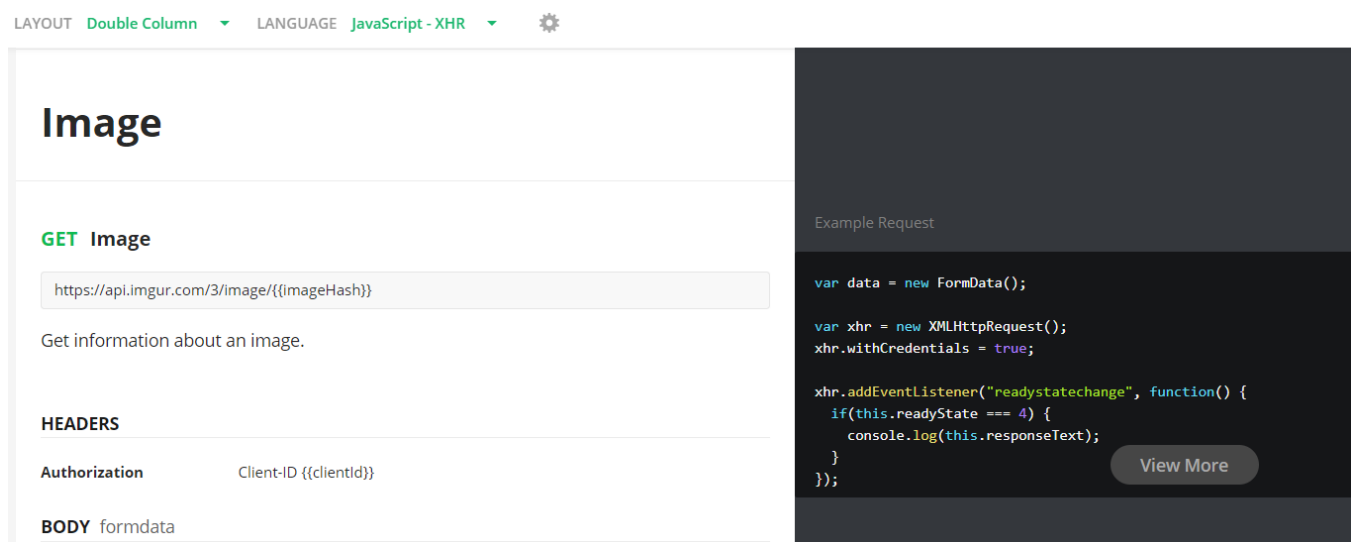
6.4. Página de inicio de Pixabay: <https://pixabay.com/es/>

Como comenté en apartados anteriores, la subida de imágenes se me limitaba mucho por usar un hosting gratuito, por ello, recurrí a otra API de un sitio web llamado Imgur. Con ella, puedo almacenar las imágenes que quiera sin temor a quedarme sin espacio en mi hosting gratuito.

Imgur es un sitio web para alojar imágenes en línea, fundado por Alan Schaaf en el 2009 en Ohio, Estados Unidos, como un regalo para la comunidad de Reddit. Imgur es, actualmente, una de las mayores webs de contenido de imágenes de Internet donde son juzgadas en tiempo real por una comunidad de usuarios a través de comentarios y votos.

La API de Imgur requiere doble autenticación y protege muy bien las imágenes que el usuario no quiere compartir mediante dos claves. Una de ellas es para subir imágenes al alojamiento de Imgur, una clave que permite compartir fotos que serán públicas y otra clave secreta para obtener imágenes o vídeos que el usuario haya subido a su perfil personal.

Su API fue difícil de comprender en un inicio hasta que encontré detalles perfectamente explicados con imágenes de código en función del tipo de petición que quieres enviar. En mi caso peticiones “XMLHttpRequest”.



LAYOUT Double Column ▾ LANGUAGE JavaScript - XHR ▾ ⚙

Image

GET Image

`https://api.imgur.com/3/image/{{imageHash}}`

Get information about an image.

HEADERS

Authorization	Client-ID {{clientId}}
---------------	------------------------

BODY

formdata

Example Request

```
var data = new FormData();  
  
var xhr = new XMLHttpRequest();  
xhr.withCredentials = true;  
  
xhr.addEventListener("readystatechange", function() {  
  if(this.readyState === 4) {  
    console.log(this.responseText);  
  }  
});
```

View More

6.5. Captura de API de Imgur:

<https://apidocs.imgur.com/#de179b6a-3eda-4406-a8d7-1fb06c17cb9c>

Con todo esto, consigo una aplicación muy ligera (literalmente con 2 imágenes y mi código sintetizado) que podrá almacenar mucho contenido muy pesado (gracias a las 2 APIs utilizadas).

Para la base de datos, utilice 3 tablas:

3852338_picsite usuarios id_usuario : int(11) email : varchar(50) contrasena : varchar(20) edad : varchar(3)	3852338_picsite comentarios comentario : varchar(250) id_imagen : int(11) id_comentario : int(11) id_usuario : int(11) fecha : timestamp	3852338_picsite imagenes id_imagen2 : int(11) likes : int(11) descripcion : varchar(200) autor : varchar(40) id_imagen : int(11) url : varchar(200) etiquetas : varchar(200)
---	--	--

6.6. Captura de las tablas utilizadas para mi proyecto.

Usuarios: con la información básica recibida, nombre de usuario, correo electrónico, contraseña o id de usuario para manejar su información de manera sencilla y compartirla en las demás tablas de nuestra base de datos.

