# **PRÁCTICA 2.**

# **ARQUITECTURA Y PROTOTIPO FUNCIONAL DEL PROYECTO**

## **Nombres**

* Emilio José Valencia Calvopiña.
* Manuel Espinosa Guerra.
* Geraldyn Carrero Azuaje.
* Cheng Jun Liu Zheng.
* Miriam Elizabeth Cabana Ramírez.
* Jaime Madriñán Fernández.
* Mingyang Chen.

## **Descripción general**

Tal como indicamos en la práctica anterior, Hércules es una herramienta perfecta para aquellas personas que quieren llevar un control sobre su alimentación y sus deportes, pudiendo mantener un historial de estos y, además, recibiendo recomendaciones de sus entrenadores.

Así mismo, en la práctica 1 incluíamos la posibilidad de una tienda online. Siguiendo las recomendaciones del profesor, vamos a obviar de momento este apartado.



**A continuación, antes de describir nuestra aplicación web, presentamos algunos aspectos importantes:**

### **Datos de acceso a la base de datos**

**BD principal:** mysql/hercules.sql

**Nombre de la BD:** hercules

**Datos usuario administrador para ser importados:** mysql/usuarios.sql

**Conexión con BD:** Se realiza en el DAO principal.

|  |  |  |  |
| --- | --- | --- | --- |
| Perfil de Usuario | Nombre | Nif | Contraseña |
| Administrador | Hercules |  | iG8hC62acnPrvIeU |
| Entrenador | PETER PARKER | 12345678A | 12345678 |
| Entrenador | TONY STARK | 12345678B | 12345678 |
| Cliente | MIKE | 12345678C | 12345678 |
| Cliente | HUGAN | 12345678D | 12345678 |

En esta versión, un entrenador no puede ser cliente, solo puede ser entrenador (se indica al momento de registrarse).

La BD también incluye 4 ejercicios en la tabla **ejercicio**, y 5 alimentos en la tabla **alimento.**

### **Estructura de carpetas**

A continuación, se aprecia la estructura de carpetas.

**hercules**

**│** buzon.php

│ comunidad.php

│ contacto.php

│ contenido.php

│ eliminarEntrenador.php

│ entrenadores.php

│ entrena\_check.php

│ faqs.php

│ index.php

│ InsertaComida.php

│ login.php

│ logout.php

│ miPerfil.php

│ miPerfilCabecera.php

│ miPerfilComidas.php

│ miPerfilEntrenamientos.php

│ miPerfilMisClientes.php

│ miPerfilMisEntrenadores.php

│ perfil\_Cliente.php

│ perfil\_Entrenador.php

│ quienes\_somos.php

│ registrarEntrenamiento.php

│ registro.php

│ RegistroComida.php

**│**

**├───.settings**

**│** org.eclipse.php.core.prefs

│ org.eclipse.wst.common.project.facet.core.xml

**│**

**├───includes**

**│ │** aplicacion.php

│ │ config.php

│ │ controller.php

│ │ estilo.css

**│ │**

**│ ├───comun**

**│ │** cabecera.php

│ │ pie.php

**│ │**

**│ ├───DAOs**

**│ │** alimentoDAO.php

│ │ comentariosDAO.php

│ │ comidaDAO.php

│ │ DAO.php

│ │ ejercicioDAO.php

│ │ entrenamientoDAO.php

│ │ recomendacionesDAO.php

│ │ usuarioDAO.php

**│ │**

**│ ├───Forms**

**│** │ Form.php

│ │ FormularioLogin.php

│ │ FormularioRegistrarEntrenamiento.php

│ │ FormularioRegistro.php

**│ │**

**│ ├───img**

**│ │ │** foto\_inicio.jpg

│ │ │ Hercules\_logo.png

│ │ │ sentadillas.png

**│ │ │**

**│ │ └───usuarios**

**│ └───TOs**

**│** alimentoTO.php

│ comentarioTO.php

│ comidaTO.php

│ ejercicioTO.php

│ entrenamientoTO.php

│ recomendacionesTO.php

│ TOUsuario.php

**│**

**└───mysql**

hercules.sql

usuarios.sql

### **Funcionalidades práctica 2**

Además del registro e inicio de sesión del usuario, las funcionalidades que implementamos en esta práctica son:

