



**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**Trabajo fin de Grado**

**GIS**

**AlterUno**

**Realizado por  
Elena Tomás Vela**

**Dirigido por  
José Antonio Parejo Maestre**

**Departamento  
Departamento de Lenguajes y Sistemas Informáticos**

**Sevilla, 1 de marzo de 2017 (v.0.91)**

---

## Resumen

---

El resumen del Trabajo fin de Grado consiste, como su propio nombre indica, es un resumen de la memoria en formato apropiado para ser indexado en las bases de datos bibliotecarias. No debe ocupar más de una carilla de texto y en ella hay que exponer en pocas palabras la finalidad y objetivos del trabajo, así como las aportaciones realizadas. En general, no incluirá figuras, cuadros ni referencias bibliográficas.

El resumen es obligatorio en español para todos los TfG. Es opcional incluir un *Abstract* (resumen en inglés) en una hoja separada. Debe ser traducción del correspondiente resumen en español. En los TfG presentados en inglés, la inclusión del *abstract* es también obligatoria, así como en los TfG correspondientes a la titulación de Ingeniería de la Salud.

El resumen de este documento es el siguiente:

Este trabajo pretende ser una guía para uniformar los formatos de las memorias de los Trabajos fin de Grado de las titulaciones:

- Grado en Ingeniería Informática — Ingeniería de Computadores
- Grado en Ingeniería Informática — Ingeniería del Software
- Grado en Ingeniería Informática — Tecnologías Informáticas

de la Escuela Técnica Superior de Ingeniería Informática de la Universidad de Sevilla.

Al mismo tiempo, se pretende que el documento sea un ejemplo de la realización de un memoria de Trabajo fin de Grado. Debido a ello, hemos estructurado el documento en capítulos e incluido diversos índices y bibliografía, aunque obviamente no hubiera sido necesario.

---

## **Agradecimientos**

---

A nuestros alumnos y a nuestras alumnas.

---

## Índice general

---

<b>Índice general</b>	<b>III</b>
<b>Índice de cuadros</b>	<b>V</b>
<b>Índice de figuras</b>	<b>VI</b>
<b>Índice de código</b>	<b>VII</b>
<b>1 Formato</b>	<b>1</b>
1.1 Introducción . . . . .	1
1.2 Formato general . . . . .	1
1.3 Cuerpo y tipos de letra . . . . .	1
1.4 Márgenes y párrafos . . . . .	1
1.5 Lengua . . . . .	2
<b>2 Partes y plantillas</b>	<b>3</b>
2.1 Partes . . . . .	3
2.2 Impresión . . . . .	4
2.3 Plantillas . . . . .	4
<b>3 Ejemplos</b>	<b>5</b>
3.1 Manejo de la bibliografía . . . . .	5
3.2 Código . . . . .	5
3.3 Imágenes . . . . .	6
3.4 Cuadros (mal llamados Tablas) . . . . .	6
<b>4 Definición de objetivos</b>	<b>7</b>
4.1 Objetivos principales . . . . .	7
4.2 Objetivos técnicos . . . . .	7
4.3 Objetivos formativos . . . . .	8

<b>5</b>	<b>Análisis de requisitos, diseño e implementación</b>	<b>9</b>
5.1	Requisitos de información . . . . .	9
5.1.1	Usuarios . . . . .	9
5.1.2	Sala de espera . . . . .	9
5.1.3	Cartas . . . . .	9
5.1.4	Partida . . . . .	9
5.1.5	Jugadores . . . . .	10
5.2	Tipos de usuario . . . . .	10
5.3	Historias de usuario . . . . .	10
5.3.1	Autenticación de usuario . . . . .	10
5.3.2	Sala de espera . . . . .	11
5.3.3	Juego . . . . .	12
5.3.4	Chat . . . . .	12
5.4	Matriz de trazabilidad de requisitos . . . . .	13
5.5	Diseño e implementación . . . . .	13
5.5.1	Herramientas utilizadas . . . . .	13
5.5.2	Arquitectura del sistema . . . . .	14
<b>6</b>	<b>Análisis de antecedentes y aportación realizada</b>	<b>20</b>
<b>7</b>	<b>Análisis temporal y costes de desarrollo</b>	<b>21</b>
7.1	Análisis temporal . . . . .	21
7.2	Costes de desarrollo . . . . .	21
<b>8</b>	<b>Comparación con otras alternativas</b>	<b>22</b>
<b>9</b>	<b>Pruebas</b>	<b>23</b>
<b>10</b>	<b>Manual</b>	<b>24</b>
	<b>Apéndices</b>	<b>25</b>
	<b>Referencias</b>	<b>26</b>

---

# Índice de cuadros

---

3.1 Cuadro de prueba . . . . . 6

---

## Índice de figuras

---

3.1	Ada Lovelace . . . . .	6
5.1	Representación UML de la base de datos . . . . .	15
5.2	Mockup de la pantalla de inicio . . . . .	17
5.3	Mockup de la pantalla de inicio de sesión . . . . .	17
5.4	Mockup de la pantalla de registro . . . . .	18
5.5	Mockup de la pantalla de sala de espera . . . . .	18
5.6	Mockup de la pantalla del juego . . . . .	19

---

## Índice de código

---

3.1	Código de ejemplo en LaTeX . . . . .	5
-----	--------------------------------------	---



## Formato

---

### 1.1– Introducción

La memoria del Trabajo fin de Grado se entrega utilizando la aplicación de la escuela para Trabajos fin de Carrera, sita en <https://tfc.eii.us.es/TfG/>. Debe entregarse en formato PDF y podrá utilizarse cualquier aplicación informática para generar la misma, siempre que el resultado final verifique los requisitos que se exponen en este documento.

