Life To Fit

DAW2  Credito de síntesis

----

**Índice**

[Introducción 4](#_Toc483694002)

[Asignación de roles en el proyecto 5](#_Toc483694003)

[Planificación del proyecto 5](#_Toc483694004)

[Aplicación Freemium vs Premium 5](#_Toc483694005)

[Tecnología usada 5](#_Toc483694006)

[Organización del sitio web 7](#_Toc483694007)

[Página Índex 7](#_Toc483694008)

[Programación del registro 9](#_Toc483694009)

[Programación del cuestionario 10](#_Toc483694010)

[Programación del login 11](#_Toc483694011)

[Página Userperfil.php 14](#_Toc483694012)

[Apartado técnico de Userperfil.php 19](#_Toc483694013)

[Diagrama: 19](#_Toc483694014)

[Body\_history.php 22](#_Toc483694015)

[Weight.php 24](#_Toc483694016)

[Add.peso.proc.php 25](#_Toc483694017)

[Add\_medidas.proc.php 26](#_Toc483694018)

[Ch\_user\_data.proc.php 26](#_Toc483694019)

[Jpgraph 27](#_Toc483694020)

[Weight\_grafic.proc.php y history\_body\_grafic.proc.php 27](#_Toc483694021)

[Página de Rutina 29](#_Toc483694022)

[Programación Mostrar Rutinas Disponibles 30](#_Toc483694023)

[Programación Asignar Rutina 32](#_Toc483694024)

[Programación Detalle Rutina 35](#_Toc483694025)

[Programación Detalle Ejercicio 36](#_Toc483694026)

[Programación Empezar Rutina 37](#_Toc483694027)

[Programación Finalizar Rutina 41](#_Toc483694028)

[Página de Dietas 42](#_Toc483694029)

[Programación Selección de Dieta 42](#_Toc483694030)

[Programación Asignar Dieta 44](#_Toc483694031)

[Programación Descargar Dieta 45](#_Toc483694032)

[Programación Finalizar Dieta 46](#_Toc483694033)

[Conclusiones Tecnicas y opinión personal 47](#_Toc483694034)

[Índice de imágenes 47](#_Toc483694035)

[Webgrafía 47](#_Toc483694036)

# Introducción

LifeToFit es la aplicación web llevada a cabo como proyecto de síntesis del ciclo superior de Desarrollo de Aplicaciones Web del colegio Joan XXIII de Bellvitge.

La idea de este proyecto surge del interés por el mundo del Fitness y del ejercicio en general y nos percatamos de que mucha gente que va al gimnasio se lleva sus smartphones para escuchar música, WhatsApp etc., pero a la hora de hacer ejercicio seguían llevando la típica hoja con sus rutinas y ejercicios y vimos una necesidad por una aplicación que te llevara esas rutinas al día sin necesidad de recurrir al papel que en cualquier momento se puede deteriorar por agua, sudor, un accidente, etc.

Por otra parte, también nos dimos cuenta que estaría bien que la aplicación diera detalles de cómo realizar ese ejercicio y enseñara a los recién iniciados a realizar correctamente la actividad.

A todo esto, también vimos factible añadir un apartado de dietas y alimentación sana para que la gente tuviera en su smartphone tanto las rutinas como la dieta a seguir.

La idea principal del proyecto era que la aplicación tuviera un contenido gratuito, para que todo el mundo pudiera probarla, con unas pocas rutinas y dietas disponibles y un contenido premium en el que los usuarios por un pago desbloquean los servicios de un entrenador personal que le asigna una rutina específica y personalizada según las necesidades del usuario y también por otro pago adicional desbloquean el servicio de un nutricionista que le haría una dieta totalmente personalizada según las especificaciones del usuario.

Paralelamente a esta idea, también queríamos desarrollar esta app utilizando un framework de PHP llamado laravel con el que realizar una programación orientada a objetos. Lamentablemente, para aprender cómo utilizar este framework se necesita de mucho tiempo, así que decidimos solamente desarrollar la aplicación con PHP orientado a objetos utilizado el patrón de diseño de las 3 capas (modelo vista controlador).

Empezamos a investigar cómo se desarrollan aplicaciones utilizando el patrón MVC, que ficheros había que configurar y el controlador cómo funcionaba y llegamos a la conclusión que para desarrollar correctamente una aplicación con este patrón de diseño requería también mucho más tiempo del necesario que si empleáramos PHP como hasta ahora habíamos hecho. Por este motivo volvimos a cambiar de método de programación y al final nos quedamos con PHP normal.

A todo esto, ya habíamos empleado una semana en darnos cuenta que hay que dedicar mucho tiempo a algo que nunca se ha dado y solo teníamos 3 semanas más para realizar el proyecto. A pesar de este pequeño hándicap de una semana, hemos conseguido desarrollar la parte gratuita de nuestra aplicación.

## Asignación de roles en el proyecto

# Planificación del proyecto

Explicar cuál era nuestra primera idea tanto de la app como de cómo lo íbamos a hacer..

## Aplicación Freemium vs Premium

Hemos querido dar una experiencia al usuario de forma gratuita prácticamente total (prueba de ello es que sólo nos hemos concentrado en este apartado), entendemos que la temática de nuestra aplicación tiene que estar orientada a un público cuánto más grande mejor ( y para ello limitar funcionalidad o cortar las funcionalidades que consideramos básicas, nos parecía un error), por dos motivos bastante claros:

* Sin una buena experiencia Freemium, el usuario difícilmente cogerá servicios Premium, sobretodo en temas de dietas y deportes, dónde el usuario suele ser bastante inconstante y si se encuentra un mensaje de “pagar” lo más seguro es que abandone la aplicación
* Nos interesa que la usen un tiempo en sus rutinas diarias, para que, cuándo decidan dar el siguiente paso (contratar especialistas) vayan directamente a nuestra aplicación

Luego esta el motivo de todas las aplicaciones, que es el marketing del “boca a boca”.

Pese a esta convicción, obviamente, nuestra idea es, en un futuro, colocar servicios Premium para poder mantener la aplicación y obtener beneficios.

La principalmente diferencia entre los servicios, es la contratación de expertos, que vayan a comisión para el desarrollo de dietas y/o rutinas personalizadas para los usuarios.  
Los expertos y sus funciones serían:

* Monitores deportivos: Que serían los encargados de desarrollar las rutinas acorde con los requisitos del usuario, adaptarlo al entorno de entrenamiento y a las propiedades físicas del usuario.
* Nutricionista: Que serían los encargados de desarrollar una dieta acorde al usuario, ritmo de vida y deporte que realice.
* Perfil hibrido: Tanto nutricionista como monitor, es capaz de desarrollar ambos roles.

Para ello, el usuario pagaría una cuota mensual que le incluiría el seguimiento por parte del experto (por ende, tendría acceso a las estadísticas de ese usuario), dietas personalizadas (por ende, el experto tendría acceso a un creador de dietas y/o rutinas, que se asignan al usuario) y un chat que ponga en contacto con el especialista y el usuario.

## Tecnología usada

En un principio queríamos utilizar para desarrollar la app web en lenguaje de programación PHP orientado a objetos utilizando el *framework laravel*.

Pero por falta de tiempo para poder aprender cómo trabajar con este *framework*, decidimos que lo mejor para el proyecto era trabajar solamente utilizando programación orientada a objetos usando el patrón de diseño “modelo-vista-controlador”.

El patrón de diseño de software modelo vista controlador tiene como objetivo separar la lógica de la aplicación y los datos de la interfaz de usuario. Los puntos fuertes del patrón MVC es la reutilización de código y la separación de conceptos y gracias a estas características facilitan la tarea de desarrollar aplicaciones en cualquier lenguaje de programación.

Al cabo de una semana de investigación y de aprendizaje empleando el patrón de diseño modelo vista controlador, llegamos a la conclusión que había que dedicarle mucho tiempo para aprender y dominar cómo realizar aplicaciones web empleando esta tecnología. Por ese motivo, decidimos desarrollar todo el proyecto empleando PHP “normal”.

A parte de *PHP*, también hemos desarrollado la aplicación usando los siguientes lenguaje programación y tecnologías:

* Del lado del cliente hemos utilizado*JavaScript* utilizando *AJAX*(*Asynchronous JavaScript And XML*) como técnica de desarrollo web asíncrona con el servidor.
* De lenguaje de diseño gráfico Hojas de estilo en cascada (*CSS* 3).
* De *framework* hemos empleado *bootstrap*.
* Para los gráficos hemos utilizado la librería de *PHP jpgraph*.

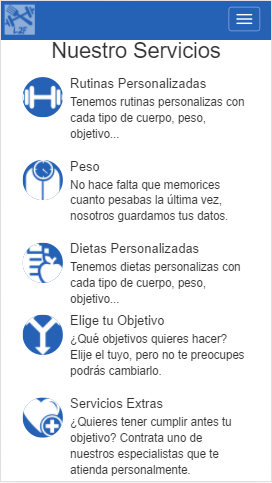
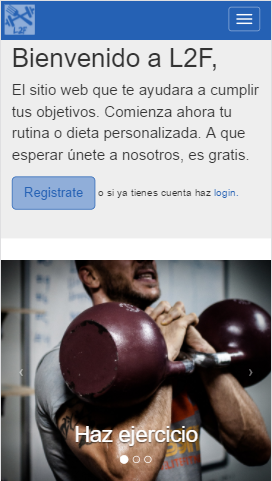
# Organización del sitio web

LifetoFit es estructurada con una página inicial en la cual el usuario deberá de hacer login o registrarse para poder acceder al resto de contenido. Una vez logueado el usuario se encuentra con la main page en la que puede navegar al resto de sitios ya sea su perfil, la sección de rutinas o la dietas.

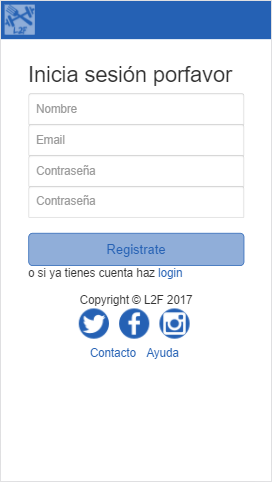
Esquema 1 : Diagrama por bloques del sitio web

# Página Índex

La página de índex, es la primera página que vera el usuario al entrar a nuestro sitio web. Aquí el usuario tendrá una pequeña impresión de los servicios que ofrece L2F, sabrá quien ha desarrollado la aplicación, podrá localizarnos y contactar con nosotros.

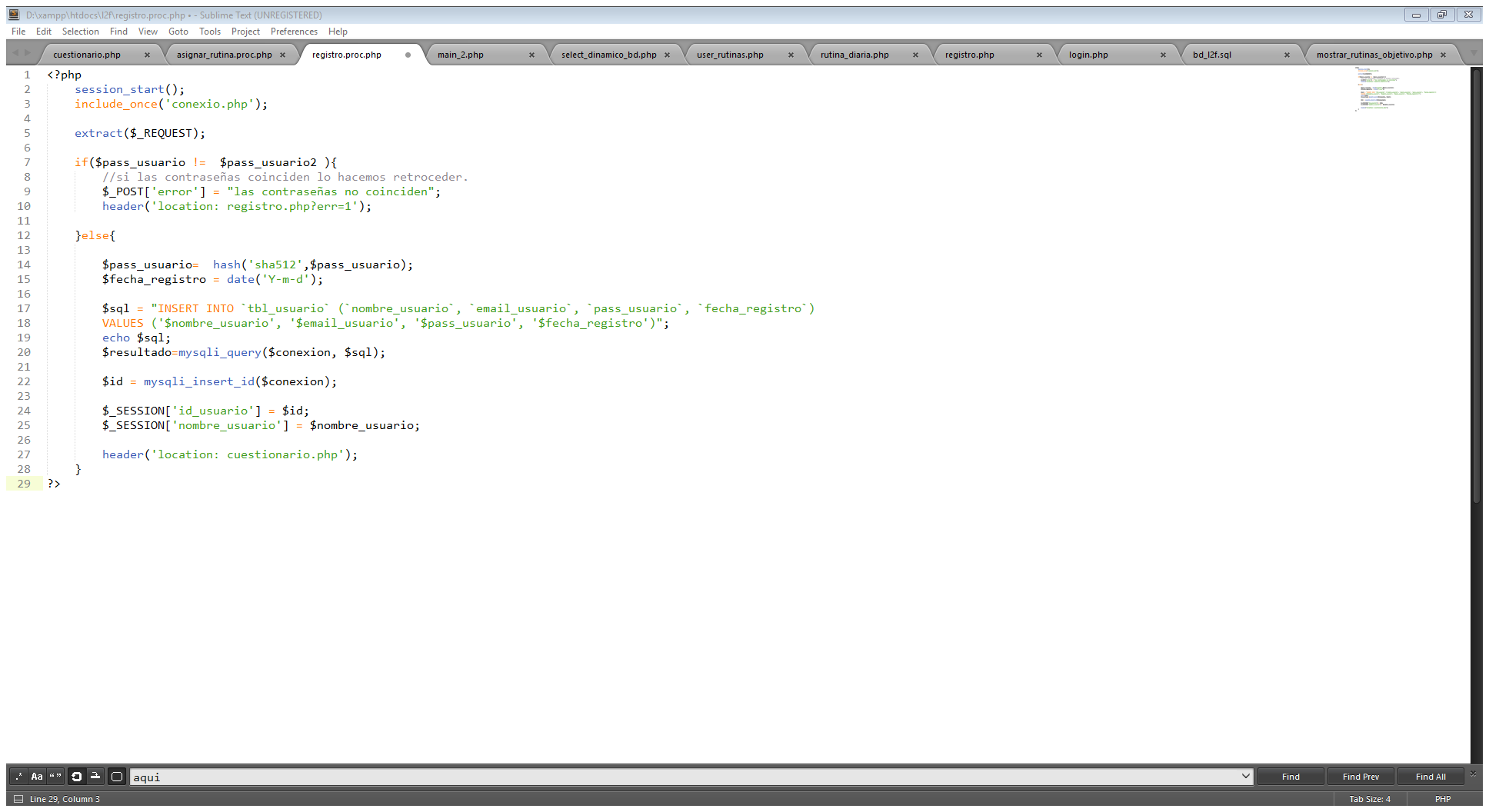


En la parte superior de la página el usuario podrá loguearse o registrarse. Si el usuario decide registrarse, le pedimos como datos iniciales su nombre, el email, una contraseña y su confirmación.