1. **Del cliente al entrenador:**
   1. **Solicitar un entrenador:** Esta solicitud se realiza accediendo a la pestaña **“Nuestros entrenadores”** para comprobar los entrenadores disponibles. Se explicará con más detalle en los apartados 3.2 y 3.3.
   2. **Registrar una comida:** Esta solicitud se realiza accediendo a la pestaña **“Mi perfil”** y luego **“Comidas”**. Además de registrar las comidas, se podrán ver las comidas registradas hasta el momento. Se explicará con más detalle en el apartado 3.3
   3. **Eliminar entrenadores:** Se puede eliminar un entrenador en la pestaña **“Mi Perfil” -> “Mis entrenadores” -> “Mostrar Perfil”** al pulsar en el enlace “Eliminar de la lista”.
   4. **Ver entrenamientos:** En la pestaña **“Mi Perfil” -> “Mis entrenadores” -> “Mostrar Perfil”** también se pueden ver los entrenamientos que ha propuesto dicho entrenador.
2. **Del entrenador al cliente:**
   1. **Proponer un entrenamiento:** Esta propuesta se realiza en la pestaña **“Mi perfil”** y luego **“Mis clientes”** (ya veremos que el perfil del entrenador contiene pestañas distintas del cliente). Se explicará con más detalle en el apartado 3.3

## **Listado de scripts para las vistas**

En este apartado se explicarán cada uno de los scripts utilizados para implementar las vistas de aplicación web.

### **INICIO**

**Ruta**: hercules/index.php

Este script se encarga de cargar la página inicial de la aplicación web. El contenido que se muestra está disponible para cualquier usuario (registrado o no) que acceda a la aplicación.

### **NUESTROS ENTRENADORES**

**Ruta**: hercules/entrenadores.php

Con este script se mostrarán los entrenadores que estén disponibles para todos los usuarios (registrados o no). Solo un usuario registrado que **sea cliente** podrá enviar solicitudes a los entrenadores listados. Si no se es cliente mostrará un aviso informativo para indicar al usuario que debe registrarse.

Respecto a las solicitudes, aparte de poder solicitar a un entrenador, se podrá ver si se ha enviado una solicitud a un entrenador o no (“solicitud enviada”), y además, si ese entrenador la ha aceptado “Ya soy tu entrenador/a”.

Este script esta apoyado con otro script (formulario) para procesar datos (**entrena\_check.php**)

Esta ventana es accesible desde el menú principal.

### **MI PERFIL**

**Ruta**: hercules/miPerfil.php

Este script muestra el perfil de un usuario registrado, ya sea entrenador o cliente. En esta vista aparecerán un buzón de solicitudes, si eres entrenador, en el que se podrá aceptar o rechazar las solicitudes de clientes y un resumen de tus datos. Esta vista corresponde a los scripts **perfil\_Cliente.php y perfil\_Entrenador.php.**

También aparecerá un submenú con las opciones comida y mis entrenadores o mis clientes, dependiendo de si has entrado como cliente o entrenador.

**Para los clientes:**

* La pestaña comida llevará al usuario al editor de dietas, que podrá registrar lo que ha desayunado/comido o cenado. Existirá un listado de alimentos para seleccionar uno entre los posibles, y podrán añadirse, de momento, 3 alimentos, que se corresponden con el primer plato, el segundo plato y el postre. También se puede ver todas las comidas que haya registrado el usuario. Estas comidas aparecerán en un cuadro de texto según se vayan insertando. Para registrar una comida se pasará por los siguientes scripts:

**miPerfilComidas.php -> RegistroComida.php -> InsertaComida.php**

* La pestaña mis entrenadores muestra los entrenadores de un cliente. Además, a ver el perfil del entrenador, es posible visualizar los entrenamientos propuestos por dicho entrenador. Si hay un entrenador que todavía no ha aceptado una solicitud, es posible verlo. También dentro del perfil del entrenador hay un enlace para eliminar al entrenador de la lista, el enlace redirige a la página de procesamiento **eliminarEntrenador.php.**

**Para los entrenadores:**

* En la pestaña mis clientes se muestra una lista con los clientes que han enviado la solicitud. Cada uno tendrá un perfil resumido al que se puede acceder con un enlace. Dentro de este también habrá otro enlace que llevará a la tabla de entrenamientos propuesta.
* El enlace para proponer entrenamientos redirige a **registrarEntrenamiento.php** conteniendo un formulario hecho con la clase del archivo **FormularioRegistrarEntrenamiento.php.**

Las vistas de estas dos explicaciones (Clientes y entrenadores) corresponden a los scripts: **miPerfilMisClientes.php y miPerfilMisEntrenadores.php**

Sucesión de scripts a partir de **MiPerfil.php:**

**MiPerfil.php ->**

**miPerfilMisClientes.php ->**

**perfil\_Entrenador ->**

miPerfilEntrenamientos.php

eliminarEntrenador.php

**miPerfilMisEntrenadores.php ->**

**perfil\_Cliente ->**

registrarEntrenamiento.php

Esta ventana es accesible desde el menú principal.

