



# Data Beach Volley

Proyecto Final Ciclo DAW

Por: **Óscar Noguero**

Centro: **C.P.I.F.P. Los Enlaces**

Curso: **2024 - 2025**

Tutor:  
**Roberto Marín.**

Fecha:  
**7 de enero de 2024**

# Data Beach Volley

## Índice de contenido

<b>1. Descripción del Proyecto.....</b>	<b>3</b>
<b>1.1. Contexto del proyecto.....</b>	<b>3</b>
1.1.1. Ámbito del proyecto y entorno.....	3
1.1.2. Análisis de la realidad.....	3
1.1.3. Solución y justificación de la propuesta.....	4
1.1.4. Destinatarios.....	5
1.1.5. Marco legal.....	5
<b>1.2. Objetivo.....</b>	<b>6</b>
<b>1.3. Objetivo en inglés.....</b>	<b>6</b>
<b>2. Documento de Acuerdo del proyecto.....</b>	<b>7</b>
<b>2.1. Casos de Uso (CU).....</b>	<b>7</b>
<b>2.2. Tareas.....</b>	<b>11</b>
<b>2.3. Metodología a seguir para la realización del proyecto.....</b>	<b>12</b>
<b>2.4. Planificación temporal de tareas.....</b>	<b>13</b>
<b>2.5. Presupuesto.....</b>	<b>14</b>
2.5.1. Beneficios.....	15
<b>2.6. Contrato y pliego de condiciones.....</b>	<b>16</b>
<b>2.7. Análisis de Riesgos (DAFO).....</b>	<b>20</b>
<b>3. Análisis y Diseño.....</b>	<b>22</b>
<b>3.1. Análisis y diseño de la arquitectura de la aplicación.....</b>	<b>22</b>
3.1.1. Tecnologías/Herramientas usadas.....	22
Otros formatos de datos utilizados en el proyecto.....	24
3.1.2. Arquitectura de componentes de la aplicación.....	24
Diagrama de flujo de datos.....	25
<b>3.6. Modelado de datos. (Análisis y diseño de la BBDD).....</b>	<b>27</b>
<b>3.7. Análisis y diseño de sistema funcional (Diagrama de Casos De Uso).....</b>	<b>29</b>
<b>3.8. Análisis y diseño de la interfaz de usuario (Mockup).....</b>	<b>30</b>

## Índice de tablas

Tabla 1: Casos de uso.....	8
Tabla 2: Tareas.....	12
Tabla 3: Salarios.....	15
Tabla 4: Costes fijos y costes variables.....	15
Tabla 5: Costes totales.....	15
Tabla 6: Beneficios.....	16



## Índice de imágenes

Figura 1: Diagrama de gantt.....	14
Figura 2: Logo Jira.....	23
Figura 3: Logo Git.....	23
Figura 4: Logo Visual Studio Code.....	23
Figura 5: Logo Docker.....	23
Figura 6: Logo MySQL.....	24
Figura 7: Logo Laravel.....	24
Figura 8: Logo Balsamiq.....	24
Figura 9: Logo Draw.io.....	24
Figura 10: Logo Django.....	24
Figura 11: Logo Reactjs.....	24
Figura 12: Diagrama de flujo lógico de la arquitectura de la aplicación.....	26
Figura 13: Diagrama de flujo físico de la arquitectura de la aplicación.....	27
Figura 14: Modelo relacional de la base de datos.....	28
Figura 15: Modelo entidad relación de la base de datos.....	29
Figura 16: Mockup 1.....	31
Figura 17: Mockup 2.....	32
Figura 18: Mockup 3.....	33
Figura 19: Mockup 4.....	34
Figura 20: Mockup 5.....	35
Figura 21: Mockup 6.....	36

## 1. Descripción del Proyecto

---

### 1.1. Contexto del proyecto

#### 1.1.1. Ámbito del proyecto y entorno

La motivación para la ejecución de este proyecto es derivada de una necesidad. Como entrenador y jugador semi-profesional de vóley playa me he encontrado con la falta de una aplicación específica para la gestión de equipos de voleibol, y en concreto, voleibol de playa. Además, tampoco contamos con aplicaciones para generar estadísticas que sea en cierto modo asequible.

El vóley playa es un deporte olímpico como su hermano el voleibol.

Sin embargo, mientras que el voleibol se desarrolló en España en los años 20, no es hasta la década de los 70 que llega el vóley playa a las costas de Cataluña.

Esto hace que su desarrollo se vea ralentizado por el voleibol y por su tardía incorporación.

Surge entonces la necesidad de crear proyectos de estadísticas para el voleibol como *Data Volley 4* que facilita mucho la labor del entrenador.

Sin embargo nadie ha dado el paso en unir estas aplicaciones estadísticas con un modelo de gestión de equipos, ni se ha creado una aplicación similar para el vóley playa que permita tomar estadísticas.

#### 1.1.2. Análisis de la realidad

El sistema propuesto anteriormente quiere solucionar la necesidad que ha surgido a través de la evolución del vóley playa.

La realidad es que no hay aplicaciones web o móviles que compartan el mismo propósito que Data Beach Volley.

Pongo un ejemplo de 3 aplicaciones semejantes:

### DATA VOLEY 4

Este software para pc no se puede tener ni como aplicación móvil ni como aplicación web. Su uso está limitado en su versión gratuita y solo se puede descargar poniéndose en contacto con la federación española de voleibol. Esta versión “Gratuita” limita la búsqueda de filtrado de

---



información, además de que solo sirve para hacer estadísticas de voleibol. Por lo tanto, no tiene en cuenta las estadísticas de voleibol de playa ni sirve para la gestión de equipos deportivos.

## **VOLLEYMETRICS**

Esta aplicación web cumple con alguna finalidad de mi aplicación propuesta.

Volleymetrics se basa en una enorme base de datos web donde se puede filtrar por jugador o por equipo y se sincronizan los videos con los gestos técnicos o acciones filtradas.

Los contras que tiene esta aplicación es que no tienes la opción de hacerte tus propias estadísticas y si entras en su base de datos automáticamente los demás pueden anaizarte.

Además, su precio es tan desorbitado que solo se lo pueden permitir algunas federaciones.

Para finalizar, tampoco permite la gestión de equipos.

## **SportEasy**

Esta aplicación web es de las tres propuestas la que mas se acerca a los propósitos de DataBeachVolley.

SportEasy funciona sobre todo como una aplicación de gestión de equipos deportivos, permitiéndonos crear un grupo de trabajo, un calendario y anotar los resultados de los campeonatos.

De estos mismos resultados, SportEasy nos genera automáticamente estadísticas muy sencillas, aptas para cualquier deporte, pero que no profundizan en lo importante.

Además, la mayoría de opciones son premium y no permite subir estadísticas más complejas.

### **1.1.3. Solución y justificación de la propuesta**

La solución como en muchos ámbitos se encuentra en el equilibrio.

Uniendo lo mejor de estas aplicaciones podemos crear una aplicación mejorada y completa.

Las claves serían desarrollar una aplicación que:

- sirva para gestionar equipos de voleibol y voleibol de playa
  - genere gráficos mas complejos y específicos
  - permita introducir los datos al estadístico y no dependa de una base de datos global.
  - sea exclusiva de voleibol y vóley playa.
  - almacene los datos y permita el filtrado por jugador y equipo.
  - sea asequible para la mayoría de clubes.
-

- permita tener más herramientas, no solo al alcance del estadístico sino del entrenador y sus jugadores.

#### 1.1.4. Destinatarios

El público objetivo serían los clubes de desarrollo medio en adelante. La idea es que sea una aplicación medianamente asequible para todos los clubes que estén buscando una mejora en la gestión de sus equipos.

Por eso los destinatarios son los clubes de mediano tamaño que podrán comprarla y distribuirla a sus entrenadores, estadísticos y jugadores.

Entendemos también que no debería ser de uso gratuito (al menos la parte de la estadística y toma de datos) y que los clubes que no se la puedan permitir por el hecho de estar empezando, tampoco la necesitarían ya que este aspecto solo es útil para equipos de alto nivel.

#### 1.1.5. Marco legal

Dado que esta aplicación recolecta y gestiona datos sensibles de los usuarios (jugadores, entrenadores, clubes), es fundamental garantizar el cumplimiento de varias leyes y normativas nacionales e internacionales, para proteger los derechos de los usuarios y evitar posibles infracciones.

Fundamentalmente nos centraremos en cumplir con la **Protección de Datos Personales**:

- **Reglamento General de Protección de Datos (GDPR):** Este reglamento es obligatorio para cualquier aplicación que trate con datos de ciudadanos de la Unión Europea. Entre sus requisitos clave se encuentran:

- **Obtención de consentimiento informado:** Los usuarios deben ser claramente informados sobre el uso que se dará a sus datos y deben proporcionar su consentimiento explícito.

Para ello, en el formulario de registro se creará un apartado para explicarle al usuario cómo se van a manejar los datos y que pueda dar consentimiento al uso de sus datos.

- **Derecho de acceso, rectificación y supresión:** Los usuarios deben tener derecho a acceder a sus datos, corregir cualquier información inexacta y solicitar la eliminación de sus datos.

Para ello, en la sección de perfil y ajustes, se podrá modificar los datos que se consideren oportunos así como eliminar la cuenta (y con esto todos los datos de la base de datos) si fuera necesario.

- **Cifrado y seguridad:** Los datos personales deben protegerse mediante tecnologías de cifrado tanto en reposo como en tránsito para evitar accesos no autorizados.
-

Para ello usaremos métodos de cifrado a la hora de introducir los datos a la base de datos, para así almacenar la información de forma protegida.

