# **PRÁCTICA 3.**

# **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.

## **Principales cambios con respecto la práctica anterior**

* **Cambios en la BD:**
  + Tablas Comentarios y Recomendaciones eliminadas.
  + Añadidas las tablas Mensajes y Valoraciones detalladas en la sección de la BD más abajo.
* **Funcionalidades añadidas:**
  + Se han añadido mas funcionalidades con respecto la gestión de entrenamientos y comidas por parte del cliente.
  + Funcionalidad de chat añadida.
  + Funcionalidad de reseña/valoración añadida.
  + El usuario puede editar su perfil ahora.
* **Funcionalidades mejoradas:**
  + La visualización de las comidas registradas se ha implementado de una manera más clara y ordenada.
  + La visualización de los entrenamientos registrados se ha implementado de una manera más clara y ordenada.

## **Descripción general**

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, obviamos este apartado y hemos llegado a la conclusión de no incluirlo.

**Captura de pantalla de un celular

Descripción generada automáticamente**

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

ACTUALIZAR

### **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.

Usuarios para realizar pruebas y verificar el contenido:

|  |  |  |  |
| --- | --- | --- | --- |
| Perfil de Usuario | Nombre | Nif | Contraseña |
| Administrador | Hercules |  | iG8hC62acnPrvIeU |
| Entrenador | LARA IBARRA | 12345678A | 12345678 |
| Entrenador | SERGIO PEINADO | 12345678B | 12345678 |
| Cliente | HUGAN | 12345678F | 12345678 |
| Cliente | PACO | 12345678G | 12345678 |

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

**La BD también incluye:**

* 14 ejercicios en la tabla **ejercicio**, con sus respectiva imagen y descripción.
* 14 alimentos en la tabla **alimento.**
* 5 usuarios de tipo entrenador.
* 4 usuarios de tipo cliente.

ACTUALIZAR

### **Estructura de carpetas**

A continuación, se aprecia la estructura de carpetas.

