

# PHP - Encriptació d'Strings

## DAW-MP07-UF1 - Exercici de Desenvolupament web en entorn servidor.

En un fitxer del servidor ens trobem amb el següent codi. Sabem que les dues cadenes de caràcters s'han encriptat mitjançant la següent codificació:

1. Es divideix el text amb cadenes de 3 caràcters. A cada tercet s'inverteix l'ordre dels caràcters, de manera que "abc" passa a ser "cba".
2. Remplecem els caràcters alfabètics per el seu oposat, de manera que 'a' passa a ser 'z', 'b' passa a ser 'y'... Els caràcters no alfabètics es mantenen.

```
<?php
```

```
$sp = "kfhxivrozziuortghrvxrrkcrozxlwflrh";  
$mr = " hv ovxozwozv vj o vfrfjvivfj h vmzvlo e hrxvhlmov oz ozx.vw z xve hv  
loqvn il hv lmnlg izxvwrhrvml ,hv b lh mv,rhhv mf w zrxvlrh.m";
```

```
echo decrypt($sp);  
echo "<br>";  
echo decrypt($mr);
```

```
?>
```

## Activitats

1. Crea la funció per descriptar els diferents textos. *Recomenable fer una ullada a les [funcions de tractament d'strings](#)*

El codi php complet de la funció és el següent:

```
/**
 * Creem la funció decrypt on hi passem per paràmetre l'string
 */
function decrypt($str) {
    /**
     * Variable amb la que indicarem que agafem trossos
     * de la cadena de 3 en 3
     */
    $mida = 3;

    /**
     * Les variables del primer bucle
     */
    $comptador = 0;
    $arrayFragmentat = array();
    $cadenaRevertida = "";

    /**
     * Les variables del segon bucle
     */
}
```

```
$segonComptador = 0;

$cadenaFinal= "";

$posicio = 0;

/**

 * Aquí guardem la cadena dins d'un array i el
 * particionem de tres en tres
 */

$arrayFragmentat = str_split ($str, $mida);

/**

 * En aquest bucle, per cada tros que es recorre, es reverteix
 * i els trossos revertits es van acumulant en una nova variable
 * anomenada $cadenaRevertida
 */

for ($comptador = 0; $comptador < count($arrayFragmentat);
$comptador++) {

    $cadenaRevertida .= strrev($arrayFragmentat[$comptador]) ;

}

/**

 * Un cop revertida la cadena, substituïm els caràcters alfabètics
 * trobats pel seu oposat en aquest segon bucle
 */

for ($segonComptador = 0; $segonComptador <
strlen($cadenaRevertida); $segonComptador++) {
```

```
/**  
  
 * En una nova variable anomenada $posicio obtenim el número  
  
 * del caràcter dins la taula ASCII  
  
 */  
  
$posicio = ord($cadenaRevertida[$segonComptador]);  
  
  
/**  
  
 * En cas de que aquest número estigui entre el 97 i el 122,  
  
 * és a dir, sigui [a-z], el substituïm pel seu oposat,  
  
 * transformem el número obtingut de la taula ASCII en el  
  
 * caràcter corresponent i una nova variable anomenada  
  
 * $cadenaFinal els va acumulant.  
  
 */  
  
if ($posicio >= 97 | $posicio <= 122) {  
  
    $posicioObtinguda = 122 -  
(ord($cadenaRevertida[$segonComptador]) - 97);  
  
    $cadenaFinal .= chr($posicioObtinguda);  
  
}  
  
  
/**  
  
 * En cas contrari, simplement es mantenen els caràcters i  
  
 * també es van acumulant a la nova variable  
  
 */  
  
else {  
  
    $posicioObtinguda = ord($cadenaRevertida[$segonComptador]);
```

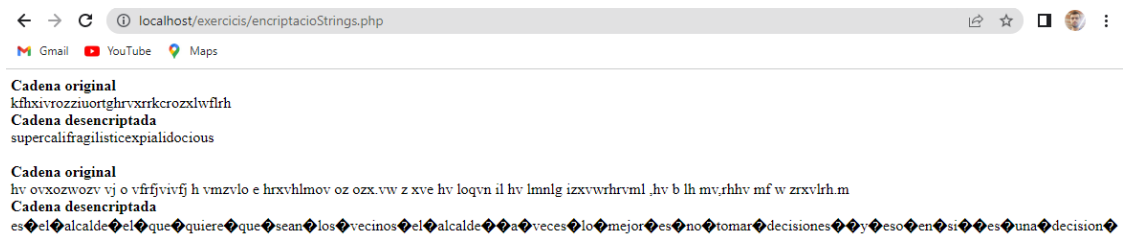
```
        $cadenaFinal .= chr($posicioObtinguda);

    }

}

/**
 * Finalment, retornem la cadena obtinguda
 */

return $cadenaFinal;
}
```



Aquí comprovem i verifiquem que funciona correctament.

2. El sistema proposat per encriptar és poc segur i una mica rudimentari. Busca una solució segura per encriptar i desenscriptar text amb php. Explica breument com funciona, i mostra un exemple del seu funcionament.

Amb la solució que he implementat necessitem els següents requisits:

- La **clau** o **contrasenya** per quan vulguem encriptar i/o desenscriptar.
- El **IV** o **Vector d'inicialització**. Consisteix en un conjunt de caràcters que s'utilitza juntament amb la clau a l'hora d'encriptar i/o desenscriptar. Per millorar la seguretat, convé sempre que aquest número:
  - Sigui únic
  - O sinó aleatori perquè, durant el procés, només s'utilitzi una vegada i cada vegada que es torni a fer el procés de xifratge o desxifratge sigui un número diferent.
- El mètode de xifratge de les dades que utilitzarem és **AES-256**. Recordem que el 256 defineix la força de xifratge en format de bits o la quantitat de combinacions úniques, el qual el fa un dels mètodes de

xifratge més segurs del món (també existeixen els de 128 o 192 bits, però són més febles).

