



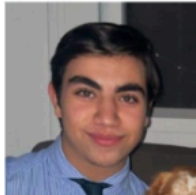
TERCFIT

Miembros del equipo

Antonio Armenteros Iranzo

amai0001@red.ujaen.es

http://34.170.202.243



David García Padilla

dgp00036@red.ujaen.es

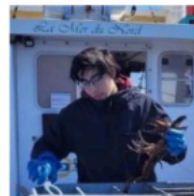
http://35.239.143.249



Hechun Ouyang

ho000006@red.ujaen.es

http://35.202.21.149



Prof.^a de prácticas: Alba María Mármol Romero

Prof. de teoría: José Ramón Balsas Almagro

URL repositorio: <https://github.com/DAWUJA-2025/g1-equipo-103>

ÍNDICE:

0.- ÍNDICE:	2
1.- DESCRIPCIÓN GENERAL DEL PROYECTO	3
1.1.- INFORMACIÓN SOBRE TERCFIT	3
1.2.- REQUISITOS FUNCIONALES DEL USUARIO	3
1.3.- REQUISITOS FUNCIONALES DEL ADMINISTRADOR	3
2.- DIAGRAMAS Y STORYBOARDS	4
2.1.- DIAGRAMA ENTIDAD / RELACIÓN	4
2.2.- DIAGRAMA DE CLASES	4
2.3.- STORYBOARDS	5
3.- METODOLOGÍAS Y TECNOLOGÍAS	5
3.1.- METODOLOGÍA ÁGIL KANBAN	6
3.2.- BEANS	6
3.3.- CONTROLLERS	7
3.4.- DAO	7
3.5.- API REST	7
4.- MANUAL DE USUARIO	7
4.1.- CÓMO NAVEGAR POR NUESTRA WEBAPP	7
4.2.- MENÚ PRINCIPAL	8
4.3.- APARTADO PERFIL DE USUARIO	8
4.4.- APARTADO RECETAS	8
4.5.- APARTADO EJERCICIOS	8
5.- MEMORIA DE ACTIVIDAD	9
5.1.- ITERACIÓN INICIAL	9
5.2.- ITERACIÓN 1	9
5.3.- ITERACIÓN 2	9
5.4.- ITERACIÓN 3	9
5.5.- ITERACIÓN 4	9
5.6.- ITERACIÓN 5	9
6.- CONCLUSIONES	10
7.- BIBLIOGRAFÍA Y RECURSOS ONLINE	10

1.- DESCRIPCIÓN GENERAL DEL PROYECTO

1.1.- INFORMACIÓN SOBRE TERCFIT

TERCFIT es una WebApp diseñada para ayudar a los usuarios a planificar, personalizar y registrar sus ejercicios, además de realizar un seguimiento de sus estadísticas corporales y poder acceder a recetas saludables y fáciles de hacer.

1.2.- REQUISITOS FUNCIONALES DEL USUARIO

El usuario podrá:

- Identificarse con unas credenciales correctas
- Ingresar datos biométricos
- Consultar registros de los datos biométricos realizados
- Buscar ejercicios
- Añadir ejercicios a favoritos
- Crear nuevos ejercicios
- Consultar ejercicios de recetas para gestionarlos
- Modificar las estadísticas de un ejercicio
- Crear nuevas recetas
- Ver detalles de las recetas
- Consultar registros de recetas para gestionarlas

