MANUAL PHP

Nombre:Aarón
Apellido: Agullo Sanchez
Curso: 2ºJ.

Índice

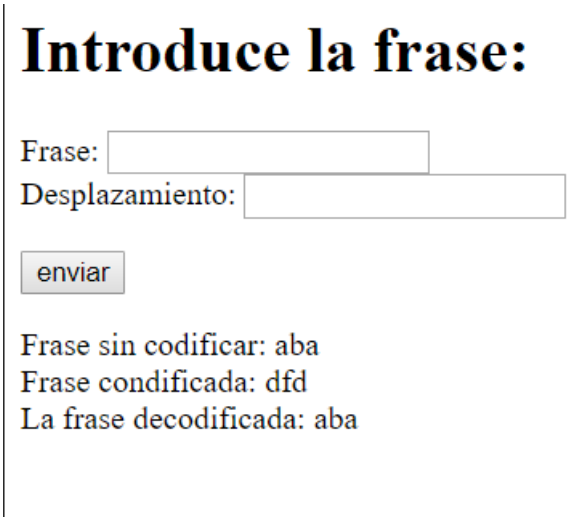
1. CIFRADO Y DESCIFRADO CON LETRAS....3
1.1. Manual de usuario....3
1.2. Manual técnico....5
2. CIFRADO Y DESCIFRADO CON NUMERO....9
2.1. Manual de usuario....9
2.2. Manual técnico....10
3. Búsqueda del tesoro....14
3.1. Manual de usuario....14
3.2. Manual Técnico....19

1. CIFRADO Y DESCIFRADO CON LETRAS
En esta pequeña aplicación vamos a cifrar y descifrar un mensaje introducido por el usuario con la codificación que el usuario quiera.

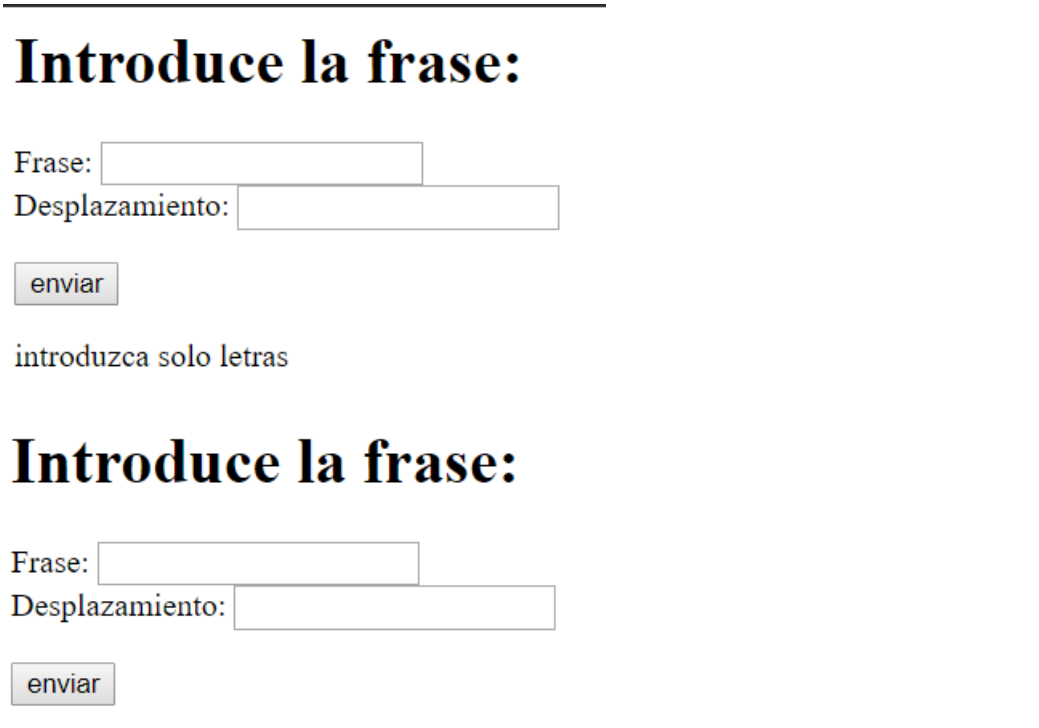
1.1. Manual de usuario.

En la pantalla que vera el usuario solo tenemos dos campos, el primero es el campo en el cual introducirá la frase que desea codificar y en el segundo la combinacional de letras para hacer la codificación. Sumaremos la posición del abecedario de las letras elegidas y como resultado nos dará una letra totalmente distinta.

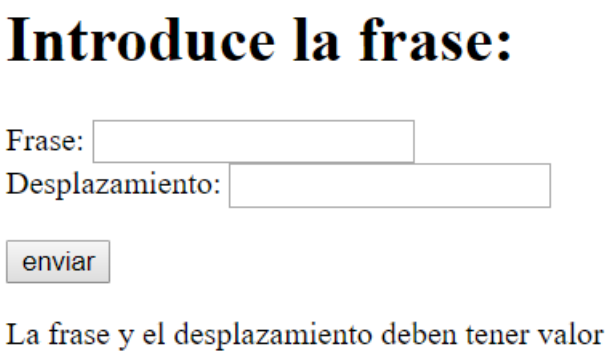
Cuando introduzcamos la frase y el desplazamiento solo le tendrá que dar a enviar y le aparecerá la frase introducida, la frase codificada y y la frase decodificada.