Imágenes: con un identificador de imagen, su enlace reducido en la web y los likes que recibirá. He de comentar también que esta es la tabla más dinámica de todas pese a su escaso contenido. Como trabajo con imágenes de sitios web externos, no puedo almacenar desde el principio todo el contenido de estas APIs. En este caso, se generará un registro nuevo cuando accedamos a una imagen nueva y sus atributos.

Comentarios: cuyos campos contiene el id de usuario, identificador de imagen, fecha e identificador de comentario.

Esta tabla, como se ve, esta conectada con todas las tablas y es muy dinámica y recibirá muchas consultas para actualizar una de las partes más importantes de nuestra aplicación, los comentarios en las imágenes para la interacción con los usuarios.

Como dije varias veces, solo tengo 2 imágenes alojadas en mi servidor, el favicon y el logo de nuestra aplicación. Pero ¿de dónde saco los iconos?

Esta parte tan importante en una aplicación que aspira a ser tan visual, también las recibimos de un script externo que no llega a ser una API. Es un framework.

El sitio web de donde saco los iconos se llama FontAwesome y me permite obtener unos pocos iconos, pero indispensables iconos para la parte visual de nuestra aplicación. Son fáciles de manipular y quedan bien en aplicaciones de tipo red social.

Descubrí este sitio web gracias a un compañero en México que, como nosotros, está adentrándose en este mundillo. Es un estudiante de ingeniería de software que me aporta consejos y herramientas para seguir creciendo profesionalmente. He aplicado muchos de sus “tips” para este proyecto y me siento muy agradecido con ello.

Font Awesome es un framework de iconos vectoriales y estilos CSS. Este framework es utilizado para sustituir imágenes de iconos comunes por gráficos vectoriales convertidos en fuentes. Para ello utiliza una librería de más de 400 iconos transformadas en fuentes.



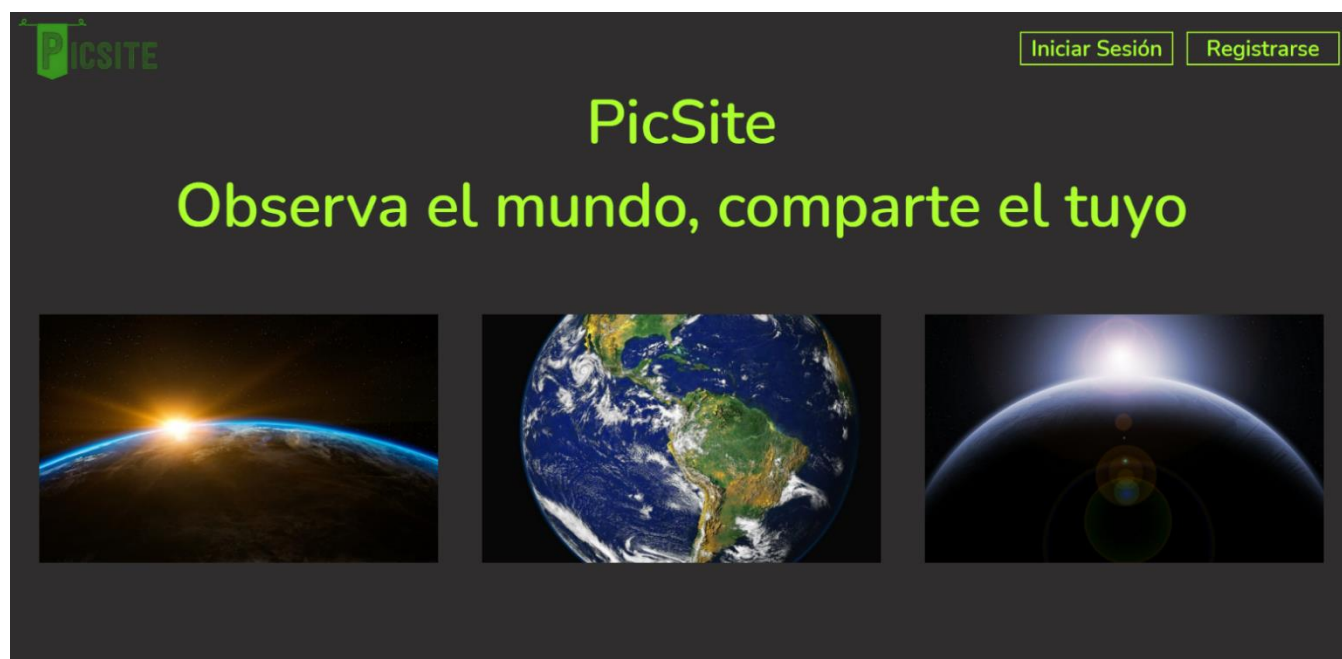
6.7. Logo de Font Awesome: <https://fontawesome.com/>

7. Implementación del diseño:

En esta sección hablaré del diseño de mi página web y un poco por encima de las funciones utilizadas y eventos añadidos.

Este es el índice, que es sobre todo HTML,CSS. Las imágenes se generaron con una función de JavaScript que me da 3 imágenes de Pixabay, relacionadas con el “mundo”.