- **Ley Orgánica de Protección de Datos Personales y Garantía de los Derechos Digitales (LOPDGDD - España):** Esta ley complementa el GDPR y aplica a nivel local en España. Es esencial garantizar que la recopilación y tratamiento de datos sensibles en la aplicación cumpla con las disposiciones de esta ley.

### 1.2. **Objetivo**

El objetivo principal es la realización de una aplicación web para la gestión de equipos o grupos de trabajo de voleibol y voleibol de playa. Además, como objetivo secundario, se podrá realizar la estadística de un partido de voleibol de playa en diferido.

Este sistema ayudará con la planificación y comunicación de tus equipos de voley playa, facilitando también la toma de datos de los partidos así como su almacenamiento y filtrado de la información, dando una ventaja estratégica a quienes utilizan Data Beach Volley.

Este ofrecería tanto a los entrenadores como a los jugadores una plataforma para poder consultar su calendario de entrenamientos, mensajes personalizados con material personalizado y la información de los rivales o su propio feedback, así como un sistema de gráficas de estadísticas personalizadas.

### 1.3. **Objetivo en inglés**

The main objective is the realization of a web application for the management of volleyball and beach volleyball teams or workgroups. In addition, as a secondary objective, it will be possible to perform the statistics of a deferred beach volleyball match.

This system will help with the planning and communication of your beach volleyball teams, facilitating also the data collection of the matches as well as the storage and filtering of the information, giving a strategic advantage to those who use Data Beach Volley.

It would offer both coaches and players a platform to consult their training schedule, personalized messages with customized material and information from opponents or their own feedback, as well as a personalized statistics graphing system.

---



## 2. Documento de Acuerdo del proyecto.

### 2.1. Casos de Uso (CU)

Tabla 1: Casos de uso

ID	Casos de Uso	Criterios de aceptación	Prioridad
CU-1	Como entrenador, quiero poder mandar sesiones de planificación a mis jugadores para tener organizada la planificación de los mesociclos o de la temporada.	Habrà un calendario en el grupo de entrenamiento donde se podrá clicar en un día para introducir los datos de cada sesión concreta, acompañada de dos estados (realizada/por realizar)	Alta
CU-2	Como entrenador, quiero poder enviar estadísticas filtradas y específicas, además de material de trabajo como videos o ejemplos a mis jugadores para que sean conscientes de su progreso y sus puntos flojos, así como también las estadísticas simplificadas de los rivales.	En el foro del grupo habrá un apartado para añadir material por parte del entrenador. Este apartado tendrá una parte para añadir archivos, un apartado para comentarios y un apartado para insertar estadísticas filtradas que se efectúan desde allí.	Alta
CU-3	Como entrenador, quiero poder crear grupos de entrenamiento en los que incluir al estadístico y a los jugadores.	Debe haber un botón de crear nuevos grupos. En un grupo deberé poder añadir nuevos usuarios invitándoles en un apartado donde se requerirá de introducir el nombre de usuario de los usuarios nuevos, así como mas datos personales, y opciones de añadir categoría del equipo o grupo de trabajo, foto o	Alta





		descripción.	
CU-4	Como estadístico, quiero poder registrar los datos estadísticos de un partido de voley playa, con nomenclatura específica para la estadística de voley.	Tendrá una interfaz diferente para ir añadiendo las estadísticas con la visualización de los diferentes equipos, podrá añadir puntos a través de código complejo, apuntar el marcador, finalizar los sets, etc.	Media
CU-5	Como estadístico, quiero poder filtrar y analizar los datos estadísticos de un partido para poder identificar patrones y tendencias, para poder ofrecer información relevante al entrenador y jugadores.	Podrá acceder a la base de datos de los partidos donde se almacenan por jugador y encuentro y con un botón de filtro seleccionará las estadísticas que les resulten más interesantes de un encuentro	Alta
CU-6	Como estadístico, quiero poder exportar los datos estadísticos a diferentes formatos (por ejemplo, CSV o Excel) para poder compartirlos con otros miembros del equipo o para su análisis posterior.	Habrà un botón de exportar estadística para exportarlo en distintos formatos.	Media
CU-7	Como jugador, quiero comentar en el foro del grupo, para exponer mis dudas sobre algunas estadísticas o de los multimedia que añada el entrenador.	El jugador podrá tener acceso al foro, pero solo podrá escribir comentarios, no podrá añadir información.	Alta
CU-8	Como jugador, quiero poder marcar mi	El jugador podrá modificar el estado de	



	progreso como realizado para que mi entrenador sea consciente de cómo trabajo y me corrija si estoy haciendo algo mal.	las sesiones de entrenamiento una vez vaya completando los mesociclos de la temporada.	Media
CU-9	Como jugador, quiero poder acceder a mi planificación personal del mesociclo para poder establecer objetivos de mejora.	El jugador tendrá acceso al calendario de la planificación. Si hace clic en una sesión dentro de un día se le desglosa la sesión entera.	Alta
CU-10	Como usuario de la aplicación quiero poder logearme con mi usuario y acceder a mis grupos de trabajo.	Se rellenará un formulario para darse de alta como usuario. Será necesario introducir nombre de usuario, contraseña y tipo de usuario. Un usuario de tipo jugador es gratuito mientras que el estadístico y el entrenador son usuarios de pago.	Alta
CU-11	Como usuario de la aplicación quiero tener un perfil al que poder acceder para editar, añadir o eliminar información personal.	Habrà un apartado de perfil en el menú, que permita hacer la típica gestión de cuenta de usuario. Se podrá añadir una foto, y completar los datos no obligatorios que no se completaron en el registro, así como habrá un apartado de seguridad para eliminar la cuenta si se desea.	Alta
CU-12	Como entrenador, quiero poder ver gráficas relacionadas con las estadísticas subidas por el estadístico, así como decidir que graficas mostrar a mis jugadores.	El entrenador podrá filtrar las estadísticas subidas por el estadístico para generar distintas gráficas, como el porcentajes de victorias, ataques convertidos, defensas, bloqueos, etc.	Media
CU-13	Como usuario, deberé tener un menú para	A la derecha de la pantalla, se situará un	



	cambiar a todas las diferentes pestañas de la aplicación.	menú lateral para cambiar entre el foro, los mensajes, las gráficas estadísticas, y funciones específicas de cada tipo de usuario. Contará también con un menú de navegación superior con las opciones básicas (gestión de perfil, login /logout y registro)	Alta
CU-14	Como usuario general, quiero entrar a la página y que me aparezca una página explicativa de la aplicación antes de registrarme.	Habrà un index sencillo sin menú lateral para cuando se navega en la página sin registrarse, con información acerca de la aplicación.	Media
CU-15	Como usuario y estadístico, me gustaría que hubiera un tutorial para la toma de datos y otro para la navegación en la página.	Mediante pop-ups, se guiará a los usuarios en la página para que la navegación en ella sea mas sencilla.	Muy baja



## 2.2. Tareas

Tabla 2: Tareas

TAREA	FCH INICIO	FCH FIN
<b>T.0 Propuesta de proyecto</b>	04/10/2024	06/10/2024
<b>T0.1 Diseño de Estructura de BBDD</b>	07/10/2024	08/10/2024
<b>T1. Descripción de Proyecto</b>	09/10/2024	12/10/2024
<b>T.2 Aprender introducción Django</b>	12/10/2024	15/10/2024
<b>T2.1 Documento Acuerdo de Proyecto</b>	16/10/2024	18/10/2024
<b>T3. Documento Análisis y Diseño del Primer Sprint</b>	19/10/2024	22/10/2024
<b>T3.2 Diseño Interfaz Mockup</b>	23/10/2024	25/10/2024
<b>T4. Primer Sprint</b>	25/10/2024	12/11/2024
<b>T4.1. Arquitectura y estructura básica de la aplicación</b>	25/10/2024	2/11/2024
<b>T4.2. Sección Usuarios (Login, Registro, Editar)</b>	2/11/2024	12/11/2024
<b>T5. Documento Análisis y Diseño del Segundo Sprint</b>	13/11/2024	15/11/2024
<b>T6. Segundo Sprint</b>	16/11/2024	08/12/2024
<b>T6.1. Sección Estadístico e implementación Calendario</b>	16/11/2024	30/11/2024
<b>T6.2. Implementación del Foro</b>	30/11/2024	08/11/2024
<b>T7. Tercer Sprint (parte estadística)</b>	08/12/2024	25/12/2024
<b>T7. 1.Implementación parte estadística</b>	09/12/2024	16/12/2024
<b>T7.2 Implementación gráficas</b>	17/12/2024	25/12/2024
<b>T8. Documentos finales</b>	26/12/2024	31/12/2024



### **2.3. Metodología a seguir para la realización del proyecto.**

Para este proyecto, se ha elegido la metodología ágil Kanban, y se trabajará con la aplicación Atlassian y Jira en el siguiente repositorio: <https://data-beach-volley.atlassian.net/>.

Esta metodología consiste en dividir las distintas tareas en bloques diferentes, de forma que el trabajo se ve limitado. Así, solo se comienza un nuevo bloque de trabajo cuando el bloque anterior se ha terminado.

De esta manera, nos aseguramos de que los bloques se completen de forma más compacta y logramos ajustarnos mejor a las fechas de entrega.

La idea es dividir el proyecto en 3 sprints diferentes:

#### **25/10/2024 – 12/11/2024**

El primer Sprint, en resumen, constará de Integrar y Montar la Arquitectura básica de la aplicación, de Integrar los FrameWorks, y de implementar el sistema de registros de usuarios.

