



AMERICAN KENPO

Amat Victoria Curam

PROYECTO DEPORTIVO

CLUB KENPO HIKEN

Alberto Bravo

-junio 2020-

INDICE

1. Introducción.....	2
2. Objetivos y problemas por solucionar.....	3
3. Planificación y seguimiento.....	4
1. Fase 1	4
2. Fase 2	4
3. Fase 3	5
4. Planificación general del proyecto.....	6
4. Guía de estilos.....	8
1. Paleta de colores	8
2. Texto principal.....	9
3. Títulos y subtítulos	9
4. Barras de navegación	9
5. Fondos	9
6. Tipografía	9
5. Diseño.....	10
1. Fase inicial	10
2. Fase evolutiva	12
6. Entorno de uso de la aplicación.....	15
7. Casos de uso.....	16
8. Aplicación de las tecnologías utilizadas.....	18
1. Tecnologías back end.....	18
2. Tecnologías front end	18
3. Base de datos	18
4. Librerías y recursos adicionales	18
9. Repositorio en GitHub.....	20
10. Creación de la base de datos y sus tablas	24
1. Tablas.....	24
2. Esquema Entidad-Relación	26
11. Mapa de la web	27
12. Estructura principal del código.....	28
1. Diagrama UML.....	28
2. Estructura organizativa	29
3. Secciones relevantes del código back end	31
13. Implementaciones futuras.....	41
14. Conclusiones.....	42
15. Bibliografía.....	43

1.Introducción

Este proyecto consiste en el desarrollo de una plataforma web destinada a un Club deportivo “Club deportivo Elemental Kenpo Hiken”.

Esta proyecto tiene una doble funcionalidad.

Por un lado, pretende dar a conocer el Club deportivo y este deporte a cualquier persona, de cualquier edad con ganas de practicar un arte marcial para la defensa personal.

Por otro lado, busca digitalizar y gestionar la administración de todos los procesos de inscripción necesarios para que un deportista pueda darse de alta y sea federado.

La web filtrará los accesos y permitirá a los diferentes tipos de usuarios, deportistas o instructores, acceder a las diferentes funcionalidades dependiendo de los privilegios que tengan concedidos.

Al ser un entorno web, los usuarios podrán utilizarla y navegar por la misma sin necesidad de instalar ningún software adicional. Como únicos requisitos necesitarán un navegador web, una conexión de internet y acceso a un disco duro (interno o externo) para descargar los formularios que cumplimenten.

2. Objetivos y problemas por solucionar

El motivo por el cual se propone esta plataforma web es para cubrir una necesidad inherente en el club que ralentiza y burocratiza innecesariamente el mantenimiento y el crecimiento de la escuela.

El C.D.E. Kenpo Hiken tiene un flujo de deportistas que varía cada mes ya que constantemente se dan procesos de alta y de baja. Con cada nuevo o nueva deportista se debe iniciar un proceso administrativo que actualmente se realiza manualmente y que implica el manejo de un variado número de documentos y pasos que van desde el alta del deportista hasta su federación, pasando por la firma/aceptación de diferentes documentos para la cesión y el tratamiento de datos e imágenes.

El origen de los documentos es diverso. Los documentos se reciben en papel, por email, WhatsApp, escaneados, fotografiados y a veces son manuscritos ininteligibles son difíciles de entender...etc.

Todo lo anterior se ve incrementado al inicio de cada temporada en la que nuevamente se debe dar de alta y federar a cada deportista ya que la vinculación es anual.

Con el inicio de la pandemia se eliminó el uso del papel para reducir los contactos, lo cual centralizó de forma digital los documentos, pero con la flexibilidad de las restricciones se volvió el problema original ya que aunque los documentos se podían seguir enviando telemáticamente (vía email o WhatsApp) volvieron a empezar a enviarse en papel. Cada documentación volvió a enviarse en formatos muy dispares, con poca calidad en algunos casos y además la privacidad la gestión de los documentos tampoco era la adecuada.

Hoy en día se combinan ambos sistemas.

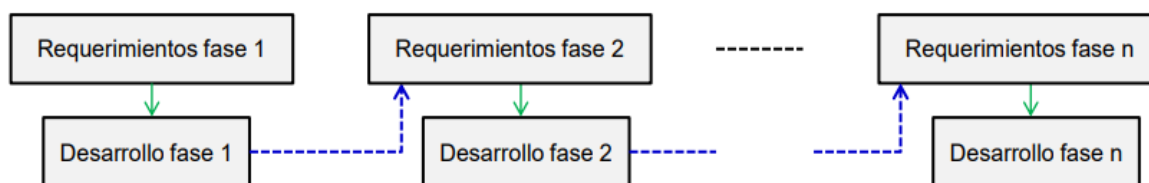
En ese momento se volvió a tener la conciencia de que el proceso no era óptimo ni adecuado y que se desaprovechaban oportunidades digitales que estaban al alcance del Club deportivo y es en ese momento en el que se empezó a gestar la idea de este proyecto.

Por todo lo expuesto, aunque con la idea en una fase muy preliminar, se propuso al Club deportivo implementar esta web. Se les explicó que se buscaría optimizar todos estos procesos permitiendo a cada deportista, tras darse de alta, el acceso individualizado a una web donde gestionar de un modo personalizado todos sus documentos, y claro está solicitar también su baja como deportista en el Club. A su vez la idea permitiría que el Club pudiera acceder a los datos generales y gestionar desde la web todos los documentos que le son necesarios para poder, con cada temporada y cada nuevo o nueva deportista, realizar las federaciones y así optimizar los procesos de forma segura, eficiente y minimizando los posibles errores que se producen por la gestión manual de tantos documentos.

3. Planificación y seguimiento

Una vez propuesta la idea se realizó una reunión inicial con el Club Deportivo para iniciar con la fase de análisis que ayudaría a definir las necesidades del club y el objetivo a buscar.

Ya que el Club tenía Claro el objetivo a lograr, pero no lo que necesitaba y los medios para conseguirlo, se ha optado por un desarrollo por etapas siguiendo un **Modelo de Desarrollo Evolutivo** al ser el más conveniente y ya que además la implementación inicial no tiene coste y se ha limitado inicialmente **a 3 fases**.



1. Modelo de Desarrollo Evolutivo

1. Fase 1

En la fase inicial se determinaron modelos fijos de estilo y colores vinculados a la marca corporativa del Club (sección Guía de Estilos). También se marcó un objetivo principal que fue el de priorizar la gestión de los formularios frente a las necesidades de los deportistas. El motivo, se busca inicialmente que sea un beneficio para los administradores del Club.

Se aprovecha para hablar con el Club al respecto de la fase de producción y alojamiento de la web. Pero el Club no tiene claro cómo quiere hacerlo ya que disponen de un alojamiento con la empresa IONOS, pero quieren cancelar la colaboración por lo que de momento no sabe en qué hosting quieren que se aloje la web. Hasta la decisión final el despliegue se realizará en un entorno local.

2. Fase 2

Tras la fase 1 se procedió a la realización de la maquetación inicial de modo que se pudiera revisar en esta segunda fase el cumplimiento de objetivos y si se seguían las pautas de diseño acordadas.

Quedó aprobado por el Club el diseño realizado y se inició con el proceso de desarrollo de la web para en la tercera y última revisión se pudieran comprobar los aspectos funcionales de la web.

- *Landing page básica.*
- *Incorporación de todas las redes sociales del Club.*
- *Registro básico de usuarios (usuario y contraseña).*
- *Funcionalidad para la cumplimentación de un único formulario de la ficha del deportista con la inclusión de la aceptación de condiciones y tratamiento e imágenes en un mismo formulario. Generación y envío del formulario por email al Club.*

- Los formularios se generarían en formato *.pdf.
- Sección de administrador con la posibilidad de obtener un listado actualizado de alumnos (usuarios) con cada nuevo registro.

3. Fase 3

En esta fase quedaron aprobadas las funcionalidades de la fase previa, pero viendo que, tras la revisión de objetivos estos se habían cumplido, técnicamente se abrió la posibilidad de una mejora en la funcionalidad de la web.

Ya que los objetivos que necesitaba la administración del Club estaban cubiertos, las nuevas mejoras irían destinadas a los usuarios. En esta etapa se buscaría proporcionar más información a los deportistas actuales y potenciales y hacer la web más fácil de utilizar para los usuarios. Por tanto se procedió a probar una implementación que incluiría los siguientes aspectos.

- *Landing page con información sobre el Kenpo como deporte y del Club como punto de encuentro.*
Creación de secciones informativas sobre horarios, precios y localizaciones.
- *Para hacer más dinámico y cercano el contacto entre los deportistas y el Club, se incluiría un acceso a WhatsApp en diferentes secciones. De esta forma se solventarían dudas de forma rápida, se podrían atender bajas o modificaciones de datos de forma más dinámica y se facilitaría la captación de deportistas potenciales.*
- *Viendo que la captación de datos de la ficha podría ayudar a la formalización de los demás formularios del Club decidió realizar un cambio el modelo de registro de usuarios.*
En la nueva funcionalidad, al tiempo que se registraba la cuenta del deportista se le solicitarían los personales más importantes y estos se almacenarían en la base de datos del Club.
- *Se crearían 2 formularios nuevos para tener un total de 3 formularios que se auto rellenarían al acceder cada usuario a su cuenta. Esto presentaría 2 ventajas,*
 - *Los usuarios tendrían que cumplimentar muchos menos datos haciendo el proceso mucho más sencillo para ellos.*
 - *El uso de los datos grabados y el autocompletado de los formularios permitiría al club tener 3 formularios separados en lugar de uno sólo. Esto atendería a la realidad ya que físicamente se trabaja con 3 formularios en lugar del inicial que aglutinaba toda la información de los 3.*
 - *Ficha del deportista.*
 - *Autorización para el tratamiento de datos e imágenes.*
 - *Mandato de autorización para registrar a los alumnos en la Federación Española de Karate en su nombre.*
(este documento además podría enviarse a la misma federación de forma individualizada) como se hace hasta ahora.

- *En lugar de enviar los formularios por email. Estos se generarían en tiempo real en formato *.pdf así se permitirá al usuario su descarga.*
- *En la sección del administrador junto al listado se creará una función para que se pueda acceder al formulario de cada uno de los deportistas y que se podría utilizar la función de generación del archivo *.pdf en tiempo real.*

El envío por email implicaría que se tendrían que ir revisando los email uno a uno y descargando los ficheros no siendo un proceso muy práctico. Del modo propuesto, a la hora de proceder con las federaciones sólo habría que acceder a la web para encontrar todos los ficheros de los deportistas a mano y ordenados, sugerido por el director técnico del Club, por apellido.