**hercules**

**│** administrador.php

**│** contacto.php

│ entrenadores.php

│ faqs.php

│ foro.php

│ gestionarEjercicios.php

│ gestionarUsuario.php

│ index.php

│ login.php

│ logout.php

│ mensaje.php

│ miPerfil.php

│ miPerfilBuzon.php

│ miPerfilCabecera.php

│ miPerfilComidas.php

│ miPerfilComidasEditar.php

│ miPerfilComidasEliminar.php

│ miPerfilComidasRegistrar.php

│ miPerfilComidasVer.php

│ miPerfilComidasVerCalendario.php

│ miPerfilComidasVerTablas.php

│ miPerfilEditar.php

│ miPerfilEntrenamientos.php

│ miPerfilEntrenamientosRegistrar.php

│ miPerfilEntrenamientosVer.php

│ miPerfilEntrenamientosVerDetalle.php

│ miPerfilMisClientes.php

│ miPerfilMisClientesPerfiles.php

│ miPerfilMisEntrenadores.php

│ miPerfilMisEntrenadoresPerfiles.php

│ nuevoTema.php

│ PR\_eliminarEntrenador.php

│ PR\_eliminarEntrenamiento.php

│ PR\_entrenadores\_check.php

│ PR\_miPerfilBuzon\_chat.php

│ PR\_miPerfilMisClientesPerfiles\_buzon.php

│ PR\_miPerfilMisEntrenadoresPerfiles\_valorar.php

│ quienes\_somos.php

│ registro.php

│ respuesta.php

**│**

**├───.settings**

**│** org.eclipse.php.core.prefs

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

**│**

**├───includes**

│ │ config.php

│ │ controller.php

│ │ estiloPaginaPrincipal.css

│ │ uploader.php

**│ ├───comun**

**│ │** cabecera.php

│ │ pie.php

**│ │**

**│ ├───css**

**│ │** estilo.css

│ │ estiloFormularios.css

│ │ estiloPagsMiPerfil.css

**│ │**

**│ ├───DAOs**

**│ │** alimentoDAO.php

│ │ comidaDAO.php

│ │ DAO.php

│ │ ejercicioDAO.php

│ │ entrenamientoDAO.php

│ │ foroDAO.php

│ │ mensajesDAO.php

│ │ usuarioDAO.php

│ │ valoracionDAO.php

**│ │**

**│ ├───Forms**

**│** │ Form.php

│ │ FormEliminarComida.php

│ │ FormRegistroComida.php

│ │ FormularioEditarPerfil.php

│ │ FormularioLogin.php

│ │ FormularioNuevoTema.php

│ │ FormularioRegistrarEntrenamiento.php

│ │ FormularioRegistro.php

│ │ FormularioRespuesta.php

**│ │**

**│ ├───img**

│ │ │Hercules\_logo.png

│ │ │ y todas las imágenes.

**│ │ │**

**│ │ └───nosotros**

**│ │ │** imágenes del equipo

│ │ │

**│ │ └───usuarios**

**│ │** imágenes de perfil de usuarios

**│ │**

**│ ├───js**

│ │ banner.js

│ │ filtroEntrenadores.js

│ │ scripts.js

│ │

**│ └───TOs**

**│** alimentoTO.php

│ comidaTO.php

│ ejercicioTO.php

│ entrenamientoTO.php

│ TOUsuario.php

**│**

**└───mysql**

hercules.sql

usuarios.sql

YA ESTÁ ACTUALIZADO PERO SI A PARTIR DEL 10/05 (a las 11:00 aprox) SE AÑADEN MÁS FICHEROS HAY QUE METERLOS.

### **Funcionalidades práctica 3**

Las funcionalidades que implementamos en esta práctica junto con las funcionalidades anteriores son:

1. **Del cliente:**
   1. **Solicitar un entrenador:** Esta solicitud se realiza accediendo a la pestaña **“Nuestros entrenadores”**, ver el perfil de uno de ellos y mandar la solicitud con el botón correspondiente.
   2. **Ver comidas:** No es una nueva funcionalidad, pero está completamente cambiada. Se puede acceder de la siguiente manera: **“Mi Perfil” -> “Comidas” -> “Ver Comidas”.** A continuación aparecerán dos botones y funcionalidades: **“Tabla comidas**” y **“Calendario comidas”**.
   3. **Registrar una comida:** Esta solicitud se realiza accediendo a la pestaña **“Mi perfil”** y luego **“Comidas” -> “Registrar Comida”**.
   4. **Eliminar comidas (nuevo):** En la pestaña **“Mi Perfil” -> “Comidas” -> “Eliminar Comidas”** se puede eliminar una comida seleccionando su fecha.
   5. **Eliminar entrenadores(nuevo):** Se puede eliminar un entrenador en la pestaña **“Mi Perfil” -> “Mis entrenadores” -> “Mostrar Perfil”** al pulsar en el enlace “Eliminar de la lista”.
   6. **Ver entrenamientos:** En la pestaña **“Mi Perfil” -> “Mis entrenadores” -> “Mostrar Perfil”** también se pueden ver los entrenamientos que ha propuesto dicho entrenador.
   7. **Escribir reseña/valoración(nuevo):** En la pestaña **“Mi Perfil” -> “Mis entrenadores” -> “Mostrar Perfil”** de una de los entrenadores hay un botón para “Dejar una reseña”. Al pulsar, el botón aparecerá un formulario en forma de Popup con el que se podrá rellenar la reseña y elegir si es visible para el público.
   8. **Editar perfil(nuevo):** En la pestaña **“Mi Perfil” -> “Editar Perfil”** aparecerá un formulario con todos los datos del perfil. El usuario puede cambiarlos libremente y mandar el formulario para confirmar los cambios.
   9. **Enviar/Recibir mensajes(nuevo):** En la pestaña **“Mi Perfil” -> “Mensajes”** aparecerá una “bandeja de entrada” con todos los mensajes no borrados hacia el usuario, además de la opción de mandar un mensaje de saludo a un usuario mediante su nif.
2. **Del entrenador:**
   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
   2. **Crear entrenamientos(nuevo):** En la pestaña **“Mi Perfil” -> “Mis entrenadores” -> “”** también se pueden ver los entrenamientos que ha propuesto dicho entrenador.
   3. **Eliminar entrenamientos(nuevo):**
   4. **Enviar/recibir mensajes(nuevo):** Exactamente igual que en el apartado l de clientes.
   5. **Editar perfil(nuevo):** Exactamente igual que en el apartado k de clientes.

