



*ugr*

Universidad  
de **Granada**

# Simulador 3D multijugador del Cubo de Rubik en Web

## **Autor**

Paula Hernández Lozano

## **Directores**

Carlos Ureña Almagro

<b>Simulador 3D multijugador del Cubo de Rubik en Web</b>	<b>1</b>
Autor	1
Directores	1
<b>1. Resumen y palabras clave</b>	<b>2</b>
<b>2. Resumen extendido y palabras clave en inglés</b>	<b>2</b>
<b>3. Motivación e introducción</b>	<b>2</b>
<b>4. Objetivos del trabajo</b>	<b>2</b>
<b>5. Resolución del trabajo</b>	<b>2</b>
5.1. Planificación y presupuesto	2
5.2. Análisis y diseño	3
5.2.1. Especificación de requerimientos funcionales	3
5.2.2. Especificación de requerimientos no funcionales	4
5.2.3. Metodología de desarrollo	5
5.2.4. Bocetos de la interfaz de usuario	5
5.2.5. Estructuras de datos fundamentales y desarrollos algorítmicos no triviales	6
5.3. Implementación y pruebas	6
<b>6. Conclusiones y vías futura</b>	<b>6</b>
<b>7. Bibliografía final</b>	<b>6</b>

1. Resumen y palabras clave
2. Resumen extendido y palabras clave en inglés
3. Motivación e introducción
4. Objetivos del trabajo
5. Resolución del trabajo
  - 5.1. Planificación y presupuesto

Fase	Tarea	Duración estimada	Fecha de inicio	Fecha de finalización
Planificación	Bocetos de interfaz de usuario	1 día	25/02/2023	26/02/2023
Planificación	Especificación de	1 semana	23/02/2023	2/03/2023

	requerimientos			
Diseño				

## 5.2. Análisis y diseño

### 5.2.1. Especificación de requerimientos funcionales

Se podrá:

- Crear una nueva partida. Los jugadores podrán crear una nueva partida. El software generará un código de la sala que deberá compartirse con los otros jugadores para unirse a la partida.
- Unirse a una partida. Los jugadores podrán unirse a una partida ya creada pero no abierta. La participación en la partida se realizará mediante el código de la sala generado.

Los estados de una partida pueden ser:

- Creada: Hay por lo menos un jugador dentro de la partida
- Disponible para abrir: Tiene entre 2 y n jugadores
- Abierta: se considera el inicio de la partida. Cuando la partida alcance este estado, cada jugador obtendrá el cubo de Rubix en una posición aleatoria (todos la misma posición) y se iniciará el cronómetro.
- Cerrada: Alguno de los jugadores ya ha terminado el cubo de Rubix y por lo tanto ha ganado.

Resolución del cubo de Rubix: Un cubo de Rubix se considerará resuelto cuando cada cara del cubo tenga un único color.

Ganador de la partida: El jugador que resuelva el cubo de Rubix en el menor tiempo será el ganador de la partida.

Empate: En caso de empate, se propondrá otra partida para desempatar entre los jugadores que hayan resuelto el cubo de Rubix en el mismo tiempo.

Los jugadores podrían mover el cubo de diferentes maneras. Por lo general, la forma más común es a través del uso del mouse, arrastrando el cursor para rotar una determinada sección del cubo en diferentes direcciones. Por ejemplo, si un jugador quiere rotar la sección frontal del cubo hacia la derecha, simplemente tendría que hacer clic y arrastrar el cursor hacia la derecha en esa sección; de manera que cuando soltase se produciría un giro de 90 grados. Por lo que, los movimientos de las 9 rebanadas posibles (en 2 sentidos cada una) se harían haciendo clic y arrastrando el cursor hacia la dirección deseada.

Otra forma de rotar las secciones del cubo, que sería una primera versión (previa al uso del ratón anteriormente comentado), es mediante el uso de botones a través de la página web.

(insertar foto cuando esté hecho)

En cualquier caso, los jugadores deben tener en cuenta que solo pueden rotar una sección del cubo a la vez, y que cada rotación afectará a las secciones adyacentes. Por lo tanto, deben planificar cuidadosamente sus movimientos y tener en cuenta cómo afectará a todo el cubo.