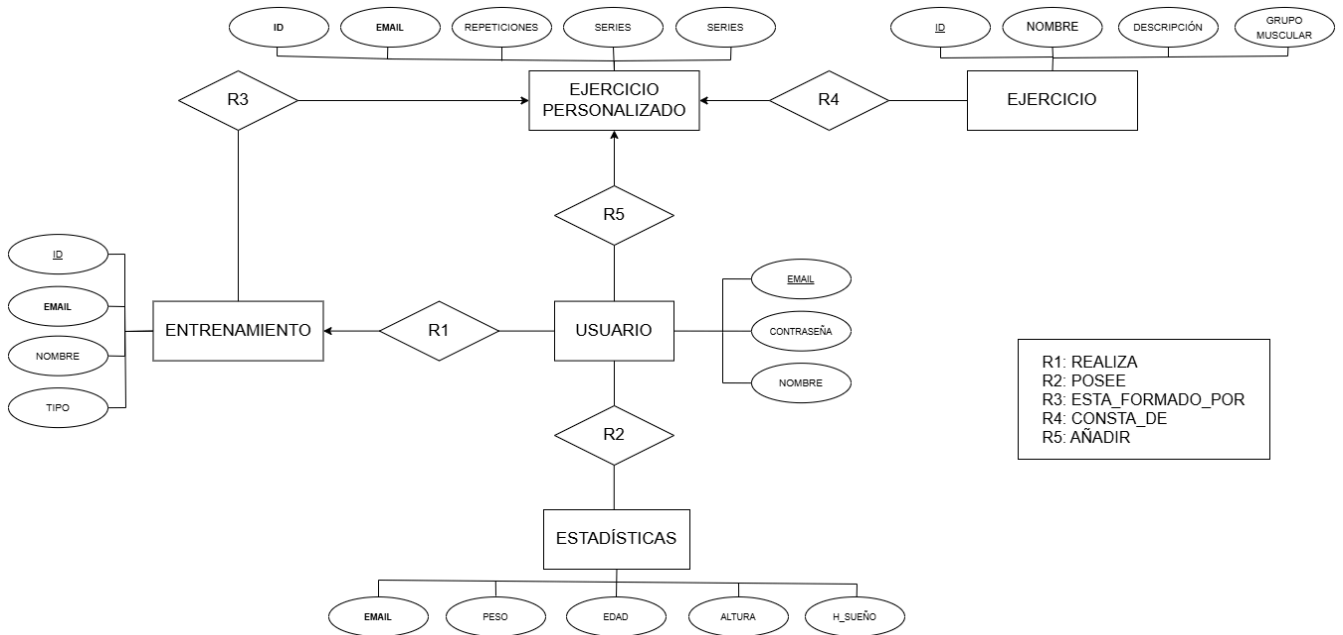
1.3.- REQUISITOS FUNCIONALES DEL ADMINISTRADOR

El administrador podrá:

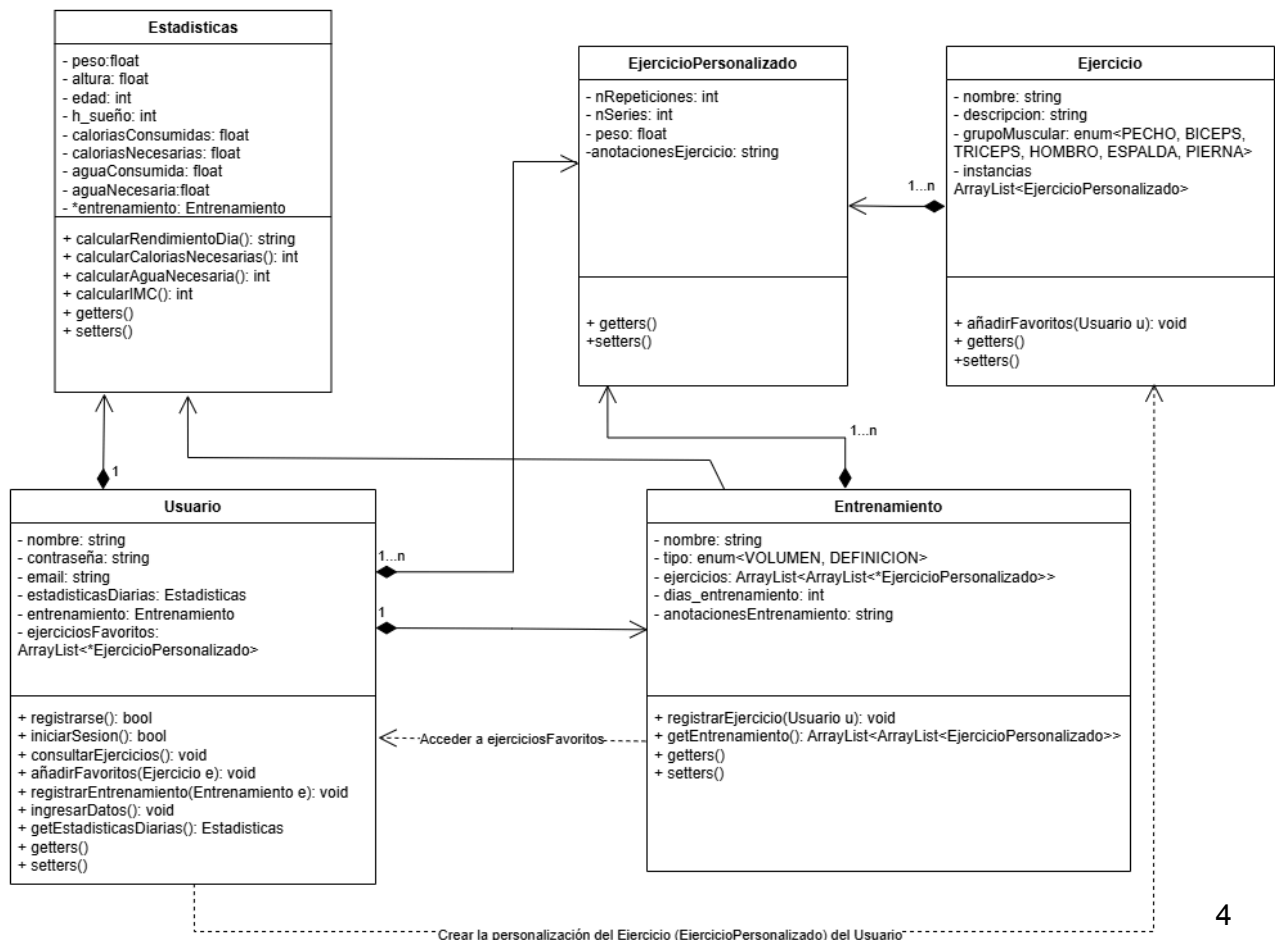
- Identificarse con unas credenciales correctas
- Acceder al panel de control del admin
- Ver las credenciales activas del sistema
- Acceder a la API de los clientes
- Acceder a la API de los ejercicios
- Acceder a la API de las recetas
- Acceder a la lista de usuarios en tiempo real
- Así como todas las acciones que puede realizar el usuario

