

## Trabajo Práctico Nro 5. Librerías qrcodereader y php-image-resize

Grupo 4.1  
Sabrina Gaspart  
Norberto Gualda

### Introducción

Se desarrollo una aplicación que lee desde una imagen jpg un código qr. Para la lectura en si del código QR se utilizo la libreria qrcodereader. Esta librería tiene como limitación de tamaño de imagen aproximadamente 1000x1000 pixels. Con tamaños de imagen superiores aparecen errores no determinados. Ante esta situación para asegurar solo lecturas correctas o incorrectas de la imagen pero libre de errores al llamar la libreria se recurre a una segunda libreria que se llama php-image-resize para retamaño y unificar el formato de las imagenes que se suben.

### Instalación

La manera mas sencilla de instalar las dos librerias es via composer con los siguientes comandos en la carpeta del proyecto:

```
composer require libern/qrcode-reader  
composer require gumlet/php-image-resize
```

### Requerimientos y documentación de qrcode-reader

El enlace del repositorio es <https://github.com/libern/qrcode-reader>

Es importante que este activaba la extension fileinfo en php.ini =>extension=php\_fileinfo.dll  
Este detalle no esta incluido en la documentación por eso nos parece importante destacarlo.

Los requerimientos son  
PHP >= 5.6 y la libreria GD instalada.

Es una libreria de uso libre sin restricciones.

Funciona correctamente invocando el autoload de composer via:

```
require '../vendor/autoload.php';
```

y despues usando un alias:

```
use Libern\QRCodeReader\QRCodeReader;
```

Finalmente se puede invocar primero creando una instancia de la clase QRCodeReader y luego enviando como parámetro el archivo de imagen.

```
$QRCodeReader = new QRCodeReader();  
$qrcode_text = $QRCodeReader->decode($file);
```

Existe tambien la posibilidad de decodificar sobre stream de video pero no obtuvimos estabilidad en los resultados.

## Requerimientos y documentación de php-image-resize

El enlace del repositorio es <https://github.com/gumlet/php-image-resize>

La libreria funciona con archivos de imagen jpg y png, no asi gif.

Los requerimientos son  
PHP >= 5.6 y la libreria GD instalada.

Es una libreria de uso libre sin restricciones.

Funciona correctamente invocando el autoload de composer via:

```
require '../vendor/autoload.php';
```

y despues usando un alias:

```
use Gumlet\ImageResize;
```

Y luego la invocación es con este formato:

```
$image = new ImageResize($dir.$nombreArchivo);  
$image->resizeToBestFit(640, 640);  
$image->save($dir."scale.jpg");
```

Es una libreria de uso libre sin restricciones.

En la documentación encontramos varios métodos para retamañar con y sin resguardar el aspecto y una funcionalidad adicional via filtros que permite incrustar firmas a la imagen.

## Descripción del proyecto

La funcionalidad principal consiste en subir una imagen de cualquier tamaño en pixels (limitada a 2 mb) y esta será retamañada a un maximo de 640 pixels en su lado mas extenso.

Una vez realizado esto se procede a leer el codigo QR. Si la lectura es correcta o no incrusta la imagen sobre un back verde o rojo segun corresponda en un tamaño de 640 x 640 pixels.

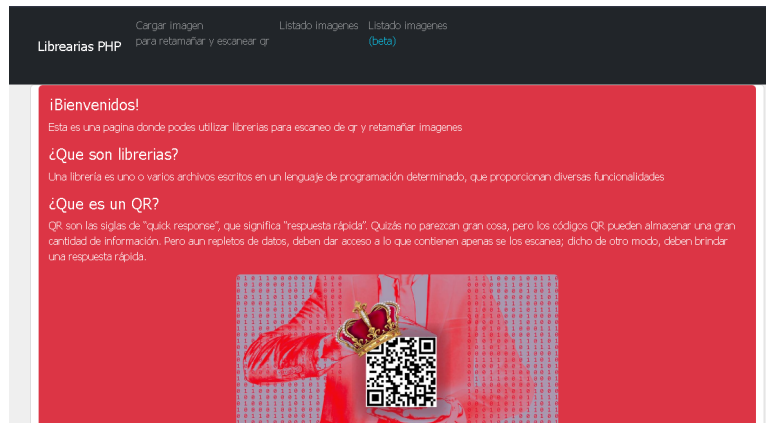
Esta imagen a su vez es incrustada en una imagen de 720x720 centrado, en la parte superior aparece el nombre de archivo (generado automaticamente por la fecha-hora de carga) y en la parte inferior, si hubiera sido posible, la lectura del codigo QR.

Toda la información está contenida en el archivo de imagen sin requerir el uso de base de datos.