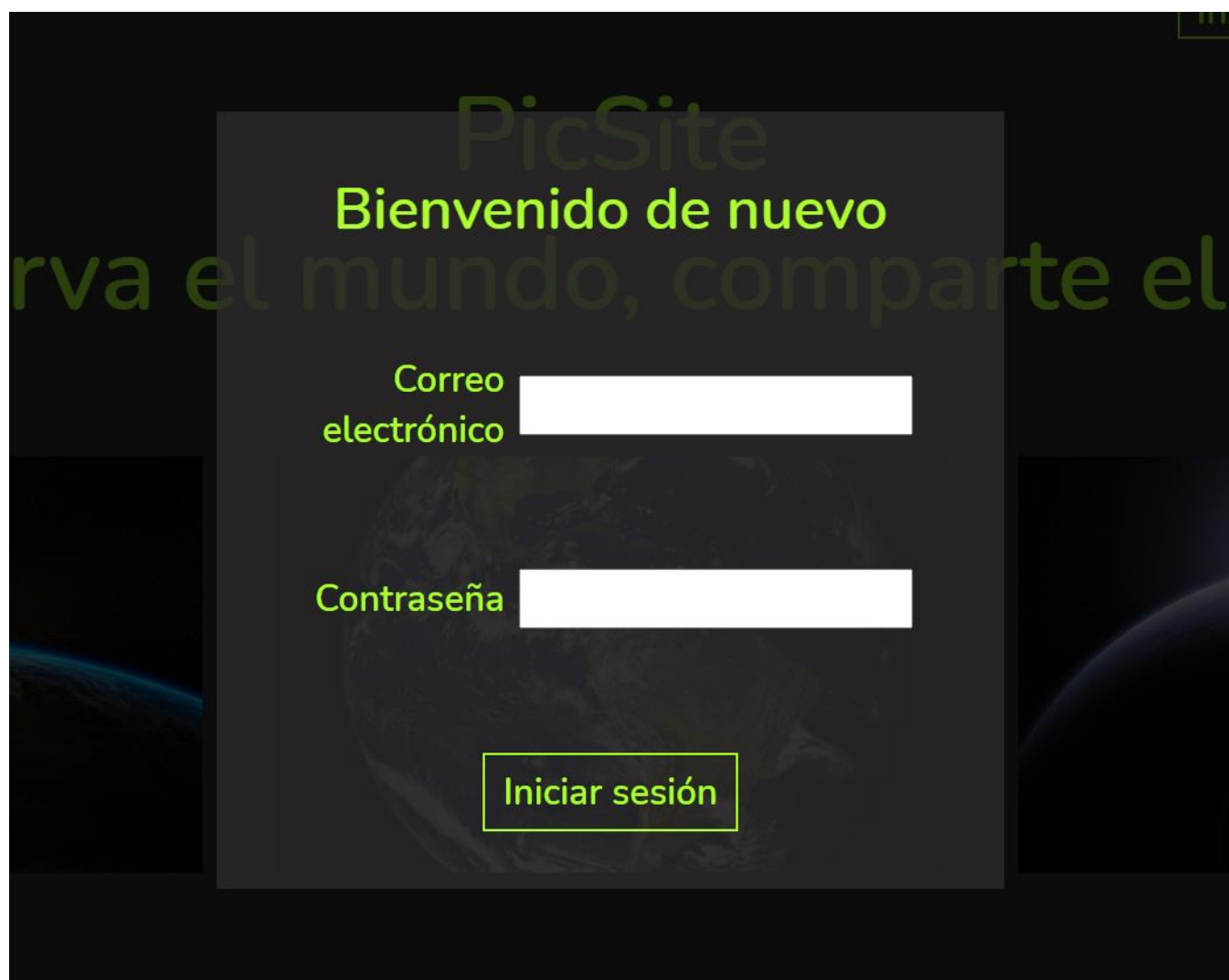
Los botones llaman a funciones JavaScript que activarán modales. Estos modales tienen el mismo diseño y funcionalidad en cualquiera de los formularios que presente la aplicación.



7.1. Index de PicSite “Hello World”

Al presionar el botón iniciar sesión, abriremos el modal con un “display:flex”. Esta función se aplicará a todos los formularios de la aplicación. Con solo añadir la clase “modal” activaremos el diseño y eventos JS para cada formulario.

Huelga decir, que para entrar a la aplicación hay que registrarse, pues en el inicio de sesión buscamos los datos introducidos en el formulario en nuestra base de datos. Si aportamos bien la información, el modal desaparecerá e iremos a la pantalla de inicio de nuestra aplicación.

A login modal form titled "Bienvenido de nuevo" (Welcome back) in green text. The form is centered on a dark background with a faint image of Earth. It contains two input fields: "Correo electrónico" (Email) and "Contraseña" (Password), both with white text labels and white input boxes. Below the fields is a green button labeled "Iniciar sesión" (Log in). The modal has a dark gray background and a thin green border.

