



## **ISWD652 CALIDAD DE SOFTWARE Período 2024B**

<b>NOMBRE ESTUDIANTES:</b>	Said Luna Gabriel Jaya Edison Jimenez Damaris Suquillo Kerly Vizuite
<b>FECHA:</b>	08-11-2024
<b>TEMA:</b>	Proyecto-Version 1-SigmaSoftQA

### **APRENDIZAJE OBTENIDO**

Al comenzar el proyecto "Living Memory", nuestro equipo tenía conocimientos limitados sobre React y Phaser, las principales tecnologías de desarrollo que íbamos a emplear. React, aunque familiar en el ámbito del desarrollo web, era nuevo para algunos de nosotros en un contexto de aplicaciones interactivas como el juego que estábamos diseñando. Phaser, por otro lado, era completamente nuevo y supuso un reto inicial para todos. Durante el desarrollo del proyecto, logramos no solo entender los fundamentos de ambas herramientas, sino también implementarlas de manera efectiva para crear un juego interactivo y orientado a mejorar la calidad de vida de los pacientes.

El aprendizaje obtenido nos permitió adquirir habilidades fundamentales en el desarrollo de interfaces interactivas y en el uso de motores gráficos, mejorando nuestra comprensión sobre cómo estructurar un proyecto complejo y multidisciplinario. Este proyecto también fortaleció nuestra capacidad para colaborar en un equipo de desarrollo, coordinar tareas, y documentar adecuadamente nuestro progreso, todo lo cual contribuye a nuestra formación profesional en ingeniería de software.

### **DESARROLLO**

El proyecto "Living Memory" es un juego diseñado para ayudar a personas de la tercera edad con Alzheimer, enfocado en la mejora de habilidades de orientación espacial y praxias visoconstructivas. La meta del juego es que los pacientes colorean la parte faltante de un modelo presentado, bajo la supervisión de un psico-terapeuta. A través del juego, se registra una puntuación basada en los aciertos de color y el tiempo que toma un intento exitoso, lo cual permite a los terapeutas medir los avances en el rendimiento cognitivo de cada paciente.



## ISWD652 CALIDAD DE SOFTWARE Período 2024B

### Arquitectura y Diseño del Producto:

La estructura del proyecto "Living Memory" está organizada para mantener una clara separación entre los recursos estáticos y el código de la aplicación. En el directorio raíz, contamos con la carpeta public, que almacena archivos estáticos como index.html y elementos gráficos en Assets. El código fuente principal se encuentra en la carpeta src, donde los componentes están organizados en Components, facilitando el desarrollo modular y la reusabilidad. Los archivos principales, como App.js, index.js y las configuraciones de rutas en Routes.js, permiten estructurar la lógica y la navegación del juego.

### Historias de usuario:

<b>Nro:</b> HU-001	<b>Título:</b> Comenzar juego	<b>Prioridad:</b> Baja <b>Estimación:</b> 4 horas
<b>Historia de usuario:</b>  Como paciente, quiero ver cuadro que diga "Play" en la pantalla de inicio, para comenzar a pintar el dibujo.		
<b>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</b>  1. Dado que el paciente quiere comenzar a pintar, cuando el paciente mira el cuadro de "Play", entonces el cuadro debe estar claramente visible. 2. Dado que el paciente da clic en el cuadro, cuando el paciente hace clic en "Play", el usuario debe ser redirigido a la pantalla de dibujo.		
<b>Tareas de implementación:</b>  1. Diseñar la interfaz gráfica de presentación (2 horas) 2. Diseñar el cuadro de play para que sea agradable para el paciente (1 horas) 3. Implementar la lógica para redirección del cuadro a la pantalla de dibujo (1 horas).		

<b>Nro:</b> HU-002	<b>Título:</b> Seleccionar el color de la paleta	<b>Prioridad:</b> Alta <b>Estimación:</b> 6 horas
--------------------	--	--



## ISWD652 CALIDAD DE SOFTWARE Período 2024B

### Historia de usuario:

Como paciente, quiero seleccionar un color de la paleta de colores mostrada en la parte superior de la pantalla, para poder pintar las celdas de la figura con el color deseado.

### Sabré que he terminado cuando se cumplan todos los criterios de aceptación:

3. Dado que el paciente observa la paleta de colores, cuando el paciente selecciona la paleta de colores, entonces la paleta debe contener al menos 8 colores diferentes.
4. Dado que el paciente quiere pintar la figura, cuando ha seleccionado el color, entonces aplicará al hacer clic en una celda de la figura.

### Tareas de implementación:

4. Diseñar la interfaz gráfica del dibujo (2 horas)
5. Implementar la paleta de colores en la parte superior de la interfaz (2 horas).
6. Codificar la lógica para aplicar colores (2 horas).

