

TECNOLOGÍAS DE LA INFORMACIÓN PARA LAS ORGANIZACIONES

MyPhoto

Josué Toledo Castro
José Alberto Mena García
Gonzalo García León
María Nayra Rodríguez Pérez

ÍNDICE:

Introducción.....	2
Definición de modelo y oportunidades de negocio.....	2
Tecnologías implicadas en la aplicación.....	4
Descripción de la problemática y la aplicación desarrollada.....	8
Problemática.....	8
Aplicación desarrollada.....	10
Conclusiones.....	15
Bibliografía.....	16

Introducción

En este documento se redactará un informe sobre la realización del proyecto final de la asignatura Tecnologías de la Información para las Organizaciones, del Grado en Ingeniería Informática de La Universidad de La Laguna, desarrollado por Josué Toledo Castro, María Nayra Rodríguez Pérez, José Alberto Mena García y Gonzalo García León.

Dicho proyecto consiste en la elaboración de una aplicación que genera información útil a partir de imágenes de cualquier usuario con acceso a la aplicación, extrayendo los metadatos de las propias imágenes y procesándolos para generar información de manera que se puedan sacar conclusiones sobre ellas, el entorno o sobre las personas que realizan dichas imágenes.

La aplicación trata de un portal en el que los usuarios comparten fotos sobre sus experiencias, que luego serán procesadas para extraer sus metadatos y poder mostrar información acerca de ellas cuando alguien acceda a ver dicha foto, y donde también tendrán la posibilidad de buscar fotos de otros usuarios y ver la información relacionada a dichas fotos.

Nuestra herramienta ejerce la función de obtener los datos de las imágenes que han sido subidas por los usuarios, y convertirlos en información, mostrándola de manera fácilmente entendible para nosotros con el objetivo de sacar conclusiones sobre las circunstancias que rodean a las imágenes y a sus autores.

Definición de modelo y oportunidades de negocio.

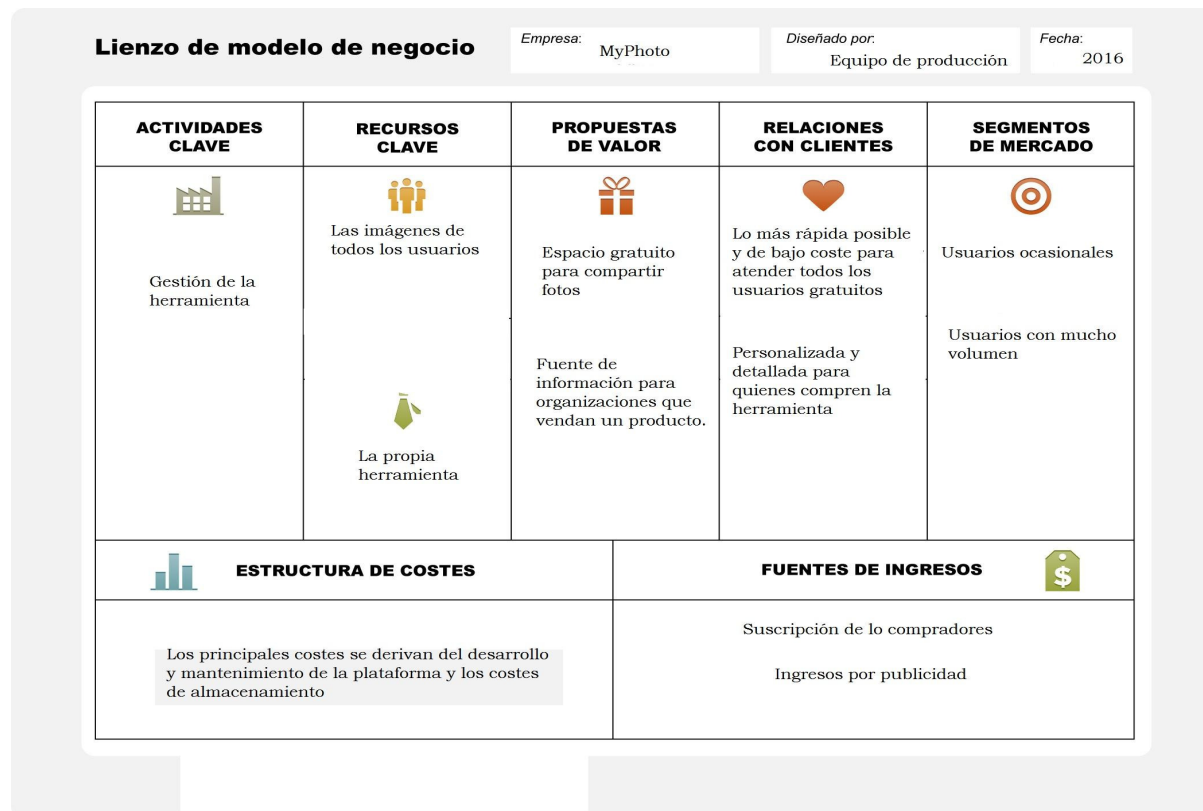
Para que un negocio en internet sea rentable debe cumplir las siguientes premisas:

- Ser considerado de gran valor para los usuarios/clientes.
- Modelo de negocio replicable (utilizable fácilmente por muchos usuarios).
- Modelo de negocio altamente escalable (el incremento de las ventas o ingresos ni implica el mismo incremento en costes o estructura).

Nuestra herramienta buscará generar este negocio a través de la consecución de una comunidad importante de usuarios, a través de la cual se puedan obtener datos de los que sacar información concluyente con el fin de que crezca el interés de entidades y organizaciones en obtener la aplicación para abastecerse de sus utilidades y así poder incluirla a sus herramientas internas para favorecer su negocio.

Nuestro modelo de negocio se basa en la participación e interacción de una comunidad de usuarios que comparten sus experiencias a través de fotos que realizan durante sus vivencias, nuestra herramienta se sirve de esas imágenes para generar información que pueda ser de utilidad para diferentes tipos de organizaciones que busquen vender un producto o un servicio a las personas.

A continuación describiremos gráficamente el modelo de negocio en el que se desarrollaría nuestra herramienta, que no es más que describir el modo en que buscaremos crear, distribuir y capturar la atención de un segmento de mercado para conseguir generar negocio y hacer que prospere el mismo:



En lo que se refiere a las oportunidades de negocio, deberíamos explorar empresas que quisieran saber información sobre dónde se sacan más fotos generales (lo cual podría hacernos ver en los lugares donde se concentra mayor número de personas) o específicas de una categoría (esto nos podría indicar donde se mueven las familias, el turismo o determinados grupos de personas), entidades que quieran saber el tipo de perfil de persona que saca fotos en ciertos lugares, los motivos por los que se sacan estas fotos o las maneras en que se toman las instantáneas.

Toda esta información puede servir para saber dónde puede colocar una empresa su establecimiento, en qué productos enfatizar según los grupos de personas que suelen frecuentar la zona donde está mi establecimiento, labores de marketing o para realizar eventos en zonas donde se tendrá cierta certeza de que encontraremos personas con el perfil que buscamos.

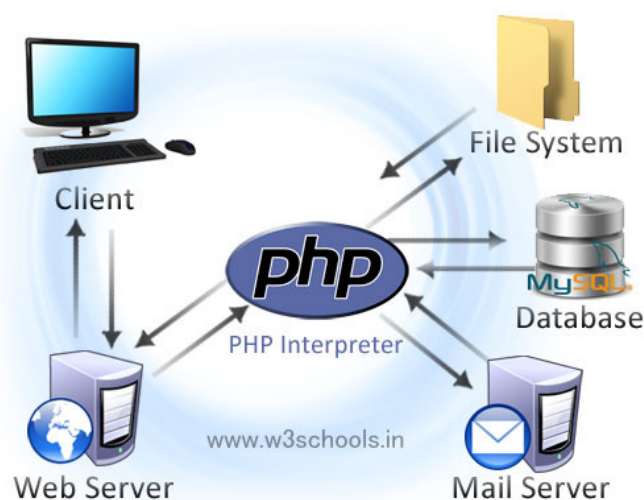
En resumen, el beneficio final que puede obtener una organización de esta herramienta no es más que orientarse a la hora de ofrecer un producto o servicio, qué ofrecer, cómo ofrecerlo o dónde ofrecerlo.