## **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.

En esta práctica 3 la página de inicio integra un banner de javascript que se puede ver en:

hercules/includes/js/banner.js para dar una apariencia mejor a la página.

### **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, mensajes y mis entrenadores o mis clientes, dependiendo de si has entrado como cliente o entrenador.

**Para los clientes:**

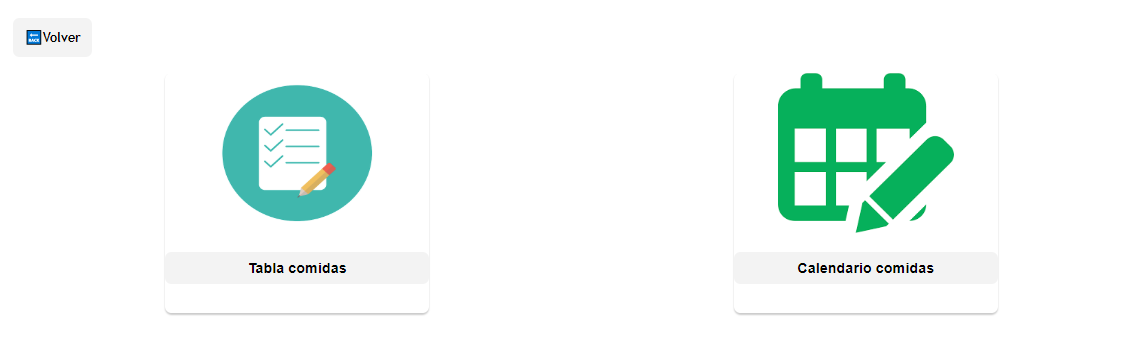
* Las pestañas **“Comidas” -> “Registrar Comida”** llevarán al usuario al editor de dietas, que podrá registrar lo que ha desayunado/comido o cenado. Existirá un listado de alimentos para seleccionar uno entre los posibles, y podrán añadirse, de momento, 3 alimentos, que se corresponden con el primer plato, el segundo plato y el postre. Se ha añadido una tabla con toda la información (calorías, grasas, …) relevante sobre los alimentos disponibles, con el objetivo de facilitar las decisiones al usuario a la hora de escoger una comida u otra. Cuando se realice el registro correctamente, el usuario pasará a ver todas las comidas registradas en una tabla (funcionalidad “Tabla comidas”, explicada más abajo), para comprobar que se ha insertado su comida. Para registrar una comida se pasará por los siguientes scripts: **miPerfilComidas.php -> miPerfilComidasRegistrar.php -> miPerfilComidasVerTablas.php.**



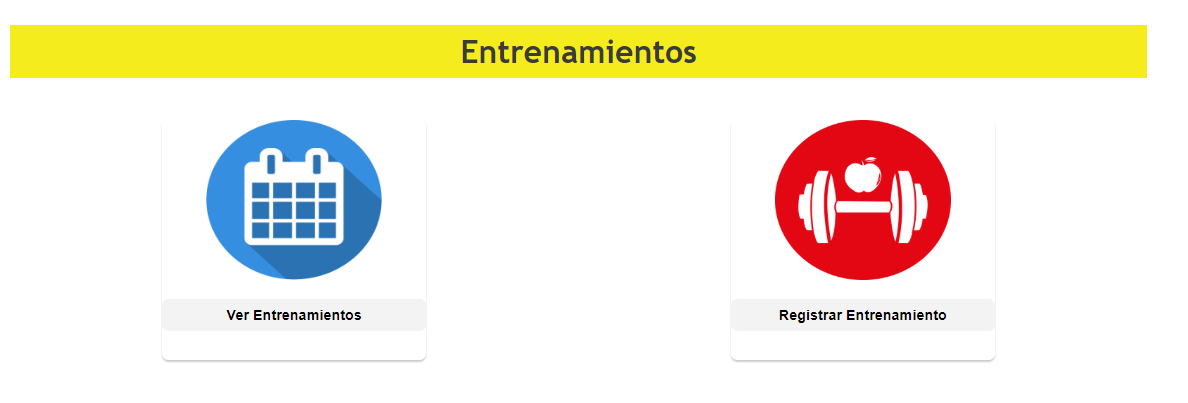
* **“Comidas” -> “Ver comidas”** incluye dos scripts:

**“Tabla comidas**”, que sirve para ver todas las comidas que ha registrado el usuario. Se trata de una tabla que contiene datos sobre las comidas, como la fecha de registro, los alimentos que pertenecen a esa comida, o las calorías totales de la comida. (**miPerfilComidas.php -> miPerfilComidasVer.php -> miPerfilComidasVerTablas.php**)

El otro script, **“Calendario comidas”**, permite ver las comidas registradas por el usuario la semana actual, y la semana anterior a esta. También se ha realizado con una tabla, donde cada columna es un día de la semana. En la tabla aparece una comida en cada fila, y se establece como orden la fecha de registro. En cada celda se muestra la hora de registro y el tipo de comida. (**miPerfilComidas.php -> miPerfilComidasVer.php -> miPerfilComidasVerCalendario.php)**



* A través de **“Comidas” -> “Eliminar Comida”** se podrá eliminar una fecha. Simplemente hay que introducir la fecha de registro, que aparecerá en una lista de selección con todas las fechas. Para consultar la fecha de registro habrá que acceder a la funcionalidad **“Tabla comidas**” o **“Calendario comidas”**, si la fecha no es antigua. Cuando se borra una comida se accede directamente al script **miPerfilComidasVerTablas.php**, para comprobar que la acción se ha realizado con éxito. (**miPerfilComidas.php -> miPerfilComidasEliminar.php -> miPerfilComidasVerTablas.php)**
* **“Comidas” -> “Editar Comida”**. Estará disponible para la próxima entrega. (**miPerfilComidas.php -> miPerfilComidasEditar.php**)
* La pestaña Entrenamientos muestra los entrenamientos del propio cliente. Además, a ver el perfil del entrenador, es posible visualizar los entrenamientos propuestos por dicho entrenador. Se ha añadido un enlace para empezar un chat con él/ella. Si hay un entrenador que todavía no ha aceptado una solicitud, no 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.**



* 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. Se ha añadido un enlace para empezar un chat con él/ella. Si hay un entrenador que todavía no ha aceptado una solicitud, no 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.**
* La pestaña EditarPerfil muestra los datos del perfil de un cliente. La página en si es un formulario con los campos propios del perfil de un cliente. La página de procesamiento a la que se envía el formulario es **FormularioEditarPerfil.php**
* La pestaña Mensajes muestra los chats activos que tiene el usuario**.** Un chat está activo si existe al menos un mensaje entre el usuario y el otro extremo que no haya borrado el propio usuario. Cada chat tiene la opción de Borrar la Conversación, lo cuál borrará todos los mensajes dentro del chat para el propio usuario (el otro extremo no notará diferencia). Además al principio de todo hay un pequeño formulario para mandar un mensaje, y así empezar un chat, con un usuario según su nif. El fichero de esta vista es **miPerfilBuzon.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.**
* La pestaña Editar Perfil junto la de Mensajes son similares a los de la vista de Cliente.

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 ->**

**miPerfilComidas.php ->**

miPerfilComidasVer.php ->

miPerfilComidasVerTablas.php

miPerfilComidasVerCalendario.php

miPerfilComidasRegistrar.php

miPerfilComidasEditar.php

miPerfilComidasEliminar.php

**miPerfilEntrenamientos.php ->**

miPerfilEntrenamientosRegistrar.php

miPerfilEntrenamientosVer.php

miPerfilEntrenamientosVerDetalle.php

PR\_eliminarEntrenamiento.php

**miPerfilMisEntrenadores.php ->**

**perfil\_Entrenador ->**

miPerfilMisEntrenadoresPerfiles.php

PR\_miPerfilMisEntrenadoresPerfiles\_valorar.php

**miPerfilMisClientes.php ->**

**perfil\_Cliente ->**

miPerfilMisClientesPerfiles.php

PR\_miPerfilMisClientesPerfiles\_buzon.php

**miPerfilBuzon.php ->**

PR\_miPerfilBuzon\_chat.php

**miPerfilEditar.php ->**

Esta ventana es accesible desde el menú principal.

