



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Diseño y desarrollo de funcionalidades generales de
Buzzer Beater, una red social para la práctica deportiva
amateur.

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Tormo Torres, Vicente

Tutor/a: Albert Albiol, Manuela

Cotutor/a: Torres Bosch, María Victoria

CURSO ACADÉMICO: 2022/2023



AGRADECIMIENTOS

Quiero agradecer en primer lugar a David por estar dispuesto a formar equipo conmigo y embarcarse en este proyecto además de por su enorme disposición para solventar cualquier duda o problema que ha surgido durante el camino.

También deseo agradecer a mis tutoras por sus valiosos consejos y recomendaciones que han sido determinantes para poder dar forma al trabajo.

Finalmente, mi más especial agradecimiento a Laura. Su inestimable apoyo y constantes ánimos han sido lo que me ha animado a seguir adelante todos estos meses. Sin no fuera por ella este proyecto se habría demorado mucho más y el resultado habría sido mucho peor.



RESUMEN

Este Trabajo Fin de Grado (TFG) tiene como finalidad extender *Buzzer Beater*, una red social dedicada a la organización, publicación y participación de eventos deportivos a nivel aficionado. El TFG extenderá *Buzzer Beater* con funcionalidades extra orientadas a fomentar el aspecto más social de esta red. Las extensiones planteadas en este TFG incluyen el diseño y desarrollo de 1) una *timeline* desde donde los usuarios podrán compartir publicaciones con la finalidad de hacer partícipes a otros usuarios de momentos, experiencias o recuerdos relacionados con el deporte, 2) un sistema de mensajes directos denominados "*Buzzers*" los cuales servirán para poner en contacto de forma directa a distintos usuarios y 3) el perfil del usuario en el que se mostrará la información más relevante de la actividad de este en la red social. El desarrollo de dichas extensiones se abordará como una aplicación web responsive a la que se pueda acceder tanto desde ordenadores (escritorio y portátiles) como desde dispositivos móviles. La tecnología para utilizar será WordPress para el frontend, donde se desarrollará un tema desde cero utilizando los lenguajes HTML y SASS (CSS) para llevar a cabo la maquetación de todas las páginas y JQuery (JavaScript) para los aspectos más interactivos del interfaz. En cuanto al backend, se empleará PHP para realizar la conexión entre el servidor y la interfaz y SQL como lenguaje de consulta sobre la base de datos relacional.

Palabras clave: red social, deporte, eventos, frontend, backend, desarrollo web, WordPress.



ABSTRACT

This Final Degree Project aims to extend Buzzer Beater, a social network dedicated to the organisation, publication and participation of amateur sporting events. The project will extend Buzzer Beater with extra functionalities aimed at promoting the more social aspect of this network. The extensions proposed in this project include the design and development of 1) a timeline from where users can share publications in order to share moments, experiences or memories related to sport with other users, 2) a system of direct messages called "Buzzers" which will serve to directly contact different users and 3) the user profile which will show the most relevant information of the user's activity on the social network. The development of these extensions will be approached as a responsive web application that can be accessed from both computers (desktop and laptops) and mobile devices. The technology to be used will be WordPress for the frontend, where a theme will be developed from scratch using HTML and SASS (CSS) languages to carry out the layout of all pages and JQuery (JavaScript) for the more interactive aspects of the interface. As for the backend, PHP will be used to make the connection between the server and the interface and SQL as a query language on the relational database.

Keywords: social network, sport, events, frontend, backend, web development, WordPress.



RESUM

Aquest Treball Fi de Grau (TFG) té com a finalitat estendre *Buzzer Beater*, una xarxa social dedicada a l'organització, publicació i participació d'esdeveniments esportius a nivell aficionat. El TFG estendrà *Buzzer Beater* amb funcionalitats extra orientades a fomentar l'aspecte més social d'aquesta xarxa. Les extensions plantejades en aquest TFG inclouen el disseny i desenvolupament de 1) una *timeline* des d'on els usuaris podran compartir publicacions amb la finalitat de fer partícips a altres usuaris de moments, experiències o records relacionats amb l'esport, 2) un sistema de missatges directes denominats "*Buzzers*" els quals serviran per a posar en contacte de manera directa a diferents usuaris i 3) el perfil de l'usuari en el qual es mostrarà la informació més rellevant de l'activitat d'aquest en la xarxa social. El desenvolupament d'aquestes extensions s'abordarà com una aplicació web *responsive* a la qual es puga accedir tant des d'ordinadors (escriptori i portàtils) com des de dispositius mòbils. La tecnologia per a utilitzar serà WordPress per al *frontend*, on es desenvoluparà un tema des de zero utilitzant els llenguatges HTML i SASS (CSS) per a dur a terme la maquetació de totes les pàgines i JQuery (JavaScript) per als aspectes més interactius de la interfície. Quant al *backend*, s'emprarà PHP per a realitzar la connexió entre el servidor i la interfície i SQL com a llenguatge de consulta sobre la base de dades relacional.

Paraules clau: xarxa social, esport, esdeveniments, *frontend*, *backend*, desenvolupament web, WordPress



CONTENIDO

Capítulo 1: Introducción	9
1.1 Motivación	9
1.2 Objetivos	11
1.3 Estructura	11
1.4 Colaboraciones	12
Capítulo 2: Metodología y Soporte Tecnológico	13
2.1 Metodología	13
2.2 Soporte tecnológico al desarrollo	16
Capítulo 3: Estado del arte	18
3.1 Situación actual de las Redes Sociales	18
3.2 Propuesta	21
3.2.1 Propuesta gráfica	21
3.2.2 Propuesta tecnológica	22
Capítulo 4: Análisis del problema	23
4.1 Requisitos de la aplicación	23
4.2 Identificación de posibles soluciones	26
Capítulo 5: Diseño de la solución	27
5.1. Diseño de la interfaz gráfica de usuario	27
5.1.1 Bocetaje	27
5.1.2 Prototipado	31
5.2 Arquitectura del sistema	35
5.2.1 Arquitectura de la <i>timeline</i>	36
5.2.2 Arquitectura de los <i>Buzzers</i>	37
5.2.3 Arquitectura del perfil	38
Capítulo 6: Desarrollo de la solución propuesta	39
6.1 Desarrollo del proyecto	39
6.1.1 Estructura de archivos de WordPress	40
6.1.2 Estructura de la base de datos de WordPress	43
6.1.3 Custom Post Types	44
6.1.4 Base de datos común	48
6.1.5 Proceso de desarrollo	49
6.2 Resolución de adversidades	51
Capítulo 7: Implantación	53
7.1 Resultado final	53
7.2 Despliegue	57



Capítulo 8: Conclusiones	60
8.1 Conclusiones	60
8.2 Relación del trabajo desarrollado con los estudios cursados	60
8.3 Trabajos futuros	61
8.4 Lecciones aprendidas	61
Bibliografía	63
Anexos	64
Anexo A. Objetivos de Desarrollo Sostenible	64



TABLA DE FIGURAS

Figura 1. Penetración de las redes sociales en España de 2010 a 2022. Fuente: (Fernández, 2022).	9
Figura 2. Personas que practicaron deporte en el último año según frecuencia. Fuente: (Ministerio de Cultura y Deporte, 2021).	10
Figura 3. Metodología Scrum. Fuente: (Deemer, Benefield, Larman, & Vodde, 2009)	14
Figura 4. Captura tablero utilizado de Trello	15
Figura 5. Redes sociales más utilizadas en España. Fuente: (Statista, 2021)	18
Figura 6. Barra lateral de Facebook, Instagram y Twitter.	19
Figura 7. Mensajes directos de Instagram.	20
Figura 8. Perfil de usuario en Twitter.	20
Figura 9. Boceto timeline en Uizard.	28
Figura 10. Boceto Buzzers en Uizard.	29
Figura 11. Boceto perfil de usuario en Uizard.	30
Figura 12. Algunos de los flujos entre las distintas pantallas del boceto en Uizard.	31
Figura 13. Prototipo Timeline diseñada en Figma.	32
Figura 14. Prototipo Buzzer diseñado en Figma.	33
Figura 15. Prototipo del perfil de usuario diseñado en Figma.	34
Figura 16. Flujos entre las distintas pantallas del prototipo de Figma.	35
Figura 17. Ventana de XAMPP.	40
Figura 18. Estructura de archivos WordPress.	41
Figura 19. Archivo index.php.	42
Figura 20. Interior carpeta wp-content.	43
Figura 21. Estructura base de datos de WordPress Fuente: (Guzmán, 2019).	44
Figura 22. Advanced Custom Fields PRO para la creación de Custom Post Types.	45
Figura 23. Página de edición del Custom Post Type de los posts.	46
Figura 24. Página de edición del Custom Post Type de los parámetros de usuario..	47
Figura 25. Creación base de datos en Hostinger.	48
Figura 26. Línea de comandos integrada en Visual Studio Code..	49
Figura 27. Línea de comandos integrada en Visual Studio Code, Gulp.js iniciado.	49
Figura 28. Visualización de las ramas de trabajo en BitBucket.	50
Figura 29. Código de creación de etiquetas para especificar tamaño de pantalla.	51
Figura 30. Captura timeline de Buzzer Beater.	53
Figura 31. Captura ventana emergente creación nuevo post selección de deporte asociado.	54



Figura 32. Captura ventana emergente creación nuevo post ajustes de visibilidad.	54
Figura 33. Captura ventana emergente creación nuevo post.	54
Figura 34. Detalle botón "Redactar post"	55
Figura 35. Detalle card de un Buzzer.	55
Figura 36. Captura pantalla Buzzers.	55
Figura 37. Captura del perfil de un usuario en Buzzer Beater.	56
Figura 38. Detalle módulo superior del perfil de un usuario en Buzzer Beater.	56
Figura 39. Plugin de WordPress Duplicator.	57
Figura 40. Captura de FileZilla.	58
Figura 41. Campo para la activación de la indexación de la página en los buscadores.	59



CAPÍTULO 1: INTRODUCCIÓN

En este primer capítulo de la memoria del Trabajo Fin de Grado (TFG) se presenta la motivación que ha llevado al desarrollo de una red social, *Buzzer Beater*, como la propuesta, se exponen los objetivos que se quieren alcanzar con la realización del trabajo, y la estructura en la que está organizada la memoria. Finalmente, se menciona el trabajo del compañero que completa las funcionalidades de *Buzzer Beater* y con el cual se ha colaborado estrechamente en el desarrollo de la red social.

1.1 Motivación

Las redes sociales poseen un impacto relevante en la mayoría de las personas en la sociedad actual. De hecho, aproximadamente un 85% de la población se encuentra en redes sociales tal y como se plasma en la Figura 1:

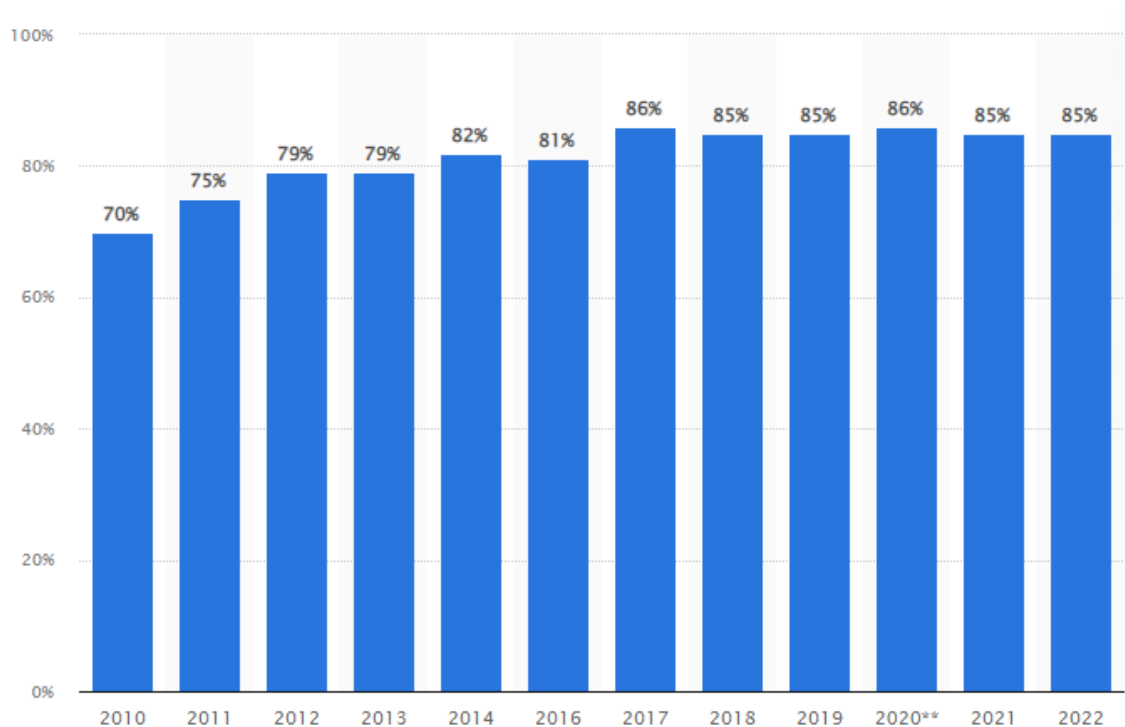


Figura 1. Penetración de las redes sociales en España de 2010 a 2022. Fuente: (Fernández, 2022).

En la figura se puede observar como la tendencia en la última década ha sido ascendente, con unos números muy estables en los años más recientes.

La variedad de redes sociales ha conseguido que muchos ciudadanos las utilicen, dado que es más probable que alguna de las funcionalidades de estas se adapte a sus necesidades. Ya sea con la finalidad de buscar información, mantenerse al tanto de la actualidad, conocer las actividades de personas cercanas, etc. Es destacable que la población que más utiliza estas redes sociales



son las generaciones z y millennial, con un 97% y 96% de uso respectivamente (Carat, 2022). Es decir, el uso de las redes sociales es una tendencia general de la población, pero con especial fuerza en los más jóvenes.

Otra de las tendencias de la sociedad actual es la práctica habitual de deporte. Especialmente tras la pandemia del COVID-19, en España ha aumentado la práctica de deporte significativamente. Casi un 60% de los españoles admitieron haber realizado deporte durante 2020 lo cual supone un aumento de 6,1 puntos con respecto a la misma encuesta realizada en 2015 (Ministerio de Cultura y Deporte, 2021). La conciencia social sobre la importancia que tiene el deporte en la salud individual ha aumentado, y esto, junto con otros factores, ha provocado una práctica deportiva habitual. La Figura 2 muestra una comparación de la frecuencia de entrenamiento entre los años 2020 y 2015:

Personas que practicaron deporte en el último año según frecuencia.

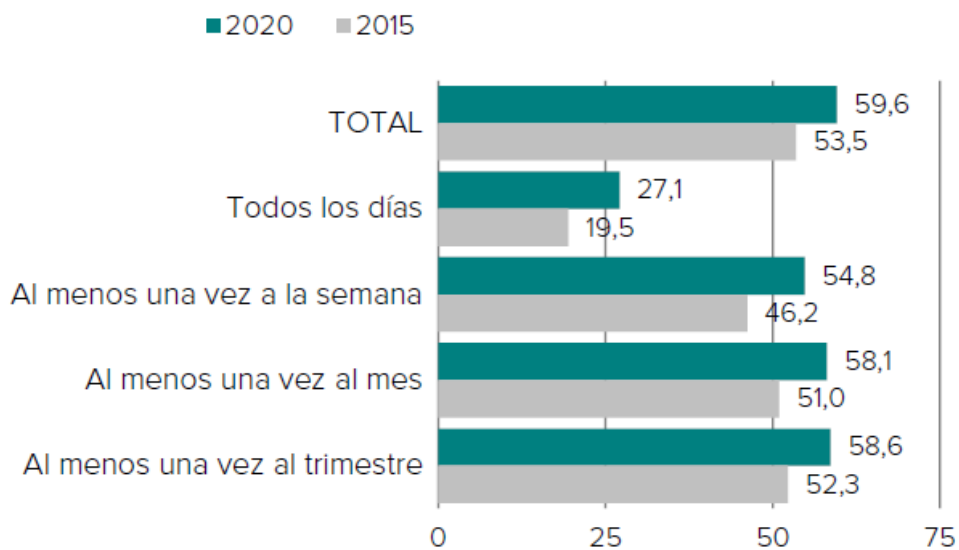


Figura 2. Personas que practicaron deporte en el último año según frecuencia. Fuente: (Ministerio de Cultura y Deporte, 2021).

En esta misma encuesta se revela que un 80% de los jóvenes practican deporte, mientras que en los mayores de 55 este porcentaje disminuye a la mitad. De nuevo, la práctica de algún deporte es una tendencia común a la población española, pero en concreto con mucho impacto entre la población más joven.

Teniendo en cuenta ambas tendencias, en el presente TFG se desarrollan parte de las funcionalidades de una red social, *Buzzer Beater*, que tiene la finalidad de satisfacer las necesidades de la población joven que practica algún deporte.

Por otra parte, también existe la motivación de contribuir con este TFG al cumplimiento de los Objetivos de Desarrollo Sostenible (ODS). Especialmente al ODS número 3 “Salud y bienestar” el cual habla de fomentar un estilo de vida saludable, así como la práctica del deporte.



1.2 Objetivos

El objetivo del presente Trabajo Fin de Grado es el diseño y desarrollo de funcionalidades extra de la red social *Buzzer Beater*. En concreto, las funcionalidades a diseñar y desarrollar son las siguientes:

- una *timeline* donde los usuarios compartirán publicaciones sobre actividades deportivas.
- un sistema de mensajes directos denominados *Buzzers* para facilitar la comunicación entre usuarios.
- un perfil único para cada usuario de la plataforma que contenga la información más relevante en lo relacionado a su actividad deportiva y en la red social.

1.3 Estructura

Esta memoria está compuesta por 11 apartados o capítulos. Los capítulos son los siguientes:

- **Capítulo 1: Introducción.** La introducción tiene como finalidad exponer la motivación del proyecto y los objetivos que se pretenden alcanzar con él. Además, se describirá la colaboración con el compañero junto con el cual se ha desarrollado la totalidad de la red social.
- **Capítulo 2: Metodología.** Se presenta la metodología aplicada para el desarrollo del TFG. La metodología se ha dividido entre la relativa a la planificación y la relativa al desarrollo de *Buzzer Beater*.
- **Capítulo 3: Estado del arte.** En este apartado se mostrarán portales o aplicaciones móviles con funcionalidades similares a la desarrollada aquí. También se mostrará la propuesta que se realiza en este TFG y en qué se diferencia de lo que actualmente existe.
- **Capítulo 4: Análisis del problema.** Este capítulo mostrará las necesidades que debe satisfacer *Buzzer Beater* en forma de requisitos de la aplicación. Después, se detallarán las soluciones propuestas para ellos.
- **Capítulo 5: Diseño de la solución.** En este capítulo se detallará la arquitectura del sistema, se mostrará el diseño detallado de la interfaz y se explicará la tecnología utilizada para implementarla.
- **Capítulo 6: Desarrollo de la solución propuesta.** Se detallará cómo se ha llevado a cabo el desarrollo de la red social, así como las diferentes estrategias de coordinación con el compañero con el que se ha trabajado en la totalidad de la web. También se mostrarán las principales adversidades superadas durante el desarrollo del proyecto.
- **Capítulo 7: Implantación.** En este capítulo será mostrado el resultado final del proyecto mediante capturas de pantalla y breves explicaciones de las funcionalidades logradas.