## Programación del registro

Cuando el usuario hace clic en el botón regístrate, el formulario manda los datos por el método POST al proceso *registro.proc.php.* En este proceso se realizará una sentencia sqlInsert a la tabla usuario tal como se puede comprobar en la siguiente captura:



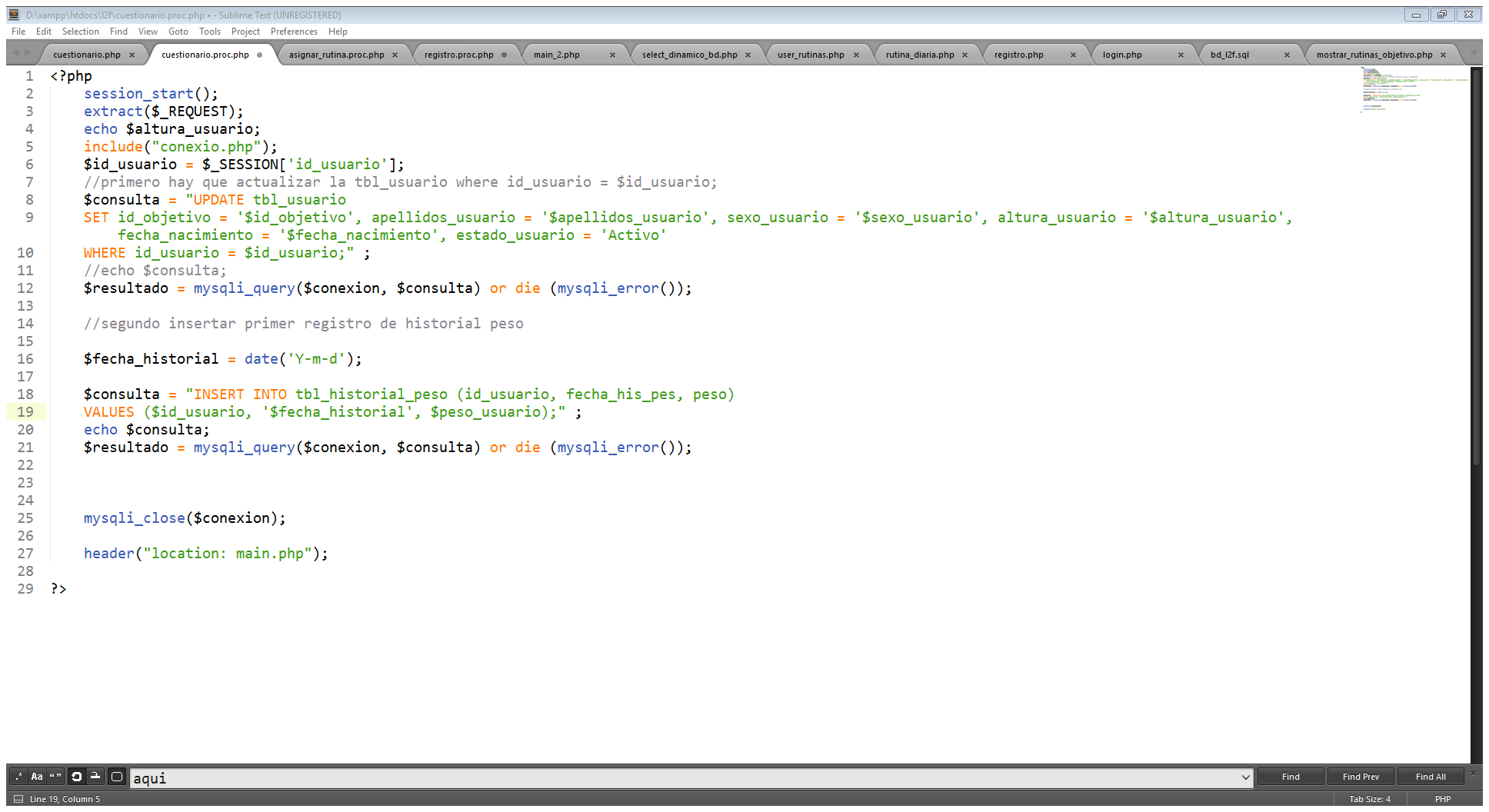
Una vez se ha registrado el usuario, desde un principio, obtenemos en la base de datos esta información, aunque el usuario decida cancelar el proceso de registro. A continuación, obligamos al usuario a completar un cuestionario en el que le pedimos sus apellidos, su fecha de nacimiento, su peso, su altura, su sexo, el objetivo por el cual quiere utilizar la aplicación y que tipo de cuerpo tiene(aproximadamente).



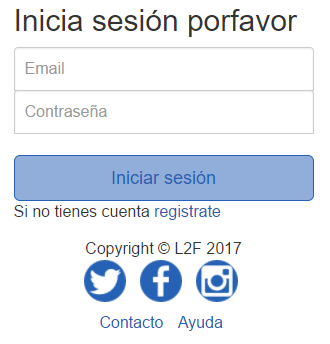
Si el usuario por algún motivo cierra la cesión sin haber completado el formulario obligatorio, podrá loguearse más adelante y rellenarlo. De este modo, no podrá entrar a la página principal y utilizar nuestros servicios.

## Programación del cuestionario

Al hacer clic en enviar se enviará toda la información del formulario al proceso *cuestionario.proc.php* vía método POST. En este apartado, se actualizará la tabla y hará un primer insert a la tabla historial peso.



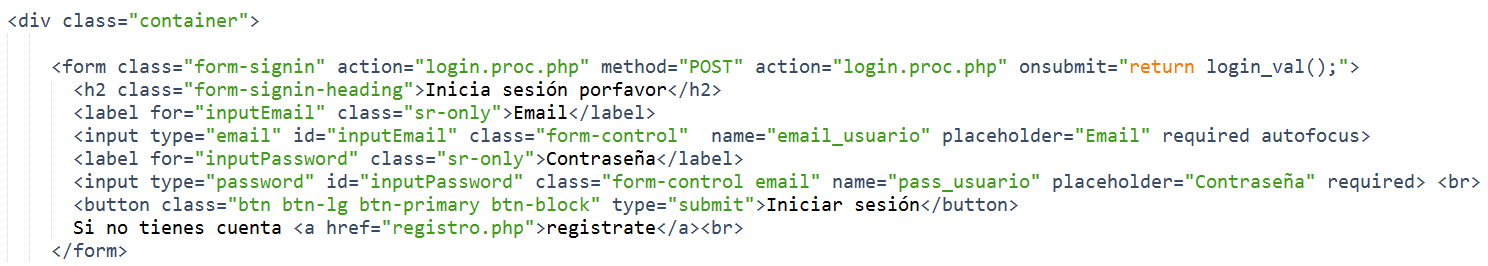
Por el otro lado, si el usuario ya estaba registrado y quiere hacer loginaccedera a la página login.php donde el usuario deberá de ingresar su email y su contraseña.



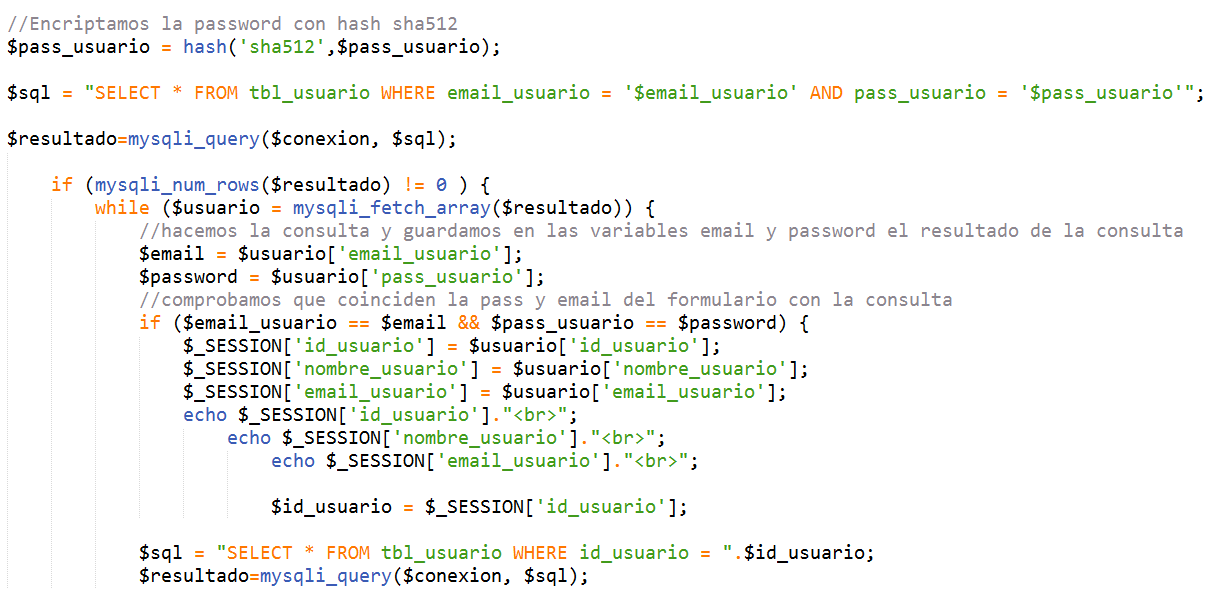
## Programación del login

Si el usuario o la contraseña son incorrectos, el sistema no le dejará acceder, además tiene una validación realizada en HTML y JavaScript para que el email tenga el formato correcto y no esté vacio:



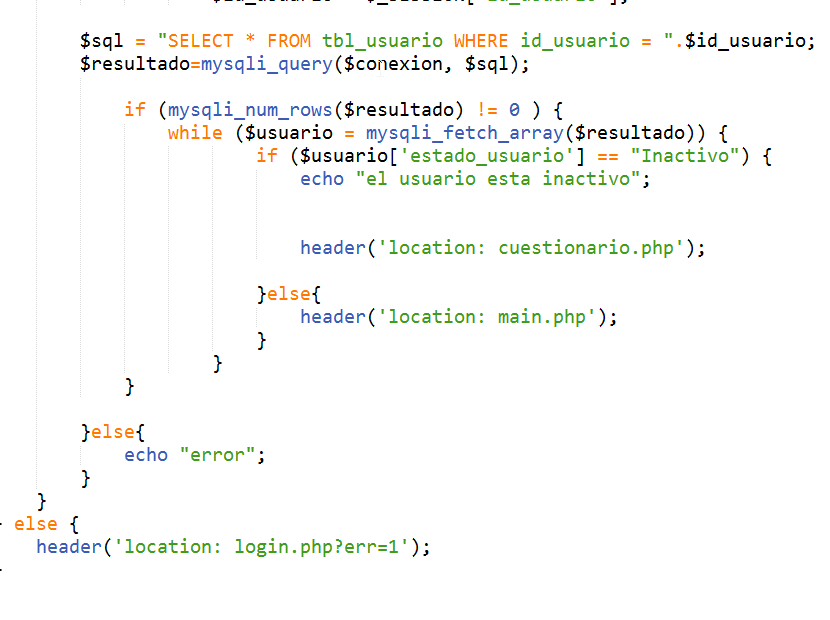


Al completar este pequeño formulario se llama al proceso login.proc.php en el cual se realiza la consulta a la base de datos para comprobar que tanto el usuario como la contraseña con correctos.



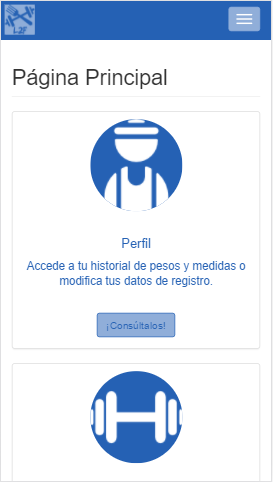
SI el usuario y la contraseña coinciden, se guardan los datos del usuario en las variables de sesión para poder trabajar con ellos más adelante

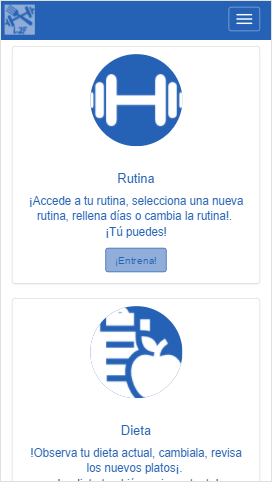
Adicionalmente, se controla que el usuario esté activado o no, lo que significa si el usuario ha pasado por el cuestionario obligatorio o no, si está inactivo aún, se redirige al usuario al cuestionario para que lo realice, de ese modo si no lo ha hecho todavía, le obligamos a pasar siempre por ahí.



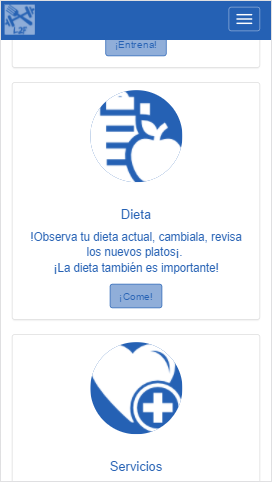
Por lo tanto, si los datos introducidos son incorrectos se redirige al usuario al login de nuevo para que lo vuelva a hacer, o si el usuario está inactivo, se le manda al cuestionario y para acabar, si el login es correcto lo mandamos al main.php.

Una vez ha completado el formulario de información, el usuario es redirigido a la página principal donde podrá elegir a que sección acceder:

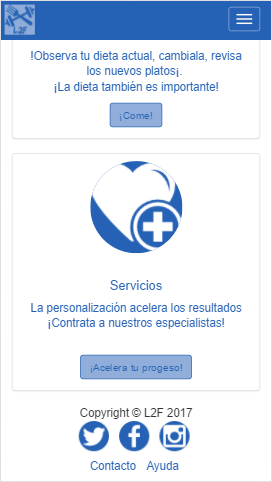
* Perfil: El usuario podrá registrar su peso actual, comprobar la diferencia de peso respecto a su último registro, también podrá apuntarse las medidas de sus músculos y ver su evolución. A parte, podrá modificar sus datos personales y el objetivo elegido.



* Rutina: En esta sección el usuario podrá seleccionar la rutina que mejor le convenga según el objetivo que haya seleccionado Una vez que se ha elegido la rutina, el usuario podrá ver más detalles de los ejercicios o bien empezar a hacer ejercicio.



