

## UD2. Pt2 – Programación basada en código embebido. Sintaxis básica PHP (2)

Los siguientes apartados servirán para poner en práctica las librerías de PHP para trabajar con cadenas de caracteres.

Programa scripts de servidor que realicen las siguientes tareas:

- 1) A partir de una cadena ya dada en el código, contar cuántas vocales contiene.

Intenta resolver el ejercicio de 3 formas:

- mediante **for's** recorriendo como un array `$cadena` (cadena analizada) y `$vocals` (cadena con todas las vocales, incluyendo diéresis y acentos abiertos y cerrados).

Podemos tratar como un array las 2 cadenas.

Por ejemplo si `$cadena = 'hola' → $cadena[0]` es el carácter 'h'

- mediante la función de PHP **`preg_match()`** y expresiones regulares (regex).
- mediante la función de PHP **`preg_match_all()`**

Para trabajar con expresiones regex puede servir de ayuda este tutorial:

<https://medium.com/factory-mind/regex-tutorial-a-simple-cheatsheet-by-examples-649dc1c3f285>

- 2) Resolver el mismo ejercicio que en el apartado 1, pero ahora la cadena nos llega desde un formulario Html.
- 3) A partir de una cadena de texto que nos llega desde un formulario Html, mostrar la palabra que contiene más vocales y la que contiene menos.

Se pide resolver el ejercicio declarando una función *cuentaVocales()*, y llamarla en el controlador.

Ayuda: la instrucción **`explode`** nos permite separar una cadena por espacios en blanco, dando como resultado un array de palabras.

- 4) Un formulario Html nos enviará:
  - una subcadena para realizar una búsqueda.
  - una cadena de texto.

El script de PHP:

- buscará todas las palabras de la cadena de texto que contienen la subcadena de búsqueda.
- mostrará en pantalla los resultados separados por comas.

Por ejemplo, buscamos «ti» en la cadena «en el instituto Abastos se estudian ciclos de la familia de Administración y de la familia de Informàtica».

El script generará como respuesta:

Se han encontrado las siguientes palabras que contienen la subcadena «ti»: instituto, Informática (elimina la coma final si hay varios resultados).

Se pide resolver el ejercicio declarando una función *contieneSubcadena (\$cadena, \$subcadena)*,

que retorne un booleano, y llamarla en el controlador.

5) Vuelve a resolver el ejercicio anterior, efectuando algunas mejoras:

- el buscador no será sensible a mayúsculas.
- tampoco será sensible a acentos.
- además de separar palabras por espacios en blanco, separará por comas.

Por ejemplo, si buscamos «dÁ» en la cadena «los ciclos de informática de grado superior son DAM, DAW y ASIR»

, saldrá como resultado «DAM, DAW» (y no «DAM,, DAW» que saldría si solo separara por espacios.)

**ENTREGA DE LA PRÁCTICA: carpeta comprimida con los ficheros de código fuente de cada apartado (una subcarpeta por apartado).**