### **QUIENES SOMOS**

**Ruta**: hercules/quienesSomos.php

Este script contiene la información relacionada al equipo que trabaja en la aplicación. Es informativa. De momento no posee datos disponibles.

### **FAQS**

**Ruta**: hercules/faqs.php

Este script es la pestaña de preguntas frecuentes de nuestra aplicación web. Es informativo. De momento no posee información disponible.

### **CONTACTO**

**Ruta**: hercules/contacto.php

Este script es la pestaña de contacto de nuestra aplicación web. Es informativo. De momento no posee información disponible.

### **LOGIN**

**Ruta**: hercules/login.php

Pagina de login con un formulario simple para entrar como usuario. Se pide el NIF/NIE y la contraseña de un usuario. Esta página hace uso de **FormularioLogin.php.**

* FormularioLogin.php: Clase FormularioLogin que hereda de Form. Contiene la estructura del formulario y la lógica del procesamiento de los datos a rellenar.

### **LOGOUT**

**Ruta**: hercules/logout.php

La página que aparecerá al cerrar sesión. Simplemente muestra un mensaje al cliente confirmando que ha cerrado la sesión.

Aquí se destruye la sesión del usuario.

Accesible mediante el botón logout una vez iniciada la sesión.

### **REGISTRATE**

**Ruta**: hercules/registro.php

Pagina de registro con un formulario simple. Se pide el NIF/NIE, nombre, email y contraseña para registrarse. Si se quiere registrar como entrenador se tiene que marcar la casilla y además habrá que rellenar titulación, especialidad, experiencia. Esta página hace uso de **FormularioRegistro.php.** Una vez registrado se entrara a la página con la cuenta que se acaba de crear.

* FormularioRegistro.php: Clase FormularioRegistro que hereda de Form. Contiene la estructura del formulario y la lógica del procesamiento de los datos a rellenar.

### **CABECERA**

**Ruta**: hercules/includes/comun/cabecera.php

Este script contiene la cabecera, común a las distintas pestañas de la aplicación web. En dicho script se incluye el menú de acceso/navegación por las pestañas disponibles.

### **PIE DE PÁGINA**

**Ruta**: hercules/includes/comun/pie.php

Este script contiene el pie de página, común a las distintas pestañas de la aplicación web.

## **LISTADO DE SCRIPTS ADICIONALES**

En este apartado describiremos los scripts relacionados con la lógica de la aplicación, algunos de ellos abstraen el manejo de la base de datos del sistema. Comenzaremos por las capas a más bajo **nivel, hasta acercarnos lo mayor posible a la vista.**

### **TRANSFER OBJECT**

Cada transfer se encuentra en la ruta: hercules/includes/TOs

#### **alimentoTO**

Se encarga de almacenar los alimentos que estarán disponibles para el usuario. Se relaciona con comidaTO a través de una tabla intermedia de la base de datos, llamada alimentocomida. Sus campos corresponden a los de la tabla **Alimento** de la base de datos. Además, se implementan las operaciones get y set de cada uno de estos campos. En principio, ningún usuario tiene acceso a dicho transfer, solo quien posea privilegios de administrador puede interactuar con este transfer.

#### **comidaTO**

Este transfer contiene el registro de las comidas que un usuario ha añadido. Sus campos coinciden con los de la tabla **Comida**, correspondiente a la base de datos. Las operaciones de get y set están implementadas. El campo fecha nos permitirá más adelante mostrar un histórico de las comidas de un usuario, así como listarlo mediante el campo “tipo”, que podrá ser desayuno, comida o cena.

#### **ejercicioTO**

Se encarga de almacenar los alimentos que estarán disponibles para que el usuario. Se relaciona con ejercicioTO a través de una tabla intermedia de la base de datos. Sus campos corresponden a los de la tabla **Ejercicio** de la base de datos. Además, se implementan las operaciones get y set de cada uno de estos campos. En principio, ningún usuario tiene acceso a dicho transfer, solo quien posea privilegios de administrador puede interactuar con este transfer.

#### **entrenamientoTO**

Este transfer contiene el registro de los entrenamientos que un usuario ha realizado. Sus campos coinciden con los de la tabla **Entrenamiento** correspondiente a la base de datos. Las operaciones de get y set están implementadas.