#### **16/11/2024 - 08/12/2024**

El segundo Sprint se centrará en la implementación del apartado de los mensajes o Foro, así como de desarrollar las opciones específicas del usuario estadístico y la implementación del calendario.

#### **08/11/2024 - 25/12/2024**

Por último, el tercer Sprint se centrará en desarrollar la toma de estadísticas, el filtrado de éstas y la generación de diagramas y gráficas.

---

*Figura 1: Diagrama de gantt*



## 2.5. Presupuesto.

Para hacernos una idea de el valor de la aplicación propuesta, vamos a calcular primero los posibles gastos que vamos a tener:

En esta tabla podemos observar las horas empleadas en la elaboración del proyecto, agrupadas en función de las distintas tareas.

Tabla 3: Salarios

Apartado	Precio/Hora	Horas	Importe
Análisis	9,00€	45	405,00€
Diseño	10,00€	100	1000,00€
Programación	15,00€	180	2700,00€
Pruebas	10,00€	60	600,00€
Documentación	9,00€	80	720,00€
<b>TOTAL</b>			<b>4885,00€</b>

El total de gastos se distribuye entre los costes fijos, que serán cargados en su totalidad al proyecto (luz, teléfono, Internet, horas, etc.), y costes variables (herramientas, software, etc.).

Tabla 4: Costes fijos y costes variables

Costes Variables			Costes Fijos	
Concepto	Cantidad	Precio	Concepto	Precio
Ordenador	1	150,00€	Luz y teléfono	70,00€
Pantalla y Periféricos	1	50,00€	Internet	80,00€
<b>TOTAL</b>		<b>200€</b>	<b>TOTAL</b>	<b>150,00€</b>

Tabla 5: Costes totales

Costes Totales	
<b>Costes Variables</b>	200,00€
<b>Costes Fijos</b>	150,00€
<b>Salarios</b>	4885,00€
<b>TOTAL</b>	<b>5235,00€</b>



### 2.5.1. Beneficios

El precio técnico (Pt), que en este caso coincide con el coste total, por encontrarnos ante un único proyecto, asciende a 5.235,00 € (suma entre costes fijos y variables).

Para realizar el cálculo hemos aplicado la siguiente fórmula:

$$Pt = CF + CVu$$

siendo “CF” los costes fijos, y “CVu” los costes variables por unidad.

Sabiendo todo ésto, queremos sacar un beneficio del 30%. Vamos a calcular el Precio de Venta (P) a través de la siguiente fórmula:

$$B = P - (CVu + CF)$$

B (beneficio) será el 30% de 5235,00€ = 1570€ aproximadamente.

Ya podemos calcular el Precio de venta sustituyendo todos los datos en la fórmula. Obtenemos que el Precio de Venta será el resultado de la operación  $1570 = P - (200 + 5035)$ , es decir 6805,00 euros.

Tabla 6: Beneficios

RESUMEN - BENEFICIOS	
Costes Totales	5235,00€
Precio de venta	6805,00€
Porcentaje (%)	30%
Beneficios totales	1570,00€



**2.6. Contrato y pliego de condiciones.****CONTRATO DE DESARROLLO DE SOFTWARE****FECHA:** 2/11/2024**DE UNA PARTE,**

DataBeachVolley, con domicilio en Zaragoza, España y número de identificación/registro fiscal .....XXXXX....., representada por  
Óscar Noguero Casorrán, Desarrollador .

**Y DE OTRA,**

...XXXXXXX...., con domicilio social en Zaragoza, España y número de identificación/registro fiscal, XXXXXXXXXX , representada por XXXXXXXXX.

Ambas partes podrán denominarse conjuntamente como “las Partes” o individualmente como “la Parte”. Las Partes, de sus libres y espontáneas voluntades, manifiestan tener y se reconocen, mutua y recíprocamente, la capacidad legal necesaria para otorgar el presente Contrato, a cuyos efectos

**MANIFIESTAN**

- Que el Desarrollador, es una empresa que desarrolla software y programas informáticos a medida.
  - Que el Cliente desea disponer de un software o programa informático para gestionar o realizar parte de las tareas de su empresa o negocio.
  - Que, en virtud de las consideraciones precedentes, las Partes, de sus libres y espontáneas voluntades, han acordado otorgar el presente Contrato de desarrollo de Software, (en adelante, “el Contrato”) con sujeción a las siguientes
-

## CLÁUSULAS

### 1. OBJETO

Por medio del presente Contrato, el Cliente encarga al Desarrollador, que acepta, el desarrollo de una aplicación informática para la realización de la aplicación DataBeachVolley, según las características y especificaciones establecidas en el Anexo I al presente Contrato, (en adelante, “el Software o el Programa Informático”).

### 2. DEFINICIONES

“**Actualizaciones**” significa la nueva versión del Programa Informático que corrija errores en el Software o que, sin ningún tipo de obligación bajo el presente, mejora la funcionalidad del Software proporcionando funciones adicionales o cualquier incremento en su rendimiento.

“**Especificaciones**” o “**Funcionalidades**” significa la funcionalidad del Programa Informático como se describe en las características del mismo y las funciones de ayuda, si las hubiese, proporcionadas en soporte electrónico o en papel por el Desarrollador juntamente con el Programa informático.

“**Errores**” Significa fallos en el Programa que impiden su utilización o que no permiten realizar todas las funcionalidades especificadas.

“**Entrega del Programa informático**” momento en el que el Cliente da su conformidad con el Programa informático mediante documento por escrito o transcurridos 15 días desde la finalización del plazo de revisión sin haber notificado ninguna corrección o modificación al Desarrollador.

### 3. CONTRAPRESTACIÓN

En contraprestación por el desarrollo y cesión de la titularidad del Programa informático y su puesta en funcionamiento, el Cliente se obliga a abonar al

---



Desarrollador el importe total de 6805,00 euros, dicho importe le serán adicionados los impuestos vigentes (en adelante, “el Precio”).

El importe total establecido como contraprestación es total y único, por lo que se entenderá incluido en el mismo cualesquiera pagos debiera efectuar el Desarrollador a terceros, por licencias o derechos de propiedad intelectual que pudieran incluirse en las extensiones para el desarrollo del Programa informático.

#### **4. FORMA DE PAGO**

*Alternativa A.*

El pago del Precio se realizará en los siguientes plazos:

1. En el momento de la firma del presente Contrato el Cliente paga al Desarrollador la cantidad de 3402,50 euros. El presente documento es la más eficaz carta de pago de dicha cantidad.
2. La cantidad restante del Precio se abonará por el Cliente en el momento de la entrega del Programa informático.

El pago del Precio se realizará mediante transferencia bancaria a la cuenta número XXXXXXXXXXX del banco XXXXX del Desarrollador.

#### **5. PLAZO DE DESARROLLO DEL SOFTWARE**

*Alternativa A.*

El Desarrollador deberá entregar el software completamente terminado antes del 31 de diciembre de 2024.

#### **6. PROPIEDAD INTELECTUAL**

---

La titularidad de los derechos de propiedad, propiedad intelectual sobre el Programa informático objeto del presente Contrato pertenecen al Cliente. El Desarrollador manifiesta que tiene derecho a ceder dicha propiedad al Cliente. La cesión no es en exclusiva, el Desarrollador podrá utilizar el código fuente del Programa informático para realizar otras aplicaciones.

## 7. GARANTÍA

El Desarrollador garantiza el buen funcionamiento del Programa informático. No obstante, en caso de que el mismo presente algún error o defecto, el Desarrollador se compromete a poner sus mejores esfuerzos en la subsanación de los mismos. El Desarrollador no será responsable de los daños directos e indirectos que pudieran derivarse de la utilización del Programa informático, ni tendrá responsabilidad alguna por la pérdida de datos o información en relación con el uso del mismo, cuando no se utilice según sus instrucciones o no se sigan las consideraciones establecidas en la presente cláusula.

En relación a la garantía, el Desarrollador establece las siguientes limitaciones a la garantía de buen funcionamiento que otorga por el periodo de (*indicar*) meses/años:

1. El Programa informático se ha creado para realizar las funcionalidades descritas en el Contrato, pero no para otras.
2. El Programa informático funcionará correctamente en los equipos detallados en el Anexo II al presente Contrato.
3. EL Programa informático funcionará correctamente si se utiliza con los programas informáticos y versiones detalladas en el Anexo II. No se garantiza el funcionamiento del Programa informático con otras versiones de dichos programas o actualizaciones.
4. La garantía no operará para el caso que el Cliente o un tercero bajo sus indicaciones, modifique instale o desinstale el Programa informático.
5. La interoperatividad con otros programas tampoco se garantiza. Únicamente interoperará con los programas detallados en el Anexo II al presente Contrato.
6. Capacidad o número de usuarios. El Programa informático funcionará correctamente con el volumen de operaciones/datos detallados en el Anexo I. El Programa informático podrá ser utilizado simultáneamente por el número de usuarios detallados en el Anexo I.

## 2.7. Análisis de Riesgos (DAFO).

Para analizar bien los riesgos, presentamos el siguiente análisis DAFO de la aplicación:

### Fortalezas

- **Enfoque específico:** Está diseñada principalmente para el voleibol y el vóley playa, lo cual la hace altamente relevante y adaptada para este deporte específico, a diferencia de otras aplicaciones de análisis que cubren diferentes disciplinas.
- **Almacenamiento y filtrado de datos:** La aplicación permite almacenar y filtrar estadísticas por jugador y equipo, ofreciendo una herramienta útil para el análisis profundo de rendimientos individuales y colectivos.
- **Accesibilidad y coste moderado:** A diferencia de herramientas más costosas, esta aplicación busca ser asequible para clubes medianos, lo que facilita su adopción en un segmento amplio del mercado.

