

## **PHPExcel Developer Documentation**

# 1. Contenido

## [PHPExcel Developer Documentation](#)

### [1. Contenido](#)

### [2. Requisitos previos](#)

- [2.1. Requisitos de software](#)
- [2.2. Instrucciones de instalación](#)
- [2.3. Introducción](#)
- [2.4. Herramientas y enlaces útiles](#)
  - [2.4.1. OpenXML / SpreadsheetML](#)
  - [2.4.2. Preguntas frecuentes](#)
  - [2.4.3. Tutoriales](#)

### [3. Arquitectura](#)

- [3.1. Schematic](#)
- [3.2. Cargador perezoso](#)
- [3.3. Hoja de cálculo en la memoria](#)
- [3.4. Lectores y escritores](#)
- [3.5. Fluent interfaces](#)

### [4. Crear una hoja de cálculo](#)

- [4.1. PHPExcel la clase](#)
  - [4.1.1. Cargando un libro desde un archivo](#)
  - [4.1.2. Creando un nuevo libro](#)
- [4.2. De configuración](#)
  - [4.2.1. Célula de almacenamiento en caché](#)
  - [4.2.2. Idioma/Locale](#)
- [4.3. Borrar un libro de la memoria](#)
- [4.4. Hojas de trabajo](#)
  - [4.4.1. Agregar una nueva hoja de cálculo](#)
  - [4.4.2. Copiar hojas de cálculo](#)
  - [4.4.3. Quitar una hoja de cálculo](#)
- [4.5. Acceso a las células](#)
  - [4.5.1. Establecer un valor de celda por coordenadas](#)
  - [4.5.2. Recuperar una celda por coordenadas](#)
  - [4.5.3. Establecer un valor de celda por columna y fila](#)
  - [4.5.4. Recuperar una celda por columna y fila](#)
  - [4.5.5. Células Looping](#)
  - [4.5.6. Usando ligantes de valor para facilitar la entrada de datos](#)
- [4.6. PHPExcel recetas](#)
  - [4.6.1. Configuración de metadatos de una hoja de cálculo](#)
  - [4.6.2. Ajuste de la hoja activa de una hoja de cálculo](#)
  - [4.6.3. Escribir una fecha u hora en una celda](#)
  - [4.6.4. Escribir una fórmula en una celda](#)
  - [4.6.5. Opciones de localización para las fórmulas](#)
  - [4.6.6. Escribir un carácter de nueva línea "\n" en una celda \(ALT + "Enter"\)](#)
  - [4.6.7. Establecer explícitamente el tipo de datos de una celda](#)
  - [4.6.8. Cambiar una celda en una URL cliqueable](#)
  - [4.6.9. Ajuste de tamaño y orientación de página de una hoja de cálculo](#)
  - [4.6.10. Page Setup: opciones de escala](#)
  - [4.6.11. Los márgenes de página](#)
  - [4.6.12. Centro de una página horizontal/vertical](#)
  - [4.6.13. Configuración de la impresión encabezado y el pie de una hoja de cálculo](#)
  - [4.6.14. Saltos impresión de ajuste en una fila o columna](#)
  - [4.6.15. Mostrar u ocultar la cuadrícula cuando se imprime](#)
  - [4.6.16. Ajuste filas o columnas para repetir en el superior izquierdo](#)
  - [4.6.17. Especificar área de impresión](#)
  - [4.6.18. Formato celdas](#)
  - [4.6.19. Número de formatos](#)
  - [4.6.20. Alineación y texto](#)
  - [4.6.21. Establecer el estilo predeterminado de un libro](#)

- [4.6.22. Estilo bordes de celda](#)
- [4.6.23. Una célula de formato condicional](#)
- [4.6.24. Agregar un comentario a una celda](#)
- [4.6.25. Aplicar Autofiltro a un rango de celdas](#)
- [4.6.26. Seguridad de ajuste en una hoja de cálculo](#)
- [4.6.27. Validación de datos de configuración en una celda](#)
- [4.6.28. Ajuste ancho de una columna](#)
- [4.6.29. Mostrar u ocultar una columna](#)
- [4.6.30. Grupo/contorno una columna](#)
- [4.6.31. Ajuste de altura de una fila](#)
- [4.6.32. Mostrar u ocultar una fila](#)
- [4.6.33. Grupo/contorno una fila](#)
- [4.6.34. Merge/Separar celdas](#)
- [4.6.35. Insertar filas o columnas](#)
- [4.6.36. Agregar un dibujo a una hoja de cálculo](#)
- [4.6.37. Imágenes de la lectura de una hoja de cálculo](#)
- [4.6.38. Agregar texto a una celda](#)
- [4.6.39. Definir un rango con nombre](#)
- [4.6.40. Redirigir la salida a explorador web del cliente](#)
- [4.6.41. Ajuste el ancho de columna predeterminado](#)
- [4.6.42. Ajuste de la altura de fila predeterminado](#)
- [4.6.43. Agrega una GD dibujo a una hoja de cálculo](#)
- [4.6.44. Hoja de ajuste nivel de zoom](#)
- [4.6.45. Color de ficha de hoja](#)
- [4.6.46. Crear hojas de cálculo en un libro](#)
- [4.6.47. Hojas ocultas \(Estados de hoja\)](#)
- [4.6.48. Hoja de derecha a izquierda](#)
- [5. Escénicas fórmulas cálculos](#)
  - [5.1. Utilizando el motor de cálculo PHPExcel](#)
  - [5.2. Conoce limitaciones](#)
    - [5.2.1. Precedencias de los operadores](#)
    - [5.2.2. Fórmulas que involucran números y texto](#)
- [6. De lectura y escritura al archivo](#)
  - [6.1. PHPExcel\\_IOFactory](#)
    - [6.1.1. PHPExcel\\_Reader\\_IReader crear usando PHPExcel\\_IOFactory](#)
    - [6.1.2. PHPExcel\\_Writer\\_IWriter crear usando PHPExcel\\_IOFactory](#)
  - [6.2. Formato de archivo de Excel 2007 \(SpreadsheetML\)](#)
    - [6.2.1. PHPExcel\\_Reader\\_Excel2007](#)
    - [6.2.2. PHPExcel\\_Writer\\_Excel2007](#)
  - [6.3. Formato de archivo de Excel 5 \(BIFF\)](#)
    - [6.3.1. PHPExcel\\_Reader\\_Excel5](#)
    - [6.3.2. PHPExcel\\_Writer\\_Excel5](#)
  - [6.4. Formato de archivo XML de Excel 2003](#)
    - [6.4.1. PHPExcel\\_Reader\\_Excel2003XML](#)
  - [6.5. Enlace simbólico \(SYLK\)](#)
    - [6.5.1. PHPExcel\\_Reader\\_SYLK](#)
  - [6.6. Oficina abierta/Libre \(.ods\)](#)
    - [6.6.1. PHPExcel\\_Reader\\_OOCalc](#)
  - [6.7. CSV \(valores separados por comas\)](#)
    - [6.7.1. PHPExcel\\_Reader\\_CSV](#)
    - [6.7.2. PHPExcel\\_Writer\\_CSV](#)
  - [6.8. HTML](#)
    - [6.8.1. PHPExcel\\_Reader\\_HTML](#)
    - [6.8.2. PHPExcel\\_Writer\\_HTML](#)
  - [6.9. PDF](#)
    - [6.9.1. PHPExcel\\_Writer\\_PDF](#)
  - [6.10. Archivos Excel generación de plantillas \(leer, modificar, escribir\)](#)
- [7. Créditos](#)
- [Apéndice A: claves de matriz válida para estilo applyFromArray\(\)](#)

## 2. Requisitos previos

### 2.1. Requisitos de software

El siguiente software es necesaria para desarrollar usando PHPEXcel:

- » PHP versión 5.2.0 o posterior
- » PHP extensión php\_zip activado \*)
- » Php\_xml de extensión PHP habilitado
- » Php\_gd2 extensión PHP activado (si no compilado)

\*) php\_zip sólo es necesario PHPEXcel\_Reader\_Excel2007, PHPEXcel\_Writer\_Excel2007 y PHPEXcel\_Reader\_OOCalc. En otras palabras, si necesitas PHPEXcel para manejar archivos.xlsx o .ods necesitará la extensión zip, pero lo contrario no.  
Puede eliminar esta dependencia para escribir archivos Excel2007 (aunque aún no para lectura) mediante el uso de la biblioteca PCLZip que se lía con PHPEXcel. Ver la sección de preguntas frecuentes de este documento (2.4.2) para más detalles sobre esto. PCLZip tiene una dependencia en la extensión de zlib de PHP ser habilitado.

### 2.2. Instrucciones de instalación

La instalación es muy fácil: copiar el contenido de la carpeta Clases en cualquier ubicación dentro de los directorios fuente de aplicación.

*Ejemplo:*

Si la carpeta raíz del web es /var/www/quiza quieras crear una subcarpeta llamada /var/www/clases/ y copie los archivos en esa carpeta y terminas con archivos:

```
/var/www/clases/PHPEXcel.php  
/var/www/clases/PHPEXcel/Calculation.php  
/var/www/clases/PHPEXcel/Cell.php  
...
```

### 2.3. Getting started

Una buena manera de empezar es ejecutar algunas de las pruebas incluidas en la descarga.

Copiar la carpeta "Ejemplos" junto a la carpeta "Clases" de arriba para que usted termina con:

```
/ var/www/Examples/01simple.php  
/ var/www/Examples/02types.php  
...
```

Empieza a correr las pruebas señalando su navegador a los scripts de prueba:

```
http://example.com/ Examples/01simple.php  
http://example.com/ Examples/02types.php  
...
```

**Nota:** Puede ser necesario modificar el include / requieren declaraciones al principio de cada uno de los scripts de prueba si la carpeta "Clases" de arriba se llama diferentemente.

### 2.4. Herramientas y enlaces útiles

Hay algunos enlaces y herramientas que son muy útiles en el desarrollo de utilizar PHPEXcel. Por favor consulte las [páginas PHPEXcel CodePlex](#) para una versión de actualización de la lista siguiente.

#### 2.4.1. OpenXML / SpreadsheetML

- » Documentación de formato de archivo  
[http://www.Ecma-International.org/news/TC45\\_current\\_work/TC45\\_available\\_docs.htm](http://www.Ecma-International.org/news/TC45_current_work/TC45_available_docs.htm)
- » OpenXML explicó e-book  
<http://openxmldeveloper.org/articles/1970.aspx>

- » **Paquete de compatibilidad de Microsoft Office para Word, Excel y PowerPoint 2007 formatos de archivo**  
<http://www.Microsoft.com/downloads/details.aspx?FamilyId=941b3470-3ae9-4aee-8f43-c6bb74cd1466&displaylang=en>
- » **OpenXML paquete Explorer**  
<http://www.CodePlex.com/PackageExplorer/>

## 2.4.2. Preguntas frecuentes

La página de preguntas frecuentes actualizada para PHPEXcel puede encontrarse en <http://www.codeplex.com/PHPEXcel/Wiki/View.aspx?title=FAQ&referringTitle=Requirements>

.

### **Parece haber un problema con la codificación de caracteres...**

Es necesario utilizar para todos los textos en PHPEXcel de codificación UTF-8. Si la secuencia de comandos utiliza una codificación diferente entonces usted puede convertir esos textos con iconv() de PHP o mb\_convert\_encoding() funciones.

### **PHP se queja de no ser encontrado ZipArchive**

Asegúrate de cumplir todos los requisitos, especialmente php\_zip extensión debe estar habilitado.

La clase ZipArchive es sólo necesaria al leer o escribir los formatos que utilizan compresión Zip (Excel2007 y OoCalc). Desde la versión 1.7.6 la biblioteca PCLZip ha estado liada con PHPEXcel como alternativa a la clase ZipArchive.

Esto puede activar llamando al:

```
PHPEXcel_Settings::setZipClass(PHPEXcel_Settings::PCLZIP);
```

antes de llamando al guardar método del escritor Excel2007.

Puede volver a usar ZipArchive llamando al:

```
PHPEXcel_Settings::setZipClass(PHPEXcel_Settings::ZIPARCHIVE);
```

En la actualidad, esto sólo le permite escribir Excel2007 archivos sin necesidad de ZipArchive (no para leer Excel2007 o OoCalc)

### **Excel 2007 no se puede abrir el archivo generado por PHPEXcel\_Writer\_2007 en Windows**

*"Excel encontró contenido ilegible en ' \*.xlsx '. ¿Quieres recuperar el contenido de este libro? Si confías en el origen de este libro, haga clic en sí."*

Algunas versiones antiguas de la 5.2. xphp\_zip extensión en Windows contienen un error al crear archivos ZIP. Funcionará en todo momento la versión que se puede encontrar en <http://snaps.php.net/win32/php5.2-win32-latest.zip>.

Alternativamente, la actualización a por lo menos PHP 5.2.9 debe solucionar el problema.

Si no puede encontrar una copia limpia de ZipArchive, entonces puede utilizar la biblioteca PCLZip como alternativa al escribir archivos de Excel2007, como se describió anteriormente.

### **Fatal error: tamaño de la memoria de xxx bytes agotados (intentado asignar yyy bytes) puede entrar zzz en línea aaa**

PHPEXcel posee una representación "en memoria" de una hoja de cálculo, así que está restringido por limitaciones de memoria de PHP. Puede aumentar la memoria a disposición de

PHP editando el valor de la Directiva memory\_limit en tu archivo php.ini, o usando ini\_set ('memory\_limit', '128 M') dentro de su código (ISP).

Algunos lectores y escritores son más rápidos que otros, y también usan diferentes cantidades de memoria. Usted puede encontrar indicios del uso de memoria y rendimiento relativo de los diferentes lectores y escritores, sobre las diferentes versiones del PHPEXcel, en el [Foro de discusión](#).

Si ya has aumentado memoria hasta un máximo, o no puedes cambiar tu límite de memoria, entonces [esta discusión](#) en el Consejo describe algunos de los métodos que se pueden aplicar para reducir el uso de la memoria de tus scripts usando PHPEXcel.

### ¿Protección en mi hoja de cálculo no funciona?

Al hacer uso de cualquiera de las funciones de protección de hoja de cálculo (por ejemplo gama protección celular, que prohíbe eliminar filas,...), asegúrese de habilitar seguridad de hoja de cálculo. Esto puede ser por ejemplo así:

```
$objPHPExcel -> getActiveSheet() -> getProtection() -> setSheet(true);
```

### Función X no está trabajando con PHPEXcel\_Reader\_Y / PHPEXcel\_Writer\_Z

No todas las funciones de PHPEXcel se implementan en todo del lector / clases de escritor. Esto es debido principalmente a las bibliotecas subyacentes no apoyar una característica específica o no haber implementado una característica específica.

Por ejemplo Autofiltro no está implementado en pera Spreadsheet\_Excel\_writer, que es la base de nuestro escritor Excel5.

Poco a poco estamos construyendo una lista de características, junto con los diferentes lectores y escritores que apoyan, en el archivo "funcionalidad cruzada-Reference.xls" en la carpeta /Documentation.

### ¿Las fórmulas no parecen calculados en Excel2003 usando el paquete de compatibilidad?

Este es un comportamiento normal del paquete de compatibilidad, Excel2007 muestra esto correctamente. Utilice PHPEXcel\_Writer\_Excel5 si realmente necesita los valores calculados, o fuerza de recálculo en Excel2003.

### Establecer el ancho de columna no es fiable al 100%

Tratando de establecer el ancho de columna, he experimentado un problema. Cuando abra el archivo en Excel, la anchura real es 0,71 inferior debe ser.

La respuesta corta es que PHPEXcel utiliza una medida donde está incluido el acolchado. Vea la sección: "Ajuste de ancho de una columna" para más detalles.

### ¿Cómo utilizo PHPEXcel con mi marco

- » Hay algunas instrucciones para utilizar PHPEXcel con Joomla en el [tablero de mensajes de Joomla](#)
- » Una página de consejos sobre cómo utilizar [PHPEXcel en Yii framework](#)
- » [La panadería](#) tiene algunas clases auxiliares para leer y escribir con PHPEXcel dentro de CakePHP
- » Integración [PHPEXcel en Kohana](#) <http://www.flynsarmy.com/2010/07/phpexcel-module-for-kohana-3/> y [Интеграция PHPEXcel и Kohana Framework](#)
- » Utilizar [PHPEXcel con TYPO3](#)

### Joomla Autoloader interfiere con PHPEXcel Autoloader

Gracias a peterrlynch por los siguientes consejos para resolver problemas entre el [autoloader PHPEXcel y Joomla Autoloader](#)

### 2.4.3. Tutoriales

- » PHPExcel inglés tutorial  
<http://openxmldeveloper.org>
- » Tutorial francés PHPExcel  
<http://g-ernaelsten.developpez.com/Tutoriels/excel2007/>
- » Ruso PHPExcel Blog anuncios  
<http://www.Web-Junior.net/sozdanie-Excel-fajjlov-s-pomoshhyu-PHPExcel/>
- » Una introducción japonés PHPExcel  
<http://Journal.MYCOM.co.jp/articles/2009/03/06/PHPExcel/index.html>

## 3. Arquitectura

### 3.1. Schematical

### 3.2. Lazy Loader

PHPExcel implementa un autoloader o "lazy loader", que significa que no es necesario incluir todos los archivos dentro de PHPExcel. Sólo es necesario incluir el archivo de clase PHPExcel inicial, luego el autocargador incluirá otros archivos de clase como y cuando sea necesario, tan sólo los archivos que en realidad son requeridos por su guión se cargará en la memoria PHP. El principal beneficio de esto es que reduce la huella de la memoria del PHPExcel sí mismo, para que utiliza menos memoria PHP.