ACTUALIZAR

### **QUIENES SOMOS**

**Ruta**: hercules/quienesSomos.php

Esta página contiene la información relacionada al equipo que trabaja en la aplicación. Es de carácter informativo.

### **FAQS**

**Ruta**: hercules/faqs.php

Esta página es la pestaña de preguntas frecuentes de nuestra aplicación web. Es de carácter informativo.

### **CONTACTO**

**Ruta**: hercules/contacto.php

Esta página es la pestaña de contacto de nuestra aplicación web. Es de carácter informativo.

### **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 que en esta práctica ha sido mejorado. Además de pedir el NIF/NIE, nombre, email y contraseña de forma **obligatoria** para registrarse, se pide subir una foto de perfil, fecha de nacimiento, número de teléfono (opcional), peso, altura, preferencias, ubicación. 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 entrará a la página con la cuenta que se acaba de crear.

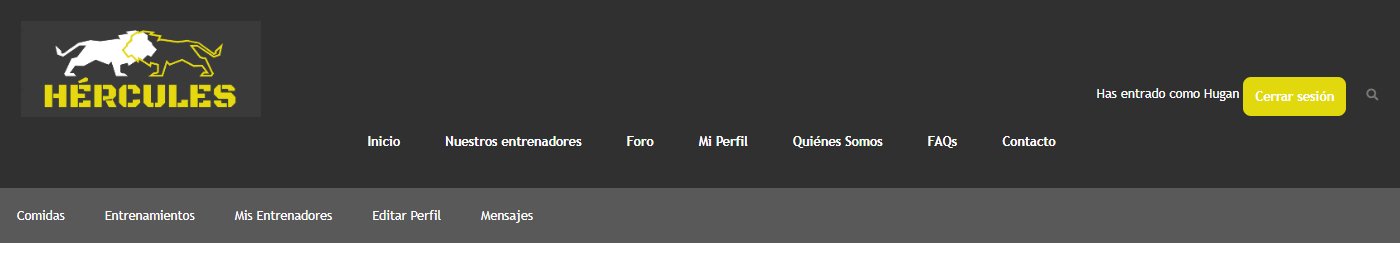
* FormularioRegistro.php: Clase FormularioRegistro que hereda de Form. Contiene la estructura del formulario y la lógica del procesamiento de los datos a rellenar.



### **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.

#### **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 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.

#### **mensajesDAO**

Esta clase aloja las operaciones CRUD relacionadas con los chats.

#### **valoracionDAO**

Esta clase aloja las operaciones CRUD relacionadas con las valoraciones de los usuarios hacia un entrenador.

### **JavaScript**

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

#### **scripts**

Este archivo contiene scripts que se usan de forma general. Por ahora solo se usa para mostrar o esconder popups.

#### **Banner**

Este archivo se utiliza en la página principal, es el banner que integra 3 imágenes y van pasando en forma de slider.

#### **filtroEntrenadores**

Este archivo se utilizará en la página de nuestros entrenadores. Es un filtro para clasificar los entrenadores según los parámetros que se pasen. Finalmente para esta práctica está totalmente implementado. En proceso.

### **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 valoraciones, funciones de comidas, alimentos, entrenamientos y ejercicios.

### **CONFIG**

En esta clase instanciamos el controller de la aplicación.

### **css**

Los estilos que hemos añadido en nuestra aplicación web hemos querido clasificarlos para separar determinadas vistas y poder gestionar de una manera más asequible toda la estructura visual.

#### **Estilos**

Integra el estilo general de la página, todos los scripts de vista incluyen este estilo. Todo lo correspondiente al menú y submenú lo integra este estilo.

