

TP5: LIBRERIAS

Integrantes

Salazar Daniel fai-962

Exequiel Calderon fai-4432

Jonathan Alveal fai-3581

Julian tolosa fai-3182

INTRODUCCIÓN:

Elegimos las siguientes 3 librerías para mostrar nuestra demo:

-Código QR – PHP

-SPOUT (EXCEL)

-GREGWAR/CAPTCHA

Empezaremos explicando cada una de las librerías en el siguiente orden :

1)QR-PHP 2)SPOUT(EXCEL) 3)CAPTCHA

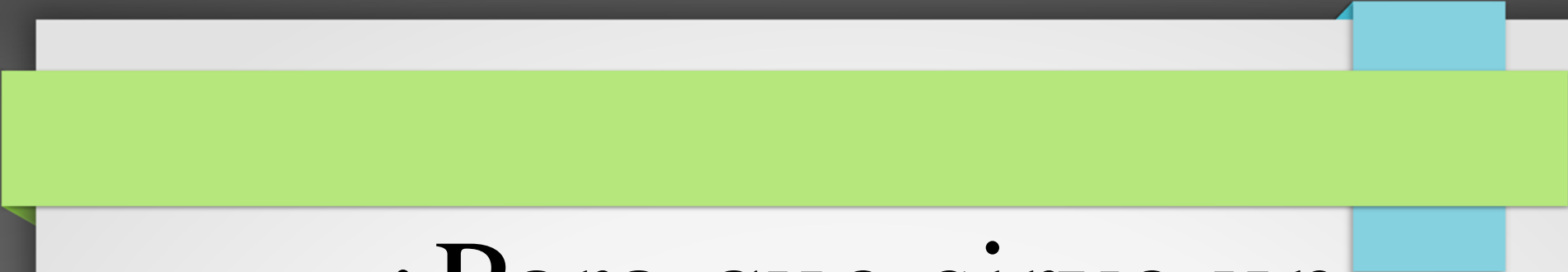
Código QR – PHP



¿Que es un Código QR?

Es un código de barras bidimensional
cuadrado que puede almacenar
datos de forma codificada

Code QR (del inglés Quick Response code,
"código de respuesta rápida")



¿Para que sirve un Código QR?

- Abrir la URL de una página Web o perfil social
- Leer un Texto
- Enviar un email
- Enviar un SMS
- Realizar un llamada telefónica
- Guardar un evento en la agenda
- Ubicar una posición geográfica
- etc



¿Como leer un Código QR?

APP que permita leer los códigos QR
desde tu dispositivo o teléfono móvil

Lectores webs online (se sube la imagen)

Tipos de Códigos QR



QR Code
model 1 model 2



Micro QR Code



iQR Code



SQRC



Frame QR



PHP QR Code

PHP QR Code es una biblioteca de código abierto (LGPL) para generar código QR

Proporciona una API para crear imágenes de código QR (PNG, JPEG y SVG)

Implementado puramente en PHP
(excepto GD2)

Implementación

Descargar la librería desde
<http://phpqrcode.sourceforge.net/>

Incluir

```
include('phpqrcode/qrlib.php');
```

Método estático

```
qrcode::png($contenido, $nombreArchivo, $nivel,  
             $tamaño, $margen);
```



Contenido

Según sea el tipo de dato lleva un formato específico

Hay distintos tipos de contenidos de datos

Nombre del Archivo

Se especifica el directorio y el nombre para la imagen generada

Este atributo es opcional, sino se especifica este parámetro, la imagen se mostrará en el navegador

Level

Capacidad de corrección de errores

- L: 7% (Baja) (valor por defecto)
 - M: 15% (Media)
 - Q: 25% (Alta)
 - H: 30% (Muy Alta)
-
- Porcentaje de daño que puede contener

Tamaño

Cada pixel de la imagen se multiplicará por este valor numérico (Píxeles Virtuales)

Valor por defecto: 3

Margen

Es el espacio entre el contorno de la imagen y la zona de los puntos

Se especifica un valor numérico (Píxeles Virtuales)