- **Capítulo 8: Conclusiones.** Se realizará una conclusión del proyecto junto con la relación que este ha guardado con lo aprendido en algunas de las diversas asignaturas cursadas durante el grado. Además, se incluirán una serie de reflexiones a nivel personal de lo que ha supuesto el desarrollo de este Trabajo de Fin de Grado.

1.4 Colaboraciones

Este proyecto es un proyecto complementario al Trabajo de Fin de Grado de David Millán Roldán bajo el título de “*Diseño y desarrollo de funcionalidades específicas de Buzzer Beater, una red social para la práctica deportiva amateur*”. En su trabajo se han desarrollado las funcionalidades de la red social más específicas que diferencian a *Buzzer Beater* de una red social generalista.

En su proyecto han sido diseñadas y desarrolladas partes de la aplicación web como los eventos y los grupos de usuarios. Estos son elementos indispensables para el funcionamiento de *Buzzer Beater* tal y como ha sido concebida: como una red social que fomente la práctica deportiva entre los usuarios y les facilite practicar en compañía. Por otra parte, en este TFG se tratan las partes de *Buzzer Beater* más relacionadas con una red social de ámbito menos específico como es la *timeline* o cronología, los mensajes directos, el perfil de cada usuario o el registro en la plataforma.

Como se ha mencionado, ambos trabajos son complementarios y la red social no podría funcionar sin la buena coordinación entre ambos proyectos. Es por ello por lo que el trabajo en equipo y las buenas prácticas en cuanto a planificación y desarrollo han sido fundamentales para llevar a cabo este trabajo.

Han sido utilizadas diversas herramientas como Trello, BitBucket, Figma, Uizard o Google Workspace que han permitido la coordinación y el trabajo en equipo. Gracias a ello, se ha logrado con éxito una planificación temporal marcando los objetivos que se debían alcanzar en cada momento.



CAPÍTULO 2: METODOLOGÍA Y SOPORTE TECNOLÓGICO

En este capítulo se va a exponer la metodología seguida en este TFG, así como el conjunto de herramientas utilizadas que han posibilitado la elaboración del presente trabajo. Tanto la metodología como las herramientas han sido escogidas para facilitar la colaboración y el trabajo en equipo.

2.1 Metodología

El plan de trabajo seguido en este Trabajo de Fin de Grado está fuertemente influenciado por su carácter complementario al otro TFG que completa *Buzzer Beater*. Es por esto por lo que, todas las herramientas utilizadas, la metodología empleada y las planificaciones hechas están orientadas al correcto funcionamiento del trabajo en equipo.

Se consideró que la metodología adecuada para este tipo de trabajo era la metodología ágil **Scrum**. Scrum se basa en un ciclo de trabajo iterativo y a la vez incremental sobre el cual se avanza mediante *sprints* (Deemer, Benefield, Larman, & Vodde, 2009). Estos *sprints* son de una duración determinada y los integrantes del equipo se comprometen a realizar una serie de tareas durante este período de tiempo. Normalmente la duración es de entre una y cuatro semanas. Las tareas que se realizan en los *sprints* son las que se han obtenido previamente de los requisitos del cliente. Se trata de dividir en tareas pequeñas todo el trabajo. Durante el sprint se realizan una serie de reuniones periódicas y rápidas donde se pone en común lo realizado y las dificultades encontradas.

En la siguiente imagen se resumen los roles, elementos y fases del proceso de un proyecto llevado a cabo con la metodología Scrum:

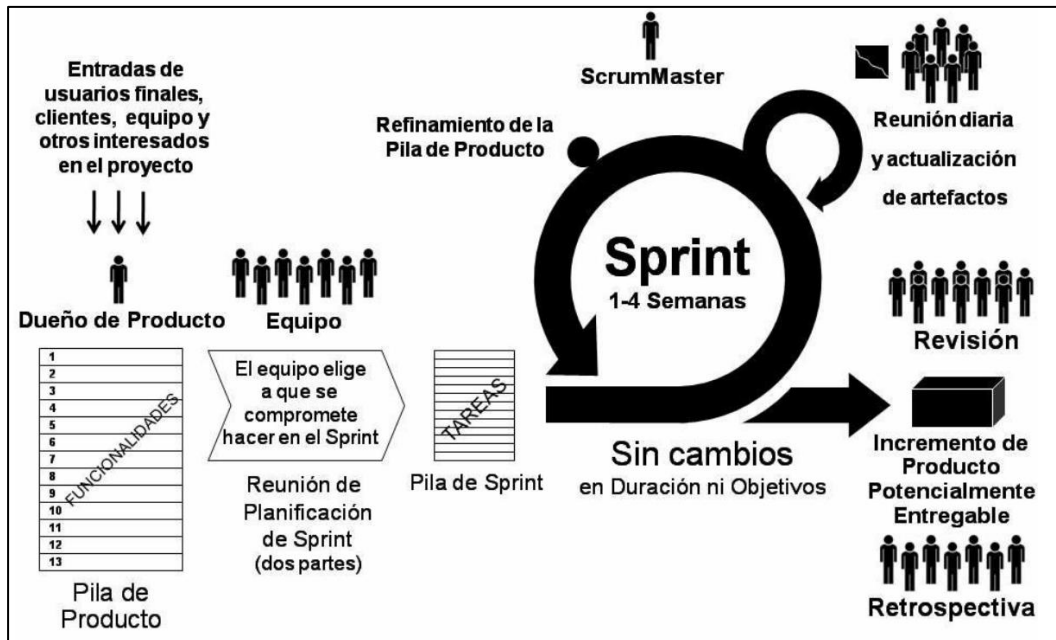


Figura 3. Metodología Scrum. Fuente: (Deemer, Benefield, Larman, & Vodde, 2009)

Scrum da mucha importancia a la adaptación frente a las adversidades que vayan surgiendo y este es uno de los motivos por los que se escogió esta metodología para implementar *Buzzer Beater*. Al tratar con muchas tecnologías nuevas, es de esperar que continuamente durante el desarrollo surjan numerosas dudas y dificultades y esta metodología facilita la resolución de estas o el paso a otra tarea que no se encuentre bloqueada.

Para realizar la segmentación de tareas de manera adecuada y que durante todo el proceso se tenga una visión clara del estado del proyecto, se utilizó la herramienta **Trello**. Esta es una herramienta especializada en la administración de proyectos. Trello funciona mediante la creación de tarjetas a las cuales se les puede añadir texto, imágenes, archivos, enlaces, etc. Estas tarjetas se organizan en una serie de columnas según su función. De esta manera, todo el equipo puede conocer el propósito de cada tarjeta según la columna en la que se encuentren y moverlas de una a otra según consideren.

En el caso de este proyecto se crearon cuatro columnas en Trello con los siguientes títulos:

- “Por hacer”
- “Haciendo”
- “Hechas”
- “Bloqueadas”

Además, se creó una tarjeta para cada tarea identificada previamente en el análisis de requisitos. Estas tarjetas se iban poniendo en la columna correspondiente según el estado de la tarea. Otra ventaja que ofrece Trello es que permite asignar las tarjetas a un usuario mediante etiquetas de color, así que el reparto de ellas también se puede visualizar de manera clara en la herramienta. La siguiente imagen es una captura de pantalla del tablero utilizado en Trello.

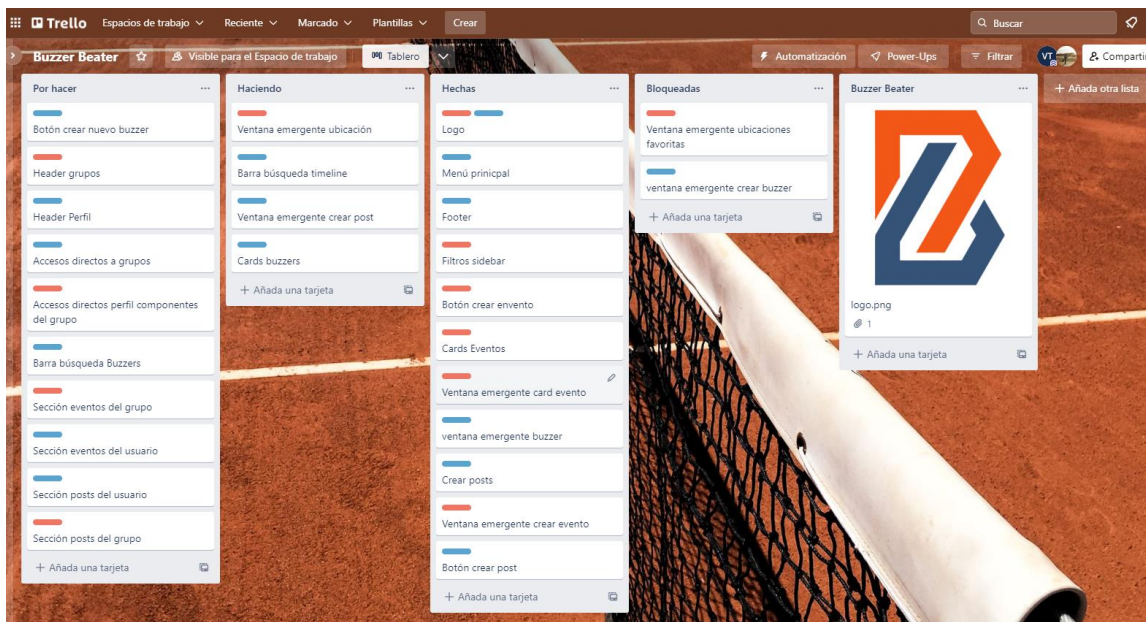


Figura 4. Captura tablero utilizado de Trello

Las reuniones estipuladas en la metodología Scrum fueron realizadas en su mayoría mediante videollamadas a través de la aplicación **Discord** la cual permitía la compartición de pantalla, así como la transmisión de archivos lo que facilita la interacción.

Sobre la planificación temporal, se establecía en una reunión de inicio de sprint un plazo y se trataban de completar las tareas asignadas en ese plazo. Las dificultades eran comentadas en las reuniones. Las más habituales eran las relacionadas con el desconocimiento de las tecnologías utilizadas y cuando una tarea quedaba bloqueada hasta que se terminara otra que se necesitaba como recurso.

Las fases que fueron establecidas fueron las siguientes:

- lluvia de ideas
- análisis de requisitos
- fases de diseño inicial y final
- fase de desarrollo
- despliegue de la red social en un dominio y servidor reales para que todo el mundo pueda acceder a ella.

También es importante reconocer que no siempre se fue capaz de cumplir los plazos estipulados. En ocasiones, las dificultades retrasaban las tareas y no se acababan a tiempo. A medida que el proyecto avanzaba y se aprendía de los errores de estimación anteriores, los tiempos y las tareas a realizar se fueron midiendo con mayor precisión. A la par que se realizaba un aprendizaje sobre las tecnologías, se realizó un aprendizaje sobre la capacidad individual de planificación y desarrollo que permitió experimentar un crecimiento a varios niveles a lo largo del desarrollo del trabajo.



2.2 Soporte tecnológico al desarrollo

Durante el desarrollo del proyecto, para poder colaborar y construir la página web en colaboración con mi compañero, se han utilizado las siguientes herramientas:

- **Bitbucket:** esta herramienta de alojamiento de código ha sido la elegida para la colaboración basada en Git. Bitbucket no solo permite guardar el código en sus distintas versiones si no que ofrece multitud de herramientas para facilitar el trabajo en equipo. Principalmente se han utilizado sus representaciones visuales de las distintas ramas de Git en las que se iba trabajando para, de un golpe de vista, conocer el estado del proyecto. También ha sido útil para comparar distintas versiones del código y para realizar retrocesos a versiones anteriores cuando ha sido necesario.
- **Visual Studio Code:** es un editor de código fuente desarrollado por Microsoft. Una de las principales ventajas que ofrece con respecto a otros editores es su buena integración con el control de versiones Git. La inmensa cantidad de plugins que se pueden instalar en este editor son capaces de adaptarlo al programador y al tipo de trabajo que esté realizando en cada momento. Los plugins más utilizados para el desarrollo de este proyecto han sido Git Lens para la buena integración de Git en el editor y PHP Intelephense para realizar más intuitiva la programación en PHP. Desde el propio editor, mediante una consola integrada, se realizaban las operaciones básicas de Git incluida la resolución de conflictos.
- **WordPress:** este CMS o sistema de gestión de contenidos es sobre el cual se ha construido *Buzzer Beater*. La estructura que ofrece ha sido muy útil para la organización de las principales páginas. Además, su capacidad de instalación de plugins ha permitido que se pudiera ampliar su funcionalidad mucho más allá de lo que ofrece de “serie”. Un ejemplo claro es que se hizo posible gracias a un plugin la creación de los Custom Post Types en los cuales se profundizará más adelante pero que fueron fundamentales para posibilitar la creación de estructuras de información necesarias que se ajustaran a lo requerido.
- **XAMPP:** el principal uso que se le ha dado a XAMPP es el de levantar el servidor Apache que permitía el funcionamiento de WordPress. Otra funcionalidad para la que se utilizó al principio del proceso fue para levantar la base de datos con MySQL, no obstante, finalmente se optó por una base de datos conjunta en la nube para unificar contenido y evitar errores de coherencia.
- **Opera DevTools:** el navegador en el que se ha ido probando la web conforme se desarrollaba es Opera que, al igual que la mayoría de los navegadores, ofrecen las llamadas “DevTools”. Estas son una serie de funcionalidades extra añadidas al inspector de Opera que ofrece toda la información sobre el código HTML, CSS o JavaScript cargado en la página. Ahí mismo podías diagnosticar algún problema de la página o hacer modificaciones rápidas para previsualizar como quedaría la web haciendo algún cambio en el código.
- **Google Workspace:** del conocido como Google Workspace se ha hecho uso principalmente de Google Drive, Google Sheets y Google Docs. Google Drive ha sido



útil para compartir documentos de interés, guardar en carpetas comunes documentación importante o albergar documentos sobre los que se ha trabajado de manera conjunta. Google Sheets se ha utilizado para plasmar las primeras ideas del proyecto, realizar esquemas como el análisis de requisitos y repartir de ellos derivar las tareas correspondientes a cada trabajo. Por último, los documentos de Google Docs se han utilizado para recoger anotaciones y recopilar información de interés.

- **Uizard:** esta ha sido la herramienta escogida para realizar los primeros *sketches* de los cuales nacería el prototipo de la red social. Esta herramienta permitió dar forma a las primeras ideas y luego hacer mucho más ágil la elaboración de un prototipo más completo.
- **Figma:** esta es una herramienta de prototipado que ha permitido diseñar tanto visual como funcionalmente la red social con el aspecto tal y como se quería obtener en el resultado final. Además, también permite la creación de flujos de navegación gracias a los cuales, la visualización del prototipado es como si fuera la aplicación real ya que deja navegar entre las diferentes pantallas.



CAPÍTULO 3: ESTADO DEL ARTE

Hoy en día, tal y como se ha expuesto en la introducción, el uso de las redes sociales constituye el día a día de una amplia mayoría de ciudadanos. En este capítulo se expondrán las principales redes sociales que conforman el panorama actual y se justificará la diferencia de *Buzzer Beater* con todas ellas.

3.1 Situación actual de las Redes Sociales

Dado que este trabajo se centra en las partes de *Buzzer Beater* relativas a una red social generalista, se estudiarán las redes sociales de ámbito general y no las específicas enfocadas a la práctica deportiva. De esta manera, se tratará de comprender qué ofrecen otras redes sociales en aquellas funciones que tienen en común con *Buzzer Beater*.

En la Figura 5 se pueden observar las redes sociales más utilizadas en España en el año 2021.

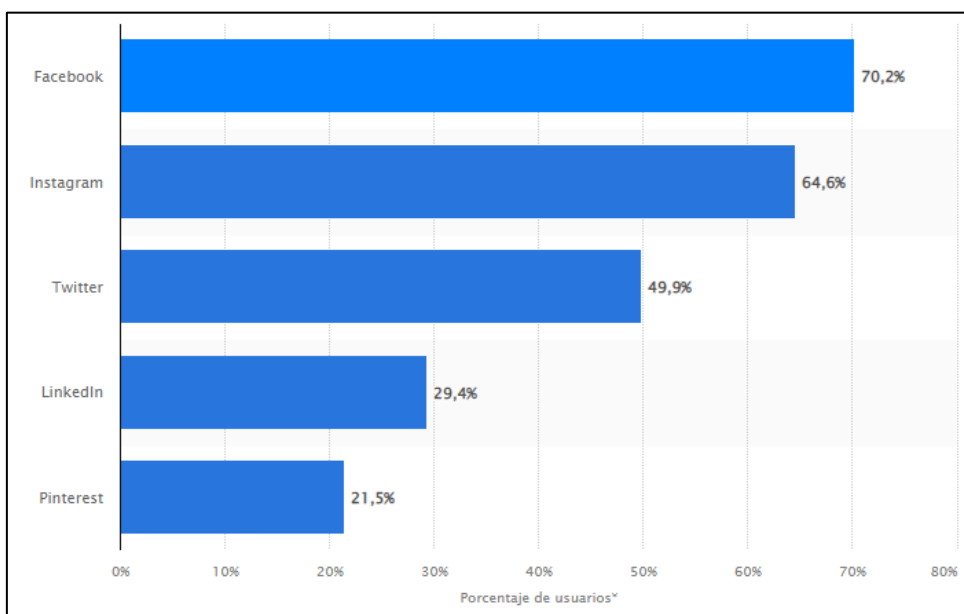


Figura 5. Redes sociales más utilizadas en España. Fuente: (Statista, 2021)

Tal y como se observa en la figura, Facebook, Instagram y Twitter son las redes sociales más extendidas y utilizadas por los ciudadanos. Es por ello por lo que este análisis se centrará en estas tres redes sociales líderes. En concreto, se analizan tres elementos básicos que son comunes de las redes sociales: *timeline*, *mensajes directos* y *perfil de usuario*. Estos son los elementos que se implementan en este trabajo para la red social *Buzzer Beater*.

- **Timeline**

En las tres redes sociales, la *timeline*, *feed* o cronología sigue un esquema muy similar. Todas ellas están llenas de *posts* de distintos tipos que pueden contener textos, imágenes, vídeos e incluso todas ellas a la vez. Las tres organizan su *timeline* en un esquema vertical en el cual las nuevas publicaciones aparecen siempre en la parte superior de la pantalla y se van moviendo hacia abajo conforme aparecen en ella otras nuevas.



Otra práctica común en los tres casos es la utilización de barras laterales o *sidebars*. Es común en las redes sociales aprovechar los espacios laterales que deja la *timeline* para mostrar información adicional, enlaces directos o botones útiles para la navegación por la página.

No obstante, según la red social, encontramos particularidades en ellas. Por ejemplo, en Facebook encontramos dos *sidebars* con diferente propósito, el de la izquierda para accesos directos dentro de la red social y el de la derecha para publicidad y mostrar sugerencias de perfiles. En cuanto a Instagram también existen dos. La situada a la izquierda sirve de menú principal de la página mientras que la de la derecha muestra sus *stories*. Por último, en Twitter, al igual que las anteriores, utiliza una barra lateral a la izquierda a modo de menú principal con accesos directos a otras páginas mientras que en la parte derecha sitúa diferentes módulos con noticias de interés o perfiles sugeridos.

En la siguiente imagen se muestra la barra lateral izquierda de las tres redes sociales para apreciar con claridad sus similitudes.

