

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**NOMBRES: EVELYN ANAGUMBLA**

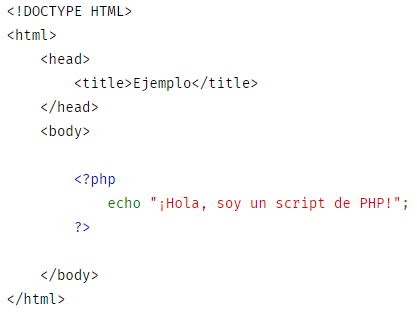
**FECHA: 2019-04-07**

**PROGRAMACIÓN AVANZADA**

**DEBER N°2**

**PHP**

Hypertext Preprocessor es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.



Utiliza código que contiene HTML, cada comando realiza una acción en especifico, !). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final <?php y ?> que permiten entrar y salir del "modo PHP".

**MÉTODOS**

Existen dos métodos HTTP que un cliente puede utilizar para pasar los datos del formulario al servidor: GET y POST. El método que utiliza un formulario particular, se especifica con el atributo method en la etiqueta form. En teoría, los métodos son sensibles a mayúsculas en el código HTML, pero en la práctica algunos navegadores fallan si el el nombre del método no está en mayúsculas.

A solicitud GET codifica los parámetros del formulario en la dirección URL en lo que se llama una cadena de consulta, el texto que sigue al carácter?

Es la cadena de consulta: **/path/to/page.php?keyword=bell&length=3**

Una solicitud POST pasa los parámetros del formulario en el cuerpo de la solicitud HTTP, dejando intacta la URL. El tipo de método que se utilizó para solicitar una página PHP está disponible a través de $\_SERVER[“REQUEST\_METHOD”]. Por ejemplo:

**if ($\_SERVER['REQUEST\_METHOD'] == 'GET') {**

**// handle a GET request**

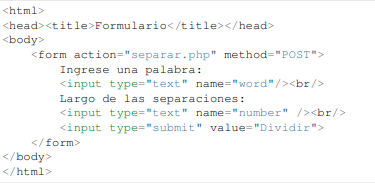
**} else {**

**die("You may only GET this page.");**

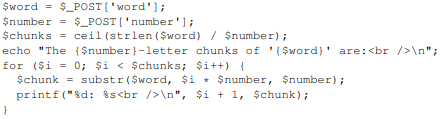
**}**

**PARÁMETROS**

Se utilizan los arreglos \_POST, \_GET y \_FILES para acceder a los parámetros de formulario desde el código PHP. Las llaves son los nombres de los parámetros y los valores son los valores de esos parámetros. Por ejemplo, considere la siguiente página utilizada para separar una palabra:



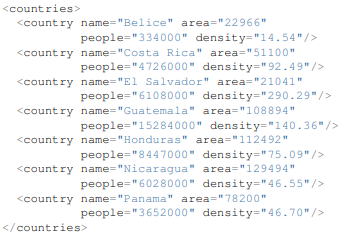
El programa PHP para procesar dicho formulario sería el siguiente:



**MANIPULANDO XML Y JSON**

**Lectura de datos XML**

El método XMLReader::open, utilizado como método estático, establece el URL de entrada para el contenido XML que se procesará y XMLReader::close cierra dicha entrada. La función XMLReader::read mueve al siguiente nodo en el documento. Por su parte la función XMLReader::next mueve el cursor al siguiente nodo saltándose todos los subárboles.

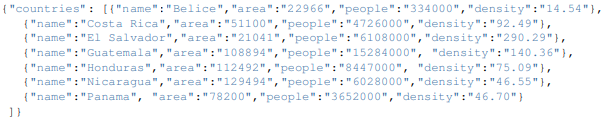


**Escritura de datos mediante XML**

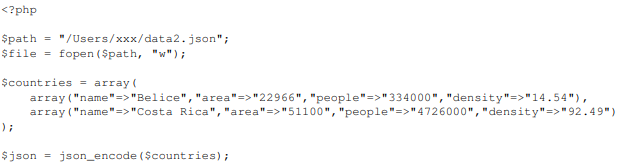
La clase XMLWriter crea un objeto de escritura XML, y el método XMLWriter::openURI(path) establece el URI de salida para el contenido XML que se generará y XMLReader::flush escribe dicha entrada. La función XMLReader::startDocument crea el nodo principal de un documento xml, y el método XMLReader::endDocument finaliza el documento. Por su parte la función XMLReader::startElement crea un elemento XML y el método XMLReader::endElement lo finaliza.

**Lectura de datos JSON**

El método utilizado por PHP para tratar datos JSON es simplemente convertir hileras de texto (string) en formato JSON a arreglos de PHP. Para ello se utiliza la función json\_decode(string) la cual recibe dicha hilera de texto y retorna el arreglo. Por ejemplo considere el siguiente archivo de datos en formato JSON:



**Escritura de datos JSON**



**BIBLIOGRAFÍA**

PHP, P. (s.f.). *buildmedia.readthedocs.org*. Obtenido de buildmedia.readthedocs.org: https://buildmedia.readthedocs.org/media/pdf/programacion-php/latest/programacion-php.pdf