<b>Nro:</b> HU-003	<b>Título:</b> Pintar el dibujo	<b>Prioridad:</b> Alta <b>Estimación:</b> 8 horas
<b>Historia de usuario:</b>  Como paciente, quiero pintar un dibujo en la aplicación para ayudar en mi proceso terapéutico.		
<b>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</b>  <ol style="list-style-type: none"> <li>1. Dado que el paciente selecciona un dibujo, cuando el paciente pinte del color correcto, entonces los cambios serán guardados correctamente.</li> <li>2. Dado que el paciente comete un error, cuando el paciente desee deshacerlo, entonces podrá volver a repintarlo.</li> </ol>		
<b>Tareas de implementación:</b>  <ol style="list-style-type: none"> <li>1. Diseñar la interfaz gráfica del dibujo (2 horas)</li> </ol>		



## ISWD652 CALIDAD DE SOFTWARE Período 2024B

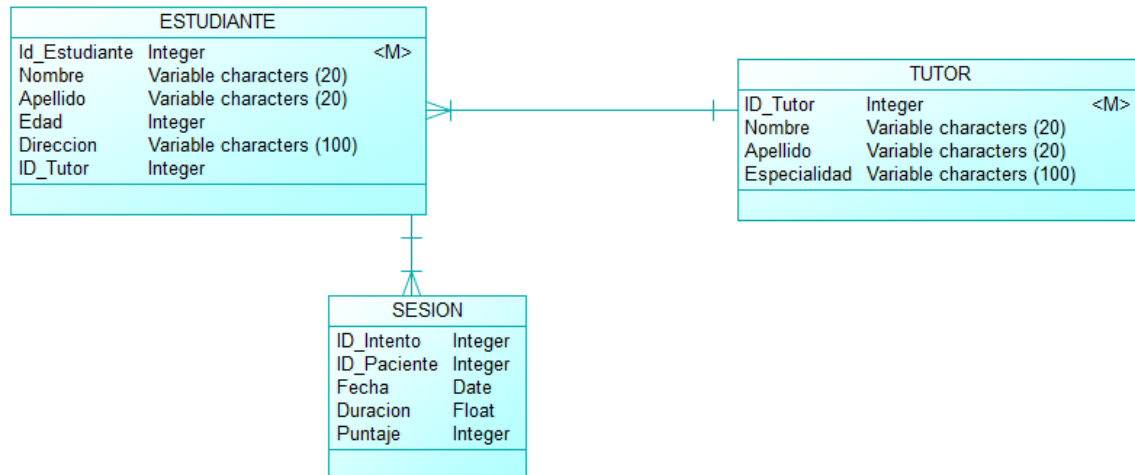
2. Diseñar la funcionalidad para pintar el dibujo (2 horas).
3. Codificar la lógica para aplicar colores y deshacer cambios (2 horas).
4. Probar la funcionalidad de pintar y deshacer con diferentes escenarios (2 horas).

<b>Nro:</b> 004	HU-	<b>Título:</b> Ver resultados del dibujo	<b>Prioridad:</b> Media <b>Estimación:</b> 6 horas
<b>Historia de usuario:</b>  Como paciente, quiero ver los resultados que tuve al finalizar el dibujo, para conocer mi avance			
<b>Sabré que he terminado cuando se cumplan todos los criterios de aceptación:</b>  <ol style="list-style-type: none"> <li>1. Dado que el paciente selecciona un dibujo, cuando termine de pintar el dibujo, entonces podrá ver los resultados.</li> <li>2. Dado que el paciente pinta el dibujo, cuando el paciente desee terminar el pintado, entonces aparecerá la pantalla de resultados.</li> </ol>			
<b>Tareas de implementación:</b>  <ol style="list-style-type: none"> <li>1. Diseñar la interfaz gráfica de resultados (2 horas)</li> <li>2. Diseñar la funcionalidad para que la interfaz gráfica sea visible al salir y finalizar el juego (2 horas).</li> <li>3. Codificar la lógica para obtener aciertos y errores que tuvo el paciente (2 horas).</li> </ol>			

### Modelo arquitectónico

Para el desarrollo de "Living Memory" se usarán las siguientes tablas para la base de datos:

## ISWD652 CALIDAD DE SOFTWARE Período 2024B



### Prototipos de interfaces de usuario:

La siguientes imágenes son prototipos de "Living Memory":