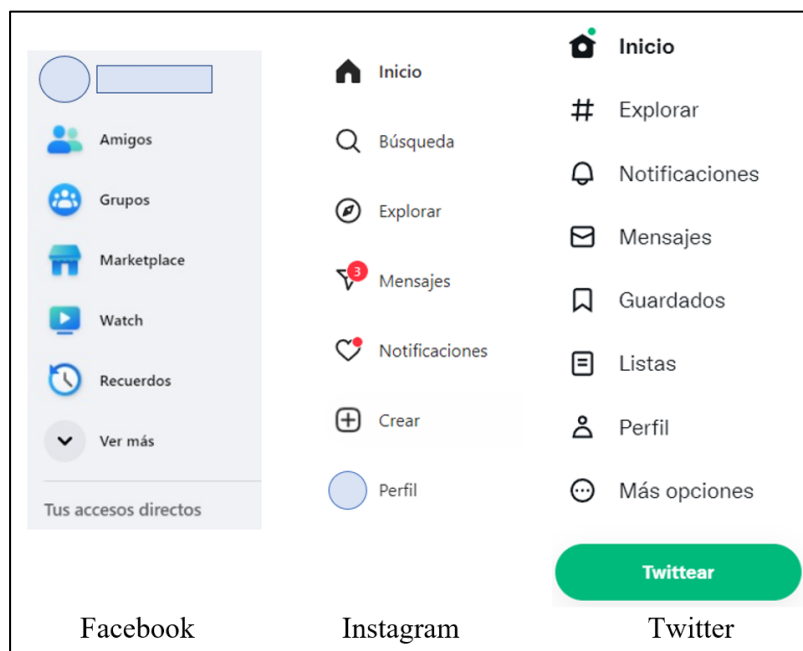


Figura 6. Barra lateral de Facebook, Instagram y Twitter.

- **Mensajes directos**

Los mensajes directos son un modo de comunicación con otros usuarios de la red social mediante mensajes de texto, imágenes o vídeos generalmente. La principal particularidad de estos mensajes, a diferencia de las publicaciones que aparecen en la *timeline*, es que solamente son visibles para el remitente y el destinatario de este.

De nuevo, las tres redes sociales que se están estudiando, funcionan de una manera casi idéntica. En todas ellas estos mensajes pueden contener además de texto variado contenido multimedia. También se organizan en una vista que muestra una lista ordenada con una pequeña previsualización del mensaje y del remitente.

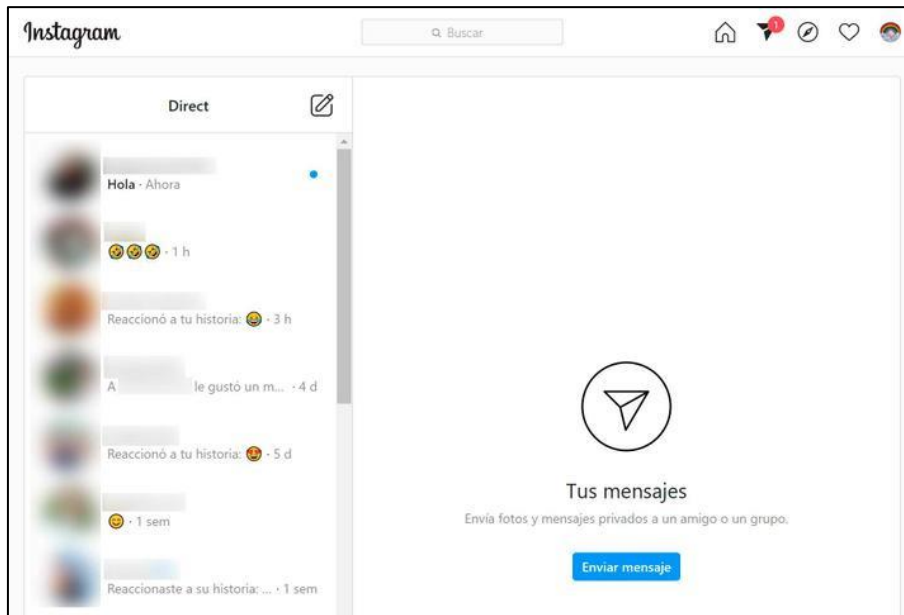


Figura 7. Mensajes directos de Instagram.

- **Perfil de usuario**

El perfil es aquella página de la red social en la que se muestra la información relativa a un usuario. Aparecen datos básicos como su nombre, una imagen que el usuario haya escogido y otros datos específicos según la red social. De nuevo, la información mostrada en Facebook, Instagram y Twitter es muy similar. En todas, además del nombre y una imagen, aparece un número de seguidores y seguidos y las publicaciones pertenecientes a ese usuario en concreto o relacionadas con él.

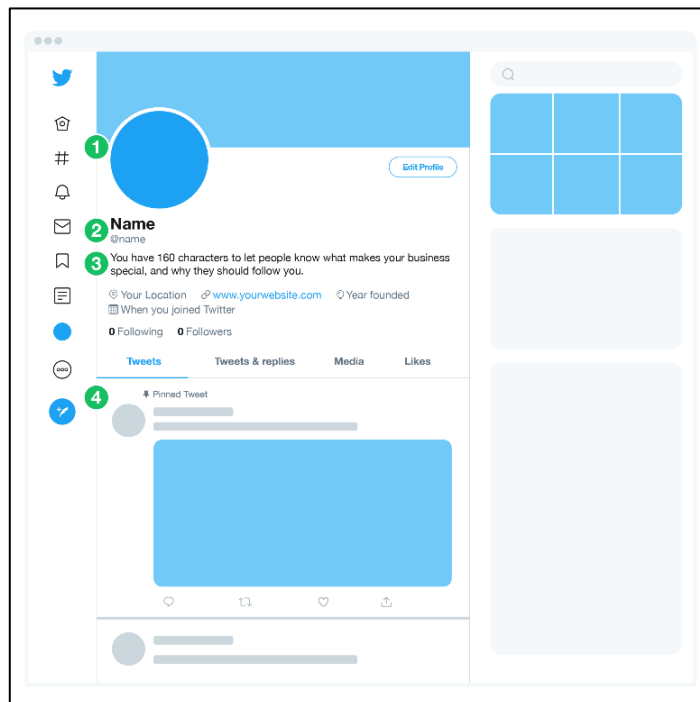


Figura 8. Perfil de usuario en Twitter.



3.2 Propuesta

Una vez se ha visto cómo las principales redes sociales dan uso a los elementos más comunes de ellas, se procede a explicar la propuesta que se hace en *Buzzer Beater* con respecto a estos mismos elementos.

3.2.1 Propuesta gráfica

Timeline

En cuanto a la *timeline*, en *Buzzer Beater* se ha querido prescindir de las *sidebars* con la intención de lograr una interfaz más limpia y comprensible. Tanto en Facebook como Twitter como Instagram, existía una barra lateral izquierda prácticamente idéntica con unas funciones de navegación por la página muy similares. En *Buzzer Beater* estas funciones las realiza el menú superior que se mantiene fijo para todas las páginas de la web. De esta manera, se logra una coherencia a lo largo de las diferentes ubicaciones de la red social y se logra también ofrecer al usuario una navegación sencilla e intuitiva por la web.

Las publicaciones en este *timeline* también aparecen con orientación vertical y como es habitual, la más reciente se sitúa siempre en la parte superior desplazando el resto hacia abajo. Las publicaciones podrán mostrar tanto texto como contenido multimedia y permitirán respuestas de otros usuarios con la finalidad de facilitar la interacción.

Mensajes directos o *Buzzers*

De la misma manera que en las tres redes sociales estudiadas, los mensajes directos de *Buzzer Beater* cumplen con la función de ofrecer un canal de comunicación usuario – usuario de manera privada entre ellos. La particularidad reside en el nombre de estos ya que en *Buzzer Beater* se les denomina *Buzzers*. Este nombre busca ser fácil de recordar por su relación con el nombre de la red social y además traer a la mente del usuario la imagen de una bocina como si por ella se anunciara que se tiene un nuevo *Buzzer* esperando.

En los *Buzzers* también se podrá compartir contenido multimedia y tendrán una especial orientación a ayudar a los usuarios a ponerse en contacto para la práctica deportiva ya que es uno de los objetivos de esta red social.

Perfil de usuario

El perfil de un usuario en *Buzzer Beater* constará de distintos módulos. El principal se situará en la parte superior y muestra un nombre, una imagen, número de eventos creados dentro de *Buzzer Beater*, los deportes favoritos y el número de amistades establecidas en la red social.

Más abajo se mostrará en una lista horizontal los grupos a los que pertenece el usuario y finalmente otras dos listas horizontales con los eventos y *posts* que el usuario ha creado. De esta manera, se espera que cualquier persona pueda consultar la información más relevante sobre otros usuarios de una manera sencilla y además comprobar sus afinidades en cuanto a lo deportivo de un golpe de vista.

En resumen, la propuesta que realiza este trabajo mantiene los estándares seguidos en las redes sociales para los elementos *timeline*, mensajes directos y perfil de usuario, pero otorgándoles una orientación especial hacia la práctica deportiva en comunidad que permite distinguirlos.



3.2.2 Propuesta tecnológica

Dadas las necesidades previstas para el proyecto, se han precisado de muchas tecnologías diferentes. La mayoría de ellas destinadas al desarrollo, pero también otras necesarias para el trabajo en equipo. Además, muchas de ellas se han ido aprendiendo y dominando desde cero gracias a este proyecto y las exigencias que, conforme se avanzaba en él, aparecían.

Para construir las páginas se ha utilizado HTML y PHP intercalados. HTML para estructurar el aspecto de la página y PHP permitía llamar a elementos de otros archivos, crear bucles, reglas condicionales, etc. La parte estética de *Buzzer Beater* se ha programado en SASS que se compilaba a CSS y JavaScript nativo, pero también se ha utilizado a menudo la librería JQuery. También ha sido necesario aprender a utilizar el formato de intercambio de datos JSON para la transmisión y utilización de estructuras de datos de unos archivos a otros. De la mano de lo anterior, con la finalidad de realizar llamadas a servidores sin refrescar la página (por ejemplo, para aplicar un filtro) se han utilizado las llamadas AJAX o *AJAX calls*. En cuanto a la instalación de la herramienta de compilación Gulp.js se ha utilizado node.

Finalmente, ha sido necesario comprender y aprender a manejar con precisión el software de control de versiones Git con el que se coordinaba el desarrollo en equipo y se combinaba el código resolviendo los conflictos.



CAPÍTULO 4: ANÁLISIS DEL PROBLEMA

En este capítulo se presenta un análisis del problema. En primer lugar, se introducen los requisitos que la red social debe cumplir para alcanzar los objetivos propuestos para este TFG. A continuación, se mostrarán las soluciones propuestas para satisfacer estos requisitos.

4.1 Requisitos de la aplicación

Con la finalidad de elaborar una red social que satisfaga los objetivos propuestos al inicio del TFG, se ha de realizar un análisis de las necesidades que estos conllevan. Estas necesidades no representan solamente un problema que se ha de resolver sino también una oportunidad para otorgar de una identidad propia al proyecto que lo diferencie de otras redes sociales.

A continuación, se presentan estas necesidades en forma de requisitos, los cuales se agruparán según con qué parte de la aplicación web tienen relación. Cabe destacar que también se han incluido en este apartado los requisitos necesarios de la página de registro que se necesita para acceder a la aplicación web y que se ha elaborado de forma adicional a este TFG.

- Requisitos registro

ID	REQ1
Título	Registro de usuario
Descripción	Un usuario anónimo debe poder registrarse en la red social

ID	REQ2
Título	Inicio de sesión
Descripción	Un usuario registrado debe poder iniciar sesión y acceder a la red social

ID	REQ3
Título	Restablecimiento de contraseña
Descripción	Un usuario registrado debe poder solicitar un restablecimiento de contraseña desde la página de registro

ID	REQ4
Título	Mensajes de error
Descripción	El usuario debe de disponer de mensajes de error que le indiquen si algo ha provocado un error en el registro (REQ1) o inicio de sesión (REQ2)



- Requisitos *timeline*

ID	REQ5
Título	Creación de <i>posts</i>
Descripción	El usuario debe ser capaz de compartir lo que desee mediante texto o archivos multimedia y que esto se muestre en la página de la <i>timeline</i>

ID	REQ6
Título	Visualización de <i>posts</i>
Descripción	El usuario debe ser capaz de visualizar en la <i>timeline</i> los <i>posts</i> creados tanto por él como por otros usuarios

ID	REQ7
Título	Búsqueda de <i>posts</i>
Descripción	El usuario debe ser capaz de filtrar los <i>posts</i> que visualice en la <i>timeline</i> de manera que le sea sencillo encontrar los que desee de manera rápida

- Requisitos *Buzzers*

ID	REQ8
Título	Visualización <i>Buzzers</i>
Descripción	El usuario debe poder visualizar los <i>Buzzers</i> con otros usuarios en su conjunto

ID	REQ9
Título	Vista <i>Buzzer</i>
Descripción	El usuario podrá visualizar con más detalle el chat con otro usuario concreto

ID	REQ10
Título	Notificación <i>Buzzer</i> nuevo
Descripción	El usuario debe poder visualizar en algún lugar que ha recibido un nuevo <i>Buzzer</i> de otro usuario concreto

ID	REQ11
Título	Búsqueda de <i>Buzzers</i>
Descripción	El usuario debe ser capaz de localizar aquellos chats que desee de una manera sencilla



- Requisitos perfil

ID	REQ12
Título	Foto de perfil
Descripción	El usuario podrá decidir si mostrar una foto de su elección en su perfil o no

ID	REQ13
Título	Información relevante
Descripción	El usuario podrá decidir si comparte o no información relevante en su perfil como edad o deportes favoritos

ID	REQ14
Título	Número de amigos
Descripción	El perfil de un usuario mostrará el número de amigos que el usuario tiene dentro de la red social

ID	REQ15
Título	Visualización de grupos
Descripción	El perfil de un usuario mostrará los grupos de usuarios a los que este pertenece

ID	REQ16
Título	Visualización eventos
Descripción	El perfil de un usuario mostrará los eventos que este usuario ha creado desde su registro en la red social

ID	REQ17
Título	Visualización de <i>posts</i>
Descripción	El perfil de un usuario mostrará los <i>posts</i> que este usuario ha creado desde su registro en la red social



4.2 Identificación de posibles soluciones

Tal y como ha sido expuesto en el apartado de motivación de este TFG (ver sección 1.1), la oportunidad de diseñar y desarrollar una red social enfocada en fomentar la práctica deportiva es la consecuencia lógica de las dos tendencias allí vistas: el aumento del uso de las redes sociales y de la práctica deportiva entre la gente joven.

Con todo ello, se ha realizado un análisis de requisitos en el apartado anterior valorando aquello que sería indispensable para lograr los objetivos propuestos y finalizar con un producto que atraiga a aquellas personas que conformarían el público objetivo de este proyecto.

En primer lugar, para la página de la *timeline* es necesario la creación de los *posts* o publicaciones que la conformarían. Para complementarla y que cumpla con todas las funciones esperadas de ella se debe diseñar e implementar un botón que te permita crear *posts* (REQ5), una ventana emergente con todo lo necesario para crear el *post* que aparezca al pulsar el botón y una barra de búsqueda en la parte superior que te permite filtrar publicaciones según usuario o palabras clave (REQ7). De esta manera, se visualizarían en ella todos los *posts* creados tanto por el usuario como por el resto de los usuarios de la página (REQ6). En el Estado del Arte se ha visto como las redes sociales más importantes se apoyan mucho en las barras laterales para otorgar de más funcionalidades a *timelines* en el caso de *Buzzer Beater* se ha prescindido de las barras laterales por dos motivos: 1) se apuesta por una interfaz limpia y clara con el menor número de elementos posible que distraigan la atención del usuario y 2) la labor de navegación por la página web que normalmente se delega a las *sidebars* en el caso de este proyecto recae sobre el menú superior siempre presente.

En cuanto al sistema de mensajes directos o *Buzzers*, se debe disponer de una lista que contenga las *cards* donde se vea el usuario al que corresponde ese chat y un pequeño distintivo que indique existe un mensaje nuevo proveniente de ese chat en concreto (REQ8 y REQ10). De nuevo, existirá una barra buscadora para filtrar por nombre de usuario y así hacer más rápida la búsqueda del chat que se desea (REQ11), así como un botón con el crear un nuevo chat con un usuario. Al hacer clicar sobre un *Buzzer* se abrirá una ventana emergente en la cual se podrá escribir el mensaje deseado tal y como los usuarios están acostumbrados en otras redes sociales (REQ9).

Finalmente, el **perfil** de cada usuario contendrá la información base que contienen las redes sociales más importantes como se ha visto en el estado del arte. Esto incluirá una foto de perfil o en caso de que el usuario no escoja ninguna aparecerá un icono genérico, la edad del usuario en caso de que este desee mostrarla, los deportes favoritos del usuario y los amigos que tiene dentro de la red social (REQ12 a REQ15). Además, es conveniente mostrar también los grupos a los que pertenece, así como los eventos y publicaciones que ha creado el usuario (REQ16 y REQ17).

Como parte adicional de la red social, se elabora una página inicial de **registro** para que los usuarios se creen la cuenta y accedan a *Buzzer Beater* mediante una contraseña segura (REQ1 y REQ2). Esta página de inicio de sesión también posee las principales características que se encuentran en las páginas de inicio de sesión de otras redes sociales. Esta debe contener una zona para iniciar sesión y otra para crear usuario si es el caso de que se accede por primera vez. También son importantes las funcionalidades de restablecimiento de contraseña en caso de olvido (REQ3) y los mensajes de error en caso de que no corresponda el usuario o contraseña introducidos (REQ4).



CAPÍTULO 5: DISEÑO DE LA SOLUCIÓN

El presente capítulo expondrá en primer lugar el bocetaje y prototipo final de *Buzzer Beater*. Más adelante, se mostrará la arquitectura que existe debajo de la red social. Se comentarán las tecnologías empleadas, así como la lógica empleada durante el desarrollo y las necesidades de coordinación en el trabajo en equipo.

5.1. Diseño de la interfaz gráfica de usuario

La fase de diseño de la interfaz gráfica de usuario (IGU) de *Buzzer Beater* tuvo dos etapas: el bocetaje en la herramienta Uizard y el prototipado final en Figma.

5.1.1 Bocetaje

Para realizar los primeros bocetos y formar una idea de cómo sería la apariencia de *Buzzer Beater* se utilizó la herramienta: **Uizard**. Uizard es una herramienta de diseño para no diseñadores que desean plasmar de una manera ágil y sencilla sus ideas. Además, permite la creación de flujos entre las pantallas lo que supone una funcionalidad interesante para darle “vida” a estos bocetos.

Como se ha comentado anteriormente, la idea inicial era la de construir una página web agradable a la vista, sin muchos elementos que la recargaran a ojos del usuario y donde el protagonista fuera el deporte y las actividades deportivas.

Para que el diseño fuese intuitivo para el usuario, el aspecto debía guardar semejanzas con el de las redes sociales más usadas. De esta manera, el usuario ya sería capaz de navegar por la página minimizando la cantidad de cosas nuevas que aprender sobre *Buzzer Beater*.

Estos bocetos son el producto de los bocetos originales con modificaciones que fueron sugeridas tanto por las tutoras de este Trabajo de Fin de Grado como por compañeros y otros usuarios que conformarían el público objetivo.

A continuación, se pueden observar los bocetos de las partes de *Buzzer Beater* desarrolladas en este TFG.

