

LIFETY

Aplicaciones Web

Integrantes:

Alejandro Magarzo Gonzalo

Leyre Herranz Fernandez

Sandra Ramos Ramos

Iván Ledesma Casado

Diego Álvarez Acarregui

Marcos Docampo Prieto-Puga

ÍNDICE

Listado de scripts para las vistas	4
Comunes	4
cabecera.php	4
menu.php	4
anuncios.php	4
pie.php	4
plantilla.php	4
Vistas	4
index.php	4
login.php	5
logout.php	5
registro.php	5
registronutri.php	5
rutinas_dietas.php	5
planificaciondietas.php	5
tablaRutina.php	6
EntrenadorPersonalEnt.php	6
EntrenadorPersonalUsu.php	6
Esquema de flujo	7
Listado de scripts adicionales	8
Manejo de la base de datos	8
Aplicacion.php	8
Usuario.php	8
Rutina.php	9
Dieta.php	9
Implementación de contenido html	9
ChatEntrenador.php	9
ChatUsuario.php	9
Planificacion.php	9
PlanificacionDietas.php	10
Formularios	10
FormularioDietas.php	10
FormularioLogin.php	10
FormularioLogros.php	11
FormularioPago.php	11
FormularioRegistro.php	11
FormularioRutinas.php	11
Estructura de la base de datos	12
Tablas	12
chat	12
comidas	12

dieta	12
ejercicios	12
planificacion	13
premium	14
profesional	14
usuario	14
Relaciones	15
Prototipo funcional del proyecto	15
Ejecución	15
Como nuevo usuario	15
Como usuario ya existente	18
Como usuario ya existente premium	24
Como entrenador	26
Datos de usuarios para la prueba	27
Errores conocidos	27
Planteamiento para el resto de funcionalidades	28

1. Listado de scripts para las vistas

a. Comunes

Las vistas comunes: menú, pie, anuncios y cabeceras se encuentran en `‘/includes/vistas/comun/’` y la plantilla está en `‘/includes/vistas/plantillas/’`.

i. cabecera.php

El fichero *cabecera.php* muestra un mensaje de bienvenida y a la derecha contiene la opción para loguearse (llamando a login) o registrarse (llamando al registro). En caso de que un usuario haya iniciado sesión, mostrará un mensaje dando la bienvenida a ese usuario.

ii. menu.php

El fichero *menu.php* ubica una lista de navegación en la parte izquierda de la pantalla mediante un `<nav>`. Se puede navegar por la aplicación clicando en los enlaces, que varían según el tipo de internauta que se encuentra en la página. Un usuario no registrado solo puede ver de momento el *Inicio*, un usuario registrado puede ver *Inicio*, *Planificación* y *Nutricionista...*

iii. anuncios.php

El fichero *anuncios.php* ubica la región de los anuncios en la zona derecha de la pantalla. Es parte de una funcionalidad todavía no implementada. De momento solo contiene un `<aside>` con texto, pero en el futuro podrá alojar el anuncio que contrate a alguien.

iv. pie.php

El fichero *pie.php* contiene la parte baja de la página web. Ahora mismo solo contiene un `<footer>` con texto, pero en el futuro incluimos links a nuestras “redes sociales” o a una pantalla para solicitar el uso de nuestro espacio de anuncios.

v. plantilla.php

El fichero *plantilla.php* sirve para establecer una pantalla donde mediante el uso de requires aparecen las vistas (menu, anuncios, cabecera y pie). También se usa la ruta *RAIZ_APP* para obtener la ruta de dichas vistas guardadas.

b. Vistas

i. index.php

Es la página principal donde se muestra el contenido que es público y sería visible para todos los usuarios. La funcionalidad que interactúa con la página de inicio todavía no está implementada y de momento solo muestra párrafos de texto.

ii. login.php

Permite al usuario iniciar sesión con un nick y una contraseña.

El fichero *login.php* crea una instancia de la clase “`class FormularioLogin extends Formulario`” y llama a su método de gestión de formularios y guarda en una variable el HTML que mostraría por pantalla el formulario login. El contenido se inserta en la plantilla.

iii. logout.php

Permite al usuario cerrar sesión y de esta manera se pierden las variables de sesión que almacenaba el usuario y permite que se vuelva a iniciar sesión desde otro usuario o a registrarse por medio de login o registro.

iv. registro.php

Permite al usuario introducir todos los datos necesarios para registrarse. Estos datos son ‘nombre’, ‘apellidos’, ‘nick’, ‘contraseña’.

El fichero *registro.php* crea una instancia de la clase “`class FormularioRegistro extends Formulario`” y llama a su método de gestión de formularios y guarda en una variable el HTML que mostraría por pantalla el formulario registro. El contenido se inserta en la plantilla.

v. registronutri.php

Permite al usuario solicitar un nutricionista/entrenador para poder (de momento) chatear con él. Se rellena un formulario con los campos útiles para el nutricionista/entrenador ‘altura’, ‘peso’, ‘alergias’ y ‘observaciones’.

El fichero *registronutri.php* crea una instancia de la clase “`class FormularioPago extends Formulario`” y llama a su método de gestión de formularios y guarda en una variable el HTML que mostraría por pantalla el formulario registro. El contenido se inserta en la plantilla.

vi. rutinas_dietas.php

Es el menú donde salen tanto el formulario de dietas como el formulario de rutinas para que el usuario pueda introducir los campos y enviar el formulario que desee y así obtener una tabla de una dieta o de una rutina.

El fichero *rutinas_dietas.php* crea una instancia de la clase “`class FormularioRutinas extends Formulario`” y “`class FormularioDietas extends Formulario`”, llama a sus métodos de gestión de formularios y los guarda en una variable el HTML. El contenido se inserta en la plantilla.

vii. planificaciondietas.php

Muestra la tabla con la dieta personalizada del usuario.

El fichero *planificaciondietas.php*, usando la configuración crea una instancia de la clase “`class PlanificacionDietas`” para así poder llamar a su método que muestra la tabla. El contenido se inserta en la plantilla.

viii. tablaRutina.php

Muestra la tabla con la rutina personalizada del usuario.

El fichero *tablaRutina.php*, usando la configuración crea una instancia de la clase “`class Planificacion`” para así poder llamar a su método que muestra la tabla. El contenido se inserta en la plantilla.

ix. EntrenadorPersonalEnt.php

Muestra el chat del entrenador con el usuario, y un formulario para añadir logros.