Tecnologías implicadas en la aplicación

Para el desarrollo de esta aplicación se han utilizado las siguientes herramientas y los siguientes lenguajes de programación web:

- **PHP.**

Es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Su desarrollo está centrado en la programación de scripts y el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. Una de las características más potentes y destacables de PHP es su soporte para un amplio abanico de bases de datos, utilizando extensiones específicas de bases de datos.



- **JavaScript.**

Lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

- **HTML5.**

Es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. Establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos.

- **CSS.**

Es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.



Por otro lado, se han utilizado otras tecnologías alternativas para la elaboración de la aplicación Myphoto.

- **API de Google Maps**

Es un servicio de un conjunto de APIs que ofrece Google y que permite superponer nuestros propios datos en un mapa de Google personalizado. Con ésta herramienta se pueden crear webs más atractivas y aplicaciones móviles con cartografía de Google incluyendo: mapas desplazables del mundo entero, fotos satélites, vista de calle, direcciones, la ruta más corta entre dos ubicaciones y muchas más características. Con la API de Google Maps es posible hacer todo tipo de aplicaciones para mostrar mapas de cualquier parte del mundo, ya que dispone de la cobertura global más precisa del mundo haciendo actualizaciones diarias.

El sistema de negocio de la API se basa en dos planes, un plan Estándar y otro Premium. El plan Estándar, que es el que nos interesa para este proyecto, permite al usuario usar la API de forma gratuita hasta un cierto número de peticiones diarias, en este caso, Google nos permite realizar 25.000 cargas diarias durante 90 días antes de empezar a cobrarnos, más que suficiente para lo que pretendemos con nuestra aplicación. Es conveniente puntualizar que este plan se divide para Android, iOS y Web, variando en el número de cargas diarias permitidas.

Google cuenta además con una API similar llamada Google Places centrada más en ofrecer datos geográficos y comerciales que posee Google. La siguiente tabla refleja las restricciones para el plan Estándar de esas dos APIs principales.

	Google Maps API	Google Places API
Android	Gratuito ilimitado.	1000 solicitudes gratuitas por día predeterminadas, aumentadas a 150.000 solicitudes gratuitas por día después de la validación de la tarjeta de crédito.
iOS	Gratuito ilimitado.	1000 solicitudes gratuitas por día predeterminadas, aumentadas a 150.000 solicitudes gratuitas por día después de la validación de la tarjeta de crédito
Web	Gratis hasta exceder 25 000 cargas de mapas por día durante 90 días consecutivos.	No disponible.

Entrando en un apartado más técnico, la API de Google Maps provee de una serie de funciones java script que hace al programador muy fácil insertar mapas personalizados dentro de una web. Google pone a disposición una documentación muy completa que además cuenta con un tutorial de aspectos básicos que estaría bien mencionar:

En el siguiente ejemplo de código se muestra cómo colocar un marcador en una localización concreta y mostrar el marcador en el mapa:

```
function initMap() {
  var myLatLng = {lat: -25.363, lng: 131.044};

  // Create a map object and specify the DOM element for display.
  var map = new google.maps.Map(document.getElementById('map'), {
    center: myLatLng,
    scrollwheel: false,
    zoom: 4
  });

  // Create a marker and set its position.
  var marker = new google.maps.Marker({
    map: map,
    position: myLatLng,
    title: 'Hello World!'
  });
}
```

En el ejemplo anterior observamos como Google nos provee de una interfaz orientada a objetos muy sencilla, el primer objeto que se crea es el propio mapa, se editan ciertos atributos estableciendo un centro donde presumiblemente deberá de ir colocado el marcador, el mapa tiene múltiples opciones de visualización, se puede usar modo híbrido, modo satélite, modo mapa. Permite también estilizar los mapas con colores personalizados que puedan encajar mejor con el estilo visual de nuestra aplicación.