Si sus propios scripts ya definen una función autoload, entonces esto puede sobrescribirse por la función autoload PHPExcel. Por ejemplo, si usted tiene:

```
función __autoload($class) {}  
    ...  
}
```

Hacer esto en su lugar:

```
función myAutoload($class) {}  
    ...  
}  
spl_autoload_register('myAutoload');
```

El autocargador entonces coexistirán con el autocargador de PHPExcel.

### 3.3. Hoja de cálculo en la memoria

Arquitectura de PHPExcel está construida de una manera que puede servir como una hoja de cálculo en la memoria. Esto significa que, si uno quisiera crear una vista basada en la web de una hoja de cálculo que se comunica con el modelo de objetos de PHPExcel, sólo tendría que escribir el código de front-end.

Como software de escritorio de hoja de cálculo, PHPExcel representa una hoja de cálculo que contiene una o más hojas, que contienen células con datos, fórmulas, imágenes...



### **3.4. Lectores y escritores**

Por su parte, PHPExcel no proporciona la funcionalidad para leer o escribir en una hoja de cálculo persistentes (en disco o en una base de datos). Para proporcionar esta funcionalidad, pueden utilizarse lectores y escritores.

De forma predeterminada, el paquete PHPExcel proporciona algunos lectores y escritores, entre ellos uno para el formato de hoja de cálculo XML abierto (también conocido como formato de archivo de Excel 2007). No está limitado a los predeterminados lectores y escritores, como eres libre para implementar la interfaz `PHPExcel_Writer_IReader` y `PHPExcel_Writer_IWriter` en una clase personalizada.

### 3.5. *Fluent interfaces*

PHPExcel soporta interfaces con fluidez en la mayoría de las localizaciones. Esto significa que usted puede fácilmente "cadena" llamadas a métodos específicos sin necesidad de una nueva declaración de PHP. Por ejemplo, tomemos el siguiente código:

```
$objPHPExcel -> getProperties() -> setCreator ("Martín Balliauw");
$objPHPExcel -> getProperties() -> setLastModifiedBy ("Martín Balliauw");
$objPHPExcel -> getProperties() -> setTitle ("Office 2007 XLSX prueba
documento");
$objPHPExcel -> getProperties() -> setSubject ("Office 2007 XLSX prueba
documento");
$objPHPExcel -> getProperties() -> descripciónDelConjunto ("prueba documento
para Office 2007 XLSX, generado mediante clases PHP.");
$objPHPExcel -> getProperties() -> setKeywords ("office 2007 openxml php");
$objPHPExcel -> getProperties() -> setCategory ("resultado archivo de
prueba");
```

Esto puede ser reescrita como:

```
$objPHPExcel -> getProperties()
-> setCreator ("Martín Balliauw")
-> setLastModifiedBy ("Martín Balliauw")
-> setTitle ("Office 2007 XLSX prueba documento")
-> setSubject ("Office 2007 XLSX prueba documento")
-> descripciónDelConjunto ("documento de prueba para Office 2007 XLSX,
generado mediante clases PHP.")
-> setKeywords ("office 2007 openxml php")
-> setCategory ("resultado archivo de prueba");
```



#### **Usando interfaces con fluidez es no necesario**

Fluent interfaces se han implementado para proveer una API de programación conveniente. Uso de las mismas no es necesaria, pero puede hacer más fácil de leer y mantener tu código. También puede mejorar rendimiento, como está reduciendo el número total de llamadas a métodos PHPExcel.

## 4. Crear una hoja de cálculo

### 4.1. PHPExcel la clase

La clase PHPExcel es el núcleo de PHPExcel. Contiene referencias al contenido hojas de cálculo, configuración de seguridad del documento y metadatos del documento.

Para simplificar el concepto PHPExcel: la clase PHPExcel representa su libro. Por lo general, va a crear un libro en uno de dos maneras, ya sea por cargarlo desde un archivo de hoja de cálculo, o crearlo manualmente. Una tercera opción, aunque menos comúnmente utilizado, es la clonación un libro existente que ha sido creado usando uno de los dos métodos anteriores.

#### 4.1.1. Cargando un libro desde un archivo

Detalles de los hoja de cálculo diferentes formatos compatibles, y las opciones disponibles para leerlos en un objeto PHPExcel describen completamente en el documento "PHPExcel documentación - lectura de hoja de cálculo archivos de usuario".

```
$inputFileName = './sampleData/example1.xls';

/** Load $inputFileName to a PHPExcel Object */
$objPHPExcel = PHPExcel_IOFactory::load($inputFileName);
```

#### 4.1.2. Creación de un nuevo libro

Si desea crear un nuevo libro, en lugar de cargar un archivo, entonces simplemente necesita instanciarlo como un nuevo objeto PHPExcel.

```
/** Create a new PHPExcel Object */
$objPHPExcel = new PHPExcel();
```

Siempre se creará un nuevo libro con una sola hoja de cálculo.

## 4.2. Opciones de configuración

Una vez que usted ha incluido los archivos PHPExcel en tu script, pero antes de crear instancias de un objeto PHPExcel o cargar un archivo de libro, hay una serie de opciones de configuración que se puede establecer que afectará el comportamiento posterior del guión.

### 4.2.1. Almacenamiento en caché de la célula

PHPExcel utiliza un promedio de cerca de 1k/celda en sus hojas, grandes libros pueden usar rápidamente la memoria disponible. Almacenamiento en caché de la célula proporciona un mecanismo que permite PHPExcel mantener los objetos de la célula en un tamaño más pequeño de la memoria, disco, o en APC, memcache o Wincache, en lugar de la memoria PHP. Esto le permite reducir el uso de la memoria de los grandes libros, aunque a un costo de velocidad para acceder a los datos de la célula.

De forma predeterminada, PHPExcel conserva todos los objetos de la célula en la memoria, pero puede especificar alternativas. Para habilitar el almacenamiento en caché de la célula, debe llamar al método PHPExcel\_Settings::setCacheStorageMethod() en el método de almacenamiento en caché que desea utilizar.

```
$cacheMethod = PHPExcel_CachedObjectStorageFactory::cache_in_memory;
PHPExcel_Settings::setCacheStorageMethod($cacheMethod);
```

setCacheStorageMethod() devolverá un booleano verdadero éxito, false si falla (por ejemplo si tratando de caché a APC cuando APC no está habilitada).

Una caché independiente para cada hoja de cálculo individual se mantiene y se crea automáticamente cuando la hoja de cálculo es instanciado basado en el método de almacenamiento en caché y ajustes que se han configurado. No puede cambiar los ajustes de configuración una vez que usted ha comenzado a leer un libro, o han creado la primera hoja de cálculo.

Actualmente, están disponibles los siguientes métodos de almacenamiento en caché.

PHPExcel_CachedObjectStorageFactory::cache_in_memory;
El valor por defecto. Si usted no inicializar cualquier método de almacenamiento en caché, entonces este es el método que utilizará PHPEXcel. Celda objetos se mantienen en la memoria PHP como en la actualidad.
PHPExcel_CachedObjectStorageFactory::cache_in_memory_serialized;
Empleando este método de almacenamiento en caché, las células se llevan a cabo en memoria PHP como una matriz de objetos serializados, que reduce la huella de memoria con rendimiento mínimo.
PHPExcel_CachedObjectStorageFactory::cache_in_memory_gzip;
Como cache_in_memory_serialized, este método tiene células en memoria PHP como una matriz de objetos serializados, pero comprimido con gzip para reducir el uso de la memoria aún más, aunque el acceso a leer o a escribir una celda es ligeramente más lenta.
PHPExcel_CachedObjectStorageFactory::cache_igbinary;
Utiliza la extensión de PHP igbinary (si está disponible) para serializar objetos de la célula de memoria. Esto es normalmente más rápido y utiliza menos memoria que serialización estándar de PHP, pero no está disponible en la mayoría de entornos hosting.
PHPExcel_CachedObjectStorageFactory::cache_to_discISAM;
<p>Cuando se utiliza cache_to_discISAM que todas las células se llevan a cabo en un archivo de disco temporal, con sólo un índice para su ubicación en el archivo mantenido en memoria PHP. Esto es más lento que cualquiera de los métodos de cache_in_memory, pero reduce significativamente la huella de la memoria. De forma predeterminada, PHPEXcel utilizará el directorio temp de PHP para el archivo de caché, pero puede especificar un directorio diferente al inicializar cache_to_discISAM.</p> <pre><code>\$cacheMethod = PHPEXcel_CachedObjectStorageFactory:: cache_to_discISAM; \$cacheSettings = array ('dir' =&gt; '/usr/local/tmp' ); PHPExcel_Settings::setCacheStorageMethod (\$cacheMethod, \$cacheSettings);</code></pre> <p>El archivo de disco temporal se borra automáticamente cuando termina la secuencia de comandos.</p>
PHPExcel_CachedObjectStorageFactory::cache_to_phpTemp;
<p>Como cache_to_discISAM, cuando se usa cache_to_phpTemp que todas las células se llevan a cabo en la corriente php://temp I/O, con sólo un índice para su ubicación mantenida en memoria PHP. En PHP, la envoltura php://memory almacena los datos en la memoria: php://temp se comporta semejantemente, pero utiliza un archivo temporal para almacenar los datos cuando se alcanza un cierto límite de memoria. El valor predeterminado es 1 MB, pero esto puede cambiar cuando inicializar cache_to_phpTemp.</p> <pre><code>\$cacheMethod = PHPEXcel_CachedObjectStorageFactory:: cache_to_phpTemp; \$cacheSettings = array ('memoryCacheSize' =&gt; 8 MB' ); PHPExcel_Settings::setCacheStorageMethod (\$cacheMethod, \$cacheSettings);</code></pre>

El archivo `php://temp` se borra automáticamente cuando termina la secuencia de comandos.

#### `PHPExcel_CachedObjectStorageFactory::cache_to_apc;`

Cuando se utiliza `cache_to_apc`, objetos de la célula se mantienen en APC<sup>[1]</sup> con solamente un índice mantenido en memoria PHP para identificar la existencia de la célula. De forma predeterminada, se utiliza un tiempo de espera de caché APC de 600 segundos, que debería ser suficiente para la mayoría de las aplicaciones: aunque es posible cambiar esto cuando inicializar `cache_to_apc`.

```
$cacheMethod = PHPEXCEL_CachedObjectStorageFactory::cache_to_apc;  
$cacheSettings = array ('cacheTime' => 600  
    );  
PHPExcel_Settings::setCacheStorageMethod ($cacheMethod, $cacheSettings);
```

Cuando termina la secuencia de comandos todas las entradas se borrará de APC, independientemente del valor `cacheTime`, así que no puede utilizarse para almacenamiento persistente mediante este mecanismo.

#### `PHPExcel_CachedObjectStorageFactory::cache_to_memcache`

Cuando se utiliza `cache_to_memcache`, objetos de la célula se mantienen en memcache<sup>[2]</sup> con solamente un índice mantenido en memoria PHP para identificar la existencia de la célula.

De forma predeterminada, PHPEXCEL busca un servidor memcache en localhost puerto 11211. También establece un límite de tiempo de espera de memcache de 600 segundos. Si está ejecutando en un servidor diferente o puerto memcache, entonces puede cambiar estos valores por defecto cuando se inicialice `cache_to_memcache`:

```
$cacheMethod = PHPEXCEL_CachedObjectStorageFactory::cache_to_memcache;  
$cacheSettings = array ('memcacheServer' => 'localhost',  
    'memcachePort' => 11211,  
    'cacheTime' => 600  
    );  
PHPExcel_Settings::setCacheStorageMethod ($cacheMethod, $cacheSettings);
```

Cuando termina la secuencia de comandos se borrará todas las entradas de memcache, independientemente del valor `cacheTime`, así que no puede utilizarse para almacenamiento persistente mediante este mecanismo.

#### `PHPExcel_CachedObjectStorageFactory::cache_to_wincache;`

Cuando se utiliza `cache_to_wincache`, objetos de la célula se mantienen en Wincache<sup>[3]</sup> con solamente un índice mantenido en memoria PHP para identificar la existencia de la célula. De forma predeterminada, se utiliza un tiempo de espera de caché de Wincache de 600 segundos, que debería ser suficiente para la mayoría de las aplicaciones: aunque es posible cambiar esto cuando inicializar `cache_to_wincache`.

```
$cacheMethod = PHPEXCEL_CachedObjectStorageFactory::cache_to_wincache;  
$cacheSettings = array ('cacheTime' => 600  
    );  
PHPExcel_Settings::setCacheStorageMethod ($cacheMethod, $cacheSettings);
```

Cuando termina la secuencia de comandos todas las entradas se borrará de Wincache, independientemente del valor `cacheTime`, así que no puede utilizarse para almacenamiento persistente mediante este mecanismo.

#### `PHPExcel_CachedObjectStorageFactory::cache_to_sqlite;`

Utiliza una base de datos de SQLite 2 en memoria para almacenar en caché datos de la celda. A diferencia de otros métodos de almacenamiento en caché, las células ni un índice se imparten en memoria PHP - una tabla de base de datos indexadas hace innecesario sostener cualquier índice en memoria PHP - haciendo de este el más memoria eficiente de la célula de almacenamiento en caché de métodos.

#### `PHPExcel_CachedObjectStorageFactory::cache_to_sqlite3;`

Utiliza una base de datos de SQLite 3 en memoria para almacenar en caché datos de la celda. A diferencia de otros métodos de almacenamiento en caché, las células ni un índice se imparten en memoria PHP - una tabla de base de datos indexadas hace innecesario sostener

cualquier índice en memoria PHP - haciendo de este el más memoria eficiente de la célula de almacenamiento en caché de métodos.

### 4.2.2. Idioma/Locale

Se han incluido algunos elementos de localización en PHPEXcel. Se puede establecer un escenario cambiando la configuración. Para establecer la configuración regional al portugués brasileño se usaría:

```
$locale = 'pt_br';
$validLocale = PHPEXcel_Settings::setLocale($locale);
Si (! $validLocale) {}
echo 'Unable to set locale '. $locale. "- volver a es_ES < br / > \n";
}
```

Si no disponibles archivos de idioma portugués brasileño, entonces los portugueses se habilitará en su lugar: si archivos de idioma portugués no disponibles, entonces el método setLocale() devolverá un error, y ajustes de inglés americano (en\_us) se utilizarán a lo largo. Más detalles de las características disponibles una vez que se ha establecido una configuración regional, incluyendo una lista de los idiomas y locales soportados actualmente, pueden encontrarse en la sección 4.5.5 Opciones de localización para las fórmulas.

### 4.3. Despejando un libro de la memoria

El objeto PHPEXcel contiene referencias cíclicas (por ejemplo el libro está vinculado a las hojas de trabajo, y las hojas de trabajo están vinculados a su libro de los padres) que causan problemas cuando PHP intenta borrar los objetos de memoria cuando son unset(), al final de una función o cuando están en el ámbito local. El resultado de esto es "memory leaks", que pueden usar fácilmente una gran cantidad de memoria limitada de PHP.

Esto sólo se puede resolver manualmente: Si tienes que desarmar un libro, entonces también debe "romper" estas referencias cíclicas antes de hacerlo. PHPEXcel proporciona el método disconnectWorksheets() para este propósito.

```
$objPHPEXcel -> disconnectWorksheets();
unset($objPHPEXcel);
```

### 4.4. Hojas de cálculo

Una hoja de cálculo es una colección de celdas, fórmula, imágenes, gráficos... Contiene todos los datos necesarios representar como una hoja de cálculo hoja de cálculo.

Cuando se carga un libro desde un archivo de hoja de cálculo, se cargarán con todas sus hojas existentes (a menos que especificó que sólo ciertas hojas deben ser cargadas). Cuando la carga de archivos no- hoja de cálculo (como un archivo CSV o HTML) o de formatos de hoja de cálculo que no se identifican por su nombre (por ejemplo, SYLK) hojas de trabajo, entonces una sola hoja de cálculo llamado "Hoja de cálculo" se crearán que contiene los datos del fichero.

Cuando crea una instancia de un nuevo libro, PHPEXcel lo creará con una sola hoja de cálculo llamado "Hoja de cálculo".

El método getSheetCount() le dirá el número de hojas de cálculo en el libro; mientras que el método getSheetNames() devuelve una lista de todas las hojas de cálculo en el libro, indexado por el orden en el que sus "fichas" aparecería cuando abrió sus puertas en MS Excel (o cualquier otro programa de hoja de cálculo apropiado).

Las hojas de cálculo individuales pueden accederse por nombre o por su posición de índice en el libro. La posición del índice representa el orden en que cada "ficha" de hoja de cálculo se muestra cuando se abra el libro en MS Excel (u otro programa de hoja de cálculo apropiado). Para acceder a una hoja por su índice, utilice el método getSheet().

```
// Obtener la segunda hoja del libro
```

```
// Nota que las hojas están indexadas de 0
$objPHPExcel -> getSheet(1);
```

Si no se especifica un índice de la hoja, entonces se devolverá la primera hoja.

También existen métodos permitiéndole reordenar las hojas del libro.

Para acceder a una hoja por su nombre, utilice el método `getSheetByName()`, especificando el nombre de la hoja de cálculo que desea tener acceso.

```
// Recuperar la hoja de cálculo llamada 'Hoja 1'
$objPHPExcel -> getSheetByName (' hoja 1');
```

Alternativamente, una hoja de cálculo es siempre la hoja de cálculo activa actualmente, y puede acceder directamente. La hoja de cálculo activa es la que se activará cuando se abra el libro en MS Excel (u otro programa de hoja de cálculo apropiado).

```
// Recuperar la hoja de cálculo activa actual
$objPHPExcel -> getActiveSheet();
```

Puede cambiar la hoja activa actualmente por índice o por nombre métodos `setActiveSheetIndex()` y `setActiveSheetIndexByName ()`.