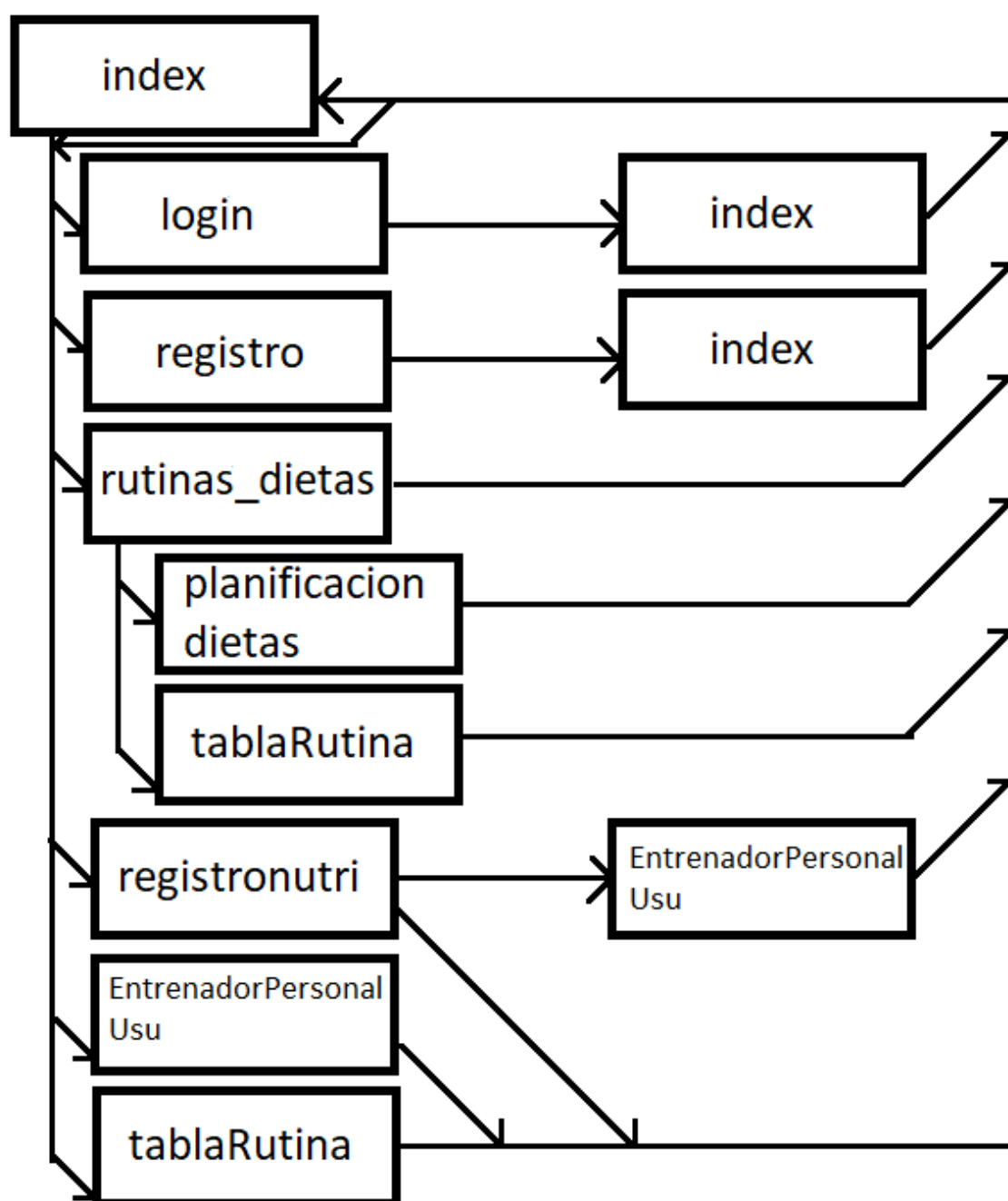
El fichero *EntrenadorPersonalEnt.php*, usando la configuración crea una instancia de la clase “`class ChatEntrenador`” para así poder llamar a su método que muestra el chat. El contenido se inserta en la plantilla. También se crea una instancia de “`class FormularioLogros extends Formulario`”, que ejecuta la lógica para asignar los logros.

x. EntrenadorPersonalUsu.php

Muestra el chat del usuario con el entrenador y los logros.

El fichero *EntrenadorPersonalUsu.php*, usando la configuración crea una instancia de la clase “`class ChatUsuario`” y “`class ChatLogros`” para así poder llamar a sus métodos que muestran el chat y los logros.

c. Esquema de flujo



2. Listado de scripts adicionales

a. Manejo de la base de datos

i. Aplicacion.php

En primer lugar tratamos Aplicacion.php que es una “`class Aplicacion`”. Su principal finalidad es la gestión de la conexión con la BBDD. Se inicia la conexión, se cierra la conexión de la base de datos y además, se crea la instancia de la base de datos para que otros phps puedan acceder a través de un get a dicha instancia y puedan acceder para leer o modificar las distintas tablas que componen la base de datos.

ii. Usuario.php

Aquí podemos encontrar “`class Usuario`” que permite realizar operaciones sobre los usuarios de la base de datos. Encontramos un constructor junto con la lista de variables necesarias para la clase que representan todas las columnas de la tabla usuario en la base de datos.

El método `login($alias, $password)` sirve para poder comprobar que existe un usuario con su alias correspondiente a la contraseña introducida comprobando los datos en la base de datos. El método crea instancia un usuario y procede a llamar al método insertar para corroborar que se puede insertar al usuario en la base de datos con los parámetros introducidos. `buscaPorId($id)` y `buscaPorAlias($alias)` permite buscar a un usuario por esos parámetros y si tiene éxito devuelve al usuario. Se puede borrar un usuario de la base de datos. Además se puede establecer el atributo premium del usuario a 1. Y también se puede obtener un valor de las variables de la clase usuario mediante los getters.

iii. Nutri.php

La clase “`class Nutri`” lleva un funcionamiento similar al de usuario con métodos muy parecidos pero necesarios igualmente variando la tabla de la base de datos a la que se accede que en este caso es profesional. Aquí podemos destacar dos funciones. En primer lugar, `buscaPorMenosUsuarios()` que devolverá en caso de éxito una instancia de nutri correspondiente al nutricionista con menor cantidad de usuarios, este método se emplea para poder asignar un usuario a un nutricionista cuando realiza un pago pero únicamente al nutricionista con menor cantidad de usuarios para no sobrecargar a un solo nutricionista.

iv. Premium.php

Esta clase también es similar a la de usuario con métodos de crea, busca y borra pero con los atributos correspondientes para la tabla de premium.

Rutina.php

En este fichero hay una clase Rutina, “`class Rutina`”, que obtiene Ejercicios de una base de datos, tabla “ejercicios” para crear una rutina para el usuario. Sólo se podrá crear dicha rutina en caso de que el usuario esté logueado.

El usuario introduce el objetivo deseado {Hipertrofia-Fuerza-Resistencia}, su nivel {Principiante-Medio-Avanzado} y el número de días {3-5}. De esa manera, se insertará en el caso de que sea su primera rutina o se actualizará en la tabla “planificacion” en un json según la variable de sesión id del usuario.