Posteriormente se crea un objeto marcador al que se le asigna como mapa el objeto creado anteriormente, se indica cual es la posición del marcador y un nombre. También es posible agregarle una imagen personalizada al marcador.

- **XAMPP.**

Es un servidor independiente de plataforma, software libre, que consiste principalmente en la base de datos MySQL, el servidor Web Apache y los intérpretes para lenguajes de script: PHP y Perl.

Para el uso de este servidor, una vez instalado (Windows) se activa Apache y MySQL. Como Apache abre puertos en el ordenador, por lo general el firewall de Windows pide al usuario confirmación. Para poder utilizarlo se autoriza el acceso a las redes privadas. Si el arranque de Apache tiene éxito, el panel de control mostrará el nombre del módulo con fondo verde, su identificador de proceso, los puertos abiertos (http, https y ssl para mysql), el botón “Start” se convertirá en “Stop” y en la zona de notificación se verá el resultado de las operaciones realizadas.

Si el arranque no tiene éxito, es posible que los puertos 80 y 443 estén siendo utilizados por alguna otra aplicación, para solucionarlo se cambian los puertos que por defecto utiliza Apache y MySQL mediante la edición de los ficheros de configuración: “httpd.conf” y “httpd-ssl.conf”. En estos ficheros se sustituye el puerto 80 por el 8080 y el puerto 443 por el 4430. Al realizar este cambio en la configuración del XAMPP también se cambia la forma de llamar a nuestro servidor local, ahora debemos escribir en el navegador: localhost:8080.

Por último, para añadir nuestra aplicación web al servidor y poder ejecutarla se ha introducido en la carpeta “htdocs” situada en la carpeta principal de la instalación de XAMPP. Para acceder a las aplicaciones creadas lo hacemos a través de la ruta de nuestro localhost (http://localhost/nombre_archivo).



Descripción de la problemática y la aplicación desarrollada

Problemática.

En el desarrollo de la aplicación se han encontrado algunas dificultades con el tratamiento de las imágenes y la gestión de la información y las acciones con el servidor dónde se ha alojado la página web. Como servidor se planteó la posibilidad de utilizar el servidor banot de la Escuela Superior de Ingeniería y Tecnología (ESIT). El uso del mismo requiere un proceso de autenticación de usuario, aportando las credenciales necesarias para garantizar el acceso. En este sentido, este servidor originó problemas en la administración de la base de datos necesaria para el almacenamiento de, entre otros aspectos, los datos de los usuarios de la aplicación, como, por ejemplo, imágenes y fotos con metadatos. En relación con estas últimas, se presentaron problemas en relación principalmente al espacio necesario para el almacenamiento de las mismas y a las restricciones y aspectos de configuración impuestos por el servidor banot que dificultan su manipulación. En este sentido, podemos destacar algunos de los siguientes problemas:

1. **Cuota de disco.** El servidor establece una cuota de disco para el almacenamiento de archivos para los distintos usuarios, limitando las necesidades de la página web en cuanto a espacio, necesarias para la subida de imágenes por parte de dichos usuarios y afectando a la posibilidad de dotar al proyecto de un grado de escalabilidad alto y suficiente, en la medida de que no resulta viable la subida de demasiadas fotos ni el registro de demasiados usuarios.
2. **Tamaño de archivos.** Es común en los archivos de configuración de un servidor el establecimiento de parámetros como el tamaño máximo de archivos (archivo *php.ini*). En el caso de banot, el tamaño máximo permitido se correspondía a 2Mb, imposibilitando en gran medida la subida de numerosas imágenes cuyo tamaño

podría ser superior al comentado por razones como la calidad configurada, flash, etc.