#### 4.4.1. Agregar una nueva hoja de cálculo

Puede agregar una nueva hoja de cálculo al libro utilizando el método `createSheet()` del objeto `PHPExcel`. De forma predeterminada, esto se creará como una nueva hoja "pasada"; Pero también se puede especificar una posición de índice como argumento y la hoja de cálculo se insertará en esa posición, barajando todas las hojas de cálculo subsiguientes de la colección por un lugar.

```
$objPHPExcel -> createSheet();
```

Una nueva hoja de cálculo creada usando este método será llamado "Hoja de cálculo" u "Hoja de cálculo < n >" donde "< n >" es el número más bajo posible garantizar que el título es único.

Alternativamente, puede crear una instancia de una nueva hoja de cálculo (puesta en el título que elijas lo que elijas) y luego insértela en su libro utilizando el método `addSheet()`.

```
// Crear una nueva hoja de cálculo llamado "Mis datos"
$myWorkSheet = new PHPEXcel_Worksheet ($objPHPExcel, 'Mis datos');
// Coloque la hoja de cálculo "Mis datos" como la primera hoja del objeto
PHPExcel
$objPHPExcel -> addSheet ($myWorkSheet, 0);
```

Si no se especifica una posición de índice como segundo argumento, se añadirá la hoja de cálculo nueva después de la última hoja de cálculo existente.

#### 4.4.2. Copia de las hojas de cálculo

Las hojas dentro del mismo libro pueden ser copiadas por crear un clon de la hoja de cálculo que desea copiar y luego utilizando el método `addSheet()` para insertar el clon en el libro.

```
$objClonedWorksheet = clon $objPHPExcel -> getSheetByName (' hoja 1');
$objClonedWorksheet -> setTitle ("copia de hoja de cálculo 1")
$objPHPExcel -> addSheet($objClonedWorksheet);
```

También puede copiar hojas de un libro a otro, aunque esto es más complejo como `PHPExcel` también tiene que replicar el estilo entre los dos libros. El método `addExternalSheet()` se proporciona para este propósito.

```
$objClonedWorksheet = clon $objPHPExcel1 -> getSheetByName (' hoja 1');
$objPHPExcel -> addExternalSheet($objClonedWorksheet);
```

En ambos casos, es responsabilidad de los desarrolladores para asegurar que no se duplican nombres de hoja de cálculo. `PHPExcel` producirá una excepción si se intenta copiar hojas de trabajo que dará como resultado un nombre duplicado.

### 4.4.3. Eliminación de una hoja de cálculo

Usted puede borrar una hoja de cálculo de un libro, identificado por su posición de índice, utilizando el método de `removeSheetByIndex()`

```
$sheetIndex = $objPHPExcel -> getIndex ($objPHPExcel -> getSheetByName (' hoja 1'));  
$objPHPExcel -> removeSheetByIndex($sheetIndex);
```

Si se elimina la hoja de cálculo activa, entonces la hoja en la posición de índice anterior se convertirá en la hoja activa actualmente.

## 4.5. Acceso a las células

Acceso a las células en una hoja de cálculo PHPExcel debería ser muy sencillo. Este tema enumera algunas de las opciones para acceder a una celda.

### 4.5.1. Establecer un valor de celda por coordenadas

Establecer un valor de celda de coordenadas puede hacerse utilizando la hoja de cálculo `setCellValue()` método.

```
$objPHPExcel -> getActiveSheet() -> setCellValue ('B8', 'Un valor');
```

### 4.5.2. Recuperación de una célula por coordenadas

Para recuperar el valor de una celda, la celda primero debe recuperarse de la hoja de cálculo usando el `getCell()` método. Valor de la celda se puede leer otra vez usando la siguiente línea de código:

```
$objPHPExcel -> getActiveSheet() -> getCell('B8') -> getValue();
```

Si usted necesita el valor calculado de una celda, utilice el código siguiente. Esto se explica más en 4.4.35.

```
$objPHPExcel -> getActiveSheet() -> getCell('B8') -> getCalculatedValue();
```

### 4.5.3. Establecer un valor de celda por columna y fila

Establecer un valor de celda de coordenadas puede hacerse utilizando la hoja de cálculo `setCellValueByColumnAndRow()` método.

```
// Establecer la celda B8  
$objPHPExcel -> getActiveSheet() -> setCellValueByColumnAndRow (1, 8, 'Algún valor');
```

### 4.5.4. Recuperación de una celda de la columna y fila

Para recuperar el valor de una celda, la celda primero debe recuperarse de la hoja de cálculo usando el `getCellByColumnAndRow()` método. Valor de la celda se puede leer otra vez usando la siguiente línea de código:

```
// Celda B8  
$objPHPExcel -> getActiveSheet() -> getCellByColumnAndRow (1, 8) -> getValue();
```

Si usted necesita el valor calculado de una celda, utilice el código siguiente. Esto se explica más en 4.4.35

```
// Celda B8  
$objPHPExcel -> getActiveSheet() -> getCellByColumnAndRow (1, 8) -> getCalculatedValue();
```

### 4.5.5. Bucle de células

#### Bucle de células usando iteradores

La forma más fácil de bucle células es mediante el uso de iteradores. Usando iteradores, uno puede utilizar `foreach` para hojas de cálculo, filas y celdas de bucle.



A continuación se ofrece un ejemplo donde leemos todos los valores en una hoja de cálculo y los muestra en una tabla.

```
<? php
$objReader = PHPExcel_IOFactory::createReader('Excel2007');
$objReader -> setReadDataOnly(true);


$objPHPExcel = $objReader -> load("test.xlsx");
$objWorksheet = $objPHPExcel -> getActiveSheet();

echo '< table >'. "\n";
foreach (getRowIterator() como $row -> $objWorksheet) {}
echo '< tr >'. "\n";

$cellIterator = $row -> getCellIterator();
$cellIterator -> setIterateOnlyExistingCells(false); // Esto coloca todas las
células,
// incluso si no se establece.
// Por defecto, sólo células
// que se establecen voluntad ser
// iterada.
foreach ($cellIterator como $cell) {}
echo '< td >'. $cell -> getValue(). '< /td >'. "\n";
}


echo '< /tr >'. "\n";
}
echo ' < / table >'. "\n";
?>
```

Tenga en cuenta que nos hemos fijado el iterador la `setIterateOnlyExistingCells()` a `false`. Esto hace el iterador lazo todas las células, aunque no fueron instalados antes de.

 **Devolverá el iterador de la célula null como la celda si no se establece en la hoja de cálculo.** Ajuste el iterador la célula `setIterateOnlyExistingCells()` a `false` se enrole todas las celdas de la hoja de cálculo que pueden estar disponibles en ese momento. Esto creará nuevas células, si es necesario y aumentar el uso de memoria. Utilice únicamente si se pretende desviar todas las células que son posiblemente disponibles.

## Las células utilizando los índices de colocación

Uno puede utilizar la posibilidad de acceso a los valores de celda por índice de columna y fila como (0,1) en lugar de 'A1' para leer y escribir los valores de celda en bucles.

 **Nota:** índice de columna en PHPExcel está basado en 0 mientras que el índice de fila está basada en 1. Eso significa 'A1' - (0,1)

A continuación se ofrece un ejemplo donde leemos todos los valores en una hoja de cálculo y los muestra en una tabla.

```
<? php
$objReader = PHPExcel_IOFactory::createReader('Excel2007');
$objReader -> setReadDataOnly(true);

$objPHPExcel = $objReader -> load("test.xlsx");
$objWorksheet = $objPHPExcel -> getActiveSheet();

$highestRow = $objWorksheet -> getHighestRow(); // por ejemplo 10
$highestColumn = $objWorksheet -> getHighestColumn(); // e.g 'F'

$highestColumnIndex = PHPExcel_Cell::columnIndexFromString($highestColumn); //
/ por ejemplo 5

echo '< table >'. "\n";
```

```

para ($row = 1; $row <= $highestRow; ++ $row) {}
echo '< tr >'. "\n";

para ($col = 0; $col <= $highestColumnIndex; ++ $col) {}
echo '< td >'. $objWorksheet -> getCellByColumnAndRow ($row, $col) ->
getValue(). '< /td >'. "\n";
}

echo '< /tr >'. "\n";
}
echo ' < / table >'. "\n";
?>

```

#### 4.5.6. Uso de carpetas de valor para facilitar la entrada de datos

Internamente, PHPEXcel utiliza un valor

predeterminado `PHPEXcel_Cell_IValueBinder` implementación

(`PHPEXcel_Cell_DefaultValueBinder`) determinar los tipos de datos de introdujo los datos utilizando una célula `setValue()` método.

Opcionalmente, el comportamiento predeterminado de PHPEXcel puede ser modificado, lo que permite la fácil entrada de datos. Por ejemplo, un

`PHPEXcel_Cell_AdvancedValueBinder` clase está presente. Convierte automáticamente los porcentajes y fechas entradas como cadenas al formato correcto, también configurar la información de estilo de la celda. El ejemplo siguiente muestra cómo establecer la carpeta valor en PHPEXcel:

```

/ ** PHPEXcel */
require_once 'PHPEXcel.php';

/ ** PHPEXcel_Cell_AdvancedValueBinder */
require_once 'PHPEXcel/Cell/AdvancedValueBinder.php';

/ ** PHPEXcel_IOFactory */
require_once 'PHPEXcel/IOFactory.php';

/ / Set carpeta del valor
PHPEXcel_Cell::setValueBinder (new PHPEXcel_Cell_AdvancedValueBinder());

/ / Crear nuevo objeto PHPEXcel
$objPHPEXcel = new PHPEXcel();

// ...

/ / Agregar algunos datos, que se asemeja a algunos tipos de datos diferentes
$objPHPEXcel -> getActiveSheet() -> setCellValue ('A4', ' valor porcentual:');
$objPHPEXcel -> getActiveSheet() -> setCellValue ('B4', 10% ');
/ / Convierte a 0.1 y establece el estilo de celda porcentaje
$objPHPEXcel -> getActiveSheet() -> setCellValue ('A5', ' valor de fecha y
hora:');
$objPHPEXcel -> getActiveSheet() -> setCellValue ('B5', 21 Diciembre 1983');
/ / Convierte a la fecha y establece el estilo de celda de formato de fecha

```



#### Crear su propia carpeta de valor es fácil.

Cuando avanzaron vinculante de valor es necesario, puede implementar la

`PHPEXcel_Cell_IValueBinder` interfaz o extender

el `PHPEXcel_Cell_DefaultValueBinder` o `PHPEXcel_Cell_AdvancedValueBinder` clases.

### 4.6. PHPEXcel recetas

Las siguientes páginas ofrecen algunas recetas PHPEXcel ampliamente utilizado. Por favor tenga en cuenta que estos no ofrecen documentación completa sobre las funciones específicas de la API PHPEXcel, pero sólo un golpe para que usted consiga comenzado. Si necesitas funciones API específicas, consulte la documentación de la API.

Por ejemplo, 4.4.7 De ajuste de tamaño y orientación de página de una hoja de cálculo cubre establecer una orientación de página A4. Otros formatos de papel, como nosotros, no están cubiertos en este documento, pero en la documentación de la API PHPEXcel.

### 4.6.1. Configuración de metadatos de una hoja de cálculo

PHPEXcel permite una manera fácil de metadatos de una hoja de cálculo, utilizando descriptores de propiedades de documento. Metadatos de hoja de cálculo pueden ser útil para encontrar un documento específico en un repositorio de archivos o un sistema de gestión documental. Por ejemplo Microsoft Sharepoint utiliza metadatos para buscar un documento específico en sus listas de documentos.

Configuración de metadatos de hoja de cálculo se realiza de la siguiente manera:

```
$objPHPEXcel -> getProperties() -> setCreator ("Martín Balliauw");
$objPHPEXcel -> getProperties() -> setLastModifiedBy ("Martín Balliauw");
$objPHPEXcel -> getProperties() -> setTitle ("Office 2007 XLSX prueba documento");
$objPHPEXcel -> getProperties() -> setSubject ("Office 2007 XLSX prueba documento");
$objPHPEXcel -> getProperties() -> descripciónDelConjunto ("prueba documento para Office 2007 XLSX, generado mediante clases PHP.");
$objPHPEXcel -> getProperties() -> setKeywords ("office 2007 openxml php");
$objPHPEXcel -> getProperties() -> setCategory ("resultado archivo de prueba");
```

### 4.6.2. Ajuste de la hoja activa de una hoja de cálculo

La siguiente línea de código establece el índice de la hoja activa a la primera hoja:

```
$objPHPEXcel -> setActiveSheetIndex(0);
```

### 4.6.3. Escriba una fecha u hora en una celda

En Excel, las fechas y horas se almacenan como valores numéricos, contando el número de días transcurridos desde 1900-01-01. Por ejemplo, la fecha '2008-12-31' se representa como 39813. Usted puede verificar esto en Microsoft Office Excel entrando en esa fecha en una celda y luego cambiar el formato de número a 'General' así se revela el verdadero valor numérico. Asimismo, '3:15' se representa como 0.135417.

PHPEXcel trabaja con UST fecha (hora Universal) y valores de tiempo, pero no sin conversiones internas; Así es el desarrollador para asegurar que los valores pasan a las funciones de conversión de fecha/hora son UST.

Escribir un valor de fecha en una celda consta de 2 líneas de código. Seleccione el método que más le convenga mejor. Estos son algunos ejemplos:

```
/ * PHPEXcel_Cell_AdvancedValueBinder requerido para esta muestra * /
require_once 'PHPEXcel/Cell/AdvancedValueBinder.php';

/ / MySQL-como timestamp '2008-12-31' o fecha de cadena
PHPEXcel_Cell::setValueBinder (new PHPEXcel_Cell_AdvancedValueBinder());
$objPHPEXcel -> getActiveSheet()
-> setCellValue ('D1', '2008-12-31');
$objPHPEXcel -> getActiveSheet()
-> getStyle('D1')
-> getNumberFormat()
-> setFormatCode(PHPEXcel_Style_NumberFormat::FORMAT_DATE_YYYYMMDDSLASH)

/ / PHP-tiempo (Unix)
$time = gmmktime(0,0,0,12,31,2008); / / int(1230681600)
$objPHPEXcel -> getActiveSheet()
-> setCellValue ('D1', PHPEXcel_Shared_Date::PHPToExcel($time));
$objPHPEXcel -> getActiveSheet()
-> getStyle('D1')
-> getNumberFormat()
-> setFormatCode(PHPEXcel_Style_NumberFormat::FORMAT_DATE_YYYYMMDDSLASH)
```

```
// Excel-fecha/hora
$objPHPExcel -> getActiveSheet()
-> setCellValue ('D1', 39813)
$objPHPExcel -> getActiveSheet()
-> getStyle('D1')
-> getNumberFormat()
-> setFormatCode(PHPExcel_Style_NumberFormat::FORMAT_DATE_YYYYMMDDSLASH)
```

Los métodos anteriores para entrar en una fecha que todos producen el mismo resultado. `PHPExcel_Style_NumberFormat` ofrece un montón de formatos de fecha previamente definidos.

El método `PHPExcel_Shared_Date::PHPToExcel()` también trabajará con un objeto `DateTime` PHP.

Del mismo modo, tiempos (o valores de fecha y hora) pueden introducirse en la misma manera: Sólo recuerde que debe utilizar un código de formato adecuado.

#### Notas:

1. Vea la sección "uso valor ligantes para facilitar la entrada de datos" para aprender más sobre el `AdvancedValueBinder` utilizado en el primer ejemplo.
2. En versiones anteriores de `PHPExcel` hasta e incluyendo 1.6.6, cuando una célula tenía un código de formato número fecha-como fue posible entrar en una cita directamente utilizando un entero PHP-tiempo sin convertir a formato de fecha de Excel. A partir de `PHPExcel` 1.6.7 que esto ya no es compatible.
3. Excel también puede funcionar en un calendario basado en 1904 (por defecto para los libros guardados en Mac). Normalmente, no tienes que preocuparse por esto al utilizar `PHPExcel`.

### 4.6.4. Escribir una fórmula en una celda

Dentro del archivo de Excel, fórmulas siempre se almacenan como aparecería en una versión en Inglés de Microsoft Office Excel y `PHPExcel` maneja todas las fórmulas internamente en este formato. Esto significa que tienen las siguientes reglas:

- El separador decimal es '.' (período)
- Separador de argumento de la función es ',' (coma)
- Separador de fila de la matriz es ';' (punto y coma)
- Deben utilizarse los nombres de función inglés

Esto es independientemente de qué versión de idioma de Microsoft Office Excel puede se han utilizado para crear el archivo de Excel.

Cuando se abra el libro final por parte del usuario, Microsoft Office Excel se encargará de Mostrar la fórmula según el lenguaje de las aplicaciones. Traducción es atendida por la aplicación.

