

EJERCICIOS UNIDAD 3

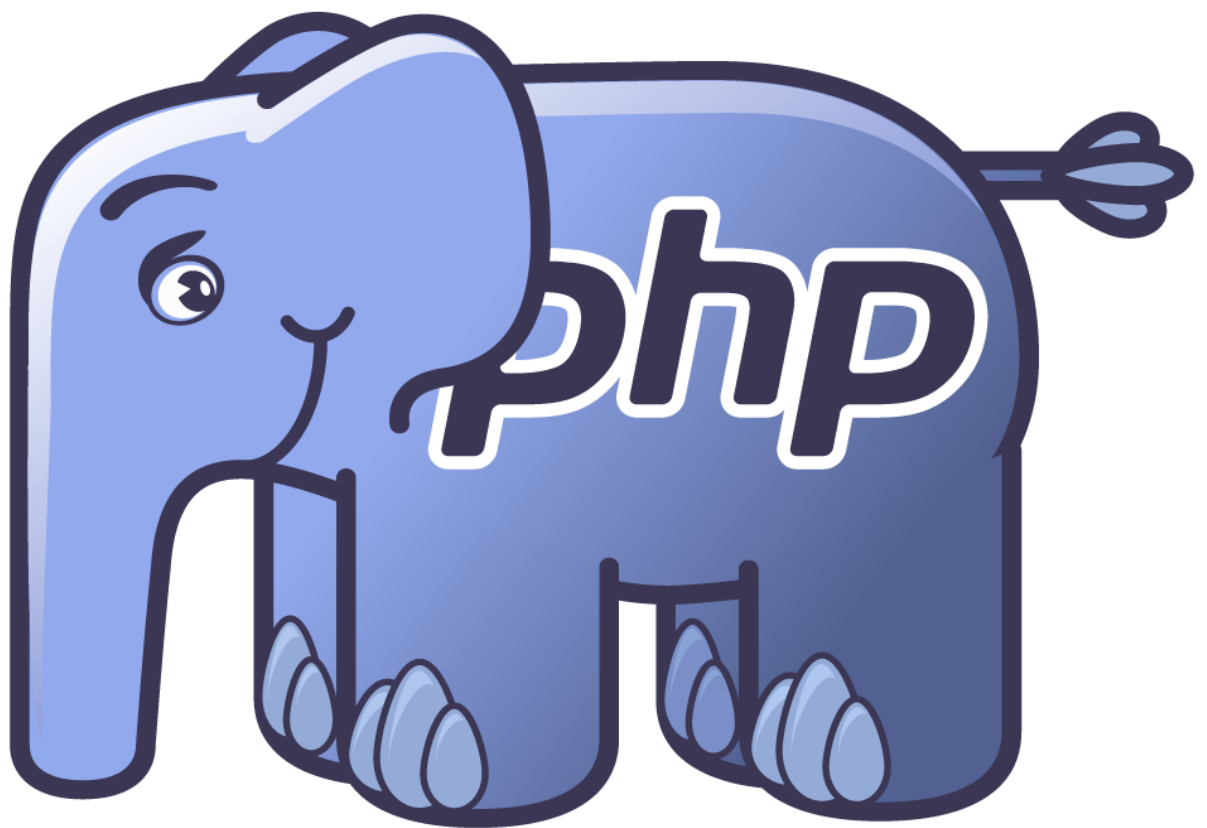


Tabla de contenido

1. Cifrado	3
1.1. Manual de usuario	3
1.2. Manual técnico.....	4
1.2.1. Funciones y variables	4
1.2.2. Cuerpo general del programa	5
2. Comprobar fecha	6
2.1. Manual de usuario	6
2.2. Manual técnico.....	7
2.2.1. Funciones y variables	7
2.2.2. Cuerpo general del programa	7
3. Creación de pirámide	8
3.1. Manual de usuario	8
3.2. Manual técnico.....	9
3.2.1. Funciones y variables	9
3.2.2. Cuerpo general del programa	9
4. Gimnasio	10
4.1. Manual de usuario	10
4.2. Manual técnico.....	11
4.2.1. Cuerpo general del programa	11
5. Vendedores y productos	12
5.1. Manual de usuario	12
5.2. Manual técnico.....	15
5.2.1. Funciones y variables	15
5.2.2. Cuerpo general del programa	16
6. Trabajadores y secciones	19
6.1. Manual de usuario	19
6.2. Manual técnico.....	22
6.2.1. Funciones y variables	22
6.2.2. Cuerpo general del programa	23
7. Reserva de avión	27
7.1. Manual de usuario	27
7.2. Manual técnico.....	28