3. **Gestión de la base de datos.** Se presentaron dificultades durante las interacciones con el sistema gestor de base de datos en banot: Mysql. Algunos comandos y manejadores para la administración de las tablas de la base de datos de Myphoto como, por ejemplo, *describe* (comando para visualizar la estructura y las características de cada tabla), no funcionaban correctamente.
4. **Tiempo de carga de la API de Google.** Debido a los distintos módulos y características de la API de Google para la creación de mapas, el tiempo de carga necesario para su utilización constituye un incremento importante de tiempo que hay que sumar al tiempo total de carga de Myphoto.
5. **Tiempo de carga de la página web.** El hecho de precisar la utilización de recursos externos como, por ejemplo, la API de Google (para la gestión y la creación de mapas a partir de los parámetros de latitud y longitud), repositorios en Internet que albergan distintos tipos de iconos (como, por ejemplo, el sitio web Font Awesome Icons) o diversas urls necesarias para la utilización de jquery, generaba una demora importante en la visualización de los contenidos de Myphoto, lo cual no resultaba productivo en la satisfacción del cliente durante su interacción con la página web. Además, otros aspectos también importantes en este sentido han sido el tratamiento con imágenes de distintos tamaños y características y el acceso al servidor banot a través de la red.

En consecuencia de estas dificultades, se decidió abandonar el uso del servidor de la ESIT, pasando a la utilización de XAMPP trabajando de una manera local y gestionando la información a través de la aplicación phpmyadmin, cuyo uso agiliza en gran medida las labores de mantenimiento y gestión de los datos, incluidas las imágenes. No obstante, se presentaron dos problemas:

1. **Configuración de puertos para el XAMP.** Debido a las características de cada máquina local y a los puertos que utiliza para posibilitar los distintos servicios, se incluyó y configuró el puerto 8080 para dar cobertura y garantizar el funcionamiento correcto de Apache (servidor web HTTP de código abierto), debido a que el puerto configurado por defecto se encontraba ocupado gestionando otros servicios en la máquina.
2. **Uso prioritario de versiones actualizadas y actuales de php.** Se hizo necesaria una serie de adaptaciones en código php para la gestión de los datos y las interacciones entre el código html y la información almacenada en la base de datos, principalmente en las funciones predefinidas para mysql como *mysqli_query* o *mysqli_connect*, entre otras.

Aplicación desarrollada.

Myphoto es una aplicación web destinada a la compartición de imágenes realizadas con dispositivos móviles y dotadas de metadatos. En ella se puede obtener información con los metadatos extraídos en las mismas y mostrando al usuario en un mapa de Google la posición exacta en el momento de realizar la fotografía.

La aplicación ha sido desarrollada basándonos en una red social tipo Flickr, en la cual se realiza una autenticación de usuarios y se comparten fotografías, pudiendo filtrar éstas por categorías, buscar una foto, ver fotos de otros usuarios, etc.

En primer lugar, al acceder a la página de Myphoto el usuario deberá autenticarse, para ello se deberá realizar antes un registro de usuario rellenando los campos en el formulario correspondiente. Después de efectuar el registro ya el usuario se encuentra en la base de datos de nuestra aplicación y por tanto ya puede proceder a iniciar sesión.

MYPHOTO

Registrarme

Inicio de sesión

Introduce tus datos de usuario

INICIA SESIÓN



MYPHOTO

Iniciar sesión

Formulario de registro

Introduce tus datos de usuario

Nombre:

Apellidos:

Usuario:

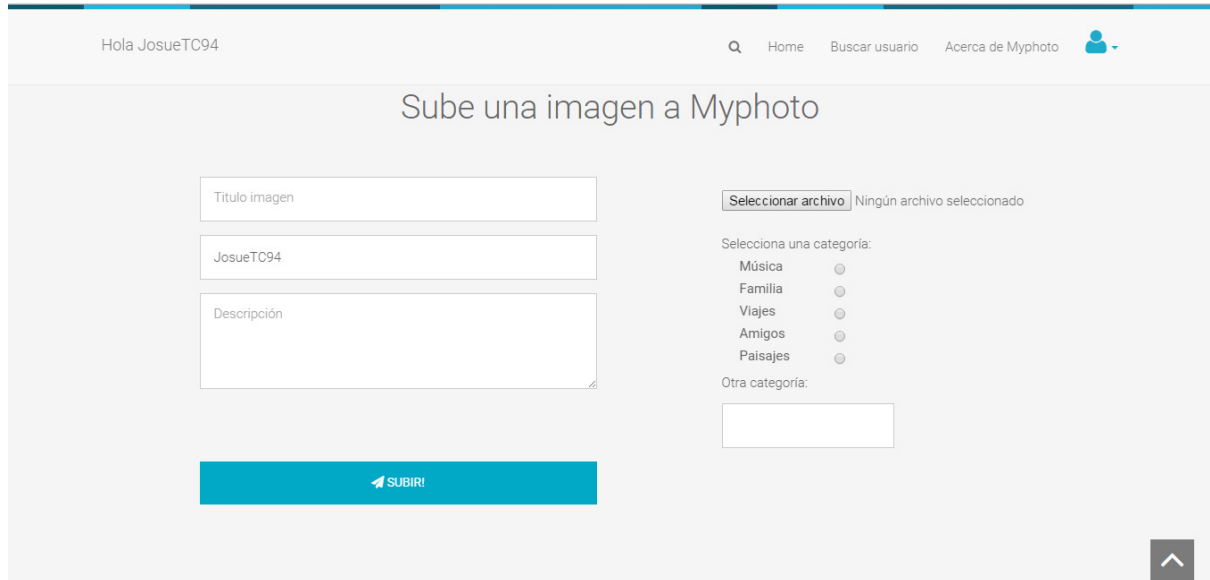
Password:

Confirmar password:

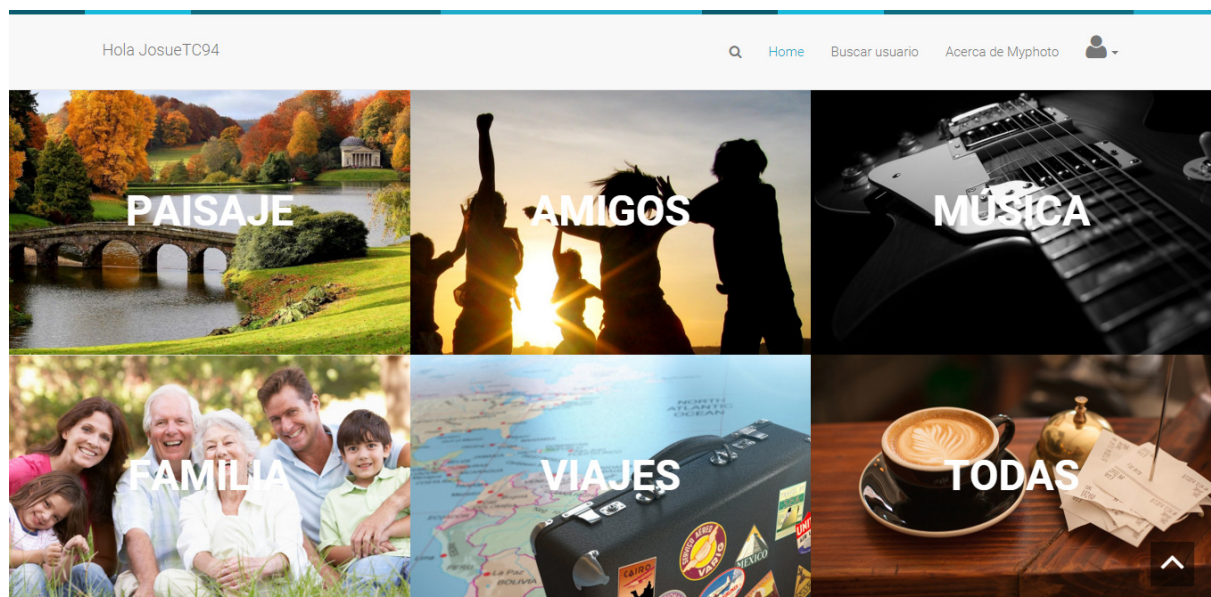
Email:



La principal herramienta que ofrece nuestra aplicación es la subida de imágenes a la plataforma, el usuario accede a la sección de ‘Subida de imágenes’ y realiza un update con la imagen que desee especificando los campos: Título, Categoría, Descripción y Nombre de usuario. Esto se podrá realizar siempre y cuando sea una imagen tomada desde un dispositivo móvil y que contenga metadatos, ya que de no ser así la plataforma no permitirá el update de la misma.



A continuación, en la página principal de Myphoto, el usuario puede consultar las fotos disponibles en las distintas categorías como también realizar una búsqueda de las imágenes de un usuario específico. O por otro lado, consultar en su perfil sus fotos subidas.



Una de las bases para la creación de mapas y georreferenciar las distintas fotos tomadas por los usuarios y subidas a la base de datos de la aplicación, es el tratamiento de los metadatos y la extracción de aquellos que aporten información acerca de la latitud y longitud exactas que geolocalizan cada foto:

- **Obtención y utilización de los metadatos de las imágenes.** Los metadatos de las fotos o imágenes se conciben como ciertos datos que describen una imagen. Entre ellos destacan algunos como el tamaño de la misma, el tipo de flash utilizado, la fecha en la que se ha sacado la foto y datos sobre el dispositivo empleado para tomarla,... Entre estos destacan aquellos que ofrecen información acerca de los parámetros de longitud y latitud que se han indexado a la foto durante su ejecución, permitiendo georreferenciarla e, incluso, determinar otros aspectos como la altitud utilizando otras herramientas.

Para la extracción de dichos metadatos se ha utilizado una función predefinida en el lenguaje php: `exif_read_data()`. Dicha función devuelve un array con los metadatos de la imagen o foto, conforme podemos ver en la siguiente captura de pantalla en la que se utiliza para una foto en la base de datos de Myphoto:

```
FILE.FileName: IMG_2688.JPG
FILE.FileDateTime: 1452516437
FILE.FileSize: 366771
FILE.FileType: 2
FILE.MimeType: image/jpeg
FILE.SectionsFound: ANY_TAG, IFD0, THUMBNAIL, EXIF, GPS
COMPUTED.html: width="1280" height="960"
COMPUTED.Height: 960
COMPUTED.Width: 1280
COMPUTED.IsColor: 1
COMPUTED.ByteOrderMotorola: 1
COMPUTED.ApertureFNumber: f/2.2
COMPUTED.Thumbnail.FileType: 2
COMPUTED.Thumbnail.MimeType: image/jpeg
IFD0.Make: Apple
IFD0.Model: iPhone 6
```

Como podemos observar, se ofrece un listado de características de un archivo imagen, tales como altura, anchura, tipo de imagen,... Para el acceso a cada una de ellas basta con escribir las siguientes sentencias en código php:

```
//Capturando metadatos de la imagen foto1.jpg en la variable $metadatos
$metadatos = exif_read_data(foto1.jpg)

//Capturando el tipo de dispositivo que ha tomando la imagen
$v_dispositivo = $v_exif["Make"];

//Capturando la fecha en la que se ha tomado la imagen
$v_datetime = $v_exif["DateTime"];
```

```
//Capturando el tipo de la imagen
    $v_tipoimagen = $v_exif["MimeType"];
```

```
//Capturando el nombre del archivo
    $v_nombreamarchivo = $v_exif["FileName"];
```