La siguiente línea de código escribe la fórmula `"= si (C4 > 500,"ganancia","")"` en la celda B8. Tenga en cuenta que la fórmula debe empezar con `"="` para hacer `PHPExcel` reconocer esto como una fórmula.

```
$objPHPExcel -> getActiveSheet() -> setCellValue ('B8','= si (C4 > 500,
"ganancia","")');
```

Si quieres escribir un comienzo de cadena con un `"="` a una celda, entonces debe utilizar el método `setCellValueExplicit()`.

```
$objPHPExcel -> getActiveSheet()
```

```
-> setCellValueExplicit ('B8',
"= Si (C4 > 500,"ganancia"," ")',
PHPExcel_Cell_DataType::TYPE_STRING
);
```

Fórmula de una celda puede leerse otra vez usando la siguiente línea de código:

```
$formula = $objPHPExcel -> getActiveSheet() -> getCell('B8') -> getValue();
```

Si usted necesita el valor calculado de una celda, utilice el código siguiente. Esto se explica más en 4.4.35.

```
$value = $objPHPExcel -> getActiveSheet() -> getCell('B8') ->
getCalculatedValue();
```

#### 4.6.5. Opciones de localización para las fórmulas

Se han incluido algunos elementos de localización en PHPEXcel. Se puede establecer un escenario cambiando la configuración. Para establecer la configuración regional en Rusia utilizaría:

```
$locale = 'ru';
$validLocale = PHPEXcel_Settings::setLocale($locale);
Si (! $validLocale) {}
echo 'Unable to set locale '. $locale. "- volver a es_ES < br / > \n";
}
```

Si los archivos de idioma ruso no están disponibles, el método setLocale() devolverá un error y configuración inglés se utilizarán a lo largo.

Una vez que usted ha establecido una configuración regional, usted puede traducir una fórmula de su codificación interno español.

```
$formula = $objPHPExcel -> getActiveSheet() -> getCell('B8') -> getValue();
$translatedFormula =
    PHPEXcel_Calculation::getInstance() ->
_translateFormulaToLocale($formula);
```

También puede crear una fórmula utilizando los nombres de función y separadores de argumento correspondiente a la configuración regional definida; luego traducirlo al inglés antes de establecer el valor de la celda:

```
$formula = '= ДНЕЙ360(ДАТА(2010;2;5);ДАТА(2010;12;31);ИСТИНА)';
$internalFormula =
    PHPEXcel_Calculation::getInstance() ->
translateFormulaToEnglish($formula);
$objPHPExcel -> getActiveSheet() -> setCellValue('B8',$internalFormula);
```

En la actualidad traducción fórmula sólo traduce los separadores del argumento de la función, nombres de las funciones y las constantes TRUE y FALSE.

En la actualidad, se soportan los siguientes valores de configuración regional:

Idioma		Código de configuración regional
Checo	Čeština	CS
Danés	Dansk	da
Alemán	Deutsch	de
Español	English	es
Finlandés	Suomi	Fi
Francés	Français	fr
Húngaro	Magyar	Hu
Italiano	Italiano	lo
Holandés	Nederlands	NL
Noruego	Norsk	No
Polaco	Język polski	PL
Portugués	Português	PT

Idioma		Código de configuración regional
Portugués Brasileño	Português Brasileiro	pt_BR
Ruso	РУССКИЙ ЯЗЫК	RU
Sueco	Svenska	SV
Turco	Türkçe	TR

#### 4.6.6. Escribir un carácter de nueva línea "\n" en una celda (ALT + "Enter")

En Microsoft Office Excel obtienes un salto de línea en una celda por golpear ALT + "Enter". Cuando haces eso, se enciende automáticamente "ajustar texto" de la celda.

Aquí es cómo lograr esto en PHPEXcel:

```
$objPHPEXcel -> getActiveSheet() -> getCell('A1') ->
setValue("hello\nworld");
$objPHPEXcel -> getActiveSheet() -> getStyle('A1') -> getAlignment() ->
setWrapText(true);
```



**Punta**  
Leer más sobre el formato de las células mediante `getStyle()` en otro sitio.



**Punta**  
`AdvancedValueBinder.php` enciende automáticamente "ajustar texto" de la celda cuando ve un carácter en una cadena que se está insertando en una celda. Al igual que Microsoft Office Excel. Pruebe esto:

```
require_once 'PHPEXcel/Cell/AdvancedValueBinder.php';
PHPEXcel_Cell::setValueBinder (new PHPEXcel_Cell_AdvancedValueBinder());

$objPHPEXcel -> getActiveSheet() -> getCell('A1') ->
setValue("hello\nworld");
```

Lea más acerca de `AdvancedValueBinder.php` en otro sitio.

#### 4.6.7. Establecer explícitamente el tipo de datos de una celda

Se puede establecer el tipo de datos de una celda explícitamente mediante el uso de la célula `setValueExplicit` método, o el `setCellValueExplicit` método de una hoja de cálculo. Aquí está un ejemplo:

```
$objPHPEXcel -> getActiveSheet() -> getCell('A1') -> setValueExplicit ('25',
PHPEXcel_Cell_DataType::TYPE_NUMERIC);
```

#### 4.6.8. Cambio una celda en una URL cliqueable

Usted puede hacer una célula un enlace clicable estableciendo su propiedad de hipervínculo:

```
$objPHPEXcel -> getActiveSheet() -> setCellValue ('E26', 'www.phpexcel.net');
$objPHPEXcel -> getActiveSheet() -> getCell('E26') -> getHyperlink() ->
setUrl ('http://www.phpexcel.net');
```

Si quieres hacer un hipervínculo a otra hoja de cálculo/celda, utilice el siguiente código:

```
$objPHPEXcel -> getActiveSheet() -> setCellValue ('E26', 'www.phpexcel.net');
$objPHPEXcel -> getActiveSheet() -> getCell('E26') -> getHyperlink() ->
setUrl ("hoja: / / 'Sheetname'!A1 ");
```

#### 4.6.9. Ajuste el tamaño y la orientación de página de una hoja de cálculo

Ajuste la orientación de página y el tamaño de una hoja de cálculo puede hacerse utilizando las siguientes líneas de código:

```
$objPHPEXcel -> getActiveSheet() -> getPageSetup() ->
setOrientation (PHPEXcel_Worksheet_PageSetup::ORIENTATION_LANDSCAPE);
```

```
$objPHPExcel -> getActiveSheet() -> getPageSetup() ->
setPaperSize(PHPExcel_Worksheet_PageSetup::PAPERSIZE_A4);
```

Tenga en cuenta que hay configuración de página adicionales disponibles. Por favor, consulte la documentación de la API para todas las opciones posibles.

#### 4.6.10. Page Setup: opciones de escala

Las opciones de configuración de página escalado en PHPExcel se relacionan directamente con las opciones de escala en el cuadro de diálogo "Configurar página" como se muestra en la ilustración.

Los valores predeterminados en PHPExcel corresponden a valores por defecto en MS Office Excel como se muestra en la ilustración


método	valor inicial	método se activará	Nota
setFitToPage(...)	falso	-	
setScale(...)	100	setFitToPage(false)	
setFitToWidth(...)	1	setFitToPage(true)	valor 0 significa hacer-no-fit- a-anchura
setFitToHeight(...)	1	setFitToPage(true)	valor 0 significa hacer-no- ajuste de altura

#### Ejemplo

Aquí está cómo ajustar a 1 página de ancho por infinitas páginas de altos:

```
$objPHPExcel -> getActiveSheet() -> getPageSetup() -> setFitToWidth(1);
$objPHPExcel -> getActiveSheet() -> getPageSetup() -> setFitToHeight(0);
```

Como puedes ver, no es necesario llamar `setFitToPage(true)` desde `setFitToWidth(...)` y `setFitToHeight(...)` esto desencadena.

 Si utilizas `setFitToWidth()` en general también debe especificar `setFitToHeight()` explícitamente como en el ejemplo. Tenga cuidado de depender de los valores iniciales. Esto es especialmente cierto si está actualizando desde PHPEXcel 1.7.0 a 1.7.1 donde los valores por defecto para ajuste de altura y ancho ajuste cambiaron de 0 a 1.

#### 4.6.11. Márgenes de página

Para establecer los márgenes de página para una hoja de cálculo, utilice este código:

```
$objPHPEXcel -> getActiveSheet() -> getPageMargins() -> setTop(1);
$objPHPEXcel -> getActiveSheet() -> getPageMargins() -> setRight(0.75);
$objPHPEXcel -> getActiveSheet() -> getPageMargins() -> setLeft(0.75);
$objPHPEXcel -> getActiveSheet() -> getPageMargins() -> setBottom(1);
```

Tenga en cuenta que los valores de margen se especifiquen en pulgadas.

#### 4.6.12. Centro de una página horizontal/vertical

Para centrar una página horizontal/vertical, puede utilizar el código siguiente:

```
$objPHPEXcel -> getActiveSheet() -> getPageSetup() ->
setHorizontalCentered(true);
$objPHPEXcel -> getActiveSheet() -> getPageSetup() ->
setVerticalCentered(false);
```

#### 4.6.13. Configuración de la impresión encabezado y el pie de una hoja de cálculo

Configuración de una hoja de cálculo impresa encabezado y el pie puede hacerse utilizando las siguientes líneas de código:

```
$objPHPEXcel -> getActiveSheet() -> getHeaderFooter() -> setOddHeader ('& C &
HPlease considerar este documento confidencial!');
$objPHPEXcel -> getActiveSheet() -> getHeaderFooter() -> setOddFooter ('& L &
B'. $objPHPEXcel -> getProperties() -> getTitle(). DPágina & P de N');
```



Sustitución y códigos de formato (comenzando con &) puede ser utilizado dentro de los encabezados y pies de página. No hay ninguna orden requiere que estos códigos deben aparecer.

La primera aparición de los siguientes códigos enciende el formato, el segundo acontecimiento lo apaga otra vez:

- » Tachado
- » Superíndice
- » Subíndice

Superíndice y subíndice no pueden tanto ser ON al mismo tiempo. Lo que ocurra primero gana y el otro es ignorado, mientras que el primero está encendido.

Excel2007 admite los siguientes códigos:

& L	Código de la sección de "izquierda" (hay tres encabezado / pie de página ubicaciones, "left", "centro" y "derecha"). Cuando existen dos o más apariciones de este marcador de sección, el contenido de todos los marcadores está concatenado, en el orden de aparición y colocado en la sección izquierda.
& P	Código para "actual página #"
& N	Código para "totales páginas"
& tamaño de fuente	El código para el "tamaño de texto", donde el tamaño de fuente es un tamaño de fuente en puntos.
& K	Código para "color de fuente del texto" <ul style="list-style-type: none"> <li>» Color RGB se especifica como 100s</li> <li>» Color del tema es especificada como TTNN donde TT es el color del tema Id, tampoco es "+" o "-" del valor del tinte/pantalla, NN es el valor de matiz/pantalla.</li> </ul>
& S	Código para "texto tachado" on / off
& X	Código para "texto super script" on / off
& Y	Código para "subíndice texto" on / off
& C	Código de "sección de centro". Cuando existen dos o más apariciones de este marcador de sección, el contenido de todos los marcadores está concatenado, en el orden de aparición y colocado en la sección central.
& D	Código para "fecha"
& T	Código para "tiempo"
& G	Código para el "cuadro de fondo" <p>Por favor asegúrese de agregar la imagen a la cabecera y pie de página:</p> <pre>\$objDrawing = new PHPExcel_Worksheet_HeaderFooterDrawing(); \$objDrawing -&gt; setName ('PHPExcel logo'); \$objDrawing -&gt; setPath ('./images/phpexcel_logo.gif'); \$objDrawing -&gt; setHeight(36); \$objPHPExcel -&gt; getActiveSheet() -&gt; getHeaderFooter() -&gt; addImage (\$objDrawing, PHPExcel_Worksheet_HeaderFooter::IMAGE_HEADER_LEFT);</pre>
& U	Código para "texto subrayado simple"
& E	Código para "doble subrayado"
& R	Código de "sección de la derecha". Cuando existen dos o más apariciones de este marcador de sección, el contenido de todos los marcadores está concatenado, en el orden de aparición y colocado en la sección derecha.
& Z	Código para la "ruta de archivo de este libro"
& F	Código para "nombre de archivo de este libro"
& A	Código para "nombre de la ficha de hoja"

&+	Código para agregar a la página #
&-	Código para restar página #
& "el nombre de fuente, tipo de letra"	El código de "nombre de la fuente de texto" y "texto tipo de fuente", donde el nombre de fuente y fuente tipo son cadenas especificando el nombre y tipo de la fuente, separada por una coma. Cuando aparezca un guión en nombre de la fuente, que significa "no especificado". Nombre de fuente y tipo de letra pueden ser localizados valores.
& "-", audaz"	Código de estilo "negrita"
& B	Código de estilo "negrita"
& "-", regular"	Código para "regular font style"
& "-", cursiva"	Código de "estilo de letra cursiva"
& me	Código de "estilo de letra cursiva"
& "-", negrita cursiva"	Código de "estilo de negrita cursiva"
& O	Código de "estilo de contorno"
& H	Código de estilo "sombra"

#### ! 📌 Punta

La tabla de códigos puede parecer abrumadora primera vez que intentan averiguar cómo escribir un encabezado o pie de página. Afortunadamente, hay una manera más fácil. Que Microsoft Office Excel haga el trabajo por ti.

Por ejemplo, crear en Microsoft Office Excel un `xlsx` archivo donde insertar el encabezado y el pie como se desee utilizando los programas propia interfaz. Guardar el archivo como `test.xlsx`. Ahora, toma ese archivo y leer los valores usando PHPEXcel como sigue:

```
$objPHPExcel = PHPEXcel_IOFactory::load('test.xlsx');
$objWorksheet = $objPHPExcel -> getActiveSheet();
var_dump ($objWorksheet -> getHeaderFooter() -> getOddFooter());
var_dump ($objWorksheet -> getHeaderFooter() -> getEvenFooter());
var_dump ($objWorksheet -> getHeaderFooter() -> getOddHeader());
var_dump ($objWorksheet -> getHeaderFooter() -> getEvenHeader());
```

Que revela los códigos para los pares/impares encabezado y pie de página. Los usuarios experimentados pueden ser más fácil de cambiar el nombre de `test.xlsx` a `test.zip`, lo descomprimos e inspeccionar directamente el contenido de los pertinentes `xl/worksheets/sheetX.xml` para encontrar los códigos para encabezado y pie de página.

### 4.6.14. Configuración de impresión se rompe en una fila o columna

Para establecer un descanso de impresión, utilice el siguiente código, que establece un descanso de fila en fila 10.

```
$objPHPExcel -> getActiveSheet() -> setBreak ('A10',
PHPExcel_Worksheet::BREAK_ROW);
```

La siguiente línea de código establece una impresión rotura en la columna D:

```
$objPHPExcel -> getActiveSheet() -> setBreak ('D10',
PHPExcel_Worksheet::BREAK_COLUMN);
```

### 4.6.15. Mostrar u ocultar la cuadrícula cuando se imprime

Para mostrar u ocultar la cuadrícula cuando se imprime, utilice el siguiente código:

```
$objPHPExcel -> getActiveSheet() -> setShowGridlines(true);
```

### 4.6.16. Ajuste de filas o columnas para repetir en el superior izquierdo

PHPExcel puede repetir filas/células específicas en la parte superior izquierda de una página. El siguiente código es un ejemplo de cómo repetir la fila 1 a 5 en cada página impresa de una hoja de cálculo específico:

```
$objPHPExcel -> getActiveSheet() -> getPageSetup() ->
setRowsToRepeatAtTopByStartAndEnd (1, 5);
```

## 4.6.17. Área de impresión especificar

Para especificar el área de impresión de una hoja de cálculo, utilice el siguiente código:

```
$objPHPExcel -> getActiveSheet() -> getPageSetup() -> setPrintArea('A1:E5');
```

También puede haber múltiples áreas de impresión en una sola hoja de cálculo:

```
$objPHPExcel -> getActiveSheet() -> getPageSetup() ->
setPrintArea('A1:E5,G4:M20');
```

## 4.6.18. Formato de celdas

Una célula puede formatearse con fuente, frontera, relleno,... información de estilo. Por ejemplo, uno puede establecer el color de primer plano de una célula al rojo, alineada a la derecha y la frontera al estilo de borde negro y espeso. Vamos a hacerlo en la celda B2:

```
$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getFont() ->
getColor() -> setARGB(PHPEXcel_Style_Color::COLOR_RED);

$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getAlignment() ->
setHorizontal(PHPEXcel_Style_Alignment::HORIZONTAL_RIGHT);

$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getBorders() ->
getTop() -> setBorderStyle(PHPEXcel_Style_Border::BORDER_THICK);
$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getBorders() ->
getBottom() -> setBorderStyle(PHPEXcel_Style_Border::BORDER_THICK);
$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getBorders() ->
getLeft() -> setBorderStyle(PHPEXcel_Style_Border::BORDER_THICK);
$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getBorders() ->
getRight() -> setBorderStyle(PHPEXcel_Style_Border::BORDER_THICK);

$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getFill() ->
setFillType(PHPEXcel_Style_Fill::FILL_SOLID);
$objPHPExcel -> getActiveSheet() -> getStyle('B2') -> getFill() ->
getStartColor() -> setARGB('FFFF0000');
```

A partir de PHPEXcel 1.7.0 `getStyle()` también acepta un rango de celdas como parámetro. Por ejemplo, puede establecer un color de fondo rojo en un rango de celdas:

```
$objPHPExcel -> getActiveSheet() -> getStyle('B3:B7') -> getFill()
-> setFillType(PHPEXcel_Style_Fill::FILL_SOLID)
-> getStartColor() -> setARGB('FFFF0000');
```

### Punta

Se recomienda a muchas células del estilo a la vez, usando por ejemplo `getStyle('A1:M500')`, en lugar de labrar las células individualmente en un bucle. Esto es mucho más rápido comparado con bucle a través de las células y peinarlos individualmente.