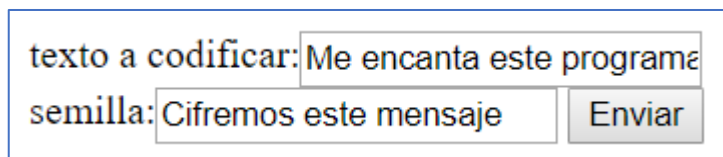
7.2.1.	Funciones y variables	28
7.2.2.	Cuerpo general del programa	28
8.	Búsqueda del tesoro	29
8.1.	Manual de usuario	29
8.2.	Manual técnico.....	31
8.2.1.	Funciones y variables	31
8.2.2.	Cuerpo general del programa	32

1. Cifrado

1.1. Manual de usuario

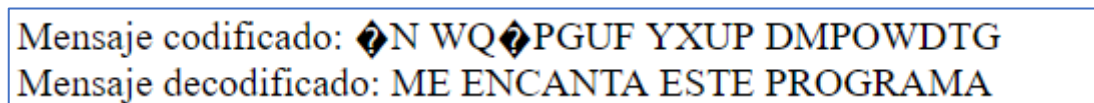
Al iniciar el programa le aparecerá un formulario, rellene los datos, se acepta cualquier tipo de dato, sin embargo, cualquier carácter introducido que no este en el alfabeto español será ignorado y se establecerá una codificación que variara según el apartado del formulario que se haya introducido el carácter extraño.

En caso de introducir una semilla vacia se tomara como semilla "ABCD".



The screenshot shows a web form with two input fields and a button. The first field is labeled 'texto a codificar:' and contains the text 'Me encanta este programa'. The second field is labeled 'semilla:' and contains the text 'Cifremos este mensaje'. To the right of the second field is a button labeled 'Enviar'.

Envíe el contenido y verá el resultado.



The screenshot shows the results of the encoding and decoding process. It displays two lines of text: 'Mensaje codificado: ?N WQ?PGUF YXUP DMPOWDTG' and 'Mensaje decodificado: ME ENCANTA ESTE PROGRAMA'. The question marks represent characters that were not in the Spanish alphabet and were therefore not encoded or decoded.

Si en el contenido de los mensajes visualiza un carácter "?" significa que el carácter que ha introducido no esta en el alfabeto español (a-Z) por lo que no se ha codificado ni decodificado

1.2. Manual técnico

1.2.1. Funciones y variables

La variable principal seria \$abecedario, la cual contiene el abecedario español

```
$abecedario = "ABCDEFGHIJKLMNÑOPQRSTUVWXYZ";
```

El programa principalmente está compuesto por dos funciones a las cuales se les pasa un texto y una semilla, dichas funciones son las siguientes:

Codificar: Partiendo del mensaje original, retorna el mensaje codificado

Decodificar: Partiendo del mensaje codificado, retorna el mensaje original.

Ambas funciones siguen el mismo procedimiento:

Pasan a mayúsculas el texto para evitar cualquier error sintáctico ya que el abecedario se ha introducido en mayúsculas.

Buscan las posiciones del texto y de la semilla en el abecedario, operan estas posiciones y añaden la posición del resultado (en \$abecedario) a la cadena que contiene el texto decodificado.

Si la posición es false (no retorna numero) quiere decir que no existe en \$abecedario, si se da ese caso, se imprimirá el carácter “?”

La principal diferencia entre una y otra es que codificar al operar las posiciones suma y decodificar resta.

También, codificar si llega a la letra z (ultima posición) realiza una resta para “dar la vuelta” y llegar a la posición 0.

1.2.2. Cuerpo general del programa

Solo se podrá ingresar a este apartado una vez enviado el formulario, el texto puede estar vacío, no lo he considerado como un error, sin embargo, en caso de introducir una semilla vacía se tomará como semilla "ABCD".

Esto lo he realizado así ya que no se estableció ningún criterio para ello y resulta mas cómodo tanto para el programador como para el usuario ya que no será necesario que el usuario vea un mensaje de error y tenga que reintroducir una semilla.

Una vez pasado ese control se llamará a las dos funciones y mostrará el resultado.

2. Comprobar fecha

2.1. Manual de usuario

Al iniciar el programa le aparecerá un formulario, rellénelo correctamente en el siguiente formato: dd/mm/aaaa

No introduzca ningún tipo de carácter no numérico

Si introduce 2 cifras en el año, se interpretara que usted se refiere a 2017, si quiere referirse al milenio anterior, tendrá que escribirlo completamente

Validez de los datos:

- La fecha contendrá únicamente números y el delimitador /
- El día estará entre 1 y 31 exceptuando el 30 y 31 de febrero.
- El mes estará entre 1 y 12.
- El año será superior a 1900 e inferior a 2018.