PicSite

Bienvenido de nuevo

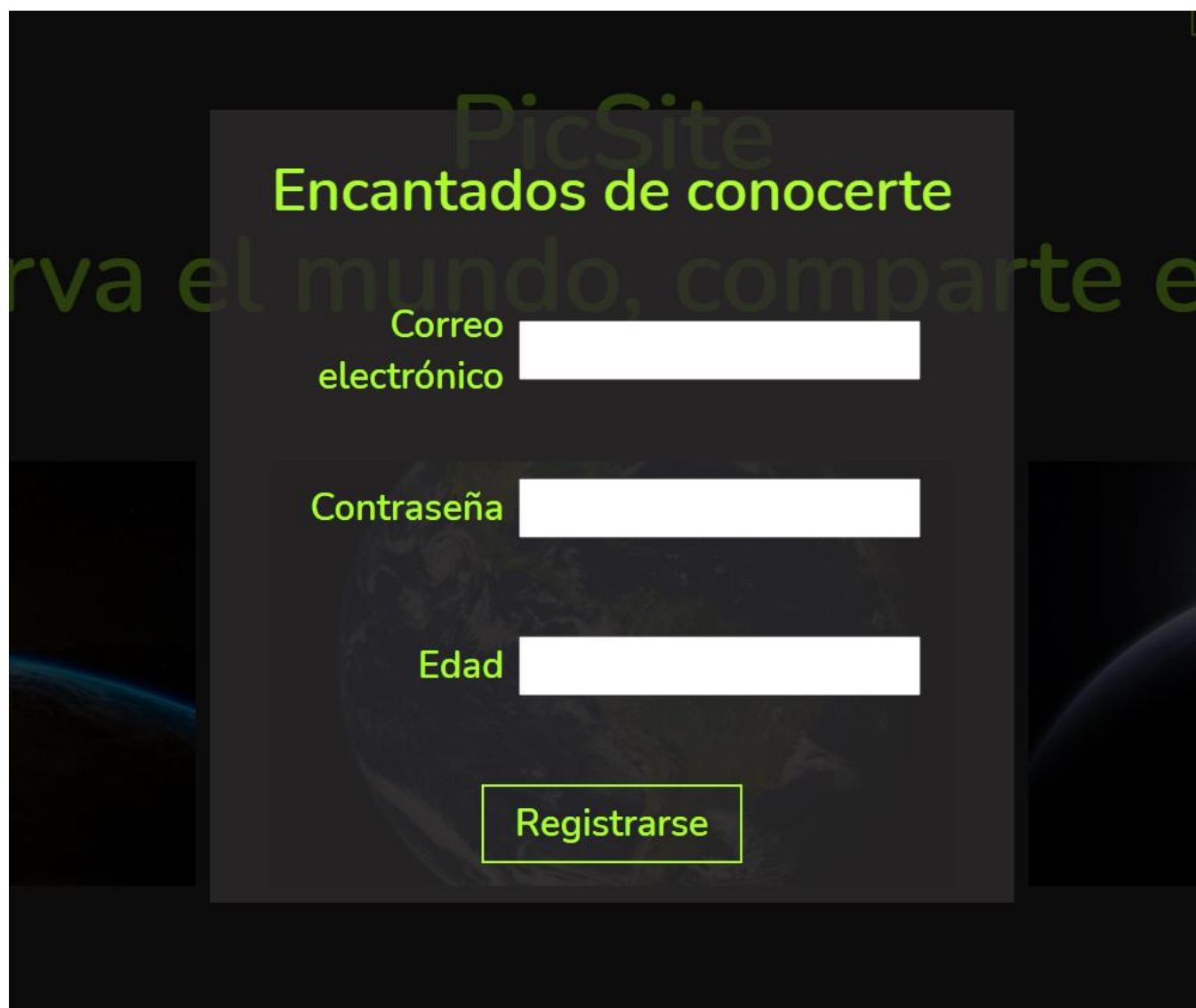
Correo electrónico

Contraseña

Iniciar sesión

7.2. Formulario de inicio de sesión

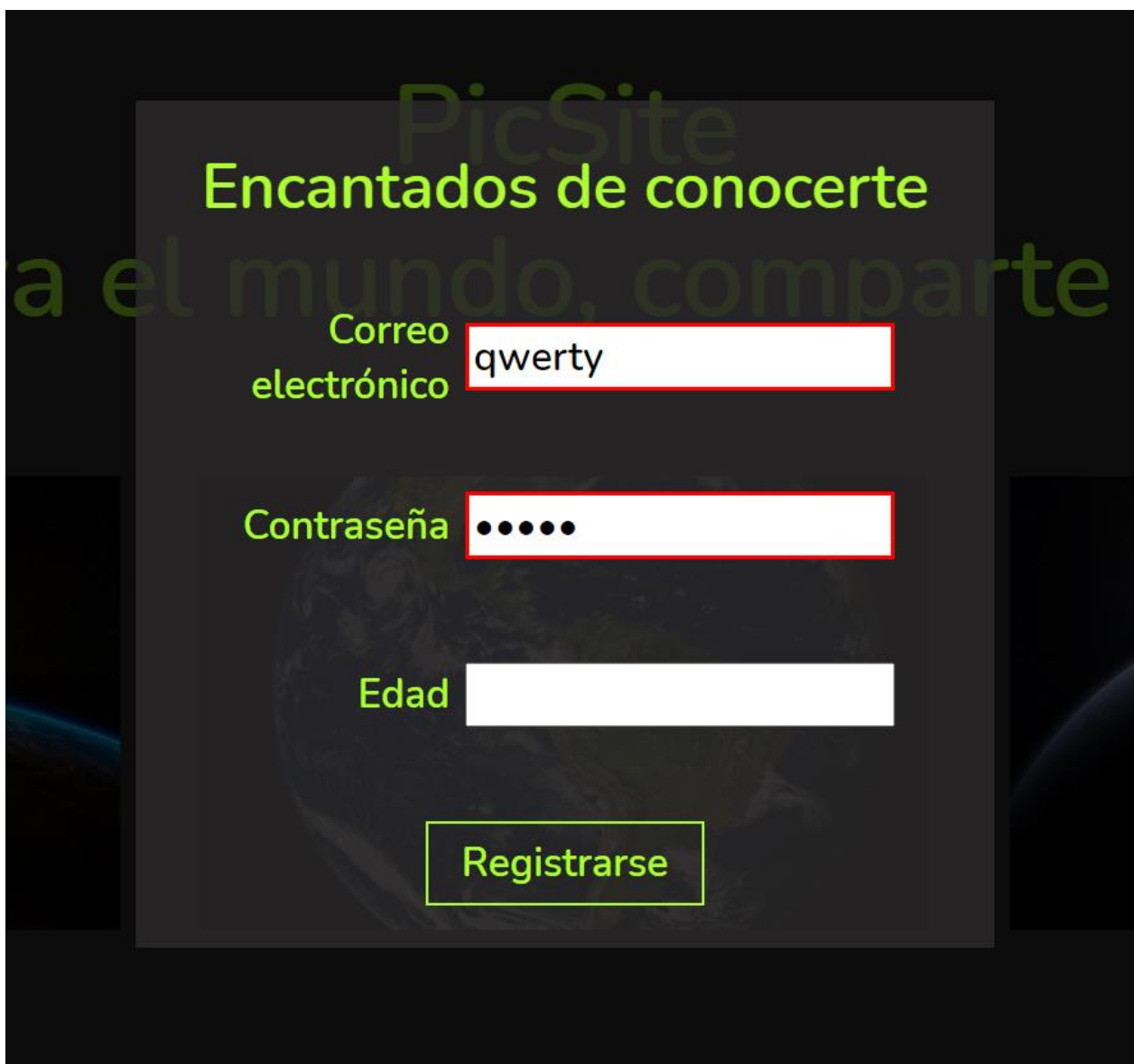
Como explicamos anteriormente, la clase modal aportada a todos los formularios nos provee del diseño y funciones. De ahí su similitud y de paso mantenemos un diseño más o menos homogéneo. En este caso simplemente enviamos una petición al servidor con la información en el login. Pero antes, hay que validarla.