Valor por defecto: 4

Referencias

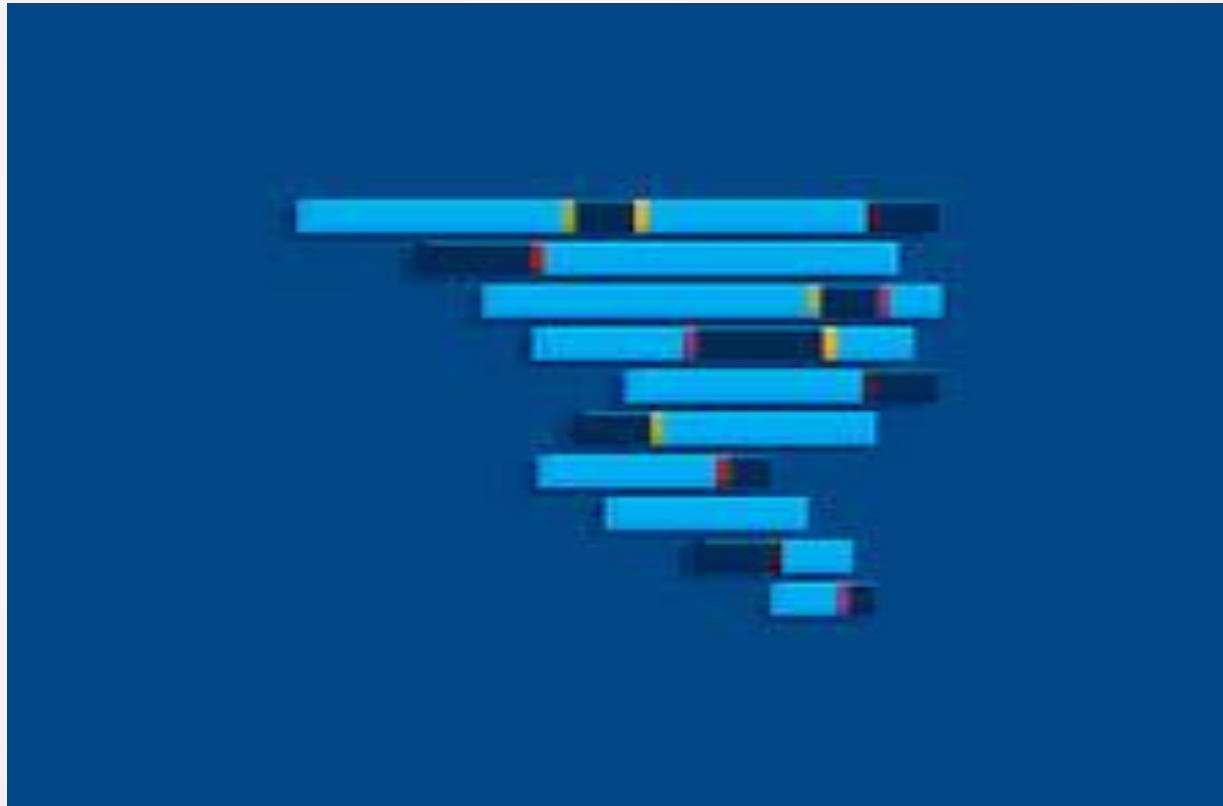
<http://phpqrcode.sourceforge.net/>

<https://www.qrcode.com/en/about/>

https://es.wikipedia.org/wiki/C%C3%B3digo_QR

<http://www.codigos-qr.com/lector-qr-online/>

SPOUT (EXCEL)





¿PARA QUE SIRVE SPOUT(EXCEL)?

Para leer y escribir hojas de cálculo

rápidamente y a escala

¿Por qué utilizar SPOUT(EXCEL)?

- ❖ SPOUT es capaz de procesar Archivos de cualquier tamaño
- ❖ SPOUT necesita solo 3 MB de memoria para procesar cualquier archivo
- ❖ El mecanismo de transmisión de SPOUT LO HACE increíblemente rápido
- ❖ La API de SPOUT es amigable para los desarrolladores

¿Qué tipos de formatos acepta Spout(Excel)?

SPOUT admite 3 tipos de hojas de cálculo: XLSX , ODS y CSV .

SPOUT proporciona una API simple y unificada para leer o crear estos diferentes tipos de hojas de cálculo.

REQUISITOS PREVIOS PARA LA INSTALACIÓN

- ☐ PHP versión 7.2 o superior
- ☐ Extensión PHP ext-zip habilitada
- ☐ Extensión PHP ext-xmlreader habilitada



Instalación

Dirigirse al siguiente url:

<https://opensource.box.com/spout/>

Ahí encontraremos 2 formas de instalarlo manualmente y por composer que es la recomendada.

METODOS DE INSTALACIONES

- ❖ **Por composer**: directamente instalar el composer y ejecute el siguiente comando:

\$composer require box/spout;

- ❖ **Manualmente**: 1) Descargar el código fuente de <https://github.com/box/spout/releases>

2)Extraiga el contenido descargado en su proyecto.