A screenshot of a web form. On the left, the text 'Fecha:' is followed by a text input field containing '01/1/1990'. The input field has a yellow highlight. To the right of the input field is a button labeled 'Enviar'.

Envíe la fecha y se le mostrara si esta es correcta

Fecha introducida: 01/1/1990

Día obtenido: 01

Mes obtenido: 01

Año obtenido: 1990

Resultado: la fecha es correcta

2.2. Manual técnico

Este programa se ha creado directamente en el cuerpo del programa dado que no he visto necesario crear funciones para estas operaciones.

2.2.1. Funciones y variables

Los nombres de las variables son totalmente comprensibles y su explicación es innecesaria, no obstante comentare las principales variables las cuales son \$día, \$mes y \$año que equivalen al día, mes y año introducidos en la fecha

2.2.2. Cuerpo general del programa

En un inicio, se mostrara el formulario para rellenar la fecha si no se ha recibido una fecha.

Se dividirá la fecha en partes usando de delimitador / y se harán controles para garantizar que si el usuario no ha introducido una fecha con el formato dd/mm/aaaa

Estos controles se validaran haciendo substrings de lo que se ha introducido y dependiendo de lo que se recoja se realizaran unas acciones u otras,

Y ha introducido por ejemplo, 2/2/17 esta sea convertida en 02/02/2017.

Si el usuario, en este mismo caso quería introducir 1917 es imposible que yo lo sepa, he puesto que se introduzca 2017 dado que es el milenio actual.

Cumpliendo todo esto se entra en la recta final del programa, la cual comprobara la validez de los datos, estos son validos siempre que:

- La fecha contenga únicamente números y el delimitador /
- El día este entre 1 y 31 exceptuando el 31 y 30 de febrero.
- El mes este entre 1 y 12
- El año sea superior a 1900 e inferior a 2018

3. Creación de pirámide

3.1. Manual de usuario

Introduzca un numero en el formulario, este número representara el número de filas que se muestra en la pirámide.

El numero debe ser superior a dos.

Numero:	<input type="text" value="5"/>	<input type="button" value="Enviar"/>
---------	--------------------------------	---------------------------------------

```
543212345
 4321234
  32123
   212
    1
      1
     212
    32123
   4321234
  543212345
```

3.2. Manual técnico

Opcionalmente se ha añadido una pirámide inversa.

3.2.1. Funciones y variables

Se han creado las funciones:

Esnumero: que comprueba que el parámetro introducido sea numérico.

Esmayora2: que comprueba que esnumero es correcto y además el resultado es mayor a 2.

3.2.2. Cuerpo general del programa

Si todos los datos son correctos, y por correctos nos referimos únicamente a que sea un numero mayor a 2 se recorren dos for, uno representando filas y otro columnas y se van imprimiendo los datos.

4. Gimnasio

4.1. Manual de usuario

Rellene el formulario con el tiempo de entrenamiento y pulse enviar.

Puede rellenarlo todas las veces que quiera, cada vez que envíe el formulario se interpretara como una nueva sesión de entrenamiento.

El tiempo máximo de entrenamiento por sesión son 60 minutos por lo cual, si usted introduce 200 minutos solamente se tendrán en cuenta y se almacenaran los primeros 60, si quiere introducir mas de 60 minutos tendrá que hacerlo en diferentes sesiones aunque las lleve a cabo el mismo día, es decir, si quiere introducir 100 minutos tendrá que introducir una sesión de 60 y otra de 40, o dos de 50 minutos, etc.

Si introduce una cantidad de tiempo negativa se entenderá que ha cometido un error y esta se pasara a positivo.

Se entenderán los datos por correctos siempre y cuando sean únicamente números.

Introduzca 0 cuando quiera ver los resultados.

Tiempo: 21 Calorias: 72
Tiempo: 4 Calorias: 11
Tiempo: 60 Calorias: 324
Tiempo: 12 Calorias: 36
Tiempo total: 97
Calorias totales: 443

4.2. Manual técnico

4.2.1. Cuerpo general del programa

Estando enviados y siendo correctos los datos se entra al cuerpo del programa.

Se entenderán los datos por correctos siempre y cuando sean números.

- En caso de introducir un número y después letras se cogerá únicamente el número
- En caso de introducir letras y después un número no se cogerá nada.
- En caso de que el usuario introduzca un número negativo como tiempo de entrenamiento este lo convertirá a positivo

El programa efectuará en primera instancia la suma del primer minuto y posteriormente las de los siguientes minutos por medio de un while.

Se ha creado una variable a modo de contador, este ciclo se ejecutará hasta que ese contador sea el mismo que la cantidad de minutos introducida, esto, a mi punto de vista y manera de programar facilita el funcionamiento del programa al no tener que estar restando.