### 1.2– Formato general

Salvo excepciones justificadas, el PDF a entregar debe tener sus páginas del tamaño estandarizado más común en Europa: DIN A4. Con el objetivo de facilitar su lectura en dispositivos electrónicos, se considerará que se imprime a una sóla cara (ver 2.2 para los casos en que la memoria deba ser impresa).

### 1.3– Cuerpo y tipos de letra

Se recomienda utilizar, dada su simplicidad, claridad y legibilidad, los tipos de letra Arial (preferiblemente en caja alta) o Helvética, en un tamaño para el cuerpo del texto de 11 pt, con un interlineado sencillo o de 1,5.

Los títulos de capítulos, secciones y subsecciones, así como las notas al pie de texto<sup>1</sup> y las cabeceras o pies de cuadros, figuras y trozos de código quedan a libertad del redactor de la memoria. Sin embargo, es buena idea que los títulos tengan un tamaño igual o superior a 11pt. y los pies y cabeceras sean de tamaño igual o inferior a 11pt. Se ruega encarecidamente que, en lo posible, se evite el subrayado. Este puede sustituirse por el uso de **negritas** o *cursivas* o por el cambio de formato del tipo de letra (Se recomienda prestar atención y no abusar del cambio de [color](#), pues puede dar problemas de accesibilidad).

### 1.4– Márgenes y párrafos

Es necesario configurar la página seleccionando los márgenes siguientes:

---

<sup>1</sup>Por favor, no abusad de las notas a pie de texto

- Márgenes superior e inferior: 2,5 cm.
- Márgenes laterales (izquierdo y derecho): 3 cm.

En el caso de no usar cabeceras, las páginas deberán ir numeradas en el centro del pie. Si se usan cabeceras, no se utilizarán estas en las páginas que comienzan capítulo, las cuales se numerarán en el centro del pie. Las restantes páginas pueden ir numeradas en el pie o en la cabecera, pero deberán mantener coherencia de formato a lo largo de todo el capítulo.

Las páginas previas al cuerpo de la memoria del Trabajo fin de Grado (agradecimientos, resumen, índices,...) pueden no numerarse o numerarse independientemente de la misma, en cuyo caso se numerarán con números romanos. Se recuerda que los números romanos se escriben con letras mayúsculas (la numeración i,ii,iii,iv... es propia del idioma inglés y no es admisible en español, ni siquiera para enumeraciones)

Los párrafos comenzarán con sangrado. El espacio entre los mismos no debe ser excesivo.

## 1.5– Lengua

La Normativa académica de los Trabajos fin de Grado indica:

*Como norma general, el TfG deberá estar escrito y ser expuesto oralmente en castellano. Podrá también estar escrito y ser expuesto en inglés, previa solicitud.*

En cualquier caso, la memoria debe respetar los usos y costumbres del idioma en que sea escrita. Debe prestarse especial atención al guionado de las palabras, debido a que muchas aplicaciones informáticas usan el propio del inglés y no el del castellano.

En cuanto a la portada, se debe utilizar la portada oficial de la ETSII. Un ejemplo de la misma puede encontrarse al principio (como portada) de este documento.

### Partes y plantillas

---

#### 2.1– Partes

Todas las memorias de Trabajo fin de Grado deberán constar de las siguientes partes

- Portada (según formato oficial). No debe incluir número de página. Debe incluir:
  - Sello de la universidad de Sevilla a dos tintas
  - ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA
  - Trabajo fin de Grado
  - *Denominación del Grado*
  - Realizado por: *Nombre y apellidos del estudiante*
  - Dirigido por: *Nombre y apellidos del tutor o tutores*
  - Departamento *Nombre del departamento en el que se lee el Tfg*
  - Sevilla, *Mes y año de la convocatoria de entrega*
- Preliminares: las páginas no se numeran o se numeran con números romanos.
  - Resumen en castellano (máximo una hoja)
  - Abstract (resumen en inglés, obligatorio para el caso de las memorias escritas en inglés, opcional para las escritas en castellano)
  - Agradecimientos (opcional)
  - Índice general (contenido de la memoria, con mención de las partes en que está dividida)
  - Índice de figuras (opcional)
  - Índice de cuadros (mal llamados en general, tablas) (opcional)
  - Índice de código o algoritmos (opcional).
- Cuerpo de la memoria, dividida en capítulos. El contenido de la memoria ha de incluir los elementos característicos de un proyecto de ingeniería o de un estudio o trabajo en el ámbito de una investigación, los cuales, en el sentido más amplio, son:
  - Definición de objetivos.
  - Análisis de antecedentes y aportación realizada.

- Análisis temporal y de costes de desarrollo.
- Análisis de requisitos, diseño e implementación.
- Manual de usuario, en su caso.
- Pruebas.
- Comparación con otras alternativas.
- Conclusiones y desarrollos futuros

Estos puntos podrán ser ajustados y modificados en función de la naturaleza del proyecto realizado.

- Bibliografía: se deben documentar las fuentes bibliográficas utilizadas en el formato APA 2009.
- Índice alfabético o glosario (Lista ordenada de los conceptos, los nombres propios, etc.; que aparecen en la memoria, con las indicaciones necesarias para su localización)(opcional)
- Apéndices: Si la memoria contiene algún artículo de investigación o similar, este se incluirá en los Apéndices. Es necesario, en ese caso, incluir en la página anterior una hoja con la citación bibliográfica. En esta citación, el título del artículo se debe enlazar con la página web de la revista en la que aparecen el resumen o abstract y el acceso al texto completo.