### Debilidades

- **Dependencia de infraestructura:** Para manejar adecuadamente el tráfico y las demandas de procesamiento en tiempo real, la aplicación podría enfrentar problemas de rendimiento, especialmente si se da un crecimiento repentino de usuarios.
- **Posibles problemas de integración y seguridad:** Debido a la falta de experiencia, existen algunos riesgos de integración entre componentes y amenazas de seguridad que podrían comprometer la privacidad de la aplicación.
- **Falta de conocimientos técnicos específicos:** Al utilizar herramientas nuevas como Django, programar con Python, o el uso de Atlassian (Jira), se puede llegar a dar una falta de experiencia con estas herramientas, lo que podría retrasar el desarrollo o afectar la calidad final.

### Oportunidades

- **Mercado desatendido:** No existen aplicaciones específicas para vóley playa en el mercado, lo que representa una ventaja competitiva para captar usuarios en este nicho.
- **Creciente popularidad del vóley playa:** A medida que el vóley playa gana popularidad, clubes y federaciones pueden mostrar más interés en el análisis de datos y la tecnología para mejorar el rendimiento de los jugadores.
- **Posibilidad de expansión:** La aplicación podría evolucionar para incluir funciones adicionales, como análisis predictivo de jugadas o herramientas de inteligencia artificial para ofrecer funcionalidades más avanzadas.

### Amenazas

---



- **Competencia potencial:** Empresas de análisis deportivo mas reconocidas podrían llevarse los usuarios de la aplicación si no tenemos claro quienes son nuestros clientes potenciales y buscan funcionalidades más genéricas.
- **Escalabilidad limitada:** Si la aplicación no está diseñada para manejar grandes volúmenes de datos y usuarios simultáneos, puede enfrentar problemas de rendimiento durante eventos importantes.
- **Riesgos de seguridad:** Dado que manejará datos personales y estadísticas, una brecha de seguridad podría dañar la reputación de la aplicación y afectar la confianza de los usuarios.

## Resumen del Análisis

La aplicación se distingue entre la competencia por su enfoque exclusivo en el voleibol y vóley playa, por tener funcionalidades estadísticas avanzadas como el almacenamiento y filtrado de datos, y un coste accesible, pero enfrenta retos técnicos y de seguridad debido a la falta de experiencia y escalabilidad. Con un mercado desatendido y la creciente popularidad de estos deportes, tiene oportunidades de expansión, aunque debe anticiparse a la competencia y a posibles problemas de rendimiento y confianza por riesgos de seguridad.

### 3. Análisis y Diseño.

---

#### 3.1. Análisis y diseño de la arquitectura de la aplicación.

##### 3.1.1. Tecnologías/Herramientas usadas

Para las aplicaciones que se utilizan de manera online, en el apartado versión utilizaré el término: “Cloud”.

##### **Jira (Atlassian)**

*Versión:* Cloud

*Descripción:* Plataforma de gestión de proyectos y seguimiento de tareas, ampliamente utilizada para metodologías ágiles. En este proyecto, Jira se emplea para planificar y gestionar el desarrollo, organizando las tareas en sprints y asignando responsables para cada una. Permite un seguimiento preciso del progreso y estado de cada fase del proyecto.



Figura 2: Logo Jira

##### **Git**

*Versión:* 2.42.0 (windows)

*Descripción:* Sistema de control de versiones que permite gestionar y rastrear cambios en el código fuente de forma eficiente. El repositorio principal se aloja en una plataforma como GitHub.



Figura 3: Logo Git

##### **Visual Studio Code**

*Versión:* 1.95.0

*Descripción:* Editor de código principal de la aplicación, facilita la escritura, depuración y pruebas de código en diferentes lenguajes como Python y JavaScript. Este editor se utiliza para el desarrollo tanto del frontend como del backend, para trabajar con entornos de desarrollo como Django y React.

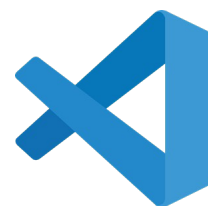


Figura 4: Logo  
Visual Studio  
Code

##### **Docker**

*Versión:* 24.0.7

*Descripción:* Plataforma de virtualización que permite empaquetar aplicaciones en contenedores. Docker se emplea para crear entornos consistentes y replicables en cada etapa del desarrollo y despliegue. Es la herramienta con la cual vamos a desplegar la aplicación.



Figura 5: Logo Docker

**MySQL**

*Versión:* 8.0.30

*Descripción:* Sistema de gestión de bases de datos relacional. MySQL se utiliza para almacenar y gestionar los datos de la aplicación, asegurando integridad y eficiencia en las operaciones de consulta. Utilizaremos otra herramienta llamada Laragon para gestionar de mejor manera el acceso a las bases de datos que creemos en MySQL.



*Figura 6: Logo  
MySQL*

**Balsamiq Wireframes**

*Versión:* Cloud

*Descripción:* Herramienta de creación de mockups que permite diseñar la interfaz de usuario y conceptualizar la navegación de la aplicación de forma rápida y visual. Balsamiq será la aplicación por excelencia que usaremos a la hora de crear los mockups.



*Figura 8:  
Logo  
Balsamiq*



*Figura 7:  
Logo  
Laravel*

**Draw.io**

*Versión:* Cloud

*Descripción:* Aplicación para la creación de diagramas. Utilizada para diseñar diagramas de flujo de trabajo, arquitectura de la aplicación y diagramas de entidad-relación para la base de datos. Draw.io será la herramienta perfecta para crear los diagramas en la fase de diseño.



*Figura 9:  
Logo  
Draw.io*

**Django**

*Versión:* 4.2.16

*Descripción:* Framework de desarrollo web en Python, utilizado para el desarrollo del backend de la aplicación. Django ofrece una estructura robusta para la creación de aplicaciones seguras y escalables, permitiendo la gestión de usuarios, autenticación, y operaciones CRUD (Create, Read, Update, Delete) con la base de datos.

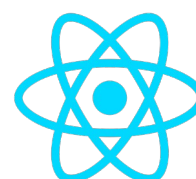


*Figura 10:  
Logo  
Django*

**React.js**

*Versión:* 18.3.1

*Descripción:* Biblioteca de JavaScript para la creación de interfaces de usuario. En este proyecto, React se utiliza para desarrollar el frontend de la aplicación, permitiendo la creación de componentes dinámicos y reutilizables.



*Figura 11:  
Logo Reactjs*



Otros formatos de datos utilizados en el proyecto

### 1. Exportación de datos en JSON

Durante la comunicación entre el frontend y el backend, se utilizará seguramente JSON como formato para el intercambio de datos. Además, será una opción para exportar estadísticas.

### 2. Exportación de estadísticas en PDF, XML, etc

Se podrá extraer estadísticas en distintos formatos para una mejor cobertura.

#### 3.1.2. Arquitectura de componentes de la aplicación

##### 3.1.2.1. Arquitectura lógica

Para la arquitectura lógica de nuestra aplicación web, utilizaremos el patrón de arquitectura MVC(Modelo - Vista – Controlador).

La arquitectura constará de las siguientes partes:

- **Modelo:** Contiene la lógica de negocio y los datos con los que opera la aplicación. Este componente se encarga de comunicarse con la base de datos y gestionar la información que posteriormente será enviada a las vistas. Es responsable de realizar operaciones CRUD (Crear, Leer, Actualizar, Borrar) en los datos.
- **Controlador:** Es el intermediario entre el usuario (a través de la vista) y el modelo. Procesa las solicitudes de los usuarios, solicita información al modelo y decide qué vista presentar de vuelta al cliente. Además, se encarga de aplicar la lógica necesaria para modificar el modelo cuando sea pertinente.
- **Vista:** Representa la interfaz de usuario. Se encarga de mostrar los datos que recibe del controlador y de notificar cualquier acción del usuario al controlador.