- *Se ve conveniente la inclusión de una sección que pueda contener la política del club y la política de tratamiento de datos acorde a la legalidad vigente.*
- *Respecto a la fase de producción del Club no tiene claro cómo quiere hacerlo. Disponen de un alojamiento con la empresa IONOS, pero quieren cancelar la colaboración por lo que de momento no sabe en qué hosting quieren que se aloje la web.*

4. Planificación general del proyecto

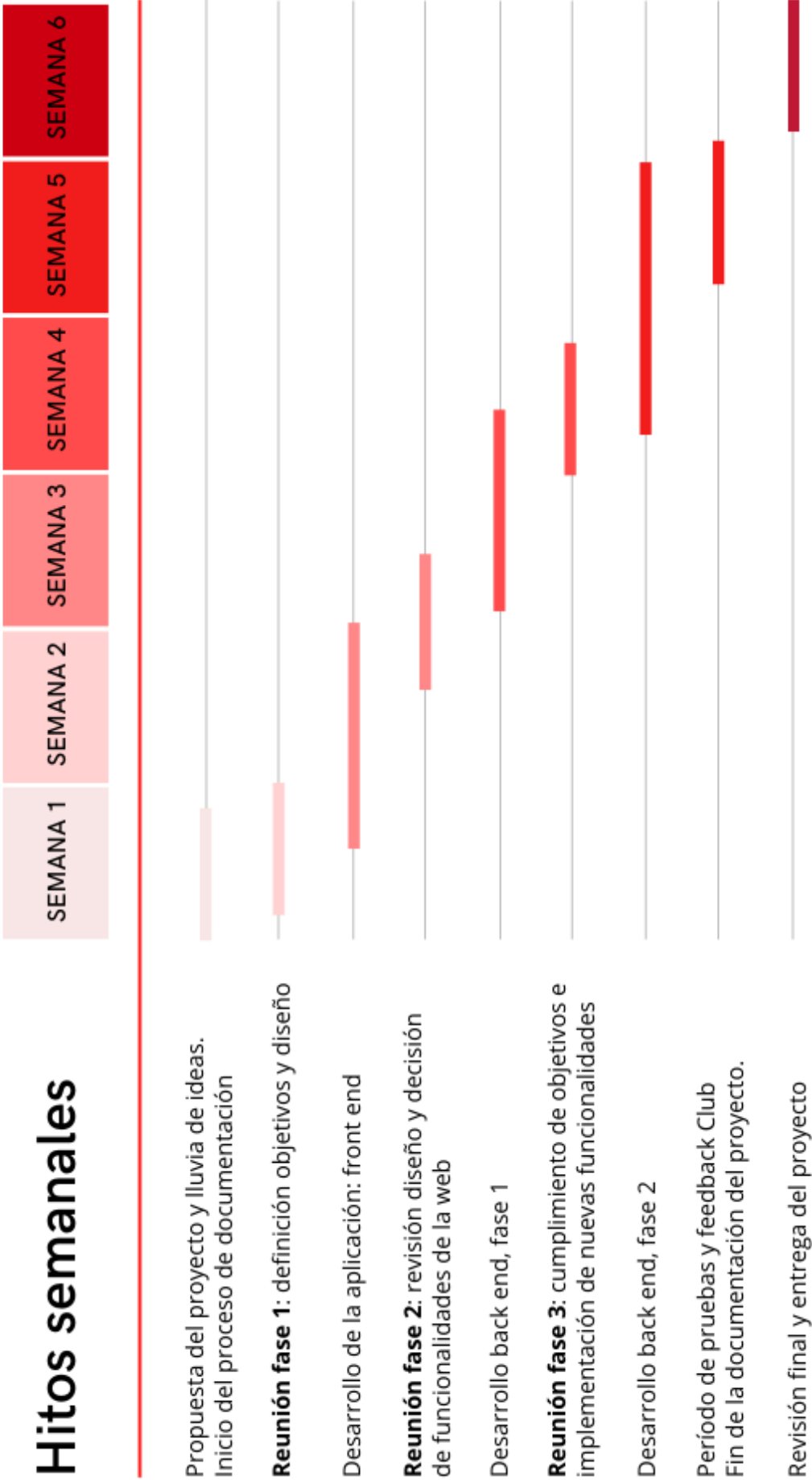
Desde el momento que se propuso el proyecto se empezó con su planificación general de cara a la puntual entrega de este a efectos académicos y del Club. Se contaban con algo más de 6 semanas para la realización completa del proyecto, pautándose la fecha final del mismo el día 13 de junio del 2022. Se empezó a trabajar por cumplimiento de hitos semanales y se planificaron las reuniones con el Club en semanas clave que podrían modificarse según evolucionará el desarrollo de este.

La planificación general, el seguimiento del proyecto y las reuniones quedan registrados en las actas del Club y en el siguiente diagrama de Gantt.



PLANIFICACIÓN

Proyecto Club Deportivo Kenpo Hiken



2. Diagrama de Gantt_Planificación General

4. Guía de estilos

Según se acordó en la reunión inicial y fase 1 del desarrollo del proyecto. El estilo de la web debía seguir los colores de la marca que están reflejados el logotipo del Club Deportivo. Con estas consideraciones quedaron definidos los siguientes parámetros a la hora de realizar la parte visual y el front end de la web,

1. Paleta de colores

Considerando que la identidad de una marca suele tener entre 4 y 5 colores y considerando los colores del logotipo, se ha decidido que debe utilizarse la siguiente paleta de colores.



3. Paleta de colores

2. Texto principal

Se utilizará el color negro (#1C2622, RGB 28, 38, 34)

3. Títulos y subtítulos

Se utilizarán las 2 variedades del color rojo,

(#BF1725, RGB 191, 23, 37)

(#BF1736, RGB 191, 23, 54)

4. Barras de navegación

Se utilizarán tanto el color negro como el color gris,

(#1C2622, RGB 28, 38, 34)

(#F2F2F2, RGB 242, 242, 242)

5. Fondos

Se utilizará el color blanco (#FCFEFF, RGB 252, 254, 255)

6. Tipografía

(sujeta a los diferentes soportes visuales)

Títulos y encabezados

Roboto, 19.5px

Subtítulos

Roboto, 14px

Texto general

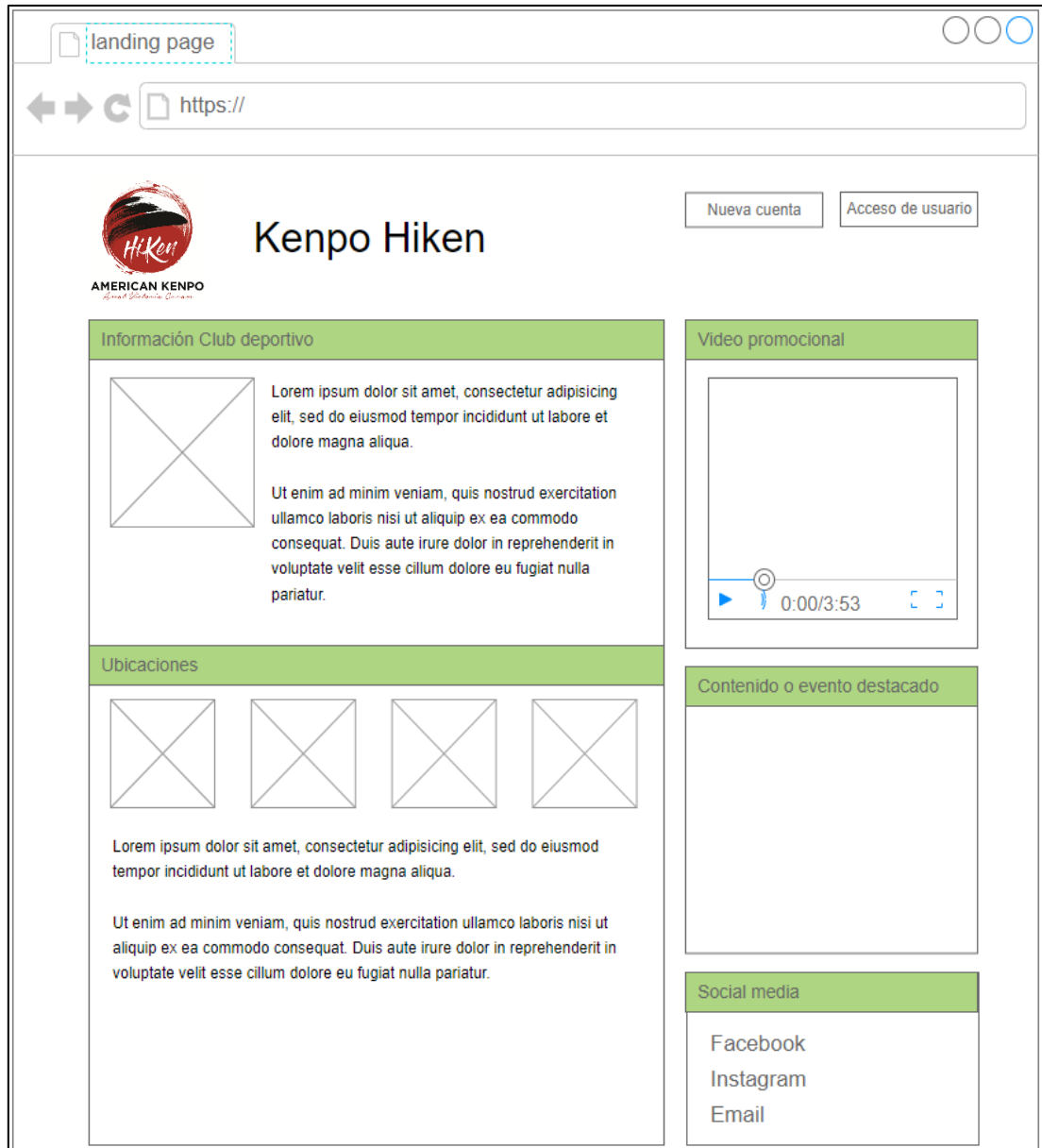
Roboto, 13px

5. Diseño

A continuación se muestra la evolución preliminar y evolutiva de la parte del diseño front and de la web.

1. Fase inicial

En una primera etapa se buscaba que el diseño y la funcionalidad de la web fueran más relevantes para el club que para los deportistas. Se propuso el siguiente diseño básico.



4. Página de inicio preliminar

Esta página principal enlazará con otras webs relevantes para los deportistas y el Club así como con las redes sociales del mismo.

A través de la página principal se podrá acceder a las cuentas de usuario o se podrá crear una nueva cuenta.

Formulario de creación de cuenta

Nueva cuenta de usuario

Crea tu cuenta a continuación.

Si ya tienes una cuenta [inicia sesión, aquí](#)

Nombre de usuario

johndoe

Email:

inbox@email.com

Contraseña

☐ Aceptación de las condiciones

Crear cuenta

5. Creación de una cuenta de usuario

Formulario de acceso

Bienvenid@

Ingresa tus datos de acceso

Nombre de usuario

johndoe

Contraseña

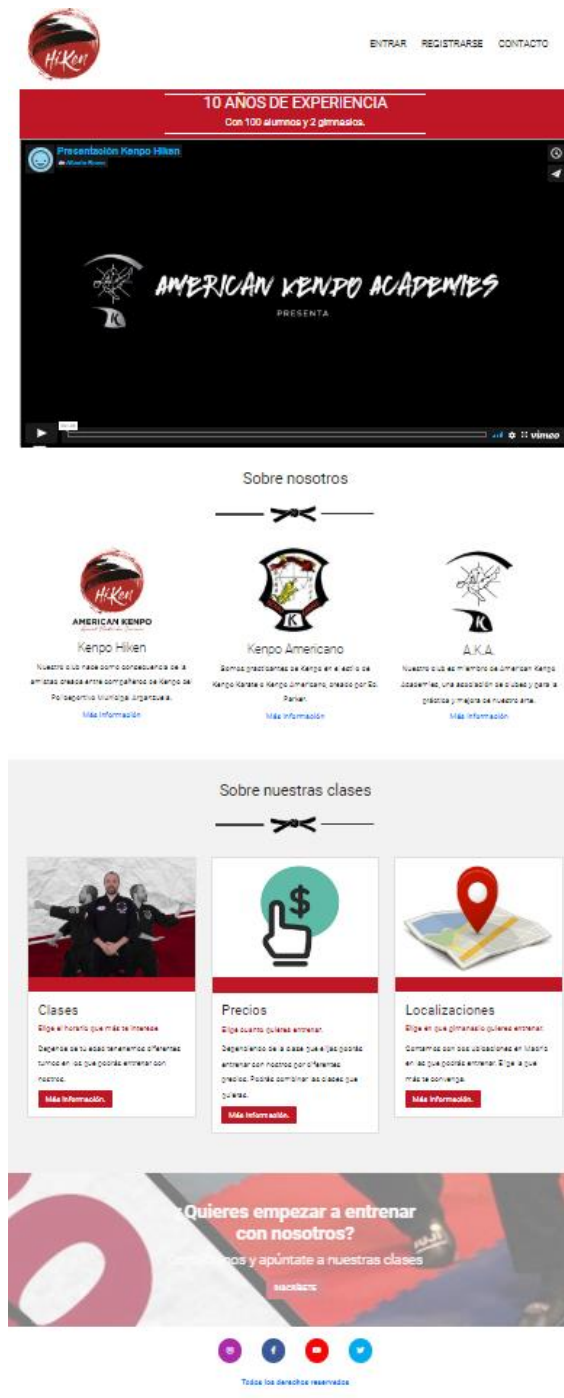
Entrar

6. Acceso de un usuario existente

2. Fase evolutiva


Entre la fase 1 en la que se definen los aspectos básicos de diseño y la fase 3 ya aprobada la maquetación y definidas las nuevas funcionalidades, el cambio visual es muy significativo. La fase de diseño se considera finalizada a excepción de cambios menores y la identidad del Club como marca queda definida.