La numeración de las páginas de la bibliografía y del glosario debe continuar la del cuerpo de la memoria. Los apéndices pueden llevar su propia numeración independiente o usar la general del cuerpo de la memoria.

## 2.2– Impresión

La entrega del memoria y en su caso, el depósito en biblioteca, se hacen en formato electrónico. Debido a ello, la memoria se presentará *a una cara*. Si se requiriera por algún motivo la impresión de la misma, se recomienda vivamente preparar la memoria adecuadamente para su impresión. Algunas sugerencias:

- Dejar una página en blanco cuando sea necesario para que los capítulos comiencen siempre en página impar (derecha)
- Ajustar los márgenes para que el exterior sea ligeramente más grande que el interior.
- Ajustar las cabeceras y pies de página (en su caso). Por ejemplo, si el número de página ocurriese en un lateral de la cabecera o pie, este debe ser siempre el exterior (derecho para las páginas impares, izquierdo para las pares)

## 2.3– Plantillas

Se encuentra disponible una plantilla LaTeX en la sección de documentos de la plataforma de la aplicación de TfG de la ETSII <https://tfc.eii.us.es/TfG/>. Esa plantilla ha sido utilizada para preparar este documento. Se espera en breve disponer de plantillas de ejemplo para las aplicaciones OpenOffice Writer y Office Word.

Nota: Las plantillas se proporcionan como ejemplo, las condiciones obligatorias son las que constan en este procedimiento.

## Ejemplos

### 3.1– Manejo de la bibliografía

En esta sección mostramos brevemente ejemplos de referencia a la bibliografía citando un libro (Sou04, ou04), un artículo (Bez07, ez07) y una página web (Aut14, ut14).

Se recuerda que son campos obligatorios en todos los ítems de la bibliografía: autor(es), título del libro o artículo y año de publicación. En el caso de páginas web, es obligatoria la fecha de la última consulta. En general, la bibliografía debe ayudar al lector a encontrar fácilmente los ítems citados.

### 3.2– Código

En general se debe evitar incluir código o pseudocódigo en la memoria. Si fuese preciso, se destacará de forma que sea fácilmente identificable y se indexarán los trozos de código incluidos. Un ejemplo puede verse a continuación.

```

1  %COMANDO PARA INSERTAR UN CUADRO UTILIZANDO EL FORMATO:
2  %1---> especificar numero de columnas y su alineacion ejm:
3      % |r||c||c| r=right, c=center, l=left
4  %2---> especificar el caption o titulo de la figura
5  %3---> label para hacer referencia a la tabla insertada
6  %4---> contenido de tabla separando columnas con & y filas con \\
7  \newcommand{\cuadro}[4]{
8      \begin{table}[htb]
9          \centering
10         \begin{tabular}{#1}
11             \hline
12             #4
13             \hline
14         \end{tabular}
15         \caption{#2}
16         \label{#3}
17     \end{table}
18 }
```

Código 3.1: Código de ejemplo en LaTeX

### 3.3– Imágenes

Este es un ejemplo de inclusión de figura en el texto (véase la figura 3.1).



Figura 3.1: Ada Lovelace

Figuras y cuadros se colocarán preferentemente tras el párrafo en el que son llamados por primera vez. Si no cupieran, se colocarán (en orden de preferencia):

- Al final de la página en que se llaman
- Al principio de la siguiente página
- Al final del capítulo

siempre respetando el orden de aparición en el texto.

### 3.4– Cuadros (mal llamados Tablas)

Este es un ejemplo de inclusión de cuadro en el texto. Véase el cuadro 3.1

elemento	elemento	elemento
elemento	elemento	elemento

Cuadro 3.1: Cuadro de prueba

### Definición de objetivos

---

Los objetivos a cumplir durante la realización de este proyecto se pueden dividir en varias categorías.

#### 4.1– Objetivos principales

- Proporcionar un entorno de juego en el que los jugadores puedan divertirse.
- Permitir que los jugadores puedan interactuar y relacionarse durante las partidas
- Crear una experiencia customizable para los jugadores.

#### 4.2– Objetivos técnicos

En el amplio mundo de la programación de páginas web, hay muchas tecnologías entre las que escoger que no se han dado durante el Grado. Por ello, se quiere usar esta oportunidad para explorar las nuevas herramientas que se pueden usar para estos fines.

Especialmente, se quiere experimentar con el uso de:

**Bun:** Lanzado a finales de 2023, se trata de un conjunto de herramientas para el desarrollo en JavaScript, prometiendo un mayor rendimiento que Node.js, aunque manteniendo compatibilidad con sus paquetes. Entre sus características se encuentran la ejecución nativa de ficheros de TypeScript, ejecución de tests, y gestión de paquetes.

**MongoDB:** Durante la carrera, se ha hecho mucho hincapié en bases de datos relacionales. Con la incorporación de MongoDB, se quiere explorar el mundo de las bases de datos no relacionales, además de hacer uso de sus puntos fuertes como el mejor rendimiento y escalabilidad, que hacen de ello una herramienta idónea para aplicaciones a tiempo real.

**React:** Aunque este framework se haya usado en algunos de los proyectos del Grado, la popularidad de este framework es innegable, debido a que se sigue usando frecuentemente hoy en día. Adicionalmente, se quiere explorar su combinación con otras librerías de componentes y de estilos, para así ver las posibilidades que puede abarcar el diseño web.