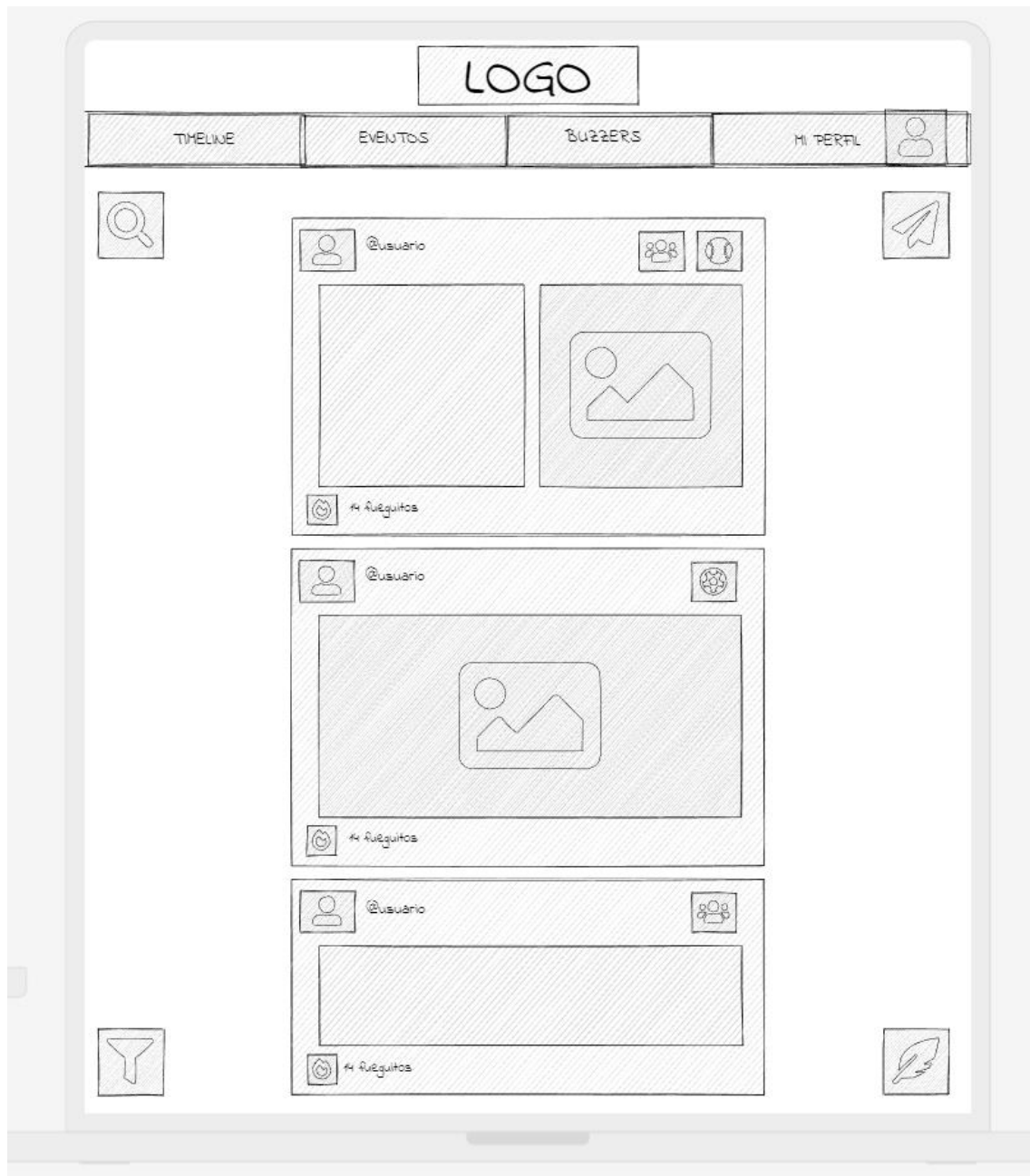


Figura 9. Boceto timeline en Uizard.

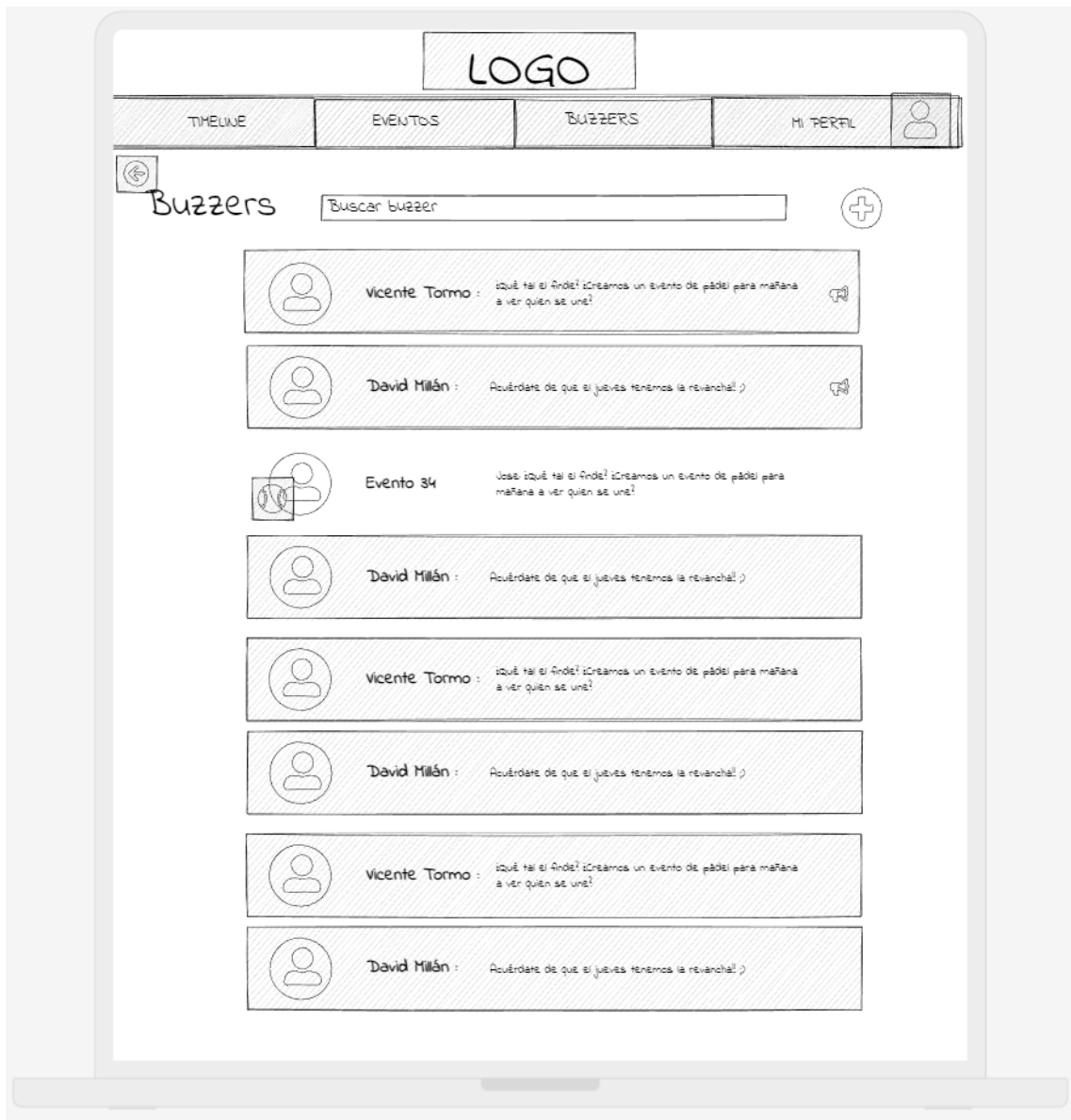


Figura 10. Boceto Buzzers en Uizard.

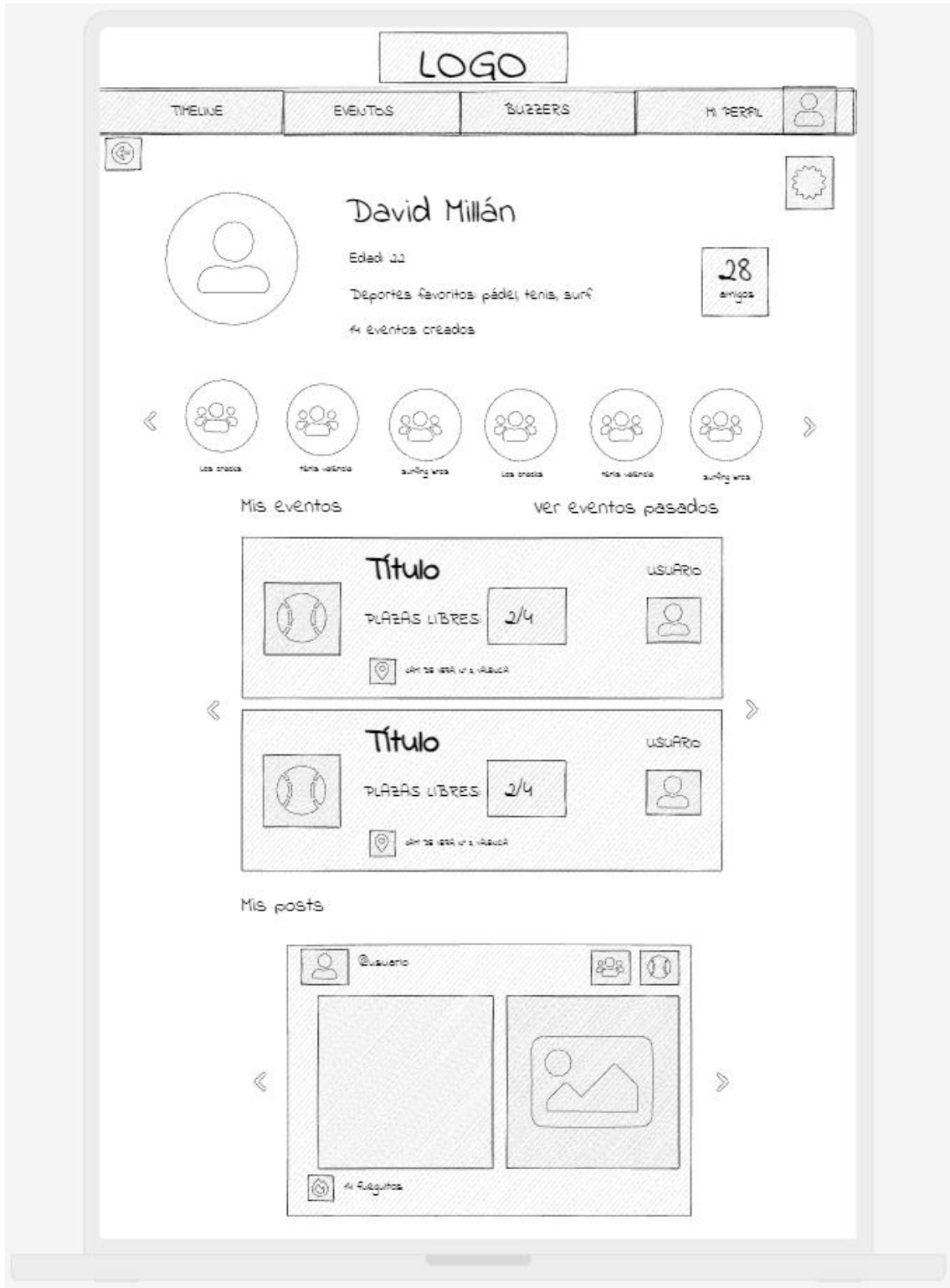


Figura 11. Boceto perfil de usuario en Uizard.

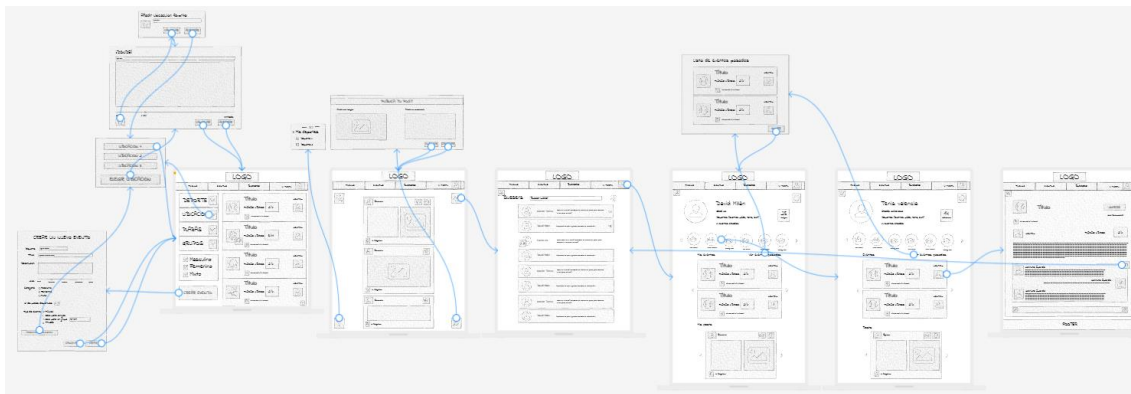


Figura 12. Algunos de los flujos entre las distintas pantallas del boceto en Uizard.

5.1.2 Prototipado

Una vez se tenían los bocetos iniciales, el paso al prototipo resulta más sencillo ya que ya se tiene la idea definida de cómo se desea que sea la apariencia final de la aplicación. Para realizarlos se utilizó la herramienta **Figma**. Esta herramienta era ideal para la realización de los prototipos. A diferencia de Uizard, esta sí está pensada para diseñadores profesionales y su manejo era más complejo que el de la anterior herramienta. No obstante, el resultado final también es más preciso. De nuevo, sobre estos prototipos hubo cambios impulsados por opiniones recibidas en pro de construir una página web que fuese más intuitiva para la mayoría de los usuarios.

En Figma también se tuvo la oportunidad de reflejar los flujos de navegación que seguirían los usuarios dentro de la página web lo cual fue muy útil para mostrar el prototipo a las tutoras de este TFG y a compañeros. De esta manera, gracias a la funcionalidad que Figma ofrece, podían ver los prototipos a pantalla completa y navegar entre las distintas pantallas y ventanas emergentes haciendo clic en los menús o botones como si de la web real se tratara.

En las siguientes imágenes se muestran los mismos elementos de *Buzzer Beater* antes vistos como bocetos, pero ahora siendo el prototipo final realizado con Figma.

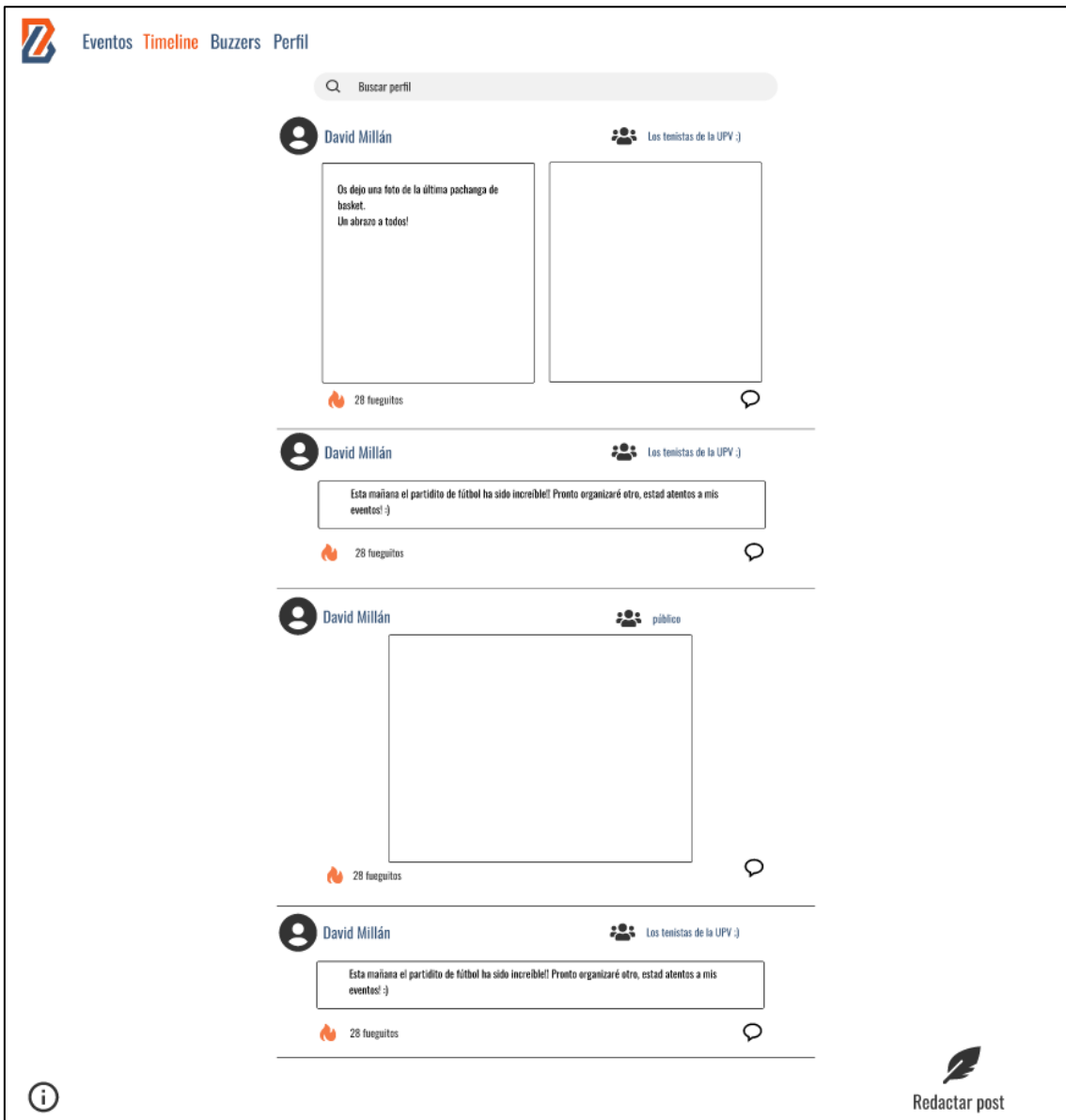


Figura 13. Prototipo Timeline diseñada en Figma.

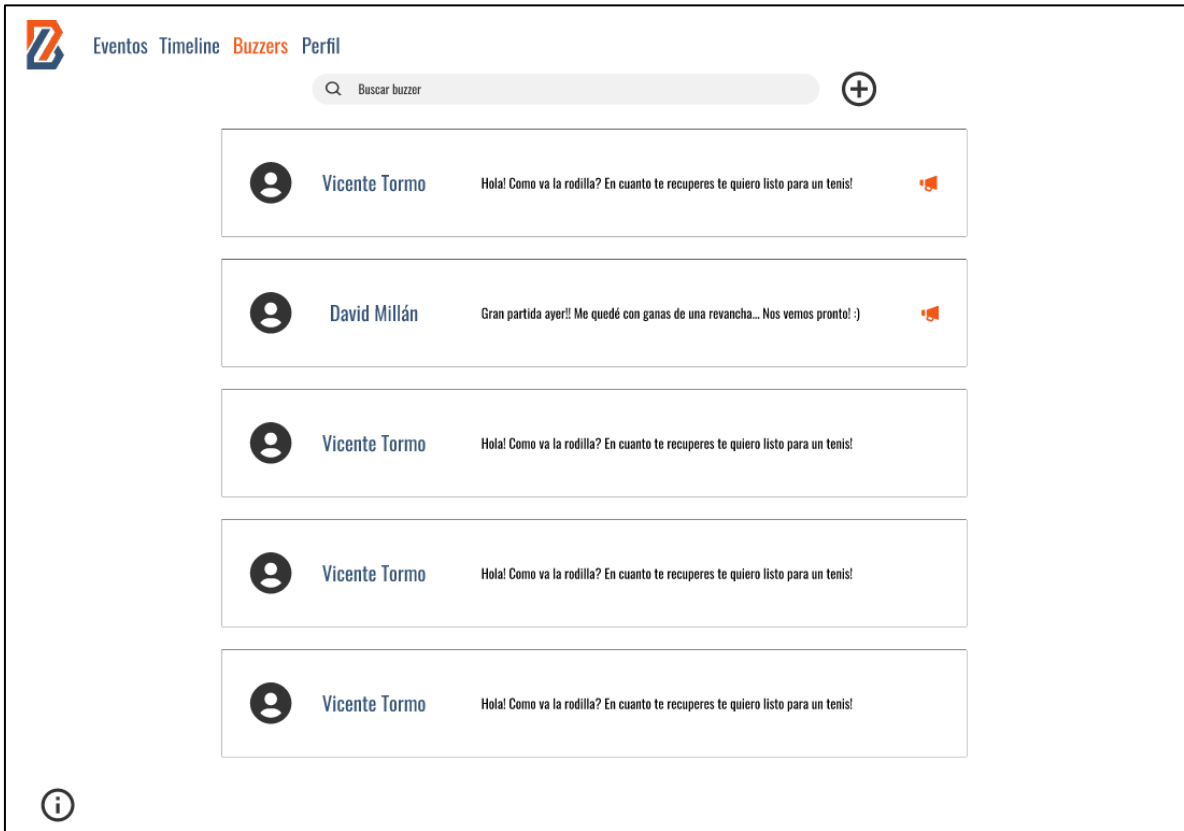


Figura 14. Prototipo Buzzer diseñado en Figma.