El usuario tiene un entorno visual intuitivo y fácil de utilizar. Se definen los formularios y la parte del administrador. Se incluyen videos, aplicaciones importantes como WhatsApp y páginas informativas para los usuarios como precios, horarios y Localizaciones.



7. Diseño final de la web - Landing Page

Regístrate



Introduce los datos con los se accederá a la cuenta de usuario

La contraseña debe tener entre 6 y 8 caracteres

Todos los campos son obligatorios

Nombre de usuario

Contraseña de acceso

Introduce ahora los siguientes datos del **deportista** para que podamos crear su ficha.

Todos los campos son obligatorios

Nombre

Primer apellidoSegundo apellido

Tipo de documentoNúmero de documentoFecha de nacimiento

Lugar de nacimientoNacionalidad

Dirección

CiudadProvinciaCódigo Postal

TeléfonoE-mail

¿Padece usted alguna enfermedad o trastorno que impida o dificulte la práctica de kenpo?

En caso afirmativo, indique cuales

8. Fase de registro y recolección de datos personales

Con el objetivo de hacer más rápido el proceso de inscripción, en la fase de registro se creará el usuario y al mismo tiempo se grabarán los datos personales del deportista, los cuales serán utilizados para el autocompletado de todos los formularios que encontrará el usuario en su cuenta. Esto minimizará el tiempo que cada deportista necesita invertir en cumplimentarlos.

El acceso al usuario es simple y sirve de filtro para la entrada de cualquier usuario, ya sea un deportista o el administrador.

Inicio de sesión



Usuario

Contraseña

Acceder

9. Inicio de sesión para deportistas y administrador

Desde la cuenta del usuario, este puede cumplimentar y descargar todos o alguno de los formularios necesarios para que el Club pueda registrarles.

Formularios temporada 22/23 ▾

Ficha Deportiva Club Kenpo Hiken
Cláusula - Protección de Datos < 18 años
Mandato inscripción

10. Acceso a los formularios desde la cuenta de usuario

El administrador también podrá gestionar todos los formularios, descargar un listado actualizado automáticamente y dar de baja cuentas de deportistas. También podrá dirigirse a la cuenta bancaria del Club y a la web de la Federación Nacional de Karate para procesar las federaciones.

Listado de deportistas						
Nombre	Apellido.1º	Apellido.2º	Ficha	Imágenes	Mandato	Eliminar
Alberto	Bravo	García	PDF	PDF	PDF	X
Antonio	Casares	López	PDF	PDF	PDF	X
Pedro	García	Esteban	PDF	PDF	PDF	X
Temporada 2021/2022						
Listado de alumnos						
						

11. Acceso a los formularios desde la sección del administrador. Listado y gestión de cuentas

6. Entorno de uso de la aplicación

La web, por medio de la página de inicio de sesión, permitirá el acceso a su contenido a diferentes tipos de usuario.

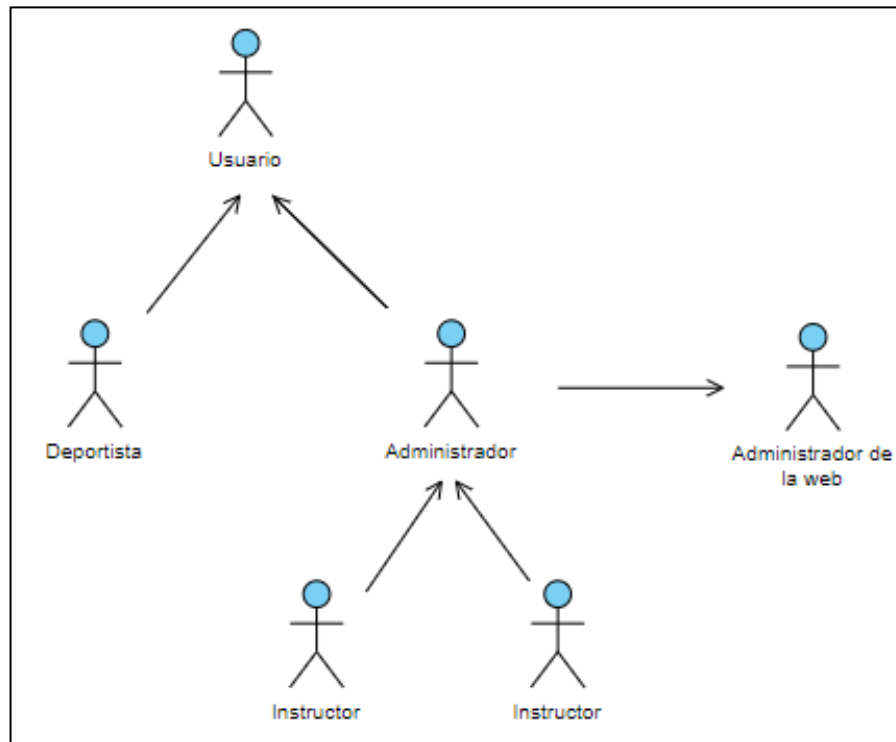
Usuarios anónimos – quienes podrán acceder a la web y ver diferentes contenidos y explicaciones referentes al Club Deportivo sin comprometerse a nada con el Club deportivo.

Usuarios registrados – quienes podrán acceder, tras registrarse y por medio de una autenticación segura a su cuenta y datos y a diferentes secciones de la web, aunque eso dependerá de los permisos que tenga cada usuario.

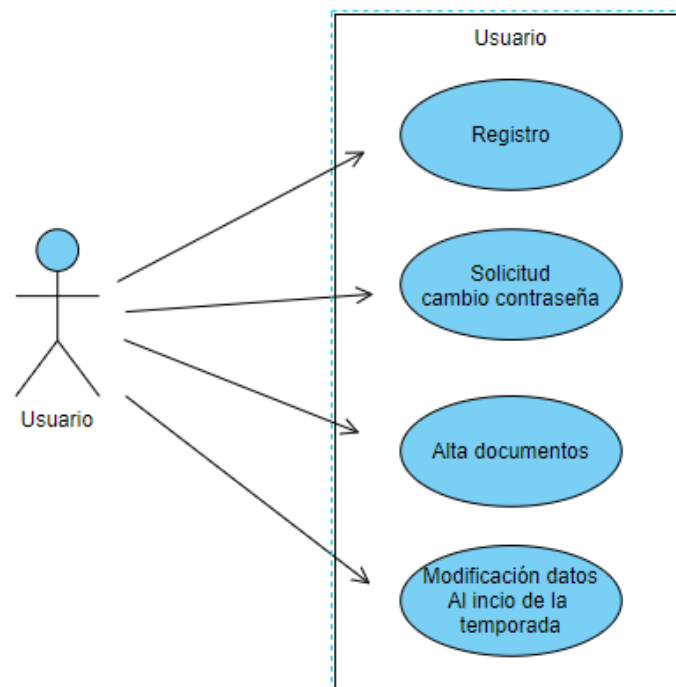
- **Deportistas**, *podrán acceder a sus datos como deportistas y podrán ver y editar los mismos (según cada caso y el momento de la temporada). También podrán cumplimentar los formularios necesarios para inscribirse y descargarlos en el momento que deseen y tantas veces como lo deseen. También podrán visualizar y estar al tanto de otras posibles documentaciones o circulares que al Club Deportivo le interese trasladar a los deportistas.*
- **Administrador (Instructores)**, *podrán acceder a la web como administradores, aunque se podrá delegar la función de administrador a uno sólo de los instructores (es algo que puede decidir el Club deportivo en cualquier momento incluso ya en fase de producción). El acceso como administrador les permitirá acceder a un listado de deportistas y a sus datos personales sin que tengan que elaborarse manualmente con cada nuevo deportista. Como función principal, podrán gestionar todos los documentos de los deportistas. Descargarlos y trabajar con ellos de forma digital para el alta y federación de cada deportista. Como otras funciones podrá gestionar las cuentas y dar de baja a los deportistas que lo soliciten. Las actualizaciones de las fichas se podrán realizar cada inicio de temporada y los cambios de contraseña se solicitarán por el administrador al administrador (desarrollador) de la web.*
- **Administrador de la web (desarrollador)**, *es miembro del Club y de la directiva y tendrá acceso total a la web y a la gestión de los documentos de los deportistas al igual que los instructores. Podrá realizar cambios en la web y en la base de datos y podrá modificar datos de deportistas en el transcurso de la temporada así como modificar contraseñas.*

A continuación se muestra un esquema de los casos de uso definidos para cada usuario.

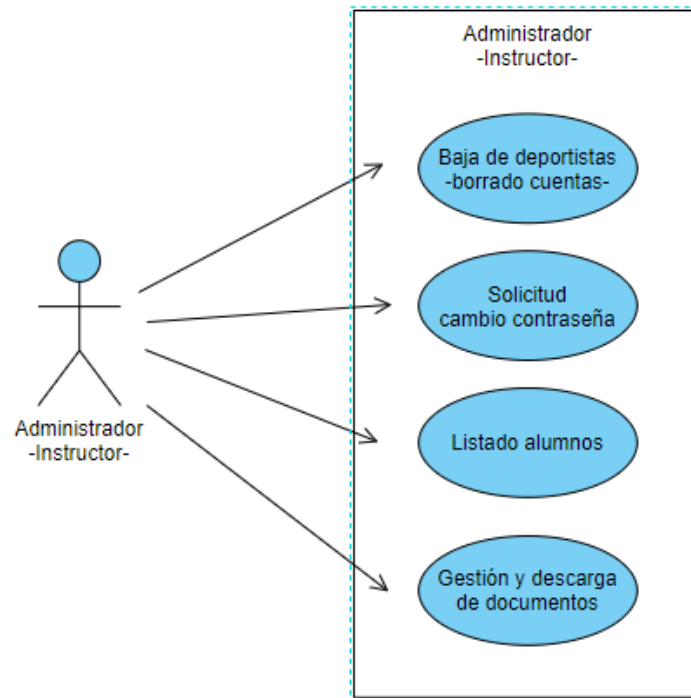
7. Casos de uso



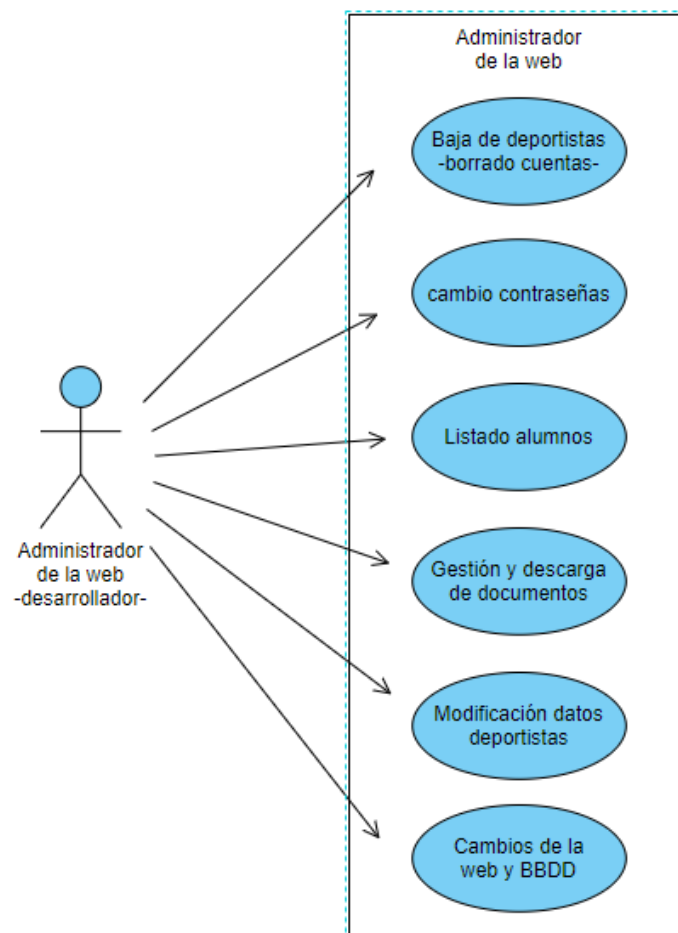
12. Casos de uso



13. Rol del usuario



14. Rol del administrador (instructor)



15. Rol del administrador de la web

8. Aplicación de las tecnologías utilizadas

1. Tecnologías back end

La plataforma web se desarrollará en su mayor parte utilizando PHP como lenguaje del servidor incorporando JavaScript. Aunque es posible utilizar JavaScript desde el lado del servidor. Para este proyecto sólo se ha utilizado desde el lado del cliente.