### **4.3– Objetivos formativos**

Adicionalmente, este proyecto pretende poner en práctica todos los conocimientos adquiridos durante el Grado mediante la abarcación de un proyecto mayor.



### Análisis de requisitos, diseño e implementación

---

#### 5.1– Requisitos de información

##### 5.1.1. Usuarios

- Nombre de usuario
- Contraseña

##### 5.1.2. Sala de espera

- Usuarios en la sala
- Reglas de la casa
- Baraja escogida

##### 5.1.3. Cartas

- Color
- Símbolo

##### 5.1.4. Partida

- Jugador del turno actual
- Cartas en la baraja
- Cartas en la pila de descartes
- Jugadores asociados
- Orientación de turnos

### 5.1.5. Jugadores

- Cartas en mano

## 5.2– Tipos de usuario

En la aplicación, solo habrá una distinción de los usuarios en la sala de espera. En los demás aspectos, todos los usuarios tendrán los mismos permisos.

En la sala de espera, habrá 2 roles distintos:

**Jugador normal:** Usuario que únicamente puede marcar si está listo o no y mandar mensajes en el chat.

**Huésped:** Usuario que ha creado la sala de espera y puede modificar las reglas y mazo a usar en el juego.

## 5.3– Historias de usuario

### 5.3.1. Autenticación de usuario

#### Registro de usuario

Como usuario sin autenticar, quiero poder registrarme en el sistema con un usuario y contraseña para poder acceder al juego.

**Caso positivo** El usuario rellena un formulario en el que pone de nombre de usuario Pepito123 y contraseña MeGustanLosJabalies, tras hacer click en el botón de enviar, se mandarán los datos al servidor, que creará el usuario correspondiente e iniciará la sesión del usuario.

#### Inicio de sesión de usuario

Como usuario sin autenticar, quiero poder usar mi usuario y contraseña para acceder a mi cuenta dentro de la aplicación y así acceder al juego.

**Caso positivo** El usuario sin autenticar, que se ha registrado anteriormente como Pepito123 y ha puesto de contraseña MeGustanLosJabalies, rellena un formulario en el que provee dichos datos, y tras darle al botón de enviar el sistema comprobará que la contraseña es la correcta para ese usuario y le dejará iniciar sesión.

**Caso negativo 1** Un usuario sin autenticar, que no se ha registrado anteriormente, provee en el formulario de inicio de sesión datos de usuario y contraseña que no existen en el sistema. El formulario se vuelve a mostrar, mostrando un mensaje de que los datos dados son incorrectos.

**Caso negativo 2** El usuario sin autenticar provee en el formulario un nombre correcto de usuario, pero pone una contraseña incorrecta. El formulario se vuelve a mostrar, con un mensaje mostrando que los datos introducidos son incorrectos.

### 5.3.2. Sala de espera

#### Crear una sala de espera

Como usuario autenticado, quiero poder crear una sala de juego, para esperar a que otras personas entren antes de empezar el juego.

**Caso positivo** El usuario, tras autenticarse, hace click en un botón de “crear partida” y se le llevará directamente a una nueva sala de espera

**Caso negativo** El usuario, sin autenticar, hace click en un botón de “crear partida”, pero como no está autenticado, se le llevará a la página de autenticarse, indicando que debe iniciar sesión para crear la partida.

#### Unirse a una sala de espera

Como usuario, quiero poder acceder a una sala de juego ya existente, para poder participar en el juego con los jugadores de dicha sala.

**Caso positivo 1** El usuario, tras autenticarse, hace click en un botón de “unirse a partida”, donde se le presentará un formulario donde podrá proveer el código de la partida, y una vez introduzca un código válido y le dé al botón de enviar, se le llevará a la sala de espera correspondiente

**Caso positivo 2** El usuario, tras autenticarse, usa un enlace que le ha dado el huésped de la partida que va a jugar, y al abrirlo se meterá automáticamente en la sala de espera

**Caso negativo 1** El usuario, sin autenticarse, hace click en un botón de “unirse a partida” y se le llevará a la página de autenticación, indicando que tiene que iniciar sesión.

**Caso negativo 2** El usuario, sin autenticarse, usa un enlace que le ha dado el huésped de la partida, y al abrirlo se le redirigirá a la página de autenticarse, indicando que tiene que iniciar sesión.

**Caso negativo 3** El usuario, tras autenticarse, pone en el formulario de unirse a una partida un código de una partida que no existe. Al darle click al botón de enviar, se le indicará que no existe esa partida y se le mostrará de nuevo el formulario.

**Caso negativo 4** El usuario, tras autenticarse, pone en el formulario de unirse a una partida un código de una partida que está en curso. Al darle click al botón de enviar, se le indicará que la partida está en curso y se le mostrará de nuevo el formulario.

#### Marcar estado de “listo” en sala de espera

Como usuario, quiero poder marcar que estoy listo para empezar la partida, para así comunicar al huésped de la sala que puede comenzar la partida sin miedo a que yo me la pierda.

**Caso positivo** Un jugador no huésped le da a un botón de “Listo”, y para todos los demás jugadores y el huésped se les mostrará dicho estado.

### **Iniciar partida en sala de espera**

Como usuario y huésped en una sala de espera, quiero empezar la partida una vez haya observado que se han unido a la sala todas las personas que esperaba y que estén listas, para poder jugar con todas ellas.