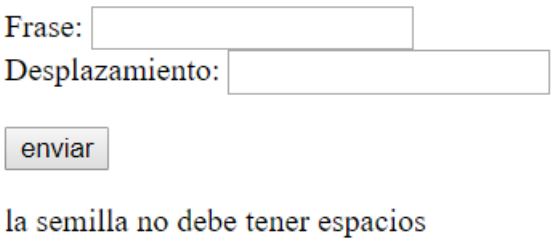
Si introducimos un desplazamiento mas corto que la frase nos dará un error, al igual que si intentamos introducir números. Para introducir otra frase solo tendremos que volverla a introducir.



La frase y la semilla deben tener algo escrito.



Pos ultimo la semilla no debe tener espacios en blanco.



1.2. Manual técnico

Lo primero que vemos en el código es el formulario en el cual recogemos tanto la frase, la semilla y el botón para poder enviar las variables que nos harán falta en este programa.

```
<html>
    <head>
        <title>"Codificacion"</title>
    </head>
<body>
<form method='POST' action='<?=$_SERVER['PHP_SELF']?>'>
    <h1>Introduce la frase: </h1>
    Frase: <input type='text' name='frase' id='frase' ><br>
    Desplazamiento: <input type='text' name='desplazamineto' id='desplazamineto' ><br>
    <p></p>
    <input type='submit' value="enviar" name="Enviar" id="Enviar" >

</form>
```

A continuación comienza nuestro código PHP, lo primero que tenemos son nuestras funciones (que las enseñaremos conforme aparezcan), tenemos el isset para comprobar que exista la frase y que comience a ejecutar nuestro programa, en primer lugar recogemos todas las variables del formulario como son la frase y el desplazamiento, creamos un array con todas las letras del abecedario en minúscula para poder trabajar con el.

```
<?php
include "funciones.php";
if (isset($_POST['frase'])) { //comprobamos que existen las variables



    $frase = strtolower($_POST['frase']);
    $desplazamiento = $_POST['desplazamineto'];
    $letra = array('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'K', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z');
```

Lo siguiente que hacemos son las comprobaciones, la primera que tenemos consiste en que tanto la frase como la semilla solo pueden contener letras, para ello hemos hecho una función (que tenemos en nuestras funciones) para que nos sea mas fácil, en esta función tenemos declarada una variable de todas las letras tanto minúscula como mayúscula, solo nos harían falta las minúsculas pero así podemos reutilizar el código, recorremos con un for la cadena pasada, con el strpos conseguimos que nos diga la primera vez que aparezca un carácter igual al permitido, con substr dándole como parámetro la frase pasada y el contador del for vamos recorriendo la frase y comparándola con la permitida, si en algún momento hay algún carácter que no sea permitido nos devolverá false, pero si son todos permitido devolverá true.

```php
function solo_letras($cadena){
    $permitidos = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ ";
    for ($i=0; $i<strlen($cadena); $i++){
        if (strpos($permitidos, substr($cadena,$i,1))===false){
            //no es válido;
            return false;
        }
    }
    //si estoy aqui es que todos los caracteres son validos
    return true;
}
```

```php
if (solo_letras($frase) == false || solo_letras($desplazamiento) == false) { //comprobacion de solo letras en la frase y en el desplazamiento
    echo "introduzca solo letras";
} else if (strlen($frase) > strlen($desplazamiento)) {//La frase debe ser mas larga o igual al desplazamiento
    echo "La frase y el desplazamiento deben tener la misma longitud o superior a la frase";
} else {
```

La siguiente condición consiste en que la frase debe ser siempre igual o inferior al desplazamiento, para ellos con strlen contamos las palabras y si la frase es mayor le notificamos el error al usuario.
También ponemos una condición para que la semilla no pueda tener espacios en blanco.

```php
} else if (strpos($desplazamiento, " ") == true) {
    echo "la semilla no debe tener espacios";
} else {
```

Y la ultima comprobación consiste en impedir que si el usuario no a escrito nada el programa no se ejecuta, con ello ahorramos muchos errores futuros.

```php
} else if ($frase == null || $desplazamiento == null) {
    echo "La frase y el desplazamiento deben tener valor";
```

A continuación simplemente imprimimos las frase para el usuario, para que tenga mas información visual.

```php
echo "Frase sin codificar: " . $frase;
echo "<br>";
echo "Frase condificada: ";
```

Lo siguiente es hacer el codificado de la frase, primero planteamos un for para que recorra la frase,
nuestra primera condición dentro del mismo sera si tenemos un espacio en blanco en la frase que ponga un espacio en blanco en la codificación.

```
for ($i = 0; $i < strlen($frase); $i++) {
    if ($frase[$i] ==" ") {
        echo " ";
        $enigma[$i] =" ";
```