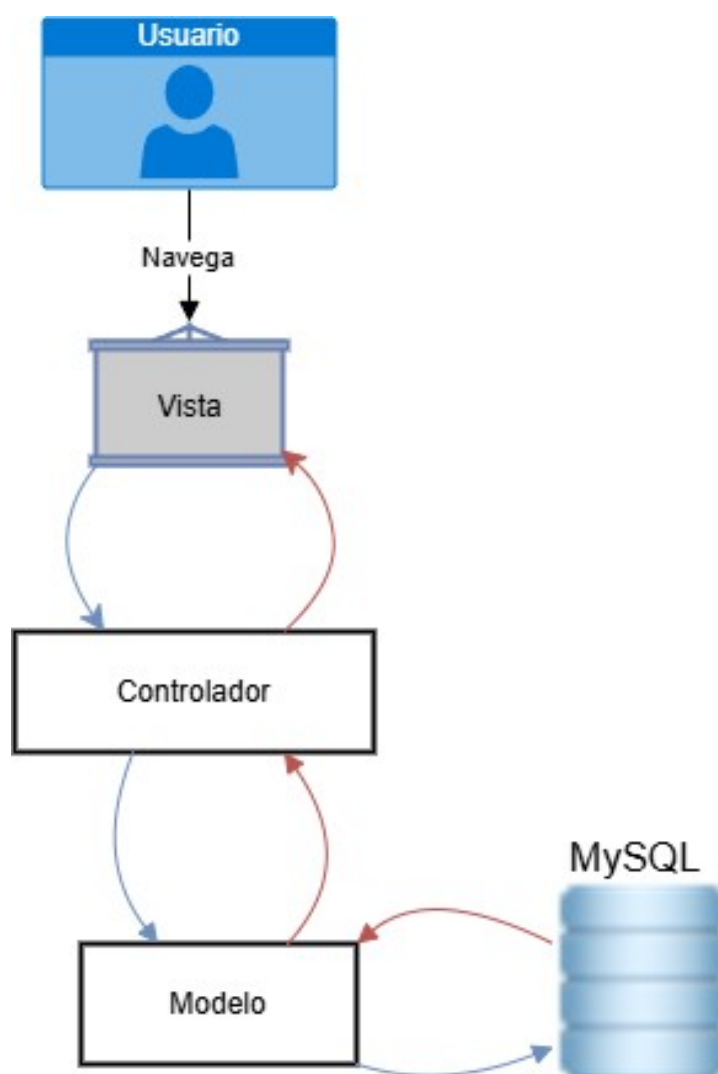
El flujo de interacción entre los componentes es el siguiente:

1. El usuario realiza una solicitud a través de la interfaz web (vista).
  2. La solicitud se envía al controlador, que la procesa.
  3. El controlador consulta los datos necesarios al modelo.
  4. El modelo se comunica con la base de datos para obtener o modificar la información requerida.
  5. El modelo envía la respuesta al controlador.
  6. El controlador selecciona y retorna una vista al usuario, mostrando los datos obtenidos.
-



### Diagrama de flujo de datos

El siguiente diagrama representa el flujo de datos en la aplicación web:



*Figura 12: Diagrama de flujo lógico de la arquitectura de la aplicación*

### 3.1.2.2. Arquitectura física

En cuanto a la arquitectura física, los usuarios acceden a la aplicación a través de un navegador web en sus dispositivos, utilizando conexiones a Internet (Wi-Fi o datos móviles). El servidor web actúa como intermediario entre el cliente y la base de datos, gestionando las solicitudes del usuario y garantizando una comunicación segura. Los principales componentes son:

- **Navegador web:** Permite al usuario interactuar con la aplicación.
- **Servidor web:** Aloja la lógica del controlador y sirve los archivos necesarios para renderizar la interfaz.
- **Base de datos:** Almacena toda la información de la aplicación y se comunica con el modelo para realizar operaciones de lectura o escritura.