Por cada periodo de tiempo el while sumará un minuto y la cantidad de calorías indicadas.

Esto, se reenviará por medio de un get implícito a la página de inicio, este get contendrá todos los datos introducidos en formato array convertido a cadena.

También se enviarán las sumas totales de calorías y de tiempo para mostrar el resultado global al introducir 0

5. Vendedores y productos

5.1. Manual de usuario

Inicialmente se le mostraran sus datos actuales, en la parte superior dispone de un menú donde podrá hacer la gestión que quiera.

<div> Inicio Ver total producto Ver total vendedor Modificar ventas Ver Ingresos totales </div>										
Actualmente nuestros datos son los siguientes:										
Vendedor	Producto 1	Producto 2	Producto 3	Producto 4	Producto 5	Producto 6	Producto 7	Producto 8	Producto 9	Producto 10
1	4	9	9	0	0	7	1	5	9	9
2	5	9	3	8	7	8	4	0	1	9
3	3	3	4	4	5	1	8	6	3	4
4	8	6	1	4	2	4	5	3	6	7
5	4	6	8	6	7	5	3	6	0	1
6	1	2	0	4	3	9	4	1	4	1
7	6	5	7	0	1	7	0	8	0	3
8	9	4	7	8	3	3	6	2	5	1
9	6	9	4	4	9	4	5	2	0	3
10	1	3	3	3	1	5	8	3	8	1
11	2	7	1	7	0	1	9	9	8	5
12	2	9	3	3	1	2	6	0	8	7
13	4	4	7	3	5	7	6	7	4	3
14	4	5	2	8	4	4	5	8	7	4
15	1	2	0	9	2	3	2	1	8	1
16	5	9	2	0	5	0	6	2	4	2
17	6	3	6	9	1	2	4	3	1	3
18	9	0	9	0	6	3	1	3	2	0

- Ver total de producto y vendedor

Introduzca un numero de producto/vendedor y clic en mostrar

Inicio
Ver total producto
Ver total vendedor
Modificar ventas
Ver Ingresos totales

Producto 1
▼

Mostrar

Producto 1
Producto 2
Producto 3
Producto 4
Producto 5
Producto 6
Producto 7
Producto 8
Producto 9
Producto 10

Finalmente se le mostrara el resultado

Inicio
Ver total producto
Ver total vendedor
Modificar ventas
Ver Ingresos totales

Vendedor	Ventas
1	4
2	5
3	3
4	8
5	4
6	1
7	6
8	9
9	6
10	1
11	2
12	2
13	4
14	4
15	1
16	5
17	6
18	9

El resultado de todas las ventas del producto es: 80

- Modificar ventas

Introduzca vendedor producto y el nuevo valor

Inicio Ver total producto Ver total vendedor Modificar ventas Ver Ingresos totales

Vendedor 1 ▼ Producto 1 ▼ Introduzca el nuevo valor Modificar

Clic en modificar y le aparecerá una pantalla donde se le mostrara el dato que va a cambiar y una confirmación.

Inicio Ver total producto Ver total vendedor Modificar ventas Ver Ingresos totales

Estas apunto de cambiar el valor 8 por 2

Confirmar

Confírmelo y se le volverán a mostrar los datos.

Si introduce un numero negativo, el nuevo valor será 0.

Si no introduce ningún numero, el nuevo valor será 0.

Puede introducir letras debido a si quiere introducir unidades de medida, sin embargo, si realiza esto, la comprensión de los datos se podrían ver afectados, téngalo en cuenta

- Ver ingresos totales

Clic y se le mostraran los ingresos totales

Vendedor	Ventas
1	53
2	54
3	41
4	46
5	46
6	29
7	37
8	48
9	46
10	36
11	49
12	41
13	50
14	51
15	29
16	35
17	38
18	33

Vendedor	Ventas
1	80
2	95
3	76
4	80
5	62
6	75
7	83
8	69
9	78
10	64

5.2. Manual técnico

5.2.1. Funciones y variables

La variable principal es \$negocio, esta equivale a un array bidimensional en el que están almacenados los vendedores y los productos

Podemos encontrar esta misma variable como negocioencadena, esta simplemente es \$negocio convertida a cadena de texto para realizar el envío de formulario.