**Caso positivo** El jugador huésped de una sala de espera, tras observar que todos los demás jugadores están listos para empezar la partida, hace click en un botón de Empezar Partida, y, tras ello, todos los jugadores serán redirigidos a la partida.

### **Configurar Reglas de la Casa**

Como usuario, al crear una sala de espera quiero poder escoger las reglas especiales a aplicar, para acomodar las preferencias mías y de mis compañeros al jugar.

**Caso positivo** El jugador huésped de una sala de espera hace click en la casilla de la regla de la intercepción para activarla durante la partida.

Como jugador dentro de una partida, quiero poder especificar que digo “Uno” cuando me quedo con 1 carta en mano para poder cumplir con las normas del juego.

### **Configurar mazo**

Como usuario, quiero poder configurar el mazo de cartas de la partida antes de comenzarla, para customizar la experiencia del juego a mi gusto.

**Caso positivo** El jugador huésped de una sala de espera hace click en una casilla correspondiente al mazo que se quiera escoger. Cuando se inicie el juego, este usará la baraja seleccionada.

## **5.3.3. Juego**

Como usuario, quiero poder jugar al juego de cartas con otros jugadores, para poder socializar y divertirme.

## **5.3.4. Chat**

Como usuario, quiero poder comunicarme con otros jugadores en la partida y en la sala de espera, para así poder socializar durante esta y coordinarme con los demás jugadores para coordinarnos en las customizaciones que queramos aplicar.

**Caso positivo** El jugador dentro de una sala de espera rellena un campo de texto para el chat y le da al botón de enviar. Los demás jugadores en la sala reciben el mensaje en la sección del chat.

**Caso positivo** El jugador dentro de un juego rellena un campo de texto para el chat y le da al botón de enviar. Los demás jugadores en la sala reciben el mensaje en la sección del chat.

## 5.4– Matriz de trazabilidad de requisitos

Requisitos	Entorno de ocio	Socialización	Experiencia customizable
Autenticación de usuario			
Inicio de sesión			
Sala de espera		✓	✓
Configuración reglas de la casa			✓
Configuración mazos de cartas			✓
Chat	✓	✓	
Juego	✓	✓	

## 5.5– Diseño e implementación

### 5.5.1. Herramientas utilizadas

#### Git

Git es un sistema de control de versiones

#### MongoDB

MongoDB es un sistema de bases de datos NoSQL diseñado para ser altamente escalable. Sus documentos, que representan entidades de la base de datos, poseen una estructura parecida a la de un archivo JSON, haciendo que su uso sea más intuitivo para el desarrollador.

#### JavaScript

#### Bun

Bun es un nuevo conjunto de herramientas de JavaScript, lanzado a finales de 2023. Su mayor objetivo es proveer una mejor experiencia para el desarrollador, sin sacrificar en velocidad y optimización.

Algunas de sus funcionalidades constan de:

- Gestión de paquetes, con compatibilidad con Node, permitiendo una fácil migración.
- Runtime con soporte nativo con archivos con TypeScript y JSX
- Rápida ejecución de tests con sintaxis similar a Jest y con soporte de dobles de prueba.

### **Elysia**

Elysia es un framework de JavaScript que hace uso de Bun y TypeScript para proveer una mejor experiencia en el desarrollo de APIs.

Posee una estructura simple de usar y, según sus estadísticas, tiene un rendimiento considerablemente mayor que otros frameworks como Spring, FastAPI, Nest y Express.

### **Vite**

### **React**

React es un framework de JavaScript intencionado para el desarrollo web, facilitando la experiencia gracias a su sistema de componentes y su capacidad para manejar estados en ellos, permitiendo manejar los datos a mostrar fácilmente.

Siendo uno de los frameworks más populares, tiene una amplia documentación y contenido hecho por la comunidad que hace más accesible su aprendizaje.

### **Tailwind**

Tailwind se trata de un framework para CSS que crea clases de utilidad para el estilado de componentes. Su filosofía proviene de la prevención de abstracciones prematuras, y provee valores por defecto para factores como márgenes, espaciados, etc. que hacen la consistencia visual una tarea sencilla. Adicionalmente, hace una buena combinación con componentes en frameworks como React, permitiendo las abstracciones justas y necesarias, en vez de lidiar con el dificultoso mantenimiento de CSS semántico.

### **Shadcn.ui**

#### **5.5.2. Arquitectura del sistema**

Se realizará una arquitectura por capas compuesta por:

- Capa de presentación
- Capa de servidor
- Capa de datos

#### **Capa de datos**

Se realizará usando MongoDB para una base de datos NoSQL, debido a que puede acomodar más fácilmente la estructura del juego, comparado con una base de datos relacional.

Esto también resulta en mayor rendimiento, ya que se evitan algunas operaciones de unión de tablas.