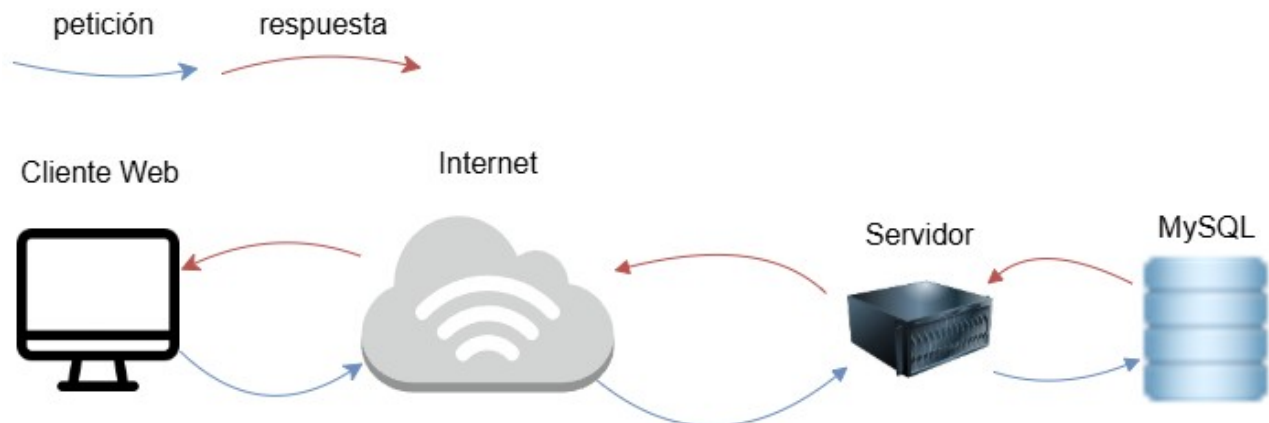


Figura 13: Diagrama de flujo físico de la arquitectura de la aplicación

## 3.5. Modelado de datos. (Análisis y diseño de la BBDD)

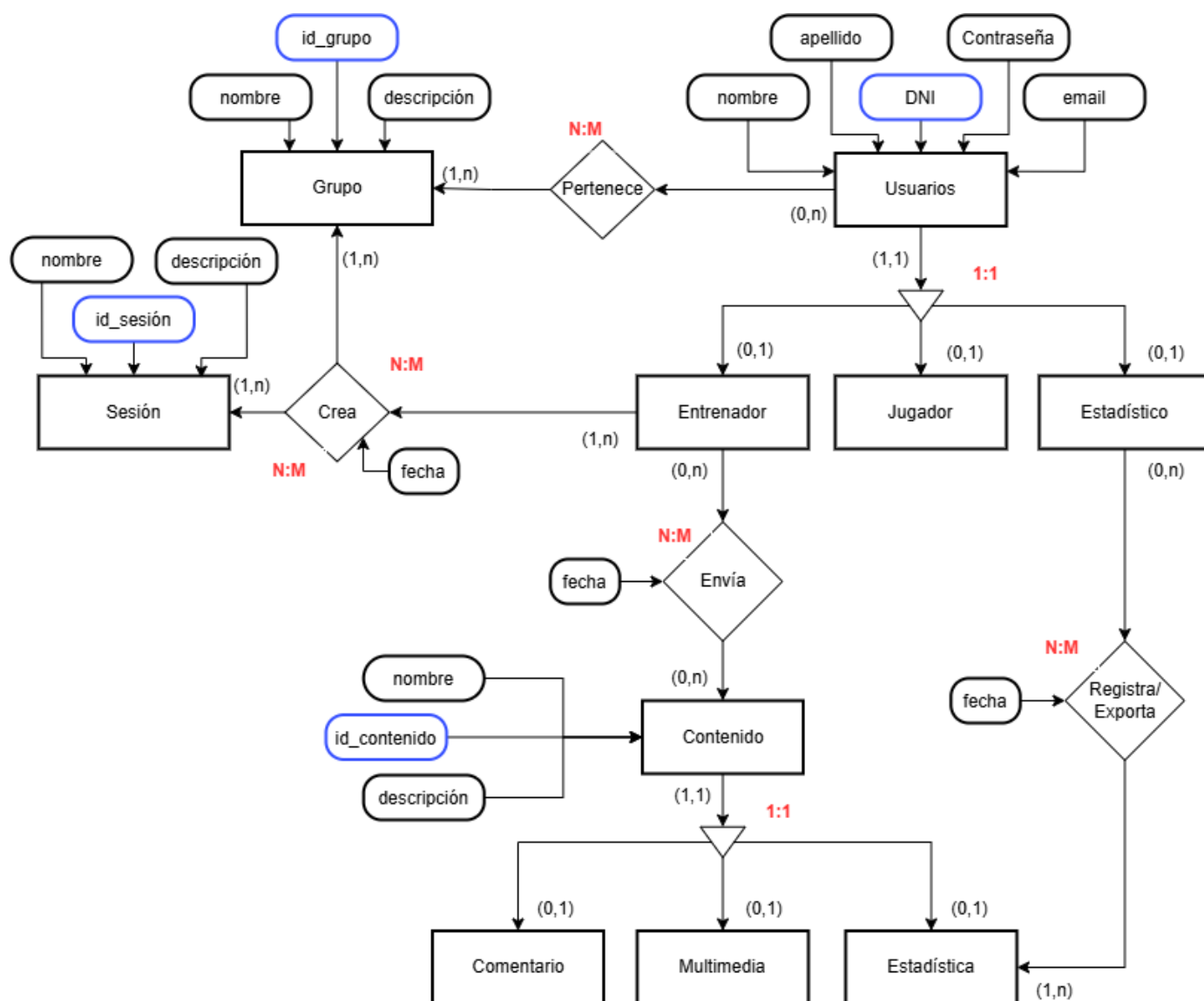


Figura 14: Modelo relacional de la base de datos

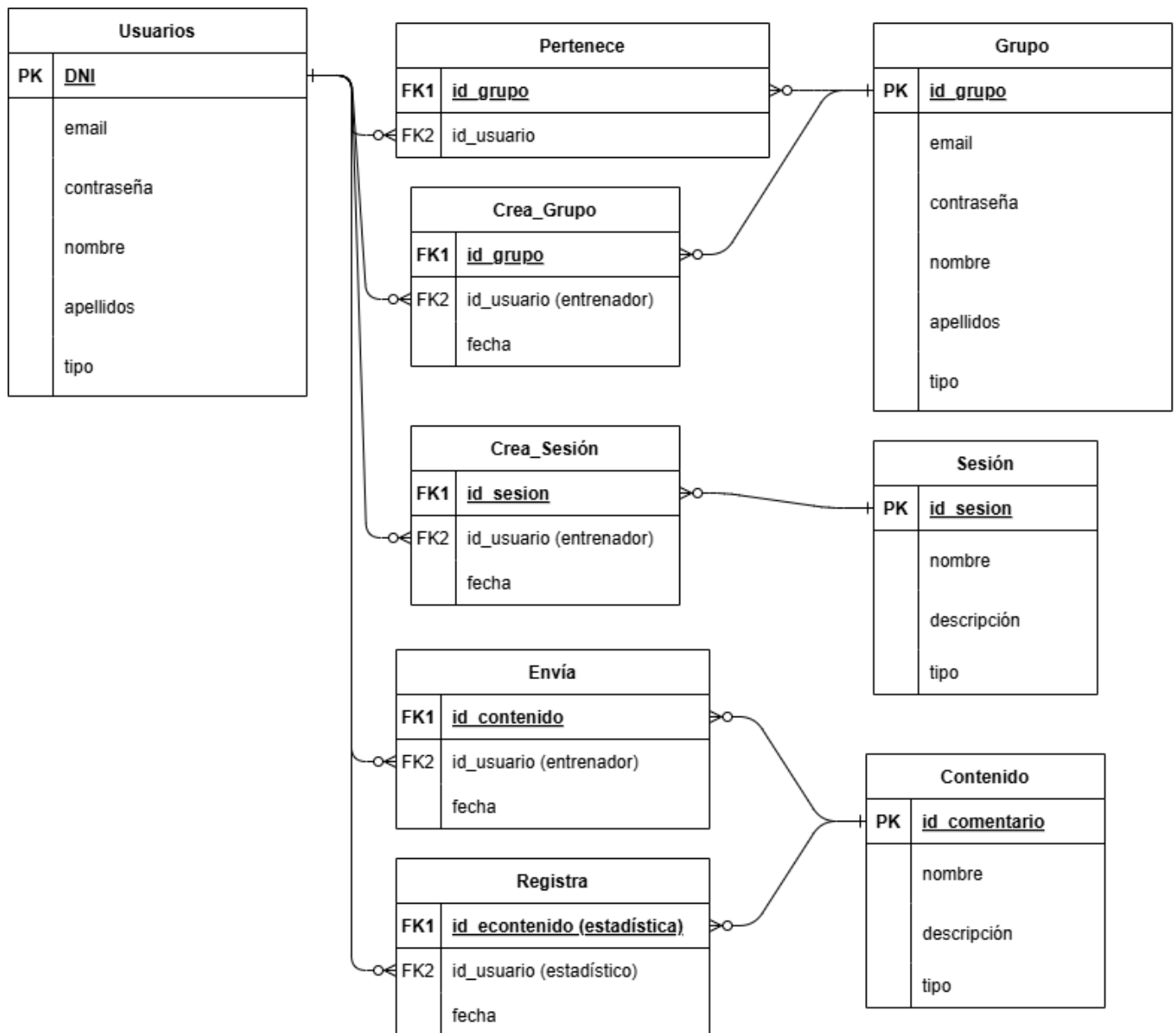
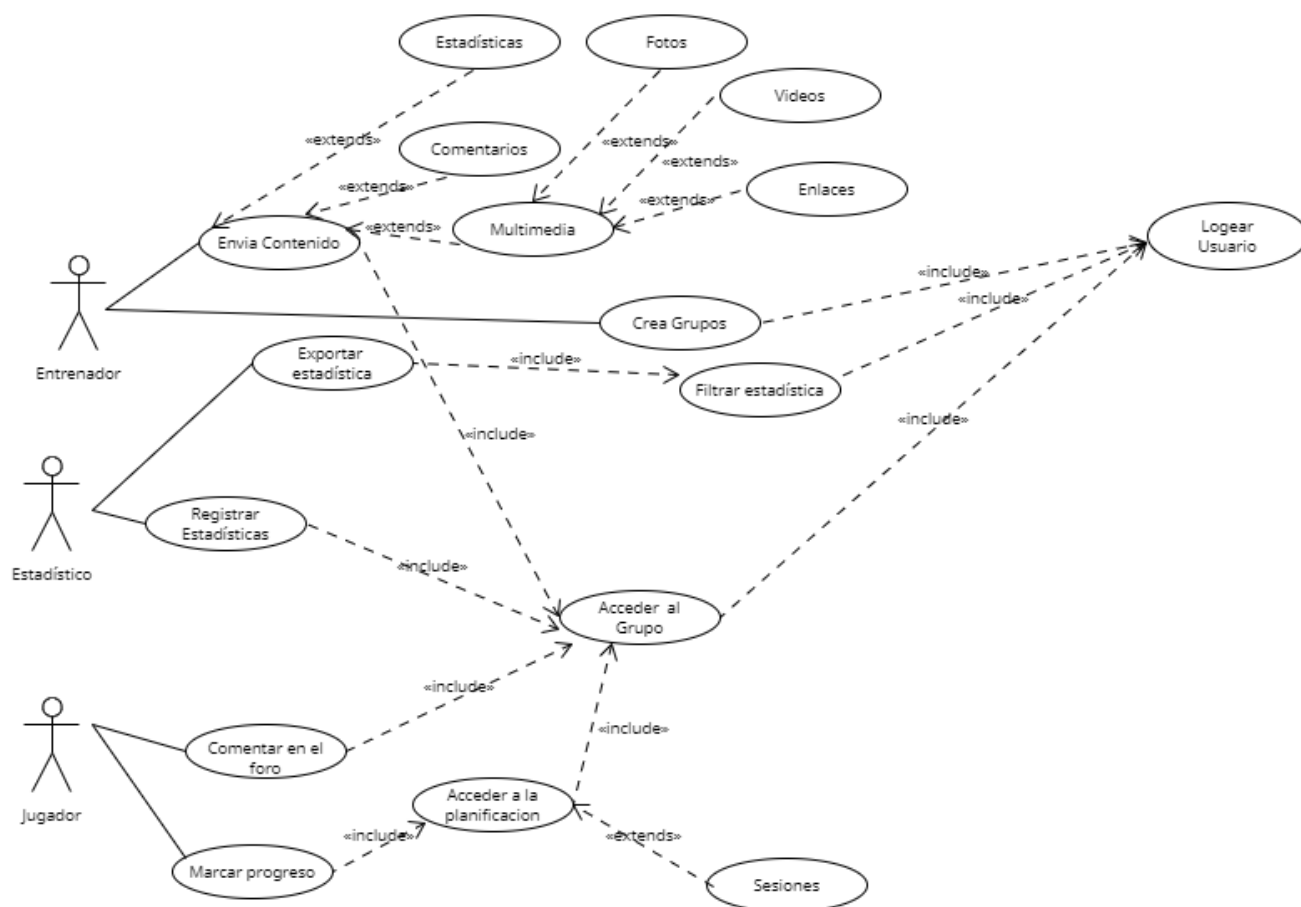


Figura 15: Modelo entidad relación de la base de datos



### 3.6. Análisis y diseño de sistema funcional (Diagrama de Casos De Uso)



Dibujo 1: Diagrama de Casos de uso



### 3.7. Análisis y diseño de la interfaz de usuario (Mockup)



Figura 16: Mockup 1



A Web Page

https://Register.html

### Registrate

Nombre	Apellidos
DNI (Será el nombre de Usuario)	Email
Contraseña	
Repetir Contraseña	