* Dieta: En la sección de dieta, el usuario elegirá la dieta que se adapte a su objetivo y podrá descargarla para tenerla siempre a mano.



* Servicios: Esta sección estará disponible en la versión Premium, aquí los usuarios se pondrán en contacto con los especialistas (entrenadores y nutricionistas) y les harán rutinas o dietas personalizadas según sus requisitos.

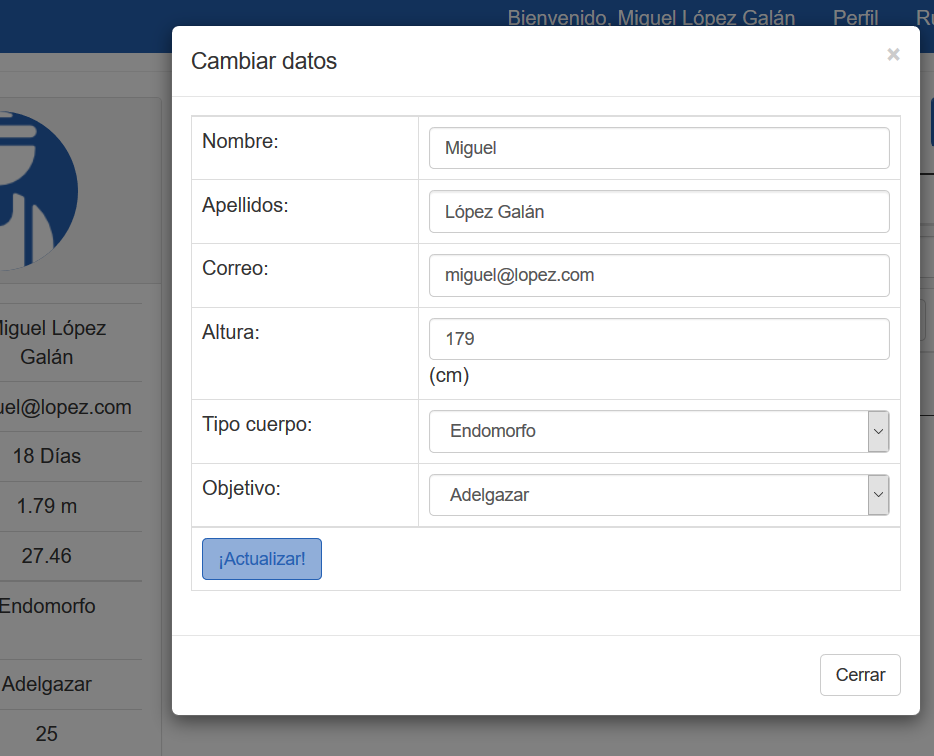
# Página Userperfil.php

En el apartado “Perfil” es dónde mostramos toda la información propia del usuario y permitimos que a pueda modificar.

Esta información se divide básicamente en 3 bloques que serían:

* Información personal: Nombre, correo, tiempo que llevas registrado, altura, IMC (Que se calcula de forma automática y se actualiza conforme el usuario cambia la altura y/o el peso), el tipo de cuerpo, su objetivo y la edad(que se calcula de forma dinámica a partir de su fecha de nacimiento).

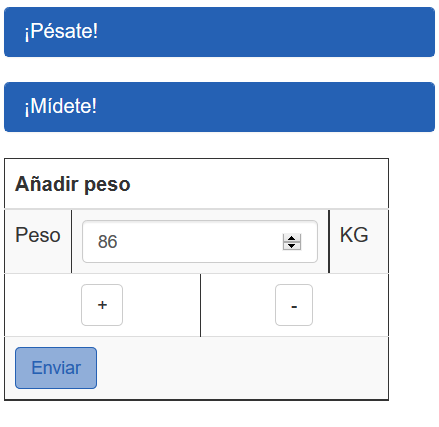
  
Cuándo pulsamos “Editar información”, se nos hable una ventana modal con un formulario que nos dirige a la página “ch\_user\_data.proc.php” (invisible para el usuario) que nos actualiza los datos.



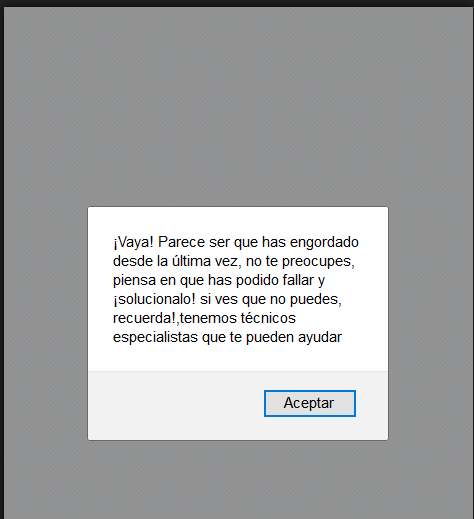
Tanto el “Tipo cuerpo” como “Objetivo” es un Select dinámico, que nos coge los datos de la BBDD

* Peso: Aquí tenemos que diferenciar en dos apartados, el “Historial peso” dónde el usuario puede ver sus últimos tres pesos (y si tiene 3 o más registros le mostramos un botón con un enlace que le muestra TODOS los registros de pesos y una gráfica con su evolución de peso.

  
Luego tenemos el apartado de “Pésate” dónde introducimos nuestro peso actual, cogiendo el valor de nuestro último registro:

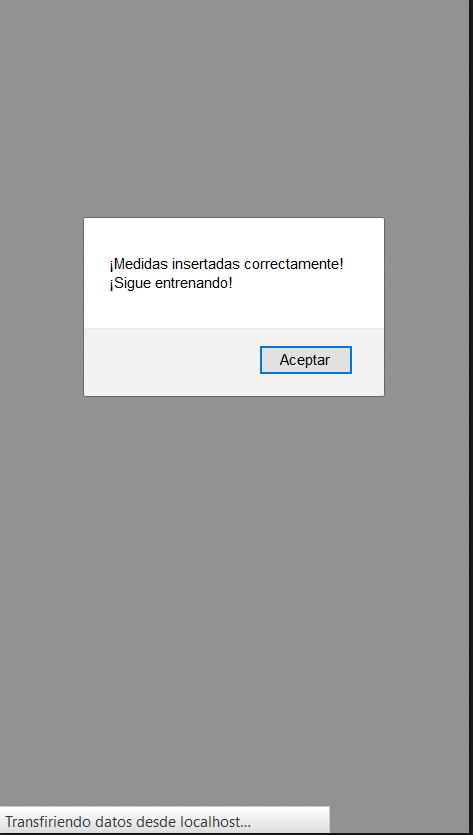


Cuándo le damos a “Enviar” se dirige a una página invisible para el usuario, que introduce el peso y controla si ese peso es mayor o menor que el anterior y comprueba el objetivo, si el usuario ha engordado y su objetivo es adelgazar, le sale el siguiente mensaje:

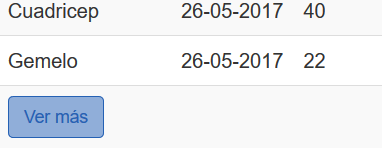


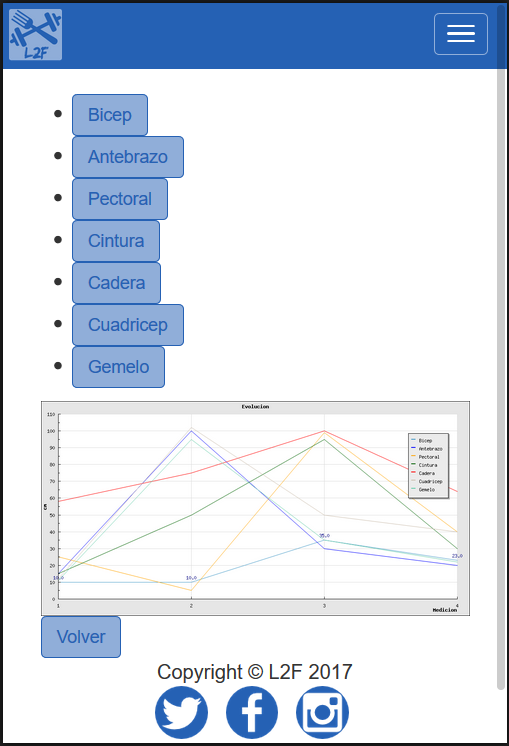
* Medidas: En este apartado, al igual que en el apartado de peso tenemos dos sub-apartados, uno para ver nuestras últimas mediciones en los diferentes grupos musculares (a diferencia del peso, sólo podemos observar la última medida, ya que, son 7 apartados y si se pusieran sus 3 últimas medidas, simplemente no cabría en ninguna pantalla Smartphone).

El insertar medidas se vería así (al igual que el peso, el valor por defecto es nuestra última medida correspondiente a la parte de ese cuerpo).



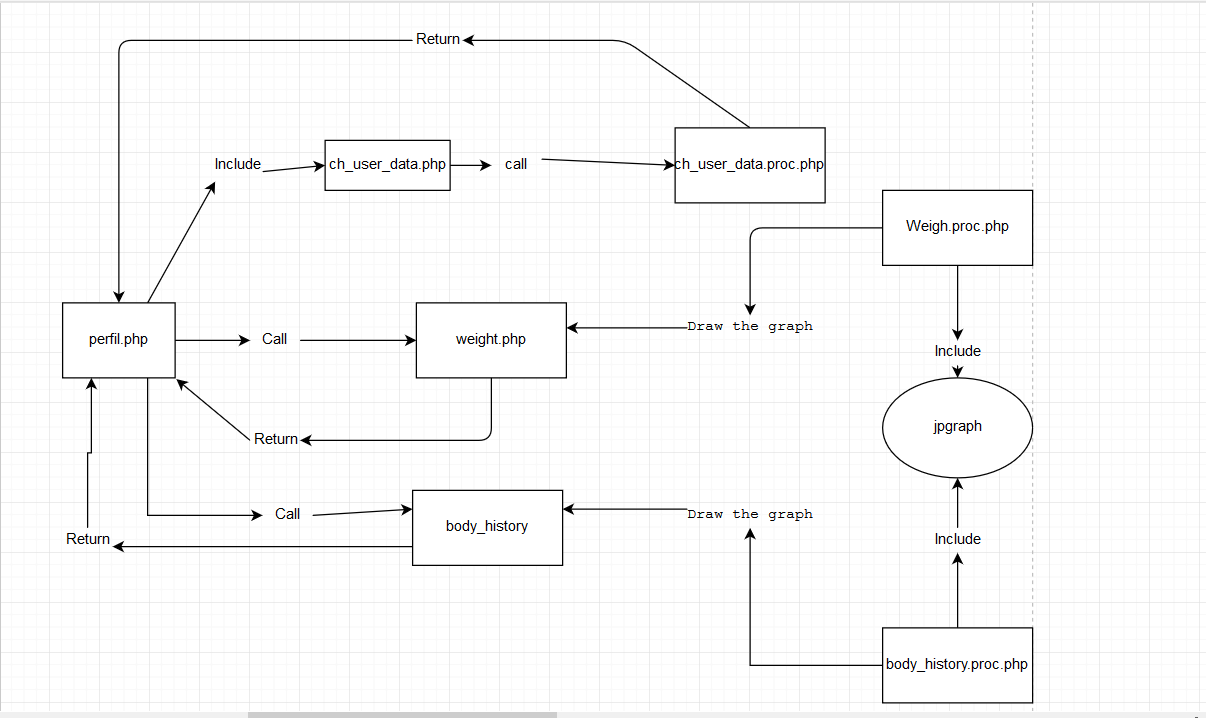
En el historial medidas, también podemos acceder a la página dónde se ven todas las medidas y el gráfico evolutivo, para ello tenemos que pulsar el botón “ver más”:





# Apartado técnico de Userperfil.php

## Diagrama:



En la página perfil.php contiene los historiales y las inserciones de datos en DIVS ocultos, y se muestran usando funciones de JavasCript:

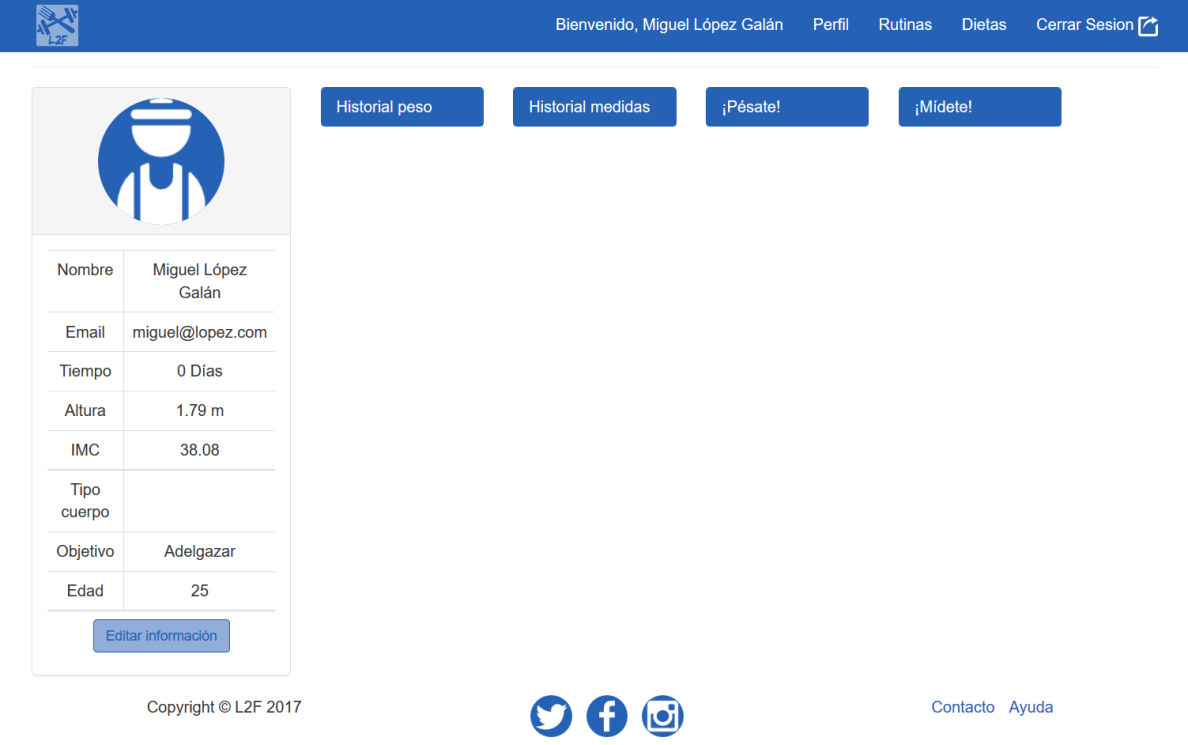


Imagen de cómo se vería el perfil.php nada más entrar

Para ocultar los divs se usa la función “hidden\_elements()” que coge las capas y las oculta:

function hidden\_elements(){

document.getElementById('pr\_cuerpo').style.display='none';

document.getElementById('pr\_peso').style.display='none';

document.getElementById('sh\_peso').style.display='none';

document.getElementById('sh\_medidas').style.display='none';

}

Para mostrar las capas ocultas usamos la función display\_element y le envía el id del elemento para saber si la puede mostrar u ocultar la capa:

function display\_element(table\_id){

if(document.getElementById(table\_id).style.display=='none')

{

document.getElementById(table\_id).style.display='inline-table';

}

else

{

document.getElementById(table\_id).style.display='none';

}

}

Para facilitar la experiencia de usuario, tanto la sección “Pésate” el “Mídete” tienen botones que pulsando incrementa o decrementa el valor de la caja (tanto de peso como de las medidas).

function plus\_cm(id)

{

var cm = document.getElementById(id).value;

cm++;

document.getElementById(id).value=cm

}

function rest\_cm(id)

{

var cm = document.getElementById(id).value;

cm--;

if(cm>=0)

{

document.getElementById(id).value=cm;

}

else

{

document.getElementById(id).value.value=0;

}

}

Como se puede ver se comprueba que el peso o el cm no pueda bajar de 0.