3) Agregue este código al controlador superior (por ejemplo, index.php) o donde sea más apropiado:

```
require_once '[PATH/TO]/src/Spout/Autoloader/autoload.php';
```

USO BASICO

LECTOR: Independientemente del tipo de archivo, la interfaz para leer un archivo es siempre la misma:

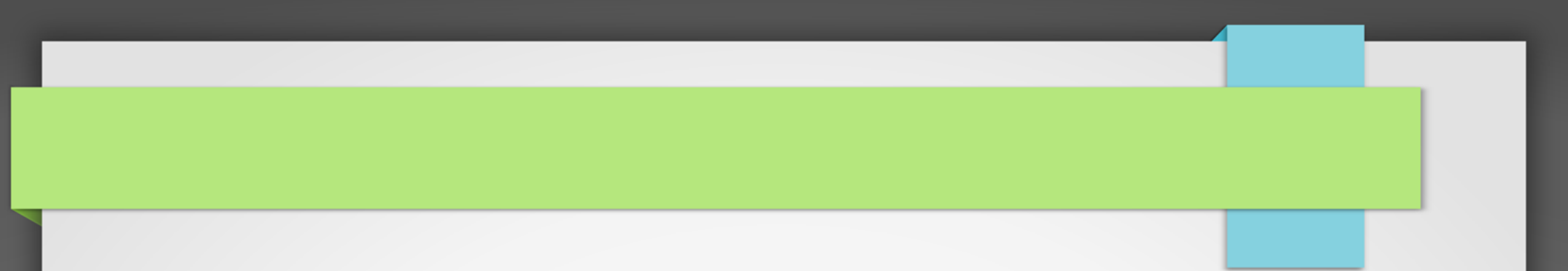
```
use box\Spout\Reader\Common\Creator\ReaderEntityFactory;

$reader = ReaderEntityFactory :: createReaderFromFile('/patch/to/file.ext');

$reader ->open($filePath) ;

foreach ($reader->getSheetIterator () as $sheet) {
    foreach ($sheet ->getRowIterator() as $row) {
        $cells = $row ->getCells();
        .....
    }
}

$reader ->close();
```

ESCRITOR : Al igual que con el lector, existe una interfaz común para escribir datos en un archivo.

```
use box\Spout\Writer\Common\Creator\WriterEntityFactory;
```

```
use box\Spout\Common\Entity\Row;
```

```
$writer = WriterEntityFactory :: createXLSXWriter();
```

REFERENCIAS

<https://opensource.box.com/spout/getting-started/>

<https://opensource.box.com/spout/docs/>

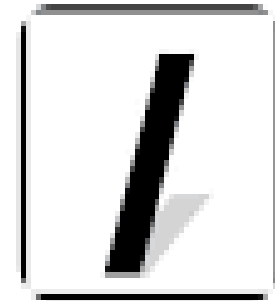
<https://opensource.box.com/spout/guides/>

<https://opensource.box.com/spout/faq/>

GREGWAR/CAPTCHA

Gregwar/Captcha

PHP Captcha library



 27

Contributors

 36

Issues

 2k

Stars

 314

Forks





¿PARA QUE SIRVE EL CAPTCHA?

Un CAPTCHA te ayuda a protegerte del spam y del descifrado de contraseñas pidiéndote que completes una simple prueba que demuestre que eres humano y no un ordenador que intenta acceder a una cuenta protegida con contraseña.

Tipos de captcha

- Basados en texto o **CAPTCHAs textuales**
- Basados en cálculos matemáticos
- Basados en juegos de palabras
- Basados preguntas lógicas
- Basados en redes sociales
- Basados en límite de tiempo
- CAPTCHAs visuales o gráficos
- Basados en audio o **CAPTCHAs auditivos**
- **Captchas lúdicos**
- Basados en vídeos
- Basados en análisis de conducta o comportamiento

INSTALACIÓN

Por composer: directamente instalar el composer y ejecute el siguiente comando:

```
{  
    ...  
    "requiere": {  
        "gregwar/captcha": "1.*"  
    }  
}
```

Como usar gregwar/captcha

Para crear un captcha con captchaBuilder:

```
<?php
    use Gregwar\Captcha\CaptchaBuilder;

    $builder = new CaptchaBuilder;

    $builder->build();
```

Para guardar un archivo:

```
<?php
    $builder->save('out.jpg');
```

Para enviar el archivo directamente:

```
<?php
    header('Content-type: image/jpeg');

    $builder->output();
```

Para insertarlo ah una pagina html:

```
<img src= "<?php echo $builder->inline(); ?>" />
```

Referencias

<https://github.com/Gregwar/Captcha#readme>

<https://es.wikipedia.org/wiki/Captcha>

<https://blog.hubspot.es/website/que-es-captcha>