Los jugadores contarán con la opción de poder hacer replay en sus propios movimientos. Esto es útil para varios propósitos, como analizar los movimientos que han hecho para resolver el cubo de Rubik y mejorar su estrategia, o para compartir una grabación de sus movimientos con otros jugadores para mostrar cómo han resuelto el cubo.

El replay puede ser una característica muy útil para los jugadores, especialmente si están compitiendo entre sí o si están tratando de mejorar su tiempo de resolución. Al ver sus movimientos previos, pueden analizar lo que han hecho bien o mal, y así mejorar su técnica y su velocidad.

Además, proporciona la función de guardar una partida de un jugador la cual permite a los demás jugadores ver lo que ha hecho el jugador en su intento por resolver el cubo de Rubik. Esto significa que el jugador puede grabar su partida y compartirla con otros jugadores en la misma partida o incluso con otros jugadores en línea.

La función de guardar una partida es especialmente útil en un entorno multijugador, ya que permite a los jugadores aprender unos de otros y mejorar su técnica al ver cómo otros han abordado el cubo. Al guardar una partida, los jugadores pueden analizar los movimientos y estrategias que han utilizado otros jugadores para resolver el cubo, y así mejorar su propio enfoque.

Nota: El número mínimo de jugadores para una partida es de 2 y el máximo es variable (n).

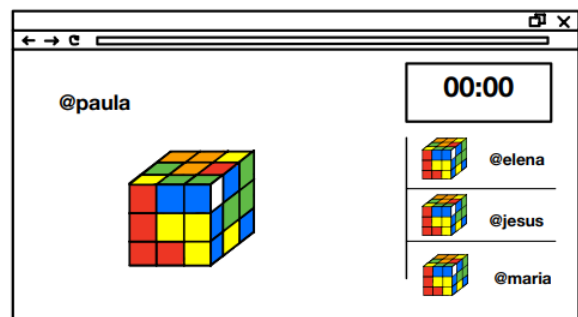
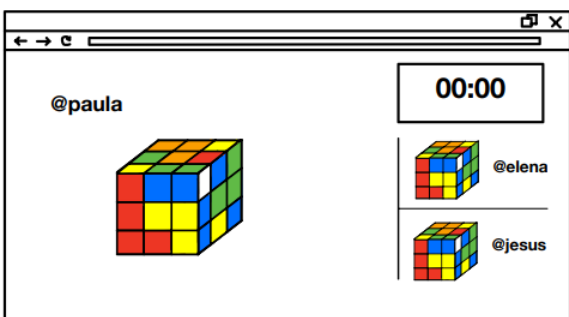
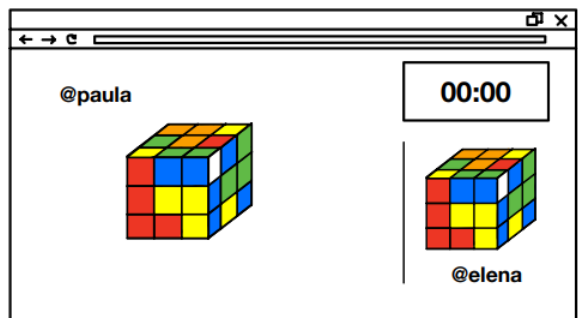
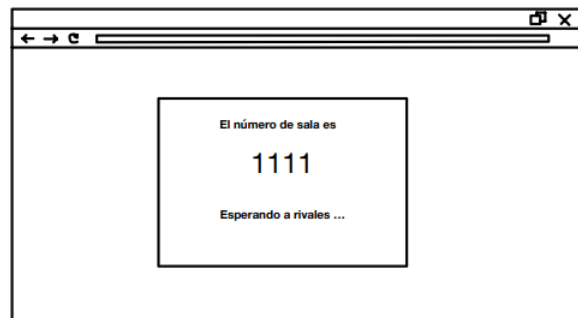
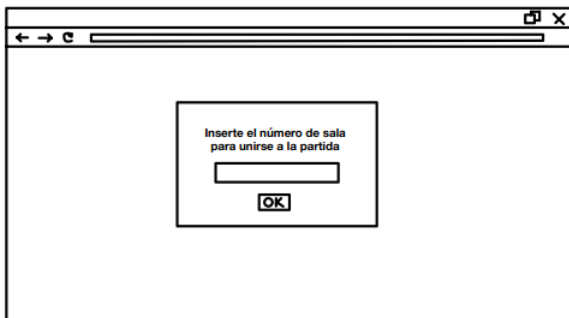
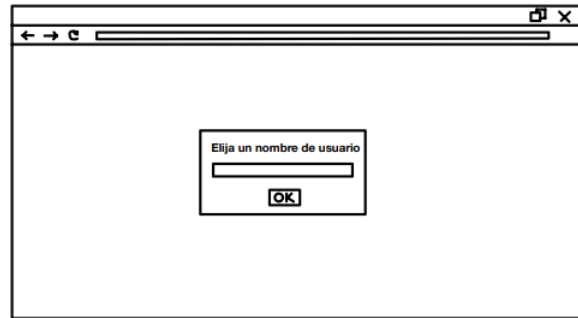
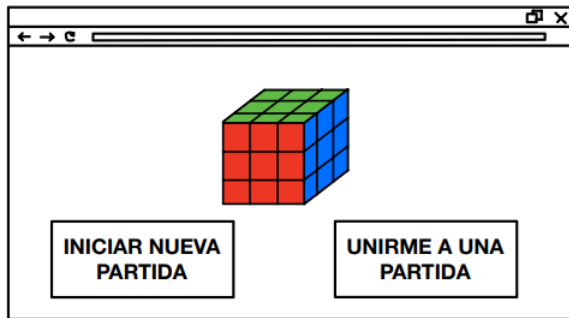
#### 5.2.2. Especificación de requerimientos no funcionales