#### **recomendacionesTO**

Este transfer contiene el registro de las recomendaciones (consejos) que un entrenador ha hecho a un cliente. Sus campos coinciden con los de la tabla **Recomendación** correspondiente a la base de datos. Las operaciones de get y set están implementadas.

Para esta práctica no estamos haciendo uso de este TO.

#### **comentariosTO**

Este transfer contiene el registro de los comentarios (valoraciones) que un cliente ha hecho a su entrenador. Sus campos coinciden con los de la tabla **Comentarios** correspondiente a la base de datos. Las operaciones de get y set están implementadas.

#### **usuarioTO**

Este transfer contiene toda la información relacionada con un usuario, que puede ser del tipo cliente (0) o entrenador (1). En caso de ser de tipo entrenador, los campos titulación, especialidad y experiencia están rellenas. Sus campos coinciden con los de la tabla **Usuarios** correspondiente a la base de datos. Las operaciones de get y set están implementadas.

### **DATA ACCESS OBJECT**

Cada DAO se encuentra en la ruta: hercules/includes/DAOs

#### **DAO**

Esta clase DAO aloja la conexión con la base de datos y de momento cuenta también con el método para realizar consultas. De esta clase extienden los distintos DAOs que se explicarán en los próximos apartados.

#### **alimentoDAO**

Esta clase aloja las operaciones CRUD relacionadas con los alimentos. En principio, los usuarios solo podrán ver los alimentos que hay en la base de datos. En cuanto al resto de operaciones CRUD, solo un usuario con privilegios de administrador podrá acceder a ellas.

#### **comidaDAO**

Esta clase aloja las operaciones CRUD relacionadas con las comidas. Además, se añaden algunos métodos más (sobre todo consultas), para hacer un uso más especializado y ofrecer más beneficios a los usuarios.

#### **entrenamientoDAO**

Esta clase aloja las operaciones CRUD relacionadas con los entrenamientos. Además, se añaden algunos métodos más (sobre todo consultas), para hacer un uso más especializado y ofrecer más beneficios a los usuarios.

#### **ejercicioDAO**

Esta clase aloja las operaciones CRUD relacionadas con los ejercicios. En principio, ningún usuario accede a estos datos, a excepción de un usuario con privilegios de administrador.

#### **recomendacionesDAO**

Esta clase aloja las operaciones CRUD relacionadas con las recomendaciones que realizan los entrenadores a sus clientes.

#### **comentariosDAO**

Esta clase aloja las operaciones CRUD relacionadas con los comentarios que realizan los clientes a sus entrenadores.

#### **usuarioDAO**

Esta clase aloja las operaciones CRUD relacionadas con los usuarios. Además, se añaden los métodos afines al login del usuario, así como algunos métodos (sobre todo consultas) para listar entrenadores, clientes de un entrenador, e implementar la funcionalidad de **Solicitar un entrenador.**

### **CONTROLLER**

Se encuentra en la ruta: hercules/includes/controller.php

Como ya sabemos, el Controller actúa como intermediario entre la vista y las operaciones a más bajo nivel, se relaciona directamente con los DAOs. Por tanto, el Controller se encarga de instanciar cada uno de los DAOs y de invocar las distintas funciones de los DAOs.

Su estructura actual está comprendida en: funciones relacionadas con el usuario, funciones de comidas y funciones de entrenamientos.