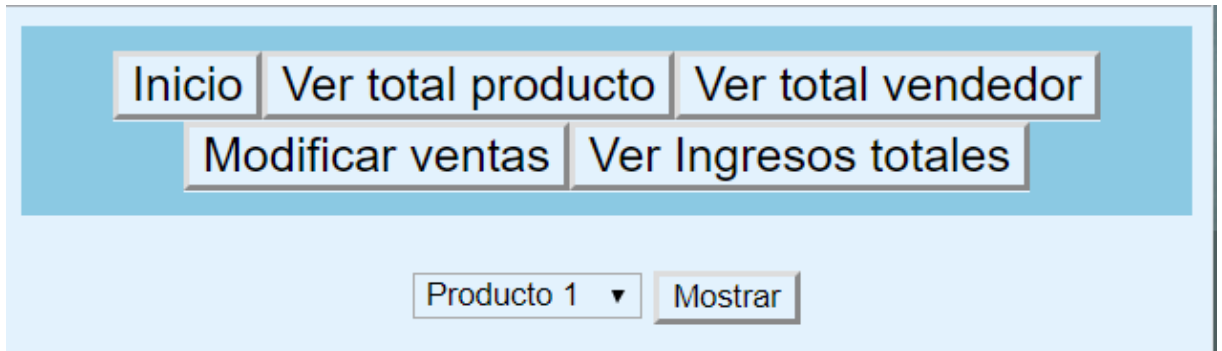
Para este programa he creado dos ficheros de funciones.

- Funciones.php → Contiene las funciones básicas que suelo poner en cada programa
- Funcionesespecificas.php → Contiene una unica función que ha sido la que he usado en este ejercicio únicamente.
 - Esta función es un menú realizado con formularios
 - ¿Porque con formularios?
 - Me ha parecido lógico y sencillo, además es la única posibilidad que se me ha ocurrido
 - ¿Por qué esta en una función y no directamente?
 - Porque he decidido que esto seria mas cómodo ya que por un lado no tenia que estar copiando y pegando código en todas las paginas.
 - Porque no ocupa espacio en la pagina principal.
 - Porque solamente quiero llamar a esta función cuando exista un negocio ya que este menú son formularios y en dichos formularios envío los datos del negocio y primero necesito recogerlos.

5.2.2. Cuerpo general del programa

Toda la pagina esta realizada bajo un modelo de cajas flexible.

La forma de hacerlo la he aprendido a través de este [video](#).



Inicio: Muestra el array que tenemos con los datos, inicialmente si estos datos no existen se generan de forma aleatoria con números del 0 al 9. Esto lo he hecho para dar una sensación de autenticidad a la pagina dado que queda un poco cutre mostrar todos los datos en 0.

Realmente todas las paginas usan el mismo sistema, por lo que vamos a centrarnos principalmente en la primera.

- Ver total producto:
 - si no se ha recibido la cadena, se muestra por pantalla que ha ocurrido un error
 - Si se ha recibido la cadena con los datos se convierte a array.
 - Se pone el menú
 - Se comprueba si ha llegado un producto para ver su total
 - Si ha llegado se muestran los datos
 - \$suma almacena la suma cada vez que el for muestra un resultado y finalmente muestra el resultado de la suma
 - Si no ha llegado se muestra un formulario para pedir el producto, en este formulario enviaremos también el array completo
- Ver total vendedor:
 - Es lo mismo que ver total producto cambiando que en este caso en vez de 10 productos serán 18 vendedores y no se comprueba si ha llegado un producto, sino un vendedor.
- Modificar ventas:
 - Parece diferente a simple vista, pero si nos fijamos bien, es juntar ver total producto con ver total vendedor.
 - Si ha llegado producto y vendedor se pondrá un nuevo valor en \$Negocio[\$vendedor][\$producto]
 - Sino, se pedirá vendedor y producto
 - Se volverá a enviar el array en cadena
 - Si se introduce un valor nulo o inferior a 0, el valor será sustituido por 0, esto es así dado que no se ha especificado que resultado debe tener y me ha parecido correcto

- Ver Ingresos totales:
 - Sigue el mismo sistema que los anteriores pero con una pequeña diferencia, nos olvidamos de las tablas, columnas y filas. Y simplemente hacemos lo siguiente:

```
$suma = 0;
for ($v = 0; $v < 18; $v++) {
    for ($p = 0; $p < 10; $p++) {
        $suma += $negocio[$v][$p];
    }
    echo "Vendedor" . ($v+1) . " = " . $suma . " ventas<br/>";
    $suma = 0;
}
```

- Se recorrerá un for para todos los vendedores, cuando acabe la fila de vendedor, se mostrara la suma y un se pondrá la variable suma a 0 para que el próximo vendedor parta de ese valor.
- En caso de productos será exactamente lo mismo pero cambiando de posición \$p y \$v así como sus valores máximos.

Se ha decidido utilizar en todos los métodos de introducción de datos excepto el valor a modificar para minimizar los errores de usuario dado que no se ha especificado ninguna forma de realizarlo.