The image shows a registration form for 'PicSite'. The form is a dark gray modal box centered on a black background with a faint image of Earth. The title 'Encantados de conocerte' is in green. There are three input fields: 'Correo electrónico', 'Contraseña', and 'Edad', all with white text and white input boxes. A green 'Registrarse' button is at the bottom. Faint background text reads 'PicSite' and 'Observa el mundo, comparte el tuyo'.

7.3. Formulario de registro

La validación se hará con un evento “onkeyup” y con expresiones regulares. Tienen que cumplirse unas condiciones para que el correo y la contraseña sean válidos. En caso contrario, saltará un alert personalizado en función del campo erróneo.

También quiero comentar que, como en todas las aplicaciones, no puede haber varias cuentas asociadas a un correo. Pero esta validación se realiza en el servidor. Si la petición, mediante una consulta SQL encuentra algún resultado con esa dirección de email. Retornará como respuesta una cadena que significará que el correo ya existe. En tal caso, saltará otro alert.

A registration form titled "Encantados de conocerte a el mundo, comparte" (Encantados de conocerte a el mundo, comparte) for PicSite. The form is set against a dark background with a faint image of Earth. It contains three input fields: "Correo electrónico" (Email) with the text "qwerty", "Contraseña" (Password) with five dots, and "Edad" (Age) which is empty. A green "Registrarse" (Register) button is at the bottom. The text "PicSite" is visible in the background.

Encantados de conocerte
a el mundo, comparte

Correo electrónico

Contraseña

Edad

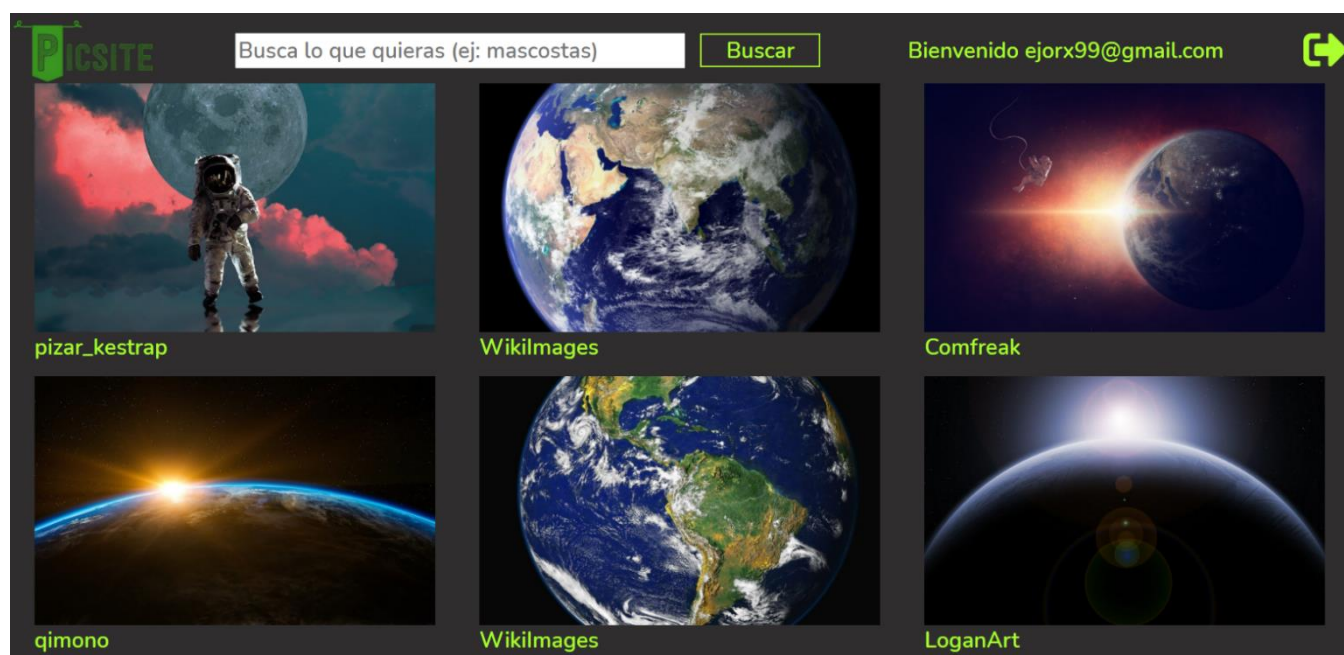
7.4. Formulario de registro con validaciones