#### **EstiloFormularios**

Este estilo se utiliza para los formularios. Para registrar al usuario, iniciar sesión, registrar comidas, entrenamientos y el resto de los formularios. Su estructura normalmente varía de 2 maneras (al menos en el width) y puede ser que en algún caso exista algún añadido o pequeña variación muy concreta de ese formulario.

#### **EstiloPagsMiPerfil**

Este estilo se utiliza para todas las páginas que dependen de la pestaña del menú principal “mi perfil”. Es decir, desde el lado del cliente: comidas, entrenamientos, mis entrenadores, editar perfil y mensajes. En cuanto al entrenador mis clientes y editar mi perfil.

#### **EstiloPagPrincipal**

Este estilo integra expresamente lo relacionado con el banner principal de la página, hemos querido separarlo del estilo general, ya que se utiliza de manera exclusiva en la página de inicio. Las mejoras que se realicen posteriormente sobre la página principal serán integradas en este .css.

## **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.
* **miPerfilEditar.php:** Consulta y modificación.
* **miPerfilMisEntrenadoresPerfiles:** Consulta.
* **miPerfilMisClientesPerfiles:** Consulta.

### **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 -> PR\_eliminarEntrenador.php:** Modifica**.**
* **PR\_miPerfilMisEntrenadoresPerfiles\_valorar.php:** 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.