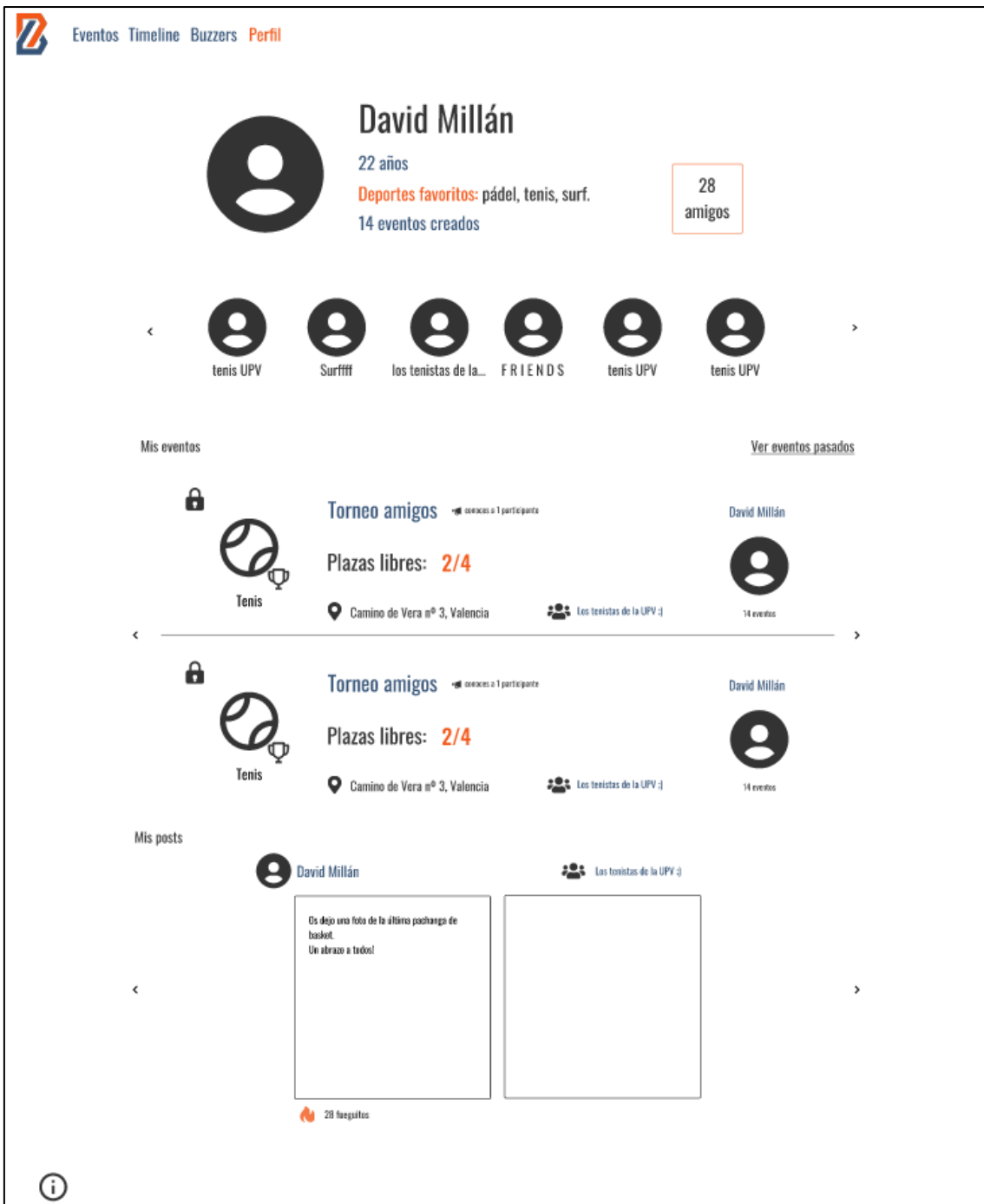


Figura 15. Prototipo del perfil de usuario diseñado en Figma.

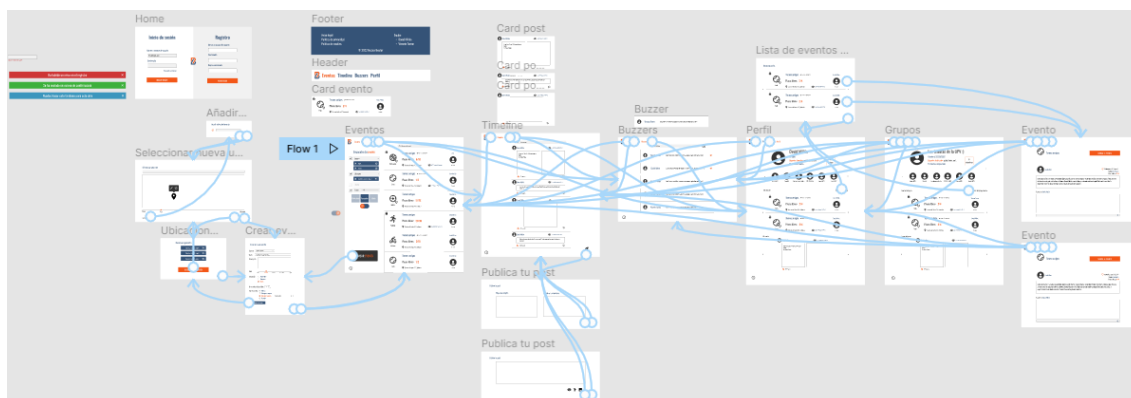


Figura 16. Flujos entre las distintas pantallas del prototipo de Figma.

Como se observará en próximos apartados del trabajo, el aspecto visto en el prototipo no es exactamente el aspecto que finalmente tiene la red social dado que durante el desarrollo fueron cambiados algunos detalles principalmente por ajustar el tamaño de los elementos a los distintos tamaños de pantalla.

5.2 Arquitectura del sistema

En este apartado, se expondrán los principales elementos de la arquitectura del proyecto elaborado con WordPress. Para ello, el apartado se dividirá en tres partes que se desarrollan en este proyecto de *Buzzer Beater* y se expondrán los distintos módulos que forman cada una de ellas.

Antes de ver los subapartados se muestra la siguiente figura donde se puede comprender de un golpe de vista la arquitectura del CMS WordPress.

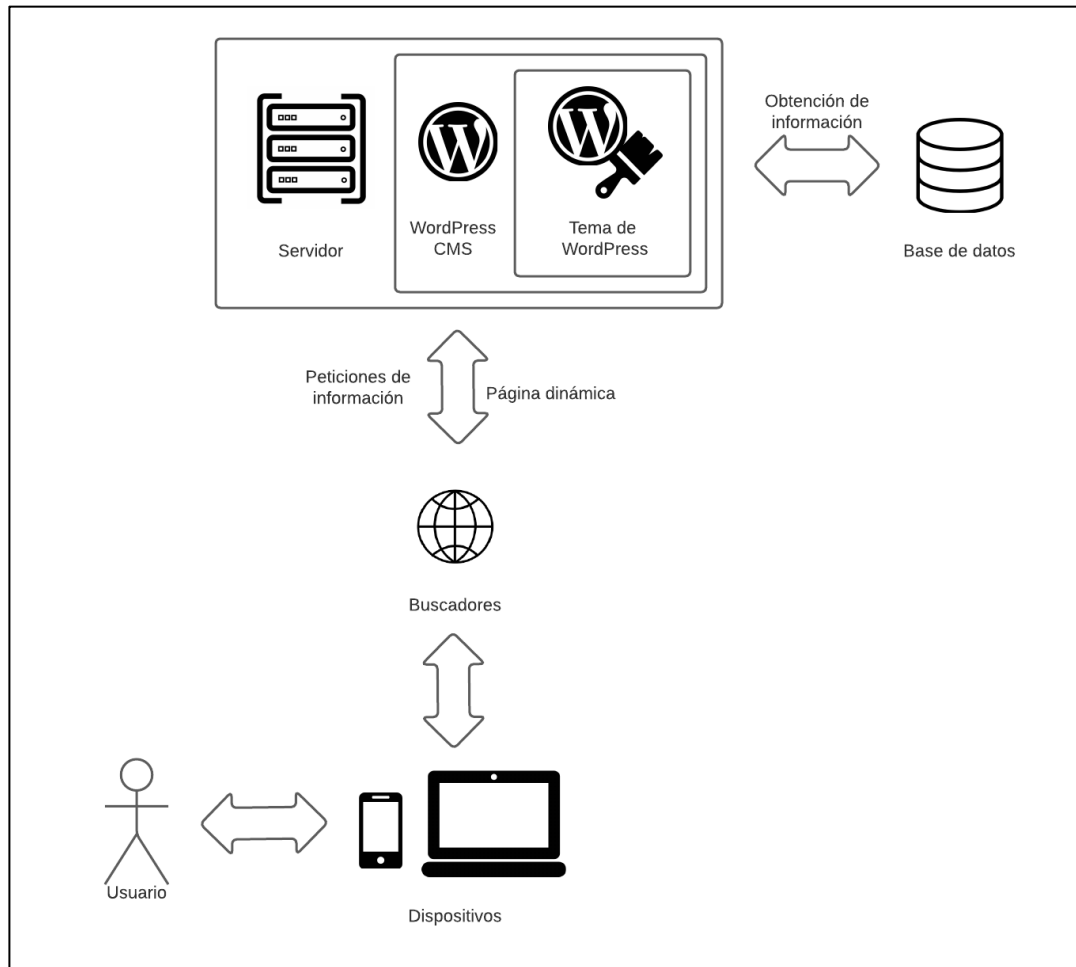


Figura 17. Diagrama con la arquitectura de WordPress. Fuente: Elaboración propia

En la Figura 17 se puede apreciar como la base de datos es un elementos primordial para el funcionamiento de WordPress. El servidor donde esté alojada la página junto con el CMS que a su vez contiene el tema desarrollado realizan las conexiones pertinentes con la BBDD para disponer de toda la información necesaria.

Después, mediante los navegadores, se puede acceder a la página web creada desde cualquier dispositivo y la página se mostrará correctamente si el diseño ha tenido en cuenta los distintos tamaños de pantalla.

5.2.1 Arquitectura de la *timeline*

Como se ha visto en los diseños en el apartado anterior, la *timeline* de *Buzzer Beater* destaca por su diseño simple y limpio. Es por esto por lo que no tiene un gran número de componentes. Los elementos en los que está dividida la *timeline* son básicamente dos: los *posts* y el botón para crear un nuevo *post*.

Los *posts* consisten en tarjetas o *cards* que contienen distintos tipos de información:

- Foto de perfil, nombre y *nick* del usuario que publica el post
- Hace cuanto se publicó
- Botón y contador de fogueitos
- Número de comentarios



El primero de estos elementos se extrae de la información del usuario que publica el *post* y es necesario para identificar quien lo publica. En cuanto al contador de fueguitos (el cual hace la función de los conocidos “me gusta” de las redes sociales), este se incrementa guardando el número en un campo del Custom Post Type de los *posts*. De los Custom Post Types se hablará en el próximo capítulo.

Por último, los comentarios también son almacenados junto con el resto de la información relativa a un mismo *post*. Internamente, estos comentarios se guardan también como *posts*, pero dependen de un *post* principal.

Además, los *posts* pueden ir asociados a algún deporte de los deportes creados dentro de *Buzzer Beater*. Esto será visible mediante la aparición de un pequeño icono en el *post* que se extraerá también del icono asociado al deporte seleccionado.

Con respecto al botón para crear un nuevo *post*, este se encontrará situado en la parte inferior izquierda de la pantalla y al hacer clic sobre él mostrará una ventana emergente que enseñará lo necesario para crear una publicación. Desde esta ventana emergente es desde donde se podrán adjuntar imágenes al *post* las cuales serán almacenadas en la biblioteca de medios de WordPress. En caso de no llegar a publicar el *post*, al cabo de un tiempo, la imagen subida será eliminada de la biblioteca de medios automáticamente.

Además, también se ofrecerá la opción de ajustar la visibilidad del *post*. Se hará asociando el *post* a un grupo de usuarios o a los amigos del usuario que lo publica. Esto se pondrá en práctica mostrando el *post* en cuestión solamente a aquellos usuarios que estén asociados a él.

5.2.2 Arquitectura de los *Buzzers*

Al acceder en el menú a la página de *Buzzers* se puede observar como están organizados en tarjetas, perteneciendo cada tarjeta al chat con un usuario o grupo distinto. De cada usuario se puede ver la foto de perfil, el nombre y apellidos y una previsualización del último mensaje enviado. La información que se muestra de cada usuario es obtenida de la almacenada es el Custom Post Type de cada uno.

También existe una barra de búsqueda superior que filtra los *Buzzers* según el nombre de usuario. Esto lo hace mediante llamadas AJAX (*Asynchronous JavaScript And XML*) las cuales son un método de obtención de datos sin necesidad de recarga de la página. De esta manera, mediante código se puede pedir al servidor información que se utiliza para modificar el aspecto de la página sin una recarga de por medio.

Cada tarjeta de *Buzzer* procede de un mismo archivo el cual se alimenta de otros componentes a su vez. Con un bucle se muestra en la pantalla con las variables pertinentes que son rellenadas para cada usuario según el caso.

Para el funcionamiento del chat se requiere de la creación en la base de datos de una tabla que contenga los mensajes de este. Además, se guardará la información extraída de los usuarios sobre a quienes pertenece este chat. Los mensajes contendrán un identificador propio del chat para que pueda ser recuperado correctamente y mostrado donde debe. Es importante en este caso que la información de los usuarios quede relacionada con un id único para cada chat conformado por el id de cada uno de los dos participantes.



5.2.3 Arquitectura del perfil

El perfil del usuario cuenta con tres grandes módulos: el de información personal, el de los eventos creados y, por último, el de los *posts* creados.

Para la parte de información personal, aparecen distintos elementos como el nombre y apellidos del usuario, su *nick*, su edad, sus deportes favoritos, el número de eventos que ha creado y el número de amigos que el usuario tiene en *Buzzer Beater*. Toda esta información es directamente proporcionada por el usuario. En otros lugares de la red social como los *posts* la información almacenada de cada usuario es utilizada. Por otra parte, la foto de perfil puede ser subida a *Buzzer Beater* por el usuario la cual se almacenará en la biblioteca de medios de WordPress. En caso de no elegir ninguna imagen, la foto de perfil será un icono genérico igual al resto de usuarios sin foto propia.

Dentro de este mismo módulo se encuentran también los grupos de usuarios a los que pertenece el usuario. De nuevo, los grupos a los que pertenece aparecen almacenados junto con la información del usuario y se utiliza para ajustar la visibilidad de ciertos elementos como pueden ser algunos tipos de *post*.

El segundo módulo contiene todos los eventos que ha creado el usuario. La arquitectura de estos eventos es desarrollada en el TFG que complementa a este. Se muestran ordenados cronológicamente extrayendo la fecha de creación de la información del evento mediante un *slider* horizontal que se desplaza con flechas o arrastrando.

Finalmente, el tercer módulo es muy similar al anterior, pero en lugar de mostrar los eventos creados por el usuario se muestran los *posts* que este ha publicado. De nuevo, se ordenan de más reciente a más antiguo gracias a la fecha de creación del *post* y se organizan en un *slider* horizontal que funciona de la misma manera que el previo.

Estos tres módulos se crean en tres archivos distintos los cuales contienen la mayoría del contenido. El archivo principal reúne estos tres módulos y los muestra en pantalla de la manera deseada.



CAPÍTULO 6: DESARROLLO DE LA SOLUCIÓN PROPUESTA

Una vez visto el diseño planteado para desarrollar *Buzzer Beater*, en este capítulo se expondrá como ha sido este desarrollo. Se mostrarán las distintas fases llevadas a cabo, así como algunas de las dificultades que iban surgiendo y cómo se les puso remedio.

6.1 Desarrollo del proyecto

Buzzer Beater ha sido construido sobre el CMS (*Content Management System*) **WordPress**. WordPress es la herramienta más utilizada en el mundo para crear páginas webs. Según ellos mismos, hasta un 43% de internet son páginas webs creadas con este sistema de gestión de contenidos (WordPress, 2022).

WordPress es una herramienta útil para aquellos que deseen crear una página web sencilla y no tengan conocimientos de programación. Por sí sola, ofrece herramientas visuales que permiten al usuario inexperto construir webs de manera rápida. Es el propio CMS el que se encarga de generar todo el código de los elementos incorporados mediante el editor visual. No obstante, para un proyecto tan particular como *Buzzer Beater*, el uso de este editor visual no era una opción. El nivel de detalle y lo particular de esta web hacen que, para llevarla a cabo, se necesitara acceder al código y se modificara y añadieran casi la totalidad del código.

También es habitual, para la parte visual de la página web, hacer uso de los múltiples temas prediseñados que WordPress ofrece, así como librerías de libre acceso para los estilos como Bootstrap. De nuevo, a pesar de que eso habría facilitado el proyecto, no se habría logrado el nivel de personalización que se necesitaba para conseguir la red social que se había diseñado en un primer lugar.

Es por esto por lo que, a pesar del apoyo que las librerías o los temas nativos ofrecerían, se tomó la decisión de no contar con ninguna de esas ayudas. De esta manera, todo el diseño dependería única y exclusivamente del desarrollador y se lograría una mayor comprensión y aprendizaje de las estructuras internas del CMS y de los distintos lenguajes de programación que intervienen en el proceso. En el siguiente apartado, se explicará con más detalle cómo se ha construido la red social y qué elementos de WordPress han sido útiles para ello.

Por otra parte, es importante conocer también los elementos que son necesarios para el funcionamiento de WordPress. Se necesita de un servidor web, un intérprete de PHP y una base de datos MySQL.

Para el servidor web se ha utilizado Apache. Este, se encarga de proporcionar a la página de todos los ficheros necesarios para que se pueda visualizar en un navegador. La herramienta que se utiliza para levantar este servidor web es XAMPP.



XAMPP permite al usuario probar una página web que se base en PHP en el propio ordenador, sin necesidad de utilizar internet. Esto es comúnmente conocido como tener la página web en local. Esto es importante dado que el método de trabajo, tal y como se detallará más adelante, consistió en el desarrollo en local de la página web y la fusión de las distintas partes mediante control de versiones en un repositorio.