Después de hacer todas las comprobaciones pertinentes pasamos a la página de inicio. La barra superior se mantiene, pero gracias a funciones JS ocultamos los botones de inicio y registro y mostramos los nuevos botones. Como vemos, la estructura se mantiene ordenada por el “display:flex” y sus atributos correspondientes.

Por defecto, al entrar en la aplicación, se genera una búsqueda que nos saca nueve objetos fotos, que son resultado de la respuesta de la API de Pixabay Son nueve, para que el scroll se active. El scroll tendrá un papel importante en nuestra aplicación, pues buscamos que el usuario se quede en nuestro sitio web, para ello, el scroll infinito dará cabida a nuevos objetos, con sus respectivas imágenes de la API y los añadirá a nuestro contenedor de imágenes.

La barra superior permanecerá siempre allí con un “position:fixed”, de forma que el usuario pueda cerrar sesión o realizar una nueva búsqueda.

En este punto, dotamos a cada objeto generado de funciones para que nos aporten información a la base de datos sólo con clicar sobre ellos. A cada imagen nueva que cliquemos, generaremos un registro en nuestra BBDD. Me gusta esta funcionalidad porque un solo evento (clic) generará mucho movimiento entre cliente y servidor. Asombroso.

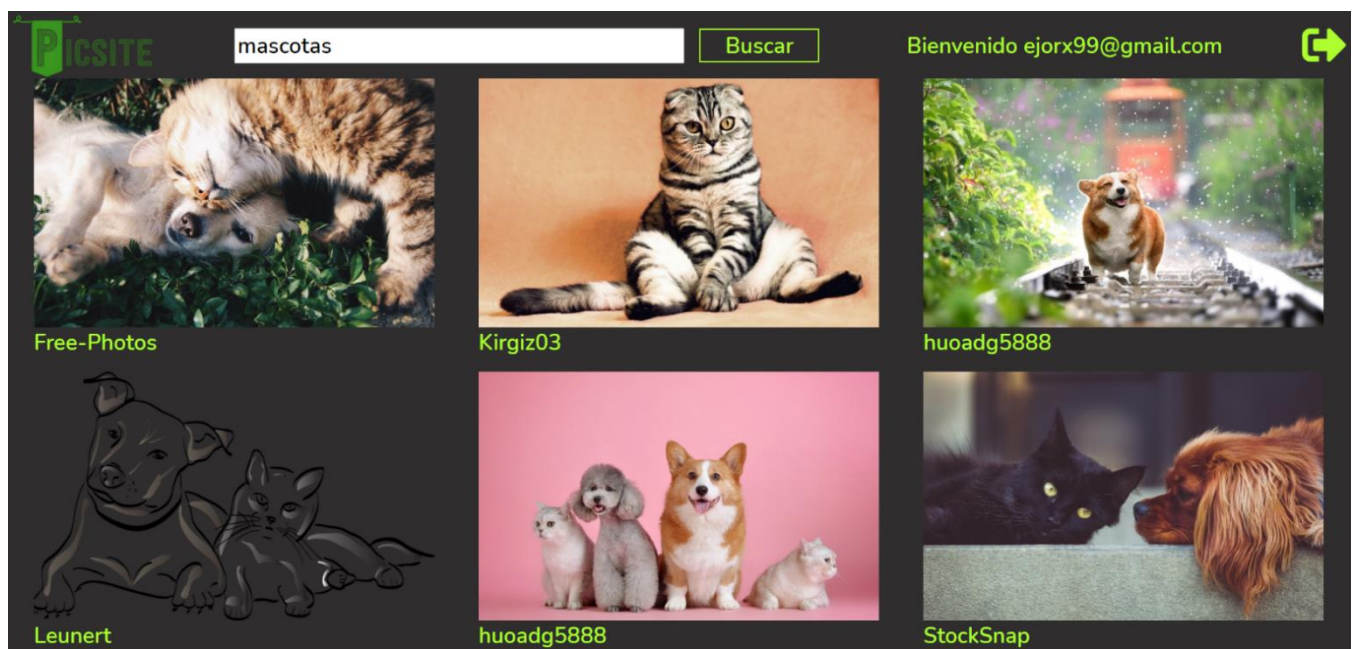


7.5. Página de sesión iniciada de PicSite

Como es lógico, podemos hacer una nueva búsqueda en cualquier momento y, como sucedía en la página de inicio, esta búsqueda nos dará nueve elementos para activar el scroll.

Para favorecer la experiencia de usuario, el buscador y el propio botón de búsqueda contienen la función para ejecutar la consulta de imágenes. Con un simple enter en el buscador, borraremos los elementos de nuestro contenedor de imágenes e insertaremos nuevos objetos con sus respectivas fotos.

Quiero comentar que los usuarios que figuran en las imágenes son los usuarios reales que subieron la foto en Pixabay. También los añadimos junto al enlace a la imagen al objeto Foto.