El motivo de la elección es que el uso de ambos es gratuito de modo que no se repercutirá ningún coste al Club Deportivo.

Pero ese no es el único motivo. La prioridad a la hora de usar PHP es que su interacción con HTML es muy buena y permite crear webs de un modo sencillo. Además es fiable y eficiente además que es muy compatible si se quiere acceder a software de terceros como puedan ser APIs o MySQL.

La elección de JavaScript es clara. Es perfecto para añadir características interactivas a la web. Es uno de los lenguajes más utilizado por lo que es plenamente compatible con otros lenguajes y por la mayoría de los navegadores y dispositivos actuales. Es rápido y sencillo de utilizar.

2. Tecnologías front end

La parte Font-end se realizará utilizando HTML5 y CSS. se ha elegido HTML5 ya que permite trabajar de un modo limpio y ordenado. Es plenamente compatible con cualquier navegador, móviles actuales e incluso con plataformas como Google Map además de ofrecer funcionalidades de video y audio, aspectos necesarios ya que este proyecto integra videos y el uso de mapas.

CSS es inequívocamente indispensable para poder crear estilos personalizados o modificar clases ya definidas por otras bibliotecas. Se ha complementado su uso con Bootstrap (se mencionará más adelante).

3. Base de datos

Para el motor de la base de datos se utilizará MySQL ya que es un software gratuito bajo licencia GPL muy utilizado, es multiplataforma, por lo que ofrecerá compatibilidad para los sistemas operativos más utilizados y además porque su mantenimiento es sencillo y se puede mantener por el mismo desarrollador. Otro aspecto por el que se ha elegido es debido a su rapidez

4. Librerías y recursos adicionales

Para este proyecto se ha utilizado otros recursos y librerías para poder realizar un mejor desarrollo y un mejor y más atractivo diseño de la parte visual de la web.

Bootstrap – Se ha utilizado ya que su amplia cantidad de clases abre un abanico muy extenso de funcionalidades que hacen que una web sea muy atractiva. Además con Bootstrap es sencillo hacer que la web sea responsive.

Para este proyecto Bootstrap se ha utilizado ampliamente para la generación de menús de navegación responsive, para la creación de formularios, así como para los estilos de botones y modales (haciendo la web más dinámica).

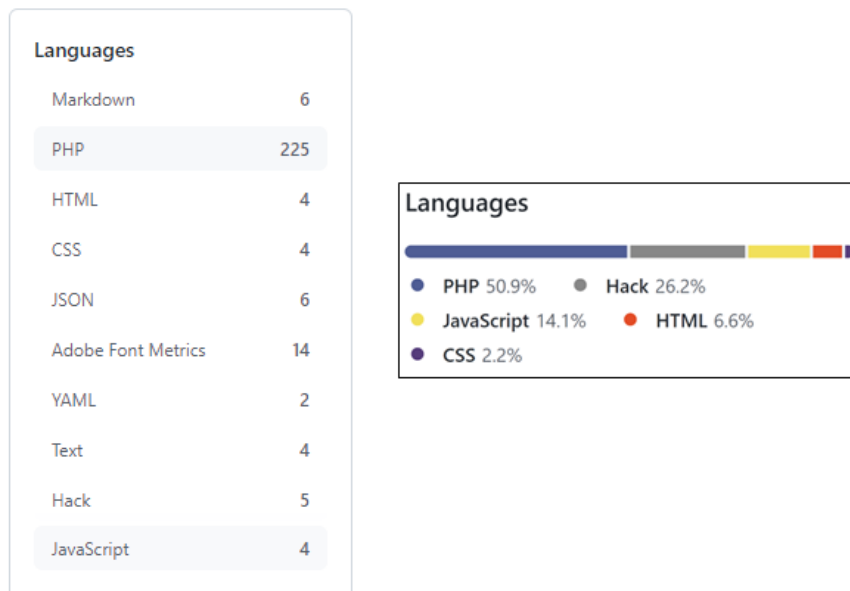
Ajax – se ha elegido su uso por el uso de formularios y la necesidad de validación de datos. Además pq ha permitido el intercambio de información de forma rápida sin necesidad de recargar la página (importante por el uso que se ha dado a las sesiones durante el proyecto), generando en los usuarios una sensación de carga instantánea a la que suelen estar acostumbrados.

jQuery – es una librería multiplataforma de JavaScript. Permite interactuar con el código HTML de una forma muy eficaz y dinámica. Se ha utilizado ampliamente en este proyecto para manejar eventos y facilitar el uso de JavaScript. Sus sentencias son más cortas e intuitivas que las de JavaScript a la hora de implementarlas. Además se ha utilizado combinado con Ajax a la hora de enviar información entre cliente y servidor.

DOMPDF– A la hora de generar los formularios de la web en formato *.pdf se consideraron diferentes opciones como HTML2PDF o FPDF pero se descartaron y se optó por el uso de DOMPDF al ser una librería con mucha documentación, frecuentemente actualizada y una buena opción para el diseño de documentos usando tablas.

La generación de los documentos se ha realizado combinando PHP y HTML. Los documentos se general directamente a partir del mismo código HTML lo cual permite realizar diseños muy cuidados y precisos a los originales como era el caso de este proyecto en el que ya se contaba con diseños preliminares y se necesitaban emular del modo más preciso posible.

Tras subir el proyecto a GitHub, si se revisa la sección de la codificación se puede ver que GitHub analiza el código y realiza un análisis porcentual de los lenguajes utilizados. Es un dato curioso si se quiere conocer cómo se ha contraído un proyecto.



16. Tecnologías detectadas por GitHub

9. Repositorio en GitHub

A la hora de gestionar el proyecto y para poder tener un seguimiento frecuente de los cambios que se realizan mientras se desarrolla se ha creado una cuenta en GitHub.

El usuario de esta cuenta de GitHub se ha llamado BravoAlberto y se ha vinculado a una de las cuentas del Club Deportivo proyectohiken@gmail.com creada a estos efectos. En el futuro será esta cuenta de correo la que reciba las comunicaciones y emails de la fase de testeo del proyecto.


El repositorio creado para el proyecto se llama KenpoHiken.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *

Repository name *


 BravoAlberto ▾


/ KenpoHiken ✓

Great repository names are short and memorable. Need inspiration? How about [friendly-eureka?](#)

Description (optional)

Proyecto de digitalización del Club Deportivo Kenpo Hiken

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.


☐ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: None ▾


Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

License: None ▾

 You are creating a public repository in your personal account.

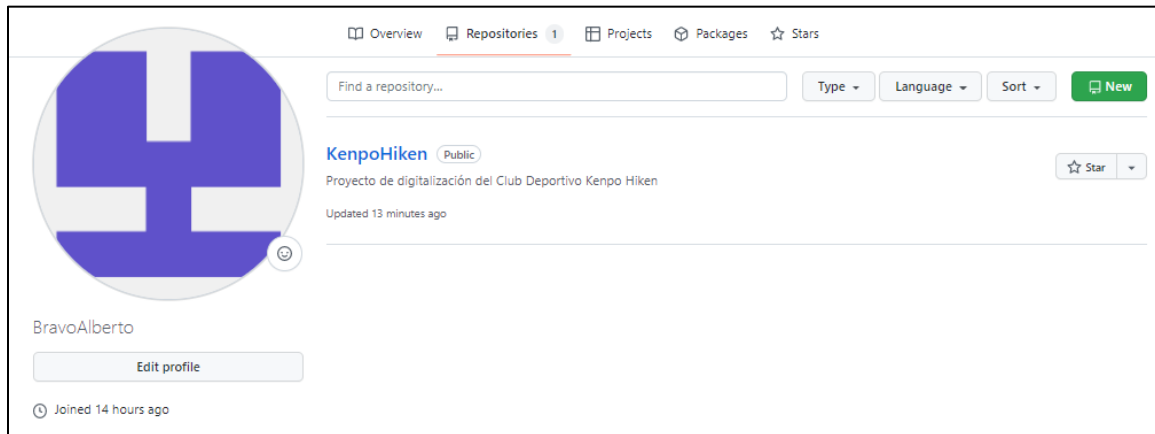
Create repository

Quick setup — if you've done this kind of thing before

 Set up in Desktop or **HTTPS** **SSH** <https://github.com/BravoAlberto/KenpoHiken.git>

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

18 Ruta del repositorio



19 Repositorio creado

Para este proyecto se ha decidido utilizar un equipo local con Windows 10 de modo que para gestionar el repositorio se ha utilizado Git Bash.

```
MINGW64:/c:/xampp/htdocs/pruebas/KenpoHiken

bravo@ThinkPad MINGW64 /c:/xampp/htdocs/pruebas/KenpoHiken
$ git init
Initialized empty Git repository in C:/xampp/htdocs/pruebas/KenpoHiken/.git/

bravo@ThinkPad MINGW64 /c:/xampp/htdocs/pruebas/KenpoHiken (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        .DS_Store
        Clases.html
        contacto.html
        css/
        funciones/
        img/
        incio.php
        index.php
        localizaciones.html
        precios.html
        registro.php
        tablas.sql
        template.html

nothing added to commit but untracked files present (use "git add" to track)
```

20 Creación del repositorio local y primer status

Proyecto Club Deportivo Kenpo Hiken

Al ser la primera subida del proyecto al repositorio se ha utilizado **git add** . y posteriormente se ha realizado el primer commit y sincronizado con GitHub haciendo un push, **push -u origin main**

```
bravo@ThinkPad MINGW64 /c/xampp/htdocs/pruebas/KenpoHiken (main)
$ git push -u origin main
Enumerating objects: 46, done.
Counting objects: 100% (46/46), done.
Delta compression using up to 8 threads
Compressing objects: 100% (46/46), done.
Writing objects: 100% (46/46), 3.11 MiB | 4.54 MiB/s, done.
Total 46 (delta 11), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (11/11), done.
To https://github.com/BravoAlberto/KenpoHiken.git
* [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
```

21 Primer push del equipo local a GitHub

BravoAlberto / KenpoHiken Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 1 branch 0 tags

Go to file Add file Code

bravoga1977 Se sube el proyecto 2337ebb 24 minutes ago 1 commit

css	Se sube el proyecto	24 minutes ago
funciones	Se sube el proyecto	24 minutes ago
img	Se sube el proyecto	24 minutes ago
.DS_Store	Se sube el proyecto	24 minutes ago
Clases.html	Se sube el proyecto	24 minutes ago
contacto.html	Se sube el proyecto	24 minutes ago
inicio.php	Se sube el proyecto	24 minutes ago
index.php	Se sube el proyecto	24 minutes ago
localizaciones.html	Se sube el proyecto	24 minutes ago
precios.html	Se sube el proyecto	24 minutes ago
registro.php	Se sube el proyecto	24 minutes ago
tablas.sql	Se sube el proyecto	24 minutes ago
template.html	Se sube el proyecto	24 minutes ago

Help people interested in this repository understand your project by adding a README. Add a README

About

Proyecto de digitalización del Club Deportivo Kenpo Hiken

0 stars
1 watching
0 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Languages