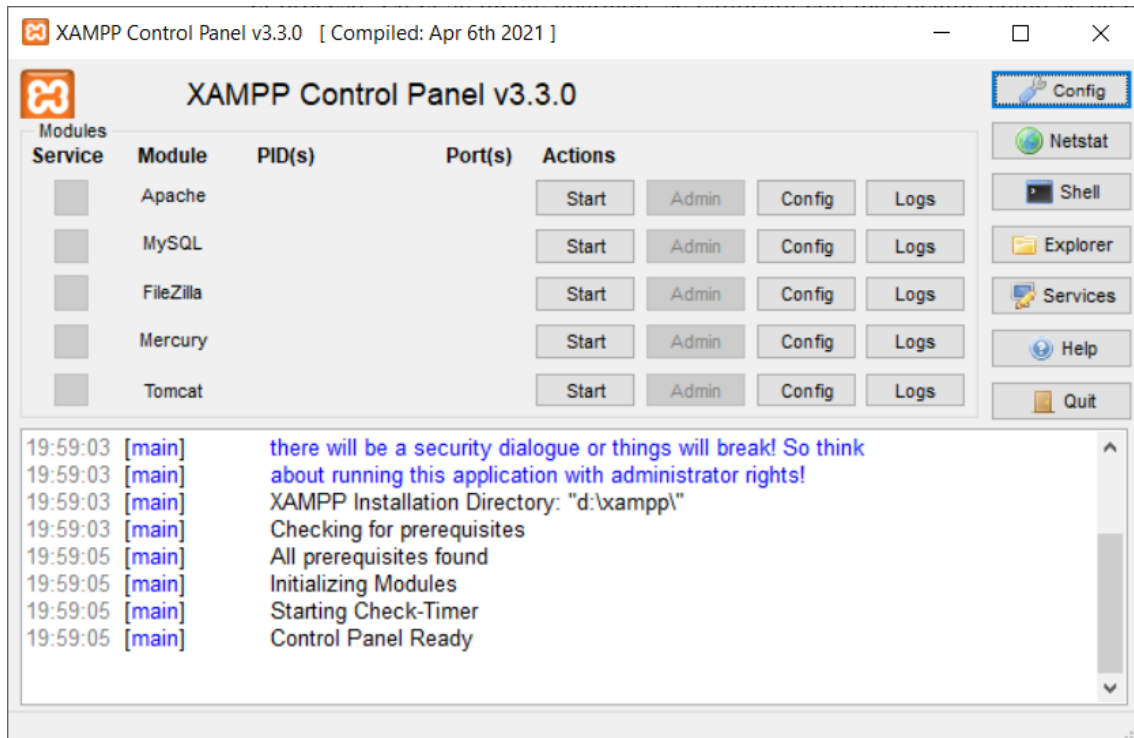


Figura 18. Ventana de XAMPP.

Como se puede observar en la Figura 17, XAMPP ofrece, además de iniciar el servidor Apache, la posibilidad de iniciar otros servicios entre los que se encuentra MySQL.

6.1.1 Estructura de archivos de WordPress

En el apartado anterior se explicó *grosso modo* en qué consistía el sistema de gestión de contenidos WordPress. Ahora se conocerá más en profundidad sobre su estructura, sus partes y cómo ha sido útil para el planteamiento y desarrollo de *Buzzer Beater*.



Para comprender el funcionamiento de WordPress es buena idea conocer primero el esquema de archivos que lo compone. En la siguiente imagen se puede observar la raíz del proyecto donde se ubican los archivos básicos del CMS.

Nombre	Fecha de modificación	Tipo	Tamaño
wp-admin	05/04/2022 0:59	Carpeta de archivos	
wp-content	05/11/2022 11:07	Carpeta de archivos	
wp-includes	05/04/2022 0:59	Carpeta de archivos	
.htaccess	23/08/2022 10:03	Archivo HTACCESS	1 KB
.htaccess.bk	18/02/2022 23:37	Archivo BK	1 KB
index.php	18/02/2022 23:37	Archivo de origen ...	1 KB
license.txt	18/10/2022 14:42	Documento de tex...	20 KB
readme.html	18/10/2022 14:42	Opera Web Docu...	8 KB
wp-activate.php	18/02/2022 23:37	Archivo de origen ...	7 KB
wp-blog-header.php	18/02/2022 23:37	Archivo de origen ...	1 KB
wp-comments-post.php	18/02/2022 23:37	Archivo de origen ...	3 KB
wp-config.php	04/10/2022 9:52	Archivo de origen ...	4 KB
wp-config-sample.php	18/02/2022 23:37	Archivo de origen ...	3 KB
wp-cron.php	23/08/2022 11:26	Archivo de origen ...	4 KB
wp-links-opml.php	23/08/2022 11:26	Archivo de origen ...	3 KB
wp-load.php	23/08/2022 11:26	Archivo de origen ...	4 KB
wp-login.php	23/08/2022 11:26	Archivo de origen ...	48 KB
wp-mail.php	18/10/2022 14:42	Archivo de origen ...	9 KB
wp-settings.php	23/08/2022 11:26	Archivo de origen ...	24 KB
wp-signup.php	23/08/2022 11:26	Archivo de origen ...	32 KB
wp-trackback.php	18/10/2022 14:42	Archivo de origen ...	5 KB
xmlrpc.php	18/02/2022 23:37	Archivo de origen ...	4 KB

Figura 19. Estructura de archivos WordPress.

- Carpeta **wp-content** es la que contiene los temas (en este caso el tema creado para *Buzzer Beater*), los archivos multimedia, los plugins instalados, los idiomas de la web, etc. Todo lo que no se guarda en base de datos está contenido en ella, de ahí su capital importancia.
- Archivo **htaccess** es fundamental para el correcto funcionamiento del servidor web Apache necesario para arrancar WordPress. En él se definen los directorios mediante los cuales se navega dentro de la web. También es necesario para que funcionen las redirecciones, las URL en formato amigable y algunos *plugins* del CMS.



→ Archivo **Index.php** tal y como indican los comentarios del propio archivo este sirve para cargar el tema con el que está hecha la página. Este archivo es el primero que leen los navegadores ya que les indica a que recursos acceder a continuación.

```
index.php X
D: > xampp > htdocs > index.php > ...
1  <?php
2  /**
3   * Front to the WordPress application. This file doesn't do anything, but loads
4   * wp-blog-header.php which does and tells WordPress to load the theme.
5   *
6   * @package WordPress
7   */
8
9  /**
10 * Tells WordPress to load the WordPress theme and output it.
11 *
12 * @var bool
13 */
14 define( 'WP_USE_THEMES', true );
15
16 /** Loads the WordPress Environment and Template */
17 require __DIR__ . '/wp-blog-header.php';
18
```

Figura 20. Archivo *index.php*.

→ Archivo **wp-config.php** que contiene la configuración de los aspectos más elementales del sitio web. Uno de ellos es la conexión con la base de datos de lo cual se ha hablado en el epígrafe anterior.

Los anteriores son algunos de los elementos más importantes de los que intervienen para el correcto funcionamiento de un sitio web basado en WordPress. No obstante, no se puede dejar de mencionar a las carpetas **wp-admin** y **wp-includes**. En la primera se incluyen multitud de archivos del backend del CMS. Todo lo que contiene es no editable. Son instrucciones estáticas que se encargan de hacer funcionar el sistema interno del CMS. En el caso de la segunda, **wp-includes**, contiene la API del CMS además de todas las librerías necesarias para que ofrezca sus funcionalidades. Ambas carpetas se instalan automáticamente al hacer la instalación de WordPress y su contenido no está destinado a ser modificado por el usuario como sí lo está el de la carpeta **wp-content**.

Por último, se debe prestar especial atención a la carpeta **wp-content** ya mencionada. En siguiente captura podemos observar su contenido.



Nombre	Fecha de modificación	Tipo
backups-dup-lite	05/04/2022 1:01	Carpeta de archivo:
languages	18/10/2022 18:37	Carpeta de archivo:
plugins	06/11/2022 12:00	Carpeta de archivo:
themes	06/11/2022 12:00	Carpeta de archivo:
upgrade	18/10/2022 18:37	Carpeta de archivo:
uploads	05/04/2022 0:59	Carpeta de archivo:
index.php	18/02/2022 23:37	Archivo de origen ..

Figura 21. Interior carpeta wp-content.

- Carpeta **languages**: en ella se almacenan todos los archivos de traducción que normalmente corren a cuenta de plugins de terceros, de los temas o, en ocasiones, del propio WordPress. Lo habitual es encontrar en ella multitud de archivos .po y .mo además de .json.
- Carpeta **plugins**: una de las ventajas de WordPress es la gran cantidad de funcionalidad que se puede obtener gracias a la multitud de plugins disponibles. Estos se instalan en esta carpeta y se integran en el CMS. La configuración de estos plugins se almacena en la base de datos por lo que la carpeta solo contiene sus archivos.
- Carpeta **themes**: en ella están los distintos temas instalados en la página web tanto si se encuentran activos en ese momento como si no. Más adelante se observará la estructura interna del tema creado para esta web el cual se aloja en esta carpeta.
- Carpeta **upgrade**: normalmente se encuentra vacía, pero en ella se guardan algunos archivos sensibles durante las actualizaciones del CMS.
- Carpeta **uploads**: en ella se almacena el contenido multimedia que se sube a la página web.

6.1.2 Estructura de la base de datos de WordPress

WordPress precisa de una base de datos para su funcionamiento. Esto es debido a que WordPress por “detrás” tiene una compleja estructura que debe almacenar de una manera específica y prediseñada que el CMS aporta.

WordPress utiliza el gestor de bases de datos MySQL. Esto significa que cuando desea hacer cualquier operación sobre la base de datos realiza una consulta MySQL sobre esta. Al realizar la instalación de este CMS, se ha de indicar la base de datos sobre la que se desea que opere WordPress. Una vez hecho, cuando se ha completado la instalación, hay una estructura creada y una serie de tablas en ella. Cuenta con doce tablas de las cuales se utilizan solamente once que son las siguientes:

- wp_comments: en la que se almacenan los comentarios de los usuarios a las entradas de la página.
- wp_commentmeta: contiene información adicional sobre los comentarios. Se utiliza por algunos plugins para marcar comentarios como spam, por ejemplo.



- `wp_post`: en ella se almacena la información referente a los *posts* o distintos tipos de contenido que se publique. Los Custom Post Types de los que se hablará a continuación van incluidos en ella.
- `wp_postmeta`: contiene información adicional sobre los posts.
- `wp_user`: información relativa a los usuarios de la web (login, correo, contraseña).
- `wp_usermeta`: contiene información adicional de los usuarios (nombre, apellido, rol, preferencias)
- `wp_terms`: contiene la información relativa a las etiquetas y categorías que se creen en la página.
- `wp_termmeta`: contiene información adicional sobre la tabla `wp_terms`.
- `wp_options`: esta tabla contiene los datos de la configuración de la página web, así como información sobre los plugins o el tema empleado.
- `wp_term_taxonomy`: se almacena la tipología de la taxonomía asociada a cada registro de la tabla `wp_terms`.
- `wp_term_relationship`: relaciona la taxonomía de un registro de contenido.

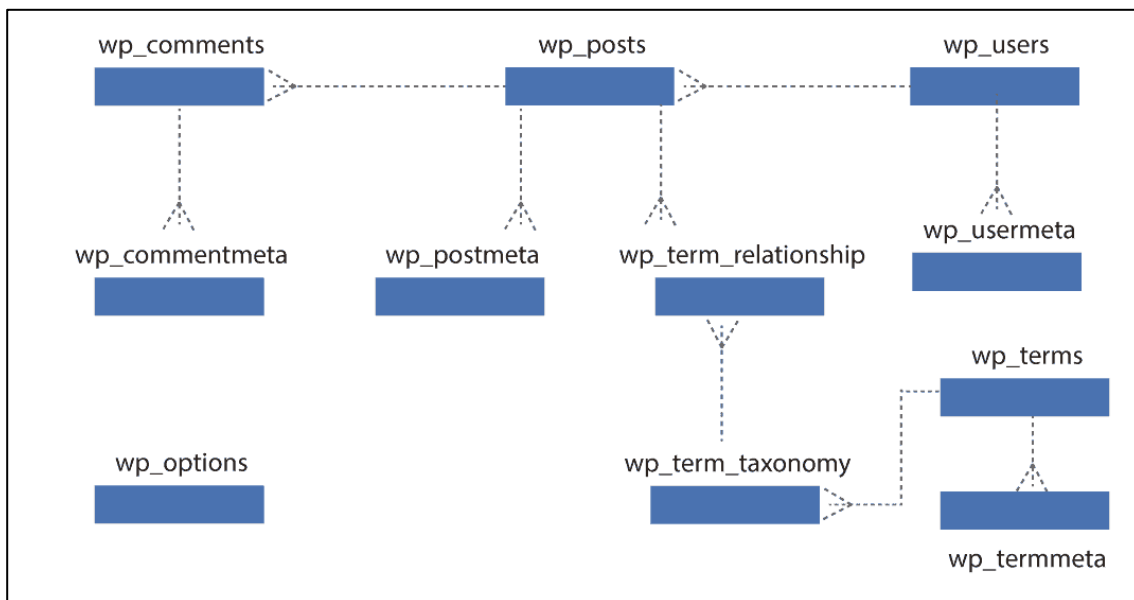


Figura 22. Estructura base de datos de WordPress Fuente: (Guzmán, 2019).

La tabla faltante es la tabla `wp_links` la cual en la actualidad se encuentra en desuso y solamente se mantiene en la estructura de las bases de datos de WordPress por asuntos de compatibilidad con versiones anteriores del CMS.

6.1.3 Custom Post Types

Un sitio WordPress recién instalado cuenta con una serie de requisitos básicos que suelen ser insuficientes para satisfacer la mayoría de las páginas web. En un principio, solamente ofrece páginas y entradas tipo blog como posibilidades para publicar tu contenido.

Buzzer Beater tiene muchos más requisitos y tipos de información que son imposibles de mostrar mediante solamente entradas y páginas. Es necesario por tanto crear nuevas estructuras de información a medida que encajen con lo que se desea para así lograr los objetivos.



Para ello se ha utilizado un plugin llamado Advanced Custom Fields Pro. Gracias a él se han podido generar los modelos de información necesarios para poder representar dentro de *Buzzer Beater* los usuarios o los *posts* por ejemplo con todos los parámetros deseados.

Advanced Custom Fields PRO

Descripción Instalación Registro de cambios Upgrade Notice

Advanced Custom Fields turns WordPress sites into a fully-fledged content management system by giving you all the tools to do more with your data.

Use the Advanced Custom Fields plugin to take full control of your WordPress edit screens & custom field data.

Add fields on demand. Our field builder allows you to quickly and easily add fields to WP edit screens with only the click of a few buttons!

Add them anywhere. Fields can be added all over WP including posts, users, taxonomy terms, media, comments and even custom options pages!

Show them everywhere. Load and display your custom field values in any theme template file with our hassle free developer friendly functions!

Versión: 6.0.4
Autor: [WP Engine](#)
Última actualización: hace 3 días
Necesita la versión de WordPress: 4.7 o superior
Compatible con: 6.1
Necesita la versión de PHP: 5.6 o superior
[Página principal del plugin »](#)

COLABORADORES

[ACT Advanced Custom Fields](#)

Figura 23. *Advanced Custom Fields PRO* para la creación de *Custom Post Types*.

Para entender mejor en qué ayudan exactamente los Custom Post Types a una página web como la desarrollada, se explicará a continuación detalladamente el diseño de los creados para los *posts* y para los usuarios.



→ Posts:

Editar Grupo de Campos [Añadir nuevo](#)

Social posts parameters

Orden	Etiqueta	Nombre	Tipo
1	Attachments	attachments	Relación
2	Post text	post_text	Editor Wysiwyg
3	Related Sport	related_sport	Objecto Post
4	Visibility	visibility	Elige
5	Fueguitos	fueguitos	Usuario
6	In reply to	in_reply_to	Objecto Post
7	Group	group	Objecto Post

[+ Añadir Campo](#)

Ubicación

Reglas
Crea un conjunto de reglas para determinar qué pantallas de edición utilizarán estos campos personalizados

Mostrar este grupo de campos si

Post Type es igual a Social post

o

[Agregar grupo de reglas](#)

Figura 24. Página de edición del Custom Post Type de los posts.

Este cuenta con los siguientes campos:

- Adjuntos (*Attachments*): para permitir adjuntar el contenido multimedia deseado a cada post.
- Texto (*Post text*): párrafo de texto que el usuario puede escribir si así lo desea.
- Deporte relacionado (*Related Sport*): cada post puede ser asociado a uno de los deportes existentes en *Buzzer Beater*.
- Visibilidad (*Visibility*): es posible restringir la visibilidad de un *post* para que solo sea visto por los amigos del usuario o por los miembros de determinados grupos en lugar de que todo el mundo lo pueda ver en su *timeline*.
- Fueguitos: son lo que en otra red social se conoce como “Me Gusta”. De esta manera se permite al *post* llevar un contador de los fueguitos que lleva cada *post*.
- Respuestas (*In reply to*): permite responder a otros *posts*. A nivel interno, cada una de estas respuestas es otro objeto *post*.
- Grupo (*Group*): este campo se muestra solo en los *posts* si el campo *Visibility* está seleccionado como solo para grupos. En ese caso permite seleccionar a qué grupo se muestra el *post*.

→ **Parámetros del usuario:**



Orden	Etiqueta	Nombre	Tipo
1	Friends	friends	Usuario
2	Groups	groups	Objeto Post
3	Favourite sports	favourite_sports	Objeto Post
4	Saved custom locations	saved_custom_locations	Repeater
5	Birth date	birth_date	Selector de Fecha
6	User Picture	user_picture	Imagen
7	Draft social post	draft_social_post	Objeto Post
8	Abandoned uploaded files	abandoned_uploaded_files	Relación

[+ Añadir Campo](#)

Ubicación

Reglas
Crea un conjunto de reglas para determinar qué pantallas de edición utilizarán estos campos personalizados

Mostrar este grupo de campos si

es igual a y

o

Figura 25. Página de edición del Custom Post Type de los parámetros de usuario..

Este cuenta con los siguientes campos:

- Amigos (*Friends*): permite guardar para cada usuario que otros usuarios son sus amigos en la red social.
- Grupos (*Groups*): permite conocer a qué grupos pertenece cada usuario.
- Deportes favoritos (*Favourite Sports*): aquí se guardan cuales de los deportes existentes dentro de *Buzzer Beater* son marcados como favoritos por el usuario.
- Localizaciones personalizadas (*Saved Custom Locations*): de esta manera, el usuario puede renombrar las ubicaciones que desee donde practique deporte o quiera crear un evento en ellas.
- Fecha de nacimiento (*Birth Date*): se guarda la fecha de nacimiento indicada por el usuario.
- Foto de perfil (*User Picture*): si el usuario lo desea, puede establecer una foto como foto de perfil. En su defecto, se pone un icóno genérico de foto de perfil.
- Borrador de *post* (*Draf Social Post*): para guardar posts que no hayan llegado a publicarse.
- Archivos multimedia abandonados (*Abandoned uploaded files*): también relacionado con los borradores, guarda el contenido multimedia de estos.



Cada uno de los campos de los Custom Post Types tiene en su interior una configuración en la cual se puede seleccionar que tipo de objeto es cada campo, cuando debe aparecer, que tipo de información se guarda en él, si se permite que esté vacío o no, etc.

Toda esta configuración se puede hacer de manera sencilla y visual con el plugin de Advanced Custom Fields PRO pero también desde el archivo en el que se guarda toda la información: el **functions.php**. Este archivo también es útil para consultar exactamente el nombre que se le da a cada campo o a cada Custom Post Type con la finalidad de poder llamar durante el desarrollo a estos elementos para colocarlos donde se desee.