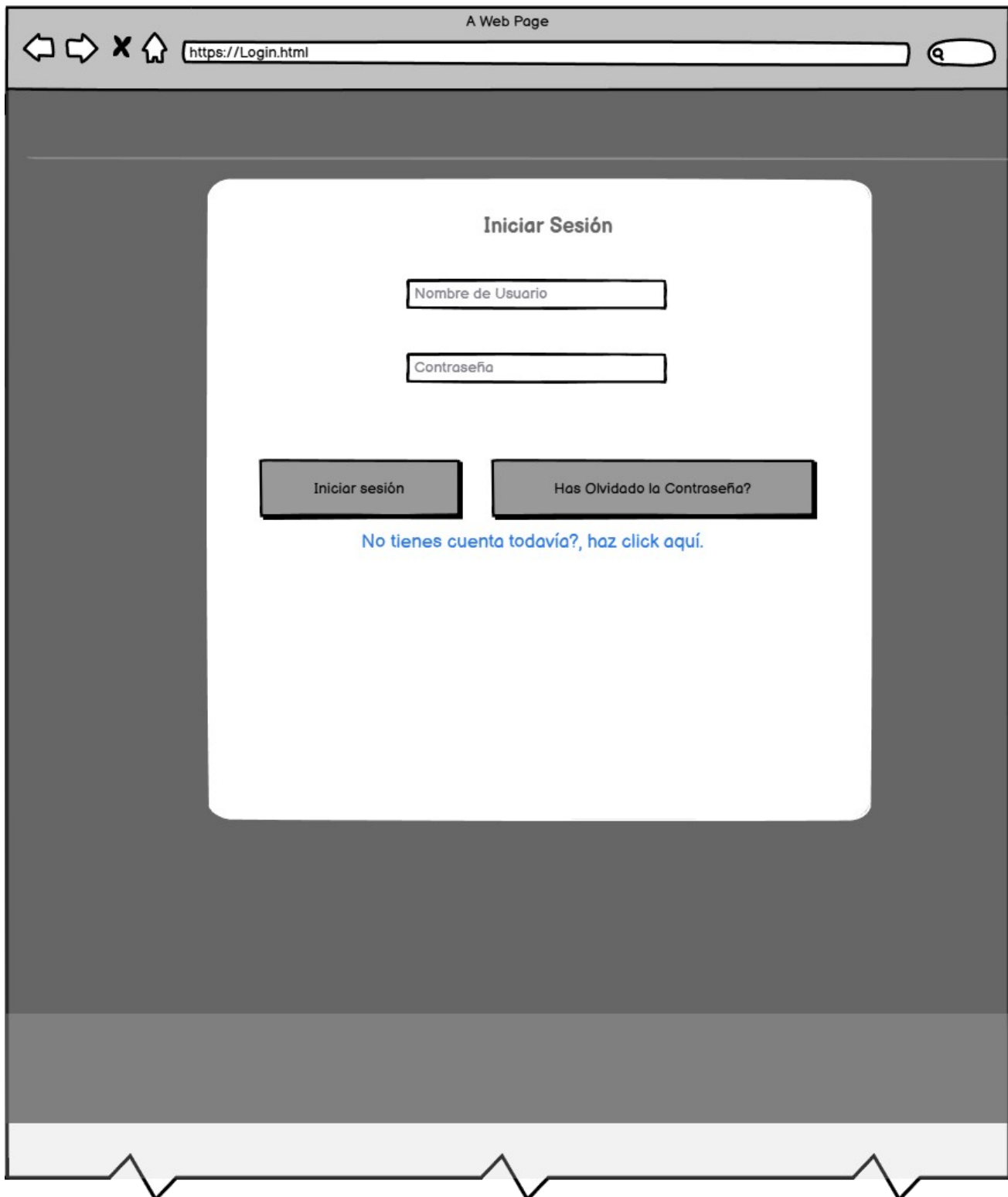
☐ He leído los Términos de Servicio

☐ Estoy de Acuerdo con la política de privacidad de la Empresa

Sign Up for Balsamiq Cloud

Figura 17: Mockup 2





A Web Page

https://Login.html

### Iniciar Sesión

Nombre de Usuario

Contraseña

Iniciar sesión

Has Olvidado la Contraseña?

[No tienes cuenta todavía?, haz click aquí.](#)