PHP 34.7% HTML 29.8%
Hack 20.1% JavaScript 8.9%
CSS 6.5%

22 Primer commit recibido y sincronizado en GitHub

Al momento de finalizar los últimos detalles de esta la documentación y estando el proyecto casi acabado se concluye que durante 23 días, desde la creación del repositorio, se han realizado un total de 58 commits.

```
bravo@ThinkPad MINGW64 /c/xampp/htdocs/pruebas/KenpoHiken (main)
$ git log --oneline
ebc909b (HEAD -> main, origin/main) Actualización código 05.06.2022
16d9ab2 Update codigo_03.05.2022
e037599 Revisión de la clase filtrado()
31e9751 problema footer arreglado
82360e7 problame footer arreglado
8a32b21 problame footer arreglado
1e7ba57 Ordenación apellido y enlaces a FMK y banco
76c7b63 Solucionado problema con el collapse del nav
c3c10dd Solucionado problema con el collapse del nav
694a806 Añado filtro DNI
c599198 Añado filtro DNI, CP, Mail, Telefono
09f6514 Nueva clase para centrado de menu dropdown
acfa734 Listas con ajustes de ordenación y diseño
9bccf4d Ruta de PDF establecida para descarga local
0e261de Fin codigo de formularios a PDF
b1f8482 Actualización codigo 28.05.2022
b1b0f02 Preparando acceso y consulta para gestion cuentas
5020024 Generación PDF con datos desde variable_Un no OK
7b12ad5 Update codigo_27.05.2022
4e547f5 Sello y firma para PDFs
1086e25 Inserción de variables para obtención de fecha sistema (JS)
4adabe3 Finalizados los formularios a PDF y cambio ruta
b1bab2e Finalizados los formularios a PDF
1e47b7f Ajuste del foter y control acceso administrador.php
0761788 Incorporadas imágenes para docs. PDF
80af223 INCorporado contedinodo de DCOMPDF (generador de PDF)
72c33a2 Creado fichero en PDF (con DCOMPDF) para la ficha
883c099 Codigo actualizado a 25.05.2022
2892300 Se incorporan 2 nuevas imágenes
54981ea Se incorporan las funcionalidades para general PDFs
ff424cc Se crean funciones para traer y enviar a formularios.php
699393f Se crean las 2 tablas de tutores
b0b65c6 actualizaciones menores y revisiones de código
72b7cc0 acceso por usuario y campos rellenos desde query
11c6ba5 cambio de contacto de html a php y funcional con envío de datos
26e8660 página correcta y funcional con denegación de acceso
d186a9a Comienzo del back para traer campos de la BBDD a front
b29c78f Separación de campos de nombre y apellidos
78669aa Actualización con revisión de ids
a43e9ca Actualización columnas y nueva tabla tutor
a2e67ea Nueva página redireccionamiento de acceso denegado
91211c6 Actualización documentación con datos tablas BBDD
1fe134a Actualización de códigos PHP y JS
e443f30 Actualización y corrección errores
6463d5a Creación pivacidad y protección de datos con enlace a paginas
921ad9f Actualización del código
d1a9a3e Actualización formularios y modales
1b1a563 Creación página para el administrador
812666a Actualización CSS y Formularios
97c9a16 update codigo registro&contacto. Creación pagina formularios
8659389 Update de registro y spinners
f539b38 Actualizada documentación_18.05.22
3e48332 Bubble whatsapp con cambios en contacto y documentación
cfa7675 Actualización documentación_BBDD
e525df6 Update codigo
ff2c5f3 Se sube la documentación activa del proyecto
2337ebb (origin, master) Se sube el proyecto

bravo@ThinkPad MINGW64 /c/xampp/htdocs/pruebas/KenpoHiken (main)
$
```

23.git log del proyecto

10. Creación de la base de datos y sus tablas

Para el funcionamiento correcto del proyecto es necesario que tanto los datos de acceso a las cuentas de los usuarios (deportistas) así como todos los datos personales de cada uno de ellos, los cuales constituirán el contenido de cada uno de los formularios, se almacenen en una base de datos. Se ha decidido que el uso más adecuado es el de una base de datos con modelo relacional ya que se necesitará acceso a datos relacionados entre sí. Es por ello por lo que se utiliza MySQL y para tal propósito se crea la base de datos llamada kenpohiken y un usuario administrador para las tareas básicas de mantenimiento y gestión de la base de datos.

```
Grants for administrador@localhost
-----
GRANT USAGE ON *.* TO `administrador`@`localhost` IDENTIFIED BY PASSWORD '*6776DB585830298C5D2E53098D94EE790B12675E'
GRANT SELECT, INSERT, UPDATE, DELETE ON `kenpohiken`.* TO `administrador`@`localhost`
```

24. Creación del usuario limitado administrador

1. Tablas

La primera tabla que se crea es la de usuario. La tabla contiene dos columnas, una para el usuario (*Primary key*) y una segunda columna para la contraseña. Por seguridad la contraseña queda hasheada al momento de grabar el valor en la tabla.

usuario	clave
Alberto	\$2y\$15\$JgGrQIFVRvEtLRYa9cu4X.b31YFP14BF/106.0jKm7h4pmMjvLc9K

25. Muestra del "hasheo" de la contraseña de un usuario

Ambas columnas son obligatorias y así se ha configurado no permitiendo que haya valores nulos.

El objetivo de esta tabla es la de permitir que cada usuario pueda crearse una cuenta y grabar sus datos personales y así cumplimentar los formularios necesarios para su inscripción y federación en cada nueva temporada.

```
MariaDB [(none)]> use kenpohiken
Database changed
MariaDB [kenpohiken]> desc usuario;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| usuario | varchar(25)   | NO   | PRI | NULL    |       |
| clave   | varchar(100)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.024 sec)
```

26. Tabla usuario con los datos de acceso de los usuarios

la segunda tabla a creas es la de ficha. Esta tabla cuya *Primary Key* es también usuario, almacena todos los datos personales de cada usuario y se utiliza para poder precargar esta información en cada formulario de modo que el trabajo de cumplimentación de cada formulario sea vea reducido significativamente en el tiempo al no ser necesario repetir campos que ya se han insertado anteriormente.

En esta tabla todas las columnas, excepto la columna del segundo apellido y la del mensaje, son obligatorias.

```
MariaDB [kenpohiken]> desc ficha;
```

Field	Type	Null	Key	Default	Extra
usuario	varchar(25)	NO	PRI	NULL	
nombre	varchar(25)	NO		NULL	
apellido1	varchar(100)	NO		NULL	
apellido2	varchar(100)	YES		NULL	
tipo	varchar(10)	NO		NULL	
documento	varchar(9)	NO		NULL	
nacimiento	date	NO		NULL	
lugarnacim	varchar(50)	NO		NULL	
nacionalidad	varchar(50)	NO		NULL	
direccion	varchar(200)	NO		NULL	
ciudad	varchar(50)	NO		NULL	
provincia	varchar(50)	NO		NULL	
codpostal	decimal(6,0)	NO		NULL	
telefono	decimal(10,0)	NO		NULL	
mail	varchar(200)	NO		NULL	
enfermedad	varchar(2)	NO		NULL	
mensaje	varchar(500)	YES		NULL	

17 rows in set (0.009 sec)

27. Tabla ficha con los datos de los deportistas

La tercera tabla que se crea es la que contiene los datos de los tutores. Necesarios para poder autorizar el tratamiento de los datos y las imágenes de los menores de 18 años.

En esta tabla todas las columnas, excepto la columna del segundo apellido, son obligatorias.

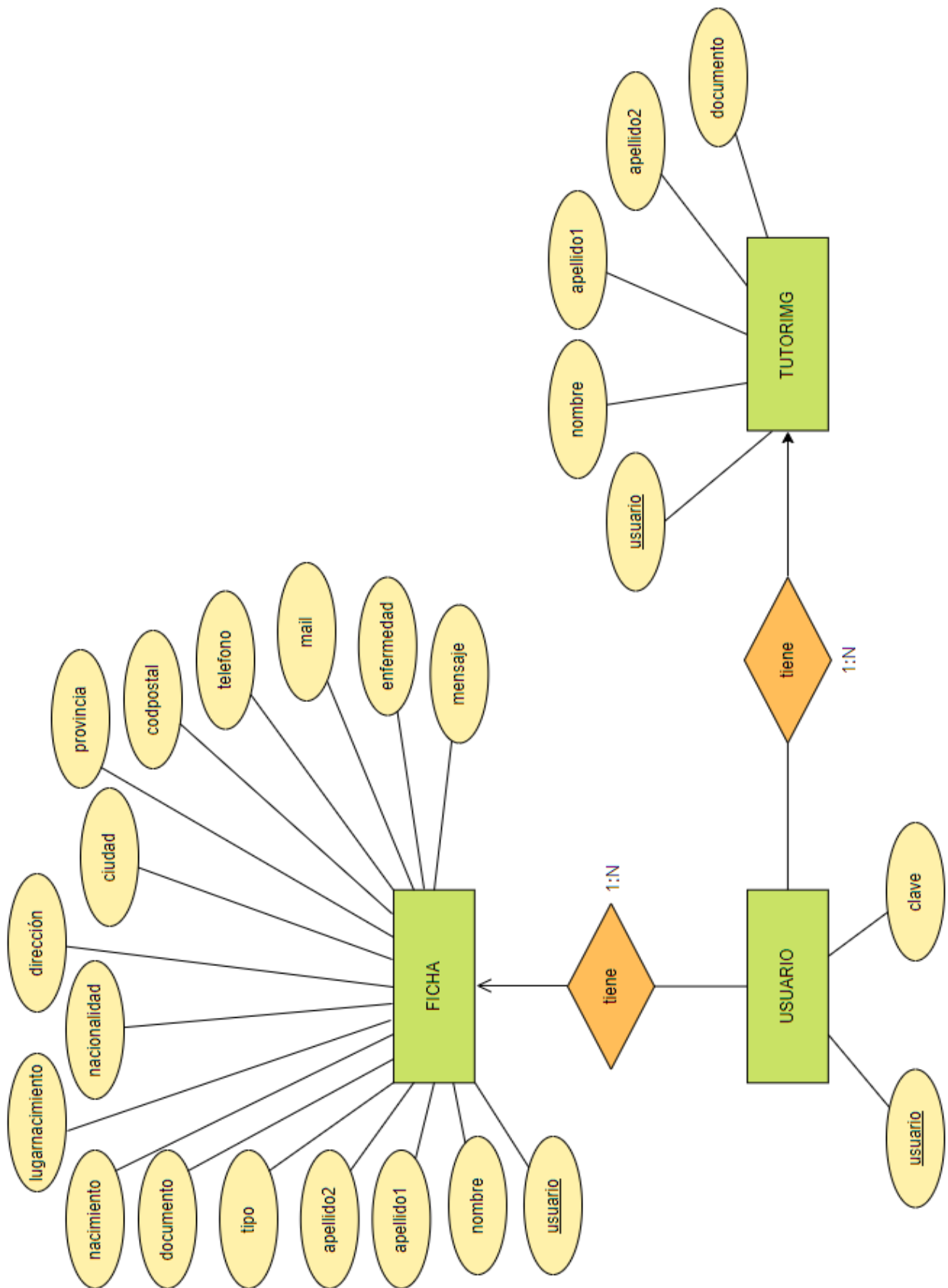
```
MariaDB [kenpohiken]> desc tutorImg;
```

Field	Type	Null	Key	Default	Extra
usuario	varchar(25)	NO	PRI	NULL	
nombre	varchar(25)	NO		NULL	
apellido1	varchar(100)	NO		NULL	
apellido2	varchar(100)	YES		NULL	
documento	varchar(9)	NO		NULL	

5 rows in set (0.030 sec)

