

Actividad práctica

Procesamiento de archivos CSV en PHP

Introducción

El formato CSV (Comma-Separated Values) se ha convertido en uno de los estándares más utilizados **para la representación y el intercambio de datos tabulares**. Se caracteriza por su simplicidad y facilidad de uso, permitiendo que los datos se estructuren en forma de filas y columnas, donde cada valor está separado por comas (o, en algunos casos, otros delimitadores como punto y coma o tabulaciones).

Este formato es ampliamente utilizado en diversas aplicaciones, desde la gestión de bases de datos hasta la exportación de datos desde hojas de cálculo. En esta actividad, vas a aprender cómo procesar este tipo de archivos en PHP.

Ejemplo de archivo cvs

```
"OrderID", "CustomerID", "EmployeeID", "OrderDate"  
"10248", "VINET", "5", "7/4/1996"  
"10249", "TOMSP", "6", "7/5/1996"  
"10250", "HANAR", "4", "7/8/1996"  
"10251", "VICTE", "3", "7/8/1996"  
"10252", "SUPRD", "4", "7/9/1996"
```

Fíjate que todo archivo cvs tiene una cabecera de campos y un carácter separador de campo. En este caso es el carácter ','

Funciones relacionadas con la lectura de archivos csv

fopen()

fopen es una función de PHP que se utiliza para **abrir un archivo o una URL**. Permite establecer una conexión con el archivo y **devolver un puntero de archivo** que se puede utilizar para realizar operaciones de lectura o escritura.

```
$handle = fopen('libros.csv', 'r');
```

Devuelve un puntero de archivo en caso de éxito, o false si falla.

fgetcsv()

fgetcsv es una función de PHP que se utiliza para **leer una línea de un archivo y analizarla como un CSV**. Esta función es muy útil para extraer datos de archivos CSV de manera fácil y eficiente.

Devuelve un array con los campos leídos desde la línea del archivo. Si se llega al final del archivo, devuelve false.

Ejemplo común de uso de fgetcsv

```
$cabecera = fgetcsv($handle); // quitamos la cabecera

// Procesar el resto del archivo CSV
while (($datos = fgetcsv($handle)) !== false) {
    // procesar cada línea de datos
    // $datos[0], $datos[1], etc.
}
```

NOTA: Como ves, fgetcsv se mete en un bucle donde la condición de iteración será verdadera mientras haya líneas que en el archivo csv representado por \$handle

Desarrollo de la actividad práctica

En esta práctica vamos a procesar los campos de un archivo csv subido al servidor.

Para ello sigue las siguientes indicaciones para cada uno de los siguiente componentes del proyecto:

index.html

Desarrolla un formulario html de subida de archivos como has hecho en las anteriores actividades.

Dataset de Libros de Informática

Este dataset muestra en formato csv información sobre libros de informática clásicos. Guarda esta información en un archivo y ponle extensión cvs

```
"title","author","year","genre"
```

```
"Clean Code","Robert C. Martin",2008,"Programming"  
"The Pragmatic Programmer","Andrew Hunt",1999,"Programming"  
"Design Patterns","Erich Gamma",1994,"Software Engineering"  
"Introduction to Algorithms","Thomas H. Cormen",2009,"Algorithms"  
"PHP Objects, Patterns, and Practice","Mika Schwartz",2013,"PHP"
```

Modulo procesar_csv.php

Queremos poner en práctica la lectura de archivos y procesar cada campo. Para ello debes desarrollar el siguiente módulo php siguiendo las instrucciones

```
<?php  
  
Aplica las partes explicadas de como procesar el archivo para conseguir mandar  
a la salida el título del libro y su autor solamente del data set del ejemplo.  
  
// 1.-Guarda el archivo subido a la ruta local ./archivos/  
  
//2.-Lee el archivo subido y procesa cada línea , manda a la salida el título y  
autor solamente
```

Se pide :

1. Implementa un proyecto web que realice la funcionalidad descrita en la práctica usando los módulos indicados. **Omite la comprobación de errores y usa exclusivamente las funciones explicadas.**
2. Todo esta manera de acceder al archivo está condicionado a que el archivo sea cvs . ¿Qué añadirías en el código para comprobar que el archivo es csv? ¿Dónde realizarías la comprobación en el cliente o servidor?