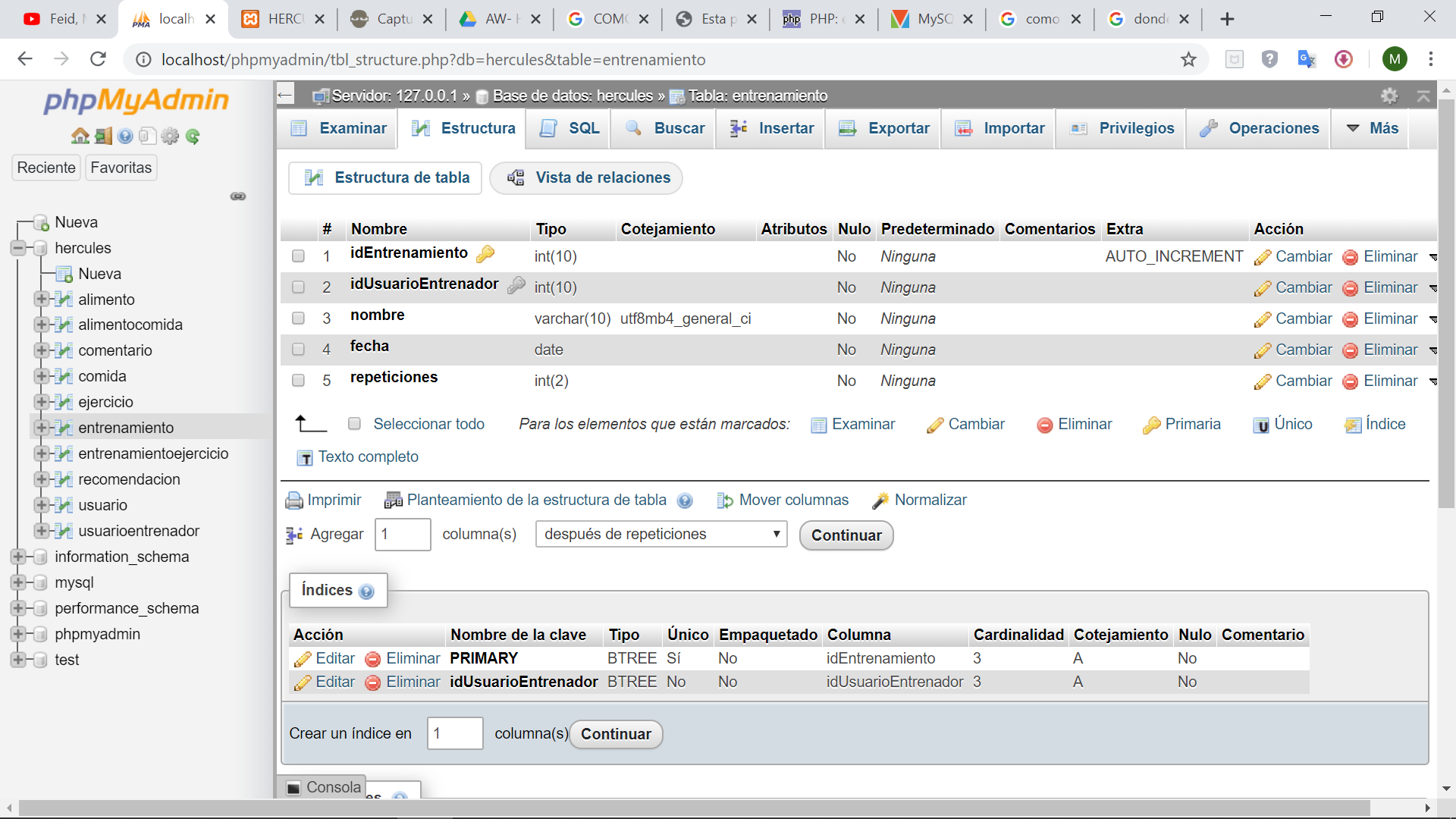


**Scripts que acceden a la tabla:**

* **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.