6. Trabajadores y secciones

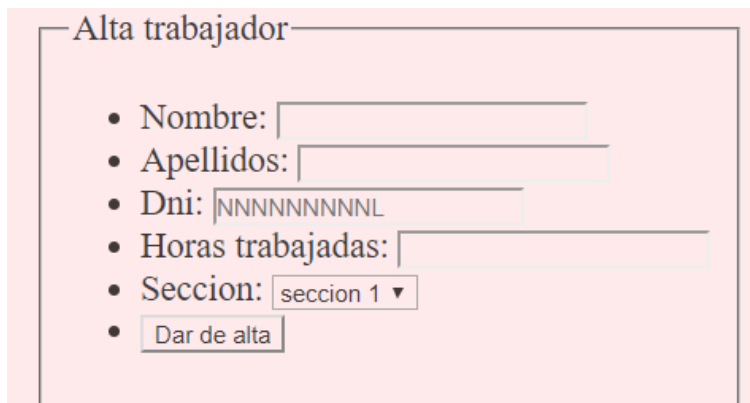
6.1. Manual de usuario

En inicio visualizaría los datos, la pagina no contendrá datos inicialmente

Inicio Alta Baja ver trabajador	
Actualmente nuestros trabajadores son los siguientes:	
<hr/>	
	SECCION 1
Sin trabajadores	
	SECCION 2
Sin trabajadores	

Dispone de 3 opciones adicionales a inicio.

- Alta, servirá para dar de alta a un trabajador, tenga en mente lo siguiente:
 - Es necesario asignar un nombre al trabajador.
 - Los apellidos no son necesarios
 - El DNI del trabajador debe ser valido
 - Las horas trabajadas no pueden exceder a 50
 - Usted puede introducir decimales en las horas, sin embargo, solamente se calcularan las horas enteras, si un trabajador hace 5 horas y media se mostraran, y si posteriormente hace otras 5 y media y se las modifica se mostraran un total de 11 horas

El formulario, titulado "Alta trabajador", está presentado sobre un fondo rosa claro y está rodeado por una línea negra. Incluye los siguientes campos: "Nombre:" con un campo de texto; "Apellidos:" con un campo de texto; "Dni:" con un campo de texto que muestra "NNNNNNNNNL"; "Horas trabajadas:" con un campo de texto; "Seccion:" con un menú desplegable que muestra "seccion 1" y un icono de flecha hacia abajo; y un botón rectangular etiquetado "Dar de alta".

Alta trabajador


- Nombre:
- Apellidos:
- Dni:
- Horas trabajadas:
- Seccion:
-

- Baja, Servirá para dar de baja a un trabajador de una única sección,
 - tenga en mente que el DNI de dicho trabajador debe existir y debe ser valido, así como trabajar en dicha sección

Baja trabajador

- Dni:
- Seccion:
-

- Ver trabajador, Servirá para visualizar Y/O editar a un trabajador de una única sección
 - tenga en mente que el DNI de dicho trabajador debe existir y debe ser valido
 - Para editarlo pulse en el lápiz

Nombre	Apellidos	Dni	Horas	Suelo bruto	Suelo neto	Impuestos	Editar
Daniel	Ortega Conesa	74445783G	13	78	70.1	7.9	

Modificar trabajador

No introduzcas datos y no se modificaran estos

Datos de el trabajador con dni: 74445783G

- Nombre:
- Apellidos:
- Horas trabajadas:
-

6.2. Manual técnico

6.2.1. Funciones y variables

Este apartado seguirá la misma estructura que el del ejercicio 5

La variable principal es \$empresa, esta equivale a un array tridimensional asociativo compuesto por [seccion][DNItrabajador][datos]

Podemos encontrar esta misma variable como empresaencadena, esta simplemente es \$empresa convertida a cadena de texto para realizar el envío de formulario.

Para este programa he creado dos ficheros de funciones.

- Funciones.php → Contiene las funciones básicas que suelo poner en cada programa
- Funcionesespecificas.php → Contiene funciones para este ejercicio únicamente.
 - Ponermenu: Esta función es un menú realizado con formularios
 - ¿Porque con formularios?
 - Me ha parecido lógico y sencillo, además es la única posibilidad que se me ha ocurrido
 - ¿Por qué esta en una función y no directamente?
 - Porque he decidido que esto seria mas cómodo ya que por un lado no tenia que estar copiando y pegando código en todas las paginas.
 - Porque no ocupa espacio en la pagina principal.
 - Porque solamente quiero llamar a esta función cuando existan secciones ya que este menú son formularios y en dichos formularios envío los datos de las secciones y primero necesito recogerlas.
 - Sueldobruto: Calcula el sueldo bruto de la misma forma que el ejercicio del gimnasio calcula las calorías, finalmente retorna dicha cantidad
 - impuestos: a partir del resultado de sueldobruto, aplica el mismo sistema para calcular los impuestos

6.2.2. Cuerpo general del programa

Toda la pagina esta realizada bajo un modelo de cajas flexible.