La clase rutina es llamada desde FormularioRutinas donde se instancia dicha clase para tomar los campos de los valores introducidos en el formulario que se han detallado anteriormente. Es en FormularioRutinas.php donde se validan y comprueban los datos introducidos por el usuario.

En Rutina.php se usan variables auxiliares como músculos y cont y otras funciones como `fill_array(&$cont, $nveces , &$arrayaux)` que nos permiten usar un algoritmo para crear una rutina para el usuario en función de los campos introducidos anteriormente en el formulario.

v. Dieta.php

El fichero *Dieta.php* tiene una clase “`class Dieta`” que interactúa con las tablas “comida”, “dieta” y “usuario” de la BD. El objetivo de esta clase es allanar el terreno para que se muestre contenido cuando el usuario requiere usar la planificación de dietas.

Esta clase comprueba si existe una entrada en la tabla “dieta” del *usuario logeado*. Si no existe, la crea, y también crea una dieta sacando información de la tabla “comida”. Si existe la entrada, comprueba si el tipo de dieta solicitado es el mismo que el que tenía en la base de datos. Si la dieta es la misma, saca información de la tabla “dieta”, y si no, actualiza la tabla con un nuevo plan de dieta sacando información de la tabla “comida”.

b. Implementación de contenido html

i. ChatEntrenador.php

El fichero *ChatEntrenador.php* tiene una clase “`class ChatEntrenador`” que interactúa con las tablas “profesional”, “chat” y “usuario” de la BD. Esta clase tiene tres funciones: `dataChat()`, `Usuarios()` y `mostrarChat()`.

La primera muestra el área del chat, la segunda hace una consulta para encontrar el usuario en la tabla del profesional y la tercera crea el código HTML para mostrar por pantalla el chat.

ii. ChatUsuario.php

El fichero *ChatUsuario.php* tiene una clase “`class ChatUsuario`” que interactúa con las tablas “profesional”, “chat” y “usuario” de la BD. Esta clase tiene tres funciones: `dataChat()` y `mostrarChat()`.

La primera muestra el área del chat, la segunda crea el código HTML para mostrar por pantalla el chat.

iii. Planificacion.php

El fichero *Planificacion.php* contiene una clase “`class Planificacion`” que conecta con la tabla “planificacion” de manera que coge la rutina del usuario que esté logueado y la muestra por una tabla (`<table>`) almacenando la información en la variable `$html`. Esto se mostrará desde el método `mostrar()`.

Planificación.php es llamado por *tablaRutina.php* y es lo que se muestra en la pantalla después de que el usuario introduzca los datos en *FormularioRutinas.php*, se procesen y se cree la rutina en *Rutina.php*, se llama a *tablaRutina.php* y posteriormente, en *Planificación.php* se muestra la rutina del usuario accediendo a la base de datos.

iv. PlanificacionDietas.php

El fichero *PlanificacionDietas.php* contiene una clase “`class PlanificacionDietas`” que interactúa con la tabla “dieta”. Esta clase tiene la responsabilidad de generar el contenido HTML de la planificación de la dieta. En nuestro caso es una tabla con 7 columnas (L, M, X, J, V, S y D) por 3 filas (desayuno, comida y cena). La constructora se encarga de coger de la información de la base de datos, y el método “`muestra_tabla()`” de construir la tabla con la información de la constructora.

c. Formularios

i. FormularioDietas.php

El fichero *FormularioDietas.php* implementa un formulario mediante la clase “`class FormularioDietas extends Formulario`”. En este formulario se implementan dos funciones heredadas: `generaCamposFormulario()` y `procesaFormulario()`.

La primera función retorna una variable `$html` que genera una lista desplegable en HTML que usa un campo “choose-diet” con los posibles valores: ‘1’, pérdida de peso; ‘2’, aumento de peso; y ‘3’, mantenimiento de peso.

La segunda función asegura que los valores recogidos sean válidos, y si no lo son, no procesa la acción del formulario. En el caso de que todo sea correcto, usa la clase “`class Dieta`” para hacer las modificaciones necesarias en la BD antes de redirigir a la pestaña *planificaciondietas.php*.

ii. FormularioLogin.php

El fichero *FormularioLogin.php* es un formulario que contiene la clase “`class FormularioLogin extends Formulario`”. Tiene dos funciones heredadas `generaCamposFormulario()` y `procesaFormulario()`.

La primera retorna una variable `$html` con el HTML del formulario y la segunda mediante una variable que recibe obtiene información de los campos recibidos por el formulario. Son validados y se comprueba que tengan unos valores correctos.

Los campos de este formulario son los siguientes: alias que hace referencia al alias del usuario que lo identifica y distingue del resto y por último un campo password para comprobar que ese usuario no se está haciendo pasar por otro y pueda acceder a sus datos. Se llama a la función `login()` de ambos (*Usuario.php* y *Nutri.php*) para corroborar que tipo de usuario está accediendo con los valores del formulario. Se guardan en variables de sesión información relevante a ese usuario para evitar accesos posteriores a la base de datos.

iii. FormularioLogros.php

El fichero *FormularioLogros.php* implementa un formulario mediante la clase “`class FormularioLogros extends Formulario`”. En este formulario se implementa una funcione heredada: `mostrarFormulario()` y `Usuarios()`.

La primera función retorna una variable `$html` que genera una lista desplegable en HTML que usa un campo “logrosE” con los posibles valores: ‘5logros’, has completado los 5 logros restantes; ‘AccesoTodos’, consigues acceso a todas las áreas; ‘ComenzarChat’, has comenzado un chat; ‘Completa1Plan’, has completado un plan; ‘Completa5Plan’, has completado 5 planes; ‘ContrataNutri’, has contratado un nutricionista; ‘Foro’, has entrado en el foro; ‘Permanencia’, has iniciado sesión 5 días seguidos; y ‘Permanencialm’, has iniciado sesión un mes seguido.

También asegura que los valores recogidos sean válidos, y si no lo son, no procesa la acción del formulario. En el caso de que todo sea correcto, hace las modificaciones necesarias en la BD antes de redirigir a la pestaña *EntrenadorPersonalEnt.php*

iv. FormularioPago.php

El fichero *FormularioPagos.php* es un formulario que contiene la clase “`class FormularioPagos extends Formulario`”. Implementa dos funciones heredadas `generaCamposFormulario()` y `procesaFormulario()`.

La primera función retorna una variable `$html` que genera unos campos a rellenar en HTML, cada una con un campo: “peso”, “altura”, “alergias”, “observaciones”.