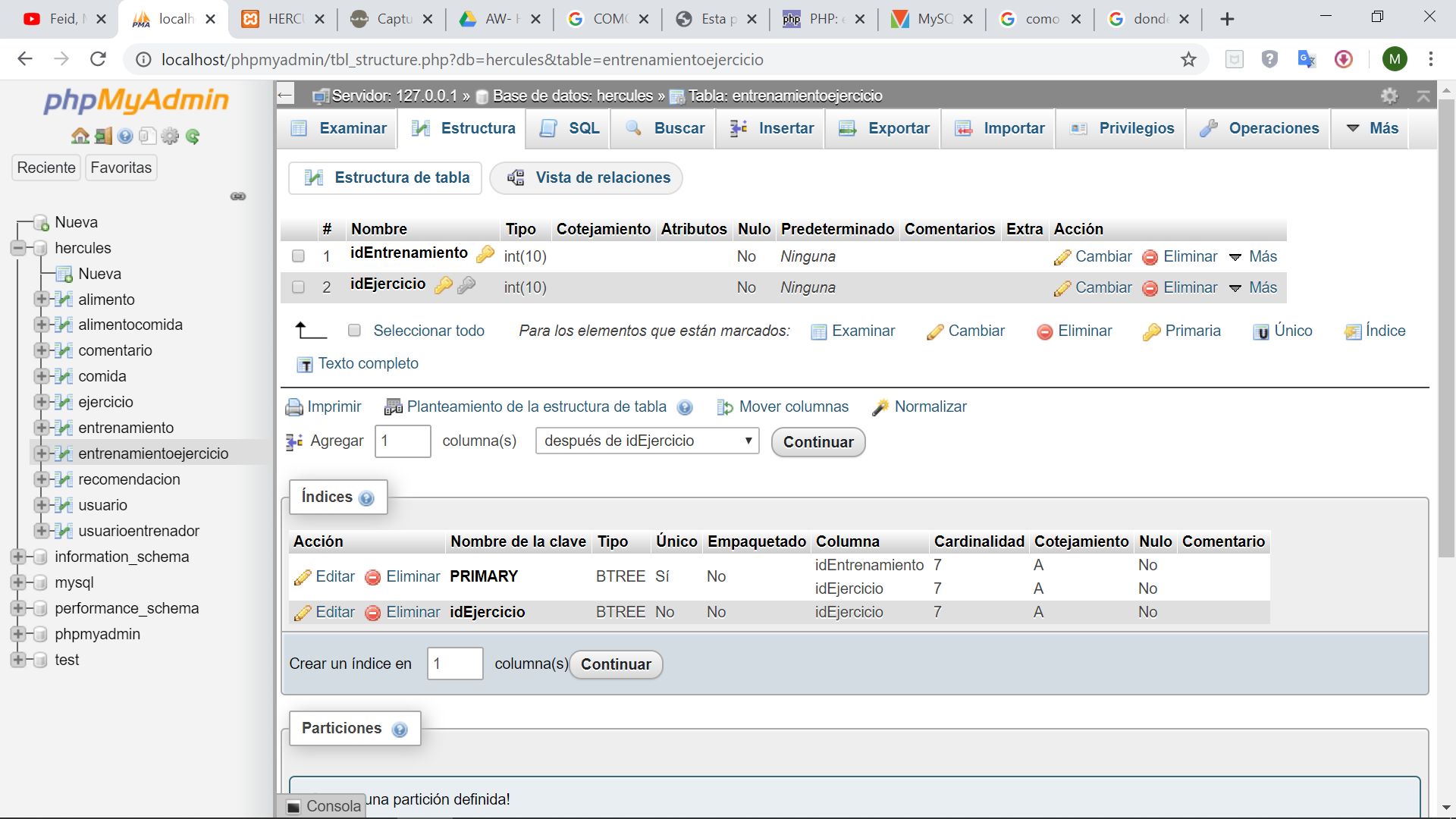


**Scripts que acceden a la tabla:**

* **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**, la cual modificará 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.php:** Consulta.
* **comidaDAO.php:** Consulta.
* **miPerfilComidasVerTablas.php:** Consulta.
* **miPerfilComidasVerCalendario.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.
* **miPerfilComidasVerTablas.php:** Consulta.
* **miPerfilComidasVerCalendario.php:** Consulta.
* **miPerfilComidasRegistrar.php:** Consulta y modificación.
* **miPerfilComidasEliminar.php:** 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:** Consulta y modificación.
* **miPerfilComidasVerTablas.php:** Consulta.
* **miPerfilComidasVerCalendario.php:** Consulta.
* **miPerfilComidasRegistrar.php:** Modificación.

### **Valoracion**

Esta tabla se encarga de almacenar todos las reseñas/valoraciones que los clientes dan a su entrenador.

Su clave primaria es la composición de hacia y de que contendrán nifs de entrenador(hacia) y cliente(de). Texto es el comentario de la valoración, valor son las “estrellas” que se les da(del 1 al 5), visible es un booleano que indica si la valoración estará disponible para el público o no y fecha es de tipo current\_timestamp.

Una captura de pantalla de una red social

Descripción generada automáticamente

**Scripts que acceden a la tabla:**

* **valoracionDAO:** Consulta y modificación.

### **MENSAJES**

Esta tabla se encarga de almacenar todos los mensajes que los usuarios se mandan entre sí.

Su clave primaria es id, de tipo auto\_increment. Emisor y receptor contendrán los nif de los correspondientes usuarios, texto es el texto del mensaje, fecha es de tipo current\_timestamp y visto indica si el receptor ha visto el mensaje o no.

Del\_1 y del\_2 son booleanos que indican si el emisor(del\_1) o el receptor(del\_2) de un mensaje lo ha borrado. Si estas dos columnas están en true(1) se procederá a eliminarlas.

Captura de pantalla de un celular

Descripción generada automáticamente

**Scripts que acceden a la tabla:**

* **mensajesDAO:** Consulta y modificación.
* **miPerfilBuzon.php:** Consulta.

### **foro**

Esta tabla hará algo.

**Scripts que acceden a la tabla:**

* **foroDAO:** Consulta y modificación.

## **Tabla de distribución de trabajo**

|  |  |
| --- | --- |
| **Nombre y Apellidos** | **Aportación** |
| Emilio José Valencia Calvopiña |  |
| Manuel Espinosa Guerra |  |
| Geraldyn Carrero Azuaje | **CSS** de: principal, cabecera (menús y submenús de la página), formularios de registro e inicio de sesión (usados también en otras partes), páginas internas (acceso a partir de mí perfil). Página inicio con banner. |
| Cheng Jun Liu Zheng | Nuevas funcionalidades de Chat y Creación de Valoraciones/Reseñas**.** |
| Miriam Elizabeth Cabana Ramírez |  |
| Jaime Madriñán Fernández | Funcionalidades de comidas. |
| Mingyang Chen |  |