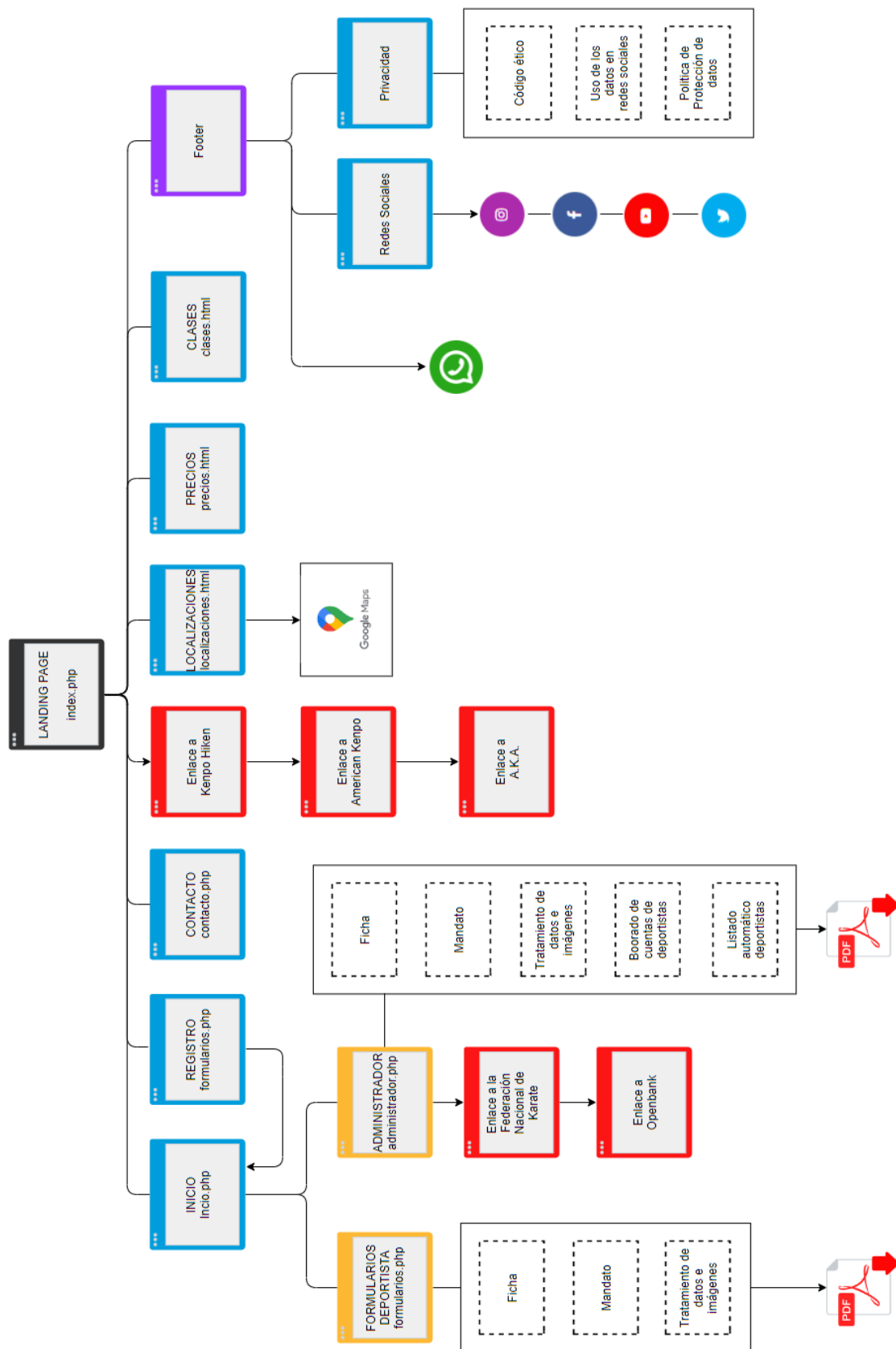
28. Tabla tutorImg con los datos de tutores

2. Esquema Entidad-Relación



29. Esquema ER

11. Mapa de la web



30.Mapa web

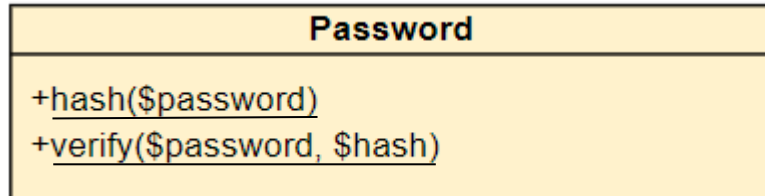
12. Estructura principal del código

1. Diagrama UML

La estructura de clases de este proyecto es sencilla. Se ha limitado el uso a 3 clases principales.

Password

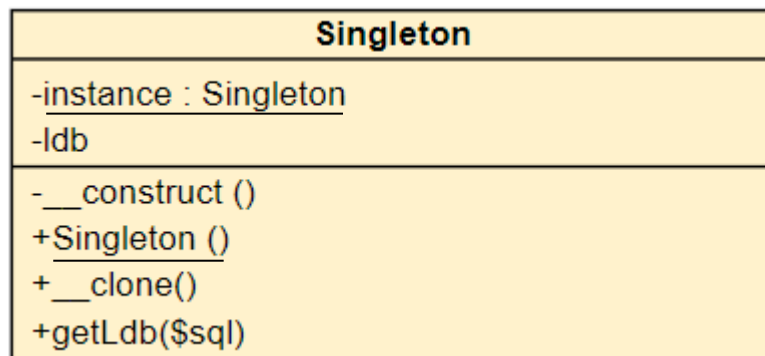
Utilizada para hashear las contraseñas de los usuarios



31.Clase Password

Singleton

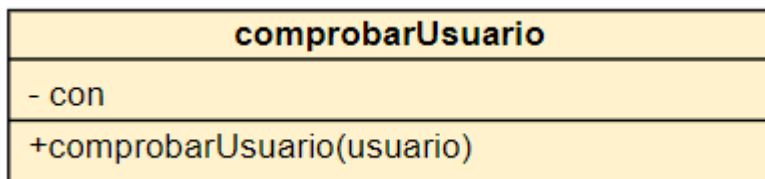
Utilizada para establecer la conexión con la base de datos como en la página administrador.php



32.Clase Singleton

ComprobarUsuario

Utilizada la página formularios.php para comprobar la existencia del usuario en la base de datos y filtrar su acceso a los formularios y la cuenta del usuario

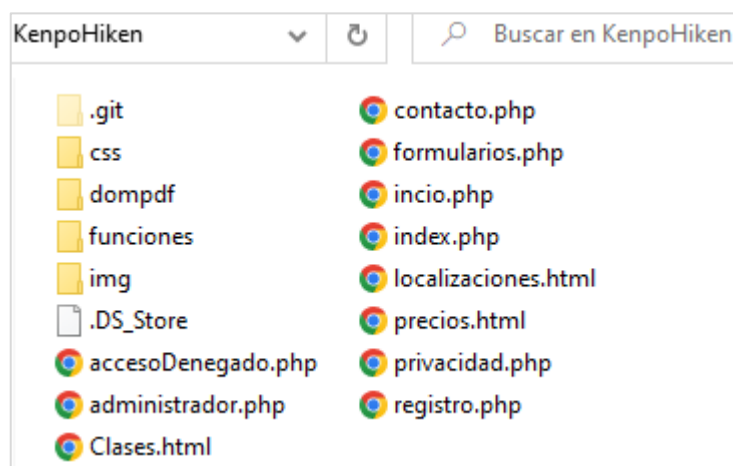


33.Clase comprobarUsuario

2. Estructura organizativa

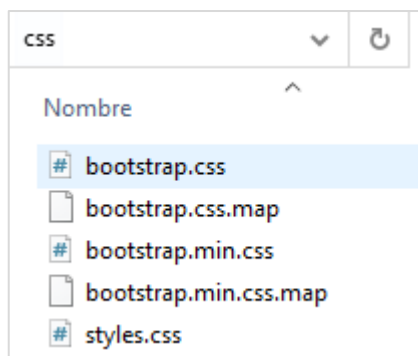
Todo el código del proyecto ha sido estructurado de modo que la parte front end quede separada de la parte back end, de las librerías y de los estilos. Para esto se ha seguido una estructura de rutas bastante sencilla de cara a que se pueda discernir el código del proyecto de un modo más intuitivo

En la carpeta raíz, encontramos todos los archivos front end y las carpetas de las librerías, funciones y estilos.



34. Contenido carpeta raíz

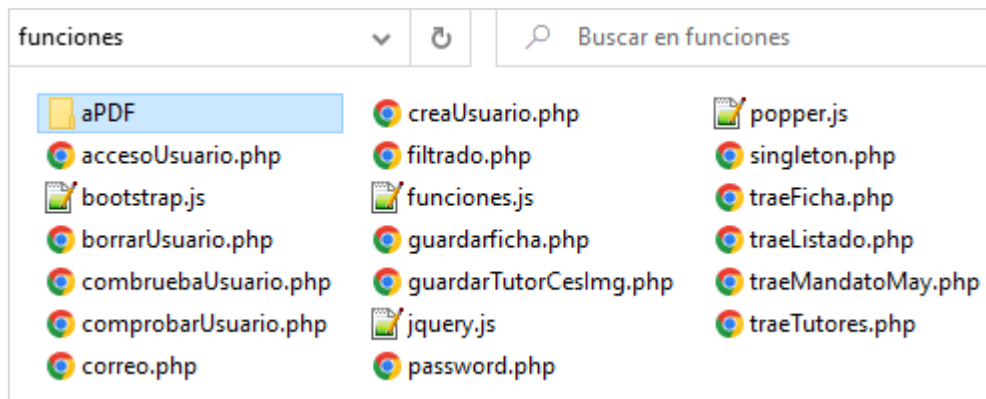
En la carpeta de estilos css, encontramos el fichero de estilos personalizado, styles.css y las librerías de Bootstrap.



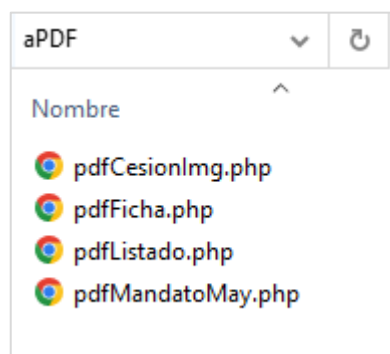
35. Contenido carpeta de estilos css

En la carpeta funciones, encontramos todo el código back end. En esta carpeta encontramos los ficheros que contienen código JavaScript, sus librerías y los ficheros que contiene el código PHP.

A su vez existen otra carpeta adicional. La carpeta aPDF que contiene el código PHP de los ficheros que se generarán por petición de los usuarios o el administrador. de ella contiene

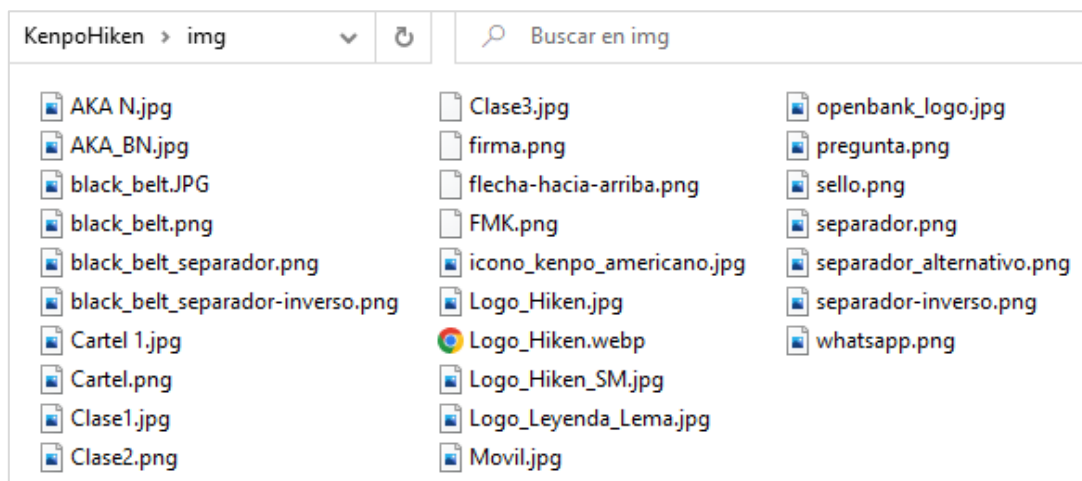


36. Contenido carpeta de funciones



37. formularios para pasar a PDF

La carpeta img, encontramos todas las imágenes utilizadas para la parte visual del proyecto, así como para la confección de los documentos pdf.