La segunda función asegura que los valores recogidos sean válidos, y si no lo son, no procesa la acción del formulario. En el caso de que todo sea correcto, usa las clases “`class`

`Nutri`”, `class Premium`” y `class Usuario`”, para hacer las modificaciones necesarias en la BD antes de redirigir a la pestaña *EntrenadorPersonalUsu.php*.

- `class Nutri`” se usa para buscar al nutricionista con menos usuarios.
- `class Premium`” se usa para crear una nueva entrada de usuario premium.
- `class Usuario`” se usa para actualizar el bool premium en la entrada del usuario.

v. FormularioRegistro.php

El fichero *FormularioRegistro.php* es un formulario que contiene la clase `class FormularioRegistro extends Formulario`”. Implementa dos funciones heredadas `generaCamposFormulario()` y `procesaFormulario()`.

La primera retorna una variable `$html` con el HTML del formulario y la segunda mediante una variable que recibe obtiene información de los campos recibidos por el formulario. Son validados y se comprueba que tengan unos valores correctos.

Los campos de este formulario son los siguientes: nombre y apellidos del usuario, alias que hace referencia al alias del usuario que lo identifica y distingue del resto, su mail y por último un campo password y password2 para comprobar que se introdujo correctamente la contraseña. Se llama con los valores del formulario a la función `buscaPorAlias()` para comprobar que no hay ningún usuario con ese nombre de usuario y si no lo hay se llama a la función `crea()` de usuario con los datos de usuario. Se guardan en variables de sesión información relevante a ese usuario para evitar accesos posteriores a la base de datos.

vi. FormularioRutinas.php

El fichero *FormularioRutinas.php* es un formulario que contiene una clase `class FormularioRutinas extends Formulario`” y extiende de la clase Formulario. Tiene dos funciones protegidas; `generaCamposFormulario()` y `procesaFormulario()`. La primera retorna una variable `$html` con el html del formulario y la segunda mediante una variable que recibe obtiene información de los campos recibidos por el formulario. Son validados y se comprueba que tengan unos valores correctos.

Los campos de este formulario son los siguientes; objetivo, puede ser ‘1’, fuerza; ‘2’, hipertrofia; o ‘3’, resistencia. Nivel, toma como valores ‘P’, ‘M’ o ‘A’ que hacen referencia a Principiante, Medio o Avanzado. El tercer campo es días que puede valer un 3 o un 5. Se instancia la clase Rutina.php con los valores del formulario y se llama a la función `comprobarRutina()` de dicha clase para que actualice o inserte una nueva rutina en la tabla “planificacion”.

3. Estructura de la base de datos

a. Tablas

i. chat

Nombre	Receptor	Origen	Cocontenido	Tiempo	Tipo
Tipo	varchar	varchar	text	time	varchar
Detalles	nick del receptor	nick del emisor	mensaje	Hora del mensaje	{ E-U, U-E }

ii. comidas

La tabla 'comidas' guarda todas las comidas que se pueden hacer. Las comidas pueden ser para (1) perder peso, (2) ganar peso y (3) mantener peso, y se clasifican en (1) desayuno, (2) comida y (3) cena, representado en los siguientes campos:

Nombre	[PK] id_comida	objetivo	tipo	descripción
Tipo	int	int	enum	text
Detalles	auto incremental	$0 < x < 3$	{ Desayuno, Comida, Cena }	Texto de la comida

iii. dieta

La tabla 'dieta' guarda a los usuarios que hayan solicitado una dieta, con sus desayunos, comidas y cenas.

Nombre	[PK] id_usuario	objetivo	desayunos	comidas	cenas
Tipo	int	int	text	text	text
Detalles	Debe existir en la tabla usuario	$0 < x < 3$	json array de desayunos	json array de comidas	json array de cenas

Esta tabla tiene una clave primaria 'id_usuario', que debe existir en la tabla 'usuarios'. Los demás campos guardan información relativa al tipo de dieta.

iv. ejercicios

Nombre	músculo	nombre
Tipo	enum	varchar

Detalles	{ Hombro, Pierna, Pecho, Triceps, Biceps, Espalda }	tipo de ejercicio a realizar
----------	---	------------------------------

Esta tabla contenida en la base de datos de 'lifety' contiene dos atributos: 'musculo' y 'nombre'. La variable 'musculo' trata del tipo de músculo afectado por un ejercicio. La variable 'nombre' trata el ejercicio en sí. Utilizamos esta tabla para rellenar una estructura compuesta por arrays de arrays, donde cada componente del array principal es un array compuesto por los diversos ejercicios a realizar en el día 'x'. El relleno de dicho array lo podemos encontrar en la función `fill_array(&$cont, $nveces, &$arrayaux)` de Rutinas.php donde se rellena la rutina del usuario accediendo a esta tabla para seleccionar ejercicios.

En la próxima entrega realizaremos un mejor algoritmo que sea capaz de diseñar una tabla de forma más realista a la hora de escoger los ejercicios que componen la rutina.

v. planificacion

Nombre	[PK] id_usuario	eobjetivo	dias	nivel	rutina
Tipo	int	int	int	char	text
Detalles	Debe existir en la tabla usuario	$0 < x < 3$	{ 3, 5 }	{ P, M, A }	json array de ejercicios

La tabla de planificación sirve para guardar las rutinas de los usuarios generadas en Rutina.php y poder acceder a dichas rutinas y mostrarlas para que el usuario las pueda ver. Tiene los siguientes atributos: 'id_usuario', 'rutina', 'eobjetivo', 'dias' y 'nivel'. Sirve para almacenar la información del formulario y que la tenga el nutricionista y además guarda la 'rutina' en un json. La clave primaria es id_usuario. El valor de 'eobjetivo', dias y nivel está definido en *FormularioRutinas.php*.

El atributo 'eobjetivo', puede ser 1, 2 o 3 (Fuerza, Hipertrofia o Resistencia), 'nivel', toma como valores 'P', 'M' o 'A' que hacen referencia a Principiante, Medio o Avanzado. Por otra parte, 'dias' puede valer un 3 o un 5.

El almacenamiento de estos valores además de la rutina es beneficioso para el usuario ya que el nutricionista que interactúe con el usuario podrá ver estos aspectos del usuario ya que se encuentran en esta tabla planificación.

Es importante mencionar que esta parte de la base de datos va a cambiar para la siguiente entrega; la idea consistiría en hacer una tabla rutina que mediante otra tabla contiene que se relaciona con rutina y ejercicios. Respecto a la relación, sería n y n.