Si de lo contrario esa posición de la frase no es un espacio en blanco, lo primero que tenemos que hacer es transformar la primera letra del desplazamiento en numero, transformar la primera letra de la frase a su equivalente en el codigo ascii y sumarla, le restamos 193 para que nos de la letra en el codigo ascii y hacemos el resto del array de abecedario que se llama letra, esto lo hacemos para poder darle la vuelta al array y empezar otra vez desde el principio, imprimimos la posición del array que nos de estos cálculos hacemos lo mismo con las siguientes letras.

```
//codificamos la frase
$j = 0;
for ($i = 0; $i < strlen($frase); $i++) {
    if ($frase[$i] ==" ") {
        echo " ";
        $enigma[$i] =" ";
    } else {
        $operacion = ((ord($desplazamiento[$j]) + ord($frase[$i])) - 193) % count($letra);
        echo $letra[$operacion];
        $enigma[$i] = $letra[$operacion];
        $j++;

    }
}
```

El siguiente fragmento de código se trata de la decodificación de la frase, en primer lugar ponemos la frase para mostrársela al usuario y declaramos las variables que vamos a usar.

```
echo "</br>";
echo "La frase decodificada: ";
$p = 0;
$k = 0;
```

A continuación hacemos un for para recorrer la semilla, y dentro declaramos una condición en la cual si encontramos un espacio en blanco imprimirá un espacio en blanco.

```
for ($i = 0; $p < count($enigma); $i++) {
    if ($enigma[$p] == " ") {
        echo " ";
        $p++;
    }
```

Seguimos recorriendo el for y cuando la letra de la semilla coincide con la de nuestro array del abecedario hacemos la operación de restar el valor que tiene la la letra al pasarla al código ascii y el desplazamiento, después hacemos el resto del array para poder dar vueltas al array, usamos una condición por si la operación da 0 o menos para que no de un error, por ultimo imprimimos el resultado

```
if ($letra[$i] == $enigma[$p]) {
    $opr = ((ord($enigma[$p]) - ord($desplazamiento[$k]))) % count($letra);
    if ($opr <= 0) {
        $opr = count($letra) + $opr;
    }
    echo $letra[$opr - 1];
```

Por ultimo tenemos una condición para que se mueva el desplazamiento y no de errores.

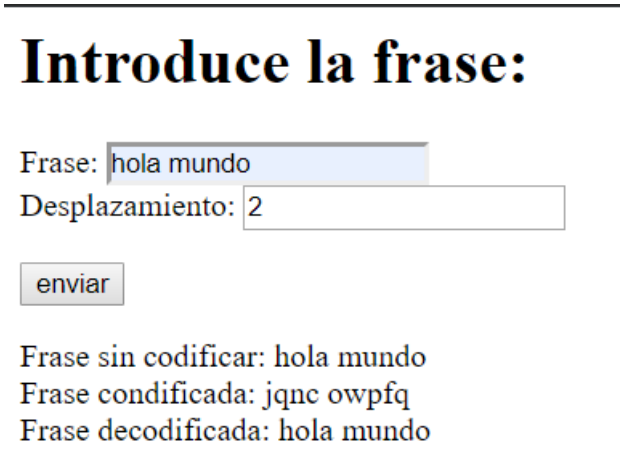
```
if ($k > strlen($desplazamiento)) {
    $k = 0;
}
```

2. CIFRADO Y DESCIFRADO CON NUMERO
En esta pequeña aplicación vamos a cifrar y descifrar un mensaje introducido por el usuario con la codificación que el usuario quiera. El programa suma la posición del abecedario de las letras introducidas y la suma con el numero introducido por el usuario.

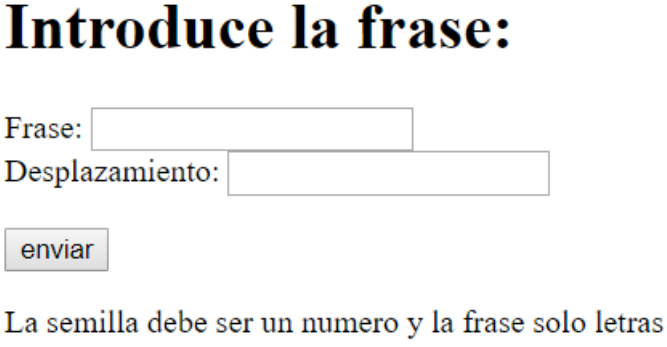
2.1. Manual de usuario.

Lo que el usuario ve cuando abre la pagina son dos campos que tiene que rellenar con lo que el quiera, en el primero se introducirá la frase que el quiera y en el segundo el numero para poder hacer la codificación.

Cuando el usuario introduzca la frase y la codificación le dará a enviar y le aparecerá la frase codificada y la frase descodificada.



En caso de que el usuario introduzca una semilla que no es valida el programa le dará un error. La semilla debe ser siempre un numero y la frase siempre letras, si introduces algo distinto el prgrama te lo dice.