2.- DIAGRAMAS Y STORYBOARDS

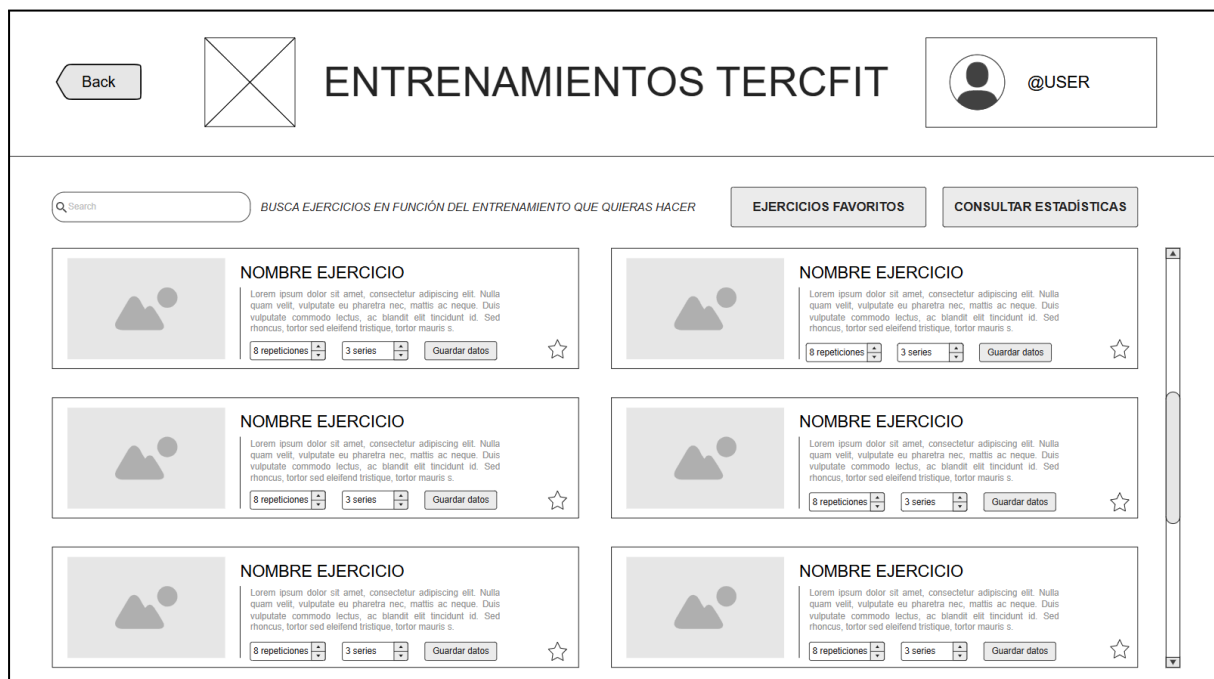
2.1.- DIAGRAMA ENTIDAD / RELACIÓN



2.2.- DIAGRAMA DE CLASES



2.3.- STORYBOARDS

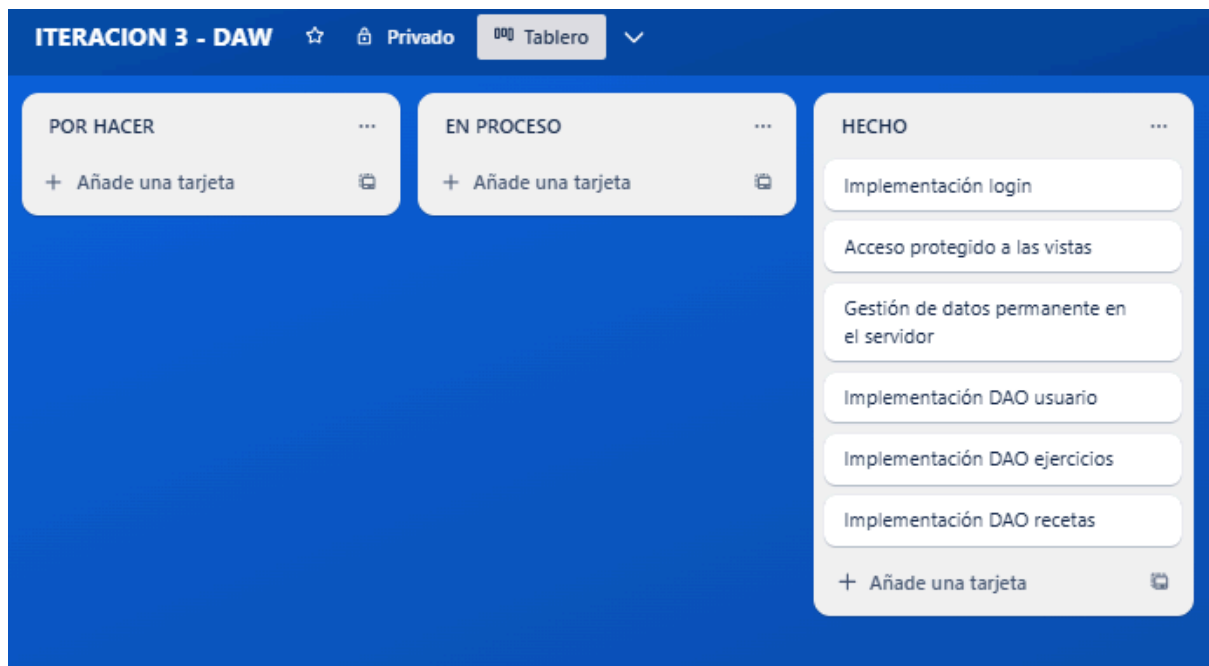


No hemos añadido todos los Storyboards, solo los que consideramos más importantes.

3.- METODOLOGÍAS Y TECNOLOGÍAS

3.1.- METODOLOGÍA ÁGIL KANBAN

Para el desarrollo del proyecto en cada iteración hemos creado un tablero en trello y hemos aplicado la metodología KANBAN mediante tarjetas para saber quien implementaba y lo que quedaba por implementar en cada iteración.



Como se aprecia en la foto ese es el tablero que teníamos para la iteración 3 una vez ya la finalizamos, al principio todas las tareas estaban en la columna *POR HACER* y cuando algún miembro del equipo trabaja en la iteración la movía a la columna *EN PROCESO* de esta manera si otro miembro empezaba a trabajar sabía que tenía que elegir otra tarea, una vez finalizada lo movía a *HECHO* y actualizaba el archivo CHANGELOG.md.

3.2.- BEANS

En el proyecto empezamos con más beans implementados pero con el desarrollo de las iteraciones se ha optado por el uso de controladores debido a las necesidades de ciertas clases y se han reducido a dos, uno para tener la fecha actual siempre en el perfil de usuario y otro para obtener las credenciales activas almacenadas en una lista en el panel del administrador.

3.3.- CONTROLLERS

Hemos implementado un total de 6 controladores. Uno para el admin, biometría, cliente, login, ejercicio y receta. Cada controlador tiene implementado el CRUD excepto el del login que tiene lo necesario para conectarse y desconectarse.