Figura 18: Mockup 3

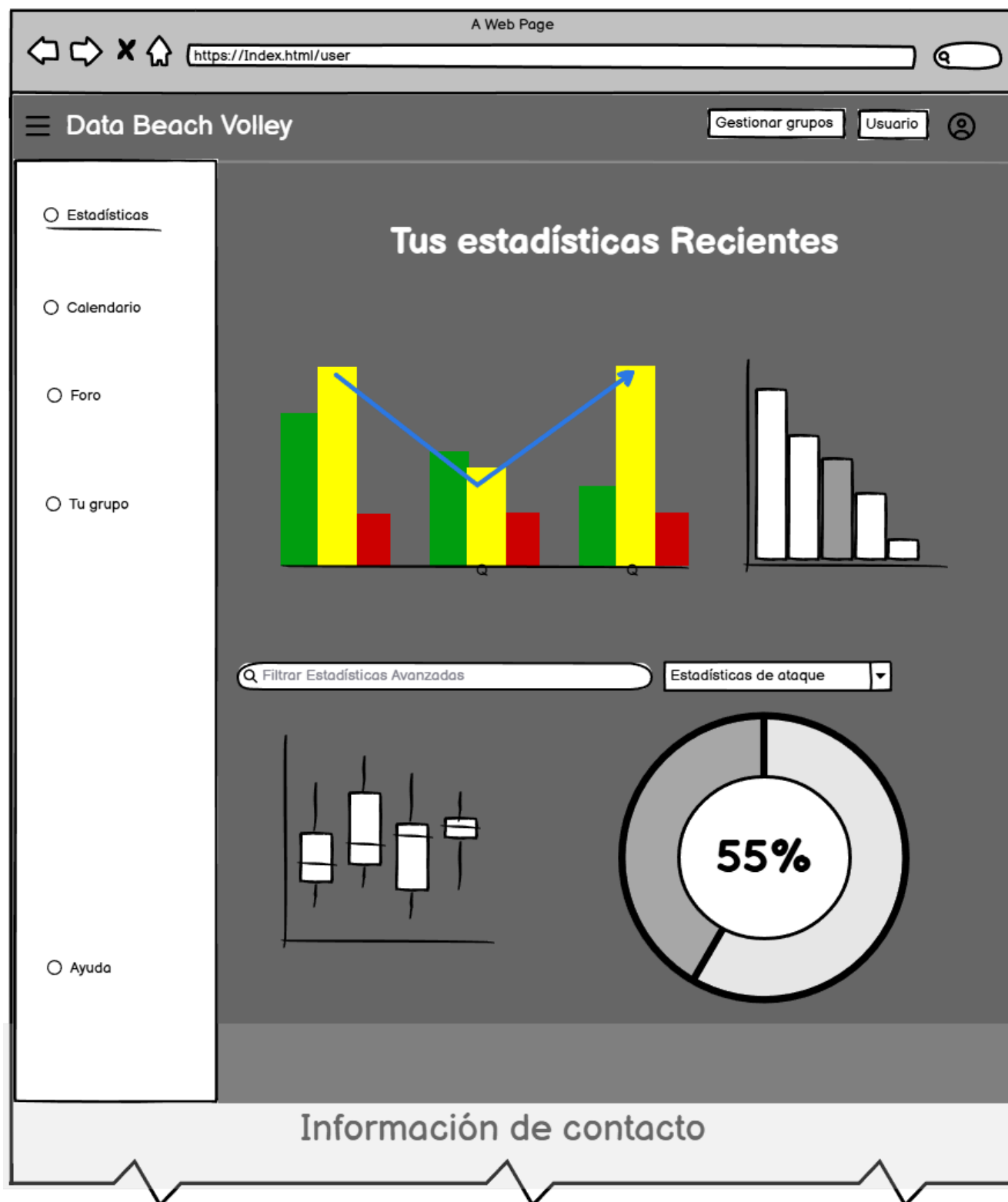


Figura 19: Mockup 4

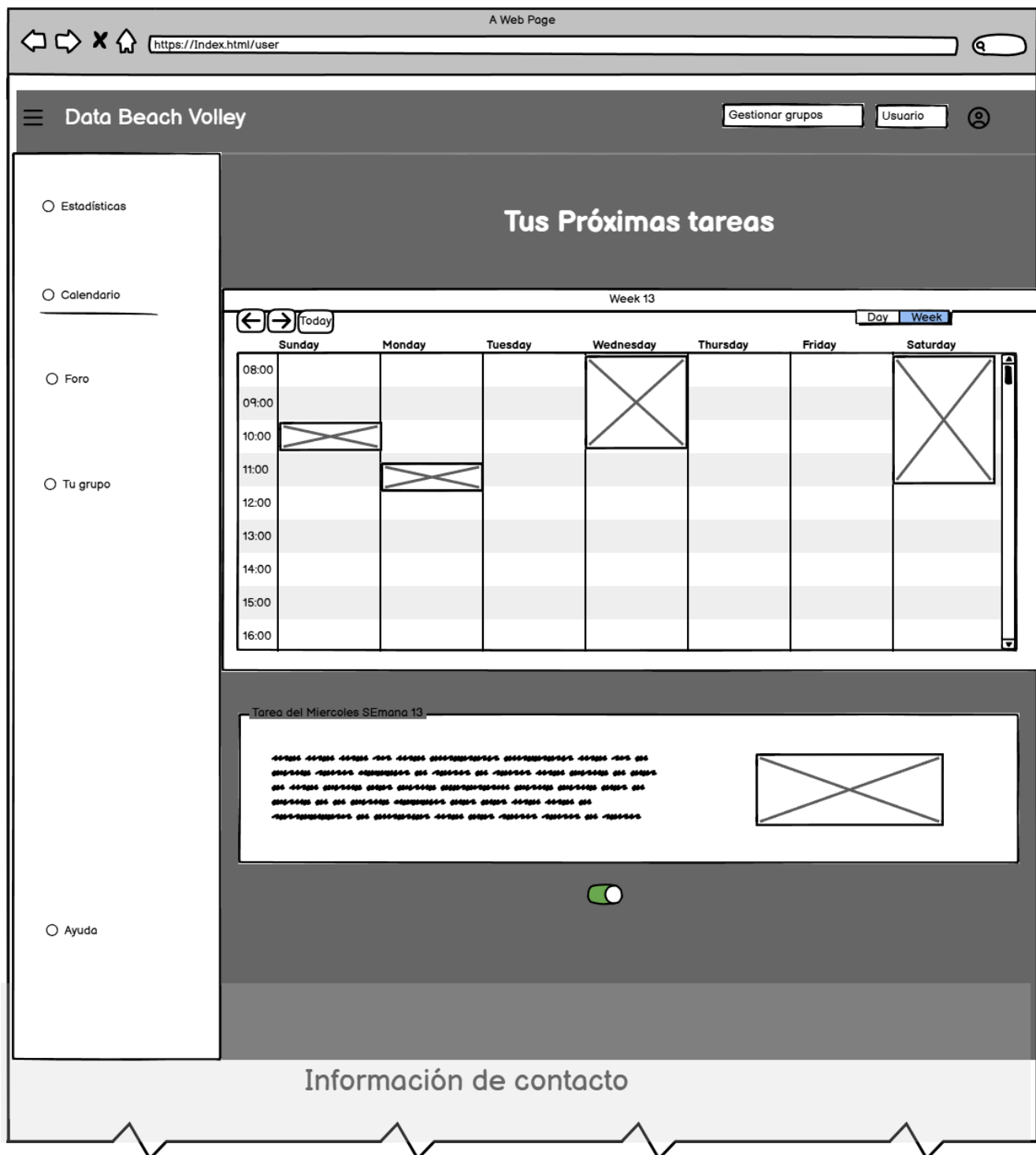


Figura 20: Mockup 5

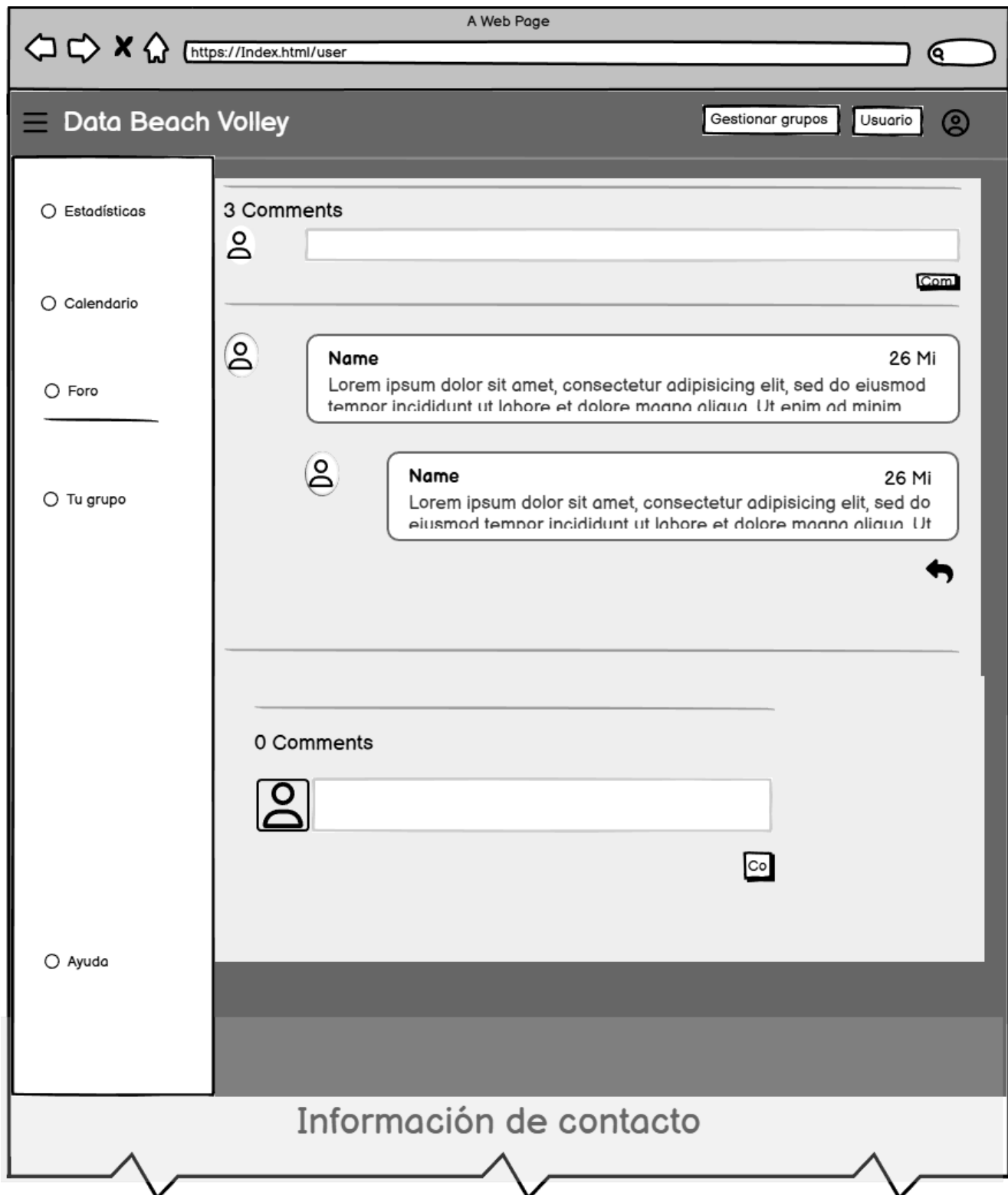


Figura 21: Mockup 6



## 4. Documento de Implementación e Implantación del Sistema

---

### 4.1. Implementación

#### Clases y Métodos Importantes

A continuación, se describen las clases y métodos más relevantes implementados en el backend utilizando el framework Django:

- **UsuariosAPIView:**
    - *Descripción:* Clase que permite listar todos los usuarios registrados en el sistema.
    - *Metodos clave:* queryset para definir la fuente de datos y serializer\_class para especificar el serializador.
    - *Propósito:* Garantizar acceso público a los datos de los usuarios con restricciones adecuadas.
  - **UserRegister:**
    - *Descripción:* Implementa el registro de nuevos usuarios con validaciones personalizadas.
    - *Metodos clave:*
      - post: Valida y limpia los datos antes de pasarlos al serializador para crear un nuevo usuario.
    - *Propósito:* Asegurar que los datos de entrada cumplen con los requisitos de seguridad y formato.
  - **ChangePasswordView:**
    - *Descripción:* Facilita el cambio de contraseña para usuarios autenticados.
    - *Metodos clave:*
      - put: Verifica la contraseña anterior y actualiza la nueva si es válida.
    - *Propósito:* Mejorar la seguridad al permitir a los usuarios gestionar sus credenciales.
  - **RecoverPasswordView:**
    - *Descripción:* Genera y envía una contraseña temporal por correo electrónico en caso de olvido.
    - *Metodos clave:*
      - post: Genera una contraseña aleatoria y actualiza la base de datos tras verificar el correo del usuario.
    - *Propósito:* Recuperación segura y eficiente de contraseñas.
  - **get\_user\_groups:**
-



- *Descripción:* Devuelve los grupos a los que pertenece un usuario.
- *Metodos clave:*
  - GET: Filtra grupos activos y proporciona información sobre sus miembros y creadores.
- *Propósito:* Facilitar la gestión de grupos y accesos.

## 4.2. Pruebas

### Plan de Pruebas

Tipo de prueba	Objetivo
Unitarias	Validar el correcto funcionamiento de funciones y métodos individuales.
Integración	Comprobar la interacción entre diferentes módulos y componentes del sistema.
Funcionales	Verificar que las características cumplen los requisitos especificados.
Seguridad	Identificar vulnerabilidades como inyección SQL y ataques XSS.(sin implementar)
Rendimiento	Asegurar que el sistema maneja cargas previstas sin degradación notable.(sin implementar)

### Ejemplo de Documentación de Pruebas

Caso de Prueba	Descripción	Resultados Esperados	Estado
Registro de Usuario	Comprobar que un usuario puede registrarse exitosamente.	Usuario creado con datos válidos.	Aprobado
Recuperación de Contraseña	Validar la generación de contraseña temporal.	Email enviado y contraseña actualizada.	Aprobado
Consulta de Grupos Activos	Verificar que se listan solo los grupos activos.	Lista correcta de grupos retornada.	Aprobado



### **4.3. Instalación y configuración**

#### Dockerización

Se ha implementado el uso de Docker para simplificar el despliegue y la configuración del sistema. Los pasos son los siguientes:

#### **1. Instalar Docker:**

- Descargar e instalar Docker desde su [sitio oficial](#).

#### **2. Construir la Imagen Docker:**

- Crear un archivo Dockerfile que incluya las dependencias y configuraciones necesarias.
- Ejecutar:  
`docker build -t data-beach-volley .`

#### **3. Ejecutar el Contenedor:**

- Configurar un archivo docker-compose.yml para definir servicios.
- Ejecutar:  
`docker-compose up`

#### **4. Acceso al Sistema:**

- La aplicación estará disponible en `http://localhost:8000` (por defecto)
-



#### **4.4. Manual de usuario**

##### **Descripción General**

Data Beach Volley es una aplicación para gestionar equipos de vóley playa, facilitando la toma y análisis de datos estadísticos.

##### **Puesta en Marcha**

1. Accede a la URL proporcionada tras la instalación.
2. Regístrate como usuario: jugador, entrenador o estadístico.
3. Inicia sesión y accede a tus grupos.

##### **Manejo de Menús**

- **Inicio:** Vista general de tus actividades.
- **Grupos:** Gestiona tus equipos y miembros.
- **Usuario:** Edita tu perfil y configura ajustes.
- **Foro:** Comunícate con tu equipo.

##### **Mensajes de Error**

- "Contraseña incorrecta": Verifica tus credenciales.
  - "No autorizado": Asegúrate de tener permisos adecuados.
-





## 5. Documento de Cierre

---

### 5.1 Diario de Bitácora

El desarrollo se organizó en tres sprints:

1. **Primer Sprint:** Arquitectura y gestión de usuarios.
2. **Segundo Sprint:** Implementación de foro y estadísticas.
3. **Tercer Sprint:** Gráficas y filtrado avanzado.

### 5.2 Temporalización y Desviaciones

- **Planificación inicial:** 12 semanas.
- **Desviación:** Incremento de 6 semanas debido a problemas de integración.

### 5.3 Resultados y Conclusiones

El proyecto alcanzó alguno de sus objetivos:

- Creación de una aplicación funcional.
- Aprendizaje y dominio de nuevas tecnologías.
- Posibilidades de mejora: optimización de rendimiento y expansiones futuras.

En resumen, el aprendizaje a lo largo de este proyecto ha sido notorio, tanto en temas meramente de conocimiento, como puede ser el aprendizaje de *django* y su uso, de un nuevo lenguaje de programación como python, etc. Pero lo más valioso y lo que mas valoro de lo que he aprendido ha sido la resolución de conflictos y problemas.

Estoy conforme con el funcionamiento de la aplicación final, pero no me satisface, ya que había planificado integrar muchas mas funcionalidades que al final, por la falta del tiempo empleado en subsanar errores de código no he podido implementar.

---