- **base64** per codificar de binari a text que representa dades binàries que formen part de la taula ASCII. Com a mètode per utilitzar únicament caràcters d'aquesta taula, és també el més segur.
- Amb les eines esmentades anteriorment podrem encriptar i/o desencriptar text de forma segura.

Com veus en el codi que he ficat a sota que he implementat com a solució, utilitzo una funció tant per encriptar com per desencriptar. Que porten la \$ a davant perquè es declarin com a anònimes, mètode que fa que no haguem d'enviar a cada funció les dades d'enciptació.

Per altra part, com he dit anteriorment tenim l'opció de canviar el valor de la variable \$iv o utilitzar la funció getIV per generar-ne un de diferent.

El codi PHP hauria de ser el següent.

```
<?php

/**
 * A continuació veuràs el procés complet per
 * encriptar i desencriptar
 */

/**
 * Aquí creem la clau de seguretat per encriptar
 * i desencriptar
 */

$clau = 'Com mes llarga sigui la cadena o clau per encriptar i
desencriptar,

millor, i evidentment, conve que la canviis de tant
en tant';

//El mètode que utilitzarem per encriptar
```

```
$method = 'aes-256-cbc';

// En podem generar una de diferent utilitzant la funció
getIV()

$iv = base64_decode("C9fBxl1EWtYTL1/M8jfstw==");

/**
 * Enviat com a un paràmetre, encriptem el text mitjançant
 * una funció
 */

$encriptar = function ($cadena) use ($method, $clau, $iv) {
    return openssl_encrypt ($cadena, $method, $clau, false,
$iv);
};

/**
 * Un cop encriptat el text, aquí el desencriptem amb una
 * altra funció
 */

$desencriptar = function ($cadena) use ($method, $clau, $iv) {
    $encrypted_data = base64_decode($cadena);
    return openssl_decrypt($cadena, $method, $clau, false,
$iv);
};

/**
```

```
* Generem un valor per IV

*/

$getIV = function () use ($method) {

    return
base64_encode(openssl_random_pseudo_bytes(openssl_cipher_iv_length($met
hod)));

};

/**

* Obtenir el nom del mètode que hem utilitzat

*/

$getNomMetode = function () use ($method) {

    return $method;

}

?>
```

Un cop tinguem la solució creada, només queda implementar-la en un fitxer per comprovar que funcioni. Implementant el codi que veuràs a continuació:

```
<?php

/**

* Implementem el fitxer php que conté totes les funcions

* i eines necessaries per realitzar el procés

* encriptar/desencriptar

*/

include "encrypt.php";
```



```
// La informació o dades de prova que utilitzarem

$dades = "Informació important que haurem de tractar";

// Aquí encriptem la informació

$dadesEncriptades = $encriptar($dades);

/**
 * I aquí la desencriptem, a més d'esmentar eines que
 * hem utilitzat
 */

$dadesDesencriptades = $desencriptar($dadesEncriptades);

echo "<b>Dades encriptades</b>: ". $dadesEncriptades .
"<br><br>";




echo "<b>Dades desencriptades</b>: ". $dadesDesencriptades .
"<br><br>";

echo "<b>IV o Vector d'Inicialització generat</b>: " . $getIV()
. "<br><br>";

echo "<b>Mètode utilitzat</b>: " . $getNomMetode();

?>
```

← → ↻ ⓘ localhost/exercicis/provaEncriptarDesencriptar.php

 Gmail  YouTube  Maps

**Dades encriptades:** zEX1WMnt5CF/9jrPCv7aOYoleHrR1LXVGmU4i+/r+IjcoGxxc+6ZOT6+7PvTdoYi

**Dades desencriptades:** Informació important que haurem de tractar

**IV o Vector d'Inicialització generat:** K9itDeRXhNPCU5QeOZ41Ew==

**Mètode utilitzat:** aes-256-cbc

Si obro el fitxer en el navegador, veurem que ens funciona correctament.

En aquests enllaços s'explica amb més profunditat la informació que he esmentat i aquest mètode que he volgut utilitzar:

<https://codigonaranja.com/como-encriptar-y-desencriptar-informacion-en-php>

<https://programacionx.net/programacion/encriptar-desencriptar-manera-simpl-e-php/>

3. Crea una tècnica d'enciptament i desenciptament pròpia i original que compleixi els diferents requisits:

- Ha de funcionar per qualsevol caràcter UTF8.

<https://www.toptal.com/php/codificacion-de-datos-una-guia-utf-8-para-php-y-mysql>

<https://es.stackoverflow.com/questions/13557/agregar-caracteres-especiales-utf-8-en-como-%C3%B1-y-acentos-en-mysqli>

<https://desarrolloweb.com/articulos/convertir-caracteres-utf-8-con-php.html>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

Per permetre els caràcters UTF-8, simplement cal que ho fem a l'etiqueta meta del fitxer HTML des d'on estan col·locades.

- El text encriptat resultant contindrà només caràcters alfanumèrics.

Com que només ha de contenir caràcters alfanumèrics, utilitzaré la `baseConvert()`.

- El sistema d'enciptació ha de dependre de l'IP d'accés, de manera que amb una IP diferent no hauriem de ser capaços d'obtenir el text encriptat.

ENLLAÇ A LA SUBCARPETA DEL GITHUB (PRÀCTICA):

<https://github.com/janEstrada24/2DAW/tree/DWES/UF1/A1>

ENLLAÇ A LA SUBCARPETA (PROGRAMES PHP):  
<https://github.com/janEstrada24/2DAW/tree/DWES/UF1/A1/PHP>