3.4.- DAO

Tenemos un modelo DAO general para cada entidad del proyecto del cual se extienden tanto un DAOMap como un DAOJpa. Esto se aplica para las clases clientes, biometría, ejercicios y recetas. Además hemos almacenado datos en los DAOMap utilizándolos como herramienta principal para la API RES, así como se han implementado los métodos necesarios para buscar en ella. Los DAOJPAs se han utilizado principalmente para garantizar la persistencia con de los DAo y que la comunicación se realice a través de jpa tal y como se pedía.

3.5.- API REST

La API REST de nuestro proyecto utiliza métodos HTTP como GET, POST, PUT y DELETE para interactuar con los recursos de la WebApp. Tenemos creada una para los clientes para las recetas y otra para los ejercicios. Además en el directorio /webservices hemos añadido un método para validar las excepciones y generar JSON's de respuesta.

4.- MANUAL DE USUARIO

4.1.- CÓMO NAVEGAR POR NUESTRA WEBAPP

Para empezar a usar la aplicación, primero debes identificarte con las credenciales correctas. En el apartado de más información hay un botón privado para acceder al panel de control del administrador.

Hemos creado un usuario administrador de prueba para que el profesor pueda probar la aplicación, las credenciales son las siguientes.

Usuario: profedaw

Contraseña: profedaw

Cabe destacar que estas vistas constan de un acceso protegido, si no te identificas de forma correcta no podrás acceder a la aplicación o al panel de administrador. También tenemos una navegación como invitado, no obstante es una simple versión demo de nuestra WebApp muy simple.

4.2.- MENÚ PRINCIPAL

Esta es la primera vista que el usuario se encuentra al iniciar la WebApp. Presenta informaciones sobre los casos de uso, como la creación de recetas, de ejercicios, y las ventajas de utilizar nuestra aplicación. Hay botones al final de la página uno muestra más información sobre la aplicación y el otro permite descargar la memoria final del proyecto.

La vista contiene un espacio header que contiene un botón identificarse. Este botón lleva a una página donde el usuario puede introducir sus credenciales o entrar en la aplicación como invitado.

Los credenciales son determinados en el AppConfig de manera fija, y el usuario debe contactarnos para crear una cuenta si no tiene credenciales. Si el usuario decide entrar como invitado, las funciones de la aplicación son limitadas.

Cuando el usuario se identifica correctamente, le aparece el menú de navegación con recetas y ejercicios en el header por el cual puede navegar.

4.3.- APARTADO PERFIL DE USUARIO

En esta vista el usuario puede encontrar la fecha actual, el menú de navegación de la aplicación, el botón para cerrar sesión, información y un gráfico sobre su biometría, un carrusel de ejercicios favoritos que puede ir guardando y por último un botón para poder añadir nuevas estadísticas biométricas.

4.4.- APARTADO RECETAS

En esta vista el usuario puede navegar por el menú de navegación, puede ver los valores nutricionales de las recetas creadas e incluso desplegar un modal para ver más detalles como por ejemplo los ingredientes y los pasos a seguir de cada receta. También podemos acceder a un registro de las recetas existentes en el DAO a través del botón “Nueva Receta” . En esta vista se nos permite editar, crear y eliminar recetas a nuestro antojo.

4.5.- APARTADO EJERCICIOS

En esta vista el usuario puede navegar por el menú de navegación, ver la información de cada ejercicio e incluso añadirlos a favoritos. Además consta de un buscador de ejercicios el cual está implementado con HTMX que es nuestro trabajo dirigido. Al igual que con recetas, al pulsar el botón “Nuevo Ejercicio” accedemos al registro donde podemos realizar las operaciones CRUD con dichos ejercicios.

5.- MEMORIA DE ACTIVIDAD

5.1.- ITERACIÓN INICIAL

No tuvimos mucho código, añadimos imágenes de los integrantes del equipo, los diagramas, los storyboards, modificamos los archivos de entidades de creación y de código simplemente hicimos ficheros html básicos usando Bootstrap.

5.2.- ITERACIÓN 1

Añadimos mucho código, la página de inicio, la de información adicional y el primer login que hicimos antes de que nos lo explicarán mediante credenciales, además de una página de bienvenida y éxito. También implementamos los primeros beans que más tarde algunos de ellos serán eliminados y hojas de estilo para ir definiendo el front-end de la WebApp.