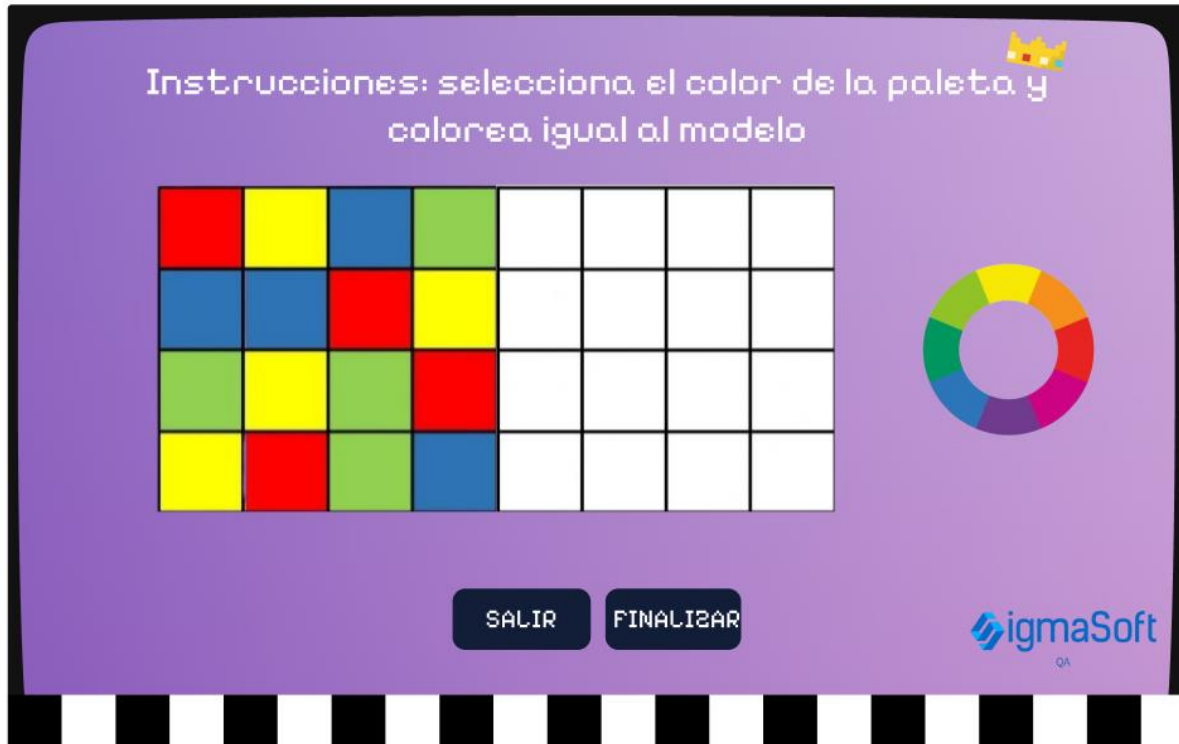


**ISWD652 CALIDAD DE SOFTWARE**  
**Período 2024B**





**ISWD652 CALIDAD DE SOFTWARE**  
**Período 2024B**





## ISWD652 CALIDAD DE SOFTWARE Período 2024B



### Herramientas y Tecnologías:

Para el desarrollo de "Living Memory" utilizamos principalmente JavaScript junto con React y Phaser, tecnologías que se complementan para crear aplicaciones interactivas. React facilita la creación de interfaces de usuario mediante componentes reutilizables, lo cual permite organizar de manera modular la interfaz del juego, mejorar la eficiencia y simplificar el proceso de actualización de la UI. Por otro lado, Phaser es el motor gráfico que usamos para implementar la lógica y los elementos visuales del juego, lo que nos permite manejar gráficos y animaciones de forma optimizada.

Además de estas tecnologías, utilizamos Git como herramienta de control de versiones. El control de versiones nos permite gestionar y registrar los cambios en el código fuente a lo largo del tiempo, facilitando el trabajo colaborativo y el almacenamiento seguro del proyecto en un repositorio compartido en GitHub. En nuestro flujo de trabajo, empleamos dos ramas: "main" y "develop". La





## **ISWD652 CALIDAD DE SOFTWARE Período 2024B**

rama "main" representa la versión estable y lista para producción del proyecto, mientras que "develop" es el espacio donde realizamos el desarrollo activo y probamos nuevas funcionalidades. Este enfoque modular permite que cada miembro del equipo trabaje en componentes específicos sin interferir en el trabajo de los demás, simplificando la integración de funcionalidades y el seguimiento del progreso del proyecto.

### **Resultados Obtenidos:**

En esta primera versión de "Living Memory", el desarrollo se centró en implementar la lógica principal del juego. Esto incluye la funcionalidad de colorear cuadros en la pantalla, graficar el dibujo en tiempo real y validar si el color seleccionado es el correcto según el modelo de referencia. Estas características permiten a los jugadores interactuar de forma intuitiva con el juego y proporcionan una base sólida para futuras mejoras en la jugabilidad y en la personalización de niveles de dificultad.

### **Enlace Repositorio:**

<https://github.com/KeiMagInc/alzheimer-game.git>

### **REFERENCIAS**

- [1] R. Arjona, "Beneficios de la estimulación cognitiva en personas con Alzheimer", Revista de Neurología, vol. 27, pp. 1-10, 2022.
- [2] P. C. Newton y H. E. Hernández, "Diseño y desarrollo de interfaces de usuario interactivas", IEEE Transactions on Human-Machine Systems, vol. 44, no. 5, pp. 578-591, 2020.
- [3] J. L. Moreno y C. A. Sánchez, Introducción a los motores gráficos para videojuegos, Editorial Alfa, 2019.
- [4] S. Vázquez, "Uso de videojuegos para la estimulación cognitiva en adultos mayores", IEEE Journal on Games and Simulation, vol. 5, pp. 122-130, 2023.