La forma de hacerlo la he aprendido a través de este [video](#).

Inicio: Muestra el array que tenemos con los datos, inicialmente estos datos no existen no se generaran y se mostrara que no existen datos.

Realmente todas las paginas siguen el mismo sistema, por lo que vamos a centrarnos principalmente en la primera y veremos por encima las demás.

- Alta:
 - si no se ha recibido la cadena, no permite la entrada
 - Si se ha recibido la cadena con los datos se convierte a array.
 - Se pone el menú
 - Se comprueba
 - Que no se haya enviado un formulario con datos
 - Que de haberse enviado, las horas no sean mas de 50, esta pagina y modificar serán las únicas que comprueben las horas.
 - Si alguna de las anteriores se cumple, se muestra un formulario

- De no mostrarse formulario
 - Se recogen los datos
 - Se ajusta su validez de modo que
 - De no introducir horas, estas serán 0.
 - De introducir horas negativas estas serán 0.
 - De introducir texto en las horas, estas serán 0.
 - Se comprueba que se ha recibido un nombre
 - De lo contrario mostrara error
 - Nunca se realiza una comprobación de sección puesto que los valores están predefinidos
 - Se pasa a añadir los datos siempre que el DNI sea válido y no esté introducido
 - De lo contrario mostrar error
- Finalmente se convierte el array a cadena y se envía

- Baja
 - Partimos de que el formulario ya ha sido enviado
 - Se recogen los datos y se ajustan de la misma forma
 - Se comprueba que el DNI exista, esto es al contrario ya que en alta queríamos añadir uno que no exista y en baja queremos borrar uno que exista.
 - Se borra de la siguiente forma:

```
unset($empresa[$seccion][$DNI]);
```
 - Se pide una confirmación y de darse, se enviará el array modificado mediante una cadena de texto por el formulario.
- Ver trabajador
 - Partimos de que el formulario ya ha sido enviado
 - Se recoge el DNI
 - Se comprueba que el DNI exista, esto es al contrario ya que en alta queríamos añadir uno que no exista y en ver queremos ver uno que exista.
 - Se muestran sus datos o bien que no existe
 - No se modifica en ningún momento ningún dato.

- Modificar trabajador.
 - Forma parte de ver trabajador, para acceder a esta sección hay que pulsar en el botón de editar (el lápiz) en la vista de un trabajador
 - Estando validados los datos previamente por ver trabajador, se envía DNI y sección y se accede a un menú en el que se pueden modificar estos datos
 - Se hace la siguiente comprobación con todos los datos:

```
• if ($_POST["nombre"] != "") {  
    $nombre = $_POST["nombre"];  
} else {  
    $nombre = $empresa[$seccion][$DNI]["nombre"];  
}
```

- De esta manera si algún dato no se rellena se dejara el que estaba.
- Al igual que en alta trabajador el nombre no puede estar vacío.
- Se hacen los mismos pasos que para dar de alta a un trabajador.

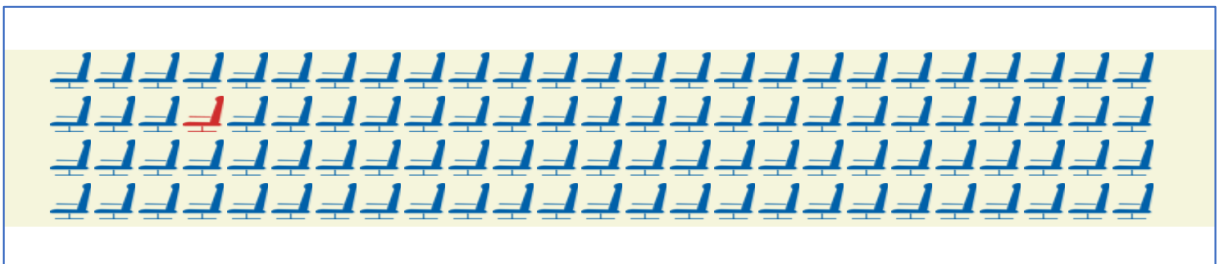
7. Reserva de avión

7.1. Manual de usuario

Al inicio del programa se mostrara la imagen de los asientos disponibles de su avión



Pulse sobre uno de ellos para reservarlo, este aparecerá en rojo indicando que ahora esta reservado



Vuelva a pulsar el asiento en rojo si desea cancelar la reserva

Los datos se guardan automáticamente, una vez termine de realizar su cancelación o reserva puede cerrar el programa sin que sea necesario aplicar cambios.