También hay una manera alternativa para establecer estilos. El siguiente código establece el estilo de una célula de fuente negrita, alineación derecha superior borde delgado y un relleno de degradado:

```
$styleArray = array ()
'font' => array ()
'audaz' => true,
),
'alineación' => array ()
'horizontal' => PHPEXcel_Style_Alignment::HORIZONTAL_RIGHT,
),
'fronteras' => array ()
'top' => array ()
'estilo' => PHPEXcel_Style_Border::BORDER_THIN,
),
),
'relleno' => array ()
'tipo' => PHPEXcel_Style_Fill::FILL_GRADIENT_LINEAR,
```

```
'rotación' => 90,
'startcolor' => array ()
'argb' => 'FFA0A0A0',
),
'endcolor' => array ()
'argb' => 'FFFFFFF',
),
),
);

$objPHPExcel -> getActiveSheet() -> getStyle('A3') ->
applyFromArray($styleArray);
```

O con un rango de celdas:

```
$objPHPExcel -> getActiveSheet() -> getStyle('B3:B7') ->
applyFromArray($styleArray);
```

Este método alternativo usando matrices debería ser más rápido en términos de ejecución cuando se establecen más de una propiedad de estilo. Pero la diferencia apenas puede ser mensurable a menos que usted tiene muchos estilos diferentes en su libro.

⚠ Antes PHPExcel 1.7.0 `duplicateStyleArray()` fue el método recomendado para labrar un rango de celdas, pero este método ya ha quedado obsoleto ya que `getStyle()` ha comenzado a aceptar un rango de celdas.

#### 4.6.19. Número de formatos

A menudo desea dar formato de números en Excel. Por ejemplo puede que desee un miles separador más un número fijo de decimales después del separador decimal. O tal vez quieres algunos números para ser con dos dígitos.

En Microsoft Office Excel se puede estar familiarizado con seleccionar un formato de número en el cuadro de diálogo "Formato de celdas". Aquí hay algunos formatos predefinidos disponibles incluyendo algunas fechas. El cuadro de diálogo está diseñado de una manera así que no tienes que interactuar con el código de formato de número crudo subyacente a menos que usted necesita un formato de número personalizado.

En PHPExcel, también puede aplicar diversos formatos predefinidos. Ejemplo:

```
$objPHPExcel -> getActiveSheet() -> getStyle('A1') -> getNumberFormat()
-> setFormatCode(PHPExcel_Style_NumberFormat::FORMAT_NUMBER_COMMA_SEPARATED1);
```

Esto formateará un número por ejemplo 1587.2 y aparece como 1,587.20 cuando se abre el libro en MS Office Excel. (Dependiendo de la configuración de decimales y miles separadores en Microsoft Office Excel puede demostrar para arriba como 1.587,20)

Mediante el cual puede conseguir exactamente igual que el anterior:

```
$objPHPExcel -> getActiveSheet() -> getStyle('A1') -> getNumberFormat()
-> setFormatCode('#,##0.00');
```

En Microsoft Office Excel, así como en PHPExcel, tendrás que interactuar con códigos de formato de número crudo cuando usted necesita algo especial formato de número personalizado. Ejemplo:

```
$objPHPExcel -> getActiveSheet() -> getStyle('A1') -> getNumberFormat()
-> setFormatCode('[azul] [> = 3000] $#, ## 0; [Rojo] [< 0] $#, ## 0; $#, ## 0');
```

Otro ejemplo es cuando quieres números cero-se rellena con ceros a una longitud fija:

```
$objPHPExcel -> getActiveSheet() -> getCell('A1') -> setValue(19);
$objPHPExcel -> getActiveSheet() -> getStyle('A1') -> getNumberFormat()
-> setFormatCode('0000'); // mostrará como 0019 en Excel
```

#### Punta

Las reglas para componer un número código de formato en Excel pueden ser bastante complicadas. A veces usted sabe cómo crear un formato de número en Microsoft Office Excel, pero no sé cómo se ve el código subyacente de formato número. ¿Cómo lo encuentras?

Los lectores envían con PHPEXcel acudir al rescate. Carga tu libro de plantilla utilizando por ejemplo Excel2007 lector para revelar el número de código de formato. Ejemplo de cómo leer un número código de formato de celda A1:

```
$objReader = PHPEXcel_IOFactory::createReader('Excel2007');
$objPHPEXcel = $objReader -> load('template.xlsx');
var_dump($objPHPEXcel -> getActiveSheet() -> getStyle('A1') ->
getNumberFormat()
-> getFormatCode());
```

Los usuarios avanzados pueden encontrar más rápido para inspeccionar el número de código de formato directamente por renombrar template.xlsx a template.zip, descomprimir y buscando la correspondiente pieza de código XML con el número de código de formato en *xl/styles.xml*.

### 4.6.20. Alineación y texto

Vamos a poner la alineación vertical en la parte superior de las células A1:D4

```
$objPHPEXcel -> getActiveSheet() -> getStyle('A1:D4')
-> getAlignment() ->
setVertical(PHPEXcel_Style_Alignment::VERTICAL_TOP);
```

Aquí es cómo lograr el texto:

```
$objPHPEXcel -> getActiveSheet() -> getStyle('A1:D4')
-> getAlignment() -> setWrapText(true);
```

### 4.6.21. Ajuste el estilo predeterminado de un libro

Es posible establecer el estilo predeterminado de un libro. Vamos a poner la fuente predeterminada Arial tamaño 8:

```
$objPHPEXcel -> getDefaultStyle() -> getFont() -> setName('Arial');
$objPHPEXcel -> getDefaultStyle() -> getFont() -> setSize(8);
```

### 4.6.22. Estilo bordes de celda

En PHPEXcel es fácil de aplicar distintas fronteras en una selección rectangular. Aquí es cómo aplicar un contorno borde rojo grueso alrededor de las células B2:G8.

```
$styleArray = array ()
'fronteras' => array ()
'esquema' => array ()
'estilo' => PHPEXcel_Style_Border::BORDER_THICK,
'color' => array ('argb' => 'FFFF0000'),
),
);
$objWorksheet -> getStyle('B2:G8') -> applyFromArray($styleArray);
```

En Microsoft Office Excel, correspondería a la operación anterior para seleccionar las células B2:G8, lanzando el cuadro de diálogo estilo, eligiendo un borde rojo grueso y haciendo clic en el componente de frontera "Esquema".



Tenga en cuenta que el contorno de la frontera se aplica a la selección rectangular B2:G8 en su conjunto, no en cada célula individualmente.

Puede lograr cualquier efecto frontera usando sólo las fronteras básicas 5 y operan en una sola célula a la vez:

Clave de la matriz	Se asigna a la propiedad
izquierda	getLeft()
derecho	getRight()
Arriba	getTop()
parte inferior	getBottom()
diagonal	getDiagonal()

Fronteras de accesos directos adicionales vienen práctico como en el ejemplo anterior. Estas son las fronteras de método abreviado disponibles:

Clave de la matriz	Se asigna a la propiedad
allborders	getAllBorders()
esquema	getOutline()
interior	getInside()
vertical	getVertical()
horizontal	getHorizontal()

Una visión general de todos los métodos abreviados frontera puede verse en la siguiente imagen:



Si al mismo tiempo establece por ejemplo allborders y vertical, entonces tenemos "superposición" de las fronteras, y uno de los componentes tiene que ganar en el otro donde hay superposición de frontera. En PHPExcel, de más débil a las fronteras más fuertes, la lista es la siguiente: allborders, contorno/adentro, vertical/horizontal, izquierda/derecha/arriba/abajo/diagonal.

Esta jerarquía de frontera puede ser utilizada para lograr diferentes efectos de una manera fácil.

#### 4.6.23. Una célula de formato condicional

Una célula puede ser formateada condicional, basada en una regla específica. Por ejemplo, uno puede establecer el color de primer plano de una célula roja si su valor es menor que cero y verde si su valor es cero o más.

Uno puede establecer un conjunto de reglas de estilo condicional a una celda usando el siguiente código:

```
$objConditional1 = new PHPExcel_Style_Conditional();
$objConditional1 ->
setConditionType(PHPExcel_Style_Conditional::CONDITION_CELLIS);
$objConditional1 ->
setOperatorType(PHPExcel_Style_Conditional::OPERATOR_LESSTHAN);
$objConditional1 -> addCondition('0');
```

```

$objConditional1 -> getStyle() -> getFont() -> getColor() ->
setARGB(PHPExcel_Style_Color::COLOR_RED);
$objConditional1 -> getStyle() -> getFont() -> setBold(true);

$objConditional2 = new PHPExcel_Style_Conditional();
$objConditional2 ->
setConditionType(PHPExcel_Style_Conditional::CONDITION_CELLIS);
$objConditional2 ->
setOperatorType(PHPExcel_Style_Conditional::OPERATOR_GREATERTHANOREQUAL);
$objConditional2 -> addCondition('0');
$objConditional2 -> getStyle() -> getFont() -> getColor() ->
setARGB(PHPExcel_Style_Color::COLOR_GREEN);
$objConditional2 -> getStyle() -> getFont() -> setBold(true);

$conditionalStyles = $objPHPExcel -> getActiveSheet() -> getStyle('B2') ->
getConditionalStyles();
array_push ($conditionalStyles, $objConditional1);
array_push ($conditionalStyles, $objConditional2);
$objPHPExcel -> getActiveSheet() -> getStyle('B2') ->
setConditionalStyles($conditionalStyles);

```

Si desea copiar el conjunto de reglas a otras células, puede duplicar el objeto de estilo:

```

$objPHPExcel -> getActiveSheet() -> duplicateStyle ($objPHPExcel ->
getActiveSheet() -> getStyle('B2'), 'B3:B7');

```

#### 4.6.24. Añadir un comentario a una celda

Para agregar un comentario a una celda, utilice el código siguiente. El ejemplo siguiente agrega un comentario a la celda E11:

```

$objPHPExcel -> getActiveSheet() -> getComment('E11') ->
setAuthor('PHPExcel');
$objCommentRichText = $objPHPExcel -> getActiveSheet() -> getComment('E11') -
> getText() -> createTextRun('PHPExcel:');

$objCommentRichText -> getFont() -> setBold(true);

$objPHPExcel -> getActiveSheet() -> getComment('E11') -> getText() ->
createTextRun("\r\n");

$objPHPExcel -> getActiveSheet() -> getComment('E11') -> getText() ->
createTextRun ('Total importe de la factura actual, excluyendo VAT.');
```

#### 4.6.25. Autofiltro de aplicar a un rango de celdas

Para aplicar un Autofiltro a un rango de celdas, utilice el siguiente código:

```

$objPHPExcel -> getActiveSheet() -> setAutoFilter('A1:C9');
```



**Asegúrese de que siempre incluye la gama completa de filtros!**

Excel soporta configuración solamente la fila de subtítulos, pero eso no es una mejor práctica.

## 4.6.26. Configuración de seguridad en una hoja de cálculo

Excel ofrece 3 niveles de «protección»: documento de seguridad, hoja de seguridad y seguridad de celda.

- Seguridad de documentos le permite establecer una contraseña en una hoja de cálculo completa, permitiendo cambios sólo cuando se introduce la contraseña.
- Hoja seguridad ofrece otras opciones de seguridad: usted puede no permitir insertar filas en una hoja específica, no permitir ordenar,...
- Seguridad de celda ofrece la opción de bloquear/desbloquear un celular, así como Mostrar u ocultar la fórmula interna

Un ejemplo de configuración de seguridad de documentos:

```
$objPHPEXcel -> getSecurity() -> setLockWindows(true);  
$objPHPEXcel -> getSecurity() -> setLockStructure(true);  
$objPHPEXcel -> getSecurity() -> setWorkbookPassword("PHPEXcel");
```

Un ejemplo de configuración de seguridad de la hoja de cálculo:

```
$objPHPEXcel -> getActiveSheet() -> getProtection() ->  
setPassword('PHPEXcel');  
$objPHPEXcel -> getActiveSheet() -> getProtection() -> setSheet(true);  
$objPHPEXcel -> getActiveSheet() -> getProtection() -> setSort(true);  
$objPHPEXcel -> getActiveSheet() -> getProtection() -> setInsertRows(true);  
$objPHPEXcel -> getActiveSheet() -> getProtection() -> setFormatCells(true);
```

Un ejemplo de configuración de seguridad de celda:

```
$objPHPEXcel -> getActiveSheet() -> getStyle('B1') -> getProtection() ->  
setLocked()  
    PHPEXcel_Style_Protection::PROTECTION_UNPROTECTED  
);
```



**Asegúrese de habilitar la protección de la hoja de cálculo si necesita alguna de las características de protección de hoja de cálculo!** Esto se puede hacer utilizando el siguiente código:  
\$objPHPEXcel -> getActiveSheet() -> getProtection() -> setSheet(true);

## 4.6.27. Configuración de validación de datos en una celda

Validación de datos es una característica de gran alcance de Excel2007. Permite para especificar un filtro de entrada de datos que pueden ser insertados en una célula específica. Este filtro puede ser un rango (es decir, valor debe estar entre 0 y 10), una lista (es decir, valor debe ser escogido de una lista),...

El siguiente fragmento de código sólo permite números entre 10 y 20 para introducirse en la celda B3:

```
$objValidation = $objPHPEXcel -> getActiveSheet() -> getCell('B3')  
    -> getDataValidation();  
$objValidation -> setType (PHPEXcel_Cell_DataValidation::TYPE_WHOLE);  
$objValidation -> setErrorStyle (PHPEXcel_Cell_DataValidation::STYLE_STOP);  
$objValidation -> setAllowBlank(true);  
$objValidation -> setShowInputMessage(true);  
$objValidation -> setShowErrorMessage(true);  
$objValidation -> setErrorTitle ('entrada error');  
$objValidation -> setError ('número no está permitido!');  
$objValidation -> setPromptTitle ('entrada permitida');  
$objValidation -> setPrompt ('solamente números entre 10 y 20 están  
permitidos.');
```

El siguiente fragmento de código permite sólo un elemento seleccionado de una lista de datos debe introducirse en la celda B3:

```
$objValidation = $objPHPEXcel -> getActiveSheet() -> getCell('B5')  
    -> getDataValidation();
```



```
$objValidation -> setType (PHPExcel_Cell_DataValidation::TYPE_LIST);
$objValidation -> setErrorStyle
(PHPExcel_Cell_DataValidation::STYLE_INFORMATION);
$objValidation -> setAllowBlank(false);
$objValidation -> setShowInputMessage(true);
$objValidation -> setShowErrorMessage(true);
$objValidation -> setShowDropDown(true);
$objValidation -> setErrorTitle ('entrada error');
$objValidation -> setError ('valor no está en lista. ');
$objValidation -> setPromptTitle ('Pick de lista');
$objValidation -> setPrompt ('por favor, tome un valor de la lista
desplegable. ');
$objValidation -> setFormula1 ('"del artículo A, elemento B, artículo C" ');
```

❗ Cuando usando una validación de datos lista como el anterior, asegúrense de poner la lista entre "y" y dividió los elementos con una coma (,).

❗ Es importante recordar que cualquier cadena participando en una fórmula Excel le permite tener máximos 255 caracteres (no bytes). Este establece un límite de cuántos elementos te puede tener en la cadena de "Elemento A, elemento B, artículo C". Por lo tanto, normalmente es una idea mejor que escriba los valores del artículo directamente en un rango de celdas, dicen A1:A3 y utilizar en su lugar, decir, \$objValidation -> setFormula1('Sheet!\$A\$1:\$A\$3');. Otro beneficio es que los valores del elemento se puede contener la coma ',' carácter mismo.

Si usted necesita validación de datos en varias celdas, uno puede clonar el conjunto de reglas:

```
$objPHPExcel -> getActiveSheet() -> getCell('B8') -> setDataValidation (clon
$objValidation);
```

#### 4.6.28. Ajuste ancho de una columna

Ancho de una columna puede establecerse mediante el siguiente código:

```
$objPHPExcel -> getActiveSheet() -> getColumnDimension('D') -> setWidth(12);
```

Si quieres PHPExcel para realizar un cálculo automático ancho, utilice el siguiente código. PHPExcel será aproximado de la columna con el ancho del valor de la columna más ancho.

```
$objPHPExcel -> getActiveSheet() -> getColumnDimension('B') ->
setAutoSize(true);
```

La medida para el ancho de columna en PHPExcel hace **no** corresponden exactamente a la medida que puede utilizarse para en Microsoft Office Excel. Anchos de columna son difíciles de tratar en Excel, y hay varias medidas para el ancho de columna.

1) **Anchura interna en unidades de carácter** (e.g. 8,43 esto es probablemente lo que usted está familiarizado con en Excel)

2) **Ancho en píxeles** (por ejemplo 64 píxeles)

3) **Ancho total en unidades de carácter** (e.g. 9.140625, valor -1 indica ancho desactivada)

PHPExcel siempre opera con 3) "Full width en unidades de carácter" que de hecho es el único valor que está almacenado en cualquier archivo de Excel, por lo tanto el más medida confiable. Por desgracia, **Microsoft Office Excel no presenta con esta medida**. En cambio medidas 1) y 2) son computados por la aplicación cuando el archivo se abre y estos valores se presentan en varios diálogos e información sobre herramientas.

La unidad de la anchura del carácter es el ancho de un '0' (cero) glifo en la fuente predeterminada de los libros. Por lo tanto, anchos de columna medidos en unidades de carácter en dos libros diferentes pueden compararse sólo si tienen la misma fuente de libro predeterminada.

Si usted tiene algunos Excel archivo y necesito saber los anchos de columna en medida 3), puede leer el archivo de Excel con PHPEXcel y eco los valores obtenidos.