#### **Esquema de la base de datos**

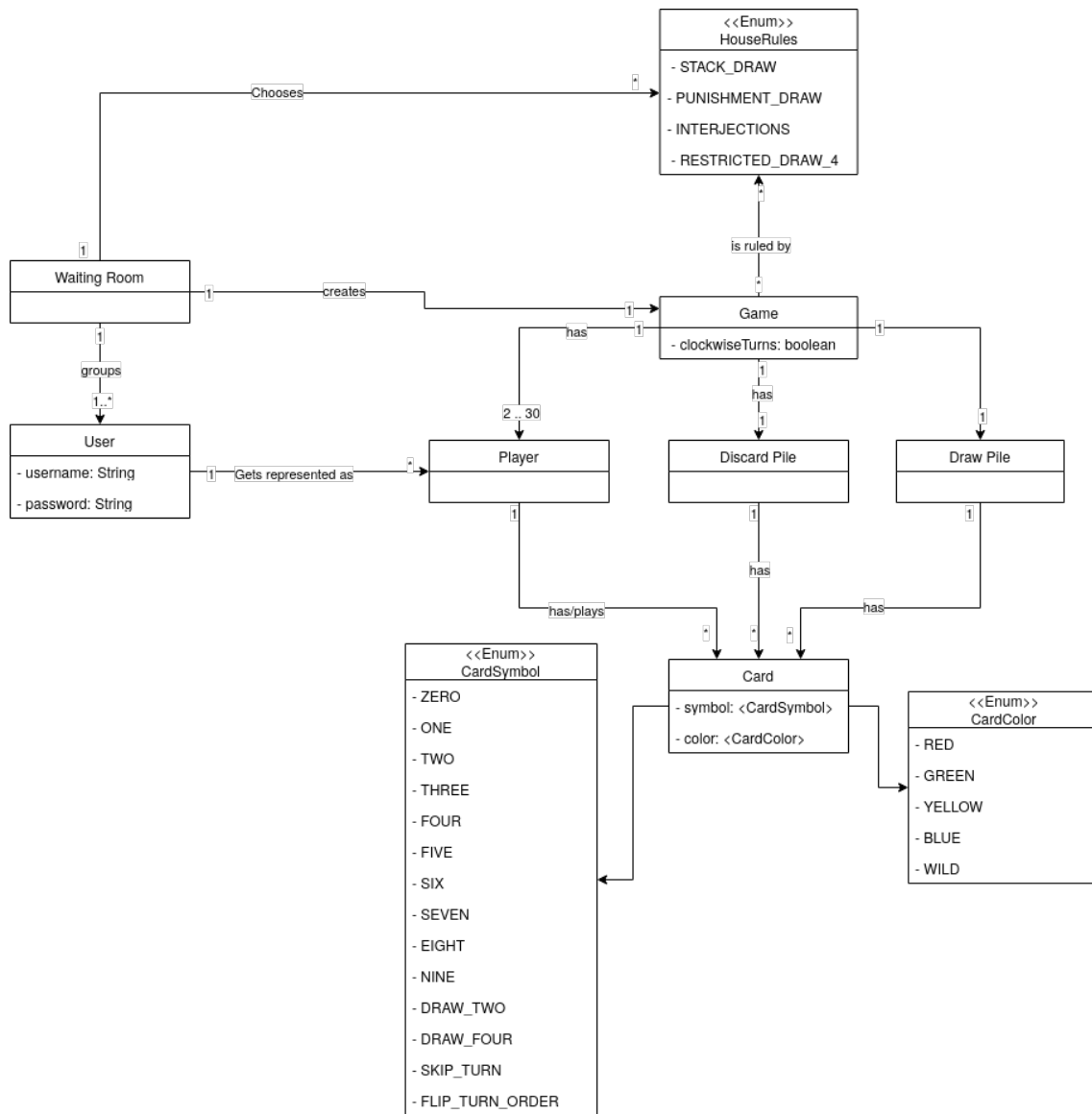


Figura 5.1: Representación UML de la base de datos

### Capa de servidor

Se creará en lenguaje JavaScript debido a la familiaridad con el mismo, aunque se añadirá al mismo tiempo TypeScript.

Al mismo tiempo, se incorporará Bun, ya que provee mayor rendimiento y soporte nativo para TypeScript, así como otras funcionalidades como la ejecución de tests.

Como framework, se hará uso de Elysia como framework de Backend, ya que está intencionado para su uso con Bun y es soportado por varias plataformas de despliegue.

**Documentación de la API:** La API del servidor se documentará con las especificaciones de OpenAPI para las rutas normales. Las rutas de WebSocket se documentarán manualmente.

#### Rutas de la API REST:

- /user

**POST:** Registra un nuevo usuario

**username:** Nombre de usuario

**password:** Contraseña

- /login

**POST:** Inicia sesión del usuario

**username:** Nombre de usuario

**password:** Contraseña

- /room

**POST:** Crea una sala de espera nueva asociada al usuario actual

#### Rutas de la API WebSockets:

- Test

### Capa de presentación

Se realizará también con JavaScript y Bun, por razones similares a las de la capa de servidor.

Se usará el framework de React debido a su gran cantidad de documentación y popularidad, así como por la experiencia previa que se ha tenido con dicho framework.

Adicionalmente, otras tecnologías que se usarán en este apartado son Vite para utilidades de desarrollo, y para el estilado se usará Tailwind, así como Shadcn.ui.

Tailwind se ha escogido debido a que simplifica el proceso de diseño de estilos, y gracias a los componentes de React se pueden abstraer y reutilizar fácilmente a través de toda la aplicación.

Shadcn.ui, por otro lado, se ha escogido debido a que proporciona componentes que se instalan directamente en el código en vez de depender de una librería externa. Esto logra que los componentes sean muy customizables y se puedan usar como punto de partida, ya que proveen los detalles más difíciles como la accesibilidad ya integrados.