Tanto en el peso como en las medidas se comprueba que el valor solo sea número tanto en validación HTML5 como en validación JavasCript usando una expresión regular:

if(!/^([0-9])\*$/.test(document.getElementById('cm\_brazo').value))

{

msg+="¡ep! la medida del bicep en números \n";

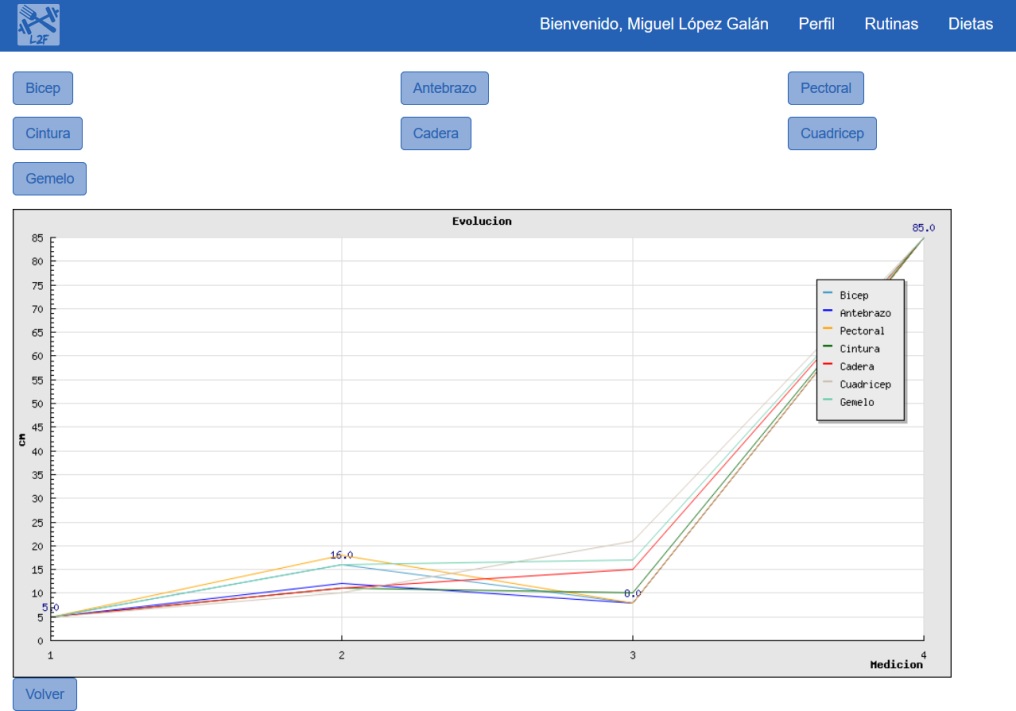
document.getElementById('cm\_brazo').style.borderColor="red";

}

## Body\_history.php

En esta página se puede acceder únicamente cuándo hemos introducido 3 medidas de brazo (al ser obligatorio introducir todos los datos, obviamente tendrá 3 medidas de todas las partes del cuerpo).

Al igual que en perfil, está página al iniciarse están todos los divs ocultos a excepción de los botones para mostrarlos y el gráfico:



Cómo se puede observar solo muestra el div al pulsar el botón.

A nivel de código, no tiene mucho misterio, ya que la librería gráfica (de la que hablaremos más adelante) pinta el gráfico al recibir una llamada del tag img:

<div class='col-md-10'>

<img src="proc/history\_body\_grafic.proc.php" style="width:100%">

<a href="perfilusuario.php"><button class="btn btn-primary">Volver</button></a>

</div>

Con esto nos monta el gráfico.

Luego, para pintar los valores, los coge de forma diferenciada por partes del cuerpo, para ello cogemos los ID con una consulta SQL a la base de datos de partes de cuerpo:

$part\_body\_id\_sql = "SELECT \* FROM tbl\_parte\_cuerpo";

Y luego recorremos los datos y según el valor del nombre asignamos el ID a la parte del cuerpo:

switch ($part\_body\_id['nombre\_parte\_cuerpo']) {

case 'Brazo':

$id\_brazo = $part\_body\_id['id\_parte\_cuerpo'];

break;

Y ya teniendo el ID, solo necesitamos pintar los valores con un SELECT;

while($brazo\_data = mysqli\_fetch\_array($select\_brazo\_query))

{

echo "<tr><td>".date("d-m-Y", strtotime($brazo\_data['fecha\_his\_med']))."</td>";

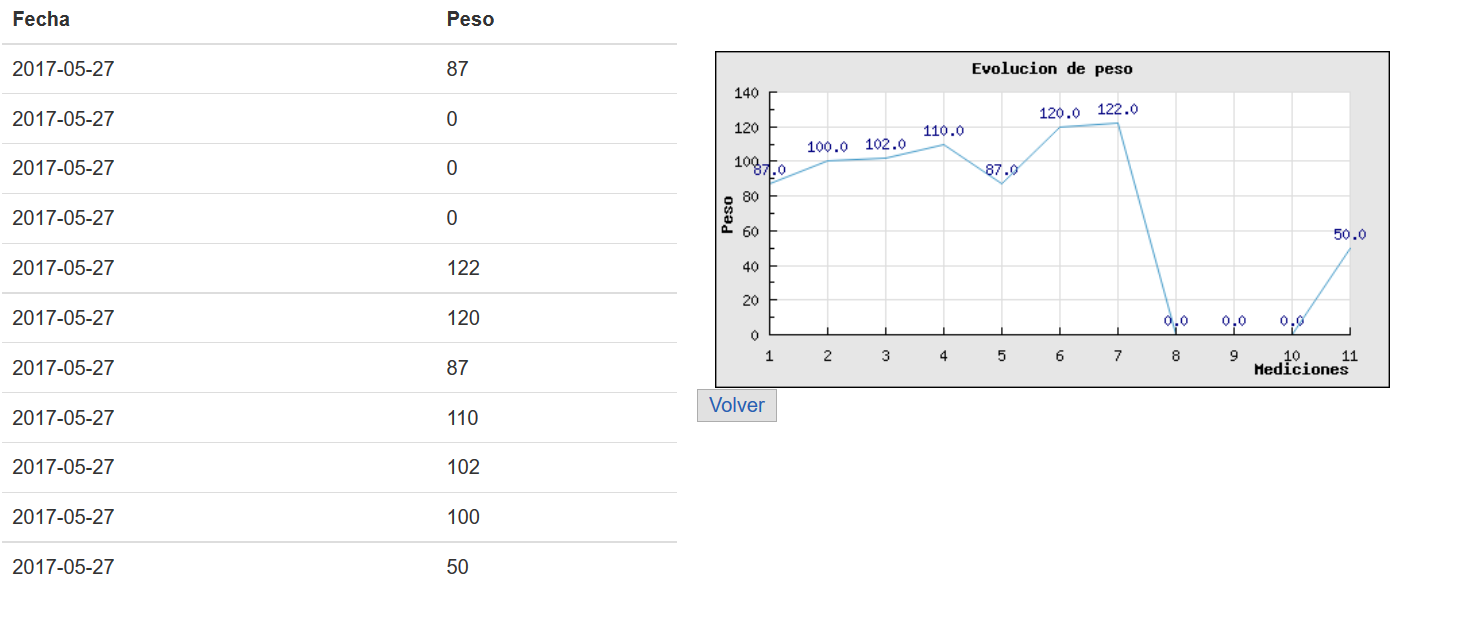
echo "<td>".$brazo\_data['cm']."</td></tr>";

}

Los apartados de mostrar y ocultar divs funcionan exactamente igual que en el perfilusuario.php.

## Weight.php

Esta página es más simple que la anterior, ya que, pese a que también muestra datos, únicamente muestra un único conjunto (en vez de 7 conjuntos como la anterior), por lo tanto podemos mostrar directamente el gráfico y los datos ya que no dificulta la usabilidad de la página el mostrarlo así.



Únicamente tiene una consulta SQL y una el bucle que lo pinta:

$select\_weight\_sql = "SELECT \* FROM `tbl\_historial\_peso` WHERE id\_usuario = ".$\_SESSION['id\_usuario']." ORDER BY `fecha\_his\_pes` ASC";

El bucle que lo pinta:

while($data\_weight = mysqli\_fetch\_array($select\_weight\_query))

{

echo "<tr><td>".$data\_weight['fecha\_his\_pes']."</td><td>".$data\_weight['peso']."</td></tr>";

}

<img src="proc/weight\_grafic.proc.php" style="width:100%">

Y la llamada al gráfico:

## Add.peso.proc.php

Esta página no tiene una apartado de vista de cara al usuario, ya que inserta el peso especificado por el usuario en el apartado de “Pésate”

Hacemos una consulta para obtener su peso actual (antes de la inserción) los datos del usuario y los datos del historial de peso:

$last\_height\_sql = "SELECT \*

FROM tbl\_usuario

RIGHT JOIN tbl\_historial\_peso ON tbl\_usuario.id\_usuario = tbl\_historial\_peso.id\_usuario

RIGHT JOIN tbl\_objetivo ON tbl\_usuario.id\_objetivo = tbl\_objetivo.id\_objetivo

WHERE tbl\_usuario.id\_usuario=".$\_SESSION['id\_usuario']."

ORDER BY `tbl\_historial\_peso`.`fecha\_his\_pes` DESC

LIMIT 0,1";

Mientras insertamos los datos, comprobamos que el peso antes de la inserción sea mayor o menor al que ingresa y su objetivo sea adelgazar sacara un mensaje de alert indicando que tenemos técnicos especialistas que le pueden ayudar:

if($last\_height\_data['nombre\_objetivo']== 'Adelgazar')

{

if($last\_height\_data['peso']<$height)

{

echo "<script type='text/javascript'>

alert('¡Vaya! Parece ser que has engordado desde la última vez, no te preocupes, piensa en que has podido fallar y ¡solucionalo! si ves que no puedes, recuerda!,tenemos técnicos especialistas que te pueden ayudar');

</script>";

}

}

Como futura mejora, haríamos diferentes comprobaciones según objetivos y peso, además de controlar los datos también en el servidor.

## Add\_medidas.proc.php

Este apartado es muy similar al anterior, pero tratamos cada apartado de forma individual (para ello también sacamos los ID de forma dinámica (como cuándo mostramos las medidas).

Luego tratamos las inserciones de forma individual (para asegurarnos que los datos son correctos);

$sql\_his\_brazo = "INSERT INTO `tbl\_historial\_medidas` (`id\_usuario`, `fecha\_his\_med`, `id\_parte\_cuerpo`, `cm`) VALUES ('".$\_SESSION['id\_usuario']."', '".$fecha\_actual."', '".$id\_brazo."', '".$cm\_brazo."');";

## Ch\_user\_data.proc.php

Este apartado es bastante sencillo, recoge los datos del formulario, los inserta muestra un mensaje y devuelve a la página anterior.

if(mysqli\_query($conexion,$updt\_user\_sql))

{

echo "<script type='text/javascript'>alert('¡Los cambios han sido realizados correctamente!');

location.href='../perfilusuario.php';</script>";

}

$updt\_user\_sql = "UPDATE `tbl\_usuario` SET `nombre\_usuario` = '".$ch\_name."', `email\_usuario` = '".$ch\_mail."', `apellidos\_usuario` = '".$ch\_lastname."', `altura\_usuario` = '".$ch\_tall."', `id\_objetivo` = '".$ch\_objective."', `id\_tipo\_cuerpo` = '".$ch\_type\_body."' WHERE `tbl\_usuario`.`id\_usuario` = ".$\_SESSION['id\_usuario'].";";

# Jpgraph

Esta librería es la que nos pinta los gráficos y as llamadas se ejecutan mediante el tag “<IMG>” de HTML.

Es una librería con un estilo algo antiguo, pero de fácil uso, ligera y libre, por eso nos hemos decantado por ella.

Para explicar su funcionamiento aprovecharemos la explicación de las páginas de weight\_grafic.proc y de history\_body\_grafic.proc para hablar de su funcionamiento.

## Weight\_grafic.proc.php y history\_body\_grafic.proc.php

A diferencia de las otras páginas, estas dos páginas tienen la conexión a la base de datos escritas directamente en el código, ya que da un error si se ejecuta el include.