La tabla ejercicio seguiría teniendo los atributos id (clave primaria) y nombre. La tabla Rutina tendría un id (clave primaria) para almacenar todas las rutinas que vaya a tener el usuario por si se actualiza, id-usuario y activa (booleano para saber si actualmente la utiliza el usuario). Y la tercera tabla, contiene, que parte de una relación n-n tendría de atributos

id-rutina, id-ejercicio y días para saber que ejercicio va en cada día. Realmente modificaremos el algoritmo para generar rutinas y lo mejoraremos integrando estas nuevas tablas para almacenar la información de esta manera que se ha explicado.

vi. premium

Nombre	[PK] id_usuario	[PK] id_profesional	peso	altura	alergias
Tipo	int	int	float	float	text
Detalles	Debe existir en la tabla usuario	Debe existir en la tabla profesional			

Nombre	[PK] id_usuario	[PK] id_profesional	observaciones	num_logros	logros
Tipo	int	int	text	int	enum
Detalles	Debe existir en la tabla usuario	Debe existir en la tabla profesional			

En esta tabla se almacenan los atributos de los usuarios registrados que han pagado un nutricionista. Contiene las siguientes columnas: `peso`, `altura`, `alergias`, `observaciones_adicionales`, `num_logros`, `logros`, `id_usuario` e `id_profesional`. Se guarda la información relevante para el nutricionista (peso, altura, alergias y observaciones) junto con los logros que ha ido consiguiendo el usuario durante su tiempo como premium. Además se almacena el usuario con el que está relacionado y el profesional encargado de ese usuario.

vii. profesional

En esta tabla se almacenan los atributos de los profesionales registrados. Contiene las siguientes columnas: `nombre`, `apellidos`, `correo`, `password`, `nutri`, `usuarios`, `num_usuarios`, `id_usuario`. Se guardan el nombre y apellidos del usuario además de su correo electrónico que introduce al registrarse, también introduce su nick de nutri. Num_usuarios es un int que representa el número de usuarios que tiene el profesional guardado en usuarios. El id_profesional hace referencia al identificador que se asocia con cada profesional, es un int y sería la clave primaria de la tabla. Respecto a la contraseña, se almacena en la base de datos un hash en esta columna.

viii. usuario

En esta tabla se almacenan los atributos de los usuarios registrados. Contiene las siguientes columnas: `nombre`, `apellidos`, `correo`, `password`, `usuario`, `premium`, `id_usuario`. Se guardan el nombre y apellidos del usuario además de su correo electrónico que introduce al registrarse, también introduce su nick de usuario. Premium es un int que toma valores 0 o 1 y representa si el usuario es premium para acceder a distintos servicios que ofrece la página web. El id_usuario hace referencia al identificador que se

asocia con cada usuario, es un int y sería la clave primaria de la tabla. Respecto a la contraseña, se almacena en la base de datos un hash en esta columna.

b. Relaciones

En las tablas de la base de datos podemos encontrar diversas relaciones entre ellas. La **clave primaria de usuario** se utiliza como clave foránea en premium, planificación, dietas y rutinas. El usuario premium se identifica mediante su clave primaria id_usuario que es foránea a la creada en usuario relacionando ambas tablas. Por otro lado, planificación posee también como clave primaria el id de usuario que relaciona cada fila de planificación con un usuario siendo este id de usuario una foreign key a la clave primaria de usuario.

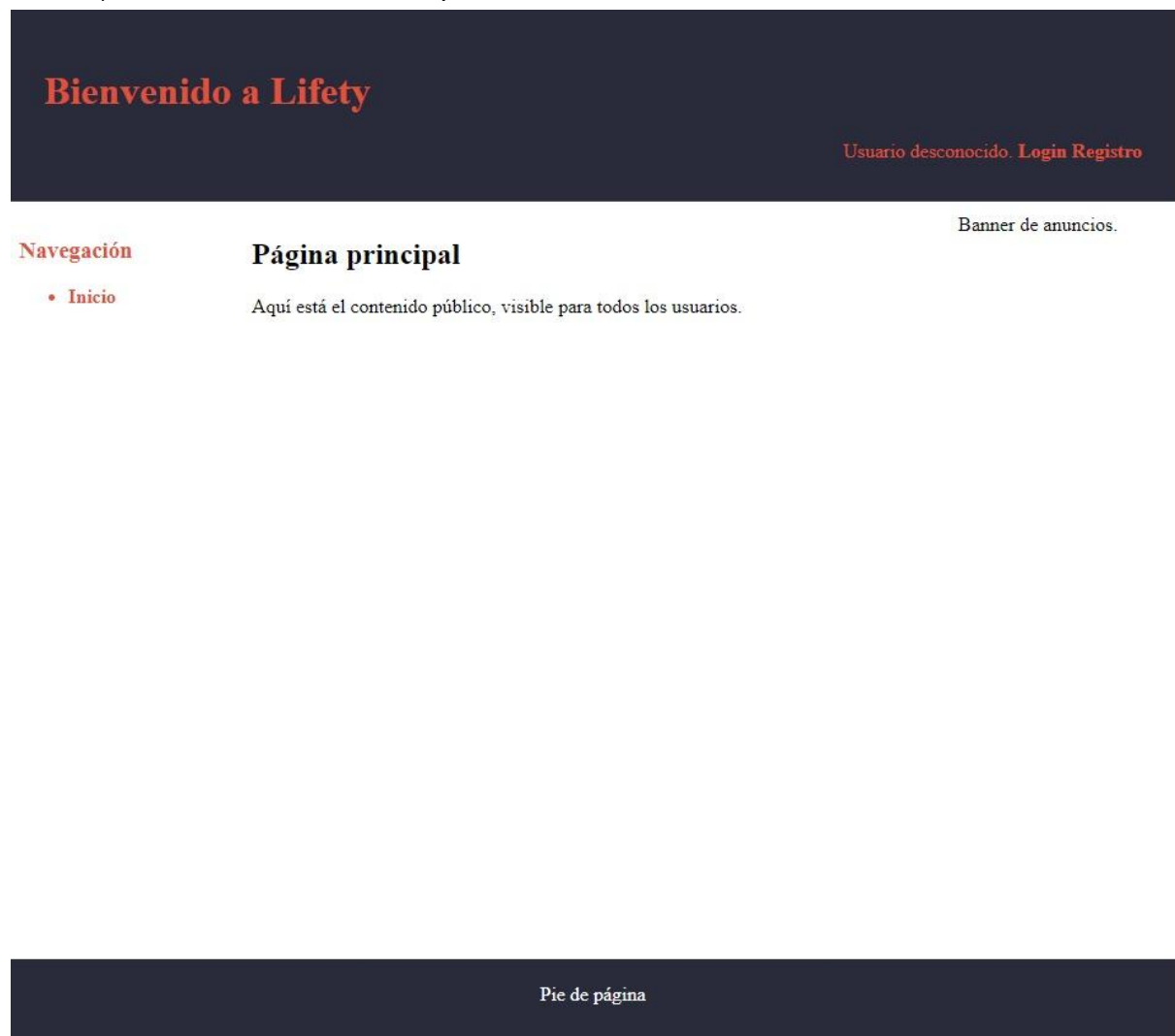
También se encuentra la **clave primaria en profesional** que es id_profesional e identifica inequívocamente a un profesional. En premium se utiliza la columna de id_profesional como foreign key a esta clave permitiendo identificar quien es el profesional para cada usuario premium.