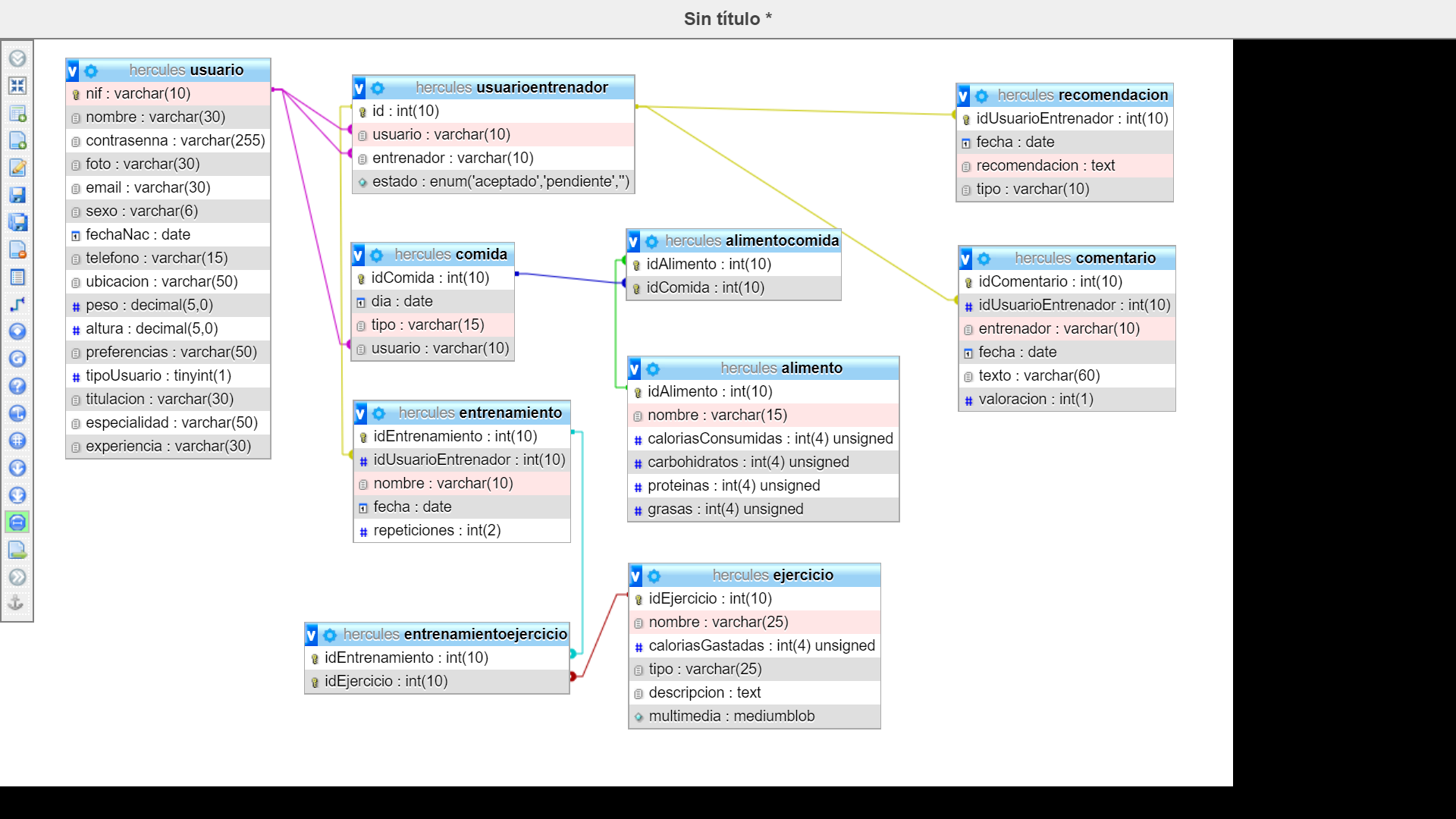
### **CONFIG**

En esta clase definimos los parámetros de conexión a la base de datos, y también, instanciamos el controller de la aplicación.

## **ESTRUCTURA DE LA BASE DE DATOS**

En el apartado 2.1 hemos especificado los datos de acceso a la BD. Ya sabemos su nombre: **hercules.sql**. A continuación, describimos cada una de las tablas que la componen:

Imagen con BD modelo entidad relación



### **Usuario**

Esta tabla almacena los usuarios registrados en nuestra aplicación web. En ella se encuentran los usuarios del tipo: Cliente (0) y Entrenador (1), cuyo valor se especifica en el campo **tipoUsuario**. Todo usuario debe rellenar los campos del 1 al 12, aunque para esta primera versión solo se rellenan los campos esenciales: nif, nombre, contraseña, email. Los demás se incluirán en la próxima versión y nos permitirán realizar distintas consultas o mejorar a la aplicación a partir de los datos obtenidos.

Además, si el usuario es de tipo Entrenador, deberá rellenar la información referente a los campos 14,15 y 16, es decir, titulación, especialidad y experiencia.

Es una tabla muy importante ya que se relaciona con gran parte del resto de las tablas para poder implementar las distintas funcionales de la aplicación. Su clave primaria es el nif.



**Scripts que acceden a la tabla:**

* **usuarioDAO.php:** Consulta.
* **Login.php:** Consulta.
* **Registro.php:** Consulta y modificación.

### **Usuarioentrenador**

Esta tabla intermedia tiene el propósito de relacionar un usuario del tipo Cliente (0) con un usuario del tipo Entrenador (1). Contiene un id incremental y, además, un campo **estado** que puede ser “aceptado” o “pendiente”, para almacenar cuáles son los clientes que entrena un entrenador, y viceversa.