### Mockups



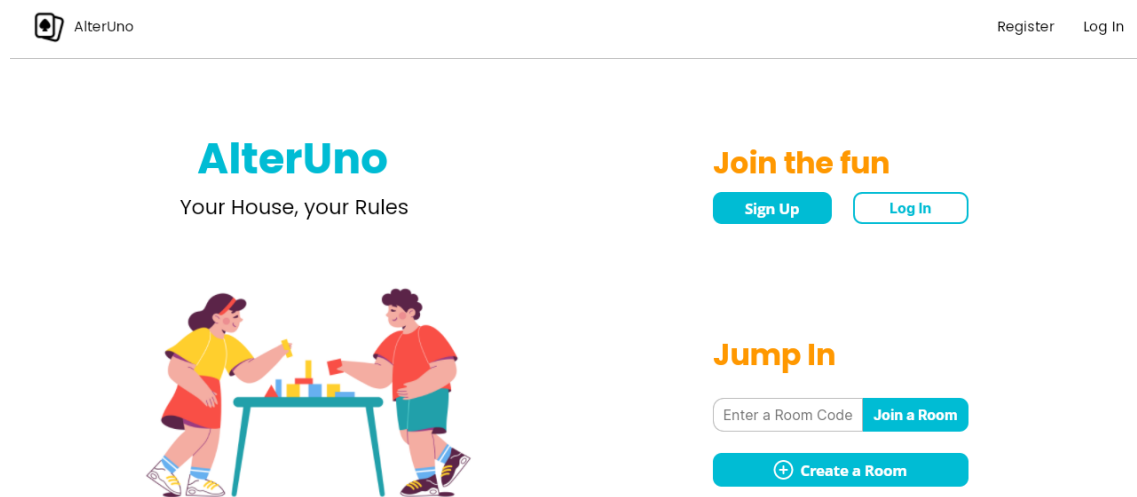


Figura 5.2: Mockup de la pantalla de inicio

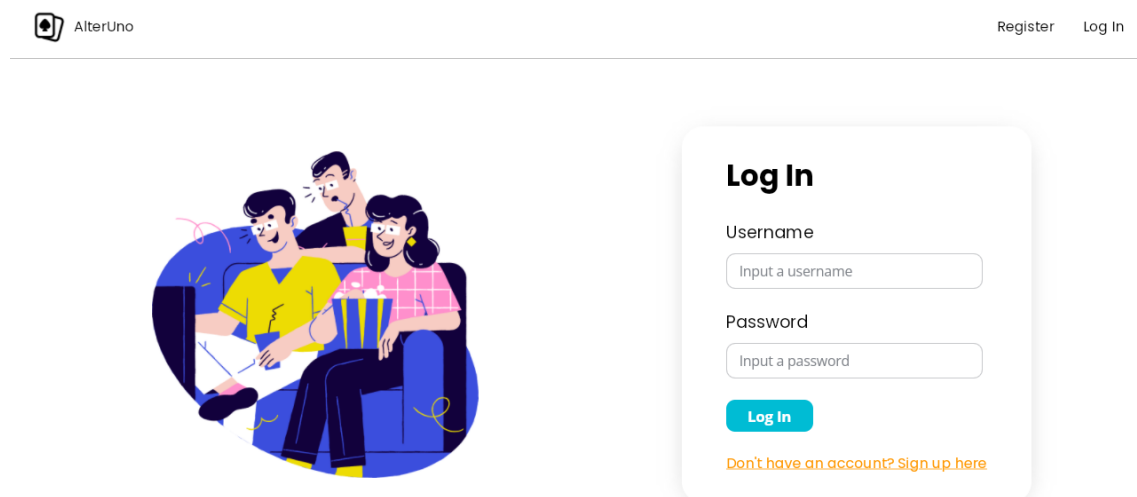


Figura 5.3: Mockup de la pantalla de inicio de sesión

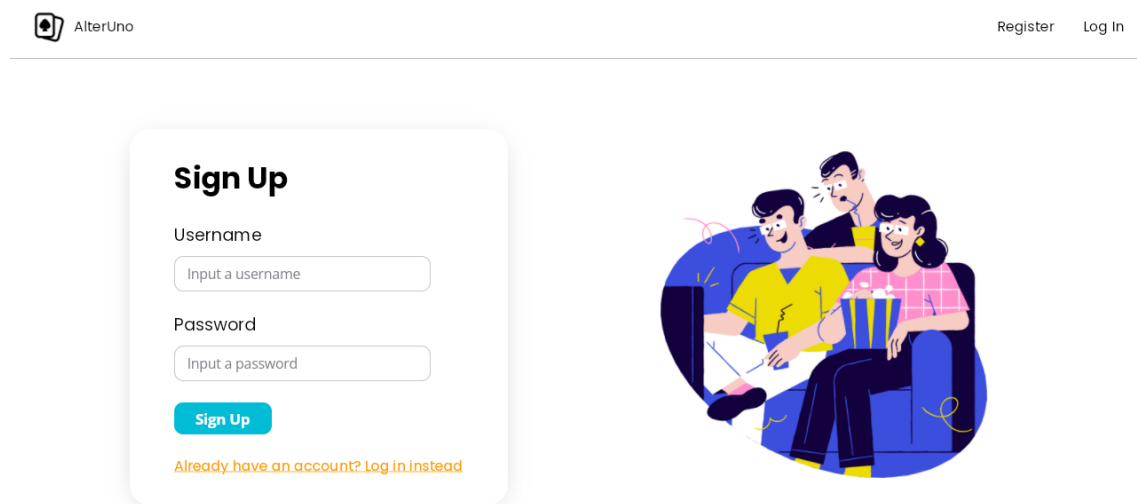


Figura 5.4: Mockup de la pantalla de registro

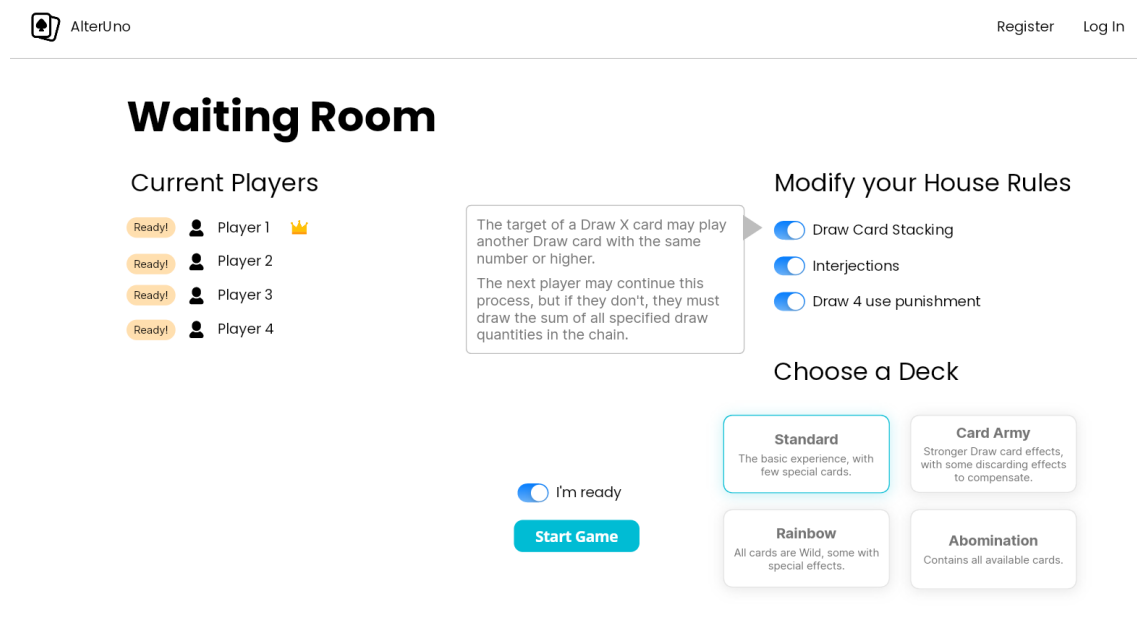


Figura 5.5: Mockup de la pantalla de sala de espera

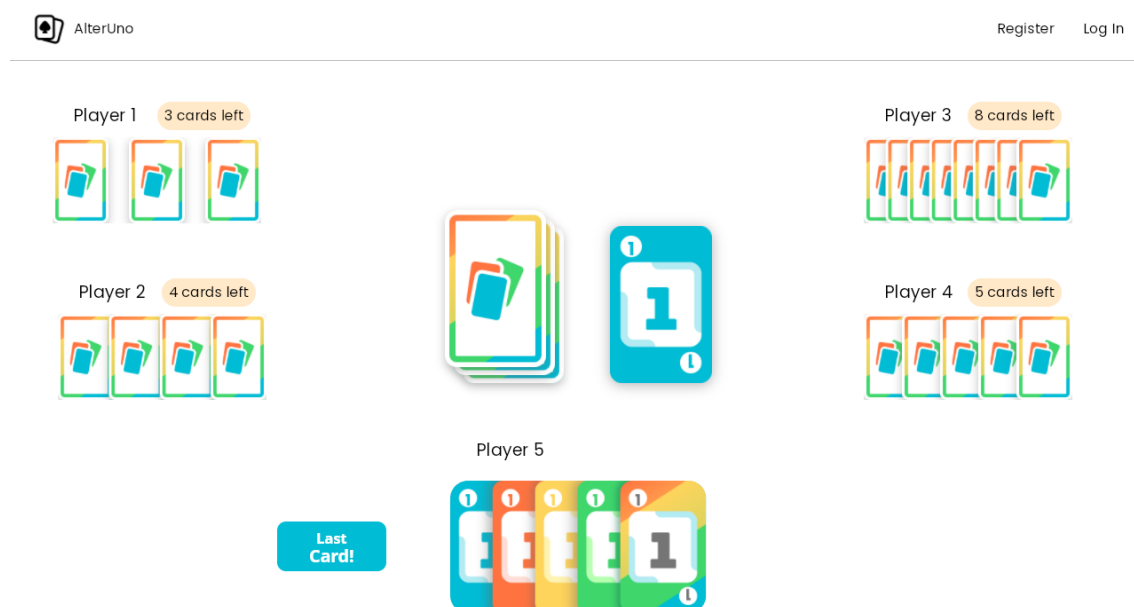


Figura 5.6: Mockup de la pantalla del juego

## CAPÍTULO 6

---

### Análisis de antecedentes y aportación realizada

---

## **CAPÍTULO 7**

---

### **Análisis temporal y costes de desarrollo**

---

#### **7.1– Análisis temporal**

#### **7.2– Costes de desarrollo**

## CAPÍTULO 8

---

### Comparación con otras alternativas

---

## CAPÍTULO 9

---

### Pruebas

---

## CAPÍTULO 10

---

**Manual**

---



---

## **Apéndices**

---

---

## Referencias

---

Autores, Varios. *Escuela Técnica Superior de Ingeniería Informática*. Fecha de consulta: 24 de Noviembre de 2014, 2014.

Bezoz, Javier. *The titlesec and titletoc packages*. *TexEmplares*, 8:283–298, 2007.

de Sousa, José Martínez. *Ortografía y ortotipografía del español actual*. Trea, 2004.