# **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.

**(Imagen con página principal)**

**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 | Contraseña |
| Administrador | hercules | iG8hC62acnPrvIeU |
| Cliente |  |  |
| Cliente |  |  |
| Entrenador |  |  |
| Entrenador |  |  |

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

### **Estructura de carpetas**

A continuación, se aprecia la estructura de carpetas. Se detallará en el **apartado 6: Prototipo funcional del proyecto.**

**(Se incluye una única imagen con la estructura de carpetas).**

…

…

…

### **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 el apartado (...)
   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 (…)
2. **Del entrenador al cliente:**
   1. **Recomendar un entrenamiento:** Esta recomendación 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 (…)

## **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

Por completar con los scripts que se utilizan (formulario…) y explicación de funcionamiento: diagrama que indique el flujo de la aplicación…

### **MI PERFIL**

**Ruta**: hercules/miPerfil.php

Por completar con los scripts que se utilizan (formulario…) y explicación de funcionamiento: diagrama que indique el flujo de la aplicación…

### **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

Por completar con los scripts que se utilizan (formulario…) y explicación de funcionamiento: diagrama que indique el flujo de la aplicación…

### **LOGOUT**

**Ruta**: hercules/logout.php

Por completar con los scripts que se utilizan (formulario…) y explicación de funcionamiento: diagrama que indique el flujo de la aplicación…

### **REGISTRATE**

**Ruta**: hercules/registro.php

Por completar con los scripts que se utilizan (formulario…) y explicación de funcionamiento: diagrama que indique el flujo de la aplicación…

### **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 que el usuario. Se relaciona con comidaTO a través de una tabla intermedia de la base de datos. 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 añadirá?**

#### **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.

(Revisar que los campos coinciden con los de la tabla de la BD).

#### **comentariosTO: Pendiente de añadirse**

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: Cambiar nombre (Actualmente TOUsuario)**

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, ningún usuario accede a estos datos, a excepción de un usuario con privilegios de administrador.

#### **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: ¿Se añadirá?**

#### **recomendacionesDAO**

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

#### **comentariosDAO: Pendiente de añadir**

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:**

* **usuarioTO.php:** Consulta y modificación.
* **usuarioDAO.php:** Consulta.
* **Añadir scripts restantes.**

### **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:**

* **usuarioTO.php:** consulta y modificación.
* **usuarioDAO.php:** consulta y modificación. Es posible modificar el estado.
* **Añadir scripts restantes.**

### **Ejercicio**

…. (Explicar en qué consiste la tabla/campos y relaciones, su propósito.

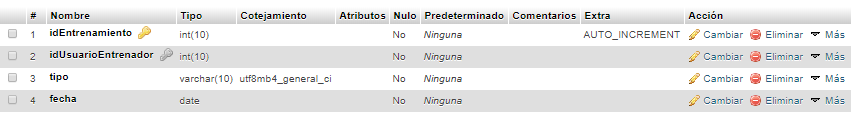


**Scripts que acceden a la tabla:**

…. (Añadir scripts restantes e Indicar si se consultan o también se modifican)

### **Entrenamiento**

…. (Explicar en qué consiste la tabla/campos y relaciones, su propósito.



**Scripts que acceden a la tabla:**

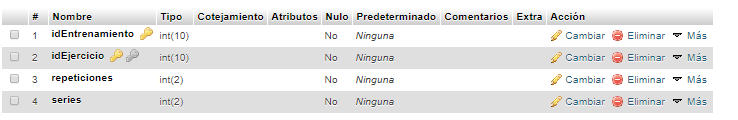
* **entrenamientoTO:** Consulta y modificación.
* **entrenamientoDAO:** Consulta.

…. (Añadir scripts restantes e Indicar si se consultan o también se modifican)

….

### **Entrenamientoejercicio**

…. (Explicar en qué consiste la tabla/campos y relaciones, su propósito.



**Scripts que acceden a la tabla:**

…. (Añadir scripts restantes e Indicar si se consultan o también se modifican)

….

### **Alimento**

…. (Explicar en qué consiste la tabla/campos y relaciones, su propósito.



**Scripts que acceden a la tabla:**

* **entrenamientoTO:** Consulta y modificación.
* **entrenamientoDAO:** Consulta.

…. (Añadir scripts restantes e Indicar si se consultan o también se modifican)

….

### **Comida**

…. (Explicar en qué consiste la tabla/campos y relaciones, su propósito.



**Scripts que acceden a la tabla:**

* **comidaTO:** Consulta y modificación.
* **comidaDAO:** Consulta.

…. (Añadir scripts restantes e Indicar si se consultan o también se modifican)

….

### **Alimentocomida**

…. (Explicar en qué consiste la tabla/campos y relaciones, su propósito.



**Scripts que acceden a la tabla:**

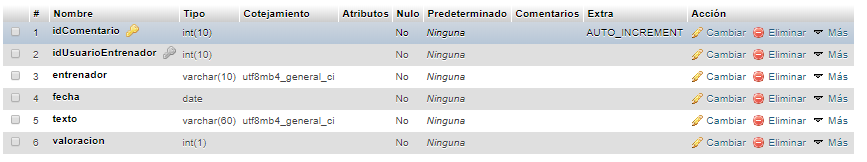
…. (Añadir scripts restantes e Indicar si se consultan o también se modifican)

….

### **Comentario**

…. (Explicar en qué consiste la tabla/campos y relaciones, su propósito.

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.



**Scripts que acceden a la tabla:**

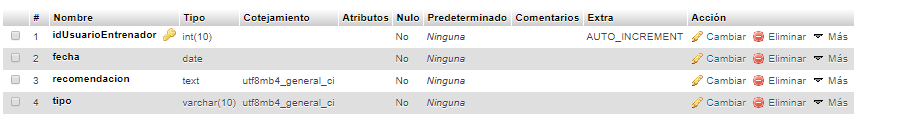
**Estos dos hay que crearlos (aunque de momento estén vacíos).**

* **comentarioTO:** Consulta y modificación.
* **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.

En esta tabla falta campo idRecomendacion

**Scripts que acceden a la tabla:**

* **recomendacionesTO:** Consulta y modificación.
* **recomendacionesDAO:** Consulta.

De momento estos scripts no cuentan con ninguna implementación.

## **PROTOTIPO FUNCIONAL DEL PROYECTO**

Aquí hay que añadir fotos con la estructura de las carpetas detallada…