**Scripts que acceden a la tabla:**

* **usuarioDAO.php:** consulta y modificación. Es posible modificar el estado.
* **Entrenadores.php:** Consulta y modificación.
* **Perfil\_Entrenador -> eliminarEntrenador:** Modifica**.**

### **Ejercicio**

La tabla ejercicio, es una tabla que aloja los distintos ejercicios que estarán disponibles para el usuario. El usuario tendrá acceso a estos ejercicios a través de la tabla intermedia **entrenamientoejercicio**, a la que accederá cuando quiera registrar/modificar/eliminar un entrenamiento. Esta tabla solo puede modificarla el usuario con privilegios de administrador.

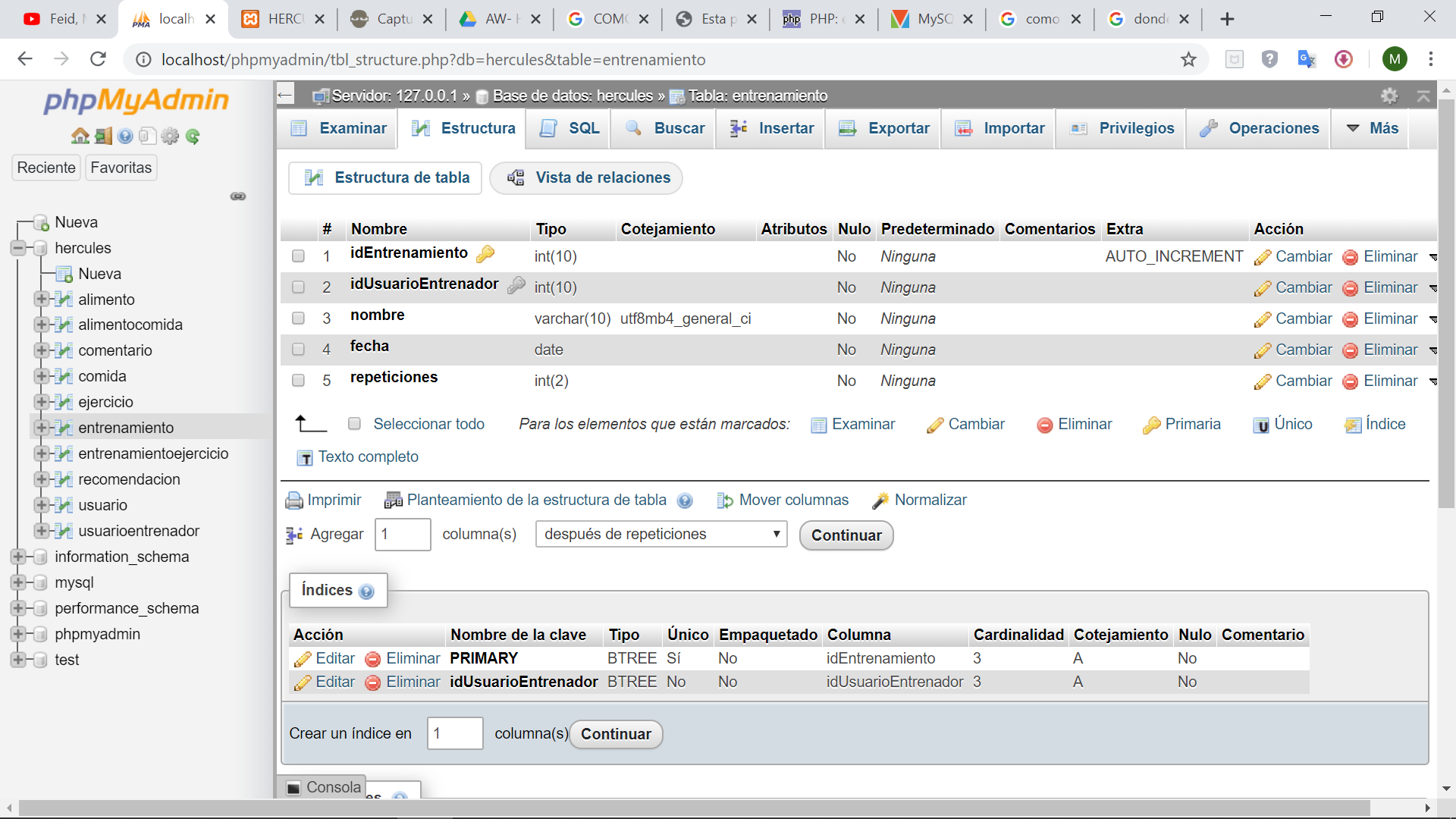


**Scripts que acceden a la tabla:**

* **ejercicioDAO.php:** Consulta.