7.6. Búsqueda de imágenes de “mascotas”

Como dije en párrafos anteriores, el hacer clic en una foto, genera mucho movimiento entre el cliente y servidor.

De parte de el cliente, oculta el menú de inicio y su contenedor de imágenes y muestra con un “display” la nueva sección de comentar imagen. Además de vaciar la sección de comentarios y mostrar la ficha del objeto Foto, con algunos atributos extra. En este caso los likes y la descripción.

Algo que queda regular, es la ausencia de descripción de estas imágenes. Pues la API no nos facilita esa información. Pero las imágenes que subirán los usuarios si se les podrá añadir una.

Los likes si son generados en la respuesta JSON dada por la API. Conque muchas de estas imágenes tendrán un gran número de likes provenientes del mismo sitio web de Pixabay.

Podemos comentar la imagen y, se generará automáticamente el comentario en tiempo real en la sección de comentarios, situada debajo del “textarea”. Estos comentarios generan la fecha, el autor y su comentario.



7.7. Captura de imagen de la API siendo comentada

En la barra superior, encontramos un icono para subir imágenes. Como en los botones de inicio de sesión y registro este botón generará un modal con un formulario de subida de imágenes.

Vemos como se mantiene el diseño con los demás formularios y los pocos campos que nos ofrece. Es importante saber que el campo etiquetas será crucial en nuestra subida de imágenes, en este campo podemos añadir varias palabras que ayudarán al buscador a encontrarlas. Si no ponemos ninguna, no encontraremos la imagen, aunque esté registrada en nuestra base de datos.

The image shows a dark-themed modal window for uploading a new image. At the top, there is a large green text prompt "¡Sube tu propia imagen!". Below this, there is a file selection area with a button labeled "Seleccionar archivo" and a text input field containing "Ni...o". Underneath, there is a section for "Descripción" with a white text input field. Below the description field, there is a section for "Etiquetas" with a white text input field. At the bottom of the modal, there is a green button labeled "Subir mi imagen". The background of the modal features a faint, dark image of a cat's face.

7.8. Formulario de subida de nueva imagen

Al subir alguna imagen al formulario con el “input” correspondiente, se nos mostrará una vista previa. Como sabemos, JavaScript no puede ver los archivos de nuestro ordenador, pero puede ver la imagen que hemos subido y mostrarla mientras la subimos a nuestro alojamiento de imágenes, Imgur.

Al hacer clic en “Subir imagen”, guardaremos el enlace generado con Imgur, la descripción y las etiquetas a la base de datos. Ya podremos encontrar nuestra imagen escribiendo alguna de las etiquetas escritas.

ej: ma

¡Sube tu propia imagen!



Seleccionar archivo

JI...PG

Descripción

mola ese videojuego

Etiquetas

personaje, valhalla,juego

Subir mi imagen

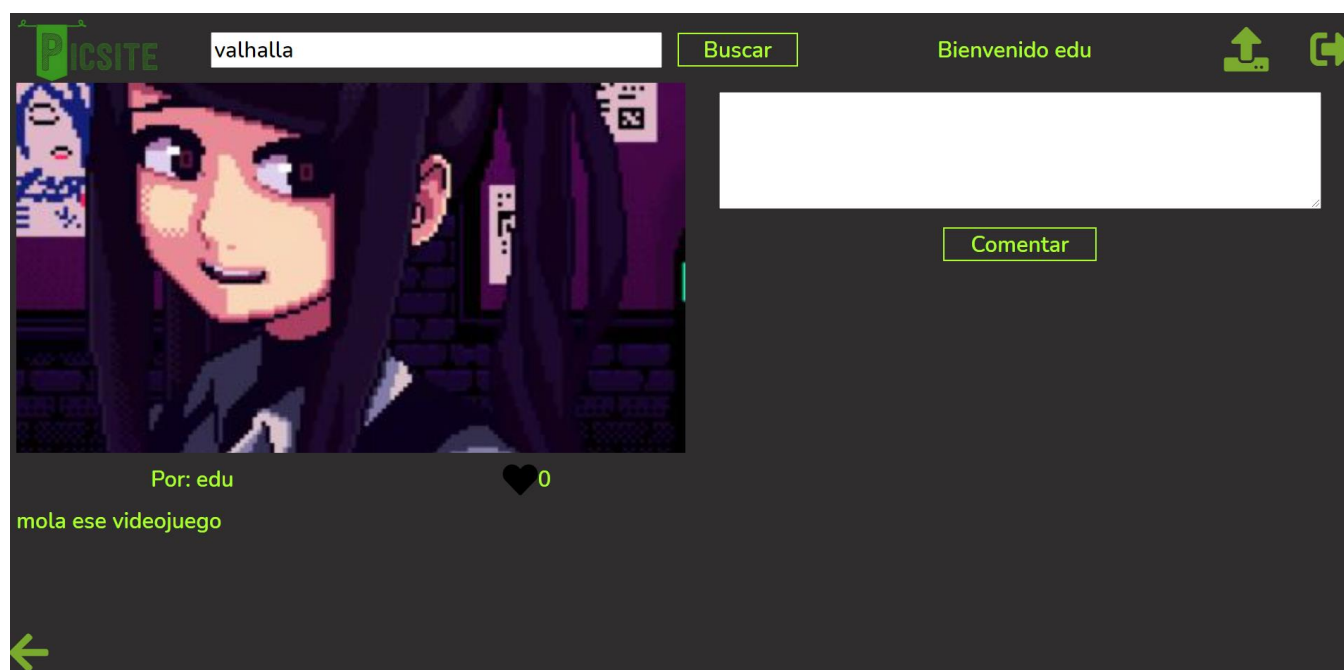
Por: Free-Photo

7.9. Previsualización de subida de imagen.

La imagen aparecerá en el contenedor de imágenes como una foto más. Podemos realizar las mismas funciones en ella como en cualquier otro objeto Foto. Dar «me gusta», comentar y ver comentarios.