4. Prototipo funcional del proyecto

a. Ejecución

i. Como nuevo usuario

La primera pantalla que se muestra es la página principal de Lifety (index.php), sin estar logueado. En ella se puede observar en la parte superior el nombre de nuestra empresa. En la parte derecha de arriba se muestra la opción de iniciar sesión (Login) o registrarse como nuevo usuario. En la barra lateral izquierda se muestra el menú de navegación, estando solo disponible la opción de Inicio (página donde actualmente se encuentra el nuevo usuario). Y en el lateral derecho aparece el banner de anuncios.



Cuando el usuario selecciona la opción de registrarse, ya que al ser nuevo no tiene ninguna cuenta asignada, se redirige a la página registro.php. En ella aparece un formulario de registro con los siguientes campos: Nombre, Apellidos, Nombre de usuario, Dirección de correo, Password, Reintroduce el password.

Navegación

- Inicio

Registro de usuario

Banner de anuncios.

Datos para el registro

Nombre:

Apellidos:

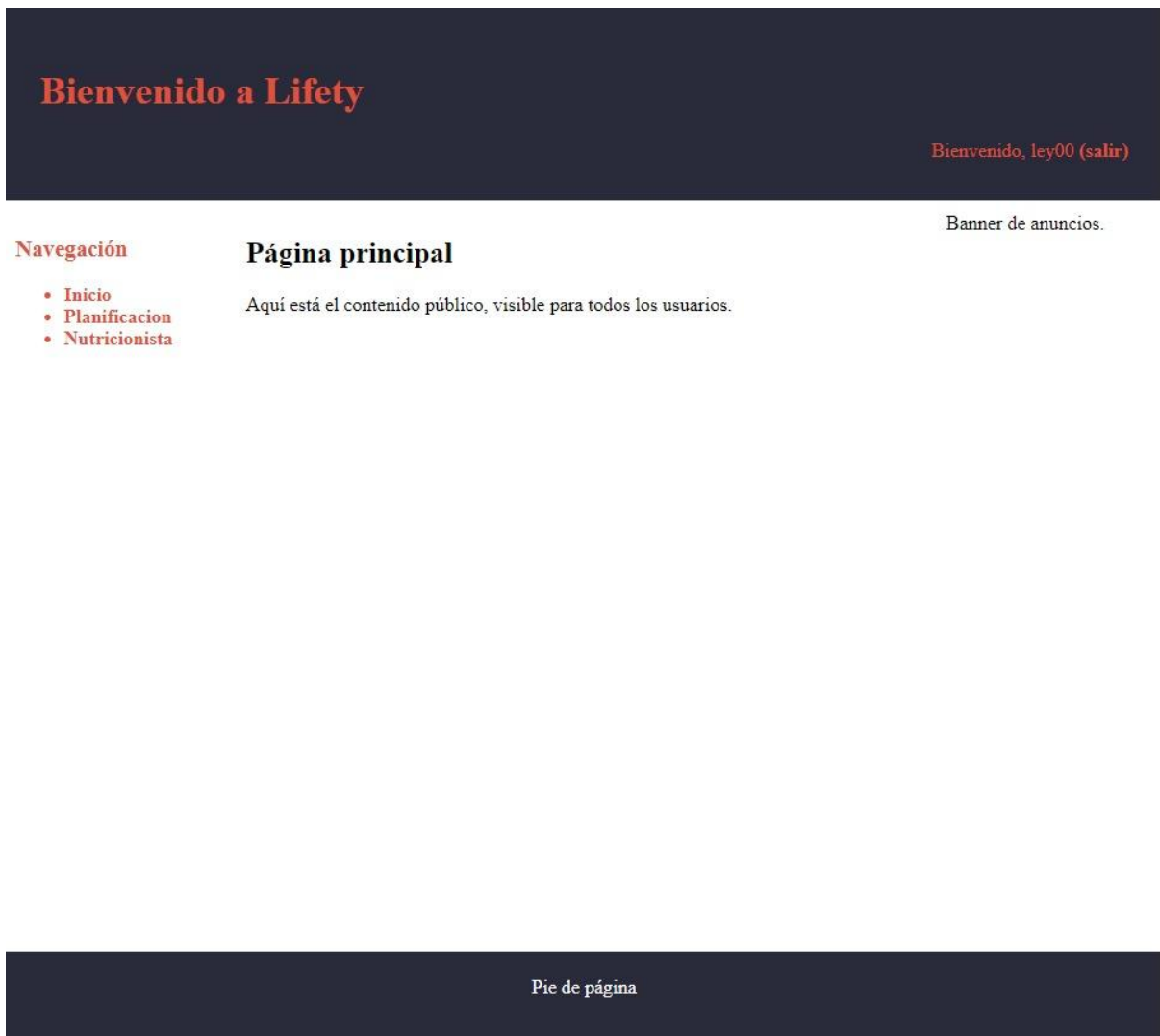
Nombre de usuario:

Dirección de correo:

Password:

Reintroduce el password:

Una vez introducidos todos los datos, comprobando que la contraseña tiene una longitud mínima de 5 caracteres (si esto no ocurre, se vuelve a solicitar introducir una nueva contraseña que cumpla el requisito requerido), al usuario se le redirige a la página principal (index.php). En ella puede observar en el menú de la derecha unas nuevas opciones, por ejemplo, planificación y nutricionista.



ii. Como usuario ya existente

La primera pantalla que se muestra, al igual que si es nuevo usuario, es la página principal de Lifety (index.php), también sin estar logueado. Una vez que selecciona la opción de login, se muestra la pantalla login.php, en la cual se visualiza un formulario para acceder al

sistema. Se deben introducir los siguientes datos: Usuario y password.

Bienvenido a Lifety

Usuario desconocido. [Login](#) [Registro](#)

Navegación

- Inicio

Acceso al sistema

Usuario y contraseña

Usuario:

Password:

Entrar

Banner de anuncios.

Pie de página

Si alguno de los datos introducidos no coinciden con los guardados en la base de datos, salta el mensaje “El usuario o password no coinciden”. En caso contrario, te redirige a la página principal, teniendo como opciones de página las mismas descritas en nuevo usuario (planificación y nutricionista).

Bienvenido a Lifety

Bienvenido, ley00 (salir)

Navegación

- Inicio
- Planificación
- Nutricionista

Página principal

Aquí está el contenido público, visible para todos los usuarios.

Banner de anuncios.