#### 4.6.29. Mostrar u ocultar una columna

Para establecer la visibilidad de columna de una hoja de cálculo, puede utilizar el código siguiente. La primera línea muestra explícitamente la columna C, la segunda línea oculta la columna D.

```
$objPHPEXcel -> getActiveSheet() -> getColumnDimension('C') ->
setVisible(true);
$objPHPEXcel -> getActiveSheet() -> getColumnDimension('D') ->
setVisible(false);
```

#### 4.6.30. Grupo/contorno una columna

Para grupo/contorno una columna, puede utilizar el código siguiente:

```
$objPHPEXcel -> getActiveSheet() -> getColumnDimension('E') ->
setOutlineLevel(1);
```

También puede contraer la columna. Tenga en cuenta que también debe definir la columna invisible, de lo contrario el colapso no será visible en Excel 2007.

```
$objPHPEXcel -> getActiveSheet() -> getColumnDimension('E') ->
setCollapsed(true);
$objPHPEXcel -> getActiveSheet() -> getColumnDimension('E') ->
setVisible(false);
```

Por favor consulte la parte "grupo/contorno una fila" para obtener un ejemplo completo en colapso.

Puede indicar PHPEXcel añadir un resumen a la derecha (por defecto), o a la izquierda. El código siguiente agrega el resumen a la izquierda:

```
$objPHPEXcel -> getActiveSheet() -> setShowSummaryRight(false);
```

#### 4.6.31. Ajuste de altura de una fila

Altura de una fila se puede definir mediante el siguiente código:

```
$objPHPEXcel -> getActiveSheet() -> getRowDimension('10') ->
setRowHeight(100);
```

#### 4.6.32. Mostrar u ocultar una fila

Para establecer la visibilidad de fila de una hoja de cálculo, puede utilizar el código siguiente. En el siguiente ejemplo se esconde el número de la fila 10.

```
$objPHPEXcel -> getActiveSheet() -> getRowDimension('10') ->
setVisible(false);
```

Tenga en cuenta que si usted aplica filtros activos usando un Autofiltro, entonces esto anulará todas las filas que ocultar o Mostrar manualmente dentro de ese rango de Autofiltro si guarda el archivo.

#### 4.6.33. Grupo/contorno una fila

Para grupo/contorno una fila, puede utilizar el código siguiente:

```
$objPHPEXcel -> getActiveSheet() -> getRowDimension('5') ->
setOutlineLevel(1);
```

También puede contraer la fila. Tenga en cuenta que también debe definir la fila invisible, de lo contrario el colapso no será visible en Excel 2007.

```
$objPHPEXcel -> getActiveSheet() -> getRowDimension('5') ->
setCollapsed(true);
```

```
$objPHPExcel -> getActiveSheet() -> getRowDimension('5') ->
setVisible(false);
```

Aquí hay un ejemplo que se derrumba filas 50 a 80:

```
para ($i = 51; $i <= 80; $i++) {}
$objPHPExcel -> getActiveSheet() -> setCellValue ('A'. $i, "FName $i");
$objPHPExcel -> getActiveSheet() -> setCellValue ('B'. $i, "LName $i");
$objPHPExcel -> getActiveSheet() -> setCellValue ('C'. $i, "PhoneNo $i");
$objPHPExcel -> getActiveSheet() -> setCellValue (sería '. $i, "FaxNo $i");
$objPHPExcel -> getActiveSheet() -> setCellValue ('E'. $i, true);

$objPHPExcel -> getActiveSheet() -> getRowDimension($i) ->
setOutlineLevel(1);
$objPHPExcel -> getActiveSheet() -> getRowDimension($i) -> setVisible(false);
}
$objPHPExcel -> getActiveSheet() -> getRowDimension(81) ->
setCollapsed(true);
```

Puede indicar PHPEXcel añadir un resumen debajo de las filas plegables (por defecto), o por encima. El código siguiente agrega el Resumen anterior:

```
$objPHPExcel -> getActiveSheet() -> setShowSummaryBelow(false);
```

#### 4.6.34. Fusión/separación de células

Si usted tiene un gran trozo de datos que desea mostrar en una hoja de cálculo, puede combinar dos o más células juntos, para convertirse en una célula. Esto puede hacerse usando el siguiente código:

```
$objPHPExcel -> getActiveSheet() -> mergeCells('A18:E22');
```

Eliminar una combinación puede hacerse usando el `unmergeCells` método:

```
$objPHPExcel -> getActiveSheet() -> unmergeCells('A18:E22');
```

#### 4.6.35. Insertar filas o columnas

Usted puede insertar o quitar filas o columnas en una posición específica. El código siguiente inserta 2 nuevas filas, justo antes de fila 7:

```
$objPHPExcel -> getActiveSheet() -> insertNewRowBefore (7, 2);
```

#### 4.6.36. Agregar un dibujo a una hoja de cálculo

A dibujar siempre se representa como un objeto separado, que puede ser agregado a una hoja de cálculo. Por lo tanto, primero debe crear un nuevo `PHPExcel_Worksheet_Drawing` y asignar un valor significativo a sus propiedades:

```
$objDrawing = new PHPExcel_Worksheet_Drawing();
$objDrawing -> setName('Logo');
$objDrawing -> setDescription('Logo');
$objDrawing -> setPath('./images/officelogo.jpg');
$objDrawing -> setHeight(36);
```

Para agregar el dibujo anterior a la hoja de cálculo, utilice el siguiente fragmento de código. PHPEXcel crea el vínculo entre el dibujo y la hoja de cálculo:

```
$objDrawing -> setWorksheet ($objPHPExcel -> getActiveSheet());
```

Puede establecer numerosas propiedades en un dibujo, aquí hay algunos ejemplos:

```
$objDrawing -> setName('Paid');
$objDrawing -> setDescription('Paid');
$objDrawing -> setPath('./images/paid.png');
$objDrawing -> setCoordinates('B15');
$objDrawing -> setOffsetX(110);
$objDrawing -> setRotation(25);
$objDrawing -> getShadow() -> setVisible(true);
$objDrawing -> getShadow() -> setDirection(45);
```

También puede agregar imágenes creadas utilizando las funciones GD sin necesidad de guardarlos en el disco primero como dibujos en memoria.

```
// // Use GD para crear una imagen en la memoria
$gdImage = @imagecreatetruecolor (120, 20) or die ('no se puede inicializar
nuevas GD secuencia de imagen');
$textColor = imagecolorallocate ($gdImage, 255, 255, 255);
ImageString ($gdImage, 1, 5, 5, 'Created with PHPEXcel', $textColor);

// // Agregar la imagen en la memoria a una hoja de cálculo
$objDrawing = new PHPEXcel_Worksheet_MemoryDrawing();
$objDrawing -> setName (' imagen en la memoria 1');
$objDrawing -> descripciónDelConjunto (' imagen en la memoria 1');
$objDrawing -> setCoordinates('A1');
$objDrawing -> setImageResource($gdImage);
$objDrawing -> setRenderingFunction()
PHPEXcel_Worksheet_MemoryDrawing::RENDERING_JPEG
);
$objDrawing ->
setMimeType(PHPEXcel_Worksheet_MemoryDrawing::MIMETYPE_DEFAULT);
$objDrawing -> setHeight(36);
$objDrawing -> setWorksheet ($objPHPEXcel -> getActiveSheet());
```

#### 4.6.37. Imágenes de la lectura de una hoja de cálculo

Una pregunta comúnmente es cómo recuperar las imágenes de un libro que se ha cargado, y guardarlos como archivos de imágenes individuales en el disco.

El siguiente código extrae imágenes de la hoja de cálculo activa actual y escribe cada uno como un archivo independiente.

```
$i = 0;
foreach ($objPHPEXcel -> getActiveSheet() -> getDrawingCollection() como
$drawing) {}
Si ($drawing instanceof PHPEXcel_Worksheet_MemoryDrawing) {}
ob_start ();
call_user_func()
$drawing -> getRenderingFunction(),
$drawing -> getImageResource()
);
$imageContents = ob_get_contents();
ob_end_clean();
interruptor ($drawing -> {getMimeType()})
caso PHPEXcel_Worksheet_MemoryDrawing::MIMETYPE_PNG:
$extension = 'png'; rotura;
caso PHPEXcel_Worksheet_MemoryDrawing::MIMETYPE_GIF:
$extension = 'gif'; rotura;
caso PHPEXcel_Worksheet_MemoryDrawing::MIMETYPE_JPEG:
$extension = 'jpg'; rotura;
}
} {mas}
$zipReader = fopen ($drawing -> getPath(), 'r');
$imageContents = '';
mientras que (! feof($zipReader)) {}
$imageContents. = fread($zipReader, 1024);
}
fclose($zipReader);
$extension = $drawing -> getExtension();
}
$myFileName = '00_Image_'. ++ $i.'. '.$extension;
file_put_contents($myFileName, $imageContents);
}
```

#### 4.6.38. Agregar texto a una celda

Agregar texto a una celda puede hacerse usando `PHPEXcel_RichText` casos. Aquí hay un ejemplo, que crea la siguiente cadena de texto enriquecido:

Esta factura es ***pagadero dentro de los treinta días después del final del mes*** a menos que se especifique lo contrario en la factura.

```
$objRichText = new PHPExcel_RichText();
$objRichText -> createText ('esta factura es');

$objPayable = $objRichText -> createTextRun ('pagaderos dentro de los treinta
días después del final del mes');
$objPayable -> getFont() -> setBold(true);
$objPayable -> getFont() -> setItalic(true);
$objPayable -> getFont() -> colores (nuevo PHPExcel_Style_Color
(PHPExcel_Style_Color::COLOR_DARKGREEN));

$objRichText -> createText (' , a menos que se especifique lo contrario en la
factura.);

$objPHPExcel -> getActiveSheet() -> getCell('A18') -> setValue($objRichText);
```

### 4.6.39. Definir un rango con nombre

PHPExcel apoya la definición de rangos con nombre. Estos pueden definirse utilizando el siguiente código:

```
// Agregar algunos datos
$objPHPExcel -> setActiveSheetIndex(0);
$objPHPExcel -> getActiveSheet() -> setCellValue ('A1', ' nombre:');
$objPHPExcel -> getActiveSheet() -> setCellValue ('A2', ' Lastname:');
$objPHPExcel -> getActiveSheet() -> setCellValue ('B1', 'Martín');
$objPHPExcel -> getActiveSheet() -> setCellValue ('B2', 'Balliauw');

// Definir rangos con nombre
$objPHPExcel -> addNamedRange (new PHPExcel_NamedRange ('PersonFN',
$objPHPExcel -> getActiveSheet(), 'B1'));
$objPHPExcel -> addNamedRange (new PHPExcel_NamedRange ('PersonLN',
$objPHPExcel -> getActiveSheet(), 'B2'));
```

Opcionalmente, se puede pasar un cuarto parámetro definir el rango con nombre local (es decir, sólo utilizables en la hoja de cálculo actual). Rangos con nombre son globales por defecto.

### 4.6.40. Redirigir la salida a explorador web del cliente

A veces, uno realmente quiere salida a un archivo en el explorador del cliente, especialmente al crear hojas de cálculo on-the-fly. Hay algunos pasos sencillos que pueden seguirse para hacer esto:

1. Crear la hoja de cálculo PHPExcel
2. Las cabeceras HTTP para el tipo de documento que desea la salida de la salida
3. Utilice el PHPExcel\_Writer\_ \* de su elección y guardar en "php://output"

PHPExcel\_Writer\_Excel2007 utiliza almacenamiento temporal al escribir a php://output. De forma predeterminada, los archivos temporales se almacenan en directorio de trabajo de la secuencia de comandos. Cuando no hay acceso, cae vuelve a la ubicación de los archivos temporales del sistema operativo.



**Esto puede no ser seguro para su visualización no autorizado!**

Dependiendo de la configuración de su sistema operativo, almacenamiento temporal puede ser leído por cualquiera que utilice la misma carpeta de almacenamiento temporal. Cuando es necesaria la confidencialidad del documento, se recomienda no usar php://output.

### Cabeceras HTTP

Ejemplo de un script redireccionar un archivo de Excel 2007 para el navegador del cliente:

```
<? php
/ * Aquí habrá algún código donde crear $objPHPExcel * /
```

```
// // redirigir la salida a explorador cliente
header ('Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet');
header ('Content-Disposition: attachment;filename="myfile.xlsx"');
header ('Cache-Control: max-age = 0');

$objWriter = PHPExcel_IOFactory::createWriter ($objPHPExcel, 'Excel2007');
$objWriter -> save ('php://output');
?>
```

**Ejemplo de un script redireccionar un archivo Excel5 en el navegador del cliente:**

```
<? php
/ * Aquí habrá algún código donde crear $objPHPExcel * /

// // redirigir la salida a explorador cliente
header ('Content-Type: application/vnd.ms-excel');
header ('Content-Disposition: attachment;filename="myfile.xls"');
header ('Cache-Control: max-age = 0');

$objWriter = PHPExcel_IOFactory::createWriter ($objPHPExcel, 'Excel5');
$objWriter -> save ('php://output');
?>
```

#### **PRECAUCIÓN:**

- Asegúrese de que no se incluyen las declaraciones de Eco o salida de cualquier otro contenido que el archivo de Excel. No debe haber ningún espacio en blanco antes de la apertura <? php tag y a lo sumo una línea rompen después del cierre? > etiqueta (que también puede ser omitido para evitar problemas).
- Asegúrese de que el guión está guardado sin un BOM (marca de orden de bytes). (Porque esto cuenta como haciéndose eco de salida)
- Mismas cosas se aplican a archivos todo incluidos

No seguir los lineamientos anteriores puede resultar en Excel corrupto arribando al navegador del cliente, o los encabezados de los archivos no se puede establecer por PHP (resultando en mensajes de advertencia).

#### **4.6.41. Ajuste del ancho de columna predeterminado**

Ancho de columna predeterminado se puede definir mediante el siguiente código:

```
$objPHPExcel -> getActiveSheet() -> getDefaultColumnDimension() ->
setWidth(12);
```

#### **4.6.42. Ajuste la altura de fila predeterminado**

Alto de fila predeterminado se puede definir mediante el siguiente código:

```
$objPHPExcel -> getActiveSheet() -> getDefaultRowDimension() ->
setRowHeight(15);
```

#### **4.6.43. Añade una GD dibujo a una hoja de cálculo**

Puede haber una situación donde usted quiere generar una imagen en la memoria mediante GD y agregarlo a una hoja de cálculo PHPExcel sin tener que guardar este archivo en una ubicación temporal.

Aquí hay un ejemplo que genera una imagen en la memoria y se agrega a la hoja de cálculo activa:

```
// // Generar una imagen
$gdImage = @imagecreatetruecolor (120, 20) or die ('no se puede inicializar
nuevas GD secuencia de imagen');
$textColor = imagecolorallocate ($gdImage, 255, 255, 255);
ImageString ($gdImage, 1, 5, 5, 'Created with PHPExcel', $textColor);

// // Agregar un dibujo a la hoja de cálculo
```

```
$objDrawing = new PHPExcel_Worksheet_MemoryDrawing();
$objDrawing -> setName ('imagen muestra');
$objDrawing -> descripciónDelConjunto ('imagen muestra');
$objDrawing -> setImageResource($gdImage);
$objDrawing ->
setRenderingFunction(PHPExcel_Worksheet_MemoryDrawing::RENDERING_JPEG);
$objDrawing ->
setMimeType(PHPExcel_Worksheet_MemoryDrawing::MIMETYPE_DEFAULT);
$objDrawing -> setHeight(36);
$objDrawing -> setWorksheet ($objPHPExcel -> getActiveSheet());
```

#### 4.6.44. Ajuste el nivel de zoom de la hoja de cálculo

Para establecer el nivel de zoom de una hoja de cálculo, puede utilizarse el código siguiente:

```
$objPHPExcel -> getActiveSheet() -> getSheetView() -> setZoomScale(75);
```

Tenga en cuenta que el nivel de zoom debe ser en gama 10 - 400.

#### 4.6.45. Color de la ficha de hoja

A veces quieres establecer un color de la ficha hoja. Por ejemplo puede tener una ficha de hoja roja:

```
$objWorksheet -> getTabColor() -> setRGB('FF0000');
```

#### 4.6.46. Crear hojas de cálculo en un libro

Si necesitas crear hojas de trabajo más en el libro, aquí está cómo:

```
$objWorksheet1 = $objPHPExcel -> createSheet();
$objWorksheet1 -> setTitle ('otra hoja');
```

Piensa que createSheet() el botón de "Insertar hoja" en Excel. Cuando llegues a ese botón una nueva hoja se agrega a la colección existente de hojas de cálculo en el libro.

#### 4.6.47. Hojas de trabajo oculto (Estados de hoja)

Establecer una hoja de cálculo que **oculta** utilizando este código:

```
$objPHPExcel -> getActiveSheet()
-> setSheetState(PHPExcel_Worksheet::SHEETSTATE_HIDDEN);
```

A veces incluso es recomendable la hoja de cálculo para ser "muy oculto". Los Estados de hoja disponibles son:

```
PHPExcel_Worksheet::SHEETSTATE_VISIBLE
PHPExcel_Worksheet::SHEETSTATE_HIDDEN
PHPExcel_Worksheet::SHEETSTATE_VERYHIDDEN
```

En Excel la hoja estado "muy oculto" sólo puede ser establecido mediante programación, por ejemplo con Macro de Visual Basic. No es posible realizar tal una hoja visible mediante la interfaz de usuario.

#### 4.6.48. Hoja de trabajo de derecha a izquierda

Las hojas de cálculo se pueden ajustar individualmente si columna 'A' debe comenzar en el lado izquierdo o derecho. Queda por defecto. Aquí es cómo establecer columnas de derecha a izquierda.

```
// right-to-left hoja de cálculo
$objPHPExcel -> getActiveSheet()
-> setRightToLeft(true);
```

## 5. Realizar cálculos fórmulas

### 5.1. Usando el motor de cálculo PHPEXcel

Como PHPEXcel representa una hoja de cálculo en la memoria, también ofrece capacidades de cálculo fórmula. Una célula puede ser de un tipo de valor (contiene un número o texto), o un fórmula tipo (contiene una fórmula que se puede evaluar). Por ejemplo, la fórmula "=SUM(A1:A10)" evalúa a la suma de los valores de A1, A2,..., A10.