### **Entrenamiento**

La tabla entrenamiento contiene los distintos entrenamientos que el usuario entrenador ha registrado para un cliente seleccionado. El idEntrenamiento se autoincrementa cada vez que se registra un nuevo entrenamiento, además, de que este campo se utiliza para la tabla intermedia **entrenamientoejercicio** para asociar la información adicional de los ejercicios seleccionados. El campo nombre es el nombre del entrenamiento. El campo fecha es la fecha prevista para realizar el entrenamiento. El campo repeticiones nos indican el número de repeticiones a realizar en cada ejercicio.

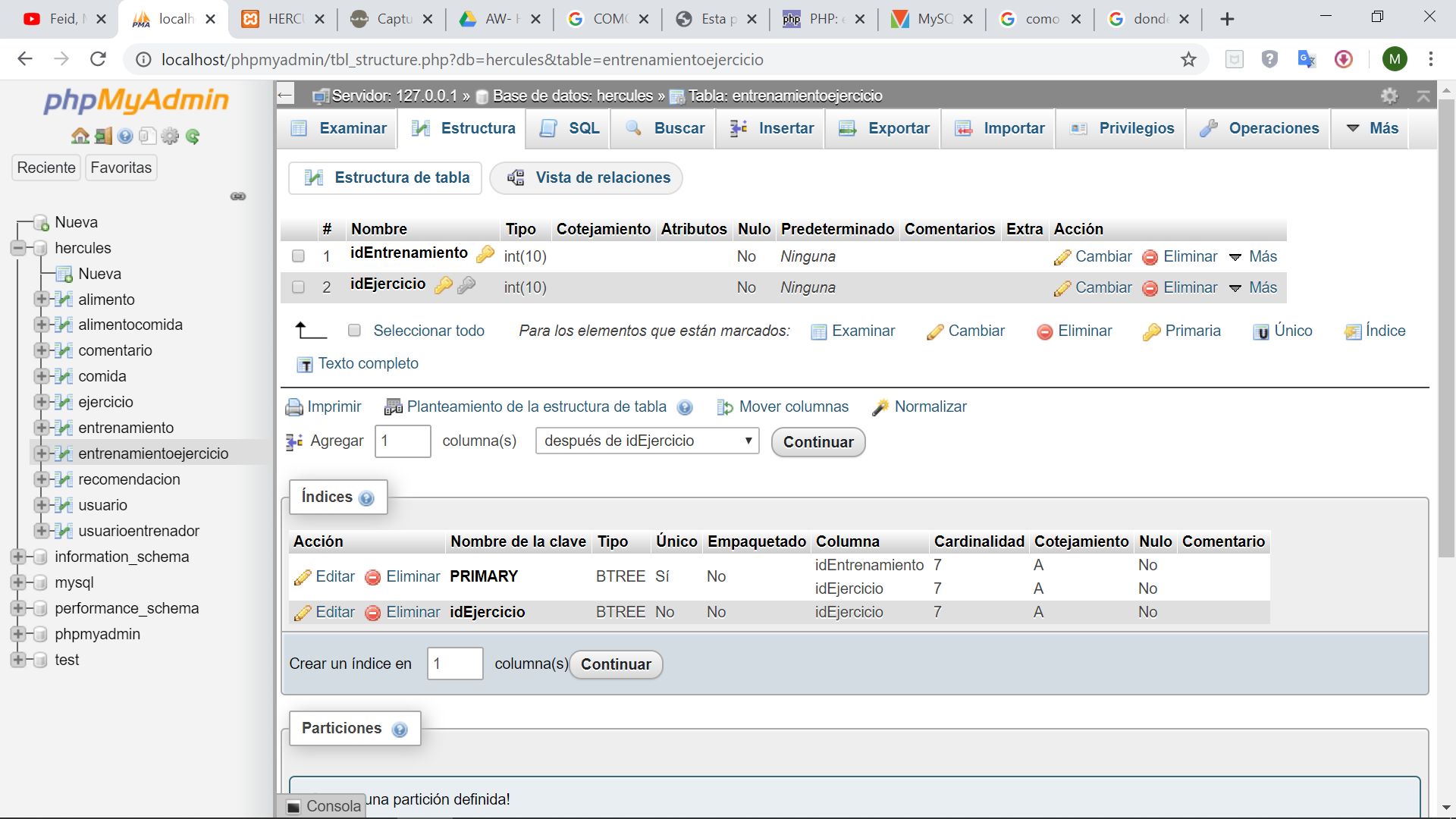


**Scripts que acceden a la tabla:**

* **entrenamientoDAO:** Consulta.
* **registrarEntrenamientos.**php: Modificación.
* **perfilMisEntrenamientos.**php: Consulta.