2.2. Manual técnico.
Lo primero que nos encontramos es el formulario necesario para que el usuario pueda introducir por teclado tanto la frase como el desplazamiento deseado.

```html
<html>

<head>
    <title>"Codificacion"</title>
</head>

<body>
    <form method='POST' action='<?= $_SERVER['PHP_SELF'] ?>'>
        <h1>Introduce la frase: </h1>
        Frase: <input type='text' name='frase' id='frase'><br>
        Desplazamiento: <input type='text' name='desplazamineto' id='desplazamineto'><br>
        <p></p>
        <input type='submit' value="enviar" name="Enviar" id="Enviar">

    </form>
```

Lo siguiente que tenemos es el include de las funciones (que las explicaremos cuando salgan), el isset para comprobar que exista la frase enviada por el usuario y asi poder empezar nuestra ejecución del programa, recogemos la frase y la pasamos a minúsculas para que sea mas facil trabajar con ella, recogemos el desplazamiento y creamos un array letras que es el abecedario.

```php
<?php
include "funciones.php";
if (isset($_POST['frase'])) { //comprobamos que existen las variables

    $frase = strtolower($_POST['frase']);
    $des = $_POST['desplazamineto'];
    $letra = array('a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'K', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z');
```

Lo siguiente que nos encontramos es la comprobación de la variable frase para que solo sean letras y la comprobación de la variable desplazamiento para que solo sean numero, e imprimimos la frase sin codificar y donde va a ir la frase codificada para mayor entendimiento del usuario.

```php
if (solo_letras($frase) && es_numero($des)) {
    echo "Frase sin codificar: " . $frase;
    echo "<br>";
    echo "Frase condificada: ";
```

Para hacer dichas comprobaciones hemos usado dos funciones, la primera de ellas es solo_letras en la que tenemos declarada una variable de todas las letras tanto minúscula como mayúscula, solo nos harían falta las minúsculas pero así podemos reutilizar el código, recorremos con un for la cadena pasada, con el strpos conseguimos que nos diga la primera vez que aparezca un carácter igual al permitido, con substr dándole como parámetro la frase pasada y el contador del for vamos recorriendo la frase y comparándola con la permitida, si en algún momento hay algún carácter que no sea permitido nos devolverá false, pero si son todos permitido devolverá true.

```php
function solo_letras($cadena){
    $permitidos = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ ";
    for ($i=0; $i<strlen($cadena); $i++){
        if (strpos($permitidos, substr($cadena,$i,1))===false){
            //no es válido;
            return false;
        }
    }
    //si estoy aqui es que todos los caracteres son validos
    return true;
}
```

La segunda que tenemos es la función es_numero que comprueba que el parámetro pasado sean solo números, en este caso sencillamente lo comprobamos con is_numeric y devuelve true si son todos números y si no false.

```php
function es_numero($cadena){
    if (is_numeric($cadena)) {
        return true;
    }
    else {
       return false;
    }
}
```

Empezamos a codificar la frase, primeramente tenemos un for que recorre la frase introducida por el usuario, declaramos una variable carácter y en ella almacenamos cada letra de la frase, a continuación recorremos con otro for todo el array de letras que se trata del abecedario, comprobamos con un if cuando la letra de la frase coincide con la letra del abecedario, cuando coincide declaramos una variable llamada posición para hacer los cálculos que en este caso son sumas el valor de la letra del array con el numero de la semilla, hacemos el resto de esta operación por si la suma es mas grande que el array de la vuelta y se correctamente, declaramos una nueva variable enigma que va a ser la frase codificada, en ella guardamos la posición del array que nos a dado la operación, que es el desplazamiento que a elegido el usuario, por ultimo imprimimos enigma que es el resultado.

```php
for ($i = 0; $i < strlen($frase); $i++) {
    $caracter = substr($frase, $i, 1);
    for ($j = 0; $j < count($letra); $j++) {
        if ($letra[$j] == $caracter) {
            $posicion = ($j + $des) % count($letra);
            $enigma[$i] = $letra[$posicion];
            echo $enigma[$i];
        }
```

Lo siguiente que tenemos es la condición del espacio en blanco, si el for se encuentra con un espacio en blanco en el mensaje codificado tendremos un espacio en blanco y lo imprimirá.

```php
if ($caracter == " ") {
    $enigma[$i] = " ";
    echo " ";
}
```

Ahora nos encontramos con la decodificación, en ella lo primero que nos encontramos es un for que recorre la frase codificada en este caso enigma, declaramos una variable que almacenamos una por una las letras para ir comprobándolas, a continuación tenemos un for que recorre el array de letras, seguidamente una condición que cuando el carácter coincida con la letra del array, el programa comprueba que la resta entre la posicion de la letra en el array menos el desplazamiento es menos que cero, si es asi al desplazamiento le hacemos el residuo con el total del array para que nunca sea negativo y despues a en la posicion averiaguamos el total de letras del array y lo restamos a la operación de restar el desplazamiento con la posicion del array todo essto despues le hacemos el residuo para que nunca de negativo o se pase del total del array y siempre se quede el resultado dentro y por ultimo imprimimos el resultado. Si por el contrario no es menos que cero la resta de la posicion por el desplazamiento solo tenemos que

hacer el residup del desplazamiento para saber el desplazamiento y solo lo tendriamos que restar a la posicion del array y lo imprimimos.

```
for ($i = 0; $i < count($enigma); $i++) {
    $caracter = $enigma[$i];
    for ($j = 0; $j < count($letra); $j++) {
        if ($letra[$j] == $caracter) {
            if (($j - $des) < 0) {
                $des = $des % count($letra);
                $posicion = (count($letra) - ($des - $j)) % count($letra);
                echo $letra[$posicion];
            } else {
                $des = $des % count($letra);
                echo $letra[$j - $des];
```