Para calcular una fórmula, puede llamar a la celda que contiene el método de la fórmula `getCalculatedValue()` , por ejemplo:

```
$objPHPEXcel -> getActiveSheet() -> getCell('E11') -> getCalculatedValue();
```

Si usted escribe la siguiente línea de código en la factura demo incluida con PHPEXcel, se evalúa como el valor "64":

Otra característica interesante del analizador fórmula de PHPEXcel, es que puede ajustar automáticamente una fórmula al insertar o quitar filas o columnas. Aquí está un ejemplo:

Verás que la fórmula contenida en celda E11 es "SUM(E4:E9)". Ahora, cuando escribo la siguiente línea de código, se añaden dos nuevas líneas de productos:

```
$objPHPEXcel -> getActiveSheet() -> insertNewRowBefore (7, 2);
```



¿Notaste? La fórmula en la celda E11 (ahora E13, como he insertado 2 nuevas filas), cambió a "SUM(E4:E11)". Además, las células insertadas duplican información de estilo de la celda anterior, como comportamiento de Excel. Nota que ambos puedes insertar filas y columnas.

## **5.2. Conoce limitaciones**

Existen algunas limitaciones conocidas para el motor de cálculo PHPEXCEL. La mayoría de ellos son debido al hecho de que una fórmula de Excel se convierte en código PHP antes de ser ejecutado. Esto significa que cálculo fórmula Excel está sujeta a las características de lenguaje de PHP.

### **5.2.1. Precedencias de los operadores**

En Excel gana '+' sobre '&', como '\*' gana a '+' en álgebra ordinaria. La regla anterior no es lo que uno encuentra utilizando el motor de cálculo enviados con PHPEXCEL.

Referencia de precedencias de los operadores en Excel:

<http://support.microsoft.com/kb/25189>

Referencia de precedencias de los operadores en PHP:

<http://www.php.net/Operators>

### **5.2.2. Fórmulas que involucran números y texto**

Fórmulas que involucran números y el texto pueden producir resultados inesperados o incluso contenido del archivo ilegible. Por ejemplo, la fórmula '= 3 + "Hola"' se espera que produzca un error en Excel (#VALUE!). Debido al hecho de que PHP convierte "Hola" a un valor numérico (cero), el resultado de esta fórmula se evalúa como 3 en lugar de evaluar como un error. Esto también hace que el documento de Excel que se genera como contenido ilegible.

Referencia de este comportamiento en PHP:

<http://be.php.net/manual/en/Language.Types.String.php#Language.Types.String.Conversion>

## 6. Lectura y escritura al archivo

Como ustedes ya saben de parte 3.3 Lectores y escritores, leer y escribir en un almacenamiento persistente no es posible utilizar las clases base PHPExcel. Para este propósito, PHPExcel proporciona a los lectores y escritores, que son implementaciones de PHPExcel\_Reader\_IReader y PHPExcel\_Writer\_IWriter.

### 6.1. PHPExcel\_IOFactory

La API PHPExcel ofrece varios métodos para crear una instancia de PHPExcel\_Reader\_IReader o PHPExcel\_Writer\_IWriter:

- Creación directa
- Vía PHPExcel\_IOFactory

Todos los ejemplos debajo de demuestran el método de creación directa. Tenga en cuenta que también puede utilizar la clase PHPExcel\_IOFactory para hacerlo.

#### 6.1.1. Creación de PHPExcel\_Reader\_IReader con PHPExcel\_IOFactory

Existen 2 métodos para la lectura de un archivo en PHPExcel: usando el fichero automatizado tipo resolver o explícitamente.

Tipo de archivo automático resolución cheques distribuidos los diferentes PHPExcel\_Reader\_IReader con PHPExcel. Si uno de ellos puede cargar el nombre de archivo especificado, se carga el archivo usando el PHPExcel\_Reader\_IReader. Modo explícito requiere que especifique que PHPExcel\_Reader\_IReader debe ser utilizado.

Puede crear una instancia de PHPExcel\_Reader\_IReader con PHPExcel\_IOFactory en tipo de archivo automático resolución modo usando el ejemplo de código siguiente:

```
$objPHPExcel = PHPExcel_IOFactory::load("05featuredemo.xlsx");
```

Un uso típico de esta característica es cuando tienes que leer los archivos subidos por los usuarios, y no sabes si están cargando archivos xls oxlsx.

Si es necesario establecer algunas propiedades en el lector, (e.g. solamente leer datos, vea más sobre esto más adelante), entonces en vez de eso querrá usar esta variante:

```
$objReader = PHPExcel_IOFactory::createReaderForFile("05featuredemo.xlsx");  
$objReader -> setReadDataOnly(true);  
$objReader -> load("05featuredemo.xlsx");
```

Puede crear una instancia de PHPExcel\_Reader\_IReader con PHPExcel\_IOFactory en modo explícito mediante el ejemplo de código siguiente:

```
$objReader = PHPExcel_IOFactory::createReader("Excel2007");  
$objPHPExcel = $objReader -> load("05featuredemo.xlsx");
```



Nota ese tipo automático resolución modo es ligeramente más lento que el modo explicit.

#### 6.1.2. Creación de PHPExcel\_Writer\_IWriter con PHPExcel\_IOFactory

Puede crear una instancia de PHPExcel\_Writer\_IWriter con PHPExcel\_IOFactory:

```
$objWriter = PHPExcel_IOFactory::createWriter($objPHPExcel, "Excel2007");  
$objWriter -> save("05featuredemo.xlsx");
```

## 6.2. Formato de archivo de Excel 2007 (SpreadsheetML)

Formato archivo Excel2007 es el archivo principal de PHPExcel. Permite la salida de la hoja de cálculo en la memoria a un archivo xlsx.

## 6.2.1. PHPExcel\_Reader\_Excel2007

### Leer una hoja de cálculo

Puede leer un archivo .xlsx utilizando el siguiente código:

```
$objReader = new PHPExcel_Reader_Excel2007();  
$objPHPExcel = $objReader -> load("05featuredemo.xlsx");
```

### Leer sólo los datos

Puede establecer la opción setReadDataOnly en el lector, para instruir al lector a ignorar para estilizar, validación de datos,... y datos de sólo lectura de la célula:

```
$objReader = new PHPExcel_Reader_Excel2007();  
$objReader -> setReadDataOnly(true);  
$objPHPExcel = $objReader -> load("05featuredemo.xlsx");
```

### Lea las hojas específicas sólo

Puede establecer la opción setLoadSheetsOnly en el lector, para indicar al lector que sólo cargue las hojas con un nombre:

```
$objReader = new PHPExcel_Reader_Excel2007();  
$objReader -> setLoadSheetsOnly (array ("hoja 1", "Mi hoja especial"));  
$objPHPExcel = $objReader -> load("05featuredemo.xlsx");
```

### Células específicas de lectura solamente

Puede establecer la opción setReadFilter en el lector, para instruir al lector a cargar sólo las células que coincida con una regla dada. Un filtro de lectura puede ser cualquier clase que implementa PHPExcel\_Reader\_IReadFilter. De forma predeterminada, todas las células se leen usando el PHPExcel\_Reader\_DefaultReadFilter.

El siguiente código sólo leerá fila 1 y 20 - 30 filas de cualquier hoja en el archivo de Excel:

```
class MyReadFilter implements PHPExcel_Reader_IReadFilter  
{  
    readCell de la función pública ($columna, $row, $worksheetName = '') {}  
    // Leen título filas y filas de 20-30  
    Si ($row == 1 || ($row >= 20 && $row <= 30)) {  
        devuelve true;  
    }  
  
    devolver false;  
}  
  
$objReader = new PHPExcel_Reader_Excel2007();  
$objReader -> setReadFilter (new MyReadFilter());  
$objPHPExcel = $objReader -> load("06largescale.xlsx");
```

## 6.2.2. PHPExcel\_Writer\_Excel2007

### Escribir una hoja de cálculo

Puede escribir un archivo .xlsx utilizando el siguiente código:

```
$objWriter = new PHPExcel_Writer_Excel2007($objPHPExcel);  
$objWriter -> save("05featuredemo.xlsx");
```

### Fórmula cálculo previo

De forma predeterminada, este escritor calcula previamente todas las fórmulas en la hoja de cálculo. Esto puede ser lento en grandes hojas de cálculo y tal vez no deseados. Sin embargo puede desactivar el cálculo de la fórmula:

```
$objWriter = new PHPExcel_Writer_Excel2007($objPHPExcel);  
$objWriter -> setPreCalculateFormulas(false);  
$objWriter -> save("05featuredemo.xlsx");
```

## Paquete de compatibilidad de Office 2003

Debido a un bug en el paquete de compatibilidad de Office2003, puede haber algunos pequeños problemas al abrir hojas de cálculo de Excel2007 (sobre todo relacionados con cálculo de fórmula). Puede habilitar la compatibilidad Office2003 con el siguiente código:

```
$objWriter = new PHPExcel_Writer_Excel2007($objPHPExcel);  
$objWriter -> setOffice2003Compatibility(true);  
$objWriter -> save("05featuredemo.xlsx");
```



### Office2003 compatibilidad debe utilizarse sólo cuando sea necesario

Opción de compatibilidad con office2003 debe utilizarse sólo cuando sea necesario. Esta opción deshabilita varias opciones de formato de archivo de Office2007, resultando en una hoja de cálculo de Office2007 bajo-aparece cuando se utiliza esta opción.

## 6.3. Formato de archivo de Excel 5 (BIFF)

Formato Excel5 es el viejo formato de archivo Excel aplicado en PHPExcel para proporcionar una manera uniforme para crear archivos .xls tanto .xlsx. Es básicamente una versión modificada de [pera Spreadsheet\\_Excel\\_Writer](#), aunque se ha extendido y tiene menos limitaciones y más características que la antigua biblioteca de pera. Esto puede leer todas las versiones BIFF que utilizan OLE2: BIFF5 (introducido con office 95) a través de BIFF8, pero no puede leer las versiones anteriores.

Formato de archivo Excel5 no se desarrollará ninguna más, sólo proporciona un formato de archivo adicionales para PHPExcel.



### Limitaciones de Excel5 (BIFF)

Por favor note que formato de archivo BIFF tiene algunos límites en cuanto a estilo células y manejo de hojas de cálculo grandes mediante PHP.

### 6.3.1. PHPExcel\_Reader\_Excel5

#### Leer una hoja de cálculo

Puede leer un archivo .xls utilizando el siguiente código:

```
$objReader = new PHPExcel_Reader_Excel5();  
$objPHPExcel = $objReader -> load("05featuredemo.xls");
```

#### Leer sólo los datos

Puede establecer la opción `setReadDataOnly` en el lector, para instruir al lector a ignorar para estilizar, validación de datos,... y datos de sólo lectura de la célula:

```
$objReader = new PHPExcel_Reader_Excel5();  
$objReader -> setReadDataOnly(true);  
$objPHPExcel = $objReader -> load("05featuredemo.xls");
```

#### Lea las hojas específicas sólo

Puede establecer la opción `setLoadSheetsOnly` en el lector, para indicar al lector que sólo cargue las hojas con un nombre:

```
$objReader = new PHPExcel_Reader_Excel5();  
$objReader -> setLoadSheetsOnly(array("hoja 1", "Mi hoja especial"));  
$objPHPExcel = $objReader -> load("05featuredemo.xls");
```

#### Células específicas de lectura solamente

Puede establecer la opción `setReadFilter` en el lector, para instruir al lector a cargar sólo las células que coincida con una regla dada. Un filtro de lectura puede ser cualquier clase que implementa `PHPExcel_Reader_IReadFilter`. De forma predeterminada, todas las células se leen usando el `PHPExcel_Reader_DefaultReadFilter`.

El siguiente código sólo leerá fila 1 y 20 - 30 filas de cualquier hoja en el archivo de Excel:

```
class MyReadFilter implements PHPExcel_Reader_IReadFilter  
{  
    readCell de la función pública ($columna, $row, $worksheetName = '') {}  
}
```

```
// // Leen título filas y filas de 20-30
Si ($row == 1 || ($row >= 20 && $row <= 30)) {
    devuelve true;
}

devolver false;
}
}

$objReader = new PHPExcel_Reader_Excel5();
$objReader -> setReadFilter (new MyReadFilter());
$objPHPExcel = $objReader -> load("06largescale.xls");
```

### 6.3.2. PHPExcel\_Writer\_Excel5

#### Escribir una hoja de cálculo

Puede escribir un archivo .xls utilizando el siguiente código:

```
$objWriter = new PHPExcel_Writer_Excel5($objPHPExcel);
$objWriter -> save("05featuredemo.xls");
```

## 6.4. Formato de archivo XML de Excel 2003

Formato de archivo Excel 2003 XML es un formato de archivo que puede ser utilizado en versiones anteriores de Microsoft Excel.



#### Limitaciones de Excel 2003 XML

Tenga en cuenta que el formato XML de Excel 2003 tiene algunos límites en cuanto a estilo células y manejo de hojas de cálculo grandes mediante PHP.

### 6.4.1. PHPExcel\_Reader\_Excel2003XML

#### Leer una hoja de cálculo

Puede leer un archivo .xml mediante el siguiente código:

```
$objReader = new PHPExcel_Reader_Excel2003XML();
$objPHPExcel = $objReader -> load("05featuredemo.xml");
```

#### Células específicas de lectura solamente

Puede establecer la opción setReadFilter en el lector, para instruir al lector a cargar sólo las células que coincida con una regla dada. Un filtro de lectura puede ser cualquier clase que implementa PHPExcel\_Reader\_IReadFilter. De forma predeterminada, todas las células se leen usando el PHPExcel\_Reader\_DefaultReadFilter.

El siguiente código sólo leerá fila 1 y 20 - 30 filas de cualquier hoja en el archivo de Excel:

```
class MyReadFilter implements PHPExcel_Reader_IReadFilter
{
    readCell de la función pública ($columna, $row, $worksheetName = '') {}
    // // Leen título filas y filas de 20-30
    Si ($row == 1 || ($row >= 20 && $row <= 30)) {
        devuelve true;
    }

    devolver false;
}

$objReader = new PHPExcel_Reader_Excel2003XML();
$objReader -> setReadFilter (new MyReadFilter());
$objPHPExcel = $objReader -> load("06largescale.xml");
```

## 6.5. Enlace simbólico (SYLK)

Enlace simbólico (SYLK) es un formato de archivo de Microsoft suelen utilizado para intercambiar datos entre aplicaciones, especialmente hojas de cálculo. Archivos SYLK convencionalmente tienen un sufijo .slk. Compuesto por sólo pueden mostrables caracteres ANSI, puede ser fácilmente creado y procesado por otras aplicaciones, como bases de datos.

### ♥ Limitaciones SYLK

Tenga en cuenta que el formato SYLK tiene algunos límites en cuanto a estilo células y manejo de hojas de cálculo grandes mediante PHP.

### 6.5.1. PHPExcel\_Reader\_SYLK

#### Leer una hoja de cálculo

Puede leer un archivo .slk utilizando el siguiente código:

```
$objReader = new PHPExcel_Reader_SYLK();  
$objPHPExcel = $objReader -> load("05featuredemo.slk");
```

#### Células específicas de lectura solamente

Puede establecer la opción setReadFilter en el lector, para instruir al lector a cargar sólo las células que coincida con una regla dada. Un filtro de lectura puede ser cualquier clase que implementa PHPExcel\_Reader\_IReadFilter. De forma predeterminada, todas las células se leen usando el PHPExcel\_Reader\_DefaultReadFilter.

El siguiente código sólo leerá fila 1 y 20 - 30 filas de cualquier hoja en el archivo SYLK:

```
class MyReadFilter implements PHPExcel_Reader_IReadFilter  
{  
    readCell de la función pública ($columna, $row, $worksheetName = '') {}  
    / / Leen título filas y filas de 20-30  
    Si ($row == 1 || ($row >= 20 && $row <= 30)) {  
        devuelve true;  
    }  
  
    devolver false;  
}  
  
$objReader = new PHPExcel_Reader_SYLK();  
$objReader -> setReadFilter (new MyReadFilter());  
$objPHPExcel = $objReader -> load("06largescale.slk");
```

## 6.6. Oficina abierta/Libre (.ods)

Abrir archivos .ods oficina o Libre son los estándar de archivo fopr Open Office o Libre Office Calc archivos en formato.

### 6.6.1. PHPExcel\_Reader\_OOCalc

#### Leer una hoja de cálculo

Puede leer un archivo .ods utilizando el siguiente código:

```
$objReader = new PHPExcel_Reader_OOCalc();  
$objPHPExcel = $objReader -> load("05featuredemo.ods");
```

#### Células específicas de lectura solamente

Puede establecer la opción setReadFilter en el lector, para instruir al lector a cargar sólo las células que coincida con una regla dada. Un filtro de lectura puede ser cualquier clase que implementa PHPExcel\_Reader\_IReadFilter. De forma predeterminada, todas las células se leen usando el PHPExcel\_Reader\_DefaultReadFilter.

El siguiente código sólo leerá la fila 1 y filas 20 - 30 de cualquier hoja en el archivo de Calc:

```
class MyReadFilter implements PHPExcel_Reader_IReadFilter
```

```
{
readCell de la función pública ($columna, $row, $worksheetName = '') {}
// Leen título filas y filas de 20-30
Si ($row == 1 || ($row >= 20 && $row <= 30)) {
devuelve true;
}

devolver false;
}
}

$objReader = new PHPExcel_Reader_OOcalc();
$objReader -> setReadFilter (new MyReadFilter());
$objPHPExcel = $objReader -> load("06largescale.ods");
```

## 6.7. CSV (valores separados por comas)

CSV (Comma Separated Values) se utilizan a menudo como un formato de archivo de importación/exportación con otros sistemas. PHPExcel permite leer y escribir en archivos CSV.