### **Entrenamientoejercicio**

La tabla entrenamientoejercicio es la tabla intermedia que vincula un entrenamiento con sus respectivos ejercicios a través de sus ids.



**Scripts que acceden a la tabla:**

* **entrenamientoDAO.php:** Consulta y modificación.
* **ejercicioDAO.php:** Consulta.

### **Alimento**

La tabla alimento, es una tabla que aloja los distintos alimentos que estarán disponibles para el usuario. El usuario tendrá acceso a estos alimentos a través de la tabla intermedia **alimentocomida**, a la que accederá cuando quiera registrar/modificar/eliminar una comida. La tabla alimento solo puede modificarla el usuario con privilegios de administrador.

Está compuesta por: un idAlimento, que es clave primaria y a su vez tendrá utilidad en la tabla intermediaria alimentocomida, y los distintos campos que son descripciones de ese alimento. Datos de interés para el usuario en cuestión.



**Scripts que acceden a la tabla:**

* **alimentoDAO:** Consulta.
* **comidaDAO:** Consulta.
* **miPerfilComidas.php -> RegistroComida.php -> InsertaComida.php:** Consulta.

### **Comida**

La tabla comida, es una tabla con la que el cliente interactúa, para registrar (más adelante también modificar y/o eliminar) las comidas y poseer más adelante un histórico de las mismas. Su clave primaria es el idComida (que se autoincrementa cada vez que se añade una comida a la tabla) y además, tiene el campo día (para conocer la fecha de registro), el tipo (que podrá ser desayuno, comida o cena) y el usuario, que es una clave externa que hace referencia al NIF del usuario cliente que tenga la sesión iniciada.



**Scripts que acceden a la tabla:**

* **comidaDAO:** Consulta y modificación.
* **miPerfilComidas.php -> RegistroComida.php -> InsertaComida.php:** Consulta y modificación.

### **Alimentocomida**

El propósito de esta tabla intermedia es relacionar los alimentos disponibles con las comidas que un usuario registra. Al ser una relación de muchos a muchos se ha creado esta tabla intermedia.



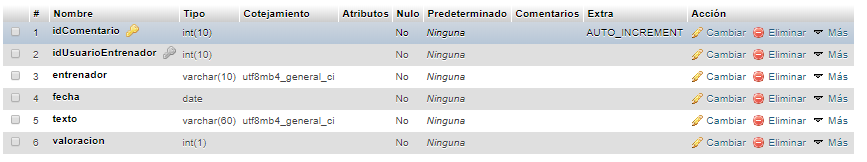
**Scripts que acceden a la tabla:**

* **comidaDAO:** Modificación.
* **miPerfilComidas.php -> RegistroComida.php -> InsertaComida.php:** Consulta y modificación.

### **Comentario**

El propósito de esta tabla es registrar los comentarios (valoraciones) que realiza un cliente a su entrenador. De ahí que uno de los campos sea “valoración”, para dar una puntuación a su entrenador. Su clave primaria es idComentario y posee distintos campos para conocer la fecha de la valoración, el texto (comentario) que hace el usuario, y la valoración en forma de puntuación (del 1 al 5) para su entrenador.

Las funcionalidades de esta tabla no están implementadas en esta práctica.



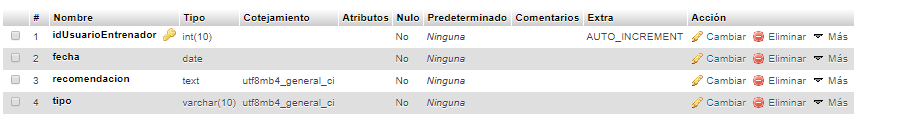
**Scripts que acceden a la tabla:**

* **comentarioDAO:** Consulta.

De momento estos scripts no cuentan con ninguna implementación.

### **recomendación**

El propósito de esta tabla es registrar las recomendaciones (consejos) que el entrenador realiza a un cliente. Va acompañado simplemente de un campo de texto y en el que se diferencia si la recomendación es del tipo comida o entrenamiento.

Su clave primaria es idRecomendacion y posee distintos campos para conocer la fecha de la valoración, el texto (recomendación) que hace el entrenador, y el tipo de recomendación que realiza (puede ser sobre comidas o sobre entrenamientos).

**Scripts que acceden a la tabla:**

* **recomendacionesDAO:** Consulta.

De momento estos scripts no cuentan con ninguna implementación.