Si el progrma al recorrer el array detecta un epacio en blanco en la codificacion, simplemente imprime un espacion en blanco.

```
}
if ($enigma[$i] == " ") echo " ";
```

Por ultimo nos encontramos la el caso de que la frase no sean solo letras o la semilla contenga letras.

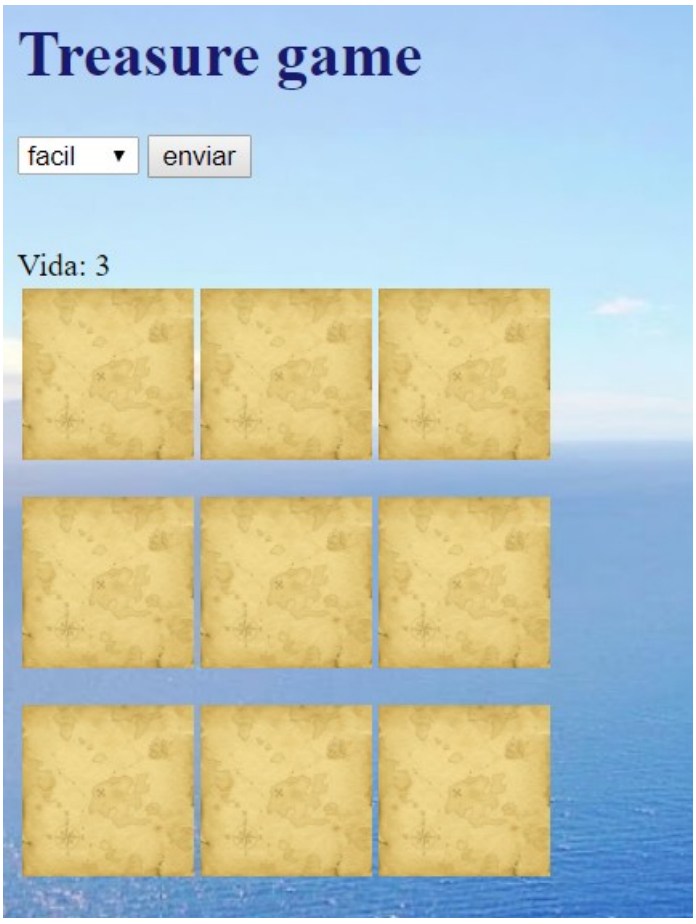
```
} else {
    echo "La semilla debe ser un numero y la frase solo letras";
}
```

3. Búsqueda del tesoro.

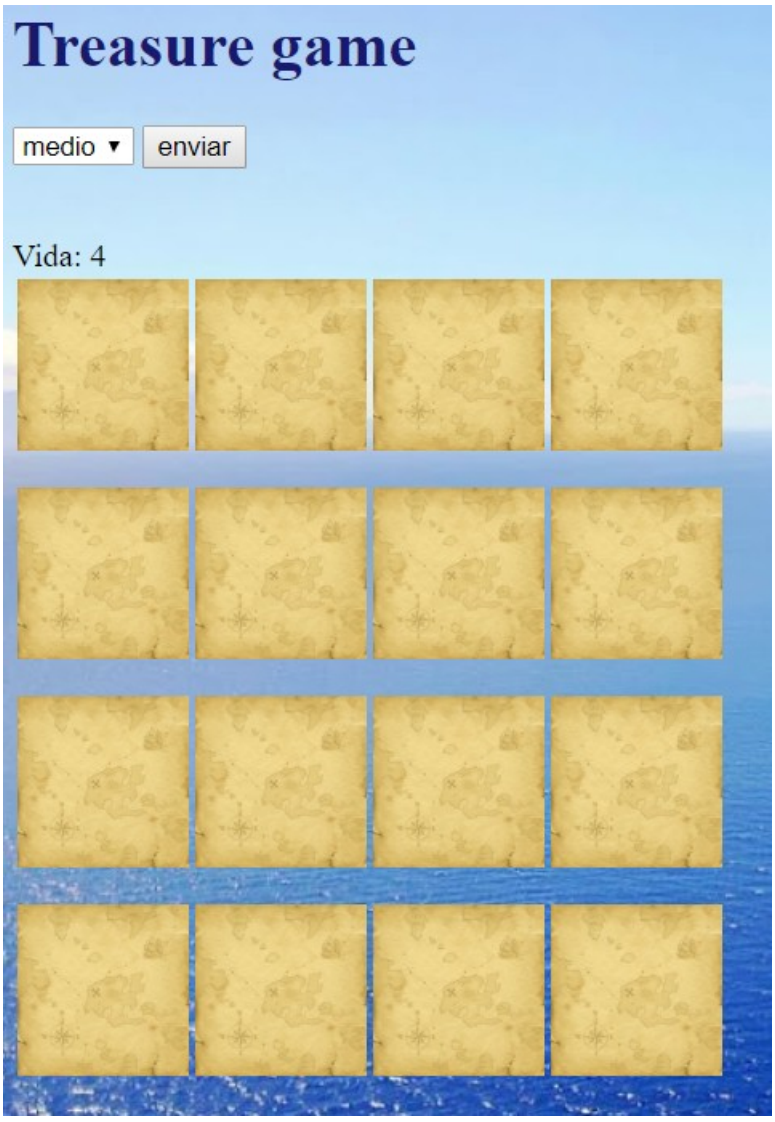
En este programa vamos a jugar a buscar el tesoro, el usuario deberá clicar en el cuadrado que el quiera y deberá descubrir donde esta el tesoro, pero deberá de llevar cuidado para no encontrar ninguna trampa, tendrá ayuda al encontrar alguna vida extra para seguir jugando.