6.1.4 Base de datos común

Al principio del proyecto, la base de datos fue instalada en local y gestionada mediante la aplicación web phpMyAdmin. No obstante, debido a problemas de coherencia entre los datos (los cuales se expondrán más adelante en el TFG), se decidió optar por una base de datos común, alojada en la nube más concretamente en Hostinger.

Hostinger es una herramienta que ofrece servicios de alojamiento web y de dominios web. Esta herramienta permite la creación de una base de datos MySQL de manera sencilla.

The screenshot shows the Hostinger MySQL database creation interface. At the top, there is a breadcrumb trail: "Bases de datos MySQL" > "Hosting" > "Bases de datos" > "Bases de datos MySQL". Below this, there is a heading: "+ Crear nueva base de datos MySQL y usuario de base de datos". The form contains three input fields: "Nombre de la base de datos MySQL" with the value "u536365345_", "Nombre de usuario MySQL" with the value "u536365345_", and "Contraseña" with the value "Contraseña". A green "Crear" button with a checkmark is highlighted with a red box, and a red arrow points to it from the right.

Figura 26. Creación base de datos en Hostinger.

Una vez creada, ya se puede comenzar a administrar con la multitud de herramientas que Hostinger ofrece.

En esta base de datos recién creada, será necesario realizar una instalación nueva de WordPress o importar una base de datos ya existente para que se cree en ella la estructura necesaria de tablas y relaciones con la finalidad de que el sitio web basado en este CMS funcione adecuadamente.

Las ventajas que ofrece poseer una base de datos común al equipo están principalmente orientadas hacia la coherencia de los datos y sus estructuras. De esta manera, se realizarán las pruebas con los mismos datos en las distintas partes de la página web lo que permite probar de una forma más sólida los mismos escenarios a lo largo de todas las funcionalidades.



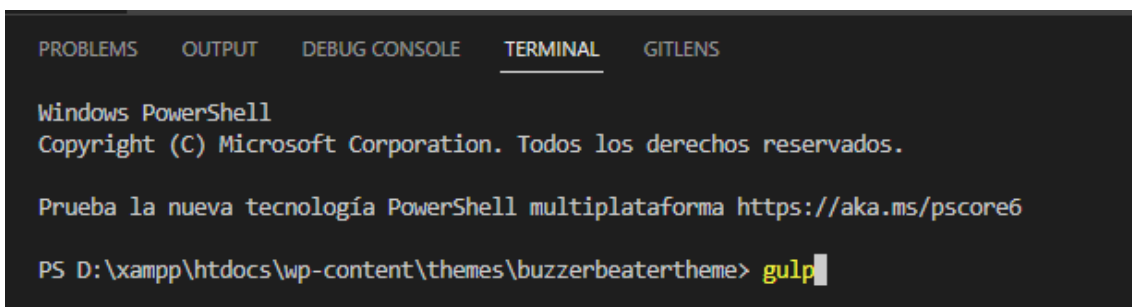
6.1.5 Proceso de desarrollo

En epígrafes anteriores se ha visto el diseño del proyecto, tanto el bocetaje inicial como el posterior prototipado, y es desde ese punto desde el que se parte para desarrollar la red social. Los primeros pasos, como se vio también en el subapartado anterior, fueron los de conocer y comprender la estructura del CMS WordPress. De esta manera, se detectó la necesidad de implementar los Custom Post Types y de establecer una estructura de archivos propia. Es importante que todo esto fuese establecido de antemano, no solo para evitar retroceder a fases de diseño cuando se esté desarrollando sino también para que el trabajo en equipo fuese lo más fluido posible.

La forma de trabajar ha sido mediante la metodología Scrum. Con ella, se hizo una división en pequeñas tareas de los requisitos de la aplicación. Una vez se contó con una lista de tareas pequeñas que podían ser acometidas individualmente, se comenzó a organizar el primer sprint.

Una vez iniciado el sprint, el desarrollo discurría individualmente en la máquina local de cada uno. En el caso particular de este TFG, se desarrollaba de las partes más grandes a las más pequeñas de cada elemento que tocaba generar. Es decir, se creaba primero una estructura grande y se seguía completando su interior paso a paso. Para probar el código había que iniciar mediante consola el compilador Gulp.js. Este compilador era el responsable de compilar todo el código escrito en SASS y JavaScript.

Se configuró de tal manera que bastaba con situarse en la carpeta de *Buzzerbeatertheme* en la consola integrada en Visual Studio Code, y se ejecutaba la línea “gulp”.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS

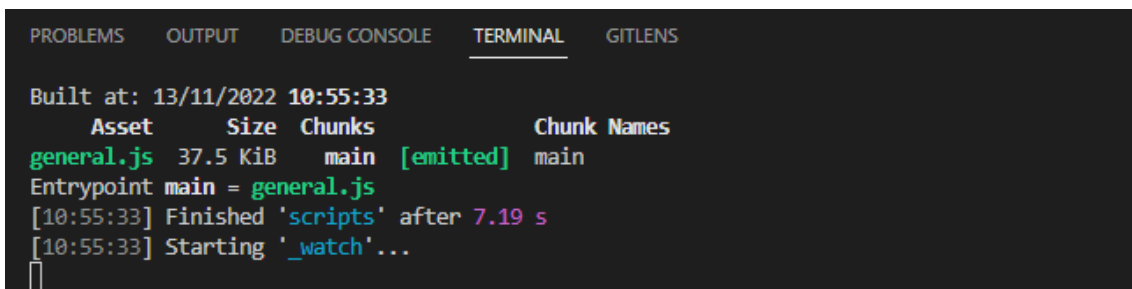
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS D:\xampp\htdocs\wp-content\themes\buzzerbeatertheme> gulp
```

Figura 27. Línea de comandos integrada en Visual Studio Code..

Una vez ejecutado, se esperaba unos segundos a que arrancara y se pusiera en modo “watch”, esto significaba que cada vez que se grabara un cambio en un fichero .sass, .css o .js, Gulp.js automáticamente compilaría los ficheros sin ninguna acción más requerida.



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  GITLENS

Built at: 13/11/2022 10:55:33
   Asset      Size  Chunks             Chunk Names
general.js  37.5 KiB   main  [emitted]  main
Entrypoint main = general.js
[10:55:33] Finished 'scripts' after 7.19 s
[10:55:33] Starting '_watch'...
```

Figura 28. Línea de comandos integrada en Visual Studio Code, Gulp.js iniciado.



De esta manera, para visualizar un cambio en la página, bastaba con guardar los archivos modificados y recargar la página en el navegador. En ocasiones, era necesario recargar sin caché debido a que no se visualizaban los cambios aplicados.

Cuando una tarea del sprint era finalizada, lo habitual era utilizar la herramienta de control de versiones Git para subir los ficheros modificados con la nueva versión del código al repositorio común alojado en BitBucket. Git era utilizado principalmente mediante los botones accesos directos que otorga Visual Studio Code para ello, aunque en ocasiones se utilizaba también la consola integrada. Es importante mencionar que para cada tarea o conjunto de tareas relacionadas se creaba una nueva rama de trabajo de manera que durante el desarrollo de un sprint no hubiera conflicto alguno.

Una vez finalizado el sprint, se realizaba un *merge* o unión de códigos. El contenido de las ramas en las que se había estado trabajando se subía a la rama principal y se resolvían los conflictos en aquellos archivos que habían sido modificados en distintas ramas.



Figura 29. Visualización de las ramas de trabajo en BitBucket.

Finalmente, para terminar de exponer el desarrollo del proyecto, una dificultad añadida fue la de que toda la página web tuviese un diseño *responsive*, es decir, que se adaptara a los distintos tamaños de pantalla desde donde iba a ser visualizada. Esto obligaba a probar como se veía cada nuevo añadido en múltiples tamaños de pantalla. Además, al no contar con ayudas de librerías como Bootstrap que ofrece elementos que automáticamente se adaptan a cualquier pantalla, se tuvo que desarrollar en muchas ocasiones íntegramente una versión de cada elemento para cada tamaño.



Para ello, se crearon etiquetas SASS en las cuales se podían contener las líneas de código que afectaban al diseño del elemento solo para un tamaño de pantalla en concreto. En la siguiente imagen se muestra una captura de cómo fueron definidas algunas de estas etiquetas.

```
$tablet-width: 768px;
$small-desktop-width: 900px;
$desktop-width: 1024px;

@mixin mobile {
  @media (max-width: #{$tablet-width - 1px}) {
    @content;
  }
}

@mixin tablet-mx {
  @media (min-width: #{$tablet-width}) and (max-width: #{$desktop-width - 1px}) {
    @content;
  }
}

@mixin tablet {
  @media (min-width: #{$tablet-width}) {
    @content;
  }
}

@mixin small-desktop {
  @media (min-width: #{$small-desktop-width}) {
    @content;
  }
}

@mixin desktop {
  @media (min-width: #{$desktop-width}) {
    @content;
  }
}
```

Figura 30. Código de creación de etiquetas para especificar tamaño de pantalla.

6.2 Resolución de adversidades

Durante el desarrollo del proyecto, se tuvieron que afrontar numerosos contratiempos debido a la extensión de este, el desconocimiento de muchas de las tecnologías, errores en el código, etc. Para poder hacer frente a ellos, lo primero a lo que se acudía era a internet, donde mediante tutoriales o respuestas en foros se lograba en la mayoría de los casos solventar el problema.

Por supuesto, en las reuniones periódicas que se tenían, algunos de estos problemas eran expuestos y se les lograba poner solución en el momento o se concertaba una reunión en otro momento para buscar maneras de solucionar o rodear la situación.

Como se ha comentado anteriormente, los errores de coherencia debido a la disparidad de los datos que se tenían en las bases de datos locales llevaron a la necesidad de crear una única base de datos en la nube que estuviera vacía y en la que se generaran datos nuevos.



Dado que es necesario para el funcionamiento de WordPress, al principio del proceso se comenzó arrancando también una base de datos local con XAMPP. No obstante, a mitad del desarrollo se decidió cambiar a una base de datos común y en la nube. Esto se decidió debido a los problemas de coherencia que se sufrían al tener distintos datos en cada base de datos lo cual dio lugar a diversos errores durante el principio del proceso.

En primer lugar, se creó una base de datos en el proveedor de almacenamiento web Hostinger (véase sección 5.2.4). Más adelante, se realizó una importación de una de las bases de datos ya existentes y se procedió al vaciado de todos los datos. Una vez hecho esto, para cambiar la base de datos a una en la nube basta con configurar adecuadamente el archivo *wp-config.php* cambiando los valores de los campos *DB_NAME*, *DB_USER*, *DB_PASSWORD* y *DB_HOST* a los que correspondan a la nueva base de datos. Finalmente, se comenzó a poblar la nueva base de datos con conjuntos de datos más fiables y completos que permitieron avanzar con el desarrollo de una forma más coherente y sólida.

También se tuvieron problemas en algunas ocasiones con el control de versiones Git. Sucedió durante el desarrollo que se subían al repositorio común versiones del código que no eran las adecuadas y que hacían fallar la aplicación. Para resolver este tipo de incidencias hubo que profundizar en los conocimientos que se tenían de Git y se descubrieron maneras de retroceder a versiones anteriores o de restaurar líneas de código eliminadas.

Estos contratiempos sumados a la dificultad añadida de realizar todo el diseño *responsive* sin ayuda de librerías especializadas, o el mal dimensionamiento de algunas de las tareas de los *sprints* llevaron a retrasos en la programación temporal frente a los cuales hubo que adaptarse, redefinir plazos y establecer nuevas formas de trabajo.



CAPÍTULO 7: IMPLANTACIÓN

Una vez visto tanto el diseño de la red social *Buzzer Beater* como el desarrollo del proyecto con sus contratiempos incluidos, queda por mostrar el resultado final y como ha quedado el proyecto. En este capítulo además de mostrar esto, se facilitará información relativa a cómo sería el despliegue de la aplicación en un servidor real al que cualquiera pudiera acceder.

7.1 Resultado final

Una vez acabado el desarrollo, en este epígrafe se mostrarán como han quedado las distintas páginas realizadas en este Trabajo de Fin de Grado.

Timeline

En las siguientes imágenes se muestra el resultado final de la *timeline* de *Buzzer Beater*:

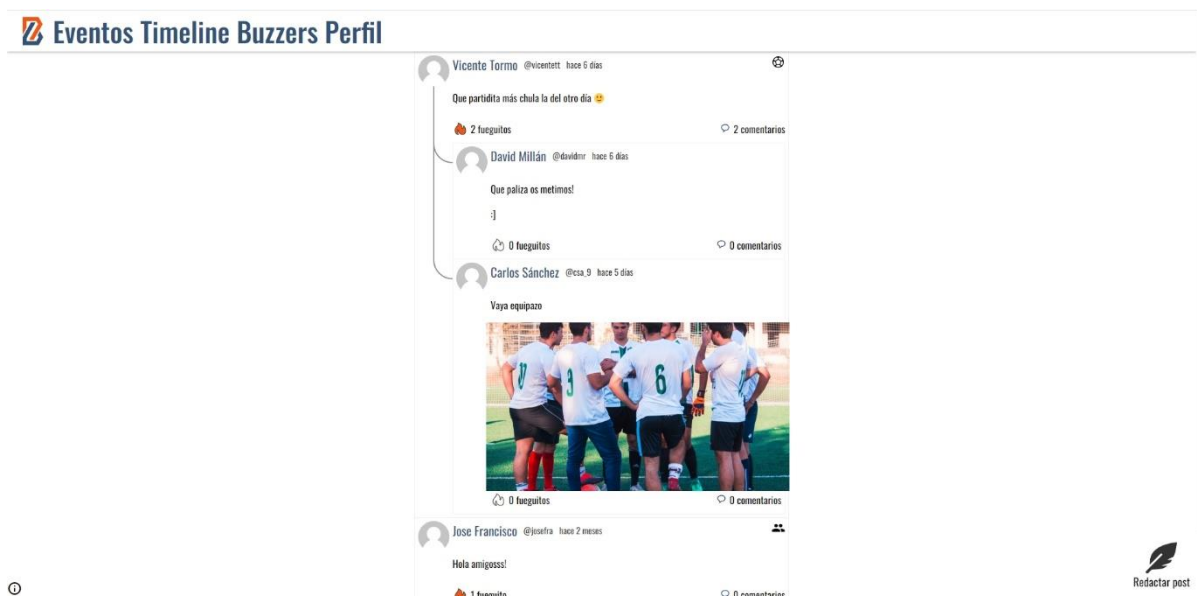


Figura 31. Captura timeline de Buzzer Beater.

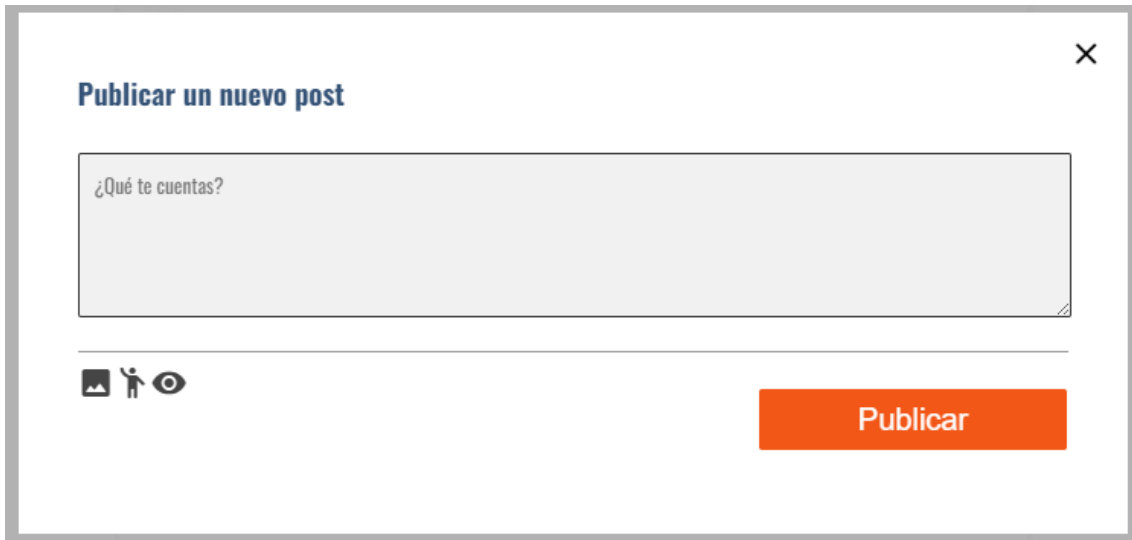


Figura 34. Captura ventana emergente creación nuevo post.

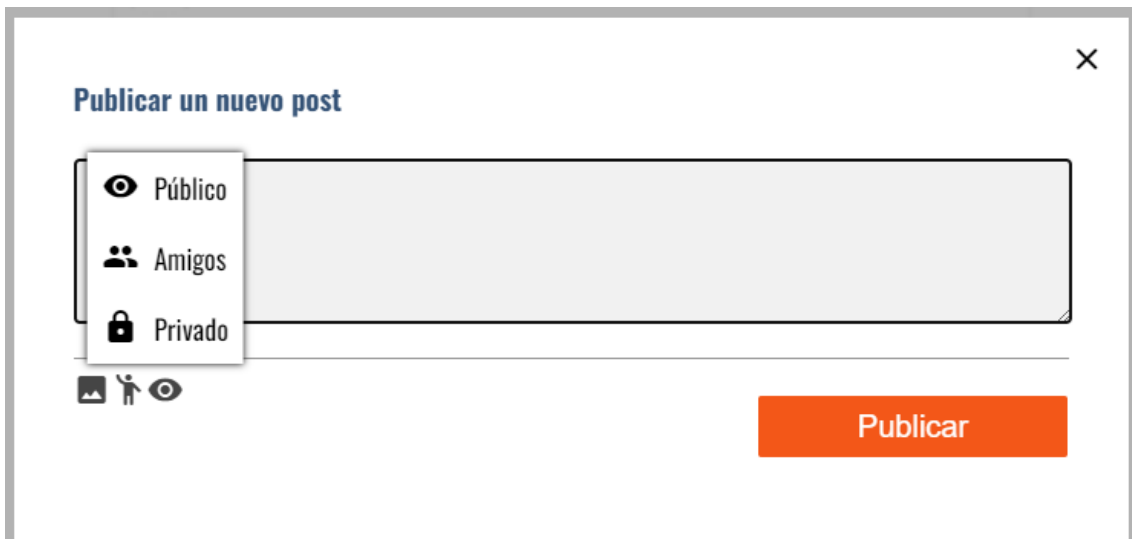


Figura 33. Captura ventana emergente creación nuevo post ajustes de visibilidad.



Figura 32. Captura ventana emergente creación nuevo post selección de deporte asociado.