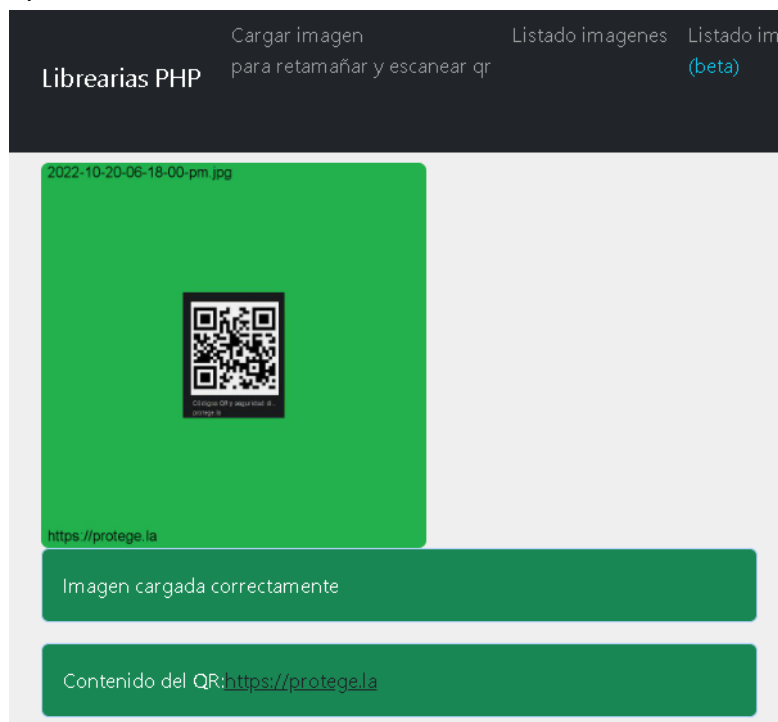
La opción para listar las imágenes escaneadas permite ampliarlas y ver -si hubieran- enlaces que se correspondan con los códigos QR. Como la aplicación solo guarda las imágenes se vuelven a invocar los métodos que leen el QR.

## Capturas de pantalla

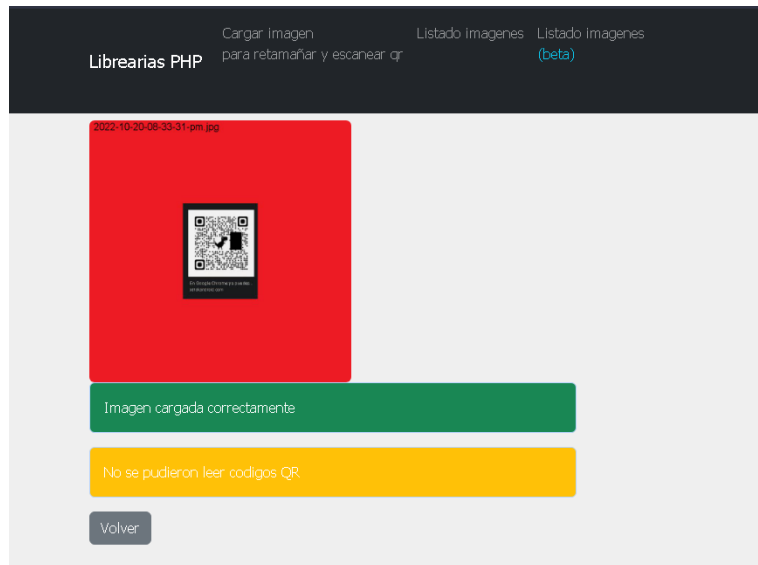
### Menu principal



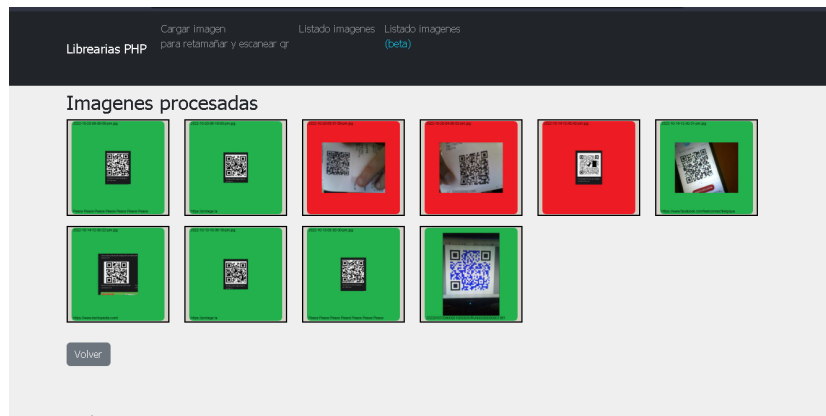
### Upload correcto



## Upload incorrecto



## Lista de imágenes retamafiadadas



## Detalle de imagen indicando el qr leído