3.1. Manual de usuario.
El juego consiste en buscar el tesoro,para ganar tendremos que encontrar el cofre del tesoro, también podremos encontrar corazones que no sumaran una vida extra y encontraremos fallos, que restaran una vida a tu total de vidas, solo podrás jugar cuando tengas vidas, dependiendo de la dificultad que elijamos tendremos mas vidas y aumentara el numero de cuadrados, en este juego tendremos tres dificultades fácil, medio y difícil.

En fácil tendremos 3 vidas ,el tablero sera de 3*3 y el numero de vidas que podremos encontrar dentro del juego sera 1.



En medio tendremos 4 vidas, el tablero sera de 4*4 y el numero de vidas que podremos encontrar sera de 1.

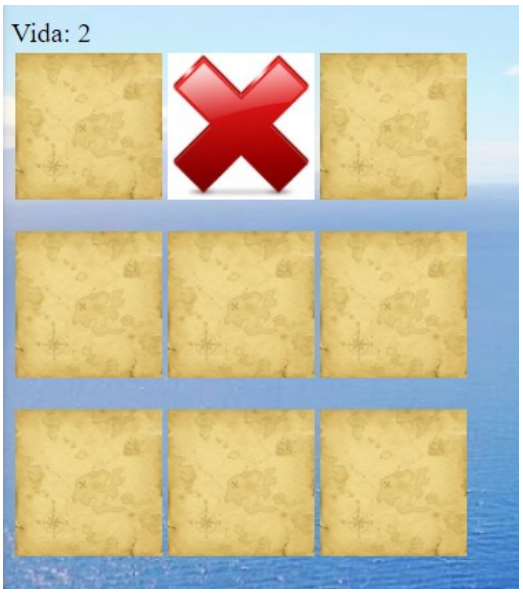


En difícil tendremos 5 vidas, el tablero sera de 6*6 y el numero de vidas que puedes encontrar sera 1.

Conforme el usuario clique en los diferentes rectángulos perderá vidas a no ser que encuentre una vida o el tesoro que de esa manera gana el juego.

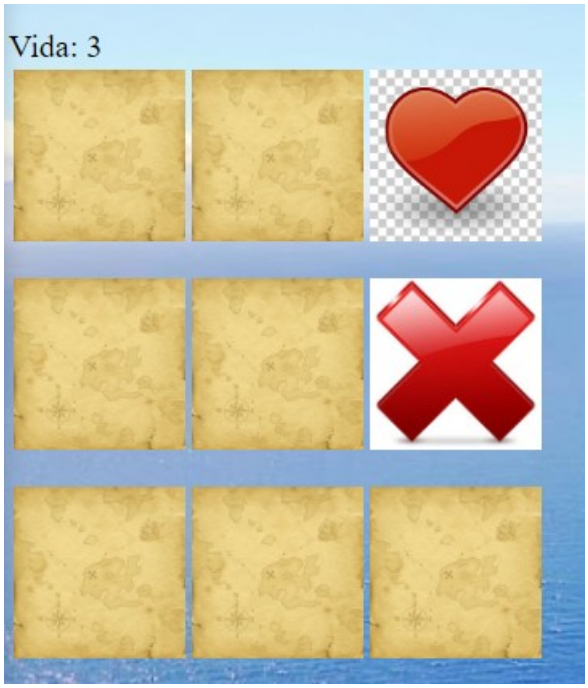
Vamos a ver los como funciona el juego:

En la primera tirada vemos que nos a salido una equis eso significa que no hemos encontrado el tesoro y tampoco una vida por lo tanto perdemos una vida.



En la segunda tirada vemos que nos a salido el tesoro por lo tanto hemos ganado la partida. Si quisiéramos volver a jugar solo tendríamos que darle al botón jugar.

Si en algún caso nos saliese la vida no nos resta movimiento y nos suma una vida.