A diferencia de las demás imágenes, que no tenían descripción porque la API de Pixabay no nos la proveía, las imágenes subidas por el usuario pueden tener la descripción que ellos pongan en el formulario de carga de imágenes.

El autor se informa correctamente en la base de datos gracias al valor de la sesión obtenida en PHP pues no se puede subir una imagen sin haber iniciado sesión.



7.10. Comentando la imagen subida.

8. Pruebas:

Las pruebas ejecutadas en la aplicación se pueden dividir en tres grandes pruebas. Cada uno referido a una tabla.

Las validaciones y registros de muchos usuarios para que se verifique su estado de registro o no referentes a la tabla usuarios.

La creación de imágenes en la tabla imágenes al clicar en una imagen nueva y al subirla desde el botón de cargar imagen y el funcionamiento correcto de los eventos y funciones asociados a sus objetos.

Y la creación de comentarios en su tabla verificando que todo se cumpla correctamente. El usuario correspondiente, la hora y el mismo comentario y que se carguen de forma dinámica en la sección comentarios de su imagen correspondiente.

Después de hacer decenas de pruebas, verifiqué su correcto funcionamiento y procedí a dar por finiquitado el proyecto.

9. Conclusiones:

9.1. Valoración personal

Este proyecto ha sido una gran experiencia. No sólo hemos tenido que aplicar nuestros conocimientos aprendidos durante estos dos años, sino que hemos tenido que buscar soluciones a los problemas que iban surgiendo a lo largo del desarrollo de nuestro proyecto.

Gracias a la preparación y experiencia previa durante el ciclo formativo se han superado todos los obstáculos y hemos seguido aprendiendo aún sin asistir a clase. Me parece interesante que este segundo año de curso sea más independiente porque nos preparó para esto.

Recuerdo que tenía dudas sobre hacer el segundo año de DAW, mi familia no confiaba en mi y primero me costó sacarlo. Pero, reconozco que mi tutor de primero, Marcos, me animó un poco a la hora de decidirme. Me dijo que no sabía evaluar cuál de los dos años era más complicado, pero me hizo una comparación curiosa. El primer año es estudiar para crear cohetes, puede resultar muy aburrido y cansado. Pero segundo es más como crear tus propios cohetes, tienes más libertad a la hora de crearlos y ver su resultado te fascina.

Pues tenía mucha razón. Este año hemos sido más independientes. Y aunque no me parece del todo correcto que no tengamos una atención tan personalizada, aunque seamos solo 15 alumnos, apelaré a la situación actual de pandemia y daré gracias a todo lo que aprendí en el centro con mis profesores y fuera del centro con mi propio esfuerzo.

Aunque anteriormente lo he comentado, hacen falta más proyectos como estos durante los dos años de curso. Las prácticas y ejercicios en clase están bien, pero unir muchas de las tecnologías que conocemos para enfocarnos en un único objetivo es muy fructífero y nos obliga a dar todo lo que tenemos. Además de darnos cuenta de los problemas que van surgiendo a lo largo del desarrollo y ayudarnos a mejorar nuestro código para que, tanto nosotros como nuestro equipo, pueda entender nuestra aplicación.

También valoro positivamente la estancia en la empresa mientras realizamos el proyecto. Aunque es un poco duro compaginar el desarrollo con las 8 horas en la empresa, si uno está atento y tiene ganas de aprender, aplicará todo lo que capta en la empresa y en clase para mejorar su proyecto y prepararse para entrar en el mundo laboral.

9.2. Objetivos alcanzados

Esta sección la escribiré como un texto más. Pues no me gusta hacer una lista de objetivos y ponerle un “tick” al lado.

El objetivo primordial de mi aplicación era utilizar las imágenes de una API de forma que pueda manipular el contenido visual y utilizarlo a mi favor. Creo que lo he cumplido con creces gracias a las claras explicaciones que proporcionó la API y a mis conocimientos adquiridos durante el curso.

El segundo objetivo era hacer que la aplicación sea dinámica, sin redireccionamiento y sin apenas esperas. Consideraba que hacer muchos ficheros HTML, CSS y PHP y conectarlos con un elemento de enlace (“<a>”) no era un desafío y quise manejar las peticiones “XMLHttpRequest”, que tanto me habían gustado desde que nos las mostró el profesor de Desarrollo de Aplicaciones Web en entorno cliente. Además, en la actualidad se buscan páginas rápidas y sin esperas, conque mejor familiarizarse con ellas desde ahora.

El tercer objetivo es crear algo útil. Hace poco leí un libro que daba muchas definiciones sobre qué es ser un programador. Una de ellas (y una de las más acertadas según el autor) dice que uno se convierte en programador al crear algo útil.

Puede que muchas de nuestras aplicaciones sean sencillas, pero si son capaces de solventar algún error o mostrar alguna utilidad al mundo, seremos pequeños programadores.

Y, en mi caso, estoy comenzando varios proyectos y quiero seguir creciendo para consolidar ese título que tenemos todos los que superamos el ciclo de Desarrollo de Aplicaciones Web.