Pie de página

Si se selecciona la página de Planificación te redirige a la página de *rutinas_dietas.php*. En ella se muestran dos formularios.

Navegación

- Inicio
- Planificación
- Nutricionista

Acceso al sistema

Banner de anuncios.

Rutinas

Selecciona tu nivel:

- ☒ Principiante
☐ Medio
☐ Avanzado

Selecciona el número de días:

3 Días ▼

Selecciona tu objetivo de entrenamiento:

Fuerza ▼

La actividad física regular puede mejorar la fuerza muscular y aumentar la resistencia. El ejercicio suministra oxígeno y nutrientes a los tejidos y ayuda a que el sistema cardiovascular funcione de manera más eficiente. Y cuando tu salud cardíaca y pulmonar mejora, tienes más energía para hacer las tareas diarias.

Quiero esta rutina

Dietas

Selecciona tu dieta:

Pérdida de peso ▼

Según la mayoría de los dietistas, no existen alimentos "malos", sino dietas "poco sanas". Una dieta saludable se consigue comiendo la cantidad correcta de alimentos en la proporción adecuada, con continuidad. Comer una proporción adecuada de alimentos de los principales grupos constituye la base del bienestar cotidiano, y reducirá el riesgo de enfermedades a largo plazo.

Quiero esta dieta

En primer lugar, el formulario de rutinas en el cual se debe indicar el nivel de intensidad (principiante, medio y avanzado), el número de días que se desea realizar la rutina (es un menú desplegable que te da la opción de indicar o 3 o 5 días), el objetivo de entrenamiento (fuerza, hipertrofia y resistencia). Además, antes de indicar que se desea esa rutina, se muestra una breve descripción de lo que es capaz de conseguir la actividad física. Una vez seleccionada la información se muestra por pantalla la página de `tablarutina.php` que consta de una tabla indicando los días que se debe realizar el entrenamiento, con su respectivos ejercicios y el número de repeticiones y series que se debe realizar, dependiendo el primero de éstos y del objetivo escogido por el usuario. A continuación, se muestra un ejemplo de rutina: principiante, 5 días e hipertrofia.

Navegación

- Inicio
- Planificación
- Nutricionista

Rutina de entrenamiento

Banner de anuncios.

Rutina de entrenamiento:

Día 1	Día 2	Día 3	Día 4	Día 5
Press banca	Jalon	Sentadilla	Press banca	Curl spider
Aperturas con mancuernas	Remo en T	Prensa	Aperturas con mancuernas	Predicador
Press banca inclinado	Remo con barra	Extension de cuadriceps	Press banca inclinado	Martillo
Maquina de empuje	Renegade row	Hip thrust	Maquina de empuje	Chin-ups
Elevacion lateral	Curl spider	Fondos	Elevacion lateral	Sentadilla
Press hombro	Predicador	Press frances	Press hombro	Prensa
Remo al menton	Martillo	Extensiones	Remo al menton	Extension de cuadriceps
Press militar	Chin-ups	Barras paralelas	Press militar	Hip thrust
			Jalon	Fondos
			Remo en T	Press frances
			Remo con barra	Extensiones
			Renegade row	Barras paralelas

Nº de repeticiones = 16. Nº de series: 3

Pie de página

Y en segundo lugar, el formulario para seleccionar la dieta acorde a tus preferencias, indicando el objetivo que se desea lograr (perder peso, ganar peso o mantener peso). Al igual que en el formulario de rutina, en éste después de seleccionar el objetivo también aparece un breve resumen de la importancia de realizar una dieta saludable. A continuación, se muestra un ejemplo de una dieta para ganar peso.

Bienvenido a Lifty

Bienvenido, Alexin (salir)

Navegación

- Inicio
- Planificación
- Nutricionista

Dietas

Planificación de tu dieta:

	L	M	X	J	V	S	D
Desayuno	Gofres proteicos	Sandwich de queso gouda con huevo duro	Sandwich de queso gouda con huevo duro	Brownie proteico	Requeson con fruta	Requeson con fruta	Porridge de avena y frutos secos
Comida	Pasta con setas salteadas	Revuelto de gambas y champiñones	Berenjena y calabacín asado con pimentón dulce	Cocido	Arroz integral con pasas y almendras	Hamburguesa de pollo zingy	Cocido
Cena	Minipizza cuatro quesos con naranja troceada	Macarrones con pisto	Hamburguesa de jamón york con mayonesa y queso	Minipizza cuatro quesos con naranja troceada	Minipizza cuatro quesos con naranja troceada	Filete de buey con patata al horno y verduras	Tagliatelle con salmón y nata

Banner de anuncios.

Pie de página

Si se selecciona la página de Nutricionista te lleva a *registronutri.php*. En ella se puede visualizar un formulario para introducir los siguientes datos: peso, altura (estos dos campos son obligatorios) y alergias y observaciones adicionales (estos campos son obligatorios), junto con una imagen que representa el estilo de vida saludable.

Navegación

- Inicio
- Planificacion
- Nutricionista

Contratacion de nutricionista

Banner de anuncios.

Por favor, introduzca sus datos:

Introduzca su peso:

Introduzca su altura:

Si tiene alguna alergia rellene este campo:

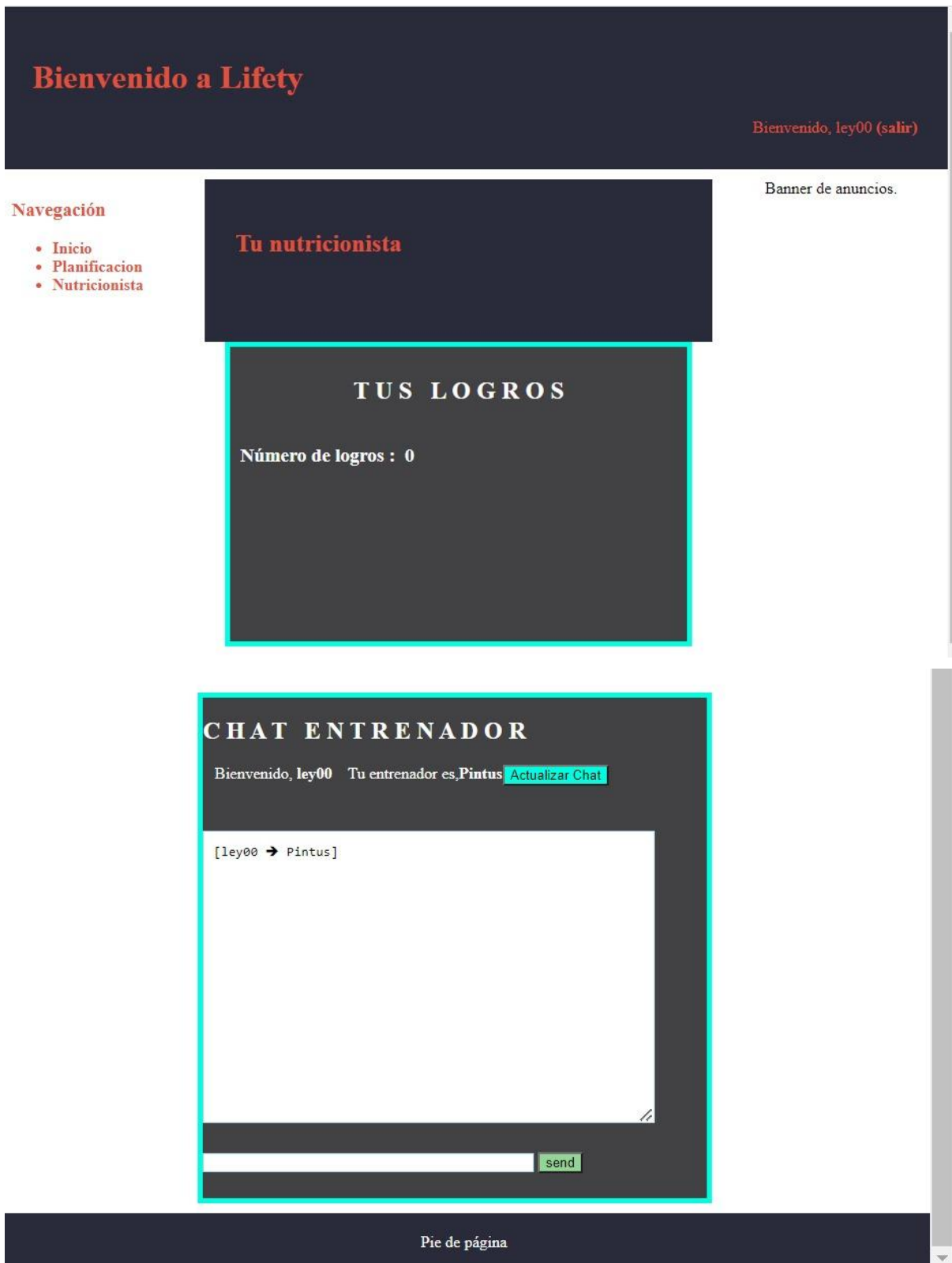
Alguna observacion adicional:



Una vez que se da la orden de pagar y se procesa correctamente el pago con la entidad bancaria, el usuario pasa a ser premium.

iii. Como usuario ya existente premium

Teniendo en cuenta el apartado de usuario no premium, una vez que contrata el servicio, pasa a ser usuario premium. La página te redirige a una pantalla que muestra información personal de cada usuario.



A medida que el usuario va logrando los retos u objetivos que le propone el nutricionista, en su pantalla de nutricionista le aparecerá un número con la cantidad total de logros que ha conseguido, junto con un pequeño logo en representación de la actividad llevada a cabo. Además, tendrá a su disposición un chat privado con su entrenador para poder preguntar todas aquellas dudas que le surjan durante el entrenamiento o dieta

iv. Como entrenador

Por último, si se entra con las credenciales de entrenador, la página principal la ve exactamente igual que los demás usuarios. La diferencia es que en la página de nutricionista en vez de visualizar el cuadro con los logros alcanzados, se muestra por pantalla dos desplegables con la posibilidad de añadir o quitar logros al usuario al que se quiera proceder a realizar la acción. Además, a través de un desplegable, el entrenador tiene la opción de comunicarse a través del chat con el usuario que escoja.

LOGROS

Seleccione el Logro :
Completar 5 Logros ▼

Seleccione al Usuario :
Floren ▼

Quitar Logro

Añadir Logro

CHAT CON USUARIO

Bienvenido, Pintus

Elige usuario: Floren ▼

Actualizar Chat

Debes Actualizar Chat para ver la información

D: Floren ▼

Send

b. Datos de usuarios para la prueba

Usuario	Contraseña	Premium
Floren	12345	Si
P\$G	12345	No

c. Errores conocidos

- Inconsistencias entre los nombres.
- Clase entidad no implementada.
- Relaciones entre tablas y planteamiento de estas erróneo.
 - Datos como JSON arrays
 - Datos separados con ','

5. Planteamiento para el resto de funcionalidades

De la funcionalidad banner sería necesaria una tabla de la base de datos que contenga información en cada fila de los elementos necesarios para que haya un banner como sería por ejemplo el id del banner la imagen que lo representa y el enlace donde se redirige y en esa tabla banner o anuncios es donde se recogerá la información del banner. Para se muestre se tendría que realizar algún PHP relacionado con la vista en dónde se realice una HTML que muestra la imagen y que pueda reconducir a otra página a través de un link. Esto se adecuará para que funcionará para que estuviera a la derecha el banner, a la izquierda estaría el menú y en el centro la pantalla principal de contenido que representaría cualquier cosa.

Contenido se crearía una tabla donde se puede a encontrar todo el tipo de contenido, entonces la tabla sería contenido y tendría atributos como tipo de contenido (un int para ver si es un video, texto...) identificador de contenido y otros atributos secundarios como fecha, descripción... De esta manera tendríamos una fila por cada entrada de contenido y esa entrada podría ser de distinto tipo ya sea un video, un texto de una receta, de algo de entrenamiento, etc. Se mostraría mediante una carga de la bbdd.

Respecto a la recomendación de suplementos y a su funcionalidad, a la base de datos realizaremos una tabla de suplementos que se relacionen con la de usuario y recomienda cosas en base a lo que el usuario quiera es decir se guardarán en una tabla todos los suplementos disponibles y mediante una relación de interés entre la tabla de suplementos y la de usuario se podrían recomendar ciertos suplementos que interesan a los usuarios según atributos. Esto se haría mediante 3 tablas; usuario, suplementos y recomendar. La tabla de recomendar tendría una clave de el id del usuario y del id del suplemento para relacionar según los atributos que determinemos qué suplementos convienen o les interesa a los usuarios.

En cuanto a la funcionalidad de foros, necesitaríamos un formulario para la creación de un tema ya que los foros serán gestión de temas, cada tema sería un foro distinto. Se realizará un formulario para la creación de un tema y esa información sería guardada en formulario tema que crearía una nueva clase tema.php insertando la información en la base de datos en una tabla tema que contenga parámetros como el id del creador, id tema, título, descripción, fecha... Mediante una relación contenida se establecerán comentarios en cada tema y se realizaría un mecanismo de relación similar a otras funcionalidades. Contiene almacenaría que comentarios están en cada tema y se mostrarían según la fecha de introducción del comentario y los temas se harían de igual manera. Para el almacenamiento en la bbdd de comentarios sería de una manera similar a los temas o al chat que hemos implementado en las funcionalidades.