3.2. Manual Técnico.

En primer lugar vemos un desplegable para que usuario elija la dificultad, nos enviamos lo que el usuario elija y con esa información determinamos el tamaño del tablero, si es fácil 3*3,si es medio 4*4 y si es difícil 6*6. Usaremos el value para saber que dificultad nos han pasado y saber diferenciarlas en cada caso.

```
<body background="fondo.jpg">
    <h1 style="color:midnightblue">Treasure game</h1>
    <form action="<?= $_SERVER['PHP_SELF'] ?>" method="POST">
        <select name="dificultad">
            <option value="3">facil</option>
            <option value="4">medio</option>
            <option value="6">dificil</option>
        </select>
        <input type='submit' value="enviar" name="Enviar" id="Enviar">
    </form>

</body>
```

A partir de aquí entramos en el código PHP, en primer lugar incluimos las funciones por si las necesitamos, declaramos un array vació y con un ternario preguntamos si existe la dificultad, si existe es que el usuario la a enviado y la usamos para determinar la dificultad y saber lo grande que es el tablero y el numero de vidas que tenemos, pero no si nos enviá ninguna por defecto ponemos la dificultad mas fácil, con ello conseguimos que siempre se cree un tablero y nunca se quede vacío el juego. Hemos decidido poner por defecto 3 para que el jugador pueda ganar fácil y así le guste el juego y siga jugando.

```
include_once "funciones.php";
$array = [];

$dificultad = isset($_POST['dificultad']) ? $_POST['dificultad'] : 3;
if ($dificultad == 3) {
    $vida = 3;
} else if ($dificultad == 4) {
    $vida = 4;
} else if ($dificultad == 6) {
    $vida = 5;
}
```

Comprobamos si la variable cadena existe, si es así es que el usuario a echo clic en alguno de los formulario, lo primero que hacemos es recoger las variables pasadas, las veremos mas adelante, y pasar la cadena a array para poder trabajar con ella.

```php
if (isset($_POST['cadena'])) {

    $cadena = $_POST['cadena'];
    $vida = $_POST['vida'];
    $fila = $_POST['fila'];
    $columna = $_POST['columna'];
    $fila_tesor = $_POST['fila_tesor'];
    $columna_tesoro = $_POST['columna_tesoro'];
    $fila_vida = $_POST['fila_vida'];
    $columna_vida = $_POST['columna_vida'];


    $array = cadena_a_array($cadena);
```

Lo siguiente que hacemos es comprobar donde a hecho clic el usuario para poder ver si a ganado, si tiene una vida mas o a fallado. En el primer if que tenemos comprobamos si es la vida, si es así mostraríamos la imagen de la vida y sumaríamos una vida.

```php
if ($fila == $fila_vida && $columna_vida == $columna) {
    $array[$fila][$columna] = "vida.png";
    $vida++;
```

La siguiente comprobación que hacemos es si ha ganado encontrando el tesoro, en ella comprobamos si a clicado en el tesoro y si es así mostramos el tesoro la vida y todos los errores del tablero. A continuación sacamos un botón para que el usuario pueda volver a jugar.
Para ellos lo recorreremos el array, si la fila y la columna coincide con la del tesoro lo mostrara, cuando coincida con la vida mostrara la imagen de la vida y todo lo demás mostrara la imagen del error. Para mostrar el botón, simplemente hacemos un formulario con un botón y lo redirigimos a la misma pagina, de ese modo conseguimos que reinicie el juego.

```php
} else if ($fila == $fila_tesor && $columna_tesoro == $columna) {
    $vida == 0;
    echo "Winner";

    for ($i = 0; $i < $dificultad; $i++) {
        for ($j = 0; $j < $dificultad; $j++) {
            $array[$fila_tesor][$columna_tesoro] = "tesoro.png";
            $array[$fila_vida][$columna_vida] = "vida.png";
            $array[$i][$j] = "error.jpg";
        }
    }

    ?>
    <form action="<?= $_SERVER['PHP_SELF'] ?>" method="POST">
        ¿Quieres volver a jugar? <input type='submit' value="jugar" name="jugar" id="jugar2">
    </form>
```

La siguiente que tenemos es si el usuario pierde todas las vidas, en esta comprobación vamos a poner las vidas igual a 1 para que el ultimo clic sea el perdedor y no cuente mal, si lo ponemos a cero el programa hará el ultimo clic y nos contara como un movimiento mas, cuando perdamos mostrara donde estaba el tesoro, las vidas y todos los errores de la partida y por ultimo te mostrara un botón para volver a jugar. Para ello usaremos el mismo metodo que antes si la fila y la columna coincide con la vida o el tesoro la mostraremos y si no sera un error, usaremos el mismo mismo botón para poder volver a jugar si queremos.

```php
} else if ($vida == 1) {
    $vida = 0;
    echo "Looser";
    for ($i = 0; $i < $dificultad; $i++) {
        for ($j = 0; $j < $dificultad; $j++) {
            $array[$fila_tesor][$columna_tesoro] = "tesoro.png";
            $array[$fila_vida][$columna_vida] = "vida.png";
            $array[$i][$j] = "error.jpg";
        }
    }

    ?>
<form action="<?= $_SERVER['PHP_SELF'] ?>" method="POST">
    ¿Quieres volver a jugar? <input type='submit' value="jugar" name="jugar" id="jugar">
</form>
```