7.2. Manual técnico

7.2.1. Funciones y variables

La función principal de este programa es mostrar. A esta se le pasa un array, que en este caso sería el de los asientos disponibles que tiene el avión, y mediante el uso de 2 for anidados (filas y columnas) si el contenido de esa posición es una l (libre) muestra un asiento azul, de lo contrario, muestra uno rojo

7.2.2. Cuerpo general del programa

Si no existe el array, se genera uno con todos los datos en l, es decir libres todos los asientos.

En caso de que exista:

- Si el asiento que se ha pulsado tiene el valor l se le establece como valor r
- Si el asiento que se ha pulsado tiene el valor diferente a l se le establece como valor l

Finalmente se llama a la función mostrar

8. Búsqueda del tesoro

8.1. Manual de usuario

La pantalla de inicio será la siguiente:



En la zona superior izquierda tiene un desplegable, clic en el para seleccionar el tamaño del tablero y pulse comenzar para empezar una partida con el tamaño seleccionado.

También puede darle a comenzar para borrar la partida que está jugando y empezar otra de nuevo

El juego:

Usted debe ir haciendo clic en el tablero situado en la parte derecha de la pantalla, cada vez que usted realice esta acción se mostrara un resultado en esa misma casilla del tablero.

Durante la partida usted encontrara principalmente zonas vacías, pero también puede encontrar lagos, trampas y el tesoro.

- Lagos: Utilícelos para refrescarse puesto que el tesoro está escondido en una isla remota y en la época de los corsarios no abundaban los grifos de agua.
- Trampas: Evítelas a toda costa, los malvados piratas posicionan trampas a lo largo y ancho del mapa, si cae en una le atraparan, le apalearan y finalmente le robaran todo lo que tenga, incluido el mapa con el que pretende robarles el tesoro...

En parte se lo ha buscado, ¿no?

- Tesoro: Ha terminado la partida.
- Zona vacía: Simples zonas vacías ya exploradas

8.2. Manual técnico

8.2.1. Funciones y variables

En un inicio para evitar problemas definimos el tamaño del tablero a 4 si no se ha establecido otro.

- \$tamanyo: Tamaño del tablero
- Funcion aleatorio: simple función que retorna un numero aleatorio
- Función mostrar: función principal del programa.
 - En un inicio, si no están definidas las posiciones del tesoro, trampas, y lagos hace diversas llamadas a la función aleatorio para determinar las posiciones en un eje x y, estas posiciones son guardadas en un array sea cual sea el tamaño del tablero, a excepción del tesoro que siempre se va a guardar en una variable simple dado que únicamente habrá un tesoro sea cual sea el tamaño del tablero. Posteriormente estos datos se enviarán mediante un formulario que tiene por botón de enviar la propia casilla.
 - En el caso de las posiciones array se convertirán a cadena para enviarlas y en el caso de recibirlas se convertirá esa cadena a array.
 - En el caso de la variable simple, como su valor es únicamente numérico dado que ha sido generado por el programa y no puede cometer errores de usuario se envía de forma simple por el formulario.
 - Finalmente, recogidos o establecidos los datos de las posiciones, dependiendo del valor que tenga cada una se mostrara una imagen u otra.

8.2.2. Cuerpo general del programa

Este se divide en dos zonas, la zona cuyo nombre es menú, es un menú con el desplegable para seleccionar el tamaño del tablero y la leyenda del juego.

A la parte cuyo nombre es tablero es la que nos centraremos a continuación:

- Si el tablero NO ha sido enviado, se genera uno nuevo con todos los valores en ar (arena) y se pasa a la función mostrar la cual al ver que no existe ningún valor de tesoro, lago, etc generará unos y serán enviados continuamente para no perderlos.
- Si por el contrario, si ha sido enviado, se recogerán las posiciones x e y de la casilla ya que la única forma de que exista el tablero es que se haya pulsado alguna casilla del tablero ya que esta es la que actúa de formulario de envío.
 - En caso de que x e y coincida con alguna posición secreta se cambiara la posición[x][y] por el valor correspondiente, es decir, por na (nada), te(tesoro), la(lago) o tr(trampa) posteriormente será mostrado por la función mostrar.
 - En el caso de que sea trampa, se crearan unos nuevos valores x e y que recorrerán todo el tablero y en caso de que los valores sean diferentes a tr, estos se establecerán en ar(arena) borrando así toda la partida a excepción de las trampas, esto se ha hecho para evitar caer en la misma trampa.