### Limitaciones de CSV

Tenga en cuenta que el formato de archivo CSV tiene algunos límites en cuanto a estilo celdas, formato, número...

### 6.7.1. PHPExcel\_Reader\_CSV

#### Lectura de un archivo CSV

Puede leer un archivo .csv usando el siguiente código:

```
$objReader = new PHPExcel_Reader_CSV();
$objPHPExcel = $objReader -> load("sample.csv");
```

#### Configuración de opciones de CSV

A menudo, no son realmente "separados por comas" archivos CSV o usar punto y coma (;) como separador. Puede indicar PHPExcel\_Reader\_CSV algunas opciones antes de leer un archivo CSV.

Tenga en cuenta que PHPExcel\_Reader\_CSV por defecto se asume que el archivo CSV cargado es codificado en UTF-8. Si estás leyendo archivos CSV que fueron creados en Microsoft Office Excel la codificación correcta de entrada puede ser bastante Windows-1252 (CP1252). Asegúrese siempre de que la codificación de entrada está ajustada correctamente.

```
$objReader = new PHPExcel_Reader_CSV();
$objReader -> setInputEncoding('CP1252');
$objReader -> setDelimiter(';');
$objReader -> setEnclosure ('');
$objReader -> setLineEnding("\r\n");
$objReader -> setSheetIndex(0);
$objPHPExcel = $objReader -> load("sample.csv");
```

#### Leer una hoja de cálculo específico

Archivos CSV sólo pueden contener una hoja de cálculo. Por lo tanto, puede especificar qué hoja leer desde CSV:

```
$objReader -> setSheetIndex(0);
```

#### Leer en hoja de cálculo existente

Cuando trabaje con archivos CSV, puede ocurrir que desea importar datos CSV en un objeto PHPExcel existente. El código siguiente carga un archivo CSV en un \$objPHPExcel existente que contiene unas hojas y las importaciones sobre la hoja de<sup>th</sup> 6:

```
$objReader = new PHPExcel_Reader_CSV();
$objReader -> setDelimiter(';');
$objReader -> setEnclosure ('');
$objReader -> setLineEnding("\r\n");
$objReader -> setSheetIndex(5);
$objReader -> loadIntoExisting ("05featuredemo.csv", $objPHPExcel);
```

## 6.7.2. PHPExcel\_Writer\_CSV

### Escribir un archivo CSV

Puede escribir un archivo .csv usando el siguiente código:

```
$objWriter = new PHPExcel_Writer_CSV($objPHPExcel);
$objWriter -> save("05featuredemo.csv");
```

### Configuración de opciones de CSV

A menudo, no son realmente "separados por comas" archivos CSV o usar punto y coma (;) como separador. Puede indicar a PHPExcel\_Writer\_CSV algunas de las opciones antes de escribir un archivo CSV:

```
$objWriter = new PHPExcel_Writer_CSV($objPHPExcel);
$objWriter -> setDelimiter(';');
$objWriter -> setEnclosure ('');
$objWriter -> setLineEnding("\r\n");
$objWriter -> setSheetIndex(0);
$objWriter -> save("05featuredemo.csv");
```

### Escribir una hoja de cálculo específico

Archivos CSV sólo pueden contener una hoja de cálculo. Por lo tanto, puede especificar que la hoja para escribir a CSV:

```
$objWriter -> setSheetIndex(0);
```

### Fórmula cálculo previo

De forma predeterminada, este escritor calcula previamente todas las fórmulas en la hoja de cálculo. Esto puede ser lento en grandes hojas de cálculo y tal vez no deseados. Sin embargo puede desactivar el cálculo de la fórmula:

```
$objWriter = new PHPExcel_Writer_CSV($objPHPExcel);
$objWriter -> setPreCalculateFormulas(false);
$objWriter -> save("05featuredemo.csv");
```

### Escribir archivos CSV UTF-8

Un archivo CSV puede marcarse como UTF-8 escribiendo un encabezado de archivo BOM. Esto puede ser activado usando el siguiente código:

```
$objWriter = new PHPExcel_Writer_CSV($objPHPExcel);
$objWriter -> setUseBOM(true);
$objWriter -> save("05featuredemo.csv");
```

### Decimales y miles separadores

Si la hoja de cálculo que está exportando contiene números con decimales o miles separadores entonces deberías pensar en qué personajes te quiere usar para aquellos que antes de realizar la exportación.

De forma predeterminada PHPExcel busca en configuración regional del servidor para decidir qué caracteres para utilizar. Pero para evitar problemas se recomienda establecer los caracteres explícitamente como se muestra a continuación.

Inglés usuarios querrán usar esto antes de hacer la exportación:

```
require_once 'PHPExcel/Shared/String.php'
PHPExcel_Shared_String::setDecimalSeparator('.');
PHPExcel_Shared_String::setThousandsSeparator(',');
```

Los usuarios alemanes querrán utilizar los valores opuestos.



```
require_once 'PHPExcel/Shared/String.php'
PHPExcel_Shared_String::setDecimalSeparator(',');
PHPExcel_Shared_String::setThousandsSeparator('.');
```

Tenga en cuenta que el código anterior establece decimales y separadores de mil opciones como globales. Esto también afecta a cómo se exporta HTML y PDF.

## 6.8. HTML

PHPExcel permite leer o escribir una hoja de cálculo en formato HTML, para la representación rápida de los datos en él a alguien que no tiene una hoja de cálculo en su PC, o cargar archivos salvados por otros scripts que simplemente crean código HTML y le dan una extensión de archivo .xls.

### 💡 Limitaciones de HTML

Tenga en cuenta que el formato de archivo HTML tiene algunos límites en cuanto a estilo celdas, formato, número...

### 6.8.1. PHPEXcel\_Reader\_HTML

#### Leer una hoja de cálculo

Puede leer un archivo .html o .htm mediante el siguiente código:

```
$objReader = new PHPEXcel_Reader_HTML();
$objPHPExcel = $objReader->load("05featuredemo.html");
```

### 💡 Limitaciones de HTML

Tenga en cuenta que lector HTML aún es experimental y no todavía admite las celdas combinadas ni tablas anidadas limpiamente

### 6.8.2. PHPEXcel\_Writer\_HTML



Tenga en cuenta que PHPEXcel\_Writer\_HTML sólo los resultados de la primera hoja de cálculo predeterminada.

#### Escribir una hoja de cálculo

Puede escribir un archivo .htm usando el siguiente código:

```
$objWriter = new PHPEXcel_Writer_HTML($objPHPExcel);
$objWriter->save("05featuredemo.htm");
```

#### Escriba todas las hojas de cálculo

Los archivos HTML pueden contener uno o más hojas de cálculo. Si quieres escribir todas las hojas en un solo archivo HTML, utilice el siguiente código:

```
$objWriter->writeAllSheets();
```

#### Escribir una hoja de cálculo específico

Los archivos HTML pueden contener uno o más hojas de cálculo. Por lo tanto, puede especificar que la hoja para escribir en HTML:

```
$objWriter->setSheetIndex(0);
```

#### Ajuste de la raíz de imágenes del archivo HTML

Allí podría ser situaciones donde desee establecer explícitamente la raíz imágenes incluidas.

Por ejemplo, uno podría querer ver `<img style = "posición: relativa; izquierda: 0px; superior: 0px; ancho: 140px; altura: 78px;" src = "http://www.domain.com/images/logo.jpg" frontera = "0" >` en vez de `<img style = "posición: relativa; izquierda: 0px; superior: 0px; ancho: 140px; altura: 78px;" src = "../images/logo.jpg" frontera = "0" >`.

Para lograr este resultado puede utilizar el código siguiente:

```
$objWriter->setImagesRoot('http://www.example.com');
```

## Fórmula cálculo previo

De forma predeterminada, este escritor calcula previamente todas las fórmulas en la hoja de cálculo. Esto puede ser lento en grandes hojas de cálculo y tal vez no deseados. Sin embargo puede desactivar el cálculo de la fórmula:

```
$objWriter = new PHPExcel_Writer_HTML($objPHPExcel);  
$objWriter -> setPreCalculateFormulas(false);  
$objWriter -> save("05featuredemo.htm");
```

## Incrustar genera HTML en una página web

Puede haber una situación donde quieres insertar el código HTML generado en un sitio web existente. PHPExcel\_Writer\_HTML proporciona soporte para generar sólo partes específicas del código HTML, que permite utilizar estas piezas en su sitio de Internet.

Métodos soportados:

- generateHTMLHeader()
- generateStyles()
- generateSheetData()
- generateHTMLFooter()

Aquí hay un ejemplo que recupera todas las partes de forma independiente y los combina en una página HTML resultante:

```
<? php  
$objWriter = new PHPExcel_Writer_HTML($objPHPExcel);  
echo $objWriter -> generateHTMLHeader();  
?>  
  
< style >  
<!--  
HTML {}  
font-family: Times New Roman;  
font-size: 9pt;  
background-color: white;  
}  
  
<? php  
echo $objWriter -> generateStyles(false); / / No escriba < style > y < /  
estilo >  
?>  
  
-->  
< / estilo >  
  
<? php  
echo $objWriter -> generateSheetData();  
echo $objWriter -> generateHTMLFooter();  
?>
```

## Escribir archivos HTML UTF-8

Un archivo HTML puede ser marcado como UTF-8 por escribir un encabezado de archivo BOM. Esto puede ser activado usando el siguiente código:

```
$objWriter = new PHPExcel_Writer_HTML($objPHPExcel);  
$objWriter -> setUseBOM(true);  
$objWriter -> save("05featuredemo.htm");
```

## Decimales y miles separadores

Consulte Sección PHPExcel\_Writer\_CSV cómo controlar la aparición de estos.

## 6.9. PDF

PHPExcel le permite escribir una hoja de cálculo en formato PDF, para la rápida distribución de datos mencionadas.



#### Limitaciones de PDF

Tenga en cuenta que el formato de archivo PDF tiene algunos límites en cuanto a estilo celdas, formato, número...

### 6.9.1. PHPEXcel\_Writer\_PDF

PDF Writer de PHPEXcel es un contenedor para un 3<sup>rd</sup>-biblioteca partido PDF Rendering como tcPDF, mPDF o DomPDF. Antes de la versión puntos 1.7.8 de PHPEXcel, la librería tcPDF estaba integrada con PHPEXcel; Pero desde la versión puntos 1.7.8 esto fue quitada. Por el contrario, debe ahora instalar una biblioteca PDF Rendering Pero PHPEXcel trabajará con un número de diferentes bibliotecas.

En la actualidad, se soportan las siguientes bibliotecas:

Biblioteca	Versión usada para la prueba	Descargable desde	PHPEXcel interna constante
tcPDF	5.9	<a href="http://www.TCPDF.org/">http://www.TCPDF.org/</a>	PDF_RENDERER_TCPDF
mPDF	5.4	<a href="http://www.mpdf1.com/mpdf/">http://www.mpdf1.com/mpdf/</a>	PDF_RENDERER_MPDF
domPDF	0.6.0 beta 3	<a href="http://code.google.com/p/dompdf/">http://code.google.com/p/dompdf/</a>	PDF_RENDERER_DOMPDF

Las bibliotecas diferentes tienen diferentes fortalezas y debilidades. Algunos generan resultados formateados mejor que otros, algunos son más rápidos o utilizan menos memoria que otros, mientras algunos generan archivos .pdf más pequeños. Es la elección de los desarrolladores que se desean utilizar, adecuado a sus propias circunstancias.

Antes de crear instancias de un escritor para generar la salida PDF, tendrá que indicar qué biblioteca de renderizado está utilizando, y donde se encuentra.

```

$rendererName = PHPEXcel_Settings::PDF_RENDERER_MPDF;
$rendererLibrary = 'mPDF5.4';
$rendererLibraryPath = dirname(__FILE__).'/../../Libraries/PDF /'.
$rendererLibrary;
if (!PHPEXcel_Settings::setPdfRenderer(
$rendererName,
$rendererLibraryPath
)) {
morir)
'Configure los valores $rendererName y $rendererLibraryPath'.
PHP_EOL.
'lo más apropiado para la estructura de directorios'
);
}

```

### Escribir una hoja de cálculo

Una vez que haya identificado el renderizador que desea utilizar para la generación de PDF, puede escribir un archivo .pdf usando el siguiente código:

```

$objWriter = new PHPEXcel_Writer_PDF($objPHPEXcel);
$objWriter -> save("05featuredemo.pdf");

```



Tenga en cuenta que PHPEXcel\_Writer\_PDF sólo los resultados de la primera hoja de cálculo predeterminada.

### Escriba todas las hojas de cálculo

Los archivos PDF pueden contener uno o más hojas de cálculo. Si quieres escribir todas las hojas en un solo archivo PDF, utilice el siguiente código:

```

$objWriter -> writeAllSheets();

```

### **Escribir una hoja de cálculo específico**

Los archivos PDF pueden contener uno o más hojas de cálculo. Por lo tanto, puede especificar que la hoja para escribir en PDF:

```
$objWriter -> setSheetIndex(0);
```

### **Fórmula cálculo previo**

De forma predeterminada, este escritor calcula previamente todas las fórmulas en la hoja de cálculo. Esto puede ser lento en grandes hojas de cálculo y tal vez no deseados. Sin embargo puede desactivar el cálculo de la fórmula:

```
$objWriter = new PHPExcel_Writer_PDF($objPHPExcel);  
$objWriter -> setPreCalculateFormulas(false);  
$objWriter -> save("05featuredemo.pdf");
```

### **Decimales y miles separadores**

Consulte Sección PHPExcel\_Writer\_CSV cómo controlar la aparición de estos.

## **6.10. Archivos de Excel generando de plantillas (leer, modificar, escribir)**

Lectores y escritores son las herramientas que le permiten generar archivos Excel desde plantillas. Esto requiere menos esfuerzo de codificación que generar el archivo de Excel desde cero, especialmente si la plantilla tiene muchos estilos, las propiedades de configuración de página, encabezados etc..

Aquí hay un ejemplo de cómo abrir un archivo de plantilla, rellenar un par de campos y guardarlo otra vez:

```
$objPHPExcel = PHPExcel_IOFactory::load('template.xlsx');  
  
$objWorksheet = $objPHPExcel -> getActiveSheet();  
$objWorksheet -> getCell('A1') -> setValue('John');  
$objWorksheet -> getCell('A2') -> setValue('Smith');  
  
$objWriter = PHPExcel_IOFactory::createWriter ($objPHPExcel, 'Excel5');  
$objWriter -> save('write.xls');
```

Observe que está bien para cargar un archivo xlsx y generar un archivo xls.

## 7. Créditos

Por favor consulte la página de internet

<http://www.codeplex.com/PHPExcel/Wiki/View.aspx?title=Credits&referringTitle=Home> para créditos hasta la fecha.

## Apéndice A: claves de matriz válida para estilo applyFromArray()

La tabla siguiente enumeran los keys válidas para las clases de applyFromArray() de PHPExcel\_Style. Si la columna "Se asigna a la propiedad" asigna una clave a un setter, se aplicará directamente el valor de esa clave. Si la columna "Se asigna a la propiedad" asigna una clave a un comprador, se aplicará el valor de esa clave como matriz de otro estilo.

PHPExcel_Style	
Clave de la matriz:	Se asigna a la propiedad:
relleno	getFill()
fuentes	getFont()
fronteras	getBorders()
alineación	getAlignment()
NumberFormat	getNumberFormat()
protección	getProtection()
PHPExcel_Style_Fill	
Clave de la matriz:	Se asigna a la propiedad:
tipo	setFillType()
rotación	setRotation()
StartColor	getStartColor()
EndColor	getEndColor()
Color	getStartColor()
PHPExcel_Style_Font	
Clave de la matriz:	Se asigna a la propiedad:
nombre	setName()
negrita	setBold()
cursiva	setItalic()
subrayado	setUnderline()
huelga	setStrikethrough()
Color	getColor()
tamaño	método setSize()
Superíndice	setSuperScript()
Subíndice	setSubScript()
PHPExcel_Style_Borders	
Clave de la matriz:	Se asigna a la propiedad:
allborders	getLeft(); getRight(); getTop(); getBottom()
izquierda	getLeft()
derecho	getRight()
Arriba	getTop()
parte inferior	getBottom()
diagonal	getDiagonal()
vertical	getVertical()
horizontal	getHorizontal()
diagonaldirection	setDiagonalDirection()
esquema	setOutline()
PHPExcel_Style_Border	
Clave de la matriz:	Se asigna a la propiedad:
estilo	setBorderStyle()
Color	getColor()
PHPExcel_Style_Alignment	
Clave de la matriz:	Se asigna a la propiedad:

horizontal	setHorizontal()
vertical	setVertical()
rotación	setTextRotation()
abrigo	setWrapText()
shrinkToFit	setShrinkToFit()
guión	setIndent()
<b>PHPExcel_Style_NumberFormat</b>	
Clave de la matriz:	Se asigna a la propiedad:
Código	setFormatCode()
<b>PHPExcel_Style_Protection</b>	
Clave de la matriz:	Se asigna a la propiedad:
bloqueado	setLocked()
escondido	setHidden()

[1] Debe tener habilitado para que PHP utilizar esta opción APC.

[2] Debe tener un servidor memcache y han permitido memcache para tu PHP utilizar esta opción.

[3] Debe tener habilitado para que PHP utilizar esta opción de Wincache.