Por ultimo comprobamos que sea un fallo y aquí simplemente restamos una vida. Para ellos simplemente comprobamos en que fila y en que columna a clicado el usuario y si es algún cuadrado que aun no tengamos clicado entrara y comprobara que es un error, si hubiese sido la vida o el tesoro habríamos parado antes la ejecución porque lo compramos antes.

```
} else if ($array[$fila][$columna] = "playa.jpg") {
    $array[$fila][$columna] = "error.jpg";
    $vida--;
}
```

Como hemos dicho antes todo esto lo haríamos si era la segunda vez o mas, pero la primera vez tenemos que rellenar el array con imágenes de playa. Para ello usamos dos fors para recorrer el array y lo llenamos de imágenes de playa, la longitud del array nos la dará la dificultad elegida por el usuario.

```
else { //primera vez

  for ($i = 0; $i < $dificultad; $i++) {
      for ($j = 0; $j < $dificultad; $j++) {
          $array[$i][$j] = "playa.jpg";
      }
  }
```

A continuación introducimos la posición del tesoro y de la vida totalmente aleatorio para que el juego sea justo y no sea repetitivo, para ello a la fila del tesoro y la columna tanto de la vida como de la columna de la de un numero aleatorio entre 0 y la dificultad menos dos para que nunca se salga del array y siempre estén tanto el tesoro como el corazón.

```
$fila_tesor = random_int(0, $dificultad - 2);
$columna_tesoro = random_int(0, $dificultad - 2);
$fila_vida = random_int(0, $dificultad - 2);
$columna_vida = random_int(0, $dificultad - 2);
```

Comprobaremos que el tesoro y el corazón nunca caigan en la misma posición, en ese caso le restaremos una posición a la fila y a la columna del corazón para que cambie su posición, si no hiciésemos este podría caer en la misma posición el tesoro y el corazón.

```php
while ($fila_tesor == $fila_vida && $columna_tesoro == $columna_vida) {
    $fila_vida = random_int(0, $dificultad - 1);
    $columna_vida = random_int(0, $dificultad - 1);
}
```

Por ultimo hacemos lo mas importante del código, introducimos en cada posición del array toda la información del formulario y de esta manera poder comprobar que casilla esta clicando el usuario y poder hacer las comprobaciones, cada vez que el usuario clica nos enviá toda esta información y es la que recogemos al principio de la pagina para poder trabajar con ella. Para evitar que cuando el usuario haya clicado ya nos enviemos toda esta información que ya no nos hará falta y que el usuario pueda seguir clicando comprobamos que la imagen sea igual a la inicial que en esta caso es playa para saber que no a clicado. Todas las imágenes están dentro de una tabla para que quede mucho mas bonito y ordenado.

```php
for ($j = 0; $j < $dificultad; $j++) {
    if ($array[$i][$j] == 'playa.jpg') {
        ?>
    <td>
        <form action="<?= $_SERVER['PHP_SELF'] ?>" method="POST">
            <input type="image" name="<?php echo "f" . $i . "c" . $j ?>" id="<?php echo "f" . $i . "c" . $j ?>"
            src="<?php echo $array[$i][$j] ?>" alt="imagen" height="85" width="85">
            <input type="hidden" name="fila" id="fila" value="<?php echo $i ?>">
            <input type="hidden" name="columna" id="columna" value="<?php echo $j ?>">
            <input type="hidden" name="cadena" id="cadena" value="<?php echo $cadena ?>">
            <input type="hidden" name="vida" id="vida" value="<?php echo $vida ?>">
            <input type="hidden" name="winner" id="winner" value="<?php echo $winner ?>">

            <input type="hidden" name="dificultad" id="dificultad" value="<?php echo $dificultad ?>">
            <input type="hidden" name="fila_tesor" id="fila_tesor" value="<?php echo $fila_tesor ?>">
            <input type="hidden" name="columna_tesoro" id="columna_tesoro" value="<?php echo $columna_tesoro ?>">
            <input type="hidden" name="fila_vida" id="fila_vida" value="<?php echo $fila_vida ?>">
            <input type="hidden" name="perdido" id="perdido" value="<?php echo $perdido ?>">
            <input type="hidden" name="columna_vida" id="columna_vida" value="<?php echo $columna_vida ?>">
        </form>
```

Si el usuario ya ha hecho un clic la imagen que tendremos no sera la de la playa por lo tanto ya no tendremos que enviar toda esta información de los formulario porque no tendremos que comprobarla para hacer comprobaciones, por lo tanto imprimimos la imagen pero en formato imagen sin ningún formulario, de esta manera nos evitamos que el usuario pueda seguir clicando y pueda llegar a causar algún error a la pagina.

```
} else {
    echo "<td><img src='" . $array[$i][$j] . "' alt='imagen' height='85' width='85'></td>";
}
```