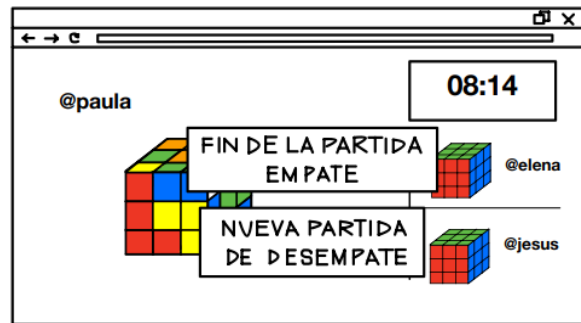
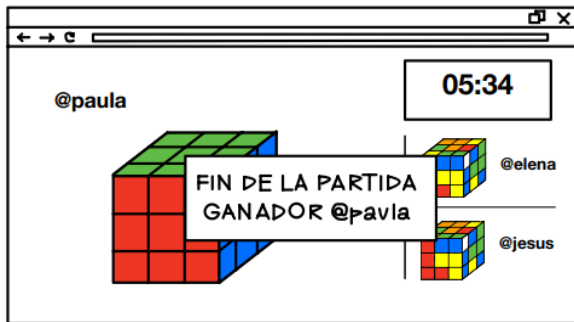
- Rendimiento: El simulador debe ser capaz de manejar una gran cantidad de usuarios simultáneamente sin retrasos significativos en la ejecución y sin afectar negativamente la experiencia del usuario.
- Usabilidad: El simulador debe ser fácil de usar e intuitivo para los usuarios, con una interfaz de usuario clara y concisa que permita la interacción con el cubo de Rubik y las funciones del simulador.
- Compatibilidad: El simulador debe ser compatible con una amplia gama de dispositivos y navegadores web, y debe funcionar de manera consistente en diferentes plataformas y sistemas operativos.

- Escalabilidad: El simulador debe ser escalable para manejar un aumento en el número de usuarios y el tamaño de los datos sin afectar negativamente el rendimiento o la funcionalidad.
- Mantenibilidad: El simulador debe ser fácil de mantener y actualizar, con una arquitectura modular y código limpio y bien documentado.

5.2.3. Metodología de desarrollo

5.2.4. Bocetos de la interfaz de usuario





5.2.5. Estructuras de datos fundamentales y desarrollos algorítmicos no triviales

5.3. Implementación y pruebas

6. Conclusiones y vías futura

7. Bibliografía final