Esta librería pinta las gráficas usando arrays, por eso necesitamos iniciar el array que contendrá los datos del peso:

$weight\_array=array();

Y en el bucle lo rellenamos de datos:

while($weight = mysqli\_fetch\_array($select\_weight\_query))

{

array\_push($weight\_array,$weight['peso']);

Para iniciar el gráfico, necesitamos incluir la librería y el php correspondiente al tipo de gráfico que queremos implementar:

require\_once ('../libs/jpgraph/src/jpgraph.php');

require\_once ('../libs/jpgraph/src/jpgraph\_line.php');

Añadimos la configuración de los estilos y los márgenes:

// Creamos el grafico

$grafico = new Graph(500,250,"auto");

$grafico->SetScale('textlin');

$grafico->img->SetAntiAliasing();

$grafico->xgrid->Show();

// Ajustamos los margenes del grafico

$grafico->SetMargin(40,30,30,40);

Ahora creamos la barra que contiene los datos del peso:

// Creamos barras de datos a partir del array de datos

$bplot = new LinePlot($weight\_array);

//Días

$lineplot=new LinePlot($data\_array);

$lineplot->SetColor("black");

$lineplot->SetWeight(2);

// Configuramos color de las barras

$bplot->SetColor('#479CC9');

// Queremos mostrar el valor numerico de la barra

$bplot->value->Show();

La añadimos al gráfico:

//Añadimos barra de datos al grafico

$grafico->Add($bplot);

$grafico->Add($lineplot);

Configuramos los títulos:

// Configuracion de los titulos

$grafico->title->Set('Evolucion de peso');

$grafico->xaxis->title->Set('Mediciones');

//$grafico->xaxis->hide();

$grafico->yaxis->title->Set('Peso');

Configuramos las propiedades del texto:

$grafico->title->SetFont(FF\_FONT1,FS\_BOLD);

$grafico->yaxis->title->SetFont(FF\_FONT1,FS\_BOLD);

$grafico->xaxis->title->SetFont(FF\_FONT1,FS\_BOLD);

Mostramos el gráfico:

$grafico->Stroke();

# Página de Rutina

El usuario en el apartado de rutinas deberá de seleccionar una rutina de las disponibles según su objetivo para poder hacer los ejercicios correspondientes y así llevar el control de qué sesión lleva y que le toca hacer en ese momento.

La primera vez que accede el usuario a la sección de rutinas, deberá seleccionar una de las que les aparece, según su objetivo.

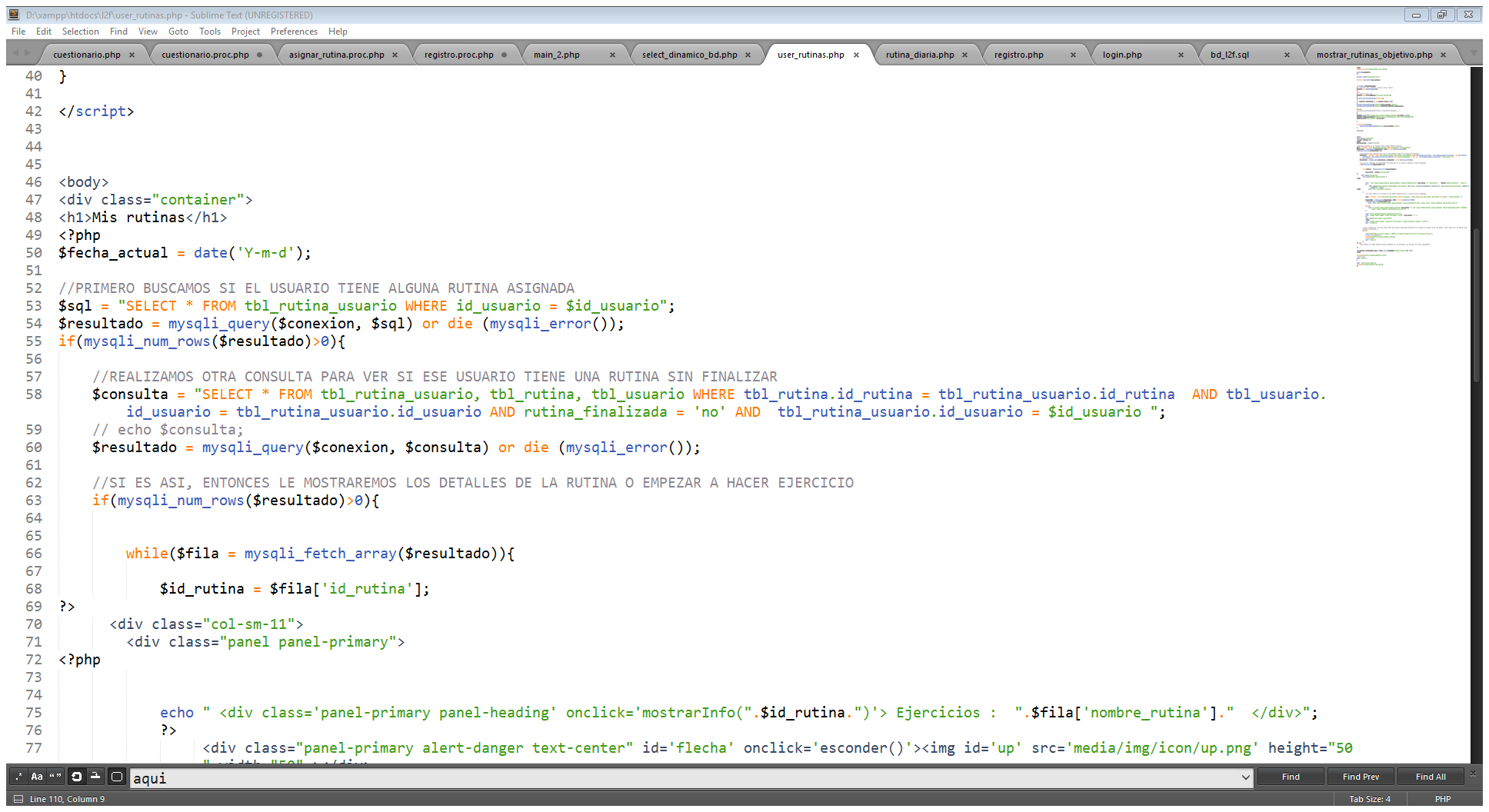


## Programación Mostrar Rutinas Disponibles

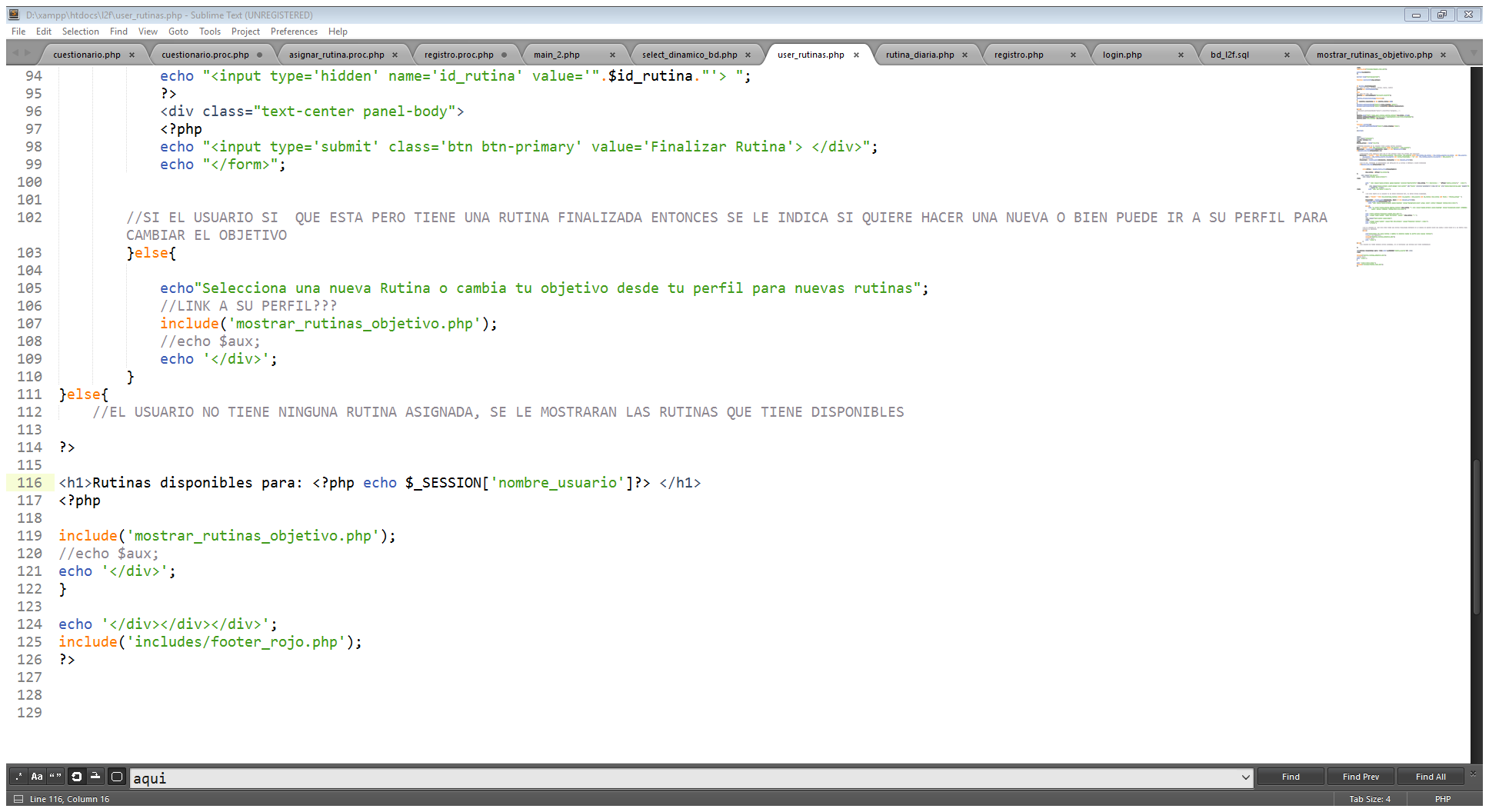
En este apartado se ha programado teniendo en cuenta los posibles siguientes escenarios:

1. Si el usuario tiene alguna rutina asignada ya
2. Si el usuario tiene alguna rutina sin finalizar
3. Si el usuario tiene una rutina finalizada ya
4. Si el usuario ya ha hecho ejercicio hoy
5. Si el usuario no tiene ninguna rutina asignada.

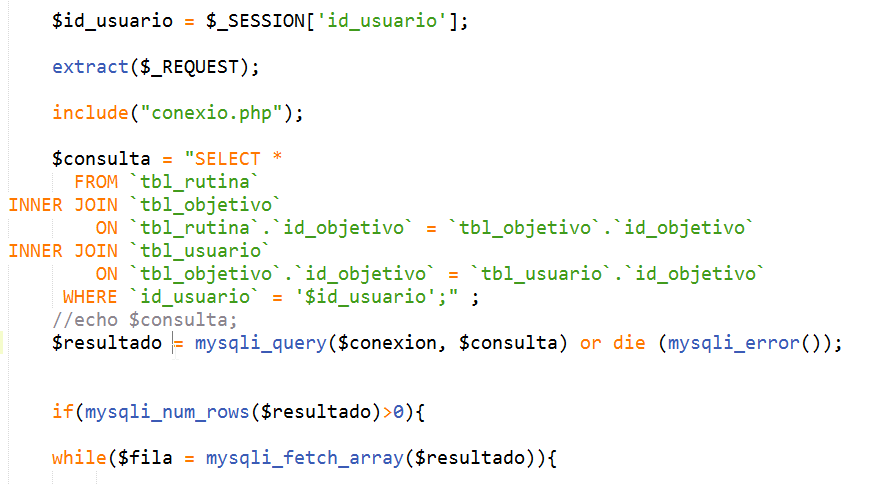
Primero de todo, buscamos en la base de datos si el usuario está presente en la tabla rutina usuario, es a decir, si el usuario tiene una rutina asignada, sin tener en cuenta el estado de ella. Si no hay coincidencias en la consulta, entonces significa que el usuario no tiene ninguna rutina asignada (el caso 5), por lo tanto, le mostramos todas las rutinas que tiene disponible.



En ese caso se incluye la él PHP *mostrar\_rutinas\_objetivo.php* que realiza está consulta:

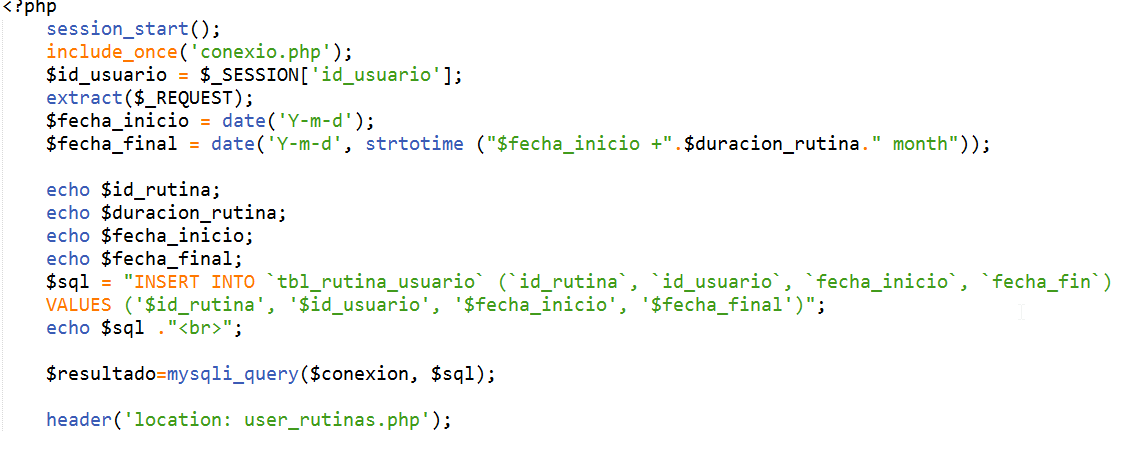


Y este PHP tiene la siguiente sentencia SQL para mostrar todas las rutinas que tienen el mismo objetivo que el usuario:

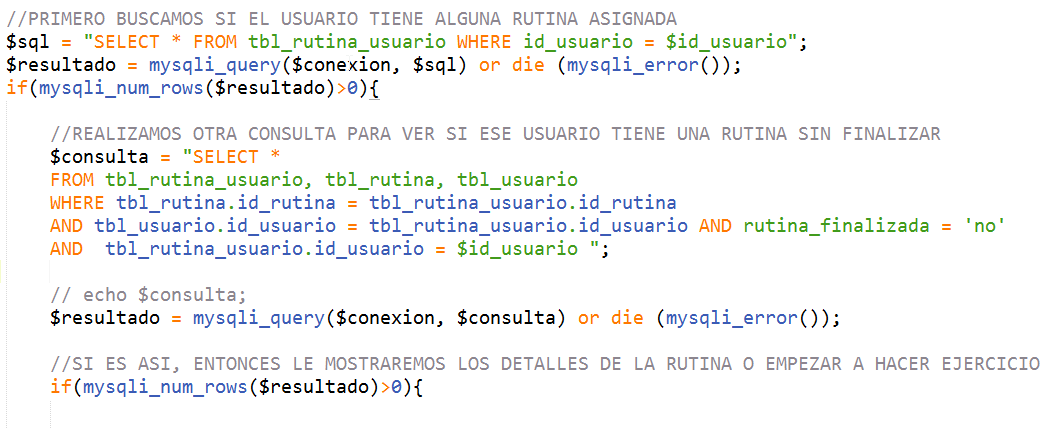


## Programación Asignar Rutina

SI el usuario ha decidido a que rutina quiere apuntarse, solo debe apretar el botón de apúntate, el cual llama al procedimiento asignar\_rutina.proc.php el cual hacer una sentencia sql para insertar en la tabla\_rutina\_usuario los datos del usuario y la rutina seleccionada.