38. Contenido carpeta de imágenes

La carpeta dompdf, encontramos la librería de DOMPDF la cual se utiliza para la conversión de las páginas HTML a formato *.pdf.

3. Secciones relevantes del código back end

Para poder explicar las partes más relevantes del código implementado es necesario mencionar las 3 secciones que constituyen el eje y esencia del proyecto, sin las cuales no se hubieran cumplido los objetivos que el Club esperaba. Estas son,

- *Registrar usuarios*
- *Permitir a los usuarios grabar datos*
- *Transformar esos datos en formato *.pdf para su gestión*

Cabe decir que toda acción que fluye desde el lado del cliente al lado del servidor se controla con JavaScript, por medio de funciones que son ejecutadas a través de eventos.

Registrar usuarios

Es una parte esencial y la primera a desarrollarse ya que el usuario es el filtro utilizado para dividir la web en dos, la parte de los deportistas y la parte del administrador. A la hora de acceder a cada sección se utilizan sesiones.

Pero, antes de la creación del usuario y de grabar los datos del deportista en la base de datos hay una serie de pasos de control que son necesarios.

En una primera etapa se verifica que los campos de usuario y contraseña no están vacíos y que la contraseña contenga entre 6 y 8 caracteres.

Esta verificación se realiza utilizando el evento “blur” que se inicia al mover el cursor desde cualquiera de los campos del nombre de usuario o contraseña. Este evento lanzará dos funciones, cada una de ellas verificará los controles mencionados arriba. Lanzando diferentes alertas según la condición que se dé.

```
function comprobarUsuario() {
    $("#errorvacio17").hide();
    $("#errorvacio2").hide();
    let usuario = $("#usuario").val();
    if (!usuario) {
        $("#errorvacio1").show();
    } else {
        $("#errorvacio1").hide();
        return true;
    }
}
```

39.Función para comprobar usuario

```
function comprobarContraseña() {
    $("#errorvacio17").hide();
    $("#errorvacio1").hide();
    let clave = $("#password").val();
    if (clave.length < 6 || clave.length > 8) {
        $("#errorvacio2").show();
    } else {
        $("#errorvacio2").hide();
        return true;
    }
}
```

40.Función para comprobar la contraseña

En esa misma etapa y tras las verificaciones previas, se hace una llamada a la base de datos para verificar que el usuario seleccionado es único. Aquí entra en funcionamiento el evento vinculado al botón submit del formulario identificado como #continuar.

```
<button class="btn btn-primary" type="button" id="continuar">Continuar</button>
```

Este evento ejecuta la función seguir()

```
function seguir() {
    $("#errorvacio1").show();
    $("#errorvacio2").show();
    let usuario = $("#usuario").val();
    let clave = $("#password").val();
    if (comprobarContraseña()) {
        if (comprobarUsuario()) {
            $("#continuar").hide();
            $("#spinner").show();
            $("#cuentaNoOk").hide();
            $.ajax({
                url: "funciones/combruebaUsuario.php",
                data: {user: usuario, password: clave},
                type: 'POST',
                success: function (response) {
                    if (response === 0) {
                        //Esto significa que el usuario existe
                        $("#errorvacio17").show();
                        $("#spinner").hide();
                        $("#continuar").show();
                    } else {
                        //Esto significa que el usuario no existe
                        $('#usuario').prop("disabled", true);
                        $('#password').prop("disabled", true);
                        $("#errorvacio17").hide();
                        $("#datos").show();
                        $("#continuar").hide();
                        $("#spinner").hide();
                    }
                }
            });
        }
    }
}
```

41.Función para verificar la existencia del usuario en la BBDD

Por medio de Ajax y de JQuery se envía los datos del usuario al lado del servidor dónde se verificará con la base de datos si este usuario ya ha sido elegido.

Desde el lado del servidor se ejecuta al archivo PHP, compruebaUsuario.php.

```
<?php
//@author Alberto Bravo
include_once 'password.php';
include_once 'filtrado.php';
include_once 'singleton.php';

$usuario = filtrado($_POST['user']);
$clave = Password::hash($_POST['password']);
$con = Singleton::singleton();
if (!empty($usuario) && (strlen($clave) >= 6 || strlen($clave) <= 8)) {
    $sql = "SELECT * FROM usuario WHERE usuario = '". $usuario . "'";
    $query = $con->getLdb($sql);
    $query->execute();
    $resultado = $query->fetch();
    if ($resultado == null) {
        echo ('1');
    } else {
        echo('0');
    }
}
?>
```

42.Función para comprobar la existencia de un usuario

En esta página entran en juego las 2 clases que se utilizan en el proyecto, password, Singleton y la función de filtrado para evitar inyecciones de código.

Desde el lado del servidor se verificará de nuevo que el campo del usuario y la contraseña no están vacíos y que la contraseña tiene la extensión correcta. Si todo es correcto la consulta a la base de datos se realizaría enviándose desde el lado del servidor al del cliente un response acorde al resultado de la consulta (para este caso o un 0 o un 1).

Si existiese el usuario se habría recibido un response con valor 0 y se lanzaría una alerta informando que debe elegirse otro nombre de usuario y si no existiese, se habría recibido un response con valor de 1 y se deshabilitarían los campos de usuario y contraseña y se haría visible la sección para la cumplimentación de los datos personales.

Con la nueva sección activa el usuario puede completar la ficha pero recibirá alertas en una serie de campos específicos que contienen validaciones.

Número de documento (DNI/NIE)

Fecha de nacimiento

Código Postal

Teléfono

E-Mail

Son validaciones importantes ya que estos datos son muy sensibles al permitir el contacto con los deportistas e identificarlos legalmente por su número de documento.

Nuevamente, para verificar la validez de estos campos utiliza el evento “blur” que se activa al mover el cursor desde cualquiera de los campos a otro campo. Este evento llamará a diferentes funciones que por medio del uso de expresiones regulares validarán o no, el dato introducido.

El campo del *Número de documento* lo activa el id, documento y este junto al evento llaman a la función validaDNI() que recibe como parámetro el número de documento.

```
function validaDNI(d) {
    let numero, ok, letra;
    let expresion_regular_dni = /^[XYZ]?[d]{5,8}[A-Z]$/;
    d = $("#documento").val();
    dni = d.toUpperCase();

    if (expresion_regular_dni.test(dni) === true) {
        numero = dni.substr(0, dni.length - 1);
        numero = numero.replace('X', 0);
        numero = numero.replace('Y', 1);
        numero = numero.replace('Z', 2);
        ok = dni.substr(dni.length - 1, 1);
        numero = numero % 23;
        letra = 'TRWAGMYFPDXBNJZSQVHLCKET';
        letra = letra.substring(numero, numero + 1);
        if (letra !== ok) {
            $('#errorordni').show();
        } else {
            $('#errorordni').hide();
        }
    } else {
        $('#errorordni').show();
    }
}
```

43. Función para validar legalmente los DNI / NIE

El campo de la Fecha de nacimiento lo activa el id, nacimiento y este junto al evento llaman a la función comprobarFecha(). A excepción del resto de las demás validaciones esta validación no necesita utilizar ninguna expresión regular. Sólo valida que la fecha que se introduzca sea menor al día en el que se cumplimentan los formularios.

```
function comprobarFecha() {
    let nacimiento = $("#nacimiento").val();
    let hoy = new Date();
    let fecha = new Date(nacimiento);
    if (fecha > hoy) {
        $('#errorFecha').show();
    } else {
        $('#errorFecha').hide();
    }
}
```

44. Función para validar la fecha de nacimiento

El campo del *Código Postal* lo activa el id, CP y este junto al evento llaman a la función `comprobarCP()`.

```
function comprobarCP() {  
    let CP = $("#CP").val();  
    let expreg = /^[0-9]{5}$/g;  
    if (!expreg.test(CP)) {  
        $('#errorCP').show();  
    } else {  
        $('#errorCP').hide();  
    }  
}
```

45.Función para validar el código postal

El campo del *Teléfono* lo activa el id, telefono y este junto al evento llaman a la función `comprobarTelefono()`.

```
function comprobarTelefono() {  
    let telefono = $("#telefono").val();  
    let expreg = /^[0-9]{9}$/g;  
    if (!expreg.test(telefono)) {  
        $('#errorTelefono').show();  
    } else {  
        $('#errorTelefono').hide();  
    }  
}
```

46.Función para validar el teléfono

El campo del *E-Mail* lo activa el id, mail y este junto al evento llaman a la función `comprobarEmail()`.

```
function comprobarEmail() {  
    let email = $("#mail").val();  
    let expreg = /^[w\.\.]{1,50}@[w\.\.]{1,50}\.[a-z]{1,6}$/gi;  
    //para la correcta estructura de un email  
    if (!expreg.test(email)) {  
        $('#errormail').show();  
    } else {  
        $('#errormail').hide();  
    }  
}
```

47.Función para validar el email

Cumplimentados y validados los datos detalles de acceso del usuario y sus datos personales como deportista, se llama al escuchador vinculado al botón submit del formulario de entrada identificado como #enviar.

```
<div class="form-group">  
    <input type="submit" class="btn btn-primary" id="enviar" value="Enviar datos">
```

Y en ese momento es cuando la segunda de las secciones que constituyen el eje del proyecto, la grabación de los datos tiene lugar. Este evento llama a la función `enviarDatos()` que por medio del uso de Ajax y JQuery envía al servidor los datos recibidos desde el formulario.

```
function enviarDatos() {
    let usuario = $("#usuario").val();
    let clave = $("#password").val();
    let nombre = $("#nombre").val();
    let apellido1 = $("#apellido1").val();
    let apellido2 = $("#apellido2").val();
    let tipo = $("#tipo").val();
    let documento = $("#documento").val();
    let nacimiento = $("#nacimiento").val();
    let lugarnacim = $("#lugarNac").val();
    let nacionalidad = $("#pais").val();
    let direccion = $("#address").val();
    let ciudad = $("#ciudad").val();
    let provincia = $("#provincia").val();
    let codpostal = $("#CP").val();
    let telefono = $("#telefono").val();
    let mail = $("#mail").val();
    let enfermedad = $("#dolencia").val();
    let mensaje = $("#mensaje").val();
    if (nombre == "" || apellido1 == "" || tipo == "" ||
        documento == "" || nacimiento == "" ||
        lugarnacim == "" || nacionalidad == "" ||
        direccion == "" || ciudad == "" ||
        provincia == "" || codpostal == "" ||
        telefono == "" || mail == "" ||
        enfermedad == "") {
        $("#camposVacios").show();
        //Para ocultar el aviso a los 5 segundos.
        setInterval('$("#camposVacios").hide();', 5000);
    } else {
        $("#spinnerEnviar").show();
        $("#enviar").hide();
    }
}
```

48.Función para enviar los datos personales de los deportistas_sección 1

Se verifica que los datos no están vacíos y validado el caso, se envían los a la página de PHP `guardarficha.php`.

```
$.ajax({
    url: "funciones/guardarficha.php",
    data: {usuario: usuario,
        clave: clave,
        nombre: nombre,
        apellido1: apellido1,
        apellido2: apellido2,
        tipo: tipo,
        documento: documento,
        nacimiento: nacimiento,
        lugarnacim: lugarnacim,
        nacionalidad: nacionalidad,
        direccion: direccion,
        ciudad: ciudad,
        provincia: provincia,
        codpostal: codpostal,
        telefono: telefono,
        mail: mail,
        enfermedad: enfermedad,
        mensaje: mensaje,
    },
    type: 'POST',
})
```

Ilustración 49.Función para enviar los datos personales de los deportistas_sección 2

En el fichero guardarficha.php nuevamente se usan las clases, password y Singleton y la función de filtrado para evitar inyecciones de código.