Entre las propiedades que devuelve la función *exif_read_data()* también se ofrecen los datos de georreferenciación en el caso de que el usuario haya habilitado la opción de gps en el dispositivo durante la captura de la imagen. Si dicha opción es activada, se incluirán además de los anteriores, los siguientes datos:

```
GPS.GPSLatitudeRef: N
GPS.GPSLatitude: Array
GPS.GPSLongitudeRef: W
GPS.GPSLongitude: Array
GPS.GPSAltitudeRef:
GPS.GPSAltitude: 72784/131
GPS.GPSTimeStamp: Array
```

En este caso en concreto, se indica latitud Norte(N) y longitud Oeste(W), además de otros datos sobre la latitud y la longitud contenidos en los arrays siguientes: *GPSLatitude: Array* y *GPSLongitude: Array*. Ambos van a permitir conocer la posición exacta dónde se ha tomado la imagen.

Muestra en la aplicación web de la localización en el mapa mediante la API.

Muestra en la aplicación web de los datos extraídos de los metadatos de cada imagen.

Titulo: Palmera autoctona canaria

Descripción : Palmera del ayuntamiento de San Cristobal de La Laguna

Detalles:	
Autor imagen:	JosueTC94
Latitud imagen:	28.488339299722
Longitud imagen:	-16.3161522
Tipo de imagen:	image/jpeg
Nombre del archivo:	20151010_140608.jpg
Tamaño de la imagen:	1575731
Fecha:	2015-12-08 19:35:23
Dispositivo:	LG Electronics

Conclusiones

Todos los días tomamos fotos con nuestro smartphone. Las fotos son una fuente de información y puede ser interesante utilizar esa información para fines comerciales. Los metadatos proporcionan aquello que el computador no puede ver, por ello resulta importante la correcta conservación y tratamiento de los mismos en una base de datos para posteriormente poder realizar un análisis de esos datos y convertirlos en información útil.

La idea de MyPhoto es atraer a usuarios para que compartan sus fotos en nuestra plataforma, dar un perfil social a la aplicación la hace más amena, animando a los usuarios a subir contenido a la misma. Una vez que se cuenta con la suficiente información se pueden sacar conclusiones comerciales, por ejemplo: Si encontramos que se sacan muchas fotos en un tramo concreto de una carretera con destino al Teide, en dirección al horizonte, quizás interese poner un mirador para turistas en la zona.

Es interesante también comentar las múltiples herramientas que se encuentran en internet para facilitar la representación de la información, numerosas librerías para representar información están escritas en javascript, este es el caso de HighCharts, Google Chart Tools o la API de Google Maps que permiten estilizar la información para que sea más agradable y eficaz a la hora de ser analizada.

Por otra parte, proyectos como este fomentan el trabajo colaborativo y ayudan a enfrentarse a nuevos retos de programación, tales como aprender a obtener metadatos a partir de imágenes, comprender cómo las distintas marcas utilizan los metadatos con una nomenclatura ligeramente distinta (a pesar de que existe un estándar EXIF), aprender a interactuar con Interfaces de Programación de Aplicaciones (APIs) y construir desde cero una aplicación web utilizando las tecnologías disponibles, como Ajax, Javascript, HTML5, CSS, PHP, Apache, MySQL, PhpMyAdmin, Github etc.

Bibliografía

- Manual PHP.
<http://php.net/manual/en/index.php>
<http://www.w3schools.com/php/>
- Google Maps for Work.
https://www.google.com/intx/en_uk/work/mapsearch/products/mapsapi.html?utm_source=HouseAds&utm_medium=cpc&utm_campaign=2015-Geo-EMEA-LCS-GEO-MAB-DeveloperTargetedHouseAds&utm_content=Developers&gclid=CJy104uYoMoCFQs8GwodvRUPLQ
- Blog con manual del API de Google Maps.
<http://www.desarrolloweb.com/manuales/desarrollo-con-api-de-google-maps.html>
- Twitter Bootstrap
<http://getbootstrap.com/2.3.2/>