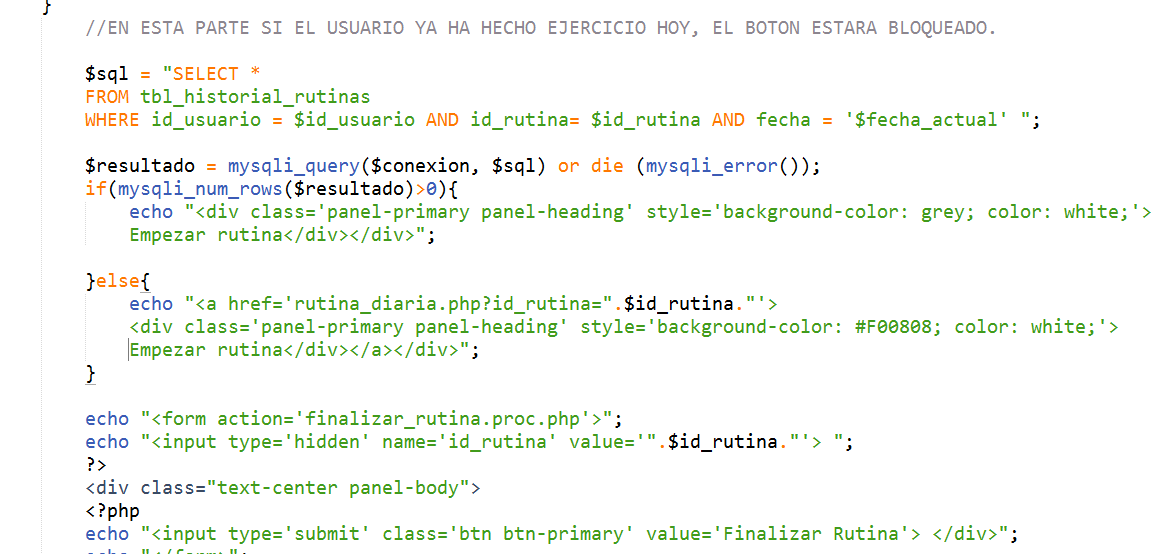
Volviendo al primer caso, si el usuario aparece en la tabla rutina usuario, entonces se realiza otra consulta para comprobar si el usuario tiene alguna rutina no finalizada, para ello se realiza la siguiente consulta:

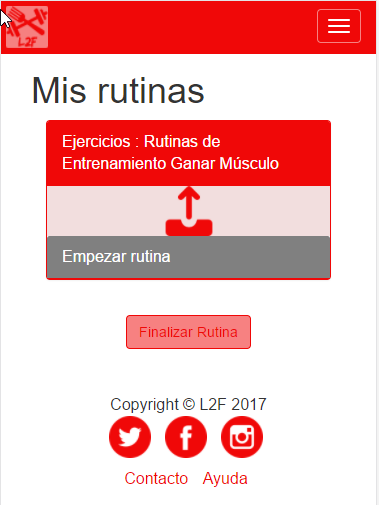


Si hay resultados (el caso 2), entonces mostraremos la opción de mostrar los ejercicios de la rutina o de empezar la rutina:

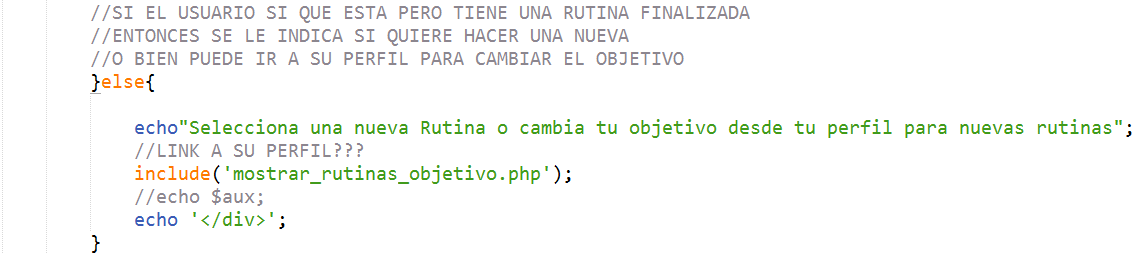


Si se da el caso 4, que el usuario ya haya hecho ejercicio hoy, antes de mostrar los botones anteriores, se realiza una consulta a la tabla historial rutinas para ver si hay coincidencias con el usuario, la rutina y la fecha actual. Si se da el caso, el botón de empezar rutina estará bloqueado.

Este sería el resultado:



En el último caso pendiente, si el usuario tiene una rutina pero se encuentra finalizada, se le mostrarán otra vez las rutinas que tiene disponible según su objetivo, pero con un mensaje que le indica que si lo desea, puede modificar su objetivo en el apartado de su perfil.



## Programación Detalle Rutina

Cuando el usuario tenga asignada la rutina le aparecerán dos paneles, uno para abrir en vertical todos los ejercicios que forman parte de esa rutina y en el otro para empezar a hacer ejercicio. En la primera opción también se permite ver más detalles de los ejercicios.

Al hacer clic en la primera pestaña, se realiza una llamada AJAX al fichero *ajax\_ejer\_rutinas.php*en el cual se mostrarán todos los ejercicios de la rutina que tiene asignada el usuario.

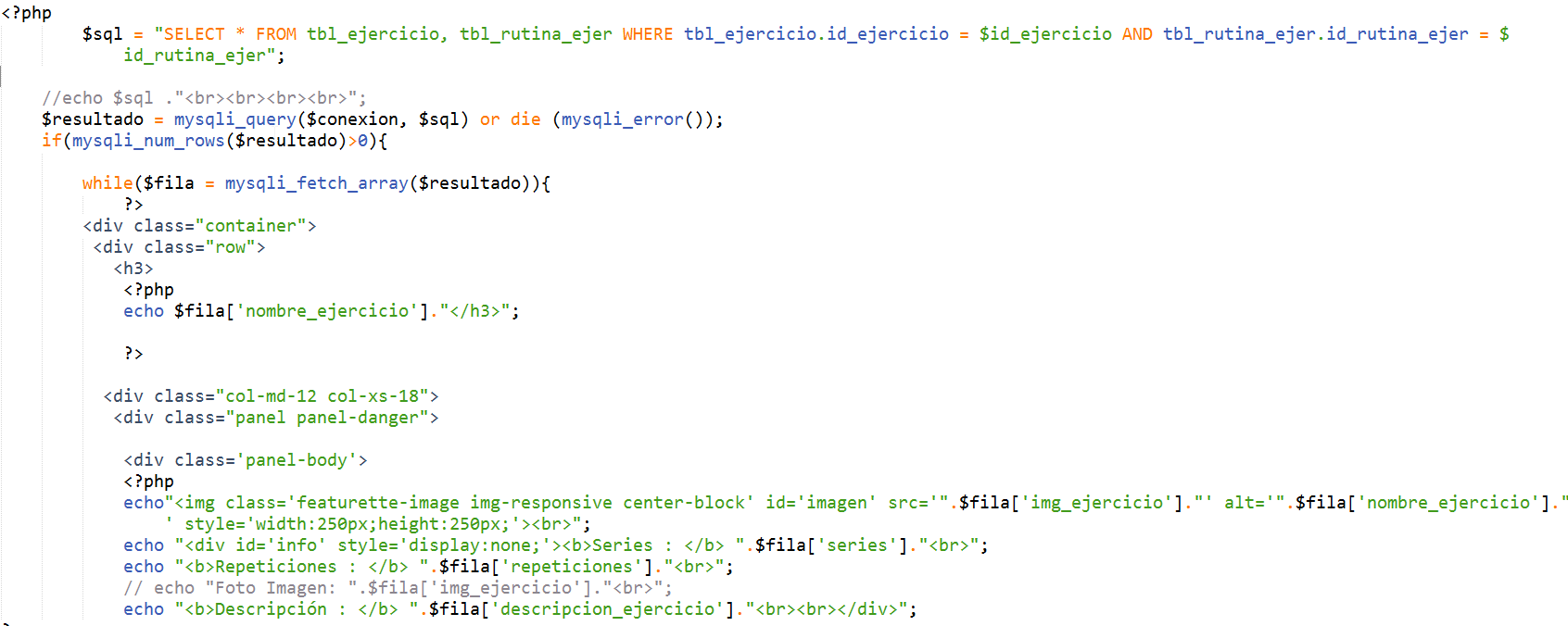


El resultado de la llamada AJAX es el siguiente:

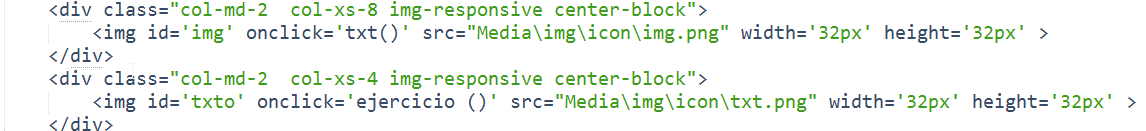


## Programación Detalle Ejercicio

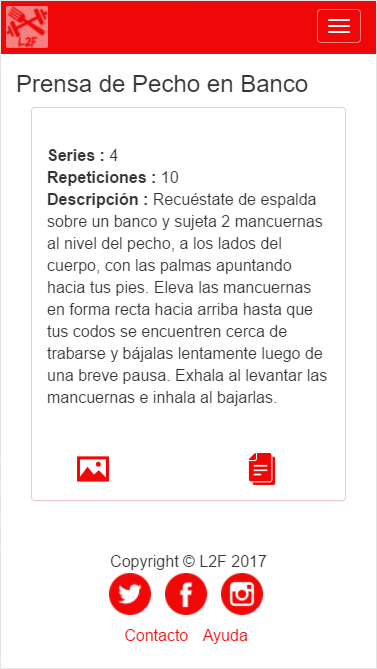
Al presionar en el botón ver más detalles se irá a la página *detalle\_ejercicio.php* en la cual se mostrará un GIF del ejercicio y la descripción de él. Aquí se realiza una consulta para mostrar el nombre, la imagen, la descripción, etc., del ejercicio seleccionado mediante su id:



También hay que comentar, que hay dos botoncitos que intercambiaran entre la imagen y la descripción gracias a los scripts JS que al hacer el evento onclick mostrará o el gif o el texto:





El resultado de este procedimiento es el siguiente:



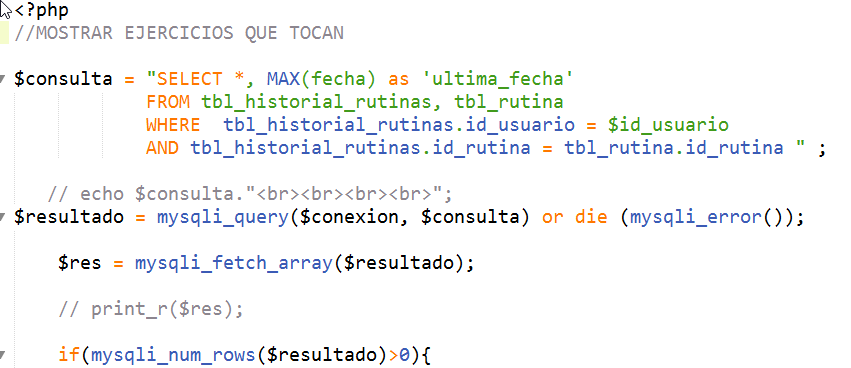
## Programación Empezar Rutina

En la pantalla anterior al hacer clic en la opción de Empezar Rutina nos mostrarán los ejercicios de la sesión que nos toca, por ejemplo, si es el primer día que hacemos la rutina empezaremos por la primera sesión.

La programación tiene los siguientes casos:

1. El usuario no tiene ningún dato en el historial de rutinas
2. El usuario tiene datos en el historial rutinas
   1. La última sesión de ejercicios es igual al número de sesiones a la semana de la rutina
   2. La última sesión de ejercicios es mayor al número de sesiones a la semana de la rutina
   3. La ultima sesión de ejercicios es menor al número de sesiones a la semana de la rutina

En el primer caso, si el usuario no está en el historial rutinas se mostrarán los ejercicios de la primera sesión de esa rutina.