5.3.- ITERACIÓN 2

En esta iteración comenzamos a añadir FrameWorks tanto el de PrimeFaces como el de HTMX para nuestro trabajo dirigido. También asignamos a cada alumno su entidad de la que se haría cargo, Antonio la de ejercicios, David la de recetas y Hechun la de biometría. Por último empezamos a implementar lo básico de los modelos DAO de las entidades.

5.4.- ITERACIÓN 3

Aquí conseguimos tener acceso protegido a las distintas vistas ya sea por el rol de CLIENTE o el de ADMINISTRADOR y por fin implementamos el login final de la aplicación. Añadimos una pequeña versión demo para navegar como invitado y finalmente acabamos los modelos DAO para que el servicio haga de intermediario entre el DAO y el controlador. Los beans antiguos se eliminan y nos quedamos solo con el de fecha.

5.5.- ITERACIÓN 4

En esta iteración implementamos más información a la vista protegida del administrador, terminamos todos los métodos de las entidades para formar el CRUD y finalmente seguimos avanzando con el trabajo dirigido.

5.6.- ITERACIÓN 5

Por último cada integrante del equipo se encargó de hacer un archivo javascript con la validación de un formulario de su entidad, además de conectar con la API REST de cada entidad. Por último hemos aplicado VUE para hacer una lista de clientes que muestra la información sin necesidad de recargar la página.

6.- CONCLUSIONES

TercFit desde un principio la hemos creado para mejorar los aspectos del bienestar personal, desde el seguimiento biométrico, pasando por controlar los ejercicios que realizamos hasta el poder registrar recetas saludables, lo cual ofrece un valor añadido real para los usuarios interesados en su salud.

Plantear el proyecto usando metodologías ágiles como Kanban nos ha ayudado a realizar una gestión eficaz del trabajo en equipo. Además, asignar las tareas en distintas iteraciones nos ha ayudado a dosificar el trabajo y conseguir entender cada aspecto a tratar con mayor profundidad.

Hemos implementado los requisitos funcionales básicos que teníamos pensado, aunque con el desarrollo de la aplicación y debido a que debíamos de ajustar nuestra carga de trabajo a una realista, prescindimos de la creación de entrenamientos y de los ejercicios personalizados para simplificarlo en una lista de ejercicios favoritos. Por el resto, hemos aplicado autenticación por credenciales, CRUD de entidades clave (ejercicios, recetas, biometría), y uso de una API REST de forma satisfactoria.

Pensamos que el uso de herramientas como Vue, PrimeFaces, HTMX muestran un conocimiento sólido de herramientas actuales y buenas prácticas de desarrollo web, ayudan a obtener un estilo más limpio y funcional con un menor esfuerzo.

La memoria de actividad la hemos ido detallando para así mostrar la evolución del proyecto, desde simples páginas HTML hasta una aplicación funcional, con mejoras continuas en funcionalidades, seguridad, y experiencia de usuario.

La satisfacción del cliente también creemos que es buena ya que hemos añadido un modo de navegación como invitado, validación de formularios, una navegación intuitiva, acceso protegido a las vistas, y una interfaz clara con gráficos y carruseles.

7.- BIBLIOGRAFÍA Y RECURSOS ONLINE

HTMX (2025). *HTMX docs*. <https://htmx.org/docs/>

HTMX guía inicial. (2024) *Guía inicial*. <https://www.youtube.com/watch>

Bootstrap (2025). *Bootstrap examples*. <https://getbootstrap.com/docs/5.3/examples/>

PrimeFaces (2025). *PrimeFaces showcase*. <https://www.primefaces.org/showcase>

Vue.js (2025). *Vue.js examples*. <https://vuejs.org/examples/#hello-world>

DawClubServlet (2025). *GitHub jrbalsas*. <https://github.com/jrbalsas/dawClubServlet>

DawClubJSF (2025). *GitHub jrbalsas*. <https://github.com/jrbalsas/dawClubJSF>

DawClubAJAX (2025). *GitHub jrbalsas*. <https://github.com/jrbalsas/dawClubAJAX>

DawClubVUE (2025). *GitHub jrbalsas*. <https://github.com/jrbalsas/dawClubVue>

ChatGPT (2025). *GPT-4 de OpenAI*. <https://chatgpt.com/?model=auto>

Claude (2025). *Anthropic 3.7 Sonnet*. <https://claude.ai/new>