Figura 35. Detalle botón "Redactar post"

Buzzers

En las siguientes imágenes se muestra el resultado final de los *Buzzers* de *Buzzer Beater*:

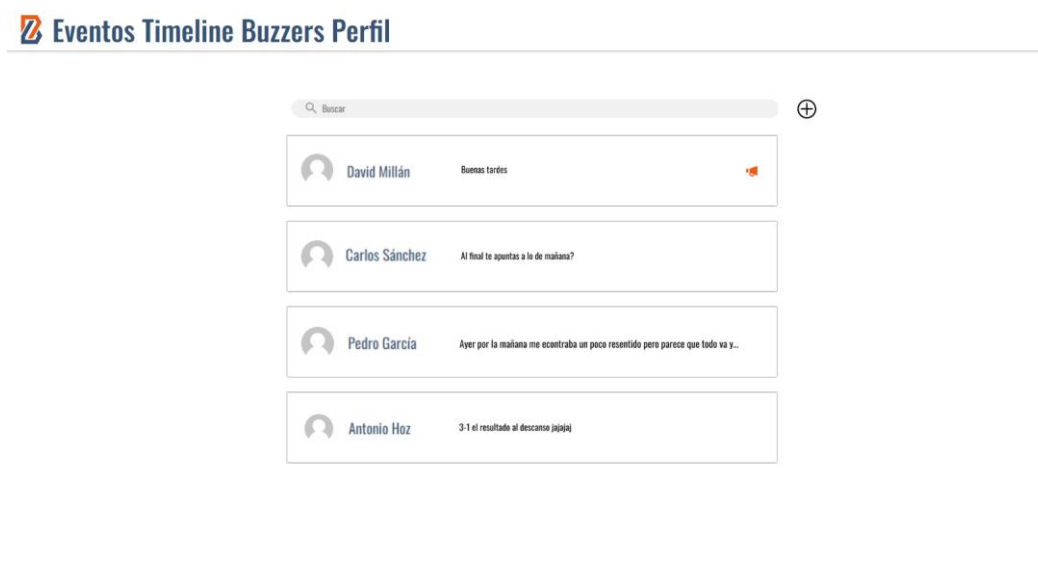


Figura 37. Captura pantalla Buzzers.



Figura 36. Detalle card de un Buzzer.



Perfil de usuario

En las siguientes imágenes se muestra el resultado final del perfil de un usuario de *Buzzer Beater*:

Eventos Timeline Buzzers Perfil

The screenshot shows a user profile for Vicente Tormo. At the top, there is a circular profile picture of a person in a dark, atmospheric setting. To the right of the picture, the name 'Vicente Tormo' is displayed in a large, bold font. Below the name, it says '23 años', 'Deportes favoritos: tenis, pádel', and '4 eventos creados'. To the right of this information is a box that says '1 amigos' and a gear icon for settings. Below the profile information is a horizontal scrollable list of event categories: 'Basket street', 'CaféNis', 'Basket 3-3', 'Pinpen club', 'Partida con Roger', 'Partida con Rata', and 'Bajket'. Underneath this is a section titled 'Eventos creados:' which contains two event cards. The first card is for 'Futbol' on '07-12-2022 00:18' with 'Plazas libres: 19/22' and location 'Ágora'. The second card is for 'Partida tenis de mesa' on '14-12-2023 20:00' with 'Plazas libres: 0/2' and location 'UPV -Valencia'. Below the events is a section titled 'Posts publicados:' showing a post from 'Vicente Tormo @vicentett' from 'hace 2 meses' with the text 'Qué partidita más chula el otro día 😊'.

Figura 38. Captura del perfil de un usuario en Buzzer Beater.

This image shows a detailed view of the top part of the user profile for Vicente Tormo. It features a circular profile picture on the left. To its right, the name 'Vicente Tormo' is prominently displayed. Below the name, the text reads '23 años', 'Deportes favoritos: tenis, pádel', and '4 eventos creados'. To the right of this text is a box containing '1 amigos' and a gear icon for settings.

Figura 39. Detalle módulo superior del perfil de un usuario en Buzzer Beater.



7.2 Despliegue

El último paso en el desarrollo de un proyecto de estas características es el del despliegue de la página web en un servidor y dominio real para que pueda ser indexado por los buscadores de internet y visitado por todo aquel que lo desee.

En primer lugar, se ha de contratar un servidor web o *hosting* en el que alojar la página web junto con toda la información que contiene. Posteriormente, es necesario también obtener un dominio web para que pueda ser encontrada por los buscadores. En el caso de *Buzzer Beater* ambos servicios son provistos por Hostinger.

Una vez se tiene el dominio, este permite crear tanto subdominios como se desee. Por lo general, el servidor suele contener una carpeta llamada `public_html` la cual es el dominio principal de la página web. Las distintas carpetas alojadas en el interior de esta son los subdominios que nacen del principal.

Una vez se ha creado una carpeta nueva dentro del servidor para contener la página web, se ha de realizar una copia de la página web existente para subirla al servidor. En el caso de *Buzzer Beater* esto se elige hacer mediante el plugin de WordPress Duplicator.

Duplicator
Migrations & Backups

Duplicator – WordPress Migration Plugin

Descripción [FAQ](#) [Registro de cambios](#) [Capturas de pantalla](#) [Valoraciones](#)

Con más de 30 millones de descargas, Duplicator ofrece a los usuarios de WordPress la capacidad de migrar, copiar, mover o clonar correctamente un sitio de una ubicación a otra y también sirve como una sencilla utilidad de copia de seguridad. Duplicator maneja tanto datos serializados como sustitución de cadenas base64 serializadas. La migración estándar de WordPress y las copias de seguridad de WordPress se manejan fácilmente con este plugin, al igual que las migraciones sin interrupciones.

Para más detalles visita snapcreek.com.

Vídeo de demostración rápida

Resumen

Duplicator es el más potente migrador disponible. Te permite:

- Mover, migrar o clonar un sitio WordPress entre dominios o alojamientos **sin interrupciones**
- Desplegar un sitio en producción en localhost para su desarrollo
- Transfiere un sitio de WordPress de un alojamiento a otro

Versión: 1.5.1
Autor: [Snap Creek](#)
Última actualización: hace 5 días
Necesita la versión de WordPress: 4.0 o superior
Compatible con: 6.1.1
Necesita la versión de PHP: 5.3.8 o superior
Instalaciones activas: 1+ millón
[Página del plugin en WordPress.org »](#)
[Página principal del plugin »](#)

PUNTUACIÓN PROMEDIO

★★★★★
(basado en 3.639 valoraciones)

VALORACIONES

¡Lee todas las valoraciones en WordPress.org o escribe la tuya!

[5 estrellas](#) 3.485
[4 estrellas](#) 49

Figura 40. Plugin de WordPress Duplicator.



Este plugin permite realizar una copia de la totalidad de la página web y toda su información para poder instalarla en otro lugar. Cuando se crea una copia con Duplicator de un sitio web, se generan dos archivos que contienen toda la información de la web.

Para subir estos archivos al servidor se ha de emplear el protocolo de transferencia de red conocido como FTP. Para utilizarlo se suele emplear el programa FileZilla. Gracias a él se puede conectar con el servidor y simplemente arrastrar una carpeta de local dentro de una carpeta del servidor.

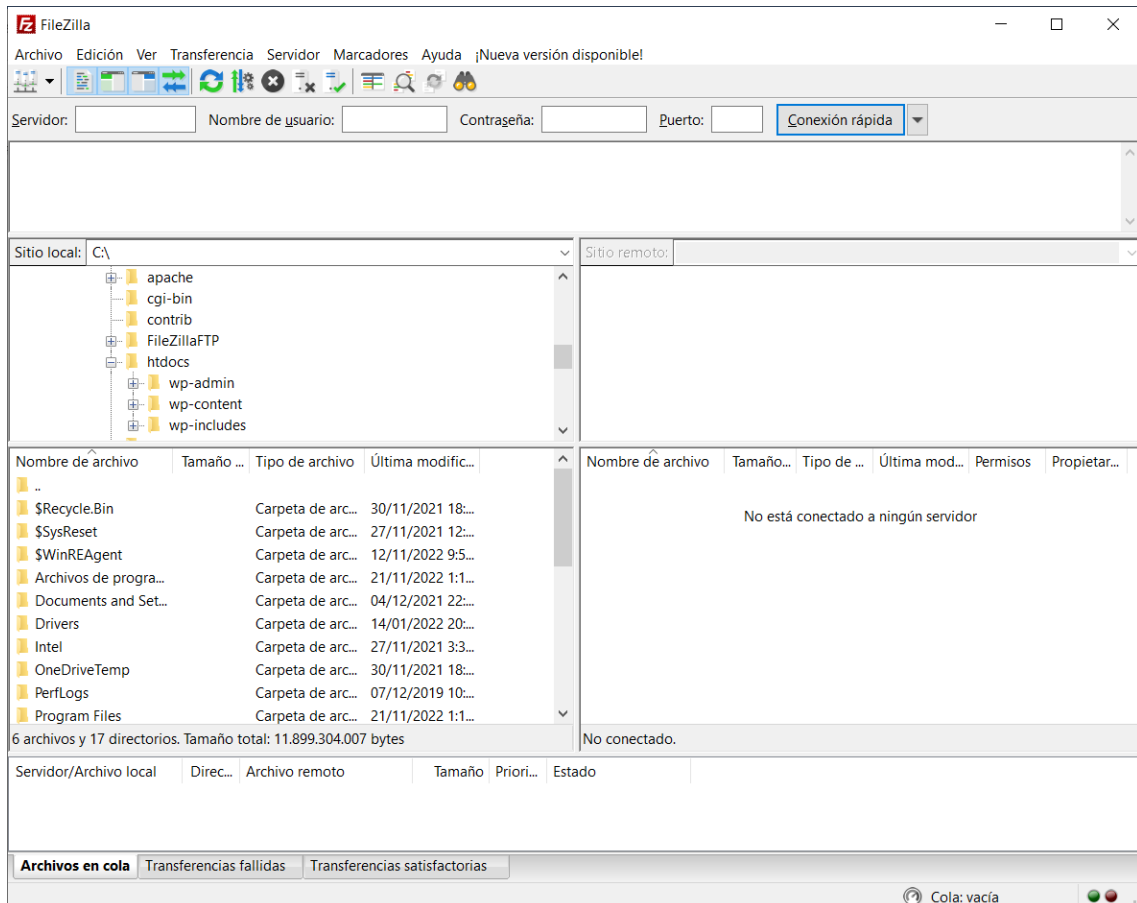


Figura 41. Captura de FileZilla.

Dado que la base de datos en la nube en el caso de este proyecto ya estaba creada durante el desarrollo no será necesario crearla de nuevo. Por tanto, bastaría con acceder en el navegador a la ruta donde se encuentren alojados los archivos y añadir al final `/installer.php`. De esta manera, se ejecutaría el lanzador de Duplicator que replicaría la página web en el servidor.

Durante el proceso se le debe indicar la base de datos que utilizará el sitio web entre otros datos relativos al dominio. Una vez finalizado el proceso, ya se dispondrá de la copia exacta (tema y plugins incluidos) de la página que estaba en “local” en el servidor web que se desee.

La alternativa a utilizar Duplicator es la de realizar una nueva instalación de WordPress en el servidor y subir mediante FTP la carpeta que contiene el tema de la página web. Además, habría que indicarle también de nuevo la base de datos e instalar todos los plugins que tuviera la página original de manera individual.

Por último, para que esta pueda ser indexada por los buscadores, se debe acceder al *backoffice* de WordPress y dentro de Ajustes/Lectura activar la opción que permite la indexación del sitio en los motores de búsqueda (ver Figura).



Visibilidad en los motores de búsqueda

Pedir a los motores de búsqueda que no indexen este sitio
Depende de los motores de búsqueda atender esta petición o no.

Figura 42. Campo para la activación de la indexación de la página en los buscadores.



CAPÍTULO 8: CONCLUSIONES

En este capítulo final se expondrán las conclusiones tanto en materia académica como personal tras el desarrollo completo de este proyecto. Además, se relacionarán los conocimientos aplicados en este Trabajo de Fin de Grado con los obtenidos en diversas asignaturas del grado en Ingeniería Informática.

8.1 Conclusiones

En este Trabajo de Fin de grado se han cumplido los objetivos propuestos al inicio de este. Se ha aprendido el manejo de diversas tecnologías desconocidas o no conocidas en profundidad al principio de este como son PHP, SASS, HTML, JQuery, Git, etc.

También es importante destacar la disciplina adquirida durante el desarrollo del proyecto. Esta disciplina es necesaria para el buen funcionamiento del trabajo en equipo. Al igual que realizar una buena selección de las herramientas necesarias para dar soporte al progreso del proyecto de la manera más fluida posible. Destaca el acierto al escoger Visual Studio Code como entorno de programación por su flexibilidad y numerosos complementos, así como el acierto de no haber dependido de librerías especializadas en diseño web. Esto ha permitido profundizar en mayor medida en el trabajo y ampliar conocimientos a muchos niveles.

Por último, sobresale también la motivación extra que supone haber creado una red social que encaja con las necesidades de los jóvenes y que pueda, quizás en un futuro, ofrecer un servicio que favorezca la práctica deportiva en comunidad.

8.2 Relación del trabajo desarrollado con los estudios cursados

Muchas son las asignaturas que, con los conocimientos aportados, han contribuido a que tenga la capacidad necesaria como para llevar adelante este proyecto. A continuación, se destacarán las principales asignaturas del grado en Ingeniería Informática que han contribuido a ello:

- **Introducción a la Informática y a la Programación y Programación:** en ellas se aprendieron las primeras nociones básicas de programación. Si bien no se ha utilizado en ningún momento el lenguaje aprendido en ellas, estas lograban estructurar la forma de pensar del alumno de manera que le fuera más sencillo aprender cualquier otro lenguaje en el futuro.
- **Bases de Datos y Sistemas de Información y Diseño y Gestión de Bases de Datos:** en ambas se exploraba el lenguaje SQL que ha sido necesario para la resolución de algunos errores y comprensión del funcionamiento interno de WordPress. Además, transmitían la importancia de comprender la estructura de las BBDD sobre las que se trabaja y los datos que contienen.
- **Ingeniería del Software:** en esta asignatura se tuvo el primer contacto con el trabajo en equipo mediante control de versiones. Se utilizó un repositorio para el código que cumplía funciones casi idénticas al empleado en este proyecto. Por otra parte, el alumno



lograba entender las diferentes capas de un proyecto y la conexión entre ellas. Además, fue en esta asignatura y en sus prácticas donde se conoció la metodología de trabajo ágil Scrum.

- **Gestión de Proyectos:** se aprendió sobre la necesidad de la definición de tareas y tiempos y de cómo llevar el control del proceso para asegurar que todo se cumpla según los plazos. También en esta asignatura se empleó Trello, aplicación que ha sido vital para la colaboración en este proyecto.
- **Interfaces persona computador:** dado que este proyecto no contaba con diseñadores profesionales para sus interfaces, los conocimientos adquiridos en esta asignatura fueron de vital importancia. Gracias a ellos se pudieron realizar diseños intuitivos para el usuario con elementos coherentes. Asimismo, se aprendió sobre la gran importancia de validar los diseños con usuarios de distintos rasgos sociodemográficos con la finalidad de recoger sugerencias y mejorar.
- **Tecnologías de Sistemas de Información en la Red:** lenguajes como JavaScript o el entorno de node fueron conocidos en esta asignatura por primera vez. Estos conocimientos fueron una base indispensable para facilitar la comprensión de estas tecnologías a lo largo del proyecto.

8.3 Trabajos futuros

A raíz de este Trabajo de Fin de Grado surgen varias posibilidades que se podrían desarrollar en un futuro. La primera y más evidente es la migración de esta red social a un formato de aplicación móvil. En la actualidad la mayoría de las redes sociales son accedidas desde dispositivos móviles más que desde ordenadores. Si bien en este TFG se ha hecho el desarrollo para que sea *responsive* y su visualización sea correcta en teléfonos inteligentes, una aplicación podría resultar más cómoda para los usuarios.

Por otra parte, otros trabajos futuros se podrían centrar en incorporar mejoras a *Buzzer Beater* para ampliar su funcionalidad. Para ello, se podrían realizar estudios de mercado y encuestas a potenciales usuarios para conocer qué aspectos mejorarían o incorporarían a la red social y así seguir mejorando sobre lo ya existente.

8.4 Lecciones aprendidas

A nivel personal, este proyecto me ha hecho crecer en numerosas virtudes que considero necesarias de cara al mundo laboral. Una de ellas es la de tener la capacidad de elaborar un proyecto conociendo y participando en todas sus fases, desde la lluvia de ideas y planificación inicial hasta los últimos retoques de este.

Este TFG también me ha dado la oportunidad de emprender este proyecto trabajando en equipo y aplicando a una situación real metodologías ágiles que son tan utilizadas en el entorno laboral como es Scrum.

La importancia de dimensionar bien las tareas que acometes y el compromiso a terminarlas es muy importante cuando de tu trabajo depende que otros puedan avanzar en el suyo. De esta



manera, siento que he aprendido a asumir responsabilidades cuando algo se tuerce o surge un contratiempo y poner los medios necesarios para volver a enderezar el rumbo del proyecto.

Por último, he obtenido un gran crecimiento en mis virtudes autodidactas ya que he tenido que manejarme fuera de mi zona de confort continuamente a lo largo del TFG. Enfrentándome a tecnologías desconocidas a medida que me era necesario, he logrado un conocimiento no solo de las tecnologías en concreto sino también de la importancia de saber formular de manera precisa las preguntas para obtener la información que se desea.



BIBLIOGRAFÍA

Carat. (8 de febrero de 2022). *REDES SOCIALES: ángeles y demonios*. Obtenido de <https://www.carat.com/es/es/como-pensamos/redes-sociales>

Deemer, P., Benefield, G., Larman, C., & Vodde, B. (2009). *Información básica de SCRUM*. California: Scrum Training Institute.

Fernández, R. (20 de julio de 2022). *Statista*. Obtenido de <https://es.statista.com/estadisticas/474955/redes-sociales-tasa-penetracion-usuarios-prevision-espana/#:~:text=El%20uso%20de%20las%20redes,más%20alto%20registrado%20desde%202017.>

Guzmán, J. M. (16 de Diciembre de 2019). *webempresa*. Obtenido de <https://www.webempresa.com/blog/estructura-de-base-de-datos-en-wordpress.html>

Ministerio de Cultura y Deporte. (junio de 2021). *Encuesta de hábitos deportivos 2020*. Obtenido de <https://www.culturaydeporte.gob.es/dam/jcr:07b62374-bfe9-4a65-9e7e-03a09c8778c3/encuesta-de-habitos-deportivos-2020.pdf>

Statista. (2021). *Redes sociales con mayor porcentaje de usuarios en España en 2021*. Obtenido de <https://es.statista.com/estadisticas/489153/porcentaje-de-internautas-en-las-redes-sociales-en-espana/>

WordPress. (2022). Obtenido de <https://wordpress.com/es/>



ANEXOS

Anexo A. Objetivos de Desarrollo Sostenible

En 2015 los líderes de las Naciones Unidas establecieron una serie de objetivos comunes con los que fijar unas metas comunes para todos los países del mundo con la finalidad de construir una sociedad más sostenible y proteger el planeta.

Los ODS son diecisiete y el grado de relación que guardan con el presente TFG es el siguiente:

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.				X
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.				X
ODS 10. Reducción de las desigualdades.		X		
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.		X		



- **Salud y Bienestar:** *Buzzer Beater* promueve la práctica del deporte en comunidad. Establece un medio más para facilitar y motivar a los ciudadanos a la práctica del deporte lo cual contribuye a mejorar su estilo de vida hacia uno más saludable.
- **Reducción de desigualdades:** la práctica del deporte en comunidad es una manera de acercar personas a otras con un objetivo común. Gracias a *Buzzer Beater* muchos usuarios podrán conocer a otras personas de ámbitos completamente diferentes con los que compartir un actividad lúdica.
- **Alianzas para lograr objetivos:** los deportes en equipo fomentan el trabajo en conjunto con personas de todo tipo y te permite desarrollar una serie de virtudes relacionadas con el trato y trabajo con otros. *Buzzer Beater* facilita este tipo de encuentros entre sus usuarios ayudando a fomentar estas cualidades que luego pueden trasladar a otros ámbitos de sus vidas contribuyendo a mejorar la sociedad en su conjunto.