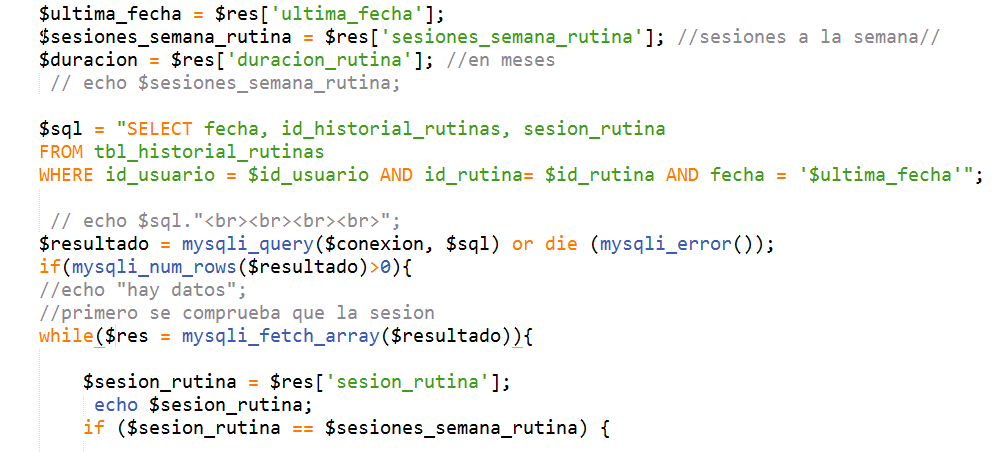




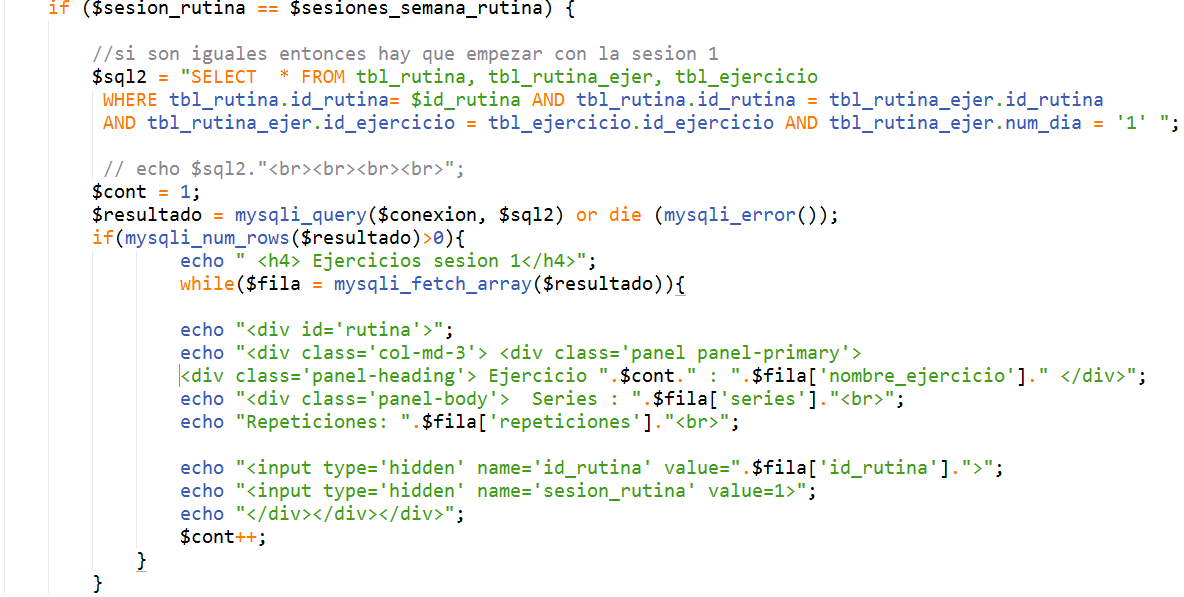
En este caso el resultado sería el siguiente:



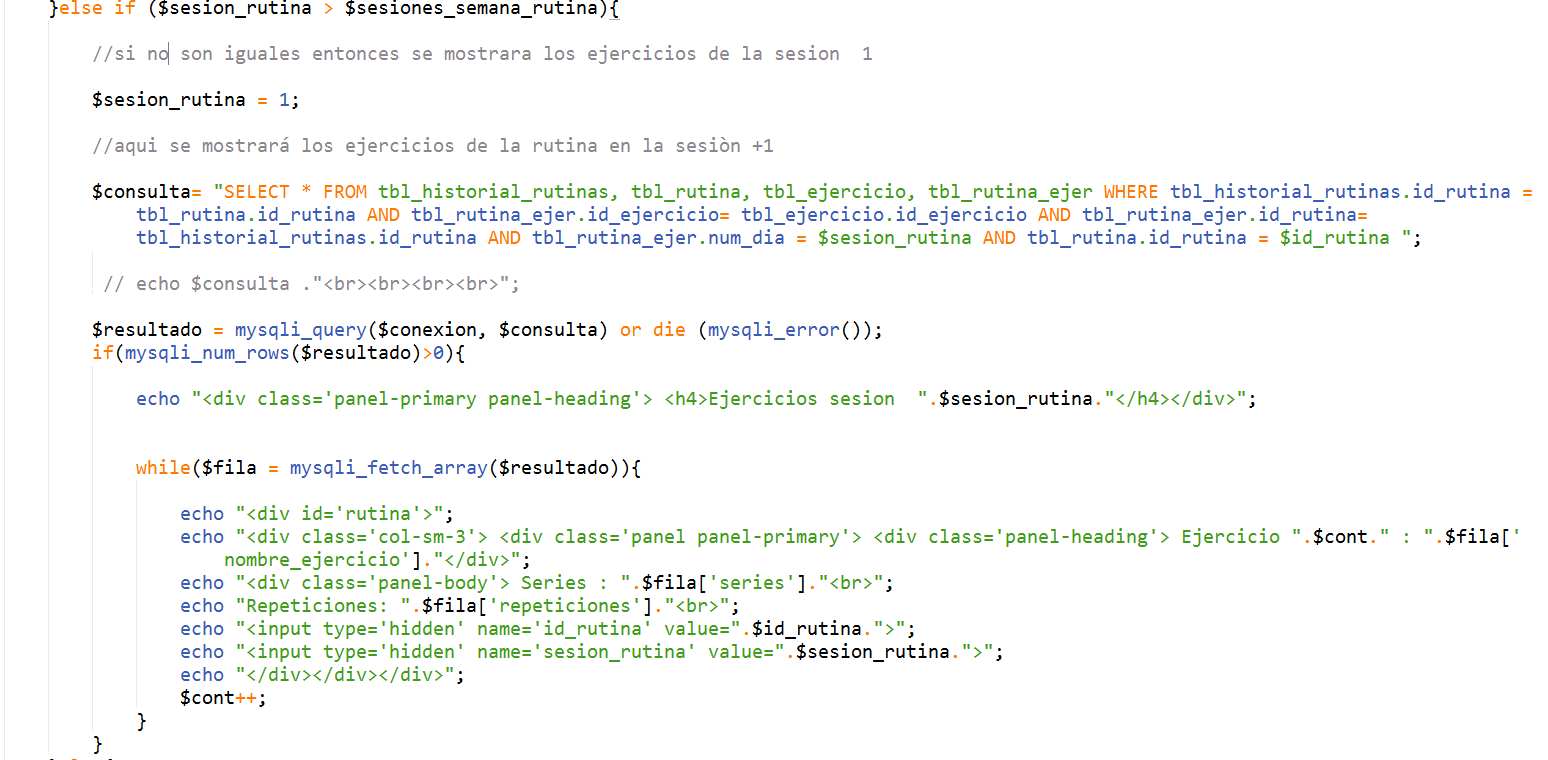
En el segundo caso, si el usuario si que aparece en el historial de rutinas, habrá que mirar cual es el número de sesión del último registro (fecha más antigua).



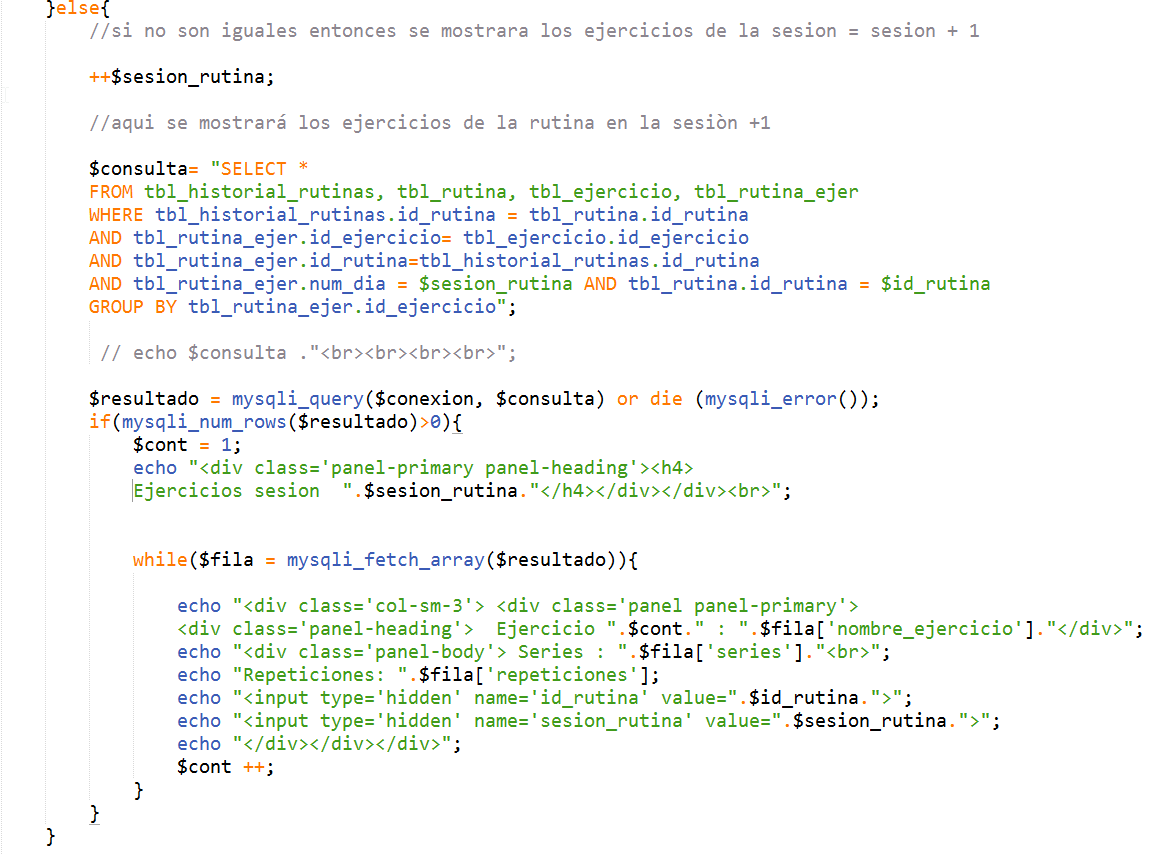
Si la sesión de la rutina es igual el número de sesiones por semana de la rutina entonces se mostrará la primera sesión de ejercicios.



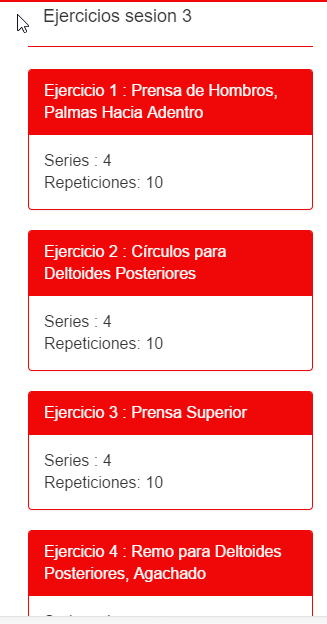
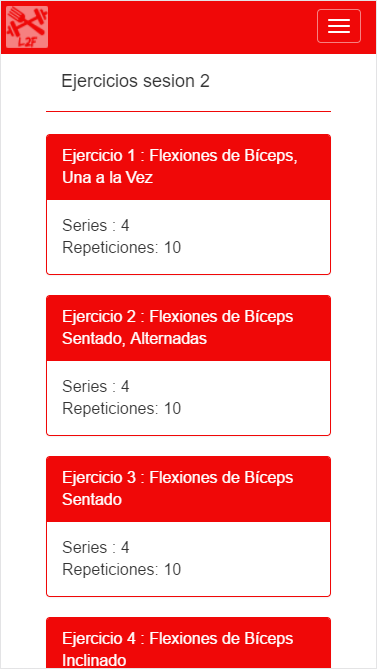
Si se diera el caso que la sesión del último registro del historial rutinas es mayor que el número de sesiones a la semana de la rutina entonces se tendría que volver a mostrar la primera sesión de ejercicios.



En el último caso, si la sesión del último registro de la tabla historial de rutinas es menor que el número de sesiones por semana de esa rutina entonces se tendrá que aumentar en 1 la sesión de la rutina y hacer una consulta a la base de datos para mostrar los ejercicios de la siguiente sesión de la rutina.



El resultado de esta consulta sería,por ejemplo, los siguientes con la sesión 2 y la sesión 3. Como se puede comprobar, cada sesión tiene sus ejercicios correspondientes para trabajar una parte del cuerpo diferente.

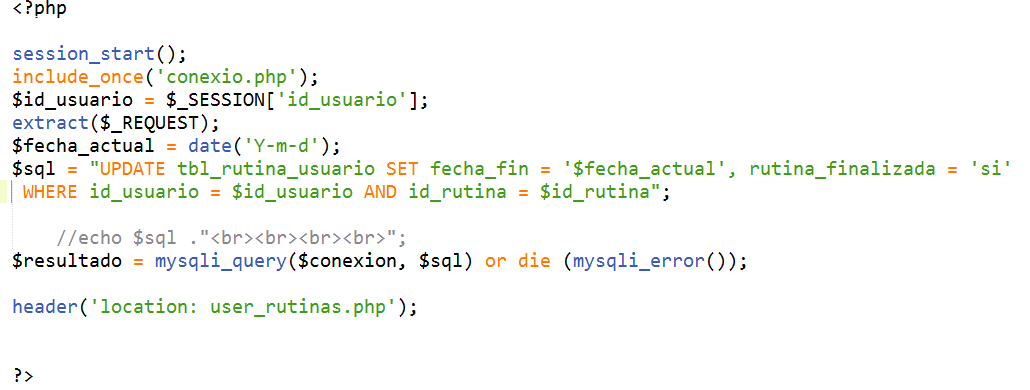


## Programación Finalizar Rutina

Cuando el usuario llegue al fin de su rutina, cumpliendo el tiempo propuesto o bien quiera cambiar de rutina o simplemente decide dejarla, tiene a su disposición un botón al final de la pantalla user\_rutias para finalizarla.