Se verifica nuevamente que el usuario no existe y que los campos no están vacíos y se procede a grabar los datos del deportista y al mismo tiempo se crea la cuenta de usuario para futuros accesos.

```
include_once 'password.php';
include_once 'filtrado.php';
include_once 'singleton.php';

$con = Singleton::singleton();

$usuario = filtrado($_REQUEST['usuario']);
$clave = Password::hash($_REQUEST['clave']);
$nombre = filtrado($_REQUEST['nombre']);
$apellidol = filtrado($_REQUEST['apellidol']);
$apellido2 = filtrado($_REQUEST['apellido2']);
$tipo = filtrado($_REQUEST['tipo']);
$documento = filtrado($_REQUEST['documento']);
$nacimientol = filtrado($_REQUEST['nacimientol']);
$lugarnacim = filtrado($_REQUEST['lugarnacim']);
$nationalidad = filtrado($_REQUEST['nationalidad']);
$direccion = filtrado($_REQUEST['direccion']);
$ciudad = filtrado($_REQUEST['ciudad']);
$provincia = filtrado($_REQUEST['provincia']);
$codpostal = filtrado($_REQUEST['codpostal']);
$telefono = filtrado($_REQUEST['telefono']);
$mail = filtrado($_REQUEST['mail']);
$enfermedad = filtrado($_REQUEST['enfermedad']);
$mensaje = filtrado($_REQUEST['mensaje']);

$sql = "SELECT * FROM usuario WHERE usuario = '" . $usuario . "'";
$query = $con->getDb($sql);
$query->execute();
$resultado = $query->fetch();
if ($resultado != null) {
    echo ('0');
} else {
    if (empty($nombre) || empty($apellidol) || empty($tipo) || empty($documento)
        || empty($nacimientol) || empty($lugarnacim) || empty($nationalidad)
        || empty($direccion) || empty($ciudad) || empty($provincia)
        || empty($codpostal) || empty($telefono) || empty($mail)
        || empty($enfermedad)) {
        echo ('2');
    } else {
        $sql1 = "INSERT INTO ficha VALUES ('" . $usuario . "','" . $nombre . "','"
            . $apellidol . "','" . $apellido2 . "','" . $tipo . "','" . $documento
            . "','" . $nacimientol . "','" . $lugarnacim . "','"
            . $nationalidad . "','" . $direccion . "','" . $ciudad . "','"
            . $provincia . "','" . $codpostal . "','" . $telefono
            . "','" . $mail . "','" . $enfermedad . "','" . $mensaje . "')";
        $query = $con->getDb($sql1);
        $resultado = $query->execute();

        $sql = "INSERT INTO usuario VALUES ('" . $usuario . "','" . $clave . "')";
        $query = $con->getDb($sql);
        $resultado = $query->execute();
        echo('1');
    }
}
```

50.fichero php para la grabación de datos y creación del usuario.

Tras realizar las operaciones solicitadas, se tenga éxito o no desde el lado del servidor nuevamente se enviará un response al cliente que interpretará JavaScript lanzando las correspondientes alertas al usuario.

```

success: function (response) {
    if (response == 1) {
        $("#cuentaOk").show();
        $("#ficha").hide();
        $("#condiciones").hide();
        $("#enviar").show();
        $("#spinnerEnviar").hide();
    } else if (response == 0) {
        $("#datos").hide();
        $("#condiciones").hide();
        $("#cuentaNoOk").show();
        $("#continuar").show();
        $("#usuario").removeAttr("disabled");
        $("#password").removeAttr("disabled");
        $("#enviar").show();
        $("#spinnerEnviar").hide();
    } else {
        $("#camposVacios").show();
        $("#enviar").show();
        $("#spinnerEnviar").hide();
    }
}
});
}
}

```

51. Función para enviar los datos personales de los deportistas_sección 3

Si todo es correcto se confirmará la grabación de los datos y si hay algún problema se dará una alerta de campos vacío o error en el procesamiento de los datos.



52. Confirmación de datos y cuenta de usuario grabada

Si todo es correcto se confirmará la grabación de los datos y si hay algún problema se dará una alerta de campos vacío o error en el procesamiento de los datos.

53. Inicio de sesión

Con los datos ya grabados en la base de datos el objetivo de facilitar la cumplimentación de los formularios a los deportistas es algo logrado ya que se llamará a los mismos en cada formulario para que aparezcan cumplimentados casi en su totalidad. Para esto se utiliza nuevamente Ajax y JQuery. Los datos se envían desde el servidor por medio de la función `json_encode()`

```
<?php
//@author Alberto Bravo
include_once 'filtrado.php';
include_once 'singleton.php';

$usuario = filtrado($_REQUEST['user']);

$con = Singleton::singleton();

$sql = "SELECT * FROM ficha WHERE usuario = '" . $usuario . "'";
$query = $con->getLdb($sql);
$query->execute();
$resultado = $query->fetch();
if ($resultado['usuario'] == null) {
    echo ('0');
} else {
    echo json_encode($resultado);
}
?>
```

54. Uso de `json_encode()`

Y se reciben en el cliente y desempaquetan con Ajax utilizando la función `JSON.parse(response)`

```
function mostrarForm4() {
    $("#downloadOK44").hide();
    $("#form11").hide();
    $("#form22").hide();
    $("#form33").hide();
    $("#form44").show();
    let hoy = new Date();
    let usuario = $("#useroculto").val();
    $.ajax({
        url: "funciones/traeMandatoMay.php",
        data: {user: usuario},
        type: 'POST',
        success: function (response) {
            if (response != 0) {
                datosFicha = JSON.parse(response);
                $("#nombreManMay").val(datosFicha.nombre);
                $("#apellidoManMay").val(datosFicha.apellido1);
                $("#apellidoManMay2").val(datosFicha.apellido2);
                $("#documentoManMay").val(datosFicha.documento);
                $("#domicilioManMay").val(datosFicha.direccion);
                $("#CPManMay").val(datosFicha.codpostal);
                $("#localidadManMay").val(datosFicha.ciudad);
                $("#ciudad5").val(datosFicha.ciudad);
                $("#fecha5").val(hoy.getDate() + "/" +
                    (hoy.getMonth() + 1) + "/" + hoy.getFullYear());
                $("#nombreManMay").prop("disabled", true);
                $("#apellidoManMay").prop("disabled", true);
                $("#apellidoManMay2").prop("disabled", true);
                $("#documentoManMay").prop("disabled", true);
                $("#domicilioManMay").prop("disabled", true);
                $("#CPManMay").prop("disabled", true);
                $("#localidadManMay").prop("disabled", true);
                $("#ciudad5").prop("disabled", true);
                $("#fecha5").prop("disabled", true);
            }
        }
    });
}
```

Ilustración 55. Uso de `JSON.parse(response)`

La última de las tres secciones que constituyen el eje de la funcionalidad de la web es la transformación de una página en HTML a formato *.pdf. Para ellos simplemente se hace uso de la biblioteca de DOMPDF.

```
require_once ('../../dompdf/autoload.inc.php');

use Dompdf\Dompdf;
use Dompdf\Options;

$options = new Options();
$options->setIsRemoteEnabled(true);
$dompdf = new Dompdf($options);
$dompdf->set_option('dpi', 200);
$dompdf->set_option('Calibri', 'Arial');
```

56.Carga de la librería e instancia de un objeto de la clase Dompdf

Para realizar el cambio de formato, tras instanciar la clase Dompdf, bastará con indicarle el texto o código (HTML, PHP o ambos) que se quiere transformar. Esto realiza conteniendo dicho contenido entre dos funciones de Dompdf, ob_start() y ob_get_clean();

```
ob_start();
?>
<!DOCTYPE html>
<html lang="es">
  <head>

  </head>
  <body class="mx-auto" style="width: 1380px;">
    <header>

    </header>
    <main>

    </main>
  </body>
</html>
<?php
$html = ob_get_clean();
```

57.Carga de contenido a transformar por Dompdf

Como pasó final bastará con indicar el formato de salida y el título del fichero que se creará, ya en formato *.pdf.

```
$html = ob_get_clean();
$dompdf->load_html($html);
$dompdf->setPaper('a4', 'portrait');
$dompdf->render();
$dompdf->stream("Ficha_" . $valor['nombre'] .
  "_" . $valor['apellido1'] .
  ".pdf", array("Attachment" => true));
```

58.Configuración de salida pdf y título

13. Implementaciones futuras

Tras finalizar el proyecto se han realizado una serie de testeos por par de los miembros de la directiva del Club Deportivo. No hubo incidencias y la web y su funcionalidad fueron bien recibidos.

Es importante mencionar que no echaron en falta ninguna funcionalidad. Todo lo deseado está plasmado en la web.

El diseño gustó. Les resultó claro y limpio y funcionalidades como el contacto directo desde *WhatsApp Chat* gustó mucho y se sugirió que lo mismo de esa forma no es necesario tener que recibir emails y se podría descartar esa sección de la web.

Otras sugerencias de cara a futuras implementaciones, además de eliminar la sección de contacto, fueron,

- *La posibilidad de crear nuevos formularios.*
- *La implementación de una plataforma de pagos para que estos se realicen directamente desde la misma web y tras cumplimentar los formularios.*
- *Incluir poco a poco más información como los turnos a los que va cada alumno y los grados de los cinturones.*

Todas estas sugerencias podrían ser implementadas en próximas fases evolutivas, tras poner la web en producción y probar su eficacia con los deportistas de cara a la próxima temporada.

14. Conclusiones

Dado el proyecto por terminado y considerando el feedback mencionado en la sección anterior tras el testeo de la web por parte de la junta directiva. Cabe decir que el Club Deportivo ve por cumplidos los objetivos propuestos y quiere proponer el lanzamiento de la web para el próximo mes de septiembre.

Los tiempos son correctos y es factible por lo que en una próxima reunión se definirá el hosting a utilizar y se marcarán los tiempos para la puesta en producción de la web.

Tras estar la web activa, se comunicará a todos los deportistas la novedad y se les pedirá que procedan a cumplimentar sus formularios para la próxima temporada 2022 / 2023.

Esta nueva web supondrá un gran beneficio a la hora de gestionar las próximas federaciones. Será más cómodo para los deportistas nuevos y actuales y mucho más eficiente para el Club teniendo centralizados todos los documentos.

La web ofrece implementaciones interesantes y permite que se combinen diferentes temporadas al mismo tiempo por lo que puede llegar a usarse activamente durante bastante tiempo.

15. Bibliografía

- Apuntes, ejercicios y proyectos de clase.
- Web oficial de Bootstrap, [Diseño y aplicación de estilos],
<<https://getbootstrap.com/>>
- Análisis de colores [análisis de colores corporativos y logotipo Club],
<<https://color.adobe.com/es/create/color-wheel>>
- Manual de PHP, [función mail()],
<<https://www.php.net/manual/es/function.mail.php>>
- Web Diego Lázaro, [Filtrado de datos de entrada en PHP],
<<https://diego.com.es/certificacion-php>>
- WhatsApp Chat, [funcionamiento e implementación],
<<https://abelosh.com/integrar-chat-messenger-y-whatsapp-en-sitio-web>>
- Guías Dompdf, [Instalación y configuración de Dompdf],
<<https://www.srcodigofuente.es/aprender-php/guia-dompdf-completa>>
<<https://github.com/dompdf/dompdf>>
- Validaciones, [validación completa de un DNI],
<<https://donnierock.com/2011/11/05/validar-un-dni-con-javascript/>>
- Esquemas y representaciones gráficas [Casos de uso, UML, esquema E/R, mapa web],
<<https://online.visual-paradigm.com/es/diagrams/templates/?search=web>>
- Representación Diagrama de Gantt,
<<https://www.canva.com/>>
- Expresiones regulares, [pruebas y testeo de expresiones regulares],
<<https://regex101.com/r/oQrgq8/1>>
- Iconos y logos [uso para la parte front end],
<<https://www.flaticon.es/>>
- Documentación MySQL,[sentencias preparadas],
<<https://dev.mysql.com/doc/refman/8.0/en/sql-prepared-statements.html>>
- Tipos de eventos en JavaScript, [eventos para la ejecución de funciones],
<<https://desarrolloweb.com/articulos/1236.php>>
- Hosting de la web, [despliegue gratuito de la web, 000WebHost],
<https://www.youtube.com/watch?v=0WO9g8rbLo8&ab_channel=Ing.Ren%C3%A9Dom%C3%ADnguez>