Este botón llama al proceso finalizar\_rutina.proc.php , que actualiza la base de datos poniendo la fecha de finalización de la rutina e indicado que esa rutina ha sido finalizada :

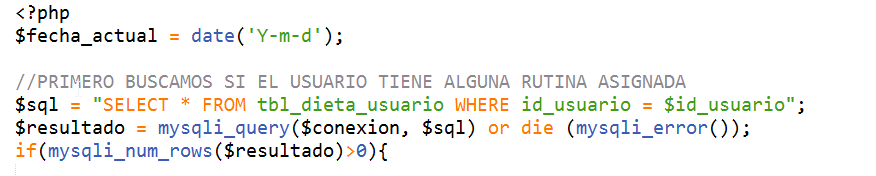


# Página de Dietas

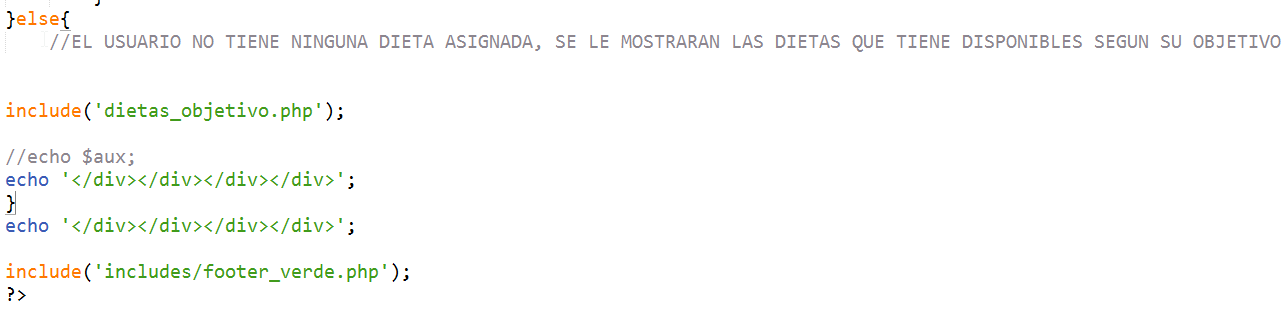
En este apartado el usuario seleccionará una dieta de las disponibles según su objetivo y podrá descargarse la dieta en un PDF para poder tenerla en cualquier momento e incluso imprimirla.

## Programación Selección de Dieta

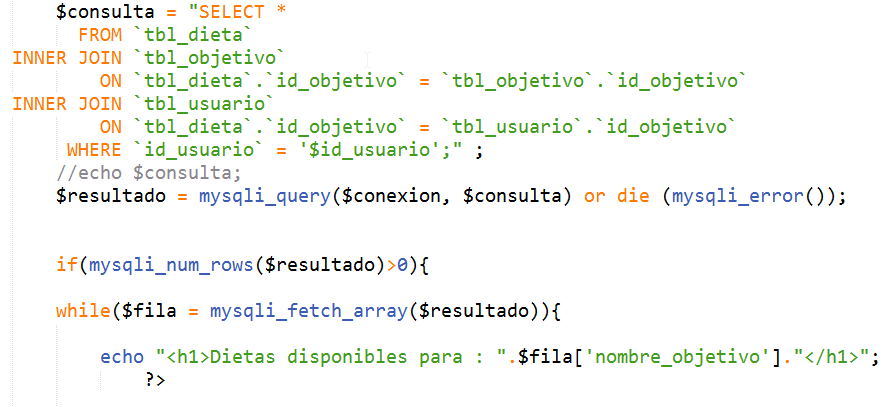
En el apartado de Mi dieta, el usuario podrá seleccionar las dietas que tenga disponible según el objetivo que tiene propuesto. Para ello, primero se hará una consulta a la base de datos para saber si el usuario tiene alguna dieta asignada.



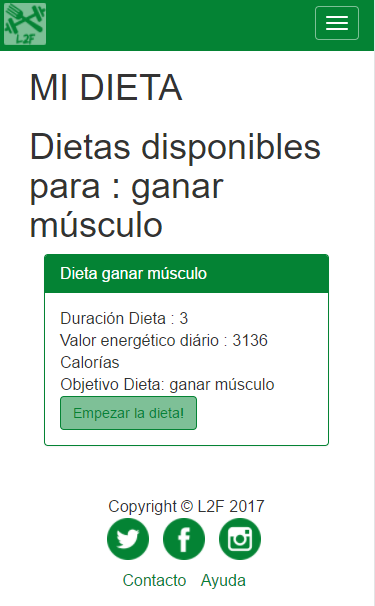
Si el usuario no está en los registros de la base de datos, entonces le mostraremos todas las dietas disponibles para su objetivo.



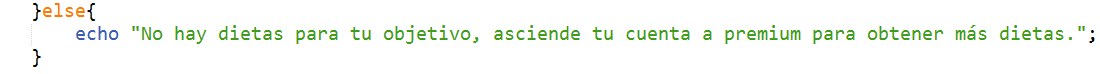
Para mostrar las dietas disponibles invocamos al fichero *dietas\_objetivo.php* que mostrará por pantalla todas la dietas que coincidan con el id objetivo del usuario.



Así es como queda el resultado de este proceso :



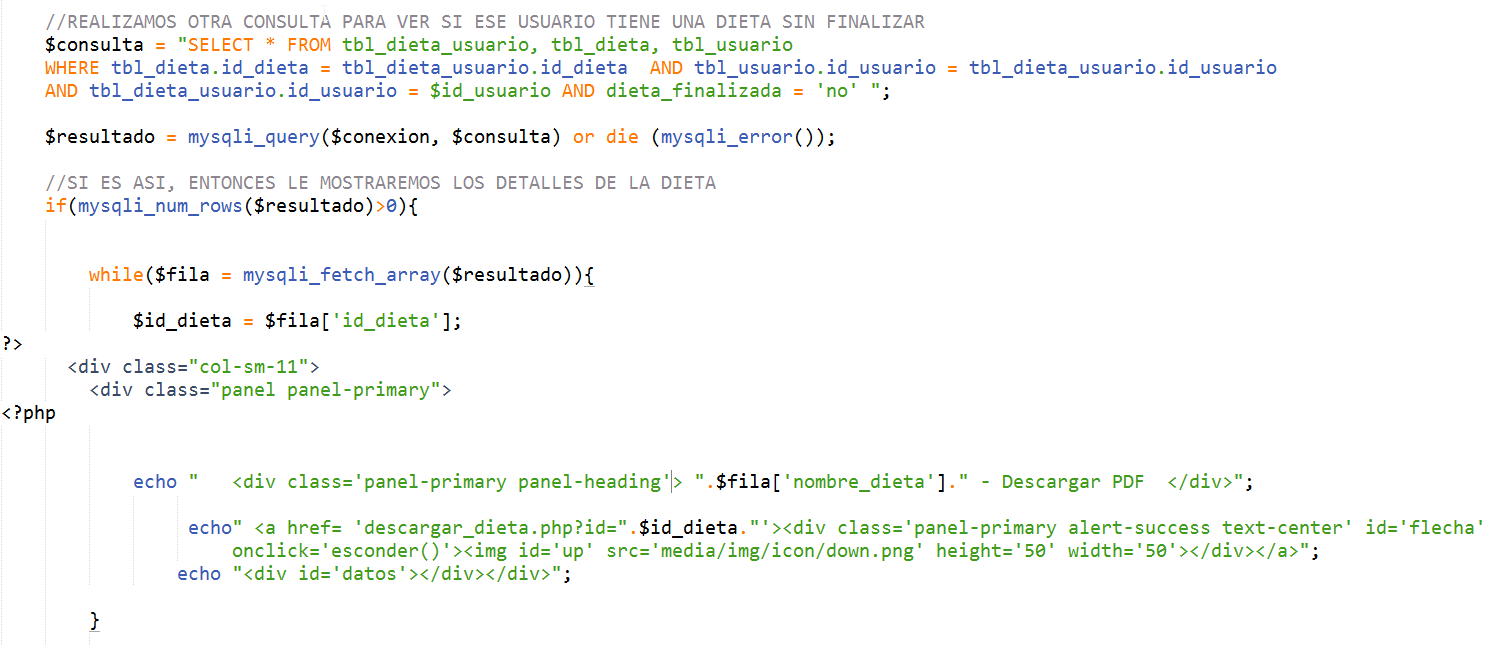
Si la consulta no devuleve ningún resultado, entonces se muestra el siguiente mensaje:



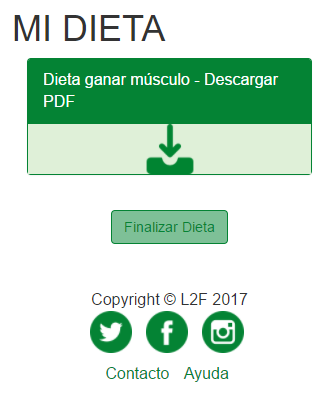
En el caso anterior, si el usuario si que aparece en la tabla dieta\_usuario , se realiza otra consulta para ver si la dieta del usuario esta sin finalizar. Si hay resultados, entonces se mostrará por pantalla un panel para poder descargar la dieta en pdf o finalizarla.

## Programación Asignar Dieta

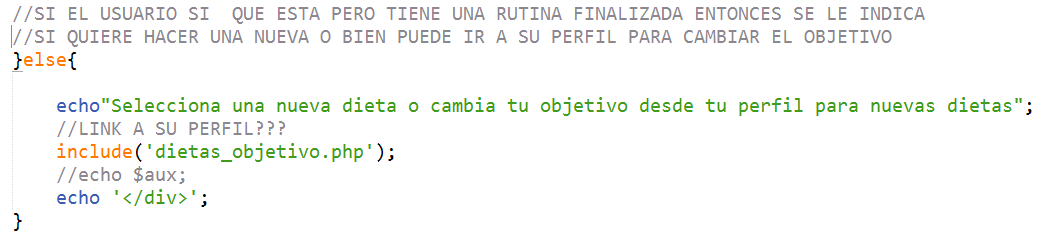
Si el usuario decide seleccionar algúna dieta, hará clic en el botón empezar la dieta, que llamará al procediemiento asignar\_dieta.proc.php

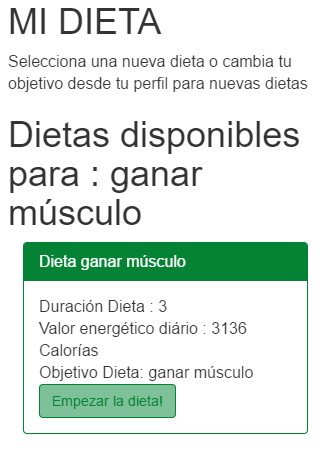


El resultado visual de esta consulta es el siguiente:



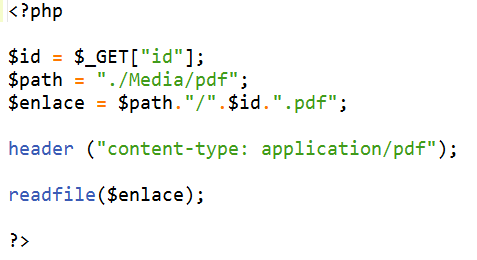
Y en el caso que el usuario tenga alguna dieta pero se encuentre finalizada ya, se le indicara un mensaje para que seleccione una nueva dieta o en su perfil, modifique el objetivo para realizar otras dietas y a continuación se llama al fichero dietas\_objetivo.php explicado anteriormente.





## Programación Descargar Dieta

En el apartado de descargar el documento pdf con la dieta se ha realizado en el fichero descargar\_dieta.php con la siguiente programación:



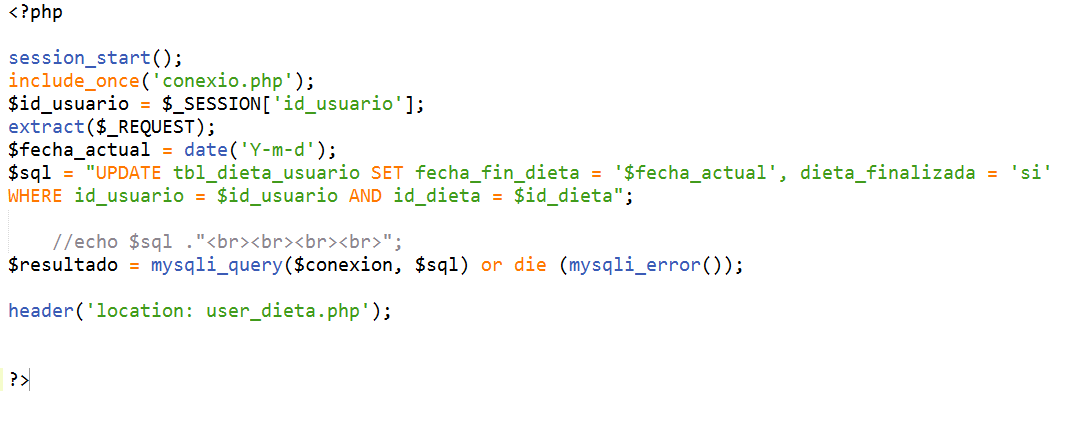
Con el botón se abre el link a este fichero y se le pasa el parametro id dieta por metodo GET, y se le indica la ubicación del fichero, a continuación se concatena con el id y el path para generar el pdf con la terminación .pdf, se le indica que tipo contenido es, en este caso abrir con la aplicación pdf y se abre el fichero para leer ubicado en el enlace.

El resultado es que nos abre en el mismo navegador el documento pdf en el cual hay la dieta a seguir :



## Programación Finalizar Dieta

Para finalizar la dieta hay que hacer clic en el botón finalizar dieta que llama al proceso finalizar\_dieta.proc.php, en el cual actualiza la base de datos poniendo la fecha en la que se ha acabado y cambiando el campo dieta\_finaliza a si.



# Conclusiones Tecnicas y opinión personal

Miguel: En mi opinión creo que ha sido un proyecto muy interesante, dónde he podido aprender de POO y del modelo de las 3 capas en un lenguaje que no había tocado esos apartados (como es PHP), también he podido recordar la presión de tener un proyecto grande con una fecha de entregan ajustada, teniendo que hacer concesiones (Como dejar laravel de lado o la POO o el MVC) para poder centrarnos más en la cantidad de trabajo que teníamos que hacer, igualmente ha sido un proyecto muy interesante y bastante potente, que no tengo dudas que continuare avanzando por mi parte.

# Índice de imágenes

# Webgrafía

<http://jpgraph.net/>

<https://desarrolloweb.com/articulos/1987.php>